

CNC Serie 10 Manuale di Programmazione

Code: 45004456R
Rev. 16

PUBBLICAZIONE EMESSA DA:

OSAI S.p.A.
Via Torino, 14 - 10010 Barone Canavese (TO) – Italy

Tel. +39-0119899711

Web: www.osai.it

e-mail: sales@osai.it
service@osai.it

Copyright © 2001-2005 by OSAI
Tutti i diritti riservati

Edizione: Giugno 2005

INFORMAZIONI IMPORTANTI PER L'UTENTE

La OSAI si riserva il diritto di apportare modifiche al prodotto descritto in questo manuale in qualsiasi momento e senza preavviso.

Il presente manuale è stato preparato da OSAI esclusivamente per l'uso da parte dei propri clienti garantendo che esso costituisce, alla data di edizione, la documentazione più aggiornata relativa al prodotto.

È inteso che l'uso del manuale avviene da parte dell'utente sotto la propria responsabilità.

Nessuna ulteriore garanzia viene pertanto prestata da OSAI, in particolare per eventuali imperfezioni, incompletezze e/o difficoltà operative, restando espressamente esclusa ogni sua responsabilità per danni diretti o indiretti comunque derivanti dall'uso di tale documentazione.

AGGIORNAMENTI RELATIVI ALLA PRESENTE EMISSIONE

Generalità

La presente pubblicazione viene emessa con il rilascio della Release Software 7.5 (E69).

PAGINA	TIPO DI AGGIORNAMENTO
INDICE	Aggiornato
CAP. 2	
pag. 97	Aggiunta descrizione relativa alla funzione UPR-Rotazione della terna cartesiana: sintassi – tipo di rotazione 5
pag. 98-99	Aggiunte nuove descrizioni: hor, ver; Tipo_req; pos-xx-H; pos-xx-V; n; ref-pos-xx-H; ref-pos-xx-V. Aggiunto “ATTENZIONE” con relativa descrizione
pag. 102	Aggiunta nota “IMPORTANTE” relativa alla release 7.4
pag. 107	Aggiunto esempio n.9 “Utilizzo dell'UPR di tipo 5”
pag. 133	Modificata descrizione “Programmazione dei parametri “m”, “n”, “o” e “u”, “v”, “w” (vettori)”
CAP. 5	
pag. 5	Aggiunta funzione: GTS-Condivisione del mandrino
CAP. 8	
pag. 13	Aggiunta descrizione alla funzione: “G84-ciclo di maschiatura senza trasduttore”
pag. 20	Aggiunta descrizione alla funzione: “G86-ciclo di barenatura”
APP. A	
pag. 8	Aggiunto codice triletterale GTS
APP. B	
pag. 8	Aggiunta descrizione nel messaggio NC094
pag. 12	Aggiunta descrizione nel messaggio NC135
	Aggiunta descrizione nel messaggio NC137
pag. 16	Aggiunti nuovi messaggi: NC290- NC291- NC292- NC293- NC294- NC295- NC296- NC297.
	Modificata descrizione del messaggio NC320
Pag. 17	Aggiunta descrizione nel messaggio NC331

PREFAZIONE

Questo manuale descrive le procedure utilizzate per la scrittura di part program con il sistema CNC Serie 10. Vengono fornite ai programmatori tutte le informazioni necessarie per la creazione di programmi per il controllo della macchina.

RIFERIMENTI

Per ulteriori informazioni:

- CNC Serie 10 Manuale di Caratterizzazione Software
- CNC Serie 10 Manuale di Uso

I capitoli di questo manuale descrivono gli elementi del linguaggio (comandi e funzioni) utilizzati per trattare una specifica attività, es. programmazione degli assi, programmazione degli utensili, gestione della tastatura. Per meglio assistere il programmatore, la descrizione dei comandi è stata integrata con degli esempi di programmazione.

SOMMARIO

1. Programmare con il sistema Serie 10

Il capitolo contiene regole di carattere generale per la programmazione secondo lo standard della International Standard Organization (ISO). Fornisce anche una panoramica dell'ambiente di programmazione ed un sommario dei codici più usati nella scrittura dei programmi.

2. Programmazione degli assi

Il capitolo descrive le procedure per la programmazione degli assi. Vengono forniti i codici G ed i comandi estesi (con relative caratteristiche). Alcuni esempi completano la descrizione del comando e suggeriscono come programmare i principali tipi di movimento.

3. Programmazione di utensili e correttori

Il capitolo descrive le procedure per la programmazione degli utensili e fornisce le funzioni e le istruzioni impiegate nelle operazioni con gli utensili.

4. Compensazione diametro utensile

Il capitolo descrive la compensazione dell'utensile. Vengono fornite le operazioni e i codici G usati nella compensazione dell'utensile con le relative caratteristiche e diversi esempi.

5. Programmazione del mandrino

Il capitolo descrive la programmazione del mandrino. Vengono forniti i codici G ed i comandi estesi utilizzati in questa attività, con le relative caratteristiche. La descrizione dei comandi è supportata da diversi esempi che suggeriscono come affrontare i casi più frequenti.

6. Funzioni ausiliarie M

Il capitolo descrive le funzioni ausiliarie e fornisce un elenco delle funzioni M disponibili con il loro significato e caratteristiche.

7. Programmazione parametrica

Il capitolo tratta speciali applicazioni di programmazione che utilizzano le variabili locali e di sistema.

8. Cicli fissi

Il capitolo fornisce una descrizione dei cicli fissi disponibili sul sistema, con i relativi codici G e i comandi estesi. La descrizione dei cicli è supportata da diversi esempi relativi all'utilizzo dei cicli fissi all'interno del part program.

9. Paramacro

Il capitolo descrive l'utilizzo delle Paramacro all'interno dei programmi.

10. Cicli di tastatura

Il capitolo fornisce una descrizione dei cicli di tastatura disponibili sul sistema, con i relativi codici G e i comandi estesi. La descrizione è completata da esempi di programmazione.

11. Gestione del video

Il capitolo contiene i comandi usati per la gestione del video da part program. La descrizione del comando è completata da esempi.

12. Modifica della sequenza di esecuzione del programma

Il capitolo descrive i comandi usati per modificare la sequenza di esecuzione di un programma: comandi di salto, ripetizione di blocchi ed esecuzione di sottoprogrammi. Vengono trattati inoltre i comandi per il rilascio e la messa in attesa su segnale del part program

13. High Speed Machining

In questo capitolo vengono descritte le prestazioni relative alla fresatura ad alta velocità su macchine utensili a 3 assi.

14. Comandi relativi alla gestione multiprocesso.

In questo capitolo vengono illustrate le potenzialità di multi processo del CNC Serie 10.

15. Programmazione geometrica ad alto livello (GTL)

Questo capitolo contiene la serie di istruzioni di programmazione disponibile con l'utility GTL.

16. Cicli di lavorazione per sistemi di tornitura

Questo capitolo fornisce le istruzioni per la programmazione dei macrocicli di sgrossatura, filettatura e taglio gola.

A. Sommario dei caratteri e dei comandi

L'appendice A fornisce un sommario dei caratteri ammessi nel sistema e gli elenchi dei codici G, delle funzioni matematiche e dei comandi estesi utilizzati.

B. Messaggi d'errore

L'appendice B fornisce l'elenco dei messaggi d'errore che si possono verificare durante la programmazione.

C. Gestione errori da part program

COMANDI

I comandi vengono trattati nei capitoli che ne descrivono l'attività specifica. Per ciascun comando vengono fornite le seguenti informazioni:

- Nome del comando
- Funzionalità
- Sintassi
- Parametri
- Caratteristiche e note
- Esempi

Dove possibile, gli esempi riportano parti di programma e diagrammi che illustrano il modo in cui operano i comandi.

Convenzioni sintattiche

Nella descrizione dei comandi vengono illustrate le seguenti convenzioni:

SIMBOLO	SIGNIFICATO
[]	Le parentesi quadre racchiudono dati opzionali. Non introdurre le parentesi quadre nel comando vero e proprio.
{ }	Le parentesi graffe indicano che l'elemento in esse racchiuso può essere ripetuto più volte. L'elemento può anche essere descritto come una serie di dati alternativi, cioè soltanto uno di essi può essere introdotto. I dati alternativi sono separati da una barra verticale (). Non introdurre le parentesi graffe nel comando vero e proprio.
	La barra verticale separa i dati alternativi. Non introdurre la barra nel comando vero e proprio.

Le parole chiave vengono scritte in neretto e devono essere digitate come vengono rappresentate nella descrizione sintattica. I parametri che devono essere passati con i comandi vengono indicati da un codice mnemonico scritto in corsivo. I valori appropriati dei parametri devono essere digitati al posto del codice mnemonico. Lo zero può essere omesso; per esempio, si può programmare G00 come G; G01 come G1.

Esempio:

(SCF, [value])

Il triletterale SCF, la virgola e la parentesi sono parole chiave e devono essere digitate nell'ordine descritto. Value è il nome di un parametro e deve essere sostituito da un valore specifico. Le parentesi quadre indicano che value è un valore opzionale.

Avvertenze

Per un uso corretto del sistema, è importante seguire le indicazioni contenute nel manuale e, inoltre, porre particolare attenzione agli argomenti contraddistinti da: ATTENZIONE, AVVERTENZA o IMPORTANTE.



E' associato a fatti o circostanze che possono causare danni al sistema, all'apparecchiatura o agli operatori.



E' associato ad informazioni da tenere in considerazione per evitare danni all'apparecchiatura in generale.



E' associato ad operazioni che occorre eseguire attentamente per assicurare il pieno successo dell'applicazione.

Terminologia

Di seguito vengono spiegati alcuni termini particolari utilizzati nel manuale.

Controllo	Si riferisce all'unità di controllo numerico Serie 10 comprendente il pannello frontale e l'unità base.
Pannello frontale	E' il modulo di Interfaccia tra macchina e operatore, dotato di uno schermo su cui i messaggi vengono visualizzati e di una tastiera per inserire i dati. E' collegato all'unità base.
Unità base	E' l'unità hardware-software che gestisce tutte le funzioni di controllo, ed è collegata al pannello frontale e alla macchina utente.

FINE PREFAZIONE

INDICE

PROGRAMMARE CON IL SISTEMA SERIE 10

I FILE DI PROGRAMMA	1-1
Componenti del programma	1-2
Blocchi	1-2
Tipi di blocchi	1-4
Funzioni programmabili	1-6
Codici G	1-9
SINCRONIZZAZIONE ED ESECUZIONE DEL PROGRAMMA.....	1-13
Sincronizzazione di default	1-13
Modifica della sincronizzazione di default	1-14
Interprete del Part Program	1-14
Ordine di attuazione.....	1-15
Limiti nella programmazione per numeri longreal (double)	1-15

PROGRAMMAZIONE DEGLI ASSI

MOVIMENTO DEGLI ASSI	2-1
Definizione del movimento degli assi	2-1
G00 - Posizionamento rapido degli assi	2-2
G01 - Interpolazione lineare	2-3
G02 G03 - Interpolazione circolare	2-4
CET (PRC) - Tolleranza di precisione in interpolazione circolare	2-7
FCT - Soglia per cerchio completo	2-8
ARM - Definizione della modalità di normalizzazione dell'arco	2-9
CRT - Soglia di riduzione della velocità in interpolazione circolare	2-13
CRK - Costante di riduzione della velocità in interpolazione circolare	2-13
Interpolazione elicoidale	2-15
G33 - Filettatura a passo costante o variabile	2-17
Assi Rotativi	2-21
Assi Rollover	2-23
G90 - Modo assoluto	2-23
G91 - Modo incrementale	2-25
Pseudo assi	2-26
Assi diametrali	2-26
UDA - Assi duali	2-29
SDA - Assi duali speciali	2-31
XDA - Assi Master/Slave	2-33
Associazione Master/Slave	2-33

Sintassi	2-33
Modo 0	2-34
Modo 2	2-36
Dissociazione Master/Slave	2-39
Sintassi	2-39
Definizione/Variazione rapporto di inseguimento	2-40
Sintassi	2-40
Attivazione inseguimento	2-41
Sintassi	2-41
Disattivazione inseguimento	2-42
Sintassi	2-42
ORIGINI E CONTROLLO DELLE COORDINATE	2-43
G17 G18 G19 - Selezione del piano di interpolazione	2-44
G16 - Definizione del piano di interpolazione	2-45
G27 G28 G29 - Definizione della modalità dinamica	2-46
DECELERAZIONE AUTOMATICA SUGLI SPIGOLI IN G27	2-51
DLA - Decelerazione Look Ahead	2-52
DYM - Modalità dinamica	2-53
MDA - Calcolo del massimo angolo di decelerazione	2-54
VEF - Fattore di velocità	2-55
Jerk Limitation	2-57
MOV - Modalità di Movimentazione	2-58
Significato dei bit 0 ÷ 3 :	2-58
Significato del bit 4:	2-59
Significato dei bit 6 ÷ 7 :	2-60
JRK - Jerk Constant	2-61
JRS - Jerk Smooth Constant	2-62
ODH - Online Debug Help	2-64
MBA – MultiBlock retrace Auxiliary functions	2-66
REM – Ritorno automatico sul profilo a fine movimento	2-67
IPB (DTL) - In Position Band	2-68
G70 G71 - Unità di misura	2-69
G90 G91 G79 - Programmazione assoluta, incrementale o riferita allo zero macchina	2-70
G92 G98 G99 - Presetting dell'asse	2-72
G04 G09 - Attributi della modalità dinamica	2-73
t - Tempo di esecuzione blocco	2-74
DWT (TMR) - Tempo di sosta	2-74
G93 - Velocità di avanzamento V/D	2-75
VFF - Velocity Feed Forward	2-76
MODIFICA DEL SISTEMA DI RIFERIMENTO DELL'ASSE	2-77
SCF - Fattori di scala	2-78
MIR - Uso della lavorazione speculare	2-79
ROT (URT) - Rotazione del piano d'interpolazione	2-82
UAO - Uso delle origini assolute	2-85
UTO (UOT) - Uso di origini temporanee	2-86
UIO - Uso delle origini incrementali	2-88
RQO - Riquifica delle origini	2-90
FINE CORSA ED AREE PROTETTE	2-91
SOL (DLO) - Fine corsa software	2-92
DPA (DSA) - Definizione delle aree protette	2-93
PAE (ASC) - Abilitazione dell'area protetta	2-95
PAD (DSC) - Disabilitazione dell'area protetta	2-95
VIRTUALIZZAZIONI	2-96
Assi Virtuali	2-96
Virtualizzazioni disponibili sul CNC Serie 10	2-96

UPR - Rotazione della terna cartesiana	2-97
Esempi d'uso di UPR.....	2-103
UVP - Programmazione in coordinate polari	2-108
Esempi di programmazione in coordinate polari	2-110
UVC - Programmazione in coordinate cilindriche	2-112
UVA - Programmazione di assi non ortogonali	2-114
Esempio di programmazione in coordinate polari	2-115
TCP - Tool Center Point per macchine con testa "Double Twist"	2-116
Programmazione dei parametri "m" ed "n" (angoli).....	2-132
Programmazione dei parametri "m", "n", "o" e "u", "v", "w" (vettori)	2-133
TCP - Tool Center Point per macchine generiche a 5 assi	2-135
Programmazione	2-140
TCP - Tool Center Point per macchine con utensile fisso e tavola rotante.....	2-144
Programmazione	2-150
TCP su multiprocesso	2-151

PROGRAMMAZIONE DI UTENSILI E CORRETTORI

Indirizzo T per programmazione utensile	3-2
Indirizzo T per programmazione multiutensile.....	3-3
Indirizzo h	3-5
TTR – Thoroidal Tool Radius	3-7
AXO - Definizione del correttore dell'asse.....	3-8
RQT (RQU) - Riquifica dei correttori utensili attuali.....	3-10
RQP - Riquifica dei correttori utensili (valori di presetting).....	3-11
TOU (TOF) - Dichiarazione di utensile scaduto	3-12
LOA - Caricamento delle tabelle in memoria.....	3-13

COMPENSAZIONE DIAMETRO UTENSILE

G40 G41 G42- Compensazione diametro utensile	4-2
Abilitazione della compensazione diametro utensile.....	4-3
Note relative all'uso della compensazione diametro utensile.....	4-5
Ottimizzazione del percorso utensile (Tool Path Optimization).....	4-5
Disabilitazione della compensazione diametro utensile	4-6
Disabilitazione della compensazione con TPO attivo	4-7
CAMBIO COMPENSAZIONE DIAMETRO UTENSILE	4-8
Percorso utensile lineare/lineare	4-8
Percorsi utensile Lineare/Circolare, Circolare/Lineare, Circolare/Circolare.....	4-10
r - Raccordi su profili compensati	4-12
b - Smussi nei profili compensati.....	4-13
TPO - Ottimizzazione del percorso utensile in G41/G42	4-16
Esempi relativi alla modalità TPO=1	4-19
Esempi relativi alla modalità TPO=2	4-22
TPT - Soglia di scostamento dallo spigolo (Tool Path Threshold)	4-25
u v w - Compensazione parassiale	4-27
Esempi d'uso dei fattori di compensazione u, v, w.....	4-28
MSA (UOV) - Definizione di sovrammetallo	4-32
CONTORNITURA AUTOMATICA	4-33
Limiti imposti nell'utilizzo della contornitura automatica	4-33
GTP - Get Point	4-34
Determinazione del punto di attacco	4-35
CCP - Cutter Compensation Profile.....	4-37

PROGRAMMAZIONE DEL MANDRINO

FUNZIONI DEL MANDRINO	5-1
G96 G97 - Programmazione CSS e RPM	5-1
SSL - Limite della velocità di rotazione del mandrino	5-3
M19 - Arresto della rotazione mandrino con orientamento.....	5-4
GTS - Condivisione del mandrino	5-5

FUNZIONI AUSILIARIE M

Le funzioni M standard.....	6-1
-----------------------------	-----

PROGRAMMAZIONE PARAMETRICA

VARIABILI LOCALI	7-4
Parametri E	7-4
! - Variabili utente	7-6
VARIABILI DI SISTEMA.....	7-8
SN - Numero di Sistema	7-8
SC - Carattere di Sistema	7-9
TIM - Temporizzatore di Sistema.....	7-11
@ - Variabili PLUS	7-12
Variabili L	7-13
Assegnazioni multiple	7-14

CICLI FISSI

CICLI FISSI G8N.....	8-1
Caratteristiche del ciclo fisso	8-2
Movimenti di un ciclo fisso	8-3
G81 - Ciclo fisso di foratura	8-5
G82 - Ciclo di lamatura	8-7
G83 - Ciclo di foratura profonda.....	8-9
DRP – Distanza di ripresa foro G83	8-12
G84 - Ciclo di maschiatura senza trasduttore.....	8-13
G84 - Ciclo di maschiatura con trasduttore montato sul mandrino	8-16
G84 - Ciclo di maschiatura rigida con trasduttore montato sul mandrino.....	8-17
TRP (RMS) - Percentuale di ritorno di maschiatura	8-18
G85 - Ciclo di alesatura (o di maschiatura tramite Tapmatic)	8-19
G86 - Ciclo di barenatura	8-20
G89 - Ciclo di barenatura con lamatura	8-21
Uso di due quote R in un ciclo fisso.....	8-22
Aggiornamento delle quote del ciclo fisso	8-23
Aggiornamento delle quote R (inferiore e superiore) durante la lavorazione.....	8-24

PARAMACRO

Definizione di paramacro	9-1
Parametri HC	9-3
DAN - Definizione del nome asse	9-6
PMS – S come paramacro	9-7
PMT – T come paramacro	9-7

PMM – M come paramacro	9-7
Sintassi	9-7
FUNZIONI AUSILIARIE S, T, M ESEGUITE COME PARAMACRO.....	9-8
Sintassi:	9-8

CICLI DI TASTATURA

GESTIONE DEL TASTATORE ELETTRONICO	10-1
PREDISPOSIZIONE DI UN CICLO DI TASTATURA	10-4
DPP (DPT) - Definizione dei parametri di tastatura.....	10-4
Misurazione dinamica del diametro della sfera	10-5
Riquilifica del tastatore	10-5
Misurazione dinamica della lunghezza del tastatore.....	10-5
Predisposizione del tastatore	10-5
CICLI DI TASTATURA	10-7
G72 - Misura di un punto con compensazione.....	10-8
G73 - Misura dei parametri di un foro.....	10-10
G74 - Ciclo di riquilifica dell'utensile.....	10-12
UPA (RTA) - Aggiornamento dell'ascissa del tastatore.....	10-14
UPO (RTO) - Aggiornamento dell'ordinata del tastatore.....	10-14
Gestione degli errori di tastatura	10-14
OPERAZIONI CON UN TASTATORE MOBILE	10-15
Riquilifica delle origini tramite tastatura di superfici di riferimento	10-15
Riquilifica delle origini centrando su un foro.....	10-17
Verifica dei diametri dei fori	10-17
Verifica delle quote dei piani e delle profondità dei fori	10-19
OPERAZIONI CHE UTILIZZANO UN TASTATORE FISSO	10-20

GESTIONE DEL VIDEO

GESTIONE DEL VIDEO GRAFICO	11-1
UGS (UCG) - Uso della scala grafica	11-2
UGS (UCG) - Uso della scala grafica in 3D	11-3
CGS (CLG) - Cancellazione dello schermo grafico.....	11-3
DGS (DCG) - Disabilitazione della scala grafica	11-4
DIS - Visualizzazione di una variabile	11-4

MODIFICA DELLA SEQUENZA DI ESECUZIONE DEL PROGRAMMA

GENERALITÀ.....	12-1
COMANDI PER LA RIPETIZIONE DI PARTI DI PROGRAMMA	12-4
RPT-ERP	12-4
COMANDI PER L'ESECUZIONE DI SOTTOPROGRAMMI.....	12-8
CLS – CLD – CLT - Chiamata di subroutine	12-8
PTH - Dichiarazione del <i>pathname</i> di default.....	12-14
EPP - Esecuzione di una porzione di programma	12-15
EPB - Esecuzione di un blocco di part program.....	12-17
COMANDI DI SALTO E DI ATTESA	12-19
GTO - Comando di salto.....	12-19
IF ELSE ENDIF	12-23
DLY - Definizione del tempo di attesa	12-24

DSB - Disabilitazione di blocchi barrati	12-25
REL - Deseleziona il part program	12-25
WOS - WAIT su segnale	12-26
COMANDI PER LA DEFINIZIONE DI DEVICE	12-27
GDV - Definizione di <i>device</i> per accesso a file	12-27
RDV - Rilascio <i>device</i>	12-28

HIGH SPEED MACHINING

CONSIDERAZIONI GENERALI	13-1
PROGRAMMAZIONE DEI PUNTI E CARATTERISTICHE DEL PROFILO	13-3
Considerazioni sull'uso delle G62, G63, G66 e G67 (codici di transizione)	13-6
STRUTTURA GENERALE DELLA PROGRAMMAZIONE HIGH SPEED MACHINING	13-7
Interazione con Logica di Macchina	13-7
CONVENZIONI PER LA DEFINIZIONE DEI PUNTI	13-8
Punti e coordinate di lavorazione	13-8
Direzione Utensile: IJK	13-9
Direzione Normale alla Superficie: MNO	13-9
Direzione Applicazione Raggio Utensile: PQD	13-10
Fattori di Compensazione Parassiale: UVW	13-11
Asse Tangente	13-12
PRESTAZIONI FORNITE CON HIGH SPEED MACHINING	13-13
Compensazione Raggio e Lunghezza utensile	13-13
Compensazione Lunghezza utensile	13-14
Nessuna compensazione utensile	13-15
Gestione asse tangente	13-15
CONFIGURAZIONE	13-16
Modalità di gestione dei versori	13-18
Gestione Look Ahead	13-20
Soglie	13-22
Definizione Utensile	13-24
Direzione Utensile (3D)	13-25
Gestione Cambio di Curvatura	13-26
Gestione Spigoli	13-27
Definizione assi	13-28
Parametri Assi	13-30
Dinamica Assi	13-31
Esempio	13-32

COMANDI RELATIVI ALLA GESTIONE MULTIPROCESSO

GENERALITA'	14-1
NOTE FUNZIONALI SULLA SINCRONIZZAZIONE TRA PROCESSI	14-2
Note sulla funzionalità "WAIT"	14-2
Note sulla funzionalità "SEND"	14-2
Scambio dati	14-3
Reset processi sincronizzati	14-3
Tabella canali	14-3
DCC - Definizione del canale di comunicazione	14-4
PVS - Selezione del canale PLUS	14-5
PRO - Definizione del processo di default	14-6
SND - invio di un messaggio di sincronizzazione (send)	14-7
WAI - Attesa di un messaggio di sincronizzazione (wait)	14-9
EXE - Attivazione automatica di un part program	14-11

ECM - Esecuzione di un blocco in modo MDI nel processo specificato	14-12
Esempio di sincronizzazione su 2 processi e di utilizzo del comando EXE:.....	14-13
NOTE SULLA FUNZIONALITA' "ASSI MIGRANTI".....	14-14
Generalità	14-14
Condizioni per l'acquisizione di un asse.....	14-14
GTA - Acquisizione assi (Get Axes)	14-15
INFORMAZIONI INIZIALIZZATE DOPO UN COMANDO GTA	14-16
INFORMAZIONI MANTENUTE DOPO UN COMANDO GTA.....	14-17
Gestione delle condizioni di errore	14-26

PROGRAMMAZIONE GEOMETRICA AD ALTO LIVELLO (GTL)

GEOMETRIA ORIENTATA	15-2
MEMORIZZAZIONE DI ENTI GEOMETRICI	15-5
DEFINIZIONE DI ORIGINE DI RIFERIMENTO	15-8
DEFINIZIONE DI PUNTI.....	15-9
DEFINIZIONE DI RETTA.....	15-15
DEFINIZIONE DI CERCHI.....	15-26
DEFINIZIONE DI PROFILO.....	15-40
Tipi di profilo	15-40
Collegamento degli enti	15-45
ESEMPI DI PROGRAMMAZIONE GTL	15-49

CICLI DI LAVORAZIONE PER SISTEMI DI TORNITURA

PROGRAMMAZIONE DEL PROFILO	16-1
Limiti imposti nella definizione di un profilo da richiamare nelle macro-istruzioni di sgrossatura/finitura.	16-2
PROGRAMMAZIONE DI CICLI SPECIALI.....	16-3
MACROISTRUZIONI DI SGROSSATURA PARASSIALE SENZA PREFINITURA	16-3
MACROISTRUZIONI DI SGROSSATURA PARASSIALI CON PREFINITURA.....	16-7
MACROISTRUZIONE DI SGROSSATURA PARALLELA AL PROFILO	16-9
MACROISTRUZIONE DI FINITURA DI UN PROFILO	16-11
CICLO DI FILETTATURA	16-12
CICLO DI TAGLIO GOLE	16-16

SOMMARIO DEI CARATTERI E DEI COMANDI

CARATTERI CONSENTITI	A-1
SOMMARIO DEI CODICI G	A-5
FUNZIONI MATEMATICHE	A-6
VARIABILI LOCALI E DI SISTEMA	A-6
TRILETTERALI	A-7
CODICI ASCII.....	A-10

MESSAGGI D'ERRORE

Descrizione dei messaggi di errore ed azione di recupero	B-1
--	-----

GESTIONE ERRORI DA PART PROGRAM

GENERALITÀ	C-1
ERR - Abilita/Disabilita la gestione errori nel Part Program	C-2
Errori durante il ciclo di tastatura	C-3
Errori nella gestione "Assi Migranti"	C-4

FINE INDICE

PROGRAMMARE CON IL SISTEMA SERIE 10

I part program vengono scritti con un linguaggio specifico definito dallo standard ISO. Questo capitolo descrive gli elementi del linguaggio ed introduce i relativi argomenti di programmazione.

I FILE DI PROGRAMMA

I part program, sul sistema Serie 10, sono memorizzati su file i quali possono essere identificati tramite nomi della SERIE 10, oppure tramite nomi DOS.

- I nomi di tipo SERIE 10 possono avere una lunghezza massima di 48 caratteri; essi identificano i programmi memorizzati nei direttori logici configurati sulla macchina.
La configurazione dei direttori logici avviene durante la fase di installazione (PPDIR config - menù human interface nell'ambito della caratterizzazione AMP).
- I nomi di tipo DOS, sono costituiti da un massimo di 8 caratteri più eventuali estensione e path; i nomi DOS identificano file che risiedono in direttori di tipo DOS.

Non è consentita la gestione "mista" dei part program; se infatti un programma viene attivato richiamandolo tramite un nome di tipo DOS, anche tutte le sue eventuali subroutine dovranno essere identificate con nomi DOS.

Analogamente, i programmi identificati con nomi tipo Serie 10 potranno utilizzare solo subroutine identificate allo stesso modo.

NOTA:

I part program possono anche risiedere su device remoti, precedentemente definiti tramite il triletterale GDV (Vedere cap. 12).

Componenti del programma

Indirizzo

Un indirizzo è una lettera che identifica il tipo di istruzione. Per esempio, i seguenti sono indirizzi:

G, X, Y, F

Word

Una word (parola) è un indirizzo seguito da un valore numerico. Per esempio, le seguenti sono word:

G1 X50.5 Z-3.15 F200 T1.1

Il valore numerico assegnato ad una word deve essere espresso nel sistema di misurazione della word senza gli zeri di testa e di coda. Se il valore numerico ha una frazione decimale, inserire la porzione decimale dopo il punto decimale.

Blocco

Un blocco di programma è composto da una serie di word, le quali identificano un'operazione o una serie di operazioni da eseguire. La lunghezza massima di un blocco è di 128 caratteri.

Il programma tecnologico è una sequenza di blocchi che descrivono un'operazione di lavorazione.

Ciascun blocco deve terminare con <CR><LF>.

Blocchi

Il blocco è la componente fondamentale di un part program e consiste di uno o più campi. Quando vengono usati più campi nello stesso blocco, essi devono apparire nell'ordine mostrato nella seguente tabella:

cancellazione di blocco	etichetta	numero di sequenza	sincronizzazione asincronizzazione	codici word
/	ETICHETTA	NUMERO	# o &	TUTTI I CARATTERI CONSENTITI

Blocco di commento

Può essere inserito in una qualsiasi posizione all'interno del blocco attuale. Qualunque carattere dopo ";" viene interpretato come un commento.

Cancellazione di blocco

Il campo cancellazione di blocco (block delete) è opzionale. Permette all'operatore di scegliere se eseguire blocchi di programma che cominciano con il carattere "/", chiamati blocchi barrati.

Esempio:

/N100 G00 X100

Il blocco dell'esempio può essere abilitato o disabilitato usando la softkey PROGRAM SET UP, o digitando da tastiera il codice triletterale DSB.

Etichetta

Il campo etichetta è opzionale. Consente al programmatore di assegnare un nome simbolico ad un blocco. Un'etichetta può essere costituita da un massimo di sei caratteri alfa numerici. I caratteri devono apparire tra doppi apici. Se il blocco è barrato, l'etichetta deve essere inserita dopo la barra.

Esempio:

"START"

/"END"

Quando un campo etichetta viene usato in un comando 'GTO' l'etichetta definisce il blocco al quale il controllo deve effettuare un salto.

Numero di sequenza

Il campo 'numero di sequenza' è opzionale. Permette al programmatore di numerare i blocchi di programma. Il numero di sequenza è composto dalla lettera N seguita da un massimo di sei cifre (N0-N999999).

Il numero di sequenza deve apparire davanti al primo operando e dopo l'etichetta.

Esempio:

N125 X0

"START" N125 X0

"END" N125 X0

Sincronizzazione/asincronizzazione

I caratteri & e # vengono utilizzati per modificare lo stato di sincronizzazione/asincronizzazione di default. Vedere in seguito la sezione "Sincronizzazione ed Esecuzione del Programma" per ulteriori informazioni relative alla sincronizzazione.

Esempio:

#{GTO,START, @PL1=1)

Tipi di blocchi

In un part program si possono usare quattro tipi di blocchi:

- Blocchi di commento
- Blocchi di movimento
- Blocchi di assegnazione
- Blocchi con codici trilettari

Blocchi di commento

Un blocco di commento permette al programmatore di inserire delle frasi libere nel programma. Queste frasi possono descrivere la funzione da eseguire o fornire elementi di informazione che rendono il programma più comprensibile e documentato.

Un blocco di commento non produce messaggi per l'operatore. Durante l'esecuzione del programma il controllo ignora i blocchi di commento.

Il primo carattere di un blocco di commento deve essere un punto e virgola (;). Il resto del blocco di commento è una sequenza di caratteri alfanumerici. Per esempio:

`;QUESTO E' UN ESEMPIO DI BLOCCO DI COMMENTO`

Oltre che in un blocco a sé stante, si può inserire un commento anche negli altri tipi di blocchi, a partire dal carattere ";".

Tutti i caratteri che seguiranno sono considerati come commento. Per esempio:

```
G1 X100 Y50 ;Blocco di movimento  
E1=10      ;Variabile locale E  
(ROT,45)   ;Comando di rotazione
```

Blocchi di movimento

I blocchi di movimento sono quelli conformi agli standard ISO e ASCII per la programmazione di blocchi. Non c'è un ordine particolare per programmare i componenti di un blocco di movimento.

Esempio:

`G1 X500 Y20 F200`

Blocchi di assegnazione

I blocchi di assegnazione vengono usati per scrivere i valori delle variabili direttamente da programma. La seguente tabella mostra i numerosi tipi di assegnazione possibili.

Tipo di assegnazione	Esempio
Assegnazione semplice	E10=123.567
Assegnazione multipla	E1=10, 15.5, 123.467 L'assegnazione carica i valori: 10 in E1 15.5 in E2 123.467 in E3
Assegnazione di espressione matematica	E20=(E10+125*SQR(E23))
Numero di sistema	SN=1.5

Blocchi con codici triletterali

I blocchi con codici triletterali definiscono un'operazione con un'istruzione triletterale conforme allo standard RS-447. Per esempio:

(ROT,45)
(DIS,"message text")

Per motivi di compatibilità con i CNC della serie 8600, alcuni comandi estesi possono essere programmati con due triletterali equivalenti:

UGS	UCG
CGS	CLG
DGS	DCG
RQT	RQU
DPA	DSA
PAE	ASC
PAD	DSC
DPP	DPT
IPB	DTL
ROT	URT
SOL	DLO
UTO	UOT
TOU	TOF

Funzioni programmabili

Coordinate degli assi

Le coordinate degli assi possono essere chiamate con le lettere ABCUVWXYZPQD (secondo la configurazione definita in AMP) e possono essere programmate nei seguenti intervalli di valori:

-99999.99999	-0.00001	mm/pollici
+0.00001	+99999.99999	mm/pollici

NOTA:

Non è possibile programmare valori nell'intervallo ± 0.00001 perché 0.00001 è il valore minimo accettato dal controllo.

Coordinata R

In una interpolazione circolare (G02- G03) la coordinata R rappresenta il raggio del cerchio.
In un ciclo fisso standard (G81-G89), la coordinata R definisce la quota di inizio lavoro o la quota di risalita. Questa funzione è programmabile nei seguenti intervalli:

-99999.99999	-0.00001	mm/pollici
+0.00001	+99999.99999	mm/pollici

NOTA:

Non è possibile programmare i valori nell'intervallo ± 0.00001 , perché 0.00001 è il valore minimo accettato dal controllo.

In un blocco per filettatura (G33), la coordinata R rappresenta lo sfasamento rispetto alla posizione angolare zero del mandrino (per filetti a più principi).

Coordinate I J

Nell'interpolazione circolare (G02-G03), I e J sono le coordinate del centro dell'arco. I specifica l'ascissa (normalmente X) e J l'ordinata (normalmente Y) del centro. I e J specificano sempre le coordinate del centro indipendentemente dal piano di interpolazione attivo.
Questa funzione è programmabile nei seguenti intervalli:

-99999.99999	-0.00001	mm/pollici
+0.00001	+99999.99999	mm/pollici

NOTA:

Non è possibile programmare i valori nell'intervallo ± 0.00001 , perché 0.00001 è il valore minimo accettato dal controllo.

Quando i valori degli assi corrispondenti (secondo la configurazione definita in AMP) sono in unità diametrali anche i valori delle coordinate del centro per I e J sono espressi in unità diametrali.

Le coordinate I e J sono anche usate nel ciclo foratura profonda (G83).

In un blocco per filettatura (G33), l'indirizzo I definisce la variazione del passo per filettature a passo variabile:

I+ passo crescente
I- passo decrescente

Funzione K

Nel ciclo di foratura profonda (G83), K definisce il valore incrementale da applicare per ridurre la profondità iniziale del passo (I) al valore di profondità minima (J).

Questa funzione è programmabile nei seguenti intervalli:

-99999.99999	-0.00001	mm/pollici
+0.00001	+99999.99999	mm/pollici

NOTA:

Non è possibile programmare i valori nell'intervallo ± 0.00001 , perché 0.00001 è il valore più piccolo accettato dal controllo.

In un blocco di filettatura (G33) o in un ciclo di maschiatura (G84), K definisce il passo del filetto. Nell'interpolazione elicoidale (G02-G03), K definisce il passo dell'elica.

Funzione F e t

La funzione F definisce la velocità di avanzamento degli assi. Questa funzione è programmabile nel seguente intervallo:

+0.00001	+99999.99999	mm/pollici
----------	--------------	------------

In G94 la funzione F definisce la velocità di avanzamento in millimetri al minuto (G71) o in pollici al minuto (G70).

Un valore "t" può essere programmato in un blocco per specificare il tempo in secondi che occorre per completare lo spostamento definito nel blocco. In questo caso la velocità di avanzamento sarà:

$$F = \frac{\text{distanza totale}}{\text{tempo}} * 60$$

Un valore "t" è valido solo all'interno del blocco in cui è programmato.

In G93 la funzione F definisce l'inverso del tempo necessario, espresso in minuti, per completare lo spostamento:

$$F = \frac{\text{velocità}}{\text{distanza totale}} = 1/t \text{ (minuti)}$$

La funzione F è obbligatoria nei blocchi quando G93 è attivo ed è valida solo per quel blocco.

In G95, F specifica la velocità di avanzamento degli assi espressa in millimetri al giro (G71) o in pollici al giro (G70) del mandrino.

Capitolo 1

Programmare con il Sistema Serie 10

Funzione a

La funzione a, definisce l'accelerazione da usare sul blocco di part program, ed è programmabile nell'intervallo:

+0.00001	+99999.99999	mm/Sec ² oppure pollici/Sec ²
----------	--------------	---

La funzione a, è considerata in mm/Sec² in presenza di G71 ed in pollici/Sec² in presenza di G70. Tale funzione è attiva solo nel blocco in cui è programmata ed è comunque limitata dalla accelerazione sul profilo calcolata dal sistema in funzione delle accelerazioni configurate.

Funzione M

La funzione M può attivare diverse operazioni di macchina. L'intervallo programmabile va da 0 a 999. Consultare il Capitolo 6 per ulteriori informazioni.

Funzione S

La funzione S specifica la velocità di rotazione del mandrino. E' programmabile nel seguente intervallo:

+0.001	999999.999	gm/gm
--------	------------	-------

In G97, la funzione S definisce la velocità di rotazione del mandrino espressa in giri/min.

In G96, la funzione S definisce la velocità di taglio espressa in metri al minuto (G71) o in piedi al minuto (G70). Tale velocità di taglio rimane costante sulla superficie.

Fare riferimento al Capitolo 5 per ulteriori informazioni sulla programmazione della funzione S.

Funzione T

La funzione T definisce l'utensile e il correttore necessari per la lavorazione. E' programmabile in un intervallo che va da 0.0 a 99999999999.300. Le 12 cifre a sinistra del punto decimale rappresentano il codice dell'utensile e le tre cifre a destra rappresentano il numero del correttore. Il Capitolo 3 fornisce una dettagliata descrizione delle funzioni T.

IMPORTANTE

Le funzioni M, S, T, hanno funzionalità diverse a seconda di come sono state caratterizzate in AMP.

Con la release SW 3.1 e successive, è possibile l'emissione di queste funzioni anche durante il movimento continuo (G27-G28).

Per questi tipi di applicazioni il costruttore deve:

- configurare in AMP la funzione desiderata come "ALLOWED IN CONTINUOUS".
- Scrivere una logica di macchina che la gestisca.

Il programmatore di part program, da parte sua deve tenere presente che queste funzioni si comportano in modo diverso a seconda della modalità in cui ci si trova:

- In **continuo**: la funzione dichiarata "ALLOWED IN CONTINUOUS" verrà emessa nello stesso ordine in cui è stata programmata. L'emissione sarà di tipo "NO WAIT" per non bloccare l'esecuzione del continuo.
- In **punto-punto**: la funzione dichiarata "ALLOWED IN CONTINUOUS" verrà emessa nella modalità standard.

Funzione h

La funzione h permette il cambio del correttore sia durante il movimento continuo sia durante il movimento punto a punto.

Deve essere l'unica funzione programmata nel blocco, il suo valore può variare tra 0 e 300.

Tale valore può essere espresso come numero intero o tramite variabile E.

Funzioni G

I codici G programmano funzioni preparatorie per la lavorazione. Essi vengono trattati nella sezione seguente.

Codici G

Questa sezione spiega come utilizzare i codici G di preparazione nei blocchi del part program. Un codice G di preparazione è identificato dall'indirizzo G seguito da una o due cifre (G00-G99). Attualmente sono disponibili solo alcuni dei 100 codici G possibili.

I sottoprogrammi paramacro possono essere richiamati con un codice G di tre cifre. Questa classe di codici G viene descritta nel Capitolo 9. I codici G a tre cifre sono classificati come segue:

G100 - G299	Riservati
G300 – G699	Paramacro non modali
G700 - G998	Paramacro modali
G999	Reset di paramacro modali

Capitolo 1

Programmare con il Sistema Serie 10

In un blocco, il codice G deve essere programmato dopo il numero di sequenza (se definito) e prima di qualunque altro operando. Per esempio:

N100 G01 X0 - operando

E' possibile programmare più codici G nello stesso blocco, ammesso che siano tra loro compatibili. La tabella seguente definisce la compatibilità tra codici G. Lo zero indica che i codici G sono compatibili e possono essere programmati nello stesso blocco; uno significa che i codici G non sono compatibili e non possono essere programmati nello stesso blocco senza determinare errori.

Codici G compatibili

G	00	01	02	33	81	80	72	93	96	41	40	27	29	04	09	90	79	70	16	92
			03		89		73	94	97	42		28				91		71	17	99
							74	95											18	19
G00	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
G01	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
G02	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
G03	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
G04	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	1	1
G09	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	1	1
G16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G27	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	1	1
G28	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	1	1
G29	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	1
G33	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1	1
G40	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	1	1
G41	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	1	1
G42	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	1	1
G70	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1
G71	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1
G72	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G73	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G74	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G79	0	0	0	0	1	1	1	0	0	1	1	0	0	0	0	1	1	0	1	1
G80	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	1	1
G81	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	1	1
G82	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1
G83	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1
G84	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1
G85	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1
G86	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1
G89	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1
G90	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1
G91	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1
G92	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G93	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1
G94	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1
G95	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1
G96	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1
G97	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1
G99	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

NOTA:

0 indica codici compatibili

1 indica codici non compatibili

La tabella che segue fornisce un riassunto dei codici G disponibili nel controllo. Questa configurazione di default può essere modificata mediante il configuratore AMP.

Riassunto codici G

CODICE	GRUPPO	MODALE	DESCRIZIONE	ALL'ACCENSIONE	
				FRESA	RETT.
G00	a	sì	Posizionamento dell'asse in rapido	sì	sì
G01	a	sì	Interpolazione lineare	no	no
G02	a	sì	Interpolazione circolare oraria	no	no
G03	a	sì	Interpolazione circolare antioraria	no	no
G33	a	sì	Filettatura a passo costante o variabile	no	no
G16	b	sì	Interpol. circolare e correzione profilo su un piano definito	no	no
G17	b	sì	Interpol. circolare e correzione profilo sul piano XY (1°-2° asse)	sì	no
G18	b	sì	Interpol. circolare e correzione profilo sul piano ZX (3°-1° asse)	no	sì
G19	b	sì	Interpol. circolare e correzione profilo sul piano YZ (2°-3° asse)	no	no
G27	c	sì	Funzionamento continuo con riduzione automatica della velocità sugli spigoli	sì	sì
G28	c	sì	Funzionamento continuo senza riduzione della velocità sugli spigoli	no	no
G29	c	sì	Funzionamento punto a punto	no	no
G92	d	no	Presetting dell'asse	no	no
G99	d	sì	Cancellazione di G92	sì	sì
G40	e	sì	Disabilita correzione su profilo	sì	sì
G41	e	sì	Correzione su profilo (utensile a sinistra)	no	no
G42	e	sì	Correzione su profilo (utensile a destra)	no	no
G20		sì	Chiude profilo GTL		
G21		sì	Apri profilo GTL		
G60		sì	Chiude il profilo HSM	no	no
G61		sì	Apri il profilo HSM	no	no
G62		no	Spezza il profilo HSM in due con continuità	no	no
G63		no	Spezza il profilo HSM in due con raccordo	no	no
G66		no	Spezza il profilo HSM in due con spigolo	no	no
G67		no	Spezza il profilo HSM in due con velocità contratta sullo spigolo	no	no

Capitolo 1

Programmare con il Sistema Serie 10

CODICE	GRUPPO	MODALE	DESCRIZIONE	ALL'ACCENSIONE	
				FRESA	RETT.
G70	f	sì	Programmazione in pollici	no	no
G71	f	sì	Programmazione in mm	sì	sì
G80	g	sì	Chiusura cicli fissi	sì	sì
G81	g	sì	Ciclo fisso foratura	no	no
G82	g	sì	Ciclo fisso lamatura	no	no
G83	g	sì	Ciclo fori profondi (con scarico trucioli)	no	no
G84	g	sì	Ciclo fisso maschiatura	no	no
G85	g	sì	Ciclo fisso alesatura	no	no
G86	g	sì	Ciclo fisso barenatura	no	no
G89	g	sì	Ciclo fisso barenatura con sosta	no	no
G90	h	sì	Programmazione assoluta	sì	sì
G91	h	sì	Programmazione incrementale	no	no
G79	k	no	Programmazione riferita allo zero macchina	no	no
G04	i	no	Sosta a fine blocco	no	no
G09	i	no	Decelerazione a fine blocco	no	no
G72	j	no	Misura di un punto con compensazione raggio	no	no
G73	j	no	Misura dei parametri di un foro	no	no
G74	j	no	Misura dello scostamento teorico da un punto senza compensazione raggio	no	no
G93	l	sì	Velocità di avanzamento espressa come inverso del tempo di esecuzione dell'ente	no	no
G94	l	sì	Velocità di avanzamento in mm/min. o pollici/min.	sì	no
G95	l	sì	Velocità di avanzamento in mm/giro o pollici/giro	no	sì
G96	m	sì	Velocità di taglio in m/min. o piedi/min.	no	sì
G97	m	sì	Velocità di rotazione mandrino in giri/min.	sì	no

SINCRONIZZAZIONE ED ESECUZIONE DEL PROGRAMMA

I termini "sincronizzato" e "non sincronizzato" si applicano solo ai blocchi di part program che non implicano uno spostamento, cioè ai blocchi di assegnazione o di calcolo. Un blocco di movimento è qualunque blocco che contenga, insieme ad altre azioni, un movimento degli assi. In altre parole:

- Spostamento dell'asse
- Codici M
- Codici S
- Codici T

Un blocco sincronizzato viene preso in considerazione, e quindi eseguito, solo dopo che il blocco di movimento che lo precede nel programma è stato completato, cioè quando il movimento effettivo dell'asse è stato eseguito.

Al contrario, un blocco non sincronizzato viene eseguito non appena è stato letto dall'interprete del part program, quindi quando magari un movimento precedente è ancora in corso.

Il vantaggio dell'esecuzione del blocco asincrono consiste nel fatto che si possono fare assegnazioni e calcoli complessi tra un movimento e l'altro. Questo permette di eliminare le attese tra due blocchi di movimento dovute al tempo di esecuzione dei calcoli.

Sincronizzazione di default

Ad ogni accensione del sistema, vengono automaticamente sincronizzati i seguenti comandi e codici:

- UDA, SCF, RQO, IPB, DLY, WOS, WAI, SND, GTA, REL, UPR, TCP, UVP, UVC
- G16, G17, G18, G19, G72, G73, G74

Tutti gli altri comandi non sono sincronizzati.

Questa assegnazione di default può essere cambiata. Ciò significa che i comandi che sono sincronizzati per default all'accensione possono diventare asincroni e i comandi che non sono sincronizzati per default all'accensione possono diventare sincroni. La sezione successiva spiega come modificare la sincronizzazione di default.

NOTA:

Non può essere modificata la sincronizzazione di default per i triletterali GTA, UPR, TCP, UVP, UVC.

Modifica della sincronizzazione di default

In certe circostanze, il part program può richiedere la modifica della sincronizzazione di default.

Se il comando è sincronizzato per default ed il programmatore desidera che il comando sia eseguito dall'interprete non appena viene letto (operazione asincrona), deve essere programmato un carattere "&" nella prima posizione del blocco, cioè immediatamente dopo il numero "n".

Se il comando è asincrono e si vuole attivare un'operazione sincrona, il primo carattere nel blocco deve essere un "#".

Sia # che & sono attivi solo nel blocco in cui sono programmati.



Per evitare possibili danni al pezzo in lavorazione, si tenga presente che programmare dei blocchi sincronizzati tra i blocchi di contornitura causa lo svuotamento del buffer dei movimenti (o coda enti) ad ogni esecuzione di un blocco sincronizzato. Questo determina un arresto per il tempo necessario a ricaricare il buffer ed eseguire tutti i calcoli.

Interprete del Part Program

La lettura di un blocco di part program genera l'esecuzione di attività diverse in funzione del tipo di blocco:

- Un blocco di movimento sarà caricato in coda al buffer di movimento. Se lo spostamento è definito da una variabile, i valori di spostamento memorizzati sono quelli della variabile. La dimensione del buffer è configurabile, tramite AMP, da 2 a 128 blocchi.
- Un blocco asincrono di assegnazione o di calcolo verrà eseguito.

Tre tipi di eventi interrompono la lettura dei blocchi da parte dell'interprete del part program:

- Il buffer di movimento è pieno. Quando il blocco di movimento attivo sarà eseguito, l'interprete potrà leggere un altro blocco di movimento e caricarlo in coda al buffer.
- Viene letto un blocco non di movimento che contiene un comando sincronizzato o un codice che forza la sincronizzazione. L'interprete non riparte finché l'ultimo blocco di movimento caricato non è eseguito. A questo punto viene eseguito il blocco che programma la sincronizzazione e l'interprete inizia la lettura dei blocchi successivi.
- Le condizioni di errore.

Ordine di attuazione

1. Asse diametrale
2. Fattore di scala (SCF)
3. Unità di misura (G70 G71)
4. Compensazioni vettoriali (u v w)
5. Modalità di programmazione (G90 G91)
6. Lavorazione in Mirror (MIR)
7. Rotazione (ROT)
8. Origini (UAO UTO UIO G92)

Limiti nella programmazione per numeri longreal (double)

- Massimo 15 cifre in totale
- Massimo 12 cifre intere
- Massimo 9 cifre decimali

Il sistema visualizzerà un errore se vengono programmate più di 12 cifre intere.

Se vengono programmate più cifre decimali di quelle consentite, il sistema non visualizza nessun errore ma tronca il numero programmato all'ultima cifra ammessa.

Capitolo 1

Programmare con il Sistema Serie 10

FINE CAPITOLO

PROGRAMMAZIONE DEGLI ASSI

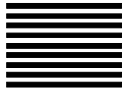
MOVIMENTO DEGLI ASSI

Definizione del movimento degli assi

In questo manuale le direzioni di movimento dell'asse vengono definite secondo lo standard EIA RS267. Per convenzione, si assume sempre che lo spostamento dell'asse è quello dell'utensile verso il pezzo in lavorazione. Non importa se l'utensile si sposta verso il pezzo in lavorazione o se è il pezzo in lavorazione a spostarsi verso l'utensile.

I principali tipi di movimento sono definiti dai codici G elencati nella seguente tabella:

CODICE G	FUNZIONE
G00	Posizionamento degli assi in rapido
G01	Interpolazione lineare
G02	Interpolazione circolare oraria
G03	Interpolazione circolare antioraria
G33	Filettatura a passo costante o variabile



G00 - Posizionamento rapido degli assi

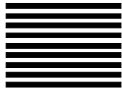
G00 definisce uno spostamento lineare alla velocità di rapido, simultaneo e coordinato su tutti gli assi programmati nel blocco.

Sintassi

G00 [*codici-G*] [*assi*] [*offset*] [*F..*] [*a*] [*ausiliarie*]

dove:

- codici-G** Altri codici G compatibili con G00 (Vedi tabella "Codici G compatibili" nel capitolo 1).
- assi** Lettera corrispondente ad un asse seguita da un valore numerico. Il valore numerico può essere programmato direttamente con un valore decimale o indirettamente con un parametro E. Possono essere definiti fino a nove assi.
- offset** Fattori di correzione sul profilo. Questi fattori vengono introdotti per gli assi X, Y, Z rispettivamente con i caratteri u, v, w. Per ulteriori informazioni consultare la sezione "Compensazione parassiale" nel capitolo 4.
- F** Velocità di avanzamento (o di lavoro) per movimenti coordinati. Viene data con l'indirizzo F seguito dal valore. Questo parametro non incide sullo spostamento degli assi programmato nel blocco G00, ma definisce la velocità di avanzamento per spostamenti successivi e come tale viene memorizzata. La velocità di avanzamento rapido forzata da G00 è la composizione vettoriale delle velocità di rapido degli assi programmati nel blocco. La velocità di avanzamento rapido massima per ogni asse viene definita durante la caratterizzazione tramite AMP.
- a** Accelerazione da utilizzare sul profilo.
- ausiliarie** Funzioni ausiliarie programmabili M, S, T. Nel blocco possono essere programmate fino a quattro funzioni M, una S (velocità di rotazione del mandrino) e una T (selezione dell'utensile).



G01 - Interpolazione lineare

G01 definisce un movimento lineare alla velocità di lavoro programmata, simultaneo e coordinato su tutti gli assi programmati nel blocco.

Sintassi

G01 [*codici-G*] [*assi*] [*offset*] [*F..*] [*a*] [*ausiliarie*]

dove:

codici-G Altri codici G compatibili con G01 (Vedi tabella "Codici G compatibili" nel cap. 1).

assi Lettera corrispondente ad un asse seguita da un valore numerico. Il valore numerico può essere programmato direttamente con un valore decimale o indirettamente con un parametro E. Possono essere definiti fino a nove assi.

offset Fattori di correzione sul profilo. Questi fattori vengono introdotti per gli assi X, Y, Z rispettivamente con i caratteri u, v, w. Per ulteriori informazioni consultare la sezione "Compensazione parassiale" nel capitolo 4.

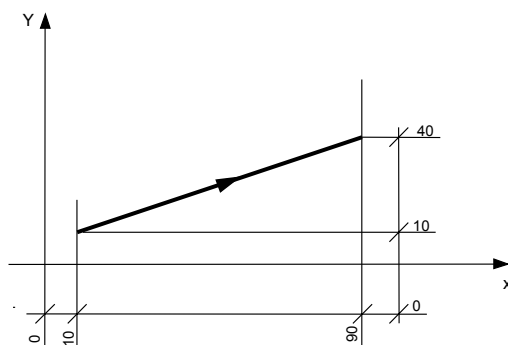
F Velocità di avanzamento (o di lavoro) con cui viene eseguito il movimento. Viene programmata con l'indirizzo F seguito dal valore. Se assente, viene assunta come velocità di lavoro quella programmata in precedenza. Se non è mai stata programmata alcuna velocità il controllo genererà un errore. La velocità si esprime in mm/min o in pollici/min.

a Accelerazione da utilizzare sul profilo.

ausiliarie Funzioni ausiliarie programmabili M, S, T. Nel blocco possono essere programmate fino a quattro funzioni M, una S (velocità di rotazione del mandrino) e una T (selezione dell'utensile).

Esempio:

Questo esempio mostra come programmare un codice G01.



Programma:

N60 (UGS,X,-10,100,Y,-10,50)

N70 G0 X10 Y10

N80 G01 X90 Y40 F200



G02 G03 - Interpolazione circolare

I movimenti circolari vengono definiti dai seguenti codici:

G02 Interpolazione circolare oraria

G03 Interpolazione circolare antioraria

Il movimento circolare viene eseguito alla velocità di lavoro programmata ed è coordinato e simultaneo su tutti gli assi programmati nel blocco.

Sintassi

G02 [*codici-G*] [*assi*] **I.. J..** [**F..**] [**a**] [*ausiliarie*]

oppure

G02 [*codici-G*] [*assi*] **R..** [**F..**] [**a**] [*ausiliarie*]

G03 [*codici-G*] [*assi*] **I.. J..** [**F..**] [**a**] [*ausiliarie*]

oppure

G03 [*codici-G*] [*assi*] **R..** [**F..**] [**a**] [*ausiliarie*]

dove:

codici-G Altri codici G compatibili con G02 e G03 (Vedi tabella "Codici G compatibili" nel capitolo 1).

assi Lettera corrispondente ad un asse seguita da un valore numerico programmato direttamente con un valore decimale o indirettamente con un parametro E.

Se non vi sono assi programmati nel blocco, il movimento eseguito è un cerchio completo nel piano di interpolazione attivo.

I Ascissa del centro del cerchio. Questo valore, espresso in millimetri o pollici, può essere programmato direttamente o indirettamente con un parametro E. Quando l'asse corrispondente è un asse diametrale, l'ascissa viene espressa in modo diametrale. Il simbolo usato per l'ascissa è sempre **I** a prescindere dal piano di interpolazione attivo.

J Ordinata del centro del cerchio. Questo valore, espresso in millimetri o pollici, può essere programmato direttamente o indirettamente con un parametro E. Quando l'asse corrispondente è un asse diametrale, l'ordinata viene espressa in modo diametrale. Il simbolo usato per l'ordinata è sempre **J** a prescindere dal piano di interpolazione attivo.

R Raggio del cerchio, alternativo alle coordinate I e J. Se l'arco di cerchio è minore o uguale a 180 gradi, il raggio deve essere programmato con segno positivo; se l'arco di cerchio è maggiore di 180 gradi il raggio deve essere programmato con segno negativo.

NOTA: Non si può usare la **R** per archi di 360 gradi.

- F** Velocità di avanzamento utilizzata per lo spostamento. Viene programmata con l'indirizzo F seguito dal valore di avanzamento. Se assente, la velocità sarà quella programmata in precedenza. Se in precedenza non è stata programmata alcuna velocità, il controllo genererà un errore.
- a** Accelerazione da usare sul profilo.
- ausiliarie* Funzioni ausiliarie programmabili M, S, T. Nel blocco possono essere programmate fino a quattro funzioni M, una S (velocità di rotazione del mandrino) e una T (selezione dell'utensile).

Caratteristiche:

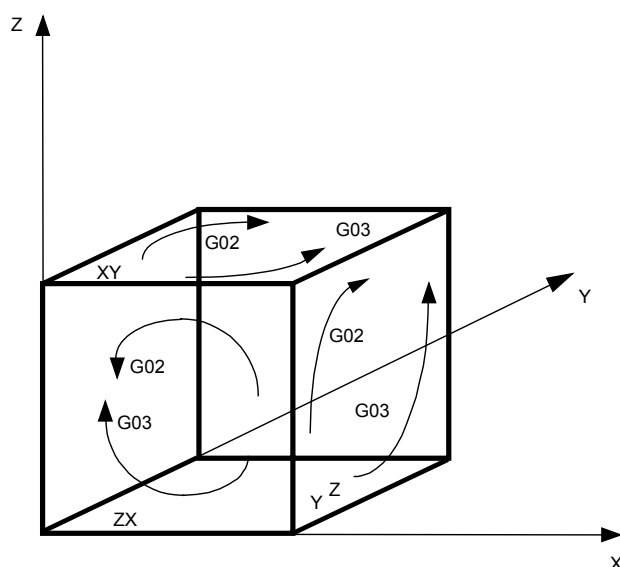
Il massimo arco programmabile è di 360 gradi, cioè un cerchio completo. Il piano di interpolazione va definito in precedenza tramite i codici G16, G17, G18, G19. All'accensione viene comunque attivato G17.

Le coordinate del punto iniziale (programmate nel blocco precedente), del punto finale e del centro della circonferenza devono essere calcolate in modo tale che la differenza tra raggio iniziale e raggio finale sia inferiore al valore di default (0,01 mm o 0,00039 pollici). Se la differenza è uguale o maggiore il controllo visualizza un messaggio di errore e la circonferenza non viene eseguita.

L'interpolazione circolare può essere programmata anche in modo incrementale (G91), cioè con le coordinate del punto finale e del centro della circonferenza riferite a quelle del punto iniziale programmato nel blocco precedente.

Il verso (orario o antiorario) dell'interpolazione circolare viene stabilito guardando il piano sul quale viene eseguita dal lato positivo dell'asse normale ad esso (vedere esempio).

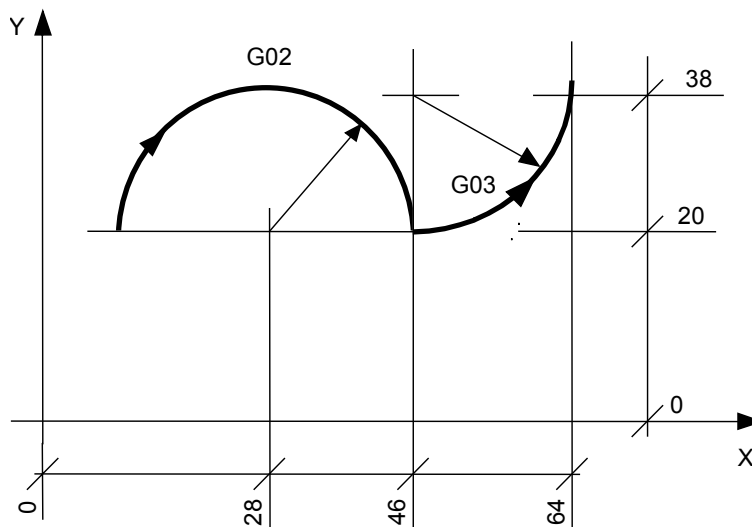
La figura seguente mostra le direzioni di interpolazione circolare riferite ai piani di esecuzione.



Direzioni per interpolazione circolare

Interpolazione circolare in programmazione assoluta con le coordinate I e J del centro del cerchio.

```
N14 X10 Y20  
N15 G2 X46 Y20 I28 J20 F200  
N16 G3 X64 Y38 I46 J38
```



Interpolazione circolare in programmazione assoluta con il valore R del raggio del cerchio.

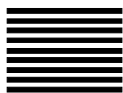
```
N14 X10 Y20  
N15 G2 X46 Y20 R18 F200  
N16 G3 X64 Y38 R18
```

Interpolazione circolare in programmazione incrementale con le coordinate.

```
N14 X10 Y20  
N15 G2 G91 X36 I18 J0 F200  
N16 G3 X18 Y18 I0 J18
```

Interpolazione circolare in programmazione incrementale con il valore del raggio.

```
N14 X10 Y20  
N15 G2 G91 X36 R18 F200  
N16 G3 X18 Y18 R18
```

CET (PRC) - Tolleranza di precisione in interpolazione circolare

Nelle interpolazioni circolari il codice CET definisce la tolleranza di precisione entro la quale deve stare la differenza tra raggio iniziale e raggio finale dell'arco di cerchio.

Sintassi

CET = *valore*

dove:

valore E' il valore della tolleranza di precisione espresso in millimetri.
 Il valore di default è 0.01 mm.

Caratteristiche:

Se la differenza fra il raggio iniziale e quello finale è minore ma non è zero, il sistema effettua una normalizzazione geometrica sui dati del cerchio secondo i valori specificati nei codici CET e ARM.

Se la differenza è uguale o maggiore del valore assegnato a CET, si verifica un errore ed i punti finali programmati non sono eseguiti. Se ciò accade, occorre modificare il programma oppure aumentare la tolleranza definita da CET.

Il valore assegnato a CET può essere modificato nei seguenti modi:

- Tramite AMP al momento della configurazione
- Tramite un data entry specifico
- Tramite part program con la sintassi descritta.

I valori assegnati a CET sono sempre espressi nell'unità di misura corrente del processo (vengono applicate le funzioni G70/G71) .

Nel caso in cui la programmazione di un arco di cerchio comporti una differenza tra il raggio iniziale e finale maggiore del valore CET, il cerchio può essere comunque eseguito in due modi:

- Aumentando il valore assegnato a CET (porlo maggiore della differenza dei raggi)
- Programmando l'arco di cerchio specificando il raggio anziché il centro del cerchio:

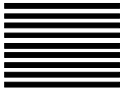
G2/G3, punto finale e raggio R

Con un'operazione di reset si ripristina la tolleranza di precisione di default.

Esempio:

CET=0.02

In questo esempio il valore della tolleranza definito da CET è di 0.02 millimetri.



FCT - Soglia per cerchio completo

Nelle interpolazioni circolari il codice FCT definisce la soglia della distanza fra il punto iniziale e quello finale in un arco. Entro tale distanza l'arco è considerato un cerchio completo.

Sintassi

FCT = *valore*

dove:

valore E' il valore della soglia per cerchio completo espresso in millimetri.

Il valore di default è 0.001 mm.

Caratteristiche:

Il comando FCT consente di superare eventuali imprecisioni di programmazione che altrimenti impedirebbero al sistema di forzare un cerchio completo. Quando la distanza fra il primo e l'ultimo punto è inferiore alla soglia FCT, il sistema utilizza i punti come coincidenti e forza un cerchio completo.

Il valore di soglia assegnato a FCT può essere modificato nei seguenti modi:

- Tramite AMP al momento della configurazione
- Tramite un data entry specifico
- Tramite part program con la sintassi descritta

I valori assegnati a FCT sono sempre espressi nell'unità di misura corrente del processo (vengono applicate le funzioni G70/G71).

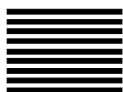
Il reset ripristina i valori di caratterizzazione.

Esempio:

G71

FCT=0.005

In questo esempio il valore della soglia definito da FCT è di 0.005 millimetri.



ARM - Definizione della modalità di normalizzazione dell'arco

Il codice ARM definisce la modalità applicata dal sistema per normalizzare e rendere geometricamente congruente un arco programmato con le coordinate del centro (I e J) ed il punto finale.

La normalizzazione viene applicata nel caso in cui la differenza fra il raggio iniziale ed il raggio finale si situa entro la tolleranza di precisione per l'interpolazione circolare (tolleranza caratterizzata o tolleranza programmata via comando CET).

Prima di eseguire un arco, il sistema calcola la differenza fra i raggi iniziale e finale.

- Se la differenza fra i raggi iniziale e finale è zero, il controllo esegue l'arco programmato senza normalizzarlo.
- Se la differenza fra i raggi iniziale e finale è superiore al valore definito dall'utente in CET, il controllo non esegue il movimento ma si ferma e visualizza un messaggio di errore profilo.
- Se la differenza fra i raggi iniziale e finale rientra nella variazione definita dall'utente in CET, il sistema esegue il movimento con la normalizzazione specificata in ARM dall'utente.
- Se la distanza fra il punto iniziale e quello finale è minore della soglia FCT, il sistema forza il cerchio completo.

Sintassi

ARM = *modalità arco*

dove:

modalità arco E' un valore numerico che definisce la modalità di normalizzazione dell'arco.

I valori ammessi sono:

- 0** centro spostato entro la tolleranza definita da CET
- 1** punto di partenza spostato entro la tolleranza specificata da CET
- 2** centro spostato indipendente dalla tolleranza definita da CET
- 3** centro non spostato entro la tolleranza definita da CET

Il valore di default è zero.

Caratteristiche:

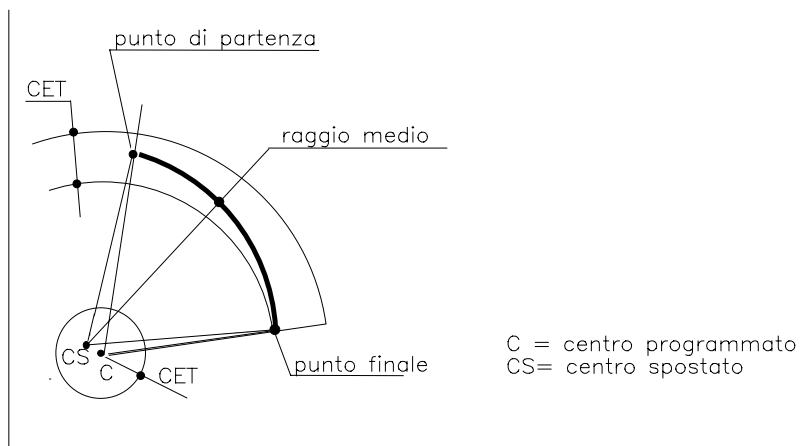
La modalità di normalizzazione dell'arco può essere modificata nei seguenti modi:

- Tramite AMP al momento della configurazione
- Tramite un data entry specifico
- Tramite part program con la sintassi descritta.

Le figure che seguono illustrano le varie modalità di normalizzazione.

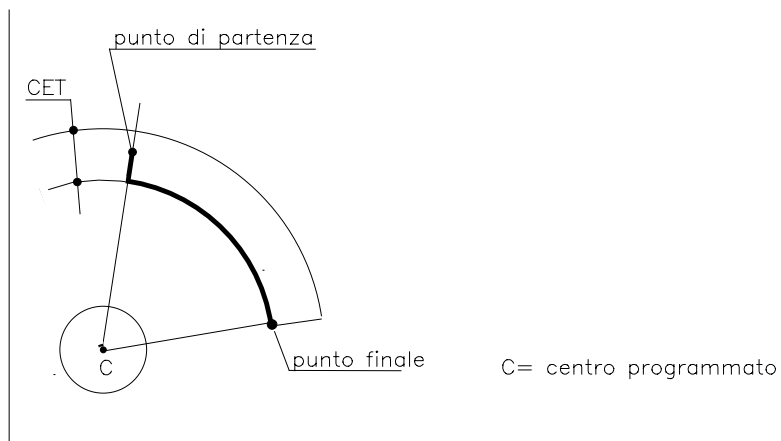
ARM=0

Questo è un arco passante per i punti iniziale e finale programmati, il cui centro è spostato entro la tolleranza definita da CET. In questo caso l'arco viene eseguito con un raggio medio.



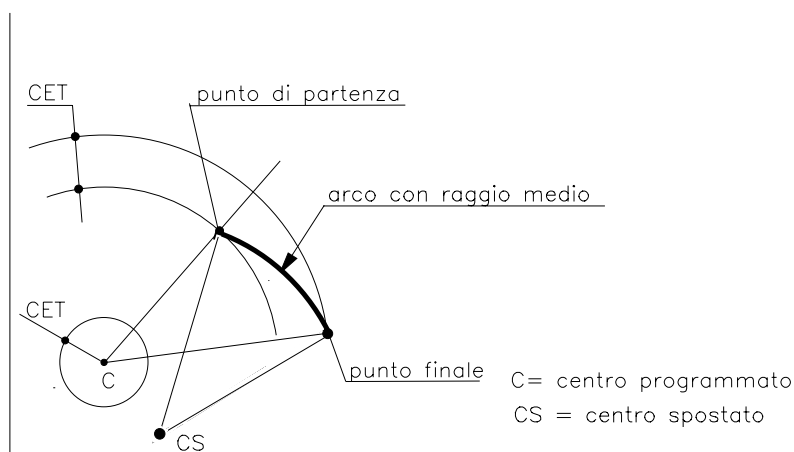
ARM=1

Questo è un arco passante per il punto iniziale corretto entro la tolleranza definita in CET e per il punto finale programmato.



ARM=2

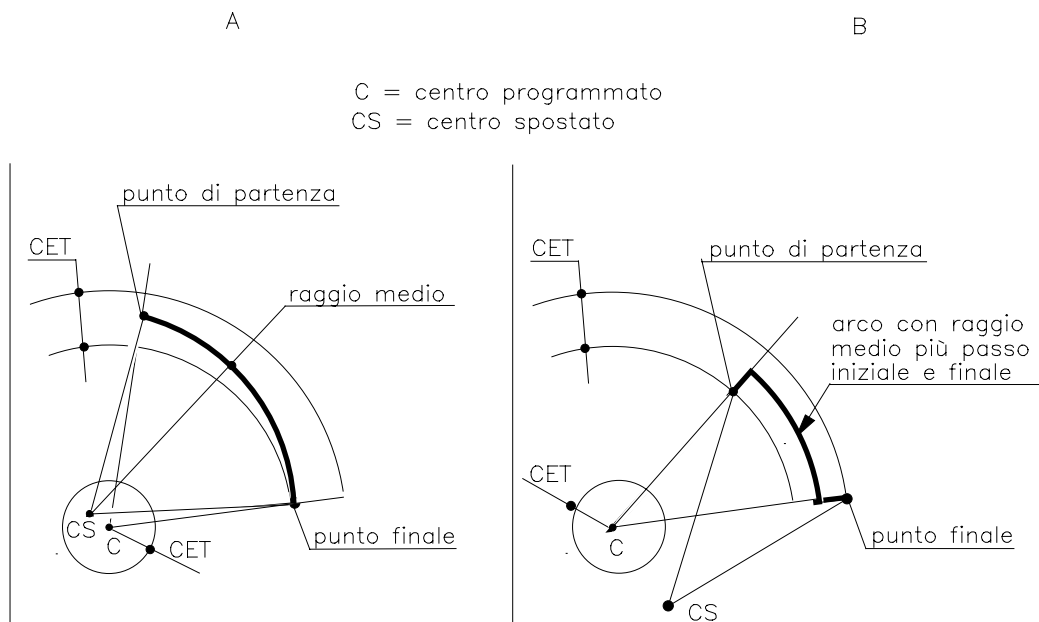
Questo è un arco il cui centro è spostato senza il vincolo della tolleranza definita in CET. In questo caso l'arco viene eseguito con un raggio medio.



ARM=3

Se lo spostamento del centro dell'arco avviene entro la tolleranza definita in CET, l'arco avrà un centro spostato e passerà per i punti iniziale e finale programmati. Se lo spostamento del centro non rientra nella tolleranza CET, l'arco avrà il centro programmato e passerà per i punti iniziale e finale spostati (entrambi i punti vengono spostati entro la tolleranza CET/2).

In questo caso l'arco è eseguito con raggio medio.



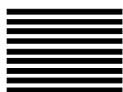
IMPORTANTE

Nella modalità ARM=1 o ARM=3, il profilo generato può presentare delle imperfezioni ("scalini") come visibile negli esempi.

Con ARM=1 si avrà un' imperfezione all'inizio dell'arco di cerchio pari alla differenza dei raggi iniziale e finale.

Con ARM=3 si avrà un' imperfezione sia all'inizio che alla fine dell'arco di cerchio.

Per evitare che le imperfezioni generate provochino "Servo Error", si consiglia di usare un valore di CET inferiore alla soglia di "Servo Error" caratterizzata.



CRT - Soglia di riduzione della velocità in interpolazione circolare

CRK - Costante di riduzione della velocità in interpolazione circolare

Le variabili CRT (Circle Reduction Threshold) e CRK (Circle Reduction Konstant) permettono di ridurre la velocità sugli enti circolari in funzione del raggio di tale ente.

Sintassi

CRT = valore

dove:

valore È il raggio di soglia al di sotto del quale applicare la riduzione. Il valore 0 (zero), che rappresenta il default, consente di modulare la limitazione della velocità dovuta all'accelerazione centrifuga.

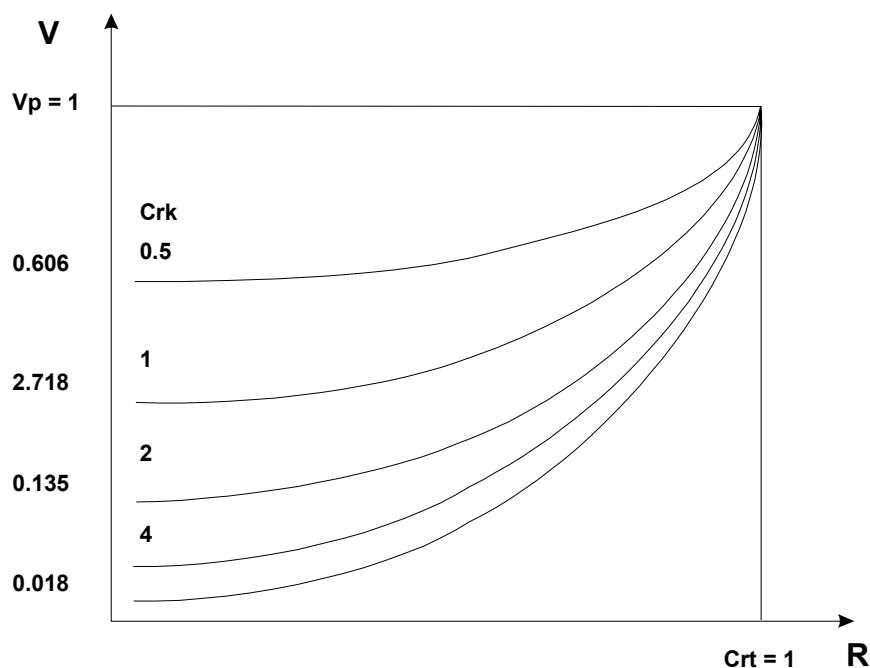
CRK = valore

dove:

valore È una costante che consente di modulare la riduzione. Il valore di default è 1.

Caratteristiche:

Assegnando un valore diverso da 0 alla variabile CRT viene ridotta la velocità su tutti gli enti circolari il cui raggio è minore del valore impostato. Il valore assegnato alla variabile CRK consente invece di modulare tale riduzione. La riduzione di velocità avviene secondo il grafico che segue, dove si suppone di avere una velocità programmata V_p uguale ad 1 ed un valore di CRT uguale ad 1.



Nel momento in cui CRT è uguale a 0, il valore CRK (se diverso da zero) viene utilizzato, nei movimenti circolari, per ricalcolare la velocità di esecuzione della lavorazione. Nei movimenti circolari, infatti, la velocità di lavorazione viene normalmente limitata dal raggio della circonferenza (accelerazione centrifuga) in base alla seguente legge:

$$V_{lav} = \text{Min} (\sqrt{a \text{ raggio}}, V_{\text{Prog}})$$

dove a , è l'accelerazione minima tra i due assi coinvolti nel movimento circolare.

Il CRK modifica la legge nel seguente modo

$$V_{lav} = \text{Min} (CRK * \sqrt{a \text{ raggio}}, V_{\text{Prog}})$$

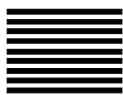
quindi consente di aumentare ($CRK > 1.0$) o diminuire ($CRK < 1.0$) la limitazione dovuta all'accelerazione centrifuga. Il valore 1.0 mantiene il calcolo standard

I valori assegnati alle variabili CRT e CRK possono essere modificati nei seguenti modi:

- tramite AMP all'atto della configurazione
- tramite part program con la sintassi descritta

I valori assegnati a CRT sono sempre espressi nell'unità di misura corrente del processo (vengono applicate le funzioni G70/G71).

Il Reset ripristina i valori di caratterizzazione.



Interpolazione elicoidale

I codici G02 o G03 permettono di programmare, in un solo blocco di programma, una traiettoria elicoidale da eseguirsi con il movimento contemporaneo degli assi del piano in interpolazione circolare e dell'asse ortogonale al piano in interpolazione lineare.

Per programmare una traiettoria elicoidale è sufficiente aggiungere, nel blocco dell'interpolazione circolare, la quota di profondità ed il passo dell'elica (K). Il formato è:

Sintassi

G02 [codici-G] [assi] I.. J.. K.. [F..] [ausiliarie]

oppure

G02 [codici-G] [assi] R.. K.. [F..] [ausiliarie]

G03 [codici-G] [assi] I.. J.. K.. [F..] [ausiliarie]

oppure

G03 [codici-G] [assi] R.. K.. [F..] [ausiliarie]

dove:

codici-G Altri codici G compatibili con G02 e G03 (Vedi tabella "Codici G compatibili" nel capitolo 1).

assi Lettera corrispondente ad un asse seguita da un valore numerico programmato direttamente con un valore decimale o indirettamente con un parametro E.

Se non vi sono assi programmati nel blocco, lo spostamento è un cerchio completo nel piano di interpolazione attivo.

I Ascissa del centro del cerchio. Questo valore, espresso in millimetri o pollici, può essere programmato direttamente o indirettamente con un parametro E. Quando l'asse corrispondente è un asse diametrale, l'ascissa viene espressa in modo diametrale. Il simbolo usato per l'ascissa è sempre **I** a prescindere dal piano di interpolazione attivo.

J Ordinata del centro del cerchio. Questo valore, espresso in millimetri o pollici, può essere programmato direttamente o indirettamente con un parametro E. Quando l'asse corrispondente è un asse diametrale, l'ordinata viene espressa in modo diametrale. Il simbolo usato per l'ordinata è sempre **J** a prescindere dal piano di interpolazione attivo.

- R** Raggio del cerchio. Viene programmato con l'indirizzo R seguito dal valore del raggio ed è alternativo alle coordinate I e J.
- K** Passo dell'elica. Questo parametro viene programmato con l'indirizzo K seguito dal valore del passo. Può essere omesso quando la profondità dell'elica è minore di un passo.
- F** Velocità di avanzamento. Viene programmata con l'indirizzo F seguito dal valore di avanzamento. Se assente, la velocità sarà quella programmata in precedenza. Se non è stata programmata alcuna velocità il controllo genererà un errore.
- ausiliarie* Funzioni ausiliarie programmabili M, S, T. Nel blocco possono essere programmate fino a quattro funzioni M, una S (velocità di rotazione del mandrino) e una T (selezione dell'utensile).

Caratteristiche:

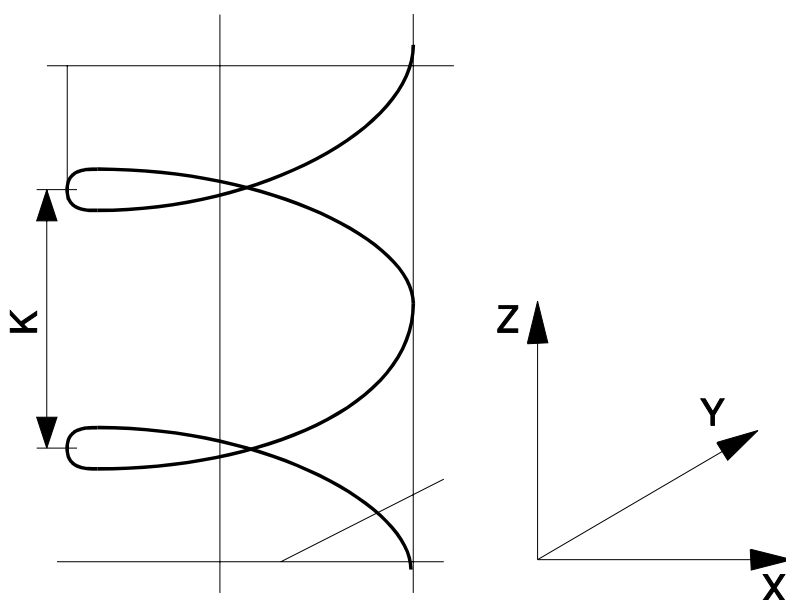
Se Z (lunghezza dell'elica) è un multiplo di K, non è necessario programmare il punto finale.

Se la lunghezza dell'elica non è uguale ad un numero intero di passi (Z diverso da $n * K$), la lunghezza dell'arco di cerchio deve essere calcolata con il resto decimale del numero di passo. Per esempio, se $Z = 2,7 * K$, allora l'arco che deve essere programmato è $360 * (2,7 - 2) = 252$ gradi.

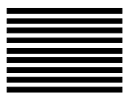
Esempio:

G2 X...Y...Z...I...J...K...F...

In questo esempio, gli indirizzi X, Y, I, J vengono utilizzati per la programmazione del cerchio; gli indirizzi Z e K vengono utilizzati per la programmazione dell'elica e sono rispettivamente la lunghezza e il passo dell'elica. La figura seguente mostra le dimensioni dell'interpolazione elicoidale.



Dimensioni dell'elica



G33 - Filettatura a passo costante o variabile

G33 definisce un movimento di filettatura cilindrica o conica a passo costante o variabile. Il movimento è coordinato con la rotazione del mandrino. I parametri programmati nel blocco identificano il tipo di filettatura da eseguire.

Sintassi

G33 [*assi*] **K**.. [**I**..] [**R**..]

dove:

- assi** Lettera corrispondente ad un asse seguita da un valore numerico programmato direttamente con un valore decimale o indirettamente con un parametro E.
- K** Passo di filettatura che deve sempre essere programmato. Per filettature a passo variabile, K è il passo iniziale.
- I** Variazione di passo per filettature a passo variabile. Per filettature a passo crescente I deve essere positivo, per filettature a passo decrescente I deve essere negativo.
- R** Sfasamento rispetto alla posizione angolare zero del mandrino espressa in gradi. R viene utilizzato nelle filettature a più principi per non spostare il punto di partenza.

Caratteristiche:

Tutti i valori numerici possono essere programmati direttamente con numeri decimali o indirettamente con parametri E. Per filettature a passo decrescente, è necessario calcolare il passo iniziale, la variazione di passo e la lunghezza della filettatura, in modo tale che il passo sia maggiore di zero prima di raggiungere la quota finale.

Usare la seguente formula:

$$I \leq \frac{K^2}{2 (Z_f - Z_i)}$$

dove:

- I** È la massima variazione di passo
- K** È il passo iniziale
- (Z_f - Z_i)** È la lunghezza della filettatura.

IMPORTANTE

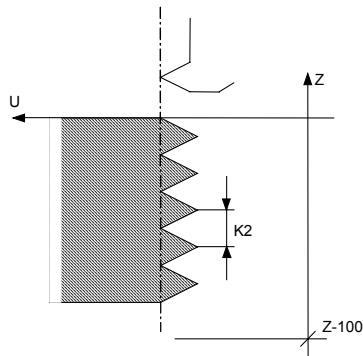
Durante un ciclo di filettatura il controllo non prende in considerazione il pulsante CYCLE STOP ed il selettore (o la softkey) di FEED OVERRIDE. La disabilitazione del selettore di SPINDLE SPEED OVERRIDE, deve essere eseguita dalla logica di macchina. E' possibile inoltre disabilitare il VFF tramite la softkey dedicata o con il comando VFF.

Filettature a passo costante

Le figure seguenti illustrano alcuni esempi di filettatura a passo costante. Si noti che l'asse U è un asse diametrale.

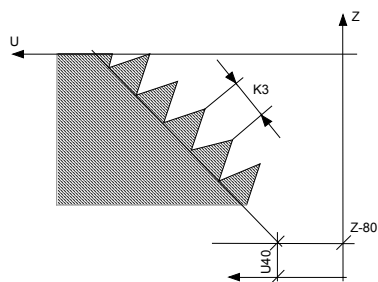
Filettatura cilindrica

Blocco di part program: G33 Z-100 K2



Filettatura conica

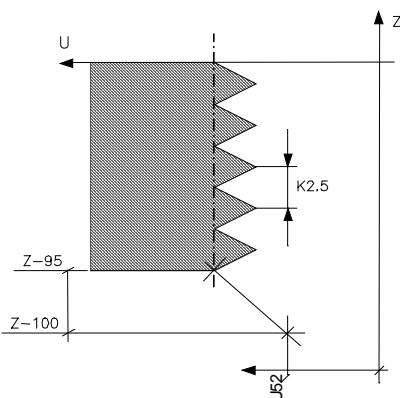
Blocco di part program: G33 U40 Z-80 K3



Filettatura cilindrica con uscita conica

Blocchi di part program : G33 Z-95 K2 .5

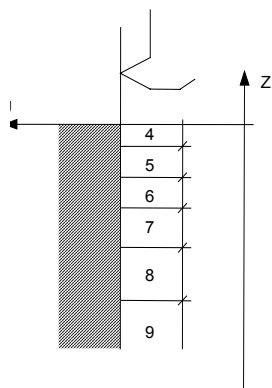
Z-100 U52 K2.5



Filettatura a passo variabile

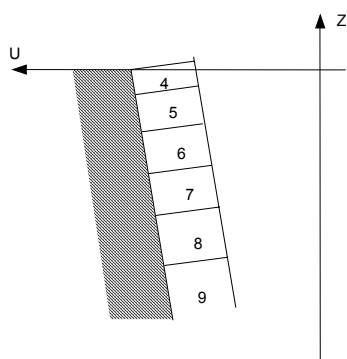
Le figure seguenti illustrano esempi di filettatura a passo variabile. Si noti che l'asse U è un asse diametrale.

Filettatura cilindrica con passo crescente



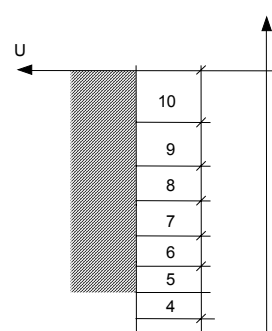
Blocco di part program: G33 Z-50 K4 I1

Filettatura conica con passo crescente



Blocco di part program: G33 U50 Z-40 K4 I1

Filettatura cilindrica con passo decrescente



Blocco di part program: G33 Z-50 K10 I-1

Filettature a più principi

La funzione R in un blocco G33 fa sì che il controllo dia il comando di partenza degli assi in posizione angolare diversa a seconda del valore R programmato.

In questo modo è possibile programmare per i vari filetti un unico punto di partenza, a differenza di altri sistemi nei quali è necessario, per ottenere filettature a più principi, spostare il punto di partenza di ogni filetto di una quantità pari al passo divisa per il numero dei principi.

Esempio:

Questo esempio mostra una filettatura a tre principi.

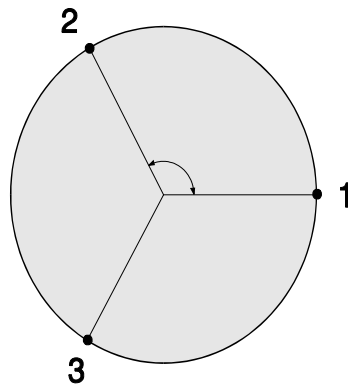
N37 G33 Z3 K6 **primo filetto**

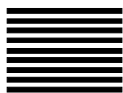
.
. .
. .

N41 G33 Z3 K6 R120 **secondo filetto**

.
. .
. .

N45 G33 Z3 K6 R240 **terzo filetto**





Assi Rotativi

In caratterizzazione è possibile dichiarare un asse come rotativo (es.: una tavola girevole).

Per poter programmare un asse rotativo con movimento simultaneo e coordinato con gli altri assi programmati nello stesso blocco:

- La programmazione deve essere sempre in gradi decimali (da +0,00001 a +99999.99999 gradi) a partire da un'origine prefissata.
- Il movimento può essere eseguito a velocità di rapido con G00 o a velocità di lavoro con G01. La velocità di rotazione F si esprime in gradi al minuto (formato F5.5). Per esempio, se F75.5 l'asse si sposta ad una velocità di 75,5 gradi al minuto.

Per effettuare delle fresature lungo una circonferenza utilizzando la tavola girevole, si può calcolare la velocità di avanzamento angolare da programmare con la seguente formula:

$$F = \frac{360}{\pi} * \frac{A}{D} = 114,64 * \frac{A}{D}$$

dove:

- F** È la velocità di avanzamento angolare espressa in gradi al minuto
- A** È la velocità di avanzamento lineare lungo l'arco espressa in millimetri al minuto o pollici al minuto
- D** È il diametro sul quale viene eseguita l'operazione di fresatura (espresso in mm o pollici).

Quando gli assi rotativi e gli assi lineari si devono spostare simultaneamente nello stesso blocco, usare una delle due formule seguenti (con G94 o G93) per calcolare la velocità di avanzamento:

con G94:

$$F = A * \frac{\sqrt{X^2+Y^2+Z^2+B^2+C^2}}{L}$$

dove:

- F** È la velocità di avanzamento da programmare
- A** È la velocità di avanzamento (in mm/min o pollici/min) voluta sul pezzo da lavorare.
- X Y Z B C** È la corsa effettiva eseguita da ogni asse (in mm o pollici per assi lineari, in gradi per assi rotativi)
- L** È la lunghezza del percorso risultante (in mm o pollici)

con G93:

$$F = \frac{A}{\sqrt{X^2 + Y^2 + B^2}}$$

dove:

- F** È la velocità di avanzamento da programmare
- A** È la velocità di avanzamento (in mm/min o pollici/min) sul pezzo
- X** È la distanza incrementale X dall'asse
- Y** È la distanza incrementale Y dall'asse
- B** È la distanza incrementale B dall'asse

Il controllo non può calcolare direttamente la velocità di avanzamento dell'utensile perché il raggio non è programmato. In questi casi viene utilizzata la velocità di avanzamento definita come inversa al tempo che può essere stabilita con G93.

Un blocco che abbia solo spostamenti di assi rotativi produce un arco. Combinando spostamenti rotativi con spostamenti lineari, il percorso risultante può essere una spirale di Archimede, un'elica cilindrica o una curva ancora più complessa, a seconda del numero di assi lineari che sono stati programmati.



Assi Rollover

L'asse rollover è un asse rotativo o lineare la cui posizione è controllata tra zero ed un valore positivo definito in configurazione come " Passo Rollover ".

Nella trattazione che segue si suppone di avere un'asse rollover rotativo con passo di 360 gradi. In questo caso la posizione dell'asse è controllata nell'intervallo da 0 a 359,9999 gradi. Questo significa che quando l'asse raggiunge 360 gradi, la posizione visualizzata sarà 0 gradi.

Un asse rollover può essere programmato da part program o da tastiera in due modi differenti:

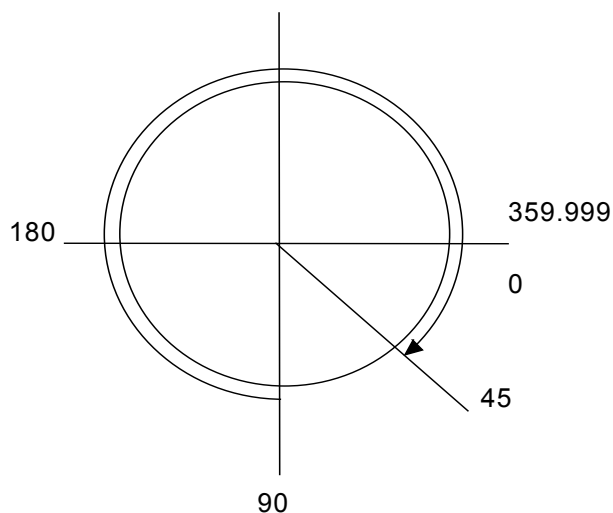
modo assoluto	Programmando lo spostamento in gradi.
modo incrementale	Programmando lo spostamento in gradi a partire dalla posizione corrente dell'asse.

G90 - Modo assoluto

Se un asse rollover viene programmato in modo assoluto, vengono applicate le seguenti condizioni:

- La posizione visualizzata è da 0 a +359,99999 gradi
- L'intervallo programmabile va da 0 a $\pm 359,99999$ gradi
- La direzione di rotazione degli assi dipende dal segno anteposto alla quota programmata. Per convenzione, uno spostamento positivo si intende in senso orario, uno negativo in senso antiorario.

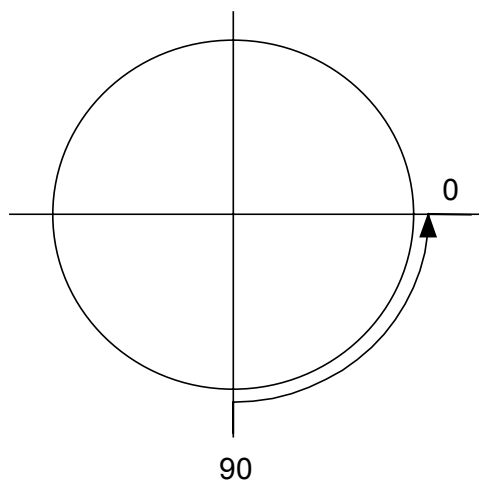
Per esempio, si supponga che l'asse rotativo B sia posizionato a 90 gradi e che da part program o da tastiera venga dato il seguente blocco:

G90 B45

Rotazione oraria

L'asse B ruota di 315 gradi in senso orario dalla posizione di 90 gradi fino a raggiungere la posizione assoluta di 45 gradi, in quanto il segno dello spostamento è positivo.

Sempre per esempio si supponga che l'asse rotativo B sia posizionato sui 90 gradi e, da tastiera o da part program venga dato il seguente blocco:

G90 B-0

Rotazione antioraria

L'asse B ruota di 90 gradi in senso antiorario fino a raggiungere la posizione assoluta di zero gradi, in quanto il segno dello spostamento è negativo.

G91 - Modo incrementale

Se un asse rolover viene programmato in modo incrementale (G91), vengono applicate le seguenti condizioni:

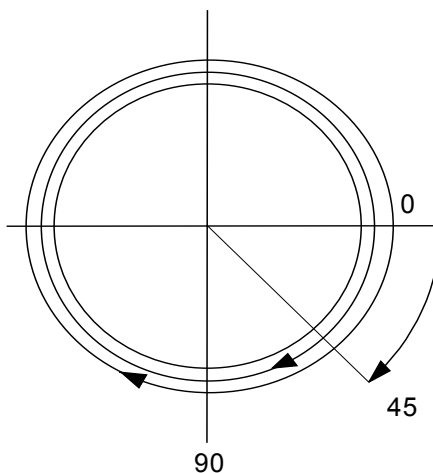
- La posizione visualizzata va da 0 a 359,99999 gradi.
- L'intervallo programmabile va da +/-0.00001 a +/-99999.99999 gradi.
- La direzione di rotazione dell'asse dipende dal segno dello spostamento programmato. Per convenzione, uno spostamento positivo è orario, uno negativo è antiorario.

IMPORTANTE

La posizione visualizzata è fuori dall'intervallo programmato quando quest'ultimo supera +359.99999 gradi.

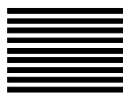
Per esempio, supponiamo che l'asse rotativo B sia posizionato sulla posizione assoluta zero gradi, e che il seguente blocco venga dato da tastiera o da part program:

G91 B765



Rotazione oraria incrementale

L'asse B ruota in senso orario compiendo due giri completi più uno spostamento addizionale di 45 gradi ($360 + 360 + 45 = 765$).



Pseudo assi

Gli pseudo assi sono funzioni ausiliarie che possono essere indirizzate come assi e vengono gestite dalla logica di macchina. Esempio: regolazione del getto per taglio ad acqua, gestione mandrino secondario. Il nome di uno pseudo asse può essere una qualsiasi delle lettere disponibili per gli assi normali (X,Y,Z,A,B,C,U,V,W,P,Q,D). E' possibile programmare fino a tre pseudo assi in un blocco di part program. Nella caratterizzazione (AMP) possono essere dichiarati fino a 6 pseudo assi.



Assi diametrali

E' possibile installare sul mandrino una testa per alesare/sfacciare e controllarla simultaneamente ad altri assi. Programmando un tale asse diametrale (generalmente denominato U), si possono ottenere:

- Operazioni di alesatura di fori cilindrici o conici
- Raccordi circolari (concavi o convessi)
- Smussi
- Gole
- Sfacciatore
- Filettature

La programmazione di un asse diametrale è analoga a quella di altri assi lineari, ma le coordinate devono essere espresse in diametri. Le unità di misura possono essere pollici o millimetri a seconda del modo attivo (G70/G71).

Quando viene programmato in uno stesso blocco con gli assi X, Y o Z, il movimento di un asse diametrale è simultaneo e coordinato con questi e può avvenire a velocità di rapido (G00) o a velocità di lavoro (G01) espressa in mm/min o pollici/min con F.

Prima di eseguire un profilo con un asse diametrale, ad esempio U, deve essere definito il piano di interpolazione tramite il seguente comando:

G16 Z U

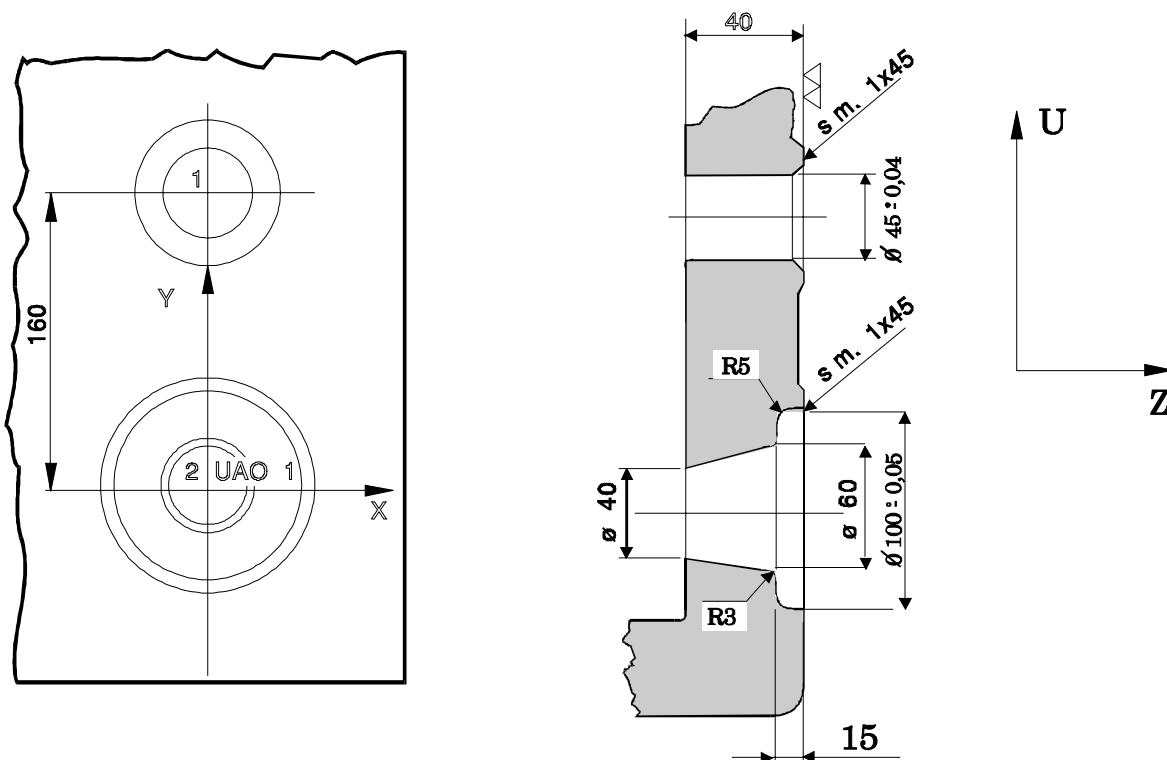
IMPORTANTE

L'ordine di Z ed U in questo comando è fondamentale. G16 UZ definisce, infatti, un piano di interpolazione differente.

Ai profili programmati con U è possibile applicare la compensazione diametro utensile (G41 o G42) e la definizione di sovrametallo (MSA).

Esempio:

Il seguente è un esempio di utilizzo di testa di alesatura/sfacciatura in un'operazione di finitura.



N116 (DIS, "FINITURA CON TESTA ALES. E SFACC.")

N117 F60 S630 T9 .9 M6

N118 G16 Z U

;Definisce il piano di interpolazione

N119 (UAO, 2)

;Chiama l'origine assoluta della testa

N120 (UTO, 1, Z-200)

;Origine temporanea per Z (a filo pezzo)

N121 X Y160 M3

;Posiziona sul foro 1

N122 G41 Z2 U51

N123 G1 Z-1 U44 .98

;Esegue lo smusso

N124 Z-44

;Esegue foro di diametro 45

N125 G G40 U40

N126 Z2 F40 S380

N127 G41 Y U106

;Posiziona sul foro 2

N128 G1 Z-1 U99 .975

;Esegue lo smusso

N129 Z-15

;Esegue foro di diametro 100

N130 r5

;Esegue raccordo R = 5

N131 U60

;Esegue battuta

N132 r-3

;Esegue raccordo R = 3

N133 Z-40 U40

;Esegue cono

N134 G40 Z-44

;Continua la corsa dell'asse Z

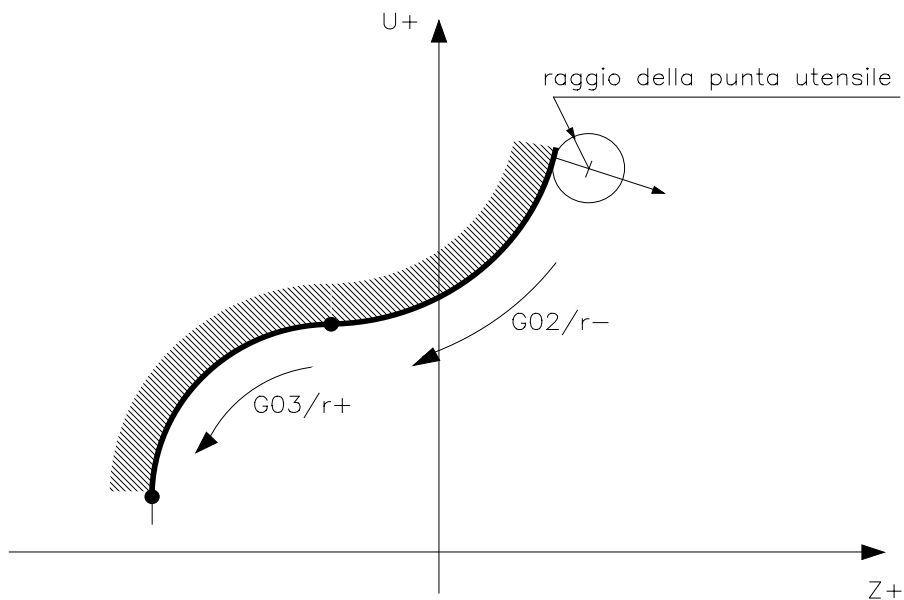
N135 G U35

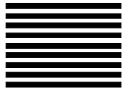
N136 Z100 M5

N137 G16 X Y

N138 (UAO,1)

Il senso di percorrenza degli archi di cerchio (programmati con le funzioni G02 o G03 oppure con l'indirizzo r) e la funzione di abilitazione della correzione del raggio utensile sul profilo (G41 o G42) sono rintracciabili guardando il profilo da eseguire nel piano Z-U. Poichè i diametri negativi di solito non vengono programmati, è sufficiente considerare solo i primi due quadranti del piano.





UDA - Assi duali

E' possibile trattare uno o più assi come "slave" ossia subordinati ad un altro definito "master". In tal modo occorre programmare unicamente gli spostamenti dei master, poiché il movimento degli slave è determinato da quelli del master a cui sono associati e dalla presenza o meno su di essa dell'inversione "mirror".

Sintassi

(UDA, master1/slave1[, master2/slave2, master3/slave3, master4/slave4])
(UDA)

dove:

<i>master1. . . master4</i>	Sono i nomi degli assi <i>master</i> (ciascuno designato da un singolo carattere ASCII). Si possono programmare fino a 4 assi master.
<i>slave1. . . slave8</i>	Sono i nomi degli assi <i>slave</i> (ciascuno designato da un singolo carattere ASCII). Si possono programmare un massimo di 8 assi slave per ogni asse master.
<i>nessun parametro</i>	(UDA) senza parametri disabilita la modalità assi duali.

Caratteristiche:

Gli assi duali non richiedono alcuna impostazione speciale durante la caratterizzazione del sistema con AMP.

Dopo un comando (UDA...) il limite operativo positivo risulta essere il minore tra il limite positivo dell'asse master e la posizione corrente del master più la quantità di spazio percorribile dall'asse slave. In breve:

$LimPositivo = \min(LimPositivoMaster, PosizioneMaster + LimPositivoSlave - Posizione Slave)$

Nel caso di "mirror" lo spazio percorribile dallo slave è riferito al suo limite negativo, per cui sarà:

$LimPositivo = \min(LimPositivoMaster, PosizioneMaster - LimNegativoSlave + Posizione Slave)$

Le stesse considerazioni fatte per il limite positivo si possono estendere naturalmente anche per il limite negativo:

$LimNegativo = \max(LimNegativoMaster, PosizioneMaster + LimNegativoSlave - Posizione Slave)$

Nel caso di "mirror" sarà:

$LimNegativo = \max(LimNegativoMaster, PosizioneMaster - LimPositivoSlave + Posizione Slave)$

All'attivazione del comando di (UDA,...) sia l'asse master che gli slave devono essere riferiti.

Il comando di RESET non toglie l'associazione master/slave.

È possibile utilizzare assi duali su piani ruotati (UPR) o in coordinate polari o cilindriche (UVP, UVC).

La definizione di assi duali, sia master che slave, **deve** essere fatta su assi reali e non su assi virtuali.

NOTA:

Il nome degli assi master e slave deve essere separato da una barra "/".

IMPORTANTE

Se si vuole rendere speculare lo spostamento dell'asse "slave" rispetto a quello "master", bisogna programmare l'operatore "-" prima del nome dell'asse. Ciò non è valido per l'asse master.

Esempio:

(UDA,X/-U)

U è reso slave e speculare rispetto ad X

(UDA,A/B-CD)

B,C,D sono resi slaves rispetto ad A, inoltre C è speculare rispetto ad A



SDA - Assi duali speciali

E' possibile trattare uno o più assi come "slave" ossia subordinati ad un altro definito "master". In tal modo occorre programmare unicamente gli spostamenti dei master, poiché il movimento degli slave è determinato da quelli del master a cui sono associati e dalla presenza o meno su di essa dell'inversione "mirror". La movimentazione dell'asse master e degli slave può avvenire senza che gli assi siano stati riferiti.

Sintassi

(SDA, master1/slave1[, master2/slave2, master3/slave3, master4/slave4])
(SDA)

dove:

master1 . . . *master4* Sono i nomi degli assi *master* (ciascuno designato da un singolo carattere ASCII). Si possono programmare fino a 4 assi master.

slave1 . . . *slave8* Sono i nomi degli assi *slave* (ciascuno designato da un singolo carattere ASCII). Si possono programmare un massimo di 8 assi slave per ogni asse master.

nessun parametro (SDA) senza parametri disabilita modalità assi duali

Caratteristiche:

Gli assi duali non richiedono alcuna impostazione speciale durante la caratterizzazione del sistema con AMP.

Dopo un comando (SDA...) il limite operativo positivo risulta essere il minore tra il limite positivo dell'asse master e la posizione corrente del master più la quantità di spazio percorribile dall'asse slave. In breve:

$LimPositivo = \min(LimPositivoMaster, PosizioneMaster + LimPositivoSlave - Posizione Slave)$

Nel caso di "mirror" lo spazio percorribile dallo slave è riferito al suo limite negativo, per cui sarà:

$LimPositivo = \min(LimPositivoMaster, PosizioneMaster - LimNegativoSlave + Posizione Slave)$

Le stesse considerazioni fatte per il limite positivo si possono estendere naturalmente anche per il limite negativo:

$LimNegativo = \max(LimNegativoMaster, PosizioneMaster + LimNegativoSlave - Posizione Slave)$

Nel caso di “mirror” sarà:

$LimNegativo = \max(LimNegativoMaster, PosizioneMaster - LimPositivoSlave + Posizione Slave)$

All'attivazione del comando di (SDA,...) sia l'asse master che gli slave possono non essere riferiti. Il comando di RESET non toglie l'associazione master/slave.

Questo comporta la possibilità di eseguire il ciclo di ricerca micro di zero con la dualizzazione attiva. In questo caso eseguendo il ciclo di ricerca micro sull'asse master, il sistema muoverà in contemporanea gli assi slave associati; a fine ricerca l'asse master risulta riferito mentre gli assi slave non risultano riferiti. Per riferire gli assi slave è necessario scambiarli, uno ad uno con l'asse master tramite nuove programmazioni di SDA, e ripetere il ciclo di ricerca micro di zero su ognuno di essi, ridichiarati come master. L'operazione di azzeramento sugli slave non è comunque necessaria se si intende programmare il solo asse master con SDA attivo.

L'utilizzo della funzione SDA è raccomandata pertanto nei casi in cui si deve eseguire un ciclo di ricerca micro in seguito ad uno spegnimento del sistema, con il pezzo da lavorare ancora sulla tavola.

NOTA:

Il nome degli assi master e slave deve essere separato da una barra "/".

IMPORTANTE

Se si vuole rendere speculare lo spostamento dell'asse "slave" rispetto a quello "master", bisogna programmare l'operatore "-" prima del nome dell'asse. Ciò non è valido per l'asse master.

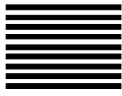
Esempio:

(SDA,X/-U)

U è reso slave e speculare rispetto ad X

(SDA,A/B-CD)

B,C,D sono resi slave rispetto ad A, inoltre C è speculare rispetto ad A



XDA - Assi Master/Slave

E' possibile trattare uno o più assi come "slave" ossia subordinati ad un altro definito "master". In tal modo occorre programmare unicamente gli spostamenti del master, poiché il movimento degli slave è determinato da quello del master a cui sono agganciati e da un fattore di inseguimento definito per ogni slave. Esistono diverse tipologie di comandi applicabili a tale prestazione, ogni comando ha una sintassi diversa.

Associazione Master/Slave

Con tale istruzione viene definita l'associazione tra l'asse master e gli assi slave (max 8 assi). Tale istruzione **NON attiva la fase di inseguimento**, la sua attivazione viene effettuata tramite una apposito comando; ciò significa che dopo tale istruzione un movimento dell'asse master non induce un movimento dello slave. **Dopo questa istruzione e sino a che non viene dissociato lo slave dal master non è possibile programmare alcun movimento dello slave.**

Sintassi

(XDA, 1, master/slave1[slave2[.]], modalità, rapporto, spazio)

dove :

master	E' il nome dell'asse master e viene designato da un singolo carattere ASCII.								
slave1...slave8	Sono i nomi degli assi slave (ciascuno designato da un singolo carattere ASCII). Si possono programmare sino ad un massimo di 8 slave.								
modalità	Definisce il metodo di inseguimento dell'asse master da parte dello slave e può valere : <table><tr><td>0</td><td>Lo slave insegue puntualmente il master</td></tr><tr><td>1</td><td>Lo slave insegue in velocità il master</td></tr><tr><td>2</td><td>Lo slave insegue in posizione il master</td></tr><tr><td>3</td><td>Lo slave insegue in posizione il master con recupero spazio di aggancio</td></tr></table>	0	Lo slave insegue puntualmente il master	1	Lo slave insegue in velocità il master	2	Lo slave insegue in posizione il master	3	Lo slave insegue in posizione il master con recupero spazio di aggancio
0	Lo slave insegue puntualmente il master								
1	Lo slave insegue in velocità il master								
2	Lo slave insegue in posizione il master								
3	Lo slave insegue in posizione il master con recupero spazio di aggancio								
rapporto	E' il rapporto di inseguimento del master da parte dello slave, tale valore deve essere considerato come fattore moltiplicativo della velocità o dello spazio percorso dal master. Con il valore 1.0 lo slave copia fedelmente il moto del master, con valori inferiori ad 1.0 lo riduce, con valori superiori ad 1.0 lo aumenta. Tale valore può essere segnato.								
spazio	E' lo spazio che lo slave deve percorrere per sincronizzarsi con il moto del master.								

Caratteristiche :

L'asse master può identificare sia un asse presente nel processo in cui viene attivato il comando XDA che un asse non presente; in quest'ultimo caso verrà creato un "asse virtuale" avente il nome specificato, tale asse avrà caratteristiche dinamiche ereditate dagli assi slave (la minore tra le velocità, accelerazioni e jerk). Tale asse potrà far parte di virtualizzazioni (UPR, UDA,...) ed anche di TCP. Non potrà essere eseguita l'operazione di Homing sull'asse master.

L'asse slave deve essere un asse presente nel processo in cui viene attivato il comando XDA; lo slave può essere comunque un asse SHARED, ovvero condiviso con l'ambiente di logica di macchina. In tal senso l'asse potrà essere mosso da logica di macchina anche dopo l'associazione con il master, non potrà essere comunque mosso nel momento in cui l'asse insegue il master. Tale asse non potrà far parte di alcuna virtualizzazione o TCP.

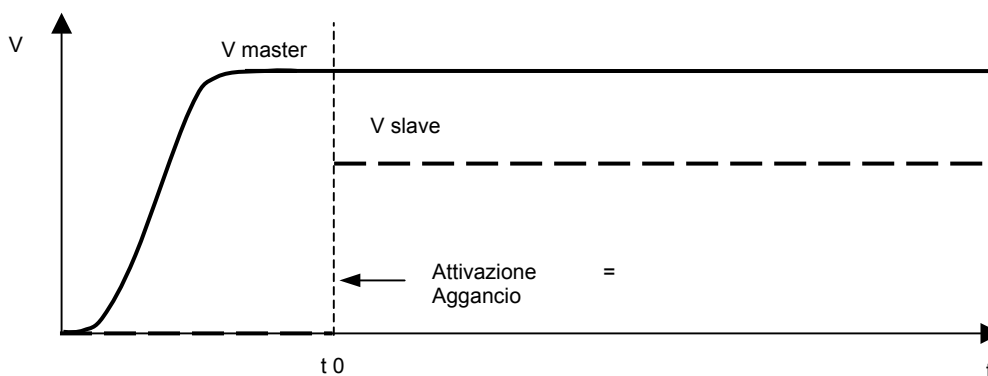
Vediamo ora le varie modalità di inseguimento possibili :

Modo 0

In questo caso l'asse slave insegue l'asse master proporzionalmente al valore di rapporto (se rapporto = 1 lo slave copia esattamente il movimento dell'asse master); l'aggancio (sincronismo) è istantaneo e la variazione di velocità dell'asse slave è a "*gradino*". I valori di posizione e di velocità dell'asse slave saranno calcolati, istante per istante, dalle seguenti formule :

$$V_{\text{slave}} = V_{\text{master}} * \text{FollowRate}$$

$$\text{PosSlave} = \text{PosSlave}_{t_0} + (\text{PosMaster} - \text{PosMaster}_{t_0}) * \text{FollowRate}$$

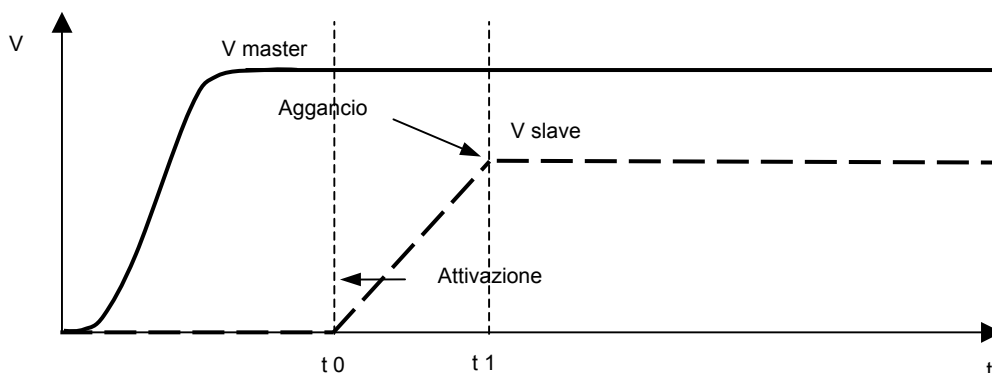


Se la richiesta di velocità risultante sull'asse slave risulta essere superiore alla massima velocità possibile per l'asse, il sistema limiterà la velocità richiesta a tale valore; di conseguenza verrà segnalata una emergenza (servo error) in quanto lo slave non riesce ad inseguire la posizione richiesta.

Modo 1

In questo caso l'asse slave insegue la velocità dell'asse master proporzionalmente al valore di rapporto (se rapporto = 1 lo slave copia esattamente il movimento dell'asse master); l'aggancio (sincronismo) è dipendente alle caratteristiche dinamiche dell'asse slave e/o dal parametro spazio che definisce lo spazio di sincronizzazione.

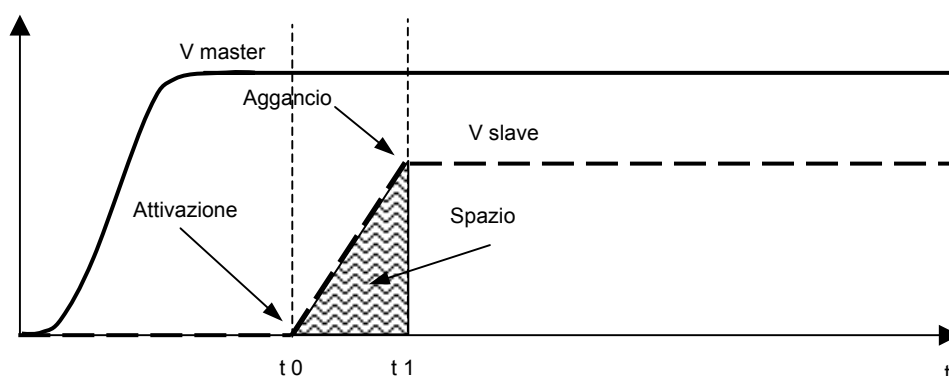
Nel caso in cui il valore di spazio è 0, l'asse slave si sincronizzerà all'asse master in base alla sua accelerazione massima ed utilizzando **esclusivamente** rampe lineari.



Nel caso in cui il valore di spazio è diverso da 0, l'asse slave si sincronizzerà all'asse master in base ad una accelerazione calcolata in base allo spazio di sincronizzazione ed utilizzando **esclusivamente** rampe lineari. L'accelerazione verrà ricalcolata ad ogni campionatura in base alla formula

$$A_{slave} = ((V_{master} * FollowRate)^2 + V_{slave}^2) / 2 * Spazio$$

in cui il valore dello spazio è man mano decrementato in base allo spazio percorso durante la fase di sincronizzazione. Non viene fatto alcun controllo sul valore dell'accelerazione risultante per cui potrebbero essere generati dei servo error nel momento in cui l'accelerazione è superiore a quella massima sopportabile dall'asse.



Nel momento in cui l'asse Master è stato agganciato lo slave si muoverà utilizzando la seguente formula :

$$V_{slave} = V_{master} * FollowRate$$

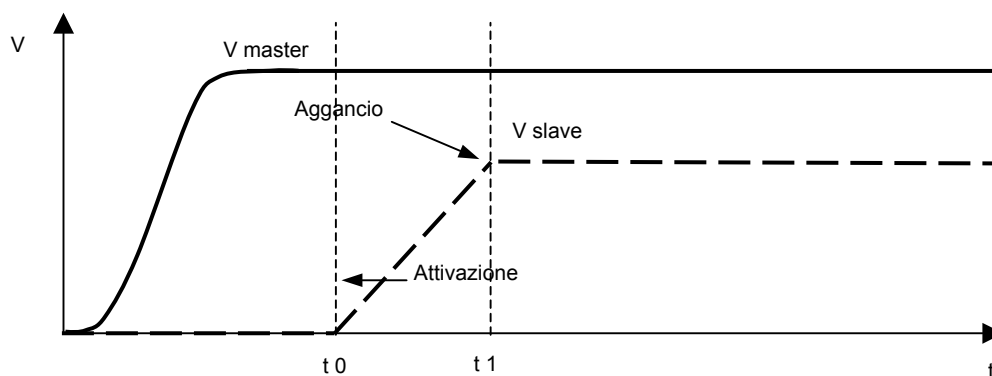
La velocità **Vslave** così calcolata è una velocità "teorica" in quanto occorre verificare se tale richiesta rispetta le caratteristiche dinamiche dell'asse (Velocità massima e accelerazione massima). Nel momento in cui la velocità dell'asse master varia, lo slave inseguirà tale variazione utilizzando il proprio valore di accelerazione. Quando la richiesta di velocità sull'asse slave risulta essere superiore alla massima velocità consentita, il sistema limiterà la velocità richiesta. In tal senso in ogni istante verranno calcolati i due valori di velocità **Vslave_i** e di accelerazione **Aslave_i** con cui muovere l'asse slave. La posizione dell'asse slave verrà quindi calcolata in base a tali valori :

$$PosSlave_{tn+1} = PosSlave_{tn} + Vslave_i + Aslave_i$$

Modo 2

In questo caso l'asse slave insegue la posizione dell'asse master proporzionalmente al valore di rapporto (se rapporto = 1 lo slave copia esattamente il movimento dell'asse master); l'aggancio (sincronismo) è dipendente alle caratteristiche dinamiche dell'asse slave e/o dal parametro spazio che definisce lo spazio di sincronizzazione.

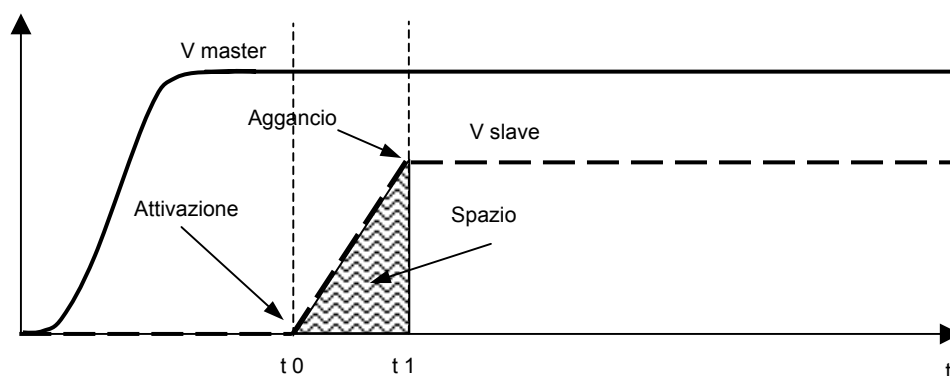
Nel caso in cui il valore di spazio è 0, l'asse slave si sincronizzerà all'asse master in base alla sua accelerazione massima ed utilizzando **esclusivamente** rampe lineari.



Nel caso in cui il valore di spazio è diverso da 0, l'asse slave si sincronizzerà all'asse master in base ad una accelerazione calcolata in base allo spazio di sincronizzazione ed utilizzando **esclusivamente** rampe lineari. L'accelerazione verrà ricalcolata ad ogni campionatura in base alla formula

$$A_{slave} = ((V_{master} * FollowRate)^2 + V_{slave}^2) / 2 * Spazio$$

in cui il valore dello spazio è man mano decrementato in base allo spazio percorso durante la fase di sincronizzazione. Non viene fatto alcun controllo sul valore dell'accelerazione risultante per cui potrebbero essere generati dei servo error nel momento in cui l'accelerazione è superiore a quella massima sopportabile dall'asse.



Nel momento in cui l'asse Master è stato agganciato lo slave si muoverà utilizzando le seguenti formule :

$$PosSlave = PosSlave_{t1} + (PosMaster - PosMaster_{t1}) * FollowRate$$

$$Vslave = Vmaster * FollowRate$$

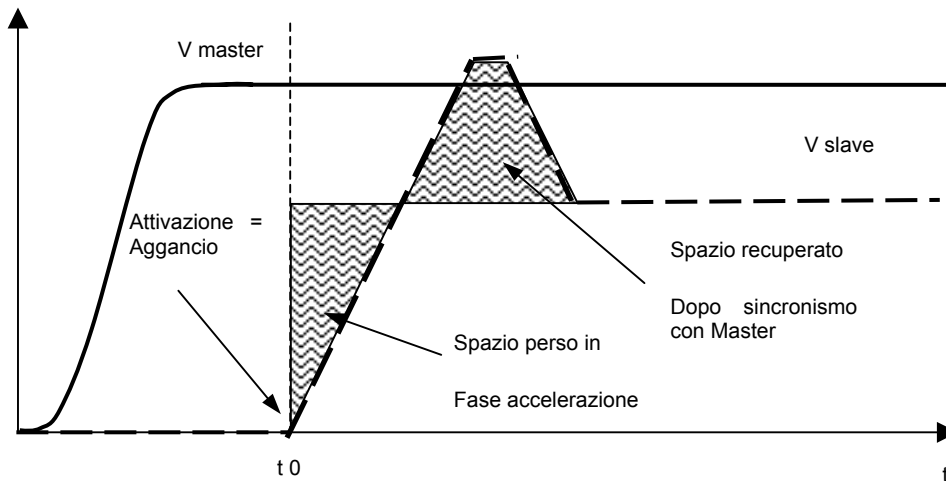
La posizione **PosSlave** e la velocità **Vslave** così calcolate sono da considerare come valori "teorici" in quanto occorre verificare se tale richieste rispettano le caratteristiche dinamiche dell'asse (Velocità massima e accelerazione massima). Nel momento in cui la velocità dell'asse master varia, lo slave inseguirà tale variazione utilizzando il proprio valore di accelerazione. Quando la richiesta di velocità sull'asse slave risulta essere superiore alla massima velocità consentita, il sistema limiterà la velocità richiesta. In tal senso in ogni istante verranno calcolati i due valori di velocità **Vslave_i** e di accelerazione **Aslave_i** con cui muovere l'asse slave. La posizione reale dell'asse slave verrà quindi calcolata in base a tali valori :

$$PosSlave_{tn+1} = PosSlave_{tn} + Vslave_i + Aslave_i$$

L'errore tra la posizione reale dell'asse e quella teorica viene recuperato dall'asse slave durante la movimentazione (ed anche quando la movimentazione dell'asse master risulta essere terminata) andando, quando possibile, a velocità superiore rispetto alla velocità **Vslave** calcolata.

Modo 3

In questo caso l'asse slave insegue la posizione e la velocità dell'asse master proporzionalmente al valore di rapporto (se rapporto = 1 lo slave copia esattamente il movimento dell'asse master); l'aggancio (sincronismo) è dipendente alle caratteristiche dinamiche dell'asse slave.



Durante tutto il movimento dell'asse slave (quindi sia in fase di aggancio che a regime) la movimentazione dell'asse viene eseguita (sempre con rampe lineari) utilizzando le formule :

$$\text{PosSlave} = \text{PosSlave}_{t_0} + (\text{PosMaster} - \text{PosMaster}_{t_0}) * \text{FollowRate}$$

$$\text{Vslave} = \text{Vmaster} * \text{FollowRate}$$

La posizione **PosSlave** e la velocità **Vslave** così calcolate sono da considerare come valori "teorici" in quanto occorre verificare se tale richieste rispettano le caratteristiche dinamiche dell'asse (Velocità massima e accelerazione massima). Nel momento in cui la velocità dell'asse master varia, lo slave insegue tale variazione utilizzando il proprio valore di accelerazione. Quando la richiesta di velocità sull'asse slave risulta essere superiore alla massima velocità consentita, il sistema limiterà la velocità richiesta. In tal senso in ogni istante verranno calcolati i due valori di velocità **Vslave_i** e di accelerazione **Aslave_i** con cui muovere l'asse slave. La posizione reale dell'asse slave verrà quindi calcolata in base a tali valori :

$$\text{PosSlave}_{t_{n+1}} = \text{PosSlave}_{t_n} + \text{Vslave}_i + \text{Aslave}_i$$

L'errore tra la Posizione reale dell'asse e quella teorica viene recuperato dall'asse slave durante la movimentazione (ed anche quando la movimentazione dell'asse master risulta essere terminata) andando, quando possibile, a velocità superiori rispetto alla velocità **Vslave** calcolata.

Dissociazione Master/Slave

Con tale istruzione viene rimossa l'associazione tra l'asse master e gli assi slave. **Dopo questa sarà possibile programmare qualsiasi movimento dello slave.**

Sintassi

(XDA)

Definizione/Variazione rapporto di inseguimento

Con tale istruzione viene definito/variato il parametro che determina il rapporto di inseguimento dell'asse master da parte degli assi slave definiti.

Sintassi

(**XDA**, 2, slave1[slave2[.]], rapporto)
(**xda**, 2, slave1[slave2[.]], rapporto)

dove :

slave1...slave8 Sono i nomi degli assi slave (ciascuno designato da un singolo carattere ASCII). Si possono programmare sino ad un massimo di 8 slave.

rapporto E' il rapporto di inseguimento del master da parte dello slave, tale valore deve essere considerato come fattore moltiplicativo della velocità o dello spazio percorso dal master. Con il valore 1.0 lo slave copia fedelmente il moto del master, con valori inferiori ad 1.0 lo riduce, con valori superiori ad 1.0 lo aumenta. Tale valore può essere segnato.

Caratteristiche :

Il comando può essere dato sia quando uno slave è già in fase di inseguimento dell'asse master (in questo caso si effettua uno sgancio dal master e si riparte con una nuova fase di aggancio utilizzando il nuovo parametro di inseguimento) che in fase di non inseguimento (in questo caso si attiva il valore di inseguimento da adottare nella successiva movimentazione).

Utilizzando la sintassi maiuscola si ha un arresto della movimentazione, terminando l'eventuale continuo in corso. La sintassi minuscola, invece, permette l'emissione del comando in continuo; comunque gli assi vengono fermati a velocità nulla e poi immediatamente ripartono. E' possibile evitare l'arresto della movimentazione eseguendo un analogo comando da parte della logica di macchina.

Attivazione inseguimento

Viene immediatamente attivato l'inseguimento dell'asse Master da parte dell'asse slave. La modalità con cui effettuare l'inseguimento viene definita col parametro "*modalità*" all'interno del comando di associazione master/slave.

Sintassi

(XDA, 3, slave1[slave2[...]])
(xda, 3, slave1[slave2[...]])

dove :

slave1...slave8 Sono i nomi degli assi slave (ciascuno designato da un singolo carattere ASCII). Si possono programmare sino ad un massimo di 8 slave.

Caratteristiche :

Utilizzando la sintassi maiuscola si ha un arresto della movimentazione, terminando l'eventuale continuo in corso. La sintassi minuscola, invece, permette l'emissione del comando in continuo; comunque gli assi vengono fermati a velocità nulla e poi immediatamente ripartono. E' possibile evitare l'arresto della movimentazione eseguendo un analogo comando da parte della logica di macchina.

Disattivazione inseguimento

Viene immediatamente disattivato l'inseguimento dell'asse Master da parte dell'asse slave. La modalità di sgancio viene definita col parametro "*modalità*" all'interno del comando di associazione master/slave.

Sintassi

(XDA, 4, slave1[slave2[...]])
(xda, 4, slave1[slave2[...]])

dove :

slave1...slave8 Sono i nomi degli assi slave (ciascuno designato da un singolo carattere ASCII). Si possono programmare sino ad un massimo di 8 slave.

Caratteristiche :

L'asse slave viene comunque mantenuto associato al master, solamente non lo insegue più. In base al parametro "*modalità*" definito all'interno del comando di associazione master/slave si ha:

- 0** Lo slave passa immediatamente dalla velocità attuale alla velocità nulla.
- altri** Lo slave frena con la propria rampa di decelerazione.

Utilizzando la sintassi maiuscola si ha un arresto della movimentazione, terminando l'eventuale continuo in corso. La sintassi minuscola, invece, permette l'emissione del comando in continuo; comunque gli assi vengono fermati a velocità nulla e poi immediatamente ripartono. E' possibile evitare l'arresto della movimentazione eseguendo un analogo comando da parte della logica di macchina.

Esempio :

N10 (XDA,1,X/ZA,3,0.8,0.0)	Attiva master X e slave Z e A
N20 (XDA,3,ZA)	Attiva inseguimento da parte di A e Z
N30 G1X100F2000	
N40 X300	
N50 (xda,4,Z)	Disattiva inseguimento da parte di Z in continuo
N60 X400	
N70 X500	
N80 (xda,4,A)	Disattiva inseguimento da parte di A in continuo
N90 X660	
N100 X700	
N110 (xda,3,ZA)	Riattiva inseguimento da parte di A e Z in continuo
N120 GX0	
N130 (XDA)	Dissocia gli slave Z e A dal Master X
N140 GX	

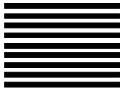
ORIGINI E CONTROLLO DELLE COORDINATE

Le funzioni facenti parte di questa classe eseguono le seguenti operazioni:

CODICE G	FUNZIONE
G04	Sosta a fine blocco
G09	Decelerazione a fine blocco
G16	Definizione del piano di interpolazione
G17	Interpolazione circolare e correzione profilo sul piano XY
G18	Interpolazione circolare e correzione profilo sul piano ZX
G19	Interpolazione circolare e correzione profilo sul piano YZ
G27	Funzionamento continuo con riduzione automatica della velocità sugli spigoli
G28	Funzionamento continuo senza riduzione automatica della velocità sugli spigoli
G29	Funzionamenti punto-a-punto
G70	Programmazione in pollici
G71	Programmazione in millimetri
G79	Programmazione riferita allo zero macchina
G90	Programmazione assoluta
G91	Programmazione incrementale
G92	Pre-impostazione dell'asse
G93	Velocità di avanzamento espressa come inverso del tempo di esecuzione dell'ente
G94	Programmazione della velocità di avanzamento in mm/min o pollici/min
G95	Programmazione della velocità di avanzamento in pollici al giro o millimetri al giro.

NOTA:

I piani specificati in G17, G18 e G19 sono validi se durante la fase di configurazione sono staticamente caratterizzati nell'ordine X, Y, Z.



G17 G18 G19 - Selezione del piano di interpolazione

I seguenti codici G vengono utilizzati per definire il piano di interpolazione come descritto di seguito:

G17 Piano di interpolazione attivo formato dagli assi 1 e 2 (XY).

G18 Piano di interpolazione attivo formato dagli assi 3 e 1 (ZX).

G19 Piano di interpolazione attivo formato dagli assi 2 e 3 (YZ)

Gli assi 1(X), 2 (Y) e 3 (Z) sono i primi tre assi dichiarati in AMP durante la caratterizzazione.

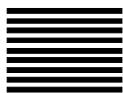
Sintassi

G17

G18

G19

La sintassi per ogni funzione è semplicemente il codice G senza parametri nè altre informazioni.



G16 - Definizione del piano di interpolazione

Il codice G16 definisce l'ascissa e l'ordinata del piano di interpolazione analogamente a G17, G18, G19, ma senza essere vincolato alla programmazione dei primi due assi definiti in caratterizzazione.

Sintassi

G16 *asse1 asse2*

dove:

- | | |
|--------------|--|
| <i>asse1</i> | E' il nome dell'asse delle ascisse (tipicamente X) del piano di interpolazione da definire. Deve essere uno degli assi configurati nel sistema. |
| <i>asse2</i> | E' il nome dell'asse delle ordinate (tipicamente Y) del piano di interpolazione da definire. Deve essere uno degli assi configurati nel sistema. |

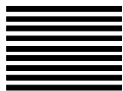
Caratteristiche:

G16, G17, G18, G19 non possono essere usate se sono attivi i seguenti codici G:

- Correzione sul profilo (G41 - G42)
- Cicli fissi standard (G81 - G89)

Esempio:

G16 X A Specifica il piano di interpolazione formato dagli assi X ed A.



G27 G28 G29 - Definizione della modalità dinamica

Le funzioni G di questa classe definiscono il tipo di movimento e di posizionamento dell'asse a fine profilo. Questi codici sono sempre accettati dal controllo.

- G27** Stabilisce un movimento continuo con riduzione automatica della velocità sugli spigoli. Ciò significa che la velocità di uscita dai singoli enti costituenti il profilo viene automaticamente calcolata dal controllo e viene ottimizzata tenendo conto della forma geometrica del profilo stesso. Concorrono al calcolo del valore opportuno della velocità di uscita i parametri DLA, MDA e VEF.
- G28** Stabilisce un movimento continuo senza riduzione automatica della velocità sugli spigoli. In tal caso la velocità di uscita dai singoli enti del profilo è uguale alla velocità programmata.
- G29** Stabilisce un movimento punto-a-punto indipendente dal tipo di traiettoria programmata (G01 - G02 - G03). La velocità di uscita dai singoli enti del profilo è uguale a 0.

Sintassi

G27 [*codici-G*] [*operandi*]

G28 [*codici-G*] [*operandi*]

G29 [*codici-G*] [*operandi*]

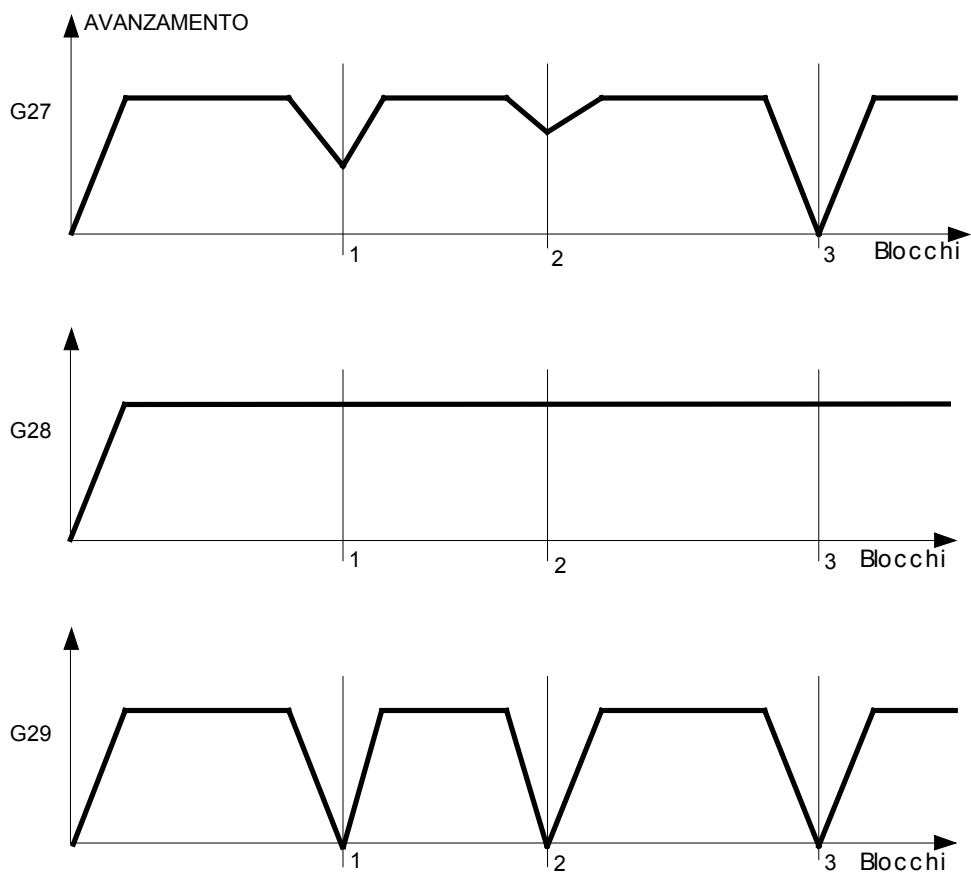
dove:

codici-G Altri codici G compatibili con G27, G28 e G29 (Vedi tabella "Codici G compatibili" nel capitolo 1).

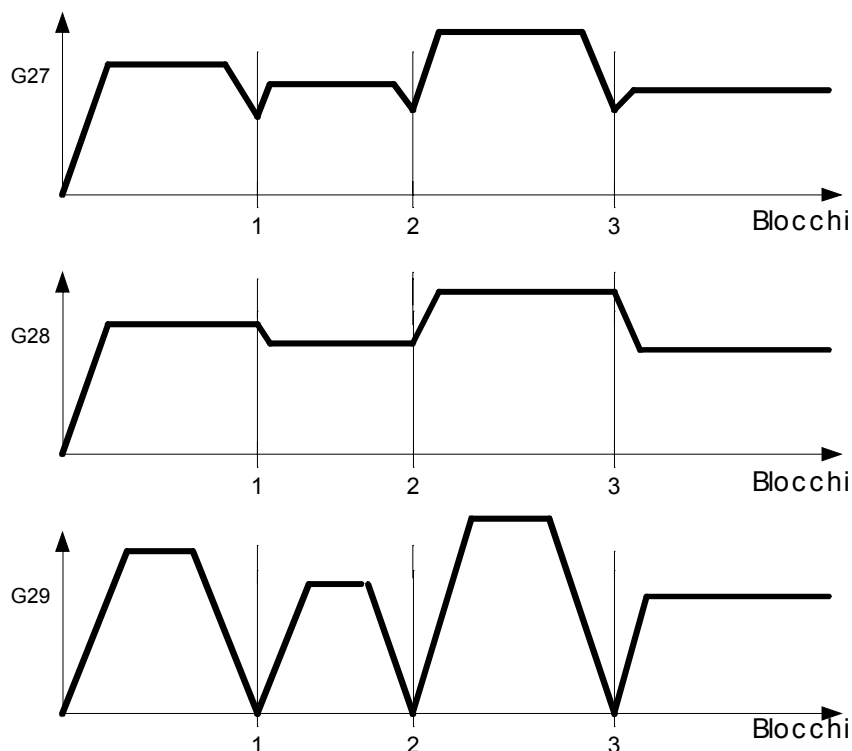
operandi Tutte le possibili classi definite per gli operandi dei blocchi con operatori G.

Caratteristiche:

Il seguente diagramma illustra il funzionamento dei codici G27, G28 e G29 quando la velocità programmata è la stessa per tutto il profilo.



I seguenti diagrammi illustrano il funzionamento dei codici G27, G28, G29, quando la velocità programmata cambia su ogni blocco del profilo.



Il movimento, all'interno di ogni blocco, è diviso in tre fasi :

1. Accelerazione
2. Moto uniforme alla velocità programmata
3. Decelerazione

Le modalità G27 e G28, differiscono solo per il tipo di decelerazione (vedi diagramma precedente). Il modo in cui avviene il posizionamento a velocità di lavoro (G1, G2, G3) è scelto con le funzioni G27, G28, G29, mentre il posizionamento rapido G0 avviene sempre in punto a punto, cioè con decelerazione fino a velocità zero e posizionamento preciso qualunque sia lo stato del sistema (G27, G28, G29).

Le funzioni G27 o G28 definiscono il cosiddetto funzionamento in "continuo" nel quale l'intero profilo, costituito dai vari enti, è considerato come un unico ente continuo e viene analizzato ed eseguito dal controllo come se fosse un solo blocco.

Per questo motivo in continuo non si possono emettere funzioni ausiliarie M, S, T.

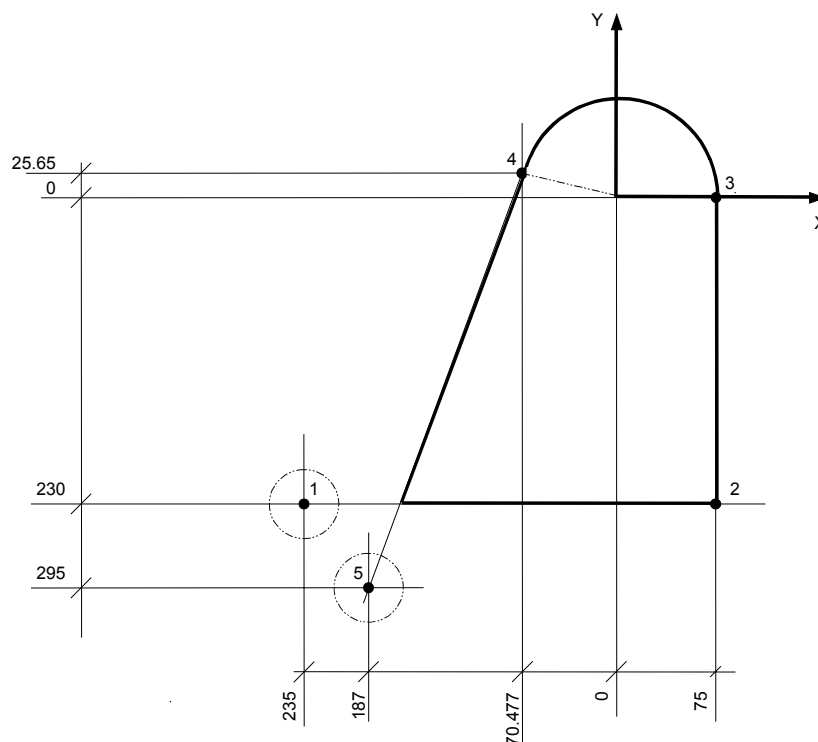
Tale funzionamento può essere chiuso temporaneamente da un movimento in G00 che faccia ancora parte del profilo, in modo tale che eventuali funzioni ausiliarie M, S, e T vengano programmate in un blocco successivo.

NOTA:

Il codice G che è stato configurato su AMP (tipicamente G27) viene automaticamente selezionato all'accensione o in seguito ad un reset.

Esempio:

Il seguente è un esempio di funzionamento in continuo e in punto-a-punto.



Programma 1 (funzionamento in continuo):

```

(UGS,X,-400,100,Y,-400,100)
N9 (DIS,"MILL DIA. 16")
N10 T4.4 M6 S800
1 N11 G X-235 Y-230 M13
  N12 Z-10
2 N13 G27 G1 X75 F500 ;inizio del funzionamento in continuo (G27)
3 N14 Y
4 N15 G3 X-70.477 Y25.651 I J
5 N16 G1 X-187 Y-295
  N17 G Z5 M5 ;chiusura temporanea del continuo (G00)
  N18 (DIS,"MILL DIA. 28");per arresto mandrino, cambio utensile e funzioni S
  N19 T5.5 M6 S1200
  N20 X.. Y.. M13
  N21 Z-..
  N22 G1 X.. Y.. ;Ripresa del funzionamento in continuo
    
```

IMPORTANTE

Se nel blocco N17 fosse programmato un G29, il funzionamento continuo terminerebbe e i successivi spostamenti in G1 - G2 - G3 verrebbero realizzati in modalità punto-a-punto.

Programma 2 (funzionamento punto-a-punto):

```
(UGS,X,-400,100,Y,-400,100)
N9 (DIS, "MILL DIA. 16")
N10 T4.4 M6 S800
1 N11 G29 G X-235 Y-230 M13      ;Inizio funzionamento punto-a-punto
  N12 Z-10
2 N13 G1 X75 F500 M5             ;Arresto mandrino
3 N14 Y S1200 M13               ;Variazione S, rotazione mandrino, refrigerante
4 N15 G3 X-70.477 Y25.651 I J
  N16 DWT=2
5 N17 G1 G4 X-187 Y-295         ;Sosta a fine ente
  N18 G Z5 M5
  N19 (DIS,"MILL DIA. 28")
  N20 T5.5 M6 S1200
  N21 G X.. Y.. M13
  N22 Z-..
  N23 G1 X.. Y..
```

IMPORTANTE

Programmando il funzionamento punto-a-punto con G29 nel blocco N11, è stato possibile programmare le funzioni M ed S all'interno del profilo (blocchi N13 ed N14). La sosta a fine ente (blocco N17), invece, può essere programmata anche in continuo.

DECELERAZIONE AUTOMATICA SUGLI SPIGOLI IN G27

Quando la G27 è attiva, il controllo calcola automaticamente il vettore velocità sugli spigoli (per esempio tra due movimenti successivi), usando un algoritmo che lavora in due fasi distinte.

Durante la prima fase, il vettore velocità viene calcolato applicando una formula che si basa sulle variazioni del profilo.

La variazione del profilo è connessa con l'angolo formato da due movimenti successivi.

Il controllo confronta l'angolo attuale con il valore MDA: se l'angolo è maggiore di quello rappresentato con il valore dell' MDA , il vettore velocità viene posto a zero con un risultato simile al modo G29.

In altre parole, il controllo calcola per quello spigolo , una velocità appropriata a seconda del valore dell'angolo e delle variabili MDA e VEF.

La seconda fase dell'algoritmo, chiamata "**look ahead**", è opzionale e viene abilitata o disabilitata a seconda del valore della variabile DLA.

La fase di "**look ahead**" è un'ottimizzazione della prima fase, poichè il vettore di velocità calcolato viene ulteriormente elaborato tenendo conto anche della distanza totale da percorrere sul profilo in G27 e delle accelerazioni configurate per ogni singolo asse coinvolto nel movimento, allo scopo di ottenere una corretta fermata alla fine del profilo.

IMPORTANTE

La **look ahead**, non gestisce il feedrate override, in questa fase infatti viene considerata una feedrate del 100%; valori maggiori di feedrate potrebbero provocare "SERVO ERROR".



DLA - Decelerazione Look Ahead

Il codice DLA abilita/disabilita il calcolo cosiddetto di "Look Ahead" (Guarda Avanti), in modalità dinamica G27. Il controllo considera i blocchi di movimento (enti) che costituiscono il profilo e che seguono il blocco in esecuzione per ricalcolare la velocità di uscita dai vari enti, tenendo conto del tipo di profilo per prevedere la decelerazione sugli spigoli. Se il profilo comprende improvvise variazioni di traiettoria e la lunghezza degli enti non è sufficiente a garantire una corretta decelerazione, è fondamentale avere la possibilità di prevedere queste situazioni in anticipo e dinamicamente, per poter regolare la velocità. Il numero di blocchi in movimento successivi a quello in esecuzione su cui è eseguito di volta in volta il calcolo è configurabile in caratterizzazione e può variare da 2 a 64.

Sintassi

DLA=*valore*

dove:

valore E' un valore numerico che può essere:

- 0** per disabilitare l'algoritmo di "Look Ahead"
- 1** per abilitare l'algoritmo di "Look Ahead".

NOTA:

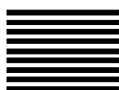
Settando DLA=1, il tempo di elaborazione di un singolo blocco (block-time) aumenta perchè il controllo deve eseguire un numero maggiore di calcoli e di controlli; si ha però come risultato finale una migliore precisione sul profilo.

Settare DLA=0, risulta conveniente nel caso in cui è chiaro che la FEED programmata e la distanza totale da percorrere in G27 sono tali da garantire una corretta fermata alla fine del profilo.

Con DLA=0 il controllo considera solamente le variazioni di profilo sugli spigoli.

Caratteristiche:

Il valore di default di questa variabile è caratterizzabile in AMP.



DYM - Modalità dinamica

Il codice DYM definisce il tipo di algoritmo da utilizzare per il calcolo della velocità tra un ente ed il successivo con il modo G27 attivo.

Sintassi

DYM = *valore*

dove:

valore E' un valore numerico che può essere:

- 0** per usare la formula standard della Serie 10
- 1** per usare la formula standard della Serie 8600
- 2** per usare la formula alternativa della Serie 10

L'algoritmo standard della Serie 10 si basa su formule matematiche esatte che ipotizzano una risposta lineare della macchina e sul fatto che i parametri dinamici configurati siano sempre applicabili in tutte le condizioni di lavoro.

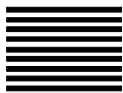
L'algoritmo già presente nella serie 8600 utilizza formule approssimate, effettuando di fatto una limitazione più drastica della movimentazione.

La formula alternativa tiene conto delle componenti dinamiche (accelerazioni) sopportabili dagli assi nel momento in cui si passa da un ente a quello successivo. In tal senso ricalcola la velocità con cui terminare il primo ente e passare al successivo senza sottoporre la macchina ad eccessive sollecitazioni

Si consiglia di verificare sperimentalmente le conseguenze sulla meccanica e sulla lavorazione dei diversi algoritmi, e di adottare quello più appropriato.

Caratteristiche:

Il valore di default di questa variabile è caratterizzabile in AMP.



MDA - Calcolo del massimo angolo di decelerazione

Il codice MDA definisce la massima deviazione angolare dell'asse in modalità G27. Il valore selezionato (fra 0 e 180 gradi) definisce l'angolo tra due enti consecutivi, oltre il quale il controllo si comporta come se fosse in modalità punto a punto.

Sintassi

MDA=*valore*

dove:

valore

È un dato numerico con le seguenti caratteristiche:

- angolo compreso tra 0° e 180° se DYM=0
- numero compreso tra 0 e 2 se DYM=1

In entrambi i casi esso rappresenta la massima deviazione tra due enti consecutivi, oltre la quale, in G27, viene forzata la fermata sul punto finale.

Nel caso DYM=1 il valore va calcolato come:

seno dell'angolo massimo per deviazioni $\leq 90^\circ$

$1 + [\text{seno}(\text{angolo} - 90^\circ)]$ per deviazioni $> 90^\circ$ e $\leq 180^\circ$.

- non significativo se DYM=2

Caratteristiche:

Il valore di default è MDA=90°. E' possibile configurare il valore di MDA in caratterizzazione e modificarlo tramite apposita finestra di introduzione dati o da part program.

IMPORTANTE

Quando la variazione di direzione è maggiore dell'angolo definito dal valore di MDA, il sistema decelera l'asse fino a velocità zero.

Poichè la decelerazione sugli spigoli calcolata dal sistema dipende dall'angolo formato dagli enti e dal valore di MDA e della variabile VEF, è possibile ottenere diversi valori di riduzione della velocità cambiando il valore di MDA.

Notare che piccoli valori di MDA comportano una forte decelerazione sugli spigoli.

Il RESET del sistema ripristina il valore di MDA configurato con l' AMP.

Esempi:

DYM=0

MDA=90°

MDA=180°

DYM=1

MDA=1

MDA=2



VEF - Fattore di velocità

Il codice VEF definisce un fattore per determinare il calcolo della velocità sugli spigoli con il modo G27 attivo. Infatti la velocità calcolata a seconda del valore di MDA può essere aumentata o diminuita cambiando il valore di VEF.

Piccoli valori di VEF comportano una più alta riduzione della velocità sugli spigoli.

Sintassi:

VEF=*valore*

dove:

valore è un valore numerico con le seguenti caratteristiche:

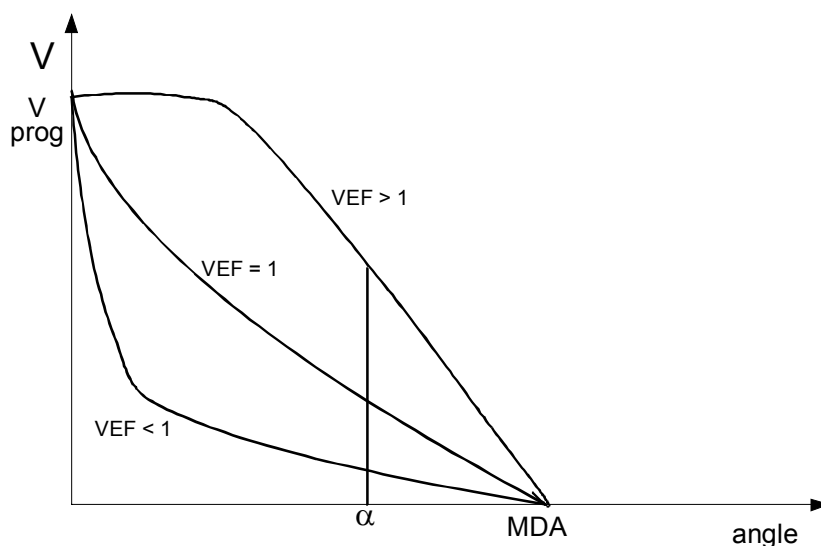
- numero da 0.1 a 8 se DYM = 0 Il valore di default è 0.8.
- numero da 0 a 99999 se DYM = 1 Il valore di default è 0.8
- numero da 0 a 99999 se DYM = 2 Il valore di default è 0.8

Caratteristiche:

Le caratteristiche di calcolo della velocità variano a seconda del valore della variabile DYM.

DYM = 0

Il seguente diagramma illustra le differenti decelerazioni calcolate dal sistema cambiando il valore di VEF mantenendo costante il valore di MDA.



dove:

V è la velocità sull'angolo calcolata dal sistema

α è l'angolo tra due movimenti successivi

V_{prog} è la feed programmata

DYM = 1

Il codice VEF definisce il massimo errore di forma ammesso sullo spigolo. Se il suo valore è 0, il sistema provoca una decelerazione degli assi fino a zero al termine di ogni blocco.

DYM = 2

Il valore VEF definisce il massimo “gradino” di velocità per l’asse nel passaggio tra un ente ed il successivo: ad esempio se VEF= 0.8 all’asse sarà permesso un gradino di velocità pari a $1 + 0.8$ dell’accelerazione di lavoro configurato.

Il sistema calcolerà la velocità sugli spigoli in base a tutti gli assi coinvolti nel movimento; ogni asse avrà una velocità diversa ed il sistema adotterà la minima tra queste.

IMPORTANTE

Il RESET di sistema ripristina il valore VFF configurato in AMP

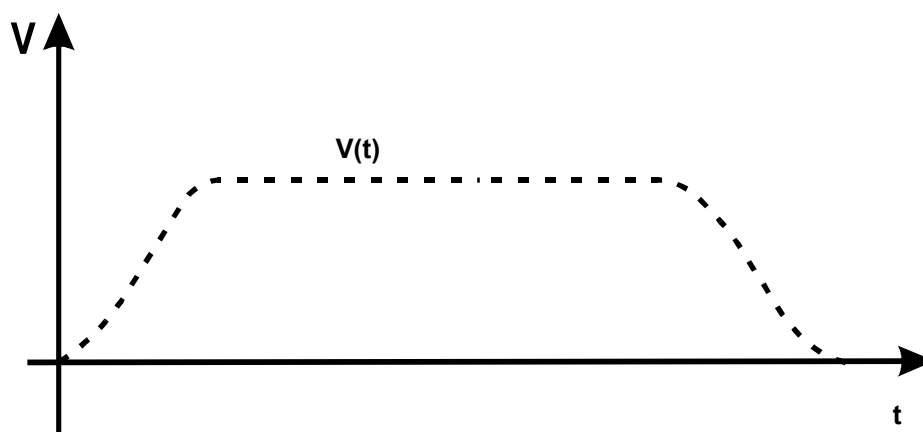


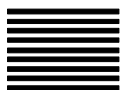
Jerk Limitation

Facendo riferimento ai diagrammi di velocità riportati nei paragrafi precedenti si nota la continuità della funzione velocità $V(t)$ mentre la funzione accelerazione $a(t)$ risulta avere un andamento a “gradini”. Questo, in funzione delle caratteristiche della macchina e del tipo di lavorazione, può causare finiture del pezzo non buone.

Tale problema può essere risolto utilizzando una funzione di accelerazione $a(t)$ che comporti una continuità nel suo andamento.

Con la “Jerk Limitation” si intende limitare le variazioni di accelerazione, così da controllarne il valore massimo, che si traduce in una maggiore “morbidezza” dei movimenti ed una conseguente migliore finitura superficiale.





MOV - Modalità di Movimentazione

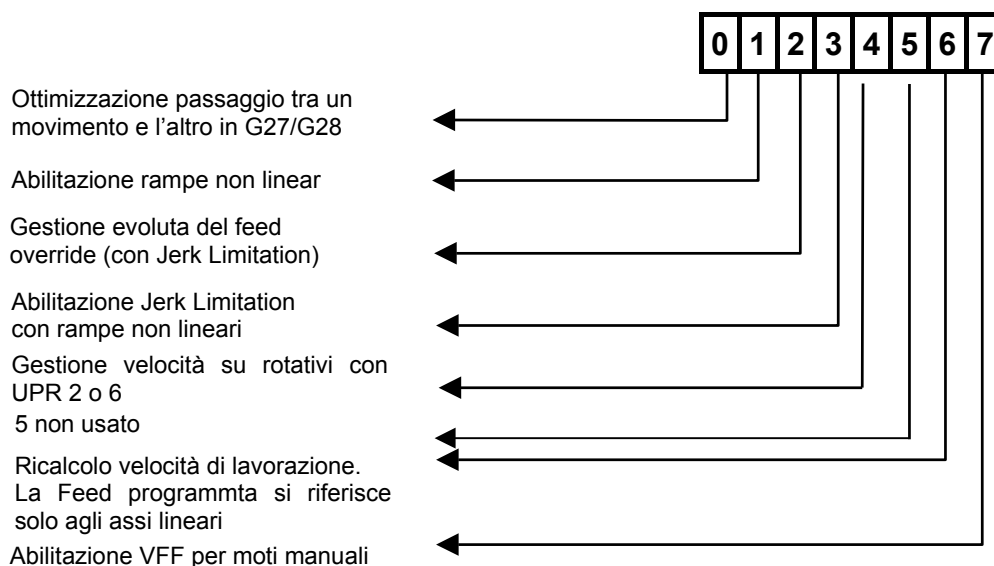
Il codice MOV serve per definire alcune caratteristiche di gestione dei movimenti.

Sintassi:

MOV = *valore*

dove:

valore Caratteristiche di movimentazione da abilitare.
Il valore da specificare si ricava sommando i pesi decimali dei bit corrispondenti a ciascuna delle prestazioni desiderate.



Significato dei bit 0 ÷ 3 :

I valori ammessi per i primi 4 bit sono i seguenti:

- 0** rampe non lineari disabilitate e feedrate override attivo
- 1** ottimizzazione del passaggio tra un ente di movimento ed il successivo durante la movimentazione in G27 e G28. Tale operazione è da utilizzare specialmente per part program che definiscono un profilo "per punti" in quanto consente una migliore gestione di curvatura del profilo (specie se descritta con tratti molto corti) evitando che gli assi sfuggano alla traiettoria.
Essa è attiva di default nel caso di abilitazione delle rampe non lineari (bit 1, 2 e 3 settati)
- 2** rampe non lineari abilitate e feedrate override gestito come variazione del tempo di rampa (di fatto variazione delle accelerazioni)

- 6** rampe non lineari abilitate e feedrate override gestito come distorsione della forma delle rampe (di fatto mantenimento quasi completo delle accelerazioni)
- 8** Jerk Limitation con rampe non lineari abilitate e feedrate override gestito come variazione del tempo di rampa (di fatto variazione delle accelerazioni)
- 12** Jerk Limitation con rampe non lineari abilitate e feedrate override gestito come distorsione della forma delle rampe (di fatto mantenimento quasi completo delle accelerazioni)

Caratteristiche:

Si consiglia di utilizzare il valore **8** per macchine che tipicamente lavorano al 100% della velocità programmata. Il valore **6 (12)** è raccomandato in fase di attrezzaggio quando è utile una risposta pronta al feedrate override.

Le rampe lineari con o senza Jerk Limitation funzionano sia su movimenti in punto a punto (G29) sia su movimenti in continuo (G27, G28). L'attivazione e la disattivazione di questo algoritmo non può però essere fatta all'interno del continuo.

Il valore di default di questa variabile è 0. È possibile comunque configurare il valore di MOV in AMP. Il RESET ripristina il valore di default.

IMPORTANTE

Quando le rampe non lineari vengono attivate con MOV=6 (12) il feedrate override è attivo solo tra 0 e 100% ed i valori superiori al 100% non sono raggiungibili.

Significato del bit 4:

Bit 4 (valore 16) : abilita il tipo di gestione delle velocità sui rotativi da adottare con UPR 2 o 6 attivi: se bit a 0 mantiene la gestione standard, se bit a 1 gli assi rotativi utilizzano diagrammi di velocità controllati.

Nel 1° caso il posizionamento degli assi rotativi è fedele alla trasformazione richiesta dall'UPR, ma i particolari algoritmi di calcolo usati possono generare dei notevoli rallentamenti.

Nel 2° caso tali rallentamenti sono eliminati, ma il posizionamento degli assi rotativi potrebbe non essere fedele alla trasformazione richiesta dall'UPR, in questi casi gli assi risultano essere in ritardo.

Significato dei bit 6 ÷ 7 :

I valori dei bit 6 e 7 hanno il seguente significato :

Bit 6 (valore 64) : definisce che la Feed programmata in un blocco di movimento è relativa agli assi lineari. L'eventuale programmazione di un asse rotativo , in aggiunta a quelli lineari, provoca un ricalcolo automatico della velocità in modo che la stessa lungo gli assi lineari rimanga invariata. Il ricalcolo della velocità viene applicata solo se in un movimento sono presenti sia gli assi lineari che gli assi rotativi. Non viene applicata nel caso di moti circolari (G2/G3), in cui un asse rotativo fa parte del piano d'interpolazione.

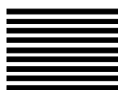
Bit 7 (valore 128): abilita l'algoritmo di VFF anche per i moti in manuale che normalmente vengono eseguiti utilizzando solo l'algoritmo di "errore di inseguimento".

Il valore derivante dal settaggio di questi bit va sommato a quello dato ai bit 0÷3 .

Esempio: se si vuole utilizzare la Jerk Limitation con rampe non lineari abilitate e feedrate gestito come variazione del tempo di rampa e contemporaneamente avere il VFF abilitato per i moti normali, la variabile MOV deve assumere il valore $128 + 8 = 136$.

IMPORTANTE

Tale prestazione **NON deve** essere attivata se la programmazione della velocità viene effettuata in G93 o con t. In tal caso la velocità di lavorazione risultante non sarà corretta.



JRK - Jerk Constant

Il codice JRK definisce la modalità di gestione delle accelerazioni durante le rampe non lineari. Viene utilizzato nel caso di MOV=2 o 6. Nel caso di MOV=8 o 12 il valore JRK viene calcolato automaticamente dal sistema in base alle caratteristiche dinamiche configurate per ogni asse in AMP; quindi il JRK usato è funzione degli assi coinvolti nel singolo movimento.

Sintassi:

JRK = *valore*

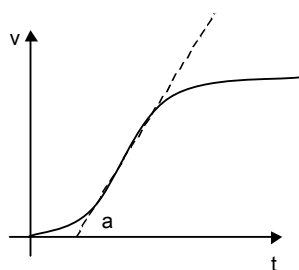
dove:

valore E' un valore numerico maggiore di 0.5 che serve per definire la modalità di gestione delle accelerazioni all'interno di una rampa non lineare

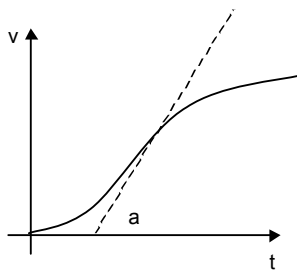
Caratteristiche:

Il valore di default è 1. E' possibile comunque configurare il valore di JRK in AMP. Il RESET ripristina il valore di default.

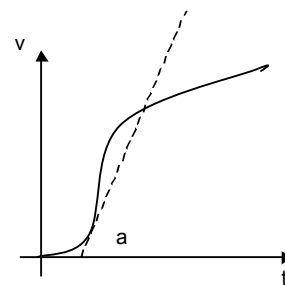
Ponendo JRK = 1, la rampa di accelerazione mantiene i valori configurati per le accelerazioni degli assi. Con valore inferiore ad 1 si diminuisce l'accelerazione, con valore superiore lo si aumenta.



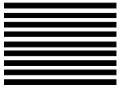
JRK=1



JRK<1



JRK>1



JRS - Jerk Smooth Constant

Il codice JRS definisce il valore di soglia per la limitazione della velocità con la Jerk Limitation attivata.

Sintassi:

JRS = *valore*

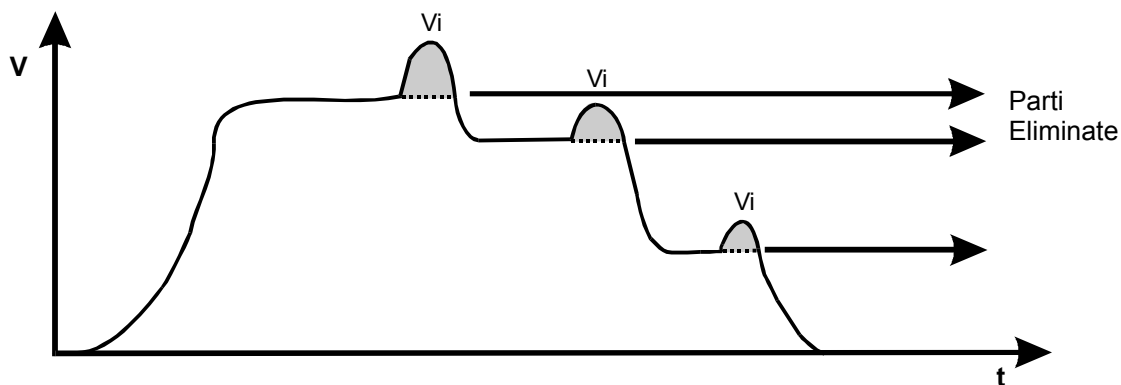
dove:

valore E' un valore numerico maggiore di 0, che serve a definire una soglia di velocità al di sotto della quale limitare la velocità stessa nel caso in cui non si riesca a raggiungere la velocità programmata.

Caratteristiche:

Una caratteristica dell'algoritmo di lookahead per la Jerk Limitation è quella di evitare accelerazioni e decelerazioni continue che comporterebbero oscillazioni sul movimento degli assi. Ciò potrebbe accadere nel caso in cui gli assi programmati non consentano il raggiungimento della velocità impostata $V(i)$.

A tale scopo il diagramma di velocità viene "tagliato" così come illustrato nella seguente figura:



Per evitare che la parte “tagliata” sia troppo grande generando un consistente aumento dei tempi di lavorazione, viene definita una soglia al di sotto della quale limitare la velocità.

Tale soglia è calcolata in funzione del parametro JRS secondo la seguente regola:

dette

Vmax il valore massimo di velocità calcolato dall’algoritmo sui movimenti presi in esame:

Vmin il valore minimo di velocità sui movimenti presi in esame;

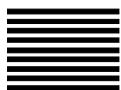
Il sistema confronta se

$$| V_{\max} - V_{\min} | < V_{\min} \cdot JRS$$

Se sì, la Vmax viene posta uguale a Vmin, altrimenti viene ricalcolata in modo tale da permettere un tratto a velocità costante nella parte superiore della rampa.

IMPORTANTE

Il valore di default è 1. Il RESET ripristina il valore di default.



ODH - Online Debug Help

Questa variabile di sistema permette di verificare se esistono, nel part program, delle movimentazioni in modalità continuo (G28) con velocità programmate sul profilo troppo alte, con conseguenti errori di traiettoria sul profilo da lavorare.

Sintassi:

ODH = *valore*

Caratteristiche:

Per ottenere risultati attendibili dall'utilizzo di ODH, occorre azzerare la variabile stessa prima di utilizzarla nel part program; ODH deve cioè essere forzata a zero prima dei blocchi da testare. L'utilizzo di questa variabile avviene solitamente in fase di messa a punto del Part Program : il sistema modifica la variabile ODH quando rileva condizioni critiche.

Tali informazioni sono codificate a bit con il seguente significato :

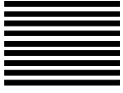
- Bit 0:** indica che la coda degli enti processati in continuo è troppo corta. Si verifica quando in G28 (e in alcuni casi in G27) , la velocità di lavorazione è troppo elevata rispetto al valore configurato in AMP come numero di blocchi da processare durante il movimento in continuo. Le soluzioni possibili sono di diminuire la velocità o di incrementare la dimensione della coda enti di precalcolo.
- Bit 1:** indica velocità troppo elevata che si verifica quando nel profilo esistono degli enti troppo corti che vengono saltati poiché percorsi a velocità troppo elevata. In tal caso si ha una deformazione del profilo lavorato .Le soluzioni possibili sono di ridurre la velocità , oppure di ricalcolare i punti sul programma .Per evitare deformazioni del profilo si consiglia di attivare nella variabile MOV il bit 0 (MOV = 1).
- Bit 2:** si verifica quando viene richiesta in G26 un'accelerazione troppo grande rispetto quella configurata , nel passaggio da un ente di movimento al successivo.
- Bit 3:** si verifica quando il sistema ha forzato una frenata per evitare di deformare il profilo nel caso in cui non riesca a calcolare completamente le rampe di accelerazione/decelerazione per gli enti di movimento analizzati , durante la lavorazione con rampe non lineari attive.

Esempio:

```

.....
.....
ODH=0                ;azzeramento variabile per verifica velocità
G1G28F1000          ;inizio movimento modalità continuo
.....
.....
.....
.....                ;movimenti in modalità continuo
.....
.....
.....
G29M5                ;fine dei movimenti in modalità continuo
(GTO,KO,ODH=1)        ;salta se ODH=1
(DIS;"VELOCITA' ENTRO I LIMITI DEL SISTEMA")
(GTO,CONTINUA)
.....
"KO"
(DIS;"PROBLEMI DI VELOCITA' TROPPO ALTA")
.....
.....
"CONTINUA"
.....
.....

```



MBA – MultiBlock retrace Auxiliary functions

La variabile MBA (Boolean) permette di abilitare/disabilitare l'emissione delle funzioni ausiliarie (funzioni M) durante l'operazione di Retrace.

Sintassi:

MBA = *valore*

Caratteristiche:

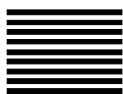
L'emissione delle funzioni ausiliarie viene effettuata solamente durante la fase di retrace in avanti (vedere manuale d'uso)

valore = 0 Disabilita l'emissione

valore = 1 Abilita l'emissione

IMPORTANTE

Dopo il reset, la variabile viene ripristinata col valore configurato in AMP



REM – Ritorno automatico sul profilo a fine movimento

La variabile REM (Boolean) determina la modalità di ritorno sul profilo (Jog return).

Se la variabile è 0 il punto di ritorno è il punto in cui si è entrati in hold (esecuzione standard).

Se la variabile è 1 il punto di ritorno è il punto finale del movimento all'interno del quale si è entrati in hold. Il ritorno è eseguito in automatico, senza movimenti in manuale (JOG).

Sintassi:

REM = *valore*

valore = 0 Ritorno sul punto di ingresso in hold

valore = 1 Ritorno automatico a fine movimento

IMPORTANTE

Il reset non modifica il valore della variabile



IPB (DTL) - In Position Band

Il comando IPB consente di definire la tolleranza di posizionamento degli assi.

Sintassi

(IPB, asse1,[asse2,. . .,asse9])

dove:

asse1 . . . asse9 E' un campo formato da un nome asse e da un valore. Il sistema consente di programmare fino a nove assi. Gli assi devono essere fra quelli configurati nel sistema. La tolleranza di posizionamento va programmata nell'unità di misura (G70/G71) attiva in quel momento.

Caratteristiche:

Questo valore specifica la tolleranza tra la posizione reale dell'asse a fine movimento e il punto di arrivo programmato. In modalità di posizionamento G00, l'asse deve essere entro tale tolleranza prima che venga eseguito il blocco successivo.

Programmando un valore zero nel comando IPB, il controllo usa la tolleranza di posizionamento di default specificata durante la configurazione di sistema.

Se nel comando IPB non si programma un asse specifico, il controllo applica la tolleranza di posizionamento correntemente attiva per quell'asse.

Non si può specificare due volte lo stesso asse in un comando IPB.

Il comando IPB causa un errore quando sono attive le seguenti condizioni:

- Correzione del profilo (G41-G42)
- Modo continuo (G27-G28)
- Il valore introdotto viene arrotondato nei calcoli interni a meno della precisione (< 1 digit). L'errore generato è l'NC 133.

Bisogna quindi disattivare queste condizioni prima di utilizzare il comando IPB.

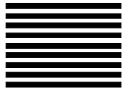
IMPORTANT

Il valore programmato con IPB sostituisce quello configurato in AMP. Il RESET non ripristina il valore configurato.

Esempio:

(IPB, X 0.1, Y 0.05)

Questo blocco specifica una tolleranza di posizionamento di 0,1 per l'asse X e di 0,05 per l'asse Y.



G70 G71 - Unità di misura

I codici G70 e G71 definiscono l'unità di misura usata dal controllo.

G70 Imposta l'unità di misura in pollici

G71 Imposta l'unità di misura in millimetri

Sintassi

G70 [*codici-G*] [*operandi*]

G71 [*codici-G*] [*operandi*]

dove:

codici-G Altri codici G compatibili con G70 e G71 (Vedi tabella "Codici G compatibili" nel capitolo 1).

operandi Tutti gli operandi e i codici ammessi nei blocchi di funzioni G.

Se non viene programmato G70 o G71, l'unità di misura di default sarà quella dichiarata nella configurazione di sistema.

Commutando da G71 a G70 o da G70 a G71, tutte le informazioni relative alla posizione e alla velocità di avanzamento vengono automaticamente convertite all'unità appropriata.

IMPORTANTE

Commutando da G71 a G70 o viceversa, le tabelle dei correttori e delle origini non vengono convertite automaticamente nelle unità alternative.



G90 G91 G79 - Programmazione assoluta, incrementale o riferita allo zero macchina

G90	Definisce la programmazione assoluta; gli spostamenti sono riferiti all'origine corrente.
G91	Definisce la programmazione incrementale; gli spostamenti sono riferiti alla posizione raggiunta con lo spostamento precedente.
G79	Definisce la programmazione riferita allo zero macchina. E' valido solo nel blocco in cui viene programmato.

Sintassi

G90 [*codici-G*] [*operandi*]

G91 [*codici-G*] [*operandi*]

G79 [*codici-G*] [*operandi*]

dove:

codici-G Altri codici G compatibili con G90, G91 e G79 (Vedi tabella "Codici G compatibili" nel capitolo 1).

operandi Tutti gli operandi e i codici che possono essere utilizzati nei blocchi di funzioni G.

Caratteristiche:

Se nessuno di questi codici viene programmato, la modalità di programmazione di default è il sistema assoluto (G90) riferito alle origini programmate.

G90 e G91 sono modali. G79 non è modale: dopo aver programmato un blocco con G79, il controllo riattiva il sistema di programmazione del blocco precedente.

Tramite l'utilizzo dei caratteri >> è possibile anche la programmazione mista incrementale/assoluta nello stesso blocco.

I caratteri >> posti prima del valore numerico di un operando, indicano che esso va considerato come valore incrementale ed è valido solo per quell'operando. I caratteri >> hanno significato solo se è attiva la programmazione assoluta G90. Essi possono essere utilizzati per tutti gli operandi su cui è possibile utilizzare la funzione G91.

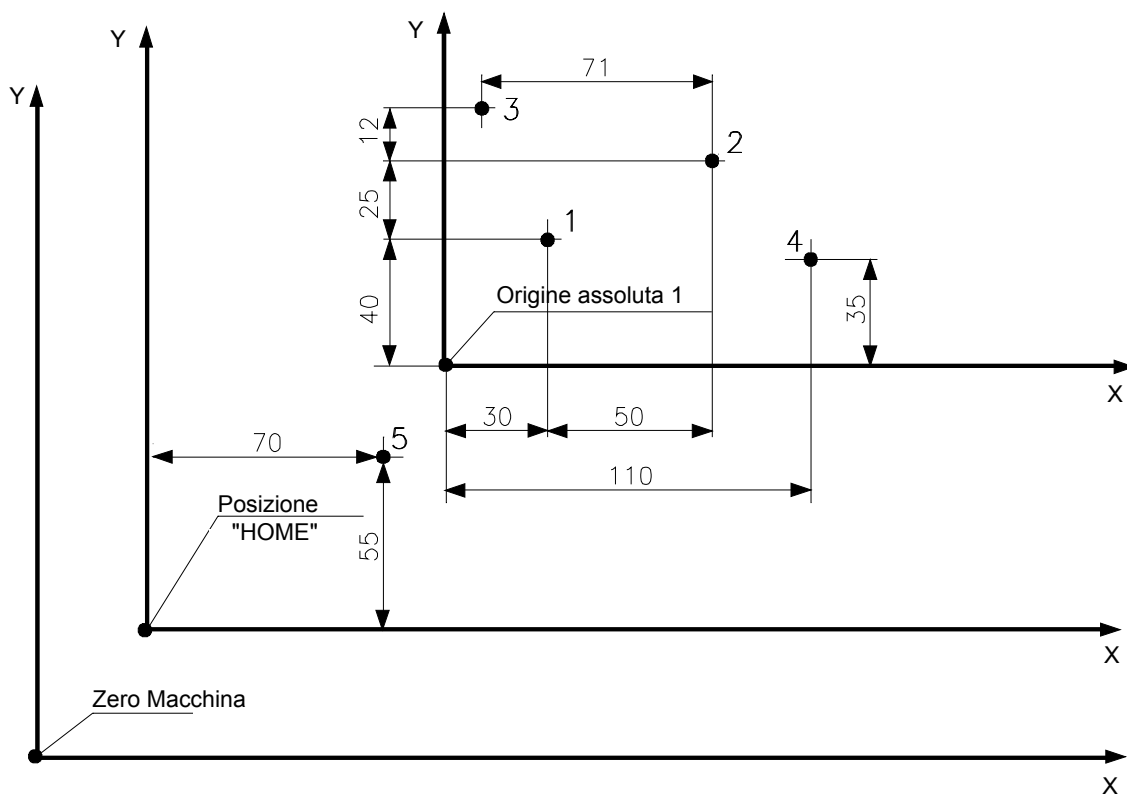
Esempio:

G90 G1 X + 80 Y >> + 35 Z-70

Il valore associato ad Y va considerato come incremento.

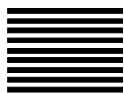
Esempio:

Questo esempio illustra i tre differenti sistemi: assoluto, incrementale e riferito alla posizione "HOME".



Programma:

```
(UGS,1,X,-50,100,Y, 50,100)
(UAO,1) ;Attiva l'origine assoluta 1
N1 G X Y ;X e Y posizionate sull'origine assoluta 1(si assume modalit  di default G90)
N2 X30 Y40 ;X e Y posizionate sul punto 1
N3 G91 X50 Y25 ;posizionamento in incrementale sul punto 2 ( X+50, Y+25 dal punto 1)
N4 X-71 Y12 ;posizionamento in incrementale sul punto 3 ( X-71, Y+12 dal punto 2)
N5 G90 X110 Y35 ;posizionamento in assoluto sul punto 4 ( X+110, Y+35 dall'origine)
N6 G79 X70 Y55 ;posizionamento riferito alla posizione "HOME" sul punto 5 ( X+70, Y+55 dalla posizione "HOME")
```



G92 G98 G99 - Presetting dell'asse

G92 e G98 sono un metodo alternativo per introdurre l'offset, o correzione, di un asse. Poiché G92/G98 definiscono la posizione di riferimento, devono essere programmate da sole nel blocco. La differenza tra la posizione attuale e la nuova viene memorizzata in un registro di offset specifico per G92/G98. In tal modo, quando viene introdotto un offset G92/G98, altri offset attivi, come i correttori utensile e le origini, non vengono persi. Il codice G99 cancella il codice G92/G98.

Sintassi

G92 assi

G98 assi

dove:

assi Sono campi del tipo nome asse-valore. Possono essere specificati al massimo 6 assi.

Caratteristiche:

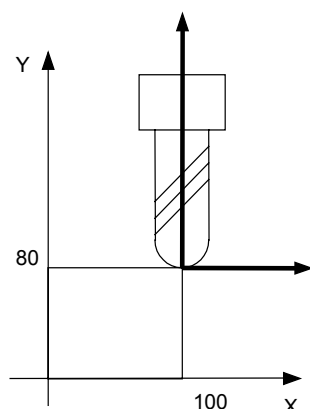
La G98 opera esattamente come la G92, ma mentre la G92 non considera il MIRROR applicato sugli assi programmati, la G98 ne tiene conto.

I codici G92 e G98 vengono annullate dalle seguenti funzioni:

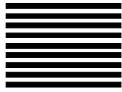
- G99
- M2
- M30
- reset di sistema
- logica di macchina

L'active reset non influenza un offset introdotto con G92 o G98. L'offset G92 o G98 sposta l'origine di un part program, senza provocare lo spostamento dell'asse. Quando viene inserito il valore dell'asse in un blocco G92 o G98, esso diventa la posizione corrente dell'asse.

Esempio:



```
.....
.....
G0 X100 Y80
G92 X0 Y0
.....
.....
```



G04 G09 - Attributi della modalità dinamica

A questa classe appartengono due codici:

G04 Sosta a fine blocco

G09 Decelerazione a fine blocco

Sintassi

G04 [*codici-G*] [*operandi*]

G09 [*codici-G*] [*operandi*]

dove:

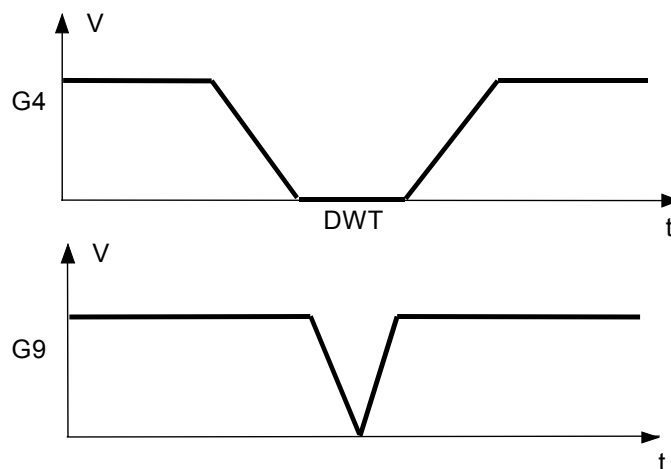
codici-G Altri codici G compatibili con G4, G9 (Vedi tabella "Codici G compatibili" nel capitolo 1).

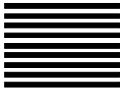
operandi Tutti gli operandi e i codici ammissibili nei blocchi di funzioni G.

Caratteristiche:

G04 provoca una sosta a fine ente. Il tempo di sosta si programma con un comando DWT in un blocco che precede G04; nel caso in cui il blocco DWT non venga programmato viene assunto come tempo di sosta quello caratterizzato. G04 è valido solo nel blocco in cui è programmato. Il valore impostato in DWT viene espresso in secondi (G94 o G95 con G0 attivo) o in giri (G95).

G09 forza la velocità a zero al termine del blocco in cui viene programmato, ma non varia lo stato della lavorazione in corso su un profilo. Se il controllo è in modalità punto-a-punto (G29) o se è programmato in un blocco alla fine di un profilo, G09 non provoca alcuna modifica nello stato del controllo. G09 ha validità solo nel blocco in cui viene programmato.





t - Tempo di esecuzione blocco

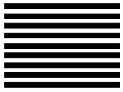
Programmando la funzione **t** alla fine del blocco è possibile stabilire il tempo di esecuzione del blocco in entrambe le funzioni G93 e G94.

Esempio:

G1 X10 Y1 t6

Caratteristiche:

La funzione **t** è valida solo nel blocco in cui è stata programmata. Il tempo viene calcolato in secondi ed il controllo calcola automaticamente l'avanzamento che verrà usato per eseguire il movimento degli assi presenti nel blocco.



DWT (TMR) - Tempo di sosta

Il comando DWT consente di programmare un tempo di sosta da eseguire a fine blocco. Definisce il tempo di sosta utilizzato per G04 e nei blocchi di ciclo fisso.

Il comando DWT può essere programmato in un blocco qualsiasi di part program, ma prima di qualsiasi G04 o ciclo fisso al quale fa riferimento.

Sintassi

DWT = valore

dove:

valore E' un valore espresso in secondi o giri. Può essere programmato direttamente con un numero decimale o indirettamente con i parametri E.

Esempio:

DWT = 12.5	assegna un tempo di sosta di 12.5 secondi
E32 = 13.4	assegna il valore 13.4 alla variabile E32
DWT = E32	assegna un tempo di sosta di 13.4 secondi



G93 - Velocità di avanzamento V/D

La funzione G93 definisce la velocità di avanzamento degli assi (F) espressa come inverso del tempo in minuti necessario per eseguire l'ente

$$F = \frac{1}{T}$$

Interpolazione Lineare:

$$F = \frac{\text{Velocità di avanzamento}}{\text{Distanza}}$$

Interpolazione Circolare:

$$F = \frac{\text{Velocità di avanzamento}}{\text{Arco}}$$

dove:

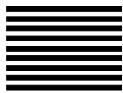
<i>Velocità di avanzamento</i>	velocità lineare o circolare espressa in mm/min (G71) o pollici/min (G70)
<i>Distanza</i>	distanza vettoriale del moto lineare programmato in mm o pollici
<i>Arco</i>	sviluppo dell'arco programmato in mm o pollici.

IMPORTANTE

Con G93 attiva, la funzione F è valida solo nel blocco in cui è stata programmata.

Esempio:

G93 G1 X... Y... F...
 X... Y... F...



VFF - Velocity Feed Forward

Questo comando abilita e disabilita il VFF. Il VFF entra in gioco nell'asservimento assi consentendo un controllo sulla velocità oltre che sulla posizione.

Sintassi

VFF=*valore*

dove:

valore E' un valore che può essere:

- 0** disabilita VFF - gli assi sono controllati solo in posizione, cioè tenendo conto solo dell'errore di inseguimento rispetto alla posizione teorica.
- 1** abilita VFF - gli assi sono controllati anche in velocità

NOTA:

Il valore di default di VFF è 1 e viene configurato al momento della caratterizzazione in AMP.

MODIFICA DEL SISTEMA DI RIFERIMENTO DELL'ASSE

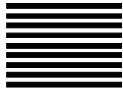
I comandi appartenenti a questa classe consentono di modificare il sistema di riferimento cartesiano rispetto al quale è stato programmato un profilo.

COMANDO	FUNZIONE
---------	----------

SCF	Fattori di scala
MIR	Lavorazione in mirror
ROT	Rotazione del piano
UAO	Uso di origini assolute
UTO	Uso di origini temporanee
UIO	Uso di origini incrementali
RQO	Riqualifica delle origini

Se attive queste funzioni vengono eseguite nel seguente ordine:

SCF - G70/G71 - MIR - ROT - ORIGINI.



SCF - Fattori di scala

Il comando SCF assegna un fattore di scala alle quote programmate. Il controllo applica i fattori di scala agli assi specificati nel comando SCF.

Sintassi

(SCF [asse1, . . . , asse9])

oppure:

(SCF [,valore])

dove:

valore Definisce il fattore di scala da applicare. Si può programmare direttamente usando un numero decimale o indirettamente usando un parametro E.

asse1 . . . asse9 Nome dell'asse e valore. Deve essere uno degli assi configurati sul sistema. Il controllo applica il fattore di scala all'asse specificato.

Caratteristiche:

Nel comando SCF è possibile specificare fino a nove assi. Il controllo disabilita i fattori di scala per assi che non siano specificati nel comando. Se programmato senza un fattore di scala e senza assi, SCF annulla i fattori di scala per tutti gli assi.

Esempio:

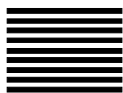
.
.
.
(SCF, 3) Applica un fattore di scala 3 alle quote di tutti gli assi configurati. In altre parole, tutte le quote programmate per gli assi vengono moltiplicate per 3.

.
.
(SCF, X2. 5, Y3) Applica un fattore di scala di 2,5 alle quote programmate dell'asse X e di 3 a quelle dell'asse Y e disattiva i fattori di scala per tutti gli altri assi.

(SCF) Disattiva i fattori di scala di tutti gli assi.

IMPORTANTE

Il RESET del sistema disabilita il fattore di scala per tutti gli assi.



MIR - Uso della lavorazione speculare

Abilita l'inversione speculare della figura per gli assi dichiarati.

Sintassi

(MIR, [asse1, . . . , asse9])

dove:

asse1. . .asse9 E' il nome dell'asse le cui quote programmate devono essere invertite

Caratteristiche:

Con il comando MIR è possibile programmare fino a 9 assi. Non è ammessa la ripetizione di un nome asse. Per gli assi non dichiarati viene annullata la lavorazione speculare programmata precedentemente.

Se non è programmato nessun operando, la lavorazione speculare viene disabilitata per tutti gli assi configurati.

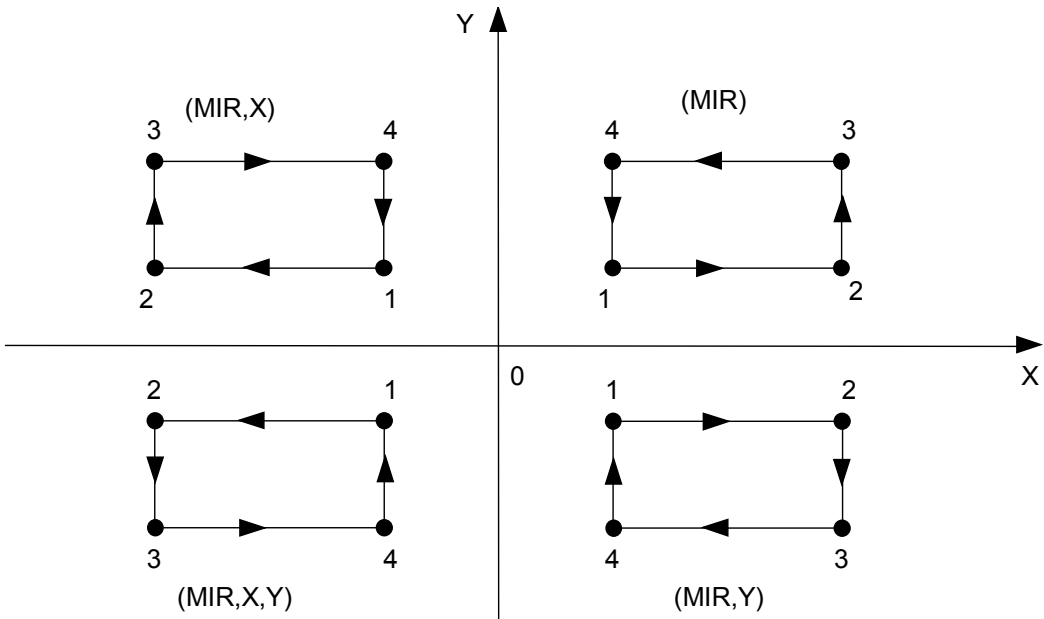
La lavorazione speculare su di un asse viene applicata al primo movimento dell'asse dopo l'istruzione MIR. L'inversione avviene attorno all'origine presente.

Usando la rotazione del piano (ROT) con il comando MIR, il controllo applica prima MIR e poi ROT.

IMPORTANTE

Il RESET di sistema disabilita la modalita' MIRROR su tutti gli assi. (equivale a (MIR) senza parametri).

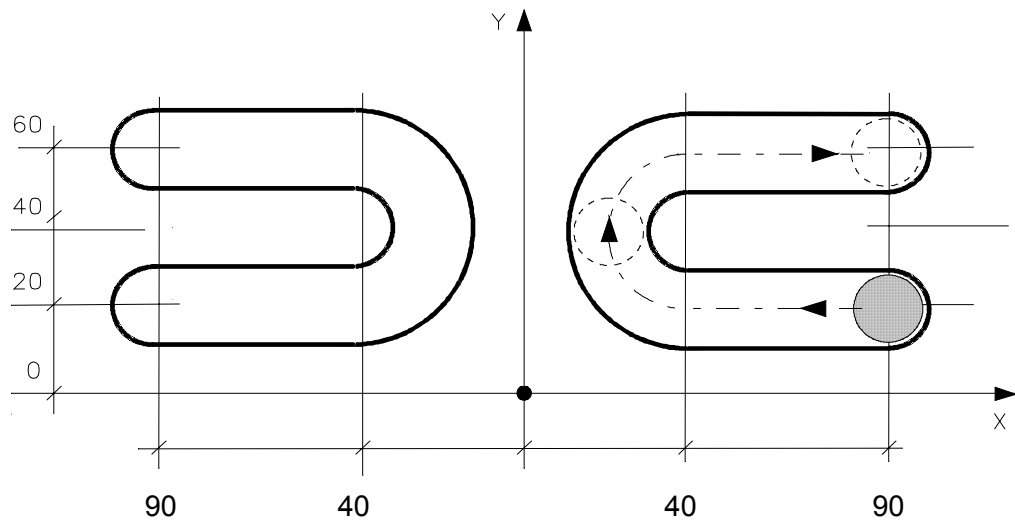
Esempio 1:
Il seguente è un esempio di come funziona la lavorazione con mirror.



.	Lavorazione speculare disattivata e spostamenti nel primo quadrante. Gli spostamenti vengono eseguiti rispetto all'origine corrente.
.	
N24 (MIR,X)	Lavorazione speculare attivata solo per l'asse X. Gli spostamenti programmati con +X hanno come risultato uno spostamento nel secondo quadrante.
.	
.	
.	
N42 (MIR, X, Y)	Lavorazione speculare attivata per gli assi X ed Y. Il risultato è uno spostamento dei movimenti nel terzo quadrante.
.	
.	
.	
N84 (MIR, Y)	Lavorazione speculare attivata per l'asse Y e disattivata per l'asse X. Spostamenti eseguiti nel quarto quadrante.
.	
.	
.	
N99 (MIR)	Lavorazione speculare disattivata per tutti gli assi. Spostamenti nel primo quadrante.

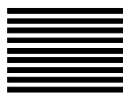
Esempio 2:

Questo esempio illustra l'utilizzo del comando MIR. Si noti l'uso dei comandi RPT ed ERP.



Programma:

```
(UGS, X, -100, 100, Y, 0, 80)
N199 (DIS."FRESA D =16")
N200 S1500 T8.8 M6
N201 (RPT, 2)
N202 G X90 Y20 M3
N203 Z-100
N204 G1 Z-105 F150
N205 X40 F200
N206 G2 Y60 I40 J40
N207 G1 X90
N208 G Z-100
N209 (MIR, X)
N210 (ERP)
N211 (MIR)
N212 Z
```



ROT (URT) - Rotazione del piano d'interpolazione

ROT determina la rotazione del piano d'interpolazione attivo di un valore angolare programmato. Il centro di rotazione è l'origine corrente. ROT può essere attivato da tastiera o da programma.

Sintassi

(ROT, *angolo*)

dove:

angolo Rappresenta il valore di un angolo espresso in gradi. E' possibile programmare l'angolo direttamente o indirettamente con un parametro E. Gli angoli positivi vengono misurati in senso antiorario rispetto all'asse delle ascisse del piano di interpolazione attivo. Gli angoli negativi sono misurati in senso orario. Se il parametro *angolo* vale zero, la rotazione del piano viene disabilitata.

Caratteristiche:

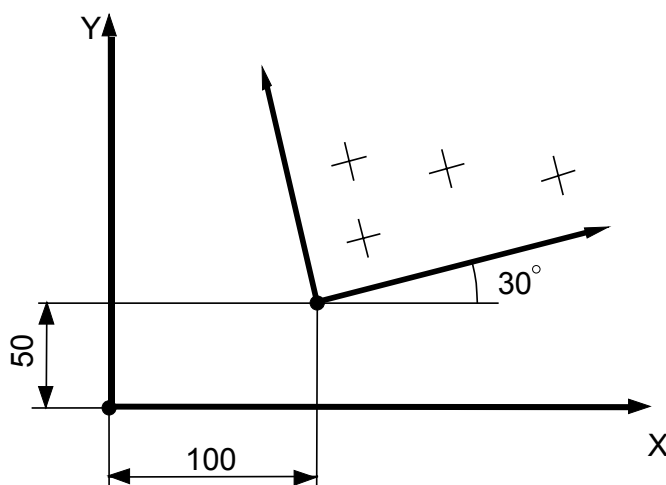
Il controllo applica la rotazione delle coordinate a partire dal primo blocco dopo il comando ROT. Il controllo non ruota le coordinate riferite allo zero macchina (G79).

Programmando la rotazione degli assi (ROT) con il comando MIR, il controllo li esegue nell'ordine seguente: prima MIR poi ROT.

IMPORTANTE

Il RESET di sistema disabilita la rotazione del piano (equivale a (ROT,0))

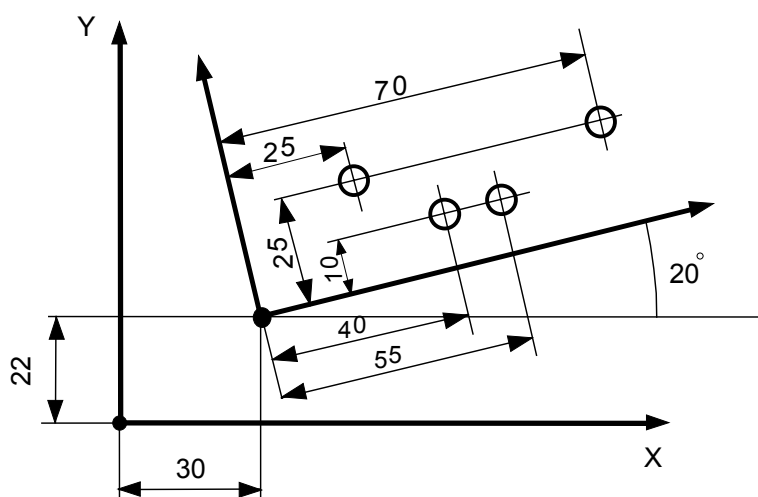
Esempio 1:



Programma:

(UTO, 1, X100, Y50)	Seleziona l'origine assoluta 1, modificandola temporaneamente, di X100 Y50
(ROT, 30)	Specifica una rotazione antioraria di 30 gradi rispetto all'origine temporanea
.	Gli spostamenti in questa parte di programma sono riferiti all'origine temporanea ruotata di 30 gradi in senso antiorario
.	
(UAO, 1)	Riattiva l'origine assoluta 1
(ROT, 0)	Disattiva la rotazione programmando una rotazione di 0 gradi rispetto all'origine 1

Esempio 2:



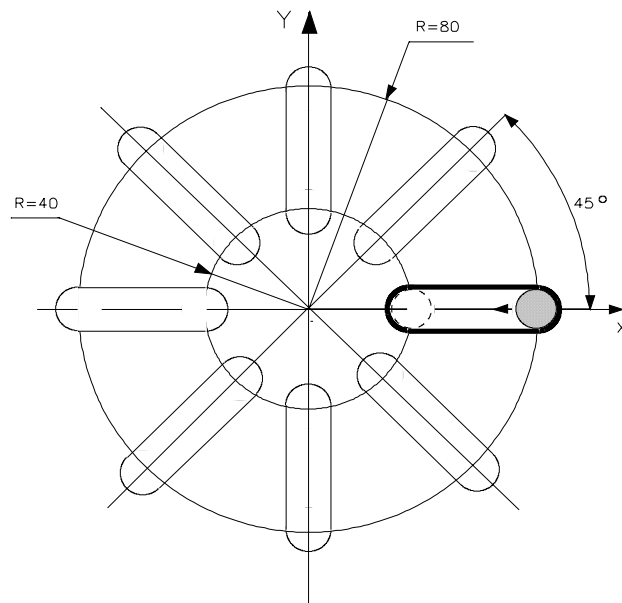
Programma:

```

(UGS, X, 0, 70, Y, 0, 70)
N99 (DIS, "PUNTA D=6")
N100 S2000 F200 T3.03 M6
N101 (UTO, 1, X30, Y22)
N102 (ROT, 20)
N103 G81 R-90 Z-110 M3
N104 X25 Y25
N105 X40 Y10
N106 X55
N107 X70 Y25
N108 G80 Z
N109 (UAO, 1)
N110 (ROT, 0)
N111 S1000 T4.4 M6
  
```

Esempio 3:

Rotazione del Piano (ROT) con Ripetizione (RPT) e programmazione parametrica.



Programma:

(UGS, X, -100, 100, Y, -100, 100)

N148 (DIS, " ...")

N149 S1500 T5.5 M6

N150 E25 =0

N151 (RPT, 8)

N152 (ROT, E25)

N153 G X40 Y M3

N154 Z0

N155 G29 G1 Z-10 F150

N156 X80 F200

N157 Z-18 F150

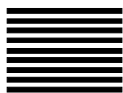
N158 X40

N159 G Z0

N160 E25 =E25 + 45

N161 (ERP)

N162 (ROT, 0)



UAO - Uso delle origini assolute

Il comando UAO consente di attivare ed utilizzare una delle origini assolute create e memorizzate precedentemente tramite l'apposito menu.

Sintassi

(UAO, n [,asse1, . . . , asse9])

dove:

n Definisce il numero dell'origine assoluta che si vuole utilizzare: è un numero intero compreso tra 0 e 10 programmato direttamente o indirettamente con un parametro E. Se questo parametro ha valore 0, viene abilitata la posizione di "HOME".

asse1,...,asse9 E' il nome dell'asse che deve utilizzare l'origine assoluta n .

Caratteristiche:

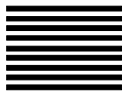
Il comando UAO ammette fino a nove nomi assi. La ripetizione di un nome asse non è ammessa. Per gli assi non dichiarati rimane abilitata l'origine corrente. Un comando UAO programmato senza assi (UAO, n) attiva l'origine n per tutti gli assi. All'accensione, dopo un reset del controllo o con $n=0$ e senza assi, tutti gli assi sono riferiti alla posizione di "HOME". Se il programma richiede origini differenti per assi differenti, programmare un comando UAO per ciascuna delle origini richieste.

Le origini vengono convertite automaticamente e visualizzate nelle unità di misura correnti (G70/G71).

Le origini vengono memorizzate riferite alla posizione di "HOME" caratterizzata in AMP.

Esempio:

(UAO,1)	Viene attivata per tutti gli assi l'origine assoluta 1.
.	
.	Questa parte di programma usa l'origine 1 per tutti gli assi.
.	
(UAO, 2, X, Y)	L'origine assoluta 2 viene attivata solo per gli assi X ed Y.
(UAO, 3, Z)	L'origine 3 viene attivata per l'asse Z.
.	
.	Questa parte di programma usa l'origine 2 per X ed Y, l'origine 3 per Z, e l'origine 1 per tutti gli altri assi.
(UAO, 1)	L'origine 1 è riattivata per tutti gli assi.
.	
.	
.	
(UAO, 0)	Tutti gli assi sono riferiti alla posizione di "HOME".



UTO (UOT) - Uso di origini temporanee

Seleziona l'origine assoluta dichiarata nel blocco modificandola temporaneamente, asse per asse, di una quantità pari a quella programmata.

Sintassi

(UTO, *n*, *asse1* [, *asse2*, . . . , *asse9*])

dove:

<i>n</i>	Definisce il numero di origine assoluta che si desidera modificare temporaneamente. E' un valore compreso tra 0 e 10. Se il valore <i>n</i> è 0, l'offset programmato viene sommato al valore della posizione "HOME".
<i>asse1</i> ,... <i>asse9</i>	E' un asse più una quota. Il controllo tratta la quota come una correzione e la aggiunge al valore dell'origine assoluta per quell'asse.

Caratteristiche:

Nel comando UTO bisogna dichiarare almeno un asse con una quota. Si possono dichiarare fino a nove assi con le relative quote ma non sono ammesse le ripetizioni.

La quota dell'asse è programmata nell'unità di misura (G70/G71) corrente.

Per gli assi non dichiarati, nel blocco UTO rimane attiva l'origine assoluta corrente.

Una volta attivata un'origine temporanea, essa rimane attiva fino a che:

- non si attiva una nuova origine temporanea con il comando UTO
- non si attiva una nuova origine assoluta con il comando UAO
- non si esegue un reset del controllo.

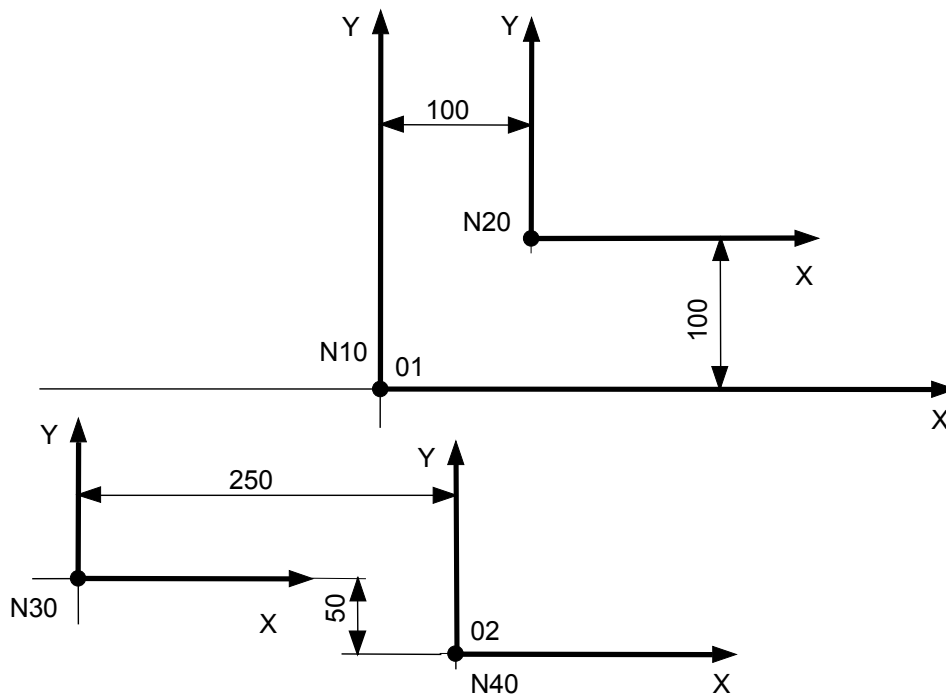
Se viene impostato un fattore di scala (SCF), il controllo lo applica all'origine temporanea UTO.

Esempio 1:

Le quote possono essere parametri E come indicato nel blocco seguente:

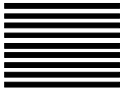
(UTO,1,XE100)

Esempio 2:



Programma:

N10 (UAO,1)	Attiva l'origine assoluta 1 per tutti gli assi.
.	
.	Questa parte di programma usa l'origine 1 per tutti gli assi.
.	
N20 (UTO, 1, X100, Y100)	Applica un'origine temporanea (offset assoluto) all'origine 1, X100 ed Y100.
.	
.	Questa parte di programma usa l'origine 1, con origine temporanea X100 Y100.
N30 (UTO, 2, X-250, Y50)	Applica un'origine temporanea (offset assoluto) all'origine 2, X-250 e Y50.
.	
.	Questa parte di programma usa l'origine assoluta 2, con X-250 ed Y50 come origine temporanea e origine 1 per gli altri assi.
N40 (UAO, 2)	Riattiva l'origine assoluta 2 per tutti gli assi.



UIO - Uso delle origini incrementali

Permette di effettuare uno spostamento incrementale, asse per asse, dell'origine del sistema di riferimento corrente.

Sintassi

(UIO, asse1 [,asse2, . . . , asse9])

dove:

asse1,...asse9 E' un asse più una quota. Il controllo tratta la quota come un offset incrementale e lo aggiunge al valore dell'origine corrente dell'asse.

Caratteristiche:

Nel comando UIO bisogna dichiarare almeno un asse e fino a un massimo di nove assi.

La quota dell'asse deve essere programmata nell'unità di misura (G70/G71) corrente.

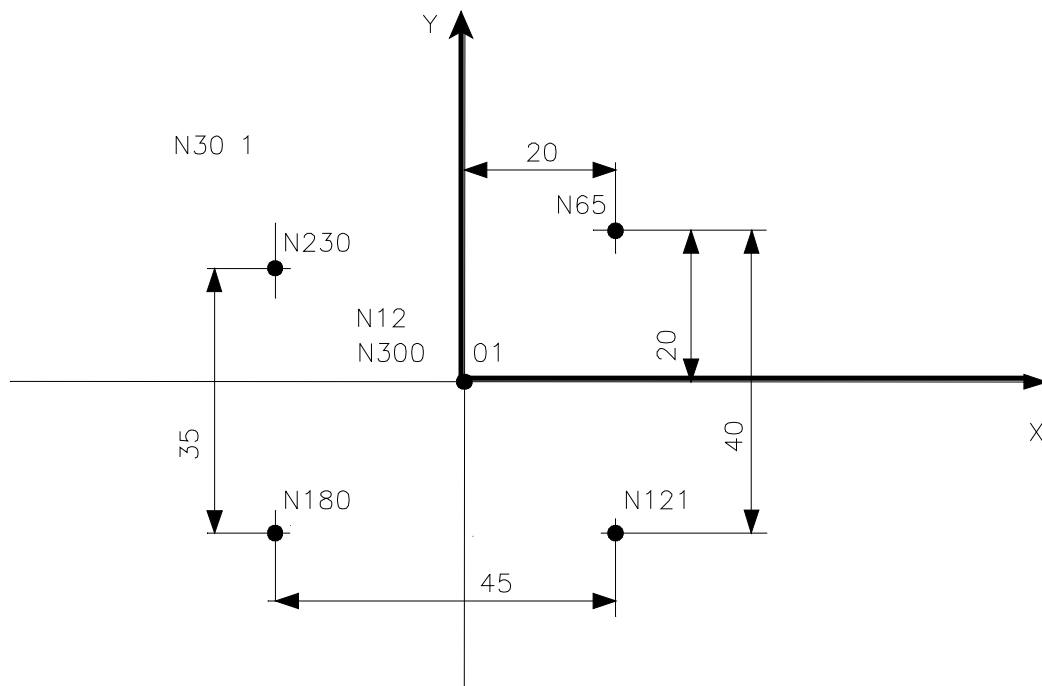
Per gli assi non dichiarati rimane abilitata l'origine corrente.

Una volta attivata un'origine incrementale per un asse, essa rimane attiva fino a che:

- non si attiva una nuova origine incrementale per l'asse tramite il comando UIO
- non si attiva un'origine assoluta tramite il comando UAO
- non si esegue un reset del controllo

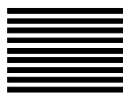
Se è stato impostato un fattore di scala (SCF), il controllo lo applica all'origine incrementale UIO.

Esempio:



Programma:

N12 (UAO,1)	Attiva l'origine assoluta 1 per tutti gli assi.
N65 (UIO, X20, Y20)	Produce un incremento di X20 e Y20 rispetto all'origine 1. Per gli altri assi rimane attiva l'origine assoluta 1.
N121 (UIO, Y-40)	Produce un incremento di Y-40 rispetto all'ultima origine attiva. L'origine incrementale X20 rimane attiva.
N180 (UIO, X-45)	Produce un'origine incrementale di X-45. L'origine incrementale Y-40 rimane attiva.
N230 (UIO, Y35)	Produce un'origine incrementale di Y35. L'origine incrementale X-45 rimane attiva.
N300 (UAO,1)	Riattiva l'origine assoluta 1 per tutti gli assi.



RQO - Riqualifica delle origini

Modifica l'origine assoluta per gli assi dichiarati nell'istruzione di una quantità programmata. L'origine deve essere una di quelle memorizzate nella tabella origini.

Sintassi

(RQO, *n*, *asse1* [, *asse2*, . . . , *asse9*])

dove:

- | | |
|--------------------------------|--|
| <i>n</i> | Definisce il numero (da 1 a 10) dell'origine assoluta da modificare. L'origine assoluta deve essere stata creata e memorizzata tramite softkey. Si può programmare il numero di origine <i>n</i> direttamente con un numero intero positivo o indirettamente tramite un parametro E. |
| <i>asse1</i> ,... <i>asse9</i> | E' un asse più una quota. La quota è un incremento aggiunto all'origine assoluta di un dato asse. Si può programmare la quota direttamente con un numero decimale o indirettamente con un parametro E. |

Caratteristiche:

Nel comando RQO bisogna dichiarare almeno un asse e la sua quota e fino ad un massimo di nove assi. Non sono ammesse le ripetizioni.

Le quote di riqualifica devono essere programmate nell'unità di misura corrente e ad esse non viene applicato il fattore di scala.

L'origine è modificata sia sul file delle origini (per cui il risultato della riqualifica è permanente) sia in memoria (se tale origine è attivata per l'asse al momento della riqualifica).

Nella tabella delle origini i valori di riqualifica vengono applicati nell'unità di misura in cui è espressa l'origine selezionata.

La quota di riqualifica nel caso di un asse diametrale deve essere programmata in radiale, in quanto la quota è un incremento da sommare al valore già presente nell'origine che è sempre un valore radiale.

Esempio:

(RQO, 3, X (E31)) Modifica l'origine assoluta n° 3 per l'asse X del valore contenuto in E31.

FINE CORSA ED AREE PROTETTE

I comandi di questa classe definiscono i limiti di fine corsa e le aree protette secondo le modalità descritte di seguito.

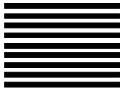
COMANDO	FUNZIONE
----------------	-----------------

SOL	Definizione dei fine corsa software
-----	-------------------------------------

DPA	Definizione delle aree protette
-----	---------------------------------

PAE	Abilitazione di un'area protetta
-----	----------------------------------

PAD	Disabilitazione di un'area protetta
-----	-------------------------------------



SOL (DLO) - Fine corsa software

Il comando SOL definisce i limiti operativi della corsa dell'asse cioè le quote oltre le quali gli assi non possono essere programmati. I limiti vengono misurati rispetto all'origine corrente.

Sintassi

(SOL, nome-asse, limite inferiore, limite superiore)

dove:

nome-asse E' il nome dell'asse di cui bisogna definire i limiti.

limite inferiore E' una quota per il limite inferiore.

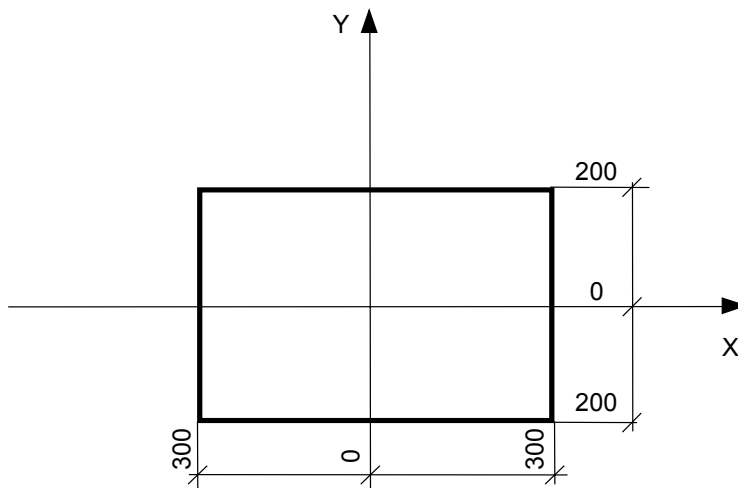
limite superiore E' una quota per il limite superiore. Il limite superiore deve essere maggiore del limite inferiore.

Caratteristiche:

Se i fine corsa software programmati eccedono i limiti specificati in AMP, il controllo segnala errore.

Bisogna programmare le dimensioni per i fine corsa software nell'unità di misura (G70/G71) attiva nel momento in cui si programma il comando SOL. Eventuali fattori di scala (SCF) attivi vengono applicati ai limiti operativi.

Esempio:



Programma:

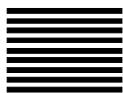
(SOL, X, -300, 300)

.

.

.

(SOL, Y, -200, 200)



DPA (DSA) - Definizione delle aree protette

Il comando DPA definisce un'area protetta, cioè un'area entro la quale non può essere programmato nessun movimento.

Sintassi

(DPA, *n*,*nome-asse1*,*limite inferiore1*,*limite superiore1*,*nome-asse2*,*limite inferiore2*, *limite superiore2*)

dove:

n E' il numero di identificazione dell'area (1-3).

nome-asse1 E' il nome dell'asse delle ascisse.

limite inferiore1 E' il limite inferiore dell'asse delle ascisse.

limite superiore1 E' il limite superiore dell'asse delle ascisse.

nome-asse2 E' il nome dell'asse delle ordinate.

limite inferiore2 E' il limite inferiore dell'asse delle ordinate.

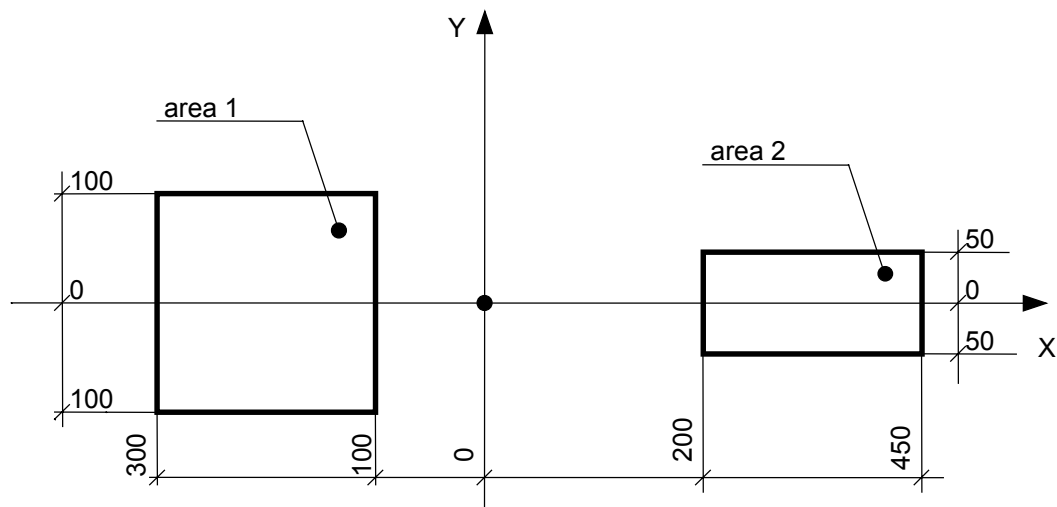
limite superiore2 E' il limite superiore dell'asse delle ordinate.

Caratteristiche:

Prima di iniziare un movimento programmato con interpolazione lineare o circolare, il controllo effettua una verifica per vedere se lo spostamento interferisce con un'area protetta.

E' possibile definire da programma fino a tre aree protette. Ogni area viene definita rispetto all'origine attiva al momento in cui viene specificato il comando DPA.

Esempio:



Programma:

N1 (DPA,1, X, -300, -100, Y, -100, 100)

N2 (DPA, 2, X, 200, 450, Y, -50, 50)

N3 (PAE, 1)

N4 (PAE, 2)

N5 T1.1 M6

.

.

.

N80 (PAD, 1)

.

.

.

N99 M30



PAE (ASC) - Abilitazione dell'area protetta

Il comando PAE attiva il controllo di un'area protetta.

Sintassi

(PAE, n)

dove:

n E' il numero di un'area protetta precedentemente definita con il comando DPA.

Esempio:

(PAE,2) abilita il controllo dell'area 2.



PAD (DSC) - Disabilitazione dell'area protetta

Il comando PAD disattiva il controllo di un'area protetta.

Sintassi

(PAD, n)

dove:

n E' il numero di un'area protetta precedentemente definita con il comando DPA.

Esempio:

(PAD,3)

Questa istruzione disabilita il controllo sull'area 3.

VIRTUALIZZAZIONI

Le virtualizzazioni sono un'insieme di prestazioni che permettono di gestire in un qualunque spazio tridimensionale, gli assi reali della macchina utensile.

Assi Virtuali

Gli assi virtuali vengono dichiarati in fase di configurazione, ma non vengono associati a nessun parametro di tipo fisico (vedi manuale AMP).

Quando usare gli assi virtuali:

- per gestire piani ruotati nello spazio
- nel caso di trasformazione da piani cartesiani a piani gestiti in coordinate polari.
- nel caso di trasformazione da piani cartesiani a piani gestiti in coordinate cilindriche.
- per effettuare movimentazioni lungo la direzione dell'utensile.
- per effettuare una compensazione della lunghezza utensile durante l'esecuzione di profili che implicano il movimento di cinque assi contemporanei (es. con utensile montato su un polso dotato di due gradi di libertà).

IMPORTANTE

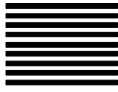
Qualora si vogliano effettuare dei movimenti circolari (G02 - G03) su un piano virtuale precedentemente programmato è necessario ridefinire il piano di interpolazione utilizzando la funzione G16.

Alla disattivazione degli assi virtuali viene forzato come piano di interpolazione quello di default configurato in AMP

Su RESET le virtualizzazioni vengono disattivate.

Virtualizzazioni disponibili sul CNC Serie 10

COMANDO	FUNZIONE
UPR	(USE PLANE ROTATED) Rotazione della terna cartesiana XYZ nello spazio.
UVP	(USE VIRTUAL POLAR) Trasformazione da coordinate cartesiane a coordinate polari.
UVC	(USE VIRTUAL CYLINDRICAL) Trasformazione da coordinate cartesiane a coordinate cilindriche.
TCP	(TOOL CENTER POINT) Programmazione in quote pezzo anzichè in quote macchina. Movimentazione lungo la direzione utensile.



UPR - Rotazione della terna cartesiana

Il comando UPR (USE PLANE ROTATED) permette la definizione di una terna di assi virtuali rototraslati.

Sintassi

```
(UPR, tipo, af1af2af3, av1av2av3[,rot1,rot2,rot3 [,q1,q2,q3 [,hor,ver]]])
(upr, tipo, af1af2af3, av1av2av3[,rot1,rot2,rot3 [,q1,q2,q3 [,hor,ver]]])
(UPR)
(UPR,99)
(UPR, 5, af1af2af3, av1av2av3, tipo_req, pos-xx-H, pos-xx-V, n, ref-pos-xx-H, ref-pos-xx-V)
```

dove:

tipo

Rappresenta il tipo di rotazione:

- 0=assoluta
gli angoli di rotazione sono riferiti agli assi fisici.
- 1= incrementale
gli angoli di rotazione sono riferiti alla terna in uso all'atto della programmazione incrementale.
Questo tipo di rotazione è possibile se è già stata programmata una precedente rotazione (assoluta la prima volta, incrementale le volte successive).
- 2= assoluta, con trasformazione su 5 assi
Le caratteristiche sono simili al tipo 0, ma la rotazione degli assi lineari influenza la posizione dei due assi rotativi.
All'attivazione della funzione UPR, gli assi rotativi assumono un nuovo valore che individua la posizione dell'asse utensile rispetto al nuovo sistema virtuale.
- 3 = incrementale, con trasformazione su 5 assi
Le caratteristiche sono simili al tipo 1 a cui vanno aggiunte le considerazioni descritte nel punto precedente.
- 4 = piano di rotazione determinato automaticamente in base alla direzione dell'asse utensile.
- 5 = non definisce una virtualizzazione nuova ma permette di richiedere al sistema le quote, fisiche o virtuali, degli assi rotativi relative alla virtualizzazione attiva in quel momento.
- 6 = come 4 ma con trasformazione 5 assi (vedi 2).
- 10 = globale di tipo 0
Le caratteristiche sono simili al tipo 0 ma la rotazione programmata rimane attiva anche con successivi UPR di tipo non globale.

- 12 = globale di tipo 2
Le caratteristiche sono simili al tipo 2 ma la rotazione programmata rimane attiva anche con successivi UPR di tipo non globale.
- 14 = globale di tipo 4
Le caratteristiche sono simili al tipo 4 ma la rotazione programmata rimane attiva anche con successivi UPR di tipo non globale.
- 16 = globale di tipo 6
Le caratteristiche sono simili al tipo 6 ma la rotazione programmata rimane attiva anche con successivi UPR di tipo non globale.
- 99
senza ulteriori parametri, disabilita tutte le modalità UPR attive compresa la rototraslazione globale.

af1af2af3 Sono i nomi dei tre assi fisici da virtualizzare (es.XYZ).

av1av2av3 Sono i nomi dei tre assi virtuali interessati dalla movimentazione (es.UVW).

rot1,rot2,rot3 Sono i valori degli angoli di rotazione espressi in gradi (vedere la pagina seguente per determinare il senso di rotazione).

q1,q2,q3 Sono i valori che servono a determinare l'origine del sistema di riferimento:

- nel caso di rotazione assoluta rappresentano le coordinate del punto di origine del sistema. Esse sono riferite all'origine in uso all'atto della programmazione di UPR.
- nel caso di rotazione incrementale rappresentano gli incrementi dell'origine in uso. Il punto determinato dai valori dell'origine già in uso sommati ai valori di incremento, viene assunto come nuovo punto di origine del sistema.

Nel caso in cui *q1,q2,q3* non siano presenti, l'origine del sistema di riferimento coincide con l'origine del sistema fisico in uso.

hor,ver Sono valori di angoli, espressi in gradi e interni all'angolo giro (compresi tra -360° e +360°), che identificano l'origine degli assi rotativi orizzontale e verticale.

Sono da considerarsi sempre anch'essi come traslazioni degli assi fisici.

Tipo_req E' il tipo di quota richiesta al sistema:

- **0** = posizione **fisica**
- **1** = posizione **virtuale**

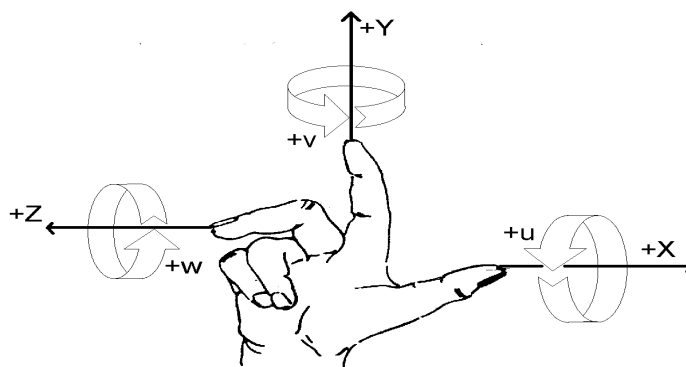
pos-xx-H Angolo espresso in gradi; è la quota fisica o virtuale dell'asse rotativo orizzontale sulla quale applicare le matrici di rotazione al fine di determinare la corrispondente quota richiesta.

pos-xx-V Angolo espresso in gradi; è la quota fisica o virtuale dell'asse rotativo verticale sulla quale applicare le matrici di rotazione al fine di determinare la corrispondente quota richiesta.

<i>n</i>	Indice delle variabili E da cui partire per salvare i valori degli angoli richiesti ritornati dal sistema: la prima coppia (quota asse orizzontale, quota asse verticale) corrisponde alla soluzione più vicina ai ref-pos-xx, la seconda (Hor, Ver) è complementare alla prima.
<i>ref-pos-xx-H</i>	Angolo espresso in gradi; è il valore nell'intorno del quale deve essere calcolata la soluzione per l'asse orizzontale.
<i>ref-pos-xx-V</i>	Angolo espresso in gradi; è il valore nell'intorno del quale deve essere calcolata la soluzione per l'asse verticale.
<i>nessun parametro</i>	(UPR) senza parametri disabilita la modalità UPR lasciando solo quella globale, se presente. Se non è stata programmata una rotazione globale i comandi (UPR) e (UPR,99) sono equivalenti.
<i>programmazione minuscola (upr)</i>	<p>Programmando il triletterale minuscolo è possibile variare gli angoli e/o origini senza disabilitare l'algoritmo UPR e senza uscire dalla modalità "MOVIMENTO CONTINUO".</p> <p>Il triletterale (upr,.....) minuscolo deve essere usato solamente per variare i parametri del (UPR,....), di tipo non globale, programmato in precedenza. Non è possibile variare la sequenza degli assi. Non è possibile programmare una rotazione di tipo globale</p> <p>In caso di trasformazione a 5 assi, il comportamento dell'upr minuscolo può, in certi casi, differire da quello dell'UPR maiuscolo. Infatti, poiché in continuo la sincronizzazione con le quote fisiche è inapplicabile, i valori degli angoli calcolati dal sistema sono normalizzati rispetto allo 0,0 e quindi potrebbero risultare complementari a quelli desiderati, fermo restando che le soluzioni dal punto di vista della posizione utensile sono corrette.</p>



Per determinare il senso di rotazione della terna virtuale fare riferimento alla regola della mano destra.



Regola della mano destra.

Caratteristiche:

La funzionalità UPR permette di programmare qualunque funzione di macchina in uno spazio ruotato con angoli definiti a piacere rispetto alla terna cartesiana della macchina utensile.

Questo consente all'operatore di programmare nel normale sistema cartesiano (XYZ) la definizione del profilo da lavorare, e poi delegare al CNC Serie 10 il compito di ricalcolare la movimentazione degli assi secondo piani virtuali derivanti dalla rotazione.

La programmazione di un **UPR globale** permette di applicare una rototraslazione iniziale alla quale tutti i successivi upr fanno riferimento. Questo consente, di eseguire lo stesso part program, che a sua volta può contenere altre istruzioni di trasformazione geometriche, anche se il pezzo da lavorare è, ad esempio, staffato sul bancale con orientamenti differenti.

L'attivazione di un UPR di tipo globale non viene vista come UPR precedente ai fini della programmazione incrementale o minuscola.

L'attivazione di un UPR di tipo globale annulla una qualunque virtualizzazione precedente con conseguente azzeramento degli eventuali spiazamenti.

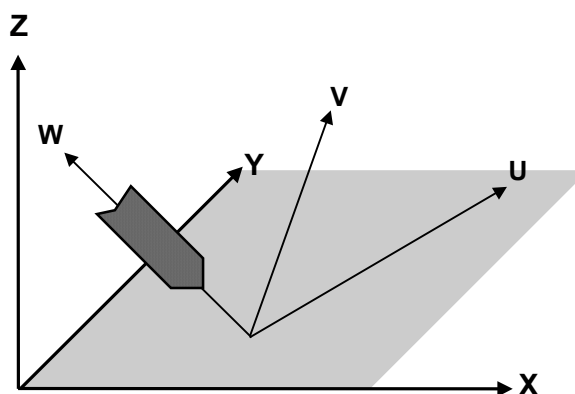
Nel caso di utilizzo di **UPR di tipo 2, 3, 4 e 6** o dei corrispondenti di tipo globale, il sistema necessita delle caratteristiche della macchina che vanno inserite nella tabella **TCP** secondo le regole definite dal TCP stesso. La prestazione è comunque indipendente dal TCP, ovvero può lavorare sia con TCP attivo che no. Quando UPR e (TCP, 5) sono attivi in contemporanea, i movimenti sull'asse utensile influenzeranno il movimento degli assi virtuali.

Le **modalità 2 e 3** fanno sì che l'asse utensile, individuato dalla programmazione degli assi rotativi, assuma nel sistema di riferimento virtuale la stessa posizione che ha nel sistema di riferimento individuato dalla terna cartesiana.

Le **modalità 4 e 6** e i corrispondenti di tipo globale, determinano il nuovo piano di rotazione in base alla direzione dell'utensile (quindi in base alla posizione degli assi rotativi). Il nuovo piano, ipotizzando la programmazione UPR, 4, XYZ, UVW ..., avrà l'asse W coincidente con la direzione utensile, l'asse U giacente nel primo quadrante del piano XY originario, l'asse V di conseguenza in base alla regola di descrizione della terna cartesiana.

Non è possibile programmare un'origine di tipo **UAO** o **UTO** con una virtualizzazione attiva.

Le origini sugli assi rotativi attuate con un comando di **UIO** non sono gestite.



Tramite gli angoli *rot1*, *rot2*, *rot3*, si potrà ulteriormente ruotare la terna *UVW* così determinata, in pratica si applicherà un incremento (vedi tipo 1 o 3) alla nuova terna.

Al comando di **RESET** viene disabilitata la modalità UPR.

IMPORTANTE

Nel caso di **UPR globale**, i valori di *rot1*, *rot2*, *rot3* devono essere compresi tra -90° e $+90^\circ$.

Per angoli esterni a tale intervallo, l'algoritmo di generazione della posizione degli assi rotativi, nel passaggio da una virtualizzazione all'altra, potrebbe generare delle rotazioni indesiderate, ma comunque congruenti alla posizione dell'utensile.

IMPORTANTE

L'**UPR** è **insensibile alla DAN**: gli assi che si devono specificare nell'upr sono i nomi degli assi fisici e quelli dei virtuali voluti, indipendentemente da eventuali comandi di DAN effettuati precedentemente.

IMPORTANTE

Dalla release 7.4 per correggere mal funzionamenti e risolvere problemi di indeterminatezza delle soluzioni durante una virtualizzazione con trasformazione a 5 assi, sono stati modificati gli algoritmi di calcolo delle quote degli assi rotativi. La compatibilità all'indietro è comunque garantita definendo la variabile utente **!R73MODE**.

!R73MODE = 1 : per il calcolo della virtualizzazione vengono usati gli algoritmi vecchi, allineamento alla release 7.3

!R73MODE = 0 o non definita: per il calcolo della virtualizzazione vengono usati gli algoritmi nuovi, allineamento alla release 7.4 o superiori

Questa variabile utente viene letta solo in corrispondenza della programmazione di UPR maiuscolo.

Viene segnalato errore di programmazione, se vengono programmate nel triletterale dell'UPR le origini sui rotativi e si verifica una delle seguenti condizioni:

- la variabile utente **!R73MODE** è settata a 1. Questo perché la suddetta programmazione viene gestita dalla release 7.5 in poi ed è in contraddizione con il voler utilizzare gli algoritmi di calcolo allineati alla release 7.3.
- il tipo di UPR programmato non richiede trasformazione 5 assi.

Modalità di rotazione:

La rotazione della terna cartesiana viene eseguita in modo sequenziale rispetto agli angoli di rotazione programmati; questo comporta che:

- A) Il sistema di coordinate *af1 af2 af3* viene ruotato di un angolo *rot1* intorno all'asse *af1*
- B) Il nuovo sistema di coordinate *av1' av2' av3'*, derivante dalla rotazione descritta al punto A), viene ruotato di un angolo *rot2* intorno all'asse *av2'*
- C) Il nuovo sistema di coordinate *av1'' av2'' av3''* derivante dalle rotazioni descritte ai punti A) e B), viene ruotato di un angolo *rot3* intorno all'asse *av3''*

Come risultato di queste operazioni si ha il sistema identificato dagli assi virtuali *av1 av2 av3*.

IMPORTANTE

Le operazioni A) B) e C) sono eseguite in modo sequenziale rispetto agli angoli ed agli assi programmati nel triletterale; pertanto l'ordine di definizione degli angoli e degli assi fisici è fondamentale.

Esempi d'uso di UPR

Negli esempi che seguono si presuppone che il sistema di riferimento sia rappresentato dalla terna cartesiana XYZ.

Negli esempi 1, 2, 3, 4, 5 l'origine del sistema di riferimento coincide con l'origine in uso all'atto della programmazione di UPR.

Esempio 1:

```
.
.
GXYZ
(UPR,0,XYZ,UVW,30,45,60)
G1F5400
U100.4V9.12 W-70
U70.345 W-20
.
.
.
G16UV
G02 U100 V70 R15

G1 U120 W10
.
.
.
(UPR)
X1 Z4.9
.
```

Il sistema UVW è ottenuto da:

- A) rotazione di 30° del sistema XYZ intorno all'asse X.
- B) rotazione del sistema U'V'W' derivante dal punto A) di 45° intorno a V'
- C) rotazione del sistema U''V''W'' derivante dai punti A) e B) di 60° intorno a W''

Esempio 2:

```
.
.
.
GXYZ20
(UPR,0,XYZ,UVW,10,0,80)
G1F4000
U50 V70
U90
V80 W60
.
.
.
(UPR)
.
.
.
```

Il sistema UVW è ottenuto da:

- A) rotazione del sistema XYZ di 10° intorno all'asse X.
- B) rotazione del sistema U'V'W' derivante dal punto A) di 80° intorno all'asse W'

Esempio 3:

```
.
.
.
GXYZ20
(UPR,0,ZYX,ABC,80,0,10)
G1F3000
A50 B70
A90
B80 C60
.
.
(UPR)
.
.
```

Il sistema ABC è ottenuto da:

- A) rotazione del sistema XYZ di 80° intorno all'asse Z.
- B) rotazione del sistema A'B'C' derivante dal punto A) di 10° intorno all'asse C'

IMPORTANTE

Il sistema virtuale UVW ottenuto nell'esempio 2) è differente dal sistema ABC dell'esempio 3).

Esempio 4:

```
.
.
.
(UPR,0,ZYX,WVU,0,50,60)
U90V30
W10
.
.
.
(UPR)
.
.
.
```

Il sistema WVU è ottenuto da:

- A) rotazione del sistema XYZ di 50° intorno all'asse Y.
- B) rotazione del sistema W'V'U' derivante dal punto A) di 60° intorno all'asse U'

Esempio 5:

```
.
.
GXYZ
(UPR,0,XYZ,UVW,30,0,0)
G16 UV
G1 F1000 U50 V0
V30
.
.
```

Il sistema UVW è ottenuto dalla rotazione del sistema XYZ di 30° intorno all'asse X

```
U25 V35
(UPR,1,XYZ,UVW,60,0,0)
U30 V20
.
.
```

Il sistema UVW, ottenuto dalla precedente rotazione, ruota ulteriormente di 60° intorno ad U.

```
(UPR)
GX10 Y25
.
.
```

Esempio 6:

```
.
.
GXYZ
(UPR,0,XYZ,UVW,30,45,60,10.8,20,-30.2)
G1F5400
U100.4V9.12 W-70
U70.345 W-20
.
.
```

```
G16UV
G02 U100 V70 R15
G1 U120 W10
.
.
```

```
(UPR)
X1 Z4.9
.
.
```

Il sistema UVW è ottenuto da:

- A) rotazione di 30° del sistema XYZ intorno all'asse X.
- B) rotazione del sistema U"V"W' derivante dal punto A) di 45° intorno a V'
- C) rotazione del sistema U"V"W" derivante dai punti A) e B) di 60° intorno a W"

L'origine del sistema di riferimento coincide con il punto di coordinate X10.8, Y20, Z-30.2

Esempio 7:

```
.
.
GXYZ
(UPR,0,XYZ,UVW,30,45,60,10.8,20,-30.2)
G1F500
U100.4 V9.12 W-70
U70.345 W-20
.
.
.
.
.
.
.
.
(UPR,1,XYZ,UVW,10,0,0,3,8,5)
U120 V30
.
.
.
.
(UPR)
GX70.5 Y10 Z25
.
.
```

Il sistema UVW è ottenuto da:

- A) rotazione di 30° del sistema XYZ intorno all'asse X.
- B) rotazione del sistema U'V'W' derivante dal punto A) di 45° intorno a V'
- C) rotazione del sistema U''V''W'' derivante dai punti A) e B) di 60° intorno a W''

L'origine del sistema di riferimento coincide con il punto di coordinate X10.8, Y20, Z-30.2

Il sistema UVW, ottenuto dalla precedente rotazione, ruota ulteriormente di 10° intorno ad U.
L'origine del sistema di riferimento coincide con il punto X13.8 Y28 Z-25.2

Esempio 8:

In questo caso si vuole applicare una rotazione iniziale, per es. di 90° attorno all'asse Z, al part program dell'esempio 2:

```
.
.
GXYZ
(UPR,10,XYZ,UVW,0,0,90)
(UPR,0,XYZ,UVW,10,0,80)
G1F4000
U50 V70
U90
V80 W60
.
.
(UPR,99)
```

Il sistema UVW è ottenuto da:

- A) rotazione di 90° del sistema XYZ intorno all'asse Z
- B) rotazione del sistema U'V'W' derivante dal punto A) di 10° intorno a U'
- C) rotazione del sistema U''V''W'' derivante dal punto B) di 80° intorno a W''

Disabilita tutti gli UPR attivi

Esempio 9:

Utilizzo dell'upr di tipo 5.

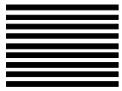
Si supponga di applicare, ad una macchina utensile di tipo Double Twist, una virtualizzazione che effettui una rotazione di 90° attorno all'asse Z.

Per brevità di notazione sia B il nome dell'asse di rotazione orizzontale, A quello dell'asse di rotazione verticale, WRK le relative quote virtuali e ABS le relative quote fisiche.

	Quote lette a video			
	WRK		ABS	
XYZAB	A 0	B 0	A 0	B 0
(UPR,2,XYZ,UVW,0,0,90)	A 0	B -90	A 0	B 0

Se si vuole far calcolare al sistema le quote fisiche relative alla quota virtuale A=0 e B=0, con inserimento delle soluzioni nelle variabili E a partire dall'indice 10, programmare il seguente upr:

	Quote lette nelle variabili E:			
(UPR,5,XYZ,UVW,0,0,0,10,0,0)	E10 = 90,	E11 = 0,	E12 = -90,	E13 = 180



UVP - Programmazione in coordinate polari

Il comando UVP (USE VIRTUAL POLAR) permette la definizione di coordinate polari.

Sintassi

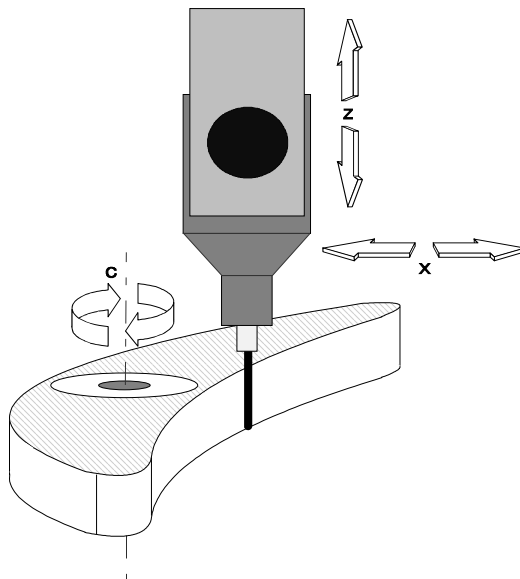
(UVP,af1af2,av1av2,r)
(UVP)

dove:

<i>af1</i>	E' il nome dell'asse fisico lineare (es. X)
<i>af2</i>	E' il nome dell'asse fisico rotativo (es. C)
<i>av1</i>	E' il nome dell'asse virtuale ascissa (es. U)
<i>av2</i>	E' il nome dell'asse virtuale ordinata (es. V)
<i>r</i>	E' il raggio minimo entro il quale non deve entrare il percorso utensile
<i>nessun parametro</i>	(UVP) senza parametri disabilita la modalità UVP.

Caratteristiche:

Questo tipo di virtualizzazione viene utilizzata nel caso si voglia movimentare un asse lineare X ed uno rotativo C programmando però le quote su un piano cartesiano UV; la posizione del generico punto P di coordinate (U,V) nel piano virtuale viene tradotta nelle coordinate (X,C) degli assi fisici.



Lavorazione in coordinate polari

IMPORTANTE

Il parametro **r** (raggio minimo) deve essere programmato tenendo conto della velocità di avanzamento F programmata, in modo da evitare che venga richiesta all'asse rotativo una velocità superiore a quella di rapido.

Il raggio minimo può essere calcolato tramite la seguente formula:

$$r = \frac{F}{V_{\max}} * \frac{360}{2\pi}$$

dove:

r = raggio minimo

F = velocità di avanzamento programmata (mm/min o inch/min)

V_{max} = velocità di rapido dell'asse rotativo

Nel caso in cui al parametro **r** venga assegnato un valore negativo, la limitazione sulla velocità di avanzamento viene effettuata dinamicamente, così da permettere avanzamenti elevati in punti di lavoro sufficientemente lontani dal centro di lavorazione.

IMPORTANTE

Le accelerazioni vengono calcolate in modo dinamico, pertanto è opportuno usare questa programmazione (**r-**) solo nel caso in cui non ci si avvicini troppo al centro di lavorazione in quanto, in questo caso, potrebbero essere imposte all'asse rotativo accelerazioni troppo elevate.

Esempio 1:

T1.1M6

GC0X50Y0

G16 UV

V20

r3

U-15

b5

V-20

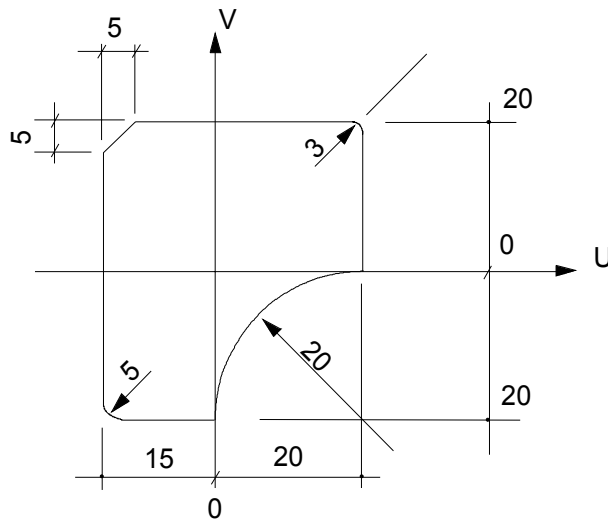
r5

U0

G40G2U20V0I20J-20

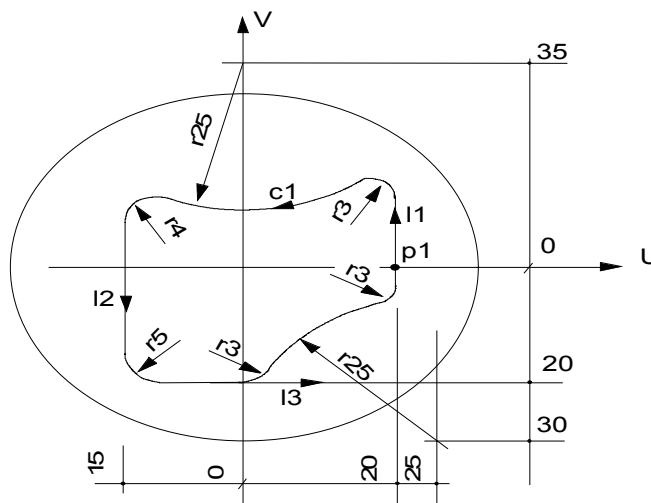
(UVP)

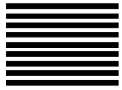
GX50



Esempio 2:

T1.1M6
S2000M3F300
GC0X80Y0-Z-5
(UVP,XC,UV,10)
G16UV
p1=U20V0
l1=p1,a90
c1=I0J35r-25
l2=U-15V0,a90
l3=U0V-20,a0
c2=I25J-30r-25
G21G42p1
l1
r3
c1
r4
l2
r5
l3
r3
c2
r3
l1
p1
G20G40
(UVP)
GX80





UVC - Programmazione in coordinate cilindriche

Il comando UVC (USE VIRTUAL CYLINDRICAL) permette la definizione di coordinate cilindriche.

Sintassi

(UVC,af1,av1,r)
(UVC)

dove:

af1 E' il nome dell'asse fisico rotativo (es. B)

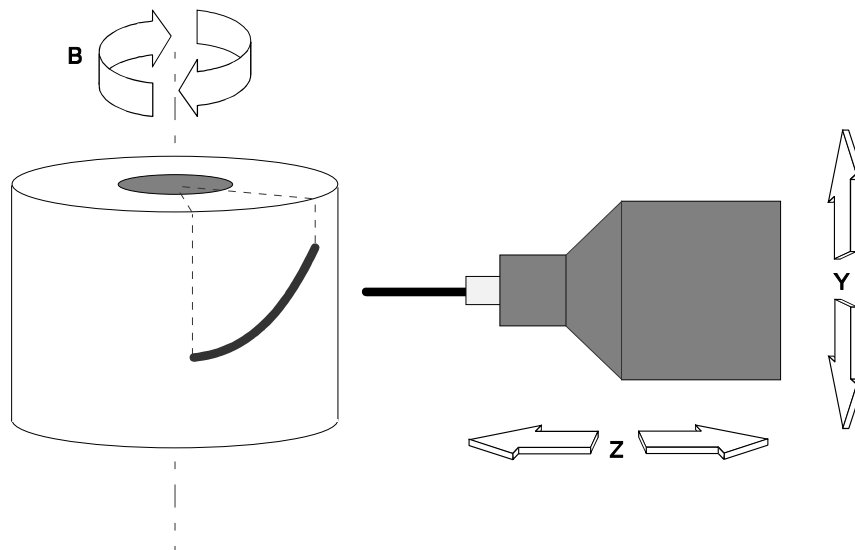
av1 E' il nome dell'asse virtuale (es. W)

r E' il valore del raggio del cilindro sul quale viene eseguito il profilo.

nessun parametro (UVC) senza parametri disabilita la modalità UVC.

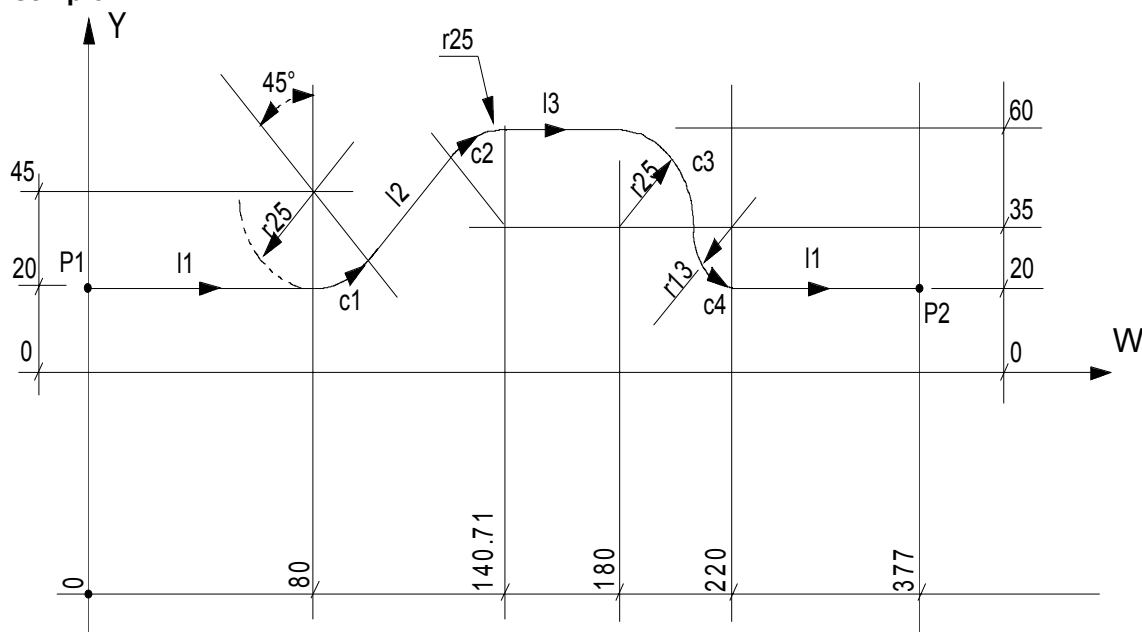
Caratteristiche:

Questo tipo di virtualizzazione viene utilizzata nel caso si voglia movimentare un asse lineare Y ed un rotativo B programmando le quote in un piano WY. Mentre i movimenti di Y virtuale coincidono con quelli di Y fisico, ogni spostamento di W corrisponde ad un arco di cerchio, funzione del raggio del cilindro, che deve essere tradotto in uno spostamento angolare dell'asse fisico B.



Lavorazione in coordinate cilindriche

Esempio:



(DIS,"ESEMPIO UVC");- Programmazione in coordinate cilindriche -

T1.1M6

S2000F300M3

B0

XY20Z10

E30=60;RAGGIO DEL CILINDRO

(UVC,B,W,E30)

G16WY

p1=W0Y20

E31=2*3.1415*E30

p2=WE31Y20

I1=p1,p2

c1=I80J45r25

c2=I140.71J35r-25

I2=c1,c2

c3=I180J35r-25

I3=c2,c3

c4=c3,I1,r15

G21G41p1

Z-12

I1

c1

I2

c2

I3

c3

c4

I1

G20G40p2

GZ20

(UVC)

M30



UVA - Programmazione di assi non ortogonali

Il comando UVA (USE VIRTUAL ANGULAR) permette la programmazione di assi non ortogonali trattandoli come se lo fossero.

Sintassi

(UVA,af1af2,av1av2,a)
(UVA)

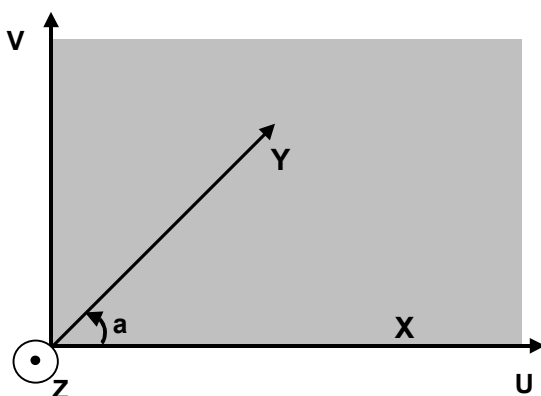
dove:

af1 E' il nome del primo asse fisico lineare (es. X)
af2 E' il nome del secondo asse fisico lineare (es. Y)
av1 E' il nome dell'asse virtuale ascissa (es. U)
av2 E' il nome dell'asse virtuale ordinata (es. V)
a E' l'angolo, in gradi, sotteso dai due semiassi positivi degli assi lineari fisici
nessun parametro (UVA) senza parametri disabilita la modalità UVA.

Caratteristiche:

Questo tipo di virtualizzazione viene utilizzata nel caso si vogliano movimentare due assi lineari X Y che non sono tra loro ortogonali programmando però le quote su un piano virtuale UV ortogonale. L'asse U, ascissa della nuova coppia di assi sarà coassiale all'asse X mentre l'asse V sarà perpendicolare a U, formante un angolo di $90^\circ - a$ con l'asse Y.

L'origine di UV sarà coincidente con quella di XY. La posizione del generico punto P di coordinate (U,V) nel piano virtuale viene tradotta nelle coordinate (X,Y) degli assi fisici. Mentre i movimenti di U coincidono con quelli dell'asse fisico X ogni spostamento di V viene tradotto in un movimento interpolato di X e Y.



Vista del piano XY e UV corrispondente

IMPORTANTE

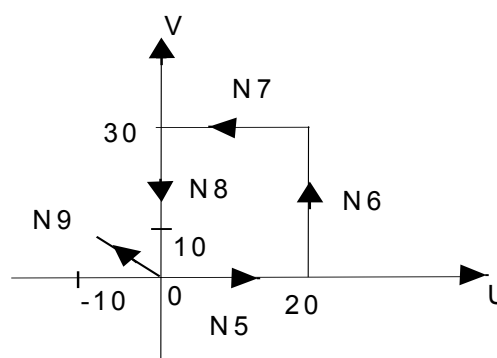
Il segno del parametro **a** è positivo se l'asse X, nel posizionarsi sull'asse Y, compie una rotazione in senso antiorario, negativo se il senso è orario (nella figura per es. **a** è positivo).

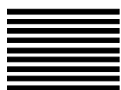
Non sono ammessi valori di **a** pari a 0° o 180° , ciò implicherebbe due assi lineari fisici paralleli.

Esempio di programmazione in coordinate polari

Esempio 1:

N1 T1.1M6
 N2 S1000M3
 N3 GX0Y0
 N4 (UVA,XY,UV,45)
 N5 G1U20VF110
 N6 V30
 N7 U0
 N8 V0
 N9 U-10V10
 N10 (UVA)
 N11 GX50





TCP - Tool Center Point per macchine con testa "Double Twist"

Questa prestazione permette la programmazione di una macchina a 5 assi (3 lineari e 2 rotativi) riferita alla punta dell'utensile piuttosto che al centro di rotazione degli assi (centro della testa).

La posizione controllata dal sistema dipende sia dalla posizione degli assi rotativi sia dalle caratteristiche geometriche della testa.

L'attivazione di questo algoritmo viene fatta direttamente nel programma tramite il triletterale TCP.

La modalità TCP viene disabilitata con **(TCP)** oppure RESET.

Programmando il triletterale minuscolo (tcp,n), è possibile variare alcuni parametri senza disabilitare l'algoritmo TCP e senza uscire dalla modalità di "MOVIMENTO CONTINUO".

Il triletterale (tcp,n) minuscolo deve essere usato solamente per variare i parametri del (TCP,n) programmato in precedenza, il parametro *n* deve essere lo stesso specificato in precedenza; (tcp,n) inoltre non può essere utilizzato nella modalità n=5.

Sintassi

(TCP[,n])
(tcp[,n])

dove:

n Tipo di compensazione desiderata (1÷5); nelle pagine seguenti vengono illustrate queste modalità.

Caratteristiche:

L'algoritmo di TCP necessita, per il suo funzionamento, di alcune informazioni relative all'utensile in uso.

E' possibile gestire in contemporanea una testa che monta sino a quattro diversi utensili.

La valutazione dei parametri di configurazione deve essere fatta dopo avere posizionato la testa seguendo le indicazioni angolari illustrate nelle figure 2.1, 2.2, 2.3.

Questi parametri devono essere inseriti nella "tabella user".

La compilazione della "tabella user" può avvenire tramite l'apposito editor, oppure da logica di macchina (tramite le apposite funzioni di logica), od ancora da part program tramite le variabili L.

E' possibile utilizzare una configurazione già predisposta del sistema contenuta nei file caricabili tramite il "SETUP" dello USER TABLE EDITOR.

NOTA:

Per maggiori informazioni riguardanti l'uso della "tabella user" da PLUS leggere la descrizione delle funzioni \$TBLPUTD e \$TBLGETD nel manuale della libreria PLUS.

Per l'utilizzo della stessa tabella da ambiente TABLE EDITOR consultare il manuale di Uso.

Tabella TCP per macchine con testa "Double Twist"

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato della variabile	TCP
Tipo compensazione	385	Record 97 - Var 1	L384	Definisce la modalita' di calcolo della compensazione in base alla configurazione di macchina. 0 - Abilita solo la parte dinamica del TCP, escludendo il calcolo geometrico. 1 - Non e' presente alcuna tavola mobile 2 - Tavola mobile sul primo asse 3 - Tavola mobile sul secondo asse 4 - Tavola mobile sul primo e secondo asse 5 - Riservato	1 2 3 5
Lunghezza utensile ^①	386	Record 97 - Var 2	L385	Definisce la lunghezza dell'utensile attualmente montato in macchina. Si intende, naturalmente, riferito al portautensile attivo.	1 2 5
Portautensile attivo ^①	387	Record 97 - Var 3	L386	Numero del portautensile attivo 1÷4.	1,2, 3,5
Parametro A ^① (figure 2.2 e 2.3)	388	Record 97 - Var 4	L387	Distanza (con segno) espressa in mm tra l'asse di rotazione della testa orizzontale e il piano parallelo contenente l'asse di rotazione della testa verticale	1 2 5
Offset testa Orizzontale ^①	389	Record 98 - Var 1	L388	E' il valore di offset, espresso in gradi, che si deve associare all'asse della testa orizzontale al fine di far coincidere il piano contenente l'utensile con il piano YZ della macchina. Per determinare tale offset ruotare l'asse orizzontale in modo tale che il piano che contiene l'utensile coincida con il piano YZ, e che dopo tale rotazione un movimento positivo della testa verticale direzioni l'utensile verso Y-. A questo punto prendere la quota a video, moltiplicare per il verso di rotazione della testa orizzontale (parametro seguente) e cambiare segno.	1 2 3 4 5
Verso testa orizzontale (figura 2.3) ^①	390	Record 98 - Var 2	L389	Definisce il verso di rotazione della testa orizzontale vista dall'alto e vale: +1 per verso orario - 1 per verso antiorario per programmazione positiva.	1 2 3 4 5
Verso testa vert. ^① (figura 2.2)	391	Record 98 - Var 3	L390	Definisce il verso di rotazione della testa verticale vista laterale sinistra e vale : +1 per verso orario - 1 per verso antiorario per programmazione positiva.	1 2 3 4 5
ID 1° asse lineare	392	Record 98 - Var 4	L391	ID dell'asse da utilizzare come asse ascissa nell'algoritmo di compensazione.	1 2 3 4,5

^① Parametro modificabile in continuo mediante programmazione minuscola (tcp,n).

Capitolo 2

Programmazione degli Assi

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato della variabile	TCP
ID 2° asse lineare	393	Record 99 - Var 1	L392	ID dell'asse da utilizzare come asse ordinata nell'algoritmo di compensazione.	1 2 3 4,5
ID 3° asse lineare	394	Record 99 - Var 2	L393	ID dell'asse da utilizzare come asse verticale nell'algoritmo di compensazione.	1 2 3 5
ID testa orizzontale	395	Record 99 - Var 3	L394	ID dell'asse su cui e' montata la testa orizzontale. Se ID nullo si intende adottare come inclinazione della testa il valore definito nel campo "Incl. testa oriz.".	1 2 3 5
ID testa verticale	396	Record 99 - Var 4	L395	ID dell'asse su cui e' montata la testa verticale. Se ID nullo si intende adottare come inclinazione della testa il valore definito nel campo "Incl. testa vert.".	1 2 3 5
Incl. testa orizzontale	397	Record 100 - Var1	L396	Inclinazione della testa orizzontale da utilizzare in alternativa all'asse omonimo (quando l'ID e' a 0). In tal senso si considerera' la testa orizzontale alla posizione cosi' specificata. E' un valore segnato, funzione del "verso testa orizzontale" configurato.	1 2 3 5
Incl. testa verticale	398	Record 100 - Var 2	L397	Inclinazione della testa verticale da utilizzare in alternativa all'asse omonimo (quando l'ID e' a 0). In tal senso si considerera' la testa verticale alla posizione cosi' specificata. E' un valore segnato, funzione del "verso testa verticale" configurato.	1 2 3 5
Raggio ^① (figura 2.1)	399	Record 100 - Var 3	L398	Definisce il raggio dell'utensile attualmente montato in macchina. Si riferisce, naturalmente, al portautensile attivo.	1 2 5
Angolo ^① (figura 2.1)	400	Record 100 - Var 4	L399	Definisce l'angolo sotteso tra il centro dell'utensile e il punto di contatto dell'utensile col pezzo dopo aver ruotato la testa verticale in modo tale che l'utensile sia orientato verso Y-.	1 2 3 5

^① Parametro modificabile in continuo mediante programmazione minuscola (tcp,n).

Portautensile 1

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B ① (figure 2.1, 2.3)	369	Record 93 - Var 1	L368	Distanza con segno espressa in mm tra il piano contenente l'utensile, direzionato verso Y-, ed il piano parallelo contenente l'asse di rotazione della testa orizzontale.	1 2 5
Parametro C ① (figure 2.1, 2.2)	370	Record 93 - Var 2	L369	Distanza con segno espressa in mm tra il piano contenente l'utensile ed il piano parallelo contenente l'asse di rotazione della testa verticale.	1 2 5
Parametro D ① (figure 2.2, 2.3)	371	Record 93 - Var 3	L370	Distanza con segno espressa in mm tra l'asse di rotazione della testa verticale e la pinza porta utensile.	1 2 5
Offset testa vert. ①	372	Record 93 - Var 4	L371	E' il valore di offset, espresso in gradi, che si deve associare all'asse della testa verticale al fine di disporre l'asse dell'utensile parallelamente alla direzione -Y della macchina. Per determinare tale offset posizionare l'utensile verso Y-, prendere la quota a video, moltiplicare per il verso di rotazione della testa verticale configurato in precedenza, e cambiare segno.	1 2 3 5

① Parametro modificabile in continuo mediante programmazione minuscola (tcp,n).

Portautensile 2

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B ① (figure 2.1, 2.3)	373	Record 94 - Var 1	L372	Come portautensile 1	1 2 5
Parametro C ① (figure 2.1, 2.2)	374	Record 94 - Var 2	L373	Come portautensile 1	1 2 5
Parametro D ① (figure 2.2, 2.3)	375	Record 94 - Var 3	L374	Come portautensile 1	1 2 5
Offset testa vert. ①	376	Record 94 - Var 4	L375	Come portautensile 1	1 2 3 5

Capitolo 2

Programmazione degli Assi

Portautensile 3

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B ^① (figure 2.1, 2.3)	377	Record 95 - Var 1	L376	Come portautensile 1	1 2 5
Parametro C ^① (figure 2.1, 2.2)	378	Record 95 - Var 2	L377	Come portautensile 1	1 2 5
Parametro D ^① (figure 2.2, 2.3)	379	Record 95 - Var 3	L378	Come portautensile 1	1 2 5
Offset testa vert. ^①	380	Record 95 - Var 4	L379	Come portautensile 1	1 2 3 5

Portautensile 4

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B ^① (figure 2.1, 2.3)	381	Record 96 - Var 1	L380	Come portautensile 1	1 2 5
Parametro C ^① (figure 2.1, 2.2)	382	Record 96 - Var 2	L381	Come portautensile 1	1 2 5
Parametro D ^① (figure 2.2, 2.3)	383	Record 96 - Var 3	L382	Come portautensile 1	1 2 5
Offset testa vert. ^①	384	Record 96 - Var 4	L383	Come portautensile 1	1 2 3 5

^① Parametro modificabile in continuo mediante programmazione minuscola (tcp,n).

Le figure che seguono illustrano i parametri di caratterizzazione della compensazione applicati ad una macchina schematizzata nelle viste frontale, laterale, e dall'alto.

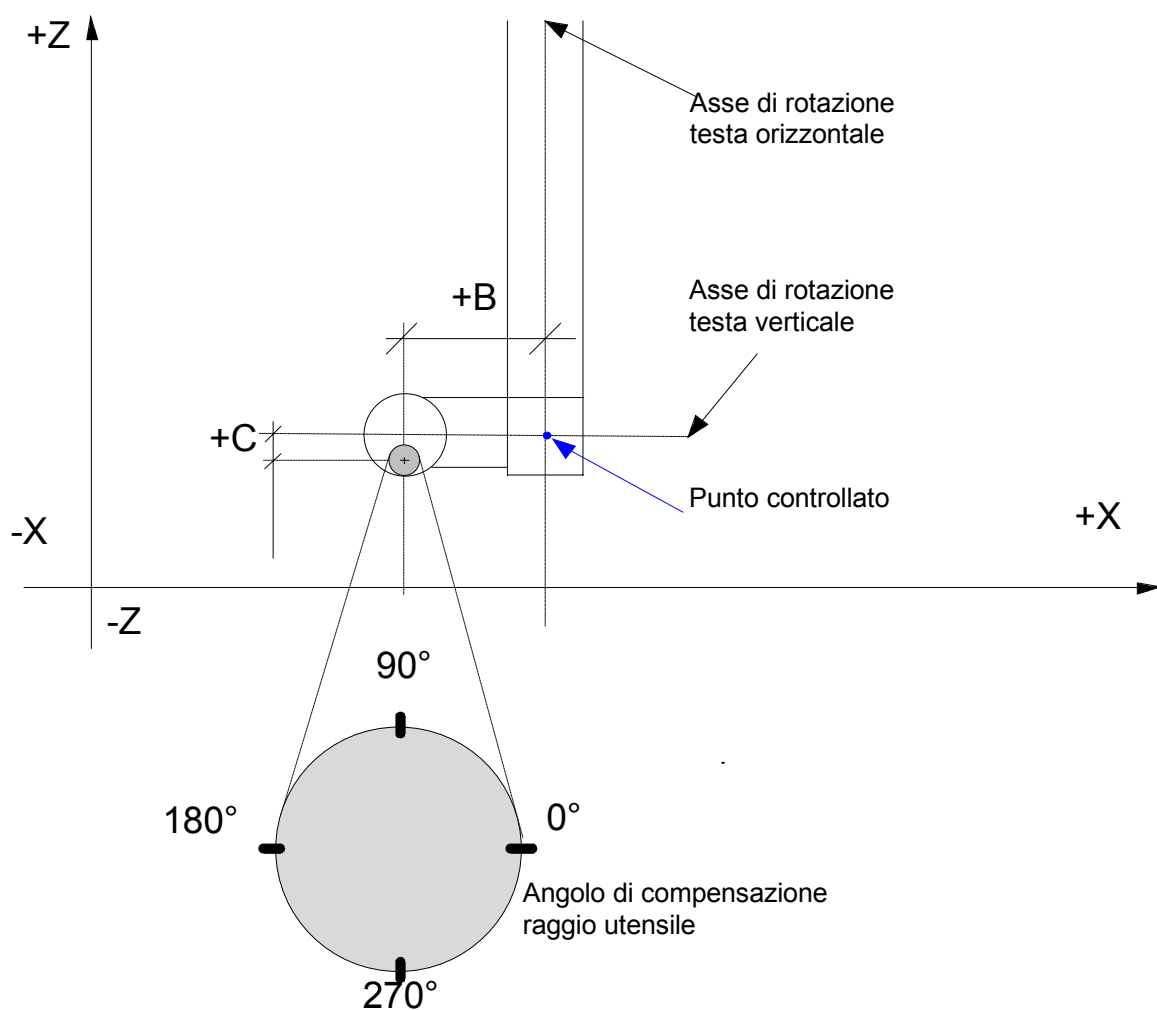


Fig. 2.1 Vista Frontale

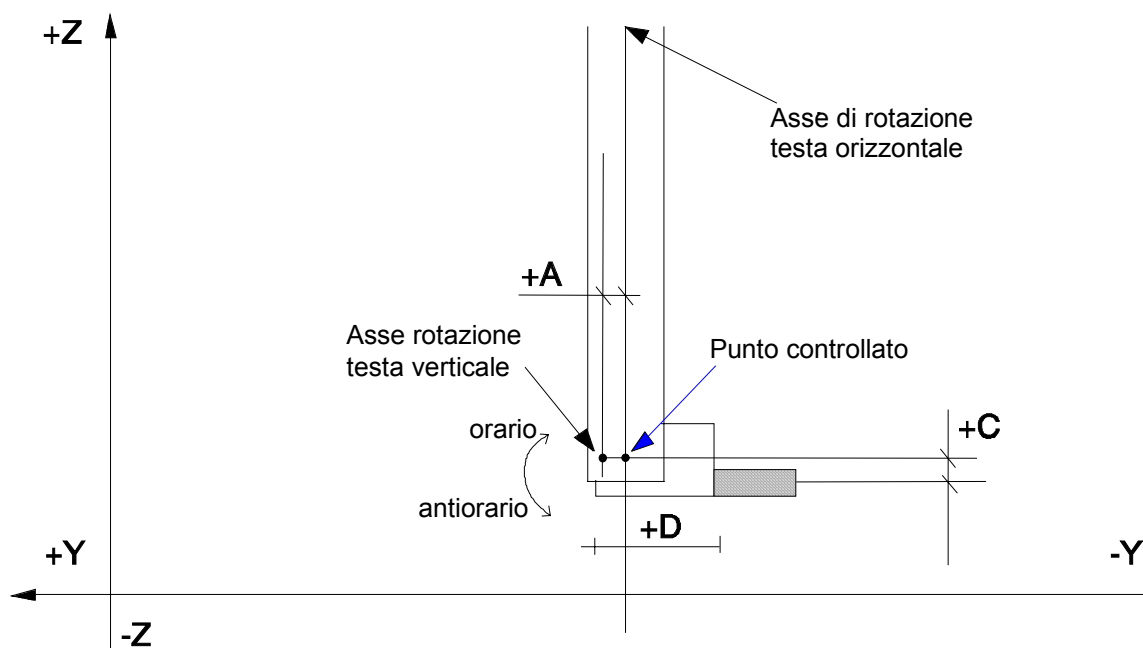


Fig 2.2 Vista laterale

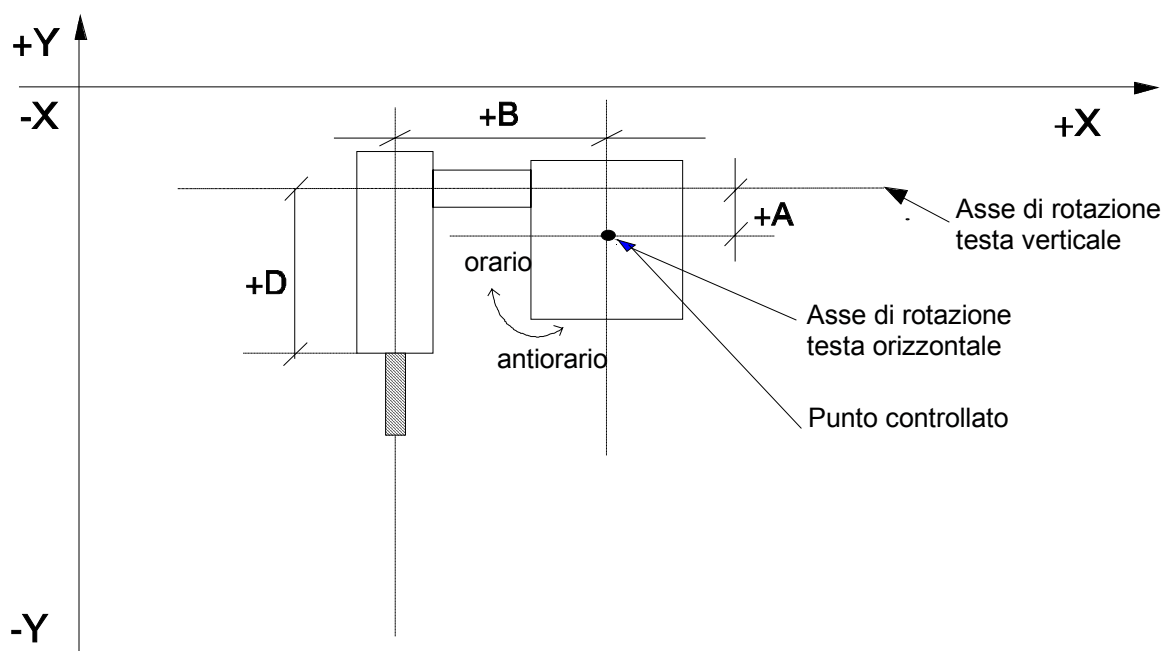


Fig 2.3 Vista dall'alto

Dinamica

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Modalità dinamica	357	Record 90 - Var 1	L356	<p>Questo campo può assumere i seguenti valori :</p> <p>0 - Le velocità e le accelerazioni degli assi lineari non vengono limitate, pertanto esse possono superare <i>run time</i> i valori configurati.</p> <p>1 - Le velocità degli assi lineari e rotativi vengono limitate <i>run time</i> , se necessario, in modo tale che esse non superino mai i valori configurati. Le accelerazioni degli assi lineari e rotativi vengono limitate a priori in modo tale che la sovrapposizione dei movimenti non richieda accelerazioni superiori a quelle configurate.</p> <p>2 - Le velocità e le accelerazioni degli assi lineari e rotativi vengono limitate a priori in modo tale che le velocità e le accelerazioni calcolate <i>run time</i> non superino mai quelle configurate.</p> <p>In questo caso se si muove, ad esempio, solo l'asse lineare, esso rimane comunque limitato contraria-mente a quanto accade nel caso 1.</p> <p>Va osservato che nel caso 2 viene garantita la velocità di costante sul profilo, cosa che non accade nel caso 1 se viene "tagliata" <i>run time</i> la velocità su un asse.</p>	1 2 3 5
Raggio dello spigolo (figure 2.5 e 2.6) ^①	358	Record 90 - Var 2	L357	Raggio dello spigolo utensile nel caso di utensili sferici o toroidali.	1,2 3,5
Angolo di contatto α (figure 2.5 e 2.6) ^①	359	Record 90 - Var 3	L358	Angolo α di contatto per utensili sferici o toroidali ($0 \leq \alpha \leq 90$).	1,2 3,5
Programmazione di "m", "n" ed "o"	360	Record 90 - Var 4	L359	<p>Definisce come deve essere gestito il contatto tra utensile e pezzo. Coi valori "0" ed "1" il controllo è definito tramite angoli la cui definizione avviene:</p> <p>0= tramite variabili T.U. 359 e 400 1= tramite m ed n da part program</p> <p>Con il valore 2, invece, il contatto è stabilito tramite un vettore normale al profilo definito nelle 3 componenti m, n, o.</p>	1 2 3 5

^① Parametro modificabile in continuo mediante programmazione minuscola (tcp,n).

Capitolo 2

Programmazione degli Assi

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Angolo di soglia (figura 2.4)	361	Record 91 - Var 1	L360	Angolo al di sopra del quale, a fine ente, viene inserito un movimento dell'asse rotativo orizzontale per posizionare l'utensile perpendicolarmente all'ente successivo. Al di sotto di questo angolo il medesimo movimento dell'asse rotativo orizzontale inizia invece a fine ente e prosegue con l'esecuzione dell'ente successivo.	2
Velocità rotativo	362	Record 91 - Var 2	L361	Velocità espressa in gradi/min. con cui ruotare l'asse rotativo orizzontale quando viene introdotto automaticamente il suo posizionamento tra due blocchi di part program. Se il parametro vale 0 viene utilizzata la feed programmata.	2
Modo interpolazione	363	Record 91 - Var 3	L362	Definisce se gli assi rotativi sono interpolati o meno insieme agli assi lineari. Nel caso in cui non sono interpolati insieme, essi inseguono il moto dei lineari. 0 - Interpolati insieme 1 - Inseguono i lineari E' utile tarare la modalità 1 nel caso in cui non si voglia che la velocità sul profilo sia influenzata dal movimento degli assi rotativi.	1 2 3 5
Tipo interpolazione	364	Record 91 - Var 4	L363	Definisce se gli assi rotativi devono essere interpolati di solo errore o se devono seguire la stessa metodologia degli assi lineari. 0 - Come gli assi lineari 1 - Interpolati di solo errore	1 2 3 5
Integratore	365	Record 92 - Var 1	L364	Definisce, per il modo di interpolazione 1, se utilizzare rampe di accelerazione/decelerazione per gli assi rotativi oppure se forzare le velocità proporzionali al moto degli assi lineari. 0 - Senza rampe 1 - Con rampe Nel caso 1 (con rampe) i diagrammi di velocità vengono calcolati in funzione dei parametri configurati per gli assi rotativi, indipendentemente dal moto degli assi lineari.	1 2 3 5

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Modalità offset	366	Record 92 - Var 2	L365	Definisce la modalità con la quale vengono attivati il correttore lunghezza e il raggio utensile (ed eventualmente anche il raggio dello spigolo utensile in base all variabile TTR). 0= valori presi da tabella TCP 1= valori con segno attivi nel sistema (lunghezza 1 e raggio utensile) 2= valori assoluti attivi nel sistema (lunghezza 1 e raggio utensile) 3= valore assoluto della sola lunghezza 1; il raggio utensile è considerato = 0	1 2 3 5
Movimento minimo	367	Record 92 - Var 3	L366	Definisce l'angolo minimo al di sopra del quale viene generato automaticamente il movimento dell'asse rotativo orizzontale per posizionare l'utensile perpendicolarmente all'ente successivo.	2
Modalità limiti operativi	368	Record 92 - Var 4	L367	Definisce la modalità con la quale effettuare il controllo dei limiti operativi configurati: 0= Il controllo viene effettuato sia sul blocco programmato (prima della sua esecuzione) sia in real time sui singoli punti. 1= Il controllo viene effettuato solo in real time sui singoli punti. NOTA: Il controllo eseguito sul blocco programmato è riferito alla punta utensile in quanto i part program sono riferiti ad esso. Il controllo eseguito in real time è riferito invece al centro di rotazione degli assi; se il sistema trova che il punto fuoriesce dal limite segnala errore e blocca il movimento degli assi decelerando.	1 2 3 4 5

NOTA:

Tutti i parametri espressi in mm, in realtà sono da intendersi nell'unità di misura configurata per la macchina.

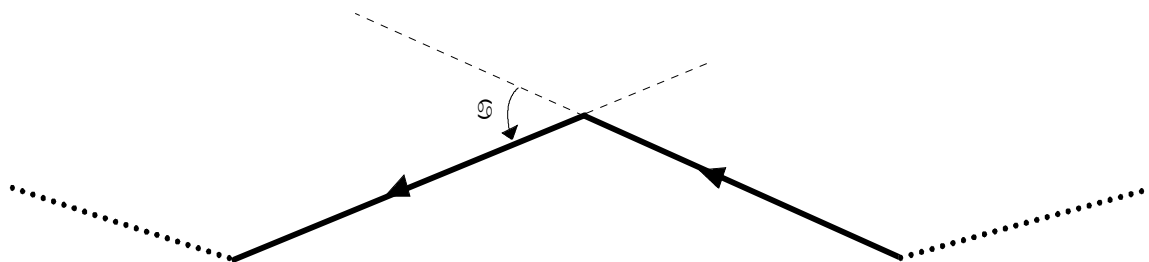


Fig. 2.4 Angolo di soglia (a)

Le figure seguenti illustrano il significato dei parametri TCP usati per la gestione di utensili sferici e toroidali; l'abbreviazione T.U. indica il numero corrispondente nella Tabella Utente.

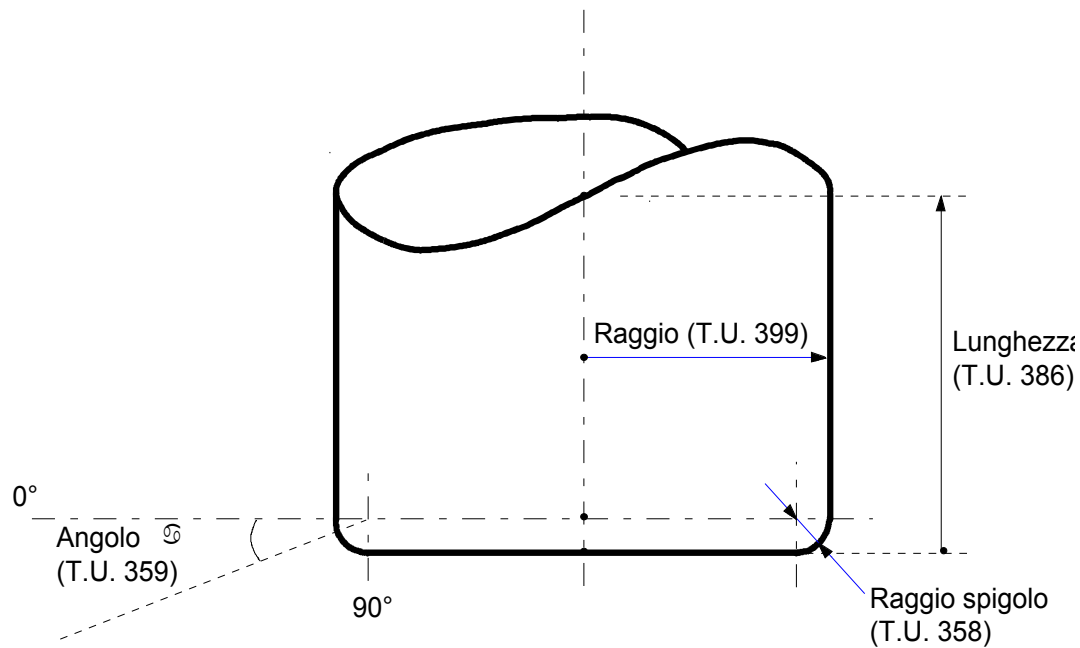


Fig. 2.5 Utensile di tipo toroidale

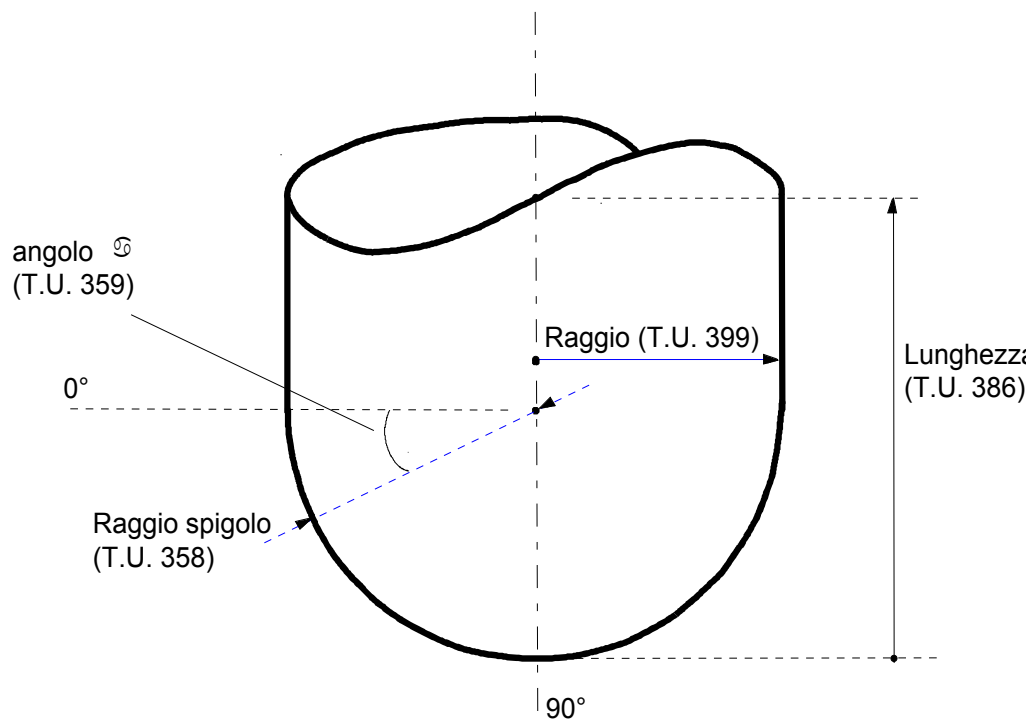


Fig. 2.6 Utensile di tipo sferico

- **Modalità 1 (TCP,1)**

Le quote programmate sono riferite al pezzo.

Questa modalità deve essere utilizzata quando i profili da eseguire interessano sia il movimento degli assi lineari (1÷3) sia quello degli assi rotativi (1÷2) e si vuole che l'utensile sia sempre a contatto con la superficie da lavorare. Inoltre le quote degli assi lineari sono riferite al pezzo.

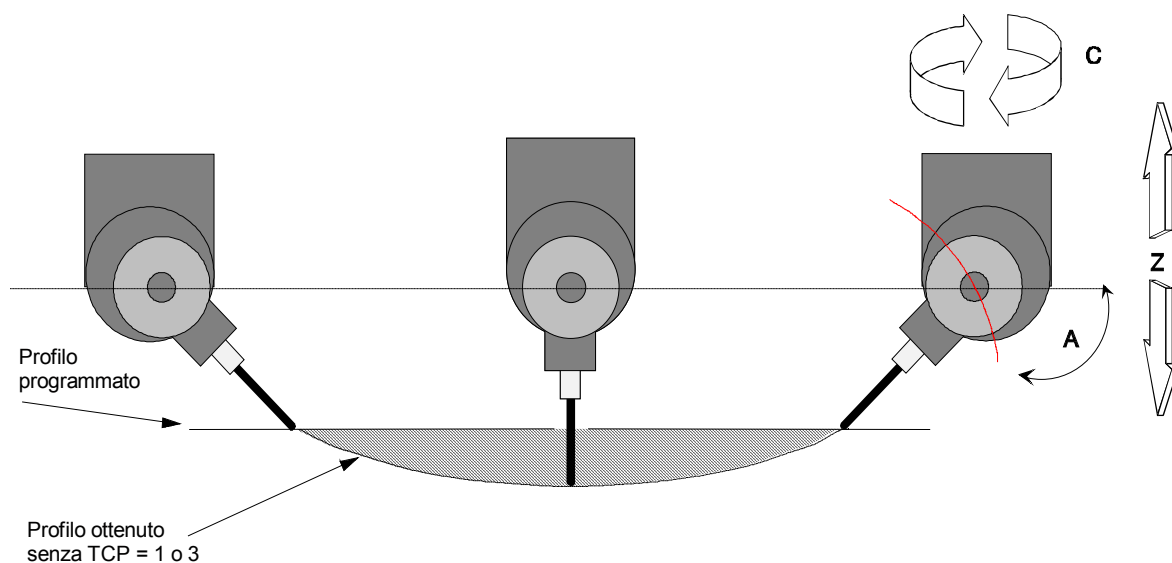
Il sistema, infatti, si assume il compito di compensare gli spostamenti adeguando automaticamente la posizione dei tre assi lineari in funzione degli assi rotativi in modo tale da mantenere la punta dell'utensile sul profilo definito degli assi lineari.

- **Modalità 3 (TCP,3)**

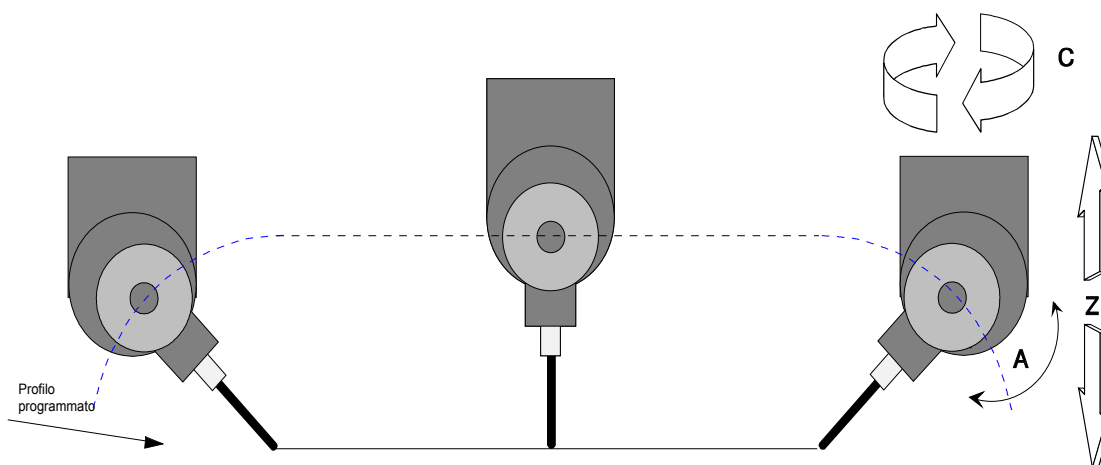
Tale modalità va usata quando si vogliono utilizzare programmi generati da sistemi CAD con utensili di lunghezza, raggio, raggio spigolo, e angoli di contatto differenti rispetto a quelli per cui è stato generato il programma (quote macchina).

Le variazioni di lunghezza e diametro vanno introdotte tramite la logica di macchina (vedi \$TCPWRT nel manuale della libreria PLUS).

Come per (TCP,1) il sistema si assume il compito di compensare gli spostamenti dovuti alle variazioni di lunghezza raggi ed angoli di contatto, in modo tale da mantenere la punta dell'utensile sul profilo definito.



Profilo ottenuto senza TCP 1 o 3 attivo



Profilo ottenuto con TCP 1 o 3 attivo.

• Modalità 2 (TCP,2)

La modalità (TCP,2) è una estensione del (TCP,1).

Oltre alle prestazioni del (TCP,1) essa automaticamente introduce il movimento dell'asse rotativo orizzontale in modo da mantenere costante la sua inclinazione lungo il profilo definito dai primi due assi lineari configurati nella tabella TCP.

Tale movimento viene generato:

- *tra due enti* per ruotare sugli spigoli
In questo caso, il movimento dell'asse rotativo, o viene creato tra la fine dell'ente in corso e l'inizio del successivo, oppure viene eseguito contemporaneamente all'ente successivo (vedi parametro "angolo di soglia").
- *su un ente circolare* per mantenere inclinazione costante su tutto l'ente
In questo caso il movimento viene eseguito contemporaneamente all'ente stesso.

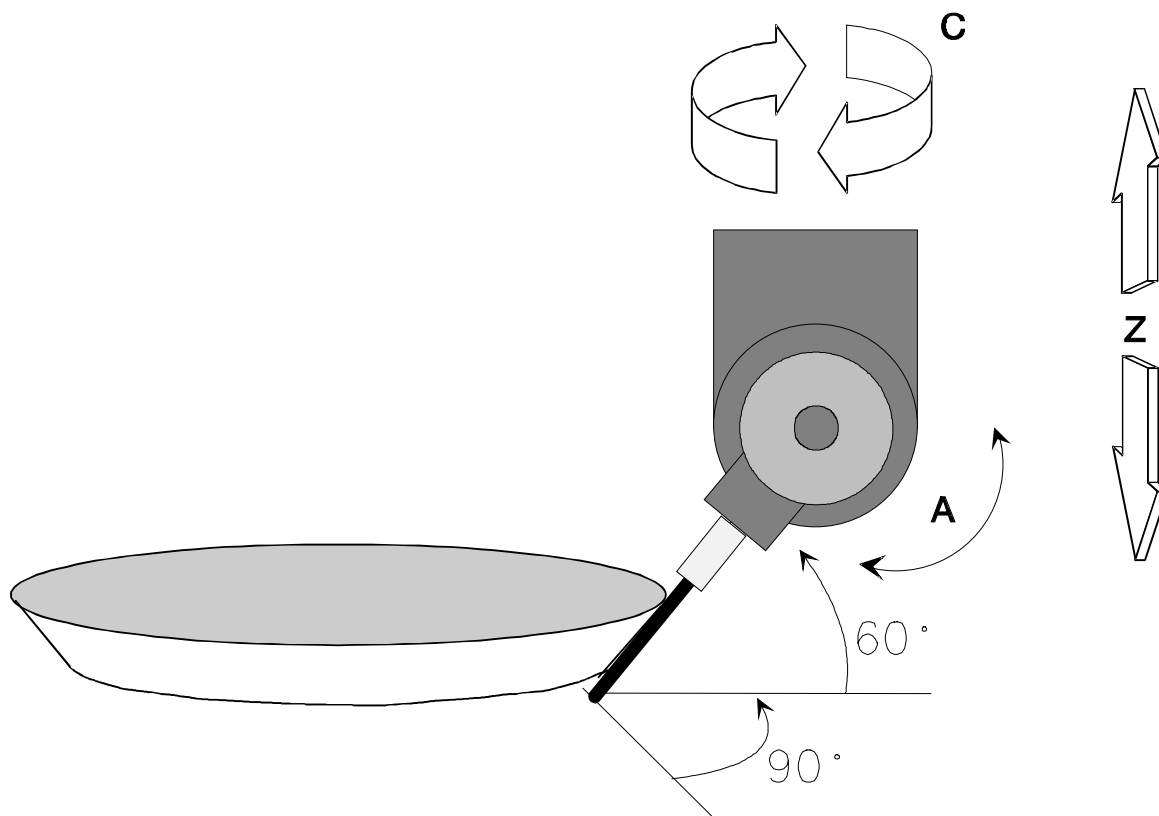
Queste considerazioni sono valide per tutti gli enti del profilo escluso il primo.

Infatti il primo ente va considerato come ente di attacco sul profilo (anche se è di tipo circolare non si ha movimento dell'asse rotativo). Il sistema inizierà a generare il movimento dell'asse rotativo orizzontale a partire dal secondo ente.

- **Modalità 4 (TCP,4)**

La modalità (TCP,4) serve a mantenere costante l'inclinazione dell'asse rotativo orizzontale lungo tutto il profilo definito dai primi due assi lineari configurati nella tabella TCP. A tal fine viene generato un adeguamento dell'asse rotativo orizzontale in modo continuo, ovvero tra i singoli punti interpolati (e non tra gli enti come TCP,2).

Questo comporta che il profilo non debba presentare discontinuità; in caso contrario potrebbero generarsi dei "SERVO ERROR".



Movimentazione con TCP 2 o 4 abilitata

IMPORTANTE

Il (TCP,4) NON ingloba le prestazioni del (TCP,1).

- **Modalità 5 (TCP,5)**

Questa modalità permette di eseguire movimenti lungo la direzione dell'utensile.

Alla sua attivazione viene generato dal sistema un asse virtuale, il cui nome deve essere stato precedentemente caratterizzato in AMP (sezione VIRTUAL AXES di PROCESS CONFIG).

La movimentazione dell'asse virtuale comporta un movimento degli assi lineari tale per cui l'utensile si sposta lungo la direzione da esso indicata, ovvero lungo la direzione individuata dalla posizione degli assi rotativi.

Può essere programmato singolarmente o in contemporanea agli altri assi; in tal caso le prestazioni date da (TCP,5) si sommano a quelle del (TCP,1); può inoltre essere mosso in manuale.

La modalità 5 può essere utile sia per eseguire fori inclinati sia per estrarre l'utensile dal pezzo in seguito al verificarsi di un'anomalia che ha provocato l'interruzione della lavorazione.

In questo secondo caso si possono avere 3 differenti situazioni:

A) Gli assi rotativi, all'atto dell'attivazione di (TCP,5), non sono riferiti e la precedente interruzione si è verificata durante una lavorazione che utilizzava una qualunque modalità TCP.

In questo caso il sistema individua automaticamente la direzione dell'utensile in base alle posizioni assunte dagli assi rotativi prima del verificarsi dell'anomalia.

Questo è possibile anche dopo un eventuale spegnimento del sistema essendo tali posizioni memorizzate in un'area di memoria non volatile. Va tenuto presente che i valori che individuano la posizione degli assi rotativi sono modificabili da logica tramite il data entry di MANUAL SETUP e tramite la funzione \$TCPWRT (funzione utile nel caso in cui si voglia variare la direzione di estrazione dell'utensile).

B) Gli assi rotativi, all'atto dell'attivazione di (TCP,5), non sono riferiti e la precedente interruzione si è verificata durante una lavorazione che NON utilizzava la modalità TCP.

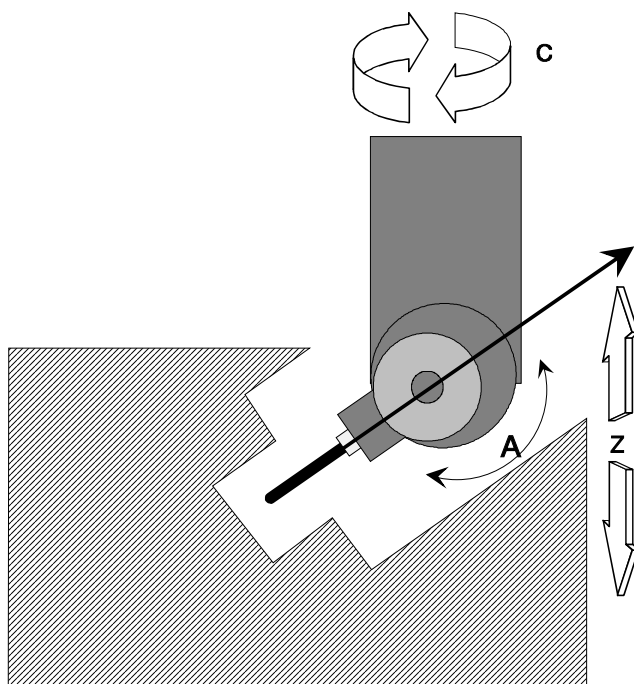
In questo caso è *necessario* comunicare al sistema la direzione dell'utensile, ovvero la posizione degli assi rotativi, in quanto eventuali posizioni memorizzate precedentemente dal sistema possono non essere significative. Le posizioni possono essere comunicate o tramite la funzione Plus \$TCPWRT o tramite il data-entry di MANUAL SETUP.

C) Gli assi all'atto dell'attivazione di (TCP,5) sono riferiti.

In questo caso, indipendentemente dal fatto che l'interruzione si sia verificata oppure no durante una lavorazione con modalità TCP, la direzione dell'utensile è data dalla posizione attuale degli assi rotativi (e non da quella degli assi al momento dell'interruzione, che comunque, in condizioni di lavoro normali, coincidono).

NOTA:

Per l'utilizzo della funzione \$TCPWRT o del data entry di MANUAL SETUP vedere i manuali LIBRERIA PLUS e MANUALE D'USO.



Uscita da ingombri tramite TCP,5

NOTA:

In tutte le modalità TCP, la logica di macchina può comunicare in tempo reale all'interpolatore una eventuale variazione della lunghezza e raggio utensile, del raggio spigolo, nonché degli angoli di contatto con il pezzo (parametri 386, 399, 400, 358 e 359 della tabella user).

In questo modo è possibile compensare l'usura dell'utensile durante la lavorazione (vedi funzione \$TCPWRT sul manuale della libreria PLUS).

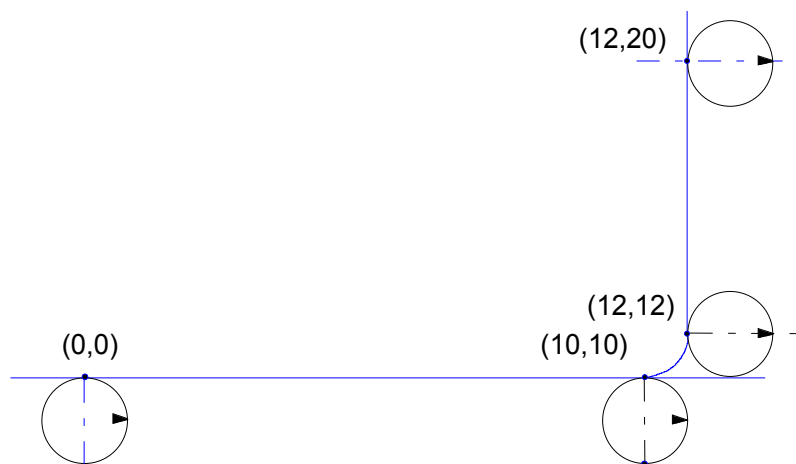
Programmazione dei parametri "m" ed "n" (angoli)

In tutte le modalità di utilizzo del TCP, precedentemente descritte, è possibile programmare oltre alle quote assi, anche i parametri "m" ed "n" i quali assumono il seguente significato:

- m angolo sotteso tra il centro utensile ed il punto di contatto dell'utensile con il pezzo (coincidente con il parametro T.U. 400). - vedere figure 2.1, 2.5 e 2.6.
- n angolo di contatto tra lo spigolo di taglio ed il pezzo (coincidente con il parametro 359) - vedere fig. 2.5 e 2.6 .

Tramite la programmazione di tali parametri (abilitata ponendo ad 1 la variabile T.U.360) è possibile variare di blocco in blocco il punto di contatto tra il pezzo e l'utensile.

Esempio:



```
G1X10Y0m270
(TCP,1)
G1XY
X10Y10
G3X12Y12I10J12m180
G1X12Y20
(TCP)
```

Programmazione dei parametri "m", "n", "o" e "u", "v", "w" (vettori)

In tutte le modalità di utilizzo del TCP precedentemente descritte è possibile programmare, oltre alle quote assi, anche i parametri "m", "n", "o" oppure "u", "v", "w" i quali rappresentano rispettivamente le 3 componenti del vettore normale al profilo da lavorare oppure i fattori di compensazione parassiale su cui applicare la compensazione del raggio utensile.

Tramite la programmazione di detti parametri è possibile variare di blocco in blocco il punto di contatto tra il pezzo e l'utensile, e tramite questo vettore ottenere la compensazione diametro utensile.

L'abilitazione della compensazione raggio utensile attraverso i parametri "m", "n", "o" avviene ponendo a 2 le variabili T.U. 360 e T.U. 366. Tali parametri vengono applicati al 1°, 2°, 3° asse configurato nella tabella TCP.

L'abilitazione della compensazione raggio utensile attraverso i parametri "u", "v", "w" avviene ponendo a 0 la variabile T.U. 360 e a 3 la variabile T.U. 366. Tali parametri vengono applicati al 1°, 2°, 3° asse configurato nel processo.

Esempio di programmazione con “m”, “n”, “o”: Esempio di programmazione con “u”, “v”, “w”

m..n..o.. ;valore iniziale del vettore

h1

(TCP,....) ;attivazione TCP

X..Y..Z.. AB m..n..o..

X..Y..Z.. AB m..n..o..

X..Z.. m..n..o..

A..B.. m..n..o..

X..Y..

.

.

.

.

.

(TCP)

h1

(TCP,....) ;attivazione TCP

X..Y..Z.. AB u..v..w..

X..Y..Z.. AB u..v..w..

X..Z..u..v..w..

A..B..u..v..w..

X..Y..

.

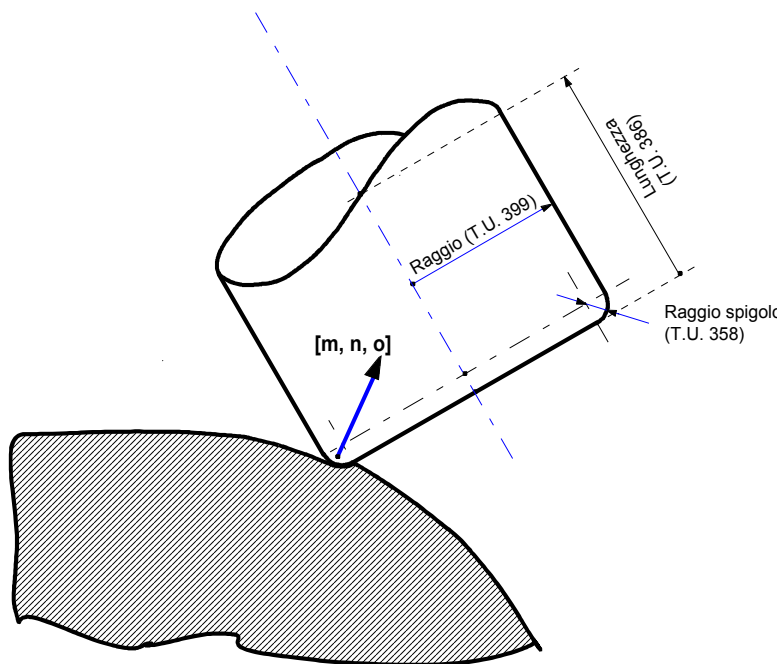
.

.

.

.

(TCP)





TCP - Tool Center Point per macchine generiche a 5 assi

Questa prestazione permette la programmazione di una macchina a 5 assi (3 lineari e 2 rotativi) indipendentemente dall'orientamento e dal tipo dell'utensile (sferico o toroidale). La posizione controllata dal sistema coincide con il punto di innesto dell'utensile nel mandrino (vedi figure 2.7 e 2.8).

L'attivazione di questo algoritmo viene fatta direttamente nel programma tramite il triletterale TCP.

La modalità TCP viene disabilitata con **(TCP)** oppure RESET.

Sintassi

(TCP[,n])

dove:

n Tipo di compensazione desiderata (1, 3, 5); nelle pagine seguenti vengono illustrate queste modalità.

Caratteristiche:

L'algoritmo di TCP necessita, per il suo funzionamento, di alcune informazioni relative all'utensile in uso, le quali devono essere inserite nella "Tabella User" - abbr. T.U.

La compilazione della "tabella user" può avvenire tramite l'apposito editor, oppure da logica di macchina (tramite le apposite funzioni di logica), od ancora da part program tramite le variabili L.

E' possibile utilizzare una configurazione già predisposta del sistema contenuta nei files caricabili tramite il "SETUP" dello USER TABLE EDITOR.

NOTA:

Per maggiori informazioni riguardanti l'uso della "tabella user" da PLUS leggere la descrizione delle funzioni \$TBLPUTD e \$TBLGETD nel manuale della libreria PLUS.

Per l'utilizzo della stessa tabella da ambiente TABLE EDITOR consultare il manuale d' Uso.

Le figure seguenti illustrano il significato dei parametri TCP usati per la gestione di utensili sferici e toroidali; l'abbreviazione T.U. indica il numero corrispondente nella Tabella Utente.

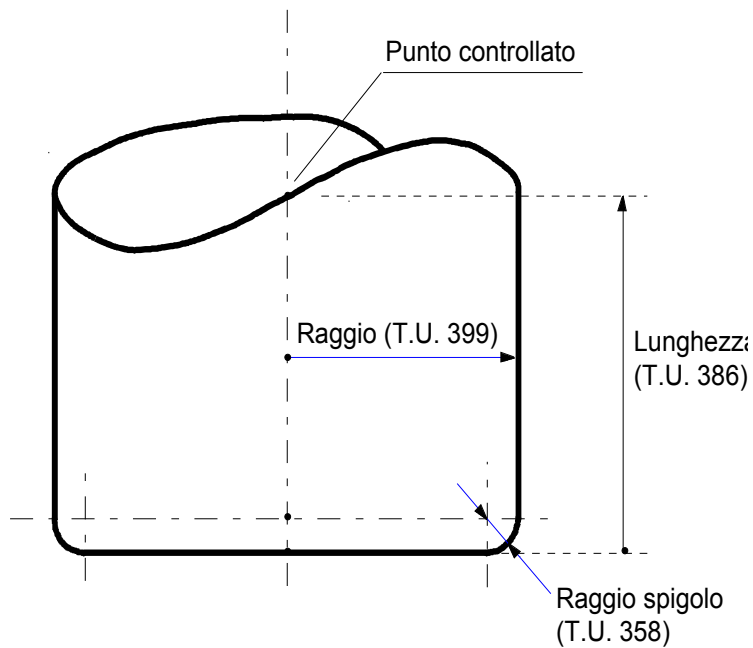


Fig. 2.7 Utensile di tipo toroidale

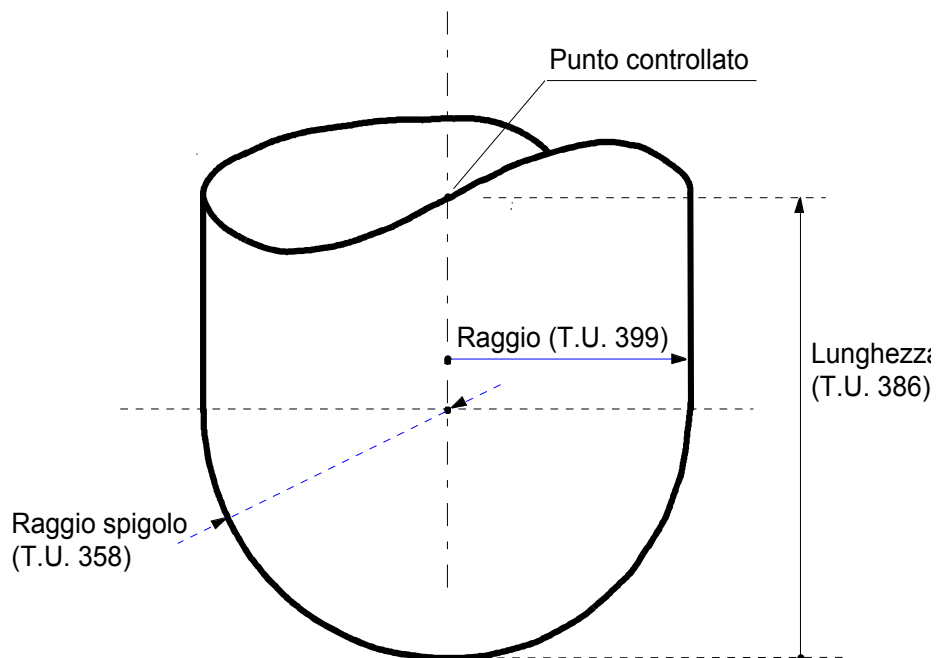


Fig. 2.8 Utensile di tipo sferico

Tabella TCP per macchine generiche a 5 assi

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Tipo compensazione	385	Record 97 - Var 1	L384	Definisce la modalita' di calcolo della compensazione in base alla configurazione di macchina. NOTA: Deve essere impostato con valore=6	1 3 5
Lunghezza utensile	386	Record 97 - Var 2	L385	Definisce la lunghezza dell'utensile attualmente montato in macchina.	1 5
Portautensile attivo	387	Record 97 - Var 3	L386	Non usato	
Parametro A	388	Record 97 - Var 4	L387	Non usato	
Offset testa Orizzontale	389	Record 98 - Var 1	L388	Non usato	
Verso testa orizzontale	390	Record 98 - Var 2	L389	Non usato	
Verso testa verticale	391	Record 98 - Var 3	L390	Non usato	
ID 1° asse lineare	392	Record 98 - Var 4	L391	ID dell'asse da utilizzare come asse ascissa nell'algoritmo di compensazione.	1 3 5
ID 2° asse lineare	393	Record 99 - Var 1	L392	ID dell'asse da utilizzare come asse ordinata nell'algoritmo di compensazione.	1 3 5
ID 3° asse lineare	394	Record 99 - Var 2	L393	ID dell'asse da utilizzare come asse verticale nell' algoritmo di compensazione	1 3 5
ID testa orizzontale	395	Record 99 - Var 3	L394	ID dell'asse su cui e' montata la testa orizzontale. Se l'ID è nullo L'asse non esiste.	1 3 5
ID testa verticale	396	Record 99 - Var 4	L395	ID dell'asse su cui e' montata la testa verticale. Se l'ID è nullo L'asse non esiste.	1 3 5
Incl. testa orizzontale	397	Record 100 Var 1	L396	Non usato	
Incl. testa verticale	398	Record 100 Var 2	L397	Non usato	
Raggio (figura 2.1)	399	Record 100 Var 3	L398	Definisce il raggio dell'utensile attualmente montato in macchina.	1 5
Angolo (figura 2.1)	400	Record 100 Var 4	L399	Non usato	

Dinamica

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Modalità dinamica	357	Record 90 - Var 1	L356	Questo campo può assumere i seguenti valori : 0 - Le velocità e le accelerazioni degli assi lineari non vengono limitate, pertanto esse possono superare <i>run time</i> i valori configurati. 1 - Le velocità degli assi lineari e rotativi vengono limitate <i>run time</i> , se necessario, in modo tale che esse non superino mai i valori configurati. Le accelerazioni degli assi lineari e rotativi vengono limitate a priori in modo tale che la sovrapposizione dei movimenti non richieda accelerazioni superiori a quelle configurate. 2 - Le velocità e le accelerazioni degli assi lineari e rotativi vengono limitate a priori in modo tale che le velocità e le accelerazioni calcolate <i>run time</i> non superino mai quelle configurate. In questo caso se si muove, ad esempio, solo l'asse lineare, esso rimane comunque limitato contrariamente a quanto accade nel caso 1. Va osservato che nel caso 2 viene garantita la velocità di costante sul profilo, cosa che non accade nel caso 1 se viene "tagliata" <i>run time</i> la velocità su un asse.	1 3 5
Raggio dello spigolo (figure 2.7 e 2.8)	358	Record 90 - Var 2	L357	Raggio dello spigolo utensile nel caso di utensili sferici o toroidali.	1 3 5
Angolo di contatto α	359	Record 90 - Var 3	L358	Non usato	
Programmazione di "m" ed "n"	360	Record 90 - Var 4	L359	Non usato	
Angolo di soglia	361	Record 91 - Var 1	L360	Non usato	
Velocità rotativo	362	Record 91 - Var 2	L361	Non usato	
Modo interpolazione	363	Record 91 - Var 3	L362	Definisce se gli assi rotativi sono interpolati o meno insieme agli assi lineari. Nel caso in cui non sono interpolati insieme, essi inseguono il moto dei lineari. 0 - Interpolati insieme 1 - Inseguono i lineari E' utile tarare la modalità 1 nel caso in cui non si voglia che la velocità sul profilo sia influenzata dal movimento degli assi rotativi.	1 3 5

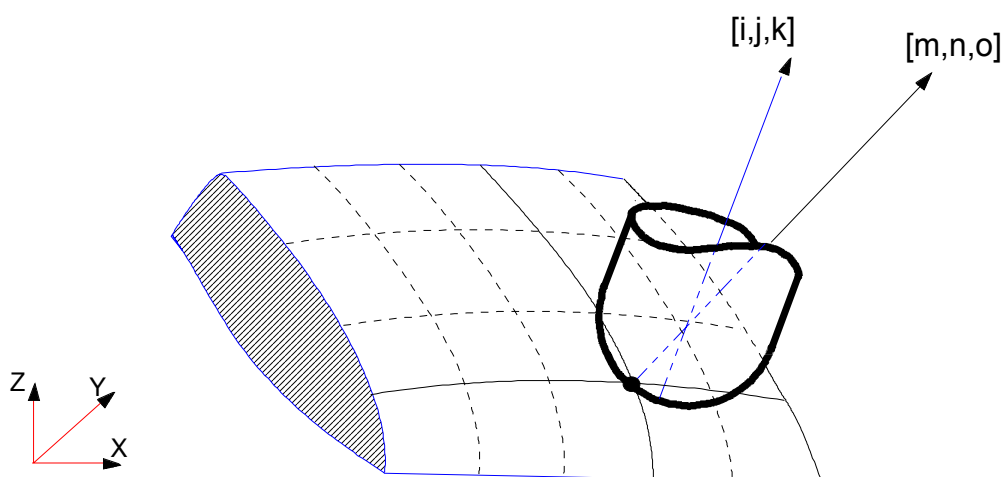
Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Tipo interpolazione	364	Record 91 - Var 4	L363	Definisce se gli assi rotativi devono essere interpolati di solo errore o se devono seguire la stessa metodo-logia degli assi lineari. 0 - Come gli assi lineari 1 - Interpolati di solo errore	1 3 5
Integratore	365	Record 92 - Var 1	L364	Definisce, per il modo di interpolazione 1, se utilizzare rampe di accelerazione/decelerazione per gli assi rotativi oppure se forzare le velocità proporzionali al moto degli assi lineari. 0 - Senza rampe 1 - Con rampe Nel caso 1 (con rampe) i diagrammi di velocità vengono calcolati in funzione dei parametri configurati per gli assi rotativi, indipendentemente dal moto degli assi lineari.	1 3 5
Modalità offset	366	Record 92 - Var 2	L365	Definisce la modalità con la quale vengono attivati il correttore lunghezza e il raggio utensile (ed eventualmente anche il raggio dello spigolo utensile in base alla variabile TTR). 0= valori presi da tabella TCP 1= valori con segno attivi nel sistema (lunghezza 1 e raggio utensile) 2= valori assoluti attivi nel sistema (lunghezza 1 e raggio utensile) 3= valore assoluto della sola lunghezza 1; il raggio utensile è considerato = 0	1 2 3 5
Movimento minimo	367	Record 92 - Var 3	L366	Non usato	
Modalità limiti operativi	368	Record 92 - Var 4	L367	Definisce la modalità con la quale effettuare il controllo dei limiti operativi configurati: = 0 Il controllo viene effettuato sia sul blocco programmato (prima della sua esecuzione) sia in real time sui singoli punti. = 1 Il controllo viene effettuato solo in real time sui singoli punti. NOTA: Il controllo eseguito sul blocco programmato è riferito alla punta utensile in quanto i part program sono riferiti ad esso. Il controllo eseguito in real time è riferito invece al centro di rotazione degli assi; se il sistema trova che il punto fuoriesce dal limite segnala errore e blocca il movimento degli assi decelerando.	1 3 5

Programmazione

Con questo tipo di TCP, per tutte le modalità di seguito descritte, è necessario definire sia la direzione (orientamento) dell'utensile che la direzione "normale" alla superficie da lavorare.

Ad ogni punto programmato sono associati 2 vettori, uno normale alla superficie da lavorare (parametri m, n, o) ed uno che rappresenta la direzione utensile (parametri i, j, k).

La programmazione dei parametri m, n, o, ed i, j, k deve avvenire in ordine alfabetico (corretti: mno, mo, no, ijk, ik, etc.; scorretti: nmo, on, om, jik, ki, etc.)



Sintassi

[XYZ] [AB] [m n o] [i j k]

dove:

XYZAB movimento assi

mno vettore normale alla superficie da lavorare.

ijk vettore che rappresenta la direzione utensile.

Caratteristiche:

Tutti i parametri sono opzionali; se non vengono programmati, essi confermano la programmazione precedente.

Prima dell'attivazione del TCP è necessario definire la posizione iniziale dei due vettori, programmandone i valori (questo blocco non provoca nessun movimento).

Con TCP attivo invece non è possibile programmare m n o e i j k senza un movimento degli assi XYZAB, inoltre il vettore da essi definito non deve essere nullo ([0,0,0]).

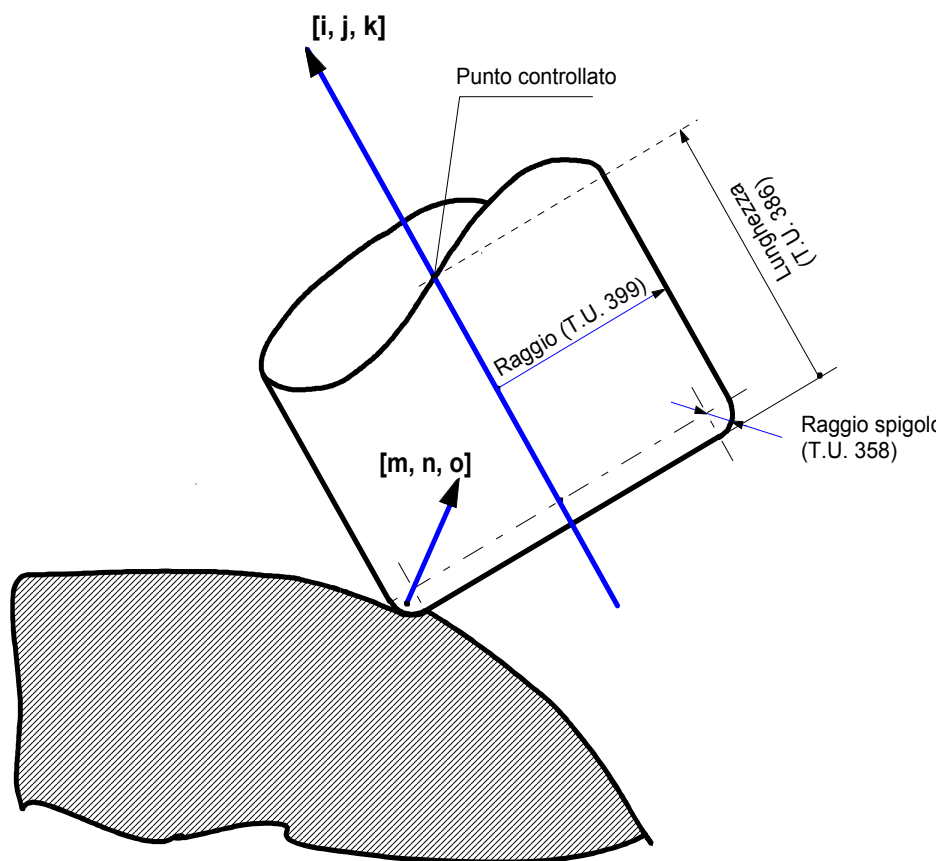
NOTA:

Il RESET azzerà i vettori

Esempio:

```

m..n..o.. i..j..k..           ;valore iniziale dei vettori
(TCP,.....)                 ;attivazione TCP
X..Y..Z.. A..B.. m..n..o.. i..j..k..
X..Y..Z.. A..B.. m..n..o..
X..Z.. m..n..o.. i..j..k..
A..B.. m..n..o..
X..Y.. i..j..k..
.
.
.
.
.
(TCP)
    
```



- **Modalità 1 (TCP,1)**

Le quote programmate sono riferite al pezzo.

Questa modalità deve essere utilizzata quando i profili da eseguire interessano sia il movimento degli assi lineari (1÷3) sia quello degli assi rotativi (1÷2) e si vuole che l'utensile sia sempre a contatto con la superficie da lavorare. Inoltre le quote degli assi lineari sono riferite al pezzo.

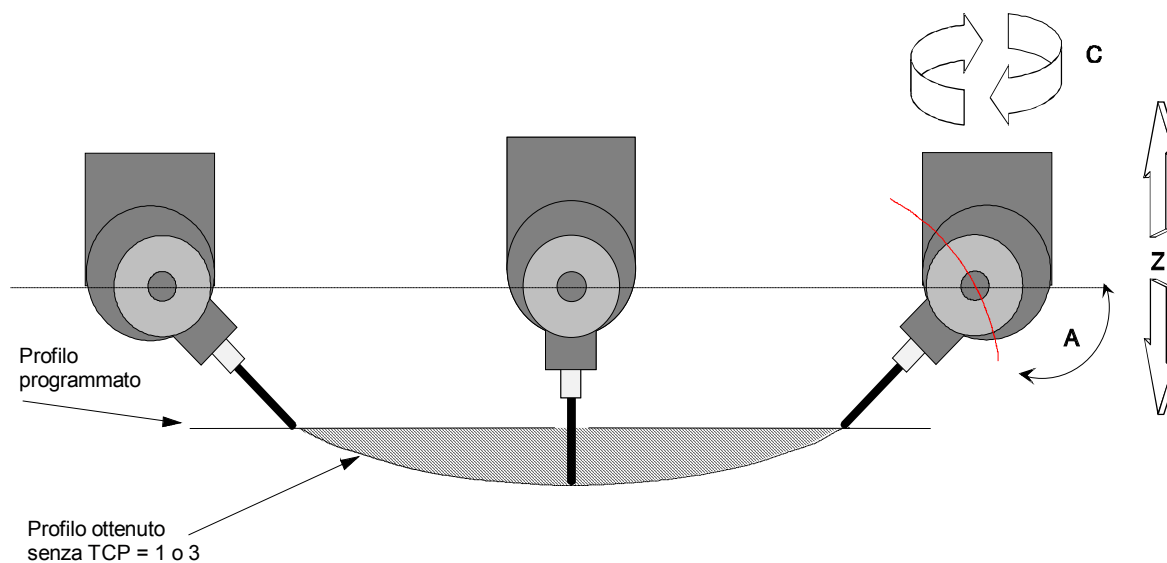
Il sistema, infatti, si assume il compito di compensare gli spostamenti adeguando automaticamente la posizione dei tre assi lineari in funzione degli assi rotativi in modo tale da mantenere il punto di contatto dell'utensile sul profilo definito degli assi lineari.

- **Modalità 3 (TCP,3)**

Tale modalità va usata quando si vogliono utilizzare programmi generati da sistemi CAD con utensili di lunghezza, raggio e raggio spigolo differenti rispetto a quelli per cui è stato generato il programma (quote macchina).

Le variazioni di lunghezza, raggio e raggio spigolo vanno introdotte tramite la logica di macchina (vedi \$TCPWRT nel manuale della libreria PLUS).

Come per (TCP,1) il sistema si assume il compito di compensare gli spostamenti dovuti alle variazioni di lunghezza e raggi in modo tale da mantenere la punta dell'utensile sul profilo definito.



Profilo ottenuto senza TCP 1 o 3 attivo

• Modalità 5 (TCP,5)

Questa modalità permette di eseguire movimenti lungo la direzione dell'utensile.

Alla sua attivazione viene generato dal sistema un asse virtuale, il cui nome deve essere stato precedentemente caratterizzato in AMP (sezione VIRTUAL AXES di PROCESS CONFIG).

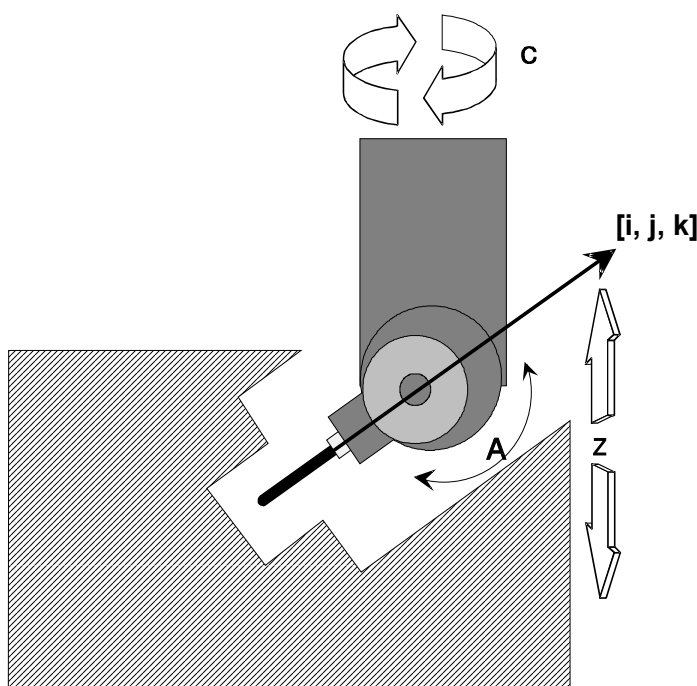
La movimentazione dell'asse virtuale comporta un movimento degli assi lineari tale per cui l'utensile si sposta lungo la direzione da esso indicata, ovvero lungo la direzione individuata dal vettore $[i, j, k]$.

Può essere programmato singolarmente o in contemporanea agli altri assi; in tal caso le prestazioni date da (TCP,5) si sommano a quelle del (TCP,1); può inoltre essere mosso in manuale.

La modalità 5 può essere utile per eseguire fori inclinati.

NOTA:

Per l'utilizzo della funzione \$TCPWRT o del data entry di MANUAL SETUP vedere i manuali LIBRERIA PLUS e MANUALE D'USO.



Uscita da ingombri tramite TCP,5

NOTA:

In tutte le modalità TCP, la logica di macchina può comunicare in tempo reale all'interpolatore una eventuale variazione della lunghezza, del raggio utensile o del raggio dello spigolo (parametri 386, 399 e 358 della tabella user).

In questo modo è possibile compensare l'usura dell'utensile durante la lavorazione (vedi funzione \$TCPWRT sul manuale della libreria PLUS).



TCP - Tool Center Point per macchine con utensile fisso e tavola rotante

Questa prestazione consente una facile programmazione su macchine in cui l'utensile è in posizione fissa ed il pezzo si muove per mezzo di tre assi, due lineari ed uno rotativo solidale ai due lineari. Il sistema determina in real time, in base alla posizione dell'asse rotativo, i punti che gli assi devono raggiungere per soddisfare il profilo programmato. È anche possibile far determinare al sistema la posizione dell'asse rotativo e quindi definire il profilo nel piano definito dai due assi lineari. L'attivazione di questo algoritmo viene fatta direttamente nel programma tramite il triletterale TCP.

La modalità TCP viene disabilitata con **(TCP)** oppure RESET.

Sintassi

(TCP[,n])

dove:

n Tipo di compensazione desiderata (1, 2,)

Caratteristiche:

L'algoritmo di TCP necessita, per il suo funzionamento, di alcune informazioni relative all'utensile in uso, le quali devono essere inserite nella "Tabella User " - abbr. T.U.

La compilazione della "tabella user" può avvenire tramite l'apposito editor, oppure da logica di macchina (tramite le apposite funzioni di logica), od ancora da part program tramite le variabili L.

E' possibile utilizzare una configurazione già predisposta del sistema contenuta nei files caricabili tramite il "SETUP" dello USER TABLE EDITOR.

NOTA:

Per maggiori informazioni riguardanti l'uso della "tabella user" da PLUS leggere la descrizione delle funzioni \$TBLPUTD e \$TBLGETD nel manuale della libreria PLUS.

Per l'utilizzo della stessa tabella da ambiente TABLE EDITOR consultare il Manuale d' Uso.

Tabella TCP per macchine con utensile fisso e tavola rotante

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Tipo compensazione	385	Record 97 - Var 1	L384	Definisce la modalità di calcolo della compensazione in base alla configurazione di macchina. NOTA: Deve essere impostato con valore=8	1 2
Lunghezza utensile	386	Record 97 - Var 2	L385	Non usato	
Portautensile attivo	387	Record 97 - Var 3	L386	Numero del portautensile attivo $1 \div 4$	1 2
Parametro A	388	Record 97 - Var 4	L387	Non usato	
Offset testa Orizzontale	389	Record 98 - Var 1	L388	Non usato	
Verso testa orizzontale	390	Record 98 - Var 2	L389	Definisce il verso di rotazione dell'asse rotativo (tavola) e vale: + 1 per verso orario - 1 per verso antiorario per programmazione positiva	1 2
Verso testa verticale	391	Record 98 - Var 3	L390	Non usato	
ID 1° asse lineare	392	Record 98 - Var 4	L391	ID dell'asse da utilizzare come asse ascissa nell'algoritmo di compensazione.	1 2
ID 2° asse lineare	393	Record 99 - Var 1	L392	ID dell'asse da utilizzare come asse ordinata nell'algoritmo di compensazione.	1 2
ID 3° asse lineare	394	Record 99 - Var 2	L393	Non usato	1 2
ID testa orizzontale	395	Record 99 - Var 3	L394	ID dell'asse rotativo (tavola)	1 2
ID testa verticale	396	Record 99 - Var 4	L395	Non usato	
Incl. testa orizzontale	397	Record 100 Var 1	L396	Non usato	
Incl. testa verticale	398	Record 100 Var 2	L397	Non usato	
Raggio	399	Record 100 Var 3	L398	Non usato	
Angolo	400	Record 100 Var 4	L399	Definisce l'angolo sotteso tra il centro dell'utensile e il punto di contatto dell'utensile col pezzo dopo aver ruotato la testa verticale in modo tale che l'utensile sia orientato verso Y-	1 2

Capitolo 2

Programmazione degli Assi

Portautensile 1

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B	369	Record 93 - Var 1	L368	Distanza con segno espressa in mm del centro tavola dalla punta utensile rispetto all'asse ascissa.	1 2
Parametro C	370	Record 93 - Var 2	L369	Distanza con segno espressa in mm del centro tavola dalla punta utensile rispetto all'asse ordinata.	1 2
Parametro D	371	Record 93 - Var 3	L370	Non usato.	
Offset testa vert.	372	Record 93 - Var 4	L371	Non usato	

Portautensile 2

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B	373	Record 94 - Var 1	L372	Come portautensile 1	1 2
Parametro C	374	Record 94 - Var 2	L373	Come portautensile 1	1 2
Parametro D	375	Record 94 - Var 3	L374	Non usato.	
Offset testa vert.	376	Record 94 - Var 4	L375	Non usato.	

Portautensile 3

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B	377	Record 95 - Var 1	L376	Come portautensile 1	1 2
Parametro C	378	Record 95 - Var 2	L377	Come portautensile 1	1 2
Parametro D	379	Record 95 - Var 3	L378	Non usato.	
Offset testa vert.	380	Record 95 - Var 4	L379	Non usato.	

Portautensile 4

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Parametro B	381	Record 96 - Var 1	L380	Come portautensile 1	1 2
Parametro C	382	Record 96 - Var 2	L381	Come portautensile 1	1 2
Parametro D	383	Record 96 - Var 3	L382	Non usato	
Offset testa vert.	384	Record 96 - Var 4	L383	Non usato	

Dinamica

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Modalità dinamica	357	Record 90 - Var 1	L356	Questo campo può assumere i seguenti valori : 0 - Le velocità e le accelerazioni degli assi lineari non vengono limitate, pertanto esse possono superare <i>run time</i> i valori configurati. 1 - Le velocità degli assi lineari e rotativi vengono limitate <i>run time</i> , se necessario, in modo tale che esse non superino mai i valori configurati. Le accelerazioni degli assi lineari e rotativi vengono limitate a priori in modo tale che la sovrapposizione dei movimenti non richieda accelerazioni superiori a quelle configurate. 2 - Le velocità e le accelerazioni degli assi lineari e rotativi vengono limitate a priori in modo tale che le velocità e le accelerazioni calcolate <i>run time</i> non superino mai quelle configurate. In questo caso se si muove, ad esempio, solo l'asse lineare, esso rimane comunque limitato contrariamente a quanto accade nel caso 1. Va osservato che nel caso 2 viene garantita la velocità di costante sul profilo, cosa che non accade nel caso 1 se viene "tagliata" <i>run time</i> la velocità su un asse.	1 2
Raggio dello spigolo	358	Record 90 - Var 2	L357	Non usato	
Angolo di contatto α	359	Record 90 - Var 3	L358	Non usato	
Programmazione di "m", "n" ed "o"	360	Record 90 - Var 4	L359	Non usato	
Angolo di soglia	361	Record 91 - Var 1	L360	Angolo al di sopra del quale, a fine ente, viene inserito un movimento dell'asse rotativo orizzontale per posizionare l'utensile perpendicolarmente all'ente successivo. Al di sotto di questo angolo il medesimo movimento dell'asse rotativo orizzontale inizia invece a fine ente e prosegue con l'esecuzione dell'ente successivo.	2
Velocità rotativo	362	Record 91 - Var 2	L361	Velocità espressa in gradi/min. con cui ruotare l'asse rotativo orizzontale quando viene introdotto automaticamente il suo posizionamento tra due blocchi di part program. Se il parametro vale 0 viene utilizzata la feed programmata.	2

Capitolo 2

Programmazione degli Assi

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Modo interpolazione	363	Record 91 - Var 3	L362	Definisce se gli assi rotativi sono interpolati o meno insieme agli assi lineari. Nel caso in cui non sono interpolati insieme, essi inseguono il moto dei lineari. 0 - Interpolati insieme 1 - Inseguono i lineari E' utile tarare la modalità 1 nel caso in cui non si voglia che la velocità sul profilo sia influenzata dal movimento degli assi rotativi.	1 2
Tipo interpolazione	364	Record 91 - Var 4	L363	Definisce se gli assi rotativi devono essere interpolati di solo errore o se devono seguire la stessa metodologia degli assi lineari. 0 - Come gli assi lineari 1 - Interpolati di solo errore	1 2
Integratore	365	Record 92 - Var 1	L364	Definisce, per il modo di interpolazione 1, se utilizzare rampe di accelerazione /decelerazione per gli assi rotativi oppure se forzare le velocità proporzionali al moto degli assi lineari. 0 - Senza rampe 1 - Con rampe Nel caso 1 (con rampe) i diagrammi di velocità vengono calcolati in funzione dei parametri configurati per gli assi rotativi, indipendentemente dal moto degli assi lineari.	1 2
Modalità offset	366	Record 92 - Var 2	L365	Non usato	
Movimento minimo	367	Record 92 - Var 3	L366	Definisce l'angolo minimo al di sopra del quale viene generato automaticamente il movimento dell'asse rotativo orizzontale per posizionare l'utensile perpendicolarmente all'ente successivo.	2

Variabile	Tabella utente	Record tabella user PLUS	Part Program	Significato	TCP
Modalità limiti operativi	368	Record 92 - Var 4	L367	<p>Definisce la modalità con la quale effettuare il controllo dei limiti operativi configurati:</p> <p>0= Il controllo viene effettuato sia sul blocco programmato (prima della sua esecuzione) sia in real time sui singoli punti.</p> <p>1= Il controllo viene effettuato solo in real time sui singoli punti.</p> <p>NOTA: Il controllo eseguito sul blocco programmato è riferito alla punta utensile in quanto i part program sono riferiti ad esso. Il controllo eseguito in real time è riferito invece al centro di rotazione degli assi; se il sistema trova che il punto fuoriesce dal limite segnala errore e blocca il movimento degli assi decelerando.</p>	1 2 3

NOTA:

Tutti i parametri espressi in mm, in realtà sono da intendersi nell'unità di misura configurata per la macchina.

Programmazione

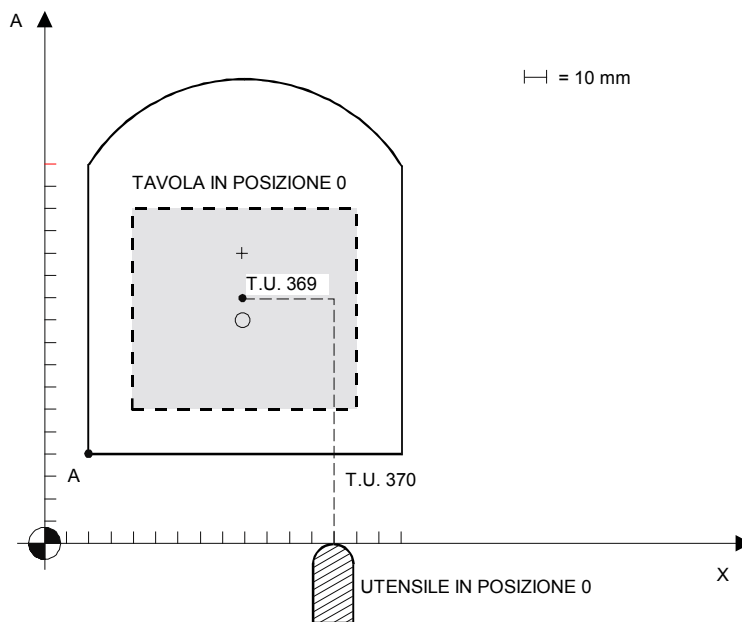
La virtualizzazione della tavola rotante consente la programmazione su macchine con l'utensile in posizione fissa ed il pezzo che si muove per mezzo di tre assi, due lineari ed uno rotativo solidale ai due lineari. Essa viene attivata tramite i comandi (TCP, 1) oppure (TCP, 2).

Programmando (TCP, 2) si ottiene l'inserimento automatico dell'asse rotativo. Le prestazioni e le caratteristiche di (TCP, 1) e (TCP, 2) descritte per macchine a 5 assi sono valide anche per questo tipo compensazione.

Per una corretta programmazione su macchine con tavola rotante, vanno fatte le seguenti considerazioni:

- il sistema considera di avere, alla fine del ciclo di homing, punta utensile e centro tavola coincidenti. Nel caso in cui questa situazione non si verifichi, i valori che rappresentano la distanza del centro utensile dalla punta utensile vanno definiti nella tabella TCP (T.U.369 e T.U. 370)
- il profilo va definito rispetto ad una origine coincidente con il centro tavola, pertanto, così come descritto al punto precedente, se centro tavola e punta utensile non coincidono, tale origine deve essere definita con i due valori ascissa e ordinata uguali agli spiazamenti definiti in tabella TCP (T.U. 369 e T.U. 370).

Esempio:



T.U. 369 = - 40
T.U. 370 = + 110

Distanza, rispetto ad X, del centro tavola dalla punta utensile
Distanza, rispetto ad Y, del centro tavola dalla punta utensile

ORIGINE 1 (PUNTO O) X-40 Y+110
(UOT,1,X-70,Y-70) ; PUNTO A
GX-10Y-10
(TCP,2)
G1 Y0 F2000
X+140
Y+130
G3 X0 Y130 I70 J90
G1Y-10
(TCP)

TCP su multiprocesso

La prestazione TCP può funzionare sui primi 4 processi del sistema. Ogni processo usa una propria area di configurazione all'interno delle tabelle User secondo il seguente schema :

Processo	Tabella	Recorda tabella da Plus	Variabile L Part Program
1	Tabella TCP	Record 97	L384 ÷ L399
	Portautensile 1	Record 93	L368 ÷ L371
	Portautensile 2	Record 94	L372 ÷ L375
	Portautensile 3	Record 95	L376 ÷ L379
	Portautensile 4	Record 96	L380 ÷ L383
	Dinamica	Record 90	L356 ÷ L367
2	Tabella TCP	Record 86	L340 ÷ L355
	Portautensile 1	Record 82	L324 ÷ L327
	Portautensile 2	Record 83	L328 ÷ L331
	Portautensile 3	Record 84	L332 ÷ L335
	Portautensile 4	Record 85	L336 ÷ L339
	Dinamica	Record 79	L312 ÷ L323
3	Tabella TCP	Record 75	L296 ÷ L311
	Portautensile 1	Record 71	L280 ÷ L283
	Portautensile 2	Record 72	L284 ÷ L287
	Portautensile 3	Record 73	L288 ÷ L291
	Portautensile 4	Record 74	L292 ÷ L295
	Dinamica	Record 68	L268 ÷ L279
4	Tabella TCP	Record 64	L252 ÷ L267
	Portautensile 1	Record 60	L236 ÷ L239
	Portautensile 2	Record 61	L240 ÷ L243
	Portautensile 3	Record 62	L244 ÷ L247
	Portautensile 4	Record 63	L248 ÷ L251
	Dinamica	Record 57	L224 ÷ L235

FINE CAPITOLO

PROGRAMMAZIONE DI UTENSILI E CORRETTORI

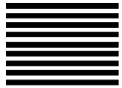
Questo capitolo tratta la programmazione di utensili e correttori; ricordiamo che le funzioni descritte devono essere gestite dalla logica di macchina.



L'installatore del sistema è responsabile dell'interfaccia tra il controllo e la macchina nonché delle funzioni T ed M specifiche per l'applicazione e del funzionamento dei codici di riqualifica e di presetting: RQT e RQP

Consultare la documentazione specifica della macchina utensile per ulteriori informazioni relative alla funzione T.

Ulteriori informazioni correlate si possono trovare nel capitolo 6 (Funzioni Ausiliarie M).



Indirizzo T per programmazione utensile

L'indirizzo T definisce l'utensile con il relativo correttore utilizzato per una determinata lavorazione.

Sintassi

T [*utensile*] [.] [*correttore*]

dove:

utensile E' il numero dell'utensile. Si può programmare il numero dell'utensile in modo esplicito (con un numero intero), oppure in modo implicito (con una variabile locale o di sistema).

correttore E' il numero del correttore da applicare.
Può essere un numero intero, oppure un parametro E.

Caratteristiche:

I valori programmabili per queste funzioni variano da 0.0 a 999999999999.300, dove le 15 cifre hanno il seguente significato:

999999999999.300

└─ numero del correttore dell'utensile

└─ codice dell'utensile

I seguenti esempi illustrano diversi modi di programmare gli utensili e i correttori.

T1	Seleziona l'utensile 1 e il correttore definito nella tabella degli utensili.
T1.1	Seleziona l'utensile 1 e il correttore 1.
T1.0	Seleziona l'utensile 1 senza correttore.
T.0	Disattiva il correttore dell'utensile attivo.
T0	Disattiva l'utensile attivo e relativo correttore.
T.1	Attiva il correttore 1 per l'utensile attivo



Indirizzo T per programmazione multiutensile

L'indirizzo T definisce più utensili da utilizzare contemporaneamente per una determinata lavorazione.

Sintassi

T [*master*] [...] [*correttore*] [/({ *slave* I} [*primo_slave*, *ultimo_slave*] { *variabile*, *num_variabili* })]

dove:

<i>master</i>	E' il numero di un utensile. Può essere un numero intero od una variabile locale o di sistema e varia da 0 a 999999999999.
<i>correttore</i>	E' il numero del correttore associato all'utensile master. Può essere un numero intero od una variabile locale o di sistema e varia da 0 a 300.
<i>slave</i>	E' il numero di un utensile. Può essere un numero intero od una variabile locale o di sistema.
<i>primo_slave</i>	E' un numero di un utensile e rappresenta il primo di una serie di utensili. Può essere un numero intero oppure una variabile locale o di sistema.
<i>ultimo_slave</i>	E' un numero di un utensile e rappresenta l'ultimo di una serie di utensili. Può essere un numero intero oppure una variabile locale o di sistema.
<i>variabile</i>	E' una variabile locale o di sistema contenente il primo di una sequenza di utensili.
<i>num_variabile</i>	E' un numero intero od una variabile locale o di sistema e rappresenta il numero di variabili da considerare a partire da " <i>variabile</i> "

Caratteristiche:

La programmazione multiutensile è utilizzata su macchine foratrici.

La gestione degli utensili associati al codice T è demandata alla logica di macchina a cui sono inviati i valori degli utensili programmati.

I valori programmabili per i codici utensili "slave" variano da 0 a 65535. Inoltre il numero massimo di utensili slave programmabili contemporaneamente è 60.

Come si nota dalla sintassi del codice T la lista degli utensili da usare si può specificare in tre formati diversi:

1. Formato singolo

Esempi:

T1.2/ 50	utensili	1, 50
	correttore	2
T1.2 /20,33,45,46	utensili	1,20,33,45,46
	correttore	2

2. Formato sequenza numerica

Semplifica la programmazione di più utensili con codice progressivo.

Esempi:

T1.3 /[30, 35]	equivale a T1.3 /30,31,32,33,34,35
T1.3 /[56, 51]	equivale a T1.3 /56,55,54,53,52,51
T1.3 /[50, 52], [10,13]	equivale a T1.3 /50,51,52,10,11,12,13

Come si nota il numero dell'utensile iniziale può essere > o < dell'utensile finale; nel primo caso i codici utensili sono considerati in senso crescente, nel secondo caso in senso decrescente

3. Formato di sequenza di variabili

Si basa su un array di variabili dal quale prelevare i numeri utensili.

Esempi:

E0 = 1, 30, 45	carica in E0, E1, E2 i valori 1, 30, 45
T1.2 /{ E0, 3 }	equivale a T1.2 /E0, E1, E2 ed a T1.2 /1, 30, 45
E0 = 1, 30, 45	carica in E0, E1, E2, E3 i valori 1, 30, 45
SN0 = 4, 77	carica in SN0, SN1 i valori 4, 77
SN4 = 3	carica in SN4 il valore 3
T1.2 /{ E0, SN4 }, { SN0,2 }	equivale a T1.2 /E0, E1, E2, SN0, SN1 ed a T1.2 /1,30,45,4,77

I tre formati precedenti possono anche essere combinati tra di loro.

Esempio:

E0 = 29, 56	carica in E0, E1 i valori 29, 56
SN6 = 2	carica in SN6 il valore 2
T1.3 /[7, 10], 15, { E0, SN6 }	equivale a T1.3 /7,8,9,10,15,29,56



Indirizzo h

L'indirizzo h permette l'introduzione del correttore sia durante il movimento punto a punto (G29) sia durante il movimento in continuo (G27 - G28).

Sintassi

h [*correttore*]

dove:

correttore Numero del correttore da applicare; può essere un numero intero di valore compreso tra 0 e 300 oppure una variabile locale o di sistema.

Caratteristiche:

L'indirizzo "h" deve essere programmato da solo nel blocco di part program.

La programmazione dell'indirizzo "h" disabilita eventuali valori di correzione attivi riferiti al comando "T".

Gli assi di applicazione dei valori sono gli stessi ai quali si fa riferimento con il comando

"T *utensile.correttore*".

I valori associati a *correttore* vengono applicati agli assi quando l'esecuzione del part program incontra l'indirizzo "h".

L'indirizzo "h" non richiede di essere sincronizzato né con la logica né con i movimenti.

IMPORTANTE

Il fatto di non sincronizzare "h" implica che la sua visualizzazione viene effettuata nel momento in cui viene letto dal part program e non quando è attuata.

Lo stesso avviene nel caso di utilizzazioni sincrone di "h", tipo G96. In tal caso occorre sincronizzare con "#" l'applicazione di "h" per avere il corretto funzionamento dei giri mandrino.

Programmare "h" senza *correttore* o con *correttore* =0 significa disattivare l'eventuale valore attivato con un precedente comando "h *correttore*".

Il RESET disabilita i valori di correzione applicati con l'indirizzo "h".

IMPORTANTE

E' compito della logica di macchina di decidere se applicare o meno dei valori di correzione nella procedura di RESET.

IMPORTANTE

L'utilizzo nello stesso part program di "h" e T per l'introduzione correttori può provocare problemi se non gestito correttamente dalla logica di macchina. Si consiglia pertanto di utilizzare una soltanto delle due modalità.

La visualizzazione del numero e del valore di correzione viene effettuata nei campi del video normalmente utilizzati per l'indirizzo T.

IMPORTANTE

L'introduzione dei correttori tramite *h* è possibile solo se la tabella dei correttori è utilizzata nel modo standard (configurazione di default).

I campi della tabella correttori (offset table) dai quali sono prelevati i correttori sono i seguenti:

Valore:	Campi offset table:
Length 1	Length 1+ Curr. requalif. length 1
Length 2	Length 2+ Curr. requalif. length 2
Diameter	Diameter+ Curr. requalif. Diameter

Esempio 1:

.

T[n] .x Programmazione di utensile e correttore

.

hy Annullamento del correttore **x** dichiarato in precedenza ed attivazione del correttore **y**

.

.

Esempio 2:

.

.

.

hy Attivazione correttore **y**

.

.

T[n] .x Annullamento del correttore **y** attivato con la **h**

.

.

Esempio 3:

.

.

.

hx

.

.

hy Annullamento del correttore **x** programmato in precedenza ed attivazione del correttore **y**.

.

.



TTR – Thoroidal Tool Radius

Questa variabile di processo permette di abilitare la gestione del raggio spigolo utensile nel caso di utensili sferici o toriodali.

Sintassi

TTR = valore

dove:

valore E' un valore che può essere:

- | | |
|---|--|
| 0 | Disabilita la gestione del raggio spigolo utensile |
| 1 | Abilita la gestione |

NOTA

Nel momento in cui viene abilitata la gestione del raggio spigolo utensile, il campo lunghezza 2 (più lunghezza di riqualifica e Max lunghezza) sono utilizzati dal sistema per definire il raggio della punta utensile. In tal senso la lunghezza non verrà più applicata come correttore lunghezza.

Il raggio spigolo utensile viene usato solo all'interno del TCP e della HSM a fronte della programmazione di un T o di una h.



AXO - Definizione del correttore dell'asse

Questo comando associa i valori lunghezza contenuti nella tabella dei correttori con gli assi della macchina.

Sintassi

(AXO, [-] nomeasse1 [, [-] nomeasse2])

dove:

nome asse 1: Nome dell'asse a cui è associata la lunghezza 1 del correttore utensile memorizzato nella tabella utensili/correttori. Se il segno "-" è inserito davanti al nome, il valore della lunghezza correttore viene applicato all'asse con segno opposto a quello presente in tabella.

nome asse 2: Nome dell'asse a cui è associata la lunghezza 2 del correttore utensile memorizzato nella tabella utensili/correttori. Se il segno "-" è inserito davanti al nome, il valore della lunghezza correttore viene applicato all'asse con segno opposto a quello presente in tabella.

Caratteristiche:

L'associazione tra correttore lunghezza e l'asse del processo su cui applicarlo viene effettuata di norma nella fase di configurazione del sistema, tramite la pagina INFORMAZIONI GENERALI SULL'ASSE dell'ambiente AMP.

Questa associazione può essere cambiata con il triletterale AXO.

I valori di preset (iniziali) contenuti nelle tabelle dei correttori hanno sempre un segno positivo. Il comando AXO consente di associare questi valori agli assi con segno negativo.

Seguono due esempi di attivazione del correttore lunghezza, il primo senza l'uso del triletterale "AXO", ed il secondo con l'uso del triletterale "AXO".

Esempio 1:

.

N1 T1.4 M6 Attiva la lunghezza 1 del correttore 4 sull'asse associato alla lunghezza 1 del correttore in fase di caratterizzazione.

.

N100 T0 M6 Disattiva il valore del correttore lunghezza sull'asse.

Esempio 2:

.

N1 (AXO,-X,Z) Associa: X al valore 1 del correttore lunghezza con segno negativo Z al valore 2 del correttore lunghezza con segno positivo

.

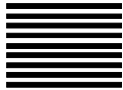
N50 T1.4 M6 Attiva i valori lunghezza 1 e 2 del correttore 4 sugli assi X e Z definiti nel comando AXO. Il valore lunghezza 1 è applicato all'asse X con segno negativo.

.

N100 T0 M6 Disattiva il valore di correzione lunghezza sugli assi X e Z.

IMPORTANTE

Il RESET del sistema, oppure l'attivazione di un altro part program ripristina l'associazione asse/correttore lunghezza caratterizzata con l'AMP.



RQT (RQU) - Riqualifica dei correttori utensili attuali

Il comando RQT riqualifica la lunghezza e/o il diametro memorizzati nella tabella dei correttori.

Sintassi

(RQT,utensile, correttore [,L..] [,I..][,d..])

dove:

utensile E' il numero dell'utensile.

correttore E' il numero del correttore da riqualificare. Il numero del correttore è un valore compreso tra 1 e 300.

L E' il valore da aggiungere alla lunghezza 1 del correttore.

I E' il valore da aggiungere alla lunghezza 2 del correttore.

d E' l'incremento del correttore da aggiungere al diametro.

Caratteristiche:

I valori possono essere dei numeri o contenuti nelle variabili.

Occorre specificare gli incrementi della lunghezza e del diametro del comando RQT nell'unità di misura corrente (pollici o millimetri con G70/G71). I valori non possono avere un fattore di scala (SCF) associato.

Esempi:

(RQT,10,1,L E40 ,d E41) Questo blocco riqualifica il correttore 1 dell'utensile 10. L'incremento della lunghezza 1 è contenuto in E40 mentre l'incremento del diametro è contenuto in E41.

(RQT,10,1,L E50 ,I E51) Questo blocco riqualifica il correttore 1 dell'utensile 10. L'incremento della lunghezza 1 è contenuto in E50 mentre l'incremento della lunghezza 2 è contenuto in E51.



RQP - Riqualifica dei correttori utensili (valori di presetting)

Il comando RQP esegue il presetting del correttore selezionato. Quando il controllo esegue il comando, scrive i corrispondenti valori di lunghezza e diametro nella tabella dei correttori. Tutti i valori di correzione correnti vengono azzerati.

Sintassi

(RQP,utensile,correttore [,L..][,I..][,d..])

dove:

utensile E' il numero dell'utensile.

correttore E' il numero del correttore da presetare. Il numero del correttore è un valore compreso tra 1 e 300. Il numero massimo dipende dal numero di record dichiarato nel file dei correttori.

L E' il valore da scrivere nella lunghezza 1 del correttore.

I E' il valore da scrivere nella lunghezza 2 del correttore.

d E' il valore da scrivere nel campo diametro del correttore.

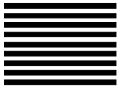
Caratteristiche:

I valori possono essere dei numeri o contenuti nelle variabili.

Occorre specificare i valori della lunghezza e del diametro del comando RQP nell'unità di misura corrente (pollici o millimetri con G70/G71). I valori non possono avere un fattore di scala (SCF) associato.

Esempi:

(RQP,10,1,L E40 ,d E41) Questo blocco presetta il correttore 1 dell'utensile 10. Il valore della lunghezza 1 è contenuto in E40 mentre il valore del diametro è contenuto in E41.



TOU (TOF) - Dichiarazione di utensile scaduto

Tramite il comando TOU è possibile dichiarare "scaduto" l'utensile specificato.

Sintassi

(TOU,*utensile*)

dove:

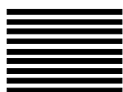
utensile E' il numero dell'utensile da dichiarare scaduto (EXPIRED). Può essere una variabile locale o di sistema.

Esempio:

(TOU,5) ;Dichiara scaduto l'utensile 5 nella tabella utensili

E1=10 ;

(TOU,10) ;Dichiara scaduto l'utensile 10 nella tabella utensili



LOA - Caricamento delle tabelle in memoria

Il comando LOA permette di caricare in memoria dual port la tabella specificata residente su disco.

Sintassi

(LOA, *nometabella* [*.estensione*][*,par1*][*,par2*]))

dove:

nome tabella E' il nome del file corrispondente alla tabella da caricare in memoria dual port.

estensione E' l'estensione del nome file ed indica il tipo di tabella da caricare:

.TOL	tabella utensili
.OFS	tabella correttori
.SPN	tabella mandrini (usata dall'opzione WOOD)
.ORG	tabella origini
.USR	tabella variabili user
.MAG	tabella magazzini

Nel caso in cui non esista nessuna estensione, la tabella considerata è quella relativa all'opzione Electronic Cam per la cui descrizione si rimanda al manuale dedicato.

Nel caso sia stata definita l'associazione utensile-correttore sul caricamento della tabella utensili vengono caricati automaticamente anche i correttori associati.

Il nome della tabella con la relativa estensione può essere definito anche tramite la variabile SC.

par 1 E' un parametro che assume significato diverso in funzione del tipo di tabella da caricare:

Tabella utensili ⇒ *par 1* è il numero del magazzino a cui gli utensili appartengono ed è obbligatorio nel caso di utilizzo dei magazzini.

Tabella origini ⇒ *par 1* è il numero di processi a cui appartengono gli assi relativi alle origini da caricare.

par 2 E' un parametro significativo solo nel caso in cui la tabella da caricare è quella utensili ed è presente anche il parametro *par 1*. Indica se vanno aggiornati oppure no i valori pocket in tabella utensili.

par 2 = 0 non aggiorna pocket

par 2 = 1 aggiorna pocket

In caso di mancanza del parametro viene assunto 0 come valore di default.

Caratteristiche:

Il comando LOA permette di caricare in memoria dual port le tabelle precedentemente salvate su file dall'ambiente TABLE EDITOR.

I nomi qui descritti per le estensioni che identificano il tipo di tabella, sono quelli di default e possono essere diversi se variati tramite l'utility SETUP del TABLE EDITOR.

Nel caso di utilizzo dei magazzini l'operazione di "inizializzazione pocket" va comunque effettuata da TABLE EDITOR. Tale operazione si rende necessaria ogni qual volta si caricano tabelle magazzini relativi a magazzini con pocket di caratteristiche diverse o dopo un reset della memoria dual port.

Esempi:

- | | |
|--------------------------|--|
| (LOA, NEW_TOOL.MAG) | Carica la tabella magazzini di nome NEW_TOOL. |
| (LOA, DRILL_T.TOL, 1, 1) | Carica la tabella utensili DRILL_T (relativa al magazzino 1, aggiornando anche il campo pocket di ciascun utensile). |
| (LOA, ORIG_1.ORG, 1) | Carica la tabella origini ORIG_1 relativa al processo 1. |

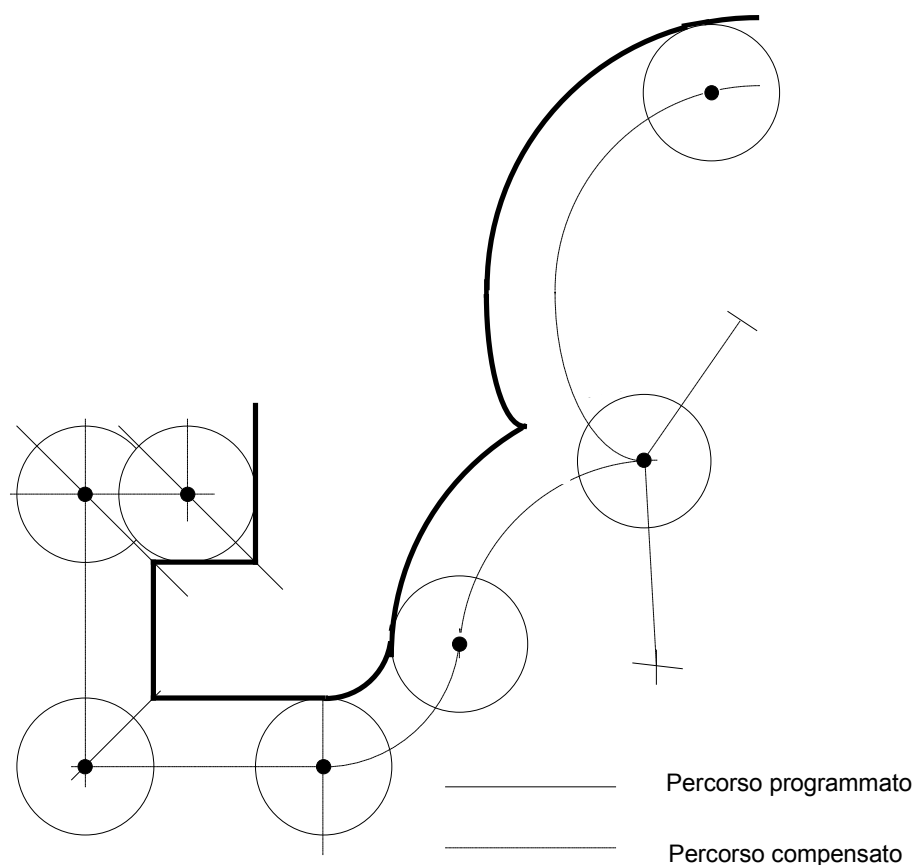
FINE CAPITOLO

COMPENSAZIONE DIAMETRO UTENSILE

La compensazione del diametro utensile è una correzione basata sulla geometria specifica dell'utensile. Poiché l'utensile generalmente viene considerato come avente una sezione circolare, la correzione viene applicata al diametro del cerchio.

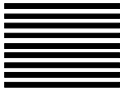
La compensazione del diametro dell'utensile agisce in direzione perpendicolare al profilo programmato, il quale è composto da segmenti di rette ed archi di circonferenza.

La figura illustra come viene applicata la compensazione diametro utensile.



Posizionamento dell'utensile durante la compensazione diametro utensile

Quando la compensazione diametro utensile è attiva sul profilo, l'utensile viene posizionato sull'intersezione tra due elementi geometrici consecutivi traslati dell'ampiezza del raggio utensile.



G40 G41 G42- Compensazione diametro utensile

Le seguenti funzioni consentono di abilitare o disabilitare la compensazione diametro utensile:

- G40 Compensazione diametro utensile disabilitata
- G41 Compensazione con utensile a sinistra del profilo
- G42 Compensazione con utensile a destra del profilo

Sintassi

G40 [*codici-G*] [*operandi*]

G41 [*codici-G*] [*operandi*]

G42 [*codici-G*] [*operandi*]

dove:

- | | |
|-----------------|---|
| <i>codici-G</i> | Altri codici G compatibili con G41, G42 e G40 (vedere tabella "Codici G compatibili" del capitolo 1). |
| <i>operandi</i> | Tutti gli operandi e codici possibili nei blocchi di funzione G. |

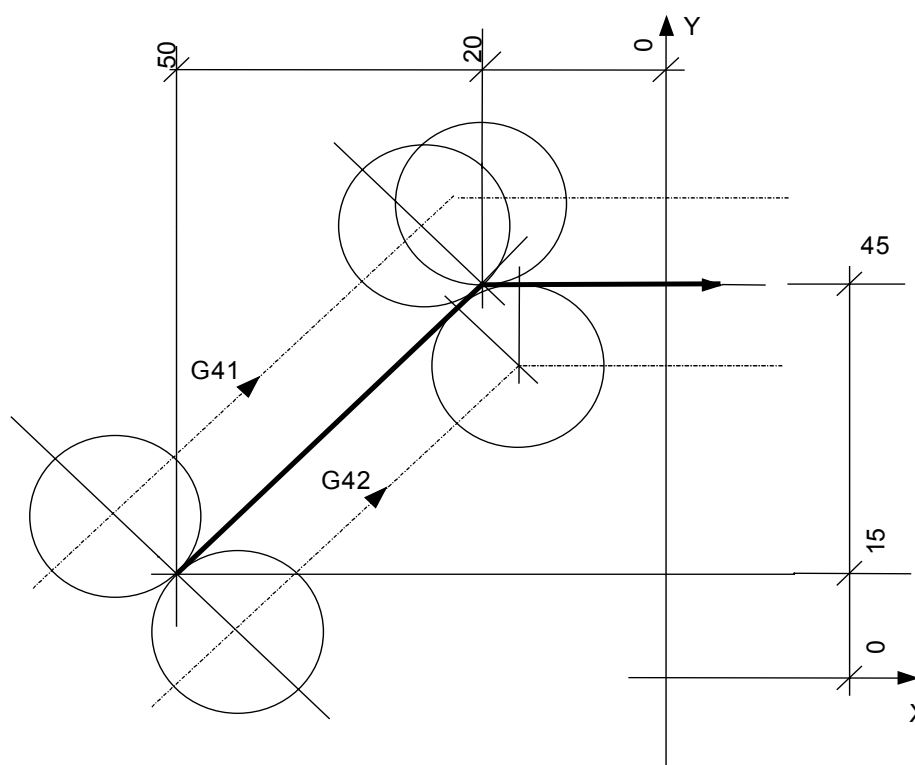
Abilitazione della compensazione diametro utensile

La compensazione diametro utensile viene abilitata dai codici G41 o G42. Lo spostamento sul primo punto del profilo deve essere di tipo lineare (G00-G01).

Su tale punto la correzione agisce in direzione normale al primo movimento nel piano, programmato successivamente alla funzione G41 o G42. Tale movimento può essere rettilineo o circolare.

Seguono i due esempi relativi.

- Il primo movimento del profilo è di tipo lineare



Programma:

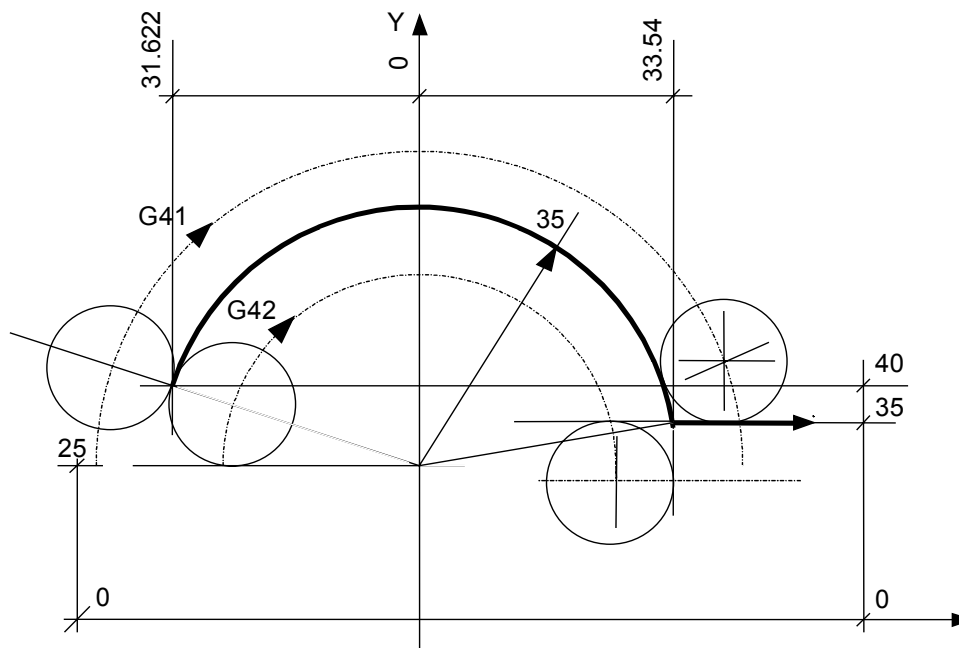
Compensazione diametro utensile sulla destra del profilo:

```
G1 G42 X-50 Y15 F200
X-20 Y45
```

Compensazione diametro utensile sulla sinistra del profilo:

```
G1 G41 X-50 Y15 F200
X-20 Y45
```

- Il primo movimento del profilo è di tipo circolare



Programma:

Compensazione diametro utensile sulla destra del profilo:

```
G1 G42 X-31.622 Y40 F200
G2 X33.541 Y35 I J25
G1 X.....
```

Compensazione diametro utensile sulla sinistra del profilo:

```
G1 G41 X-31.622 Y40 F200
G2 X33.541 Y35 I J25
G1 X....
```

Note relative all'uso della compensazione diametro utensile

Una volta abilitata, la compensazione diametro utensile agisce su tutti i movimenti programmati con velocità di rapido o di lavoro. Se la compensazione della fresa è abilitata con G41 o G42, non è possibile programmare le seguenti funzioni G:

- G81-G89 (cicli fissi)
- G70 G71 (programmazione in mm/pollici)
- G79 (programmazione riferita allo zero macchina)
- G33 (filettatura)
- G72 G73 G74 (cicli di misura)
- G16, G17, G18, G19 (cambio piano di interpolazione)

Con la compensazione diametro utensile abilitata, il controllo visualizza un messaggio di errore quando:

- si programma un raggio interno minore del raggio dell'utensile.
- l'esecuzione di un movimento lineare traslato comporterebbe l'inversione del senso di percorrenza rispetto al profilo originale.

All'interno di un profilo compensato con G41 o G42 è possibile programmare il movimento di assi non appartenenti al profilo (al massimo due movimenti consecutivi).

Ottimizzazione del percorso utensile (Tool Path Optimization)

Nell'ambito della programmazione con correzione diametro utensile, è possibile attivare (da part program o MDI) la funzionalità Tool Path Optimization.

Questa funzionalità viene attivata mediante la variabile di sistema TPO, e può inoltre essere personalizzata a seconda delle esigenze di lavorazione con la variabile di sistema TPT.

Il TPO consente due tipi di ottimizzazione:

- "Riduzione" automatica degli spigoli tra blocchi di movimento lineari e circolari.
- Entrata/uscita tangente al profilo (con un arco di cerchio).

IMPORTANTE

Le variabili TPO e TPT sono descritte dettagliatamente, più avanti, in questo stesso capitolo.

Disabilitazione della compensazione diametro utensile

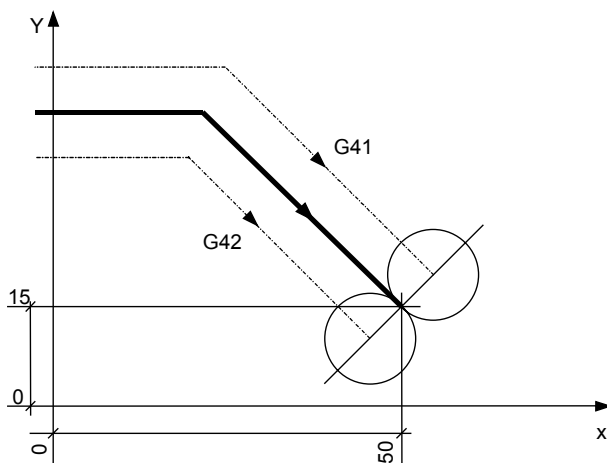
La compensazione del diametro utensile viene disabilitata tramite il codice G40.

Le modalità di uscita sono le seguenti:

- *Senza TPO* (modalità offset standard)
Il movimento programmato sul blocco G40 è considerato ancora in offset.
- *Con TPO attivo*
Il movimento programmato sul blocco G40 è da considerarsi il movimento di uscita dall'offset.

Gli esempi seguenti illustrano vari casi di uscita dal profilo:

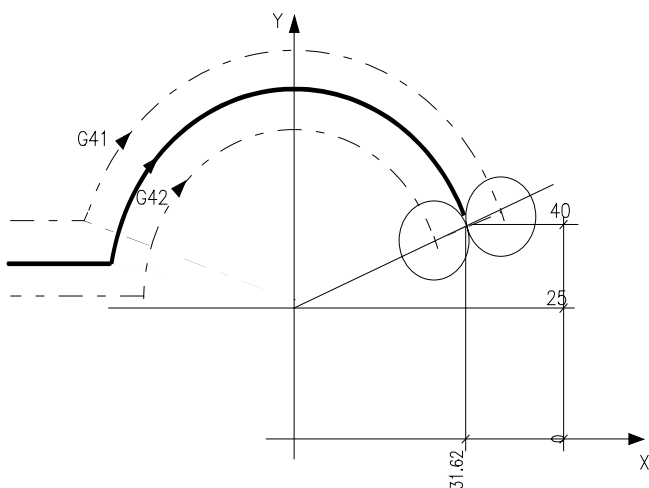
- **L'ultimo movimento del profilo è di tipo rettilineo (TPO=0)**



Programma:

```
N88 G1 G40 X50 Y15
N100 G X.. Y..
```

- **L'ultimo movimento del profilo è di tipo circolare (TPO=0)**



Programma:

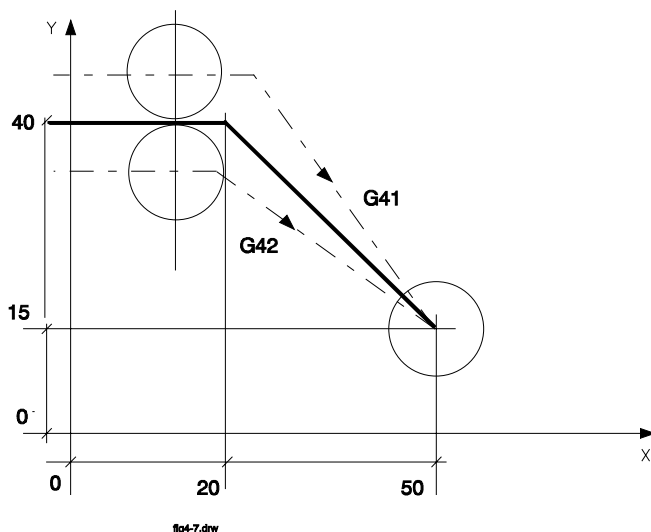
```
N99 G2 G40 X31.62 Y40 I J25
N100 G X.. Y..
```

Disabilitazione della compensazione con TPO attivo

La disabilitazione della compensazione diametro utensile con TPO attivo avviene sempre tramite il codice G40. L'ultimo punto su cui agisce la compensazione diametro utensile è quello del blocco precedente a quello che contiene G40.

Esempi:

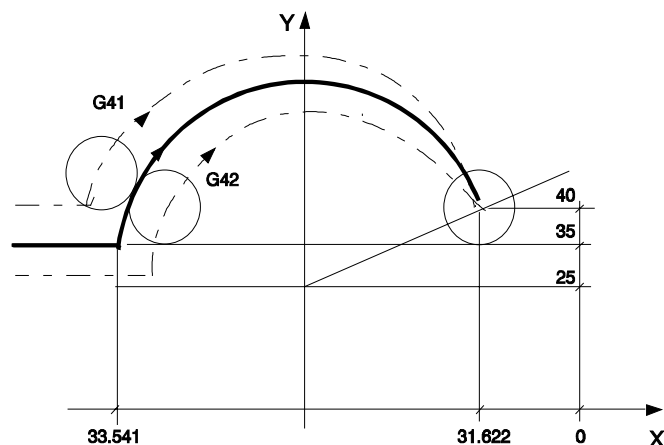
- L'ultimo movimento del profilo è di tipo rettilineo (TPO attivo)



Programma:

```
N87 G1 X20 Y40
N88 G40 X50 Y15
N100 G X.. Y..
```

- L'ultimo movimento del profilo è di tipo circolare (TPO attivo)



Programma:

```
N98 G1X-33.541 Y35
N99 G2 G40 X31.622 Y40 I J25
N100 G X.. Y..
```



Con le release antecedenti alla 3.0, occorre verificare la presenza della funzione G40 in vecchi part program; con TPO attivo, infatti la G40 esce dal modo "compensazione diametro utensile" in modo diverso.

Programmando un blocco di part program con solo "G40" (senza punto finale associato) si ottiene la stessa prestazione della compensazione diametro standard (come TPO=0).

Leggere con attenzione questa pagina e quelle relative alla gestione delle funzioni TPO e TPT prima della programmazione.

CAMBIO COMPENSAZIONE DIAMETRO UTENSILE

Questa sezione descrive la modalità con la quale viene gestito il cambio di compensazione ($G41 \Rightarrow G42$ o viceversa) durante la lavorazione di un profilo in offset. Il cambio del tipo di compensazione può avvenire nel punto di intersezione tra i percorsi programmati (con compensazione sinistra/destra o destra/sinistra), oppure tramite l'aggiunta di un nuovo blocco di movimento generato in modo automatico dal sistema.

Il tipo di cambio compensazione (sull'intersezione o con blocco di collegamento aggiuntivo) dipende dalla tipologia dei movimenti precedente e successivo al cambio stesso. Nelle pagine seguenti vengono illustrati i vari casi possibili:

- Lineare/Lineare con blocchi di movimento tangenziali
- Lineare/Lineare con inversione della direzione
- Lineare/Lineare con generazione automatica di un nuovo blocco di movimento
- Lineare/Lineare senza generazione automatica di un nuovo blocco di movimento
- Lineare/Circolare - Circolare/Lineare
- Circolare/Circolare

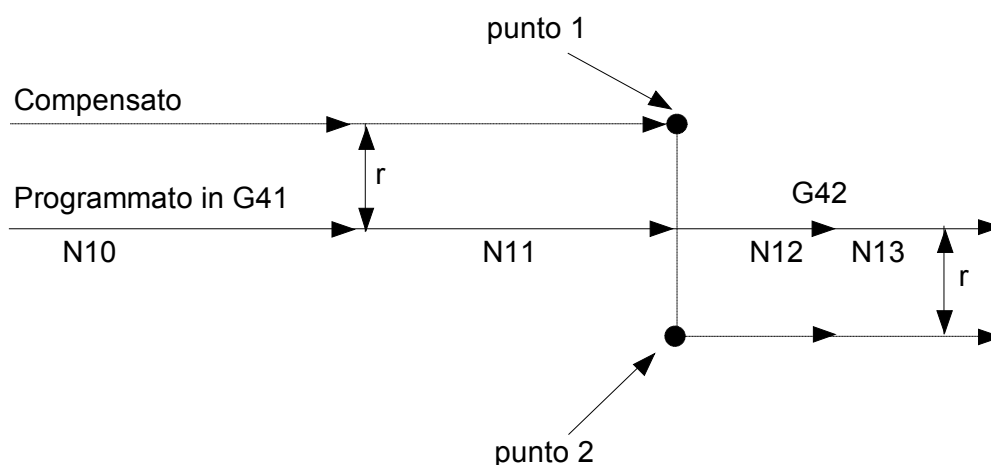
Percorso utensile lineare/lineare

La figura seguente illustra il percorso dell'utensile quando la compensazione passa da $G41$ a $G42$ durante l'esecuzione di due movimenti di tipo lineare.

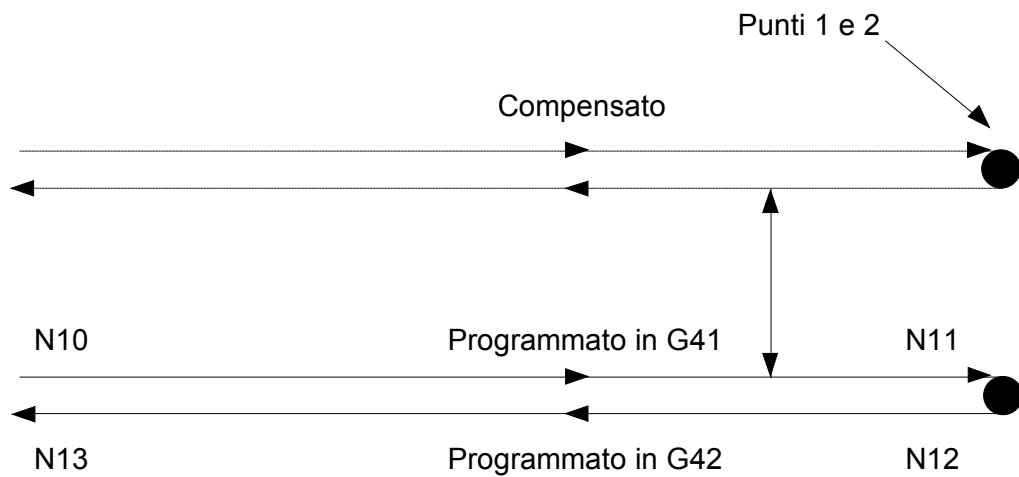
Passando da $G41$ a $G42$, il controllo genera due punti, che chiameremo punto 1 e punto 2.

- Il *punto 1* è la posizione finale dell'utensile prima del cambiamento del tipo di compensazione.
- Il *punto 2* è la posizione desiderata per l'inizio del primo blocco che usa la direzione di compensazione cambiata.

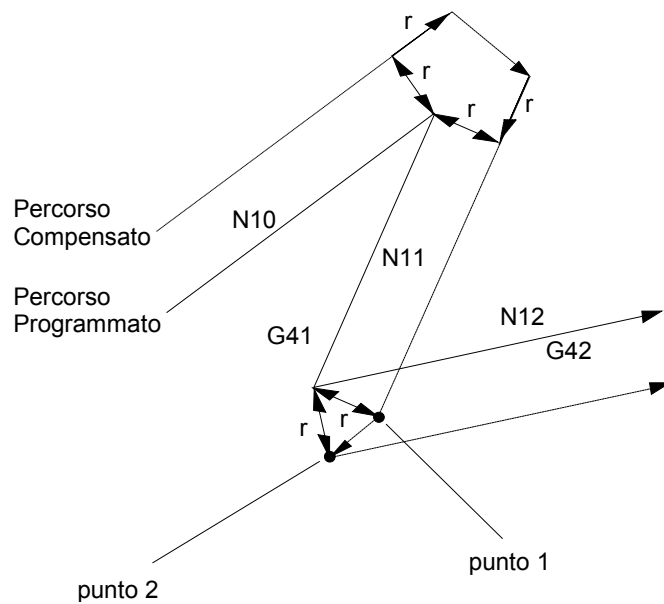
Il controllo genera automaticamente il blocco di movimento che collega il punto 1 con il punto 2:



Cambio Lineare/Lineare con blocchi di movimento tangenziali



Cambio Lineare/Lineare con inversione della direzione

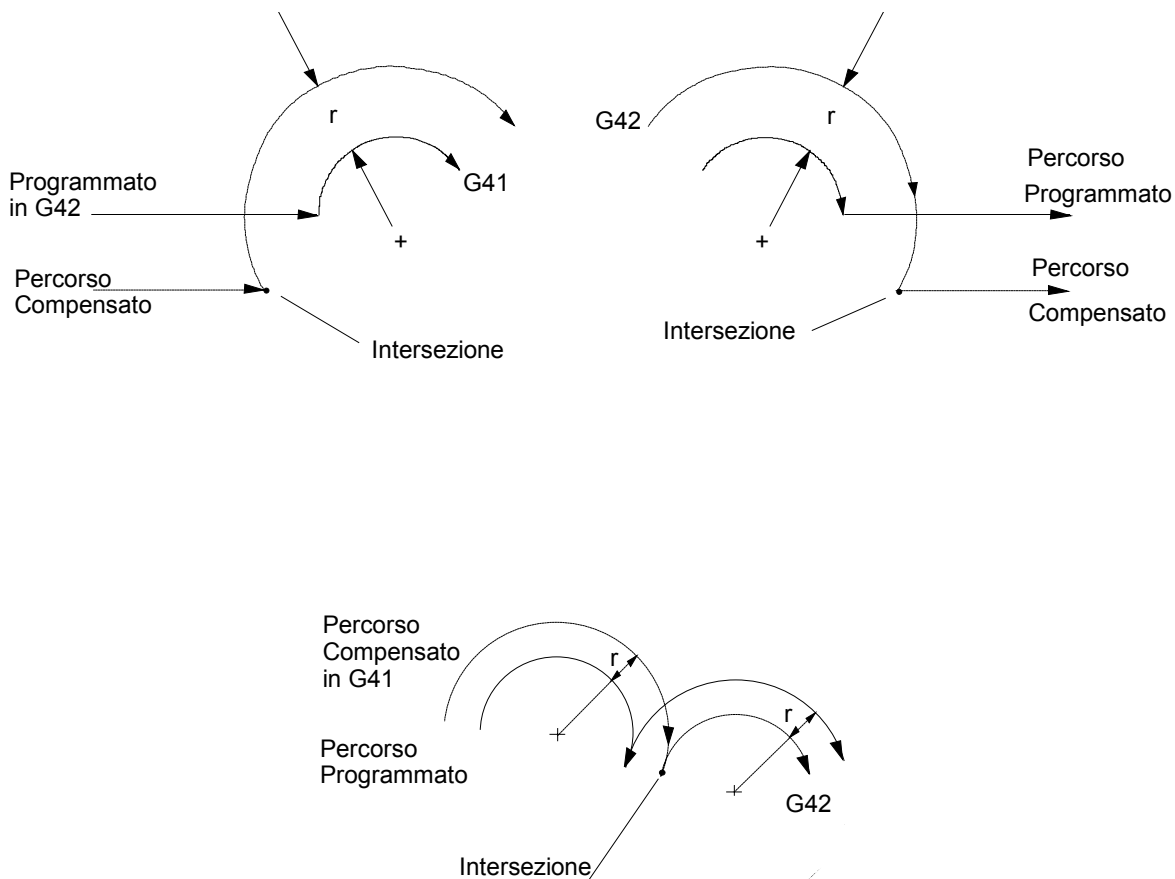


Cambio Lineare/Lineare con generazione di un nuovo blocco

Percorsi utensile Lineare/Circolare, Circolare/Lineare, Circolare/Circolare

Per ognuno dei seguenti tipi di percorso utensile, nei quali avviene un cambiamento della direzione di compensazione, il sistema Serie 10 cercherà di trovare un'intersezione tra il percorso programmato in G41 e quello programmato in G42 (o viceversa).

Se il sistema Serie 10 trova un punto di intersezione, esso modifica il punto finale del percorso utensile compensato originale, mentre il punto iniziale del nuovo percorso utensile compensato coincide con tale intersezione (figura sotto)

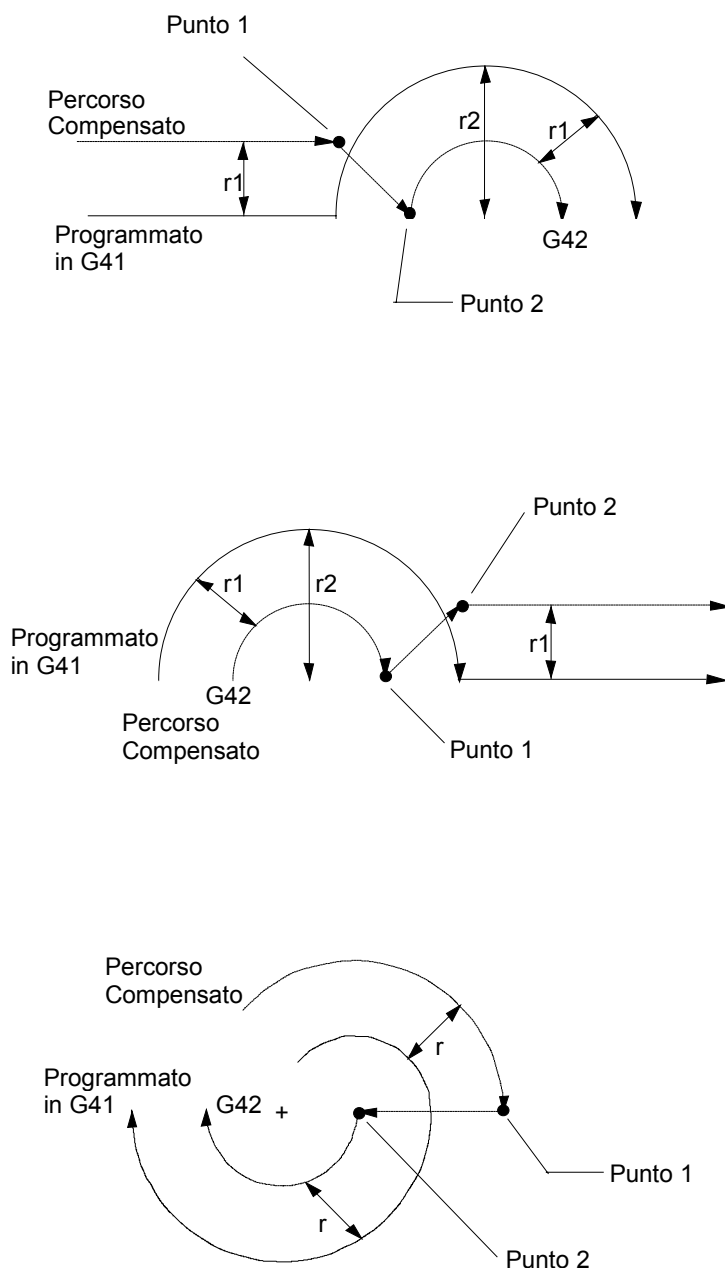


Cambio di compensazione con intersezione del percorso attuale

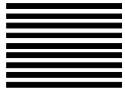
Possono verificarsi tuttavia dei casi in cui non esiste intersezione tra i percorsi utensile; in questi casi, passando da G41 a G42 (o viceversa) il sistema si comporta come illustrato nelle figure che seguono.

- Il *punto 1* è la posizione finale dell'utensile prima del cambiamento del tipo di compensazione.
- Il *punto 2* è la posizione desiderata per l'inizio del primo blocco che usa la direzione di compensazione cambiata.

Il controllo genera automaticamente il blocco di movimento che collega il punto 1 con il punto 2:



Cambio compensazione senza possibilità di intersezione tra i percorsi utensile.



r - Raccordi su profili compensati

Durante la lavorazione di profili convessi è possibile introdurre un raccordo circolare tra due elementi geometrici.

Sintassi

r *valore*

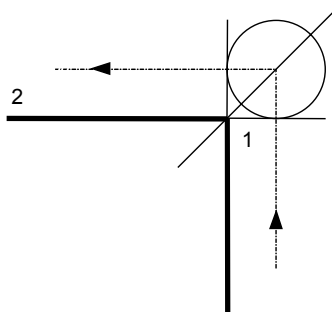
dove:

valore Il valore del raccordo da programmare: negativo per spostamenti in verso orario, positivo per spostamenti in verso antiorario.

IMPORTANTE

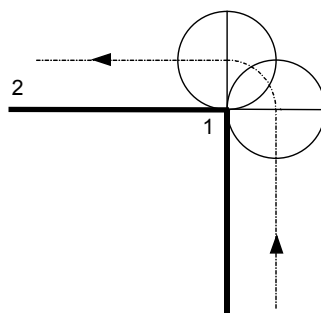
Programmando r0 (raccordo zero) il centro dell'utensile esegue un arco circolare il cui centro coincide con lo spigolo del profilo.

Esempio:



Senza raccordo

1) N20 G1 X100 Y100
2) N21 X-100



Con raccordo

1) N20 G1 X100 Y100
N21 r0
2) N22 X-100



b - Smussi nei profili compensati

E' possibile inserire uno smusso, piuttosto che un raccordo, tra due blocchi di movimento (lineari o circolari) la cui traiettoria si interseca, programmando un valore senza segno.

Sintassi

b valore

dove:

valore E' il valore dello smusso misurato a partire dal punto di intersezione.

Tale valore di **b** viene interpretato nei seguenti modi:

- **Smusso tra due profili rettilinei:**

b è la distanza che intercorre tra il punto teorico di intersezione tra i prolungamenti dei segmenti di profilo ed il punto finale generato.

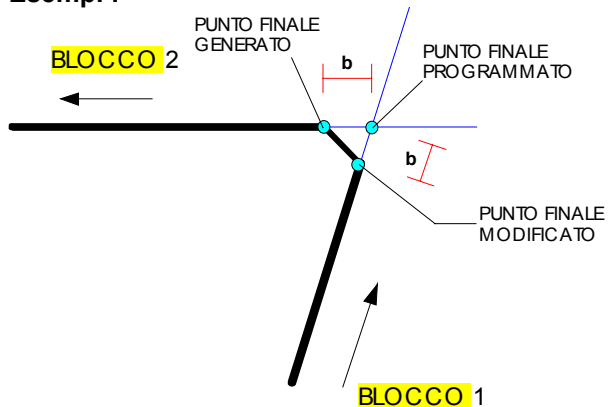
- Smusso tra un profilo rettilineo ed uno circolare:

b è la distanza che intercorre tra il punto teorico di intersezione tra il prolungamento del segmento del profilo rettilineo e la tangente del profilo circolare.

- **Smusso tra due profili circolari:**

b è la distanza che intercorre tra il punto teorico di intersezione tra il prolungamento delle tangenti dei due profili circolari.

Esempi :



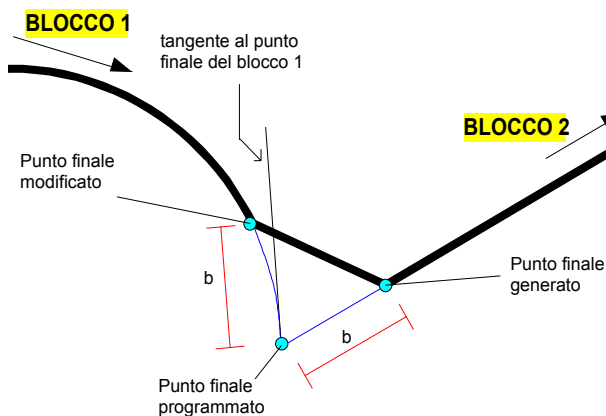
Smusso tra due blocchi lineari

Esempio di programmazione:

```

.
.
N10 G1 X100 Y100 ;blocco 1
N11 b3
N12 X-100 ;blocco 2
.
.
.

```



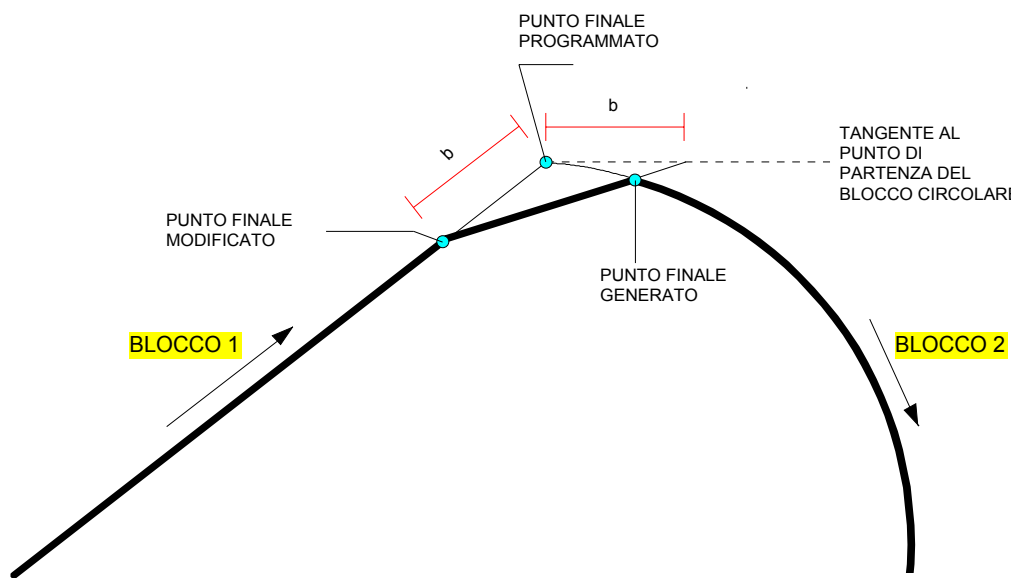
Smusso tra un blocco di movimento circolare ed uno lineare

Esempio di programmazione:

```

.
N10 GXY
N11 G42
N12 G1X10Y70F1000
N14 G2X70Y20R50      ;blocco 1
N16 b10
N18 G1X150Y90        ;blocco 2
N20 G40
.
.
.
    
```

Smusso tra un blocco di movimento lineare ed uno circolare.

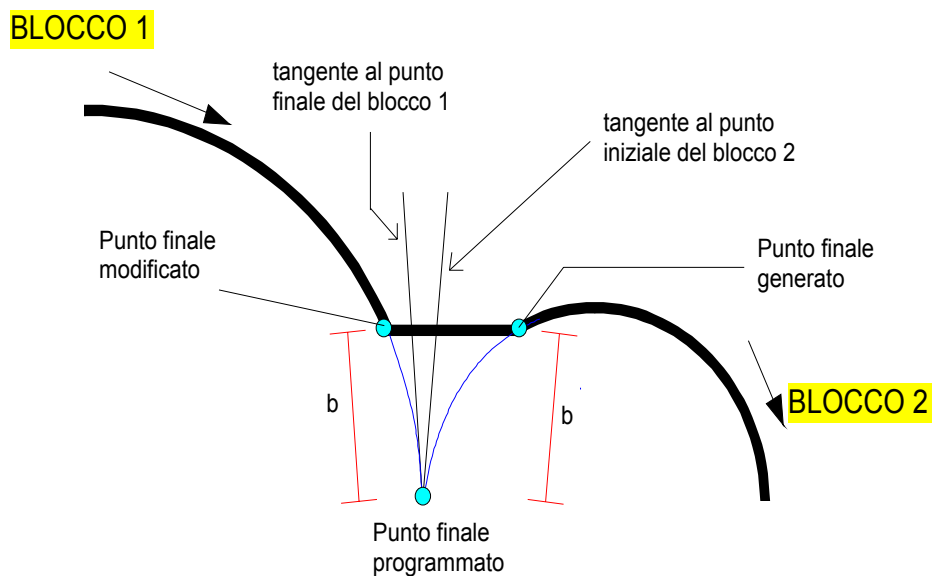


Esempio di programmazione:

```

.
N20 G42
N22 GXY
N24 G1X40Y40F1000      ;blocco 1
N26 b10
N28 G2X80Y5R70        ;blocco 2
N30 G40
.
    
```

Smusso eseguito tra due blocchi di movimento circolari.

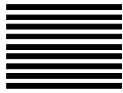


Esempio di programmazione:

```

.
.
N10 G42
N20 GX10Y60
N30 G2X50Y40R50F1000 ;blocco 1
N40 b5
N50 G2X100Y5050R30 ;blocco 2
.
.

```



TPO - Ottimizzazione del percorso utensile in G41/G42

La modalità TPO (Tool Path Optimization) permette di ottimizzare il percorso dell'utensile in presenza di G41 o G42 attive.

L'algoritmo inserisce automaticamente interpolazioni circolari all'inizio ed alla fine del profilo, nonché raccordi sugli spigoli descritti dal profilo stesso.

Sintassi

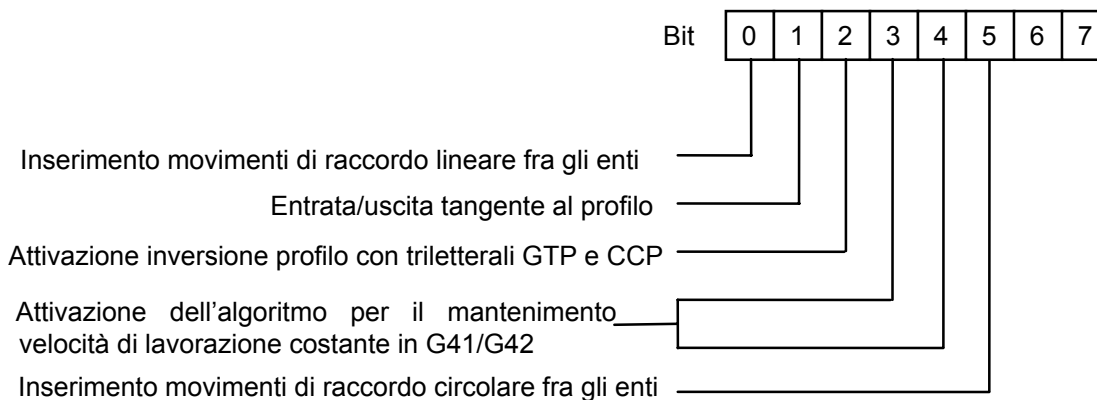
TPO = n

dove:

n Tipo di ottimizzazione da effettuare.

Il valore di n da specificare nell'istruzione si ricava sommando i pesi decimali dei bit corrispondenti a ciascuna delle prestazioni desiderate.

Se $n=0$ l'algoritmo di ottimizzazione viene disattivato.



Caratteristiche:

La programmazione di un blocco contenente l'istruzione TPO= n attiva l'algoritmo di ottimizzazione del percorso utensile.

Questo algoritmo può introdurre automaticamente dei movimenti all'inizio/fine profilo ed ottimizzare il percorso sugli spigoli, inserendo un numero variabile di movimenti (da 1 a 3) in relazione al profilo.

IMPORTANTE

Nel caso di programmazione in linguaggio GTL l'attivazione della modalità TPO non viene presa in considerazione.

IMPORTANTE

All'accensione del sistema viene attivata la modalità TPO configurata in AMP; essa può essere variata tramite la programmazione.
Su RESET viene ripristinata la modalità caratterizzata in AMP.

IMPORTANTE

L'attivazione del bit 5 ha effetto solo se è abilitato anche il bit 0, ossia, se si desidera inserire dei raccordi circolari laddove si inserirebbero, invece, dei raccordi lineari, impostare un **TPO = 33**.

- **Modalità TPO=1 e TPO=33**

La modalità TPO=1 attiva l'algoritmo di ottimizzazione del percorso utensile inserendo movimenti lineari di raccordo fra gli enti del profilo in funzione della deviazione angolare tra gli enti stessi.

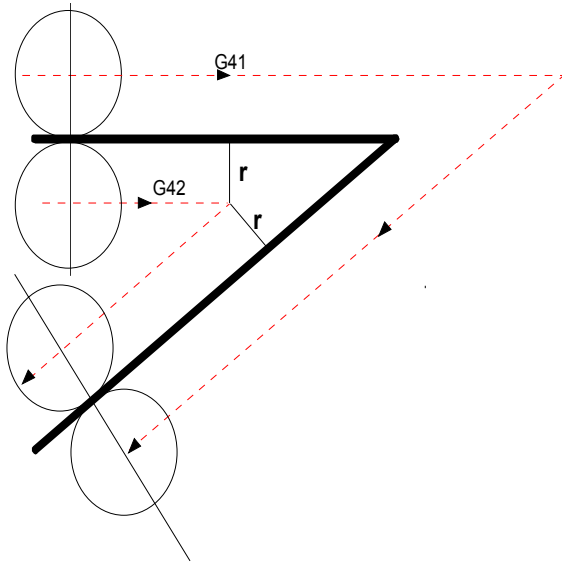
Se, inoltre, è settato anche il bit 5 (TPO=33) allora, laddove si verificano le condizioni per l'ottimizzazione dello spigolo, tramite inserimento di movimenti lineari, viene inserito un raccordo circolare avente centro nell'intersezione dei due enti non compensati e raggio pari al raggio utensile.

I casi di applicazione dell'algoritmo sono i seguenti:

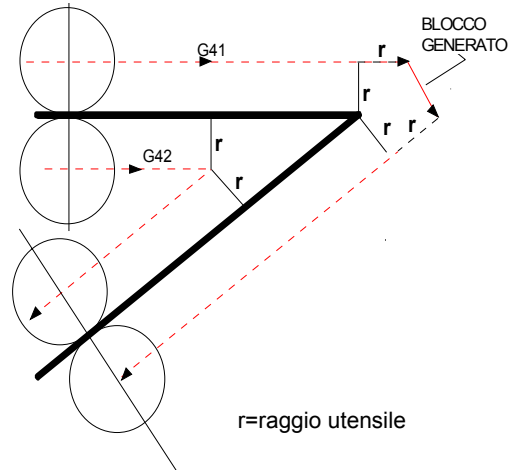
- A) Retta-retta con angolo di deviazione compreso tra 90° e 180° ($90^\circ < \alpha \Leftrightarrow 180^\circ$).
- B) Cerchio-retta, retta-cerchio, cerchio-cerchio con angolo di deviazione compreso tra 0° e 180° ($0^\circ < \alpha \Leftrightarrow 180^\circ$).

Se tra gli enti descritti ai punti A) e B) è stato programmato uno smusso (b), od un raccordo, (r) l'algoritmo non viene invece applicato.

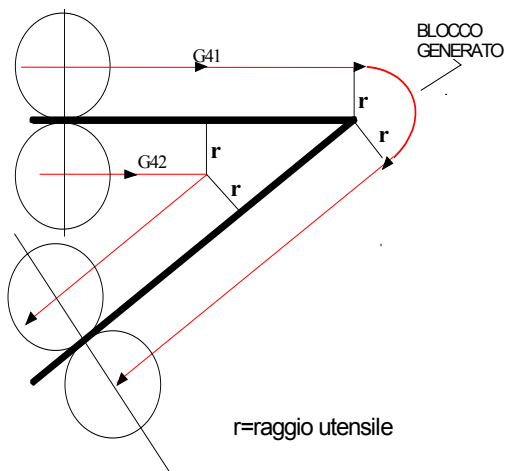
Esempio :



Senza ottimizzazione (TPO=0)



Con ottimizzazione (TPO=1)



Con ottimizzazione (TPO=33)

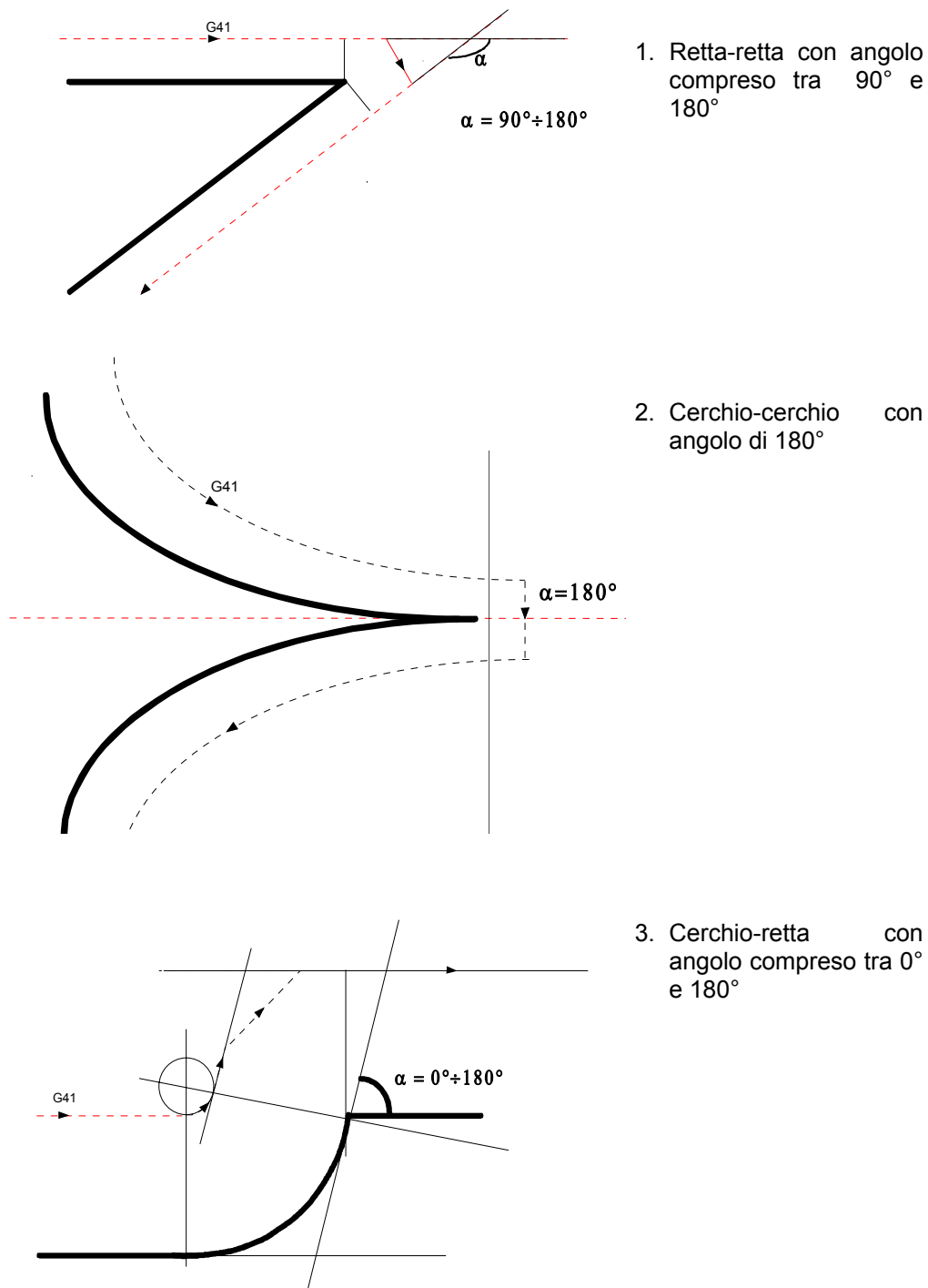
IMPORTANTE

Nell'esempio precedente non si è tenuto conto della programmazione della soglia TPT; si è assunto cioè che TPT=0.

Se, invece, viene impostato uno scostamento di soglia dallo spigolo (TPT diverso da zero) il suddetto raccordo circolare avrà il centro traslato, seguendo gli stessi criteri di calcolo usati per il raccordo lineare, e il raggio ridotto di conseguenza.

Esempi relativi alla modalità TPO=1

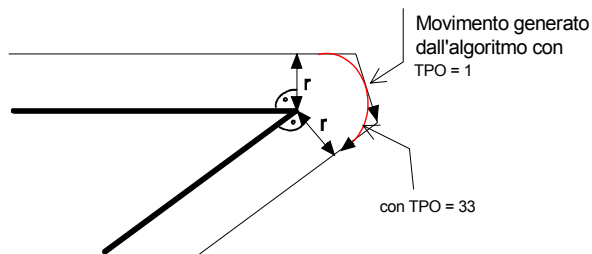
Esempi di profili in cui la deviazione angolare comporta l'applicazione dell'algoritmo di ottimizzazione:



Esempi di profili in cui l'algoritmo comporta l'inserimento di movimenti

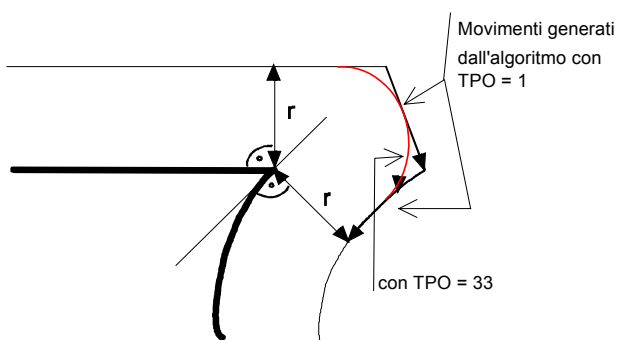
Il profilo ricalcolato in presenza di spigoli sarà caratterizzato dall'introduzione automatica di movimenti (da 1 a 3) a seconda del tipo di profilo che forma lo spigolo da ottimizzare.

1) Retta - retta



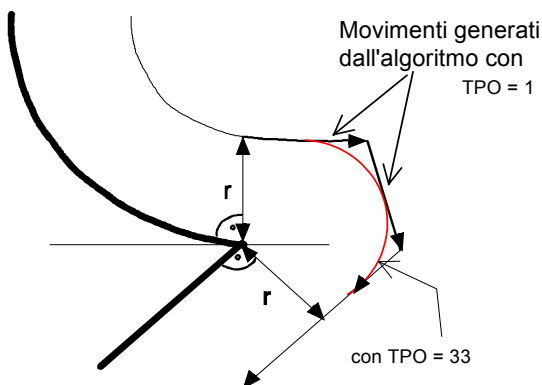
L'algoritmo genera 1 movimento

2) Retta - cerchio



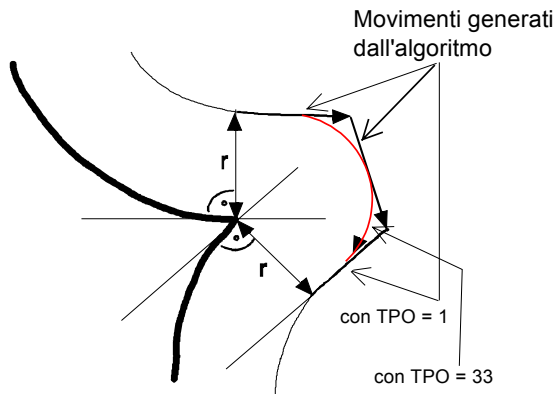
L'algoritmo genera 2 movimenti

3) Cerchio - retta



L'algoritmo genera 2 movimenti

4) Cerchio - cerchio



L'algoritmo genera 3 movimenti

- **Modalità TPO=2**

La modalità TPO=2 attiva l'algoritmo di entrata/uscita tangente al profilo tramite l'inserimento di enti circolari all'inizio ed alla fine del profilo stesso:

- A) All'entrata del profilo viene generato un ente circolare fra il punto precedente (P0) ed il primo punto del profilo offsettato (P1), intendendo come primo punto quello programmato nel blocco contenente la funzione G41 o G42.

```
.
X80 Y70                ;P0
G41 X100 Y100;P1
X140                   ;P2
.                       ;ecc.
.
```

- B) All'uscita del profilo viene generato un ente circolare fra l'ultimo punto del profilo offsettato (P99) ed il punto finale dell'ente di uscita del profilo (P100) intendendo come ente di uscita quello programmato nel blocco contenente la funzione G40.

```
.
G41 X100 Y100;P1
.
.
.
.
X170 Y160              ;P99
G40 X180 Y185;P100
.                       ;ecc.
.
.
```

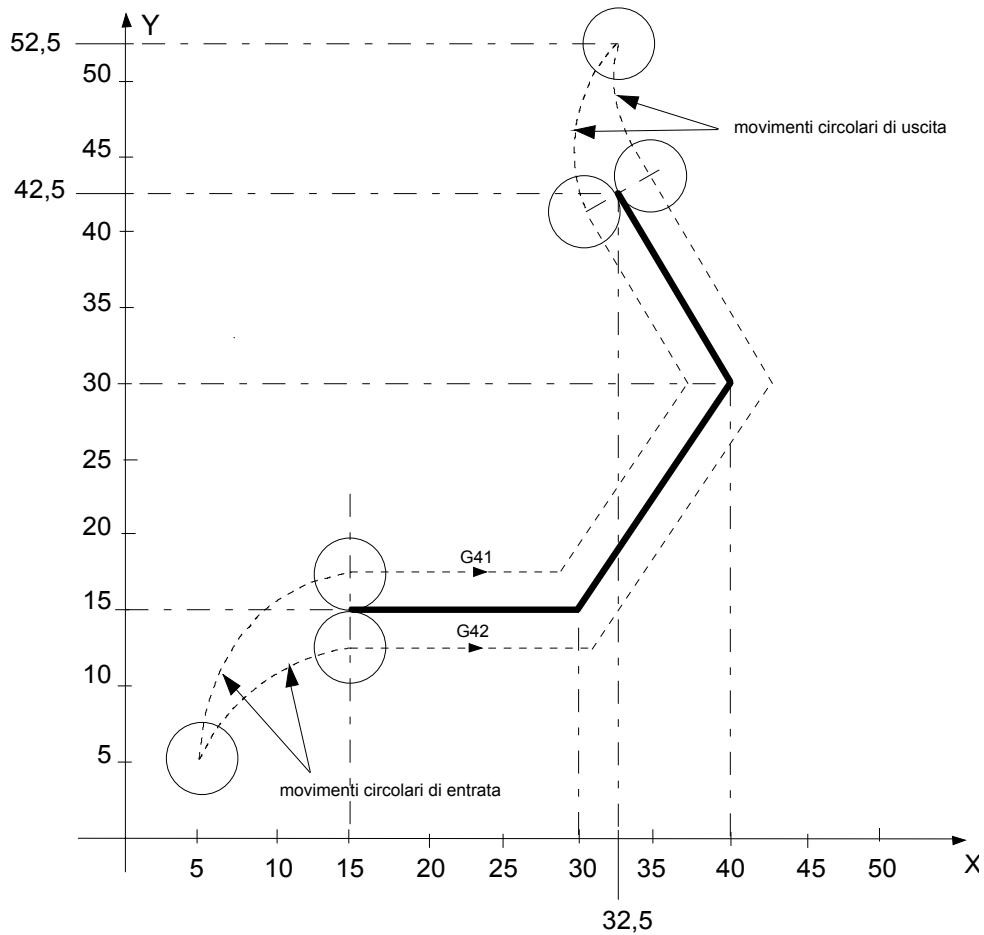
- **Modalità TPO = 3**

La modalità TPO=3 attiva contemporaneamente gli algoritmi delle modalità 1 e 2, inserendo così sia movimenti di raccordo tra enti, che movimenti circolari di entrata ed uscita dal profilo.

Le regole definite per le due modalità precedenti sono da ritenersi valide anche per TPO=3.

Esempi relativi alla modalità TPO=2

Esempio 1:



```

N1  S1000 M3 T1.1M6      ;raggio utensile = 2.5
N2  X5 Y5
N3  G1 G42 X15 Y15 F500   ;primo punto del profilo
N4  X30
N5  X40 Y30
N6  X32.5 Y42.5           ;ultimo punto del profilo
N7  G40 X32.5 Y52.5
N8  GX100
    
```

Esempio 2:

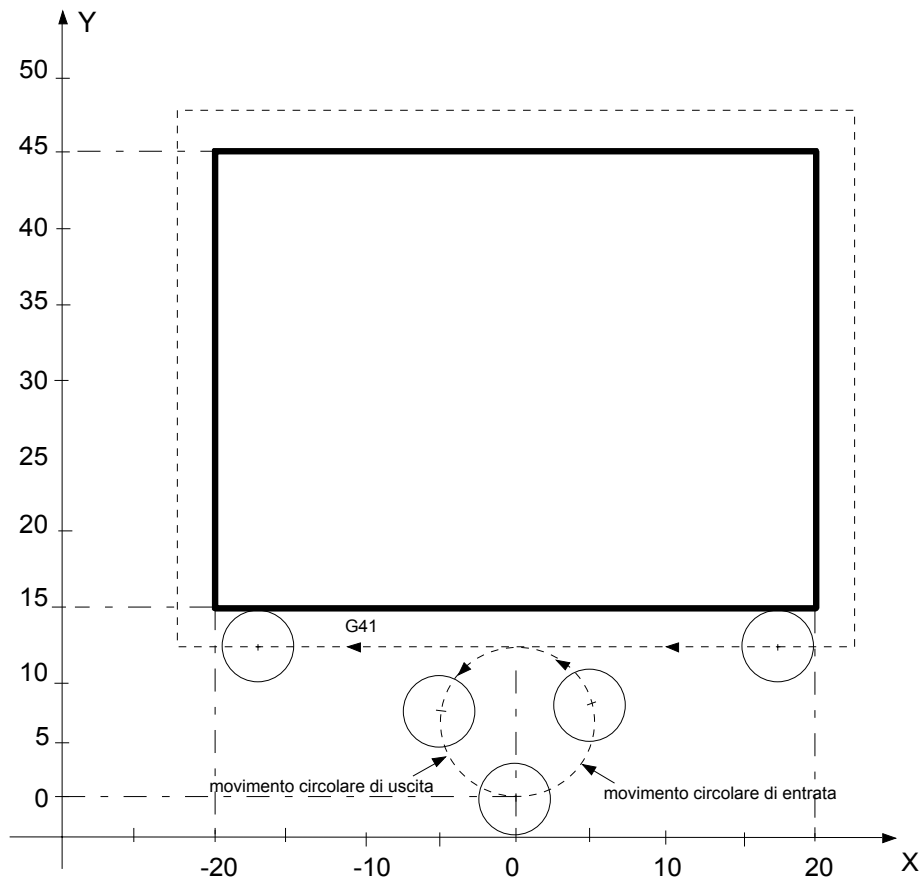


fig511

N1	S1000 M3 T1.1 M6	;raggio utensile = 2.5
N2	X0 Y0	
N3	G1 G41 Y15 F800	;primo punto del profilo
N4	X -20	
N5	Y45	
N6	X40	
N7	Y15	
N8	X0	;ultimo punto del profilo
N9	G40 Y0	

• Modalità TPO=4

La modalità TPO=4 attiva l'algoritmo di inversione del profilo.

Questa funzionalità permette la lavorazione del profilo partendo dall'ultimo movimento, per finire sul primo (lavorazione al contrario).

IMPORTANTE

- Questa prestazione è gestita solo nell'ambito dei comandi GTP e CCP.
- L'algoritmo provvede anche ad invertire le eventuali modalità offset, in modo da mantenere la posizione dell'utensile interno/esterno come richiesto.
- Per maggiori informazioni, consultare le pagine relative alla trattazione dei triletterali GTP e CCP (pag. 4-33 e 4-36).

• Modalità TPO=8 e TPO=16

Le modalità TPO=8 e TPO=16 attivano l'algoritmo che mantiene costante la velocità di lavorazione in G41/G42. In particolare viene mantenuta costante la velocità di "contatto" tra utensile e pezzo, variando la velocità di avanzamento riferita al centro utensile.

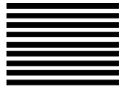
Tale variazione è funzione del raggio utensile e viene applicata ai soli movimenti circolari. Essa produrrà un aumento o una diminuzione della velocità di avanzamento riferita al centro utensile a seconda che il raggio della circonferenza percorsa dal centro utensile sia maggiore o inferiore al raggio della circonferenza programmata. Settando il bit 3 della variabile TPO (TPO=8) viene abilitato l'incremento della velocità, mentre settando il bit 4 (TPO=16) viene abilitata la diminuzione della velocità.

I bit 3 e 4 di TPO possono essere settati insieme ed anche contemporaneamente a tutti gli altri bit della stessa variabile per abilitare le prestazioni corrispondenti.

• Modalità TPO=32

Equivale ad aver programmato TPO = 0.

Questo perché l'attivazione dell'ottimizzazione degli spigoli tramite inserimento di un raccordo circolare è subordinata all'aver abilitato il TPO a 1. Se il bit 0 è nullo e il bit 5 è a 1 (TPO=32) l'algoritmo di ottimizzazione a raccordi circolari non viene abilitata.



TPT - Soglia di scostamento dallo spigolo (Tool Path Threshold)

Dichiarazione del valore della soglia di scostamento dallo spigolo durante l'ottimizzazione del percorso utensile con TPO=1 (Tool Path Optimization).

Sintassi

TPT = *valore*

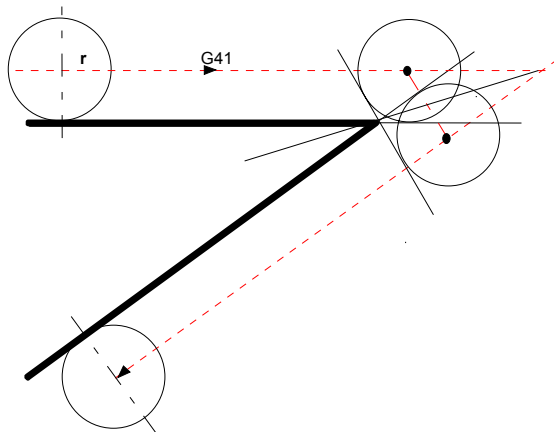
dove:

valore

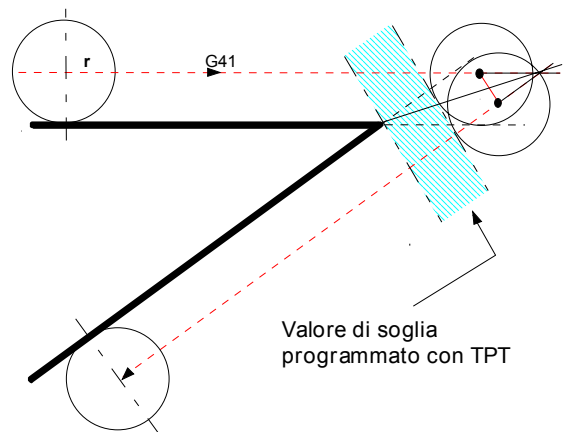
Quota di scostamento espressa nell'unità di misura corrente (mm/inch.). Essa rappresenta la distanza tra il tagliente dell'utensile e lo spigolo descritto dal profilo programmato.

Questo valore può essere anche configurato in AMP.

Esempio:



Con TPT=0



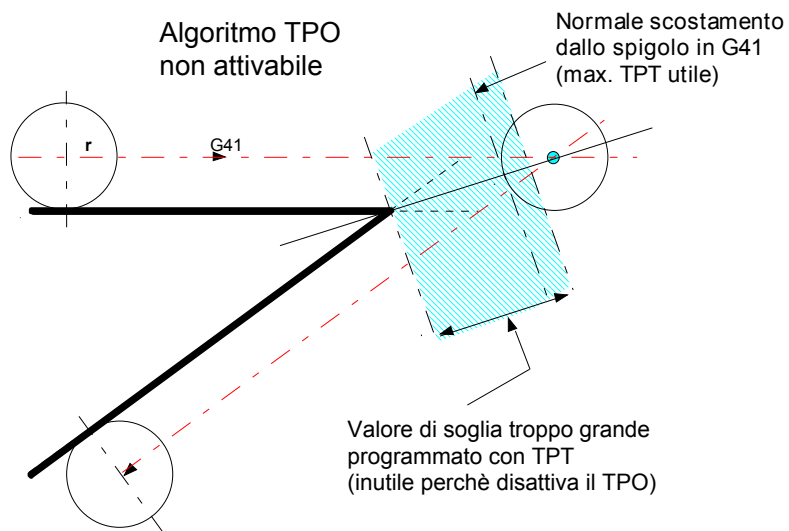
Con programmazione TPT

Caratteristiche:

Se il valore di soglia programmato con TPT porta il centro fresa a superare il punto teorico della normale compensazione diametro utensile (G41/G42 senza programmazione TPO), tale valore sarà ignorato dal sistema.

In questo caso l'algoritmo TPO viene temporaneamente disabilitato, mentre viene utilizzata la normale gestione G41/G42, non vi sarà inoltre alcuna segnalazione d'errore.

Esempio:





u v w - Compensazione parassiale

Quando in un blocco vengono programmati i fattori di compensazione u, v, w il controllo si posiziona su di un punto le cui coordinate sono uguali alle coordinate programmate modificate di un valore che è uguale al raggio dell'utensile moltiplicato per il fattore di compensazione:

posizione X = X programmata + (raggio dell'utensile * u)

posizione Y = Y programmata + (raggio dell'utensile * v)

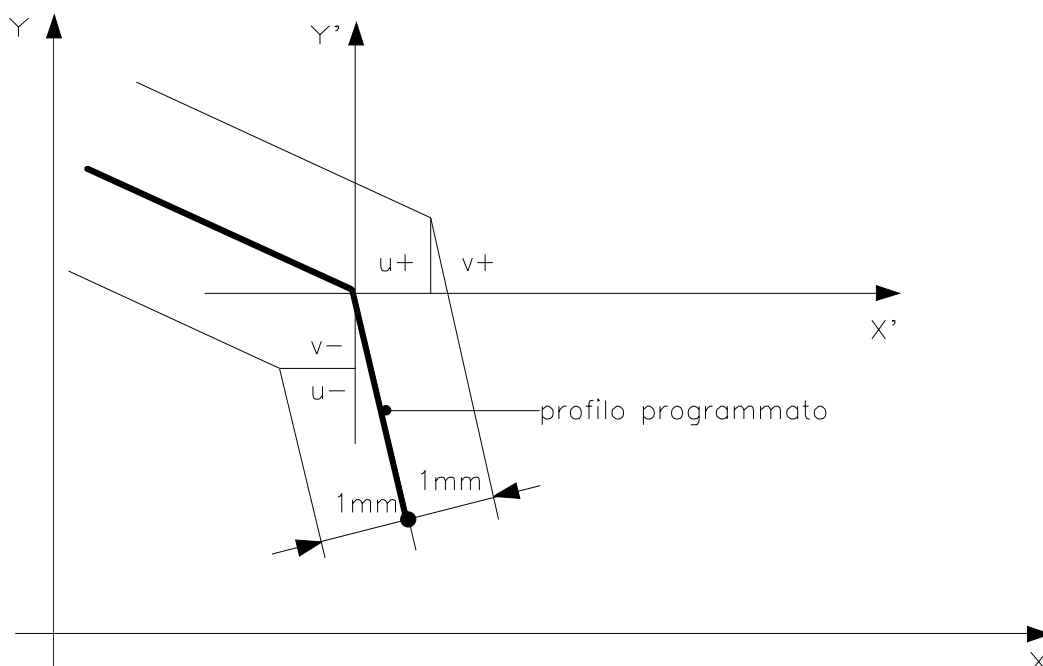
posizione Z = Z programmata + (raggio dell'utensile * w)

Questi fattori di compensazione vengono utilizzati per profili molto semplici (contorniture parallele agli assi) o per fresare superfici tridimensionali.

Non si possono utilizzare i fattori u, v, w quando la compensazione utensile (G41-G42) è abilitata.

Quando i fattori u, v, w sono negativi, devono essere seguiti da un segno meno (-). Se sono positivi, il segno più (+) può essere omissso.

Per determinare il valore e il segno basta considerare i fattori parassiali, u, v, w rispettivamente come le coordinate X, Y, Z del vertice del profilo corretto di un valore unitario. Queste coordinate sono riferite ad un sistema di assi cartesiani paralleli a quelli della macchina la cui origine è nel punto da correggere.



Determinazione del segno dei fattori di compensazione u, v, w.

Il vettore unità rappresenta un correttore raggio utensile lungo un'unità. Pertanto, gli incrementi di traslazione vengono ottenuti moltiplicando u, v, w, per il valore del raggio utensile.

E' possibile compensare i profili formati da:

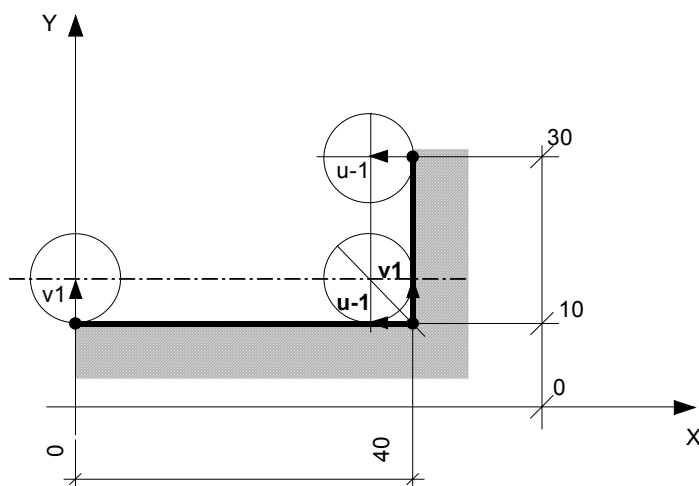
- segmenti rettilinei paralleli agli assi o formanti un angolo con gli assi
- segmenti rettilinei ed archi ad essi tangenti
- archi tangenti tra loro (ammesso che rimangano ancora tali dopo essere stati traslati parallelamente).

I fattori u , v , w sono validi solo nel blocco in cui sono programmati e agiscono solo se sono associati alle coordinate corrispondenti:

- u al primo asse configurato (tipicamente X)
- v al secondo asse configurato (tipicamente Y)
- w al terzo asse configurato (tipicamente Z)

Esempi d'uso dei fattori di compensazione u , v , w

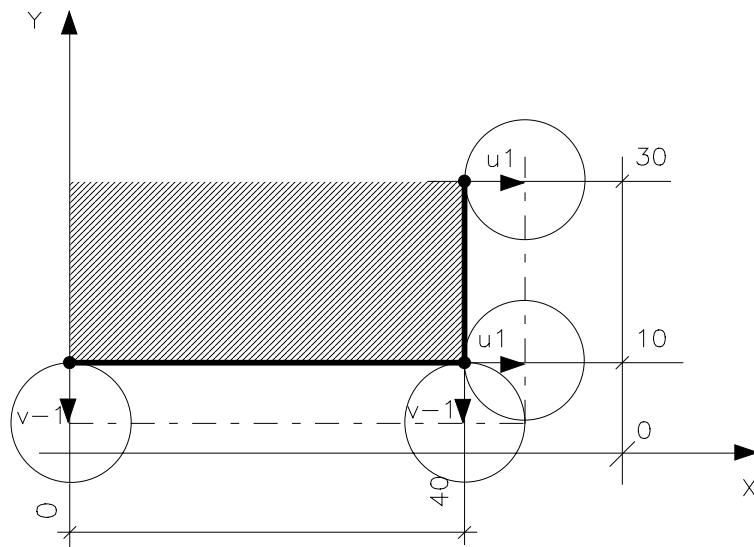
Esempio 1:



Programma:

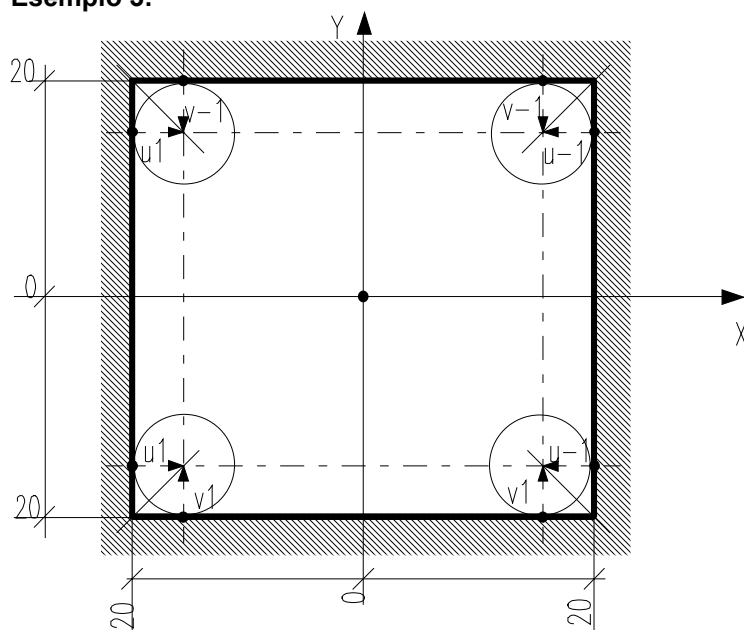
```
N5 T1.01 M6 S. . F. .  
N6 G X Y30  
N7 G1 Y10 v1  
N8 X40 u-1  
N9 Y30  
N10 G X. . Y. .
```

Programma:



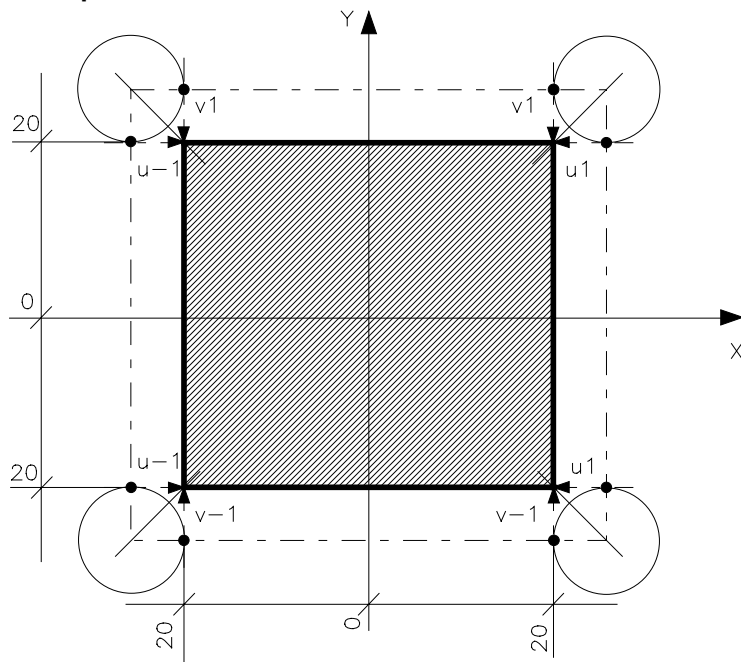
N13 G X Y
N14 Y10 v-1
N15 X40 u1
N16 Y30
N17 G X. . Y. .

Programma:



N13 G X Y
N14 G1 Z-10
N15 X-20 Y-20 u1 v1
N16 X20 u-1
N17 Y20 v-1
N18 X-20 u1
N19 Y-20 v1
N20 G X Y

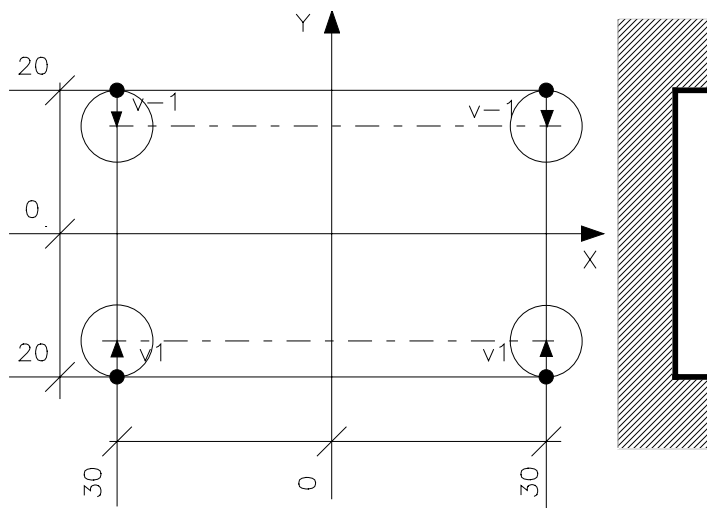
Esempio 4:



Programma:

```
N13 G X-35 Y-35
N14 Z-10
N15 G1 X-20 Y-20 u-1 v-1
N16 X20 u1
N17 Y20 v1
N18 X-20 u-1
N19 Y-20 v-1
N20 GZ
```

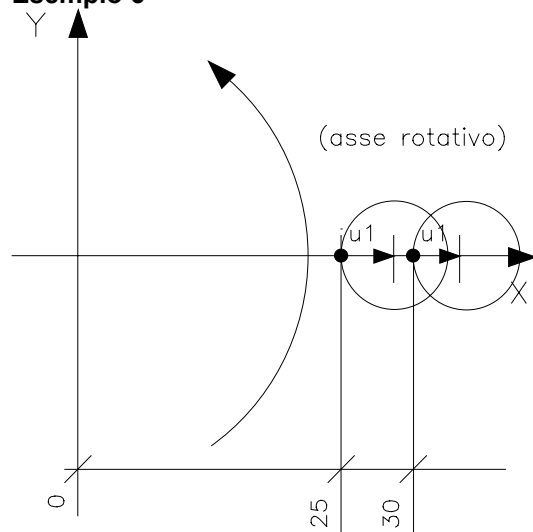
Esempio 5:



Programma:

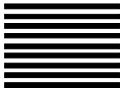
```
N12 ...
N13 G X-30 Y
N14 G1 Y20 v-1
N15 X30
N16 Y-20 v1
N17 X-30
N18 ...
```

Esempio 6



Programma:

```
N12 G X40 Y B0
N13 G1 X30 u1
N14 X25 B360 u1
N15 G X40
```



MSA (UOV) - Definizione di sovrammetallo

Il comando MSA definisce il valore di sovrammetallo da lasciare lungo il profilo. Viene utilizzato durante i cicli di sgrossatura e di pre-finitura. E' possibile programmarlo in un blocco o inserirlo da tastiera o tramite una softkey.

Sintassi

MSA = *valore*

dove:

valore Può essere un numero decimale o un parametro E. Viene programmato nella stessa unità di misura (G70-G71) attiva nel programma.

Caratteristiche:

Il valore del sovrammetallo programmato viene applicato dal sistema nel calcolo della traslazione del profilo quando sono attivi G41 o G42 (compensazione diametro utensile).

La traslazione su profilo, infatti, è la somma tra raggio utensile e MSA.

Esempio:

MSA = 0.5 Assegna un sovrammetallo di 0.5

E30 = 1.5

MSA = E30 Assegna un sovrammetallo di 1.5

CONTORNITURA AUTOMATICA

I comandi di contornitura automatica (GTP e CCP) sono orientati alla lavorazione di profili in correzione raggio (G41/G42), generalmente definiti tramite strumenti grafici.

Gli strumenti grafici, (es. opzione Editor Grafico), traducono il disegno del particolare da lavorare in un programma tecnologico scritto in linguaggio ISO elementare.

Una delle caratteristiche principali di questa prestazione è determinata dal fatto che i profili da lavorare vengono visti come sottoprogrammi.

Tali profili devono essere definiti in un piano (es. XY) e possono contenere solo gli operatori G che determinano il tipo di movimento (G1, G2, G3).

Le prestazioni fornite dai comandi di contornitura sono le seguenti:

- Determinazione automatica del punto di attacco esterno al profilo.
- Esecuzione della lavorazione così come descritta nel sottoprogramma
- Rotazione del profilo, assumendo come origine di rotazione il primo punto del profilo stesso.
- Esecuzione della lavorazione a partire dall'ultimo ente del sottoprogramma, fino ad arrivare al primo (esecuzione al contrario).

In tutti i casi è previsto l'attacco in offset sul primo punto del profilo (primo blocco del programma con movimenti in ascissa/ordinata).

Limiti imposti nell'utilizzo della contornitura automatica

I limiti nell'utilizzo della contornitura automatica sono i seguenti:

- Il profilo deve essere interamente descritto in una subroutine, precedentemente indicata come *nome profilo*.
- Il profilo sarà ritenuto chiuso se il primo punto descritto nella subroutine di part program sarà coincidente con l'ultimo.
- La correzione raggio (*modalità_offset*) verrà attivata automaticamente sul primo blocco della subroutine e disattivata sull'ultimo blocco.
- I blocchi contenuti nella subroutine dovranno essere strettamente di tipo ISO, sarà ammessa la programmazione di:
 - Nomi asse [quote]
 - Funzioni G di movimento (G1, G2, G3 ed eventuali loro operandi)
 - NON saranno quindi presi in considerazione (durante la fase di analisi del profilo) i comandi triletterali in generale e comunque tutti quelli che potrebbero modificare la descrizione del profilo stesso (origini, mirror, fattore di scala).

NOTA:

Il non rispetto dei punti elencati, potrà provocare una errata interpretazione del profilo descritto nella subroutine.



GTP - Get Point

Determina il punto di attacco esterno al profilo.

Sintassi

(**GTP**, nome_profilo, modalità_offset, *E-par*, tipo_attacco , angolo)

dove:

nome_profilo	È una stringa ASCII che rappresenta il nome del part program nel quale è descritto il profilo.
modalità_offset	Può valere 0, 1, 2. Rappresenta la modalità compensazione diametro utensile da utilizzare: 0 = senza compensazione 1 = compensazione con utensile a sinistra del profilo 2 = compensazione con utensile a destra del profilo
<i>E-par</i>	Identifica la variabile nella quale saranno caricate le coordinate del punto di attacco calcolato dal sistema: indice → ascissa indice+1 → ordinata
tipo_attacco	Può valere 0, 1, 2, 3. Identifica il tipo di attacco esterno al profilo; ad ogni codice corrisponde un diverso punto calcolato (vedere pagina seguente).
angolo	Angolo di rotazione: Rappresenta il valore dell'angolo di rotazione con cui dovrà essere eseguito il profilo.

NOTE:

- La distanza tra il punto d'attacco esterno al profilo ed il punto di inizio lavorazione viene determinata in base al valore contenuto nella variabile di sistema MSA.
- È possibile attivare la prestazione che inverte il verso di percorrenza del profilo utilizzando la variabile di sistema TPO (TPO=4 → inversione del profilo).
Per ulteriori informazioni riguardanti la codifica dei valori di TPO, vedere a pag. 4-16.



Programmando la modalità_offset=0, il tipo_attacco viene forzato ad "1" (approccio tangente al profilo) senza ulteriori segnalazioni da parte del sistema.

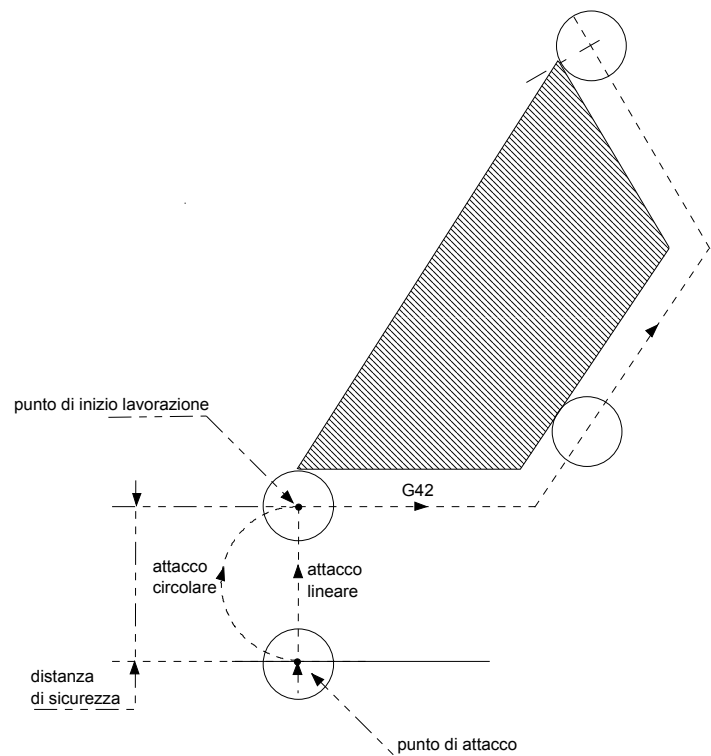
Determinazione del punto di attacco

tipo_attacco=0

Il posizionamento avviene in modo perpendicolare al primo ente del profilo.

L'attacco può essere lineare o circolare in funzione del valore della variabile TPO.

Approccio perpendicolare al profilo

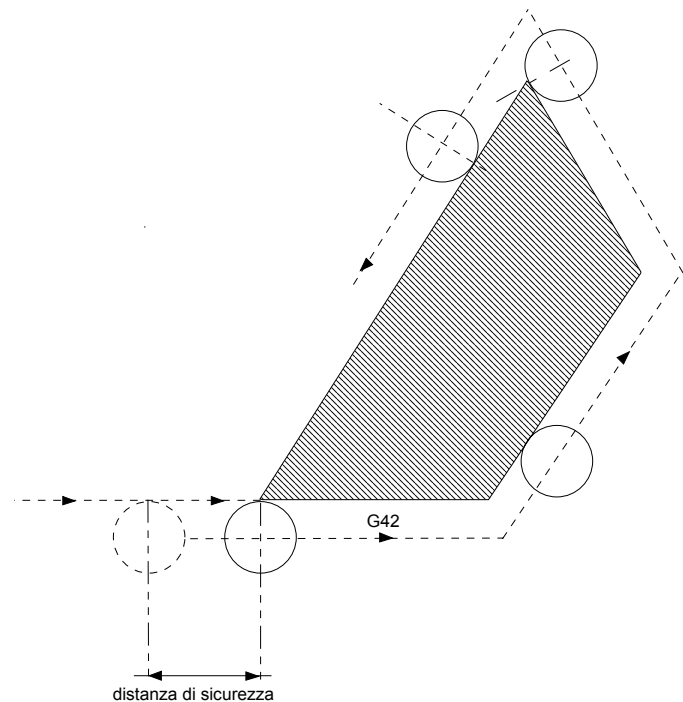


tipo_attacco=1

Il posizionamento avviene in modo tangente al primo ente del profilo.

L'attacco è sempre di tipo lineare.

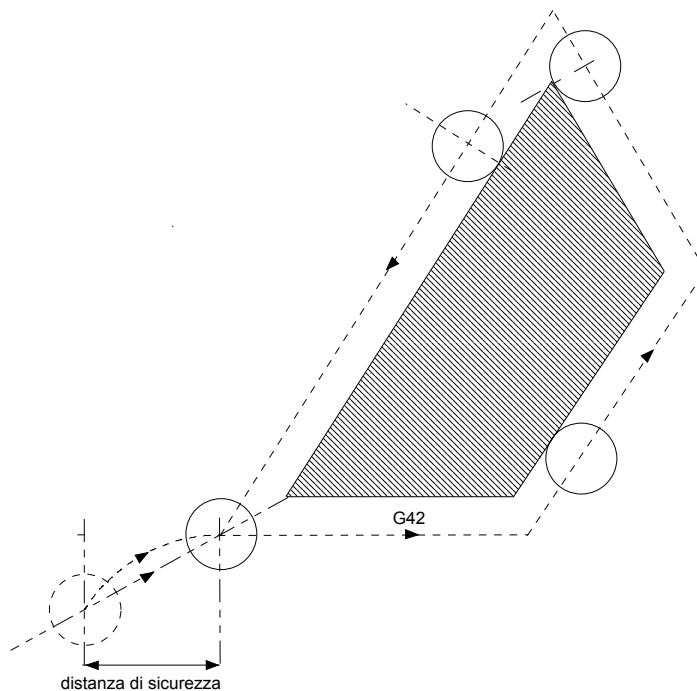
Approccio tangente al profilo



tipo_attacco=2

Il posizionamento avviene sull'intersezione tra il primo e dell'ultimo punto del profilo; l'attacco può essere lineare o circolare in funzione del valore della variabile TPO.

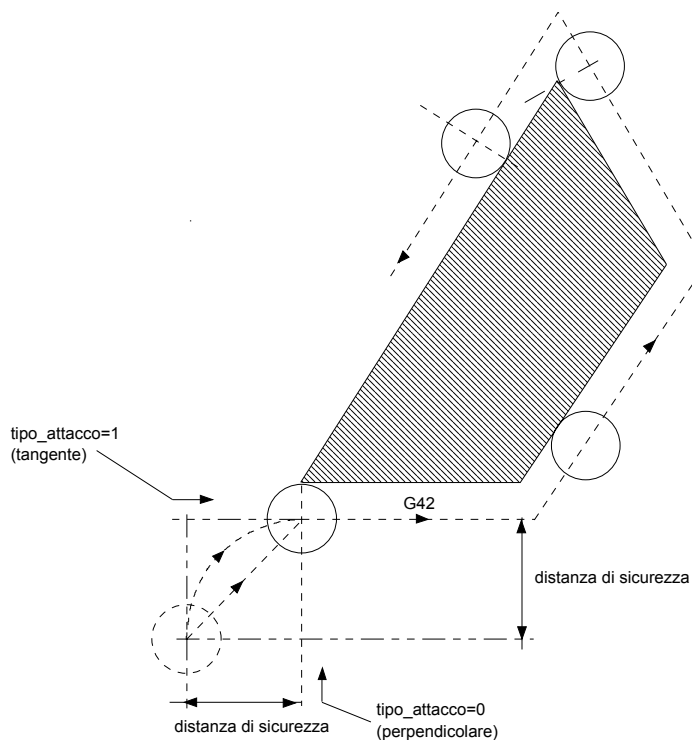
Approccio sull'intersezione del primo ed ultimo punto del profilo.

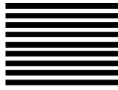


tipo_attacco=3

Il posizionamento avviene sulla bisettrice dell'angolo formato dai movimenti generati con *tipo_attacco=0* e *tipo_attacco=1*; l'attacco può essere lineare o circolare in funzione del valore della variabile TPO.

Approccio sulla bisettrice





CCP - Cutter Compensation Profile

Esegue la contornitura di un profilo, in compensazione diametro utensile.

Sintassi

(**CCP**, nome_profilo, modalità_offset, tipo_uscita, angolo)

dove:

nome_profilo	È una stringa ASCII che rappresenta il nome del part program nel quale è descritto il profilo.
modalità_offset	Può valere 0, 1, 2. Rappresenta la modalità offset con la quale effettuare la lavorazione.
tipo_uscita	Può valere 0, 1, 2, 3. Identifica il tipo di uscita esterno al profilo; ad ogni codice corrisponde un diverso punto calcolato.
angolo	Angolo di rotazione: Determina il valore dell'angolo di rotazione con cui andrà eseguito il profilo.

Caratteristiche:

Il punto di uscita esterno al profilo viene determinato considerando un valore, inteso come distanza di sicurezza.

Tale valore, che associato al tipo di punto di uscita, determina la posizione sul piano del punto di uscita stesso viene prelevato dalla variabile di sistema MSA.

NOTA:

È possibile attivare la prestazione che inverte il senso di percorrenza del profilo utilizzando la variabile di sistema TPO (TPO = 4 → inversione del profilo attiva).

Per ulteriori informazioni riguardanti la codifica dei valori di TPO, vedere a pag. 4-16.

FINE CAPITOLO

PROGRAMMAZIONE DEL MANDRINO



Le funzioni descritte in questo capitolo devono essere gestite dalla logica di macchina.

FUNZIONI DEL MANDRINO

L'indirizzo S programma la velocità di rotazione del mandrino espressa in giri/min o come velocità di taglio costante. Anche alcune funzioni M intervengono nel controllo del mandrino.



G96 G97 - Programmazione CSS e RPM

I seguenti codici G controllano la programmazione della velocità del mandrino.

G96 Programmazione della velocità come velocità di taglio costante (CSS).

G97 Programmazione della velocità in giri/min (RPM).

Sintassi

G97 [*Codici-G*] [*operandi*]

G96 [*Codici-G*] [*operandi*]

dove:

Codici-G Altri codici G compatibili con G96 e G97 (consultare la tabella "Codici G compatibili" del capitolo 1).

operandi Tutti gli operandi e i codici che possono essere utilizzati nei blocchi di funzioni G.

Caratteristiche:

G97 attiva la programmazione dell'indirizzo S come velocità del mandrino espressa in giri/min. G97 è la modalità di default del controllo ed è modale (come G96).

G96 attiva la programmazione dell'indirizzo S come velocità di taglio costante (CSS=Constant Surface Speed), espressa in piedi al minuto (G70), o metri al minuto (G71). G96 impone che la velocità del mandrino sia controllata in funzione della posizione dell'asse diametrale, in modo tale da rimanere costante sulla superficie da lavorare.

Quando viene attivata G96, la posizione dell'asse diametrale viene assunta uguale al raggio per il quale è programmata la velocità di superficie costante (S).

Modifica dell'indirizzo S durante G96 - cancellazione del modo CSS

Per modificare il valore di S mentre G96 è attivo, il controllo deve essere in modo G00 o G29. Per cancellare la programmazione CSS di G96, il controllo deve essere in modo G00 e occorre programmare un blocco con G97 ed un indirizzo S che definisca il numero di giri del mandrino.

Esempio:

L'esempio seguente illustra la programmazione in CSS.

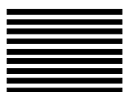
- Codici G di accensione assunti: G00, G27, G70, G95, G97
- Intervallo di accelerazione assunto: intervallo di accelerazione 1 (M41) = 800 giri/min massimo.

```

N1 G90 G00
N3 U5                ;Predisposizione dell'asse diametrale (U)
N4 SSL=700           ;Velocità di rotazione massima = 700 giri/min
N5 G96 S400 M3        ;CSS 400 piedi /minuto
N6 G1 U0 Z -2 F10     ;Impostazione del modo contornitura e velocità di 10 ;pollici/min
N7 U5 Z0
N8 G00               ;Preparazione alla modifica di S
N9 G01 S300 U0 Z -2   ;Nuova velocità di taglio = 300 piedi/minuto
N10 G00              ; Preparazione alla cancellazione di G96
N11 G97 S100          ;Impostazione di G97, S=100 giri/min
N12 G01 Z0            ;Contornitura in G01
N13 U5
N14 G00               ;Modo rapido, contornitura disattivata
N15 M05              ;Arresto del mandrino
N16 M02              ;Fine programma
    
```

IMPORTANTE

L'operatore S non si può programmare quando il processo si trova in stato di Hold.



SSL - Limite della velocità di rotazione del mandrino

Il comando SSL viene introdotto con G96 per definire la massima velocità di rotazione del mandrino durante il funzionamento in CSS espressa in giri/min.

Sintassi

SSL=valore

dove:

valore E' un valore che può essere programmato direttamente con un numero decimale o indirettamente con un parametro E.

Esempio:

SSL = 2000 Assegna una velocità massima di rotazione del mandrino pari a 2000 giri/min

E32 = 1500

SSL = E32 Assegna una velocità massima di rotazione del mandrino pari a 1500 giri/min

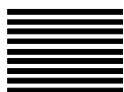
Caratteristiche:

Quando l'asse diametrale si avvicina all'asse del mandrino, il mandrino si avvicina alla velocità massima per mantenere il valore dell'indirizzo S programmato. Il comando SSL mantiene la velocità di rotazione del mandrino ad un valore inferiore ai giri/minuto massimo.

Il comando SSL deve essere programmato prima del numero di giri S.

IMPORTANTE

Assicurarsi di definire questo valore nel programma prima di scrivere i blocchi G96.



M19 - Arresto della rotazione mandrino con orientamento

La funzione ausiliaria M19 programma un arresto del mandrino con un certo orientamento angolare. Questa caratteristica è utile nelle lavorazioni in tiro, poiché consente di orientare il mandrino, spostare l'asse Y (o l'asse X a seconda dell'orientamento del taglio), penetrare nel foro, posizionare nuovamente il mandrino in asse e iniziare la lavorazione.

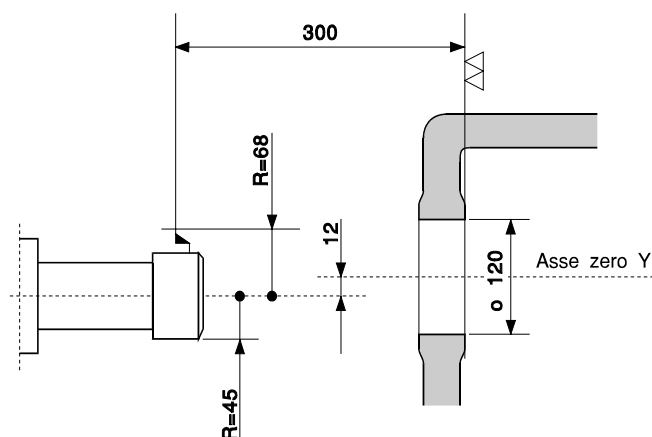
Inoltre M19 può essere utilizzata in operazioni di alesatura di elevata precisione, per evitare l'incisione della superficie alesata durante lo spostamento di ritorno dell'asse Z. In questo caso, si dovrebbe terminare il foro, orientare il mandrino, spostare l'asse X o Y a seconda dell'orientamento del taglio, ed eseguire il ritorno dell'asse Z.

M19 viene annullata da M03, M04, M13, M14. Quando il controllo legge M19 in un blocco che contiene anche informazioni di movimento, M19 precede il movimento.

IMPORTANTE

Le funzioni M attive nella vostra applicazione possono essere differenti da quelle descritte di seguito. Per ulteriori informazioni relative alle funzioni M consultare la documentazione specifica della vostra macchina.

Esempio:



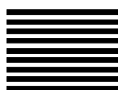
Programma:

```
N32 (DIS,"BARRA PER LAMATURA IN
TIRATA D=136")
N33 S115 F20 T7.7 M6
N34 G X250 Y-12 M19
N35 Z-306
N36 Y M3
N37 DWT=2
N38 G1 G29 G4 Z-300
N39 G Z-302 M5
N40 Y-12 M19
N41 Z
```

Nel blocco N34 vengono posizionati gli assi X ed Y e viene controllato l'orientamento del mandrino in modo che l'utensile possa scorrere attraverso il foro. Nel blocco successivo l'asse Z viene posizionato per l'inizio della lamatura.

Nel blocco N36, l'asse dell'utensile viene spostato fino a farlo coincidere con quello della lamatura.

Nel blocco N38 viene eseguita la lamatura. Nei blocchi successivi, l'utensile viene orientato, poi ritirato dal pezzo in lavorazione e posizionato lungo l'asse Y, in modo che possa scorrere attraverso il foro.



GTS - Condivisione del mandrino

Questo triletterale consente ad ogni processo di rilasciare il suo mandrino, modificarne lo stato oppure acquisirne un nuovo asse in nuova modalità. Per le varie condizioni del mandrino vedere il manuale AMP.

Sintassi

(GTS)

(GTS, *spindle_id*, *mode*)

dove:

spindle_id E' identificativo logico dell'asse mandrino interessato.

mode E' la modalità richiesta, X = esclusiva, S = condivisa (shared).

(GTS)

Serve per rilasciare il mandrino. Con questa sintassi il processo informa il sistema che non intende più utilizzare il suo mandrino e lo mette a disposizione di altri processi. L'asse passa in stato RELEASED solo se il rilascio viene effettuato dall'ultimo processo proprietario, mentre se ancora usato da altri processi continua a rimanere in stato SHARED.

(GTS, *spindle_id*, S)

Serve per condividere con altri processi il mandrino con id *spindle_id* che non deve essere di uso esclusivo di un altro processo. I casi possibili sono:

- 1) Il processo non possiede nessun mandrino e vuole acquisire, condividendolo, un nuovo mandrino.
- 2) Il processo è già proprietario esclusivo dell'asse e vuole renderlo condivisibile.
- 3) Il processo vuole abbandonare il mandrino corrente ed acquisirne uno nuovo.

(GTS, *spindle_id*, X)

Serve per utilizzare il mandrino con id *spindle_id* in modo esclusivo. L'asse non deve essere in condivisione con altri processi. I casi possibili sono:

- 1) Il processo non possiede nessun mandrino e ne vuole uno in modo esclusivo.
- 2) Il processo è l'unico proprietario di un mandrino "condiviso" e ne vuole il pieno possesso.
- 3) Il processo vuole abbandonare il mandrino corrente ed acquisirne uno nuovo.

Nei casi non previsti viene dato errore di processo.

Esempio:

(GTS, 4, S)

Caratteristiche

Il secondo campo (id mandrino) può essere una variabile mentre il terzo (modalità) può assumere solo i valori descritti. Inoltre, la chiamata all'interno del part-program forza il sincronismo di blocco-calcolo.

Capitolo 5

Programmazione del Mandrino

FINE CAPITOLO

FUNZIONI AUSILIARIE M

Le funzioni M standard

Nel seguente capitolo vengono brevemente descritte le funzioni ausiliarie M considerate standard per la maggior parte delle applicazioni. Le funzioni M che sono installate in alcune applicazioni possono essere diverse da quelle descritte in questo contesto.

L'installatore del sistema è responsabile della programmazione dell'interfaccia tra il controllo e la macchina nonché dell'installazione delle funzioni M specifiche per l'applicazione. Consultare la documentazione specifica della macchina per ulteriori informazioni relative alle funzioni M.

CODICE	ATTIVA A INIZIO/FINE MOVIMENTO	FUNZIONE CHE ANNULLA M ATTIVA	DESCRIZIONE
M0	x	Inizio ciclo	Arresto del programma
M1	x	Inizio ciclo	Arresto del programma opzionale
M2	x		Fine del programma
M3	x	M4-M5-M14-M19	Rotazione oraria del mandrino
M4	x	M3-M5-M13-M19	Rotazione antioraria del mandrino
M5	x	M3-M4-M13-M14	Arresto del mandrino
M6	x		Cambio utensile
M7	x	M9	Refrigerante secondario inserito
M8	x	M9	Refrigerante primario inserito
M9	x	M7-M8	Refrigerante disinserito
M10	x	M11	Blocco degli assi
M11	x		Sblocco degli assi
M12	x	M11	Blocco degli assi rotativi
M13	x	M4-M5-M14-M19	Rotazione oraria mandrino e refrigerante
M14	x	M3-M5-M13-M19	Rotazione antioraria mandrino e refrigerante
M19	x	M3-M4-M5-M13-M14	Arresto del mandrino con orientamento
M30	x		Fine del programma e riposizionamento sul primo blocco
M40	x		Reset gamma forzata
M41	x	M42-M43-M44-M40	Forza gamma 1
M42	x	M41-M43-M44-M40	Forza gamma 2
M43	x	M41-M42-M44-M40	Forza gamma 3
M44	x	M41-M42-M43-M40	Forza gamma 4
M45	x	M41-M42-M43-M44	Cambio gamma automatico
M60	x		Cambio pezzo

A seconda della configurazione le funzioni M possono essere disattivate da un reset del controllo.

IMPORTANTE

Un blocco che contiene un codice M "expedite" deve contenere anche un movimento assi.

A seconda del tipo di caratterizzazione della funzione M, può essere possibile la sola programmazione nel modo punto a punto (G29), oppure anche nel modo continuo (G27-G28).

CODICE	DESCRIZIONE DELLA FUNZIONE
M0	Arresto del programma: M0 arresta l'esecuzione del programma, la rotazione del mandrino ed il flusso del refrigerante dopo che il controllo ha eseguito tutte le operazioni del blocco in cui è movimentata. Dopo avere eseguito una M0 il controllo mantiene tutte le informazioni sullo stato corrente.
M1	Arresto opzionale del programma: Se abilitata tramite la softkey appropriata, M1 opera come M0. Se non abilitata, M1 viene ignorata dal controllo.
M2	Fine del programma: M2 definisce la fine di un programma
M3	Rotazione del mandrino in senso orario: M3 definisce la rotazione del mandrino in senso orario. Questa funzione viene attivata dal controllo dopo che è stata introdotta da tastiera o letta in un programma. In un blocco, M3 viene attivata prima di qualsiasi altro movimento.
M4	Rotazione del mandrino in senso antiorario: M4 definisce la rotazione del mandrino in senso antiorario. Questa funzione viene attivata dal controllo dopo che è stata introdotta da tastiera o letta in un programma. In un blocco, M4 viene attivata prima di qualsiasi altro movimento.
M5	Arresto del mandrino: M5 arresta la rotazione del mandrino. Questa funzione viene attivata dal controllo dopo che è stata introdotta da tastiera o letta in un programma. Diventa esecutiva dopo l'esecuzione delle operazioni contenute nel blocco.
M6	Cambio utensile: M6 arresta temporaneamente l'elaborazione del programma e attiva i correttori selezionati tramite la funzione T. In un blocco, viene attivata prima di qualsiasi movimento. Arresta anche la rotazione del mandrino e il flusso del liquido di raffreddamento ma non disattiva le funzioni M3, M4, M7, M8, M13, M14, che ridiventano attive dopo che M6 è completata.
M7	Erogazione del refrigerante secondario: M7 comanda l'erogazione del refrigerante secondario. Diventa esecutiva quando viene introdotta da tastiera o elaborata in un programma.
M8	Erogazione del refrigerante principale: M8 comanda l'erogazione del refrigerante principale. Diventa esecutiva quando viene introdotta da tastiera o elaborata in un programma.

CODICE	DESCRIZIONE DELLA FUNZIONE
M9	Stop refrigerante: M9 blocca completamente l'erogazione del refrigerante (principale e secondario). Diventa operativa dopo l'esecuzione delle operazioni contenute nel blocco.
M10	Blocco degli assi lineari e rotativi: M10 blocca gli assi non interessati alla lavorazione.
M11	Disattivazione M10 e M12
M12	Blocco degli assi rotativi: M12 blocca solo gli assi rotativi non interessati alla lavorazione.
M13	Rotazione del mandrino in senso orario e attivazione del refrigerante
M14	Rotazione del mandrino in senso antiorario e attivazione del refrigerante
M19	Arresto rotazione mandrino con orientamento: Diventa esecutiva prima dei movimenti contenuti nel blocco. Viene annullata dalle funzioni M03, M04, M13, M14.
M30	Reset automatico a fine programma: M30 cancella tutte le informazioni relative al buffer dinamico del controllo. L'origine assoluta 0 viene automaticamente attivata e il programma selezionato viene impostato per la ripresa. M30 non disattiva la correzione dell'utensile in mandrino.
M40	Reset gamma forzata
M41	Forza gamma 1
M42	Forza gamma 2
M43	Forza gamma 3
M44	Forza gamma 4
M45	Cambio gamma automatico
M60	Cambio pezzo

FINE CAPITOLO

PROGRAMMAZIONE PARAMETRICA

La programmazione parametrica utilizza le variabili locali e di sistema per definire i valori geometrici e tecnologici di un ciclo di lavorazione. C'è una differenza fondamentale tra variabili locali e di sistema, che differenzia gli scopi per cui queste variabili vengono usate.

Le variabili di sistema sono memorizzate in un'area di memoria di sistema accessibile da tutti i processi attivi e restano residenti anche quando il sistema viene disattivato.

Le variabili locali sono memorizzate in un'area di memoria locale e sono utilizzabili solo dal processo a cui esse sono riferite.

Il valore attuale di queste variabili, inoltre, non viene mantenuto allo spegnimento del sistema, ma alla successiva riaccensione sarà nuovamente inizializzato con il valore configurato in AMP.

La seguente tabella riassume le variabili disponibili nel sistema:

VARIABILE	TIPO	FUNZIONE
E	locale	Parametri E
!nam	locale	Variabili utente
SN	di sistema	Numero di sistema
SC	di sistema	Carattere di sistema
TIM	di sistema	Temporizzatore di sistema (sola lettura)
@nam	di sistema	Variabili PLUS

Tutte le variabili locali e le variabili di sistema ad eccezione dei Caratteri di Sistema possono essere usate in espressioni matematiche, trigonometriche e calcoli di espressione.

Un'espressione matematica è formata da operatori aritmetici, funzioni e operandi (variabili o costanti numeriche). Gli operatori aritmetici sono:

- addizione (+)
- sottrazione (-)
- moltiplicazione (*)
- divisione (/)

FUNZIONI TRIGONOMETRICHE

Le funzioni trigonometriche disponibili sono elencate nella seguente tabella.

FUNZIONE	DESCRIZIONE
SIN (A)	Calcola il seno di A
COS (A)	Calcola il coseno di A
TAN (A)	Calcola la tangente di A
ARS (A)	Calcola l'arco seno di A
ARC (A)	Calcola l'arco coseno di A
ART (A)	Calcola l'arco tangente di A
SQR (A)	Calcola la radice quadrata di A
ABS (A)	Calcola il valore assoluto di A
INT (A)	Calcola la parte intera di A
NEG (A)	Inverte il segno di A
MOD (A,B)	Calcola il resto della divisione tra A e B

Gli argomenti di una funzione (A,B) possono essere variabili o costanti numeriche. Un'espressione viene risolta dal controllo tenendo conto delle regole matematiche sulle priorità delle parentesi e dei segni. Il risultato viene convertito nel formato della variabile specificata alla sinistra del segno uguale.

IMPORTANTE

Gli argomenti delle funzioni trigonometriche (SIN, COS, TAN) devono essere espressi in gradi. Anche il risultato delle funzioni trigonometriche inverse (ARS, ARC, ART) viene espresso in gradi.

FUNZIONI BOOLEANE

Le funzioni booleane disponibili sono elencate nella seguente tabella.

FUNZIONE	DESCRIZIONE
AND(A,B)	Esegue l'AND al bit tra due numeri di valore compreso tra -32768 e +32767*
OR(A,B)	Esegue l'OR a bit tra due numeri di valore compreso tra -32768 e +32767*
NOT(A)	Complemento a 1 di un numero di valore compreso tra -32768 e +32767*

IMPORTANTE

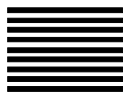
* Gli argomenti delle funzioni booleane devono essere numeri interi di valore compreso tra -32768 e +32767* (formato short con segno). E' possibile utilizzare i parametri E (long real) con le funzioni booleane purchè contengano valori compresi in questo intervallo.

FUNZIONI PER GTL

Le funzioni disponibili per il linguaggio GTL sono elencate nella seguente tabella.

FUNZIONE	DESCRIZIONE
FEL (A,B)	Estrae l'elemento di indice B(1,2,3) dall'ente geometrico Linea (retta) di indice A(1=seno dell'angolo, 2=coseno, 3=distanza della retta dall'origine). Esempio: E30=FEL(5,1) assegna ad E30 il valore del seno dell'angolo che la retta l5 forma con l'ascissa.
FEP (A,B)	Estrae l'elemento di indice B(1,2) dall'ente geometrico Punto di indice A(1=ascissa del punto, 2=ordinata). Esempio: E34=FEP(4,2) assegna ad E34 il valore dell'ordinata del punto p4.
FEC (A,B)	Estrae l'elemento di indice B (1,2,3) dall'ente geometrico Cerchio di indice A (1=ascissa del centro, 2=ordinata, 3=raggio del cerchio). Esempio: E42=FEC(8,3) assegna ad E42 il valore del raggio del cerchio c8.

VARIABILI LOCALI



Parametri E

Il numero massimo di parametri E deve essere definito durante la configurazione del sistema. In teoria, si possono configurare sino a 8000 parametri E.

I parametri E sono di tipo Long Real con 15 cifre in totale, un massimo di 12 integer prima del punto decimale e 9 decimali. Il sistema accetta più parametri E per blocco, la sola restrizione è la lunghezza del blocco. Quando si è in modalità blocco-a-blocco, i blocchi multipli saranno eseguiti come se fossero un singolo blocco. Sono ammessi due livelli di indici parametrici. Per esempio: E(E(E..)).

I parametri E ricevono i valori in speciali blocchi d'assegnazione. Il formato di un blocco di assegnazione è:

E_n = espressione

dove:

n E' il numero d'identificazione del parametro E.

espressione Può essere un valore numerico, un carattere o un'espressione matematica il cui risultato è memorizzato nel parametro E avente il numero di identificazione n .

Esempi:

I seguenti esempi illustrano blocchi d'assegnazione per parametri di calcolo:

E37=(E31*SIN(E30)+123.4567)/SQR(16)	Risolve l'espressione matematica ed assegna il risultato al parametro E37.
E39=-0.00000124+5	Calcola l'espressione e assegna il risultato al parametro E39.
E40=TAN(35)	Trova la tangente di 35 gradi e assegna il risultato al parametro E40.
E31=NEG(E31)	Cambia il segno del parametro E31.
E7=81	Assegna il valore 81 al parametro E7.
E25=E25+30	Aggiunge 30 al valore corrente di E25 e assegna il risultato a E25.
E29=1,2,3,4,5	Assegna il valore 1 al parametro E29, il valore 2 al parametro E30, il valore 3 al parametro E31, il valore 4 al parametro E32 e il valore 5 al parametro E33.

I parametri E possono essere usati sia in un programma che in un sottoprogramma. Per visualizzare il valore corrente di un parametro, utilizzare il comando DIS.

Esempio:

(DIS,E39) visualizza il valore corrente di E39.

Esempio:

Il seguente esempio illustra l'assegnazione di un carattere ASCII

SC0="P" Assegna il carattere "P" alla variabile stringa SC0

E1=SC0

(DIS,E1) Visualizza 80 (codice ASCII della lettera P)

Esempi di blocchi di movimento o comandi con parametri.

XE1

X-E1

X(E1)

X(-E1)

X(E8-14*SQR(E14))

X(-(E8-14*SQR(E14)))

X(E(E(E3)))

FE1

SE2

TE1.E2



! - Variabili utente

Le variabili definite dall'utente possono essere di due tipi:

- Long Real
- Carattere

Il nome delle variabili e il numero massimo di caratteri disponibili devono essere definiti durante la configurazione con AMP.

Il nome delle variabili utente può avere al massimo 8 caratteri, il primo dei quali deve essere "!".

L'estensione delle variabili utente può essere .LR oppure .CH.

Per le variabili utente di tipo carattere è valida la seguente regola:

!nome_var [(indice)] .[numero caratteri]CH = Parametro

dove:

<i>indice</i>	Numero che indica il carattere di partenza nell'array di caratteri della variabile. Se <i>indice</i> non è specificato, viene assunto come zero. Se specificato deve essere programmato tra parentesi tonde.
<i>numero caratteri</i>	Indica quanti caratteri, a partire da <i>indice</i> , sono interessati alla lettura/scrittura. Se non specificato, si intende 1. La somma <i>indice+numero caratteri</i> non deve superare il numero di caratteri configurato per la variabile specificata.
<i>parametro</i>	Può essere: <ul style="list-style-type: none"> – una costante stringa racchiusa tra apici singoli e doppi – una variabile stringa la cui lunghezza non sia più grande di <i>lunghezza</i> – una costante numerica di valore compreso tra 0 e 225 – una variabile numerica il cui valore sia compreso tra 0 e 225

Esempi:

```
!ABC(1) = 125
G0 X(!ABC(1))
```

Il valore 125 viene assegnato alla variabile utente !ABC(1); questa variabile viene quindi usata come argomento dell'indirizzo X in un codice G0.

```
!CHAR(2).8CH="ABC"
```

Questa assegnazione scrive "ABC" nei primi 3 caratteri (a partire dal secondo) nella variabile utente !CHAR: i restanti 5 caratteri (8-3) vengono automaticamente azzerati; per evitare questo azzeramento occorre programmare !CHAR(2).3CH="ABC"
! CHAR(1).CH="A"or!CHAR(1).CH=65

Le variabili utente di tipo Long Real hanno il seguente formato:

!nome_var [(indice)]= Espressione

dove:

indice Numero che identifica la variabile.
Se *indice* non è specificato, viene assunto come zero.
Se specificato deve essere programmato tra parentesi tonde.

espressione Può essere un valore numerico o un'espressione matematica il cui risultato è memorizzato nella variabile utente identificata dall'indice:

Esempio:

```
!ABC(1) = 125
G0 X(!ABC(1))
```

Il valore 125 viene assegnato alla variabile utente !ABC(1); questa variabile viene quindi usata come argomento dell'indirizzo X in un codice G0.

NOTE:

L'indice della variabile può essere un numero o un parametro E.
Nei blocchi di movimento, la variabile utente deve essere scritta sempre tra parentesi.

Esempi di blocchi di movimenti o comando con variabili utente Long Real:

```
X(!USER1(2))
X(!USER1(2)*10)
F(!USER1(1))
S(!USER1(1))
T(!USER1(1).(!USER1(2))
```

NOTA:

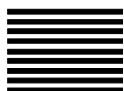
L'indice della variabile può essere un numero o un parametro E.

VARIABILI DI SISTEMA

Ci sono quattro tipi di variabili di sistema che si possono usare nei programmi:

- Numero di Sistema
- Carattere di Sistema
- Temporizzatori di Sistema
- Variabili Plus

All'interno dei programmi queste variabili possono essere usate per leggere o scrivere dei valori o delle stringhe nelle operazioni di assegnazione.



SN - Numero di Sistema

Le variabili Numero di Sistema sono del tipo Long Real a 15 cifre con segno e un massimo di 12 cifre integer. Possono essere definite fino a 25 variabili Numero di Sistema che occupano un'area di 200 byte, disponibile nella memoria dual port del sistema.

Le variabili Numero di Sistema hanno il seguente formato:

SN n = espressione

dove

n E' il numero d'identificazione della variabile Numero di Sistema. Il parametro n può essere un numero o un parametro E.

espressione Può essere un valore numerico o un'espressione matematica il cui risultato è memorizzato nel Numero di Sistema avente numero di identificazione n .

NOTA:

Una variabile Numero di Sistema può essere assegnata ad un'altra variabile numero di Sistema già definito.

Esempi:

1. **SN20 = 326.957**

Il valore decimale 326.957 è assegnato alla variabile Numero di Sistema SN20.

2. **SN20 = (SN9*SIN(30) + 12.5)/SQR(81)**

Il risultato dell'espressione matematica è assegnato alla variabile SN20.

Esempi di blocchi di movimento o comando con variabile SN

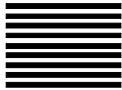
X(SN0)

X(SN0*SN1)

F(SN1)

S(SN2)

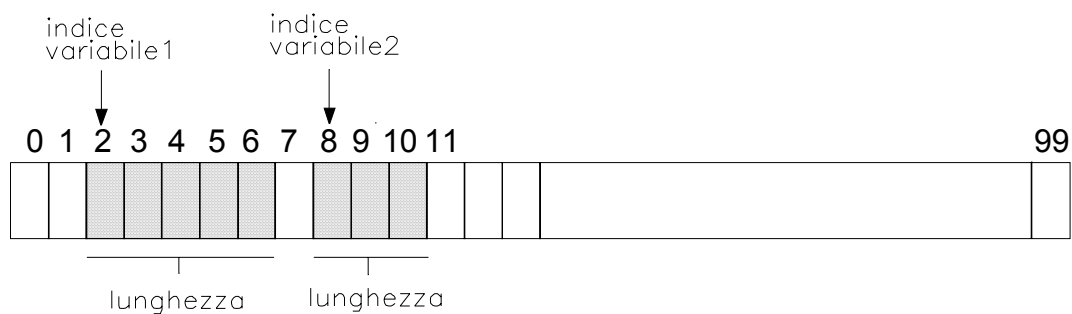
T(SN1).(SN3)



SC - Carattere di Sistema

Le variabili Carattere di Sistema sono di tipo carattere e occupano un'area di 100 byte nella memoria di sistema. Questo significa che l'insieme dei Caratteri di Sistema definiti non può essere superiore ai 100 byte.

Ogni Carattere di Sistema è identificato da un indice che specifica l'indirizzo d'inizio e da una lunghezza che specifica quanti byte occupa la variabile partendo dall'indirizzo d'inizio.



Le variabili Carattere di Sistema hanno il seguente formato:

SC*indice.lunghezza = parametro*

dove:

<i>indice</i>	E' l'indice che specifica la posizione d'inizio della variabile in memoria. Può avere dei valori nell'intervallo da 0 a 99 e può essere un numero o un parametro E.
<i>lunghezza</i>	E' la lunghezza della variabile espressa in numero di caratteri (bytes). Una singola variabile può occupare al massimo 80 caratteri. La lunghezza può essere un numero o un parametro E.
<i>parametro</i>	Può essere: <ul style="list-style-type: none">– una costante stringa racchiusa tra apici singoli o doppi– una variabile stringa la cui lunghezza non sia più grande di <i>lunghezza</i>– una costante numerica di valore compreso tra 0 e 255– una variabile numerica il cui valore sia compreso tra 0 e 255

NOTE:

- Per una singola variabile, la somma "indice+lunghezza" non deve superare 100.
- Una variabile Carattere di Sistema può essere assegnata ad un'altra variabile Carattere di Sistema.
- Alla variabile SC possono anche essere assegnate le variabili numeriche corrispondenti ai caratteri ASCII. In questo modo è possibile assegnare anche quei caratteri che non sono visualizzati, per esempio 10 (LF) e 13 (CR).
- I valori numerici sono programmati senza doppi apici (" ").

Esempio:

SC3.5="PIPPO"

SC9.3="ABC" or SC9.3=65,66,67



La stringa "PIPPO" è scritta partendo dal byte 3 e occupa 5 byte da questa posizione.

La stringa "ABC" è scritta partendo dal byte 9 e occupa 3 byte da questa posizione.

IMPORTANTE

Quando si definiscono l'indice e la lunghezza, occorre fare attenzione a non sovrapporre due variabili.

Esempio:

Assegnazione di un valore numerico ad una variabile SC.

SC0.1=80
(DIS,SC0.1)

Assegna 80 alla variabile stringa
Visualizza P (carattere ASCII 80).

E1=80
SC0.1=E1
(DIS,SC0.1)

Assegna 80 alla variabile E1
Visualizza P (carattere ASCII 80).



TIM - Temporizzatore di Sistema

L'istruzione TIM definisce una variabile usata dal programmatore per leggere il tempo indicato dal temporizzatore del controllo. Il valore della variabile è espresso in secondi.

L'istruzione TIM può essere letta, visualizzata o memorizzata su una variabile di supporto.

Il calcolo di questo valore inizia all'accensione del controllo.

Esempio:

(DIS, TIM)

E10=TIM

NOTA:

Il contenuto della variabile TIM non può essere modificato da part program.



@ - Variabili PLUS

Dal programma è possibile accedere alle variabili PLUS della logica di macchina per operazioni di scrittura o lettura.

Sintassi

@nome

dove:

nome è il nome della variabile dichiarato durante la fase di configurazione (AMP).

Esistono tre tipi di variabili PLUS:

- Short
- Booleane
- Double

Ci sono 256 variabili Short, 128 delle quali sono accessibili da programma. I nomi delle variabili Short a cui si può accedere da programma devono essere configurati in AMP. Le variabili Short sono di 16 bit e possono contenere valori tra -32768 e +32767.

Si può indirizzare un singolo bit di una variabile Short con una variabile Booleana. Ci sono 256 variabili Booleane, 128 delle quali sono accessibili da programma. I nomi delle variabili Booleane a cui si può accedere da programma devono essere configurati in AMP.

Ci sono 64 variabili Double, tutte accessibili da programma. I nomi delle variabili Double a cui si può accedere da programma devono essere configurati in AMP.

Il numero di variabili Booleane, Short e Double deve essere configurato in AMP.

I valori di tutte le variabili PLUS configurate in AMP sono caricati all'accensione del sistema. Se il campo "valore" di una variabile non contiene un valore, la variabile non è inizializzata e mantiene il valore che aveva quando il sistema era spento.

All'interno di un programma, le variabili PLUS possono essere usate:

1. nei blocchi di assegnazione e nei blocchi di triletterali fra parentesi

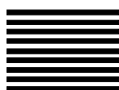
```
E10=@LOG1
(GTO,END,@LOG3=1)
```

2. blocchi di movimento o comando

```
G0 X (@LOG2)
X(@LOG2*2)
F(@LOG2)
S(@LOG3)
T(@LOG2) (@LOG3)
```



L'uso di queste variabili è direttamente collegato alla logica di macchina. Perciò è responsabilità del costruttore della macchina utensile fornire una lista delle variabili usate nella sua applicazione e tutte le informazioni necessarie per il loro corretto impiego.



Variabili L

Le variabili L sono del tipo Long Real a 15 cifre con segno ed un massimo di 12 cifre integer; esse sono 400 e indicizzate da 0 a 399.

Tali variabili coincidono con le variabili della Tabella Utente degli ambienti Table Editor e Plus.

Esse, quindi, sono variabili viste sia dall'ambiente di programmazione che da quello di logica, pertanto possono essere utilizzate per la comunicazione tra i due ambienti o in modo separato.

Esempi:

L10 = 26.9570

L15 = (L10*SIN (30)+9)/SQR(81)

(GTO,END,L2=1)

G0XL15

X(L15)

X(L15*L1)

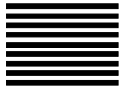
FL1

SL1

TL1.L3



L'uso di queste variabili è direttamente collegato alla logica di macchina. Perciò è responsabilità del costruttore della macchina utensile fornire una lista delle variabili usate nella sua applicazione e tutte le informazioni necessarie per il loro corretto impiego.



Assegnazioni multiple

Con l'operatore di assegnazione multipla $\{ \}$ è possibile assegnare ad un certo numero di variabili il contenuto di altre variabili.

L'assegnazione multipla è accettata solamente per le variabili di tipo numerico.

Sintassi

variabile_destinazione = $\{ \text{variabile_sorgente}, \text{numero_variabili} \}$

dove:

variabile_destinazione E' la prima delle variabili di destinazione.

variabile_sorgente E' la prima delle variabili sorgenti.

numero_variabili E' il numero di variabili da trasferire. Può essere un numero intero od una variabile locale o di sistema

Esempi:

$E0 = \{ \text{SNO}, 4 \}$

equivale a queste quattro assegnazioni:

$E0 = \text{SN0}$

$E1 = \text{SN1}$

$E2 = \text{SN2}$

$E3 = \text{SN3}$

$E100 = 5$

$E50 = \{ \text{LO}, E100 \}$

equivale a queste cinque assegnazioni:

$E50 = \text{L0}$

$E51 = \text{L1}$

$E52 = \text{L2}$

$E53 = \text{L3}$

$E54 = \text{L4}$

FINE CAPITOLO

CICLI FISSI

CICLI FISSI G8N

I codici da G81 a G89 definiscono i cicli fissi che consentono di programmare una serie di operazioni multiple (foratura, maschiatura, alesatura, ecc.) senza ripetere i parametri o i comandi di ogni singola operazione.

Nel blocco contenente la dichiarazione di ciclo fisso G81-G89 non occorre programmare alcun movimento assi. Il ciclo viene così memorizzato ma non ancora eseguito.

Il ciclo comincia ad essere applicato dal blocco successivo alla definizione del ciclo fisso G81-G89. Per ripetere un ciclo dopo averlo eseguito una volta è sufficiente programmare le coordinate di partenza del ciclo successivo.

Gli "assi mandrino" per il ciclo fisso possono essere assegnati nel blocco di definizione del ciclo fisso. Per esempio, nel blocco G81 R Y-20, l'asse Y è "l'asse mandrino" per il ciclo fisso.

Le funzioni G8n sono modali. Prima di programmare un nuovo ciclo fisso bisogna cancellare il precedente usando G80. La funzione G80 deve essere programmata in un blocco dopo l'ultimo ciclo da eseguire. Non è possibile programmare un G8n se è attiva la correzione del profilo (G41/G42).

Quando un ciclo fisso richiede un tempo di sosta (G82, G83, G89), ci sono due differenti modi per ottenerlo:

- usare come valore di default il tempo definito in AMP
- programmare un blocco contenente la variabile DWT = tempo (in secondi)

IMPORTANTE

I cicli fissi possono anche essere eseguiti su assi virtuali

Caratteristiche del ciclo fisso

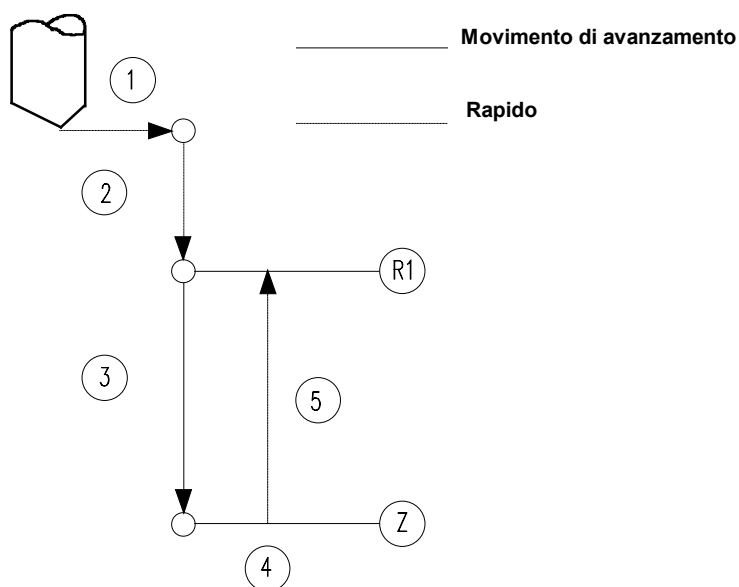
La seguente tabella elenca i cicli fissi disponibili con le relative caratteristiche.

CICLO FISSO		DISCESA	SOSTA	ROTAZIONE	RITORNO
G81	foratura	in lavoro	no	normale	in rapido
G82	lamatura	in lavoro	si	normale	in rapido
G83	foratura profonda con scarico truciolo	in lavoro intermittente (discesa in lavoro intervallata da risalite in rapido)	no	normale	in rapido
G84	maschiatura	in lavoro avviamento rotazione mandrino	no	rotazione invertita	in lavoro ad R1 in rapido ad R2 se presente
G85	alesatura o maschiatura con Tapmatic	in lavoro	no	normale	in lavoro ad R1 in rapido ad R2 se presente
G86	barenatura	in lavoro avvio rotazione mandrino	no	arresto stop	avviamento rapido
G89	barenatura con lamatura	in lavoro	si	normale	in lavoro ad R1 in rapido ad R2 se presente
G80	cancella cicli fissi				

Movimenti di un ciclo fisso

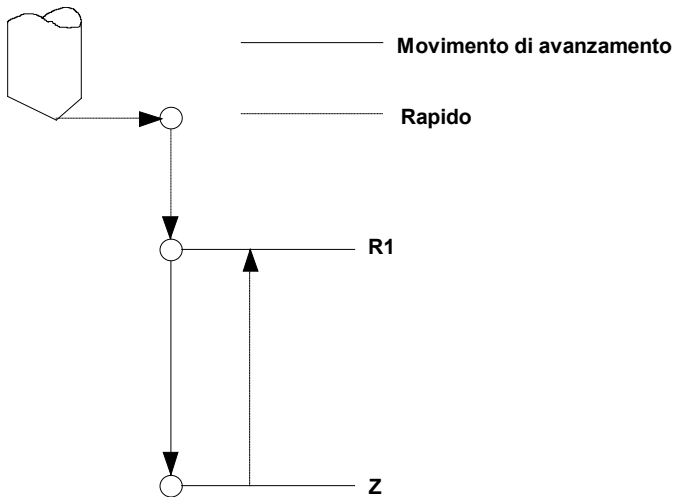
Quando viene programmato un ciclo fisso, gli assi seguono questa sequenza di movimenti.

1. Posizionamento rapido sull'asse del foro
2. Accostamento rapido al piano di lavoro (quota R1)
3. Avanzamento a velocità di lavoro fino alla quota programmata (Z)
4. Funzioni di ciclo a fondo foro
5. Ritorno rapido o a velocità di avanzamento di lavoro alla quota R (R2 se la quota di ritorno è diversa da quella di avvicinamento R1).



Esempi di cicli fissi

In questa sezione vengono mostrati due casi tipici di cicli fissi.



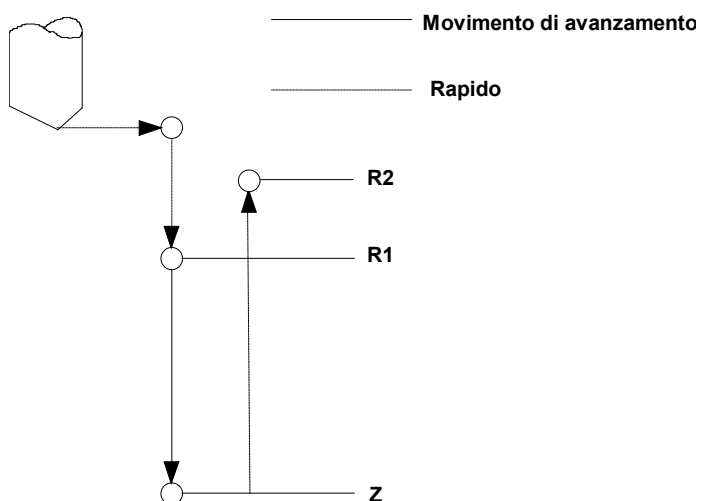
Il seguente è un ciclo fisso con quota di avvicinamento uguale alla quota di ritorno.

G81 R1 . . Z. .

R1 = quota di avvicinamento

Z = profondità del foro

Il seguente è un ciclo fisso con la quota di ritorno diversa dalla quota di avvicinamento.



G81 R1. . R2. . Z. .

R1 = quota di avvicinamento

R2 = quota di ritorno

Z = profondità del foro



G81 - Ciclo fisso di foratura

Sintassi

G81 [*codici-G*] **R1..** [**R2..**] **Z..** [**F..**] [*ausiliaria*]

dove:

codici-G

Altri codici G compatibili con il ciclo fisso G81 (consultare "Codici G compatibili" nel capitolo 1).

R1

Quota di avvicinamento in rapido al piano di lavorazione. Può essere programmata direttamente con numeri decimali o indirettamente con parametri E. La quota di avvicinamento è obbligatoria.

R2

Definisce la quota di ritorno dopo la lavorazione. Questo parametro viene dato con l'indirizzo R seguito dal valore di ritorno. In assenza di questa quota, la quota di avvicinamento (R) viene assunta automaticamente dal controllo come quota di ritorno. Questa coordinata può essere programmata direttamente con numeri decimali o indirettamente con un parametro E.

Z

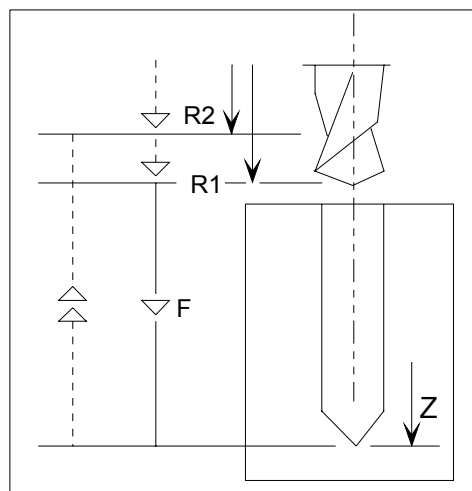
Definisce la quota della profondità del foro (tipicamente Z). E' data con l'indirizzo Z seguito dal valore della profondità del foro. Tale valore può essere programmato direttamente con un numero decimale o indirettamente con un parametro E.

F

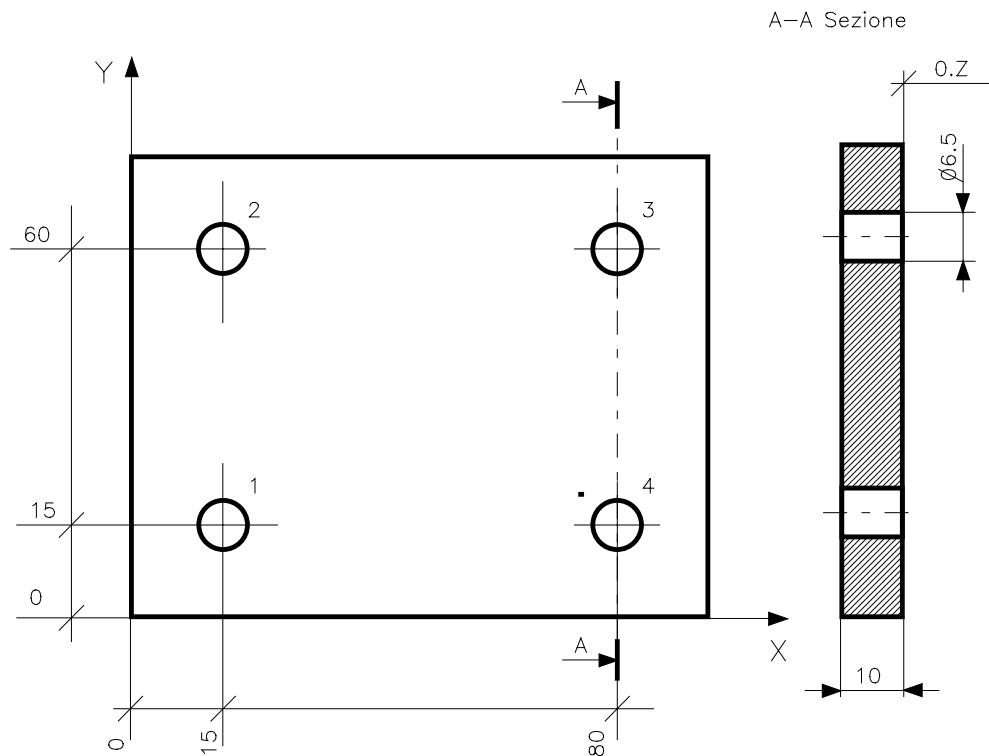
Definisce la velocità di avanzamento utilizzata durante l'operazione di ciclo fisso. Può essere programmata con l'indirizzo F seguito dal valore.

ausiliaria

Funzione programmabile ausiliaria: M, S, T. In un blocco si possono programmare fino a 4 funzioni M, una S (velocità di rotazione mandrino) ed una T (selezione dell'utensile).



Esempio:

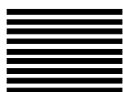


NOTA:

0.Z significa posizione Z=0

Programma:

```
(UGS,X,0,110,Y,0,80)
N31 (DIS,"PUNTA A ELICA D=6.5")
N32 S1100 T3.03 M6
N33 G81 R3 Z-15 F95 M3
1 N34 X15 Y15
2 N35 Y60
3 N36 X80
4 N37 Y15
N38 G80 Z50 M5
N39 M30
```



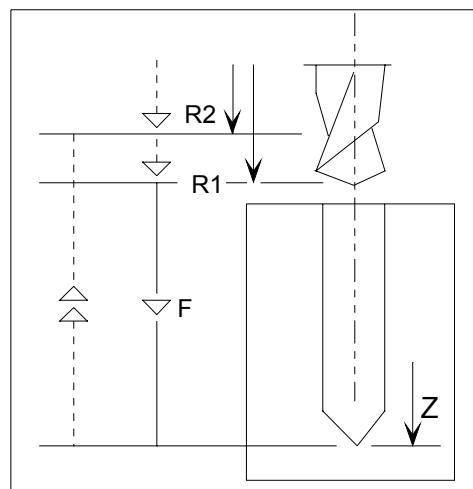
G82 - Ciclo di lamatura

Sintassi

G82[*codici-G*] **R1**.. [**R2**..] **Z**.. [**F**..] [*ausiliaria*]

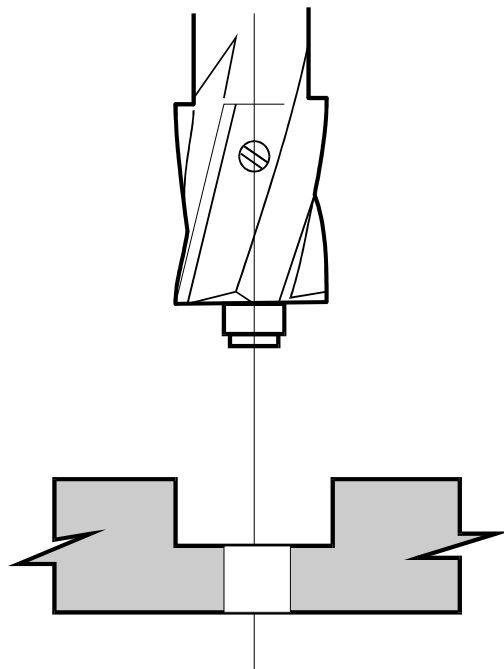
dove:

<i>codici-G</i>	Altri codici G compatibili con il ciclo fisso G82 (consultare la tabella "Codici G compatibili" nel capitolo 1).
R1	Quota di inizio foro (vedere G81).
R2	Quota di ritorno (vedere G81).
Z	Quota di fine foro (vedere G81).
F	Velocità di avanzamento (vedere G81).
<i>ausiliaria</i>	Altre funzioni M, S, T, (vedere G81).

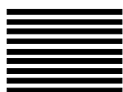


Esempio:

La figura seguente illustra una lavorazione di lamatura.



G82 R5 Z-15 F.. M..
X.. Y..



G83 - Ciclo di foratura profonda

Sintassi

G83 [*codici-G*] **R1..** [**R2..**] **Z..** **I..** [**J..**] [**K..**] [**F..**] [*ausiliaria*]

dove:

codici-G

Altri codici G compatibili con il ciclo fisso G83 (consultare la tabella "Codici G compatibili" nel capitolo 1).

R1

Quota di inizio foro (vedere G81).

R2

Quota di ritorno (vedere G81).

Z

Quota di fine foro (vedere G81).

I

Definisce l'incremento della profondità dopo ciascun ciclo di estrazione, per lo scarico del truciolo. E' dato dall'indirizzo I seguito dal valore.

J

Definisce il minimo incremento della profondità dopo il quale il ciclo procede con incrementi costanti. E' dato dall'indirizzo J seguito dal valore.

K

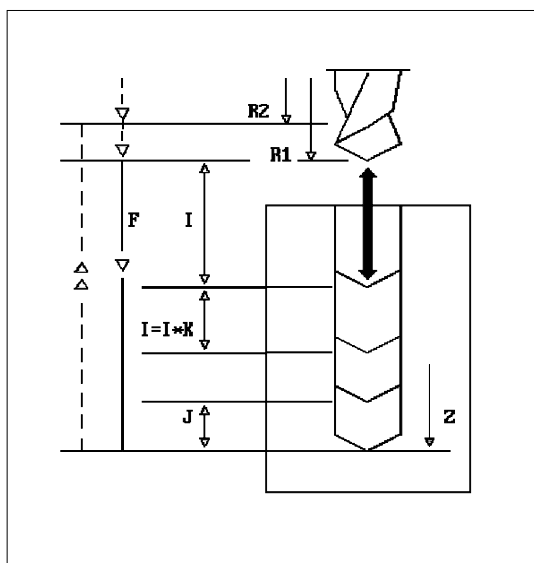
Definisce il fattore di decremento di I fino a raggiungere J. E' dato dall'indirizzo K seguito dal valore.

F

Velocità di avanzamento (vedere G81).

ausiliaria

Altre funzioni M, S, T, (vedere G81).



Caratteristiche:

Il codice G83 può generare movimenti differenti a seconda che i parametri **I**, **J** e **K** siano programmati oppure no.

Se i parametri **I**, **J** e **K** sono programmati, vengono generati i seguenti spostamenti:

1. Avvicinamento in rapido sull'asse del foro da eseguire
2. Avvicinamento in rapido alla quota R1.
3. Avanzamento a velocità di lavoro alla quota R1 + I
4. Ritorno in rapido alla quota R1 (scarico truciolo)
Il ritorno alla R iniziale avviene ad ogni movimento di scarico truciolo.
5. Avvicinamento in rapido alla quota:

$R = R_{precedente} + I - X$ (dove X vale 1 per default o, nel caso di programmazione della variabile DRP, assume il valore assegnato a tale variabile.)

6. Calcolo del nuovo valore di I con le formule:

$$\begin{array}{ll} I = I_{precedente} * K & \text{se } I * K > J \\ I = J & \text{se } I * K < J \end{array}$$

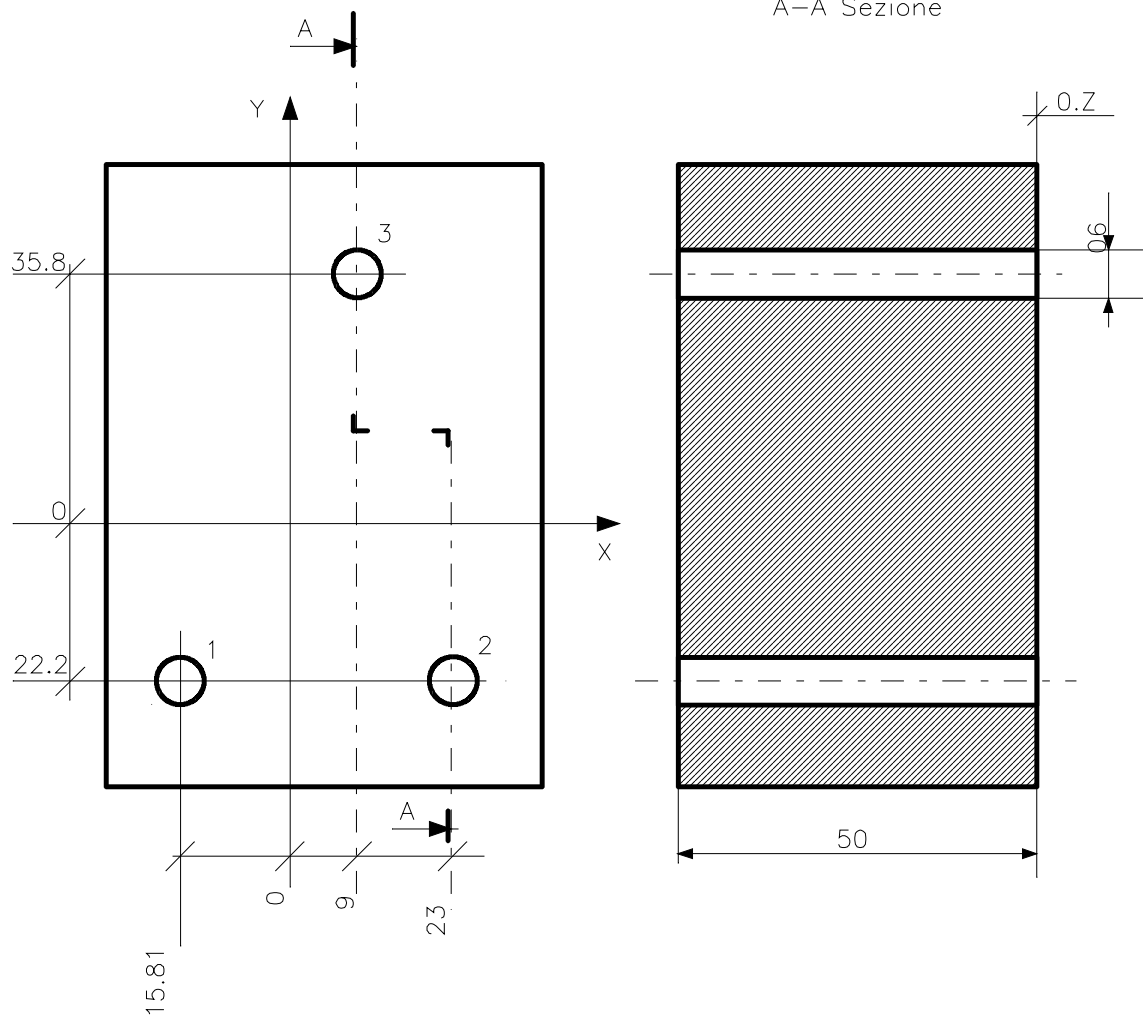
7. Ripetizione dei punti 2-3-4-5-6 fino al raggiungimento della quota finale in Z.

Ciclo rompi truciolo

Se i parametri **J** e **K** non sono programmati, vengono generati i seguenti spostamenti:

1. Avvicinamento in rapido sull'asse del foro
2. Avvicinamento in rapido a quota R1
3. Avanzamento a velocità di lavoro alla quota R + I
4. Sosta dell'asse mandrino per il tempo programmato nel DWT od in mancanza di questo per il tempo di sosta caratterizzato.
5. Ripetizione dei punti 3 e 4 fino al raggiungimento della quota finale in Z.

Esempio:

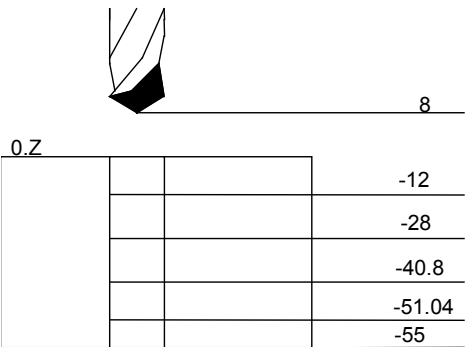


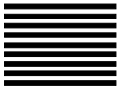
NOTA:

0.Z significa posizione Z=0

Programma:

N65 (DIS,"PUNTA A ELICA D=6")
N66 S930 F65 T6.6 M6
N67 G83 R18 Z-55 I20 K.8 J6 M13
1 N68 X-15.81 Y-22.2
2 N69 X23
3 N70 X9 Y35.8
N71 G80 Z50 M5
N72 M30





DRP – Distanza di ripresa foro G83

Il comando **DRP** definisce in mm la distanza di ripresa foro nei cicli G83 (con IJK programmati). Tale valore viene inizializzato ad 1 dal sistema. Può essere assegnato da Tastiera o da Part Program.

Sintassi

DRP=valore

dove:

valore può essere programmato direttamente o indirettamente con un parametro E.

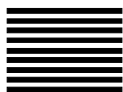
Esempio:

DRP=0.01 ripresa foro a 0.01mm

DRP=5 ripresa foro a 5 mm



Il RESET ripristina il valore 1.



G84 - Ciclo di maschiatura senza trasduttore

Questo codice G84 opera con un mandrino senza trasduttore. Il mandrino deve essere nella modalità esclusiva di appartenenza ad un processo.

Sintassi

G84 [*codici-G*] R1.. [R2..] Z.. [F..]

dove:

codici-G

Altri codici G compatibili con il ciclo fisso G84 (consultare la tabella "Codici G compatibili" nel capitolo 1).

R1

Quota di avvicinamento in rapido e ritorno in lavoro obbligatoria.

R2

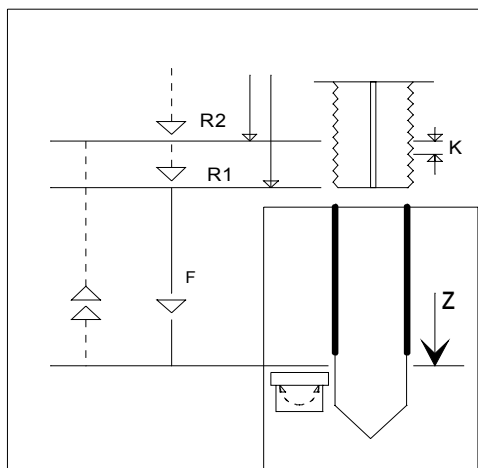
Quota di ritorno in rapido.

Z

Definisce la quota di fine maschiatura (normalmente Z). Viene data con l'indirizzo Z seguito dal valore programmato direttamente con un numero decimale o indirettamente con un parametro E.

F

Definisce la velocità di avanzamento usata quando viene eseguita un'operazione di ciclo fisso. Viene programmata con un indirizzo F seguito dal valore della velocità di avanzamento.



Caratteristiche:

L'accostamento in rapido dell'utensile al pezzo in lavorazione dovrebbe sempre terminare ad una distanza dal pezzo uguale a cinque volte il passo di maschiatura (per profondità di maschiatura inferiori o uguali a 3 diametri) o sette volte il passo di maschiatura (per profondità superiori a 3 diametri).

La velocità di avanzamento da programmare per la maschiatura deve essere calcolata tramite la seguente formula:

$$F = S * p * 0.9$$

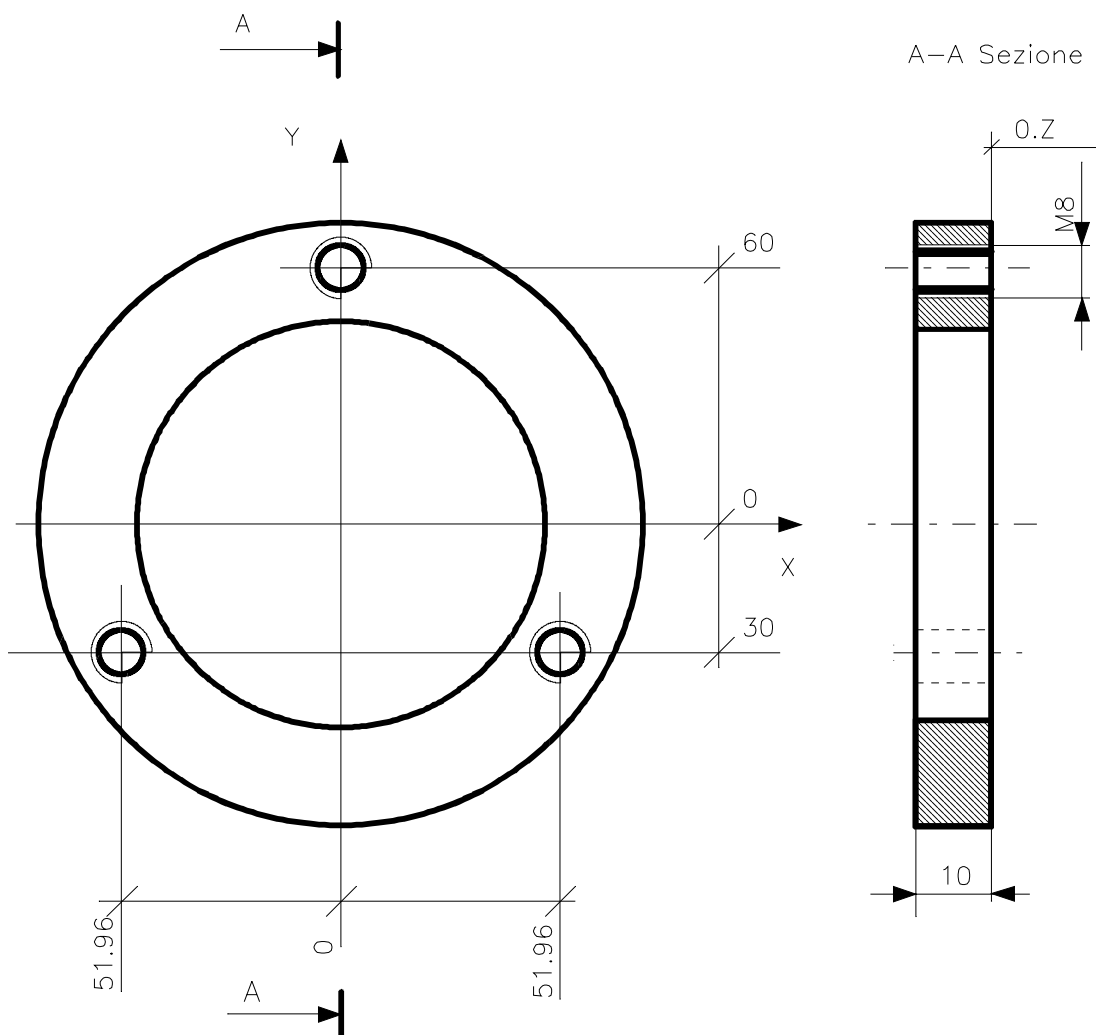
dove:

S	E' la velocità di rotazione del mandrino
p	E' il passo del maschio
0.9	E' il fattore di decremento della velocità di avanzamento per mantenere in tiro il compensatore a molla del portautensili.

La quota finale in Z deve essere decrementata di una quantità pari al 10% della corsa effettiva di lavoro del maschio.

Se la corsa di lavoro Z non è sufficientemente lunga per consentire all'asse di raggiungere la velocità di avanzamento programmata e di fermarsi con decelerazione calcolata in base al tempo di arresto della rotazione del mandrino, viene segnalato errore.

In un blocco G84 non sono ammesse funzioni M.

Esempio:**NOTA:**

0.Z significa posizione Z=0

Programma:

(UGS,X,-60,60,Y,-70,70)

N90 (DIS,"MASCHI M8 - COMPENS. A TRAZIONE")

N91 S280 F315 T8.8 M6 M13

N92 G84 R7 Z-15

N93 X-51.96 Y-30

N94 X51.96

N95 X Y60

N96 G80 Z50 M5

Questo programma è valido per operazioni di maschiatura destrorse perché nel blocco N91 è stata programmata la funzione M13. Per operazioni di maschiatura sinistrorse, è sufficiente programmare la funzione M14 (o M04) invece di M13 (o M03).



G84 - Ciclo di maschiatura con trasduttore montato sul mandrino

Sintassi

G84 [*codici-G*] **R1**.. [**R2**..] **Z**.. **K**.. [**F**..] [*ausiliaria*]

dove:

<i>codici-G</i>	Altri codici G compatibili con il ciclo fisso G84 (consultare la tabella "Codici G compatibili" nel capitolo 1).
R1	Quota di avvicinamento in rapido e ritorno in lavoro obbligatoria.
R2	Quota di ritorno in rapido.
Z	Definisce la quota di fine maschiatura (normalmente Z). Viene data dall'indirizzo Z seguito dal valore programmato direttamente con un numero decimale o indirettamente con un parametro E.
K	Definisce il passo di filettatura del maschio. Viene dato dall'indirizzo K seguito da un valore.
F	Definisce la velocità di avanzamento utilizzata durante un'operazione di ciclo fisso. E' programmata dall'indirizzo F seguito dal valore.
<i>ausiliaria</i>	Altre funzioni M, S, T, (vedere G81).

Caratteristiche:

Quando viene montato il trasduttore sul mandrino, il codice G84 può essere programmato in due modi:

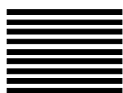
- Usando la velocità di avanzamento F, calcolata come nel caso precedente.
- Usando il passo di filettatura K. In tal caso il controllo calcola automaticamente la velocità di avanzamento moltiplicando per K il numero di giri del mandrino.



Durante un ciclo di maschiatura il controllo non prende in considerazione il pulsante CYCLE STOP (tranne che nel tratto di avvicinamento in rapido) ed il selettore (o la softkey) di FEED OVERRIDE. La disabilitazione del selettore di SPINDLE SPEED OVERRIDE, deve essere eseguita dalla logica di macchina. Per abortire il ciclo di maschiatura si può utilizzare la funzione di logica "INTP-ABO" (vedere descrizioni delle variabili sui manuali Plus).

Esempio:

```
N90 (DIS,"MASCHIO M8")
N91 S280 T8.8 M6 M3
N92 G84 R7 Z-15 K1
N93 X-51.96 Y-30
N94 X51.96
N95 X Y60
N96 G80 Z50 M5
```



G84 - Ciclo di maschiatura rigida con trasduttore montato sul mandrino

Sintassi

G84 [*codici-G*] **R1**.. [**R2**..] **Z**.. **K**..*[ausiliaria]*

dove:

<i>codici-G</i>	Altri codici G compatibili con il ciclo fisso G84 (consultare la tabella "Codici G compatibili" nel capitolo 1).
R1	Quota di avvicinamento in rapido e ritorno in lavoro obbligatoria.
R2	Quota di ritorno in rapido.
Z	Definisce la quota di fine maschiatura (normalmente Z). Viene data dall'indirizzo Z seguito dal valore programmato direttamente con un numero decimale o indirettamente con un parametro E.
K	Definisce il passo di filettatura del maschio. Viene dato dall'indirizzo K seguito da un valore.
<i>ausiliaria</i>	Altre funzioni M, S, T, (vedere G81).

Caratteristiche:

Per la corretta realizzazione del ciclo di maschiatura rigida è necessaria una predisposizione, realizzata tramite il caricamento di due parametri di macchina con i valori appropriati, che tipicamente variano da macchina a macchina.

I parametri interessati sono denominati TKG e TAG.

Sarà quindi necessario fare precedere i cicli di maschiatura rigida dall'assegnazione di questi due parametri con i valori ricavati in fase di installazione della macchina.

Si consiglia di inserire il caricamento di questi dati nelle variabili AMP ad esse associate oppure in una paramacro (per esempio G840) da chiamarsi una volta sola nell'ambito del programma che effettua la maschiatura rigida.

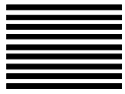
La paramacro sarà quindi così strutturata:

TKG=.....
TAG=.....



A fronte dell'esecuzione di un comando di reset (che peraltro ferma il programma in esecuzione) i parametri TKG e TAG vengono cancellati e necessitano di una nuova assegnazione per la corretta esecuzione della maschiatura rigida.

Per le modalità di calcolo dei corretti valori dei parametri si rimanda al manuale AMP(Cap.1)



TRP (RMS) - Percentuale di ritorno di maschiatura

Il comando TRP definisce la variazione percentuale della velocità di ritorno in un ciclo di maschiatura. Questo comando viene scritto normalmente nel programma, ma può anche essere introdotto da tastiera o tramite softkey.

Sintassi

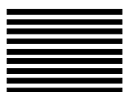
TRP = *valore*

dove:

valore Può essere programmato direttamente con un numero integer o indirettamente con un parametro E di tipo byte.

Esempio:

TRP = 110	rappresenta +10% della F programmata
TRP = 10	rappresenta - 90% della F programmata



G85 - Ciclo di alesatura (o di maschiatura tramite Tapmatic)

Sintassi

G85 [*codici-G*] **R1**.. [**R2**..] **Z**.. [**F**..] [*ausiliaria*]

dove:

codici-G

Altri codici G compatibili con il ciclo fisso G85 (consultare la tabella "Codici G compatibili" nel capitolo 1).

R1

Quota di avvicinamento in rapido e ritorno in lavoro obbligatoria.

R2

Quota di ritorno in rapido.

Z

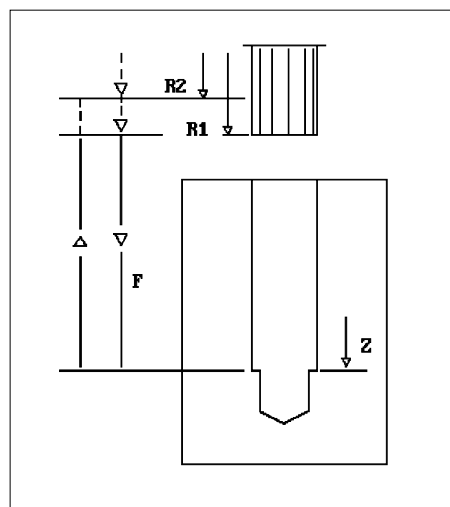
Definisce la quota di fine maschiatura (normalmente Z). Viene data dall'indirizzo Z seguito dal valore programmato direttamente con un numero decimale o indirettamente con un parametro E.

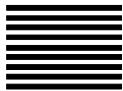
F

Definisce la velocità di avanzamento utilizzata durante un'operazione di ciclo fisso. E' programmata dall'indirizzo F seguito dal valore.

ausiliaria

Altre funzioni M, S, T, (vedere G81).





G86 - Ciclo di barenatura

Il mandrino deve essere nella modalità esclusiva di appartenenza ad un processo.

Sintassi

G86 [codici-G] R1.. [R2..] Z.. [F..] [ausiliaria]

dove:

codici-G

Altri codici G compatibili con il ciclo fisso G86 (consultare la tabella "Codici G compatibili" nel capitolo 1).

R1

Quota di avvicinamento in rapido e ritorno in lavoro obbligatoria.

R2

Quota di ritorno in rapido.

Z

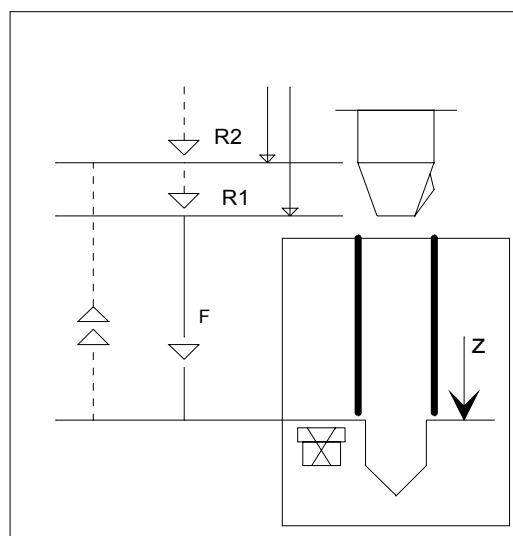
Definisce la quota di fine maschiatura (normalmente Z). Viene data dall'indirizzo Z seguito dal valore programmato direttamente con un numero decimale o indirettamente con un parametro E.

F

Definisce la velocità di avanzamento utilizzata durante un'operazione di ciclo fisso. E' programmata dall'indirizzo F seguito dal valore.

ausiliaria

Altre funzioni M, S, T, (vedere G81).





G89 - Ciclo di barenatura con lamatura

Sintassi

G89 [*codici-G*] **R1**.. [**R2**..] **Z**.. [**F**..] [*ausiliaria*]

dove:

codici-G

Altri codici G compatibili con il ciclo fisso G89 (consultare la tabella "Codici G compatibili" nel capitolo 1).

R1

Quota di avvicinamento in rapido e ritorno in lavoro obbligatoria.

R2

Quota di ritorno in rapido.

Z

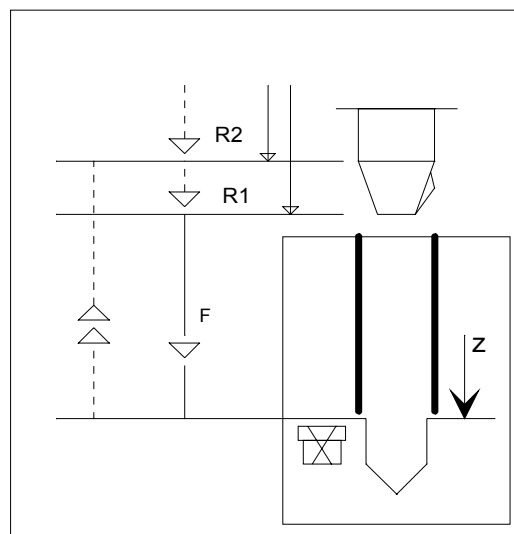
Definisce la quota di profondità del foro (normalmente Z). E' data dall'indirizzo Z seguito dal valore di profondità del foro. Il valore di profondità può essere programmato direttamente con un numero decimale o indirettamente con un parametro E.

F

Definisce la velocità di avanzamento utilizzata durante un'operazione di ciclo fisso. E' programmata dall'indirizzo F seguito dal valore.

ausiliaria

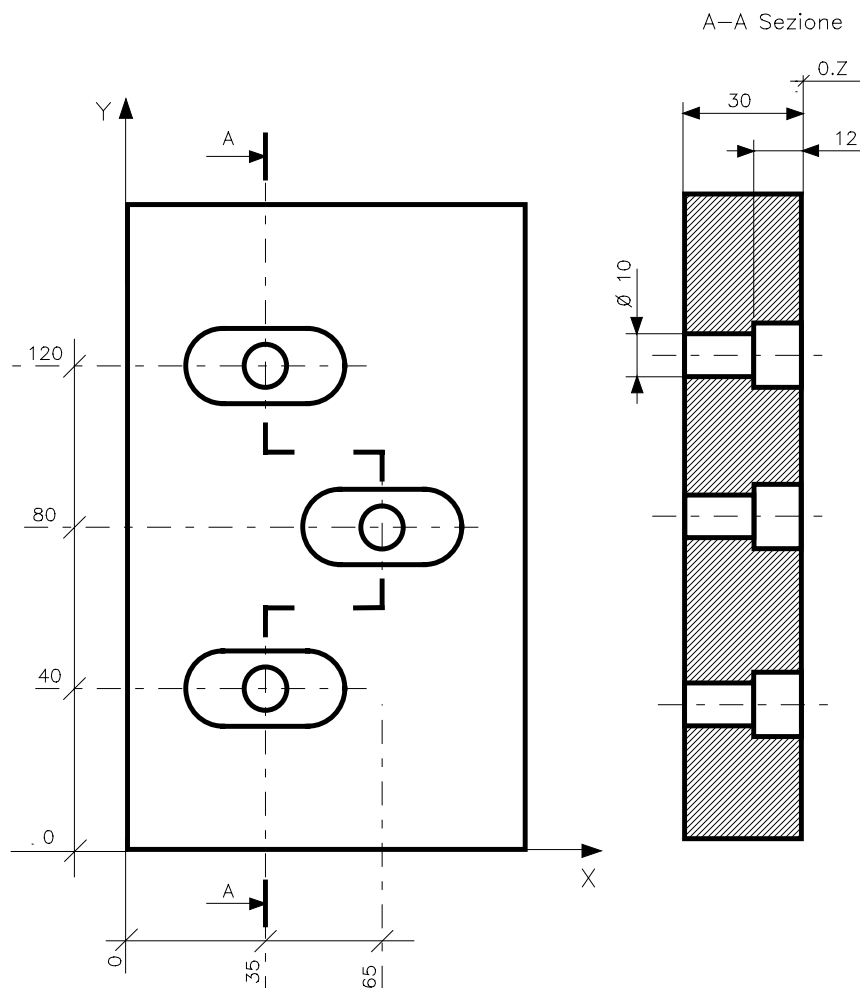
Altre funzioni M, S, T, (vedere G81).



Uso di due quote R in un ciclo fisso

Si può utilizzare questo tipo di programmazione di ciclo fisso quando devono essere eseguiti, maschiati, alesati, ecc. dei fori che sono sullo stesso piano ma separati da ostacoli (staffe, fori ripetitivi su sotto-piani, ecc.).

Esempio:



NOTA:

0.Z significa posizione Z=0

Programma:

```
N41 (UGS,X,0,100,Y,0,150)
N42 (DIS,"PUNTA AD ELICA D=10")
N43 S850 F100 T4.4 M6
N44 G81 R-10 R2 Z-36 M3
N45 X35 Y40
N46 X65 Y80
N47 X35 Y120
N48 G80 Z50 M5
```

Aggiornamento delle quote del ciclo fisso

Quando un ciclo fisso è attivo, è possibile programmare blocchi di avvicinamento in rapido, quote di ritorno e quote di profondità per aggiornare le fasi del ciclo senza dover cancellare e successivamente riprogrammare il ciclo.

I blocchi di programma formati dagli elementi X, Y, R1, R2, Z vengono eseguiti nel seguente ordine:

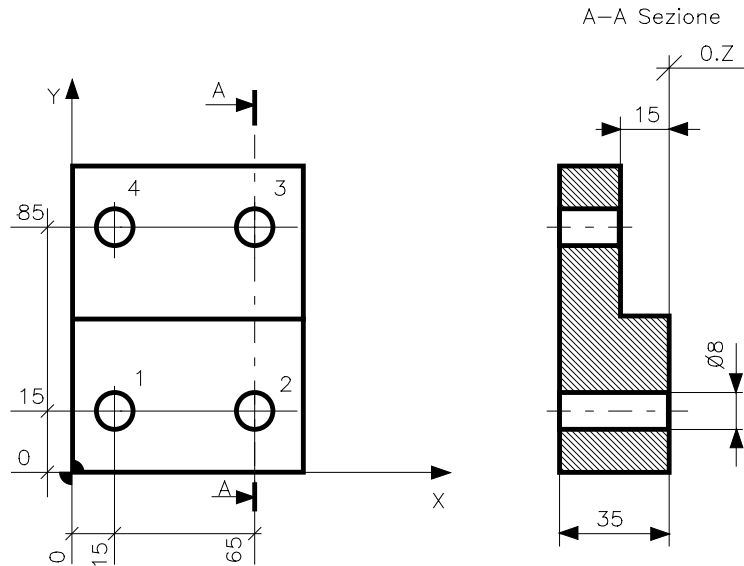
1. X ed Y
2. R1 aggiornato - nuova quota di avvicinamento rapido
3. Z aggiornato - nuova profondità
4. R2 aggiornato - nuova quota di ritorno

La tabella seguente riassume i formati dei blocchi di programma utilizzati per aggiornare le fasi dei cicli fissi:

—	
BLOCCO	AZIONE
—	
X.. Y.. Z..	Esegue un ciclo fisso in XY con una nuova profondità Z.
X.. Y.. R..	Esegue un ciclo fisso in XY con un nuovo piano di avvicinamento in rap. R1.
X.. Y.. R.. R..	Esegue un ciclo fisso in XY con nuove quote di avvicinamento (R1) e di ritorno (R2).
X.. Y.. R.. Z..	Esegue un ciclo fisso in XY con avvicinamento in rapido R1 e profondità Z nuovi.
X.. Y.. R.. R.. Z..	Esegue un ciclo fisso in XY con avvicinamento in rapido R1, quota di ritorno R2 e profondità Z nuovi.
R..	Aggiorna il piano di avvicinamento in rapido R1, non esegue il ciclo fisso nella posizione corrente.
R.. R..	Aggiorna il piano di avvicinamento in rapido R1 e la quota di ritorno R2; non esegue il ciclo fisso nella posizione corrente.
R.. Z..	Aggiorna il piano di avvicinamento in rapido R1, e la profondità Z; non esegue il ciclo fisso nella posizione corrente.
R.. R.. Z..	Aggiorna le dimensioni di piano di avvicinamento in rapido R1, ritorno R2 e profondità Z; non esegue il ciclo fisso nella posizione corrente.
—	

Aggiornamento delle quote R (inferiore e superiore) durante la lavorazione

Esempio 1:



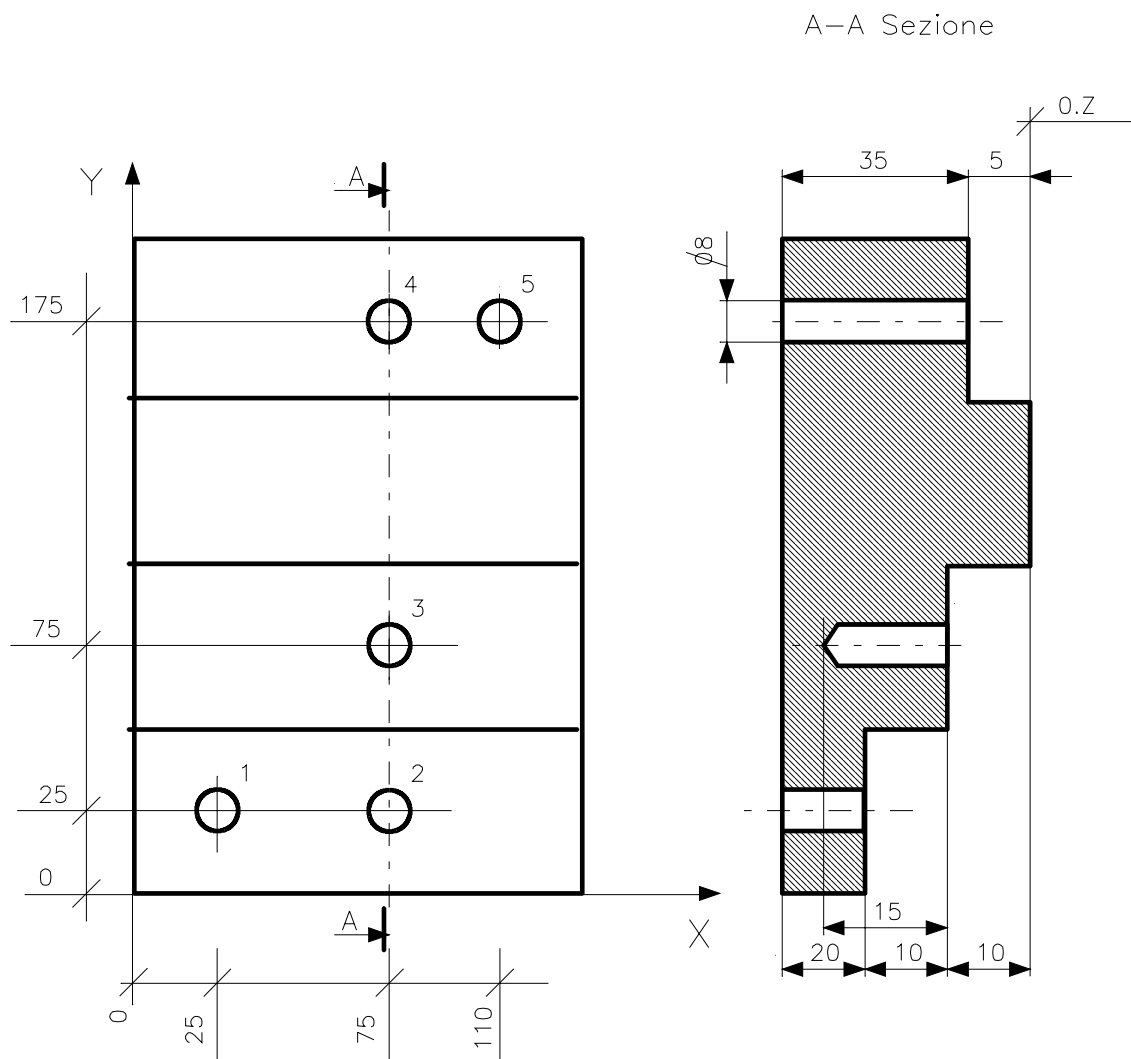
NOTA:

0.Z significa posizione Z=0

Programma:

```

N35 (DIS,"PUNTA A ELICA D=8")
N36 S1000 F100 T4.4 M6
N37 G81 R3 Z-42 M3
1  N38 X15 Y15
2  N39 X65
3  N40 Y85 R-13
4  N41 X15
   N42 G80 Z50 M5
    
```

Esempio 2:**NOTA:**

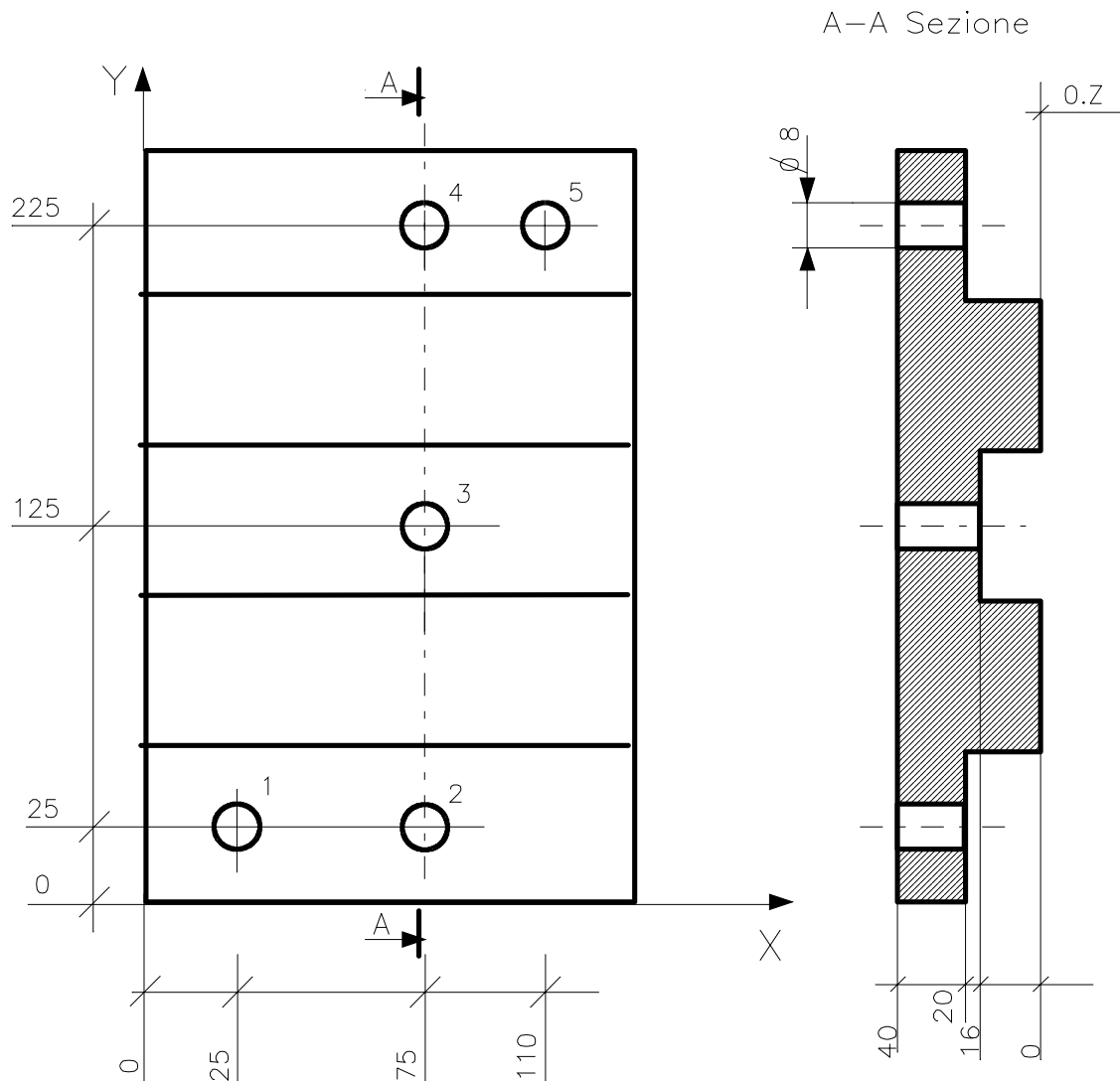
0.Z significa posizione Z=0

Programma:

```

(UGS,X,0,100,Y,0,200)
N42 (DIS,"PUNTA A ELICA D=8")
N43 S1000 F100 T5.5 M6
N44 G81 R-18 Z-46 M13
1   N45 X25 Y25
2   N46 X75 R-18 R-8
3   N47 Y75 R-8 R2 Z-25
4   N48 Y175 R-3 Z-46
5   N49 X110
    N50 G80 Z50 M5
  
```

Esempio 3:



NOTA:

0.Z significa posizione Z=0

Programma:

```
(UGS,X,0,100,Y,0,200)
N42 (DIS,"PUNTA A ELICA D=8")
N43 S1000 F100 T4.4 M6
N44 G81 R-18 Z-42 M3 M8
1 N45 X25 Y25
2 N46 X75 R-18 R4
3 N47 Y125 R-14 R4
4 N48 Y225 R-18
5 N49 X110
6 N50 G80 Z2 M5
```

FINE CAPITOLO

PARAMACRO

Definizione di paramacro

I sottoprogrammi paramacro sono disponibili per i cicli definiti dall'utente. Vengono richiamati con un codice numerico costituito da tre cifre.

Le paramacro modali sono attivate solo nei blocchi di movimento che non contengono funzioni M. Se una paramacro modale è attiva non possono essere programmate altre funzioni o codici .

Sintassi

Gn *par-name-1* [*valore-1*] . . . [*par-name-n*] [*valore-n*] . . . ["*string*"]

dove:

<i>n</i>	E' un numero compreso tra 300 e 998
<i>par-name-1</i> . . .	Possono essere tutte le lettere dell'alfabeto (vedi tabelle di corrispondenza)
<i>par-name-n</i>	
<i>valore-1</i> . . .	Può essere un numero o un parametro E o un'espressione parametrica
<i>valore-n</i>	
<i>string</i>	Stringa di caratteri (max. 99)

Caratteristiche:

Ci sono due gruppi di paramacro:

- da G300 a G699 Paramacro non modali
- da G700 a G998 Paramacro modali

La paramacro G999 effettua il reset delle paramacro modali.

Nelle paramacro vengono utilizzati i parametri H, HF, e HC. Sono disponibili 100 parametri H per realizzare delle paramacro, suddivisi in due gruppi:

- I parametri da H0 ad H51 sono abbinati a delle lettere e non possono essere utilizzati da operazioni all'interno di una paramacro (vedere tavola di corrispondenza).
- I parametri da H52 ad H99 possono essere utilizzati da operazioni matematiche all'interno di una paramacro.

Utilizzando i parametri H nelle paramacro è possibile eseguire le stesse operazioni consentite ai parametri E.

Le paramacro possono avere fino a 4 livelli di annidamento.

Nelle paramacro annidate i parametri H, HF, e HC contenuti in una paramacro inferiore cancelleranno i corrispondenti equivalenti parametri H, HF, e HC delle paramacro di livello superiore.

Quando viene programmata una lettera, il parametro HF corrispondente viene posto a 1 (tipo booleano). Ciò consente di verificare che tutti i parametri della paramacro siano stati programmati.

Esempio:

G300 A1 B2 C3 D4

H0 riceve il valore 1
H1 riceve il valore 2
H2 riceve il valore 3
H3 riceve il valore 4

HF0 è forzato a 1
HF1 è forzato a 1
HF2 è forzato a 1
HF3 è forzato a 1

Tutti gli altri HF vengono posti a zero.

Se G300 effettua la chiamata: G400 A10 C30, H0 avrà valore 10, H2 avrà valore 30, HF0 e HF2 verranno forzati a 1 e tutti gli altri HF verranno posti a zero.

I valori assegnati ad H0 e H2 in G300 vengono perduti.

Se richiesto, i parametri E (ed anche tutti gli altri tipi di parametri disponibili nel CNC Serie 10) possono essere utilizzati nelle paramacro. Tuttavia, per evitare interazioni con i parametri utilizzati altrove, è preferibile utilizzare solo parametri H.

Parametri HC

Se sul blocco di chiamata alla paramacro si specifica anche una stringa di caratteri tra doppi apici, questa viene resa disponibile alla paramacro nell'array di caratteri HC (100 caratteri).

Se la stringa non è stata specificata, tutto l'array HC viene azzerato.

Se la stringa è stata programmata, essa viene posta nelle variabili HC con l'aggiunta del terminatore '0'; per questo motivo la lunghezza massima della stringa è di 99, anche se l'array HC è lungo 100 caratteri.

Esempi:

!PROFILO=.....

G600 "PROFILO" A50 ;chiamata a paramacro G600

..

..

;----- nella paramacro

(DIS,HC0.10)

(CLS,?HC0.10)

Situazione dei parametri HC dopo la chiamata alla paramacro:

HC0="P" HC1="R" HC2="O" HC3="F" HC4="I" HC5="L" HC6="O" HC7=0

H0=50 HF0=1

La seguente tabella mostra la corrispondenza tra lettere e parametri H.

LETTERA	PARAMETRI		LETTERA	PARAMETRI	
	H	HF		H	HF
A	H0	HF0	a	H26	HF26
B	H1	HF1	b	H27	HF27
C	H2	HF2	c	H28	HF28
D	H3	HF3	d	H29	HF29
E	H4	HF4	e	H30	HF30
F	H5	HF5	f	H31	HF31
G	H6	HF6	g	H32	HF32
H	H7	HF7	h	H33	HF33
I	H8	HF8	i	H34	HF34
J	H9	HF9	j	H35	HF35
K	H10	HF10	k	H36	HF36
L	H11	HF11	l	H37	HF37
M	H12	HF12	m	H38	HF38
N	H13	HF13	n	H39	HF39
O	H14	HF14	o	H40	HF40
P	H15	HF15	p	H41	HF41
Q	H16	HF16	q	H42	HF42
R	H17	HF17	r	H43	HF43
S	H18	HF18	s	H44	HF44
T	H19	HF19	t	H45	HF45
U	H20	HF20	u	H46	HF46
V	H21	HF21	v	H47	HF47
W	H22	HF22	w	H48	HF48
X	H23	HF23	x	H49	HF49
Y	H24	HF24	y	H50	HF50
Z	H25	HF25	z	H51	HF51

Esempio 1:

N45 G777 A(E8) R22.5 F(E2) S(E3+5-E1)

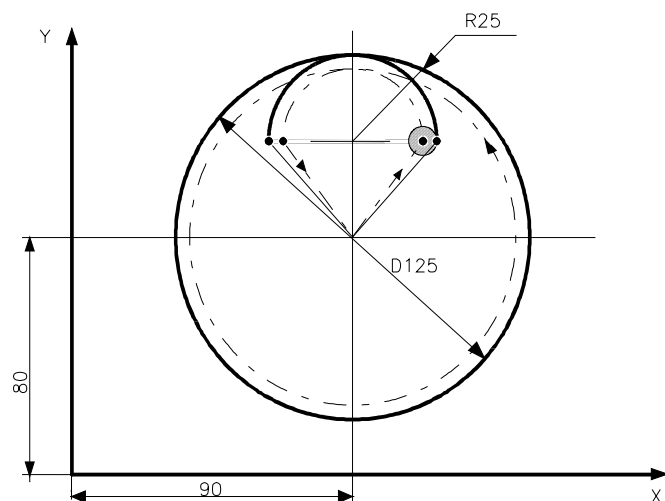
E8 viene passato ad H0

22.5 viene passato a H17

E2 viene passato a H5

Il risultato di (E3+5-E1) viene passato ad H18.

In questo esempio i parametri booleani HF0, HF17 e HF5 sono posti a 1.

Esempio 2:

D = H3 = Diametro maggiore

R = H17= Raggio minore

F = H5 = Velocità di avanzamento

Programma:

Il seguente è un esempio di programmazione di un'operazione di fresatura/alesatura utilizzando delle paramacro.

;PROGRAMMA PRINCIPALE

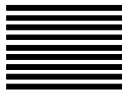
.
.
.

N20 G601 D125 R25 F160
N21 ...

```

; PARAMACRO G601
G0 X90 Y80
G92 XY
(GTO,END,HF3=0)
(GTO,END,HF17=0)
(GTO,END,HF5=0)
H57=H3/2
H58=H57-H17
G1 G41 XH17 YH58 F2000
G3 X0 YH57 I0 JH58 FH5
I0 J0
H59=NEG(H17)
G40 XH59 YH58 I0 JH58
G1 X0 Y0 F2000
(GTO,F)
"END"
G99
(DIS,"PARAMETRI OMESSI")
M..
"F"

```



DAN - Definizione del nome asse

Questo comando associa il nome di un asse caratterizzato al nome usato all'interno di una paramacro.

Sintassi

(DAN,par-ax1 char-ax1[,par-ax6 char-ax6])
(DAN)

dove:

char-ax1 ... char-ax6 Sono i nomi degli assi caratterizzati (max 6).

par-ax1 ... par-ax6 Sono i nomi degli assi usati nella paramacro.

nessun parametro (DAN) senza parametri disabilita modalità (DAN)

Esempio 1:

(DAN,PX,QY,DZ) X,Y,Z vengono sostituiti da P,Q,D quindi X,Y,Z non sono utilizzabili

Esempio 2:

(DAN,PX,QY,DZ) X,Y,Z vengono sostituiti da P,Q,D quindi X,Y,Z non sono utilizzabili

(DAN,WA) A viene sostituito da W inoltre X,Y,Z vengono riabilitati quindi P,Q,D,A non sono utilizzabili

La riprogrammazione di (DAN,...) annulla le precedenti associazioni ed attiva quelle correnti.

IMPORTANTE

Dopo l'utilizzo del triletterale DAN, nel caso in cui gli assi interessati siano quelli del piano d'interpolazione, tale piano deve essere ridefinito usando i nuovi nomi.



PMS – S come paramacro

PMT – T come paramacro

PMM – M come paramacro

Questi comandi abilitano e disabilitano la gestione delle relative funzioni ausiliarie S, T, M come paramacro.

Per maggiori informazioni sulla programmazione di queste paramacro fare riferimento al paragrafo “funzioni ausiliarie S, T, M eseguite come paramacro” di questo capitolo.

Sintassi

PMS=valore

PMT=valore

PMM=valore

dove:

valore

E' un valore che può essere:

- 0** Disabilita la gestione delle funzioni S, T, M come paramacro – tali funzioni vengono eseguite in modo standard
- 1** Abilitano la gestione delle funzioni S, T, M come paramacro – tali funzioni agganciano l'esecuzione della corrispondente paramacro

NOTA:

I valori assegnati alle variabili PMS, PMT e PMM possono essere modificati nei seguenti modi:

- Tramite AMP all'atto della configurazione
- Tramite part program con la sintassi descritta

Il Reset ripristina i valori di caratterizzazione.

FUNZIONI AUSILIARIE S, T, M ESEGUITE COME PARAMACRO

E' possibile agganciare all'esecuzione delle funzioni ausiliare S T M dei sottoprogrammi di tipo paramacro, ossia, a fronte di un blocco contenente almeno una funzione S, T e/o M, preventivamente configurata come paramacro, viene eseguito il corrispondente sottoprogramma, con le regole già attualmente definite per le paramacro.

Sintassi:

uguale a quella per le funzioni ausiliarie eseguite in modo standard.

CODICE AUSILIARIO	PARAMACRO
----------------------	-----------

S	A fronte di una funzione "Sx", ad es. sul processo "n", verrà eseguita la paramacro "P0nS" con parametro H18 = x
T	A fronte di una funzione "Tx", ad es. sul processo "n", verrà eseguita la paramacro "P0nT" con parametro H19 = x
M	A fronte di una funzione "Mx", ad es. sul processo "n", verrà eseguita la paramacro "P0nM00x" con parametro H12 = x

Caratteristiche:

- Le funzioni che generano la chiamata a paramacro possono non essere programmate a inizio blocco.
- Se si ha una programmazione contemporanea di M, S e/o T, il nome della procedura da chiamare è determinato dalla prima funzione configurata come paramacro, tutte le eventuali altre funzioni programmate nel blocco vengono passate nei rispettivi parametri H, HF e HC, come da manuale di programmazione (capitolo 9). Es.: se è configurata sia la M che la T come paramacro, la programmazione di TxMy genererà la chiamata alla macro P0nT mentre la programmazione di MyTx chiamerà la macro "P0nM00x" con i parametri H19=x e H12=y per entrambi i casi.
- Per l'esecuzione della paramacro in modalità MDI si faccia riferimento al manuale d'uso paragrafo "Esecuzione di blocchi inseriti da tastiera"
- Al termine della paramacro viene ripristinato il modo di esecuzione (MDI, AUTO, STEP) in cui si trovava il sistema alla sua chiamata.
- Il tipo di paramacro in esecuzione può essere letto nella variabile **SW21**, ed è pari a:
 - 1000 per paramacro chiamata dalla funzione S
 - 2000 per paramacro chiamata dalla funzione T
 - 3xxx per paramacro chiamata dalla funzione Mxxx

- Il *pathname* da utilizzare per il richiamo delle paramacro può essere dichiarato anche tramite il trilettale PTH descritto più avanti in questo capitolo. Se il pathname è omissso la ricerca del direttorio di appartenenza della subroutine avviene secondo la seguente modalità:

1. caso: non è stato specificato nessun pathname tramite PTH.

Il sistema cerca la paramacro nel direttorio in cui si trova il programma chiamante e, se qui non esiste, la cerca nei direttori DOS a cui sono stati associati dei nomi logici in fase di caratterizzazione della macchina.

2. caso: è stato specificato un pathname tramite l'istruzione PTH.

Il sistema cerca la paramacro nel direttorio specificato con PTH e, se qui non esiste, la cerca nei direttori DOS a cui sono stati associati i nomi logici.

Configurazione: esistono tre variabili di processo (PMM, PMS, PMT definite precedentemente) che se a 1/0 abilitano/disabilitano a livello generale l'esecuzione delle corrispondenti funzioni come paramacro.

Inoltre, per la specifica funzione ausiliaria M, esiste ancora un opportuno bit, definito in AMP, che ne identifica l'esecuzione come paramacro.

Quindi, se PMM è a 1 e la funzione "Mx":

- è configurata nel processo "n" ma il suo bit è a 0 non viene eseguita la paramacro (esecuzione di Mx standard)
- è configurata nel processo "n" e il suo bit è a 1 viene eseguita la paramacro "P0nM00x"
- non è configurata nel processo "n" viene visualizzato errore di "M non definita"

Esempio:

Esecuzione del seguente blocco sul processo 1 nel quale è configurata la funzione ausiliaria M2 con il bit di paramacro abilitato

N20 X10 M2 T1.0 S2000

PMM	PMT	PMS	Nome paramacro	Parametri
1	1 o 0	1 o 0	P01M002	H23 = 10; H12 = 2; H19 = 1.0 H18 = 2000
0	1	1 o 0	P01T	H23 = 10; H12 = 2; H19 = 1.0 H18 = 2000
0	0	1	P01S	H23 = 10; H12 = 2; H19 = 1.0 H18 = 2000

Un esempio di come sviluppare le relative paramacro è il seguente:

;PARAMACRO P01M002

```
E0 = H12
PMM = 0
ME0
"TEST_T"
(GTO,TEST_S,HF19=0)
E1 = H19
PMT = 0
TE1
"TEST_S"
(GTO,END,HF18=0)
E2 = H18
PMS = 0
SE2
"END"
```

;PARAMACRO P01T

```
E1 = H19
PMT = 0
TE1
"TEST_M"
(GTO,TEST_S,HF12=0)
E0 = H12
PMM = 0
ME0
"TEST_S"
(GTO,END,HF18=0)
E2 = H18
PMS = 0
SE2
"END"
```

;PARAMACRO P01S

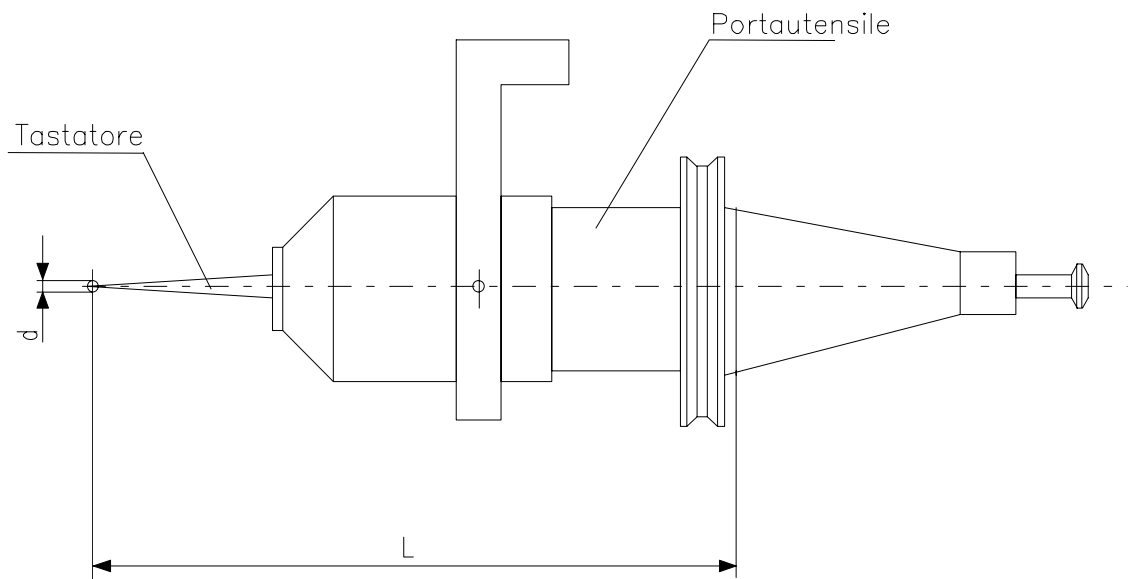
```
E2 = H18
PMS = 0
SE2
"TEST_M"
(GTO,TEST_T,HF12=0)
E0 = H12
PMM = 0
ME0
"TEST_T"
(GTO,END,HF19=0)
E1 = H19
PMT = 0
TE1
"END"
```

FINE CAPITOLO

CICLI DI TASTATURA

GESTIONE DEL TASTATORE ELETTRONICO

Un tastatore elettronico è un dispositivo di misura che può essere installato sul mandrino e controllato come un utensile associato a dei correttori di lunghezza e di diametro. Inoltre, il tastatore può essere installato in una posizione fissa ed utilizzato come comparatore elettronico per realizzare la riqualifica della lunghezza dell'utensile. La figura seguente mostra un tastatore elettronico con le proprie dimensioni tipiche.



Tastatore elettronico

I codici G che realizzano i cicli di misura sono i seguenti:

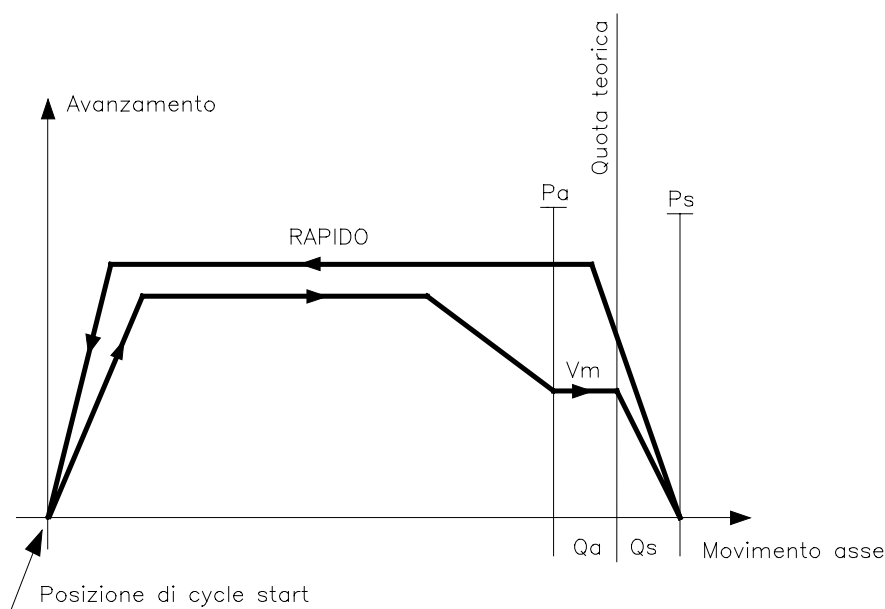
CODICE G	FUNZIONE
G72	Misura le coordinate di un punto nello spazio
G73	Misura le coordinate del centro ed il raggio di un cerchio
G74	Misura lo scostamento da punti teorici (per tastatori installati in una posizione fissa).

Il controllo memorizza i valori misurati nei parametri E definiti nel ciclo di tastatura.

Quando il controllo esegue un ciclo di tastatura per misurare un punto, vengono eseguiti i seguenti movimenti:

1. Spostamento in rapido sul punto di avvicinamento (P_a)
2. Avanzamento a velocità di misura (V_m) fino al punto teorico di tastatura, arresto e memorizzazione delle quote. Se non avviene la tastatura, gli assi si spostano fino al punto di sicurezza (P_s).
3. Ritorno in rapido alla posizione di inizio ciclo (centro del foro in G73). L'esecuzione di tale fase è subordinata ad un parametro di caratterizzazione della macchina (AMP) od al corrispondente parametro "modalità" del triletterale DPP.

Il seguente diagramma mostra gli spostamenti e la velocità di avanzamento tipici di un ciclo di tastatura.



Durante un ciclo di tastatura possono verificarsi i seguenti errori:

- Se la tastatura non avviene prima di raggiungere il punto di sicurezza, il tastatore ritorna al punto di partenza del ciclo, il controllo visualizza un messaggio e il sistema entra in condizione di errore. L'esecuzione del ritorno in rapido è subordinata ad un parametro di caratterizzazione della macchina (AMP) od al corrispondente parametro "modalità" del triletterale DPP.
- Se il tastatore non effettua il reset in modo appropriato mentre torna indietro al punto di partenza dopo un ciclo di tastatura terminato con successo, viene visualizzato un messaggio e il sistema entra in condizione di errore.

- Se la tastatura viene effettuata durante la fase di avvicinamento rapido, il tastatore ritorna al punto di partenza del ciclo, il controllo visualizza un messaggio e il sistema entra in condizione di errore.

NOTA:

Se il ciclo di tastatura viene eseguito con G27 o G28 attivi, sia il moto di avvicinamento che la tastatura vengono eseguiti in continuo, mentre il movimento di fermata che si ha dopo aver trovato l'input del tastatore avviene in modalità punto - punto.



I valori ottenuti con i cicli di tastatura sono sempre espressi nell'unità di misura corrente del processo.

PREDISPOSIZIONE DI UN CICLO DI TASTATURA

La prima volta che si utilizza un tastatore oppure ogni volta che cambiano le condizioni del ciclo di tastatura, bisogna:

1. definire i parametri di tastatura
2. misurare dinamicamente il diametro della sfera del tastatore
3. riqualificare il tastatore rispetto all'asse mandrino
4. misurare dinamicamente la lunghezza del tastatore



DPP (DPT) - Definizione dei parametri di tastatura

Per definire i parametri di tastatura introdotti da tastiera o da programma si usa il comando DPP. Questi parametri possono anche essere definiti tramite le softkey MACHINE SET-UP e PROBE SET-UP. Nel caso in cui essi non vengono definiti, il sistema assume come valori quelli definiti in AMP.

Sintassi

(DPP,avvicinamento,sicurezza,velocità [, modalità])

dove:

<i>avvicinamento</i>	E' la quota di avvicinamento espressa in mm o pollici.
<i>sicurezza</i>	E' la quota di sicurezza espressa in mm o pollici.
<i>velocità</i>	E' la velocità di misurazione espressa in mm/min o pollici/min.
<i>modalità</i>	E' un valore che indica se eseguire la fase di ritorno in rapido a fine tastatura. Può assumere i valori: 0 = esegue la fase di ritorno 1 = non esegue la fase di ritorno Se il parametro manca viene assunto per default 0.

Esempio:

Istruzioni digitate da programma

(DPP,10,12,1000)



Le quote di "avvicinamento", "sicurezza" e "velocità" vengono visualizzate nell'unità di misura del processo configurata in AMP.

Misurazione dinamica del diametro della sfera

Per misurare dinamicamente il diametro apparente della sfera del tastatore, utilizzare un anello campione, un cilindro o dispositivi simili, il centro dei quali viene assunto come origine assoluta 9 per gli assi X ed Y.

Riqualifica del tastatore

Prima di utilizzare il tastatore, il centro della sua sfera è normalmente scostato rispetto all'asse del mandrino. Per centrare la sfera di prova sull'asse del mandrino, è necessario riqualificare il tastatore.

Usare il foro dell'anello campione, un cilindro o dispositivi simili per effettuare tale operazione. Il centro del foro è assunto essere l'origine 9 per gli assi X ed Y.

Misurazione dinamica della lunghezza del tastatore

Per misurare dinamicamente la lunghezza del tastatore, utilizzare l'anello di riqualifica, un cilindro o dispositivi simili la cui superficie di riferimento viene posta l'origine 9 per l'asse Z.

Predisposizione del tastatore

Per predisporre il tastatore occorre:

1. Utilizzare un anello di riqualifica campione.
2. Installare l'anello di riqualifica sull'asse rotativo B con la superficie frontale nell'origine assoluta 9 dell'asse rotativo B.
3. Posizionare il centro del foro dell'anello sull'origine assoluta 9 per gli assi X ed Y .
4. Posizionare il piano superiore dell'anello sull'origine assoluta 9 per l'asse Z.

I valori iniziali memorizzati nel correttore del tastatore sono:

Length1	=	lunghezza nominale del tastatore rispetto all'asse della sfera.
Diameter	=	0 (diametro sfera tastatore)

Esempio di riqualifica del tastatore:

N1	(DIS, "DPP, UPA, UPO, ")	
N2	T30.30 M6	- tastatore in mandrino
N3	(UAO, 9)	
N4	(DPP, 10, 12, 600)	- definisce i parametri di tastatura
N5	UPA=0	
N6	UPO=0	
N7	E30=...	- assegna il diametro al foro dell'anello campione
N8	E31=E30/2	
N9	E32=...	- assegna la distanza da Z=0 alla superficie di tastatura sull'asse Z (di solito = 0)
N10	E33=E31+10	
N11	GB0	- solo se l'anello è installato su tavola girevole
N12	XY	
N13	Z-4	
N14	G73 rE31 E40	- misura le coordinate del foro (centro e raggio)
N15	Z100	
N16	(DIS, "UPA=", E40, "UPO=", E41)	
N17	M0	
N18	UPA=E40	- riqualifica l'ascissa del tastatore
N19	UPO=E41	- riqualifica l'ordinata del tastatore
N20	E34=(E30-E42*2)	- diametro della sfera apparente
N21	(DIS, "DIAMETER=", E34)	
N22	M0	
N23	(RQP, 30, 30, dE34)	- riqualifica il valore del correttore diametro
N24	T30.30 M6	- abilita il nuovo correttore
N25	GXYE33	
N26	G72 ZE32 E43	- misura la quota Z sulla superficie dell'anello
N27	E35=E43-E32	- scostamento tra valore dinamico e valore teorico
N28	Z100	
N29	(DIS "VARIANCE.Z=", E35)	
N30	M0	
N31	(RQP, 30, 30, LE35)	- riqualifica del correttore lunghezza Z
N32	M30	

CICLI DI TASTATURA

I seguenti codici G definiscono cicli di tastatura:

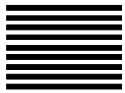
G72 Misura di un punto con compensazione del diametro della sfera del tastatore

G73 Misura dei parametri di un foro

G74 Misura degli scostamenti tra punti misurati e punti teorici senza compensazione del diametro della sfera del tastatore

Nei cicli G72-G73-G74 i parametri E vengono abbinati agli assi nell'ordine in cui gli assi sono caratterizzati (AMP) e visualizzati, non nell'ordine in cui vengono programmati.

Nel caso di tastatura su assi virtuali di piani ruotati i parametri E vengono abbinati agli assi virtuali nell'ordine in cui essi sono stati definiti nel triletterale UPR.



G72 - Misura di un punto con compensazione

G72 consente di misurare le coordinate di un punto nello spazio muovendo il tastatore con movimento lineare.

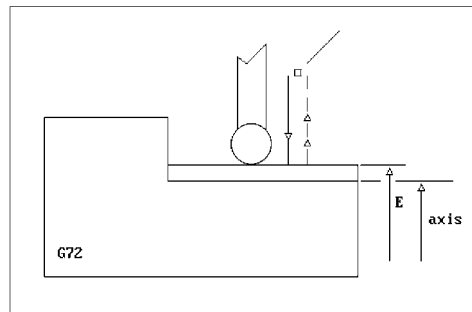
Sintassi

G72 *asse1* [*asse2*] [*asse3*] *E-par*

dove:

asse1...asse3 Sono gli assi interessati al ciclo di tastatura. Le quote programmate nel blocco G72 sono quelle teoriche del punto. E' possibile programmare fino a tre assi e tre quote nel blocco G72.

E-par Definisce il parametro E(x) per la memorizzazione della quota misurata del primo asse, in ordine di configurazione, programmato in G72; le quote degli altri assi vengono memorizzate sequenzialmente nei parametri E(x+1) e E(x+2) sempre in ordine di configurazione ed E (x+3) nel caso di tastatura su assi virtuali.



Caratteristiche:

Le coordinate misurate dal tastatore sono memorizzate in parametri E partendo dal parametro E specificato nel ciclo G72 ed espresse nell'unità di misura corrente. Il controllo effettua la misura con correzione utensile attiva sul tastatore. Per raggiungere un alto grado di precisione, la superficie deve essere perpendicolare allo spostamento di misura.

La tastatura può anche essere eseguita sugli assi virtuali attivati con la funzione UPR o sull'asse che individua la direzione utensile attivato tramite (TCP, 5).

Nel caso di tastatura eseguita sugli assi virtuali definiti con UPR, il sistema ritorna in E (x), E (x+1), E (x+2) i valori misurati relativi agli assi virtuali.

Nel caso di tastatura eseguita sull'asse direzione utensile, il sistema ritorna in E (x) il valore della quota misurata relativa a tale asse ed in E (x+1), E (x+2), E (x+3) i valori misurati, riferiti al pezzo, degli assi lineari cartesiani definiti in tabella TCP o dei corrispondenti assi virtuali nel caso di virtualizzazione UPR attiva.

Esempio:

Gli assi sono configurati nell'ordine X Y Z.

G72 X100 Y50 E32

G72 Y50 X100 E32

In entrambi i casi i valori calcolati dal controllo per X ed Y vengono memorizzati in E32 ed E33 rispettivamente.

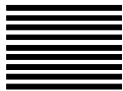
Esempio:

Tastatura su piani ruotati con assi virtuali.

(UPR,0,XYZ,UWV,30,25,0)

G72 U25 V-10 W-25 E30

Nei parametri E30 E31 E32 vengono memorizzate le quote della terna virtuale secondo l'ordine definito nel comando UPR ovvero U in E30, W in E31, V in E32.



G73 - Misura dei parametri di un foro

G73 consente di utilizzare un tastatore per misurare le dimensioni di un foro nel piano di interpolazione attivo e di memorizzarle in parametri E.

Sintassi

G73 r.. E-par1

dove:

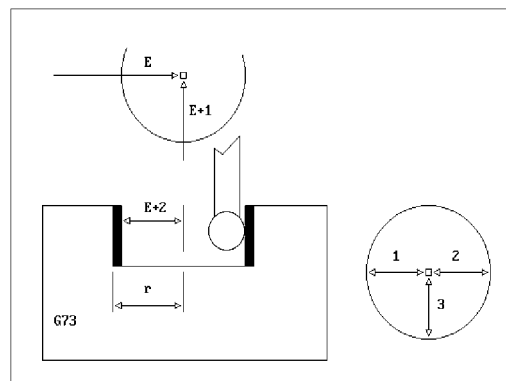
r Definisce il raggio teorico del foro.

E-par1 Sono i parametri E per la memorizzazione dei seguenti valori:

primo parametro E: ascissa del centro (normalmente X)

secondo parametro E: ordinata del centro (normalmente Y)

terzo parametro E: raggio del foro



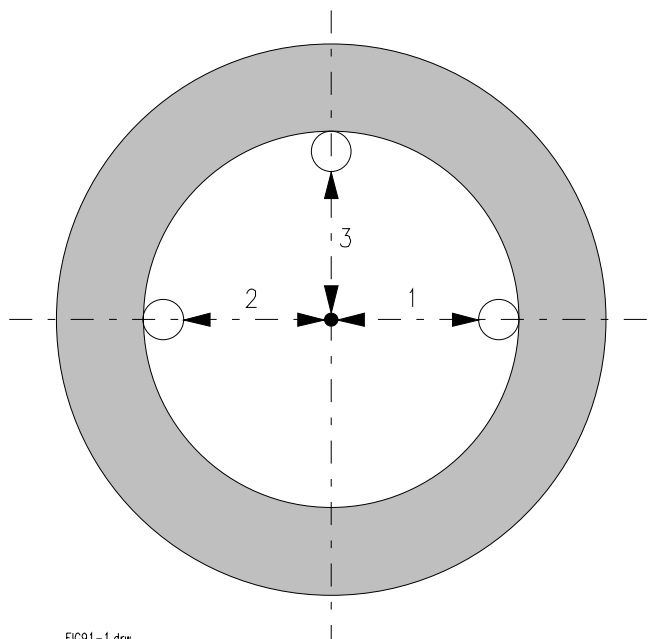
Caratteristiche:

Programmando solo un parametro E, i tre valori rilevati durante l'operazione di tastatura (coordinate del centro del cerchio e raggio) vengono memorizzati nei parametri E in ordine sequenziale partendo dal parametro E specificato, espressi nell'unità di misura corrente.

La tastatura può anche essere eseguita sugli assi virtuali attivati con la funzione UPR.

In questo caso il sistema ritorna in E (x) ed in E (x+1) le quote del centro riferite al piano virtuale ed in E (x+2) il raggio del cerchio.

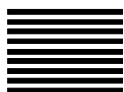
Prima di attivare il ciclo G73, gli assi della macchina utensile devono essere posizionati sul centro del foro. Il controllo effettua le misure con correzione utensile attiva sul tastatore.



Movimenti di tastatura

Esempio:
G73 r100 E55

Le coordinate del centro del cerchio (ascissa e ordinata) e il raggio effettivo vengono memorizzati rispettivamente in E55, E56 ed E57.



G74 - Ciclo di riqualifica dell'utensile

G74 è un ciclo di riqualifica dell'utensile. Con l'utensile inserito nel mandrino, G74 usa un tastatore fisso (del tipo comparatore elettronico) per misurare gli scostamenti tra i punti misurati e i punti teorici.

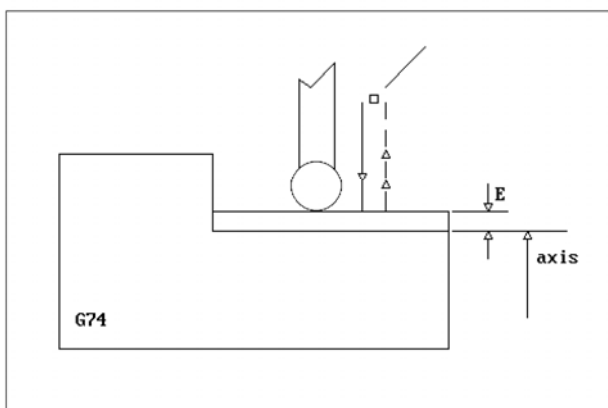
Sintassi

G74 *asse1* [*asse2*] [*asse3*] *E-par*

dove:

asse1... asse3 Sono fino a tre assi simultanei.

E-par Definisce il parametro $E(x)$ per la memorizzazione dello scostamento tra il punto misurato ed il punto programmato del primo asse programmato in G74; le quote di scostamento degli altri assi, se programmati, vengono memorizzate sequenzialmente nei parametri $E(x+1)$ ed $E(x+2)$.



Caratteristiche:

G74 può essere utilizzato per riquilibrare un utensile o per controllarne l'usura. Il calcolo delle quote misurate non tiene conto del correttore dell'utensile, in quanto verifica la dimensione effettiva dell'utensile.

Le fasi del ciclo G74 sono simili a quelle del ciclo G72. La differenza è costituita da come il controllo esegue i calcoli sulle quote misurate. Il controllo non considera il diametro della sfera del tastatore e memorizza gli scostamenti dalle dimensioni teoriche nei parametri specificati nel blocco G74, nell'unità di misura corrente.

La tastatura può anche essere eseguita sugli assi virtuali attivati con la funzione UPR o sull'asse che individua la direzione utensile attivato tramite (TCP, 5)

Nel caso di tastatura eseguita sugli assi virtuali definiti con UPR, il sistema ritorna in E (x), E (x+1), E (x+2) i valori di scostamento tra i punti misurati ed i punti programmati riferiti agli assi virtuali.

Nel caso di tastatura eseguita sull'asse direzione utensile, il sistema ritorna in E (x) il valore di scostamento tra il punto misurato ed il punto programmato di tale asse ed in E (x+1), E (x+2),

E (x+3) i valori misurati riferiti al pezzo, degli assi lineari cartesiani definiti in tabella TCP o dei corrispondenti assi virtuali nel caso di virtualizzazione UPR attiva.

Esempio:

G74 X60 E41

dove:

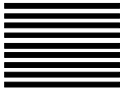
E41 è dato dalla formula:

$$E41 = P_m - P_t$$

in cui:

P_m è il punto misurato

P_t è il punto teorico



UPA (RTA) - Aggiornamento dell'ascissa del tastatore

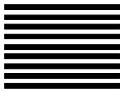
Definisce il valore di riqualifica del tastatore per l'asse delle ascisse (normalmente X).

Sintassi

UPA=*valore*

dove:

valore E' il valore di riqualifica per l'asse delle ascisse espresso in millimetri o pollici.
La variabile UPA è di tipo non volatile, il suo valore cioè permane anche durante lo spegnimento del controllo.



UPO (RTO) - Aggiornamento dell'ordinata del tastatore

Definisce il valore di riqualifica del tastatore per l'asse delle ordinate (normalmente Y).

Sintassi

UPO=*valore*

dove:

valore E' il valore di riqualifica per l'asse delle ordinate espresso in millimetri o pollici.
La variabile UPO è di tipo non volatile, il suo valore cioè permane anche durante lo spegnimento del controllo.



Gestione degli errori di tastatura

I cicli G72-G73-G74 permettono di gestire gli errori di tastatura sia automaticamente sia da part program. La scelta fra questi due modi può essere effettuata settando il parametro ERR.

Per maggiori informazioni sulla gestione da part program degli errori, consultare l'appendice C.

OPERAZIONI CON UN TASTATORE MOBILE

Con i cicli di tastatura G72 e G73 si possono eseguire le seguenti operazioni:

- Riqualifica delle origini tramite:
 - tastatura delle superfici di riferimento
 - centratura su un foro.
- Controllo delle dimensioni di:
 - diametri
 - piani e profondità fori.

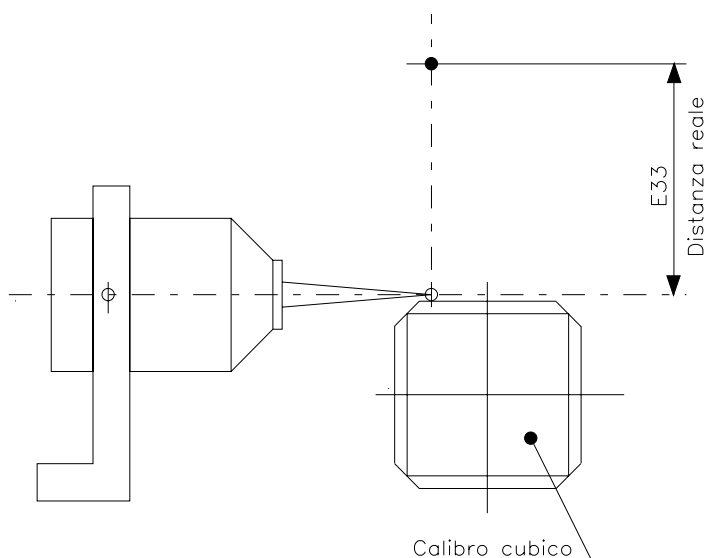
Riqualifica delle origini tramite tastatura di superfici di riferimento

Esistono due condizioni che possono portare a riqualificare le origini tastando una superficie di riferimento. Entrambe sono modifiche dell'origine dovute ai seguenti eventi.

- Variazioni della temperatura (deriva termica)
- Uso di un nuovo pallet per la lavorazione.

Deriva termica

In questo caso si usa un cubo di riqualifica posizionato sulla macchina ad una quota precisa conosciuta.



Programma principale:

```
.
.
.
N99 E33=-300
```

```
/N100 (CLS,TEST3)
```

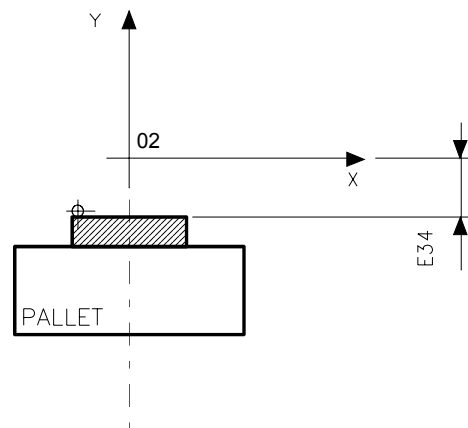
Sottoprogramma TEST3:

```
N500 (DIS,"RQO-DT")
N501 G72 Y(E33) E32          ;Distanza misurata memorizzata in E32
N502 E32=E32-E33
N503 (RQO,1,Y(E32))          ;Riqualifica l'origine 1 per l'asse Y
N504 (RQO,2,Y(E32))          ;Riqualifica l'origine 2 per l'asse Y
N505 (RQO,3,Y(E32))          ;Riqualifica l'origine 3 per l'asse Y
```

Nuovo pallett per la lavorazione

Programma principale:

```
M . . .          ;cambio di pallet
N199 T30.30M6    ;tastatore sul mandrino
N200 (UAO,2)
N201 GXY
N202 E10=2
N203 E34=-250
/N204 (CLS,TEST4)
```



Sottoprogramma TEST4:

```
N500 G72 Y(E34) E30
N501 E31=E30-E34
N502 (RQO,E10,Y(E31)) ;riqualifica l'origine 2 per l'asse Y
```

Riqualfica delle origini centrando su un foro

Prima di eseguire questo ciclo di riqualfica è necessario programmare il posizionamento degli assi X ed Y sul foro ed il posizionamento del tastatore all'interno dello stesso.

Esempio:

N200 (DIS,"RIQUALIFICA ORIGINI ASSI X E Y")

N201 T11.11 M6

N202 GX180 Y60

N203 Z-130

N204 G73 r50 E35 ;Ciclo di misura su foro di diametro 100. Le coordinate X ed ;Y vengono memorizzate nei parametri E35 ed E 36

N205 E35=E35-180

N206 E36=E36-60

N207 (RQO,1,X(E35),Y(E36)) ;Riqualfica l'origine 1 per X (E35) ed Y (E36)

Verifica dei diametri dei fori

E' possibile utilizzare un tastatore per verificare i diametri dei fori. Con una programmazione appropriata è possibile confrontare i valori rilevati con i valori ammessi e decidere se continuare la lavorazione o saltare ad un blocco contenente un'etichetta.

Nel programma seguente il controllo rileva lo scostamento tra il diametro effettivo del foro e quello teorico e confronta questo valore con le tolleranze ammesse. Il confronto ha tre esiti possibili, ad ognuno dei quali corrisponde un'azione di risposta:

- se il diametro del foro è entro le tolleranze consentite, il programma continua.
- se il diametro del foro è superiore alle tolleranze consentite, l'utensile T13 viene dichiarato fuori uso (etichetta A3). Il programma si arresta (M00) ed il pezzo viene scartato.
- se il diametro del foro è inferiore alle tolleranze consentite, l'utensile T13 viene dichiarato fuori uso (etichetta A4). Il programma effettua un salto all'etichetta A1, dove il ciclo di alesatura del foro è ripetuto usando l'utensile alternativo.

Esempio:

Il seguente è un esempio di verifica di un diametro.

Diametro del foro e tolleranza: $D=100 +0.02/-0.015$

Capitolo 10

Cicli di Tastatura

Programma:

"A1" N111

N112 GZ-150

N113 (DIS,"ALESATURA FORO D=100")

N114 F..S..T13.13 M6

N115 GX-120 Y80 M13

.

.

.

N129 (DIS,"VERIFICA TOLLERANZA FORO D=100")

N130 T14.14 M6 ;tastatore sul mandrino

N131 GX-120 Y80 ;posizionamento con XY sul centro del foro

N132 Z-85 ;posizionamento Z

N133 G73 r50 E30 ;raggio del foro memorizzato in E32

N134 E32=E32*2

N135 (DIS,E32)

N136 (GTO,A3,E32>100.02)

N137 (GTO,A4,E32<99.985)

N138 GZ150

N139 (DIS,"PEZZO IN TOLLERANZA")

.

.

.

N2100 M30

"A3" N2101 (TOU,13)

(DIS, "Foro troppo grande")

N2102 M00

"A4" N2103 (TOU,13)

(DIS, "Foro troppo piccolo")

/N2104 (GTO,A1)

Verifica delle quote dei piani e delle profondità dei fori

E' possibile utilizzare un tastatore per verificare le quote dei piani e le profondità dei fori. Con una programmazione appropriata si possono confrontare i valori rilevati con i valori ammessi e decidere se continuare la lavorazione o effettuare un salto ad un blocco contenente una etichetta.

Nel seguente esempio di programmazione, il controllo rileva lo scostamento tra le quote effettive e quelle teoriche e confronta questo valore con le tolleranze consentite. Il confronto ha tre risultati possibili:

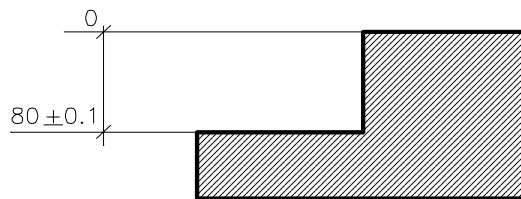
- La lavorazione continua
- La lavorazione si arresta e il pezzo viene scartato a causa della quota troppo lunga
- L'utensile viene riquilificato e il programma viene ripetuto dall'etichetta C1 perché la quota è troppo corta.

Esempio:

Il seguente è un esempio di verifica delle quote di un piano
Dimensioni richieste, rispetto allo 0 sull'asse Z: -80 ± 0.1

Programma:

```
.
.
.
"C1" N252
N253 (DIS,"...")
N254 F. . S. . T23.23 M6
```



```
.
.
.
N267 (DIS,"VERIFICA PIANO QUOTA - 80")
N268 T30.30 M6 ;Tastatore sul mandrino
N269 GX150 Y35 ;Posizionamento XY sul punto da verificare
N270 G72 Z-80 E30 ;Misura della distanza e memorizzazione in E30
N271 (GTO,C2,E30 < -80.1) ;Salta a C2 se E30 < quota minima
N272 (GTO,C3,E30 > -79.9) ;Salta a C3 se E30 > quota massima
N273 (DIS,"PEZZO IN TOLLERANZA")
.
.
.
N2100 M30
"C2" N2101 (DIS,E30) ;Quota troppo lunga
N2102 M00
"C3" N2103 (DIS,E30) ;Quota troppo corta
N2104 E31=-80-E30 ;Differenza tra quota teorica e misurata
N2105 (RQT,23,23,LE31) ;Riquilifica correttore 23 in lunghezza (Z)
N2106 (GTO,C1)
```

OPERAZIONI CHE UTILIZZANO UN TASTATORE FISSO

Utilizzando un ciclo di tastatura G74 con un tastatore in posizione fissa (come un comparatore elettronico) e un utensile montato in mandrino è possibile:

- riqualificare l'utensile (modificando automaticamente i correttori dell'utensile)
- verificare l'integrità dell'utensile.

Esempio 1:

Il seguente è un esempio di riqualifica dei correttori di lunghezza dell'utensile.

N170 G X100 Y100 ;Posizionamento sul punto di misurazione (posizione del tastatore)
N180 G74 Z-50 E30 ;Misura scostamento e valore memorizzato in E30
N190 (RQT,10,1,LE30) ;Riqualifica del correttore 1 dell'utensile 10 in lunghezza (Z) del valore E30

Esempio 2:

Il seguente è un esempio di riqualifica del correttore lunghezza e diametro dell'utensile.

N200 G X100 Y100 ;Posizionamento sul punto di misura (posizione del tastatore)
N210 G74 Z-50 E30 ;Misura scostamento (asse Z) e valore memorizzato in E30
N220 G X150
N230 Z-60
N240 G74 X130 E31 ;Misura scostamento (asse X) e valore memorizzato in E31
N250 E31=E31*2
N260 (RQT,10,1,LE30,dE31)
;Riqualifica il correttore 1 dell'utensile 10 in lunghezza (Z) del valore in E30 e in diametro (d) del valore in E31.

Esempio 3:

Il seguente è un esempio di verifica integrità utensile utilizzando la gestione dell'errore in tastatura (vedi appendice C).

```

N480      (DIS,"UTENSILE D=10")
N490      T10.10M6
.....
.....
N600      (DPP,10,5,500)
N610      (UAO,9)
N620      GXY
N630      ERR=1 ;Errore gestito da programma
N640      G74 Z0 E35 ;misura lunghezza utensile
N650      (GTO,A2,STE=1) ;salta ad A2 se l'utensile è rotto (la tastatura non è avvenuta
           ;nella quota di sicurezza di 5 mm)
N660      ERR=0 ;errore gestito da sistema
.....
.....
N1500     M30
"A2"      N2001 (TOU,10) ;dichiara fuori uso l'utensile 10
N2002     (DIS,"TOOL K.O.")
N2003     M0

```

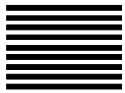
FINE CAPITOLO

GESTIONE DEL VIDEO

GESTIONE DEL VIDEO GRAFICO

La seguente serie di comandi consente di controllare il video grafico e visualizzare delle variabili da program:

COMANDO	FUNZIONE
UGS	Abilita l'uso della scala grafica
CGS	Cancella la scala grafica
DGS	Disabilita la scala grafica
DIS	Visualizza una variabile



UGS (UCG) - Uso della scala grafica

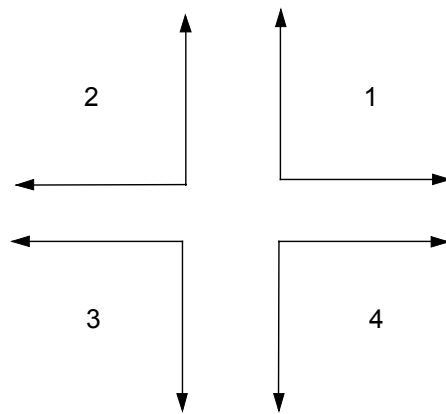
Il comando UGS inizializza il video grafico e stabilisce i limiti e l'orientamento dello stesso.

Sintassi

(UGS [,ax-orient],abs-axis,val1,val2,ord-axis,val3,val4 [,third-axis])

dove:

<i>ax-orient</i>	E' un numero (da 1 a 4) che seleziona il tipo di orientamento degli assi (vedere figura successiva). Il valore di default è 1.
<i>abs-axis</i>	E' il nome dell'asse delle ascisse sullo schermo.
<i>val1</i>	E' il limite inferiore sull'asse delle ascisse.
<i>val2</i>	E' il limite superiore sull'asse delle ascisse.
<i>ord-axis</i>	E' il nome dell'asse delle ordinate sullo schermo.
<i>val3</i>	E' il limite inferiore sull'asse delle ordinate.
<i>val4</i>	E' il limite superiore sull'asse delle ordinate.
<i>third-axis</i>	E' il nome del terzo asse (in generale l'asse mandrino).

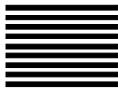


Orientamento degli assi

Esempio:

(UGS,1,X,100,150,Y,50,250,Z)

Lo schermo grafico visualizza i movimenti compresi tra X100 ed X150 per l'asse delle ascisse e tra Y50 e Y250 per l'asse delle ordinate e riferiti all'origine corrente.



UGS (UCG) - Uso della scala grafica in 3D

Sintassi

(UGS ,5,axis1,val1,val2,axis2,val3,val4,axis3,val5,val6 [α , β])

dove:

5	Modo 3D
axis1,axis2,axis3	Sono i nomi degli assi che compongono la terna da visualizzare.
val1,val2	Limiti inferiore e superiore del primo asse.
val3,val4	Limiti inferiore e superiore del secondo asse.
val5,val6	Limiti inferiore e superiore del terzo asse.
α	Parametro angolare α . E' l'angolo di rotazione da applicare al piano orizzontale durante la visualizzazione 3D. Per piano orizzontale, in genere, si intende il piano formato da XY.
β	Parametro angolare β . E' l'angolo di rotazione da applicare al piano verticale durante la visualizzazione 3D. Per piano verticale, in genere, si intende il piano formato da XZ oppure da YZ.

NOTA:

I parametri α e β sono opzionali; qualora siano omessi, il sistema assumerà per *default* che:

$$\alpha = 30^\circ$$

$$\beta = 30^\circ$$

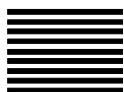


CGS (CLG) - Cancellazione dello schermo grafico

Il comando CGS cancella quanto visualizzato sullo schermo grafico lasciando il sistema di coordinate.

Sintassi

(CGS)

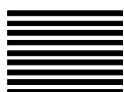


DGS (DCG) - Disabilitazione della scala grafica

Il comando DGS disabilita lo schermo grafico e cancella la visualizzazione e il sistema di coordinate dallo schermo. Dopo aver utilizzato il comando DGS sarà necessario dare un altro comando UGS per reinizializzare lo schermo grafico.

Sintassi

(DGS)



DIS - Visualizzazione di una variabile

Il comando DIS consente all'operatore di visualizzare dei valori nella zona video riservata al colloquio con l'operatore.

Sintassi

(DIS,operando [,operando] [,operando] [,operando] [,operando])

dove:

operando Può essere un numero, una variabile o una stringa ASCII. Possono essere visualizzati fino a cinque operandi che nell'insieme non possono eccedere gli 80 caratteri.

Se *operando* è un numero, può assumere i valori tipici delle variabili (formato 5,5).

Se *operando* è una variabile, può essere qualsiasi variabile utilizzata nei blocchi di assegnazione.

Se *operando* è una stringa ASCII, può essere un messaggio per l'operatore. Il messaggio può avere fino a 80 caratteri ASCII. Nel comando DIS il testo del messaggio deve essere scritto tra doppi apici (" ")

Esempi:

(DIS,100)	visualizza il valore 100
(DIS,E27)	visualizza il valore corrente di E27
(DIS,MSA)	visualizza il valore corrente di MSA (valore sovrametallo)
(DIS,"QUESTO E' UN ESEMPIO")	visualizza la stringa QUESTO E' UN ESEMPIO.

FINE CAPITOLO

MODIFICA DELLA SEQUENZA DI ESECUZIONE DEL PROGRAMMA

GENERALITÀ

Questo capitolo descrive i comandi che consentono al programmatore di modificare l'ordine di esecuzione di un part-program e per disattivarne o sospenderne l'esecuzione. Questi comandi vengono usati per:

- ripetere parti di programma
- eseguire sottoprogrammi
- modificare il flusso del programma
- impostare tempi di attesa
- rilasciare il programma o sospenderne l'esecuzione
- definire i device per operazioni di accesso a file

Comandi per la ripetizione di parti di programma

I comandi in questa classe sono:

COMANDO	FUNZIONE
RPT	Apri la ripetizione di una serie di blocchi di programma
ERP	Chiude la ripetizione di una serie di blocchi di programma

Questi comandi permettono ad una serie di blocchi di programma di essere eseguiti più volte. Possono essere usati per lavorazioni ripetitive quali la perforazione di fori multipli.

Comandi per l'esecuzione di sottoprogrammi

I comandi di questa classe sono:

COMANDO	FUNZIONE
---------	----------

CLS,CLD,CLT	Chiama una subroutine per l'esecuzione
PTH	Dichiara il pathname di default per le subroutine

Una subroutine è una sequenza di blocchi che definisce un ciclo di lavorazione. La subroutine è memorizzata come un file separato con nome file proprio. Il controllo esegue i blocchi della subroutine ogni volta che vengono invocati con il comando CLS. La subroutine può essere chiamata in qualunque momento da qualunque punto del programma principale.

Comandi per la modifica del flusso del programma

I comandi di questa classe sono:

COMANDO	FUNZIONE
---------	----------

EPP	Esegue una parte di un part-program delimitata da due etichette
EPB	Esegue un blocco di un part-program
GTO	Effettua un salto durante l'esecuzione del programma
IF ELSE ENDIF	Esegue sessioni di part program in modo condizionato

Il comando GTO fa sì che l'esecuzione di un programma salti ad un blocco che contiene un'etichetta specifica. Un salto può essere incondizionato o condizionato dai parametri E, dai segnali logici della macchina o da valori numerici.

I salti condizionati si verificano solo se il risultato del confronto è vero. Non si verifica alcun salto se il confronto è falso.

I comandi IF, ELSE, ENDIF permettono di condizionare e modificare il flusso di esecuzione del part program senza necessità di definire labels e salti.

Comandi di ritardo esecuzione programma e di disabilitazione di blocchi barrati

I comandi di questa classe sono:

COMANDO	FUNZIONE
DLY	Causa un ritardo nell'esecuzione del programma
DSB	Disabilita i blocchi barrati

I comandi di ritardo possono essere usati per ritardare l'esecuzione del programma per scopi di sincronizzazione.

Comandi per il rilascio e la sospensione dell'esecuzione del programma

I comandi di questa classe sono:

COMANDO	FUNZIONE
REL	Rilascia il part program
WOS	Mette il programma in attesa di un segnale

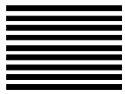
Comandi per la definizione di device

I comandi di questa classe sono:

COMANDO	FUNZIONE
GDV	Definisce il device remoto o il drive A per operazioni di accesso a file
RDV	Rilascia il device definito con GDV

Questi comandi definiscono i device remoti o il drive A e vanno utilizzati quando si eseguono operazioni di lettura/scrittura su file tramite il linguaggio ASSET.

COMANDI PER LA RIPETIZIONE DI PARTI DI PROGRAMMA



RPT-ERP

I comandi RPT e ERP definiscono una parte di programma che deve essere ripetuta un numero specificato di volte. La serie di blocchi comincia con il comando RPT e termina con il comando ERP.

Sintassi

(RPT,*n*)

·
·
·

blocchi

·
·
·

(ERP)

dove:

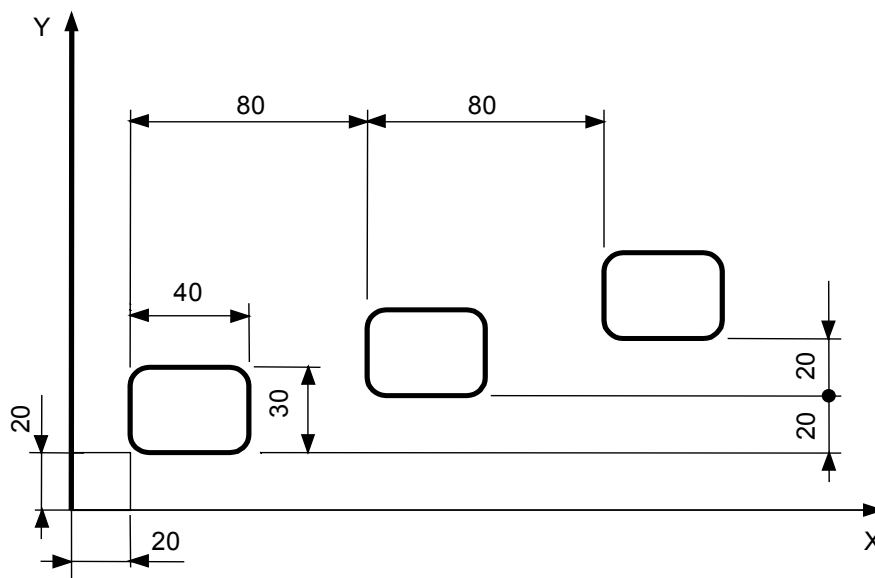
n E' il numero di ripetizioni della parte di programma. Deve essere un numero intero compreso tra 1 e 65535. Si può programmare direttamente con un numero o indirettamente con un parametro E. Il controllo accetta cinque livelli di annidamento.

blocchi E' una serie di blocchi che deve essere eseguita *n* volte.

IMPORTANTE

L'istruzione "GTO" può essere programmata all'interno di una sessione di RPT-ERP, ma se a fine programma non sono stati eseguiti tutti i cicli programmati con RPT il sistema ritorna il seguente errore:

"NC063 RPT/EPP APERTO A FINE PROGRAMMA"

Esempio 1:

Programma:

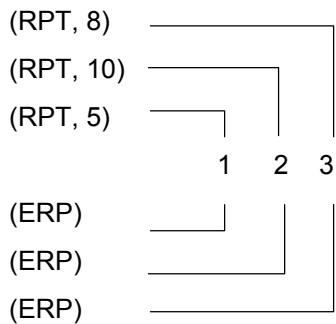
```

(UGS,X,-50,100,Y,-50,100)
(DIS,"N.3 CAVE")
(DIS,"FRESA D12")
N1 S600 T6.6 M6
N2 (RPT,3)
N3 X40 Y35 M3
N4 Z2
N5 (RPT,2)
N6 G91 Z-8
N7 G90 G1 G41 X40 Y20 F300
N8 X60
N9 Y50
N10 X20
N11 Y20
N12 G40 X40
N13 Y35 F1000
N14 (ERP)
N15 G Z2
N16 (UIO,X80,Y20)
N17 (ERP)
N18 (UAO,0)
N19 Z20
N20 X Y M30

```

Esempio 2:

Questo esempio illustra come vengono annidati tre livelli di ripetizione.

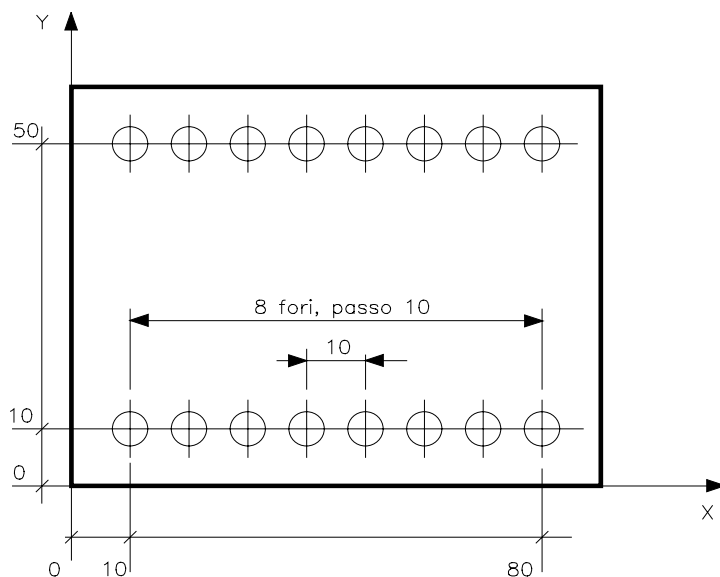


Lavorazione di fori equidistanti

Il seguente è un esempio di comando di ripetizione usato per la lavorazione fori equidistanti.

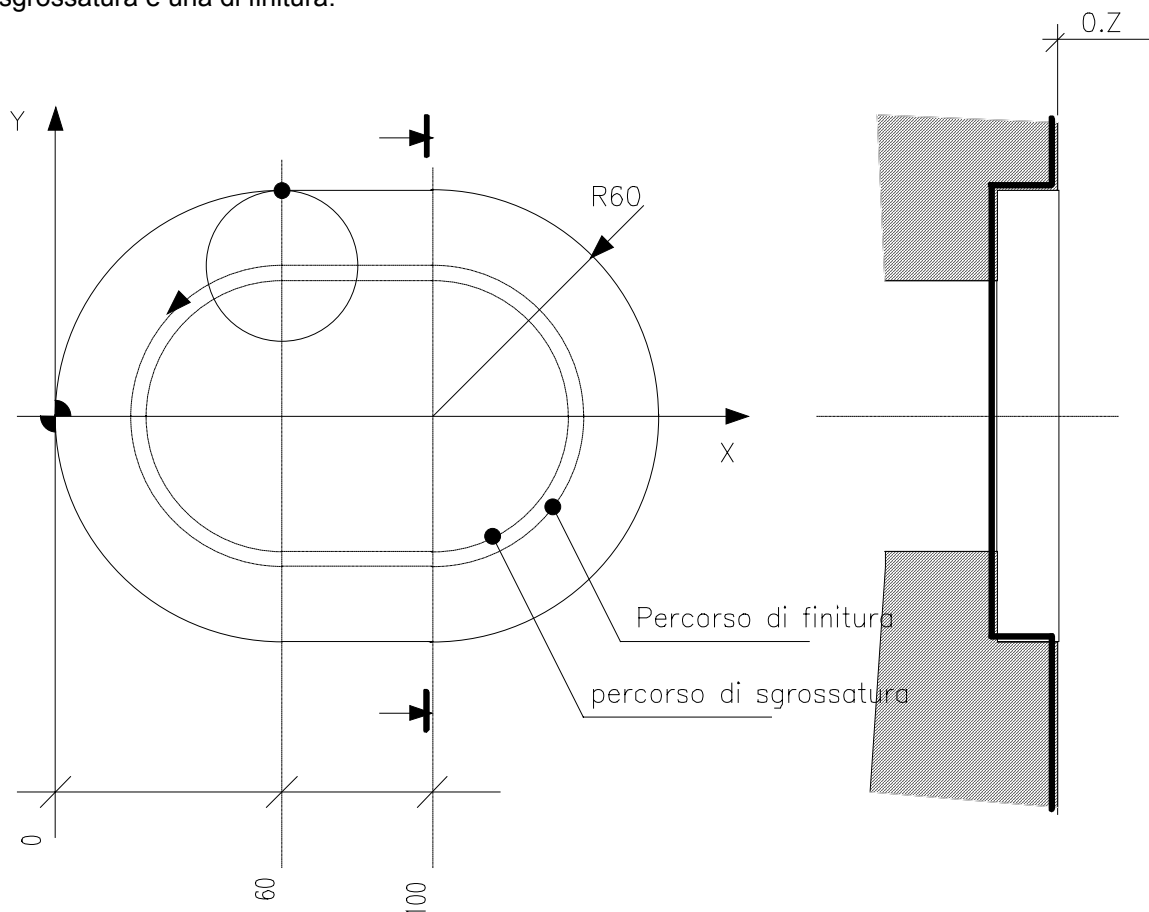
Programma:

```
(UGS,X,-50,100,Y,-50,100)
(DIS,"FORI EQUIDISTANTI")
N1 F200 S900 T1.1 M6
N2 G81 R5 Z-10 M3
N3 X10 Y10
N4 (RPT,7)
N5 G91 X10
N6 (ERP)
N7 Y40
N8 (RPT,7)
N9 X-10
N10 (ERP)
N11 G80 G90 XY M5
```



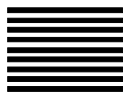
Lavorazione con tagli di sgrossatura e finitura

Il seguente è un esempio di comandi di ripetizione usati per la lavorazione con una passata di sgrossatura e una di finitura.



Programma:

```
(UGS,X,-20,150,Y,-65,60)
(DIS,"DEFINIZIONE DI SOVRAMMETALLO")
N1 S350 T6.6 M6
N2 X60 Y M3
N3 Z-50
N4 MSA=0.5
N5 (RPT,2)
N6 G1 G41 X60 Y60 F500
N7 G3 Y-60 I60 J
N8 G1 X100
N9 G3 Y60 I100J
N10 G1 G40 X60
N11 MSA=0
N12 (ERP)
N13 GZ20 M5
N14 X Y M30
```

COMANDI PER L'ESECUZIONE DI SOTTOPROGRAMMI
**CLS – CLD – CLT - Chiamata di subroutine**

I comandi CLS, CLD e CLT richiamano l'esecuzione di una subroutine cioè di un programma separato memorizzato in un file. Tali comandi possono chiamare una subroutine da un programma principale o da un'altra subroutine. Si possono avere sino a quattro livelli di annidamento.

CLS	Chiamata a subroutine Statica:	analizzata all'atto dell'attivazione del part program principale
CLD	Chiamata a subroutine Dinamica:	analizzata durante l'esecuzione del part program e solo se il nome non è stato trovato prima
CLT	Chiamata a subroutine Temporanea:	analizzata durante l'esecuzione del part program ogni volta che viene incontrata la sua chiamata

Sintassi**(CLS,nome)****(CLD,nome)****(CLT,nome)**

dove:

nome

Nome del file della subroutine.

nome deve essere specificato tramite una stringa di caratteri maiuscoli, oppure una variabile stringa preceduta dalla chiave "?" (punto interrogativo).

nome può specificare sia un nome file tipo SERIE 10 (40 caratteri alfanumerici) sia un file con nome di tipo DOS (8 caratteri più eventuali estensione e percorso).

Esempio:

(CLS,E:\FILE\PROGRAM.PRG) ;chiamata a subroutine con nome DOS

Esempio:

(CLS,PROGRAM.MIO)

NOTE:

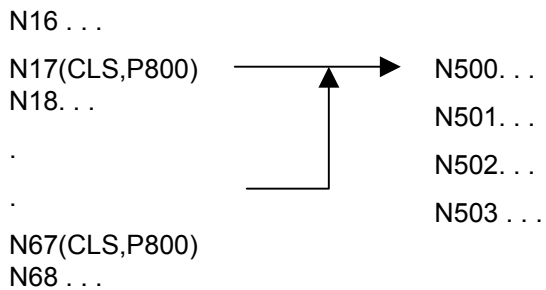
- Se *nome* è una variabile stringa preceduta dalla chiave "?" la subroutine non viene analizzata all'atto dell'attivazione del part program principale ma solo durante l'esecuzione dello stesso. Questo comporta che se la subroutine è stata chiamata attraverso il triletterale:
 - CLS: tale subroutine non debba contenere istruzioni di salto
 - CLD o CLT: la subroutine può contenere istruzioni di salto in quanto analizzate run time

- Se il MAIN è in un direttorio SERIE 10 (con nomi file di 48 caratteri) la subroutine non può risiedere in un direttorio DOS, quindi nel triletterale CLS non si può specificare un *pathname* completo.
- Il *pathname* da utilizzare per il richiamo delle subroutine che risiedono su direttori DOS può essere dichiarato anche tramite il triletterale PTH descritto più avanti in questo capitolo. Se il *pathname* è omissso la ricerca del direttorio di appartenenza della subroutine avviene secondo la seguente modalità:
 - 1. caso:** non è stato specificato nessun *pathname* tramite PTH.
Il sistema cerca la subroutine nel direttorio in cui si trova il programma chiamante e, se qui non esiste, la cerca nei direttori DOS a cui sono stati associati dei nomi logici in fase di caratterizzazione della macchina.
 - 2. caso:** è stato specificato un *pathname* tramite l'istruzione PTH.
Il sistema cerca la subroutine nel direttorio specificato con PTH e, se qui non esiste, la cerca nei direttori DOS a cui sono stati associati i nomi logici.
- Per la **modalità PASSANTE**: è stata mantenuta la compatibilità con l'istruzione CLS, ossia, se la subroutine chiamata contiene delle label restituisce errore, in quanto tale modalità non prevede un'attivazione del programma principale a monte. Le CLD, CLT e paramacro, invece, vengono gestite analogamente alla modalità non passante.
- Il **numero di sottoprogrammi e label "statici"** gestito dal controllo dipende dai corrispondenti valori configurati in AMP. In caso di tabelle piene viene attivata una procedura di gestione della memoria che va a liberare, dove è possibile, le tabelle dalle subroutine dinamiche e paramacro, compattando quelle statiche (CLS). Ogniqualvolta non sia più possibile liberare spazio in memoria viene visualizzato il messaggio di tabelle piene e il sistema si pone nelle stesse condizioni di errore che si avrebbero durante una selezione del part program. L'operatore può quindi procedere modificando il partprogram o aumentando i limiti massimi per le tabelle.
- Il fatto di **non bloccare l'esecuzione** di un programma a fronte di tabelle piene, grazie al compattamento della memoria, va a scapito di un rallentamento nell'esecuzione del programma dovuto alla eventuale successiva fase di analisi; fase peraltro inevitabile in caso di errore bloccante.
- Il superamento della fase di attivazione di un programma non dà la garanzia che questo poi venga eseguito senza errori di memoria piena. Questo, ad es., è il caso di una CLD che chiama un numero tale di CLS da riempire le tabelle.
- **Rieseguire un programma** principale può generare il messaggio di tabelle piene anche se la prima volta (dopo la sua attivazione) è stato eseguito interamente. Questo perchè dopo l'attivazione di un programma le tabelle non contengono i nomi files e le labels di subroutine statiche chiamate da subroutine dinamiche o temporanee.
- L'ordine di **chiamata ad uno stesso nome** di programma attraverso triletterali di tipo differente (ad es. CLT prima e CLS dopo) può generare il messaggio di etichetta sconosciuta; messaggio che può essere ovviato sostituendo la chiamata CLS con una chiamata CLD.
- La **modifica run time di una subroutine**, senza la successiva riattivazione del programma principale, verrà interpretata solo se la sua chiamata sarà effettuata tramite CLT; in caso contrario verrà rieseguita quella precedentemente analizzata con conseguenze imprevedibili sull'esecuzione del programma.

ESEMPIO 1:

Di seguito è riportato un esempio di chiamata a una subroutine.

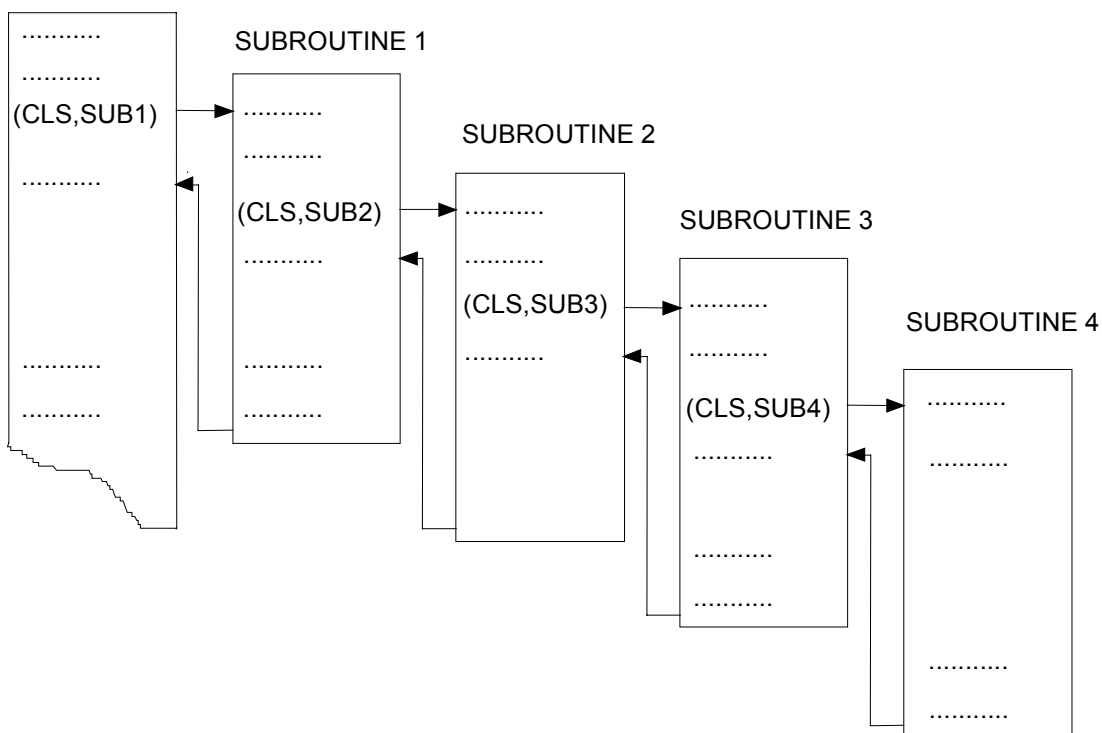
Programma principale		Subroutine P800
----------------------	--	-----------------



ESEMPIO 2:

Sequenza di esecuzione di quattro subroutine annidate.

MAIN PROGRAM



ESEMPIO 3:

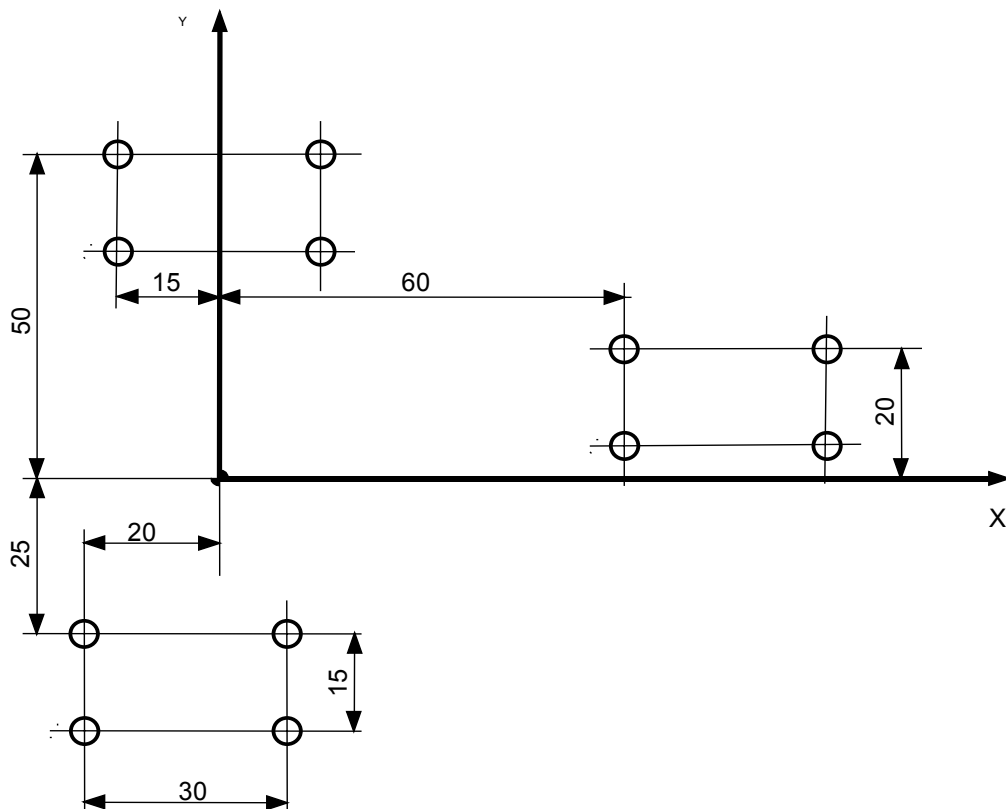
Chiamata alla subroutine specificata in modo implicito

```

SC0.5="PIPP0"
(CLS,?SC0.5)
    
```

Esempio 4:

Uso di subroutine per più operazioni di foratura.



Programma principale:

```

N19 (DIS,"...")
N20 S2000 F180 T2.02M6
N21 (UTO,1,X-20,Y-25)
N22 (CLS,S600)
N23 (UTO,1,X-15,Y50)
N24 (CLS,S600)
N25 (UTO,1,X60,Y20)
N26 (CLS,S600)
N27 Z0
  
```

Subroutine S600

```

N501 G81 R-108 Z-130 M3
N502 XY
N503 Y-15
N504 X30
N505 Y0
N506 G80
  
```

Esempio 5:

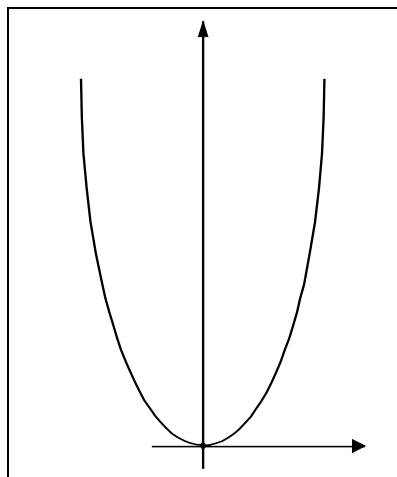
Esecuzione di un profilo parabolico usando un sottoprogramma parametrizzato.

Programma principale:

```
;N10 MAIN PAR
N20 T1.1M6 S1000 M3 F700
N30 E30=72.795           ;Valore iniziale di X
N40 E31=24.28            ;Lunghezza focale
N50 E32=2                ;Incremento di Y
N60 E33=108.24           ;Valore iniziale di Y
N70 E34=0                ;Valore finale di Y
N80 GX0 Y120
N90 (CLS,PAR)
```

Sottoprogramma PAR:

```
;N500 PAR
;N501 Sottoprogramma parametrizzato: esecuzione completa di parabola.
N502 G1 G42 XE30 YE33
N503 E36 = E33
"START" N504
N505 E36=E36-E32
N506 (GTO,END,E36<E34)
N507 E35=SQR( 2*E31*ABS(E36))
N508 XE35 YE36
N509 (GTO,START)
"END" N510
N511 E35=SQR( 2*E31*ABS(E34))
N512 XE35 YE34
;N513 Seconda parte della parabola
N514 E42=E32
N515 E43=E34
N516 E44=E33
N517 XE35 YE43
"START2" N518
N519 E43=E43+E42
N520 (GTO,END2,E43>E44)
N521 E35= -(SQR( 2*E31*ABS(E43)))
N522 XE35 YE43
N523 (GTO,START2)
"END2" N524
N525 E35= - (SQR(2*E31*ABS(E44)))
N526 G40 XE35 YE44
N527 GX0
```

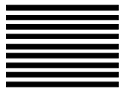


ESEMPIO 6:

Chiamata alla subroutine dinamica, specificata in modo implicito con sottoprogramma PAR dell'esempio precedente

Programma principale:

```
;N10 MAIN PAR
N20 T1.1M6 S1000 M3 F700
N30 E30=72.795           ;Valore iniziale di X
N40 E31=24.28            ;Lunghezza focale
N50 E32=2                ;Incremento di Y
N60 E33=108.24           ;Valore iniziale di Y
N70 E34=0                ;Valore finale di Y
N80 GX0 Y120
SC0.5="PAR"
(CLD,?SC0.5)
```



PTH - Dichiarazione del *pathname* di default

Il comando PTH dichiara il *pathname* da utilizzare come default nelle chiamate a subroutine e paramacro identificate tramite nomi DOS.

Sintassi

(PTH,modo [,pathname])

dove:

modo può assumere i seguenti significati:

modo=0

Viene assunto come *pathname* di default per le chiamate CLS, il *path* del programma principale.

modo=1

Dichiara il *path* da utilizzare a fronte di una istruzione (PTH,2).

Questa modalità può essere utile se nel part program deve essere richiamato più volte, tramite (PTH,2), un *pathname* definito inizialmente con (PTH,1,*pathname*).

modo=2

Attiva il *path* specificato nell'istruzione oppure, se non specificato, quello dichiarato in precedenza con un'istruzione (PTH,1,*pathname*).

pathname Percorso da dichiarare come *path* per le chiamate CLS; tale parametro è opzionale.

NOTA:

Il *pathname* dichiarato tramite il triletterale PTH viene mantenuto anche dopo un RESET del processo nel quale è stato programmato.



EPP - Esecuzione di una porzione di programma

Il comando EPP consente di eseguire una porzione di un programma delimitata da due blocchi con campi etichetta.

Sintassi

(EPP,etichetta1,etichetta2)

dove:

etichetta1 Campo etichetta del primo blocco da eseguire. Un'etichetta è una stringa alfanumerica lunga al massimo sei caratteri.

etichetta2 Campo etichetta dell'ultimo blocco della parte di programma che si vuole eseguire.

Caratteristiche:

In un blocco l'etichetta va programmata tra doppi apici ("ETICHETTA1") mentre in un comando EPP le etichette vanno dichiarate senza doppi apici. Il sistema accetta fino a cinque livelli di annidamento.

In una contornitura, questa prestazione può essere utilizzata, ad esempio, per richiamare in fase di finitura un profilo programmato per la fresa di sgrossatura. E' chiaro che in fase di sgrossatura va programmato un sovrammetallo di finitura con il codice MSA.

In operazioni di punto-punto è possibile programmare tutti i punti su cui avviene ad esempio la centrinatura e successivamente richiamare per gli utensili di foratura, barenatura, ecc. solo i punti interessati. Il comando EPP può essere usato per eseguire un'operazione completa di lavorazione a più orientamenti sul piano di interpolazione attivo.

Esempio 1:

```

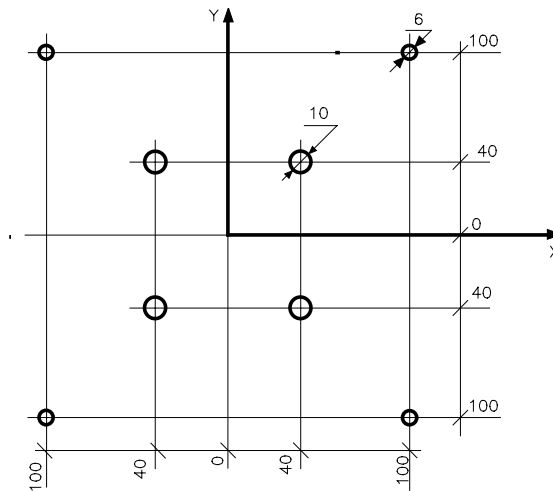
.
.
.
"START"N25          Primo blocco con etichetta
.
.                  Qui non si può richiamare un EPP
.
"END"N100           Ultimo blocco con etichetta
.
.
.
N150 (EPP,START,END)  Comando EPP che specifica le etichette. Il controllo esegue i
                      blocchi da N25 a N100, quindi riprende l'esecuzione dal blocco
                      che segue il blocco del comando EPP,N150.
```

IMPORTANTE

Si determina la condizione di errore **"NC062 ANNIDAMENTO DI EPP MAGGIORE DI 5"** se si annidano più di cinque istruzioni EPP. L'istruzione GTO può essere programmata nella porzione di part-program descrittta da EPP, ma se a fine programma non è stata trovata l'etichetta che chiude l'esecuzione di EPP il sistema ritorna l'errore **"NC063 RPT/EPP APERTO A FINE PROGRAMMA"**

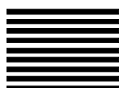
ESEMPIO 2:

Come usare il comando EPP in un'operazione di centrinatura:



Programma:

```
(UGS,X,-110,110,Y,-110,110)
N1 (DIS,"CENTRINATURA")
N2 F300 S2000 T1.1 M3 M6
N3 G81 R0 Z-3
"D6" N4
N5 X100 Y100
N6 X-100
N7 Y-100
N8 X100
"D10" N9
N10 X40 Y40
N11 X-40
N12 Y-40
N13 X40
"FINE" N14
N15 G80
N16 (DIS,"PUNTA D6")
N17 F200 S1800 T2.02 M3 M6
N18 G81 R Z-22
N19 (EPP,D6,D10)
N20 G80
N21 (DIS,"PUNTA D10")
N22 F220 S1600 T3.3 M3 M6
N23 G81 R Z-24
N24 (EPP,D10,FINE)
N25 G80
```



EPB - Esecuzione di un blocco di part program

Il trilettale EPB permette di eseguire un blocco di part program .

L'esecuzione può essere condizionata dal risultato di un confronto specificato nel comando; se la condizione non è soddisfatta, si può definire un blocco di part program alternativo da eseguire.

Sintassi

(EPB, blocco_di_part_program1 [, par1 operatore par2 [, blocco_di_part_program2]])

dove :

blocco_di_part_program1 E' il blocco di part program che verrà eseguito.

Può essere una stringa tra apici o una variabile di tipo carattere locale o di sistema.

par1 E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro *par2*.

operatore Sono operatori logici che possono essere utilizzati nelle espressioni :

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

par2 E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro *par1*.

blocco_di_part_program2 E' il blocco di part program che verrà eseguito solo se la condizione specificata per l'esecuzione del *blocco_di_part_program1* non è stata soddisfatta. E' un parametro opzionale.

Può essere una stringa tra apici o una variabile di tipo carattere, locale o di sistema.

Caratteristiche:

Se *par1* ,*operatore* e *par2* non sono stati specificati, il programma esegue sempre il blocco *blocco_di_part_program1*.

E' accettato dal sistema un solo livello di annidamento del triletterale EPB.

ESEMPIO:

(EPB, "(EPB,'E1=1')") : accettato dal sistema
 (EPB, "(EPB,'(EPB,SC0.100)')") : NON accettato dal sistema (errore FORMAT ERROR)

I blocchi di part program specificati nel triletterale EPB non vengono analizzati in fase di attivazione del part program, per cui e' cura del programmatore far sì che questi blocchi non possano creare malfunzionamenti al part program.

Esempi:

(EPB,"(CLS,SUBROUT)") : il sottoprogramma SUBROUT non viene preanalizzato in fase di attivazione, non potrà quindi contenere istruzioni di salto (GTO).

(EPB,' "LABEL" ') : la label LABEL non verrà inserita nella tabella label del part program.)

(EPB, " (DIS, E1) ") : equivale al blocco (DIS, E1)

SC0.30= " E1 =10"
 (EPB, SC0.30) : equivale al blocco E1=10

SC40.30 = "# X10"
 (EPB, SC40.30) : equivale al blocco #X10

(EPB,"(CLS,SUBROUT)",E1=34) : chiama il sottoprogramma SUBROUT solo se E1=34

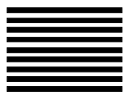
(EPB," E1=100", SN1=25," E1=0") : assegna 100 ad E1 solo se SN1=25 , altrimenti assegna 0 ad E1

(EPB,"(EPP,LAB1,LAB2)","SC0.2="OK","(EPP,LAB3,LAB4)") : esegue dalla label LAB1 alla label LAB2 se SC0.2 e' uguale a OK altrimenti esegue dalla label LAB3 alla label LAB4.

(EPB, "(EPB,' E1 = 2', E0 < 100)", E0 > 70) : assegna 2 ad E1 solo se E0 e' compreso tra 70 e 100.

(EPB," (EPB,' E0 = 5', E0 < 5) ",E0 < 10,E0=10) : assegna ad E0 il valore 5 se E0<5; assegna ad E0 il valore 10 se E0>10.
 Se 5<E0<0, E0 rimane invariata.

COMANDI DI SALTO E DI ATTESA



GTO - Comando di salto

Il comando GTO esegue un salto ad un blocco identificato con un'etichetta.

Il salto può essere condizionato dal risultato di un confronto specificato nel comando .

Sintassi

(GTO,etichetta [,par1 operatore par2])

dove:

<i>etichetta</i>	Etichetta del blocco di programma a cui saltare. Un' etichetta è una stringa costituita da un massimo di sei caratteri alfanumerici. Nel blocco in cui vi è l'etichetta, essa deve essere tra doppi apici, invece in questo contesto, l'etichetta deve essere programmata senza apici. L'etichetta può inoltre essere specificata con una variabile di tipo carattere preceduta dalla chiave "?" (punto interrogativo).
<i>par1</i>	E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro <i>par2</i> .
<i>operatore</i>	Operatori logici che possono essere utilizzati nelle espressioni: = uguale < minore di > maggiore di <> diverso <= minore o uguale >= maggiore o uguale
<i>par2</i>	E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro <i>par1</i> .

Caratteristiche:

Se *par1*, *operatore*, e *par2* non sono stati specificati, il programma salta sempre al blocco descritto con l'etichetta.

L'etichetta su cui posizionare il part program può essere espressa anche con una variabile locale o di sistema (preceduta dalla chiave ?).

Ciò permette di semplificare i part program con un elevato numero di blocchi programmati per modificarne il flusso di esecuzione.

In particolare, programmando l'etichetta con una variabile, è possibile programmare una etichetta di default, chiamandola "DEFLAB": il part program si posiziona su di essa se l'etichetta contenuta nella variabile non esiste.

Esempio 1:

```
;SC0.3 contiene l'etichetta su cui posizionarsi
(GTO,?SC0.3)
"UNO"
.
.
.
.
"DUE"
.
.
.
.
"TRE"
.
.
.
.
"DEFLAB"
.
.
.
```

Se SC0.3 non contiene "UNO", "DUE", o "TRE" il part program si posiziona sul blocco "DEFLAB". Se l'etichetta "DEFLAB" non fosse programmata verrebbe visualizzato l'errore **"NC054 UNDEFINED LABEL"** e l'esecuzione del part program verrebbe interrotta.

Esempio 2:

N01 (GTO,START)	salto incondizionato all'etichetta "START"
N10 (GTO,END,E1 > 123)	salto ad "END" se il valore di E1 è maggiore di 123
N20 (GTO,LAB1,@COOLANT= 1)	salto a "LAB1" se la variabile PLUS @COOLANT è attivata
N30 (GTO,START,E1 <> E5)	salto a "START" se il valore di E1 è diverso da quello di E5
N40 (GTO,LAB1,SC1.2H = "OK")	salto a "LAB1" se i due caratteri che iniziano con SC1 sono uguali ad OK
N50 SC1.3="ABC"	prepara la variabile per blocco successivo
N60 (GTO,?SC1.3)	salto alla label ABC

Esempio 3:

L'istruzione:

```
(GTO,END,SC2.3H = "ABC")
```

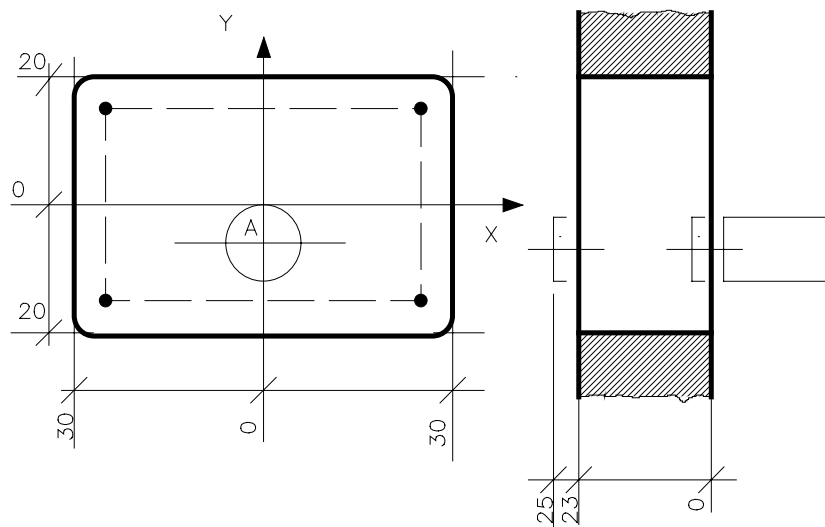
se i tre caratteri (.3) che iniziano da SC2 sono ABC il programma salta all'etichetta "END".

NOTA:

Nel comando di salto i caratteri da confrontare vanno programmati tra doppi apici.

Esempio 4:

Esempio di uso di salto condizionato per la fresatura di una cava.



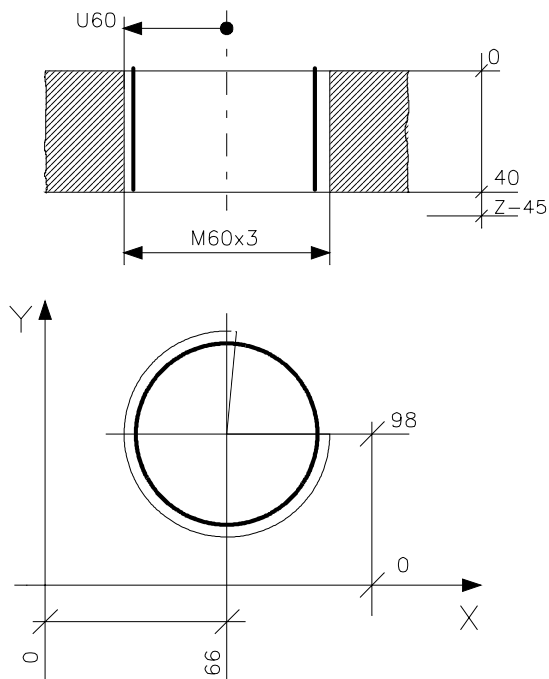
Programma:

```

(UGS,X,-50,50,Y,-50,50)
N1 (DIS,"FRESATURA CAVA")
N2 F500 S2000 T1.1 M3 M6
N3 E31=-3.5
N4 E32=-24
"START"
N6 G X Y-10
N7 Z(E31)
N8 G1 G42 X Y-20
N9 X-30
N10 Y20
N11 X30
N12 Y-20
N13 G40 X
N14 Y-10
"END"
N16 E31=E31-3.5
N17 (GTO,START,E31>E32)
N18 E31=-25
N19 (EPP,START,END)
N20 G Z10
  
```

ESEMPIO 5:

Uso di salto condizionato per l'esecuzione di una filettatura cilindrica.



E30 = Diametro 1^ passata

E31 = Incremento della profondità di passata (diametrale)

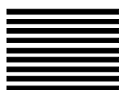
E32 = Diametro di ritorno

E33 = Diametro finale

Programma:

```

N1 (DIS,"FILETTATURA DIAMETRO 60")
N2 S150 T5.5 M6
N3 G0 X66 Y98 Z5 M3
N4 E30=56.8
N5 E31=0.5
N6 E32=50
N7 E33=60
"I" N8
N9 G0 Z5
N10 U(E30)
N11 G33 Z-45 K3
N12 GU(E32)
N13 E30=E30+E31
N14 (GTO,F,E30>E33)
N15 (GTO,I)
"F" N16
N17 GU(E32)
N18 Z5
N19 U(E33)
N20 G33 Z-45 K3
N21 GU(E32)
N22 Z20
    
```



IF ELSE ENDIF

Il comando IF permette di aprire una sessione di part program che verrà eseguita solo se la condizione specificata nel comando è vera. Questa sessione di part program deve terminare con il comando ENDIF: al suo interno può contenere opzionalmente il comando ELSE che delimita, fino al comando ENDIF una sessione di part program che verrà eseguita se la condizione

specificata nel comando IF non è vera.

Sintassi

```
( IF, par1 operator par2 )
( ELSE )
( ENDIF )
```

dove :

par1 E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro par2.

operatore Sono operatori logici che possono essere utilizzati nelle espressioni:

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

par2 E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro par1.

Caratteristiche.

Se la condizione specificata nel comando IF è vera i blocchi specificati fino al comando ELSE (se esiste) o fino al comando ENDIF (se ELSE non esiste) verranno eseguiti, altrimenti verranno ignorati.

I blocchi compresi tra il comando ELSE e il comando ENDIF verranno eseguiti solo se la condizione specificata nel comando IF non è vera .

Si possono annidare fino a 5 livelli di cicli IF ELSE ENDIF.

A ogni comando IF deve corrispondere un comando ENDIF.

Esempi:

```
( INP," E0 VALUE " ,30 , E0)
( IF , E0 = 3)
( DIS , " E0 is equal to 3 ")
(ELSE)
( IF E0 > 3)
( DIS , " E0 is greater than 3 ")
(ELSE)
( DIS , " E0 is less than 3 ")
( ENDIF)
( ENDIF)
```

```
( INP," E0 VALUE" ,30 , E0)
( IF , E0 >10 )
( IF , E0 >20 )
( IF , E0 >30 )
( IF , E0 >40 )
( IF , E0 >50 )
( DIS , " E0 is greater than 50 ")
( ENDIF)
( ENDIF)
( ENDIF)
( ENDIF)
( ENDIF)
```



DLY - Definizione del tempo di attesa

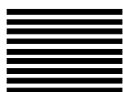
Il comando DLY permette di programmare una sosta durante l'esecuzione del programma.

Sintassi

(DLY,tempo)

dove:

<i>tempo</i>	Tempo di sosta espresso in secondi (valore minimo 0.1 Secondi). Si può programmare il valore del tempo con una costante numerica o con un parametro E.
--------------	--



DSB - Disabilitazione di blocchi barrati

Questo comando viene utilizzato per abilitare/disabilitare i blocchi barrati;

I blocchi barrati contengono il segno "/" come primo carattere e la loro esecuzione è condizionata al valore di DSB.

Sintassi

DSB= *valore*

dove:

valore

Il valore può essere:

DSB=0 i blocchi barrati vengono eseguiti

DSB=1 i blocchi barrati non vengono eseguiti



REL - Deseleziona il part program

Il comando REL disattiva il part program e tutte le eventuali subroutine.

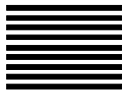
Sintassi

(REL)

Caratteristiche:

La funzione REL disattiva il part program; essa è programmabile anche in modalità MDI.

Quando il part program è stato disattivato, sul video viene visualizzato il messaggio '**NC058 FINE PROGRAMMA**'.



WOS - WAIT su segnale

Il comando WOS sospende l'esecuzione del part program finché la variabile specificata non soddisfa la condizione voluta.

Sintassi

(WOS,par1 operatore par2)

dove:

<i>par1</i>	E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro <i>par2</i> .
<i>operatore</i>	Operatori logici che possono essere utilizzati nelle espressioni:
=	uguale
<	minore di
>	maggiore di
<>	diverso
<=	minore o uguale
>=	maggiore o uguale
<i>par2</i>	E' una variabile, locale o di sistema, oppure una costante, il cui valore viene confrontato con il valore del parametro <i>par1</i> .

Caratteristiche:

Il processo in cui è programmata la funzione WOS passa nello stato WAIT finché la condizione non è soddisfatta.

Il comando WOS è sincronizzato di default.

Se durante la fase di debug del part program il sistema si trova in stato di WAIT (dovuto alla programmazione di una funzione WOS), è possibile assegnare alla variabile *par1* il valore che soddisfa la condizione specificata nel comando WOS.

Questo si può ottenere eseguendo le seguenti operazioni:

- Premere CYCLE STOP
- Assegnare in MDI il valore corretto alla variabile *par1*
- Premere CYCLE STOP

ESEMPI DI PROGRAMMAZIONE:

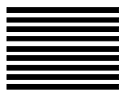
(WOS,E1>4)

(WOS,SC10.5 = "PIPP0")

(WOS,@TOOL_STATUS >= E3)

(WOS,!USER_VAR3.5CH = SC4.5)

COMANDI PER LA DEFINIZIONE DI DEVICE



GDV - Definizione di *device* per accesso a file

Questo triletterale permette di riservare l'uso del drive A per operazioni di scrittura/lettura file e di definire *device* remoti su cui operare.

Sintassi

(GDV, *device*)

dove:

device Nome del device.

Può essere una lettera o, se preceduto dalla chiave " ? ", una variabile locale o di sistema che specifica il drive che si vuole utilizzare.

Per il floppy va usato come identificativo la lettera "A".

Nel caso di device remoto va indicato il nome con cui esso è stato definito tramite l'opzione MINI DNC E65/E66 (es. "K").

Caratteristiche:

Se il drive A è già in uso da un altro processo o UTILITY (es. DOS SHELL), viene generato l'errore **NC251 Driver busy or not configured**; questo errore è gestibile da part program (per maggiori dettagli consultare l'appendice C).

ESEMPI:

(GDV,A)

Riserva l'uso del drive A.

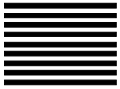
SC0.1='A'

(GDV,?SC0.1)

Viene riservato l'uso del drive A, specificato nella variabile SC0.1

IMPORTANTE

Il RESET di sistema annulla l'accesso al drive A ma non quello ai device remoti.



RDV - Rilascio *device*

Questo trilletterale permette di rilasciare il drive A od il device remoto precedentemente definito con il trilletterale GDV, in modo da renderlo disponibile ad altri utilizzatori.

Sintassi

(RDV, *device*)

dove:

device Nome del device.

Può essere una lettera o, se preceduto dalla chiave ?, una variabile locale o di sistema che specifica il drive che si vuole utilizzare.

Per il floppy va usato come identificativo la lettera "A".

Nel caso di device remoto va indicato il nome con cui esso è stato definito tramite l'opzione MINI DNC E65/E66 (es. "K").

Caratteristiche:

Il reset del processo equivale all'esecuzione automatica di un comando (RDV,A).

Esempi:

(RDV,A)

Rilascia il drive A precedentemente allocato con GDV.

SC0.1='A'

(RDV,?SC0.1)

È analogo al precedente esempio, ma con il drive specificato nella variabile SC0.1.

FINE CAPITOLO

HIGH SPEED MACHINING

CONSIDERAZIONI GENERALI

La prestazione High Speed Machining permette la lavorazione di superfici (profili) definite per punti, generate da sistemi CAD/CAM, su macchine utensili che hanno da 3 a 5 assi (3 lineari + 2 rotativi).

La prestazione High Speed Machining deve essere abilitata in ambiente AMP, selezionando il campo specifico nella sezione PROCESS CONFIG (softkey **PROC CHAR**).



Questa prestazione può essere abilitata solamente per i primi 4 processi.

Per programmare correttamente questa prestazione è necessario:

1. creare un file (part program) di configurazione che contiene tutti i parametri per la gestione del profilo: utensili, assi e cinematica della macchina.

Il file di configurazione verrà richiamato nel programma principale tramite un comando trilettale con il seguente formato:

(HSM, nome file di configurazione)

2. creare il profilo inserendolo direttamente nel programma principale:

Esempio:

```
; ESEMPIO DI PROGRAMMAZIONE HSM
G0 X..Y..Z.. A.. B.. F...
```

```
-----
(HSM, CONFIG1)
-----
```

```
(CLS,PROFILO1 -----> G61
G1 X...Y..Z..A..B..
-----
; FINE PROGRAMMA
-----
G60
```

Il profilo che può anche essere inserito in un file specifico che verrà richiamato come un sottoprogramma tramite l'istruzione **CLS**.

Esempio 2:

; ESEMPIO DI PROGRAMMAZIONE HSM

G1 X..Y..Z.. A.. B.. F...

(HSM, CONFIG1)

G61

G1 X..Y..Z..A..B..

G60

; FINE PROGRAMMA

PROGRAMMAZIONE DEI PUNTI E CARATTERISTICHE DEL PROFILO

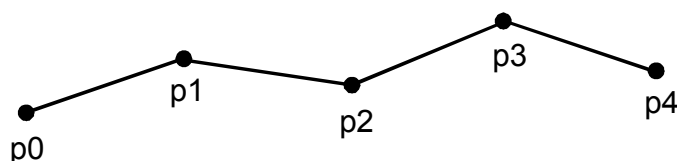
Un profilo è un insieme di punti che formano il part-program ISO della superficie da lavorare generato dal CAD/CAM nel quale devono essere rispettate le caratteristiche descritte in questo capitolo.

Sui punti programmati verrà costruita una curva polinomiale che descriverà la traiettoria da percorrere. Tale traiettoria passerà per i punti programmati a meno di una soglia di tolleranza configurabile. Le modalità di congiungimento dei punti verranno definite dai codici G01 e G00 programmabili assieme ai punti.

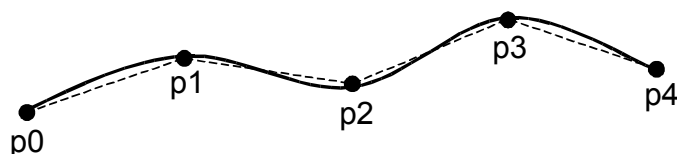
IMPORTANTE

Per un corretto calcolo delle curve polinomiali, si consiglia di programmare i punti utilizzando almeno 5 cifre dopo il punto decimale (es 10.37854); nel caso in cui vengano programmate meno cifre potrebbero insorgere delle piccole irregolarità sul profilo.

I tratti percorsi in **G00** saranno considerati come dei singoli posizionamenti; ogni punto verrà collegato col successivo tramite un moto "lineare" da effettuarsi con le caratteristiche dinamiche dei posizionamento in rapido (ogni tratto in G00 partirà da una velocità nulla e terminerà a velocità nulla). Per tale motivo la modalità G00 non effettuerà il calcolo della curva polinomiale. Al termine di un tratto in G00 non verrà effettuata alcuna sosta (sincronismi di fine movimento, entrata in tolleranza, ecc..) ma si ripartirà immediatamente col movimento successivo. Tale comportamento è assimilabile alla programmazione in uno stesso blocco dei codici G01 e G09.



Nel caso di **G01** ogni punto verrà geometricamente collegato con i successivi tramite una curva polinomiale, quindi la traiettoria generata potrà essere considerata "**continua**". Tale collegamento verrà interrotto dalla programmazione di una G00 oppure dalla programmazione di appositi codici G descritti di seguito. Le caratteristiche dinamiche dei tratti in G01 sono le stesse delle movimentazioni in "lavoro" (come le normali G01 della programmazione ISO).



Nel profilo oltre alle funzioni **G01** e **G00** , possono essere programmate le seguenti funzioni G, specifiche per la prestazione HSM.

G61

determina l'inizio del profilo e deve essere programmata in un blocco da sola. Quando viene attivata la G61 non deve essere presente alcuna virtualizzazione (UPR,UVP,UVC,TCP).

Prima di attivare la G61 è necessario definire il file di configurazione tramite l'istruzione:

(HSM, nome file di configurazione)

Il comando G61 può essere eseguito SOLO all'interno di un part program in stato di AUTO o BLK BY BLK.

G60

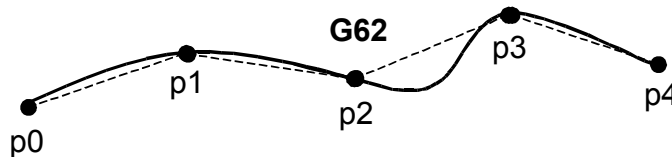
determina la fine del profilo e deve essere programmata in un blocco da sola.

Nel caso in cui la macchina sia in esecuzione single STEP il profilo compreso tra G61 e G60 viene considerato come un'unica istruzione. Per fermarne l'esecuzione è necessario, quindi, passare in stato di HOLD.

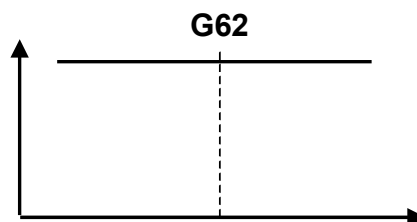
G62

spezza un profilo in due parti e determina il punto in cui finisce un profilo e ne inizia un altro mantenendo però la continuità tra le due curve.

I punti precedenti la G62 verranno usati per generare una prima curva, mentre i punti successivi verranno usati per generarne un'altra. Tali curve saranno raccordate e quindi saranno continue; la seconda curva avrà una inclinazione iniziale pari a quella di uscita della curva precedente.

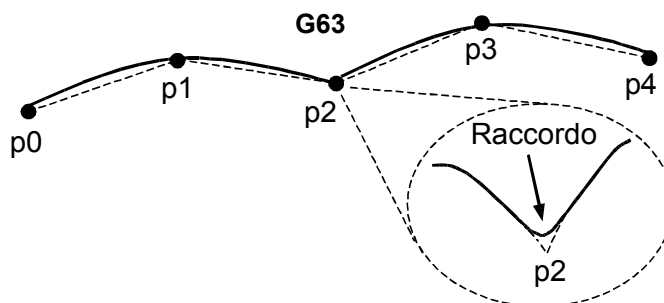


A livello dinamico, nel caso di G62 non verrà generata alcuna rampa di decelerazione e accelerazione per raccordare le due curve. Tale G deve essere programmata da sola all'interno di un blocco di part program.



G63

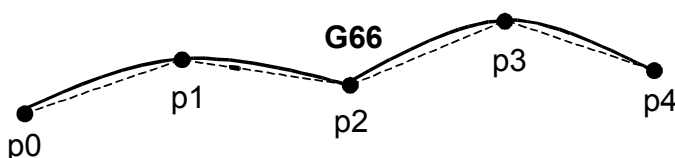
Tramite il codice G63 si potrà definire un punto in cui termina un profilo e ne parte un altro mantenendo però la “continuità” tra le due curve. I punti precedenti la G63 verranno usati per generare una prima curva, mentre i punti successivi verranno usati per generarne un'altra. Mentre tramite la G62 l'inclinazione iniziale della seconda curva dipende strettamente dalla inclinazione finale della prima, con la G63 l'inclinazione iniziale della seconda curva **NON** viene influenzata da quella della prima. Per mantenere la continuità viene inserito un “raccordo” dipendente dalla precisione con cui viene richiesto il calcolo delle splines.



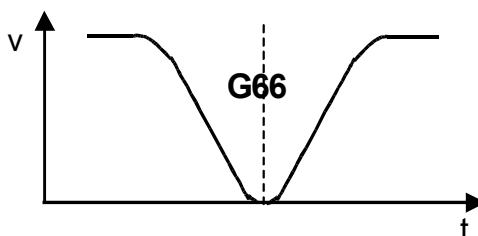
A livello dinamico, nel caso di G63 potrà essere generato un rallentamento sul punto di raccordo delle due curve. Tale G deve essere programmata da sola all'interno di un blocco di part program

G66

spezza un profilo in due parti e determina il punto in cui finisce un profilo e ne inizia un altro creando una **discontinuità** tra le due curve, ovvero il punto precedente la G66 rappresenta uno spigolo. A questo punto verranno generate due curve, la prima utilizzando i punti precedenti la G66, la seconda utilizzando i punti successivi. Tali curve **NON saranno** raccordate, quindi presenteranno una discontinuità.

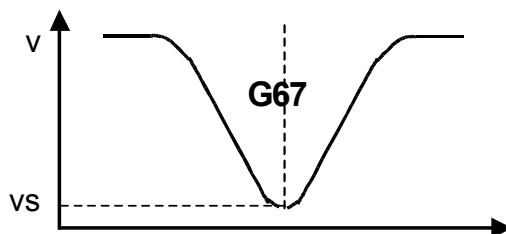


A livello dinamico tale discontinuità verrà affrontata a velocità nulla; quindi, la prima curva verrà terminata con una rampa di decelerazione sino a velocità 0 (zero), e al termine si affronterà la seconda curva con una rampa di accelerazione per raggiungere la velocità di lavorazione richiesta. Tale G deve essere programmata da sola all'interno di un blocco di part program.



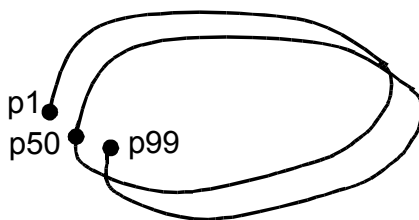
G67

Tramite il codice **G67** si potrà definire una “discontinuità” sul profilo così come definito con la G66. Ciò che cambia è l’approccio dinamico allo spigolo, ovvero non si termina a velocità nulla bensì ad un valore di velocità (v_s) tale per cui gli assi possano affrontare lo spigolo senza avere problemi dinamici. Tale valore di velocità viene calcolato in base alle accelerazioni sopportabili da ogni asse. Tale G deve essere programmata da sola all’interno di un blocco di part program.

**Considerazioni sull’uso delle G62,G63,G66 e G67 (codici di transizione)**

L’utilizzo dei codici G di transizione è particolarmente utile quando si vogliono definire delle curve “simili” e ripetitive (a patto che lo siano anche i punti programmati).

Supponiamo di avere un profilo definito da 100 punti di cui i primi 50 rappresentano la prima passata di lavorazione (da **p1** a **p50**) e gli altri 50 (da **p50** a **p99**) lo stesso profilo leggermente spostato (seconda passata).



Essendo i punti compresi tra **p1** e **p50** “simili” ai punti compresi tra **p50** e **p99** accadrà che le condizioni di calcolo delle due curve polinomiali saranno simili. Verranno pertanto generate due curve quasi uguali e “parallele”.

Nel caso in cui non sia stata programmata la funzione G62 sul punto p50 può accadere che il CN generi delle curve non perfettamente parallele. Questo effetto, normalmente indesiderato, è dovuto al fatto che il calcolo dei polinomi tiene conto della “storia” precedente lungo le traiettorie calcolate.

Appare evidente che la “storia” del punto p1 è diversa da quella del punto p50. Nel caso di p1, infatti, la “storia” è inesistente, mentre in p50 il CN conosce tutto l’andamento dovuto ai primi 49 punti.

Inserendo la funzione G62 si ottiene l’effetto di cancellare la “storia” precedente, ottenendo un andamento geometrico pressoché identico a quello calcolato a partire da p1.

Nel caso di lavorazioni con più passate quindi, la mancata programmazione di G62 avrebbe l’indesiderato effetto di avere diversi livelli di lavorazione tra una passata e l’altra.

STRUTTURA GENERALE DELLA PROGRAMMAZIONE HIGH SPEED MACHINING

Tra i blocchi G61 e G60 sarà possibile programmare solamente i punti che costituiscono il profilo da lavorare oppure i codici G per definirne la modalità di gestione: nessun altro tipo di programmazione sarà permesso.

Per quanto riguarda la programmazione dei punti sarà possibile utilizzare la programmazione assoluta tramite **G90** oppure la programmazione in incrementale con **G91**. Tutti gli eventuali parametri numerici potranno essere definiti direttamente oppure tramite le variabili **E** o **L**: non è valida la programmazione tramite espressioni, quindi la programmazione tipo XE(E2) oppure X(E1+E2) non è permessa, mentre è permessa XE1.

Sintatticamente le linee di programma ammesse saranno :

N... [G00 | G01] [G90 | G91] [punti] F....

N... [G62 | G63 | G66 | G67]

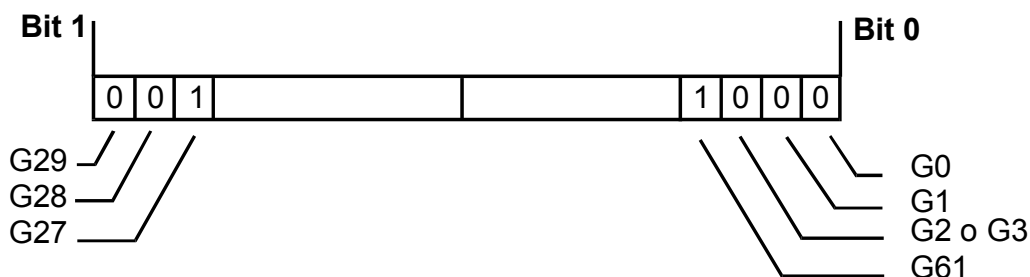
L'attivazione della prestazione HSM (High Speed Machining) **G61** forza implicitamente la programmazione di G01 e G90, in uscita (**G60**) verranno ripristinate le G attive all'atto della programmazione della G61.

Il primo punto programmato **DEVE** essere espresso in quote assolute (G90) e deve contenere la programmazione di tutti gli assi associati alla programmazione HSM (assi configurati nel file di setup di HSM).

Interazione con Logica di Macchina

La sezione di programma G61/G60 verrà considerata, dal punto di vista del sistema, come un unico blocco di programma. A livello di interfacciamento con la logica di macchina, quindi, verrà lanciata una richiesta di consenso al movimento in corrispondenza della G61, ed una richiesta di fine movimento sulla G60 (analogamente a quanto fatto per le varie movimentazioni in continuo G27 e G28).

A livello di consenso al movimento, la variabile **XW03**, contenente il tipo di movimento, verrà compilata come riportato di seguito:



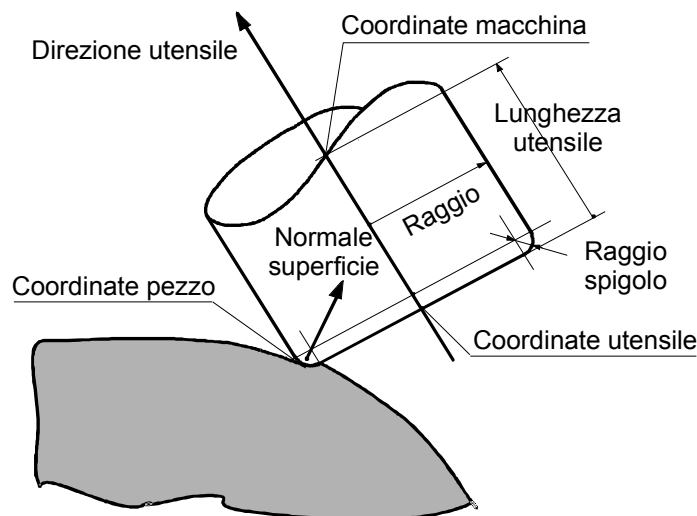
CONVENZIONI PER LA DEFINIZIONE DEI PUNTI

Punti e coordinate di lavorazione

Prima di definire come vengono trattati i punti è necessario specificare che cosa essi rappresentano, in quanto la programmazione può essere effettuata relativamente a tre diversi tipi di coordinate, ovvero:

- Coordinate pezzo (**Cutter Contact Points**), riferite all'effettivo punto di taglio
- Coordinate utensile (**Cutter Location Points**), riferite al punto normalmente indicato come centro utensile
- Coordinate macchina (**Axis Location Points**), riferite ad un arbitrario punto solidale con gli assi di lavorazione

Le coordinate pezzo sono legate alle coordinate utensile attraverso la geometria e l'orientamento dell'utensile stesso. Le coordinate macchina sono legate alle coordinate utensile attraverso la geometria della macchina utensile. Nel caso di lavorazioni a tre assi si tratterà soltanto di una traslazione di coordinate, mentre nel caso di lavorazioni a cinque assi coordinati si tratterà dell'applicazione di matrici di roto-traslazione che tengono conto delle trasformazioni geometriche dovute alla movimentazione degli assi rotativi. Nella figura seguente è possibile vedere che cosa si intende come coordinate pezzo, coordinate utensile e coordinate macchina.



I punti vengono definiti tramite normali coordinate assi nel formato **[Nome asse][Quota]**; ad esempio X100 Y200 Z40.

Direzione Utensile: IJK

La direzione utensile rappresenta l'orientamento dell'utensile (dalla punta all'attacco) all'interno del sistema di riferimento del pezzo.



Nelle successive sezioni, per versore si intende vettore di lunghezza unitaria.

Per definire la direzione utensile si potranno adottare due tecniche. La prima programmando direttamente il vettore che individua la direzione utensile. Tale vettore (di lunghezza unitaria) viene espresso tramite le coordinate **ijk** nel formato:

[i] [Componente Ascissa] [j][Componente Ordinata] [k][Componente Verticale]

Il sistema normalizzerà automaticamente la lunghezza del vettore alla lunghezza unitaria (1.0).

La seconda possibilità di definizione della direzione utensile consiste nella programmazione degli assi rotativi. Il sistema, conoscendo la cinematica della macchina, sarà in grado di determinare automaticamente le tre componenti del vettore **ijk**.

Direzione Normale alla Superficie: MNO

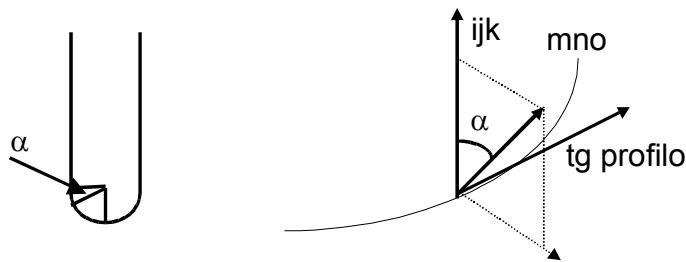
La direzione normale alla superficie rappresenta la direzione della "retta" perpendicolare alla superficie da lavorare (a partire dalla superficie) all'interno del sistema di riferimento del pezzo.

Per definire la direzione normale al pezzo si potranno adottare due tecniche. La prima programmando direttamente il vettore che individua la direzione normale. Tale vettore (di lunghezza unitaria) viene espresso tramite le coordinate **mno** nel formato:

[m] [Componente Ascissa] [n][Componente Ordinata] [o][Componente Verticale]

Il sistema normalizzerà automaticamente la lunghezza del vettore alla lunghezza unitaria (1.0).

La seconda possibilità consiste nel richiedere al sistema il calcolo automatico di tale direzione. La direzione viene calcolata in base alla tangente al profilo (direzione di spostamento), in base alla direzione utensile (vettore **ijk**) ed un angolo α di contatto tra pezzo ed utensile. L'angolo α viene definito per convenzione uguale a 0° se l'utensile tocca il pezzo con la punta, uguale a 90° se tocca con il lato sinistro e -90° se tocca con il lato destro (nell'esempio sottostante α vale 90°). Tale calcolo fa sì che il versore **mno** sia normale alla tangente al profilo e che definisca un angolo α (angolo di contatto) con il versore utensile **ijk**.



Questo tipo di approccio è significativo quando si tratta di “scontornare” un profilo. Nel momento in cui si deve lavorare una superficie questo approccio potrebbe fallire poiché non si hanno informazioni sulla “superficie” da lavorare, ma si posseggono solamente le informazioni relative alla “direzione” di spostamento.

Direzione Applicazione Raggio Utensile: PQD

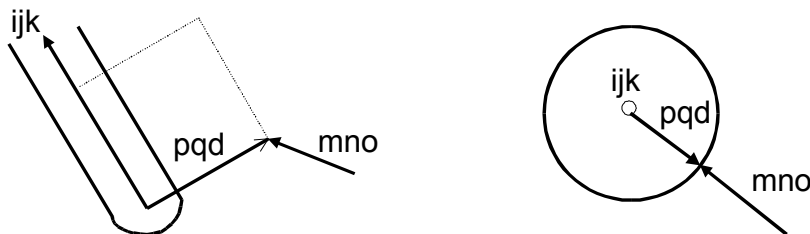
La direzione di applicazione del raggio utensile rappresenta la direzione su cui applicare la compensazione raggio (a partire dal centro dell'utensile) all'interno del sistema di riferimento del pezzo.

Per definire la direzione di applicazione del raggio utensile si potranno adottare due tecniche. La prima programmando direttamente il vettore che individua la direzione. Tale vettore (di lunghezza unitaria) viene espresso tramite le coordinate **pqd** nel formato

[p] [Componente Ascissa] [q][Componente Ordinata] [d][Componente Verticale]

Il sistema normalizzerà automaticamente la lunghezza del vettore alla lunghezza unitaria (1.0).

La seconda possibilità consiste nel richiedere al sistema il calcolo automatico di tale direzione. La direzione viene calcolata automaticamente in base alla direzione utensile (versore **ijk**) ed alla normale al pezzo (versore **mno**). Tale calcolo fa sì che il versore **pqd** sia normale alla direzione utensile e che sia contenuto nel piano formato dai versori **ijk** ed **mno**.



La programmazione del vettore **pqd** è significativo solamente nel momento in cui si vuol far “tallonare” l'utensile.

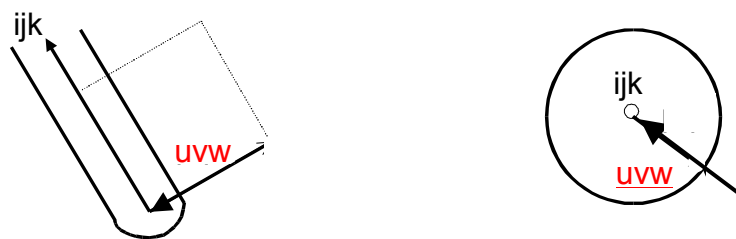
Fattori di Compensazione Parassiale: UVW

Costituiscono una metodologia alternativa ai versori mno e pqd per il calcolo della compensazione raggio utensile: compensazione che, anziché agire in direzione perpendicolare al profilo programmato (mno), è ottenuta spostando il punto programmato lungo la direzione uvw di una quantità pari al raggio. Il vettore uvw può essere di modulo anche non unitario.

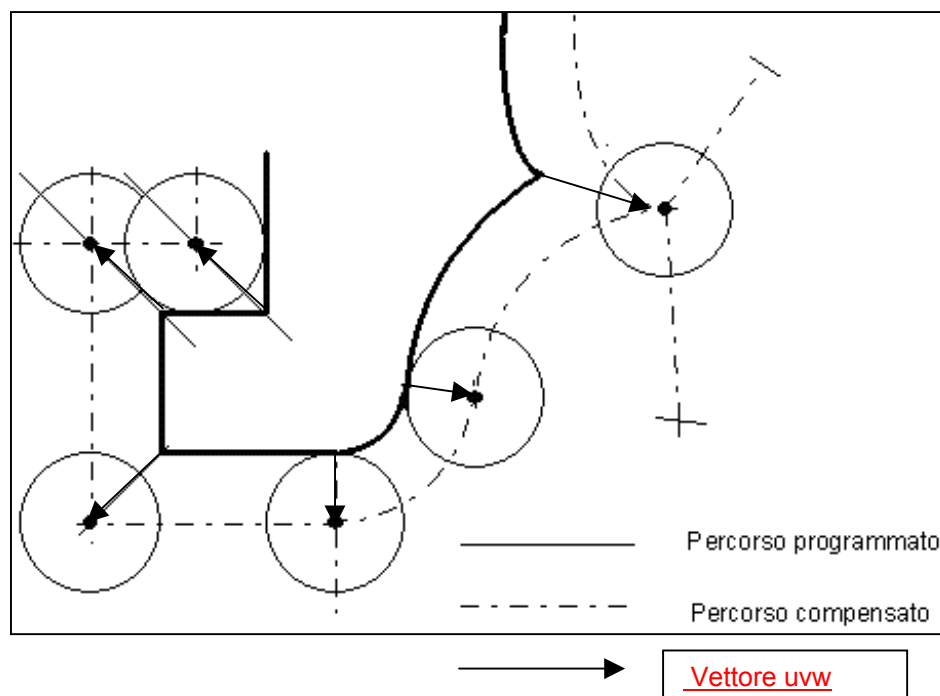
Per definire i fattori di compensazione parassiale si potranno adottare due tecniche. La prima programmando direttamente il vettore che individua tale direzione, espresso tramite le coordinate **uvw** nel formato

[u] [Componente Ascissa] [v][Componente Ordinata] [w][Componente Verticale]

La seconda possibilità consiste nel richiedere al sistema il calcolo automatico di tale direzione. La direzione viene calcolata in modo analogo a quello per il versore mno ma l'angolo α può assumere solo i seguenti valori: $+90^\circ$, se l'utensile tocca il pezzo con il lato sinistro; -90° se tocca con il lato destro. Per tale motivo la gestione dei vettori uvw è compatibile con utensili a candela o con frese sferiche ma che lavorano sul fianco. Non sono ammessi utensili toroidali. Inoltre il vettore uvw differisce da mno negli spigoli nei quali la lunghezza del vettore viene moltiplicata per un fattore di correzione opportuno.

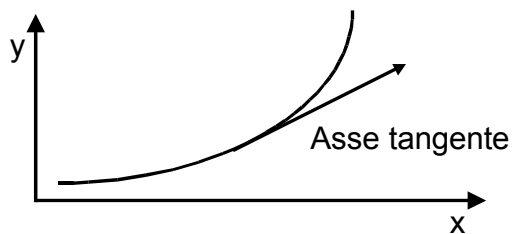


La differenza sostanziale con i versori mno, infatti, si riscontra sui punti “spigolosi” del profilo, come evidenziato nella figura successiva.



Asse Tangente

L'asse tangente è un asse la cui posizione viene calcolata in modo da rimanere tangente al profilo descritto. Il suo calcolo viene fatto in base alla tangente alla curva polinomiale nel piano di lavoro.



E' possibile definire un valore iniziale dell'asse tangente (primo punto programmato) e calcolare le successive posizioni a partire da quest'ultimo valore.

PRESTAZIONI FORNITE CON HIGH SPEED MACHINING

In base alla tipologia della macchina utensile, dei punti programmati e ad una serie di parametri aggiuntivi, tramite la funzionalità “High Speed Machining” si potranno realizzare diverse prestazioni.

Le macchine a 3 assi o generiche sono caratterizzate dal fatto di non avere assi che permettono di orientare l'utensile durante la fase di lavorazione. La direzione utensile è generalmente fissa.

Compensazione Raggio e Lunghezza utensile

Per poter effettuare la compensazione Raggio e Lunghezza utensile occorrerà fornire nel part program la tipologia di quote assi e vettori come definito nella successiva tabella. La velocità di lavorazione potrà essere riferita al punto di contatto utensile/pezzo (velocità di asportazione truciolo quindi sul Cutter Contact Point) oppure al centro utensile (sul Cutter Location Point). La differenza tra questi due tipi di parametrizzazione è significativa nel momento in cui si utilizzano utensili di grosso diametro, quindi ove il percorso del centro utensile è significativamente diverso dal profilo del pezzo.

Programmazione	Descrizione
Quote assi	I punti devono essere espressi in “ coordinate pezzo ” (Cutter Contact Points)
Direzione Utensile	Il vettore direzione utensile viene definito in fase di configurazione (su una macchina 3 assi è fisso e definito dal parametro TOD) e rimane invariato durante tutta la lavorazione. La programmazione di ijk viene ignorata.
Normale Superficie	E' necessaria la programmazione del vettore mno od in alternativa richiederne il calcolo automatico
Applicazione Raggio	E' necessaria la programmazione del vettore pqd od in alternativa richiederne il calcolo automatico

Esempio:

```
N001 G61
N002 G1 X10Y10Z10 m0n0o1 p0q0d1 F10000
N002 X20Y10Z10 ....
.....
N100 G60
```

Compensazione Lunghezza utensile

Per poter effettuare la compensazione Lunghezza utensile occorrerà fornire nel part program la tipologia di quote assi e vettori come definito nella successiva tabella. La velocità di lavorazione sarà riferita al centro utensile (sul Cutter Location Point). E' comunque possibile, partendo dai punti riferiti al centro utensile, ricalcolare i punti pezzo (Cutter Contact Point) per poter applicare la compensazione diametro utensile (si vedano i parametri di configurazione TOL) e parametrizzare la velocità di lavorazione relativamente profilo effettivamente lavorato (sul Cutter Contact Point).

Programmazione	Descrizione
Quote assi	I punti devono essere espressi in “ coordinate utensile ” (Cutter Location Points)
Direzione Utensile	Il vettore direzione utensile viene definito in fase di configurazione (su una macchina 3 assi è fisso e definito dal parametro TOD) e rimane invariato durante tutta la lavorazione. La programmazione di ijk viene ignorata.
Normale Superficie	E' necessaria la programmazione del vettore mno nel momento in cui si intende applicare la compensazione diametro utensile.
Applicazione Raggio	E' necessaria la programmazione del vettore pqr (od al più richiederne il calcolo automatico) nel momento in cui si intende applicare la compensazione diametro utensile.

Esempio:

```
N001 G61
N002 G1 X10Y10Z10 F10000
N002 X20Y10Z10 ....
.....
N100 G60
```

Nessuna compensazione utensile

In questo caso occorrerà fornire nel part program solamente le quote di posizionamento degli assi come definito nella successiva tabella. La velocità di lavorazione sarà essere riferita al movimento effettivo degli assi (Axis Location Point). E' comunque possibile, partendo da punti riferiti alle quote pezzo, ricalcolare i punti centro utensile (Cutter Location Point) per applicare la compensazione lunghezza utensile ed eventualmente ricalcolare anche i punti pezzo (Cutter Contact Point) per applicare anche la compensazione diametro utensile (si vedano i parametri di configurazione TOL). Sarà quindi possibile parametrizzare la velocità di lavorazione rispetto al moto del centro utensile (sul Cutter Location Point nel caso di compensazione lunghezza) oppure rispetto al profilo lavorato (sul Cutter Contact Point).

Programmazione	Descrizione
Quote assi	I punti devono essere espressi in “ coordinate macchina ” (Axis Location Points)
Direzione Utensile	Il vettore direzione utensile viene definito in fase di configurazione (su una macchina 3 assi è fisso e definito dal parametro TOD) e rimane invariato durante tutta la lavorazione, viene utilizzato solamente se è stata richiesta la parametrizzazione della velocità di avanzamento rispetto al centro utensile. Nel caso opposto viene ignorata. La programmazione di ijk viene ignorata.
Normale Superficie	E' necessaria la programmazione del vettore mno nel momento in cui si intende applicare la compensazione diametro utensile.
Applicazione Raggio	E' necessaria la programmazione del vettore pqd (od al più richiederne il calcolo automatico) nel momento in cui si intende applicare la compensazione diametro utensile.

Esempio

```

N001 G61
N002 G1 X10Y10Z10 F10000
N002 X20Y10Z10 ....
.....
N100 G60

```

Gestione asse tangente

Nel momento in cui viene richiesta la gestione dell'asse tangente la sua posizione viene calcolata automaticamente dal sistema. Eventuali programmazioni dell'asse tangente all'interno del part program definiscono ulteriori rotazioni dell'asse rispetto alla tangente calcolata dal sistema. La programmazione del valore 0 (zero) sull'asse attiva esclusivamente il calcolo effettuato dagli algoritmi di High Speed.

CONFIGURAZIONE

Un apposito file di setup contiene la configurazione dell'ambiente HSM: tutti i parametri dimensionali dovranno essere espressi nell'unità di misura della macchina utensile.

Tale configurazione viene attivata ad ogni programmazione del codice G61.

Tutti i valori numerici del part program potranno essere definiti direttamente oppure tramite parametri **E** o **L**.

Il file di configurazione può essere idealmente suddiviso in quattro sezioni:

- una sezione di *Triletterali Generali*
- una sezione di *Triletterali di Configurazione degli Assi*
- una sezione di *Triletterali di Configurazione della Macchina* o di definizione della cinematica di macchina.
- una sezione di *Triletterali di Configurazione del PathOptimizer*.

Triletterali Generali

(PNT ,tipo,param,poly)	Definizione tipologia punti
(VER ,ijk,mno,pqd,uvw)	Programmazione dei Versori
(JRK ,abilita,jrs,enti, TsTab, TinTgr, TTop)	Gestione Look Ahead
(THR ,toll,tollV,scala,movNullo,corda,ErrRot,MaxLen)	Programmazione delle soglie
(CRV ,Lun,Rapp,Mode)	Gestione cambio curvatura
(EDG ,Angolo,AngoloV,Modo,Acce)	Gestione degli spigoli
(TOL ,lung,raggio,R.spigolo,angolo,Lorig,Rorig,RgOrig)	Definizione Utensile
(TOD ,Xcomp,Ycomp,Zcomp)	Direzione utensile (3D)

Triletterali di Configurazione Assi

(AXI ,nome,id,tip,tipCin)	Definizione Tipo asse
(AXP ,nome,mov.nul,passo,lop-,lop+)	Definizione parametri assi
(DIN ,Vmax,Amax,Jmax,Vrap,Arap,Jrap)	Definizione dinamica degli assi

Triletterali di Configurazione della Macchina

(MAC ,Tipo,Bed,Um)	Definizione tipo macchina
(CIN ,Nome,Xoff,Yoff,Zoff,Xrot,Yrot,Zrot)	Definizione cinematica di macchina

Triletterali di Configurazione del PathOptimizer

(SMT ,Toll,TollX,TollV,Racc,RaccX,RaccV,C0,C0X,C0V)	Tolleranze e parametri smoothing
(SMS ,Split,SplitX,SplitV,Cond,CondX,CondV,Len,Pnt,C1,Progr,Rcnc)	Parametri speciali smoothing

Tipo dei punti descritti nel part program

La sintassi del triletterale che definisce il tipo dei punti descritti nel part program è la seguente:

(PNT, Tipo, Param, Poly, Formato)

esempio: (PNT, CLP, CLP, QUI, PNT)

Parametro	Tipo	Valori	Descrizione
Tipo	Carattere	CCP CLP AXI Obbligatorio	Definisce la tipologia dei punti descritti nel Part Program CCP Cutter Contact Points : I punti in input sono definiti in "Coordinate pezzo", quindi su tali punti è possibile eseguire le compensazioni utensile (raggio e lunghezza). CLP Cutter Location Points : I punti in input sono definiti in "Coordinate utensile", quindi su tali punti è possibile eseguire le compensazioni utensile (solo lunghezza). AXI Axis Location Points : i punti in input sono definiti in "Coordinate macchina", quindi su tali punti non è possibile effettuare alcun tipo di compensazione.
Param	Carattere	CCP CLP AXI Obbligatorio	Definisce il tipo di profilo per cui è programmata la velocità (Feed) nel Part Program e dipende dalla tipologia dei punti in input. La velocità è comunque sempre riferita al profilo in "3D". Gli eventuali assi rotativi o aggiuntivi non vengono coinvolti nel calcolo della velocità ma "inseguono" l'esecuzione del profilo in 3D. La velocità viene comunque limitata quando questi ultimi assi, nell'inseguire gli assi 3D, tendono a superare i propri limiti dinamici. CCP La Feed programmata è riferita al profilo generato sulle "Coordinate Pezzo"; chiaramente tale tipo di parametrizzazione è valida solo se i punti in input sono del tipo CCP. CLP La Feed programmata è riferita al profilo generato sulle "Coordinate Utensile"; è utilizzabile per i punti di tipo CCP e CLP, mentre per i punti di tipo AXI è valida solo nel momento in cui è possibile determinare la direzione utensile (o dalla programmazione degli assi rotativi, oppure dalla programmazione dei versori ijk). AXI La Feed programmata è riferita al profilo generato sulle "Coordinate Macchina"; è utilizzabile solamente per i punti di tipo AXI.
Poly	Car.	CUB o QUI Obbligatorio	Definisce il grado della curva polinomiale generata per la esecuzione del profilo programmato. CUB La polinomiale generata è una cubica. QUI La polinomiale generata è un quintica.
Formato	Carattere	PNT, POL o BSP Facoltativo	Definisce la metodologia di programmazione del profilo, ovvero il formato di input utilizzato. PNT Input in formato punti POL Input in formato polinomiale BSP Input in formato B-Spline

Modalità di gestione dei versori

I versori, se programmati, devono essere riferiti rispettivamente agli assi: ascissa, ordinata e verticale. La sintassi del triletterale che definisce come gestire tali versori è la seguente:

(VER, Ijk, Mno, Pqd, Uvw)

esempio: (VER , REL , PRG , PRG)

(VER , ROT, , REL , PRG)

Parametro	Tipo	Valori	Descrizione
Ijk	Carattere	PRG o REL Obbligatorio	<p>Definisce se il vettore “Direzione Utensile” è programmato direttamente tramite le componenti ijk oppure se deve essere calcolato in base alla posizione degli assi rotativi. Nel momento in cui il versore non è necessario, tale programmazione viene ignorata.</p> <p>PRG Il versore viene programmato tramite le componenti ijk.</p> <p>REL Il versore deve essere determinato dalla posizione programmata degli assi rotativi.</p> <p>ROT Il versore viene determinato dalla posizione istantanea degli assi rotativi</p>
Mno	Carattere	PRG o REL Facoltativo	<p>Definisce se il vettore “Normale alla Superficie” è programmato direttamente tramite le componenti mno oppure se deve essere calcolato automaticamente dal sistema. Nel momento in cui il versore non è necessario, tale programmazione viene ignorata.</p> <p>PRG Il versore viene programmato tramite le componenti mno.</p> <p>REL Il versore deve essere determinato automaticamente dal sistema.</p> <p>Se omesso, significa che si lavora con vettore uvw</p>
Pqd	Carattere	PRG o REL Obbligatorio	<p>Definisce se il vettore “Direzione Applicazione Raggio Utensile” è programmato direttamente tramite le componenti pqd oppure se deve essere calcolato automaticamente dal sistema. Nel momento in cui il versore non è necessario, tale programmazione viene ignorata.</p> <p>PRG Il versore viene programmato tramite le componenti pqd.</p> <p>REL Il versore deve essere determinato automaticamente dal sistema.</p>

Uvw	Carattere	PRG o REL Facoltativo	<p>Definisce se il “Fattore di Compensazione Raggio Utensile” è programmato direttamente tramite le componenti uvw oppure se deve essere calcolato automaticamente dal sistema. Nel momento in cui il versore non è necessario, tale programmazione viene ignorata.</p> <p>PRG Il vettore viene programmato tramite le componenti uvw.</p> <p>REL Il vettore deve essere determinato automaticamente dal sistema.</p> <p>Se omesso, significa che si lavora con versore mno</p>
------------	-----------	--------------------------	---

NOTA:

Se è richiesta una compensazione raggio utensile I vettori MNO e UVW non possono essere omessi entrambi.

Gestione Look Ahead

La sintassi del triletterale che definisce la modalità di gestione della **Look Ahead** e della dinamica è la seguente:

(JRK, Modo, Jrs, Enti, Tstab, Tintgr, Ttop)

esempio: (JRK , ENA , ,)

Parametro	Tipo	Valori	Descrizione
Modo	Carattere	ENA o DIS o AXI Obbligatorio	Definisce il tipo di dinamica da usare. ENA Abilita l'utilizzo della Jerk Limitation (MOV=8). Il Jerk specificato è riferito alla dinamica descritta sul profilo. DIS Disabilita la Jerk Limitation, usa solo "rampe a S" (MOV=2). AXI Abilita l'utilizzo della Jerk Limitation (MOV=8). Il Jerk specificato è riferito alla dinamica descritta dagli assi e comunque questa si ripercuote sulla dinamica del profilo. L'utilizzo di tale modalità è fortemente legata alla qualità dei punti in input (numero di decimali e distribuzione)
Jrs	Numero	Prelevato da variabile JRS di sistema Facoltativo	Modifica il valore della variabile JRS del sistema solamente per la sezione G61 in fase di elaborazione; se il valore viene omissso si utilizza la JRS attiva sul sistema.
Enti	Numero	Facoltativo	L'algoritmo di High Speed Machining si appoggia ad una coda dinamica di 64 enti (curve polinomiali) e solo dopo aver calcolato il 64° da inizio alla movimentazione. È possibile anticipare la partenza programmando il numero di enti da calcolare prima di avviare la movimentazione. 0 significa: massimo numero di enti.
Tstab	Numero	Facoltativo	Definisce la finestra temporale (in ms) entro la quale applicare l'algoritmo di smoothing delle velocità. Tale algoritmo rimuovi inutili fasi di accelerazioni onde evitare oscillazioni di macchina. (Vedere figura 1)
Tintgr	Numero	Facoltativo	Definisce il tempo (in ms) massimo di integrazione delle fasi di accelerazione (o decelerazione) con i tratti a velocità costante. (Vedere figura 1)

Parametro	Tipo	Valori	Descrizione
Ttop	Numero	Facoltativo	Definisce il tempo minimo (in ms) di esecuzione di un tratto a V costante. (Vedere figura1)

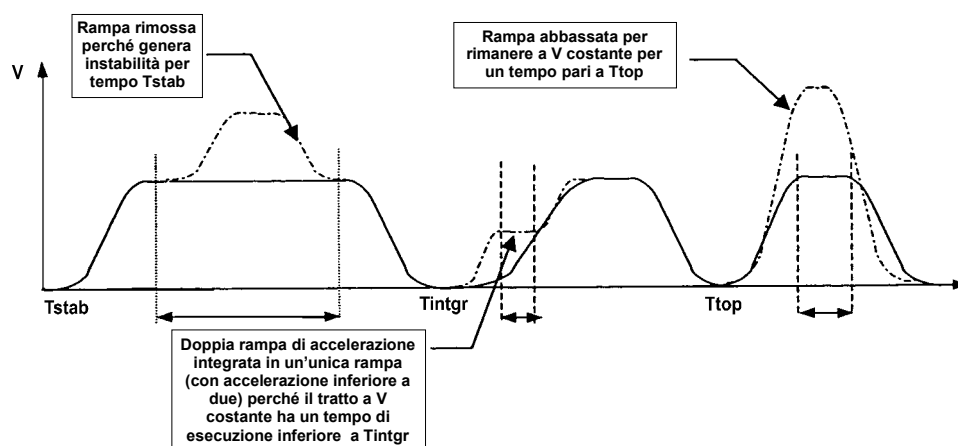


Figura 1

Soglie

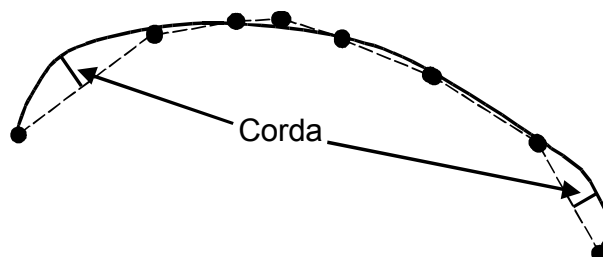
La sintassi del triletterale che definisce il valore di alcune soglie utilizzate nella generazione delle curve polinomiali è la seguente:

(THR, Toll, TolIV, Scala, MovNullo, Corda, ErrRot, MaxLen)

esempio: (THR , 0.01 , 0.0001 , 0 , 0 , 0.1)

Parametro	Tipo	Valori	Descrizione
Toll	Numero	Obbligatorio	Definisce la banda di tolleranza da utilizzare nella generazione delle curve polinomiali. Come definito in precedenza le curve polinomiali generate passano, a meno di una certa soglia di tolleranza, per i punti programmati. Tale parametro definisce appunto questo valore di tolleranza. È definito in mm (o inch se la macchina è configurata in inch).
TolIV	Numero	Obbligatorio	Definisce la banda di tolleranza da utilizzare nella generazione delle curve polinomiali calcolate sui versori. Tale valore è importante poiché dalla precisione con cui vengono generate le curve sui versori (in particolar modo per il versore ijk) deriva la precisione di posizionamento degli assi rotativi. Il numero è adimensionale poiché riferito ad entità come i versori che hanno dimensione sempre di 1. Si suggerisce di impostare un valore pari a 0.1 volte il valore Toll definito prima. 0 disabilita la gestione della tolleranza sui versori.
Scala	Numero	Obbligatorio	Fattore di scala da applicare sugli assi programmati. 0 significa non applicare il fattore di scala. Non viene applicato ai versori ed agli assi rotativi, quando programmati.
MovNullo	Numero	Obbligatorio	Soglia di movimento nullo. Se la distanza tra un punto ed il successivo è inferiore a tale soglia (in mm o inch) il punto successivo viene eliminato.
Corda	Numero	Obbligatorio	Errore cordale massimo ammesso in ingresso ed in uscita durante la generazione di una polinomiale. Valore espresso in mm (o inch).
ErrRot	Numero	Facoltativo	Massimo errore cordale ammesso nel calcolo delle polinomiali sugli assi rotativi. Valore espresso in gradi

MaxLen	Numero	Facoltativo	Lunghezza massima dei polinomi generati. Valore espresso in mm (o inch). Consente un maggiore controllo della tolleranza di posizione. (Toll) Valori consigliati >5 mm, valori inferiori potrebbero rallentare la lavorazione.
--------	--------	-------------	---



Definizione Utensile

Nel sistema High Speed Machining è possibile gestire utensili di tipo “Candela”, “Sferico” e “Toroidale”. La sintassi del triletterale che definisce le caratteristiche dell’utensile con cui lavorare è la seguente:

(TOL, Lung, Raggio, RaggSpig, Angolo, Lorig, Rorig, RgOrig) *esempio: (TOL , , , 1 , -90 , 0)*

Parametro	Tipo	Valori	Descrizione
Lung	Numero	Prelevato dal correttore attivo su G61 Facoltativo	Definisce la lunghezza dell’utensile da usare per la compensazione lunghezza utensile. Se non viene impostato alcun valore, la lunghezza utensile viene prelevata dai correttori attivi nel sistema sulla attivazione della G61. Il valore qui impostato è attivo solamente durante la lavorazione della corrente sezione di G61/G60.
Raggio	Numero	Prelevato dal correttore attivo su G61 Facoltativo su CN Obbligatorio su Path Opti.	Definisce il raggio dello spigolo dell’utensile da usare per la compensazione raggio utensile. Se non viene impostato alcun valore, il raggio utensile viene prelevato dai correttori attivi nel sistema sulla attivazione della G61. Il valore qui impostato è attivo solamente durante la lavorazione della corrente sezione di G61/G60.
RaggSpig	Numero	Prelevato dal correttore attivo su G61 Facoltativo su CN Obbligatorio su Path Opti.	Definisce il raggio dello spigolo dell’utensile. Se non viene impostato alcun valore , il raggio spigolo viene prelevato dai correttori attivi nel sistema sull’attivazione del G61. Vale la regola di utilizzo del parametro TTR ; se attivo usa come raggio del toroidale la lunghezza 2 , se disattivo forza a 0.
Angolo	Numero	Facoltativo	Definisce l’angolo di contatto tra utensile e pezzo. Viene utilizzato nel calcolo automatico del vettore normale alla superficie.
LOrig	Numero	Facoltativo	Definisce la lunghezza dell’utensile per cui è stato generato il part program. Tale campo viene utilizzato quando i punti in input sono riferiti a “Coordinate Macchina” e si vuole applicare la compensazione lunghezza utensile”.
ROrig	Numero	Facoltativo	Definisce il raggio dell’utensile per cui è stato generato il part program. Tale campo viene utilizzato quando i punti in input sono riferiti a “Coordinate Macchina” e si vuole applicare la compensazione lunghezza e diametro utensile”.
RgOrig	Numero	Facoltativo	Definisce il raggio dell’utensile per cui è stato generato il part program. Tale campo viene utilizzato quando i punti in input sono riferiti a “Coordinate Macchina” e si vuole applicare la compensazione lunghezza e diametro utensile”.

Direzione Utensile (3D)

Definisce la direzione dell'utensile (da usare per le compensazioni) per macchine a 3 assi o generiche (per le macchine a 5 assi non serve). In pratica occorre definire il vettore unitario (analogo ad *ijk*) che individua la direzione dell'utensile nel sistema di riferimento del pezzo.

La sintassi del triletterale che definisce la direzione dell'utensile con cui lavorare è la seguente:

(TOD, Xcomp, Ycomp, Zcomp)

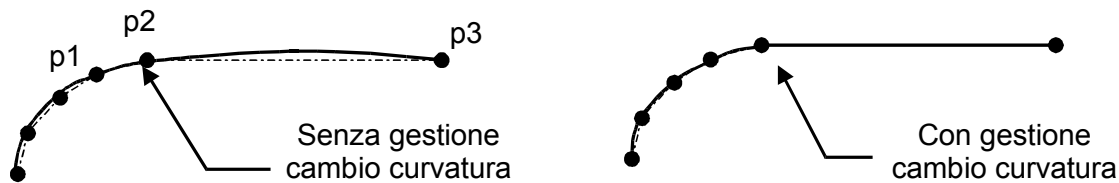
esempio: (TOD , 0 , 0 , 1)

Parametro	Tipo	Valori	Descrizione
Xcomp	Numero	Obbligatorio	Componente della direzione utensile lungo l'asse ascissa.
Ycomp	Numero	Obbligatorio	Componente della direzione utensile lungo l'asse ordinata.
Zcomp	Numero	Obbligatorio	Componente della direzione utensile lungo l'asse verticale.

Gestione Cambio di Curvatura

L'algoritmo utilizzato dai sistemi CAM per generare i punti di un profilo tiene conto dell' "errore cordale", ovvero genera punti più fitti quando più accentuato è il raggio di curvatura del profilo da descrivere. Un cambio di curvatura, quindi, può essere genericamente individuato con una variazione della distanza tra una serie di punti ed i successivi. Questa variazione di curvatura può essere individuata automaticamente dagli algoritmi di High Speed Machining in modo tale da evitare possibili oscillazioni nell'intorno del punto in cui avviene tale variazione.

Supponiamo, ad esempio, di avere un tratto curvilineo seguito da un tratto rettilineo. La spline sul tratto rettilineo tenderebbe ad oscillare od a generare una "gobba", quindi è importante individuare questo fenomeno.



Nel punto di "cambio di curvatura" il sistema introdurrà automaticamente una G62 od una G63 a seconda di quanto richiesto dall'utente. In alternativa potrà inserire dei punti (infittimento) all'interno del tratto più lungo; il numero di punti inseriti dipende dal rapporto tra le lunghezze dei due tratti (lungo/corto).

La sintassi del triletterale che definisce come identificare e quindi gestire il cambio di curvatura è la seguente:

(CRV, Lun, Rapp, Mode)

esempio: (CRV , 1 , 6 , G63)

Parametro	Tipo	Valori	Descrizione
Lun	Numero	Obbligatorio	Definisce la lunghezza minima che deve assumere il "tratto lungo" al fine di poter gestire la variazione di curvatura. È definito in mm (o inch se la macchina è configurata in inch).
Rapp	Numero	Obbligatorio	Definisce il rapporto tra il tratto lungo ed il tratto corto affinché venga individuata la variazione di curvatura. Ad esempio il valore 6 definisce che la distanza tra p2 e p3 deve essere superiore di 6 volte la distanza tra p1 e p2 affinché scatti il cambio di curvatura.
Mode	Carattere	G62 o G63 o PNT Obbligatorio	Definisce il codice di transizione da porre sul punto di "cambio curvatura" G62 I due segmenti sono generati con due splines tangenti tra di loro, quindi la seconda spline avrà inclinazione iniziale uguale all'inclinazione finale della prima spline. G63 I due segmenti sono generati da due splines non tangenti, ma vengono raccordate in base alla tolleranza di calcolo impostata. PNT Viene eseguito un infittimento introducendo dei punti intermedi sul tratto più lungo.

Gestione Spigoli

Lo stesso motivo definito in precedenza per l'individuazione del cambio di curvatura, è valido per l'identificazione automatica di spigoli. La mancata individuazione di spigoli potrebbe generare delle errate oscillazioni sulle splines. L'approccio dinamico ottimale per affrontare uno spigolo è fermarsi a velocità 0 e poi ripartire sul tratto successivo. Talvolta il fatto di fermarsi può essere dannoso in quanto l'utensile, ruotando, continua ad asportare materiale e quindi si potrebbero vedere delle "tacche" sul pezzo. In tal senso si potrà definire se e come fermarsi sullo spigolo.

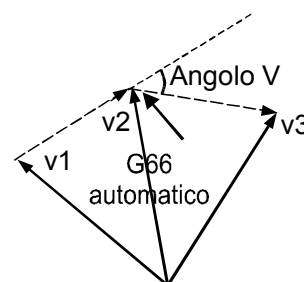
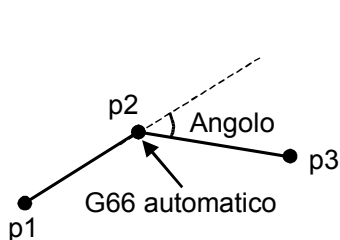


La sintassi del triletterale che definisce come identificare e quindi come gestire la presenza di spigoli è la seguente:

(EDG, Angolo, AngoloV, Modo, Acce)

esempio: (EDG , 30 , 0 , G66 , 1)

Parametro	Tipo	Valori	Descrizione
Angolo	Numero	Obbligatorio	Il sistema è in grado di determinare automaticamente gli spigoli definiti dai punti programmati (spigoli sugli assi lineari) e di generare la programmazione automatica di una G66. Tale valore, espresso in gradi, definisce l'angolo di soglia, oltre il quale un punto viene definito "di spigolo". Vedi figura seguente.
AngoloV	Numero	Obbligatorio	È analogo al valore precedente da utilizzarsi per i versori. La presenza di uno spigolo sui versori genera uno "spigolo" nella movimentazione degli assi rotativi, quindi è bene introdurre anche in tali casi una G66. Vedi figura seguente.
Modo	Carattere	G63, G66 o G67 Obbligatorio	Definisce il codice di transizione da porre sullo spigolo G63 Lo spigolo viene seguito inserendo un raccordo che viene generato tenendo conto dell'errore cordale configurato. G66 Il movimento termina a velocità nulla G67 Il movimento termina alla massima velocità possibile per affrontare nel modo ottimale lo spigolo
Acce	Numero	Obbligatorio	Quantità di accelerazione sopportabile dagli assi nell'affrontare lo spigolo: 1 significa che un asse può sopportare un salto di velocità pari ad 1 accelerazione. Vale solo nel caso di G67.



Definizione assi

È possibile definire gli assi da sottoporre agli algoritmi di High Speed Machining. Possono essere definiti un massimo di 6 assi di cui: 3 assi costituiscono la terna cartesiana, eventuali 2 assi rotativi (per le macchine a 5 assi) ed altri assi aggiuntivi. La sintassi del triletterale di definizione degli assi è la seguente:

(AXI, Nome, Id, Tipo, TipoCin, Diam, LinRot)

esempio: (AXI, X, 1, ABS, L1, DIS)

esempio: (AXI, C, 4, OTH, OTH, LIN)

Parametro	Tipo	Valori	Descrizione
Nome	Carattere	Obbligatorio	Definisce il nome dell'asse. Tale nome può essere diverso da quello configurato in AMP per l'asse, quindi è paragonabile ad una "DIN" implicita nella configurazione dell'HSM. Tale associazione vale solamente per la sezione G61/G60 in fase di lavorazione.
ID	Numero	Obbligatorio	Definisce l'ID dell'asse. Tale asse deve essere sotto controllo del Processo in cui si sta eseguendo il part program.
Tipo	Carattere	ABS, ORD, VRT, OTH, TAN o TAO Obbligatorio	Definisce il tipo asse, ed influenza il modo in cui l'asse viene trattato dagli algoritmi di HSM. ABS L'asse è di tipo Ascissa (nel sistema di orientamento cartesiano) ORD L'asse è di tipo Ordinata (nel sistema di orientamento cartesiano) VRT L'asse è di tipo Verticale (nel sistema di orientamento cartesiano) TAN Asse tangente, calcolato automaticamente dal sistema TAO Asse tangente, calcolato automaticamente dal sistema, con asse che si muove (direzione positiva) in senso orario, quindi in senso contrario alla convenzione trigonometrica OTH Altro tipo di asse, diverso dai precedenti.

Parametro	Tipo	Valori	Descrizione
TipoCin	Carattere	LI1, LI2, LI3, WRK, TOL o OTH Obbligatorio	<p>Definisce la posizione dell'asse nella catena cinematica della macchina (vedere prossimi paragrafi). Per macchine 3 assi o generiche di cui non è necessario definire la catena cinematica consigliamo la seguente configurazione:</p> <p>LI1 Da associare all'asse definito di tipo ABS (Ascissa) nel precedente campo.</p> <p>LI2 Da associare all'asse definito di tipo ORD (Ordinata) nel precedente campo.</p> <p>LI3 Da associare all'asse definito di tipo VRT (Verticale) nel precedente campo.</p> <p>OTH Asse aggiuntivo.</p> <p>Per le macchine a 5 assi i valori di configurazione sono i seguenti:</p> <p>LI1 Primo asse della catena cinematica.</p> <p>LI2 Secondo asse della catena cinematica.</p> <p>LI3 Terzo asse della catena cinematica.</p> <p>WRK Asse rotativo più vicino al pezzo.</p> <p>TOL Asse rotativo più vicino all'utensile.</p> <p>OTH Asse aggiuntivo.</p>
Diam	Carattere	ENA o DIS Facoltativo	<p>Consente di caratterizzare l'asse come Diametrico. Se non viene definito, viene utilizzato l'analogo parametro attivo nel sistema.</p> <p>ENA Asse definito come diametrico.</p> <p>DIS Asse normale.</p>
LinRot	Carattere	LIN o ROT Facoltativo su CN Obbligatorio su Path Optimizer	<p>Consente di caratterizzare l'eventuale asse aggiuntivo come lineare o rotativo. Se non impostato, per esecuzione sul CN, viene utilizzato l'analogo parametro attivo nel sistema; per esecuzione sul Path Optimizer, viene segnalato errore.</p> <p>LIN Asse aggiuntivo di tipo lineare.</p> <p>ROT Asse aggiuntivo di tipo rotativo.</p>

Parametri Assi

È possibile variare alcune caratteristiche dell'asse impostate nel sistema. Tali variazioni sono attive solamente nella sezione G61/G60 in fase di lavorazione. La sintassi del triletterale è la seguente:

(AXP, Nome, MovNul, Passo, Lop-, Lop+)

esempio: (AXP, X, 0, , ,)

Parametro	Tipo	Valori	Descrizione
Nome	Carattere	Obbligatorio	Nome dell'asse così come definito nel triletterale (AXI).
MovNul	Numero	Obbligatorio	Definisce il movimento nullo per l'asse in questione. Se la quota programmata per un asse differisce dalla posizione attuale (od ultima programmata) per un valore inferiore a tale movimento nullo, lo spostamento dell'asse (la nuova quota) non viene considerato. Si consiglia di lasciarlo sempre a 0 e di attivarlo solo in casi in cui il part program in lavorazione contenga delle imprecisioni sulla programmazione degli assi.
Passo	Numero	Prelevato dal sistema Facoltativo	Consente ridefinire il passo rollover (in gradi) per l'asse in questione. Può essere usato, ad esempio, per programmare un asse rotativo non rollover con quote rollover e viceversa.
Lop-	Numero	Prelevato dal sistema Facoltativo	Consente ridefinire il limite operativo inferiore per l'asse in questione.
Lop+	Numero	Prelevato dal sistema Facoltativo	Consente ridefinire il limite operativo superiore per l'asse in questione.

Dinamica Assi

È possibile variare alcune caratteristiche dinamiche dell'asse impostate nel sistema. Tali variazioni sono attive solamente nella sezione G61/G60 in fase di lavorazione. La sintassi del triletterale è la seguente:

(DIN, Nome, Vmax, Amax, Jmax, Vrap, Arap, Jrap)

esempio: (DIN, X, , , , ,)

Parametro	Tipo	Valori	Descrizione
Nome	Carattere	Obbligatorio	Nome dell'asse così come definito nel triletterale (AXI).
Vmax	Numero	Prelevato dal sistema Facoltativo	Consente di ridefinire la velocità massima a cui poter muovere l'asse quando è attiva una lavorazione in G01. Deve essere definita in mm/min. Se non viene definito, viene utilizzato l'analogo parametro attivo nel sistema
Amax	Numero	Prelevato dal sistema Facoltativo	Consente di ridefinire la accelerazione massima a cui poter muovere l'asse quando è attiva una lavorazione in G01. Deve essere definita in mm/sec ² . Se non viene definito, viene utilizzato l'analogo parametro attivo nel sistema
Jmax	Numero	Prelevato dal sistema Facoltativo	Consente di ridefinire il Jerk massimo a cui poter muovere l'asse quando è attiva una lavorazione in G01. Deve essere definita in mm/sec ³ . Se non viene definito, viene utilizzato l'analogo parametro attivo nel sistema
Vrap	Numero	Prelevato dal sistema Facoltativo	Consente di ridefinire la velocità massima a cui poter muovere l'asse quando è attiva una lavorazione in G00. Deve essere definita in mm/min. Se non viene definito, viene utilizzato l'analogo parametro attivo nel sistema
Arap	Numero	Prelevato dal sistema Facoltativo	Consente di ridefinire la accelerazione massima a cui poter muovere l'asse quando è attiva una lavorazione in G00. Deve essere definita in mm/sec ² . Se non viene definito, viene utilizzato l'analogo parametro attivo nel sistema
Jrap	Numero	Prelevato dal sistema Facoltativo	Consente di ridefinire il Jerk massimo a cui poter muovere l'asse quando è attiva una lavorazione in G00. Deve essere definita in mm/sec ³ . Se non viene definito, viene utilizzato l'analogo parametro attivo nel sistema

Esempio

Configurazione per macchina a 3 assi:

```
(PNT , AXI , AXI , QUI )
(VER , REL , PRG , PRG )
(JRK, ENA , , 400 , 150 ,200 )
(THR , 0.01 , 0.0001 , 0 , 0.001 , 0.05 )
(TOL , , , 0 , -90 , )
(TOD , 0 , 0 , 1 )
(CRV , 1 , 6 , G63 )
(EDG , 30 , 0 , G66 , 1 )
;
(AXI , X , 1 , ABS , LI1 )
(AXP, X , 0 , , , )
(DIN , X , , , , , )
;
;
;
(AXI ,Y, 2 , ORD, LI2 )
(AXP,Y, 0 , , , )
(DIN ,Y, , , , , , )
;
;
;
(AXI , Z , 3 , VRT, LI3 )
(AXP, Z , 0 , , , )
(DIN , Z , , , , , , )
;
;
```

FINE CAPITOLO

COMANDI RELATIVI ALLA GESTIONE MULTIPROCESSO

GENERALITA'

Questo capitolo descrive i comandi interprocesso.

In particolare vengono descritte le funzionalità che consentono al programmatore di sincronizzare l'esecuzione parallela di part program su processi diversi, o di utilizzare la risorsa "assi" da un processo ad un altro a seconda delle esigenze.

La seguente tabella riassume questi comandi:

TRILETTERALE	FUNZIONALITÀ
DCC	Definizione di canali remoti.
PVS	Definizione ambiente variabili PLUS remote.
PRO	Determina il processo di default al quale inviare i comandi o le sincronizzazioni.
SND	Invia un messaggio di sincronizzazione al processo specificato.
WAI	Pone il processo in cui è eseguito, in stato di WAIT; il processo interessato rimane bloccato in questo stato fino all'arrivo di un messaggio da un altro processo.
EXE	Attiva l'esecuzione automatica di un part program nel processo specificato.
ECM	Esecuzione di un blocco di MDI sul processo specificato.
GTA	Gestisce la migrazione di risorse "assi" tra processi.

IMPORTANTE

I comandi SND e WAI sono sincronizzati di default.

NOTE FUNZIONALI SULLA SINCRONIZZAZIONE TRA PROCESSI

Note sulla funzionalità "WAIT"

I messaggi di sincronizzazione che un processo riceve vengono registrati in un'area di memoria detta "CODA MESSAGGI"; quest' area è locale, ed è perciò riservata a quel processo (ogni processo ne possiede una).

Se durante l'esecuzione di un part program viene eseguito il triletterale (WAI,Pn), il processo deve porsi in attesa di un messaggio liberatorio dal processo <n>, questo messaggio viene perciò ricercato all'interno della CODA MESSAGGI:

- **Se il messaggio non è ancora presente nella CODA MESSAGGI**, il processo si pone nello stato WAIT sospendendo l'esecuzione del part program.
- **Se il messaggio è già in CODA MESSAGGI**, il processo prosegue con l'esecuzione del part program, ed il messaggio viene eliminato dalla CODA MESSAGGI.
- **Se nella coda è presente un messaggio proveniente da un processo diverso da <n>**, esso non viene eliminato, ma conservato per essere utilizzato nel momento opportuno (la programmazione di una WAI del processo appropriato).

Il processo può infatti uscire dallo stato di WAIT solo quando in coda è presente un messaggio il cui mittente <n> è il processo specificato nell'istruzione WAI.

Note sulla funzionalità "SEND"

Tramite il triletterale SND si inviano messaggi di sincronizzazione al processo specificato; tale messaggio può essere sincrono, oppure asincrono:

- **Se è sincrono**, il processo che ha eseguito la SND passerà nello stato di WAIT in attesa di un ACK (segnalazione di messaggio utilizzato) dal processo destinatario della SND.
- **Se è Asincrono**, il processo che ha eseguito la SND prosegue l'esecuzione del part program comunque; senza cioè aspettare che il messaggio abbia effettivamente fatto ripartire il processo destinatario della SND.

IMPORTANTE

Un dato processo (es. processo 1) non può inviare più di un messaggio per volta ad un'altro processo (es. processo 2) almeno fino a chè il processo 2 non sarà uscito dallo stato WAIT (ed eliminato il messaggio dalla sua coda).

Può però inviare messaggi ad altri processi (es. processo 3).

IMPORTANTE

I reset del processo o l'attivazione di un altro part program azzerano la coda messaggi del processo stesso.

Scambio dati

I comandi SND e WAI consentono lo scambio di dati tra processi, quali variabili locali o di sistema, costanti, stringhe.

Questi dati vengono inviati (SND) o ricevuti (WAI) dai processi in causa e possono essere specificati nella programmazione delle rispettive funzioni; per maggiori dettagli consultare la sintassi dei due trilettari, nelle pagine che seguono.

Reset processi sincronizzati

Se si effettua un reset, bisogna eseguirlo sia sui processi che mandano messaggi di sincronizzazione (SND), sia sui processi che hanno ricevuto i messaggi di sincronizzazione (WAI).

Questo per evitare che rimangano messaggi non desiderati nella "CODA MESSAGGI" dei processi, che potrebbero portare a dei malfunzionamenti.

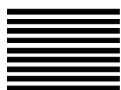
Tabella canali

La tabella canali è una tabella interna di sistema, che contiene gli identificativi dei canali logici su cui un processo può operare, intendendo come canale un ambiente remoto o locale.

Ogni processo dispone di una propria tabella, ciascuna contenente 40 canali. All'accensione il sistema definisce di default i 20 processi locali come i primi 20 canali della tabella, identificabili con i numeri 1,2,3,4, ... 20.

Se si dispone di sistemi Serie 10 collegati in rete tra di loro tramite l'opzione MINI DNC E65/E66 o superiore, è possibile definire altri canali in tabella associando gli identificativi (1÷40) agli ambienti remoti precedentemente configurati tramite l'opzione MINI DNC.

Nel caso in cui venga utilizzato come identificativo di un ambiente remoto un valore da 1 a 20, il processo locale corrispondente viene eliminato dalla tabella canali.



DCC - Definizione del canale di comunicazione

Il triletterale DCC permette al programmatore di definire i canali della tabella del processo in cui è programmato.

Sintassi

(DCC , numero canale, nome canale)

dove:

numero canale: Numero del canale da definire; può essere un numero o una variabile locale o di sistema, di valore compreso tra 1 e 40.

nome canale: Nome dell'ambiente remoto definito con il configuratore di rete dell'opzione MINI DNC.

Caratteristiche:

Il *canale* definito con questo comando deve essere precedentemente configurato tramite l'opzione MINI DNC. Tra i tipi di canale definibili dal MINI DNC, il comando DCC può operare su due tipi predefiniti.

- Tipo 1 - canale di PROCESSO:
È un processo remoto al quale possono essere inviati tutti i comandi accettati dai processi locali.

Esempi:

(DCC,25,remoto_1) ;definisce il canale n° 25

(WAI,P25) ;attende messaggio da canale (processo remoto) n°25

(PRO,25)

(EXE,PROGRAM) ;lancio part program su processo remoto n°25

- Tipo 2 - canale PLUS:
È un canale sul quale, con il comando PVS, si può redirezionare la lettura e la scrittura di variabili PLUS (SN, SC, L, @, \$ASSET).

IMPORTANTE

Il RESET del processo o l'attivazione di un part program reinizializza la tabella dei canali:

canale 1 = processo 1 locale

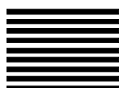
canale 2 = processo 2 locale

canale 3 = processo 3 locale

NOTA:

Ogni comando emesso deve essere congruente con il tipo di canale a cui esso è inviato.

Ad esempio, se viene inviato un comando accettato dai canali di processo (es. WAI, SND, EXE), ad un canale di tipo 2, il sistema ritorna l'errore **"NC221 TIPO PROCESSO ERRATO"**.



PVS - Selezione del canale PLUS

Il triletterale PVS permette di redirezionare la lettura e la scrittura di variabili PLUS su un canale remoto di tipo 2, precedentemente definito con il comando DCC.

Sintassi

(**PVS** ,numero canale)

dove:

numero canale: Numero del canale da definire; può essere un numero o una variabile locale o di sistema, di valore compreso tra 0 e 40.

Caratteristiche:

Dopo la programmazione del comando PVS, la lettura e la scrittura delle variabili di PLUS (SN, SC, L) e @xxxx, \$xxxx (con opzione ASSET abilitata), avviene sul canale remoto selezionato.

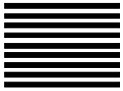
Per ripristinare la lettura delle variabili PLUS locali paragonare (PVS,0).

Se il canale specificato non è di tipo 2 (definito con il configuratore di rete), il sistema ritorna l'errore **NC221 TIPO PROCESSO ERRATO**.

Se il canale specificato non è stato precedentemente definito con DCC il sistema ritorna l'errore **NC220 PROCESSO NON DEFINITO**.

IMPORTANTE

Il RESET del processo annulla la programmazione del comando PVS (equivale alla programmazione (PVS,0)).



PRO - Definizione del processo di default

Il trilettale PRO permette al programmatore di specificare il processo di default al quale sono riferiti i comandi di sincronizzazione: il numero di processo è infatti opzionale in tutti i comandi di questo tipo.

Sintassi

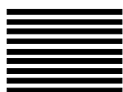
(PRO ,numero processo)

dove:

numero processo: Processo di default se non specificato nei comandi di sincronizzazione.
Può essere un numero od una variabile locale o di sistema, e deve assumere valori compresi tra 1 e 20 (1 ÷ 40 con opzione E65/E66).

IMPORTANTE

Il reset del sistema o l'attivazione di un altro part program, annullano le precedenti definizioni del processo di default.



SND - invio di un messaggio di sincronizzazione (send)

Il trilettale SND permette di inviare un messaggio di sincronizzazione al processo specificato, ed eventualmente l'invio di dati.

Sintassi

(SND,[,P numero processo], {S|A} [,dato] [,dato] [,dato])

dove:

numero di processo: processo a cui inviare il messaggio.

Può essere un numero, od una variabile locale o di sistema e deve essere compreso tra 1 e 20 (1 ÷ 40 con opzione E65/E66).

Se questo numero è uguale al numero del processo che sta eseguendo il comando SND, viene generato l'errore **"NC222 NUMERO DI PROCESSO ERRATO"**.

Qualora il *numero processo* non fosse specificato, il messaggio verrà inviato al processo di default dichiarato con il trilettale PRO.

S o A: parametro che specifica le due funzionalità del comando SND:

S - funzionamento in modo sincrono:

il processo corrente passa nello stato WAIT; questo stato Permane finché il processo destinatario non ha ricevuto effettivamente il messaggio (lo ha cioè cancellato dalla sua coda ed ha ripreso l'esecuzione del part program).

A - funzionamento in modo asincrono:

non appena il messaggio è stato inviato, l'esecuzione del part program riprende.

dati: I dati, (opzionali ed al massimo 20), sono valori numerici (da considerare Long Real), stringhe tra apici, variabili locali o di sistema.

Il sistema preleva i dati programmati (direttamente o come contenuto delle variabili) e li trasmette al processo specificato insieme al messaggio di sincronizzazione.

La dimensione totale dei dati non può superare i 174 byte; per la lunghezza dei vari tipi di dato consultare la seguente tabella:

Tipo dato:	lunghezza in byte
Boolean / Byte	2
Short	3
Long / Real	5
Double (Long Real)	9
Stringa	n° caratteri+2

Esempio:

(SND,P1,S,E1,SN12,"TOOL",SC0.30,@BOOL_PLUS(0),33.6)

Calcolo del numero dei byte trasmessi:

Variabile o costante:	Tipo:	lunghezza in byte:
E1	Long Real	9
SN12	Long Real	9
"TOOL"	Stringa	6
SC0.30	Stringa	32
@BOOL_PLUS(0)	Boolean	2
33.6	Long Real	9
Totale:		67 byte trasmessi

Il sistema genera l'errore **"NC224 INVIO DATI TROPPO LUNGO"** quando si tenta di trasmettere un numero di byte maggiore di 174.

– **Situazioni di errore:**

Vi sono due casi in cui viene generato l'errore **"NC225 CARICAMENTO DATI FALLITO"**:

• **1 - SND in modo sincrono:**

Il processo 1 invia una (SND,P2,S,E1) al processo 2, il quale si trova nello stato di WAIT (WAI,P1,SC0). Visto che il tipo della variabile trasmessa non è congruente con quella da ricevere, e la SND è stata programmata con modo sincrono, si verifica che:

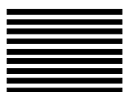
- entrambi i processi visualizzeranno l'errore **"NC225 CARICAMENTO DATI FALLITO"**, ed i rispettivi part program non potranno proseguire.

• **2 - SND in modo asincrono:**

Il processo 1 invia una (SND,P2,A,E1) al processo 2, nel cui part program è stata programmata una funzione (WAI,P1,SC0.30).

Visto che il dato trasmesso non è congruente con quello che il processo 2 si aspettava di ricevere, e che la SND è di tipo asincrono, si verifica che:

- Il processo 1 prosegue l'esecuzione del suo part program
- Il processo 2 visualizza l'errore **NC225** e l'esecuzione del suo part program non potrà proseguire



WAI - Attesa di un messaggio di sincronizzazione (wait)

Il trilettale WAI pone il processo in cui è programmato nello stato di WAIT, in attesa cioè di un messaggio di sincronizzazione da processo specificato; se il messaggio atteso è già presente nella coda messaggi si prosegue nell'esecuzione del part program ed eventualmente si risponde al task mittente se quest'ultimo aveva eseguito il comando SND di tipo S (sincrono).

Sintassi

(WAI,[P numero processo], [,variabile] [,variabile] [,variabile])

dove:

numero di processo: Processo da cui si aspetta il messaggio.

Può essere un numero, od una variabile locale o di sistema e deve essere compreso tra 1 e 20 (1 ÷ 40 con opzione E65/E66).

Se questo numero è uguale al numero del processo che sta seguendo il comando WAI, viene generato l'errore **"NC222 NUMERO DI PROCESSO ERRATO"**.

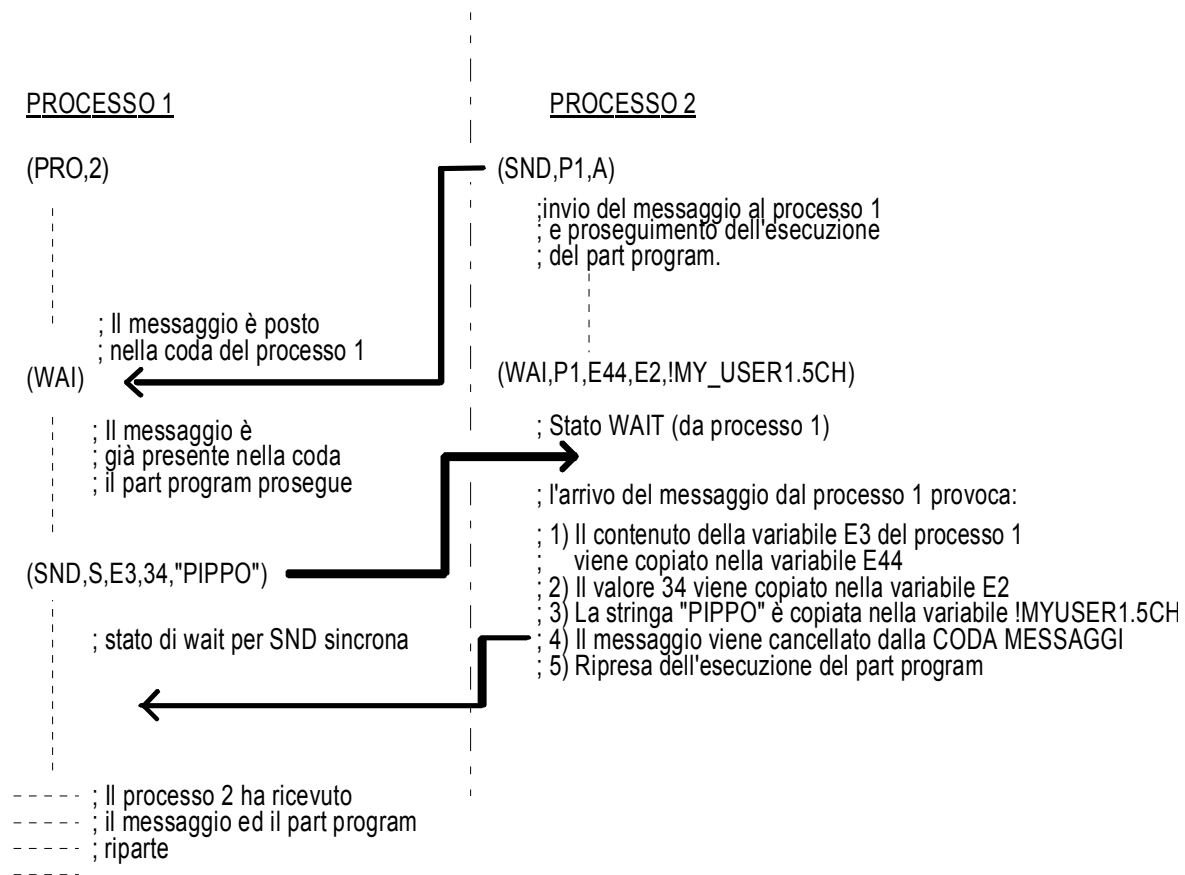
Qualora il *numero processo* non fosse specificato, il messaggio verrà inviato al processo di default dichiarato con il trilettale PRO.

variabili:

Lista delle variabili, al massimo 20, nelle quali ricevere i valori alfa. Se il numero od il tipo delle variabili programmate non è congruente con quelle inviate dal processo mittente viene generato l'errore **"NC225 CARICAMENTO DATI FALLITO"** (vedere anche la funzione SND per ulteriori dettagli).

IMPORTANTE

In stato WAIT, all'arrivo del messaggio di sincronizzazione con dei dati associati, l'invio dell'eventuale ACK richiesto dal mittente viene effettuato dopo che il caricamento dei dati nelle variabili programmate si è concluso; quindi, se il caricamento dei dati fallisce (per errore di numero o di formato), viene trasmesso un ACK negativo al processo mittente dei dati.



Esempi di utilizzo dei comandi SND e WAI



EXE - Attivazione automatica di un part program

Il triletterale EXE permette di attivare ed eseguire in modo automatico un part program sul processo specificato.

Sintassi

(EXE,nome part program [,P numero processo])

dove:

nome part program: Nome del part program (max. 48 caratteri) da eseguire nel modo automatico (AUTO).

Può essere una stringa di caratteri (non tra apici), od una variabile di tipo *carattere preceduta dalla chiave ?*.

numero di processo: Processo su cui attivare il part program.

Può essere un numero, od una variabile locale o di sistema e deve essere compreso tra 1 e 20 (1 ÷ 40 con opzione E65/E66).

Se questo numero è uguale al numero del processo che sta eseguendo il comando EXE, viene generato **NC222 NUMERO DI PROCESSO ERRATO**.

Qualora il *numero processo* non fosse specificato, il messaggio verrà inviato al processo di default dichiarato con il triletterale PRO.

Caratteristiche:

Il comando EXE forza il modo automatico sul processo specificato, quindi attiva il part program e lo manda in esecuzione.

Si ricorda che questi comandi ("SETMODE", "SPG" e "CYCLE") possono essere filtrati dalla logica di macchina.

Se tali comandi giungono ad un processo che non è in grado di eseguirli, per esempio perché si trova nello stato di "RUN", oppure "HOLD" ecc. , viene visualizzato l'errore **NC227 COMANDO EXE O ECM FALLITO**.

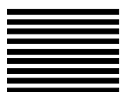
Esempi:

(EXE, MAIN_PROG ,P3) attiva l'esecuzione sul processo 3 del part program MAIN_PROG

(PRO,2)

SC1.10="MAIN_2"

(EXE,?SC1.10) attiva l'esecuzione sul processo 2 del part program MAIN_2



ECM - Esecuzione di un blocco in modo MDI nel processo specificato

Il triletterale ECM permette di attivare l'esecuzione automatica di un blocco in modo MDI sul processo specificato.

Sintassi

(ECM,comando [,P numero processo])

dove:

Comando: Stringa tra apici singoli o doppi, oppure una variabile di tipo stringa. Gli apici singoli si usano quando nel comando vi sono apici doppi.

Esempio: (ECM,'(DIS,"PROVA")',P2)

numero di processo: Processo su cui attivare il part program.

Può essere un numero, od una variabile locale o di sistema e deve essere compreso tra 1 e 20 (1 ÷ 40 con opzione E65/E66).

Se questo numero è uguale al numero del processo che sta eseguendo il comando ECM, viene generato l'errore **"NC222 NUMERO DI PROCESSO ERRATO"**.

Qualora il *numero processo* non fosse specificato, il messaggio verrà inviato al processo di default dichiarato con il triletterale PRO.

Caratteristiche:

ECM forza il modo "MDI" sul processo specificato, manda il comando programmato al processo destinatario e lo fa eseguire tramite l'invio di un "CYCLE ON" su quel processo.

Si ricorda che questi comandi ("SETMODE" e "CYCLE") possono essere filtrati dalla logica di macchina.

Il comando ECM è di tipo asincrono; questo significa che dopo l'esecuzione di ECM il part program prosegue con i blocchi successivi indipendentemente dal fatto che il comando inviato tramite ECM su di un altro processo sia concluso oppure no.

E' importante perciò non mandare un altro comando ECM sullo stesso processo finché non è finita l'esecuzione del comando precedente

Se questo avviene, viene segnalato l'errore **"NC 227 COMANDO EXE O ECM FALLITO"**.

In generale, se i comandi di forzature del modo "MDI" o "CYCLE ON" giungono ad un processo che non è in grado di eseguirli, per esempio perché si trova nello stato di "RUN", oppure "HOLD" ecc. , viene visualizzato l'errore **"NC227 COMANDO EXE O ECM FALLITO"**.

Esempi:

(ECM,"E1=12",P2) Setta il valore della variabile locale E1 del processo 2 al valore 12.

(PRO,2) Altro esempio:

SC0.5="E1=12" Prepara il comando nella variabile SC0.5

(ECM,SC0.5) Esegue ECM con comando costituito dal contenuto della variabile

Esempio di sincronizzazione su 2 processi e di utilizzo del comando EXE:

;N1 SNA_N		;N1 SNB_N
;N2 MULTIPROCESSO:		;N2 MULTIPROCESSO;
SINCRONIZZAZIONE		SINCRONIZZAZIONE
N3 (DIS,"PROGRAMMA SYSTEM 1")		N3 (DIS,"PROGRAMMA SYSTEM 2")
N4 (EXE,SNB_N,P2)		/N4 (UGS,X,-100,100,Y,-130,150,Z)
N5 (UGS,X,-100,100,Y,-100,100,Z)		N5 SN5=0
N6 T2.2M6		N6 T4.4M6
N7 S1500F500M3		N7 (WAI,P1)
N8 XYZ	→	N8 S400F400M3
N9 E25=0		N9 XY
N10 G81R-14Z-35		N10 E25=0
N11 (RPT,6)		N11 G84R-10Z-22
N12 (ROT,E25)		N12 (RPT,6)
N13 X-80Y		N13 (ROT,E25)
N14 E25=E25+60		N14 X-80Y
N15 (ERP)		N15 E25=E25+60
N16 G80Z50		N16 (ERP)
N17 (ROT,0)		N17 (ROT,0)
N18 (SND,P2,A)	→	N18 G80ZM5
N19 (CLS,MA)		N19 (WAI,P1)
N20 (SND,P2,S)	→	N20 GZ40
N21 GZ90		N21 (WAI,P1)
N22 (SND,P2,A)	→	N22 E8=0
N23 (WOS,SN5>=9)	←	N23 T2,2M6
N24 S200F500M3		N24 S1300F500M3
N25 X-10Y50		N25 G81R2Z-25
N26 Z-25		"D" N26
N27 G1G41X		N27 G91X10
N28 G2IJ		N28 G90
N29 G1G40X10		N29 E8=E8+1
N30 GZM5		N30 #SN5=E8
N31 M30	←	N31 (GTO,D,E8<10)
		N32 G80M5

NOTE SULLA FUNZIONALITA' "ASSI MIGRANTI"

Generalità

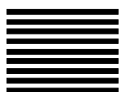
La funzionalità "assi migranti" permette ad uno o più processi di acquisire o rilasciare le "risorse assi", indipendentemente dalla caratterizzazione dei processi stessi. L'acquisizione, od il rilascio di un asse avviene infatti associando un nome asse (X,Y,Z,...) ad un identificatore compreso tra quelli configurati in AMP (1→32).

- Il nome asse è univoco solamente all'interno di un determinato processo.
- L'identificatore di un asse (ID) è univoco su tutta la macchina.

Condizioni per l'acquisizione di un asse

Vi sono alcune condizioni da rispettare quando si intende programmare l'acquisizione di un asse:

- L'asse richiesto deve essere caratterizzato in AMP (deve esistere l'identificativo richiesto nell'ambito dei processi configurati).
- Quando un processo richiede di acquisire uno o più assi, essi debbono essere stati precedentemente rilasciati dal loro processo di appartenenza.
- Il nome dell'asse, deve essere valido; deve cioè essere uno di quelli permessi nella programmazione.
- L'identificatore (ID) dell'asse richiesto non deve essere associato ad un asse ausiliario.
- L'identificatore (ID) dell'asse non deve essere associato ad un asse mandrino.
- L'identificativo (ID) dell'asse deve essere un numero compreso tra 1 e 32 .
- Il nome che si vuole associare all'asse non deve essere già usato dal processo stesso come pseudo asse.



GTA - Acquisizione assi (Get Axes)

La programmazione del triletterale GTA permette la migrazione di uno o più assi da un processo ad un altro, nonché di ridefinire l'associazione correttori/assi.

GTA permette inoltre di impostare l'ordine di visualizzazione degli assi.

Sintassi

(GTA,[Tvalue,]nome_asse ID[,nome_asse ID ...] [/] [nome_asse_1][,nome_asse_2])

dove:

T value	value = 0: perdita origini attivate (funzionamento uguale all'omettere il parametro T0) value = 1: mantenimento origini attivate sugli assi riprogrammati (con nome, ID ed ordine invariati) e sugli assi programmati con nome diverso ma ID ed ordine invariati. value = 2: mantenimento origini attivate su tutti gli assi settati, indipendentemente dall'origine di programmazione degli ID.
/	Il carattere "/" è un separatore che deve essere inserito se si desidera specificare almeno uno dei correttori lunghezza.
nome_asse_1 nome_asse_2	Rappresentano gli assi sui quali saranno applicati i correttori lunghezza 1 e 2 durante un'eventuale successiva attivazione di un correttore.

Caratteristiche:

- Quando si utilizza il triletterale GTA, è obbligatorio dichiarare tutti gli assi che il processo deve utilizzare, anche se già appartenenti al processo stesso.
- Non specificare un asse nel comando GTA significa rilasciarlo per renderlo disponibile ad altri processi.
- Quando un comando GTA è eseguito, l'ordine di visualizzazione degli assi è quello dichiarato nel comando; GTA può quindi essere utilizzato per cambiare l'ordine di visualizzazione degli assi.
- Il comando di RESET **non ripristina** i cambiamenti derivanti dall'utilizzo del GTA.
- Alla riaccensione del controllo la situazione degli assi di tutti i processi sarà quella configurata in AMP.
- Durante le operazioni di cambio utensile, ed in particolare durante il comando di attivazione offset, gli assi ai quali associare i correttori lunghezza devono essere già acquisiti dal processo.

INIBIZIONE DELLE OPERAZIONI DI ACQUISIZIONE E RILASCIO ASSI.

Vi sono dei casi in cui le operazioni eseguite dal triletterale GTA sono inibite:

- Con la modalità correzione diametro utensile attiva
- In presenza di ciclo fisso attivo (G81-89)
- Nello stato di hold
- Nello stato IDLE-MAS
- In presenza di virtualizzazioni attive
- Con la modalità correzione utensile (h oppure T) attiva e si sta abbandonando un asse che ha applicata la correzione utensile oppure si sta acquisendo un nuovo asse sul quale verrà applicata la correzione utensile.
- Rilascio di assi condivisi con la logica

INFORMAZIONI INIZIALIZZATE DOPO UN COMANDO GTA

Piano di interpolazione.

- Il piano di interpolazione sarà composto da una coppia tra i primi tre assi programmati in base alla "G" di cambio piano attiva --> G17, G18, G19.
- Qualora il numero degli assi definiti nel comando GTA sia inferiore a 3, verranno forzati rispettivamente il primo ed il secondo asse come ascissa e come ordinata.
- Per un corretto funzionamento occorre programmare con la G16 il piano desiderato.

Origini.

- Se il parametro "Tvalue" è omesso oppure è impostato con "value=0" le origine attive saranno resettate.
- Se il parametro "Tvalue" è impostato con value=1 le origini verranno mantenute sugli assi riprogrammati con nome, ID ed ordine invariati e sugli assi programmati con nome diverso ma ID ed ordine invariati.

Mirror.

- Se il parametro "Tvalue" è impostato con value = 2 le origini verranno mantenute su tutti gli assi programmati anche se con ID differenti.
- Il mirror sarà resettato su tutti gli assi.

Fattore di scala.

- Il fattore di scala sarà resettato su tutti gli assi.

Rotazione "ROT".

- La rotazione attivata da part program con il triletterale "ROT" sarà annullata.

Limiti operativi.

- I valori dei limiti operativi di tutti gli assi saranno quelli di configurazione.

Selezione asse per i movimenti manuali.

- Dopo l'esecuzione di un comando GTA sarà eseguita la selezione, per i movimenti manuali, del primo asse in ordine di visualizzazione.

Correttori lunghezza.

- Gli identificativi degli assi ai quali saranno associati i correttori lunghezza 1 e/o 2 saranno programmati opzionalmente sul triletterale GTA.
Le associazioni di default definite in AMP saranno perse e riattivate dopo una riaccensione del controllo.

INFORMAZIONI MANTENUTE DOPO UN COMANDO GTA

- **Axis locked**
- **Rotazione piano attivata con comando dalla finestra di input "PROGRAM SETUP".**
- **Origini incrementali e temporanee.**

Esempio 1:

Condizioni iniziali:

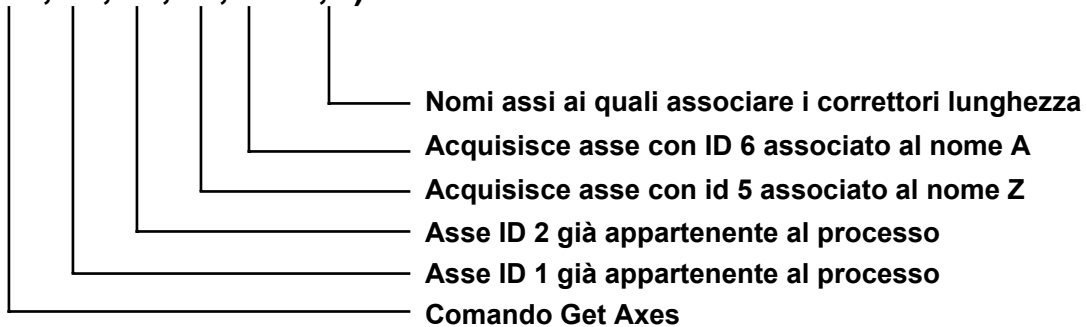
- Il processo interessato detiene gli assi X (con ID 1), ed Y (con ID 2).

Obiettivo da raggiungere:

- Mantenere la proprietà degli assi X e Y (ID 1 e 2).
- Acquisire gli assi con ID 5 e 6 ed assegnare loro i nomi Z ed A.
- Associare il correttore lunghezza 1 all'asse Y (ID 2).
- Associare il correttore lunghezza 2 all'asse Z (ID 5).
- Ottenere un'ordine di visualizzazione (sul video di sistema); X,Y,Z,A.

Esempio di programmazione:

(GTA,X1,Y2,Z5,A6/Y,Z)



IMPORTANTE

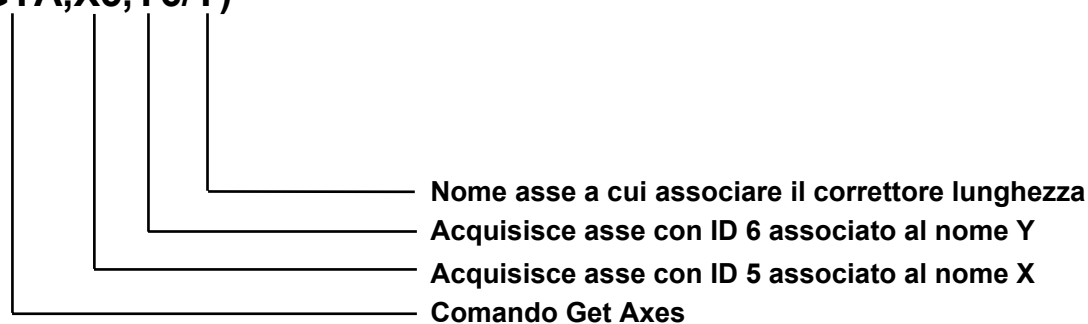
La programmazione di tutti gli assi, anche di quelli già appartenenti al processo, è obbligatoria e permette di ridefinire sia il nome dell'asse che l'ordine di visualizzazione.

Esempio 2:**Condizioni iniziali:**

- Il processo interessato detiene gli assi X (con ID 1), ed Y (con ID 2).

Obiettivo da raggiungere:

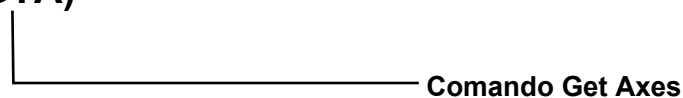
- Rilasciare gli assi X e Y (ID 1 e 2).
- Acquisire gli assi con ID 5 e 6 e chiamarli X ed Y.
- Associare il correttore lunghezza 1 all'asse Y (ID 6).
- Ottenere un'ordine di visualizzazione (sul video di sistema) X,Y.

Esempio di programmazione:**(GTA,X5,Y6/Y)****Esempio 3:****Condizioni iniziali:**

- Il processo interessato detiene un numero imprecisato di assi.

Obiettivo da raggiungere:

- Rilasciare tutti gli assi in suo possesso.

Esempio di programmazione:**(GTA)**

Esempio 4:

Condizioni iniziali:

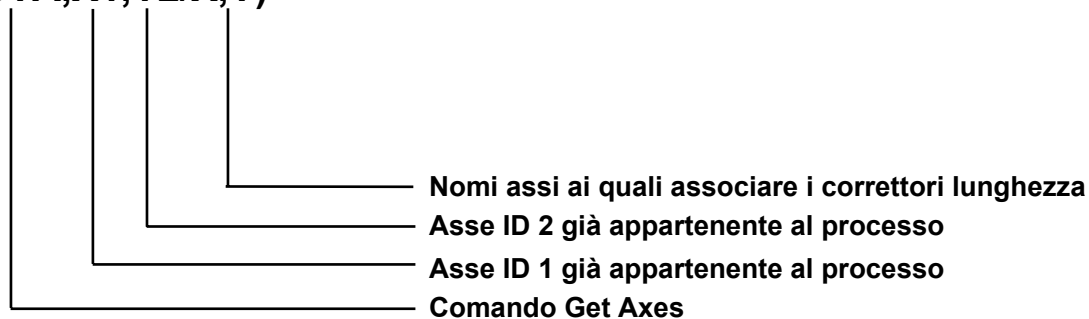
- Il processo interessato detiene gli assi X,Y,Z,A,B con ID 1,2,3,4,5.

Obiettivo da raggiungere:

- Mantenere le proprietà degli assi con ID 1 e 2 (X,Y).
- Rilasciare gli assi Z,A,B con ID 3,4,5.
- Associare il correttore lunghezza 1 all'asse X (ID 1).
- Associare il correttore lunghezza 2 all'asse Y (ID 2).

Esempio di programmazione:

(GTA,X1,Y2/X,Y)

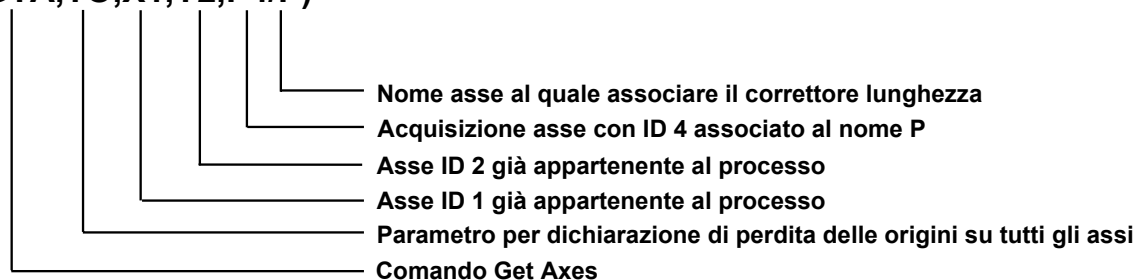


Esempio 5:**Condizioni iniziali:**

- Il processo interessato detiene gli assi X,Y,Z con ID 1,2,3.
- Agli assi X,Y,Z è associata l'origine 1.

Obiettivo da raggiungere:

- Mantenere le proprietà degli assi con ID 1 e 2 (X,Y).
- Rilasciare l'asse Z (ID 3).
- Acquisire l'asse P con ID 4.
- Associare il correttore lunghezza 1 all'asse P (ID 4).
- Ottenere un'ordine di visualizzazione (sul video di sistema): X,Y,P.
- Perdere l'origine associata di tutti gli assi.

Esempio di programmazione:**(GTA,TO,X1,Y2,P4/P)**

Esempio 6:

Condizioni iniziali:

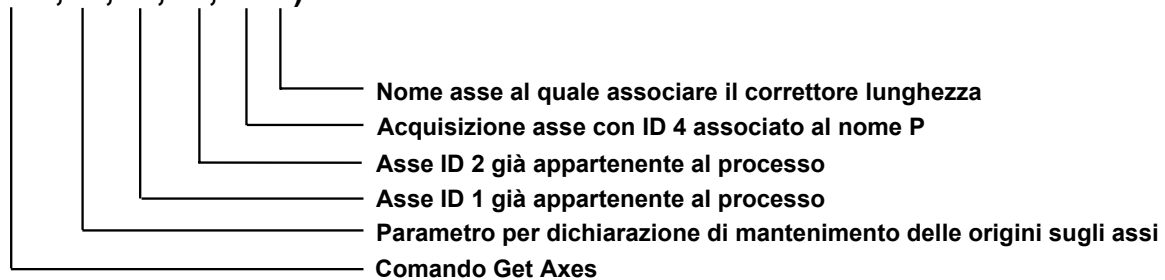
- Il processo interessato detiene gli assi X,Y,Z con ID 1,2,3.
- Agli assi X,Y,Z è associata l'origine 1.

Obiettivo da raggiungere:

- Mantenere le proprietà degli assi con ID 1 e 2 (X,Y).
- Rilasciare l'asse Z (ID 3).
- Acquisire l'asse P con ID 4.
- Associare il correttore lunghezza 1 all'asse P (ID 4).
- Ottenere un'ordine di visualizzazione (sul video di sistema): X,Y,P.
- Mantenere l'origine associata agli assi con ID 1 e ID 2.

Esempio di programmazione:

(GTA,T1,X1,Y2,P4/P)

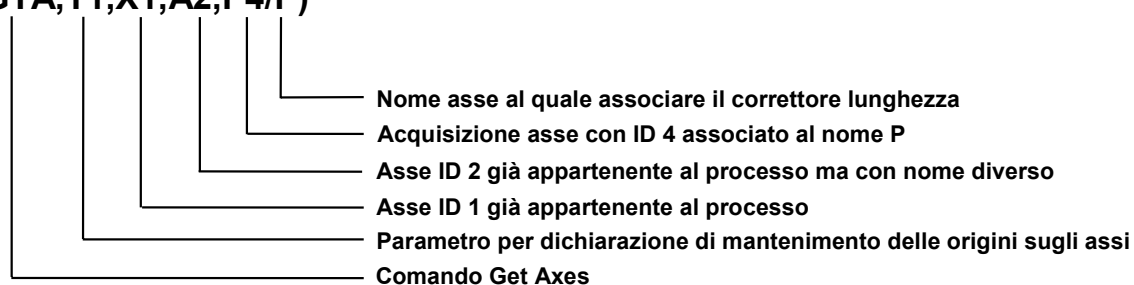


Esempio 7:**Condizioni iniziali:**

- Il processo interessato detiene gli assi X,Y,Z con ID 1,2,3.
- Agli assi X,Y,Z è associata l'origine 1.

Obiettivo da raggiungere:

- Mantenere le proprietà dell'asse con ID 1 (X).
- Mantenere le proprietà dell'asse con ID 2 (Y) cambiando il nome in (A).
- Rilasciare l'asse Z (ID 3).
- Acquisire l'asse P con ID 4.
- Associare il correttore lunghezza 1 all'asse P (ID 4).
- Ottenere un'ordine di visualizzazione (sul video di sistema): X,A,P.
- Mantenere l'origine associata agli assi con ID 1 e ID 2.

Esempio di programmazione:**(GTA,T1,X1,A2,P4/P)**

Esempio 8:

Condizioni iniziali:

- Il processo interessato detiene gli assi X,Y,Z con ID 1,2,3.
- Agli assi X,Y,Z è associata l'origine 1.
- Un 2° processo detiene gli assi A,B con ID 4,5 ai quali è associata una origine 2

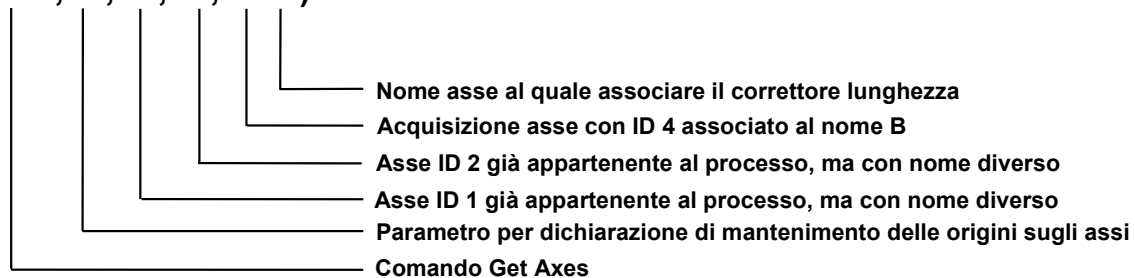
Obiettivo da raggiungere:

- Mantenere le proprietà degli assi con ID 1, 2 e 4.
- Rilasciare l'asse Z (ID 3).
- Acquisire l'asse B con ID 4 precedentemente rilasciato dal processo 2.
- Associare il correttore lunghezza 1 all'asse B (ID 4).
- Ottenere un'ordine di visualizzazione (sul video di sistema): Y,X,B.
- Mantenere l'origine associata a tutti gli assi, ovvero origine 1 su assi con id 1 (Y), 2 (X) e origine 2 su asse id 4 (B).

Esempio di programmazione:

(GTA) sul processo 2 per il rilascio di A e B

(GTA,T2,Y1,X2,B4/B)



Esempio 9:**Condizioni iniziali:**

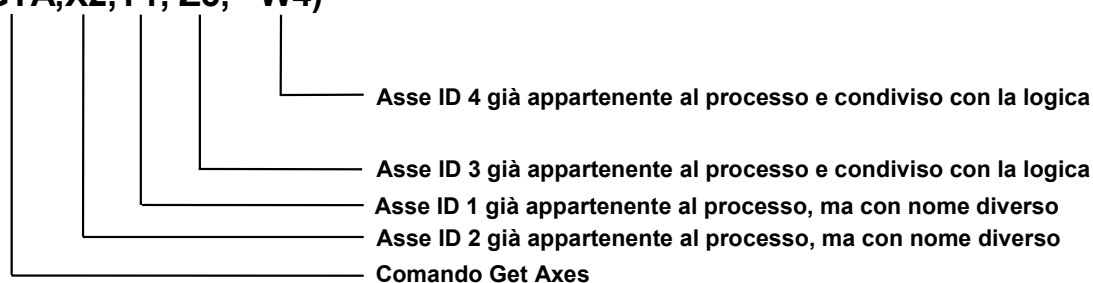
- Il processo interessato detiene gli assi X,Y,Z con ID 1,2,3,4.
- L'asse con ID 4 è "condiviso con la logica"

Obiettivo da raggiungere:

- Scambiare l'ascissa con l'ordinata.

Esempio di programmazione:

(GTA,X2,Y1, Z3, W4)

**NOTA:**

Un asse condiviso con la logica non può essere rilasciato tramite GTA (il sistema segnala errore NC124) e, ovviamente, appartenendo alla logica, se pur visibile dall'ambiente di processo, non può essere programmato.

Se serve utilizzare il comando GTA su un processo in cui siano presenti uno o più assi condivisi con la logica è necessario pertanto riconfermare gli assi condivisi come scritto nell'esempio.

Gestione delle condizioni di errore

In caso di errore, riscontrato durante l'esecuzione del triletterale (es. risorsa asse non disponibile, ID asse inesistente, . . .), il programmatore di part program può gestire l'evento nel modo standard, (stop dell'esecuzione del part program e visualizzazione errore a video) oppure da part program utilizzando le variabili "ERR" e "STE".

Per maggiori informazioni riguardanri queste variabili, consultare l'appendice C.

Esempio di acquisizione assi da parte di un processo:

```
;N1 SINUS_AM
;N2 ESECUZIONE SU PROC.1
;N3 ACQUISIZIONE ID9-ID10 (NUOVI ASSI A-B) DA PROC. 2
N6 (EXE,GTA1_AM,P2)
N7 (DLY,0.1)
N8 (GTA,X1,Y2,Z3,A9,B10,-Z)
N9 T1.1M6
N10 S1000M3
N11 G0G90XYZAB
N12 G1G91G28
/N13 (UGS,1,Z,-80,80,Y,-50,50,Z)
N14 E25=0.03
N15 X0.2713Y0.0947Z-0.1852A-0.1815B0.1007tE25
N16 X0.2711Y0.0886Z-0.1887A-0.1776B0.1066tE25
N17 X0.2708Y0.0825Z-0.1922A-0.1736B0.1125tE25
N18 X0.2703Y0.0764Z-0.1955A-0.1696B0.1184tE25
N19 X0.2697Y0.0702Z-0.1987A-0.1654B0.1241tE25
N20 X0.2689Y0.0640Z-0.2017A-0.1610B0.1299tE25
.....
.....
.....
N2010 X0.2697Y0.1241Z-0.1654A-0.1987B0.0702tE25
N2011 X0.2703Y0.1184Z-0.1696A-0.1955B0.0764tE25
N2012 X0.2708Y0.1125Z-0.1736A-0.1922B0.0825tE25
N2013 X0.2711Y0.1066Z-0.1776A-0.1887B0.0886tE25
N2014 X0.2713Y0.1007Z-0.1815A-0.1852B0.0947tE25
N2015 G0
N2016 (SND,P2,S)
N2017 (GTA,X1,Y2,Z3)
N2018 M30
```

FINE CAPITOLO

PROGRAMMAZIONE GEOMETRICA AD ALTO LIVELLO (GTL)

All'interno del sistema, in un programma, è possibile descrivere un profilo geometrico nel piano utilizzando oltre al linguaggio di programmazione standard (G1-G2-G3), anche il linguaggio di programmazione ad alto livello GTL.

Questo linguaggio permette di programmare un profilo fatto di rette e cerchi con le sole informazioni ricavate dal disegno, delegando al CNC Serie 10 il compito di calcolare i punti di intersezione e tangenza fra gli elementi geometrici.

Il linguaggio di programmazione GTL e quello standard possono essere utilizzati contemporaneamente sullo stesso programma, ma non nello stesso profilo. La geometria GTL funziona solo in programmazione assoluta (G90).

GEOMETRIA ORIENTATA

La definizione di un profilo mediante GTL è basata sull'utilizzazione di quattro tipi di "enti geometrici":

- origini di riferimento
- punti
- rette
- cerchi

Poiché un profilo, oltre che da enti geometrici, è definito anche da un senso di percorrenza, per la definizione degli enti geometrici nel linguaggio GTL viene utilizzato un particolare tipo di geometria:

GEOMETRIA ORIENTATA.

Per la geometria orientata, la definizione di un ente richiede, oltre ai parametri necessari per individuarne la posizione nel piano, anche l'assegnazione di un verso di percorrenza.

Ad esempio, per i punti A e B passa una retta I' percorribile muovendo da A verso B oppure una retta I , giacente su I' , ma percorribile da B verso A.



Rette I e I' orientate

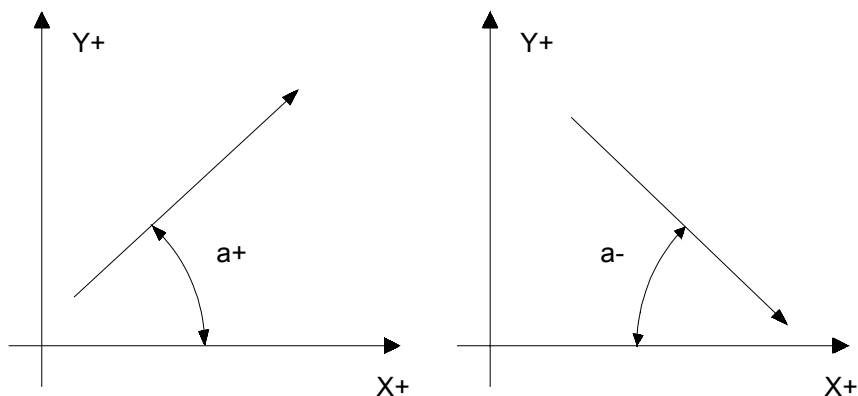
Nella geometria orientata, I e I' sono due rette diverse.

La programmazione mediante il GTL, basata sulla geometria orientata, richiede l'assegnazione ad ogni retta di un verso presunto di percorrenza.

IMPORTANTE

Il verso di percorrenza della retta è dato, per convenzione, dall'angolo che essa forma con l'asse delle X positive.

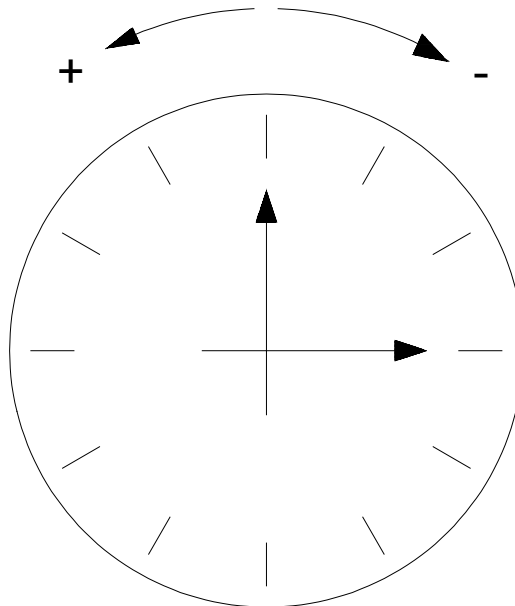
L'angolo corretto si ottiene ruotando l'asse delle X positive fino a sovrapporlo in direzione e verso a quello della retta da definire. L'angolo avrà segno positivo se l'asse X ruoterà in senso antiorario, negativo in caso contrario.



Convenzione per gli angoli

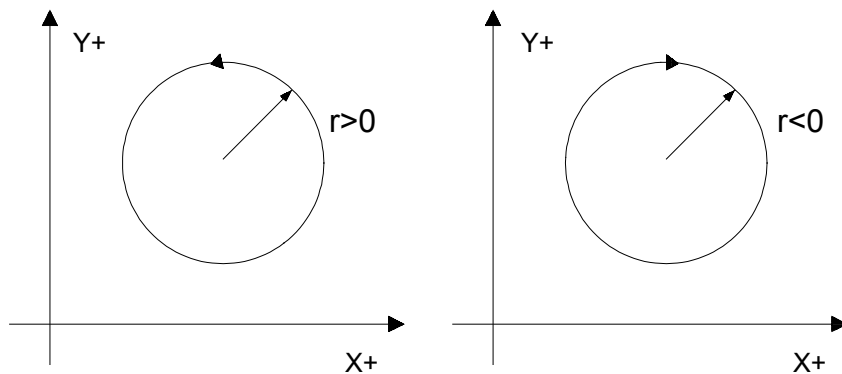
IMPORTANTE

Anche ai cerchi si deve associare un verso di percorrenza. Convenzionalmente viene scelto come verso di percorrenza positivo quello antiorario, e come negativo quello contrario.



Verso di percorrenza orario e antiorario

La convenzione adottata associa un valore positivo del raggio a cerchi percorsi in senso antiorario e negativo nel caso opposto.



Raggio di un cerchio e verso di percorrenza

E' comunque possibile, in fase di definizione del profilo, cambiare la direzione dell'elemento, nel caso risulti contraria.

MEMORIZZAZIONE DI ENTI GEOMETRICI

Il linguaggio GTL prevede l'utilizzo dei caratteri minuscoli a-l-c-d-m-o-r-p-s-b per definire rispettivamente angoli, rette, cerchi, distanze, modulo, origine di riferimento, raggio, punto, numero di intersezione, smusso.

La necessità di utilizzare per queste informazioni i caratteri minuscoli è dovuta al fatto che gli stessi caratteri maiuscoli sono utilizzati nel linguaggio del CN per altre informazioni.

La memorizzazione di enti geometrici va fatta prima della definizione del profilo.

Gli enti trattati dal GTL sono rette, cerchi, punti, origini di riferimento: variabili geometriche identificate da un NOME e da un INDICE. L'ente viene definito in un blocco di assegnazione.

Il formato del blocco è il seguente:

NOME INDICE = <Espressione>

dove:

NOME è uno dei quattro nomi simboli previsti per gli enti geometrici:

- o** per definizione di origini di riferimento
- p** per definizione di punti
- l** per definizione di rette
- c** per definizione di cerchi

INDICE Definisce il numero della variabile geometrica **NOME** (ente).
E' un numero compreso fra 0 ed il limite massimo configurato.

espressione Contiene tutte le informazioni per descrivere l'ente geometrico.

Gli enti possono essere definiti:

- **in modo diretto (esplicito)**, programmando nel blocco tutte le informazioni necessarie a identificare l'ente geometrico;
- **in modo indiretto (implicito)**, richiamando altri enti geometrici precedentemente definiti.

Esempio di memorizzazione di enti:

```

o1  =  X30 Y30 a45
p1  =  o1 X15 Y15
p2  =  X60 Y30
l1  =  p1, p2
l2  =  X30 Y50, a45
c1  =  l1,l2,r15
l3  =  X0 Y0, X100 Y60
p3  =  l3, c1
c2  =  p3,r8

```

Il numero di elementi geometrici memorizzabili è definito all'atto della configurazione del sistema.
Il formato di scrittura delle definizioni geometriche prevede l'utilizzo del carattere "," (virgola) per separare l'ente geometrico (retta - punto - cerchio) dall'ente successivo o dall'informazione successiva tipo raggio "r" o angolo "a".

ESEMPLI:

p1 = X30 Y30 il separatore non è richiesto

c1 = I10 J20 r30

I1 = X20 Y20, X100 Y-10

 | |
 punto punto

I2 = I30 J30 r10, X80 Y80

 | |
 cerchio punto

I3 = X100 Y100, a45

 | |
 punto angolo

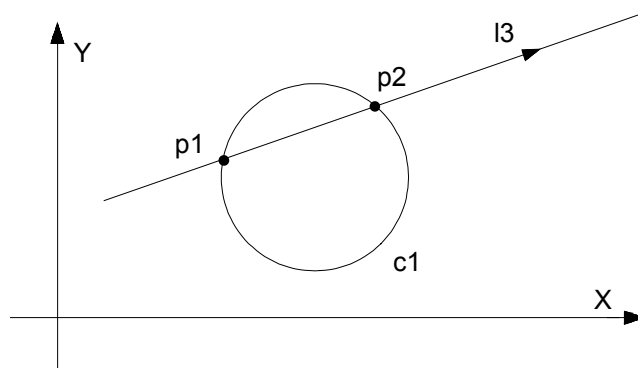
p5 = c1, c2

 | |
 cerchio cerchio

c3 = I1, I2, r18

 | | |
 retta retta raggio

Il discriminatore s2 serve per scegliere la seconda intersezione (s2).



Punti di intersezione tra retta e cerchio

p1 = c1, I3

p2 = c1, I3, s2

DEFINIZIONI GEOMETRICHE

Origine di riferimento	on = X.. Y.. a..
Punti	pn = [on] X.. Y.. pn = [on] m.. a.. pn = ln, ln' pn = [-] ln, cn [,s2] pn = cn, [-]ln [,s2] pn = cn, cn [,s2]
Rette	ln = [on] X.. Y.., [on] X.. Y.. ln = [on] X.. Y , a.. ln = [on] l.. J.. r.., [on] l.. J.. r.. ln = [on] l.. J.. r.., a.. ln = [on] l.. J.. r.., [on'] X.. Y.. ln = [on] X.. Y.., [on'] l.. J.. r.. ln = pn, pn' ln = pn, a.. ln = [-]cn, [-]cn' ln = [-]cn, a.. ln = [-]cn, pn' ln = pn, [-]cn' ln = [-]ln, d..
Cerchi	cn = [on] l.. J.. r.. cn = [on] m.. a.. r.. cn = [-]ln, [-]ln, r.. cn = [-]ln, [-]cn, r.. cn = [-]cn, [-]ln, r.. cn = pn, [-]ln, r.. cn = [-]ln, pn, r.. cn = [-]cn, [-]cn', r.. cn = pn, [-]cn, r.. cn = [-]cn, pn, r.. cn = pn, pn', r.. cn = pn, [-]ln cn = pn, [-]cp [,s2] cn = pn, pn', pn" cn = pn, r.. cn = [-]cn, [-]d..

NOTA:

Una serie di punti segnala la necessità di fissare valori numerici. Le voci racchiuse fra parentesi quadre [] sono facoltative e possono essere omesse.

DEFINIZIONE DI ORIGINE DI RIFERIMENTO

Funzione Il GTL consente di definire origini di riferimento in un formato diretto (esplicito).

Descrizione Normalmente le informazioni contenute nel programma sono riferite ad un unico sistema di assi coincidente con gli assi macchina. Tuttavia, per esigenze di progettazione, il particolare può essere stato disegnato impiegando diversi sistemi cartesiani: il sistema assoluto, e altri sistemi (origini) di riferimento riconducibili al sistema assoluto con rotazioni e traslazioni di assi. La geometria GTL può essere definita sotto qualsiasi sistema di riferimento.

Formato diretto:

on = X.. Y.. a..

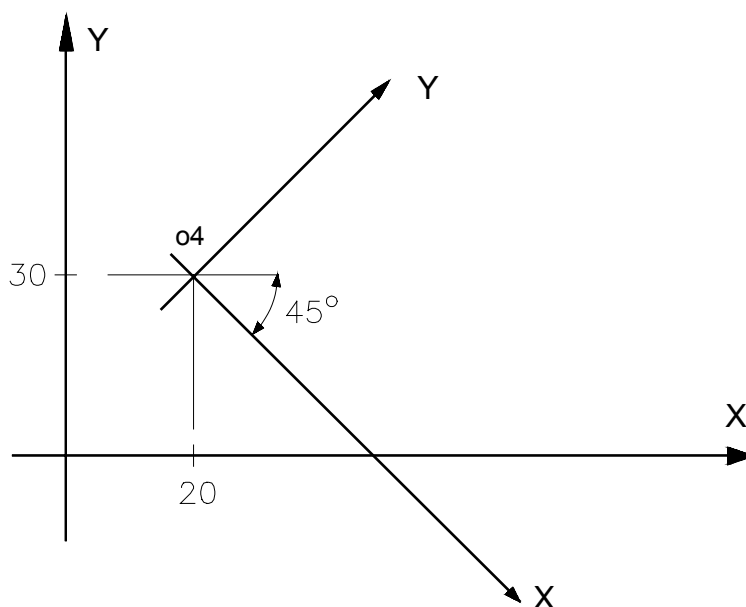
dove:

on identifica il nome dell'origine di riferimento

X.. Y.. coordinate della nuova origine

a.. angolo di rotazione (positivo antiorario).

Esempio:

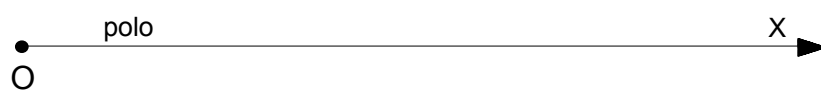


o4 = X20 Y30 a-45

DEFINIZIONE DI PUNTI

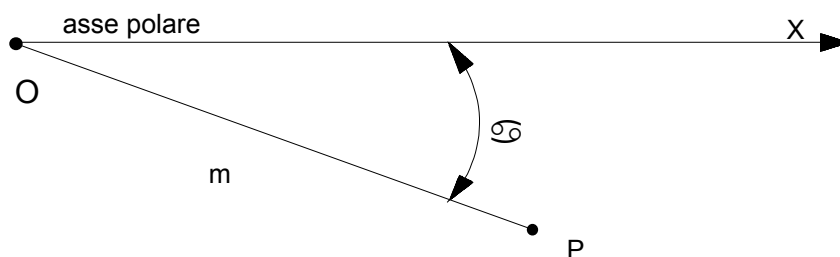
Funzione IL GTL permette di definire i punti in forma diretta (esplicita) o in forma indiretta (implicita). La definizione può essere data sia in coordinate cartesiane che in coordinate polari.

Descrizione Il sistema di riferimento polare è costituito da una origine detta polo da cui esce l'asse X, detto asse polare.



Asse polare

Un punto qualsiasi del piano può essere identificato mediante la lunghezza del segmento che unisce il polo al punto P (detto modulo) ed il valore dell'angolo che tale segmento forma con l'asse polare.



Coordinate polari

Formato Diretto

Punto in coordinate cartesiane

pn = [on] X.. Y..

Punto in coordinate polari

pn = [on] m.. a..

Formato Indiretto

Punto intersezione di due rette predefinite

pn = ln, ln'

Punto intersezione di una retta e di un cerchio predefiniti

pn = [-]ln, cn' [,s2]

pn = cn, [-]ln [,s2]

Punto intersezione di due cerchi

pn = cn, cn' [,s2]

dove:

pn identifica il nome del punto di indice n (n è un numero compreso fra 1 ed il massimo configurato)

X.. Y.. coordinate del punto

[on] origine di riferimento precedentemente definito di indice n cui si riferiscono le coordinate X e Y

m.. modulo del vettore polare

a.. angolo del vettore polare

cn elementi cerchio predefiniti di indici n e n'

cn'

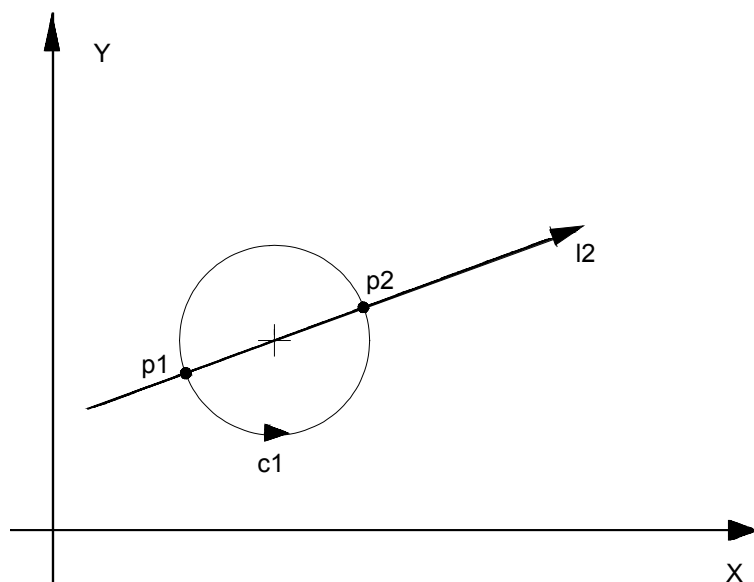
[-]ln elementi retta predefiniti di indice n e n'. E' possibile cambiare il verso inserendo il segno "-"

[-]ln

[,s2] indicatore di seconda intersezione

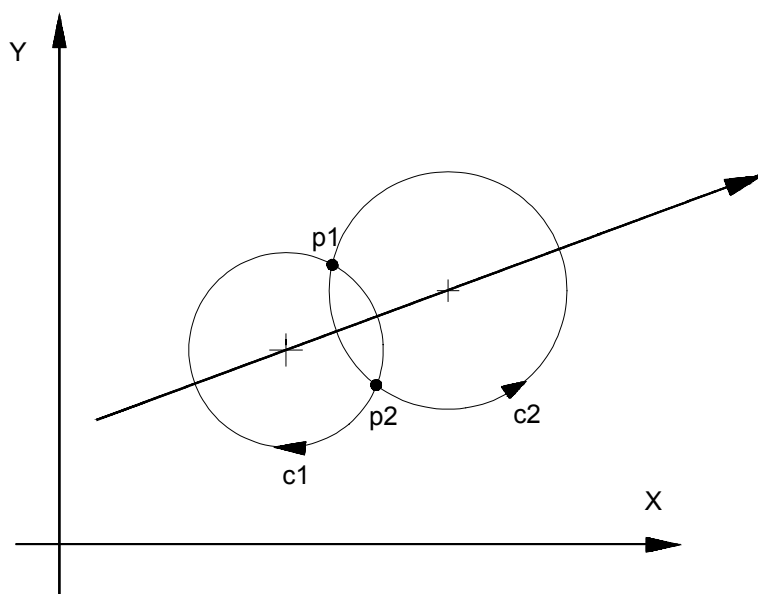
NOTA:

In caso di intersezione retta-cerchio o viceversa, ci sono due possibili soluzioni: il cerchio $c1$ e la retta $l2$ si intersecano nei punti $p1$ e $p2$. Percorrendo la retta $l2$, secondo il suo verso si incontra prima il punto $p1$ (1° intersezione), quindi il punto $p2$ (2° intersezione). Per selezionare la seconda intersezione ($p2$) si deve fare uso dell'indicatore $s2$. Se omesso si seleziona la prima intersezione ($p1$).

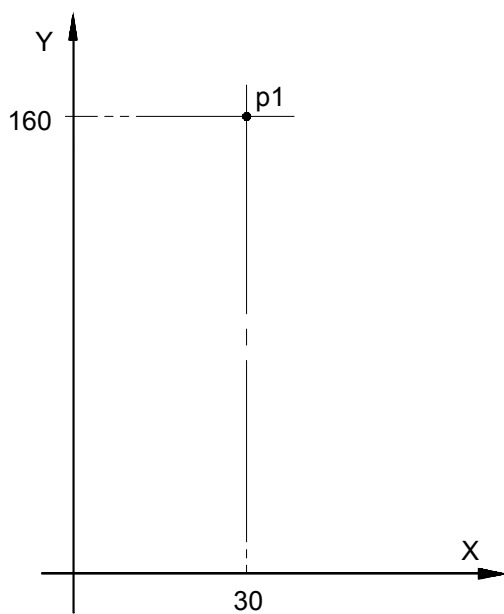


Intersezione retta-cerchio

Anche in caso di intersezione cerchio-cerchio ci sono due possibili soluzioni: i cerchi $c1$ e $c2$ si intersecano nei punti $p1$ e $p2$. Si consideri la retta orientata congiungente il centro del 1° cerchio (cui si fa riferimento) con centro del 2° cerchio. Essa divide il piano in due semipiani. Per selezionare il punto che si trova sul semipiano destro ($p2$) si deve fare uso dell' indicatore $s2$. Se omesso, si seleziona automaticamente il punto sul semipiano sinistro ($p1$).

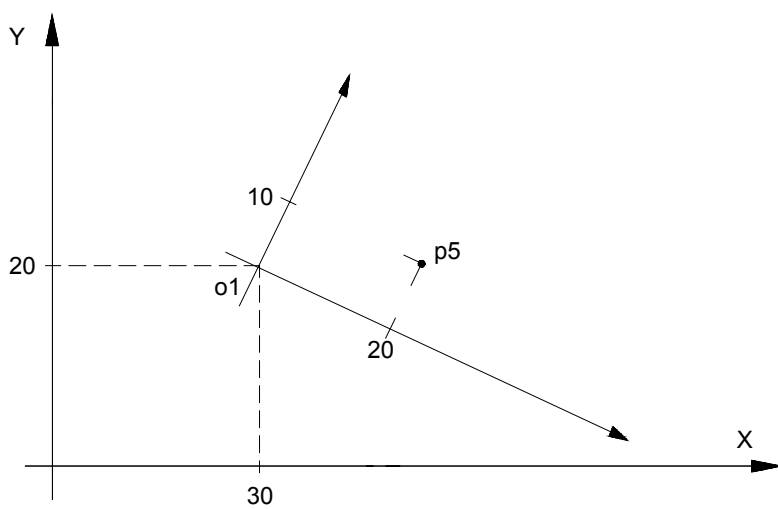


Intersezione cerchio-cerchio.



Punto in coordinate cartesiane

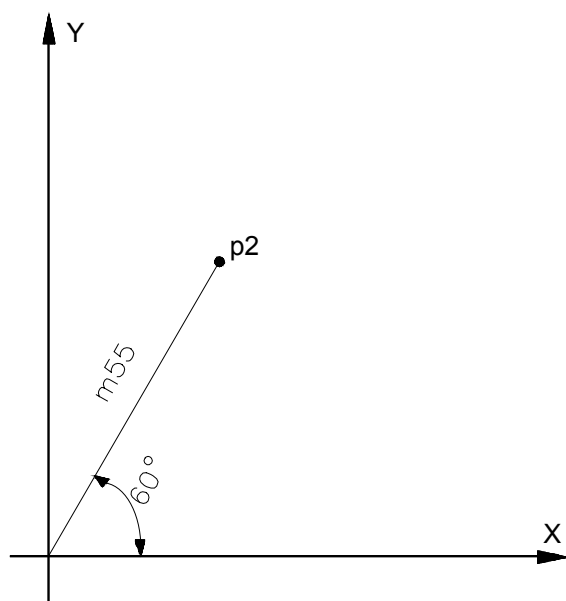
p1 = X30 Y160



Punto in coordinate cartesiane

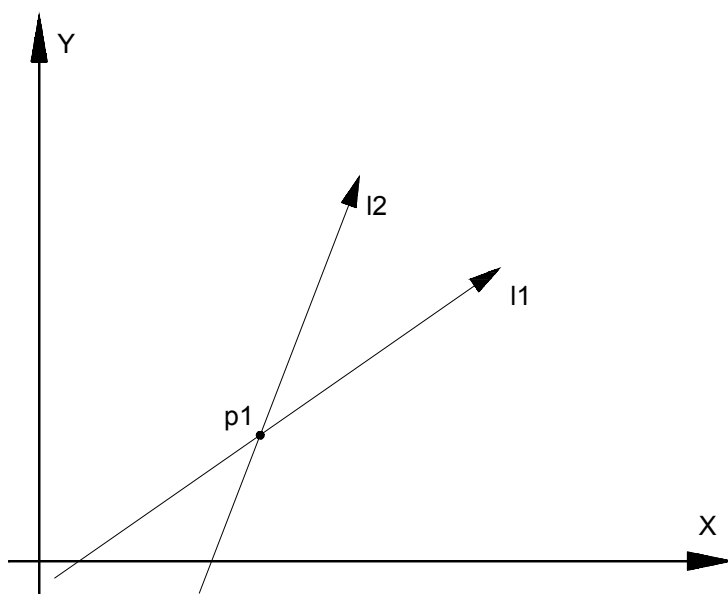
o1 = X30 Y20 a-20

p5 = o1 X20 Y10



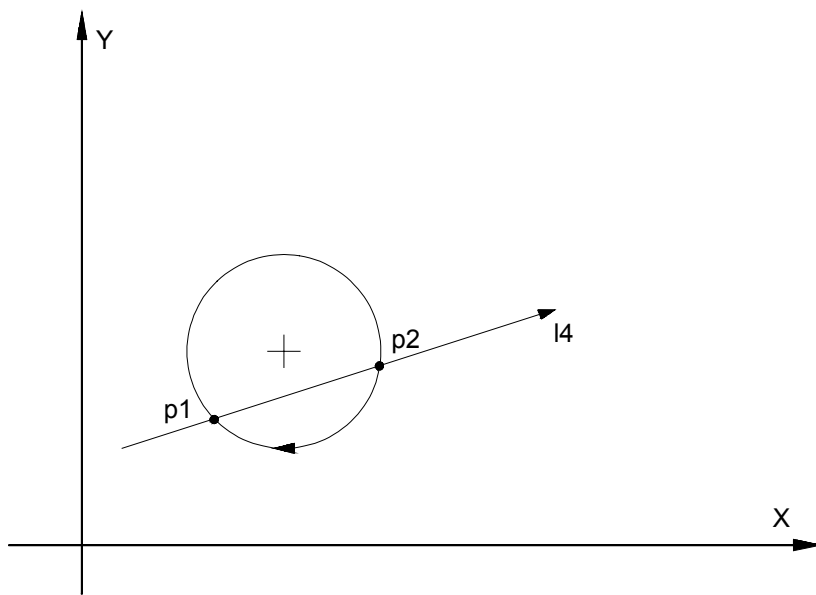
Punto in coordinate polari

$p2 = m55 \ a60$



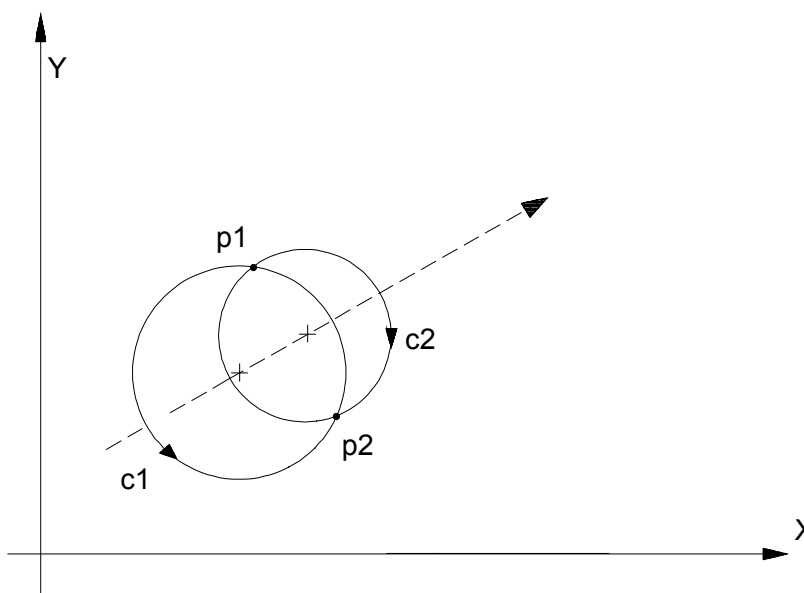
Punto di intersezione tra due
rette predefinite

$p1 = l1, l2$



Punti di intersezione tra
retta e cerchio

$p1 = l4, c3$
 $p2 = l4, c3, s2$
 $p1 = -l4, c3, s2$



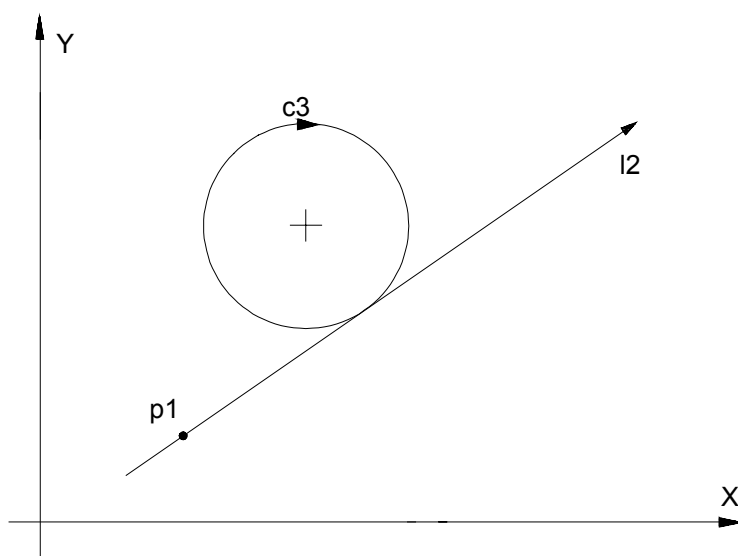
Punti di intersezione tra due
cerchi

$p1 = c1, c2$
 $p2 = c1, c2, s2$
 $p1 = c2, c1, s2$

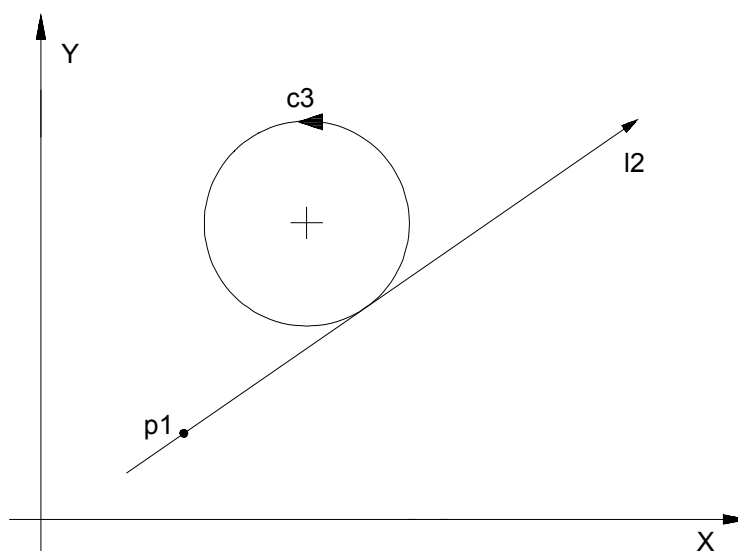
DEFINIZIONE DI RETTA

Funzione Il GTL consente di definire le rette in forma diretta (esplicita) o indiretta (implicita).

Descrizione Il verso assunto dalla retta è sempre quello che va dal primo al secondo ente definito. Nel caso di una retta tangente ad un cerchio sono possibili due soluzioni, poiché la retta può trovarsi in condizione di tangenza con l'uno o l'altro lato del cerchio. Per selezionare, tra le due soluzioni, quella richiesta, occorre assicurarsi che nel punto di tangenza il cerchio e la retta abbiano lo stesso verso.



Incompatibilità dei versi di percorrenza di enti geometrici



Compatibilità tra i versi di percorrenza di enti geometrici

Formato Diretto

Retta passante per due punti

$$In = [on] X.. Y.., [on'] X.. Y..$$

Retta passante per un punto e formante un angolo con l'asse ascissa:

$$In = [on] X.. Y.., a..$$

Retta tangente ad un cerchio e formante un angolo con l'asse ascissa:

$$In = [on] I.. J.. r.., a..$$

Retta tangente a due cerchi:

$$In = [on] I.. J.. r.., [on'] I.. J.. r..$$

Retta tangente ad un cerchio e passante per un punto:

$$In = [on] I.. J.. r.., [on] X..Y..$$

$$In = [on] X.. Y.., [on] I.. J.. r..$$

Formato Indiretto

Retta passante per due punti:

$$In = pn, pn'$$

Retta passante per un punto e formante un angolo con l'asse ascissa:

$$In = pn, a..$$

Retta tangente a due cerchi:

$$In = [-]cn, [-]cn'$$

Retta tangente ad un cerchio e formante un angolo con l'asse ascissa:

$$In = [-]cn, a..$$

Retta tangente ad un cerchio e passante per un punto:

$$In = [-] cn, pn$$

$$In = pn, [-]cn$$

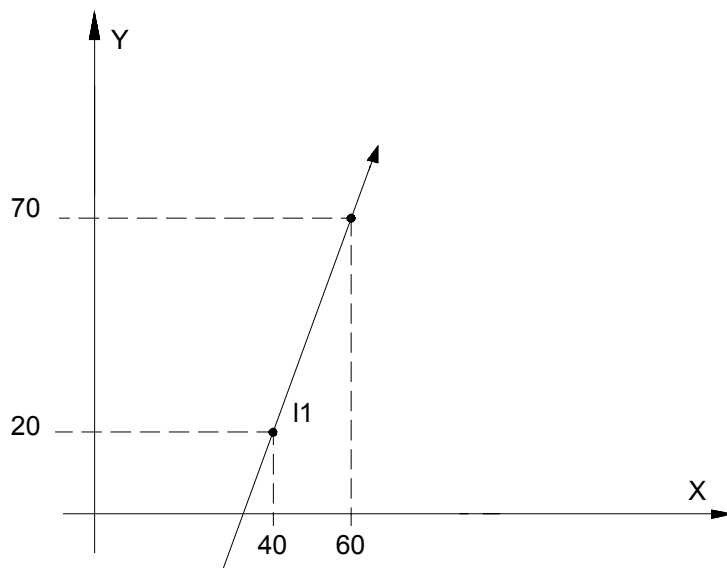
Retta parallela ad una retta a distanza d:

$$In = [-]In, d..$$

$$In = [-]cn, a..$$

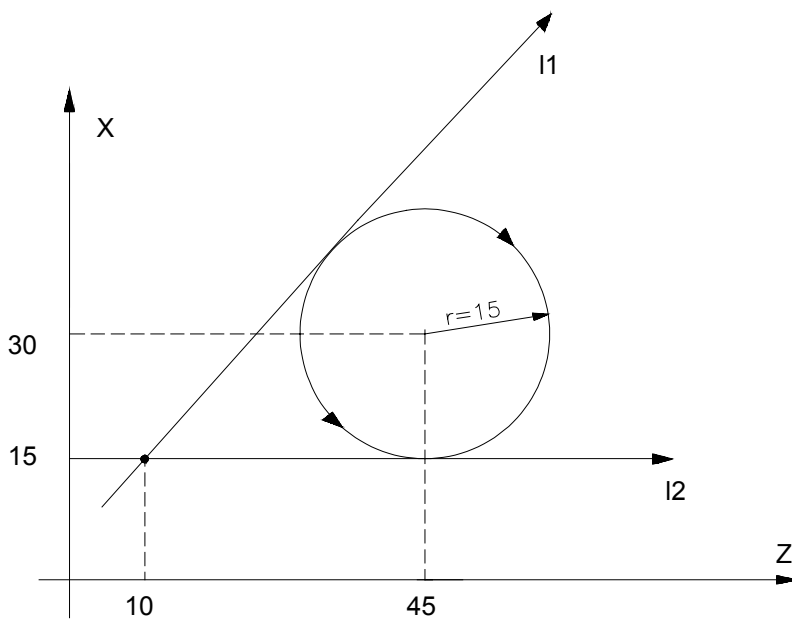
dove:

In	identifica il nome della retta di indice n (n è un numero compreso tra 0 ed il numero massimo configurato)
X.. Y..	coordinate di un punto
a..	angolo formato dall'asse ascissa e dalla retta (positivo antiorario)
r..	raggio del cerchio (positivo antiorario)
pn pn'	elementi punto predefiniti di indice n e n'
[-]cn	elementi cerchio predefiniti di indice n e n'. Il verso di percorrenza del cerchio può essere cambiato mediante il segno negativo per garantire la compatibilità dei versi tra retta e cerchio nel punto di tangenza.
[-]cn'	
[-]ln	elemento retta predefinito di indice n
d..	distanza tra le due rette: positiva se la retta è a sinistra, negativa in caso contrario, guardando nella direzione della retta predefinita.



Retta passante da due punti

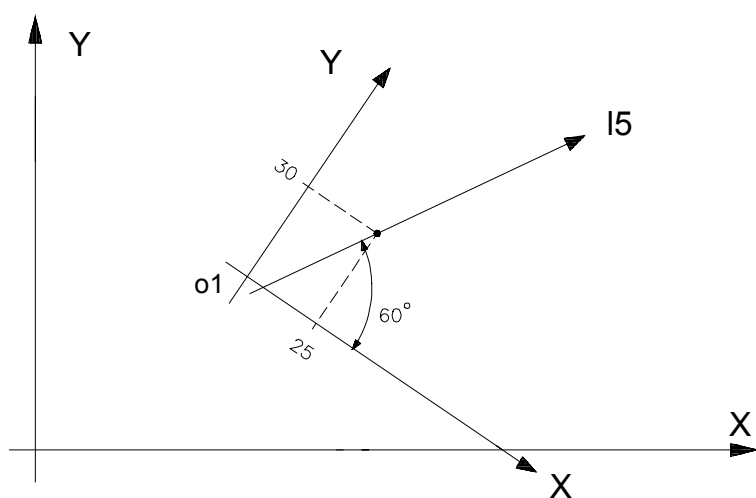
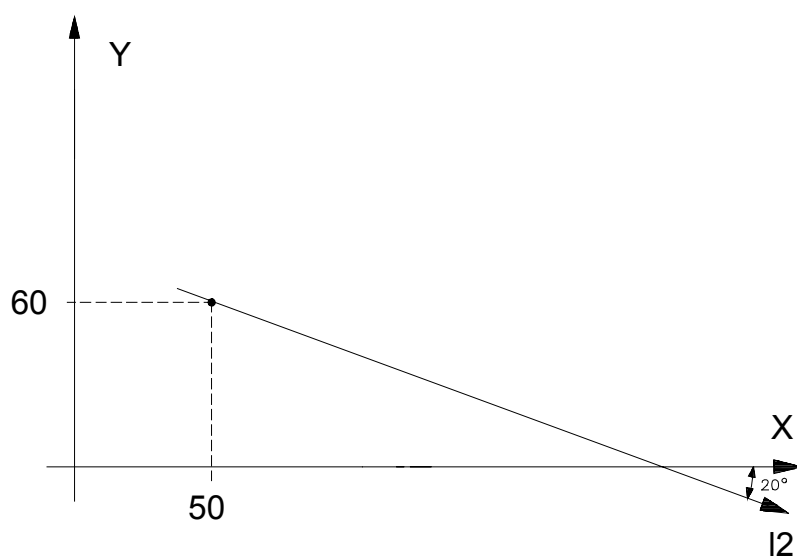
I1 = X40 Y20, X60 Y70

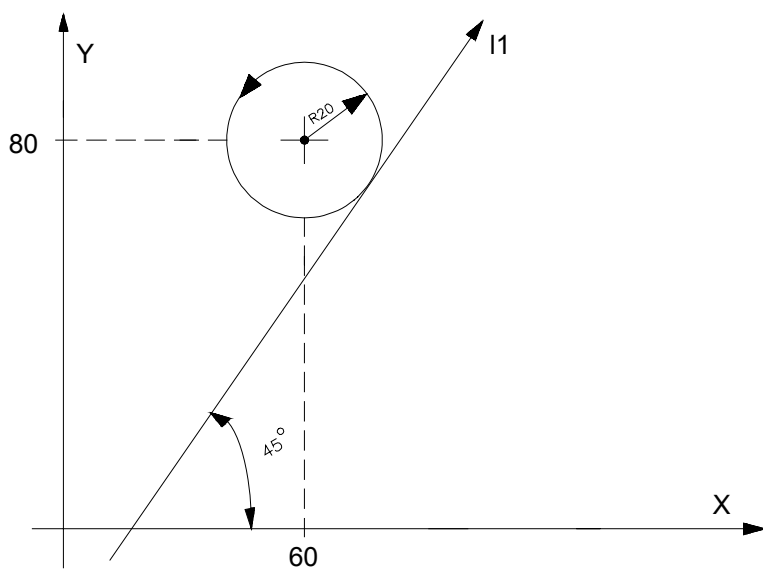


Retta tangente ad un cerchio e
passante per un punto

I1 = X10 Y15,I45 J30 r-15

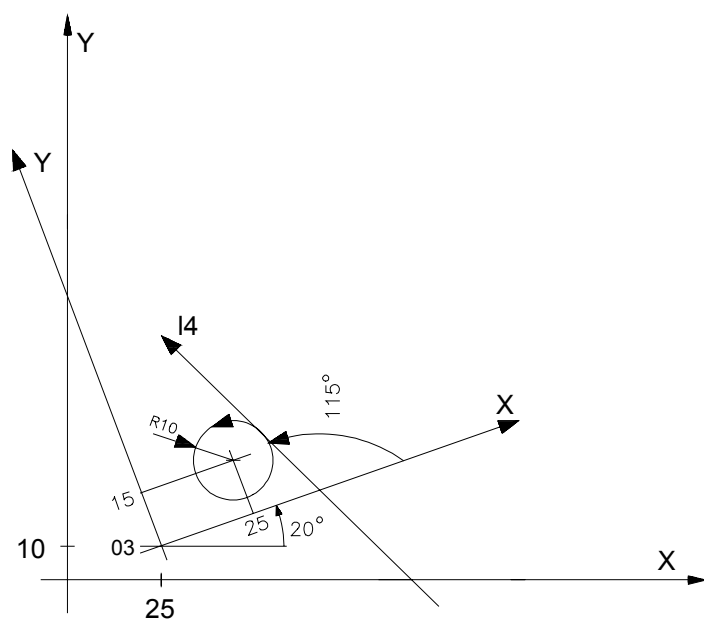
I2 = X10 Y15,I45 J30 r15





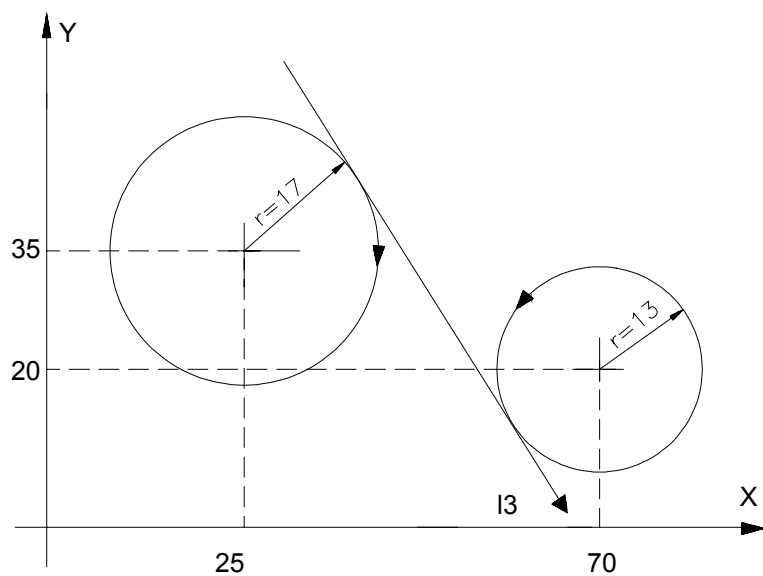
Retta tangente ad un cerchio e formante un angolo con l'asse ascissa

$I1 = I60 J80 r20, a45$



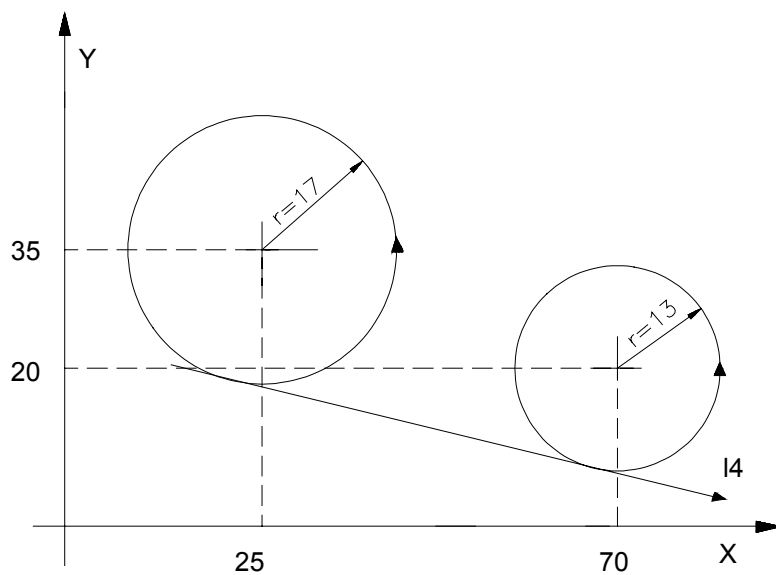
Retta tangente ad un cerchio e formante un angolo con l'asse ascissa

$I4 = o3 I25 J15 r10, a115$



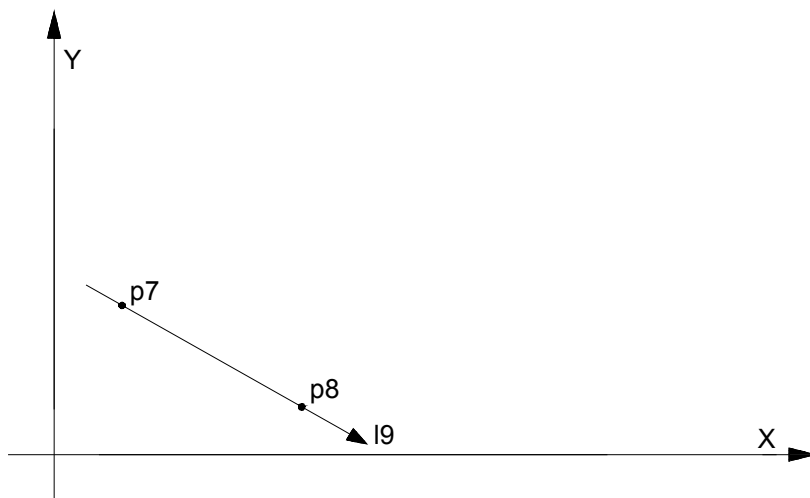
Retta tangente a due cerchi

I3 = I25 J35 r-17,I70 J20 r13



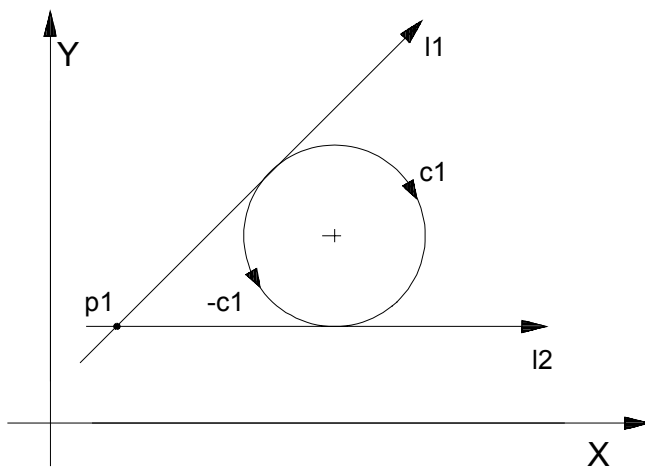
Retta tangente a due cerchi

I4 = I25 J35 r17, I70 J20 r13



Retta passante per due punti

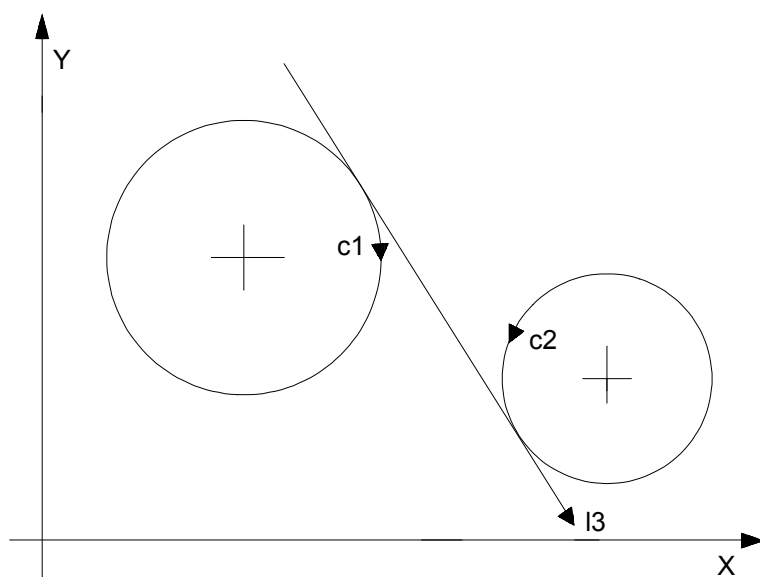
I9 = p7, p8



Retta tangente ad un cerchio e passante per un punto

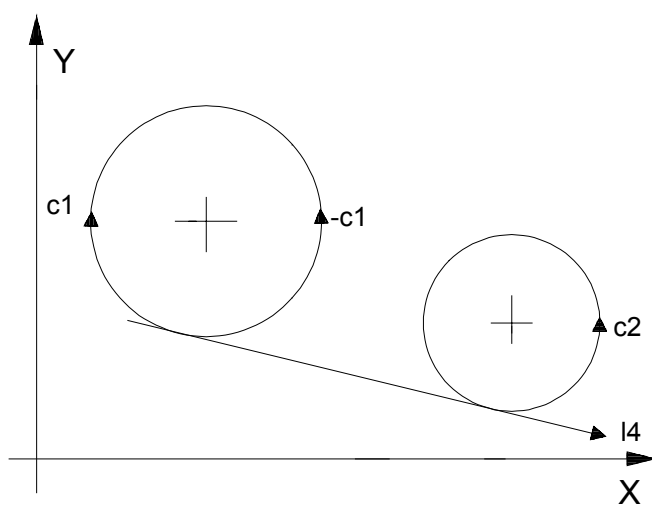
I1 = p1, c1

I2 = p1, -c1



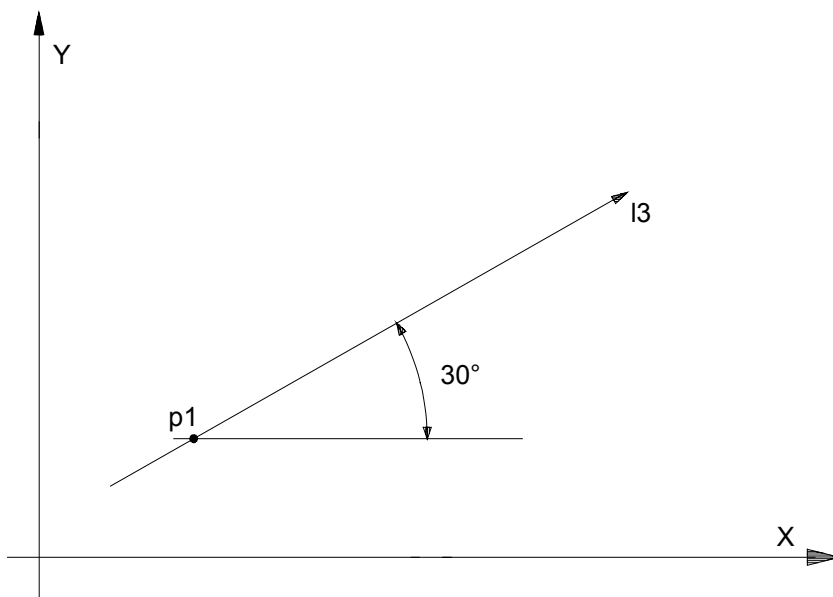
Retta tangente a due cerchi

$$I3 = c1, c2$$



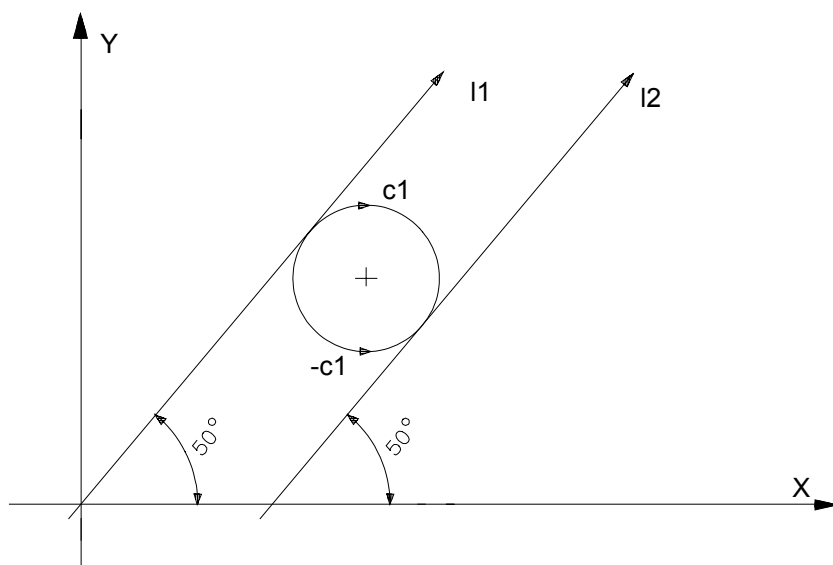
Retta tangente a due cerchi

$$I4 = -c1, c2$$



Retta passante per un punto e formante un angolo con l'asse ascissa

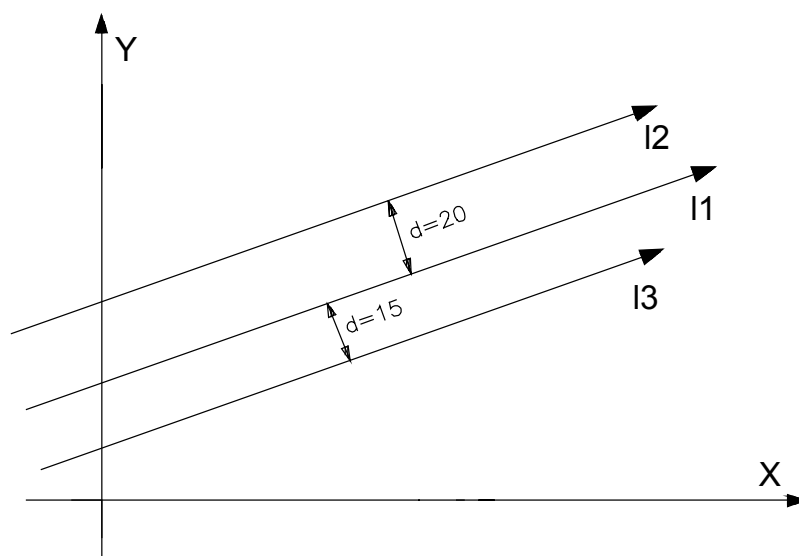
$l3 = p1, a30$



Retta tangente ad un cerchio e formante un angolo con l'asse ascissa

$l1 = c1, a50$

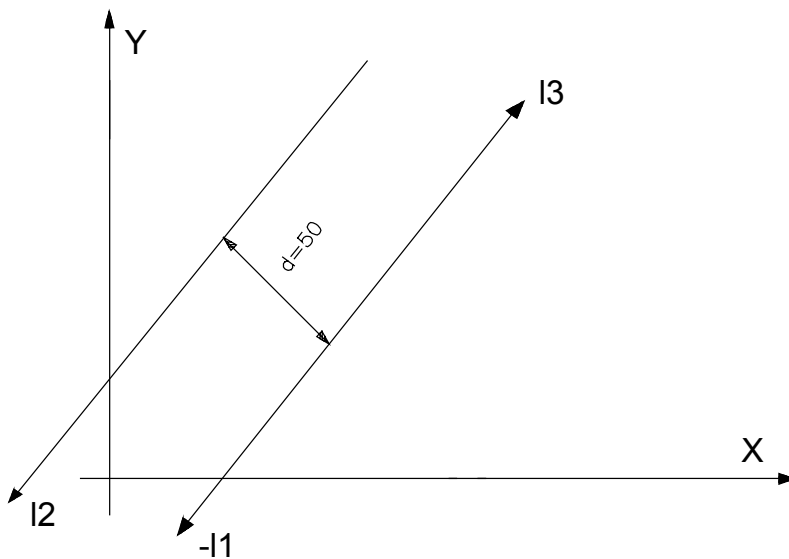
$l2 = -c1, a50$



Retta parallela ad un'altra

$$I2 = I1, d20$$

$$I3 = I1, d-15$$



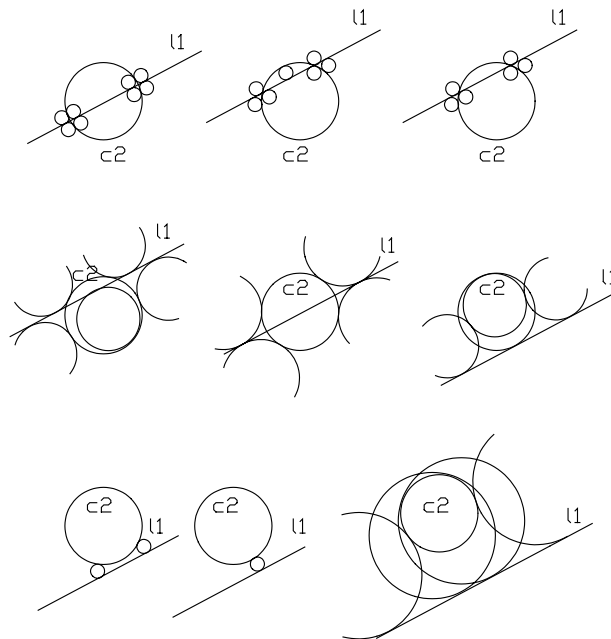
Retta parallela ad un'altra

$$I2 = -I1, d-50$$

DEFINIZIONE DI CERCHI

Funzione Definire cerchi in forma diretta (esplicita) o in forma indiretta (implicita).

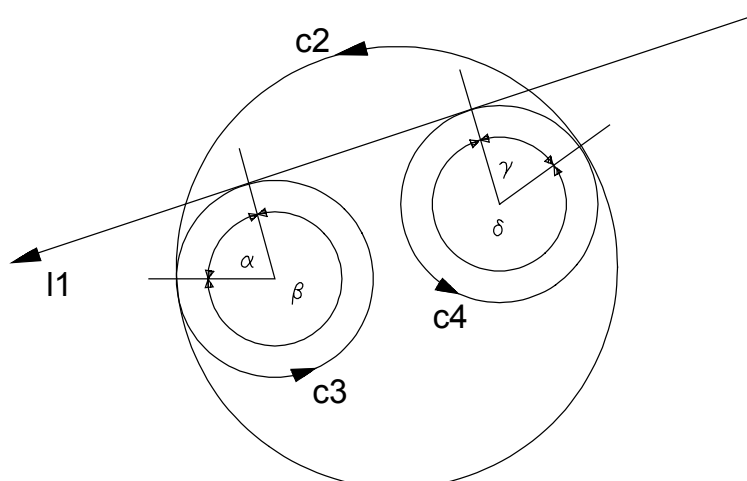
Descrizione Definendo cerchi in forma indiretta, il programmatore deve considerare la compatibilità dei versi fra gli elementi (il segno "-" può cambiare il verso degli elementi predefiniti). Non considerando i versi degli elementi, dati un cerchio e una retta, possono esistere, noto il raggio, da 1 a 8 soluzioni di cerchio tangente ai due elementi (fig. seguente).



Cerchi tangenti a una retta e un cerchio

Tenendo conto della compatibilità dei versi di percorrenza degli elementi predefiniti e del cerchio da definire, le soluzioni si riducono a due.

Per discriminare tra i due possibili cerchi, aventi lo stesso verso e lo stesso raggio, bisogna prendere in considerazione il verso degli enti cui si fa riferimento nella definizione e i due archi di circonferenza in cui l'ente che si vuole ottenere viene diviso dai punti di tangenza con gli elementi predefiniti: il GTL genera sempre il cerchio con verso che va dal primo al secondo ente ed a cui appartiene l'arco che sottende l'angolo minore (fig. seguente).

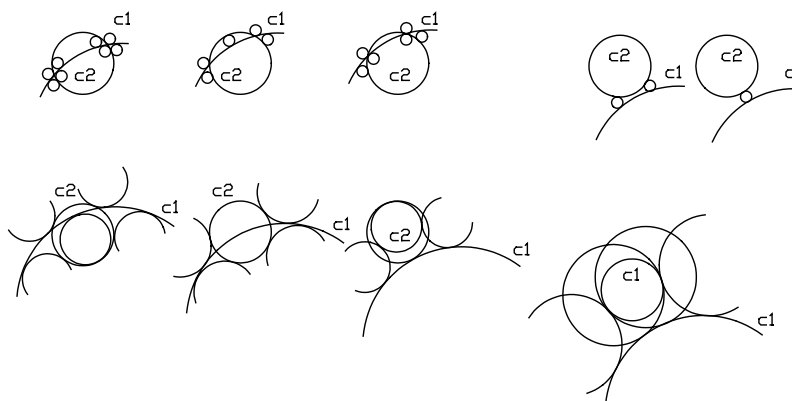


Cerchi tangenti con arco minore

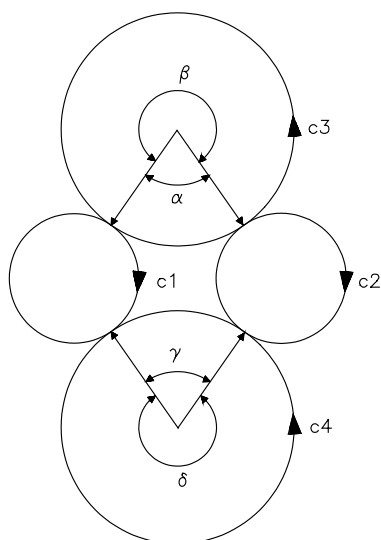
Il cerchio c3 si ottiene ponendo nella definizione la retta l1 in prima posizione ed il cerchio c2 in seconda, poiché l'ente c3 è quello che consente di muovere dalla retta l1 al cerchio c2 con arco avente angolo al centro minore.

Il cerchio c4 si ottiene ponendo nella direzione il cerchio c2 in prima posizione e la retta l1 in seconda, poiché l'ente c4 è quello che consente di muovere dal cerchio c2 alla retta l1 con arco avente angolo al centro minore.

Le stesse considerazioni vanno fatte per la definizione di cerchio tangente a due cerchi predefiniti, infatti anche in questo caso sono possibili da 1 ad 8 soluzioni (Vedi fig. seguente).



Cerchi tangenti a due cerchi predefiniti



Cerchi tangenti a due cerchi predefiniti

La compatibilità dei versi di percorrenza degli enti predefiniti e del cerchio da definire riduce a due le soluzioni possibili.

Per discriminare tra i due cerchi, aventi lo stesso verso e lo stesso raggio, si considerino i due archi in cui il nuovo ente viene diviso dal primo al secondo cerchio con arco avente angolo al centro minore (fig. sopra).

Per ottenere il cerchio c3, nella definizione bisogna mettere in prima posizione l'ente c1 e quindi l'ente c2.

Per ottenere il cerchio c4, l'ente c2 deve precedere l'ente c1 nella definizione.

Formato Diretto

Cerchio in coordinate cartesiane del centro e raggio:

$$cn = [on] I.. J.. r..$$

Cerchio in coordinate polari del centro e raggio:

$$cn = [on] m.. a.. r..$$
Formato Indiretto

Cerchio tangente a due rette predefinite e di raggio dato:

$$cn = [-] ln, ln, r..$$

Un cerchio tangente a una retta e un cerchio predefinito e di raggio dato:

$$cn = [-]ln,[-]cn,r..$$

$$cn = [-]cn,[-]ln,r..$$

Cerchio passante per un punto predefinito, tangente ad una retta predefinita e di raggio dato:

$$cn = pn,[-]ln,r..$$

$$cn = [-]ln,pn,r$$

Cerchio tangente a due cerchi predefiniti e di raggio dato:

$$cn = [-]cn,[-]cn',r$$

Cerchio passante per un punto predefinito, tangente ad un cerchio predefinito e di raggio dato:

$$cn = pn,[-]cn,r..$$

$$cn = [-]cn,pn,r..$$

Cerchio passante per due punti predefiniti e di raggio dato:

$$cn = pn,pn',r..$$

Cerchio con centro in un punto predefinito e tangenza con una retta predefinita:

$$cn = pn, [-]ln'$$

Cerchio con centro in un punto predefinito e tangenza con un cerchio predefinito:

cn = pn,[-]cn [,s2]

Cerchio passante per tre punti:

cn = pn,pn',pn''

Cerchio con centro in un punto e di raggio dato:

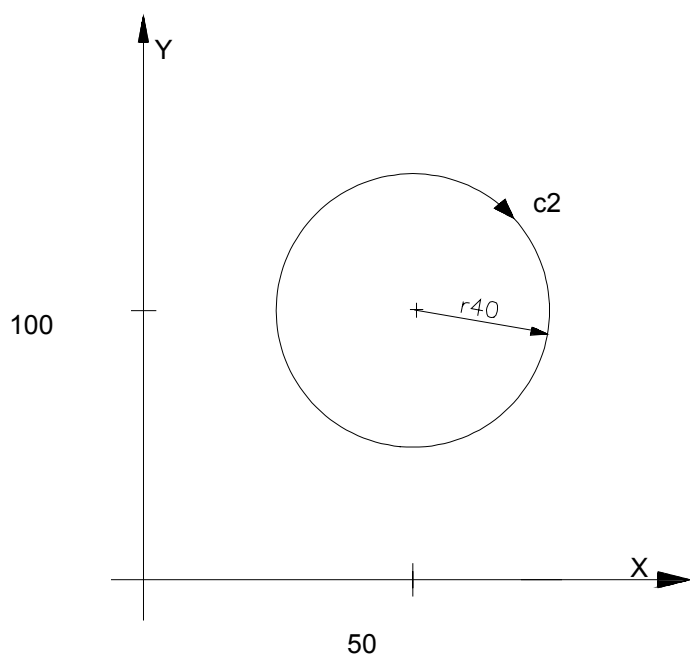
cn = pn,r..

Cerchio concentrico ad un cerchio predefinito e distante da esso di un valore dato:

cn = [-]cn,d..

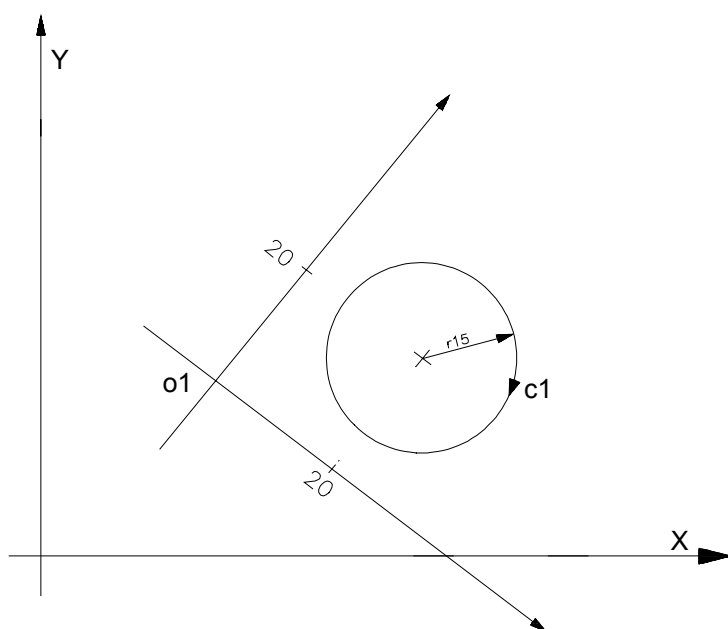
dove:

cn	identifica il nome del cerchio di indice n (n è un numero compreso tra 0 ed il numero massimo configurato).
I..J..	coordinate del centro del cerchio: I=ascissa, J=ordinata.
r..	raggio del cerchio (positivo per verso antiorario, negativo per verso orario).
[-]In [-]In'	elementi retta predefiniti di indice n e n'. Può assumere verso opposto usando il segno "-"
pn pn' pn''	elementi punto predefiniti di indice n, n' ed n''
[-]cn [-]cn'	elementi cerchio predefiniti di indice n e n'. Può assumere verso opposto usando il segno "-"
[s2]	attributo per il maggiore dei due cerchi possibili
d..	distanza tra due cerchi: è positiva se, osservando [-]cn, cn' si trova alla sua sinistra; è negativa se si trova alla sua destra.



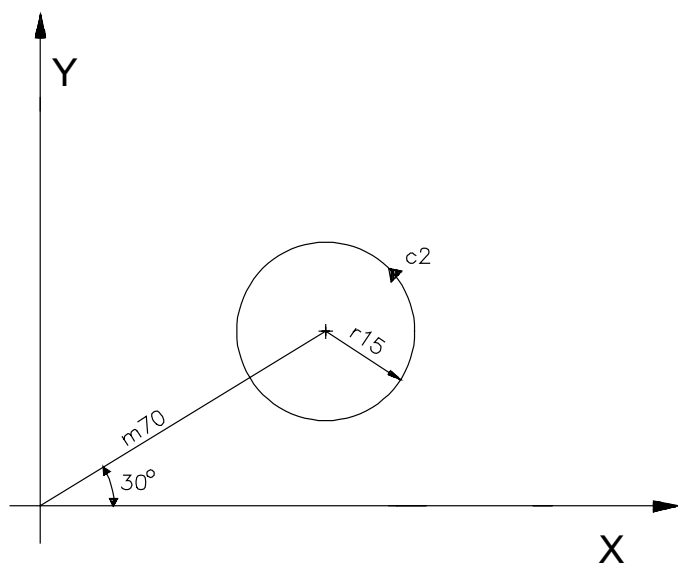
Cerchio in coordinate cartesiane
centro e raggio

$c2=I50J100r-40$



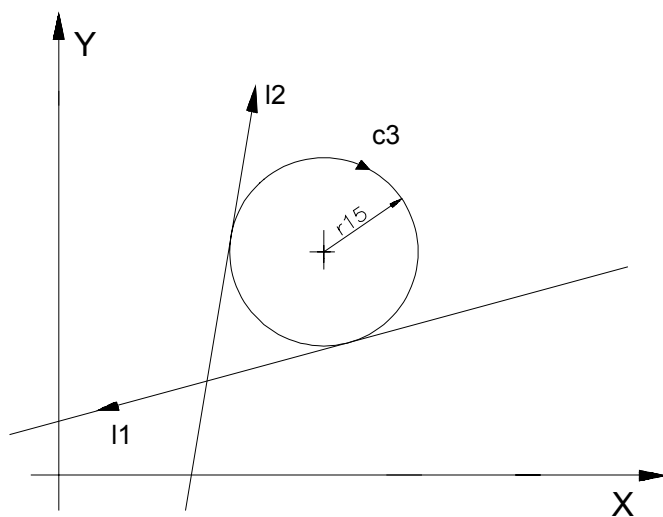
Cerchio in coordinate
cartesiane centro e raggio

$c1=o1I20J20r-15$



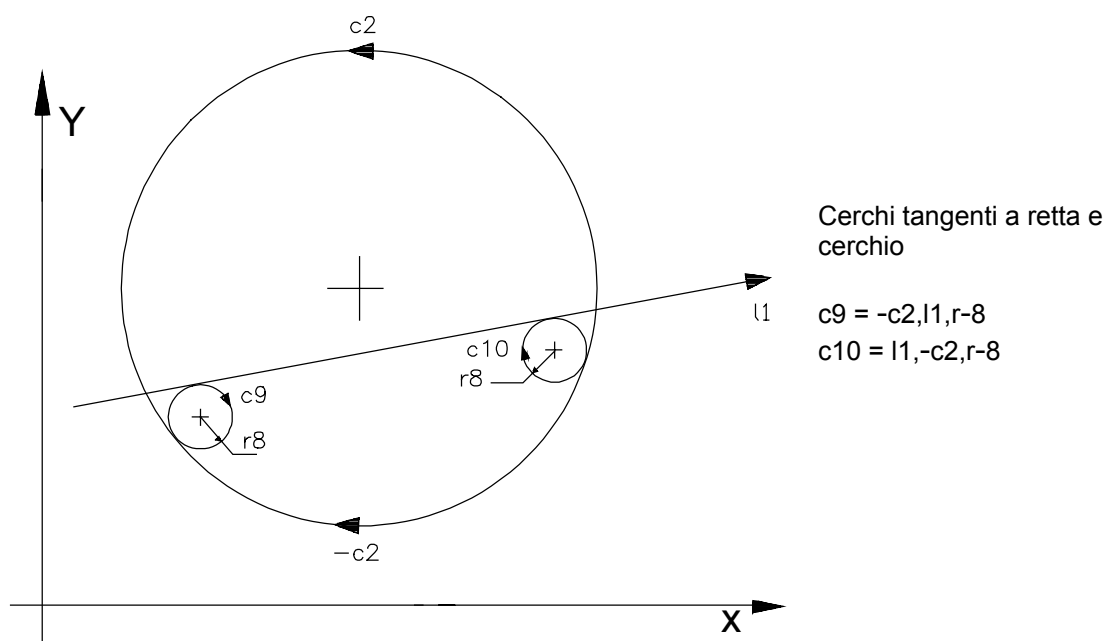
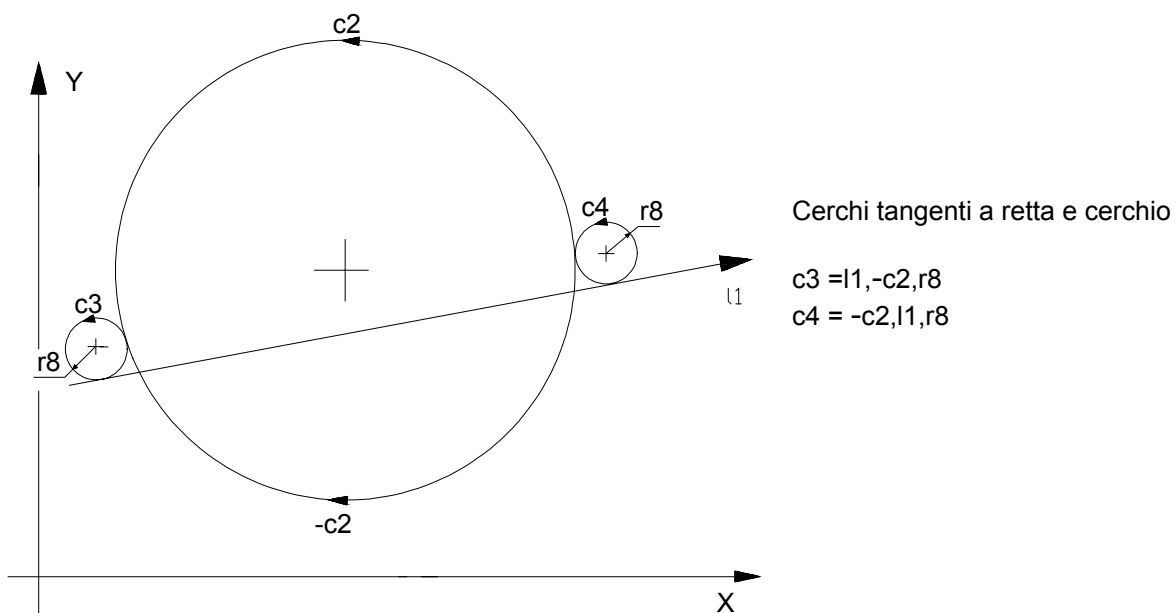
Cerchio in coordinate polari
centro e raggio

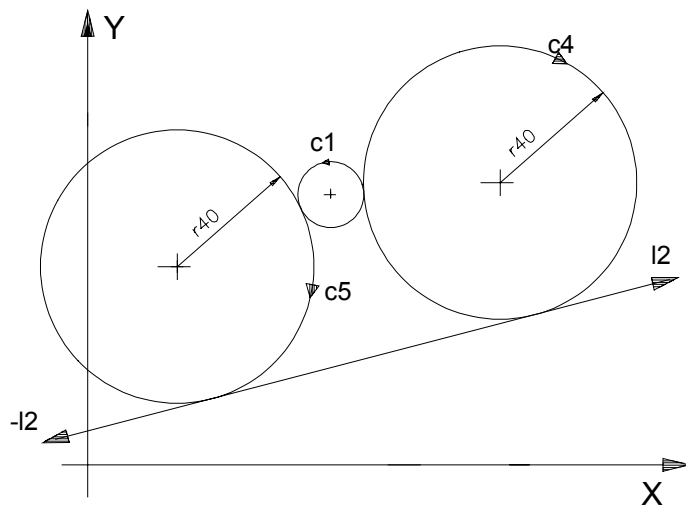
c2 = m70 a30 r15



Cerchio tangente a due rette
predefinite e di raggio r

c3 = l1,l2,r-15

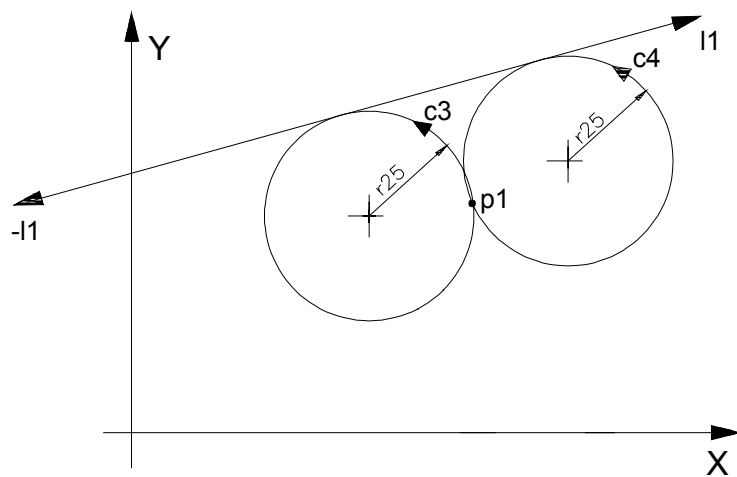




Cerchi tangenti a retta e cerchio

$$c4 = -l2, c1, r-40$$

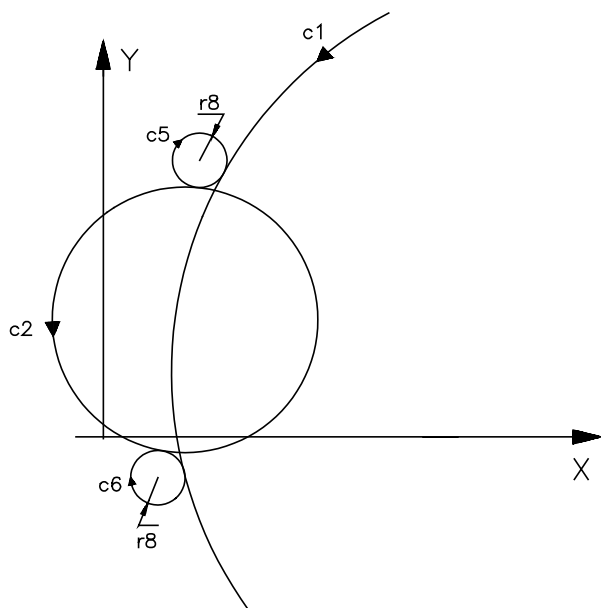
$$c5 = c1, -l2, r-40$$



Cerchi passanti per un punto e tangenti ad una retta

$$c3 = p1, -l1, r25$$

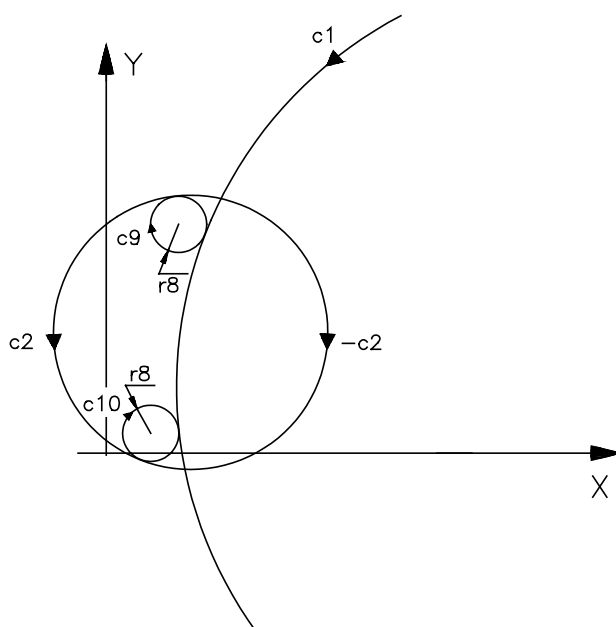
$$c4 = -l1, p1, r25$$



Cerchio tangente a due cerchi

$$c5 = c1, c2, r-8$$

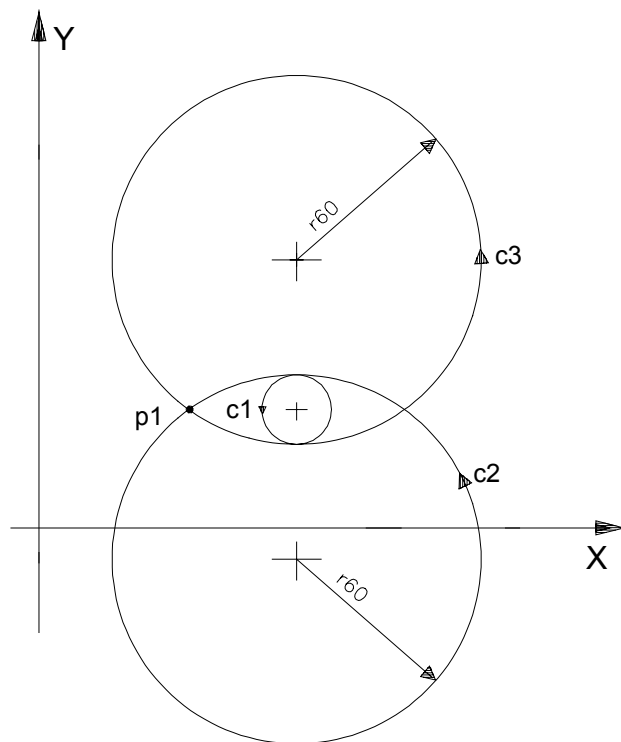
$$c6 = c2, c1, r-8$$



Cerchio tangente a due cerchi

$$c9 = -c2, c1, r-8$$

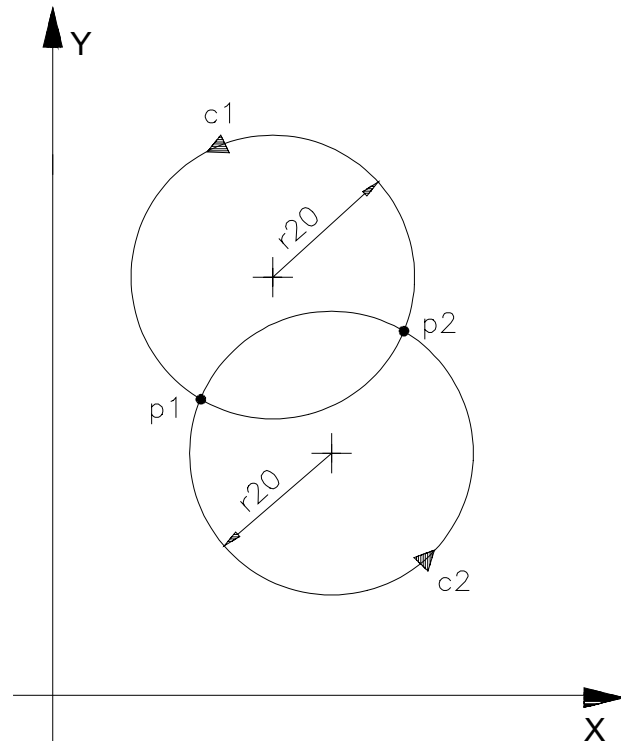
$$c10 = c1, -c2, r-8$$



Cerchio passante per un punto
e tangente ad un cerchio

$c2 = c1, p1, r60$

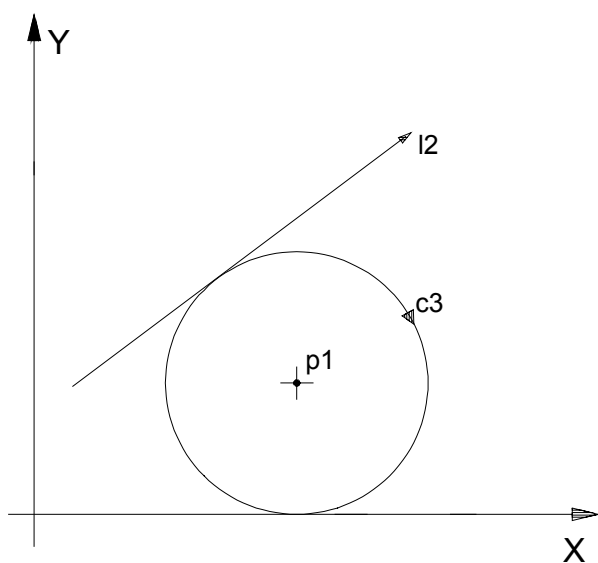
$c3 = p1, c1, r60$



Cerchi passanti per due punti

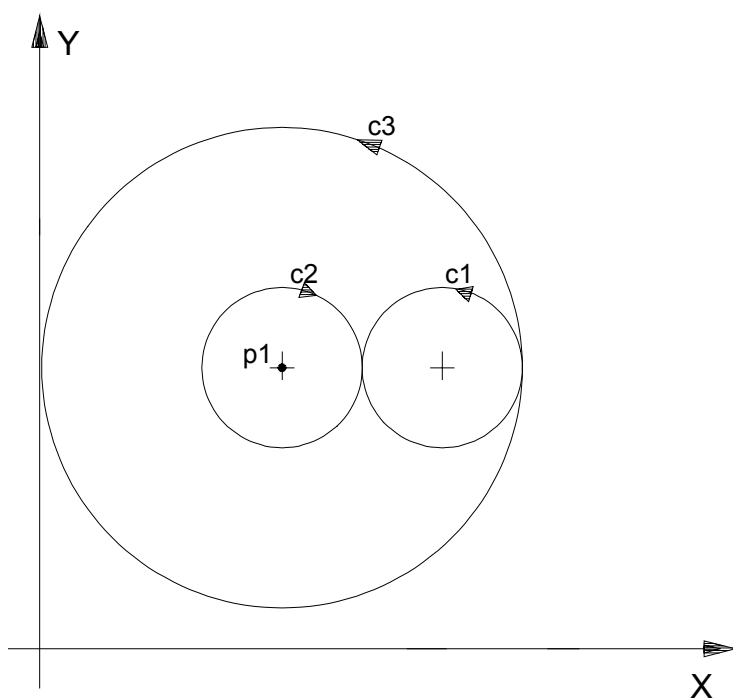
$c1 = p1, p2, r20$

$c2 = p2, p1, r20$



Cerchio con centro definito e
tangente ad una retta

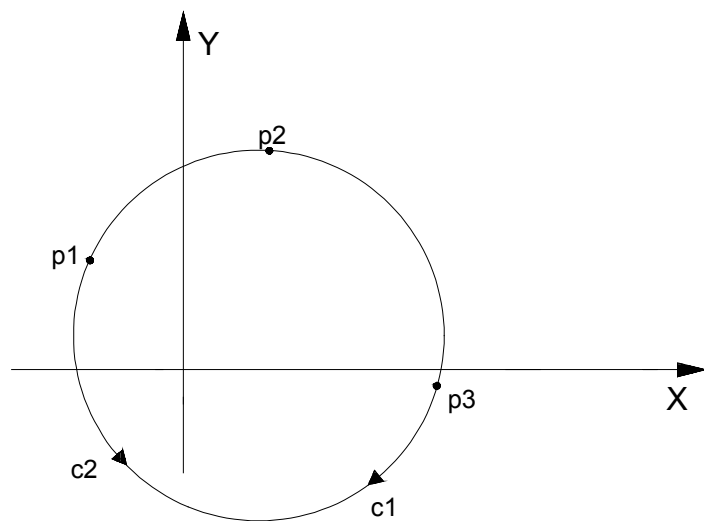
$c3 = p1, l2$



Cerchio di centro definito e
tangente ad un cerchio

$c2 = p1, c1$

$c3 = p1, c1, s2$

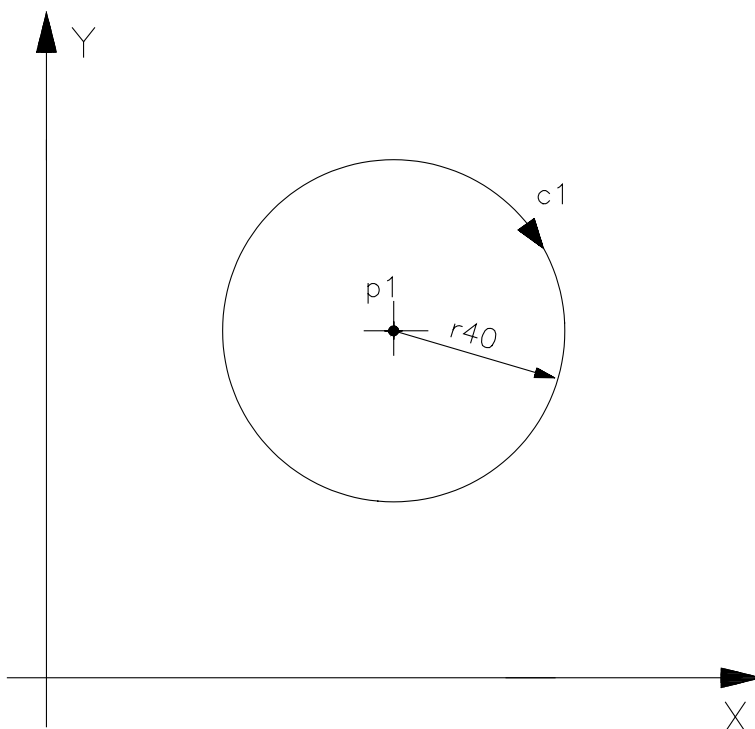


Cerchio passante per tre punti

c1 = p1,p2,p3

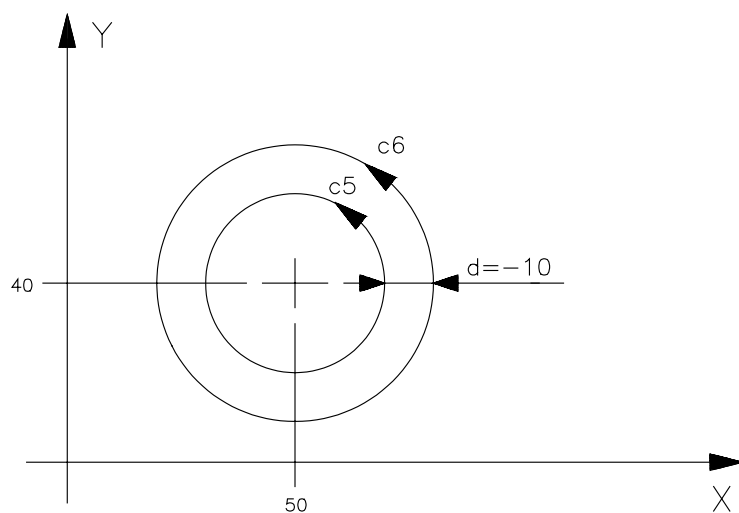
c2 = p3,p2,p1

Il verso del cerchio va dal primo al secondo e al terzo punto.



Cerchio con centro in un punto

c1 = p1,r-40



Cerchio concentrico ad un'altro

$$c6 = c5, d-10$$

DEFINIZIONE DI PROFILO

Inizio e Fine Profilo

Un programma in geometria GTL è definito dalle funzioni G21 e G20.

G21 identifica l'inizio del profilo

G20 identifica la fine del profilo

Tipi di profilo

Si intende per profilo una successione di enti geometrici precedentemente memorizzati. Il profilo può essere aperto o chiuso.

Profilo aperto

Nel caso di profilo aperto questo deve iniziare con un punto (pn) e finire con un punto diverso dal primo.

La correzione raggio utensile agisce in modo normale al primo ente sul punto di inizio profilo e normale all'ultimo ente sul punto finale.

La correzione raggio va aperta sul primo punto del profilo programmando nel blocco le funzioni G21 G41/G42 e chiusa sull'ultimo con le funzioni G20 G40.

Profilo chiuso

Se il profilo è chiuso prima occorre programmare l'ultimo ente, e dopo l'ultimo ente occorre richiamare il primo ente del profilo.

il primo punto del profilo offsettato è l'intersezione del primo ed ultimo ente traslati (primo punto = ultimo punto).

La correzione raggio va aperta a inizio profilo nel blocco di richiamo dell'ultimo ente programmando le funzioni G21 G41/G42 e chiusa a fine profilo nel blocco di richiamo del primo ente con le funzioni G20 G40.

Se il primo e/o ultimo ente sono cerchi le intersezioni possibili sono due.

Se non si danno informazioni aggiuntive il sistema sceglie la prima.

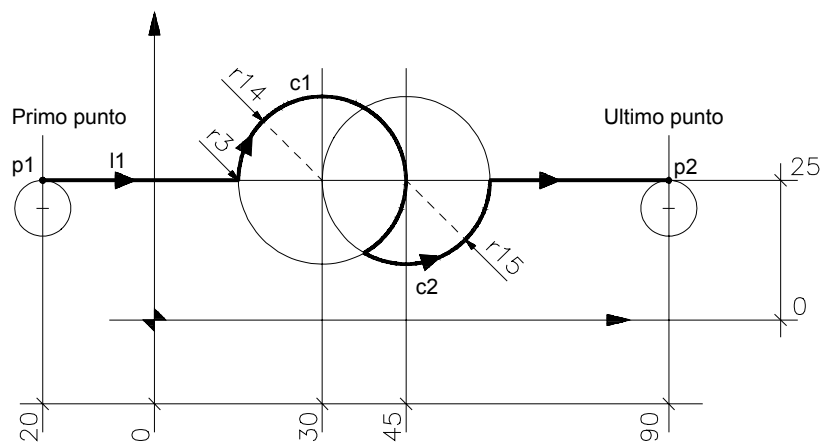
Nel caso in cui si voglia la seconda intersezione, occorre programmare il discriminatore s2. Il discriminatore s2 va programmato nel blocco di richiamo dell'ultimo ente a inizio profilo, e sull'ultimo ente a fine profilo.

Funzioni G accettate

Le uniche funzioni G accettabili all'interno del profilo sono G27, G28, G04, G09, G41, G42.

IMPORTANTE

Se è programmato pn nel blocco di inizio profilo, significa che il profilo è aperto; pn può essere programmato all'inizio ed alla fine del profilo, ma non all'interno di esso.

Profilo aperto

Esempio di profilo aperto

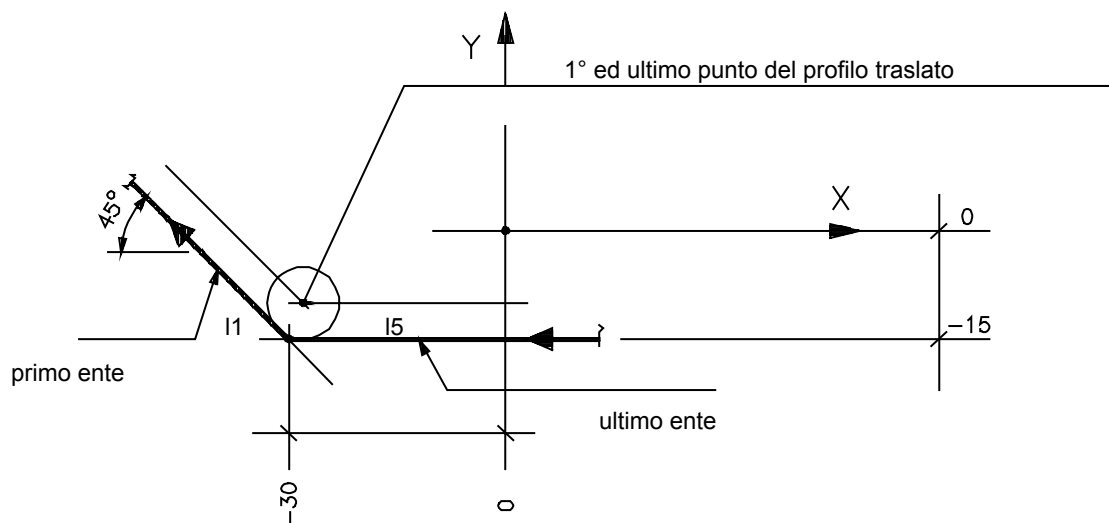
.....
 I1 = X Y25,a
 p1 = X-20 Y25
 p2 = X90 Y25
 c1 = I30 J25 r-14
 c2 = I45 J25 r15

.....
 G21 G42 p1 - primo punto
 I1
 r3
 c1 s2
 c2 s2
 I1
 G20 G40 p2 - ultimo punto

IMPORTANTE

La correzione raggio va aperta sul primo punto del profilo e chiusa sull'ultimo.
 La correzione raggio è disattivata nel primo blocco di movimento degli assi del piano seguente la funzione G40.

Profilo Chiuso



Esempio di profilo chiuso retta-retta

.....

I5 = X Y-15,a180

.....

I1 = X-30 Y-15,a135

.....

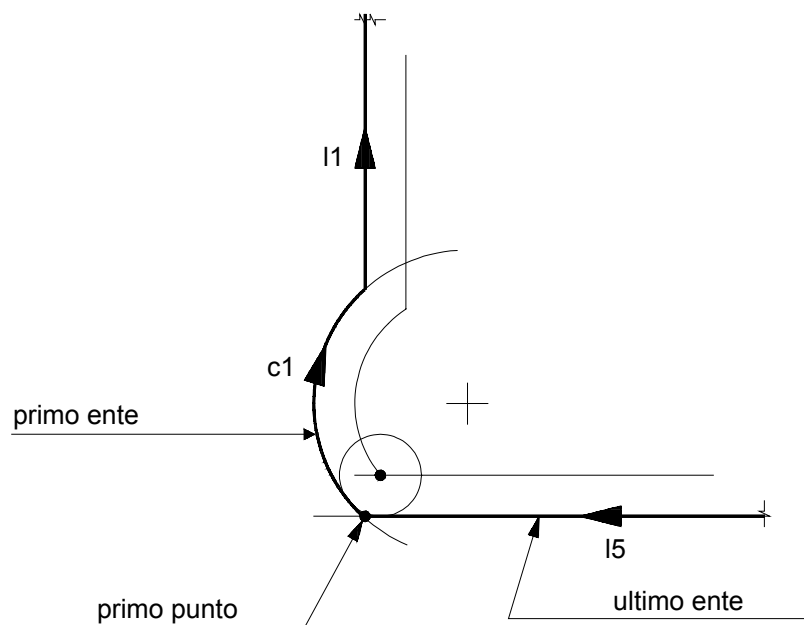
G21 G42 I5 - ultimo ente

I1 - primo ente

.....

I5 - ultimo ente

G20 G40 I1 - primo ente



Esempio di profilo chiuso retta-cerchio

c1 = I.. J.. r

l1 = X.. Y.. a90

l5 = X.. Y..,a180

G21 G42 l5 s2 - ultimo ente

c1 s2 - primo ente

l1

l5 s2 - ultimo ente

G40 c1 - primo ente

IMPORTANTE

La correzione raggio va aperta ad inizio profilo nel blocco di richiamo dell'ultimo ente e chiusa a fine profilo nel blocco di richiamo del primo ente. La correzione raggio è disattivata nel primo blocco di movimento degli assi del piano seguente la funzione G40.

Movimento dell'asse Mandrino

Su qualunque punto del profilo è possibile muovere gli assi non interessati alla contornitura anche sul primo punto, ad esempio per penetrare nel pezzo.

Sul primo punto, in caso di profili aperti il movimento va programmato dopo la programmazione del punto, nel caso di profili chiusi va programmato tra la definizione dell'ultimo ente di supporto ed il primo ente.

Esempio:

```
.....  
G21G42 p1  
Z-10  
I1  
.....
```

```
.....  
G21G42 I5  
Z-10  
I1  
.....
```

Collegamento degli enti

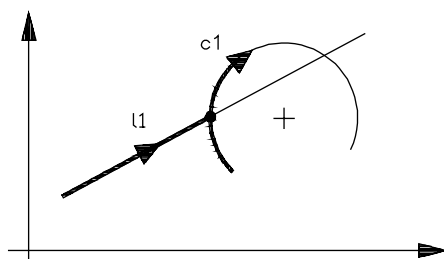
Gli enti geometrici del profilo possono essere collegati fra loro da condizioni di tangenza, intersezione o dalla presenza di un raccordo automatico, o di uno smusso.

1. Intersezione fra enti

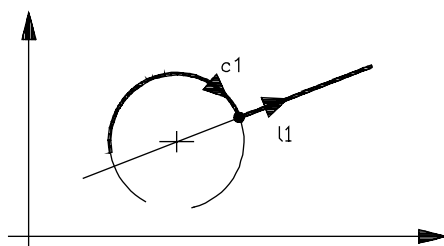
Nel caso di intersezione fra due rette non vi è ambiguità, essendo unica la soluzione.

Nel caso di intersezione retta-cerchio normalmente l'unità sceglie automaticamente la prima intersezione.

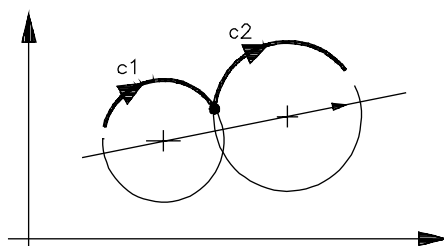
Se invece si vuole selezionare la seconda intersezione, occorre programmare il discriminatore s2 dopo la definizione del primo ente.



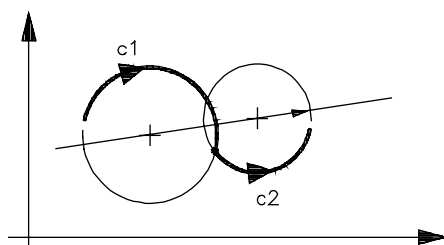
N24 l1
N25 c1



N28 c1 s2
N29 l1



N33 c1
N34 c2

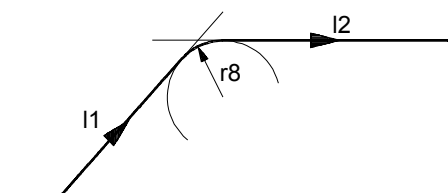


N36 c1 s2
N37 c2

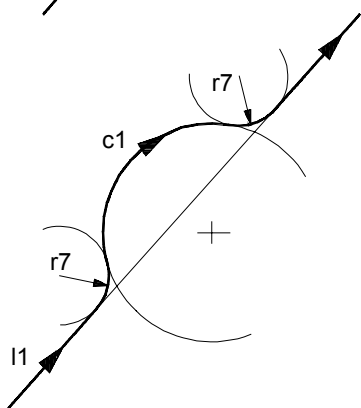
Nel caso di intersezione fra retta e cerchio la prima e seconda intersezione vanno viste nel senso di percorrenza della retta. Nel caso di intersezione cerchio-cerchio la prima intersezione è quella che sta a sinistra della retta che unisce il centro del primo cerchio con quello del secondo, la seconda intersezione è quella che sta a destra di tale retta.

2. Raccordi automatici fra enti

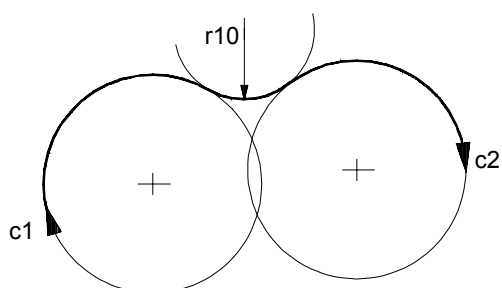
E' possibile definire cerchi di raccordo fra i vari enti, programmando semplicemente il valore del raggio segnato, positivo se percorso in senso antiorario, negativo se percorso in senso orario.



.....
N20 I1
N21 r-8
N22 I2
.....



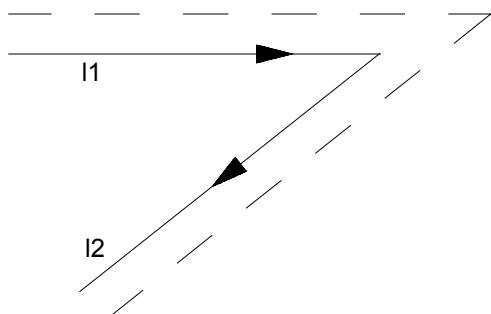
.....
N24 I1
N25 r7
N26 c1
N27 r7
N28 I1
.....



.....
N90 c1
N91 r10
N92 c2
.....

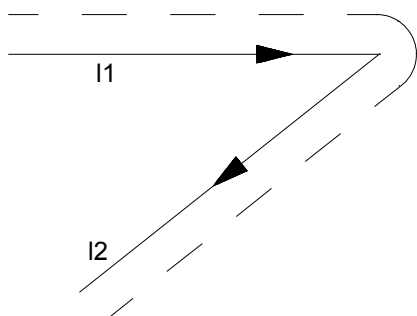
Raccordo automatico fra enti

Nel caso di correzione raggio attiva, l'utensile viene posizionato sull'intersezione dei due elementi geometrici traslati del raggio utensile. Se si vuole l'introduzione di un raccordo occorre programmare un raccordo di raggio 0.



Senza raccordo

.....
L1
L2
.....



Raccordo r0

.....
L1
r0
L2
.....

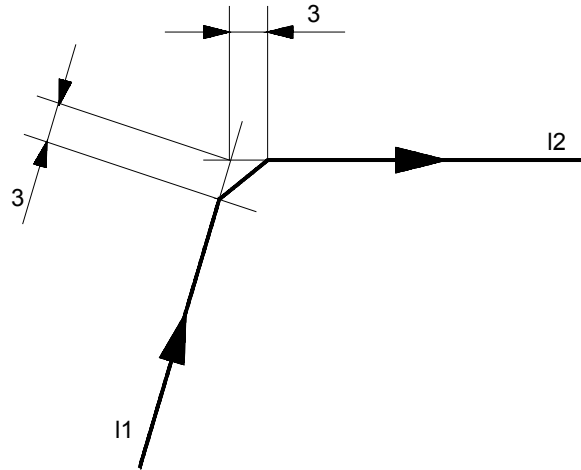
Raccordo zero con intersezione e correzione utensile

IMPORTANTE

Il raccordo r non può essere programmato nel blocco immediatamente successivo a quello con G21 o in quello precedente al G20 (cioè un profilo non può iniziare o finire con un raccordo).

3. Smussi

E' possibile definire smussi fra enti rettilinei programmando semplicemente il valore dello smusso (b) senza segno, inteso come distanza dal punto di intersezione.



```
N30 I1
N31 b3      Smusso
N32 I2
```

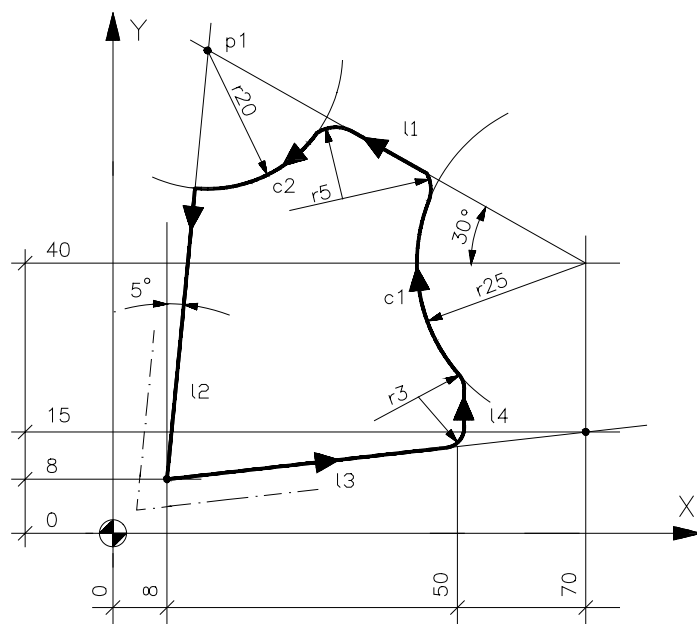
NOTE:

- Lo smusso b non può essere programmato nel blocco immediatamente successivo a quello con G21 od in quello precedente al G20 (cioè un profilo non può iniziare o finire con uno smusso).
- In programmazione geometrica GTL, gli spostamenti sono sempre effettuati a velocità di lavoro; per programmare uno spostamento in rapido occorre programmare una velocità F molto alta.
- Se il piano di interpolazione non è quello formato dagli assi X ed Y, occorre definire il piano e quindi memorizzare gli enti considerando rispettivamente come ascissa ed ordinata gli assi del piano definito.

Esempio:

```
N1      G16BY
N2      I1 = B70,Y40,a150
N3      I2 = B8,Y8,a-95
N4      p1, = I1,I2
N5      I3 = B8,Y8,B70,Y15
N6      c1 = I70,J40,r-25
.....
.....
N12     G21 G42 I2
.....
```

ESEMPI DI PROGRAMMAZIONE GTL

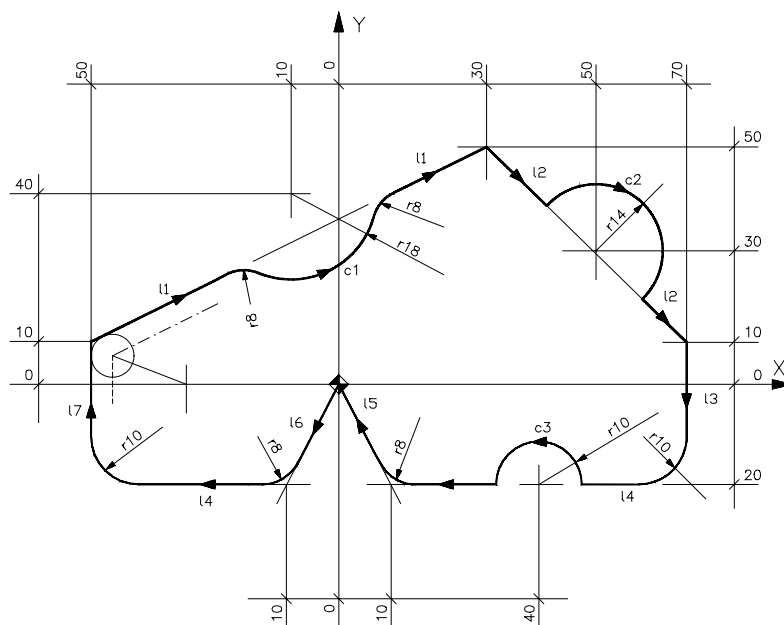


Esempio 1

```

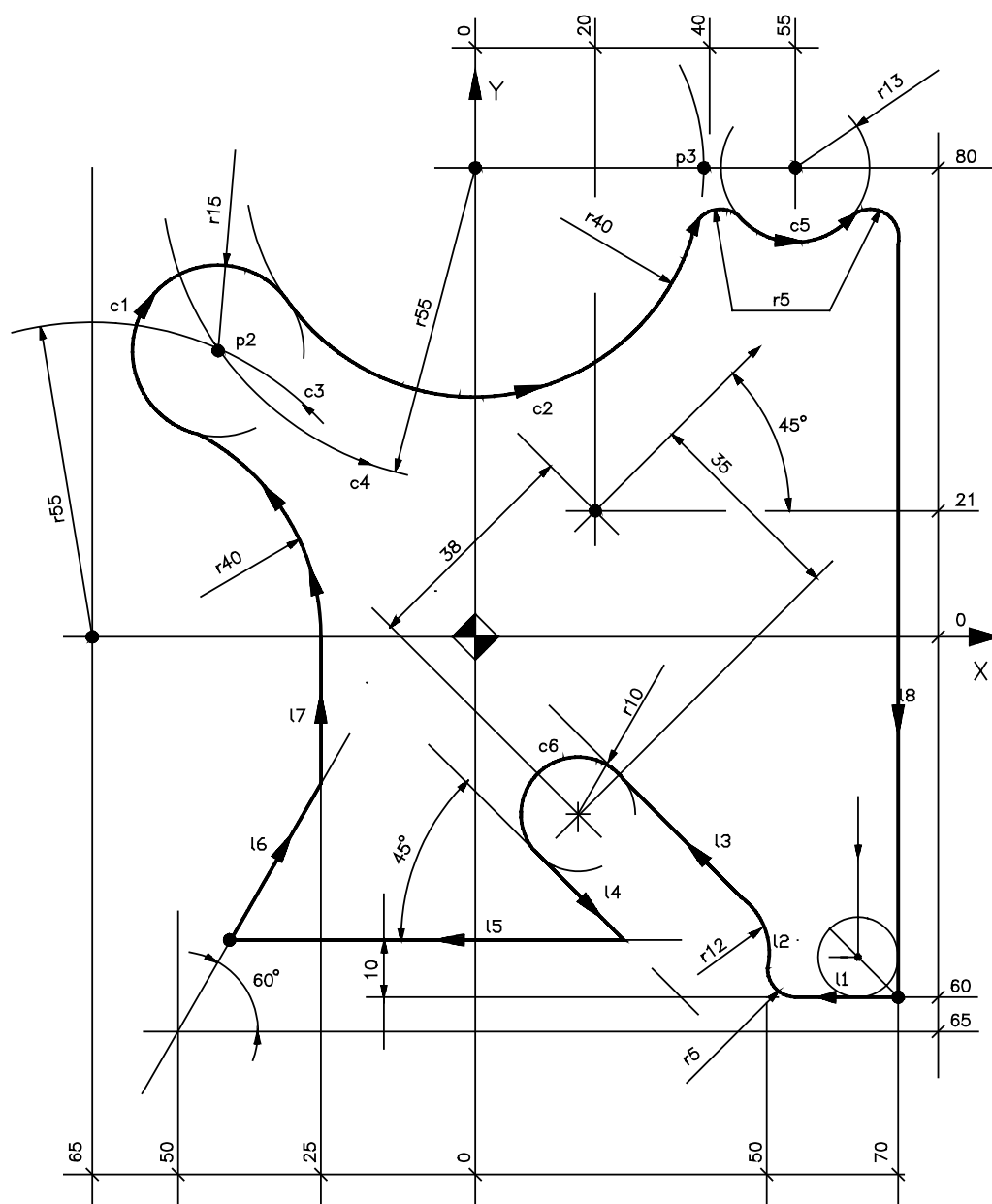
N1 (DIS,"ESEMPIO GTL")
N2 l1=X70Y40,a150
N3 l2=X8Y8,a-95
N4 p1=l1,l2
N5 l3=X8Y8,X70Y15
N6 l4=X50Y,a90
N7 c1=I70J40 r-25
N8 c2=p1,r-20
N9 F250 S800 T1.1 M6M3
N10 GXY
N11 Z-10
N12 G21G42I2
N13 I3
N14 r3
N15 I4
N16 r3
N17 c1
N18 r5
N19 I1
N20 r5
N21 c2s2
N22 I2
N23 XY M30
N24 G20 G40 I3
N25 GZ2

```



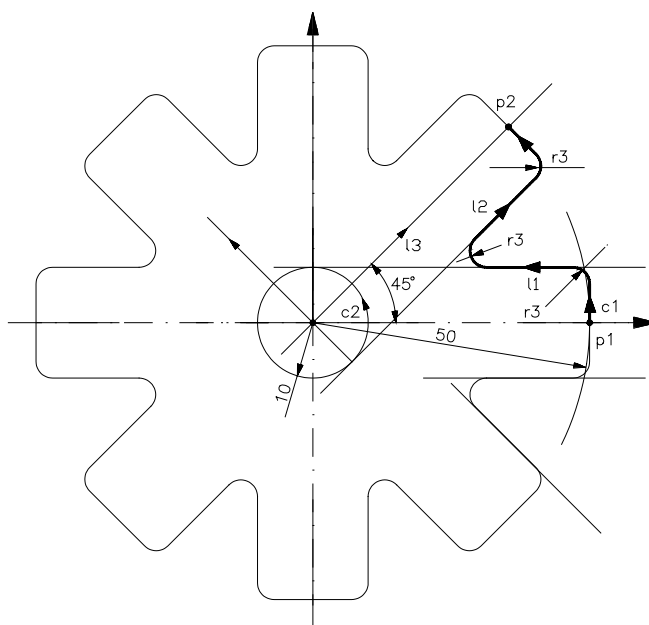
Esempio 2

N1	(DIS,"ESEMPIO GTL")	N20	I1
N2	I1=X-50Y10,X30Y50	N21	I2
N3	I2=X30Y50,X70Y10	N22	c2s2
N4	I3=X70Y0,a-90	N23	I2
N5	I4=X=Y-20,a180	N24	I3
N6	I5=X10Y-20,X0Y0	N25	r-10
N7	I6=X0Y0,X-10Y-20	N26	I4
N8	I7=X-50Y0,a90	N27	c3s2
N9	c1=I-10J40r18	N28	I4
N10	c2=I50J30r-14	N29	r-8
N11	c3=I40J-20r10	N30	I5
N12	S...M...T3.3M6M....	N31	I6
N13	G0X-30Y0	N32	r-8
N14	Z-10	N33	I4
N15	G21G42I7	N34	r-10
N16	I1	N35	I7
N17	r-8	N36	G20G40I1
N18	c1	N37	G0Z20
N19	r-8		



Esempio 3

N1 (DIS,"ESEMPIO 3")
N2 S...F...T1.1M6 M...
N3 o1=X20 Y21 a45
N4 l1=X0 Y-60,a180
N5 l2=X50 Y0,a90
N6 c6=o1 l-38 J-35 r10
N7 l3=c6,a135
N8 l4=c6,a-45
N9 l5=X0 Y-50,a180
N10 l6=X-50 Y-65,a60
N11 l7=X-25 Y0,a90
N12 c3=l-65 J0 r55
N13 c4=l0 J80 r55
N14 p2=c3,c4
N15 c1=p2,r-15
N16 p3=X40 Y80
N17 c2=c1,p3,r40
N18 c5=l55 J80 r13
N19 l8=X70 Y0,a-90
N20 G21 G42 l8
N21 Z-10
N22 l1
N23 r-5
N24 l2
N25 r12
N26 l3
N27 c6
N28 l4
N29 l5
N30 l6
N31 l7
N32 r40
N33 c1
N34 c2
N35 r-5
N36 c5
N37 r-5
N38 l8
N39 G20 G40 l1
N40 G Z20
N41 X... Y... M30



Esempio 4: Profilo ripetuto 8 volte (eseguito in due passate)

```

N1 (DIS,"ESEMPIO GTL CON ROTAZIONE")
N2 F...S...T2.2 M6
N3 MSA=2
N4 p1=X50 Y0
N5 c1=I0 J0 r50
N6 c2=I0 J0 r10
N7 l1=c2,a180
N8 l3=X0 Y0,a45
N9 l2=c2,a45
N10 p2=l3,c1,s2
N11 GX60 Y0
N12 Z-2
"START" N13
N14 E25=0
N15 (RPT,8)
N16 (ROT,E25)
N17 G21 G42 p1
N18 c1
N19 r3
N20 l1
N21 r-3
N22 l2
N23 r3
N24 c1
N25 G20 G40 p2
N26 E25=E25+45
N27 (ERP)
N28 (ROT,0)
"END" N29
N30 MSA=0
N31 (EPP,START,END)
N32 GZ20
N33 XY M30

```

FINE CAPITOLO

CICLI DI LAVORAZIONE PER SISTEMI DI TORNITURA

PROGRAMMAZIONE DEL PROFILO

Un profilo è un programma composto da blocchi di movimento che deve essere richiamato da un ciclo di sgrossatura o di finitura. Ci sono tre metodi per programmare i profili:

1. Scrivere il profilo in linguaggio ISO utilizzando l'editor di linea

```
G1Z0X30
Z-15
Z-20.80127X50.09618
G3 Z-2629165X61.07694 I-33.79165 J35.09618
G1 Z-55.82309 X95.17692
G3 Z-64.82309 X100 I-64.82309 J64
G1 Z-76
X140
```

2. Scrivere il profilo in linguaggio GTL utilizzando l'editor di linea

```
p1=Z0 X40
l1=p1, a180
c1=l-40 J40
l2=c1, a180
p2=Z-55 X100
l3=p2, a90
G21 p1
l1
r-5
c1
l2
r-5
l3
G20 p2
```

3. Scrivere il profilo con l' editor grafico. In questo caso il profilo viene costruito con il supporto della geometria diretta disponibile a livello di editor e quindi il programma verrà tradotto in uscita nel linguaggio ISO.

Limiti imposti nella definizione di un profilo da richiamare nelle macro-istruzioni di sgrossatura/finitura.

- Il profilo deve essere interamente descritto in una subroutine, indicata nelle pagine seguenti come "nome_profilo".
- I blocchi contenuti nella subroutine dovranno essere strettamente di tipo ISO o GTL, sarà ammessa la programmazione di:
 - Nomi asse [quote]
 - Funzioni G di movimento (G1, G2, G3 ed eventuali loro operandi)
 - NON saranno quindi presi in considerazione i comandi triletterali in generale e comunque tutti quelli che potrebbero modificare la descrizione del profilo stesso (origini, mirror, fattore di scala)

NOTA:

Il non rispetto dei punti elencati, potrà provocare una errata interpretazione del profilo descritto nella subroutine.

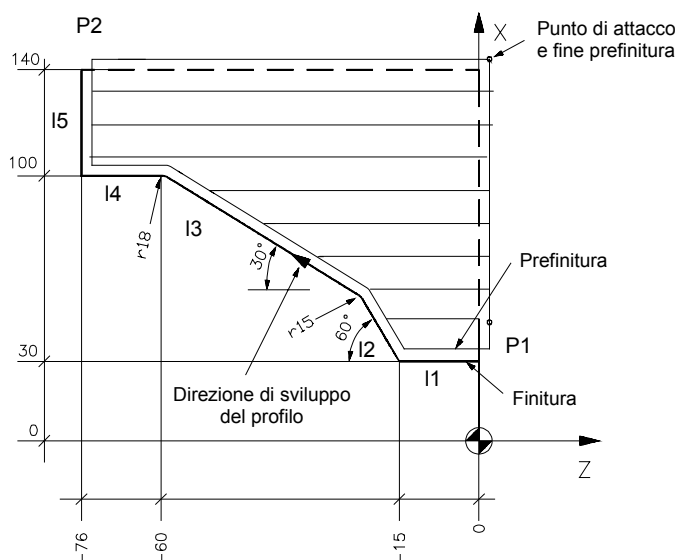


Figura 16-1 Sgrossatura parassiale con prefinitura e finitura

T1.1 M6 S.. F..
 G X143 Z1.5
 (SPF,PROF2,Z,9,X1,Z1)
 X300 Z200
 T2.2 M6 S.. F..
 G X30 Z2
 (CLP,PROF2)
 G X300 Z200

PROF2

p1 = Z0X30
 I1 = p1, a180
 I2 = Z-15X30, a120
 I3 = Z-60X100, a150
 I4 = Z-75X100, a180
 p2 = Z-75X140
 I5 = p2, a90
 G21 p1
 I1
 I2
 r15
 I3
 r18
 I4
 I5
 G20 p2

PROGRAMMAZIONE DI CICLI SPECIALI

I seguenti codici trilettari consentono di definire e richiamare cicli speciali all'interno del programma:

SPA	sgrossatura parassiale senza prefinitura
SPF	sgrossatura parassiale con prefinitura
SPP	sgrossatura parallela al profilo
CLP	ciclo di finitura
FIL	ciclo di filettatura
TGL	ciclo di taglio gole.

MACROISTRUZIONI DI SGROSSATURA PARASSIALE SENZA PREFINITURA

Sgrossatura di un profilo, con passate di lavoro parallele all'asse ordinata o ascissa:

(SPA , nome_profilo , nome_asse , n_pass. [, asse1 sovram] [,asse2 sovram])

Esempio: (SPA , PROF1, Z , 6 , X1 , Z2)

dove:

nome_profilo	è una stringa ASCII che rappresenta il nome del part program nel quale è descritto il profilo.
nome_asse	indica il nome dell'asse ascissa o ordinata (Z o X) lungo il quale eseguire le passate di sgrossatura o lasciare il sovrametallo programmato
n_pass	indica il numero di passate di sgrossatura. Può assumere un valore compreso tra 1 e 255.
asse1 sovram	indica il sovrametallo da lasciare sull'asse1 (tipicamente X). Il valore deve essere positivo.
asse2 sovram	indica il sovrametallo da lasciare sull'asse2 (tipicamente Z). Il valore deve essere sempre positivo.

In base al punto di attacco e alla direzione del profilo, il controllo decide automaticamente se la sgrossatura si riferisce a un interno o a un esterno e quindi attribuisce il segno opportuno ai sovrametalli.

Il punto di attacco deve essere esterno al campo di sgrossatura almeno quanto il valore programmato di sovrametallo.

Se il profilo non è monotono, cioè contiene delle cavità, durante l'esecuzione del ciclo di sgrossatura queste vengono automaticamente sorvolate.

IMPORTANTE

Il profilo deve essere monotono per l'asse di sgrossatura. In caso contrario il sistema segnala errore.

Il profilo, cioè, può presentare delle cavità, ma per quanto riguarda l'andamento delle quote lungo l'asse di sgrossatura, esso deve essere sempre crescente o sempre decrescente.

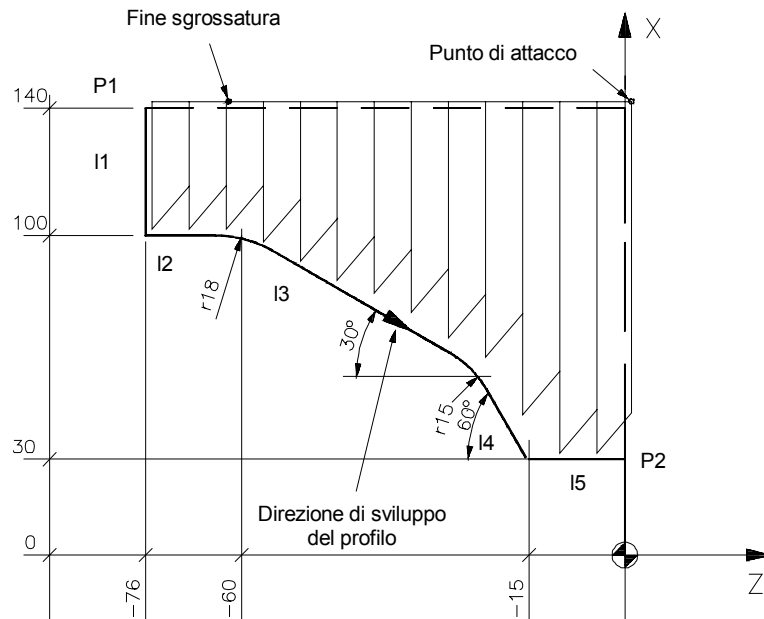


Figura 16-2 Esempio di sgrossatura di un profilo con passate di lavoro parallele all'asse X

.....	PROF1
.....	p1 = Z-76X140
T1.1 M6 S.. F..	I1 = p1, a-90
G X143 Z1.5	I2 = Z0X100, a0
(SPA,PROF1,X,12,X1,Z1)	I3 = Z-60X100, a-30
.....	I4 = Z-15X30, a-60
.....	p2 = Z0X30
	I5 = p2, a0
	G21 p1
	I1
	I2
	r-18
	I3
	r-15
	I4
	I5
	G20 p2

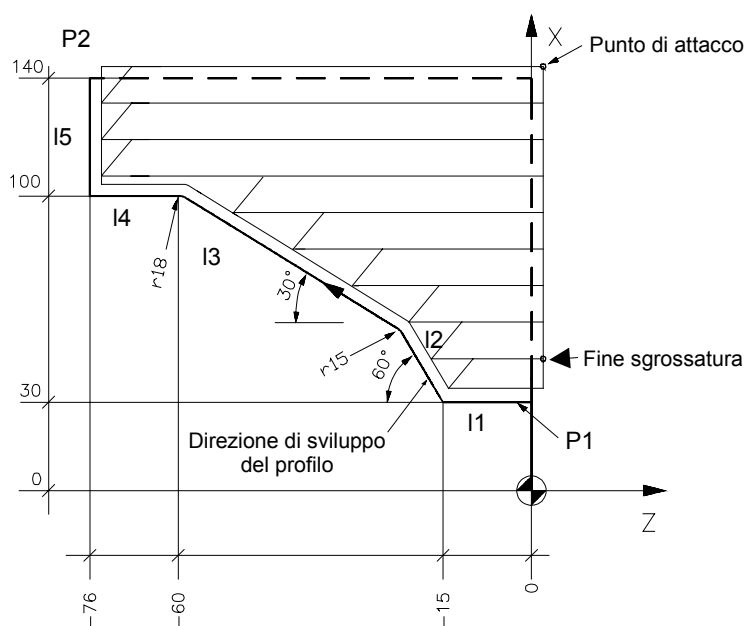


Figura 16-3 Esempio di sgrossatura parallela all'asse Z

PROF2

p1 = Z0X30

I1 = p1, a180

I2 = Z-15X30, a120

I3 = Z-60X100, a150

I4 = Z-75X100, a180

p2 = Z-75X140

I5 = p2, a90

G21 p1

I1

I2

r15

I3

r18

I4

I5

G20 p2

.....

.....

T1.1 M6 S..F..

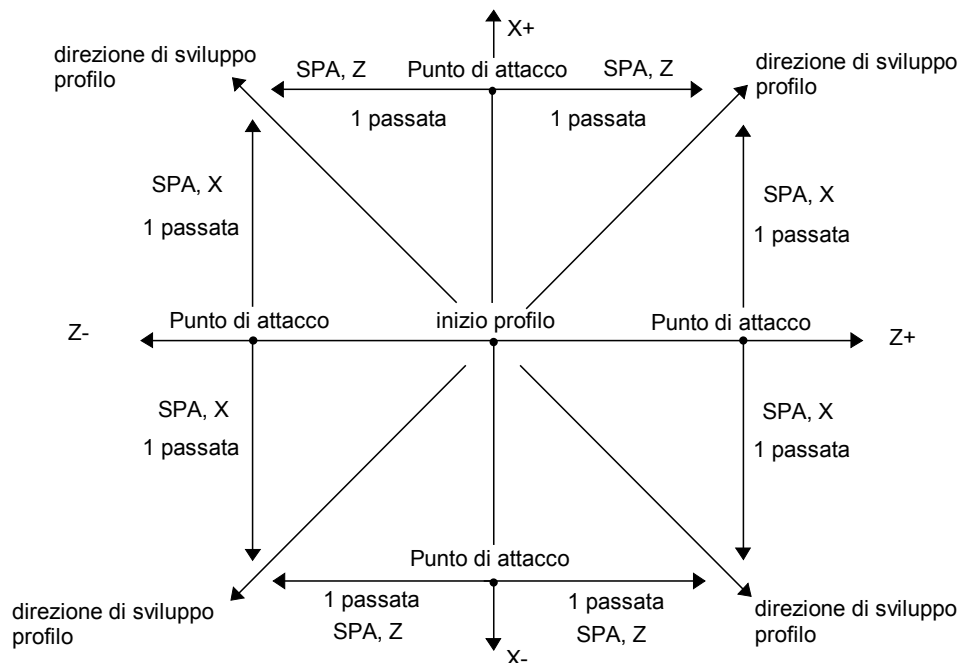
GX143 Z1.5

(SPA, PROF2, Z, 9, X1, Z1)

.....

.....

Grafico per determinare la direzione di sviluppo del profilo in funzione del punto di attacco.



La direzione di sviluppo del profilo deve essere definita considerando la direzione che assume l'utensile per effettuare il ciclo di sgrossatura, che è anche la direzione di sviluppo del profilo.

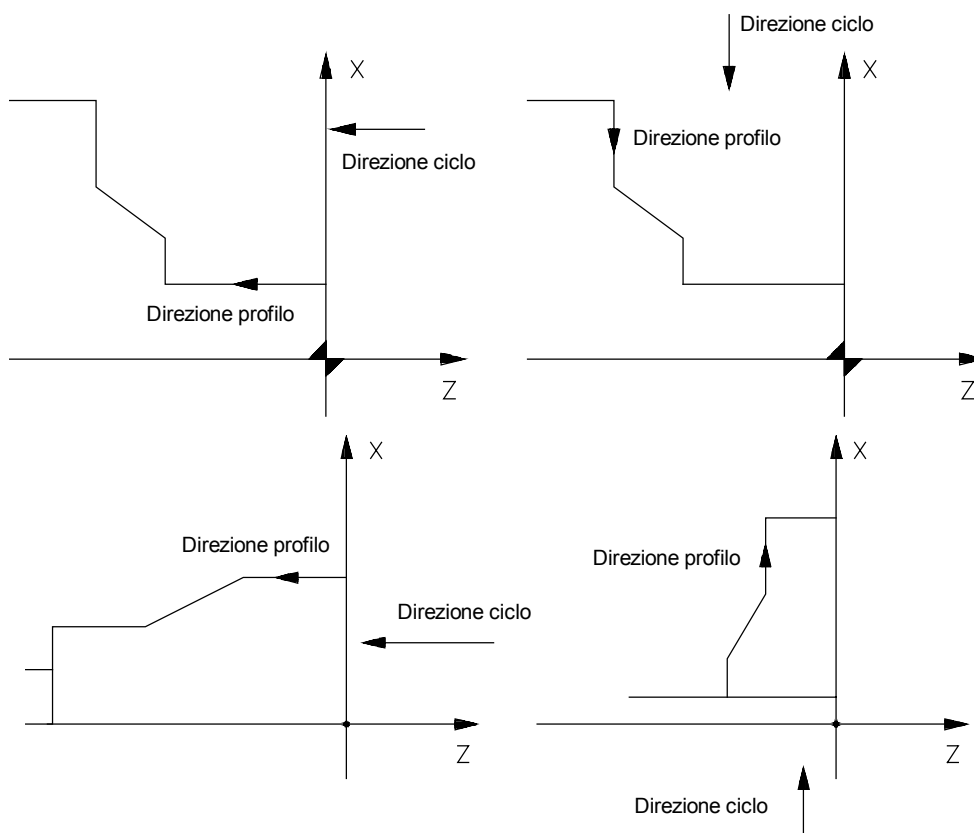


Figura 16-4 Profilo e direzione dell'utensile

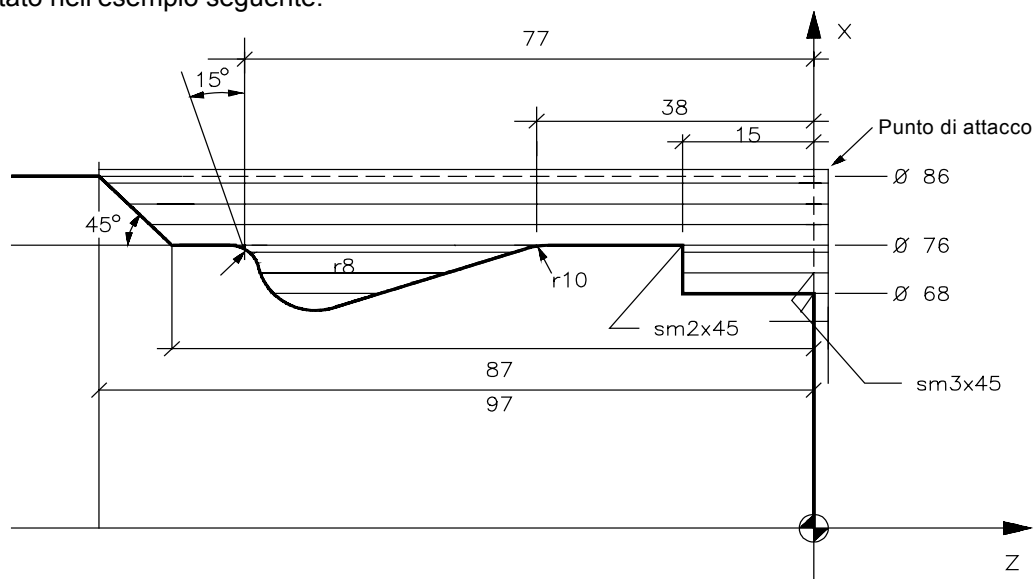
MACROISTRUZIONI DI SGROSSATURA PARASSIALI CON PREFINITURA

Sgrossatura con passate di lavoro parallele all'asse ordinata e ascissa e passata finale parallela al profilo:

(SPF, nome_profilo, nome_asse, n_pass [,asse1 sovram] [,asse2 sovram])

Esempio: (SPF, PROF3, Z, 7, X1, Z1.5)

Per il significato dei parametri e la scelta del punto di attacco, fare riferimento alla macroistruzione SPA. Se il profilo presenta delle cavità, esse verranno sgrossate durante il ciclo SPF come riportato nell'esempio seguente:



PROF3

p1 = Z0X62

l1 = p1, a135

l2 = Z0X68, a180

l3 = Z-15X0, a90

l4 = Z0X76, a180

l5 = Z-38X76, a198

l6 = Z-77X76, a105

p2 = Z-97, X86

l7 = p2, a135

G21 p1

l1

l2

l3

r2

l4

r10

l5

r-8

l6

r4

l4

l7

G20 p2

.....

.....

T1.1 M6 S..F..

GX90 Z2

(SPF, PROF3, Z, 7, X1, Z1)

.....

.....

IMPORTANTE

Il profilo deve essere monotono per l'asse di sgrossatura. In caso contrario il sistema segnala errore.

Il profilo, cioè, può presentare delle cavità, ma per quanto riguarda l'andamento delle quote lungo l'asse di sgrossatura, esso deve essere sempre crescente o sempre decrescente.

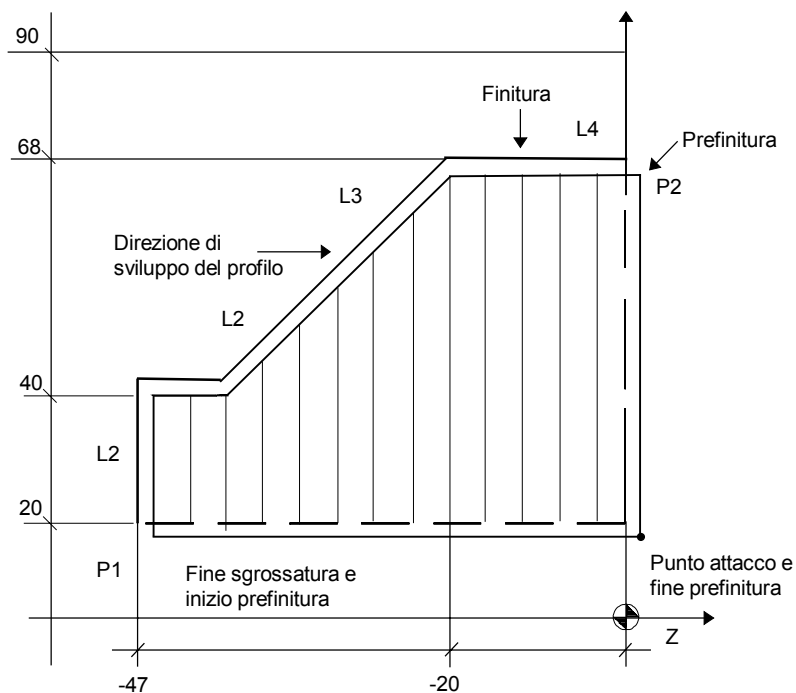


Figura 16-5 Esempio di sgrossatura interna parassiale con prefinitura parallela all'asse X

PROF4

p1 = Z-47X20

I1 = p1, a90

I2 = Z0X40, a0

I3 = Z-20X68, a45

p2 = Z0X68

I4 = p2, a0

G21 p1

I1

I2

I3

I4

G20 p2

.....

.....

T..S..F..

GX15 Z2.5

(SPF, PROF4, X, 10, X2, Z2)

.....

.....

(CLP, PROF4)

MACROISTRUZIONE DI SGROSSATURA PARALLELA AL PROFILO

Questa macroistruzione è studiata per la sgrossatura di pezzi preformati, ovvero forniti di un sovrametallo più o meno costante sul pezzo grezzo.

Il formato è:

(SPP,nome_profilo, n_pass, nome_asse, sovr1, sovr2, nome_asse, sovr1, sovr2)

dove:

nome_profilo è una stringa ASCII che rappresenta il nome del part program nel quale è descritto il profilo.

n_pass numero delle passate di sgrossatura. Può assumere un valore compreso tra 1 e 255.

nome_asse è il nome dell'asse (ascissa Z / ordinata X) sul quale lasciare il sovrametallo indicato

sovr1 sovrametallo da lasciare sul pezzo finito in Z / X

sovr2 sovrametallo sul pezzo grezzo in Z / X

I valori di sovrametallo devono essere sempre programmati, anche quando il loro valore è zero. Per il punto di attacco valgono le stesse considerazioni fatte per le macroistruzioni SPA e SPF.

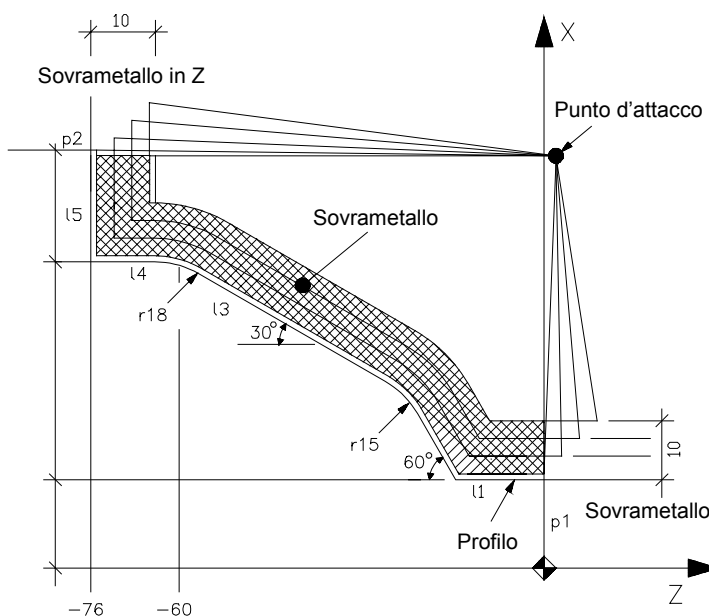


Figura 16-6 Uso di SPP

```
.....
.....
T1.1 M6 S...F...M...
T.. S.. F.. M..
G X143 Z2
(SPP,PROF2,4,Z,1,10,X,1,10)
.....
```

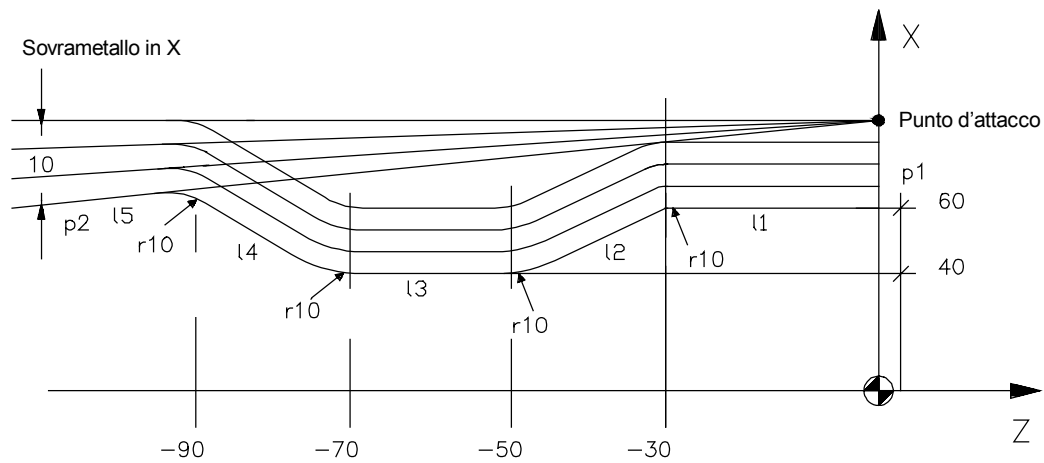


Figura 16-7

.....

 T.. S.. F..M..
 G X84 Z1
 (SPP,PROF5,4,X,0,10,Z,0,0)

PROF5

p1 = Z0X60
 l1 = p1, a180
 l2 = Z-30X60,Z-50X40
 l3 = Z0X40, a180
 l4 = Z-70X40, z-90X60
 p2 = Z-110X60
 l5 = p2, a180
 G21 p1
 l1
 r10
 l2
 r-10
 l3
 r-10
 l4
 r10
 l5
 G20 p2

MACROISTRUZIONE DI FINITURA DI UN PROFILO

Il formato di programmazione è:

(CLP, nome_profilo)

dove:

nome_profilo è una stringa ASCII che rappresenta il nome del part program nel quale è descritto il profilo.

CLP è l'unico ciclo di lavorazione di un profilo durante l'esecuzione del quale vengono attivate le eventuali funzioni F programmate all'interno del profilo stesso.

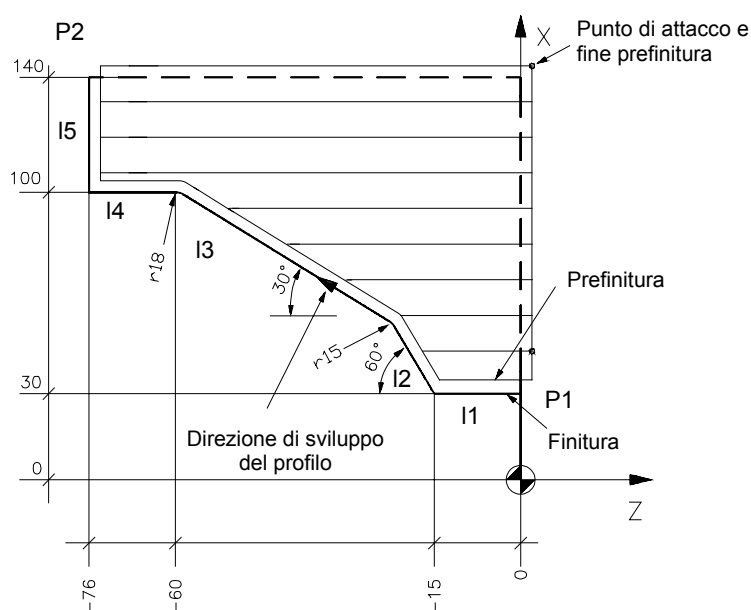


Figura 16-8 Sgrossatura parassiale con prefinitura e finitura

```

.....
T1.1M6 S..F..
GX143 Z1.5
(SPF, PROF2,Z,9,X1,Z1)
X300 Z200
T2.2 M6 S..F..
GX30 Z2
(CLP, PROF2)
GX300 Z200
.....

```

CICLO DI FILETTATURA

Il ciclo di filettatura permette di programmare in un solo blocco una filettatura cilindrica o conica da eseguire in più passate. Il formato di programmazione è:

(FIL,asse1 , [asse2 ,] passo, n_pass [F .] [R .] [G ..] [P...] [M .] [T ..] [H ..] [a ...] [b..] [r ..])

dove:

asse1	nome dell'asse ascissa e/o ordinata (tipicamente Z o X) e quota finale per l'asse indicato
asse2	nome dell'asse ascissa e/o ordinata (tipicamente Z o X) e quota finale per l'asse indicato (opzionale , viene programmato nella filettatura conica)
passo	passo di filettatura
n_pass	numero di passate di filettatura. e un valore compreso tra 1 e 255.
F...	numero di passate di lucidatura. Può assumere un valore compreso tra 1 e 255. Default: 0
R...	distanza dell'utensile dal pezzo nel ritorno (valore di default 1)
G...	filettatura interna/esterna: 0 = filettatura esterna 1 = filettatura interna. Default: esterna (0)
P...	numero di principi (valore di default 1)
M...	modalità di entrata 0 = sgrossatura ad 1 angolo (fianco) 1 = sgrossatura a 2 angoli (zig zag). Default: 0
T...	modo di uscita: 0 = filettatura con gola finale 1 = filettatura senza gola finale (strappo) 2 = uscita con interpolazione circolare (raccordo). Default:0
H...	metrico/pollici/non-standard 0 = filettatura metrica 1 = filettatura whitworth 2 = filettatura non standard con profondità e angolo definiti dai parametri a e b. Default: 0
a...	angolo del filetto non-standard
b...	profondità del filetto non-standard
r...	raggio di uscita per lo stacco. Obbligatorio se è stato programmato il modo di uscita 2 (T2).

I parametri racchiusi fra parentesi quadre possono essere omessi.

Quando si programmano filetti non standard (H2), il passo (K) deve essere:

$$K > 2 * b * \operatorname{tg}(a/2)$$

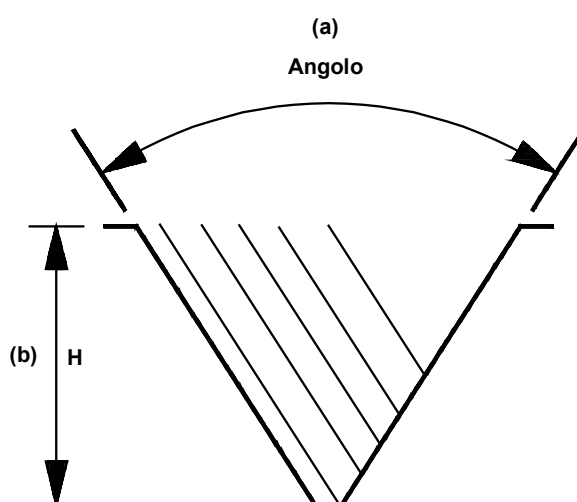
NOTE

- a) L'unità calcola i posizionamenti scivolando lungo il fianco del filetto in modo che le passate avvengano a sezione di truciolo costante.

Nel caso di filettature a più principi, il passo da definire è quello della singola spira. Il sistema esegue ogni passata su tutti i principi prima di passare all'esecuzione delle passate successive. L'esecuzione dei vari principi è realizzata non spostando il punto di partenza delle filettature, ma introducendo uno sfasamento rispetto alla posizione angolare zero del mandrino.

- b) Per filettature con gola finale, si deve programmare la Z finale teorica, poichè il ciclo fisso prevede una maggioranza della corsa pari a metà del passo.

I parametri **a** (angolo del filetto) e **b** (profondità del filetto) sono richiesti solo per programmare filettature non standard (H2).



	metrica	whitw.	altre
angolo	60°	50°	a
H	f(passo)	f(passo)	b

Figura 16-9

Filettatura ad un angolo

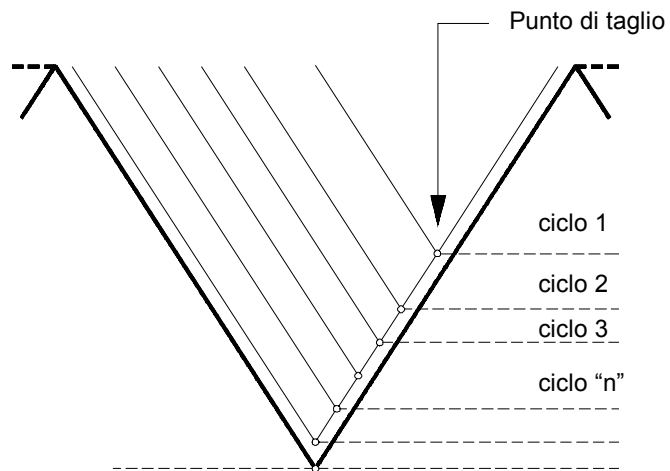


Figura 16-10

Filettatura a due angoli

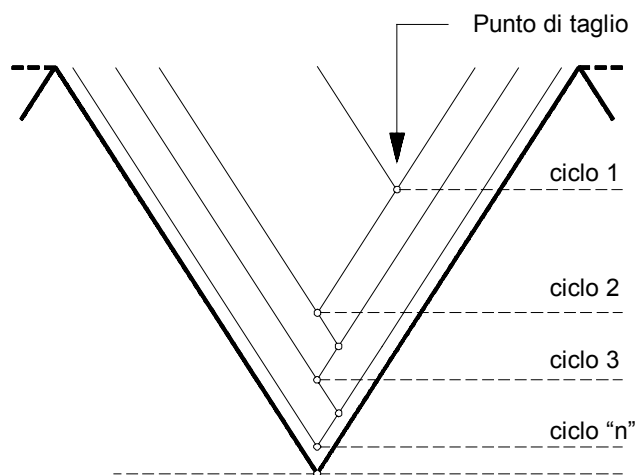


Figura 16- 11

- c) Per filettature a strappo, l'utensile raggiunge la quota programmata e poi esce filettando conico sul diametro di ritorno.
- d) In filettatura a strappo, cioè senza gola finale, non si deve lavorare in SEMIAUTO, altrimenti si ottiene una gola finale e non lo strappo.
- e) Il ciclo di filettatura non può essere definito in G28.
- f) Nel caso di filettatura conica, la massima conicità ammessa è uguale a 1/2 dell'angolo del filetto.

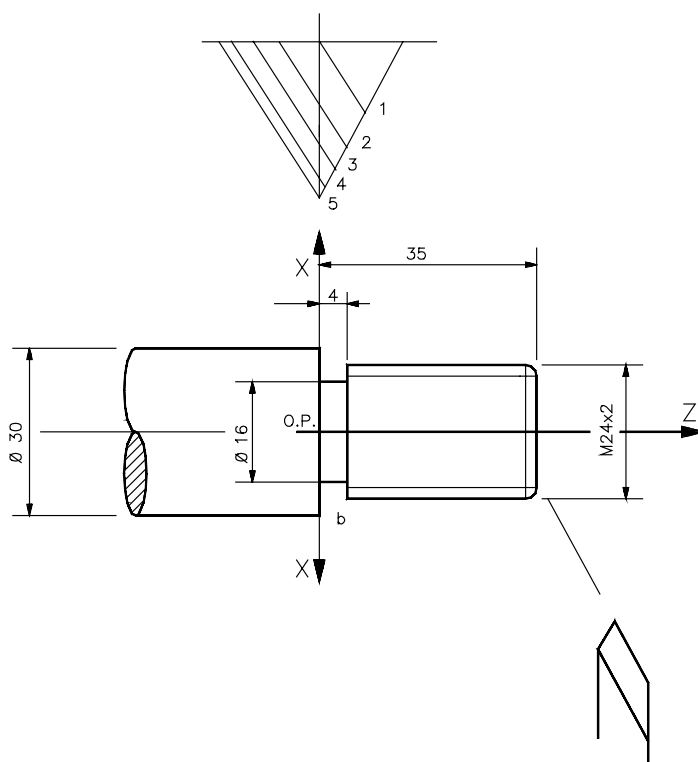


Figura 16-12 Esempio di ciclo di filettatura

N35	T5.5 M6
N36	G0 G97 X24 Z37 S250 M3 M8
N37	(FIL,Z4,2,5,F1,R2)
N38	G0 X250 Z215

CICLO DI TAGLIO GOLE

Il ciclo genera una sequenza di lavorazione per eseguire una gola parallela all'asse ascissa o ordinata (tipicamente Z o X), interna o esterna.

Il formato di programmazione per ottenere una gola parallela ad un asse del piano è:

(TGL,asse1, asse2, larg_ute [, quota_esterna,B/R ..., B/R ...])

Il minimo formato richiesto è:

(TGL, asse1, asse2 , larg_ute)

dove:

asse1	Nome dell'asse ascissa o ordinata del piano e quota finale della gola (la quota iniziale deve essere programmata prima di definire il ciclo di taglio gole)
asse2	Nome del 2 asse del piano e quota interna alla gola (vedere il disegno)
larg_ute	Larghezza utensile
quota_esterna	Quota esterna necessaria per smussi o raccordi esterni (opzionale)
B/R	Raccordo/smusso all, esterno della gola (opzionale)
B/R	Raccordo/smusso sul fondo della gola (opzionale)

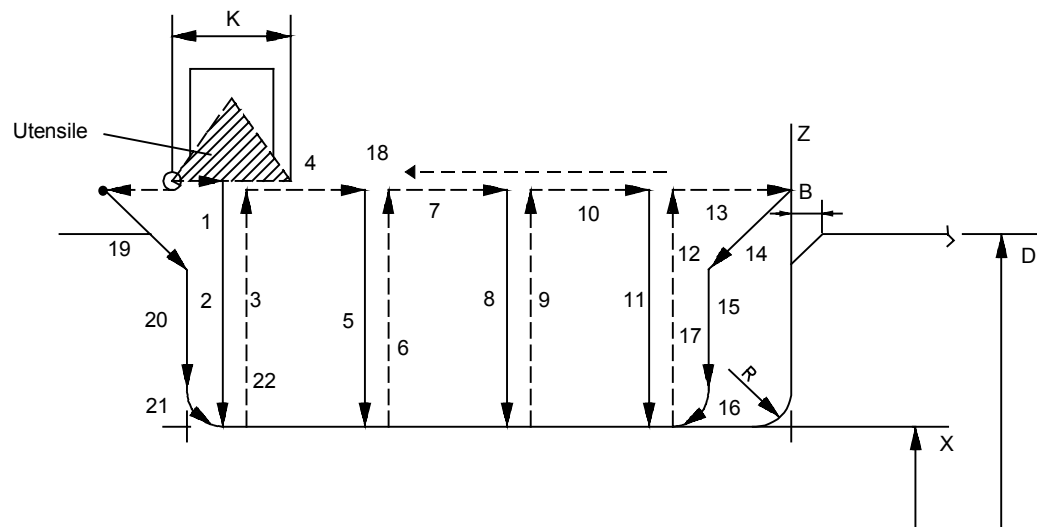


Figura 16-13 Ciclo di taglio gole

Il comando TGL deve essere preceduto da un blocco di movimento in G0/G1 sul punto di inizio ciclo.

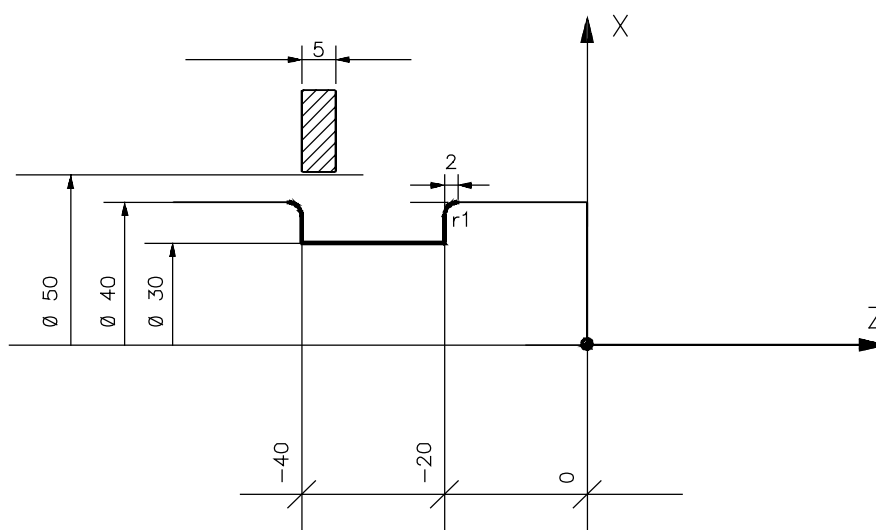


Figura 16-14 Esempio di gola parallela all'asse Z

```
N1 T1.1 M6 S.. F..
N2 G X50 Z-40
N3 TMR=2
N4 (TGL,Z-20,X30,5,40,R1,R0)
N5G X10 Z100
```

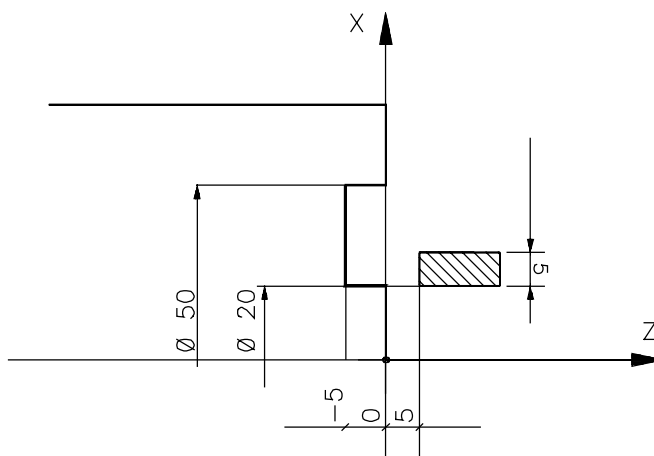


Figura 16-15 Esempio di gola parallela all'asse X

```
N1 T1.1 M6
N2 G X20 Z5
N3 TMR=2
N4 (TGL,X50,Z-5,5)
```

Il sistema forza automaticamente una sosta a fine gola.

Il tempo di sosta è definito dal triletterale TMR. Se non si vuole fare una sosta, programmare TMR=0 prima del ciclo di taglio gola.

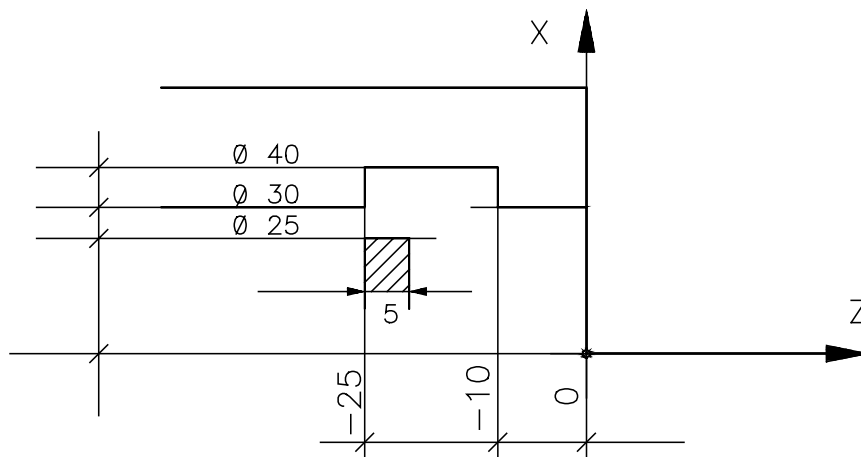


Figura 16-16 Gola interna

N1 T1.1 M6 S.. F.. M..
N2 G X25
N3 Z-25
N4 TMR=2
N5 (TGL,Z-10,X40,5)
N6

A fine ciclo l'utensile ritorna sul punto di inizio ciclo programmato nel blocco precedente.

FINE CAPITOLO

SOMMARIO DEI CARATTERI E DEI COMANDI

CARATTERI CONSENTITI

Questa tabella riporta l'elenco dei caratteri riconosciuti dal controllo su nastro perforato con formato ASCII, EIA ed ISO. Notare che a seconda del codice utilizzato c'è parità dispari (EIA), parità pari (ISO) o nessuna parità (ASCII).

CARATTERE	DESCRIZIONE
0 - 9	• Numeri
+	• Addizione
-	• Sottrazione
*	• Moltiplicazione
/	• Divisione
.	• Punto decimale
"	• Identificatore etichetta
(• Parentesi tonda aperta
)	• Parentesi tonda chiusa
[• Parentesi quadra aperta
]	• Parentesi quadra chiusa
{	• Parentesi graffa aperta

Appendice A

Sommario dei Caratteri e dei Comandi

CARATTERE	DESCRIZIONE
}	<ul style="list-style-type: none">Parentesi graffa chiusa
,	<ul style="list-style-type: none">Separatore di parametri
=	<ul style="list-style-type: none">Simbolo di assegnazione
<	<ul style="list-style-type: none">Minore di (codice di salto)
>	<ul style="list-style-type: none">Maggiore di (codice di salto)
LF (Avanzamento di una linea ISO o ASCII) CR (Ritorno di carrello EIA)	<ul style="list-style-type: none">Fine blocco
#	<ul style="list-style-type: none">Sincronizzazione
&	<ul style="list-style-type: none">Asincronizzazione
!	<ul style="list-style-type: none">Prefisso di variabile utente
@	<ul style="list-style-type: none">Prefisso di variabile PLUS
>>	<ul style="list-style-type: none">Incremento sul singolo operando
A a	<ul style="list-style-type: none">Nome asseAccelerazione sul profiloAngolo con GTL
B b	<ul style="list-style-type: none">Nome asseSmusso in correzione di utensile
C c	<ul style="list-style-type: none">Nome asseCerchio con GTL
D d	<ul style="list-style-type: none">Nome asseDiametro dell'utensile (RQT,RQP)Distanza con GTL
E	<ul style="list-style-type: none">Parametri da utilizzare bei cicli di lavorazione
F	<ul style="list-style-type: none">Velocità di avanzamento degli assi in G1, G2, G3
G	<ul style="list-style-type: none">Codice preparatorio (G00 - G99)Riservato (G100 - G299)Sottoprogrammi di Paramacro (G300 - G999)

CARATTERE	DESCRIZIONE
H h	<ul style="list-style-type: none"> Parametri da utilizzare nelle Paramacro Cambio correttore durante il movimento continuo
I i	<ul style="list-style-type: none"> Coordinate del centro dell'arco in una interpolazione circolare (G2 - G3) (ascissa) Passo variabile in G33 Vettore dimensione utensile (con j e k)
J j	<ul style="list-style-type: none"> Coordinate del centro dell'arco in una interpolazione circolare (G2-G3) (ordinata) Min. incremento di profondità in (G83) Vettore dimensione utensile (con i e k)
K k	<ul style="list-style-type: none"> Fattore di riduz. per I e J nel ciclo di foratura G83 Passo di filettatura (G33) Passo di filettatura (G84) Passo dell'elica in un'interpolazione elicoidale Vettore dimensione utensile (con j e i)
L l	<ul style="list-style-type: none"> Variabili della user table Lunghezza 1 del correttore utensile (RQT,RQP) Lunghezza 2 del correttore utensile (RQT,RQP) Retta con GTL
M m	<ul style="list-style-type: none"> Funzioni ausiliarie Vettore normale alla superficie (con n e o)
N n	<ul style="list-style-type: none"> Numero di blocco del Part Program Vettore normale alla superficie (con m e o)
o	<ul style="list-style-type: none"> Numero di blocco del Part Program Vettore normale alla superficie (con m e o)
P p	<ul style="list-style-type: none"> Nome di asse Punto con GTL
Q	<ul style="list-style-type: none"> Nome di asse
R r	<ul style="list-style-type: none"> Posizionamento rapido nei cicli G81 - G89 Scostamento dallo zero del mandrino (usato nelle filettature a più principi) Raggio in un'interpolazione circolare G02-G03 Raggio sul profilo (solo per correzione utensile) Ciclo G73 Raggio con GTL
S s	<ul style="list-style-type: none"> Velocità del mandrino Intersezione con GTL

Appendice A

Sommario dei Caratteri e dei Comandi

CARATTERE	DESCRIZIONE
T t	<ul style="list-style-type: none">• Indirizzo dell'utensile e del correttore• Tempo necessario per completare lo spostamento in un blocco
U u	<ul style="list-style-type: none">• Nome di asse• Fattore di compensazione sull'asse 1 (offset)
V v	<ul style="list-style-type: none">• Nome di asse• Fattore di compensazione sull'asse 2 (offset)
W w	<ul style="list-style-type: none">• Nome di asse• Fattore di compensazione sull'asse 3 (offset)
X	<ul style="list-style-type: none">• Nome di asse
Y	<ul style="list-style-type: none">• Nome di asse
Z	<ul style="list-style-type: none">• Nome di asse

SOMMARIO DEI CODICI G

CODICE G	FUNZIONE
G00	Posizionamento rapido dell'asse
G01	Interpolazione lineare
G02	Interpolazione circolare oraria
G03	Interpolazione circolare antioraria
G04	Attesa a fine passo
G09	Decelerazione a fine passo
G16	Piano di interpolazione definito
G17	Interpolazione circolare e correzione di utensile sul piano XY
G18	Interpolazione circolare e correzione di utensile sul piano ZX
G19	Interpolazione circolare e correzione di utensile sul piano YZ
G20	Chiude profilo GTL
G21	Apri profilo GTL
G27	Operazione in sequenza continua con riduzione automatica di velocità sugli spigoli
G28	Operazione in sequenza continua senza riduzione di velocità sugli spigoli
G29	Operazione punto-a-punto
G33	Filettatura a passo costante o variabile
G40	Disabilitazione di correzione sul profilo
G41	Correzione sul profilo con utensile a sx
G42	Correzione sul profilo con utensile a dx
G70	Programmazione in pollici
G71	Programmazione in millimetri
G79	Programmazione riferita allo zero macchina
G80	Disabilitazione dei cicli fissi
G81	Ciclo di foratura
G82	Ciclo di lamatura
G83	Ciclo di foratura profonda
G84	Ciclo di maschiatura
G85	Ciclo di alesatura
G86	Ciclo di barenatura
G89	Ciclo di barenatura con sosta
G90	Programmazione assoluta
G91	Programmazione incrementale
G92	Pre-impostazione dell'asse
G93	Modalità di programmazione della velocità di avanzamento a tempo inverso (V/D)
G94	Programmazione della velocità di avanzamento in pollici/min o mm/min
G95	Programmazione della velocità di avanzamento in pollici per giro o millimetri per giro
G96	Velocità di taglio in piedi al minuto o metri al minuto
G97	Programmazione della velocità del mandrino espressa in giri/min
G72	Tastatura del punto con correzione del raggio della sfera del tastatore
G73	Tastatura del foro con correzione del raggio della sfera del tastatore
G74	Tastatura per la definizione dello scostamento teorico dal punto senza compensazione del raggio della sfera
G99	Cancella G92

FUNZIONI MATEMATICHE

- | | | |
|-------|-------|-------|
| • SIN | • SQR | • OR |
| • COS | • ABS | • NOT |
| • TAN | • INT | • FEL |
| • ARS | • NEG | • FEC |
| • ARC | • MOD | • FEP |
| • ART | • AND | |

VARIABILI LOCALI E DI SISTEMA

NOME VARIABILE	FUNZIONE
L	Variabile della user table del PLUS
SN	Numero di sistema
SC	Carattere di sistema
TIM	Temporizzazione di sistema
E	Variabile locale
H	Variabile di paramacro
HF	Flag di paramacro
HC	Variabili stringa di paramacro
!name	Variabile utente
@name	Variabile PLUS
VEF	Fattore di velocità
TPO	Ottimizzazione percorso utensile
TPT	Soglia per TPO
CET (PRC)	Tolleranza di precisione interpolazione circolare
FCT	Soglia per cerchio completo
ARM	Definizione della modalità normalizzazione arco
DLA	Abilita/disabilita look ahead
MDA	Massimo angolo di deviazione
DWT (TMR)	Tempo di sosta
SSL	Limite della velocità del mandrino
ERR	Abilita/disabilita l'uso dell'errore di sistema
STE	Errore di sistema
MSA (UOV)	Definizione di sovrametallo
TRP (RMS)	Velocità di ritorno in maschiatura
DSB	Blocchi barrati disabilitati
UPA (RTA)	Aggiornamento dell'ascissa del tastatore
UPO (RTO)	Aggiornamento dell'ordinata del tastatore
VFF	"Velocity Feed Forward"
ODH	"Online Debug Help"
DYM	Modalità di esecuzione per la G27

TRILETTERALI

CODICE TRILETTERALE	FUNZIONE
DAN	– Definisce il nome dell'asse
IPB (DTL)	– Banda di posizionamento
UAO	– Uso di origini assolute
UTO (IOT)	– Uso di origini temporanee
UIO	– Uso di origini incrementali
RQO	– Riqualfica delle origini
SOL (DLO)	– Limiti di fine corsa software
DPA (DSA)	– Definizione di aree protette
PAE (ASC)	– Abilitazione di area protetta
PAD (DSC)	– Disabilitazione di area protetta
MIR	– Lavorazione speculare
ROT (URT)	– Rotazione del piano attivo
SCF	– Fattori di scala
AXO	– Definizione offset dell'asse
RQT (RQU)	– Riqualfica del correttore e della vita utensile
RQP	– Riqualfica correttore di utensile
TOU (TOF)	– Dichiarazione di utensile scaduto
DPP (DPT)	– Definizione dei parametri di tastatura
LOA	– Caricamento Tabelle in memoria dual port
RPT	– Apertura di una ripetizione di un gruppo di blocchi
ERP	– Chiusura di una ripetizione
CLS	– Chiamata di sottoprogrammi
PTH	– Definizione del direttorio di subroutine e paramacro
EPP	– Esecuzione di una parte di programma
EPB	– Esecuzione di un blocco di programma
GTO	– Esecuzione di salto
IF,ELSE,ENDIF	– Esecuzione condizionata di parti di programma

Appendice A

Sommario dei Caratteri e dei Comandi

CODICE TRILETTTERALE	FUNZIONE
GDV	– Definizione device
RDV	– Rilascio device
UDA	– Assi duali
SDA	– Assi duali
UGS (UCG)	– Usa la scala grafica
CGS (CLG)	– Cancella il campo grafico
DGS (DCG)	– Disabilita il campo grafico
UPR	– Usa rotazione del piano
UVP	– Usa virtualizzazione polare
UVC	– Usa virtualizzazione cilindrica
TCP	– Movimentazione lungo la direzione utensile
DIS	– Visualizza un messaggio per l'operatore
DLY	– Tempo di ritardo
DSB	– Disabilitazione blocchi barrati
REL	– Disattivazione del part program
WOS	– Wait su segnale
GTA	– Acquisizione assi
GTS	– Condivisione mandrino
SND	– Invio di un messaggio di sincronizzazione
WAI	– Attesa di un messaggio di sincronizzazione
EXE	– Attivazione automatica di un part program
ECM	– Esecuzione blocco in modo MDI
PRO	– Definizione processo di default
DCC	– Definizione canale di comunicazione
PVS	– Selezione variabile PLUS
GTP	– Determinazione punto di attacco per contornitura automatica
CCP	– Esecuzione contornitura automatica

CODICE TRILETTERALE	FUNZIONE
SPA	– Sgrossatura parassiale senza prefinitura
SPF	– Sgrossatura parassiale con prefinitura
SPP	– Sgrossatura parallela al profilo
CLP	– Ciclo di finitura
FIL	– Ciclo di filettatura
TGL	– Ciclo di taglio gole

CODICI ASCII

Le tabelle seguenti mostrano i 256 elementi del set di caratteri ASCII esteso con i loro equivalenti decimali ed esadecimali.

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
000	00	BLANK (NULL)	016	10	▶ (DLE)	032	20	BLANK (SPACED)	048	30	0
001	01	☺ (SOH)	017	11	◀ (DC1)	033	21	!	049	31	1
002	02	☹ (STX)	018	12	↕ (DC2)	034	22	"	050	32	2
003	03	♥ (ETX)	019	13	!! (DC3)	035	23	#	051	33	3
004	04	♦ (EOT)	020	14	¶ (DC4)	036	24	\$	052	34	4
005	05	♣ (ENQ)	021	15	§ (NAC)	037	25	%	053	35	5
006	06	♠ (ACH)	022	16	■ (SYN)	038	26	&	054	36	6
007	07	● (BEL)	023	17	↕ (ETB)	039	27	'	055	37	7
008	08	◼ (BS)	024	18	↑ (CAN)	040	28	(056	38	8
009	09	◯ (HT)	025	19	↓ (EM)	041	29)	057	39	9
010	0A	◐ (LF)	026	1A	→ (SUB)	042	2A	*	058	3A	:
011	0B	◯ [↻] (VT)	027	1B	← (ESC)	043	2B	+	059	3B	;
012	0C	♀ (FF)	028	1C	└ (FS)	044	2C	,	060	3C	<
013	0D	♪ (CR)	029	1D	↔ (GS)	045	2D	-	061	3D	=
014	0E	♫ (SO)	030	1E	▲ (RS)	046	2E	.	062	3E	>
015	0F	☼ (SI)	031	1F	▼ (US)	047	2F	/	063	3F	?

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
064	40	@	080	50	P	096	60	`	112	70	p
065	41	A	081	51	Q	097	61	a	113	71	q
066	42	B	082	52	R	098	62	b	114	72	r
067	43	C	083	53	S	099	63	c	115	73	s
068	44	D	084	54	T	100	64	d	116	74	t
069	45	E	085	55	U	101	65	e	117	75	u
070	46	F	086	56	V	102	66	f	118	76	v
071	47	G	087	57	W	103	67	g	119	77	w
072	48	H	088	58	X	104	68	h	120	78	x
073	49	I	089	59	Y	105	69	i	121	79	y
074	4A	J	090	5A	Z	106	6A	j	122	7A	z
075	4B	K	091	5B	[107	6B	k	123	7B	{
076	4C	L	092	5C	\	108	6C	l	124	7C	
077	4D	M	093	5D]	109	6D	m	125	7D	}
078	4E	N	094	5E	^	110	6E	n	126	7E	~
079	4F	O	095	5F	_	111	6F	o	127	7F	△

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
128	80	Ç	144	90	É	160	A0	á	176	B0	≡
129	81	ü	145	91	æ	161	A1	í	177	B1	≡
130	82	é	146	92	Æ	162	A2	ó	178	B2	≡
131	83	â	147	93	ô	163	A3	ú	179	B3	
132	84	ä	148	94	ö	164	A4	ñ	180	B4	†
133	85	à	149	95	ò	165	A5	Ñ	181	B5	‡
134	86	å	150	96	û	166	A6	ä	182	B6	
135	87	ç	151	97	ù	167	A7	o	183	B7	π
136	88	ê	152	98	ÿ	168	A8	ı	184	B8	‡
137	89	ë	153	99	Ö	169	A9	┐	185	B9	
138	8A	è	154	9A	Ü	170	AA	┐	186	BA	
139	8B	ï	155	9B	ϕ	171	AB	½	187	BB	π
140	8C	î	156	9C	£	172	AC	¼	188	BC	⌋
141	8D	ì	157	9D	Ÿ	173	AD	ı	189	BD	⌋
142	8E	Ä	158	9E	Pt	174	AE	«	190	BE	‡
143	8F	Å	159	9F	f	175	AF	»	191	BF	⌋

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
192	C0	┐	208	D0	⌋	224	E0	α	240	F0	≡
193	C1	┐	209	D1	┐	225	E1	β	241	F1	±
194	C2	┐	210	D2	π	226	E2	Γ	242	F2	≥
195	C3	┐	211	D3	⌋	227	E3	π	243	F3	≤
196	C4	—	212	D4	⌋	228	E4	Σ	244	F4	┐
197	C5	┐	213	D5	F	229	E5	σ	245	F5	J
198	C6	┐	214	D6	π	230	E6	μ	246	F6	÷
199	C7		215	D7		231	E7	τ	247	F7	≈
200	C8	⌋	216	D8	‡	232	E8	φ	248	F8	°
201	C9	┐	217	D9	┐	233	E9	θ	249	F9	•
202	CA	⌋	218	DA	┐	234	EA	Ω	250	FA	•
203	CB	┐	219	DB	■	235	EB	δ	251	FB	√
204	CC		220	DC	■	236	EC	∞	252	FC	∩
205	CD	=	221	DD	┐	237	ED	∅	253	FD	₂
206	CE		222	DE	┐	238	EE	€	254	FE	■
207	CF	┐	223	DF	■	239	EF	∩	255	FF	BLANK "FF"

Set di caratteri ASCII esteso

Appendice A

Sommario dei Caratteri e dei Comandi

FINE APPENDICE

MESSAGGI D'ERRORE

Descrizione dei messaggi di errore ed azione di recupero

Codice	Descrizione del messaggio ed azione di recupero
NC001	Errore di sintassi Errore di sintassi rilevato nel blocco di programma o nel blocco MDI.
NC002	Numero assi errato per i codici G Errore visualizzato per indicare che: <ul style="list-style-type: none"> • si deve programmare almeno un asse in G04 • si deve programmare un solo asse nel blocco con cicli fissi (da G81 a G89).
NC003	Mancano parametri per il ciclo fisso Sono stati omessi i parametri del ciclo fisso (esempio K, I, ...).
NC004	Mancano parametri per i codici G I parametri per il codice G richiesto sono stati omessi (esempio G33 ... K).
NC005	Mancano J e/o K per il ciclo G83 I parametri K o J sono stati omessi nella programmazione del ciclo fisso G83.
NC006	Mancano I e/o J per i codici G2/G3 I parametri I e/o J sono stati omessi nella programmazione dei codici G2/G3 (cerchi).
NC007	Parametri mancanti in tastatura I parametri del ciclo di tastatura sono stati omessi.
NC008	Errore di formato Questo errore viene visualizzato nei casi seguenti: <ul style="list-style-type: none"> • Indice della variabile errato • Velocità di avanzamento (F) con valore 0 o negativo • Formato di variabile errato • Numero di ripetizione non ammesso (deve essere compreso tra 1 e 65535) • Errore di formato in assegnazione (esempio: tra stringhe di lunghezza differente) • Errore di lettura/scrittura di variabili PLUS • Errore di formato di una variabile carattere nel triletterale DIS (variabile non specificata come CHAR) • Numero di area protetta da definire non ammesso: $0 < \text{numero di area protetta} < 4$. • Variabile non configurata

Codice	Descrizione del messaggio ed azione di recupero
NC009	Simbolo non definito Viene visualizzato nei casi seguenti: <ul style="list-style-type: none"> nome dell'asse non configurato in AMP variabile non configurata.
NC010	Overflow Espressione troppo lunga.
NC011	Funzione non disponibile <ul style="list-style-type: none"> è stata richiesta l'attivazione di una M di tipo blocco al calcolo con il blocco al calcolo già attivo, oppure con il sistema nello stato di HOLD. richiesta di bypass di avanzamento con blocco attivo non lineare.
NC012	Uso errato dell'asse slave Un asse dichiarato in precedenza come "slave" nella programmazione di assi duali con triletterale UDA è programmato direttamente in un part program o in un blocco MDI. Questo errore viene visualizzato anche nei seguenti casi: <ol style="list-style-type: none"> quando si tenta di spostare manualmente l'asse slave. quando programmando UDA, l'asse slave è già interessato in una programmazione TCP come asse lineare o rotativo. quando programmando UDA, l'asse slave è già interessato in una virtualizzazione e (UPR, UVP, UVC) come asse reale o virtuale. quando programmando UDA, l'asse master è già interessato in una virtualizzazione (UPR, UVP, UVC) come asse virtuale.
NC013	Operando non disp. nel ciclo fisso Operando non ammesso dai cicli fissi G72 G73 G74.
NC014	Parametro K non disponibile in G84 Parametro K non ammesso durante la programmazione di G84 con mandrino senza trasduttore.
NC015	Programmazione errata di G2/G3 Sono stati specificati sia il centro, che il raggio (R) del cerchio da programmare. Eliminare o il raggio (R) o gli operandi I J che specificano il centro.
NC016	Numero di operandi illegale Numero illegale di operandi nel triletterale AXO.
NC017	Numero di pseudo assi illegale Troppi pseudo assi programmati nel blocco: non ne sono ammessi più di sei.
NC018	Numero assi illegale in G33 Programmati più di due assi in G33.
NC020	G non permessa G non consentita nel ciclo di sgrossatura.

Codice	Descrizione del messaggio ed azione di recupero
NC021	Operando non permesso con G Operando incompatibile con il tipo di movimento.
NC022	Blocco e stato del sistema incongruenti Blocco programmato o digitato non compatibile con lo stato del sistema: <ul style="list-style-type: none"> • abilitazione/disabilitazione UDA con altre virtualizzazioni attive • sincronizzazione con blocchi sospesi in esecuzione (cioè sincronizzazione con compensazione diametro utensile attiva).
NC024	G e stato del programma non congruenti Funzione G non congruente con lo stato del programma attivo. Per esempio: <ul style="list-style-type: none"> • con compensazione diametro utensile attiva (G41,G42) non possono essere programmate gli operatori G di ciclo fisso. • le filettature non possono essere programmate durante una compensazione diametro utensile, o durante un ciclo fisso. • le funzioni del piano di interpolazione (G17, G18, G19) non possono essere programmate durante una compensazione diametro utensile (G41, G42).
NC025	G e modo dinamico non congruenti Funzione G non compatibile con la modalità dinamica corrente. Per esempio: le funzioni G72, G73, G74 non vengono accettate nelle modalità di avanzamento continuo (G27, G28). Passaggio da G27 a G28 o viceversa con rampe non lineari attive (MOV > 1).
NC026	G41/G42 e stato PP non congruenti Incongruenza tra lo stato attuale del part program e la programmazione della compensazione diametro utensile (G41/G42).
NC027	G necessita mandrino con trasduttore G33 e il macrociclo FIL di filettatura richiedono mandrino con trasduttore.
NC028	G non congruente con il modo feedrate G72, G73, G74 devono essere eseguiti con G94 attivo
NC029	Operando e stato PP non congruente Operando incongruente con lo stato attuale del part program. Per esempio: gli operandi r, b non sono consentiti nello stato ISO standard (G40)
NC030	M e modo dinamico non congruente Operandi logici di macchina incongruenti con la modalità dinamica corrente. Per esempio: <ul style="list-style-type: none"> • M di inizio/fine movimento non compatibile con i codici G27/G28. • programmazione T con compensazione diametro utensile attiva (G41, G42).
NC031	M/T/S e tipo di movimento incongruenti Operandi logici di macchina incongruenti con il tipo di movimento. Per esempio: G33 + funzione M di fine movimento.

Codice	Descrizione del messaggio ed azione di recupero
NC032	Operando inibito nel ciclo di tastatura Operandi inibiti per il ciclo di tastatura attivo Per esempio: <ul style="list-style-type: none"> • con G73 attivo è inibita la programmazione degli operandi I,J,K,R,u,v,w,b,t • con G72 attivo è inibita la programmazione degli operandi I,J,K,R,u,v,w,b,t,r
NC033	Terzo asse non programmato Manca la programmazione del terzo asse dell'elica.
NC034	Funzione "Expedite" senza movimento Una M "expedite" è presente nel blocco in esecuzione ma non c'è movimento associato. La funzione M "expedite" deve essere associata ad un movimento.
NC035	Feed o speed non programmata <ul style="list-style-type: none"> • Velocità di avanzamento o velocità rotazione mandrino non programmata nell'esecuzione di cicli fissi. • Blocco di movimento in G1/G2/G3 programmato senza velocità di avanzamento.
NC036	'Asse 'Z' mancante in G87 Non è stato programmato l'asse Z nel ciclo G87.
NC037	Variabile di sola lettura Si tenta di accedere in scrittura su una variabile dichiarata di sola lettura (esempio: variabile TIM).
NC038	Blocco Part Program troppo lungo È stato programmato un record troppo lungo (>127 caratteri) nel part program che compare accanto all'errore (NOME PART PROGRAM).
NC039	Accesso negato al Part Program Il file di part program specificato accanto a questo errore non è accessibile in lettura, in quanto già aperto in scrittura da un altro utilizzatore (editor, DOS Real Time, ecc...).
NC040	Blocco PP non permesso dalla seriale Blocco non permesso durante l'esecuzione di part program da linea seriale.
NC041	Config. linea seriale errata per EPS Configurazione della linea seriale non corretta per EPS.
NC042	Livello di annidamento "IF" magg. di 5 Superato il massimo livello di annidamento di comandi (IF)
NC043	ELSE non permesso Programmato un comando (ELSE) senza che sia stato preceduto da un comando (IF)
NC044	ENDIF non permesso Programmato un comando (ENDIF) non preceduto da un comando (IF).
NC048	Argomento illegale per TAN L'argomento dell'operatore TAN è 90° (il risultato sarebbe infinito).

Codice	Descrizione del messaggio ed azione di recupero
NC049	Argomento illegale per SQR L'argomento dell'operatore SQR è un numero negativo.
NC050	Troppi assi programmati Sono stati programmati più di nove assi in un blocco.
NC051	Divisione per zero È stata programmata una divisione per zero nel calcolo di una quota asse (ad esempio X 10/0).
NC052	Stringa troppo lunga La lunghezza di una stringa non può eccedere gli 127 caratteri. Questo errore è visualizzato se viene utilizzata una stringa che supera tale limite nei casi seguenti: <ul style="list-style-type: none"> • visualizzazione di una stringa con il triletterale "DIS" • assegnazione di una variabile stringa (SC).
NC053	Etichetta duplicata Questo errore viene visualizzato in fase di attivazione del part program e indica che sono state programmate due etichette identiche. Accanto all'errore viene visualizzata anche l'etichetta duplicata.
NC054	Etichetta non definita L'etichetta programmata in un'istruzione di salto (GTO) o in una istruzione di esecuzione di subroutine (EPP) non esiste.
NC055	Etichetta troppo lunga Questo errore, che viene visualizzato in fase di attivazione del part program, indica che è stata programmata un'etichetta con più di sei caratteri. L'etichetta viene visualizzata accanto all'errore.
NC056	N° sottoprogr. > di quelli configurati Questo errore, che viene visualizzato in fase di attivazione di un part program, indica che il numero di CLS a subroutine nel part program è maggiore del massimo configurato in AMP. Modificare tramite AMP questo valore nella sezione "PROC. CONFIGURATION".
NC057	N° etichette maggiore di quelle config. Questo errore, che viene visualizzato quando è attivato il part program indica che il numero di etichette programmate nel part program è maggiore del massimo configurato in AMP. Modificare tramite AMP questo valore nella sezione "PROC CONFIGURATION".
NC058	Fine programma Segnalazione di fine programma per operazioni di: <ul style="list-style-type: none"> • Skip • Modifica • Ricerca stringa • Esecuzione programma

Codice	Descrizione del messaggio ed azione di recupero
NC059	Inizio programma Segnalazione di inizio programma per Skip o Ricerca stringa.
NC060	Annidamento di RPT maggiore di 5 Superato il massimo livello di annidamento per ripetizioni (5).
NC061	Annidamento di subroutine maggiore di 4 Superato il massimo livello di annidamento per subroutine (4).
NC062	Annidamento di EPP maggiore di 5 Superato il massimo livello di annidamento di triletterali EPP(5).
NC063	RPT/EPP aperto a fine programma <ul style="list-style-type: none"> E' stata raggiunta la fine del programma senza trovare il blocco ERP che chiude il ciclo RPT programmato. Fine del file raggiunta senza completare la subroutine definita con EPP.
NC064	ERP senza RPT Il triletterale ERP è stato programmato senza essere preceduto dal triletterale RPT.
NC065	Errore durante la gestione PP Errore durante la lettura/scrittura del part program che può verificarsi nelle seguenti operazioni: <ul style="list-style-type: none"> SKIP di un blocco di part program SPG/REL di un part program Apertura/chiusura di subroutine Tentativo di accesso ad un part program inesistente o protetto.
NC066	Part program non trovato Il part program o la subroutine da selezionare o da eseguire non sono stati trovati nella directory E:\UPP.
NC067	Part program non selezionato Questo errore viene visualizzato negli stati seguenti: <ul style="list-style-type: none"> CYCLE START in modo AUTO o BLK/BLK con part program non attivo comandi SKIP, MODIFICA, ESCAPE con part program non selezionato istruzione di salto (GTO) eseguita dal sistema in modalità MDI.
NC068	Numero di processo fuori range Il numero di processo scritto nel triletterale non può essere superiore a quello configurato in AMP o inferiore a 1.
NC069	Paramacro modale ancora attiva E' stata programmata una paramacro quando una paramacro modale era già attiva.
NC070	Paramacro non configurate Si cerca di utilizzare una paramacro senza averla configurata in AMP.
NC078	Opzione Software non installata
NC079	Opzione Software non disponibile. Vedi security

Codice	Descrizione del messaggio ed azione di recupero
NC080	Asse non riferito <ul style="list-style-type: none"> • L'asse programmato non è stato azzerato. • Errore nella definizione di un'area protetta con il trilletterale DPA: l'asse specificato non è stato azzerato. • Tentativo di riqualifica di un correttore associato ad un asse non azzerato. • Tentativo di presetting di un correttore associato ad un asse non azzerato.
NC081	DPP non definito Parametri del ciclo di tastatura (quota di accostamento, quota di sicurezza, velocità) non definiti con il trilletterale DPP.
NC082	Troppi codici M "expedite" Nel blocco è stato programmato più di un codice M "Expedite".
NC083	Codice M non definito E' stato programmato un codice "M" non configurato in AMP. Configurare in AMP il codice M richiesto e riavviare il sistema.
NC084	Cerchio incongruente Cerchio non congruente: il raggio o i punti finali del cerchio non sono corretti.
NC085	Parametri di filettatura (I,K,R) errati Parametri di filettatura incongruenti. Calcolare il parametro I in base alla seguente formula: $\frac{16 k}{2(\text{distanza filettatura})}$
NC086	Passo elica non congruente Passo dell'elica incongruente.
NC087	Assi del piano necessitano di SCF Gli assi del piano, nella programmazione di G02/G03 (cerchio), devono avere lo stesso fattore di scala. Modificare il fattore di scala tramite il trilletterale SCF.
NC088	Profilo non congruente Programmazione errata di un profilo ISO.
NC089	Direzione del profilo errata Il valore della compensazione diametro utensile da applicare in G41/G42 (RAGGIO UTENSILE + MSA) provoca un inversione della direzione sul profilo.
NC090	Errore nella disabilitazione G42/G41 Uscita errata dalla modalità compensazione diametro utensile (G40).
NC091	Troppi blocchi da risolvere Durante la programmazione con compensazione diametro utensile attiva, (G41 - G42) sono stati programmati troppi movimenti fuori dal piano di interpolazione (ne sono ammessi al massimo due).

Codice	Descrizione del messaggio ed azione di recupero
NC092	Entrata in zona protetta Il movimento programmato entra in una delle tre aree protette attive.
NC093	Ciclo fisso nel piano ruotato Programmazione di un ciclo fisso su un piano ruotato. Disabilitare la rotazione sul piano.
NC094	Dati ciclo fisso non congruenti Parametri associati a cicli fissi incongruenti (I, J, K, R) Per esempio: K = 0. Si sta eseguendo un ciclo G84 o G86 con il mandrino non in stato esclusivo
NC095	Parametri mancanti per G87 Nel ciclo fisso di foratura G87 mancano dei parametri questo ciclo viene usato nelle paramacro del WOOD
NC096	Program. ciclo di tastatura errato La distanza di accostamento è nulla. Misura di un foro con raggio nullo (per esempio G73r0E5).
NC097	Ciclo di tastatura Foro non completo Ciclo di tastatura del foro non completato.
NC098	Ciclo di tastatura non eseguito Ciclo di tastatura non eseguito. Il tastatore non ha trovato il punto da misurare prima di raggiungere la zona di sicurezza.
NC099	Tastatore non rientrato Il tastatore è già a contatto con la superficie all'inizio del ciclo di tastatura.
NC100	Hardware fuori corsa L'asse programmato è in oltre -corsa hardware. Riportare manualmente l'asse entro i limiti operativi.
NC101	Limite positivo fine corsa SW Il movimento programmato provoca l'uscita dell'asse dal limite operativo positivo programmato o configurato.
NC102	Limite positivo fine corsa HW Si è cercato di effettuare uno spostamento manuale con direzione positiva (JOG DIR +) con l'asse già oltre il limite dell'oltre-corsa positivo. Impostare JOG DIR in direzione negativa (-) e premere CYCLE START per riportare l'asse entro i limiti operativi hardware. N.B. questo è l'unico modo per rientrare nei limiti operativi hardware.
NC103	Limite negativo fine corsa HW Si è cercato di effettuare uno spostamento manuale con direzione negativa (JOG DIR -) con l'asse già oltre il limite dell'oltre-corsa negativo. Impostare JOG DIR in direzione positiva (+) e premere CYCLE START per riportare l'asse indietro entro i limiti operativi hardware. N.B.: questo è l'unico modo per tornare entro i limiti operativi hardware.

Codice	Descrizione del messaggio ed azione di recupero
NC104	<p>Limite positivo fine corsa SW</p> <p>Questo messaggio viene visualizzato quando l'asse è sul limite operativo positivo (programmato o caratterizzato) e lo si desidera muovere con uno spostamento manuale con JOG DIR = +, oppure quando il movimento è terminato perché ha l'asse ha raggiunto il limite operativo positivo.</p>
NC105	<p>Limite negativo fine corsa SW</p> <p>Questo messaggio viene visualizzato quando l'asse è sul limite operativo negativo (programmato o caratterizzato) e lo si desidera muovere con uno spostamento manuale con JOG DIR = -, oppure quando il movimento è terminato perché ha l'asse ha raggiunto il limite operativo negativo.</p>
NC106	<p>JOG past limite finecorsa SW</p> <p>Il valore JOG programmato per uno spostamento manuale in JOG INCR porterebbe l'asse oltre il limite operativo caratterizzato o programmato.</p>
NC107	<p>Assi non sul profilo</p> <p>Questo messaggio viene visualizzato quando si cerca di uscire dallo stato di CYCLE STOP dopo aver effettuato spostamenti manuali su uno o più assi senza riportarli sul profilo. Selezionare "JOG RETURN" ed eseguire il "ritorno al profilo".</p>
NC108	<p>Home e JOG DIR non congruenti</p> <p>Si è cercato di azzerare un asse con una direzione JOG DIR opposta alla direzione di ricerca configurata per quell'asse in AMP. NOTA: Se il ciclo HOME è configurato in AMP come "Automatic", il sistema automaticamente corregge JOG DIR e non visualizza questo errore. Premere la softkey JOG DIR per rendere congruente la direzione di azzeramento con quella configurata in AMP.</p>
NC109	<p>Errore in uscita da HOLD: modo cambiato</p> <p>Si tenta di uscire dallo stato di HOLD con una modalità operativa diversa da quella attiva quando è avvenuto l'ingresso nello stato di HOLD. Selezionare la stessa modalità operativa e ripetere l'uscita da HOLD.</p>
NC110	<p>Blocco non permesso in HOLD</p> <p>E' stato programmato un blocco di MDI in stato di HOLD che comporta un movimento degli assi: in HOLD gli assi possono essere spostati solo manualmente. In stato di HOLD è stata programmata una funzione M del tipo "NON ACCETTABILE IN HOLD".</p>
NC111	<p>Active reset negato</p> <p>Si è cercato di eseguire un ACTIVE RESET su un blocco in esecuzione con G27-G28 o su un blocco che è seguito da un blocco circolare (G02/G03), o su un blocco che rappresenta l'ultimo blocco in esecuzione prima di un blocco sintatticamente non corretto. A questo punto il sistema accetta solo un altro ACTIVE RESET, utile quando si cerca di bypassare il blocco circolare successivo, o un RESET.</p>

Codice	Descrizione del messaggio ed azione di recupero
NC112	Errato uso degli assi rollover con G90 È stato programmato, con G90 attivo, una quota per l'asse rollover maggiore del suo passo.
NC113	JOG DIR e JOG RETURN incongruenti Se durante l'esecuzione del JOG RETURN (automatico o manuale) la direzione è negativa, il sistema forza la direzione positiva. Se questa inversione è impedita dalla logica di macchina il sistema visualizza questo errore.
NC115	Tastatura eseguito prima della fine del movimento di avvicinamento Tastatura effettuata durante la fase di avvicinamento rapido.
NC116	Uso non corretto dell'asse reale in virtualizzazione È stato programmato un asse reale con una virtualizzazione attiva.
NC117	Direzione tool attiva: movimento non permesso È attiva la movimentazione lungo la direzione utensile: tutti gli altri movimenti non sono permessi.
NC118	Limite inferiore fine corsa SW Il movimento programmato provoca l'uscita dell'asse dal limite operativo negativo programmato o configurato.
NC119	Comando non permesso durante la ricerca in memoria Comando non permesso durante la ricerca memorizzata.
NC120	Modo di selezione errato Si è cercato di selezionare un modo operativo non ammesso. I modi operativi vanno da 1 a 8 e sono i seguenti: <ol style="list-style-type: none"> 1. MDI 2. AUTO 3. BLOCK by BLOCK 4. CONTINUOUS JOG 5. INCREMENTAL JOG 6. RETURN ON PROFILE 7. HOMING FILE 8. HPG
NC121	Numero asse selezionato fuori range Il numero di assi da selezionare per spostamenti manuali con la chiamata di libreria NC_SELAXI è errato. L' intervallo ammesso va da 1 al numero di assi configurato per quel processo. $1 < \text{intervallo ammesso} < \text{No. di assi configurati} + 1$
NC122	Troppi assi selezionati per movimento Nel blocco di part program sono stati inseriti troppi nomi di assi rispetto a quelli che sono accettati. Modificare il blocco di part program

Codice	Descrizione del messaggio ed azione di recupero
NC123	<p>Modo selezionato e ciclo incongruenti</p> <p>Questo errore viene visualizzato quando viene premuto un CYCLE START nelle seguenti condizioni del sistema:</p> <ul style="list-style-type: none"> • selezionata una modalità diversa da MDI durante l'esecuzione di movimenti assi relativi ad un cambio utensile. • stato HOLD e modo AUTO o BLK/BLK con MBR non configurato in AMP • stato HOLD con MBR attivo e modo selezionato diverso da AUTO o BLK/BLK • stato IDLE ed ACTIVE_RESET con modo selezionato diverso da AUTO o BLK/BLK • stato IDLE con MBR attivo e modo diverso da AUTO o BLK/BLK • stato HRUN con MBR attivo e modo selezionato diverso da AUTO o BLK/BLK • comando di ACTIVE RESET in stato di HOLD con modo selezionato diverso da MDI , AUTO o BLK/BLK. <p>N.B: Per maggiori informazioni sugli stati macchina consultare il Manuale d'Uso.</p>
NC124	<p>Nome asse non corretto</p> <p>Questo errore è visualizzato nei seguenti casi :</p> <ul style="list-style-type: none"> • il nome dell'asse da selezionare non è stato trovato nella tabella degli assi relativa a quel processo: non è stato configurato • la definizione del piano di lavoro è errata perché l'asse del piano da definire non è stato configurato • il piano da definire con G17, G18, G19, G16 non può essere definito perché l'asse specificato non è configurato • l'asse specificato nella chiamata di libreria NC_ACTUALOFS non esiste • l'asse specificato nei triletterali SCF, MIR non è configurato • errori di lettura quote asse, perché l'asse specificato non esiste. • l'asse specificato nei triletterali SOL, DPA, UDA, UGS, AXO, UAO, non è configurato o è duplicato. • l'asse specificato nella chiamata di Libreria AX_SHARE non esiste. • si sta rilasciando un asse condiviso con la logica attraverso il comando GTA.
NC125	<p>Lunghezza dato fuori range</p> <p>La lunghezza del buffer di tastiera per blocco in MDI è fuori intervallo. La lunghezza ammessa va da 1 a 127 caratteri.</p>
NC126	<p>Scrittura variabile non avvenuta</p> <p>Errore di scrittura del valore di una variabile.</p>
NC127	<p>Lettura variabile non avvenuta</p> <p>Errore di lettura del valore di una variabile.</p>
NC128	<p>Lettura limiti operativi errata</p> <ul style="list-style-type: none"> • Errore durante la definizione dei limiti operativi software con il triletterale SOL. • I limiti software programmati devono essere definiti entro software configurati. • I limiti operativi software non sono configurati in AMP.
NC129	<p>Area protetta non definita</p> <p>Si è cercato di abilitare con un'istruzione PAE un'area protetta non definita. Definire l'area protetta con DPA.</p>

Codice	Descrizione del messaggio ed azione di recupero
NC130	Correttore non definito <ul style="list-style-type: none"> • Si è cercato di pre-impostare un correttore non associato all'asse specificato. • Si è cercato di riqualificare un correttore di lunghezza non associato ad alcun asse.
NC131	Codice orientamento utensile errato Codice di orientamento dell'utensile errato nella chiamata a libreria NC_ACTUALOFS.
NC132	Errore dall'ambiente plus Errore dall'ambiente PLUS a seguito di chiamate di libreria "PLUS": PL_SET92, PL_RES92, PL_PRESCOR, PL_UAO, PL_UTO, PL_UIO, PL_RQT, PL_RQP, PL_RQO , durante le esecuzioni: RQO, UAO, UTO, UIO, RQT, RQP, G92, GTS.
NC133	Errore dall'ambiente Servo Errore dall'ambiente SERVO durante il presetting di un'origine o di un correttore L'errore può essere provocato anche dall'istruzione IPB quando il valore di In Position Band passato ad un asse viene arrotondato dai calcoli interni a meno di 1 digit.
NC134	Movimento manuale non eseguito: nessun asse configurato Non è configurato nessun asse; i movimenti manuali sono impossibili.
NC135	Asse non configurato L'id programmato nel triletterale GTA o GTS non è configurato Nel triletterale GTS è stato programmato un asse non mandrino
NC136	L'ID programmato identifica un asse ausiliario L'id programmato nel triletterale GTA identifica un asse ausiliario; non potrà perciò essere utilizzato.
NC137	Asse o mandrino non disponibile <ul style="list-style-type: none"> • L'id dell'asse programmato nel triletterale GTA appartiene ad un altro processo. • L'id dell'asse programmato nella funzione della Libreria AX_SHARE non è disponibile. • Richieste di condivisione di un asse mandrino con GTS non possibile.
NC138	ID asse duplicato Lo stesso ID asse è ripetuto più volte nella programmazione del triletterale GTA.
NC 139	L'ID programmato identifica un mandrino L'ID programmato nel triletterale GTA identifica un asse mandrino, non può essere quindi utilizzato.
NC140	Set rotazione mandrino fallita La logica di macchina (task \$SPROG) non accetta la variazione della velocità mandrino.
NC141	Richiesta nuovo utensile fallita La logica di macchina (task \$nTPROG) non accetta la programmazione del codice T.

Codice	Descrizione del messaggio ed azione di recupero
NC142	Esecuzione M fallita La logica di macchina (task \$mDECOD) non accetta la programmazione del codice M.
NC143	Programmazione pseudo-asse fallita La logica di macchina (task \$nPSEUDO) non accetta la programmazione dello pseudo asse.
NC144	Consenso al movimento negato Consenso al movimento negato dalla logica di macchina (task \$nCONMOV).
NC145	Richiesta fine movimento fallita La logica di macchina risponde con errore sulla segnalazione di fine movimento (task \$nENDMOV).
NC146	Troppi blocchi senza movimento in continuo Sono stati programmati troppi blocchi senza movimento in modalità dinamica continua.
NC149	Programma già selezionato Questo errore si verifica in modalità MDI quando si cerca di attivare più volte la stessa paramacro.
NC150	Asse azzerato Indica che l'azzeramento dell'asse è stato eseguito.
NC151	Asse sul profilo Segnala che l'asse è sul profilo e che il "ritorno sul profilo" è terminato correttamente.
NC152	Fine ritorno automatico sul profilo L'operazione di "ritorno sul profilo automatico" è terminata correttamente e tutti gli assi sono sul profilo.
NC153	Fine del block retrace L'esecuzione all'indietro dei blocchi con MBR è terminata. Per eseguire all'indietro un maggior numero di blocchi è necessario modificare la configurazione.
NC156	Fine della ricerca in memoria Fine ricerca memorizzata.
NC160	Comando e stato sistema incongruenti Comando non ammesso nello stato attuale del sistema.
NC161	Errore interno : classe inesistente
NC162	Errore interno : messaggio errore nc Spegner e riaccendere il controllo: se l'errore permane, contattare l'assistenza tecnica.

Codice	Descrizione del messaggio ed azione di recupero
NC190	Lungh. insufficiente per la maschiatura Questo errore si verifica quando la lunghezza percorsa nella fase di accelerazione e decelerazione del ciclo fisso di maschiatura senza trasduttore, supera la lunghezza da percorrere e non viene lasciato spazio per la lavorazione.
NC191	Lungh. insufficiente per la maschiatura con trasduttore Questo errore si verifica quando la distanza percorsa nella fase di accelerazione e decelerazione del ciclo fisso di maschiatura con trasduttore sul mandrino supera la lunghezza da percorrere e non viene lasciato lo spazio per la lavorazione.
NC192	Lunghezza insufficiente per il ciclo Questo errore si verifica quando la distanza percorsa nella fase di accelerazione e decelerazione nel ciclo fisso di filettatura supera la lunghezza da percorrere e non viene lasciato spazio per la lavorazione.
NC199	Mandrino non attivato
NC200	Errore accesso file Errore durante la lettura/scrittura di un file.
NC201	Errore nel caricamento del file setup Errore durante la fase di caricamento del file di setup TE_CONFIG.TXT.
NC202	Errore in configurazione file Dual Port La configurazione degli assi contenuta nel file a cui si accede è diversa da quella presente nella dual port.
NC203	Attenzione: tabella di sola lettura Accesso alla tabella nella dual port negato da PLUS.
NC204	Grandezza file illegale La tabella (su file) ha dimensioni errate.
NC205	Magazzino vuoto Il magazzino selezionato non ha pocket definite.
NC206	Pocket occupata La pocket definita per un utensile è già occupata o riservata per un altro utensile.
NC207	Pocket precedente illegale Un utensile che occupa più pocket interferisce con la pocket occupata da un altro utensile (pocket precedente).
NC208	Pocket successiva illegale Un utensile che occupa più pocket interferisce con la pocket occupata da un altro utensile (pocket successiva).
NC209	Random illegale E' stata trovata in memoria una classe random non ammessa.

Codice	Descrizione del messaggio ed azione di recupero
NC210	Tabella utensili piena Dual port piena durante una funzione LOAD di una Tabella Utensili associata ad un determinato magazzino.
NC211	Doppio formato per editor illegale E' stata trovata un variabile il cui formato non è accessibile all'editor (max 5.5).
NC212	Numero di magazzino illegale nel file Il numero di magazzino specificato nel file Tool si riferisce ad un magazzino inesistente.
NC213	Pocket non inizializzata Pocket non inizializzata.
NC214	Pocket non compatibile Le pocket associate agli utensili non sono congruenti con quelle attualmente in uso.
NC215	Nome tabella illegale Il nome della tabella da caricare non è valido. Controllare che l'estensione del nome della tabella sia tra quelle consentite: .TOL .USR .MAG .OFS .ORG .SPN
NC220	Processo non definito Processo non definito o non configurato. Definire il processo di default con il comando PRO o selezionare un processo esistente per i comandi di sincronizzazione.
NC221	Tipo processo errato Utilizzato un canale di comunicazione non adatto per il comando impostato. Esempio: canale tipo 2 (PLUS) per esecuzione comando EXE.
NC222	Numero di processo errato Il numero di processo specificato per i comandi di sincronizzazione identifica il processo corrente.
NC223	Coda processo errata La coda del processo (locale o remoto) al quale è stato inviato un messaggio, è piena.
NC224	Invio dati troppo lungo I dati che si vogliono trasmettere con il comando SND superano i 174 caratteri.
NC225	Caricamento dati fallito Il tipo o il numero di dati trasmessi col comando SND non è congruente con quello atteso.
NC226	Messaggio ancora in coda Si è eseguito un comando SND verso un processo, in un momento in cui quest'ultimo non aveva ancora utilizzato il messaggio che gli era stato inviato precedentemente.

Codice	Descrizione del messaggio ed azione di recupero
NC227	Comando EXE o ECM fallito <ul style="list-style-type: none"> Il processo a cui è rivolto il comando (EXE o ECM), è in uno stato non congruente per l'esecuzione automatica di un part program (es. RUN, HRUN, RUNH, HOLD) o del comando impostato in MDI.. Il programma a cui è rivolto il comando "EXE" ha un errore di sintassi.
NC290	Attivazione programma rifiutata La logica di macchina ha rifiutato l'attivazione di un part program.
NC291	Disattivazione programma rifiutata La logica di macchina ha rifiutato il rilascio del programma attivo.
NC292	Richiesta acquisizione assi rifiutata La logica di macchina ha rifiutato il rilascio di alcuni assi durante l'esecuzione del triletterale GTA.
NC293	Richiesta rilascio assi rifiutata La logica di macchina ha rifiutato l'acquisizione di alcuni assi durante l'esecuzione del triletterale GTA.
NC294	Richiesta acquisizione o condivisione mandrino rifiutata La logica di macchina ha rifiutato l'acquisizione o la condivisione di un mandrino tramite il triletterale GTS.
NC295	Richiesta rilascio mandrino rifiutata La logica di macchina ha rifiutato il rilascio di un mandrino tramite il triletterale GTS.
NC296	Richiesta modifica condivisione mandrino rifiutata La logica di macchina ha rifiutato la modifica dello stato di condivisione di un mandrino tramite il triletterale GTS.
NC297	Ciclo fisso non possibile con mandrino condiviso E' stata effettuata la programmazione di un ciclo G84 o G86 con il mandrino impostato in modalità non esclusiva.
NC320	Programmazione UPR non disponibile Non è permessa la programmazione del comando UPR con altre virtualizzazioni attive. Questo errore viene visualizzato anche in caso in cui: <ul style="list-style-type: none"> Uno degli assi reali risulta essere SLAVE in una programmazione UDA/SDA. Viene programmato un UPR di tipo 5 o minuscolo e non è attivo un UPR precedente. E' settata a 1 la variabile utente !R73 MODE e vengono programmate le origini sugli assi rotativi nell'UPR oppure viene programmato un UPR di tipo 5. Vengono programmate le origini sugli assi rotativi in un UPR di tipo Ø, 1 o 10.
NC321	Errata programmazione incrementale UPR La programmazione del comando UPR incrementale è ammessa solo con UPR già attivo.

Codice	Descrizione del messaggio ed azione di recupero
NC322	Programmazione UVP non disponibile Non è permessa la programmazione del comando UVP con altre virtualizzazioni attive. Questo errore viene visualizzato anche in caso in cui uno degli assi reali risulta essere slave in una programmazione UDA/SDA.
NC323	Tipo asse errato con programmazione UVP Il tipo degli assi reali programmati nel comando UVP non è compatibile con la virtualizzazione stessa.
NC324	Valore raggio errato Il valore del raggio nella programmazione del comando UVP non è compatibile con la posizione dell'asse lineare.
NC325	Programmazione UVC non permessa Non è permessa la programmazione del comando UVC con altre virtualizzazioni attive. Questo errore viene visualizzato anche quando l'asse reale risulta essere slave in una programmazione UDA/SDA.
NC326	Codice TCP programmato fuori range Il codice di attivazione del comando TCP non è tra quelli previsti.
NC327	Programmazione TCP non permessa Non è permessa la programmazione del comando (TCP,5) con altre virtualizzazioni attive. Questo errore viene visualizzato anche nel caso in cui uno degli assi lineari o rotativi del TCP risulta essere uno slave in una programmazione UDA/SDA.
NC328	Programmazione TCP non congruente La richiesta di attivazione del TCP non è compatibile con il tipo di TCP attivo.
NC329	Errore di attivazione TCP Errore durante l'attivazione del comando (TCP,4), verificare la presenza degli identificativi degli assi nella user table.
NC330	Errore nel prendere o rilasciare l'asse Non è possibile attivare il comando GTA in presenza di: Offset, Cicli fissi, Virtualizzazioni.
NC331	Clock di interp. assi non congruente Uno o più assi oggetto del comando GTA o GTS sono stati caratterizzati con clock interpolatore diverso da quello del processo corrente.
NC332	Valore zero dei moduli i j k Programmazione errata di ijk con TCP attivo: il modulo dei valori risulta uguale a zero.
NC333	Programmazione di ijk,mno errata Errata programmazione di ijk e/o mno.

Codice	Descrizione del messaggio ed azione di recupero
NC334	Troppi blocchi nel profilo Il numero massimo di blocchi definito in AMP per la contornitura automatica o per i cicli di sgrossatura è inferiore a quello necessario.
NC340	Cerchio/linea non definito Durante la definizione/esecuzione di un profilo GTL, si fa riferimento ad un cerchio/retta non definiti.
NC341	Definizione di cerchio/linea errata Errore nella definizione di rette/cerchi in un profilo GTL.
NC342	Mancata intersezione di Cerchio/linea Mancata intersezione tra cerchi/rette nella definizione di un profilo GTL.
NC343	Cerchi coincidenti Durante la definizione di un profilo GTL viene richiesta l'intersezione tra due cerchi coincidenti.
NC344	Cerchi/linee/punti coincidenti Errore nella definizione di un profilo GTL dovuto alla presenza di cerchi/rette /punti coincidenti.
NC345	Punto nel cerchio Errore nella definizione di un profilo GTL dovuto alla presenza di un punto definito all'interno di un cerchio.
NC346	Linee parallele Errore nella definizione di un punto cerchio dovuto alla presenza di rette parallele.
NC347	Punti allineati Errore nella definizione di un cerchio dovuto alla presenza di punti allineati.
NC360	Troppi blocchi di movimento Si è raggiunto il numero massimo di blocchi di movimento permesso all'interno di un profilo richiamato da una macro di sgrossatura (SPA, SPF). Controllare il valore di questo limite impostato in AMP.
NC361	Errore nel profilo Il profilo richiamato dalla macro di sgrossatura (SPA, SPF) non è sgrossabile. In generale, sono sgrossabili soltanto i profili monotoni per l'asse di sgrossatura (cioè X o Z sempre decrescenti oppure sempre crescenti).
NC362	Area di lavoro non definita Spegner e riaccendere il controllo: se l'errore permane, contattare l'assistenza.
NC363	Piano di interpolazione e assi incongruenti Nelle macro di sgrossatura (SPA, SPF) l'asse di sgrossatura deve appartenere al piano di interpolazione, così come gli assi per i quali è definito il sovrametallo. Anche nella macro di filettatura, l'asse del filetto e l'asse di ritorno devono appartenere al piano di interpolazione.

Codice	Descrizione del messaggio ed azione di recupero
NC364	Attacco errato al profilo Punto di attacco non permesso per la macro di sgrossatura (SPA, SPF). Il punto di attacco deve sempre essere esterno al campo di sgrossatura in X, per sgrossature parallele all'asse X, ed esterno al campo di sgrossatura in Z, per sgrossature parallele all'asse Z.
NC365	Tipo di interpolazione non permessa Nel profilo richiamato dalla macro di sgrossatura (SPA, SPF) sono ammessi soltanto blocchi di movimento di tipo lineare o circolare.
NC366	Punti allineati in sgrossatura Durante la fase di sgrossatura del profilo si è raggiunta una zona non sgrossabile. Controllare la coerenza del profilo e i parametri della macro.
NC367	Attacco al profilo non corretto Il punto di attacco e la direzione di sviluppo del profilo non permettono di proseguire la sgrossatura.
NC370	Parametri R o B mancanti Nella macro di taglio gola non è accettata la presenza del raccordo o smusso iniziale o finale nel caso in cui non sia stata programmata la quota esterna.
NC371	Utensile più grande della gola Errore nella macro di taglio gola dovuto al fatto che la larghezza dell'utensile è maggiore della larghezza della gola.
NC372	Utensile non congruente con i parametri R o B Errore nella macro di taglio gola dovuto al fatto che la larghezza utensile è nulla oppure è minore della somma dei raccordi o smussi programmati.
NC373	Posizione iniziale per TGL errata La posizione di attacco per la macro di taglio gola non è coerente con i parametri dichiarati nel blocco.
NC375	Parametri 'a' e/o 'b' mancanti Se la filettatura è stata programmata di tipo non standard, devono essere presenti nel blocco anche i parametri "a" e "b".
NC376	Passo errato in filettatura Nel caso di filettatura non standard è necessario che il passo programmato rispetti la seguente formula. Deve essere: $\text{Passo} \geq 2 * \text{Profondità filetto} * \text{tg} \frac{\text{angolo filetto}}{2} * \text{numero principi.}$
NC377	Angolo del filetto più grande di 180° Errore nella macro di filettatura dovuto all'angolo del filetto $\geq 180^\circ$

Codice	Descrizione del messaggio ed azione di recupero
NC378	Lunghezza della filettatura nulla Errore nella macro di filettatura se la lunghezza del filetto lungo l'asse mandrino è nulla.
NC379	Angolo cono errato Nel caso di filettatura conica, la massima conicità ammessa è uguale alla metà dell'angolo del filetto.
NC380	Piano di rotazione non permesso Non è permesso eseguire un ciclo di filettatura se c'è rotazione attiva per il piano di interpolazione.
NC381	Uscita circolare non permessa senza il parametro 'r' Errore nella macro di filettatura dovuto alla programmazione di un'uscita con raccordo senza valore del raggio.
NC401	HSM Part program non trovato o errore in apertura Possibile errore nella condivisione del part program tra i moduli esecutivi del controllo. Ricaricare il programma oppure contattare assistenza tecnica.
NC402	Errore durante lettura part program HSM Part program in esecuzione rovinato Spegner e riaccendere il controllo oppure contattare assistenza
NC403	HSM file di configurazione non trovato o errore in apertura File di configurazione non presente. Verificare presenza del file ed il triletterale (HSM, che ne definisce il nome)
NC404	Errore di sintassi nel file di configurazione HSM alla linea La linea specificata contiene un errore di sintassi. Verificare sintassi del triletterale di configurazione sul manuale.
NC405	Posizione iniziale richiesta per tutti gli assi configurati in HSM Il primo punto programmato dopo la G61 deve contenere tutti gli assi associati alla configurazione HSM. Programmare tutti gli assi eventualmente mancanti riconfermando le eventuali quote che non cambiano.
NC406	Parametri HSM richiesti nel file di configurazione alla linea Il triletterale di configurazione impostato alla linea specificata richiede ulteriori parametri Verificare sintassi del triletterale di configurazione sul manuale.
NC407	Errore nei parametri HSM Il triletterale di configurazione impostato alla linea specificata non contiene un parametro obbligatorio Verificare sintassi del triletterale di configurazione sul manuale.
NC408	Parametri HSM scritti in posizione errata Riservato a futuri sviluppi. Contattare assistenza tecnica.

Codice	Descrizione del messaggio ed azione di recupero
NC409	Parametri HSM non permessi nel Part Program Riservato a futuri sviluppi Contattare assistenza tecnica.
NC410	Sono richiesti almeno due punti per definire un segmento HSM Tra i codici G61 e G60 devono essere presenti almeno due punti. Modificare il part program o non usare G61/G60.
NC411	Assi definiti in HSM non presenti tra gli assi del processo. Alla linea Il triletterale di configurazione impostato alla linea specificata fa riferimento a un identificatore asse non associato al processo su cui viene eseguito il part program Verificare il triletterale di configurazione sul manuale oppure gli identificatori degli assi associati al processo.
NC412	Parametri generali di configurazione HSM devono essere definiti prima dei parametri assi Nel file di configurazione prima devono essere definiti i triletterali generali, poi quelli di configurazione assi. Verificare sequenza di configurazione sul manuale.
NC413	Parametri di configurazione assi HSM devono essere scritti dopo i parametri generali. Alla linea Nel file di configurazione prima devono essere definiti i triletterali generali, poi quelli di configurazione assi Verificare sequenza di configurazione sul manuale.
NC414	Sono richiesti ulteriori parametri di configurazione HSM Nel file di configurazione mancano dei triletterali di configurazione Verificare sequenza di configurazione sul manuale.
NC415	Sono richiesti ulteriori parametri per gli assi di configurazione in HSM Nel file di configurazione sono stati specificati degli assi a cui mancano ulteriori triletterali di configurazione. Verificare sequenza di configurazione sul manuale.
NC416	Sono richiesti ulteriori parametri utensili HSM Nel file di configurazione non sono stati specificati i triletterali di configurazione dell'utensile Verificare sequenza di configurazione sul manuale.
NC417	Sono richiesti ulteriori parametri cinematici HSM Nel file di configurazione non sono stati specificati i triletterali di configurazione della terna assi riferita all' utensile e della terna assi riferita allo staffaggio del pezzo Verificare sequenza di configurazione sul manuale per i triletterali CIN,t e CIN,w.
NC418	Manca la configurazione assi HSM alla linea Nel file di configurazione alla linea specificata e' stato configurato un asse non precedentemente definito col triletterale AXI, Verificare sequenza di configurazione sul manuale

Codice	Descrizione del messaggio ed azione di recupero
NC419	Asse già definito nella configurazione HSM alla linea Nel file di configurazione alla linea specificata è stato definito un asse già definito in precedenza. Verificare sequenza di configurazione sul manuale.
NC420	Troppi assi aggiuntivi (max 3) nella configurazione HSM Nel file di configurazione possono essere definiti al massimo 3 assi aggiuntivi (assi non appartenenti alla terna cartesiana o rotativi) Verificare sequenza di configurazione sul manuale.
NC421	Troppi assi (max 6) nella configurazione HSM Nel file di configurazione possono essere definiti al massimo 6 assi Verificare sequenza di configurazione sul manuale.
NC422	Tipo asse errato nella configurazione HSM Nel file di configurazione è stato definito un tipo asse non congruente o precedentemente associato ad altri assi. Verificare sequenza di configurazione sul manuale relativamente al triletterale AXI,.
NC423	Limiti operativi raggiunti nel programma HSM per l'asse Sono stati raggiunti i limiti operativi software per l'asse specificato. Verificare part program.
NC424	Virtualizzazioni o TCP non devono essere attivi con HSM All'atto della attivazione della G61, non devono essere attive le virtualizzazioni od il TCP. Verificare part program.
NC425	Errore in lettura file di configurazione HSM File di configurazione rovinato. Ricaricare il file di configurazione oppure contattare assistenza tecnica.
NC426	HSM non abilitato in AMP La prestazione HSM non è stata abilitata in AMP. Abilitarla.
NC427	HSM non disponibile come opzione L'opzione HSM non è stata abilitata. Per utilizzare la prestazione con un numero di assi maggiore a 3 occorre abilitare l'opzione a livello di Product Key Contattare assistenza tecnica.
NC428	Opzione HSM non caricata L'opzione HSM è stata abilitata a livello di Product Key ma non è stata caricata sul CN. Caricare l'opzione..
NC429	Parametro errato nella configurazione HSM alla linea Nel file di configurazione alla linea specificata è stato definito un parametro con valore sbagliato (deve essere positivo). Verificare sequenza di configurazione sul manuale.

Codice	Descrizione del messaggio ed azione di recupero
NC430	Valore di Feed errato o non definito nel Part Program HSM Manca il valore della Feed od e' mionore o uguale a 0. Impostare un valore corretto di Feed.
NC431	Errore di sintassi nel Part Program HSM Errore sintattico nel part program durante una lavorazione HSM. Correggere programma e vedere manuale di programmazione relativamente ai blocchi ammessi tra G61 e G60.
NC432	Errato uso dell'asse tangente in HSM Puo' essere presente un solo asse tangente oppure errato uso dell'asse tangente. Verificare sequenza di configurazione sul manuale.
NC433	Modalità di parametrizzazione errata Programmazione del tipo di punti e relativa parametrizzazione non corretta nel triletterale PNT del file di setup dell'alta velocità.
NC434	La programmazione polinomiale non ammette la parametrizzazione richiesta La configurazione del triletterale PNT nel file di setup dell'alta velocità non è corretta per la programmazione polinomiale.
NC435	I nodi devono essere programmati in modo crescente I nodi delle Bspline programmate in input devono essere ordinati in senso crescente.
NC436	E' richiesta la programmazione del nodo Non si è programmato un numero sufficiente di nodi: nel caso di input con Bspilne il numero di nodi deve essere pari al numero dei punti di controllo più il grado della Bspline + 1.
NC437	Deve essere confermato il punto finale della Bspline precedente Le Bspline programmate devono essere continue ossia l'ultimo punto di una Bspline deve essere uguale al primo punto della successiva.
NC438	Mancano dei punti di controllo per definire correttamente la Bspline Il numero minimo di punti di controllo, per input con Bspline, è pari a: (grado della Bspline + 1) *2.
NC439	I polinomi programmati mancano di continuità I polinomi programmati devono essere continui ossia l'ultimo punto di un polinomio deve essere uguale al primo punto del successivo.
NC440	Vettore IJK di tipo ROT non valido richiedendo calcolo di assi Tangenti Se si lavora con un asse tangente il vettore ijk non deve essere di tipo ROT nel triletterale VER del file di setup dell'alta velocità.
NC441	Vettore IJK di tipo ROT non valido perché si usano i parametri di (TOD) Nel triletterale VER, del file di setup dell'alta velocità, il vettore ijk non può essere impostato a ROT quando ad es. si hanno meno di due rotativi nella catena cinematica.

Codice	Descrizione del messaggio ed azione di recupero
NC442	Vettore IJK di tipo PRG non valido programmando punti di tipo AXI/CLP Se la programmazione dei punti è di tipo (PNT,AXI,CLP,... oppure (PNT,AXI,CCP,... i vettori ijk non devono essere programmati; utilizzare una delle seguenti programmazioni: (VER,REL/ROT,...
NC443	Vettore MNO di tipo REL non valido programmando punti di tipo AXI/CCP Se la programmazione dei punti è di tipo (PNT,AXI,CCP,... oppure (PNT,CLP,CCP,... i vettori mno devono essere programmati, utilizzare questa programmazione: (VER,...,PRG,...
NC444	Asse condiviso con l'ambiente PLUS L'asse che si sta cercando di muovere o sul quale si vuole effettuare una virtualizzazione è stato precedentemente acquisito dalla logica con la funzione AX_SHARE.
NC445	Unità di misura macchina non congruente in HSM Questo errore si verifica durante l'esecuzione di un part program ottimizzato con Path Optimizer quando l'unità di misura specificata nel file di setup non è congruente con quella della macchina.
NC446	Programmazione mno/uvw errata in HSM file Questo errore si verifica quando entrambi i vettori mno/uvw sono definiti, programmati o calcolati nel triletterale VER del file di setup dell'alta velocità.
NC447	Programmazione uvw non ammessa con utensile toroidale I fattori di compensazione uvw non sono gestiti con utensili toroidali.
NC448	Spigolo per il calcolo di uvw non valido Questo errore si verifica quando si richiede di calcolare i fattori di compensazione uvw in un punto in cui si è verificata una sovraelongazione. Avviene cioè un ribaltamento di 180° dal punto di contatto sull'utensile.
NC456	Versore ijk nullo E' stato programmato un versore ijk di modulo nullo
NC457	Versore mno nullo
NC458	Versore pqd nullo

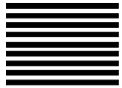
FINE APPENDICE

GESTIONE ERRORI DA PART PROGRAM

GENERALITÀ

Questa appendice descrive la funzionalità che consente al programmatore di gestire nel part program alcuni errori riscontrati dal sistema eseguendo specifiche istruzioni in modo tale da non bloccare l'esecuzione del part program stesso.

VARIABILE DI SISTEMA	FUNZIONE
ERR	Abilita la gestione degli errori nel part program
STE	<p>Contiene l'errore generato dal comando precedente (se è attiva la gestione automatica (ERR=1)).</p> <p>E' una variabile di sistema di sola lettura.</p>



ERR - Abilita/Disabilita la gestione errori nel Part Program

La variabile di sistema ERR abilita o disabilita la gestione di alcuni errori riscontrati dal sistema seguendo specifiche istruzioni.

Sintassi

ERR = *valore*

dove:

valore Può essere un numero od una variabile locale o di sistema e deve valere 0 o 1.
Se vale 0, la gestione degli errori nel part program è disabilitata e qualsiasi errore riscontrato blocca l'esecuzione del programma stesso; l'errore viene quindi visualizzato. **Se vale 1**, la gestione da part program degli errori è attiva.

Caratteristiche:

Ponendo ERR=1, la gestione da Part Program degli errori è abilitata.

All'accensione il sistema si predispose con ERR=0.

Le pagine seguenti descrivono:

- I comandi di programmazione che possono generare uno degli errori gestibili nel part program.
- I codici numerici relativi a questi errori (il valore della variabile STE) e relativo significato.
- Il numero dell'errore NCxx che sarebbe visualizzato nel caso di ERR=0. (Vedi appendice B).

Errori durante il ciclo di tastatura

Comando:		Valore della variabile STE	Errore visualizzato se ERR=0
G72,G73,G74	0	ciclo di tastatura terminato regolarmente	-----
	1	tastatura non avvenuta	NC098
	2	tastatore non rilasciato	NC099
	3	parametri tastatore non specificati	NC007
	4	tastatura avvenuta nella fase di avvicinamento rapido	NC115

ESEMPIO:

Verifica della presenza del pezzo utilizzando il metodo della gestione dell'errore:

```
(DPP,30,15,500)
G0 Z50
X80 Y100
ERR = 1
G72 Z0 E1
(GTO,END,STE = 1)
ERR = 0
.....
.....
"END" (DIS, "PEZZO NON PRESENTE")
M...
.....
```

IMPORTANTE

G72, G73, G74 modificano il valore della variabile STE (se ERR=1); è quindi consigliabile utilizzare la variabile STE subito dopo il comando che ne va a modificare il valore.

Errori nella gestione "Assi Migranti"

Comando:		Valore della variabile STE	Errore visualizzato se ERR=0
Triletterale GTA	0	esecuzione senza errori	-----
	10	ID asse non configurato	NC135
	11	ID asse appartenente alla logica	NC136
	12	ID asse appartenente ad un altro processo	NC137
	13	ID asse appartenente ad un asse mandrino	NC139

Esempio di programmazione GTA con gestione errore:

```

;.....
;.....
ERR = 1
;.....
"RETRY"
(GTA,X1,Y2,Z5)      ;Richiede acquisizione ID 1,2,5
(GTO,NEXT,STE<=>12)  ;Test errore asse occupato.
(DLY,0.5)            ;Se l'asse è occupato attende che si liberi
(GTO,RETRY)
"NEXT"
(GTO,ERROR,STE<=>0);In caso di altro errore abortisce l'operazione
G1 X10 Y10 F1000      ;Ad acquisizione avvenuta effettua la movimentazione
Z50
;.....
;.....
;.....
"ERROR"
;.....                ;Gestione dei casi di errore

```

IMPORTANTE

- La gestione dell'errore da part program a cui si fa riferimento nell'esempio, rappresenta anche un modo semplice per sincronizzare due processi, qualora un processo debba attendere che uno o più assi si liberino sull'altro, prima di eseguire una data lavorazione.
- GTA modifica il valore della variabile STE (se ERR=1); è quindi consigliabile utilizzare la variabile STE subito dopo il comando che ne va a modificare il valore.

FINE APPENDICE