

SINUMERIK

SINUMERIK 840D sl/840Di sl/840D/840Di/810D Preparazione del lavoro

Manuale di programmazione

Valido per

Controlli numerici

SINUMERIK 840D sl/840DE sl
SINUMERIK 840Di sl/840DiE sl
SINUMERIK 840D powerline/840DE powerline
SINUMERIK 840Di powerline/840DiE powerline
SINUMERIK 810D powerline/810DE powerline

Software

NCU software di sistema per 840D sl/840DE sl	1.4
NCU software di sistema per 840Di sl/DiE sl	1.0
NCU software di sistema per 840D/840DE	7.4
NCU software di sistema per 840Di/840DiE	3.3
NCU software di sistema per 810D/810DE	7.4

11/2006

6FC5398-2BP10-2CA0

Premessa

Programmazione NC flessibile	1

Tecnica dei sottoprogrammi, tecnica delle macro	2

Gestione dei file e dei programmi	3

Settori di protezioni	4

Istruzioni di percorso speciali	5

Frame	6

Trasformazioni	7

Correzioni utensile	8

Comportamento sulla traiettoria	9

Azioni sincrone al movimento	10

Pendolamento	11

Punzonatura e roditura	12

Ulteriori funzioni	13

Programmi di sgrossatura personalizzati	14

Tabelle	15

Appendice	A

Istruzioni di sicurezza

Questo manuale contiene delle norme di sicurezza che devono essere rispettate per salvaguardare l'incolumità personale e per evitare danni materiali. Le indicazioni da rispettare per garantire la sicurezza personale sono evidenziate da un simbolo a forma di triangolo mentre quelle per evitare danni materiali non sono precedute dal triangolo. Gli avvisi di pericolo sono rappresentati come segue e segnalano in ordine decrescente i diversi livelli di rischio.



Pericolo

questo simbolo indica che la mancata osservanza delle opportune misure di sicurezza **provoca** la morte o gravi lesioni fisiche.



Avvertenza

il simbolo indica che la mancata osservanza delle relative misure di sicurezza **può causare** la morte o gravi lesioni fisiche.



Cautela

con il triangolo di pericolo indica che la mancata osservanza delle relative misure di sicurezza può causare lesioni fisiche non gravi.

Cautela

senza triangolo di pericolo indica che la mancata osservanza delle relative misure di sicurezza può causare danni materiali.

Attenzione

indica che, se non vengono rispettate le relative misure di sicurezza, possono subentrare condizioni o conseguenze indesiderate.

Nel caso in cui ci siano più livelli di rischio l'avviso di pericolo segnala sempre quello più elevato. Se in un avviso di pericolo si richiama l'attenzione con il triangolo sul rischio di lesioni alle persone, può anche essere contemporaneamente segnalato il rischio di possibili danni materiali.

Personale qualificato

L'apparecchio/sistema in questione deve essere installato e messo in servizio solo rispettando le indicazioni contenute in questa documentazione. La messa in servizio e l'esercizio di un apparecchio/sistema devono essere eseguiti solo da **personale qualificato**. Con riferimento alle indicazioni contenute in questa documentazione in merito alla sicurezza, come personale qualificato si intende quello autorizzato a mettere in servizio, eseguire la relativa messa a terra e contrassegnare le apparecchiature, i sistemi e i circuiti elettrici rispettando gli standard della tecnica di sicurezza.

Uso regolamentare delle apparecchiature/dei sistemi:

Si prega di tener presente quanto segue:



Avvertenza

L'apparecchiatura può essere destinata solo agli impieghi previsti nel catalogo e nella descrizione tecnica e può essere utilizzata solo insieme a apparecchiature e componenti di Siemens o di altri costruttori raccomandati o omologati dalla Siemens. Per garantire un funzionamento ineccepibile e sicuro del prodotto è assolutamente necessario che le modalità di trasporto, di immagazzinamento, di installazione e di montaggio siano corrette, che l'apparecchiatura venga usata con cura e che si provveda ad una manutenzione appropriata.

Marchio di prodotto

Tutti i nomi di prodotto contrassegnati con ® sono marchi registrati della Siemens AG. Gli altri nomi di prodotto citati in questo manuale possono essere dei marchi il cui utilizzo da parte di terzi per i propri scopi può violare i diritti dei proprietari.

Esclusione di responsabilità

Abbiamo controllato che il contenuto di questa documentazione corrisponda all'hardware e al software descritti. Non potendo comunque escludere eventuali differenze, non possiamo garantire una concordanza perfetta. Il contenuto di questa documentazione viene tuttavia verificato periodicamente e le eventuali correzioni o modifiche vengono inserite nelle successive edizioni.

Premessa

Prefazione

Documentazione SINUMERIK

La documentazione SINUMERIK è suddivisa in 3 livelli:

- Documentazione generale
- Documentazione per l'utente
- Documentazione per il costruttore / per il service

Un elenco delle pubblicazioni, con le rispettive lingue disponibili, viene aggiornato mensilmente e si trova in Internet sotto:

<http://www.siemens.com/motioncontrol>

Seguono le voci di menu "Supporto" → "Documentazione tecnica" → "Sommario delle pubblicazioni".

L'edizione internet del DOConCD, il DOConWEB, si trova nel sito:

<http://www.automation.siemens.com/doconweb>

Per informazioni sull'offerta di corsi e sulle FAQ (Frequently Asked Questions) consultare l'indirizzo Internet:

<http://www.siemens.com/motioncontrol> e poi nel menu "Support"

Destinatari

La presente documentazione è rivolta a:

- programmatori
- progettisti

Vantaggi

Con l'ausilio del Manuale di programmazione i destinatari hanno la possibilità di progettare, scrivere e testare programmi e interfacce software e di eliminare gli eventuali errori.

Configurazione standard

Nel presente manuale di programmazione è descritta la funzionalità delle prestazioni standard. Per le funzionalità aggiuntive o sostitutive apportate dal costruttore della macchina si veda la documentazione del costruttore della macchina.

Il controllore può contenere altre funzioni oltre a quelle descritte in questo manuale. Ciò non costituisce però obbligo di implementazione di tali funzioni in caso di nuove forniture o di assistenza tecnica.

Inoltre, per motivi di chiarezza, questa documentazione non riporta tutte le informazioni dettagliate relative alle varie esecuzioni del prodotto e non può nemmeno prendere in considerazione e trattare ogni possibile caso di montaggio, funzionamento e manutenzione.

Supporto tecnico

Per chiarimenti tecnici rivolgersi alla seguente hotline:

	Europa/Africa	Asia/Australia	America
Telefono	+49 180 5050 222	+86 1064 719 990	+1 423 262 2522
Fax	+49 180 5050 223	+86 1064 747 474	+1 423 262 2289
Internet	http://www.siemens.com/automation/support-request		
E-mail	mailto:adsupport@siemens.com		

Nota

Per i numeri telefonici dell'assistenza tecnica specifica dei vari paesi, vedere in Internet:

<http://www.siemens.com/automation/service&support>

Domande sulla documentazione

Per domande relative alla documentazione (suggerimenti, correzioni) inviare un fax o una E-Mail al seguente indirizzo:

Fax: +49 (0) 9131 / 98 - 63315

E-mail: <mailto:docu.motioncontrol@siemens.com>

Modulo fax: vedere il modulo in fondo al manuale

Indirizzo Internet per SINUMERIK

<http://www.siemens.com/sinumerik>

Dichiarazione di conformità CE

La dichiarazione di conformità UE alla direttiva EMC si trova:

- in Internet:

<http://www.ad.siemens.de/csinfo>

con il numero di ordinazione del prodotto 15257461

- presso la filiale competente dell'area A&D MC di Siemens AG

Versione export

La variante per l'esportazione non comprende le seguenti funzioni:

Funzione	810DE	840DE sl	840DE	840DiE sl	840DiE
Interpolazione elicoidale 2D+6 (Versione di base, senza opzioni)	–	–	–	–	–
Pacchetto di elaborazione Fresatura	–	–	–	–	–
Pacchetto per lavorazioni in 5 assi	–	–	–	–	–
Pacchetto di trasformazione per l'handling	–	–	–	–	–
Interpolazione multiasse (>4 assi interpolanti)	–	–	–	–	–
Cicli compilati OA-NCK	–	–	–	–	–
Regolazione della distanza 1D/3D nel clock LR ¹⁾	–	–	–	–	–
Azioni sincrone ¹⁾ (Versione di base, senza opzioni)	#	#	#	#	#
Accoppiamento del valore master ed interpolazione tabellare	#	#	#	#	#
Compensazione della flessione multidimensionale	#	#	#	#	#
Misure sincrone livello 2 ¹⁾	–	–	#	–	#
Riduttore elettronico ¹⁾	–	–	#	–	#
Transfer elettronico	–	–	#	–	#
	# Funzionalità limitata				
	– Funzione non possibile				

¹⁾ Le limitazioni funzionali per le varianti export SINUMERIK 810DE powerline / SINUMERIK 840DE sl / SINUMERIK 840DE powerline / SINUMERIK 840DiE sl / SINUMERIK 840DiE powerline sono limitate a "max. 4 assi interpolanti".

Descrizione

Concetti fondamentali

Il manuale di programmazione "Concetti fondamentali" serve all'operaio specializzato addetto alla macchina e presuppone adeguate conoscenze nell'ambito della foratura, fresatura e tornitura. Sulla base di semplici esempi di programmazione vengono spiegati i comandi e le istruzioni note anche dalla norma DIN 66025

Preparazione del lavoro

Il manuale di programmazione "Preparazione del lavoro" informa il tecnico su tutte le possibilità di programmazione esistenti. Grazie ad un linguaggio di programmazione speciale, il SINUMERIK 840D sl/840Di sl/840D/840Di/810D consente la programmazione di un programma pezzo complesso (ad es. superfici a forma libera, coordinamento dei canali,...) e rende meno impegnative le programmazioni per il tecnico.

I comandi e le istruzioni descritti nel presente Manuale di programmazione sono indipendenti dalla tecnologia.

Essi possono essere utilizzati ad esempio per:

- Tornitura, fresatura e rettifica
- Macchine cicliche (settore dell'imballaggio, lavorazione del legno)
- Controlli laser

Indice

	Premessa	3
1	Programmazione NC flessibile.....	15
1.1	Variabili e parametri di calcolo (variabili definite dall'utente, parametri di calcolo, variabili di sistema).....	15
1.2	Definizione delle variabili (DEF variabili definite dall'utente LUD, GUD, PUD)	17
1.3	Definizioni dei campi (DEF, SET, REP)	21
1.4	Programmazione indiretta	28
1.4.1	Elaborazione di una stringa come riga del partprogram (EXECSTRING).....	31
1.5	Assegnazioni.....	32
1.6	Operazioni/funzioni di calcolo	33
1.7	Operazioni logiche e di confronto	36
1.7.1	Correttore di precisione in caso di errori di confronto (TRUNC).....	39
1.7.2	Minimo, massimo e campo delle variabili (MINVAL, MAXVAL e BOUND)	40
1.8	Priorità delle operazioni	42
1.9	Possibili conversioni di tipi	43
1.10	Operazioni su stringhe	44
1.10.1	Conversione di tipi in STRING	45
1.10.2	Conversione di tipi da STRING.....	46
1.10.3	Concatenamento di stringhe	47
1.10.4	Conversione in minuscolo/maiuscolo	48
1.10.5	Lunghezza della stringa	49
1.10.6	Ricerca di un carattere/stringa nella stringa	49
1.10.7	Selezione di una parte di stringa.....	51
1.10.8	Selezione di un singolo carattere.....	51
1.11	Istruzione CASE.....	53
1.12	Strutture di controllo.....	55
1.13	Coordinamento dei programmi	59
1.14	Routine di interrupt (SETINT, DISABLE, ENABLE, CLRINT).....	64
1.15	Scambio assi, scambio mandrini (RELEASE, GET, GETD).....	73
1.16	Spostamento di un asse in un altro canale (AXTOCHAN)	77
1.17	NEWCONF: attivazione di dati macchina	78
1.18	WRITE: Scrittura di file.....	79
1.19	DELETE: cancella il file.....	81
1.20	READ: lettura di righe nel file.....	83
1.21	ISFILE: File esistente nella memoria utente dell'NCK.....	85

1.22	FILEDATE/TIME/SIZE/STAT/INFO: informazioni file	86
1.23	CHECKSUM: Creazione del checksum per un campo	88
1.24	ROUNDUP: arrotondamento.....	89
2	Tecnica dei sottoprogrammi, tecnica delle macro	91
2.1	Utilizzo dei sottoprogrammi.....	91
2.2	Sottoprogrammi con meccanismo SAVE	94
2.3	Sottoprogrammi con assegnazione di parametri (PROC, VAR)	95
2.4	Richiamo di sottoprogrammi (L oppure EXTERN).....	99
2.5	Salto di ritorno parametrizzabile di un sottoprogramma (RET).....	104
2.6	Sottoprogramma con ripetizione del programma (P).....	108
2.7	Sottoprogramma modale (MCALL)	110
2.8	Richiamo indiretto di sottoprogrammi (CALL)	112
2.9	Ripetizione di una parte di programma con programmazione indiretta (CALL).....	112
2.10	Richiamo indiretto di un programma programmato in linguaggio ISO (ISOCALL)	114
2.11	Richiamo sottoprogramma con indicazione del percorso e parametri (PCALL).....	115
2.12	Estensione del percorso di ricerca nei richiami di sottoprogramma con CALLPATH.....	116
2.13	Esecuzione di un sottoprogramma esterno (EXTCALL).....	118
2.14	Esclusione blocco singolo (SBLOF, SBLON)	122
2.15	Disattivazione della visualizzazione del blocco attuale (DISPLOF).....	126
2.16	Identificazione di sottoprogrammi con preparazione (PREPRO).....	128
2.17	Cicli: Parametrizzazione dei cicli utente.....	129
2.18	Tecnica delle macro (DEFINE...AS)	132
3	Gestione dei file e dei programmi	135
3.1	Memoria dei programmi	135
3.2	Memoria di lavoro.....	140
3.3	Definizione dei dati utente.....	142
3.4	Livelli di protezione per i dati utente, DM, DS e istruzioni NC	145
3.4.1	Definizione dei livelli di protezione per i dati utente (GUD).....	145
3.4.2	Attivazione automatica di GUD e MAC	147
3.4.3	Modificare i livelli di protezione per dati macchina e dati di setting (REDEF, DM, DS)	148
3.4.4	Livelli di protezione per istruzioni NC (REDEF)	150
3.5	REDEF: Modifica degli attributi degli elementi di linguaggio NC	152
3.6	Istruzione di strutturazione SEFORM nell'editor Step.....	158
4	Settori di protezioni	159
4.1	Definizione dei settori di protezione (CPROTDEF, NPROTDEF).....	159
4.2	Attivazione e disattivazione dei settori di protezione (CPROT, NPROT)	162
4.3	Verifica del superamento del settore di protezione, limitazione del campo di lavoro e finecorsa software	166

5	Istruzioni di percorso speciali.....	173
5.1	Raggiungimento di posizioni codificate (CAC, CIC, CDC, CACP, CACN)	173
5.2	Interpolazione Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN)	174
5.3	Raggruppamento spline (SPLINEPATH)	183
5.4	Compressore (COMPOF/ON, COMPCURV, COMPCAD)	185
5.5	Interpolazione polinomiale (POLY, POLYPATH)	189
5.6	Riferimento vettoriale impostabile, (SPATH, UPATH)	195
5.7	Misura con tastatore a commutazione (MEAS, MEAW)	197
5.8	Funzione di misura ampliata (MEASA, MEAWA, MEAC) (opzione)	200
5.9	Funzioni speciali per l'utente OEM (OEMIPO1, OEMIPO2, G810 ... G829)	209
5.10	Riduzione dell'avanzamento con rallentamento sugli spigoli (FENDNORM, G62, G621)	209
5.11	Criterio programmabile di fine movimento (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)	210
5.12	Blocco parametri servo programmabile (SCPARA)	214
6	Frame	217
6.1	Trasformazione di coordinate con variabili Frame	217
6.1.1	Variabili frame predefinite (\$P_BFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME)	219
6.2	Variabili frame / assegnazione di valori frame	224
6.2.1	Assegnazione di valori diretti (valore asse, angolo, scala)	224
6.2.2	Leggere e modificare le componenti frame (TR, FI, RT, SC, MI)	226
6.2.3	Combinazione di frame completi	228
6.2.4	Definizione di nuovi frame (DEF FRAME)	229
6.3	Traslazione fine e grossolana (CFINE; CTRANS)	230
6.4	Spostamento DRF	232
6.5	Spostamento origine esterno	233
6.6	Traslazione del preset (PRESETON)	234
6.7	Calcolo del frame in base a 3 punti di misura nello spazio (MEAFRAME)	235
6.8	Frame NCU globali	238
6.8.1	Frame specifici per canali (\$P_CHBFR, \$P_UBFR)	239
6.8.2	Frame attivi nel canale	241
7	Trasformazioni.....	247
7.1	Programmazione generale dei tipi di trasformazione	247
7.1.1	Movimenti di orientamento nelle trasformazioni	249
7.1.2	Panoramica della trasformazione dell'orientamento TRAORI	253
7.2	Trasformazione a tre, quattro e cinque assi (TRAORI)	255
7.2.1	Relazioni generali della testa portautensili cardanica	255
7.2.2	Trasformazione a tre, quattro e cinque assi (TRAORI)	258
7.2.3	Varianti della programmazione dell'orientamento e della posizione base (OTIRESET)	259
7.2.4	Programmazione dell'orientamento utensile (A..., B..., C..., LEAD, TILT)	261
7.2.5	Fresatura frontale (fresatura 3D A4, B4, C4, A5, B5, C5)	267
7.2.6	Riferimento degli assi di orientamento (ORIWKS, ORIMKS)	268
7.2.7	Programmazione degli assi di orientamento (ORIAxes, ORIVect, ORIEuler, ORIRPY)	271

7.2.8	Programmazione dell'orientamento lungo una superficie conica (ORIPLANE, ORICONxx)....	273
7.2.9	Impostazione dell'orientamento di due punti di contatto (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)	276
7.3	Polinomi di orientamento (PO[angolo], PO[coordinate]).....	279
7.4	Rotazioni dell'orientamento utensile (ORIROTA, ORIROTR/TT, ORIROTC, THETA).....	281
7.5	Orientamenti relativi al percorso	283
7.5.1	Tipi di orientamento relativi al percorso	283
7.5.2	Rotazione relativa al percorso dell'orientamento dell'utensile (ORIPATH, ORIPATHS, angolo di rotazione).....	285
7.5.3	Interpolazione relativa al percorso della rotazione dell'utensile (ORIROTC, THETA).....	286
7.5.4	Spianamento del percorso di orientamento (ORIPATHS A8=, B8=, C8=)	288
7.6	Compressione dell'orientamento COMPON (A..., B..., C..., THETA)	289
7.7	Correzione della lunghezza utensile on line (TOFFON, TOFFOF).....	293
7.8	Trasformazione cinematica	295
7.8.1	Fresatura su pezzi torniti (TRANSMIT).....	295
7.8.2	Trasformazione della superficie esterna del cilindro (TRACYL)	299
7.8.3	Asse inclinato (TRAANG).....	307
7.8.4	Programmazione dell'asse inclinato (G05, G07)	310
7.9	Movimento cartesiano PTP	312
7.9.1	PTP con TRANSMIT	316
7.10	Condizioni marginali per l'attivazione di una trasformazione	320
7.11	Disattivazione della trasformazione (TRAFOOF)	321
7.12	Trasformazioni concatenate (TRACON, TRAFOOF).....	321
7.13	Assi geometrici commutabili (GEOAX)	323
8	Correzioni utensile	327
8.1	Memoria di correzione.....	327
8.2	Istruzioni per la gestione utensili	330
8.3	Correzione utensile on-line (PUTFTOCF, PUTFTOC, FTOCON, FTOCOF)	333
8.4	Mantenimento costante della correzione del raggio utensile (CUTCONON)	338
8.5	Attivazione delle correzioni utensili 3D (CUT3DC..., CUT3DF...)	340
8.5.1	Attivazione delle correzioni utensili 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF).....	340
8.5.2	Correzione raggio utensile 3D: fresatura periferica e frontale	342
8.5.3	Tipi di utensile/cambio utensile con dimensioni modificate (G40, G41, G42)	343
8.5.4	Correzione sulla traiettoria, curvatura della traiettoria, profondità di inserzione ISD e posizione utensile (CUT3DC).....	345
8.5.5	Spigolo interno/esterno e andamento sul punto di intersezione (G450/G451).....	347
8.5.6	Fresatura periferica 3D con superfici limite, impiego generico	349
8.5.7	Considerazione di una superficie limite (CUT3DCC, CUT3DCCD).....	349
8.6	Orientamento dell'utensile (ORIC, ORID, OSOF, OSC, OSS, OSSE, OSD, OST).....	353
8.7	Assegnazione di un numero D libero, numero del tagliente	359
8.7.1	Assegnazione di un numero D libero, numero del tagliente (indirizzo CE)	359
8.7.2	Verifica del numero D (CHKDNO)	360
8.7.3	Rinominare i numeri D (GETDNO, SETDNO)	360
8.7.4	Determinazione del numero T per il numero D indicato (GETACTTD)	361

8.7.5	Definizione di numero D come non valido (DZERO)	362
8.8	Cinematica del portautensile.....	363
9	Comportamento sulla traiettoria	369
9.1	Controllo tangenziale (TANG, TANGON, TANGOF, TANGDEL)	369
9.2	Trascinamento (TRAILON, TRAILOF)	375
9.3	Tabelle di curve (CTAB).....	379
9.3.1	Tabelle delle curve: relazioni generiche	379
9.3.2	Funzioni principali delle tabelle di curve (CTABDEF, CATBEND, CTABDEL)	380
9.3.3	Forme delle tabelle di curve (CTABDEL, CTABNOMEM, CTABFNO, CTABID, CTABLOCK, CTABUNLOCK).....	385
9.3.4	Comportamento sui bordi delle tabelle di curve (CTABTSV, CATBTSP, CTABMIN, CTABMAX).....	390
9.3.5	Accesso alle posizioni ed ai segmenti delle tabelle di curve (CTAB, CTABINV, CTABSSV, CATBSEV)	394
9.4	Accoppiamento assiale tramite valore master (LEADON, LEADOF)	397
9.5	Andamento dell'avanzamento (FNORM, FLIN, FCUB, FPO).....	403
9.6	Esecuzione del programma con buffer di preelaborazione (STARTFIFO, STOPFIFO, STOPRE)	410
9.7	Interruzioni condizionate delle sezioni di programma (DELAYFSTON, DELAYFSTOF)	412
9.8	Inibire la zona di programma per SERUPRO (IPTRLOCK, IPTRUNLOCK)	417
9.9	Riaccostamento al profilo (REPOSA/L, REPOSQ/H, RMI, RMN, RMB, RME)	419
10	Azioni sincrone al movimento	429
10.1	Struttura, concetti base generici	429
10.1.1	Programmazione ed elementi di comando	431
10.1.2	Campo di validità: numero identificativo ID.....	432
10.1.3	Verifica ciclica della condizione	433
10.1.4	Azioni	435
10.2	Operatori per condizioni e azioni	436
10.3	Variabili di esecuzione per le azioni sincrone	437
10.3.1	Informazioni generali sulle variabili di sistema.....	437
10.3.2	Conversione implicita del tipo	438
10.3.3	Variabili GUD per azioni sincrone	440
10.3.4	Identificatore asse di default (NO_AXIS)	441
10.3.5	Marker per azioni sincrone \$AC_MARKER[n]	442
10.3.6	Parametri per azioni sincrone \$AC_PARAM[n]	443
10.3.7	Parametri di calcolo \$R[n]	444
10.3.8	Lettura/scrittura dei dati macchina NC e dei dati setting NC	445
10.3.9	Variabile timer \$AC_Timer[n]	446
10.3.10	Variabile FIFO \$AC_FIFO1[n] ... \$AC_FIFO10[n].....	447
10.3.11	Informazione sui tipi di blocco nell'interpolatore	449
10.4	Azioni nelle azioni sincrone.....	451
10.4.1	Panoramica	451
10.4.2	Emissione di funzioni ausiliarie	454
10.4.3	Impostazione del blocco lettura (RDISABLE)	455
10.4.4	Disattivazione stop preelaborazione (STOPREOF).....	456
10.4.5	Cancellazione percorso residuo (DELDTG)	456
10.4.6	Definizione dei polinomi (FCTDEF)	458

10.4.7	Funzione sincrona (SYNFCT)	460
10.4.8	Regolazione della distanza con correzione limitata \$AA_OFF_MODE	463
10.4.9	Correzione utensile on-line (FTOC)	465
10.4.10	Correzione online della lunghezza utensile (\$AA_TOFF[Direzione_utensile])	467
10.4.11	Movimenti di posizionamento	468
10.4.12	Posizionamento dell'asse (POS)	469
10.4.13	Posizione nell'intervallo di riferimento predefinito (POSRANGE)	470
10.4.14	Start/stop asse (MOV)	471
10.4.15	Scambio assi (RELEASE, GET)	472
10.4.16	Avanzamento assiale (FA)	476
10.4.17	Finecorsa SW	476
10.4.18	Coordinamento assi	476
10.4.19	Impostazione del valore reale (PRESETON)	477
10.4.20	Movimenti del mandrino	478
10.4.21	Trascinamento (TRAILON, TRAILOF)	479
10.4.22	Accoppiamento al valore master (LEADON, LEADOF)	481
10.4.23	Misura (MEAWA, MEAC)	483
10.4.24	Inizializzazione delle variabili di campo con SET, REP	484
10.4.25	Impostazione/cancellazione delle label di attesa con SETM, CLEARM	485
10.4.26	Reazioni agli errori in caso di allarmi dei cicli SETAL	485
10.4.27	Posizionamento su riscontro fisso (FXS e FOCON/FOCOF)	486
10.4.28	Definizione dell'angolo della tangente vettoriale nella azioni sincrone	488
10.4.29	Determinazione dell'override attuale	488
10.4.30	Rilevazione del sovraccarico tramite il fabbisogno di tempo delle azioni sincrone	489
10.5	Cicli tecnologici	491
10.5.1	Variabile di contesto (\$P_TECCYCLE)	493
10.5.2	Parametri Call by Value	494
10.5.3	Inizializzazione dei parametri di default	495
10.5.4	Controllo ed esecuzione di cicli tecnologici (ICYCOF, ICYCON)	495
10.5.5	Strutturazioni in cascata dei cicli tecnologici	496
10.5.6	Cicli tecnologici in azioni sincrone blocco per blocco	497
10.5.7	Costrutti di controllo IF	497
10.5.8	Istruzioni di salto (GOTO, GOTOF, GOTOB)	497
10.5.9	Blocco, abilitazione, interruzione (LOCK, UNLOCK, RESET)	498
10.6	Cancellazione dell'azione sincrona (CANCEL)	499
10.7	Condizioni marginali	500
11	Pendolamento	505
11.1	Pendolamento asincrono	505
11.2	Pendolamento gestito da azioni sincrone	510
12	Punzonatura e roditura	519
12.1	Attivazione, disattivazione	519
12.1.1	Punzonatura e roditura ON oppure OFF (SPOF, SON, PON, SONS, PONS, PDELAYON/OF)	519
12.2	Preparazione automatica del percorso	522
12.2.1	Suddivisione del percorso per assi di contornitura	525
12.2.2	Suddivisione del percorso per singoli assi	527

13	Ulteriori funzioni.....	529
13.1	Funzioni per gli assi (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING)	529
13.2	Verifica del set di istruzioni del linguaggio NC presente (STRINGIS)	531
13.3	Richiamo della funzione ISVAR () e lettura dei dati macchina indice dell'array	535
13.4	Apprendimento delle curve di compensazione (QECLRNON, QECLRNOF)	538
13.5	Mandrino sincrono	540
13.5.1	Mandrini sincroni (COUPDEF, COUPDEL, COUPON/ONC, COUPOF/OFS, COUPRES)	540
13.6	Cambio elettronico (EG)	552
13.6.1	Definizione del cambio elettronico (EGDEF)	552
13.6.2	Attivazione del cambio elettronico (EGON)	553
13.6.3	Disattivazione del cambio elettronico (EGOFS)	557
13.6.4	Avanzamento al giro (G95)/cambio elettronico (FPR).....	558
13.7	Funzione ampliata di arresto e svincolo	558
13.7.1	Reazioni autarchiche degli azionamenti a ESR.....	560
13.7.2	Reazioni allo svincolo controllate dall'NC	562
13.7.3	Reazioni all'arresto controllate dall'NC	566
13.7.4	Funzionamento generatorico/mantenimento del circuito intermedio	566
13.7.5	Arresto autarchico tramite azionamento	567
13.7.6	Svincolo autarchico tramite azionamento	568
13.8	Comunicazione link.....	568
13.8.1	Accesso ad un settore di memoria globale della NCU	569
13.9	Container assi (AXCTWE, AXCTWED)	570
13.10	Tempo di esecuzione del programma/contatore pezzi	573
13.10.1	Informazioni generali.....	573
13.10.2	Tempo di esecuzione del programma	573
13.10.3	Funzione di conteggio dei pezzi.....	574
13.11	Istruzione per il richiamo interattivo delle maschere dal partprogram (MMC):	576
13.12	Influenza della gestione dei movimenti.....	577
13.12.1	Correzione percentuale dello strappo (JERKLIM)	577
13.12.2	Correzione percentuale della velocità (VELOLIM).....	578
13.13	Raggruppamento Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)	579
14	Programmi di sgrossatura personalizzati	583
14.1	Funzioni di supporto per l'asportazione	583
14.2	Preparazione del profilo (CONTPRON).....	584
14.3	Decodifica del profilo (CONTDCON)	591
14.4	Punto di intersezione di due elementi del profilo (INTERSEC)	595
14.5	Esecuzione di un elemento di profilo dalla tabella (EXECTAB)	597
14.6	Calcolo dei dati del cerchio (CALCDAT).....	597
15	Tabelle.....	599
15.1	Lista istruzioni	599

A	Appendice.....	641
A.1	Elenco delle abbreviazioni	642
A.2	Informazioni specifiche per la pubblicazione	647
A.2.1	Modulo per le correzioni - Modulo fax.....	647
A.2.2	Panoramica della documentazione	649
	Glossaire	651
	Indice analitico	675

Programmazione NC flessibile

1.1 Variabili e parametri di calcolo (variabili definite dall'utente, parametri di calcolo, variabili di sistema)

Funzione

Utilizzando le variabili invece di valori fissi, è possibile gestire un programma in modo flessibile. È possibile così reagire a segnali come ad esempio valori di misura oppure, utilizzando variabili come riferimento, si può sfruttare lo stesso programma per diverse geometrie.

Utilizzando calcoli con variabili e salti di programma, il programmatore ha la possibilità di creare archivi di programma flessibili e ridurre notevolmente l'onere della programmazione.

Tipi di variabili

Il controllo numerico distingue tre tipi di variabili:

Variabili definite dall'utente	Variabili definite dall'utente con nome e tipo, es. parametri di calcolo.
Parametri di calcolo	Variabili di calcolo predefinite per le quali è previsto l'indirizzo R seguito da un numero. Le variabili di calcolo predefinite sono del tipo REAL.
Variabile di sistema	Variabili messe a disposizione dal controllo numerico che possono essere elaborate nel programma (scrittura, lettura). Le variabili di sistema consentono l'accesso a spostamenti origine, correzioni utensile, valori reali, valori di misura degli assi, stati del controllo numerico etc. (per il significato delle variabili di sistema vedere l'appendice).

Tipi di variabili

Tipo	Significato	Campo di valori
INT	valori interi (integer) con segno	$\pm(2^{31} - 1)$
REAL	Numeri reali (numeri con punto decimale, LONG REAL secondo IEEE)	$\pm(10^{-300} \dots 10^{+300})$

BOOL	valori logici: TRUE (1) e FALSE (0)	1.0
CHAR	Carattere ASCII, codice corrispondente	0 ... 255
STRING	sequenza di caratteri, numero dei caratteri in [...], max. 200 caratteri	Sequenza di valori con 0 ... 255
AXIS	solo nomi degli assi (indirizzi assi)	tutti gli identificatori degli assi presenti nel canale
FRAME	impostazioni geometriche per traslazioni, rotazioni, fattori di scala, specularità, vedere capitolo "Frames"	

Variabili di calcolo

Con l'indirizzo R, se non diversamente definito, sono disponibili come standard 100 variabili di calcolo del tipo REAL.

Il numero esatto di variabili di calcolo (massimo 32535) viene stabilito tramite i dati macchina.

Esempio: R10=5

Variabile di sistema

Il controllo numerico mette a disposizione variabili di sistema che possono essere utilizzate ed elaborate da tutti i programmi in esecuzione.

Le variabili di sistema forniscono stati della macchina e del controllo. A volte non possono essere scritte.

Panoramica sulle variabili di sistema

Per una identificazione particolare, il nome delle variabili di sistema inizia sempre con il carattere "\$". Seguono poi le definizioni specifiche.

1. Lettera dell'alfabeto	Significato
\$M	Dati macchina
\$S	Dati setting
\$T	Dati di gestione utensili
\$P	Valori programmati
\$A	Valori attuali
\$V	Dati di service
2. Lettera dell'alfabeto	Significato
N	NCK-globale
C	specifici per canali
A	specifici per asse

Esempio: \$AA_IM

Significa: valore reale attuale dell'asse nel sistema di coordinate macchina.

1.2 Definizione delle variabili (DEF variabili definite dall'utente LUD, GUD, PUD)

Funzione

Oltre alle variabili predefinite il programmatore può definire variabili proprie e preimpostarle con valori.

Le variabili locali (LUD) sono valide solo nel programma in cui sono definite.

Le variabili globali (GUD) sono valide in tutti i programmi.

Tramite il dato macchina, le variabili utente locali (LUD) definite nel programma principale vengono ridefinite come variabili utente globali del programma (PUD), vedere esempio applicativo.

Costruttore della macchina

vedere i dati del costruttore della macchina.

Programmazione

Variabili del tipo INT

DEF INT name

oppure

DEF INT name=valore

Variabili del tipo REAL

DEF REAL name

oppure

DEF REAL name1,name2=3,name4

oppure

DEF REAL name[indice di campo1,indice di campo2]

Variabili del tipo BOOL

DEF BOOL name

Variabili del tipo CHAR

DEF CHAR name

oppure

DEF CHAR name[indice di campo]=("A","B",...)

Variabili del tipo STRING

DEF STRING[lunghezza stringa] name

Variabili del tipo AXIS

DEF AXIS name

oppure

DEF AXIS name[indice di campo]

Variabili del tipo FRAME

DEF FRAME name

Nota

Se in fase di definizione alle variabili non si è attribuito alcun valore, ad esse viene automaticamente assegnato il valore zero.

Le variabili devono essere definite all'inizio del programma prima di essere impiegate. La definizione deve avvenire in un blocco a sé stante; per ogni blocco può essere definito un solo tipo di variabile.

Parametri

INT	Tipo di variabile Integer, con numero intero
REAL	Tipo di variabile Real, numero con punto decimale
BOOL	Tipo di variabile Bool, 1 oppure 0 (TRUE oppure FALSE)
CHAR	Tipo di variabile Char, un carattere corrispondente al codice ASCII (0 ... 255)
STRING	Tipo di variabile String, sequenza di caratteri
AXIS	Tipo di variabile Axis, indirizzi assi e mandrini
FRAME	Tipo di variabile Frame, dati geometrici

Esempio

Tipo di variabile	Significato
INT	
DEF INT ANZAHL	Viene definita una variabile del tipo Integer con nome ANZAHL. Preimpostata dal sistema con zero.
DEF INT ANZAHL=7	Viene definita una variabile del tipo Integer con nome ANZAHL. La variabile ha il valore iniziale 7.
REAL	
DEF REAL TIEFE	Viene creata una variabile del tipo Real con il nome TIEFE. Preimpostazione di sistema con zero (0.0).
DEF REAL TIEFE=6.25	Viene creata una variabile del tipo Real con il nome TIEFE. La variabile ha il valore iniziale 6.25.
DEF REAL TIEFE=3.1,LAENGE=2,ANZAHL	In una riga possono anche essere definite più variabili.
BOOL	
DEF BOOL WENN_ZUVIEL	Viene creata una variabile del tipo Bool con il nome WENN_ZUVIEL. Preimpostata dal sistema con zero (FALSE).
DEF BOOL WENN_ZUVIEL=1 oppure DEF BOOL WENN_ZUVIEL=TRUE oppure DEF BOOL WENN_ZUVIEL=FALSE	Viene creata una variabile del tipo Bool con il nome WENN_ZUVIEL.
CHAR	

1.2 Definizione delle variabili (DEF variabili definite dall'utente LUD, GUD, PUD)

DEF CHAR GUSTAV_1=65	Alle variabili del tipo Char può essere abbinato un valore numerico corrispondente al carattere ASCII oppure
DEF CHAR GUSTAV_1="A"	direttamente il relativo carattere ASCII (il valore numerico 65 corrisponde alla lettera "A").
STRING	
DEF STRING[6] MUSTER_1="ANFANG"	Le variabili del tipo String possono accettare una sequenza di caratteri. Il numero massimo di caratteri viene indicato tra parentesi quadre dopo il tipo di variabile.
AXIS	
DEF AXIS ACHSNAME=(X1)	Le variabili del tipo AXIS hanno il nome ACHSNAME e comprendono l'identificatore asse di un canale - in questo caso X1 (i nomi degli assi con indirizzo ampliato sono scritti tra parentesi tonde).
FRAME	
DEF FRAME SCHRAEG_1	Le variabili del tipo FRAME hanno il nome SCHRAEG_1.

Nota

Una variabile del tipo AXIS assume l'identificatore asse e l'identificatore mandrino di un canale.

Nota

I nomi degli assi con indirizzo ampliato devono essere scritti tra parentesi tonde.

Esempio di modifica della definizione di variabili applicative locali (LUD) e a livello di programma (PUD)

Se le stesse sono definite nel programma principale, hanno validità anche negli altri livelli dei sottoprogrammi richiamati. Vengono create con l'avvio del partprogram e cancellate con la fine del partprogram e/o con RESET.

Se il dato macchina \$MN_LUD_EXTENDED_SCOPE è settato, non è più possibile definire in un programma principale o in un sottoprogramma una variabile con lo stesso nome.

\$MN_LUD_EXTENDED_SCOPE=1	
PROC MAIN	;Programma principale
DEF INT VAR1	;Definizione PUD
...	;Richiamo del sottoprogramma
SUB2	
...	
M30	
PROC SUB2	;Sottoprogramma SUB2
DEF INT VAR2	;DEFINIZIONE LUD
...	
IF (VAR1==1)	;Lettura PUD

VAR1=VAR1+1	;Lettura e scrittura PUD
VAR2=1	;Scrittura LUD
ENDIF	;Richiamo del sottoprogramma
SUB3	
...	
M17	
PROC SUB3	;Sottoprogramma SUB3
...	
IF (VAR1==1)	;Lettura PUD
VAR1=VAR1+1	;Lettura e scrittura PUD
VAR2=1	;Errori: LUD in SUB2 non riconosciuta
ENDIF	
...	
M17	

Nomi delle variabili

Il nome di una variabile è composto da max. 31 caratteri. I primi due caratteri devono essere lettere o underscore.

Il carattere "\$" non può essere utilizzato per le variabili definite dall'utente in quanto questo carattere viene già utilizzato per le variabili di sistema.

Esempio di programma con variabili locali

DEF INT CONTATORE	
LOOP: GO X...	;Loop
CONTATORE=CONTATORE+1	
IF CONTATORE<50 GOTOB LOOP	
M30	

Esempio di interrogazione di assi geometrici esistenti

DEF AXIS ASCISSE IF ISAXIS(1) == FALSE GOTOF CONTINUA ASCISSE = \$P_AXN1 CONTINUA:	;1. Asse geometrico

Programmazione indiretta del mandrino

DEF AXIS SPINDLE SPINDLE= (S1) OVRA [SPINDLE] =80 SPINDLE= (S3) ...	;Override mandrino = 80%

1.3 Definizioni dei campi (DEF, SET, REP)

Funzione

Un campo è una cella di memoria che viene definita attraverso il tipo della variabile con nome e dimensione. Si possono definire campi con al max. 2 dimensioni.

Nota

Grandezza massima del campo

Nella definizione dei campi, la dimensione massima della 1^a e della 2^a dimensione è pari a 32767 per l'indice di campo [n, m].

Inizializzazione dei campi

Gli elementi di campo possono essere abbinati a valori di inizializzazione:

- durante l'esecuzione del programma

oppure

- anche durante la definizione del campo.

Nel caso di campi bidimensionali viene incrementato per primo l'indice di campo di destra.

Programmazione

```
DEF CHAR NAME [n,m]
oppure
DEF INT NAME [n,m]
oppure
DEF REAL NAME [n,m]
oppure
DEF AXIS NAME [n,m]
oppure
DEF FRAME NAME [n,m]
oppure
DEF STRING [lunghezza stringa] NAME [m]
oppure
DEF BOOL [n,m]
```

- **Inizializzazione con liste di valori, SET**

Possibilità nella definizione del campo

```
DEF Tipo VARIABILE = SET (VALORE)
DEF Tipo CAMPO [n,m] = SET (VALORE, Valore, ...)
oppure
DEF Tipo VARIABILE = valore
DEF Tipo CAMPO [n,m] = (valore, valore, ...)
```

Nota

Nella definizione del campo la programmazione SET è opzionale.

Possibilità nell'esecuzione del programma

```
CAMPO [n,m] = SET (valore, valore, valore, ...)
FELD [n,m] = SET (espressione, espressione, espressione, ...)
```

- **Inizializzazione con gli stessi valori, REP**

Possibilità nella definizione del campo

Al momento della creazione delle variabili "Campo" tutti gli elementi vengono inizializzati con

```
DEF Tipo CAMPOI [n,m] = REP (valore). Con
```

DEF Tipo CAMPO [n,m] = REP (valore, numero elementi di campo) solo gli elementi di campo definiti con "numero elementi di campo" vengono inizializzati con "valore".

L'inizializzazione comincia con l'indice di campo 0. Ai restanti elementi delle variabili viene assegnato il valore zero. Se "numero elementi di campo" = 0, all'intero campo delle variabili viene assegnato il valore zero, come ad es. a tutti i 30 elementi di campo di varName:

```
DEF REAL varName [30] = REP (3.5, 0)
```

Nota

Le variabili del tipo FRAME non possono essere inizializzate.

Se per il dato del campo viene programmato un valore minore di zero o superiore al numero degli elementi di campo, viene emesso un allarme "Campo di valori superato".

Possibilità nell'esecuzione del programma

CAMPO [n,m] = REP(valore)

CAMPO [n,m] = REP(espressione)

CAMPO [n,m] = REP(valore, numero elementi di campo)

CAMPO [n,m] = REP(espressione, numero elementi di campo)

Se il dato del campo è "numero elementi di campo" = 0, i valori attuali di tutte le variabili di campo rimangono invariati, ad es.

varName [12] =REP(44.5, 0)

Nota

Le variabili del tipo FRAME sono ammesse e possono in questo modo essere inizializzate con facilità.

Parametri

DEF Tipo variabile	Definizione del campo
SET VALORE oppure espressione	Inizializzazione con liste di valori nella definizione del campo oppure durante l'esecuzione del programma
REP VALORE oppure espressione	Inizializzazione con gli stessi valori nella definizione del campo oppure durante l'esecuzione del programma
REP VALORE, numero el. di campo	Inizializzazione con gli stessi valori per un determinato numero di elementi di campo da sovrascrivere. Il valore zero del secondo parametro, ad es. REP(10,0) determina la mancata descrizione dei valori di inizializzazione.
CHAR NAME [n,m] INT NAME [n,m] REAL NAME [n,m] AXIS NAME [n,m] FRAME NAME [n,m] BOOL [n,m]	Tipo di variabile (CHAR, INTEGER, REAL, AXIS, FRAME, BOOL)
STRING [lunghezza stringa] NAME [m]	Il tipo di dati STRING può essere definito solo con campi monodimensionali La lunghezza della stringa viene indicata dopo il tipo di dati STRING.
NOME	Nome della variabile
Tipo VARIABILE	Tipo di variabile (CHAR, INTEGER, REAL, AXIS, FRAME, BOOL)
TIPO CAMPO [n,m] = SET(valore, valore, ...)	Inizializzazione di tutti gli elementi di un campo con i valori elencati nella definizione del campo
TIPO CAMPO [n,m] = REP(valore)	Inizializzazione di tutti gli elementi di un campo con lo stesso valore della definizione del campo

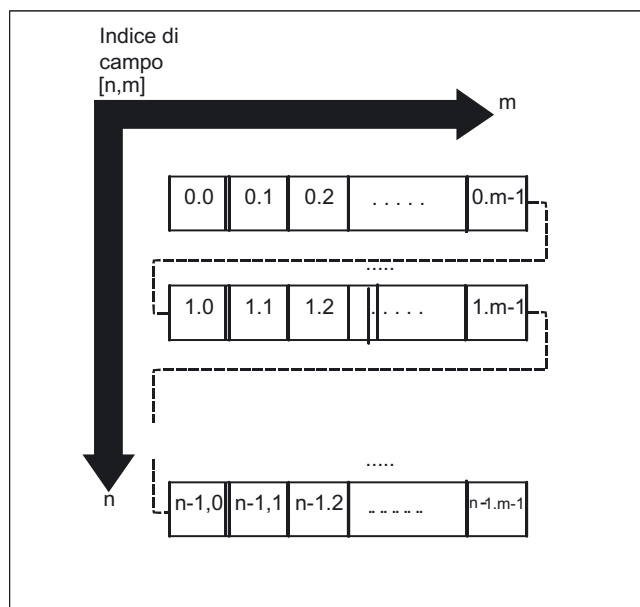
CAMPO[n,m] = SET(valore, valore, ...) CAMPO[n,m] = SET(espressione, ...)	Inizializzazione di tutti gli elementi di un campo con i valori elencati nell'esecuzione del programma
CAMPO[n,m] = REP(valore CAMPO[n,m] = REP(espressione)	Inizializzazione di tutti gli elementi di un campo con lo stesso valore nell'esecuzione del programma
CAMPO[n, m]	Indice di campo
n	Grandezza del campo per la 1° dimensione
m	Grandezza del campo per la 2° dimensione
Grandezza max.del campo	ad es. DEF INT NAME[32767]
Numero el. di campo	Numero degli elementi di campo da sovrascrivere con "valore" del comando REP. Tutti gli elementi di campo vengono sovrascritti con "valore".

I campi con variabili del tipo STRING possono essere solo monodimensionali.

Indice di campo [n,m]

Con l'indice di campo è possibile accedere direttamente agli elementi dello stesso. Tramite questi indici, gli elementi di campo possono essere preimpostati con dei valori oppure ne possono essere letti i valori.

Il primo elemento di campo inizia sempre con l'indice [0,0]; per una grandezza di campo di [3,4] l'indice di campo max. possibile è per esempio [2,3].



Memoria necessaria

Tipo di variabile	Occupazione di memoria per ogni elemento
BOOL	1 byte
CHAR	1 byte
INT	4 byte
REAL	8 byte
STRING	Lunghezza stringa + 1
FRAME	~ 400 byte in funzione del numero di assi
AXIS	4 byte

Nota

La dimensione massima di campo definisce anche le dimensioni dei blocchi di memoria in cui viene gestita la memoria variabile. Essa non dovrebbe essere impostata con una dimensione più grande del necessario.

Standard: 812 byte

Se non vengono definiti campi grandi, scegliere: 256 byte.

Esempio: definizione di campi BOOL

I dati utente globali devono contenere dei dati macchina PLC per l'attivazione/disattivazione del controllo numerico.

Esempio Definizione di campi con dimensione massima per la 1ª e la 2ª dimensione

```
DEF INT NAME [32767, 32767]
```

Esempio di inizializzazione di campi variabili di memoria

Per la rispettiva occupazione attuale vedere il disegno.

```
N10 DEF REAL FELD1[10,3] = SET(0, 0, 0, 10, 11, 12, 20, 20, 20, 30, 30, 30, 40, 40,
40,)
N20 FELD1[0,0] = REP(100)
N30 FELD1[5,0] = REP(-100)
N40 FELD1[0,0] = SET(0, 1, 2, -10, -11, -12, -20, -20, -20, -30, , , ,
-40, -40, -50, -60, -70)
N50 FELD1[8,1] = SET(8.1, 8.2, 9.0, 9.1, 9.2)
```

Indice di campo										2
1	[1,2]	N10: Inizializzazione in fase di definizione			N20/N30: Inizializzazione con lo stesso valore			N40/N50: Inizializzazione con valori diversi		
		0	1	2	0	1	2	0	1	2
	0	0	0	0	100	100	100	0	1	2
	1	10	11	12	100	100	100	-10	-11	-12
	2	20	20	20	100	100	100	-20	-20	-20
	3	30	30	30	100	100	100	-30	0	0
	4	40	40	40	100	100	100	0	-40	-40
	5	0	0	0	-100	-100	-100	-50	-60	-70
	6	0	0	0	-100	-100	-100	-100	-100	-100
	7	0	0	0	-100	-100	-100	-100	-100	-100
	8	0	0	0	-100	-100	-100	-100	8.1	8.2
	9	0	0	0	-100	-100	-100	9.0	9.1	9.2
		Gli elementi di campo [5,0] ... [9,2] sono stati inizializzati con il valore di default (0.0).						Gli elementi di campo [3,1] ... [4,0] sono stati inizializzati con il valore di default (0.0). Gli elementi di campo [6,0] ... [8,0] non sono stati modificati.		

Inizializzazione con liste di valori nella definizione del campo, SET

- Vengono assegnati tanti elementi di campo quanti valori di inizializzazione sono stati programmati.
- Gli elementi di campo senza valori (lacune nella lista dei valori) vengono preimpostati automaticamente a 0.
- Non sono consentiti buchi nella lista dei valori nel caso di variabili del tipo AXIS.
- Quando i valori programmati superano gli elementi di campo rimanenti, viene generato un allarme dal sistema.

Esempio:

```
DEF REAL FELD[2,3] = (10, 20, 30, 40)
```

Inizializzazione con liste di valori durante l'esecuzione del programma, SET

- L'inizializzazione avviene analogamente alla definizione.
- Come valori sono consentite anche espressioni.
- L'inizializzazione parte dall'indice di campo programmato. È possibile quindi assegnare dei valori anche a campi parziali.

Esempio: Assegnazione di espressioni

```
DEF INT CAMPO[5, 5]
CAMPO[0,0] = SET(1, 2, 3, 4, 5)
```

```
CAMPO[2,3] = SET(VARIABILE, 4*5.6)
```

Nel caso di variabili asse, l'indice dell'asse non viene elaborato:

Esempio: Inizializzazione di una riga

```
$MA_AX_VELO_LIMIT[1, AX1] = SET(1.1, 2.2, 3.3)
```

Corrisponde a:

```
$MA_AX_VELO_LIMIT[1,AX1] = 1.1
```

```
$MA_AX_VELO_LIMIT[2,AX1] = 2.2
```

```
$MA_AX_VELO_LIMIT[3,AX1] = 3.3
```

Inizializzazione con gli stessi valori della definizione del campo, REP

Tutti gli elementi del campo vengono preimpostati con lo stesso valore (costante).

Le variabili del tipo FRAME non possono essere inizializzate.

Esempi:

```
DEF REAL CAMPO5[10,3] = REP(9.9)
```

Inizializzare i campi di variabili da varName[0] a varName[3] con 3,5:

```
DEF REAL varName[30] = REP(3.5, 4)
```

Impostare la variabile in base al campo su un valore programmato:

```
varName[5] = REP(4.5, 10) ; impostare da varName[5] a varName[14] sul valore 4
```

Impostare i parametri R in base al campo su determinati valori:

```
R10 = REP(2.4, 15) ; impostare da R10 a R24 sul valore 2,4
```

Inizializzazione con gli stessi valori dell'esecuzione del programma

- Come valori sono consentite anche espressioni.
- Tutti gli elementi del campo vengono inizializzati con lo stesso valore.
- L'inizializzazione parte dall'indice di campo programmato. È possibile quindi assegnare dei valori anche a campi parziali.

Esempio: Inizializzazione di tutti gli elementi con un solo valore

```
DEF FRAME FRM[10]
```

```
FRM[5] = REP(CTTRANS (X,5))
```

1.4 Programmazione indiretta

Funzione

La programmazione indiretta consente di utilizzare i programmi in forma universale. L'indirizzo ampliato (indice) viene sostituito da una variabile di tipo idoneo.

Programmazione indiretta dei codici G

La programmazione indiretta di codici G attraverso variabili consente una effettiva programmazione dei cicli. Sono disponibili due parametri.

Gruppi codici G con costanti Integer

Numeri di codici G con variabile del tipo Integer/Real

.

Programmazione

INDIRIZZO [INDICE]

oppure

G[<indice gruppi>] = <variabile Integer/Real>

Programmazione indiretta di codici G attraverso variabili per una effettiva programmazione dei cicli

Parametri

Tutti gli indirizzi sono parametrizzabili ad eccezione di:

- N - numero di blocco
- L - sottoprogramma

Non è possibile una programmazione indiretta per tutti gli indirizzi impostabili.

Esempio: Non è consentito X1 invece di X[1].

INDIRIZZO	Indirizzo con impostazione del parametro come indice
[INDICE]	Indice Variabile p. es. n. mandrino, asse
G>indice gruppi	Gruppi codici G: Costante Integer con la quale viene selezionato il gruppo di codici G
<Variabile Integer/Real>	Numero codici G: Variabile del tipo Integere oppure Real con la quale viene selezionato il numero di codici G

Gruppi validi di codici G

Si possono programmare in modo indiretto solo dei gruppi di codici G con azione modale.

Con l'allarme 12470 viene rifiutata la programmazione dei gruppi di codici G con validità blocco-blocco.

Numeri validi di codice G

Nella programmazione indiretta di codici G non sono consentite le funzioni aritmetiche.

Il numero di codice G va memorizzato in una variabile del tipo Integer o Real. Con l'allarme 12475 vengono rifiutati eventuali numeri di codici G non validi.

Se vi è la necessità di calcolare i numeri di codici G, questo dovrà avvenire in una riga specifica del partprogram prima della programmazione indiretta dei codici G.

Nota

Tutti i codici G validi si trovano elencati in gruppi diversi nel Manuale di programmazione, al Capitolo "Elenco delle funzioni G/funzioni preparatorie". Vedere /PG/ Manuale di programmazione, Concetti fondamentali, "Tabelle"

Esempio

Mandrino	
S1=300	;Programmazione diretta
DEF INT SPINU=1	;Programmazione indiretta:
S [SPINU] =300	;300 giri/min per il mandrino, ;il cui numero è inserito nella variabile SPINU ;(in questo esempio 1).
Avanzamento	
FA [U] =300	;Programmazione diretta
DEF AXIS AXVAR2=U	;Programmazione indiretta:
FA [AXVAR2] =300	;Avanzamento per l'asse di posizionamento il cui ;indirizzo è inserito nella variabile di tipo AXIS con ;il nome AXVAR2.
Valore di misura	
\$AA_MM [X]	;Programmazione diretta
DEF AXIS AXVAR3=X	;Programmazione indiretta:
\$AA_MM [AXVAR3]	;Valore di misura nel sistema di coordinate macchina ;per l'asse ;il cui nome è inserito nella ;variabile AXVAR3.
Elemento di campo	
DEF INT FELD1 [4,5]	;Programmazione diretta
DEFINE DIM1 AS 4	;Programmazione indiretta:
DEFINE DIM2 AS 5	;Per le dimensioni del campo
DEF INT FELD [DIM1,DIM2]	;è necessario impostare dei valori fissi.
FELD [DIM1-1,DIM2-1] =5	
Istruzioni assi con variabili assi	
X1=100 X2=200	;Programmazione diretta
DEF AXIS AXVAR1 AXVAR2	;Programmazione indiretta:
AXVAR1= (X1) AXVAR2= (X2)	;Definizione delle variabili
AX [AXVAR1] =100 AX [AXVAR2] =200	;Assegnazione dei nomi degli assi. ;Movimento degli assi memorizzati nelle variabili ;a 100 e 200.
Parametri di interpolazione con variabili assi	
G2 X100 I20	;Programmazione diretta

DEF AXIS AXVAR1=X	;Programmazione indiretta:
G2 X100 IP[AXVAR1]=20	;Definizione ed assegnazione dei nomi degli assi.
	;Programmazione indiretta del centro del
	;cerchio.
Richiamo indiretto di sottoprogrammi	
CALL "L" << R10	;Richiamo del programma il cui numero si trova in R10

Nota

Anche i parametri R possono essere utilizzati come campi monodimensionali con scrittura abbreviata (R10 corrisponde a R[10]).

Esempio di programmazione indiretta dei codici G

Spostamento origine impostabile, gruppo 8 di codici G

N1010 DEF INT INT_VAR	
N1020 INT_VAR = 2	
...	
N1090 G[8] = INT_VAR G1 X0 Y0	;G54
N1100 INT_VAR = INT_VAR + 1	;calcolo codice G
N1110 G[8] = INT_VAR G1 X0 Y0	;G55

Selezione del piano, gruppo 6 di codici G

N2010 R10 = \$P_GG[6]	;lettura codici G per il piano attuale
...	
N2090 G[6] = R10	;G17

1.4.1 Elaborazione di una stringa come riga del partprogram (EXECSTRING)

Funzione

Con l'istruzione di programma EXECSTRING, come parametro viene passata stringa che contiene la riga di partprogram da eseguire.

Programmazione

EXECSTRING (<variabile String>)

Parametri

EXECSTRING	Trasmissione di una variabile String con la riga di partprogram da eseguire
(<variabile String>)	Parametro con la riga effettiva di partprogram da eseguire

Nota

Si possono trasferire tutti gli elementi di un programma che possono essere programmati all'interno di un partprogram. Sono escluse le istruzioni PROC e DEF e in generale l'utilizzo in file INI e DEF.

Esempio di riga di partprogram indiretta

N100 DEF STRING[100] BLOCK	;Variabile String contenente la riga del partprogram
N110 DEF STRING[10] MFCT1 = "M7"	
N200 EXECSTRING(MFCT1 << " M4711")	;Esecuzione della riga del partprogram "M7 M4711"
N300 R10 = 1	
N310 BLOCK = "M3"	
N320 IF (R10)	
N330 BLOCK = BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING (BLOCK)	;Esecuzione della riga del partprogram "M3 M4711"

1.5 Assegnazioni

Funzione

Alle variabili/parametri di calcolo è possibile assegnare nel programma valori di un tipo adeguato.

Programmazione

L'assegnazione richiede sempre un blocco a sé stante; per ogni blocco sono possibili più assegnazioni. Le assegnazioni per indirizzi di asse (istruzioni di movimento) richiedono un blocco separato tramite assegnazioni di variabili.

Parametri

Assegnazione a variabili stringa

Nell'ambito di una sequenza CHAR oppure STRING occorre distinguere tra scrittura maiuscola e minuscola.

Se i caratteri ' oppure " devono fare parte della sequenza di caratteri, essi vanno inseriti tra i caratteri '... '.

Esempio:

```
MSG("Viene lavorata l'"ultima figura")
```

visualizza su video il testo "Viene lavorata l'ultima figura".

I caratteri non visualizzabili possono essere inseriti nella stringa come costante binaria o esadecimale.

Esempio

```
R1=10.518 R2=4 VARI1=45
```

;Assegnazione di un valore numerico

```
X=47.11 Y=R2
```

;Assegnazione di una variabile del tipo adeguato

```
R1=R3 VARI1=R4
```

;Assegnazione con segno opposto (consentito

```
R4=-R5 R7=-VARI8
```

;solo per tipi INT e REAL)

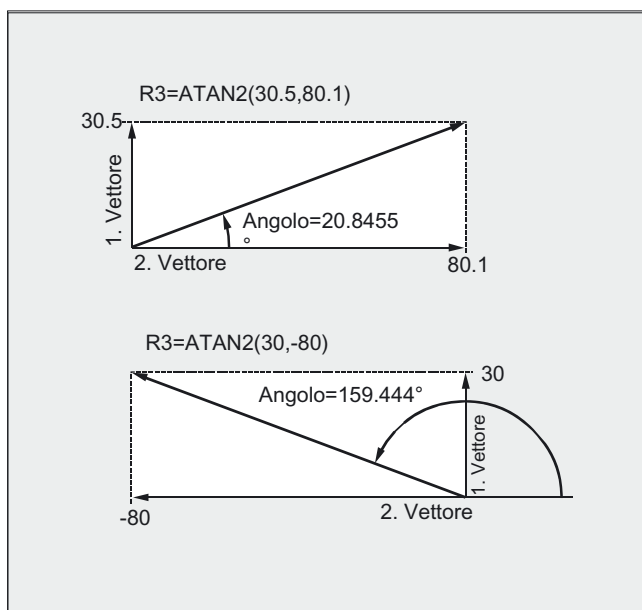
1.6 Operazioni/funzioni di calcolo

Funzione

Le funzioni di calcolo possono essere utilizzate in particolare per parametri R (oppure costanti e funzioni) del tipo REAL. Sono consentiti anche i tipi INT e CHAR.

Funzione di calcolo ATAN2(,)

La funzione calcola l'angolo del vettore somma di due vettori ortogonali. Il risultato è compreso nel campo dei quattro quadranti ($-180^\circ < \theta < +180^\circ$). La base per l'angolo di riferimento è sempre il 2° valore in direzione positiva.



Precisioni con istruzioni di confronto impostabili con TRUNC()

Vedere "Correttore di precisione con istruzioni di confronto"

Minimo, massimo e campo delle variabili

Vedere "Minimo, massimo e campo delle variabili (MINVAL, MAXVAL e BOUND)"

Programmazione

Per gli operatori di calcolo vale la scrittura matematica tradizionale. Le priorità dell'elaborazione vengono impostate tramite parentesi rotonde. Per le funzioni trigonometriche e le relative funzioni inverse vale l'impostazione in gradi (angolo retto=90°).

Parametri

Operatori/funzioni di calcolo

+	addizione
-	sottrazione
*	moltiplicazione
/	divisione
	Attenzione: (tipo INT)/(tipo INT)=(tipo REAL); Esempio: 3/4 = 0.75
DIV	divisione, solo per variabili di tipo INT e REAL
	Attenzione: (tipo INT)DIV(tipo INT)=(tipo INT); Esempio: 3 DIV 4 = 0
MOD	il modulo della divisione (solo per tipo INT) fornisce il resto di una divisione INT; Esempio 3 MOD 4=3
:	operatore di concatenamento (con variabili FRAME)
Sin()	Seno
COS()	coseno
TAN()	tangente
ASIN()	arcoseno
ACOS()	arcocoseno
ATAN2(,)	arcotangente2
SQRT()	radice quadrata
ABS()	valore assoluto
POT()	2. 2° potenza (quadrato)
TRUNC()	parte intera
ROUND()	arrotondamento a intero
LN()	logaritmo naturale
EXP()	funzione esponenziale
MINVAL()	valore più piccolo di due variabili
MAXVAL()	valore più grande di due variabili
BOUND()	valore della variabile rientrante nel campo di valori definito
CTRANS()	Traslazione
CROT()	rotazione
CSCALE()	variazione del fattore di scala
CMIRROR()	Specularità

Esempio di inizializzazione di campi variabili di memoria

R1=R1+1	;nuovo R1 = vecchio R1 +1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	;La moltiplicazione viene eseguita prima dell'addizione
R14= (R1+R2) *R3	;Vengono eseguite per prime le parentesi
R15=SQRT(POT(R1)+POT(R2))	;Vengono eseguite per prime le parentesi interne
	;R15 = radice quadrata calcolata da (R1+R2)
RESFRAME= FRAME1:FRAME2	;Con l'operatore di concatenamento vengono
FRAME3=CTTRANS (...) :CROT(...)	;collegati Frame ad un Frame risultante oppure
	;ai valori dei componenti Frame

1.7 Operazioni logiche e di confronto

Funzione

Le **operazioni di confronto** possono essere utilizzate ad esempio per formulare delle condizioni di salto. In questo modo si possono confrontare anche espressioni complesse.

Le operazioni di confronto possono essere utilizzate per le variabili del tipo `CHAR`, `INT`, `REAL` e `BOOL`. Per il tipo `CHAR` viene confrontato il valore del codice.

Per i tipi `STRING`, `AXIS` e `FRAME` sono ammessi: `==` e `<>`, che possono essere applicati per le operazioni del tipo `STRING` anche nelle azioni sincrone.

Il risultato delle operazioni confrontate è sempre del tipo `BOOL`.

Operatori logici servono per operazioni con valori di verità.

Le operazioni logiche sono utilizzabili solo con variabili del tipo `BOOL`. Tramite conversione interna del tipo si possono applicare anche ai tipi di dati `CHAR`, `INT` e `REAL`.

Nel caso delle operazioni logiche (booleane) per i tipi di dati `BOOL`, `CHAR`, `INT` e `REAL` vale quanto segue:

- 0 corrisponde a: `FALSE`
- diverso da 0 corrisponde a: `TRUE`

Operatori logici a bit

Con le variabili del tipo `CHAR` e `INT` si possono eseguire anche operazioni logiche a bit. Eventualmente avviene automaticamente una conversione di tipo.

Programmazione

Operatori di confronto

`==`

oppure

`<>`

oppure

`>`

oppure

`<`

oppure

`>=`

oppure

`<=`

Operatori logici

`AND`

oppure

`OR`

oppure

NOT

oppure

XOR

Tra gli operandi BOOLEANI e gli operatori si devono lasciare degli spazi intermedi.

Operatori logici a bit

B_AND

oppure

B_OR

oppure

B_NOT

oppure

B_XOR

Parametri

Significato degli operatori di confronto

==	uguale
<>	diverso
>	maggiore
<	minore
>=	maggiore o uguale
<=	minore o uguale

Significato degli operatori logici

AND	AND
OR	OR
NOT	Negazione
XOR	OR esclusivo

Nelle espressioni aritmetiche la sequenza di elaborazione di tutti gli operatori può essere determinata con parentesi tonde, scostandosi così dalle normali regole di priorità.

Significato degli operatori logici a bit

B_AND	AND a bit
B_OR	OR a bit
B_NOT	negazione a bit
B_XOR	OR esclusivo a bit

Nota

L'operatore B_NOT si riferisce ad un solo operando che si trova immediatamente dopo l'operatore stesso.

Esempio di operatori di confronto

```
IF R10>=100 GOTO DESTINAZIONE
```

oppure

```
R11=R10>=100
```

```
IF R11 GOTO DESTINAZIONE
```

Il risultato del confronto $R10 \geq 100$ viene memorizzato temporaneamente in R11.

Esempio di operatori logici

```
IF (R10<50) AND ($AA_IM[X] >=17.5) GOTO DESTINAZIONE
```

oppure

```
IF NOT R10 GOTO START
```

NOT si riferisce solo ad un operando.

Esempio di operatori logici a bit

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTO ACT_PLANE
```

1.7.1 Correttore di precisione in caso di errori di confronto (TRUNC)

Funzione

L'istruzione TRUNC tronca le cifre decimali dell'operando moltiplicato con un fattore di precisione.

Precisione impostabile in caso di istruzioni di confronto

I dati del partprogram del tipo REAL vengono rappresentati internamente nel formato IEEE a 64 bit. Questo formato di rappresentazione può causare delle inesattezze nei numeri decimali e quindi, in un confronto con valori ideali, si possono verificare risultati errati.

Uguaglianza relativa

Per evitare che le inesattezze dovute al formato di rappresentazione incidano sullo svolgimento del programma, nelle istruzioni di confronto non si verifica l'uguaglianza assoluta ma soltanto quella relativa.

Programmazione

Correttore di precisione in caso di errori di confronto

TRUNC (R1*1000)

Parametri

TRUNC ()	Esclusione delle cifre decimali

Uguaglianza relativa considerata pari a 10^{-12} con

- uguale: (==)
- diverso: (< >)
- maggiore o uguale: (>=)
- minore o uguale: (<=)
- maggiore/minore: (> <) con uguaglianza assoluta
- maggiore: (>)
- minore: (<)

Compatibilità

Per motivi di compatibilità la verifica di uguaglianza relativa con (>) e (<) può essere disabilitata settando il DM 10280: PROG_FUNCTION_MASK Bit0 = 1.

Nota

Confronti con dati del tipo REAL, per i motivi citati in precedenza, sono affetti generalmente da una certa imprecisione. In caso di differenze non accettabili, si deve ricorrere al calcolo INTEGER moltiplicando gli operandi con un fattore di precisione e successivamente troncadoli con l'istruzione TRUNC.

Azioni sincrone

Il comportamento descritto per quanto riguarda le istruzioni di confronto vale anche per le azioni sincrone.

Esempio di valutazione della precisione

N40 R1=61.01 R2=61.02 R3=0.01	;assegnazione dei valori iniziali
N41 IF ABS(R2-R1) > R3 GOTOF ERRORE	;il salto finora veniva eseguito
N42 M30	;Fine programma
N43 ERRORE: SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	;assegnazione dei valori iniziali
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)	;correttore di precisione
R13=TRUNC(R3*1000)	
IF ABS(R12-R11) > R13 GOTOF ERRORE	;il salto non viene più eseguito
M30	;Fine programma
ERRORE: SETAL(66000)	

Esempio di definizione e valorizzazione del quoziente di entrambi gli operandi

R1=61.01 R2=61.02 R3=0.01	;assegnazione dei valori iniziali
IF ABS((R2-R1)/R3)-1 > 10EX-5 GOTOF	;il salto non viene eseguito
ERRORE	
M30	;Fine programma
ERRORE: SETAL(66000)	

1.7.2 Minimo, massimo e campo delle variabili (MINVAL, MAXVAL e BOUND)

Funzione

Con le due funzioni MINVAL e MAXVAL è possibile operare un confronto tra due variabili e calcolare il valore più piccolo o più grande (risultato).

La funzione BOUND consente di verificare se il valore di una variabile verificata rientra o meno nel campo di valori definito. In caso affermativo, il valore della variabile viene restituito. Se il valore della variabile da verificare supera il valore massimo, viene fornito il valore massimo. In modo analogo, se il valore della variabile è inferiore al valore minimo, viene fornito il valore minimo.

Comportamento in caso di uguaglianza

In caso di uguaglianza, per le funzioni MINVAL e MAXVAL viene fornito il valore uguale e per la funzione BOUND il valore della variabile da verificare.

Programmazione

Minimo

Risultato: valore più piccolo = <MINVAL>(<variabile1>, <variabile2>)

Massimo

Risultato: valore più grande = <MAXVAL>(<variabile1>, <variabile2>)

Campo

Valore fornito = <BOUND>(<variabile Min>, <variabile Max>, <variabile Verifica>)

Nota

Le tre funzioni MINVAL(), MAXVAL() e BOUND() possono essere programmate anche con azioni sincrone.

Parametri

MINVAL ()	determina il valore più piccolo di due variabili
MAXVAL ()	determina il valore più grande di due variabili
BOUND ()	stabilisce se la variabile da verificare rientra o meno in un campo di valori Min / Max definito
Variabile1, Variabile2	prima variabile e seconda variabile il cui valore viene verificato per Minimo / Massimo
Variabile Min	Campo di definizione inferiore per il valore della variabile da verificare
Variabile Max	Campo di definizione superiore per il valore della variabile da verificare
Variabile Verifica	Variabile per cui occorre verificare se il relativo valore rientra nel campo definito

Esempi

DEF REAL rVar1 = 10.5, rVar2 = 33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar	
rValMin = MINVAL(rVar1, rVar2)	;rValMin viene impostata sul valore 10.5
rValMax = MAXVAL(rVar1, rVar2)	;rValMax viene impostata sul valore 33.7
rVar3 = 19.7	
rRetVar = BOUND(rVar1, rVar2, rVar3)	;rVar3 rientra nei limiti, ;rRetVar viene impostata su 19.7
rVar3 = 1.8	
rRetVar = BOUND(rVar1, rVar2, rVar3)	;rVar3 è al di sotto del limite minimo, ;rRetVar viene impostata su 10.5
rVar3 = 45.2	
rRetVar = BOUND(rVar1, rVar2, rVar3)	;rVar3 è al di sopra del limite massimo, ;rRetVar viene impostata su 33.7

1.8 Priorità delle operazioni

Funzione

Ad ogni operatore è assegnata una priorità. Nella valutazione di un'espressione vengono considerati sempre prima gli operatori con priorità più elevata. Per operatori dello stesso livello, la valutazione viene effettuata da sinistra verso destra.

Nelle espressioni aritmetiche la sequenza di elaborazione di tutti gli operatori può essere determinata con parentesi tonde, scostandosi così dalle normali regole di priorità.

Sequenza degli operatori

Dalla priorità più alta alla più bassa

1.	NOT, B_NOT	negazione, negazione a bit
2.	*, /, DIV, MOD	moltiplicazione, divisione
3.	+, -	addizione, sottrazione
4.	B_AND	AND a bit
5.	B_XOR	OR esclusivo a bit
6.	B_OR	OR a bit
7.	AND	AND
8.	XOR	OR esclusivo
9.	OR	OR
10.	<<	concatenamento di stringhe, tipo risultante STRING
11.	==, <>, >, <, >=, <=	Operatori di confronto

Nota

L'operatore di concatenamento ":" per Frame non può sussistere nella stessa espressione insieme ad altri operatori. Pertanto non è necessario assegnare ad esso un livello di priorità.

Esempio di istruzione IF

```
If (otto==10) and (anna==20) gotof end
```

1.9 Possibili conversioni di tipi

Funzione

Conversione del tipo nell'assegnazione

Il valore numerico costante, la variabile oppure l'espressione alla quale viene assegnata una variabile, devono essere compatibili con il tipo di variabile. Se questo presupposto è soddisfatto, il tipo viene convertito automaticamente in fase di assegnazione.

Possibili conversioni di tipi

a	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
da							
REAL	sì	sì*	sì ¹⁾	sì*	–	–	–
INT	sì	sì	sì ¹⁾	sì ²⁾	–	–	–
BOOL	sì	sì	sì	sì	sì	–	–
CHAR	sì	sì	sì ¹⁾	sì	sì	–	–
STRING	–	–	sì ⁴⁾	sì ³⁾	sì	–	–
AXIS	–	–	–	–	–	sì	–
FRAME	–	–	–	–	–	–	sì

Spiegazioni

- * Per la conversione da REAL a INT, un valore frazionario ≥ 0.5 viene arrotondato alla cifra successiva, gli altri valori vengono arrotondati alla cifra precedente (vedere funzione ROUND).
- 1) Un valore ≤ 0 corrisponde a TRUE, un valore $\neq 0$ corrisponde a FALSE
- 2) Se il valore si trova in un campo numerico consentito
- 3) Se ha solo 1 carattere
- 4) Lunghezza stringa 0 $\Rightarrow$ FALSE, altrimenti TRUE

Nota

Se nella conversione il valore è più grande del campo di destinazione, viene emessa una segnalazione di errore.

Se in una espressione sono presenti dei tipi misti, viene eseguito automaticamente un adeguamento del tipo. Le conversioni possono avvenire anche nelle azioni sincrone, vedere in proposito il capitolo "Azioni sincrone di movimento, conversione implicita del tipo".

1.10 Operazioni su stringhe

Panoramica

Oltre alle operazioni classiche descritte in questo capitolo di "assegnazione" e "confronto", sono disponibili altre possibilità per la manipolazione di stringhe:

Parametri

Conversioni dei tipi in STRING:

STRING_ERG = <<tipo qualsiasi¹⁾

Tipo risultante: STRING

STRING_ERG = AXSTRING (AXIS)

Tipo risultante: STRING

Conversioni dei tipi da STRING:

BOOL_ERG = ISNUMBER (STRING)

Tipo risultante: BOOL

REAL_ERG = NUMBER (STRING)

Tipo risultante: REAL

AXIS_ERG = AXNAME (STRING)

Tipo risultante: AXIS

Concatenamento di stringhe:

tipo qualsiasi¹⁾ << tipo qualsiasi¹⁾

Tipo risultante: STRING

Conversione in minuscolo/maiuscolo:

STRING_ERG = TOUPPER (STRING)

Tipo risultante: STRING

STRING_ERG = TOLOWER (STRING)

Tipo risultante: STRING

Lunghezza della stringa:

INT_ERG = STRLEN (STRING)

Tipo risultante: INT

Ricerca di un carattere/stringa nella stringa:

INT_ERG = INDEX (STRING, CHAR)

Tipo risultante: INT

INT_ERG = RINDEX (STRING, CHAR)

Tipo risultante: INT

INT_ERG = MINDEX (STRING, STRING)

Tipo risultante: INT

INT_ERG = MATCH (STRING, STRING)

Tipo risultante: INT

Selezione di una parte di stringa:

STRING_ERG = SUBSTR (STRING, INT)

Tipo risultante: INT

STRING_ERG = SUBSTR (STRING, INT, INT)

Tipo risultante: INT

Selezione di un singolo carattere:

CHAR_ERG = STRINGVAR [IDX]

Tipo risultante: CHAR

CHAR_ERG = STRINGFELD [IDX_FELD, IDX_CHAR]

Tipo risultante: CHAR

¹⁾ "Tipo qualsiasi" è valido per i tipi di variabili INT, REAL, CHAR, STRING e BOOL.

Significato particolare del carattere 0

Il carattere 0 viene interpretato internamente come indicatore di fine stringa. Se un carattere viene sostituito da uno 0, la stringa viene accorciata.

Esempio

```
DEF STRING[20] STRG = "Asse . fermo"
STRG[6] = "X" ;emette il messaggio "Asse X fermo"
MSG(STRG)
STRG[6] = 0
MSG(STRG) ;emette il messaggio "Asse"
```

1.10.1 Conversione di tipi in STRING

Funzione

È possibile utilizzare variabili di diversi tipi come parte integrante di un messaggio (MSG).

Avviene utilizzando l'operatore << per i tipi di dati INT, REAL, CHAR e BOOL (vedere "Concatenamento di stringhe").

I valori INT vengono convertiti nella forma normalmente leggibile. Con valori REAL vengono indicate fino a 10 cifre dopo la virgola.

Programmazione

Sintassi

STRING_ERG = AXSTRING (AXIS)	Tipo risultante: STRING

Semantica:

AXSTRING (AXIS) fornisce l'identificatore asse indicato sotto forma di stringa.

Parametri

Le variabili del tipo AXIS possono essere trasformate in variabili STRING mediante la funzione AXSTRING.

Le variabili FRAME invece non possono essere convertite.

Esempio:

```
MSG("Posizione:" << $AA_IM[X])
```

Esempio

```
DEF STRING[32] STRING_ERG
STRING_ERG = AXSTRING(X) ;adesso: STRING_ERG == "X"
```

1.10.2 Conversione di tipi da STRING

Funzione

Con la funzione NUMBER si esegue la conversione da STRING a REAL.

Se ISNUMBER fornisce il valore FALSE, al RICHIAMO di NUMBER con lo stesso parametro viene emesso un allarme.

Con la funzione AXNAME, è possibile convertire una stringa in un tipo di dati AXIS. Se la stringa non può essere abbinata a nessun indicatore di asse progettato viene emesso un allarme.

Programmazione

Sintassi

REAL_ERG = NUMBER (STRING)	Tipo risultante: REAL
BOOL_ERG = ISNUMBER (STRING)	Tipo risultante: BOOL
AXIS_ERG = AXNAME (STRING)	Tipo risultante: AXIS

Semantica:

NUMBER (STRING) restituisce il valore rappresentato dalla stringa come valore REAL.

ISNUMBER fornisce il valore TRUE se la stringa rappresenta, secondo le regole del linguaggio, un numero REAL valido. In tal modo è possibile verificare se la stringa può essere convertita in un numero valido.

AXNAME (STRING) trasforma la stringa indicata in un identificatore di asse.

Esempio

```
DEF BOOL BOOL_ERG
DEF REAL REAL_ERG
DEF AXIS AXIS_ERG
BOOL_ERG = ISNUMBER ("1234.9876Ex-7") ;adesso: BOOL_ERG == TRUE
BOOL_ERG = ISNUMBER ("1234XYZ") ;adesso: BOOL_ERG == FALSE
REAL_ERG = NUMBER ("1234.9876Ex-7") ;adesso: REAL_ERG == 1234.9876Ex-7
AXIS_ERG = AXNAME ("X") ;adesso: AXIS_ERG == X
```

1.10.3 Concatenamento di stringhe

Funzione

Questa funzione offre la possibilità di raggruppare diverse stringhe in una stringa unica.

Il concatenamento viene realizzato tramite l'operatore: <<. Il tipo restituito da questo operatore è sempre STRING per tutte le combinazioni dei tipi base CHAR, BOOL, INT, REAL e STRING. Nel caso si renda necessaria una conversione, questa viene eseguita secondo le regole esistenti.

Programmazione

Sintassi

tipo_qualsiasi << tipo_qualsiasi	Tipo risultante: STRING

Descrizione

Le stringhe indicate (o eventualmente l'altro tipo implicitamente convertito) vengono concatenate tra di loro.

Questo operatore è disponibile anche come cosiddetta variante "Unitaria". In questo modo è possibile eseguire una conversione di tipo esplicita in STRING (non per FRAME e AXIS).

I tipi FRAME e AXIS non possono essere utilizzati con questo operatore.

Sintassi

<< tipo_qualsiasi	Tipo risultante: STRING

Descrizione

Il tipo indicato viene convertito implicitamente nel tipo STRING.

In tal modo è possibile ad esempio unire un messaggio oppure un comando da una lista di testi ed introdurre parametri (ad es. un nome di un blocco):

MSG(STRG_TAB[LOAD_IDX]<<BLOCCO_NOME)

Cautela

I risultati intermedi nel concatenamento di stringhe non possono superare la lunghezza massima della stringa.

Esempio di concatenamento di stringhe

```

DEF INT IDX = 2
DEF REAL VALUE = 9.654
DEF STRING[20] STRG = "INDEX:2"
IF STRG == "Index:" <<IDX GOTO NO_MSG
MSG ("Index:" <<IDX <<"/Valore:" <<VALUE) ;Visualizzazione: "Index: 2/Valore: 9.654"
<<VALUE)
NO_MSG:

```

1.10.4 Conversione in minuscolo/maiuscolo

Funzione

Questa funzione consente di trasformare tutte le lettere di una stringa in una rappresentazione univoca.

Sintassi

STRING_ERG = TOUPPER (STRING)	Tipo risultante: STRING
STRING_ERG = TOLOWER (STRING)	Tipo risultante: STRING

Descrizione

Tutte le lettere vengono trasformate in minuscolo o maiuscolo.

Esempio

Poiché sull'HMI è anche possibile impostare dati utente, si può realizzare una rappresentazione univoca con caratteri minuscoli o maiuscoli:

```

DEF STRING [29] STRG
...
IF "LEARN.CNC" == TOUPPER (STRG) GOTO LOAD_LEARN

```


1.10.5 Lunghezza della stringa

Funzione

Questa funzione consente di determinare la lunghezza di una stringa.

Sintassi

INT_ERG = STRLEN (STRING)	Tipo risultante: INT

Descrizione

Viene restituito il numero dei caratteri diversi da 0 conteggiati dall'inizio della stringa.

Esempio

In questo modo, utilizzando ad esempio l'accesso al singolo carattere di seguito descritto, è possibile definire la fine della stringa:

```
IF (STRLEN (NOME_BLOCCO) > 10) GOTOF ERRORE
```

1.10.6 Ricerca di un carattere/stringa nella stringa

Funzione

Questa funzione consente di ricercare un carattere o una stringa all'interno di un'altra stringa. La funzione indica in quale posizione della stringa sono stati trovati il carattere o la stringa ricercati.

Programmazione

Sintassi

INT_ERG = INDEX	(STRING,CHAR)	Tipo risultante: INT
INT_ERG = RINDEX	(STRING,CHAR)	Tipo risultante: INT
INT_ERG = MINDEX	(STRING,STRING)	Tipo risultante: INT
INT_ERG = MATCH	(STRING,STRING)	Tipo risultante: INT

Descrizione

Funzioni di ricerca: restituiscono la posizione nell'ambito della stringa (primo parametro), se la ricerca è andata a buon fine. Se il carattere o la stringa non vengono trovati, viene restituito il valore -1. Il primo carattere ha la posizione 0.

Parametri

INDEX	ricerca nel primo parametro il carattere definito come secondo parametro (dall'inizio).
RINDEX	ricerca nel primo parametro il carattere definito come secondo parametro (dal fondo).
MINDEX	corrisponde alla funzione INDEX, con la differenza che viene fornita una lista di caratteri (come stringa) e viene restituito un indice alla posizione del primo carattere trovato.
MATCH	cerca una stringa in un'altra stringa.

Le stringhe possono essere scomposte rispettando determinati criteri ed inserendo nella posizione desiderata uno spazio o il carattere separatore ("/").

Esempio di scomposizione di una registrazione in percorso e nome del blocco

DEF INT PFADIX, PROGIX	
DEF STRING[26] EINGABE	
DEF INT LISTIX	
EINGABE = "/_N_MPF_DIR/_N_EXECUTE_MPF"	
LISTIX = MINDEX (EINGABE, "M,N,O,P") + 1	Come valore in LISTIX viene restituito 3, poiché "N" risulta essere il primo carattere nel parametro EINGABE della lista di selezione dall'inizio.
PFADIX = INDEX (EINGABE, "/") +1	;Vale quindi: PFADIX = 1
PROGIX = RINDEX (EINGABE, "/") +1	;Vale quindi: PROGIX = 12
	;utilizzando la funzione SUBSTR inserita nella
	;sezione successiva, la variabile
	;EINGABE viene scomposta nei componenti
	"percorso" e "blocco":
VARIABLE = SUBSTR (EINGABE, PFADIX,	;restituisce quindi "_N_MPF_DIR"
PROGIX-PFADIX-1)	
VARIABLE = SUBSTR (EINGABE, PROGIX)	;restituisce quindi "_N_EXECUTE_MPF"

1.10.7 Selezione di una parte di stringa

Funzione

Questa funzione consente di estrarre una parte di stringa dalla stringa stessa. A tale scopo viene indicato l'indice del primo carattere ed eventualmente la lunghezza della stringa desiderata. Se non viene impostata la lunghezza, viene estratta la parte restante della stringa.

Programmazione

Sintassi

STRING_ERG = SUBSTR (STRING,INT)	Tipo risultante: INT
STRING_ERG = SUBSTR (STRING,INT, INT)	Tipo risultante: INT

Descrizione

Nel primo caso viene restituita la parte della stringa dalla posizione definita dal primo parametro fino alla fine della stringa.

Nel secondo caso la stringa risultante viene limitata nella lunghezza massima, dal valore definito nel terzo parametro.

Se la posizione iniziale si trova oltre la fine della stringa, viene restituita la stringa vuota (" ").

Se la posizione iniziale oppure la lunghezza sono negative, viene emesso un allarme.

Esempio

```
DEF STRING [29] ERG
ERG = SUBSTR ("CONFERMA: da 10 a 99, 10, 2) ;Vale quindi: ERG == "10"
```

1.10.8 Selezione di un singolo carattere

Funzione

Questa funzionalità consente di selezionare un singolo carattere di una stringa. Questo vale tanto per un accesso in lettura, quanto per uno in scrittura.

Programmazione

Sintassi

CHAR_ERG = STRINGVAR [IDX]	Tipo risultante: CHAR
CHAR_ERG = STRINGFELD [IDX_FELD, IDX_CHAR]	Tipo risultante: CHAR

Descrizione

Viene scritto/letto un carattere nell'ambito di una stringa nella posizione indicata. Se l'indicazione della posizione è negativa o maggiore della stringa, viene emesso un allarme.

Esempio di messaggi:

Impiego di un indicatore dell'asse in una stringa già esistente.

```
DEF STRING [50] MESSAGGIO = "L'asse n ha
raggiunto la posizione"
MESSAGGIO [6] = "X"
MSG (MESSAGGIO)                ;emette il messaggio "L'asse X ha
                                ;raggiunto la posizione"
```

Parametri

L'accesso al singolo carattere è possibile solo per le variabili definite dall'utente (dati LUD, GUD e PUD).

Inoltre questo tipo di accesso, nel caso di un richiamo di un sottoprogramma, è possibile solo per i parametri del tipo "Call-By-Value".

Esempio di accesso ad un singolo carattere di un dato macchina, di sistema, ...

```
DEF STRING [50] STRG
DEF CHAR CONFERMA
...
STRG = $P_MMCA
CONFERMA = STRG [0]            ;Valutazione dei componenti di CONFERMA
```

Accesso al singolo carattere con parametri Call-By- Reference

```
DEF STRING [50] STRG
DEF CHAR CHR1
EXTERN UP_CALL (VAR CHAR1)    ;Parametro Call-By-Reference!
...
CHR1 = STRG [5]
UP_CALL (CHR1)                ;Call-By-Reference
STRG [5] = CHR1
```

1.11 Istruzione CASE

Funzione

L'istruzione CASE offre la possibilità di diverse diramazioni in funzione del valore attuale del tipo INT.

In funzione del valore assunto dalla costante controllata dall'istruzione CASE, il programma si dirama nella posizione determinata dalla corrispondente destinazione del salto.

Programmazione

```
CASE (espressione) OF costante1 GOTO LABEL1 ... DEFAULT GOTO LABELn
CASE (espressione) OF costante1 GOTOB LABEL1 ... DEFAULT GOTOB LABELn
```

Parametri

CASE	Parola chiave per l'istruzione di salto
GOTOB	Istruzione di salto, con salto all'indietro (direzione inizio programma)
GOTOF	Istruzione di salto con destinazione di salto in avanti (direzione fine programma)
GOTO	Istruzione di salto dove la destinazione di salto è in primo luogo in avanti e poi all'indietro (prima verso la fine programma e poi verso l'inizio)
GOTOC	Soppressione dell'allarme 14080 "Destinazione di salto non trovata". Istruzione di salto dove la destinazione di salto è in primo luogo in avanti e poi all'indietro (prima verso la fine programma e poi verso l'inizio)
LABEL	Destinazione (label all'interno del programma)
LABEL:	Dopo il nome della destinazione del salto deve seguire un doppio punto
espressione	Espressione aritmetica
costante	Costante del tipo INT
DEFAULT	Percorso programma, quando non subentra alcuna delle costanti citate

Nota

Per ulteriori informazioni sui comandi GOTO, consultare il cap.10 Parametri di calcolo e salti in programma.

Nei casi in cui la costante non assume alcuno dei valori predeterminati, con l'istruzione DEFAULT è possibile determinare una destinazione del salto.

Se non viene programmata l'istruzione DEFAULT, come destinazione del salto viene considerato il blocco successivo all'istruzione CASE.

Esempio 1

```
CASE(espressione) OF 1 GOTOF LABEL1 2 GOTOF LABEL2 ... DEFAULT GOTOF LABELn
```

"1" e "2" sono costanti possibili.

Se il valore dell'espressione è = 1 (costante INT), salta al blocco con LABEL1

Se il valore dell'espressione è = 2 (costante INT), salta al blocco con LABEL2

...

altrimenti salta al blocco con LABELn

Esempio 2

```
DEF INT VAR1 VAR2 VAR3  
CASE (VAR1+VAR2-VAR3) OF 7 GOTOF MARKE1 9 GOTOF MARKE2 DEFAULT GOTOF MARKE3  
MARKE1: G0 X1 Y1  
MARKE2: G0 X2 Y2  
MARKE3: G0 X3 Y3
```

1.12 Strutture di controllo

Funzione

Il controllo numerico normalmente elabora i blocchi nella sequenza programmata.

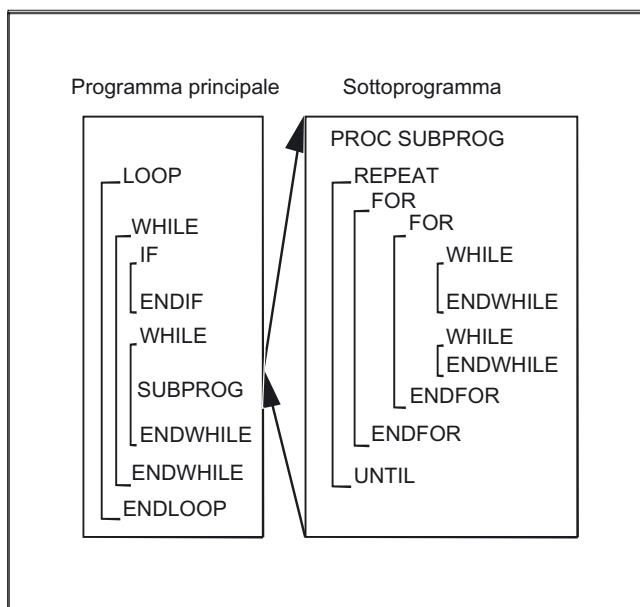
Con questi comandi è possibile definire, oltre ai salti di programma descritti in questo capitolo, ulteriori alternative e loop di programma.

Questi comandi consentono programmazioni strutturate e una buona leggibilità dei programmi.

Programmazione

Profondità di annidamento

Le strutture di controllo hanno validità locale. All'interno di ogni livello di sottoprogramma è consentita una profondità di annidamento max. di 8 strutture di controllo.



Cautela

Le strutture di controllo sono consentite solo nella parte istruzioni di un programma. Le definizioni nell'intestazione del programma non possono essere ripetute o eseguite in modo condizionato.

Le parole chiave per le strutture di controllo e per le destinazioni dei salti non possono essere sovrapposte dalle macro. La definizione delle macro non viene effettuata.

Parametri

IF	Scelta tra 2 alternative
LOOP	Loop senza fine
FOR	Loop con conteggio
WHILE	Loop con condizione all'inizio del Loop
REPEAT	Loop con condizione alla fine del Loop

Esempio di programma senza fine

```
%_N_LOOP_MPF
LOOP
  IF NOT $P_SEARCH ;nessuna ricerca blocco
  G01 G90 X0 Z10 F1000
  WHILE $AA_IM[X] <= 100
  G1 G91 X10 F500 ;dima di foratura
  Z-F100
  Z5
  ENDWHILE
  Z10
  ELSE ;ricerca blocco
  MSG("In ricerca non viene forato")
  ENDIF
  $A_OUT[1] = 1 ;foratura successiva
  G4 F2
  ENDLOOP
M30
```

Esempio di produzione di un numero fisso di pezzi

```
%_N_NRPEZZI_MPF

DEF INT NR_PEZZI
FOR NR_PEZZI = 0 TO 100
  G01 ...
ENDFOR
M30
```


Esempio per variabile di tipo INT e REAL come loop con conteggio

%_N_CONTEGGIOVAR._MPF	
	;attuale possibilità di programmazione
DEF INT iVARIABILE1	
R10 = R12-R20*R1 R11 = 6	
FOR iVARIABILE1 = R10 TO R11	;Variabile di conteggio = Variabile Integer
R20 = R21*R22 + R33	
ENDFOR	
M30	
	;possibilità di programmazione ampliata
R11 = 6	
FOR R10 = R12-R20*R1 TO R11	;Variabile di conteggio = Parametri R (variabile reale)
R20 = R21*R22 + R33	
ENDFOR	
M30	

Influenza sui tempi di elaborazione

Di regola, con il funzionamento attivo dell'interprete, utilizzando salti in programma si può velocizzare notevolmente l'elaborazione del programma stesso rispetto alle strutture di controllo.

Nel caso di cicli precompilati non esiste alcuna differenza tra salti in programma e strutture di controllo.

Condizioni marginali

I blocchi con elementi della struttura di controllo, non possono essere mascherati. Le label ("etichette"), in alcuni blocchi non sono consentite.

Le strutture di controllo vengono elaborate in modo interpretativo. Se viene riconosciuta la fine di un loop, la ricerca avviene secondo la struttura di controllo a partire dall'inizio del loop.

Di conseguenza, nel funzionamento normale dell'interprete dei comandi la struttura a blocchi del programma non viene verificata completamente.

In linea di massima si consiglia di non utilizzare in contemporanea strutture di controllo e salti in programma.

Nella preelaborazione dei cicli può essere verificata la correttezza dell'annidamento delle strutture di controllo.

Esecuzione

1. IF-ELSE-ENDIF

Un blocco IF-ELSE-ENDIF-serve per la scelta tra 2 alternative:

IF (espressione)

Blocchi NC

ELSE

Blocchi NC

ENDIF

Se l'espressione ha valore TRUE ossia la condizione è soddisfatta, viene eseguito il successivo blocco di programma. Se la condizione non è soddisfatta, viene eseguita la diramazione ELSE.

La diramazione ELSE può anche essere omessa.

2. Loop di programma permanente LOOP

Il loop di programma permanente, trova applicazione in programmi senza termine. A fine programma avviene sempre un salto a ritroso all'inizio del loop.

LOOP

Blocchi NC

ENDLOOP

3. Loop di conteggio FOR

Il loop FOR viene utilizzato quando un ciclo di lavorazione deve essere ripetuto un determinato numero di volte. La variabile di conteggio viene incrementata dal valore iniziale a quello finale. Il valore iniziale deve essere sempre minore di quello finale. La variabile di conteggio può essere di tipo INT o REAL. Il tipo REAL serve, ad esempio, per programmare parametri R per un loop di conteggio. Se la variabile di conteggio è di tipo REAL, il relativo valore viene arrotondato a un numero intero.

Variabile **FOR** = valore iniziale **TO** valore finale

Blocchi NC

ENDFOR

4. Loop di programma con condizione a inizio loop WHILE

Il loop WHILE viene eseguito finché la condizione rimane soddisfatta.

WHILE espressione

Blocchi NC

ENDWHILE

5. Loop di programma con condizione a fine loop REPEAT

Il loop REPEAT viene eseguito una volta e ripetuto finché la condizione non viene soddisfatta.

REPEAT

Blocchi NC

UNTIL (espressione)

1.13 Coordinamento dei programmi

Funzione

Canali

Un canale è in grado di elaborare il proprio programma indipendentemente da altri canali. Gli assi ed i mandrini, che gli vengono attribuiti temporaneamente, vengono controllati tramite il programma.

Al momento della messa in servizio si possono creare due o più canali per il controllo numerico.

Coordinamento dei programmi

Se per la produzione di un pezzo vengono impiegati più canali può rendersi necessaria la sincronizzazione delle sequenze di programma.

Per la sincronizzazione sono necessarie istruzioni particolari (comandi) che si trovano in un blocco a sé stante.

Nota

Il coordinamento dei programmi è possibile anche nel proprio canale.

Istruzioni per il coordinamento dei programmi

- Impostazione con immissione assoluta del percorso

INIT (n,"/_HUGO_DIR/_N_name_MPF")
oppure

INIT (n,"/_N_MPF_DIR/_N_name_MPF")

Esempio:

INIT(2,"/_N_WKS_DIR/_ABRICHT_MPF")
G01 F0.1
START

INIT
(2,"/_N_WKS_DIR/_N_UNTER_1_SPF")

- Indicazione con impostazione relativa del percorso

Il percorso assoluto viene impostato secondo le seguenti regole:

- directory corrente/_N_name_MPF
"directory corrente" indica la directory del pezzo selezionata oppure la directory standard /_N_MPF_DIR.

- Scelta di un determinato programma per l'elaborazione in un determinato canale:
n: numero del canale, il valore dipende dalla configurazione del controllo
- Nome programma completo

Fino alla versione software 3:

Tra un comando **init** (senza sincronizzazione) e un comando **NC-Start** deve trovarsi almeno un blocco eseguibile.

Nei richiami di sottoprogrammi occorre aggiungere "_SPF" all'indicazione del percorso

Esempio:

```
INIT(2,"ABRICHT")
INIT(3,"UNTER_1_SPF")
```

Per l'impostazione relativa del percorso valgono le stesse regole del richiamo di sottoprogrammi.

Per richiami di sottoprogrammi occorre aggiungere "_SPF" al nome del programma

Parametri

Per lo scambio dei dati tra i programmi si possono utilizzare le variabili che sono comuni a tutti i canali (variabili globali specifiche per NCK). Negli altri casi i programmi per i canali vengono generati separatamente.

INIT(n, indicazione di percorso, modo di tacitazione)

Istruzione per l'esecuzione in un canale. Selezione di un determinato programma con indicazione di percorso assoluta o relativa.

START (n, n)

Avvio dei programmi selezionati negli altri canali. n,n: Enumerazione del numero di canale: il valore dipende dalla configurazione del controllo

WAITM (numero marker, n, n, ...)

Settare il marker "n. marker" nel proprio canale. Concludere il blocco precedente con un arresto preciso. Attendere i marker con lo stesso "n. marker" nei canali "n" impostati (non è necessario impostare il proprio canale). Il marker viene cancellato dopo la sincronizzazione.

È possibile settare fino a 10 marker per ogni canale.

WAITMC (n. marker, n, n, ...)

Settare il marker "n. marker" nel proprio canale. L'arresto preciso viene avviato solo se gli altri canali non hanno ancora raggiunto il marker. Attendere il marker con lo stesso "n. marker" dei canali "n" impostati (non è necessario impostare il proprio canale). La lavorazione può essere proseguita senza terminare l'arresto preciso non appena il marker "n. marker" è stato raggiunto nei canali impostati.

WAITE (n, n, ...)

Attendere la fine del programma dei canali impostati (non impostare il proprio canale) Esempio: programmazione di un tempo di sosta dopo il comando di Start.

```
N30 START(2)
N31 G4 F0.01
N40 WAITE(2)
```

SETM (n. marker, n. marker, ...)

Settare il marker "Marker-Nr." nel proprio canale, senza influenzare la lavorazione in corso. Il comando SETM(0) rimane valido anche in seguito a RESET o START-NC e

CLEARM (n. marker, n. marker, ...)

n

Cancellare il marker "Marker-Nr." nel proprio canale, senza influenzare la lavorazione in corso. Tutti i marker del canale possono essere cancellati con CLEARM(). CLEARM (0) cancella il marker "0". Il comando CLEARM() rimane valido anche in seguito a RESET e START-NC.

Numero di canale o nome di canale corrispondente

Nota

Tutti i comandi nominati sopra devono trovarsi in blocchi a sé stante.

Il numero dei marker dipende dalla CPU installata.

Numeri di canale

Per i canali da coordinare è possibile immettere un numero di canali fino a 10 (valore Integer).

Nomi dei canali

I nomi dei canali devono essere convertiti in numeri mediante variabile (vedere capitolo "Variabili e parametri di calcolo") oppure al posto dei numeri dei canali possono essere programmati i nomi dei canale (identificatori o parole chiave) definiti tramite \$MC_CHAN_NAME. I nomi definiti devono corrispondere alle convenzioni linguistiche NC (cioè i primi due caratteri devono essere o lettere o un carattere di sottolineatura).



Cautela

Occorre assicurarsi che i numeri assegnati non possano essere facilmente modificati!

I nomi non devono essere già presenti nell'NC in un'altra accezione, ad es. come parola chiave, comando di linguaggio, nome di asse, ecc.

SETM() e CLEARM()

SETM() e CLEARM() possono anche essere programmati in un'azione sincrona. Vedere capitolo "Settare/cancellare label di attesa: SETM CLEARM"

Esempio

Al canale dal nome "MACCHINA" si vuole assegnare il numero di canale 1,

Al canale dal nome "CARICATORE" si vuole assegnare il numero di canale 2:

```
DEF INT MACCHINA = 1, CARICATORE = 2
```

Le variabili ricevono lo stesso nome dei canali.

L'istruzione `START` sarà quindi per esempio:

```
START (MACCHINA)
```

Esempio di coordinamento di programmi

Canale 1:

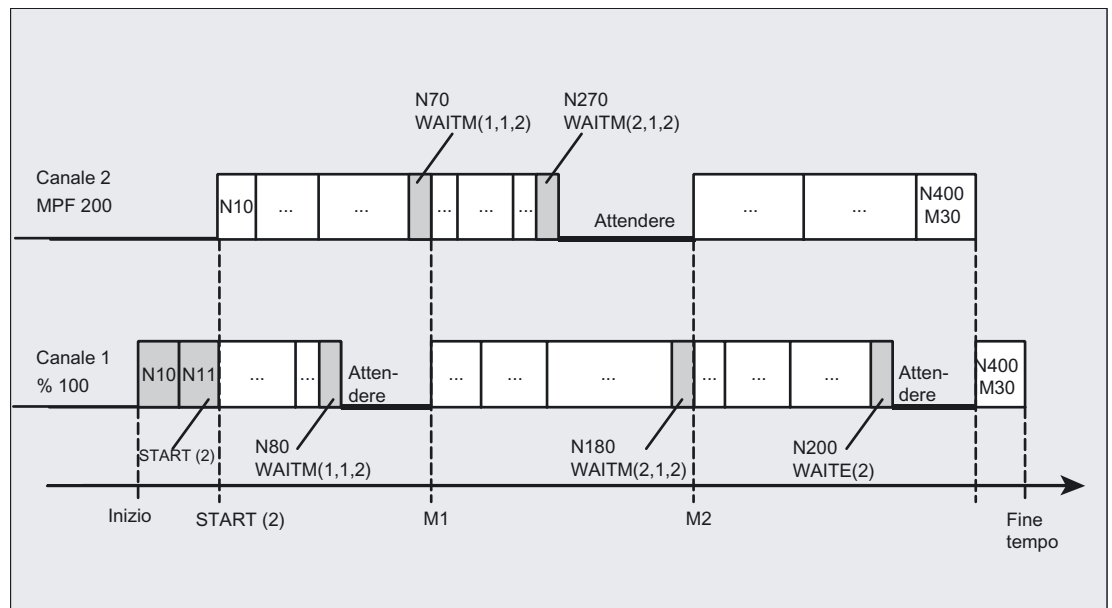
```
%_N_MPF100_MPF
```

N10	INIT(2, "MPF200")	
N11	START(2)	;Lavorazione nel canale 2
.		
N80	WAITM(1,1,2)	;Attendere il Marker WAIT 1 nel canale 1 e 2
.		;ulteriore lavorazione nel canale 1
N180	WAITM(2,1,2)	;Attendere il Marker WAIT 2 nel canale 1 e 2
.		;ulteriore lavorazione nel canale 1
N200	WAITE(2)	;Attendere la fine programma del canale 2
N201	M30	;Fine programma canale 1, fine globale
...		

Canale 2:

```
%_N_MPF200_MPF
```

	; \$PATH=/_N_MPF_DIR	
		;Lavorazione nel canale 2
N70	WAITM(1,1,2)	;Attendere il Marker WAIT 1 nel canale 1 e 2
.		;ulteriore lavorazione nel canale 1
N270	WAITM(2,1,2)	;Attendere il Marker WAIT 2 nel canale 1 e 2
.		;ulteriore lavorazione nel canale 2
N400	M30	;Fine programma del canale 2



Esempio di programma dal pezzo

```
N10 INIT(2, "/_N_WKS_DIR/_N_ALBERO1_WPD/_N_ABSPAN1_MPF")
```

Esempio di comando INIT con relativa indicazione del percorso

Nel canale 1 è stato selezionato il programma /_N_MPF_DIR/_N_MAIN_MPF

```
N10 INIT(2, "MYPROG") ;selezionare il programma /_N_MPF_DIR/_N_MYPROG_MPF nel
;canale 2
```

Esempio di nome di canale e numero di canale con variabile Integer

```
$MC_CHAN_NAME[0] = "CHAN_X" ;nome del primo canale
$MC_CHAN_NAME[1] = "CHAN_Y" ;nome del secondo canale
```

```
START(1, 2) ;eseguire Start nel 1° e nel 2° canale
```

Analogicamente, programmazione con gli identificatori di canale:

```
START(CHAN_X, CHAN_Y) ;Eseguire Start nel 1° e nel 2° canale
;Gli identificatori canale_X e canale_Y
;rappresentano internamente in base al dato macchina
;$MC_CHAN_NAME
;i numeri di canale 1 e 2. Eseguire quindi
;uno Start nel primo e nel secondo canale
```

Programmazione con variabile Integer:

```
DEF INT chanNo1, chanNo2) ;Definire il numero di funzione.
chanNo1=CHAN_X
chanNo2=CHAN_Y
START(chanNo1, chanNo2)
```

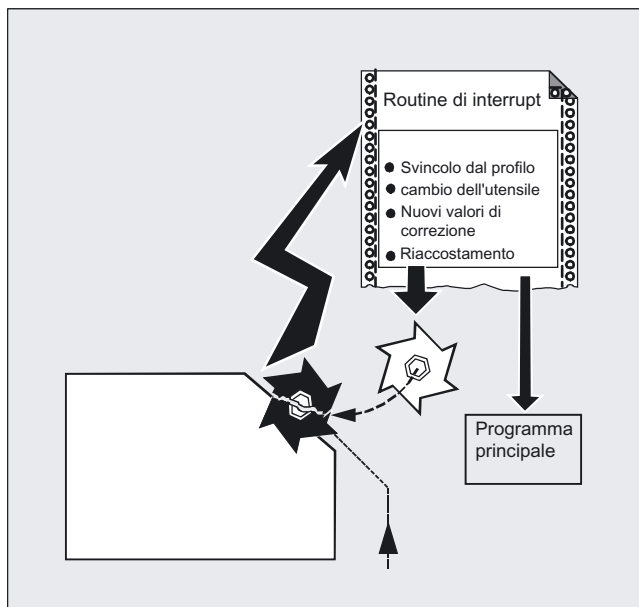
1.14 Routine di interrupt (SETINT, DISABLE, ENABLE, CLRINT)

Funzione

Le relazioni relative alla programmazione di una routine di interrupt vengono chiarite sulla base di un esempio tipico:

durante la lavorazione si rompe un utensile. Questo provoca un segnale che interrompe la lavorazione in corso e inizia un sottoprogramma - la cosiddetta routine di interrupt. Questo sottoprogramma contiene tutte le istruzioni che devono essere eseguite in questo caso.

Terminato il sottoprogramma (che ha ristabilito lo stato di pronto al funzionamento) il controllo ritorna al programma principale e, in base al comando REPOS , prosegue la lavorazione dal punto in cui era stata interrotta.



Per ulteriori informazioni relative a REPOS vedere "Riaccostamento al profilo".

Programmazione

```
SETINT(3) PRIO=1 NAME  
SETINT(3) PRIO=1 LIFTFAST  
SETINT(3) PRIO=1 NAME LIFTFAST  
G... X... Y... ALF=...  
DISABLE (3)  
ENABLE (3)  
CLRINT (3)
```


Parametri

SETINT (n)	Inizializzare la routine di interrupt quando commuta l'ingresso n, n (1...8) indica il numero dell'ingresso
PRIO=1	Definizione della priorità da 1 a 128 (1 ha la precedenza)
LIFTFAST	Svincolo rapido dal profilo
NOME	Qui si trova il nome del sottoprogramma che deve essere eseguito
ALF=...	Direzione di movimento programmabile (si trova nel blocco di movimento)
DISABLE (n)	Disattivare la routine di interrupt n
ENABLE (n)	Riattivare la routine di interrupt n
CLRINT (n)	Cancellare le assegnazioni di interrupt dai numeri n delle routine di interrupt

Movimento di ritorno

La direzione del movimento di svincolo viene programmata con i codici G **LFTXT** oppure **LFWP** con la variabile **ALF**.

- **LFTXT**
Il piano del movimento di svincolo viene determinato dalla tangente al profilo e dalla direzione utensile. Con questo codice G (preimpostazione standard) viene programmato l'abituale svincolo rapido.
- **LFWP**
Il piano del movimento di svincolo è il piano di lavoro attivo, che viene selezionato con i codici G G17, G18 o G19. La direzione del movimento di svincolo è indipendente dalla tangente al profilo. È possibile programmare lo svincolo parallelo all'asse.
- **LFPOS**
Svincolo dell'asse dichiarato con POLFMASK sulla posizione assoluta dell'asse programmata con POLF. Vedere anche "Svincolo NC controllato" nel Manuale delle funzioni M3.
ALF non ha nessuna influenza sulla direzione di svincolo per più assi e nemmeno per più assi in vincolo lineare.

Nel piano del movimento di svincolo è ancora possibile programmare con **ALF** la direzione in passi discreti di 45 gradi. Con **LFTXT**, lo svincolo per ALF=1 è definito nella direzione dell'utensile.

Con **LFWP** la direzione nel piano di lavoro si ricava nel modo seguente:

- **G17**:piano X/Y ALF=1 svincolo in direzione X
ALF=3 svincolo in direzione Y
- **G18**:piano Z/X ALF=1 svincolo in direzione Z
ALF=3 svincolo in direzione X
- **G19**:piano Y/Z ALF=1 svincolo in direzione Y
ALF=3 svincolo in direzione Z

Esempio

In questo esempio si intende sostituire automaticamente un utensile non più utilizzabile con un utensile gemello. La lavorazione viene proseguita quindi con il nuovo utensile.

Programma principale

```
N10 SETINT(1) PRIO=1 W_WECHS ->
-> LIFTFAST

N20 G0 Z100 G17 T1 ALF=7 D1
N30 G0 X-5 Y-22 Z2 M3 S300
N40 Z-7
N50 G41 G1 X16 Y16 F200
N60 Y35
N70 X53 Y65
N90 X71.5 Y16
N100 X16
N110 G40 G0 Z100 M30
```

Quando diventa attivo l'ingresso 1, l'utensile viene allontanato subito dal pezzo mediante il distacco rapido (nr. di codice 7 per la correzione del raggio utensile G41). Viene poi eseguita la routine di interrupt W_WECHS.

Sottoprogramma

```
PROC W_WECHS SAVE

N10 G0 Z100 M5
N20 T11 M6 D1 G41
N30 REPOS L RMB M3

-> viene programmata in un blocco a sé
stante.
```

Sottoprogramma con memorizzazione dello stato operativo attuale
;Posizione di cambio utensile, arresto mandrino
;cambio utensile
;Riaccostamento al profilo e ritorno al
;programma principale



Cautela

Se nel sottoprogramma non viene programmato alcun comando REPOS, il posizionamento avverrà sul punto finale del blocco successivo al blocco interrotto.

Generazione di una routine di interrupt come sottoprogramma

La routine di interrupt viene definita come un sottoprogramma.

Esempio:

```
PROC ABHEB_Z
N10 ...
N50 M17
```

Nome programma ABHEB_Z, poi seguono i blocchi NC, la fine programma M17 ed il ritorno al programma principale.

Nota

All'interno di una routine di interrupt si possono programmare comandi SETINT, abilitando così ulteriori routine di interrupt che vengono attivate dall'ingresso.

Per ulteriori informazioni riguardanti la generazione di sottoprogrammi vedere il capitolo "Tecnica dei sottoprogrammi, tecnica macro".

Memorizzazione della posizione di interruzione, SAVE

La routine di interrupt nelle riga di definizione può essere contrassegnata con `SAVE`.

Esempio:

```
PROC ABHEB_Z SAVE
N10 ...
N50 M17
```

Tramite l'attributo `SAVE` le funzioni G modali vengono impostate, alla fine della routine di interrupt, al valore che avevano all'avvio della routine di interrupt. Oltre allo spostamento origine impostabile (gruppo di funzioni G modali 8), vengono ripristinati lo spostamento origine programmabile e la traslazione di base. Se si verifica una variazione del gruppo di funzioni G 15 (tipo di avanzamento), per esempio da `G94` a `G95`, viene ripristinato anche il corrispondente valore F.

Grazie a queste funzioni la lavorazione può proseguire dal punto in cui era stata interrotta.

Assegnazione e avvio della routine di interrupt, SETINT

Il controllo dispone di segnali

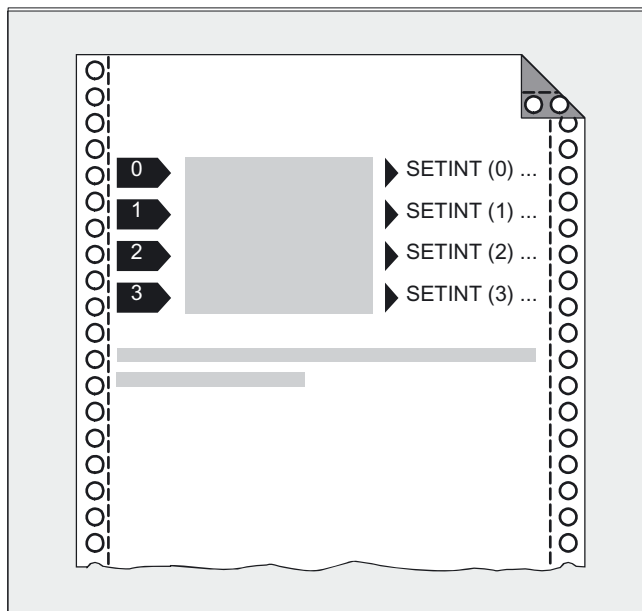
(ingressi 1...8) che sono in grado di provocare un'interruzione del programma in corso e di avviare una routine di interrupt.

Nel programma di lavorazione ad ogni programma verrà assegnato il relativo ingresso, che avrà il compito di inizializzarlo.

Esempio:

```
N10 SETINT(3) PRIO=1 ABHEB_Z
```

Attivando l'ingresso 3 viene subito inizializzata la routine `ABHEB_Z`.



Avvio di piú routine di interrupt, definizione della sequenza, PRIO=

Se in un programma NC sono impostati diverse istruzioni SETINT e risulta quindi possibile che arrivino piú segnali contemporaneamente, è necessario ordinare le routine di interrupt secondo la gerarchia in cui si intende che vengano eseguite: PRIO da 1 a 128 ha la priorità.

Esempio:

```
N10 SETINT(3) PRIO=1 ABHEB_Z
N20 SETINT(2) PRIO=2 ABHEB_X
```

Le routine vengono eseguite secondo l'ordine gerarchico dato se gli ingressi dovessero essere presenti contemporaneamente. Prima SETINT(3), poi SETINT(2).

Se durante l'esecuzione della routine di interrupt arrivano nuovi segnali, le routine con priorità maggiore interrompono quella in corso.

Disattivazione/riattivazione delle routine di interrupt, DISABLE, ENABLE

Le routine di interrupt nel programma NC possono essere disattivate con DISABLE(n) e riattivate con ENABLE(n) (n si riferisce al numero dell'ingresso).

L'abbinamento ingresso/routine viene mantenuto anche in caso di DISABLE e viene riattivato con ENABLE.

Nuova assegnazione di routine di interrupt

Se si assegna una nuova routine a un ingresso già occupato, la vecchia assegnazione diventa automaticamente invalida.

Esempio:

```
N20 SETINT(3) PRIO=2 ABHEB_Z
```

...

...

```
N120 SETINT(3) PRIO=1 ABHEB_X
```

Cancellazione dell'abbinamento, CLRINT

Con CLRINT(n) si possono cancellare gli abbinamenti.

Esempio:

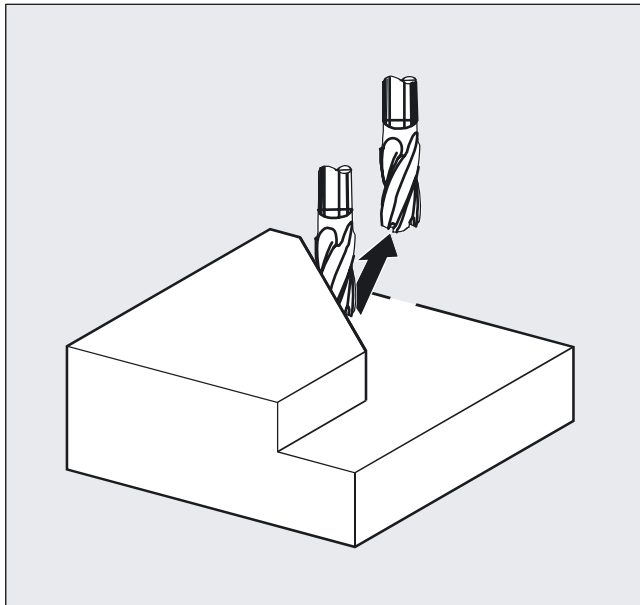
```
N20 SETINT(3) PRIO=2 ABHEB_Z
```

```
N50 CLRINT(3)
```

L'abbinamento fra l'ingresso 3 e la routine ABHEB_Z viene cancellato.

Distacco rapido dal profilo, LIFTFAST

LIFTFAST consente il distacco rapido dell'utensile dal profilo del pezzo con l'attivazione di un ingresso.



Se l'istruzione SETINT oltre a LIFTFAST contiene anche una routine di interrupt, il distacco rapido viene effettuato **prima** della routine di interrupt.

Esempio:

```
N10 SETINT(2) PRIO=1 LIFTFAST
```

oppure

```
N30 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST
```

In entrambi i casi al momento dell'attivazione dell'ingresso 2 il distacco rapido viene eseguito con priorità assoluta.

- In N10 la lavorazione viene interrotta con l'allarme 16010 (perchè non è stato indicato un sottoprogramma asincrono, ASUP).
- In N30 viene eseguito l' ASUP "ABHEB-Z".

Nella definizione della direzione del distacco viene controllato se è attivo un frame con specularità. In questo caso vengono scambiate tra di loro destra e sinistra con la direzione di distacco riferita alla tangente. Le componenti dello spostamento nella direzione dell'utensile non vengono speculate. Questo comportamento viene attivato tramite DM \$MC_LIFTFAST_WITH_MIRROR=TRUE

Sequenza di movimento con distacco rapido

La distanza che dovranno osservare gli assi geometrici al momento del distacco rapido è impostabile con dato macchina.

Routine di interrupt senza LIFTFAST

Frenatura in traiettoria ed avvio della routine di interrupt non appena il movimento in traiettoria è terminato.

Questa posizione viene memorizzata come posizione di interruzione e viene raggiunta in REPOS con RMI al termine della routine di interrupt.

Routine di interrupt con LIFTFAST

Frenatura in traiettoria ed esecuzione contemporanea del movimento LIFTFAST come movimento sovrapposto. Al termine dei movimenti secondo la traiettoria e LIFTFAST, viene startata la routine di interrupt.

Come posizione di interruzione viene memorizzata quella sul profilo al momento dell'avvio del movimento LIFTFAST con conseguente abbandono della traiettoria.

La routine di interrupt con LIFTFAST e ALF=0 si comporta allo stesso modo di quella senza LIFTFAST.

Direzione di movimento programmabile, ALF=...

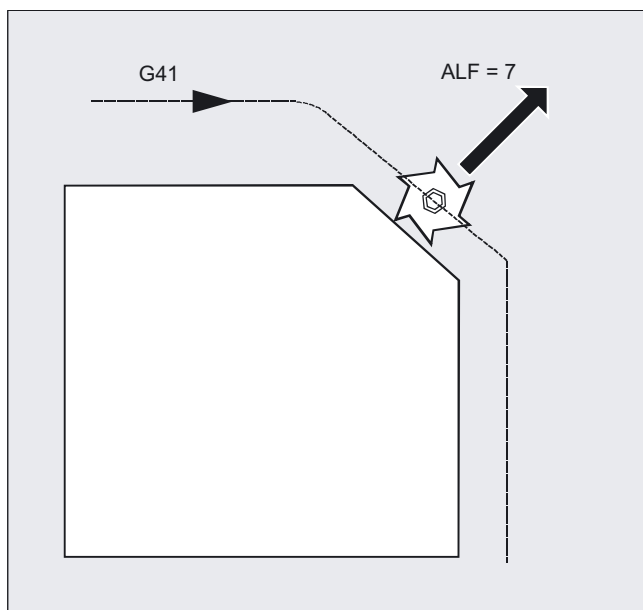
La direzione in cui deve spostarsi l'utensile al momento del distacco rapido va impostata nel programma NC.

Le direzioni possibili sono memorizzate nel controllo NC con numeri di codice specifici e vanno anche richiamate con tali numeri.

Esempio:

```
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST  
ALF=7
```

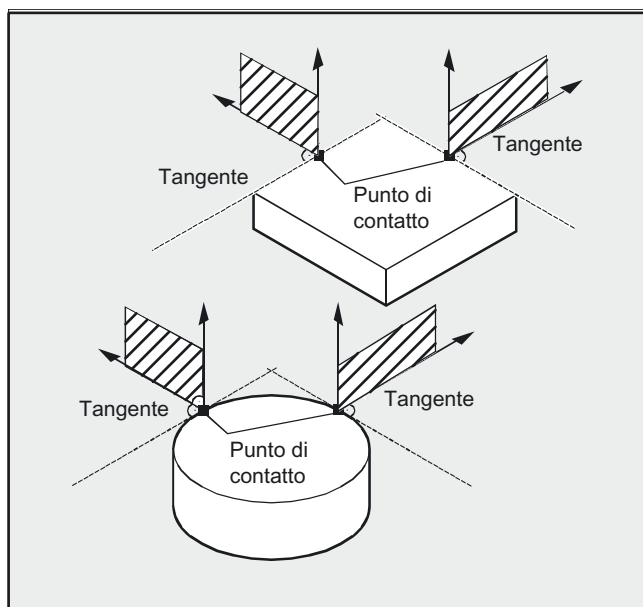
Osservando dall'alto, l'utensile si stacca ortogonalmente dal profilo con G41 attivata (direzione di lavorazione a sinistra del profilo).



Piano di riferimento per la descrizione delle direzioni di movimento

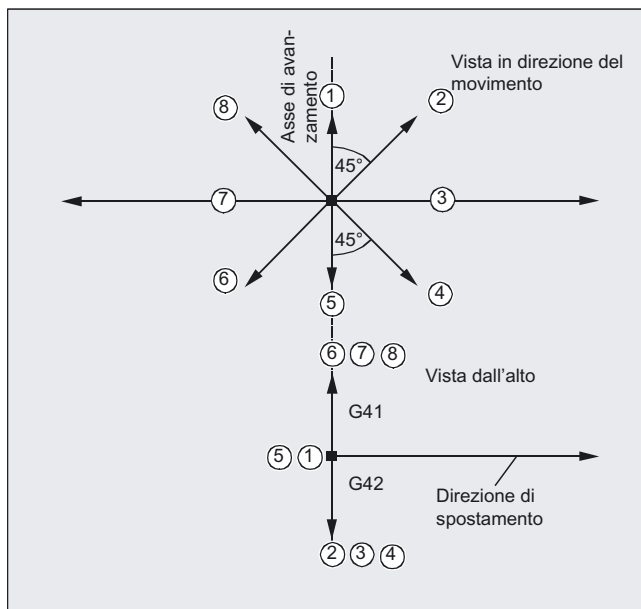
Sul punto di contatto dell'utensile sul profilo programmato viene definito un piano che servirà come riferimento per le indicazioni relative al movimento di distacco con il numero di codice corrispondente.

I limiti del piano di riferimento sono costituiti dall'asse longitudinale dell'utensile (direzione di incremento) e da un vettore, posizionato perpendicolarmente rispetto alla tangente sul punto di contatto dell'utensile sul profilo.



Panoramica dei numeri di codice con le direzioni di movimento

Partendo dal piano di riferimento, la figura a fianco riporta i numeri di codice con le direzioni di movimento.



Con $ALF=0$ la funzione distacco rapido viene disattivata.



Cautela

Quando è attiva la correzione raggio utensile
con G41 le codifiche 2, 3, 4 e
con G42 le codifiche 6, 7, 8
non dovrebbero essere utilizzate.

In caso contrario l'utensile proseguirebbe verso il profilo e andrebbe a collidere con il pezzo.

1.15 Scambio assi, scambio mandrini (RELEASE, GET, GETD)

Funzione

Uno o più assi o mandrini possono essere interpolati sempre in un solo canale. Se un asse deve lavorare alternativamente in due diversi canali (per es. un cambio-pallet), occorre abilitarlo prima nel canale attuale e poi prenderlo in gestione nel secondo canale. L'asse viene scambiato tra i canali.

Ampliamenti per lo scambio assi

Un asse/mandrino può essere scambiato con stop preelaborazione e sincronizzazione tra preelaborazione ed elaborazione principale o alternativamente anche senza stop preelaborazione. Inoltre lo scambio asse è possibile anche tramite:

- Rotazione del container assi AXCTSWE o AXCTWED tramite GET/GETD implicito.
- Frame o rotazione se questo asse è collegato ad altri assi.
- Azioni sincrone, vedere azioni sincrone di movimento, "Scambio asse RELEASE, GET".

Costruttore della macchina

Fare attenzione alle indicazioni del costruttore della macchina. Mediante dati macchina progettabili un asse deve essere definito in modo univoco per lo scambio in tutti i canali e il processo di scambio assi è modificabile anche mediante dato macchina.

Programmazione

RELEASE (nomeasse, nomeasse, ...) oppure RELEASE (S1)

GET (nomeasse, nomeasse, ...) oppure GET (S2)

oppure

GETD (nome asse, nome asse, ...) oppure GET (S3)

Con GETD (GET Directly) un asse viene trasferito direttamente da un altro canale. Ciò significa che per questo GETD non occorre programmare RELEASE in un altro canale. Questo significa però anche che sarà necessario realizzare una nuova comunicazione tra canali (ad es. marker WAIT).

Parametri

RELEASE (nomeasse, nomeasse, ...)	Abilitazione asse(i)
GET (nomeasse, nomeasse, ...)	Conferma asse(i)
GETD (nomeasse, nomeasse, ...)	Conferma diretta asse/i
Nome degli assi	Abbinamento dell'asse nel sistema: AX1, AX2, ... oppure indicazione del nome dell'asse macchina
RELEASE (S1)	Abilitazione dei mandrini S1, S2, ...
GET (S2)	Conferme dei mandrini S1, S2, ...
GETD (S3)	Conferme diretta dei mandrini S1, S2, ...

Richiesta GET senza stop preelaborazione

Se dopo una richiesta GET **senza** stop preelaborazione l'asse viene riabilitato con RELEASE (asse) o WAITP (asse), un successivo GET comporta un GET **con** stop preelaborazione.



Cautela

Un asse o un mandrino confermato con GET rimane assegnato al canale anche dopo un RESET da tastiera o programma.

Con un nuovo start programma, nel caso l'asse risulti necessario nel canale di base, l'assegnazione degli assi o dei mandrini scambiati deve avvenire da programma.

In caso di POWER On l'asse viene assegnato al canale definito nel dato macchina.

Esempio di scambio assi tra due canali

Dei 6 assi, per l'elaborazione nel canale 1 vengono utilizzati: 1°, 2°, 3° e 4° asse. Gli assi 5 e 6 vengono impiegati nel canale 2 per il cambio del pezzo.

L'asse 2 deve essere cambiato fra i due canali e venire riassegnato al canale 1 dopo POWER ON.

Programma "MAIN" nel canale 1

%_N_MAIN_MPF	
INIT (2, "CAMBIO2")	;Selezionare il programma CAMBIO2 nel canale 2
N... START (2)	;Avviare il programma nel canale 2
N... GET (AX2)	;Accettare l'asse AX2
...	
...	
N... RELEASE (AX2)	;Abilitare l'asse AX2
N... WAITM (1,1,2)	;Attendere il marker WAIT nel canale 1 e 2 ;per la sincronizzazione in entrambi i canali
N ...	;Proseguimento normale dopo lo scambio dell'asse
N... M30	

Programma "Cambio2" nel canale 2

%_N_CAMBIO2_MPF	
N... RELEASE (AX2)	
N160 WAITM (1,1,2)	;Attendere il marker WAIT nel canale 1 e 2 ;per la sincronizzazione in entrambi i canali
N150 GET (AX2)	;Accettare l'asse AX2
N ...	;Proseguimento normale dopo lo scambio dell'asse
N... M30	

Esempio di scambio senza sincronizzazione

Se l'asse non deve essere sincronizzato, GET non provoca l'arresto della preelaborazione.

N01 G0 X0	
N02 RELEASE (AX5)	
N03 G64 X10	
N04 X20	
N05 GET (AX5)	;Se non è necessaria una sincronizzazione ;questo blocco non è eseguibile.
N06 G01 F5000	;Blocco non eseguibile.
N07 X20	;Blocco non eseguibile in quanto la posizione X ;è come in N04.
N08 X30	;Primo blocco eseguibile dopo N05.
N09 ...	

Esempio di attivazione di uno scambio assi senza stop elaborazione

Premessa

Lo scambio assi senza stop preelaborazione deve essere progettato mediante un dato macchina.

N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	;L'asse B diventa l'asse neutrale
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	;l'asse non provoca stop preelaborazione/REORG
N041 G4 F2	
N050 M5	
N099 M30	

Se il mandrino o l'asse B viene spostato immediatamente dopo il blocco N023 come **asse PLC**, ad es. a 180 gradi e di nuovo a 1 grado, questo asse diventa nuovamente un asse neutro e non provoca alcuno stop preelaborazione nel blocco N40.

Premessa

Premesse per il cambio assi

- Tramite i dati macchina l'asse deve essere definito in tutti i canali in cui lo si vuole utilizzare.
- Con il dato macchina specifico per **asse** si stabilisce a quale canale deve essere abbinato l'asse dopo POWER ON.

Descrizione

Abilitazione asse: RELEASE

Per l'abilitazione dell'asse occorre considerare quanto segue:

1. L'asse non deve far parte di alcuna trasformazione.
2. Nel caso di accoppiamenti di assi (comando tangenziale), tutti gli assi del raggruppamento devono essere abilitati.
3. In questa condizione non è possibile scambiare un asse di posizionamento simultaneo.
4. Nel caso di un asse Master-Gantry vanno scambiati anche tutti gli assi a seguire.
5. Nel caso di accoppiamenti di assi (assi trascinati, accoppiamento a valore guida, riduttore elettronico) può essere abilitato solo l'asse master del raggruppamento.

Accettazione asse: GET

Con questo comando viene eseguito il vero e proprio cambio assi. La responsabilità per l'asse ricade ora del tutto sul canale in cui è stata programmata l'istruzione.

Effetti di GET:

Cambio asse con sincronizzazione:

La sincronizzazione di un asse si rende necessaria ogni volta che l'asse è stato assegnato temporaneamente ad un altro canale o al PLC, senza che prima di GET fosse avvenuta una sincronizzazione tramite "WAITP", G74 o cancellazione del percorso residuo.

- Arresto della preelaborazione (come per STOPRE)
- La lavorazione viene interrotta fino a quando il cambio non sia completamente terminato.

GET automatico

Se un asse risulta disponibile in un canale ma attualmente non è presente come "Asse di canale" viene eseguito automaticamente GET. Se l'asse (o gli assi) risulta(no) già sincronizzato(i) non viene generato lo stop preelaborazione.

Impostazione variabile del comportamento dello scambio assi

Il punto di rilascio degli assi viene impostato tramite un dato macchina come segue:

- Lo scambio dati automatico fra due canali può avvenire anche quando l'asse è stato portato in una condizione neutrale con WAITP (comportamento come in precedenza)
- Alla richiesta di una rotazione del container assi, tutti gli assi assegnabili al canale attivo del container assi vengono trasferiti nel canale tramite GET o GETD. Un successivo scambio assi è nuovamente ammesso solo una volta conclusa la rotazione del container assi.
- Dopo l'inserimento di un blocco intermedio nell'elaborazione principale viene verificato se è necessaria o meno una riorganizzazione. La riorganizzazione sarà necessaria solo se le condizioni asse di questo blocco **non** coincidono con le condizioni asse attuali.
- A differenza di un blocco GET con stop preelaborazione e sincronizzazione tra preelaborazione ed elaborazione principale, uno scambio assi può avvenire anche senza stop preelaborazione. Viene quindi generato solo un blocco intermedio con la richiesta GET. Nell'elaborazione principale, all'esecuzione di questo blocco viene controllato se gli stati dell'asse nel blocco corrispondono agli stati attuali dell'asse.

Per ulteriori informazioni sulla funzionalità di uno scambio assi o mandrini, vedere /FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; BAG, canali, scambio assi (K5).

1.16 Spostamento di un asse in un altro canale (AXTOCHAN)

Funzione

Con l'istruzione di linguaggio AXTOCHAN può essere richiesto un asse per spostare questo asse in un altro canale. L'asse può essere spostato nel canale corrispondente sia dal partprogram dell'NC, sia da un'azione sincrona.

Programmazione

AXTOCHAN(nome asse,n. canale[,nome asse,n. canale [...]])

Parametro

AXTOCHAN	Richiesta di un asse per un determinato canale
Nome degli assi	Abbinamento dell'asse nel sistema: X, Y, ... oppure indicazione del nome dell'asse macchina interessato. Il canale da eseguire non deve essere il proprio canale e non deve nemmeno essere il canale che dispone dell'autorizzazione di interpolazione attuale per l'asse
Numero del canale	Numero del canale cui deve essere assegnato l'asse

Nota

Asse di posizionamento simultaneo e asse controllato esclusivamente dal PLC

Un asse PLC non può, in quanto asse di posizionamento simultaneo (concorrente), cambiare il canale. Un asse controllato esclusivamente dal PLC non può essere assegnato al programma NC.

Bibliografia

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; assi di posizionamento (P2)

Esempio di AXTOCHAN nel programma NC

Gli assi X e Y sono conosciuti nel 1° canale e nel 2° canale. Attualmente il canale 1 dispone dell'autorizzazione di interpolazione e nel canale 1 viene avviato il seguente programma:

N110 AXTOCHAN(Y, 2)	;spostamento asse Y nel 2° canale
N111 M0	
N120 AXTOCHAN(Y, 1)	;richiamo dell'asse Y (neutro)
N121 M0	
N130 AXTOCHAN(Y, 2, X, 2)	;spostamento asse Y e asse X nel 2° canale (assi neutri)
N131 M0	
N140 AXTOCHAN(Y, 2)	;spostamento asse Y nel 2° canale (programma NC)
N141 M0	

Descrizione

AXTOCHAN nel programma NC

Per il programma NC nel proprio canale, solo ad una richiesta dell'asse viene eseguito un comando GET in modo da attendere anche l'effettiva modifica di stato. Solo se l'asse è richiesto per un altro canale o deve diventare un asse neutro nel proprio canale, viene emessa la richiesta relativa.

AXTOCHAN da un'azione sincrona

Se viene richiesto un asse per il proprio canale, AXTOCHAN viene riprodotto da un'azione sincrona su un comando GET. In questo caso l'asse diventa un asse reale alla prima richiesta per il proprio canale. Alla seconda richiesta l'asse viene assegnato al programma NC in modo analogo alla richiesta GET nel programma NC. Per la richiesta GET da un'azione sincrona vedere il capitolo "Azioni sincrone di movimento".

1.17 NEWCONF: attivazione di dati macchina

Funzione

Con l'istruzione NEWCONF tutti i dati macchina della configurazione "NEW_CONFIG" vengono attivati. Questa funzione può anche essere attivata dalla superficie operativa HMI con il softkey "Attivare DM".

Quando viene eseguita la funzione NEWCONF si verifica uno stop avanzamento implicito, ossia il movimento vettoriale viene interrotto.

Programmazione

NEWCONF

Parametro

NEWCONF

Tutti i dati macchina della configurazione "NEW_CONFIG" vengono attivati.

Esecuzione estesa a tutti canali di NEWCONF dal partprogram

Se i dati macchina assiali del partprogram vengono modificati e successivamente attivati con NEWCONF, NEWCONF attiva solo i dati macchina che comportano modifiche per il canale del partprogram.

Nota

Per rendere attive tutte le modifiche, l'istruzione NEWCONF deve essere eseguita in ogni canale nel quale vengono attualmente calcolati anche gli assi o le funzioni interessate, modificate dai dati macchina.

Con NEWCONF non vengono attivati dati macchina assiali.

Per gli assi controllati dal PLC è necessario eseguire un RESET assiale.

Esempio

Fresatura: lavorare la posizione di foratura con varie tecnologie

N10 \$MA_CONTOUR_TOL[AX] = 1.0

;modificare il dato macchina

N20 NEWCONF

;attivare i dati macchina

1.18 WRITE: Scrittura di file

Funzione

Con l'istruzione WRITE si possono aggiungere dei dati (per es. risultato di misura nel caso di cicli di misura) alla fine del file specificato.

I file creati possono essere

- letti, modificati e cancellati da tutti gli utenti, ed
- essere scritti nel partprogram attualmente in esecuzione.

I blocchi vengono inseriti a fine file, quindi dopo M30.

Il livello di protezione attuale impostato deve essere uguale o maggiore al diritto WRITE del file. In caso contrario, l'accesso viene rifiutato con un messaggio di errore (error=13)

Programmazione

```
WRITE (VAR INT error, CHAR[160] filename, CHAR[200] STRING)
```

Parametri

Costruttore della macchina

Con il comando WRITE si possono salvare in un file dei blocchi del partprogram. Le dimensioni dei file di protocollo (in kbyte) vengono definite con dato macchina.

Con il DM 11420: Con LEN_PROTOCOL_FILE viene impostata la lunghezza massima dei file di protocollo in kbyte. Questa lunghezza vale per tutti i file che sono stati creati con il comando WRITE.

Se il file raggiunge la lunghezza indicata, viene emesso un messaggio di errore e il dato STRING non viene memorizzato. Se la memoria è sufficiente, si può quindi creare un nuovo file.

WRITE	Aggiunge dati al termine del file specificato
error	Variabile di errore restituita 0: nessun errore 1: percorso non consentito 2: percorso non trovato 3: file non trovato 4: tipo di file errato 10: file pieno 11: file già in uso 12: risorse insufficienti 13: mancano i diritti di accesso 20: altro errore
filename	Nome del file in cui deve essere scritta la stringa. Se filename contiene caratteri vuoti o di controllo (caratteri con codice decimale ASCII <=32), l'istruzione WRITE viene terminata con il codice di errore 1 "Percorso non consentito". Il nome del file si può indicare con il percorso e l'identificatore del file. I nomi di percorso devono essere assoluti, ossia iniziare con "/". Se il nome del file non contiene un identificatore del dominio (_N_), questo viene aggiunto di conseguenza. Se non è indicato nessun identificatore ID (_MPF o _SPF) viene automaticamente inserito l'identificatore _MPF. Se non viene specificato il percorso, il file viene salvato nella directory corrente (=directory del programma selezionato). La lunghezza massima del nome file è di 32 byte, quella del percorso di 128 byte. Esempio: PROTFILE _N_PROTFILE _N_PROTFILE_MPF /_N_MPF_DIR/_N_PROTFILE_MPF/
STRING	Testo da scrivere. Internamente viene inoltre aggiunto LF quindi il testo diventa più lungo di 1 carattere.

Nota

Un file che deve essere descritto tramite il comando WRITE viene creato ex novo se non esiste già nel controllo numerico.

Se sul disco rigido esiste già un file con lo stesso nome, questo verrà sovrascritto (nel controllo numerico) alla chiusura del file.

Rimedio: modificare il nome nel controllo numerico tramite il softkey "Proprietà" nel settore operativo Servizi.

Esempio

```
N10 DEF INT ERROR
N20 WRITE(ERROR, "TEST1", "PROTOCOLLO DEL 7.2.97") ;scrive il testo PROTOCOLLO DEL 7.2.97 nel
                                                    ;file TEST1
N30 IF ERROR
N40 MSG ("errore nel comando WRITE:"
        <<ERROR)
N50 M0
N60 ENDIF
...
WRITE(ERROR, ;Impostazione assoluta del percorso
"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF",
"PROTOCOLLO DEL 7.2.97")
```

1.19 DELETE: cancella il file

Funzione

Con il comando DELETE si possono cancellare tutti i file, indipendentemente dal fatto che siano stati creati con il comando WRITE o meno. Anche i file creati da un livello di accesso superiore possono essere cancellati tramite l'istruzione DELETE.

Programmazione

```
DELETE(VAR INT error, CHAR[160] filename)
```

Parametri

DELETE	Cancellazione del file indicato.
error	Variabile di errore restituita 0: nessun errore 1: percorso non consentito 2: percorso non trovato 3: file non trovato 4: tipo di file errato 11: file già in uso 12: risorse insufficienti 20: altro errore
filename	Nome del file che deve essere cancellato Il nome del file si può indicare con il percorso e l'identificatore del file. I nomi di percorso devono essere assoluti, ossia iniziare con "/". Se il nome del file non contiene un identificatore del dominio (_N_), questo viene aggiunto di conseguenza. L'identificatore del file ("_" più tre caratteri, ad es. _SPF) è facoltativo. Se non vi è alcun identificatore, il nome file viene assegnato automaticamente con _MPF. Se non viene specificato il percorso, il file viene salvato nella directory corrente (=directory del programma selezionato). La lunghezza massima del nome file è di 32 byte, quella del percorso di 128 byte. Esempio: PROTFILE _N_PROTFILE _N_PROTFILE_MPF /_N_MPF_DIR/_N_PROTFILE_MPF/

Esempio

```

N10 DEF INT ERROR
N15 STOPRE                                     ;stop preelaborazione
N20 DELETE (ERROR,                             ;cancella il file TEST1 nel
    "/_N_SPF_DIR/_N_TEST1_SPF")               ;ramo del sottoprogramma
N30 IF ERROR
N40 MSG ("errore nel comando DELETE:"
    <<ERROR)
N50 M0
N60 ENDIF

```

1.20 READ: lettura di righe nel file

Funzione

Il comando READ legge una o più righe nel file specificato e deposita le informazioni lette in un campo del tipo STRING. Ogni riga letta copre in questo campo un elemento di campo.

Il livello di protezione attuale impostato deve essere uguale o maggiore al diritto READ del file. In caso contrario, l'accesso viene rifiutato con un messaggio di errore (error=13).

Programmazione

```
READ(VAR INT error, STRING[160] file, INT line, INT number,  
VAR STRING[255] result[])
```

Parametri

READ	Legge una o più righe nel file specificato e salva le informazioni in un elemento di un campo. L'informazione è del tipo STRING.
error	Variabile di errore per conferma (parametri di riferimento Call-By, tipo INT) 0: nessun errore 1: percorso non consentito 2: percorso non trovato 3: file non trovato 4: tipo di file errato 13: diritti di accesso insufficienti 21: riga non esistente (parametro "line" o "number" maggiore del numero di righe del file) 22: lunghezza del campo della variabile risultato "result" troppo piccolo 23: intervallo delle righe troppo grande (le dimensioni del parametro "number" sono state selezionate in modo tale che la lettura avviene oltre la fine del file)
file	Nome/percorso del file da leggere (parametro Call-By-Value del tipo STRING con lunghezza max. di 160 byte). Il file deve trovarsi nella memoria utente dell'NCK (file system passivo). Al nome file può essere preposto l'identificatore del dominio _N_. Se l'identificatore del dominio manca, viene sostituito in modo corrispondente. L'identificatore del file ("_" più tre caratteri, ad es. _SPF) è facoltativo Se non vi è alcun identificatore, al nome del file viene assegnato automaticamente _MPF Se in "file" non viene specificato il percorso, il file viene cercato nella directory corrente (=directory del programma selezionato). Le indicazioni del percorso in "file" devono iniziare con "/" (indicazione del percorso assoluta).

line	Indicazione della posizione del campo di righe da leggere (parametro Call-By-Value del tipo INT). 0: Viene letto prima della fine del file un numero di righe pari al valore indicato nel parametro "number"
number	Da 1 a n: Numero della prima riga da leggere Numero delle righe da leggere (parametro Call-By-Value del tipo INT)
result	Campo del tipo STRING in cui viene memorizzato il testo letto (parametro Call-By-Reference con una lunghezza di 255)

Se nel parametro "number" sono specificate meno righe rispetto a quanto risulta dalla lunghezza di campo di "result", gli altri elementi di campo non vengono modificati.

La conclusione di una riga tramite il carattere di controllo "LF" (Line Feed) o "CR LF" (Carriage Return Line Feed) non viene memorizzata nella variabile di destinazione "result". Le righe lette vengono troncate se la riga è maggiore della lunghezza della stringa della variabile di destinazione "result". Non viene emesso alcun messaggio di errore.

Nota

I file binari non possono essere letti.

Viene emesso l'errore error=4: tipo di file errato. I seguenti tipi di file non possono essere letti: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Esempi

```

N10 DEF INT ERROR ; variabile di errore
N20 STRING[255] RESULT[5] ; variabile risultato
...
N30 READ(ERROR, "TESTFILE", 1, 5, RESULT) ;nome file senza identificatore di dominio
;e di file
...
N30 READ(ERROR, "TESTFILE_MPF", 1, 5, RESULT) ;nome file senza identificatore di dominio
;e con identificatore di file
...
N30 READ(ERROR, "_N_TESTFILE_MPF", 1, 5, RESULT) ;nome file con identificatore di dominio
;e senza identificatore di file
...
N30 READ(ERROR, "/_N_CST_DIR/_N_TESTFILE_MPF" , ;nome file con identificatore di dominio,
1, 5 RESULT) ;identificatore di file e indicazione del
;percorso
...
N40 IF ERROR <> 0 ;valutazione dell'errore
N50 MSG("FEHLER "<<ERROR<<" BEI READ-BEFEHL")
N60 M0
N70 ENDIF
...

```

1.21 ISFILE: File esistente nella memoria utente dell'NCK

Funzione

Con l'istruzione ISFILE si verifica se un file esiste nella memoria utente dell'NCK (file system passivo). Come risultato viene fornito TRUE (file esistente) oppure FALSE (file non esistente).

Programmazione

```
result=ISFILE (STRING [160] file)
```

Parametri

ISFILE	Verifica se un file è presente nella memoria utente dell'NCK.
file	Nome/percorso del file da leggere (parametro Call-By-Value del tipo STRING con lunghezza max. di 160 byte). Il file deve trovarsi nella memoria utente dell'NCK (file system passivo). Al nome file può essere preposto l'identificatore del dominio _N_. Se l'identificatore del dominio manca, viene sostituito in modo corrispondente. L'identificatore del file ("_" più tre caratteri, ad es. _SPF) è facoltativo. Se non vi è alcun identificatore, il nome file viene assegnato automaticamente con _MPF. Se in "file" non viene specificato il percorso, il file viene cercato nella directory corrente (=directory del programma selezionato). Le indicazioni del percorso in "file" devono iniziare con "/" (indicazione del percorso assoluta).
result	Variabile per la registrazione del risultato del tipo BOOL (TRUE o FALSE)

Esempio

```
N10 DEF BOOL RESULT
N20 RESULT=ISFILE ("TESTFILE")
N30 IF (RESULT==FALSE)
N40 MSG ("FILE NON TROVATO")
N50 M0
N60 ENDIF
...
oppure:
N30 IF (NOT ISFILE ("TESTFILE"))
N40 MSG ("FILE NON TROVATO")
N50 M0
N60 ENDIF
...
```

1.22 FILEDATE/TIME/SIZE/STAT/INFO: informazioni file

Funzione

Tramite i comandi di linguaggio FILEDATE, FILETIME, FILESIZE, FILESTAT e FILEINFO è possibile leggere certe informazioni file quali data, ora, dimensione attuale file, stato file o l'insieme di queste informazioni dalla memoria utente dell'NCK (file system passivo).

Il livello di protezione attualmente impostato deve essere uguale al o maggiore del diritto SHOW della directory sovraordinata. In caso contrario, l'accesso viene rifiutato con un messaggio di errore (error=13).

Utilizzo:

Approntamento di nuove informazioni file se un file per l'utente è stato modificato e deve essere ad es. ricalcolato.

Programmazione

```
FILExxxx(VAR INT error, STRING[160] file, VAR {STRING[yy]INT}result)
```

Parametri

FILEDATE	fornisce la data dell'ultimo accesso in scrittura al file
FILETIME	fornisce l'ora dell'ultimo accesso in scrittura al file
FILESIZE	fornisce le dimensioni attuali del file
FILESTAT	fornisce lo stato del file, quali i diritti di lettura, scrittura ed esecuzione
FILEINFO	fornisce l'insieme delle informazioni da una voce di directory
error	Rappresentazione della variabile di errore restituita
	0: nessun errore
	1: percorso non consentito
	2: percorso non trovato
	3: file non trovato
	13: diritti di accesso insufficienti
	22: lunghezza del campo della variabile risultato "result" troppo piccolo

file	Nome/percorso del file da leggere (parametro Call-By-Value del tipo STRING con lunghezza max. di 160 byte). Il file deve trovarsi nella memoria utente dell'NCK (file system passivo). Al nome file può essere preposto l'identificatore del dominio _N_. Se l'identificatore del dominio manca, viene sostituito in modo corrispondente. L'identificatore del file ("_" più tre caratteri, ad es. _SPF) è facoltativo. Se non vi è alcun identificatore, il nome file viene assegnato automaticamente con _MPF. Se in "file" non viene specificato il percorso, il file viene cercato nella directory corrente (=directory del programma selezionato). Le indicazioni del percorso in "file" devono iniziare con "/" (indicazione di percorso assoluta).
result	Variabile con il risultato in cui viene salvata l'informazione file (parametro Call-By-Reference) del tipo STRING per: FILEDATE, la lunghezza deve essere di 8 caratteri, il formato è "gg.mm.aa" FILETIME, la lunghezza deve essere di 8 caratteri, il formato è "hh:mm:ss" FILESTAT, la lunghezza deve essere di 5 caratteri, il formato è "rwxsd" FILEINFO, la lunghezza deve essere di 32 caratteri, il formato è "rwxsd nnnnnnnn gg.mm.aa hh:mm:ss" (Parametro Call-By-Reference) del tipo INT per: FILESIZE, la dimensione file viene emessa in byte "rwxsd" (Read, write, eXecute, Show, Delete)

Esempi

N10 DEF INT ERROR	;variabile di errore
N20 STRING[32] RESULT	;variabile risultato
...	
N30 FILEINFO(ERROR, "TESTFILE", RESULT)	;nome del file senza identificatore di dominio e ;di file
...	
N30 FILEINFO(ERROR, "TESTFILE_MPF", RESULT)	;nome del file senza identificatore del dominio e ;con identificatore del file
...	
N30 FILEINFO(ERROR, "_N_TESTFILE_MPF", RESULT)	;nome del file con identificatore di dominio e di ;file
...	
N30 FILEINFO (ERROR, "/_N_MPF_DIR/_N_TESTFILE_MPF", RESULT)	;nome del file con identificatore del dominio, ;identificatore del file ;e indicazione del percorso
...	
N40 IF ERROR <>0	;valutazione dell'errore
N50 MSG("ERRORE "<<ERROR<<" CON COMANDO FILEINFO")	
N60 M0	
N70 ENDIF	
...	
Fornisce la variabile di risultato RESULT: "77777 12345678 26.05.00 13:51:30"	

1.23 CHECKSUM: Creazione del checksum per un campo

Funzione

Il comando CHECKSUM produce il checksum per un campo.

Utilizzo:

Verifica se il profilo di ingresso è stato modificato durante l'asportazione.

Programmazione

```
error=CHECKSUM(VAR STRING[16] chksum, STRING[32]array, INT first,
INT last)
```

Parametri

CHECKSUM	Formare la somma di controllo su un campo.
error	Rappresentazione della variabile di errore restituita 0: nessun errore 1: simbolo non trovato 2: nessun campo 3: indice 1 troppo grande 4: indice 2 troppo grande 5: tipo di dato errato 10: overflow del checksum
chksum	Checksum per un campo come STRING (parametro Call-By-Reference del tipo STRING con una lunghezza fissa di 16 caratteri). Il checksum viene rappresentato come stringa di caratteri di 16 cifre esadecimali. Non vengono tuttavia specificati i caratteri di formato. Esempio: "A6FC3404E534047C"
array	Nome del campo per il quale deve essere formato il checksum (parametro Call-By-Value del tipo STRING) con una lunghezza max. di 32 caratteri. Campi ammessi: campi mono o bidimensionali dei tipi BOOL, CHAR, INT, REAL, STRING I campi dei dati macchina non sono ammessi.
first	Numero della colonna iniziale (facoltativo)
last	Numero della colonna finale (facoltativo)

Nota

I parametri first e last sono facoltativi. Se non vengono specificati indici delle colonne, il checksum viene formato considerando il campo completo.

Il risultato del checksum è sempre univoco. In caso di variazioni di un elemento di campo, si ha anche un'altra stringa di risultato.

Esempio

```
N10 DEF INT ERROR
N20 DEF STRING[16] MY_CHECKSUM
N30 DEF INT MY_VAR[4,4]
N40 MY_VAR=...
N50 ERROR=CHECKSUM (CHECKSUM;"MY_VAR", 0, 2)
...
fornisce in MY_CHECKSUM il valore "A6FC3404E534047C"
```

1.24 ROUNDUP: arrotondamento

Funzione

La funzione ROUNDUP fornisce per

- **valori di ingresso positivi**
il successivo numero intero superiore
- **valori di ingresso negativi**
il successivo numero intero inferiore

Se il valore di ingresso è del tipo Integer (un numero intero), il valore viene restituito invariato.

Programmazione

ROUNDUP(variabile Real)

Parametri

ROUNDUP	Arrotonda al successivo numero intero superiore con segno corretto.
Variabili	Valore di ingresso del tipo Real
Real	Tipo di variabili per numeri frazionari con punto decimale

ROUNDUP nel partprogram NC

```
N10 X = ROUNDUP(3.5) Y = ROUNDUP(R2+2)
N15 R2 = ROUNDUP($AA_IM[Y])
N20 WHEN X = = 100 DO Y = ROUNDUP($AA_IM[X])
```

Esempi

ROUNDUP (3 . 1) fornisce 4.0
ROUNDUP (3 . 6) fornisce 4.0
ROUNDUP (-3 . 1) fornisce -3.0
ROUNDUP (-3 . 6) fornisce -3.0
ROUNDUP (3 . 0) fornisce 3.0
ROUNDUP (3) fornisce 3.0

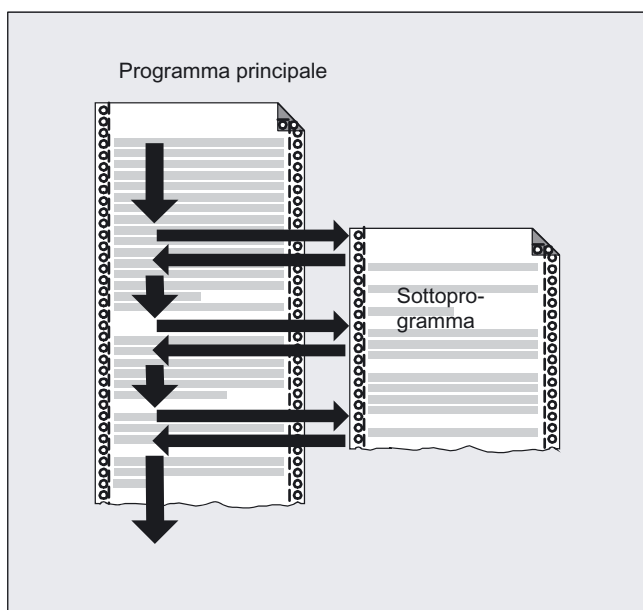
Tecnica dei sottoprogrammi, tecnica delle macro

2.1 Utilizzo dei sottoprogrammi

Funzione

Un sottoprogramma ha fondamentalmente la struttura di un partprogram. Esso si compone di blocchi NC con istruzioni di movimento e di commutazione.

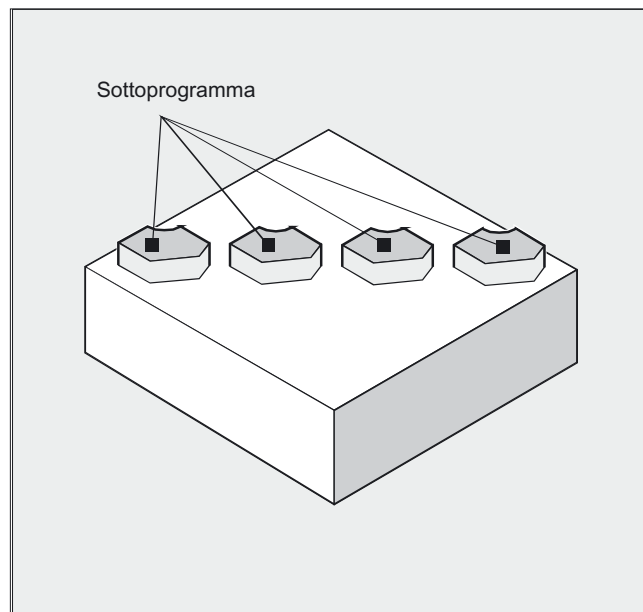
In sostanza tra un programma principale e un sottoprogramma non vi è alcuna differenza. Il sottoprogramma contiene o delle sequenze di lavoro o delle sezioni di lavoro da ripetere più volte.



Applicazione

Sequenze di lavoro che si ripetono vengono programmate una sola volta in un sottoprogramma. Ciò riguarda ad esempio determinate forme di contorno che ritornano continuamente oppure anche cicli di lavoro.

Questo sottoprogramma può essere poi richiamato ed eseguito in qualunque programma principale.



Struttura di un sottoprogramma

La struttura di un sottoprogramma è identica a quella del programma principale.

In un sottoprogramma è inoltre possibile programmare un'intestazione con la definizione dei parametri.

I sottoprogrammi vengono concepiti con la fine programma M17. Ciò permette di ritornare al livello di programma dal quale è avvenuto il richiamo. Tramite il dato macchina è possibile sopprimere la fine programma M17 (ad es. per un vantaggio a livello di tempi di elaborazione).

Fine sottoprogramma con RET

In sostituzione del salto a ritroso con M17 è possibile utilizzare nel sottoprogramma anche l'istruzione RET. RET richiede un blocco a sé stante.

L'istruzione RET deve essere utilizzata se un funzionamento continuo G64 (G641, G642, G643) **non** deve essere interrotto dallo spostamento a ritroso. La condizione è che il sottoprogramma **non contenga** l'attributo SAVE.

Se M17 viene programmata in un blocco a sé stante, G64 viene interrotto, con creazione di un arresto preciso. Rimedio: non scrivere M17 soltanto in un blocco UP, ma ad as. anche con un determinato percorso:

```
G1 X=YY M17
```

Nel dato macchina devono essere impostati: "nessun M17 da PLC".

Nome del sottoprogramma

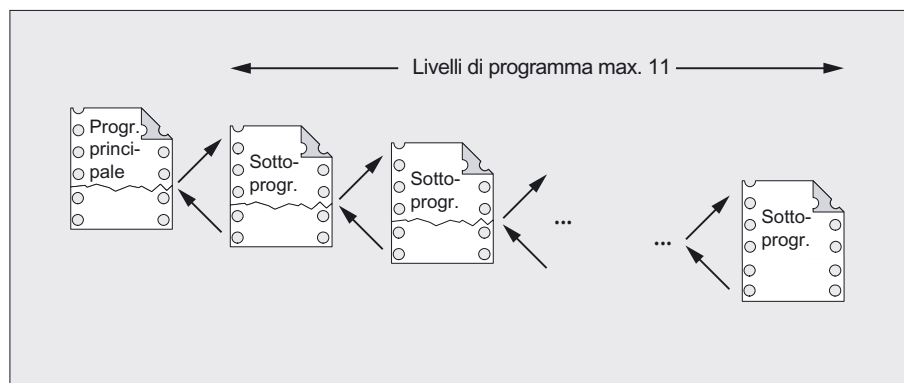
Per consentire la selezione di un determinato sottoprogramma tra molti, ogni programma è dotato di un nome. Il nome può essere scelto liberamente al momento della generazione del programma, tenendo conto delle seguenti limitazioni:

- i primi due caratteri devono essere lettere dell'alfabeto
- gli altri possono essere lettere, cifre o caratteri di sottolineatura
- utilizzare max. 31 caratteri
- non utilizzare caratteri di separazione (vedere capitolo "Elementi del linguaggio di programmazione")

Valgono le stesse regole dei nomi dei programmi principali.

Annidamento dei sottoprogrammi

I sottoprogrammi possono essere richiamati dal programma principale e da altri sottoprogrammi. Complessivamente per questo tipo di richiamo sono disponibili 12 livelli di programma, compreso quello di programmazione principale. Questo significa che in un programma principale si possono annidare 11 richiami di sottoprogrammi.



Limitazioni per sottoprogrammi in routine di interrupt ed esecuzione di cicli

I sottoprogrammi possono essere richiamati anche in routine di interrupt. Lavorando con routine di interrupt è bene mantenere liberi 4 livelli, ovvero annidare non più di 7 richiami di sottoprogramma.

Per i cicli di lavorazione e di misura SIEMENS sono necessari 3 livelli. Se si intende richiamare un ciclo da un sottoprogramma occorre farlo al massimo nel livello 5 (qualora 4 livelli rimangano liberi per routine di interrupt).

2.2 Sottoprogrammi con meccanismo SAVE

Funzione

Nell'istruzione di definizione con PROC indicare anche l'istruzione SAVE.

Programmazione

Nel sottoprogramma

```
PROC Nome sottoprogramma SAVE
```

Tramite l'attributo SAVE, le funzioni G modali, al termine dei sottoprogrammi vengono impostate sul valore che avevano all'avvio dei sottoprogrammi. Ne deriva quindi una variazione di

gruppo 8 funzioni G (spostamento origine impostabile)

oppure

gruppo 52 funzioni G (rotazione frame di un pezzo in rotazione)

oppure

gruppo 53 funzioni G (rotazione frame nella direzione dell'utensile),

e verranno quindi ricreati i relativi frame.

- Il frame base attivo non viene modificato quando si ritorna al salto del sottoprogramma
- Lo spostamento origine programmabile viene ripristinato

Parametri

Il comportamento dello spostamento origine programmabile e quello del frame base può essere modificato tramite il dato macchina DM 10617: FRAME_SAVE_MASK. Vedere

Bibliografia,

/FB1/ Manuale delle funzioni di base; Assi, sistemi di coordinate, frame (K2), capitolo "Uscita da sottoprogramma con SAVE".

Esempio

Definizione di un sottoprogramma

```
PROC KONTUR (REAL WERT1) SAVE
```

```
N10 G91 ...
```

```
N100 M17
```

Programma principale

```
%123
```

```
N10 G0 X... Y... G90
```

```
N20...
```

```
N50 KONTUR (12.4)
```

```
N60 X... Y...
```

Nel sottoprogramma KONTUR è attiva la funzione G91 quota incrementale. Dopo il ritorno al programma principale è di nuovo attiva la quota assoluta, poiché le funzioni modali del programma principale erano state memorizzate con SAVE.

2.3 Sottoprogrammi con assegnazione di parametri (PROC, VAR)

Funzione

Inizio programma, PROC

Un sottoprogramma che nel corso dello svolgimento deve assumere i parametri del programma che lo ha richiamato, deve essere contrassegnato con la parola chiave PROC.

Fine sottoprogramma M17, RET

Con l'istruzione M17 si indica allo stesso tempo sia la fine di un sottoprogramma che l'istruzione di ritorno al programma principale, dal quale il sottoprogramma era stato richiamato. In alternativa a M17: la parola chiave RET indica la fine del sottoprogramma senza tuttavia interrompere il funzionamento continuo e senza emissione di funzioni al PLC.

Programmazione

I parametri rilevanti per l'assegnazione devono essere contrassegnati all'inizio del sottoprogramma con il tipo ed il nome.

Assegnazione di parametri Call-by-value

```
PROC PROGRAMMNAME (VARIABLENTYP1 VARIABLE1, VARIABLENTYP2 VARIABLE2, ...)
```

Esempio:

```
PROC PROFILO (REAL LUNGHEZZA, REAL LARGHEZZA)
```

Assegnazione di parametri Call-by-reference, identificazione con parola chiave VAR

```
PROC PROGRAMMNAME (VAR VARIABLENTYP1 VARIABLE1, VAR VARIABLENTYP2 ..., )
```

Esempio:

```
PROC PROFILO (VAR REAL LUNGHEZZA, VAR REAL LARGHEZZA)
```

Trasmissione di campi Call-by-reference, identificazione con parola chiave VAR

```
PROC NOME PROGRAMMA (VAR TIPO VARIABILE 1, NOME CAMPO 1 [dimensioni  
campo],  
VAR TIPO VARIABILE 2 NOME CAMPO 2 [dimensioni campo],  
VAR TIPO VARIABILE 3 NOME CAMPO 3 [dimensioni campo1, dimensioni  
campo2],  
VAR TIPO VARIABILE 4 NOME CAMPO 4 [ ],  
VAR TIPO VARIABILE 5 NOME CAMPO 5 [, dimensioni campo])
```

Esempio:

```
PROC PALETTE (VAR INT CAMPO [, 10])
```

Parametri

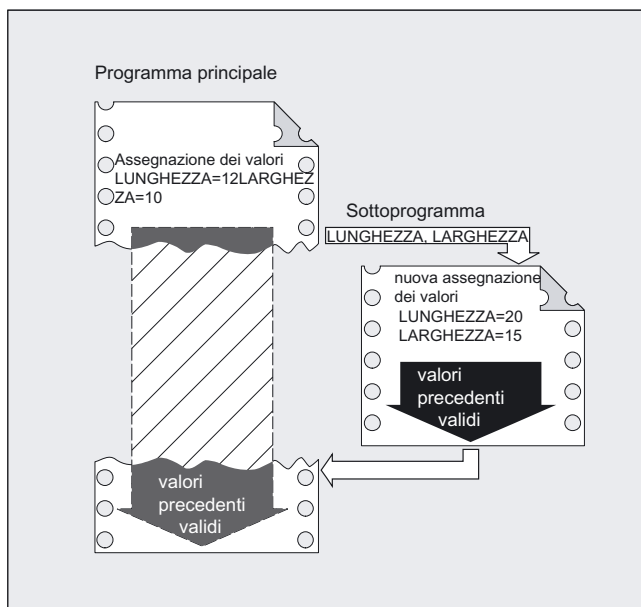
PROC	Prima istruzione di un programma
PROGRAMMNAME	Nome del sottoprogramma che deve assumere i valori rilevanti dei parametri
VARIABLENTYP	Tipi di variabili con indicazione dei valori delle variabili.
VARIABLE	Sono possibili diverse indicazioni.
VAR	Parola chiave per il tipo di assegnazione dei parametri
FELDDNAME	Elementi di un campo con elenco dei valori nella definizione dei campi
Dimensione campo 1	per un campo monodimensionale
Dimensione campo 2	per un campo bidimensionale

Nota

L'istruzione di definizione con PROC deve essere programmata in un blocco NC a sè stante. Per l'assegnazione dei parametri possono essere scelti al massimo 127 parametri.

Esempio di assegnazione di parametri tra programma principale e sottoprogramma

```
N10 DEF REAL LUNGHEZZA, LARGHEZZA
N20 LUNGHEZZA=12 LARGHEZZA=10
N30 CORNICE (LUNGHEZZA, LUNGHEZZA)
```



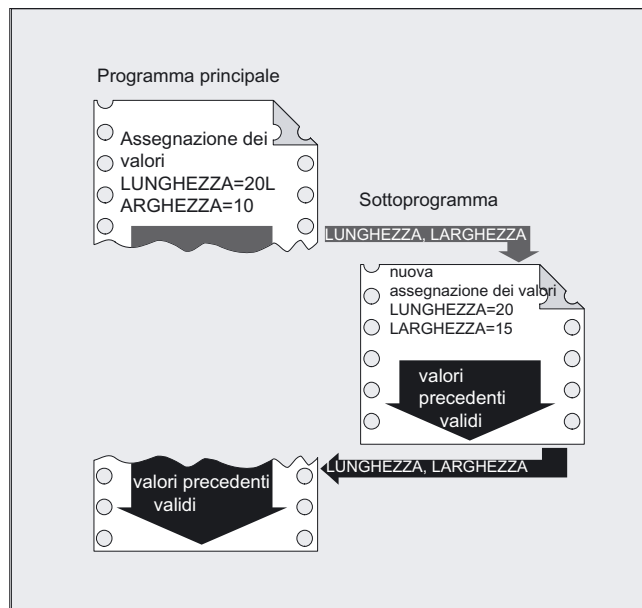
Al momento del richiamo del sottoprogramma i valori assegnati a N20 nel programma principale vengono trasferiti a N30. La trasmissione dei parametri avviene nella sequenza indicata.

Non è necessario che i nomi dei parametri nel programma principale e nel sottoprogramma siano uguali.

Seconda possibilità di assegnazione dei parametri:

- I valori vengono solo trasferiti (Call-by-value)

Se nel corso dello svolgimento di un sottoprogramma i parametri assegnati vengono modificati, ciò non ha influenza sul programma principale. In quest'ultimo i parametri rimangono invariati (vedi figura).



- **Assegnazione di parametri con scambio dati (Call-by-reference)**

Ad ogni modifica dei parametri nel sottoprogramma segue necessariamente la modifica dei parametri del programma principale (vedi figura).

Esempio di lunghezza campi variabile

<pre>%_N_DIMA_DI_FORATURA_MPF DEF REAL TABELLA[100,2] EXTERN DIMA_DI_FORATURA (VAR REAL[,2],INT) TABELLA[0,0]=-17.5 ... TABELLA[99,1]=45 DIMA_DI_FORATURA (TABELLA,100) M30</pre>	Programma principale ;Definizione della tabella delle posizioni ;Definizione della posizione ;Richiamo del sottoprogramma

Generazione di una dima di foratura mediante una tabella posizioni assegnata di lunghezza variabile

<pre>%_N_DIMA_DI_FORATURA_SPF PROC DIMA_DI_FORATURA (VAR REAL CAMPO[,2],-> -> INT NUMERO) DEF INT CONTATORE STEP: G1 X=CAMPO[CONTATORE,0]-> -> Y=CAMPO[CONTATORE,1] F100 Z=IC(-5) Z=IC(5) CONTATORE=CONTATORE+1</pre>	Sottoprogramma ;Assegnazione di parametri ;Fasi di lavorazione

```
IF CONTATORE<NUMERO GOTOB STEP  
RET
```

;Fine sottoprogramma

Funzionamento continuo senza interruzioni

La condizione essenziale perchè il funzionamento continuo non venga interrotto è la seguente:

il sottoprogramma **non** può avere un attributo SAVE. Per ulteriori informazioni sulla modalità SAVE vedere il capitolo Sottoprogrammi con modalità SAVE.

RET deve essere programmata in un blocco a sè stante.

```
PROC PROFILO  
N10...  
...  
N100 M17
```

Esempio di assegnazione di parametri tra programma principale e sottoprogramma

Se in un programma principale si lavora con i parametri, è possibile utilizzare anche nel sottoprogramma i corrispondenti valori calcolati o assegnati. A questo scopo, al momento del richiamo del sottoprogramma, i valori dei **parametri attuali** del programma principale vengono assegnati ai **parametri formali** del sottoprogramma e quindi elaborati nel corso dell'esecuzione del sottoprogramma.

Definizione del campo

Per la definizione di parametri formali vale quanto segue: Nel caso di campi bidimensionali non è necessario impostare il numero di campi della prima dimensione, ma occorre scrivere la virgola.

Esempio:

```
VAR REAL CAMPO[,5]
```

Nel caso di lunghezze di campo indefinite i sottoprogrammi possono elaborare campi di lunghezza variabile. Al momento della definizione delle variabili occorre però stabilire quanti elementi ne dovranno far parte. Per chiarimenti relativi alla definizione dei campi vedere "Programmazione NC flessibile" nel capitolo omonimo.

2.4 Richiamo di sottoprogrammi (L oppure EXTERN)

Funzione

Richiamo di sottoprogrammi senza trasferimento di parametri

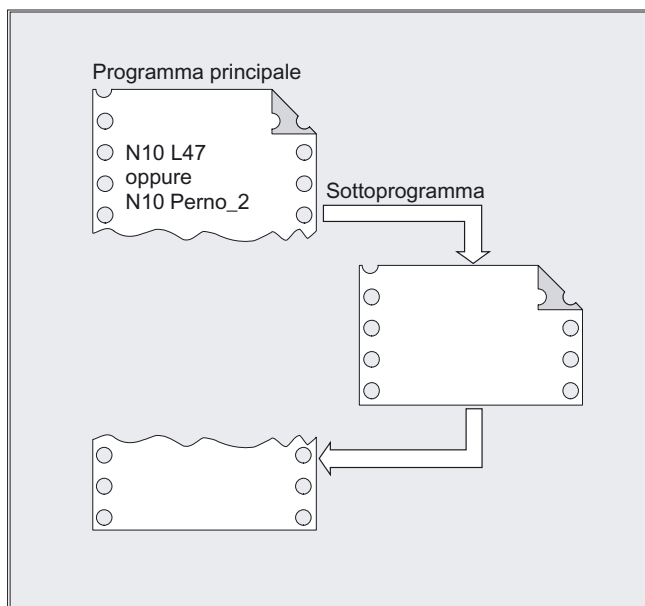
Nel programma principale il richiamo del sottoprogramma avviene

- con l'indirizzo L seguito dal numero del sottoprogramma oppure
- con l'indicazione del nome del sottoprogramma.

Esempio:

N10 L47 oppure

N10 PERNO_2



Programmazione

Sottoprogramma con trasferimento di parametri, descrizione con EXTERN

EXTERN

I sottoprogrammi con trasferimento di parametri devono essere contrassegnati con `EXTERN` prima del richiamo nel programma principale, ad es. all'inizio del programma.

Vengono impostati il nome del sottoprogramma ed i tipi delle variabili nella sequenza dell'assegnazione, vedere esempio.

Sottoprogrammi con assegnazione di parametri

Nel programma principale il sottoprogramma va richiamato impostando il nome del sottoprogramma e assegnando i parametri. Nell'assegnazione di parametri si possono trasferire variabili oppure direttamente valori (non per parametri VAR), vedere esempio.

Parametri

Indirizzo L	Numero sottoprogramma. Per il valore sono possibili 7 cifre decimali (solo numeri interi). Attenzione: per l'assegnazione del nome, gli zero in prima posizione sono significativi (ad. L123, L0123 e L00123 sono tre sottoprogrammi diversi).
EXTERN	Rendere noto un sottoprogramma con assegnazione dei parametri. EXTERN deve essere impostato solo se il sottoprogramma si trova nella directory pezzo o nella directory globale dei sottoprogrammi. I cicli non devono essere definiti con EXTERN .

Assegnazione incompleta di parametri

Al momento del richiamo di un sottoprogramma si possono anche omettere dei valori o parametri prescritti. In questo caso al parametro corrispondente nel sottoprogramma viene assegnato il valore **zero**.

Per quanto riguarda l'impostazione della sequenza, occorre scrivere la virgola. Se invece i parametri si trovano alla fine di una sequenza è possibile ometterla.



Cautela

Parametri attuali del tipo AXIS non possono essere omessi. Parametri VAR devono essere assegnati completamente.

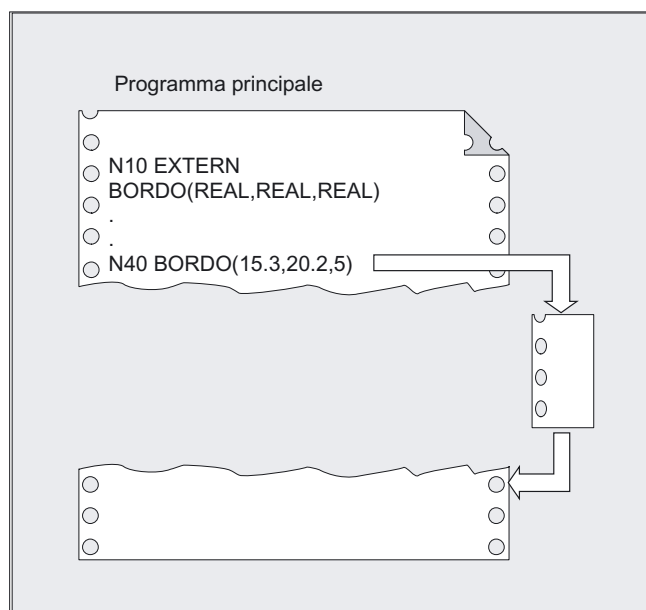
Esempio di sottoprogramma con assegnazione di parametri, descrizione con EXTERN

```
N10 EXTERN CORNICE (REAL, REAL, REAL)
```

...

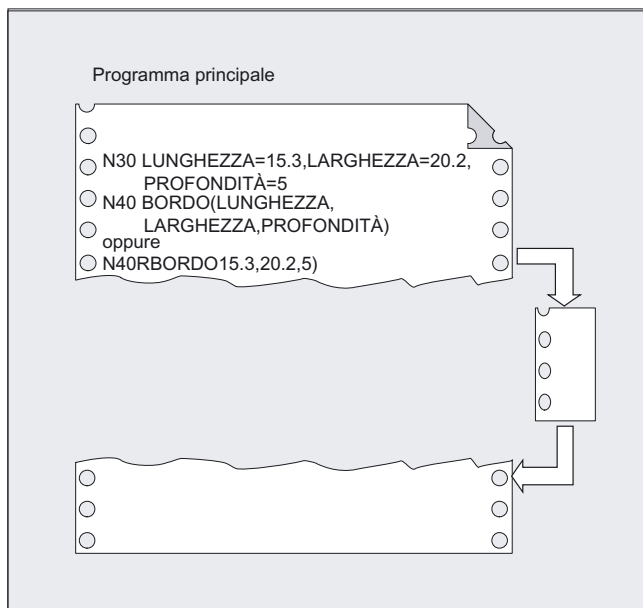
```
N40 CORNICE (15.3, 20.2, 5)
```

N10 Indicazione del sottoprogramma, N40 richiamo del sottoprogramma con assegnazione dei parametri.



Esempi odi richiamo di sottoprogrammi con assegnazione di parametri

```
N10 DEF REAL LUNGHEZZA, LARGHEZZA, PROFONDITÀ
N20...
N30 LUNGHEZZA=15.3, LARGHEZZA=20.2, PROFONDITÀ=5
N40 CORNICE (LUNGHEZZA, LARGHEZZA, PROFONDITÀ)
oppure
N40 CORNICE (15.3, 20.2, 5)
```



Esempio di sottoprogramma

```
PROC SUB1 (INT VAR1, DOUBLE VAR2)
IF $P_SUBPAR[1]==TRUE
;Il parametro VAR1 è stato programmato nel richiamo del
sottoprogramma
ELSE
;Il parametro VAR1 non è stato programmato nel richiamo del
sottoprogramma
;e come default gli è stato assegnato il valore 0 dal sistema
ENDIF
IF $P_SUBPAR[2]==TRUE
;Il parametro VAR2 è stato programmato nel richiamo del
sottoprogramma
ELSE
;Il parametro VAR2 non è stato programmato nel richiamo del
sottoprogramma
;e come default gli è stato assegnato il valore 0.0 dal sistema
ENDIF
;il parametro 3 non è definito
IF $P_SUBPAR[3]==TRUE -> allarme 17020
M17
```

Descrizione



Cautela

Definizione del sottoprogramma corrispondente al richiamo del sottoprogramma

Sia i tipi di variabile sia la sequenza dell'assegnazione devono coincidere con le definizioni che sono state stabilite nel nome del sottoprogramma sotto PROC. I nomi dei parametri nel programma principale e nel sottoprogramma possono essere diversi.

Definizione nel sottoprogramma:

```
PROC CORNICE (REAL LUNGHEZZA, REAL LARGHEZZA, REAL PROFONDITÀ)
```

Richiamo nel programma principale:

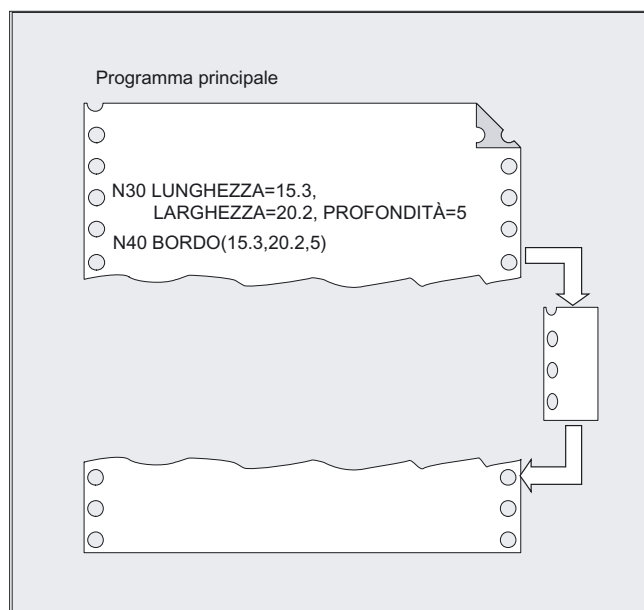
```
N30 CORNICE (LUNGHEZZA, LARGHEZZA, PROFONDITÀ)
```

Assegnazione incompleta di parametri

Ritorno all'ultimo esempio:

```
N40 CORNICE (15.3, , 5)
```

Il valore intermedio 20.2 è stato omissso.



In caso di assegnazione dei parametri incompleta, con la variabile di sistema `$P_SUBPAR` si può riconoscere se è stato effettivamente programmato il parametro di trasferimento per i sottoprogrammi. La variabile di sistema contiene come argomento (i) il numero del parametro di trasferimento.

La variabile di sistema `$P_SUBPAR` restituisce

- TRUE, se è stato programmato il parametro di trasferimento
- FALSE, se come parametro di trasferimento non è stato inserito alcun valore.

Se è stato impostato un numero di parametro non ammesso, l'elaborazione del partprogram viene interrotta con un allarme.

Richiamo di un programma principale come sottoprogramma

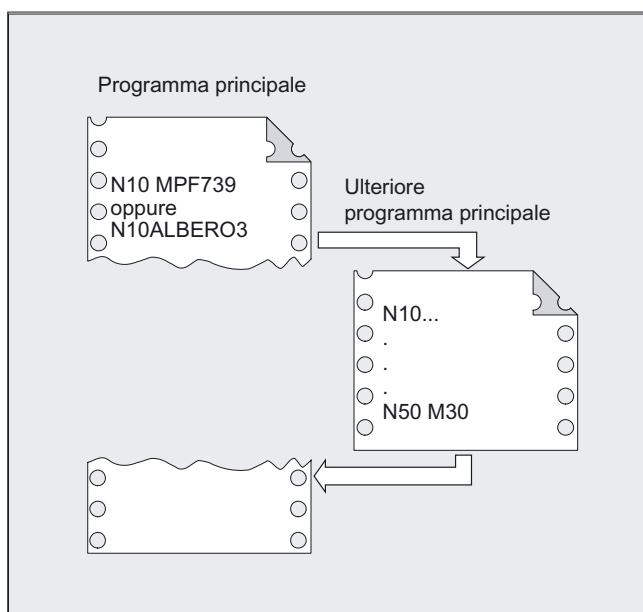
Anche un programma principale può essere richiamato come un sottoprogramma. In questo caso la fine programma M2 o M30 impostata nel programma principale viene valutata come se si trattasse di M17 (fine programma con ritorno al programma che l'ha richiamato).

Il richiamo viene programmato con indicazione del nome del programma.

Esempio:

N10 MPF739 oppure

N20 Albero3



Analogamente, un sottoprogramma può essere avviato anche come un programma principale.

Nota

Strategia di ricerca del controllo numerico:

Esiste *_MPF ?

Esiste *_SPF ?

Ne consegue: se il nome del sottoprogramma da richiamare è identico a quello del programma principale, viene richiamato nuovamente il programma principale. Per evitare questo inconveniente occorre attribuire ai sottoprogrammi e programmi principali nomi univoci.

Richiamo di sottoprogrammi con il file INI

Dal file di inizializzazione è possibile richiamare i sottoprogrammi che non richiedono trasmissione di parametri:

Esempio:

N10 MYINISUB1 ;richiamo di sottoprogramma senza parametri

2.5 Salto di ritorno parametrizzabile di un sottoprogramma (RET)

Funzione

Generalmente, da un sottoprogramma il ritorno nel programma richiamante avviene con una fine sottoprogramma RET oppure M17, l'elaborazione del partprogram prosegue con la riga di programma successiva al richiamo del sottoprogramma. Ci sono però dei casi in cui può essere necessario proseguire l'elaborazione partendo da un punto diverso:

- Proseguimento dell'esecuzione del programma dopo il richiamo dei cicli di asportazione nel modo ISO, dopo la descrizione del profilo.
- Ritorno al programma principale da un qualsiasi livello di sottoprogramma (anche dopo ASUP) nella gestione errori.
- Ritorno attraverso più livelli di programma per casi speciali nei cicli compilati e nel modo ISO.

Programmazione

```
RET (<Numero di blocco/Label>, <Blocco successivo al numero di
blocco/Label>{
<Numero dei livelli del salto di ritorno>}, <Salto all'inizio del
programma>)
```

oppure

```
RET (<Nr.blocco/Label>, < >, < >)
```

oppure salto di ritorno dal sottoprogramma su più livelli

(salto di ritorno del numero di livelli di sottoprogrammi indicati).

```
RET (, , <Numero_livelli salto di ritorno>, <Salto all'inizio del
programma>)
```

Parametri

Con il comando parametrizzabile RET si può soddisfare la necessità del proseguimento o del salto di ritorno con 4 parametri:

1. <Numero di blocco/Label>
2. <Blocco successivo al numero di blocco/Label>
3. <Numero dei livelli del salto di ritorno>
4. <Salto all'inizio del programma>

RET

<Numero di blocco/Label>

Fine sottoprogramma
(utilizzo in sostituzione di M17)

Parametri: numero di blocco o Label come
STRING (costante o variabile) del blocco
dal quale si prosegue l'esecuzione del
programma.

L'esecuzione nel programma che esegue il
richiamo viene ripresa a partire dal blocco
identificato con "Numero di blocco/Label".

<Blocco successivo al numero di blocco/Label> ,	parametro del tipo INTEGER Se il valore è maggiore di 0 , l'elaborazione prosegue con il blocco successivo il "Numero di blocco/Label". Se il valore è uguale a 0 , il salto al sottoprogramma avviene sul blocco con <Numero di blocco/Label>.
<Numerolivelli salto di ritorno> ,	Parametri del tipo INTEGER con i valori 1 ... 11 consentiti. valore = 1: il programma viene proseguito nel livello di programma attuale -1 (come RET senza parametro). Valore = 2: il programma viene proseguito nel livello di programma attuale -2 saltando quindi un livello etc.
<Salto all'inizio del programma> ,	parametro del tipo BOOL Valore 1 oppure 0. Valore = 1 quando il ritorno avviene nel programma principale e qui è attivo un modo dialetto ISO viene eseguita una diramazione all'inizio del programma.

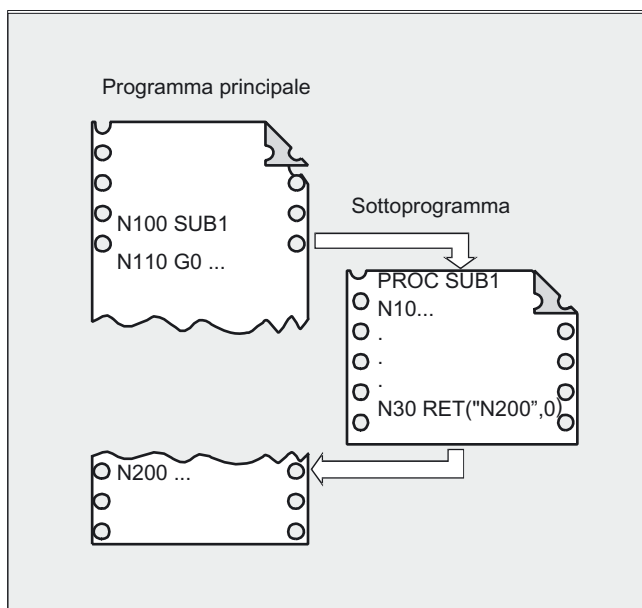
Esempio di gestione dell'errore: risincronizzazione nel programma principale dopo l'elaborazione di un ASUP

N10010 CALL "UP1"	;livello di programma 0 programma ;principale
N11000 PROC UP1	;livello di programma 1
N11010 CALL "UP2"	
N12000 PROC UP2	;livello di programma 2
N19000 PROC ASUP	;livello di programma 2 (esecuzione di un ;programma ASUP)
... RET("N10900" , , ...	;livello di programma 3
N19100 RET(N10900, \$P_STACK)	;Ritorno al sottoprogramma
N10900	;ritorno al programma principale
N10910 MCALL	;disabilitazione del sottoprogramma modale
N10920 G0 G60 G40 M5	;correzione di altre impostazioni ;modali

Descrizione

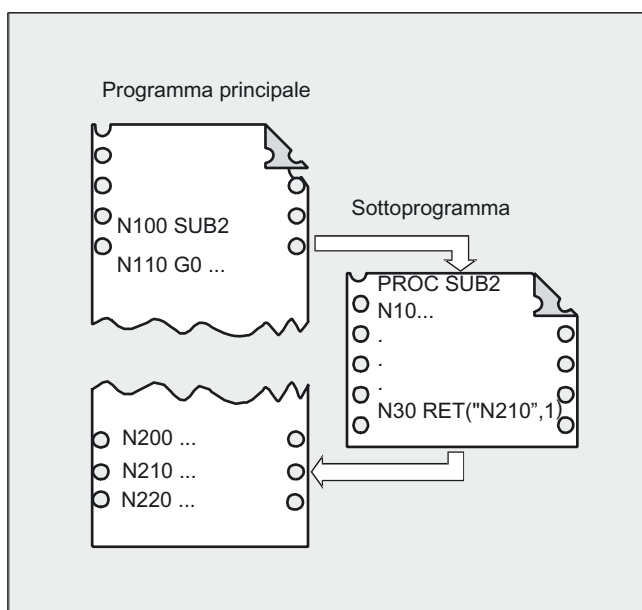
1. <Numero di blocco/Label>

Nel programma che esegue il richiamo (programma principale) l'elaborazione prosegue dal blocco con <Numero di blocco/Label>.



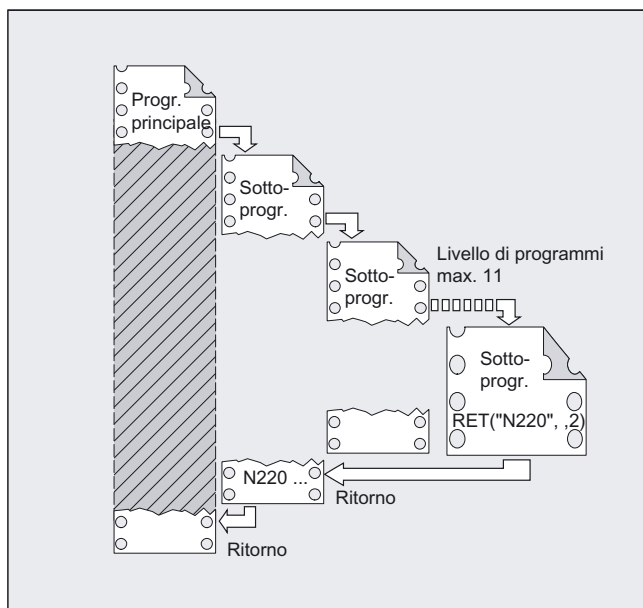
2. <Blocco successivo al numero di blocco/Label>

Il ritorno al sottoprogramma avviene in corrispondenza del blocco successivo al blocco identificato con <Numero di blocco/Label>.



3. <Numero dei livelli del salto di ritorno>

Il programma viene proseguito nel livello di programma attuale meno il <Numero dei livelli del salto di ritorno>.



Livelli per il salto di ritorno non consentiti

Se, per il numero di livelli per il salto di ritorno, è stato programmato un

- valore negativo oppure
 - un valore maggiore del livello di programma attualmente attivo –(max 11)
- , viene emesso l'allarme 14091 con il parametro 5.

Ritorno con istruzioni SAVE

Per il salto di ritorno su più livelli di programma, vengono valutate le istruzioni SAVE dei singoli livelli di programma.

Sottoprogramma modale attivo nel salto di ritorno

Se in un salto di ritorno su più livelli di programma è attivo un sottoprogramma modale e, in uno dei sottoprogrammi intermedi è stata programmata l'istruzione MCALL per il sottoprogramma modale, il sottoprogramma modale continua a restare attivo.



Cautela

L'operatore deve **sempre assicurarsi** che nel salto su più livelli di programma si prosegua con le impostazioni modali corrette. Questo può avvenire ad es. con la programmazione di un blocco principale.

2.6 Sottoprogramma con ripetizione del programma (P)

Funzione

Se un sottoprogramma deve essere elaborato più volte in successione, è possibile programmare il numero desiderato di ripetizioni del programma nel blocco contenente il richiamo del sottoprogramma sotto l'indirizzo P.

Parametri



Cautela

Richiamo del sottoprogramma con ripetizione del programma e trasferimento dei parametri

I parametri vengono trasferiti solo al momento del richiamo del programma oppure alla prima ripetizione. Per le ripetizioni successive i parametri restano invariati.

Se si desidera modificare i parametri nelle ripetizioni del programma, occorre effettuare le impostazioni corrispondenti nel sottoprogramma.

P	numero delle ripetizioni del sottoprogramma.
Campo dei valori:	1...9999 (Integer senza segno)



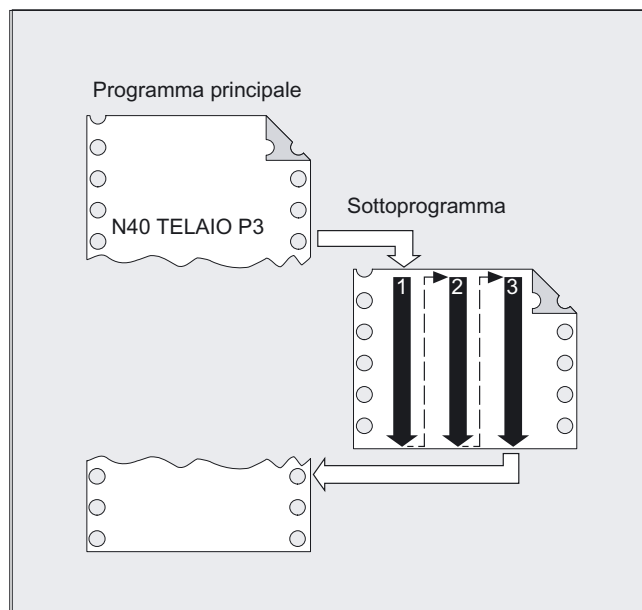
Cautela

Per ogni richiamo di sottoprogramma vale quanto segue:

il richiamo del sottoprogramma deve essere sempre programmato in un blocco NC a sé stante.

Esempio

N40 TELAIO P3



Il sottoprogramma Telaio deve essere elaborato tre volte in successione.

2.7 Sottoprogramma modale (MCALL)

Funzione

Con questa funzione il sottoprogramma viene richiamato ed eseguito automaticamente dopo ogni blocco con movimento interpolato. In questo modo è possibile automatizzare il richiamo di sottoprogrammi che devono essere eseguiti su posizioni diverse del pezzo, per esempio per la realizzazione di dime di foratura.

Disattivazione di un richiamo di sottoprogramma modale

Con MCALL senza richiamo di sottoprogramma, oppure programmando un nuovo richiamo di sottoprogramma modale per un nuovo sottoprogramma.

Parametri

MCALL	Richiamo sottoprogramma modale
Indirizzo L	Numero di sottoprogramma



Cautela

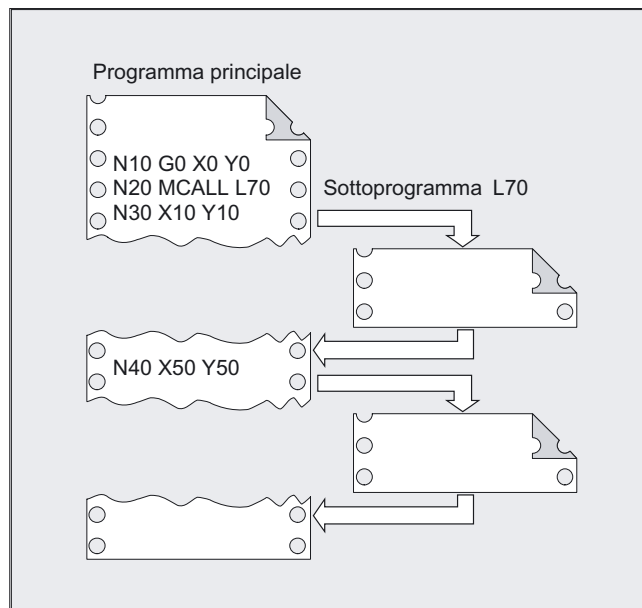
Nell'esecuzione del programma può essere attivo **un solo** richiamo MCALL alla volta. I parametri vengono trasferiti solo una volta al momento del richiamo MCALL. Il sottoprogramma modale viene richiamato nelle seguenti situazioni anche senza la programmazione di un movimento: nella programmazione degli indirizzi S e F se è attivo G0 o G1. Se è stato programmato G0/G1 da solo nel blocco o con ulteriori codici G.

Esempio

```
N10 G0 X0 Y0
N20 MCALL L70
N30 X10 Y10
N40 X50 Y50
```

Nei blocchi N30 e N40 ci si accosta alla posizione programmata e viene quindi eseguito il sottoprogramma L70.

```
N10 G0 X0 Y0
N20 MCALL L70
N30 L80
```



In questo esempio, i blocchi NC seguenti e la programmazione degli assi di contornitura, si trovano nel sottoprogramma L80. L70 viene richiamato da L80.

2.8 Richiamo indiretto di sottoprogrammi (CALL)

Funzione

A seconda delle condizioni assegnate può anche essere possibile richiamare diversi sottoprogrammi. A questo scopo in una variabile del tipo STRING va depositato il nome del sottoprogramma. Il richiamo di un sottoprogramma avviene con CALL e il nome della variabile.

Programmazione

```
CALL <Programmname>
```

Parametri

CALL	Parola chiave per il richiamo indiretto del sottoprogramma
<Programmname>	Variabile o costante del tipo String
	Nome del programma contenente la parte di programma da eseguire



Cautela

Possono essere richiamati indirettamente soltanto sottoprogrammi senza assegnazione di parametri. Per il richiamo diretto di un sottoprogramma occorre depositarne il nome in una delle costanti String

Esempio

Richiamo diretto con costante String

```
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PEUZZO1_SPF"
```

Richiamo indiretto tramite variabile

```
DEF STRING[100] PROGNAME  
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PEUZZO1_SPF"  
CALL PROGNAME
```

Il sottoprogramma PEZZO1 viene assegnato alla variabile PROGNAME. Mediante CALL e l'indicazione delle percorso, il programma viene richiamato indirettamente.

2.9 Ripetizione di una parte di programma con programmazione indiretta (CALL)

Funzione

Con CALL si possono richiamare in modo indiretto i sottoprogrammi nei quali vengono eseguite le ripetizioni di parti di programma definite con BLOCK in base a label iniziale e label finale.

Programmazione

```
CALL <Programmname> BLOCK <Startlabel> TO <Endlabel>  
CALL BLOCK <Startlabel> TO <Endlabel>
```

Parametri

CALL	Parola chiave per il richiamo indiretto del sottoprogramma
<Programmname> (opzione)	Variable o costante del tipo String Nome del programma che contiene la parte di programma da eseguire. Se non è stato programmato nessun <Programmname>, nel programma attuale si cercherà la parte di programma con <Startlabel> <Endlabel> che verrà eseguita.
BLOCK ... TO ...	Parola chiave per ripetizione indiretta della parte di programma
<Startlabel> <Endlabel>	Variable o costante del tipo String rimanda all'inizio o alla fine delle parti di programma da eseguire

Esempio

DEF STRING[20] label iniziale, LABEL FINALE	
LABEL INIZIALE = "LABEL_1"	
LABEL FINALE = "LABEL_2"	
...	
CALL "CONTUR_1" BLOCK LABEL INIZIALE TO LABEL FINALE ...	
M17	
PROC CONTUR_1 ...	
LABEL_1	;inizio ripetizione della parte di programma
N1000 G1 ...	
LABEL_2	;fine ripetizione della parte di programma

2.10 Richiamo indiretto di un programma programmato in linguaggio ISO (ISOCALL)

Funzione

Con il richiamo indiretto ISOCALL si può richiamare un programma che è stato programmato in linguaggio ISO. In questo caso viene attivato il modo ISO impostato nei dati di macchina. Alla fine del programma viene ripristinato il modo di elaborazione originario. Se nei dati di macchina non è stato impostato nessun modo ISO, il richiamo del sottoprogramma avviene nel modo Siemens.

Per ulteriori informazioni sul modo ISO, vedere
/FBFA/ Descrizione delle funzioni, Dialetti ISO.

Programmazione

```
ISOCALL <nome programma>
```

Parametri

ISOCALL	Richiamo del sottoprogramma con il quale viene abilitato il modo ISO impostato nei dati di macchina
<nome programma>	Variabile o costante del tipo String Nome del programma programmato in linguaggio ISO

Esempio: richiamo di un profilo con programmazione dei cicli nel modo ISO

```

%_N_0122_SPF                                ;Descrizione del profilo nel modo ISO
N1010 G1 X10 Z20
N1020 X30 R5
N1030 Z50 C10
N1040 X50
N1050 M99
N0010 DEF STRING[5] PROGNAME = "0122"        ;Partprogram Siemens (ciclo)
...
N2000 R11 = $AA_IW[X]
N2010 ISOCALL PROGNAME
N2020 R10 = R10+1                             ;Esecuzione del programma 0122.spf nel
N2300 ...                                     ;modo ISO
N2400 M30

```

2.11 Richiamo sottoprogramma con indicazione del percorso e parametri (PCALL)

Funzione

Con PCALL si possono richiamare dei sottoprogrammi con indicazione assoluta del percorso e trasferimento dei parametri.

Programmazione

```
PCALL <Percorso/NomeProgramma>(Parametro 1, ..., Parametro n)
```

Parametri

PCALL	Parola chiave per il richiamo di un sottoprogramma con indicazione assoluta del percorso.
<Percorso>	Indicazione assoluta delle percorso iniziante con "/", compresi i nomi dei sottoprogrammi. Se non è stato impostato un percorso assoluto, PCALL si comporta come un richiamo standard di sottoprogramma con identificatori di programma. L'identificatore di programma viene impostato senza intestazione _N_ e senza estensione. Se si desidera programmare il nome del programma con intestazione ed estensione occorre definirlo esplicitamente come Extern indicando intestazione ed estensione.
Parametri 1 ... n	Parametro attuale corrispondente all'istruzione PROC del sottoprogramma.

Esempio

```
PCALL/_N_WKS_DIR/_N_ALBERO_WPD/ALBERO(parametro1, parametro2, ...)
```

2.12 Estensione del percorso di ricerca nei richiami di sottoprogramma con CALLPATH

Funzione

Con l'istruzione CALLPATH si può ampliare il percorso di ricerca per il richiamo dei sottoprogrammi. In questo modo si possono anche richiamare i sottoprogrammi da una directory pezzo non selezionata senza dover indicare il nome di percorso assoluto completo del sottoprogramma.

L'ampliamento del percorso di ricerca avviene prima della impostazione per i cicli utente (_N_CUS-DIR).

Disabilitazione dell'ampliamento del percorso di ricerca

L'ampliamento del percorso di ricerca viene disabilitato con queste condizioni:

- CALLPATH con stringa vuota
- CALLPATH senza parametri
- Fine partprogram
- Reset

Programmazione

Aggiunta dei sottoprogrammi, esistenti all'esterno del Filesystems dell'NCK ai file di sistema già presenti nell'NCK.

CALLPATH <Percorso>

Parametri

CALLPATH	Parola chiave per l'ampliamento programmabile del percorso di ricerca. L'istruzione CALLPATH viene programmata in una propria riga del partprogram.
<Percorso>	Variabile o costante del tipo String. Contiene l'indicazione assoluta del percorso, che inizia con "/", per una directory dove si deve ampliare il percorso di ricerca. il percorso dovrà essere completato con l'indicazione di prefisso e suffisso (es.: /_N_WKS_DIR/_N_WST_WPD). Se <Percorso> contiene una stringa vuota oppure se CALLPATH viene richiamato senza parametri, l'istruzione per il percorso di ricerca viene annullata. La lunghezza max.del percorso è di 128 Byte.

Nota

CALLPATH verifica se è effettivamente disponibile il nome del percorso programmato. In caso contrario l'esecuzione del partprogram viene interrotta con l'allarme blocco di correzione 14009 .

Esempio

```
CALLPATH ("/_N_WKS_DIR/_N_MYWPD_WPD")
```

In questo modo si imposta il seguente percorso di ricerca (la posizione 5. è nuova):

1. directory attuale/identificatore sottoprogramma
2. directory attuale/identificatore sottoprogramma_SPF
3. directory attuale/identificatore sottoprogramma_MPF
4. /_N_SPF_DIR/ identificatore sottoprogramma_SPF
5. /_N_WKS_DIR/_N_MYWPD/identificatore sottoprogramma_SPF
6. N_CUS_DIR/_N_MYWPD/identificatore sottoprogramma_SPF
7. /_N_CMA_DIR/ identificatore sottoprogramma_SPF
8. /_N_CST_DIR/ identificatore sottoprogramma_SPF

Nota

CALLPATH può essere programmato anche nei file INI. In questo caso sarà attivo per tutta la durata di esecuzione del file INI (file WPD-INI o programma di inizializzazione per dati NC attivi ad es. frame nel 1° canale _N__N_CH1_UFR_INI). Successivamente il programma di inizializzazione verrà di nuovo resettato

2.13 Esecuzione di un sottoprogramma esterno (EXTCALL)

Funzione

Con EXTCALL è possibile caricare un programma da HMI nel modo "Elaborazione dall'esterno". In questo modo si possono caricare ed eseguire tutti i programmi disponibili nelle directory della struttura dell'HMI.

Programmazione

EXTCALL ("**<percorso/nome programma>**")

Parametri

EXTCALL
<percorso/nome programma>

Parola chiave per il richiamo del sottoprogramma

Costante/variabile del tipo STRING

È possibile inserire un percorso assoluto (risp. un percorso relativo) o un nome di programma.

Il nome del programma viene immesso con/senza intestazione _N_ e senza estensione.

È possibile aggiungere un'estensione al nome del programma con il carattere <_>.

Esempio:

EXTCALL ("/_N_WKS_DIR/_N_ALBERO_WPD/_N_ALBERO_SPF") oppure EXTCALL ("ALBERO")

Nota

I sottoprogrammi esterni non possono contenere istruzioni di salto come GOTO, GOTOB, CASE, FOR, LOOP, WHILE o REPEAT.

Sono consentiti costrutti IF-ELSE-ENDIF.

Inoltre sono possibili richiami di sottoprogrammi e annidamenti di richiami EXTCALL.

RESET, POWER ON

Con RESET e POWER ON vengono interrotte le chiamate esterne ai sottoprogrammi e cancellata la relativa memoria di caricamento (swapping).

Un sottoprogramma selezionato per l'"Elaborazione dall'esterno", anche con attivazione di RESET / fine partprogram rimane selezionato per l'"elaborazione dall'esterno". La selezione perde valore se si attiva POWER ON.

Esempi

1. Elaborazione dal disco rigido locale

Sistemi: SINUMERIK solution line / powerline con HMI Advanced

Il programma principale "_N_MAIN_MPF" si trova nella memoria NC ed è selezionato per l'elaborazione:

```
N010 PROC MAIN
N020 ...
N030 EXTCALL ("SGROSSARE")
N040 ...
N050 M30
```

Il sottoprogramma da caricare "_N_SGROSSARE_SPF" si trova sul disco rigido locale, nella directory "_N_WKS_DIR/_N_WST1".

Il percorso per il sottoprogramma è predefinito in SD42700:

SD42700 \$SC_EXT_PROG_PATH = "_N_WKS_DIR/_N_WST1"

```
N010 PROC SGROSSARE
N020 G1 F1000
N030 X= ... Y= ... Z= ...
N040 ...
...
...
N999999 M17
```

2. Elaborazione dell'unità di rete

Sistemi: SINUMERIK solution line / powerline con HMI sl / HMI Advanced / HMI Embedded

Il sottoprogramma da caricare "Contorno2.spf" si trova sull'unità di rete, nella directory "\\R4711\Pezzi".

```
...
N... EXTCALL ("\\R4711\Pezzi\Contorno2.spf")
...
```

Impostazione di un percorso esterno del programma

Il percorso per la directory del sottoprogramma esterna può essere preimpostato con il dato setting:

SD42700 \$SC_EXT_PROG_PATH

Insieme al percorso e/o all'identificatore del sottoprogramma definito con il richiamo EXTCALL, si ottiene il percorso assoluto del programma da richiamare.

Conseguenze

Richiamo EXTCALL con indicazione del percorso assoluto

Se il sottoprogramma esiste sotto il percorso indicato, viene eseguito dopo il richiamo EXTCALL. Se non esiste, l'esecuzione del programma viene interrotta.

Richiamo EXTCALL con indicazione del percorso relativo / senza indicazione del percorso

Con un richiamo EXTCALL con indicazione del percorso relativo oppure senza indicazione del percorso, viene effettuata una ricerca nelle memorie di programma disponibili in base al modello seguente:

- Se in SD42700 è preimpostata un'indicazione del percorso, la ricerca viene effettuata innanzitutto a partire da questo percorso, in base all'indicazione nel richiamo EXTCALL (nome programma ev. con indicazione percorso relativo). Il percorso assoluto si ottiene dalla concatenazione di caratteri:
 - l'indicazione di percorso preimpostata in SD42700
 - il carattere "/" come separatore
 - il percorso e/o l'identificatore del sottoprogramma definito con EXTCALL
- Se il sottoprogramma richiamato non viene individuato nel percorso preimpostato, la successiva ricerca viene effettuata nelle directory della memoria utente in base all'indicazione nel richiamo EXTCALL.
- Se il sottoprogramma richiamato non viene individuato nella memoria di programma in esame (ad es. CompactFlash Card), la ricerca viene effettuata nella memoria di programma successiva (ad es. unità di rete) in base al punto 1 e 2.
- La ricerca termina quando il sottoprogramma viene individuato per la prima volta. Qualora la ricerca non portasse ad alcun risultato, si ha un'interruzione del programma.

Nota

SINUMERIK powerline con HMI Embedded

Con SINUMERIK powerline con HMI Embedded occorre sempre indicare un percorso assoluto.

Memoria di programma esterna

In base al sistema (SINUMERIK solution line / powerline), all'interfaccia presente (HMI sl / HMI Advanced / HMI Embedded) e alle opzioni, memorie di programma esterne possono trovarsi sui seguenti supporti dati:

- CompactFlash Card
- Unità di rete
- Unità USB
- Disco rigido locale

Nota

Elaborazione dall'esterno tramite interfaccia USB con SINUMERIK solution line

Se occorre trasferire programmi esterni da un'unità USB esterna tramite interfaccia USB, è possibile utilizzare solo l'interfaccia tramite X203 con il nome "TCU_1".

L'uso di un FlashDrive USB come memoria persistente non è consigliabile.

Nota

Elaborazione dall'esterno tramite interfaccia V24 con SINUMERIK solution line

Con HMI Embedded, utilizzando il softkey "Elaborazione dall'esterno" è possibile trasferire programmi esterni tramite l'interfaccia V24 in NC.

Memoria di caricamento impostabile (buffer FIFO)

Per l'elaborazione di un programma nel modo "Elaborazione dall'esterno" (programma principale o sottoprogramma) è necessaria una memoria di caricamento nell'NCK. La dimensione della memoria di caricamento è preimpostata a 30 KB e può essere modificata in funzione delle esigenze, come altri dati macchina attinenti alla memoria, solo dal costruttore della macchina.

Per tutti i programmi (programmi principali o sottoprogrammi) che devono essere elaborati contemporaneamente nel modo "Elaborazione dall'esterno", deve essere impostata una memoria di caricamento.

Costruttore della macchina

Rivolgersi al costruttore della macchina se si rende necessario ampliare le dimensioni e il numero delle memorie di caricamento.

Per ulteriori informazioni sull'"Elaborazione dall'esterno" vedere:

Bibliografia:

/FB1/ Manuale delle funzioni di base; BAG, canale, funzionamento del programma (K1)

Visualizzazione blocco, blocco singolo e comportamento con arresto NC

In caso di elaborazione dal disco rigido, nonché di `EXTCALL`, è possibile solo la visualizzazione blocco 3 HMI Advanced "Svolgimento del programma". Questa impostazione viene mantenuta per blocco singolo o stato arresto NC.

2.14 Esclusione blocco singolo (SBLOF, SBLON)

Funzione

Disattivazione del blocco singolo specifica per programma

I programmi contrassegnati con SBLOF vengono eseguiti per ogni tipo di blocco singolo, come un blocco completo. SBLOF si trova nella riga PROC ed ha effetto fino alla fine o all'interruzione di un sottoprogramma. Con l'istruzione del salto di ritorno si stabilisce se alla fine del sottoprogramma deve avvenire l'arresto oppure no.

Salto di ritorno con M17: arresto al termine del sottoprogramma

Salto di ritorno con RET: nessun arresto al termine del sottoprogramma

SBLOF vale anche nei sottoprogrammi richiamati.

Esempio di sottoprogramma senza arresto in blocco singolo:

```
PROC ESEMPIO SBLOF
G1 X10
RET
```

Programmazione

PROC ... SBLOF ;l'istruzione può trovarsi in un blocco PROC o da sola nel blocco
SBLON ;l'istruzione deve trovarsi in un blocco a sé stante

Disattivazione del blocco singolo all'interno del programma

SBLOF deve trovarsi da solo in un blocco. A partire da questo blocco, il blocco singolo viene disattivato

- fino al prossimo SBLON oppure
- fino alla fine del livello di sottoprogramma attivo.

Parametri

SBLOF	Disattivare il blocco singolo
SBLON	Riattivare il blocco singolo

Esempio di disattivazione del blocco singolo all'interno del programma

N10 G1 X100 F1000	
N20 SBLOF	
N30 Y20	;Disattivazione del blocco singolo
N40 M100	
N50 R10=90	
N60 SBLON	
N70 M110	;Riattivazione del blocco singolo
N80 ...	

Nel funzionamento a blocchi singoli il settore compreso fra N20 e N60 viene eseguito come un unico passo.

Esempio di ciclo che per l'utente deve avere l'effetto di un comando

Programma principale

```
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30
Programma del ciclo:1
N100 PROC CYCLE1 DISPLOF SBLOF ;disattivazione del blocco singolo
N110 R10=3*SIN(R20)+5
N120 IF (R11 <= 0)
N130 SETAL(61000)
N140 ENDIF
N150 G1 G91 Z=R10 F=R11
N160 M17
```

Il ciclo CYCLE1 viene elaborato con il blocco singolo attivo, per l'elaborazione del CYCLE1 è necessario quindi premere una volta il tasto di start.

Esempio di ASUP non visibile, attivato da PLC, per attivare gli spostamenti origine modificati e le correzioni utensile.

```
N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF 0 GOTOF _G500
-->1 GOTOF _G54 2 GOTOF _G55 3
-->GOTOF _G56 4 GOTOF _G57
-->DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET
```

Esempio con DM 10702 IGNORE_SINGLEBLOCK_MASK, Bit 12 = 1 senza arresto

Nel tipo di blocco singolo SBL2 (Stop ad ogni riga del partprogram) nell'istruzione SBLON.

;SBL2 è abilitato	
;\$MN_IGNORE_SINGLEBLOCK_MASK = 'H1000'	;nel DM 10702: settare il bit 12 = 1
N10 G0 X0	;arrestare l'elaborazione in questa riga del ;partprogram
N20 X10	;arrestare l'elaborazione in questa riga del ;partprogram
N30 CYCLE	;blocco di movimento generato dal ciclo
PROC CYCLE SBLOF	;disattivare lo stop blocco singolo
N100 R0 = 1	
N110 SBLON	; siccome il DM 10702: bit 12 = 1 non ;gestopt
N120 X1	;arrestare l'elaborazione in questa riga del ;partprogram
N140 SBLOF	
N150 R0 = 2	
RET	
N50 G90 X20	;arrestare l'elaborazione in questa riga del ;partprogram
M30	

Esempio di soppressione blocco singolo nell'annidamento di programmi

N10 X0 F1000	;blocco singolo abilitato
N20 UP1(0)	;in questo blocco avviene l'interruzione
PROC UP1(INT _NR) SBLOF	
N100 X10	;blocco singolo OFF
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 SBLON	;blocco singolo ON
N220 X22	;in questo blocco avviene l'interruzione
N230 UP3(0)	
PROC UP3(INT _NR)	
N302 SBLOF	;blocco singolo OFF
N300 X30	
N310 SBLON	;blocco singolo ON
N320 X32	;in questo blocco avviene l'interruzione
N330 SBLOF	;blocco singolo OFF
N340 X34	
N350 M17	;SBLOF abilitato
N240 X24	;in questo blocco avviene l'interruzione, ;SBLON attivo
N250 M17	;in questo blocco avviene l'interruzione, ;SBLON attivo
N120 X12	

N130 M17	;in questo blocco di salto avviene ;l'interruzione, ;SBLOF dell'istruzione PROC attivo
N30 X0	;in questo blocco avviene l'interruzione
N40 M30	;in questo blocco avviene l'interruzione

Condizioni marginali

- Nei cicli la visualizzazione del blocco attuale può essere soppressa con DISPLOF.
- Se DISPLOF viene programmata insieme a SBLOF, in caso di arresti del blocco singolo all'interno del ciclo, il richiamo dei cicli viene visualizzato come prima.
- Se l'arresto in blocco singolo è stato disattivato nell'ASUP di sistema oppure nell'ASUP utente con bit0 = 1 oppure bit1 = 1 del DM 10702: IGNORE_SINGLEBLOCK_MASK, programmando SBLON nell'ASUP viene riattivato l'arresto in blocco singolo.
- L'arresto in blocco singolo nell'ASUP utente viene disattivato con il DM 20117: IGNORE_SINGLEBLOCK_ASUP e non può più essere riattivato programmando SBLON.
- Selezionando SBL3 viene annullata l'istruzione SBLOF.
- Ignorare lo stop blocco singolo nel blocco singolo tipo 2. Nel tipo di blocco singolo tipo 2 (SBL2) l'arresto nel blocco SBLON **non** avviene se è impostato il Bit12 = 1 del DM 10702: IGNORE_SINGLEBLOCK_MASK.

Nota

Per ulteriori informazioni sulla visualizzazione blocco con/senza annullamento del blocco singolo vedere
/FB/, Manuale delle funzioni di base; BAG, canale, funzionamento da programma (K1), capitolo "Blocco singolo".

Disattivazione del blocco singolo per sottoprogrammi asincroni

Per poter eseguire nella funzione blocco singolo un ASUP (sottoprogramma asincrono) in un passo, bisogna programmare nell'ASUP un'istruzione PROC con SBLOF. Questo vale anche per la funzione "ASUP di sistema editabile" attraverso il DM 11610: ASUP_EDITABLE.

Esempio di "ASUP di sistema editabile":

N10 PROC ASUP1 SBLOF DISPLOF	
N20 IF \$AC_ASUP=='H200'	
N30 RET	;Nessun Repos al cambio dei modi operativi
N40 ELSE	
N50 REPOSA	;Repos in tutti gli altri casi
N60 ENDIF	

Influenze sul programma in blocco singolo

Con la funzione blocco singolo l'operatore ha la possibilità di eseguire il partprogram blocco per blocco. Per la funzione blocco singolo si possono utilizzare i seguenti modi di impostazione:

- SBL1: blocco singolo IPO con stop dopo ogni blocco contenente funzioni di macchina.
- SBL2: blocco singolo IPO con stop dopo ogni blocco.
- SBL3: arresto in ciclo (selezionando SBL3 viene disattivata l'istruzione SBLOF).

Soppressione blocco singolo nell'annidamento programmi

Se in un sottoprogramma è stato programmato SBLOF nell'istruzione PROC, avviene l'arresto sull'istruzione del salto di ritorno M17. In questo modo si evita che nel programma che ha eseguito il richiamo possa essere eseguito subito il blocco successivo. Se in un sottoprogramma con SBLOF, senza SBLOF nell'istruzione PROC, viene eseguita la disattivazione del blocco singolo, l'elaborazione si arresta solo dopo il primo blocco contenente funzioni macchina, del programma che ha eseguito il richiamo. Se non si vuole che questo avvenga si dovrà riprogrammare l'istruzione SBLON nel sottoprogramma ancora prima del salto di ritorno (M17). In caso di salto di ritorno in un programma sovraordinato con RET l'arresto non avviene.

2.15 Disattivazione della visualizzazione del blocco attuale (DISPLOF)

Funzione

Con DISPLOF è possibile escludere la visualizzazione del blocco attuale per un sottoprogramma. DISPLOF si trova alla fine dell'istruzione PROC. Al posto del blocco attuale viene visualizzato il richiamo del ciclo o del sottoprogramma.

Come standard è attiva la visualizzazione del blocco. Disattivando la visualizzazione blocco mediante DISPLOF la soppressione rimane attiva fino al ritorno dal sottoprogramma o fino alla fine del programma.

Programmazione

Nella riga di istruzioni PROC

```
PROC ... DISPLOF
```

Se dal sottoprogramma con attributo DISPLOF vengono richiamati ulteriori sottoprogrammi, anche in questi la visualizzazione del blocco attuale risulta soppressa. Se un sottoprogramma con visualizzazione del blocco soppressa viene interrotto da un sottoprogramma asincrono, i blocchi del sottoprogramma attuale vengono visualizzati.

Parametri

DISPLOF

Escludere la visualizzazione del blocco attuale

Esempio di disattivazione della visualizzazione del blocco attuale nel ciclo

<pre>%_N_CYCLE_SPF PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF DEF REAL DIFF G01 RET</pre>	<pre>;\$PATH=/_N_CUS_DIR ; sopprimere la visualizzazione del blocco ; attuale ; come blocco attuale viene ora ; visualizzato il richiamo del ciclo ; ad es.: CYCLE(X, 100.0) ; contenuto cicli ; ritorno al sottoprogramma, viene ; visualizzato di nuovo il blocco successivo ; del programma che ha effettuato il richiamo</pre>

2.16 Identificazione di sottoprogrammi con preparazione (PREPRO)

Funzione

Con la parola chiave `PREPRO` durante l'avviamento alla fine della riga di istruzioni `PROC` è possibile contrassegnare tutti i dati.

Costruttore della macchina

Questo tipo di preparazione dipende dal corrispondente dato macchina impostato. Fare attenzione alle indicazioni fornite in merito dal costruttore della macchina.

/FB3/ Manuale delle funzioni speciali; Preparazione (V2)

Programmazione

Nella riga di istruzioni `PROC`

`PROC ... PREPERO`

Parametri

`PREPERO`

Parola chiave per l'identificazione di tutti i file preparati al caricamento, dei programmi NC salvati nelle directory dei cicli

Lettura dei sottoprogrammi con preparazione e chiamata ai sottoprogrammi

Sia durante l'avviamento dei sottoprogrammi predisposti con parametri, sia durante la chiamata ai sottoprogrammi, gli elenchi dei cicli vengono trattati nella stessa sequenza:

1. `_N_CUS_DIR` cicli utente
2. `_N_CMA_DIR` cicli costruttore
3. `_N_CST_DIR` cicli standard

. In caso di programmi NC con lo stesso nome e di diverso tipo, viene attivata la prima istruzione `PROC` trovata e ignorata (senza messaggio di allarme) l'altra istruzione `PROC`.

2.17 Cicli: Parametrizzazione dei cicli utente

Funzione

Con i file cov.com e uc.com si possono parametrizzare cicli propri.

Il file cov.com viene fornito con i cicli standard e deve essere opportunamente ampliato. Il file uc.com deve essere creato dall'utilizzatore.

Entrambi i file sono da caricare nella directory "Cicli utente" del file system passivo (oppure con l'indicazione del relativo percorso):

```
; $PATH=/_N_CUS_DIR
```

.

File e percorsi

cov.com_COM	Panoramica dei cicli
uc.com	Descrizione del richiamo dei cicli

Adattamento di cov.com – panoramica dei cicli

Il file cov.com fornito con i cicli standard ha la seguente struttura:

%_N_COV_COM	nome del file
;\$PATH=/_N_CST_DIR	indicazione del percorso
;Vxxx 11.12.95 panoramica dei cicli	riga di commento
C1(CYCLE81) foratura, centratura	richiamo del 1° ciclo
C2(CYCLE82) foratura, svasatura	richiamo del 2° ciclo
...	
C24(CYCLE98) sequenza di filetti	richiamo dell'ultimo ciclo
M17	fine del file

Programmazione

Per ogni nuovo ciclo inserito si deve inserire una riga con la seguente sintassi:

C<Numero> (<Nome del ciclo>) Blocco di commento

Numero: un numero intero qualsiasi non ancora utilizzato nel file;

Nome del ciclo: nome del ciclo da implementare

Testo del commento: testo di commento al ciclo

Esempio:

```
C25 (MIO_CICLO_1) Ciclo_utente_1
```

```
C26 (CICLO_SPECIALE)
```

Esempio di file uc.com - descrizione cicli utente

La descrizione avviene nel proseguimento dell'esempio:
per entrambi i cicli seguenti deve avvenire una nuova parametrizzazione.

PROC MIO_CICLO_1 (REAL PAR1, INT PAR2, CHAR PAR3, STRING[10] PAR4)	
; il ciclo ha i seguenti parametri di trasferimento:	
;PAR1:	valore REAL nel campo da -1000.001 <= PAR2 <= 123.456, preimpostato con 100
;PAR2:	valore positivo intero tra 0 <= PAR3 <= 999999, preimpostazione con 0
;PAR3:	1 carattere ASCII
;PAR4:	stringa di lunghezza 10 per il nome di un sottoprogramma
...	
M17	

PROC CICLO_SPECIALE (REAL VALORE1, INT VALORE2)	
; il ciclo ha i seguenti parametri di trasferimento:	
;	
;VALORE1:	valore REAL senza limitazione del campo dei valori e senza preimpostazione
;VALORE2:	valore intero senza limitazione del campo dei valori e senza preimpostazione
...	
M17	

Relativo file uc.com:

%_N_UC_COM
;\$PATH=/_N_CUS_DIR
//C25 (MIO_CICLO_1) Ciclo_utente_1
(R/-1000.001 123.456 / 100 /Parametro_2 del ciclo)
(I/0 999999 / 1 / Valore intero)
(C/"A" / Parametro carattere)
(S///Nome del sottoprogramma)
//C26 (CICLO_SPECIALE)
(R///Lunghezza totale)
(I/*123456/3/Tipo di elaborazione)
M17

Esempio di entrambi i cicli

Maschera di visualizzazione per il ciclo MIO_CICLO_1

Parametro 2 del ciclo	100
valore intero	1
Parametro carattere	
Sottoprogrammi	

Maschera di visualizzazione per il ciclo CICLO_SPECIALE

Lunghezza totale	100
Tipo di lavorazione	1

Descrizione della sintassi per il file uc.com - descrizione cicli utente**Riga di intestazione per ogni ciclo:**

come nel file cov.com preceduto da "//"

//C <Numero> (<Nome del ciclo>) Testo di commento

Esempio:

//C25 (MIO_CICLO_1) Ciclo_utente_

Riga di descrizione per ogni parametro:(<Codice tipo dato>)/ <valore minimo> <valore massimo>
/ <preimpostazione> /<commento>)**Codice tipo dato:**

R	per REAL
I	per Integer
C	per carattere (1 carattere)
S	per String

Valore minimo, valore massimo (può mancare)

Limiti del valore da impostare che vengono verificati al momento dell'introduzione; non possono essere impostati valori al di fuori di questo campo. Possono essere impostati valori di conteggio gestibili con il tasto di Toggle; essi vengono conteggiati a partire da "*", non sono possibili altri valori.

Esempio:

(I/*123456/1/Tipo di elaborazione)

per i tipi String e Char non sono previsti limiti;

Valore di preimpostazione (può mancare)

Valore preimpostato al richiamo del ciclo nella relativa maschera; esso può essere modificato dalle pagine operative.

Commento

Testo, al massimo di 50 caratteri, che viene visualizzato nella maschera di richiamo per il ciclo prima del campo di introduzione del parametro.

2.18 Tecnica delle macro (DEFINE...AS)

Funzione

Come macro si intende il raggruppamento di singole istruzioni in un'unica istruzione con un proprio nome. Anche le funzioni G, M, H oppure i nomi dei sottoprogrammi L possono essere creati come macro. Al richiamo della macro durante l'elaborazione del programma, le istruzioni programmate come nome della macro vengono eseguite in sequenza.

Utilizzo delle macro

Sequenze di istruzioni che si ripetono, possono essere programmate solo una volta come macro in un unico blocco oppure una sola volta all'inizio del programma. Questa macro può essere poi richiamata ed eseguita in qualunque programma principale o sottoprogramma.

Programmazione

Le macro vengono identificate con la parola chiave DEFINE...AS.

La definizione della macro si presenta nel seguente modo:

DEFINE NAME AS <istruzione>

Esempio:

Definizione di macro:

DEFINE LINIE AS G1 G94 F300

Richiamo nel programma NC:

N20 LINIE X10 Y20

Attivazione della macro

Quando la macro viene caricata nell' NC (softkey "Caricare"), essa diventa attiva.

Parametri



Cautela

Parole chiave e nomi riservati non possono essere ridefiniti nelle macro.

Con la tecnica macro si può modificare pesantemente il linguaggio di programmazione del controllo. Utilizzare quindi la tecnica macro con accortezza!

DEFINE	Definizione di macro
NOME	In questo punto viene indicato il nome della macro
AS	Definizione macro STRING
Istruzione	Istruzioni di programmazione, es. funzioni G, M, H e L

Con la tecnica macro si può definire qualsiasi identificatore, funzione G, M o nome di programma L. Le funzioni H ed L sono programmabili solo a due cifre.

Funzione M / G a tre cifre

Possono essere programmate funzioni M e G a tre cifre.

Esempio:

```
N20 DEFINE M100 AS M6
N80 DEFINE M999 AS M6
```

Nota

Le macro possono essere dichiarate anche nel programma NC. Come nomi di macro sono consentiti solo identificatori. Macro di funzioni G possono essere dichiarate solo in blocchi macro globali del controllo.

Non è possibile l'annidamento di macro.

Esempio di definizione di macro

DEFINE M6 AS L6	Durante il cambio utensile viene richiamato un sottoprogramma che gestisce il trasferimento dei dati necessari. Nel sottoprogramma viene emessa l'effettiva funzione M per il cambio utensile (es. M106).
DEFINE G81 AS DRILL(81)	Emulazione della funzione G DIN
DEFINE G33 AS M333 G333	Durante la filettatura viene richiesta la sincronizzazione con il PLC. La funzione G originaria è stata rinominata tramite dati macchina in G333, per l'utilizzatore la programmazione resta inalterata.

Esempio di file per macro

dopo il caricamento del file macro nel controllo, attivare le macro (vedere sopra). Le macro possono essere utilizzate solo nel partprogram.

```
%_N_UMAC_DEF
```

;\$PATH=_/ _N_DEF_DIR	;macro specifiche dell'utente
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	;Mandrino destrorso, refrigerante on
DEFINE M14 AS M4 M7	;Mandrino sinistrorso, refrigerante on
DEFINE M15 AS M5 M9	;Arresto mandrino, refrigerante off
DEFINE M6 AS L6	;richiamo del programma di cambio utensile
DEFINE G80 AS MCALL	;disattivazione del ciclo di foratura
M30	

Gestione dei file e dei programmi

3.1 Memoria dei programmi

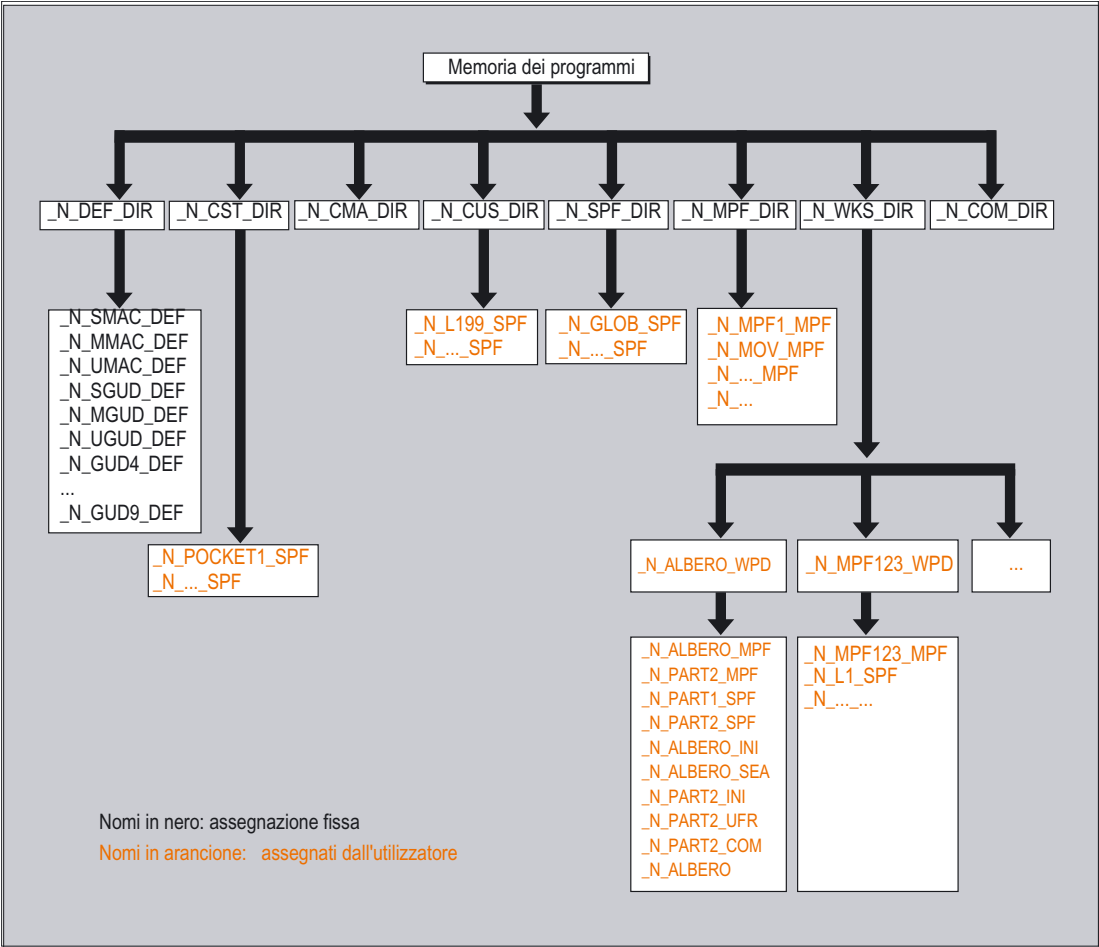
Funzione

Nella memoria dei programmi i file e i dati (ad es. programmi principali e sottoprogrammi, definizioni macro) vengono salvati in modo persistente (→ system file passivi).

Bibliografia:

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Configurazione memoria (S7)

Esistono inoltre alcuni tipi di file che possono essere memorizzati e che, se necessario, devono essere trasferiti nella memoria di lavoro (es. nel caso dell'esecuzione di un determinato pezzo o per inizializzazione).



Directory standard

Le seguenti directory sono standard:

Directory	Contenuto
<code>_N_DEF_DIR</code>	Blocchi dati e blocchi macro
<code>_N_CST_DIR</code>	Cicli standard
<code>_N_CMA_DIR</code>	Cicli costruttore
<code>_N_CUS_DIR</code>	Cicli utente
<code>_N_WKS_DIR</code>	Pezzi
<code>_N_SPF_DIR</code>	Sottoprogrammi globali
<code>_N_MPF_DIR</code>	Programmi principali
<code>_N_COM_DIR</code>	Commenti

Tipi di file

Nella memoria dei programmi possono essere inseriti i seguenti tipi di file:

Tipo di file	Descrizione
<i>name</i> _MPF	Programma principale
<i>name</i> _SPF	Sottoprogramma
<i>name</i> _TEA	Dati macchina
<i>name</i> _SEA	Dati setting
<i>name</i> _TOA	Correzioni utensile
<i>name</i> _UFR	Spostamenti origine/Frame
<i>name</i> _INI	File di inizializzazione
<i>name</i> _GUD	Dati utente globali
<i>name</i> _RPA	Parametri R
<i>name</i> _COM	Commento
<i>name</i> _DEF	Definizioni per dati utente globali e macro

Directory principale dei pezzi (_N_WKS_DIR)

La directory principale dei pezzi è configurata nella memoria dei programmi sotto l'identificativo `_N_WKS_DIR`. La directory principale dei pezzi contiene le relative directory di tutti i pezzi programmati.

Directory dei pezzi (..._WPD)

Per una gestione più flessibile dei dati e dei programmi, è consigliabile raggruppare i dati e i programmi all'interno di un'unica directory.

La directory di un pezzo contiene tutti i file necessari per la lavorazione del pezzo stesso. I file contenuti possono essere programmi principali, sottoprogrammi, programmi di inizializzazione qualsiasi, e file di commento.

I programmi di inizializzazione vengono eseguiti una volta dopo la selezione del programma con il primo avvio del partprogram (in base al dato macchina MD11280 \$MN_WPD_INI_MODE).

Esempio:

La directory `_N_WELLE_WPD`, che è stata creata per il pezzo WELLE, contiene i seguenti file:

<code>_N_ALBERO_MPF</code>	Programma principale
<code>_N_PART2_MPF</code>	Programma principale
<code>_N_PART1_SPF</code>	Sottoprogramma
<code>_N_PART2_SPF</code>	Sottoprogramma
<code>_N_ALBERO_INI</code>	Programma generico di inizializzazione dei dati per il pezzo
<code>_N_ALBERO_SEA</code>	Programma di inizializzazione dei dati di setting

<code>_N_PART2_INI</code>	Programma generico di inizializzazione dei dati per il programma Part 2
<code>_N_PART2_UFR</code>	Programma di inizializzazione per dati di frame per il programma Part 2
<code>_N_ALBERO_COM</code>	File di commento

Creazione di una directory pezzo su un PC esterno

La procedura che segue viene eseguita su una stazione dati esterna. Per la gestione dei dati e dei programmi (dal PC al controllo numerico) direttamente a bordo del controllo numerico, consultare il manuale operativo.

Creazione directory pezzi con indicazione del percorso (\$PATH=...)

Nella seconda riga di un file viene indicato il percorso di destinazione con `$PATH=...`. Il file viene quindi salvato nel percorso indicato.

Esempio:

```
%_N_ALBERO_MPF
; $PATH=/_N_WKS_DIR/_N_ALBERO_WPD
N10 G0 X... Z...
...
M2
```

Il file `_N_ALBERO_MPF` viene salvato nella directory `/_N_WKS_DIR/_N_ALBERO_WPD`.

Creazione directory pezzi senza indicazione del percorso

Se manca l'indicazione del percorso, i file che terminano con `_SPF` vengono salvati nella directory `/_N_SPF_DIR`, quelli che terminano con `_INI` nella memoria di lavoro e tutti i restanti file nella directory `/_N_MPF_DIR`.

Esempio:

```
%_N_ALBERO_SPF
...
M17
```

Il file `_N_ALBERO_SPF` viene salvato nella directory `/_N_SPF_DIR`.

Selezione di un pezzo per la lavorazione

Una directory del pezzo può essere selezionata per l'elaborazione in un determinato canale. Se in questa directory è presente un programma principale con **nome uguale** oppure soltanto un unico programma principale (`_MPF`), questo viene selezionato automaticamente per l'elaborazione.

Bibliografia:

/BAD/ Manuale d'uso HMI Advanced; Capitolo "Joblist" nonché "Selezione di un programma per l'elaborazione"

Percorso di ricerca per il richiamo di un sottoprogramma

Se per il richiamo di un sottoprogramma (o file di inizializzazione) da un partprogram non si indica il percorso di richiamo, il programma richiamato verrà cercato in base ad un percorso prestabilito.

Richiamo sottoprogramma con indicazione del percorso assoluto

Esempio:

```
...
CALL"/_N_CST_DIR/_N_CYCLE1_SPF"
...
```

Richiamo sottoprogramma senza indicazione del percorso assoluto

Di solito i programmi vengono richiamati senza indicazione del percorso.

Esempio:

```
...
CYCLE1
...
```

In base al programma richiamato, la ricerca nelle directory viene effettuata con la successione seguente:

N.	Directory	Descrizione
1	directory attuale / <i>nome</i>	Directory principale dei pezzi senza directory standard _N_MPF_DIR
2	directory attuale / <i>nome_SPF</i>	
3	directory attuale / <i>nome_MPF</i>	
4	/_N_SPF_DIR / <i>nome_SPF</i>	Sottoprogrammi globali
5	/_N_CUS_DIR / <i>nome_SPF</i>	Cicli utente
6	/_N_CMA_DIR / <i>nome_SPF</i>	Cicli costruttore
7	/_N_CST_DIR / <i>nome_SPF</i>	Cicli standard

Programmazione del percorso di ricerca per il richiamo di sottoprogrammi (CALLPATH)

Il percorso di ricerca per il richiamo di un sottoprogramma può essere ampliato con il comando partprogram CALLPATH.

Esempio:

```
...
CALLPATH ("/_N_WKS_DIR/_N_MYWPD_WPD")
...
```

Il percorso di ricerca viene definito a partire dalla 5. posizione (ciclo utente) in funzione della programmazione indicata.

Per ulteriori informazioni per il percorso di ricerca programmabile per i richiami di sottoprogrammi con CALLPATH, vedere capitolo "Estensione del percorso di ricerca nei richiami di sottoprogramma con CALLPATH".

3.2 Memoria di lavoro

Funzione

La memoria di lavoro contiene i dati di sistema ed i dati utente attuali che vengono elaborati dal controllo numerico (file system attivo).

Esempio: dati macchina attivi, correttori utensili, spostamenti origine.

Parametri

Programmi di inizializzazione

In questo caso si tratta di programmi tramite i quali vengono preimpostati i dati della memoria di lavoro (inizializzazione). A questo scopo possono essere utilizzati i seguenti tipi di file.

<code>name_TEA</code>	Dati macchina
<code>name_SEA</code>	Dati setting
<code>name_TOA</code>	Correzioni utensile
<code>name_UFR</code>	Spostamenti origine/Frame
<code>name_INI</code>	File di inizializzazione
<code>name_GUD</code>	Dati utente globali
<code>name_RPA</code>	Parametri R

Settori dati

I dati possono essere suddivisi in diversi settori, nei quali devono avere validità. Un controllo numerico può avere ad esempio diversi canali (non possibile nei controlli 810D CCU1, 840D NCU 571) o anche diversi assi. Ci sono:

Codice	Settori dati
NCK	Dati specifici NCK
CHn	Dati specifici per canale (n identifica il numero del canale)
AXn	Dati specifici per asse (n identifica il numero dell'asse macchina)
TO	Dati dell'utensile
COMPLETE	Tutti i dati

Esempio di creazione del programma di inizializzazione su un PC esterno

Con l'ausilio dell'identificatore del settore dati e dell'identificatore del tipo di file, possono essere definiti i settori che, durante il salvataggio dei dati, vengono considerati come un'unità.

<code>_N_AX5_TEA_INI</code>	Dati macchina per l'asse 5
<code>_N_CH2_UFR_INI</code>	Frame del canale 2
<code>_N_COMPLETE_TEA_INI</code>	Tutti i dati macchina

Dopo la messa in servizio del controllo numerico, nella memoria di lavoro si trova un set di dati che garantisce il corretto funzionamento dello stesso.

Esempio di procedura per controlli numerici a più canali

CHANDATA (numero di canale) per più canali è ammesso solo nel file `N_INITIAL_INI`.

`N_INITIAL_INI` è il file di messa in servizio con il quale vengono inizializzati tutti i dati del controllo numerico.

```
% N_INITIAL_INI
CHANDATA(1)
;Abbinamento assi macchina canale 1
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED[2]=3
CHANDATA(2)
;Abbinamento assi macchina canale 2
$MC_AXCONF_MACHAX_USED[0]=4
$MC_AXCONF_MACHAX_USED[1]=5
CHANDATA(1)
;Dati macchina assiali
;Finestra di arresto preciso grossolano:
$MA_STOP_LIMIT_COARSE[AX1]=0.2 ;Asse 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2 ;Asse 2
Finestra di arresto preciso fine:
$MA_STOP_LIMIT_FINE[AX1]=0.01 ;Asse 1
$MA_STOP_LIMIT_FINE[AX1]=0.01 ;Asse 2
```



Cautela

Istruzione CHANDATA

Nel partprogram, l'istruzione CHANDATA può essere utilizzata esclusivamente per il canale nel quale viene elaborato il programma NC; l'istruzione può essere utilizzata cioè come protezione contro il richiamo involontario di un programma NC in un canale errato.

In caso di errore, l'elaborazione viene interrotta.

Nota

I file INI nelle Joblist non contengono istruzioni CHANDATA.

Salvataggio dei programmi di inizializzazione

I file della memoria di lavoro possono essere salvati su un PC esterno ed in seguito nuovamente ricaricati dallo stesso.

- I file vengono salvati con COMPLETE.
- Con INITIALSi genera per tutti i settori un file INI: `_N_INITIAL_INI`.

Caricamento dei programmi di inizializzazione

I programmi INI possono anche essere selezionati e richiamati come partprogram se vengono utilizzati i dati di un solo canale. In questo modo è anche possibile inizializzare dati controllati da programma.

Nel manuale operativo si possono trovare informazioni relative a tutti i tipi di file.

3.3 Definizione dei dati utente

Funzione

Attenzione

La definizione dei dati utente (GUD) avviene al momento della messa in servizio. I dati macchina necessari devono essere attivati di conseguenza. La memoria utente deve essere già configurata. Tutti i dati macchina rilevanti contengono la particella GUD.

La definizione dei dati utente (GUD) può avvenire nel settore Servizi della superficie HMI. In questo modo si può evitare un caricamento dell'archivio dei dati (%_N_INITIAL_INI).

In questo caso vale:

- I file di definizione che si trovano sul disco fisso non sono attivi.
- I file di definizione che si trovano sull' NC sono sempre attivi.

Programmazione

La programmazione delle singole variabili GUD avviene con l'istruzione DEF:

```
DEF Settore VL-Stop nome del tipo[.., ...]=valore
```

Parametri

Settore	Il settore identifica la variabile come variabile GUD e definisce il campo di validità: NCK NCK CHAN Canale
Stop VL	Attributo opzionale di stop preelaborazione: SYNR stop preelaborazione in lettura SYNW stop preelaborazione in scrittura SYNRW stop preelaborazione in lettura/scrittura
Tipo	Tipo di dati BOOL REAL INT AXIS FRAME STRING CHAR
Nome	Nome della variabile
[... , ...]	Limiti opzionali per variabili di campo
Valore	Valore preimpostato opzionale, impostazione di più valori nei campi, separati dalla virgola o REP (w1) , SET(w1, W2, ...), (w1, w2, ...) Nel tipo Frame non sono possibili valori di inizializzazione.

Esempio di un file di definizione, dati globali (Siemens)

```
%_N_SGUD_DEF
; $PATH=/_N_DEF_DIR
DEF NCK REAL RTP           ;piano di svincolo
DEF CHAN INT SDIS          ;distanza di sicurezza
M30
```

Esempio di un file di definizione, dati globali (costruttore macchina)

```
%_N_MGUD_DEF
; $PATH=/_N_DEF_DIR
; definizioni dei dati globali del costruttore della macchina
```

DEF NCK SYNRW INT NUMERO PEZZI	;blocco preelaborazione implicito; in ;lettura/scrittura ;dato speciale presente nel controllo numerico ;accesso da parte di tutti i canali
DEF CHAN INT TABELLE UTENSILI[100]	;tabella utensili per immagine specifica ;Rappresentazione ;il numero utensile nei posti magazzino
M30	;tabella generata separatamente per ogni canale

Nomi riservati dei blocchi

Nella directory / _N_DEF_DIR possono essere memorizzati i seguenti blocchi:

_N_SMAC_DEF	contiene le definizioni delle macro (Siemens – applicazioni di sistema)
_N_MMAC_DEF	contiene le definizioni delle macro (costruttore della macchina)
_N_UMAC_DEF	contiene le definizioni delle macro (utilizzatore)
_N_SGUD_DEF	contiene le definizioni per dati globali (Siemens, applicazioni di sistema)
_N_MGUD_DEF	contiene le definizioni per dati globali (costruttore della macchina)
_N_UGUD_DEF	contiene le definizioni per dati globali (utilizzatore)
_N_GUD4_DEF	definibile liberamente
_N_GUD5_DEF	contiene le definizioni per i cicli di misura (Siemens - applicazioni di sistema)
_N_GUD6_DEF	contiene le definizioni per i cicli di misura (Siemens - applicazioni di sistema)
_N_GUD7_DEF	contiene le definizioni per i cicli standard (Siemens - applicazioni di sistema)
_N_GUD8_DEF	definibile liberamente
_N_GUD9_DEF	definibile liberamente

Nota

Se non vengono utilizzati cicli di misura o cicli standard, si possono utilizzare liberamente anche i blocchi riservati per questi cicli

1. Definizione dei dati utente (GUD)

- 1.1 Salvare il blocco _N_INITIAL_INI.
- 1.2 Creare il file delle definizioni per i dati utente nel settore operativo HMI Servizi
- 1.3 Caricare il file di definizione nella memoria dei programmi del controllo numerico
- 1.4 Archiviare i file di definizione
- 1.5 Salvataggio dei dati

2. Generare il file di definizione per i dati utente

I file di definizione possono essere creati su un PC esterno o nel settore operativo Servizi. Esistono anche nomi di file predefiniti (vedere “Nomi riservati dei blocchi”):

```
_N_SGUD_DEF
_N_MGUD_DEF
_N_UGUD_DEF
```


_N_GUD4_DEF ... _N_GUD9_DEF

I file con questi nomi possono contenere definizioni per le variabili GUD.

3. Caricare il file di definizione nella memoria dei programmi del controllo numerico

Il controllo numerico come default crea sempre una directory _N_DEF_DIR. Questo nome viene registrato come percorso nell'intestazione del file di definizione GUD e viene analizzato durante la lettura del file tramite l'interfaccia corrispondente.

4. Attivazione dei file di definizione e riattivazione del loro contenuto

Quando il file di definizione GUD viene caricato nell'NC (softkey "Caricare"), diventa attivo. Vedere "Attivazione automatica...". Se il contenuto di un determinato file di definizione viene riattivato, il set di dati GUD precedente viene cancellato nel file system attivo e vengono ripristinati i nuovi parametri. Se questo processo avviene tramite la finestra di dialogo Servizi HMI => Gestione dati => Definisci e attiva dati utente (GUD), i contenuti delle variabili vengono salvati mediante file INI e ripristinati al termine di questo processo.

5. Salvataggio dei dati

Con l'archiviazione del file _N_COMPLETE_GUD dalla memoria di lavoro viene salvato solo il contenuto dei dati. I file di definizione generati per le variabili utente globali devono essere archiviati separatamente.

Anche le attribuzioni delle variabili ai dati utente globali vengono memorizzate in _N_INITIAL_INI; i nomi devono essere identici a quelli contenuti nei file di definizione.

3.4 Livelli di protezione per i dati utente, DM, DS e istruzioni NC

3.4.1 Definizione dei livelli di protezione per i dati utente (GUD)

Funzione

Definendo i criteri di accesso, si possono proteggere i blocchi GUD da manipolazioni. Nei cicli si possono interrogare le variabili GUD che vengono quindi protette da modifiche non ammesse attraverso l'interfaccia HMI o dal programma. La protezione all'accesso si riferisce a **tutte** le variabili che sono state definite in questo blocco. Nel caso di accesso non consentito, il controllo numerico emette l'allarme corrispondente.

Programmazione

I livelli di protezione per l'intero blocco vengono indicati dopo le intestazioni.

%_N_MGUD_DEF	;tipo di blocco
; \$PATH=/_N_DEF_DIR	;percorso
APR valore APW n	;livelli di protezione in una riga separata

La protezione con l'accesso con il livello di protezione desiderato viene programmata nel blocco GUD, precedentemente alla prima definizione di una variabile. Le parole chiave devono trovarsi in un proprio blocco.

Parametri

livello di protezione:	all'accesso (Access Protection)
APW n	in scrittura (Write)
APR n	in lettura (Read)
n	Livello di protezione n
	da 0 o 10 (livello max.)
	a 7 o 17 (livello min.)
Significato dei livelli di protezione n:	
0 oppure 10	SIEMENS
1 oppure 11	OEM_HIGH
2 oppure 12	OEM_LOW
3 oppure 13	Utente finale
4 oppure 14	Selettore a chiave 3
...	...
7 oppure 17	Selettore a chiave 0
APW 0-7, APR 0-7	Questi valori sono consentiti nei blocchi GUD e nell'indicazione dei livelli di protezione per le singole variabili nell'istruzione REDEF.
La protezione di lettura e scrittura non ha effetto dalla superficie operativa e nel programma NC o nel modo operativo MDA.	
APW 10...17, APR 10...17:	Questi valori sono consentiti solo per il livello di protezione specifico per un blocco GUD.
in questo caso la protezione di lettura e scrittura ha effetto per la superficie operativa.	

Nota

Se la protezione riguarda un intero file, le istruzioni devono trovarsi prima delle prime definizioni del file oppure nell'istruzione **REDEF** del relativo dato, vedere il capitolo "Livelli di protezione per istruzioni NC"

Esempio di file di definizione con protezione all'accesso in scrittura

(costruttore della macchina), lettura (selettore a chiave 2 sulla superficie operativa):

```
%_N_GUD6_DEF
```

```
;$PATH=/_N_DEF_DIR
```

```
APR 15 APW 12
```

;livelli di protezione per tutte le seguenti variabili

```
DEF CHAN REAL_CORRVAL
```

```
DEF NCK INT MYCOUNT
```

```
...
```

```
M30
```

Prima attivazione del file di definizione GUD

La prima volta che si attiva un file di definizione GUD, l'eventuale diritto di accesso definito in esso contenuto viene analizzato e sottoposto automaticamente ai diritti di scrittura e lettura del file di definizione GUD.

Nota

Le registrazioni dei diritti di accesso nel file di definizione GUD possono limitare ma non ampliare la necessaria autorizzazione di accesso al file di definizione GUD.

Esempio:

Nel file di definizione _N_GUD7_DEF si trova: APW2

1. Il file _N_GUD7_DEF ha come protezione da scrittura il valore 3. Il valore 3 viene quindi sovrascritto con il valore 2.
2. Il file _N_GUD7_DEF ha come protezione da scrittura il valore 0. Non avviene alcuna modifica.

L'istruzione APW ha effetto anche per l'accesso in scrittura del file.

L'istruzione APR ha effetto anche sul diritto di lettura del file.

Nota

Se inavvertitamente si immette nel file di definizione un valore superiore a quello del diritto di accesso, è necessario caricare nuovamente il file di archivio.

3.4.2 Attivazione automatica di GUD e MAC

Funzione

I file di definizione per le definizioni di GUD e macro, per le HMI Advanced vengono editati nel settore operativo Servizi.

Se un file di definizione viene editato nell' NC, all'uscita dall'editor il sistema chiede se le definizioni devono essere attivate.

Scaricare le definizioni per GUD e macro

Se un file di definizione viene scaricato, il relativo blocco dati, dopo la visualizzazione di una richiesta di conferma del sistema, viene cancellato.

Caricare le definizioni per GUD e macro

Se un file di definizione viene caricato, il sistema chiede se il file deve essere attivato oppure se i dati devono essere salvati. Se si rifiuta l'attivazione, il file non viene caricato.

Se il cursore si trova su un file di definizione caricato, la dicitura del softkey cambia da "Caricare" ad "Attivare" per permettere l'attivazione delle definizioni. Scegliendo "Attivare" il sistema richiede se i dati devono essere mantenuti.

Il salvataggio dei dati avviene solo per i file di definizione di variabili, non per le macro.

Nota

HMI-Advanced

Qualora non sia disponibile una memoria sufficiente per l'attivazione del file di definizione, dopo la modifica dello spazio di memoria occorre caricare il file dall'NC alla PCU e poi nuovamente nell'NC e quindi attivarlo.

Esempio di richiesta all'uscita dell'editor

"Si vogliono attivare le definizioni del file GUD7.DEF?"

"OK": Viene richiesto se i dati attualmente attivi devono essere salvati.

"Devono essere mantenuti i precedenti delle definizioni?"

OK: I blocchi GUD del file di definizione elaborato vengono salvati, le nuove definizioni vengono attivate e i dati salvati vengono nuovamente caricati.

"Interruzione": Le nuove definizioni vengono attivate, i vecchi dati vengono persi.

"Interruzione": Le modifiche eseguite nel file di definizione vengono perse, il relativo blocco dati non viene modificato.

3.4.3 Modificare i livelli di protezione per dati macchina e dati di setting (REDEF, DM, DS)

Funzione

L'utente ha la possibilità di **modificare** i livelli di protezione. Nei dati macchina si possono definire solo dei livelli di protezione bassi mentre per i dati setting si possono impostare anche livelli di protezione superiori.

Programmazione

REDEF Livello di protezione per dati macchina/dati di setting

Parametri

REDEF	Nuova definizione (REDEF inition) es. impostazione di dati macchina e dati di setting
Dato macchina/dato di setting	Dato macchina o dato di setting al quale deve essere abbinato un livello di protezione.
livello di protezione:	all'accesso (A ccess P rotection)
APW n	in scrittura (W rite)
APR n	in lettura (R ead)
n	Livello di protezione n da 0 (livello max.) a 7 (livello minimo)

Resetare i dati macchina/dati di setting

Se si intende annullare la modifica eseguita sarà necessario reimpostare i livelli di protezione originari.

Ampliamenti di REDEF

Per ulteriori informazioni sull'efficacia dell'istruzione **REDEF** nel partprogram, vedere il capitolo "Livelli di protezione per istruzioni NC".

Esempio di modifica dei diritti di accesso dei singoli DM

```
%_N_SGUD_DEF
; $PATH=/_N_DEF_DIR
REDEF $MA_CTRLOUT_SEGMENT_NR APR 2 APW 2
REDEF $MA_ENC_SEGMENT_NR APR 2 APW 2
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 2 APW 2
M30
```

Esempio di ripristino dei diritti di accesso dei singoli DM alle impostazioni originarie

```
%_N_SGUD_DEF  
; $PATH=/_N_DEF_DIR  
REDEF $MA_CTRLOUT_SEGMENT_NR APR 7 APW 2  
REDEF $MA_ENC_SEGMENT_NR APR 0 APW 0  
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 7 APW 7  
M30
```

3.4.4 Livelli di protezione per istruzioni NC (REDEF)

Funzione

Il concetto di protezione attuale per l'accesso ai dati macchina/dati di setting e GUD, viene ampliato con le istruzioni di programma elencate precedentemente. Con REDEF, ad una determinata istruzione di programma, viene associato un livello di protezione 0 ... 7.

Nota

Questa istruzione viene eseguita nell'elaborazione del programma solo quando è presente il relativo diritto di esecuzione.

Programmazione

Codici G come da "lista delle funzioni G/condizioni di percorso"

REDEF (elemento del linguaggio NC) valore **APX**

oppure per un accesso in scrittura dal partprogram oppure da azioni sincrone su variabili di sistema

REDEF (variabile di sistema) valore **APW**

oppure per modificare gli accessi in scrittura o lettura a dati macchina e dati di setting come in precedenza

REDEF (dato macchina/setting) valore **APW**

REDEF (dato macchina/setting) valore **APR**

Parametri

L'istruzione REDEF ha un effetto globale per tutti i canali e i BAG

REDEF	Effetto ed impiego del comando REDEF
Elementi del linguaggio NC	Elemento del linguaggio al quale deve essere abbinato un livello di protezione per l'esecuzione dell'istruzione: 1. Sottoprogrammi/funzioni predefiniti (lista con lo stesso nome). 2. Parola chiave "DO" - istruzione per azioni sincrone 3. Funzioni G (funzione G/condizioni del percorso) 4. Identificatore programma per ciclo Il ciclo deve essere creato in una delle directory per i cicli e deve contenere un'istruzione PROC.
Variabile di sistema	Variabile di sistema alla quale deve essere abbinato un livello di protezione in scrittura. L'accesso in lettura è sempre possibile. (vedere lista delle variabili di sistema).
Dato macchina/dato di setting	Dato macchina o dato di setting al quale deve essere abbinato un livello di protezione in scrittura/lettura.
APX	Parola chiave per livello di protezione
APW, APR	Eseguire scrivere, leggere
Valore	Valore numerico del livello di protezione (0 ... 7) da 0 (livello max.) a 7 (livello minimo)
valore 7	La posizione del selettore a chiave 0 corrisponde alla preimpostazione di tutte le istruzioni di programma eseguibili.

Esempio di richiamo di sottoprogrammi nei file di definizione

```

N10 REDEF GEOAX APX 3
N20 IF (ISFILE("/_N_CST_DIR/_N_SACCESS_SUB1_SPF"))
N30 PCALL /_N_CST_DIR/_N_SACCESS_SUB1_SPF
N40 ENDIF
N40 M17

```

Descrizione

Analogamente alle definizioni GUD, sono a disposizione file di definizione indipendenti che vengono interpretati all'avvio del controllo:

utente finale: /_N_DEF_DIR/_N_UACCESS_DEF

costruttore: /_N_DEF_DIR/_N_MACCESS_DEF

Siemens: /_N_DEF_DIR/_N_SACCESS_DEF

Richiamo di sottoprogrammi nei file di definizione

Nei file di definizione sopracitati è possibile anche richiamare sottoprogrammi che contengono le istruzioni REDEF. In genere le istruzioni REDEF devono essere inserite all'inizio del file come le istruzioni DEF. I sottoprogrammi devono contenere l'estensione SPF oppure MPF ed ereditano il livello di protezione del file di definizione impostato con \$MN_ACCESS_WRITE_xACCESS.

Nota

Ampliamento dell'istruzione REDEF

Non appena la funzione "Livelli di protezione per istruzioni del linguaggio NC" è attiva, tutte le ridefinizioni per dati macchina/dati setting impostate precedentemente nei file di definizione GUD, devono essere modificate nei nuovi file di definizione per l'assegnazione dei livelli di protezione, il settaggio dei livelli di protezione per dati macchina/dati setting quindi è solo più consentito nei suddetti file di definizione per l'assegnazione dei livelli di protezione, nei file di definizione GUD invece viene rifiutato con l'allarme 15420.

Nota

Il settaggio degli attributi di inizializzazione e di sincronizzazione è possibile sempre solo nei file di definizione dei GUD.

Livelli di protezione per variabili di sistema

I livelli di protezione per le variabili di sistema valgono solo per le assegnazioni di valori tramite istruzioni da partprogram. Nella superficie operativa vale il concetto del livello di protezione della rispettiva HMI-Advanced/Embedded.

Per ulteriori informazioni sul "Concetto dei livelli di protezione", consultare:
/BAD/, Manuale operativo HMI, al capitolo "Selettore a chiave" e "Dati macchina"
/IAD/, Manuale di messa in servizio, "Parametrizzazione del controllo"

3.5 REDEF: Modifica degli attributi degli elementi di linguaggio NC

Funzione

Le funzioni descritte nei sottocapitoli precedenti e dedicate alla definizione di oggetti di dati e all'impostazione dei livelli di protezione, con l'istruzione REDEF, vengono ridefinite con l'ampliamento dell'istruzione REDEF in un'unica interfaccia utilizzabile per l'impostazione di attributi e valori.

Programmazione

REDEF attributo valore per elemento del linguaggio NC

oppure

REDEF name (senza ulteriore indicazione del parametro)

Parametri

Elementi del linguaggio NC	Sono compresi: GUD Parametri R Dati macchina/dati di setting Variabili sincrone (\$AC_PARAM, \$AC_MARKER, \$AC_TIMER) Variabili di sistema scrivibili da partprogram (vedere PGA1) Frame utente (G500, etc.) Configurazioni magazzino/utensili	
name	Le impostazioni per APX, APR, APW vengono impostate ai valori predefiniti e INIPO, INIRE, INICF, PRLOC vengono ripristinati nuovamente.	
Attributo	Consentito per:	
Inizializzazioni		
INIPO	GUD, parametri R, variabili sincrone	
INIRE	GUD, parametri R, variabili sincrone	
INICF	GUD, parametri R, variabili sincrone	
PRLOC	Dati setting	
Sincronizzazione	Consentito per:	Impostazione di un valore di default:
SYNR	GUD	Stop preelaborazione in lettura
SYNW	GUD	Stop preelaborazione in scrittura
SYNRW	GUD	Stop preelaborazione in lettura/scrittura
Consenso all'accesso	Consentito per:	
APW	Dati macchina e dati setting	Diritto di accesso in scrittura
APR	Dati macchina e dati setting	Diritto di accesso in lettura
		Per i dati macchina e i dati setting si possono sovrascrivere in un secondo tempo i diritti di accesso predefiniti. I valori ammessi vanno da '0' (password Siemens) fino a '7' (posizione 0 del selettore a chiave)
Parametri opzionali		
valore (opzionale)	Parametri opzionali degli attributi INIPO, INIRE, INICF, PRLOC: successivo(i) , valore(i) iniziale(i) forme:	

Valore singolo	ad es. 5
Liste di valori	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9) per variabile con 10 elementi con w1: lista di valori da ripetere per variabili con più elementi, es. REP(12)
REP (w1)	oppure
SET(w1, w2, w3, ...)	Liste di valori
(w1, w2, w3, ...)	livello di protezione richiesto per parametri con attributi APR o APW
n:	Per i dati GUD , nella definizione si può indicare un valore iniziale (DEF NCK INT _MYGUD=5). Se questo valore iniziale non viene indicato (ad es. in DEF NCK INT _MYINT), si ha la possibilità di definirlo successivamente con l'istruzione REDEF. Il valore di inizializzazione per un campo vale per tutti gli elementi del campo. I singoli elementi possono essere impostati con una lista di inizializzazione oppure con REP (). Esempi: REDEF_MYGUD INIRE 5 REDEF_MYGUD INIRE 0,1,2,3,4,5,6,7,8,9 REDEF_MYGUD INIRE REP(12,14,16,18,20) Non utilizzabile per parametri R e variabili di sistema. Si possono assegnare solo delle costanti. Non sono ammesse espressioni come valore.

Significato degli attributi

INIPO	INIt con Power On I dati, in caso di riavviamento bufferizzato dell'NC, vengono sovrascritti con il/i valore/i di default.
INIRE	INIt con Reset da pannello operativo o fine partprogram (TP) Al termine di un programma principale ad es. con M2, M30 , ecc. o in caso di interruzione con Reset, i dati verranno sovrascritti con il valore di default. INIRE è attivo anche in INIPO.
INICF	INIt con richiesta NewConf o nell'istruzione NEWCONF da TP Nella richiesta NewConf o nell'istruzione NEWCONF da TP i dati vengono sovrascritti con il valore di default. INICF è attivo anche in INIRE e INIPO.
PRLOC	Solo modifica programm locale Se viene modificato un dato in un partprogram, in un sottoprogramma, in un ciclo o in un ASUP, dopo la fine del programma principale (fine ad es. con M2, M30, ecc. o in caso di interruzione con reset da pannello operativo) verrà ripristinato il suo valore originario. Questo attributo è consentito solo per i dati setting programmabili, vedere dati setting programmabili.

La **sincronizzazione** degli eventi che richiedono l'inizializzazione deve essere eseguita dall'operatore. Questo vuol dire che se un partprogram viene eseguito ad es. in **due diversi canali**, le variabili vengono inizializzate in ognuno dei due canali. Questo ha ripercussioni sui dati globali o sui dati assiali!

Dati setting programmabili e variabili di sistema scrivibili da partprogram

I seguenti DS possono essere inizializzati in abbinamento all'istruzione REDEF:

Numero	identificatore	GCODE
42000	\$SC_THREAD_START_ANGLE	SF
42010	\$SC_THREAD_RAMP_DISP	DITS/DITE
42400	\$SA_PUNCH_DWELLTIME	PDELAYON
42800	\$SA_SPIND_ASSIGN_TAB	SETMS
43210	\$SA_SPIND_MIN_VELO_G25	G25
43220	\$SA_SPIND_MAX_VELO_G26	G26
43230	\$SA_SPIND_MAX_VELO_LIMS	LIMS
43300	\$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
43420	\$SA_WORKAREA_LIMIT_PLUS	G26
43430	\$SA_WORKAREA_LIMIT_MINUS	G25
43510	\$SA_FIXED_STOP_TORQUE	FXST
43520	\$SA_FIXED_STOP_WINDOW	FXSW
43700	\$SA_OSCILL_REVERSE_POS1	OSP1
43710	\$SA_OSCILL_REVERSE_POS2	OSP2
43720	\$SA_OSCILL_DWELL_TIME1	OST1
43730	\$SA_OSCILL_DWELL_TIME2	OST2
43740	\$SA_OSCILL_VELO	FA
43750	\$SA_OSCILL_NUM_SPARK_CYCLES	OSNSC
43760	\$SA_OSCILL_END_POS	OSE
43770	\$SA_OSCILL_CTRL_MASK	OSCTRL
43780	\$SA_OSCILL_IS_ACTIVE	OS
43790	\$SA_OSCILL_START_POS	OSB

Nel PGA1 "Lista delle variabili di sistema" è disponibile un elenco delle variabili di sistema. Tutte le variabili di sistema che nella colonna Partprogram sono contrassegnate con W (write) o WS (write con stop preelaborazione), possono essere inizializzate con l'istruzione RESET.

Esempio

```
Comportamento in fase di reset dei GUD:
/_N_DEF_DIR/_N_SGUD_DEF
DEF NCK INT _MYGUD1                                ;Definizioni
DEF NCK INT _MYGUD2 = 2
DEF NCK INT _MYGUD3 = 3
Inizializzazione con reset da pannello operativo/fine partprogram:
DEF _MYGUD2 INIRE                                    ;Inizializzazione
M17
```

In questo modo con reset da pannello operativo/fine partprogram "_MYGUD2" verrà reimpostata sul valore "2" mentre "_MYGUD1" e "_MYGUD3" manterranno il loro valore.

Esempio di limitazione modale della velocità nel partprogram (dato setting)

```

/_N_DEF_DIR/_N_SGUD_DEF
REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC      ;Dato setting per la limitazione di velocità
M17

```

```

/_N_MPF_DIR/_N_MY_MPF
N10 SETMS (3)
N20 G96 S100 LIMS=2500
...
M30

```

Supponendo che la velocità limite impostata nel dato setting (\$SA_SPIND_MAX_VELO_LIMS) sia di 1200 giri/min. Dato che però in un partprogram generato e collaudato si può consentire una velocità superiore, qui si programma LIMS=2500. Dopo la fine del programma tuttavia tornerà ad essere attivo il valore che è stato programmato nel dato setting.

Impostazione dei valori predefiniti e cancellazione delle inizializzazioni

Nuova definizione	Attributo	Valori predefiniti e ripristino delle inizializzazioni
REDEF	Elementi del linguaggio NC	APX = 7
REDEF	Dati macchina/dati di setting	Ripristinare APW = 7 APR = 7 PRLOC
REDEF	Variabile di sincronizzazione	Ripristinare APW = 7 INIRE, INIPO, INICF
REDEF	GUD, LUD	Ripristinare INIRE, INIPO, INICF
Esempio		
REDEF MASLON APX 2		
REDEF SYG RS INIRE APW3		
REDEF R[] INIRE		
REDEF MASLON		;APX impostato su 7
REDEF SYG RS		;APW impostato su 7 e INIRE cancellato
REDEF R[]		;INIRE cancellato

Condizioni marginali

- La **modifica** degli attributi di oggetti NC può avvenire soltanto **dopo la definizione** dell'oggetto. In particolare per i GUD si deve fare attenzione alla sequenza DEF.../ REDEF. (i dati setting/le variabili di sistema vengono impostati implicitamente già prima di elaborare i file di definizione). È necessario sempre definire per prima cosa il simbolo (in modo implicito da sistema o con l'istruzione DEF) e solo successivamente si potrà eseguire una modifica con REDEF.
- Se si programmano più modifiche degli attributi contemporaneamente sarà sempre attiva solo l'ultima modifica.
- Gli **attributi relativi ai campi** non possono essere impostati per i singoli elementi ma sempre soltanto **per l'intero campo**:

```

DEF CHAN INT _MYGUD[10,10]
REDEF _MYGUD INIRE          // ok

```

```
REDEF _MYGUD[1,1] INIRE // non possibile, viene emesso un allarme
                        // (valore del campo)
```

- Anche l'inizializzazione dei **campi GUD** non viene influenzata.

```
DEF NCK INT _MYGUD[10] =(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
DEF NCK INT _MYGUD[100,100] = REP (12)
DEF NCK INT _MYGUD[100,100] ;
```

- Le istruzioni REDEF con **parametri R** devono essere indicate con parentesi.

```
REDEF R[ ] INIRE
```

- **Attributi INI**
Si dovrà tuttavia fare attenzione che, quando si impostano gli attributi INI per queste variabili, sia disponibile una **memoria sufficiente per i valori di inizializzazione**, impostabile con il DM 18150: MM_GUD_VAL_MEM. Nel DM 11270: DEFAULT_VALUES_MEM_MASK deve essere settato il bit1 = 1 (memoria attiva per i valori di inizializzazione). Una memoria insufficiente provoca l'emissione dell'allarme 12261 "Inizializzazione non consentita"
- **Parametri R e variabili di sistema**
Per le variabili di sistema e per i parametri R non è possibile indicare un valore di default diverso dal valore impostato. È possibile invece il reset sul valore impostato con INIPO, INIRE o INICF.
- Anche per il **tipo di dati FRAME delle GUD** non si può indicare (come già per la definizione del dato) un valore di default diverso dal valore impostato.
- **GUD (DEF NCK INT_MYGUD)**
Per i GUD globali (DEF NCK INT_MYGUD) è consentito solo l'attributo INIPO.
Per i GUD specifici per canale (DEF CHAN INT_MYGUD), in caso di risultato corrispondente (RESET, BAG-RESET o NewConfig) vengono inizializzati solo i dati nel relativo canale.
Esempio: Sono definiti due canali con GUD specifico per canale che deve essere inizializzato al momento del RESET:
DEF CHAN INT_MYGUD
REDEF _MYGUD INIRE
Al RESET nel primo canale il GUD per questo canale viene ripristinato, il valore nel secondo canale resta invariato.

Impostazione di un valore di default

Se con REDEF <name> INIRE, INIPO; INICF; PRLOC viene modificato il comportamento di una variabile di sistema oppure GUD, deve essere settato il dato macchina DM 11270: DEFAULT_VALUES_MEM_MASK = 1 (memoria attiva per i valori di inizializzazione). In caso contrario viene emesso l'allarme 12261 "Inizializzazione non consentita".

3.6 Istruzione di strutturazione SEFORM nell'editor Step

Funzione

L'istruzione SEFORM viene interpretata nell'editor Step, per potere generare la vista del passo per HMI-Advanced. La vista del passo è disponibile nell'HMI-Advanced e serve per migliorare la leggibilità del sottoprogramma NC. Con l'istruzione di strutturazione SEFORM l'editor Step (supporto al programma basato sull'editor) viene supportato dai tre parametri indicati.

Programmazione

```
SEFORM (STRING[128] nome passo, INT livello, STRING[128] icona)
```

Parametri

SEFORM

nome sezione

piano

icona

Richiamo funzionale dell'istruzione di strutturazione con i parametri: nome della sezione, livello e icona

Identificatore del passo di lavorazione

Indice per livello principale o secondario.

=0 corrisponde al livello principale

=1, ... corrisponde al livello secondario da 1 a n

Nome dell'icona da visualizzare per questo passo di lavorazione.

Nota

Le istruzioni SEFORM vengono generate nell'editor Step.

La stringa trasmessa con il parametro <nome sezione>, analogamente all'istruzione MSG, viene inserita in sincronismo con il ciclo principale nella variabile BTSS. L'informazione resta memorizzata fino a quando viene sovrascritta dalla successiva istruzione SEFROM. Con Reset e alla fine del partprogram il contenuto viene cancellato.

I parametri Livello e Icona vengono verificati dall'NCK nell'elaborazione del partprogram ma non vengono rielaborati.

Per ulteriori informazioni sul supporto alla programmazione basato sull'editor si prega di consultare la documentazione:
/BAD/ Manuale d'uso HMI Advanced.

Settori di protezioni

4.1 Definizione dei settori di protezione (CROTDEF, NROTDEF)

Funzione

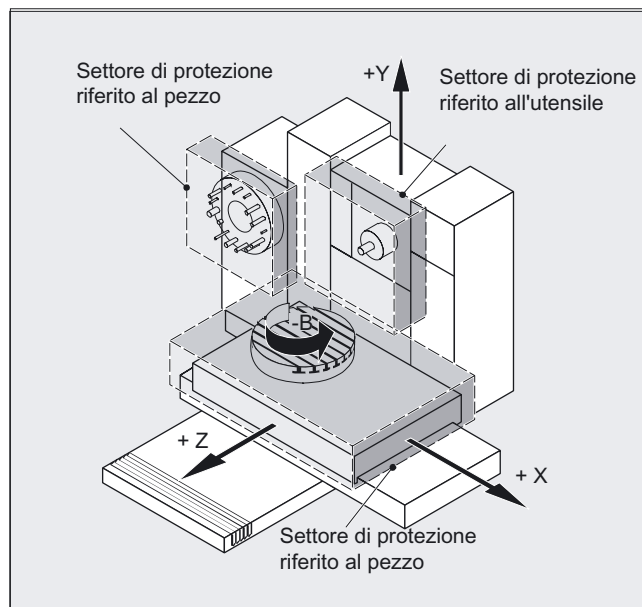
Con l'ausilio dei settori di protezione si possono proteggere i diversi elementi di una macchina da movimenti indesiderati, dall'attrezzatura al pezzo.

Settori di protezione **referiti all'utensile**:

per elementi che appartengono all'utensile (es. utensile, portautensile).

Settori di protezione **referiti al pezzo**:

per elementi che appartengono al pezzo (es. parti del pezzo, tavola portapezzo, griffe di bloccaggio, autocentrante, contropunta).



Programmazione

```
DEF INT NOT_USED
CPROTDEF (n, t, applim, appplus, appminus)
NPROTDEF (n, t, applim, appplus, appminus)
EXECUTE (NOT_USED)
```

Parametri

DEF INT NOT_USED	Definizione di variabili locali, tipo di dati Integer (confr. cap.10 Azioni sincrone al movimento)
CPROTDEF	Definizione dei settori di protezione specifici per canale (solo per NCU 572/573)
NPROTDEF	Definizione dei settori di protezione specifici della macchina
EXECUTE	Terminare la definizione
n	Numero del settore di protezione definito
t	TRUE = settore di protezione riferito all' utensile FALSE = settore di protezione riferito al pezzo
applim	Tipo della limitazione nella 3. dimensione 0 = nessuna limitazione 1 = limitazione in direzione positiva 2 = limitazione in direzione negativa 3 = limitazione in direzione positiva e negativa
appplus	Valore della limitazione in direzione positiva della 3. dimensione
appminus	Valore della limitazione in direzione negativa della 3a dimensione
NOT_USED	La variabile di errore non ha alcun significato per i settori di protezione con EXECUTE

Descrizione

Fanno parte della definizione dei settori di protezione:

- CPROTDEF per settori di protezione specifici per canale
- NPROTDEF per settori di protezione specifici per la macchina
- Descrizione del profilo del settore di protezione
- Fine della definizione con EXECUTE

Attivando il settore di protezione nel partprogram NC, si può spostare il punto di riferimento in modo relativo.

Punto di riferimento della descrizione del profilo

I settori di protezione riferiti al pezzo vengono definiti nel sistema di coordinate di base. I settori di protezione riferiti all'utensile vengono indicati rispetto al punto di riferimento F del portautensile.

Descrizione del profilo dei settori di protezione

Il profilo dei settori di protezione viene impostato con al massimo 11 movimenti nel piano indicato. In questo caso, il primo movimento è quello sul profilo. Come settore di protezione viene considerato quindi quello alla sinistra del profilo. I movimenti situati tra CPROTDEF oppure NPROTDEF e EXECUTE non vengono eseguiti ma servono a definire il settore di protezione.

Piano di lavoro

Il piano desiderato viene selezionato prima di CPROTDEF o NPROTDEF con G17, G18, G19 e non può essere modificato prima di EXECUTE. Non è consentita una programmazione delle applicate tra CPROTDEF o NPROTDEF e EXECUTE.

Elementi del profilo

Sono consentiti:

- G0, G1 per elementi di profilo rettilinei
- G2 per tratti circolari in senso orario (solo per settori di protezione riferiti al pezzo)
- G3 per tratti circolari in senso antiorario

Nota

Per l'810D sono possibili al massimo 4 elementi di profilo per la definizione del rispettivo settore di protezione (max. 4 settori di protezione specifici per canale e 4 specifici per NCK).

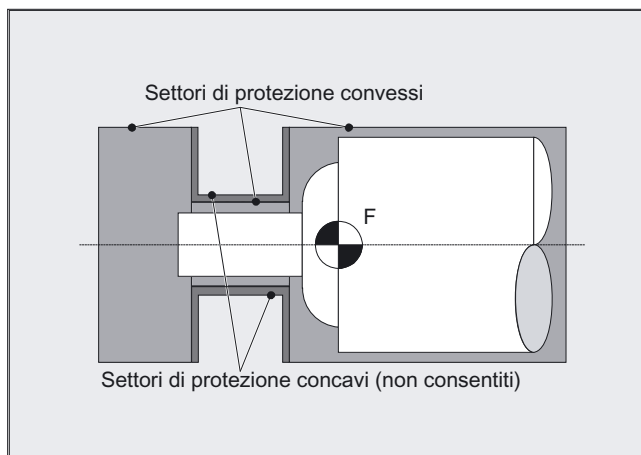
Se il settore di protezione deve descrivere un cerchio completo, è necessario suddividerlo in due semicerchi. La sequenza G2, G3 o G3, G2 non è consentita. Eventualmente occorre inserire un breve blocco con G1.

L'ultimo punto della descrizione del profilo deve coincidere con il primo punto.

I **settori di protezione esterni** (possibile solo per settori di protezione riferiti al pezzo) devono essere definiti in **senso orario**.

Con settori di protezione **simmetrici alla rotazione** (es.autocentrante) si deve descrivere il **profilo globale** (non solo fino al centro di rotazione!).

I settori di protezione **riferiti all'utensile** devono essere sempre **convessi**. Nel caso sia necessario un settore di protezione concavo, esso deve essere realizzato con più settori convessi.



Durante la definizione dei settori di protezione non possono essere attivi

- correttori raggio fresa o raggio utensile
- trasformazioni
- o frame.

Inoltre non si possono programmare la ricerca del punto di riferimento (G74), l'accostamento a un punto fisso (G75), il blocco preelaborazione oppure un fine programma.

4.2 Attivazione e disattivazione dei settori di protezione (CPROT, NPROT)

Funzione

Attivazione dei settori di protezione precedentemente definiti per la sorveglianza anticollisione, preattivazione o disattivazione dei settori di protezione attivi.

Il numero massimo dei settori di protezione attivi in un canale viene impostato tramite i dati macchina.

Se non è attivo alcun settore di protezione riferito all'utensile, il percorso dell'utensile viene verificato rispetto al settore di protezione riferito al pezzo.

Nota

Se non è attivo alcun settore di protezione riferito al pezzo, non avviene alcuna sorveglianza del settore di protezione.

Programmazione

CPROT (n, state, xMov, yMov, zMov)

NPROT (n, state, xMov, yMov, zMov)

Parametri

CPROT

Richiamo dei settori di protezione specifici per canale (solo per NCU 572/573)

NProt	Richiamo del settore di protezione specifico per la macchina
n	Numero del settore di protezione
state	Indicazione dello stato
	0 = disattivazione del settore di protezione
	1 = preattivazione del settore di protezione
	2 = attivazione del settore di protezione
	3 = preattivazione del settore di protezione con stop condizionato
xMov, yMov, zMov	Traslazione lungo gli assi geometrici del settore di protezione già definito

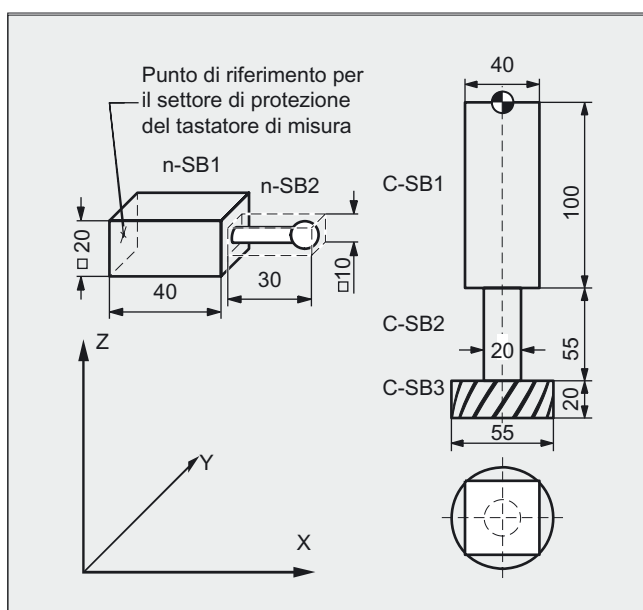
Esempio fresatura

In una fresatrice deve essere sorvegliata una possibile collisione della fresa con il tastatore di misura. La posizione del tastatore di misura deve essere impostata all'attivazione con una traslazione. Per questo motivo vengono definiti i seguenti settori di protezione:

- un settore di protezione specifico per la macchina e riferito al pezzo per il porta tastatore di misura (n-SB1) e per il tastatore di misura stesso (n-SB2).
- un settore di protezione specifico per il canale e riferito all'utensile per il porta fresa (c-SB1), per il codolo (c-SB2) e per la fresa stessa (c-SB3).

L'orientamento di tutti i settori di protezione avviene in direzione di Z.

La posizione del punto di riferimento del tastatore di misura all'attivazione deve trovarsi in $X = -120$, $Y = 60$ e $Z = 80$.



DEF INT SCHUTZB	Definizione di una variabile ausiliaria
Definizione dei settori di protezione	Impostazione dell'orientamento
G17	

NPROTDEF (1, FALSE, 3, 10, -10) G01 X0 Y-10 X40 Y10 X0 Y-10 EXECUTE (SCHUTZB)	Settore di protezione n-SB1
NPROTDEF (2, FALSE, 3, 5, -5) G01 X40 Y-5 X70 Y5 X40 Y-5 EXECUTE (SCHUTZB)	Settore di protezione n-SB2
CPROTDEF (1, TRUE, 3, 0, -100) G01 X-20 Y-20 X20 Y20 X-20 Y-20 EXECUTE (SCHUTZB)	Settore di protezione c-SB1
CPROTDEF (2, TRUE, 3, -100, -150) G01 X0 Y-10 G03 X0 Y10 J10 X0 Y-10 J-10 EXECUTE (SCHUTZB)	Settore di protezione c-SB2
CPROTDEF (3, TRUE, 3, -150, -170) G01 X0 Y-27,5 G03 X0 Y27,5 J27,5 X0 Y27,5 J-27,5 EXECUTE (SCHUTZB)	Settore di protezione c-SB3
Attivazione dei settori di protezione:	
NPROT (1, 2, -120, 60, 80)	Attivazione del settore di protezione n-SB1 con traslazione
NPROT (2, 2, -120, 60, 80)	Attivazione del settore di protezione n-SB2 con traslazione
CPROT (1, 2, 0, 0, 0)	Attivazione del settore di protezione c-SB1 con traslazione
CPROT (2, 2, 0, 0, 0)	Attivazione del settore di protezione c-SB2 con traslazione
CPROT (3, 2, 0, 0, 0)	Attivazione del settore di protezione c-SB3 con traslazione

Stato di attivazione

Un settore di protezione normalmente viene attivato nel partprogram con lo stato = 2.

Lo stato è sempre specifico per canale anche per i settori di protezione riferiti alla macchina.

Se è previsto che un settore di protezione debba essere attivato dal programma PLC utente, la relativa preattivazione necessaria, avviene con lo stato = 1.

La disattivazione, e conseguentemente l'esclusione dei settori di protezione, avviene con lo stato = 0. Non è necessaria alcuna traslazione.

Traslazione dei settori di protezione durante l'attivazione/preattivazione

La traslazione può avvenire in 1, 2 o 3 dimensioni. L'impostazione della traslazione si riferisce al:

- punto zero macchina per i settori di protezione specifici del pezzo,
- punto di riferimento F del portautensile per i settori di protezione specifici dell'utensile.

Stato dopo l'avviamento

I settori di protezione si possono attivare già dopo aver eseguito l'avviamento e la ricerca del punto di riferimento. Per questo motivo si devono settare le variabili di sistema

`$SN_PA_ACTIV_IMMED [n]` oppure

`$SN_PA_ACTIV_IMMED [n] = TRUE.`

Esse vengono subito settate con lo stato = 2 e non hanno alcuna traslazione.

Attivazione multipla dei settori di protezione

Un settore di protezione può essere attivo contemporaneamente anche in diversi canali (es. contropunta di due slitte contrapposte). La sorveglianza dei settori di protezione avviene solo quando tutti gli assi geometrici sono stati riferiti. In questo caso vale quanto segue:

- il settore di protezione non può essere attivato nello stesso canale contemporaneamente con diverse traslazioni.
- il settori di protezione riferiti alla macchina devono avere lo stesso orientamento in entrambi i canali.

4.3 Verifica del superamento del settore di protezione, limitazione del campo di lavoro e finecorsa software

Funzione

La funzione CALCPOSI serve a verificare se, a partire da un determinato punto iniziale, gli assi geometrici possono eseguire un percorso impostato senza che vengano superati i limiti degli assi stessi (limiti software), le limitazioni del campo di lavoro o i settori di protezione.

Se il percorso impostato non può essere eseguito, viene restituito il valore massimo consentito.

La funzione CALCPOSI è un sottoprogramma predefinito. Essa deve essere programmata in un blocco a sè stante.

Programmazione

```
Stato=CALCPOSI (_STARTPOS, _MOVDIST, _DLIMIT, _MAXDIST, _BASE_SYS,
_TESTLIM)
```

Parametri

stato

0: Funzione O.K.,
il percorso impostato può essere eseguito completamente.
-: In _DLIMIT è presente almeno una componente negativa
-: Si è verificato un errore di calcolo in una trasformazione

Se il percorso impostato non può essere eseguito completamente, viene restituito un valore positivo codificato in decimale:

Posizione delle unità (tipo di limite superato):

- 1: limiti software che delimitano il percorso di spostamento.
- 2: limitazione del campo di lavoro che delimita il percorso di spostamento.
- 3: settori di protezione che delimitano il percorso di spostamento.

Se vengono superati più limiti contemporaneamente (es. limiti software e settori di protezione), nella posizione delle unità viene segnalato quello che provoca la limitazione maggiore del movimento.

Posizione delle decine:

10:

il valore iniziale supera il limite

20:

La retta indicata supera il limite. Questo valore viene anche restituito quando il punto finale stesso non supera nessun limite però, nel tratto dal punto iniziale a quello finale, potrebbe verificarsi un superamento del valore limite (es. superamento di un settore di protezione, limite software curvilineo nel SCP con trasformazioni non lineari, es. Transmit).

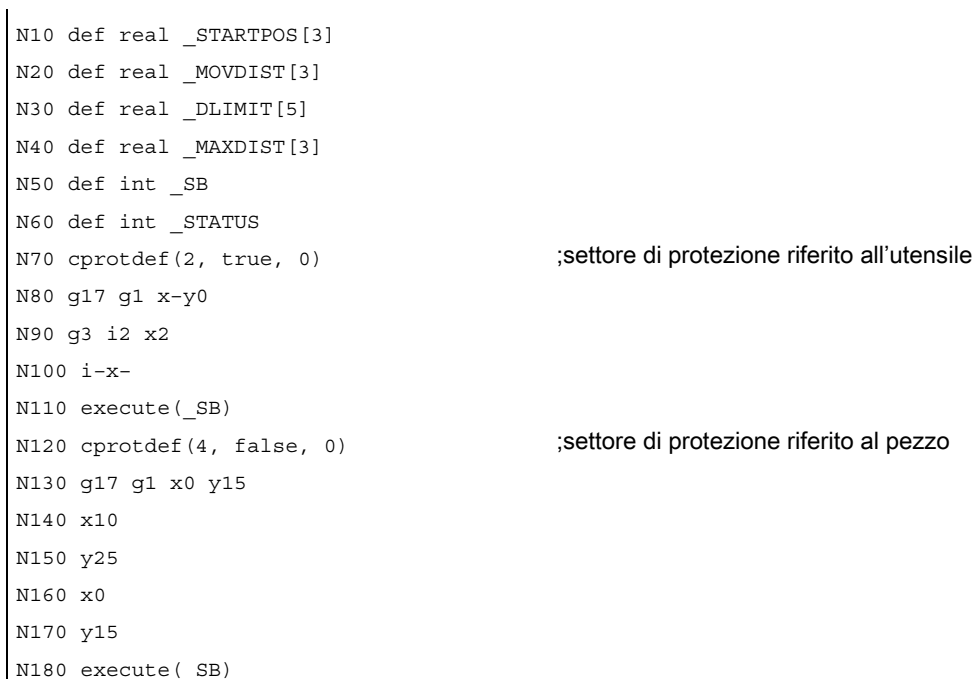
4.3 Verifica del superamento del settore di protezione, limitazione del campo di lavoro e finecorsa software

	Posizione delle centinaia: 100: il valore limite positivo è stato superato (solo se la posizione delle unità contiene 1 oppure 2, cioè in caso di limiti software o di limitazione del campo di lavoro) 100: È stato superato un settore di protezione specifico dell'NCK (solo se la posizione delle unità contiene 3). 200: il valore limite negativo è stato superato (solo se la posizione delle unità contiene 1 oppure 2, cioè in caso di limiti software o di limitazione del campo di lavoro) 200: È stato superato un settore di protezione specifico per canale (solo se la posizione delle unità contiene 3). Posizione delle migliaia 1000: fattore per il quale viene moltiplicato il numero dell'asse che ha superato il limite (solo se la posizione delle unità contiene 1 oppure 2, cioè in caso di limiti software o di limitazione del campo di lavoro). La numerazione degli assi inizia da 1 e si riferisce, in caso di limiti software superati (unità = 1) agli assi macchina e in caso di superamento della limitazione del campo di lavoro (unità = 2) agli assi geometrici. 1000: fattore per il quale viene moltiplicato il numero del settore di protezione superato (solo se la posizione delle unità contiene 3). Se vengono superati più settori di protezione, nella posizione delle centinaia e delle migliaia viene segnalato il settore di protezione che ha provocato un superamento più grave del percorso impostato. Valore iniziale per le ascisse [0], ordinate [1] e applicate [2] nel (SCP) Valore incrementale per le ascisse [0], ordinate [1] e applicate [2] [0] - [2]: Distanze minime abbinate agli assi geometrici. [3]: Distanza minima che viene abbinata ad un asse macchina lineare con una trasformazione non lineare quando non può essere abbinato nessun asse geometrico in modo univoco. [4]: Distanza minima che viene abbinata ad un asse macchina rotante con una trasformazione non lineare quando non può essere abbinato nessun asse geometrico in modo univoco. Solo per trasformazioni speciali quando devono essere sorvegliati limiti SW. Campo [0] – [2] per il valore restituito. Percorso incrementale in tutti e tre gli assi geometrici senza che la distanza minima impostata venga superata (in negativo) da un limite assiale negli assi macchina interessati. Se il percorso non è limitato, il contenuto di questo parametro è identico a quello di _MOVDIST.
_STARTPOS	
_MOVEDIST	
_DLIMIT	
_MAXDIST	

_BASE_SYS	FALSE oppure parametro non indicato: durante la valutazione delle impostazioni di posizione e lunghezza viene interpretato il codice G del gruppo 13 (G70, G71, G700, G710; inch/metrico). Con G70 attivo e sistema di base metrico (o con G71 attivo e sistema di base in inch) vengono trasferite al sistema base le variabili di sistema \$AA_IW[X] e \$AA_MW[X] riferite al SCP che devono essere eventualmente ricalcolate con la funzione CALCPOSI. TRUE: durante la valutazione delle impostazioni di posizione e lunghezza viene sempre utilizzato il sistema di base del controllo indipendentemente dal valore del codice G attivo del gruppo 13.
_TESTLIM	Limitazioni da verificare (codificate in binario): 1: sorveglianza limiti software 2: sorveglianza delle limitazioni del campo di lavoro 3: sorveglianza dei settori di protezione attivati 4: sorveglianza dei settori di protezione pre-attivati combinazioni tramite somma dei valori. Default: 15; verificare tutto.

Esempio

Nell'esempio (figura) nelle X vengono riportati i limiti software e la limitazione del campo di lavoro. Inoltre sono stati definiti tre settori di protezione, entrambi i settori di protezione specifici per canale C2 e C4 ed il settore di protezione specifico NCK N3. C2 è un settore di protezione attivo di forma circolare riferito all'utensile con un raggio di 2 mm. C4 è un settore di protezione di forma quadratica preattivato e riferito al pezzo con una lunghezza laterale di 10 mm mentre N3 è di forma rettangolare con lati di 10 mm e 15 mm. Nel programma NC seguente vengono definiti innanzitutto i settori di protezione e le limitazioni dei campi di lavoro come dallo schizzo e successivamente avviene il richiamo della funzione CALCPOSI con diverse parametrizzazioni. I risultati dei singoli richiami di CALCPOSI vengono riepilogati nella tabella alla fine dell'esempio.



```

N190 nprotdef(3, false, 0) ;settore di protezione riferito alla macchina
N200 g17 g1 x10 y5
N210 x25
N220 y15
N230 x10
N240 y5
N250 execute(_SB)
N260 cprot(2,2,0, 0, 0) ;Attivazione o preattivazione dei
N270 cprot(4,1,0, 0, 0) ;settori di protezione
N280 nprot(3,2,0, 0, 0)
N290 g25 XX=-YY=- ;definizione delle limitazioni dei campi di lavoro
N300 g26 xx= 20 yy= 21
N310 _STARTPOS[0] = 0.
N320 _STARTPOS[1] = 0.
N330 _STARTPOS[2] = 0.
N340 _MOVDIST[0] = 35.
N350 _MOVDIST[1] = 20.
N360 _MOVDIST[2] = 0.
N370 _DLIMIT[0] = 0.
N380 _DLIMIT[1] = 0.
N390 _DLIMIT[2] = 0.
N400 _DLIMIT[3] = 0.
N410 _DLIMIT[4] = 0.
; diversi richiami della funzione
N420 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)
N430 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,3)
N440 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,1)
N450 _STARTPOS[0] = 5. ;altro punto iniziale
N460 _STARTPOS[1] = 17.
N470 _STARTPOS[2] = 0.
N480 _MOVDIST[0] = 0. ;altra destinazione
N490 _MOVDIST[1] = -.
N500 _MOVDIST[2] = 0.
; diversi richiami della funzione
N510 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,14)
N520 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)
N530 _DLIMIT[1] = 2.
N540 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)
N550 _STARTPOS[0] = 27.
N560 _STARTPOS[1] = 17.1
N570 _STARTPOS[2] = 0.
N580 _MOVDIST[0] = -.
N590 _MOVDIST[1] = 0.
N600 _MOVDIST[2] = 0.

```

4.3 Verifica del superamento del settore di protezione, limitazione del campo di lavoro e fincorsa software

```

N610 _DLIMIT[3] = 2.
N620 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 12)
N630 _STARTPOS[0] = 0.
N640 _STARTPOS[1] = 0.
N650 _STARTPOS[2] = 0.
N660 _MOVDIST[0] = 0.
N670 _MOVDIST[1] = 30.
N680 _MOVDIST[2] = 0.
N690 trans x10
N700 arot z45
N710 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)
N720 M30

```

Risultati delle verifiche nell'esempio:

Nr. blocco N...	_STATUS	_MAXDIST [0] (= X)	_MAXDIST [1] (= Y)	Annotazioni
420	3123	8.040	4.594	Il settore di protezione SB N3 viene superato.
430	1122	20.000	11.429	Nessuna sorveglianza del settore di protezione, superamento della limitazione del campo di lavoro.
440	1121	30.000	17.143	È attiva solo più la sorveglianza dei limiti software.
510	4213	0.000	0.000	Punto iniziale superato SB C4
520	0000	0.000	-0.000	Il settore di protezione SB C4 preattivo non viene sorvegliato. Il percorso impostato può essere eseguito completamente.
540	2222	0.000	-0.000	A causa di _DLIMIT[1]=2 il percorso di movimento viene ridotto dalla limitazione del campo di lavoro.
620	4223	-0.000	0.000	La distanza totale verso C4, a causa di C2 e _DLIMIT[3], è di 4 mm. La distanza C2 - N3 di 0.1 mm non comporta una limitazione del percorso di movimento.
710	1221	0.000	21.213	Frame attivo con traslazione e rotazione. Il percorso di movimento in _MOVDIST vale nel sistema di coordinate (SCP) traslato e ruotato.

Casi speciali ed ulteriori dettagli

Tutte le impostazioni del percorso si intendono in quote radiali, anche per assi radiali con codice G "DIAMON" attivo. Se il percorso di uno degli assi interessati non può essere eseguito completamente, nel valore restituito _MAXDIST vengono ridotti proporzionalmente anche i percorsi degli altri assi in modo che il punto finale risultante si trovi sulla traiettoria impostata.

È consentito che per uno o più degli assi interessati non vengano definiti limiti software, limitazioni del campo di lavoro oppure settori di protezione. I limiti vengono solo verificati quando i relativi assi hanno eseguito la ricerca del punto di riferimento. Se gli stessi non sono stati dichiarati come assi modulo.

Le sorveglianze dei limiti software e delle limitazioni dei campi di lavoro dipendono, come durante il normale funzionamento, dalle impostazioni attive (segnali di interfaccia per la selezione dei limiti software 1 o 2, codice G WALIMON / WALIMOF, dati setting per l'attivazione individuale delle limitazioni dei campi di lavoro e per determinare se nella sorveglianza dei campi di lavoro deve essere o meno considerato il raggio dell'utensile attivo).

Con determinate trasformazioni cinematiche (ad es. TRANSMIT) non è possibile determinare in modo univoco la posizione degli assi macchina dalle posizioni del sistema di coordinate pezzo SCP (ambiguità). Durante il normale funzionamento, l'univocità deriva normalmente dall'antefatto e dalla condizione che, un movimento continuo nel SCP deve corrispondere ad un movimento degli assi macchina. Nella sorveglianza dei limiti software con l'ausilio della funzione CALCPOSI, per ovviare all'ambiguità in questi casi viene utilizzata perciò la posizione attuale degli assi macchina. Eventualmente prima di CALCPOSI deve essere programmato uno **STOPRE** per poter utilizzare la funzione con posizioni degli assi macchina valide.

Durante il movimento nel percorso impostato, dai settori di protezione non viene assicurato il mantenimento della distanza specificata in _DLIMIT[3]. Per questo al prolungamento del punto finale restituito in _MOVDIST, nessun settore di protezione viene superato di questa distanza. La retta però, nel suo andamento, può passare senza limitazioni davanti ad un settore di protezione.

Nota

Per maggiori informazioni sulle limitazioni del campo di lavoro vedere
/PG/ Manuale di programmazione, Concetti fondamentali,

sui limiti software vedere
/FB1/ Manuale delle funzioni di base; Sorveglianze assi, Settori di protezione (A3).

Istruzioni di percorso speciali

5.1 Raggiungimento di posizioni codificate (CAC, CIC, CDC, CACP, CACN)

Funzione

Tramite dati macchina è possibile definire per 2 assi, nelle tabelle delle posizioni, fino a 60 posizioni (0...59) per ogni asse.

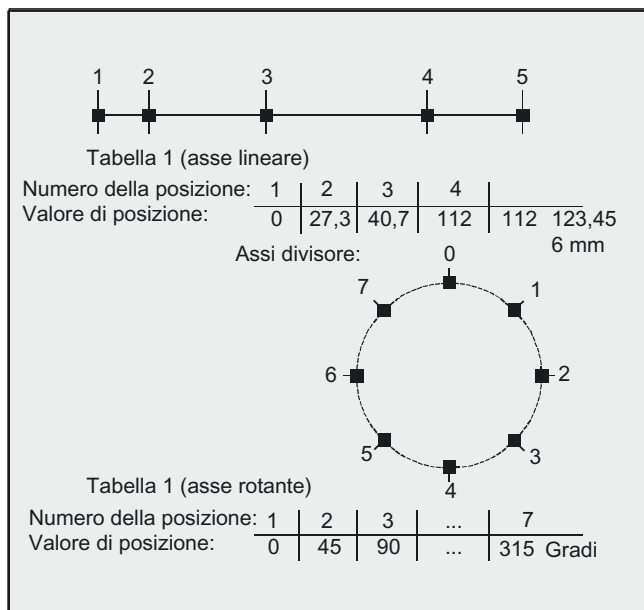
Programmazione

CAC (n)
oppure
CIC (n)
oppure
CACP (n)
oppure
CACN (n)

Parametri

CAC (n)	Raggiungimento della posizione assoluta codificata
CIC (n)	Raggiungimento della posizione incrementale codificata di n posti avanti (+) o indietro (-)
CDC (n)	Raggiungimento della posizione codificata attraverso il percorso più breve (solo per assi rotanti)
CACP (n)	Raggiungimento della posizione assoluta codificata in direzione positiva (solo per assi rotanti)
CACN (n)	Raggiungimento della posizione assoluta codificata in direzione negativa (solo per assi rotanti)
(n)	Numero di posizione 1, 2, ...max. 60 posizioni per ogni asse

Esempio di tabella delle posizioni per asse lineare e rotante

**Nota**

Se l'asse si trova tra due posizioni, programmando l'istruzione incrementale CIC(...) non si ha alcun movimento. Si consiglia di programmare il primo movimento sempre con l'indicazione assoluta della posizione.

Esempio 2

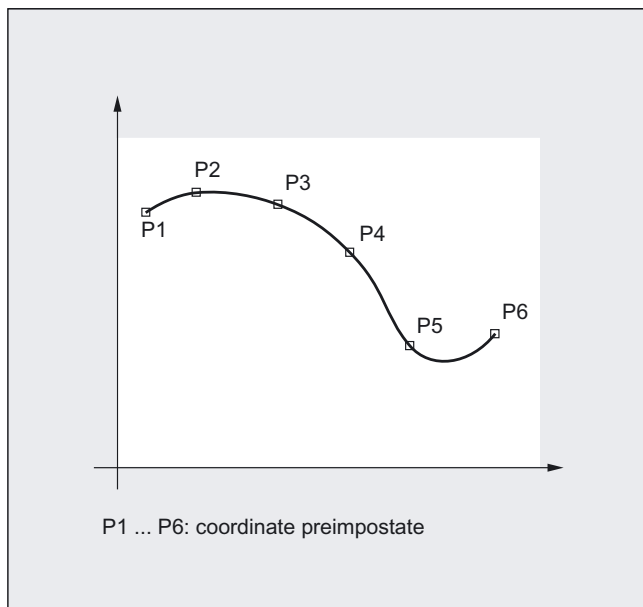
N10	FA[B] = 300		;avanzamento per l'asse di posizionamento B
N20	POS[B] = CAC	(10)	;raggiungimento della posizione codificata 10 (assoluta)
N30	POS[B] = CIC	(-4)	;spostarsi di 4 posti indietro rispetto alla posizione attuale

5.2 Interpolazione Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN)

Funzione

Con l'interpolazione spline è possibile collegare successioni di punti mediante curve. La spline può essere utilizzata ad esempio per trasformare in curve i punti ottenuti con la digitalizzazione.

Ci sono diversi tipi di spline con differenti caratteristiche e quindi anche differenti risultati. Oltre alla scelta del tipo di spline, l'utente ha la possibilità di influenzare una serie di parametri. Spesso sono necessari alcuni tentativi prima di ottenere la figura desiderata.



Una spline viene programmata quando si vuole unire con una curva una serie di punti. Sono possibili tre tipi di spline:

- A-Spline (Akima-Spline)
- B-spline (NURBS not uniform, rational Basis-Spline)
- C-Spline (Spline cubica)

Programmazione

ASPLINE X Y Z A B C

oppure

BSPLINE X Y Z A B C

oppure

CSPLINE X Y Z A B C

Parametri

ASPLINE	La Akima-Spline passa in modo tangenziale per i punti di appoggio programmati.
BSPLINE	La B-Spline non passa direttamente attraverso i punti di controllo ma immediatamente nelle vicinanze. Le posizioni programmate non sono punti di appoggio ma punti di controllo.
CSPLINE	Spline cubica con raccordi sui punti di appoggio con curvatura e tangenza costanti.

Le Spline A, B e C hanno validità modale ed appartengono al gruppo delle funzioni preparatorie.

Si può utilizzare la correzione del raggio utensile.

La sorveglianza anticollisione avviene nella proiezione sul piano.

Nota

Parametri per A-SPLINE e C-SPLINE

Per la Akima-Spline (A-Spline) e la Spline cubica (C-Spline) possono essere programmate delle limitazioni per il comportamento di raccordo all'inizio ed alla fine della curva Spline.

Mediante due gruppi di istruzioni, rispettivamente di tre comandi, è possibile suddividere il comportamento di raccordo di queste curve spline.

Inizio della curva spline:

BAUTO Nessuna impostazione, l'inizio viene determinato dalla posizione del primo punto

BNAT curvatura nulla

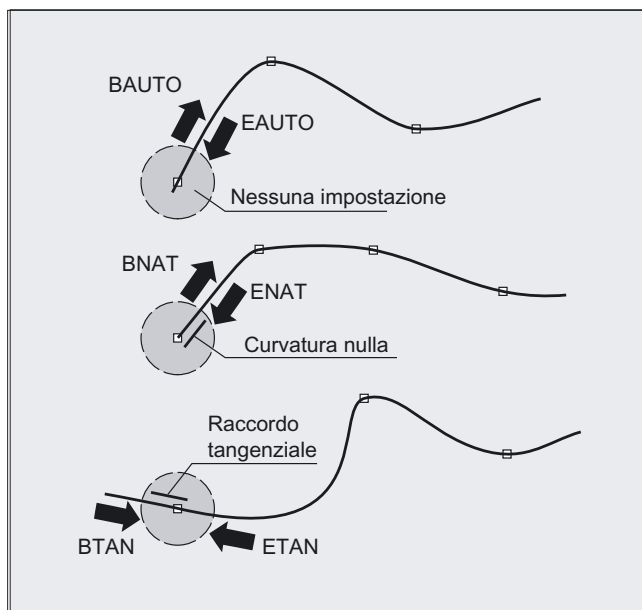
BTAN Passaggio tangenziale al blocco precedente (posizione di default)

Fine della curva spline:

EAUTO Nessuna impostazione; la fine viene determinata dalla posizione dell'ultimo punto

ENAT curvatura nulla

ETAN Passaggio tangenziale al blocco seguente (posizione di default)



Nota**Parametri per B-Spline**

Le limitazioni programmate (vedere Spline A oppure C) non hanno alcuna influenza sulla C-Spline. La B-Spline è sempre tangente al poligono di controllo, sia nel punto di partenza sia in quello di arrivo.

Classificazione dei punti:

PW = n Per ogni punto di appoggio è possibile programmare un criterio di classificazione come "peso PW del punto".

Campo dei valori:

<= n <= 3 in incrementi di 0.0001

Effetto:

n > 1 La curva viene influenzata maggiormente dal punto di controllo

n < 1 La curva viene influenzata in modo inferiore dal punto di controllo

Grado della Spline:

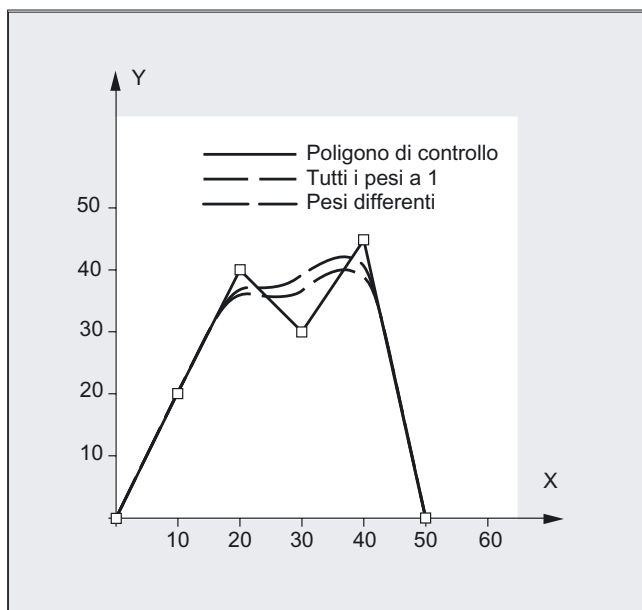
SD = 2 Come standard viene utilizzato un poligono di 3° grado. È possibile tuttavia utilizzare anche un poligono di 2° grado.

Distanza tra i nodi:

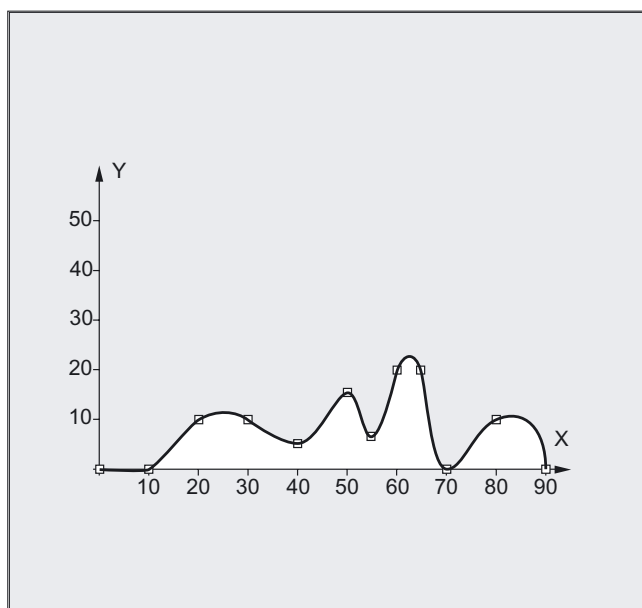
PL = valore Le distanze tra i nodi vengono calcolate internamente nel modo più adatto. Il controllo numerico è comunque in grado di elaborare distanze tra i nodi preimpostate che vengono impostate nella suddetta lunghezza dell'intervallo parametrico PL.

Valore Campo dei valori come quota del percorso

Esempio di B-Spline



Tutti i pesi a 1	Pesi differenti	Poligono di controllo
N10 G1 X0 Y0 F300 G64	N10 G1 X0 Y0 F300 G64	N10 G1 X0 Y0 F300 G64
N20 BSPLINE	N20 BSPLINE	N20 ;non necessario
N30 X10 Y20	N30 X10 Y20 PW=2	N30 X10 Y20
N40 X20 Y40	N40 X20 Y40	N40 X20 Y40
N50 X30 Y30	N50 X30 Y30 PW=0.5	N50 X30 Y30
N60 X40 Y45	N60 X40 Y45	N60 X40 Y45
N70 X50 Y0	N70 X50 Y0	N70 X50 Y0

Esempio di C-Spline, all'inizio e alla fine curvatura nulla

```
N10 G1 X0 Y0 F300
```

```
N15 X10
```

```
N20 BNAT ENAT
```

;C-Spline, all'inizio e alla fine curvatura nulla

```
N30 CSPLINE X20 Y10
```

```
N40 X30
```

```
N50 X40 Y5
```

```
N60 X50 Y15
```

```
N70 X55 Y7
```

```
N80 X60 Y20
```

```
N90 X65 Y20
```

```
N100 X70 Y0
```

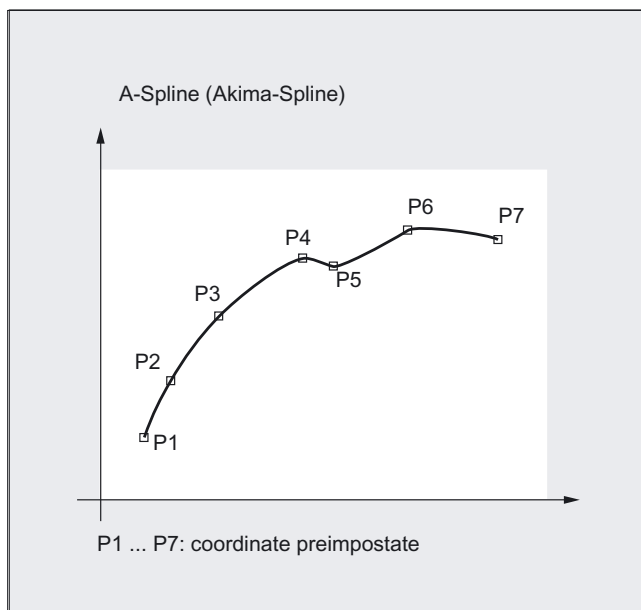
```
N110 X80 Y10
```

```
N120 X90 Y0
```

```
N130 M30
```

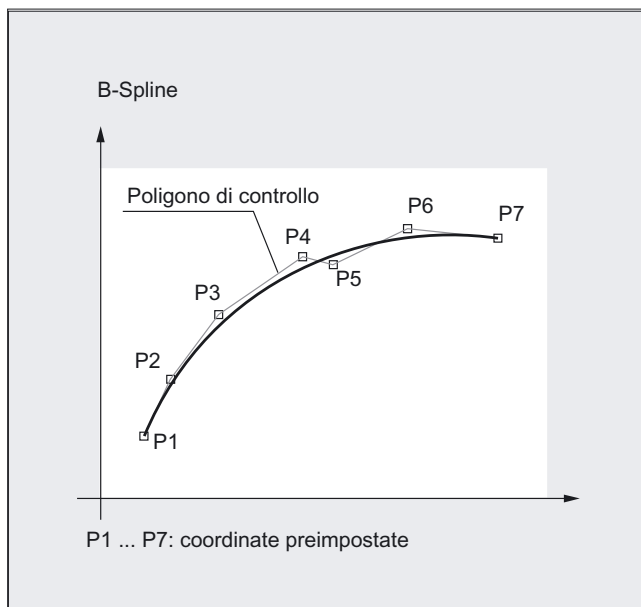
A-Spline

La A-Spline (Akima-Spline) passa esattamente per i punti di appoggio. Essa genera poche oscillazioni indesiderate, ma non è sempre costantemente curviforme sui punti di appoggio. La Akima-Spline è locale, ciò significa che la modifica di un punto di appoggio ha effetto al massimo su 6 punti vicini. Essa si presta in particolare per l'interpolazione di punti derivanti da digitalizzazioni. Per l'interpolazione viene utilizzato un polinomio di 3. grado.



B-Spline

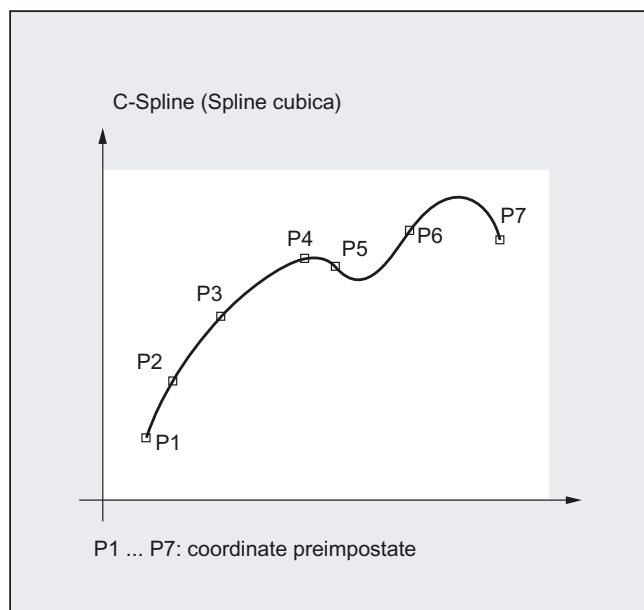
Nella B-Spline le posizioni programmate non sono punti di appoggio, ma solo punti di controllo della spline. Perciò la curva non passa direttamente sui punti, ma viene da essi "influenzata". Il collegamento dei punti mediante rette forma il poligono di controllo della spline. La B-Spline si presta in particolare per descrivere i percorsi dell'utensile su superfici a forme libere. Essa è stata concepita fundamentalmente come interfaccia con i sistemi CAD. Una B-Spline di 3° grado non genera oscillazioni nonostante i passaggi curvilinei del profilo.



C-Spline

A differenza della Akima-Spline, la spline cubica (C-Spline) ha una curvatura costante sui punti di appoggio. Essa tende tuttavia a creare oscillazioni inattese. Può pertanto essere utilizzata quando i punti si trovano su una curva analiticamente nota. La C-Spline utilizza polinomi di 3° grado.

La spline non è locale, pertanto la variazione di un punto di appoggio può avere effetti su diversi blocchi (con influenza decrescente).

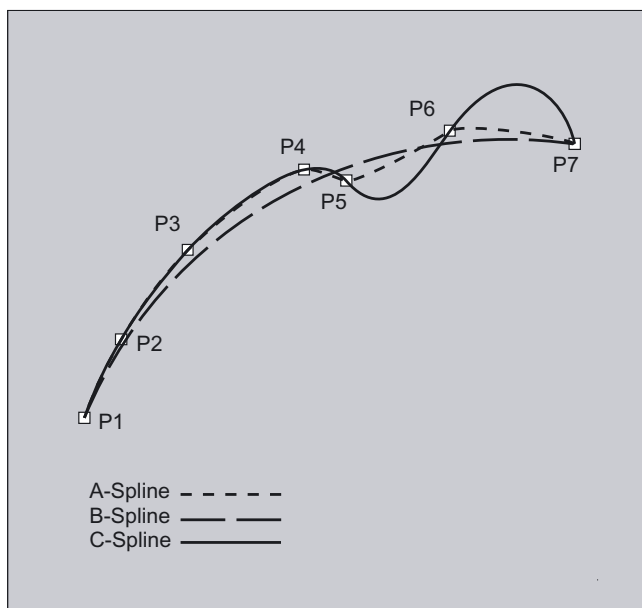


Confronto tra tre tipi di spline con gli stessi punti di appoggio:

A-Spline (Akima-Spline)

B-Spline (Bezier-Spline)

C-Spline (Spline cubica)



Numero minimo di blocchi per spline

I codici G ASPLINE, BSPLINE e CSPLINE collegano i punti finali di blocchi con delle spline. In questa operazione deve essere calcolata contemporaneamente una sequenza di blocchi (punti finali). La dimensione standard del buffer per il calcolo è di 10 blocchi. Non tutte le informazioni del blocco sono punti finali di una spline. Il controllo numerico ha comunque bisogno nei 10 blocchi di un determinato numero di blocchi di punti finali di spline.

Tipo di spline	Numero minimo di blocchi per spline
A-Spline:	di ogni 10 blocchi, almeno 4 devono essere blocchi spline. I blocchi di commento e i calcoli parametrici non vengono considerati in questo conteggio.
B-Spline:	di ogni 10 blocchi, almeno 6 devono essere blocchi spline. I blocchi di commento e i calcoli parametrici non vengono considerati in questo conteggio.
C-Spline:	Il numero minimo necessario di blocchi spline si ottiene dalla somma seguente: valore dato da MD20160 \$MC_CUBIC_SPLINE_BLOCKS + 1 In MD20160 viene inserito il numero di punti per il calcolo della sezione spline. L'impostazione standard è 8. Di ogni 10 blocchi, almeno 9 devono essere blocchi spline.

Nota

Se il valore tollerato viene superato per difetto e se un asse compreso nella spline viene programmato come asse di posizionamento, viene emesso un allarme.

Raggruppamento di blocchi spline brevi

Dall'interpolazione Spline si possono ottenere blocchi brevi, che determinano una riduzione superflua della velocità di avanzamento. Con la funzione "Raggruppamento di blocchi spline

brevi" è possibile riunire questi blocchi in modo che la lunghezza del blocco risultante sia sufficientemente elevata e non determini una riduzione della velocità di avanzamento.

La funzione viene attivata tramite il dato macchina specifico per canale:

MD20488 \$MC_SPLINE_MODE (impostazione per interpolazione Spline)

Bibliografia:

/FB1/ Manuale delle funzioni di base; Avanzamenti, arresto preciso, LookAhead (B1), capitolo: Raggruppamento di blocchi spline brevi

5.3 Raggruppamento spline (SPLINEPATH)

Funzione

Gli assi da interpolare coinvolti in un raggruppamento Spline vengono selezionati con l'istruzione SPLINEPATH. In una interpolazione spline possono essere coinvolti fino ad 8 assi di contornatura. Con l'istruzione SPLINEPATH vengono definiti gli assi interessati dall'interpolazione spline.

Programmazione

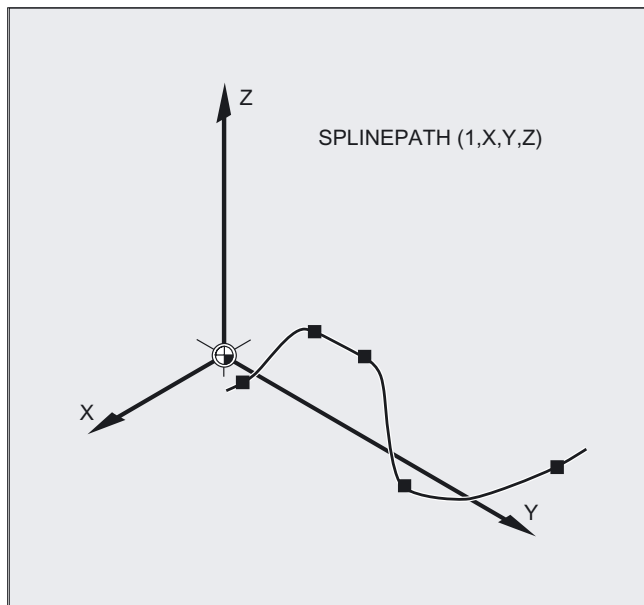
SPLINEPATH (n, X, Y, Z, ...)

La definizione avviene in un blocco a sé stante. Se SPLINEPATH non viene programmata esplicitamente, i primi tre assi del canale vengono considerati come gruppo di assi per l'interpolazione spline.

Parametri

SPLINEPATH	Definizione del raggruppamento Spline
n = 1	valore fisso
X, Y, Z, ...	Indicazioni degli assi di contornatura

Esempio di raggruppamento Spline con tre assi di contornitura



```
N10 G1 X10 Y20 Z30 A40 B50 F350
```

```
N11 SPLINEPATH(1,X,Y,Z)
```

```
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60
```

```
N14 X30 Y40 Z50 A60 B70
```

```
...
```

```
N100 G1 X... Y...
```

;Raggruppamento spline

;C-Spline

;Punti di appoggio

;Disattivazione dell'interpolazione spline

5.4 Compressore (COMPOF/ON, COMPCURV, COMPCAD)

Funzione

Con il codice G COMPON i passaggi di blocco vengono eseguiti a **velocità** costante, mentre l'accelerazione degli assi interessati può subire dei salti in corrispondenza dei passaggi di blocco. Questo può provocare vibrazioni della macchina.

Con il codice G COMPCURV i passaggi di blocco vengono impostati ad **accelerazione costante**. In questo modo sui raccordi di blocco viene garantita una velocità e un'accelerazione costanti per tutti gli assi. Con il codice G COMPCAD può essere selezionata una ulteriore compressione ottimizzata per la **velocità e per la qualità della superficie**.

Costruttore della macchina

Le funzioni compressore sono progettabili e dipendono quindi dalle impostazioni dei dati macchina.

Programmazione

COMPON

oppure

COMPOF

oppure

COMPCURV

oppure

COMPCAD

Condizioni di impiego per blocchi NC programmati

Questo processo di compressione viene eseguito solo per blocchi lineari (G1). Esso viene interrotto da ogni altra istruzione NC, ad esempio con l'emissione di funzioni ausiliarie, ma non da calcoli parametrici. Vengono compressi soltanto dei blocchi che non contengono più come numero di blocco G1, indirizzi assi, velocità di avanzamento e commenti. L'elaborazione degli altri blocchi avviene senza che vi siano modifiche (senza compressione). Non è consentito l'impiego di variabili.

Parametri

COMPON /	Compressore ON, velocità costante
COMPOF	Compressore OFF
COMPCURV	Compressore ON, polinomi a curvatura continua (accelerazione costante)
COMPCAD	Compressore ON, qualità della superficie ottimizzata (velocità ottimizzata)

Nota

I codici G COMPOF/COMPON, COMPCURV e COMPCAD hanno un effetto di tipo modale.

Esempio COMPON

N10 COMPON	;opp. COMPCURV, compressore ON
N11 G1 X0.37 Y2.9 F600	;G1 deve trovarsi tra punto finale e ;velocità di avanzamento
N12 X16.87 Y-.698	
N13 X16.865 Y-.72	
N14 X16.91 Y-.799...	
N1037 COMPOF	;Compressore OFF
...	

Nota

Vengono compressi tutti i blocchi che rispettano una sintassi semplice, ad es.

N19 X0.103 Y0. Z0.
N20 X0.102 Y-0.018
N21 X0.097 Y-0.036
N22 X0.089 Y-0.052
N23 X0.078 Y-0.067

Vengono anche compressi i blocchi di movimento con indirizzi ampliati come C=100 oppure A=AC(100).

Esempio COMPCAD

G00 X30 Y6 Z40	
G1 F10000 G642	
SOFT	
COMPCAD	;Compressore, ottimizzazione della superficie ON
STOPFIFO	
N24050 Z32.499	
N24051 X41.365 Z32.500	
N24052 X43.115 Z32.497	
N24053 X43.365 Z32.477	
N24054 X43.556 Z32.449	
N24055 X43.818 Z32.387	
N24056 X44.076 Z32.300	
...	
COMPOF	;Compressore OFF
G00 Z50	
M30	

Premessa

Costruttore della macchina

Per la funzione di compressione sono disponibili tre dati macchina:

- **\$MC_COMPRESS_BLOCK_PATH_LIMIT**
Viene impostata una lunghezza di percorso massima che comprende i blocchi comprimibili. I blocchi più lunghi non vengono compressi.
- **\$MA_COMPRESS_POS_TOL**
Per ogni asse è possibile impostare una tolleranza. La spline generata si discosta al massimo di questo valore rispetto ai punti programmati. Quanto più alto è questo valore, tanto più aumenta il numero di blocchi comprimibili.
- **\$MC_COMPRESS_VELO_TOL**
È possibile impostare lo scostamento massimo ammesso dell'avanzamento vettoriale con compressore attivo in relazione a FLIN e FCUB.

COMPCAD

- **\$MN_MM_EXT_PROG_BUFFER_SIZE** deve essere selezionato abbastanza grande, ad es. 100 (kB).
- **\$MC_COMPRESS_BLOCK_PATH_LIMIT** deve essere impostato decisamente più grande, ad es. 50 (mm).
- **\$MC_MM_NUM_BLOCKS_IN_PREP** deve essere impostato ≥ 60 , per poter elaborare molto più di 10 punti.
- FLIN e FCUB non possono essere utilizzate.

Consigliate per ampie lunghezze di blocchi e ottimale velocità:

- **\$MC_MM_MAX_AXISPOLY_PER_BLOCK = 5**
\$MC_MM_PATH_VELO_SEGMENTS = 5
\$MC_MM_ARCLENGTH_SEGMENTS = 10

Descrizione

I sistemi CAD/CAM forniscono generalmente blocchi lineari che rispettano la precisione parametrizzata. Questo provoca, nel caso di profili complessi, una considerevole quantità di dati ed eventualmente dei segmenti di percorso brevi. Questi segmenti brevi limitano la velocità di elaborazione.

Il compressore ha la possibilità di raggruppare un determinato numero (max. 10) di questi segmenti di percorso brevi in un unico segmento.

Con il codice G modale **COMPON** o **COMPCURV** è possibile attivare un "compressore di blocchi NC". Questa funzione raccoglie una serie di blocchi con interpolazione lineare (il numero di blocchi è limitato a 10) e li approssima, secondo una funzione di tolleranza di errore definita in un dato macchina, mediante polinomi di 3° grado (**COMPON**) o di 5° grado (**COMPCURV**). Al posto di tanti blocchi brevi, viene elaborato dall'NC un blocco di movimento più grande. Per ulteriori informazioni sulla compressione dei blocchi spline brevi, vedere il capitolo "Interpolazione Spline".

COMPCAD

COMPCAD richiede molto tempo per i calcoli e notevole spazio di memoria. Si dovrebbe utilizzare solo se gli accorgimenti per il miglioramento della superficie non possono essere garantiti con programmi CAD/CAM. Caratteristiche:

- COMPCAD genera blocchi polinomiali che si trasformano l'uno nell'altro con accelerazione costante.
- Per profili adiacenti gli scostamenti portano comunque nella stessa direzione.
- Con il dato di setting \$SC_CRIT_SPLINE_ANGLE può essere fissato un angolo limite, a partire dal quale COMPCAD non modifica gli spigoli.
- Il numero di blocchi da comprimere non è limitato a 10.
- COMPCAD elimina le irregolarità di superfici difettose. Le tolleranze vengono ancora mantenute, l'angolo limite al vertice non viene però tenuto in considerazione.
- La funzione di raccordo G642 è ancora utilizzabile.

Ampliamenti di COMPON, COMPCURV e COMPCAD

I compressori COMPON, COMPCURV e COMPCAD sono stati ampliati in modo da poter comprimere anche i programmi NC nei quali l'orientamento è stato programmato mediante vettori direzionali mantenendo una tolleranza preimpostabile.

Trasformazione dell'orientamento TRAORI

Per la funzione "Compressore per orientamenti" deve essere presente l'opzione Trasformazione dell'orientamento. Le suddette restrizioni citate nelle "Condizioni di impiego" vengono ora attenuate e quindi è possibile assegnare dei valori di posizione attraverso l'assegnazione di parametri.

Formato base del blocco NC:

N10 G1 X=<...> Y=<...> Z=<...>	;Posizioni degli assi come espressioni di
A=<...> B=<...> F=<...>	parametri con
;Commento	< ... > espressione parametrica, ad es.
	X=R1 * (R2+R3)

Attivazione

L'attivazione del "Compressore per orientamenti" avviene con una delle istruzioni: COMPON, COMPCURV (COMPCAD non è possibile).

Con la trasformazione per l'orientamento (TRAORI) attiva, i compressori possono essere utilizzati sia nelle

- macchine a 5 assi, sia nelle
- macchine a 6 assi nelle quali può essere inoltre programmata anche la rotazione dell'utensile per l'orientamento dell'utensile stesso.

Per ulteriori descrizioni sull'impiego di compressori in macchine a 5 e 6 assi vedere il capitolo Trasformazioni "Compressione dell'orientamento"

Bibliografia: /FB3/ Manuale delle funzioni speciali; Trasformazioni a 3 ... 5 assi (F2).

5.5 Interpolazione polinomiale (POLY, POLYPATH)

Funzione

Propriamente, nel caso dell'interpolazione polinomiale (POLY), non si tratta di un tipo di interpolazione Spline. Essa è stata pensata inizialmente come interfaccia verso una programmazione di curve Spline generate esternamente. In questo caso le sezioni spline possono essere programmate direttamente.

Questo tipo di interpolazione sgrava il NC dal calcolo dei coefficienti polinomiali. Essa può essere utilizzata in modo ottimale quando i coefficienti provengono direttamente da un sistema CAD o da un postprocessore.

Programmazione

POLY PO [X] = (x_e, a₂, a₃) PO [Y] = (y_e, b₂, b₃) PO [Z] = (z_e, c₂, c₃) PL=n polinomio di 3. grado

oppure mplemento al polinomio di 5. grado e alla nuova sintassi polinomiale

POLY X=PO (x_e, a₂, a₃, a₄, a₅) Y=PO (y_e, b₂, b₃, b₄, b₅) Z=PO (z_e, c₂, c₃, c₄, c₅) PL=n
POLYPATH ("AXES", VECT")

Parametri

POLY	Attivazione dell'interpolazione polinomiale tramite un blocco con POLY.
POLYPATH	Interpolazione polinomiale selezionabile per entrambi i gruppi di assi AXIS o VECT
PO [identificatore asse/variabile] = (... , ..., ...)	Punti di destinazione e coefficienti del polinomio
X, Y, Z	indicatore dell'asse
x _e , y _e , z _e	Indicazione della posizione di destinazione nei singoli assi; campo dei valori come per le quote di percorso
a ₂ , a ₃ , a ₄ , a ₅	I coefficienti a ₂ , a ₃ , a ₄ , und a ₅ vengono scritti con il loro valore; campo dei valori come quote di percorso. L'ultimo coefficiente può essere omesso se ha valore nullo.
PL	Lunghezza dell'intervallo parametrico sul quale sono definiti i polinomi (campo di definizione della funzione f (p)). L'intervallo inizia sempre con 0. p può assumere valori da 0 a PL. Campo dei valori teorico per PL: 0,0001 ... 99 999,9999. Il valore PL vale per il blocco in cui è scritto. Se PL non è stato programmato viene considerato PL=1.

Attivazione/disattivazione POLY

L'interpolazione polinomiale è inserita insieme a G0, G1, G2, G3, A-Spline, B-Spline e C-Spline, nel primo gruppo di funzioni G. Quando essa è attiva, non è indispensabile programmare la sintassi del polinomio: gli assi che sono programmati solo con il loro nome e punto finale, vengono mossi in modo lineare verso il punto finale. Se tutti gli assi vengono programmati in questo modo, il controllo numerico si comporta come per G1.

L'interpolazione polinomiale viene disattivata con un altro comando del gruppo G (ad esempio G0, G1).

Coefficiente polinomiale

Il valore PO (PO [=] oppure ...=PO (...)) indica tutti i coefficienti polinomiali per un asse. A seconda del grado del polinomio, i singoli valori vengono separati con la virgola. Nell'ambito di un blocco sono ammessi polinomi di diverso grado per i singoli assi.

Nuova sintassi polinomiale con PO: L'attuale sintassi resta ancora valida.

Richiamo del sottoprogramma POLYPATH

Con POLYPATH l'interpolazione polinomiale può essere indicata in modo selettivo per i seguenti gruppi di assi:

- POLYPATH ("AXES")
Tutti gli assi di contornitura e gli assi supplementari.
- POLYPATH ("VECT") Assi di orientamento
(con trasformazioni di orientamento).

Di default, per entrambi i gruppi di assi, i polinomi programmati vengono anche interpolati come polinomi.

Esempio:

```
POLYPATH ("VECT")
```

Per l'interpolazione polinomiale sono selezionati solo gli assi di orientamento, tutti gli altri assi si muovono in modo lineare.

```
POLYPATH ( )
```

Disattivazione dell'interpolazione polinomiale per tutti gli assi

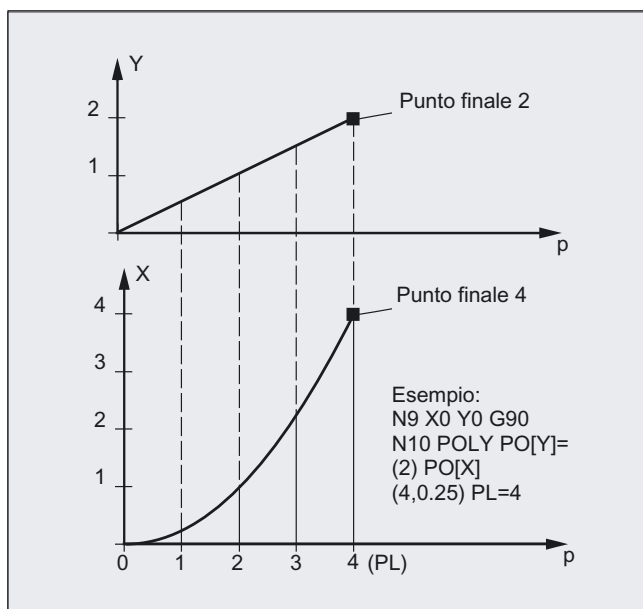
Esempio

```
N10 G1 X... Y... Z... F600
N11 POLY PO[X]=(1,2.5,0.7) -> ;Interpolazione polinomiale ON
-> PO[Y]=(0.3,1,3.2) PL=1.5
N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7) PL=3
...
N20 M8 H126 ...
N25 X70 PO[Y]=(9.3,1,7.67) PL=5 ;Impostazione mista per gli assi
N27 PO[X]=(10,2.5) PO[Y]=(2.3) ;PL non è stato programmato, vale PL=1
N30 G1 X... Y... Z. ;Interpolazione polinomiale OFF
...
```

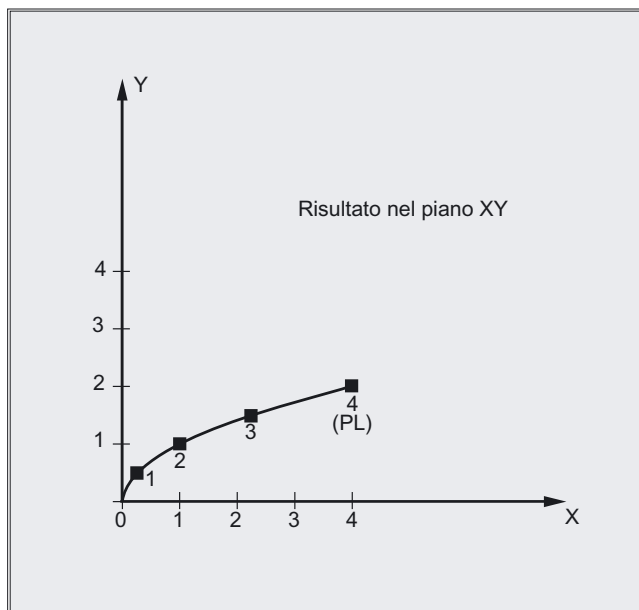
Esempio di sintassi polinomiale valida con PO

<pre>L'attuale sintassi polinomiale resta valida PO[identificatore asse]=(.. , ..) PO[PHI]=(.. , ..) PO[PSI]=(.. , ..) PO[THT]=(.. , ..) PO[]=(.. , ..) PO[variabile]=IC(.. , ..)</pre>	<pre>;Nuova sintassi polinomiale (da SW 6) ;Identificatore asse=PO(.. , ..) ;PHI=PO(.. , ..) ;PSI=PO(.. , ..) ;THT=PO(.. , ..) ;PO(.. , ..) ;Variabile=PO IC(.. , ..)</pre>

Esempio di curva nel piano X/Y



```
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4
```



Descrizione

Il controllo è in grado di eseguire curve (traiettorie) per le quali ogni asse di contornitura selezionato segue una funzione polinomiale max. 5° grado.

La formula generica della funzione polinomiale è:

$$f(p) = a_0 + a_1p + a_2p^2 + a_3p^3$$

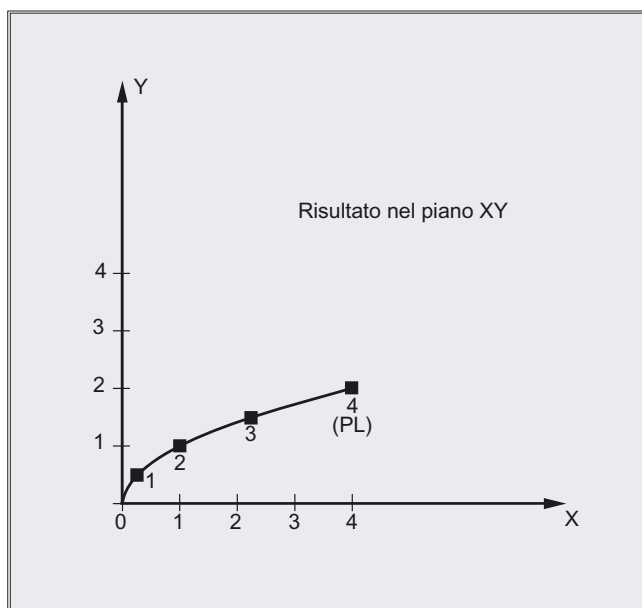
oppure

$$f(p) = a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

I relativi significati sono:

a_n : coefficienti costanti

p: Parametri



Definendo i coefficienti con valori concreti possono essere generati i più diversi tipi di curve (funzione retta, parabola, esponenziale).

Impostando i coefficienti $a_2 = a_3 = 0$ oppure $a_2 = a_3 = a_4 = a_5 = 0$ si ottiene, ad esempio, una retta con:

$$f(p) = a_0 + a_1 p$$

Vale:

a_0 = posizione dell'asse al termine del blocco precedente

$p = PL$

$$a_1 = (x_E - a_0 - a_2 p^2 - a_3 p^3) / p$$

È possibile programmare polinomi, **senza** che il codice G POLY sia attivo. In questo caso non vengono comunque interpolati i polinomi programmati, ma viene raggiunto in modo lineare (G1) il rispettivo punto finale programmato di ogni asse. Tramite la programmazione di POLY si attiva l'interpolazione polinomiale.

Inoltre, con codice G POLY attivo, con il sottoprogramma predefinito POLYPATH (...) è possibile selezionare quali assi devono essere interpolati con il polinomio.

Particolarità del polinomio denominatore

Per gli assi geometrici è possibile programmare con $PO[]=(...)$ senza indicazione del nome dell'asse, un polinomio denominatore comune. Questo significa che il movimento degli assi geometrici viene interpolato come quoziente di due polinomi.

In questo modo è possibile per esempio rappresentare con esattezza sezioni coniche (cerchio, ellisse, parabola, iperbole).

Esempio:

```
POLY G90 X10 Y0 F100
```

;Gli assi geometrici si muovono
;linearmente sulla posizione X10, Y0

```
PO[X] = (0, -) PO[Y] = (10) PO[] = (2, 1)
```

;Gli assi geometrici si muovono con un
;quarto di cerchio alla posizione X0, Y10

Il coefficiente costante (a_0) del polinomio denominatore viene considerato sempre 1, il punto di destinazione indicato è indipendente da G90/G91.

Dall'esempio sopraindicato si ha il seguente risultato:

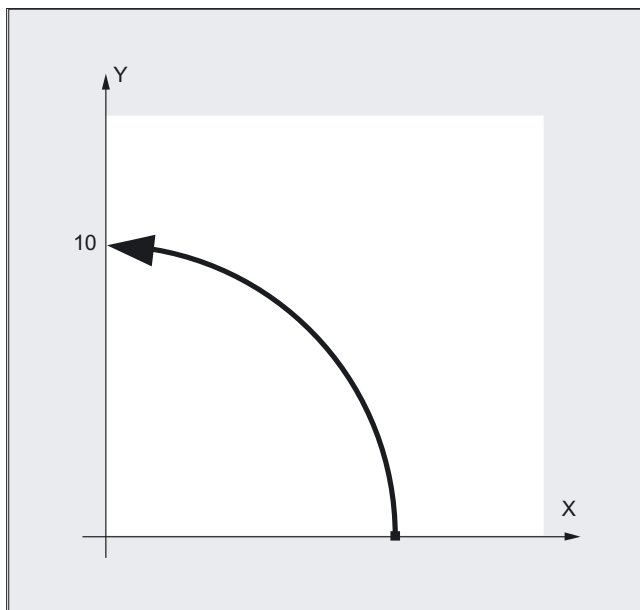
$X(p) = 10(1)/(1+p^2)$ e $Y(p) = 20p/(1+p^2)$ con $0 \leq p \leq 1$

In funzione dei punti di partenza, dei punti di destinazione, del coefficiente a_2 e $PL=1$ programmati si hanno i seguenti valori intermedi:

Numeratore (X) = $10+0 \cdot p - p^2$

Numeratore (Y) = $0+20 \cdot p + 0 \cdot p^2$

Denominatore = $1+2 \cdot p + 1 \cdot p^2$



Quando l'interpolazione polinomiale è attiva, la programmazione di un polinomio denominatore con posizioni nulle nell'ambito dell'intervallo $[0, PL]$ viene rifiutata con un allarme. Il polinomio denominatore non ha alcuna influenza sul movimento di assi supplementari.

Nota

La correzione del raggio utensile è inseribile con G41, G42 anche per l'interpolazione polinomiale, ed è utilizzabile analogamente all'interpolazione lineare e circolare.

5.6 Riferimento vettoriale impostabile, (SPATH, UPATH)

Funzione

Durante l'interpolazione polinomiale possono essere selezionati dall'utente due diversi rapporti tra gli assi FGROUP che definiscono la velocità e gli altri assi di contornitura: questi ultimi devono essere gestiti

- in modo sincrono al percorso vettoriale degli assi FGROUP oppure
- in modo sincrono al parametro di curvatura.

Per gli assi non contenuti in FGROUP esistono pertanto due possibilità di seguire il percorso:

1. sincrono al percorso S (SPATH)
2. oppure sincrono al parametro di curvatura U degli assi FGROUP (UPATH).

I due tipi di interpolazione vettoriale vengono utilizzati in applicazioni diverse e possono essere commutati con i codici G SPATH e UPATH.

Programmazione

SPATH

oppure

UPATH

Parametri

SPATH	Il riferimento di percorso per gli assi FGROUP è la lunghezza dell'arco
UPATH	Il riferimento di percorso per gli assi FGROUP è il parametro della curva
FGROUP	Definizione degli assi con avanzamento vettoriale

SPATH, UPATH

Con uno dei due codici G (SPATH, UPATH) si può selezionare e programmare il comportamento desiderato.

I comandi hanno validità modale. Se è attivo SPATH gli assi vengono traslati in modo sincrono al percorso, se è attivo UPATH vengono traslati in modo sincrono al parametro di curvatura.

UPATH e SPATH determinano anche il rapporto del polinomio F-Wort (FPOLY, FCUB, FLIN) con il movimento vettoriale.

Attivazione FGROUP

Il riferimento vettoriale per gli assi non contenuti in FGROUP viene impostato con le due istruzioni SPATH e UPATH contenute nel 45° gruppo di codici G.

Esempio 1

L'esempio che segue descrive di un quadrato di 20 mm di lato raccordato con G643. Lo scostamento massimo dal profilo esatto, viene stabilito per ogni asse con i dati macchina DM 33100: COMPRESS_POS_TOL[...].

```
N10 G1 X... Y... Z... F500
```

```
N20 G643
```

;Raccordo interno al blocco con G643

```
N30 X0 Y0
```

```
N40 X20 Y0
```

;mm lunghezza del lato per gli assi

```
N50 X20 Y20
```

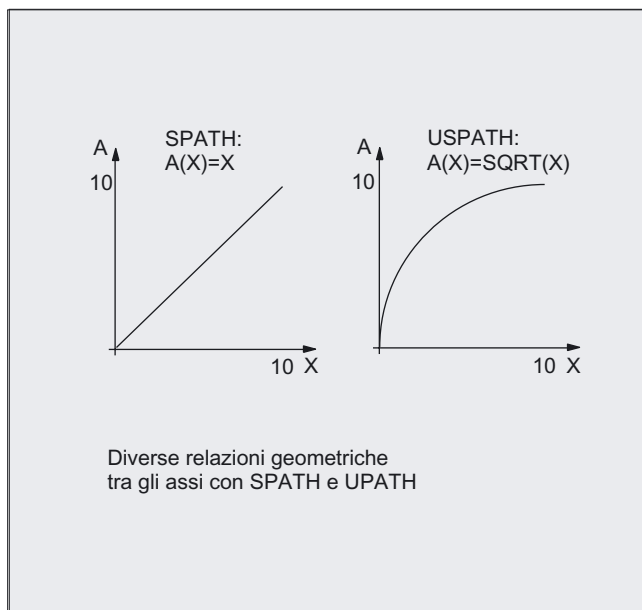
```
N60 X0 Y20
```

```
N70 X0 Y0
```

```
N100 M30
```

Esempio 2

Il seguente esempio illustra la differenza tra i due tipi di gestione del movimento. In entrambi i casi è attiva la preimpostazione FGROUP(X,Y,Z).



```
N10 G1 X0 A0 F1000 SPATH
```

```
N20 POLY PO[X]=(10, 10) A10
```

oppure

```
N10 G1 X0 F1000 UPATH
```

```
N20 POLY PO[X]=(10, 10) A10
```

Nel blocco N20 il percorso S degli assi FGROUP dipende dal quadrato del parametro di curvatura U. Risultano pertanto diverse posizioni dell'asse sincrono A lungo il percorso di X, a seconda se è attivo SPATH o UPATH :

Condizioni marginali

Il riferimento vettoriale impostato non ha significato in caso di

- Interpolazioni lineare e circolare,
- in blocchi di filettatura e
- se tutti gli assi di contornitura sono contenuti in FGROUP.

Descrizione

Durante l'interpolazione polinomiale - e con questo siano sempre intese l'interpolazione polinomiale

- nel senso più stretto (POLY),
- tutti i tipi di interpolazione Spline (ASPLINE, BSPLINE, CSPLINE) e
- interpolazione lineare con compressore (COMPON, COMPCURV)

- vengono impostate le posizioni di tutti gli assi di contornitura attraverso i polinomi $\pi(U)$. Il parametro di curvatura U si muove all'interno di un blocco NC da 0 a 1; è quindi normalizzato.

Con l'istruzione FGROUP, tra gli assi di contornitura è possibile selezionare quelli ai quali riferire l'avanzamento vettoriale programmato. Una interpolazione a velocità costante sul percorso S di questi assi, durante l'interpolazione polinomiale, significa comunque generalmente una variazione non costante del parametro di curvatura U .

Comportamento del controllo numerico con Reset e dati macchina/opzionali

Dopo Reset con il DM 20150: GCODE_RESET_VALUES [44] viene attivato il codice G attivo (45. gruppo codici G).

Il valore di default per il tipo di raccordo viene impostato con il DM 20150: GCODE_RESET_VALUES [9] (10. gruppo codici G).

Il valore del gruppo di codici G attivo dopo un reset viene determinato dal dato macchina DM 20150: GCODE_RESET_VALUES [44]. Per mantenere la compatibilità con gli impianti esistenti, come valore standard viene impostato SPATH.

I dati macchina assiali DM 33100: COMPRESS_POS_TOL hanno un significato ampliato: essi contengono la tolleranza per le funzioni di compressione e per i raccordi con G642.

5.7 Misura con tastatore a commutazione (MEAS, MEAW)

Funzione

Per tutti gli assi programmati nel blocco NC vengono registrate le posizioni all'intervento del fronte del tastatore di misura. Tali posizioni vengono scritte per ogni asse nelle corrispondenti celle di memoria. Sono previsti al massimo due tastatori di misura.

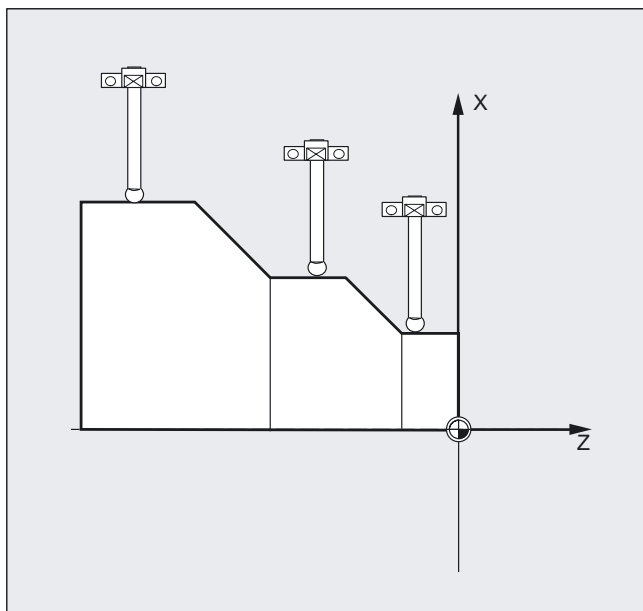
Lettura dei risultati di misura

Il risultato della misura, per gli assi rilevati con il tastatore, è disponibile nelle seguenti variabili:

- nel sistema di coordinate macchina in `$AA_MM[Asse]`
- nel sistema di coordinate pezzo in `$AA_MW[Asse]`

Nella lettura di queste variabili non viene generato internamente alcuno stop preelaborazione.

Con STOPRE nel programma NC si deve programmare uno stop preelaborazione nella posizione più idonea. Altrimenti vengono letti dei valori errati.



Programmazione

Programmazione di blocchi di misura, MEAS, MEAW

Con l'istruzione **MEAS** ed un tipo di interpolazione vengono raggiunte le posizioni reali nel pezzo e vengono registrati i valori di misura. Il percorso residuo tra posizione programmata e posizione reale viene cancellato.

Per particolari misure, per le quali è necessario raggiungere in ogni caso la posizione programmata, viene utilizzata la funzione **MEAW**. **MEAS** e **MEAW** sono validi solo nel blocco in cui sono stati programmati.

MEAS=±1	G... X... Y... Z...	(+1/+2 misure con cancellazione del
MEAS=±2	G... X... Y... Z...	percorso residuo e fronte positivo)
		(-/- misure con cancellazione del
		percorso residuo e fronte negativo)
MEAW=±1	G... X... Y... Z...	(+1/+2 misure senza cancellazione
MEAW=±2	G... X... Y... Z...	percorso residuo e fronte positivo)
		(-/- misure senza cancellazione del
		percorso residuo e fronte negativo)

Parametri

MEAS=±1	Misura con tastatore 1 sull'ingresso di misura 1
MEAS=±2*	Misura con tastatore 2 sull'ingresso di misura 2
MEAW=±1	Misura con tastatore 1 sull'ingresso di misura 1
MEAW=±2*	Misura con tastatore 2 sull'ingresso di misura 2
G...	Tipo di interpolazione, es. G0, G1, G2 o G3
X... Y... Z...	punto di arrivo in coordinate cartesiane

*In base all'esecuzione, max. 2 ingressi

Esempio di programmazione di blocchi di misura

MEAS e MEAW vengono programmate nel blocco con le istruzioni di movimento. Gli avanzamenti ed i tipi di interpolazione (G0, G1, ...) vanno scelti in funzione delle varie problematiche della misura; altrettanto vale per il numero di assi.

N10 MEAS=1 G1 F1000 X100 Y730 Z40

Blocco di misura con tastatore di misura sul primo ingresso di misura ed interpolazione lineare. Lo stop preelaborazione viene generato automaticamente.

Descrizione

Stato dell'ordine di misura

Se nel programma risulta necessario valutare se il tastatore ha commutato o meno, è possibile interrogare la variabile di stato `$AC_MEA[n]` (n= numero del tastatore di misura):

0 Ordine di misura non eseguito

1 Ordine di misura concluso con successo (il tastatore di misura ha commutato)

Nota

Se il tastatore commuta durante l'esecuzione del programma, la variabile viene settata a 1. Startando un blocco di misura, la variabile viene aggiornata subito all'inizio del blocco.

Acquisizione dei valori di misura

Vengono registrate le posizioni di tutti gli assi di contornitura e di posizionamento che sono in movimento (il numero max. degli assi dipende dalla configurazione del controllo numerico). Con MEAS, dopo la commutazione del tastatore, il movimento viene arrestato con la frenatura predefinita.

Nota

Se in un blocco di misura è stato programmato un'asse GEO, vengono memorizzati i valori di misura per tutti gli assi GEO attuali.

Se in un blocco di misura è stato programmato un'asse facente parte di una trasformazione vengono definiti i valori di misura di tutti gli assi partecipanti alla trasformazione.

5.8 Funzione di misura ampliata (MEASA, MEAWA, MEAC) (opzione)

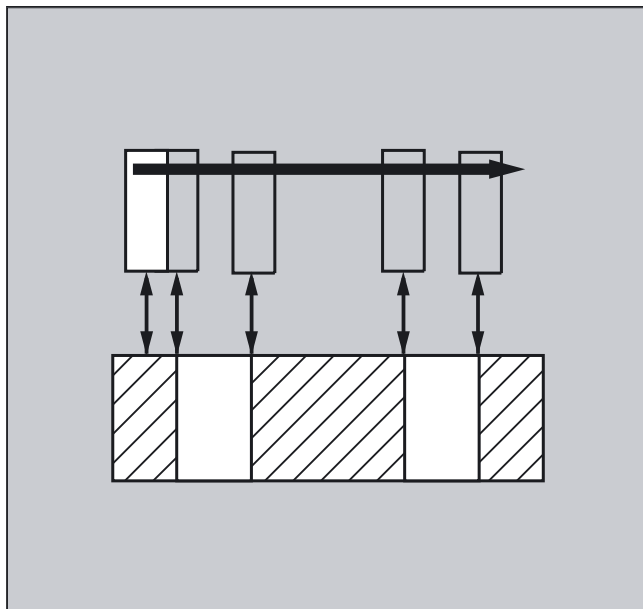
Funzione

Nelle misure assiali si possono utilizzare più tastatori di misura e più sistemi di misura.

Con MEASA e MEAWA per ogni misura vengono acquisiti fino a quattro valori per ogni asse programmato e memorizzati poi nelle variabili di sistema in funzione dell'evento di trigger.

Con MEAC si possono eseguire misure continue. In questo caso i risultati delle misure vengono depositati nelle variabili FIFO. Anche per MEAC sono possibili al massimo 4 valori per ogni misura:

- nel sistema di coordinate macchina in \$AA_MM1 . . . 4 [Asse]
- nel sistema di coordinate pezzo in \$AA_WM1 . . . 4 [Asse]



Programmazione

MEASA e MEAWA sono attivi nel blocco e possono essere programmati in un blocco a sè stante. Se MEASA/MEAWA vengono programmati in un blocco con MEAS/MEAW viene emessa una segnalazione di errore.

```
MEASA[asse] = (Modo, TE1, . . . , TE4)
```

oppure

```
MEAWA[asse] = (Modo, TE1, . . . , TE4)
```

oppure

```
MEAC[asse] = (Modo, memoria di misura, TE1, . . . , TE4)
```


Parametri

MEASA	Misura con cancellazione del percorso residuo
MEAWA	Misura senza cancellazione del percorso residuo
MEAC	Misura continua senza cancellazione del percorso residuo
Asse	Nome dell'asse canale utilizzato per la misura
Modo	Indicazione a due cifre del modo operativo, composto da Modo di misura (decade delle unità) e 0: modo 0: interruzione funzione di misura 1: modo 1: fino a 4 eventi di trigger attivabili contemporaneamente . 2: modo 2: fino a 4 diversi eventi di trigger attivabili in sequenza 3: modo 3: fino a 4 eventi di trigger attivabili in sequenza , tuttavia senza sorveglianza dell'evento di trigger 1 allo START (vengono disabilitati gli allarmi 21700/21703) Nota: il modo 3 MEAC non è possibile Sistema di misura (decade delle decine) 0 oppure nessuna indicazione: sistema di misura attivo 1: sistema di misura 1 2: sistema di misura 2 3: entrambi i sistemi di misura
TE 1...4	Evento di trigger 1: fronte di salita, tastatore di misura 1 -1: fronte di discesa, tastatore di misura 1 2: fronte di salita, tastatore di misura 2 -2: fronte di discesa, tastatore di misura 2
Memoria di misura	Numero del FIFO (memoria ad anello)

Esempio di misure con cancellazione del percorso residuo nel modo 1

(acquisizione in sequenza temporale)

a) con sistema di misura 1

...	
N100 MEASA[X] = (1,1,-1) G01 X100 F100	;Misura in modo 1 con sistema di misura ;attivo. Attesa del segnale di misura ;con fronte di salita/discesa del tastatore nel ;tratto di percorso verso ;X = 100.
N110 STOPRE	;Stop preelaborazione
N120 IF \$AC_MEA[1] == FALSE gotof END	;Controllare la buona riuscita della misura.
N130 R10 = \$AA_MM1[X]	;Memorizzare il valore di misura ;corrispondente al primo evento di trigger ;programmato (fronte di salita).
N140 R11 = \$AA_MM2[X]	;Memorizzare il valore di misura ;corrispondente al secondo evento di trigger ;programmato (fronte di discesa).
N150 END:	

Esempio di misure con cancellazione del percorso residuo nel modo 1**b) con 2 sistemi di misura**

...	
N200 MEASA[X] = (31,1-1) G01 X100 F100	;Misura in modo 1 con entrambi i sistemi di misura. Attesa del segnale di misura con fronte di salita/discesa del tastatore nel tratto di percorso verso X = 100.
N210 STOPRE	;Stop preelaborazione
N220 IF \$AC_MEA[1] == FALSE gotof END	;Controllare la buona riuscita della misura.
N230 R10 = \$AA_MM1[X]	;Memorizzare il valore del sistema di misura 1 con fronte di salita.
N240 R11 = \$AA_MM2[X]	;Memorizzare il valore del sistema di misura 2 con fronte di salita.
N250 R12 = \$AA_MM3[X]	;Memorizzare il valore del sistema di misura 1 con fronte di discesa.
N260 R13 = \$AA_MM4[X]	;Memorizzare il valore del sistema di misura 2 con fronte di discesa.
N270 END;	

Esempio di misure con cancellazione del percorso residuo nel modo 2

(acquisizione nella sequenza programmata)

...	
N100 MEASA[X] = (2,1,-1,2,-2) G01 X100 F100	;Misura in modo 2 con sistema di misura attivo. Attesa del segnale di misura nella sequenza: fronte di salita del tastatore 1, fronte di discesa del tastatore 1, fronte di salita del tastatore 2, fronte di discesa del tastatore 2 nel tratto di percorso verso X = 100.
N110 STOPRE	;Stop preelaborazione
N120 IF \$AC_MEA[1] == FALSE gotof	;Controllare la buona riuscita della misura con il tastatore 1.
TASTATORE_DI_MISURA2	
N130 R10 = \$AA_MM1[X]	;Memorizzare il valore di misura relativo al primo evento di trigger programmato (fronte di salita tastatore di misura 1).
N140 R11 = \$AA_MM2[X]	;Memorizzare il valore di misura relativo al secondo evento di trigger programmato (fronte di salita tastatore di misura 1).
N150 TASTATORE_DI_MISURA2:	
N160 IF \$AC_MEA[2] == FALSE gotof END	;Controllare la buona riuscita della misura con il tastatore 2.

<pre>N170 R12 = \$AA_MM3[X]</pre>	<pre>;Memorizzare il valore di misura ;relativo al terzo evento ;di trigger programmato ;(fronte di salita tastatore di misura 2).</pre>
<pre>N180 R13 = \$AA_MM4[X]</pre>	<pre>;Memorizzare il valore di misura ;relativo al quarto evento ;di trigger programmato ;(fronte di salita tastatore di misura 2).</pre>
<pre>N190 END:</pre>	

Esempio di misura continua nel modo 1

(acquisizione in sequenza temporale)

a) Misura di fino a 100 valori di misura

<pre>... N110 DEF REAL VALORE_MISURA[100] N120 DEF INT Loop = 0 N130 MEAC [X] = (1,1,-1) G01 X1000 F100</pre>	<pre>;Misure in modo 1 con sistema di misura ;attivo, memorizzazione dei valori in ;\$AC_FIFO1, attesa del ;segnale di misura con fronte di discesa del ;tastatore 1 nel tratto di percorso verso ;X = 1000.</pre>
<pre>N135 STOPRE N140 MEAC[X] = (0)</pre>	<pre>;Interruzione della misura al raggiungimento ;della posizione dell'asse.</pre>
<pre>N150 R1 = \$AC_FIFO1[4]</pre>	<pre>;Memorizzare nel parametro R1 i valori di ;misura rilevati.</pre>
<pre>N160 FOR Loop = 0 TO R1-1 N170 VALORE_MISURA[Loop] = \$AC_FIFO1[0]</pre>	<pre>;Leggere e memorizzare i valori di misura da ;\$AC_FIFO1.</pre>
<pre>N180 ENDFOR</pre>	

Esempio di misura continua nel modo 1

(acquisizione in sequenza temporale)

b) Misura con cancellazione del percorso residuo secondo 10 valori

<pre>... N10 WHEN \$AC_FIFO1[4] >= 10 DO MEAC[x] = (0) DELDTG (x)</pre>	<pre>;Cancellazione del percorso residuo</pre>
<pre>N20 MEAC[x] = (1,1,1,-1) G01 X100 F500</pre>	
<pre>N30 MEAC [X] = (0)</pre>	
<pre>N40 R1=\$AC_FIFO1[4]</pre>	<pre>;Numero dei valori di misura</pre>
<pre>...</pre>	

Descrizione

La programmazione può avvenire nel partprogram **oppure** nelle azioni sincrone (vedere capitolo "Azioni sincrone al movimento"). Per ogni asse, nello stesso istante, può essere attiva una sola funzione di misura.

Nota

L'avanzamento deve essere adeguato al tipo di misura.

Con MEASA e MEAWA possono essere garantiti risultati di misura affidabili solo con avanzamenti che non generino più di un evento di trigger uguale e non più di 4 diversi eventi di trigger per ogni clock del regolatore di posizione.

Con misure continue tramite MEAC, il rapporto tra il clock di interpolazione e quello di regolazione non può essere superiore a 8 : 1.

Eventi di trigger

Un evento di trigger è costituito dal numero di tastatori di misura e dal criterio di esecuzione (fronte di salita o di discesa) del segnale di misura.

Per ogni misura possono essere elaborati fino a 4 eventi di trigger del relativo tastatore, quindi fino a due tastatori di misura e per ognuno due fronti. La sequenza dell'elaborazione ed il numero massimo degli eventi di trigger dipendono perciò dal modo selezionato.

Nota

Uno stesso evento di trigger può essere programmato solo una volta per funzione di misura (valido per il modo 1)!

Modo operativo

Con la prima cifra del modo si seleziona il sistema di misura desiderato. Se è presente un solo sistema di misura ma viene programmato comunque il secondo, viene utilizzato automaticamente quello esistente.

Con la seconda cifra del **modo di misura**, avviene l'adeguamento della sequenza di misura alle possibilità del relativo controllo.

- **Modo 1:** La rilevazione degli eventi di trigger avviene nella sequenza **temporale** in cui si sono verificati. In questo modo, impiegando moduli a sei assi, è possibile programmare solo un evento di trigger oppure impostando più eventi di trigger avviene la commutazione automatica al modo 2 (senza segnalazione).
- **Modo 2:** La rilevazione degli eventi di trigger avviene nella sequenza in cui sono stati **programmati**.
- **Modo 3:** La rilevazione degli eventi di trigger avviene nella sequenza in cui sono stati **programmati**, tuttavia non c'è alcuna verifica dell'evento di trigger 1 allo START.

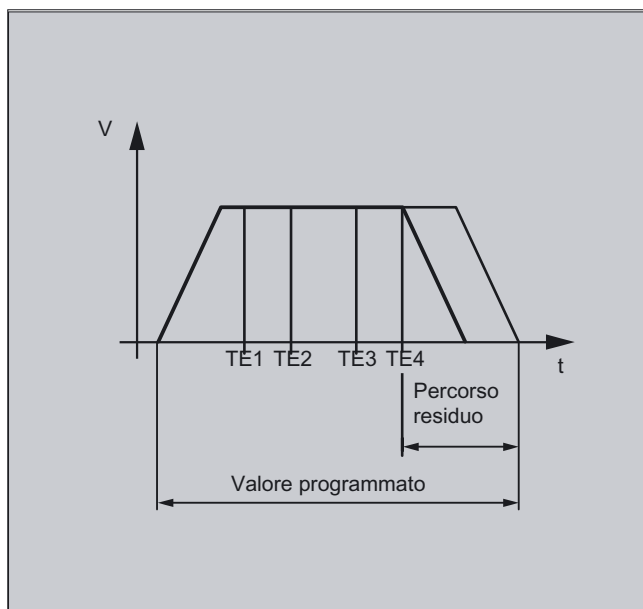
Nota

Utilizzando 2 sistemi di misura si possono programmare solo due eventi di trigger.

Misure con e senza cancellazione del percorso residuo, MEASA, MEAWA

Programmando **MEASA**, il percorso residuo viene cancellato solo dopo l'acquisizione dei valori di misura richiesti.

Per particolari misure, per le quali è necessario raggiungere in ogni caso la posizione programmata, viene utilizzata la funzione **MEAWA**.



- **MEASA** non è programmabile nelle azioni sincrone. Al suo posto, in una azione sincrona, si può utilizzare **MEAWA** seguito dalla cancellazione del percorso residuo
- Se la funzione di misura con **MEAWA** viene startata in un'azione sincrona, i valori di misura sono disponibili solo nel sistema di coordinate macchina.

Risultati delle misure per MEASA, MEAWA

I risultati di misura sono disponibili nelle seguenti variabili:

- Nel sistema di coordinate macchina:

\$AA_MM1 [Asse]	valore misurato del sistema di misura programmato con l'evento di trigger 1
...	...
\$AA_MM4 [Asse]	valore misurato del sistema di misura programmato con l'evento di trigger 4

- Nel sistema di coordinate pezzo:

\$AA_WM1 [Asse]	valore misurato del sistema di misura programmato con l'evento di trigger 1
...	...
\$AA_WM4 [Asse]	valore misurato del sistema di misura programmato con l'evento di trigger 4

Nota

Nella lettura di queste variabili non viene generato internamente alcuno stop preelaborazione. Con STOPRE (capitolo "Lista delle istruzioni") nel programma NC si deve programmare uno stop preelaborazione nella posizione più idonea. Altrimenti vengono letti dei valori errati.

Se la misura assiale deve essere eseguita per un asse geometrico, è necessario programmare la stessa funzione di misura in modo esplicito anche per i restanti assi geometrici. La stessa cosa vale per gli assi coinvolti in una trasformazione.

Esempio:

```
N10 MEASA[Z] = (1,1) MEASA[Y] = (1,1) MEASA[X] = (1,1) G0 Z100;
```

oppure

```
N10 MEASA[Z] = (1,1) POS[Z] = 100
```

Misura con 2 sistemi di misura

Quando una misura viene eseguita con due sistemi di misura, ognuno dei due eventi di trigger possibili viene rilevato da entrambi i sistemi di misura del relativo asse. L'occupazione delle variabili riservate è definita nel seguente modo:

\$AA_MM1 [Asse]	opp ure	\$AA_MW1 [Asse]	Valore di misura del sistema di misura 1 con evento di trigger 1
\$AA_MM2 [Asse]	opp ure	\$AA_MW2 [Asse]	Valore di misura del sistema di misura 2 con evento di trigger 1
\$AA_MM3 [Asse]	opp ure	\$AA_MW3 [Asse]	Valore di misura del sistema di misura 1 con evento di trigger 2
\$AA_MM4 [Asse]	opp ure	\$AA_MW4 [Asse]	Valore di misura del sistema di misura 2 con evento di trigger 2

Stato del tastatore di misura rilevabile con \$A_PROBE[n]

n=tastatore di misura

1==tastatore di misura flessio

0==tastatore di misura non flessio

Stato della funzione di misura con MEASA, MEAWA

Se nel programma è necessario eseguire un'analisi, si può interrogare lo stato della misura tramite \$AC_MEA[n], dove n = numero del tastatore di misura. Non appena tutti gli eventi di trigger del tastatore di misura "n" programmati in un blocco sono stati eseguiti, il valore della variabile è = 1. Diversamente il valore è = 0.

Nota

Se la misura viene startata da azioni sincrone, \$AC_MEA non viene più aggiornato. In questo caso devono essere interrogati nuovi segnali di stato del PLC DB(31-48) DBB62 bit 3 oppure la variabile equivalente \$AA_MEAACT["Asse"].

significato:

\$AA_MEAACT==1: misura attiva

\$AA_MEAACT==0: misura non attiva

Bibliografia:

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Misura (M5).

Misura continua MEAC

Utilizzando **MEAC**, i valori di misura vengono rilevati nel sistema di coordinate macchina e vengono depositati nella memoria FIFO[n] (memoria circolare). Se per la misura sono stati progettati due tastatori, i valori misurati del secondo tastatore vengono depositati separatamente nella memoria FIFO[n+1] aggiuntiva appositamente progettata (impostabile tramite dati macchina).

Il FIFO è una memoria del tipo circolare nella quale vengono inseriti i valori di misura, secondo il principio della rotazione, nelle variabili `$AC_FIFO`, vedere capitolo "Azioni sincrone al movimento".

Nota

Il contenuto FIFO può essere letto solo una volta dalla memoria circolare. Per poter utilizzare più volte i valori di misura, è necessario che gli stessi vengano memorizzati temporaneamente nei dati utente.

Se il numero dei valori di misura per la memoria FIFO supera il numero massimo impostato nel dato macchina, la misura viene automaticamente interrotta.

Per realizzare una serie di misure senza fine, i valori di misura devono essere letti ciclicamente. La lettura deve avvenire almeno con la stessa frequenza di ingresso dei nuovi valori di misura.

Errori di programmazione rilevati

Vengono rilevati i seguenti errori di programmazione e visualizzati con un allarme:

- Se **MEASA/MEAWA** vengono programmati in un blocco con **MEAS/MEAW** viene emessa una segnalazione di errore

Esempio:

```
N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
```

- **MEASA/MEAWA** con numero dei parametri <2 oppure >5

Esempio:

```
N01 MEAWA[X]=(1) G01 F100 POS[X]=100
```

- **MEASA/MEAWA** con evento di trigger diverso da 1/ -1/ 2/ -2

Esempio:

```
N01 MEASA[B]=(1,1,3) B100
```

- **MEASA/MEAWA** con modo errato

Esempio:

```
N01 MEAWA[B]=(4,1) B100
```

- **MEASA/MEAWA** con evento di trigger programmato due volte

Esempio:

```
N01 MEASA[B]=(1,1,-1,2,-1) B100
```

- **MEASA/MEAWA** e asse GEO mancante

Esempio:

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;Asse GEO  
X/Y/Z
```

- Funzione di misura non univoca con assi GEO

Esempio:

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2)  
G01 X50 Y50 Z50 F100
```


5.9 Funzioni speciali per l'utente OEM (OEMIPO1, OEMIPO2, G810 ... G829)

Funzione

Indirizzi OEM

Il significato degli indirizzi OEM viene definito dallo stesso utente OEM. La funzionalità viene implementata con cicli compilati. 5 indirizzi OEM sono riservati. Gli identificatori degli indirizzi sono impostabili. Gli indirizzi OEM sono consentiti in ogni blocco.

Parametri

Gruppi G riservati

Gruppo 1 con OEMIPO1, OEMIPO2

L'utente OEM può definire due ulteriori nomi per le funzioni G OEMIPO1, OEMIPO2 . Questa funzionalità viene implementata con i cicli compilati ed è riservata agli utenti OEM.

- Gruppo 31 con G810 bis G819
- Gruppo 32 con G820 bis G829

Per l'utente OEM sono riservati due gruppi di funzioni G con ognuno 10 funzioni G disponibili. **In questo modo le funzioni implementate dall'utente OEM possono essere utilizzate esternamente.**

Funzioni e sottoprogrammi

Inoltre gli utenti OEM possono anche creare funzioni e sottoprogrammi predefiniti con trasferimento dei parametri.

5.10 Riduzione dell'avanzamento con rallentamento sugli spigoli (FENDNORM, G62, G621)

Funzione

Con il rallentamento automatico sugli spigoli, l'avanzamento viene ridotto a forma di campana poco prima dello spigolo interessato. Inoltre la misura dei comportamenti dell'utensile rilevanti per la lavorazione, può essere parametrizzata tramite dati di setting. Si tratta di:

- Inizio e fine della riduzione dell'avanzamento
- Override con il quale viene ridotto l'avanzamento
- riconoscimento degli spigoli rilevanti

Come spigoli rilevanti vengono intesi quelli il cui angolo interno è inferiore al valore impostato nei dati di setting.

Con il valore di default FENDNORM viene disattivata la funzione dell'override automatico sugli spigoli.

Nota

Questa funzione non appartiene alla fornitura standard del SINUMERIK e deve essere attivata in funzione delle versioni di software.

Bibliografia:

/FBA/ Descrizione delle funzioni, Dialetti ISO.

Programmazione

FENDNORM
G62 G41
oppure
G621

Parametri

FENDNORM	Rallentamento automatico sullo spigolo OFF
G62	Rallentamento d'angolo sugli spigoli interni con correzione raggio utensile attiva
G621	Rallentamento su tutti gli spigoli con correzione raggio utensile attiva

G62 ha effetto solo sugli spigoli interni con

- correzione raggio utensile attiva G41, G42 e
- funzionamento continuo attivo G64, G641

Il relativo spigolo viene raggiunto con l'avanzamento ridotto che deriva da:

$F * (\text{override per la riduzione dell'avanzamento}) * \text{override dell'avanzamento}$

La riduzione max. dell'avanzamento viene raggiunta esattamente quando l'utensile, con traiettoria riferita al centro, deve eseguire il cambio di direzione sul relativo spigolo.

G621 ha un effetto analogo a G62 su ogni spigolo, degli assi definiti con FGROUP.

5.11 Criterio programmabile di fine movimento (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Funzione

In modo analogo al criterio di cambio blocco con interpolazione vettoriale (G601, G602 e G603), il criterio di fine movimento può essere programmato con interpolazione del singolo asse in un partprogram oppure in azioni sincrone per assi comandati da PLC.

5.11 Criterio programmabile di fine movimento (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

In funzione di quale criterio di fine movimento è stato impostato, i blocchi del partprogram o dei cicli tecnologici con movimenti dei singoli assi, vengono conclusi rapidamente in modo diverso. La stessa cosa vale il PLC tramite i blocchi FC15, FC16 e FC18.

Programmazione

FINEA [Asse]

oppure

COARSEA [Asse]

oppure

IPOENDA [Asse]

oppure

IPOBRKA(Asse, [, [valore in percentuale]]) sono consentite impostazioni multiple

oppure

ADISPOSA(Asse, [Int] [, [Real]]) sono consentite impostazioni multiple

Parametri

FINEA	Fine movimento al raggiungimento della soglia di "arresto preciso fine"
COARSEA	Fine movimento al raggiungimento della soglia di "arresto preciso GROSSOLANO"
IPOENDA	Fine movimento al raggiungimento di "Stop interpolatore"
IPOBRKA	Cambio blocco possibile durante la rampa di frenatura (dal SW 6.2)
ADISPOSA	Dimensione della finestra di tolleranza al criterio di fine movimento (dal SW 6.4)
Asse	Nome dell'asse canale (X, Y, ...)
Valore percentuale	Se deve avvenire riferito alla rampa di frenatura al cambio di blocco, deve essere in %
Int	Modo 0: finestra di tolleranza non attiva Modo 1: finestra di tolleranza riferita alla posizione di riferimento Modo 2: finestra di tolleranza riferita alla posizione attuale
Real	Dimensione della finestra di tolleranza. Questo valore viene impostato sincrono al ciclo di elaborazione principale nel dato di setting 43610: impostare ADISPOSA_VALUE

Esempio di fine movimento al raggiungimento dello stop interpolatore

```
...
N110 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 IPOENDA[X]
      Movimento alla posizione X100 con una velocità vettoriale di 1000 giri/min, con un valore
      di accelerazione del 90% e fine del movimento al raggiungimento dello stop interpolatore.
...
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140 IPOENDA[X]
      Movimento alla posizione X50 quando l'ingresso 1 è attivo, con una velocità vettoriale di
      2000 giri/min, un valore di accelerazione del 140% e fine del movimento al
      raggiungimento dello stop interpolatore.
...
```

Esempio di criterio di cambio blocco, rampa di frenatura nel partprogram

```
      ; impostazione di default attiva
N40 POS[X]=100
      ;il cambio blocco avviene quando l'asse X ha raggiunto la posizione 100
      ;e la soglia di arresto preciso fine.
N20 IPOBRKA(X,100)      ;attivazione del criterio di cambio blocco con rampa di frenatura
N30 POS[X]=200          ;il cambio blocco avviene non appena l'asse X inizia a frenare
N40 POS[X]=250
      ;l'asse X non frena alla posizione 200 ma prosegue fino alla
      ;posizione 250, il cambio blocco avviene
      ;non appena l'asse X inizia a frenare
N50 POS[X]=0            ;l'asse X frena e ritorna alla posizione 0; il cambio blocco
      ;avviene alla posizione 0 con soglia di arresto preciso fine
N60 X10 F100
N70 M30
...
```

Esempio di criterio di cambio blocco, rampa di frenatura in azioni sincrone

Nel ciclo tecnologico:	
FINEA	;criterio di fine blocco come arresto preciso fine
POS[X]=100	;il cambio blocco del ciclo tecnologico avviene quando l'asse X ha raggiunto la posizione 100 e la soglia di arresto preciso fine.
IPOBRKA(X,100)	;attivazione del criterio di cambio blocco con rampa di frenatura
POS[X]=100	;POS[X]=100; il cambio blocco del ciclo tecnologico avviene, non appena l'asse X inizia a frenare
POS[X]=250	;l'asse X non frena alla posizione 200 ma prosegue fino alla posizione 250, il cambio blocco nel ciclo tecnologico avviene non appena l'asse X inizia a frenare
POS[X]=250	;l'asse X frena e ritorna alla posizione 0; il cambio blocco avviene alla posizione 0 con soglia di arresto preciso fine
M17	

Descrizione**Variabile di sistema \$AA_MOTEND**

Il criterio di fine movimento impostato può essere interrogato tramite la variabile di sistema \$AA_MOTEND [asse] .

\$AA_MOTEND [Asse] =1	Fine movimento con "arresto preciso fine"
\$AA_MOTEND [Asse] =2	Fine movimento con "arresto preciso grossolano"
\$AA_MOTEND [Asse] =3	Fine movimento con "Stop-IPO".
\$AA_MOTEND [Asse] =4	Criterio di cambio blocco, rampa di frenatura del movimento asse
\$AA_MOTEND [Asse] =5	Cambio blocco nella rampa di frenatura con finestra di tolleranza riferita alla "posizione di riferimento".
\$AA_MOTEND [Asse] =6	Cambio blocco nella rampa di frenatura con finestra di tolleranza riferita alla "posizione reale".

Nota

Dopo RESET viene mantenuto l'ultimo valore programmato.

Bibliografia:

/FB1/ Manuale delle funzioni di base; Avanzamenti (V1).

Criterio di cambio blocco durante la rampa di frenatura

Il valore percentuale viene impostato sincrono all'elaborazione principale nel DS 43600: IPOBRAKE_BLOCK_EXCHANGE . Se non viene introdotto nessun valore, viene attivato quello attualmente impostato in questo dato di setting. Si può impostare un campo da 0% a 100%.

Ulteriore finestra di tolleranza per IPOBRKA

Oltre al criterio già presente di cambio blocco durante la rampa di frenatura è possibile selezionare una ulteriore finestra di tolleranza per il criterio stesso. L'abilitazione avviene solo quando l'asse,

- come finora, ha raggiunto il valore % impostato della rampa di frenatura e
- la posizione attuale o di riferimento non sono distanti, più di una determinata tolleranza, dalla posizione finale dell'asse nel blocco.

Per ulteriori informazioni sul criterio di cambio blocco degli assi di posizionamento, consultare:

Bibliografia:

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Assi di posizionamento (P2).
/PG/ Manuale di programmazione Concetti fondamentali; "Regolazione dell'avanzamento e movimento del mandrino".

5.12 Blocco parametri servo programmabile (SCPARA)

Funzione

Con SCPARA, il blocco di parametri (composto da DM) può essere programmato nel partprogram e nelle azioni sincrone (finora solo tramite PLC)

DB3n DBB9 Bit3

Per evitare conflitti tra richieste dal PLC e dall'NC, è stato definito un ulteriore bit nelle interconnessioni PLC→NCK:

DB3n DBB9 Bit3 "Impostazione inibita del blocco di parametri tramite SCPARA".

Con l'impostazione del blocco di parametri tramite SCPARA inibita, viene emessa una segnalazione di errore nel caso la stessa venga comunque programmata.

Programmazione

SCPARA[asse]=valore

Parametri

SCPARA	Definizione del blocco di parametri
Asse	Nome dell'asse canale (X, Y, ...)
Valore	Blocco di parametri desiderato (1<= valore <=6)

Nota

Il blocco di parametri attuale può essere interrogato tramite la variabile di sistema\$AA_SCPAR [<Asse>] .

Con G33, G331 o G332 il blocco di parametri più adatto viene selezionato dal controllo.

Nel caso il **blocco di parametri servo** debba essere **commutato** sia dal partprogram, o da azioni sincrone, che dal PLC, il programma PLC utente deve essere opportunamente ampliato.

Bibliografia:

/FB1/ Manuale delle funzioni di base; Avanzamenti (V1),
capitolo "Influenza sull'avanzamento".

Esempio

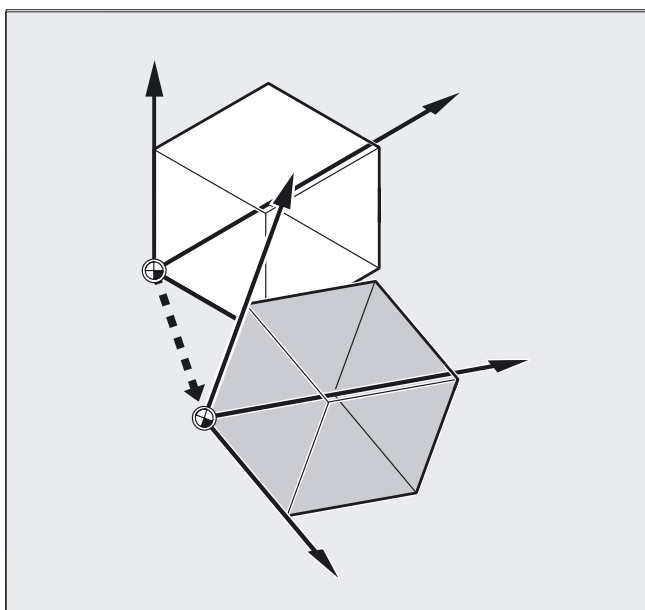
```
...  
N110 SCPARA[X] = 3      ;Per l'asse X viene selezionato il 3. blocco di parametri  
...
```


Frame

6.1 Trasformazione di coordinate con variabili Frame

Funzione

Oltre alle possibilità di programmazione già descritte nel manuale di programmazione "Concetti fondamentali", è possibile determinare la posizione di sistemi di coordinate anche mediante variabili frame predefinite.



Vengono definiti i seguenti sistemi di coordinate:

SCM: Sistema di coordinate macchina

SCB: Sistema di coordinate base

SPZ: Sistema di coordinate del punto zero di base

SZI: Sistema di coordinate del punto zero impostabile

SCP: Sistema di coordinate pezzo

Cos'è una variabile frame predefinita?

Le variabili frame predefinite sono parole chiave, ben definite all'interno del linguaggio del controllo numerico e dotate di un effetto ben preciso e possono essere elaborate in un programma NC.

Variabili frame possibili:

- Frame di base (spostamento base)
- Frame impostabili
- Frame programmabili

Assegnazioni di valori e lettura dei valori reali

Relazione variabile frame/frame

Una trasformazione di coordinate può essere attivata mediante l'assegnazione dei valori di un frame a una variabile frame.

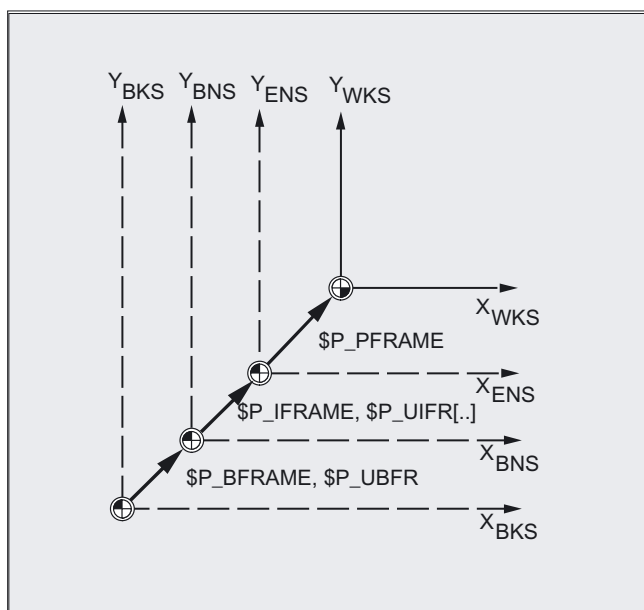
Esempio: `$P_PFRAME=CTTRANS (X, 10)`

Variabile frame:

`$P_PFRAME` significa: frame attuale programmabile.

Frame:

`CTTRANS (X, 10)` significa: spostamento origine programmabile dell'asse X di 10 mm.



Lettura dei valori reali

Mediante variabili predefinite nel partprogram è possibile leggere i valori reali attuali dei sistemi di coordinate:

`$AA_IM[asse]`: lettura valore reale nel MKS

`$AA_IB[asse]`: Lettura valore reale nel SCB

`$AA_IBN[asse]`: lettura valore reale nel BNS

`$AA_IEN[asse]`: lettura valore reale nel ENS

`$AA_IW[asse]`: lettura valore reale nel WKS

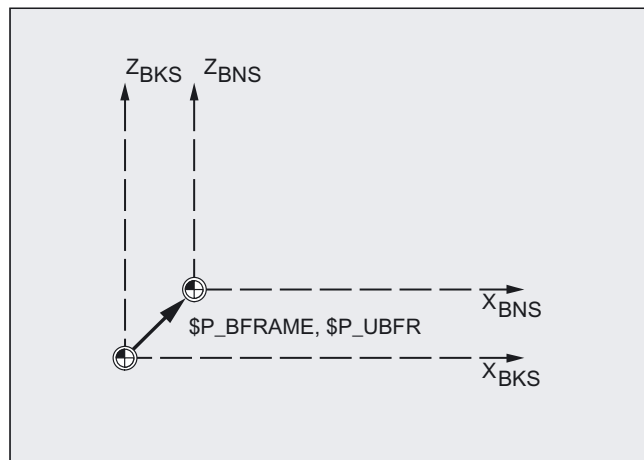
6.1.1 Variabili frame predefinite (\$P_BFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME)

\$P_BFRAME

Variabile frame di base impostabile che crea il riferimento tra il sistema di coordinate di base (BKS) e il sistema di coordinate del punto zero di base (BNS).

Se il frame di base descritto con \$P_UBFR deve diventare immediatamente attivo nel programma, occorre

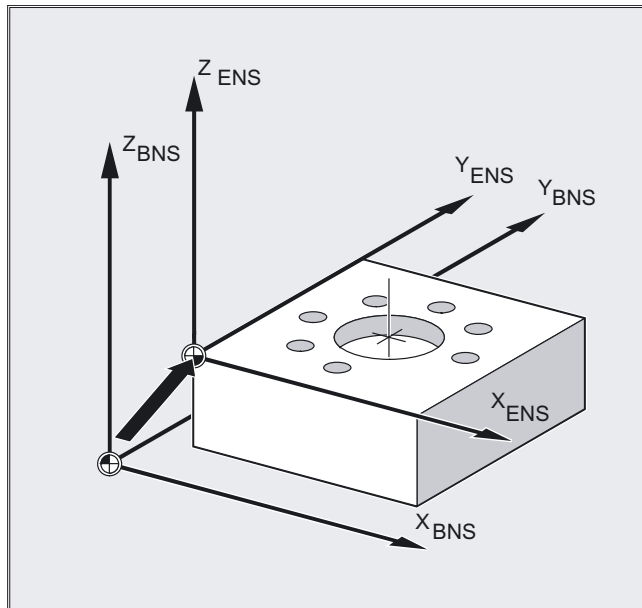
- programmare un G500, G54 . . . G599 oppure
- descrivere \$P_BFRAME con \$ \$P_UBFR .



\$P_IFRAME

Variabile frame attuale impostabile che crea il riferimento tra il sistema del punto zero di base (BNS) e il sistema del punto zero impostabile (ENS).

- \$P_IFRAME corrisponde a \$P_UIFR[\$P_IFRNUM]
- \$P_IFRAME contiene, avendo programmato per es. G54, la traslazione, la rotazione, il fattore di scala e la specularità definite tramite G54 .

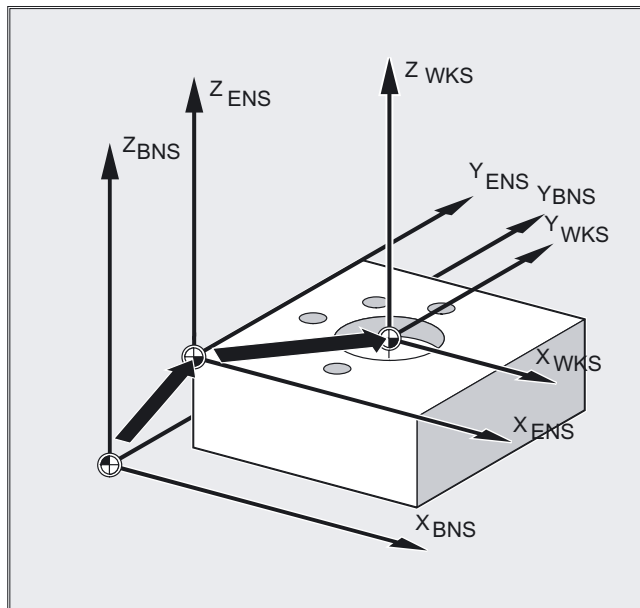


\$P_PFRAME

Variabile frame attuale programmabile che crea il riferimento fra il sistema di punti zero impostabile (ENS) e il sistema di coordinate del pezzo (WKS).

\$P_PFRAME contiene il frame risultante

- **dalla programmazione** di TRANS/ATRANS, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR oppure
- **dalla assegnazione** di CTRANS, CROT, CMIRROR, CSCALE al FRAME programmabile

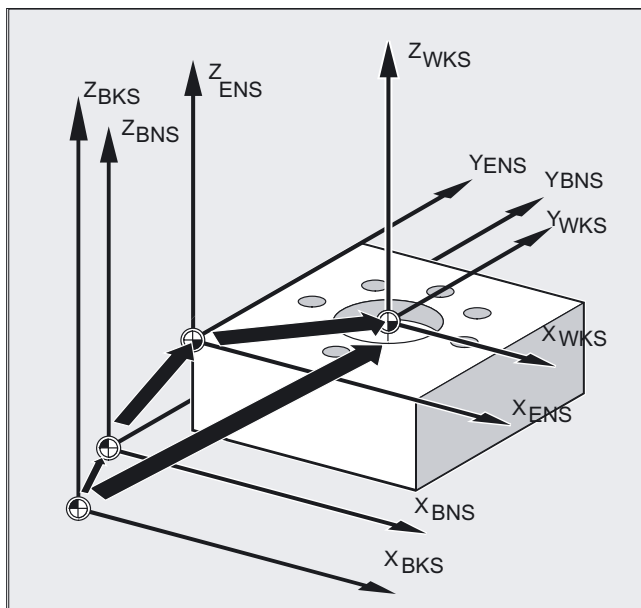


\$P_ACTFRAME

Frame generale attuale risultante, costituito dalle

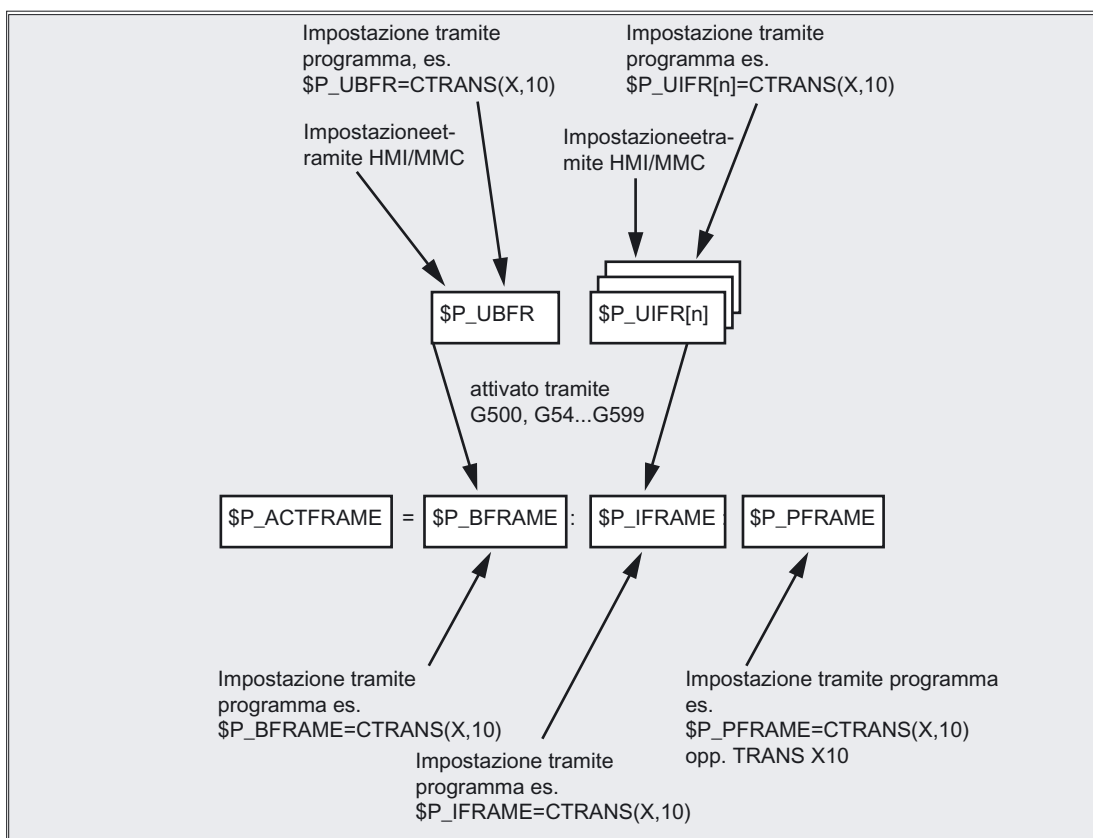
- variabili frame attuali \$P_BFRAME,
 - dalle variabili frame attuali impostabili \$P_IFRAME con frame di sistema e
 - dalle variabili frame attuali programmabili \$P_PFRAME con frame di sistema
- . Frame di sistema, vedere capitolo "Frame attivi nel canale"

\$P_ACTFRAME descrive il punto zero attualmente valido del pezzo.



Nel caso che $\$P\_BFRAME$, $\$P\_IFRAME$ oppure $\$P\_PFRAME$ engano modificati, viene ricalcolata anche $\$P\_ACTFRAME$.

$\$P\_ACTFRAME$ corrisponde a $\$P\_BFRAME:\$P\_IFRAME:\$P\_PFRAME$



Frame di base e frame impostabile diventano attivi dopo un reset, se il DM 20110 RESET_MODE_MASK è impostato nel seguente modo:

Bit0=1, Bit14=1 --> \$P_UBFR (frame base) vale

Bit0=1, Bit5=1 --> \$P_UIFR[\$P_UIFRNUM] (frame impostabile) vale

Frame predefiniti impostabili \$P_UBFR

Con \$P_UBFR viene programmato il frame di base, che però non diventa contemporaneamente attivo nel partprogram. Il frame di base descritto con \$P_UBFR viene calcolato una volta che:

- è stato attivato il reset e sono settati i bit 0 e 14 del DM RESET_MODE_MASK,
- vengono eseguite le istruzioni G500, G54...G599.

Frame predefiniti impostabili \$P_UIFR[n]

Con la variabile predefinita \$P_UIFR[n] è possibile leggere o scrivere gli spostamenti origine impostabili G54 ... G599 dal partprogram.

Nella struttura tali variabili costituiscono un campo unidimensionale del tipo FRAME con il nome \$P_UIFR[n].

Abbinamento alle istruzioni G

Come default si possono definire 5 frame impostabili \$P_UIFR[0]...\$P_UIFR[4] oppure 5 istruzioni G equivalenti G500 e G54 ... G57, che con i relativi indirizzi si possono memorizzare valori.

\$P_IFRAME=\$P_UIFR[0] corrisponde a G500

\$P_IFRAME=\$P_UIFR[1] corrisponde a G54

\$P_IFRAME=\$P_UIFR[2] corrisponde a G55

\$P_IFRAME=\$P_UIFR[3] corrisponde a G56

\$P_IFRAME=\$P_UIFR[4] corrisponde a G57

Mediante dati macchina è possibile modificare il numero dei frame:

\$P_IFRAME=\$P_UIFR[5] corrisponde a G505

... ..

\$P_IFRAME=\$P_UIFR[99] corrisponde a G599

Nota

In tal modo è possibile generare fino a 100 sistemi di coordinate, che possono essere richiamati indipendentemente dal programma, per esempio come punto zero per diversi attrezzaggi.



Cautela

La programmazione di variabili frame e di frame richiede un blocco a sé stante all'interno del programma NC. **Eccezione:** la programmazione di un frame impostabile con G54, G55, ...

6.2 Variabili frame / assegnazione di valori frame

6.2.1 Assegnazione di valori diretti (valore asse, angolo, scala)

Funzione

Queste funzioni permettono di assegnare valori a frame o a variabili di frame direttamente nel programma NC.

Programmazione

```
$P_PFRAME=CTRANS (X, valore asse, Y, valore asse, Z, valore asse,
...)
```

oppure

```
$P_PFRAME=CROT (X, angolo, Y, angolo, Z, angolo, ...)
```

oppure

```
$P_UIFR[...]=CROT (X, angolo, Y, angolo, Z, angolo, ...)
```

oppure

```
$P_PFRAME=CSCALE (X, scala, Y, scala, Z, scala, ..)
```

oppure

```
$P_PFRAME=CMIRROR (X, Y, Z)
```

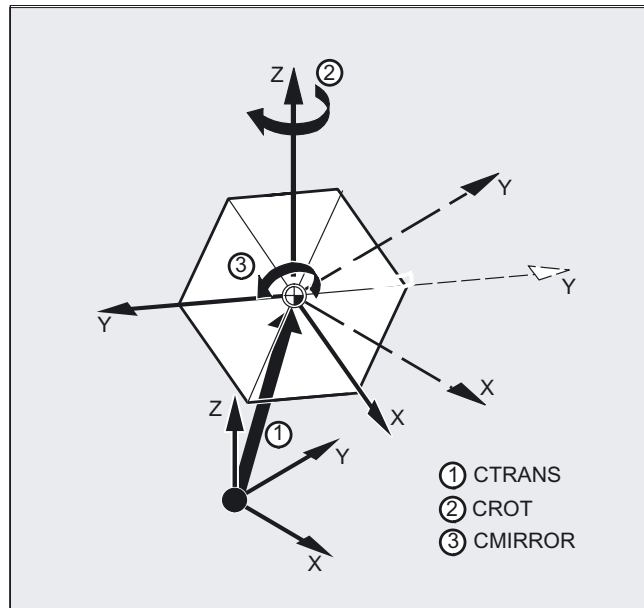
La programmazione di \$P_BFRAME avviene allo stesso modo di \$P_PFRAME.

Parametri

CTRANS	Spostamento degli assi indicati
CROT	Rotazione sugli assi indicati
CSCALE	Modifica della scala per gli assi indicati
CMIRROR	Inversione di direzione degli assi indicati
X Y Z	valore di spostamento in direzione dell'asse geometrico indicato
Valore asse	Assegnazione del valore asse della traslazione
Angolo	Assegnazione dell'angolo di rotazione agli assi indicati
Scala	Modificare la scala

Esempio

Assegnando dei valori ai frame attuali programmabili vengono attivate la traslazione, la rotazione e la specularità.



```
N10 $P_PFRAME=CTRANS (X,10,Y,20,Z,5):CROT (Z,45):CMIRROR (Y)
```

Impostare con altri valori i componenti ROT del frame

Impostare con CROT tutti e tre i componenti di UIFR

```
$P_UIFR[5]=CROT (X, 0, Y, 0, Z, 0)
```

```
N100 $P_UIFR[5, y, rt]=0
```

```
N100 $P_UIFR[5, x, rt]=0
```

```
N100 $P_UIFR[5, z, rt]=0
```

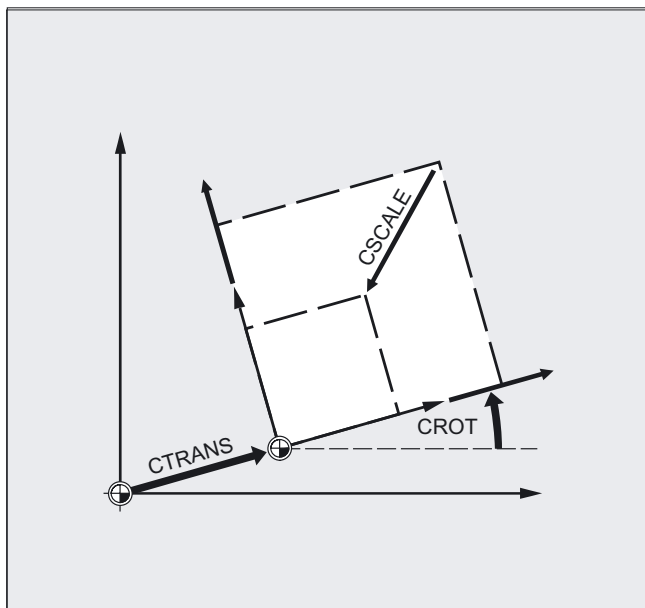
Descrizione

È possibile programmare più regole di calcolo l'una dopo l'altra.

Esempio:

```
$P_PFRAME=CTRANS (...):CROT (...):CSCALE...
```

È importante tenere conto del fatto che i comandi devono essere concatenati l'uno con l'altro con il due punti (...):(...). In tal modo i comandi vengono collegati fra di loro e poi eseguiti nella sequenza programmata in modo additivo.



Nota

I valori programmati con le istruzioni citate vengono assegnati ai frame e memorizzati.

I valori diventano attivi solo dopo essere stati assegnati ad una variabile frame attiva
`$P_BFRAME` oppure `$P_PFRAME`.

6.2.2 Leggere e modificare le componenti frame (TR, FI, RT, SC, MI)

Funzione

Viene data la possibilità di accedere a **singolo** dati di un frame, per es. ad un determinato valore di traslazione o angolo di rotazione. Tali valori possono quindi essere modificati o assegnati ad un'altra variabile.

Programmazione

```
R10=$P_UIFR[$P_UIFNUM, X, RT]  
]
```

L'angolo di rotazione RT dell'asse X dello spostamento origine impostabile attualmente valido \$P_UIFRNUM deve essere assegnato alla variabile R10.

`R12=$P_UIFR[25, Z, TR]`

Il valore di spostamento TR in Z del blocco dati del frame impostabile numero 25 deve essere assegnato alla variabile R12.

`R15=$P_PFRAME[Y, TR]`

Il valore di spostamento TR in Y del frame impostabile attuale deve essere assegnato alla variabile R15.

`$P_PFRAME[X, TR]=25`

Il valore di spostamento TR in X del frame programmabile attuale deve essere modificato. A partire da subito vale X25.

Parametri

<code>\$P_UIFRNUM</code>	Con questa variabile viene creata automaticamente la relazione con lo spostamento origine impostabile attualmente valido.
<code>P_UIFR[n, ..., ...]</code>	Indicando il numero di frame n è possibile accedere al numero di frame n impostabile.
	Impostazione del componente che deve essere letto o modificato:
<code>TR</code>	TR Translation
<code>FI</code>	FI Translation Fine
<code>RT</code>	RT Rotation
<code>SC</code>	SC Scale, variazione della scala
<code>MI</code>	MI Specularità
<code>X Y Z</code>	Inoltre (vedere esempio) viene indicato il relativo asse X, Y, Z.

Campo dei valori per la rotazione RT

Rotazione intorno al 1° asse geometrico:	-180° ... +180°
Rotazione intorno al 2° asse geometrico:	-90° ... +90°
Rotazione intorno al 3° asse geometrico:	-180° ... +180°

Descrizione

Richiamo di un frame

Impostando le variabili di sistema `$P_UIFRNUM` è possibile accedere direttamente allo spostamento origine attualmente impostato mediante `$P_UIFR` oppure `G54, G55, ...` (`$P_UIFRNUM` contiene il numero del frame rappresentato attualmente).

Tutti gli altri frame `$P_UIFR` impostabili memorizzati vanno richiamati impostando il corrispondente numero `$P_UIFR[n]`.

Per le variabili frame predefinite e per frame definiti singolarmente va impostato il nome, per es. \$P_IFRAME.

Richiamare i dati

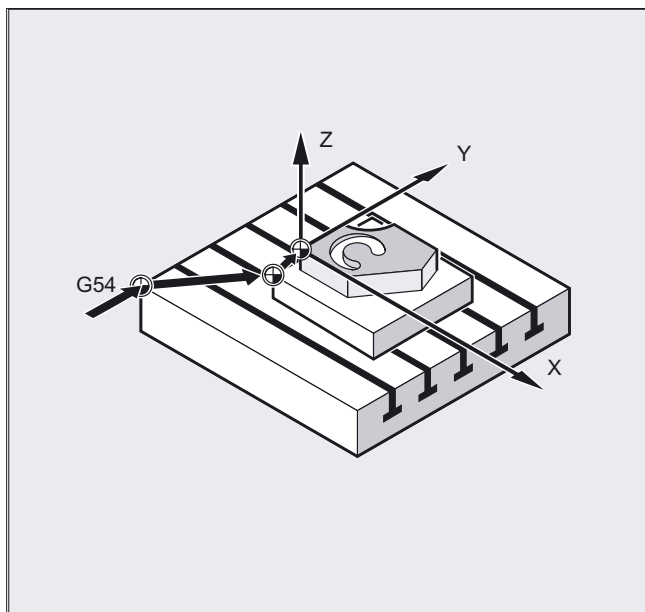
Nelle parentesi quadre sono contenuti il nome dell'asse e il componente frame del valore a cui si intende accedere o che si desidera modificare, per es. [X, RT] oppure [Z, MI].

6.2.3 Combinazione di frame completi

Funzione

Nel partprogram un frame completo può essere assegnato a un altro frame oppure si possono concatenare frame.

I concatenamenti di frame sono opportuni per es. quando si tratta di descrivere diversi pezzi posizionati su un pallet e che devono essere lavorati nell'ambito di una sequenza di produzione.



Per la descrizione di lavorazioni con pallet i componenti frame possono contenere soltanto determinati valori parziali attraverso il cui concatenamento vengono generati diversi punti zero del pezzo.

Programmazione

Assegnazione di frame

```
DEF FRAME IMPOSTAZIONE1
IMPOSTAZIONE1=CTTRANS (X, 10)
$P_PFRAME=IMPOSTAZIONE1

DEF FRAME IMPOSTAZIONE4
IMPOSTAZIONE4=$P_PFRAME
$P_PFRAME=IMPOSTAZIONE4
```

Al frame programmabile attuale vengono assegnati i valori del frame autodefinito IMPOSTAZIONE1.

Il frame programmabile attuale viene memorizzato nella memoria intermedia e su richiesta rimemorizzato.

Catene di frame

I frame vengono concatenati nella sequenza in cui sono stati programmati, i componenti frame invece, come ad esempio spostamenti, rotazioni, ecc.), vengono eseguiti in modo additivo uno dopo l'altro.

```
$P_IFRAME=$P_UIFR[15]:$P_UIFR[16]
```

\$P_UIFR[15] contiene ad esempio dati per gli spostamenti origine. Al termine vengono elaborati, sulla base degli stessi, i dati di \$P_UIFR[16], es. dati per rotazioni.

```
$P_UIFR[3]=$P_UIFR[4]:$P_UIFR[5]
```

Il frame impostabile 3 viene generato concatenando i frame impostabili 4 e 5.

Nota

Occorre osservare che i frame devono essere collegati insieme con l'operatore di concatenamento due punti.

6.2.4 Definizione di nuovi frame (DEF FRAME)

Funzione

Oltre ai frame impostabili predefiniti di cui si è parlato finora è possibile generare anche nuovi frame. Si tratta in tal caso di variabili del tipo FRAME che vanno definite con un nome a libera scelta.

Con le funzioni CTRANS, CROT, CSCALE e CMIRROR è possibile assegnare dei valori ai frame nel programma NC.

Programmazione

```
DEF FRAME PALETTE1
oppure
PALETTE1=CTTRANS (...) : CROT (...) ...
```

Parametri

DEF FRAME	Creazione di nuovi frame.
PALETTE1	Nome del nuovo frame
=CTTRANS (...) :	Assegnare i valori alle possibili funzioni
CROT (...) ...	

6.3 Traslazione fine e grossolana (CFINE; CTRANS)

Funzione

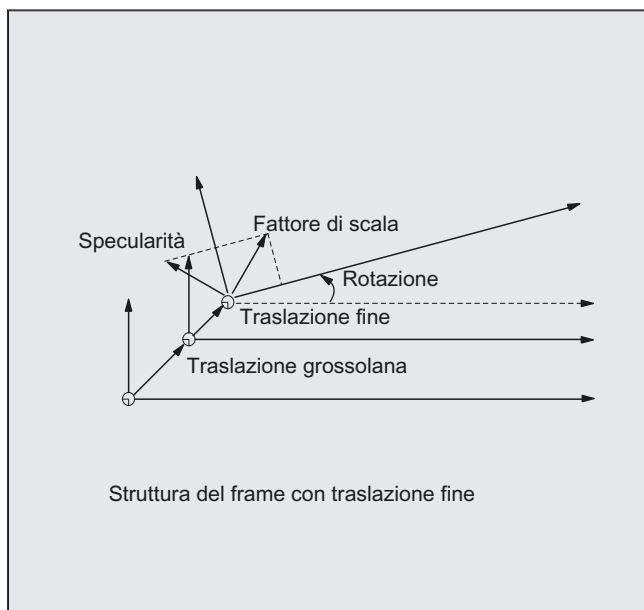
Traslazione fine

Con l'istruzione `CFINE(X, ..., Y ...)` può essere programmata una traslazione fine del frame di base e di tutti i frame impostabili.

Una traslazione fine può essere eseguita solo se il DM18600: `MM_FRAME_FINE_TRANS` è = 1.

Traslazione grossolana

Con `CTTRANS(...)` viene definita la traslazione grossolana.



Le traslazioni grossolane e fini vengono sommate e formano la traslazione complessiva.

Programmazione

```
$P_UBFR=CTTRANS(x, 10) : CFINE(x, 0.1) ;concatenamento di traslazione,
: CROT(x, 45) ;traslazione fine e rotazione

$P_UIFR[1]=CFINE(x, 0.5 y, 1.0, z, ;tutto il frame viene sovrascritto
0.1) ;con CFINE, inclusa
;la traslazione grossolana
```

L'accesso ai singoli componenti della traslazione fine avviene con l'impostazione della FI (Translation Fine).

```
DEF REAL FINEX ;definizione della variabile FINEX
FINEX=$P_UIFR[$P_UIFNUM, x, FI] ;lettura della traslazione fine tramite
;la variabile FINEX

FINEX=$P_UIFR[3, x, FI]$P ;lettura della traslazione fine
;dell'asse X nel 3.Frame
;tramite la variabile FINEX
```

Parametri

CFINE(x, valore, y, valore, z, valore)	Traslazione fine per più assi. Traslazione additiva (Translation).
CTTRANS(x, valore, y, valore, z, valore)	Traslazione grossolana per più assi. Traslazione assoluta (Translation).
x y z	Spostamento origine degli assi (max. 8)
Valore	Entità di traslazione

Costruttore della macchina

Con il DM18600: Con MM_FRAME_FINE_TRANS la traslazione fine può essere progettata nelle seguenti varianti:

0:

la traslazione fine non può essere impostata né programmata. G58 e G59 non sono possibili.

1:

la traslazione fine per frame impostabili, frame di base, frame programmabili, G58 e G59 può essere impostata o programmata

Descrizione

Una traslazione fine modificata dalla superficie operativa HMI, diventa attiva solo dopo l'attivazione del relativo frame, cioè l'attivazione avviene con G500, G54...G599. Una traslazione fine attiva di un frame resta tale finché anche il frame è attivo.

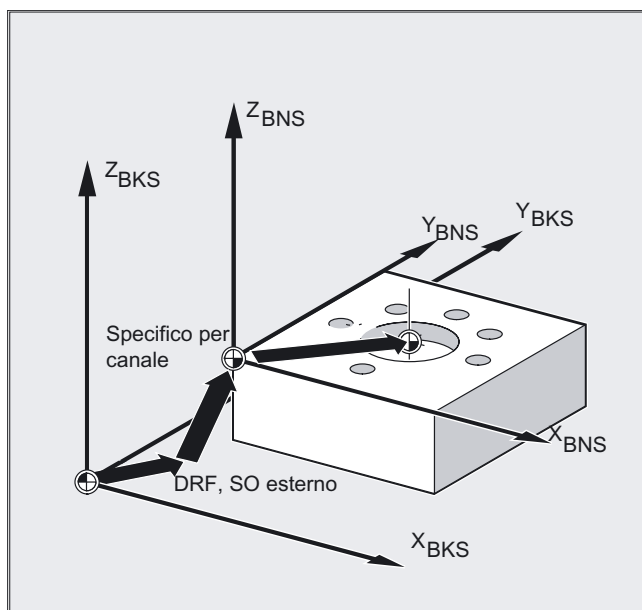
Per un frame programmabile la traslazione fine non è possibile. Se ad un frame programmabile risulta assegnato un frame con traslazione fine, la traslazione complessiva verrà formata dalla somma della traslazione grossolana e fine. Leggendo il frame programmabile la traslazione fine è sempre pari a zero.

6.4 Spostamento DRF

Traslazione con volantino, DRF

Oltre a tutti gli spostamenti trattati nel presente capitolo è anche possibile definire lo spostamento origine con il volantino (spostamento DRF).

Nel sistema di coordinate di base lo spostamento DRF agisce sugli assi geometrici geometrici e aggiuntivi:



L'assegnazione del volantino deve essere tuttavia eseguita per l'asse macchina (p. es. tramite segnali di interconnessione NC/PLC "Attiva volantino"), sul quale viene formato l'asse geometrico e l'asse aggiuntivo. Per ulteriori informazioni vedere il relativo manuale operativo.

Cancellazione della traslazione DRF, DRFOF

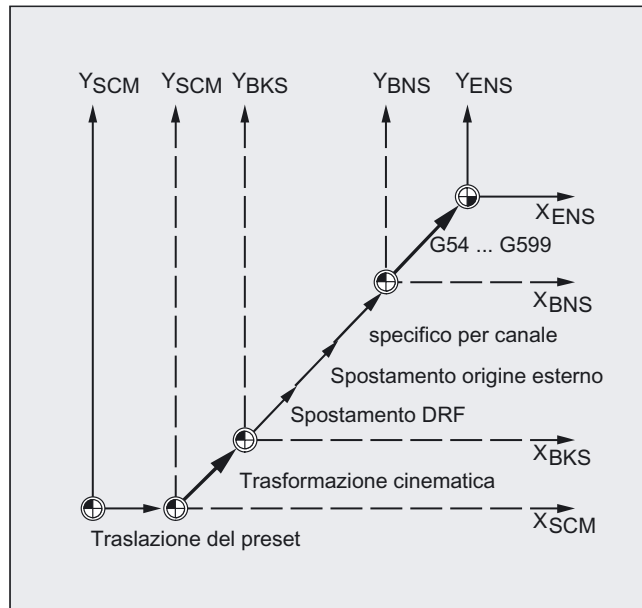
Mediante DRFOF lo spostamento origine da volantino viene cancellato per tutti gli assi assegnati al canale. DRFOF si trova in un blocco NC a sé stante.

6.5 Spostamento origine esterno

Funzione

Lo spostamento origine esterno costituisce un'ulteriore possibilità di spostare il punto zero fra sistema di coordinate di base e sistema di coordinate del pezzo.

Per lo spostamento origine esterno possono essere impostati soltanto spostamenti lineari.



Programmazione

La programmazione dei valori di spostamento, \$AA_ETRANS avviene occupando le variabili di sistema specifiche degli assi.

Assegnazione dei valori di spostamento

\$AA_ETRANS [asse] = RI

R è la variabile di calcolo del tipo REAL che contiene il nuovo valore.

Lo spostamento origine esterno in genere non viene impostato nel partprogram ma viene settato da PLC.

Nota

Il valore scritto nel partprogram diventa attivo soltanto se nell'interfaccia VDI (interfaccia NCU-PLC) è settato il segnale corrispondente.

6.6 Traslazione del preset (PRESETON)

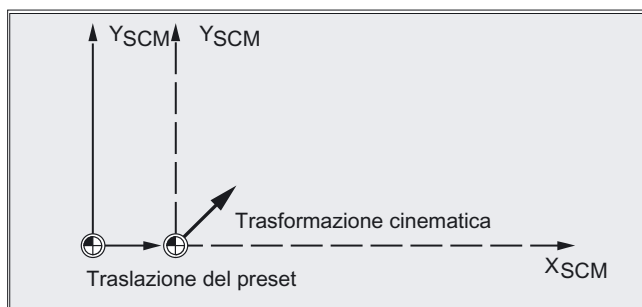
Funzione

Nel caso di impieghi speciali può rendersi necessario assegnare un nuovo valore reale programmato ad uno o più assi sulla posizione attuale (in stato di fermo).



Cautela

Con la funzione PRESETON non è più valido il punto di riferimento. Per questo motivo è bene impiegare questa funzione solo nel caso di assi senza obbligo di punto di riferimento. Se si desidera ripristinare il sistema originario occorre eseguire la ricerca del punto di riferimento con G74 - vedere il capitolo "Gestione dei file e del programma".



Programmazione

PRESETON(asse, valore, ...)

Parametri

PRESETON	Impostazione del valore reale
Asse	Impostazione dell'asse macchina
Valore	Nuovo valore reale relativo all'asse impostato

Nota

L'impostazione del valore attuale con azioni sincrone dovrebbe avvenire solo con la parola chiave "WHEN" o "EVERY".

Esempio

L'assegnazione dei valori reali viene eseguita nel sistema di coordinate di macchina - i valori si riferiscono agli assi macchina.

```
N10 G0 A760
```

```
N20 PRESETON(A1,60)
```

L'asse A si porta sulla posizione 760. Su tale posizione l'asse di macchina A1 riceve il nuovo valore reale 60. A partire da questo momento il posizionamento avverrà nel nuovo sistema di valori reali.

6.7 Calcolo del frame in base a 3 punti di misura nello spazio (MEAFRAME)

Funzione

MEAFRAME è un ampliamento del linguaggio 840D per il supporto dei cicli di misura.

La funzione MEAFRAME calcola il frame risultante dai tre punti ideali e dai punti corrispondenti che vengono misurati.

Se un pezzo viene posizionato per la lavorazione, nell'ambito del sistema di coordinate cartesiane della macchina, la sua posizione rispetto alla posizione ideale di lavorazione risulta sia spostata sia ruotata. Per una lavorazione o una misura precise sono necessarie una taratura impegnativa oppure modifiche dei movimenti nel partprogram.

Un frame può essere definito mediante rilevamento di tre punti nello spazio, dei quali sono note le posizioni ideali. Il rilevamento può avvenire con sensore a contatto oppure ottico che eseguono la tastatura su fori precisi o su sfere di misura situati sulla piastra di supporto.

Programmazione

```
MEAFRAME IDEAL_POINT, MEAS_POINT, FIT_QUALITY)
```

Parametri

MEAFRAME	Calcolo frame con 3 punti di misura nello spazio
IDEAL_POINT	dim. bidimensionale che contiene le tre coordinate dei punti ideali.
MEAS_POINT	dim. campo Real che contiene le tre coordinate dei punti misurati.
FIT_QUALITY	Variabile REAL, che restituisce le seguenti informazioni:
-1:	I punti ideali si trovano quasi su una retta: frame non può essere calcolato. La variabile Frame restituita contiene un Frame neutro.
-2:	I punti di misura si trovano quasi su una retta: frame non può essere calcolato. La variabile Frame restituita contiene un Frame neutro.
-4:	Il calcolo della matrice di rotazione può non andare a buon fine per un altro motivo.
valore positivo:	Somma delle deformazioni (distanze tra i punti), che servono per il trasferimento del triangolo misurato in un triangolo ideale congruente.

Nota**Qualità della misura**

Affinché sia possibile assegnare le coordinate misurate alle coordinate ideali con una traslazione/rotazione combinata, occorre che il triangolo generato dai tre punti sia congruente al triangolo ideale. Ciò si ottiene con l'ausilio di un algoritmo di compensazione, che minimizza la somma dei quadrati degli scostamenti che risultano dal trasferimento del triangolo misurato in quello ideale.

L'alterazione effettivamente necessaria dei punti di misura può essere vista come indicatore per la misura della qualità e viene quindi emessa come ulteriore variabile di `MEAFRAME`.

Nota

Il frame creato da `MEAFRAME` può essere trasformato con la funzione `ADDFRAME` in un altro frame della catena.

Vedere esempio: Concatenamento di Frame "Verkettung mit ADDFRAME".

Per ulteriori informazioni sui parametri di `ADDFRAME(FRAME, STRING)` vedere /FB1/ Manuale delle funzioni di base; Assi, sistemi di coordinate, frame (K2), capitolo "Concatenamento di FRAME".

Esempio

```
; Partprogram 1
;
DEF FRAME CORR_FRAME
;
; Impostazione dei punti di misura
DEF REAL IDEAL_POINT[3,3] = SET(10.0,0.0,0.0, 0.0,10.0,0.0,
0.0,0.0,10.0)
DEF REAL MEAS_POINT[3,3] = SET
(10.1,0.2,-0.2, -0.2,10.2,0.1, -0.2,0.2,9.8); per test
DEF REAL FIT_QUALITY = 0
;
DEF REAL ROT_FRAME_LIMIT = 5 ;ammette una rotazione max. di 5 gradi
;della posizione del pezzo
DEF REAL FIT_QUALITY_LIMIT = 3 ;ammette una traslazione di max. 3 mm tra
;il triangolo ideale e quello misurato
DEF REAL SHOW_MCS_POS1[3]
DEF REAL SHOW_MCS_POS2[3]
DEF REAL SHOW_MCS_POS3[3]
;=====
;
N100 G01 G90 F5000
N110 X0 Y0 Z0
;
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY)
;
N230 IF FIT_QUALITY < 0
SETAL(65000)
GOTOF NO_FRAME
```

6.7 Calcolo del frame in base a 3 punti di misura nello spazio (MEAFRAME)

```

ENDIF
,
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT
SETAL(65010)
GOTOF NO_FRAME
ENDIF
;
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT ;Limitazione del 1. angolo
;RPY
SETAL(65020)
GOTOF NO_FRAME
ENDIF
;
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT ;Limitazione del 2. angolo
;RPY
SETAL(65021)
GOTOF NO_FRAME
ENDIF
;
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT ;Limitazione del 3. angolo
;RPY
SETAL(65022)
GOTOF NO_FRAME
ENDIF
;
N300 $P_IFRAME=CORR_FRAME ; Attivare il frame di tastatura mediante un frame impostabile
;
;verificare il frame posizionando l'asse geometrico sui punti ideali
;
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]
N410 SHOW_MCS_POS1[0]=$AA_IM[X]
N420 SHOW_MCS_POS1[1]=$AA_IM[Y]
N430 SHOW_MCS_POS1[2]=$AA_IM[Z]
;
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]
N510 SHOW_MCS_POS2[0]=$AA_IM[X]
N520 SHOW_MCS_POS2[1]=$AA_IM[Y]
N530 SHOW_MCS_POS2[2]=$AA_IM[Z]
;
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]
N610 SHOW_MCS_POS3[0]=$AA_IM[X]
N620 SHOW_MCS_POS3[1]=$AA_IM[Y]
N630 SHOW_MCS_POS3[2]=$AA_IM[Z]
;

```

```

N700 G500 ; disattivazione del frame impostabile in quanto preimpostato con frame zero
;(nessun valore)
;
NO_FRAME:
M0
M30

```

Esempio di concatenamento di Frame

Concatenamento di MEAFRAME per correzioni

La funzione MEAFRAME () fornisce un frame di correzione. Se questo frame di correzione viene concatenato con il frame impostabile \$P_UIFR[1] che era attivo al richiamo della funzione, es. G54, si ottiene un frame impostabile per ulteriori calcoli del percorso oppure lavorazioni.

Concatenamento con ADDFRAME

Se questo frame di correzione deve avere effetto in un altro punto della catena dei frame oppure se sono attivi altri frame prima di quello impostabile, è possibile utilizzare la funzione ADDFRAME () per il concatenamento in uno dei frame base del canale oppure in un frame di sistema.

Nei frame non possono essere attivi:

- specularità con MIRROR
- fattore di scala con SCALE

I parametri di ingresso per i valori di riferimento ed i valori attuali sono le coordinate del pezzo. Nel sistema di base del controllo, queste coordinate devono essere sempre impostate in

- metrico oppure inch (G71/G70) e
- riferite al raggio (DIAMOF).

6.8 Frame NCU globali

Funzione

In ogni NCU i frame globali NCU esistono una sola volta per tutti i canali. I frame globali NCU possono essere scritti e letti da tutti i canali. L'attivazione dei frame globali NCU avviene nel rispettivo canale.

Con i frame globali si possono scalare e speculare gli **assi canali e gli assi macchina** con traslazioni.

Relazioni geometriche e concatenamenti di frame

Per i frame globali non esiste alcuna correlazione geometrica tra gli assi. Per questo motivo non possono essere eseguite rotazioni né programmazioni di identificatori di assi geometrici.

- Sui frame globali non è possibile applicare le rotazioni. La programmazione di una rotazione viene rifiutata con l'allarme: "18310 Canale %1 Blocco %2 Frame: rotazione non consentita".

- È possibile il concatenamento di frame globali e frame specifici per canale. Il frame risultante contiene tutte le parti di frame incluse le rotazioni per tutti gli assi. L'assegnazione di un frame con parti di rotazione a un frame globale viene rifiutata con l'allarme "Frame: rotazione non consentita".

Frame NCU globali

Frame base NCU globali \$P_NCBFR[n]

È possibile progettare fino a 8 frame base NCU globali.

Contemporaneamente possono essere presenti frame di base specifici per canale.

I frame globali possono essere scritti e letti da tutti i canali di una NCU. Nella scrittura di frame globali l'utente deve fare attenzione alla coordinazione dei canali. Questo può avvenire ad esempio mediante marker di Wait (WAITMC).

Costruttore della macchina

Il numero di frame base globali viene progettato; vedere /FB1/ Manuale delle funzioni di base; Assi, sistemi di coordinate, frame (K2).

Frame globali NCU impostabili \$P_UIFR[n]

Tutti i frame impostabili G500, G54 . . . G599 possono essere progettati come frame globali NCU o specifici per canale.

Costruttore della macchina

Tutti i frame impostabili possono essere riprogettati in frame globali con l'ausilio di un dato macchina \$MN_MM_NUM_GLOBAL_USER_FRAMES.

Come indicatore assi nelle istruzioni per frame si possono utilizzare degli identificatori di assi canale e degli identificatori di assi di macchina. La programmazione di identificatori di assi geometrici viene respinta con un allarme.

6.8.1 Frame specifici per canali (\$P_CHBFR, \$P_UBFR)

Funzione

I frame impostabili oppure i frame di base possono essere letti e scritti

- dal partprogram e
- tramite BTSS

sia dalla superficie operativa HMI Advanced che dal PLC.

La traslazione fine è possibile anche per i frame globali. La soppressione dei frame globali si verifica come per i frame specifici per canale tramite G53, G153, SUPA e G500.

Costruttore della macchina

Tramite il dato macchina DM28081 MM_NUM_BASE_FRAMES si può progettare il numero di frame di base nel canale. La configurazione standard è concepita in modo che vi sia almeno un frame di base per canale. Sono ammessi al massimo 8 frame di base per canale. Oltre agli 8 frame di base possono esservi anche 8 frame di base globali per l'NCU.

Frame specifici per un canale**\$P_CHBFR[n]**

I frame di base si possono leggere e scrivere tramite la variabile di sistema `$P_CHBFR[n]`. In fase di scrittura di un frame di base, il frame di base generale non viene attivato finché non viene eseguita un'istruzione `G500`, `G54`...`G599`. La variabile funge prevalentemente da memoria per i processi di scrittura sul frame di base dall'HMI o dal PLC. Queste variabili frame vengono salvate tramite il salvataggio dati.

Primo frame di base nel canale

Una scrittura sulla variabile predefinita `$P_UBFR` non attiva contemporaneamente il frame di base con l'indice di campo 0; l'attivazione si verifica solo quando viene eseguita un'istruzione `G500`, `G54`...`G599`. La variabile può anche essere scritta e letta nel programma.

\$P_UBFR

`$P_UBFR` è identico a `$P_CHBFR[0]`. Come default è sempre presente un frame base nel canale in modo che la variabile di sistema sia sempre compatibile con le versioni precedenti. Se non vi sono frame di base specifici, durante la scrittura o lettura viene emesso l'allarme "Frame: Istruzione non consentita".

6.8.2 Frame attivi nel canale

Funzione

I frame attivi nel canale vengono impostati con le relative variabili di sistema di questi frame. Questo vale anche per i frame di sistema. Attraverso queste variabili di sistema, nel partprogram è possibile leggere e scrivere il frame di sistema attuale.

Frame attivi attualmente nel canale

Panoramica

Frame di sistema attuali

\$P_PARTFRAME

\$P_SETFRAME

\$P_EXTFRAME

\$P_NCBFRAME[n]

\$P_CHBFRAME[n]

\$P_BFRAME

\$P_ACTBFRAME

\$P_CHBFRMASK e \$P_NCBFRMASK

\$P_IFFRAME

Frame di sistema attuali

\$P_TOOLFRAME

\$P_WPFRAME

\$P_TRAFRAME

\$P_PFRAME

Frame di sistema attuale

\$P_CYCFRAME

P_ACTFRAME

Concatenamento di frame

per:

TCARR e PAROT

impostazione valore reale e sfioro

Spostamento origine esterno

Frame di base globali NCU attuali

Frame di base attuali del canale

1. frame base attuale nel canale

Frame di base globale

Frame di base globale

Frame impostabile attuale

per:

TOROT e TOFRAME

Punti di riferimento del pezzo

Trasformazioni

Frame programmabile attuale

per:

Cicli

Frame attuale globale

Il frame attuale è formato dal frame di base globale

\$P_NCBFRAME[n] Frame di base globali NCU attuali

Gli elementi di campo del frame di base globale attuale possono essere letti e scritti tramite la variabile di sistema `$P_NCBFRAME[n]`. Il frame di base così ottenuto, viene incluso nel canale con il processo di scrittura.

Il frame modificato è attivo solo nel canale in cui è stato programmato. Se il frame deve essere modificato per tutti i canali di una NCU, occorre scrivere contemporaneamente `$P_NCBFR[n]` e `$P_NCBFRAME[n]`. Gli altri canali devono quindi solo attivare il frame, ad esempio con G54. Quando viene scritto un frame di base, viene ricalcolato il frame di base generale.

\$P_CHBFRAME[n] Frame canale attuali

Gli elementi di campo del frame di base canale attuale possono essere letti e scritti tramite la variabile di sistema `$P_CHBFRAME[n]`. Il frame di base così ottenuto viene calcolato nel canale insieme al processo di scrittura. Quando viene scritto un frame di base, viene ricalcolato il frame di base generale.

\$P_BFRAME 1. frame base attuale nel canale

Con la variabile frame predefinita `$P_BFRAME`, nel partprogramm è possibile leggere e scrivere il frame di base attuale con l'indice di campo 0 che è valido nel canale. Il frame di base scritto viene subito calcolato.

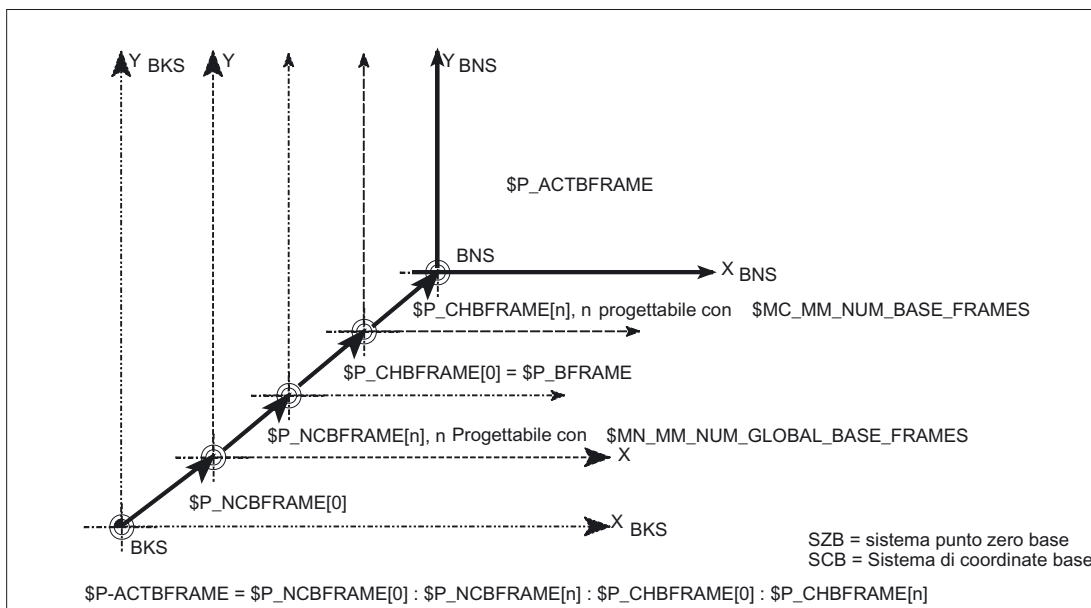
`$P_BFRAME` è identico a `$P_CHBFRAME[0]`. La variabile di sistema, come default ha sempre un valore valido. Se non vi sono frame di base specifici, durante la scrittura o lettura viene emesso l'allarme "Frame: Istruzione non consentita".

\$P_ACTBFRAME Frame base globale

La variabile `$P_ACTBFRAME` definisce il frame di base globale concatenato.. Questa variabile è di sola lettura.

`$P_ACTBFRAME` corrisponde a

`$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n]`.

**\$P_CHBFRMASK e \$P_NCBFRMASK Frame base globali**

Frame di base globale Tramite le variabili di sistema `$P_CHBFRMASK` e `$P_NCBFRMASK` l'utente può scegliere quali frame di base includere nel calcolo del frame di base "globale". Le variabili possono essere programmate solo nel programma e lette tramite l'interfaccia del pannello operativo (BTSS). Il valore delle variabili viene interpretato come maschera di bit e indica quale elemento di campo del frame di base di `$P_ACTBFRAME` rientra nel calcolo.

Con `$P_CHBFRMASK` si può impostare quali frame di base specifici del canale, mentre con `$P_NCBFRMASK`, si definisce quali frame di base globali NCU devono essere calcolati.

Con la programmazione delle variabili vengono ricalcolati il frame di base ed il frame. Dopo un reset e nell'impostazione di base valgono i seguenti valori

`$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK` e

`$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK.`

ad es.

`$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]`

`$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]`

\$P_IFRAME Frame impostabile attuale

Con la variabile frame predefinita `$P_IFRAME`, nel partprogramm è possibile leggere e scrivere il frame impostabile attuale che è valido nel canale. Il frame impostabile scritto viene subito incluso nel calcolo.

Nei frame impostabili globali NCU il frame modificato è attivo solo nel canale in cui il frame è stato programmato. Se il frame deve essere modificato per tutti i canali di una NCU, occorre descrivere contemporaneamente `$P_UIFR[n]` e `$P_IFRAME`. Gli altri canali devono quindi solo attivare il relativo frame, ad esempio con G54.

\$P_PFRAME Frame programmabile attuale

`$P_PFRAME` PFRAME è il frame programmabile risultante dalla programmazione di TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR o dall'assegnazione di CTRANS, CROT, CMIRROR, CSCALE al FRAME programmabile.

Variabile frame attuale programmabile che crea il riferimento fra il sistema di punti zero impostabile

- (ENS) e il
- sistema di coordinate del pezzo (SCP)

.

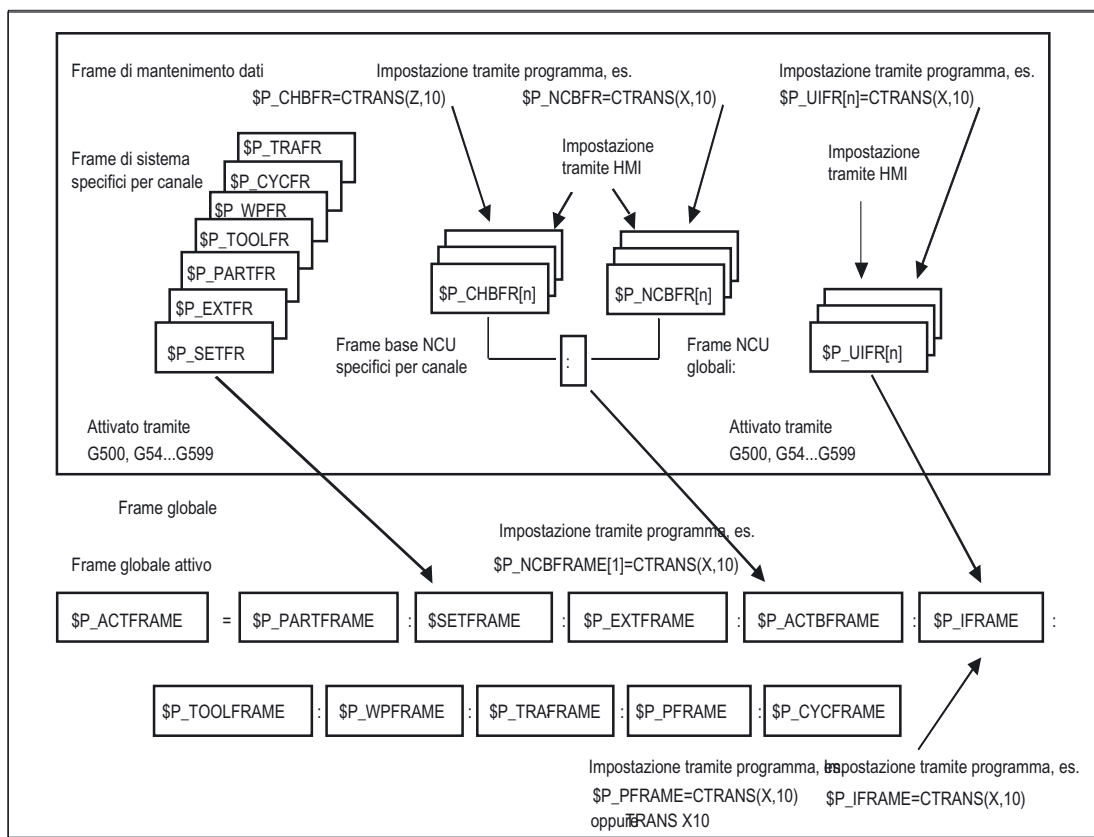
P_ACTFRAME Frame attuale globale

Il frame attuale globale risultante `$P_ACTFRAME` risulta dal concatenamento di tutti i frame di base, dal frame attuale impostabile e dal frame programmabile. Il frame attuale viene aggiornato ogni volta che si modifica una parte di frame.

`$P_ACTFRAME` corrisponde a

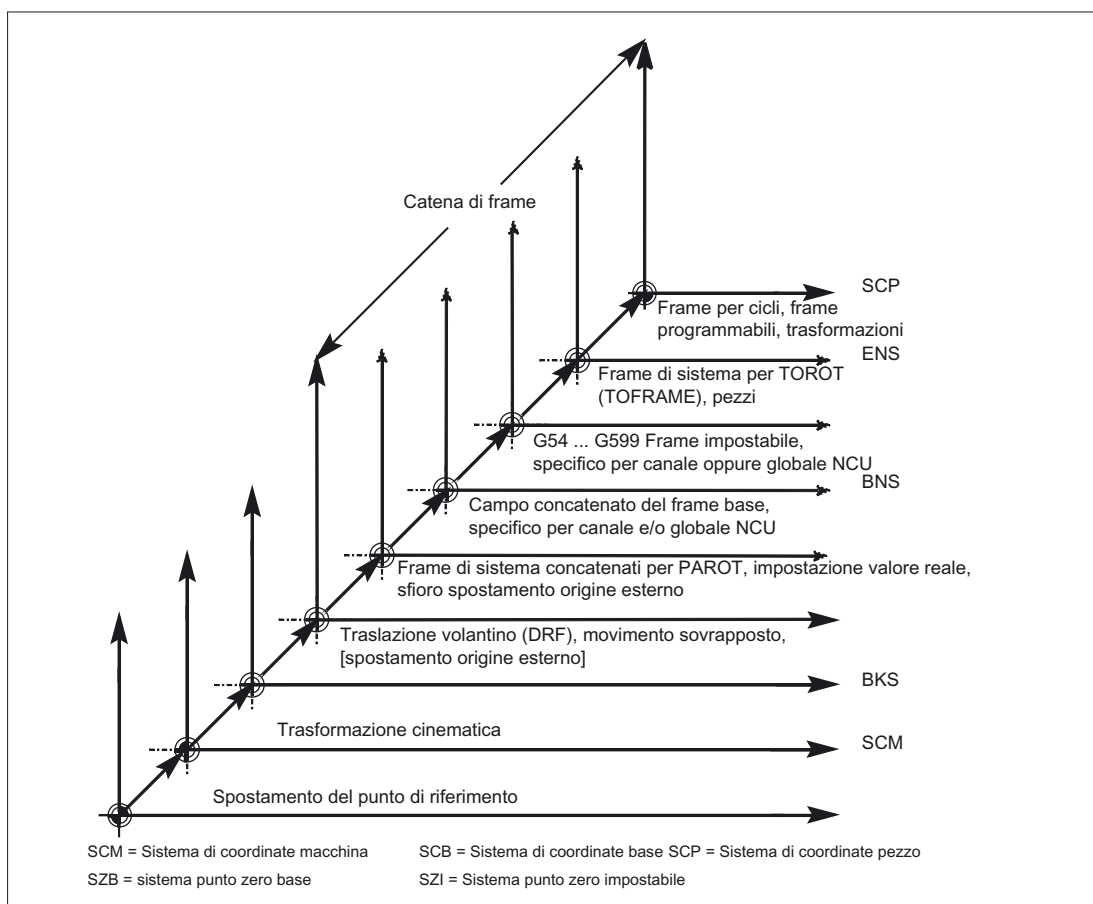
`$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :`

`$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME`



Concatenamento di frame

Il frame attuale è formato dal frame di base globale, dal frame impostabile, dal frame di sistema e dal frame programmabile in base al frame globale attuale descritto in precedenza.



Trasformazioni

7.1 Programmazione generale dei tipi di trasformazione

Funzione generale

Per adattare il controllo a diverse cinematiche delle macchine esiste la possibilità di programmare la selezione delle trasformazioni con parametri adatti. Con questi parametri è possibile dichiarare conformemente alla trasformazione selezionata sia l'orientamento dell'utensile nello spazio, sia i movimenti di orientamento degli assi rotanti.

Nelle trasformazioni a tre, quattro e cinque assi le indicazioni delle posizioni programmate si riferiscono sempre alla punta dell'utensile, trascinato ortogonalmente alla superficie di lavorazione che si trova nello spazio. Le coordinate cartesiane vengono convertite dal sistema di coordinate di base al sistema di coordinate della macchina e si riferiscono agli assi geometrici. Esse descrivono il punto di lavorazione. Gli assi rotanti virtuali descrivono gli orientamenti dell'utensile nello spazio e vengono programmati con TRAORI.

Nella trasformazione cinematica possono essere programmate posizioni nel sistema di coordinate cartesiano. Il controllo trasforma i movimenti programmati con TRANSMIT, TRACYL e TRAANG del sistema di coordinate cartesiano in movimenti degli assi reali della macchina.

Programmazione

Trasformazioni a tre, quattro e cinque assi TRAORI

La trasformazione di orientamento viene attivata con il comando TRAORI e i tre parametri possibili per numero di trasformatori, vettore di orientamento e offset asse rotante.

TRAORI (numero trasformatore, vettore di orientamento, offset asse rotante)

Trasformazioni cinematiche

Delle trasformazioni cinematiche fanno parte le trasformazioni dichiarate TRANSMIT (numero trasformatore)

TRACYL (diametro di lavorazione, numero trasformatore)

TRAANG (angolo dell'asse inclinato, numero trasformatore)

Disattivazione della trasformazione attiva

TRAFOOF consente di disattivare la trasformazione attualmente attiva.

Trasformazione dell'orientamento

Trasformazioni a tre, quattro e cinque assi TRAORI

Per la lavorazione ottimizzata delle superfici tridimensionali nel campo di lavoro della macchina, oltre ai tre assi lineari X, Y e Z, le macchine utensili necessitano anche di assi supplementari. Gli assi supplementari descrivono l'orientamento nello spazio e verranno successivamente denominati assi di orientamento. Essi sono disponibili come assi rotanti nei quattro tipi di macchina con cinematica diversa.

1. Testina orientabile a due assi, p. es. testina utensile cardanica con un asse rotante parallelo ad un asse lineare nella tavola utensile fissa.
2. Tavola rotante a due assi, es. testina orientabile fissa con tavola rotante intorno a due assi.
3. Testina orientabile monoasse e tavola rotante monoasse, p. es. una testina orientabile rotante con utensile ruotato su tavola rotante intorno ad un asse.
4. Testina orientabile a due assi e tavola rotante monoasse, p. es. una testina utensile rotante intorno a un asse e una testina orientabile rotante con due utensili rotanti su sé stessi.

Le **trasformazioni a 3 e 4 assi** sono forme speciali della trasformazione a 5 assi e vengono programmati in modo analogo alle trasformazioni a 5 assi.

Con le sue funzioni, la "**trasformazione a 3/4/5/6 assi**" gestisce gli assi rotanti ordinati ad angolo retto nonché le trasformazioni per la testa di fresatura cardanica e, come qualsiasi altra trasformazione dell'orientamento, anche questa può essere attivata per questi quattro tipi di macchina con TRAORI. Nella trasformazione generica a 5/6 assi l'orientamento dell'utensile ha un ulteriore terzo grado di libertà nel quale, per qualsiasi direzione dell'utensile nello spazio, lo stesso può essere ruotato sul proprio asse.

Bibliografia: /FB3/ Manuale delle funzioni speciali; Trasformazione a 3 ... 5 assi (F2)

Posizione base, indipendente dalla cinematica, dell'orientamento dell'utensile

ORIRESET

Se è attiva una trasformazione di orientamento con TRAORI, è possibile definire con ORIRESET le posizioni base di fino a 3 assi di orientamento con i parametri opzionali A, B e C. L'assegnazione dei parametri programmati agli assi rotanti avviene secondo la sequenza definita dalla trasformazione degli assi di orientamento. La programmazione di ORIRESET(A, B, C) fa sì che gli assi di orientamento intraprendano un movimento lineare e sincrono dalla loro posizione temporanea alla posizione base definita.

Trasformazioni cinematiche

TRANSMIT e TRACYL

Per le fresature su torni può essere programmata per la trasformazione dichiarata

1. una lavorazione sul lato frontale nel portautensile con TRANSMIT oppure
2. una lavorazione di una qualsiasi scanalatura su corpi cilindrici con TRACYL.

TRAANG

Se l'asse di incremento, p. es. per la rettifica tecnologica, deve poter essere posizionato anche inclinato, con TRAANG può essere programmato un angolo parametrizzabile per la trasformazione dichiarata.

Movimento cartesiano PTP

Alla trasformazione cinematica appartiene anche il "Movimento cartesiano PTP" con il quale si possono programmare fino a 8 diverse posizioni dello snodo STAT=. Le posizioni vengono programmate nel sistema di coordinate cartesiano; il movimento della macchina avviene in coordinate macchina.

Bibliografia:

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazione cinematica (M1)

Trasformazioni concatenate

È possibile attivare in successione rispettivamente due trasformazioni. In questo modo le quote di movimento per gli assi della prima trasformazione, diventano i dati di ingresso per la seconda trasformazione concatenata.

Come prima trasformazione sono possibili:

- Trasformazione dell'orientamento TRAORI
- Trasformazione polare TRANSMIT
- Trasformazione cilindrica TRACYL
- Trasformazione per asse inclinato TRAANG

La seconda trasformazione deve essere del tipo "Asse inclinato TRAANG"

7.1.1 Movimenti di orientamento nelle trasformazioni**Movimenti di traslazione e movimenti di orientamento**

I movimenti degli orientamenti programmabili dipendono principalmente dal tipo di macchina. Nella trasformazione a tre, quattro e cinque assi con TRAORI, gli assi rotanti o gli assi lineari orientabili descrivono i movimenti di orientamento dell'utensile.

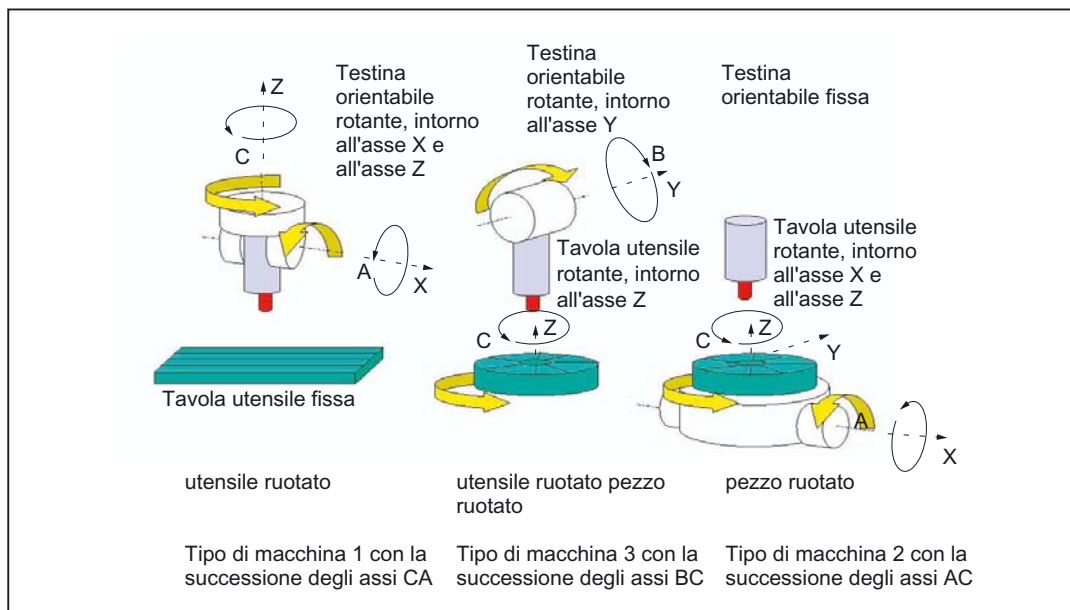
Le variazioni delle posizioni degli assi rotanti coinvolti nella trasformazioni dell'orientamento comportano movimenti di compensazione degli altri assi della macchina. La posizione della punta dell'utensile rimane invariata.

A seconda dell'impiego, i movimenti di orientamento dell'utensile possono essere programmati con un identificatore di asse rotante A..., B..., C... degli assi virtuali indicando angoli di Eulero e RPY oppure vettori di direzione e vettori normalizzati alla superficie, vettori normalizzati per l'asse di rotazione di un cono oppure per l'orientamento intermedio su una superficie conica.

Nella trasformazione cinematica con TRANSMIT, TRACYL e TRAANG, il controllo trasforma i movimenti programmati del sistema di coordinate cartesiano in movimenti degli assi reali della macchina.

Cinematica di macchina durante la trasformazione a tre, quattro e cinque assi TRAORI

Può essere ruotato sia l'utensile, sia la tavola utensile con uno o due assi rotanti. È possibile anche una combinazione di rispettivamente una testina orientabile monoasse e una tavola rotante.



Tipo di macchina

Programmazione dell'orientamento

Trasformazione a tre assi, tipi di macchina 1 e 2

Programmazione dell'orientamento dell'utensile solo nel piano ortogonale all'asse rotante. Esistono **due** assi traslatori (assi lineari) e **un** asse rotante.

Trasformazione a quattro assi, tipi di macchina 1 e 2

Programmazione dell'orientamento dell'utensile solo nel piano ortogonale all'asse rotante. Esistono **tre** assi traslatori (assi lineari) e **un** asse rotante.

Trasformazione a cinque assi, tipi di macchina 3
Testina orientabile monoasse e tavola monoasse

Programmazione della trasformazione dell'orientamento Cinematica con **tre** assi lineari e **due** assi rotanti ortogonali. Gli assi rotanti sono paralleli a due dei tre assi lineari.. Il primo asse rotante viene mosso da due assi lineari cartesiani. Esso ruota il terzo asse lineare con l'utensile. Il secondo asse rotante fa ruotare il pezzo.

Trasformazioni generiche a 5/6 assi

Tipo di macchina	Programmazione della trasformazione dell'orientamento
Trasformazione generica a cinque assi, tipi di macchina 4	Programmazione della trasformazione dell'orientamento Cinematica con tre assi lineari e tre assi rotanti ortogonali.
Testina orientabile a due assi con utensile rotante su se stesso e tavola monoasse	Gli assi rotanti sono paralleli a due dei tre assi lineari.. Il primo asse rotante viene mosso da due assi lineari cartesiani. Esso ruota il terzo asse lineare con l'utensile. Il secondo asse rotante fa ruotare il pezzo. L'orientamento base dell'utensile può essere programmato tramite una sua rotazione aggiuntiva su se stesso con l'angolo di rotazione THETA.

Durante il richiamo della "trasformazione a tre, quattro e cinque/sei assi generica" è inoltre possibile trasmettere l'orientamento di base dell'utensile. Non valgono più le limitazioni relative alle direzioni degli assi rotanti. Se gli assi rotanti non si trovano esattamente in posizione ortogonale l'uno rispetto all'altro oppure se gli assi rotanti esistenti non sono esattamente paralleli agli assi lineari, la "trasformazione a cinque/sei assi generica" può fornire risultati migliori dell'orientamento dell'utensile.

Trasformazioni cinematiche TRANSMIT, TRACYL e TRAANG

Per le fresature su torni o di un asse posizionabile inclinato per la rettifica, in base alla trasformazioni standard valgono le seguenti assegnazioni dell'asse:

TRANSMIT	Attivazione della trasformazione polare
lavorazione frontale in direzione del bloccaggio del pezzo	un asse rotante, un asse di incremento ortogonale all'asse di rotazione, un asse longitudinale parallelo all'asse di rotazione
TRACYL	Attivazione della trasformazione sulla superficie cilindrica
Lavorazione di un numero qualsiasi di cave sul corpo cilindrico	un asse rotante, un asse di incremento ortogonale all'asse di rotazione, un asse longitudinale parallelo all'asse di rotazione
TRAANG	Attivazione della trasformazione per asse inclinato
lavorazione con asse di incremento inclinato	un asse rotante, un asse di incremento ortogonale con angolo parametrizzabile, un asse longitudinale parallelo all'asse di rotazione

Movimento cartesiano PTP

Il movimento della macchina avviene nelle coordinate della macchina e viene programmato con:

TRAORI	Attivazione della trasformazione
Movimento punto a punto PTP	Accostamento alla posizione nel sistema di coordinate cartesiano (MKS)
CP	Movimento vettoriale degli assi cartesiani in (BKS)
STAT	La posizione dello snodo dipende dalla trasformazione
TU	Angolo che devono compiere gli assi per effettuare il percorso più breve

Movimento PTP durante la trasformazione generica a 5/6 assi

Il movimento della macchina avviene in coordinate macchina e l'orientamento dell'utensile può essere programmato sia con posizioni di asse rotante, sia con vettori indipendenti dalla cinematica (angolo di Eulero o RPY) o con i vettori direzionali.

Sono possibili l'interpolazione dell'asse rotante, l'interpolazione vettoriale con interpolazione circolare ad ampio raggio o l'interpolazione del vettore di orientamento su una superficie conica.

Esempio di trasformazione a tre ... cinque assi per una testa di fresatura cardanica

La macchina utensile ha almeno 5 assi dei quali

- Tre assi traslatori per movimenti rettilinei che muovono il punto di lavorazione in una posizione qualsiasi del campo di lavoro.
- Due assi rotanti orientabili disposti con un angolo progettabile (normalmente 45 gradi) che consentono all'utensile orientamenti nello spazio limitati ad un posizionamento di 45 gradi su una semisfera.

7.1.2 Panoramica della trasformazione dell'orientamento TRAORI

Possibili tipi di programmazione con TRAORI

Tipo di macchina	Programmazione per la trasformazione attiva TRAORI
ipi di macchina 1, 2 o 3, testina orientabile a due assi o tavola a due assi o una combinazione di testina orientabile monoasse e tavola.	La successione degli assi di orientamento e la direzione di orientamento dell'utensile sono progettabili in funzione della macchina tramite i dati della macchina e in funzione della cinematica della macchina, oppure sono progettabili in funzione del pezzo con orientamento programmabile indipendentemente dalla cinematica della macchina. Le direzioni di rotazione degli assi di orientamento vengono programmate nel sistema di riferimento con: - ORIMKS sistema di riferimento = sistema di coordinate macchina - ORIWKS sistema di riferimento = sistema di coordinate macchina L'impostazione di base è ORIWKS. Programmazione degli assi di orientamento con: A, B, C delle posizioni degli assi della macchina direttamente A2, B2, C2 programmazione dell'angolo degli assi virtuali con - ORIEULER tramite angolo di Eulero (standard) - ORIRPY tramite angolo RPY - ORIVIRT1 tramite assi di orientamento virtuali della prima definizione - ORIVIRT2 tramite assi di orientamento virtuali della seconda definizione con distinzione del tipo di interpolazione: interpolazione lineare - ORIAXES degli assi di orientamento o degli assi della macchina interpolazione del cerchio estesa (interpolazione del vettore di orientamento) - ORIVECT degli assi di orientamento Programmazione degli assi di orientamento tramite immissione A3, B3, C3 dei componenti dei vettori (direzionali/normali alla superficie) Programmazione dell'orientamento utensile risultante A4, B4, C4 del vettore normale alla superficie a inizio blocco A5, B5, C5 del vettore normale alla superficie a fine blocco LEAD angolo d'anticipo per l'orientamento dell'utensile TILT angolo laterale per l'orientamento dell'utensile interpolazione del vettore di orientamento su una superficie conica Modifiche dell'orientamento su una superficie conica che si trova in uno spazio qualsiasi tramite interpolazione: - ORIPANE nel piano (interpolazione del cerchio estesa) - ORICONCW su una superficie conica in senso orario - ORICONCCW su una superficie conica in senso antiorario

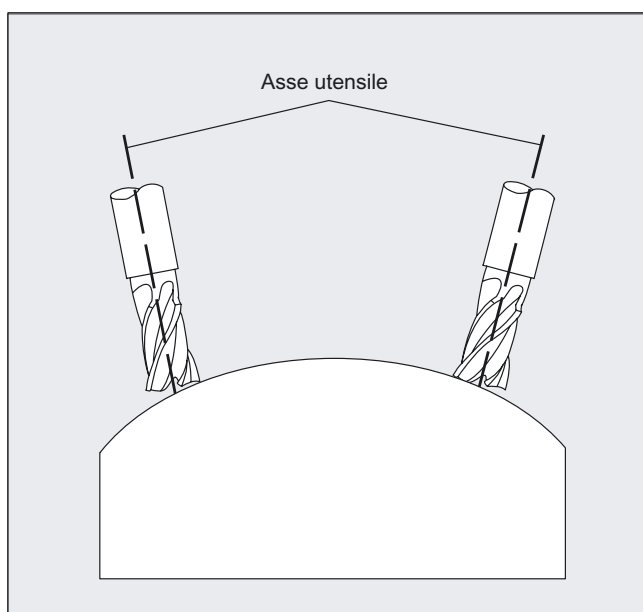
Tipo di macchina	Programmazione per la trasformazione attiva TRAORI
	A6, B6, C6 vettore direzionale (asse di rotazione del cono) -OICONIO interpolazione su una superficie conica con: A7, B7, C7 vettori intermedi (orientamento iniziale e finale) oppure - ORICONTO su una superficie conica del raccordo tangenziale Modifiche dell'orientamento riferite ad un percorso con - ORICURVE impostazione del movimento di due punti di contatto tramite PO[XH]=(xe, x2, x3, x4, x5) polinomi di orientamento fino al 5° grado PO[YH]=(ye, y2, y3, y4, y5) polinomi di orientamento fino al 5° grado PO[ZH]=(ze, z2, z3, z4, z5) polinomi di orientamento fino al 5° grado - ORIPATHS spianamento del percorso di orientamento con A8, B8, C8 La fase di modifica dell'orientamento dell'utensile comprende: direzione e lunghezza del percorso dell'utensile durante il movimento di svincolo
Tipi di macchina 1 e 3	Programmazione delle rotazioni dell'orientamento utensile con LEAD angolo d'anticipo relativo al vettore normale alla superficie PO[PHI] Programmazione di un polinomio fino al 5° grado TILT angolo laterale rotazione intorno alla tangente del percorso (direzione Z)
Altri tipi di macchina con una rotazione aggiuntiva dell'utensile su se stesso richiedono un terzo asse rotante	PO[PSI] Programmazione di un polinomio fino al 5° grado THETA angolo di rotazione (rotazione intorno alla direzione dell'utensile in Z) THETA= valore che viene raggiunto a fine blocco THETA=AC(...) commutazione all'impostazione in quote assolute nel blocco
Trasformazione dell'orientamento, come ad es. trasformazione generica a 6 assi. Rotazioni del vettore di orientamento.	THETA=IC(...) commutazione nell'impostazione in quote incrementali nel blocco THETA=Θ_e Interpolazione dell'angolo programmato G90/G91 PO[THT]=(..) programmazione di un polinomio fino al 5° grado Programmazione del vettore di rotazione - ORIROTA rotazione assoluta - ORIROTR vettore di rotazione relativo - ORIROTT vettore di rotazione tangenziale
Orientamento rispetto al percorso per modifiche di orientamento relative alla traiettoria o rotazione del vettore di rotazione tangenzialmente alla traiettoria	Modifiche dell'orientamento relative al percorso con - ORIPATH orientamento dell'utensile riferito al percorso - ORIPATHS programmazione aggiuntiva del vettore di rotazione in caso di una piega nel percorso di orientamento - ORIROTC vettore di rotazione tangenziale, rotazione verso la tangente vettoriale

7.2 Trasformazione a tre, quattro e cinque assi (TRAORI)

7.2.1 Relazioni generali della testa portautensili cardanica

Funzione

Per avere condizioni di taglio ottimali nella lavorazione di superfici curve nello spazio, l'angolo di orientamento dell'utensile deve essere variabile.

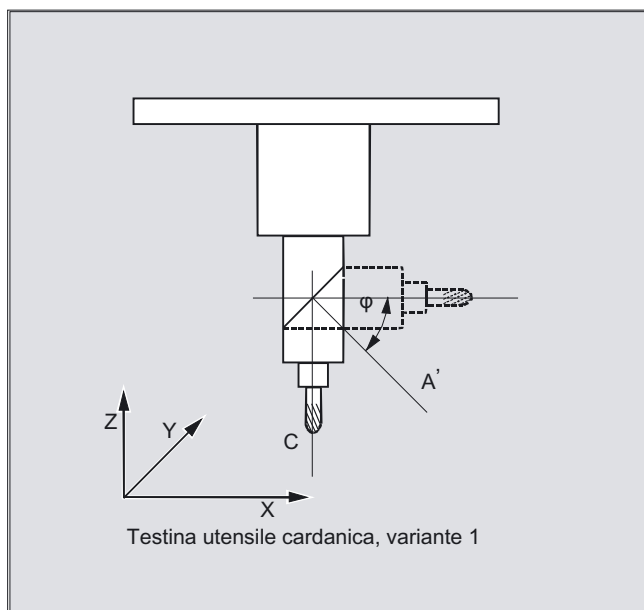


Nei dati degli assi è definita la struttura di macchina con la quale è possibile effettuare questo tipo di lavorazione.

Trasformazione a 5 assi

Testa utensile cardanica

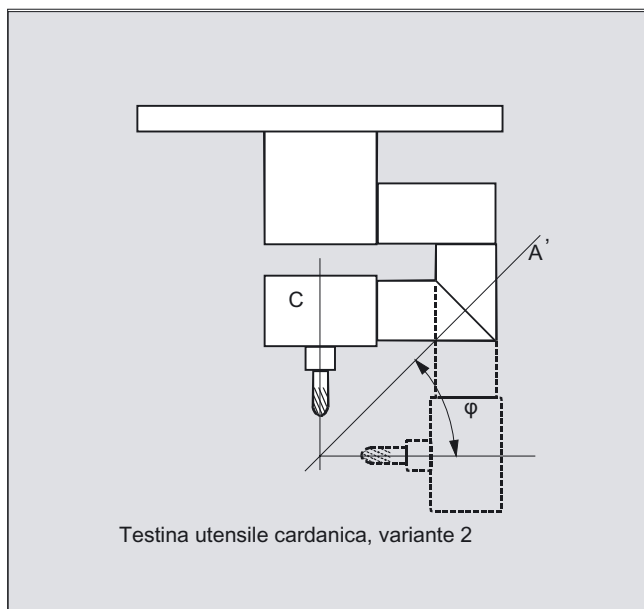
In questo caso, sono i tre assi lineari (X, Y, Z) e i due assi di orientamento (C, A) a determinare l'angolo di orientamento ed il punto di lavoro dell'utensile. Uno dei due assi di orientamento viene utilizzato come asse inclinato (qui nell'esempio A') - molto spesso con un angolo di 45°.



Negli esempi qui riportati la disposizione si riferisce ad una cinematica di macchina CA con testina utensile cardanica!.

Costruttore della macchina

La successione degli assi di orientamento e la direzione di orientamento dell'utensile possono essere stabilite in funzione della cinematica della macchina tramite dati macchina.



In questo esempio A' si trova sotto l'angolo ϕ verso l'asse X

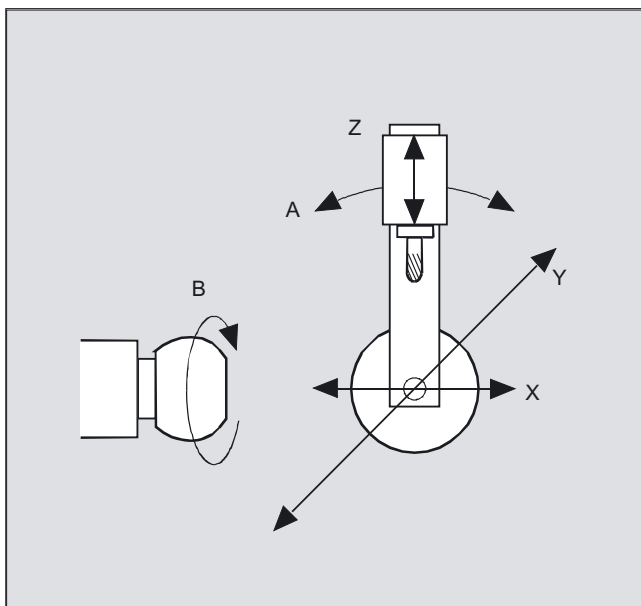
In generale valgono le seguenti relazioni possibili:

A' si trova sotto l'angolo φ verso	l'asse X
B' si trova sotto l'angolo φ verso	l'asse Y
C' si trova sotto l'angolo φ verso	l'asse Z

L'angolo l'asse φ può essere progettato tramite dati macchina nel campo da $0^\circ \dots +89^\circ$.

Con asse lineare orientabile

Si tratta di un allineamento tra utensili mobili e pezzi mobili. La cinematica si compone di tre assi lineari (X, Y, Z) e di due assi rotanti disposti ad angolo retto. Il primo asse rotante, ad esempio, viene mosso su di una slitta a croce da due assi lineari; l'utensile è disposto parallelamente e si trova sul terzo asse. Il secondo asse rotante fa ruotare il pezzo. Il terzo asse lineare (asse orientabile) si trova nel piano della slitta a croce.



La successione degli assi rotanti e la direzione di orientamento dell'utensile possono essere stabilite in funzione della cinematica della macchina tramite dati macchina.

Sono valide le seguenti correlazioni:

Assi:	Sequenza degli assi:
1. Asse rotante	A A B B C C
2. Asse rotante	B C A C A B
Asse lineare orientabile	Z Y Z X Y X

Per ulteriori descrizioni sulle sequenze degli assi configurabili per la direzione di orientamento dell'utensile vedere

Bibliografia: /FB3/ Manuale delle funzioni speciali; Trasformazioni a 3 ... 5 assi (F2), capitolo Testina di fresatura cardanica, "Parametrizzazione".

7.2.2 Trasformazione a tre, quattro e cinque assi (TRAORI)

Funzione

L'utente può progettare due o tre assi lineari e un asse rotante. Nelle trasformazioni si parte dal presupposto che l'asse rotante sia in posizione ortogonale rispetto al piano di orientamento.

L'orientamento dell'utensile è possibile solo sul piano disposto perpendicolarmente rispetto all'asse rotante. La trasformazione supporta i tipi di macchina con utensile e pezzo mobili.

La progettazione e la programmazione delle trasformazioni a 3 e 4 assi sono uguali a quelle delle trasformazioni a 5 assi.

Bibliografia:

/FB3/ Manuale delle funzioni speciali; Trasformazioni a 3 ... 5 assi (F2)

Programmazione

TRAORI (n)
oppure
TRAORI (n, X, Y, Z, A, B)
oppure
TRAFOOF

Parametri

TRAORI	Attivazione della prima trasformazione dell'orientamento
TRAORI (n)	Attivazione della trasformazione di orientamento definita con n
n	Numero della trasformazione (n = 1 oppure 2), TRAORI (1) corrisponde alla trasformazione di orientamento
X, Y, Z	Componente del vettore di orientamento al quale punta l'utensile.
A, B	Offset programmabile per gli assi rotanti
TRAFOOF	Disattivazione della trasformazione

Orientamento dell'utensile

In funzione del senso di orientamento dell'utensile selezionato, nel programma NC i piani di lavoro attivi (G17, G18, G19) devono essere impostati in modo tale che la correzione della lunghezza utensile sia valida nel senso di orientamento dell'utensile.

Nota

Dopo l'inserzione della trasformazione, i dati di posizione (X, Y, Z) si riferiscono sempre alla punta dell'utensile. Variando le posizioni degli assi rotanti interessati alla trasformazione, nei restanti assi di macchina si hanno movimenti di compensazione tali che la posizione della punta dell'utensile resta invariata.

La trasformazione di orientamento è sempre diretta dalla punta dell'utensile verso il portautensile.

Esempio di trasformazioni generiche

l'orientamento di base dell'utensile punta in:

TRAORI (1, 0, 0, 1) direzione Z

TRAORI (1, 0, 1, 0) direzione Y

TRAORI (1, 0, 1, 1) direzione Y/Z (corrisponde alla posizione -45°)

Offset per gli assi di orientamento

Con l'attivazione della trasformazione di orientamento è possibile programmare un ulteriore offset per gli assi di orientamento.

Si possono tralasciare i parametri se viene rispettata la sequenza corretta della programmazione.

Esempio

TRAORI (, , , , A, B) se deve essere impostato un unico offset.

In alternativa alla programmazione diretta, questo offset supplementare per gli assi di orientamento può anche essere rilevato automaticamente dallo spostamento origine momentaneamente attivo. Il rilevamento viene progettato tramite dati macchina.

7.2.3 Varianti della programmazione dell'orientamento e della posizione base (OTIRESET)

Programmazione dell'orientamento dell'utensile con TRAORI

In combinazione con una trasformazione dell'orientamento programmabile TRAORI è possibile programmare tramite gli indicatori di asse rotanti A..., B..., C..., oltre agli assi lineari X, Y e Z, anche posizioni dell'asse o assi virtuali con angoli o componenti vettoriali. Per assi di orientamento e assi macchina sono possibili diversi tipi di interpolazione.

Indipendentemente da quali polinomi di orientamento PO[angolo] e polinomi di asse PO[asse] sono attualmente attivi, possono essere programmati diversi tipi di polinomi come ad es. G1, G2, G3, CIP o POLY.

La modifica dell'orientamento dell'utensile può essere programmata anche tramite vettori di orientamento. In questo caso l'orientamento finale di ogni blocco può essere eseguito tramite programmazione diretta del vettore o tramite programmazione delle posizioni dell'asse rotante.

Nota

Varianti della programmazione dell'orientamento per trasformazioni a tre ... cinque assi

Nella trasformazione a 3 .. 5 assi, le varianti

1. A, B, C indicazione diretta delle posizioni dell'asse della macchina
2. A2, B2, C2 programmazione dell'angolo degli assi virtuali tramite angolo di Eulero o angolo RPY
3. A3 ,B3, C3 indicazione dei componenti vettoriali
4. LEAD, TILT indicazione dell'angolo di anticipo e dell'angolo laterale riferito al percorso e alla superficie
5. A4, B4, C4 e A5, B5, C5 vettore normale alla superficie all'inizio e alla fine del blocco
6. A6, B6, C6 e A7, B7, C7 interpolazione e vettore di orientamento su una superficie conica
7. A8, B8, C8 modifica dell'orientamento dell'utensile, direzione e lunghezza di percorso del movimento di svincolo

si escludono reciprocamente.

I valori con programmazione mista vengono impediti da messaggi di allarme.

Posizione base dell'orientamento dell'utensile ORIRESET

La programmazione di ORIRESET(A, B, C) fa sì che gli assi di orientamento intraprendano un movimento lineare e sincrono dalla loro posizione temporanea alla posizione base definita.

Se per un asse non viene programmata alcuna posizione base, viene utilizzata la posizione definita a partire dal relativo dato macchina \$MC_TRAFO5_ROT_AX_OFFSET_1/2. I frame eventualmente attivi degli assi rotanti non vengono considerati.

Nota

Solo se è attiva una trasformazione dell'orientamento con TRAORI(...) è possibile programmare una posizione base dell'orientamento dell'utensile indipendentemente dalla cinematica con ORIRESET(...) senza che scatti l'allarme 14101.

Esempi

```

1. Esempio di cinematica di macchina CA (nomi dell'asse canale C, A)
ORIRESET(90, 45)           ;C a 90 gradi, A a 45 gradi
ORIRESET(, 30)             ;C su $MC_TRAFO5_ROT_AX_OFFSET_1/2[0], A a 30 gradi
ORIRESET( )                ;C su $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                           ;A su $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

2. Esempio di cinematica di macchina CAC (nomi dell'asse canale C, A, B)
ORIRESET(90, 45, 90)       ;C a 90 gradi, A a 45 gradi, B a 90 gradi
ORIRESET( )                ;C su $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                           ;A su $MC_TRAFO5_ROT_AX_OFFSET_1/2[1],
                           ;B su $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]
```

Programmazione delle rotazioni LEAD, TILT e THETA

Nella trasformazione a 3 ... 5 assi le rotazioni dell'orientamento dell'utensile vengono programmate con l'angolo d'anticipo LEAD e l'angolo laterale TILT.

In una trasformazione con terzo asse rotante sono consentite programmazioni supplementari di C2 (rotazioni del vettore di orientamento) sia per l'orientamento con componenti vettoriali, sia con indicazione dell'angolo LEAD, TILT.

Con un terzo asse rotante supplementare, la rotazione dell'utensile su se stesso può essere programmata con l'angolo di rotazione THETA.

7.2.4 Programmazione dell'orientamento utensile (A..., B..., C..., LEAD, TILT)

Funzione

Per la programmazione dell'orientamento utensile esistono le seguenti possibilità:

1. Programmazione diretta del movimento degli assi rotanti. La variazione dell'orientamento avviene sempre nel sistema di coordinate base o di macchina. Gli assi di orientamento vengono mossi come assi sincroni.
2. Programmazione con angoli di Eulero o di RPY secondo la definizione dell'angolo tramite A2, B2, C2.
3. Programmazione del vettore direzionale tramite A3, B3, C3. Il vettore direzionale indica dalla punta dell'utensile in direzione del portautensile.
4. Programmazione del vettore normale alla superficie all'inizio del blocco con A4, B4, C4 e alla fine del blocco con A5, B5, C5 (fresatura frontale).
5. Programmazione tramite angolo d'anticipo LEAD e angolo laterale TILT
6. Per la programmazione dell'asse di rotazione del cono come vettore normalizzato tramite A6, B6, C6 o dell'orientamento intermedio sulla superficie conica tramite A7, B7, C7, vedere il capitolo "Programmazione dell'orientamento lungo una superficie conica (ORIPANE, ORICONxx)".
7. Per la programmazione della modifica di orientamento, direzione e lunghezza di percorso dell'utensile durante il movimento di svincolo tramite A8, B8, C8, vedere il capitolo "Livellamento del percorso di orientamento (ORIPATHS A8=, B8=, C8=)".

Nota

In tutti i casi la programmazione dell'orientamento è consentita solo se è stata attivata una trasformazione di orientamento.

Vantaggio: questi programmi sono utilizzabili con qualsiasi tipo di cinematica di macchina.

Definizione dell'orientamento dell'utensile tramite codice G

Nota

Costruttore della macchina

Tramite dati macchina è possibile commutare tra angoli di Eulero o RPY. Con opportune impostazioni dei dati macchina, una commutazione è possibile sia in funzione del codice G attivo del gruppo 50, sia indipendentemente da esso. Sono disponibili le seguenti possibilità d'impostazione:

1. Se entrambi i dati macchina per la definizione degli assi di orientamento e la definizione dell'angolo di orientamento tramite codice G sono impostati a zero:
Gli angoli programmati con A2, B2, C2 vengono interpretati, **a seconda del dato macchina** definizione angolo della programmazione dell'orientamento, come angoli di Eulero o RPY.
2. Se il dato macchina per la definizione degli assi di orientamento tramite codice G è impostato a uno, la commutazione avviene **in funzione** del codice G attivo del gruppo 50:
Gli angoli programmati con A2, B2, C2 vengono interpretati secondo uno dei codici G attivi ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAPOS e ORIPY2. I valori programmati con gli assi di orientamento vengono interpretati secondo il codice G attivo del gruppo 50 anche come angoli di orientamento.
3. Se il dato macchina per la definizione degli angoli di orientamento tramite codice G è impostato a uno e il dato macchina per la definizione degli assi di orientamento tramite codice G è impostato a zero, la commutazione avviene **indipendentemente** dal codice G attivo del gruppo 50:
Gli angoli programmati con A2, B2, C2 vengono interpretati secondo uno dei codici G attivi ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAPOS e ORIPY2. I valori programmati con gli assi di orientamento vengono interpretati, indipendentemente dal codice G attivo del gruppo 50, sempre come posizioni di asse rotante.

Programmazione

G1 X Y Z A B C

Programmazione del movimento degli assi di rotazione

G1 X Y Z A2= B2= C2=

Programmazione con angoli di Eulero

G1 X Y Z A3== B3== C3==

Programmazione del vettore direzionale

G1 X Y Z A4== B4== C4==

Programmazione del vettore normale alla superficie a inizio blocco

G1 X Y Z A5== B5== C5==

Programmazione del vettore normale alla superficie a fine blocco

LEAD=

Angolo di anticipo per la programmazione dell'orientamento utensile

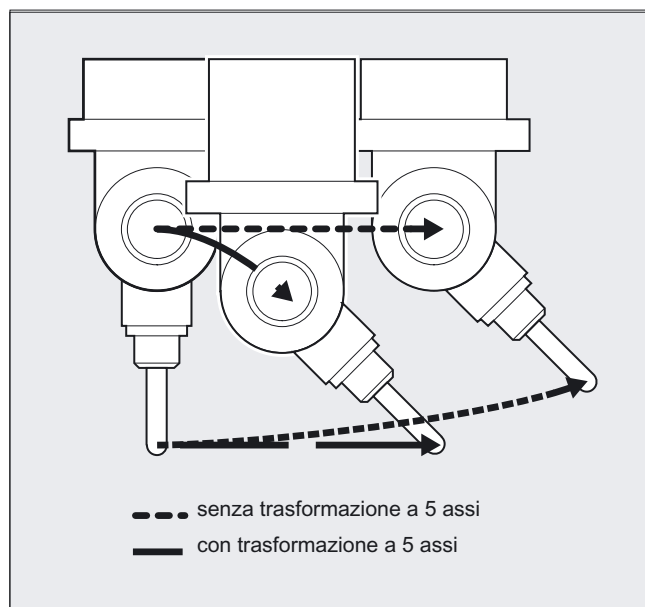
TILT=

Angolo laterale per la programmazione dell'orientamento utensile

Parametri

G	Impostazione del tipo di movimento degli assi rotanti
X Y Z	Impostazione degli assi lineari
A B C	Impostazione delle posizioni macchina degli assi rotanti
A2 B2 C2	Programmazione angolare (angolo di Eulero o RPY) degli assi virtuali o di orientamento
A3 B3 C3	Impostazione dei componenti vettoriali, vettore direzionale
A4 B4 C4	Impostazione, es. per fresatura frontale, all'inizio del blocco delle componenti del vettore normale alla superficie
A5 B5 C5	Impostazione, es. per fresatura frontale, alla fine del blocco delle componenti del vettore normale alla superficie
LEAD	Angolo relativo al vettore normale alla superficie nel piano determinato dalla tangente al percorso e dal vettore normale alla superficie
TILT	Angolo nel piano, perpendicolare alla tangente al percorso relativamente al vettore normale alla superficie

Esempio di confronto senza e con trasformazione a 5 assi



Descrizione

Solitamente i programmi in 5 assi vengono generati da sistemi CAD/CAM e non impostati direttamente nel controllo numerico. Pertanto i seguenti chiarimenti si rivolgono essenzialmente ai programmatori di postprocessori.

Il tipo di programmazione dell'orientamento viene definito nel codice G, gruppo 50:

ORIEULER tramite angolo di Eulero

ORIRPY tramite angolo RPY (successione di rotazione ZYX)

ORIVIRT1 tramite assi di orientamento virtuali (definizione 1)

ORIVIRT2 tramite assi di orientamento virtuali (definizione 2)

ORIAXPOS tramite assi di orientamento virtuali con posizioni di asse rotante

ORIPY2 tramite angolo RPY (successione di rotazione XYZ)

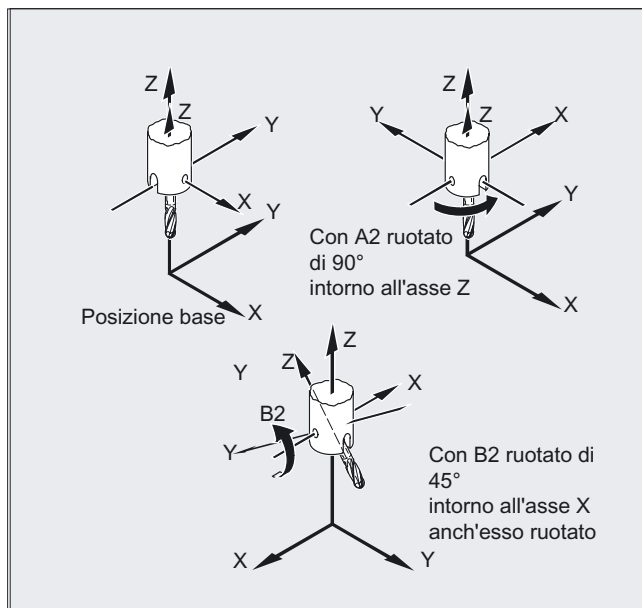
Costruttore della macchina

Tramite i dati macchina il costruttore della macchina può definire diverse varianti. Fare attenzione alle indicazioni del costruttore della macchina

Programmazione con angoli di Eulero ORIEULER

I valori programmati per l'orientamento con A_2 , B_2 , C_2 vengono interpretati come angoli di Eulero (in gradi).

Il vettore di orientamento si ricava ruotando prima il vettore con A_2 intorno all'asse Z, poi con B_2 intorno al nuovo asse X ed infine con C_2 intorno al nuovo asse Z.



In questo caso il valore di C_2 (rotazione intorno al nuovo asse Z) è irrilevante e non va pertanto programmato.

Programmazione con angoli RPY ORIRPY

I valori programmati per l'orientamento con A_2 , B_2 , C_2 vengono interpretati come angoli RPY (in gradi).

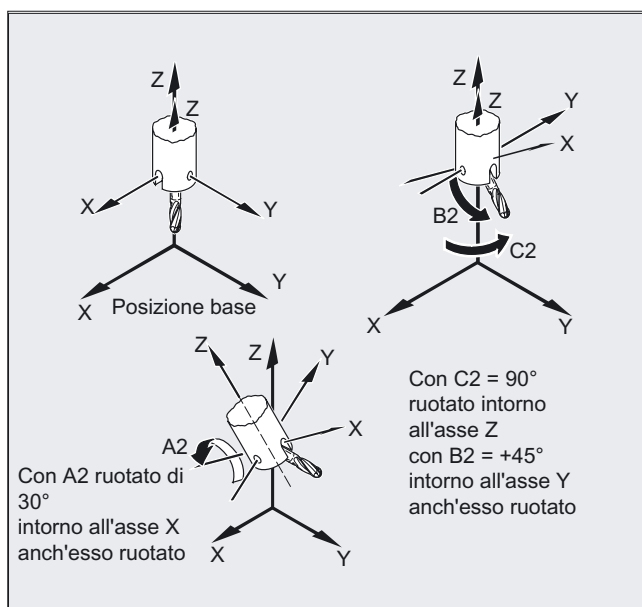
Nota

Al contrario della programmazione con angoli di Eulero, tutti e tre i valori hanno influenza sul vettore di orientamento.

Costruttore della macchina

Nel caso di definizione dell'angolo con angolo di orientamento mediante angolo RPY vale per gli assi di orientamento $\$MC\_ORI\_DEF\_WITH\_G\_CODE = 0$

Il vettore di orientamento si ricava ruotando prima di C_2 il vettore intorno all'asse Z, poi di B_2 intorno al nuovo asse Y ed infine di A_2 intorno al nuovo asse X.



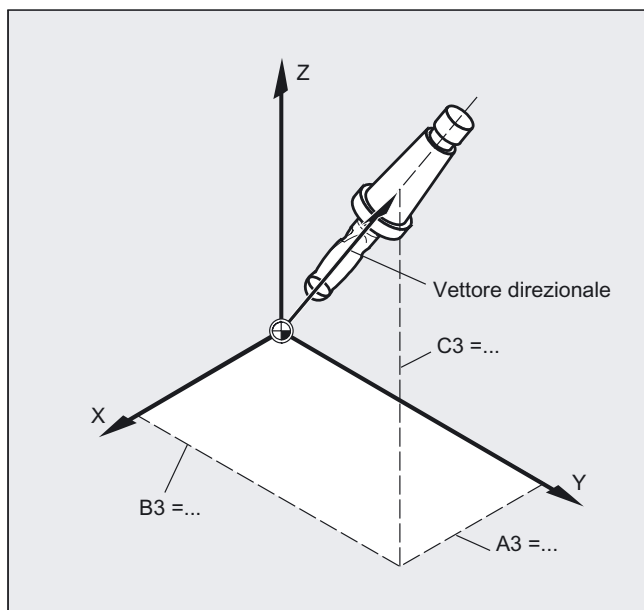
Se il dato macchina per la definizione degli assi di orientamento tramite codice G è $\$MC\_ORI\_DEF\_WITH\_G\_CODE = 1$, vale ciò che segue:

Il vettore di orientamento si ricava ruotando di A_2 un vettore in direzione Z intorno all'asse Z, poi di B_2 intorno al nuovo asse Y ed infine di C_2 intorno al nuovo asse X.

Programmazione del vettore direzionale

Le componenti del vettore direzionale vengono programmate con A_3 , B_3 , C_3 . Il vettore va dalla punta alla base di fissaggio dell'utensile; la lunghezza del vettore è irrilevante.

Le componenti del vettore non programmate vengono interpretate come zero.



Programmazione dell'orientamento utensile con LEAD= e TILT=

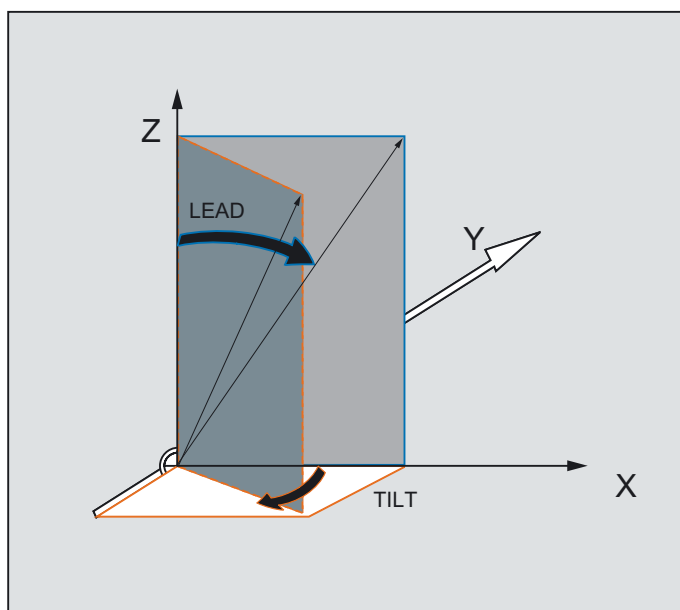
L'orientamento risultante dell'utensile viene ricavato da:

- Tangente al percorso
- Vettore normale alla superficie
all'inizio del blocco con A4, B4, C4 e alla fine del blocco con A5, B6, C5
- Angolo d'anticipo LEAD
nel piano determinato dalla tangente al percorso e dal vettore normale alla superficie
- Angolo laterale TILT alla fine del blocco
in posizione ortogonale alla tangente vettoriale e relativo al vettore normale alla superficie

Comportamento in presenza di angoli interni (per correzione utensile 3D)

Se si accorcia il blocco su un angolo interno, l'orientamento utensile risultante viene ugualmente raggiunto a fine blocco.

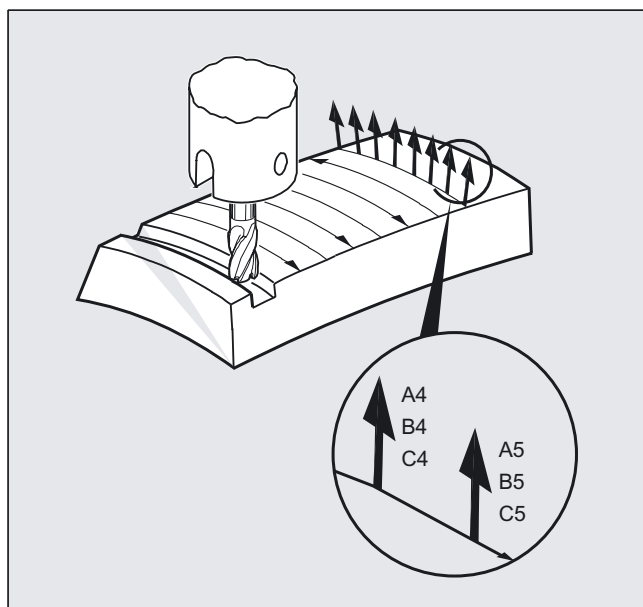
Definizione dell'orientamento utensile con LEAD= e TILT=



7.2.5 Fresatura frontale (fresatura 3D A4, B4, C4, A5, B5, C5)

Funzione

La fresatura frontale viene impiegata per la lavorazione di superfici curve.



Per questo tipo di fresatura 3D è necessaria la descrizione riga per riga della traiettoria 3D sulla superficie del pezzo.

Il calcolo viene effettuato normalmente con un CAM tenendo conto della forma e delle dimensioni dell'utensile. I blocchi NC calcolati vengono poi letti nel controllo numerico attraverso un postprocessore.

Programmazione della curvatura della traiettoria

Descrizione delle superfici

La descrizione della curvatura si effettua mediante vettori normali alla superficie con i seguenti componenti:

A4, B4, C4 vettore iniziale all'inizio del blocco

A5, B5, C5 vettore finale alla fine del blocco

Se in un blocco è presente solo il vettore iniziale, il vettore normale alla superficie resta costante per tutto il blocco. Se in un blocco è presente solo il vettore finale, viene effettuata un'interpolazione circolare grossolana verso il valore finale programmato facendo riferimento al vettore finale del blocco precedente.

Se sono programmati entrambi i vettori, viene effettuata un'interpolazione circolare tra ambedue le direzioni. In questo modo possono essere realizzati percorsi costanti e regolari.

Nella posizione di base i vettori normali alla superficie sono posizionati in direzione Z indipendentemente dal piano attivo G17 ... G19.

La lunghezza dei vettori è irrilevante.

I componenti di vettori non programmati vengono impostati a zero.

Se è attivo ORIWKS, vedere il capitolo "Riferimento degli assi di orientamento (ORIWKS, ORIMKS)" i vettori normali alla superficie si riferiscono al frame attivo e ruotano nel caso di rotazione del frame stesso.

Costruttore della macchina

Il vettore normale alla superficie deve essere perpendicolare alla tangente del profilo nell'ambito di un campo di valori impostabile tramite un dato macchina. In caso contrario viene emesso un allarme.

7.2.6 Riferimento degli assi di orientamento (ORIWKS, ORIMKS)

Funzione

Nella programmazione dell'orientamento nel sistema di coordinate pezzo tramite

- angolo di Eulero, angolo RPY oppure
- vettore di orientamento

è possibile stabilire il percorso del movimento di rotazione tramite ORIMKS / ORIWKS.

Costruttore della macchina

Con il dato macchina \$MC_ORI_IPO_WITH_G_CODE viene definito il tipo di interpolazione attivo:

ORIMKS/ORIWKS

oppure

ORIMACHAX/ORIVIRTAX.

Programmazione

N. . ORIMKS=

oppure

N. . ORIWKS=

Parametri

ORIMKS	Rotazione nel sistema di coordinate macchina
ORIWKS	Rotazione nel sistema di coordinate pezzo

Nota

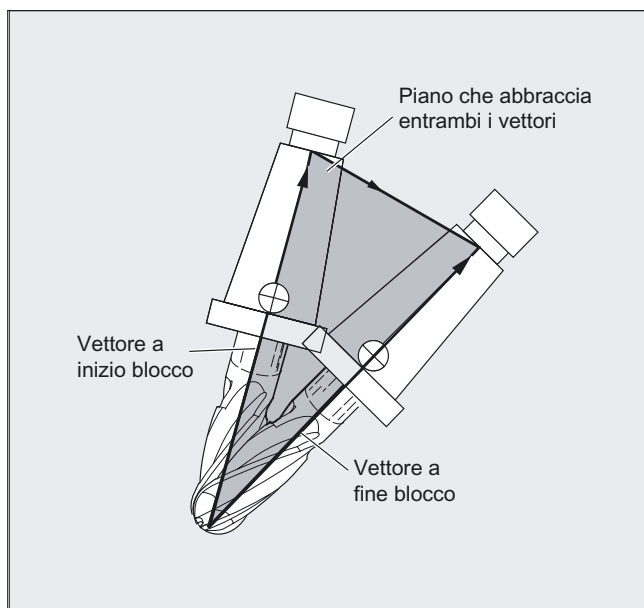
ORIWKS rappresenta la posizione di default. Se per un programma a cinque assi non si conosce a priori su quale macchina esso debba funzionare, va scelta sempre la funzione ORIWKS . I movimenti che la macchina esegue realmente dipendono dalla cinematica della macchina stessa.

Con ORIMKS è possibile programmare i movimenti reali di macchina per evitare, ad esempio, collisioni con gli attrezzi o altri oggetti.

Descrizione

Con ORIMKS il movimento eseguito dall'utensile **dipende** dalla cinematica della macchina. Nella variazione dell'orientamento, con punta dell'utensile fissa nello spazio, avviene un'interpolazione lineare tra le posizioni degli assi rotanti.

Con ORIWKS il movimento dall'utensile **non dipende** dalla cinematica della macchina. Nella variazione dell'orientamento, con punta dell'utensile fissa nello spazio, l'utensile muove nel piano definito dai vettori di partenza e di arrivo.



Posizioni particolari

Nota

ORIWKS

I movimenti di orientamento nel campo delle posizioni singolari della macchina a cinque assi richiedono ampi movimenti degli assi macchina. (ad esempio per una testa orientabile rotante con C come asse rotante ed A come asse di ribaltamento, tutte le posizioni con A=0 sono posizioni singolari).

Costruttore della macchina

Per non sovraccaricare gli assi di macchina, la gestione della velocità riduce la velocità vettoriale in prossimità delle posizioni singolari.

Con i dati macchina

`$MC_TRAFO5_NON_POLE_LIMIT`

`$MC_TRAFO5_POLE_LIMIT`

è possibile parametrizzare la trasformazione in modo che i movimenti di orientamento in prossimità del polo avvengano attraverso il polo stesso per consentire una lavorazione rapida.

A partire dalla versione SW 5.2 le posizioni singolari vengono elaborate solo con il dato macchina `$MC_TRAFO5_POLE_LIMIT`.

Bibliografia:

/FB3/ Manuale delle funzioni speciali; Trasformazione a 3 ... 5 assi (F2), capitolo "Posizioni singolari e relativa gestione".

7.2.7 Programmazione degli assi di orientamento (ORIAxes, ORIVect, ORIEuler, ORIRPY)

Funzione

La funzione degli assi di orientamento descrive l'orientamento dell'utensile nello spazio e viene ottenuta programmando l'offset per gli assi rotanti. Un ulteriore terzo grado di libertà può essere ottenuto ruotando ulteriormente l'utensile su sé stesso. Questo orientamento viene eseguito in una qualsiasi posizione nello spazio tramite un terzo asse rotante e richiede la trasformazione a sei assi. L'autorotazione dell'utensile su sé stesso viene definita in base al tipo di interpolazione dei vettori di rotazione con l'angolo di rotazione THETA, vedere il capitolo "Rotazioni dell'orientamento dell'utensile (ORIROT/TR/TT, ORIROT, THETA)".

Programmazione

Gli assi di orientamento vengono programmati tramite gli identificatori di asse A2, B2, C2.

N... ORIAxes oppure ORIVect	Interpolazione lineare oppure circolare
N... G1 X Y Z A B C	
oppure	oppure
N... ORIPLANE	Interpolazione di orientamento del piano
oppure	
N... ORIEuler oppure ORIRPY o ORIRPY2	oppure
N... G1 X Y Z A2= B2= C2=	Angolo di orientamento, angolo di Eulero/RPY
oppure	
N... ORIVIRT1 oppure ORIVIRT2	Programmazione angolare di assi virtuali
N... G1 X Y Z A3= B3= C3=	
	oppure
	assi di orientamento virtuali definizione 1 o 2 programmazione del vettore direzionale

Per le variazioni di orientamento lungo una superficie conica che si trova nello spazio possono essere programmati altri offset asse rotante degli assi di orientamento, vedere il capitolo "Programmazione dell'orientamento lungo una superficie conica (ORIPLANE, ORICONxx).

Parametri

ORIAxes	Interpolazione lineare degli assi macchina o degli assi di orientamento
ORIVect	Interpolazione del cerchio più grande (identico a ORIPLANE)
ORIMKS	Rotazione nel sistema di coordinate macchina
ORIWKS	Rotazione nel sistema di coordinate pezzo
	Per la descrizione vedere il capitolo Rotazioni dell'orientamento utensile
A= B= C=	Programmazione della posizione dell'asse di macchina

ORIEULER	Programmazione dell'orientamento tramite angolo di Eulero
ORIRPY	Programmazione dell'orientamento tramite angolo RPY. La successione di rotazione è XYZ, dove: A2 è l'angolo di rotazione intorno a X B2 è l'angolo di rotazione intorno a Y C2 è l'angolo di rotazione intorno a Z
ORIRPY2	Programmazione dell'orientamento tramite angolo RPY. La successione di rotazione è ZYX, dove: A2 è l'angolo di rotazione intorno a Z B2 è l'angolo di rotazione intorno a Y C2 è l'angolo di rotazione intorno a X
A2= B2= C2=	Programmazione angolare di assi virtuali
ORIVIRT1	Programmazione dell'orientamento tramite assi di orientamento virtuali
ORIVIRT2	(definizione 1), definizione secondo DM \$MC_ORIAX_TURN_TAB_1 (definizione 2), definizione secondo DM \$MC_ORIAX_TURN_TAB_2
A3= B3= C3=	Programmazione del vettore direzionale dell'asse direzionale

Descrizione

Costruttore della macchina

Con il dato macchina `$MC_ORI_DEF_WITH_G_CODE` si stabilisce come vengono definiti gli angoli programmati A2, B2, C2 :

La definizione avviene in base al dato macchina `$MC_ORIENTATION_IS_EULER` (Standard) oppure in base al gruppo G 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Con il dato macchina `$MC_ORI_IPO_WITH_G_CODE` viene definito il tipo di interpolazione attivo: ORIWKS/ORIMKS oppure ORIAXES/ORIVECT.

Modo operativo JOG

In questo modo operativo gli angoli di orientamento vengono interpolati sempre in modo lineare. In caso di movimento continuo e incrementale mediante i tasti di movimento, può essere traslato un solo asse di orientamento. Tramite i volantini è possibile muovere contemporaneamente gli assi di orientamento.

Per il movimento manuale degli assi di orientamento è attivo l'override avanzamento specifico per canale oppure l'override del rapido nella sovrapposizione del rapido.

Con i seguenti dati macchina è possibile impostare separatamente la velocità:

```
$MC_JOG_VELO_RAPID_GEO
$MC_JOG_VELO_GEO
$MC_JOG_VELO_RAPID_ORI
$MC_JOG_VELO_ORI
```

La funzione di movimento manuale cartesiano può essere impostata separatamente nel funzionamento in JOG con il SINUMERIK 840D power line e 840D solution line con il pacchetto "Pacchetto di trasformazione Handling" e SINUMERIK 810D power line dalla traslazione degli assi geometrici nei sistemi SCM, SCP e SCT.

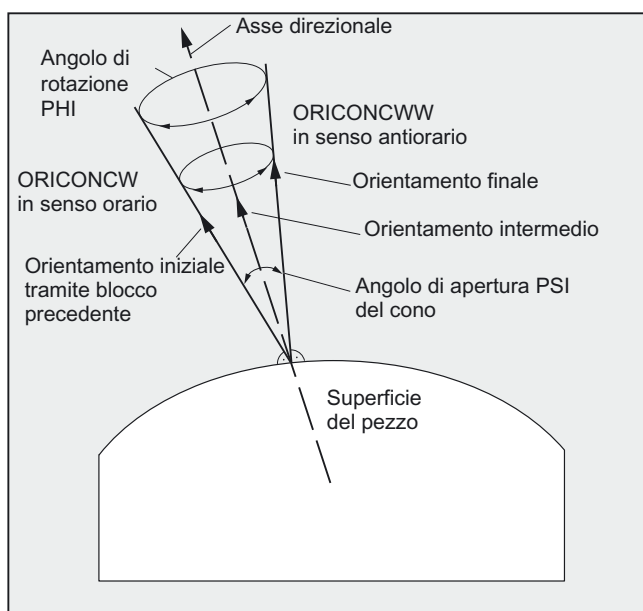
Riferimenti bibliografici

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazione cinematica (M1)

7.2.8 Programmazione dell'orientamento lungo una superficie conica (ORIPLANE, ORICONxx)

Funzione

Con l'orientamento ampliato è possibile eseguire modifiche dell'orientamento lungo una superficie conica che si trova nello spazio. L'interpolazione del vettore di orientamento su una superficie conica viene eseguita con i comandi modali ORICONxx. Per l'interpolazione in un piano può essere programmato un orientamento finale con ORIPLANE. Generalmente l'orientamento iniziale viene definito con i blocchi precedenti.

**Programmazione**

L'orientamento finale viene definito indicando la programmazione dell'angolo nell'angolo Eulero o nell'angolo RPY con A2, B2, C2 oppure programmando le posizioni dell'asse rotante con A, B, C. Per gli assi di orientamento lungo la superficie conica sono necessarie altre indicazioni di programmazione:

- Asse di rotazione del cono come vettore con A6, B6, C6
- Angolo di apertura ψ (PSI) con l'identificatore NUT
- Orientamento intermedio nel cono con A7, B7, C7

Nota

Programmazione del vettore direzionale A6, B6, C6 per l'asse rotante del cono

La programmazione di un orientamento finale non è indispensabile. Se non è indicato un orientamento finale, viene interpolato un cono intero con 360 gradi.

Programmazione dell'angolo di apertura del cono con NUT=angolo

L'indicazione di un orientamento finale è indispensabile.

Un cono completo con 360 gradi non può essere interpolato in questo modo.

Programmazione dell'orientamento intermedio A7, B7, C7 nel cono

L'indicazione di un orientamento finale è indispensabile. La modifica dell'orientamento e la direzione di rotazione vengono definite in modo univoco con i tre vettori orientamento iniziale, finale e intermedio. Tutti i tre vettori devono essere diversi. Se l'orientamento intermedio programmato è parallelo all'orientamento iniziale o finale, viene eseguita un'interpolazione del cerchio estesa dell'orientamento nel piano determinato dal vettore iniziale e finale.

Interpolazione dell'orientamento ampliato su una superficie conica

N... ORICONCW oppure ORICONCCW

N... A6= B6= C6= A3= B3= C3=
oppure

N... ORICONTO

N... G1 X Y Z A6= B6= C6=

oppure

N... ORICONIO

N... G1 X Y Z A7= B7= C7=

N... PO [PHI] = (a2, a3, a4, a5)

N... PO [PSI] = (b2, b3, b4, b5)

Interpolazione su un cono con

vettore direzionale in senso orario/antiorario del cono e orientamento finale oppure

raccordo tangenziale e indicazione dell'orientamento finale

oppure

indicazione dell'orientamento finale e di un

orientamento intermedio nel cono con

polinomi per l'angolo di rotazione e polinomi per l'angolo di apertura

Parametro

ORIPANE	Interpolazione nel piano (interpolazione circolare)
ORICONCW	Interpolazione su una superficie conica in senso orario
ORICONCCW	Interpolazione su una superficie conica in senso antiorario
ORICONTO	Interpolazione su una superficie conica con raccordo tangenziale
A6= B6= C6=	Programmazione dell'asse di rotazione del cono (vettore normalizzato)
NUT=angolo	Angolo di apertura del cono in gradi
NUT=+179	Angolo di movimento inferiore o uguale a 180 gradi
NUT=-181	Angolo di movimento superiore o uguale a 180 gradi
ORICONIO	Interpolazione su una superficie conica
A7= B7= C7=	Orientamento intermedio (programmazione come vettore normalizzato)

PHI	Angolo di rotazione dell'orientamento intorno all'asse direzionale del cono
PSI	Angolo di apertura del cono
polinomi possibili	Oltre ai rispettivi angolo possono essere programmati polinomi di max.
PO[PHI]=(a2, a3, a4, a5)	5° grado
PO[PSI]=(b2, b3, b4, b5)	

Esempio di diverse modifiche dell'orientamento

...	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	;Trasformazione orientamento ON
N30 ORIVECT	;Interpolazione dell'orientamento utensile come vettore
...	;Orientamento dell'utensile nel piano
N40 ORIPLANE	;Selezione dell'interpolazione cerchio estesa
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	;Orientamento nel piano Y/Z di 45 gradi
	;grado ruotato, alla fine del blocco viene raggiunto
	;l'orientamento (0, 1/√2, 1/√2)
...	;Programmazione dell'orientamento sul cono
N70 ORICONCW	;il vettore dell'orientamento viene interpolato su un cono con la
	;direzione (0,0,1) fino all'orientamento
	;(1/√2, 0, 1/√2) in senso orario
	;l'angolo di rotazione è quindi di 270 gradi
N90 A6=0 B6=0 C6=1	;l'orientamento dell'utensile compie una rotazione completa sullo stesso cono

Descrizione

Se si devono descrivere variazioni dell'orientamento su una superficie conica che si trova in uno spazio qualsiasi, si deve conoscere il vettore intorno al quale deve essere ruotato l'orientamento dell'utensile. Inoltre deve essere impostato l'orientamento iniziale e finale. L'orientamento iniziale risulta dal blocco precedente e l'orientamento finale deve essere programmato o definito da altre condizioni.

La programmazione nel piano ORIPLANE corrisponde a ORIVECT

La programmazione dell'interpolazione del cerchio estesa insieme ai polinomi dell'angolo corrisponde all'interpolazione lineare e del polinomio dei profili. L'orientamento degli utensili viene interpolato in un piano determinato dall'orientamento iniziale e finale. Se vengono programmati altri polinomi, il vettore di orientamento può essere invertito anche dal piano.

Programmazione di cerchi in un piano G2/G3, CIP e CT

L'orientamento ampliato corrisponde all'interpolazione di cerchi in un piano. Per le relative possibilità di programmazione di cerchi con indicazione del centro o indicazione del raggio come G2/G3, cerchio sul punto intermedio CIP e cerchi tangenziali CT vedere

Bibliografia: Manuale di programmazione, Nozioni di base, "Programmazione delle funzioni preparatorie".

Programmazione dell'orientamento

Interpolazione del vettore di orientamento su una superficie conica ORICONxx

Per l'interpolazione di orientamenti su una superficie conica possono essere selezionati quattro diversi tipi di interpolazione del gruppo codice G 51:

1. Interpolazione su un cono in senso orario ORICONCW con indicazione dell'orientamento finale e della direzione del cono o dell'angolo di apertura. Il vettore direzionale viene programmato con gli identificatori A6, B6, C6 e l'angolo di apertura del cono con l'identificatore NUT= campo di valori nell'intervallo 0 ... 180 gradi.
2. Interpolazione su un cono in senso antiorario ORICONCWW con indicazione dell'orientamento finale e della direzione del cono o dell'angolo di apertura. Il vettore direzionale viene programmato con gli identificatori A6, B6, C6 e l'angolo di apertura del cono con l'identificatore NUT= campo di valori nell'intervallo 0 ... 180 gradi.
3. Interpolazione su un cono ORICONIO con indicazione dell'orientamento finale e di un orientamento intermedio che viene programmato con gli identificatori A7, B7, C7.
4. Interpolazione su un cono ORICONTO con raccordo tangenziale e indicazione dell'orientamento finale. Il vettore direzionale viene programmato con gli identificatori A6, B6, C6.

7.2.9 Impostazione dell'orientamento di due punti di contatto (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Funzione

Programmazione della modifica dell'orientamento tramite la seconda curva nello spazio ORICURVE

Oltre alla punta dell'utensile lungo una curva nello spazio, un'ulteriore possibilità per programmare modifiche dell'orientamento consiste nella programmazione del movimento di un secondo punto di contatto dell'utensile con ORICURVE. In questo modo le modifiche dell'orientamento dell'utensile possono essere definite in modo univoco come per la programmazione del vettore dell'utensile stesso.

Costruttore della macchina

Osservare le avvertenze del costruttore della macchina relative agli identificatori degli assi impostabili tramite il dato macchina per la programmazione del secondo vettore di orientamento dell'utensile.

Programmazione

In questo tipo di interpolazione è possibile programmare punti con G1 o polinomi con POLY per entrambe le curve nello spazio. Non sono consentiti cerchi ed evolventi. Inoltre è possibile attivare un'interpolazione Spline con BSPLINE e con la funzione "Raggruppamento di blocchi spline brevi".

Bibliografia:

/FB1/ Manuale delle funzioni di base; Avanzamenti, arresto preciso, LookAhead (B1), capitolo: Raggruppamento di blocchi spline brevi

Gli altri tipi di spline ASPLINE e CSPLINE, nonché l'attivazione di un compressore con COMPON, COMPCURV o COMPCAD non sono consentiti.

Il movimento dei due punti di contatto dell'utensile può essere impostato durante la programmazione dei polinomi di orientamento per le coordinate fino a max. il 5° grado.

Interpolazione dell'orientamento ampliato con ulteriore curva nello spazio e polinomi per coordinate

N... ORICURVE	
N... PO [XH] = (xe, x2, x3, x4, x5)	Indicazione del movimento del secondo
N... PO [YH] = (ye, y2, y3, y4, y5)	punto di contatto dell'utensile e polinomi
N... PO [ZH] = (ze, z2, z3, z4, z5)	supplementari delle relative coordinate

Parametri

ORICURVE	Interpolazione dell'orientamento con preimpostazione del movimento di due punti di contatto dell'utensile
XH YH ZH	Identificatori delle coordinate del secondo punto di contatto dell'utensile del profilo supplementare come curva nello spazio
Polinomi possibili PO [XH] = (xe, x2, x3, x4, x5) PO [YH] = (ye, y2, y3, y4, y5) PO [ZH] = (ze, z2, z3, z4, z5)	Oltre ai rispettivi punti finali, si possono inoltre programmare curve nello spazio con polinomi.
xe, ye, ze	Punti finali della curva nello spazio
xi, yi, zi	Coefficienti dei polinomi al massimo di 5° grado

Nota

Identificatori XH YH ZH per la programmazione di un secondo vettore di orientamento

Gli identificatori devono essere selezionati in modo che non sussistano conflitti con altri identificatori degli assi lineari

assi X Y Z

e assi rotanti come

angolo Eulero o angolo RPY A2 B2 C2

vettori direzionali A3 B3 C3

vettori normali alla superficie A4 B4 C4 e A5 B5 C5

vettori di rotazione A6 B6 C6 e coordinate del punto intermedio A7 B7 C7

o altri parametri di interpolazione.

7.3 Polinomi di orientamento (PO[angolo], PO[coordinate])

Funzione

Indipendentemente da quale interpolazione di polinomio del gruppo codice G 1 è attualmente attiva, è possibile programmare due diversi tipi di polinomi di orientamento al massimo del 5° grado in una trasformazione a 3...5 assi.

1. Polinomi per **angolo**: angolo d'anticipo LEAD, angolo laterale TILT riferito al piano che viene determinato dall'orientamento iniziale e finale.
2. Polinomi per **coordinate**: XH, YH, ZH della seconda curva nello spazio per l'orientamento dell'utensile di un punto di riferimento sull'utensile.

In una trasformazione a sei assi, per l'orientamento dell'utensile è possibile inoltre programmare la rotazione del vettore di rotazione THT con polinomi al massimo fino al 5° grado per la rotazione dell'utensile stesso.

Programmazione

Polinomi di orientamento del tipo 1 per **angolo**

N... PO [PHI] = (a2, a3, a4, a5)	Trasformazioni a 3 ... 5 assi
oppure	
N... PO [PSI] = (b2, b3, b4, b5)	Trasformazioni a 3 ... 5 assi

Polinomi di orientamento del tipo 2 per **coordinate**

N... PO [XH] = (xe, x2, x3, x4, x5)	Identificatori per le coordinate del secondo vettore di orientamento per l'orientamento dell'utensile
N... PO [YH] = (ye, y2, y3, y4, y5)	
N... PO [ZH] = (ze, z2, z3, z4, z5)	

In entrambi i casi è inoltre possibile programmare un polinomio per la **rotazione** nelle trasformazioni a sei assi con

N... PO [THT] = (c2, c3, c4, c5)	interpolazione relativa del vettore della rotazione
oppure	
N... PO [THT] = (d2, d3, d4, d5)	interpolazione assoluta, relativa e tangenziale per la modifica dell'orientamento

del vettore di orientamento. Questo è possibile se è supportata la trasformazione di un vettore di rotazione con un offset programmabile e interpolabile tramite l'angolo di rotazione THETA.

Parametro

PO [PHI]	Angolo nel piano fra orientamento iniziale e finale
PO [PSI]	Angolo che descrive l'inversione dell'orientamento dal piano tra l'orientamento iniziale e quello finale
PO [THT]	Angolo di rotazione che descrive un codice G del gruppo 54 programmato con THETA tramite la rotazione del vettore di rotazione
PHI	Anticipo angolare LEAD
PSI	Angolo laterale TILT
THETA	Rotazione intorno alla direzione dell'utensile in Z
PO [XH]	Coordinata X del punto di riferimento sull'utensile
PO [YH]	Coordinata Y del punto di riferimento sull'utensile
PO [ZH]	Coordinata Z del punto di riferimento sull'utensile

Descrizione

I polinomi di orientamento non possono essere programmati

- se sono attive le interpolazioni spline ASPLINE, BSPLINE, CSPLINE.
I polinomi del tipo 1 per l'angolo di orientamento sono possibili per ogni tipo di interpolazione eccetto spline, vale a dire nelle interpolazioni lineari con rapido G00 o con avanzamento G01
nell'interpolazione dei polinomi con POLY e
nell'interpolazione del cerchio o evolvente con G02, G03, CIP, CT, INVCW e INCCCW
- I polinomi del tipo 2 per le coordinate dell'orientamento sono invece possibili solo se è attiva
l'interpolazione lineare con rapido G00 o con avanzamento G01 o
l'interpolazione dei polinomi con POLY.
- se l'orientamento viene interpolato tramite interpolazione asse ORIAXES. In questo caso possono essere programmati direttamente polinomi con PO[A] e PO[B] per gli assi di orientamento A e B.

Polinomi di orientamento del tipo 1 con ORIVECT, ORIPLANE e ORICONxx

Nell'interpolazione del cerchio estesa e nell'interpolazione del cono con ORIVECT, ORIPLANE e ORICONxx sono possibili solo polinomi di orientamento del tipo 1.

Polinomi di orientamento del tipo 2 con ORICURVE

Se è attiva l'interpolazione con curva nello spazio supplementare ORICURVE, vengono interpolati i componenti cartesiani del vettore di orientamento e sono possibili solo polinomi di orientamento del tipo 2.

7.4 Rotazioni dell'orientamento utensile (ORIOTA, ORIOTR/TT, ORIOTC, THETA)

Funzione

Se per determinati tipi di macchine con utensile in movimento deve essere modificabile anche l'orientamento dell'utensile, in ogni blocco è necessario programmare l'orientamento finale. In funzione della cinematica della macchina, è possibile programmare la direzione di orientamento degli assi di orientamento oppure il senso di rotazione del vettore di orientamento THETA. Per questi vettori di orientamento si possono programmare diverse tipologie di interpolazione:

- ORIOTA: Angolo di rotazione in un senso di rotazione preimpostato in modo assoluto.
- ORIOTR: Angolo di rotazione relativo per il piano fra orientamento iniziale e finale.
- ORIOTT: Angolo di rotazione riferito alla variazione del vettore di orientamento.
- ORIOTC: Angolo di rotazione tangenziale rispetto alla tangente vettoriale.

Programmazione

Solo se è attiva l'interpolazione ORIOTA l'angolo di rotazione oppure il vettore di rotazione possono essere programmati nei quattro modi seguenti:

1. direttamente le posizioni degli assi rotanti A, B, C
2. angolo di Eulero (in gradi) tramite A2, B2, C2
3. angolo RPY (in gradi) tramite A2, B2, C2
4. vettore direzionale tramite A3, B3, C3 (angolo di rotazione tramite THETA=<Valore>).

Se sono attivi ORIOTR oppure ORIOTT, l'angolo di rotazione può essere programmato solo direttamente THETA.

Una rotazione può essere programmata da sola in un blocco senza che avvenga una variazione dell'orientamento. ORIOTR e ORIOTT non hanno perciò alcun significato. In questo caso l'angolo di rotazione viene sempre interpretato riferito alla direzione assoluta (ORIOTA).

N... ORIOTA	Definizione dell'interpolazione del vettore di rotazione
oppure	
N... ORIOTR	
oppure	
N... ORIOTT	
oppure	
N... ORIOTC	Definizione della rotazione del vettore di orientamento
N... A3= B3= C3= THETA=valore	
N... PO [THT] = (d ₂ , d ₃ , d ₄ , d ₅)	Interpolazione dell'angolo di rotazione con polinomio di 5. grado

Parametri

ORIOTA	Angolo di rotazione in un senso di rotazione preimpostato in modo assoluto
ORIOTR	Angolo di rotazione relativo al piano fra orientamento iniziale e finale
ORIOTT	Angolo di rotazione come vettore di rotazione tangenziale per la modifica dell'orientamento
ORIOTC	Angolo di rotazione come vettore di rotazione tangenziale rispetto alla tangente vettoriale
THETA	Rotazione del vettore di orientamento
THETA=valore	Angolo di rotazione in gradi che viene raggiunto a fine blocco
THETA=Θ _e	Angolo di rotazione con angolo finale Θ _e del vettore di rotazione
THETA=AC (...)	Commutazione all'impostazione in quote assolute nel blocco
THETA=AC (...)	Commutazione nell'impostazione in quote incrementali nel blocco
Θ _e	L'angolo finale del vettore di rotazione è attivo sia assoluto con G90 che relativo con G91 (impostazione incrementale delle quote)
PO[THT] = (. . . .)	Polinomio per l'angolo di rotazione

Esempio di rotazioni degli orientamenti

N10 TRAORI	;attivazione della trasformazione orientamento
N20 G1 X0 Y0 Z0 F5000	
	;orientamento dell'utensile
N30 A3=0 B3=0 C3=1 THETA=0	;in direzione Z con angolo di rotazione 0
N40 A3=1 B3=0 C3=0 THETA=90	;in direzione X e rotazione di 90 gradi
	;orientamento
N50 A3=0 B3=1 C3=0 PO[THT] = (180, 90)	;in direzione Y e rotazione a 180 gradi
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	;rimane costante e rotazione a 90 gradi
N70 ORIOTT	;angolo di rotazione relativo alla variazione dell'orientamento
N80 A3=1 B3=0 C3=0 THETA=30	;vettore di rotazione nell'angolo di 30 gradi rispetto al pianto X Y

Durante l'interpolazione del blocco

N40 l'angolo di rotazione viene interpolato linearmente dal valore iniziale di 0 gradi al valore finale di 90 gradi. Nel blocco N50 l'angolo di rotazione varia da 90 gradi a 180 gradi in base alla

parabola $\theta(u) = +90u^2$. In N60 può essere eseguita anche una rotazione senza che si verifichi una modifica dell'orientamento.

In N80 l'orientamento dell'utensile viene ruotato dalla direzione Y alla direzione X. La modifica dell'orientamento si trova quindi nel piano X-Y e il vettore di rotazione forma rispetto a questo piano un angolo di 30 gradi.

Descrizione

ORIOTA

L'angolo di rotazione THETA viene interpolato in funzione di una direzione assoluta stabilita nello spazio. La direzione di rotazione di base è definita nei dati macchina

ORIROTR

L'angolo di rotazione **THETA** viene considerato relativo al piano che si apre tra l'angolo di orientamento iniziale e finale.

ORIROTT

L'angolo di rotazione **THETA** viene considerato relativo alla variazione dell'orientamento. Per **THETA=0** il vettore di rotazione viene interpolato in modo tangenziale alla variazione dell'orientamento e si differisce rispetto a **ORIROTR**, solo quando per l'orientamento è stato programmato almeno un polinomio per "l'angolo di rovesciamento **PSI**". In questo modo ne deriva una variazione dell'orientamento che non avviene nel piano. Con un angolo di rotazione supplementare programmabile **THETA** il vettore di rotazione può essere quindi interpolato in modo che esso formi sempre un determinato valore rispetto alla variazione dell'orientamento.

ORIOTC

Il vettore direzionale viene interpolato in modo relativo rispetto alla tangente vettoriale con un offset programmabile con l'angolo **THETA**. Per l'angolo offset può essere programmato anche un polinomio **PO [THT] = (c2, c3, c4, c5)** di max. 5° grado.

7.5 Orientamenti relativi al percorso

7.5.1 Tipi di orientamento relativi al percorso

Funzione

Con questa funzione ampliata l'orientamento relativo non viene ottenuto solo a fine blocco, ma sull'intero percorso. Viene trasformato l'orientamento ottenuto nel blocco precedente tramite interpolazione del cerchio estesa nell'orientamento finale programmato. In linea di principio esistono due possibilità di programmare l'orientamento desiderato relativo al percorso:

1. L'orientamento dell'utensile e la rotazione dell'utensile vengono interpolate relativamente al percorso con **ORIPATH**, **ORPATHS**.
2. Il vettore di orientamento viene programmato e interpolato come di consueto. Con **ORIOTC** la rotazione del vettore di orientamento viene accostata in modo relativo alla tangente del percorso.

Programmazione

Il tipo di interpolazione dell'orientamento e della rotazione dell'utensile viene programmato con:

N... **ORIPATH**

Orientamento relativo al percorso

N... **ORIPATHS**

Orientamento relativo al percorso con spianamento del percorso di orientamento

N... **ORIOTC**

Interpolazione del vettore di rotazione relativo al percorso

Una piega dell'orientamento causata da uno spigolo nell'andamento del percorso può essere livellata con ORIPATHS. La direzione e la lunghezza del percorso del movimento di svincolo viene programmato tramite il vettore con i componenti $A8=X$, $B8=Y$ $C8=Z$.

Con ORIPATH/ORIPATHS possono essere programmati diversi riferimenti alla tangente del percorso tramite i tre angoli

- LEAD= indicazione dell'angolo d'anticipo riferito al percorso e alla superficie
- TILT= indicazione dell'angolo laterale riferito al percorso e alla superficie
- THETA= Angolo di rotazione

per l'intero andamento del percorso. Per l'angolo di rotazione THETA possono essere programmati con $PO[THT] = (\dots)$ ulteriori polinomi di max. 5° grado.

Nota

Costruttore della macchina

Fare attenzione alle indicazioni del costruttore della macchina. Con i dati macchina e di setting progettabili è possibile eseguire ulteriori impostazioni per il tipo di orientamento relativo al percorso. Per ulteriori descrizioni vedere

Bibliografia:

/FB3/ Manuale delle funzioni speciali; Trasformazione a 3 ... 5 assi (F2), capitolo "Orientamento"

Parametro

L'interpolazione dell'angolo LEAD e TILT può essere impostata in diversi modi tramite il dato macchina:

- Il riferimento programmato con LEAD e TILT dell'orientamento dell'utensile viene mantenuto per l'intero blocco.
- Anticipo angolare LEAD: Rotazione intorno alla direzione in posizione ortogonale rispetto alla tangente e vettore normale TILT: Rotazione dell'orientamento intorno al vettore normale.
- Anticipo angolare LEAD: Rotazione intorno alla direzione in posizione ortogonale rispetto alla tangente e vettore normale angolo laterale TILT: Rotazione dell'orientamento intorno alla direzione della tangente del percorso.
- Angolo di rotazione THETA: Rotazione dell'utensile su sé stesso con un ulteriore terzo asse rotante come asse di orientamento per la trasformazione a 6 assi.

Nota

Orientamento relativo al percorso non consentito in combinazione con OSC, OSS, OSSE, OSD, OST

L'interpolazione dell'orientamento relativa al percorso ORIPATH o ORIPATHS e ORIOTC non può essere programmata in combinazione con uno spianamento del percorso di orientamento con uno dei codici G del gruppo 34. In questo caso OSOF deve essere attivo.

7.5.2 Rotazione relativa al percorso dell'orientamento dell'utensile (ORIPATH, ORIPATHS, angolo di rotazione)

Funzione

In una trasformazione a sei assi, per l'orientamento dell'utensile in una posizione qualsiasi nello spazio, anche l'utensile può essere ruotato su sé stesso con un terzo asse rotante. Per la rotazione relativa al percorso dell'orientamento dell'utensile con ORIPATH o ORIPATHS può essere programmata la rotazione supplementare tramite l'angolo di rotazione THETA. In alternativa gli angoli LEAD e TILT possono essere programmati con un vettore che si trova nel piano in posizione ortogonale rispetto alla direzione dell'utensile.

Costruttore della macchina

Fare attenzione alle indicazioni del costruttore della macchina. Con il dato macchina è possibile impostare diversamente l'interpolazione degli angoli LEAD e TILT.

Programmazione

Rotazione dell'orientamento utensile e dell'utensile

Il tipo di orientamento dell'utensile relativo al percorso viene attivato con ORIPATH o ORIPATHS.

N... ORIPATH

Attivazione del tipo di orientamento riferito al percorso

N... ORIPATHS

Attivazione del tipo di orientamento riferito al percorso con spianamento del percorso di orientamento

Attivazione dei tre angoli possibili con effetto di rotazione:

N... LEAD=

Angolo per l'orientamento programmato relativo al vettore normale alla superficie

N... TILT=

Angolo per l'orientamento programmato nel piano in posizione ortogonale rispetto alla tangente del percorso relativo al vettore normale alla superficie

N... THETA=

Angolo di rotazione relativo alla modifica dell'orientamento intorno alla direzione dell'utensile del terzo asse rotante

I valori dell'angolo a fine blocco vengono programmati con LEAD=valore, TILT=valore o THETA=valore. Oltre agli angoli costanti, per tutti e tre gli angoli possono essere programmati polinomi di max. 5° grado.

N... PO[PHI] = (a2, a3, a4, a5)
oppure

Polinomio per l'angolo d'anticipo LEAD

N... PO[PSI] = (b2, b3, b4, b5)
oppure

Polinomio per l'angolo laterale TILT

N... PO[THT] = (d2, d3, d4, d5)

Polinomio per l'angolo di rotazione THETA

Durante la programmazione è possibile tralasciare i coefficienti dei polinomi più alti che sono pari a zero. Esempio Con PO[PHI] = a2 si ottiene una parabola per l'angolo d'anticipo LEAD.

Parametro

Orientamento dell'utensile relativo al percorso

ORIPATH	Orientamento dell'utensile riferito al percorso
ORIPATHS	L'orientamento dell'utensile riferito alla piega nel percorso di orientamento viene raddrizzato
LEAD	Angolo relativo al vettore normale alla superficie nel piano determinato dalla tangente al percorso e dal vettore normale alla superficie
TILT	Rotazione dell'orientamento intorno alla direzione Z o rotazione intorno alla tangente del percorso
THETA	Rotazione intorno alla direzione dell'utensile verso Z
PO[PHI]	Polinomio dell'orientamento per l'angolo d'anticipo LEAD
PO[PSI]	Polinomio dell'orientamento per l'angolo laterale TILT
PO[THT] (	Polinomio dell'orientamento per l'angolo di rotazione THETA

Nota

Angolo di rotazione THETA

Per la rotazione dell'utensile su sé stesso con un terzo asse rotante come asse di orientamento è necessaria una trasformazione a 6 assi.

7.5.3 Interpolazione relativa al percorso della rotazione dell'utensile (ORIOTC, THETA)

Funzione

Interpolazione con vettori di rotazione

Il vettore di rotazione può essere interpolato con la rotazione dell'utensile relativa alla tangente vettoriale programmata con ORIOTC anche con un offset programmabile mediante l'angolo di rotazione THETA. Per l'angolo di offset può essere programmato un polinomio di max. 5° grado con PO[THT].

Programmazione

N... ORIOTC

Controllo della rotazione dell'utensile relativa alla tangente vettoriale

N... A3= B3= C3= THETA=valore

Definizione della rotazione del vettore di orientamento

N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)

Interpolazione dell'angolo di offset con un polinomio di max. 5° grado

Una rotazione può essere programmata da sola in un blocco senza che avvenga una variazione dell'orientamento.

Parametro

Interpolazione relativa al percorso della rotazione dell'utensile nella trasformazione a sei assi

ORIROTC	Posizionamento del vettore di rotazione tangenziale rispetto alla tangente vettoriale
THETA=valore	Angolo di rotazione in gradi che viene raggiunto a fine blocco
THETA= θ_e	Angolo di rotazione con angolo finale θ_e del vettore di rotazione
THETA=AC (...)	Commutazione all'impostazione in quote assolute nel blocco
THETA=IC (...)	Commutazione nell'impostazione in quote incrementali nel blocco
PO [THT] = (C ₂ , C ₃ , C ₄ , C ₅)	Interpolazione dell'angolo di offset con un polinomio di 5° grado

Nota

Interpolazione del vettore di rotazione ORIROTC

Se anche la rotazione dell'utensile rispetto alla tangente vettoriale deve essere impostata nella direzione opposta all'orientamento dell'utensile, ciò può avvenire solo in una trasformazione a sei assi.

Con ORIROTC attivo

Il vettore di rotazione ORIROTA non può essere programmato. In caso di programmazione viene emesso l'ALLARME 14128 "Programmazione assoluta della rotazione utensile con ORIROTC attivo".

Direzione di orientamento dell'utensile per trasformazioni a tre ... cinque assi

La direzione di orientamento dell'utensile può essere programmata mediante l'angolo di Eulero, l'angolo RPY o il vettore direzionale, come nel caso di una trasformazione a tre ... cinque assi. Inoltre sono possibili modifiche dell'orientamento dell'utensile nello spazio tramite programmazione dell'interpolazione del cerchio estesa ORIVECT, dell'interpolazione lineare degli assi di orientamento ORIAXES, di tutte le interpolazioni su una superficie conica ORICONxx, nonché dell'interpolazione aggiuntiva della curva nello spazio con due punti di contatto dell'utensile ORICURVE.

G	Impostazione del tipo di movimento degli assi rotanti
X Y Z	Impostazione degli assi lineari
ORIAXES	Interpolazione lineare degli assi macchina o degli assi di orientamento
ORIVECT	Interpolazione del cerchio estesa (identico a ORIPLANE)
ORIMKS	Rotazione nel sistema di coordinate macchina
ORIWKS	Rotazione nel sistema di coordinate pezzo
	Per la descrizione vedere il capitolo Rotazioni dell'orientamento utensile
A= B= C=	Programmazione della posizione dell'asse di macchina
ORIEULER	Programmazione dell'orientamento tramite angolo di Eulero
ORIRPY	Programmazione dell'orientamento tramite angolo RPY
A2= B2= C2=	Programmazione angolare di assi virtuali
ORIVIRT1	Programmazione dell'orientamento tramite assi di orientamento virtuali
ORIVIRT2	(definizione 1), definizione secondo DM \$MC_ORIAX_TURN_TAB_1 (definizione 2), definizione secondo DM \$MC_ORIAX_TURN_TAB_2

A3= B3= C3=	Programmazione del vettore direzionale dell'asse direzionale
ORIPLANE	Interpolazione nel piano (interpolazione del cerchio estesa)
ORICONCW	Interpolazione su una superficie conica in senso orario
ORICONCCW	Interpolazione su una superficie conica in senso antiorario
ORICONTO	Interpolazione su una superficie conica con raccordo tangenziale
A6= B6= C6=	Programmazione dell'asse di rotazione del cono (vettore normalizzato)
NUT=angolo	Angolo di apertura del cono in gradi
NUT=+179	Angolo di posizionamento minore o uguale 180 gradi
NUT=-181	Angolo di posizionamento maggiore o uguale 180 gradi
ORICONIO	Interpolazione su una superficie conica
A7= B7= C7=	Orientamento intermedio (programmazione come vettore normalizzato)
ORICURVE	Interpolazione dell'orientamento con preimpostazione del movimento di due punti di contatto dell'utensile Oltre ai rispettivi punti finali, si possono programmare ulteriori polinomi di curve nello spazio.
XH YH ZH ad es. con polinomi PO [XH] = (xe, x2, x3, x4, x5)	

Nota

Se l'orientamento utensile con ORIAxes attivo viene interpolato tramite gli assi di orientamento, l'accostamento dell'angolo di rotazione viene raggiunto solo a fine blocco.

7.5.4 Spianamento del percorso di orientamento (ORIPATHS A8=, B8=, C8=)

Funzione

Nelle modifiche dell'orientamento con accelerazione costante sul profilo vanno evitate le interruzioni dei movimenti vettoriali, le quali possono verificarsi specialmente sugli angoli del profilo. La piega risultante nel percorso di orientamento può essere spianata inserendo un apposito blocco intermedio. La modifica dell'orientamento avviene quindi con accelerazione costante, se durante la modifica dell'orientamento è attivo anche ORIPATHS. In questa fase è possibile eseguire un movimento di svincolo dell'utensile.

Costruttore della macchina

Rispettare le indicazioni del costruttore della macchina sugli eventuali dati macchina e dati di setting predefiniti con i quali viene attivata questa funzione.

Il dato macchina consente di impostare come viene interpretato il vettore di svincolo:

1. nel sistema di coordinate dell'utensile la coordinata Z viene definita mediante la direzione utensile.
2. nel sistema di coordinate del pezzo la coordinata Z viene definita mediante il piano attivo.

Per ulteriori dettagli sulla funzione "Orientamento relativo al percorso" vedere

Bibliografia: /FB3/ Manuale delle funzioni speciali; Trasformazione a 3 ... 5 assi (F2)

Programmazione

Per gli orientamenti utensile costanti riferiti all'intero percorso, su un angolo del profilo vanno effettuate ulteriori operazioni di programmazione. La direzione e la lunghezza del percorso di questo movimento vengono programmate tramite il vettore con i componenti A8=X, B8=Y, C8=Z:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Parametro

ORIPATHS	Orientamento dell'utensile riferito al percorso, viene raddrizzata una piega durante l'orientamento.
A8= B8= C8=	Componenti vettore per direzione e lunghezza del percorso
X, Y, Z	Movimento di svincolo nella direzione dell'utensile

Nota

Programmazione del vettore direzionale A8, B8, C8

Se la lunghezza di questo vettore è uguale a zero non ha luogo alcun movimento di svincolo.

ORIPATHS

L'orientamento utensile riferito al percorso viene attivato con ORIPATHS. In caso contrario, l'orientamento viene commutato dall'orientamento iniziale a quello finale mediante interpolazione del cerchio estesa lineare.

7.6 Compressione dell'orientamento COMPON (A..., B..., C..., THETA)

Funzione

I programmi NC nei quali l'orientamento è programmato mediante vettori direzionali possono essere compressi rispettando le tolleranze preimpostate. Il compressore per orientamenti è disponibile solo in combinazione con una trasformazione per l'orientamento.

Costruttore della macchina

Il movimento per orientamento viene compresso solo se è attiva l'interpolazione del cerchio estesa e quindi dipende dal codice G per l'interpolazione di orientamento. Questo codice è impostabile con i dati macchina proprio come la lunghezza massima del percorso e la tolleranza ammessa per ogni asse per l'avanzamento vettoriale per la funzione compressore. Fare attenzione alle indicazioni del costruttore della macchina

Programmazione

Formato base del blocco NC

I blocchi da comprimere devono contenere solo il numero di blocco, l'interpolazione lineare G1, gli indirizzi degli assi, l'avanzamento e un commento e dunque presentano la seguente sintassi di programma:

```
N... G1 X=... Y=... Z=... A=... B=... ;Commento
F=...
```

I valori di posizione possono essere inseriti direttamente, ad es. X90, oppure indirettamente mediante l'assegnazione di parametri $X=R1*(R2+R3)$.

Con trasformazione di orientamento TRAORI attiva

L'orientamento dell'utensile può essere programmato indipendentemente dalla cinematica.

Per una macchina con **trasformazione a tre ... cinque assi** vale:

```
N... TRAORI
A3=... B3=... C3=... ;vettore direzionale
A2=... B2=... C2=... ;angolo di Eulero o angolo RPY
```

Nelle macchine **con trasformazione a sei assi** oltre all'orientamento utensile può essere programmata anche la rotazione dell'utensile.

```
N... X... Y... Z... A3=... B3=... C3=... THETA=... F=... opp.
N... X... Y... Z... A2=... B2=... C2=... THETA=... F=...
```

Se l'orientamento dell'utensile viene indicato con le posizioni dell'asse di rotazione, ad es. nel formato

```
N... X... Y... Z... A=... B=... THETA=... F=...
```

la compressione viene interpretata in modo differente a seconda che venga eseguita o meno l'interpolazione del cerchio estesa. Se non viene eseguita alcuna interpolazione del cerchio estesa la modifica dell'orientamento compresso verrà rappresentata con polinomi assiali per gli assi rotanti.

Parametro

Le assegnazioni dei parametri valide per il compressore possono essere utilizzate anche per le posizioni degli assi di rotazione.

TRAORI	Attivazione trasformazione di orientamento
COMPON	Compressore ON
G1	Interpolazioni lineare
X= Y= Z=	Indirizzi assi lineari
A= B= C=	Programmazione diretta posizioni assi di rotazione
A2= B2= C2=	Indirizzi assi di rotazione con angolo di Eulero o angolo RPY
A3= B3= C3=	Indirizzi assi di rotazione del vettore direzionale
THETA	Rotazione del vettore di orientamento
E	Avanzamento vettoriale

Per ulteriori dettagli sulla programmazione dell'angolo di rotazione THETA=... vedere il capitolo

"Rotazione dell'orientamento utensile (ORIOTA/TR/TT, ORIOTC, THETA).

Nota

Compressione solo con interpolazione del cerchio estesa attiva

Questo caso si verifica quando l'orientamento utensile si modifica nel piano determinato dall'orientamento iniziale e finale. Le condizioni valide per un'interpolazione del cerchio estesa vanno impostate tramite i dati macchina.

1. dato macchina: codice G per l'interpolazione di orientamento = FALSE
ORIWKS attivo e orientamento programmato come vettore con A3, B3, C3 oppure 2, B2, C2.

2. dato macchina: codice G per l'interpolazione di orientamento = TRUE
ORIVECT o ORIPLANE attivo. L'orientamento dell'utensile può essere programmato o come vettore di direzione o con le posizioni dell'asse di rotazione. Se è attivo uno dei codici G ORICONxx o ORICURVE o se sono stati programmati dei polinomi per l'angolo di orientamento (PO[PHI] e PO[PSI]), l'interpolazione del cerchio estesa non viene eseguita e quindi questi blocchi non vengono compressi.

Esempio "Compressore per orientamenti"

Nell'esempio che segue viene realizzata la compressione di un cerchio che viene accostato con un tratto poligonale. L'orientamento dell'utensile avviene sincrono ad una superficie conica. Sebbene le modifiche dell'orientamento programmate in successione avvengano in modo discontinuo, il compressore genera un andamento costante dell'orientamento.

<pre> DEF INT NUMERO = 60 DEF REAL RAGGIO = 20 DEF INT COUNTER DEF REAL ANGOLO N10 G1 X0 Y0 F5000 G64 \$SC_COMPRESS_CONTUR_TOL = 0.05 \$SC_COMPRESS_ORI_TOL = 5 TRAORI COMPCURV N100 X0 Y0 A3=0 B3==1 N110 FOR COUNTER = 0 TO NUMERO N120 ANGOLO= 360 * COUNTER /NUMERO N130 X=RAGGIO*COS(ANGOLO) Y=RAGGIO* SIN(ANGOLO) A3=SIN(ANGOLO) B3=(ANGOLO) C3=1 N140 ENDFOR ... </pre>	<pre> ;Scostamento massimo dal profilo: 0.05 mm ;Scostamento massimo dell'orientamento: ;5 gradi ;Il movimento avviene su un cerchio che è ;formato da poligoni. ;Quindi l'orientamento avviene su un cono ;intorno all'asse Z con un ;angolo di apertura di ;45 gradi. </pre>

Descrizione

Precisione

I blocchi NC possono essere compressi solo se sono consentiti scostamenti del profilo da quello programmato. Lo scostamento max. è impostabile come tolleranza di compressione nei dati di setting. Maggiori sono le tolleranze consentite, maggiore sarà il numero di blocchi che possono essere compressi.

Precisione degli assi

Per ogni asse il compressore genera una Spline che può avere uno scostamento max. dal punto finale programmato per ogni asse pari al valore di tolleranza impostato con il dato macchina assiale.

Precisione del profilo

Vengono verificati gli scostamenti max. dalla geometria del profilo (assi geometrici) e quelli dell'orientamento dell'utensile. Questo avviene con i dati di setting per:

1. max. tolleranza per il profilo
2. max. scostamento angolare per l'orientamento utensile
3. max. scostamento angolare dell'angolo di rotazione THEATA dell'utensile (solo per macchine a 6 assi)

Con il dato di macchina DM 20482 COMPRESSOR_MODE specifico per un canale si possono preimpostare i valori di tolleranza:

0: Precisione degli assi: tolleranze assiali per tutti gli assi (assi geometrici e assi di orientamento).

1: Precisione del profilo: impostazione della tolleranza massima (1°), tolleranza per l'orientamento tramite la tolleranza assiale (a.).

2: impostazione dello scostamento angolare per l'orientamento utensile (2°), tolleranza per il profilo tramite la tolleranza assiale (a.).

3: impostazione della tolleranza sul profilo con (1°) e Impostazione del massimo scostamento angolare per l'orientamento dell'utensile (2°).

La preimpostazione del max.disassamento assiale per l'orientamento dell'utensile è possibile solo quando è attiva una trasformazione dell'orientamento (TRAORI).

7.7 Correzione della lunghezza utensile on line (TOFFON, TOFFOF)

Funzione

Con la variabile \$AA_TOFF[] si può eseguire una correzione tridimensionale in tempo reale sulla base delle tre dimensioni effettive dell'utensile.

Come indice si utilizzano i tre identificativi degli assi geometrici. In questo modo il numero delle direzioni di correzione attive viene definito attraverso gli assi geometrici abilitati nello stesso momento.

Tutte le correzioni possono essere abilitate contemporaneamente.

La funzione di correzione online della lunghezza utensile è utilizzabile in questi casi:

- Trasformazione dell'orientamento TRAORI
- carrello portautensili orientabile TCARR

Costruttore della macchina

La correzione online della lunghezza utensile è un'**opzione** che va abilitata prima dell'uso. La funzione è utile solo se abbinata a una trasformazione dell'orientamento attiva o a carrello portautensili orientabile attivo.

Programmazione

```
N.. TRAORI
```

```
N.. TOFFON(X,25)
```

```
N.. WHEN TRUE DO $AA_TOFF[direzione utensile] in azioni sincrone
```

Per ulteriori dettagli sulla programmazione della correzione online della lunghezza utensile in azioni sincrone di movimento vedere il capitolo "Azioni in azioni sincrone".

Parametri

TOFFON	Tool Offset ON (attivare la correzione online della lunghezza utensile) Quando si attiva la funzione, per la relativa direzione di correzione si può indicare un valore di offset che verrà subito eseguito.
TOFFOF	Tool Offset ON (disattivare la correzione online della lunghezza utensile) I valori di correzione vengono resettati e viene attivato uno stop preelaborazione.
X, Y, Z,	Direzione di correzione per il valore di offset indicato in TOFFON

Esempio di attivazione della correzione lunghezza utensile

DM 21190): TOFF_MODE =1	;vengono raggiunti i valori assoluti
DM 21194): TOFF_VELO[0] =1000	
DM 21196): TOFF_VELO[1] =1000	
DM 21194): TOFF_VELO[2] =1000	
DM 21196): TOFF_ACCEL[0] =1	
DM 21196): TOFF_ACCEL[1] =1	
DM 21196): TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	;trasformazione ON
N20 TOFFON(Z)	;Attivazione della correzione online della lunghezza utensile per la direzione utensile Z
N30 WHEN TRUE DO \$AA_TOFF[Z] = 10	;per la direzione utensile Z viene interpolata una
G4 F5	;correzione della lunghezza utensile di 10
...	
N100 XOFFSET = \$AA_TOFF_VAL[X]	;assegnare la correzione attuale in direzione X
N120 TOFFON(X, -XOFFSET)	;per la direzione utensile X la
G4 F5	;correzione della lunghezza utensile viene riportata a 0

Esempio di disattivazione della correzione lunghezza utensile

N10 TRAORI(1)	;trasformazione ON
N20 TOFFON(X)	;attivazione della direzione utensile Z
N30 WHEN TRUE DO \$AA_TOFF[X] = 10	;per la direzione utensile X viene interpolata una
G4 F5	;correzione della lunghezza utensile di 10
...	
N80 TOFFOF(X)	;l'offset di posizione della direzione utensile; in X viene cancellato: ...\$AA_TOFF[X] = 0 ;Gli assi non vengono mossi ;alla posizione attuale nel sistema SCP viene aggiunto l'offset di posizione in base ;all'orientamento attuale

Descrizione**Preparazione del blocco**

Nella preparazione del blocco in fase di preelaborazione viene preso in considerazione l'offset attuale per la lunghezza utensile valido nel ciclo principale. Per poter sfruttare completamente la velocità max. consentita per gli assi, è necessario interrompere la preparazione del blocco con uno stop della preelaborazione STOPRE mentre si predispone un offset per l'utensile.

L'offset per l'utensile nella fase di preelaborazione viene sempre considerato anche quando le correzioni delle lunghezze dell'utensile non vengono più modificate dopo l'avvio del programma oppure quando, dopo una modifica di queste correzioni, è stato elaborato un numero di blocchi superiore a quello accettato dal buffer IPO fra la fase di preelaborazione e il ciclo principale.

Variabile \$AA_TOFF_PREP_DIFF

Il valore della differenza fra la correzione attuale attiva nell'interpolatore e la correzione che era attiva nella fase di preparazione del blocco può essere interrogato nella variabile \$AA_TOFF_PREP_DIFF[] .

Impostazione dei dati macchina e dati setting

Per la correzione online della lunghezza utensile sono disponibili i seguenti dati macchina:

- DM 20610: ADD_MOVE_ACCEL_RESERVE Riserva di accelerazione per movimento sovrapposto.
- DM 21190: TOFF_MODE contenuto della variabile di sistema \$AA_TOFF[] , viene indicato come valore assoluto oppure viene aggiunto.
- DM 21194: TOFF_VELO velocità della correzione della lunghezza utensile online.
- DM 21196: TOFF_ACCEL accelerazione della correzione della lunghezza utensile online.
- Dato setting per l'impostazione dei valori limite
DS 42970: TOFF_LIMIT limite superiore del valore di correzione lunghezza utensile.

Bibliografia: /FB3/ Manuale delle funzioni speciali; Trasformazioni a 3 ... 5 assi (F2).

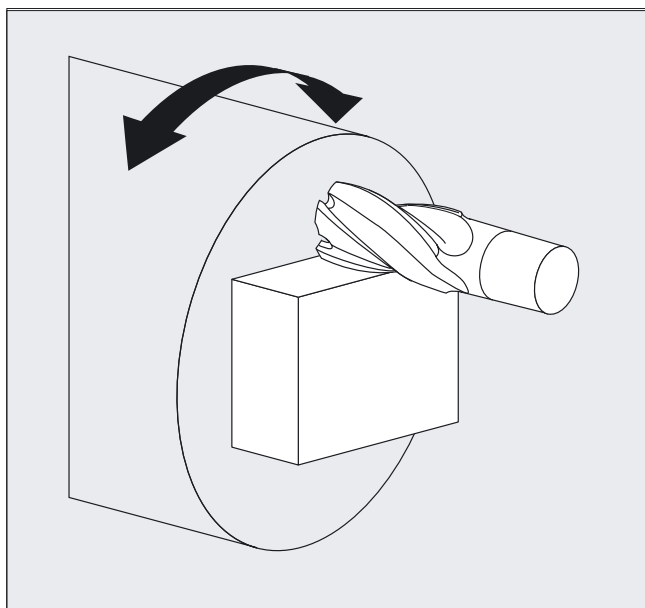
7.8 Trasformazione cinematica

7.8.1 Fresatura su pezzi torniti (TRANSMIT)

Funzione

La funzione TRANSMIT permette di eseguire:

- Lavorazione frontale su pezzi torniti e bloccati (fori, profili).
- Per la programmazione di queste lavorazioni può essere utilizzato un sistema di coordinate cartesiane.
- Il controllo numerico trasforma i movimenti programmati del sistema di coordinate cartesiane nei movimenti degli assi macchina reali (caso standard):
 - Asse rotante
 - Asse di incremento perpendicolare all'asse rotante
 - Asse longitudinale parallelo all'asse di rotazione
 - Gli assi lineari sono ortogonali tra di loro.
- Lo spostamento del centro dell'utensile rispetto al centro di rotazione è consentito.
- La guida della velocità tiene conto delle limitazioni definite per i movimenti rotatori.



Tipi di trasformazione TRANSMIT

Per le lavorazioni TRANSMIT si possono impostare due varianti:

- TRANSMIT come default con (TRAFO_TYPE_n = 256)
- TRANSMIT con asse Y lineare supplementare (TRAFO_TYPE_n = 257)

Il tipo di trasformazione ampliato 257 può essere utilizzato, ad esempio, per compensare correzioni del bloccaggio di un utensile con l'asse reale Y.

Programmazione

TRANSMIT o TRANSMIT(n)

oppure

TRAFOOF

Asse rotante

L'asse rotante non può essere programmato in quanto viene occupato da un asse geometrico e quindi non è più programmabile direttamente come asse canale.

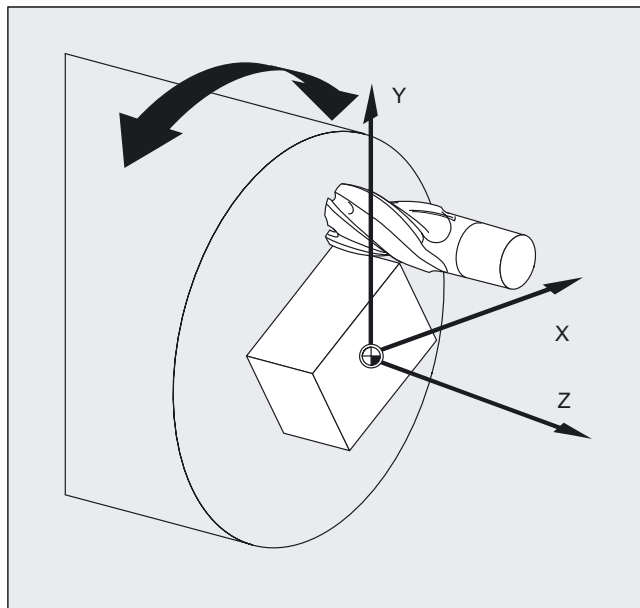
Parametri

TRANSMIT	Attiva la prima funzione TRANSMIT definita. Questa funzione viene anche denominata Trasformazione polare.
TRANSMIT (n)	Attiva la successiva funzione TRANSMIT concordata; n può essere al massimo 2 (TRANSMIT(1) corrisponde a TRANSMIT).
TRAFOOF	Disattiva una trasformazione attiva
OFFN	Offset profilo normale: distanza della lavorazione frontale dal profilo di riferimento programmato

Nota

Una trasformazione TRANSMIT attiva viene anche disattivata quando nel rispettivo canale viene attivata una delle restanti trasformazioni (es. TRACYL, TRAANG, TRAORI).

Esempio



N10 T1 D1 G54 G17 G90 F5000 G94	;Selezione utensile
N20 G0 X20 Z10 SPOS=45	;Accostamento alla posizione di partenza
N30 TRANSMIT	;Attivazione della funzione TRANSMIT
N40 ROT RPL=-45	;Impostazione frame
N50 ATRANS X-2 Y10	
N60 G1 X10 Y-10 G41 OFFN=1	;Sgrossatura del parallelepipedo
N70 X-10	;sovrametallo 1 mm
N80 Y10	
N90 X10	
N100 Y-10	

N110 G0 Z20 G40 OFFN=0	;Cambio dell'utensile
N120 T2 D1 X15 Y-15	
N130 Z10 G41	
N140 G1 X10 Y-10	;Finitura del parallelepipedo
N150 X-10	
N160 Y10	
N170 X10	
N180 Y-10	
N190 Z20 G40	;Disattivazione frame
N200 TRANS	
N210 TRAFOOF	
N220 G0 X20 Z10 SPOS=45	;Accostamento alla posizione di partenza
N230 M30	

Descrizione

Polo

Per l'attraversamento del polo esistono due possibilità:

- movimento del solo asse lineare
- movimento nel polo con rotazione dell'asse rotante e successivo proseguimento oltre il polo

La selezione avviene con i DM 24911 e 24951.

TRANSMIT con asse Y lineare supplementare (tipo di trasformazione 257):

Questa variante di trasformazione della trasformazione polare, in una macchina con un ulteriore asse lineare, sfrutta la ridondanza per eseguire una correzione utensile migliore. Per il secondo asse lineare vale quindi:

- un campo di lavoro inferiore e
- che il secondo asse lineare non può essere utilizzato per arrestare il partprogram.

Per il partprogram e l'assegnazione dei relativi assi nel SCN oppure SCM, sono necessarie determinate impostazioni dei dati macchina, vedere

Bibliografia

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazioni cinematiche (M1)

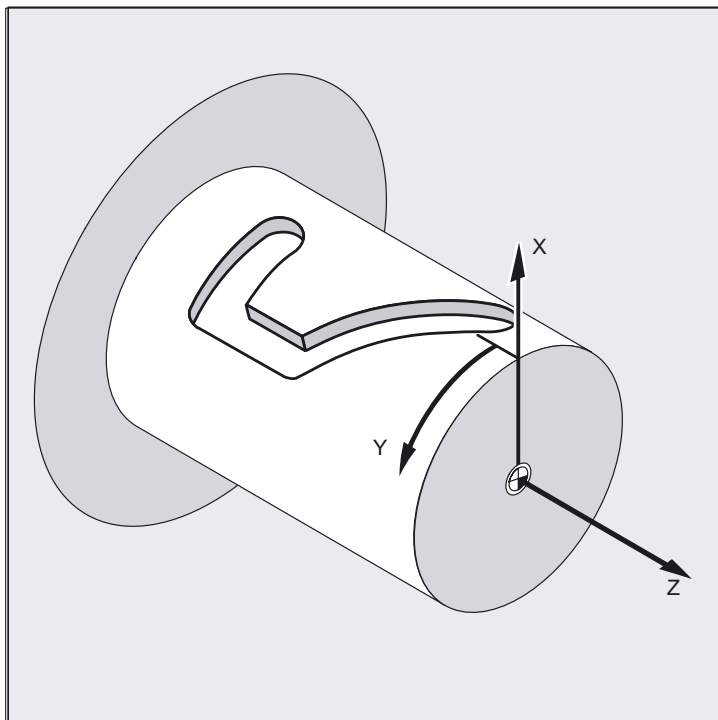
7.8.2 Trasformazione della superficie esterna del cilindro (TRACYL)

Funzione

La trasformazione di curve su una superficie cilindrica TRACYL permette di eseguire:
la lavorazione di

- cave longitudinali su corpi cilindrici,
- cave trasversali su corpi cilindrici,
- cave ad andamento libero su corpi cilindrici.

L'andamento delle cave viene programmato in riferimento alla superficie cilindrica.



Tipi di trasformazione TRACYL

La trasformazione curvilinea su una superficie cilindrica è disponibile in due varianti:

- TRACYL senza correzione della parete della cava: (TRAFO_TYPE_n=512)
- TRACYL con correzione della parete della cava: (TRAFO_TYPE_n=513)
- TRACYL con ulteriore asse lineare e correzione della parete della cava: (TRAFO_TYPE_n=514)

La correzione della parete della cava viene parametrizzata con TRACYL tramite il terzo parametro.

Nella trasformazione curvilinea su una superficie cilindrica con correzione della cava, l'asse utilizzato per la correzione dovrebbe trovarsi a zero ($y=0$) in modo che la cava venga eseguita centrata sulla linea mediana programmata.

Utilizzo dell'asse

I seguenti assi non possono essere utilizzati come assi di posizionamento o di pendolamento:

- l'asse geometrico in direzione periferica della superficie del cilindro (asse Y)
- l'asse lineare supplementare per correzione della cava (asse Z)

Programmazione

TRACYL(d) oppure TRACYL(d, n) oppure

per il tipo di trasformazione 514

TRACYL(d, n, correzione della parete della cava)

oppure

TRAFOOF

Asse rotante

L'asse rotante non può essere programmato in quanto viene occupato da un asse geometrico e quindi non è più programmabile direttamente come asse canale.

Parametri

TRACYL (d)	Attiva la prima delle funzioni TRACYL definite nei dati macchina del canale: il parametro d rappresenta il diametro di lavoro.
TRACYL (d, n)	Attiva la n. funzione TRANSMIT definita nei dati macchina del canale; n può essere al massimo 2, TRACYL(d,1) corrisponde a TRACYL(d)
d	Valore per il diametro di lavorazione. Il diametro di lavorazione rappresenta la distanza doppia tra la punta dell'utensile ed il centro di rotazione. Questo diametro deve essere sempre impostato e deve essere maggiore di 1.
n	2° parametro opzionale per il blocco dati TRACYL 1 (preselezionato) oppure 2.
Correzione parete cava	3° parametro opzionale il cui valore per TRACYL viene preselezionato dal Mode dei dati macchina. Campo di valori: 0: Tipo di trasformazione 514 senza correzione della parete della cava come finora 1: Tipo di trasformazione 514 con correzione della parete della cava
TRAFOOF	Trasformazione off (SCB e SCM sono nuovamente identici).
OFFN	Offset profilo normale: distanza tra la superficie della cava ed il profilo di riferimento programmato

Nota

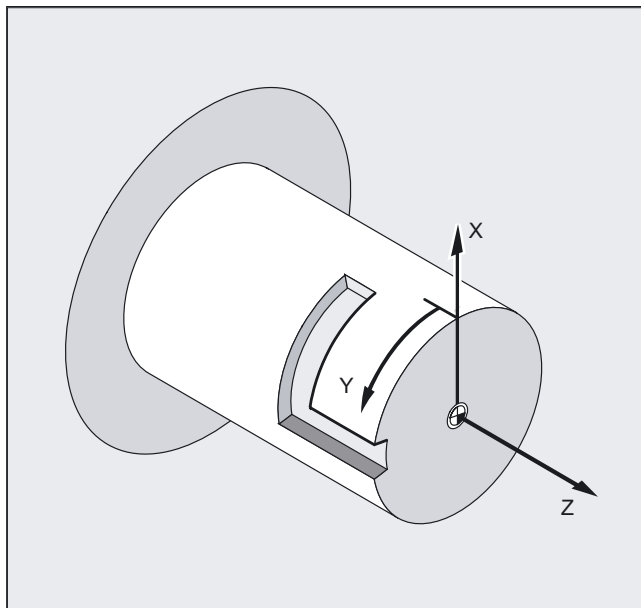
Una trasformazione TRACYL attiva viene anche disattivata quando nel rispettivo canale viene attivata una delle restanti trasformazioni (es. TRANSMIT, TRAANG, TRAORI).

Esempio di definizione dell'utensile

L'esempio seguente è adatto per verificare la parametrizzazione della trasformazione cilindrica TRACYL :

Parametri utensile Numero (DP)	Significato	Note
\$TC_DP1[1,1]=120	Tipo di utensile	Fresa
\$TC_DP2[1,1]=0	Posizione di taglio	Solo per utensili da tornio
Geometria		
Correzione lunghezza		
\$TC_DP3[1,1]=8.	Vettore di correzione longitudinale	Calcolo in base al tipo
\$TC_DP4[1,1]=9.		ed al piano
\$TC_DP5[1,1]=7.		
Geometria		
raggio		
\$TC_DP6[1,1]=6.	raggio	raggio dell'utensile
\$TC_DP7[1,1]=0	Larghezza della cava b per seghe per cave, raggio del raccordo per utensili di fresatura	
\$TC_DP8[1,1]=0	Spoglia k	Solo per sega per cave
\$TC_DP9[1,1]=0		
\$TC_DP10[1,1]=0		
\$TC_DP11[1,1]=0	Angolo per utensili conici di fresatura	
usura		
Correzione lunghezza e raggio		
\$TC_DP12[1,1]=0	I restanti parametri fino a \$TC_DP24=0	Quota base/adattat.

Esempio di esecuzione di una cava con forma ad uncino



Attivare la trasformazione su superficie cilindrica

N10 T1 D1 G54 G90 F5000 G94

;Scelta dell'utensile, compensazione del
;bloccaggio

N20 SPOS=0

;Accostamento alla posizione di partenza

N30 G0 X25 Y0 Z105 CC=200

N40 TRACYL (40)

;Attivazione della trasformazione curvilinea su
;una superficie cilindrica

N50 G19

;Scelta del piano

Esecuzione della cava con forma ad uncino

N60 G1 X20

;Posizionare l'utensile alla base della cava

N70 OFFN=12

;Definizione della distanza della cava di 12 mm
;rispetto alla linea mediana

N80 G1 Z100 G42

;Accostamento sulla parete destra della cava

N90 G1 Z50

;Sezione della cava parallela all'asse del
;cilindro

N100 G1 Y10

;Sezione della cava parallela alla circonferenza

N110 OFFN=4 G42

;Accostamento sulla parete sinistra della cava
;definizione della distanza della cava di 4 mm
;rispetto alla linea mediana

N120 G1 Y70

;Sezione della cava parallela alla circonferenza

N130 G1 Z100

;Sezione della cava parallela all'asse del cilindro

N140 G1 Z105 G40

;Svincolo dalla cava

N150 G1 X25

;Svincolo

N160 TRAFOOF

N170 G0 X25 Y0 Z105 CC=200

;Accostamento alla posizione di partenza

N180 M30

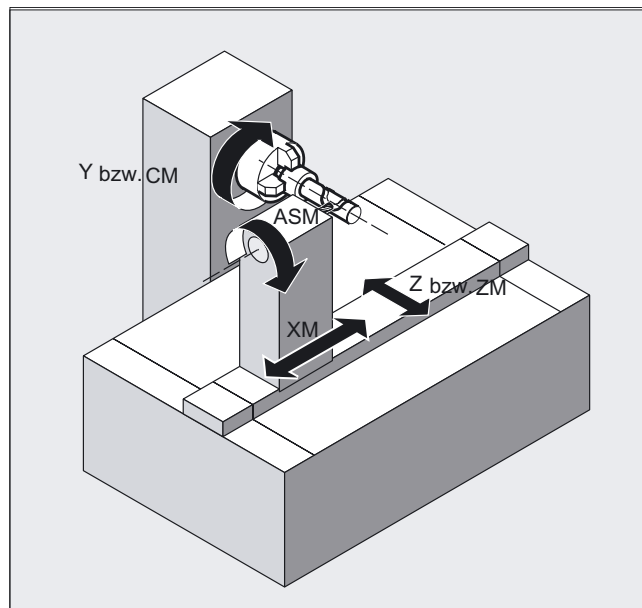
Descrizione

Senza correzione della parete della cava (tipo di trasformazione 512):

il controllo numerico trasforma i movimenti programmati del sistema di coordinate cilindriche nei movimenti degli assi macchina reali:

- Asse rotante
- Asse di incremento perpendicolare all'asse rotante
- Asse longitudinale parallelo all'asse di rotazione

Gli assi lineari sono ortogonali tra di loro. L'asse di incremento interseca l'asse rotante.

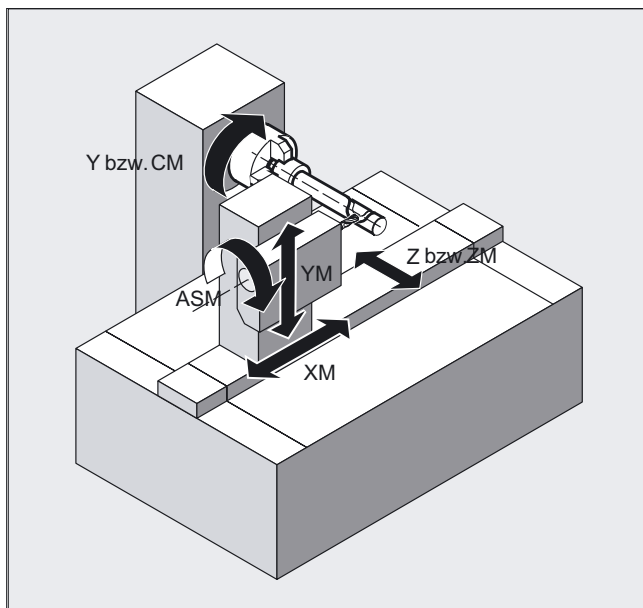


Con correzione della parete della cava (tipo di trasformazione 513):

Per la cinematica vedere l'esempio precedente ma con un ulteriore asse longitudinale parallelo alla direzione della circonferenza

Gli assi lineari sono ortogonali tra di loro.

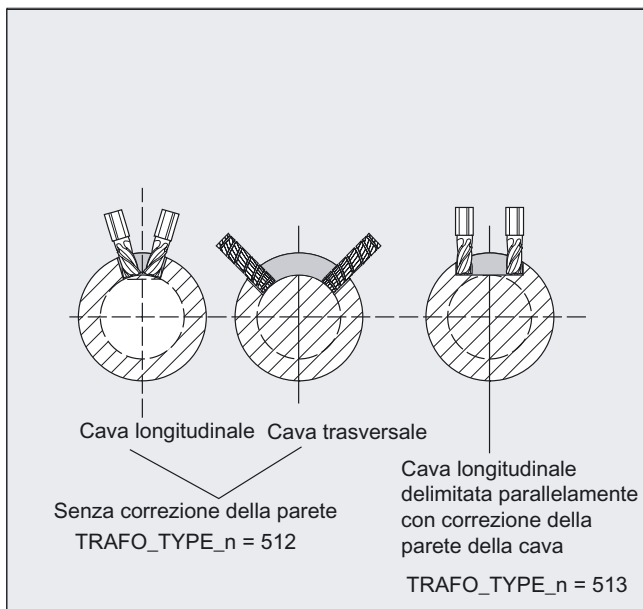
La guida della velocità tiene conto delle limitazioni definite per i movimenti rotatori.



Sezione trasversale della cava

Nella configurazione assiale 1, le cave longitudinali all'asse rotante sono limitate parallelamente solo quando la larghezza delle stesse corrisponde esattamente al raggio dell'utensile.

Le cave parallele alla circonferenza (cave trasversali), all'inizio ed alla fine non sono parallele.



Con ulteriore asse lineare e correzione della parete della cava (tipo di trasformazione 514):

Questa variante di trasformazione, in una macchina con un ulteriore asse lineare, sfrutta la ridondanza per eseguire una correzione utensile migliore. Per il secondo asse lineare vale quindi:

- un campo di lavoro inferiore e
- che il secondo asse lineare non può essere utilizzato per arrestare il partprogram.

Per il partprogram e l'assegnazione dei relativi assi nel SCN oppure SCM, sono necessarie determinate impostazioni dei dati macchina, vedere

Bibliografia

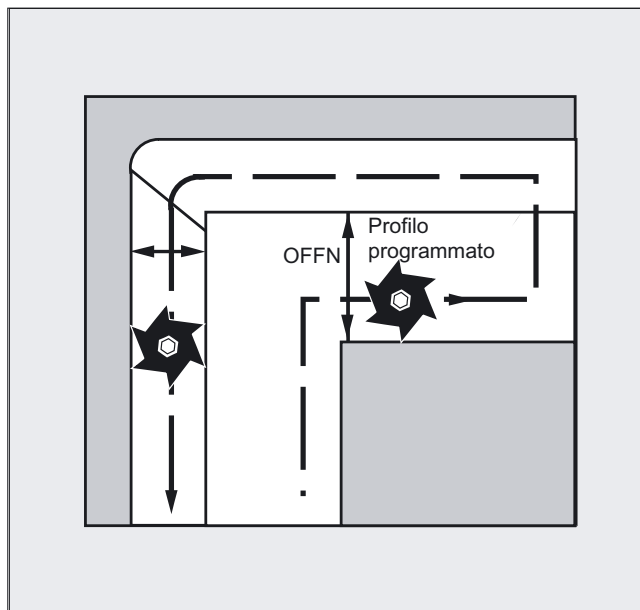
/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazioni cinematiche (M1)

Offset profilo normale OFFN (tipo di trasformazione 513)

Per poter fresare cave con `TRACYL`, viene programmata

- nel partprogram la linea mediana della cava
- con `OFFN` la semi-cava.

`OFFN` diventa attivo solo con correzione raggio utensile attivata per evitare un danneggiamento della parete della cava). In seguito, `OFFN` dovrebbe essere $\geq$ al raggio utensile per escludere un danneggiamento della parete opposta della cava.



Un partprogram per la fresatura di una cava, normalmente è composto dai seguenti passi:

1. selezione dell'utensile
2. attivazione di `TRACYL`
3. selezionare l'opportuna traslazione delle coordinate (FRAME)
4. Posizionamento
5. programmare `OFFN`
6. attivare la CRU
7. blocco di accostamento (esecuzione della CRU e raggiungimento della cava)

8. profilo della linea mediana della cava
9. disattivare la CRU
10. blocco di svincolo (disattivazione della CRU e svincolo dalla cava)
11. Posizionamento
12. TRAFOOF
13. selezionare nuovamente la traslazione delle coordinate originaria (FRAME)

Particolarità

- Attivazione della CRU:

La CRU non viene programmata rispetto alla superficie della cava ma in modo incrementale rispetto alla linea mediana della cava programmata. Per fare in modo che l'utensile avanzi sulla sinistra della cava, viene programmato G42 (invece di G41). Questo viene evitato se in `OFFN` si imposta la larghezza della cava con segno negativo.

- `OFFN` con `TRACYL` agisce in modo diverso da `OFFN` senza `TRACYL`. Siccome `OFFN` con CRU attiva viene considerato anche senza `TRACYL`, dopo `TRAFOOF` si dovrebbe settare nuovamente a zero `OFFN`.
- È consentita una variazione di `OFFN` all'interno del partprogram. In questo modo la linea mediana potrebbe essere spostata rispetto al centro (vedere figura).
- Guide di scorrimento:

Con `TRACYL` non viene realizzata la stessa cava per guide di scorrimento, come se fosse realizzata con un utensile di diametro pari alla larghezza della cava stessa. Fondamentalmente, con un utensile cilindrico più piccolo, non è possibile realizzare la stessa geometria della cava eseguita con un utensile più grande. `TRACYL` minimizza l'errore. Per evitare problemi di precisione, il raggio dell'utensile dovrebbe essere solo leggermente inferiore alla metà della larghezza della cava.

Nota

OFFN e CRU

Con `TRAFO_TYPE_n = 512` il valore di `OFFN` vale come sovrametallo per la CRU.

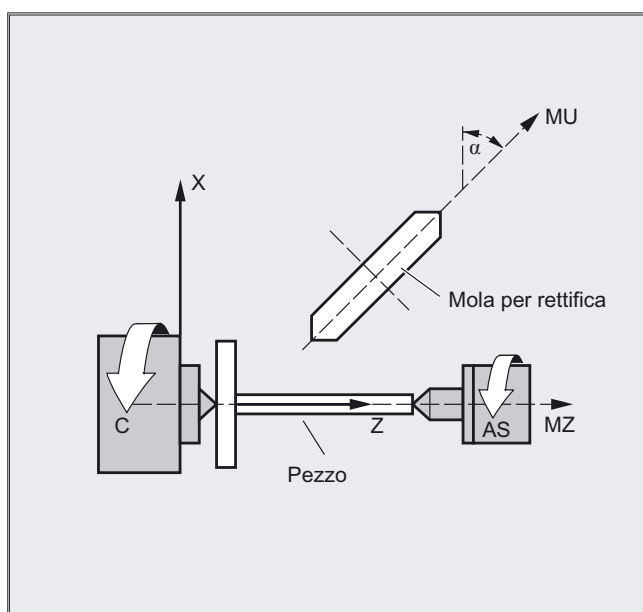
Con `TRAFO_TYPE_n = 513` in `OFFN` viene programmata la metà della larghezza della cava. Lo svincolo dal profilo avviene con `OFFN-CRU`.

7.8.3 Asse inclinato (TRAANG)

Funzione

La funzione asse inclinato è stata realizzata per le rettifiche e consente le seguenti prestazioni:

- lavorazione con asse di incremento inclinato
- per la programmazione si può utilizzare un sistema di coordinate cartesiane.
- Il controllo numerico trasforma i movimenti programmati del sistema di coordinate cartesiano nei movimenti degli assi macchina reali (caso standard): asse di incremento inclinato.



Programmazione

TRAANG(α) oppure TRAANG(α , n)

oppure

TRAFOOF

Parametri

TRAANG () oppure
TRAANG (, n)
TRAANG (α)

Attivazione della trasformazione con la parametrizzazione della selezione precedente.
Attivazione della prima trasformazione per asse inclinato dichiarata.

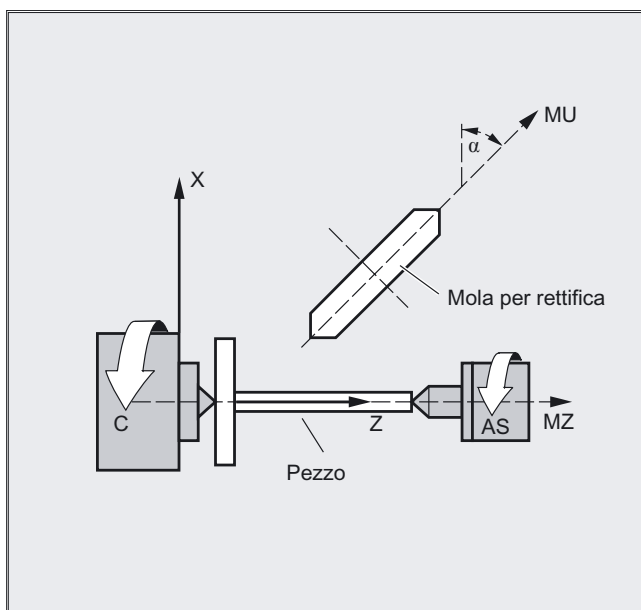
TRAANG (α , n)	Attivazione della n. trasformazione per asse inclinato dichiarata. n può essere al max. 2. TRAANG(α ,1) corrisponde a TRAANG(α).
α	Angolo di inclinazione dell'asse I valori possibili per α sono: -90 gradi < α < + 90 gradi
TRAFOOF	Disattiva la trasformazione
n	Numero di trasformazioni dichiarate

Angolo α tralasciato oppure zero

Se l'angolo α viene tralasciato (ad es. TRAANG () , TRAANG (, n)), la trasformazione viene attivata con la parametrizzazione della selezione precedente. Alla prima attivazione vale la preimpostazione dei dati macchina.

Un angolo $\alpha = 0$ (es. TRAANG (0) , TRAANG (0 , n)) costituisce una parametrizzazione valida e non corrisponde più a tralasciare il parametro nelle versioni precedenti.

Esempio



N10 G0 G90 Z0 MU=10 G54 F5000 ->	;Scelta dell'utensile, compensazione del
-> G18 G64 T1 D1	;bloccaggio
	;Scelta del piano
N20 TRAANG(45)	;Attivazione della trasformazione per asse
	;inclinato
N30 G0 Z10 X5	;Accostamento alla posizione di partenza
N40 WAITP(Z)	;Abilitazione dell'asse per il pendolamento

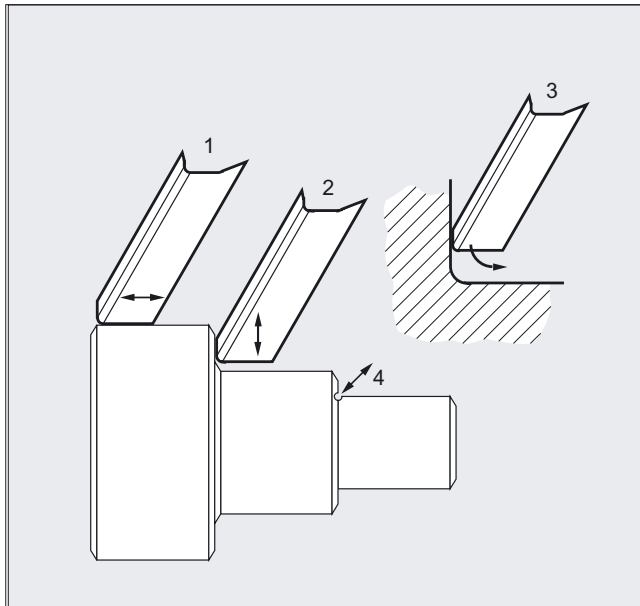
<pre>N50 OSP[Z]=10 OSP2[Z]=5 OST1[Z]=-2 -> -> OST2[Z]=-2 FA[Z]=5000 N60 OS[Z]=1 N70 POS[X]=4.5 FA[X]=50 N80 OS[Z]=0 N90 WAITP(Z) N100 TRAFOOF N110 G0 Z10 MU=10 N120 M30</pre>	<pre>;Quota di pendolamento raggiunta ;(per il pendolamento vedere il capitolo ;"Pendolamento") ;Abilitazione dell'asse di pendolamento ;come asse di posizionamento ;Disattivazione della trasformazione ;Svincolo</pre>

-> da programmare in un blocco a sé stante

Descrizione

Sono possibili le seguenti lavorazioni:

1. rettifica longitudinale
2. rettifica in piano
3. rettifica di un determinato profilo
4. Rettifica con mola obliqua



Costruttore della macchina

Tramite dati macchina vengono eseguite le seguenti impostazioni:

- l'angolo tra un asse macchina e l'asse inclinato,
- la posizione del punto zero utensile riferito all'origine del sistema di coordinate dichiarato per la funzione "Asse inclinato",

- la riserva di velocità che viene mantenuta sull'asse parallelo per il movimento di compensazione,
- la riserva di accelerazione che viene mantenuta sull'asse parallelo per il movimento di compensazione.

Configurazione dell'asse

Per poter programmare nel sistema di coordinate cartesiano, bisogna segnalare al controllo la relazione esistente tra il sistema di coordinate e gli assi macchina (MU, MZ) realmente esistenti:

- Denominazione degli assi geometrici
- Abbinamento degli assi geometrici agli assi canale
 - caso generico (asse inclinato non attivo)
 - asse inclinato attivo
- Abbinamento degli assi canale al numero di asse macchina
- Codifica dei mandrini
- Assegnazione dei nomi degli assi macchina

Il procedimento, con eccezione di "Asse inclinato attivo", corrisponde a quello della normale configurazione degli assi.

7.8.4 Programmazione dell'asse inclinato (G05, G07)

Funzione

Nel funzionamento Jog, la mola può essere mossa in modo cartesiano oppure in direzione dell'asse inclinato (la visualizzazione resta cartesiana). Si muove solo il primo asse reale U, la visualizzazione dell'asse Z viene aggiornata.

Nel funzionamento Jog, le traslazioni REPOS devono essere eseguite in modo cartesiano.

Nel funzionamento Jog con

"Movimento PTP" attivo viene sorvegliato il superamento del limite del campo di lavoro cartesiano e il relativo asse viene frenato in anticipo. Se il "Movimento PTP" non è attivo, l'asse può raggiungere esattamente il limite di lavoro.

Bibliografia

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazione cinematica (M1)

Programmazione

G07

G05

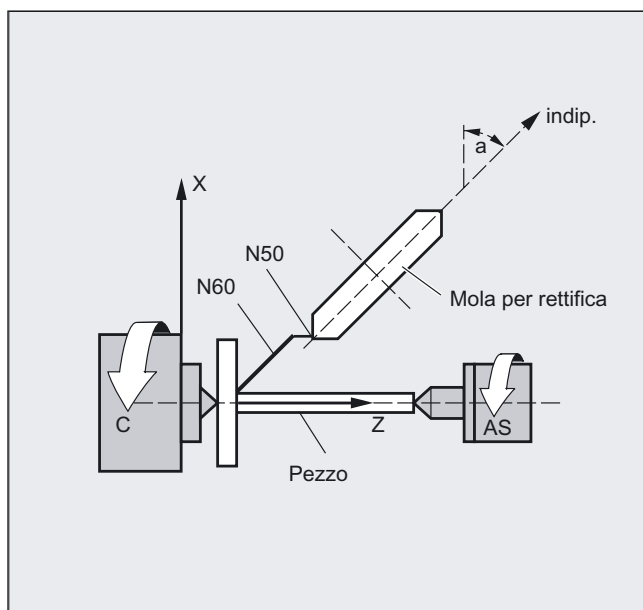
Le istruzioni G07/G05 servono per semplificare la programmazione dell'asse inclinato. In questo modo si possono programmare e visualizzare posizioni nel sistema di coordinate cartesiano. La correzione utensile e lo spostamento origine vengono calcolate come cartesiane. Dopo aver programmato l'angolo per l'asse inclinato nel programma NC, si può raggiungere la posizione iniziale (G07) e in seguito eseguire completamente la troncatura inclinata (G05).

Parametri

G07
G05

Accostamento alla posizione iniziale
Attivazione troncatura con mola inclinata

Esempio



N.. G18

;Programmazione dell'angolo per l'asse
;inclinato

N50 G07 X70 Z40 F4000

;Accostamento alla posizione iniziale

N60 G05 X70 F100

;Troncatura inclinata

N70 ...

7.9 Movimento cartesiano PTP

Funzione

Con questa funzione è possibile programmare una posizione in un sistema di coordinate cartesiano, il movimento della macchina avviene però nel sistema di coordinate di macchina. La funzione può essere applicata ad esempio nel cambio di posizione dell'articolazione se il movimento passa attraverso una singolarità.

Nota

La funzione può essere applicata solo in collegamento ad una trasformazione attiva. Inoltre il "movimento PTP" è ammesso solo in collegamento con G0 e G1.

Programmazione

```
N... TRAORI
N... STAT='B10' TU='B100' PTP
N... CP
```

Movimento PTP durante la trasformazione generica a 5/6 assi

Se durante la trasformazione generica a 5/6 assi con PTP viene attivato un movimento punto a punto nel sistema di coordinate macchina (ORIMKS), è possibile programmare l'orientamento dell'utensile tanto con posizioni di asse rotante

```
N... G1 X Y Z A B C
```

quanto con vettori angolo di Eulero o RPY indipendenti dalla cinematica

```
N... ORIEULER oppure ORIRPY
```

```
N... G1 X Y Z A2 B2 C2
```

o con i vettori direzionali

```
N... G1 X Y Z A3 B3 C3
```

. Può essere attiva l'interpolazione dell'asse rotante, l'interpolazione vettoriale con interpolazione circolare ad ampio raggio ORIVECT o l'interpolazione del vettore di orientamento su una superficie conica ORICONxx.

Ambiguità dell'orientamento con i vettori

La programmazione dell'orientamento con vettori dà luogo a un'ambiguità nelle possibili posizioni di asse rotante. Le posizioni di asse rotante da raggiungere possono essere selezionate con la programmazione di STAT = <...>. Se

viene programmato STAT = 0 (corrisponde all'impostazione standard), vengono accostate le posizioni che distano meno dalle posizioni iniziali. Se

viene programmato STAT = 1, vengono accostate le posizioni che distano maggiormente dalle posizioni iniziali.

Parametri

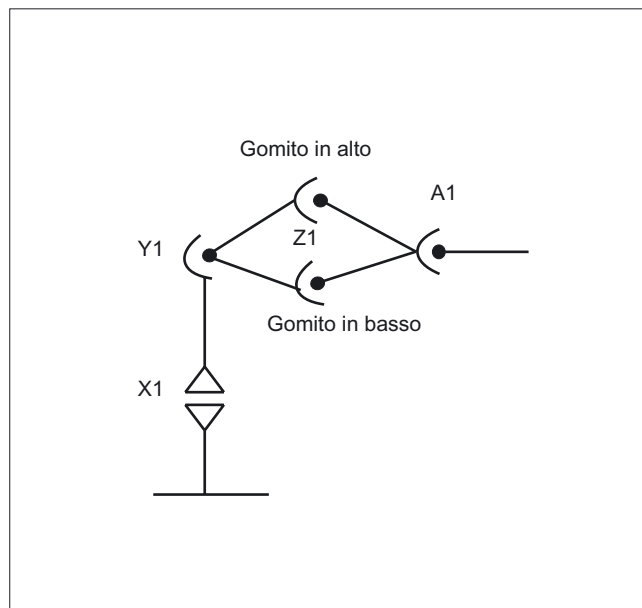
I comandi **PTP** e **CP** hanno efficacia modale. **CP** costituisce l'impostazione standard.

Mentre la programmazione del valore **STAT** ha validità modale, la programmazione di **TU = <...>** agisce blocco per blocco.

Un'altra differenza è che la programmazione di un valore **STAT** ha effetto solo in presenza di interpolazione vettoriale, mentre la programmazione di **TU** viene valutata anche con l'interpolazione dell'asse rotante attiva.

PTP	Point to Point; (movimento da punto a punto) Il movimento viene eseguito come movimento di assi sincroni; l'asse più lento interessato al movimento è l'asse dominante per la velocità.
CP	continuous path (movimento vettoriale) Il movimento viene eseguito come movimento vettoriale cartesiano.
STAT=	Posizione dello snodo; il valore è in funzione della trasformazione.
TU=	l'informazione TURN è valida solo nel blocco. In questo modo è possibile raggiungere univocamente l'angolo dell'asse tra -360 gradi e +360 gradi.

Esempio



N10 G0 X0 Y-30 Z60 A-30 F10000	;Posizione di partenza → Gomito in alto
N20 TRAORI(1)	;trasformazione ON
N30 X1000 Y0 Z400 A0	
N40 X1000 Z500 A0 STAT='B10' TU='B100' PTP	;Nuovo orientamento senza trasformazione → Gomito in basso
N50 X1200 Z400 CP	;Trasformazione nuovamente attiva

```
N60 X1000 Z500 A20
N70 M30
```

Movimento PTP durante la trasformazione generica a 5 assi

Presupposti: l'esempio si basa su una cinematica CA ortogonale.

TRAORI	;trasformazione cinematica CA On
PTP	;attivazione movimento PTP
N10 A3 = 0 B3 = 0 C3 = 1	;posizioni di asse rotante C = 0 A = 0
N20 A3 = 1 B3 = 0 C3 = 1	;posizioni di asse rotante C = 90 A = 45
N30 A3 = 1 B3 = 0 C3 = 0	;posizioni di asse rotante C = 90 A = 90
N40 A3 = 1 B3 = 0 C3 = 1 STAT = 1	;posizioni di asse rotante C = 270 A = -45

Selezione di una posizione di accostamento univoca degli assi rotanti:

Nel blocco N40 gli assi rotanti coprono, con la programmazione di `STAT = 1`, il percorso più lungo dal loro punto di partenza (C=90, A=90) al punto finale (C=270, A=-45), diversamente da come sarebbe con `STAT = 0` ossia il percorso più breve verso il punto finale (C=90, A=45).

Descrizione

La commutazione tra il movimento cartesiano e il movimento degli assi di macchina avviene con le istruzioni `PTP` e `CP`.

Movimento PTP durante la trasformazione generica a 5/6 assi

Durante il movimento PTP il TCP non resta generalmente stazionario, al contrario della trasformazione a 5/6 assi, nel caso in cui si modifichi solo l'orientamento. Le posizioni finali trasformate di tutti gli assi di trasformazione (3 assi lineari e fino a 3 assi rotanti) vengono accostate linearmente senza che, durante tale fase, la trasformazione sia ancora propriamente attiva.

Il movimento PTP viene disattivato dalla programmazione del codice G modale `CP`.

Le varie trasformazioni sono descritte nella documentazione:

/FB3/ Manuale delle funzioni speciali; Pacchetto di trasformazione Handling (TE4).

Programmazione della posizione (STAT=)

Una posizione di macchina non è determinata in modo univoco solo dall'indicazione della posizione con coordinate cartesiane e dall'orientamento dell'utensile. A seconda di quale cinematica si tratta, esistono fino a 8 posizioni diverse o differenzianti dell'articolazione. Tali posizioni sono quindi specifiche della trasformazione. Per poter convertire in modo univoco una posizione cartesiana nell'angolo dell'asse, è necessario che la posizione dell'articolazione venga indicata con il comando `STAT=`. Il comando "STAT" contiene un bit come valore binario per ognuna delle posizioni possibili.

Per i bit delle posizioni da programmare con "STAT", vedere:

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazione cinematica (M1), capitolo "Movimento cartesiano PTP".

Programmazione dell'angolo dell'asse (TU=)

Per poter raggiungere in modo univoco l'angolo dell'asse $< \pm 360$ gradi, è necessario che questa informazione venga programmata con l'istruzione `TU=`.

Gli assi si muovono lungo il percorso più breve:

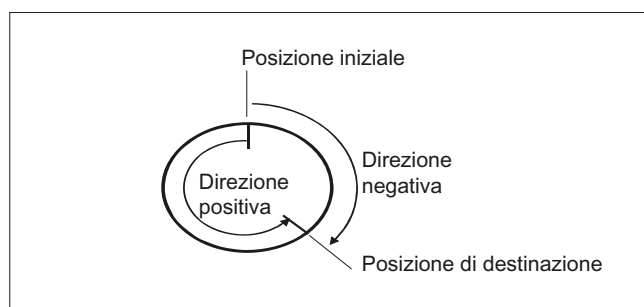
- se per una posizione non viene programmato alcun TU
- se gli assi hanno un campo di posizionamento $> \pm 360$ gradi

Esempio:

La posizione di destinazione indicata nella figura può essere raggiunta in direzione negativa o positiva. Sotto l'indirizzo A1 viene programmata la direzione.

A1=225°, TU=bit 0, → direzione positiva

A1=-135°, TU=Bit 1, → direzione negativa



Esempio di valutazione della variabile TU per la trasformazione generica a 5/6 assi e le posizioni di destinazione

Per ogni asse interessato alla trasformazione, la variabile TU contiene un bit che indica la direzione del movimento. L'assegnazione del bit TU corrisponde al livello asse canale degli assi rotanti. L'informazione TU viene valutata solo per gli assi rotanti possibili (fino a 3) interessati alla trasformazione:

Bit0: asse 1, bit TU = 0 : 0 gradi $\leq$ angolo asse rotante < 360 gradi

Bit1: asse 2, bit TU = 1: -360 gradi $<$ angolo asse rotante < 0 gradi

La posizione iniziale di un asse rotante è C = 0; tramite la programmazione di C = 270, l'asse rotante si sposta sulla seguente posizione di destinazione:

C = 270: bit TU 0, senso di rotazione positivo

C = -90: bit TU 1, senso di rotazione negativo

Ulteriore comportamento

Cambio del modo operativo

La funzione "Movimento PTP cartesiano" è applicabile solo nei modi operativi AUTO e MDA. Nel cambio del modo operativo in JOG viene mantenuta l'impostazione corrente.

Se è impostato il codice G_{PTP}, gli assi si posizionano nel sistema di coordinate di macchina. Se è impostato il codice G_{CP}, gli assi si posizionano nel sistema di coordinate del pezzo.

Power On / RESET

Dopo Power On o dopo RESET l'impostazione dipende dal dato macchina

\$MC_GCODE_REST_VALUES [48]. L'impostazione standard è il tipo di movimento "CP".

REPOS

Se durante il blocco di interruzione era impostata la funzione "Movimento PTP cartesiano", anche con `PTP` viene eseguito il riposizionamento.

Movimenti sovrapposti

La traslazione DRF o lo spostamento origine esterno sono supportati solo parzialmente nel posizionamento PTP cartesiano. Al passaggio da un movimento PTP ad uno CP non devono verificarsi sovrapposizioni nel sistema di coordinate di base.

Movimento raccordato tra CP e PTP

Con `G641` si può eseguire un raccordo programmabile tra i blocchi.

La grandezza dell'area di raccordo è il percorso in mm o in pollici, a partire dal quale o fino al quale deve avvenire il passaggio di blocco. La grandezza deve essere indicata come segue:

- per blocchi G0 con `ADISPOS`
- per tutti gli altri comandi di movimento con `ADIS`

Il calcolo del percorso equivale a prendere in considerazione gli indirizzi F nei blocchi non G0. L'avanzamento viene mantenuto sugli assi specificati in `FGROUP ( . . . )`.

Calcolo dell'avanzamento

Per il calcolo dei blocchi CP si utilizzano gli assi cartesiani del sistema di coordinate base.

Per i blocchi PTP si utilizzano gli assi corrispondenti del sistema di coordinate di macchina.

7.9.1 PTP con TRANSMIT

Funzione

Con PTP in TRANSMIT si possono eseguire blocchi G0 e G1 ottimizzati come tempo. Invece di muovere linearmente gli assi del sistema di coordinate base (CP) vengono mossi linearmente gli assi macchina (PTP). In questo modo il comportamento degli assi macchina nei pressi del polo consente di raggiungere più rapidamente il punto di fine blocco.

Il partprogram viene comunque scritto nel sistema di coordinate cartesiano del pezzo e restano valide tutte le traslazioni di coordinate, rotazioni e programmazioni di frame. La rappresentazione della simulazione su HMI avviene comunque nel sistema di coordinate pezzo cartesiano.

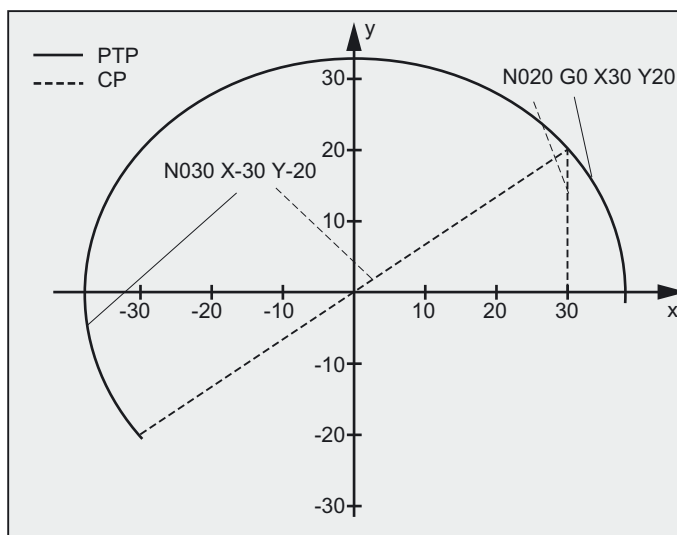
Programmazione

```
N... TRANSMIT
N... PTPG0
N... G0 ...
...
N... G1 ...
```

Parametri

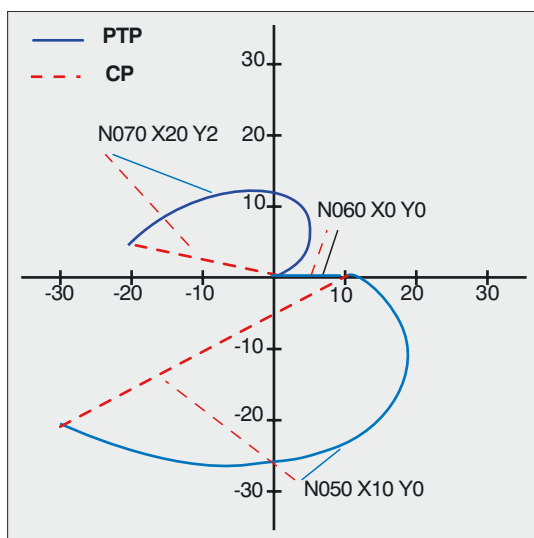
TRANSMIT	Attiva la prima funzione TRANSMIT definita (vedere capitolo "Fresature di pezzi in rotazione: TRANSMIT")
PTPG0	Point to Point G0 (movimento punto-punto automatico per ogni blocco in G0 dopodiché nuovamente CP) Siccome STAT e TU sono modali, resta sempre valido l'ultimo valore programmato.
PTP	Point to Point; (movimento da punto a punto) Nel caso di TRANSMIT programmare in PTP significa che lo svincolo avviene in cartesiano con spirali di Archimede intorno al polo o dal polo. I movimenti risultanti degli utensili in questo caso sono decisamente diversi da quelli in CP e vengono rappresentati nei relativi esempi di programmazione.
STAT=	Risoluzione dell'ambiguità relativa al polo.
TU=	TU nel caso di PTP con TRANSMIT non è rilevante

Esempio di aggiramento del polo con PTP e TRANSMIT



N001 G0 X30 Z0 F10000 T1 D1 G90	;Misura assoluta posizione di partenza
N002 SPOS=0	
N003 TRANSMIT	;Trasformazione TRANSMIT
N010 PTPG0	;PTP automatico ad ogni blocco G0 ;dopodiché nuovamente CP
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Esempio di svincolo dal polo con PTP e TRANSMIT



```

N001 G0 X90 Z0 F10000 T1 D1 G90
N002 SPOS=0
N003 TRANSMIT
N010 PTPG0

N020 G0 X90 Y60
N030 X-90 Y-60
N040 X-30 Y-20
N050 X10 Y0
N060 X0 Y0
N070 X-20 Y2
N170 G1 X0 Y0
N160 X10 Y0
N150 X-30 Y-20
M30
    
```

;Posizione di partenza

;Trasformazione TRANSMIT

;PTP automatico ad ogni blocco G0
;dopodichè nuovamente CP

Descrizione

PTP e PTPG0

PTPG0 viene sorvegliato in tutte le trasformazioni che possono gestire il PTP. In tutti gli altri casi PTPG0 non è rilevante.

I blocchi G0 vengono eseguiti in modalità CP.

La scelta di PTP oppure PTPG0 avviene nel partprogram oppure disattivando CP nel dato macchina \$MC_GCODE_RESET_VALUES[48].

**Cautela****Condizioni marginali**

Nel caso dei movimenti degli utensili e delle collisioni valgono diverse condizioni marginali e determinate esclusioni di funzioni come:

con PTP non può essere attivata la correzione raggio utensile (CRU);

con PTPG0 ed una correzione raggio utensile attiva (CRU), il movimento avviene in CP ;

con PTP non è possibile un accostamento/svincolo dolce dal profilo (WAB);

con PTPG0 ed un accostamento/svincolo dal profilo, il movimento avviene in CP ;

con PTP non sono possibili cicli di sgrossatura (CONTPRON, CONTDCON);

con PTPG0 nei cicli di sgrossatura (CONTPRON, CONTDCON) il movimento avviene in CP ;

smussi (CHF, CHR) e raccordi (RND, RNDM) vengono ignorati;

il compressore non è consentito con PTP e viene automaticamente disattivato nei blocchi con PTP;

non è possibile modificare una sovrapposizione degli assi durante sezioni di programma in PTP;

con G643 avviene automaticamente una commutazione in "Movimento continuo con precisione assiale" G642 ;

con PTP attivo gli assi della trasformazione non possono essere contemporaneamente assi di posizionamento.

Bibliografia:

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazione cinematica (M1),
capitolo "Movimento cartesiano PTP"

PTP con TRACON:

PTP può essere utilizzato anche con TRACON se la prima trasformazione concatenata supporta il PTP .

Significato di STAT= e TU= con TRANSMIT

Se l'asse rotante deve ruotare di 180 gradi oppure il profilo con CP passa attraverso il polo, gli stessi possono essere comandati in senso orario o antiorario di +/- 180 gradi in funzione del dato macchina \$MC_TRANSMIT_POLE_SIDE_FIX_1/2 [48] . Allo stesso modo si può impostare se il movimento avviene attraverso il polo oppure intorno al polo.

7.10 Condizioni marginali per l'attivazione di una trasformazione

Funzione

La selezione delle trasformazioni è possibile tramite partprogram oppure MDA. Attenzione:

- non venga inserito un blocco di movimento intermedio (smussi/raccordi).
- Una sequenza di blocchi Spline sia conclusa; in caso contrario viene emessa una segnalazione.
- La correzione utensile fine sia disattivata (FTOCOF); in caso contrario viene emessa una segnalazione.
- La correzione raggio utensile sia disattivata (G40); in caso contrario viene emessa una segnalazione.
- un correzione della lunghezza utensile viene assunta dal controllo nella trasformazione.
- Il Frame attivo prima della trasformazione viene disattivato dal controllo.
- una limitazione del campo di lavoro per gli assi coinvolti nella trasformazione, viene disattivata dal controllo (corrisponde a WALIMOF).
- la sorveglianza del settore di protezione viene disattivata.
- il funzionamento continuo e la raccordatura vengono interrotti.
- tutti gli assi impostati nel dato macchina, devono essere sincronizzati riferiti al blocco.
- gli assi scambiati devono essere riportati nella condizione originale; in caso contrario viene emessa una segnalazione.
- per gli assi dipendenti viene emessa una segnalazione.

cambio dell'utensile

Il cambio utensile è possibile solo con la correzione raggio utensile disattivata.

Una commutazione della correzione lunghezza utensile e una attivazione/disattivazione della correzione raggio utensile non possono essere programmati nello stesso blocco.

Cambio di frame

Sono consentite tutte le istruzioni che si riferiscono solo al sistema di coordinate base, (FRAME, correzione raggio utensile). Una commutazione del Frame con G91 (quote incrementali), a differenza del caso con trasformazione non attiva, non viene gestito in modo particolare. L'incremento da eseguire viene considerato nel sistema di coordinate pezzo del nuovo Frame, indipendentemente da quale Frame era attivo nel blocco precedente.

Esclusioni

Gli assi coinvolti nella trasformazione non possono essere utilizzati:

- come assi di preset (allarme),
- per il movimento su riscontro fisso (allarme),
- per la ricerca del punto di riferimento (allarme),

7.11 Disattivazione della trasformazione (TRAFOOF)

Funzione

Con l'istruzione TRAFOOF vengono disattivate tutte le trasformazioni e tutti i frame attivi.

Nota

I Frame necessari successivamente, devono essere riattivati con una nuova programmazione.

Attenzione:

per la disattivazione della trasformazione valgono le stesse condizioni marginali valide per l'attivazione (vedere il capitolo ("Condizioni marginali per l'attivazione di una trasformazione"))

Programmazione

TRAFOOF

Parametri

TRAFOOF	Disattivazione di tutte le trasformazioni/Frame

7.12 Trasformazioni concatenate (TRACON, TRAFOOF)

Funzione

Si possono attivare fino a **due** trasformazioni in successione (concatenate) cosicché le quote di movimento per gli assi della prima trasformazione diventano i dati di ingresso per la seconda trasformazione concatenata. Le quote di movimento della seconda trasformazione hanno effetto sugli assi macchina.

Il concatenamento può comprendere fino a **due** trasformazioni.

Nota

Un utensile viene sempre abbinato alla prima trasformazione di un concatenamento. La trasformazione successiva si comporta quindi come se la lunghezza dell'utensile fosse uguale a zero. Per la prima trasformazione di un concatenamento sono attive solo le lunghezze base di un utensile impostate nei dati macchina (BASE_TOOL_).

Costruttore della macchina

Osservare le indicazioni del costruttore della macchina, per le eventuali trasformazioni predefinite nei dati macchina.

Le trasformazioni e le trasformazioni concatenate sono opzioni. Sulla disponibilità di determinate trasformazioni concatenate in certi controlli, osservare le indicazioni del catalogo attuale.

Applicazioni

- Rettifica di profili per i quali come linea superficiale è stato programmato uno sviluppo del cilindro (TRACYL) con una mola inclinata, es. rettifica di utensili.
- Lavorazione fine di un profilo non tondo con mola inclinata realizzato con TRANSMIT.

Programmazione

TRACON(trf, par)
TRAFOOF

Attivazione di una trasformazione concatenata.

Parametri

TRACON	Attivazione della trasformazione concatenata. Una trasformazione precedentemente attiva, con TRACON() viene implicitamente disattivata.
TRAFOOF	L'ultima trasformazione attivata (concatenata) viene disattivata.
trf	Numero della trasformazione concatenata: 0 oppure 1 per la prima/unica trasformazione concatenata. Se in questa posizione non viene programmato nulla, si ha lo stesso significato dell'introduzione del valore 0 oppure 1, cioè viene attivata la prima/unica trasformazione concatenata. 2 per la seconda trasformazione concatenata (valori diversi da 0 ... 2 provocano una segnalazione di allarme).
par	Uno o più parametri separati da una virgola per le trasformazioni concatenate, che si aspettano un parametro, es. angolo dell'asse inclinato. Se non viene settato alcun parametro, vengono utilizzate le preimpostazioni oppure gli ultimi parametri utilizzati. Utilizzando le virgole, è necessario prestare attenzione che i parametri indicati vengano interpretati nella sequenza impostata, quando per i parametri precedenti devono essere valide le preimpostazioni. In particolare, prima della dichiarazione deve essere inserito almeno un parametro anche quando la dichiarazione di trf non è necessaria, ad esempio TRACON(, 3,.7).

Premessa

La **seconda** trasformazione deve essere del tipo **"Asse inclinato"** (TRAANG). Come prima trasformazione sono possibili:

- Trasformazioni di orientamento (TRAORI), compresa la testina di fresatura cardanica
- TRANSMIT
- TRACYL
- TRAANG

Per l'utilizzo delle istruzioni di attivazione di una trasformazione concatenata, è necessario che le singole trasformazioni da concatenare e la trasformazione concatenata da attivare, siano definite tramite dati macchina.

Le condizioni marginali ed i casi speciali riportati nelle singole descrizioni delle trasformazioni, devono essere rispettate anche nel caso di utilizzo in un concatenamento.

Per ulteriori informazioni sulla progettazione dei dati macchina delle trasformazioni, consultare:

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Trasformazioni cinematiche (M1), e

/FB3/ Manuale delle funzioni speciali; Trasformazioni a 3 ... 5 assi (F2).

7.13 Assi geometrici commutabili (GEOAX)

Funzione

Con la funzione "Assi geometrici commutabili", il raggruppamento di assi geometrici configurato tramite dati macchina può essere modificato da partprogram. Un asse canale definito come asse supplementare sincrono può sostituire qualsiasi asse geometrico.

Programmazione

GEOAX(n,asse canale,n, asse canale,...)

oppure

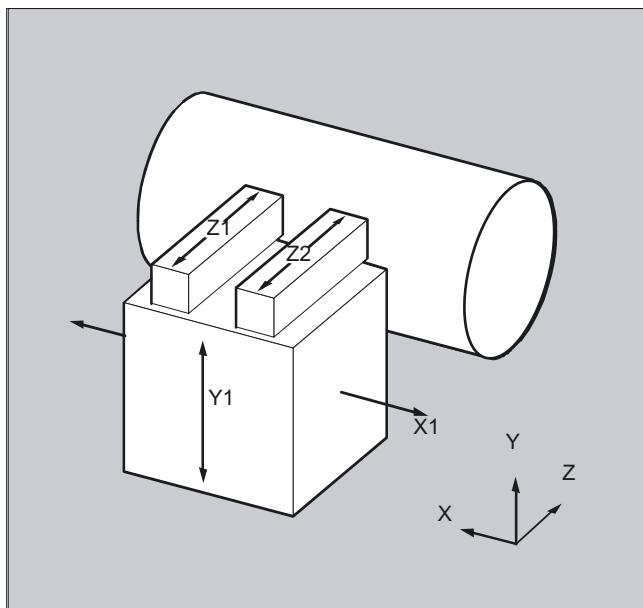
GEOAX ()

Parametri

GEOAX(n,asse canale,n, asse canale,...)	Commutazione degli assi geometrici.
GEOAX ()	Richiamare la configurazione di base degli assi geometrici
n	Numero dell'asse geometrico (n=1, 2 oppure 3), che deve essere abbinato ad un altro asse canale. n=0: Asse canale che deve essere rimosso, senza essere sostituito, dal raggruppamento di assi geometrici.
Asse canale	Nome dell'asse canale che deve essere inserito nel raggruppamento di assi geometrici.

Esempio di come commutare alternativamente due assi di geometria

Una slitta utensile può essere mossa tramite gli assi canali X1, Y1, Z1, Z2. Nel partprogram devono essere impiegati gli assi Z1 e Z2 come sostituzione dell'asse geometrico Z. La commutazione tra gli assi nel partprogram avviene con GEOAX.



Dopo l'inserzione è attivo il collegamento X1, Y1, Z1 (impostabile tramite DM).

N100 GEOAX (3,Z2)	;L'asse canale Z2 funge da asse Z
N110 G1	
N120 GEOAX (3,Z1)	;L'asse canale Z1 funge da asse Z

Esempio di configurazione di assi di geometria per 6 assi canale

Una macchina possiede 6 assi canale con i nomi XX, YY, ZZ, U, V, W. La configurazione di base degli assi geometrici nei dati macchina è la seguente:

asse canale XX = 1° asse geometrico (asse X)

asse canale YY = 2° asse geometrico (asse Y)

asse canale ZZ = 3° asse geometrico (asse Z)

N10 GEOAX()	;È attiva la configurazione di base degli assi geometrici.
N20 G0 X0 Y0 Z0 U0 V0 W0	;Tutti gli assi in rapido alla posizione 0.
N30 GEOAX(1,U,2,V,3,W)	;L'asse canale U diventa il primo (X), V il secondo (Y) e W il terzo asse geometrico (Z).
N40 GEOAX(1,XX,3,ZZ)	;L'asse canale XX diventa il primo (X), ZZ il terzo asse geometrico (Z). ;L'asse canale V resta il secondo asse geometrico (Y).
N50 G17 G2 X20 I10 F1000	;Cerchio completo nel piano X – Y. Si muovono gli assi canale XX e V
N60 GEOAX(2,W)	;L'asse canale W diventa il secondo asse geometrico (Y).
N80 G17 G2 X20 I10 F1000	;Cerchio completo nel piano X – Y. Si muovono gli assi canale XX e W
N90 GEOAX()	;Ripristino allo stato iniziale
N100 GEOAX(1,U,2,V,3,W)	;L'asse canale U diventa il primo (X), V il secondo (Y) e W il terzo asse geometrico (Z).

N110 G1 X10 Y10 Z10 XX=25	;Gli assi canale U, V, Z muovono alla posizione 10, XX come asse ;supplementare muove alla posizione 25.
N120 GEOAX(0,V)	;V viene eliminato dal raggruppamento di assi geometrici. U e W restano ;il primo (X) ed il terzo asse geometrico (Z). Il secondo asse geometrico ;(Y) resta libero.
N130 GEOAX(1,U,2,V,3,W)	;L'asse canale U diventa il primo (X), V il secondo (Y) e W il terzo asse ;geometrico (Z).
N140 GEOAX(3,V)	;V diventa il terzo asse geometrico (Z) mentre W viene sovrascritto e ;viene quindi cancellato dal raggruppamento di assi geometrici. Il ;secondo asse geometrico (Y) resta libero.

Premesse e limitazioni

1. La commutazione degli assi geometrici non è possibile con:
 - trasformazione attiva,
 - interpolazione Spline attiva,
 - correzione raggio utensile attiva (vedere manuale di programmazione al capitolo "Correzioni utensili")
 - correzione utensile fine attiva (vedere manuale di programmazione al capitolo "Correzioni utensili")
2. Se l'asse geometrico e l'asse canale fanno riferimento allo stesso nome, non è possibile la commutazione del rispettivo asse geometrico.
3. Nessuno degli assi coinvolti nella commutazione può essere interessato in un'azione che dura oltre il cambio blocco, come ad esempio per assi di posizionamento del tipo A oppure assi a seguire.
4. Con l'istruzione `GEOAX` si possono sostituire solo assi geometrici già esistenti prima dell'attivazione (quindi non possono essere definiti nuovi assi).
5. Uno scambio assi con `GEOAX` durante la preparazione della **tabella del profilo** (`CONTPRON`, `CONTDCON`) provoca l'emissione di un allarme.

Descrizione

Il numero di asse geometrico

Nell'istruzione `GEOAX(n,asse canale,...)` il numero `n` identifica l'asse geometrico al quale deve essere abbinato il successivo asse canale impostato.

Per la commutazione di un asse canale sono consentiti i numeri di assi geometrici da 1 a 3 (asse X, Y, Z).

Con `n = 0` viene rimosso un asse canale dal raggruppamento di assi geometrici senza essere rimpiazzato.

Un asse sostituito nel raggruppamento di assi geometrici durante una commutazione, dopo la procedura di commutazione può essere programmato come asse supplementare con il suo nome di asse canale.

Con la commutazione degli assi geometrici vengono cancellati tutti i frame, i settori di protezione e le limitazioni dei campi di lavoro.

Coordinate polari

Uno scambio degli assi geometrici con `GEOAX` setta, analogamente ad un cambio di piano (G17-G19), le coordinate polari al valore 0.

DRF, NPV

Una eventuale traslazione volante (DRF) oppure uno spostamento origine esterno restano attivi anche dopo la commutazione.

Scambio delle posizioni degli assi

Con una riassegnazione dei numeri degli assi agli assi canali già esistenti, è possibile anche uno scambio delle posizioni nell'ambito del raggruppamento geometrico.

N... `GEOAX (1, XX, 2, YY, 3, ZZ)`

;L'asse canale XX è il primo, YY il secondo

N... `GEOAX (1, U, 2, V, 3, W)`

;e ZZ il terzo asse geometrico,

;l'asse canale U è il primo, V il secondo

;e W il terzo asse geometrico.

Disattivazione della commutazione

L'istruzione `GEOAX()` richiama la configurazione base del raggruppamento di assi geometrici.

Dopo POWER ON e con la commutazione nel modo operativo "Ricerca del punto di riferimento", viene riattivata automaticamente la configurazione di base.

Procedura di commutazione e correzione lunghezza utensile

Una correzione lunghezza utensile attiva resta tale anche dopo la procedura di commutazione. Tuttavia per i nuovi assi geometrici appena inseriti o con la posizione scambiata, viene considerata come non ancora eseguita. Al primo comando di movimento per questi assi geometrici, il percorso risultante è composto dalla somma della correzione di lunghezza utensile e dal percorso programmato.

Gli assi geometrici che durante una commutazione mantengono la loro posizione, conservano anche il loro stato riferito alla correzione della lunghezza utensile.

Configurazione degli assi geometrici e commutazione della trasformazione

La configurazione degli assi geometrici valida in una trasformazione (definita tramite dati macchina), non può essere modificata con la funzione "Assi geometrici commutabili".

Se è necessario modificare la configurazione degli assi geometrici in abbinamento a trasformazioni, questo è possibile solo tramite un'ulteriore trasformazione.

Una configurazione di assi geometrici modificata con `GEOAX`, viene cancellata all'attivazione di una trasformazione.

Se determinate impostazioni dei dati macchina dovessero essere discordanti per la trasformazione e per la commutazione degli assi geometrici, le impostazioni nella trasformazioni vengono considerate come prioritarie.

Esempio:

Sia attiva una trasformazione. Secondo i dati macchina la trasformazione deve restare attiva anche dopo un RESET, contemporaneamente, sempre dopo un RESET deve essere ripristinata la configurazione di base degli assi geometrici. In questo caso la configurazione degli assi geometrici viene mantenuta come definito con la trasformazione.

Correzioni utensile

8.1 Memoria di correzione

Funzione

Struttura della memoria di correzione

Ogni campo di dati può essere richiamato con un numero T e D (eccetto i "numeri D piani") e contiene per l'utensile, oltre ai dati geometrici, anche altre informazioni, per es. il tipo di utensile.

Struttura piatta dei numeri D

La "struttura piatta dei numeri D" viene utilizzata quando la gestione utensili avviene al di fuori dell'NCK. In questo caso i numeri D vengono creati con i relativi blocchi di correzione utensile senza assegnazione a utensili.

Nel partprogram si può continuare a programmare T. Questo valore T non ha tuttavia alcun riferimento al numero D programmato.

Costruttore della macchina

Con un dato macchina è possibile configurare i dati utente dei taglienti. Fare attenzione alle indicazioni del costruttore della macchina

Parametri

Nota

Singoli valori nella memoria di correzione

I singoli valori della memoria di correzione P1...P25 possono essere letti o scritti da programma mediante variabili di sistema. Tutti i restanti parametri sono riservati.

I parametri utensile

\$TC_DP6 bis \$TC_DP8, \$TC_DP10 e \$TC_DP11 come pure \$TC_DP15 bis \$TC_DP17, \$TC_DP19 e \$TC_DP20 hanno un significato diverso a seconda del tipo di utensile.

¹ Vale anche con utensili per la fresatura frontale 3D.

² Per tipo di utensile sega per scanalare.

³ Riservato: non utilizzato da SINUMERIK 840/810D.

Numero parametro utensile (DP)	Significato delle variabili di sistema	Nota
\$TC_DP1	Tipo di utensile	Panoramica: vedere lista
\$TC_DP2	Posizione di taglio	Solo per utensili da tornio
Geometria	Correzione lunghezza	
\$TC_DP3	Lunghezza 1	Calcolo
\$TC_DP4	Lunghezza 2	Tipo e livello
\$TC_DP5	Lunghezza 3	
Geometria	raggio	
\$TC_DP6 ¹ \$TC_DP6 ²	Raggio 1 / lunghezza 1 diametro d	Utensile per fresatura/tornio/ rettifica, sega per scanalare
\$TC_DP6 ¹ \$TC_DP6 ²	Lunghezza 2 / raggio agli spigoli frese coniche larghezza della cava b raggio agli spigoli	Utensili per fresatura sega per scanalare
\$TC_DP8 ¹ \$TC_DP8 ²	Raggio di arrotondamento 1 per utensili per fresatura sporgenza k	Utensili per fresatura sega per scanalare
\$TC_DP9 ^{1,3}	Raggio di arrotondamento 2	riservato
\$TC_DP10 ¹	Angolo 1 superficie frontale dell'utensile	Utensili conici per fresatura
\$TC_DP11 ¹	Angolo 2 asse longitudinale utensile	Utensili conici per fresatura
usura	Correzione lunghezza e raggio	
\$TC_DP12	Lunghezza 1	
\$TC_DP13	Lunghezza 2	
\$TC_DP14	Lunghezza 3	
\$TC_DP15 ¹ \$TC_DP15 ²	Raggio 1 / lunghezza 1 diametro d	Ut. per fresatura/tornio/rettifica sega per scanalare
\$TC_DP16 ¹ \$TC_DP16 ³	Lunghezza 2 / raggio agli spigoli frese coniche, larghezza della cava b raggio agli spigoli	Utensili per fresatura sega per scanalare
\$TC_DP17 ¹ \$TC_DP17 ²	Raggio di arrotondamento 1 per utensili per fresatura sporgenza k	Fresatura / 3D fresatura frontale 3D sega per scanalare
\$TC_DP18 ^{1,3}	Raggio di arrotondamento 2	riservato
\$TC_DP19 ¹	Angolo 1 superficie frontale dell'utensile	Utensili conici per fresatura
\$TC_DP20 ¹	Angolo 2 asse longitudinale utensile	Utensili conici per fresatura
Quota base/adattat.	Correzioni di lunghezza	
\$TC_DP21	Lunghezza 1	
\$TC_DP22	Lunghezza 2	
\$TC_DP23	Lunghezza 3	
Tecnologia		
\$TC_DP24	Angolo libero	Solo per utensili da tornio
\$TC_DP25		riservato

Note

Per le dimensioni geometriche (ad esempio lunghezza 1 oppure raggio) sono previsti più componenti da impostare. Esse vengono poi sommate algebricamente ottenendo un'unica grandezza risultante (ad esempio lunghezza totale 1, raggio totale).

Le correzioni non necessarie devono essere preimpostate a zero.

Parametri utensile da \$TC-DP1 a \$TC-DP23 con utensili di contornitura

Nota

I parametri utensile non riportati nella tabella, come ad es. \$TC_DP7, non vengono valutati, ossia il loro contenuto non è significativo.

Numero parametro utensile (DP)	Significato	Taglio Dn		Nota
\$TC_DP1	Tipo di utensile			da 400 a 599
\$TC_DP2	Posizione di taglio			
Geometria	Correzione lunghezza			
\$TC_DP3	Lunghezza 1			
\$TC_DP4	Lunghezza 2			
\$TC_DP5	Lunghezza 3			
Geometria	Raggio			
\$TC_DP6	Raggio			
Geometria	Angolo limite			
\$TC_DP10	Angolo limite minimo			
\$TC_DP11	Angolo limite massimo			
Usura	Correzione lunghezza e raggio			
\$TC_DP12	Usura lunghezza 1			
\$TC_DP13	Usura lunghezza 2			
\$TC_DP14	Usura lunghezza 3			
\$TC_DP15	Raggio di usura			
Usura	Angolo limite			
\$TC_DP19	Usura, angolo limite minimo			
\$TC_DP20	Usura, angolo limite massimo			
Quota base/adattat.	Correzioni di lunghezza			
\$TC_DP21	Lunghezza 1			
\$TC_DP22	Lunghezza 2			
\$TC_DP23	Lunghezza 3			

Valore base e valore di usura

Le grandezze risultanti sono di volta in volta la somma tra valore base e valore di usura (ad es. \$TC_DP6 + \$TC_DP15 per il raggio). Alla lunghezza utensile del primo taglio viene aggiunta la quota base (\$TC_DP21 – \$TC_DP23). Inoltre, su questa lunghezza utensile agiscono tutte le altre grandezze capaci di influenzare la lunghezza utensile effettiva anche nel caso di un utensile tradizionale (adattatore, portautensile orientabile, dati setting).

Angoli limite 1 e 2

Gli angoli limite 1 e 2 si riferiscono al vettore dal centro di taglio al punto di riferimento di taglio e vengono numerati in senso antiorario.

8.2 Istruzioni per la gestione utensili

Funzione

Utilizzando la gestione utensili è possibile modificare ed aggiornare i dati degli utensili.

Tramite funzioni predefinite nel programma NC è possibile:

- Creare e richiamare utensili con il relativo nome.
- Creare un nuovo utensile oppure cancellarne uno esistente.
- Assegnare il numero T necessario ad un utensile con nome noto.
- Aggiornare i dati per la sorveglianza del contapezzi.
- Leggere il numero T dell'utente scelto per il mandrino.

Programmazione

```
T="PUNTA" oppure T="123" Utensili con nomi
oppure
parametro di conferma =NEWT("WZ", DUPLO_NR)
oppure
DELT("WZ", NR_DUPLO)
oppure
Parametro di conferma =GETT("WZ", DUPLO_NR)
oppure
SETPIECE(x,y)
oppure
GETSELT (x)
```

Parametri

T="WZ"	Selezione dell'utensile con il nome
NEWT ("WZ", DUPLO_NR)	Creazione di un nuovo utensile, n. duplo opzionale
DELT ("WZ", DUPLO_NR)	Cancellazione di un utensile, n. duplo opzionale
GETT ("WZ", DUPLO_NR)	Determinazione del numero T
SETPIECE(x,y)	Impostazione del numero di pezzi
GETSELT (x)	Lettura del numero utensile selezionato (nr. T)
"UT"	Identificatore utensile
DUPLO_NR	n. dei pezzi
x, y	Numero del mandrino, impostazione opzionale

Esempio di funzione NEWT

Con la funzione **NEWT**, nel programma NC può essere progettato un nuovo utensile con un nome. La funzione fornisce, come parametro di conferma, il numero T generato automaticamente, con il quale l'utensile può essere ora indirizzato.

Se non viene indicato alcun numero duplo, esso viene generato dalla gestione utensili.

```
DEF INT NR_DUPLO
DEF INT T_NR
DUPLO_NR = 7
T_NR=NEWT("PUNTA", NR_DUPLO)
```

;Creazione dell'utensile "PUNTA" con numero duplo 7.
;Il numero di T generato viene depositato in T_NR.

Esempio di funzione DELT

Con la funzione **DELT** può essere cancellato un utensile senza riferimento a un numero T.

Esempio di funzione GETT

La funzione **GETT** fornisce il numero T necessario per il preset dei dati utensili a un utensile del quale è noto soltanto il nome.

Se con l'identificatore impostato sono previsti più utensili, viene restituito il primo possibile tra questi utensili.

Viene restituito = -1: all'identificatore utensile o al numero duplo non può essere abbinato alcun utensile.

```
T="PUNTA"
R10=GETT("PUNTA", NR_DUPLO)
```

;Determina il numero T per PUNTA con il
;numero duplo = NR_DUPLO

La "PUNTA" deve essere precedentemente dichiarata con **NEWT** oppure `$TC_TP1 [ ]`.

```
$TC_DP1[GETT("PUNTA", NR_DUPLO),1]=100
```

;Scrittura di un parametro utensile (variabile
;di sistema) con il nome dell'utensile

Esempio di funzione SETPIECE

Questa funzione permette di aggiornare i dati di sorveglianza del numero di pezzi. La funzione comprende tutti gli inserti utensile che sono stati inseriti sul mandrino del numero indicato dopo l'ultima attivazione di **SETPIECE**.

SETPIECE (x, y)

```
x
y
```

;Numero dei pezzi prodotti
;y numero del mandrino, 0 significa mandrino
;master (impostazione standard)

Esempio di funzione GETSELT

Questa funzione fornisce il numero T dell'utensile scelto per il mandrino. In questo modo è possibile accedere ai dati di correzione dell'utensile già prima di M6 e quindi approntare la sincronizzazione con l'esecuzione del programma con un certo anticipo.

Esempio di cambio utensile con gestione utensili

T1:

preselezione dell'utensile; cioè il magazzino utensili può essere collocato nella posizione dell'utensile parallelamente alla lavorazione.

M6:

sostituzione di un utensile preselezionato (in funzione dell'impostazione dei dati macchina si può programmare anche senza M6).

T1 M6	;Sostituzione dell'utensile 1
D1	;Attivazione della correzione lunghezza ;utensile
G1 X10 ...	;Lavorazione con T1
T= "PUNTA"	;Preselezione dell'utensile per foratura
D2 Y20 ...	;Cambio tagliente dell'utensile T1
X10 ...	;Lavorazione con T1
M6	;Sostituzione dell'utensile per foratura
SETPIECE(4)	;Numero dei pezzi prodotti
D1 G1 X10 ...	;Lavorazione con utensile per foratura

Nota

Per una lista completa di tutte le variabili per la gestione utensili, consultare:

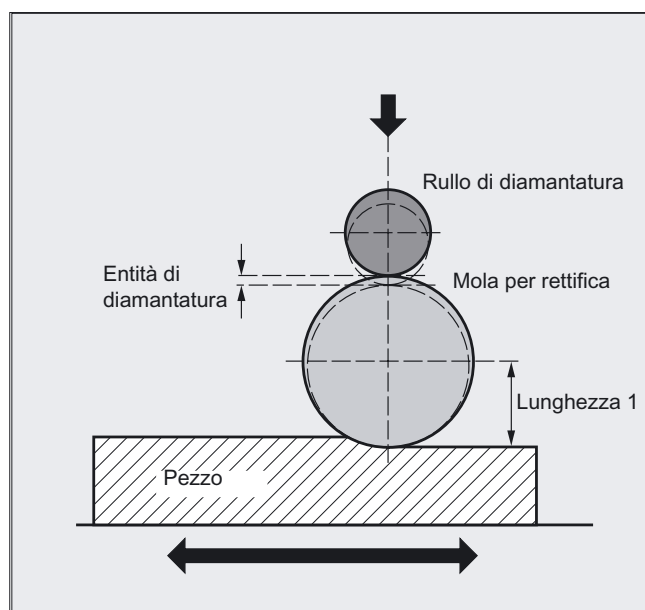
Bibliografia:

/PGA1/ Lista delle variabili di sistema.

8.3 Correzione utensile on-line (PUTFTOCF, PUTFTOC, FTOCON, FTOCOF)

Funzione

Con questa funzione, le correzioni utensili derivanti dalla lavorazione, vengono calcolate immediatamente grazie alla correzione utensile on-line (es. diamantatura CD: la mola viene diamantata parallelamente alla lavorazione). La correzione lunghezza utensile può essere modificata dal canale di lavorazione o da un canale parallelo (canale di diamantatura).



Nota

La correzione utensile on-line può essere utilizzata solo per utensili di rettificazione.

Programmazione

```
FCTDEF(Nr. polinomio, LLimit, ULimit,a0,a1,a2,a3)
```

oppure

```
PUTFTOCF(Nr. polinomio, valore di riferim., lungh.1_2_3, canale, mandr.)
```

oppure

```
PUTFTOC(valore, lunghezza1_2_3, canale, mandrino)
```

oppure

```
FTOCON
```

oppure

```
FTOCOF
```

Parametri

PUTFTOCF	Scrittura continua della correzione utensile on-line
FCTDEF	Parametrizzazione della funzione PUTFTOCF
PUTFTOC	Scrittura della correzione utensile on-line in forma discreta
FTOCON	Attivazione della correzione utensile on-line
FTOCOF	Disattivazione della correzione utensile on-line
Nr. polinomio	Valore 1...3 sono possibili al massimo 3 polinomi contemporaneamente, polinomi fino al 3. grado
Valore di riferimento	Valore di riferimento dal quale viene ricavata la correzione
Lunghezza1_2_3	Parametro di usura nel quale viene sommato il valore di correzione utensile
Canale	Numero del canale nel quale viene attivata la correzione utensile; l'indicazione è necessaria solo se non si tratta del proprio canale
Mandrino	Numero del mandrino per il quale agisce la correzione utensile online; indicazione necessaria solo per la mola non attiva
LLimit	Valore limite superiore
ULimit	Valore limite inferiore
a_0, a_1, a_2, a_3	Coefficienti della funzione polinomica
Valore	Valore che viene sommato nel parametro di usura

Esempio

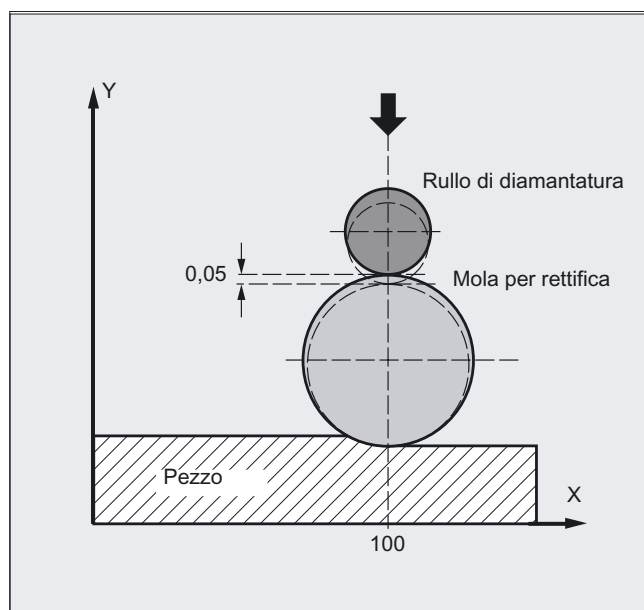
In una rettificatrice con le caratteristiche sottoportate, dopo l'inizio della passata di rettificazione su X100 la mola deve essere diamantata di 0,05. Il valore di diamantatura deve essere attivato continuamente con scrittura CUT on-line.

Y: asse di incremento per la mola

V: asse di incremento per il rullo diamantatore

Lavorazione: canale 1 con gli assi X, Z, Y

Diamantatura: canale 2 con l'asse V



Programma di lavorazione nel canale 1:

%_N_BEARB_MPF	
...	
N110 G1 G18 F10 G90	;Posizione base
N120 T1 D1	;Selezionare l'utensile attuale
N130 S100 M3 X100	;Mandrino ON, raggiungere la posizione iniziale
N140 INIT (2, "DIAMANTATURA", "S")	;Selezione del programma di diamantatura nel canale 2
N150 START (2)	;Avvio del programma di diamantatura nel canale 2
N160 X200	;Raggiungimento della posizione finale
N170 FTOCON	;Inserire la correzione on-line
N... G1 X100	;Ulteriori lavorazioni
N... M30	

Programma di diamantatura nel canale 2:

%_N_ABRICHT_MPF	
...	
N40 FCTDEF (1, -1000, 1000, -\$AA_IW[V], 1)	;Definizione della funzione: Retta
N50 PUTFTOCF (1, \$AA_IW[V], 3, 1)	;Scrittura continua della CUT on-line: la lunghezza 3 della mola attuale, derivata dal movimento dell'asse V, viene corretta nel canale 1.
N60 V-0.05 G1 F0.01 G91	;Movimento di incremento per la diamantatura, PUTFTOCF è attivo solo in questo blocco.
...	
N... M30	

Programma di diamantatura modale:

%_N_ABRICHT_MPF	
FCTDEF (1, -1000, 1000, -\$AA_IW[V], 1)	;Definizione della funzione.
ID=1 DO FTOC(1, \$AA_IW[V], 3, 1)	;Selezionare la correzione utensile online:
	;il valore reale dell'asse V costituisce il valore
	;d'ingresso del polinomio 1 il risultato viene
	;sommato nel canale 1 quale valore di correzione
	;alla lunghezza 3 della mola attiva.
WAITM(1, 1, 2)	;Eeguire la sincronizzazione con il canale di
	;lavorazione
G1 V-0.05 F0.01, G91	;Movimento di incremento per la diamantatura
G1 V-0.05 F0.02	
...	
CANCEL(1)	;Disattivare la correzione on-line
...	

Descrizione**Informazioni generali sulla CUT (correz. utensile) on-line**

A seconda del momento in cui avviene la diamantatura, per la scrittura della CUT on-line vengono utilizzate diverse funzioni:

- Scrittura continua blocco-blocco: PUTFTOCF
- Scrittura continua modale: ID=1 DO FTOC (vedere capitolo azioni sincrone)
- Scrittura discreta: PUTFTOC

Nella scrittura continua (per ogni clock IPO), allo scopo di evitare salti dei valori di riferimento, dopo l'attivazione della funzione di valutazione ogni modifica viene memorizzata in modo additivo nella memoria di usura. In ogni caso vale: la CUT on-line può avere effetto su ogni canale, per ogni mandrino e per la lunghezza 1, 2 **oppure** 3 dei parametri di usura.

L'abbinamento delle lunghezze agli assi geometrici avviene in funzione del piano attuale.

L'abbinamento del mandrino all'utensile avviene tramite i dati dell'utensile con GWPSON oppure TMON, sempre che non si tratti della mola attiva (vedere Manuale di programmazione "Concetti fondamentali"). La correzione avviene sempre nel parametro di usura per il fianco attuale oppure per il fianco sinistro della mola nel caso di utensili non attivi.

Nota

Se la correzione è identica per più lati della mola, bisogna fare in modo tramite le prescrizioni di concatenamento (vedere manuale operativo) che i valori vengano assegnati automaticamente anche al secondo lato della mola.

Se vengono preimpostate delle correzioni on-line per un canale di lavorazione, i valori di usura per l'utensile attuale di questo canale non possono essere variati dal programma di lavoro o attraverso manovre operative.

La correzione utensile on-line viene presa in considerazione anche per la velocità periferica costante della mola (VPM) e per la sorveglianza utensile TMON.

PUTFTOCF = scrittura continua

La diamantatura avviene contemporaneamente alla lavorazione: diamantatura su tutta la larghezza della mola con rullo diamantatore o con diamante da una fiancata all'altra della mola.

Lavorazione e diamantatura avvengono di regola in canali differenti. Se non è programmato nessun canale, la correzione ha effetto nel canale attivo.

PUTFTOCF (Nr. polinomio, valore di riferim., lungh.1_2_3, canale, mandr.)

La correzione utensile viene variata continuamente nel canale di lavoro secondo una funzione polinomica di 1°, 2° o 3° grado che deve essere definita in precedenza con FCTDEF. Dalla variabile "Valore di riferimento" viene ricavata la correzione es.: variazione del valore reale. Se non è stato programmato nessun numero di mandrino, la correzione viene applicata all'utensile attivo impiegato al momento.

Parametrizzazione della funzione FCTDEF

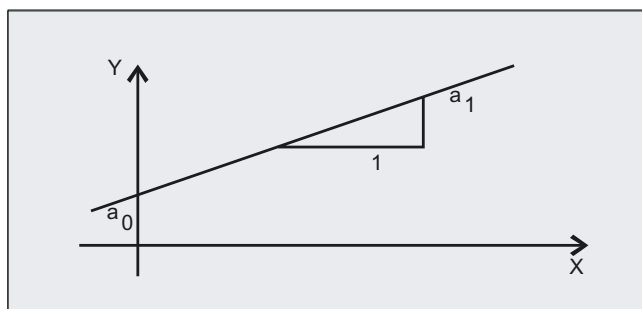
La parametrizzazione avviene in un blocco a sé stante:

FCTDEF (Nr. polinomio, LLimit, ULimit, a0,a1,a2,a3)

Il polinomio può essere di 1°, 2° oppure di 3° grado. Limit indica il valore limite (LLimit = valore inferiore, ULimit = valore superiore).

Esempio: Retta ($y = a_0 + a_1x$) con incremento 1

FCTDEF (1, -1000, 1000, -\$AA_IW[X], 1)

**Scrittura discreta della CUT on-line: PUTFTOC**

Con questa istruzione è possibile scrivere **una volta** un valore di correzione. La correzione diventa immediatamente attiva nel canale di destinazione.

Utilizzo di PUTFTOC: la mola viene diamantata da un canale parallelo, ma non contemporaneamente alla lavorazione.

PUTFTOC (valore, lunghezza1_2_3, canale, mandrino)

La CUT on-line per le lunghezze 1, 2 **oppure** 3 viene variata del valore indicato; il valore viene quindi sommato nel parametro dell'usura.

Calcolo online della CUT: FTOCON, FTOCOF

Il canale di destinazione può ricevere le correzioni utensili on-line solo se è attiva FTOCON.

- FTOCON deve essere scritta nel canale in cui deve essere attivata la correzione. Con FTOCOF la correzione non viene eseguita: nei dati di correzione specifici per il taglio, l'intero valore scritto con PUTFTOC viene tuttavia corretto.
- FTOCOF rappresenta sempre la condizione di reset.

- PUTFTOCF agisce sempre blocco-blocco, cioè nel blocco di movimento successivo.
- Con FTOC la correzione on-line può essere selezionata anche modalmente. Per ulteriori informazioni consultare il capitolo "Azioni sincrone al movimento".

8.4 Mantenimento costante della correzione del raggio utensile (CUTCONON)

Funzione

La funzione "Mantenere costante la correzione del raggio utensile" permette di sopprimere la correzione del raggio utensile per un certo numero di blocchi, mantenendo tuttavia come traslazione una differenza generata nei blocchi precedenti dalla correzione del raggio utensile tra il percorso programmato e quello effettivamente eseguito dal centro dell'utensile. Può essere utilizzata, ad esempio, quando nella fresatura lineare si rendono necessari nei punti di inversione più blocchi di movimento, ma non si desiderano i profili generati dalla correzione raggio utensile (strategie di aggiramento). Si può impiegare indipendentemente dal tipo di correzione del raggio utensile (fresatura frontale 2½D, fresatura frontale 3D, fresatura perimetrale 3D).

Programmazione

CUTCONON
CUTCONOF

Parametri

CUTCONON	Mantiene costantemente attiva la funzione Correzione del raggio utensile
CUTCONOF	Disattiva la funzione Correzione del raggio utensile costante (impostazione predefinita)

Esempio

N10		;Definizione dell'utensile d1
N20	\$TC_DP1[1,1] = 110	;Tipo
N30	\$TC_DP6[1,1] = 10.	;raggio
N40		
N50	X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60		
N70	X20 G42 NORM	
N80	X30	
N90	Y20	
N100	X10 CUTCONON	;Attivazione della soppressione della correzione
N110	Y30 KONT	;Inserire eventualmente il cerchio di aggiramento quando viene disattivata la soppressione della correzione
N120	X-10 CUTCONOF	

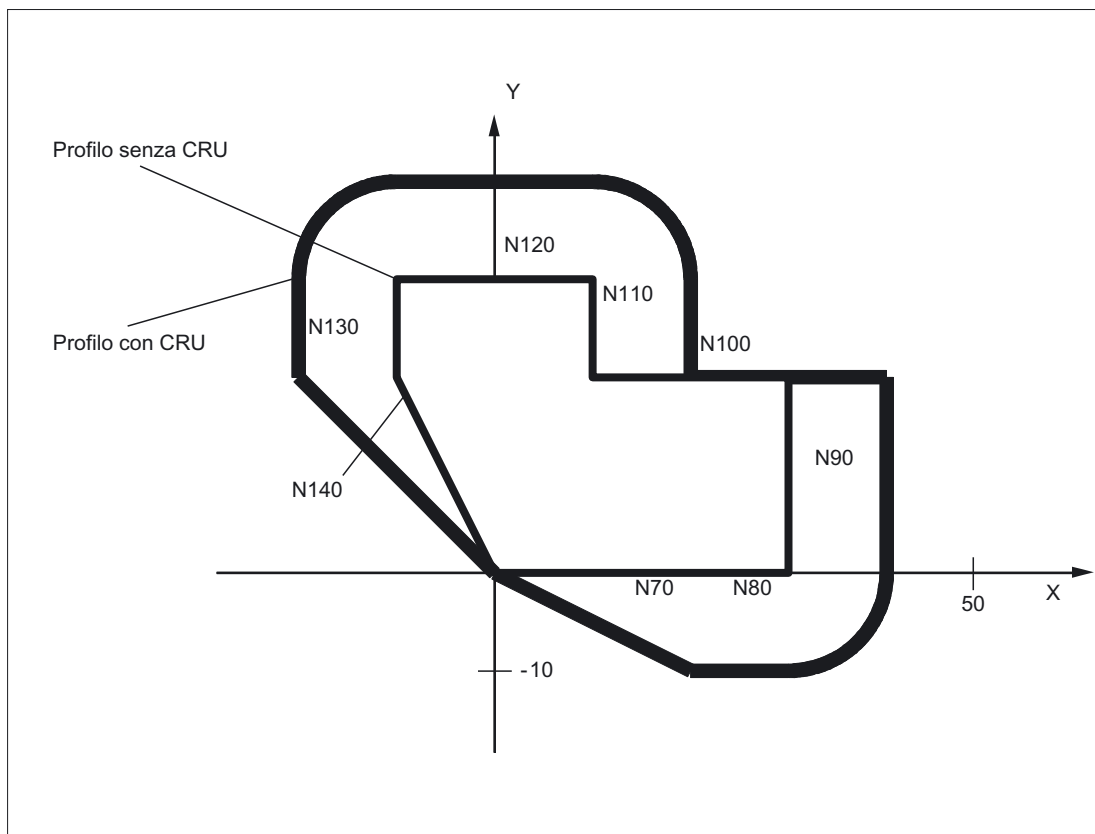
8.4 Mantenimento costante della correzione del raggio utensile (CUTCONON)

```

N130 Y20 NORM
N140 X0 Y0 G40
N150 M30

```

;Nessun cerchio di aggiramento alla disattivazione della
;correzione del raggio utensile



Descrizione

Nel caso normale la correzione del raggio utensile è già attiva prima che venga attivata la cancellazione della correzione, ed è ancora attiva quando quest'ultima viene di nuovo disattivata. Nell'ultimo blocco di movimento prima di `CUTCONON` il punto di offset viene raggiunto nel punto finale del blocco. Tutti i blocchi successivi, in cui è attiva la soppressione della correzione, vengono eseguiti senza correzione. Essi vengono tuttavia spostati intorno al vettore dal punto finale dell'ultimo blocco di correzione fino al punto di offset. Il tipo di interpolazione di questi blocchi (lineare, circolare, polinomiale) è libero.

Il blocco di disattivazione della soppressione di correzione, ovvero il blocco che contiene `CUTCONOF`, viene corretto normalmente e inizia nel punto di offset del punto iniziale. Tra questo punto e il punto finale del blocco precedente, cioè dell'ultimo blocco di movimento programmato con `CUTCONON` attivo, viene inserito un blocco lineare.

I blocchi circolari in cui il piano di lavoro del cerchio è perpendicolare al piano di correzione (cerchi verticali) vengono trattati come se fosse stata programmata la funzione `CUTCONON`. Questa attivazione implicita della soppressione di correzione viene automaticamente annullata nel primo blocco di movimento, che contiene un movimento di posizionamento nel piano di correzione e che non è un cerchio di questo tipo. I cerchi verticali, in questo senso, possono presentarsi solo nella fresatura perimetrale.

8.5 Attivazione delle correzioni utensili 3D (CUT3DC..., CUT3DF...)

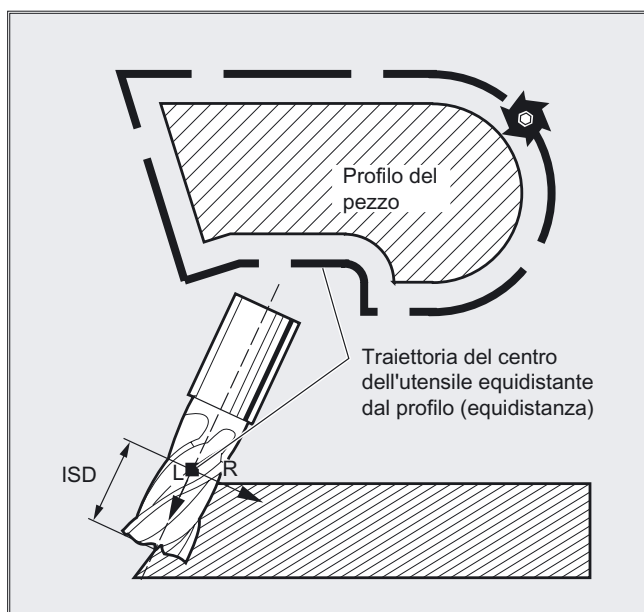
8.5.1 Attivazione delle correzioni utensili 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF)

Funzione

Nel caso della correzione del raggio utensile per utensili cilindrici, viene tenuto in considerazione l'orientamento variabile

Per l'attivazione della correzione raggio utensile 3D valgono le stesse istruzioni di programmazione della correzione 2D. Con G41/G42 la correzione viene indicata a sinistra/destra della direzione di movimento. L'accostamento è sempre del tipo NORM. La correzione raggio utensile 3D ha effetto solo con la trasformazione 5 assi attiva.

La correzione raggio utensile 3D viene denominata anche correzione 5D poiché in questo caso sono disponibili 5 gradi di libertà per la posizione dell'utensile nello spazio.



Differenza tra correzione raggio utensile 2 1/2D e 3D

Nella correzione raggio utensile 3D l'orientamento dell'utensile è variabile.

Con correzione raggio utensile 2 1/2D viene considerato solo un utensile con orientamento costante nello spazio.

Programmazione

CUT3DC

oppure

CUT3DFS

oppure

CUT3DFF

oppure

CUT3DF

L'istruzione ha validità modale e fa parte dello stesso gruppo di CUT2D e CUT2DF. La disattivazione è effettiva solo con il successivo movimento nel piano di lavoro attuale. Questo è sempre valido per G40 ed è indipendente dal comando CUT.

Con correzione raggio utensile 3D attiva sono consentiti blocchi intermedi. Valgono le specifiche della correzione raggio utensile 2 1/2D.

Parametri

CUT3DC	Attivazione della correzione raggio 3D per la fresatura periferica
CUT3DFS	Correzione utensile D per la fresatura frontale con orientamento costante. L'orientamento dell'utensile è definito da G17 - G19 e non viene influenzato dai frame.
CUT3DFF	Correzione utensile D per la fresatura frontale con orientamento costante. L'orientamento dell'utensile è definito da G17 - G19 ed eventualmente da un frame in direzione ruotata.
CUT3DF	Correzione utensile D per fresatura frontale con orientamento variabile (solo con trasformazione a 5 assi attiva).
G40 X Y Z	Per la disattivazione: blocco lineare G0/G1 con assi geometrici
ISD=valore	profondità di penetrazione

G450/G451 e DISC

Sugli spigoli esterni viene sempre inserito un blocco circolare. G450/G451 sono irrilevanti.

L'istruzione DISC non viene interpretata.

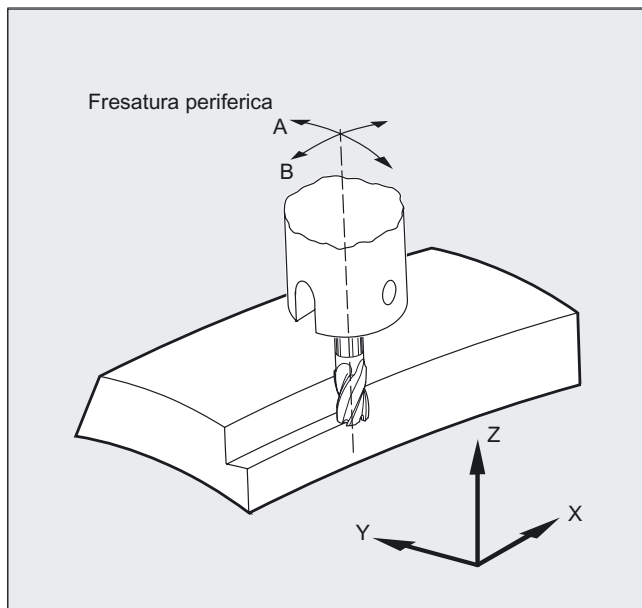
Esempio

N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	;Richiamo dell'utensile e dei valori di correzione ;utensile
N30 TRAORI(1)	;Attivazione della trasformazione
N40 CUT3DC	;Attivazione della correzione raggio utensile 3D
N50 G42 X10 Y10	;Attivazione della correzione raggio utensile
N60 X60	
N70 ...	

8.5.2 Correzione raggio utensile 3D: fresatura periferica e frontale

Fresatura periferica

La variante di fresatura periferica qui utilizzata viene realizzata preimpostando un profilo (linea guida) ed il relativo orientamento. Con questo tipo di lavorazione, sul percorso e sugli spigoli esterni la forma dell'utensile non è rilevante. E' determinante solo il raggio sul punto di contatto dell'utensile.

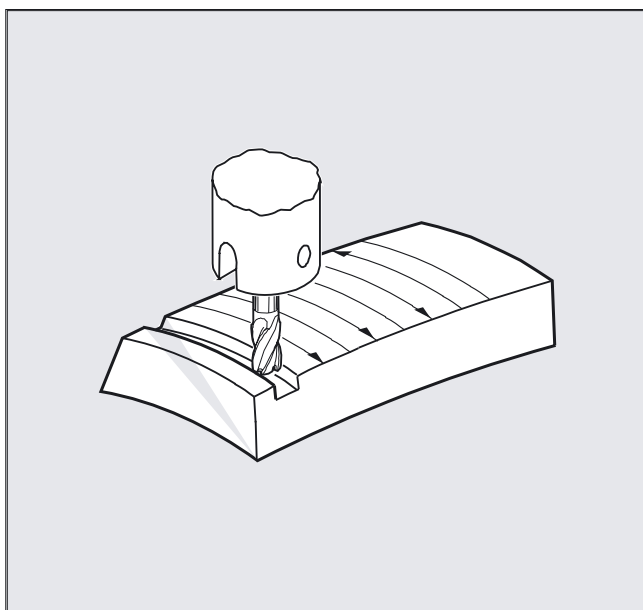


Nota

La funzione 3D-WRK è limitata agli utensili cilindrici.

Fresatura frontale

Per questo tipo di fresatura 3D è necessaria la descrizione riga per riga della traiettoria 3D sulla superficie del pezzo. Il calcolo viene effettuato normalmente con un CAM tenendo conto della forma e delle dimensioni dell'utensile. Il postprocessor trascrive nel partprogram, oltre ai blocchi NC, gli orientamenti dell'utensile (con trasformazione a 5 assi attiva) e il codice G per la correzione utensile 3D desiderata. In tal modo l'utente ha la possibilità di impiegare anche utensili di dimensioni leggermente inferiori rispetto a quelle degli utensili utilizzati per il calcolo della traiettoria del NC.

**Esempio:**

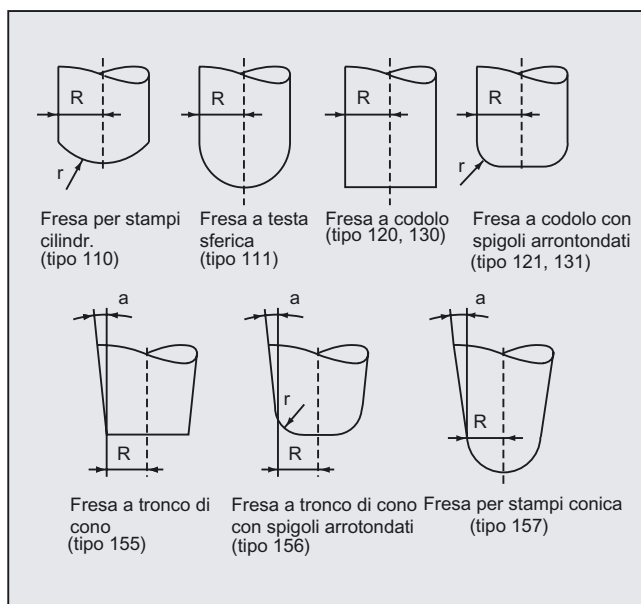
I blocchi NC sono stati calcolati sulla base di una fresa di diametro 10 mm. In questo caso si potrebbe impiegare anche una fresa di diametro 9,9 mm, tenendo conto tuttavia che la rugosità del profilo risulterebbe lievemente modificata.

8.5.3 Tipi di utensile/cambio utensile con dimensioni modificate (G40, G41, G42)

Funzione

Forme degli utensili per fresatura, dati utensile

La tabella a destra riporta le forme di utensili possibili per la fresatura frontale e i valori limite dei dati utensile. La forma del codolo dell'utensile non viene considerata; l'effetto degli utensili del tipo 120 e 156 è identico.



Se nel partprogram NC è indicato un numero di utensile non contenuto nella tabella, il sistema utilizza automaticamente l'utensile 110 (fresa cilindrica per stampi). Se i valori impostati per i dati utensili superano il valore limite viene emesso un allarme.

Parametri

Tipo di fresa	N. del tipo	R	r	a
Fresa cilindrica per stampi	110	>0	X	X
Fresa a testa cilindrica	111	>0	>R	X
Fresa a codolo, fresa a testa angolare	120, 130	>0	X	X
Fresa a codolo, fresa a testa angolare con arrotondamento degli spigoli	121, 131	>r	>0	X
Fresa a tronco di cono	155	>0	X	>0
Fresa a tronco di cono con arrotondamento degli spigoli	156	>0	>0	>0
Fresa per stampi conica	157	>0	X	>0

Dati dell'utensile	Parametri utensile		X = non viene interpretato
Dimensioni dell'utensile	Geometria	usura	
R	\$TC_DP6	\$TC_DP15	R = raggio del codolo (raggio dell'utensile)
r	\$TC_DP7	\$TC_DP16	r = raggio dello spigolo
a	\$TC_DP11	\$TC_DP20	a = angolo tra l'asse longitudinale dell'utensile e l'estremità superiore della superficie toroidale

Correzione lunghezza utensile (CLU)

Come punto di riferimento per la correzione della lunghezza viene considerata la punta dell'utensile (punto di intersezione asse longitudinale/superficie).

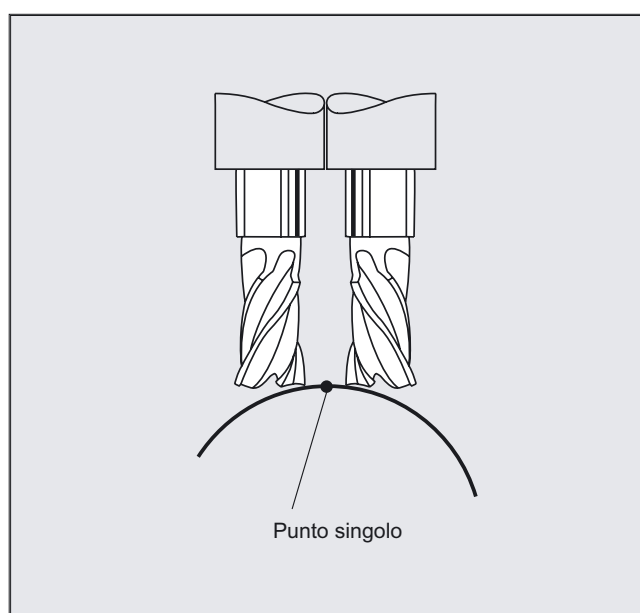
Correzione utensile 3D, cambio utensile

L'indicazione di un nuovo utensile con quote modificate (R, r, a) o con un'altra forma è consentita solo se sono stati programmati G41 oppure G42 (passaggio da G40 a G41 o G42, riprogrammare G41 oppure G42). Questa regola non riguarda tutti gli altri dati utensile, come ad es. la lunghezza, questi tipi di utensile possono pertanto essere sostituiti anche senza riprogrammare necessariamente G41 oppure G42.

8.5.4 Correzione sulla traiettoria, curvatura della traiettoria, profondità di inserzione ISD e posizione utensile (CUT3DC)

Funzione**Correzione sulla traiettoria**

Nel caso della fresatura frontale bisogna tener conto del fatto che il punto di contatto sulla superficie dell'utensile "salta", come in questo esempio nella lavorazione di una superficie convessa con utensile disposto verticalmente. L'esempio mostrato nella figura può essere considerato come un caso limite.



Il controllo sorveglia tale caso limite e riconosce gli spostamenti del punto di contatto sulla base della posizione dell'angolo tra l'utensile e il vettore normale alla superficie. In corrispondenza di questi punti il controllo inserisce un blocco con spostamento lineare in modo che il movimento possa essere eseguito correttamente.

Per calcolare i blocchi lineari vengono inseriti nei dati macchina i valori consentiti per l'angolo laterale. Se i valori limite consentiti per gli angoli vengono superati, il controllo emette un allarme.

Curvatura della traiettoria

La curvatura della traiettoria non viene sorvegliata. Anche in questo caso si raccomanda di impiegare solo quei tipi di utensile che consentono una lavorazione del profilo precisa.

Programmazione

Profondità di inserzione ISD

ISD viene considerato solo se è attiva la correzione raggio utensile 3D.

Con l'istruzione di programma ISD (Insertion Depth) viene programmata la profondità di inserzione dell'utensile nella fresatura periferica. Questa consente la variazione del punto di lavorazione dell'utensile sulla superficie.

Correzione utensile 3D, fresatura periferica

CUT3DC

Parametri

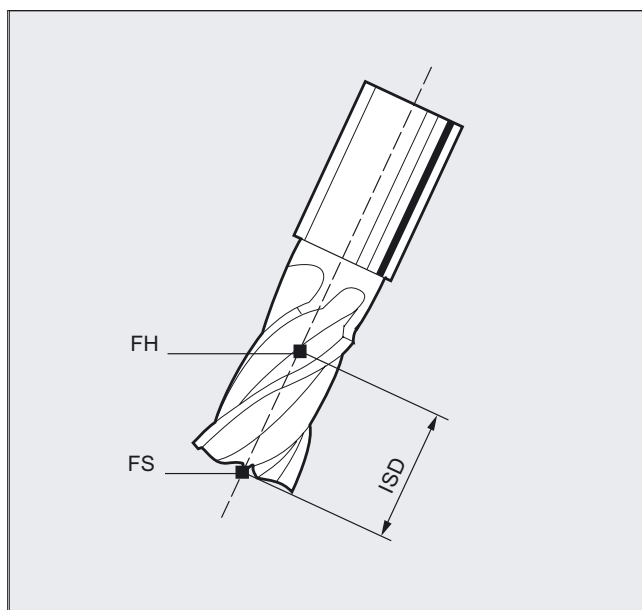
CUT3DC

Attivazione delle correzioni utensili 3D per fresatura periferica, es. fresatura di tasche con pareti laterali inclinate.

ISD

ISD indica la distanza fra la punta della fresa (FS)) ed il punto fresa ausiliario (FH).

Il punto FH si ottiene dalla proiezione del punto di lavoro programmato sull'asse dell'utensile.



Descrizione

Fresatura di tasche con pareti inclinate per fresatura periferica con CUT3DC

Con questa correzione raggio utensile 3D viene compensata una differenza del raggio utensile mentre si incrementa in direzione della normale alla superficie da lavorare. In questo modo, il piano in cui si trova il lato frontale della fresa resta invariato quando la profondità di inserzione I_{SD} rimane identica. Una fresa con, ad esempio, un raggio inferiore rispetto ad un utensile standard, non avrebbe raggiunto il fondo della tasca che rappresenta anche la superficie limite. Per un incremento automatico dell'utensile il controllo deve conoscere questa superficie limite. Per ulteriori informazioni vedere il capitolo "Fresatura periferica 3D con superfici limite"

Per ulteriori informazioni sulla sorveglianza anticollisione, consultare

Bibliografia: /PG/ Manuale di programmazione, Concetti fondamentali, capitolo "Correzioni utensili".

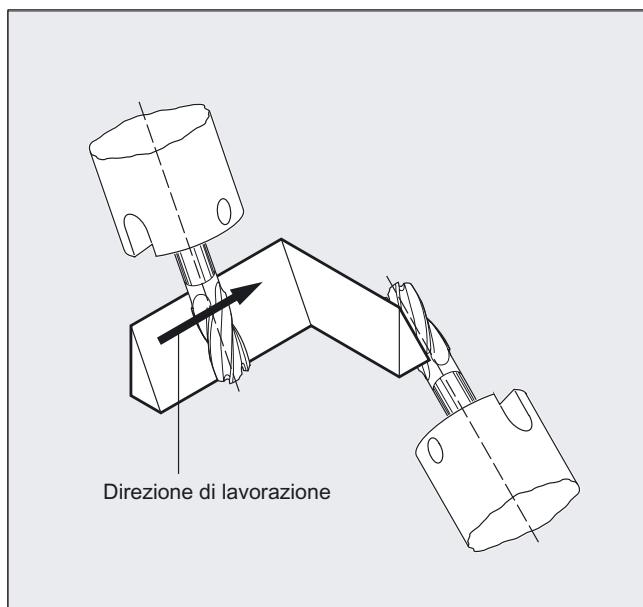
8.5.5 Spigolo interno/esterno e andamento sul punto di intersezione (G450/G451)

Funzione

Spigoli interni/esterni

Gli spigoli interni ed esterni vanno trattati separatamente. La definizione di spigolo interno ed esterno dipende dall'orientamento dell'utensile.

Nella variazione dell'orientamento su uno spigolo può verificarsi che durante la lavorazione il tipo di spigolo venga modificato. In questo caso la lavorazione viene interrotta con un messaggio di errore.



Programmazione

G450

oppure

G451

Parametri

G450	Cerchio di raccordo (l'utensile aggira gli spigoli del pezzo con una traiettoria circolare).
G451	Punto d'intersezione delle equidistanti (l'utensile esegue uno scarico nello spigolo del pezzo).

Descrizione

Punto di intersezione per correzione 3D

Nel caso della fresatura periferica 3D, sugli spigoli esterni viene valutato il codice G G450/G451, ovvero può essere raggiunto il punto di intersezione delle curve di offset. Fino al SW 4 è stato sempre inserito un cerchio in corrispondenza degli spigoli esterni. Il comportamento sul punto di intersezione è particolarmente vantaggioso per programmi 3D tipicamente realizzati con CAD. Tali programmi sono spesso costituiti da brevi blocchi di rette (per approssimare curve piatte), per le quali i raccordi tra blocchi contigui sono quasi tangenziali.

Per la correzione del raggio utensile sul lato esterno del profilo fino ad ora venivano inseriti fondamentalmente dei cerchi per aggirare gli spigoli esterni. Dal momento che questi blocchi diventano molto brevi se i raccordi sono quasi tangenziali, ne conseguono interruzioni indesiderate della velocità.

In questi casi, analogamente alla correzione del raggio $2 \frac{1}{2} D$, le due curve interessate vengono prolungate e viene raggiunto il punto di intersezione delle due curve prolungate.

Le curve di offset dei due blocchi interessati vengono prolungate e il loro punto di intersezione viene determinato nel piano perpendicolarmente all'orientamento dell'utensile sullo spigolo. Se non esiste un punto di intersezione, lo spigolo viene trattato come in precedenza, ovvero viene inserito un cerchio.

Bibliografia:

Per ulteriori informazioni sull'andamento nel punto di intersezione vedere: /FB/ W5, Correzione raggio utensile 3D.

8.5.6 Fresatura periferica 3D con superfici limite, impiego generico

Funzione

Adattamento di fresature periferiche 3D a situazioni di programmi CAD

I programmi NC generati da sistemi CAD 3D, approssimano normalmente il percorso del centro utensile con un notevole numero di brevi tratti lineari. Per fare in modo che i blocchi generati con questo sistema riformino il profilo originale con la massima precisione possibile, è necessario apportare alcuni adattamenti nel partprogram.

Informazioni importanti che sarebbero necessarie per una correzione ottimale ma che però non sono più disponibili nel partprogram, devono essere sostituite da opportuni accorgimenti. Di seguito vengono riportati alcuni metodi tipici per compensare raccordi critici

- direttamente nel partprogram oppure
- rilevando il profilo reale, es. avanzando con l'utensile.

Applicazioni

Oltre al caso tipico di impiego, nei quali invece dell'utensile normalizzato un utensile reale descrive la traiettoria del centro, vengono trattati anche utensili cilindrici con correzione utensile 3D. Con CUT3DCC il partprogram si riferisce al profilo della superficie da lavorare. In questo caso la superficie limite è indipendente dall'utensile. Come nel caso della correzione raggio utensile standard, per il calcolo dell'offset ortogonale rispetto alla superficie limite viene utilizzato il raggio globale.

8.5.7 Considerazione di una superficie limite (CUT3DCC, CUT3DCCD)

Funzione

Fresatura periferica 3D con utensili reali

Nella fresatura periferica 3D con orientamento dell'utensile continuo o costante, viene programmata spesso la traiettoria rispetto al centro di un utensile normalizzato. Siccome nella pratica molto spesso non sono disponibili gli utensili standard adeguati, è possibile utilizzare un utensile che non si discosti troppo da quello previsto.

Con CUT3DCCD viene considerata una superficie limite per un utensile differenziale reale, che descriverebbe l'utensile normalizzato programmato. Il programma NC descrive il percorso riferito al centro dell'utensile normalizzato.

Con CUT3DCC utilizzando utensili cilindrici viene considerata una superficie limite che è stata raggiunta dall'utensile normalizzato programmato. Il programma NC descrive il profilo sulla superficie di lavorazione.

Programmazione

CUT3DCCD

oppure

CUT3DCC

Parametri

CUT3DCCD

Attivazione della correzione utensile 3D per la fresatura periferica con superfici limite ed utensile differenziale sulla traiettoria del centro utensile:
Incremento sulla superficie limite.

CUT3DCC

Attivazione della correzione utensile 3D per la fresatura periferica con superfici limite e correzione raggio 3D.
Profilo sulla superficie di lavorazione

Nota

Correzione raggio utensile con G41, G42

Per la correzione raggio utensile con G41, G42 e CUT3DCCD oppure CUT3DCC attivi, si deve attivare l'opzione "Trasformazione dell'orientamento".

Utensili standard con arrotondamento degli spigoli

L'arrotondamento degli spigoli con gli utensili standard viene definito con il parametro utensile \$TC_DP7. Dal parametro utensile \$TC_DP16 si ricava la differenza dell'arrotondamento dello spigolo dell'utensile reale rispetto all'utensile standard.

Esempio

Dimensioni di una fresa toroidale con raggio ridotto rispetto ad un utensile standard.

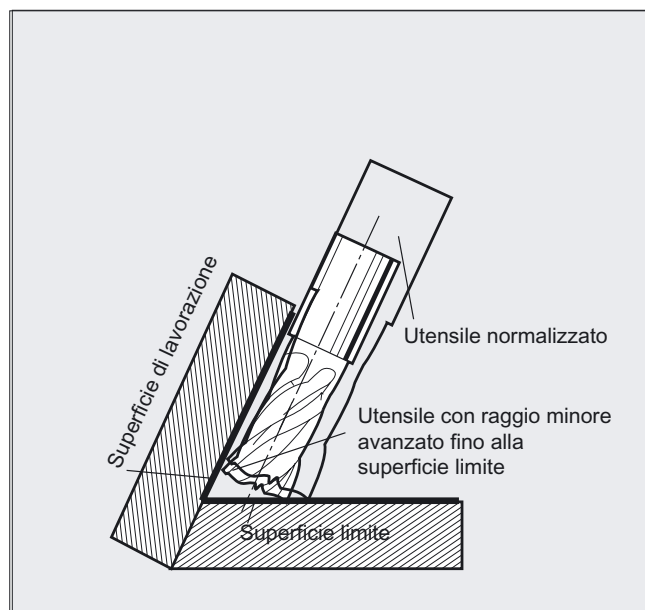
Tipo di utensile	R = raggio del codolo	r = raggio dello spigolo
Utensile standard con arrotondamento degli spigoli	$R = \$TC\_DP6$	$r = \$TC\_DP7$
Utensile reale con arrotondamento degli spigoli: Tipi di utensile 121 e 131, fresa toroidale (fresa a codolo)	$R' = \$TC\_DP6 + \$TC\_DP15 + OFFN$	$r' = \$TC\_DP7 + \$TC\_DP16$
In questo esempio sia \$TC_DP15 + OFFN che \$TC_DP16 sono negativi. Il tipo di utensile (\$TC_DP1) viene considerato.		

Tipo di utensile	R = raggio del codolo	r = raggio dello spigolo
Sono consentiti solo tipi di frese con codolo cilindrico (fresa cilindrica o a codolo) oppure frese toroidali (tipi 121 e 131) e al limite frese cilindriche per stampi (tipo 110).	Per questi tipi di frese consentiti, il raggio dello spigolo r è uguale al raggio del codolo R . Tutti gli altri tipi di utensili consentiti vengono interpretati come frese cilindriche ed un eventuale dimensione per l'arrotondamento dello spigolo non viene considerata.	
Sono consentiti tutti i tipi di utensili di numeri da 1 a 399 ad eccezione dei numeri 111 e da 155 a 157.		

Descrizione

Traiettoria del centro utensile con incremento fino alla superficie limite CUT3DCCD

Se viene utilizzato un utensile che, rispetto agli utensili standard possiede un raggio inferiore, l'incremento della fresa in direzione longitudinale avviene finché questa non tocca il fondo della tasca. In questo modo, lo spigolo formato dalla superficie di lavorazione e da quella limite, viene asportato finché l'utensile lo consente. Si tratta quindi di un tipo di lavorazione misto tra fresatura periferica e frontale. Analogamente ad un utensile con raggio ridotto, nel caso di utensili con raggio maggiore, l'incremento avviene nella direzione opposta.



Rispetto a tutte le altre correzioni utensili del gruppo 22 di codici G, un parametro utensile $\$TC\_DP6$ per CUT3DCCD non ha alcun significato per il raggio dell'utensile e non influenza la correzione risultante.

L'offset di correzione si ricava dalla somma del

- valore di usura del raggio utensile (parametro utensile $\$TC\_DP15$)

e da un offset utensile programmato per il calcolo dell'offset ortogonale

- alla superficie limite `OFFN`.

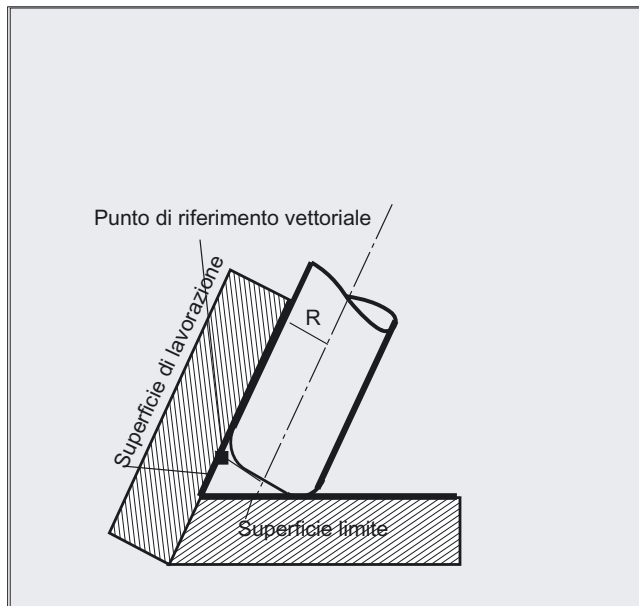
Dal partprogram non si può dedurre se la superficie da lavorare si trova sulla destra o sulla sinistra della traiettoria. Si parte quindi da un raggio positivo ed un valore di usura negativo dell'utensile originale. Un valore di usura negativo descrive sempre un utensile con diametro ridotto.

Utilizzo di utensili cilindrici

Utilizzando utensili cilindrici, l'incremento è solo necessario quando la superficie da lavorare e la superficie limite formano un angolo acuto (minore di 90 gradi). Se vengono utilizzate frese toroidali (cilindro con spigoli arrotondati), è necessario un incremento in direzione longitudinale sia nel caso di angoli acuti sia nel caso di angoli ottusi.

Correzione raggio 3D con CUT3DCC, profilo della superficie da lavorare

Se viene attivato `CUT3DCC` con una fresa toroidale, la traiettoria programmata si riferisce ad una fresa cilindrica dello stesso diametro. Il punto di riferimento della traiettoria risultante, utilizzando una fresa toroidale, viene rappresentato nella figura seguente.



È consentito che l'angolo tra la superficie limite e quella di lavorazione, anche nell'ambito di un blocco, passi da un angolo acuto ad uno ottuso o viceversa.

L'utensile reale utilizzato può essere sia più grande che più piccolo rispetto ad un utensile normalizzato. Il raggio risultante dello spigolo non può diventare quindi negativo ed il segno del raggio risultante dell'utensile deve essere mantenuto.

Con `CUT3DCC` il partprogram NC si riferisce al profilo della superficie da lavorare. Come nel caso della correzione raggio utensile tradizionale, il raggio globale deriva dalla somma del

- raggio dell'utensile (parametro utensile `$TC_DP6`)
- valore di usura (parametro utensile `$TC_DP15`)

e da un offset utensile programmato per il calcolo dell'offset ortogonale

- alla superficie limite OFFN.

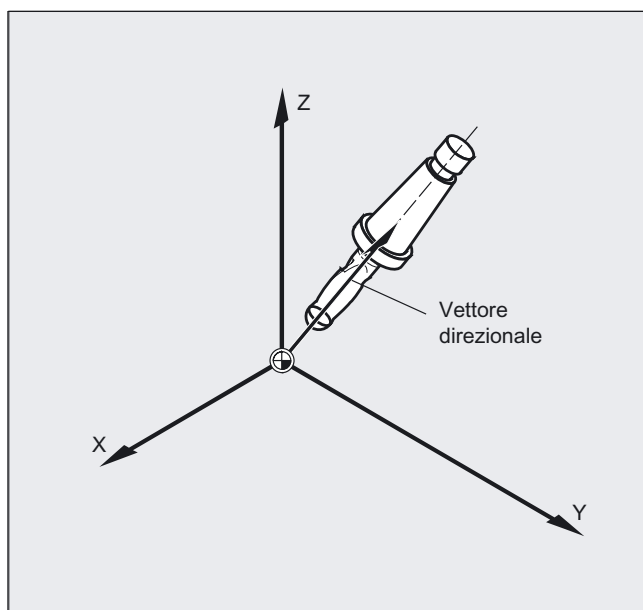
La posizione della superficie limite viene determinata dalla differenza di entrambi i valori

- Dimensioni dell'utensile standard e
- raggio dell'utensile (parametro utensile \$TC_DP6).

8.6 Orientamento dell'utensile (ORIC, ORID, OSOF, OSC, OSS, OSSE, OSD, OST)

Funzione

Per orientamento dell'utensile si intende la posizione geometrica dell'utensile nello spazio. Con una macchina per la lavorazione a 5 assi, l'orientamento dell'utensile può essere definito mediante istruzioni nel programma.



I movimenti di raccordo dell'orientamento attivati con OSD e OST vengono formati in modo diverso a seconda del tipo di interpolazione per l'orientamento utensile.

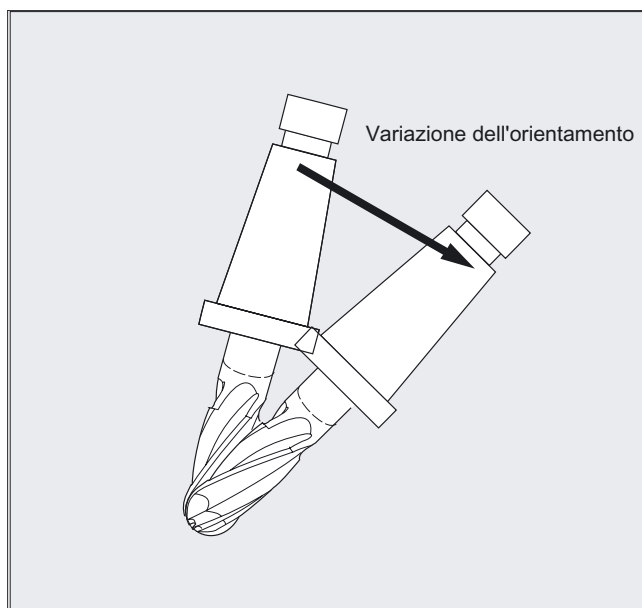
Quando è attiva l'interpolazione vettoriale, l'andamento livellato dell'orientamento viene interpolato anche vettorialmente. Quando invece è attiva l'interpolazione dell'asse rotante, l'orientamento viene livellato direttamente tramite i movimenti dell'asse rotante.

Programmazione

Una variazione dell'orientamento utensile può essere programmata con:

- Programmazione diretta degli assi rotanti A, B, C (interpolazione asse rotante)
- Angolo di Eulero oppure angolo RPY
- Vettore direzionale (interpolazione vettoriale con indicazione di A3 o B3 o C3)
- LEAD/TILT (fresatura frontale)

Il sistema di coordinate di riferimento è il sistema di coordinate macchina (ORIMKS) oppure il sistema di coordinate pezzo attuale (ORIWKS).



Parametri

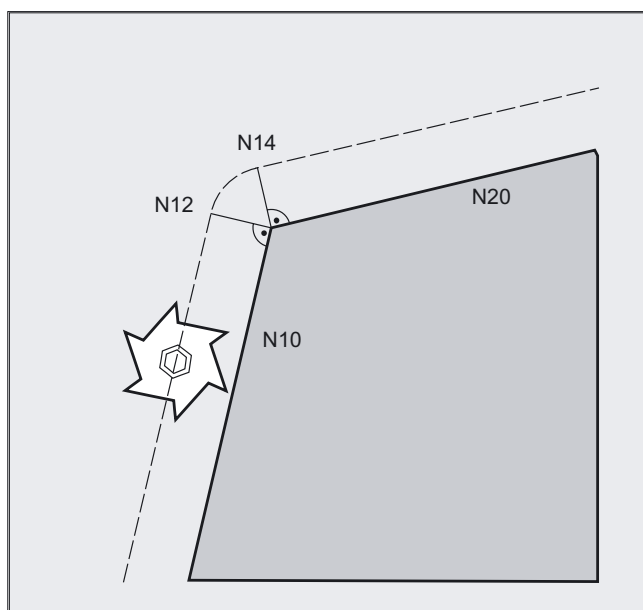
ORIC	Orientamento e movimento vettoriale paralleli
ORID	Orientamento e movimento vettoriale successivi
OSOF	Nessuno smorzamento dell'orientamento
OSC	Orientamento costante
OSS	Smorzamento dell'orientamento solo all'inizio del blocco
OSSE	Smorzamento dell'orientamento all'inizio ed alla fine del blocco
ORIS	La velocità di variazione dell'orientamento con smorzamento dell'orientamento attiva in gradi per mm, vale per OSS e OSSE
OSD	Raccordo del movimento dell'orientamento con l'impostazione della lunghezza di raccordo con il DS \$SC_ORI_SMOOTH_DIST.

OST

Raccordo del movimento di orientamento con l'impostazione della tolleranza d'angolo in gradi in caso di interpolazione vettoriale con il DS \$SC_ORI_SMOOTH_TOL. In caso di interpolazione dell'asse rotante, la tolleranza impostata viene considerata lo scostamento massimo degli assi di orientamento.

Esempio ORIC

Se tra i blocchi N10 e N20 sono stati programmati due o più blocchi con variazioni dell'orientamento (es. A2= B2= C2=) e ORIC è attivo, il movimento circolare inserito viene suddiviso in blocchi intermedi in funzione dell'ammontare delle variazioni dell'angolo.



ORIC

N8 A2=... B2=... C2=...

N10 X... Y... Z...

N12 C2=... B2=...

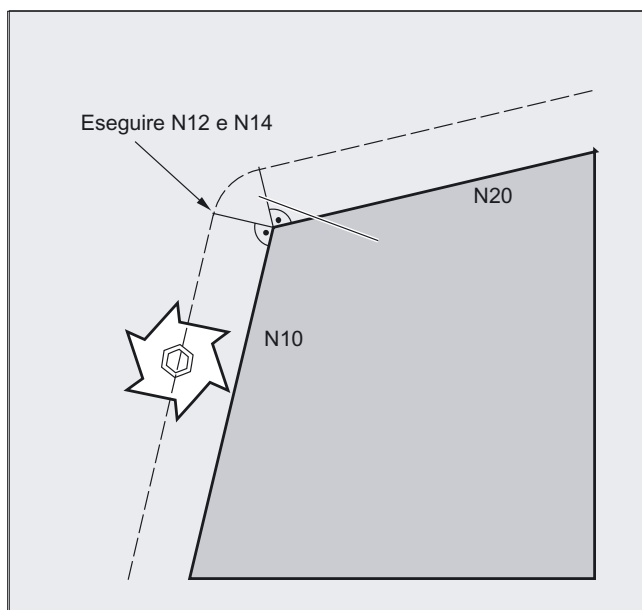
N14 C2=... B2=...

N20 X =...Y=... Z=... G1 F200

;Il blocco circolare, che viene inserito sullo spigolo esterno, si suddivide su N12 e N14 in funzione delle variazioni dell'orientamento. Il movimento circolare e la variazione dell'orientamento vengono eseguiti parallelamente.

Esempio ORID

Se ORID è attivo, tutti i blocchi tra i due di movimento, vengono eseguiti alla fine del primo blocco di movimento. Il blocco circolare con orientamento costante viene sempre eseguito prima del secondo blocco di movimento.



ORID

N8 A2=... B2=... C2=...

N10 X... Y... Z...

N12 A2=... B2=... C2=...

N14 M20

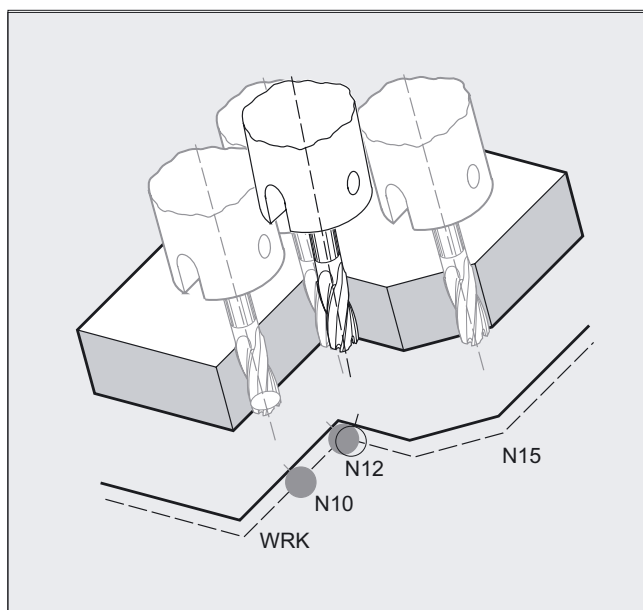
N20 X... Y... Z...

;I blocchi N12 e N14 vengono eseguiti alla fine di N10.
;Successivamente viene eseguito il blocco circolare con
;l'orientamento attuale.
;Funzioni ausiliarie, etc.

Nota

Per il tipo di variazione dell'orientamento su uno spigolo esterno, è determinante l'istruzione di programmazione attiva nel primo blocco di movimento di uno spigolo esterno

Senza variazione dell'orientamento: Se l'orientamento non viene modificato alla fine del blocco, la sezione trasversale dell'utensile è un cerchio che tocca entrambi i profili

Esempio di variazione dell'orientamento in uno spigolo interno

```

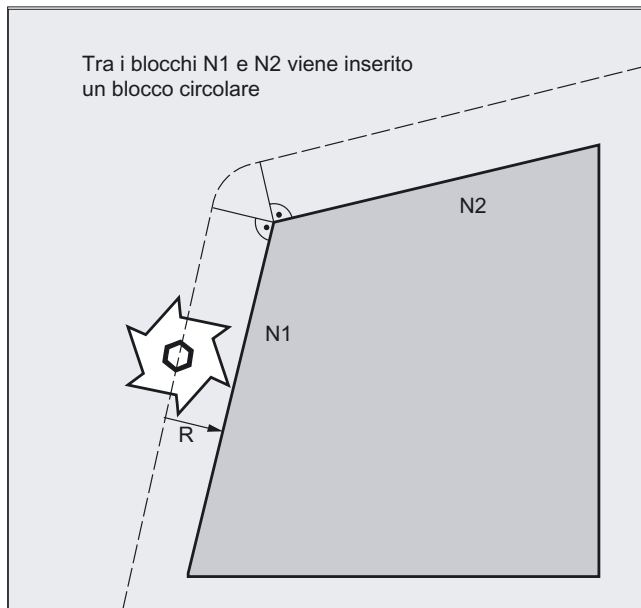
| ORIC
| N10 X ...Y... Z... G1 F500
| N12 X ...Y... Z... A2=... B2=..., C2=...
| N15 X Y Z A2 B2 C2

```

Comportamento sugli spigoli esterni

Su uno spigolo esterno viene sempre inserito un blocco circolare con il raggio della fresa.

Con le istruzioni di programmazione ORIC oppure ORID si può definire se le variazioni dell'orientamento, programmate tra i blocchi N1 e N2, devono essere eseguite prima dell'inizio o contemporaneamente al blocco circolare inserito.



Se è necessaria una variazione dell'orientamento su uno spigolo esterno, questa può avvenire sia parallelamente all'interpolazione che separatamente con il movimento vettoriale.

Con **ORID** vengono prima eseguiti i blocchi inseriti senza movimento vettoriale. Il blocco circolare viene sempre inserito prima del secondo blocco di movimento, attraverso il quale viene formato lo spigolo.

Se su uno spigolo esterno vengono inseriti più blocchi di orientamento e **ORIC** è attivo, il movimento circolare viene suddiviso in funzione dell'ammontare delle variazioni dell'orientamento dei singoli

Raccordo dell'orientamento con OSD e OST

In caso di raccordo con G642 lo scostamento massimo per gli assi di profilo e gli assi di orientamento non può essere molto diverso. Questa tolleranza minore consente di livellare in modo relativamente forte

- il movimento di raccordo o la tolleranza d'angolo,
- il percorso dell'orientamento

senza dover incorrere in scostamenti significativi del profilo.

Attivando **OSD** e **OST** è possibile, con una lunghezza di raccordo e una tolleranza d'angolo preimpostate, livellare ampiamente scostamenti molto ridotti del percorso di orientamento senza provocare gravi scostamenti dal profilo.

Nota

A differenza del raccordo del profilo (e del percorso di orientamento) con G642, nel raccordo dell'orientamento con OSD o OST non viene formato un blocco a sé stante, bensì il movimento di raccordo viene inserito direttamente nei blocchi originali programmati.

Con OSD o OST non si possono raccordare passaggi di blocchi nei quali avviene un cambio del tipo di interpolazione per l'orientamento dell'utensile (vettore → asse rotante, asse rotante → vettore). Questi passaggi di blocco possono essere eventualmente raccordati con le funzioni di raccordatura tradizionali G641, G642 o G643.

8.7 Assegnazione di un numero D libero, numero del tagliente

8.7.1 Assegnazione di un numero D libero, numero del tagliente (indirizzo CE)

Funzione

I numeri D possono essere utilizzati come numeri di correttori. Inoltre con l'indirizzo CE si può indirizzare il numero del tagliente. Con la variabile di sistema \$TC_DPCE è possibile rappresentare il numero del tagliente.

Preimpostazione: n. correttore == n. tagliente

Bibliografia:

/FB1/ Descrizione delle funzioni di base; Correzione utensile (W1).

Costruttore della macchina

Tramite i dati macchina vengono definiti la quantità massima di numeri D (numero dei taglienti) e dei taglienti per ogni utensile. Le seguenti istruzioni sono utilizzabili solo se il numero massimo dei taglienti (DM 18105) è stato definito più grande del numero di taglienti per utensile (DM 18106). Tenere in considerazione i dati del costruttore della macchina.

Nota

Oltre all'assegnazione relativa dei numeri D, gli stessi si possono impostare come 'piani' oppure 'assoluti' (1...32000) senza riferimento ad un numero T (nell'ambito della funzione 'struttura del numero D piana').

8.7.2 Verifica del numero D (CHKDNO)

Funzione

Con CKKDNO si verifica se i numeri D esistenti sono stati impostati in modo univoco. I numeri D di tutti gli utensili definiti nell'ambito di un'unità TO, possono essere assegnati solo una volta. Gli utensili sostitutivi non vengono verificati.

Programmazione

```
state=CHKDNO (Tno1, Tno2, Dno)
```

Parametri

state	TRUE: i numeri D del settore verificato, sono stati assegnati in modo univoco. FALSE: avviene una collisione tra i numeri D oppure la parametrizzazione è errata. I parametri che provocano la collisione vengono assegnati Con Tno1, Tno2 e Dno. Questi dati possono essere analizzati nel partprogram.
CHKDNO (Tno1, Tno2)	Vengono verificati tutti i numeri D dell'utensile indicato.
CHKDNO (Tno1)	Vengono verificati tutti i numeri D di Tno1 rispetto agli altri utensili.
CHKDNO	Vengono verificati tutti i numeri D di tutti gli utensili rispetto agli altri utensili.

8.7.3 Rinominare i numeri D (GETDNO, SETDNO)

Funzione

I numeri D devono essere sempre definiti in modo univoco. Due differenti taglienti di un utensile non possono avere lo stesso numero D.

GETDNO

Questa istruzione fornisce un numero D di un determinato tagliente (ce) di un utensile con il numero T t. Se per il parametro indicato non esiste nessun numero D, viene settato d=0. Se il numero D non è valido, viene restituito un valore > 32000.

SETDNO

Con questa istruzione viene assegnato il valore d al numero D di un tagliente ce dell'utensile t. Con state viene restituito il risultato di questa istruzione (TRUE oppure FALSE). Se per il parametro indicato non esiste nessun blocco dati, viene restituito FALSE. Gli errori di sintassi generano un allarme. Il numero D non può essere settato a 0 in modo esplicito.

Programmazione

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Parametri

d	Numero D del tagliente dell'utensile
t	Numero T dell'utensile
ce	Numero del tagliente (numero CE) dell'utensile
state	Indica se il comando è stato eseguito senza errori (TRUE oppure FALSE).

Esempio di ridenominazione di un numero D

```
$TC_DP2 [1,2] = 120
$TC_DP3 [1,2] = 5.5
$TC_DPCE [1,2] = 3; numero tagliente CE
...
N10 def int NrDVecchio, NrDNuovo = 17
N20 NrDVecchio = GETDNO (1,3)
N30 SETDNO (1,3,NrDNuovo)
```

In questo modo al tagliente CE=3 viene assegnato il nuovo valore D 17. Ora i dati di questo tagliente vengono interrogati tramite il numero D 17; sia con le variabili di sistema che da programmazione con l'indirizzo NC.

8.7.4 Determinazione del numero T per il numero D indicato (GETACTTD)**Funzione**

Con GETACTTD viene segnalato il relativo numero T ad un numero D assoluto. Non avviene nessuna verifica dell'univocità. Se nell'ambito di un'unità TO sono presenti più numeri D identici, viene restituito il numero T del primo utensile trovato. Utilizzando i numeri D 'piani', l'utilizzo di questa istruzione non è sensato in quanto verrebbe restituito sempre il numero 1 (nessun numero T nel mantenimento dati).

Programmazione

```
status = GETACTTD (Tnr, Dnr)
```

Parametri

Dnr	Numero D per il quale deve essere ricercato il numero T.
Tnr	Numero T trovato
stato	0: Il numero T è stato trovato. Tnr contiene il valore del numero T. -1: per il numero D indicato non esiste nessun numero T; Tnr=0. -2: il numero D non è assoluto. Tnr contiene il valore del primo utensile trovato contenente il numero D con il valore Dnr. -5: non è stato possibile eseguire la funzione per un altro motivo.

8.7.5 Definizione di numero D come non valido (DZERO)**Funzione**

L'istruzione serve da supporto durante il riattrezzaggio. I blocchi dati di correzione contraddistinti in questo modo non vengono verificati dall'istruzione CHKDNO. Per poterli rendere nuovamente gestibili, i numeri D devono essere settati nuovamente con SETDNO

Programmazione

DZERO

Parametri

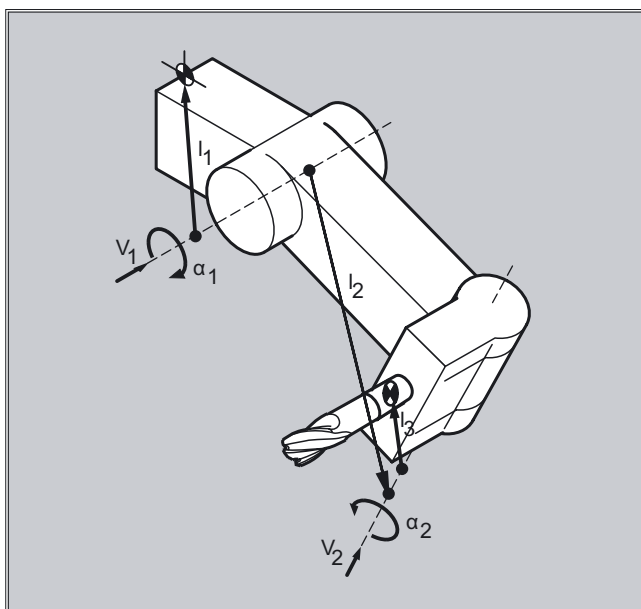
DZERO	Contraddistingue tutti i numeri D dell'unità TO come non validi

8.8 Cinematica del portautensile

Funzione

La cinematica del portautensile con al massimo due assi rotanti v_1 oppure v_2 viene definita attraverso le 17 variabili $\$TC\_CARR1 [m] \dots \$TC\_CARR17 [m]$. La definizione del portautensile è composta:

- dalla distanza vettoriale del primo asse rotante rispetto al punto di riferimento del portautensile I_1 , dalla distanza vettoriale del primo rispetto al secondo asse rotante I_2 , dalla distanza vettoriale del secondo asse rotante rispetto al punto di riferimento dell'utensile I_3 .
- dei vettori direzionali di entrambi gli assi rotanti V_1 , V_2 .
- dagli angoli di rotazione α_1 , α_2 intorno ad entrambi gli assi. Gli angoli di rotazione vengono conteggiati osservando in direzione dei vettori degli assi rotanti, in senso orario positivo.



Per macchine con **cinematica risolta** (sia utensili che pezzi possono essere ruotati), le variabili di sistema vengono ampliate con le registrazioni

- $\$TC\_CARR18 [m] \dots \$TC\_CARR23 [m]$.

Parametri

Funzione delle variabili di sistema per il portautensile orientabile			
Identificazione	Componente x	Componente y	Componente z
I_1 Vettore offset	$\$TC\_CARR1[m]$	$\$TC\_CARR2[m]$	$\$TC\_CARR3[m]$
I_2 Vettore offset	$\$TC\_CARR4[m]$	$\$TC\_CARR5[m]$	$\$TC\_CARR6[m]$
v_1 Asse rotante	$\$TC\_CARR7[m]$	$\$TC\_CARR8[m]$	$\$TC\_CARR9[m]$
v_2 Asse rotante	$\$TC\_CARR10[m]$	$\$TC\_CARR11[m]$	$\$TC\_CARR12[m]$

Funzione delle variabili di sistema per il portautensile orientabile			
α_1 Angolo di rotazione	\$TC_CARR13[m]		
α_2 Angolo di rotazione	\$TC_CARR14[m]		
l_3 Vettore offset	\$TC_CARR15[m]	\$TC_CARR16[m]	\$TC_CARR17[m]

Ampliamento delle variabili di sistema per il portautensile orientabile			
Identificazione	Componente x	Componente y	Componente z
l_4 Vettore offset	\$TC_CARR18[m]	\$TC_CARR19[m]	\$TC_CARR20[m]
Identificatori degli assi rotanti v_1 e v_2	Identificatori degli assi rotanti v_1 e v_2 (default = 0) \$TC_CARR21[m] \$TC_CARR22[m]		
Tipo di cinematica	\$TC_CARR23[m]		
Tool	Tipo di cinematica T ->	Tipo di cinematica P ->	Tipo di cinematica M
Part	Solo l'utensile può essere ruotato (default)	Solo il pezzo può essere ruotato.	Pezzo e utensile possono essere ruotati
Mixed mode			
Identificatori degli assi rotanti v_1 e v_2	Angolo in gradi degli assi rotanti v_1 e v_2 assumendo la posizione base \$TC_CARR21[m] \$TC_CARR22[m]		
Offset angolare degli assi rotanti v_1 e v_2	Offset della dentatura Hirth in gradi degli assi rotanti v_1 e v_2 \$TC_CARR26[m] \$TC_CARR27[m]		
Incremento angolare v_1 asse rotante v_2 asse rotante	Incremento della dentatura Hirth in gradi degli assi rotanti v_1 e v_2 \$TC_CARR28[m] \$TC_CARR29[m]		
Posiz.minima asse rotante v_1 asse rotante v_2	Limite software per la posizione minima degli assi rotanti v_1 e v_2 \$TC_CARR30[m] \$TC_CARR31[m]		
Posiz.massima asse rotante v_1 asse rotante v_2	Limite software per la posizione massima degli assi rotanti v_1 e v_2 \$TC_CARR32[m] \$TC_CARR33[m]		
Nome del portautensile	Il portautensile può essere definito con il nome anziché con un numero. \$TC_CARR34[m]		
Utilizzatore: nome asse 1 nome asse 2 codice	Utilizzo previsto nell'ambito dei cicli di misura dell'utente: \$TC_CARR35[m] \$TC_CARR36[m] \$TC_CARR37[m]		
Posizione	\$TC_CARR38[m]	\$TC_CARR39[m]	\$TC_CARR40[m]
Traslazione fine	Parametro che può essere sommato ai valori nei parametri di base.		
l_1 Vettore offset	\$TC_CARR41[m]	\$TC_CARR42[m]	\$TC_CARR43[m]
l_2 Vettore offset	\$TC_CARR44[m]	\$TC_CARR45[m]	\$TC_CARR46[m]
l_3 Vettore offset	\$TC_CARR55[m]	\$TC_CARR56[m]	\$TC_CARR57[m]
l_4 Vettore offset	\$TC_CARR58[m]	\$TC_CARR59[m]	\$TC_CARR60[m]
v_1 Asse rotante	\$TC_CARR64[m]		
v_2 Asse rotante	\$TC_CARR65[m]		

Ampliamento delle variabili di sistema per il portautensile orientabile

Nota

Chiarimenti sui parametri

Con "m" viene indicato il numero del portautensile da definire.

\$TC_CARR47 ... \$TC_CARR54 e \$TC_CARR61 ... \$TC_CARR63 non sono definiti e tentando di leggerli o scriverli viene emesso un allarme.

I punti iniziali e finali dei vettori della distanza riferita agli assi, possono essere scelti liberamente. Gli angoli di rotazione α_1 , α_2 intorno ad entrambi gli assi, vengono impostati a 0° nella posizione di riposo del portautensile. La cinematica di un portautensile, in questo modo, può essere definita liberamente.

Portautensili con solo uno o nessun asse rotante, possono essere definiti ad uno o ad entrambi gli assi rotanti settando a zero i vettori direzionali.

Nel caso di un portautensile senza asse rotante, i vettori della distanza hanno l'effetto di correttori utensili supplementari i cui componenti non vengono influenzati durante la commutazione dei piani di lavoro ($G17 \dots G19$)°.

Ampliamenti dei parametri

Parametri degli assi rotanti \$TC_CARR24 ... \$TC_CARR33

Le variabili di sistema sono state ampliate con le registrazioni \$TC_CARR24[m] ... \$TC_CARR33[m] e vengono descritte come segue:

L'Offset degli assi rotanti v_1 , v_2	Variazione della posizione degli assi rotanti v_1 oppure v_2 nella posizione di base del portautensile orientabile.
Gli offset angolari/incrementi angolari degli assi rotanti v_1 , v_2	Offset oppure incremento angolare della dentatura Hirth degli assi rotanti v_1 e v_2 . L'angolo programmato o calcolato viene arrotondato al valore più vicino derivante dall'intero n da $\phi = s + n * d$.
Posiz.minima e massima assi rotanti v_1 , v_2	La posizione minima e massima dell'asse rotante Angolo limite (limite software) degli assi rotanti v_1 e v_2 .

Parametri per l'utente \$TC_CARR34 ... \$TC_CARR40 contengono parametri

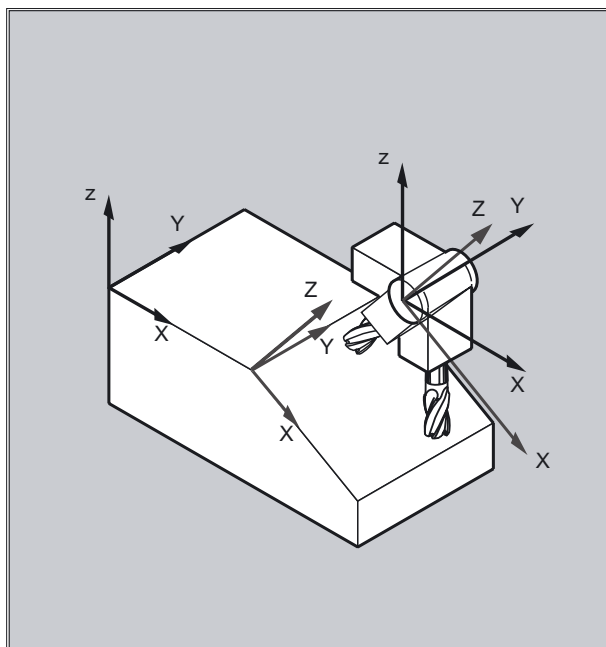
Utente	disponibili per l'utente che fino al SW 6.4 come default non vengono utilizzati nell'NCK e non hanno alcun significato.

Parametri della traslazione fine \$TC_CARR41 bis \$TC_CARR65 contengono

Traslazione fine	parametri per la traslazione fine che possono essere sommati ai valori nei parametri di base Il valore di traslazione fine abbinato ad un parametro di base si ricava sommando il valore 40 al numero del parametro.

Esempio

Il portautensile utilizzato nel seguente esempio viene definito completamente con una rotazione intorno all'asse Y.



N10 \$TC_CARR8[1]=1	;Definizione della componente Y del primo asse ;rotante del portautensile 1
N20 \$TC_DP1[1,1]= 120	;Definizione di una fresa a codolo
N30 \$TC_DP3[1,1]=20	;Definizione di una fresa a codolo con ;lunghezza 20 mm
N40 \$TC_DP6[1,1]=5	;Definizione di una fresa a codolo con ;raggio 5 mm
N50 ROT Y37	;Definizione del Frame con rotazione di 37° ;intorno all'asse Y
N60 X0 Y0 Z0 F10000	;Accostamento alla posizione iniziale
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	;Impostazione della correzione raggio, ;correzione lunghezza utensile con Frame ruotato, ;selezionare portautensile 1, utensile 1
N80 X40	;Esecuzione della lavorazione con una ;rotazione di 37°
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Premessa

Un portautensile può orientare un utensile in tutte le posizioni possibili nello spazio solo se

- sono presenti due assi rotanti V_1 e V_2 .
- i due assi rotanti si trovano in posizione ortogonale l'uno rispetto all'altro.
- l'asse longitudinale dell'utensile è ortogonale al secondo asse rotante V_2 .

Inoltre, per le macchine nelle quali devono essere impostabili tutti gli orientamenti possibili, è necessario che:

- l'orientamento utensile deve essere ortogonale al primo asse rotante V_1 .

Descrizione

Cinematica risolta

Per macchine con cinematica risolta (sia utensili che pezzi possono essere ruotati), le variabili di sistema vengono ampliate con le registrazioni `$TC_CARR18 [m]` bis `$TC_CARR23 [m]` e descritte nel seguente modo:

La tavola utensile girevole si compone:

- della distanza vettoriale del secondo asse rotante V_2 rispetto al punto di riferimento di una tavola utensile girevole I_4 del terzo asse rotante.

Gli assi rotanti si compongono:

- di entrambi gli identificatori dei canali per il riferimento degli assi rotanti V_1 e V_2 , la cui posizione viene utilizzata eventualmente per la determinazione dell'orientamento del portautensile orientabile.

Il tipo di cinematica con uno dei valori T, P oppure M:

- tipo di cinematica T: solo l'utensile può essere ruotato.
- tipo di cinematica P: solo il pezzo può essere ruotato.
- tipo di cinematica M: utensile e pezzo possono essere entrambi ruotati.

Cancellazione dei dati del portautensile

Con `$TC_CARR1 [0] = 0` si possono cancellare tutti i blocchi dati del portautensile.

Il tipo di cinematica `$TC_CARR23 [T] = T` deve essere definito con una delle tre lettere minuscole o maiuscole consentite (T,P,M) e, per questo motivo, non deve essere cancellato.

Modifica dei dati del portautensile

Ognuno dei valori definiti può essere modificato assegnando un nuovo valore nel partprogram. Ogni altra lettera diversa da T, P oppure M, comporta l'emissione di un allarme al momento dell'attivazione del portautensile orientabile.

Lettura dei dati del portautensile

Ognuno dei valori definiti può essere letto assegnandolo ad una variabile nel partprogram.

Traslazioni fini

Un valore di traslazione fine errato viene riconosciuto solo all'attivazione di un portautensile orientabile che contiene questo valore e contemporaneamente il dato di setting DS 42974 :
TOCARR_FINE_CORRECTION = TRUE .

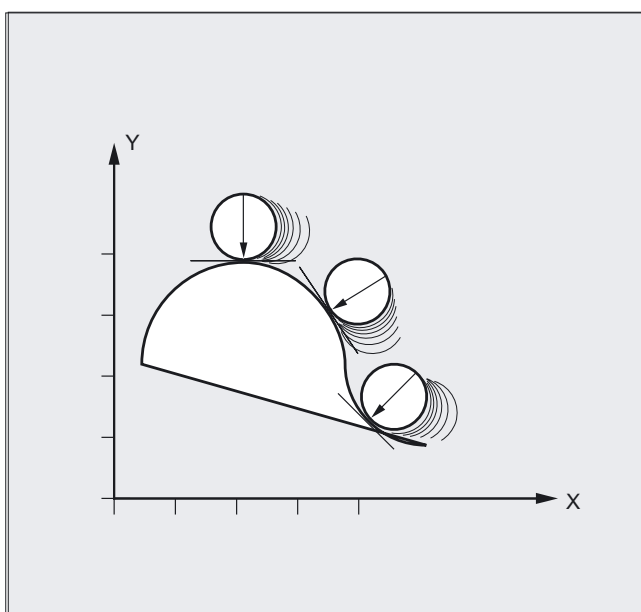
La somma delle traslazioni fine consentite viene limitata ad un valore massimo tramite dati macchina.

Comportamento sulla traiettoria

9.1 Controllo tangenziale (TANG, TANGON, TANGOF, TANGDEL)

Funzione

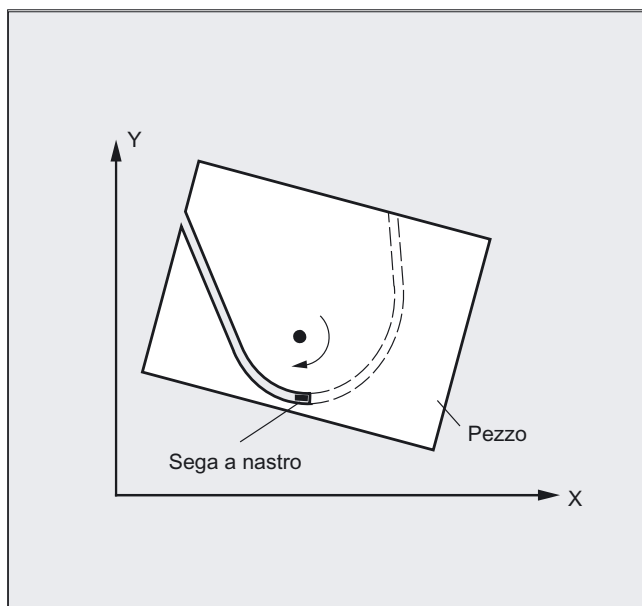
L'asse slave viene gestito a seguire in funzione della tangente alla traiettoria definita attraverso l'asse master. In questo modo un utensile viene orientato parallelamente al profilo. Con l'angolo programmato con l'istruzione TANGON, l'utensile può essere posizionato relativo alla tangente.



Campi di impiego

Il controllo tangenziale può essere tra l'altro utilizzato per:

- controllo tangenziale di un utensile rotativo per la roditura
- gestione dell'attrezzo portapezzo per una sega a nastro (vedere disegno)
- posizionamento di un utensile di diamantatura su una mola
- posizionamento di un diamante circolare per il taglio del vetro o per la lavorazione della carta
- posizionamento tangenziale del filo per la saldatura su 5 assi



Programmazione

TANG (FAsse, LAssel, LAssel2, Accopp., KS, Opt.)

oppure

TANGON (FAsse, Angolo, Dist, Angolotol)

oppure

TANGDEL (FAsse)

oppure

TLIFT (FAsse)

oppure

TANGDEL (FAsse)

Programmazioni semplificate:

Un fattore di accoppiamento di 1 non deve essere obbligatoriamente programmato in modo esplicito.

TANG(C, X, Y, 1, "B", "P") può essere abbreviato come TANG(C, X, Y, , , , "P"). Come finora, invece di TANG(C, X, Y, 1, "B", "S") si può scrivere TANG(C, X, Y).

L'istruzione TLIFT(...) deve essere impostata insieme all'abbinamento degli assi con TANG(...). Esempio:

TANG(C, X, Y...)

TLIFT(C)

Disattivazione di TLIFT

Ripetere a tal fine l'abbinamento degli assi TANG(...) senza il successivo TLIFT(...).

TANGDEL Cancellazione della definizione di un funzionamento a seguire tangenziale

Un funzionamento a seguire tangenziale definito dall'utente dovrà essere cancellato, in caso si presenti la necessità di definire un nuovo inseguimento tangenziale con lo stesso asse a

9.1 Controllo tangenziale (TANG, TANGON, TANGOF, TANGDEL)

seguire nel richiamo TANG . La cancellazione è possibile solo se l'accoppiamento a TANGOF (asse F) è disabilitato.

Parametri

TANG	Istruzione preparatoria per la definizione del funzionamento a seguire tangenziale; preimpostazione: 1 TANG(C,X,Y,1,"B") significa: l'asse rotante C segue gli assi geometrici X e Y. Disattivazione di TLIFT
TANGON	Attivazione del controllo tangenziale con indicazione dell'asse slave e dell'angolo di offset desiderato per l'asse slave, eventualmente percorso di raccordo, deviazione angolare. TANGON(C,90) significa: L'asse C è un asse slave. Esso viene tenuto a 90° rispetto alla tangente sul punto di lavoro degli assi di contornatura.
TANGOF	Disattivazione del controllo tangenziale con indicazione dell'asse slave. Per escludere il controllo tangenziale dell'asse slave impostare: TANGOF(C)
TLIFT	Inserimento di blocchi intermedi sugli spigoli del profilo.
TANGDEL	Cancellazione della definizione di un funzionamento a seguire tangenziale. Esempio: TANGDEL (asse slave)
Asse slave	Asse slave: asse rotante supplementare in funzionamento a seguire tangenziale
AsseMaster1, asseMaster2	Assi master: assi di contornatura dai quali viene ricavata la tangente per il funzionamento a seguire
Accoppiamento	Fattore di accoppiamento: relazione tra variazione angolare della tangente ed asse slave. Indicazione opzionale; preimpostazione: 1
KS	Lettera identificativa per sistema di coordinate "B" = sistema di coordinate di base; indicazione opzionale; preimpostazione "W" = il sistema di coordinate del pezzo non è disponibile
Opt	Ottimizzazione: "S" Standard, Default "P" adeguamento automatico della caratteristica di tempo dell'asse tangenziale e del profilo
Angolo	Angolo di offset dell'asse slave
Dist	Percorso di raccordo dell'asse slave, necessario per Opt. "P"
Angolotol	Tolleranza angolare dell'asse slave, (opzionale), viene interpretato solo con Opt.="P"

Possibilità di ottimizzazione Opt, Dist e Winkeltol

Con Opt="P" viene sorvegliata la dinamica dell'asse slave al limite di velocità dell'asse master e deve essere utilizzata in primo luogo quando si utilizzano trasformazioni cinematiche.

I parametri (Dist e Winkeltol) limitano l'errore presente tra l'asse slave e la tangente dell'asse master.

Esempio di cambio di piano

N10 TANG(A, X, Y, 1)	;1. definizione del funz. a seguire tangenz.
N20 TANGON(A)	;Attivazione dell'accoppiamento
N30 X10 Y20	;raggio
...	
N80 TANGOF(A)	;Disattivazione del 1° accoppiamento
N90 TANGDEL(A)	;Cancellazione della 1° definizione
...	
TANG(A, X, Z)	;2. definizione del funz. a seguire tangenz.
TANGON(A)	;Attivazione del nuovo accoppiamento
...	
N200 M30	

Esempio di commutazione di assi geometrici e TANGDEL

Non viene generato alcun allarme.

N10 GEOAX(2, Y1)	;Y1 è l'asse geometrico 2
N20 TANG(A, X, Y)	
N30 TANGON(A, 90)	
N40 G2 F8000 X0 Y0 I0 J50	
N50 TANGOF(A)	;Disattivazione funzionam. a seguire con Y1
N60 TANGDEL(A)	;Cancellazione della 1° definizione
N70 GEOAX(2, Y2)	;Y2 è il nuovo asse geometrico 2
N80 TANG(A, X, Y)	;2. definizione del funz. a seguire tangenz.
N90 TANGON(A, 90)	;Attivazione del funz. a seguire con la 2° def.
...	

Esempio di inseguimento tangenziale con ottimizzazione automatica

Ottimizzazione automatica con Dist e tolleranza angolare

N80 G0 C0	;Y1 è l'asse geometrico 2
N100 F=50000	
N110 G1 X1000 Y500	
N120 TRAORI	;Raccordo con tolleranza assiale
N130 G642	
N171 TRANS X-Y-	;Ottimizzazione automatica della velocità vettoriale
N180 TANG(C, X, Y, 1, , "P")	;Percorso di raccordo 5 mm,
N190 TANGON(C, 0, 5.0, 2.0)	;Tolleranza angolare 2 gradi
N210 G1 X1310 Y500	;Attivazione del funz. a seguire con la 2° def.
N215 G1 X1420 Y500	
N220 G3 X1500 Y580 I=AC(1420)_ J=AC(580)	

```

N230 G1 X1500 Y760
N240 G3 X1360 Y900 I=AC(1360) _
      J=AC(760)
N250 G1 X1000 Y900
N280 TANGOF(C)
N290 TRAFOOF
N300 M02

```

Definizione dell'asse slave e dell'asse master

La definizione di asse master e di asse slave avviene con **TANG**.

Un fattore di accoppiamento fornisce la relazione tra una variazione angolare della tangente e l'asse slave a seguire. Il suo valore, in genere, è 1 (preimpostazione di default).

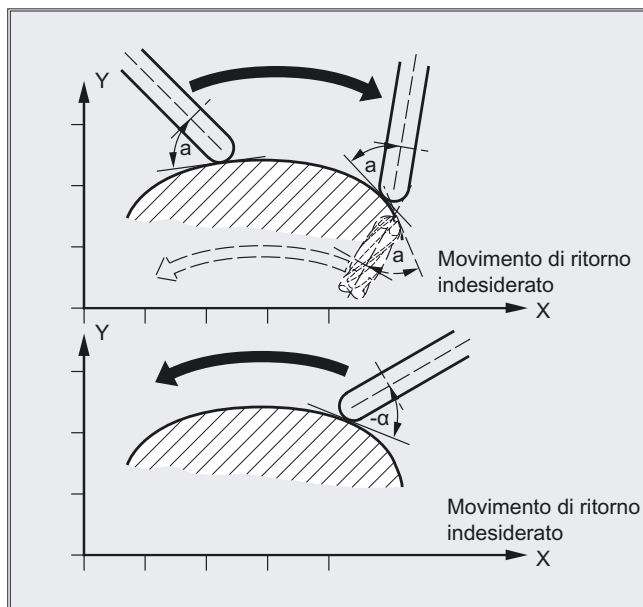
Angolo limite tramite limitazione del campo di lavoro

Per movimenti pendolari la tangente ruota di 180° sui punti di inversione del percorso e l'orientamento dell'asse slave varia di conseguenza.

Normalmente questo comportamento non è sensato: il movimento di ritorno deve essere eseguito con lo stesso angolo di offset negativo del movimento di andata.

Limitare pertanto il campo di lavoro dell'asse slave (G25, G26). La limitazione del campo di lavoro deve essere attiva al momento dell'inversione del percorso (WALIMON).

Se l'angolo di offset va oltre il limite del campo di lavoro, con un angolo di offset negativo si tenta di rimanere nel campo di lavoro consentito.



Inserimento di blocchi intermedi sugli spigoli del profilo, TLIFT

Su uno spigolo del profilo la tangente, e di conseguenza la posizione di riferimento dell'asse trascinato, sono soggette a variazioni brusche e irregolari. L'asse tenta normalmente di assorbire tali variazioni con la velocità massima possibile, causando tuttavia su un tratto del profilo uno scostamento rispetto alla posizione tangenziale desiderata. Se questo scostamento per motivi tecnologici non è ammissibile, con l'istruzione `TLIFT` si può fare in modo che il controllo si arresti in corrispondenza dello spigolo e ruoti l'asse trascinato nella nuova direzione tangenziale mediante un blocco intermedio creato automaticamente.

La rotazione avviene con l'asse di contornitura programmato quando l'asse slave è stato mosso una volta come asse di contornitura. Con la funzione `TFGREF[ax] = 0.001` si può raggiungere la velocità massima dell'asse slave.

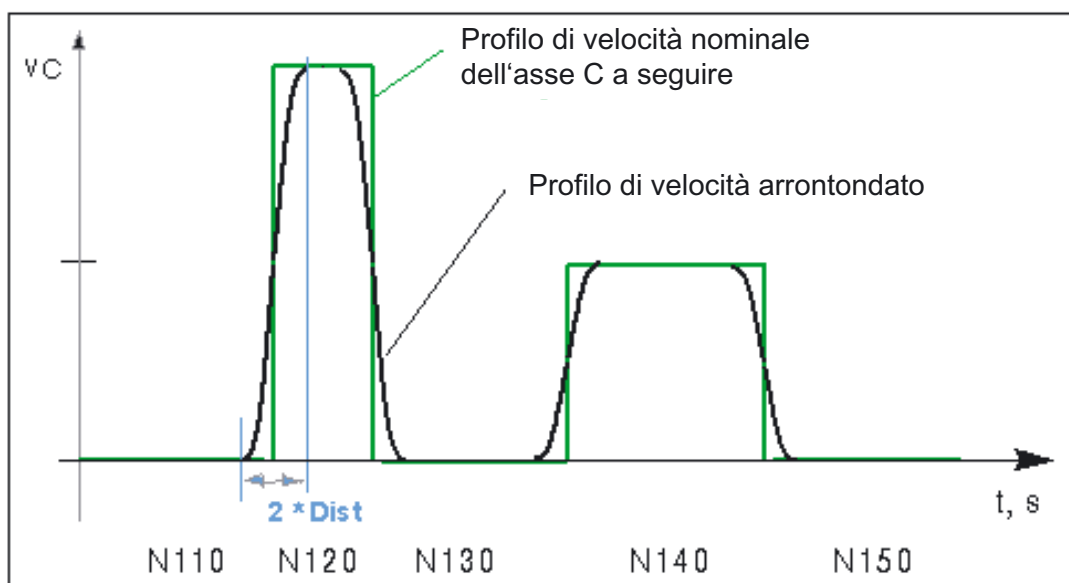
Se l'asse slave in precedenza non era mai stato mosso come asse di contornitura, esso viene gestito come asse di posizionamento. La velocità dipende quindi dall'impostazione dei dati macchina per la velocità di posizionamento.

La rotazione viene effettuata alla velocità massima dell'asse trascinato.

Possibilità di ottimizzazione

I gradini di velocità dell'asse slave, conseguenti ai gradini nel profilo dell'asse master, vengono smorzati o raccordati (Dist e Winkeltol).

In questo modo l'asse slave viene gestito "in anticipo" (vedere diagramma) per ridurre il più possibile gli scostamenti.



Definizione della variazione dell'angolo

La variazione dell'angolo a partire dalla quale viene inserito un blocco intermedio viene definita nel dato macchina `$MA_EPS_TLIFT_TANG_STEP`.

Influenza sulle trasformazioni

La posizione di un asse trascinato può essere il valore di ingresso per una trasformazione.

Posizionamento esplicito dell'asse slave

Se un asse slave viene posizionato esplicitamente, il dato di posizione viene sommato algebricamente all'angolo di offset.

Sono consentite tutte le impostazioni di percorso: movimenti degli assi di posizionamento e di contornitura.

Stato dell'accoppiamento

Nel partprogram dell'NC è possibile interrogare lo stato dell'accoppiamento con la seguente variabile di sistema:

`$AA_COUP_ACT[asse]`

0: Nessun accoppiamento attivo

1,2,3: Funzionamento a seguire tangenziale attivo

9.2 Trascinamento (TRAILON, TRAILOF)

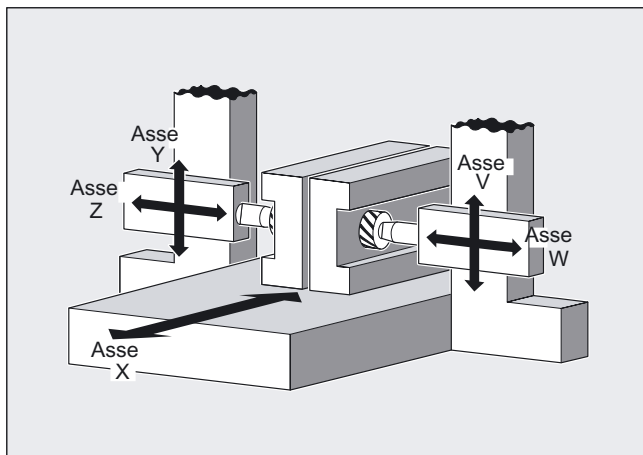
Funzione

Quando l'asse master muove, gli assi trascinati (= assi slave) ad esso abbinati, effettuano un movimento correlato considerando un fattore di accoppiamento.

Asse master ed assi slave costituiscono insieme un gruppo di trascinamento.

Campi di impiego

- Movimento di un asse attraverso un asse simulato. L'asse master è un asse simulato mentre l'asse trascinato è un asse reale. In questo modo l'asse reale può essere mosso considerando il fattore di accoppiamento.
- Lavorazione su due lati con 2 gruppi di trascinamento:
 1. asse master Y, asse trascinato V
 2. asse master Z, asse trascinato W



Programmazione

TRAILON(asse S, asse M, accopp.)

oppure

TRAILOF(asse S, asse M, asse M 2)

oppure disattivazione senza indicazione dell'asse master

TRAILOF(asse S)

TRAILON e TRAILOF hanno effetto modale.

Parametri

TRAILON	Attivazione e definizione del raggruppamento di trascinamento Esempio: V = asse trascinato, Y = asse master TRAILON(V,Y)
TRAILOF	Disattivazione del gruppo di trascinamento Esempio: V = asse trascinato, Y = asse master TRAILOF(V,Y) TRAILOF con 2 parametri disattiva soltanto l'accoppiamento a 1 asse master. Se un asse trascinato possiede 2 assi master, per es. V = asse trascinato e X, Y = assi master, per disattivare l'accoppiamento si può richiamare TRAILOF con 3 parametri: TRAILOF(V,X,Y) TRAILOF(V) Disattivazione dell'accoppiamento senza indicazione dell'asse master. Se l'asse trascinato possiede 2 assi master, entrambi gli accoppiamenti vengono disinseriti.
Asse F	Indicatore dell'asse slave Un asse trascinato può essere anche master di ulteriori assi trascinati. In questo modo è possibile formare diversi gruppi di trascinamento.
asse M	Indicatore dell'asse master

Accoppiamento

Fattore di accoppiamento = percorso dell'asse trascinato/percorso dell'asse master
preimpostazione = 1

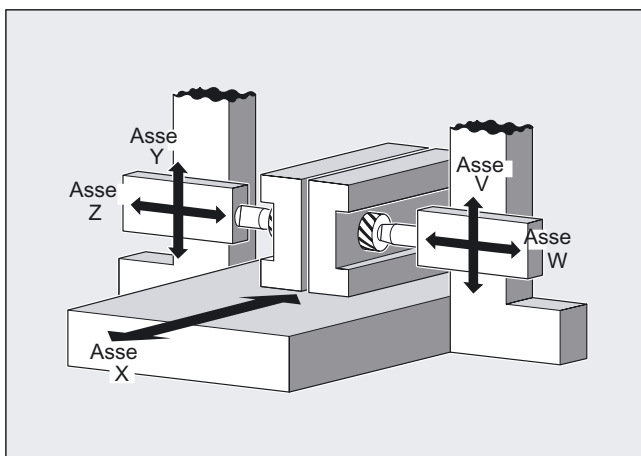
Nota

Il trascinamento avviene sempre nel sistema di coordinate base (SCB).

Il numero dei gruppi di trascinamento attivabili contemporaneamente viene limitato solo dalle possibili combinazioni degli assi presenti nella macchina.

Esempio

Il pezzo deve essere lavorato su due lati con la configurazione di assi qui rappresentata. Vengono formati 2 gruppi di trascinamento.



```

...
N100 TRAILON(V,Y)           ;Attivazione del 1. gruppo di trascinamento
N110 TRAILON(W,Z,-1)        ;Attivazione del 2. gruppo di trascinamento, fattore di accoppiamento
                             ;negativo: l'asse trascinato muove in senso opposto
                             ;all'asse master
N120 G0 Z10                 ;Posizionamento degli assi Z e W in
                             ;direzione opposta
N130 G0 Y20                 ;Posizionamento degli assi Y e V nella stessa direzione
...
N200 G1 Y22 V25 F200        ;Sovrapposizione di un movimento dipendente ed uno indipendente
                             ;dell'asse trascinato "V"
...
TRAILOF(V,Y)                ;Disattivazione del 1. gruppo di trascinamento
TRAILOF(W,Z)                ;Disattivazione del 2. gruppo di trascinamento

```

Tipi di assi nel gruppo

Un gruppo di trascinamento può essere composto da una combinazione qualsiasi di assi lineari e rotanti. Come asse master può essere definito anche un asse simulato.

Assi trascinati

Ad un asse trascinato possono essere assegnati al massimo 2 assi master. L'abbinamento avviene in diversi gruppi di trascinamento.

Un asse trascinato può essere programmato con tutti i comandi di movimento disponibili (G0, G1, G2, G3, ...). Oltre ai percorsi definiti in forma indipendente, l'asse di trascinamento esegue i percorsi derivati dai suoi assi master con i relativi fattori di accoppiamento.

fattore di accoppiamento

Il fattore di accoppiamento indica il rapporto desiderato tra i percorsi dell'asse trascinato e dell'asse master.

Formula: Fattore di accoppiamento = percorso dell'asse trascinato/percorso dell'asse master

Se nella programmazione non viene indicato il fattore di accoppiamento, viene considerato automaticamente il fattore 1.

Il fattore viene impostato come numero frazionario con punto decimale (tipo REAL).
L'impostazione di un valore negativo comporta un movimento opposto per asse master ed asse trascinato.

Accelerazione e velocità

I limiti di accelerazione e di velocità degli assi appartenenti all'accoppiamento vengono stabiliti dall'asse "più debole" del gruppo di trascinamento.

Stato dell'accoppiamento

Nel partprogram dell'NC è possibile informarsi sullo stato dell'accoppiamento mediante le seguenti variabili di sistema:

`$AA_COUP_ACT [asse]`

0: Nessun accoppiamento attivo

8: Trascinamento attivo

9.3 Tabelle di curve (CTAB)

9.3.1 Tabelle delle curve: relazioni generiche

Funzione

Nel capitolo Tabelle di curve si possono trovare i comandi per la programmazione di relazioni tra due assi (asse master e slave).

In un determinato campo di valori, ad ogni valore del master si può abbinare in modo univoco la grandezza slave. Se il valore master si trova al di fuori del campo di definizione, si può programmare il comportamento ai bordi della tabella di curve per tabelle periodiche e aperiodiche.

Descrizione

Le camme meccaniche vengono sostituite da tabelle di curve con le quali si possono definire

- determinati andamenti della curva in un campo di definizione
- singole sezioni, cosiddetti segmenti di curve
- i bordi della curva per tabelle di curve periodiche e aperiodiche
- le posizioni interessate dei segmenti delle curve

.

In un determinato campo dei valori, si possono leggere

- il relativo valore slave di un valore master e
- il valore master di un valore slave

delle posizioni della tabella ed inoltre il valore iniziale e finale di un segmento della tabella.

Vengono rappresentate tutte le ulteriori forme e parametri opzionali per abbinare le relative istruzioni di programmazione. Le possibilità che ne conseguono per influenzare in modo mirato una o più tabelle di curve nel relativo tipo di memoria permettono una programmazione flessibile per ulteriori applicazioni. In questo modo sussistono parecchie possibilità di programmazione anche per la diagnosi di accoppiamenti assi.

Per la definizione di tabelle di curve e dell'accesso alle posizioni delle tabelle di curve, vengono utilizzati ogni volta tipici esempi di programmazione.

9.3.2 Funzioni principali delle tabelle di curve (CTABDEF, CATBEND, CTABDEL)

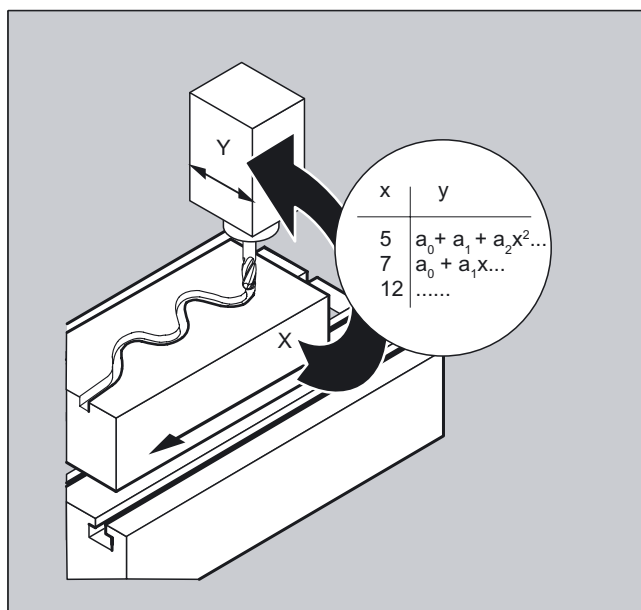
Funzione

Mediante le tabelle è possibile programmare i rapporti di posizione e velocità fra due assi. Le definizioni delle tabelle avvengono in un partprogram.

Esempio di sostituzione di una camma curvilinea meccanica:

la tabella costituisce la base per l'accoppiamento assiale al valore master, in quanto crea il rapporto funzionale fra asse master e asse slave:

Il controllo, in base alla programmazione eseguita, calcola dalle posizioni reciproche dell'asse master e dell'asse slave un polinomio che corrisponde al profilo di camma.



Programmazione

Istruzioni di linguaggio a validità modale con tabelle di curve

CTABDEF(asse S, asse M, n, applim, memType)

oppure

CTABEND ()

oppure

CTABDEL () , CTABDEL (, , memType)

Parametri

Funzioni principali

CTABDEF ()	Definire l'inizio della tabella.
CTABEND ()	Definire la fine della tabella.
CTABDEL ()	Cancellare tutte le tabelle delle curve, indipendentemente dal tipo di memoria
Asse slave	asse slave Asse che viene programmato tramite la tabella.
Asse master	asse master Asse attraverso cui viene programmato il valore master.
n, m	Numero della tabella di curve; $n < m$, es. con CTABDEL(n, m) Il numero assegnato a una tabella delle curve è unico ed è indipendente dal tipo di memoria. Non possono coesistere tabelle con lo stesso numero nelle memorie SRAM e DRAM.
applim	Codice per la periodicità delle tabelle Tabella non periodica Tabella periodica riferita all'asse master Tabella periodica riferita all'asse master e all'asse slave
memType	Indicazione opzionale del tipo di memoria dell'NC: "DRAM" / "SRAM" Se per questo parametro non viene programmato nessun valore, viene utilizzato il tipo di memoria standard definito con il DM 20905: CTAB_DEFAULT_MEMORY_TYPE.

Costruttore della macchina

Per la realizzazione delle tabelle è necessario riservare l'opportuno spazio di memoria, settando i relativi dati macchina.

Esempio di utilizzo di CTABDEF e CTABEND

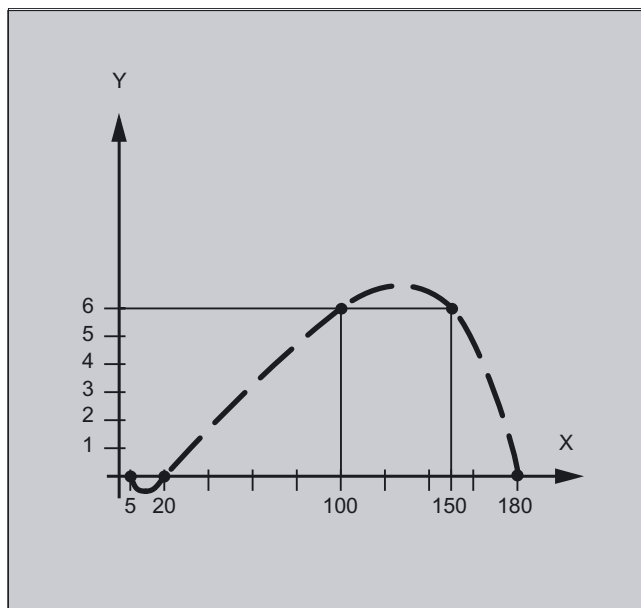
Si intende impiegare una sezione inalterata di partprogram per la definizione di una tabella. Il comando di stop preelaborazione STOPRE che vi compare può restare e ridiventa attivo non appena la sezione di programma non viene più impiegata per la definizione della tabelle e CTABDEF e CTABEND sono state eliminate:

```

...
CTABDEF (Y, X, 1, 1)                ;Definizione di una tabella di curve
...
...
IF NOT ($P_CTABDEF)
STOPRE
ENDIF
...
...
CTABEND

```

Esempio di definizione di una tabella di curve



```
N100 CTABDEF (Y,X,3,0)
```

```
;Inizio della definizione di una
;tabella di curve non periodica con
;numero 3
```

```
N110 X0 Y0
```

```
;1. Istruzione di movimento, definisce
;i valori di partenza e il 1° punto di appoggio:
;valore master: 0; valore slave: 0
```

```
N120 X20 Y0
```

```
;2. Posizioni di appoggio: valore master: 0...20
;Valore slave: valore iniziale...0
```

```
N130 X100 Y6
```

```
;3. Posizioni di appoggio: valore master: 20...100
;Valore slave: 0...6
```

```
N140 X150 Y6
```

```
;4. Posizioni di appoggio: valore master: 100...150
;Valore slave: 6...6
```

```
N150 X180 Y0
```

```
;5. Posizioni di appoggio: valore master: 150...180
;Valore slave: 6...0
```

```
N200 CTABEND
```

```
;Conclusione della definizione
;la tabella di curve viene generata come
;polinomio di max. 5° grado;
;il calcolo del tratto curvilineo con i
;punti di appoggio indicati dipende
;dal tipo di interpolazione modale scelto
;(interpolazione circolare, lineare, spline)
;viene ripristinato lo stato del partprogram
;precedente alla
;definizione.
```

Esempio di definizione di una tabella periodica di curve

Definizione di una tabella periodica con numero 2, campo dei valori master da 0 a 360, movimento dell'asse slave da 0 a 45 e ritorno a 0:

```
N10 DEF REAL DEPPOS
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1) ;Inizio della definizione
N40 G1 X=0 Y=0
N50 POLY
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)
N100 PO[X]=(360.0)
N110 CTABEND ;Fine della definizione

;Test della curva mediante accoppiamento di Y a X
N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)

N170 DEPPOS=CTAB(75.0,2,GRADIENTE) ;Lettura della funzione tabellare con valore master
;75.0

N180 G0 X75 Y=DEPPOS ;Posizionamento di asse master ed asse slave

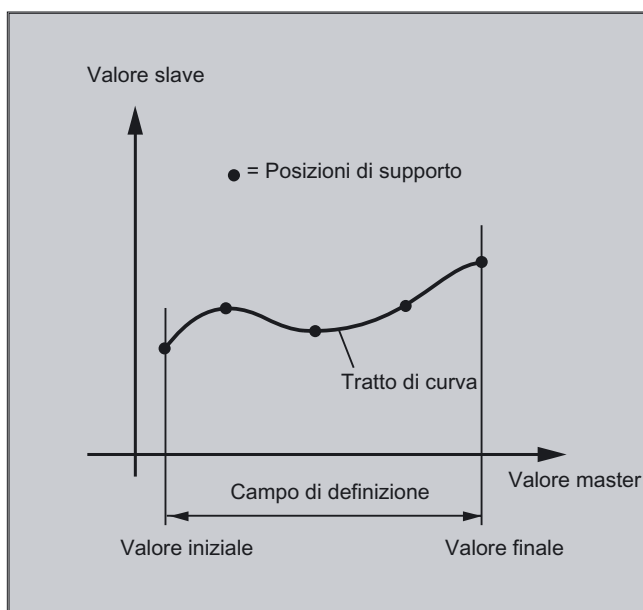
;Dopo l'attivazione dell'accoppiamento non è più necessario sincronizzare l'asse
slave
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30
```

Definizione della tabella di curve

CTABDEF, CTABEND

Una tabella per una curva costituisce un partprogram o una sezione di partprogram, contrassegnato all'inizio con l'istruzione `CTABDEF` e alla fine dalla istruzione `CTABEND`.

All'interno di questa sezione di partprogram alle singole posizioni dell'asse master vengono assegnate, mediante istruzioni di movimento, posizioni ben determinate dell'asse slave, che servono come supporto per il calcolo del tratto curvilineo sotto forma di un polinomio al massimo di 5° grado.



Valore iniziale e finale della tabella di curve

Quale valore iniziale, per il settore di definizione della tabelle, vale la prima impostazione di posizioni di assi raggruppati (la prima istruzione di movimento) all'interno della definizione della tabella di curve. Il valore finale del settore di definizione della tabella viene definito dall'ultima istruzione di movimento.

All'interno della definizione della tabella è disponibile l'intero set di istruzioni NC.

Tutte le istruzioni con effetto modale che vengono generate all'interno della definizione della tabella di curve perdono validità alla conclusione della definizione della tabella. Prima e dopo la definizione di una tabella, il partprogram in cui la definizione viene eseguita si trova nello stesso stato.

Nota

Non sono ammessi:

Stop preelaborazione

Salto nel movimento dell'asse master (per es. al cambio di trasformazioni)

Istruzioni di movimento solo per l'asse slave

L'inversione di movimento dell'asse master, cioè la posizione dell'asse master, deve sempre essere univoca

Istruzione CTABDEF e CTABEND in diversi livelli del programma.

Attivazione di ASPLINE, BSPLINE, CSPLINE

Se nell'ambito di una tabella di curve viene attivata una CTABDEF () ... CTABEND, una ASPLINE, BSPLINE oppure CSPLINE, prima di questa attivazione deve essere programmato almeno un punto iniziale. E' da evitare un'attivazione immediata dopo CTABDEF altrimenti la Spline dipende dalla posizione attuale dell'asse prima della definizione della tabella di curve.

Esempio:

```

CTABDEF(Y, X, 1, 0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
CTABEND

```

In funzione del dato macchina DM 20900: CTAB_ENABLE_NO_LEADMOTION possono essere tollerati salti dell'asse slave in assenza di movimento dell'asse master. Continuano a valere le altre limitazioni citate nelle avvertenze.

Nella creazione e nella cancellazione delle tabelle si può utilizzare l'indicazione del tipo di memoria dell'NC.

Cancellazione di tabelle per curve, CTABDEL

Con CTABDEL si possono cancellare tabelle di curve. Questa funzione non può però cancellare tabelle di un accoppiamento assi attivo. Se in un accoppiamento è attiva almeno una tabella di curve, con un comando di cancellazione multiplo CTABDEL() oppure CTABDEL(n, m) non verrà cancellata **nessuna** delle tabelle indirizzate. Con l'indicazione opzionale del tipo di memoria si possono cancellare tabelle di un determinato tipo di memoria; vedere il capitolo "Forme delle tabelle di curve (CTABDEL, ... CTABUNLOCK)".

9.3.3 Forme delle tabelle di curve (CTABDEL, CTABNOMEM, CTABFNO, CTABID, CTABLOCK, CTABUNLOCK)

Funzione

Ulteriori applicazioni di tabelle di curve:

- Cancellazione in un determinato tipo di memoria SRAM o DRAM.
- Definizione del numero delle tabelle di curve **definite** ed ancora **possibili** nel tipo di memoria.
- **Blocco** della cancellazione o della sovrascrittura delle tabelle di curve oppure **disattivazione** del blocco.
- Impostazioni opzionali per le selezioni: cancellare **una** tabella di curve. Cancellare **un** settore della tabella curve. Cancellare **tutte** le tabelle delle curve che si trovano nella memoria indicata, **e bloccosovrascrittura** nonché **cancellazione** del blocco.
- Indicazioni sulla diagnosi di accoppiamenti assi come fornitura, restituzione e verifica di determinate proprietà delle tabelle di curve
Definizione del numero di tabelle di curve, segmenti di curve e polinomi di curve

Programmazione

Istruzioni di linguaggio a validità modale con tabelle di curve

CTABDEL(n, m, memType)

oppure

CTABNOMEM (memType)

oppure

CTABFNO(memType)

oppure

CTABID(n, memType)

oppure

CTABLOCK(n, m, memType) oppure CTABUNLOCK(n, m, memType)

oppure

CTABDEL(n) oppure CTABDEL(n, m)

oppure

CTABLOCK(n) oppure CTABLOCK(n, m) oppure CTABLOCK() oppure
CTABLOCK(, , memType)

oppure

CTABUNLOCK(n) oppure CTABUNLOCK(n, m) oppure CTABUNLOCK() oppure
CTABUNLOCK(, , memType)

oppure

CTABID(n) oppure CTABID(n, memType) oppure CTABID(p, memType)

oppure

CTABISLOCK(n)

oppure

CTABEXISTS(n)

oppure

CTABMEMTYP(n)

oppure

CTABPERIOD(n)

oppure

CTABSEGID(n, segType)

oppure

CTABSEG(memType, segType) oppure CTABFSEG(memType, segType) oppure
CTABMSEG(memType, segType)

oppure

CTABPOLID(n) oppure CTABMPOL(memType)

Parametri

Forma generale nella memoria NC statica o dinamica:

CTABDEL (n, m, memType)	Cancellare le tabelle appartenenti a un settore delle tabelle delle curve, che sono memorizzate in memType.
CTABNOMEM (memType)	Numero di tabelle delle curve definite .
CTABFNO (memType)	Numero di tabelle delle curve possibili .
CTABID (n, memType)	Restituisce il n. di tabella che nel tipo di memoria è inserito come n.ma tabella delle curve.
CTABLOCK (n, m, memType)	Attivazione del blocco contro cancellazione e sovrascrittura.
CTABUNLOCK (n, m, memType)	Cancellazione del blocco contro cancellazione e sovrascrittura. CTABUNLOCK annulla il blocco attivato precedentemente per le tabelle con CTABLOCK. In presenza di un accoppiamento attivo le tabelle continuano a rimanere bloccate e non possono essere cancellate. Il blocco con CTABLOCK risulta disattivato non appena viene rimosso tramite l'accoppiamento attivo. In questo modo è possibile cancellare questa tabella. Non è necessario un altro richiamo di CTABUNLOCK.

Utilizzi di ulteriori forme Indicazioni opzionali per le selezioni:

CTABDEL (n)	Cancellare una tabella di curve.
CTABDEL (, , memType)	Cancellare un settore della tabella di curve.
CTABLOCK (n)	Cancellare tutte le tabelle di curve che si trovano nella memoria indicata. Blocco cancellazione e sovrascrittura: tabella delle curve con numero n.
CTABLOCK (n, m)	Bloccare la tabella delle curve nel campo numerico da n a m.
CTABLOCK ()	Tutte le tabelle curve già presenti.
CTABLOCK (, , memType)	Tutte le tabelle delle curve nel tipo di memoria.
CTABUNLOCK (n)	Disattivazione cancellazione e sovrascrittura: tabella delle curve con numero n.
CTABUNLOCK (n, m)	Sbloccare la tabella di curve nel campo numerico da n ... m.
CTABUNLOCK ()	Tutte le tabelle curve già presenti.
CTABUNLOCK (, , memType)	Tutte le tabelle delle curve nel tipo di memoria.

Utilizzi di ulteriori forme per la diagnosi di accoppiamenti assi:

CTABID (n, memType)	Restituisce numeri di tabelle delle curve n/p del tipo di memoria memType.
CTABID (p, memType)	
CTABID (n)	Restituisce numeri di tabelle delle curve n con il tipo di memoria definito con il DM 20905: CTAB_DEFAULT_MEMORY_TYPE tipo di memoria definito .
CTABISLOCK (n)	Restituisce lo stato del blocco della tabelle di curve con il numero n .
CTABEXISTS (n)	Verifica la tabella di curve con il numero n.
CTABMEMTYP (n)	Restituisce la memoria nella quale è stata creata la tabella di curve con il numero n.

CTABPERIOD (n)	Restituisce la periodicità delle tabelle.
CTABSEG (memType)	Numero dei segmenti di curve già utilizzati nel tipo di memoria interessato.
CTABSEGID (n)	Numero dei segmenti di curve utilizzati dalla tabella di curve con il numero n
CTABFSEG (memType)	Numero dei segmenti di curve possibili .
CTABMSEG (memType)	Numero massimo di segmenti possibili.
CTABPOLID (n)	Numero dei polinomi di curve con numero n utilizzati dalla tabella di curve .
CTABSEG (memTyp, segType)	Numero dei segmenti di curve già utilizzati del tipo "L" o "P" nel tipo di memoria.
CTABFSEGID (n, segType)	Numero dei segmenti di curve utilizzati dalla tabella di curve del tipo "L" o "P".
CTABFSEG (memTyp, segType)	Numero dei segmenti di curve ancora possibili del tipo "L" o "P" nel tipo di memoria.
CTABMSEG (memTyp, segType)	Numero dei segmenti di curve massimi possibili del tipo "L" o "P" nel tipo di memoria.
CTABFPOL (memType)	Numero dei polinomi di curve ancora possibili nel tipo di memoria interessato.
CTABMPOL (memType)	Numero massimo dei polinomi di curve possibili nel tipo di memoria interessato.
n, m	Numero della tabella di curve; $n < m$, es. con CTABDEL(n, m) Il numero assegnato a una tabella delle curve è unico ed è indipendente dal tipo di memoria. Non possono trovarsi tabelle con lo stesso numero nella memoria NC statica e dinamica.
p	Spazio nella memoria (nell'area di memoria memType)
memType	Indicazione opzionale del tipo di memoria dell'NC: È possibile sia la "memoria dinamica" che la "memoria statica" Se per questo parametro non viene programmato nessun valore, viene utilizzato il tipo di memoria standard definito con il DM 20905: CTAB_DEFAULT_MEMORY_TYPE.
segType	Indicazione opzionale del tipo di segmento. Valori possibili sono: segType "L" segmenti lineari segType "P" segmenti polinomi

Descrizione

Caricamento di tabelle di curve tramite "Elaborazione dall'esterno"

Per l'elaborazione dall'esterno delle tabelle di curve è necessario definire la dimensione del buffer di caricamento (DRAM) con il DM D18360 \$MN_MM_EXT_PROG_BUFFER_SIZE in modo che tutte le tabelle di curve definite possano essere create contemporaneamente nel buffer di caricamento. In caso contrario l'elaborazione del partprogram viene interrotta con l'allarme 15150.

Impiego ripetuto di tabelle di curve

Il rapporto funzionale di asser master ed asse slave calcolato tramite la tabella per una curva resta abbinato al numero di tabella scelto anche dopo la fine del partprogram e dopo Power Off, solo se la tabella stessa è stata creata nella memoria NC statica (SRAM).

Una tabella creata nella memoria dinamica (DRAM) viene cancellata con Power On e deve essere eventualmente ricreata.

Una tabella di curve creata una volta può essere utilizzata a piacere per qualsiasi combinazione di asse master ed asse slave ed è indipendente dagli assi impiegati per la generazione della tabella stessa.

Sovrascrittura di tabelle per curve

Una tabella viene sovrascritta quando, definendo una nuova tabella, per la definizione ne viene impiegato il numero.

Eccezione: Una tabella di curve è attiva in un'accoppiamento assi oppure bloccata con CTABLOCK() .

Nota

Al momento della sovrascrittura di tabelle non viene emessa alcuna segnalazione di avvertimento!

Mediante la variabile di sistema `$P_CTABDEF` è possibile richiedere in qualsiasi momento a un partprogram se è attiva la definizione di tabella di curve.

Dopo aver estratto le istruzioni per la definizione della tabella di curve è possibile utilizzare nuovamente la sezione di partprogram come un partprogram reale.

9.3.4 Comportamento sui bordi delle tabelle di curve (CTABTSV, CATBTSP, CTABMIN, CTABMAX)

Funzione

Se il valore master si trova al di fuori del campo di definizione, il valore di un asse slave può essere letto all'inizio ed alla fine della tabella di curve.

Con CTABTSV può essere letto il valore di un **asse slave** all'inizio della tabella di curve. Con CTABTSP può essere letto il valore di un **asse slave** alla fine della tabella di curve.

I valori iniziali e finali di una tabella di curve dipendono dal fatto che la tabella sia stata definita con valori master crescenti o decrescenti. Il valore iniziale viene sempre definito dal limite inferiore dell'intervallo ed il valore finale dal limite superiore dell'intervallo.

Con CTABTMIN e CTABTMAX si possono stabilire i **valori minimi** e **massimi** del campo globale oppure di un determinato intervallo di una tabella di curve. Per il relativo intervallo del valore master vengono indicati due limiti.

Programmazione

Valore slave, valore iniziale e finale dell'asse slave:

CTABTSV(n, gradi, asse S), CTABTEV(n, gradi, asse S)

Valore master, valore iniziale e finale dell'asse master:

CTABTSP(n, gradi, asse S), CTABTEP(n, gradi, asse S)

Campi di valori min. e max.:

CTABTMIN(n, asse S)

oppure

CTABTMAX(n, asse S)

Parametri

CTABTSV	Lettura del valore iniziale della tabella di curve da un asse slave.
CTABTEV ()	Lettura del valore finale della tabella di curve da un asse slave.
CTABTSP ()	Lettura del valore iniziale della tabella di curve da un asse master.
CTABTEP ()	Lettura del valore finale della tabella di curve da un asse master.
CTABMIN ()	Definizione del valore minimo di una tabella di curve nell'intero campo oppure in un determinato intervallo.
CTABMAX ()	Definizione del valore massimo di una tabella di curve nell'intero campo oppure in un determinato intervallo.
Asse slave	asse slave
	Asse che viene programmato tramite la tabella.
Asse master	asse master
	Asse attraverso cui viene programmato il valore master.

n, m	Numero delle tabelle di curve I numeri di tabelle di curve possono essere assegnati liberamente. Esso serve esclusivamente per una identificazione univoca.
grado	Gradiente per l'incremento all'inizio / alla fine della tabella curve

Valori e campo dei valori

Valori degli assi slave e master che si trovano all'inizio ed alla fine di una tabella di curve CTABTSV, CTABTEV, CTABTSP, CTABTEP

R10=CTABTSV(n, gradi, asse S).	Valore slave all'inizio della tabella di curve
R10=CTABTEV(n, gradi, asse S).	Valore slave all'inizio della tabella di curve
R10=CTABTSP(n, gradi, asse M).	Valore master all'inizio della tabella di curve
R10=CTABTEP(n, gradi, asse M).	Valore master alla fine della tabella di curve

Campo dei valori della tabella di curve del valore slave CTABTMIN, CTABTMAX

R10=CTABTMIN(n, asse S)	Valore slave minimo della tabella di curve nell'intervallo globale
R10=CTABTMAX(n, asse S)	Valore slave massimo della tabella di curve nell'intervallo globale
R10=CTABTMIN(n, a, b, asse S, asse M)	Valore slave minimo della tabella di curve nell'intervallo a...b del valore master
R10=CTABTMAX(n, a, b, asse S, asse M)	Valore slave massimo della tabella di curve nell'intervallo a...b del valore master

Nota

Le assegnazioni a parametri R nell'ambito della definizione delle tabelle vengono annullate.

Esempio di assegnazioni a parametri R

```

...
R10=5 R11=20
...
CTABDEF
G1 X=10 Y=20 F1000
R10=R11+5           ;R10=25
X=R10
CTABEND
...                 ;R10=5

```

Esempio di utilizzo di CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABTMIN, CTABTMAX

Definizione dei valori minimi e massimi delle tabelle di curve.

N10 DEF REAL STARTVAL	
N20 DEF REAL ENDVAL	
N30 DEF REAL STARTPARA	
N40 DEF REAL ENDPARA	
N50 DEF REAL MINVAL	
N60 DEF REAL MAXVAL	
N70 DEF REAL GRADIENTE	
...	
N100 CTABDEF(Y,X,1,0)	;Inizio della definizione della tabella
N110 X0 Y10	;Valore iniziale del 1° segmento della tabella
N120 X30 Y40	;Valore finale del 1° segmento della tabella =
N130 X60 Y5	;Valore iniziale del 2° segmento della tabella ...
N140 X70 Y30	
N150 X80 Y20	
N160 CTABEND	;Fine della definizione della tabella
...	
N200 STARTPOS = CTABTSV(1, GRADIENTE)	;Posizione iniziale STARTPOS = 10,
N210 ENDPOS = CTABTEV(1, GRADIENTE)	;Posizione finale ENDPOS = 20 della tabella, e
N220 SRARTPARA = CTABTSP(1, GRADIENTE)	;STARTPARA = 10,
N230 ENDPARA = CTABTEP(1, GRADIENTE)	;ENDPARA = 80 lettura del campo di valori
...	;dell'asse slave.
N240 MINVAL = CTABTMIN(1)	;Valore minimo con Y = 5 e
N250 MAXVAL = CTABTMAX(1)	;valore massimo con Y = 40

Tabella di curve non periodica

Se il valore master si trova al di fuori del campo di definizione come valore slave viene emesso il limite superiore o inferiore.

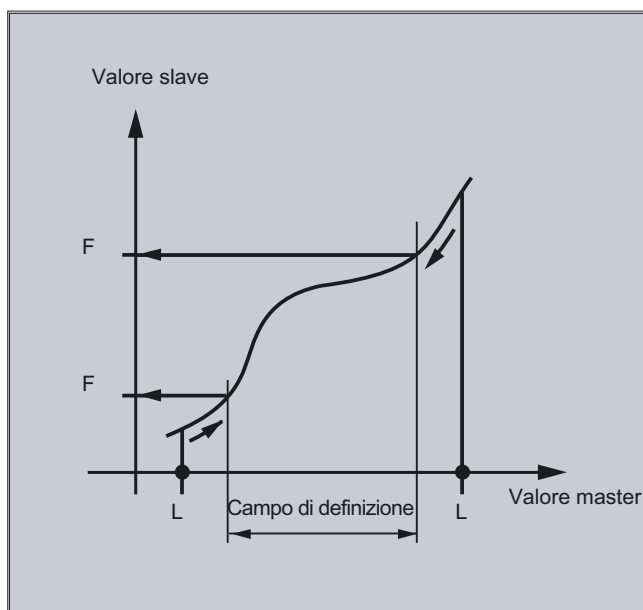
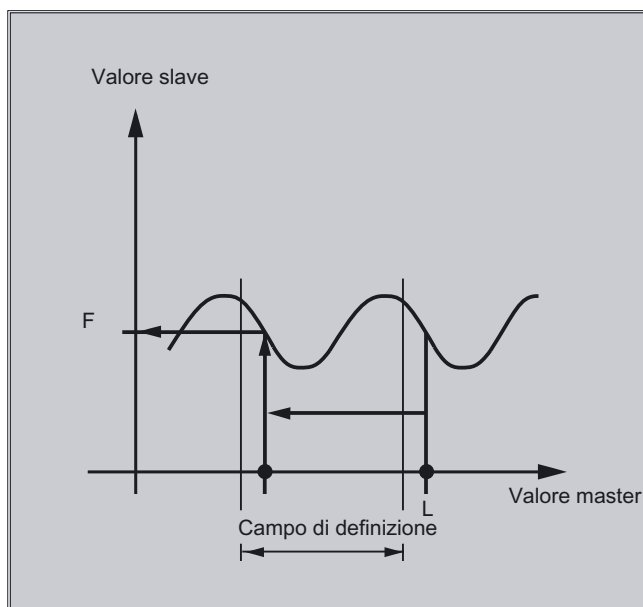


Tabella di curve periodica

Se il valore master si trova al di fuori del campo di definizione viene valutato il valore Modulo del campo di definizione e viene emesso il corrispondente valore slave.



Nota**CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABTMIN, CTABTMAX**

Queste istruzioni possono essere utilizzate dal partprogram oppure direttamente da azioni sincrone.

Il tempo di esecuzione interno della funzione dipende dal numero dei segmenti nella tabella:	
CTABINV()	dipendente
CTABTSV, CTABTEV, CTABTSP, CTABTEP (CTABTMIN, CTABTMAX solo se non viene indicato alcun intervallo del valore master)	indipendente

Lettura in azioni sincrone

L'utilizzatore deve prestare attenzione che utilizzando le istruzioni CTABINV() oppure CTABTMIN() e CTABTMAX() in azioni sincrone, nel punto esatto dell'esecuzione

- siano disponibili sufficienti risorse NC oppure
- il numero dei segmenti della tabella di curve deve essere interrogato prima del richiamo per poter eventualmente suddividere la relativa tabella.

Nel capitolo "Azioni sincrone al movimento" vengono descritti le ulteriori relazioni con la programmazione di azioni sincrone.

9.3.5 Accesso alle posizioni ed ai segmenti delle tabelle di curve (CTAB, CTABINV, CTABSSV, CATBSEV)

Funzione**Lettura di posizioni delle tabelle, CTAB, CTABINV**

CTAB consente di leggere direttamente dal partprogram o da azioni sincrone il valore slave relativo ad un valore master.

CTABINV consente di trasferire un valore master al valore slave. Questo abbinamento non deve essere necessariamente univoco. CTABINV necessita quindi di un valore approssimativo per il valore master atteso.

Programmazione**Valore slave ad un valore master**

```
CTAB(valore master, n, gradi, [asse slave, asse master])
```

Lettura di un valore slave ad un valore master

```
CTAB(valore slave, valore master appross., n, gradi, [asse slave, asse master])
```

Lettura del valore master iniziale e finale di un segmento della tabella

```
CTABSSV(valore master, n, gradi, [asse S]),  
CTABSEV(valore master, n, gradi, [asse S])
```

Parametri

CTAB	Lettura del valore slave direttamente da un valore master.
CTABINV	Trasferire il valore master ad un valore slave.
CTABSSV	Di un asse slave viene letto il valore iniziale del segmento della tabella.
CTABSEV	Di un asse slave viene letto il valore finale del segmento della tabella.
Asse slave	asse slave
	Asse che viene programmato tramite la tabella.
Asse master	asse master
	Asse attraverso cui viene programmato il valore master.
n, m	Numero delle tabelle di curve
	Il numero di tabelle di curve può essere assegnato liberamente. Esso serve esclusivamente per una identificazione univoca.
Gradi	Gradiente per l'incremento all'inizio / alla fine della tabella curve
Valore master appross.	Valore di posizione del valore approssimativo previsto con il quale può essere definito un valore master univoco.

- **CTABSSV, CTABSEV**

Con CTABSSV si può leggere il **valore iniziale** del segmento di curve che appartiene al valore master indicato. Con CTABSEV si può leggere il **valore finale** del segmento di curve che appartiene al valore master indicato.

- **Posizione master o slave rilevata dalla tabella di curve con CTAB, CTABINV**

R10=CTAB(LW, n, gradi, Asse S, Asse M)	Valore slave ad un valore master
R10=CTABINV(FW, valore master appross., n, gradi, asse S, asse M)	Valore master ad un valore slave

- **Definizione di segmenti delle tabelle di curve tramite l'impostazione di un valore master con CTABSSV, CTABSEV**

R10=CTABSSV(LW, n, gradi, Asse S, Asse M)	Valore iniziale dell'asse slave nel segmento che appartiene al valore master
R10=CTABSEV(LW, n, gradi, Asse S, Asse M)	Valore finale dell'asse slave nel segmento che appartiene al valore master

Esempio di utilizzo di CTABSSV e CTABSEV

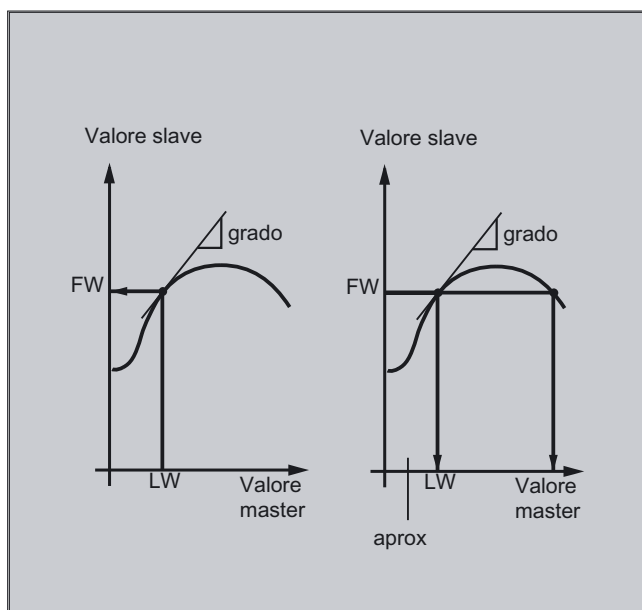
Definizione del segmento di curve appartenente al valore master X = 30.

N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENTE	
...	
N100 CTABDEF(Y,X,1,0)	; Inizio della definizione della tabella
N110 X0 Y0	; Posizione iniziale del 1° segmento di tabella

N120 X20 Y10	; Posizione finale del 1° segmento di tabella = Posizione iniziale del 2° segmento della tabella...
N130 X40 Y40	
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; Fine della definizione della tabella
...	
N200 STARTPOS = CTABSSV(30.0,1,GRADIENT)	; Posizione iniziale Y nel segmento 2 = 10
...	
N210 ENDPOS = CTABSEV(30.0,1,GRADIENT)	; Posizione finale Y nel segmento 2 = 40
	; A LW X = 30.0 appartiene il segmento 2.

Lettura di posizioni delle tabelle, CTAB, CTABINV

CTABINV necessita quindi di un valore approssimativo (*aproxLW*) per il valore master atteso. CTABINV restituisce il valore master più vicino a quello approssimativo. Tale valore approssimativo può essere per esempio il valore master del clock di interpolazione precedente.



Entrambe le funzioni emettono inoltre anche l'incremento della funzione tabellare sulla corrispondente posizione del parametro di incremento (*grad*). In tal modo è possibile calcolare la velocità dell'asse master o dell'asse slave nella relativa posizione.

Nota**CTAB, CTABINV, CTABSSV e CTABSEV**

Le istruzioni CTAB, CTABINV e CTABSSV, CTABSEV possono essere utilizzate dal partprogram oppure direttamente da azioni sincrone. Nel capitolo "Azioni sincrone al movimento" vengono descritte tutte le ulteriori relazioni con la programmazione di azioni sincrone.

L'indicazione opzionale di asse master o asse slave in CTAB/CTABINV/CTABSSV/CTABSEV è importante nel caso in cui questi due assi siano stati programmati in unità di lunghezza diverse.

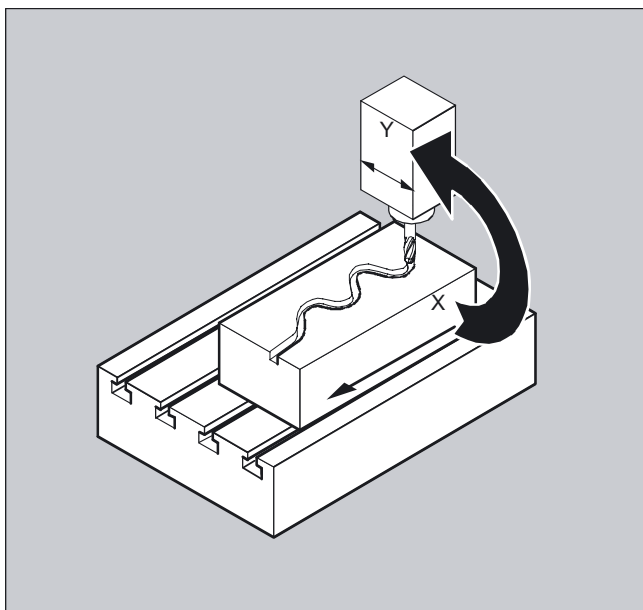
Le istruzioni CTABSSV e CTABSEV nei seguenti casi **non sono adatte** per interrogare i segmenti programmati:

- sono stati programmati cerchi o evolventi.
- sono attivi smussi o raccordi con CHF, RND .
- è attivo un raccordo con G643 .
- è attivo il compressore con COMPON, COMPCURV, COMPCAD .

9.4 Accoppiamento assiale tramite valore master (LEADON, LEADOF)

Funzione

Con l'accoppiamento assiale al valore master un asse master ed un asse slave vengono mossi in maniera sincronizzata. La posizione dell'asse slave è sempre assegnata in modo univoco alla posizione dell'asse master, mediante una tabella o un polinomio da essa calcolato.



L'**asse master** è quello che fornisce i valori di ingresso per la tabella delle curve. Si definisce **asse slave** l'asse che acquisisce le posizioni calcolate tramite la tabella per curve.

Accoppiamento al valore di riferimento e al valore reale

Quali valori master dell'asse slave, cioè come valori di partenza per il rilevamento della posizione, si possono utilizzare:

- i valori reali di posizione dell'asse master: accoppiamento al valore reale
- i valori di riferimento della posizione dell'asse master: Accoppiamento tramite riferimento

L'accoppiamento al valore master ha validità sempre nel sistema di coordinate di base.

Per la realizzazione delle tabelle di curve vedere la sezione "Tabelle di curve" in questo capitolo.

Per l'accoppiamento al valore master, vedere /FB/, M3, Trascinamento e accoppiamento al valore master.

Programmazione

LEADON (asse S, asse M, n)

oppure

LEADOF (asse S, asse M)

o disattivazione senza indicazione dell'asse master

LEADOF (asse S)

L'accoppiamento al valore master può essere attivato e disattivato sia dal partprogram che da azioni sincrone durante il movimento, vedere capitolo "Azioni sincrone al movimento").

Parametri

LEADON	Attivare l'accoppiamento al valore master (di riferimento)
LEADOF	Disattivare l'accoppiamento al valore master (di riferimento)
Asse slave	asse slave
Asse master	asse master
n	Numero della tabella
\$SA_LEAD_TYPE	Commutazione tra accoppiamento al valore di riferimento e accoppiamento al valore reale

Disattivazione dell'accoppiamento, LEADOF

Disattivando l'accoppiamento al valore master l'asse slave torna ad essere un normale asse di comando.

Accoppiamento assiale al valore master e diversi stati operativi, RESET

In funzione dell'impostazione eseguita nel dato macchina gli accoppiamenti al valore master vengono disattivati con RESET.

Esempio di accoppiamento al valore master da azioni sincrone

In un impianto di presse si intende sostituire un accoppiamento meccanico tradizionale fra un asse master (mazza) e gli assi di un sistema transfer costituito da assi di trasferimento e assi ausiliari con un sistema di accoppiamento elettronico.

L'esempio che segue è tratto da un impiego reale e dimostra come un sistema di accoppiamento meccanico viene sostituito con un sistema di accoppiamento elettronico. Le procedure di accoppiamento e di disaccoppiamento sono realizzate come **azioni sincrone statiche**.

L'asse master LW (mazza) comanda assi di trasferimento e assi ausiliari come assi slave definiti tramite tabella per curve.

Assi slave

X Asse di avanzamento o longitudinale
 YL Asse di chiusura o asse trasversale
 ZL Asse di sollevamento
 U Avanzamento lamiera, asse ausiliario
 V Testa orientabile, asse ausiliario
 W Ingrassaggio, asse ausiliario

Azioni

Le azioni presenti nelle azioni sincrone sono per es.:

- Accoppiamento, LEADON (asse slave, asse master, numero delle tabelle)
- Disaccoppiamento, LEADOF (asse slave, asse master)
- Settare il valore reale, PRESETON (asse, valore)
- Settare merker, \$AC_MARKER[i] = valore
- Tipo di accoppiamento: valore master reale/virtuale
- Raggiungimento delle posizioni asse, POS[Asse]=valore

Condizioni

Come condizioni vengono valutati ingressi digitali veloci, variabili in tempo reale \$AC_MARKER e confronti di posizione, combinati con l'operatore logico AND.

Nota

Nell'esempio seguente il cambio riga, i rientri e la scrittura in **grassetto** sono stati impiegati esclusivamente allo scopo di migliorare la lettura della programmazione. Per il controllo numerico tutto ciò che si trova sotto un numero di riga fa parte di una riga unica.

Commento

```
; definisce tutte le azioni sincrone statiche.
; **** reset marker
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0
$AC_MARKER[2]=0 $AC_MARKER[3]=0
$AC_MARKER[4]=0 $AC_MARKER[5]=0
$AC_MARKER[6]=0 $AC_MARKER[7]=0
; **** E1 0=>1 accoppiamento transfer ON
```

```

N10 IDS=1 EVERY ($A_IN[1]==1) AND
($A_IN[16]==1) AND ($AC_MARKER[0]==0)
DO LEADON(X,LW,1) LEADON(YL,LW,2)
LEADON(ZL,LW,3) $AC_MARKER[0]=1
; **** E1 0=>1 accoppiamento avanzamento
lamiera ON

N20 IDS=11 EVERY ($A_IN[1]==1) AND
($A_IN[5]==0) AND ($AC_MARKER[5]==0)
DO LEADON(U,LW,4) PRESETON(U,0)
$AC_MARKER[5]=1
; **** E1 0=>1 accoppiamento testa orientabile ON

N21 IDS=12 EVERY ($A_IN[1]==1) AND
($A_IN[5]==0) AND ($AC_MARKER[6]==0)
DO LEADON(V,LW,4) PRESETON(V,0)
$AC_MARKER[6]=1
; **** E1 0=>1 accoppiamento ingrassaggio ON

N22 IDS=13 EVERY ($A_IN[1]==1) AND
($A_IN[5]==0) AND ($AC_MARKER[7]==0)
DO LEADON(W,LW,4) PRESETON(W,0)
$AC_MARKER[7]=1
; **** E2 0=>1 accoppiamento OFF

N30 IDS=3 EVERY ($A_IN[2]==1)
DO LEADOF(X,LW) LEADOF(YL,LW)
LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW)
LEADOF(W,LW) $AC_MARKER[0]=0
$AC_MARKER[1]=0 $AC_MARKER[3]=0
$AC_MARKER[4]=0 $AC_MARKER[5]=0
$AC_MARKER[6]=0 $AC_MARKER[7]=0
. . . .
N110 G04 F01
N120 M30

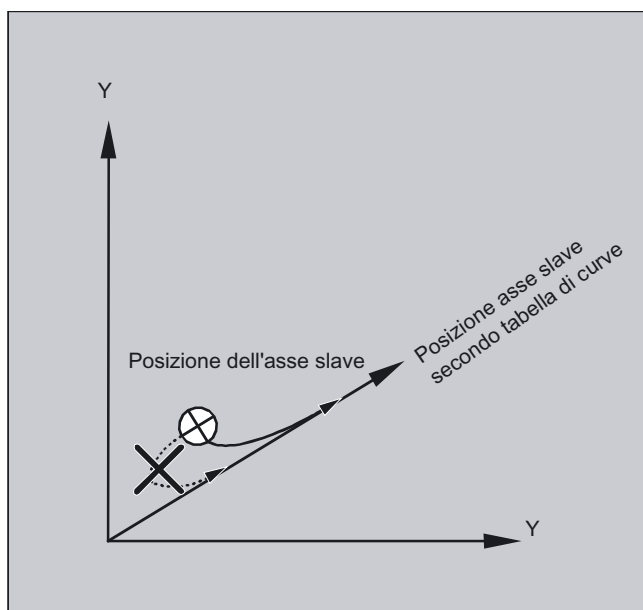
```

Descrizione

L'accoppiamento al valore master richiede la sincronizzazione di asse master ed asse slave. Tale sincronizzazione si può ottenere soltanto se l'asse slave al momento dell'attivazione dell'accoppiamento si trova all'interno del campo di tolleranza del tratto curvilineo che è stato calcolato mediante la tabella di curve.

Il campo di tolleranza per la posizione dell'asse slave viene definito con il dato macchina DM 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE .

Se l'asse slave al momento dell'attivazione dell'accoppiamento al valore master non si trova ancora nella posizione opportuna, la sincronizzazione viene eseguita automaticamente non appena il valore di posizione calcolato dell'asse slave si avvicina alla posizione reale di detto asse. L'asse slave viene mosso solo in direzione delle posizioni per esso calcolate. Nel corso della sincronizzazione l'asse slave viene mosso nella direzione definita dalla velocità di riferimento dell'asse slave (risultante dalla velocità dell'asse master e da CTAB).

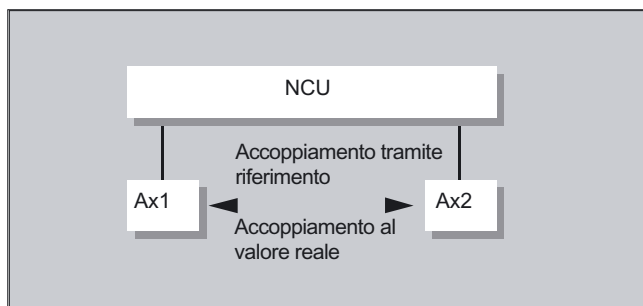


Nessun sincronismo

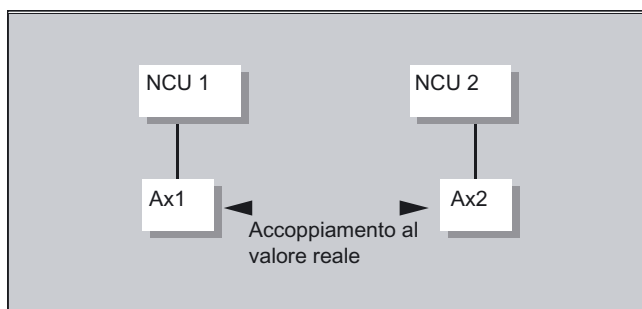
Se al momento dell'attivazione dell'accoppiamento la posizione calcolata dell'asse slave non corrisponde alla posizione attuale dell'asse slave, il sincronismo non viene attivato

Accoppiamento al valore di riferimento e al valore reale

Rispetto all'accoppiamento al valore reale, l'accoppiamento al valore di riferimento offre una migliore marcia sincrona di asse master ed asse slave ed è quindi preimpostato come standard.



L'accoppiamento al valore di riferimento è possibile solo se asse slave ed asse master vengono interpolati dalla stessa NCU. In caso di asse master esterno l'asse slave può essere accoppiato all'asse master solo tramite valori reali.



Una **commutazione** è possibile con il dato di setting `$SA_LEAD_TYPE` .

La commutazione tra accoppiamento al valore di riferimento e accoppiamento al valore reale dovrebbe avvenire sempre ad asse slave fermo. Una nuova sincronizzazione dopo la commutazione viene eseguita solo in stato di fermo.

Esempio d'utilizzo

Nel caso la macchina sia sottoposta a forti vibrazioni non è possibile leggere senza errori i valori reali. Se per es. nel caso di presse è in funzione l'accoppiamento al valore reale, nei passi di lavoro con le più forti vibrazioni può rendersi quindi necessario commutare all'accoppiamento al valore di riferimento.

Simulazione del valore master in caso di accoppiamento al valore di riferimento

Mediante dato macchina è possibile separare l'interpolatore dell'asse master dal servo. In tal modo nel caso di accoppiamento al valore di riferimento è possibile generare valori di riferimento anche senza movimenti reali dell'asse master.

Per il loro impiego, per es. in azioni sincrone, i valori master generati mediante accoppiamento al valore di riferimento possono essere letti nelle seguenti variabili:

- <code>\$AA_LEAD_P</code>	posizione master
- <code>\$AA_LEAD_V</code>	velocità master

Generazione dei valori master

I valori master possono essere generati a scelta in altri modi programmati dall'utente. I valori master generati in tal modo vengono scritti e letti nella variabile

- <code>\$AA_LEAD_SP</code>	posizione master
- <code>\$AA_LEAD_SV</code>	velocità master

. Per l'utilizzo di queste variabili il dato di setting `$SA_LEAD_TYPE = 2` .

Stato dell'accoppiamento

Nel partprogram dell'NC è possibile informarsi sullo stato dell'accoppiamento mediante le seguenti variabili di sistema:

\$AA_COUP_ACT[asse]

0: Nessun accoppiamento attivo

16: Accoppiamento al valore master attivo

Stato della gestione con azioni sincrone

Le procedure di inserzione e di accoppiamento vengono gestite con variabili in tempo reale:

\$AC_MARKER[i] = n

gestito con:

i Merker nr.

n Valore di stato

9.5 Andamento dell'avanzamento (FNORM, FLIN, FCUB, FPO)

Funzione

Per impieghi sempre più flessibili dell'avanzamento, la programmazione viene ampliata con la caratteristica lineare e cubica secondo DIN 66025.

Le caratteristiche cubiche possono essere programmate o direttamente, o come spline interpolate. In questo modo è possibile programmare, a seconda della curvatura del pezzo in lavorazione, caratteristiche di velocità con variazione graduale continua.

Queste caratteristiche di velocità consentono variazioni di accelerazioni senza overshoot e quindi una finitura uniforme della superficie del pezzo.

Programmazione

F... FNORM

oppure

F... FLIN

oppure

F... FCUB

oppure

F=FPO (... , ... , ...)

Parametri

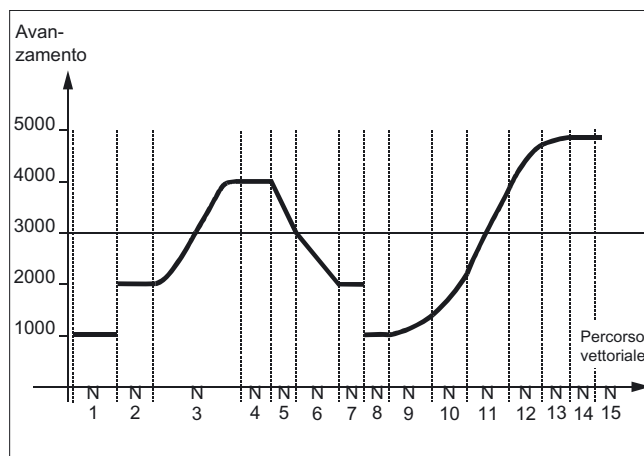
FNORM	Posizione base. Il valore di avanzamento viene specificato nei blocchi di percorso ed ha validità modale.
FLIN	Profilo velocità vettoriale lineare: Il valore dell'avanzamento viene impostato linearmente dal valore attuale all'inizio del blocco fino alla fine dello stesso ed è valido poi come valore modale. Questo comportamento può essere combinato con G93 e G94.
FCUB	Profilo velocità vettoriale cubico: I valori di F programmati con validità blocco-blocco vengono collegati tramite una spline, prendendo come riferimento il punto finale del blocco. La Spline inizia e finisce tangenzialmente all'impostazione dell'avanzamento precedente o successivo ed agisce con G93 e G94. Se in un blocco manca l'indirizzo F, viene utilizzato l'ultimo valore di F programmato.
F=FPO...	Profilo velocità vettoriale tramite polinomi: L'indirizzo F identifica la caratteristica dell'avanzamento con polinomio dal valore attuale fino a fine blocco. Il valore finale vale pertanto come valore modale.

Ottimizzazione dell'avanzamento con tratti di percorso curvilinei

Il polinomio di avanzamento $F=FPO$ e la spline di avanzamento $FCUB$ devono essere sempre utilizzati con la velocità di taglio costante CFC . In questo modo è possibile generare un profilo di avanzamento ed accelerazione continua.

Esempio di diversi profili di avanzamento

In questo esempio vengono programmati e rappresentati graficamente diverse tipologie di profili di avanzamento.

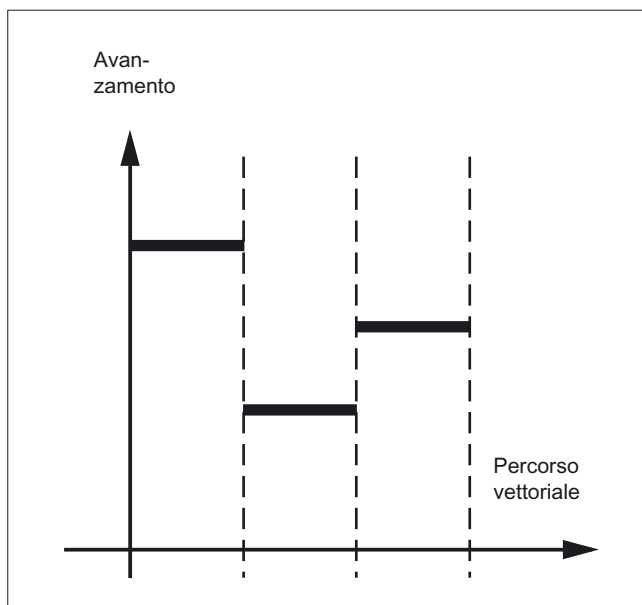


N1 F1000 FNORM G1 X8 G91 G64	;Profilo di avanzamento costante, Impostazione incrementale.
N2 F2000 X7	;Modifica discontinua della velocità di riferimento
N3 F=FPO(4000, 6000, -4000)	;Profilo di avanzamento tramite polinomio con avanzamento ;4000 a fine blocco
N4 X6	;L'avanzamento polinomiale 4000 vale come valore modale
N5 F3000 FLIN X5	;Profilo di avanzamento lineare
N6 F2000 X8	;Profilo di avanzamento lineare
N7 X5	;Il profilo di avanzamento lineare vale come valore modale
N8 F1000 FNORM X5	;Profilo di avanzamento costante con variazione ;a gradino dell'accelerazione
N9 F1400 FCUB X8	;Tutti i successivi valori di F programmati con validità ;blocco-blocco vengono collegati con la spline
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	;Disattivazione del profilo spline
N14 FNORM X5	
N15 X20	

FNORM

L'indirizzo F indica l'avanzamento lineare come valore costante secondo DIN 66025.

Ulteriori indicazioni si possono trovare nel manuale di programmazione "Concetti fondamentali".

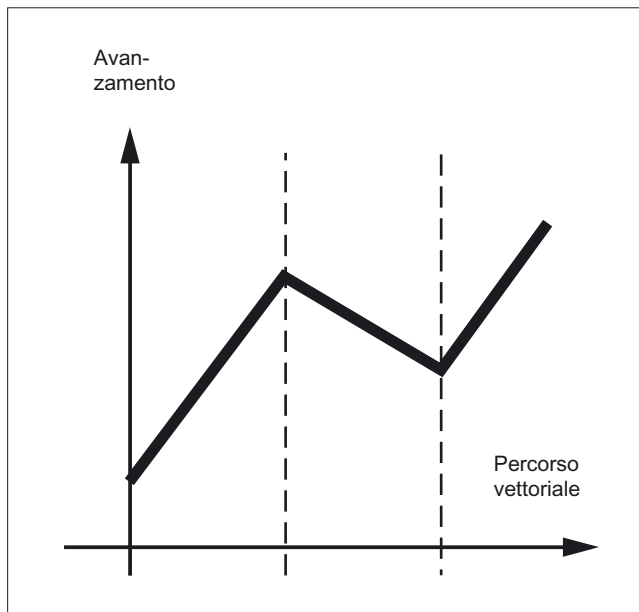


FLIN

La caratteristica dell'avanzamento viene impostata dal valore di avanzamento attuale al valore di F programmata, linearmente fino alla fine del blocco.

Esempio:

N30 F1400 FLIN X50



FCUB

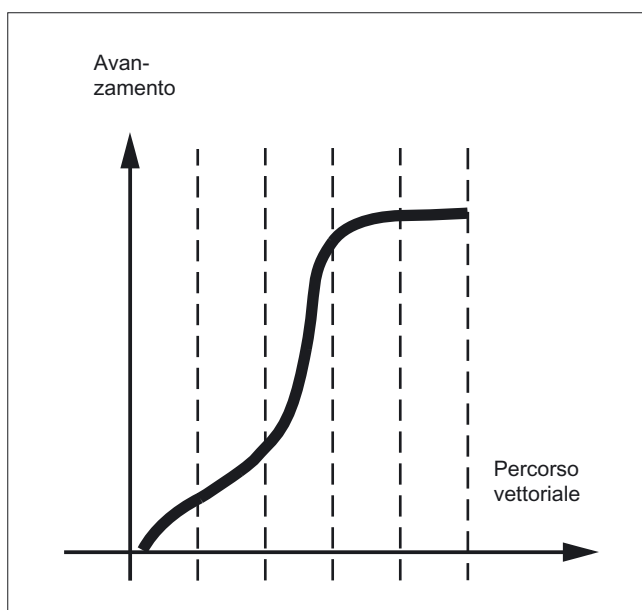
L'avanzamento viene impostato con caratteristica cubica dal valore di avanzamento attuale al valore di F programmato fino a fine blocco. Il controllo numerico collega tramite spline tutti i valori di avanzamento programmati con validità blocco-blocco ed FCUB attiva. I valori di avanzamento, in questo caso, servono come punti di supporto per il calcolo dell'interpolazione spline.

Esempio:

N50 F1400 FCUB X50

N60 F2000 X47

N70 F3800 X52



$F=FPO(...,...,...)$

La caratteristica dell'avanzamento viene programmata direttamente con un polinomio. L'impostazione dei coefficienti polinomici avviene analogamente all'interpolazione polinomica.

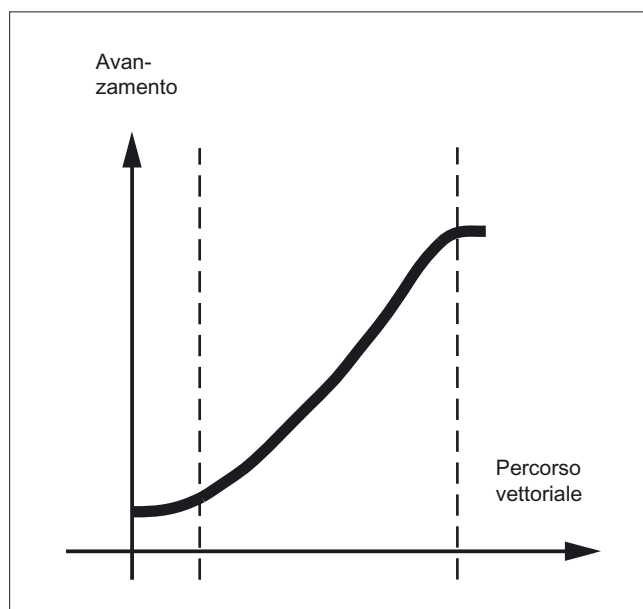
Esempio:

$F=FPO(endfeed, quadf, cubf)$

endfeed, quadf e cubf sono variabili definite in precedenza.

endfeed:	avanzamento a fine blocco
quadf:	coefficiente polinomico quadratico
cubf:	coefficiente polinomico cubico

Con **FCUB** attiva, la spline collega tangenzialmente ad inizio e a fine blocco la caratteristica definita tramite **FPO**.



Condizioni marginali

Le funzioni per la programmazione del movimento sul profilo sono valide indipendentemente dalla caratteristica dell'avanzamento programmata.

La caratteristica dell'avanzamento programmabile è sempre assoluta, indipendentemente da G90 oppure G91.

L'andamento dell'avanzamento FLIN e FCUB agisce con

G93 e G94.

FLIN e FCUB non agiscono con

G95, G96/G961 e G97/G971.

Compressore attivo COMPON

Con compressore attivo COMPON , per la raccolta di più blocchi in un segmento spline vale quanto segue:

FNORM:

Per il segmento spline vale la word F dell'ultimo blocco di appartenenza.

FLIN:

Per il segmento spline vale la parola F dell'ultimo blocco di appartenenza.

Il valore di F programmato vale fino al termine del segmento; l'accostamento viene poi effettuato linearmente.

FCUB:

La spline di avanzamento generata, rispetto ai punti finali programmati, ha uno scostamento max. pari al valore definito nel dato macchina C \$MC_COMPRESS_VELO_TOL.

F=FPO(...,...,...)

Questi blocchi non vengono compressi.

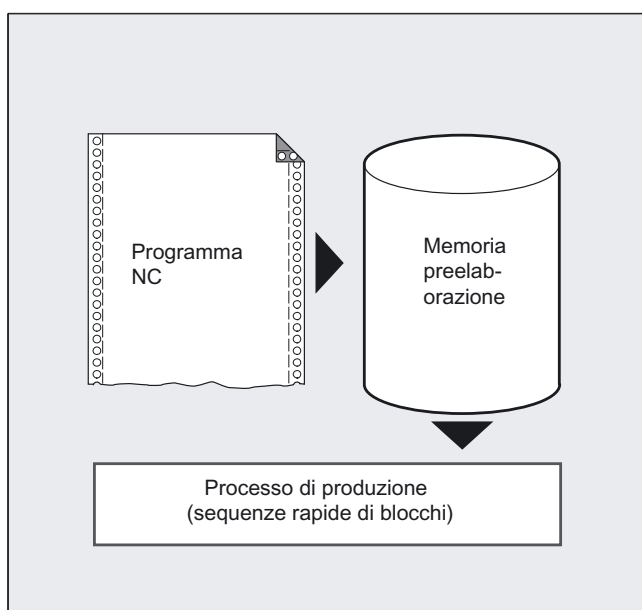
9.6 Esecuzione del programma con buffer di preelaborazione (STARTFIFO, STOPFIFO, STOPRE)

Funzione

A seconda del tipo, il controllo numerico dispone di una determinata quantità di memoria di preelaborazione, che memorizza i blocchi preparati prima della loro esecuzione per poi emetterli come sequenze rapide di blocchi nel corso della produzione.

In tal modo è possibile eseguire percorsi brevi a grande velocità.

Se il tempo residuo del controllo lo permette, la memoria di preelaborazione viene normalmente riempita.



Programmazione

STARTFIFO

oppure

STOPFIFO

oppure

STOPRE

Parametri

STOPFIFO

Arresto della lavorazione rapida, riempimento il buffer di preelaborazione fino a quando viene riconosciuto STARTFIFO, "Buffer di preelaborazione pieno" oppure "fine programma"

STARTFIFO

Inizio della lavorazione rapida, parallelamente riempimento del buffer di preelaborazione

STOPRE

Arresto dell'avanzamento

Nota

STOPFIFO arresta l'elaborazione finchè la memoria di preelaborazione è piena oppure vengono riconosciuti STARTFIFO o STOPRE .

Esempio di identificazione di una sezione di programma

Il passo di lavorazione da salvare temporaneamente nella memoria di preelaborazione viene evidenziato all'inizio ed alla fine con STOPFIFO oppure STARTFIFO.

```
N10 STOPFIFO
N20...
N100
N110 STARTFIFO
```

L'elaborazione dei blocchi inizia solo quando la memoria di preelaborazione è piena oppure dopo il comando STARTFIFO.

Eccezione:

Il riempimento della memoria di preelaborazione non viene eseguito o interrotto se la fase di lavorazione contiene comandi che forzano un funzionamento non bufferizzato (ricerca del punto di riferimento, funzioni di misura, ...).

Esempio di stop elaborazione con STOPRE

In caso di programmazione di **STOPRE** il blocco successivo viene eseguito solo quando tutti i blocchi preparati e memorizzati sono stati completamente eseguiti. Il blocco precedente viene arrestato con arresto preciso (come G9).

Esempio:

```
N10 ...
N30 MEAW=1 G1 F1000 X100 Y100 Z50
N40 STOPRE
```

Nell'accesso ai dati di stato della macchina (\$SA. . .), il controllo numerico genera un arresto interno della preelaborazione.

Esempio:

```
| R10 = $AA_IM[X]           ;lettura del valore reale dell'asse X
```

**Cautela**

Evitare di programmare STOPRE quando sono attive la correzione utensile e interpolazioni spline, altrimenti vengono interrotte sequenze connesse di blocchi.

9.7 Interruzioni condizionate delle sezioni di programma (DELAYFSTON, DELAYFSTOF)

Funzione

Interruzioni condizionate di sezioni di programma vengono definite Settori Stop Delay, nell'ambito di determinate sezioni di programma non sono consentite **interruzioni** e nemmeno modifiche dell'**avanzamento**. Fondamentalmente, le brevi sezioni di programma che servono per esempio per la realizzazione di una filettatura, devono essere protette da quasi tutti gli eventi di arresto. Un eventuale stop agisce solo al termine di questa sezione di programma.

Programmazione

N... DELAYFSTON Le istruzioni vengono programmate in una riga a sé stante del
N... DELAYFSTOF partprogram. **DELAYFeed STop ON/OF**

Entrambe le istruzioni sono solo consentite in un partprogram e non in azioni sincrone.

Parametri

DELAYFSTON	Definizione dell'inizio di un settore nel quale deve avvenire uno stop "leggero" fino a quando non viene raggiunta la fine del settore Stop Delay.
DELAYFSTOF	Definizione della fine di un settore Stop Delay

Nota

Nel dato macchina DM 11550: STOP_MODE_MASK Bit 0 = 0 (Default) viene definito implicitamente un settore Stop Delay quando sono attive G331/G332 e quando sono stati programmati un avanzamento o G4. Vedere indicazioni sottostanti.

Esempio di eventi di stop

Nel settore di Stop Delay vengono ignorate le **variazioni e gli arresti dell'avanzamento**. Essi agiscono solo dopo il settore di Stop Delay.

Gli eventi di stop vengono distinti in:

eventi di stop "leggeri"	Reazione: ritardata
eventi di stop "duri"	Reazione: immediata

Scelta di alcuni eventi di stop, che eseguono almeno uno stop di breve durata.

9.7 Interruzioni condizionate delle sezioni di programma (DELAYFSTON, DELAYFSTOF)

Nome dell'evento	Reazione	Parametro di interruzione
RESET	immediata	NST: DB21,... DBX7.7 e DB11, ... DBX20.7
PROG_END	Allarme 16954	Prog.NC: M30
INTERRUPT	ritardata	NST: FC-9 e ASUP DB10, ... DBB1
SINGLEBLOCKSTOP	ritardata	Funzionamento in blocco singolo nel settore Stop Delay: l'NC si arresta alla fine del 1° blocco al di fuori del settore Stop Delay. Nel blocco singolo, già prima che venga attivato il settore Stop Delay: NST: "NC-Stop a fine blocco" DB21, ... DBX7.2
STOPPROG	ritardata	NST: DB21,... DBX7.3 e DB11, ... DBX20.5
PROG_STOP	Allarme 16954	Prog.NC: M0 e M1
WAITM	Allarme 16954	Prog.NC: WAITM
WAITE	Allarme 16954	Prog.NC: WAITE
STOP_ALARM	immediata	Allarme: progettazione allarme STOPBYALARM
RETREAT_MOVE_THREAD	Allarme 16954	Prog.NC: allarme 16954 con LFON (Stop & Fastlift in G33 non possibile)
WAITMC	Allarme 16954	Prog.NC: WAITMC
NEWCONF_PREP_STOP	Allarme 16954	Prog.NC: NEWCONF
SYSTEM_SHUTDOWN	immediata	System-Shutdown con 840Di
ESR	ritardata	Funzione ampliata di arresto e svincolo
EXT_ZERO_POINT	ritardata	Spostamento origine esterno
STOPRUN	Allarme 16955	BTSS: PI "_N_FINDST" STOPRUN

Descrizione delle reazioni

immediata (evento di stop "duro")	Stop immediato anche nel settore Stop Delay
ritardata (evento di stop "leggero")	Lo stop (anche breve) avviene solo dopo il settore Stop Delay.
Allarme 16954	Il programma viene interrotto in quanto nel settore di Stop Delay sono state utilizzate istruzioni non consentite.
Allarme 16955	Il programma viene proseguito, nel settore Stop Delay ha avuto luogo un'azione non consentita.
Allarme 16957	Il settore di programma (settore Stop Delay), riportato tra parentesi con DELAYFSTON e DELAYFSTOF, non ha potuto essere attivato. In questo modo ogni stop agisce immediatamente nel settore e non viene ritardato.

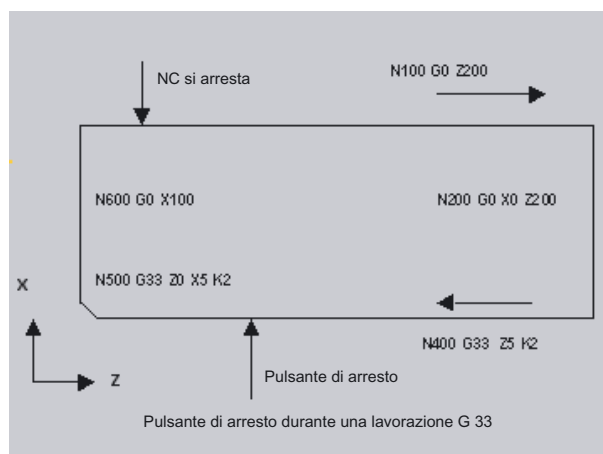
Per una sintesi di altre reazioni agli eventi che provocano lo stop, vedere /FB1/ Manuale delle funzioni di base; BAG, canale, funzionamento del programma (K1), capitolo "Influenza ed effetto sugli eventi di stop".

Esempio di annidamento di settori Stop Delay in due livelli di programma

N10010 DELAYFSTON()	;blocchi con N10xxx livello programma 1
N10020 R1 = R1 + 1	
N10030 G4 F1	;inizio del settore Stop Delay.
...	
N10040 Sottoprogramma2	
...	
...	;interpretazione del sottoprogramma 2
N20010 DELAYFSTON()	;enza effetto, ripete l'inizio, 2.livello
...	
N20020 DELAYFSTOF()	;senza effetto, fine in un altro livello
N20030 RET	
N10050 DELAYFSTOF()	;fine settore Stop Delay nello stesso livello
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	;fine del settore Stop Delay. Gli stop da questo punto ;hanno effetto immediato

Esempio di estratto di programma

In un loop viene ripetuto il seguente blocco di programma:



Dalla figura si capisce che l'utilizzatore ha premuto lo "Stop" nel settore Stop Delay e l'NC inizia la procedura di frenata al di fuori del settore stesso, cioè nel blocco N100. In questo modo l'NC si arresta nel settore precedente a N100.

```

...
N99 MY_LOOP:
N100 G0 Z200
N200 G0 X0 Z200
N300 DELAYFSTON()

```

```

N400 G33 Z5 K2 M3 S1000
N500 G33 Z0 X5 K3
N600 G0 X100
N700 DELAYFSTOF()
N800 GOTOB MY_LOOP

```

Per dettagli sulla ricerca blocco del tipo SERUPRO e gli avanzamenti relativi a G331/G332 durante la maschiatura senza utensile compensato, vedere

/FB1/ Manuale delle funzioni di base; BAG, canale, funzionamento del programma (K1).

/FB1/ Manuale delle funzioni di base; Avanzamenti (V1).

Vantaggi del settore Stop Delay

Una sezione di programma è stata elaborata senza interruzione di velocità.

Se l'utente interrompe il programma con RESET dopo averlo sospeso, il blocco di programma interrotto si trova dopo il settore protetto. Questo blocco di programma è adatto quindi come destinazione di ricerca per una successiva ricerca blocco.

Finché un settore Stop Delay viene elaborato, i seguenti assi dell'elaborazione principale non vengono arrestati:

- assi di comando e
- assi di posizionamento gestiti con POSA

L'istruzione di programma G4 è consentita nel settore Stop Delay tuttavia ci sono altre istruzioni non consentite che provocano un arresto temporaneo (es. WAITM).

G4, come anche un movimento vettoriale, attiva il settore Stop Delay e mantiene quindi la sua efficacia.

Esempio: Interventi sull'avanzamento

Se l'override viene ridotto al 6% prima del settore Stop Delay, esso resta valido anche nel settore Stop Delay.

Se l'override viene ridotto dal 100% al 6% nel settore Stop Delay, quest'ultimo viene eseguito fino alla fine al 100% e successivamente il movimento prosegue al 6%.

Il blocco avanzamento non è efficace nel settore Stop Delay e viene attivato solo all'uscita del settore stesso.

Sovrapposizione/annidamento:

Se due settori Stop Delay si sovrappongono, uno da un'istruzione di programma e l'altro dal dato macchina DM 11550: STOP_MODE_MASK, viene formato il settore Stop Delay più grande possibile.

I seguenti punti regolano la correlazione delle istruzioni di programma DELAYFSTON e DELAYFSTOF con annidamenti e fine programma:

1. con la fine del sottoprogramma, nel quale è stato richiamato DELAYFSTON viene attivato implicitamente DELAYFSTOF.
2. DELAYFSTON Il settore Stop Delay non ha alcun effetto.

3. se il sottoprogramma 1 richiama il sottoprogramma 2 in un settore Stop-Delay, tutto il sottoprogramma 2 è un settore Stop Delay. in particolare DELAYFSTOF nel sottoprogramma 2 non ha alcun effetto.

Nota

REPOSA è un fine sottoprogramma e DELAYFSTON viene in ogni caso disattivato.

Se si verifica un evento di stop “duro” nel settore Stop Delay, quest’ultimo viene completamente disattivato! Se in questa sezione di programma si verifica un ulteriore (qualsiasi) stop, avviene un arresto immediato. Solo una nuova programmazione (nuovo DELAYFSTON) consente di riattivare un nuovo settore Stop Delay.

Se viene premuto il tasto di stop prima del settore Stop Delay e l’NCK deve entrare nel settore Stop Delay per frenare, allora l’NCK frena nel settore Stop Delay e lo stesso resta disattivato!

Se ad un settore Stop Delay si accede con override 0%, il settore Stop Delay **non** viene accettato!

Questo vale per tutti gli eventi di stop “dolci”.

Con STOPALL è possibile frenare nel settore Stop Delay. Con STOPALL vengono però attivati immediatamente tutti gli eventi di stop che finora erano stati ritardati.

Variabili di sistema

Un settore Stop Delay può essere riconosciuto nel partprogram con \$P_DELAYFST. Se il bit 0 della variabile di sistema è settato a 1, l’elaborazione del partprogram in questo momento si trova in un settore Stop Delay.

Un settore Stop Delay può essere riconosciuto nelle azioni sincrone con \$AC_DELAYFST. Se il bit 0 della variabile di sistema è settato a 1, l’elaborazione del partprogram in questo momento si trova in un settore Stop Delay.

Compatibilità

La preimpostazione del dato macchina DM 11550: STOP_MODE_MASK Bit 0 = 0 comporta implicitamente un settore Stop Delay quando vengono programmati codici G del gruppo G331/G332 oppure un movimento vettoriale o G4.

Il bit 0 = 1 consente uno stop quando vengono programmati codici G del gruppo G331/G332 oppure un movimento vettoriale o G4 ((comportamento fino al SW 6). Per la definizione di un settore Stop Delay devono essere utilizzate le istruzioni DELAYFSTON/DELAYFSTOF.

9.8 Inibire la zona di programma per SERUPRO (IPTRLOCK, IPTRUNLOCK)

Funzione

Per determinate situazioni meccaniche complicate della macchina, è necessario impedire la ricerca blocco SERUPRO.

Con un puntatore di interruzione programmabile, si può impostare una possibilità di accesso in fase di "ricerca del punto di interruzione" prima della posizione senza possibilità di ricerca.

Si possono definire anche settori nel partprogram senza possibilità di ricerca nei quali l'NCK non può ancora rientrare. Con l'interruzione del programma, l'NCK si ricorda l'ultimo blocco elaborato che può essere ricercato tramite l'HMI.

Programmazione

N... IPTRLOCK	Le istruzioni vengono programmate da sole in una riga di
oppure	programma e consentono un puntatore di interruzione
N... IPTRUNLOCK	programmabile

Parametri

IPTRLOCK	Inizio della sezione di programma senza possibilità di ricerca
IPTRUNLOCK	Fine della sezione di programma senza possibilità di ricerca

Entrambe le istruzioni sono solo consentite in un partprogram e **non** in azioni sincrone.

Esempio

Annidamenti di sezioni di programma senza possibilità di ricerca in due livelli di programma con IPTRUNLOCK implicito. IPTRUNLOCK implicito nel sottoprogramma 1 conclude il settore senza possibilità di ricerca.

N10010 IPTRLOCK()	
N10020 R1 = R1 + 1	
N10030 G4 F1	;blocco di arresto, inizia la sezione di
...	;programma senza possibilità di ricerca
N10040 Sottoprogramma2	
...	;interpretazione del sottoprogramma 2
N20010 IPTRLOCK ()	;senza effetto, inizio ripetuto
...	
N20020 IPTRUNLOCK ()	;senza effetto, fine in un altro livello
N20030 RET	
...	
N10060 R2 = R2 + 2	

N10070 RET	;Fine della sezione di
	;programma senza possibilità di ricerca
N100 G4 F2	;il programma principale viene proseguito
Una interruzione a 100 restituisce nuovamente	
il puntatore di interruzione.	

Trovare e rilevare i settori senza possibilità di ricerca

Le sezioni di programma senza possibilità di ricerca vengono identificate con le istruzioni IPTRLOCK e IPTRUNLOCK.

L'istruzione IPTRLOCK congela il puntatore di interruzione ad un blocco singolo (SBL1) eseguibile nei cicli di elaborazione principale. Questo blocco viene denominato di seguito come blocco di arresto. Se dopo IPTRLOCK si verifica un'interruzione del programma, tramite la superficie HMI si può ricercare il blocco di arresto citato in precedenza.

Ritornare nuovamente al blocco attuale

Il puntatore di interruzione viene settato con IPTRUNLOCK sul punto di interruzione del blocco attuale per la successiva sezione di programma.

Dopo aver trovato una destinazione di ricerca, con lo stesso blocco di arresto si può ripetere una nuova destinazione di ricerca.

Un puntatore di interruzione editato dall'utilizzatore, deve essere eliminato dalla superficie operativa HMI.

Regole per gli annidamenti

I seguenti punti regolano la correlazione delle istruzioni di programma IPTRLOCK e IPTRUNLOCK con annidamenti e fine programma:

1. con la fine del sottoprogramma, nel quale è stato richiamato IPTRLOCK viene attivato implicitamente IPTRUNLOCK .
2. IPTRLOCK non ha alcun effetto in un settore senza possibilità di ricerca.
3. Se il sottoprogramma1 richiama il sottoprogramma2 in un settore senza possibilità di ricerca, tutto il sottoprogramma2 resta senza possibilità di ricerca. in particolare, IPTRUNLOCK nel sottoprogramma 2 non ha alcun effetto.

Per ulteriori informazioni vedere
/FB1/ Manuale delle funzioni di base; BAG, canale, funzionamento del programma (K1).

Variabile di sistema

Un settore senza possibilità di ricerca può essere riconosciuto nel partprogram con \$P_IPTRLOCK .

Puntatore di interruzione automatico

La funzione „puntatore di interruzione automatico” definisce automaticamente una tipologia di accoppiamento, definita precedentemente, come “senza possibilità di ricerca”. Tramite dato macchina, per

- cambio elettronico con EGON
- accoppiamento assiale tramite valore di riferimento LEADON

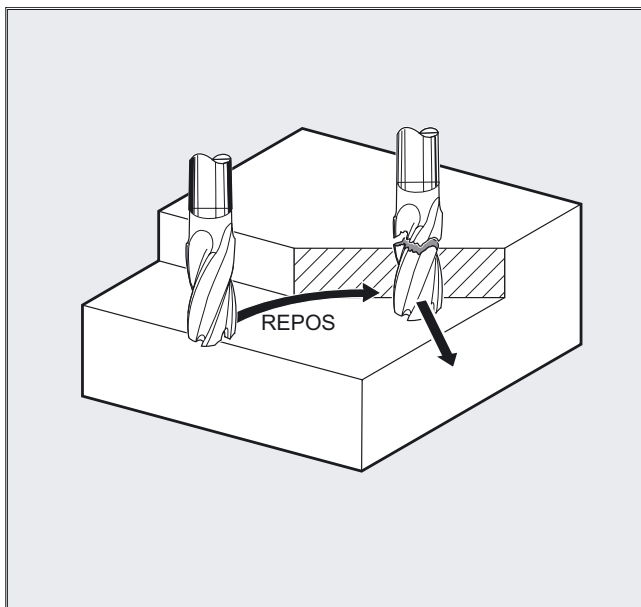
viene attivato il puntatore di interruzione automatico. Se i puntatori di interruzione programmato ed automatico attivabile tramite dato macchina si sovrappongono, allora il settore senza possibilità di ricerca, viene formato più grande possibile.

9.9 Riaccostamento al profilo (REPOSA/L, REPOSQ/H, RMI, RMN, RMB, RME)

Funzione

Se nel corso della lavorazione il programma viene interrotto e l'utensile staccato (per es. a causa della rottura dell'utensile o perché è necessaria una misura), è possibile il riaccostamento al profilo da programma su un punto scelto.

Il comando REPOS agisce come un salto di ritorno da un sottoprogramma (per es. tramite M17). I blocchi successivi della routine di interrupt non vengono più eseguiti.



Per l'interruzione della sequenza di programma vedere anche la sezione "Programmazione NC flessibile", capitolo "Routine di interrupt" in questo manuale di programmazione.

Programmazione

REPOSA RMI DISPR=... oppure REPOSA RMB oppure REPOSA RME oppure REPOSA RMN

oppure

REPOSL RMI DISPR=... oppure REPOSL RMB oppure REPOSL RME oppure REPOSL RMN

oppure

REPOSQ RMI DISPR=...DISR=... oppure REPOSQ RMBDISR=... oppure REPOSQ RME DISR=... oppure REPOSQA DISR=...

oppure

REPOSH RMI DISPR=... DISR=...oppure REPOSH RMB DISR=... oppure REPOSH RME DISR=... oppure

REPOSHA DISR=...

Parametri

Percorso di accostamento

REPOSA	Accostamento su una retta con tutti gli assi
REPOSL	Accostamento su una retta
REPOSQ DISR=...	Accostamento con un quarto di cerchio con raggio DISR
REPOSQA DISR=...	Accostamento con tutti gli assi con un quarto di cerchio con raggio DISR
REPOSH DISR=...	Accostamento su un semicerchio con diametro DISR
REPOSHA DISR=...	Accostamento con tutti gli assi con un semicerchio con raggio DISR

Punto di riposizionamento

RMI	Accostamento al punto di interruzione
RMI DISPR=...	Punto di ingresso a distanza DISPR in mm/inch prima del punto di interruzione
RMB	Accostamento al punto di inizio blocco
RME	Accostamento al punto di fine blocco
RME DISPR=...	Accostamento al punto di fine blocco, con distanza DISPR dal punto finale
RMN	Accostamento al successivo punto vettoriale
A0 B0 C0	Assi con cui deve avvenire l'accostamento

Esempio di accostamento su una retta, REPOSA, REPOSL

L'utensile si avvicina al punto di riaccostamento direttamente con una retta.

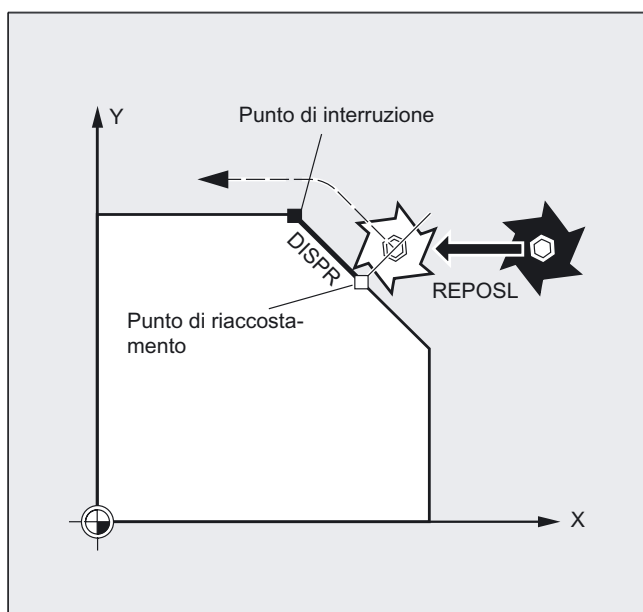
Con REPOSA vengono mossi automaticamente tutti gli assi. Con REPOSL è invece possibile impostare gli assi da muovere.

Esempio:

REPOSL RMI DISPR=6 F400

oppure

REPOSA RMI DISPR=6 F400

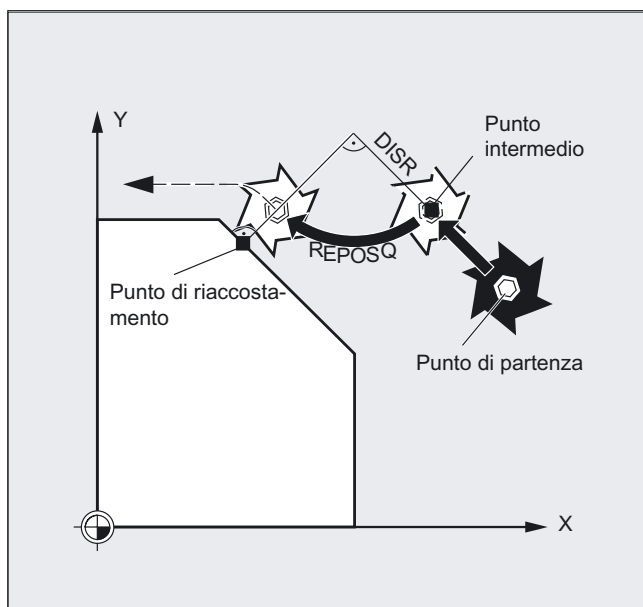


Esempio di accostamento con un arco di cerchio, REPOSQ, REPOSQA

L'utensile raggiunge il punto di riaccostamento compiendo un quarto di cerchio con raggio $DISR = \dots$. Il necessario punto intermedio fra punto di partenza e punto di riaccostamento viene calcolato automaticamente dal controllo numerico.

Esempio:

REPOSQ RMI DISR=10 F400

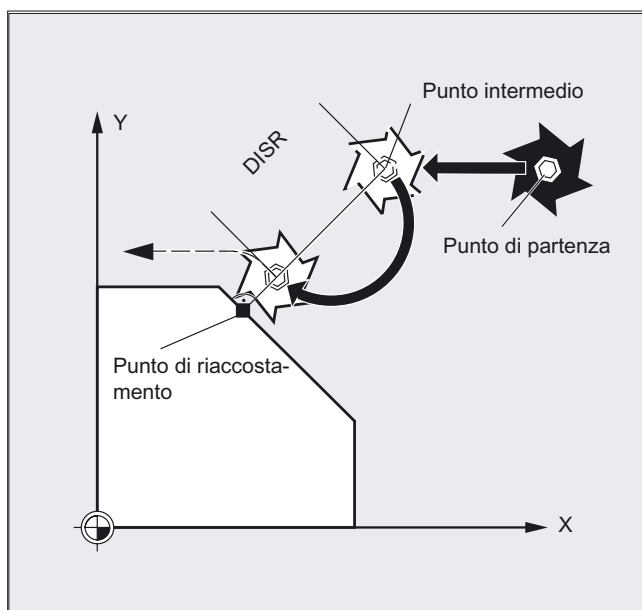


Esempio di accostamento utensile con semicerchio, REPOSH, REPOSHA

L'utensile raggiunge il punto di riaccostamento compiendo un semicerchio con diametro DISR= Il necessario punto intermedio fra punto di partenza e punto di riaccostamento viene calcolato automaticamente dal controllo numerico.

Esempio:

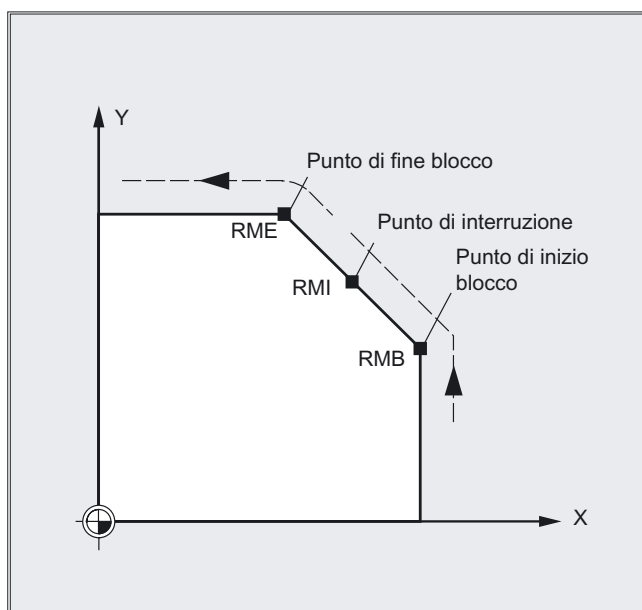
REPOSH RMI DISR=20 F400



Definizione del punto di riaccostamento (non per accostamento SERUPRO con RMN)

In relazione al blocco NC in cui è stato interrotto il programma è possibile scegliere fra tre diversi punti di riaccostamento:

- RMI, punto di interruzione
- RMB, punto di inizio blocco o ultimo punto finale
- RME, punto di fine blocco



Con `RMI DISPR=...` oppure con `RME DISPR=...` si può definire un punto di riaccostamento posto prima del punto di interruzione o della fine blocco.

Con `DISPR=...` si può descrivere il percorso del profilo in mm/inch sul quale si trova il punto di riaccostamento **prima** del punto di interruzione o del punto finale. Questo punto, anche per valori più grandi, può trovarsi al massimo nel punto di inizio blocco.

Se non viene programmato un `DISPR=...`, vale `DISPR=0` e quindi il punto di interruzione (per `RMI`) oppure il punto di fine blocco (per `RME`).

Segno di DISPR

Il segno di `DISPR` viene considerato. Se il segno è positivo, il comportamento è come in precedenza.

Se il segno è negativo, la ripresa avviene monte del punto di interruzione o con `RMB` a monte del punto di partenza.

La distanza tra punto di interruzione e punto di ripresa si ricava dal valore di `DISPR`. Questo punto può trovarsi al massimo sul punto di fine blocco anche per valori più alti.

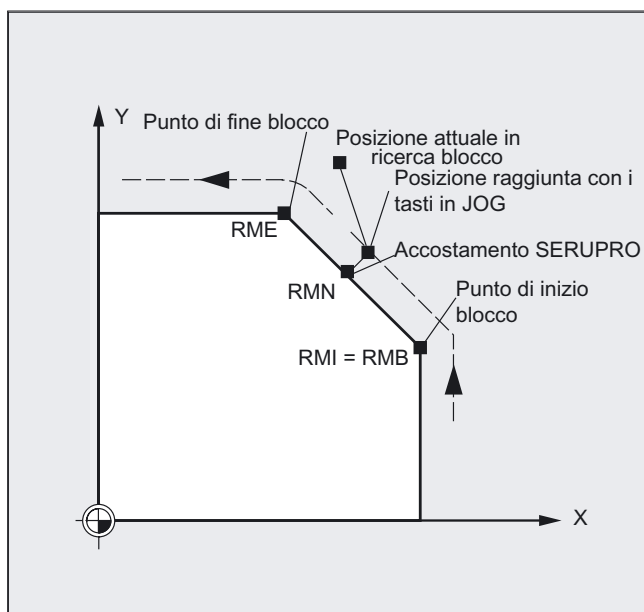
Esempio applicativo:

Con un sensore si individua l'accostamento a una staffa di serraggio. Viene emesso un `ASUP` con cui si aggira la staffa di serraggio.

Infine, con `DISPR` negativo viene eseguito il riposizionamento a monte della staffa di serraggio e il programma viene proseguito.

Accostamento SERUPRO con RMN

Se si verifica un'interruzione della lavorazione in un punto qualsiasi, con SERUPRO avviene l'accostamento al punto di interruzione con RMN per la via più breve per poter eseguire il percorso residuo. A questo scopo l'utilizzatore deve startare una procedura SERUPRO sul blocco di interruzione e posizionare gli assi in JOG prima della posizione danneggiata del blocco di destinazione.



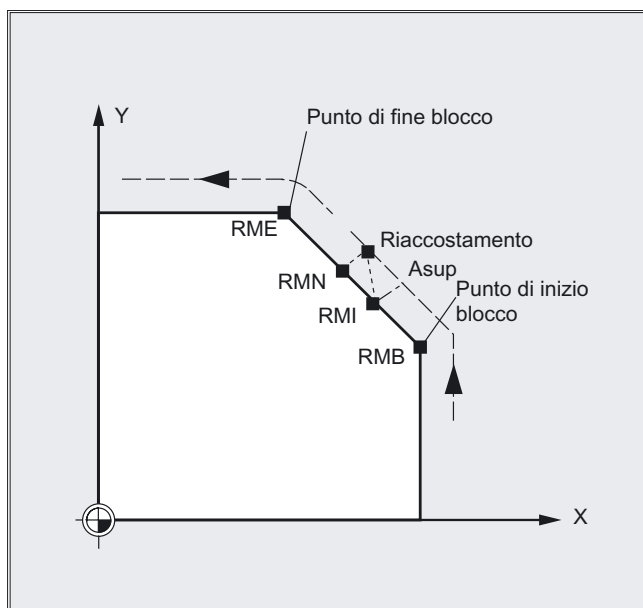
Nota

SERUPRO

Per SERUPRO RMI e RMB sono identici. RMN non si può utilizzare solo con SERUPRO ma in modo generico.

Accostamento sul punto vettoriale successivo RMN

Nel punto di interpretazione di REPOSA dopo un'interruzione, il blocco di riaccostamento con RMN non viene ripetuto completamente ma viene solo eseguito il percorso residuo. Viene accostato il punto vettoriale successivo al blocco di interruzione.



Stato per il modo REPOS valido

Il modo REPOS valido del blocco interrotto può essere letto tramite azioni sincrone con la variabile di sistema `$AC_REPOS_PATH_MODE`.

0: Accostamento non definito

1 RMB: accostamento all'inizio

2 RMI: accostamento nel punto di interruzione

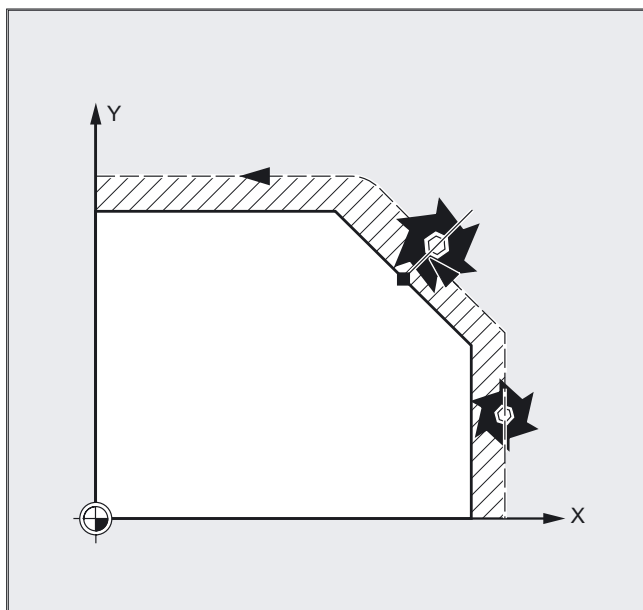
3 RME: accostamento al punto finale del blocco

4 RMN: accostamento al punto vettoriale successivo al blocco di interruzione.

Accostamento con nuovo utensile

Se l'esecuzione del programma è stata interrotta per la rottura di un utensile, programmando il nuovo numero D il programma viene proseguito a partire dal punto di riaccostamento con i valori modificati della correzione utensile.

Se i valori della correzione utensile sono stati modificati può accadere che il punto di interruzione non possa più essere riaccostato. In questo caso viene riaccostato il punto del nuovo profilo che si trova il più vicino possibile al punto di interruzione (modificato eventualmente del valore DISPR).



Accostamento al profilo

Il movimento con cui l'utensile si riaccosta al profilo può essere programmato. Per gli indirizzi degli assi da muovere impostare il valore zero.

Con le istruzioni `REPOSA`, `REPOSQA` e `REPOSHA` vengono riposizionati automaticamente tutti gli assi. Non è necessario fare alcuna impostazione per gli assi.

Programmando `REPOSL`, `REPOSQ` e `REPOSH` gli assi geometrici si muovono automaticamente, anche senza impostazione nel comando. Tutti gli altri assi invece devono essere impostati nel comando.

Per i movimenti circolari `REPOSH` e `REPOSQ` vale quanto segue:

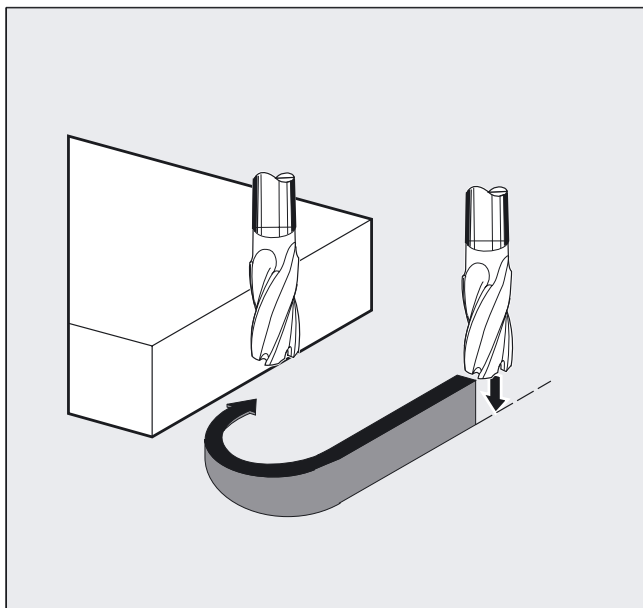
Il cerchio viene eseguito sul piano di lavoro impostato `G17 ... G19`.

Se nel blocco di accostamento viene impostato il terzo asse geometrico (direzione incremento), il punto di riaccostamento viene raggiunto in forma elicoidale, nel caso in cui la posizione dell'utensile e la posizione programmata in direzione dell'incremento non coincidano.

Nei seguenti casi viene automaticamente

commutato in accostamento lineare `REPOSL` :

- non è stato impostato alcun valore per `DISR`.
- manca una direzione di accostamento definita (interruzione programma in un blocco privo di informazione di movimento).
- la direzione di accostamento è perpendicolare al piano di lavoro attuale.



Azioni sincrone al movimento

10.1 Struttura, concetti base generici

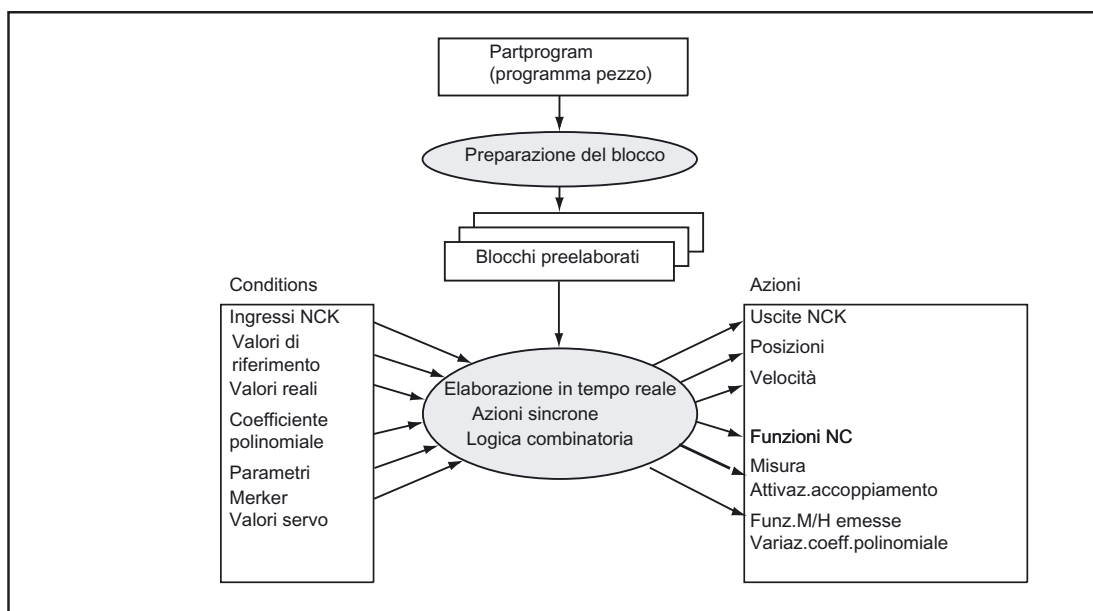
Funzione

Le azioni sincrone offrono la possibilità di eseguire azioni in modo sincrono rispetto ai blocchi di lavorazione.

Il momento di esecuzione delle azioni può essere definito tramite condizioni. Le condizioni vengono sorvegliate nel clock di interpolazione. Le azioni costituiscono quindi una reazione ad eventi in tempo reale; la loro esecuzione non è vincolata a limiti di blocco.

Un'azione sincrona contiene inoltre indicazioni sulla sua durata e sulla frequenza di interrogazione delle variabili di esecuzione programmate, quindi anche alla frequenza di esecuzione delle azioni da avviare. Un'azione sincrona può essere attivata una sola volta oppure ciclicamente (sempre in un clock di interpolazione).

Possibili impieghi:



- Ottimizzazione di operazioni di difficile realizzazione in tempo reale (per es. cambi utensile)
- Reazione rapida ad eventi esterni
- Programmazione di regolazioni AC
- Generazione di funzioni di sicurezza
-

Programmazione

```
DO Azione1 Azione2 ...
PAROLA CHIAVE condizione DO azione1 azione2 ...
ID=n PAROLA CHIAVE condizione DO azione1 azione2 ...
IDS=n PAROLA CHIAVE condizione DO azione1 azione2 ...
```

Elementi di comando

Numero di identificazione ID/IDS:

ID=n	Azioni sincrone modali attive in funzionamento automatico, programma locale ; n = 1... 255
IDS=n	Azioni sincrone modali in ogni modo operativo, statiche ; n = 1... 255
senza ID/IDS	Azioni sincrone con effetto blocco per blocco nel funzionamento automatico

Parola chiave:

Nessuna parola chiave	L'esecuzione dell'azione non è legata a nessuna condizione. Esecuzione ciclica nel clock IPO.
WHEN, WHENEVER, FROM, EVERY	Frequenza di interrogazione dell'azione da partire

Condizione:

Variabile di esecuzione.

Le variabili impiegate vengono valutate nel clock di interpolazione. Le variabili di esecuzione nelle azioni sincrone non attivano gli stop preelaborazione.

Analisi:

Se un partprogram contiene delle variabili di esecuzione (ad es. un valore attuale, la posizione di un ingresso o di un'uscita digitali, ecc.), la preelaborazione viene arrestata fino a quando il blocco precedente è stato eseguito e sono stati emessi i valori delle variabili di esecuzione.

DO:

Esecuzione dell'azione

Coordinamento di azioni sincrone/cicli tecnologici:

CANCEL [n]	Cancellazione delle azioni sincrone
LOCK [n]	Blocco delle azioni sincrone
UNLOCK [n]	Abilitazione delle azioni sincrone
RESET	Reset del ciclo tecnologico

Esempio

```
WHEN $AA_IW[Q1]>5 DO M172 H510
```

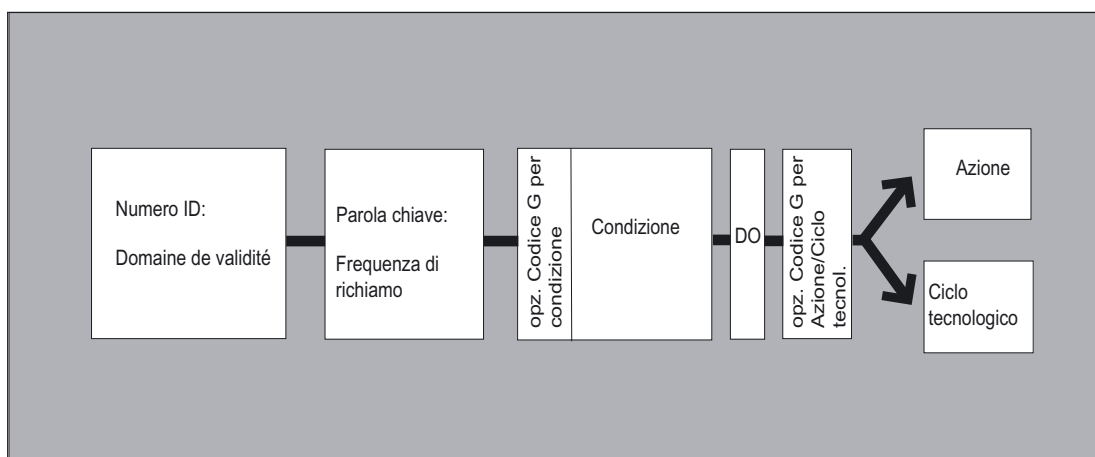
;Se il valore reale dell'asse Q1 supera i 5 mm, vengono inviate all'interfaccia con il PLC le funzioni ausiliarie M172 e H510.

10.1.1 Programmazione ed elementi di comando

Funzione

Un'azione sincrona si trova solo in un blocco e ha effetto a partire dal successivo blocco eseguibile di una funzione macchina (ad es., movimento di avanzamento con G0, G1, G2, G3).

Le azioni sincrone comprendono fino a 5 elementi di comando con varie funzioni:



Programmazione

ID=n Parola chiave Condizione DO Azione 1 Azione 2 ...

Elementi di comando

Numero di identificazione ID/IDS	Campo di validità delle azioni sincrone modali attive nel funzionamento automatico oppure in ogni modo operativo
Parola chiave	Nessuna frequenza di interrogazione, WHEN, WHENEVER, FROM, EVERY
Condizione	Logica combinatoria per variabili di esecuzione; la condizione viene verificata nel clock IPO.
DO	Quando la condizione è soddisfatta viene eseguita l'azione.
Azione	Azione eseguita quando la condizione risulta soddisfatta, es. assegnazione di una variabile.

Ciclo tecnologico	Con condizione soddisfatta, come azione viene eseguito un programma.

Esempio

ID=1	WHENEVER	\$A_IN[1]==1	DO	\$A_OUT[1] = 1
Azione sincrona 1:	ogni qualvolta	è attivo l'ingresso 1	allora	impostaz. uscita 1

10.1.2 Campo di validità: numero identificativo ID**Funzione**

Il campo di validità di un'azione sincrona viene stabilito con il numero di identificazione:

- **ID non modale:** azione sincrona attiva blocco per blocco in funzionamento automatico
- **ID=n** azione sincrona modale in funzionamento automatico al termine del programma
- **IDS=n** azione sincrona modale in ogni modo operativo, statica, anche oltre il termine del programma

Impiego

- Rettifica AC nel funzionamento JOG
- Logica combinatoria per Safety Integrated
- Funzioni di sorveglianza, reazioni agli stati di macchina in tutti i modi operativi

Sequenza operativa

Azioni sincrone attive in maniera modale e statica vengono eseguite secondo i loro numeri ID(S) (nel clock d'interpolazione).

Le azioni sincrone attive blocco per blocco (senza numero ID) vengono eseguite dopo l'esecuzione delle azioni sincrone modali, secondo l'ordine in cui sono state programmate.

Costruttore della macchina

Tramite le impostazioni dei dati macchina è possibile proteggere le azioni sincrone modali dalle modifiche o cancellazioni.

numero identificativo ID

- **ID non modale**

L'azione sincrona è attiva solo nel funzionamento automatico. Vale soltanto per il blocco successivo eseguibile (blocco con istruzioni di movimento o altra azione di macchina), è quindi attiva **blocco-blocco**.

Esempio:

WHEN \$A_IN[3]==TRUE DO \$A_OUTA[4]=10	
G1 X20	;blocco eseguibile

- **ID=n; n=1..255**

L'azione sincrona agisce nei blocchi successivi in modo **modale** ed essere sovrascritta con CANCEL(n) o programmando una nuova azione sincrona con lo stesso ID. Le azioni sincrone attive nel blocco M30 ritardano la fine del programma. Le azioni sincrone ID sono attive **solo** nel **funzionamento automatico**.

Esempio:

```
ID=2 EVERY $A_IN[1]==1 DO POS[X]=0
```

- **IDS=n; n=1..255**

Le azioni sincrone **statiche** agiscono in modo **modale** in **tutti i modi operativi**. Restano attive anche dopo la fine del programma e possono essere attivate subito dopo il Power On con un ASUP.

In tal modo è possibile definire azioni che si intende far eseguire nell'NC indipendentemente dal modo operativo scelto.

Esempio:

```
IDS=1 EVERY $A_IN[1]==1 DO POS[X]=100
```

10.1.3 Verifica ciclica della condizione

Funzione

Tramite una parola chiave viene definita la verifica ciclica della condizione di un'azione sincrona.

Se non è programmata una parola chiave, ciascuna azione sincrona viene eseguita in un clock IPO.

Parole chiavi

Nessuna parola chiave	L'esecuzione dell'azione non è legata a nessuna condizione. L'azione viene eseguita ciclicamente in ogni clock di interpolazione.
WHEN	La condizione viene interrogata ad ogni clock di interpolazione fino a quando non risulta soddisfatta, la relativa azione viene eseguita una sola volta.
WHENEVER	La condizione viene verificata ciclicamente in ogni clock dell'interpolatore. Fintanto che la condizione risulta soddisfatta anche la relativa azione viene eseguita in ogni clock di interpolazione.

FROM	La condizione viene verificata in ogni clock di interpolazione, fino a quando non risulta soddisfatta una volta. A partire da questo momento l'azione viene eseguita fintanto che è attiva l'azione sincrona, cioè anche nel caso in cui la condizione non risulti più soddisfatta.
EVERY	La condizione viene interrogata in ogni clock di interpolazione. L'azione viene eseguita una volta quando è soddisfatta la condizione. Gestita su fronte: l'azione viene rieseguita quando la condizione passa da FALSE a TRUE.

Esempio

Nessuna parola chiave

DO \$A_OUTA[1]=\$AA_IN[X] ;Emissione del valore reale su uscita analogica

EVERY

ID=1 EVERY \$AA_IM[B]>75 DO POS[U]=IC(10) FA[U]=900

;sempre quando il valore reale dell'asse B del sistema di coordinate di macchina supera il valore 75, l'asse U deve progredire con un avanzamento assiale pari a 10.

WHENEVER

WHENEVER \$AA_IM[X] > 10.5*SIN(45) DO ...	;Confronto con l'espressione calcolata nella preelaborazione
WHENEVER \$AA_IM[X] > \$AA_IM[X1] DO ...	;Confronto con un'altra variabile di esecuzione
WHENEVER (\$A_IN[1]==1) OR (\$A_IN[3]==0) DO ...	;Due confronti collegati l'uno con l'altro

Condizione

La condizione rappresenta un'espressione logica formulabile liberamente mediante operatori booleani. Le espressioni booleane devono essere sempre scritte tra parentesi.

La condizione viene verificata in ogni clock di interpolazione.

Prima della condizione è possibile indicare un codice G. In questo modo è possibile ottenere che esistano delle impostazioni per la valorizzazione della condizione e l'azione/il ciclo tecnologico da eseguire indipendentemente dallo stato del partprogram attivo. Il disaccoppiamento, delle azioni sincrone dall'ambiente di programma, è necessario poiché le azioni sincrone devono eseguire le loro azioni in una condizione base definita in determinati momenti, quando sono soddisfatte le condizioni di attivazione.

Casi di impiego

Definizione dei sistemi di misura per la valutazione delle condizioni e l'azione mediante i codici G G70, G71, G700, G710.

Un codice G indicato per la condizione vale per la valutazione della condizione e per l'azione, se per quest'ultima non è indicato alcun codice G proprio.

Per ogni parte della condizione può essere programmato solo **un codice G** del gruppo di codici G.

Possibili condizioni

- Confronto tra variabili di esecuzione (ingressi/uscite analogici/digitali, ecc.)
- Collegamento booleano fra risultati di un confronto
- Calcolo di valori in tempo reale
- Tempo/distanza dall'inizio blocco
- Distanza dalla fine blocco
- Valori di misura, risultati di misura
- Valori servo
- Velocità, stato degli assi

10.1.4 Azioni

Funzione

Nelle azioni sincrone è possibile programmare una o più azioni. Tutte le azioni programmate in un blocco si attivano nello stesso clock di interpolazione.

Elementi di comando

DO	Quando la condizione è soddisfatta viene eseguita l'azione o un ciclo tecnologico.
Azione	Azione attivata quando la condizione risulta soddisfatta. Per es. assegnazione di variabili, attivazione dell'accoppiamento assi, settare uscite NCK, emissione di funzioni M, S e H, predisposizione dei codici G programmati, ...

I **codici G** possono essere programmati in azioni sincrone per le azioni/cicli tecnologici. Per tutte le azioni nel blocco e i cicli tecnologici questo codice G indica eventualmente un codice G diverso da quello impostato per la condizione. Se i cicli tecnologici sono nella parte di azione, il codice G continua ad avere validità modale, anche una volta terminato il ciclo tecnologico, per tutte le azioni seguenti fino al successivo codice G.

Per ogni parte di azione può essere programmato un solo **un codice G** del gruppo di codici G (G70, G71, G700, G710).

Esempio di azione sincrona con due azioni

WHEN \$AA_IM[Y] >= 35.7 DO M135 \$AC_PARAM=50	;se la condizione viene soddisfatta, ;M153 viene emessa al PLC e l'override viene ;settato al 50%.

10.2 Operatori per condizioni e azioni

Confronti
(==, <>, <, >, <=, >=)

operatori booleani
(NOT, AND, OR, XOR)

Operatori a bit
(B_NOT, B_AND, B_OR, B_XOR)

Operazioni di base
(+, -, *, /, DIV, MOD)

Funzioni matematiche
(SIN, COS, TAN, ASIN, ACOS, ABS, TRUNC,
ROUND, LN, EXP, ATAN2, POT, SQRT, CTAB,
CTABINV).

Indicizzazione

Nell'ambito delle condizioni è possibile confrontare variabili o espressioni parziali. Il risultato è sempre del tipo BOOL. Sono ammessi tutti gli operatori di confronto noti.

Variabili, costanti oppure confronti possono essere collegati tra di loro con i già noti operatori booleani.

Sono possibili gli operatori a bit B_NOT, B_AND, B_OR, B_XOR.

Le variabili di esecuzione possono essere collegate, tramite operazioni matematiche, tra di loro o con costanti.

Alle variabili del tipo di dati REAL possono essere applicate funzioni matematiche.

L'indicizzazione è possibile con espressioni di esecuzione.

Esempio

- **Operazioni di base**

Vale il calcolo „punto prima del trattino“, le espressioni in parentesi sono consentite. Gli operatori DIV e MOD sono ammessi anche per dati del tipo REAL

```
DO $AC_PARAM[3] = $A_INA[1] - $AA_IM[Z1]           ; sottrazione di due
                                                    ; variabili di esecuzione
WHENEVER $AA_IM[x2] < $AA_IM[x1] - 1.9 DO $A_OUT[5] = 1
                                                    ; sottrazione di una costante di variabili
DO $AC_PARAM[3] = $INA[1] - 4 * SIN(45.7 $P_EP[Y]) * R4
                                                    ; espressione costante da calcolare in anticipo
```

- **Funzioni matematiche**

```
DO $AC_PARAM[3] = COS($AC_PARAM[1])
```

- **Espressioni in tempo reale**

```
ID=1 WHENEVER ($AA_IM[Y] > 30) AND ($AA_IM[Y] < 40) ; Scelta di una finestra di posizione
DO $AA_OVR[S1] = 80
ID=67 DO $A_OUT[1] = $A_IN[2] XOR $AN_MARKER[1]    ; Elaborazione di 2 segnali booleani
ID=89 DO $A_OUT[4] = $A_IN[1] OR ($AA_IM[Y] > 10)   ; Emissione del risultato di
                                                    ; un confronto
```

- **Variabile di esecuzione indicizzata**

```

WHEN...DO $AC_PARAM[$AC_MARKER[1]] = 3
Non è ammesso
$AC_PARAM[1] = $P_EP[$AC_MARKER]

```

10.3 Variabili di esecuzione per le azioni sincrone

10.3.1 Informazioni generali sulle variabili di sistema

Funzione

Con l'ausilio di variabili di sistema è possibile leggere e scrivere i dati dell'NC. Le variabili di sistema si distinguono in variabili di preelaborazione e variabili di esecuzione. Le variabili di preelaborazione vengono eseguite sempre al momento della preelaborazione. Le variabili di esecuzione determinano il loro valore sempre in rapporto allo stato attuale dell'esecuzione principale.

Sintassi delle variabili di sistema

Il nome delle variabili di sistema inizia generalmente con il carattere \$.

Variabili di preelaborazione:

- \$M . . . , dati macchina
- \$S . . . , dati setting, zone di protezione
- \$T . . . , dati di gestione utensile
- \$P . . . , valori programmati, dati di preelaborazione
- \$C . . . , variabili cicliche dei cicli shell ISO
- \$O . . . , dati opzionali
- R . . . , parametro R

Variabili di esecuzione:

- \$A . . . , dati attuali di esecuzione principale
- \$V . . . , dati servo
- \$R . . . , parametri R

Una seconda lettera descrive la possibilità di accedere alla variabile:

- N . . . , valore globale NCK (di validità generale)
- C . . . , valore specifico di canale
- A . . . , valore specifico dell'asse

La seconda lettera viene generalmente utilizzata solo per le variabili di esecuzione. Le variabili di preelaborazione, ad es. \$P_, vengono generalmente eseguite senza seconda lettera.

Al prefisso (\$) seguito da una o due lettere) seguono sempre un carattere di sottolineatura e il successivo nome di variabile, generalmente nella sua forma inglese o abbreviato.

Tipi di dati

Le variabili di esecuzione possono avere i seguenti tipi di dati:

INT	Integer per i valori interi con segno
REAL	Real per i numeri frazionari razionali
BOOL	Boolean TRUE e FALSE
CHAR	Carattere ASCII
STRING	Stringa di caratteri alfanumerici
AXIS	Indirizzi d'asse e mandrini

Le variabili di preelaborazione possono inoltre avere i seguenti tipi di dati:

FRAME	Trasformazioni delle coordinate

10.3.2 Conversione implicita del tipo

Funzione

Durante le assegnazioni di valori e i trasferimenti di parametri è possibile assegnare o trasmettere variabili di diversi tipi di dati.

La conversione implicita del tipo attiva una conversione del tipo di valori.

Possibili conversioni di tipi

a	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
da							
REAL	sì	sì*	sì ¹⁾	–	–	–	–
INT	sì	sì	sì ¹⁾	–	–	–	–
BOOL	sì	sì	sì	–	–	–	–

Spiegazioni

- * Per la conversione da REAL a INT, un valore frazionario ≥ 0.5 viene arrotondato alla cifra successiva, gli altri valori vengono arrotondati alla cifra precedente (vedere funzione ROUND).
In caso di superamento dei valori viene attivato un allarme.
- ¹⁾ Un valore $\neq 0$ corrisponde a TRUE, un valore $= 0$ corrisponde a FALSE

Risultati

```
Conversione del tipo da REAL o INTEGER a BOOL
Risultato BOOL = TRUE          se il valore di REAL o INTEGER è diverso da zero
Risultato BOOL = FALSE        se il valore di REAL o INTEGER è uguale a zero
Conversione del tipo da BOOL a REAL o INTEGER
Risultato REAL TRUE           se il valore di BOOL = TRUE è (1)
Risultato INTEGER = TRUE      se il valore di BOOL = TRUE è (1)
Conversione del tipo da BOOL a REAL o INTEGER
Risultato REAL FALSE          se il valore di BOOL = FALSE è (0)
Risultato INTEGER = FALSE     se il valore di BOOL = FALSE è (0)
```

Esempi di conversioni implicita del tipo

```
Conversione del tipo da INTEGER a BOOL
$AC_MARKER[1] = 561
ID=1 WHEN $A_IN[1] == TRUE DO $A_OUT[0]=$AC_MARKER[1]

Conversione del tipo da REAL a BOOL
R401 = 100.542
WHEN $A_IN[0] == TRUE DO $A_OUT[2]=$R401

Conversione del tipo da BOOL a INTEGER
ID=1 WHEN $A_IN[2] == TRUE DO $AC_MARKER[4] = $A_OUT[1]]

Conversione del tipo da BOOL a REAL
R401 = 100.542
WHEN $A_IN[3] == TRUE DO $R10 = $A_OUT[3]
```

10.3.3 Variabili GUD per azioni sincrone

Funzione

Oltre alle variabili predefinite, il programmatore può utilizzare nelle azioni sincrone le variabili GUD speciali. Le variabili vengono visualizzate su HMI nel settore operativo Parametri e possono essere utilizzate nel wizard, nella visualizzazione variabili e nel protocollo variabili.

Settori progettabili dei parametri

Costruttore della macchina

Tramite i dati macchina è possibile estendere i singoli blocchi GUD per il tipo di dati REAL, INT e BOOL a ulteriori campi di parametro specifici per canale del tipo di dati AXIS, CHAR e STRING, leggibili e scrivibili dal partprogram e anche mediante azioni sincrone.

I parametri sono disponibili dopo l'impostazione dei relativi dati macchina al successivo avvio del controllore.

Per la progettazione dei relativi dati macchina, vedere le indicazioni del costruttore della macchina.

Variabili predefinite

Nota

Anche se non è attivo alcun file di definizione GUD, è possibile leggere nel rispettivo blocco GUD, a partire dall'HMI, i nuovi parametri definiti tramite i dati macchina.

Lista dei nomi delle variabili predefinite			
Nome delle Synact-GUD			
del tipo di dati REAL	del tipo di dati INT	del tipo di dati BOOL	nel blocco
SYG_RS[]	SYG_IS[]	SYG_BS[]	blocco SGUD
SYG_RM[]	SYG_IM[]	SYG_BM[]	blocco MGUD
SYG_RU[]	SYG_IU[]	SYG_BU[]	blocco UGUD
SYG_R4[]	SYG_I4[]	SYG_B4[]	blocco GUD4
SYG_R5[]	SYG_I5[]	SYG_B5[]	blocco GUD5
SYG_R6[]	SYG_I6[]	SYG_B6[]	blocco GUD6
SYG_R7[]	SYG_I7[]	SYG_B7[]	blocco GUD7
SYG_R8[]	SYG_I8[]	SYG_B8[]	blocco GUD8
SYG_R9[]	SYG_I9[]	SYG_B9[]	blocco GUD9

Lista dei nomi delle variabili predefinite			
Nome delle Synact-GUD			
del tipo di dati AXIS	del tipo di dati CHAR	del tipo di dati STRING	nel blocco
SYG_AS[]	SYG_CS[]	SYG_SS[]	blocco SGUD
SYG_AM[]	SYG_CM[]	SYG_SM[]	blocco MGUD
SYG_AU[]	SYG_CU[]	SYG_SU[]	blocco UGUD
SYG_A4[]	SYG_C4[]	SYG_S4[]	blocco GUD4
SYG_A5[]	SYG_C5[]	SYG_S5[]	blocco GUD5
SYG_A6[]	SYG_C6[]	SYG_S6[]	blocco GUD6
SYG_A7[]	SYG_C7[]	SYG_S7[]	blocco GUD7
SYG_A8[]	SYG_C8[]	SYG_S8[]	blocco GUD8
SYG_A9[]	SYG_C9[]	SYG_S9[]	blocco GUD9

Nota

Le variabili del tipo STRING nelle azioni sincrone hanno una lunghezza fissa di 32 caratteri.

- Dimensione del campo corrispondente al <valore> del dato macchina
- Nomi predefiniti secondo l'elenco precedente dei nomi di variabile predefiniti.
- L'accesso tramite HMI avviene come l'accesso ai GUD creati mediante file di definizione.
- Le assegnazioni dei livelli di protezione, che sono già possibili in un file di definizione GUD mediante parole chiave APR e APW, continuano a restare valide e si riferiscono solo ai GUD specificati in questo file di definizione.
- Comportamento in cancellazione: se il contenuto di un determinato file di definizione GUD viene riattivato, il vecchio blocco dati GUD viene cancellato nella memoria del file system attivo. In questa occasione i parametri vengono ripristinati. Questo processo è possibile anche tramite HMI nel settore operativo Servizi nella superficie operativa "Definizione e attivazione dei dati utente (GUD)".

10.3.4 Identificatore asse di default (NO_AXIS)

Funzione

Alle variabili o ai parametri del tipo AXIS non inizializzati con un valore è possibile assegnare degli identificatori asse di default definiti. Anche le variabili d'asse non definite vengono inizializzate con questo valore di default.

I nomi d'asse validi non inizializzati vengono rilevati mediante una richiesta della variabile "NO_AXIS" nelle azioni sincrone. A questo identificatore asse non inizializzato viene assegnato l'identificatore asse di default progettato tramite un dato macchina.

Costruttore della macchina

Mediante i dati macchina è necessario definire e impostare almeno un identificatore asse esistente valido. Si possono tuttavia impostare anche tutti gli identificatori asse validi esistenti. Fare attenzione alle indicazioni del costruttore della macchina

Nota

Alle nuove variabili create viene così automaticamente assegnato durante la definizione il valore registrato nel dato macchina per i nomi asse di default. Per ulteriori informazioni su una definizione valida tramite dato macchina vedere

Bibliografia

/FBSY/ Descrizione delle funzioni, azioni sincrone

Programmazione

```
PROC UP (AXIS PAR1=NO_AXIS, AXIS PAR2=NO_AXIS)
IF PAR1 <>NO_AXIS...
```

Definizione di un sottoprogramma

PROC	Definizione di un sottoprogramma
UP	Nome di sottoprogramma per il rilevamento
PARn	Parametro n
NO_AXIS	Inizializzazione del parametro formale con identificatore asse di default

Esempio per la definizione di una variabile asse nel programma principale

```
DEF AXIS AXVAR
UP( , AXVAR)
```

10.3.5 Marker per azioni sincrone \$AC_MARKER[n]**Funzione**

La variabile di campo \$AC_MARKER[n] può essere letta e scritta nelle azioni sincrone. Queste variabili possono risiedere nella memoria del file system attivo o passivo.

Variabile di sincronizzazione: tipo di dati INT

\$AC_MARKER [n]	Merker/contatori specifici per canale del tipo di dati INTEGER
\$MC_MM_NUM_AC_MARKER	Dato macchina per l'impostazione del numero di merker specifici di canale per le azioni sincrone di movimento
n	Indice di campo delle variabili 0-n

Esempio di lettura/scrittura di variabili marker

```

WHEN ... DO $AC_MARKER[0] = 2
WHEN ... DO $AC_MARKER[0] = 3
WHENEVER $AC_MARKER[0] == 3 DO $AC_OVR=50

```

10.3.6 Parametri per azioni sincrone \$AC_PARAM[n]**Funzione**

I parametri di azioni sincrone \$AC_PARAM[n] servono per i calcoli e come memoria intermedia in azioni sincrone. Queste variabili possono risiedere nella memoria del file system attivo o passivo.

Variabile di sincronizzazione: tipo di dati REAL

Tali parametri sono contenuti con lo stesso nome, una volta in ogni canale.

\$AC_PARAM[n]	Variabile di calcolo per le azioni sincrone di movimento (REAL)
\$MC_MM_NUM_AC_PARAM	Dato macchina per l'impostazione del numero dei parametri per le azioni sincrone di movimento fino a 20000 max.
n	Indice di campo del parametro 0-n

Esempio di parametri per azioni sincrone \$AC_PARAM[n]

```

$AC_PARAM[0]=1.5
$AC_MARKER[0]=1
ID=1 WHEN $AA_IW[X]>100 DO $AC_PARAM[1]=$AA_IW[X]
ID=2 WHEN $AA_IW[X]>100 DO $AC_MARKER[1]=$AC_MARKER[2]

```

10.3.7 Parametri di calcolo \$R[n]

Funzione

Questa variabile statica di campo serve per operazioni di calcolo nel partprogram e nelle azioni sincrone.

Programmazione

Programmazione nel partprogram:

```
REAL R[n]
```

oppure

```
REAL Rn
```

Programmazione nelle azioni sincrone:

```
REAL $R[n]
```

oppure

```
REAL $Rn
```

Parametri di calcolo

L'uso di parametri di calcolo consente:

- la memorizzazione di valori che devono mantenere validità anche dopo la fine programma, Reset NC e Power On.
- La visualizzazione di valori memorizzati nella maschera di parametri R.

Esempi

WHEN \$AA_IM[X] >=40.5 DO \$R10=\$AA_MM[Y]	;Impiego di R10 in un'azione sincrona
G01 X500 Y70 F1000	
STOPRE	;stop preelaborazione
IF R10>20	;Valutazione della variabile di calcolo

WHEN \$AA_IM[X] >=40.5 DO \$R10=\$AA_MM[Y]	;Accesso in lettura del parametro R10.
WHEN \$AA_IM[X] >=6.7 DO	;Accesso in lettura al parametro R
\$R[\$AC_MARKER[1]] =30.6	;il cui numero si trova nel merker 1

```

SYG_AS[2]=X
SYG_IS[1]=1
WHEN $AA_IM[SGY_AS[2]]>10 DO $R3=$AA_EG_DENOM[SYG_AS[1]], SYG_AS[2]]
WHEN $AA_IM[SGY_AS[2]]>12 DO $AA_SCTRACE[SYG_AS[2]]=1

SYG_AS[1]=X

```

```

SYG_IS[0]=1
WHEN $AA_IM[SGY_AS[1]]>10 DO $R3=$$MA_POSCTRL_GAIN[SYG_IS[0]],SYG_AS[1]]
WHEN $AA_IM[SGY_AS[1]]>10 DO $R3=$$MA_POSCTRL_GAIN[SYG_AS[1]]
WHEN $AA_IM[SGY_AS[1]]>15 DO $$MA_POSCTRL_GAIN[SYG_AS[0]],SYG_AS[1]]=$R3

```

10.3.8 Lettura/scrittura dei dati macchina NC e dei dati setting NC

Funzione

La lettura e la scrittura di dati macchina e dati setting NC è anche possibile da azioni sincrone. Durante la lettura e scrittura di elementi di campo dei dati macchina è possibile omettere un indice durante la programmazione. Se ciò accade nel partprogram, durante la lettura viene letto il **primo** elemento di campo e durante la scrittura tutti gli elementi del campo sono descritti con il valore.

In questo caso, nelle azioni sincrone viene letto o scritto solo il **primo** elemento.

Definizione

DM, DS con

\$: lettura del valore al momento dell'interpretazione delle azioni sincrone

\$\$: lettura del valore nell'esecuzione principale

Lettura dei valori DM e DS al momento della preelaborazione

Vengono indirizzati dall'azione sincrona con il carattere \$ e valutati al momento della preelaborazione.

```

ID=2 WHENEVER $AA_IM[z]<$$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
;In questo caso per il pendolamento viene richiamato il settore di inversione 2
considerato come costante

```

Lettura dei valori DM e DS al momento dell'esecuzione

Vengono indirizzati dall'azione sincrona con il carattere \$\$ e valutati al momento dell'esecuzione.

```

ID=1 WHENEVER $AA_IM[z]<$$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
;In questo caso si presuppone che la posizione di inversione possa essere modificata
dalla superficie HMI durante la lavorazione.

```

Scrittura dei valori DM e DS al momento dell'esecuzione

il diritto di accesso attualmente impostato deve permettere anche l'accesso in scrittura. L'efficacia per tutti i DM e DS è indicata nella **bibliografia**: /LIS/, Liste (volume 1).

I dati DM e DS da scrivere vanno indirizzati facendoli precedere dal prefisso \$\$.

Esempio

```
ID=1 WHEN $AA_IW[X]>10 DO $$SN_SW_CAM_PLUS_POS_TAB_1[0]=20
                                $$$SN_SW_CAM_MINUS_POS_TAB_1[0]=30
;modifica della posizione di comando dalle camme software. Nota: devono essere
modificate 2-3 clock IPO prima del raggiungimento della posizione.
```

10.3.9 Variabile timer \$AC_Timer[n]**Funzione**

La variabile di sistema \$AC_TIMER[n] consente di avviare azioni dopo periodi di attesa prestabiliti.

Variabile timer: tipo di dati REAL

\$AC_TIMER[n]	Timer specifico per canale del tipo di dati REAL
s	Unità in secondi
n	Indice della variabile Timer

Impostazione del timer

Settare TimerIl conteggio di una variabile timer viene avviato assegnando il valore

\$AC_TIMER[n] = value

n: numero della variabile di tempo

value: valore iniziale (normalmente 0)

Arresto del timer

Il conteggio di una variabile contatore viene arrestato assegnando un valore negativo

\$AC_TIMER[n] = -1

Lettura del timer

Il valore in tempo attuale può essere letto con variabile timer attiva o ferma. Dopo l'arresto di una variabile timer assegnando -1, l'ultimo valore permane e può essere letto.

Esempio

Emissione di un valore reale attraverso l'uscita analogica

500 ms dopo il riconoscimento di un ingresso digitale

```
WHEN $A_IN[1]==1 DO $AC_TIMER[1]=0 ;resettare ed avviare timer
WHEN $AC_TIMER[1]>=0.5 DO $A_OUTA[3]=$AA_IM[X] $AC_TIMER[1]=-1
```

10.3.10 Variabile FIFO \$AC_FIFO1[n] ... \$AC_FIFO10[n]

Funzione

Per la memorizzazione di sequenze di dati relazionate sono disponibili 10 variabili FIFO (memoria ad anello).

Tipo di dati: REAL

Utilizzo:

- misure cicliche
- ripetizioni

Per ogni elemento è possibile l'accesso in lettura e scrittura.

Variabile FIFO

Il numero di variabili FIFO disponibili di volta in volta viene stabilita nel dato macchina

DM 28260: NUM_AC_FIFO .

Il numero di valori che possono essere scritti in una variabile FIFO viene stabilita con il dato macchina

DM 28264: LEN_AC_FIFO . Tutte le variabili FIFO hanno la stessa lunghezza.

La somma degli elementi FIFO viene solo formata quando nel DM 28266:

MODE_AC_FIFO è settato il bit 0.

Gli indici da 0 a 5 hanno un significato particolare:

n=0: In scrittura: il nuovo valore viene depositato nel FIFO

In lettura: l'elemento depositato da più tempo viene letto e rimosso dal FIFO

n=1: accesso all'elemento depositato da più tempo

n=2: accesso all'elemento depositato da meno tempo

n=3: somma di tutti gli elementi FIFO

n=4: numero di elementi disponibili nel FIFO.

Per ogni elemento del FIFO è possibile l'accesso in lettura e scrittura. Il reset delle variabili FIFO avviene resettando il numero degli elementi, es. per la prima variabile FIFO:

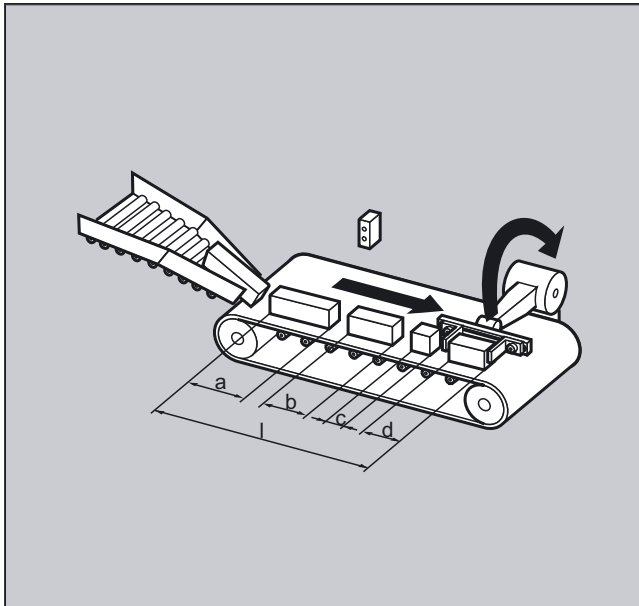
\$AC_FIFO1[4] = 0

n=5: indice di scrittura attuale relativo all'inizio del FIFO

n=6 fino a 6+nmax: accesso all'elemento FIFO n

Esempio: memoria ad anello

Nel corso di un processo produttivo un nastro trasportatore viene impiegato per il trasporto di prodotti di lunghezza diversa (a, b, c, d). Sul nastro trasportatore di lunghezza "l" vengono quindi trasportate contemporaneamente quantità diverse di prodotti, a seconda delle lunghezze dei prodotti in questione. Tenendo conto che la velocità di trasporto non cambia, occorre adattare il prelevamento dei prodotti dal nastro ai tempi variabili del loro arrivo.



```

DEF REAL ZWI=2.5
DEF REAL GESAMT=270

EVERY $A_IN[1]==1 DO $AC_FIFO1[4]=0
EVERY $A_IN[2]==1 DO $AC_TIMER[0]=0

EVERY $A_IN[2]==0 DO $AC_FIFO1[0]=$AC_TIMER[0]*$AA_VACTM[B]
;Se la barriera luminosa è libera, calcolare la lunghezza del prodotto, attraverso
il
tempo misurato e la velocità del trasporto, dopodiché memorizzarla nel FIFO
EVERY $AC_FIFO1[3]+$AC_FIFO1[4]*ZWI>=GESAMT DO POS[Y]=-30
$R1=$AC_FIFO1[0]
;Non appena la somma di tutte le lunghezze dei prodotti e delle distanze tra gli
stessi è
maggiore/uguale alla lunghezza tra la posizione di prelievo e di deposito, prelevare
il prodotto
dal nastro trasportatore e leggere la relativa lunghezza del prodotto dal FIFO.

```

;Spazio sempre uguale fra i prodotti sul
nastro.

;Distanza fra la posizione di misura della
lunghezza e posizione di prelevamento dal
nastro.

;Resetare FIFO all'inizio
del processo.

;Startare il tempo quando un prodotto arriva
davanti alla fotocellula.

10.3.11 Informazione sui tipi di blocco nell'interpolatore

Funzione

Per le azioni sincrone sono disponibili le seguenti variabili di sistema che forniscono un'indicazione sul blocco attualmente in elaborazione:

\$AC_BLOCKTYPE

\$AC_BLOCKTYPEINFO

\$AC_SPLITBLOCK

Variabile Blocktype e Blocktypeinfo

\$AC_BLOCKTYPE		\$AC_BLOCKTYPEINFO					
Valore:		Valore:					
0	diverso da 0	T	H	Z	E	significato:	
Blocco originale	Blocco intermedio					esecutore per blocco intermedio:	
	1	1	0	0	0	Blocco generato internamente, nessuna ulteriore indicazione	
	2	2	0	0	1	Smussi/raccordi: Retta	
	2	2	0	0	2	Smussi/raccordi: Cerchio	
	3	3	0	0	1	WAB: accostamento con retta	
	3	3	0	0	2	WAB: accostamento con quarto di cerchio	
	3	3	0	0	3	WAB: accostamento con semicerchio	
						Correzione utensile:	
	4	4	0	0	1	Blocco di accostamento dopo STOPRE	
	4	4	0	0	2	Blocchi di collegamento nel caso di punto di intersezione non trovato	
	4	4	0	0	3	Cerchio a punti su spigoli interni (solo con TRACYL)	
	4	4	0	0	4	Cerchio di aggiramento (opp.sezione di cono) sugli spigoli esterni	
	4	4	0	0	5	Blocchi di accostamento con correzione disattivata	
	4	4	0	0	6	Blocchi di accostamento con CRU riattivata	
	4	4	0	0	7	Suddivisione del blocco a causa di una curvatura eccessiva	
	4	4	0	0	8	Blocchi di compensazione con fresatura frontale 3D (vettore utensile vettore della superficie)	
						Raccordatura tramite:	
	5	5	0	0	1	G641	
	5	5	0	0	2	G642	
	5	5	0	0	3	G643	
	5	5	0	0	4	G644	

\$AC_BLOCKTYPE		\$AC_BLOCKTYPEINFO				
Valore:		Valore:				
0	diverso da 0	T	H	Z	E	significato:
Blocco originale	Blocco intermedio					esecutore per blocco intermedio:
						Blocco TLIFT con:
	6	6	0	0	1	Movimento lineare dell'asse tangenziale e senza movimento di svincolo
	6	6	0	0	2	Movimento non lineare dell'asse tangenziale (polinomio) e senza movimento di svincolo
	6	6	0	0	3	Movimento di svincolo, il movimento dell'asse tangenziale e quello di svincolo vengono avviati contemporaneamente
	6	6	0	0	4	Movimento di svincolo, l'asse tangenziale si avvia solo quando la posizione di svincolo è stata raggiunta.
						Suddivisione del percorso:
	7	7	0	0	1	suddivisione del percorso programmata senza che siano attivi una punzonatura o roditura.
	7	7	0	0	2	suddivisione del percorso programmata con punzonatura o roditura attivi
	7	7	0	0	3	suddivisione del percorso generata automaticamente internamente
						Cicli compilati:
	8	Applicazione ID			ID dell'applicazione dei cicli compilati che ha generato il blocco	

Nota

Nella posizione delle migliaia (T) \$AC_BLOCKTYPEINFO contiene sempre anche il valore del tipo di blocco nel caso che sia presente un blocco intermedio. In \$AC_BLOCKTYPE diverso da 0 non viene registrata la posizione delle migliaia.

T: Posizione delle migliaia

H: Posizione delle centinaia:

Z: Posizione delle decine:

E: Posizione delle unità

\$AC_SPLITBLOCK	
Valore:	significato:
0	Blocco programmato immutato (un blocco generato dal compressore viene anche trattato come blocco programmato)
1	È presente un blocco generato internamente oppure un blocco originale accorciato
3	È presente l'ultimo blocco di una catena di blocchi generati internamente oppure di blocchi originali accorciati

Esempio di conteggio di blocchi di raccordo

```
$AC_MARKER[0]=0
$AC_MARKER[1]=0
$AC_MARKER[2]=0
...
;definizione di azioni sincrone con le quali
;vengono conteggiati i blocchi di raccordo
; tutti i blocchi di raccordo vengono conteggiati in $AC_MARKER[0]
ID = 1 WHENEVER ($AC_TIMEC ==0) AND ($AC_BLOCKTYPE==5) DO _
  $AC_MARKER[0]= $AC_MARKER[0] + 1
...
; i blocchi di raccordo generati con G641 vengono conteggiati in $AC_MARKER[1]
ID = 2 WHENEVER ($AC_TIMEC ==0) AND ($AC_BLOCKTYPEINFO==5001) DO _
  $AC_MARKER[1]= $AC_MARKER[1] + 1
...
; i blocchi di raccordo generati con G642 vengono conteggiati in $AC_MARKER[2]
ID = 3 WHENEVER ($AC_TIMEC ==0) AND ($AC_BLOCKTYPEINFO==5002) DO _
  $AC_MARKER[2]= $AC_MARKER[2] + 1
...
```

10.4 Azioni nelle azioni sincrone

10.4.1 Panoramica

Generalità

Nelle azioni sincrone, le azioni sono costituite da assegnazioni di valori, chiamate di funzioni o parametri, parole chiave o cicli tecnologici.

Gli operatori rendono possibili le esecuzioni complesse.

Le azioni sincrone vengono aggiornate continuamente in diverse versioni di software per espressioni, variabili utilizzabili durante il ciclo e condizioni complesse nelle azioni sincrone.

Sono possibili le seguenti applicazioni:

- Calcoli di espressioni complesse nel clock IPO
- Movimenti asse e comandi mandrino
- modificare ed elaborare online i dati setting da azioni sincrone come ad esempio l'emissione di posizioni e tempi di camme software al PLC o all'NC
- Emissione funzioni ausiliarie al PLC

- Realizzazione di ulteriori funzioni di sicurezza
- impostazione del movimento sovrapposto, della correzione utensile online e della regolazione della distanza
- Esecuzione di azioni in tutti i modi operativi
- Influenzare le azioni sincrone dal PLC
- Esecuzione di cicli tecnologici
- Emissione di segnali digitali ed analogici
- Rilevamento delle prestazioni delle azioni sincrone nel clock dell'interpolatore e del tempo di calcolo del regolatore di posizione per valutare il fattore di utilizzo
- Possibilità di diagnosi nella superficie operativa

Possibilità di impiego di azioni sincrone di movimento	
Azione sincrona	Descrizione
DO \$V...= DO \$A...=	Assegnazione (valori servo); assegnazione di variabili (variabile di esecuzione)
DO \$AC...[n]= DO \$AC_MARKER[n]= DO \$AC_PARAM[n]=	Variabile di esecuzione speciale; lettura o scrittura dei marker per azione sincrona, lettura o scrittura dei parametri per azione sincrona
DO \$R[n]=	Lettura o scrittura della variabile di calcolo
DO \$MD...= DO \$\$SD...=	Lettura del valore DM al momento dell'interpolazione, scrittura del valore DS nell'esecuzione principale
DO \$AC_TIMER[n]=valore iniziale	Timer
DO \$AC_FIFO1[n] ...FIFO10[n]=	Variabile FIFO
DO \$AC_BLOCKTYPE= DO \$AC_BLOCKTYPEINFO= DO \$AC_SPLITBLOCK=	Interpretazione del blocco attuale (variabile di esecuzione)
DO M, S e H, ad es. M07	Emissione di funzioni ausiliarie M, S e H
DO RDISABLE	Settare il blocco lettura
DO STOPREOF	Eliminare il blocco preelaborazione
DO DELDTG	Cancellazione rapida del percorso residuo senza arresto avanzamento
FTCDEF(polin., LL, UL , coeff.) DO SYNFACT(polin., uscita, ingresso)	Definizione di polinomi; attivazione di funzioni sincrone: Regolazione AC
DO FTOC	Correzione utensile online
DO G70/G71/G700/G710	Definizione del sistema di misura per compiti di posizionamento Impostazione delle quote in inch o metrica
DO POS[asse]= / DO MOV[asse]= DO SPOS[mandrino]=	Avvio/posizionamento/arresto degli assi di comando; avvio/posizionamento/arresto del mandrino
DO MOV[asse] = valore	Avvio/arresto dei movimenti senza fine di un asse di comando
DO POS[asse]= FA [asse]=	Avanzamento assiale FA
DO \$A_WORAREA_PLUS_ENABLE]=	Limitazione del campo di lavoro
ID=1 ... DO POS[asse]= FA [asse]= ID=2 ... DO POS[asse]= \$AA_IM[asse] FA [asse]=	Posizionamenti da azioni sincrone
DO PRESETON (asse, valore)	Impostazione valore reale (Preset da azioni sincrone)

Possibilità di impiego di azioni sincrone di movimento	
ID=1 EVERY \$A_IN[1]=1 DO M3 S.... ID=2 EVERY \$A_IN[2]=1 DO SPOS=	Avvio/posizionamento/arresto mandrini
DO TRAILON(FA, LA, fattore di accoppiamento) DO LEADON(FA, LA, NRCTAB, OVW)	Attivazione del trascinamento; attivazione dell'accoppiamento del valore master
DO MEAWA(asse)= DO MEAC(asse)=	Attivazione misura assiale; attivazione misura continua
DO [campo n, m]=SET(valore, valore, ...) DO [campo n, m]=REP(valore, valore, ...)	Inizializzazione delle variabili di campo con liste di valori; inizializzazione delle variabili di campo con uguale valore
DO SETM(n. marker) DO CLEARM(n. marker)	Impostazione label di attesa Cancellazione label di attesa
DO SETAL(Nr.allarme)	Impostazione allarmi dei cicli (funzione di sicurezza supplementare)
DO FXS[asse]= DO FXST[asse]= DO FXSW[asse]= DO FOCON[asse]= DO FOCOF[asse]=	Selezione del movimento contro riscontro fisso; modifica della coppia di serraggio; modifica della finestra di sorveglianza; attivazione del movimento con coppia/forza limitata (modale), disattivazione FOC (l'azione sincrona si attiva in riferimento al blocco)
ID=2 EVEREY \$AC_BLOCKTYPE==0 DO \$R1 = \$AC_TANEB	Angolo tra la tangente vettoriale nel punto finale del blocco attuale e la tangente vettoriale nel punto finale del blocco successivo programmato
DO \$AA_OVR= DO \$AC_OVR= DO \$AA_PLC_OVR DO \$AC_PLC_OVR DO \$AA_TOTAL_OVR DO \$AC_TOTAL_OVR	Override assiale; override vettoriale; override assiale predefinito da PLC; override vettoriale predefinito da PLC; override assiale risultante; override vettoriale risultante
\$AN_IPO_ACT_LOAD= \$AN_IPO_MAX_LOAD= \$AN_IPO_MIN_LOAD= \$AN_IPO_LOAD_PERCENT= \$AN_SYNC_ACT_LOAD= \$AN_SYNC_MAX_LOAD= \$AN_SYNC_TO_IPO=	Tempo di calcolo IPO attuale; tempo di calcolo IPO più lungo; tempo di calcolo IPO più breve; tempo di calcolo IPO attuale in relazione al clock IPO; tempo di calcolo attuale per l'azione sincrona in tutti i canali; tempo di calcolo più lungo per l'azione sincrona in tutti i canali; componente percentuale di tutta l'azione sincrona
DO TECCYCLE	Esecuzione di ciclo tecnologico
DO LOCK(n, n, ...) DO UNLOCK(n, n, ...) DO RESET(n, n, ...)	Blocco, abilitazione, RESET di un ciclo tecnologico
CANCEL(n, n, ...)	Cancellazione di azioni sincrone modali con la designazione ID(S) nel partprogram

10.4.2 Emissione di funzioni ausiliarie

Funzione

L'emissione delle funzioni ausiliarie avviene nell'azione sincrona direttamente al momento dell'emissione dell'azione. Il momento di emissione di tali funzioni che era stato stabilito nei dati macchina non è più valido.

Il momento dell'emissione si verifica quando la condizione è soddisfatta.

Esempio:

Attivare il refrigerante con una posizione particolare dell'asse:

```
WHEN $AA_IM[X] >=15 DO M07 POS[X]=20 FA[X]=250
```

Parole chiave consentite in azioni sincrone blocco per blocco (senza ID modale)

Le funzioni ausiliarie possono essere programmate solo con le parole chiave `WHEN` o `EVERY`.

Nota

Le seguenti funzioni ausiliarie non sono consentite nell'azione sincrona:

- M0, M1, M2, M17, M30: fine/arresto programma (M2, M17, M30 possibili nei cicli tecnologici)
 - M70: funzione mandrino
 - M6 o funzioni M impostate tramite il dato macchina per il cambio utensile
 - M40, M41, M42, M43, M44, M45: cambio gamma
-

Esempio

```
WHEN $AA_IW[Q1] >5 DO M172 H510
```

;Emettere le funzioni ausiliarie M172 e H510 al PLC
;quando il valore reale dell'asse Q1 supera i 5 mm.

10.4.3 Impostazione del blocco lettura (RDISABLE)

Funzione

Se è stata attivata l'istruzione RDISABLE viene arrestata l'ulteriore elaborazione di blocchi nel programma principale. Le azioni sincrone di movimento programmate vengono proseguite; continua anche la elaborazione dei blocchi successivi.

Nel funzionamento continuo, all'inizio di un blocco con RDISABLE nelle azioni sincrone viene sempre attivato un arresto preciso, indipendentemente dal fatto che RDISABLE sia attiva o meno.

Esempio

Si intende avviare un programma nel clock di interpolazione in funzione di ingressi esterni.

... WHENEVER \$A_INA[2]<7000 DO RDISABLE N10 G1 X10 N20 G1 X10 Y20 ...	;arrestare il proseguimento del programma se ;all'ingresso 2 ;la tensione scende al di sotto di 7V (1000= 1V). ;se la condizione risulta soddisfatta, ;il blocco lettura diventa attivo alla fine di N10

10.4.4 Disattivazione stop preelaborazione (STOPREOF)

Funzione

STOPREOF disattiva lo stop preelaborazione per il blocco successivo non appena la condizione è soddisfatta, nel caso di uno stop preelaborazione esplicitamente programmato con STOPRE o attivato implicitamente tramite un'azione sincrona attiva.

Nota

STOPREOF deve essere programmato con la parola chiave `WHEN` e blocco-blocco (senza numero ID).

Esempio

Diramazione rapida del programma alla fine del blocco.

<code>WHEN \$AC_DTEB<5 DO STOPREOF</code>	<code>;Disattivare lo stop preelaborazione se la distanza</code> <code>;rispetto alla fine blocco è inferiore a 5 mm.</code>
<code>G01 X100</code>	<code>;Lo stop preelaborazione viene disattivato</code> <code>;dopo l'esecuzione dell'interpolazione lineare.</code>
<code>IF \$A_INA[7]>500 GOTOF MARK1=X100</code>	<code>;Saltare all'etichetta 1 se all'ingresso 7 la tensione</code> <code>;supera 5V.</code>

10.4.5 Cancellazione percorso residuo (DELDTG)

Funzione

In relazione ad una condizione è anche possibile attivare la cancellazione del percorso residuo per il percorso e per gli assi indicati.

Sono a disposizione:

- la cancellazione del percorso residuo rapida, già preparata
- Cancellazione del percorso residuo senza preparazione

La cancellazione del percorso residuo con preparazione tramite DELDTG consente una reazione molto rapida all'evento di attivazione e viene utilizzata per applicazioni con criticità di tempo, es. quando

- il tempo tra la cancellazione del percorso residuo e lo start del blocco successivo deve essere molto breve.
- quando la condizione per la cancellazione del percorso residuo verrà soddisfatta con alta probabilità.

Nota

L'identificatore dell'asse racchiuso tra parentesi di DELDTG è valido solo per **un** asse di posizionamento.

Programmazione

Cancellazione del percorso residuo per la traiettoria
DO DELDTG

oppure

cancellazione del percorso residuo assiale
DO DELDTG(asse1, asse2, ...)

Esempio di cancellazione del percorso residuo per la traiettoria

```
WHEN $A_IN[1]==1 DO DELDTG
N100 G01 X100 Y100 F1000           ;quando è settato l'ingresso viene interrotto
                                   ;il movimento
N110 G01 X...
IF $AA_DELT>50...
```

Esempio di cancellazione rapida del percorso residuo assiale

```
Interruzione di un movimento di posizionamento:
ID=1 WHEN $A_IN[1]==1 DO MOV[V]=3 FA[V]=700           ;Start asse
WHEN $A_IN[2]==1 DO DELDTG(V)           ;Cancellazione del percorso residuo, l'arresto asse avviene
                                         con MOV=0

Cancellazione del percorso residuo dipendente
dalla tensione all'ingresso:
WHEN $A_INA[5]>8000 DO DELDTG(X1)
;non appena all'ingresso 5 la tensione supera gli 8V, viene cancellato il percorso
residuo
dell'asse X1. Il movimento interpolato prosegue.
POS[X1]=100 FA[X1]=10 G1 Z100 F1000
```

Descrizione

Alla fine di un blocco di movimento, in cui è stato avviata una cancellazione del percorso residuo già preparata, viene attivato implicitamente anche lo stop preelaborazione.

Ciò significa che il funzionamento continuo o i movimenti dell'asse di posizionamento vengono interrotti o arrestati alla fine del blocco mediante la cancellazione rapida del percorso residuo.

Nota

La cancellazione del percorso residuo preparata

- non può essere impiegata quando è attiva la correzione del raggio utensile.
 - può essere programmata solo in azioni sincrone attive blocco per blocco (senza numero ID).
-

10.4.6 Definizione dei polinomi (FCTDEF)

Funzione

Con FCTDEF possono essere definiti polinomi di 3. grado nella forma $y=a_0+a_1x+a_2x^2+a_3x^3$. Questi polinomi vengono utilizzati dalla correzione utensile online FTOC e dalla funzione di valutazione SYNFACT.

Programmazione

FCTDEF (n. polinomio, LLIMIT, ULIMIT, a₀, a₁, a₂, a₃)

Parametri

N. polinomio	Numero del polinomio di 3° grado
LLIMIT	Limite inferiore del valore della funzione
ULIMIT	Limite superiore del valore della funzione
a ₀ , a ₁ , a ₂ , a ₃	Coefficienti del polinomio

Questi valori sono accessibili anche tramite variabile di sistema

\$AC_FCTL[n]	Limite inferiore del valore della funzione
\$AC_FCTUL[n]	Limite superiore del valore della funzione
\$AC_FCT0[n]	a ₀
\$AC_FCT1[n]	a ₁
\$AC_FCT2[n]	a ₂
\$AC_FCT3[n]	a ₃

Nota

Scrittura di variabili di sistema

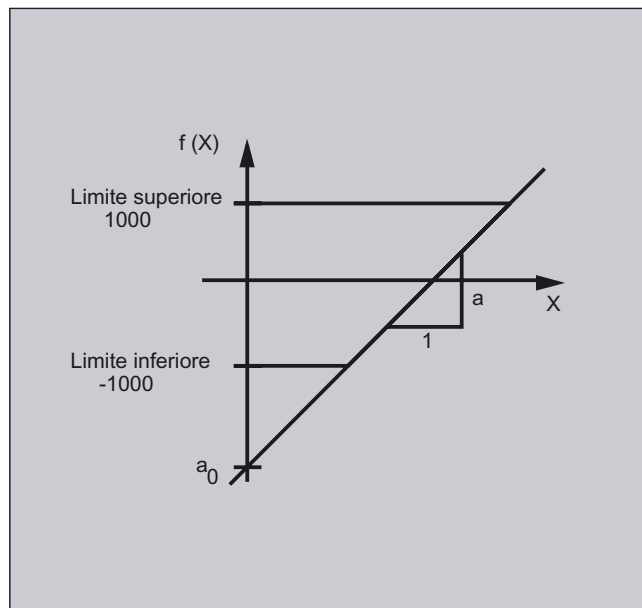
- Le variabili di sistema possono essere scritte anche da un partprogram o da un'azione sincronizzata. Scrivendole da un partprogram è necessario programmare STOPRE per garantire che la scrittura sia sincrona al blocco.
- Le variabili di sistema \$AC_FCTL[n], \$AC_FCTUL[n], \$AC_FCT0[n] bis \$AC_FCTn[n] possono essere modificate anche da azioni sincrone

I coefficienti di polinomi e i limiti dei valori di funzione scritti da azioni sincrone sono attivi immediatamente.

Esempio di polinomio per sezione di retta

La definizione del polinomio, con limite superiore 1000, limite inferiore -1000, sezione di ordinata a₀=\$AA_IM[X] e inclinazione della retta pari a 1, è la seguente:

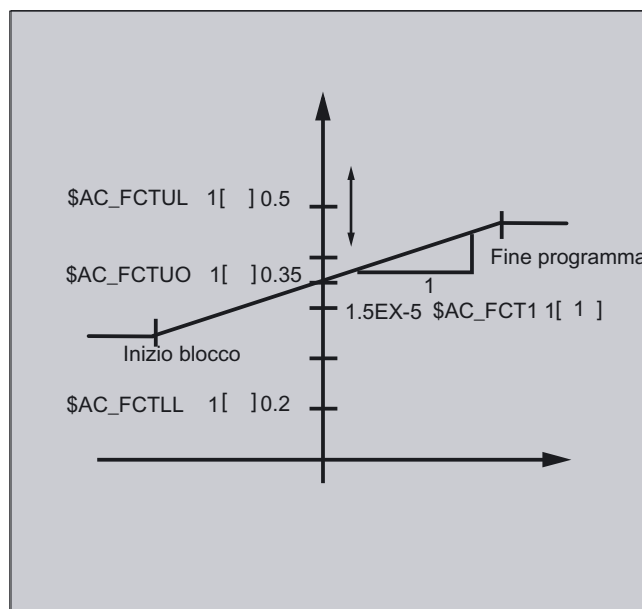
FCTDEF(1, -1000, 1000, \$AA_IM[X], 1)



Esempio di controllo della potenza del laser

Uno dei possibili impieghi della definizione di polinomi è il controllo della potenza del laser.

Controllo della potenza del laser significa:
influenzare un'uscita analogica in funzione per es. della velocità vettoriale.



$\$AC\_FCTLL\ [1] = 0.2$
 $\$AC\_FCTUL\ [1] = 0.5$

;Definizione dei coefficienti del polinomio

```

$AC_FCT0[1]=0.35
$AC_FCT1[1]=1.5EX-5
STOPRE
ID=1 DO $AC_FCTUL[1]=$A_INA[2]*0.1 +0.35 ;Modificare il limite superiore online.
ID=2 DO SYNFACT(1,$A_OUTA[1],$AC_VACTW)
;in funzione della velocità vettoriale (impostata in $AC_VACTW) viene controllata
; il comando della potenza del laser tramite l'uscita analogica 1

```

Nota

Il polinomio suddetto viene impiegato con SYNFACT.

10.4.7 Funzione sincrona (SYNFACT)

Funzione

SYNFACT calcola il valore di uscita di un polinomio di terzo grado ponderato con la variabile d'ingresso. Il risultato si trova nella variabile d'uscita e viene limitato in alto e in basso.

La funzione di valutazione viene impiegata

- nella regolazione AC (Controllo Adattativo),
- nel controllo della potenza del laser,
- per offset di posizione.

Programmazione

SYNFACT(N. di polinomio, uscita della variabile di esecuzione, ingresso variabile di esecuzione)

Parametri

Come variabili di uscita si possono selezionare le variabili che confluiscono

- con influenza additiva
- con influenza moltiplicativa
- come offset di posizione
- direttamente

nel processo di lavorazione.

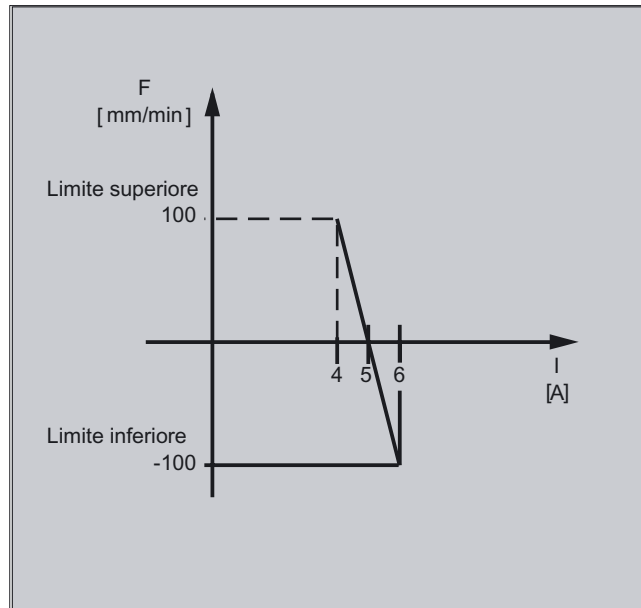
DO SYNFACT	Attivazione della funzione di elaborazione
N. polinomio	Il polinomio definito con FCTDEF (vedere "Definizione polinomi")
Uscita della variabile di esecuzione	Scrittura della variabile di esecuzione
Ingresso della variabile di esecuzione	Lettura della variabile di esecuzione

Esempio di regolazione AC (additiva)

Influenza additiva sull'avanzamento programmato

Si desidera regolare additivamente un avanzamento programmato mediante la corrente dell'asse X (asse di incremento):

l'avanzamento dovrà variare di +/- 100 mm/min, tenendo conto che la corrente di 5A sul punto di lavorazione varia di +/- 1A.



1. Definizione del polinomio

Definizione dei coefficienti

$$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$a_1 = -100\text{mm}/1 \text{ min A}$$

$$a_0 = -(-100) \cdot 5 = 500$$

$$a_2 = a_3 = 0 \text{ (nessun elemento quadratico o cubico)}$$

$$\text{Limite superiore} = 100$$

$$\text{Limite inferiore} = -100$$

Ne consegue:

$$\text{FCTDEF}(1, -100, 100, 500, -100, 0, 0)$$

2. Attivazione della regolazione AC

ID=1 DO SYNFACT(1, \$AC_VC, \$AA_LOAD[x])

;Leggere il carico attuale dell'asse (% max. corrente azionamento) con \$AA_LOAD[x] ,
;con il polinomio definito in precedenza calcolare la correzione dell'avanzamento vettoriale.

Esempio di regolazione AC (moltiplicativa)

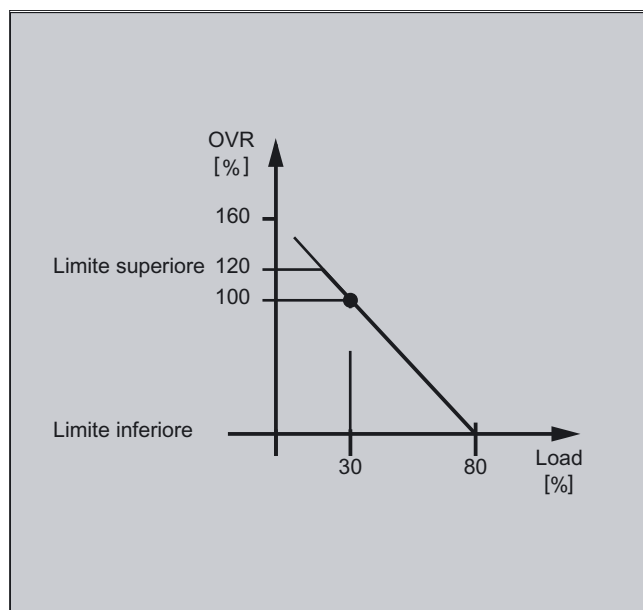
Influenza a carattere moltiplicativo dell'avanzamento programmato

Si desidera modificare in maniera moltiplicativa l'avanzamento programmato, senza però che l'avanzamento, a seconda del carico dell'azionamento, superi determinati limiti:

- Con carico dell'azionamento di 80% l'avanzamento deve arrestarsi: override = 0.
- Con carico dell'azionamento pari a 30% si può procedere con l'avanzamento programmato:
Override = 100%.

La velocità dell'avanzamento può essere superata al massimo del 20%:

Max. Override = 120%.

**1. Definizione del polinomio**

Definizione dei coefficienti

$$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$a_1 = -100\% / (80-30)\% = -2$$

$$a_0 = 100 + (2 \cdot 30) = 160$$

$$a_2 = a_3 = 0 \text{ (nessun elemento quadratico o cubico)}$$

$$\text{Limite superiore} = 120$$

$$\text{Limite inferiore} = 0$$

Ne consegue:

$$\text{FCTDEF}(2, 0, 120, 160, -2, 0, 0)$$

2. Attivazione della regolazione AC

```
ID=1 DO SYNFACT(2, $AC_OVR, $AA_LOAD[x])
```

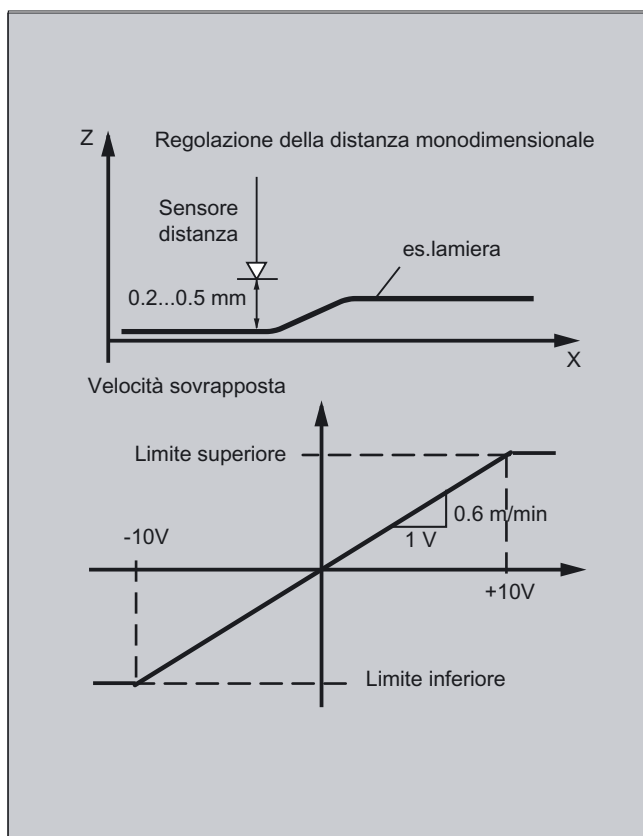
;Leggere il carico attuale dell'asse (% max. corrente azionamento) con \$AA_LOAD[x] ,
;con il polinomio definito in precedenza calcolare l'override dell'avanzamento

10.4.8 Regolazione della distanza con correzione limitata \$AA_OFF_MODE

Funzione

Calcolo integrativo dei valori di distanza con verifica del campo limite

\$AA_OFF_MODE = 1



Attenzione

L'amplificazione del circuito di regolazione sovrapposto dipende dalla impostazione del clock di interpolazione.

Rimedio: Leggere il DM del clock IPO e considerarlo nel calcolo.

Nota

Limitazione della velocità dell'interpolatore sovrapposto tramite il DM 32020: JOG_VELO con clock IPO di 12 ms. Formula per la velocità:

$$\frac{0.120mm}{0.6ms} / mV = 0.6 \frac{m}{min} / V$$

Esempio**Sottoprogramma: regolazione della distanza ON**

%_N_AON_SPF	;Sottoprogramma per la regolazione della ;distanza ON
PROC AON	
\$AA_OFF_LIMIT[Z]=1	;Definire il valore limite
FCTDEF(1, -10, +10, 0, 0.6, 0.12)	;Definizione del polinomio
ID=1 DO SYNFACT(1,\$AA_OFF[Z],\$A_INA[3])	;Regolazione della distanza attiva
ID=2 WHENEVER \$AA_OFF_LIMIT[Z]<>0	;Bloccare l'asse X se il valore limite viene
DO \$AA_OVR[X] = 0	;superato
RET	
ENDPROC	

Sottoprogramma: regolazione della distanza OFF

%_N_AOFF_SPF	
PROC AOFF	;Sottoprogramma per la regolazione della ;distanza. OFF
CANCEL(1)	;Cancellare l'azione sincrona regolaz. distanza
CANCEL(2)	;Cancellare la verifica del campo limite
RET	
ENDPROC	

Programma principale

%_N_MAIN_MPF	
AON	;Regolazione della distanza ON
...	
G1 X100 F1000	
AOFF	;Regolazione della distanza OFF
M30	

Offset di posizione nel sistema di coordinate di base

Con la variabile di sistema \$AA_OFF[asse] è possibile sovrapporre un movimento ad ogni asse del canale. Essa è attiva come offset di posizione nel sistema di coordinate di base.

L'offset di posizione così programmato viene sovrapposto immediatamente all'asse in questione, indipendentemente dal fatto che il movimento dell'asse sia programmato o meno.

Limitazione dell'uscita della variabile di esecuzione:

È possibile limitare il valore da correggere in modo assoluto (uscita della variabile di esecuzione) al valore registrato nel dato setting

DS 43350: AA_OFF_LIMIT.

Con il dato macchina DM 36750: AA_OFF_MODE viene stabilito il tipo di sovrapposizione della distanza:

0: valutazione proporzionale

1: valutazione integrale

Con la variabile di sistema \$AA_OFF_LIMIT[asse] è possibile interrogare, in funzione delle direzioni, se il valore di correzione si trova ancora all'interno dei limiti. Detta variabile di sistema può essere interrogata da azioni sincrone e può arrestare l'asse o settare un allarme se viene raggiunto un valore limite.

0: Valore di correzione al di fuori del campo limite.

1: Limite del valore di correzione raggiunto in direzione positiva

-1: Limite del valore di correzione raggiunto in direzione negativa

10.4.9 Correzione utensile on-line (FTOC)

Funzione

FTOC consente movimenti sovrapposti per uno o più assi dopo un polinomio programmato con FCTDEF in funzione di un valore di riferimento, che per es. può essere il valore reale di un asse.

Il coefficiente a_0 della definizione della funzione FCTDEF() viene analizzato per FTOC. I limiti superiore e inferiore dipendono da a_0 .

In questo modo è possibile anche programmare correzioni utensili on-line modali oppure regolazioni differenziali come azioni sincrone.

Viene impiegata per l'esecuzione del pezzo e per la diamantatura della mola in uno stesso canale o in canali diversi (canale di lavorazione e canale di diamantatura).

Le condizioni generali e le definizioni per la diamantatura di mole valgono per FTOC in modo analogo alla correzione utensile on-line con PUTFTOCF. Per ulteriori informazioni consultare il capitolo "Correzioni utensile".

Programmazione

FTOC(n. polinomio, HVL, lungh.1_2_3 oppure raggio4, nr. canale, n. mandrino)

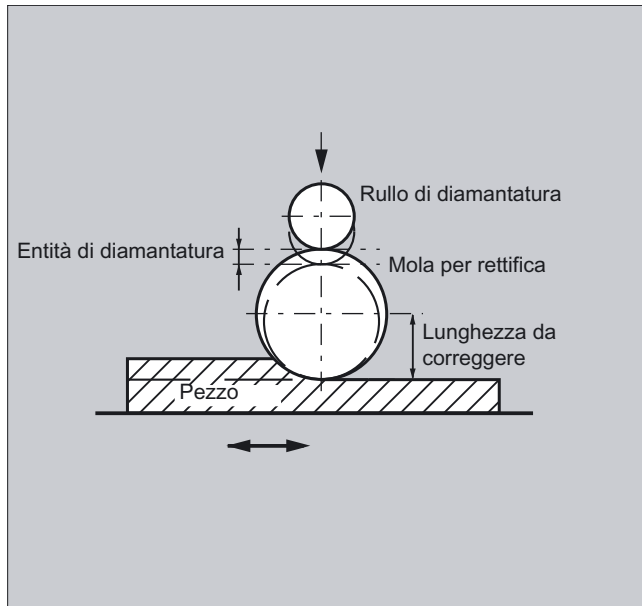
Parametri

DO FTOC	Esecuzione della correzione utensile online
N. polinomio	Polinomio definito con FCTDEF, vedere sezione "Definizione dei polinomi" in questo capitolo.
EV	Variabile di esecuzione, per la quale deve essere calcolato un valore di funzione tramite il polinomio specificato.
Lunghezza1_2_3	Correzione lunghezza (\$TC_DP1 ... 3) opp. correzione raggio, a cui
Raggio4	viene sommato il valore della funzione calcolato.

Canale	Canale di destinazione, nel quale deve essere attiva la correzione. Per la correzione nel canale attivo, l'indicazione può essere omessa. Nel canale di destinazione deve essere attivo FTOCON.
Mandrino	Impostare solo se non deve essere corretto il mandrino attivo.

Esempio

Nell'esempio si intende correggere la lunghezza della mola attiva che sta lavorando.



```

%_N_ABRICHT_MPF
FCTDEF(1,-1000,1000,-$AA_IW[V],1)
ID=1 DO FTOC(1,$AA_IW[V],3,1)

WAITM(1,1,2)

G1 V-0.05 F0.01 G91
G1 V-0.05 F0.02
...
CANCEL(1)
...

```

;Definizione della funzione.

;Selezionare la correzione utensile online:

;il valore reale dell'asse V costituisce il

;valore d'ingresso del polinomio 1; il risultato

;viene sommato nel canale 1 quale valore

;di correzione alla lunghezza 3 della mola attiva.

;Eeguire la sincronizzazione con il canale di

;lavorazione

;Movimento di incremento per la diamantatura

;Disattivare la correzione on-line

10.4.10 Correzione online della lunghezza utensile (\$AA_TOFF[Direzione_utensile])

Funzione

Con la variabile \$AA_TOFF[] si può eseguire una correzione tridimensionale in tempo reale sulla base delle tre dimensioni effettive dell'utensile.

Come indice si utilizzano i tre identificativi degli assi geometrici. In questo modo il numero delle direzioni di correzione attive viene definito attraverso gli assi geometrici abilitati nello stesso momento.

Tutte le correzioni possono essere abilitate contemporaneamente.

Programmazione

```
N.. TRAORI
N.. TOFFON(X, 25)
N.. WHEN TRUE DO $AA_TOFF[X]
N.. TOFFON(Y, 25)
N.. WHEN TRUE DO $AA_TOFF[Y]
N.. TOFFON(Z, 25)
N.. WHEN TRUE DO $AA_TOFF[Z]
```

Parametro

TOFFON	Tool Offset ON (attivare la correzione online della lunghezza utensile) Quando si attiva la funzione, per la relativa direzione di correzione si può indicare un valore di offset che verrà subito eseguito.
TOFFOF	Tool Offset ON (disattivare la correzione online della lunghezza utensile) I valori di correzione vengono resettati e viene attivato uno stop preelaborazione.
X, Y, Z,	Direzione di correzione per il valore di offset indicato in TOFFON
\$AA_TOFF[X]=valore	Sovrapposizione in direzione X
\$AA_TOFF[Y]=valore	Sovrapposizione in direzione Y
\$AA_TOFF[Z]=valore	Sovrapposizione in direzione Z

Esempio di attivazione della correzione lunghezza utensile

N10 TRAORI(1)	; trasformazione ON
N20 TOFFON(Z)	;Attivazione della correzione online della ;lunghezza utensile per la direzione utensile Z
N30 WHEN TRUE DO \$AA_TOFF[Z] = 10	;per la direzione utensile Z viene interpolata una
G4 F5	;correzione della lunghezza utensile di 10
N40 TOFFON(X)	;Attivazione della correzione online della ;lunghezza utensile per la direzione utensile X
N50 ID=1 DO \$AA_TOFF[X] = \$AA_IW[X2]	;per la direzione utensile X viene eseguita una
G4 F5	;correzione in base alla posizione dell' ;asse X2

...	
N100 XOFFSET = \$AA_TOFF_VAL[X]	;assegnare la correzione attuale in direzione X
N120 TOFFON(X, -XOFFSET)	;per la direzione utensile X la
G4 F5	;correzione della lunghezza utensile viene riportata
	;a 0

Esempio di disattivazione della correzione lunghezza utensile

N10 TRAORI(1)	; trasformazione ON
N20 TOFFON(X)	;attivazione della direzione utensile Z
N30 WHEN TRUE DO \$AA_TOFF[X] = 10	;per la direzione utensile X viene interpolata una
G4 F5	;correzione della lunghezza utensile di 10
...	
N80 TOFFOF(X)	;l'offset di posizione della direzione utensile; in X
	;viene cancellato: ...\$AA_TOFF[X] = 0
	;Gli assi non vengono mossi
	;alla posizione attuale nel sistema SCP viene
	;aggiunto l'offset di posizione in base
	;all'orientamento attuale

10.4.11 Movimenti di posizionamento

Funzione

Gli assi possono essere posizionati da azioni sincrone anche in maniera del tutto asincrona rispetto al partprogram. La programmazione di assi di posizionamento da azioni sincrone è consigliata per sequenze cicliche dipendenti in maniera determinante da eventi. Gli assi programmati da azioni sincrone sono chiamati **assi di comando**.

Programmazione

Bibliografia:

/PG/ Manuale di programmazione, Concetti fondamentali, capitolo "Dati del percorso"
/FBSY/ Descrizione delle funzioni, Azioni sincrone; "Avvio di assi di comando"

Parametri

Il sistema di misura per ordini di posizionamento in azioni sincrone, viene definito con G70/G71/G700/G710 .

Mediante la programmazione di funzioni G nell'azione sincrona è possibile definire la valutazione inch/metrica per l'azione sincrona indipendentemente dal contesto del partprogram.

10.4.12 Posizionamento dell'asse (POS)

Funzione

Contrariamente alla programmazione da partprogram, il movimento dell'asse di posizionamento non ha influenza sull'esecuzione del partprogram stesso.

Programmazione

POS [asse] =valore

Parametri

DO POS	Start/posizionamento asse di comando
Asse	Nome dell'asse che deve essere mosso
Valore	Indicazione del valore da eseguire (in base al modo del movimento)

Esempio

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100
;movimento incrementale dell'asse U, indipendentemente dal modo, di 100 (inch/mm)
oppure
;alla posizione 100 (inch/mm) dal punto di riferimento del controllo.
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=$AA_MW[V] - $AA_IM[W] +13.5
;movimento dell'asse U nella misura del percorso calcolato dalla variabile di
esecuzione.
```

Esempio

Il programma influenza la traiettoria di posizionamento del relativo asse (nessuna funzione G nella parte azione dell'azione sincrona)

```
N100 R1=0
N110 G0 X0 Z0
N120 WAITP(X)
N130 ID=1 WHENEVER $R==1 DO POS[X]=10
N140 R1=1
N150 G71 Z10 F10 ;Z=10 mm X=10 mm
N160 G70 Z10 F10 ;Z=254 mm X=254 mm
N170 G71 Z10 F10 ;Z=10 mm X=10 mm
N180 M30
```

G71 nella parte azione dell'azione sincrona determina in modo univoco (metrico) il percorso del relativo asse di posizionamento indipendentemente dal contesto del programma.

```

N100 R1=0
N110 G0 X0 Z0
N120 WAITP(X)
N130 ID=1 WHENEVER $R==1 DO G71 POS[X]=10
N140 R1=1
N150 G71 Z10 F10 ;Z=10 mm X=10 mm
N160 G70 Z10 F10 ;Z=254 mm X=10 mm (X posiziona sempre
;a 10 mm)
N170 G71 Z10 F10 ;Z=10 mm X=10 mm
N180 M30

```

Se si desidera che il movimento dell'asse non venga attivato con l'inizio del blocco è possibile mantenere, da un'azione sincrone, l'override dell'asse a 0 fino al punto di inizio desiderato.

```

WHENEVER      $A_IN[1]==0 DO $AA_OVR[W]=0
               G01 X10 Y25 F750 POS[W]=1500
               FA=1000
               ;l'asse di posizionamento viene arrestato finché l'ingresso digitale 1 = 0

```

10.4.13 Posizione nell'intervallo di riferimento predefinito (POSRANGE)

Funzione

Con la funzione POSRANGE() è possibile determinare se la posizione attuale di riferimento interpolata di un asse si trovi in una finestra attinente alla posizione di riferimento predefinita. I dati della posizione si possono riferire a sistemi di coordinate impostabili.

Nella richiesta della posizione reale di un asse modulo si tiene conto della correzione del modulo.

Nota

La funzione può essere richiamata solo dall'azione sincrone. Se la chiamata avviene dal partprogram, viene emesso l'allarme 14091 %1 blocco %2 Funzione non consentita, indice: %3 richiamato con l'indice 5.

Programmazione

```

BOOL POSRANGE(asse, Refpos, Winlimit, [Coord])

```

Parametri:

BOOL POSRANGE	La posizione attuale dell'asse di comando è nella finestra della posizione di riferimento predefinita.
AXIS <asse>	Identificatore dell'asse macchina, asse canale o asse di geometria
REAL Refpos	Posizione di riferimento nel sistema di coordinate Coord
REAL Winlimit	Valore che esprime il limite per la finestra di posizione
INT Coord	Il sistema di coordinate macchina è attivo opzionalmente. Sono possibili: 0 per MKS (SCM, sistema di coordinate macchina) 1 per BKS (SCB, sistema di coordinate di base) 2 per ENS (sistema punto di origine impostabile) 3 per WKS (SCP, sistema di coordinate pezzo)

Valore della funzione

Posizione di riferimento attuale a seconda del dato di posizione nel sistema di coordinate predefinito

Valore della funzione:	se Refpos(Coord)
TRUE	- abs(Winlimit) ≤ Actpos(Coord) ≤ Refpos(Coord) + abs(Winlimit)
	altrimenti
Valore della funzione:	
FALSE	

10.4.14 Start/stop asse (MOV)

Funzione

Con MOV[asse]=valore è possibile attivare un asse di comando senza necessità di impostare una posizione finale. L'asse muove nella direzione programmata fino a quando un nuovo comando di movimento o di posizionamento non implica un nuovo movimento oppure fino a quando l'asse non viene arrestato con un comando di stop.

Programmazione

MOV[asse] = valore

Parametri

DO MOV	Avvio del movimento dell'asse di comando
Asse	Nome dell'asse che deve essere attivato
Valore	Comando di avvio/arresto del movimento Il segno stabilisce la direzione di movimento Tipo di dati: INTEGER.
Valore >0 (generalmente +1)	Direzione positiva
Valore <0 (generalmente -1)	Direzione negativa
Valore ==0	Arrestare il movimento asse

Nota

Se un asse di divisione viene arrestato con `MOV [asse] = 0`, l'asse viene bloccato alla successiva posizione di divisione.

Esempio

```
| ... DO MOV [U] = 0 ;L'asse U viene arrestato
```

10.4.15 Scambio assi (RELEASE, GET)**Funzione**

Per un cambio utensile possono venire richiesti i relativi assi di comando come azione sincrona con `GET(asse)`. Il tipo di asse assegnato a questo canale e la relativa autorizzazione di interpolazione contemporanea possono essere oggetto di interrogazione tramite la variabile di sistema `$AA_AXCHANGE_TYP`. A seconda dello stato effettivo e del canale che possiede l'attuale autorizzazione di interpolazione di questo asse, sono possibili varie sequenze.

Se il cambio utensile è terminato, questo asse di comando può venire abilitato, quale azione sincrona con `RELEASE(asse)`, per il canale.

Costruttore della macchina

L'asse interessato deve essere assegnato al canale tramite dati macchina. Fare attenzione alle indicazioni del costruttore della macchina.

Programmazione

`GET (Asse [, asse{ , ... }])` richiesta asse

`RELEASE (asse [, asse{ , ... }])` abilitazione asse

Parametro

```
| DO RELEASE
| DO GET
| Asse
```

Abilitazione asse come asse neutro

Prelievo asse per scambio assi

Nome dell'asse che deve essere attivato

Esempio di esecuzione del programma per uno scambio assi di due canali

L'asse Z è conosciuto nel 1° canale e nel 2° canale.

Esecuzione del programma nel 1° canale:

	WHEN TRUE DO RELEASE(Z)	;asse Z diventa asse neutro
	WHENEVER(\$AA_TYP[Z]==1) DO	;blocco lettura finché asse Z è asse programma
	RDISABLE	
N110	G4 F0.1	
	WHEN TRUE DO GET(Z)	;asse Z diventa di nuovo asse programma NC
	WHENEVER(\$AA_TYP[Z]<>1) DO	;blocco lettura finché asse Z è asse programma
	RDISABLE	
N120	G4 F0.1	
	WHEN TRUE DO RELEASE(Z)	;asse Z diventa asse neutro
	WHENEVER(\$AA_TYP[Z]==1) DO	;blocco lettura finché asse Z è asse programma
	RDISABLE	
N130	G4 F0.1	
N140	START(2)	;avvio 2° canale

Esecuzione del programma nel 2° canale:

	WHEN TRUE DO GET(Z)	;prelievo asse Z nel 2° canale
	WHENEVER(\$AA_TYP[Z]==0) DO	;blocco lettura finché l'asse Z è in un altro
	RDISABLE	;canale
N210	G4 F0.1	
	WHEN TRUE DO GET(Z)	;asse Z diventa asse programma NC
	WHENEVER(\$AA_TYP[Z]<>1) DO	;blocco lettura finché asse Z è asse programma
	RDISABLE	
N220	G4 F0.1	
	WHEN TRUE DO RELEASE(Z)	;asse Z nel 2° canale asse neutro
	WHENEVER(\$AA_TYP[Z]==1) DO	;blocco lettura finché asse Z è asse programma
	RDISABLE	
N230	G4 F0.1	
N250	WAITM(10, 1, 2)	;sincronizzazione con canale 1

Ulteriore esecuzione del programma nel 1° canale:

N150	WAIM(10, 1, 2)	;sincronizzazione con canale 2
	WHEN TRUE DO GET(Z)	;prelievo asse Z in questo canale
	WHENEVER(\$AA_TYP[Z]==0) DO	;blocco lettura finché l'asse Z è in un altro
	RDISABLE	;canale
N160	G4 F0.1	
N199	WAITE(2)	;attesa della fine del programma nel canale 2
N999	M30	

Esempio di scambio assi nel ciclo tecnologico

L'asse U (\$MA_AUTO_GET_TYPE=2) è conosciuto nel 1° canale e nel 2° canale e attualmente il canale 1 ha il diritto di interpolazione. Nel canale 2 viene avviato il seguente ciclo tecnologico:

GET (U)	;prelievo asse U nel canale
POS [U] =100	;l'asse U deve essere spostato sulla posizione 100

La riga del movimento dell'asse di comando POS[U] viene eseguita solo se è stato prelevato l'asse U nel canale 2.

Sequenza

L'asse richiesto al momento dell'attivazione dell'azione GET (asse) può essere letto in rapporto al tipo di asse per uno scambio di asse con la variabile di sistema (\$AA_AXCHANGE_TYP[<asse>]):

- 0: asse assegnato al programma NC
- 1: asse assegnato al PLC o attivo come asse di comando o asse di pendolamento
- 2: un altro canale dispone del diritto di interpolazione
- 3: l'asse è un asse neutro
- 4: l'asse neutro è controllato dal PLC
- 5: un altro canale dispone del diritto di interpolazione, l'asse è richiesto per il programma NC
- 6: un altro canale dispone del diritto di interpolazione, l'asse è richiesto come asse neutro
- 7: asse del PLC o attivo come asse di comando o asse di pendolamento, l'asse è richiesto per il programma NC
- 8: asse del PLC o attivo come asse di comando o asse di pendolamento, l'asse è richiesto come asse neutro

Condizioni marginali

L'asse interessato deve essere assegnato al canale tramite dati macchina.

Un asse controllato esclusivamente dal PLC non può essere assegnato al programma NC.

Bibliografia:

/FB2/ Manuale delle funzioni di ampliamento; assi di posizionamento (P2)

Richiesta dell'asse da un altro canale con l'azione GET

se al momento dell'attivazione dell'azione GET un **altro canale ha l'autorizzazione di scrittura** (autorizzazione di interpolazione) per l'asse (\$AA_AXCHANGE_TYP[<asse>] == 2), l'asse viene richiesto da questo canale tramite scambio assi (\$AA_AXCHANGE_TYP[<asse>]==6) e assegnato al canale richiesto non appena possibile.

L'asse assume quindi lo stato di asse neutro (\$AA_AXCHANGE_TYP[<asse>]==3).

Non viene eseguita una riorganizzazione nel canale richiesto.

Assegnazione come programma NC asse con riorganizzazione:

Se già al momento dell'attivazione dell'azione GET l'asse è già stato richiesto come asse neutro (\$AA_AXCHANGE_TYP[<asse>]==6), l'asse viene richiesto per il programma NC (\$AA_AXCHANGE_TYP[<asse>]==5) e non appena possibile assegnato al programma NC del canale (\$AA_AXCHANGE_TYP[<asse>]==0).

Asse già assegnato al canale richiesto

Assegnazione come programma NC asse con riorganizzazione:

Se al momento dell'attivazione l'asse richiesto è già assegnato al canale richiesto, e nello stato asse neutro – non controllato dal PLC – (\$AA_AXCHANGE_TYP[<asse>]==3), esso viene assegnato al programma NC (\$AA_AXCHANGE_TYP[<asse>]==0).

L'asse nello stato asse neutro è controllato dal PLC

Se l'asse nello stato asse neutro è controllato dal PLC (\$AA_AXCHANGE_TYP[<asse>]==4), esso viene richiesto come asse neutro (\$AA_AXCHANGE_TYP[<asse>]==8), in base al bit 0 nel dato macchina DM 10722: AXCHANGE_MASK l'asse viene disabilitato per uno scambio assi automatico tra i canali (Bit 0 == 0). Questo corrisponde a (\$AA_AXCHANGE_STAT[<asse>]==1).

L'asse è attivo come asse di comando neutro o asse di pendolamento o assegnato al PLC

Se l'asse è attivo come asse di comando o asse di pendolamento oppure assegnato per il movimento al PLC, asse PLC == asse di posizionamento simultaneo, (\$AA_AXCHANGE_TYP[<asse>]==1), l'asse viene richiesto come asse neutro (\$AA_AXCHANGE_TYP[<asse>]==8), in base al bit 0 nel dato macchina DM 10722: AXCHANGE_MASK l'asse viene disabilitato per uno scambio assi automatico tra i canali (Bit 0 == 0). Questo corrisponde a (\$AA_AXCHANGE_STAT[<asse>]==1).

Una nuova azione GET richiede quindi l'asse per il programma NC (\$AA_AXCHANGE_TYP[<asse>] diventa == 7).

L'asse è già assegnato al programma NC

Se l'asse è già assegnato al programma NC del canale (\$AA_AXCHANGE_TYP[<asse>]==0) o se è richiesta questa assegnazione, p. es. scambio assi attivato dal programma NC (\$AA_AXCHANGE_TYP[<asse>]==5 o \$AA_AXCHANGE_TYP[<asse>]==7), non si verifica nessuna variazione di stato.

10.4.16 Avanzamento assiale (FA)

Funzione

L'avanzamento assiale per gli assi di comando ha effetto modale.

Programmazione

FA[asse]= avanzamento

Esempio

```
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=990  
; impostazione fissa del valore di avanzamento  
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=$AA_VACTM[W]+100  
;formare il valore di avanzamento dalla variabile di esecuzione
```

10.4.17 Finecorsa SW

Funzione

La limitazione del campo di lavoro programmata con G25/G26 viene tenuta in considerazione per gli assi di comando in relazione al dato setting \$SA_WORKAREA_PLUS_ENABLE.

L'attivazione e la disattivazione della limitazione del campo di lavoro con le funzioni G WALIMON/WALIMOF in un partprogram non ha influenza sugli assi comando.

10.4.18 Coordinamento assi

Funzione

Generalmente un asse viene mosso da un partprogram oppure da un'azione sincrona come asse di posizionamento.

Se invece uno stesso asse deve essere mosso alternativamente dal partprogram come asse di contornitura o di posizionamento ed anche da azioni sincrone, avviene una trasmissione coordinata tra i due tipi di movimento.

Se l'asse di comando viene poi mosso da un partprogram si rende necessaria una riorganizzazione della preelaborazione. Ciò causa a sua volta un'interruzione dell'esecuzione del partprogram, similmente ad uno stop preelaborazione.

Esempio di movimento dell'asse X da partprogram e da azioni sincrone

<pre>N10 G01 X100 Y200 F1000 ... N20 ID=1 WHEN \$A_IN[1]==1 DO POS[X]=150 FA[X]=200 ... CANCEL(1) ... N100 G01 X240 Y200 F1000</pre>	<pre>;Asse X programmato nel partprogram ;Avviare il posizionamento dall'azione sincrone quando ;viene attivato l'ingresso digitale ;Disattivare l'azione sincrone</pre>
<pre>;X diventa asse vettoriale; prima del movimento viene attivato un tempo di attesa a causa del passaggio di controllo dell'asse, ;nel caso sia stato attivato l'ingresso digitale 1 e se X è stato posizionato con azioni sincrone.</pre>	

Esempio di modifica del comando di movimento per lo stesso asse

<pre>ID=1 EVERY \$A_IN[1]>=1 DO POS[V]=100 FA[V]=560 ID=2 EVERY \$A_IN[2]>=1 DO POS[V]=\$AA_IM[V] FA[V]=790</pre>	<pre>;Avviare il posizionamento dall'azione sincrone quando l'ingresso digitale >= 1 ;L'asse muove quando viene settato il 2. ingresso, cioè la posizione finale e ;l'avanzamento per l'asse V vengono inseguite da due azioni sincrone ;contemporaneamente attive con movimento in corso.</pre>

10.4.19 Impostazione del valore reale (PRESETON)

Funzione

L'esecuzione di PRESETON (asse, valore) non ha effetto sulla posizione attuale dell'asse, ma le assegna un nuovo valore.

PRESETON da azioni sincrone è possibile:

- per assi di assi rotanti con funzione modulo che sono stati attivati dal partprogram
- per tutti gli assi di comando che sono stati attivati dall'azione sincrone

Programmazione

```
DO PRESETON (asse, valore)
```

Parametri

DO PRESETON	Impostazione del valore reale in azioni sincrone
Asse	Asse il cui punto di riferimento del controllo deve essere modificato
Valore	Valore del quale viene modificato il punto di riferimento del controllo

Limitazioni per assi

PRESETON non è possibile per gli assi che fanno parte di una trasformazione.

Lo stesso asse può essere mosso da un partprogram o da un'azione sincrona solo in momenti diversi, il che significa che, nella programmazione di un asse da un partprogram, possono verificarsi tempi di attesa, dipendenti da una precedente programmazione dello stesso asse in un'azione sincrona.

Se lo stesso asse viene impiegato alternativamente, avviene un trasferimento coordinato tra i due movimenti dell'asse. A questo scopo viene interrotta l'esecuzione del partprogram.

Esempio

Traslazione del punto di riferimento del controllo di un asse

```
WHEN $AA_IM[a] >= 89.5 DO PRESETON(a4,10.5)
;Spostare il punto di riferimento del controllo dell'asse a di 10.5 unità di
;lunghezza (inch opp. mm) in
```

10.4.20 Movimenti del mandrino

Funzione

I mandrini possono essere posizionati da azioni sincrone del tutto indipendentemente dal partprogram. Questo tipo di programmazione è adatto per sequenze o processi ciclici che dipendono fortemente da eventi.

Se, a causa di due azioni sincrone attive contemporaneamente, per un mandrino vengono emessi comandi contraddittori, vale sempre l'ultimo comando in ordine in tempo.

Esempio di attivazione/arresto/posizionamento mandrino

ID=1 EVERY \$A_IN[1]==1 DO M3 S1000	;Impostare il senso di rotazione e il numero di
	;giri
ID=2 EVERY \$A_IN[2]==1 DO SPOS=270	;Posizionamento mandrino

Esempio di impostazione del senso di rotazione e del numero di giri/posizionamento del mandrino

ID=1 EVERY \$A_IN[1]==1 DO M3 S300	;Impostare il senso di rotazione e il numero di giri
ID=2 EVERY \$A_IN[2]==1 DO M4 S500	;Impostare il nuovo senso di rotazione e il nuovo numero di giri
ID=3 EVERY \$A_IN[3]==1 DO S1000	;Impostare il nuovo numero di giri
ID=4 EVERY (\$A_IN[4]==1) AND (\$A_IN[1]==0)	;Posizionamento mandrino
DO SPOS=0	

10.4.21 Trascinamento (TRAILON, TRAILOF)

Funzione

Al momento dell'attivazione dell'accoppiamento dall'azione sincrona può accadere che l'asse master sia in movimento. In questo caso la velocità dell'asse a seguire viene aumentata fino a raggiungere la velocità di riferimento. La posizione dell'asse master al momento della sincronizzazione delle diverse velocità viene presa come punto di partenza del trascinamento. La funzionalità del trascinamento viene descritta nel capitolo "Comportamento del movimento sul profilo".

Programmazione

Attivare il trascinamento

```
DO TRAILON(asse a seguire, asse master,  
fattore di accoppiamento)
```

Disattivare il trascinamento

```
DO TRAILOF(asse a seguire, asse master, asse  
master 2)
```

Parametri

Attivazione del trascinamento asincrono:

```
... DO TRAILON(FA, LA, Kf)
```

con:

FA: Asse slave

LA: Asse master

Kf: fattore di accoppiamento

Disattivazione del trascinamento asincrono:

```
... DO TRAILOF(FA, LA, LA2)
```

con:

FA: Asse slave

LA: Asse master, opzionale

LA2: asse master 2, opzionale

```
... DO TRAILOF(FA)
```

Tutti gli accoppiamenti all'asse slave vengono disattivati.

Esempio

\$A_IN[1]==0 DO TRAILON(Y,V,1)	;Attivazione del 1° gruppo di trascinamento quando ;l'ingresso digitale è 1
\$A_IN[2]==0 DO TRAILON(Z,W,-1)	;Attivazione del 2. gruppo di trascinamento
G0 Z10	;Posizionamento degli assi Z e W in direzione opposta alla ;direzione assiale
G0 Y20	;Posizionamento degli assi Y e V nella stessa ;direzione assiale
...	
G1 Y22 V25	;Sovrapposizione di un movimento dipendente ed uno ;indipendente dell'asse trascinato "V"
...	
TRAILOF(Y,V)	;Disattivazione del 1. gruppo di trascinamento
TRAILOF(Z,W)	;Disattivazione del 2. gruppo di trascinamento

Esempio per evitare il conflitto con TRAILOF

Per poter consentire nuovamente la gestione di un asse accoppiato come asse canale, deve essere prima richiamata la funzione `TRAILOF`. È necessario assicurarsi che `TRAILOF` sia stata eseguita prima che il canale

richiami il relativo asse. Non è questo il caso dell'esempio che segue

```
...
N50 WHEN TRUE DO TRAILOF(Y,X)
N60 Y100
...
```

In questo caso l'asse non è stato svincolato in tempo utile in quanto l'azione sincrona attiva nel blocco, viene attivata con `TRAILOF` sincrono con `N60`, vedere capitolo Azioni sincrone al movimento, "Struttura, concetti fondamentali".

Per evitare situazioni di conflitto, si deve procedere

nel seguente modo

```
...
N50 WHEN TRUE DO TRAILOF(Y,X)
N55 WAITP(Y)
N60 Y100
```


10.4.22 Accoppiamento al valore master (LEADON, LEADOF)

Funzione

L'accoppiamento al valore master assiale può essere programmato nelle azioni sincrone senza limitazioni di nessun tipo. La variazione di una tabella di curve con accoppiamento attivo senza una precedente risincronizzazione, è possibile in modo opzionale solo nelle azioni sincrone.

Programmazione

Attivare l'accoppiamento al valore master (di riferimento)

DO LEADON(asse slave, asse master, tabella di curve
Nr., OVW)

Disattivare l'accoppiamento al valore master (di riferimento)

DO LEADOF(asse a seguire, asse master, asse master
2)

Parametri

Attivazione dell'accoppiamento al
valore master assiale:

...DO LEADON(FA, LA, NR, OVW)

con:

FA: Asse slave

LA: Asse master

NR: Numero della tabella di curve memorizzata

OVW: Consentire la sovrascrittura di una tabella già
esistente con tabella di curve modificata

Disattivazione dell'accoppiamento al
valore master assiale:

...DO LEADOF(FA, LA)

con:

FA: Asse slave

LA: Asse master, opzionale

... DO LEADOF(FA)

forma abbreviata senza indicazione dell'asse master

Attivazione dell'accesso tramite azioni sincrone RELEASE

Per attivare un asse da accoppiare per l'accesso tramite azioni sincrone, è necessario richiamare prima la funzione RELEASE per l'asse slave da accoppiare.

Esempio:

RELEASE (XKAN)

ID=1 every SR1==1 to LEADON(CACH,XKAN,1)

OVW=0 (valore di default)

Ad un'accoppiamento esistente non è possibile trasmettere nessuna tabella di curve senza una nuova sincronizzazione. Una modifica della tabella di curve richiede prima una disattivazione dell'accoppiamento attivo e poi una successiva attivazione con il numero di tabella di curve modificata. Questo comporta una risincronizzazione dell'accoppiamento.

Modifica della tabella di curve con accoppiamento esistente tramite OVW=1

Con OVW=1 si può trasmettere una nuova tabella di curve ad un'accoppiamento esistente. Non avviene alcuna risincronizzazione. L'asse slave cerca di seguire il più rapidamente possibile i valori di posizione trasmessi dalla nuova tabella di curve.

Esempio di distacco al volo

Un materiale estruso in continuazione deve essere tagliato a misura transitando sotto un dispositivo di taglio.

Asse X: asse su cui si muove il materiale estruso. SCP

Asse X1: asse macchina del materiale estruso, SCM

Asse Y: asse lungo il quale il dispositivo di taglio "muove sincronizzato" con il materiale

Si presume che l'avanzamento dell'accessorio separatore e il suo comando vengano controllati da PLC. Per stabilire il sincronismo tra il materiale e l'utensile di taglio si possono valutare i segnali dell'interfaccia PLC.

Azioni

Attivazione dell'accoppiamento, LEADON

Disattivazione dell'accoppiamento, LEADOF

Impostazione del valore reale, PRESETON

```
%_N_SCHERE1_MPF
;$PATH=/_N_WKS_DIR/_N_DEMOFBE_WPD
N100 R3=1500 ;Lunghezza di un pezzo da tagliare
N200 R2=100000 R13=R2/300
N300 R4=100000
N400 R6=30 ;Posizione di partenza dell'asse Y
N500 R1=1 ;Condizione di partenza per l'asse del nastro
;trasportatore
N600 LEADOF(Y,X) ;Cancellazione di un eventuale accoppiamento
;preesistente
N700 CTABDEF(Y,X,1,0) ;Definizione della tabella
N800 X=30 Y=30 ;Coppie di valori
N900 X=R13 Y=R13
N1000 X=2*R13 Y=30
N1100 CTABEND ;Fine della definizione della tabella
N1200 PRESETON(X1,0) ;PRESET all'inizio
N1300 Y=R6 GO ;Posizione iniziale asse Y, l'asse è lineare
N1400 ID=1 WHENEVER $AA_IW[X]>$R3 DO PRESETON(X1,0)
;PRESET in funzione della lunghezza R3, nuovo inizio dopo il taglio
N1500 RELEASE(Y)
N1800 ID=6 EVERY $AA_IM[X]<10 DO LEADON(Y,X,1)
;Accoppiare Y a X mediante tabella 1 con X < 10
N1900 ID=10 EVERY $AA_IM[X]>$R3-30 DO LEADOF(Y,X)
;Disaccoppiare >30 della lunghezza di taglio
;eseguita
N2000 WAITP(X)
N2100 ID=7 WHEN $R1==1 DO MOV[X]=1 ;Mantenere in movimento continuo l'asse di
FA[X]=$R4 ;alimentazione
N2200 M30
```

10.4.23 Misura (MEAWA, MEAC)

Funzione

Al contrario di quando viene impiegata in blocchi di movimento del partprogram, la funzione di misura nelle azioni sincrone può essere attivata e disattivata a piacere.

Per ulteriori informazioni sulle misure, vedere le istruzioni speciali per il percorso "Funzione di misura ampliata"

Programmazione

Misura assiale senza cancellazione del percorso residuo

```
MEAWA[Asse] = (Modo, Evento di trigger_1, ..._4)
```

oppure

Misura continua senza cancellazione del percorso residuo

```
MEAC[Asse] = (Modo, memoria di misura, Evento di trigger_1, ..._4))
```

Parametri

DO MEAWA	Attivazione misura assiale	
DO MEAC	Attivazione misura continua	
Asse	Nome dell'asse per il quale avviene la misura	
Modo	Impostazione della decade delle decine	Impostazione della decade delle unità
	0: sistema di misura attivo	0: interruzione funzione di misura
	numero dei sistemi di misura (in base al modo)	fino a 4 diversi eventi di trigger attivabili
	1: 1. sistema di misura	1: contemporaneamente
	2: 2. sistema di misura	2: in sequenza
	3: entrambi i sistemi di misura	3: come 2 ma senza sorveglianza dell'evento di trigger1 allo start
Evento di trigger_1 ... _4	: fronte di salita del tastatore di misura 1	
	-1: fronte di discesa del tastatore di misura 1	
	2: fronte di salita del tastatore di misura 2	
	-2: fronte di discesa del tastatore di misura 2, opzionale	
Memoria di misura	Numero della memoria ad anello FIFO	

10.4.24 Inizializzazione delle variabili di campo con SET, REP

Funzione

Nelle azioni sincrone è possibile inizializzare le variabili di campo o descriverle con determinati valori.

Programmazione

```
DO CAMPO [n,m] = SET(valore1, valore2, ...)
```

oppure

```
DO CAMPO [n,m] = REP(valore)
```

L'inizializzazione parte dall'indice di campo programmato. Nel caso dei campi bidimensionali viene prima incrementato il secondo indice; non viene invece elaborato per gli indici d'asse.

Assegnazioni di valori delle variabili di campo

Sono possibili solo le variabili descrivibili nelle azioni sincrone. I dati macchina non sono pertanto inizializzabili. Le variabili d'asse non sono definibili con il valore NO_AXIS.

SET(lista valori)

Inizializzazione con liste di valori

REP (valore)

Inizializzazione con valori uguali

Lista di valori

Con il numero dei valori specificati

Valore

Con lo stesso valore fino alla fine del campo

SET(lista valori)

Il campo viene descritto con i parametri di SET dagli indici di campo programmati. Vengono assegnati tanti elementi di campo quanti valori sono stati programmati. Se i valori programmati superano gli elementi di campo rimanenti, viene generato un allarme dal sistema.

REP(valore)

Il campo viene descritto ripetutamente con i parametri di REP dagli indici di campo programmati fino alla fine del campo.

Esempio

```
WHEN TRUE DO SYG_IS[0]=REP(0)
WHEN TRUE DO SYG_IS[1]=SET(3,4,5)
Risultato:
SYG_IS[0]=0
SYG_IS[1]=3
SYG_IS[2]=4
SYG_IS[3]=5
SYG_IS[4]=0
```

10.4.25 Impostazione/cancellazione delle label di attesa con SETM, CLEARM

Funzione

In azioni sincrone le label di attesa possono essere settate o cancellate per es. per coordinare fra loro i programmi.

Programmazione

```
DO SETM(numero marker)
oppure
DO CLEARM(numero marker)
```

Impostazione/cancellazione delle label di attesa per il canale

SETM(numero marker)	Set della label di attesa del canale
CLEARM(numero marker)	Cancellazione della label di attesa del canale

SETM

L'istruzione `SETM` può essere scritta in un partprogram e in una parte di azione facente parte di un'azione sincrona. Essa setta la label Numero marker per il canale in cui si trova l'istruzione.

CLEARM

L'istruzione `CLEARM` può essere scritta in un partprogram e in una parte di azione facente parte di un'azione sincrona. Essa cancella la label (NumeroMarker) per il canale in cui si trova l'istruzione.

10.4.26 Reazioni agli errori in caso di allarmi dei cicli SETAL

Funzione

Le reazioni ad errori possono essere programmate con azioni sincrone interrogando variabili di stato e attivando le relative azioni.

Le possibili reazioni agli errori sono le seguenti:

- arresto dell'asse: `override=0`
- emissione di un allarme: con SETAL possono essere settati allarmi per cicli da azioni sincrone.
- settare un'uscita
- tutte le azioni possibili in azioni sincrone

Settare gli allarmi dei cicli

```
DO SETAL(numero di allarme)
```

Intervallo di allarme dei cicli per l'utente: da 65000 a 69999

Esempio

```
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO $AA_OVR[X2]=0
;Quando la distanza di sicurezza tra gli assi X1 e X2 è troppo piccola, arrestare
;l'asse X2.
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO SETAL(61000)
;Quando la distanza di sicurezza tra gli assi X1 e X2 è troppo piccola, settare
;l'allarme.
```

10.4.27 Posizionamento su riscontro fisso (FXS e FOCON/FOCOF)

Funzione

I comandi per **posizionamento su riscontro fisso** sono programmati con le istruzioni FXS, FXST e FXSW in azioni sincrone/cicli tecnologici.

L'attivazione può avvenire senza movimento, la coppia viene limitata immediatamente. Non appena l'asse si sposta in direzione del setpoint, si attiva la sorveglianza di riscontro fisso.

Avanzamento con coppia/forza limitata (FOC):

La funzione consente di modificare in qualsiasi momento la coppia/forza tramite azioni sincrone e può essere attivata come modale o riferita al blocco.

Parametri

FXS [asse]	Selezione solo in sistemi con azionamenti digitali (VSA, HSA, HLA)
FXST [asse]	Variazione della coppia di spinta FXST
FXSW [asse]	Variazione della finestra di sorveglianza FXSW
FOCON [asse]	Attivazione della limitazione di coppia/potenza con efficacia modale
FOCOF [asse]	Disattivare la limitazione coppia/potenza
FOCON/FOCOF	La programmazione dell'asse avviene nelle parentesi quadre. Sono consentiti: – identificatore assi geometrici – identificatore assi canale – identificatore assi macchina

Nota

Una selezione può avvenire solo una volta.

Esempio di posizionamento su riscontro fisso (FXS)

Attivato da una azione sincrona

```
Asse Y: ;Attivazione di azioni sincrone statiche
attivazione:
N10 IDS=1 WHENEVER (($R1==1) AND ;Settando $R1=1 viene attivato FXS per
```

	($\$AA\_FXS[Y]==0$) DO \$R1=0 FXS[Y]=1 FXST[Y]=10 FA[Y]=200 POS[Y]=150	;l'asse Y, la coppia effettiva viene ridotta al 10% ;ed inizia un movimento in direzione del ;riscontro fisso
N11	IDS=2 WHENEVER ($\$AA\_FXS[Y]==4$) DO FXST[Y]=30	;Non appena viene riconosciuto il riscontro fisso ;($\$AA\_FXS[Y]==4$), la coppia viene ridotta ;al 30%
N12	IDS=3 WHENEVER ($\$AA\_FXS[Y]==1$) DO FXST[Y]=\$R0	;Al raggiungimento del riscontro fisso viene ;gestita la coppia in base a R0
N13	IDS=4 WHENEVER (($\$R3==1$) AND ($\$AA\_FXS[Y]==1$)) DO FXS[Y]=0 FA[Y]=1000 POS[Y]=0	;Disattivazione in funzione ;di R3 e ;ritorno
N20	FXS[Y]=0 G0 G90 X0 Y0	; normale esecuzione del programma: ;Asse Y per la
N30	RELEASE(Y)	;abilitare il movimento in azione sincrona
N40	G1 F1000 X100	;movimento di un altro asse
N50		
N60	GET(Y)	;includere nuovamente l'asse Y nel gruppo di ;interpolazione

Esempio di attivazione delle limitazioni di coppia/potenza (FOC)

N10	FOCON[X]	;attivazione modale della limitazione
N20	X100 Y200 FXST[X]=15	;X muove con coppia ridotta (15%)
N30	FXST[X]=75 X20	;variazione della coppia al 75%, X si muove con ;questa coppia limitata
N40	FOCOF[X]	;disattivare la limitazione di coppia

Selezione multipla

Se per un errore di programmazione viene richiamata ancora una volta la funzione dopo la sua attivazione $FXS[asse]=1$, viene emesso l'allarme 20092 "Posizionamento su riscontro fisso ancora attivo".

Una programmazione che preveda il richiamo o di $\$AA\_FXS[]$ o di un merker proprio (qui R1), evita una attivazione multipla della funzione Frammentazione del partprogram.

```
N10 R1=0
N20 IDS=1 WHENEVER ($R1==0 AND
$AA_IW[AX3] > 7) DO R1=1 FXST[AX1]=12
```

Azioni sincrone riferite al blocco

Il posizionamento su riscontro fisso può essere attivato durante un movimento di distacco con la programmazione blocco/blocco.

Esempio:

N10 G0 G90 X0 Y0	
N20 WHEN \$AA_IW[X] > 17 DO FXS[X]=1	;Se X raggiunge una posizione maggiore di ;17mm ;viene
N30 G1 F200 X100 Y110	;attivato FXS

Azioni sincrone statiche e riferite al blocco

Nelle azioni sincrone statiche e blocco/blocco si possono utilizzare le stesse istruzioni `FXS`, `FXST` e `FXSW`, come in un normale ciclo di programma pezzo. I valori che vengono assegnati possono essere disponibili tramite un calcolo.

10.4.28 Definizione dell'angolo della tangente vettoriale nella azioni sincrone

Funzione

La variabile di sistema che può essere letta in azioni sincrone `$AC_TANEB` (**T**angent **AN**gel at **E**nd of **B**lock), rileva l'angolo tra la tangente vettoriale nel punto finale del blocco attuale e la tangente vettoriale nel punto iniziale del blocco successivo programmato.

Parametri

L'angolo tangenziale viene sempre emesso positivo nel settore da 0.0 a 180.0 gradi. Se non è presente un blocco successivo nell'esecuzione principale, l'angolo viene emesso come -180.0 gradi.

La variabile di sistema `$AC_TANEB` non dovrebbe essere letta in blocchi che vengono generati dal sistema (blocchi intermedi). La variabile di sistema `$AC_BLOCKTYPE` viene utilizzata per distinguere se si tratta di un blocco programmato (blocco principale).

Esempio

```
ID=2 EVERY $AC_BLOCKTYPE==0 DO $SR1 = $AC_TANEB
```

10.4.29 Determinazione dell'override attuale

Funzione

Override attuale

(Parte NC) può essere letta e scritta in azioni sincrone con le variabili di sistema:

`$AA_OVR` Override assiale

`$AC_OVR` Override vettoriale.

L'override impostato dal PLC viene messo a disposizione per la lettura da azioni sincrone nelle variabili di sistema:

\$AA_PLC_OVR Override assiale

\$AC_PLC_OVR Override vettoriale.

Override risultante

viene messo a disposizione per la lettura da azioni sincrone nelle variabili di sistema

\$AA_TOTAL_OVR Override assiale

\$AC_TOTAL_OVR Override vettoriale.

L'override risultante si ricava come:

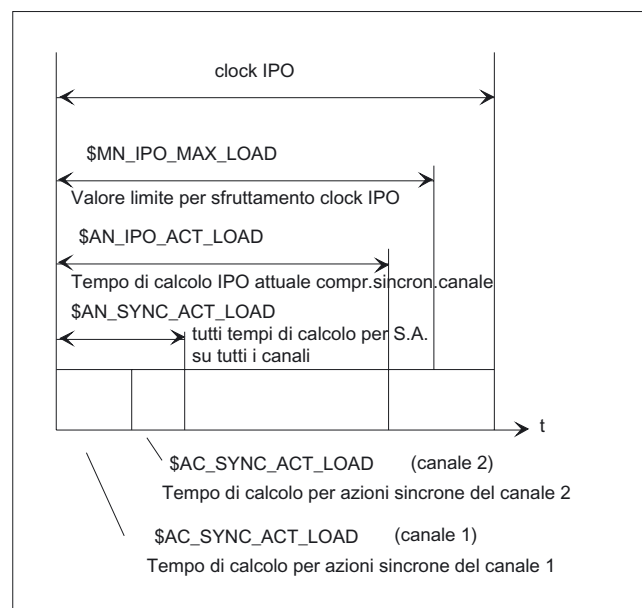
\$AA_OVR * \$AA_PLC_OVR oppure

\$AC_OVR * \$AC_PLC_OVR

10.4.30 Rilevazione del sovraccarico tramite il fabbisogno di tempo delle azioni sincrone

Funzione

In un ciclo di interpolazione, da parte dell'NC devono essere interpretate le azioni sincrone e calcolati i movimenti, etc. Con le variabili di sistema descritte in seguito, le azioni sincrone possono ricavare le informazioni sulle quote di tempo delle azioni sincrone nel clock di interpolazione e sul tempo di calcolo del regolatore di posizione.



Parametri

Le variabili contengono valori validi solo se il dato macchina `$MN_IPO_MAX_LOAD` è maggiore di 0. In caso contrario le variabili definiscono sempre, tanto per i sistemi SINUMERIK powerline quanto per quelli solution line, il tempo di calcolo netto, per il quale le interruzioni generate da HMI non vengono più considerate. Il tempo di calcolo netto risulta da:

- tempo dell'azione sincrona,
- tempo del regolatore di posizione e
- tempo di calcolo IPO residuo senza interruzioni provocate da HMI

Le variabili contengono sempre i valori del clock IPO **precedente**.

<code>\$AN_IPO_ACT_LOAD</code>	Tempo di calcolo IPO attuale (incluse le azioni sincrone di tutti i canali)
<code>\$AN_IPO_MAX_LOAD</code>	Tempo di calcolo IPO massimo (incluse le azioni sincrone di tutti i canali)
<code>\$AN_IPO_MIN_LOAD</code>	Tempo di calcolo IPO minimo (incluse le azioni sincrone di tutti i canali)
<code>\$AN_IPO_LOAD_PERCENT</code>	Tempo di calcolo IPO attuale in rapporto al clock IPO (%).
<code>\$AN_SYNC_ACT_LOAD</code>	Tempo di calcolo attuale per le azioni sincrone di tutti i canali
<code>\$AN_SYNC_MAX_LOAD</code>	Tempo di calcolo massimo per le azioni sincrone di tutti i canali
<code>\$AN_SYNC_TO_IPO</code>	Quota percentuale delle azioni sincrone globali sul tempo di calcolo IPO globale (su tutti i canali)
<code>\$AC_SYNC_ACT_LOAD</code>	Tempo di calcolo attuale per le azioni sincrone del canale
<code>\$AC_SYNC_MAX_LOAD</code>	Tempo di calcolo massimo per le azioni sincrone del canale
<code>\$AC_SYNC_AVERAGE_LOAD</code>	Tempo di calcolo medio per le azioni sincrone del canale
<code>\$AN_SERVO_ACT_LOAD</code>	Tempo di calcolo attuale del regolatore di posizione
<code>\$AN_SERVO_MAX_LOAD</code>	Tempo di calcolo massimo del regolatore di posizione
<code>\$AN_SERVO_MIN_LOAD</code>	Tempo di calcolo minimo del regolatore di posizione

Variabile per la segnalazione del sovraccarico:

Tramite il dato macchina `$MN_IPO_MAX_LOAD` viene impostato, a partire da quale tempo di calcolo IPO netto (in % del clock IPO) la variabile di sistema `$AN_IPO_LOAD_LIMIT` deve essere impostata su TRUE. Se il carico attuale ritorna al di sotto di questo limite, la variabile viene nuovamente settata a FALSE. Con il DM = 0, viene disattivata completamente la funzione di diagnostica.

Valutando `$AN_IPO_LOAD_LIMIT` l'utente può stabilire una propria strategia per evitare un overflow di livello.

10.5 Cicli tecnologici

Funzione

Come azione in azioni sincrone si possono richiamare anche dei programmi, che però vanno strutturati come funzioni che sono ammesse anche come azioni in azioni sincrone. I programmi così strutturati sono detti cicli tecnologici.

I cicli tecnologici vengono memorizzati nel controllo come sottoprogrammi.

In un canale si possono eseguire parallelamente diversi cicli tecnologici o azioni.

Programmazione

- La fine del programma viene programmata con M02/M17/M30/RET.
- Entro un livello di programma tutte le azioni specificate in ICYCOF possono essere elaborate in un clock senza cicli di attesa.
- È possibile interrogare in successione fino a 8 cicli tecnologici per ciascuna azione sincrona.
- I cicli tecnologici sono possibili anche in azioni sincrone blocco per blocco.
- È possibile programmare sia costrutti di controllo IF, sia istruzioni di salto GOTO, GOTOF e GOTOB.

Blocchi con istruzioni DEF e DEFINE nei cicli tecnologici

- Le istruzioni DEF e DEFINE vengono ignorate nei cicli tecnologici
- In presenza di una sintassi non corretta o incompleta, provocano un messaggio di allarme
- I blocchi possono essere ignorati senza creazione della variabile e senza messaggio di allarme
- Vengono completamente considerate con assegnazioni di valori come ciclo di partprogram

Assegnazione di parametri

È possibile trasmettere parametri ai cicli tecnologici. Vengono considerati tanto i tipi di dati semplici, che vengono passati come parametri formali "Call by Value", quanto le impostazioni di default, che si attivano alla chiamata dei cicli tecnologici. Si tratta di:

- I valori di default programmati, se non è programmato alcun parametro di trasferimento.
- Parametri di default con valori iniziali da assegnare.
- Parametri attuali non inizializzati trasferiti con un valore di default.

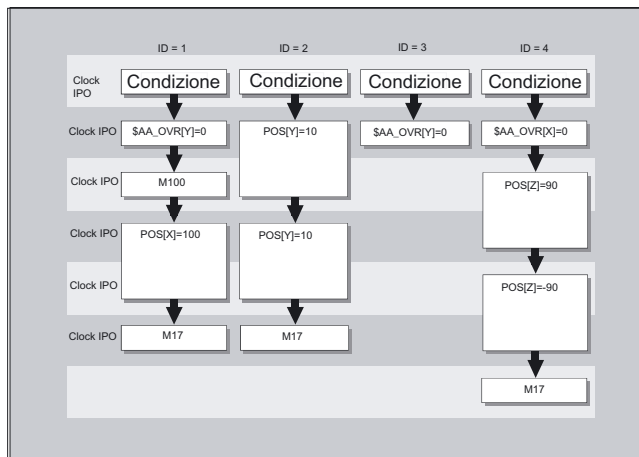
Procedura

I cicli tecnologici vengono startati non appena vengono soddisfatte le proprie condizioni. Ogni riga di un ciclo tecnologico viene elaborata in un clock IPO separato. Per assi di posizionamento, per l'esecuzione, sono necessari più clock IPO. Altre funzioni vengono eseguite in un solo clock. Nel ciclo tecnologico l'esecuzione dei blocchi avviene in maniera sequenziale.

Se in uno stesso clock di interpolazione vengono richiamate azioni che si escludono a vicenda, si attiva l'azione che l'azione sincrona richiama con il numero ID più alto.

Esempio

Settando gli ingressi digitali vengono avviati i programmi asse.



Programma principale:

```

ID=1 EVERY $A_IN[1]==1 DO ASSE_X

ID=2 EVERY $A_IN[2]==1 DO ASSE_Y

ID=3 EVERY $A_IN[3]==1 DO $AA_OVR[Y]=0

ID=4 EVERY $A_IN[4]==1 DO ASSE_Z

M30

```

Ciclo tecnologico ASSE_X:

```

$AA_OVR[Y]=0
M100
POS[X]=100 FA[X]=300
M17

```

Ciclo tecnologico ASSE_Y:

```

POS[Y]=10 FA[Y]=200
POS[Y]=-10
M17

```

Ciclo tecnologico ASSE_Z:

```

$AA_OVR[X]=0
POS[Z]=90 FA[Z]=250
POS[Z]=-90
M17

```

Se ... azione

```

;Attivare il programma asse X quando l'ingresso
;1 è a 1
;Attivare il programma asse Y quando l'ingresso
;2 è a 1
;Settare l'override dell'asse Y a 0 quando
;l'ingresso 3 è a 1
;Attivare il programma asse Z quando l'ingresso
;4 è a 1

```

Esempi di diverse sequenze di programma nel ciclo tecnologico

```
PROC CYCLE
N10 DEF REAL "valore"=12.3
N15 DEFINE ABC AS G01
```

Entrambi i blocchi vengono ignorati senza allarme e senza creazione della variabile o della macro.

```
PROC CYCLE
N10 DEF REAL
N15 DEFINE ABC G01
```

Entrambi i blocchi provocano l'allarme NC poiché la sintassi non è scritta in modo corretto.

```
PROC CYCLE
N10 DEF AXIS "asse1"=XX2
```

Se l'asse XX2 non è noto, viene emesso l'allarme 12080. In caso contrario, il blocco viene ignorato senza allarme e senza creazione della variabile.

```
PROC CYCLE
N10 DEF AXIS "asse1"
N15 G01 X100 F1000
N20 DEF REAL"valore1"
```

Il blocco N20 provoca sempre l'allarme 14500 perché l'istruzione DEF non è consentita dopo la prima riga di programma.

10.5.1 Variabile di contesto (\$P_TECCYCLE)

Funzione

Con l'ausilio della variabile \$P_TECCYCLE è possibile suddividere i programmi in programmi di azione sincrona e programmi di preelaborazione. In questo modo è possibile ottenere dalla sintassi blocchi o sequenze di programma scritti correttamente, in alternativa anche come ciclo di partprogram.

Interpretazione della variabile di contesto

La variabile di sistema \$P_TECCYCLE consente di controllare interpretazioni specifiche per contesto di parti del programma in cicli tecnologici se

```
IF $P_TECCYCLE==TRUE
```

Sequenza di programma per ciclo tecnologico nell'azione sincrona

altrimenti

```
ELSE
```

Sequenza di programma per ciclo di partprogram

Nota

Un blocco con sintassi di programma scorretta o non consentita e assegnazioni di valori sconosciute provoca, anche nel ciclo di partprogram, un messaggio di allarme.

Esempio di sequenza di programma con query di \$P_TECCYCLE nel ciclo tecnologico

```
PROC CYCLE
N10 DEF REAL "valore1" ;viene ignorato nel ciclo tecnologico
N15 G01 X100 F1000
N20 IF $P_TECCYCLE==TRUE
N25 "Sequenza di programma per ciclo tecnologico (senza variabile valore1)"
N30 ELSE
N35 "Sequenza di programma per ciclo di partprogram (la variabile valore1 è
presente)"
ENDIF
```

10.5.2 Parametri Call by Value

Funzione

I cicli tecnologici possono essere definiti con parametri Call by Value. Come parametri sono possibili i tipi semplici di dati, come INT, REAL, CHAR, STRING, AXIS e BOOL.

Nota

I parametri formali che vengono trasferiti a Call by Value non possono essere dei campi.

I parametri attuali possono essere costituiti anche di parametri di default;
vedere il capitolo "Inizializzazione dei parametri di default".

Programmazione

```
ID=1 WHEN $AA_IW[X]>50 DO TEC(IVAL, RVAL, , SVAL, AVAL)
;in caso di parametri attuali non inizializzati viene passato un valore di default.
ID=1 WHE $AA_IW[X]>50 DO TEC(IVAL, RVAL, , SYG_SS[0], AVAL)
```

10.5.3 Inizializzazione dei parametri di default

Funzione

Con l'istruzione PROC, i parametri di default possono anche essere dotati di un valore iniziale.

Programmazione

Assegnazione dei parametri di default nel ciclo tecnologico:

```
PROC TEC (INT IVAL=1, REAL RVAL=1.0, CHAR CVAL='A', STRING[10] SVAL="ABC", AXIS
AVAL=X, BOOL BVAL=TRUE)
```

Se un parametro attuale è costituito da un parametro di default, il valore iniziale viene trasferito dall'istruzione PROC. Ciò vale sia nel partprogram, sia nelle azioni sincrone.

Esempio

```
TEC (IVAL, RVAL, , SVAL, AVAL) ;per CVAL e BVAL vale il valore iniziale
```

10.5.4 Controllo ed esecuzione di cicli tecnologici (ICYCOF, ICYCON)

Funzione

Per il controllo dell'esecuzione temporale dei cicli tecnologici si utilizzano i comandi ICYCOF e ICYCON.

Con ICYCOF tutti i blocchi di un ciclo tecnologico vengono eseguiti in un unico clock di interpolazione. Tutte le azioni, la cui esecuzione richiede più clock, provocano con ICYCOF dei processi paralleli di elaborazione.

Applicazione

Con ICYCON i movimenti degli assi di comando possono avere come conseguenza che l'esecuzione di un ciclo tecnologico subisca un ritardo. Se si desidera evitarlo, con ICYCOF è possibile eseguire tutte le azioni senza tempi di attesa in clock di interpolazione.

Programmazione

Per l'esecuzione ciclica dei cicli di elaborazione vale:

ICYCON	ogni blocco di un ciclo tecnologico viene eseguito secondo ICYCON in un clock IPO separato
ICYCOF	tutti i blocchi seguenti di un ciclo tecnologico vengono eseguiti con ICYCOF in un unico clock IPO

Nota

I due comandi ICYCON e ICYCOF agiscono solo all'interno del livello di programma. Nel partprogram questi due comandi vengono semplicemente ignorati (nessuna reazione).

Esempio di modalità di esecuzione ICYCOF

clock IPO	PROC TECHNOCYC
1.	\$R1=1
2.25	POS[X]=100
26.	ICYCOF
26.	\$R1=2
26.	\$R2=\$R1+1
26.	POS[X]=110
26.	\$R3=3
26.	RET

10.5.5 Strutturazioni in cascata dei cicli tecnologici**Funzione**

È possibile eseguire fino a 8 cicli tecnologici collegati in serie. In questo modo, in un'azione sincrona sono programmabili più cicli tecnologici.

Programmazione

```
ID=1 WHEN $AA_IW[X]>50 DO TEC1($R1) TEC2 TEC3(X)
```

Sequenza operativa

I cicli tecnologici vengono eseguiti secondo la serie (in cascata) da sinistra a destra, conformemente alla programmazione sopra indicata. Se un ciclo viene eseguito nella modalità ICYCON, vengono ritardate tutte le elaborazioni successive. Viene emesso un allarme che interrompe tutte le azioni successive.

10.5.6 Cicli tecnologici in azioni sincrone blocco per blocco

Funzione

I cicli tecnologici sono possibili anche in azioni sincrone blocco per blocco.

Se il tempo di esecuzione di un ciclo tecnologico è maggiore del tempo di elaborazione del blocco relativo, il ciclo tecnologico viene interrotto al cambio blocco.

Nota

Un ciclo tecnologico non impedisce il cambio blocco.

10.5.7 Costrutti di controllo IF

Funzione

Per le diramazioni nelle frequenza del flusso dei cicli tecnologici è possibile utilizzare costrutti di controllo IF nelle azioni sincrone.

Programmazione

```
IF <condizione>  
    $R1=1  
[ELSE]                               ;opzionale  
    $R1=0  
ENDIF
```

10.5.8 Istruzioni di salto (GOTO, GOTOF, GOTOB)

Funzione

Nei cicli tecnologici sono possibili le istruzioni di salto GOTO, GOTOF, GOTOB. Le label specificate devono essere presenti nel sottoprogramma affinché non venga emesso un allarme.

Nota

Le label e i numeri di blocco possono essere solo delle costanti.

Programmazione**Salta incondizionali**

GOTO label, numero di blocco

oppure

GOTOF label, numero di blocco

oppure

GOTOB label, numero di blocco

Istruzioni di salto e destinazioni del salto

GOTO	Salta prima in avanti e poi indietro
GOTOF	Salta in avanti
GOTOB	Salta indietro
Label:	Label di salto
Numero di blocco	Destinazione del salto: questo blocco
N100	Il numero di blocco è il blocco secondario
:100	Il numero di blocco è il blocco principale

10.5.9 Blocco, abilitazione, interruzione (LOCK, UNLOCK, RESET)**Funzione**

L'esecuzione di un ciclo tecnologico originato da azioni sincrone modali può essere bloccato o nuovamente abilitato da altre azioni sincrone modali, così come è anche possibile ripristinare un ciclo tecnologico.

Programmazione

LOCK (n, n, ...)	Blocco delle azioni sincrone, l'azione attiva viene interrotta
UNLOCK (n, n, ...)	Abilitazione delle azioni sincrone
RESET (n, n, ...)	Reset del ciclo tecnologico
n	Numero di identificazione dell'azione sincrona

Interblocco da PLC

Le azioni sincrone modali con i numeri ID

n=1 ... 64 possono essere bloccate dal PLC. La condizione corrispondente non viene più valutata e l'esecuzione della funzione corrispondente viene bloccata nell'NCK.

Mediante un segnale all'interfaccia PLC è possibile bloccare tutte le azioni sincrone in blocco.

Nota

Un'azione sincrona programmata è attiva in modo standard e può essere protetta contro la lettura e la scrittura mediante un dato macchina.

Le azioni sincrone predefinite dal costruttore della macchina non devono poter essere modificate dal cliente finale.

Esempio**Blocco delle azioni sincrone, LOCK**

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
```

Abilitazione delle azioni sincrone, UNLOCK

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
...
N250 ID=3 WHENEVER $A_IN[3]==1 DO UNLOCK(1)
```

Interruzione delle azione sincrona, RESET

```
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO RESET(1)
```

10.6 Cancellazione dell'azione sincrona (CANCEL)

Funzione

Le azioni sincrone modali con la definizione ID(S)=n possono essere cancellate con CANCEL solo direttamente dal partprogram.

Nota

I movimenti attivati da un'azione sincrona che sono ancora in corso vengono conclusi secondo quanto indicato dalla programmazione.

Programmazione

`CANCEL (n, n, ...)`
`n`

Cancellare l'azione sincrona
Numero di identificazione dell'azione
sincrona

Esempio

```
N100 ID=2 WHENEVER $A_IN[1]==1 DO M130  
...  
N200 CANCEL(2)
```

;Cancellare l'azione sincrona nr. 2

10.7 Condizioni marginali

Funzione

Vi sono condizioni marginali al verificarsi dei seguenti eventi:

- **Power On**
- **Cambio del modo operativo**
- **Reset**
- **NC-Stop**
- **Fine programma**
- **Ricerca blocco**
- **Interruzione del programma con sottoprogramma asincrono ASUP**
- **Riposizionamento REPOS**
- **Disattivazione con CANCEL**

Eventi

- **Power On**

Con Power On normalmente non sono attive azioni sincrone. Le azioni sincrone statiche possono essere attivate con un sottoprogramma asincrono (ASUP) avviato da PLC.

- **Cambio del modo operativo**

Le azioni sincrone attivate con la parola chiave `IDS` restano attive anche dopo un cambio del modo operativo. Tutte le altre azioni sincrone perdono validità al cambio del modo operativo (per es. posizionamento asse) e ritornano attive dopo il riposizionamento e il ritorno al funzionamento automatico.

- **Reset**

Con Reset NC vengono terminate tutte le azioni attivate tramite azioni sincrone blocco per blocco e modali. Le azioni sincrone statiche rimangono attive. Da esse si possono far partire nuove azioni. Se al **RESET** è attivo un movimento dell'asse di comando, lo stesso viene interrotto. Dopo **RESET** le azioni sincrone del tipo **WHEN** che sono state già eseguite non vengono più elaborate.

Comportamento dopo RESET		
Azione sincrona/ ciclo tecnologico	Modale/blocco-blocco	Statica (IDS)
	l'azione attiva viene interrotta, le azioni sincrone vengono cancellate	l'azione attiva viene interrotta, il ciclo tecnologico viene resettato
Asse/ mandrino di posizionamento	il movimento viene interrotto	il movimento viene interrotto
Mandrino regolato in velocità	\$MA_SPIND_ACTIVE_AFTER_RESET==1: il mandrino resta attivo \$MA_SPIND_ACTIVE_AFTER_RESET==0: il mandrino si arresta.	
Accoppiamento del valore master	\$MC_RESET_MODE_MASK, Bit13 == 1: l'accoppiamento del valore master resta attivo \$MC_RESET_MODE_MASK, Bit13 == 0: l'accoppiamento del valore master viene disattivato	
Misure	Le misure attivate da azioni sincrone vengono interrotte	le misure attivate da azioni sincrone statiche vengono interrotte

- **NC-Stop**

Statiche Le azioni sincrone restano attive anche con NC-Stop. I movimenti attivati mediante azioni sincrone statiche non vengono interrotti. Azioni sincrone **locali del programma** appartenenti al blocco attivo mantengono la loro validità, mentre i movimenti da esse attivati vengono interrotti.

- **Fine programma**

La fine del programma e le azioni sincrone non si influenzano reciprocamente. Le azioni sincrone attive vengono proseguite anche dopo la fine del programma. Le azioni sincrone attive in un blocco M30 rimangono attive nel blocco M30; se questo non è consentito, occorre interrompere l'azione sincrona prima della fine programma con **CANCEL** (vedere capitolo precedente).

Comportamento dopo la fine programma		
Azione sincrona/ ciclo tecnologico	Modale e blocco per blocco vengono interrotti	Statici (IDS) rimangono attivi
Asse/ mandrino di posizionamento	M30 viene ritardata fino a quando l'asse/il mandrino è fermo.	Il movimento prosegue
Mandrino regolato in velocità	Fine programma: \$MA_SPIND_ACTIVE_AFTER_RESET==1 : il mandrino resta attivo \$MA_SPIND_ACTIVE_AFTER_RESET==0 : il mandrino si arresta Al cambio del modo operativo il mandrino resta attivo	Il mandrino rimane attivo
Accoppiamento del valore master	\$MC_RESET_MODE_MASK, Bit13 == 1: l'accoppiamento del valore master resta attivo \$MC_RESET_MODE_MASK, Bit13 == 0: l'accoppiamento del valore master viene disattivato	L'accoppiamento generato da azioni sincrone statiche rimane attivo
Misure	Le misure attivate da azioni sincrone vengono interrotte	Le misure attivate da azioni sincrone statiche rimangono attive

- **Ricerca blocco**

Durante la ricerca blocco le azioni sincrone vengono raggruppate e valutate con Start NC; le relative azioni vengono all'occorrenza avviate. Le azioni sincrone statiche sono attive anche durante la ricerca blocco. Se la ricerca blocco attivata con `FCTDEF` trova dei coefficienti di polinomi programmati, essi vengono attivati direttamente.

- **Interruzione del programma con sottoprogramma asincrono ASUP**

Inizio ASUP:

le azioni sincrone di movimento modali e statiche vengono mantenute e sono attive anche nel sottoprogramma asincrono.

Fine ASUP:

Se il sottoprogramma asincrono non viene proseguito con `REPOS`, le azioni sincrone modali e statiche che vi erano state modificate continuano ad essere attive nel programma principale.

- **Riposizionamento REPOS**

Dopo un riposizionamento `REPOS` le azioni sincrone attive nel blocco interrotto ridiventano attive. Le azioni sincrone modali modificate da un sottoprogramma asincrono non sono più valide quando, dopo `REPOS`, viene proseguita l'esecuzione del blocco. I coefficienti di polinomi programmati con `FCTDEF` non vengono influenzati da sottoprogrammi asincroni e da `REPOS`. Indipendentemente da dove siano stati programmati, essi possono essere impiegati in qualunque momento nel sottoprogramma asincrono e nel programma principale, anche in seguito a `REPOS`.

- **Disattivazione con CANCEL**

La disattivazione di un'azione sincrona con `CANCEL` non ha influenza sull'azione attiva. I movimenti di posizionamento vengono portati a termine come programmato.

Con l'istruzione `CANCEL` si può interrompere un'azione sincrona modale o statica attiva. Se un'azione sincrona viene interrotta mentre è ancora attivo il posizionamento, esso viene portato a termine. Se non si desidera questo comportamento, è possibile frenare il movimento dell'asse con cancellazione del percorso residuo prima del comando `CANCEL`.

Esempio di disattivazione con CANCEL

ID=17 EVERY \$A_IN[3]==1 DO POS[X]=15 FA[X]=1500	;Avviare il movimento dell'asse di ;posizionamento
...	
WHEN ... DO DELDTG(X)	;Terminare il movimento dell'asse di ;posizionamento
CANCEL(1)	

Pendolamento

11.1 Pendolamento asincrono

Funzione

Un asse di pendolamento si muove tra i punti di inversione 1 e 2 con avanzamento preimpostato fino a che non viene disattivato il movimento di pendolamento.

Durante il movimento di pendolamento gli altri assi possono essere interpolati liberamente. Con un movimento di contornitura oppure con un asse di posizionamento è possibile ottenere un incremento continuo. Non esiste però **alcuna relazione** tra movimento di pendolamento e di incremento.

Caratteristiche del pendolamento asincrono

- Il pendolamento asincrono è attivo per un asse specifico anche in corrispondenza di fine blocco.
- Tramite il partprogram è garantito un inserimento sincrono al blocco del movimento di pendolamento.
- Non è possibile una interpolazione comune di più assi ed una sovrapposizione di tratti di pendolamento.

Programmazione

Durante l'esecuzione del programma NC è possibile inserire ed influenzare con i seguenti indirizzi il pendolamento asincrono direttamente dal partprogram.

I valori programmati, sincroni al blocco nell'elaborazione principale, vengono registrati nei corrispondenti dati setting e restano attivi fino alla successiva modifica.

Attivazione/disattivazione del pendolamento: OS

OS [Asse] = 1: attivazione

OS [Asse] = 0: disattivazione

Parametri

OSP1 [asse] =	posizione del punto di inversione 1 (pendolamento: punto di inversione sinistro)
OSP2 [asse] =	posizione del punto di inversione 2 (pendolamento: punto di inversione destro)
OST1 [asse] =	tempo di arresto nei punti di inversione in secondi
OST2 [asse] =	
FA [asse] =	avanzamento dell'asse di pendolamento
OSCTRL [asse] =	(opzioni di set e di reset)

OSNSC [Asse] =	numero delle passate di spegnifiama
OSE [asse] =	posizione finale
OS [asse] =	1 = attivare l'asse di pendolamento; 0 = disattivare l'asse di pendolamento

Tempi di sosta nei punti di inversione: OST1, OST2

Tempo di sosta	Comportamento nel campo di arresto preciso sul punto di inversione
-2	L'interpolazione prosegue senza attendere l'arresto preciso
-1	Attendere l'arresto preciso grossolano
0	Attendere l'arresto preciso fine
>0	Attendere l'arresto preciso fine e poi attendere il tempo di sosta

L'unità per il tempo di sosta è identica a quella programmata con G4 .

Esempio: l'asse deve pendolare tra due punti di inversione

L'asse Z deve pendolare tra le posizioni 10 e 100. Il punto di inversione 1 viene raggiunto con arresto preciso fine, il punto di inversione 2 con arresto preciso grossolano. Si vuole lavorare con avanzamento 250 per l'asse di pendolamento. Al termine della lavorazione si vogliono eseguire 3 passate di spegnifiama e l'asse di pendolamento deve raggiungere la posizione finale 200. L'avanzamento per l'asse di incremento è 1, il punto finale di destinazione dell'asse X di incremento è 15.

WAITP (X, Y, Z)	;Posizione di partenza
G0 X100 Y100 Z100	;Commutazione in funzionamento come asse di posizionamento
N40 WAITP (X, Z)	
N50 OSP1 [Z] =10 OSP2 [Z] =100 ->	;Punto di inversione 1, punto di inversione 2
-> OSE [Z] =200 ->	;posizione finale
-> OST1 [Z] =0 OST2 [Z] =-1 ->	;Tempo di sosta in U1: arresto preciso fine; ;Tempo di sosta in U2: arresto preciso ;grossolano
-> FA [Z] =250 FA [X] =1 ->	;Avanzamento asse pendolamento e di ;incremento
-> OSCTRL [Z] = (4, 0) ->	
-> OSNSC [Z] =3 ->	;Opzioni di set
N60 OS [Z] =1	;Tre passate di spegnifiama ;Avviare il pendolamento
N70 WHEN \$A_IN [3] ==TRUE ->	;Cancellazione del percorso residuo
-> DO DELDTG (X)	
N80 POS [X] =15	;Posizioni di partenza per l'asse X
N90 POS [X] =50	
N100 OS [Z] =0	;Arresto del pendolamento
M30	

-> può essere programmato in un solo blocco.

Esempio di pendolamento con modifica on-line della posizione di inversione**Dati setting**

I dati di setting necessari per il pendolamento asincrono possono essere definiti nel partprogram.

Se i dati di setting vengono definiti direttamente nel partprogram, la variazione diventa attiva già in fase di preelaborazione. La sincronizzazione può essere ottenuta con uno stop preelaborazione STOPRE.

```

$SA_OSCILL_REVERSE_POS1[Z] = -10
$SA_OSCILL_REVERSE_POS2[Z] = 10

G0 X0 Z0
WAITP(Z)

ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X] = 0
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X] = 0
                                ;Se il valore reale dell'asse di
                                ;pendolamento ha superato il punto di
                                ;l'asse di incremento viene arrestato.
OS[Z] = 1 FA[X] = 1000 POS[X] = 40      ;Inserire il pendolamento
OS[Z] = 0                                ;Disinserire il pendolamento
M30

```

Descrizione

Per l'asse di pendolamento vale quanto segue:

- ogni asse può essere utilizzato come asse di pendolamento.
- possono essere attivati contemporaneamente più assi di pendolamento (al massimo: numero degli assi di posizionamento).
- Per l'asse di pendolamento è sempre attiva l'interpolazione lineare G1 indipendentemente dalla funzione G valida attualmente nel programma.

L'asse di pendolamento può

- essere l'asse di ingresso per la trasformazione dinamica,
- essere l'asse master per assi Gantry o di trascinamento,
- essere comandato
 - senza limitazione antistrappo (BRISK) oppure
 - con limitazione antistrappo (SOFT) oppure
 - con caratteristica di accelerazione spezzata (a due inclinazioni).

Punti di inversione pendolamento

Nella definizione delle posizioni di pendolamento bisogna verificare le traslazioni attuali:

- indicazioni assolute

`OSP1[Z] = valore 1`

posiz. punto di inversione = somma delle traslazioni + valore programmato

- impostazioni incrementali

`OSP1[Z] = IC(valore)`

posiz. punto di inversione = punto di inversione 1 + valore programmato

Esempio:

`N10 OSP1[Z] = 100 OSP2[Z] = 110`

·
·

`N40 OSP1[Z] = IC(3)`

Nota

`WAITP(asse):`

- se si deve pendolare con un asse geometrico, è indispensabile che esso sia abilitato al pendolamento con `WAITP`.
 - A pendolamento ultimato, con lo stesso comando l'asse viene dichiarato nuovamente come asse di posizionamento e può essere utilizzato normalmente.
-

Pendolamento con azioni sincrone al movimento e tempi di sosta, OST1/OST2

Allo scadere dei tempi di sosta, nel pendolamento si verifica il cambio di asse interno (osservabile dai nuovi percorsi residui degli assi). Durante il cambio di asse viene verificata la funzione di disattivazione. Dopo le impostazioni nel controllo numerico per le sequenze di movimento "OSCTRL", viene definita la funzione di disattivazione. **Questo comportamento temporale è influenzabile dall'override dell'avanzamento.**

In base alle circostanze è possibile eseguire ancora una passata di pendolamento prima che venga eseguita la passata di spegnifiamma o che sia stata raggiunta la posizione finale. **Potrebbe sembrare, in questo caso, che sia cambiato il comportamento di disattivazione. Tuttavia non è così.**

Impostazione dell'avanzamento, FA

Come velocità di avanzamento vale quella definita per l'asse di posizionamento. Se non viene definita alcuna velocità di avanzamento vale il valore inserito nel dato macchina.

Definizione della sequenza di movimento, OSCTRL

Le impostazioni del controllo numerico per la sequenza dei movimenti vengono attivate con le opzioni di set e reset.

`OSCTRL[asse pendolamento] = (opzione di set, opzione di reset)`

Le opzioni di set vengono definite come segue (le opzioni di reset disattivano le impostazioni):

Opzioni di reset

Queste opzioni vengono disinserite (necessarie solo se precedentemente attivate con l'opzione di set).

Opzioni di set

Queste opzioni vengono commutate. Programmando `OSE` (posizione finale) diventa implicitamente attiva l'opzione 4.

Valore dell'opzione	Significato
0	Alla disattivazione del pendolamento, arresto nel prossimo punto di inversione (preimpostazione); possibile solo resettando i valori 1 e 2
1	Alla disattivazione del pendolamento, arresto nel punto di inversione 1
2	Alla disattivazione del pendolamento, arresto nel punto di inversione 2
3	Alla disattivazione del pendolamento non raggiungere alcun punto di inversione se non è stata programmata alcuna passata di spegnifiamma
4	Dopo lo spegnifiamma raggiungere la posizione finale
8	Se il pendolamento viene interrotto con la cancellazione del percorso residuo: eseguire la passata di spegnifiamma ed eventualmente raggiungere la posizione finale
16	Se il pendolamento viene interrotto con la cancellazione del percorso residuo: come per la disattivazione, raggiungere il relativo punto di inversione
32	L'avanzamento modificato è attivo solo dopo il successivo punto di inversione
64	FA uguale a 0, FA = 0: sovrapposizione del movimento attiva FA uguale a 0, FA <> 0: sovrapposizione di velocità attiva
128	Per assi rotanti DC (percorso più breve)
256	=la passata di spegnifiamma viene eseguita come doppia passata.(Standard) 1=la passata di spegnifiamma viene eseguita come singola passata.

Per indicare più opzioni, interporre il segno +.

Esempio:

Il movimento di pendolamento per l'asse Z deve arrestarsi nel punto di inversione 1. Quindi si deve

- raggiungere la posizione finale,
- attivare immediatamente la variazione dell'avanzamento e dopo la cancellazione del percorso residuo l'asse deve essere subito fermato.

`OSCTRL [Z] = (1+4, 16+32+64)`

11.2 Pendolamento gestito da azioni sincrone

Funzione

Con questo tipo di pendolamento il movimento di incremento è consentito solo nei punti di inversione oppure entro aree di inversione definite.

In funzione delle esigenze, il movimento di pendolamento durante l'incremento può

- proseguire oppure
- può essere arrestato finché l'incremento non è stato completamente eseguito.

Programmazione

1. Definire i parametri per il pendolamento
2. Definizione delle azioni sincrone al movimento
3. Abbinare gli assi, definire l'incremento

Parametri

OSP1[asse di pendolamento]=	posizione del punto di inversione 1
OSP2[asse di pendolamento]=	posizione del punto di inversione 2
OST1[asse di pendolamento]=	tempo di attesa nel punto di inversione 1, in secondi
OST2[asse di pendolamento]=	tempo di attesa nel punto di inversione 2, in secondi
FA[asse di pendolam.] =	avanzamento dell'asse di pendolamento
OSCTRL[asse di pendolam.] =	opzioni di set e reset
OSNSC[asse di pendolamento] =	numero delle passate di spegnifiamma
OSE[asse di pendolam.] =	posizione finale
WAITP(asse di pendolam.)	consenso per l'asse di pendolamento

Abbinamento degli assi, incremento

```
OSCILL[asse di pendolamento] = (asse di incremento1, asse di
incremento2, asse di incremento3)
```

```
POSP[asse di incremento] = (pos.finale, lungh.parziale, modo)
```

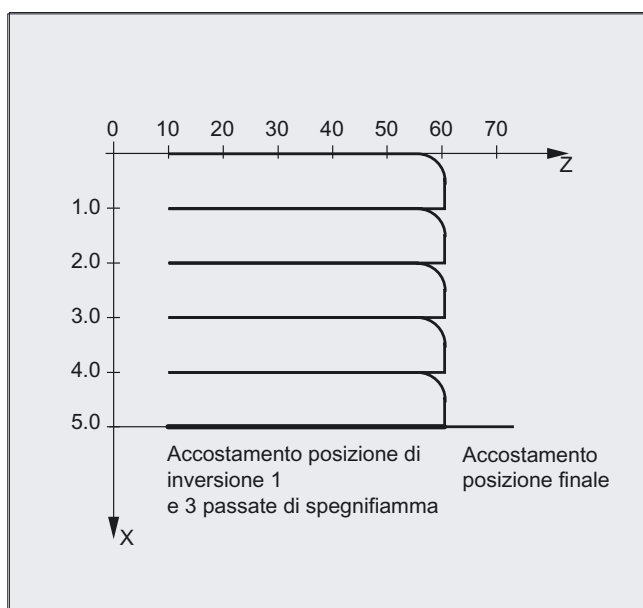
OSCILL	abbinare l'asse di incremento all'asse di pendolamento
POSP	definizione della lunghezza totale e parziale (vedi capitolo Gestione file e programmi)
Endpos	posizione finale dell'asse di incremento secondo la quale vengono eseguiti tutti gli incrementi parziali.
Lunghezza parziale	dimensione incrementi parziali sul punto di inserz./campo di inversinversione
Modo	suddivisione dell'incremento totale in incrementi parziali = due tratti restanti di uguale dimensione (preimpostazione); = tutti gli incrementi parziali sono della stessa dimensione

azioni sincrone al movimento

WHEN... .. DO	se..., esegui...
WHENEVER ... DO	sempre se..., esegui...

Esempio

Nel punto di inversione 1 non deve avvenire alcun incremento. Sul punto di inversione 2 l'incremento deve avvenire già ad una distanza $ii2$, prima del punto di inversione stesso e l'asse di pendolamento sul punto di inversione non deve attendere il completamento dell'incremento parziale. L'asse Z è quello di pendolamento e l'asse X quello di incremento.



1. Parametri per il pendolamento

DEF INT $ii2$	Definire la variabile per il campo di inversione 2
OSP1 [Z]=10 OSP2 [Z]=60	Definire i punti di inversione 1 e 2
OST1 [Z]=0 OST2 [Z]=0	Punto di inversione 1: arresto preciso fine; Punto di inversione 2: arresto preciso fine;
FA [Z]=150 FA [X]=0.5	Avanzamento per l'asse di pendolamento Z, avanzamento per l'asse di incremento X
OSCTRL [Z] = (2+8+16, 1)	Disattivare il pendolamento nel punto di inversione 2; dopo lo spegnifiamma RWL raggiungere la posizione finale; dopo RWL raggiungere la relativa posizione di inversione
OSNC [Z]=3	Passate di spegnifiamma
OSE [Z]=70	Posizione finale = 70
$ii2=2$	Definizione del campo di inversione
WAITP (Z)	Abilitazione al pendolamento per l'asse Z

2 . Azione sincrona al movimento

```

WHENEVER $AA_IM[Z] < $SA_OSCILL_REVERSE_POS2[Z] DO ->
-> $AA_OVR[X]=0 $AC_MARKER[0]=0
sempre se                                la posizione attuale dell'asse di pendolamento Z nel SCM (MKS)
minore                                  dell'inizio del settore di inversione 2,
allora                                  setta l'override assiale dell'asse di incremento X su 0%
e                                      il merker con l'indice 0 al valore 0.
WHENEVER $AA_IM[Z] >= $SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0
sempre se                                la posizione attuale dell'asse di pendolamento Z nel SCM (MKS)
maggiore o uguale                       alla posizione di inversione 2,
allora                                  setta l'override assiale dell'asse di pendolamento Z su 0%.
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[0]=1
Sempre quando                           il percorso residuo dell'incremento parziale,
è uguale                                setta il merker con l'indice 0 al valore 1.
allora
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100
Sempre quando                           il merker con l'indice 0,
è uguale                                setta l'override assiale dell'asse di incremento X su 0%; viene impedito
allora                                  così un incremento anticipato (l'asse Z non ha ancora abbandonato il
                                          campo d'inversione 2, l'asse X è però pronto per eseguire un nuovo
                                          incremento)
                                          setta l'override assiale dell'asse di pendolamento Z su 100% (viene
                                          annullata così la 2° azione sincrona).

```

-> Deve essere programmato in un blocco a sè stante

3. Avvio del pendolamento

```

OSCILL[Z] = (X)  POSP[X] = (5, 1, 1) ;Avvio dell'asse
                                          ;All'asse di pendolamento Z viene abbinato l'asse di incremento
                                          ;X.
                                          ;L'asse X deve raggiungere la posizione finale 5 a passi di 1.
M30                                     ;Fine programma

```


Descrizione

1. **Definizione dei parametri del pendolamento**
Prima del blocco di movimento contenente l'abbinamento tra asse di pendolamento ed asse di incremento, devono essere definiti i parametri di pendolamento (vedi "Pendolamento asincrono").
2. **Definizione delle azioni sincrone al movimento**
Tramite le condizioni del sincronismo avviene:
Annulla dell'incremento, finché l'asse di pendolamento non si trova all'interno di un settore di inversione
(ii1, ii2) oppure in un punto di inversione (U1, U2).
Arresto del movimento di pendolamento durante l'incremento sul **punto di inversione**.
movimento di pendolamento al termine dell'incremento parziale. Definizione dell'avvio dell'incremento parziale successivo.
3. **Abbinare l'asse di pendolamento e di incremento** e definire l'incremento totale e parziale.

Definizione dei parametri di pendolamento

Abbinamento dell'asse di pendolamento e dell'asse di incremento: OSCILL

OSCILL[asse di pendolamento] = (asse di incremento1, asse di incremento2, asse di incremento3)

Con l'istruzione OSCILL avviene l'abbinamento degli assi e lo start del pendolamento.

Ad un asse di pendolamento possono essere abbinati al massimo 3 assi di incremento.

Nota

Prima dello start di un pendolamento è necessario definire le condizioni di sincronismo per la relazione tra gli assi.

Definizione degli incrementi: POSP

POSP[asse di incremento] = (pos.fineale, lungh.parziale, modo)

Con l'istruzione POSP viene comunicato al controllo numerico:

- l'incremento totale (tramite la posizione finale)
- la grandezza dei singoli incrementi parziali sui punti di inversione o nei campi di inversione
- il comportamento dell'incremento parziale al raggiungimento della posizione finale (tramite il modo)

modo = 0 Per i due ultimi incrementi parziali avviene una suddivisione del percorso residuo in due incrementi residui di pari grandezza (preimpostazione).

modo = 1 Tutti gli incrementi parziali sono della stessa grandezza. Essi vengono calcolati dall'incremento totale.

Definizione delle azioni sincrone al movimento

Le azioni sincrone riportate in seguito, vengono utilizzate in generale per il pendolamento.

Le soluzioni riportate a titolo di esempio possono essere utilizzate come base per la programmazione di movimenti di pendolamento specifici per l'utente.

Nota

Nei singoli casi specifici le condizioni sincrone possono essere programmate anche in altro modo.

Parole chiave

WHEN ... DO ...	se..., esegui...
WHENEVER ... DO	sempre se..., esegui...

Funzioni

Con gli elementi di linguaggio, descritti dettagliatamente in seguito, è possibile realizzare le seguenti funzioni:

1. incremento sul punto di inversione.
2. incremento nel campo di inversione.
3. incremento in entrambi i punti di inversione.
4. arresto del pendolamento sul punto di inversione.
5. riavviare il movimento di pendolamento.
6. non avviare anzitempo l'incremento parziale.

Per tutte le azioni sincrone qui rappresentate come esempio, valgono le seguenti ipotesi:

- punto d'inversione 1 < punto d'inversione 2
- Z = asse di pendolamento
- X = asse di incremento

Nota

Per ulteriori chiarimenti, vedere il capitolo Azioni sincrone al movimento.

Abbinare l'asse di pendolamento e di incremento e definire l'incremento totale e parziale.**Incremento nel campo di inversione**

Il movimento di incremento deve iniziare nell'ambito di un campo di inversione prima che sia stato raggiunto il punto di inversione.

Queste azioni sincrone impediscono il movimento di incremento fino a che l'asse di pendolamento non è entrato nel settore di inversione.

Considerato quanto sopra si hanno le seguenti istruzioni:

Campo di inversione 1:

WHENEVER \$AA_IM[Z] > \$SA_OSCILL_RESERVE_POS1[Z] + ii1 DO \$AA_OVR[X] = 0
 sempre se la posizione attuale dell'asse di pendolamento nel SCM (MKS)
 è maggiore dell'inizio del settore di inversione 1,
 allora setta l'override assiale dell'asse di incremento a 0%.

Campo di inversione 2:

WHENEVER \$AA_IM[Z] < \$SA_OSCILL_RESERVE_POS2[Z] + ii2 DO \$AA_OVR[X] = 0
 sempre se la posizione attuale dell'asse di pendolamento nel SCM (MKS)
 minore dell'inizio del settore di inversione 2,
 allora setta l'override assiale dell'asse di incremento a 0%.

Incremento sul punto di inversione

Finché l'asse di pendolamento non raggiunge il punto di inversione, non avviene nessun movimento dell'asse di incremento.

Considerato quanto sopra si hanno le seguenti istruzioni:

Campo di inversione 1:

WHENEVER \$AA_IM[Z] <> \$SA_OSCILL_RESERVE_POS1[Z] DO \$AA_OVR[X] = 0 →
 → \$AA_OVR[Z] = 100
 sempre se la posizione attuale dell'asse di pendolamento Z nel SCM (MKS)
 maggiore o minore della posizione del punto di inversione 1,
 allora setta l'override assiale dell'asse di incremento X su 0%
 e setta l'override assiale dell'asse di pendolamento Z al 100%

Campo di inversione 2:

Per il punto di inversione 2:

WHENEVER \$AA_IM[Z] <> \$SA_OSCILL_RESERVE_POS2[Z] DO \$AA_OVR[X] = 0 →
 → \$AA_OVR[Z] = 100
 sempre se la posizione attuale dell'asse di pendolamento Z nel SCM (MKS)
 maggiore o minore della posizione del punto di inversione 2,
 allora setta l'override assiale dell'asse di incremento X su 0%
 e setta l'override assiale dell'asse di pendolamento Z al 100%

Arresto del pendolamento sul punto di inversione.

L'asse di pendolamento viene arrestato sul punto di inversione e contemporaneamente inizia il movimento di incremento. Il movimento di pendolamento prosegue quando l'incremento è stato eseguito completamente.

Questa azione sincrona può essere utilizzata anche per avviare il movimento di incremento nel caso, in precedenza, esso sia stato interrotto da un'azione sincrona ancora attiva.

Considerato quanto sopra si hanno le seguenti istruzioni:

Campo di inversione 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X] = 0 →  
→ $AA_OVR[Z] = 100
```

sempre se la posizione attuale dell'asse di pendolamento nel SCM (MKS)
è uguale alla posizione di inversione 1,
allora setta l'override assiale dell'asse di pendolamento a 0%
e setta l'override assiale dell'asse di incremento al 100%.

Campo di inversione 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X] = 0 →  
→ $AA_OVR[Z] = 100
```

sempre se la posizione attuale dell'asse di pendolamento nel SCM (MKS)
è uguale alla posizione di inversione 2,
allora setta l'override assiale dell'asse di pendolamento a 0%
e setta l'override assiale dell'asse di incremento al 100%.

Valutazione Online del punto di inversione

Se nella parte destra del confronto è presente una variabile dell'elaborazione principale contraddistinta con \$\$, entrambe le variabili vengono interpretate e confrontate reciprocamente in modo continuo nell'ambito del clock IPO.

Nota

Per ulteriori informazioni consultare il capitolo "Azioni sincrone al movimento".

Riavviare il movimento di pendolamento

Questa azione sincrona viene utilizzata per proseguire il movimento dell'asse di pendolamento una volta ultimato il movimento dell'incremento parziale.

Considerato quanto sopra si hanno le seguenti istruzioni:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z] = 100
```

sempre se il percorso residuo per l'incr. parziale dell'asse d'incremento X nel
è uguale SCP (WKS) è a zero,
allora setta l'override assiale dell'asse di pendolamento al 100%

Successivo incremento parziale

Ad incremento avvenuto bisogna impedire una partenza anticipata del successivo incremento parziale.

A questo scopo viene utilizzato un merker specifico per canale (\$AC_MARKER[indice]) che viene settato al termine dell'incremento parziale (percorso residuo parziale ≡ 0) e resettato all'abbandono del campo di inversione. Quindi, con l'azione sincrona, viene impedito il successivo movimento di incremento.

Supposto quanto sopra, prendendo come esempio il punto di inversione 1, si hanno le seguenti istruzioni:

1. Impostazione label (marker):

```
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[1]=1
```

sempre se il percorso residuo per l'incr. parziale dell'asse d'incremento X nel
è uguale SCP (WKS) è a zero,
allora setta il merker con indice 1 a 1.

2. Cancellazione label (marker)

```
WHENEVER $AA_IM[Z] <> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1] = 0
```

sempre se la posizione attuale dell'asse di pendolamento Z nel SCM (MKS)
maggiore o minore della posizione del punto di inversione 1,
allora setta il merker con indice 1 a 0.

3. Impedire l'incremento

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

sempre se il merker con indice 1 è,
è uguale
allora setta l'override assiale dell'asse di incremento X a 0%.

Punzonatura e roditura

12.1 Attivazione, disattivazione

12.1.1 Punzonatura e roditura ON oppure OFF (SPOF, SON, PON, SONS, PONS, PDELAYON/OF)

Funzione

Attivazione/disattivazione della punzonatura e roditura, PON/SON

Con PON e SON vengono attivate le funzioni di punzonatura o roditura. SPOF disattiva tutte le funzioni specifiche di punzonatura e roditura. I comandi modali PON e SON si escludono a vicenda, ossia PON disattiva SON e viceversa.

Punzonatura e roditura con premilastra, PONS/SONS

Le funzioni PONS e SONS attivano rispettivamente le funzioni di punzonatura e roditura.

A differenza di SON/PON – comando del colpo a livello di interpolazione – il controllo tecnico dei segnali dell'esecuzione del colpo avviene a livello del clock Servo. Di conseguenza è possibile lavorare con più elevate frequenze dei colpi e quindi con prestazioni di punzonatura superiori.

Durante l'interpretazione dei segnali vengono bloccate tutte quelle funzioni che possono portare ad una variazione della posizione degli assi di roditura o di punzonatura.

Esempio: movimenti con il volantino, variazioni di frame da PLC, funzioni di misura.

Punzonatura con ritardo, PDELAYON/PDELAYOF

PDELAYON provoca un'esecuzione ritardata del colpo. Il comando modale ha una funzione preparatoria e viene programmato normalmente prima di PON. Dopo PDELAYOF continua regolarmente la punzonatura.

Programmazione

PONS G... X... Y... Z...

oppure

SON G... X... Y... Z...

oppure

SONS G... X... Y... Z...

oppure

SPOF

oppure

PDELAYON

oppure

PDELAYOF

oppure

PUNCHACC (S_{min} , A_{min} , S_{max} , A_{max})

Parametri

PON	punzonatura ON
PONS50	Punzonatura con premilastra ON
SON	roditura ON
SONS50	Roditura con premilastra ON
SPOF	Punzonatura, roditura OFF
PDELAYON	Punzonatura con ritardo ON
PDELAYOF	Punzonatura con ritardo Off
PUNCHACC	Accelerazione dipendente dal percorso PUNCHACC (S_{min} , A_{min} , S_{max} , A_{max})
" S_{min} "	Min.distanza fori
" S_{max} "	Max.distanza fori
" A_{min} "	L'accelerazione iniziale A_{min} può essere maggiore di A_{max}
" A_{max} "	L'accelerazione finale A_{max} può essere inferiore di A_{min}

Utilizzo delle istruzioni M

Utilizzando la tecnica delle macro, invece delle istruzioni di linguaggio si possono anche utilizzare le istruzioni M:

DEFINE M25 AS PON	punzonatura ON
DEFINE M125 AS PONS	Punzonatura con premilastra ON
DEFINE M22 AS SON	roditura ON
DEFINE M122 AS SONS	Roditura con premilastra ON
DEFINE M26 AS PDELAYON	Punzonatura con ritardo ON
DEFINE M20 AS SPOF	Punzonatura, roditura OFF
DEFINE M23 AS SPOF	Punzonatura, roditura OFF

Punzonatura e roditura con premilastra, PONS/SONS

La punzonatura e la roditura con premilastra non sono possibili in più canali contemporaneamente. PONS oppure SONS possono essere attivati solo in un canale.

Se PONS oppure SONS vengono attivati in più di un canale contemporaneamente, viene emesso l'allarme 2200 "Canale %1 punzonatura/roditura rapida in più canali non possibile".

Altrimenti PONS e SONS funzionano come PON e SON.

Accelerazione in funzione del percorso, PUNCHACC

L'istruzione PUNCHACC PUNCHACC(S_{min} , A_{min} , S_{max} , A_{max}) definisce una caratteristica di accelerazione che, in base alla distanza dei fori (S), stabilisce delle diverse accelerazioni (A).

Esempio di PUNCHACC(2, 50, 10, 100):

Distanze tra i fori inferiori a 2mm:

Il movimento avviene con un'accelerazione pari al 50% dell'accelerazione max.

Distanze tra i fori da 2mm a 10mm:

L'accelerazione aumenta in modo proporzionale alle distanze fino al 100%.

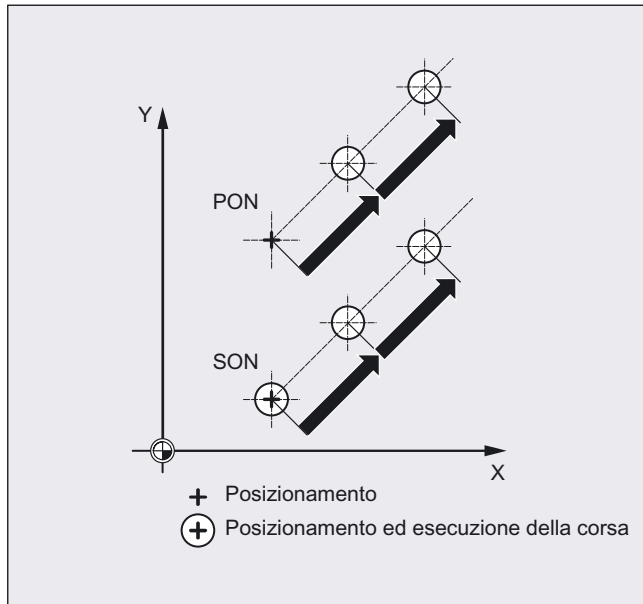
Distanze tra i fori superiori a 10mm:

Movimento con accelerazione pari al 100%.

Esecuzione del primo colpo

L'esecuzione del primo colpo dopo l'attivazione della funzione avviene, per roditura e punzonatura, in tempi diversi:

- PON/PONS:
 - Tutti i colpi – anche quello del primo blocco dopo l'attivazione – avvengono alla fine del blocco.
- SON/SONS:
 - Il primo colpo dopo l'attivazione della roditura avviene già all'inizio del blocco.
 - Tutti i colpi successivi vengono eseguiti a fine blocco.



Punzonatura e roditura sul posto

L'esecuzione del colpo avviene esclusivamente se nel blocco è presente un'istruzione di movimento degli assi di punzonatura o roditura (assi del piano attivo).

Per eseguire un colpo nella stessa posizione, si deve programmare un asse di punzonatura/roditura con spostamento = 0.

Nota

Lavorazione con utensili orientabili

Per impiegare utensili orientabili tangenzialmente alla traiettoria, si deve utilizzare il controllo tangenziale.

12.2 Preparazione automatica del percorso

Funzione

Suddivisione in tratti di percorso

Con l'attivazione della punzonatura, o della roditura, sia SPP che SPN provocano una suddivisione del percorso complessivo programmato per gli assi di contornitura, in un numero di tratti di percorso della stessa lunghezza (suddivisione in tratti equidistanti). Internamente, ad ogni tratto di percorso corrisponde un blocco.

Numero dei colpi

Nella punzonatura il primo colpo avviene in corrispondenza del punto finale del primo tratto di percorso, nel caso della roditura invece, in corrispondenza del punto iniziale. Considerando il tratto totale ne deriva che:

Punzonatura: Numero dei colpi = Numero dei tratti di percorso

Roditura: Numero dei colpi = Numero dei tratti di percorso + 1

Funzioni ausiliarie

Le funzioni ausiliarie vengono eseguite nel primo blocco generato.

Programmazione

SPP=

oppure

SPN=

Parametri

SPP

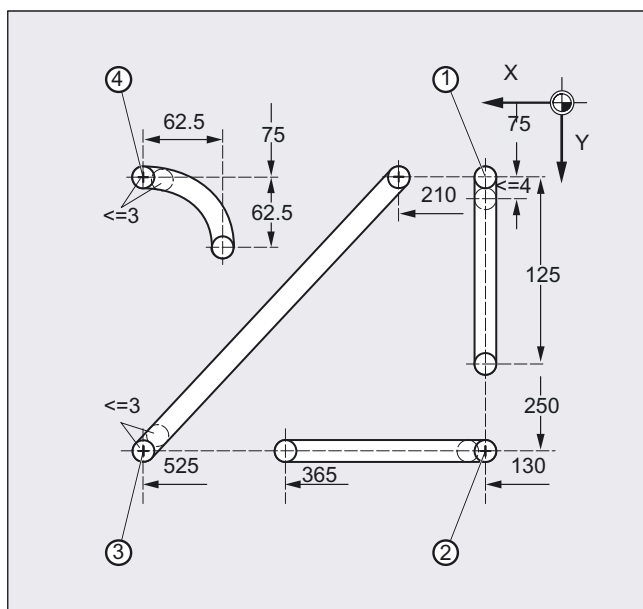
Lunghezza del tratto di percorso (intervallo tra i colpi); validità modale

SPN

Numero dei tratti di percorso per blocco; validità blocco-blocco

Esempio 1

I tratti di roditura programmati devono essere suddivisi automaticamente in tratti di percorso uguali.



N100 G90 X130 Y75 F60 SPOF

;Posizionamento al punto di inizio 1

N110 G91 Y125 SPP=4 SON

;Roditura ON; lunghezza max. dei tratti di
percorso
;per la suddivisione automatica del percorso:
;4 mm

N120 G90 Y250 SPOF

;Roditura OFF; posizionamento
;al punto di inizio 2

N130 X365 SON

;Roditura ON; lunghezza max. dei tratti di
percorso
;per la suddivisione automatica del percorso:
;4 mm

N140 X525 SPOF

;Roditura OFF; posizionamento
;al punto di inizio 3

N150 X210 Y75 SPP=3 SON

;Roditura ON; lunghezza max. dei tratti di
percorso
;per la suddivisione automatica del percorso:
;3 mm

N160 X525 SPOF

;Roditura OFF; posizionamento
;al punto di inizio 4

N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON

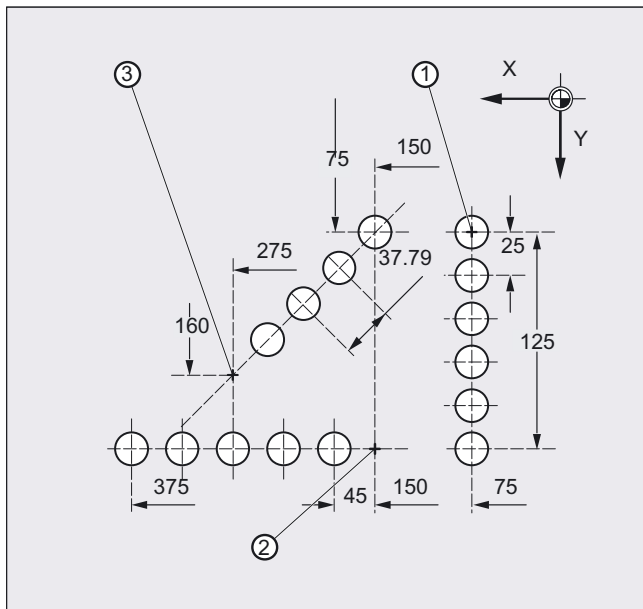
;Roditura ON; lunghezza max. dei tratti di
percorso
;per la suddivisione automatica del percorso:
;3 mm

N180 G00 G90 Y300 SPOF

;Roditura OFF

Esempio 2

Deve essere eseguita una suddivisione automatica del percorso per ogni singola serie di fori. Per la suddivisione viene impostata di volta in volta la lunghezza max. dei tratti di percorso (valore di SPP).



N100 G90 X75 Y75 F60 PON

;Posizionamento al punto di inizio 1;
;Punzonatura ON; punzonatura foro singolo

N110 G91 Y125 SPP=25

;Lunghezza max. dei tratti di percorso per la
;suddivisione automatica del percorso: 25 mm

N120 G90 X150 SPOF

;Punzonatura OFF; posizionamento
;al punto di inizio 2

N130 X375 SPP=45 PON

;Punzonatura ON; lunghezza max. dei tratti di
;percorso
;per la suddivisione automatica del percorso:
;45 mm

N140 X275 Y160 SPOF

;Punzonatura OFF; posizionamento
;al punto di inizio 3

N150 X150 Y75 SPP=40 PON

;Punzonatura ON; invece del tratto di percorso
;programmato della lunghezza di 40 mm, viene
;utilizzata la
;lunghezza calcolata di 37,79 mm.

N160 G00 Y300 SPOF

;Punzonatura OFF; posizionamento

12.2.1 Suddivisione del percorso per assi di contornitura

Lunghezza del tratto di percorso SPP

Con **SPP** si imposta la distanza massima tra i colpi e quindi la lunghezza massima dei tratti di percorso nei quali deve essere suddiviso il tratto totale. La disattivazione del comando avviene con **SPOF** oppure **SPP=0**.

Esempio:

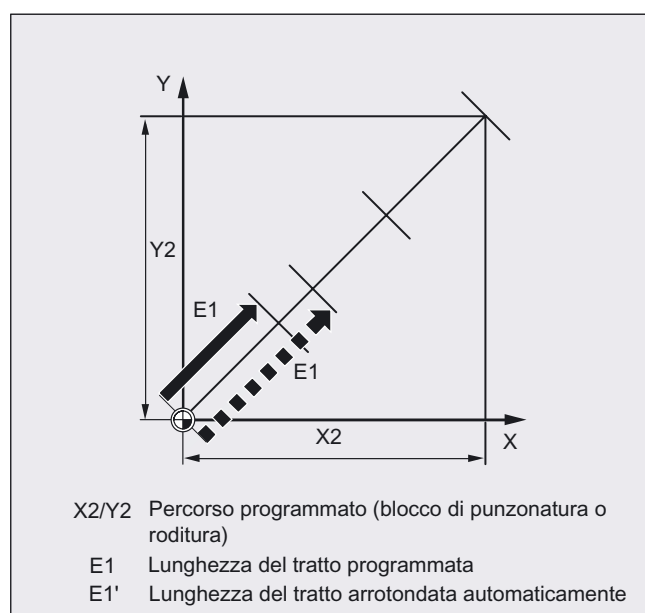
N10 SON X0 Y0

N20 **SPP=2** X10

Il tratto totale di 10 mm viene suddiviso in 5 tratti di percorso, ognuno da 2 mm (**SPP=2**).

Nota

La suddivisione del percorso con **SPP** avviene sempre in modo equidistante: tutti i tratti di percorso hanno la stessa lunghezza. Ne consegue che la lunghezza del tratto di percorso (valore di **SPP**), è valida solamente se il quoziente derivante dal tratto totale e da **SPP** è un numero intero. Se questo non si verifica, la lunghezza del tratto di percorso viene ridotta internamente per far sì che il quoziente risulti intero.



Esempio:

N10 G1 G91 SON X10 Y10

N20 **SPP=3.5** X15 Y15

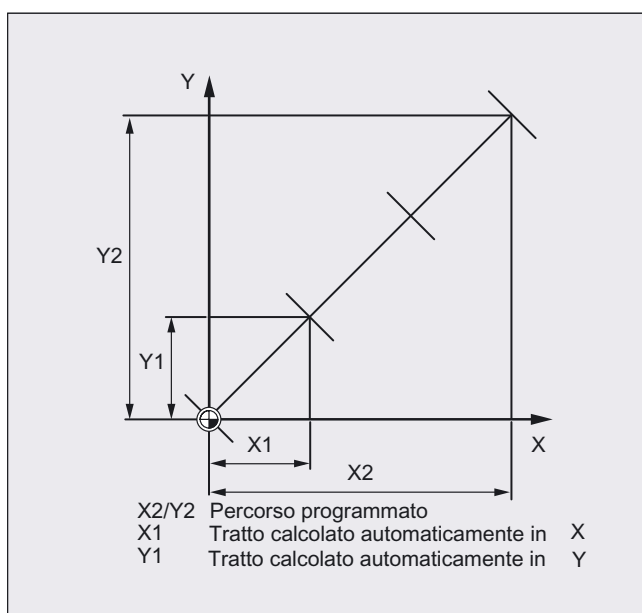
Con un tratto totale di 15 mm e una lunghezza dei tratti di percorso di 3,5 mm, si ricava un quoziente non intero (4.28). Di conseguenza, il valore di **SPP** verrà ridotto fino ad ottenere un quoziente intero. In questo caso risulta una lunghezza dei tratti di percorso di 3 mm.

Numero dei tratti di percorso SPN

Con **SPN** si definisce il numero dei tratti di percorso che devono essere ricavati dal tratto totale. La lunghezza dei tratti di percorso viene calcolata automaticamente. Poichè **SPN** ha un'azione blocco a blocco, deve prima essere attivata la punzonatura o la roditura con **PON** oppure **SON**.

SPP e SPN nello stesso blocco

Programmando nello stesso blocco la lunghezza dei tratti di percorso (**SPP**) ed il numero dei tratti di percorso (**SPN**), nel blocco in questione ha validità **SPN**, nei blocchi successivi **SPP**. Se **SPP** è già stato programmato prima di **SPN**, dopo il blocco con **SPN** esso diventa nuovamente attivo.

**Nota**

Se nel controllo sono presenti le tecnologie di punzonatura/roditura, la programmazione della suddivisione automatica del percorso è attivabile con **SPN** o **SPP** anche indipendentemente da queste tecnologie.

12.2.2 Suddivisione del percorso per singoli assi

Se oltre ad assi di contornitura, vengono definiti come assi di punzonatura/roditura anche assi singoli, questi ultimi possono sottostare alla suddivisione automatica del percorso.

Comportamento degli assi singoli con SPP

La lunghezza programmata dei tratti di percorso (SPP) si riferisce sempre agli assi di contornitura. Perciò il valore SPP è ignorato in un blocco nel quale, oltre al movimento del singolo asse e al valore SPP, non è programmato alcun asse di contornitura.

Se in un blocco vengono programmati assi singoli e di contornitura, il comportamento dell'asse singolo dipende dall'impostazione del relativo dato macchina.

1. Impostazione standard

Il percorso dell'asse singolo viene suddiviso in modo uniforme tra i blocchi intermedi generati tramite SPP.

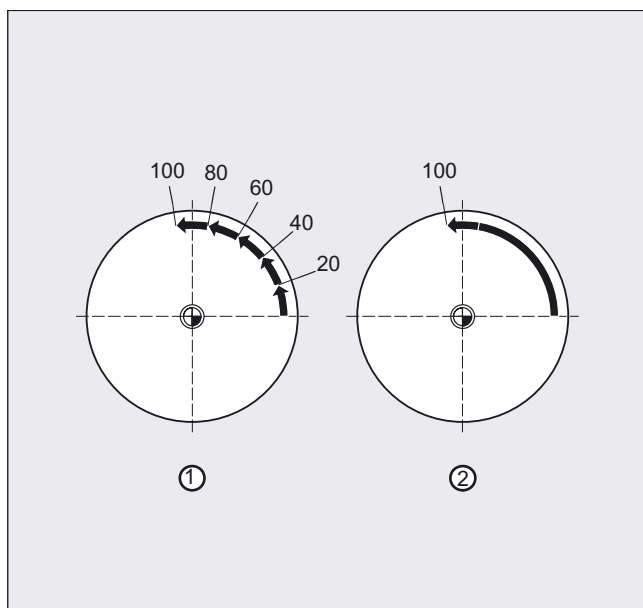
Esempio:

N10 G1 SON X10 A0

N20 SPP=3 X25 A100

Conseguentemente ai tratti di percorso di 3 mm e considerando il tratto totale dell'asse X (asse di contornitura) di 15 mm, vengono generati 5 blocchi.

L'asse A di conseguenza ruota di 20° in ogni blocco.



1. Asse singolo senza suddivisione del percorso

L'asse singolo compie il suo percorso totale nel primo dei blocchi generati.

2. Suddivisione del percorso diversa

Il comportamento dell'asse singolo dipende dalla interpolazione degli assi di contornitura:

- Interpolazione circolare: suddivisione del percorso
- Interpolazione lineare: nessuna suddivisione

Comportamento con SPN

Il numero dei tratti di percorso programmati ha validità anche se non viene programmato contemporaneamente un asse di contornitura.

Premessa: l'asse singolo è stato definito come asse di punzonatura/roditura.

Ulteriori funzioni

13.1 Funzioni per gli assi (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING)

Funzione

AXNAME viene utilizzato ad esempio nella stesura di cicli generici quando il nome degli assi non è noto (vedere anche il cap. "Funzioni stringhe").

SPI viene utilizzato quando si usano funzioni di assi per un mandrino, ad es. mandrino sincrono.

ISAXIS viene utilizzato per cicli a validità generale per garantire che un determinato asse geometrico sia presente e che quindi un successivo richiamo con \$P_AXNX non venga interrotto con un messaggio di errore.

Programmazione

```
AXNAME(asse radiale)
oppure
AX[AXNAME(stringa)]
oppure
SPI(n)
oppure
AXTOSPI(X) o AXTOSPI(Y) o AXTOSPI(Z)
oppure
AXSTRING( SPI(n) )
oppure
ISAXIS(numero asse di geometria)
```

Parametri

AXNOME	Converte una stringa in ingresso in indicatore dell'asse; la stringa in ingresso deve contenere un nome valido di asse.
AX	Indicatore assi variabile
SPI	Converte il numero del mandrino in identificatore assi; il parametro di trasmissione deve contenere un numero di mandrino valido.
n	Numero del mandrino
AXTOSPI	Trasforma un identificatore asse in un indice di mandrino del tipo Integer. AXTOSPI corrisponde alla funzione di conversione in SPI.
X, Y, Z	Identificatore asse del tipo AXIS come variabile o costante
AXSTRING	Viene emessa la stringa con il numero di mandrino assegnato.
ISAXIS	Verifica se l'asse geometrico indicato esiste.

Ampliamenti SPI

La funzione asse `SPI (n)` è utilizzabile anche per leggere e scrivere componenti di frame. In questo modo si possono ad es. scrivere dei frame con la sintassi `$P_PFRAME [SPI (1) , TR]=2.22`. Con l'ulteriore programmazione della posizione dell'asse tramite l'indirizzo `AX [SPI (1)] = <Posizione asse>`, può essere comandato un asse.

Ampliamento AXTOSPI

Con AXTOSPI è possibile convertire un identificatore asse in un numero di mandrino. Se non riesce la conversione di un identificatore asse in un numero di mandrino, viene emesso un messaggio di allarme.

Eliminazione degli errori con AXSTRING[SPI(n)]

Nella programmazione di `AXSTRING[ SPI (n) ]` l'indice asse al quale è assegnato il mandrino non viene più emesso come numero del mandrino, ma come String `"Sn"`.

Esempio:

`AXSTRING[ SPI (2) ]` fornisce la stringa `"S2"`

Esempio

Si vuole muovere l'asse definito come asse radiale.

<code>OVRA[AXNAME("Asse radiale")]=10</code>	<code>;Asse radiale</code>
<code>AX[AXNAME("Asse radiale")]=50.2</code>	<code>;Posizione finale per l'asse radiale</code>
<code>OVRA[SPI(1)]=70</code>	<code>;Override per il mandrino 1</code>
<code>IF ISAXIS(1)==FALSE GOTOF CONTINUA</code>	<code>;L'ascissa è presente?</code>
<code>AX[\$P_AXN1]=100</code>	<code>;Muovere sull'ascissa</code>
<code>CONTINUA:</code>	

13.2 Verifica del set di istruzioni del linguaggio NC presente (STRINGIS)

Funzione

Il set di istruzioni del linguaggio NC generato con un SINUMERIK 840D sl, comprese le definizioni attive di GUD/macro, e i programmi ciclici installati ed attivi dei cicli possono essere verificati con il comando STRINGIS per quanto riguarda la loro disponibilità effettiva e le loro proprietà specifiche di programma. In questo modo è possibile, ad es., già all'inizio di un'interpretazione di programma appurare l'efficacia delle funzioni non attivate.

I valori di ritorno vengono emessi in modo indicizzato tramite la superficie operativa HMI e contengono informazioni di base nonché informazioni di dettaglio indicizzate.

Programmazione

STRGINGIS (STRING name) = valore di ritorno indicizzato

Nella configurazione attuale, (STRING name) viene fondamentalmente trattato come segue:

con **000** viene identificato come sconosciuto,

con **100** viene riconosciuto come comando NC, che tuttavia non è programmabile.

Tutti i comandi NC, attivi come funzione od opzione, vengono rilevati con

2xx. Le relative informazioni di dettaglio sono contenute nei campi dei valori.

Parametri

Costruttore della macchina

Come e con quali comandi NC debba avvenire il movimento viene stabilito dal costruttore della macchina tramite un dato macchina.

Se vengono programmati comandi NC la cui funzione non è attiva o è sconosciuta nell'ambito attuale, viene generato un messaggio di allarme. Prestare particolare attenzione in questo caso alle indicazioni del costruttore della macchina

STRINGIS	Verifica l'insieme del linguaggio NC presente; per i nomi di ciclo NC, le variabili utente, le macro e i nomi di label specificamente riferiti a questo comando verifica se esistono, sono validi, definiti o attivi. Il comando NC STRINGIS è una variabile del tipo Integer.
Speciale per STRINGIS	Nomi di ciclo NC (di un ciclo attivo) Variabili GUD Variabili LUD Macro Nomi di label
STRING name	Identificatore della variabile del set di istruzioni di linguaggio NC da verificare e parametri di trasferimento di valori rilevati del tipo STRING.

Il comando NC ISVAR rappresenta un sottoinsieme del comando STRINGIS e può continuare ad essere utilizzato per determinate verifiche.

Sul comportamento di un comando STING, vedere il capitolo "Funzioni stringhe".

Set di istruzioni del linguaggio NC

Per SINUMERIK powerline continuano ad essere validi tutti i comandi NC disponibili ed anche particolarmente quelli non richiesti e attivi. Il set di istruzioni da verificare per SINUMERIK solution line dipende dal dato macchina preconfigurato e comprende tutte le opzioni conosciute oppure specificamente solo le opzioni abilitate o le funzioni attive nel set di istruzioni NC attuale.

Set di istruzioni di linguaggio NC	Al set di istruzioni di linguaggio NC appartengono: Codici G di tutti i gruppi di codice G esistenti, come ad es. G0, G1, G2, INVCW, POLY, ROT, KONT, SOFT, CUT2D, CDON, RMB, SPATH Indirizzi DIN o NC come ad es. ADIS, RNDM, SPN, SR, MEAS Funzioni di linguaggio NC come ad es. i sottoprogrammi predefiniti TANG(FAssel..n, LAssel..n, fattore di accoppiamento). Procedure di linguaggio NC (procedure predefinite con valore di ritorno), come ad es. l'esecuzione del sottoprogramma con trasferimento dei parametri GETMDACT. Procedure di linguaggio NC (procedure predefinite senza valore di ritorno), come ad es. la disattivazione dell'inibizione blocco singolo SBLOF. Parole chiave NC come ad es. ACN, ACP, AP, RP, DEFINE, SETMS Dati macchina \$MN generali, \$MA assiali, \$MC specifici di canale nonché tutti i dati setting \$S... e i dati opzionali \$O... . Variabili di sistema NC nel partprogram e nelle azioni sincrone come pure parametri di calcolo NC R.

Valori di ritorno

Informazioni di base	Il valore di ritorno è indicizzato. La ripartizione avviene così: in y le informazioni di base contenute, in x le informazioni dettagliate presenti.
Indicizzazione:	Risultato della verifica , se nella configurazione attuale:
000	STRING name non è noto all'NCK.
100	STRING name è un comando NC ma non è programmabile , cioè questa funzione è inattiva.
2xx	STRING name è un comando NC programmabile , cioè questa funzione è attiva.
y00	Non è possibile un'assegnazione.
y01 ... y11	I campi dei valori per le informazioni di dettaglio presenti sono conosciuti.
400	Per gli indirizzi NC che non presentano xx=01 o xx=10 e non sono codici G o parametri di calcolo R, vedere la nota (1).

Nota

Se la verifica con **STRINGIS** non evidenzia un'altra indicizzazione, il comando NC è considerato programmabile e vale l'indicizzazione 2xx.

Campi di valore 2xx delle informazioni dettagliate

Informazioni dettagliate	Significato del risultato della verifica:
200	Non è possibile alcuna interpretazione
201	È definito un indirizzo DIN o NC, cioè se dal nome sono stati riconosciuti gli identificatori indirizzo; vedere nota (1)
202	Sono stati riconosciuti i codici G dai gruppi esistenti di codice G.
203	Sono presenti le funzioni di linguaggio NC con valore di ritorno e trasferimento dei parametri.
204	Sono presenti le procedure di linguaggio NC senza valore di ritorno e con trasferimento dei parametri.
205	Sono presenti le parole chiave NC.
206	Sono presenti dati macchina generali, assiali o specifici di canale (\$M . . .), dati setting (\$S . . .) o dati opzionali (\$O . . .).
207	Sono presenti variabili utente come ad es. le variabili di sistema NC che iniziano con \$. . . o parametri di calcolo che iniziano con R.
208	Sono stati caricati i nomi di ciclo nell'NCK e anche i programmi ciclici sono attivati; vedere nota (2).
209	È stato riconosciuto dalle variabili utente globali (variabili GUD) il nome definito ed è stata trovata la variabile GUD attivata.
210	Sono stati trovati i nomi delle macro con i nomi definiti nei file di definizione delle macro e le macro attivate; vedere nota (3).
211	Di variabili utente locali (variabili LUD), il cui nome è contenuto in un programma attuale.

Nota**Note sui singoli valori di ritorno**

(1) Come indirizzi DIN vengono riconosciuti gli indirizzi a standard fisso. Per gli indirizzi NC con identificatori impostabili valgono le seguenti definizioni relative agli assi di geometria:

A, B, C per gli assi rotanti specificati, E è riservato per le estensioni,
I, J, K, Q, U, V, W, X, Y, Z per gli assi lineari specificati.

Gli identificatori asse possono essere programmati con ampliamento dell'indirizzo e scritti per la verifica, ad es. 201 = STRINGIS("A1").

I seguenti indirizzi non possono essere scritti con ampliamento dell'indirizzo per la verifica e forniscono sempre il valore fisso 400.

Esempio: 400 = STRINGIS("D") o l'indicazione di un ampliamento dell'indirizzo con
0 = STRINGIS("M02") determina 400 = STRINGIS("M").

(2) Non è possibile verificare con il comando STRINGIS i nomi dei parametri del ciclo.

(3) Gli identificatori indirizzo NC definiti come macro G, H, L, M vengono identificati come macro.

Indirizzi NC validi senza ampliamento dell'indirizzo con il valore fisso 400

Sono validi gli indirizzi NC D, F, G, H, R e L, M, N, O, P, S, T. Significato:

400	D come correzione utensile, numero tagliente (funzione D)
	F come avanzamento (funzione F)
	G è definito come codice G (qui non è la condizione di percorso)
	H indica funzione ausiliaria (funzione H)
	R è definito come parametro di sistema
	L indica chiamata a un sottoprogramma, M funzione aggiuntiva, N blocco secondario,
	O è libero per estensioni,
	P indica il numero delle esecuzioni del programma,
	S numero di giri del mandrino (funzione S),
	T numero utensile (funzione T).

Esempio di funzione ausiliaria T programmabile

T è definito come funzione ausiliaria e sempre programmabile.	
400 = STRINGIS("T")	;valore di ritorno senza ampliamento dell'indirizzo
0 = STRINGIS("T3")	;valore di ritorno con ampliamento dell'indirizzo

Esempi di altre verifiche del set di istruzioni di linguaggio NC programmabile 2xx

X è definito come asse	;l'asse è un asse lineare X
201 = STRINGIS("X")	;valore di ritorno asse lineare X
201 = STRINGIS("X1")	;valore di ritorno asse lineare X1
A2 è un indirizzo NC con estensione	;indirizzo NC A2 con estensione
201 = STRINGIS("A")	;valore di ritorno per indirizzo NC A
201 = STRINGIS("A2")	;con indirizzo NC esteso A2
INVCW è un codice G definito	;INVCW è codice G
	;interpolazione degli evolventi in senso
	;orario.
202 = STRINGIS("INVCW")	;valore di ritorno del codice G nominato
GETMDACT è una funzione di linguaggio NC	;la funzione di linguaggio NC GETMDACT è
	;presente.
203 = STRINGIS("GETMDACT")	;GETMDACT è una funzione di linguaggio
	;NC
DEFINE è una parola chiave NC	;esiste la parola chiave DEFINE per
	;l'identificazione delle macro.
205 = STRINGIS("DEFINE")	;DEFINE è presente come parola chiave
\$MC_GCODES_RESET_VALUES è un dato macchina	;il dato macchina
specifico di canale	;\$MC_GCODE_RESET_VALUES esiste.
206 = STRINGIS("\$MC_GCODE_RESET_VALUES")	;\$MC_GCODE_RESET_VALUES è
	;stato riconosciuto come dato macchina
\$TC_DP3 è una variabile di sistema per un	;esiste la variabile di sistema NC \$TC_DP3
componente di lunghezza utensile	;per
	;componente di lunghezza utensile.
207 = STRINGIS("\$TC_DP3")	;\$TC_DP3 riconosciuto come variabile di
	;sistema.

13.3 Richiamo della funzione ISVAR () e lettura dei dati macchina indice dell'array

\$TC_TP4 è una variabile di sistema per una dimensione utensile 207 = STRINGIS("\$TC_TP4") \$TC_MPP4 è una variabile di sistema per lo stato del posto magazzino 207 = STRINGIS("\$TC_MPP4") 0 = STRINGIS("\$TC_MPP4") MACHINERY_NAME è definito come variabile GUD 209 = STRINGIS("MACHINERY_NAME") LONGMACRO è definito come macro 210 = STRINGIS("LONGMACRO") MYVAR è definito come variabile LUD 211 = STRINGIS("MYVAR") X, Y, Z è un comando sconosciuto a NC 0 = STRINGIS("XYZ")	;esiste la variabile di sistema NC \$TC_TP4 ;per la dimensione utensile. ;\$TC_TP4 riconosciuto come variabile di ;sistema. ;verifica della gestione magazzino ;la gestione magazzino è attiva ;la gestione magazzino non è ;disponibile (4) ;la variabile utente globale è definita come ;MACHINERY_NAME. ;MACHINERY_NAME trovato come GUD ;il nome della macro è LONGMACRO ;macro identificata come LONGMACRO ;la variabile utente locale è stata ;denominata MYVAR ;la variabile LUD è contenuta come nome ;MYVAR nel programma attuale ;X,Y,Z è un comando sconosciuto ;e non è GUD / macro / nome di ciclo ;STRING name X, Y, Z è sconosciuto

(4) La proprietà vale specialmente per i parametri di sistema della gestione magazzino se la funzione non è attiva; in questo caso, STRINGIS fornisce sempre, indipendentemente dal valore impostato del dato macchina per la configurazione del set di istruzioni del linguaggio NC, il valore del risultato = 0.

13.3 Richiamo della funzione ISVAR () e lettura dei dati macchina indice dell'array

Funzione

L'istruzione ISVAR è una funzione nell'ambito del linguaggio NC con un:

- Valore della funzione del tipo BOOL
- Parametro di trasferimento del tipo STRING

L'istruzione ISVAR restituisce TRUE quando il parametro di trasferimento contiene una variabile nota all'NC (dato macchina, dato setting, variabile di sistema, variabili generiche ad esempio GUD).

Programmazione

```
ISVAR(Identificatore variabile)
oppure
ISVAR (identificatore, [valore, valore])
```

Parametri

Identificatore di variabile	Il parametro di trasferimento del tipo String può essere senza dimensioni, a una dimensione o a due dimensioni.
identificatore	Identificativo con una delle variabili note all'NC con o senza indice di array come dato macchina, dato setting, variabile di sistema o variabile generica. Ampliamento: per i dati macchina generici e specifici per canale, il primo elemento dell'array viene letto anche se manca l'indice
Valore	Valore della funzione del tipo BOOL

Verifiche

In relazione al parametro di trasferimento vengono eseguite queste verifiche:

- è presente l'identificatore
- si tratta di un campo mono o bidimensionale
- è consentito un indice Array

La funzione verrà confermata con TRUE solo se tutti questi controlli hanno avuto esito positivo. Nel caso in cui una sola condizione non venga soddisfatta oppure ci sia un errore di sintassi, la funzione verrà confermata con FALSE. Le variabili assiali vengono accettate come indici per i nomi degli assi ma non è prevista un'ulteriore verifica.

Ampliamento: lettura dell'array dei dati macchina e dati setting senza indice.

Se manca l'indice dei dati macchina **generici e specifici per canale** l'allarme 12400 "Canale % 1 Blocco % 2 Campo % 3 Elemento non esistente" **non viene più** emesso.

Inoltre per i dati macchina **specifici per canale** deve essere programmato **almeno l'indice dell'asse**. In caso contrario viene emesso l'allarme 12400.

Esempio di richiamo delle funzione ISVAR

DEF INT VAR1	
DEF BOOL IS_VAR=FALSE	;Il parametro di trasferimento è una variabile generica
N10 IS_VAR=ISVAR("VAR1")	;IS_VAR sarà in questo caso TRUE
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	;diverse varianti di sintassi
N20 IS_VAR=ISVAR("VARARRAY[,]")	;IS_VAR è TRUE con un ;Array bidimensionale
N30 IS_VAR=ISVAR("VARARRAY")	;IS_VAR è TRUE, la variabile esiste
N40 IS_VAR=ISVAR ("VARARRAY[8,11] ")	;IS_VAR è FALSE, indice Array non consentito
N50 IS_VAR=ISVAR("VARARRAY[8,8]")	;IS_VAR è FALSE, errore di sintassi ;perchè manca "]"
N60 IS_VAR=ISVAR("VARARRAY[,8]")	;IS_VAR è TRUE, indice Array consentito
N70 IS_VAR=ISVAR("VARARRAY[8,]")	;IS_VAR è TRUE
DEF BOOL IS_VAR=FALSE	;Il parametro di trasferimento è un dato macchina

13.3 Richiamo della funzione ISVAR () e lettura dei dati macchina indice dell'array

N100 IS_VAR=ISVAR	;IS_VAR è TRUE
("\$MC_GCODE_RESET_VALUES[1] "	
DEF BOOL IS_VAR=FALSE	;Il parametro di trasferimento è una variabile di sistema
N10 IS_VAR=ISVAR("\$P_EP")	;IS_VAR sarà in questo caso TRUE
N10 IS_VAR=ISVAR("\$P_EP[X] ")	;IS_VAR sarà in questo caso TRUE

Esempio di lettura di array di dati macchina con e senza indice

Il primo elemento viene letto con

R1=\$MC_EXTERN_GCODE_RESET_VALUES

questo corrisponde al precedente

R1=\$MC_EXTERN_GCODE_RESET_VALUES[0]

oppure viene letto il primo elemento

R1=\$MA_POSTCTRL_GAIN[X1]

questo corrisponde al precedente

R1=\$MA_POSTCTRL_GAIN[0, X1]

Nelle azioni sincrone il primo elemento viene anche letto con

WHEN TRUE DO \$R1 = \$MC_EXTERN_GCODE_RESET_VALUES

questo corrisponde al precedente

WHEN TRUE DO \$R1 = \$MC_EXTERN_GCODE_RESET_VALUES[0]

e finora con l'allarme 12400 **non veniva letto**.

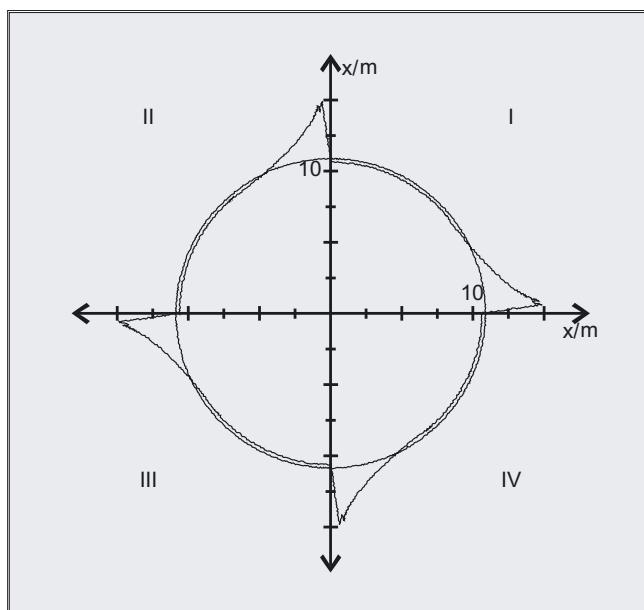
L'allarme 12400 viene comunque emesso con

R1=\$MA_POSTCTRL_GAIN

13.4 Apprendimento delle curve di compensazione (QECLRNON, QECLRNOF)

Funzione

La compensazione dell'errore del quadrante (QFK)) riduce gli errori sul profilo derivanti da non linearità meccaniche quando si inverte il movimento (ad es. attrito, giochi,) o torsione. I dati di compensazione ottimali possono essere adattati dal controllo numerico nella fase di apprendimento grazie ad una rete neurale per determinare automaticamente la curva di compensazione. L'apprendimento può avvenire contemporaneamente in max. 4 assi.



Programmazione

QECLRNON

oppure

QECLRNOF

Attivazione della procedura di apprendimento: QECLRNON

Il programma di apprendimento viene attivato nel programma NC mediante il comando QECLRNON seguito dall'indicazione degli assi:

QECLRNON (X1, Y1, Z1, Q)

Le curve vengono modificate solo se è attivo questo comando.

Disattivazione della procedura di apprendimento: QECLRNOF

A conclusione dei movimenti di apprendimento degli assi richiesti, con QECLRNOF la procedura di apprendimento viene esclusa contemporaneamente per tutti gli assi.

Parametri

QECLRNON (Asse.1,...4)	Attivazione della funzione "Apprendimento della compensazione dell'errore sul quadrante"
QECLRNOF	Disattivazione della funzione "Apprendimento della compensazione dell'errore sul quadrante"
QECLRN.SPF	Ciclo di apprendimento
QECDAT.MPF	Programma NC di esempio per la definizione delle variabili di sistema e la parametrizzazione del ciclo di apprendimento
QECTEST.MPF	Programma NC di esempio per il test della forma del cerchio

Descrizione

I movimenti degli assi necessari per l'apprendimento vengono generati con l'ausilio di un programma NC. In esso sono disponibili i movimenti di apprendimento in forma di un ciclo di apprendimento.

Primo apprendimento

Per il primo apprendimento in fase di messa in servizio sono contenuti nel dischetto del programma base PLC esempi di programmi NC per l'apprendimento dei movimenti e per la definizione delle variabili di sistema per la QFK:

Apprendimento successivo

Una ottimizzazione successiva di una curva già esistente è possibile con "Apprendimenti successivi". Vengono utilizzati i dati già esistenti nella memoria utente. Per gli apprendimenti successivi vengono adattati i programmi NC di esempio alle proprie esigenze.

I parametri del ciclo di apprendimento (ad esempio QECLRN.SPF) se necessario, vanno modificati per gli "Apprendimenti successivi":

- settare "Modo apprendimento" = 1
- ridurre eventualmente il "Numero dei cicli di apprendimento"
- disattivare eventualmente "Apprendimento per stadi" e definire i relativi limiti del settore.

13.5 Mandrino sincrono

Funzione

Nel funzionamento sincrono sono coinvolti un mandrino master (LS) ed un mandrino slave (FS), che formano la cosiddetta **coppia di mandrini sincroni**. Con accoppiamento attivo (funzionamento sincrono) il mandrino slave segue i movimenti del mandrino master in funzione della relazione definita.

Le coppie di mandrini sincroni possono essere progettate per ogni macchina con l'ausilio di dati macchina specifici per canale oppure definite tramite il partprogram NC. Per ogni canale NC sono gestibili contemporaneamente fino a 2 coppie di mandrini sincroni.

Dal partprogram l'accoppiamento può essere

- definito o modificato
- attivato
- disattivato
- cancellato.

Inoltre a seconda della versione del software è possibile

- attendere la condizione di sincronismo in corso
- modificare il comportamento di cambio blocco
- selezionare il tipo di accoppiamento (del valore di riferimento o del valore attuale) oppure preimpostare la traslazione angolare tra mandrino master e mandrino slave
- applicare una programmazione precedente del mandrino slave al momento dell'attivazione dell'accoppiamento
- correggere uno scostamento della corsa sincrona misurato o già noto.

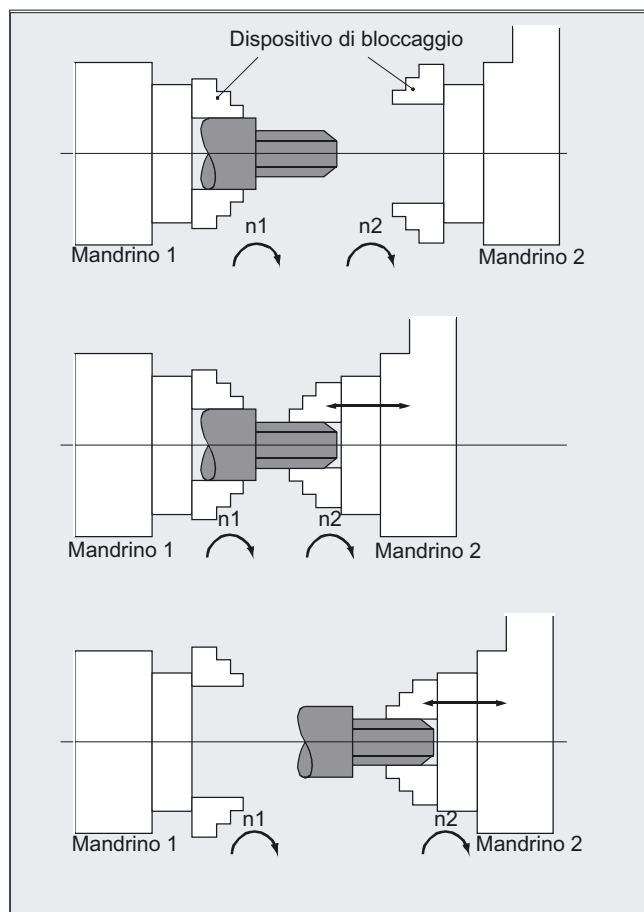
13.5.1 Mandrini sincroni (COUPDEF, COUPDEL, COUPON/ONC, COUPOF/OFS, COUPRES)

Funzione

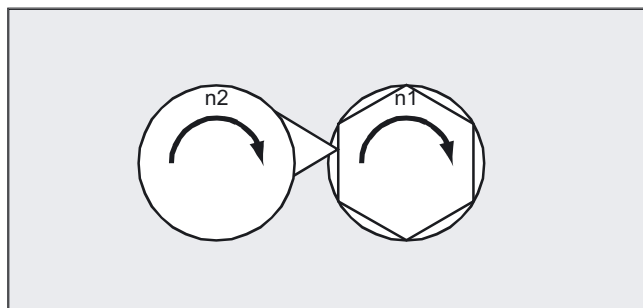
Nei torni la funzione mandrini sincroni permette lo scambio al volo del pezzo con i mandrini 1 e 2 in rotazione, ad esempio per la lavorazione completa, evitando così i tempi morti dovuti alla rotazione del pezzo nelle griffe.

Lo scambio del pezzo può avvenire con:

- Sincronismo di velocità ($n_{FS} = n_{LS}$)
- Sincronismo di posizione ($\phi_{FS} = \phi_{LS}$)
- Sincronismo di posizione con traslazione angolare ($\phi_{FS} = \phi_{LS} + \Delta\phi$)



Inoltre la definizione di un rapporto di trasmissione k_0 tra il mandrino principale e un "mandrino utensile" offre la premessa per la lavorazione poligonale (tornitura poligonale).



Programmazione

COUPDEF(FS, LS, $\ddot{U}_{FS}$, $\ddot{U}_{LS}$, comport.blocco, tipo di accoppiamento)

COUPON(FS, LS, POS_{FS})

COUPONC(FS, LS)

COUPOF(FS, LS, POS_{FS}, POS_{LS})

COUPOFS(FS, LS)

COUPOFS(FS, LS, POS_{FS})

COUPRES(FS, LS)

COUPDEL(FS, LS)

WAITC(FS, comportam.blocco., LS, comportam.blocco)

È possibile anche l'indicazione abbreviata senza mandrino master per:

COUPOF(FS), COUPOFS(FS), COUPRES(FS), COUPDEL(FS)

Nota

Per ogni istruzione COUPDEF, COUPON e COUPONC il mandrino slave e il mandrino master devono essere programmati in modo che non vengano emessi messaggi di allarme.

Gli altri parametri di accoppiamento vanno programmati solo se devono essere modificati. Per i parametri non specificati viene mantenuto l'ultimo stato.

Parametri

COUPDEF	Creare/modificare l'accoppiamento definito dall'utente
COUPON	Attivare l'accoppiamento. A partire dal numero di giri attuale, il mandrino slave si sincronizza sul mandrino master
COUPONC	Applicazione dell'accoppiamento all'attivazione con programmazione precedente di M3 S... o M4 S.... Un numero di giri differenziale del mandrino slave viene applicato immediatamente.
COUPOF	Disattivare l'accoppiamento. Cambio di blocco Rapidamente con cambio blocco immediato: COUPOF(S2, S1) cambio blocco solo dopo superamento della posizione di disattivazione: COUPOF(S2, S1, POS _{FS}) posizioni di disattivazione: COUPOF(S2, S1, POS _{FS} , POS _{LS})
COUPOFS	Disattivazione di un accoppiamento con arresto del mandrino slave. Cambio di blocco Rapidamente con cambio blocco immediato: COUPOFS(S2, S1) cambio blocco solo dopo superamento della posizione di disattivazione: COUPOFS(S2, S1, POS _{FS})
COUPRES	Ripristino dei parametri di accoppiamento ai dati macchina e ai dati setting progettati
COUPDEL	Cancellare l'accoppiamento definito dall'utente
WAITC	Attesa della condizione di sincronismo (i NOC vengono rimossi su IPO in caso di cambio blocco)
FS	Definizione del mandrino slave

Parametri opzionali

LS	Definizione del mandrino master;
	Indicazione con numero di mandrino: ad es. S2, S1
$\ddot{U}_{FS}$, $\ddot{U}_{LS}$	Parametri di trasmissione per mandrino slave = numeratore e mandrino master = denominatore preimpostazione = 1.0; impostazione del denominatore opzionale
Ritardo cambio blocco:	Comportamento al cambio blocco; il cambio blocco avviene:
"NOC"	Immediatamente
"FINE"	con "velocità sincrona fine"
"COARSE"	con "funzionamento sincrono grossolano"
"IPOSTOP"	con IPOSTOP (cioè dopo il movimento sincrono del riferimento) (preimpostazione)
	Il comportamento di cambio blocco ha validità modale
Tipo di accoppiamento	Tipo di accoppiamento: accoppiamento tra FS (slave) ed LS (master)
"DV"	Accoppiamento tramite riferimento (preimpostazione)
"AV"	accoppiamento al valore reale
"VV"	Accoppiamento di velocità
	Il tipo di accoppiamento ha validità modale.
POS_{FS}	Traslazione angolare tra mandrino master e mandrino slave
POS_{FS} , POS_{LS}	Posizioni di disattivazione di mandrino slave e mandrino master "Il cambio blocco viene abilitato dopo il superamento di POS_{FS} , POS_{LS} "

Esempio di lavorazione con mandrino master e mandrino slave

N05 M3 S3000 M2=4 S2=500	;LS = mandrino master = mandrino 1 ;FS = mandrino slave = mandrino 2 ;il mandrino master gira a 3000 giri/min, ;il mandrino slave a 500 giri/min
N10 COUPDEF (S2, S1, 1, 1, "NOC", "DV")	;definizione dell'accoppiamento; può essere anche ;progettato
...	
N70 SPCON	;attivare l'anello di posizione per il mandrino master ;(accoppiamento del riferimento)
N75 SPCON(2)	;attivare l'anello di posizione per il mandrino slave
N80 COUPON (S2, S1, 45)	;accoppiamento al volo sulla posizione di ;offset = 45 gradi
...	
N200 FA [S2] = 100	;velocità di posizionamento = 100 gradi/min
N205 SPOS[2] = IC(-90)	;muovere in sovrapposizione in ;direzione negativa di 90 gradi
N210 WAITC(S2, "Fine")	;attendere il sincronismo "fine"
N212 G1 X... Y... F...	;lavorazione
...	
N215 SPOS[2] = IC(180)	;muovere in sovrapposizione in ;direzione negativa di 180 gradi

N220 G4 S50	;tempo di attesa = 50 rotazioni del mandrino master
N225 FA [S2] = 0	;attivare la velocità progettata (DM)
N230 SPOS[2] = IC (-7200)	;eseguire 20 rotazioni Con velocità progettata in ;direzione negativa
...	
N350 COUPOF (S2, S1)	;disaccoppiamento al volo, S=S2=3000
N355 SPOSA[2] = 0	;arrestare il FS a zero gradi
N360 G0 X0 Y0	
N365 WAITS(2)	;attendere il mandrino 2
N370 M5	;arrestare il FS
N375 M30	

Esempio di programmazione del numero di giri differenziale

	;LS= mandrino master = mandrino 1
	;FS = mandrino slave = mandrino 2
N01 M3 S500	;il mandrino master gira a 500 giri/min
N02 M2=3 S2=300	;il mandrino slave gira a 300 giri/min
...	
N10 G4 F1	;tempo di sosta del mandrino master
N15 COUPDEF (S2, S1, -1)	;fattore di accoppiamento con rapporto di ;trasmissione -1:1
N20 COUPON (S2, S1)	;attivazione accoppiamento. Il numero di giri del ;mandrino slave si ricava dal numero di giri del ;mandrino master e dal fattore di accoppiamento
...	
N26 M2=3 S2=100	;programmazione del numero di giri differenziale

Esempi di applicazione di un movimento per il numero di giri differenziale

1. Attivazione dell'accoppiamento in caso di programmazione precedente del mandrino slave con COUPON

	;LS = mandrino master = mandrino 1
	;FS = mandrino slave = mandrino 2
N05 M3 S100 M2=3 S2=200	;il mandrino master gira a 1000 giri/min, ;il mandrino slave a 200 giri/min
N10 G4 F5	;tempo di attesa = 5 secondi del mandrino master
N15 COUPDEF (S2, S1, 1)	;il rapporto di trasmissione del mandrino slave con il ;mandrino slave è 1,0 (preimpostazione)
N20 COUPON (S2, S1)	;accoppiamento al volo su mandrino master
N10 G4 F5	;il mandrino slave gira a 1000 giri/min

2. Attivazione dell'accoppiamento in caso di programmazione precedente del mandrino slave con COUPONC

	;LS = mandrino master = mandrino 1
	;FS = mandrino slave = mandrino 2
N05 M3 S100 M2=3 S2=200	;il mandrino master gira a 1000 giri/min, ;il mandrino slave a 200 giri/min
N10 G4 F5	;tempo di attesa = 5 secondi del mandrino master
N15 COUPDEF (S2, S1, 1)	;il rapporto di trasmissione del mandrino slave con il ;mandrino slave è 1,0 (preimpostazione)
N20 COUPONC (S2, S1)	;accoppiamento al volo su mandrino master e ;applicazione del numero di giri precedente a S2
N10 G4 F5	;S2 gira a 100 giri/min + 200 giri/min = 300 giri/min

3. Attivazione dell'accoppiamento in caso di mandrino slave fermo con COUPON

	;LS = mandrino master = mandrino 1
	;FS = mandrino slave = mandrino 2
N05 SPOS=10 SPOS[2]=20	;mandrino slave S2 in funzionamento di ;posizionamento
N15 COUPDEF (S2, S1, 1)	;il rapporto di trasmissione del mandrino slave con il ;mandrino slave è 1,0 (preimpostazione)
N20 COUPON (S2, S1)	;accoppiamento al volo su mandrino master
N10 G4 F1	;l'accoppiamento viene concluso, ;S2 resta su 20 gradi

4. Attivazione dell'accoppiamento in caso di mandrino slave fermo con COUPONC

Funzionamento di posizionamento o funzionamento assi

Se prima dell'accoppiamento il mandrino slave si trova nel funzionamento di posizionamento o nel funzionamento assi, il mandrino slave si comporta allo stesso modo con COUPON (FS, LS) e COUPONC (FS, LS).

Definizione della coppia di mandrini sincroni

Accoppiamento progettato in forma fissa:

mandrino master e mandrino slave vengono definiti mediante dati macchina. Con questo tipo di accoppiamento, gli assi di macchina definiti come `LS` e `FS`, non possono essere variati dal programma NC. La parametrizzazione dell'accoppiamento può tuttavia avvenire nel partprogram NC con `COUPDEF` (presupposto: nessuna protezione alla scrittura impostata).

Accoppiamento definito dall'utente:

Con l'istruzione `COUPDEF` è possibile creare o modificare nuovi accoppiamenti nel partprogram NC. Se si deve definire una nuova relazione di accoppiamento, si deve cancellare innanzitutto un accoppiamento eventualmente attivo definito dall'utente con `COUPDEL`.

Definizione di un nuovo accoppiamento COUPDEF

Di seguito viene descritta la parametrizzazione del sottoprogramma predefinito:

```
COUPDEF(FS, LS,  $\ddot{U}_{FS}$ ,  $\ddot{U}_{LS}$ , comport.blocco, tipo di accoppiamento)
```

Mandrino slave e master, FS e LS

Con gli indicatori assi `FS` e `LS` viene definito l'accoppiamento in modo univoco. Essi devono essere programmati ad ogni istruzione `COUP`. Gli altri parametri di accoppiamento vanno programmati solo se devono essere variati (validità modale).

Esempio:

```
N ... COUPDEF(S2, S1,  $\ddot{U}_{FS}$ ,  $\ddot{U}_{LS}$ )
```

Significa:

`S2` = mandrino slave, `S1` = mandrino master

Rapporto di trasmissione k_U

Il rapporto di trasmissione viene preimpostato con un numero per `FS` (numeratore) ed un numero per `LS` (denominatore).

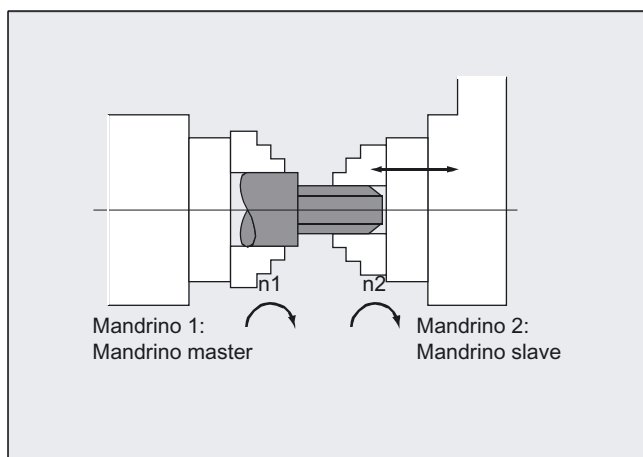
Possibilità:

- mandrino slave e mandrino master ruotano allo stesso numero di giri ($n_{FS} = n_{LS}$; k_U positivo)
- rotazione di `LS` e `FS` nello stesso senso o in senso opposto (k_U negativo)
- mandrino slave e mandrino master ruotano con numero di giri differente ($n_{FS} = k_U \cdot n_{LS}$; $k_U \neq 1$)
Impiego: tornitura poligonale

Esempio:

```
N ... COUPDEF(S2, S1, 1.0, 4.0)
```

Significa: il mandrino slave `S2` e il mandrino master `S1` ruotano con un rapporto di trasmissione di 0.25.



Nota

Deve essere programmato almeno il numeratore. Se non viene programmato alcun denominatore viene sempre supposto il valore "1".

Il rapporto di trasmissione può essere variato durante il movimento anche ad accoppiamento inserito.

Comportamento di cambio blocco NOC, FINE, COARSE, IPOSTOP

Nella definizione dell'accoppiamento il momento in cui deve avvenire il passaggio di blocco può essere definito come segue:

"NOC" subito (preimpostazione)

"FINE" con "sincronismo fine"

"COARSE" con "sincronia grossolana"

"IPOSTOP" con `IPOSTOP` (cioè dopo il funzionamento sincrono del riferimento)

Per la definizione del comportamento di cambio blocco è sufficiente scrivere le lettere in grassetto.

Tipo di accoppiamento DV, AV

Possibilità:

"DV" accoppiamento tramite riferimento tra `FS` e `LS` (preimpostazione)

"AV" accoppiamento tramite valore reale tra `FS` e `LS`



Cautela

Il tipo di accoppiamento può essere variato solo ad accoppiamento disinserito!

Attivazione del sincronismo COUPON, POS_{FS}

- Attivazione rapida dell'accoppiamento con riferimento angolare qualsiasi tra LS e FS:

```
N ... COUPON(S2, S1) ○
N ... COUPON(S2, S1, POSFS) ○
N ... COUPON(S2)
```

- Attivazione con traslazione angolare POS_{FS}

Per accoppiamenti sincroni alla posizione con pezzi profilati.

POS_{FS} si riferisce alla posizione 0° del mandrino master in direzione di rotazione positiva
campo di valori POS_{FS}: 0°... 359,999°:

```
COUPON(S2, S1, 30)
```

In questo modo la traslazione angolare può essere modificata anche quando l'accoppiamento è già attivo.

Posizionamento del mandrino slave

Quando l'accoppiamento dei mandrini sincroni è attivo, è possibile posizionare i mandrini slave nel campo $\pm 180^\circ$ anche indipendentemente dal movimento dei mandrini master.

Posizionamento SPOS

Il mandrino slave può essere interpolato con SPOS = ... Per ulteriori informazioni su SPOS vedere il Manuale di programmazione, Concetti fondamentali.

Esempio:

```
N30 SPOS[2]=IC(-90)
```

Numero di giri differenziale M3 S... o M4 S...

Un numero di giri differenziale è dato dalla sovrapposizione con segno di due sorgenti del numero di giri e viene riprogrammato per il mandrino slave ad es. con Sn=... o Mn=3, Mn=4 nel funzionamento di comando del numero di giri durante un accoppiamento attivo del mandrino sincrono. Questa componente del numero di giri viene ricavata dal mandrino master tramite il fattore di accoppiamento e il mandrino slave viene aggiunto con il segno corretto.

Nota

Con la direzione di rotazione M3 o M4 occorre riprogrammare anche il numero di giri S..., altrimenti la mancata programmazione viene segnalata con un allarme.

Per ulteriori informazioni sul numero di giri differenziale, consultare

Bibliografia: /FB2/ Manuale delle funzioni di ampliamento; Mandrino sincrono (S3)

Numero di giri differenziale per COUPONC**Applicazione di un movimento al numero di giri differenziale**

Attivando un accoppiamento sincrono con COUPONC, la programmazione precedente di M3 S... o M4 S... del mandrino slave viene sovrapposta. Con l'attivazione dell'accoppiamento viene quindi mantenuto il numero di giri del mandrino programmato in precedenza in un blocco. Il numero di giri differenziale viene applicato immediatamente.

Nota

Abilitazione del numero di giri differenziale

Il numero di giri differenziale che si imposta viene applicato se è abilitata anche una sovrapposizione del movimento. In caso contrario un allarme a cancellazione automatica segnala questa sovrapposizione non consentita.

Distribuzione della dinamica in base alla dinamica disponibile del motore

La dinamica da limitare del mandrino master deve essere limitata con la programmazione in modo tale che un'altra componente del movimento, legata ad es. a un numero di giri differenziale, non comprometta eccessivamente la dinamica del mandrino slave.

FA, ACC, OVRA, VELOLIMA: Velocità, accelerazione

Con FA[SPI] (Sn)] oppure FA[Sn], ACC[SPI(Sn)] oppure ACC[Sn] e OVRA[SPI(n)] oppure OVRA[Sn] detto anche VELOLIMA[Sn] consentono di programmare i valori di velocità e di accelerazione dei mandrini slave (vedere Manuale di programmazione, Concetti fondamentali). "n" rappresenta il numero di mandrino 1...n.

I campi di valori programmabili per la correzione della dinamica del mandrino slave Sn si ripercuotono su:

- avanzamento di assi di posizionamento o mandrini nel funzionamento di posizionamento FA[Sn] = ... fino a 999 999,999 mm/min o gradi/min
- correzione percentuale dell'accelerazione ACC[Sn] = da 1 a 200%
- correzione percentuale dell'avanzamento OVRA[Sn] = ... fino a 200%
- componente di velocità VELOLIMA[Sn] = correzione percentuale della velocità massima da 1 a 100%

Nota

Componente di accelerazione JERKLIMA[Sn]

La correzione dello strappo può essere indicata ma non ha ancora alcun effetto sul mandrino.

Per maggiori informazioni sulla progettazione della programmazione della dinamica mediante dati macchina, vedere **Bibliografia:** /FB2/ Manuale delle funzioni di ampliamento; Assi rotanti (R2)

Cambio blocco programmabile WAITC

Con WAITC si può definire il comportamento di cambio blocco con diverse condizioni di sincronismo (grossolano, fine, IPOSTOP) per la prosecuzione del programma ad es. dopo una modifica dei parametri di accoppiamento o dopo dei posizionamenti). In questo modo, il cambio blocco viene ritardato fino al raggiungimento delle condizioni di sincronismo in modo che possa essere ottenuta più velocemente la velocità sincrona. Se non vengono indicate le condizioni di sincronismo, resta valido il comportamento di cambio blocco programmato/progettato per il relativo accoppiamento.

Esempio:

N200 WAITC

Attendere le condizioni di sincronismo per tutti i mandrini slave attivi senza indicazione delle condizioni di sincronismo.

N300 WAITC(S2, "FINE", S4, "COARSE")

Attendere le condizioni di sincronismo indicate "grossolano" per i mandrini slave S2 e S4.

Disattivazione funzionamento sincrono COUPOF

Sono possibili tre varianti:

- Disattivazione rapida dell'accoppiamento; il cambio di blocco viene abilitato immediatamente:
COUPOF(S2, S1) o
COUPOF(S2); senza indicazione del mandrino master
- Dopo il superamento della posizione di disattivazione; il cambio di blocco viene abilitato solo dopo il superamento delle posizioni di disattivazione POS_{FS} ed eventualmente POS_{LS}.

Campo dei valori 0°... 359,999°:

COUPOF(S2, S1, 150)

COUPOF(S2, S1, 150, 30)

Disattivazione di un accoppiamento con arresto del mandrino slave COUPOFS

Sono possibili due varianti:

- Disattivazione rapida dell'accoppiamento ed arresto senza impostazione della posizione, il cambio blocco è subito possibile:
COUPOFS(S2, S1)
- Superando la posizione di disattivazione programmata per l'asse slave, riferita al sistema di coordinate macchina, il cambio blocco viene consentito solo dopo il superamento delle posizioni di disattivazione POS_{FS}.

Campo dei valori 0°... 359,999°:

COUPOFS(S2, S1, POS_{FS})

Cancellazione accoppiamenti COUPDEL

N ... COUPDEL(S2, S1) oder
N ... COUPDEL(S2); senza indicazione del mandrino master

Su un accoppiamento attivo tra mandrini sincroni l'accoppiamento si disattiva e cancella i dati di accoppiamento. Il mandrino slave assume l'ultimo numero di giri e corrisponde al comportamento precedente di COUPOF(FS, LS).

Parametri di accoppiamento COUPRES

Con l'istruzione "COUPRES" vengono attivati

- i parametri memorizzati nei dati macchina e nei dati setting (accoppiamento progettato fisso),
- e le preimpostazioni (accoppiamento definito dall'utente).

Con questa funzione i parametri programmati con COUPDEF (compreso il rapporto di trasmissione) vengono persi.

```
N ... COUPRES(S2, S1)oder  
N ... COUPRES(S2); senza indicazione del mandrino master
```

S2 = mandrino slave, S1 = mandrino master

Variabili di sistema

Attuale stato di accoppiamento mandrino slave

Lo stato attuale di accoppiamento del mandrino slave può essere letto nel partprogram dell' NC con la seguente variabile di sistema assiale:

```
$AA_COUP_ACT[FS]
```

FS = indicatore asse per il mandrino slave con il numero del mandrino, ad esempio S2.

Per il mandrino slave il valore letto ha il seguente significato:

0: nessun accoppiamento attivo

4: accoppiamento tra mandrini sincroni attivo

Attuale traslazione angolare

L'offset attuale di posizione del FS tramite riferimento rispetto al FS può essere letto nel partprogram del NC con la seguente variabile di sistema:

```
$AA_COUP_OFFS[S2]
```

L'offset di posizione dal lato valore reale può essere letto con:

```
$VA_COUP_OFFS[S2]
```

FS = indicatore asse per il mandrino slave con il numero del mandrino, ad esempio S2.

Nota

Se in seguito ad una disattivazione con accoppiamento inserito e funzionamento a seguire, il regolatore viene ripristinato, l'offset di posizione non corrisponde più a quello programmato in origine. In questo caso è possibile leggere il valore di offset di posizione modificato e, se necessario, correggere il partprogram.

13.6 Cambio elettronico (EG)

Funzione

Con la funzione "Cambio elettronico" si può controllare il movimento di un **asse slave** secondo un blocco di movimento lineare in funzione di un massimo di cinque **assi master**. I rapporti che intercorrono tra gli assi master e l'asse slave sono definiti per ogni asse master dal fattore di accoppiamento.

La quota calcolata di movimento dell'asse slave è ottenuta sommando le singole quote di movimento degli assi master moltiplicate per i rispettivi fattori di accoppiamento. Nel caso di attivazione di un raggruppamento di assi EG si può indurre la sincronizzazione dell'asse slave su una posizione definita. Dal partprogram un raggruppamento di rapporti può essere:

- definito,
- attivato,
- disattivato,
- cancellato.

Il movimento dell'asse slave può essere avviato con i

- riferimenti degli assi master e anche
- con i valori reali degli assi master.

Come ampliamento si possono realizzare anche relazioni non lineari tra asse master e asse slave tramite **tabelle di curve** (vedere il capitolo Comportamento vettoriale). I cambi elettronici possono essere attivati in cascata, cioè l'asse slave del primo cambio elettronico può essere l'asse master per un secondo cambio elettronico.

13.6.1 Definizione del cambio elettronico (EGDEF)

Funzione

Un raggruppamento di assi EG viene definito con l'indicazione dell'asse slave e di almeno un asse master, comunque al massimo di cinque assi master con il relativo tipo di accoppiamento:

EGDEF(asse slave, asse master1, tipo di accoppiamento1, asse master2, tipo di accoppiamento2,...).

Premessa

Premessa per la definizione di un raggruppamento di assi del cambio elettronico: per l'asse slave non può ancora essere definito un accoppiamento assi (eventualmente si deve cancellare prima quello esistente con EGDEL).

Programmazione

```
EGDEF(C, B, 1, Z, 1, Y, 1)
```

B, Z, Y influenzano C tramite il valore di riferimento

Il tipo di accoppiamento non deve essere lo stesso per tutti gli assi master e pertanto deve essere indicato singolarmente per ogni asse master. Nella definizione del gruppo di accoppiamenti del cambio elettronico (EG) il fattore di accoppiamento viene preimpostato a zero.

Nota

EGDEF provoca uno stop preelaborazione. La definizione di cambio con EGDEF si può anche utilizzare in modo invariato, se nei sistemi uno o più assi master sono legati all'asse slave tramite **tabella di curve**.

Parametri

EGDEF

Definizione di un cambio elettronico

asse slave

Asse che viene influenzato dagli assi master

Asse master1, ... asse master5

Assi che influenzano gli assi slave

Tipo di accoppiamento1, ...tip odi

L'asse slave viene influenzato da:

accoppiamento5

0: valore reale

1: riferimento

dell'asse master corrispondente

13.6.2 Attivazione del cambio elettronico (EGON)

Funzione

Per l'attivazione esistono 3 varianti.

Programmazione

Variante 1:

Il raggruppamento assi viene attivato selettivamente **senza** sincronizzazione con:

```
EGON(FA, "modo di cambio blocco", LA1, Z1, N1, LA2, Z2, N2, ..LA5, Z5, N5)
```

Variante 2:

Il raggruppamento assi viene attivato selettivamente **con** sincronizzazione con:

```
EGONSYN(FA, "modo di cambio blocco", SynPosFA, [, LAi, SynPosLAI, Zi, Ni])
```

Variante 3:

Il raggruppamento assi viene attivato selettivamente **con** sincronizzazione Il **modo di accostamento** viene preimpostato con:

```
EGONSYNE(FA, "modo di cambio blocco", SynPosFA, modo di accostamento [, LAi, SynPosLAI, Zi, Ni])
```

Parametri

Variante 1:

FA	asse slave
Modo di cambio blocco	Si possono utilizzare i seguenti modi di accostamento: "NOC" il cambio blocco avviene immediatamente "FINE" il cambio blocco avviene con "sincronismo fine" "COARSE" il cambio blocco avviene con "sincronismo grossolano" "IPOSTOP" il cambio blocco avviene con sincronismo del riferimento
LA1, ... LA5	Assi master
Z1, ... Z5	Numeratore per il fattore di accoppiamento i
N1, ... N5	Denominatore per il fattore di accoppiamento i
	Fattore di accoppiamento $i = \text{numeratore } i / \text{denominatore } i$

Possono essere programmati solo gli assi master precedentemente specificati con EGDEF .
Deve essere specificato almeno un asse master.

Variante 2:

FA	asse slave
Modo di cambio blocco	Si possono utilizzare i seguenti modi di accostamento: "NOC" il cambio blocco avviene immediatamente "FINE" il cambio blocco avviene con "sincronismo fine" "COARSE" il cambio blocco avviene con "sincronismo grossolano" "IPOSTOP" il cambio blocco avviene con sincronismo del riferimento
[, LAi, SynPosLAi, Zi, Ni]	(Non scrivere le parentesi quadre) minimo 1, max. 5 sequenze di:
LA1, ... LA5	Assi master
SynPosLAi	Posizione di sincronismo per l'asse master i.
Z1, ... Z5	Numeratore per il fattore di accoppiamento i
N1, ... N5	Denominatore per il fattore di accoppiamento i
	Fattore di accoppiamento $i = \text{numeratore } i / \text{denominatore } i$

Possono essere programmati solo gli assi master precedentemente specificati con EGDEF .
Attraverso le "posizioni di sincronismo" programmate per l'asse slave (SynPosFA) e per gli assi master (SynPosLA) vengono definite le posizioni in cui il gruppo di accoppiamento è valido in modo *sincrono*. Se all'avviamento il cambio elettronico non si trova in stato sincrono, l'asse slave procede verso la posizione di sincronismo definita.

Variante 3:

I parametri corrispondono a quelli della variante 2 con l'**aggiunta di**:

Modo di accostamento	Si possono utilizzare i seguenti modi di accostamento: "NTGT": successivo interspazio tra i denti, ottimizzato nel tempo "NTGP" : successivo interspazio tra i denti, ottimizzato nel percorso "ACN": posizionamento dell'asse rotante in direzione negativa assoluta "ACP": posizionamento dell'asse rotante in direzione positiva assoluta "DCT": ottimizzato nel tempo per la posizione sincrona programmata "DCP": ottimizzazione del percorso per la posizione sincrona programmata

La variante 3 ha effetto solo sugli assi modulo slave che sono accoppiati agli assi modulo master. L'ottimizzazione del tempo tiene conto dei limiti di velocità dell'asse slave.

Descrizione

Variante 1:

Le posizioni degli assi master così come quelle dell'asse slave nell'istante dell'attivazione vengono memorizzate come "Posizioni sincrone". Le "Posizioni sincrone" possono essere lette con la variabile di sistema \$AA_EG_SYN.

Variante 2:

Se gli assi modulo sono nel gruppo di accoppiamento, i loro valori di posizione vengono ridotti a modulo. Si garantisce così che venga raggiunta la posizione di sincronismo più vicina possibile (la cosiddetta *sincronizzazione relativa*: es. il successivo interspazio tra i denti). Se per l'asse slave non è stata data "Abilitazione sovrapposizione asse slave" segnale di interfaccia DB(30 +numero di asse), DBX 26 bit 4, non si attiva la funzione di posizione sincrona. Al contrario, il programma viene arrestato al blocco EGONSYN e viene dato l'allarme a cancellazione automatica 16771, finché non viene impostato il suddetto segnale.

Variante 3:

La distanza fra i denti (gradi) si ricava da: $360 * Z_i/N_i$. Nel caso in cui l'asse slave al momento del richiamo sia fermo, l'ottimizzazione del percorso influisce in modo analogo all'ottimizzazione del tempo.

Se l'asse slave è già in movimento, con NTGP indipendentemente dalla velocità attuale dell'asse slave, la sincronizzazione avviene nell'interspazio fra i denti successivo. Se l'asse slave è già in movimento, con NTGT in funzione della velocità attuale dell'asse slave, la sincronizzazione avviene nell'interspazio fra i denti successivo. Se necessario l'asse può anche essere frenato.

Tabelle delle curve

Se si utilizza una **tabella di curve** per l'asse master, allora:

Ni	il denominatore del fattore di accoppiamento per accoppiamenti lineari deve essere impostato a 0. (Il denominatore 0 sarebbe ammesso per accoppiamento lineari). Un denominatore nullo, per il controllo
Zi	deve essere interpretato come numero della tabella di curve da utilizzare. La tabella di curve con il numero impostato deve essere già al momento dell'attivazione.
LAI	L'indicazione dell'asse master corrisponde a quella dell'asse master con accoppiamento tramite fattore di accoppiamento (accoppiamento lineare).

Per **ulteriori informazioni** sull'uso delle tabelle per curve, per il collegamento in cascata dei cambi elettronici e la relativa sincronizzazione, consultare /FB3/ Manuale delle funzioni speciali; Accoppiamento degli assi e ESR (M3), capitolo "Trascinamento, accoppiamento al valore master"

Comportamento del cambio elettronico con power on, RESET, cambio del modo operativo, ricerca blocco

- Dopo Power On **nessun** accoppiamento è attivo.
- Gli accoppiamenti attivi sono mantenuti anche dopo RESET e cambio del modo operativo.
- Nella ricerca blocco i comandi di commutazione, cancellazione, definizione del cambio elettronico non vengono eseguiti né cumulati, ma ignorati.

Variabili di sistema del cambio elettronico

Grazie alle variabili di sistema del cambio elettronico, il partprogram può rilevare gli stati attuali di un raggruppamento assi EG ed eventualmente reagire di conseguenza.

Le variabili di sistema del cambio elettronico sono riportate in appendice. Esse sono contrassegnate con nomi che iniziano con:

\$AA_EG_ . . .

oppure

\$VA_EG_ . . .

13.6.3 Disattivazione del cambio elettronico (EGOFS)

Funzione

Per la disattivazione di un raggruppamento assi di cambio elettronico esistono tre possibilità.

Programmazione

Variante 1:

EGOFS(asse slave)

Il cambio elettronico viene disattivato. L'asse slave viene frenato fino all'arresto. Il richiamo provoca uno stop preelaborazione.

Variante 2:

EGOFS(asse slave,, asse master1, ...
asse master5)

Questa parametrizzazione del comando permette di eliminare **selettivamente** l'influenza di singoli assi master sul movimento dell'asse slave.

Deve essere specificato almeno un asse master. L'influenza degli assi master specificati sull'asse slave viene disattivata in modo mirato. Il richiamo provoca uno stop preelaborazione. Se restano degli assi master ancora attivi, l'asse slave continua a muoversi sotto la loro influenza. Una volta disattivati in questo modo tutti gli assi master, l'asse slave viene frenato fino all'arresto.

Variante 3:

EGOFC(mandrino slave1)

Il cambio elettronico viene disattivato. Il mandrino slave continua a muoversi al numero di giri/alla velocità attuali al momento della disattivazione. Il richiamo provoca uno stop preelaborazione.

Nota

Questa funzione è ammessa solo per i mandrini.

Cancellare la definizione di un cambio elettronico

Un raggruppamento di assi EG deve essere disattivato prima che possa essere cancellata la sua definizione.

EGDEL(asse slave)

La definizione dell'accoppiamento del raggruppamento di assi viene cancellata. È nuovamente possibile, fino al raggiungimento del numero massimo di raggruppamenti assi attivati contemporaneamente, ridefinire altri raggruppamenti assi con EGDEF. Il richiamo provoca uno stop preelaborazione.

13.6.4 Avanzamento al giro (G95)/cambio elettronico (FPR)

Funzione

Con l'istruzione FPR() è possibile indicare anche l'asse slave di un cambio elettronico come asse che determina l'avanzamento al giro. Per questo caso vale il seguente comportamento:

- l'avanzamento è in funzione della velocità di riferimento dell'asse slave del cambio elettronico.
- la velocità di riferimento viene calcolata a partire dalle velocità dei mandrini master e degli assi master a modulo (che non sono assi di contornatura) e dai fattori di accoppiamento assegnati.
- Le componenti di velocità di assi master lineari o non a modulo e i movimenti sovrapposti dell'asse slave non vengono considerati.

13.7 Funzione ampliata di arresto e svincolo

Funzione

La funzione "Arresto e svincolo ampliati" ESR (Extended Stopping and Retract) permette di reagire, in modo flessibile e sicuro per il pezzo, a sorgenti di errore selezionabili.

Reazioni possibili

La funzione "Arresto e svincolo ampliati" rende disponibili le seguenti reazioni:

- **"Arresto ampliato"** (azione autarchica dell'azionamento) è un arresto con ritardo definito nel tempo.
- **"Svincolo"** (azione autarchica dell'azionamento) comporta una "fuga" dal piano di lavorazione in una posizione di svincolo sicura. In questo modo si vuole evitare il pericolo di collisione esistente tra utensile e pezzo.
- **"Funzionamento generatorico"** (azione autarchica dell'azionamento)
Per i casi in cui l'energia del circuito intermedio non è sufficiente per uno svincolo sicuro, è possibile fare intervenire il funzionamento come generatore. Esso costituisce un modo operativo autonomo dell'azionamento e, in caso di guasti di rete o simili, mette a disposizione l'energia necessaria per il circuito intermedio dell'azionamento al fine di assicurare un "arresto" ed uno "svincolo" regolari.

Ulteriori ampliamenti

- **Arresto ampliato** (controllato dall'NC)
è un arresto con ritardo definito gestito dall'NC per salvaguardare il profilo.
- **Svincolo** (controllato dall'NC)
comporta una "fuga" dal piano di lavorazione in una posizione di svincolo sicura. Viene evitata una possibile collisione tra utensile e pezzo. Nel caso di lavorazione di ingranaggi si deve intendere ad es. uno stacco dal vano interdente in lavorazione.

Tutte le reazioni possono essere utilizzate indipendentemente l'una dall'altra. Per maggiori informazioni consultare

/FB3/ Manuale delle funzioni speciali; Accoppiamento degli assi e ESR (M3).

Cause possibili

Le seguenti cause di errore possono attivare un "Arresto e svincolo ampliati": **cause generiche** (esterne al NC/globali oppure specifiche del gruppo di modi operativi/canale):

- ingressi digitali (es.: su unità NCU o blocco terminale) oppure l'immagine di ingressi digitali come retroazione interna al controllo numerico \$A_IN, \$A_OUT
- Stato del canale \$AC_STAT
- Segnali VDI (\$A_DBB)
- Messaggi cumulativi per un certo numero di allarmi (\$AC_ALARM_STAT)

Cause assiali

- Soglia di svincolo di emergenza dell'asse slave (funzionamento sincrono dell'accoppiamento elettronico, \$VA_EG_SYNCDIFF[asse slave])
- Azionamento: soglia di preallarme del circuito intermedio (possibile sottotensione), \$AA_ESR_STAT[asse]
- Azionamento: soglia di velocità minima del generatore (mancanza di energia di rotazione recuperabile), \$AA_ESR_STAT[asse].

Logica combinatoria delle azioni sincrone statiche: cause/reazioni combinatorie

Le possibilità combinatorie flessibili delle azioni sincrone statiche vengono sfruttate per provocare determinate reazioni in base alle sorgenti in modo relativamente attuale.

La combinazione di tutte le cause rilevanti con l'ausilio di azioni sincrone statiche dipende dalla scelta dall'utente. L'utente può valutare le variabili di sistema delle sorgenti nel loro insieme oppure selettivamente con l'ausilio di maschere a bit e collegarvi le reazioni desiderate. Le azioni sincrone statiche sono attive in tutti i modi operativi.

Per una descrizione dettagliata dell'uso delle azioni sincrone consultare

Bibliografia: /FBSY/ Descrizione delle funzioni, Azioni sincrone.

Attivazione

Abilitazione della funzione:

`$AA_ESR_ENABLE`

Le funzioni: funzionamento come generatore, arresto, svincolo, vengono abilitate con l'impostazione del relativo segnale di controllo (`$AA_ESR_ENABLE`). Questo segnale di controllo può essere modificato da azioni sincrone.

Esecuzione della funzione (attivazione simultanea di tutti gli assi abilitati)

`$AN_ESR_TRIGGER`

Il funzionamento come generatore si mette in funzione "automaticamente" nell'azionamento al riconoscimento di un pericolo di sottotensione del circuito intermedio.

L'arresto e/o lo svincolo autarchici degli azionamenti diventano attivi al riconoscimento di un guasto della comunicazione (tra NC e azionamento), nonché al rilevamento di una sottotensione del circuito intermedio nell'azionamento (previa configurazione e abilitazione).

Arresti e/o svincoli indipendenti dall'azionamento possono essere anche attivati dal lato NC settando il relativo segnale di comando `$AN_ESR_TRIGGER` (comando di broadcast a tutti gli azionamenti).

13.7.1 Reazioni autarchiche degli azionamenti a ESR

Funzione

Le reazioni autarchiche degli azionamenti vengono definite in modo assiale, ovvero ogni azionamento, quando viene attivato, elabora in modo indipendente la richiesta di arresto/svincolo. Un accoppiamento interpolatorio o fedele al percorso degli assi non viene dato nell'arresto o nello svincolo, un riferimento agli assi avviene con comando a tempo.

Durante e dopo l'esecuzione di reazioni autarchiche degli azionamenti, l'azionamento non risponde più alle abilitazioni o ai comandi di movimento dell'NC. È necessario spegnere e riaccendere il sistema. L'allarme "26110: arresto/svincolo autarchico dell'azionamento attivato" indica proprio questo.

Parametri

Funzionamento generatorico

Il funzionamento come generatore viene

- configurato: con il DM 37500: **10**
- abilitato: variabile di sistema `$AA_ESR_ENABLE`
- attivato: in funzione dell'impostazione dei dati macchina degli azionamenti quando la tensione nel circuito intermedio scende sotto la soglia minima.

Svincolo (azione autarchica dell'azionamento)

Lo svincolo autarchico tramite azionamento viene

- configurato: con il DM 37500: **11**; l'impostazione del tempo e della velocità di svincolo avvengono con i DM, vedere "esempio: utilizzo della reazione autarchica dell'azionamento" alla fine di questo capitolo,
- abilitato: variabile di sistema `$AA_ESR_ENABLE`

- eseguito: variabile di sistema \$AN_ESR_TRIGGER.

Arresto (modo autarchico dell'azionamento)

L'arresto autarchico tramite azionamento viene

- configurato: con il DM 37500: 12 e l'impostazione del tempo viene abilitata tramite il DM,
- (\$AA_ESR_ENABLE) e
- avviato: variabile di sistema \$AN_ESR_TRIGGER.

Esempio di utilizzo della reazione autarchica dell'azionamento

Esempio di configurazione

- l'asse A deve funzionare come azionamento generatore,
- l'asse X deve ritrarsi di 10 mm alla velocità massima in caso di errore e
- gli assi Y e Z devono essere arrestati con ritardo di 100 ms, in modo che l'asse di svincolo abbia il tempo di rimuovere l'accoppiamento meccanico.

Esempio di procedimento

1. Abilitare le opzioni "Arresto e svincolo ampliati" e "Azioni indipendenti dal modo operativo" (comprese "Azioni sincrone statiche IDS ...").
2. Assegnazione delle funzioni:


```
$MA_ESR_REACTION[X] = 11,
$MA_ESR_REACTION[Y] = 12,
$MA_ESR_REACTION[Z] = 12,
$MA_ESR_REACTION[A] = 10;
```
3. Configurazione dell'azionamento:


```
DM 1639: RETRACT_SPEED[X] = 400000H in direzione pos. (velocità max.),
= FFC00000H in direzione neg.,
DM 1638: RETRACT_TIME[X] = 10ms (tempo di svincolo),
DM 1637: GEN_STOP_DELAY[Y] = 100ms,
DM 1637: GEN_STOP_DELAY[Z] = 100ms,
DM 1635: GEN_AXIS_MIN_SPEED[A] = velocità minima generatore
(giri/min) ..
```
4. Abilitazione della funzione (da partprogram o azioni sincrone):


```
$AA_ESR_ENABLE[X] = 1,
$AA_ESR_ENABLE[Y] = 1,
$AA_ESR_ENABLE[Z] = 1,
$AA_ESR_ENABLE[A] = 1.
```
5. Portare l'azionamento generatore al numero di giri di "regime" (ad es.: nel funzionamento del mandrino M03 S1000)
6. Formulare la condizione di attivazione come azione sincrona statica, es.:
 - in funzione dell'intervento dell'asse come generatore: IDS = 01 WHENEVER \$AA_ESR_STAT[A]>0 DO \$AN_ESR_TRIGGER = 1
 - e/o in funzione di allarmi che attivano il funzionamento a seguire (Bit13=2000H): IDS = 02 WHENEVER (\$AC_ALARM_STAT_B_AND 'H2000'>0 DO \$AN_ESR_TRIGGER = 1
 - come pure in funzione del controllo della sincronizzazione EG (se es. Y è definito come asse slave EG e lo scostamento massimo ammesso della sincronizzazione deve essere 100 µm):


```
IDS = 03 WHENEVER ABS($VA_E_SYNCDIFF[Y]) > 0.1
DO $AN_ESR_TRIGGER = 1
```

13.7.2 Reazioni allo svincolo controllate dall'NC

Funzione

Per le reazioni allo svincolo controllate dall'NC sono necessarie determinate condizioni iniziali che in seguito verranno denominate come presupposti. Se questi presupposti per lo svincolo vengono soddisfatti allora viene attivato lo svincolo rapido.

Nel partprogram deve essere stata programmata la posizione di svincolo POLF. Per il movimento di svincolo devono essere presenti i segnali di abilitazione.

Programmazione

```
POLF[geo |mach]=, = valore
POLFA(asse, tipo, valore)
```

```
POLFA(asse, tipo)
POLFA(asse, 0/1/2)tipo)
```

```
POLFA(asse, 0, $AA_POLFA[asse])
POLFA(asse, 0)
```

```
POLFMASK(nome asse1, nome asse2,
...)
```

```
POLFMLIN(nome asse1, nome asse2,
...)
```

Posizione di arrivo dell'asse di svincolo

Posizione di svincolo di singoli assi

Sono consentite le seguenti forme abbreviate:

Forma abbreviata per lo svincolo di singoli assi

Attivazione o disattivazione rapida

Provoca **uno** stop preelaborazione

Non provoca uno stop preelaborazione

Selezione dell'asse per lo svincolo

Assi senza relazione

Selezione dell'asse per lo svincolo

Assi con relazione lineare

Attenzione

Se utilizzando la forma abbreviata `POLFA` viene modificato solo il tipo, l'utilizzatore deve assicurarsi che la posizione o il percorso di svincolo contengano un valore ammissibile. In particolare le posizioni ed il percorso di svincolo devono essere reimpostati dopo Power On.

Parametri

geo | mach

Asse

Tipo

asse geometrico oppure asse canale/macchina che eseguono lo svincolo.

Identificatore asse dei singoli assi validi

Valori di posizione dei singoli assi del tipo:

invalidare il valore di posizione

il valore di posizione è assoluto

il valore di posizione è incrementale (distanza)

Valore	Posizione di svincolo, per asse geometrico vale il SCP, altrimenti il SCM. Nel caso di identificatori uguali per asse geo e canale/asse macchina lo svincolo avviene nel sistema di coordinate pezzo. E' consentita la programmazione incrementale. Posizione di svincolo con tipo=1 per singoli assi Percorso di svincolo con tipo=2 per singoli assi Il valore viene accettato anche con tipo=0. Solo che questo valore viene contrassegnato come non valido e per lo svincolo deve essere nuovamente programmato.
POLF	L'istruzione POLF ha validità modale.
POLFA	Se un asse non è un singolo asse oppure manca il tipo o tipo=0, vengono emessi i relativi allarmi 26080 e 26081.
POLFMASK,	Con l'istruzione POLFMASK gli assi indicati vengono abilitati per lo svincolo "senza relazione tra gli assi". L'istruzione POLFMASK() senza indicazione di un asse, disattiva lo svincolo rapido per tutti gli assi che vengono svincolati "senza relazione tra gli assi".
POLFMLIN,	Con l'istruzione POLFMLIN gli assi indicati vengono abilitati per lo svincolo "con relazione lineare tra gli assi". L'istruzione POLFMLIN() senza indicazione di un asse, disattiva lo svincolo rapido per tutti gli assi che vengono svincolati "con relazione lineare tra gli assi".
Nomi degli assi	Nomi degli assi, che devono portarsi con LIFTFAST sulle loro posizioni definite con POLF. Tutti gli assi indicati si devono trovare nello stesso sistema di coordinate. Prima che tramite POLFMASK oppure POLFMLIN possa essere abilitato lo svincolo rapido su una determinata posizione, per gli assi selezionati deve essere stata programmata una posizione con POLF. Non è previsto alcun dato macchina per la preimpostazione dei valori di POLF. Nell'interpretazione di POLFMASK oppure POLFMLIN viene emesso l'allarme 16016 se non è ancora stato programmato POLF.

Nota

Se vengono abilitati assi in sequenza con **POLFMASK**, **POLFMLIN** oppure **POLFMLIN**, **POLFMASK**, per ogni singolo asse vale l'ultima impostazione.

**Cautela**

La posizione programmata con **POLF** e l'attivazione tramite **POLFMASK** oppure **POLFMLIN** vengono cancellate allo start del partprogram. Questo significa che l'utente deve riprogrammare in ogni partprogram i valori per **POLF** e gli assi selezionati in **POLFMASK** oppure **POLFMLIN**.

Per ulteriori indicazioni sulla modifica del sistema di coordinate, sull'effetto con assi rotanti modulo, ecc., vedere
/FB3/ Manuale delle funzioni speciali; "Accoppiamenti degli assi e ESR (M3)".

Esempio di svincolo di un asse singolo

DM 37500): ESR_REACTION[AX1] = 21	;svincolo gestito da NC
...	
\$AA_ESR_ENABLE[AX1] = 1	
POLFA(AX1,1, 20.0)	;a AX1 viene assegnata la posizione ;di svincolo assiale 20.0 (assoluta).
\$AA_ESR_TRIGGER[AX1] = 1	;lo svincolo inizia da questo punto.

Premessa

Svincolo

- gli assi selezionati con POLFMASK oppure POLFMLIN ,
- le posizioni specifiche per asse definite con POLF ,
- le posizioni di svincolo di un asse singolo definite con POLFA ,
- la finestra temporale impostata nel
DM 21380: ESR_DELAY_TIME1 e
DM 21381: ESR_DELAY_TIME2,
- l'esecuzione tramite la variabile di sistema \$AC_ESR_TRIGGER
\$AA_ESR_TRIGGER per i singoli assi,
- la ESR definita
DM 37500: ESR_REACTION = 21,
- LFPOS del 46. gruppo di codici G modale.

Abilitazione e start dello svincolo controllato da NC

Se la variabile di sistema \$AC_ESR_TRIGGER = 1 è settata e se in questo canale è configurato un asse di svincolo (cioè DM 37500: ESR_REACTION = 21) e per questo asse è settato \$AA_ESR_ENABLE = 1 , allora in questo canale viene attivato **LIFTFAST** .

Nel partprogram deve essere stata programmata la posizione di svincolo POLF . Nel caso di svincolo del singolo asse con POLFA(asse, tipo, valore) deve essere stato programmato il valore e devono essere state rispettate le seguenti condizioni:

- Deve essere settato \$AA_ESR_ENABLE = 1.
- POLFA(asse) nell'istante dell'evento di trigger deve essere un singolo asse.
- POLFA(tipo) del tipo=1 o tipo=2

Per il movimento di svincolo devono essere presenti i segnali di abilitazione.

- Il movimento di svincolo configurato con LFPOS, POLF dell'asse selezionato con POLFMASK oppure POLFMLIN dell'asse o degli assi selezionati, sostituisce il movimento vettoriale programmato nel partprogram per questo asse o questi assi.
- Il movimento di svincolo ampliato (cioè LIFTFAST/LFPOS attivate tramite \$AC_ESR_TRIGGER) **non può essere interrotto** e può essere terminato prima del tempo solo con l'EMERGENZA.

Per lo svincolo è disponibile al max. la somma dei tempi DM 21380: ESR_DELAY_TIME1 e MD 21381: ESR_DELAY_TIME2. Trascorso questo intervallo di tempo viene attivata la frenatura rapida anche per l'asse di svincolo con successivo funzionamento a seguire.

Direzione di svincolo in caso di distacco rapido e cambio asse

Si tiene conto del frame valido al momento dell'attivazione dello svincolo rapido.

Nota

Frame con rotazione hanno influenza anche sulla direzione dalla quale si discosta tramite POLF . Lo svincolo gestito da NC viene

- configurato: con il DM 37500: 21 come anche 2 impostazione del tempo tramite DM vedere sopra;
 - (\$AA_ESR_ENABLE) e
 - avviato: variabile di sistema \$AC_ESR_TRIGGER per singoli assi con \$AA_ESR_TRIGGER.
-

In caso di svincolo controllato da NC viene usato LIFTFAST/LFPOS come per la filettatura e con la variabile di sistema \$AC_ESR_TRIGGER viene abilitato l'asse di svincolo configurato nel canale per il distacco rapido. Lo svincolo provocato con \$AC_ESR_TRIGGER è bloccato contro il distacco multiplo.

Gli assi di svincolo devono essere sempre assegnati precisamente a un canale NC e non possono essere scambiati tra i canali. Il tentativo di scambio di un asse di svincolo in un altro canale viene visualizzato con l'allarme 26122.

Solo dopo che questo asse è stato nuovamente disattivato con \$AA_ESR_ENABLE[AX] = 0 , l'asse può essere scambiato in un nuovo canale. Dopo il cambio asse, gli assi possono essere nuovamente abilitati con \$AA_ESR_ENABLE[AX] = 1 .

Gli assi neutri non possono eseguire alcun ESR controllato da NC.

Con \$AA_ESR_ENABLE[AX] = 1 e cambio asse nello stato neutrale, viene emesso il ShowAlarm 26121 sopprimibile.

13.7.3 Reazioni all'arresto controllate dall'NC

Funzione

Arresto

La sequenza per l'arresto ampliato (controllato da NC) è indicata dai due dati macchina

DM 21380): ESR_DELAY_TIME1 e

DM 21381): ESR_DELAY_TIME2 .

Per la durata del tempo impostato nel DM 21380, l'asse interpola indisturbato come programmato. Trascorso il tempo indicato dal DM si attiva la frenatura (stop con rampa) guidata interpolatoriamente. Per la frenatura guidata interpolatoriamente è disponibile al massimo il tempo indicato dal DM 21381; trascorso questo tempo si attiva la frenatura rapida ed infine il funzionamento a seguire.

Abilitazione e start dell'arresto controllato da NC

L'arresto controllato da NC è

configurato: con il DM 37500: **22** come 2 impostazione del tempo tramite i due DM descritti in precedenza;

abilitato (\$AA_ESR_ENABLE) e

avviato: variabile di sistema \$AC_ESR_TRIGGER per singoli assi con \$AA_ESR_TRIGGER.

Esempio di arresto di un asse singolo

```
DM 37500): ESR_REACTION[AX1] = 22 ; arresto gestito da NC
DM 21380): ESR_DELAY_TIME1[AX1] = 0.3
DM 21381): ESR_DELAY_TIME2[AX1] = 0.06
...
$AA_ESR_ENABLE[AX1] = 1
$AA_ESR_TRIGGER[AX1] = 1 ; l'arresto inizia da questo punto.
```

13.7.4 Funzionamento generatorio/mantenimento del circuito intermedio

Funzione

Mediante la progettazione del dato macchina dell'azionamento e la corrispondente programmazione con azioni sincrone statiche (\$AA_ESR_ENABLE), è possibile compensare brevi interruzioni di tensione del circuito intermedio. Il tempo che può essere compensato dipende dall'energia accumulata dal generatore utilizzato per mantenere l'alimentazione del circuito intermedio e dall'energia necessaria per mantenere i movimenti momentanei (alimentazione del circuito intermedio e controllo della velocità del generatore).

Se la tensione del circuito intermedio scende sotto la soglia minima, l'asse/mandrino interessato passa dal funzionamento regolato in posizione o velocità al funzionamento generatorio. Mediante frenatura dell'azionamento (preimpostazione riferimento di velocità = 0), il circuito intermedio viene rialimentato.

Per maggiori informazioni vedere
/FB3/ Manuale delle funzioni speciali; Accoppiamenti degli assi e ESR (M3).

13.7.5 Arresto autarchico tramite azionamento

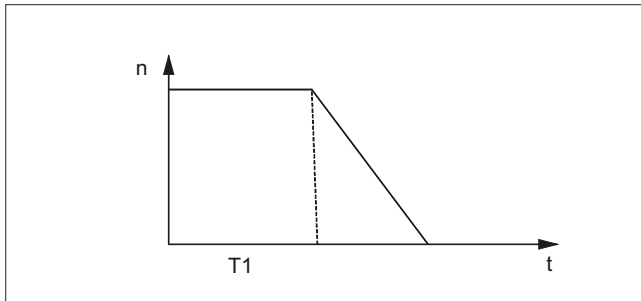
Funzione

Gli azionamenti di un raggruppamento precedentemente accoppiato possono essere arrestati mediante ritardo temporizzato alla disinserzione con il minimo di scostamenti reciproci, quando ciò non può essere eseguito dal controllo numerico.

L'arresto autarchico dell'azionamento viene configurato e abilitato tramite dato macchina (durata del ritardo T1 nel DM): viene abilitato con la variabile di sistema \$AA_ESR_ENABLE e avviato con \$AN_ESR_TRIGGER.

Reazioni

Per il tempo T1 continua ad essere emesso il valore di riferimento della velocità attivo nel momento in cui si è verificato l'errore. Si tenta in questo modo di mantenere il movimento attivo prima del guasto finché non viene rimosso l'accoppiamento geometrico o finché non è concluso il movimento di svincolo iniziato parallelamente in altri azionamenti. Questo può essere opportuno per tutti gli azionamenti master/slave o per gli azionamenti che si trovano in accoppiamento o in raggruppamento.



Trascorso questo tempo T1, tutti gli assi con valore di riferimento della velocità inserzione zero vengono arrestati al limite di corrente e quando viene raggiunto l'arresto oppure una volta trascorso il tempo (+DM azionamento), gli impulsi vengono cancellati.

13.7.6 Svincolo autarchico tramite azionamento

Funzione

Gli assi con azionamenti SIMODRIVE 611Digital possono (se progettato ed abilitato) eseguire autonomamente un movimento di svincolo

- anche in caso di guasto del controllo numerico (rilevamento di un errore dei segni di vita)
- in caso di caduta della tensione del circuito intermedio sotto una soglia di allerta,
- in caso di attivazione provocata dalla variabile di sistema \$AN_ESR_TRIGGER.

Il movimento di svincolo avviene in modo autarchico tramite l'azionamento SIMODRIVE 611Digital. Fin dall'inizio della fase di svincolo l'azionamento mantiene automaticamente le sue abilitazioni ai valori validi in precedenza.

Per maggiori informazioni vedere

/FB3/ Manuale delle funzioni speciali; Funzioni degli assi e ESR (M3).

13.8 Comunicazione link

Funzione

Il collegamento NCU, ovvero la connessione tra più unità NCU di un impianto, viene utilizzato negli impianti con struttura decentralizzata. Nel caso siano presenti molti assi e canali, p.e. tavole rotanti e macchine a più mandrini, le capacità di calcolo, le possibilità di configurazioni e le aree di memoria possono essere praticamente al limite con una sola NCU.

Più NCU collegate in rete tra di loro con un modulo NCU-link consentono una soluzione aperta verso l'alto che può soddisfare le esigenze di determinate macchine utensili. Il modulo NCU-link (HW) realizza una comunicazione rapida NCU-NCU con un accesso in scrittura e lettura su variabili di sistema.

Premessa

La funzionalità è strettamente correlata alle opzioni che si ordinano.

Variabili Link

Le variabili di collegamento sono **dati di sistema globali** che, dalla NCU collegata, possono essere utilizzate come **variabili di sistema**.

L'utente (in questo caso generalmente è il costruttore della macchina) determina:

- **contenuti** di queste variabili,
- i relativi **tipi di dati**,
- il loro **utilizzo**,
- la posizione (**indice di accesso**) nella memoria Link.

Applicazioni per variabili link

- stati macchina globali
- Apertura/chiusura del bloccaggio pezzo
- ecc.

Comportamento temporale delle applicazioni

Le applicazioni delle varie NCU che accedono simultaneamente in un **determinato istante** alla memoria di collegamento devono utilizzare in modo **uniforme** la memoria di collegamento. Per i processi del tutto disaccoppiati cronologicamente, la memoria di accoppiamento può essere occupata diversamente.



Avvertenza

Un processo di scrittura di una variabile di collegamento è concluso solo quando anche le altre NCU trovano le informazioni scritte. Per questo è necessario il tempo di circa due clock dell'interpolatore. La scrittura locale nella memoria di collegamento viene ritardata dello stesso tempo, in modo da mantenere la coerenza.

Per maggiori informazioni vedere
/FB2/ Manuale delle funzioni di ampliamento; Diversi pannelli operativi e NCU (B3).

13.8.1 Accesso ad un settore di memoria globale della NCU

Funzione

Più NCU collegate mediante moduli possono accedere in lettura e in scrittura a una memoria globale della NCU con le variabili di sistema descritte di seguito.

- Ogni NCU collegata tramite il modulo link può utilizzare in modo uniforme delle **variabili link globali** utilizzabili da tutte le NCU connesse.
- Le variabili link sono programmabili come variabili di sistema. Generalmente il significato di queste variabili viene definito e documentato dal costruttore della macchina.
- Applicazioni per variabili link
- Volumi di dati relativamente bassi
- Rapida velocità di trasferimento, quindi: l'impiego è previsto per informazioni con criticità di tempo.
- L'accesso a queste variabili di sistema è possibile dal **partprogram** e dalle **azioni sincrone**. È possibile progettare la capacità di memoria per le variabili di sistema globali.

Dopo un clock di interpolazione tutte le NCU coinvolte possono leggere in modo consistente un valore nuovo scritto in una variabile di sistema globale.

Parametri

Le **variabili link** sono memorizzate nella memoria di collegamento. Dopo l'avvio, la memoria di collegamento viene inizializzata con 0.

Nell'ambito della memoria di collegamento possono essere attivate le seguenti variabili link:

- INT \$A_DLB[i] ;byte di dati (8 Bit)
- INT \$A_DLW[i] ;word di dati (16 BIT)
- INT \$A_DLD[i] ;doppia word di dati (32 Bit)
- REAL \$A_DLR[i] ;dato Real (64 Bit)

A seconda del tipo, nella scrittura/lettura delle variabili di collegamento vengono attivati 1, 2, 4, 8 byte.

L'indice i definisce l'inizio delle variabili corrispondenti riferito all'inizio della memoria di collegamento progettata. Il conteggio dell'indice parte da 0.

Campi di valori

Ai tipi di dati sono collegati i seguenti campi di valori:

BYTE: Da 1 a 255:

WORD: Da -32768 a 32767

DWORD: Da -2147483646 a +2147483647

REAL: $\pm(2,2 \cdot 10^{-308} \dots 1,8 \cdot 10^{+308})$

Esempio

| \$A_DLB[5] = 21

Il 5° byte nella memoria di collegamento comune contiene il valore 21.

13.9 Container assi (AXCTWE, AXCTWED)

Funzione

Nelle trasferte a tavola rotante /macchine a più mandrini gli assi portapezzo si muovono da un'unità di lavorazione alla successiva.

Dato che le unità di lavoro dipendono da differenti canali NCU, al cambio di una stazione/posizione, gli assi portapezzo devono essere dinamicamente riassociati al corrispondente canale NCU. I **container assi** servono a questo scopo.

In un determinato istante nell'unità di lavorazione locale è sempre attivo un solo asse/mandrino di bloccaggio del pezzo. Il container assi riunisce le possibilità di collegamento con tutti gli assi di bloccaggio/mandrini, **una** sola delle quali è sempre **attivata** per l'unità di lavorazione.

Programmazione

La traslazione con ampiezza di incremento n delle registrazioni nel container assi avviene tramite i comandi:

AXCTSWE (CT _i)	AXIS CONTAINER SWITCH ENABLE
AXCTSWED (CT _i)	AXIS CONTAINER SWITCH ENABLE DIRECT

Parametri

AXCTSWE	Ogni canale i cui assi sono inseriti nel contenitore specificato abilita una rotazione del container
AXCTSWED	Sotto unico effetto del canale attivo per ruotare il container assi dell'incremento impostato. Gli assi inseriti nel container vengono abilitati solo se gli altri canali che hanno assi nel contenitore si trovano in stato RESET.
CT _i	numero del container assi i cui contenuti devono essere traslati o
oppure	
ad es. A_CONT1	nome individuale del container assi impostato tramite DM.

Container assi

Tramite i container assi si possono abbinare:

- assi locali e/o
- assi di collegamento (v.concetti fondamentali)

Il cambio degli assi utilizzabili definiti tramite un container assi avviene tramite traslazione delle registrazioni nel container assi.

La traslazione può essere attivata mediante il **partprogram**.

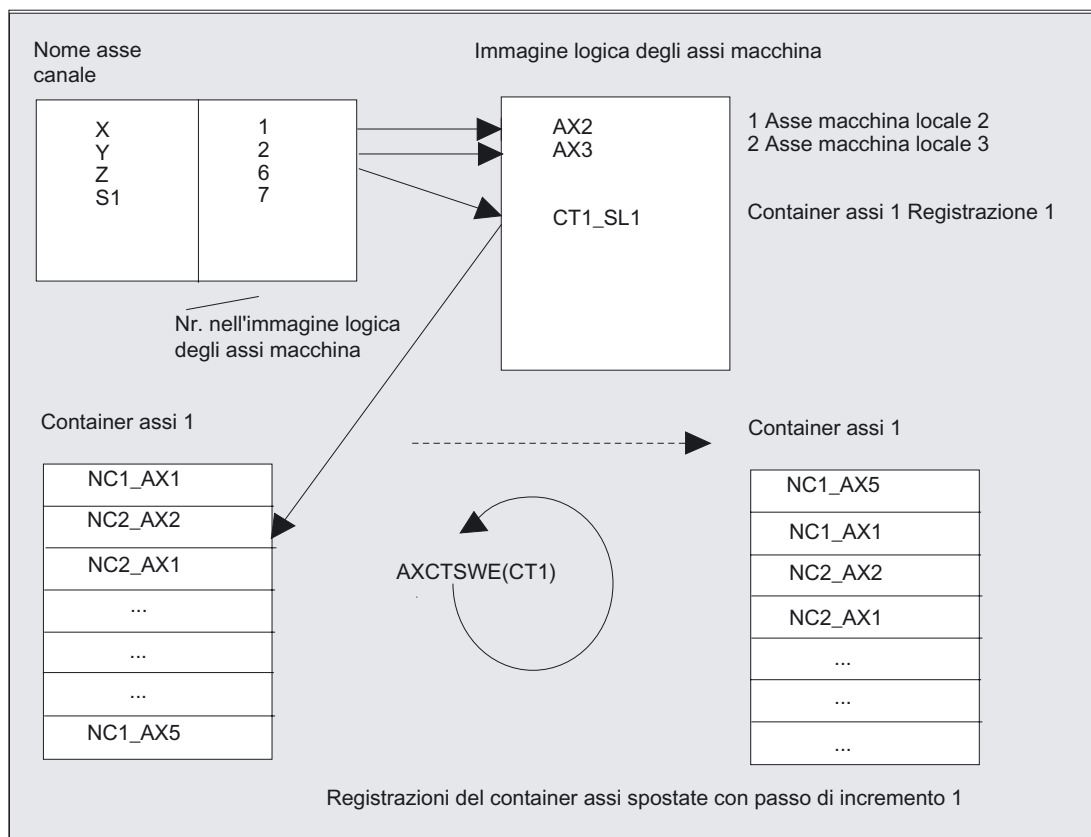
I container assi con assi di collegamento sono una risorsa operativa valida per ogni NCU (NCU globale), che viene coordinata dal controllo numerico. Sono possibili anche i container assi in cui vengono gestiti esclusivamente gli assi locali.

Per informazioni dettagliate sulla progettazione di container assi vedere /FB2/ Manuale delle funzioni di ampliamento; Diversi pannelli operativi e NCU (B3).

Criteri di abilitazione

AXCTSWE ()

Ogni canale i cui assi sono registrati nel contenitore specificato dà l'**abilitazione ad una rotazione del container** (enable), una volta terminata la lavorazione della posizione/stazione. Se nel controllo numerico sono soddisfatte le abilitazioni di **tutti** i canali per gli assi del container, avviene la rotazione del container con l'entità di incremento indicata nel dato macchina.



Dopo la rotazione del container assi di 1, all'asse canale Z dell'esempio precedente viene assegnato anziché l'asse AX1 su NCU1 l'asse AX5 su NCU1.

AXCTSWED ()

La variante di istruzione AXCTSWED(CT_i) può essere utilizzata per semplificare la messa in servizio. Il container assi ruota dell'ampiezza di incremento impostata mediante dato setting per effetto esclusivo del canale attivo. Il richiamo può essere utilizzato solo se gli altri canali che hanno assi nel contenitore si trovano in stato di **RESET**.

Dalla nuova assegnazione degli assi dopo una rotazione del container assi sono interessate **tutte le NCU** i cui canali fanno riferimento al container assi ruotato attraverso l'immagine logica degli assi di macchina.

Rotazione container assi con GET/GETD impliciti

Durante l'abilitazione di una rotazione container assi vengono assegnati al canale tutti gli assi del container assi assegnati al canale tramite GET o GETD. L'indicazione degli assi è di nuovo consentita solo dopo la rotazione del container assi

Costruttore della macchina

Questo comportamento può essere impostato con un bit per ogni dato macchina. Fare attenzione alle indicazioni del costruttore della macchina

Nota

La rotazione del container assi con GET/GETD implicito **non** può essere impiegata per un asse che si trova nello stato elaborazione principale asse, ad es. per un asse PLC, in quanto questo asse dovrebbe uscire da questo stato per la rotazione del container asse.

13.10 Tempo di esecuzione del programma/contatore pezzi

13.10.1 Informazioni generali

A supporto dell'addetto alla macchina utensile vengono messe a disposizione informazioni sul tempo di elaborazione del programma.

Queste informazioni sono specificate nei corrispondenti dati macchina e possono essere elaborate come variabile di sistema nel programma NC e/o PLC. Le informazioni restano disponibili anche per l'HMI sull'interfaccia del pannello operativo.

13.10.2 Tempo di esecuzione del programma

Funzione

Con questa funzione vengono messi a disposizione, come variabili di sistema, i temporizzatori che possono essere usati per la sorveglianza dei processi tecnologici.

Per questi temporizzatori esistono solo diritti di lettura. HMI può accedervi in lettura in qualsiasi momento.

Parametri

I due temporizzatori seguenti sono definiti come variabile di sistema specifica dell'NCK e sono sempre attivi.

Variabili di sistema

\$AN_SETUP_TIME	Tempo intercorso dall'ultimo setup in minuti; viene azzerato con SETUP
\$AN_POWERON_TIME	Tempo intercorso dall'ultimo power on in minuti; viene azzerato con POWER ON

I tre temporizzatori seguenti sono definiti come variabile di sistema specifica per canale e possono essere attivati mediante il dato macchina.

\$AC_OPERATING_TIME	Tempo di elaborazione totale dei programmi NC nel modo operativo automatico in secondi
\$AC_CYCLE_TIME	Tempo di elaborazione del programma NC selezionato in secondi
\$AC_CUTTING_TIME	Tempo di intervento dell'utensile in secondi
\$MC_RUNTIMER_MODE	Tempo di intervento dell'utensile in secondi

Nota

Tutti i temporizzatori vengono azzerati con valori predefiniti all'avvio del controllo numerico e possono essere letti indipendentemente dalla loro attivazione.

Esempio

```

1. Attivazione della misura del tempo di esecuzione per il programma NC attivo,
   ma non quando sono attivi l'avanzamento per ciclo di prova e il test del programma:
$MC_PROCESSTIMER_MODE = 'H2'

2. Attivazione della misura per il tempo di intervento dell'utensile,
   anche quando sono attivi l'avanzamento per ciclo di prova e il test del programma:
$MC_PROCESSTIMER_MODE = 'H34'

3. Attivazione della misura per tutto il tempo di esecuzione e di intervento
   dell'utensile,
   anche durante il test del programma:
$MC_PROCESSTIMER_MODE = 'H25'

```

13.10.3 Funzione di conteggio dei pezzi**Funzione**

Con la funzione "Contapezzi" vengono attivati i contatori, utilizzabili ad esempio per il conteggio dei pezzi internamente al controllo numerico. Questi contatori esistono come variabili di sistema specifiche per canale con accesso in scrittura e lettura nel campo di valori da 0 a 999 999 999.

Con i dati macchina si può agire sull'attivazione dei contatori, sull'istante di azzeramento e sull'algoritmo di conteggio.

Parametri

Sono disponibili i seguenti contatori:

Variabile di sistema

\$AC_REQUIRED_PARTS	Numero di pezzi necessari (val. nom. pezzi). In questo contatore si può definire il numero di pezzi al cui raggiungimento viene azzerato il numero attuale di pezzi \$AC_ACTUAL_PARTS. Con un dato macchina è possibile attivare la generazione dell'allarme di visualizzazione "Nr. pezzi di riferimento raggiunto" e del segnale VDI del canale "Nr. pezzi di riferimento raggiunto".

13.10 Tempo di esecuzione del programma/contatore pezzi

\$AC_TOTAL_PARTS	Numero dei pezzi complessivamente prodotti (valore reale totale) Il contatore registra il numero di pezzi prodotti dallo start. Il contatore viene azzerato automaticamente con valori predefiniti solo all'avvio del controllo numerico.
\$AC_ACTUAL_PARTS	Numero di pezzi attuali (pezzi reali). in questo contatore si registra il numero di pezzi prodotti dallo start. Al raggiungimento del numero nominale di pezzi (\$AC_REQUIRED_PARTS) il contatore viene automaticamente azzerato (a condizione che \$AC_REQUIRED_PARTS sia diverso da 0).
\$AC_SPECIAL_PARTS	Numero dei pezzi definito dall'utente Questo contatore permette all'utente di eseguire un conteggio dei pezzi definito in base a criteri personali. Può essere definita un'emissione di allarme in caso di identità con \$AC_REQUIRED_PARTS (val. nom. pezzi). L'azzeramento di questo contatore deve essere effettuato dall'utente stesso.

Nota

La funzione "Contapezzi" è indipendente dalle funzioni di gestione utensile. Tutti i contatori possono essere letti e scritti dall'HMI.

Tutti i contatori vengono azzerati con valori predefiniti all'avvio del controllo numerico e possono essere letti/scritti indipendentemente dalla loro attivazione.

Esempio

Attivazione del contapezzi \$AC_REQUIRED_PARTS: \$MC_PART_COUNTER='H3'	\$AC_REQUIRED_PARTS è attivo, di visualizzazione dell'allarme con \$AC_REQUIRED_PARTS == \$AC_SPECIAL_PARTS
Attivazione del contapezzi \$AC_TOTAL_PARTS: \$MC_PART_COUNTER='H10' \$MC_PART_COUNTER_MCODE[0]=80	\$AC_TOTAL_PARTS è attivo, ad ogni M02 il contatore aumenta del valore 1, \$MC_PART_COUNTER_MCODE[0] non ha alcuna influenza
Attivazione del contapezzi \$AC_ACTUAL_PARTS: \$MC_PART_COUNTER='H300' \$MC_PART_COUNTER_MCODE[1]=17	\$AC_TOTAL_PARTS è attivo, ad ogni M17 il contatore aumenta del valore 1
Attivazione del contapezzi \$AC_SPECIAL_PARTS: \$MC_PART_COUNTER='H3000' \$MC_PART_COUNTER_MCODE[2]=77	\$AC_SPECIAL_PARTS è attivo, ad ogni M77 il contatore aumenta del valore 1
Disattivazione del contapezzi \$AC_ACTUAL_PARTS: \$MC_PART_COUNTER='H200' \$MC_PART_COUNTER_MCODE[1]=50	\$AC_TOTAL_PARTS non è attivo, il resto non è significativo
Attivazione di tutti i contatori, esempio 1-4:	

13.11 Istruzione per il richiamo interattivo delle maschere dal partprogram (MMC):

```
$MC_PART_COUNTER = 'H3313'
$MC_PART_COUNTER_MCODE[0] = 80
$MC_PART_COUNTER_MCODE[1] = 17
$MC_PART_COUNTER_MCODE[2] = 77
```

\$AC_REQUIRED_PARTS è attivo
Visualizzazione dell'allarme per
\$AC_REQUIRED_PARTS ==
\$AC_SPECIAL_PARTS
\$AC_TOTAL_PARTS è attivo, ad ogni M02
il contatore incrementa del valore 1
\$MC_PART_COUNTER_MCODE[0] non ha
alcuna influenza
\$AC_ACTUAL_PARTS è attivo, ad ogni
M17 il contatore incrementa del valore 1
\$AC_SPECIAL_PARTS è attivo, ad ogni
M77 il contatore incrementa del valore 1

13.11 Istruzione per il richiamo interattivo delle maschere dal partprogram (MMC):

Funzione

Con le istruzioni MMC si possono visualizzare dal partprogram sull'interfaccia HMI delle finestre di dialogo interattive definite dall'utente (maschere interattive).

La struttura delle finestre di dialogo si definisce con una progettazione esclusivamente testuale (file COM nella directory cicli), il software di sistema HMI resta invariato.

Le finestre interattive definite dall'utente non si possono richiamare contemporaneamente in canali diversi.

Programmazione

```
MMC(CYCLES, PICTURE_ON, T_SK.COM, BILD, MGUD.DEF, BILD_3.AWB,
TEST_1, A1, "S")
```

Per informazioni dettagliate sulla programmazione dell'istruzione MMC (incl. esempi di programmazione) consultare /IAM/ nei manuali AE1, BE1, HE1, IM2, IM4 e IM5 in base al software MMC utilizzato.

Parametri

MMC

Dal partprogram richiamare in modo interattivo la maschera di dialogo sull'HMI.

CYCLES

Settore operativo nel quale vengono eseguiti i dialoghi utente progettati.

PICTURE_ON oppure PICTURE_OFF

Istruzione: selezione/deselezione maschera

T_SK.COM

File Com: Nome del file per le maschere interattive (cicli utente). Nel file si definisce la struttura delle maschere. Nella maschera di dialogo si possono visualizzare variabili utente e/o testi di commento.

MASCHERA	Nome della maschera di dialogo: le singole maschere vengono selezionate attraverso il relativo nome.
MGUD.DEF	File di definizione dei dati utente, al quale si accede per la lettura/scrittura di variabili.
MASCHERA_3.AWB	File di grafica
TEST_1	Tempo di visualizzazione o variabile di conferma
A1	Variabili di testo..."
"S"	Modo di conferma: sincrono, conferma con il softkey "OK"

13.12 Influenza della gestione dei movimenti

13.12.1 Correzione percentuale dello strappo (JERKLIM)

Funzione

In alcune sequenze critiche del programma può essere necessario limitare il valore max consentito per lo strappo al fine di ridurre ad esempio le sollecitazioni meccaniche sulla macchina. Deve essere abilitato il modo di accelerazione SOFT. La funzione agisce solo per gli assi di contornitura.

Programmazione

JERKLIM [Asse] = ...

Parametri

JERKLIM	Modifica in percentuale del valore impostato per l'asse nel DM che riguarda il massimo strappo consentito
Asse	Asse di macchina per il quale è necessario adattare il valore limite dello strappo
Campo dei valori: 1 ... 200	100 corrisponde a: nessuna influenza sullo strappo. 100 è attivo dopo RESET e all'inizio partprogram.

Esempio

Nel modo operativo AUTOMATICO il valore limite antistress programmato per l'asse nel dato macchina viene limitato di una certa percentuale indicata.

N60 JERKLIM [X] = 75

Significa: che la slitta dell'asse in direzione X dovrà essere accelerata/rallentata in base a un valore antistress che corrisponde al 75% del valore predefinito per l'asse.

Nota

È presente un ulteriore esempio al capitolo "Correzione percentuale della velocità:" (VELOLIM)".

13.12.2 Correzione percentuale della velocità (VELOLIM)

Funzione

In alcune sequenze critiche del programma può essere necessario limitare i max. valori di velocità consentiti, per ridurre ad esempio le sollecitazioni meccaniche sulla macchina o per migliorare la qualità della lavorazione. La funzione agisce solo sugli assi di posizionamento e di contornitura.

Programmazione

VELOLIM[Asse] = ...

Parametri

VELOLIM	Modifica in percentuale del valore max di velocità impostato per l'asse nel DM
Asse	Asse di macchina per il quale è necessario adattare il valore limite di velocità
Campo dei valori: 1 ... 100	100 corrisponde a: nessuna influenza sulla velocità. 100 è attivo dopo RESET e all'inizio partprogram.

Esempio di VELOLIM

Nel modo operativo AUTOMATICO il valore limite programmato nel DM per la velocità dell'asse viene limitato di una certa percentuale indicata.

N70 VELOLIM[X]=80

Significa: la slitta dell'asse in direzione X dovrà essere mossa con una velocità pari solo all'80% del valore consentito per l'asse

13.13 Raggruppamento Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)
Esempio di VELOLIM e JERKLIM

```

N1000 G0 X0 Y0 F10000 SOFT G64
N1100 G1 X20 RNDM = 5 ACC[X] = 20
ACC[Y]=30
N1200 G1 Y20 VELOLIM[X]=5
JERKLIM[Y]=200
N1300 G1 X0 JERKLIM[X]=2
N1400 G1 Y0
M30

```

13.13 Raggruppamento Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Funzione

Prima della versione di SW 6.4 con l'accoppiamento Master/Slave si potevano accoppiare gli assi slave al relativo asse Master solo quando questi assi erano fermi.

L'ampliamento della versione di SW 6.5 consente l'accoppiamento e la separazione di mandrini in **rotazione**, comandati in velocità, e la progettazione dinamica.

Programmazione

MASLON(Slv1, Slv2, ...,)	
MASLOF(Slv1, Slv2, ...,)	
MASLDEF(Slv1, Slv2, ..., asse Master)	Ampliamento della progettazione dinamica
MASLDEL(Slv1, Slv2, ...,)	Ampliamento della progettazione dinamica
MASLOFS(Slv1, Slv2, ...,)	Ampliamento per il mandrino slave

Nota

Con le istruzioni MASLOF/MASLOFS lo stop preelaborazione implicito non è più necessario. In mancanza dello stop preelaborazione le variabili di sistema \$P per gli assi slave, fino a quando non avviene una nuova programmazione, non presentano valori aggiornati.

Parametri

Generalità

MASLON	Inserzione di un accoppiamento temporaneo
MASLOF	Disinserzione di un accoppiamento attivo. Per i mandrini occorre osservare gli ampliamenti.

Ampliamento della progettazione dinamica

MASLDEF	Definizione/variazione dell'accoppiamento definito dall'utente tramite dati macchina oppure dal partprogram.
MASLOFS	disinserzione dell'accoppiamento analogamente a MSLOF e frenatura automatica del mandrino slave.
MASLDEL	Interruzione di un collegamento Master/Slave e cancellazione della definizione del raggruppamento
Slv1, Slv2, ...	Assi slave che sono gestiti da un asse Master.
Asse Master	Asse che in una configurazione Master/Slave gestisce degli assi slave definiti.

Esempio di progettazione dinamica di un accoppiamento master/slave

Progettazione dinamica di un accoppiamento master/slave da partprogram:

L'asse principale dopo una rotazione del container assi deve diventare un asse master.

MASLDEF (AUX, S3)	;S3 Master per AUX
MASLON (AUX)	;accoppiamento On per AUX
M3=3 S3=4000	;rotazione in senso orario
MASLDEL (AUX)	;cancellazione della progettazione e
	;disattivazione dell'accoppiamento
AXCTSWE (CT1)	;rotazione del container assi

Esempio di accoppiamento tramite valore reale di un asse slave

Accoppiamento tramite valore reale di un asse slave allo stesso valore dell'asse master tramite PRESETON.

Con un'accoppiamento permanente Master/Slave, deve essere variato il valore reale dell'asse SLAVE tramite l'istruzione PRESETON.

N37262	;Disattivare brevemente l'accoppiamento
\$MA_MS_COUPLING_ALWAYS_ACTIVE [AX2] =0	;permanente
N37263 NEWCONF	
N37264 STOPRE	
MASLOF (Y1)	;accoppiamento temporaneo OFF
N5 PRESETON (Y1, 0, Z1, 0, B1, 0, C1, 0, U1, 0)	;impostazione del valore reale degli assi slave
	;non azzerati; dato che questi vengono attivati
	;con Power On.

13.13 Raggruppamento Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

```

N37262
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1
N37263 NEWCONF

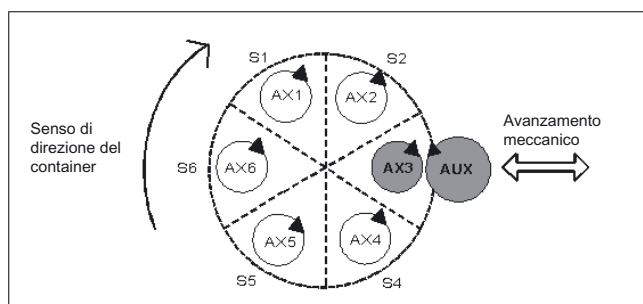
```

;Attivazione dell'accoppiamento permanente

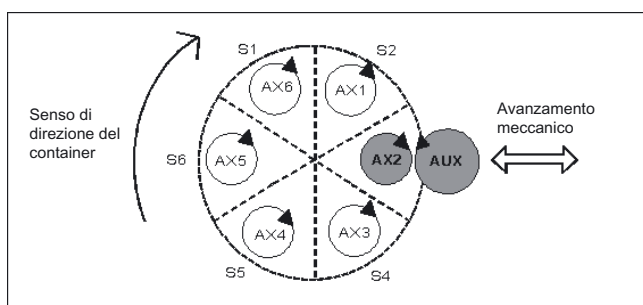
Esempio di sequenza di accoppiamento Posizione 3 / Container CT1

Per poter chiudere l'accoppiamento dopo la rotazione container con un altro mandrino si dovrà interrompere per prima cosa l'accoppiamento precedente, cancellare la progettazione e progettare il nuovo accoppiamento.

Situazione iniziale:



Dopo la rotazione di uno slot:

**Bibliografia:**

/FB2/ Manuale di guida alle funzioni, Funzioni di ampliamento; Vari pannelli operativi e NCU (B3), capitolo "Container assi".

Descrizione**Generalità****MASLOF**

Nei mandrini comandati in velocità l'istruzione viene eseguita immediatamente. I mandrini slave che si trovano in rotazione mantengono la loro velocità fino a quando viene programmato un nuovo numero di giri.

Ampliamento della progettazione dinamica

MASLDEF	Definizione dinamica di un accoppiamento master/slave da partprogram: Prima la definizione avveniva esclusivamente tramite dati macchina.
MASLDEL	L'istruzione annulla l'assegnazione degli assi slave all'asse master e contemporaneamente, analogamente a MASLOF, interrompe l'accoppiamento in corso. Le definizioni master/slave impostate nei dati macchina vengono mantenute.
MASLOFS	L'istruzione MASLOFS può essere utilizzata per arrestare automaticamente i mandrini slave quando si interrompe l'accoppiamento. Con assi e mandrini nel modo posizionamento l'accoppiamento viene chiuso ed interrotto solo in condizioni di fermo.

Nota

Per l'asse slave è possibile sincronizzare il valore reale allo stesso valore del master con l'istruzione `PRESETON`. A questo scopo è necessario disattivare temporaneamente l'accoppiamento Master/Slave per settare con Power On il valore dell'asse non azzerato allo stesso valore dell'asse master. Dopodichè viene riattivato l'accoppiamento continuo.

L'accoppiamento continuo Master/Slave viene attivato con il DM 37262:
`MS_COUPLING_ALWAYS_ACTIVE = 1` e non ha alcun effetto sulle istruzioni dell'accoppiamento temporaneo.

Comportamento dell'accoppiamento dal SW 6.5

Per i mandrini controllati in velocità, il comportamento dell'accoppiamento di `MASLON`, `MASLOF`, `MASLOFS` e `MASLDEL` viene definito in modo esplicito con il DM 37263:
`MS_SPIND_COUPLING_MODE`.

Nell'impostazione di default con il DM 37263 = 0 l'accoppiamento e la disattivazione degli assi slave avvengono esclusivamente in condizione di fermo. `MASLOFS` corrisponde a `MASLOF`.

Con il DM 37263 = 1 l'istruzione di accoppiamento viene eseguita immediatamente e anche nel movimento. L'accoppiamento viene subito inserito con `MASLON` e subito disinserito con `MASLOFS` oppure `MASLOF`. I mandrini slave in rotazione fino a questo momento, con `MASLOFS` vengono automaticamente arrestati e con `MASLOF` mantengono le loro velocità fino ad una nuova programmazione.

Programmi di sgrossatura personalizzati

14.1 Funzioni di supporto per l'asportazione

Funzione

Per la sgrossatura sono disponibili cicli di lavoro già collaudati. Inoltre, ogni utente può approntare programmi di sgrossatura personalizzati servendosi delle seguenti funzioni.

Nota

Queste funzioni possono essere usate per tutti i tipi di lavorazione e non solo per la sgrossatura.

Premessa

Prima del richiamo di CONTDCON o CONTDCON bisogna

- raggiungere un punto di partenza che consente una lavorazione libera da collisioni,
- annullare la correzione raggio utensile con G40.

Programmazione

CONTPRON

oppure

CONTDCON

con

INTERSEC

oppure

ISPOINTS

oppure

EXECTAB

oppure

CALCDAT

Ultimare la preparazione del profilo

EXECUTE (ERRORE)

Parametri

CONTDCON	Attivazione della decodifica del profilo in forma tabellare (6 colonne)
CONTPRON	Attivazione della preparazione del profilo in forma tabellare (11 colonne)
INTERSEC	Rilevamento del punto d'intersezione fra due elementi del profilo. (Solo per tabelle generate con CONTPRON).
ISPOINTS	Rilevamento dei possibili punti d'intersezione fra due elementi del profilo. (Solo per tabelle generate con CONTPRON).
EXECTAB	Elaborazione blocco-blocco di elementi del profilo di una tabella (Solo per tabelle generate con CONTPRON).
CALCDAT	Calcolo del raggio e del centro del cerchio sulla base di 3 o 4 punti noti del cerchio.
EXECUTE	Ultimare la preparazione del profilo
ERRORE	Variabile per i messaggi d'errore, Tipo INT 1 = errore; 0 = nessun errore

Con EXECUTE viene disattivata la preparazione del profilo e contemporaneamente si ha il passaggio al normale modo di lavorazione.

Esempio:

```
N30 CONTPRON(...)
N40 G1 X... Z...
N50 ...
N100 EXECUTE(...)
```

14.2 Preparazione del profilo (CONTPRON)

Funzione

I blocchi eseguiti dopo CONTPRON descrivono il profilo che viene preparato. I blocchi non vengono elaborati bensì depositati nella tabella dei profili. Ogni elemento del profilo corrisponde ad una riga nel campo bidimensionale della tabella del profilo. Il numero determinato delle passate sottosquadra viene restituito.

Programmazione

```
CONTPRON (TABNAME, TIPO_LAVORAZ, NN, MODE)
```

Conclusione della preparazione del profilo e contemporaneamente ritorno al modo di lavorazione normale:

```
EXECUTE (ERRORE)
```

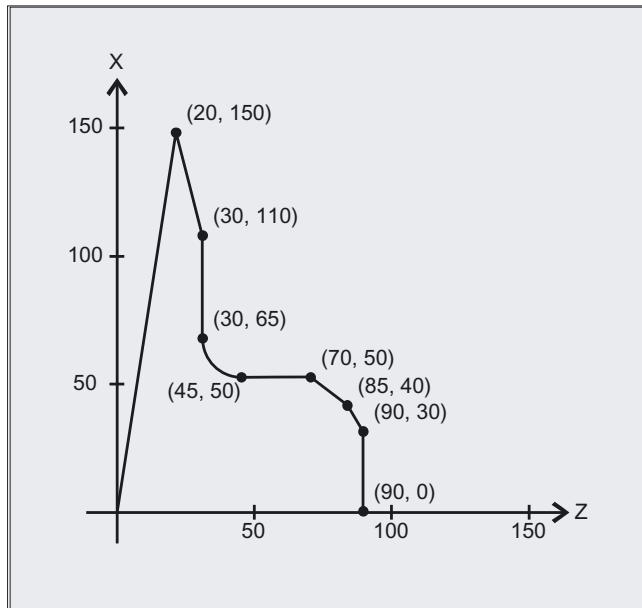

Parametri

CONTPRON	Attivazione della preparazione del profilo
TABNAME	Nome della tabella del profilo
TIPO_LAVORAZ	Parametri per il tipo di lavorazione: "G": tornitura cilindrica: Lavorazione interna "L": tornitura cilindrica: Lavorazione esterna "N": tornitura radiale: Lavorazione interna "P": tornitura radiale: Lavorazione esterna
NN	Numero dei sottosquadra nella variabile di risultato del tipo INT
MODE	Direzione di lavorazione, tipo INT 0 = preparazione del profilo in avanti (valore di default) 1 = preparazione del profilo nelle due direzioni

Esempio 1: Creazione di una tabella di curve

Creazione di una tabella di profilo con

- nome KTAB,
- max. 30 elementi di profilo (cerchi, rette),
- una variabile per il numero di elementi con taglio in ombra,
- una variabile per i messaggi di errore.



Partprogram NC

N10 DEF REAL KTAB[30,11]	;Tabella dei profilo di nome KTAB e ;per esempio con max. 30 elementi di profilo ;Il valore di parametro 11 è una grandezza definita
N20 DEF INT NR_ELEM_OMBRA	;Variabile per la quantità di ;elementi in ombra di nome ANZHINT
N30 DEF INT ERRORE	;Variabile per la conferma ;0=nessun errore, 1=errore
N40 G18	
N50 CONTPRON (KTAB, "G", NR_ELEM_OMBRA)	;Richiamo della preparazione del profilo
N60 G1 X150 Z20	;N60 ... N120 descrizione del profilo
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(ERRORE)	;Terminare la compilazione della tabella, ;commutare al normale funzionamento del ;programma
N140 ...	;Ulteriore elaborazione della tabella

Tabella relativa KTAB

Indice Riga	Colonna									
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
7	7	11	0	0	20	150	0	82.40535663	0	0
0	2	11	20	150	30	110	-1111	104.0362435	0	0
1	3	11	30	110	30	65	0	90	0	0
2	4	13	30	65	45	50	0	180	45	65
3	5	11	45	50	70	50	0	0	0	0
4	6	11	70	50	85	40	0	146.3099325	0	0
5	7	11	85	40	90	30	0	116.5650512	0	0
6	0	11	90	30	90	0	0	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

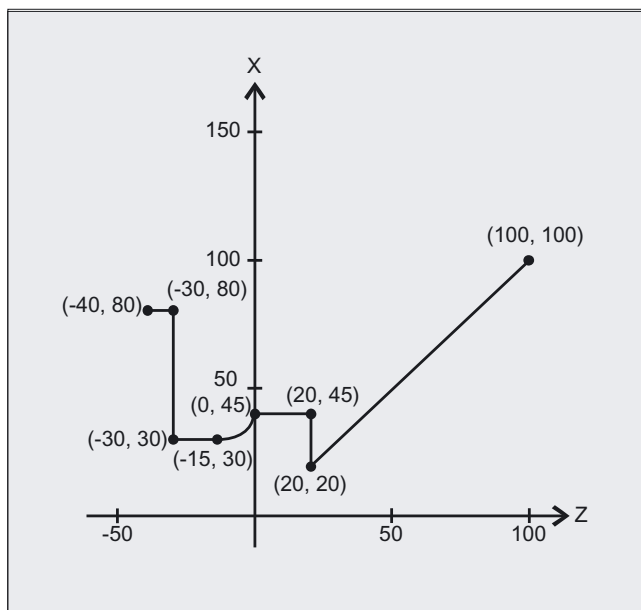
Descrizione del contenuto delle colonne

- (0) Puntatore all'elemento di profilo successivo (al numero di riga dello stesso)
- (1) puntatore all'elemento di profilo precedente
- (2) Codifica della modalità del profilo per il movimento
Valori possibili per X = abc
a = 10² G90 = 0 G91 = 1
b = 10¹ G70 = 0 G71 = 1
c = 10⁰ G0 = 0 G1 = 1 G2 = 2 G3 = 3
- (3), (4) Punto iniziale degli elementi del profilo
(3) = ascisse, (4) = ordinate nel piano attuale
- (5), (6) Punto finale degli elementi del profilo
(5) = ascisse, (6) = ordinate nel piano attuale
- (7) Indicatore max./min. identifica la massima e la minima locale nel profilo
- (8) Valore massimo tra l'elemento del profilo e le ascisse (per lavorazione longitudinale) oppure le ordinate (per lavorazione radiale). L'angolo dipende dal tipo di lavorazione programmato.
- (9), (10) Coordinate del centro dell'elemento del profilo quando si tratta di un cerchio.
(9) = ascisse, (10) = ordinate

Esempio 2: Creazione di una tabella di curve

Creazione di una tabella di profilo con

- nome KTAB,
- max. 92 elementi di profilo (cerchi, rette),
- Modo operativo: tornitura longitudinale, lavorazione esterna,
- Preparazione avanti e indietro.

**Partprogram NC**

N10 DEF REAL KTAB[92,11]	;Tabella dei profili con nome KTAB e ;per esempio con max. 92 elementi di profilo ;Il valore di parametro 11 è una grandezza definita
N20 CHAR BT="L"	;Modo operativo per CONTPRON: ;tornitura longitudinale, lavorazione esterna
N30 DEF INT HE=0	;Numero di sottosquadra=0
N40 DEF INT MODE=1	;Preparazione avanti e indietro
N50 DEF INT ERR=0	;Segnalazione di errore
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON (KTAB, BT, HE, MODE)	;Richiamo della preparazione del profilo
N110 G1 G90 Z20 X20	
N120 X45	
N130 Z0	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)	
N150 G1 Z-30	
N160 X80	
N170 Z-40	
N180 EXECUTE(ERR)	;Terminare la compilazione della tabella, ;commutare al normale funzionamento del ;programma
...	

Tabella relativa KTAB

Al termine della preparazione il profilo è a disposizione in entrambe le direzioni.

Indice	Colonna										
Riga	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
0	6 ¹⁾	7 ²⁾	11	100	100	20	20	0	45	0	0
1	0 ³⁾	2	11	20	20	20	45	-3	90	0	0
2	1	3	11	20	45	0	45	0	0	0	0
3	2	4	12	0	45	-15	30	5	90	-15	45
4	3	5	11	-15	30	-30	30	0	0	0	0
5	4	7	11	-30	30	-30	45	-1111	90	0	0
6	7	0 ⁴⁾	11	-30	80	-40	80	0	0	0	0
7	5	6	11	-30	45	-30	80	0	90	0	0
8	1 ⁵⁾	2 ⁶⁾	0	0	0	0	0	0	0	0	0
	...										
83	84	0 ⁷⁾	11	20	45	20	80	0	90	0	0
84	90	83	11	20	20	20	45	-1111	90	0	0
85	0 ⁸⁾	86	11	-40	80	-30	80	0	0	0	0
86	85	87	11	-30	80	-30	30	88	90	0	0
87	86	88	11	-30	30	-15	30	0	0	0	0
88	87	89	13	-15	30	0	45	-90	90	-15	45
89	88	90	11	0	45	20	45	0	0	0	0
90	89	84	11	20	45	20	20	84	90	0	0
91	83 ⁹⁾	85 ¹⁰⁾	11	20	20	100	100	0	45	0	0

Descrizione del contenuto delle colonne e note relative alle righe 0, 1, 6, 8, 83, 85 e 91

Valgono le descrizioni del contenuto delle colonne riportate nell'esempio 1.

Sempre nella riga 0 della tabella:

- 1) Precedente: la riga n contiene la fine profilo in avanti
- 2) Successivo: la riga n è la fine della tabella del profilo in avanti

Una volta all'interno degli elementi di profilo in avanti:

- 3) Precedente: inizio del profilo (in avanti)
- 4) Successivo: fine del profilo (in avanti)

Sempre sulla riga di fine della tabella del profilo (in avanti) +1:

- 5) Precedente: numero dei sottosquadri in avanti
- 6) Successivo: numero dei sottosquadri all'indietro

Una volta all'interno degli elementi di profilo all'indietro:

- 7) Successivo: fine del profilo (all'indietro)
- 8) Precedente: inizio del profilo (all'indietro)

Sempre nell'ultima riga della tabella:

- 9) Precedente: la riga n è l'inizio della tabella del profilo (all'indietro)
- 10) Successivo: la riga n contiene l'inizio del profilo (all'indietro)

Comandi di movimento consentiti, sistema di coordinate

Per la programmazione del profilo sono consentite le seguenti istruzioni G:

Gruppo-G 1: G0, G1, G2, G3

oltre a raccordi e smussi.

È possibile la programmazione del cerchio mediante CIP e CT.

Le funzioni Spline, polinomio, filettatura generano degli errori.

La variazione del sistema di coordinate con l'inserimento di frame tra CONTPRON ed EXECUTE non è consentita. Lo stesso dicasi per un cambio tra G70 e G71 / G700 e G710.

Lo scambio di assi geometrici con GEOAX durante la preparazione della tabella di profilo genera un allarme.

Terminare la preparazione

Con il richiamo di EXECUTE (variabile), dopo la scrittura di un profilo, si ritorna alla normale esecuzione del programma con ultimazione della preparazione del profilo. La variabile quindi mostra:

1 = Errore

1 = nessun errore (il profilo è stato preparato correttamente).

Elementi in ombra

La descrizione del profilo dei singoli elementi con taglio in ombra può avvenire in un sottoprogramma o con blocchi singoli.

Sgrossatura indipendente dalla direzione del profilo programmata

La preparazione del profilo CONTPRON è stata ampliata in modo che, dopo il suo richiamo, la tabella del profilo è a disposizione indipendentemente dalla direzione programmata.

14.3 Decodifica del profilo (CONTDCON)

Funzione

I blocchi eseguiti dopo CONTPRON descrivono il profilo che viene preparato. I blocchi non vengono eseguiti, bensì inseriti in una tabella di profilo a 6 colonne. Ogni elemento del profilo corrisponde ad una riga della tabella. Conoscendo le regole di codifica indicate di seguito, è possibile creare applicazioni e programmi (ad es. cicli) in codice DIN, partendo dalle righe delle tabelle. Nelle righe delle tabelle con il numero 0, vengono memorizzati i dati dei punti di uscita.

Programmazione

CONTDCON (TABNAME, MODE)

Disattivazione della preparazione del profilo e contemporaneamente ritorno al modo di lavorazione normale:

EXECUTE (ERRORE)

Parametri

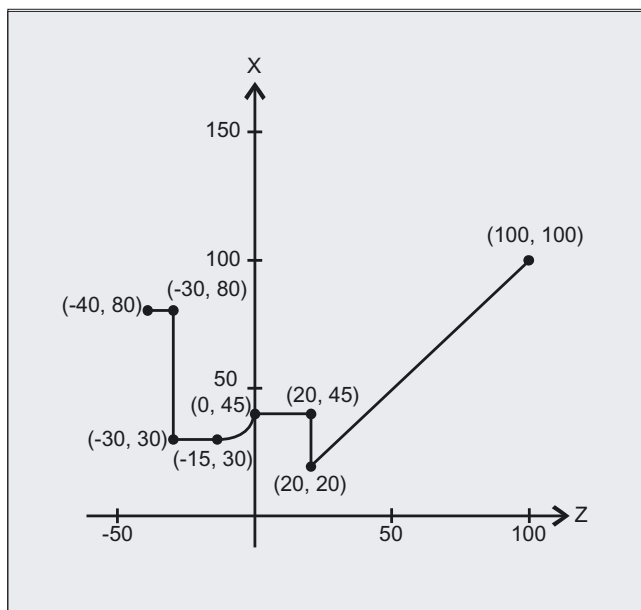
CONTDCON	Attivazione della preparazione del profilo
TABNAME	Nome della tabella del profilo
MODE	Direzione di lavorazione, tipo INT 0 = preparazione del profilo (valore di default) secondo la sequenza dei blocchi di profilo

I codici G ammessi per CONTDCON nella parte di programma in tabella, sono più completi della funzione CONTPRON. Con essi vengono memorizzati avanzamenti, tipi di avanzamento per tratto di profilo.

Esempio di creazione di tabella del profilo

Creazione di una tabella di profilo con

- nome KTAB,
- Elementi di profilo (cerchi, rette),
- Modo operativo: tornitura,
- Preparazione avanti.



Partprogram NC

N10 DEF REAL KTAB[9,6]	;Tabella dei profili con nome KTAB e 9 righe della ;tabella. ;Queste permettono 8 blocchi di profilo. Il valore del ;parametro 6 ;(numero di colonne in tabella) è una dimensione fissa
N20 DEF INT MODE = 0	;Valore standard 0: solo nella direzione di ;profilo programmata. Valore 1 non è ammesso.
N30 DEF INT ERROR = 0	;Segnalazione di errore
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	;Richiamo della preparazione del profilo ;MODE può essere tralasciato, vedere sopra.
N110 G1 Z20 X20 F200	;Descrizione del profilo
N120 G9 X45 F300	
N130 Z0 F400	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45) F100	
N150 G64 Z-30 F600	
N160 X80 F700	
N170 Z-40 F800	
N180 EXECUTE(ERROR)	;Terminare la compilazione della tabella, ;commutare al normale funzionamento del programma
...	

Tabella relativa KTAB

Indice colonne	0	1	2	3	4	5
Indice righe	Modo profilo	Punto finale delle ascisse	Punto finale delle ordinate	Centro delle ascisse	Centro delle ordinate	Avanzamento
0	30	100	100	0	0	7
1	11031	20	20	0	0	200
2	111031	20	45	0	0	300
3	11031	0	45	0	0	400
4	11032	-15	30	-15	45	100
5	11031	-30	30	0	0	600
6	11031	-30	80	0	0	700
7	11031	-40	80	0	0	800
8	0	0	0	0	0	0

Descrizione del contenuto delle colonne

Riga 0: Codifica per il **punto iniziale**:

Colonna 0:

10^0 (pos.unità): G0 = 0

10^1 (pos.decine): G70 = 0, G71 = 1, G700 = 2, G710 = 3

Colonna 1: Punto iniziale delle ascisse

Colonna 2: Punto iniziale delle ordinate

Colonna 3-4: 0

Colonna 5: Indice di riga dell'ultimo tratto di profilo nella tabella

Righe 1-n: Registrazioni dei **tratti di profilo**

Colonna 0:

10^0 (pos.unità): G0 = 0, G1 = 1, G2 = 2, G3 = 3

10^1 (pos.decine): G70 = 0, G71 = 1, G700 = 2, G710 = 3

10^2 (pos.centinaia): G90 = 0, G91 = 1

10^3 (pos.migliaia): G93 = 0, G94 = 1, G95 = 2, G96 = 3

10^4 (pos.decine di migliaia): G60 = 0, G44 = 1, G641 = 2, G642 = 3

10^5 (pos.centinaia di migliaia): G9 = 1

Colonna 1: Punto finale delle ascisse

Colonna 2: Punto finale delle ordinate

Colonna 3: Centro delle ascisse con interpolazione circolare

Colonna 4: Centro delle ordinate con interpolazione circolare

Colonna 5: Avanzamento

Comandi di movimento consentiti, sistema di coordinate

Per la programmazione del profilo sono consentiti i seguenti gruppi G e comandi G:

Gruppo-G 1:	G0, G1, G2, G3
Gruppo-G 10:	G60, G64, G641, G642
Gruppo-G 11:	G9
Gruppo-G 13:	G70, G71, G700, G710
Gruppo-G 14:	G90, G91
Gruppo-G 15:	G93, G94, G95, G96, G961

oltre a raccordi e smussi.

È possibile la programmazione del cerchio mediante CIP e CT.

Le funzioni Spline, polinomio, filettatura generano degli errori.

La modifica del sistema di coordinate con l'inserimento di frame tra CONTDCON ed EXECUTE non è consentita. Lo stesso dicasi per un cambio tra G70 e G71 / G700 e G710.

Lo scambio di assi geometrici con GEOAX durante la preparazione della tabella di profilo genera un allarme.

Terminare la preparazione

Con il richiamo di EXECUTE (ERRORE), dopo la scrittura di un profilo, si ritorna alla normale esecuzione del programma con ultimazione della preparazione della tabella. Si ha una segnalazione di risposta nelle variabili ERRORE corrispondenti:

0 = senza errori)

1 = errore

Comandi non ammessi, comandi di uscita errati, richiami di CONTDCON ripetuti senza EXECUTE(), troppo pochi blocchi di profilo o tabelle definite troppo piccole provocano ulteriori allarmi.

Sgrossatura nella direzione del profilo programmata

La tabella generata con CONTDCON è predisposta per la sgrossatura nella direzione di profilo programmata.

14.4 Punto di intersezione di due elementi del profilo (INTERSEC)

Funzione

INTERSEC rileva il punto di intersezione di due elementi del profilo normalizzati che fanno parte della tabella dei profili generata con CONTPRON .

Programmazione

```
ISPOINT = INTERSEC (TABNAME1 [n1], TABNAME2 [n2], ISCOORD, MODE)
```

Lo stato restituito di ISPOINT indica se un punto d'intersezione è disponibile (ISPOINT = TRUE) o meno (ISPOINT = FALSE).

Parametri

INTERSEC	La funzione di asportazione del tipo REAL per la determinazione di due elementi del profilo a partire dalla tabella dei profili creata con CONTPRON
ISPOINT	Variabile per lo stato del punto di intersezione del tipo BOOL: TRUE: Punto di intersezione trovato FALSE: punto di intersezione non trovato
TABNAME1 [n1]	Nome della tabella e n1. elemento del profilo della prima tabella
TABNAME2 [n2]	Nome della tabella e n2. elemento del profilo della seconda tabella
ISCOORD	Coordinate del punto di interfaccia nel piano attivo G17 - G19
MODE	Tipo di lavorazione: Mode = 0 (valore di default) o Mode = 1 (ampliamento) 0 = calcolo del punto di intersezione nel piano attivo con il parametro 2 1 = calcolo del punto di intersezione indipendentemente dal piano trasferito
G17 - G19	Piano trasferito della tabella dei profili all'attivazione di CONTPRON

Nota

Prima di essere utilizzate, le variabili devono essere definite.

Il trasferimento dei profili richiede il rispetto dei valori definiti con CONTPRON:

Parametro 2	Codifica della modalità del profilo per il movimento
Parametro 3	Ascisse del punto iniziale del profilo
Parametro 4	Ordinata del punto iniziale del profilo
Parametro 5	Ascisse del punto finale del profilo
Parametro 6	Ordinata del punto finale del profilo
Parametro 9	Coordinate del centro per l'ascissa (solo in caso di profilo circolare)
Parametro 10	Coordinate del centro per l'ordinata (solo in caso di profilo circolare)

Esempio

Determinare il punto di intersezione dell'elemento di profilo 3 della tabella TABNAME1 con l'elemento di profilo 7 della tabella TABNAME2. Le coordinate del punto di intersezione nel piano attivo vengono inserite in INTERSEZIONE (1. elemento = ascissa, 2. elemento = ordinata) Se non esiste alcun punto di intersezione, si ha un salto a KEINSCH (nessun punto di intersezione trovato).

DEF REAL TABNAME1 [12, 11]	:Tabella del profilo 1
DEF REAL TABNAME2 [10, 11]	:Tabella del profilo 2
DEF REAL ISCOORD [2]	;Coordinate del punto di intersezione con ;ISPOINT = 1
DEF BOOL ISPOINT	;Variabile per lo stato del punto di intersezione
DEF INT MODE	;Definizione del tipo di lavorazione.
...	
MODE = 1	;Calcolo indipendentemente dal piano attivo
N10 ISPOINT=INTERSEC (TABNAME1 [16,11] ,TABNAME2 [3,11] ,ISCOORD, MODE)	;Richiamo punto di intersezione degli elementi del ;profilo
N20 IF ISPOINT==FALSE GOTOF NO_INTERS	;Salto a NO_INTERS
...	

14.5 Esecuzione di un elemento di profilo dalla tabella (EXECTAB)

Funzione

con il comando EXECTAB è possibile eseguire blocco a blocco gli elementi di profilo di una tabella, generati ad esempio con il comando CONTPRON.

Programmazione

```
EXECTAB (TABNAME [n] )
```

Parametri

TABNAME [n]	Nome della tabella con numero n dell'elemento

Esempio

Con il sottoprogramma EXECTAB vengono eseguiti blocco a blocco gli elementi di profilo della tabella KTAB. Al richiamo vengono trasmessi in successione gli elementi 0 ... 2.

N10 EXECTAB (KTAB [0])	;Eeguire l'elemento 0 della tabella KTAB
N20 EXECTAB (KTAB [1])	;Eeguire l'elemento 1 della tabella KTAB
N30 EXECTAB (KTAB [2])	;Eeguire l'elemento 2 della tabella KTAB

14.6 Calcolo dei dati del cerchio (CALCDAT)

Funzione

Calcolo del raggio e delle coordinate del centro del cerchio sulla base di 3 o 4 punti noti del cerchio. I punti impostati devono essere differenti tra di loro. Con 4 punti non giacenti esattamente sullo stesso cerchio, viene calcolato un valore medio sia per il centro che per il raggio.

Programmazione

```
VARIB = CALCDAT (PKT [n,2] , ANZ, ERG)
```

Nota

Prima di essere utilizzate, le variabili devono essere definite.

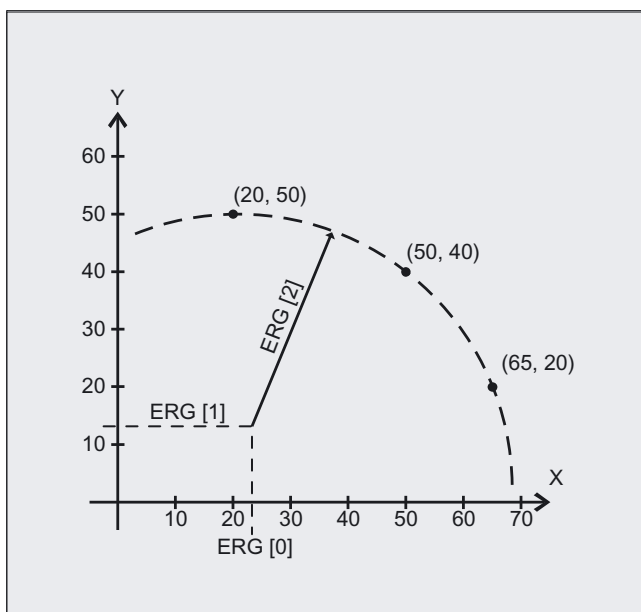
Lo stato riportato indica se i tre o quattro punti noti si trovano esattamente su un cerchio (VARIB = TRUE) o su nessun cerchio (VARIB = FALSE).

Parametri

CALCDAT	Calcolo del raggio e del centro del cerchio sulla base di 3 o 4 punti noti del cerchio.
VARIB	Variabile di stato TRUE = cerchio, FALSE = nessun cerchio
PKT [n, 2]	Punti per il calcolo n = numero dei punti (3 opp. 4); 2 = impostazione di coordinate in 2 punti
NR	Numero dei punti utilizzati per il calcolo: 3 oppure 4
ERG [3]	Variabile per il risultato: indicazione delle coordinate del centro del cerchio e del raggio; 0 = ascissa, 1 = ordinata del centro; 2 = raggio

Esempio

Si deve determinare se i tre punti si trovano sullo stesso arco di cerchio.



```
N10 DEF REAL PKT[3, 2] = (20, 50, 50, 40, 65, 20)
```

;Definizione dei punti

```
N20 DEF REAL ERG[3]
```

;Risultato

```
N30 DEF BOOL STATUS
```

;Variabile di stato

```
N40 STATUS = CALCDAT(PKT, 3, ERG)
```

;Richiamo dei dati del cerchio
;determinati

```
N50 IF STATUS == FALSE GOTO ERROR
```

;Salto all'errore

Tabelle

15.1 Lista istruzioni

Legenda:

- ¹ Impostazione standard a inizio programma (nello stato di fornitura del controllo numerico, salvo diversa programmazione).
- ² La numerazione dei gruppi corrisponde alla tabella "Lista delle funzioni G/condizioni di percorso".
- ³ Punti finali assoluti: modale (m)
punti finali incrementali: blocco per blocco(s)
Altrimenti: m/s in funzione di quanto definito dalla sintassi della funzione G
- ⁴ Come centri del cerchio valgono i parametri di interpolazione (IPO) incrementali. Con AC essi possono essere programmati in assoluto. Per altri significati (ad es. passo della filettatura) la modifica dell'indirizzo viene ignorata.
- ⁵ La parola chiave non vale per il SINUMERIK 810D.
- ⁵ La parola chiave non vale per il SINUMERIK 810D/NCU571.
- ⁷ La parola chiave vale solo per il SINUMERIK FM-NC.
- ⁸ L'utente OEM può inserire due ulteriori tipi di interpolazione. I nomi possono essere modificati dall'utente OEM.
- ⁹ Per queste funzioni non è consentito utilizzare la forma di scrittura estesa per l'indirizzo.

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
:	Numero di blocco - blocco principale (vedi N)	0 ... 99 999 999 solo numeri interi, senza segno	particolari identificazioni di blocchi - al posto di N... ; questo blocco dovrebbe contenere tutte le istruzioni per una sezione completa di lavorazione seguente	per es.: 20		
A	Asse	Real			m/s	
A2 ⁵	Orientamento utensile: angolo di Eulero	Real			s	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
A3 ⁵	Orientamento utensile: Componente vettore direzionale	Real			s	
A4 ⁵	Orientamento utensile per inizio blocco	Real			s	
A5 ⁵	Orientamento utensile per fine blocco: Componente vettore normale	Real			s	
ABS	Valore assoluto	Real				
AC	Indicazione di quota assoluta	0, ..., 359.9999°		X=AC(100)	s	
ACC ⁵	Accelerazione assiale (acceleration axial)	Real, senza segno			m	
ACCLIMA ⁵	Riduzione o aumento dell'accelerazione massima assiale (acceleration axial)	0, ..., 200	Campo di validità 1 ... 200%	ACCLIMA[X]= ...[%]	m	
ACN	Quota assoluta per assi rotanti, raggiungimento posizione in direzione negativa			A=ACN(...) B=ACN(...) C=ACN(...)	s	
ACOS	Arco-coseno (funzione trigonometrica)	Real				
ACP	Quota assoluta per assi rotanti, raggiungimento posizione in direzione negativa			A=ACP(...) B=ACP(...) C=ACP(...)	s	
ADIS	Distanza di raccordo per funzioni vettoriali G1, G2, G3, ...	Real, senza segno			m	
ADISPOS	Distanza di raccordo per rapido G0	Real, senza segno			m	
ADISPOSA	Grandezza della finestra di tolleranza per IPOBRKA	Integer, Real		ADISPOSA=... oppure ADISPOSA(<Asse>[,REA L])	m	
ALF	Angolo di svincolo rapido (angle tilt fast)	Intero, senza segno			m	
AMIRROR	Specularità programmabile (additive mirror)			AMIRROR X0 Y0 Z0; ;blocco a sé stante	s	3
AND	AND logico					
ANG	Angolo del tratto di profilo	Real			s	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
AP	Angolo polare (angle polar)	0, ..., $\pm 360^\circ$			m/s	
APR	Lettura/visualizzazione protezione accesso (access protection read)	Intero, senza segno				
APW	Scrittura protezione accesso (access protection write)	Intero, senza segno				
AR	Angolo di apertura (angle circular)	0, ..., 360°			m/s	
AROT	Rotazione programmabile (additive rotation)	Rotazione di: 1° asse geometrico: $-180^\circ \dots +180^\circ$ 2° asse geometrico: $-90^\circ \dots +90^\circ$ 3° asse geometrico: $-180^\circ \dots +180^\circ$		AROT X... Y... Z... AROT RPL= ;blocco a sé stante	s	3
AROTS	Rotazioni programmabili dei frame con angoli nello spazio (additive rotation)			AROTS X... Y... AROTS Z... X... AROTS Y... Z... AROTS RPL= ;blocco a sé stante	s	3
AS	Definizione macro	String				
ASCALE	Fattore di scala programmabile (additive scale)			ASCALE X... Y... Z... ;blocco a sé stante	s	3
ASPLINE	Spline Akima				m	1
ATAN2	Arco-tangente 2	Real				
ATRANS	Traslazione additiva programmabile (additive translation)			ATRANS X... Y... Z... ;blocco a sé stante	s	3
AX	Indicatore assi variabile	Real			m/s	
AXCSWAP	Commutazione asse contenitore			AXCSWAP(CT _n , CT _{n+1} ,...)		25
AXCTSWE	Commutazione asse contenitore			AXCTSWE(CT _i)		25
AXIS	Tipo di dati: indicatore dell'asse		Può assumere il nome di un file.			
AXNAME	Converte la stringa in ingresso in indicatore dell'asse (get axname)	String	Se la stringa di ingresso non contiene alcun nome di asse valido, viene emesso un allarme.			

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
AXSTRING	Converte in stringa il numero mandrino (get string)	String	Può assumere il nome di un file.	AXSTRING[SPI(n)]		
AXTOCHAN	Richiesta di un asse per un determinato canale. È possibile dal programma NC e dalla sincronizzazione.			AXTOCHAN(asse, numero canale[,asse, numero canale[,...]])		
B	Asse	Real			m/s	
B_AND	AND a bit					
B_OR	OR a bit					
B_NOT	Negazione a bit					
B_XOR	OR esclusivo a bit					
B2 ⁵	Orientamento utensile: angolo di Eulero	Real			s	
B3 ⁵	Orientamento utensile: Componente vettore direzionale	Real			s	
B4 ⁵	Orientamento utensile per inizio blocco	Real			s	
B5 ⁵	Orientamento utensile per fine blocco: Componente vettore normale	Real			s	
BAUTO	Definizione della prima sezione Spline tramite i 3 punti seguenti (begin not a knot)				m	19
BLSYNC	L'elaborazione della routine di interrupt deve iniziare con il successivo cambio blocco					
BNAT ¹	Transizione naturale al primo blocco Spline (begin natural)				m	19
BOOL	Tipo di dati: Valori logici TRUE/FALSE oppure 1/0					
BOUND	Controlla che il valore rientri nel campo di valori definito. In caso di uguaglianza viene restituito il valore di prova.	Real	Var1: Varmin Var2: Varmax Var3: Varcheck	RetVar =		
BRISK ¹	Accelerazione vettoriale a gradino				m	21
BRISKA	Attivazione accelerazione vettoriale a gradini per gli assi programmati					
BSPLINE	B-Spline				m	1
BTAN	Passaggio tangenziale al primo blocco Spline (begin tangential)				m	19

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
C	Asse	Real			m/s	
C2 ⁵	Orientamento utensile: angolo di Eulero	Real			s	
C3 ⁵	Orientamento utensile: Componente vettore di direzione	Real			s	
C4 ⁵	Orientamento utensile per inizio blocco	Real			s	
C5 ⁵	Orientamento utensile per fine blocco; componente vettore normale	Real			s	
CAC	Accostamento assoluto ad una posizione (coded position: absolute coordinate)		Il valore codificato è l'indice della tabella; viene raggiunto il valore della tabella			
CACN	Il valore riportato nella tabella viene raggiunto in modo assoluto in direzione negativa (coded position absolute negative)		Consentito per la programmazione e di assi rotanti come assi di posizionamento.			
CACP	Il valore riportato nella tabella viene raggiunto in modo assoluto in direzione positiva (coded position absolute positive)					
CALCDAT	Raggio e centro di un cerchio calcolati da 3 o 4 punti (calculate circle data)	VAR Real [3]	I punti devono essere diversi l'uno dall'altro.			
CALL	Richiamo indiretto di sottoprogrammi			CALL PROGVAR		
CALLPATH	Ampliamento del percorso di ricerca programmabile in un richiamo di sottoprogramma		Per il file system NCK esistente è possibile programmare un percorso con l'istruzione CALLPATH.	CALLPATH (/_N_WKS_DIR/ _N_MYWPD/ identificatore_sottoprogramma_SPF)		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
CANCEL	Interruzione di un'azione sincrona modale	INT	Interruzione con l'ID specificato. Senza parametri: Tutte le azioni sincrone vengono deselezionate.			
CASE	Diramazione condizionata					
CDC	Accostamento diretto di una posizione (coded position: direct coordinate)		Vedere CAC.			
CDOF ¹	Sorveglianza collisioni OFF (collision detection OFF)				m	23
CDON	Sorveglianza collisioni ON (collision detection ON)				m	23
CDOF2	Sorveglianza collisioni OFF (collision detection OFF)		Solo per CUT3DC.		m	23
CFC ¹	Avanzamento costante sul profilo (constant feed at contour)				m	16
CFIN	Avanzamento costante solo per curvatura interna, non per curvatura esterna (constant feed at internal radius).				m	16
CFTCP	Avanzamento costante nel punto di riferimento del tagliente utensile (percorso riferito al centro) (constant feed in tool-center-point)				m	16
CHAN	Specifica del campo di validità dei dati		Presente una volta per canale.			
CHANDATA	Impostare numero di canale per accessi ai dati del canale	INT	Consentito solo nel blocco di inizializzazione.			
CHAR	Tipo di dati: carattere ASCII	0, ..., 255				
CHECKSUM	Forma il checksum su un campo come STRING di lunghezza predefinita	Lunghezza max. 32	Fornisce una catena di caratteri di 16 cifre esadecimali	ERROR=CHECKSUM		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
CHF CHR	Smusso; valore = lunghezza dello smusso Smusso; valore = larghezza dello smusso in direzione del movimento (chamfer)	Real, senza segno			s	
CHKDNO	Prova di univocità dei numeri D					
CIC	Accostamento incrementale di una posizione (coded position: incremental coordinate)		Vedere CAC.			
CIP	interpolazione circolare tramite punto intermedio			CIP X... Y... Z... I1=... J1=... K1=...	m	1
CLEARM	Ripristino di una o più label per la coordinazione dei canali	INT, 1 - n	Non influenza la lavorazione nel proprio canale			
CLRINT	Disattivazione Interrupt	INT	Parametri: numero di interrupt			
CMIRROR	specularità rispetto a un asse delle coordinate	FRAME				
COARSEA	Fine movimento al raggiungimento di "Arresto preciso"			COARSEA=... oppure COARSEA[n]=...	m	
COMPOF ^{1,6}	Compressore OFF				m	30
COMPON ⁶	Compressore ON				m	30
COMPCURV	Compressore ON: polinomi a curvatura continua				m	30
COMPCAD	Compressore ON: qualità superficiale ottimizzata programma CAD				m	30
CONTDCON	Codifica del profilo sotto forma di tabella ON					
CONTPRON	Attivazione preparazione del riferimento (contour preparation ON)					
COS	Coseno (funzione trigonometrica)	Real				
COUPDEF	Definizione raggruppamento ELG / raggruppamento mandrini sincroni (couple definition)	String	Comportamento del cambio blocco (SW): NOC: nessun controllo SW FINE / COARSE: SW con "sincronismo fine/grossolano" IPOSTOP: SW per conclusione del movimento sovrapposto in direzione del setpoint.	COUPDEF(FS, ...)		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
COUPDEL	Cancellazione raggruppamento ELG (couple delete)			COUPDEL(FS,LS)		
COUPOF	Raggruppamento ELG / coppia di mandrini sincroni OFF (couple OFF)			COUPOF(FS,LS, POS _{FS} , POS _{LS})		
COUPOFS	Disattivazione gruppo ELG / coppia mandrini sincroni con arresto del mandrino slave			COUPOFS(FS,LS, POS _{FS})		
COUPON	Raggruppamento ELG / coppia di mandrini sincroni ON (couple ON)			COUPON(FS,LS, POS _{FS})		
COUPONC	Attivazione gruppo ELG / coppia mandrini sincroni con programmazione precedente			COUPONC(FS,LS)		
COUPRES	Reset raggruppamento ELG (couple reset)		Valori programmati non validi; valori dati macchina validi.	COUPRES(FS,LS)		
CP	Movimento vettoriale (continuous path)				m	49
CPRECOF ^{1,6}	Precisione del profilo programmabile OFF (contour precision OFF)				m	39
CPRECON ⁶	Precisione del profilo programmabile ON (contour precision ON)				m	39
CPROT	Settore di protezione specifico per canale ON/OFF					
CPROTDEF	Definizione di un settore di protezione specifico per canale (channel specific protection area definition)					
CR	Raggio del cerchio (circle radius)	Real, senza segno			s	
CROT	Rotazione del sistema di coordinate attuale	FRAME	Nr.max. di parametri: 6			
CROTS	Rotazioni programmabili dei frame con angoli nello spazio (rotazione negli assi indicati)			CROTS X... Y... CROTS Z... X... CROTS Y... Z... CROTS RPL= ;blocco a sé stante	s	
CSCALE	Fattore di scala per più assi	FRAME	N. max. di parametri: 2 * numero assi _{max}			
CSPLINE	Spline cubico				m	1
CT	cerchio con passaggio tangenziale			CT X... Y.... Z...	m	1
CTAB	Calcola la posizione dell'asse slave in base alla posizione dell'asse master nella tabella di curve	Real	Se i parametri 4/5 non sono stati programmati: fattore di scala standard			

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
CTABDEF	Definizione della tabella ON					
CTABDEL	Cancellare tabella per curve					
CTABEND	Definizione della tabella OFF					
CTABEXISTS	Verifica la tabella di curve con il numero n		Parametro n			
CTABFNO	Numero di tabelle delle curve ancora possibili in memoria		memType			
CTABFPOL	Numero di polinomi ancora possibili in memoria		memType			
CTABFSEG	Numero dei segmenti di curve ancora possibili in memoria		memType			
CTABID	Restituisce il numero di tabella della n-ma tabella di curve		Parametro n e memType			
CTABINV	Calcola la posizione dell'asse master in base alla posizione dell'asse slave nella tabella di curve	Real	Vedere CTAB.			
CTABIS LOCK	Restituisce lo stato del blocco della tabelle curve con il numero n		Parametro n			
CTABLOCK	Attivare funzione di blocco contro cancellazione e sovrascrittura		Parametro n, m e memType.			
CTABMEMTYP	Restituisce la memoria nella quale è stata creata la tabella di curve con il numero n.		Parametro n			
CTABMPOL	Numero max. di polinomi ancora possibili in memoria		memType			
CTABMSEG	Numero max. di segmenti di curve ancora possibili in memoria		memType			
CTABNO	Numero delle tabelle di curve definite indipendentemente dal tipo di memoria		Nessuna indicazione dei parametri.			
CTABNOMEM	Numero delle tabelle di curve definite in memoria SRAM o DRAM		memType			
CTABPERIOD	Restituisce la periodicità della tabella con il numero n		Parametro n			
CTABPOL	Numero max. di polinomi già utilizzati in memoria		memType			
CTABPOLID	Numero dei polinomi di curve utilizzati dalla tabella di curve con il numero n		Parametro n			
CTABSEG	Numero dei segmenti di curve già utilizzati in memoria		memType			
CTABSEGID	Numero dei segmenti di curve utilizzati dalla tabella di curve con il numero n		Parametro n			
CTABSEV	Restituisce il valore finale dell'asse slave di un segmento della tabella di curve		Il segmento viene definito con LW.	R10 = CTABSEV(LW, n, gradi, F Asse, L Asse)		
CTABSSV	Restituisce il valore iniziale dell'asse slave di un segmento della tabella di curve		Il segmento viene definito con LW.	R10 = CTABSSV(LW, n, gradi, F Asse, L Asse)		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
CTABTEP	Restituisce il valore dell'asse master alla fine della tabella di curve		Valore master alla fine della tabella di curve.	R10 = CTABTEP(n, gradi, L Asse)		
CTABTEV	Restituisce il valore dell'asse slave alla fine della tabella di curve		Valore slave alla fine della tabella di curve.	R10 = CTABTEV(n, gradi, F Asse)		
CTABTMAX	Restituisce il valore massimo dell'asse slave della tabella di curve		Valore slave della tabella di curve.	R10 = CTABTMAX(n, F Asse)		
CTABTMIN	Restituisce il valore minimo dell'asse slave della tabella di curve		Valore slave della tabella di curve.	R10 = CTABTMIN(n, F Asse)		
CTABTSP	Restituisce il valore dell'asse master all'inizio della tabella di curve		Valore master all'inizio della tabella di curve.	R10 = CTABTSP(n, gradi, L Asse)		
CTABTSV	Restituisce il valore dell'asse slave all'inizio della tabella di curve		Valore slave all'inizio della tabella di curve.	R10 = CTABTSV(n, gradi, F Asse)		
CTABUNLOCK	Disattivare la funzione di blocco contro cancellazione e sovrascrittura		Parametro n, m e memType			
CTRANS	Spostamento origine per più assi	FRAME	max. 8 assi.			
CUT2D ¹	Correzione utensile 2D (Cutter compensation type 2dimensional)				m	22
CUT2DF	Correzione utensile 2D (Cutter compensation type 2dimensional frame). La correzione utensile è attiva in funzione del frame attuale (piano inclinato).				m	22
CUT3DC ⁵	Correzione utensile 3D fresatura periferica (Cutter compensation type 3dimensional circumference)				m	22
CUT3DCC ⁵	Correzione utensile 3D fresatura periferica con superficie di limitazione (Cutter compensation type 3dimensional circumference)				m	22
CUT3DCCD ⁵	Correzione utensile 3D fresatura periferica con superficie di limitazione e con utensile differenziale (Cutter compensation type 3dimensional circumference)				m	22
CUT3DF ⁵	Correzione utensile 3D fresatura frontale (Cutter compensation type 3dimensional face)				m	22
CUT3DFF ⁵	Correzione utensile 3D fresatura frontale con orientamento utensile costante in funzione del frame attivo (Cutter compensation type 3dimensional face frame)				m	22
CUT3DFS ⁵	Correzione utensile 3D fresatura frontale con orientamento utensile costante indipendente dal frame attivo (Cutter compensation type 3dimensional face)				m	22

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
CUTCONOF ¹	Correzione raggio costante OFF				m	40
CUTCONON	Correzione raggio costante ON				m	40
D	Numero correzione utensile	1, ..., 32 000	Contiene i dati di correzione per un determinato utensile T... ; D0 → valori di correzione per un utensile	D...		
DAC	Programmazione del diametro assoluta, specifica per asse e blocco-blocco		Programmazione del diametro	DAC(50)	s	
DC	Quota assoluta per assi rotanti, accostamento diretto alla posizione			A=DC(...) B=DC(...) C=DC(...) SPOS=DC(...)	s	
DEF	Definizione delle variabili	Intero, senza segno				
DEFAULT	Diramazione in CASE		Viene raggiunto se l'espressione non soddisfa alcuno dei valori indicati.			
DELAYFSTON	Definizione dell'inizio di un settore Stop Delay (DELAY Feed Stop ON)		Implicito con G331/G332 attivi.		m	
DELAYFSTOF	Definizione della fine di un settore Stop Delay (DELAY Feed Stop OF)				m	
DELDTG	Cancellazione del percorso residuo (Delete distance to go)					
DELETE	Cancellazione del file indicato. Il nome del file si può indicare con il percorso e l'identificatore del file.		Può cancellare tutti i file.			
DELT	Cancellazione utensile		Il numero del Duplo può essere omesso.			
DIACYCOFA	Programmazione modale del diametro specifica per asse: Off nei cicli		Programmazione del raggio, ultimo codice G attivo.	DIACYCOFA[Asse]	m	
DIAM90	Programmazione del diametro per G90, programmazione del raggio per G91				m	29
DIAM90A	Programmazione modale del diametro specifica per asse per G90 e AC, programmazione del raggio per G91 e IC				m	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
DIAMCHAN	Acquisizione di tutti gli assi dagli MD delle funzioni degli assi nello stato del canale della programmazione del diametro		Acquisizione della programmazione e del diametro dall'MD.	DIAMCHAN		
DIAMCHANA	Acquisizione dello stato del canale della programmazione del diametro		Stato del canale.	DIAMCHANA[asse]		
DIAMCYCOF	Programmazione del raggio per G90/G91: ON. Per la visualizzazione rimane attivo l'ultimo codice G attivo per questo gruppo		Programmazione e del raggio, ultimo codice G attivo.		m	29
DIAMOF ¹	Programmazione diametrale: OFF (Diametral programming OFF) Impostazione di base, vedere le indicazioni del costruttore della macchina		Programmazione e del raggio per G90/G91.		m	29
DIAMOFA	Programmazione modale del diametro specifica per asse: On Impostazione di base, vedere costruttore della macchina		Programmazione e del raggio per G90/G91 e AC, IC.	DIAMOFA[asse]	m	
DIAMON	Programmazione diametrale attivata: ON (Diametral programming ON)		Programmazione e del diametro per G90/G91.		m	29
DIAMONA	Programmazione modale del diametro specifica per asse: On Attivazione, vedere costruttore della macchina		Programmazione e del diametro per G90/G91 e AC, IC.	DIAMONA[asse]	m	
DIC	Programmazione del diametro relativa, specifica per asse e blocco-blocco		Programmazione e del diametro.	DIC(50)	s	
DILF	Lunghezza per svincolo rapido				m	
DISABLE	Interrupt OFF					
DISC	Sopraelevaz. cerchio di raccordo correzione raggio utensile	0, ..., 100			m	
DISPLOF	Soppressione della visualizzazione del blocco attuale (Display OFF)					
DISPR	Differenza del profilo di riposizionamento	Real, senza segno			s	
DISR	Distanza di riposizionamento	Real, senza segno			s	
DITE	Percorso di svincolo in filettatura	Real			m	
DITS	Percorso di accostamento in filettatura	Real			m	
DIV	Divisione Interio					
DL	Correzione utensile cumulativa	INT			m	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
DRFOF	Disattivazione delle traslazioni con volantino (DRF)				m	
DRIVE ^{7, 9}	Accelerazione vettoriale in funzione della velocità				m	21
DRIVEA	Attivare la caratteristica di accelerazione a due inclinazioni per gli assi programmati					
DYNFINISH	Dinamica per microfinitura	Tecnologia Gruppo G		DYNFINISH G1 X10 Y20 Z30 F1000	m	59
DYNNORM	Dinamica normale come in precedenza			DYNNORM G1 X10	m	59
DYNPOS	Dinamica per funzionamento di posizionamento, maschiatura			DYNPOS G1 X10 Y20 Z30 F...	m	59
DYNROUGH	Dinamica per sgrossatura			DYNROUGH G1 X10 Y20 Z30 F10000	m	59
DYNSEMIFIN	Dinamica per la finitura			DYNSEMIFIN G1 X10 Y20 Z30 F2000	m	59
EAUTO	Definizione dell'ultimo segmento Spline attraverso gli ultimi 3 punti (end not a knot)				m	20
EGDEF	Definizione di un cambio elettronico (electronic gear define)		Per 1 asse slave con max. 5 assi master.			
EGDEL	Cancellare definizione accoppiamento per asse slave (electronic gear delete)		Esegue blocco pre-elaborazione.			
EGOFC	Disattivazione continua cambio elettronico (electronic gear OFF continuous)					
EGOFS	Disattivazione selettiva cambio elettronico (electronic gear OFF selectiv)					
EGON	Attivazione cambio elettronico (electronic gear ON)		Senza sincronizzazione			
EGONSYN	Attivazione cambio elettronico (electronic gear ON synchronized)		Con sincronizzazione			
EGONSYNE	Attivazione cambio elettronico, con impostazione del modo di accostamento (electronic gear ON synchronized)		Con sincronizzazione			
ELSE	Diramazione del programma, se non è soddisfatta la condizione IF					
ENABLE	Interrupt ON					
ENAT ^{1, 7}	Raccordo della curva naturale con il blocco successivo (end natural)				m	20

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
ENDFOR	Riga di fine del loop di conteggio FOR					
ENDIF	Riga di fine della diramazione IF					
ENDLOOP	Riga di fine del loop di programma senza fine LOOP					
ENDPROC	Riga di fine di un programma con riga iniziale PROC					
ENDWHILE	Riga di fine del loop WHILE					
ETAN	Raccordo della curva tangenziale con il blocco successivo a inizio Spline (end tangential)				m	20
EVERY	Esecuzione dell'azione sincrona se la condizione passa da FALSE a TRUE					
EXECSTRING	Trasmissione di una variabile String con la riga di partprogram da eseguire		Riga di partprogram indiretta.	EXECSTRING(MFCT1 << M4711)		
EXECTAB	Elaborazione di un elemento di una tabella di movimento (Execute table)					
EXECUTE	Esecuzione programma ON		Passaggio al normale modo di lavorazione dalla preparazione del profilo o dopo la creazione di un settore di protezione.			
EXP	Funzione esponenziale e ^x	Real				
EXTCALL	Esecuzione di un sottoprogramma esterno		Caricamento di un programma da HMI nel modo "elaborazione dall'esterno".			
EXTERN	Comunicazione di un sottoprogramma con trasferimento dei parametri					
F	Valore avanzamento (in combinazione con G4, in F viene programmato anche il tempo di sosta)	0.001, ..., 99999.999	Velocità vettoriale utensile/pezzo; unità di misura in mm/min o mm/giro in base a G94 o G95.	F=100 G1 ...		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
FA	Avanzamento assiale (feed axial)	0.001, ..., 999999.999 mm/min, gradi/min; 0.001, ..., 39999.9999 pollici/min		FA[X]=100	m	
FAD	Avanzamento di incremento per accostamento e svincolo morbido (Feed approach / depart)	Real, senza segno				
FALSE	Costante logica: falso	BOOL	Sostituibile con costante Integer 0.			
FCTDEF	Definizione funzione polinomica		Viene valutato in SYNFACT o PUTFTOCF.			
FCUB ⁶	Avanzamento modificabile dopo Spline cubica (feed cubic)		Ha effetto sull'avanzamento o con G93 e G94.		m	37
FD	Avanzamento vettoriale per sovrapposizione del volante (feed DRF)	Real, senza segno			s	
FDA	Avanzamento assiale per sovrapposizione del volante (feed DRF axial)	Real, senza segno			s	
FENDNORM	Rallentamento d'angolo OFF				m	57
FFWOF ¹	Precomando OFF (feed forward OFF)				m	24
FFWON	Precomando ON (feed forward ON)				m	24
FGREF	Raggio di riferimento con assi rotanti o fattori di riferimento di traiettoria negli assi di orientamento (interpolazione vettoriale)		Grandezza di riferimento valore effettivo		m	
FGROUP	Definizione dell'asse (o degli assi) con avanzamento tangenziale		F vale per tutti gli assi indicati in FGROUP.	FGROUP (Asse1, [Asse2], ...)		
FIFOCTRL	Controllo del buffer di preelaborazione				m	4
FIFOLEN	Profondità di preelaborazione programmabile (preprocessing depth)					
FILEDATE	Fornisce la data dell'ultimo accesso in scrittura al file	STRING, lunghezza 8	Il formato è "dd.mm.yy".			
FILEINFO	Fornisce la somma di FILEDATE, FILESIZE, FILESTAT e FILETIME	STRING, lunghezza 32	Formato "rwxsd nnnnnnnn dd. hh:mm:ss".			

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
FILESIZE	Fornisce le dimensioni attuali del file	Tipo INT	In BYTE.			
FILESTAT	Fornisce lo stato del file per i diritti di lettura, scrittura, esecuzione, visualizzazione, cancellazione (rwxsd)	STRING, lunghezza 5	Il formato è "rwxsd".			
FILETIME	Fornisce l'ora dell'ultimo accesso in scrittura al file	STRING, lunghezza 8	Il formato è "dd:mm:yy".			
FINEA	Fine movimento al raggiungimento di "arresto di precisione fine"			FINEA=... opp. FINEA[n]=...	m	
FL	Velocità limite per assi sincroni (feed limit)	Real, senza segno	Vale per l'unità definita con G93, G94, G95 (max. rapido).	FL [Asse] =...	m	
FLIN ⁶	Avanzamento modificabile in modo lineare (feed linear)		Ha effetto sull'avanzamento con G93 e G94.		m	37
FMA	Avanzamento multiplo assiale (feed multiple axial)	Real, senza segno			m	
FNORM ^{1,6}	Avanzamento normale secondo DIN66025 (feed normal)				m	37
FOCOF	Disattivazione avanzamento con coppia/forza limitata				m	
FOCON	Attivazione avanzamento con coppia/forza limitata				m	
FOR	Loop di conteggio con numero fisso di ripetizioni					
FP	Punto fisso: Numero del punto fisso da raggiungere	Intero, senza segno		G75 FP=1	s	
FPO	Caratteristica dell'avanzamento programmata con un polinomio (feed polynomial)	Real	Coefficiente polinomiale quadratico, cubico.			
FPR	Identificazione asse rotante	0.001, ..., 999999.999		FPR (asse rotante)		
FPRAOF	Disattivazione avanzamento per giro					
FPRAON	Attivazione avanzamento per giro					

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
FRAME	Tipo di dati per la definizione del sistema di coordinate		Per ogni asse geometrico contiene: traslazione, rotazione, angolo di taglio, fattore di scala, specularità; Per ogni asse ausiliario: traslazione, fattore di scala, specularità			
FRC	Avanzamento per raggio e smusso				s	
FRCM	Avanzamento per raggio e smusso modale				m	
FTOC	Modificare correzione fine dell'utensile		In funzione di un polinomio di 3° grado definito con FCTDEF.			
FTOCOF ^{1,6}	Correzione utensile precisa attiva online OFF (fine tool offset OFF)				m	33
FTOCON ⁶	Correzione utensile precisa attiva online ON (fine tool offset ON)				m	33
FXS	Posizionamento su riscontro fisso ON (fixed stop)	Intero, senza segno	1=attivazione; 0=disattivazione		m	
FXST	Limite di coppia per posizionamento su riscontro fisso (fixed stop torque)	%	indicazione opzionale		m	
FXSW	Finestra di sorveglianza per posizionamento su riscontro fisso (fixed stop window)	mm, inch oppure gradi	indicazione opzionale			
G	Funzione G (funzione preparatoria) Le funzioni G sono suddivise in gruppi G. In un blocco può essere scritta solo una funzione G di un gruppo. Una funzione G può avere azione modale (finché non viene annullata da un'altra funzione dello stesso gruppo) oppure è attiva solo per il blocco in cui si trova (azione blocco-blocco).	Solo valori interi, preimpostati		G...		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
G0	Interpolazione lineare in rapido (movimento rapido)		Comandi di movimento	G0 X... Z...	m	1
G1 ¹	Interpolazione lineare con avanzamento (interpolazione lineare)			G1 X... Z... F...	m	1
G2	Interpolazione circolare in senso orario			G2 X... Z... I... K... F... ;centro e punto di arrivo G2 X... Z... CR=... F... ;raggio e punto di arrivo G2 AR=... I... K... F... ;angolo di apertura e ;centro G2 AR=... X... Z... F. ;angolo di apertura e ;punto di arrivo	m	1
G3	Interpolazione circolare in senso antiorario			G3...; altrimenti come con G2	m	1
G4	Tempo di sosta, determinato		Movimento speciale	G4 F... ; tempo di sosta in s oppure G4 S... ; tempo di sosta in ;giri mandrino. ; blocco a sé stante	s	2
G5	Rettifica con mola obliqua		Rettifica con mola obliqua		s	2
G7	Movimento di compensazione nella rettifica con mola obliqua		Posizione di partenza		s	2
G9	Arresto preciso - riduzione velocità				s	11
G17 ¹	Scelta del piano di lavoro X/Y		Direz. incremento Z		m	6
G18	Scelta del piano di lavoro Z/X		Direz. incremento Y		m	6
G19	Scelta del piano di lavoro Y/Z		Direz. incremento X		m	6
G25	Limitazione inferiore del campo di lavoro		Assegnazione del valore negli asse canale.	G25 X... Y... Z... ;blocco a sé stante	s	3
G26	Limitazione superiore del campo di lavoro			G26 X... Y... Z... ;blocco a sé stante	s	3
G33	Interpolazione della filettatura con passo costante	0.001, ..., 2000.00 mm/giro	Comando di movimento	G33 Z... K... SF=... ; filettatura cilindrica G33 X... I... SF=... ; filettatura longitudinale G33 Z... X... K... SF=... ; filettatura conica (percorso su asse Z maggiore che sull'asse X) G33 Z... X... I... SF=... ; filettatura conica (percorso su asse X maggiore che sull'asse Z)	m	1

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
G34	Variazione di velocità lineare progressiva [mm/giro ²]		Comando di movimento	G34 X... Y... Z... I... J... K... F...	m	1
G35	Variazione di velocità lineare decrescente [mm/giro ²]		Comando di movimento	G35 X... Y... Z... I... J... K... F...	m	1
G40 ¹	Correzione raggio utensile OFF				m	7
G41	Correzione raggio utensile a sinistra del profilo				m	7
G42	Correzione raggio utensile a destra del profilo				m	7
G53	Soppressione dello spostamento origine attuale (blocco per blocco)		Incluse le traslazioni programmate.		s	9
G54	1° spostamento origine impostabile				m	8
G55	2. spostamento origine impostabile				m	8
G56	3. spostamento origine impostabile				m	8
G57	4. spostamento origine impostabile				m	8
G58	Spostamento origine programmabile assiale assoluto				s	3
G59	Spostamento origine programmabile assiale addizionale				s	3
G60 ¹	Arresto preciso - riduzione velocità				m	10
G62	Rallentamento d'angolo sugli spigoli interni con correzione raggio utensile attiva (G41, G42)		Solo con funzionamento continuo.	G62 Z... G1	m	57
G63	Maschiatura con compensatore			G63 Z... G1	s	2
G64	Arresto preciso - funzionamento continuo				m	10
G70	Impostazione delle quote in pollici (lunghezze)				m	13
G71 ¹	Impostazione metrica delle quote (lunghezze)				m	13
G74	Ricerca punto di riferimento			G74 X... Z... ;blocco a sé stante	s	2
G75	Ricerca di un punto fisso		Assi macchina	G75 FP=.. X1=... Z1=... ;blocco a sé stante	s	2
G90 ¹	Quota assoluta			G90 X... Y... Z...() Y=AC() o X=AC Z=AC()	m s	14
G91	Impostazione quote incrementali			G91 X... Y... Z... oppure X=IC() Y=IC() Z=IC()	m s	14
G93	Avanzamento inversamente proporzionale al tempo 1/min		esecuzione di un blocco: tempo di esecuzione	G93 G01 X... F...	m	15

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
G94 ¹	Avanzamento lineare F in mm/min oppure pollici/min e al °/min				m	15
G95	Avanzamento al giro F al mm/giro oppure pollici/giro				m	15
G96	Velocità di taglio costante (come per G95) ON			G96 S... LIMS=... F...	m	15
G97	Velocità di taglio costante (come per G95) OFF				m	15
G110	Programmazione del polo relativo all'ultima posizione programmata			G110 X... Y... Z...	s	3
G111	Programmazione dei poli relativamente al punto zero del sistema di coordinate del pezzo attuale			G110 X... Y... Z...	s	3
G112	Programmazione del polo relativa all'ultimo polo valido			G110 X... Y... Z...	s	3
G140 ¹	Direzione accostamento WAB definita da G41/G42				m	43
G141	Direzione accostamento WAB a sinistra del profilo				m	43
G142	Direzione accostamento WAB a destra del profilo				m	43
G143	Direzione accostamento WAB dipendente dalla tangente				m	43
G147	Accostamento tangenziale con retta				s	2
G148	Distacco tangenziale con retta				s	2
G153	Soppressione di frame attuali inclusi frame di base		Incluso frame di sistema.		s	9
G247	Accostamento tangenziale con quarto di cerchio				s	2
G248	Distacco tangenziale con quarto di cerchio				s	2
G290	Commutazione sul modo SINUMERIK ON				m	47
G291	Commutazione sul modo ISO2/3 ON				m	47
G331	Maschiatura	±0.001,..., 2000.00 mm/giro	Comandi di movimento		m	1
G332	Svincolo (Maschiatura)				m	1
G340 ¹	Blocco di accostamento spaziale (in profondità e nel piano contemporaneamente (elicoide))		Efficace nell'accostamento/svincolo morbido.		m	44
G341	Prima incremento nell'asse verticale (z), quindi accostamento nel piano		Efficace nell'accostamento/svincolo morbido.		m	44
G347	Accostamento tangenziale con semicerchio				s	2
G348	Distacco tangenziale con semicerchio				s	2
G450 ¹	Cerchio di raccordo		Comportamento sugli spigoli con correzione raggio utensile.		m	18

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
G451	Punto di intersezione delle equidistanti				m	18
G460 ¹	Sorveglianza collisioni per blocco accostamento e svincolo attivata				m	48
G461	Prolungamento blocco marginale con arco di cerchio, se...		... Non esiste nessun punto d'intersezione nel blocco CRU		m	48
G462	Prolungamento blocco marginale con retta, se ...				m	48
G500 ¹	disattivazione di tutti i frame impostabili se in G500 non vi è alcun valore				m	8
G505 ... G599	5 ... 99. Spostamento origine impostabile				m	8
G601 ¹	Cambio blocco con arresto preciso fine		Attivo solo: - con G60 attivo oppure - con G9 con raccordo tra i blocchi programmabile		m	12
G602	Passaggio di blocco ad arresto preciso grossolano				m	12
G603	Passaggio di blocco a fine blocco IPO				m	12
G641	Arresto preciso - funzionamento continuo			G641 ADIS=...	m	10
G642	Raccordo con precisione assiale				m	10
G643	Arrotondamento interno al blocco				m	10
G644	Movimento raccordato con impostazione della dinamica dell'asse				m	10
G621	Rallentamento d'angolo su tutti gli spigoli		Solo con funzionamento continuo.	G621 ADIS=...	m	57
G700	Impostazione delle quote in pollici e pollici/min (lunghezze + velocità + variabili di sistema)				m	13
G710 ¹	Impostazione delle quote in mm e mm/min (lunghezze + velocità + variabili di sistema)				m	13
G810 ¹ , ..., G819	Gruppo G riservato per utente OEM					31
G820 ¹ , ..., G829	Gruppo G riservato per utente OEM					32
G931	Impostazione avanzamento tramite tempo di esecuzione		Tempo di esecuzione avanzamento		m	15
G942	Congelare l'avanzamento lineare e la velocità di taglio costante oppure dei giri del mandrino				m	15
G952	Congelare l'avanzamento al giro e la velocità di taglio costante oppure dei giri del mandrino				m	15
G961	Velocità di taglio costante e avanzamento lineare		Tipo di avanzamento come per G94.	G961 S... LIMS=... F...	m	15

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Grup po ²
G962	Avanzamento lineare oppure avanzamento al giro e velocità di taglio costante				m	15
G971	Congelare i giri del mandrino e avanzamento lineare		Tipo di avanzamento come per G94.		m	15
G972	Congelare l'avanzamento lineare oppure l'avanzamento al giro e i giri del mandrino costanti				m	15
G973	Avanzamento al giro senza limitazione della velocità del mandrino		G97 senza LIMS per modo ISO.		m	15
GEOAX	Assegnare nuovi assi canale agli assi geometrici 1 - 3		Senza parametri: è attiva la definizione dei dati macchina.			
GET	Occupare l'asse macchina		L'asse deve essere abilitato nell'altro canale con RELEASE.			
GETD	Occupare direttamente l'asse macchina		Vedere GET.			
GETACTT	Determinare utensile attivo da un gruppo di utensili con lo stesso nome					
GETSELT	Restituire il numero T selezionato					
GETT	Determinare il numero della T per un nome di utensile					
GOTO	Istruzione di salto prima in avanti e poi indietro (direzione prima verso la fine e poi verso l'inizio del programma)		Utilizzabile anche nel partprogram e nei cicli tecnologici.	GOTO (label, n. di blocco) Le label devono essere presenti nel sottoprogramma.		
GTOF	Istruzione di salto in avanti (direzione fine programma)			GTOF (label, n. di blocco)		
GTOB	Istruzione di salto indietro (direzione inizio programma)			GTOB (label, n. di blocco)		
GTOC	Soppressione dell'allarme 14080 "Destinazione di salto non trovata"		Vedere GOTO.			
GWPSOF	Disattivare la velocità periferica costante della mola (VPM)			GWPSOF(Nr.T)	s	
GWPSON	Attivare la velocità periferica costante della mola (VPM)			GWPSON(Nr.T)	s	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
H...	Emissione funzioni ausiliarie al PLC	Progr. Real/INT.: REAL: 0 ... +/- 3.4028 exp38 INT: -2147483646 ... +2147483647 Visualizzazione: ± 999 999 999,9999	Impostabile tramite dato macchina (costruttore macchina).	H100 opp. H2=100		
I ⁴	Parametri di interpolazione	Real			s	
I1	Coordinata punto intermedio	Real			s	
IC	Impostazione in quote incrementali	0, ..., ±99999.999°		X=IC(10)	s	
ICYCOF	Elaborazione di tutti i blocchi di un ciclo tecnologico dopo ICYCOF in un clock IPO		Solo all'interno del livello di programma.			
ICYCON	Elaborazione di ogni blocco di un ciclo tecnologico dopo ICYCON in un clock IPO separato		Solo all'interno del livello di programma.			
IDS	Identificazione di azioni sincrone statiche					
IF	Introduzione di un salto condizionato nel partprogram / ciclo tecnologico		Struttura: IF-ELSE-ENDIF	IF (condizione)		
INCCW	Avanzamento su un evolvente del cerchio in senso antiorario con interpolazione dell'evolvente con G17/G18/G19	Real	punto finale: Centro: Raggio con CR > 0: angolo di rotazione in gradi fra vettore iniziale e finale	INCW/INCCW X... Y... Z... INCW/INCCW I... J... K... INCW/INCCW CR=... AR... Programmazione diretta: INCW/INCCW I... J... K... CR=... AR=...	m	1
INCW	Avanzamento su un evolvente del cerchio in senso orario con interpolazione dell'evolvente con G17/G18/G19	Real			m	1
INDEX	Determinazione di indice di un carattere nella stringa di ingresso	0, ..., INT	Stringa: 1° parametro Carattere: 2° parametro			

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
INIT	Selezionare blocco per esecuzione in un canale		Numero canale 1-10 o \$MC_CHAN_NAME	INIT(1,1,2) o INIT(CH_X, CH_Y)		
INT	Tipo di dati: valore intero con segno	- (2 ³¹ -1), ..., 2 ³¹ -1				
INTERSEC	Calcolo del punto di intersezione di due elementi del profilo e indicazione dello stato del punto di intersezione TRUE in ISPOINT	VAR REAL [2]	Stato dell'errore ISPOINT: BOOL FALSE	ISPOINTS= INTERSEC (TABNAME1[n1], TABNAME2[n2], ISTCOORD, MODE)		
IP	Parametro di interpolazione variabile (interpolation parameter)	Real				
IPOBRKA	Criterio di spostamento sul punto di attivazione della rampa di frenatura		Rampa di frenatura da 100% a 0%.	IPOBRKA=.. oppure IPOBRKA(<Asse>[,REAL])	m	
IPOENDA	Fine dello spostamento al raggiungimento di "IPO-Stop"			IPOENDA=.. oppure IPOENDA[n]..	m	
IPTRLOCK	Congelare l'inizio della sezione di programma senza possibilità di ricerca, sul successivo blocco di funzioni macchina.		Congelamento del puntatore di interruzione.		m	
IPTRUNLOCK	Impostare sul blocco attuale come punto di interruzione la fine della sezione di programma senza possibilità di ricerca.		Impostazione del puntatore di interruzione.		m	
ISAXIS	Verificare se l'asse geometrico 1 indicato come parametro è disponibile	BOOL				
ISD	Profondità di penetrazione (insertion depth)	Real			m	
ISFILE	Verifica se un file è presente nella memoria utente dell'NCK.	BOOL	Restituisce un risultato di tipo BOOL.	RESULT=ISFILE("Testfile") IF (RESULT==FALSE)		
ISNUMBER	Verifica se la stringa in ingresso può essere convertita in un numero	BOOL	Converte la stringa di ingresso in numero.			
ISPOINTS	Calcolo, a partire da ISTAB, dei punti di intersezione possibili tra due profili nel piano attuale	INT	Tipo di lavorazione MODE (opzionale).	STATE=ISPOINTS (KTAB1[n1], KTAB2[n2], ISTAB, [MODE])		
ISVAR	Verifica se il parametro di trasferimento contiene una variabile nota all'NC	BOOL	Dati macchina, dati setting e variabile come GUD.			

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
J ⁴	Parametri di interpolazione	Real			s	
J1	Coordinata punto intermedio	Real			s	
JERKA	Attiva il comportamento all'accelerazione degli assi programmati impostato mediante i dati macchina					
JERKLIMA ⁵	Riduzione o sopraelevazione della scossa massima assiale (jerk axial)	1, ..., 200	Campo di validità 1 ... 200%	JERKLIMA[X]= ...[%]	m	
K ⁴	Parametri di interpolazione	Real			s	
K1	Coordinata punto intermedio	Real			s	
KONT	Aggiramento del profilo in caso di correzione utensile				m	17
KONTC	Accostamento/svincolo con polinomio a curvatura continua				m	17
KONTT	Accostamento/svincolo con polinomio a tangente continua				m	17
L	Numero di sottoprogramma	Intero, fino a 7 cifre		L10	s	
LEAD ⁵	Angolo di anticipo	Real			m	
LEADOF	Accoppiamento al valore master OFF (lead off)					
LEADON	Accoppiamento al valore master ON (lead on)					
LFOF ¹	Interruzione di filettatura OFF				m	41
LFON	Interruzione di filettatura ON				m	41
LFPOS	Svincolo assiale su una posizione				m	46
LFTXT ¹	Direzione utensile nella rimozione tangenziale				m	46
LFWP	Direzione utensile nella rimozione non tangenziale				m	46
LIFTFAST	Distacco rapido prima del richiamo della routine di interrupt					
LIMS	Limitazione numero di giri con G96/G961 e G97 (limit spindle speed)	0.001, ..., 99 999. 999			m	
LN	logaritmo naturale	Real				
LOCK	Bloccare azione sincrona con ID (arresto ciclo tecnologico)					
LOG	Logaritmo decimale	Real				
LOOP	Introduzione di un loop senza fine		Struttura: LOOP- ENDLOOP			

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
M...	Funzioni di commutazione	INT Visualizzazione: 0, ..., 999 999 999 Programma: 0, ..., 2147483647	Max. 5 funzioni M libere da definire da parte del costruttore della macchina.			
M0 ⁹	Arresto programmato					
M1 ⁹	arresto opzionale					
M2 ⁹	fine programma principale con reset all'inizio del programma					
M3	rotazione destrorsa per il mandrino master					
M4	rotazione sinistrorsa per il mandrino master					
M5	Arresto mandrino per mandrino master					
M6	Cambio dell'utensile					
M17 ⁹	Fine sottoprogramma					
M19	Programmazioni mandrino raccolte con ricerca blocco					
M30 ⁹	fine programma, come M2					
M40	cambio gamma automatico					
M41... M45	Gamma 1, ..., 5					
M70	Commutazione al funzionamento assi					
MASLDEF	Definisce una configurazione master/slave					
MASLDEL	Interrompe un collegamento master/slave e cancella la definizione di collegamento					
MASLOF	Disinserzione di un accoppiamento temporaneo					
MASLOFS	Disabilitazione di un accoppiamento temporaneo con arresto automatico dell'asse slave					
MASLON	Inserzione di un accoppiamento temporaneo					
MAXVAL	Valore superiore di due variabili (funzione aritmetica)	Real	L'uguaglianza fornisce un valore identico.	ValMax = MAXVAL(Var1, Var2)		
MCALL	Richiamo sottoprogramma modale		Senza nome di sottoprogramma : disattivazione			
MEAC	Misura continua senza cancellazione del percorso residuo	Intero, senza segno			s	
MEAFRAME	Calcolo del frame da punti di misura	FRAME				
MEAS	Misure con tastatore in commutazione (measure)	Intero, senza segno			s	
MEASA	Misura con cancellazione del percorso residuo				s	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
MEAW	Misure con tastatore commutante senza cancellazione del percorso residuo (measure without deleting distance to go)	Intero, senza segno			s	
MEAWA	Misura senza cancellazione del percorso residuo				s	
MI	Accesso ai dati del Frame: specularità (mirror)				MI	
MINDEX	Determinazione di indice di un carattere nella stringa di ingresso	0, ..., INT	Stringa: 1° parametro Carattere: 2° parametro			
MINVAL	Valore inferiore di due variabili (funzione aritmetica)	Real	L'uguaglianza fornisce un valore identico.	ValMin = MINVAL(Var1, Var2)		
MIRROR	Specularità programmabile			MIRROR X0 Y0 Z0; ;blocco a sé stante	s	3
MMC	Dal partprogram richiamare in modo interattivo la finestra di dialogo sull'HMI	STRING				
MOD	Divisione modulo					
MOV	Avviare l'asse di posizionamento (start moving positioning axis)	Real				
MSG	Messaggi programmabili			MSG("messaggio")	m	
N	Numero di blocco - blocco secondario	0, ..., 9999 9999 solo numeri interi, senza segno	può essere usato per identificare blocchi mediante un numero; va scritto all'inizio del blocco	ad es. N20		
NCK	Specifica del campo di validità dei dati		Presente una volta per NCK.			
NEWCONF	Trasferimento dati macchina modificati. Corrisponde all'attivazione di un dato macchina.		Possibile anche tramite softkey dell'HMI.			
NEWT	Creazione di un nuovo utensile		Il numero del Duplo può essere omesso.			
NORM ¹	Impostazione normale nel punto di partenza, punto di arrivo nella correzione utensile				m	17

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Grup po ²
NOT	NOT logico (negazione)					
NPROT	Settore di protezione specifico per la macchina ON/OFF					
NPROTDEF	Definizione di un settore di protezione specifico per la macchina (NCK specific protection area definition)					
NUMBER	Convertire la stringa di ingresso in numero	Real				
OEMIPO1 ^{6,8}	Interpolazione OEM 1				m	1
OEMIPO2 ^{6,8}	Interpolazione OEM 2				m	1
OF	Parola chiave nella diramazione CASE					
OFFN	Sovrametallo rispetto al profilo programmato			OFFN=5		
OMA1 ⁶	Indirizzo OEM 1	Real			m	
OMA2 ⁶	Indirizzo OEM 2	Real			m	
OMA3 ⁶	Indirizzo OEM 3	Real			m	
OMA4 ⁶	Indirizzo OEM 4	Real			m	
OMA5 ⁶	Indirizzo OEM 5	Real			m	
OFFN	Correzione offset - normale	Real			m	
OR	OR logico					
ORIC ^{1,6}	Le variazioni di orientamento sugli spigoli esterni vengono sovrapposte al blocco di cerchio da inserire (orientation change continuously)				m	27
ORID ⁶	Le variazioni di orientamento vengono eseguite prima del blocco di cerchio (orientation change discontinuously)				m	27
ORIXPOS	Angolo di orientamento tramite assi di orientamento virtuali con posizioni dell'asse rotante				m	50
ORIEULER	Angolo di orientamento tramite angolo di Eulero				m	50
ORIXES	Interpolazione lineare degli assi macchina o degli assi di orientamento		Orientamento finale: indicazione vettore A3, B3, C3 oppure angolo di Eulero RPY A2, B2, C2 Dati supplementari: Vettori angolari A6, B6, C6 Angolo di apertura del cono in gradi: 0 < NUT<180	impostazione dei parametri come segue: vettori direzionali normalizzati A6=0 B6=0 C6=1 Angolo di apertura eseguito come angolo di movimento con NUT=... NUT=+... a ≤180 gradi NUT= -... a ≥ 180 gradi Orientamento intermedio normalizzato A7=0 B7=0 C7=1	m	51
ORICONCW	Interpolazione su una superficie circolare in senso orario				m	51
ORICONCCW	Interpolazione su una superficie circolare in senso antiorario				m	51
ORICONIO	Interpolazione su una superficie circolare con indicazione di un orientamento intermedio				m	51
ORICONTO	Interpolazione su una superficie esterna circolare con raccordo tangenziale (indicazione dell'orientamento finale)				m	51
ORICURVE	Interpolazione dell'orientamento con preimpostazione del movimento di due punti di contatto dell'utensile				m	51

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
ORIPLANE	Interpolazione in un piano (corrisponde a ORIVECT) interpolazione cerchio massimo		Vettori intermedi: A7, B7, C7 Punto di contatto dell'utensile: XH, YH, ZH		m	51
ORIPATH	Orientamento utensile riferito alla traiettoria		Pacchetto di trasformazione Handling (vedere /FB3/ TE4).		m	51
ORIPATHS	Orientamento dell'utensile riferito al percorso, viene raddrizzata una piega durante l'orientamento		Riferito relativamente all'intero percorso.		m	51
ORIROTA	Angolo di rotazione in un senso di rotazione preimpostato in modo assoluto				m	54
ORIROTC	Vettore di rotazione tangenziale rispetto alla tangente vettoriale		Riferito alla tangente vettoriale.		m	54
ORIROTR	Angolo di rotazione relativo al piano fra orientamento iniziale e finale				m	54
ORIROTT	Angolo di rotazione riferito alla variazione del vettore di orientamento				m	54
ORIRPY	Angolo di orientamento mediante angolo RPY (XYZ)		Successione di rotazione XYZ		m	50
ORIRPY2	Angolo di orientamento mediante angolo RPY (ZYX)		Successione di rotazione ZYX		m	50
ORIS ⁵	Variazione di orientamento (orientation smoothing factor)	Real	Riferito al percorso.		m	
ORIVECT	Interpolazione del cerchio più grande (identico a ORIPLANE)				m	51
ORIVIRT1	Angolo di orientamento tramite assi di orientamento virtuali (definizione 1)				m	50
ORIVIRT2	Angolo di orientamento tramite assi di orientamento virtuali (definizione 1)				m	50
ORIMKS ⁶	Orientamento utensile nel sistema di coordinate di macchina (tool orientation in machine coordinate system)				m	25
ORIRESET	Impostazione di base dell'orientamento utensile con max. 3 assi di orientamento		Parametro opzionale (REAL)	ORIRESET(A,B,C)		
ORIWKS ^{1,6}	Orientamento utensile nel sistema coordinate pezzo (tool orientation in workpiece coordinate system)				m	25
OS	Pendolamento on/off	Intero, senza segno				

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
OSB	Pendolamento: Punto di partenza				m	
OSC ⁶	Livellamento costante dell'orientamento utensile				m	34
OSCILL	Abbinamento asse per pendolamento - attivazione pendolamento		Axis: 1 - 3 di incremento		m	
OSCTRL	Opzione pendolamento	Intero, senza segno			m	
OSD ⁶	Raccordare il movimento dell'orientamento utensile impostando la lunghezza di raccordo con SD		Interno al blocco		m	34
OSE	Pendolamento: Punto finale				m	
OSNSC	Pendolamento: Nr.passate spegnifiamma (oscillating: number spark out cycles)				m	
OSOF ^{1,6}	Livellamento orientamento utensile OFF				m	34
OSP1	Pendolamento: punto di inversione sinistro (oscillating: position 1)	Real			m	
OSP2	Pendolamento: punto di inversione destro (oscillating: position 2)	Real			m	
OSS ⁶	Livellamento dell'orientamento utensile a fine blocco				m	34
OSSE ⁶	Livellamento dell'orientamento utensile a inizio e fine blocco				m	34
OST ⁶	Raccordo dell'orientamento utensile mediante impostazione della tolleranza angolare in gradi con DS (scostamento massimo dal valore programmato del percorso di orientamento)		Interno al blocco		m	34
OST1	Pendolamento: Punto di arresto nel punto di inversione sinistro	Real			m	
OST2	Pendolamento: Punto di arresto nel punto di inversione destro	Real			m	
OVR	Correzione giri (override)	1, ..., 200%			m	
OVRA	Correzione giri assiale (override)	1, ..., 200%			m	
P	Numero di ripetizioni di un sottoprogramma	1, ..., 9999 Intero senza segno		ad es. L781 P... ;blocco a sé stante		
PCALL	Richiamo di sottoprogrammi con indicazione assoluta del percorso e trasferimento dei parametri		Nessun percorso assoluto. Comportamento come CALL.			
PAROT	Allestimento del sistema di coordinate pezzo sul pezzo				m	52

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
PAROTOF	Disattivazione della rotazione frame riferita al pezzo				m	52
PDELAYOF ⁶	Ritardo nella punzonatura OFF (punch with delay OFF)				m	36
PDELAYON ^{1,6}	Ritardo nella punzonatura ON (punch with delay ON)				m	36
PL	Lunghezza intervalli parametri	Real, senza segno			s	
PM	al minuto		Avanzamento al minuto.			
PO	Polinomio	Real, senza segno			s	
POLF	Posizione LIFTFAST	Real, senza segno	Asse geometrico in SCP, altrimenti SCM.	POLF[Y]=10 posizione di destinazione dell'asse di svincolo	m	
POLFA	Avvio della posizione di svincolo per assi singoli con \$AA_ESR_TRIGGER		Per assi singoli.	POLFA(AX1, 1, 20.0)	m	
POLFMASK	Abilitazione degli assi per lo svincolo senza relazione tra gli assi stessi		Assi selezionati	POLFMASK(AX1, AX2, ...)	m	
POLFMLIN	Abilitazione degli assi per lo svincolo con relazione lineare tra gli assi stessi		Assi selezionati	POLFMLIN(AX1, AX2, ...)	m	
POLY ⁵	interpolazione polinomica				m	1
POLYPATH ⁵	Interpolazione polinomiale selezionabile per i gruppi di assi AXIS o VECT			POLYPATH ("AXES") POLYPATH ("VECT")	m	1
PON ⁶	Punzonatura ON (punch ON)				m	35
PONS ⁶	Punzonatura ON nel clock IPO (punch ON slow)				m	35
POS	Posizionamento asse			POS[X]=20		
POSA	Posizionamento asse con condizionamento di fine blocco			POSA[Y]=20		
POSP	Posizionamento in parti (pendolamento) (position axis in parts)	Real: End-position, lunghezza parte; Integer: Opzione				
POT	Quadrato (funzione aritmetica)	Real				
PR	Per giro (per revolution)			Avanzamento al giro		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
PRESETON	Impostazione valore reale per assi programmati		Vengono programmati rispettivamente un indicatore dell'asse e, nel successivo parametro, il relativo valore. Sono possibili max. 8 assi.	PRESETON(X,10,Y,4.5)		
PRI0	Parola chiave per impostare la priorità nel trattamento degli interrupt					
PROC	Prima istruzione di un programma			Numero blocco - PROC - identificatore		
PTP	Movimento da punto a punto (point to point)		Asse sincrono		m	49
PTPG0	Movimento da punto a punto solo con G0, altrimenti CP		Asse sincrono		m	49
PUTFTOC	Correzione utensile precisa per diamantatura parallela (continuous dressing) (Put Fine Tool Correction)		Numero canale 1-10 o \$MC _CHAN_NAME	PUTFTOC(1,1,2) o PUTFTOC(CH_name)		
PUTFTOCF	Correzione utensile precisa secondo una funzione definita con FCtDEF per diamantatura parallela (continuous dressing) (put fine tool correction function dependant)		Numero canale 1-10 o \$MC _CHAN_NAME	PUTFTOCF(1,1,2) o PUTFTOCF(CH_name)		
PW	Punto - peso (point weight)	Real, senza segno			s	
QECLRN0F	Apprendimento compensazione errore quadrante OFF (quadrant error compensation learning OFF)					
QECLRN0N	Apprendimento compensazione errore quadrante ON (quadrant error compensation learning ON)					
QU	Emissione funzione (ausiliaria) supplementare veloce					
R...	Parametri di calcolo anche sotto forma di indicatore dell'asse impostabile e con estensione numerica	± 0.0000001, ..., 9999 9999	Il numero di parametri R è impostabile mediante i dati macchina.	R10=3 ;assegnazione parametri R X=R10 ;valore asseR[R10]=6 ;programmazione indiretta		
RAC	Programmazione del raggio assoluta, specifica per asse e blocco-blocco		Programmazione e del raggio	RAC(50)	s	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ₃	Gruppo ²
RDISABLE	Blocco lettura (read in disable)					
READ	Legge una o più righe dal file indicato e memorizza le informazioni nel campo		L'informazione ha la forma di STRING.			
READAL	Lettura allarme (read alarm)		Gli allarmi vengono esaminati per ordine crescente di numero.			
REAL	Tipo di dati: variabile a virgola mobile con segno (numeri reali)	Corrisponde al formato floating-point a 64 bit del processore				
REDEF	Impostazione per dati macchina, elementi del linguaggio NC e variabili di sistema, con indicazione dei gruppi di utenti in cui vengono visualizzati					
RELEASE	Abilitazione assi macchina		Possono essere programmati più assi.			
REP	Parola chiave per l'inizializzazione di tutti gli elementi di un campo con lo stesso valore			REP(valore) o DO FELD[n, m]=REP()		
REPEAT	Ripetizione di un loop di programma		Finché (UNTIL) non è soddisfatta una condizione.			
REPEATB	Ripetizione di una riga di programma		nnn volte			
REPOSA	Riaccostamento al profilo lineare con tutti gli assi (repositioning linear all axes)				s	2
REPOSH	Riaccostamento al profilo con un semicerchio (repositioning semi circle)				s	2
REPOSHA	Riaccostamento al profilo con tutti gli assi; assi geometrici in semicerchio (repositioning semi circle all axes)				s	2
REPOSL	Riaccostamento al profilo lineare (repositioning linear)				s	2
REPOSQ	Riaccostamento al profilo con un quarto di cerchio (repositioning quarter circle)				s	2

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
REPOSQA	Riaccostamento al profilo con tutti gli assi; assi geometrici in un quarto di cerchio (repositioning quarter circle all axes)				s	2
RESET	Resettare ciclo tecnologico		Possono essere programmati uno o più ID.			
RET	Fine sottoprogramma		Utilizzo al posto di M17 – senza emissione di funzioni al PLC.	RET		
RIC	Programmazione del raggio relativa, specifica per asse e blocco-blocco		Programmazione e del raggio	RIC(50)	s	
RINDEX	Determinazione di indice di un carattere nella stringa di ingresso	0, ..., INT	Stringa: 1° parametro Carattere: 2° parametro			
RMB	Riaccostamento al punto di inizio del blocco (repos mode begin of block)				m	26
RME	Riaccostamento al punto di fine blocco (repos mode end of block)				m	26
RMI ¹	Riaccostamento al punto di interruzione (repos mode interrupt)				m	26
RMN	Riaccostamento al punto immediatamente successivo del percorso (repos mode end of nearest orbital block)				m	26
RND	Raccordo spigoli	Real, senza segno		RND=...	s	
RNDM	Raccordo modale	Real, senza segno		RNDM=... RNDM=0: disattivare racc. mod.	m	
ROT	Rotazione programmabile (rotation)	Rotazione intorno al 1° asse geometrico: -180°... +180° 2° asse geometrico: -90° ... +90° 3° asse geometrico: -180°... +180°		ROT X... Y... Z... ROT RPL= ;blocco a sé stante	s	3
ROTS	Rotazioni programmabili dei frame con angoli nello spazio (rotation)			ROTS X... Y... ROTS Z... X... ROTS Y... Z... ROTS RPL= ;blocco a sé stante	s	3

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
ROUND	Arrotondare le cifre decimali	Real				
RP	Raggio polare (radius polar)	Real			m/s	
RPL	Rotazione nel piano (rotation plane)	Real, senza segno			s	
RT	Parametro per l'accesso ai dati del Frame: rotazione (rotation)					
RTLION	G0 con interpolazione lineare				m	55
RTLIOF	G0 senza interpolazione lineare (interpolazione singolo asse)				m	55
S	Giri del mandrino oppure (per G4, G96/G961) altro significato	REAL Visualizzazione: ±999 999 999.9999 Programma: ±3,4028 ex38	Giri del mandrino in giri/min G4: Tempo di sosta in giri del mandrino G96/G961: Velocità di taglio in m/min	S...: Giri per mandrino master S1...: numero di giri per il mandrino1	m/s	
SAVE	Attributo per salvare informazioni in richiami di sottoprogrammi		Vengono salvate tutte le funzioni G modali e il frame attuale.			
SBLOF	Soppressione blocco singolo (single block OFF)		I blocchi seguenti vengono elaborati nel blocco singolo come un blocco.			
SBLON	Disattivazione soppressione del blocco singolo (single block ON)					
SC	Parametro per l'accesso ai dati del Frame: fattore di scala (scale)					
SCALE	Fattore di scala programmabile (scale)			SCALE X... Y... Z... ;blocco a sé stante	s	3
SCC	Assegnazione selettiva di un asse radiale a G96/G961/G962. Possono essere identificatori dell'asse gli assi geometrici, di canale o di macchina.		Anche con velocità di taglio costante attiva	SCC[Asse]		
SD	Gradi Spline (spline degree)	Intero, senza segno			s	
SEFORM	Istruzione di strutturazione nell'editor Step per generare la visualizzazione dei passi per HMI Advanced		Viene valutato nell'editor Step.	SEFORM (<nome sezione>, <livello>, <icon>)		
SET	Parola chiave per l'inizializzazione di tutti gli elementi di un campo con i valori elencati			SET(valore, valore, ...) o DO FELD[n,m]=SET()		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
SETAL	Settare allarme (set alarm)					
SETDNO	Impostare su "nuovo" il numero D dell'utensile (T) e del relativo tagliente					
SETINT	Determinazione della routine di interrupt che deve essere attivata se un ingresso NCK è in attesa		Viene valutato il fronte di salita 0 → 1.			
SETMS	ritorno al mandrino master definito nel dato macchina					
SETMS (n)	Mandrino n deve fungere da mandrino master					
SETPIECE	Verificare il numero di pezzi per tutti gli utensili abbinati al mandrino		Senza numero mandrino: vale solo per il mandrino master.			
SF	Traslazione del punto di partenza per filettatura (spline offset)	0.0000,..., 359.999°			m	
SIN	Seno (funzione trigonometrica)	Real				
SOFT	Accelerazione vettoriale con antistress				m	21
SOFTA	inserire l'accelerazione con funzione antistress per gli assi programmati					
SON ⁶	Roditura ON (stroke ON)				m	35
SONS ⁶	Roditura ON nel clock IPO (stroke ON slow)				m	35
SPATH ¹	Il riferimento di percorso per gli assi FGROUP è la lunghezza dell'arco				m	45
SPCOF	Commutazione mandrino master opp. mandrino/i da regolazione in posizione a regolazione in velocità			SPCOF SPCOF(n)	m	
SPCON	Commutazione mandrino master opp. mandrino/i da regolazione in velocità a regolazione in posizione			SPCON SPCON (n)	m	
SPIF1 ^{1,6}	Ingressi/uscite NCK veloci per punzonatura/roditura byte 1 (stroke/punch interface 1)				m	38
SPIF2 ⁶	Ingressi/uscite NCK veloci per punzonatura/roditura byte 2 (stroke/punch interface 2)				m	38
SPLINE-PATH	Definizione del raggruppamento Spline		Max. 8 assi.			
SPOF ^{1,6}	Tranciatura OFF, punzonatura, roditura OFF (stroke/punch OFF)				m	35
SPN ⁶	Numero di tratti di percorso per blocco (stroke/punch number)	Integer			s	
SPP ⁶	Lunghezza di un tratto di percorso (stroke/punch path)	Integer			m	
SPOS	Posizione del mandrino			SPOS=10 opp. SPOS[n]=10	m	

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
SPOSA	Posizione del mandrino oltre limiti di blocco			SPOSA=5 opp. SPOSA[n]=5	m	
SQRT	Radice quadrata (funzione aritmetica) (square root)	Real				
SR	Percorso di svincolo con pendolamento per azione sincrona (sparking out retract path)	Real, senza segno			s	
SRA	Percorso di svincolo per pendolamento con ingresso assiale esterno per azione sincrona (sparking out retract path axial)			SRA[Y]=0.2	m	
ST	Tempo di spegnifiamma per pendolamento assiale per azione sincrona (sparking out time)	Real, senza segno			s	
STA	Tempo di spegnifiamma con pendolamento assiale per azione sincrona (sparking out time axial)				m	
START	Avvio dei programmi selezionati in più canali contemporaneamente, dal programma in corso		Non attivo per il proprio canale.	START(1,1,2) o START(CH_X, CH_Y) \$MC_CHAN_NAME		
STARTFIFO ¹	Elaborazione; parallelamente riempimento della memoria di preelaborazione				m	4
STAT	Posizione dei giunti	Integer			s	
STOPFIFO	Arresto dell'elaborazione; riempimento della memoria di preelaborazione fino al riconoscimento di STARTFIFO, memoria di preelaborazione piena oppure fine programma				m	4
STOPRE	Stop preelaborazione, fino ad esecuzione avvenuta nell'esecuzione principale di tutti i blocchi preelaborati (stop preprocessing)					
STOPREOF	Disattivazione stop preelaborazione (stop preprocessing OFF)					
STRING	Tipo di dati: Stringa di caratteri	Max. 200 caratteri				
STRINGIS	Verifica il set di istruzioni del linguaggio NC e in particolare per questo comando i nomi dei cicli NC, le variabili utente, le macro e i nomi di label, per accertarsi che questi esistano, siano validi, definiti o attivi.	INT	I risultati dei valori di ritorno sono 000 non noti 100 programmabili 2XX riconosciuti come esistenti	STRINGIS (STRING,name)= valore di ritorno indicizzato		

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
STRLEN	Determinazione lunghezza di una stringa	INT				
SUBSTR	Determinazione di indice di un carattere nella stringa di ingresso	Real	String: 1° parametro, carattere: 2° parametro			
SUPA	Soppressione dello spostamento origine attuale, incluse le traslazioni programmate, i frame di sistema, le traslazioni DRF, lo spostamento origine esterno e movimento sovrapposto				s	9
SYNFCT	Valutazione di un polinomio in funzione di una condizione nell'azione sincrona al movimento	VAR REAL				
SYNR	La lettura della variabile avviene in modo sincrono, ovvero al momento dell'elaborazione (synchronous read)					
SYNRW	La lettura e la scrittura della variabile avvengono in modo sincrono, ovvero al momento dell'elaborazione (synchronous read-write)					
SYNW	La scrittura della variabile avviene in modo sincrono, ovvero al momento dell'elaborazione (synchronous write)					
T	Richiamo utensile (cambio solo se definito nel dato macchina, altrimenti è necessario il comando M6)	1, ..., 32 000	Richiamo tramite n. T oppure tramite l'identificatore dell'utensile.	ad es. T3 o T=3 ad es. T="PUNTA"		
TAN	Tangente (funzione trigonometrica)	Real				
TANG	Determinazione della tangente per l'inseguimento dai due assi master indicati					
TANGOF	Inseguimento tangenziale OFF (tangential follow up mode OFF)					
TANGON	Inseguimento tangenziale ON (tangential follow up mode ON)					
TCARR	Richiedere portautensile (numero "m")	Integer	m=0: disattiva il supporto utensile attivo	TCARR=1		
TCOABS ¹	Determinazione componenti della lunghezza utensile dall'orientamento utensile attuale		Indispensabile dopo il riattrezzaggio, ad es. dopo impostazioni manuali.		m	42
TCOFR	Determinazione componenti lunghezza utensile da orientamento del frame attuale				m	42

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
TCOFRX	Nella scelta di un utensile definizione dell'orientamento utensile di un frame attivo, utensile nella direzione di X		Utensile perpendicolare alla superficie inclinata.		m	42
TCOFRY	Nella scelta di un utensile definizione dell'orientamento utensile di un frame attivo, utensile nella direzione di Y		Utensile perpendicolare alla superficie inclinata.		m	42
TCOFRZ	Nella scelta di un utensile definizione dell'orientamento utensile di un frame attivo, utensile nella direzione di Z		Utensile perpendicolare alla superficie inclinata.		m	42
THETA	Angolo di rotazione		THETA è sempre in posizione ortogonale rispetto all'orientamento attuale dell'utensile.	THETA=valore THETA=AC THETA=IC polinomio per THETA PO[THET]=(...)	s	
TILT ⁵	Angolo laterale	Real		TILT=valore	m	
TMOF	Disattivazione della sorveglianza utensile		Il numero T è necessario solo se l'utensile con questo numero non è attivo.	TMOF (n. T)		
TMON	Selezione sorveglianza utensili		N. T. = 0: disattivare la sorveglianza per tutti gli utensili	TMON (n. T)		
TO	Definisce il valore finale in un loop di conteggio FOR					
TOFFOF	Resettare la correzione della lunghezza utensile on line					
TOFFON	Attivazione della correzione lunghezza utensile online (tool offset ON)		Impostazione di una direzione di correzione tridimensionale.	TOFFON (Z, 25) con direzione di correzione Z valore Offset 25		
TOFRAME	Settare il frame programmabile attuale sul sistema di coordinate dell'utensile		Rotazione frame nella direzione dell'utensile.		m	53
TOFRAMEX	Asse X parallelo alla direzione dell'utensile, asse secondario Y, Z				m	53
TOFRAMEY	Asse Y parallelo alla direzione dell'utensile, asse secondario Z, X				m	53
TOFRAMEZ	Asse Z parallelo alla direzione dell'utensile, asse secondario X, Y				m	53
TOLOWER	Convertire tutte le lettere della stringa di ingresso in lettere minuscole					
TOROTOF	Rotazioni frame in direzione dell'utensile OFF				m	53

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
TOROT	Asse Z parallelo all'orientamento utensile		Rotazioni frame ON Componente di rotazione del frame programmabile		m	53
TOROTX	Asse X parallelo all'orientamento utensile				m	53
TOROTY	Asse Y parallelo all'orientamento utensile				m	53
TOROTZ	Asse Z parallelo all'orientamento utensile				m	53
TOUPPER	Convertire tutte le lettere della stringa di ingresso in lettere maiuscole					
TOWSTD	Valore di posizione base per le correzioni della lunghezza utensile		Considerazione nel calcolo dell'usura utensile		m	56
TOWBCS	Valori di usura nel sistema di coordinate base (SCB)				m	56
TOWKCS	Valori di usura nel sistema di coordinate della testa dell'utensile con trasformazione cinetica (differisce dal SCM per la rotazione dell'utensile)				m	56
TOWMCS	Valori di usura nel sistema di coordinate macchina (SCM)				m	56
TOWTCS	Valori di usura nel sistema di coordinate utensile (punto di riferimento del portautensili T nel suo punto di innesto)				m	56
TOWWCS	Valori di usura nel sistema di coordinate pezzo (SCP)				m	56
TRAANG	Trasformazione per asse inclinato		Per ogni canale possono essere attivate più trasformazioni.			
TRACEOF	Test della forma del cerchio: trasferimento dei valori OFF					
TRACEON	Test della forma del cerchio: trasferimento dei valori ON					
TRACON	Trasformazione concatenata (transformation concatenated)					
TRACYL	Cilindro: trasformazione superficie del cilindro		Vedere TRAANG.			
TRAFOOF	Disattivazione della trasformazione			TRAFOOF()		
TRAILOF	Trascinamento asincrono OFF (trailing OFF)					
TRAILON	Trascinamento asincrono ON (trailing ON)					
TRANS	Spostamento programmabile (translation)			TRANS X... Y... Z... ;blocco a sé stante	s	3
TRANSMIT	Trasformazione polare		Vedere TRAANG.			

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
TRAORI	Trasformazione a 4, 5 assi, trasformazione generica (transformation oriented)		Attiva la trasformazione dell'orientamento o dichiarata.	Trasformazione generica TRAORI(1,X,Y,Z)		
TRUE	Costante logica: vero	BOOL	Sostituibile con costante Integer 1.			
TRUNC	Esclusione delle cifre decimali	Real				
TU	Angolo asse	Integer		TU=2	s	
TURN	Numero di spire per l'interpolazione elicoidale	0, ..., 999			s	
UNLOCK	Abilitare azione sincrona con ID (continuare ciclo tecnologico)					
UNTIL	Condizione per la conclusione di un loop REPEAT					
UPATH	Il riferimento di percorso per gli assi FGROU è il parametro della curva				m	45
VAR	Parola chiave: tipo di trasferimento dei parametri		Con VAR: call by reference			
VELOLIMA ⁵	Riduzione o incremento eccessivo dell'accelerazione massima assiale (velocity axial)	1, ..., 200	Campo di validità 1 ... 200%	VELOLIMA[X]= ...[%]	m	
WAITC	Attesa finché non è soddisfatto il criterio di cambio blocco di accoppiamento per gli assi / mandrini (wait for couple condition)		È possibile programmare max. 2 assi / mandrini.	WAITC(1,1,2)		
WAITE	Attendere la fine del programma in un altro canale.		Numero canale 1-10 o \$MC_CHAN_NAME	WAITE(1,1,2) o WAITE(CH_X, CH_Y)		
WAITM	Attesa del marker nel canale specificato; terminare il blocco precedente con arresto preciso.		Numero canale 1-10 o \$MC_CHAN_NAME	WAITM(1,1,2) o WAITM(CH_X, CH_Y)		
WAITMC	Attesa del marker nel canale indicato; arresto preciso solo se gli altri canali non hanno ancora raggiunto il marker.		Numero canale 1-10 o \$MC_CHAN_NAME	WAITMC(1,1,2) o WAITMC(CH_X, CH_Y)		
WAITP	Attesa di fine posizionamento			WAITP(X) ; blocco a sé stante		
WAITS	Attesa raggiungimento posizione mandrino			WAITS (mandrino master) WAITS (n,n,n)		
WALCS0	Limitazione del campo di lavoro WKS selezionata				m	60
WALCS1	Gruppo di limitazione 1 del campo di lavoro WKS attivo				m	60
WALCS2	Gruppo di limitazione 2 del campo di lavoro WKS attivo				m	60

Nome	Significato	Valore	Descrizione, commento	Sintassi	m/s ³	Gruppo ²
WALCS3	Gruppo di limitazione 3 del campo di lavoro WKS attivo				m	60
WALCS4	Gruppo di limitazione 4 del campo di lavoro WKS attivo				m	60
WALCS5	Gruppo di limitazione 5 del campo di lavoro WKS attivo				m	60
WALCS6	Gruppo di limitazione 6 del campo di lavoro WKS attivo				m	60
WALCS7	Gruppo di limitazione 7 del campo di lavoro WKS attivo				m	60
WALCS8	Gruppo di limitazione 8 del campo di lavoro WKS attivo				m	60
WALCS9	Gruppo di limitazione 9 del campo di lavoro WKS attivo				m	60
WALCS10	Gruppo di limitazione 10 del campo di lavoro WKS attivo				m	60
WALIMOF	Limitazione campo di lavoro SCB OFF (working area limitation OFF)			; blocco a sé stante	m	28
WALIMON ¹	Limitazione campo di lavoro SCB ON (working area limitation ON)			; blocco a sé stante	m	28
WHILE	Inizio del loop di programma WHILE		Fine: ENDWHILE			
WRITE	Scrivere blocco in file system. Aggiunge un blocco alla fine del file indicato.		I blocchi vengono inseriti dopo M30.			
X	Asse	Real			m/s	
XOR	OR logico esclusivo					
Y	Asse	Real			m/s	
Z	Asse	Real			m/s	

Legenda:

- ¹ Impostazione standard a inizio programma (nello stato di fornitura del controllo numerico, salvo diversa programmazione).
- ² La numerazione dei gruppi corrisponde alla tabella "Lista delle funzioni G/condizioni di percorso".
- ³ Punti finali assoluti: modale (m)
punti finali incrementali: blocco per blocco (s)
Altrimenti: m/s in funzione di quanto definito dalla sintassi della funzione G
- ⁴ Come centri del cerchio valgono i parametri di interpolazione (IPO) incrementali. Con AC essi possono essere programmati in assoluto. Per altri significati (ad es. passo della filettatura) la modifica dell'indirizzo viene ignorata.
- ⁵ La parola chiave non vale per il SINUMERIK 810D.
- ⁵ La parola chiave non vale per il SINUMERIK 810D/NCU571.
- ⁷ La parola chiave vale solo per il SINUMERIK FM-NC.
- ⁸ L'utente OEM può inserire due ulteriori tipi di interpolazione. I nomi possono essere modificati dall'utente OEM.
- ⁹ Per queste funzioni non è consentito utilizzare la forma di scrittura estesa per l'indirizzo.

Appendice

A.1 Elenco delle abbreviazioni

A	Uscita
AS	Sistema d'automazione
ASCII	American Standard Code for Information Interchange: Codice standard americano per lo scambio di informazioni
ASIC	Application Specific Integrated Circuit: circuito integrato per applicazione specifica
ASUP	Sottoprogramma asincrono
AV	Preparazione del lavoro
AWL	Lista istruzioni
BA	Modo operativo
BAG	Gruppi di modi operativi
BB	Pronto al funzionamento
BuB, B&B	Servizio e supervisione
BCD	Binary Coded Decimals: numeri decimali codificati in codice binario
BHG	Tastiera operativa manuale
BIN	File binari (B inary F iles)
BIOS	Basic Input Output System
BKS	Sistema di coordinate base
BOF	Superficie operativa
BOT	Boot Files: File di avvio per SIMODRIVE 611 digitale
BT	Pannello operatore
BTSS	Interfaccia pannello operatore
CAD	Computer-Aided Design: progettazione supportata da computer
CAM	Computer-Aided Manufacturing: produzione supportata da computer
CNC	Computerized Numerical Control: Controllo numerico computerizzato
COM	Communication
CP	Processore di comunicazione
CPU	Central Processing Unit: unità di elaborazione centrale
CR	Carriage Return
CRT	Cathode Ray Tube: tubo catodico
CSB	Central Service Board: scheda PLC
CTS	Clear To Send: messaggio di pronto per invio con interfacce dati seriali
CUTOM	Cutter radius compensation: Correzione raggio utensile
DAU	DAC (convertitore analogico-digitale)
DB	Blocco dati nel PLC
DBB	Byte nel blocco dati nel PLC
DBW	Parola nel blocco dati nel PLC
DBX	Bit nel blocco dati nel PLC
DC	Direct Control: movimento dell'asse rotante verso la posizione assoluta nell'ambito di un giro attraverso il percorso più breve
DCD	Carrier Detect
DDE	Dynamic Data Exchange: scambio dati automatico

DEE	Terminale per dati
DIN	Norme industriali tedesche
DIO	Data Input/Output: visualizzazione trasmissione dati
DIR	Directory: Directory
DLL	Dynamic Link Library
DOE	Apparecchio per la trasmissione di dati
DOS	Disk Operating System
DPM	Dual Port Memory: memoria a doppio accesso
DPR	Dual-Port-RAM: memoria di scrittura/lettura a doppio accesso
DRAM	Dynamic Random Access Memory: memoria di scrittura/lettura dinamica
DRF	Differential Resolver Function: funzione resolver differenziale (volantino)
DRY	Dry Run: Avanzamento per ciclo di prova
DSB	Decoding Single Block: decodifica blocco singolo
DW	Parola dati
E	Ingresso
E/A	Immissione/emissione
E/R	Unità di alimentazione/recupero (alimentatore) del SIMODRIVE 611 digitale
Codice EIA	Codice speciale per nastro perforato, il numero di buchi per carattere è sempre dispari
ENC	Encoder: trasduttore di posizione
EPROM	Erasable Programmable Read Only Memory (memoria di lettura cancellabile e programmabile elettricamente)
ERROR	Error from printer
FB	Blocco funzionale
FBS	Schermo piatto
FC	Function Call: blocco funzionale (nel PLC)
FDB	Banca dati dei prodotti
FDD	Floppy Disk Drive: unità a dischetti
FEPROM	Flash-EPROM: memoria di lettura e scrittura
FIFO	First in First Out: memoria che lavora senza indicazione dell'indirizzo e nella quale i dati vengono letti nella stessa sequenza in cui sono stati salvati.
FIPO	Interpolatore fine
FM	Modulo funzionale
FM-NC	Modulo funzione - controllo numerico
FPU	Floating Point Unit: unità a virgola mobile
FRA	Blocco frame
FRAME	Gruppo di dati (cumulativi)
FRK	Correzione raggio (CRF)
FST	Feed Stop: stop avanzamento
FUP	Schema funzionale (metodo di programmazione per il PLC)
GP	Programma base
GUD	Global User Data: Dati utente globali
HD	Hard Disk: Disco rigido
HEX	Acronimo per valore esadecimale

HiFu	Funzione ausiliaria
HMI	Human Machine Interface: funzionalità del SINUMERIK per operatività, programmazione e simulazione.
HMS	Sistema di misura ad alta risoluzione
HSA	Azionamento mandrino principale
HW	Hardware
IBN	messa serv.
IF	Abilitazione impulsi per il modulo di azionamento
IK (GD)	Comunicazione implicita (dati globali)
IKA	Interpolative Compensation: Compensazione interpolatoria
IM	Interface-Modul: scheda d'interfaccia
IMR	Interface-Modul Receive: scheda d'interfaccia per la ricezione
IMS	Interface-Modul Send: scheda d'interfaccia per la trasmissione
INC	Increment: Quota incrementale
INI	Initializing Data: Dati di inizializzazione
IPO	Interpolatore
ISA	International Standard Architecture
ISO	International Standard Organization
Codice ISO	Codice speciale per nastro perforato, il numero di fori per carattere è sempre pari
JOG	Jogging: messa a punto
K1 .. K4	Canale da 1 a 4
K-Bus	Bus di comunicazione
KD	Rotazione delle coordinate
KOP	Schema a contatti (metodo di programmazione per il PLC)
K_v	Fattore di guadagno dell'anello
K_U	Rapporto di trasmissione
LCD	Liquid Crystal Display: display a cristalli liquidi
LED	Light-Emitting Diode: indicatore a diodi luminosi
LF	Line Feed
LMS	Sistema di misura della posizione
LR	Regolatore di posizione
LUD	Local User Data
MB	Megabyte
MD	Dati macchina
MDA	Manual Data Automatic: immissione manuale
MK	Circuito di misura
SCM	Sistema di coordinate macchina
MLFB	Numero di ordinazione del materiale
MPF	Main Program File: partprogram NC (programma principale)
MPI	Multi Point Interface: interfaccia multipoint
MS-	Microsoft (produttore del software)
MSTT	Pulsantiera di macchina
NC	Numerical Control: controllo numerico

NCK	Numerical Control Kernel: nucleo numerico con preparazione blocco, campo di posizionamento, ecc.
NCU	Numerical Control Unit: unità hardware dell'NCK
NRK	Denominazione del sistema operativo dell'NCK
NST	Segnale di interfaccia
NURBS	Non Uniform Rational B-Spline
NV	spostamento origine (SO)
OB	Blocco organizzativo (nel PLC)
OEM	Original Equipment Manufacturer: costruttore i cui prodotti vengono venduti con il nome di altre società
OP	Operation Panel: pannello operativo
OPI	Operation Panel Interface: interfaccia pannello operativo
OPT	Options: opzioni
OSI	Open Systems Interconnection: Standard per la comunicazione fra computer
P-Bus	Bus di periferia
PC	Personal Computer
PCIN	Nome del SW per lo scambio dati con il controllo numerico
PCMCIA	Personal Computer Memory Card International Association: standard per le schede di memoria ad innesto
PCU	PC Unit: PC-Box (unità di calcolo)
PG	Console de programmation
PLC	Programmable Logic Control: controllore programmabile
POS	...di posizionamento
RAM	Random Access Memory: memoria programmabile per lettura e scrittura
REF	Funzione di ricerca del punto di riferimento
REPOS	Funzione di riposizionamento
RISC	Reduced Instruction Set Computer: tipo di processore con set di istruzioni ridotto e esecuzione rapida del comando
ROV	Rapid Override: override del rapido
RPA	R-Parameter Active: area di memoria nell'NCK per R NCK per numeri di parametri R
RPY	Roll Pitch Yaw: rotazione di un sistema di coordinate
RTS	Request To Send: attivazione del dispositivo di trasmissione, segnale di comando dalle interfacce dati seriali
SBL	Single Block: Blocco singolo
SD	Dati di setting
SDB	Blocco dati di sistema
SEA	Setting Data Active: identificatore (tipo di file) per dati di setting
SFB	Blocco funzionale di sistema
SFC	System Function Call: richiamo di funzione per sistema
SK	Softkey
SKP	Skip: esclusione blocco
SM	Motore passo-passo
SPF	Sub Program File: Sottoprogramma
SPS	Controllore programmabile (PLC)

SRAM	Memoria statica (tamponata)
SRK	Correzione raggio tagliente (CRT)
SSFK	Compensazione errore passo vite (CEPV)
SSI	Serial Synchron Interface: interfaccia seriale sincrona
SW	Software
SYF	System Files: file di sistema
TEA	Testing Data Active: codice per i dati macchina
TO	Tool Offset: Correzione utensile
TOA	Tool Offset Active: identificatore (tipo di file) per correzioni utensile
TRANSMIT	Transform Milling into Turning: commutazione di coordinate nei torni per lavorazione di fresatura
UFR	User Frame: spostamento origine (SO)
UP	Sottoprogramma
VSA	Azionamento assi
V.24	Interfaccia seriale (definizione dei collegamenti di scambio tra DEE e DÜE)
SCP	Sistema di coordinate pezzo
WKZ	Utensile (UT)
WLK	Correzione lunghezza utensile (CLU)
WOP	Programmazione di officina
WPD	Work Piece Directory: Directory dei pezzi
WRK	Correzione del raggio utensile (CRU)
WZK	Correzione utensile
WZW	cambio dell'utensile
ZOA	Zero Offset Active: identificatore (tipo di file) per i dati di spostamento del punto zero
µC	Microcontrollore

A.2 Informazioni specifiche per la pubblicazione

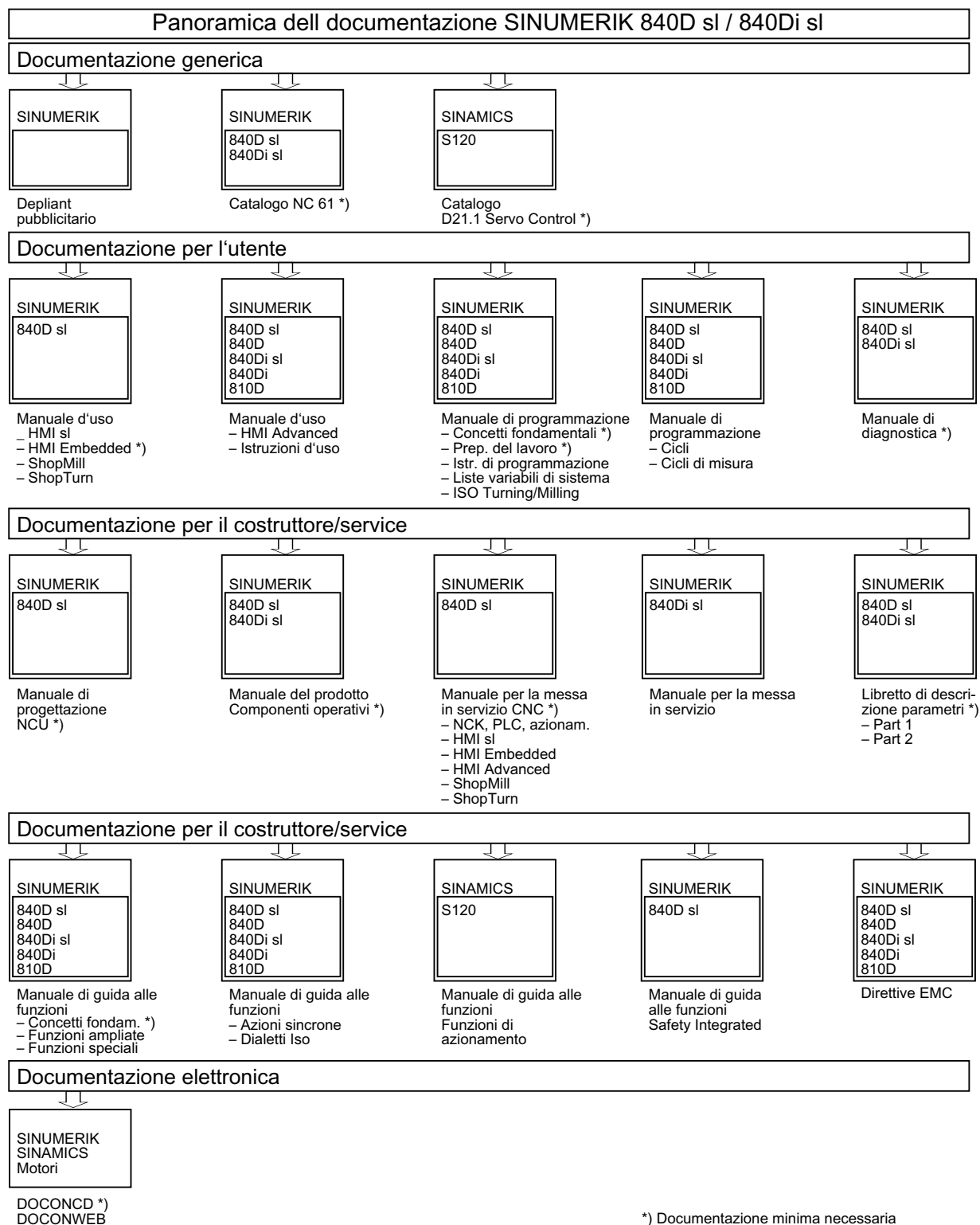
A.2.1 Modulo per le correzioni - Modulo fax

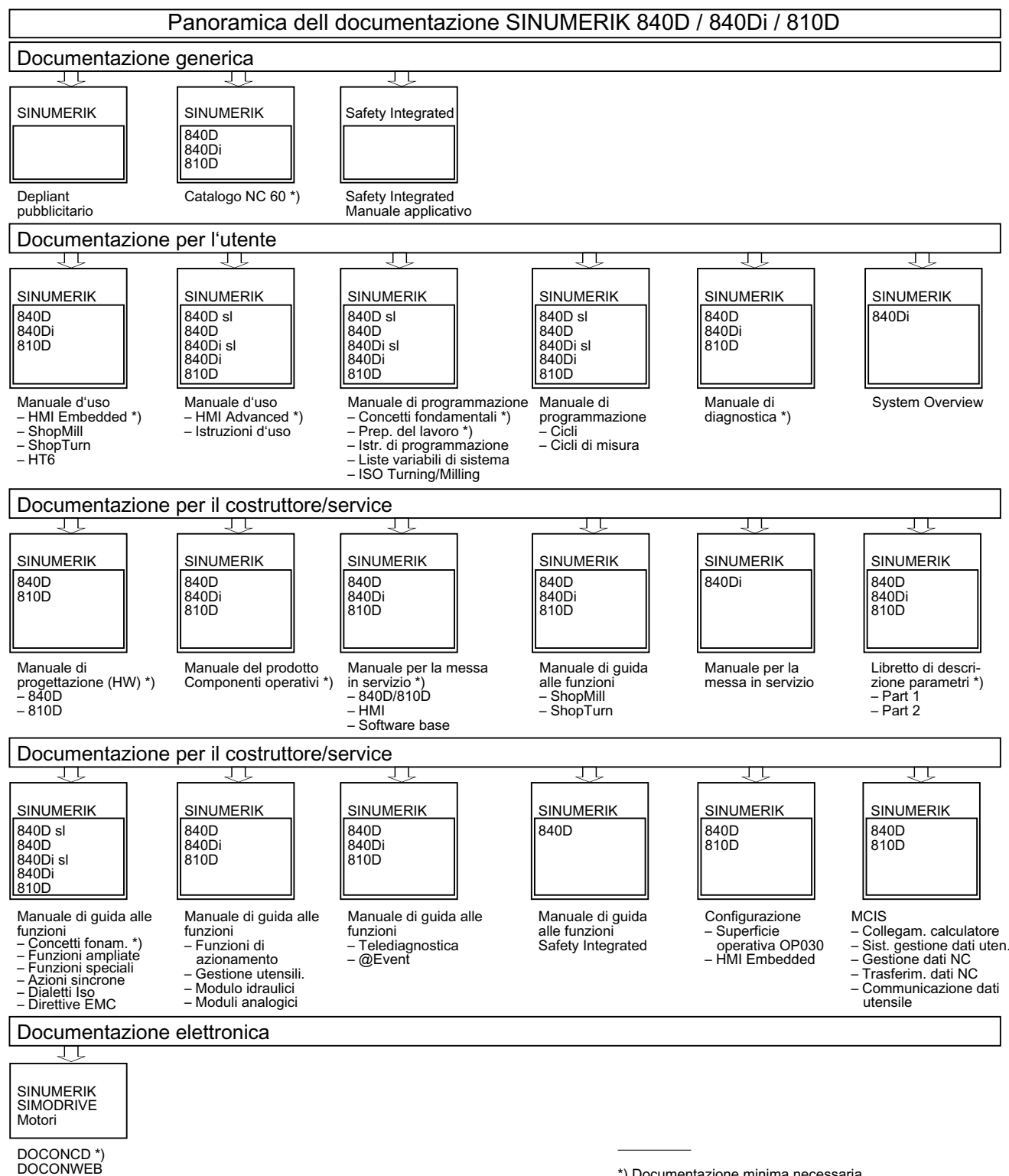
Se durante la lettura di questo manuale doveste riscontrare degli errori di stampa, Vi preghiamo di comunicarceli utilizzando il presente modulo. Vi siamo altresì grati per eventuali suggerimenti e proposte di miglioramento.

A SIEMENS AG A&D MC MS1 Postfach 3180 D-91050 Erlangen Fax.: +49 (0) 9131 / 98 - 63315 (documentazione) mailto:docu.motioncontrol@siemens.com http://www.siemens.com/automation/service&support	Mittente	
	Nome:	
	Indirizzo della Ditta/dell'ufficio	
	Via:	
	Cap	Località:
	Telefono:	/
	Telefax:	/

Proposte e/o correzioni

A.2.2 Panoramica della documentazione





*) Documentazione minima necessaria

Glossaire

Accelerazione con antistress meccanico

Per realizzare un comportamento di accelerazione ottimale sulla macchina rendendo più docili i movimenti, è possibile differenziare nel programma di lavoro tra accelerazione a gradino e accelerazione con variazione continua (senza stress).

Accoppiatore di bus

Un accoppiatore di bus è un accessorio S7-300 che viene fornito unitamente alle ->unità di periferia. L'accoppiatore di bus collega il -> bus S7-300 della ->CPU opp. un'unità di periferia all'unità di periferia contigua.

Allarmi

Tutti gli -> allarmi e messaggi vengono visualizzati in chiaro sul pannello operatore con data e ora e con il corrispondente simbolo che rappresenta il criterio di tacitazione. La visualizzazione di allarmi e messaggi avviene separatamente.

1. Allarmi e messaggi nel partprogram

Allarmi e messaggi della macchina possono essere visualizzati con testo in chiaro direttamente dal partprogram.

2. Allarmi e messaggi da PLC

Allarmi e messaggi della macchina possono essere visualizzati con testo in chiaro attraverso il programma PLC. Per fare questo non sono necessari ulteriori pacchetti di blocchi funzionali.

Archiviazione

Trasferire i file e/o le directory su un dispositivo di backup **esterno**.

Arresto orientato del mandrino

Arresto del mandrino portapezzo in una posizione angolare preimpostata, ad. es. per poter eseguire una lavorazione supplementare in una determinata posizione.

Arresto preciso

Se si programma l'istruzione di arresto preciso, la posizione indicata nel blocco viene raggiunta con precisione ed eventualmente molto lentamente. Per ridurre i tempi di accostamento, vengono definite le -> soglie di arresto preciso per rapido e avanzamento.

A-Spline

Lo Spline Akima passa in forma tangenziale sui punti di appoggio programmati (polinomio di terzo grado).

Asse base

Asse il cui valore programmato o valore reale viene preso in considerazione per il calcolo di un valore di compensazione.

Asse C

Asse che gestisce il movimento rotativo e il posizionamento utilizzando il mandrino portapezzo.

Asse di compensazione

Asse il cui valore richiesto o reale viene modificato dal valore di compensazione.

Asse di interpolazione

Gli assi di interpolazione sono tutti quegli assi di lavoro del -> canale gestiti dall' -> interpolatore in modo tale che partano, accelerino, si arrestino e raggiungano il punto di arrivo contemporaneamente.

Asse di posizionamento

Asse che esegue un movimento ausiliario della macchina utensile (ad es. magazzino utensili, trasporto pallet). Gli assi di posizionamento sono assi che non interpolano con gli -> assi interpolanti.

Asse geometrico

Gli assi geometrici servono a descrivere un settore bi- o tridimensionale nel sistema di coordinate del pezzo.

Asse lineare

L'asse lineare è un asse che, a differenza di un asse rotante, descrive una retta.

Asse rotante

Gli assi rotanti provocano una rotazione del pezzo o dell'utensile in una posizione angolare predefinita.

Gli assi rotanti eseguono una rotazione del pezzo o dell'utensile in una posizione angolare corrispondente al reticolo di divisione. Una volta raggiunta una determinata divisione, l'asse rotante è "in posizione".

Asse rotante

Gli assi rotanti provocano una rotazione del pezzo o dell'utensile in una posizione angolare predefinita.

Gli assi rotanti eseguono una rotazione del pezzo o dell'utensile in una posizione angolare corrispondente al reticolo di divisione. Una volta raggiunta una determinata divisione, l'asse rotante è "in posizione".

Assi

In base alla loro funzione, gli assi CNC vengono suddivisi in:

- Assi: assi di interpolazione lineari
- assi ausiliari: assi di posizionamento e assi ausiliari non interpolanti con avanzamenti specifici. Gli assi ausiliari non partecipano alla lavorazione vera e propria, per es. navetta utensili, magazzino utensili.

Assi della macchina

Assi fisicamente esistenti sulla macchina utensile.

Assi sincroni

Per compiere il loro percorso gli assi sincroni necessitano dello stesso tempo degli assi geometrici per il percorso di contornitura.

Automatico

Modo operativo del controllo numerico (funzionamento continuo secondo DIN): modo operativo per i sistemi NC nel quale un -> partprogram viene selezionato ed elaborato in modo continuo.

Avanzamento reciproco nel tempo

In SINUMERIK 840D è possibile programmare, invece della velocità di avanzamento per il movimento degli assi, il tempo necessario per il percorso di contornitura di un blocco (G93).

Avanzamento vettoriale

L'avanzamento vettoriale agisce sugli -> assi di interpolazione. Esso rappresenta la somma geometrica degli avanzamenti dei vari -> assi geometrici interessati.

Azionamento

Il sistema di controllo numerico SINUMERIK 840D è collegato in modo digitale con il sistema di convertitori SIMODRIVE 611 tramite un bus parallelo digitale ad alta velocità.

Azioni sincrone

1. Emissione di funzioni ausiliarie

Durante la lavorazione del pezzo, dal programma CNC è possibile emettere al PLC funzioni tecnologiche (→ funzioni ausiliarie). Con queste funzioni ausiliarie vengono comandati, ad esempio, dispositivi supplementari della macchina utensile, quali la contropunta, le pinze, l'autocentrante, etc.

2. Emissione veloce di funzioni ausiliarie

Per funzioni di comando a reazione rapida è possibile minimizzare i tempi di conferma delle → funzioni ausiliarie evitando così arresti indesiderati del processo di lavorazione.

Back up

Creazione di una copia del contenuto della memoria su un supporto dati esterno.

Batteria tampone

La batteria tampone assicura che il → programma utente memorizzato nella → CPU sia salvato anche in caso di interruzione di corrente e che i settori dati definiti, marker, temporizzatori ed i contatori vengano mantenuti in modo retentivo.

Baudrate

Velocità nella trasmissione dati (bit/s).

Blocchi intermedi

I movimenti con → correzione utensile selezionata (G41/G42) possono essere interrotti da un numero limitato di blocchi intermedi (blocchi senza movimenti degli assi nel piano di correzione), il che non impedisce che la correzione utensile possa ancora essere calcolata correttamente. Il numero ammesso di blocchi intermedi che vengono letti anticipatamente dal controllo numerico può essere impostato mediante parametri di sistema.

Blocco

Per blocchi si intendono tutti i file necessari per la stesura e l'elaborazione del programma.

Blocco dati

1. Unità di dati del → PLC, alla quale possono accedere i programmi → HIGHSTEP.
2. Unità di dati del → NC: i blocchi dati contengono definizioni per dati utente globali. I dati possono essere inizializzati direttamente durante la definizione.

Blocco di inizializzazione

I blocchi di inizializzazione sono → blocchi di programma speciali. Contengono assegnazioni di variabili che possono essere eseguite prima dell'elaborazione di un programma. I blocchi di inizializzazione servono soprattutto per l'inizializzazione di dati predefiniti o dati utente globali.

Blocco di programma

I blocchi di programma contengono i programmi principali e i sottoprogrammi dei
→ programmi pezzo.

Blocco di programma pezzo

Sezione di → programma pezzo delimitato da Line Feed. Si distinguono → blocchi principali e
→ blocchi secondari.

Blocco principale

Blocco contrassegnato con ":" che contiene tutte le informazioni necessarie per poter avviare
il ciclo di lavorazione in un → partprogram.

Blocco secondario

Blocco che inizia con "N" contenente le informazioni necessarie per un passo di lavoro, ad
es. un'indicazione di posizione.

Boot

Caricamento del sistema operativo dopo Power On.

B-Spline

Nella B-Spline le posizioni programmate non sono punti di appoggio, ma esclusivamente
"punti di controllo". La curva generata non passa sui punti di controllo, ma nelle loro
vicinanze (a scelta polinomio di primo, secondo e terzo grado).

Campo di lavoro

Spazio tridimensionale nel quale la punta dell'utensile può spostarsi in relazione alla struttura
della macchina utensile. Vedere → Zona di protezione.

Campo di posizionamento

Il massimo campo di posizionamento degli assi lineari è ± 9 decadi. Il valore assoluto
dipende dalla risoluzione di impostazione e di regolazione della posizione e dal sistema di
impostazione (in pollici o metrico) prescelti.

Campo di protezione

Spazio tridimensionale all'interno del → campo di lavoro, nel quale non deve entrare la punta
dell'utensile.

Canale

Un canale si contraddistingue per il fatto che consente di elaborare un -> partprogram indipendentemente da altri canali. Un canale controlla esclusivamente gli assi e i mandrini ad esso assegnati. I processi nei partprogram di vari canali possono essere coordinati tramite -> sincronizzazione.

Canale di lavorazione

Con una struttura a canali, grazie all'esecuzione di movimenti paralleli, ad es. movimento di un portale di carico contemporaneamente alla lavorazione, è possibile ridurre i tempi morti. Un canale va considerato come un CNC indipendente fornito di decodifica, preparazione del blocco e interpolazione autonome.

Cancellazione totale

Nella cancellazione totale vengono cancellate le seguenti memorie della → CPU:

- → Memoria di lavoro
- l'area di scrittura/lettura della → memoria di caricamento
- → Memoria di sistema
- → Memoria di backup

Cavi di collegamento

I cavi di collegamento sono cavi a 2 conduttori, preconfezionati o realizzati dall'utente, dotati di 2 connettori. Questi cavi di collegamento collegano la → CPU mediante → l'interfaccia multipoint (MPI) con un → PG o con altre CPU.

Chiave di programmazione

Caratteri e stringhe di caratteri che hanno un significato definito nel linguaggio di programmazione del → programma pezzo.

Cicli standard

Per compiti di lavorazione ripetitivi sono disponibili dei cicli standard:

- per la tecnologia di foratura/fresatura
- per la tecnologia tornitura

Nel settore operativo "Programma" sotto il menu "Supporto per cicli" vengono elencati i cicli disponibili. Dopo la selezione del ciclo di lavoro desiderato vengono visualizzati in chiaro i parametri necessari per la definizione dei valori.

Ciclo

Sottoprogramma protetto per l'esecuzione di un processo di lavorazione ripetitivo sul → pezzo.

CNC

Vedere -> NC

COM

Componente del controllo numerico preposto all'esecuzione e al coordinamento della comunicazione.

Compensazione del gioco

Compensazione del gioco meccanico della macchina, per es. gioco di inversione nelle viti a ricircolo di sfere. Per ogni asse la compensazione del gioco può essere immessa separatamente.

Compensazione dell'errore sul quadrante

Eventuali errori di profilo sui cambi del quadrante derivanti dalla variabilità dei rapporti di attrito delle guide che possono essere eliminati con la compensazione dell'errore sul quadrante. La parametrizzazione della compensazione dell'errore sul quadrante avviene con un test di prova della circolarità.

Compensazione errore passo vite

Compensazione di inesattezze meccaniche di una vite a ricircolo di sfere utilizzata come asse attraverso il controllo numerico in base a valori di correzione precedentemente definiti.

Compensazione interpolatoria

Con la compensazione interpolatoria è possibile compensare gli errori di produzione
Compensazione e errore e passo vite e compensazione errore sistema di misura (CEPV, CESM).

Controllore programmabile (PLC)

I controllori programmabili (PLC) sono apparecchiatura in grado di elaborare un programma applicativo memorizzato nel loro interno. La forma costruttiva e il cablaggio dell'apparecchiatura non dipendono quindi dalla sua funzione. Il controllore programmabile ha la stessa struttura di un computer; è costituito da CPU (unità centrale) con memoria, unità di ingresso e di uscita e sistema di bus interno. La periferia e il linguaggio di programmazione sono orientati alle esigenze della tecnica di controllo.

coordinate polari

Sistema di coordinate che definisce la posizione di un punto in un piano tramite la distanza dal punto zero e l'angolo compreso tra il vettore raggio e uno degli assi definiti.

Correzione raggio tagliente (CRT)

Nella programmazione di un profilo viene considerato un utensile a punta. Siccome questo nella pratica non è realizzabile, nel controllo numerico viene impostato il raggio di curvatura dell'utensile che viene considerato durante la lavorazione. Il profilo che viene creato è equidistante a quello programmato in funzione del raggio utensile.

Correzione raggio utensile

Per poter programmare direttamente il → profilo del pezzo desiderato, il controllo numerico, in considerazione del raggio dell'utensile utilizzato, deve calcolare ed eseguire un profilo equidistante da quello programmato (G41/G42).

Correzione utensile

Nel calcolo del percorso vengono considerate le misure dell'utensile.

CPU

Central Processor Unit, -> vedere -> controllore programmabile

C-Spline

Il C-Spline è lo spline più noto e maggiormente utilizzato. I passaggi ai punti di appoggio sono costanti dal punto di vista della tangente e della curvatura. Vengono utilizzati polinomi di terzo grado.

Dati setting

Dati che trasmettono al controllo numerico NC le caratteristiche della macchina utensile nel modo definito dal software di sistema.

Definizione delle variabili

La definizione di una variabile implica la definizione di un tipo di dati e del nome della variabile stessa. Quest'ultimo consente di interrogare il valore della variabile.

Diagnosi

1. Settore operativo del controllo numerico
2. Il controllo numerico possiede sia un programma di autodiagnostica sia test utili per il service: visualizzazioni di stato, allarme e service.

DRF

Differential Resolver Function: Funzione NC che, in concomitanza con un volantino elettronico, genera uno spostamento del punto zero incrementale in funzionamento automatico.

Editor

L'editor consente la stesura, la modifica, l'ampliamento, la composizione e l'inserimento di programmi/testi/blocchi di programma.

Editor di testi

Vedere → Editor

Fattore di scala

Componente di un → frame che determina asse per asse delle variazioni di scala.

File di inizializzazione

Ad ogni → pezzo è possibile abbinare un file di inizializzazione. In esso si possono inserire diverse istruzioni per valori di variabili, la cui validità è legata strettamente al pezzo.

Finecorsa software

I finecorsa software limitano il campo di posizionamento di un asse ed evitano che le slitte raggiungano i finecorsa hardware. Per ogni asse sono definibili 2 coppie di valori attivabili separatamente da → PLC.

Frame

Un frame rappresenta una prescrizione di calcolo che trasforma un sistema di coordinate cartesiano in un altro sistema cartesiano. Un frame contiene i componenti → spostamento origine, → rotazione, → fattore di scala, → specularità.

Frame programmabili

Con i → frame programmabili è possibile definire in modo dinamico, in base all'elaborazione del programma pezzo, nuovi punti di partenza del sistema di coordinate. Si fa distinzione tra definizione assoluta di un nuovo frame e definizione additiva rispetto ad un determinato punto di partenza.

Funzionamento continuo

Lo scopo del funzionamento continuo è quello di evitare un rallentamento eccessivo del controllo, della macchina e di altri beni aziendali e dell'utente → assi di interpolazione in corrispondenza dei cambi di blocco del programma pezzo al fine di passare al blocco successivo possibilmente con la stessa velocità vettoriale.

Funzioni ausiliarie

Le funzioni ausiliarie consentono di trasmettere al → PLC dei → parametri del → partprogram che provocano reazioni definite dal costruttore della macchina.

Funzioni di sicurezza

Il controllo numerico dispone di sorveglianze permanentemente attive che riconoscono tempestivamente eventuali errori verificatisi nel → CNC, nell'interfaccia (→ PLC) e a bordo macchina in modo tale da evitare danni gravi al pezzo, all'utensile o alla macchina. In caso di errore viene interrotta la lavorazione e vengono arrestati gli azionamenti, viene inoltre memorizzata la causa dell'errore e visualizzato l'allarme. Contemporaneamente viene segnalato al PLC che è presente un allarme CNC.

Geometria

Descrizione di un → pezzo nel → sistema di coordinate del pezzo.

Gestione del partprogram

La gestione dei programmi pezzo può essere organizzata in base ai → pezzi. L'estensione della memoria utente determina la quantità dei programmi e dati da gestire. Ogni file (programma e dati) può essere provvisto di un nome composto da max. 24 caratteri alfanumerici.

Gestione della velocità

Per poter ottenere una velocità accettabile anche con movimenti molto brevi, è possibile attivare per ogni blocco la preelaborazione anticipata di più blocchi (→ Look Ahead).

Gruppi di modi operativi

In un determinato momento tutti gli assi e mandrini vengono associati esattamente ad un canale. Ogni canale è associato ad un gruppo di modi operativi. Ai canali di un BAG è abbinato sempre lo stesso → modo operativo.

HIGHSTEP

Riassunto delle possibilità di programmazione per i → PLC del sistema AS300/AS400.

identificatore

Secondo DIN 66025 le parole vengono integrate con indicatori (nomi) per variabili (variabili di calcolo, variabili di sistema, variabili utente), per sottoprogrammi, per parole chiave e parole con più lettere di indirizzamento. Queste integrazioni sono concettualmente analoghe alle parole nella struttura del blocco. Gli indicatori devono essere univoci. Lo stesso indicatore non può essere utilizzato per oggetti differenti.

Impostazione metrica e in pollici delle quote

Nel programma di lavorazione, le quote e i valori di passo si possono programmare in pollici. Indipendentemente dal tipo di impostazione programmabile (G70/G71), il controllo numerico viene impostato su un sistema di base.

indicatore dell'asse

Secondo DIN 66217, gli assi vengono contrassegnati X, Y, Z per un -> sistema di coordinate destrorso ortogonale.

Gli -> assi rotanti intorno a X, Y, Z vengono contrassegnati con A, B e C. Gli assi supplementari paralleli agli assi fondamentali possono essere contrassegnati con altre lettere dell'alfabeto.

Indirizzo

L'indirizzo è un identificatore per un determinato operando o per un settore dello stesso, per es. un ingresso, un'uscita ecc.

Indirizzo assi

Vedere -> Indicatori assi

Ingressi/uscite digitali veloci

Tramite gli ingressi digitali si possono avviare, ad esempio, delle routine di programma CNC veloci (routine di interrupt). Tramite le uscite digitali CNC è possibile attivare delle funzioni di comando gestite da programma (SINUMERIK 840D).

Interfaccia multipoint (MPI)

L'interfaccia multi point (MPI) è costituita da una connessione D-Sub a 9-poli. Ad un'interfaccia multipoint è possibile collegare un numero parametrizzabile di apparecchi che comunicano tra di loro:

- dispositivi di programmazione (PG)
- sistemi di servizio e supervisione (HMI)
- altri controllori programmabili

Il blocco di parametri "Multipoint Interface MPI" della CPU contiene i -> parametri che definiscono le caratteristiche dell'interfaccia multipoint.

Interfaccia seriale V.24

Per l'immissione/emissione dei dati è disponibile una interfaccia seriale V.24 (RS232) sulla PCU 20, sulla PCU 50/70 sono disponibili due interfacce V.24. Con queste interfacce è possibile caricare e salvare sia i programmi di lavorazione sia i dati del costruttore e dell'utente.

Interpolator

Unità logica dell'-> NCK che, in funzione dei dati delle posizioni di destinazione nel partprogram, determina i valori intermedi per i movimenti dei singoli assi.

Interpolazione circolare

L' -> utensile deve muoversi tra punti definiti del profilo con un determinato avanzamento su un arco di cerchio e contemporaneamente lavorare il pezzo.

Interpolazione elicoidale

L'interpolazione elicoidale è particolarmente adatta per la realizzazione di filettature interne o esterne con frese sagomate e per la fresatura di cave di lubrificazione.

L'elica si compone di due movimenti combinati:

- movimento circolare in un piano
- ed un movimento lineare ortogonale a questo piano

Interpolazione lineare

L'utensile viene posizionato sul punto finale con un percorso rettilineo e contemporaneamente viene lavorato il pezzo.

interpolazione polinomica

Con l'interpolazione polinomica è possibile generare i più svariati andamenti di curve come funzioni **rettilinee, paraboliche o esponenziali** (SINUMERIK 840D).

Interpolazione Spline

Con l'interpolazione Spline il controllo numerico può generare un profilo curvilineo liscio partendo soltanto da pochi punti di appoggio preimpostati.

Intersezione preliminare

Si ha già il cambio blocco quando il percorso di contornitura si avvicina alla posizione finale di un delta predefinito.

Jog

Modo operativo del controllo numerico (funzionamento di messa a punto): nel modo operativo JOG è possibile effettuare la messa a punto della macchina. I singoli assi e mandrini possono essere mossi tramite i tasti direzionali nel funzionamento ad impulsi. Ulteriori funzioni del modo operativo Jog sono la -> Ricerca punto di riferimento, il -> Repos e il -> Preset (preimpostazione valore reale).

KÜ

Rapporto di trasmissione

Kv

Fattore di amplificazione dell'anello di posizione (guadagno); grandezza tecnica di un anello di regolazione

Lavorazione su piani inclinati

Lavorazioni di foratura e fresatura sulle superfici del pezzo non parallele al sistema di coordinate della macchina possono essere eseguite confortevolmente con l'ausilio della funzione "Lavorazione su piani inclinati".

Limitazione del campo di lavoro

Con la limitazione del campo di lavoro si può limitare il campo di posizionamento degli assi in aggiunta alla limitazione dei finecorsa. Per ogni asse è ammessa una coppia di valori per la definizione del campo di lavoro protetto.

Limitazione programmabile del campo di lavoro

Limitazione del campo di movimento dell'utensile in uno spazio definito da limitazioni programmate.

Linguaggio evoluto CNC

Il linguaggio evoluto offre: -> Variabili definite dall'utente, -> Variabili di sistema, -> Tecnica macro.

Look Ahead

Grazie alla funzione **Look Ahead**, tramite un numero parametrizzabile di blocchi di movimento "pre-elaborati", si ottiene una velocità di lavorazione ottimale.

Macchina

Settore operativo del controllo numerico

Maschiatura senza utensile compensato

Questa funzione consente di eseguire maschiature senza utensile compensato. Con il movimento di interpolazione del mandrino come asse rotante e dell'asse di foratura, vengono eseguite filettature esattamente fino alla profondità di foratura finale, ad es. filettatura cieca (presupposto: funzionamento del mandrino come asse).

Massa

Per massa si intende l'insieme di tutte le parti inattive di un'apparecchiatura elettrica collegate tra loro che non possono assumere tensioni pericolose al contatto neppure in caso di guasto.

MDA

Modo operativo del controllo numerico: Manual Data Automatic. Nel modo operativo MDA si possono impostare singoli blocchi di programmi o sequenze di blocchi senza alcun riferimento a un programma principale o sottoprogramma, che al termine possono essere eseguiti con il tasto Start-NC.

Memoria di backup

La memoria di back-up garantisce la memorizzazione delle aree di memoria della ->CPU senza memoria tampone Viene bufferizzato un numero parametrizzabile di temporizzatori, contatori, merker e byte di dati.

Memoria di caricamento

Nella CPU 314 del -> PLC, la memoria di caricamento corrisponde alla -> memoria di lavoro.

Memoria di correzione

Settore di dati del controllo numerico nel quale vengono inseriti i dati di correzione utensile.

Memoria di lavoro

La memoria di lavoro è una memoria RAM della -> CPU alla quale il processore accede durante l'elaborazione del programma utente.

Memoria di programma PLC

SINUMERIK 840D: nella memoria utente del PLC vengono inseriti il programma utente PLC ed i dati utente insieme al programma base PLC.

Memoria di sistema

La memoria di sistema è una memoria nella CPU nella quale vengono memorizzati i seguenti dati:

- dati necessari per il sistema operativo
- gli operandi tempi, contatori, indicatori

Memoria utente

Tutti i programmi e i dati come partprogram, sottoprogrammi, commenti, correzioni utensile, spostamenti origine/frame e dati utente di canale e programma possono essere memorizzati nella memoria utente CNC comune.

Messaggi

Tutte le segnalazioni programmate nel partprogram e gli -> allarmi riconosciuti dal sistema, vengono visualizzati come testo in chiaro sul pannello operatore. La visualizzazione di allarmi e messaggi avviene separatamente.

Modo operativo

Concetto esecutivo del funzionamento di un controllo numerico SINUMERIK. Sono stati definiti i modi operativi -> Jog, -> MDA, -> Automatico.

NC

Numerical Control: il controllo numerico NC comprendente tutti i componenti per la gestione di una macchina utensile: -> NCK, -> PLC, HMI, -> COM.

Nota

per i controlli SINUMERIK 840D sarebbe più corretto utilizzare il termine controllo CNC: computerized numerical control.

NCK

Numeric Control Kernel: Componente del controllo numerico NC che elabora i -> partprogram e che sostanzialmente coordina i movimenti della macchina utensile.

Nome degli assi

Vedere -> Indicatori assi

NRK

Numeric Robotic Kernel (sistema operativo dell' -> NCK)

NURBS

La gestione dei movimenti interni al controllo numerico e l'interpolazione vettoriale vengono eseguite sulla base di NURBS (Non Uniform Rational B-Splines). In questo modo internamente al controllo SINUMERIK 840D è disponibile un avanzamento univoco per tutti i tipi di interpolazione.

OEM

Per i costruttori di macchine che vogliono progettare una propria superficie operativa, oppure inserire funzioni tecnologiche nel controllo numerico, sono previsti degli spazi liberi per soluzioni individuali (applicazioni OEM) per il SINUMERIK 840D.

Override

Possibilità di accesso manuale o programmabile che consente all'utente di intervenire sugli avanzamenti o sui giri programmati per adattarli a un determinato pezzo o materiale.

Override avanzamento

La velocità programmata viene corretta in funzione della preimpostazione della velocità attuale sul → pannello di comando della macchina o dal PLC (0-200%). La velocità di avanzamento può essere corretta anche nel programma di lavorazione con un fattore percentuale programmabile (1-200%).

Parametri R

Parametro di calcolo che può essere definito e interrogato nel programma dal programmatore del → programma pezzo per qualsiasi scopo.

Parola dati

Un'unità di dati con dimensione di due byte nell'ambito di un → blocco dati del PLC.

Parole chiave

Parole con scrittura definita che hanno un significato definito nel linguaggio di programmazione del → programma pezzo.

Partprogram (programma pezzo)

Sequenza di istruzioni inviate al controllo numerico che insieme determinano l'esecuzione di un determinato → pezzo. Anche una determinata lavorazione su un determinato → pezzo grezzo.

Pezzo

Parte che deve essere approntata/lavorata dalla macchina utensile.

Pezzo grezzo

Particolare con cui si inizia la lavorazione di un pezzo.

PLC

Programmable Logic Control: → Controllore programmabile (PLC). Componente del → NC: interfaccia per l'elaborazione della logica di controllo della macchina utensile

Precomando, dinamico

Le imprecisioni del → profilo dovute a errori di inseguimento possono essere pressoché eliminate grazie al precomando dinamico in funzione dell'accelerazione. In questo modo è possibile ottenere una straordinaria precisione di lavorazione anche ad alte → velocità vettoriali. Il precomando può essere selezionato ed escluso mediante il → programma pezzo in modo specifico per asse.

Profilo

Profilo del → Pezzo

Profilo del pezzo

Profilo di riferimento del → pezzo da creare / eseguire.

Profilo finito

Profilo del pezzo finito. Vedere -> Pezzo grezzo.

Programma per il trasferimento dei dati PCIN

PCIN è un programma ausiliario per la trasmissione e la ricezione dei dati utente CNC tramite l'interfaccia seriale, ad es. partprogram, correzioni utensili, etc. Il programma PCIN funziona in MS-DOS su PC industriali standard.

Programma principale

-> Partprogram contrassegnato con un numero o un indicatore, dal quale possono essere richiamati altri programmi principali, sottoprogrammi oppure -> cicli.

Programma principale/sottoprogramma globale

Ogni programma principale/sottoprogramma globale può comparire una sola volta con il suo nome nella directory. Lo stesso nome non può essere riutilizzato come programma globale in altre directory con contenuti diversi.

Programma utente

I programmi utente per i sistemi di automazione S7-300 vengono approntati con il linguaggio di programmazione STEP 7. Il programma utente ha una struttura modulare ed è costituito da singoli blocchi.

I tipi di blocchi fondamentali sono:

blocchi di codice: questi blocchi contengono le istruzioni STEP7.

Blocchi dati: questi blocchi contengono costanti e variabili per il programma STEP7.

Programmazione del PLC

Il PLC viene programmato con il software **STEP 7**. Il software di programmazione STEP 7 si basa sul sistema operativo standard **WINDOWS** e contiene le funzioni della programmazione di STEP 5 con ulteriori sviluppi innovativi.

Pulsantiera di macchina

Pannello è possibile della macchina utensile con gli elementi operativi tasti, selettore rotativo, etc. ed inoltre semplici elementi di visualizzazione come LED. Essa consente di comandare direttamente la macchina utensile tramite il PLC.

Punto di riferimento

Punto della macchina utensile al quale fanno riferimento i trasduttori di misura degli assi di macchina.

Punto fisso della macchina

Punto della macchina utensile definito in modo univoco, ad es. punto di riferimento.

Punto zero macchina

Punto fisso della macchina utensile al quale si lasciano ricondurre tutti i trasduttori di misura (derivati).

punto zero pezzo

Il punto zero del pezzo rappresenta il punto iniziale del → sistema di coordinate pezzo e viene definito mediante distanze dal → punto zero della macchina.

Quota incrementale

Indicazione della lunghezza di movimento tramite un valore incrementale (quota incrementale). Il valore incrementale può essere inserito come → dato di setting o selezionato tramite i rispettivi tasti 10, 100, 1000, 10000.

Quote assolute

Indicazione della posizione finale di un movimento dell'asse con una quota riferita al punto zero del sistema di coordinate momentaneamente attivo. Vedere → Quota incrementale.

quote incrementali

Indicazione della posizione di arrivo di un movimento dell'asse con l'entità del percorso e la direzione rispetto a un punto già raggiunto. Vedere → Quote assolute.

Raggiungimento di un punto fisso

Le macchine utensili possono raggiungere dei punti fissi, come punti di cambio utensile, punti di carico, punti di cambio, ecc. Le coordinate di questi punti vengono inserite nel controllo numerico. Il controllo numerico muove gli assi interessati, possibilmente in → rapido.

Rapido

La velocità di movimento più elevata di un asse. Essa viene utilizzata, ad esempio, quando l'utensile da una posizione di riposo viene accostato al → profilo del pezzo o quando viene allontanato dallo stesso.

Rete

Una rete è un collegamento di più S7-300 ed altri terminali, per es. un PG, mediante → cavi di collegamento. Tramite la rete avviene lo scambio di dati tra le unità collegate.

Ricerca blocco

Per il test di programmi pezzo oppure dopo un'interruzione della lavorazione, con la funzione "Ricerca blocco" è possibile scegliere una qualsiasi posizione del programma pezzo, dalla quale deve partire o proseguire la lavorazione.

Ricerca punto fisso della macchina

Movimento verso un -> punto fisso predefinito della macchina.

Rotazione

Componente di un → frame che definisce una rotazione del sistema di coordinate attorno a un determinato angolo.

Routine di interrupt

Le routine di interrupt sono -> sottoprogrammi speciali che possono essere avviati mediante eventi (segnali esterni) dal processo di lavorazione. Un blocco del partprogram in corso di elaborazione viene interrotto e la posizione di interruzione degli assi viene memorizzata automaticamente.

Selettore a chiave

L'interruttore a chiave sul → pannello di comando della macchina ha 4 posizioni con funzioni assegnate dal sistema operativo del controllo numerico. L'interruttore a chiave è dotato di tre chiavi con colori diversi che possono essere estratte nelle posizioni definite.

Servizi

Settore operativo del controllo numerico

Settore TOA

Il settore TOA include tutti i dati dell'utensile e del magazzino. Di norma il settore, per quanto riguarda la portata dei dati, coincide con il settore → Canale. Tuttavia con i dati macchina è possibile determinare che più canali si dividano una → unità TOA, in modo tale che i canali abbiano a disposizione dei dati di gestione utensili comuni.

Sincronizzaz.

Istruzioni nei → programmi pezzo per il coordinamento dell'elaborazione nei vari → canali in determinati punti dell'elaborazione.

Sistema di coordinate

Vedere -> Sistema di coordinate macchina -> Sistema di coordinate pezzo

Sistema di coordinate base

Sistema di coordinate cartesiane che viene adattato con una trasformazione al sistema di coordinate macchina.

Nel -> partprogram il programmatore utilizza i nomi degli assi del sistema di coordinate base. Se non è attiva alcuna -> trasformazione, esso è parallelo al -> sistema di coordinate macchina. La differenza tra i due è rintracciabile negli indicatori assi.

Sistema di coordinate macchina

Sistema di coordinate riferito agli assi della macchina utensile.

Sistema di coordinate pezzo

Il sistema di coordinate pezzo (SCP) ha il suo punto iniziale nel → punto zero del pezzo. Tutte le quote e le direzioni programmate nel sistema di coordinate pezzo si riferiscono a questo sistema di coordinate.

Sistema di misura in pollici

Sistema di misura nel quale le distanze vengono definite in "pollici" o in sottomultipli di esso.

Sistema di misura metrico

Sistema di unità normalizzato: per le lunghezze, ad es. mm (millimetri), m (metri).

Softkey

Tasto la cui funzione è rappresentata in un campo del video che si adatta dinamicamente alla situazione operativa attuale. I tasti funzione di libero impiego (softkey) vengono abbinati via software a determinate funzioni.

Soglia di arresto preciso

Quando tutti gli assi di interpolazione hanno raggiunto la soglia di arresto preciso, il controllo numerico considera raggiunta la posizione di arrivo con esattezza. Si verifica quindi il passaggio al blocco successivo del → partprogram.

Sorveglianza del profilo

Per poter sorvegliare l'errore di inseguimento nell'ambito di una fascia di tolleranza definita, viene considerata come criterio di valutazione la precisione del profilo. Un errore di inseguimento superiore al limite consentito può dipendere, ad esempio, da un sovraccarico degli azionamenti. In questo caso viene attivato un allarme che arresta gli assi.

Sottoprogramma

Successione di istruzioni di un → programma pezzo che può essere richiamata più volte con differenti parametri di definizione. Un sottoprogramma viene richiamato da un programma principale. Ogni sottoprogramma può essere bloccato contro emissioni e visualizzazioni non autorizzate. → I cicli sono una forma di sottoprogrammi.

Sottoprogramma asincrono

Programma pezzo che può essere avviato in modo asincrono (indipendente) rispetto allo stato attuale del programma tramite un segnale di interrupt (ad es. segnale "Ingresso veloce NC").

Specularità

Con la specularità vengono invertiti i segni dei valori delle coordinate di un profilo rispetto ad un asse. Analogamente, nel contempo è possibile speculare più assi.

spostamento origine (SO)

Preimpostazione di un nuovo punto di riferimento per un sistema di coordinate con riferimento al punto zero attuale e a un -> frame.

1. Impostabile

SINUMERIK 840D: per ogni asse CNC è disponibile un numero progettabile di spostamenti origine impostabili. Gli spostamenti origine attivabili con funzioni G sono attivi alternativamente.

2. Esterno

Oltre a tutti gli spostamenti che definiscono la posizione del punto zero, può essere sovrapposto uno spostamento origine tramite volantino (traslazione DRF) o dal PLC.

3. Programmabile

Con l'istruzione TRANS è possibile programmare spostamenti origine per tutti gli assi vettoriali e di posizionamento.

Spostamento origine esterno

Spostamento origine preimpostato da ->PLC.

Struttura canalizzata

La struttura a canali consente l'elaborazione di -> programmi dei singoli canali in modo simultaneo o indipendente.

Superficie operativa

La superficie operativa (BOF) è il supporto di visualizzazione di un controllo CNC sotto forma di schermo. Essa è dotata di otto softkey orizzontali ed otto verticali.

Svincolo rapido dal profilo

Con l'intervento di un interrupt tramite il programma di lavoro del CNC può essere attivato un movimento che consente uno svincolo rapido dell'utensile dal profilo del pezzo in lavorazione. Inoltre può essere parametrizzato sia l'angolo di svincolo sia l'entità del percorso. Dopo lo svincolo rapido si può eseguire una routine di interrupt (SINUMERIK 840D).

Svincolo utensile orientato

RETTOOL: in caso di interruzione della lavorazione (ad es. per rottura dell'utensile) con un comando del programma è possibile svincolare l'utensile di un percorso definito con un orientamento preimpostabile.

Tabella di compensazione

Tabella con punti di appoggio. Fornisce per le posizioni prescelte dell'asse base i valori di compensazione dell'asse di compensazione.

Tecnica delle macro

Raggruppamento di singole istruzioni sotto un unico identificatore. Nel programma l'identificatore rappresenta il numero di istruzioni raggruppate.

Trasformazione

Spostamento addizionale o assoluto del punto zero di un asse.

Unità di ingresso/uscita digitale

Le unità digitali sono le interfacce per i segnali di processo binari.

Unità di periferia

Le unità di periferia rappresentano il collegamento tra la CPU e il processo.

Unità di periferia sono:

- → unità di ingresso/uscita digitali
- → unità di ingresso/uscita analogiche
- → unità di simulazione

Unità d'ingresso/uscita analogiche

Le unità d'ingresso/uscita analogiche sono le interfacce per i segnali analogici di processo.

Le unità di ingresso analogiche convertono grandezze di misura analogiche in valori digitali che possano essere elaborati dalla CPU.

Le unità di uscita analogiche convertono dei valori digitali in grandezze analogiche.

Unità TOA

Ogni → settore TOA può contenere più unità TOA. Il numero di unità TOA possibili è limitato dal numero massimo di → canali attivi. Un'unità TOA comprende esattamente un modulo di dati dell'utensile e un modulo di dati del magazzino. Inoltre può contenere anche un modulo di dati del supporto utensile (opzionale).

Utensile (UT)

Parte attiva della macchina utensile preposta alla lavorazione (ad es. utensile di tornitura, fresa, punta a forare, raggio LASER...).

Valore di compensazione

Differenza tra la posizione dell'asse misurata dal trasduttore e la posizione dell'asse programmata.

Variabile di sistema

Variabile esistente di un → programma pezzo senza definizione da parte del programmatore. Viene definita da un tipo di dati e dal nome della variabile che inizia con il carattere \$. Vedere → Variabili definite dall'utente.

Variabili definite dall'utente

Per qualsiasi impiego nel → partprogram o nel blocco dati (dati utente globali), si possono concordare delle variabili definite dagli utenti stessi. Una definizione contiene un'indicazione sul tipo di dati e sul nome della variabile. Vedere → Variabili di sistema.

Velocità limite

Velocità massima/minima (del mandrino): è possibile limitare la velocità massima di un mandrino predefinendo i dati macchina, il → PLC o i dati di setting.

Velocità vettoriale

La massima velocità programmabile dipende dalla risoluzione di impostazione. Con una risoluzione di 0,1 mm, ad esempio, la massima velocità vettoriale programmabile è di 1000 mm/min.

Volantino elettronico

Con l'aiuto di volantini elettronici è possibile spostare simultaneamente gli assi selezionati in funzionamento manuale. Il valore di ogni tacca dei volantini viene stabilito con la valenza degli incrementi dello spostamento origine esterno.

Indice analitico

\$

\$AA_COUP_ACT, 378, 403, 551
\$AA_COUP_OFFS, 551
\$AA_LEAD_SP, 402
\$AA_LEAD_SV, 402
\$AA_MOTEND, 213
\$AC_MARKER[n], 442
\$AC_PARAM, 443
\$AC_TIMER[n], 446
\$MC_COMPRESS_VELO_TOL, 409
\$P_TECCYCLE, 493
\$SA_LEAD_TYPE, 402
\$TC_CARR1...14, 363
\$TC_CARR18[m], 363, 367
\$TC_CARR24[m], 365

A

A, 308
A1, A2, 363, 365
A2, 262
A3, 262
A4, 262, 268
A5, 262, 268
ABS, 34
ACC, 549
Accelerazione in funzione del percorso
PUNCHACC, 520, 521
Accesso alle posizioni ed ai segmenti delle tabelle di curve, 394
Accoppiamento, 371, 377
accoppiamento al valore reale, 543
Accoppiamento assiale al valore master, 397
Accoppiamento del valore master, 481
 Accoppiamento al valore di riferimento e al valore reale, 398, 401
 da azioni sincrone statiche, 399
 Sincronizzazione asse master e asse slave, 400
Accoppiamento di velocità, 543
Accoppiamento tramite riferimento, 543
Accostamento sul punto vettoriale successivo, 424

ACOS, 34
Acquisizione dei valori di misura, 199
ACTFRAME, 221
ADISPOSA, 211
ALF, 65
Amax, 520
Amin, 520
Ampliamento del percorso di ricerca programmabile in un richiamo di sottoprogramma, 116
AND, 37
ANG, 600
Angoli di rotazione 1, 2, 363
Angolo di anticipo, 262
Angolo di Eulero, 354
Angolo di rotazione, 282
Angolo finale, 282
Angolo laterale, 262
Angolo RPY, 354
angolo tangente vettoriale, 488
aplim, 381
Apprendimento delle curve di compensazione, 538
APR, 146, 149, 151
APW, 146, 149, 151
APX, 151
Arresto, 566
Arresto autarchico tramite azionamento, 567
arrotondamento, 89
AS, 133
ASIN, 34
ASPLINE, 175
Asse
 dati utente locali, 571
Asse di pendolamento, 507
Asse F, 376
Asse inclinato, TRAANG, 251, 307
asse M, 376, 398
asse master, 398
Asse master, 371, 381, 390, 395
asse S, 398
asse slave, 398
Asse slave, 371, 381, 390, 395
Asse/mandrino di serraggio del pezzo/mandrino, 570
Assegnazione e avvio della routine di interrupt, 67
Assegnazioni, 32
Assi di collegamento, 571

Assi di comando, 468
Assi di orientamento, 261, 268, 271, 273
Assi geometrici commutabili, 323
Assi rotanti
 Vettori della distanza I1, I2, 363
 Vettori direzionali V1, V2, 363
assiale
 Avanzamento, 476
ATAN2, 34
Attivazione del controllo tangenziale, TANGON, 371
Attrito, 538
attuale
 Override, 488
 traslazione angolare, 551
AV, 547
Avanzamento assiale, 476
AX, 529, 530
AXCTSWE, 571
AXCTSWED, 571
AXIS, 18
AXNOME, 529, 530
AXSTRING, 45, 529, 530
AXTOCHAN, 77
AXTOSPI, 529, 530
Azione sincrona, 569
Azioni sincrone
 ASUP, 502
 Cambio del modo operativo, 500
 CANCEL, 503
 Condizioni marginali di eventi rilevanti, 500
 Fine programma, 501
 Liste di valori con SET e REP, 484
 NC-Stop, 501
 Power On, 500
 Reset, 501
 Ricerca blocco, 502
 Riposizionamento, 502
 variabile di esecuzione, 437
 Variabili di preelaborazione, 437
azioni sincrone al movimento
 Azioni, 435
 Panoramica, 451
 Programmazione, 430

B

B_AND, 37
B_NOT, 37
B_OR, 37
B_XOR, 37
B2, 262
B3, 262

B4, 262, 268
B5, 262, 268
BAUTO, 176
BFRAME, 219
Blocco di arresto, 418
blocco lettura, 455
Blocco parametri servo
 Criterio programmabile, 214
BLOCK, 113
BNAT, 176
BOOL, 18
BOUND, 34, 41
BSPLINE, 175
BTAN, 176
Buffer di preelaborazione, 410

C

C2, 262
C3, 262
C4, 262, 268
C5, 262, 268
CAC, 173
CACN, 173
CACP, 173
CALCDAT, 584, 597, 598
Calcolo dei dati del cerchio, 597
Calcolo del frame
 MEAFRAME, 235
CALL, 112, 113
CALLPATH, 116, 139
Cambio di stazione/posizione, 570
Cambio elettronico, 552
Campo di valori, 15
CANCEL, 430
Cancellare l'azione sincrona, 499
Cancellazione accoppiamenti, 550
Cancellazione del percorso residuo, 456, 506
Cancellazione del percorso residuo con
 preparazione, 456
Capacità di calcolo, 568
CASE, 53
CDC, 173
CFINE, 231
CHANDATA, 141
CHAR, 18
CHECKSUM, 88
CHKDNO, 360
CIC, 173
Cicli
 Cicli utente e cicli costruttore con programmi NC
 con lo stesso nome, 128

- Parametrizzazione dei cicli utente, 129
- Cicli tecnologici
 - strutturazioni in cascata, 496
- Cicli tecnologici, 491
 - controllo dell'esecuzione ciclica, ICYCOF, 495
 - Parametri di default con valori iniziali, 495
- Cicli tecnologici
 - in azioni sincrone blocco per blocco, 497
- Cicli tecnologici
 - Costrutti di controllo IF, 497
- Cicli tecnologici
 - Diramazione condizionata, 497
- Cicli tecnologici
 - Istruzioni di salto GOTOP, GOTOF, GOTOB, 497
- Cicli tecnologici
 - Salti incondizionali, 498
- cinematica risolta, 363, 367
- CLEARM, 61, 485
- Clock di interpolazione, 569
- CLRINT, 65
- CMIRROR, 34, 224
- COARSE, 543, 547, 550
- COARSEA, 211
- Codice G, 195
- Coefficiente polinomiale, 190
- Comandi di programmazione
 - Lista, 599, 640
- Comando tangenziale
 - Angolo limite tramite limitazione del campo di lavoro, 373
 - Definizione dell'asse slave e dell'asse master, 373
- COMCAD, 185
- COMPCURV, 185
- Compensazione dell'errore sul quadrante
 - Apprendimento successivo, 539
 - Attivazione della procedura di apprendimento, 538
 - Disattivazione della procedura di apprendimento, 538
- COMPLETE, 140, 141
- COMPOF, 185, 195
- COMPON, 185, 195, 290, 409
- Componente frame
 - FI, 227
 - MI, 227
 - SC, 227
 - TR, 227
- Componente frame RT, 227
- Componenti frame, 226
- Comportamento sui bordi della tabella, 390
- Compressore, 185, 197
- Compressore per orientamenti
 - COMPON, COMPCURV, 188
- Comunicazione link, 568
- Comunicazione NCU-NCU, 568
- concatenamento di stringhe, 47
- concatenate
 - Trasformazioni, 321
- Condizioni marginali per le trasformazioni, 320
- Conferma diretta dell'asse
 - GETD, 73
- Container
 - assi, 570
- Container assi, 570, 573
- CONTDCON, 584, 591
- CONTPRON, 584, 585, 595, 597
- Controllo della potenza del laser, 459
- Conversione di tipi, 45
- Coordinamento assi, 476
- Coordinamento dei programmi
 - Esempio, 62
 - Istruzioni per il coordinamento dei programmi, 59
 - Nomi dei canali, 61
 - Numeri di canale, 61
- Correzione del raggio utensile, 3D, 340
 - Cerchio di raccordo, 348
 - Comportamento sugli spigoli esterni, 357
 - Fresatura periferica, 342
 - Orientamento dell'utensile, 353
 - Profondità di inserzione ISD, 346
 - Programmazione dell'orientamento dell'utensile, 354
 - Punto di intersezione delle equidistanti, 348
 - Spigoli interni/esterni, 347
- Correzione online della lunghezza utensile, 293, 467
- Correzione raggio utensile
 - Fresatura periferica 3D con CUT3DC, 347
 - Fresatura periferica 3D con superfici limite, 351
 - Fresatura periferica 3D con utensili reali, 349
 - Fresatura periferica 3D senza superfici limite, 349
 - Rallentamento sugli spigoli, 209
- Correzione raggio utensile con utensili normalizzati
 - Profilo sulla superficie di lavorazione, 352
- Correzione utensile
 - Correzione sulla traiettoria, curvatura della traiettoria e profondità di inserzione, 345
 - En ligne, 333
 - Fresatura frontale, 340
 - Fresatura frontale 3D, 342
 - Memoria di correzione, 327
- Correzione utensile online, 465
- COS, 34
- COUPDEF, 542, 546
- COUPDEL, 542, 546
- COUPOF, 542, 550

COUPOFS, 542, 550
COUPON, 542, 548
COUPONC, 542
COUPRES, 542, 551
cov.com, cicli utente, 129
CP, 313
CPROT, 162
CPROTDEF, 160, 161
CROT, 34, 224
CSCALE, 34, 224
CSPLINE, 175
CTAB, 394, 395, 396
CTABDEF, 381, 383
CTABDEL, 381, 385, 387
CTABEND, 381, 383
CTABEXISTS, 387
CTABFNO, 387
CTABFPOL, 388
CTABFSEG, 388
CTABID, 387
CTABINV, 394, 395, 396
CTABISLOCK, 387
CTABLOCK, 387
CTABMAX, 390
CTABMEMTYP, 387
CTABMIN, 390
CTABMPOL, 388
CTABMSEG, 388
CTABNOMEM, 387
CTABPERIOD, 388
CTABPOLID, 388
CTABSEG, 388
CTABSEGID, 388
CTABSEV, 394, 395
CTABSSV, 394, 395
CTABTEP, 390, 391
CTABTEV, 390, 391
CTABTMAX, 391
CTABTMIN, 391
CTABTSP, 390, 391
CTABTSV, 390, 391
CTABUNLOCK, 387
CTRANS, 34, 224, 231
CUT3DC, 341, 346
CUT3DCC, 350
CUT3DCCD, 350
CUT3DF, 341
CUT3DFF, 341
CUT3DFS, 341
CUTCONOF, 338
CUTCONON, 338

D

D, 300
Dati setting, 506
DEF, 18, 23, 142, 491
DEFAULT, 53
DEFINE, 133, 491
Definizione coppia
 di mandrini sincroni, 546
Definizione dei dati utente, 142
Definizione del polinomio, 458
Definizione delle variabili, 17, 440
DELAYFSTOF, 412
DELAYFSTON, 412
DELDGT, 203, 456
DELETE, 81
DELT, 330
di arresto e svincolo
 Funzione ampliata, 558
di fine movimento
 Criterio programmabile, 210
Directory dei pezzi, 137
Directory principale dei pezzi, 137
DISABLE, 65
Disattivazione del controllo tangenziale, 371
Disattivazione della trasformazione
 TRAFOOF, 321
Disattivazione/riattivazione routine di interrupt, 68
DISPLOF, 127
DISPR, 420
DIV, 34
DO, 431, 435, 511
DRF, 232
DUPLO_NR, 330
DV, 547
DZERO, 362

E

EAUTO, 176
EG
 Cambio elettronico, 552
EGDEF, 553
EGDEL, 557
EGOFC, 557
EGOFS, 557
EGON, 553
EGONSYN, 553
EGONSYNE, 553
Elementi del profilo, punto di intersezione, 595
Elementi di comando, 431
Elementi in ombra, 590

Elemento del profilo, 587
 ELSE, 56
 ENABLE, 65
 ENAT, 176
 ENDFOR, 56
 ENDIF, 56
 ENDLOOP, 56
 Endpos, 510
 ENDPROC, 464
 ENDWHILE, 56
 ERG, 598
 ERRORE, 584
 Esclusione blocco singolo, 122
 Esecuzione dei colpi, 521
 Esecuzione del programma con buffer di preelaborazione, 410
 Esecuzione di un elemento di profilo, 597
 ETAN, 176
 EVERY, 434
 EXECSTRING, 31
 EXECTAB, 584, 597
 EXECUTE, 160, 161, 584
 EXP, 34
 EXTCALL, 118
 EXTERN, 100

F

FA, 508, 549
 FALSE, 16
 FCTDEF, 334, 458
 FCUB, 404
 FENDNORM, 210
 FGROU, 195
 FILEDATE, 86
 FILEINFO, 86
 FILESIZE, 86
 FILESTAT, 86
 FILETIME, 86
 FILETTO, 590, 594
 fine
 sincronismo, 543
 FINE, 543, 547
 FINEA, 211
 Finecorsa SW, 476
 FLIN, 404
 FMA, 614
 FNORM, 404
 FOR, 56, 58
 FPO, 404
 FPR, 558
 Frame

 assegnazione, 229
 Catene di frame, 229
 Concatenamento di frame, 244
 richiamo, 227
 FRAME, 18
 Frame attuale globale, 243
 Frame base NCU globali, 239
 Frame di base attuali del canale, 242
 Frame di base globale, 242
 Frame di base globali NCU attuali, 241
 Frame di sistema attuali, 241
 Frame globali NCU impostabili, 239
 Frame impostabile attuale, 243
 Frame programmabile attuale, 243
 Frame specifici per un canale, 239
 FRC, 613, 615
 FRCM, 615
 Fresa
 -punta (FS), 346
 -punto ausiliario (FH), 346
 Fresatura frontale 3D, 267
 Curvatura vettoriale tramite vettori normali alla superficie, 268
 Fresatura periferica, 341
 Fresatura periferica 3D con superfici limite, 349
 Fresatura periferica con superfici limite, 350
 FROM, 434
 FS, 543
 FTOCOF, 334
 FTOCON, 334
 Funzionamento a seguire tangenziale ottimizzato, 371
 Funzionamento generatore, 566
 Funzione ampliata di arresto e svincolo, 558
 Funzione di calcolo
 *, 34
 /, 34
 +, 34
 Funzione di conteggio dei pezzi, 574
 Funzione di interpretazione, 460
 Funzione di misura ampliata, 200, 312
 Funzione M
 a tre cifre, 133
 Funzione M / G a tre cifre, 133
 Funzioni ausiliarie, 454, 522
 Funzioni di calcolo, 34
 Funzioni OEM, 209

G

G[<indice gruppi>], 28
 G05, 311
 G07, 311

G1, 507
G25,G26, 373
G4, 506
G450, 348
G451, 348
G62, 210
G621, 210
G64, 92
G643, 196
Generazione di una routine di interrupt come sottoprogramma, 66
GEOAX, 323
Gestione dei movimenti, 577
Gestione utensili, 331
GET, 73
GET automatico, 76
GETACTTD, 361
GETD, 73
GETDNO, 360
GETSELT, 330
GETT, 330
Giochi, 538
GOTO, 53
GOTOB, 53
GOTOC, 53
GOTOF, 53
grado, 391, 395
grossolano
 funzionamento sincrono del riferimento, 543
 sincronismo, 543
GUD, 137, 140, 144, 145
 Attivazione automatica, 147
 Caricare le definizioni per GUD e macro, 148
 -Prima attivazione del file di definizione GUD, 147
 Scaricare le definizioni per GUD e macro, 147

I

I1,I2, 363
ICYCOF, 495
ICYCON, 495
ID, 430
identificatore asse di default
 inizializzazione delle variabili d'asse non definite del tipo AXIS, 441
IDS, 430
IF, 56
IF-ELSE-ENDIF, 57
IFRAME, 219
II1,II2, 513
Impiego ripetuto di tabelle di curve, 388
Impostazione del valore reale, 477

Incremento
 -asse, 511
 -movimento, 515
INDEX, 49
Indicatore max./min., 587
indicazione del percorso
 assoluta, 59
 relativo, 59, 60
Indice di campo, 24
Indirizzi OEM, 209
informazioni file FILExxxx
 lettura dalla memoria utente NCK, 86
Inibire determinate zone di programma per SERUPRO, 417
INIT, 60
Inizializzazione di variabili di campo, 484
INT, 18
Interpolazione del vettore di rotazione, 281, 287
Interpolazione di orientamento, 274, 288
interpolazione polinomica, 189
Interpolazione polinomica
 Polinomio denominatore, 193
Interpolazione Spline, 175, 197
 A-Spline, 179
 B-Spline, 180
 Compressore, 183
 C-Spline, 181
Interpolazioni lineare, 197
Interruzioni condizionate di sezioni di programma, 412
INTERSEC, 584, 595
Inversione
 iniziale, 510
IPOBRKA, 211
IPOENDA, 211
IPOSTOP, 543, 547, 549
IPTRLOCK, 417
IPTRUNLOCK, 417
ISAXIS, 529, 530
ISCOORD, 595
ISD, 341, 346
ISD (Insertion Depth), 340
ISFILE, 85
ISNUMBER, 46
ISOCALL, 114
ISPOINT, 595
ISPOINTS, 584
Istruzione CASE, 53
Istruzione di salto
 Istruzione CASE, 53
Istruzione di strutturazione per l'editor Step, 158
Istruzioni
 Lista, 599, 640

Istruzioni M, 520
ISVAR (), 535

J

JERKLIM, 577
Job di misura
 Misura con 2 sistemi di misura, 207
 Stato con MEASA, MEAWA, 207

K

KS, 371
KTAB, 585, 587, 591, 597

L

L..., 100
Label di attesa, 485
LEAD, 262, 354
LEADOF, 398
LEADON, 398
LIFTFAST, 65
Link
 NCU-, 568
Livelli di protezione
 Descrizione delle variabili di sistema ed esecuzione
 degli elementi del linguaggio NC, 150
 Modifica degli attributi degli elementi del
 linguaggio, 152
 Modifica dei dati macchina e dei dati di setting, 148
 Per dati utente, 145
LLIMIT, 458
LN, 34
LOCK, 430, 498
LOOP, 56
LOOP-ENDLOOP, 58
LS, 543
Lunghezza della stringa, 49
Lunghezza parziale, 510
Lunghezza parziale, 510

M

M, 365
M17, 95
MAC
 Attivazione automatica, 147
Macchina
 Stato del bloccaggio pezzo globale, 569

mandrini sincroni
 mandrini sincroni, 540
mandrino sincrono, 540
Mandrino sincrono
 Cancellazione accoppiamento, 550
 Comportamento al cambio blocco, 547
 Rapporto di trasmissione kÜ, 546
 Variabile di sistema, 16
mandrino slave
 Stato attuale dell'accoppiamento del, 551
MASLDEF, 580
MASLDEL, 580
MASLOF, 580
MASLOFS, 580
MASLON, 580
MATCH, 49
MAXVAL, 34, 41
MCALL, 110
MEAC, 201, 203
MEAFRAME, 235
MEAFRAME, 235
MEAFRAME, 238
MEAS, 198
MEASA, 201
MEAW, 198
MEAWA, 201, 483
Memoria
 Memoria dei programmi, 135
 Memoria di lavoro, 140
Memoria dei programmi, 135
 Creazione di una directory pezzi, 138
 Directory dei pezzi, 137
 Directory principale dei pezzi, 137
 Directory standard, 136
 Percorso di ricerca per il richiamo di un
 sottoprogramma, 139
 Selezione del pezzo, 138
 Tipi di file, 137
Memoria di lavoro, 140
 Nomi riservati dei blocchi, 144
 Programmi di inizializzazione, 140
 Settori dati, 140
MINDEX, 49
MINVAL, 34, 41
MIRROR, 220
Misura, 483
 Cancellazione del percorso residuo MEASA,
 MEAWA, 205
 Errori di programmazione rilevati, 208
 Eventi di trigger, 204
 Misura continua MEAC, 208
 Modo operativo, 204

Misure con tastatore in commutazione
 Programmazione di blocchi di misura, 198
 variabili di stato, 199
MMC, 576
MOD, 34
MODE
 Direzione di lavorazione, 585
 Tipo di lavorazione, 595
Modo, 510
Modo, 510
Modulo link, 569
MOV, 471
Movimenti del mandrino, 478
Movimenti di posizionamento, 468
Movimento asse singolo, 527
Movimento cartesiano PTP, 252
Movimento di pendolamento
 Campo di inversione, 513
 Incremento sul punto di inversione, 515
 Punto di inversione, 513
 Suppressione dell'incremento, 513
MPF, 137
MU, 310
MZ, 310

N

n, 395
 Numero della tabella di curve, 381, 388
 Numero delle tabelle di curve, 391
 Numero frame, 227
NCU collegate in rete, 568
NEWCONF, 78
NEWT, 330
NN, 585
NOC, 543, 547
nomi di variabili GUD predefinite, 440
NOT, 37
NPROT, 163
NPROTDEF, 160, 161
NR, 598
NR_ELEM_OMBRA, 586, 588
NUMBER, 46
Numeri D
 liberamente assegnabile, 359
 Rilevamento del numero T, 361
 Rinomina, 360
 verificare, 360
Numero dell'inserto, 359
Numero di pezzi, fisso, 56
Numero identificativo, 432

O

OEMIPO1/2, 209
OF, 54
OFFN, 297, 300
Offset angolari/incrementi angolari degli assi
 rotanti, 365
Offset degli assi rotanti, 365
Offset profilo normale OFFN, 305
Operatori di confronto
 <, 37
 <=, 37
 <>, 37
 >, 37
 >=, 37
Operatori di confronto, 37
 ==, 37
Operatori logici, 36
Operatori logici a bit, 37
Operatori logici e di confronto, 36
Operazioni su stringhe, 44
OR, 37
ORIAxes, 287
ORIAxes, 271
ORIC, 354
ORICONCCW, 274, 288
ORICONCW, 274, 288
ORICONIO, 274, 288
ORICONTO, 274, 288
ORICURVE, 277, 288
ORID, 354
Orientamento dell'utensile, 353
Orientamento relativo al percorso
 ;rotazioni dell'utensile, 285
 Inserimento di blocchi intermedi, 289
 Rotazione del vettore di orientamento, 286
 Rotazione dell'orientamento utensile, 286
ORIEULER, 287
ORIEULER, 272
ORIMKS, 269, 354
ORIPATH, 286, 627
ORIPATHS, 286, 289, 627
ORIPANE, 274, 288
ORIRESET(A, B, C), 260
ORIROTA, 282
ORIROTC, 282, 287
ORIROTR, 282
ORIROTT, 282
ORIRPY, 287
ORIRPY, 272
ORIRPY2, 272
ORIS, 354
ORIVECT, 287

ORIVECT, 271
 ORIVIRT1, 287
 ORIVIRT1, 272
 ORIVIRT2, 287
 ORIVIRT2, 272
 ORIWKS, 269, 354
 OS, 505, 506
 OSC, 354
 OSCILL, 510, 513
 OSCTRL, 505, 508
 OSD, 354
 OSE, 506, 509
 OSNSC, 506, 510
 OSOF, 354
 OSP, 508
 OSP1, 505, 510
 OSP2, 505, 510
 OSS, 354
 OSSE, 354
 OST, 355, 506
 OST1, 505, 510
 OST2, 505, 510
 Override, 516
 OVRA, 549

P

P, 108
 Panoramica
 frame attivi nel canale, 241
 parametri Call by Value per i cicli tecnologici, 494
 Parametri di calcolo, 15
 Parametri per azioni sincrone, 443
 Parametri R, 444
 Parola chiave, 433
 Partprogram, 569
 Partprogram (partprogram), 571
 Passate di spegnifiamma, 506
 PCALL, 115
 PDELAYOF, 520
 PDELAYON, 520
 Pendolamento
 Attivazione/disattivazione del pendolamento, 505
 Definizione della sequenza di movimento, 508
 gestione del pendolamento da azioni sincrone, 510
 Incremento parziale, 513
 Pendolamento asincrono, 505
 Pendolamento sincrono, 510
 Pendolamento asincrono, 505
 Pendolamento sincrono
 Abbinamento dell'asse di pendolamento e dell'asse
 di incremento, 513

Analisi, clock IPO, 516
 Arresto sul punto di inversione, 515
 Azioni sincrone, 513
 Definizione degli incrementi, 513
 Incremento nel campo di inversione, 514
 Movimento di incremento, 514
 Successivo incremento parziale, 516
 PFRAME, 220
 PHI, 280
 PKT, 598
 PL, 177, 189
 PO, 189
 PO[PHI], 280, 286
 PO[PSI], 280
 PO[THT], 280, 286
 PO[XH], 280
 PO[YH], 280
 PO[ZH], 280
 POLF, 563
 POLFA, 563
 POLFMASK, 563
 POLFMLIN, 563
 Polinomio
 -Interpolazione, 197
 POLINOMIO, 590, 594
 Polinomio al massimo di 5° grado, 383
 Polinomio denominatore, 193
 Polinomio F-Wort, 195
 POLY, 189
 POLYPATH, 189
 PON, 526
 PONS50, 520
 Portautensili
 Cancellazione/modifica/lettura dei dati, 367
 -Cinematica, 363
 Portautensili orientabili
 Numero del portautensili, 365
 Parametri degli assi rotanti, 365
 Parametri per l'utente, 365
 Variabile di sistema, 363
 POS, 469
 POSFS, 548
 POSFS POSLS, 543
 posizionamento asse, 469
 posizione di riferimento impostabile, 470
 Posizionamento su riscontro fisso FXS e
 FOCON/FOCOF, 486
 Posizione base dell'orientamento dell'utensile
 ORIRESET, 260
 Posizione di disattivazione, 550
 Posizione minima/massima dell'asse rotante, 365
 Posizioni singolari, 270

POSP, 510
POSRANGE, 470
POT, 34
Preparazione del profilo, 585, 591
 Elementi in ombra, 590
PREPRO, 128
PRESETON, 234, 478
Primo frame di base nel canale, 240
PRIO, 65
PROC, 95
Programma di inizializzazione, 140
 Caricamento del programma di inizializzazione, 142
 Creazione del programma di inizializzazione, 140
 Definizione dei dati utente GUD, 142
 Salvataggio del programma di inizializzazione, 141
Programmazione del percorso di ricerca per il richiamo di sottoprogrammi, 139
Programmazione dell'asse inclinato
 G05, G07, 310
Programmazione delle rotazioni del vettore di orientamento tramite THETA, 281
Programmazione dell'orientamento, 272, 287
Programmazione indiretta, 28
Programmazione indiretta dei codici G, 28
PSFS, 543
PSI, 280
PTP, 313, 317
PTP con TRANSMIT, 316
PTPG0, 317
PUNCHACC, 520
Puntatore di interruzione automatico, 419
puntatore di interruzione programmabile, 417
Punti di inversione pendolamento, 508
Punto d'intersezione fra elementi del profilo, 584
Punzonatura, 519, 522
Punzonatura con ritardo Off, 520
Punzonatura con ritardo ON, 520
punzonatura ON, 520
Punzonatura, roditura OFF, 520
PUTFTOC, 334
PUTFTOCF, 334
PW, 177

Q

QECDAT.MPF, 539
QECLRN.SPF, 539
QECLRNOF, 539
QECLRNON, 539
QECTEST.MPF, 539
QFK, 538

R

Raggiungimento di posizioni codificate, 173
Raggruppamento di trascinamento, 376
Raggruppamento spline, 183
Rallentamento d'angolo su tutti gli spigoli, 210
Rallentamento sugli spigoli interni, 210
Rapporto di trasmissione, 546
RDISABLE, 455
READ, 83
REAL, 18
Reazioni autarchiche degli azionamenti, 560
Reazioni controllate dall'NC, 564
REDEF, 149, 151
Refpos, 471
Regolazione AC, moltiplicativa, 462
Regolazione AC, addizionale, 461
Regolazione della distanza, 463
RELEASE, 73
REP, 22, 484
REPEAT, 56, 58
REPOS, 64, 66
REPOSA, 420
REPOSH, 420
REPOSHA, 420
REPOSL, 66, 420
REPOSQ, 420
REPOSQA, 420
RESET, 498
RET, 92, 95, 104, 122
RET (<Nr.blocco/Label>, < >, < >), 104
Riaccostamento al profilo, 419
 Accostamento con nuovo utensile, 425
 Accostamento con quarto di cerchio, 421
 Accostamento con semicerchio, 422
 Accostamento su una retta, 420
 Punto di riposizionamento, 422
Ricerca di caratteri, 49
Richiamo di un sottoprogramma, percorso di ricerca, 139
Richiamo indiretto di sottoprogrammi, 30
Richiamo indiretto di un programma programmato in linguaggio ISO con ISOCALL, 114
Richiamo sottoprogramma
 indiretto, 30
Richiamo sottoprogramma con indicazione del percorso e parametri, 115
Riferimento angolare, 548
Riferimento vettoriale
 Assi di contornitura, 197
 Assi FGROU, 195
 Avanzamento vettoriale, 197
 Blocchi di filettatura, 197

Condizioni marginali, 197
 Gruppo codici G, 195
 Interpolazione circolare e lineare, 197
 Parametro curva, 195
 Riferimento vettoriale impostabile, 195
 Rilevazione del sovraccarico, 489
 RINDEX, 49
 Ripetizione di una parte di programma con
 programmazione indiretta CALL, 112
 Ripetizione programma, 108
 risultante
 Override, 489
 Risultati delle misure per MEASA, MEAWA, 206
 Ritorno parametrizzabile al sottoprogramma, 104
 RMB, 420
 RME, 420
 RMI, 420
 Roditura, 519, 522
 roditura ON, 520
 ROUND, 34
 ROUNDUP, 89
 routine di conversione, 438
 Routine di interrupt, 64
 Definizione della sequenza, 68
 Direzione di movimento programmabile, 65
 Memorizzazione della posizione di interruzione, 67
 Svincolo rapido dal profilo, 69
 RPY, 354

S

S1, S2, 546, 551
 SAVE, 67, 94
 SBL1, 126
 SBL2, 126
 SBL3, 126
 SBLOF, 122
 SBLON, 122
 scambio assi
 Abilitazione asse, 76
 Accettazione asse, 76
 GET, 73
 RELEASE, 73
 senza sincronizzazione, 75
 senza stop preelaborazione, 76
 Scambio assi
 AXTOCHAN, 77
 Impostazione variabile del comportamento, 76
 Premesse, 75
 richiedere e abilitare tramite azioni sincrone, 472
 Scambio mandrini
 GET, 73

RELEASE, 73
 SCP, 137, 516
 SCPARA, 214
 SD, 177
 SEFORM, 158
 Segnalazione di errore, 584
 Selezione di un singolo carattere, 51
 Selezione di una parte di stringa, 51
 SET, 22, 484
 SETAL, 485
 SETDNO, 360
 SETINT, 65
 SETM, 60, 485
 SETPIECE, 330
 Settori di protezioni
 attiva, disattiva, 162
 Attivazione multipla, 165
 Definizione, 160
 Definizioni sulla macchina, 159
 Descrizione del profilo dei settori di protezione, 161
 Elementi del profilo consentiti, 161
 Piano di lavoro selezionato, 161
 Settori di protezione specifici della macchina, 160
 Settori di protezione specifici per canale, 160
 Spostamento, 165
 Stato di attivazione, 165
 Stato dopo l'avviamento, 165
 Settori progettabili dei parametri, 440
 Sgrossatura, 583
 Simulazione del valore master, 402
 SIN, 34
 Sincronismo di posizione, 540
 Smax, 520
 Smin, 520
 SON, 520, 525, 526
 SONS50, 520
 Sottoprogrammi, 91
 Annidamento, 93
 Meccanismo SAVE, 94
 Richiamo indiretto di sottoprogrammi, 112
 Richiamo sottoprogramma, 99
 Richiamo sottoprogramma modale, 110
 Ripetizione programma, 108
 Sottoprogrammi con assegnazione di parametri, 99
 Sottoprogrammi con assegnazione di parametri
 Definizione del campo, 98
 Esempio di assegnazione di parametri tra
 programma principale e sottoprogramma, 98
 Sottosquadri, 585
 Sovrascrittura di tabelle per curve, 389
 SPATH, 195
 SPF, 137

SPI, 529, 530, 549
Spianamento del percorso di orientamento, 286, 289
Spianatura
 Lavorazione esterna, 585
 Lavorazione interna, 585
SPIF1, 634
SPIF2, 634
SPLINE, 590, 594
SPLINEPATH, 183
SPN, 522
SPOF, 520
SPOS, 548
Spostamento DRF, 232
spostamento origine (SO)
 PRESETON, 234
 Spostamento origine esterno, 233
 Traslazione con volantino, 232
Spostamento origine esterno, 233
SPP, 522
SQRT, 34
SR, 635
SRA, 635
ST, 635
STA, 635
START, 60
Start/stop asse, 471
STARTFIFO, 410
STAT, 313, 317
Stato del tastatore di misura, 207
Stato dell'accoppiamento, 403
Stop preelaborazione, 456
STOPFIFO, 410
STOPRE, 198, 201, 202, 410, 507
STRING, 18
STRINGFELD, 44
STRINGIS, 531
 ambito della verifica, 531
 indirizzi NC, 533
 Informazioni dettagliate, 533
 informazioni di base, 532
 Set di istruzioni di linguaggio NC, 532
 Valori di ritorno, 532
 verifiche speciali, 531
STRINGVAR, 44
STRLEN, 49
Strutture di controllo, 55
 Condizioni marginali, 57
 Influenza sui tempi di elaborazione, 57
SUBSTR, 51
Suddivisione automatica del percorso, 522
suddivisione del percorso, 526
Suddivisione del percorso per assi di contornitura, 525

Supporto circuito intermedio, 566
Svincolo, 564
Svincolo autarchico tramite azionamento, 568
SYNFCT, 460

T

Tabella dei profili, 585, 591
Tabelle delle curve, 380
 Lettura in azioni sincrone, 394
 Tabella di curve non periodica, 392
 Tabella di curve periodica, 393
Tabelle di curve, 385
TABNAME, 585, 591, 597
TABNAME1, 595
TABNAME2, 595
TAN, 34
TANG, 371, 373
TANGDEL, 371
TANGOF, 371
TANGON, 371
TE, 201
Tecnica delle macro, 132, 520
Tempo di esecuzione del programma, 573
Tempo di sosta, 506
Tempo necessario
 Azioni sincrone, 489
THETA, 280, 282
TILT, 262, 354
Tipi di trasformazione
 Funzione generale, 247
Tipi di utensili
 Forme degli utensili per fresatura, dati utensile, 343
Tipo di accoppiamento, 543
 AV, 543
 DV, 543
 VV, 543
Tipo di cinematica, 367
tipo di cinematica M, 367
tipo di cinematica P, 367
tipo di cinematica T, 367
Tipo di variabile, 18
TIPO_LAVORAZ, 585
TLIFT, 371
TOFFOF, 293, 467
TOFFON, 293, 467
TOWER, 48
Tornitura longitudinale
 Lavorazione esterna, 585
 Lavorazione interna, 585
Torsione, 538
TOUPPER, 48

TRAANG, 251, 307
 TRACON, 322
 TRACYL, 251, 299, 300, 305
 TRAFOOF, 297, 300, 308, 321, 322
 TRAILOF, 376
 TRAILON, 376
 TRANSMIT, 251, 295, 297, 298, 317
 TRAORI, 255, 258
 Trascinamento, 375, 479
 Assi trascinati, 378
 fattore di accoppiamento, 378
 Trasformazione a 5 assi
 Programmazione del vettore direzionale, 265
 Programmazione dell'orientamento utensile con LEAD e TILT, 266
 Trasformazione a cinque assi
 Programmazione con angoli di Eulero, 264
 Programmazione con angoli RPY, 265
 Programmazione della curvatura vettoriale nei vettori normali alla superficie, 267
 Programmazione tramite LEAD/TILT, 261
 Trasformazione cinematica TRANSMIT, TRACYL e TRAANG, 251
 Trasformazione con asse lineare orientabile, 257
 Trasformazione curvilinea di una superficie cilindrica, 299
 Offset profilo normale OFFN, 305
 Trasformazione dell'orientamento TRAORI
 Cinematica di macchina, 250
 Movimenti di traslazione e movimenti di orientamento, 249
 Programmazione dell'orientamento, 259
 Trasformazione generica a 5/6 assi, 251
 Varianti della programmazione dell'orientamento, 260
 Trasformazione per asse inclinato, 307
 Trasformazione polare, 251
 Trasformazione su superficie cilindrica, 251
 Trasformazione TRACYL, 300
 Trasformazione TRANSMIT, 297
 Trasformazione TRAORI, 258
 Trasformazione, trasformazioni a tre assi, trasformazioni a quattro assi, 258
 Trasformazioni
 Posizione base, indipendente dalla cinematica, dell'orientamento dell'utensile, 248
 Trasformazione a tre, quattro e cinque assi TRAORI, 248
 Trasformazione dell'orientamento, 248
 trasformazioni cinematiche, 248
 Trasformazioni concatenate, 249
 Traslazione del preset, 234

Traslazione fine, 230
 Traslazione grossolana, 230
 Tratto di percorso, 522
 Trovare e rilevare i settori senza possibilità di ricerca, 418
 TRUE, 16
 TRUNC, 34, 39
 TU, 313, 317

U

U1,U2, 513
 uc.com, cicli utente, 130
 ULIMIT, 458
 UNLOCK, 430, 498
 UNTIL, 56, 58
 UPATH, 195

V

V1,V2, 363
 Valutazione della fresatura periferica 3D su spigoli esterni
 Andamento punto di intersezione, 348
 VAR, 95
 Variabile
 Assegnazioni, 32
 Programmazione indiretta dei codici G, 28
 Variabile di sistema, 15, 569
 globali, 569
 Variabile FIFO, 447
 Variabile frame
 Assegnazione alle istruzioni G G54 ... G599, 223
 Spostamenti origine G54 ... G599, 223
 Variabile frame predefinita, 219, 228
 variabile GUD per azioni sincrone
 variabile GUD definita dall'utente, 440
 Variabile Integer/Real, 28
 Variabile REAL come variabile di conteggio in un loop FOR, 57
 Variabile timer, 446
 Variabili, 15
 Conversione di tipi, 43
 definite dall'utente, 15
 Programmazione indiretta, 28
 Tipi di variabili, 15
 Variabile di sistema, 16
 Variabili definite dall'utente, 17
 Variabili di calcolo, 16
 Variabili di sistema, 437
 Variabili frame, 217

Assegnazione di valori, 224
Definizione di nuovi frame, 229
Richiamo di trasformazioni delle coordinate, 217
Variabili link
 globali, 569
Variabili merker, 442
VARIB, 598
VELOLIM, 578
Visualizzazione blocco, 114, 126
Visualizzazione blocco attuale, 126
Visualizzazione dell'ultimo numero di blocco
 programmato, 114

W

WAIT, 60
WAITC, 542, 549
WAITE, 60

WAITM, 60
WAITMC, 60
WALIMON, 373
WHEN, 433
WHEN-DO, 511, 514
WHENEVER, 433
WHENEVER-DO, 511, 514
WHILE, 56, 58
Winlimit, 471
WPD, 137
WRITE, 79
WZ, 330

X

x, 330
XOR, 37