NC-110, NC-210, NC-201M (TC)

PROGRAMMING AND OPERATOR'S MANUAL

Saint-Petersburg
2008

PUBLICATION ISSUED BY:

Balt-System Ltd.
198206, Sankt-Petersburg
Petergofskoje Shosse, 73

Telefax: +7(812)744 70 59
Telephone: +7(812)744 70 59

Important User Information

This document has been prepared in order to be used by Balt-System Ltd. customers. It describes the latest release of the product.

Balt-System Ltd. reserves the right to modify and improve the product described by this document at any time and without prior notice.

Actual application of this product is up to the user. In no event will Balt-System Ltd. be responsible or liable for indirect or consequential damages that may result from installation or use of the equipment or software described in this text.

This manual describes the characteristics and tells you how to use the NC-110.

We have organized the contents of this document in nine chapters and one appendix.

Chapter 1 draws an outline of the product.

Chapter 2 describes the available operator panels.

Chapter 3 describes the function and alphanumeric keyboards.

Chapter 5 lists the complete set of programming codes available with the present release.

Chapter 6 explains you how to compile your part programs.

Chapter 7 deals with the ASSET utility.

Chapter 8 describes basic system operation.

Chapter 9 explains you how to use peripheral devices.

The appendix lists and describes the error messages that may appear on the CRT.

1. INTRODUCTION

Chapter Objectives

This chapter gives you an overview of the NC-110 operating features, programming and configurations. After reading this chapter, you should have a good idea of what the NC-110 can do for you.

1.1. OPERATING FEATURES

Axes

- 3 axes for linear interpolation
- 1 spindle
- Circular interpolation applicable to any pair of axes.
- Helical interpolation.
- Possibility to combine circular interpolation, with linear and rotary movements.
- Max. radius: 99.9999 meters.
- Interpolation accuracy: 1 micron/m
- Transducers: encoders, resolvers, inductosyns (resol:0.1 um), optical lines.
- Automatic control of vectorial speed on profiles.
- Acceleration and deceleration control during circular interpolation.
- Automatic deceleration on corners.

- Dynamic speed optimization on profiles (look ahead).
- Gonfigurable storage buffer (64 blocks max.) for continuous machining.

Operator panel

The operator panel constitutes the man-machine interface.

Features a flat-foil keyboard, a 9" CRT, and a console including: on and off keys, potentiometers for spindle speed and feedrate and for manual control of jog direction and speed, operational mode selectors, feed hold, cycle start and reset pushbuttons.

CRT

9" cathode ray tube for alphanumeric and graphic display.

Alphanumeric Display

This mode allows you to display editing procedures, lists of programs, origins. tool offset and tool life tables. To switch between the available screens, press the **P1** key; the allmwable options are:

Screen 0 - offers a synthesis of the process state

Screen 1 - displays currrent and programmed dimensions, G, T, S, M, H functions, origins, offsets, power absorption, current block number, repeat cycles, subroutines, etc.

Screen 2 - is the user screen, available for menu definition

Graphic display

The first 4 lines of the graphic screen display exactly the same information appearing on the first 4 lines of the alphanumeric display, but in a smaller body size.
During machining, the remaining part of the screen displays the cartesian axes, the programmed dimensions and the profiles, the points at which fixed cycles are executed and the moves of the axis perpendicular to the machining plane.

Program loading and editing

You can load part programs either directly via keyboard or from external peripheral devices (tape reader, magnetic cassette, teletype).
The size of the user memory is 32K characters.
Once loaded on user memory, a program can be displayed on the CRT and edited (i.e. by deleting, modifying or inserting program blocks). Edited programs can be punched out on tape or stored on magnetic cassette.
Program loading and editing can be carried out during machining.

Operating modes

According to the model, you can select the desired operating mode by means of the selector or key on the operator panel.
The allowable options are:
- execution of blocks input via meyboard;
- automatic program execution;
- semiautomatic program execution;
- manual continuous jog;
- manual incremental jog;
- automatic restart after a cycle stop followed by manual joggging.

Electronic Handwheel	It allows you to manually move the axes. The available scales are: • 1 mm/turn if the selector is on the manual position; • 0.1 mm/turn if the selector is on the incremental position.
Program test	By inputting commands from the keyboard you can: • test programs with axis standstill, and graphic or alphanumeric display; • carry out the program "dry run", with prefixed high machining feedrates; • execute the program with controlled rapids, even reducible to zero, to machine the first piece.
Machine zero	One of the limit switches of each axis is used to automatically set the zero reference. At power-on each axis is sent to its limit switch, and the nearest transducer pitch becomes the absolute machine zero.
Hold	This feature allows you to stop axes movement with controlled deceleration.
Backlash Compensation	This feature allows you to automatically compensate backlash on motion reversal. The backlash value is stored in the system memory.
Geometric error Compensation	The control automatically compensates the dimensions measured by the position transducer, computing incidental errors noticed on the axis. You can establish as many as 1000 compensation points on each axis.
Split Axes	This feature allows you to move heavy parallel axes. During configuation, define one of them as the "master", considering the other as a "split" axis. Then only program the moves of the master; the movements of the split axis will be dependent on those of the master.

Transducers For spindle: resolver or encoder.
For axes: resolver, encoder, inductosyn or optical lines.

Temporary origins The code UOT allows you to program an unlimited number of temporary origins, referred to any absolute origin.

Incremental Origins The mnemonic code UIO allows you to program an unlimited number of incremental origins, i.e. referred to the origin that was valid when you specified UIO.

UIO is normally associated to the RPT (reapeat) code.

Tool offsets Tool offsets must be defined during installation.
Each offset contains the tool length on axes X and Z, the tool radius and the tool code (from 0 to 8). You can define an unlimited number of tool offsets.

Length values can be either input via keyboard or automatically calculated by the control (tool presetting).
Diameter offset values must be input via keyboard.
Gurrent offset values can be displayed or modified at any time. They can be modified from a program, after a probing cycle.

If the system features a probe for tool requalification, the tool offsets table can include additional information, such as the current offset value and the allowable limit value for both axes. When a tool exceeds the stored limit, the control declares it unusable.

Tool inspection Cycle After having established a HOLD, you can check the tool, manually jog the axes off the piece and eventually return to

the hold point.
You can select to return either automatically (JOG RETRACT) or in manual mode (axis by axis, RAP=O). In either case, the axis repeats the path performed during the removal (RAP=1).
You can store as many as 32 displacements.

Electronic probe

This omnidirectional measurement device, is mounted on the spindle. It is handled as a tool, with length and diameter offsets.
To enable the electronic probe, program code DPT followed by the necessary probing parameters (approach distance, safety distance and measurement speed).
During a machining program, the probe establishes:
- the coordinates of a point in the space (with G72);
- the variance between nominal and measured coordinates (with G74).

The resulting values are stored in the E variables contained in the measuring block.

Tool life

Within a machining cycle you can define a life monitoring cycle for every tool.
If the tool is worn out or broken, you can replace it with an alternate tool.
In order to enable tool life monitoring you must provide the control with the following information:
- tool number;
- alternate tool number;
- alternate tool offset;
- maximum theoretical life;
- minimum theoretical life;
- remaining life;
- tool condition.

Multiblock Metrace

This feature allows you to retrace the profile, going back to a desired position. You can go back by as many as 64 program blocks. The desired limit must be established during characterization.

Controlled search

The control permanently stores data related to the working cycle. This allows you to stop and automatically restart program execution at the exact point you stopped it, eyen at critical moments such as conditioned jumps, repetitions, subroutine calling, complex cycles etc.
The automatic search codes are RCM (start controlled search) and ERM (end of search).
The control simulates program execution up to the stop point, recalls the appropriate tool, sets offset values and displays the actual and desired tool positions (respectively, on partition 3 and 1 of the CRT).
To restart program execution, you must first reposition the axes and then press CYCLE START.

Types of memory

The system ambient parameters are stored on EPROM (Erasable Programmable Mead Only Memory).
The EPROM content can be modified or updated by Balt-System Ltd. engineers only.
The machine tool parameters (e.g. speed, acceleration, etc.), tool length and diameter offset values, origins and part programs, are stored on CMOS RAM. In case of power removal, the RAM content remains unchanged for 1000 hours.

Non-permanent data are stored in read/write RAMs, the content of which is lost at each power-off.

Spindle power Absorbtion

The percentage of maximum power absorbed by the spindle motor can be measured by a device installed in the electric cabinet. The current values are displayed on the CRT.

Feedrate and spindle speed override

The control panel features two potentiometers allowing you to change:
• feedrate from 0 to 125%;
• spindle speed from 75 to 125%.

Protection and Autodiagnostics

Both hardware (central processing units, cables, position transducers, etc.) and environment conditions (internal temperature, power supply, input data parity and memory capacity overflow, keyboard commands, etc.) are permanently monitored by the control, which handles servomechanisms as well.
In case of failure or error, a diagnostic message is displayed, specifying where the dysfunction to be corrected has been noticed.
Diagnostic messages are stored in the systems characterization files. You can edit and translate them into the desired language.

1.2. PROGRAMMING FEATURES

Measuring units	You can select metric/inch programming by executing functions G70/G71.
Incremental/ Absolute Programming	The preparatory function modes are: G90: absolute programming G91: incremental programming
Programming referred to machine zero	G79 allows you to reference all the co-ordinates to machine zero.
Decimal point Programming	Decimal point programming allows you to omit leading and trailing zeroes. <u>Example</u>: X-20.275
Tape code	EIA RS-244, ISO 840 with automatic acknowledge.
Programming format	N4. GZ. X/Y/Z/A/B/G/U/W/V/P/Q/D/5.4,R5.4 I/J/K5.4, F5.2, S5, T4.4, M2, H2
Axes coordinates	Inch or metric programming from ±0.0001 to ±99999.9999.
I J coordinates	They allow you to program centre coordinates in circular interpolation. Legal values range from ±0.0001 to ±99999.9999 millimeters or inches.
Rotary movements	During characterization, any axis can be configured as rotary. The allowable values range from ±0.0001 to ±99999.9999 degrees.
F functions	An F function can be programmed from 0.01 to 99999.99 mm/min. With G94, you can specify the execution time "t" (in sec) of the element declared in the block, i.e. the F of the block is the quotient between the length of the declared element and the programmed t.

G93 specifies the inverse of time expressed in min (that is, the ratio between feedrate and distance.
G95 specifies axis speed in mm/rev

S functions

You can program an S function from 1 to 99999. The S function can have two different uses, dependent on the declared unit:
- in rev/min direct spindle speed (G97);
- in m/min cutting speed (G96).

T functions

T functions allow you to specify the desired tool and to define its offset. The allowable values range from 1.0 to 9999.9999: the digits on the left of the decimal point define the tool, those on the right define the offset number.

Preparatory G functions

GOO	Rapid axis positioning
G01	Linear interpolation
G02	Gircular interpolation CW
G03	Circular interpolation CCW
GO4	Dwell at end of block
G09	Deceleration at end of block
G20	Gloses the GTL programming ambient
G21	Opens the GTL programming ambient
G27	Continuous operation with automatic speed reduction on corners
G28	Continuous operation without automatic speed reduction on corners
G29	Point to point operation
G33	Constant or variable pitch thread
G40	Offset disable on profile
G41	Offset enable on profile (tool left to part)
G42	Offset enable on profile (tool right to part)
G70	Inch programming
G71	Metric programming
G72	Point measure with radius offset
G74	Delta of a theoretical point without radius offset

G79	Programming of coordinates referred to machine zero (only valid within the block)
G80	Fixed cycle disable
G81	Drilling cycle
G82	Spot-facing cycle
G83	Deep drilling cycle
G84	Tapping cycle
G90	Absolute programming
G91	Incremental programming
G93	Feedrate expressed as the inverse of the item execution time.
G94	Feedrate expressed in mm/min or in/min
G95	Feedrate in mm/rev
G96	Spindle speed in m/min
G97	Spindle speed in rev/min
G98	Enable adaptative control
G99	Disable adaptative control

Auxiliary M functions

MOO	Program stop
M01	Optional program stop
M02	End of program
M03	Spindle rotation CW
M04	Spindle rotation CCW
M05	Spindle stop
M06	Tool change
M07	Secondary coolant on
M08	Primary coolant on
M09	Coolant off
M10	Axes lock
M11	Axes unlock
M12	Rotary axes lock
M13	Spindle rotation CW and coolant on
M14	Spindle rotation CCW and coolant on
M19	Spindle stop with angular orient.
M30	End of program and return to the first block
M41 M42 M43 M44	Selects range 1-2-3-4
M40	Selected range disable
M45	Automatic range selection
M60	Part change

Auxiliary H user Functions	These functions allow you to expand the set of auxiliary functions provided by the ISO Standard (MOO...M99).
Fixed cycles	With preparatory functions G81-G89 you can define a fixed cycle, i.e. program a series of operations (drilling, tapping, boring, etc.) without repeating the parameters of the required operation for each hole individually.
Return speed variation during tapping	Within a tapping cycle you can modify speed by programming or entering an RMS code and specifying the variation percentage via keyboard. Example: RMS=110 +10% of the programmed F RMS= 10 -90% of the programmed F
Dwelling time	You can program dwelling time by entering the TMR code followed by the desired value (in sec). Example: TMR=2
Machining time	TIM allows you to program as many as six machining times (in sec). Each value is given an index. TOT allows you to program seven partial machining times for specific cycle points.
Program messages	DIS allows you to display programmed messages, variables and numerical constants on the communication area of screen 1. Examples: (DIS,"TOOL=12") (DIS,E37) (DIS,UOV)
Scale factor	SCF allows you to program a scale factor applicable to a single axis or to the whole set of axes. Examples: (SCF,2) scale factor 2 for all axes (SGF,2,X) scale factor 2 for the X axis

Threading

G33 defines a cylindrical or conical thread movement, with constant or variable pitch.
The specified parameters characterise the type of thread.
The allowable formats are:

G33 Z..K..	cylindrical thread with constant pitch
G33 Z..X..K..	conical thread with constant pitch
G33 X..K..	front thread with constant pitch
G33 Z..K..I+..	thread with increasing variable pitch
G33 Z..K..I-..	thread with decreasing variable pitch

where:

G33	preparatory function
X,Z	coordinates of the final point
K	thread pitch
I+	pitch variation

Cutter radius vectorial offset

The control can automatically compensate the tool radius.
Select G41 to enable tool radius compensation for a tool on the left of the profile.
Select G40 to enable compensation for a tool on the right of the profile.
After having enabled a radius compensation, the axis starts moving perpendicular to the interpolation plane.
Select G40 to disable compensation.

Definition of stock allowance

By programming or entering UOV you can specify the desired stock allowance in contouring operations.
This allows you to temporarily vary the programmed offset.
Example:
UOV=1.5

To disable the stock allowance program or enter UOV=O

Mirror machining

MIR allows you to program mirror machining for all the coordinates axes.
Example:
....
(MIR,X)
....
(MIR,Y)
....
(MIR,X,Y)

Rotation in the plane

URT allows you tm rotate the whole or part of a program in the plane.
The centre of rotation is the current origin.
Example:
(URT,45)

Program repetition

RPT allows you to repeat n times a program or a subroutine, in order to obtain special cycles.
The maximum number of repetitions is 99.
It is possible to have up to three levels of repetition.
The program portion to be repeated must be closed by ERP.
Example:
(RPT,99)
....
....
(ERP)

Parametric Programming

E codes are used to parametrically program the geometrical and technological data of a machining cycle.
Programmed parameters can be used in mathematical and trigonometric operations as well as for expression calculations.
The maximum number of E parameters must be defined during characterization.
The indexes assigned to E parameters vary according to the format of the variable. They are shown in Table 1.1.

Table 1.1. - E parameters

Format	Parameters	Min/max value
BY (byte)	EO..E9	0 to 255
IN (integer)	E10..E19	-32768 to +32768
LI (longinteger)	E20..E24	-2.147.483.647 to +2.147.483.647
ME (real)	E25..E29	±7 whole or decimal digits
LR (longreal)	E30..En	±16 whole or decimal digits

n depends on characterization

- Arithmetic operators:
  - \+ (addition)
  - \- (subtraction)
  - * (multiplication)
  - / (division)

- Functions:

SIN(A)	sine of A
COS(A)	cosine of A
TAN(A)	tangent of A
ARS(A)	arcsine of A
ARC(A)	arccosine of A
ART(A)	arctangent of A
SQR(A)	square root of A
ABS(A)	absolute value of A
INT(A)	integer portion of A
NEG(A)	inverts the sign of A
LOG(A)	common logarithm for version K14
MOD(A,B)	calculates the remainder of the A,B quotient
FEL(A,B)	calculates the item having a B index (1,2,3) from the geometric item (straight line) having an A index
FEP(A,B)	calculates the item having a B index (1,2) from the geometric item (point) having an A index
FEC(A,B)	calculates the item having a B index (1,2,3) from the geometric item (circle) having an A index

The indexes (either A or A,B) can be E parameters or numerical values.

- Calculation of an expression:
The control calculates expressions containing constants, parameters or functions.
Example:

```
N1  E37=E31*SIN(E30)+123.4567/SQR(16)
```

Assignment blocks for calculation variables:

```
"LAB1" E51  = -0.00000124 + 5
/E35 = FEL(37,1)
E7 = 81
E10 = 1
E25 = E25 + 30
```

You can use and display E parameters within both programs or subroutines.

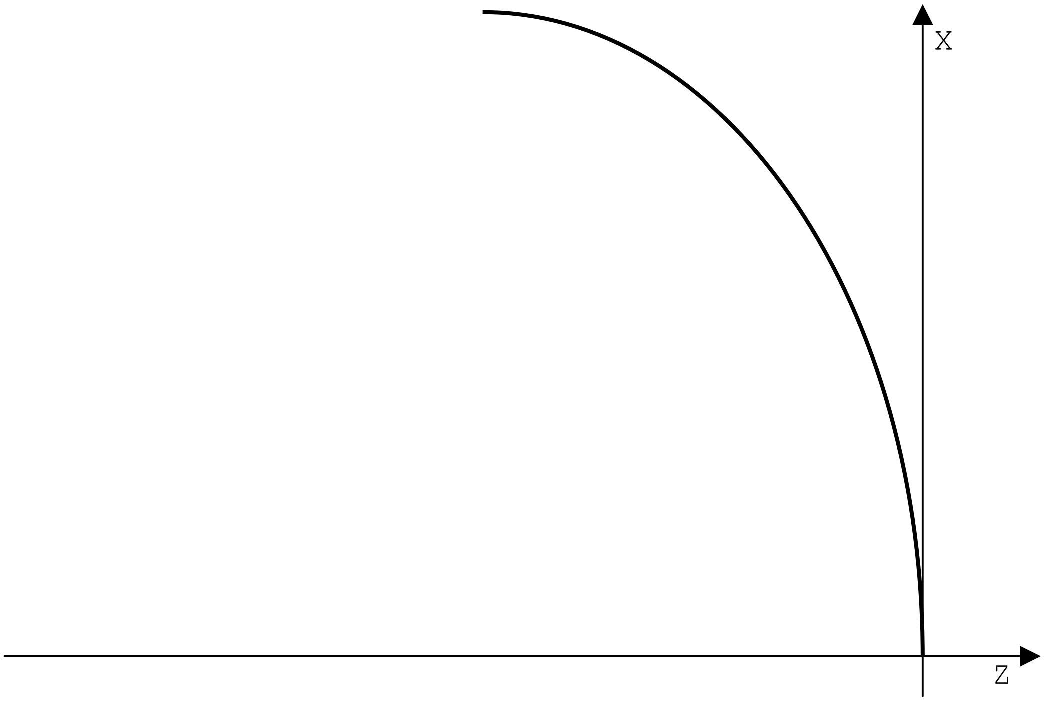
Parametric Subroutines

A subroutine is a sequence of blocks defining a customised machining cycle that can be called from a main program.
A subroutine can only call another subroutine (two nesting levels).
Subroutines are stored in the user memory. Therefore, the allowable number depends both on the length of the subroutines and on the size of the user memory.
To call a subroutine enter CLS.
Example:

```
N35  (CLS,PROG1)
```

The geometric and technological data (G, F, S, X, Z, Y, origins, etc.) characterising the working cycle can be replaced by parameters, the values of which must have been defined in the main program, before calling the subroutine.

Fig. 1.1. - How to use parametric subroutines for parabolic programming



The parameters are:
E31 = focal distance (twice the focus)
E32 = increment in Z
E33 = starting Z
E34 = final Z

<u>Main Program</u>

```
N.. T1.O1  M6 S..F..
N.. G X Z5
E31=52
E32=2
E33=0
E34=-168.8
(CLS,PARAB)
G Z  ...
.............
.............
```

<u>Subroutine "PARAB"</u>

```
G1  G42 X Z E33
"START"E33=E33-E32
(BLT,E33,E34,END)
E35=2*SQR(2*E31*ABS(E33))
XE35 ZE33
(BNC,START)
"END"E35=2*SQR(E31*ABS(E34))
G40 XE35 ZE34
E35=E35+10
G XE35
```

Program branches

To establish a branch within a program, enter any instruction containing the name of the label to branch to.
A label is an alphanumeric sequence of max. 6 characters delimited by " " (double quotation marks). The label must always be programmed before the block number (and after the / in the case of a slashed block).
Example:
/"START"N125

Branches can be conditional or unconditional. Branching codes are listed in Table 1.2.

Table 1.2. - Branching codes

Code	Meaning
(BNG,LABEL)	Unconditional branch to the LABEL
(BGT,VAR1,VAR2,LABEL)	Branch if VAR1 is greater than VAR2
(BLT,VAR1,VAR2,LABEL)	Branch if VAR1 is less than VAR2
(BEQ,VAR1,VAR2,LABEL)	Branch if VAR1 is equal to VAR2
(BNE,VAR1,VAR2,LABEL)	Branch if VAR1 is different from VAR2
(BGE,VAR1,VAR2,LABEL)	Branch if VAR1 is greater than or equal to VAR2
(BLE,VAR1,VAR2,LABEL)	Branch if VAR1 is less than or equal to VAR2

VAR1 and VAR2 are the variables to be compared. They can be parameters, machine logic signals, or numeric values.

Examples:

N1 (BGT,E1,123,END)	branch to END if the value of variable E1 is greater than 123
N2 (BEQ,SA3,1,LAB)	branch to LAB if the Boolean variable SA3 is ON
(BNE,E1,E5,START)	branch to START if the value of variable E1 is different from E5N40
(BEQ,SYVAR1,CH,OK,END)	branch to END if characters from SYVAR1 are OK

Partial program Execution

EPP allows you to only execute a part of a program enclosed between two labels.

Example:

```
"START"
..........
..........
"END"
(EPP,START,END)
..........
```

At the end of the partial execution, the program is resumed from the block following EPP.

Probing cycles

With G72-G74, you can program 2 probing cycles:

- G72 measures the coordinates of a point on the space with a linear movement and stores them in the sequence of E variables defined in the current block. This function includes probe radius compensation.

Example:
G72 X200 E32
the value calculated for X is stored in E32 and E33

- G74 measures the delta between nominal points and the coordinates established by a fixed probe.
The results are stored in the sequence of E variables defined in the block.

Example:
G74 X50 E40
(max. 3 axes in the block)
the delta between the theoretical and the measured point is stored in E40.

You can use these values to program origin and tool requalifications and to check tool wear.

Origin Requalification

RQO allows you to requalify the origin by using the E variables resulting from probing cycles G72-G73. For example:
(RQO,0,XE35)
E35 = difference between theoretical and measured coordinates

Tool Requalification

RQU allows you to program tool requalification.
The requalification value is usually stored in the E parameters obtained after probing cycles.
The format is:
(RQU,Ntool,Noffset,ZEn,KEm)
where:
Z is the axis associated to the length Offset;
K is the tool diameter.
The tool number (Ntool) must be defined according to the tool life routine, since the offset to be modified may be the one of the alternative tool.
If the offset table includes the offset value, RQU both updates the value and declares the tool out of use when the maximum limit is exceeded.

Program RQP to only update the length and diameter offsets, leaving the applied offset values unchanged.

Tool wear

Tool condition can be monitored by programming a probing cycle with G74.
If the resulting value exceeds the current tolerance, the tool is declared

out of use with code TOF.
Example:
(TOF,12)   tool 12 broken

Communication between program and machine logic

Data communication between application software and interface logic is made possible by sending I/O parameters from the program to the interface logic (SK variables) or to the machine (SA variables).

The system data structures are:
– the A buffer, which defines all the ON/OFF electrical signals connecting the C.U. to the installation;
– the K buffer, which defines all the communication variables between application software and interface logic to the machine tool.

Assignment examples:

SA12=SK	give to bit No.12 of the A buffer the value of the first bit of the K buffer
SK5=SK7	give to byte No.5 of the K buffer the value of byte No.7
SA128=1	set the signal (bit) No.128 of the A buffer to ON
SK7="RIF"	write statement MIF starting from byte No.7 of the K buffer
SA3.BY=255	give value 255 to byte No.3 of the A buffer

Software travel Limits

Before moving the tool, the control checks whether the programmed movements will exceed the machine operating field both on linear and circular paths. In case of error. a message is displayed. The operating field limits are stored in

the system characterization files and can be temporarily modified within a program by means of the DLO code.
In manual movements, the error message appears while the operating field limit is being exceeded.

Operating field Limits

To change the field limits from the program, enter a DLO instruction.
The allowable format is:
(DLO,X- X+)
(DL0,Z- Z+)
where:
X+ upper limit for X;
X- lower limit for X;
Z+ upper limit for Z;
Z- lower limit for Z.

The programmed limits refer to the current origin, i.e.:
N20 (DLO,X-50 X100)
N21 (DLO,Y-60 Y20)

Protected areas

With DSA you can create up to three protected areas, into which the control will not permit the axes to move.
The allowable format is:
(DSA,n,X- X+,Z- Z+)
where:
n = number of the protected area (1 to 3);
X- = lower limit for X;
X+ = upper limit for X;
Z- = lower limit for Z;
Z+ = upper limit for Z.

Protected areas are enabled before the movement starts.
To cancel a protected area, program ASC.
The allowable format is:
(ASC,n)
where:
n = number of the protected area.

Example:
```
(DSA,1,X0 X50,Z5 Z100)
(DSA,2,X-100 X-50, Z-20 Z150)
(ASC,1)
(ASC,2)
........
........
(DSC,1)
........
```

High level Geometric programming (GTL)

GTL allows you to create any planar profile using only the information on the drawing. The control automatically calculates the intersection and tangency points and the radius between the geometric elements of the profile.
The basic GTL elements are:
- reference origins;
- points;
- straight lines;
- circles.

To define a GTL element, you must declare its space coordinates and the direction of motion. The maximum number of elements must be specified during characterization.
The index of an element can be either a numeric value or an E parameter.
To define an implicit profile, i.e. a sequence of geometric elements, you must first store the data defining the constituents.
A profile can be open or closed. An open profile starts from one point and ends at a different point. A closed profile starts and ends at the same point.
It is possible to move any non-contouring axis to any point on the profile.
The allowable formats and definitions are shown in Table 1.3.

Table 1.3. - GTL elements

Element	**Definition**	**Description**
Reference origins	on=ZXa	
Points	pn=[om] Z X	point with cartesian coordinates
	pn=[om] m a	point with polar coordinates
	pn=±lm,±lp	intersection of two straight lines
	pn=±lm,±cp[,s2]	intersection line-circle
	pn=±cm,±lp[,s2]	intersection circle-line
	pn=±cm,±cp[,s2]	intersection of two circles
Straight lines	ln=[om]I J r,[op]I J	line tangent to two circles
	ln=[om]Z X,[op]Z X	line through two points
	ln=[om]I J r,[op]Z X	line tangent to a circle and through a point
	ln=[om]Z X,[op]I J r	line through a point and tangent to a circle
	ln=[om]I J r,a	line tangent tm a circle and forming an angle with the abscissa axis
	ln=[om]X Y,a	line through a point and forming an angle with the abscissa axis
	ln=±cm,±cp	line tangent to two circles
	ln=±cp,pm	line tangent to a circle and through a point
	ln=pm,±cp	line through a point and tangent to a circle
	ln=pm,pq	line through two points
	ln=±cm,a	line tangent tm a circle and forming an angle with the abscissa axis
	ln=pm,a	line through a point and forming an angle
	ln=ln,d	line parallel to a straight line at a d distance

Table 1.3. continued

Element	Definition	Description
Circles	cn=[om]I J r	circle with cartesian coordinates
	cn=[om]m a r	circle with polar coordinates
	cn=±lm,±lp,r	circle of given radius tangent to two lines
	cn=±lm,±cp,r	circle tangent to one line and one circle of given radius
	cn=±cp,±lm,r	circle of given radius tangent to one circle and one line
	cn=pm,±lp,r	circle of given radius through a point and tangent to a line
	cn=±lp,pm,r	circle of given radius tangent to a line and through a point
	cn=±cm,±cp,r	circle of given radius tangent to two circles
	cn=pm,±cp,r	circle of given radius through a point, tangent to a circle
	cn=±cp,pm,r	circle of given radius tangent to a circle passing through a point
	cn=pm,pq,r	circle of given radius connecting two points
	cn=pm,±lp	circle with centre at a point and tangent to a line
	cn=pm,pa,pr	circle through three points
	cn=pm,r	circle of given radius centered at a point
	cn=±cm,+d	concentric circles at a given distance
	cn=pm,±cp[,s2]	circle with center at a point and tangent to a circle

Virtual axes

To execute profiles on the plane or on cylinders with a rotary axis and a linear axis we have introduced the notion of virtual axes.

The available modalities are:

- modality 1: in profiles executed on the plane, allows you to convert cartesian coordinates into polar coordinates. The linear axis is perpendicular to the rotary axis.

- modality 2: in profiles executed on a cylinder, allows you to convert cartesian coordinates into cylindrical coordinates. The linear axis is parallel to the rotary axis.

When one of these modalities is activated, the rotary axis is positioned on 0.
You can program the profile with ISO or GTL language, depending on the axes (real or virtual) that define the cartesian plane.

Modality 1

With this modality, you can convert cartesian coordinates to polar coordinates.

The allowable format is:
(UAV,1,linear axis real axis rotary real axis, virtual abscissa axis virtual ordinate axis, minimum radius)

i.e., (UAV,1,ZC,UV,r)

where:

Z	real linear axis;
C	real rotary axis;
U	virtual abscissa axis;
V	virtual ordinate axis;
r	minimum radius.

The minimum radius establishes the inhibited area.

When calculating the minimum radius, you must take into consideration the programmed feedrate, so that the speed of the rotary axis does not exceed the rapid. To calculate the minimum radius, use this formula:

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

where:

r	minimum radius;
F	feedrate in mm/min;
Vc max	rapid of the rotary axis.

Modality 2

With this modality, you can convert cartesian coordinates to cylindrical coordinates. The profile is generated on a cartesian plane formed by a virtual rotary axis and a linear axis.

To program a profile, use the following format:

(UAV,2,real rotary axis,virtual axis, radius)

i.e., (UAV,2,C,V,r)

where:

c   real rotary axis;
V   virtual axis;
r   radius of the cylinder.

ASSET

The ASSET utility allows you to access from program the standard peripherals connected to your NC-110:
- storage devices with data structures organized as tables (CMOS, bubble memory, hard disk, floppy disk);
- keyboard;
- user screen.

It also allows you to handle communication through a serial line.

Accessing Storage devices

With CMOS, bubble memory, hard disk and floppy disk you can store data in tables. Each table constitutes a file and is made up of fixed-length records. Records can undergo reading, writing and editing operations.

The allowable instructions are:
OPN - open a file;
DER - define a record;
RED - read a record;
WRT - write in a record;
CLO - close a file;
CRE - create a file;
CAN - cancel a file.

Accessing the Keyboard

With ASSET, you can enable the keyboard from program in order to:
- enter data to be stored in variables;
- enter parameters to be displayed on the user screen.

The allowable instruction is:
INP - enable data entry from keyboard.

Accessing the User Screen

Each process can have a dedicated user screen (20 lines x 64 columns). Within each line, you can define a certain number of fields, each one of which can contain either alphanumeric strings (comments) or numeric strings (variable contents).

To handle the user screen from program, use the following instructions:
SCR - enable/disable user screen;
DEF - define fields in the user screen;
OUT - display fields in the user screen.

Managing I/O errors

With certain ASSET instructions, you can manage I/O errors either automatically or from part program.
To select the desired modality, you must set the ERR parameter:
ERR = 0 selects the automatic modality;
ERR = 1 selects error management from part program.

Managing a serial Line

Data stored in a record structure can be transmitted to or read from a peripheral device by means of a serial line RS 232 or a current loop.
The allowable communication protocols are: Hardware, Hardware Data-Set Ready non Standard or XON/X0FF.
To handle this line, use the following instructions:
SOP - initialise the serial line and enable it to transmit/receive data;
DER - define record files to be used for transmission/reception;
PUT - transmit a data record;
GET - receive a data record;
SCL - release the serial line.

Defining a Profile

By using the DFP code. you can store as many as 8 profiles (from 1 to 8) to be used in roughing and/or finishing cycles.

To program these profiles, use either the ISO or the GTL standard.

Roughing Cycles

To execute this type of cycle, you only need to recall a previously stored profile and then specify the number of cuts and the desired stock allowance.

The available roughing cycles are:
SPA - cuts parallel to axes X or Z without prefinishing;
SPP - cuts parallel to the profile;
SPF - cuts along the axes Z or X and final copy of the resulting profile.

Finishing Cycle

The CLO code allows you to recall a stored profile and then execute it leaving a stock allowance.

Threading cycle

This cycle allows you to program a threading cycle in a single block. The allowable format is:

(FIL,Z,X,K,L,R,T,P,a,b)

where:

Z final Z dimension
X final X dimension
K pitch
L number of roughing/finishing passes
R return distance between tool and piece
T three-figure code defining the type of thread:
  1st. figure:
  0= thread with final groove
  1= thread without final groove
  2nd. figure:
  0= external thread
  1= internal thread
  3rd. figure:
  0= metric thread
  1= witworth thread
  2= non standard thread

P number of principles
a angle of the non-standard thread
b depth of the non-standard thread

Grooving Cycle

The TGL code allows you to program a grooving cycle in a single block. The allowable format is:

(TGL,Z..,X..,K..)

where:

Z groove's final point
X inner groove diameter
K tool width

Table 1.4. - Mnemonic codes used in programs

Code	Description
CLS	Call subroutine
BNC	Unconditional branch
BGT	Branch if >
BLT	Branch if <
BEQ	Branch if =
BNE	Branch if < >
BGE	Branch if ≥
BLE	Branch if ≤
EPP	Execution of part of a program
RPT	Repetition of part of a program
ERP	Close a repetiton level
UAO	Origin selection
UOT	Temporary origin definition
UIO	Incremental origin definition
MIR	Mirror
URT	Rotate machining plane around current origin
SCF	Scale factor
RQO	Reapply origin
RQU	Reapply
RQP	tool
DPI	Define interpolation plane
DLO	Define operating limits
DIS	Display variable
TOF	Tool Off
UCG	Define graphic scale
CLG	Reset graphic display
DCG	Graphic field disable
DSA	Define protected area
ASC	Protected area enable
DSC	Protected area disable
DPT	Define probing parameters
DLY	Define delay interval
UAV	Enable virtual axes
DFP	Define profile
SPA	Paraxial roughing without prefinishing
SPF	Paraxial roughing with prefinishing
SPP	Roughing parallel to the profile
CLP	Profile finishing cycle
FIL	Threading cycle
TGL	Grooving cycle
DTL	Define position tolerance

Table 1.4. continued

Code	Description
ASSET CODES	
OPN	Open a file
DER	Define a record
RED	Read a record
WRT	Write in a record
CLO	Close a file
CRE	Create a file
CAN	Cancel a file
INP	Enable data entry from keyboard
SCR	Enable/disable the user screen
DEF	Define fields on the user screen
OUT	Display fields on the user screen
SOP	Initialise serial line
PUT	Transmit a data record on a serial line
GET	Receive a data record
SCL	Release serial line

Table 1.5. - Assignment codes entered by program or keyboard

Code	Description
E	Definition of the numerical variable
o	Definition of the origin
p	Definition of the point
l	Definition of the straight line
c	Definition of the circle
TMR	Dwell time at end of block in G04
UOV	Definition of stock allowance
JOG	Definition of item length in JOG status
RTA	Probe reapplying value for abscissa axis
RTO	Probe reapplying value for ordinate axis
ERF	Form error (in mm)
MCD	Max deviation value of directing cosines in dynamic mode
USB	Slashed blocks enable/disable
UVR	Rapid enable/disable
USO	Optional stop enable/disable
URL	Feedrate enable/disable
UCV	Real time dimensions display (display 1)
RAP	Automatic return to profile enable/disable - automatic limit switch search
UAS	Axes connection enable/disable
RMS	Definition of the return speed variation percentage in tapping cycle
UEP	Position error enable/disable
SA	Machine logic A buffer
SK	Machine logic K buffer
SYVAR	Area for variables available to the program
TIM	Enable system timer
TOT	Enable system counter
SSL	Define the limit spindle speed

Table 1.6. - Mnemonic codes entered from the keyboard for machine tool control

Code	Description
ORA	Define an origin
CAO	Delete an origin
VOA	Display an origin
URP	Define a workpiece rotation angle
VTU	Load tool life parameters
CTU	Delete a tool from the tool file
UCG	Define graphic scale
CLG	Enable graphic display
DCG	Disable graphic display
CAC	Delete tool offset / the whole offset file
SPG	Program selection
REL	Release a selected program
DPT	Probe parameters definition
RCM	Enable memorised search
ERM	Disable memorised search
PTM	Timer setting
VIC	Timer display
ESE	End of program block definition
DIS	Format variable display
EVA	Evaluate and display the expression value
VOL	Enable/Disable electronic handwheel
UCA	Incremental offset modification

Table 1.7. - Mnemonic codes for program management

Code	Description
EDI	Editing functions: program storage or modification
RIM	Modify a block
INS	Insert a block
CAN	Delete a block
DEL	Delete a program
COP	Copy a program from memory to peripheral or vice versa
REN	Rename a program
DIR	Directory of stored programs
INI	Magnetic cassette or user memory initialisation
CRE	Create a file with fixed length
FOR	Create file of offsets, tool life, origins, with fixed length, and formatted fields

Table 1.8. - Codes identifying memories or peripherals

Code	Description
MP	Permanent memory for program storage
PR	Tape reader
CT	Magnetic cassette
PP	Tape punch
LP	Printer
TY	Teletype
HD	Hard Disk
FD	Floppy Disk

SIPROM

SIPROM (System Integrated Programming) enables you to program the control/machine tool interface.

With a SIPROM module installed in the control, you can compile, edit, debug and implement the logic required by your machine.

When you have finished debugging, the machine logic can be punched onto tape, from which you can program EPROM's to be installed on the memory modules that contain the system software.

The transfer onto punched tape can be bypassed by connecting the NC-110 to a compatible EPROM programmer.

This programming technique allows you to easily customise, modify and update the interface, thus increasing the reliability of your system.

2. OPERATOR PANEL

Chapter Objectives

This chapter gives you an overview of the operator panel and a console. The operator panel constitutes the man-machine interface.

2.1. The features of the control panel

The operator panel (refer to fig. 2.1.) includes:
- a display;
- a functional keyboard;
- an alphanumerical keyboard;
- a key POWER ON/OFF.

In additional to the operator panel CNC may to have a console (refer to fig. 2.2.) including:

- 10 pushbuttons, 10 leds indication, 3 overrides, programmed from SOFTWARE:
• pushbuttons of operational mode selectors;
• pushbutton **CYCLE START**;
• pushbutton **HOLD**;
• potentiometer for feedrate;
• potentiometer for manual control of JOG direction and speed;
• potentiometer for spindle speed;
- 38 freely programmable from program of logic pushbuttons with LEDS indication;
• handweel;
• pushbutton **EMERGENCY STOP**.

Fig. 2.1. – THE OPERATOR PANEL (CRT)

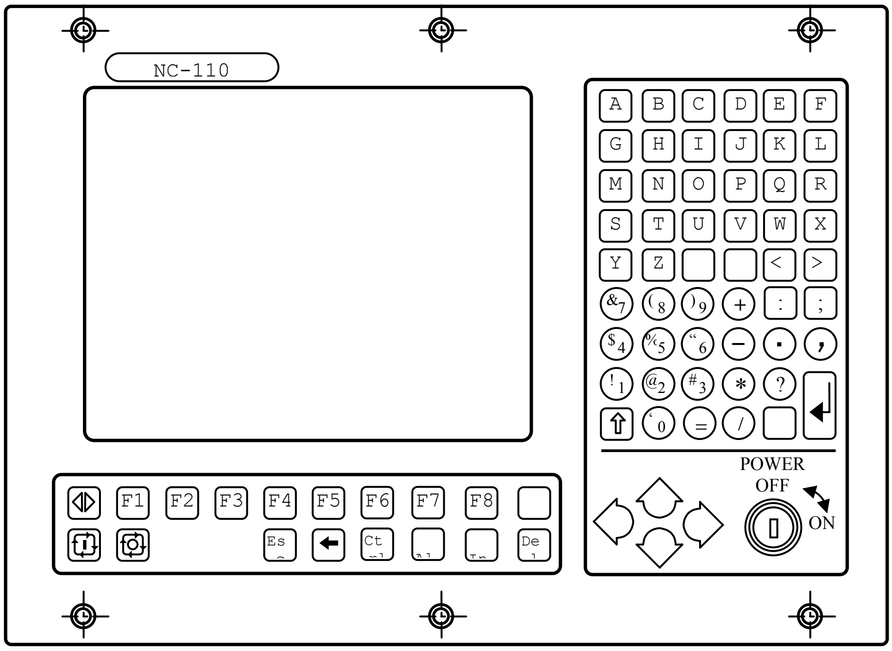


Fig. 2.2. – THE CONSOLE PANEL WITH CRT

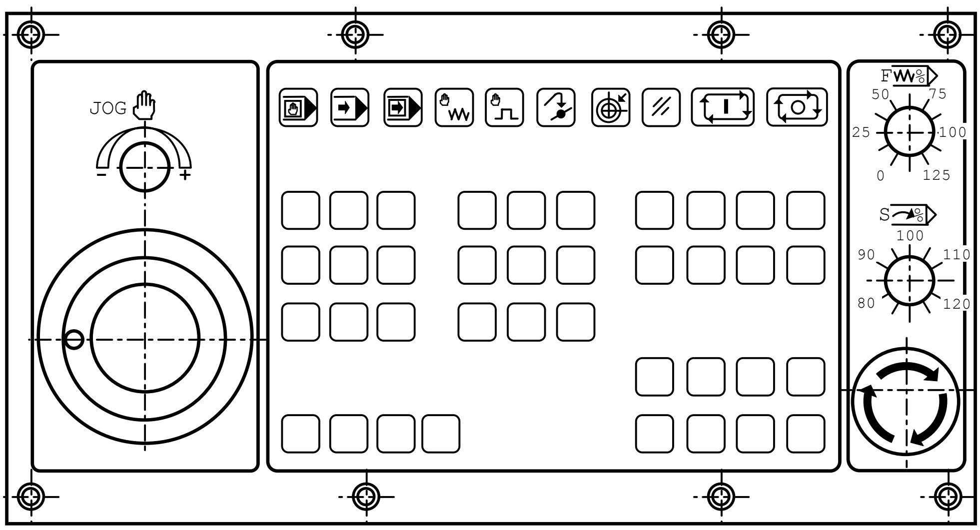


Fig. 2.3. – THE OPERATOR PANEL FOR NC-110 (TFT)

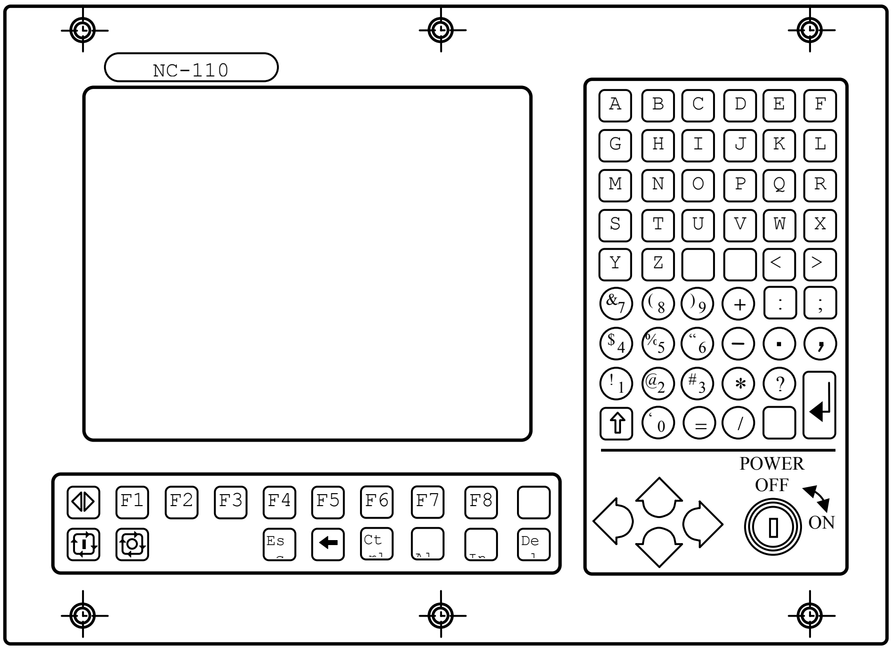
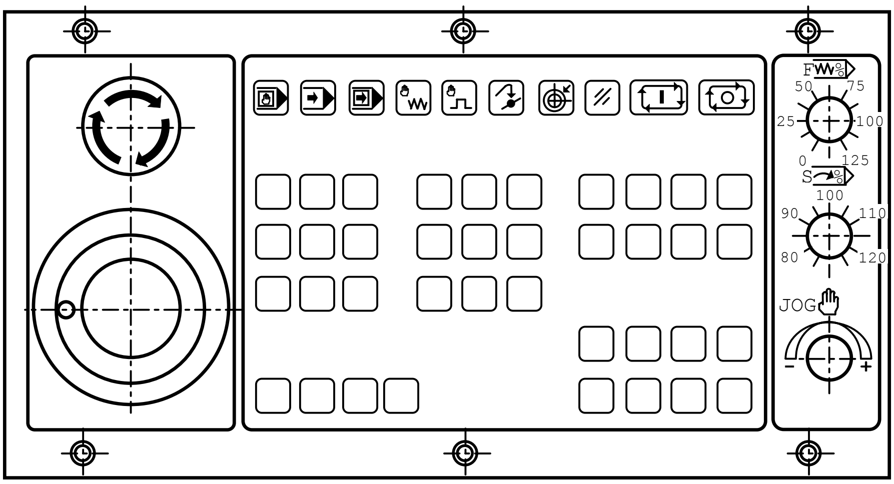


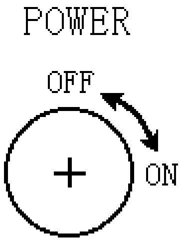
Fig. 2.4. – THE CONSOLE PANEL FOR NC-110 (WITH TFT)



2.2. Functional purpose of keyboard

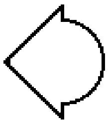
2.2.1. Functional purpose of operator panel pushbuttons

The functions of SOFT KEYs **F1÷F8** are presented in the describes of their using.



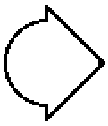
KEY – POWER ON/OFF

Use for power switching ON/OFF.



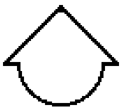
BACK SPACE

Moves the cursor to the left.



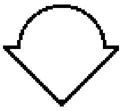
FORWARD SPACE

Moves the cursor to the right.



LINE BACK

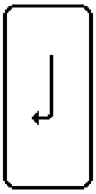
- Displays the previous data block when control is in the **STEP** or controlled search modes;
- allows the backward search to the previous data block in **EDIT** mode;
- allows the select axes in manually moves;
- in the **COMMAND** mode allows the backward search any command from 9 last entered commands for repeating of input it by pressing **ENTER;**
- in the **MACHINE TOOL CONTROL** mode with simultaneously pressing **ALT** it allows the backward search:
  1) any from 16 last entered commands for repeating of input it (by following pressing **ENTER**) in all modes besides MDI mode;
  2) any from 16 last entered data blocks for repeating of execution it by means of following pressing **CYCLE START** only in the MDI mode.



LINE FORWARD

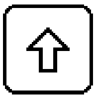
Displays the next data block when control is in the **STEP** or controlled search modes;

- allows the select axes in manually moves;
- in the **COMMAND** mode allows the forward search any from 9 last entered commands for repeating input it by following pressing **ENTER**;
- in the **MACHINE TOOL CONTROL** mode with simultaneously pressing **ALT** it allows the forward search:
  1) any from 16 last entered commands for repeating of input it (by following pressing **ENTER**) in all modes besides MDI mode;
  2) any from 16 last entered data blocks for repeating of execution it by following pressing **CYCLE START** only in the MDI mode.



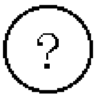
ENTER

Initiates the execution of command or executes the input of data information. The data input is enabled in any modes (MDI, AUTO, STEP, MANJ, MANU, RESE, HOME, PROF) and during execution of part program or separate block.

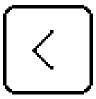


SHIFT

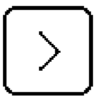
The pressing it allows you to install register of the keyboard in the direction back to current temporary.



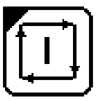
For CNC - «**?**»
For CPU – pushbutton **TAB**



For CNC - «**<**»
For CPU – pushbutton **F11**

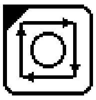


For CNC - «**>**»
For CPU - pushbutton **F12**



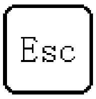
CYCLE START (in the versions with a console is not used)

Initiates a part program execution in the AUTO and STEP modes and axes motion in the MDI, MANJ, MANU, PROF, HOME modes.
Executes reset of system when control is in the RESE mode.
In the devices with console this pushbutton does not used.



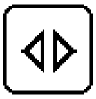
HOLD (in the versions with a console is not used)

Establishes a stop after a controlled deceleration. To resume The cycle, press **HOLD** and then **CYCLE START**. This pushbutton Does not function during a threading cycle. In the devices with Console it does not used.



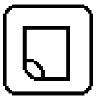
ESCAPE

Use for clear a screen. During execution it takes the cursor to the first or last block if after it mode the backward search or forward search by pressing **LINE BACK** or **LINE FORWARD**.



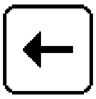
TOGGLE

Toggles the control between COMMAND screen and either MACHINE TOOL CONTROL. When you use CNC as CPU, it is used as the pushbutton **F1**.



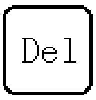
MORE MENU

When you use CNC as CPU, it is used as the pushbutton **F10**.



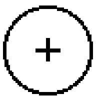
CANCEL

Delets the last symbol displayed on the screen in the string of input and editing of command, data block or in the editing session.



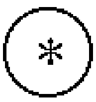
DELETE

Cancels the string of input and editing and error messages.



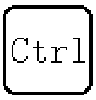
For CNC - «**+**»

For CPU – pushbutton **PgUp**. It is used for modifications of the parameters in the SETUP (DOS) mode.



For CNC - «*****»

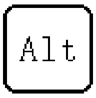
For CPU – pushbutton **PgDn**. It is used for modifications of the parameters in the SETUP (DOS) mode.



Ctrl

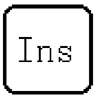
In the CNC it is used for:

1) translation from
2) reinitialization of CNC SOFTWARE by simultaneously pressing **Ctrl+Alt+Del**;
3) switching off display of operator panel by simultaneously pressing **Ctrl+Alt+any alphanumeric pushbutton.** For resume the the display press any alphanumeric pushbutton.



Alt

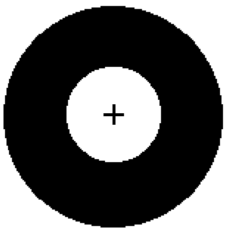
Used with the pushbutton **Ctrl** (refer to describe **Ctrl**), pushbuttons **LINE BACK** and **LINE FORWARD** (refer to describtion this pushbuttons).



Ins

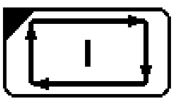
In CNC it does not used.

2.2.2. The pushbuttons of console



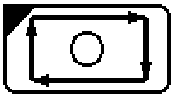
EMERGENCY STOP (red pushbutton)

Removes power from machine tool. For secondary power switching on (after emergency switching off) turn the pushbutton to direction indicated in it (to a click).



CYCLE START

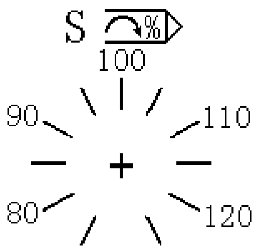
Initia tesapart program execution in the AUTO and STEP modes or separate Block execution in the MDI mode, and axes motion in the MANJ, MANU, PROF and HOME modes. Executes reset of system when control is in the RESE mode.



HOLD

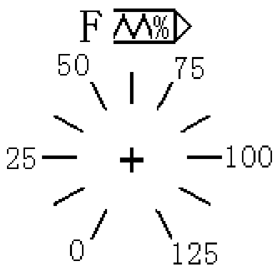
Establishes a stop after a controlled deceleration. To resume the cycle, press HOLD and the CYCLE START. This pushbutton does not function during a threading cycle.

2.2.3. Switches of a console



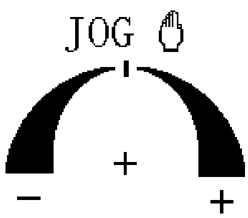
SPEED OVERRIDE - potentiometer

Modifies the programmed spindle speed. The step of modification can be installed in the software characterization. It does not operate during a threading cycle.



FEEDRATE OVERRIDE - potentiometer

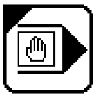
Modifies the programmed feedrate the step of modification can be installed in the software characterization. It does not operate during a threading cycle.



JOG - potentiometer

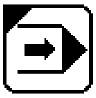
Selects the speed and direction of axes motion during joggig. When URL=1, also controls the rapid speed (G00). The left half of the selector range stands for 0% variation. The right half of selector determinates 0 to 100% variation. The step of variation can be installed in software characterization.

2.3. The selectors of modes



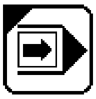
MDI – input via keyboard

Enables manual data input. To execution it press **CYCLE START**.



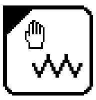
AUTO – automatic program execution

When you press **CYCLE START**, automatic program execution (block by block) is initiated.



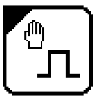
STEP – semiauto execution

When you press **CYCLE START**, a program block is executed.



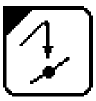
MANU – continuous jog

When you press **CYCLE START**, the axes selected via keyboard (by pressing **LINE BACK** or **LINE FORWARD**) moves at the rate and in the direction selected with **JOG**. To stop the axis, release it.



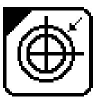
MANJ – incremental jog

When you press **CYCLE START**, the selected axis moves by an Increment equal to the value assigned to **JOG** via keyboard (for example: JOG=50), at the rate and in the direction defined by the **JOG** potentiometer.



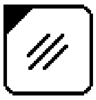
PROF - return

Allows you return the specified axis after **JOG**. RAP=0 selects a manual return (axis is selected via keyboard). RAP=1 selects an automatic return (axis by axis). The rate and direction is selected with **JOG**. When you press **CYCLE START** the motion is begined.



HOME

Allows you to take each axis to absolute zero microswitch. The axis for motion is selected via keyboard by pressing **LINE BACK** or **LINE FORWARD**. When you press **CYCLE START** the motion is begined.



RESE - reset

When you press **CYCLE START**, the data in dynamic buffer is removed. The selected program is reinitialized. Toll offsets and origins are not cancelled. The current functions M, S, T are reseted and is selected origin number 0 for all axes.

2.4. Information of modes on the screen

The functional key allows you to toggle between the available display modes:

- SAVE screen mode;
- COMMAND mode;
- MACHINE TOOL CONTROL mode with 2 alphanumeric (videopages #1, #7) and 1 graphic (videopage #6) page.

2.5. SAVE SCREEN mode

This mode allows you to save the screen, it is installed by simultaneously pressing **Ctrl+Alt+any alphanumeric pushbutton.** To resume the display press any alphanumeric pushbutton.

P.S. During the SAVE SCREEN mode the functional pushbuttons **F1÷F8** are actived.

2.6. The videopage of COMMAND mode

This mode allows you to display and edit files. For example: files of the programs, directories, origins, tool offsets, tool life tables. In this mode the following picture appears on the screen (Fig. 2.5.). In high corner of the screen is displayed COMMAND, and in low part-menu. Operator can use menu or execute data input via keyboard. In figure 5 is presented the example of command execution DIR.

```
DIR/MP3    {File name}/{memory}

↓- contine,   ENTER  -  exit


DIR/{memory}
   NAME     SECT        NREC      F      ATT
File name N sectors  N records   Flag  Attribut
File name N sectors  N records   Flag  Attribut
.............................
.............................

File name N sectors  N records   Flag  Attribut
File name N sectors  N records   Flag  Attribut
File name N sectors  N records   Flag  Attribut
1       2       3       4       5       6       7       8
1FILE   [ALL]
```

Fig. 2.5.

P.S. 1 sector = 128 bytes.

2.7. The videopages of MACHINE TOOL CONTROL mode

The videopages of MACHINE TOOL CONTROL mode are displayed during the machine tool control and shows the status of process. Are 2 types of videopages: alphanumeric and graphic pages. The type of videopage select by pushbutton F2. The alphanumeric information is displayed in pages #1 and #7, the graphic - in page #6.

The videopage #1 displays:

- axis coordinates, functions S, M, T, origins, tool offsets, indexed axes, point-to-point axes;
- repeat cycles, program and subroutines (if they are selected);
- current block of program;
- text of part program with «running» string;
- messages;
- active commands, system status, machine tool status etc.

2.7.1. Functional purpose of functional keyboard in MACHINE TOOL CONTROL mode

F1

Process

Use to select of process (in software versions allowable a process number – from 1 to 5). When you use CNC as CPU, it is used as the pushbutton **F2**.

F2

Videopage

Use for select of videopage (#1, #6, #7). When you use CNC as CPU, it is as the pushbutton **F3**.

F3

Tool length calculation

Declares new values for offsets. When you press it, on the screen in input-editing string is appeared a symbol, after that you can insert the information, described in §6.11. This pushbutton allow the control to directly calculate tool length or diameter (radius). When you use CNC as CPU, it is used as the pushbutton **F4**.

F4

Tool compensation

Allows you to enter or display the tool length or diameter compensation values for the selected offset axes: Z or K – for mill version SW, X, Z, R, O – for lathe SW. When you press it, in the input-editing string is appeared a symbol, after that you can input the information described in §6.5. When you use CNC as CPU, it is used as the pushbutton **F5**.

F5

Send to logic

When entering a symbol string addressed to PLC, it allows to send if to logic. For further information refer to the "Interface PLC". When you use CNC as CPU, it is used as the pushbutton **F6**.

F6

Input – editing of block

It allows you to take a program block to the input-editing string in order to clear or edit it and for subsequent execution in the MDI mode. When you use CNC as CPU, it is used as the pushbutton **F7**.

F7 For CNC does not used. For CPU - pushbutton **F8**.

F8 For CNC does not used. For CPU - pushbutton **F9**.

2.7.2. Videopage #1

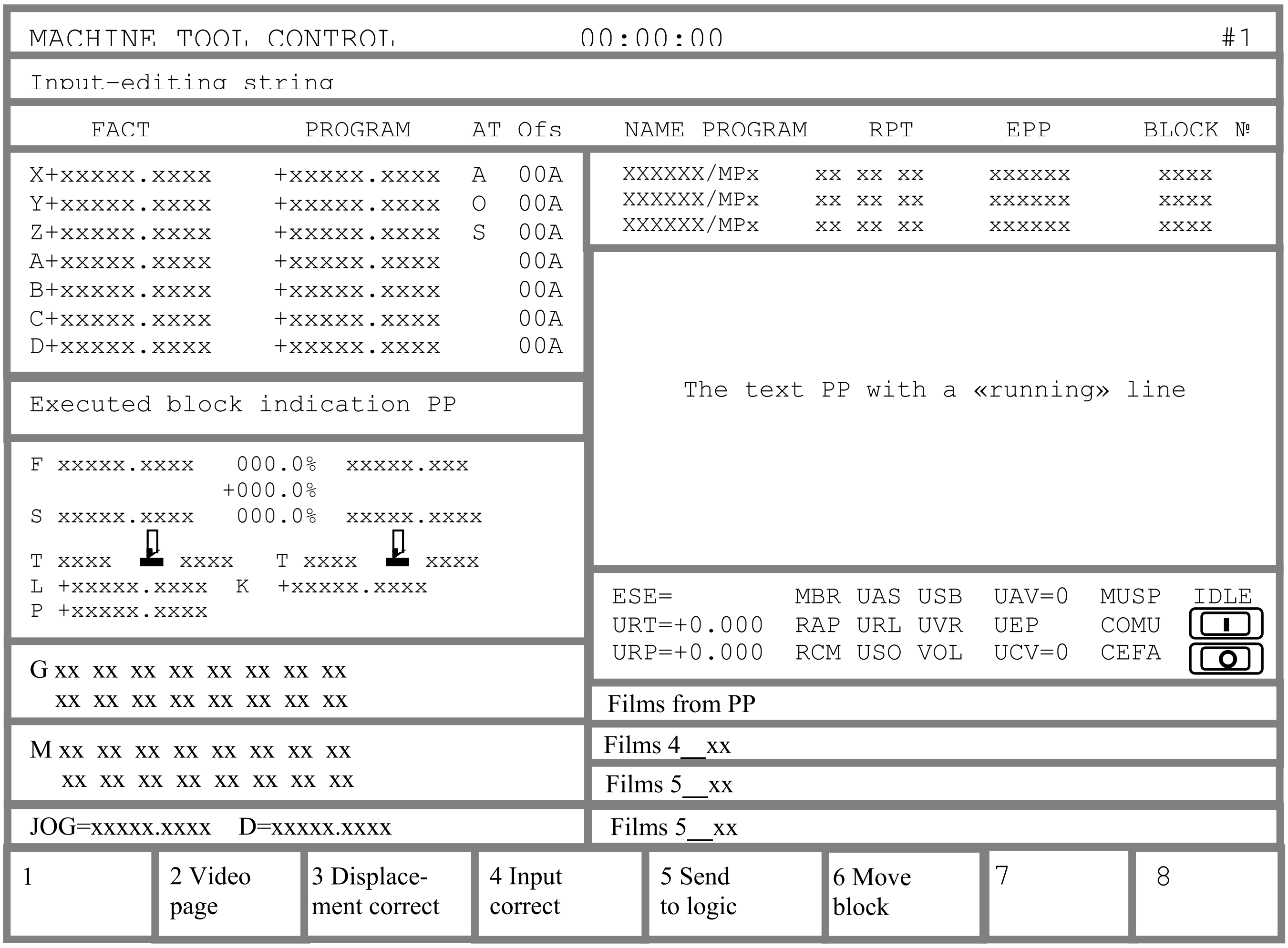


Fig. 2.6.

In the page is displayed the information of process.

Select the desired axes coordinates by setting the UCV parameters:

To display:	**Set UCV to:**
calculated coordinates	0
transducer coordinates	1
following errors	2

Transducer coordinates and following errors may be of use to the Service, whereas calculated coordinates are normally used by the operator.

The displayed codes are shown in Table 2.1.

Table 2.1. - Codes in Process Screen 1

CODE	**MEANING**
MACHINE TOOL CONTROL 1	process screen
#n	number of the screen
IDLE	system at a standstill
RUN	system in execution
HOLD	system in hold
RUNH	move and functions allowable in hold
RESE	reset
ERRO	error
LEDS	START indicator ON
LEDH	HOLD indicator ON
MODE	displays the selected mode
MDI	execution from keyboard
AUTO	automatic mode
STEP	block-by-block mode
MANU	manual mode
MANJ	manual jog
PROF	return to profile
HOME	search for home
COMU	acknowledge axes move
ESE	number of the last block to be executed
URT	rotation angle of the working plane
URP	rotation angle of the workpiece
UCV	axis coordinates 0:calculated coordinates 1:measured coordinates 2:following error
UAV	0: disable virtual axes 1: enable virtual mode 1 2: enable virtual mode 2

Table 2.1. continued

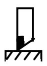
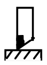
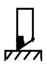
CODE	MEANING
System variables	(in reverse mode if activated)
UAS	0: axis connected 1: axis disconnected
UVR	0: program feedrate 1: rapid
MBR	0: enable multiblock retrace 1: disable multiblock retrace
USB	0: skip slashed block 1: execute slashed block
URL	0: rapid not controlled with manual selector 1: rapid controlled with manual selector
RAP	0: jog back to profile 1: auto back to profile
USO	0: enable M01 1: disable M01
VOL	0: disable handwheel 1: enable handwheel
RCM	memorised search (in reverse when activated)
System status	(in reverse mode when activated)
CEFA	wait for acknowledge to emit auxiliary functions
MUSP	wait signal from machine tool
COMU	wait for acknowledge to axis motion
X----	axis name, nominal and current coordinates
At	type of axis
o	ordinate
a	abcissa
m	spindle
mm	mirror activated
ofs	axis origin activated
xxa	number of absolute origin
xxt	number of temporary origin
xxi	number of incremental axis
F	feedrate in mm/min (values of programmed and current feedrate, percentage variance)
S	spindle speed in rev/min (values of programmed and current spindle speed and percentage variance)
T	tool on spindle
	tool offset
T	subsequent tool
	subsequent tool offset
 l	value of tool length compensation (Z)
 d	value of tool diameter compensation (K)

Table 2.1. continued

CODE	MEANING
a,b,c	indexed axes
G	G function
M	auxiliary M functions
JOG	Jog rate
D	remaining distance in jog
data set	name of selected program, program in execution or subroutine (2 levels)
RPT	repeat number and level
EPP	EPP block
N	number of block in execution

2.7.3. Videopage #6

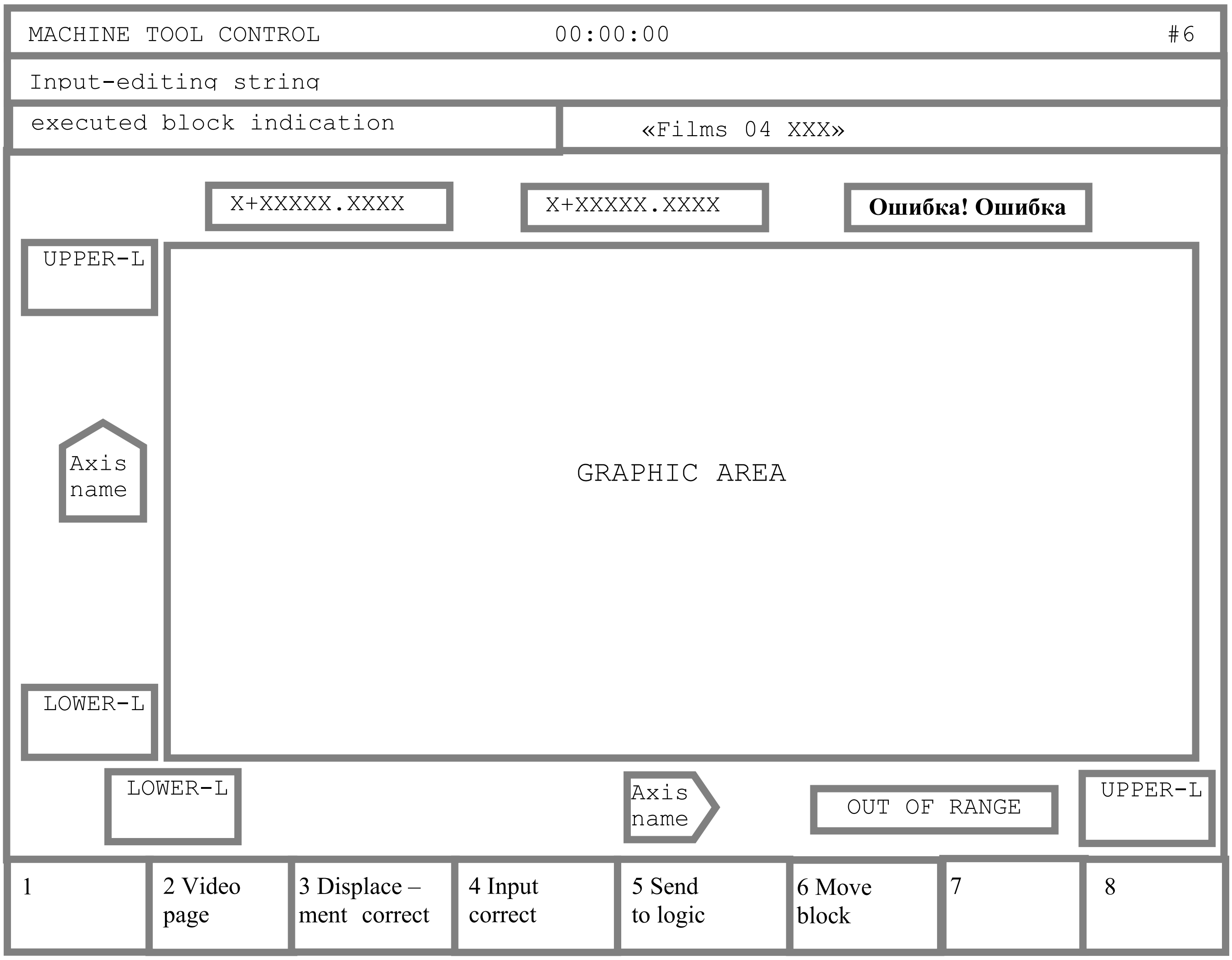


Fig. 2.7.

3. BASIC SYSTEM OPERATION

3.1. Power up

1. Turn the main switch in the electrical cabinet to the ON position.
2. Turn the **EMERGENCY STOP** pushbutton until you hear a click (direction of turning is indicated on the pushbutton).
3. Turn the key **POWER** to the **ON** position. After that the control is supplied with power and starts the autodiagnostics. If no errors are detected, videopage #1 appears on the screen and control is ready to the connection the power to the machine tool auxiliaries.

ATTENTION! AFTER THE TURNING OF THE KEY POWER TO THE ON POSITION, THE ALLOWABLE TIME-OUT OR LOADING CNC SOFTWARE IS 9 SEC.

3.2. Power OFF

In the turning of POWER key to OFF position is diactivated the machine tool and CNC is switched off. In that the stored information, including axes positions are deleted, but the programs, tool offsets and absolute origins remain stored in memory.

When you press the pushbutton EMERGENCY STOP you switch off only the supplied power of the machine tool.

3.3. Reinitialization SOFTWARE

When the machine tool is off (signal MUSPE is displayed by yellow color) you can reinitialize the SOFTWARE of CNC by simultaneously pressing **Ctrl+Alt+Del**.

3.4. CNC in the MACHINE TOOL CONTROL mode

The CNC control in the MACHINE TOOL CONTROL mode allows you to use the menu or to enter the commands via keyboard.

The main menu of the MACHINE TOOL CONTROL mode is follow:

1 DISP |2 MODIF |3 PLC |4 COP

To select of the menu option is enough to press the relative functional pushbutton Fn. The purpose of main menu options is follow:

- DISP - request to display the information about the files and memory (pushbutton **F1**);
- MODIF - request to modify the file/files or their attributs (pushbutton **F2**);
- PLC - request to start PLC for the compilation and/or debug of the machine tool logic program (pushbutton **F3**);
- COP - request to copy the files (pushbutton **F4**).

The options DISP, MODIF and COP can to execute with the relative submenu select or inputting of the command via keyboard. To return to the previous menu press the pushbutton ESC. If you don't want to use menu, refer to description of commands and order their entering via keyboard are given in text begining from §3.5.

P.S. 9 LAST ENTERED COMMANDS ARE STORED IN BUFFER. IF YOU PRESS THE PUSHBUTTON **LINE BACK** OR **LINE FORWARD**, CAN CALL THIS COMMANDS FOR REPEATED EXECUTION.

3.4.1. The orders of acts in DISP option

If you press in the main menu the pushbutton **F1** on the screen appears following submenu:

1 DIR |2 VLT |3 DIF

where:

DIR – displays all files of memory MPx;

VLT – displays the working files written in 2 section of FCRSYS file;

DIF – compares the files.

When you select DIR option, on the screen appears the submenu that requests the memory name:

1 MP1|2 MP2|3 MP3|4 MP4|5 MP5|6 MP6|7 MP0.

After the select of memory the control requests:

1 1 FILE |2 [ALL]

where: 1 FILE – the check of file existing with indicated name;
[ALL] – displays all files that are stored in the selected memory.

When you select the DIR option, on the screen appears the submenu that requests name of memory:

1 MP1|2 MP2|3 MP3|4 MP4|5 MP5|6 MP6|7 MP0.

After the select of memory, on the screen displays the list of all files that are stored in the selected memory with the cursor on one from their. Move the cursor to the file name that you want to compare. By pressing **LINE BACK** or **LINE FORWARD** pushbuttons. Own select finish by pressing **ENTER.** The second file for comparison select as first file. If the files are differented, on the screen appears the information about it.

3.4.2. The order of acts in the MOD option

If you press in the main menu the pushbutton **F2** on the screen appears following submenu:

1 EDI |2 FOR |3 ATT |4 DEL |5 REN

where:

EDI - editing of files;

FOR - creating of formatted files with the names that are writted in FIL instruction of PGCFIL file;

ATT - definition of the protection type for file/files;

DEL - deleting of file/files;

REN - renaming of file.

When you select the EDI option, on the screen at first appears the submenu that requests name of memory:

1 MP1|2 MP2|3 MP3|4 MP4|5 MP5|6 MP6|7 MP0

After the select of memory on the screen displays the list of all files selected memory with the cursor on one from their. Move the cursor to the file name that you want to edit by pressing **LINE BACK** or **LINE FORWARD** pushbuttons. Own select finish by pressing **ENTER.** After execution of that acts the control is ready for editing session. The acts of CNC in the EDI session are described in §3.4. Exit from EDI session with the writing edited file execute by pressing **F7.**

When you select the FOR option, on the screen appears the submenu that requests the file type:

1 ORIGIN |2 CORRECT |3 GETOOL

where:
ORIGIN - creating of origins file;

CORRECT - creating of tool offsets file;

GETOOL - creating of tool life file.

If the control requests the creating of file that already exist, a system request: “DELETE OLD FILE?” (Delete already existing file?). Pressing **ENTER** you answer – YES, pressing **ESC** – NO. After the select of file type, on the screen appears record: «ENTER NUMBER OF STRING» (enter the number of strings in creating file). Own input of value finish by pressing **ENTER.**

When you select the ATT option, on the screen appears submenu that requests the protection type of file:

1 PROTECT |2 DISPROT |3 HIDDEN

when:

PROTECT – disables the editing of file/files in memory MPx;

DISPROT – disables the protection of file/files in memory MPx, That were defined by PROTECT and/or HIDDEN options;

HIDDEN - disables the visualisation of file name when you will execute a command DIR.

After the select of protection type, on the screen appears the submenu that requests the memory name for protection. After the select of memory the control requests:

1 FILE |2 [ALL]

where:

1 FILE – indicates the protection of one file;
[ALL] - indicates the protection of all files stored in selected memory.

If you want to execute the protection of one file, on the screen displays the list of all files selected memory with cursor on one from their. Move the cursor to the file name that you want to protect and then press **ENTER.**
When you select the DEL option, on the screen appears the submenu that requests memory name:

1 MP1|2 MP2|3 MP3|4 MP4|5 MP5 |6 MP6 |7 MP0)

After the select of memory the control requests:

1 1 FILE|2 [ALL]

where:

1 FILE – deleting of one file;

[ALL] – deleting of all files stored in selected memory.

To deleting of one file execute the order of acts like described in the ATT option.

When you select the REN option, on the screen appears the submenu that requests memory name:

1 MP1|2 MP2|3 MP3|4 MP4|5 MP5 |6 MP6 |7 MP0

After the select of memory, on the screen displays the list of all files selected memory with cursor on one from their. Move the cursor to the file name that you want to rename and press **ENTER.** After that the control requests to enter a new name for selected file: « ENTER FILE NAME ». Entering of name finish by pressing **ENTER.**

3.4.3. The order of acts in the COP mode

If you press in the main menu the pushbutton **F4,** on the screen appears the submenu that requests name of memory:

1 MP1 |2 MP2 |3 MP3 |4 MP4 |5 MP5 |6 MP6 |7 MP0 |8TY

After the select of memory on the screen displays the list of all files selected memory with the cursor on one from their. Pressing **LINE BACK** or **LINE FORWARD** move the cursor to the file name that you want to copy. Then press the pushbutton **ENTER.** After that the control requests to enter the memory name for copying. After the select of memory the control requests to enter the file name for copying:

ENTER [FILE NAME]

The input of file name finish by pressing **ENTER.**
The functional pushbutton **F8** (8 TY) allows you to perform input/output operatations from/into a teletype using serial line RS232.

NOTE. THE PARAMETERS INDICATED IN SQUARE BRACKETS ARE OPTIONAL.

3.5. Hand-operated input of the programs and their editing

3.5.1. Input programs

In order to edit or execute a program you must store it in the control's memory MPx. The memory MPx can be situated on any peripheral device that have a file structure MS-DOS (FLASH, FDD, ZIP, LS-120, PC). Input and storing programs can be done using the keyboard or a peripheral device in the COMMAND mode by following means:

- Input the command: EDI, NAME/MPx,
  where: NAME - file name;
  MPx - selected memory name;
- The copying files from one memory into another is performed with COP command.

The input programs can perform during machining. Input-output programs using serial line RS232 is performed with the driver of teletype (TY) with 9600 boud rate. If peripheral divice is PC, it must to have a program COMNCRUS.EXE. Before connection of peripheral device to serial line RS232 it is necessary to switch off the CNC. A peripheral device and CNC must to have the earth in common point.

3.5.2. Storing programs using the keyboard

To input programs via keyboard is used the specifical editor. If you want to edit the program you must to remember:

1. Program or subprogram selected in current moment with a command SPG is disabled for editing. The control disables the execution of command SPG for a program selected in current moment with a command EDI;
2. In this case when you do not input the memory name MPx (x=0-6) the control will use the memory defined as default – it is always MP1;
3. When file is opened, on the first string of screen is displayed:
   FREE SECTORS: XXXXX
   Where: XXXXX – number of memory sectors that are still free.

ATTENTION! IN INPUT PROGRAM CHECK A NUMBER OF FREE MEMORY SECTORS. IF IT EQUAL TO ZERO IT IS NECESSARY TO EXIT FROM THE EDITING SESSION BY PRESSING **F7**. BCAUSE ALL INFORMATION THAT HAVE BEEN ENTERED IN EDITING CAN BE LOST.

TO CONTINUATION INPUT AND/OR EDITING OF THIS FILE IT IS NECESSARY TO FREE A SAME SPACE IN MEMORY DELETING THE FILES THAT YOU ISN'T NEEDED IN CURRENT TIME WITH COMMAND DEL, AFTER THAT YOU CAN EXECUTE A COMMAND EDI IN SECOND TIME. IN EXIT FROM THE EDITING SESSION BY PRESSING **F7** EDITTABLE FILE WILL CLOSED WITH NAME THAT WAS DECLARED IN OPENING IT.

Main menu of editor is following:

```
1 BEGIN |2 END    |3 FIND |4 FIND |5 DEL |6 GO TO   |7 SAVE  |8 QUIT
OF LINE | OF LINE |  UP   |  DOWN | LINE |  LINE  N | & EXIT |NO SAVE
```

Where:

BEGIN OF LINE – setting of cursor to begin of the string (pushbutton **F1**);

END OF LINE – setting of cursor to end of the string (pushbutton **F2**);

FIND UP – search of symbol(s) in direction to the previously strings (back) of the file (pushbutton **F3**);

FIND DOWN – search of symbol(s) in direction to the next strings (FORWARD) of file (pushbutton **F4**);

DEL LINE – deleting of the string (pushbutton **F5**);

GO TO LINE N – go to string n (pushbutton **F6**);

SAVE & EXIT – exit from EDIT session with the writing of editing file (pushbutton **F7**);

QUIT NO SAVE – exit from EDIT session without the writing of editing file (pushbutton **F8**).

4) There is submenu that is selected by pressing the pushbutton **MORE MENU**:

```
1 MARK  |2 COPY |3 MOVE |4 DELETE |5 UNMARK |6 GO TO   |7 SAVE  |8 QUIT
 BLOCK  | BLOCK | BLOCK |  BLOCK  |         |  LINE  N | & EXIT |NO SAVE
```

Where:

MARK BLOCK – marking of beginning and end of the block program.

COPY BLOCK – copying of the marking block to the subsequently string of cursor;

MOVE BLOCK – moving of the block to subsequently string of cursor;

DELETE BLOCK – deleting of the block from a editable program;

UNMARK – unmarking of the block.

Besides of menu options a editor can execute following functions:

- moving of a string program in direction forward and back by pressing **LINE FORWARD** and **LINE BACK**;
- moving of a cursor in the string in direction forward and back by pressing **FORWARD SPACE** and **BACK SPACE**;
- moving of program pages on the screen in direction forward and back by pressing **Alt+LINE FORWARD** and **Alt+LINE BACK**;
- deleting of the string that is displayed in the input area on the screen by pressing **DEL**;
- deleting of a symbol that is situated before a cursor on the screen by pressing **CANCEL**;
- insert of the string in a program by pressing **ENTER**.

To input a program in a memory of CNC it is necessary to go in the COMMAND mode and then enter a command:

EDI, PROG1/MPx. After that press a pushbutton **ENTER**.

Where: x – identification number of the memory MPx (available value – 0÷6).

The program name can be 6 alphanumeric characters long. The first character must be a letter. After input of a command on the screen appears the window of input and editing program PROG1. If the program already exists in memory, it appears on the screen to be modified. If the program is not in memory, the control creates in memory MPx new program.

To enter and subsequently writing of the program block it is necessary to press the pushbutton **ENTER** after entering every block.

Example:

N1 T1.01 M6 S200 M3 M7

N2 G X100 Y50 Z100

and etc.

To exit from the EDIT session it is necessary to press one from the pushbuttons **F7** or **F8**. On the screen appears a record: COMMAND, NAME/MPx. Pressing the pushbutton **TOGGLE** can turn to the previously mode.

3.5.3. Modifiging programs

To enter to the EDIT session execute the following procedure:

1) press the pushbutton **TOGGLE** for set the COMMAND mode:
2) enter via the keyboard a command:

EDI,PROG1/MPx and press **ENTER**

The allowable options of editing are:

- modifying blocks (OVR);
- inserting blocks (INS);
- cancelling blocks.

3.5.4. Modifying blocks in a stored program

To modify a block in stored program use the following procedure:

1) enter a command:

EDI,PROG1 and press **ENTER;**

2) use the **LINE FORWARD** or **LINE BACK** pushbuttons to position a cursor on the block that you want to modify. The block with the cursor will be displayed by yellow color and simultaneously will be shown in the string of input and editing;

3) to remove a character use the pushbutton **CANCEL.** Using **FORWARD SPACE** or **BACK SPACE** pushbuttons move the cursor in block to the position before deleting and/or inserting symbols. The operation editing you must to finish by pressing **ENTER;**

4) to inserting a string to program position cursor on the string after that you want to insert a new string and press **ENTER.** After that input a new string and press **ENTER;**

5) to change a string position a cursor on the string that you want to change. After that clear the string of input and editing by pressing **DEL.** Input a new string and press **ENTER.**

3.5.5. Searching for characters in the program

In the control is in the EDIT session, it is possible to search inside the selected program for a sequence of characters (maximum 32). To that execute the following procedure:

1) clear the string of input and editing;
2) input the sequence of characters and press the pushbutton **F3** or **F4** to search forward or back through the program.

If the control does not find the character sequence, the cursor positions to the beginning of the program (BOF) for search reverse or to the end of the program (EOF) for search forward.

3.5.6. DIR – listing the directory of stored programs

Enter via the keyboard a command:

DIR,MPx and press **ENTER.**

This command displays all programs that are stored in the CNC memory MPx. Each program is shown with its name and length in memory sectors (1 sector = 128 bytes).

When all programs stored in the memory MPx are shown, on the last line of the page appease the message:

FREE SECTORS:xxxxx

That indicates the number of memory sectors that are still free. To display the next page press the **LINE FORWARD.** Finish the execution of command by pressing **ENTER** twice. After that the control is waiting for new instructions. The pucture of videopage in the COMMAND mode in execution of command DIR is presented on fig. 3.1.

```
DIR/MP3

↓ - Continue, ENTER - Exit

DIR/MP3
NAME     SECT   NREC   F  ATTR
AXCFIL    6
C        15
FCR       3
G         6
IOCFIL    3
```

Fig. 3.1.

3.5.7. COP - copying a program

To copying program enter via a keyboard a command:

COP, P9/MP2, P2/MP3 and press **ENTER**

where:

P9 – a program name that you want to copy from memory MP2;
P2 – new program name that you want to copy to memory MP3.

With pressing **ENTER** the control copies program P9 from memory MP2 to memory MP3 and stores it with name P2.

When you work with memory MP1, you must remember that it is default memory. In this case you can enter a command COP without memory name.

Example: COP P9, /MP2. After execution of this command, program P9 will be copied from memory MP1 to memory MP2 with its name.

The command: COP, PROG allows you to display program PROG page by page. To display the subsequent page press the **LINE FORWARD** pushbutton. Whe the last page appears on the screen the message: «File displayed» will appear. After pressing **ENTER** the control is waiting for new instructions.

3.5.8. REN – renaming a program

Enter a command:

REN, P99, PR1 and press **ENTER**

This command renames program P99 with name PR1.

3.5.9. DEL – delete a programe

In execution a command:

DEL, PROG1 with pressing a pushbutton **ENTER**

a program PROG1 deletes from memory.

4. OPERATION AND PROGRAMMING CODES

Chapter Objectives

This chapter gives you a summary of the operation and programming codes available with the NC-110.

According to their function, we have organized codes in five tables:

- Table 4.1. lists codes used for file management;
- Table 4.2. lists peripheral codes;
- Table 4.3. lists codes used for part program control;
- Table 4.4. lists codes used for machine tool control;
- Table 4.5. lists codes used in part program blocks.

Table 4.1. - CODES FOR FILE MANAGEMENT

Code	Format	Function
EDI	EDI,NAME	Edit a file or create a new file
RIM	RIM	Modify a program block
INS	INS	Insert a program block
CAN	CAN	Delete a program block
DEL	DEL,NAME	Delete a program
COP	COP,NAME,PERIPHERAL	Copy a program from memory to a peripheral device
	COP,PERIPHERAL,NAME	Copy a program from a peripheral device to memory
REN	REN,NAME,NAME1	Rename a program
DIR	DIR	Display a directory of programs
INI	INI,NAME,/CT INI,NAME,/MP1	Initialise magnetic cassette Initialise memory boards
CRE	CRE,NAME,N,M	Create a formatted file
FOR	FOR,NAME,N	Create a formatted file for tool offsets and origins
ATT	ATT,NAME,100 ATT,NAME,0	Protects a program Disable protection

Table 4.2. - CODES FOR PERIPHERAL DEVICES

Code	Type of peripheral
MP	Permanent memory for program storage
PR	Punched tape reader
CT	Magnetic cassette
PP	Tape punch
LP	Printer
TY	Teletype

Table 4.3. - CODES USED FOR PART PROGRAM CONTROL

Code	Format	Function
E	En[.Type]=value	Define a numeric variable Type: type of variable The possible types are: BY = BYTE IN = INTEGER LI = LONG INTEGER RE = REAL LR = LONG REAL
o	on=geometric expression origin	Define the reference origin
p	pn= geometric expression point	Define a point
l	ln=geometric expression line	Define a straight line
c	cn=geometric expression circle	Define a circle
TMR	TMR=value	Define the dwell time at the end of an item in G04 or in a fixed cycle; expressed in seconds (xxxxx.xxx)
UOV	UOV=value disabled by UOV=0	Define the stock allowance
JOG	JOG=value	Define the item length to be executed in JOG
RTA	RTA=value	Define the probe requalification value for the abscissa axis
RTO	RTO=value	Define the probe requalification value for the ordinate axis
ERF	ERF=value	Define the form error (in mm)

Table 4.3. continued

Code	Format	Function
MCD	MCD=value	Define the max. deviation of cosines in dynamic mode
USB	USB=1 disabled by USB=0	Execution of slashed blocks
UVR	UVR=1 disabled by UVR=0	Enable rapid for all the axes programmed with feedrate (used for program testing). Select the rapid with the feedrate potentiometer.
URL	URL=1 disabled by URL=0	Use the feedrate potentiometer to control rapid in JOG
USO	USO=1 disabled by USO=0	Enable optional stop (function M01)
UCV	UCV=n	Select the coordinate to be displayed on screen 1: UCV=0 calculated coordinate UCV=1 transducers' coordinate UCV=2 following errors
RAP	RAP=1 disabled by RAP=0	Enable automatic return to profile after a Hold Enable automatic home search during axes referencing
UAS	UAS=1 disabled by UAS=0	Axes standstill enable (test mode with axes disconnected)
RMS	RMS=value	Define a percent feedrate variation in the return phase of a tapping cycle

Table 4.3. continued

Code	Format	Function
UEP	UEP=1 disabled by UEP=0	Enable position error
SA	SAn=value	Define A-buffer in machine logic
SK	SKn=value	Define K-buffer in machine logic
SYVAR	SYVARn=value	Define a set of variables
TIM	TIMn=value	Set the system timer TIM0=0 set the clock to zero
TOT	TOTn=value	Set the time totalizer

Table 4.4. - CODES FOR MACHINE TOOL CONTROL

Code	Format	Function
ORA	ORA,n,X..,Z..	Define absolute axes origin n: origin number
CAO	CAO,n	Cancel an origin n: origin number If n is not specified, the whole file of origins will be cleared
VOA	VOA,n	Display an origin n: origin number E.g. VOA,5 displays: VOA,5,X387.8,V12.127 (the coordinates are stored in record 5 of the file of origins)
VTU	VTU,n[,T,offset,t1, t2,t3,B]	Load the tool life parameters n: tool number T: alternative tool offset: alternative tool offset t1: max. theoretical time t2: min. theoretical time t3: residual time B: tool status (utilizable) To display the record, enter: VTU,n
CTU	CTU,n	Cancel a tool from the tool life file n: tool number If n is not specified, the tool life file will be cleared
VOL	VOL=1 disabled by VOL=0	Enables electronic handwheel

Table 4.4. continued

Code	Format	Function
UCG	UCG,n,AXIS1I AXIS1S, AXIS2I AXIS2S,[AXIS3]	Create graphic screen n=1: non-coordinated axes display n=2: coordinated axes display AXIS1I: more negative X AXIS1S: more positive X AXIS2I: more negative Y AXIS2S: more positive Y AXIS3 : axis normal to working plane
CLG	CLG	Reset graphic screen
DCG	DCG	Disable graphic screen
CAC	CAC,n	Delete a tool offset n: No. of offset to be deleted If n is not specified, the whole offset file will be cleared.
SPG	SPG,program name [/dev. name]	Select program for execution
REL	REL	Release the selected program Use REL to pass from JOB to EDP display during execution
MBR	MBR=1 MBR=0	Enable retrace mode Disable retrace mode
RCM	RCM	Enable stored search
ERM	ERM	Disable stored search
DPT	DPT,Qa,Qs,Vm	Define probe parameters Qa: approach dimension (distance from nominal point) Qs: safety dimension (max. contact dimension) Vm: measuring speed (in mm/min)

Table 4.4. continued

Code	Format	Function
PTM	PTM,h:min.:[sec.]	Set the system clock
VIC	VIC,n	Display the content of a timer variable (TIMx) n: variable number the following will be displayed: VIC,variable name,hour:min.:sec.
ESE	ESE,n	Program execution up to block n E.g. SNC,24
DIS	DIS,<variable>	Display a variable variable= E parameters, assignment codes, geometric items, messages for the operator
EVA	EVA,<expression>	Calculate and display the value of an expression. E.g.: EVA,SQM(144)+20/2 Press **SEND** displays 22 EVA,(SQR(144)+20)/2 Press **SEND** displays 16

Table 4.5. - CODES USED IN PART PROGRAM BLOCKS

Code	Format	Function
CLS	(CLS,subrout. name)	Call subroutine
BNC	(BNC,LABEL)	Unconditional branch to label
BGT	(BGT,VAR1,VAR2, LABEL)	Branch if VAR1 > VAR2
BLT	(BLT,VAR1,VAR2, LABEL)	Branch if VAR1 < VAR2
BEQ	(BEQ,VAR1,VAR2, LABEL)	Branch if VAR1 = VAR2
BNE	(BNE,VAR1,VAR2, LABEL)	Branch if VRR1 <> VARZ
BGE	(BGE,VAR1,VAR2, LABEL)	Branch if VAR1 ≥ VAR2
BLE	(BLE,VAR1,VAR2, LABEL)	Branch if VAR1 ≤ VAR2
EPP	(EPP,LABEL1,LABEL2)	Execute the part of program between LABEL1 and LABEL2
RPT	(RPT,n)	Repeat a part of program n times (n≤99). The part of program is delimited by blocks RPT and ERP
ERP	(ERP)	Delimit repeat part of program
UAO	(UAO,n)	Select absolute origin n: No. of the absolute origin previously entered from keyboard
UOT	(UOT,n,X..,Z..)	Define and apply a temporary origin for the current axes n: No. of the absolute origin
UIO	(UIO,X..,Z..)	Declare the incremental origin with reference to the current absolute origin

Table 4.5. continued

Code	Format	Function
MIR	(MIR,X,Z) disabled by (MIR)	Mirror machining on the declared axes
URT	(URT,angle) disabled by (URT,0)	Rotate the machining plane with respect to the current origin
SCF	(SCF,n[,axes..]) disabled by (SCF)	Apply a scale factor to the declared axes. n: scale factor N.B. If no axes are declared, the scale factor is intended for all the axes.
RQU	(RQU,Ntool,Noffset, Z..,X..)	Modify current Z and/or X offset and update offset file Ntool: tool number Noffset: offset number
RQP	(RQP,Ntool,Noffset, Z..,X..)	Modify current Z and/or X offset without updating correction values
DPI	(DPI,axis1,axis2)	Define interpolation plane Axis1, Axis2: name of the axes defining the plane
DLO	(DLO,AXISn+ AXISn-)	Define sw overtravel limits E.g.: (DLO,X100 X-80)
UCG	UCG,n,AXIS1I AXIS1S, AXIS2I AXIS2S,[AXIS3]	Create graphic screen n=1:non-coordinated axes display n=2: coordinated axes display AXIS1I: more negative X AXIS1S: more positive X AXIS2I: more negative Y AXIS2S: more positive Y AXIS3 : axis normal to working plane

Table 4.5. continued

Code	Format	Function
CLG	(CLG)	Reset the graphic screen
DCG	(DCG)	Disable graphic screen
DSA	(DSA,n,X- X+,Y- Y+)	Define the protected zone limits n: zone number (1,2,3) X-: more negative X X+: more positive X Y-: more negative Y Y+: more positive Y
ASC	(ASC,n)	Enable protected zone n: zone number
DSC	(DSC,n)	Disable protected zone n: zone number
DPT	(DPT,Qa,Qs,Vm)	Define probing parameters. Qa: approach dimension Qs: safety dimension Vm: measuring speed
UAV	(UAV,1,xc,pv,r) (UAV,2,b,w,r) (UAV,0)	Enable virtual axes p and v r: minimum radius Enable virtual axes b and w r: radius of the cylinder Disabl: virtual axes
DFP	(DFP,n)	Specify the profile number (1-8) to be recalled during roughing and finishing cycles
EPF	(EPF)	Close the profile definition
SPA	(SPA,a,n,l,x,z)	Roughing cycle parallel to the a axis a: x or z axis n: profile number l: pass number x: stock allowance on x
WAI	(WAI,N)	Wait the signal from SND function from N-process
SND	(SND,N)	Send signal to WAI function from N-process
EXE	(EXE,N,program name [/dev.name])	Select and execut program in N-process

Table 4.5. continued

Code	Format	Function
		z: stock allowance on z
SPF	(SPF,a,n,l,x,z)	Roughing cycle parallel to the a axis with prefinishing a: x or z axis n: profile number l: pass number x: stock allowance on x z: stock allowance on z SPF cannot be applied to non-monotonous profiles
SPP	(SPP,n,l,z1,z2,X1,X2)	Roughing cycle parallel to the profile n: profile number l: pass number Z1:stock allowance on z Z2:initial stock allowance on z X1:stock allowance on x X2:initial stock allowance on x
CLP	(CLP,n)	Call for a finishing cycle n: profile number
TGL	(TGL,z..,X..,K..) (TGL,z..,K..,K..)	Grooving cycle parallel to the X or Z axis Z: end groove dimension X: internal groove diameter K: tool width
FIL	(FIL,Z..,X..,K.., L..,R..,T..,P.., a..,b..)	Threading cycle [2]: end 2 dimension [X]: end X dimension K : pitch L : pass number [R]: tool detachment [T]: 3-letter code defining the type of threading [P]: number of [a]: thread angle [b]: thread depth

Table 4.5. continued

Code	Format	Function
OPN	(OPN,nchan,nfile[/dev][,acc])	Open a file in a logic channel nchan: number of channel nfile: file name dev: device name acc: access mode (R=reading, W=Writing)
DER	(DER,nst,nvar[,nvar]..)	Define variables in a record nst: name of record structure nvar: variable name
RED	(RED,nchan,nst[,rec])	Read a record rec: record number
WRT	(WRT,nchan,nst[,rec])	Write/edit a record
CRE	(CRE,nst,nfile[,dev],nrec)	Create a file nrec: number of records
CLO	(CLO[,nchan])	Close a file
CAN	(CAN,nfile[,/dev])	Cancel a file
INP	(INP,[comm],nvar[,field])	Enter data from keyboard comm: comment string
SCR	(SCR[,ON]) (SCR, OFF)	Enable /disable user screen
DEF	(DEF,field,line,[col], init,value [,format][,R])	Define fields col: column number init.value: default value R: reverse mode
OUT	(OUT[,field[,value]]...)	Display a field value: value to be displayed

Table 4.5. continued

Code	Format	Function
SOP	(SOP,n conn,M index ,B code,Pcode, L length,S number [,H index] [,A/B] [,E][,N pad number]	Initialize the serial line n conn: number of connector 1= OS 5000-5001 2= OS 5630 M index: communication modality 1= transmission only 2= reception only 3= transmis./reception B code: baud rate 0= 110 baud 1= 300 baud 3= 600 baud 4= 2400 baud 5= 4800 baud 6= 9600 baud P code: parity 0= no parity 1= even parity 2= odd parity L length: datum length 5= 5 bit 6= 6 bit 7= 7 bit 8= 8 bit S number: defines the stop bit x= 1 Stop bit 1= 1,5 stop bit 2= 2 stop bit H index:exchange modality 0= hardware handshake 1= hardware handshake with DSR 2= XON XOFF A= ASGII characters U= binary characters E= enable character echo N pad number: number of pad cha-racters after LF
PUT	(PUT,n,structure)	transmits data from a record structure already defined by DER

Table 4.5. continued

Code	Format	Function
		n.structure: number of record structure
GET	(GET,n,structure)	receives data from a serial line and stores them on a record n.structure: number of record structure
SCL	SCL	release the serial line

5. PROGRAMMING

Chapter Objectives

This chapter shows you the programming functions available with your NC-110.

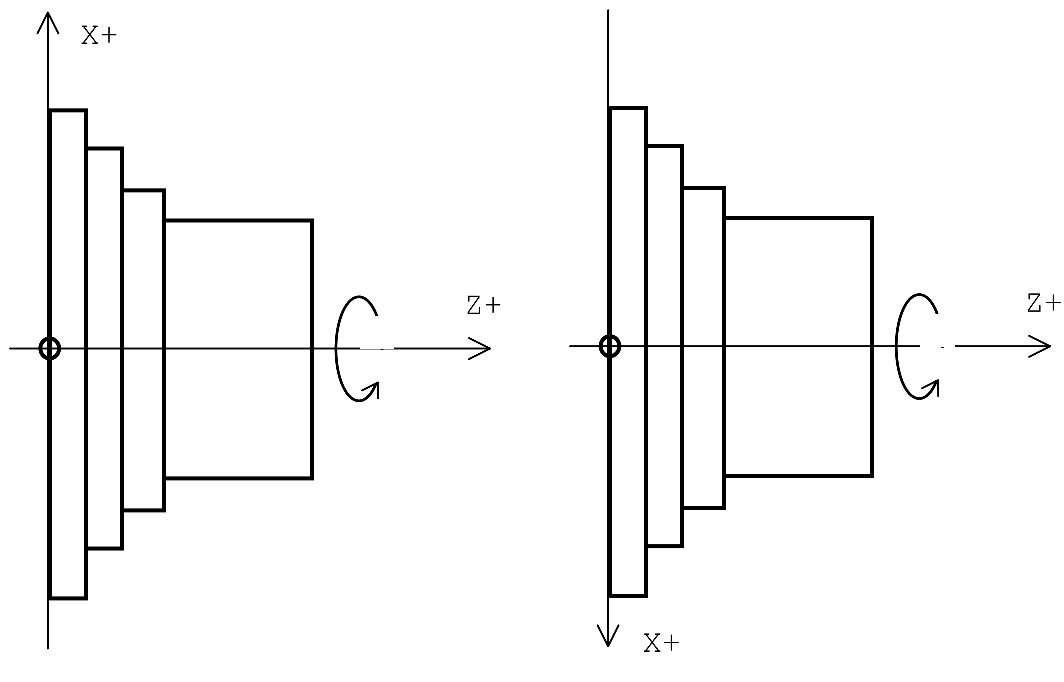
5.1. AXIS MOTIONS

The control governs the axes according to the specifications of EIA RS-267. This standard defines the motion of the tool in relation to the workpiece.

The axes are called X and Z, for diametral dimensions and longitudinal dimensions, respectively. The positive X axis is oriented in the direction moving away from the spindle axis, i.e. towards increasing diameters. The positive Z axis is oriented in the direction moving away from the spindle.

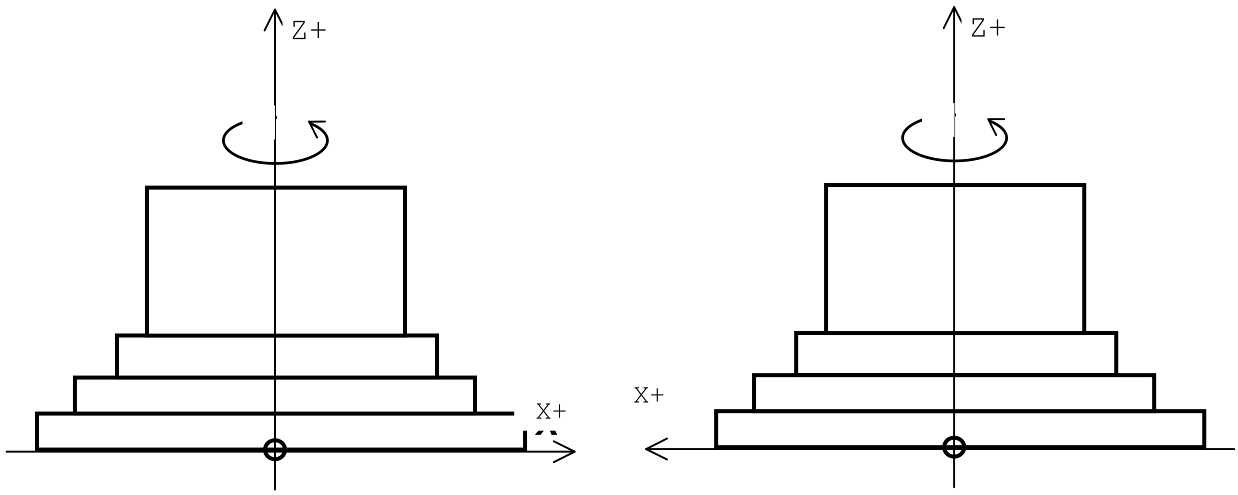
On horizontal lathes the axes can be oriented in the following ways, according to the turret position:

Fig. 5.1. - Axes on Horizontal Lathes



Also on vertical lathes the axes position can be defined in two ways, according to the type of lathe (Fig. 5.2.).

Fig. 5.2. - Axes on Vertical Lathes



5.2. PREPARING FOR PROGRAMMING

Before starting to compile your part program, you should perform the following steps. Note the necessary information on the set up sheet.

1. On the drawing for the workpiece, define the "zero point", that is, the origin and reference for all axis motion. If the drawing has a point to which all dimensions are referred, you will find it convenient to select that point as the zero point. In any case, we advise you to select a zero point that lets you easily use the dimensions from the drawing to determine coordinates for your program.

2. On the workpiece drawing define reference and clamping points.

3. Make sure that all operations you want the control to carry out are within the working area of the machine.

4. List all the necessary tools in the exact sequence you need them to produce the part.

5. Define the spindle speed and feedrate required for each tool.

5.3. TAPE FORMAT

You must store programs in the control memory before you can run them on the machine. You can enter a program at the control via keyboard, or you can prepare a punched tape or data cartridge and load the program into memory using a tape reader or a data cartridge recorder.

The control accepts two types of punched tape formats:

- ISO, when you use a tape reader;
- ISO or EIA, when you use a teletype.

To load a program in memory using other peripherals, refer to Chapter 7.

5.4. PART PROGRAM INFORMATION

We use the following terms to define the building blocks of a part program:

character - a number, letter or symbol that defines information for the control. For example, I, G, %, 3. X, LF (line feed). The allowable characters are shown in Table 5.1.

address - a letter that identifies the type of instruction. For example, G, X, Z, F

word - an address followed by a numeric value. For example: G1, X50.5, Z-3.15, F.200, T1.01 are words. Numeric values associated to each address may have different units of measurement. In general, you can leave off leading and trailing zeroes. For fractional entries always include the decimal point.

block - a set of words that defines an operation required by the control. For example, N3 G X80 Y80 M13 is a block. A block must be less than 128 characters long. A special character -- LF (IS0) or CR (EIA) -- defines the end of each block. Besides, each program block can begin with at least one of the following characters:

- slash (/): block enable/disable;
- label;
- block sequence number.

The slash allows you to indicate that the execution of a block is conditioned by the state of system variable USB. For example, /N100G00X100. If USB=1, the block will be executed as an ordinary program instruction. If USB=0, the block will be handled as a comment.

A label allows you to attribute a name to the block. You can use labels to compile branching instructions. A label must have six alphanumeric characters and is always closed between inverted commas; in slashed blocks, the label always follows the slash.

For example, "START"
/"END"

Each block begins with the number sequence N (1-9999). If present, the sequence must follow the slash and the comment.

For example: N125
"START" N125
/"END" N125

Table 5.1. - CHARACTERS

CHARACTERS	**SYMBOLS**
Capital letters	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Lower-case letters	a c d l m o p r s t u v w
Digits	from 0 to 9
Operators 1	+ -
Operator 2	*
Operator 3	/
Decimal point	.
Label separator	"
Opening bracket	(
Closing bracket	)
Comment symbol	;
Operand separator	,
Symbol	=
Tape start or end	% (ISO) er (EIA)
End of block	L.F. (LINE FEED) (IS0) CR (EIA)
Special characters	:
Prefixes	# (SYNCHRONIZATION ENABLE) & (SYNCHRONIZATION DISABLE)

5.5. TYPES OF BLOCKS

You can use four different types of blocks:

- comment blocks;
- ISO blocks;
- assignment blocks;
- blocks containing three-letter codes.

Comment blocks - allow the programmer to insert alphanumeric descriptive messages, thus increasing the program readability. They are bypassed during execution.

A comment block must begin with ;.

Example: ;THIS IS AN EXAMPLE

ISO blocks - only include ISO operators.

Example: G1 X500 Z20 F200

Assignment blocks - allow you to set the values of certain system parameters within a program. These values can successively be used in other blocks. Assignment blocks allow you to set the following variables:

- calculation variables. For example, E30=28.5
- geometric variables. For example. p2=X10 Z25
- global system variables. For example, UOV=1.5

Blocks containing three-letter codes - allow you to define the operation to be executed by means of a three-letter mnemonics (EIA 1177 B Standard). For example, (URT,45).

5.6. STARTING AND ENDING PART PROGRAMS - MESSAGES

If you produce a program on tape, the first and the last characters must be either % (ISO) or er (EIA).

If you enter a program via keyboard, you need not include either starting or ending characters.

Usually, tool change is programmed in the first block: (T...M06).

When you finish machining, you must program the axes'return to a position that enables part release and (if necessary) tool change (M6) at the beginning of the next program. You must also stop spindle rotation and coolant flow (M5) and establish the automatic program reset (M30). For example:

```
%
N1     (DIS,"ROUGHING TOOL")
N2     GO X63 2A S70 M3 M8
............
............
N236   G X250 21.5 M5 M9
N237   M30
%
```

You can insert a message for the operator inside a program.
When the control executes the block with the message, it appears on the display. To program a message, use the following format:

(DIS,text)

The brackets and the DIS code are compulsory. The text length is limited to 32 characters.

5.7. PART PROGRAM WORDS

The allowable part program words are:

- Axis coordinates: X Z B C
- R
- I J
- K
- S
- T
- F
- G
- M
- H

X Z B C - Format: ±0.0001 to ±99999.9999 mm or in.
To characterise a rotary axis, enter the corresponding word in degrees.

R - Format: ±0.0001 to ±99999.9999 mm or in.
In a standard fixed cycle, R defines the initial or final coordinate.
In a variable pitch threading operation, R defines the angle variance with respect to the spindle zero.

I (abscissa) J (ordinate) - Format: ±0.0001 to ±99999.9999 mm or in.
I and J are the center coordinates in a circular movement.
Their validity is independent from the interpolation plane. I and J are also used in a fixed boring cycle (with G83). In a variable pitch threading cycle. I defines the pitch variance: I+ for increasing pitch, I- for decreasing pitch.

K - Format: ±0.0001 to ±99999.9999 mm or in.
In a fixed cycle with chip discharge (G83), K is the multiplicator of the machining depth. In both threading (with G33) and tapping (with G84) cycles, K defines the pitch.

S - Format: 0.01 to 99999.99

S defines either spindle speed in rpm (with G97) or –if characterised-cutting speed in m/min (with G96)

T - Format: 1.0 to 9999.9999

T defines the desired toml (integer portion) and the corresponding tool offset (decimal portion). Tool offset is enabled by M06.

Offset does not require any other preparatory function.

Tool diameter offset is simultaneous with length ofisat and is enabled by preparatory functions G41/G42.

F - Format: 0.01 to 99999.99

With G93, F defines the inverse ai the item execution time (in min), i.e. the quotient between feedrate and distance. It is only valid within the block.

With G94, F defines feedrate in mm/rpm (with G71) or in/rpm (with G70). It allows you to program the desired time (t) in seconds, F being the quotient between item length and execution time.

With G95, F defines feedrate in mm/min (with G71) or in/min (with G70).

G - G codes define preparatory functions. Refer to table 5.2.

M - M codes (MO to M99) define miscellaneous modal functions. Refer to Table 5.3.

H - H codes provide an expansion of the standard M functions available with the ISO standard.

Table 5.2. - G FUNCTIONS

Code	Modal Group	Only one block?	Description	At power-up
G00	a	no	Rapid axes positioning	yes
G01	a	no	Linear interpolation	no
G02	a	no	Circular interpolation CW	no
G03	a	no	Circular interpolation CCW	no
G33	a	no	Constant or variable pitch threading	no
G27	c	no	Continuous path mode with automatic deceleration on corners	yes
G28	c	no	Continuous path mode with-out automatic deceleration on corners	no
G29	c	no	Point-to-point moves	no
G21	d	no	Open GTL ambient	no
G20	d	no	Close GTL ambient	yes
G40	e	no	Cancel tool tip radius compensation	yes
G41	e	no	Tool tip rad.compensation (tool left of part)	no
G40	e	no	Tool tip rad. compensation (tool right of part)	no
G70	f	no	Inch programming mode	yes
G71	f	no	Metric programming mode	no
G80	g	no	Delete fixed cycles	yes

Table 5.2. continued

Code	Modal Group	Only one block?	Description	At power-up
G81	g	no	Fixed drilling cycle	no
G82	g	no	Fixed spot-facing cycle	no
G83	g	no	Deep hole drilling cycle (with chip discharge)	no
G84	g	no	Fixed tapping cycle	no
G85	g	no	Fixed reaming cycle	no
G90	h	no	Absolute programming	yes
G91	h	no	Incremental programming	no
G79	k	yes	Programming referred to machine zero	no
GO4	i	yes	Dwell at end of block	no
GO9	i	yes	Deceleration at block end	no
G72	j	yes	Measure of a point with radius compensation	no
G74	j	yes	Measure of variance from nominal coordinates without radius compensation	no
G98	n	no	Enable adaptative control	
G99	n	no	Disable adaptative control	

Table 5.3. - M FUNCTIONS

Code	Active at Motion start	Active at Motion end	Can be cancelled by	Meaning
M00		x	Cycle Start	Program stop (spindle and coolant). Other functions are kept stored
M01		x	Cycle Start	Optional program stop If US0=1, M01 is handled as M00. If USO=0, MO1 is bypassed
M02		x		End of program (no tape rewind)
M03	x		M4-M5-M14-M19	CW spindle rotation
M04	x		M3-M5-M13-M19	CCW spindle rotation
M05		x	M3-M4-M13-M14	Spindle or coolant OFF
M06		x		Tool change (with program stop and spindle/coolant off) Does not cancel M03-M04-M08-M13-M14
M07	x		M9	Auxiliary coolant ON
M08	x		M9	Main coolant ON
M09		x	M7-M8	Coolant OFF
M19	x		M3-M4-M5-M13-M14	Spindle stop and angle orient
M20	x		M21	Disable coordinated spindle
M21	x		M20	Coordinate spindle axis with X and Z axes

Table 5.3. continued

Code	Active at Motion start	Active at Motion end	Can be cancelled by	Meaning
M30		x		End of program with tape rewind and control reset Clear dynamic buffer Current tool offset is kept stored
M41	x		M42-M43-M44-M40	Spindle gear range selection 1
M42	x		M41-M43-M44-M40	2
M43	x		M41-M42-M44-M40	3
M44	x		M41-M42-M43-M40	4
M40		x		Cancel spindle gear range selection
M45	x		M41-M42-M43-M44	Automatic spindle gear range change

N.B. - All M functions can be deleted with the RESET key.

5.8. PROGRAM BLOCKS WITH G FUNCTIONS

Preparatory G functions range from G00 to G99. At present, only the codes shown in Table 5.2. are avai lable with the NC-110.

Within a program block, always insert the G function immediately after the block number (if present) and before any other operand.

You can program a G operand either explicitly or implicitly, by means of an E parameter (with a variable of the byte type).

We have classified G functions in thirteen modal groups. They are listed in Table 5.4.

Table 5.4. - MODAL GROUPS

GROUP	G FUNCTIONS	DESCRIPTION
a	G00-G01-G02-G03-G33	Definition of the movement
b	G27-G28-G29	Definition dynamic mode (point-to-point or continuous)
c	G21-G20	Open and close the GTL programming ambient
d	G40-G41-G42	Tool radius offset enable and disable
e	G70-G71	Programming in alternative unit
f	G81..G89	Standard fixed cycles
g	G90-G91	Incremental/absolute program
h	G79	Programming referred to machine zero
i	G04-G09	Attributes of the dynamic mode
j	G72-G74	Measuring cycles
k	G93-G94-G95	V/D feedrate coding
l	G96-G97	Spindle speed
m	G98-G99	Enable/ disable adaptative control

5.8.1. MODAL GROUP A: DEFINITION OF YNE TYPE OF MOVEMENT

The allowable functions are:

GOO rapid axes positioning
G01 linear interpolation
GO2 circular interpolation CW
GO3 circular interpolation CCW
G33 constant or variable pitch threading

Important: Key to symbols:
[] enclose optional elements;
{} enclose alternative elements.
Intermediate zeroes can be omitted:
For example, G00 = G G01=G1

G00 - RAPID AXES POSITIONING

Use this preparatory function to program rapid linear position of the axes.
The allowable format is:

G00 [OTHER G CODES] [AXES] [FEED] [AUXILIARY FUNCTIONS]

where:

[OTHER G CODES]	all the compatible G functions
[AXES]	the axes words followed by a numerical value, explicit or implicit. To implicitly define the axes, you must first define a point according to the current abscissa and ordinate.
[FEED]	machining feedrate for coordinated movements. It remains stored but does not determine the movement of the axes specified in the block.

The block feedrate in GOO is the vectorial composition of the rapids of the programmed axes. It is stored in the characterization file.

[AUXILIARY FUNGTIOHS] the programmable auxiliary functions are M, H, S, T.
It is possible to program as many as:
4 M's, 4 H's, 1 S (speed) and 1 T.

G01 - LINEAR INTERPOLATION

Use this function to program a linear interpolation of all the specified axes at programmed federate.

The allowable format is:

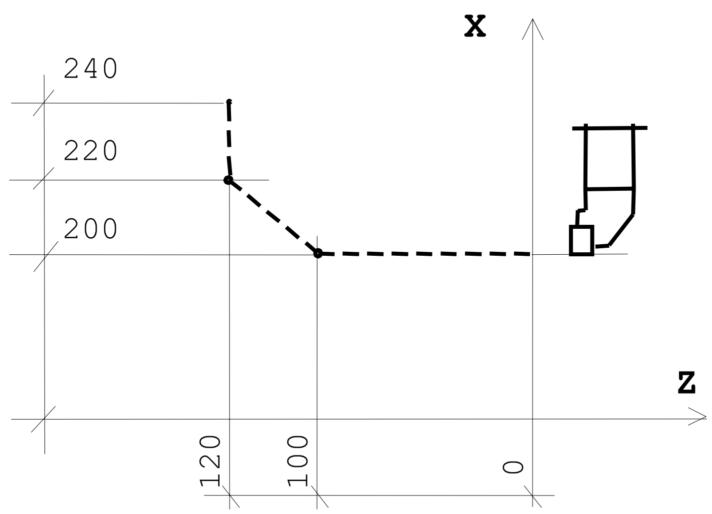
G01 [OTHER G CODES] [AXES] [FEED] [AUXILIARY FUNCTIONS]

where:

[FEED] If it is not specified, the control assumes the stored value. If no feedrate has been stored, an error message is displayed.

For the remaining elements, refer to the description of GOO.

Fig. 5.3. - Linear interpolation




```
N10  G1  Z-100 F0.3
N20  X220 Z-126
N30  X240
```

G02 GO3 - CIRCULAR INTERPOLATION

Use these codes to program circular movements, respectively CW (GO2) or CCW (GO3).
The programmed axes move along the circular path at programmed feedrate.

The allowable format are:

1) for circular interpolation with center coordinates (I and J)

```
{GO2}[OTHER G CODES][AXES] I J [F]
{G03}
```

2) for circular interpolation with the circle radius

```
{GO2}[OTHER G CODES][AXES] CR [F]
{G03}
```

where:

- G codes, miscellaneous operators and auxiliary functions are those explained in G00.

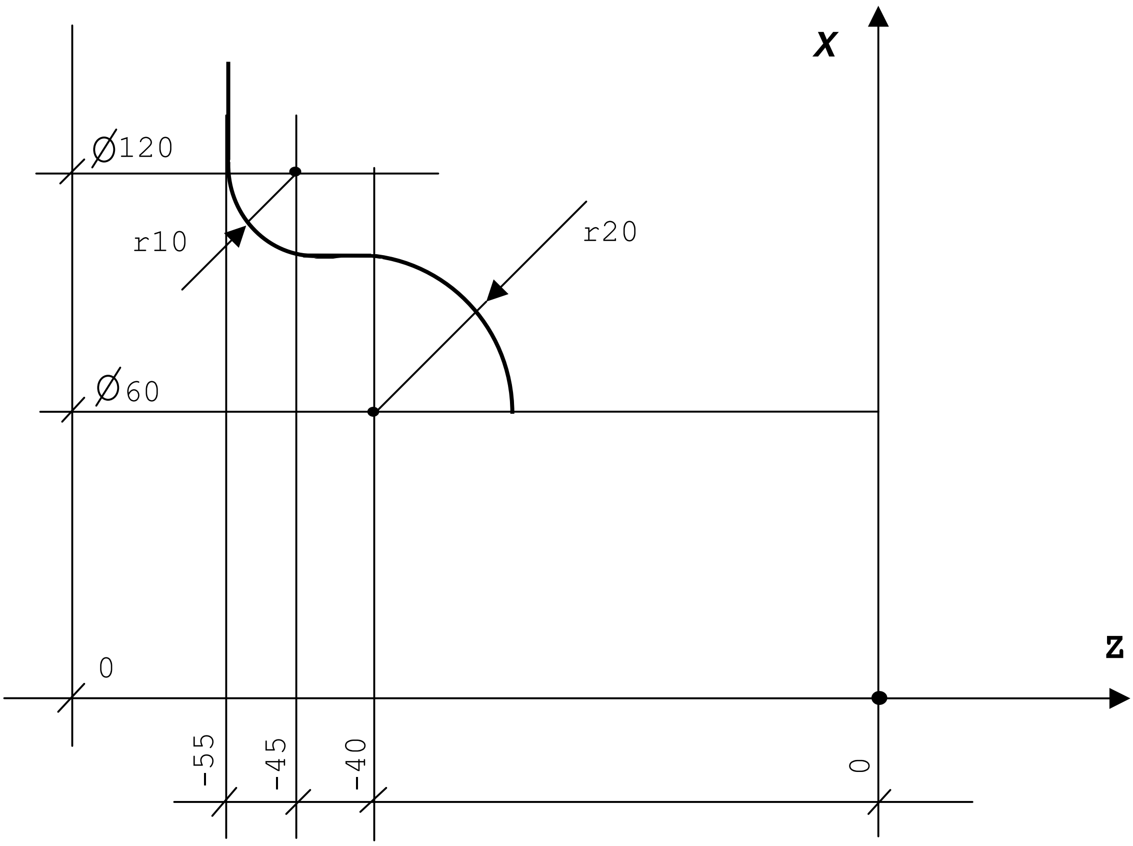
- [FEED] has been explained in GO1.

- the axes are represented by the corresponding word followed by a numerical value, either explicit or implicit (E parameter). If no axis is specified, the code generates a complete circular movement in the interpolation plane. You can define axes implicitly by specifying a geometric item (i.e. point).

- I and J are the coordinates of the circle center. They are compulsory. Always enter I and J (independent from the interpolation plane), followed by a real number, either implicit or explicit (E parameter).

- R is an address followed by a + or a - sign (+ is normally omitted) and a real numerical value (explicit or implicit, i.e. E parameter).
It defines the value of the R radius of the circle. If the circle arc ≤180 degrees, then the radius has a positive value. If the circle arc ≥180 degrees, the radius has a negative value.

Fig. 5.4. - Circular interpolation




```
N10 G1 X60 Z-20
N11 G3 X100 Z-40 I-40 J60
N12 G1 Z-45
N13 G2 X120 Z-50 I-45 J120
N14 G1  X...
```

Notes:

- The maximum programmable arc is 360 degrees.

- Before defining the path, you must specify the interpolation plane. You can either enter preparatory functions G17-G18-G19 or the three-letter code DPI (the interpolation plane is the one generated by the specified couple of axes). In any case, G17 is present at power-on.

- When establishing the coordinates of the circle start. Doint (programmed in the previous block), end point and centre, the difference between start and end radius must be less than 0.01 mm.

  If the current coordinates outnumber this value, the following message is displayed: INCONGRUENT PROFILE and the circular movement is not executed.

- You can also program a circular interpolation in incremental mode, i.e. coordinates of the circle end point and of the center referenced to the start point programmed in the previous block.

• The direction of the circular movement (CW or CCW) is defined by looking at the interpolation plane from the positive semiaxis normal to it.

G33 - CONSTANT OR VARIABLE PITCH THREAD

Use this code to program a cylindrical or conical thread with constant or variable pitch. The axes movement is coordinated with spindle rotation.

The allowable format is:

G33 [AXES] K [I] [R]

where:

[AXES]	axis name followed by a numeric value in explicit or implicit mode (E parameter).
K	thread pitch If pitch is variable, K represents the initial value. It cannot be omitted.
I	pitch variation For increasing pitch threads, program a positive I; for decreasing pitch threads, program a negative I.
R	angle deviation from the spindle zero (in degrees). Program R is used for multi-start threads, to avoid shifting the start point.

Notes:

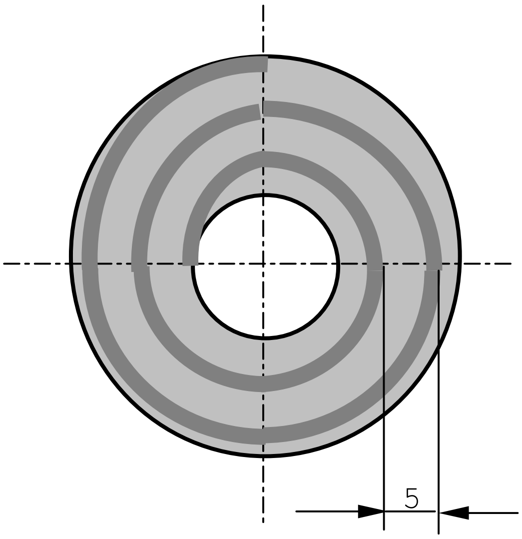
• During threading, HOLD, FEEDRATE OVERRIDE and SPINDLE SPEED OVERRIDE are disabled.

• You can only program G33 if your spindle features a transducer. It is programmed in turns per minute (G97).

Here are some examples of constant and variable pitch threading:

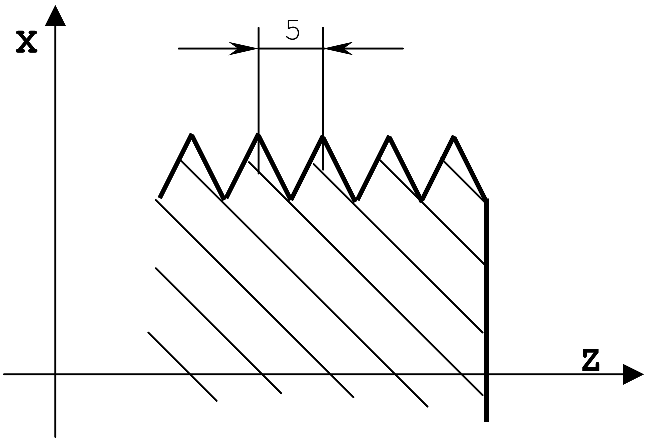
Fig. 5.5. – Constant pitch threading

a) Front thread



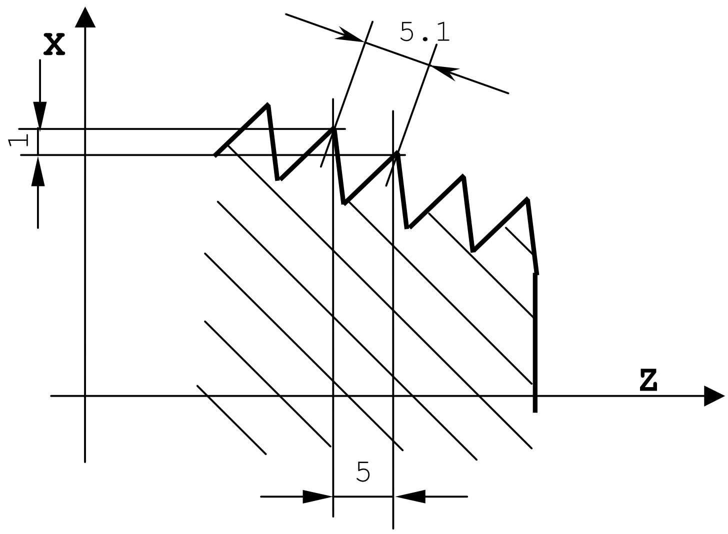
N15 G33 X0 K5

b) Cylindrical thread



N22 G33 Z-50 K5

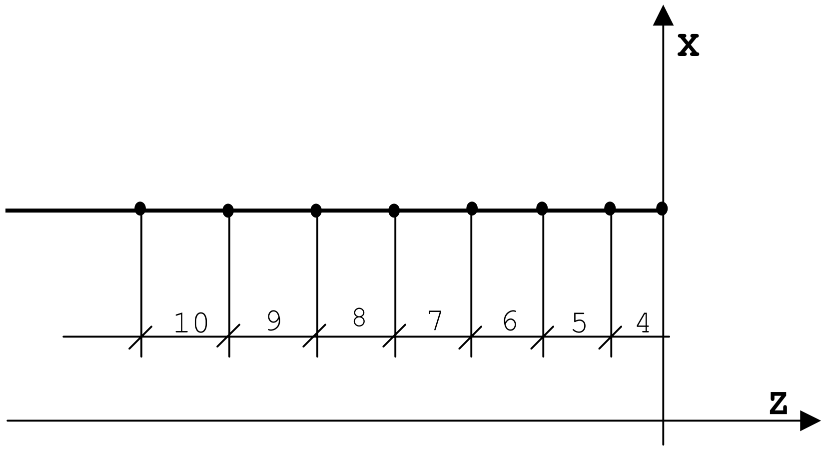
c) Conical thread



N22 G33 X50 Z-50 K5.1

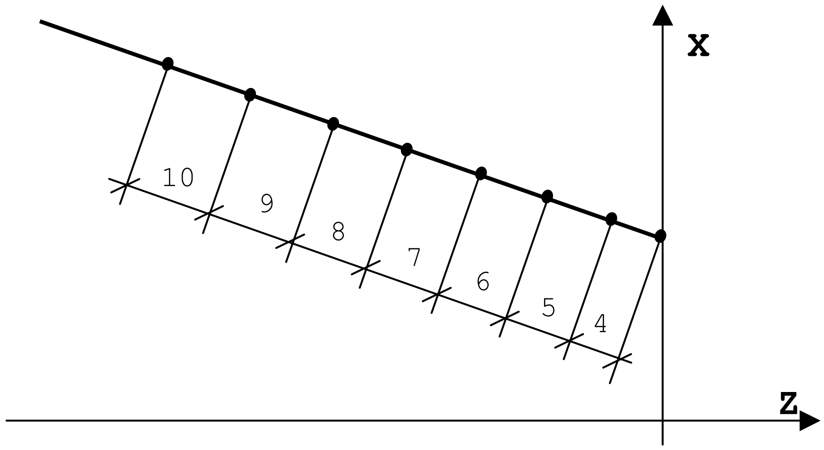
Fig. 5.6. – Variable pitch thread

a) Cylindrical thread with increasing pitch



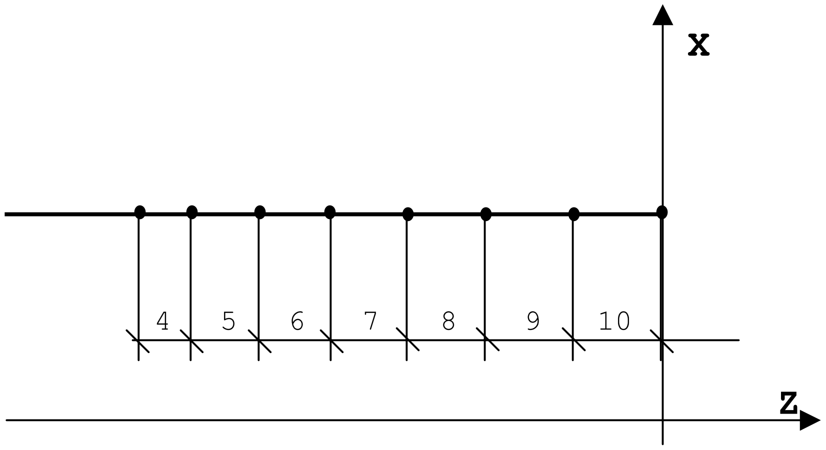
N24 G33 Z-50 K4 I1

b) Conical thread with increasing pitch



N24 G33 X50 Z-40 K4 I1

c) Cylindrical thread with decreasing pitch



N24 G33 Z-50 K10 I-1

Note. In decreasing pitch threads, initial pitch, pitch variations and thread length must be set so that the pitch does not reach zero before the end of the thread.

To check decreasing pitch threads, use the following formula:

$$I \leq \frac{K^2}{2\ (Zf - Zi)}$$

where:

```
I           = maximum pitch variation
K           = initial pitch
(Zf - Zi)   = thread length (end point coordinates-start point
              coordinates)
```

Example of triple-start thread

```
..........
N37 G33 Z3 K6           1st thread
..........
..........
N41  G33 Z3 K6 R120     2nd thread
..........
..........
N45 G33 Z3 K6 R240      3rd thread
..........
```

The R code allows you to set a series of start points, without programming coordinates for each point individually. R shifts the start point of the subsequent threading operations by a value equal to the pitch divided by the number of starts.

5.8.2. MODAL GROUP B: DEFINITION OF THE DYNAMIC MODE

The functions belonging to this class define the execution mode.

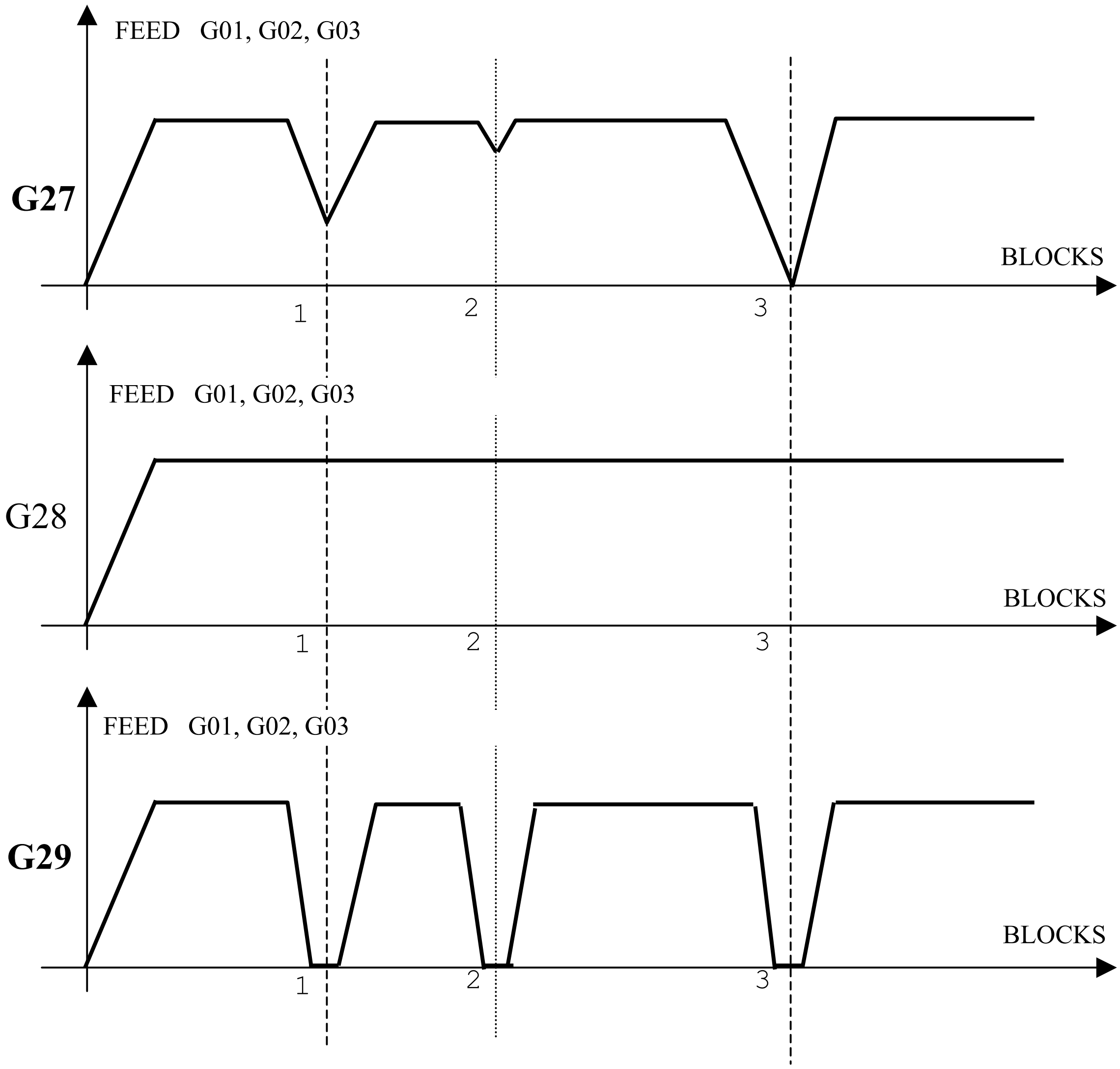
The allowable formats are:

{G27}
{G28} [OTHER G CODES] [OPERANDS]
{G29}

where:

[OPERANDS]	indicates all the compatible G codes.
G27	programs a continuous movement with automatic deceleration on corners. The control automatically calculates speed according to the shape of the profile.
G28	programs a continuous movement without automatic deceleration on corners. The output speed is equal to the programmed speed.
G29	programs a point-to-point movement, i.e. the output speed on the profile items is set to 0.

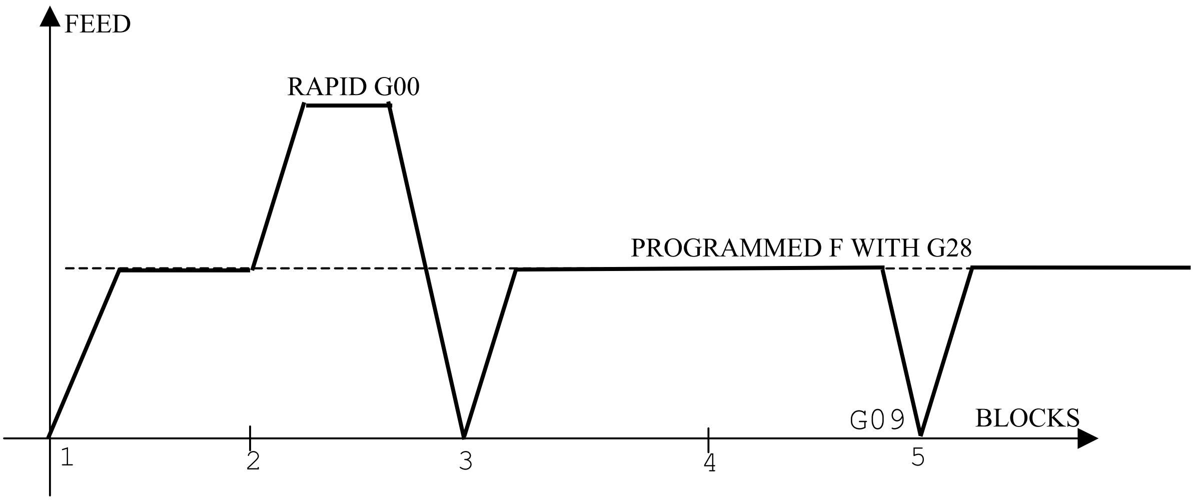
Fig. 5.7. - Graphic representation of the dynamic mode



Notes:

• With G1-G2-G3, you can program the positioning mode by entering G27, G28 or G29. With G0, positioning always occurs in point-to-point mode (i.e. with deceleration until speed reaches zero), independent from the state of the system (G27, G28 or G29).

Fig. 5.8.



• G27-G0 are automatically enabled at power-up and after each Reset.

• In continuous mode (with G27-G28), the control explores the profile to be executed and reads the profile elements as a single block. Therefore, within a G27-G28 path, you cannon program M, H, S nor T functions.

Since the continuous mode is temporarily closed by a G00 still belonging to the profile, program the desired auxiliary functions after G00.

Example:
```
N2O G1  Z-200
N21 G X200
N22 M5
```

Inside a continuous sequence (G28) you can program a deceleration at the end of the block with defined positioning, by entering a G09 valid only within the block.

5.8.3. MODAL GROUP B: GTL PROGRAMMING

To trace a profile with GTL, use the following functions:

- G21 to open the GTL ambient;
- G20 to close the GTL ambient.

The allowable formats are:

```
{G20}                  {pn}
{G21} [OTHER G CODES] {ln} [s2] [AXES] [FEED] [AUX. FUNCT.]
                       {cn}
```

where:

pn, ln, cn — indicate, respectively, a point, a straight line or a circle; n must have been previously defined. Program an to obtain an open profile; never program an inside the profile.

s2 — indicates the second intersection between two items (circles or straight lines); you need not program the first intersection.

The remaining parameters have been described in section 5.8.1. Notice that you can only specify axes not belonging to the interpolation plane.

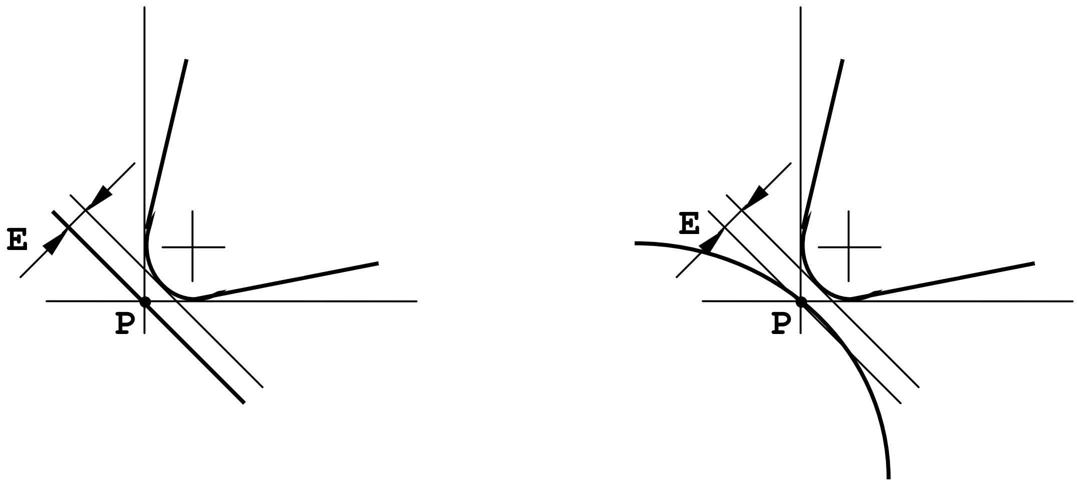
5.8.4. MODAL GROUP E: TOOL RADIUS COMPENSATION (G41-G42-G40)

When contouring a profile, the theoretical tool end P must follow a different path from the theoretical profile, so that the tool center can cover a profile perfectly equally spaced from the workpiece profile.

The resulting profile coincides with the theoretical profile only along the faces parallel to the axes.

E= 0.41 x tool radius

Fig. 5.9. - Theoretical Workpiece Profile

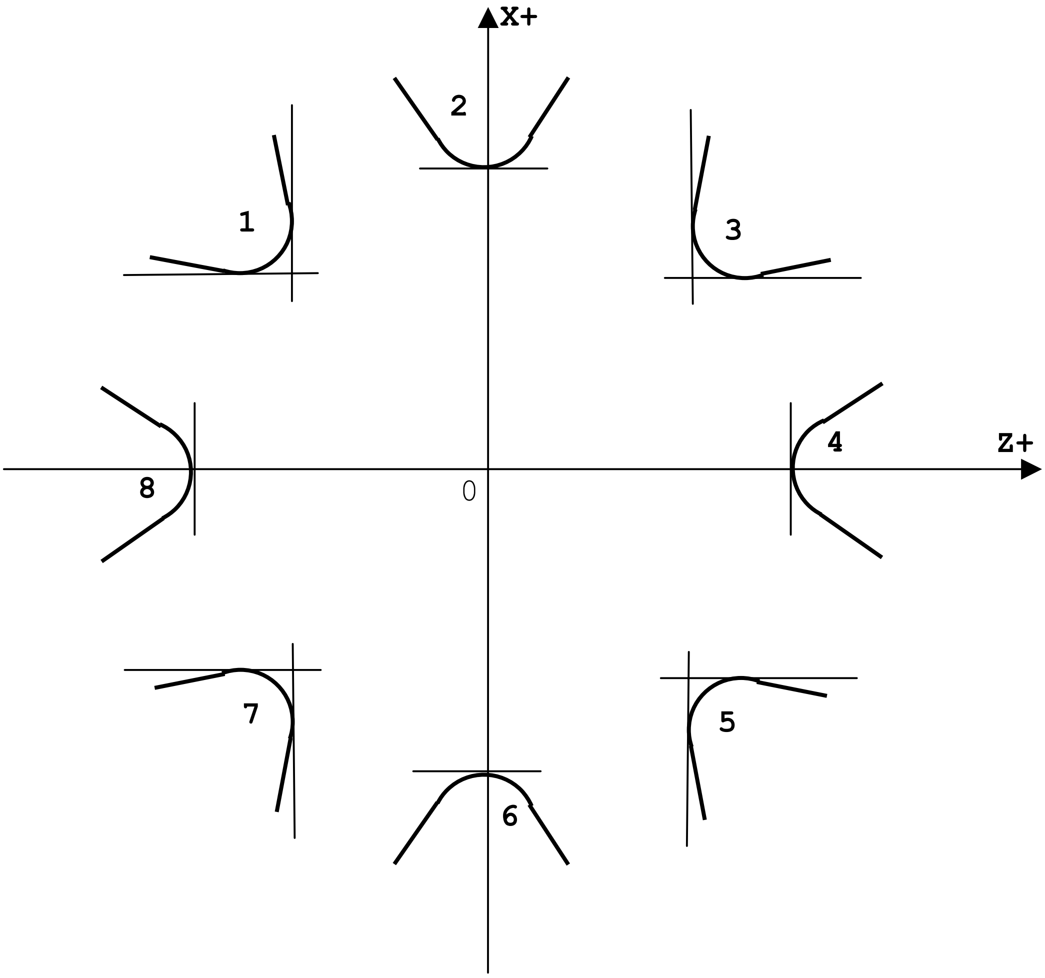


Using tool radius compensation, you must only program the theoretical profile, as the control calculates the path of the theoretical end of the tool according to the radius and the orientation of the tool. These two parameters are stored in the offsets table.

The meaning of the orientation codes (from 0 to 8) is the following:

0 = P point coinciding with the tool center
1 = P point towards X- Z+
2 = P point towards X-
3 = P point towards X- Z-
4 = P point towards Z-
5 = P point towards X+ Z-
6 = P point towards X+
7 = P point towards X+ Z+
8 = P point towards Z+

Fig. 5.10. - Orientation Codes



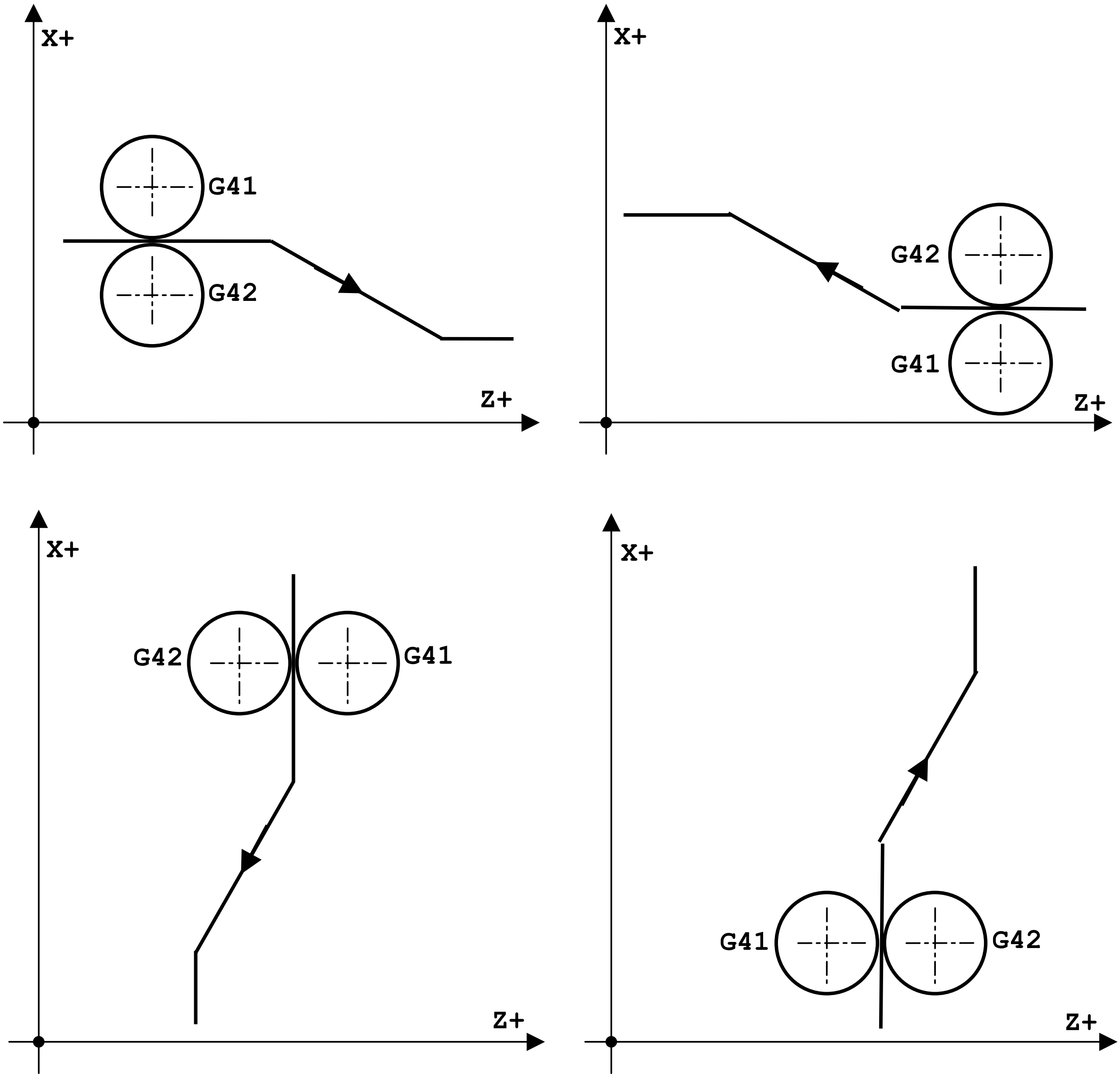
To enable/disable tool radius compensation, program the following functions:

- G41 enable compensation, tool left of part;
- G42 enable compensation, tool right of part;
- G40 disable compensation.

The allowable format is:

{G41}
{G42} [OTHER G CODES] [OPERANDS]
{G40}

Fig. 5.11. – Tool radius compensation

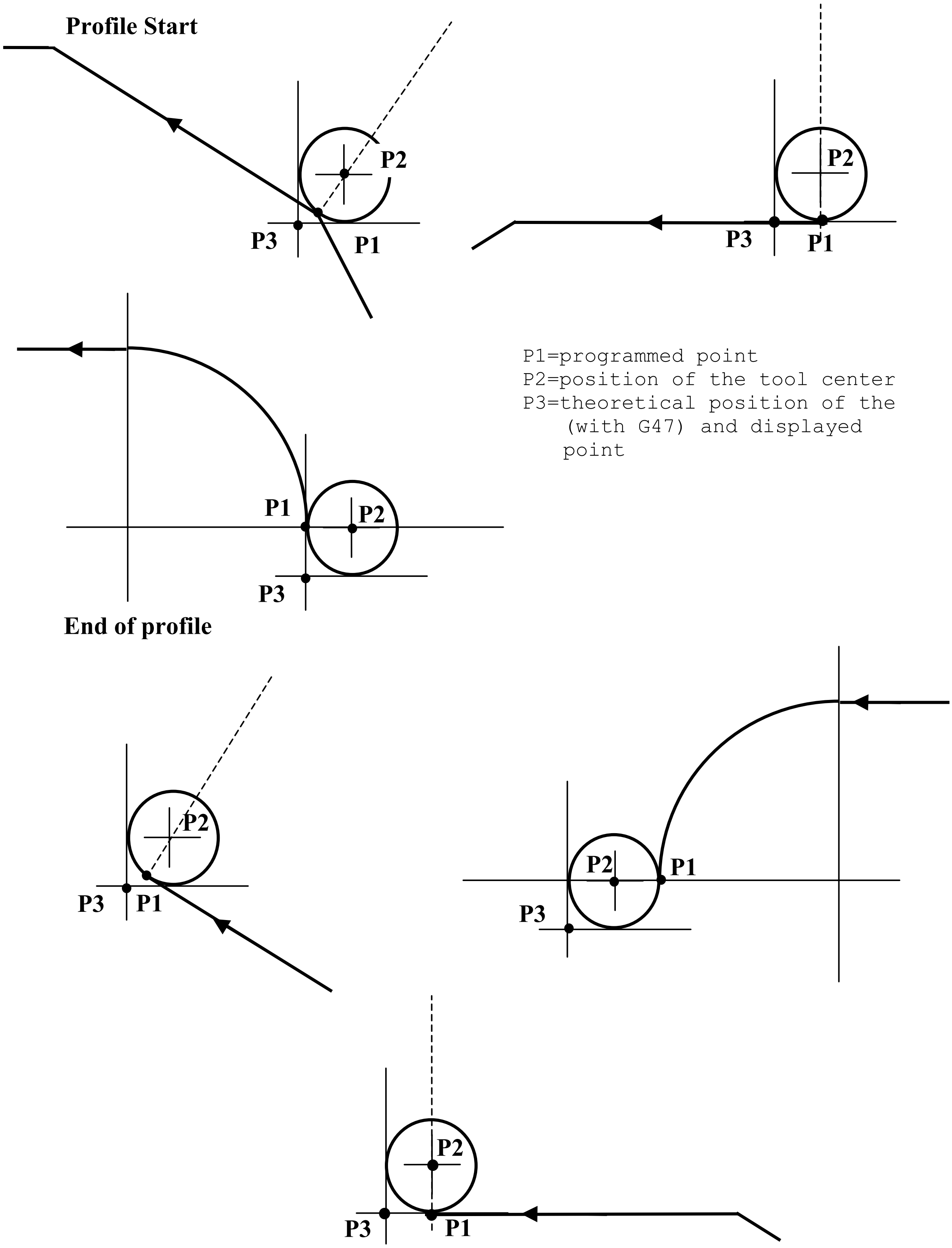


When programming a profile with cutter radius compensation, remember that:

- the first traverse must be linear, either at rapid or at machining rate (G00-G01);
- roughing, threading and grooving operations can not programmed inside the cycle;
- blocks with M, H, S and T functions cannot be programmed inside the cycle;
- the profile can be machined in continuous (G27-G28) or point-to-point mode (G29), in AUTO or SEMIAUTO;
- tool radius compensation is deactivated by G40, which must be programmed in the last profile block;
- G00 does not exclude compensation.

On the first and last point of the profile, the tool center is positioned perpendicular to the profile at the programmed point. Consequently, the position of the theoretical tool tip depends on the orientation code.

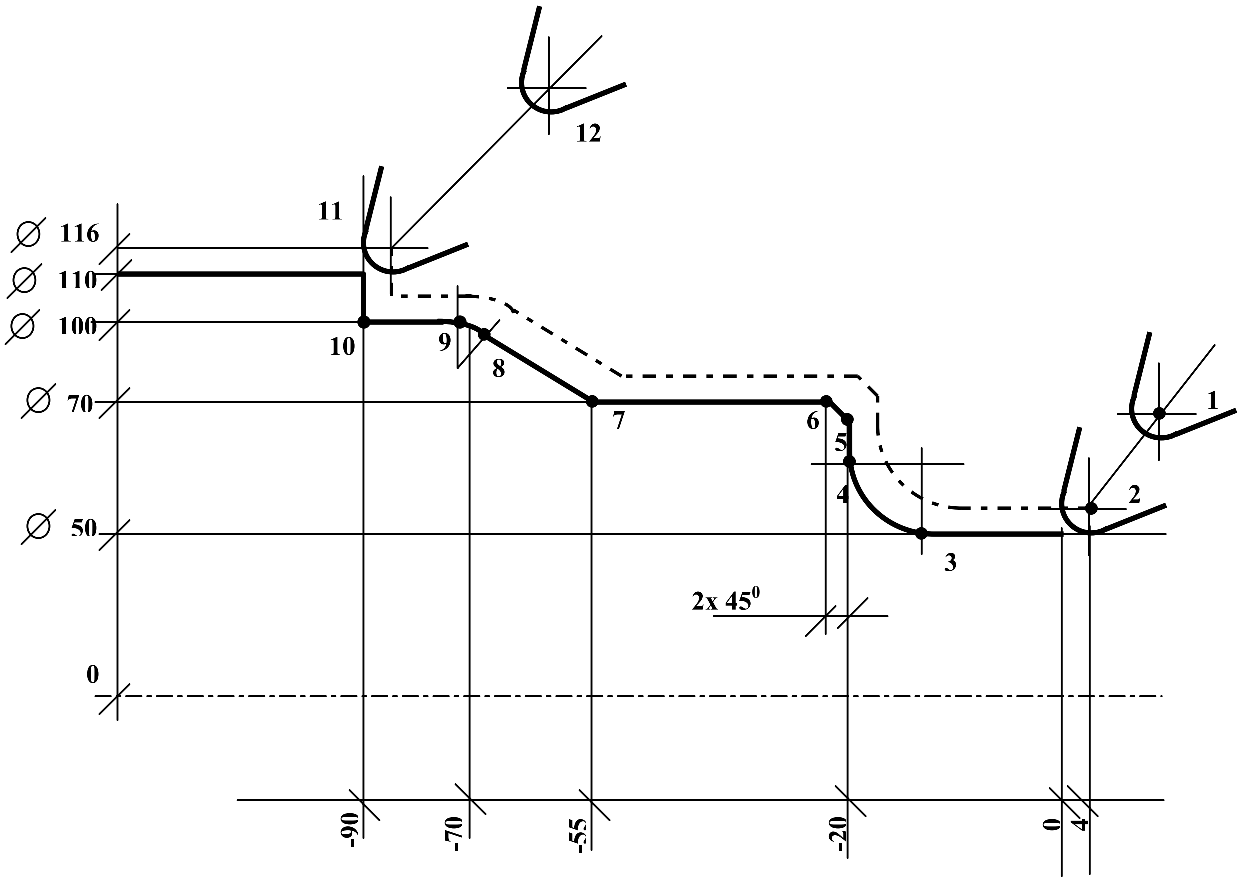
Fig. 5.12. - Profile Start and End with Tool Tip Compensation



P1=programmed point
P2=position of the tool center
P3=theoretical position of the (with G47) and displayed point

Fig. 5.13. - Programming Example

```
Tool radius = 2
Orientation Code = 3
```



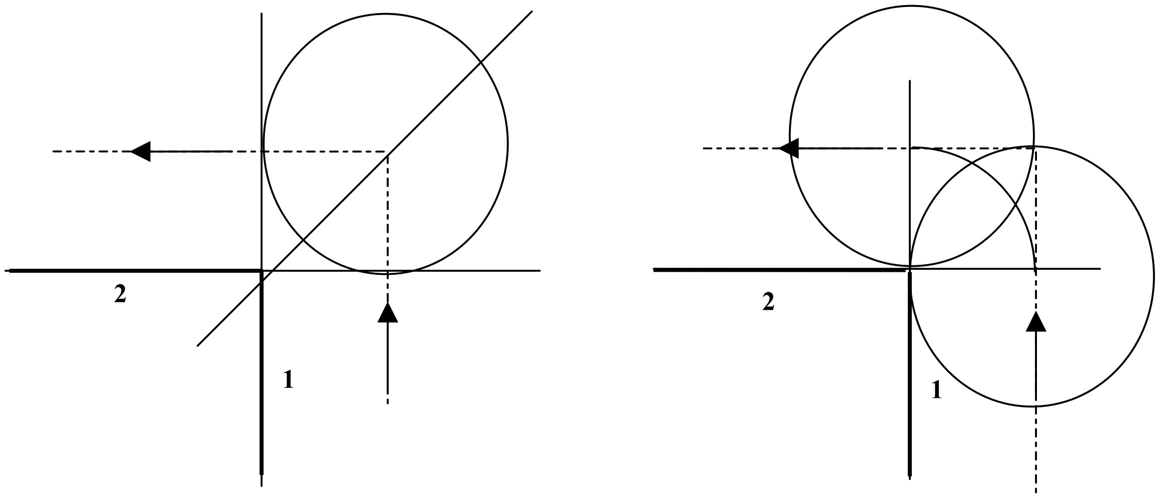

```
   N30 (DIS,"FINISH")
   N31 T4.4 M6
   N32 S100 M3 M7
1  N33 G X64 Z7
2  N34 G1  G42 X50 Z4 F.1
3  N35 Z-15
4  N36 G2 X60 Z-2O I-15 F.1
5  N37 G1 X66
6  N38 X70 Z-22
7  N39 Z-55
8  N40 X97.07 Z-68.535
9  N41 G3 X100 Z-72.O71 I-72.O71 J90
10 N42 G1 Z-90
11 N43 G40 X116
12 N44 G X.. Z..
.................
```

Notes:

• When programming a convex path with CW movement, the radius (r) linking the lines must have a positive value; for a CCW movement, program a negative radius.

Examples (Fig. 5.14.)

Without radius	With radius
....	....
1) N20 G1 X100	1) N20 G1 X100
2) N21 Z-100	N21 r0
....	2) N22 Z-100

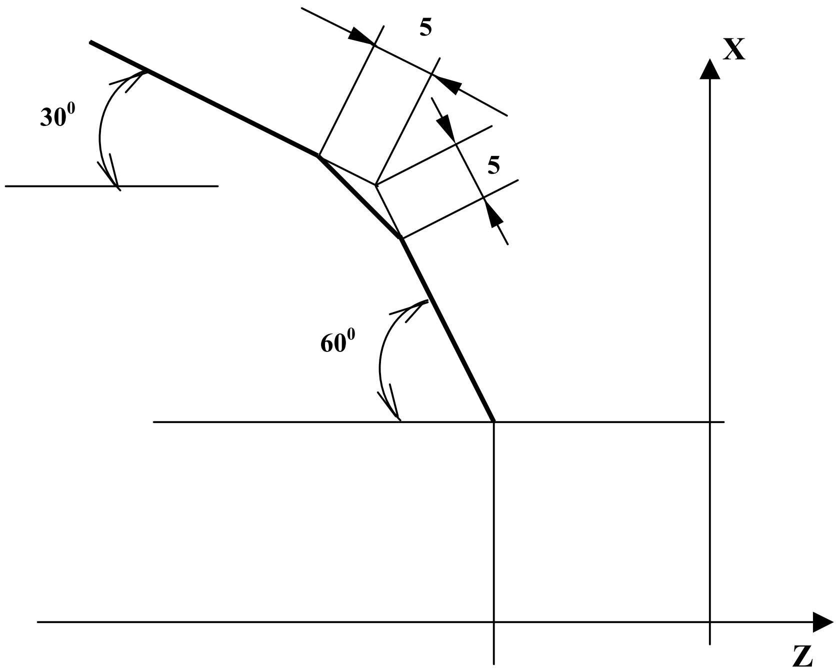


A radius r=0 optimises the tool path by generating a radius equal to zero on the part.

• To program a bevel (b) with tool compensation, enter the bevel value without sign. The control reads the bevel as the distance from the intersection between the lines.

Example (Fig. 5.15.)

```
N10 G1 X50 Z
N11 X100 Z-30
N12 b5
N13 Z-60
```



• In a GTL profile, you can program tool radius compensation by including the operators G21 and G41/G42 in the same block. In this case, you must also program deactivation codes (G20 and G40) in a single block.

5.8.5. MODAL GROUP F:MEASURING UNITS (G70-G71)

To select the unit of measure, program the following functions:

- G70 inch programming;
- G71 mm programming.

The allowable format is:

{G70}
{G71} [OTHER G CODES] [OPERANDS]

If you do not specify any unit of measure, the control automatically defaults the unit defined during characterization.

5.8.6. MODAL GROUP G: FIXED CYCLES (G81-G89)

Functions G81 to G89 allow you to program a fixed cycle of operations (hole drilling, tapping, boring, etc.), specifying only once the parameters of the hole.

The allowable format is:

G8X [OTHER G CODES] R SPINDLE AXIS [MISCELLANEOUS OPERANDS] [FEED] [AUXILIARY FUNCTIONS]

where:

[OTHER G CODES]	other preparatory functions compatible with the fixed cycle.
R	this spindle coordinate defines the start machining point (reached by the tool at rapid) and the final point. It can be specified in explicit or implicit mode (E parameter).
SPINDLE AXIS	the hole depth, expressed in explicit or implicit mode (E parameter).
[FEED]	machining feedrate; if not specified, the control automatically defaults the last programmed value. Its format is: Fnn (nn being a numeric value);
[MISCELLANEOUS OPERANDS]	operands defining parameters specific of the operation. For example, I,J,K in a deep drilling cycle.
[AUXILIARY FUNCTIONS]	S, M, T and H.

FIXED CYCLE MOVEMENTS

A typical fixed cycle includes the following stages:

Stage 1: rapid positioning on the hole axis;

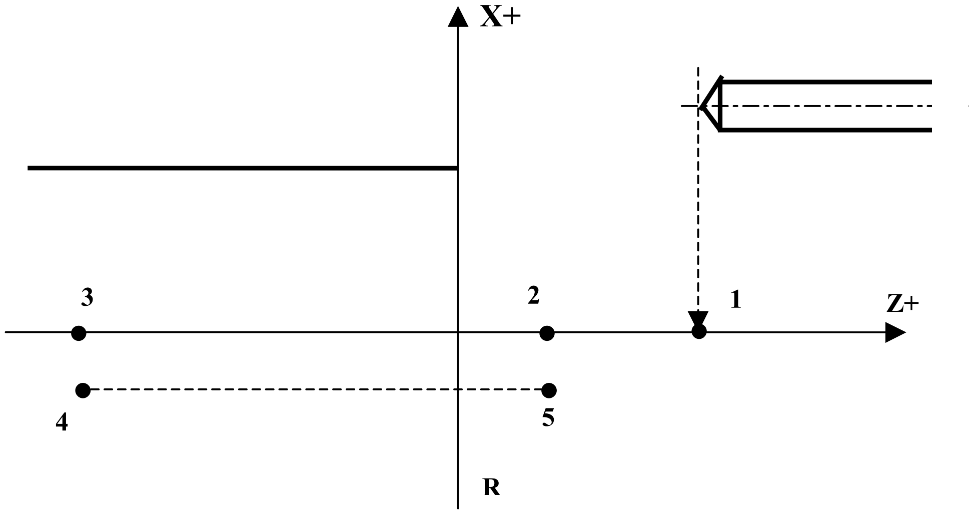
Stage 2: rapid approach to the machining plane (R);

Stage 3: feedrate movement up to the programmed depth (Z);

Stage 4: bottom of hole operations;

Stage 5: rapid or feedrate return to final point (R).

Fig. 5.16. - Example of fixed cycle



Notes:

• The block programming the fixed cycle must not include any other movement instruction for the remaining axes: only use it to store the fixed cycle. The cycle starts with the coordinates programmed in the block that follows the fixed cycle declaration.

• Program dwells with TMR.

• You cannot program a new fixed cycle unless you close the preceding one with G80.

• If tool radius compensation is active (G41/G42), you cannot program a fixed cycle.

Table 5.5. - FIXED CYCLES

Code function	Approach	Bottom end		Return
		Dwell	spindle rotation	
G81 drilling	feedrate	no	normal	rapid
G82 spot-facing	feedrate	yes	normal	rapid
G83 deep drill. <with chip discharge)	intermittent feedrate: down at feedr. retracts rapid	no	normal	rapid
G84 tapping	feedrate - spindle rot. starts	no	rotation reversal	feedr. rapid
G85 reaming	feedrate	no	normal	feedr.
G86 boring	feedrate - spindle rot. starts	no	stop	rapid
G89 boring with spot-facing	feedrate	yes	normal	feedr.
G80.cancels...... fixed cycles	. ...	....	....	......

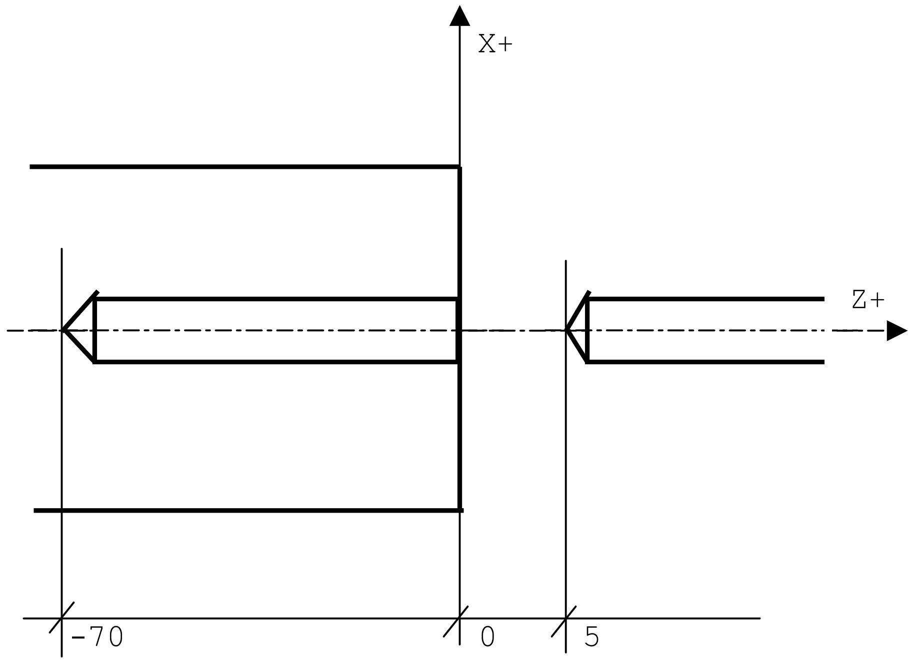
The rapid return phase is-generated as feedrate (G01) with rapid acceleration.

G81 Drilling cycle

The format of the programming block is:

G81 R.. Z..

Fig. 5.17. - Drilling cycle




```
N31  (DIS."TWIST DRILL D=6.5)
N32 G97 S1000 T4.4 M6 M3 M7
N33 G81  R5 Z-70 FO.2
N34 XO
N35 G80
N36 G.. X.. Z..
```

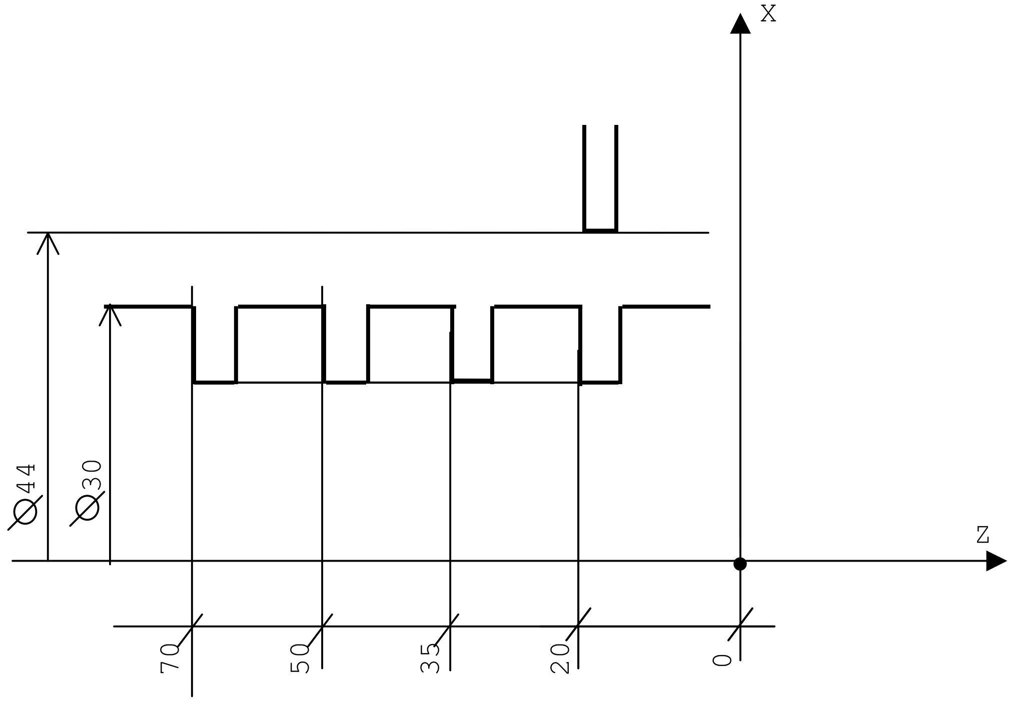
With G81 you can also program fixed cycles for center drilling, boring and reaming.

To program the remaining fixed cycles (with G82, G85, G86 and G89), follow the same steps. For G82 and G89 insert, if necessary, a dwell (with TMR)-before the fixed cycle block.

For example, if the dwell time is of 2 s. the program sequence will be:

```
N33 TMM=2
N34 G82 R5 Z-1OO FO.2
N35 XO
N36 G80
```

Fig. 5.18. - Example of G82 applied to the X axis for grooving




```
N10 T4.4 M6 S100 M3 M8
N11 TMR=2
N12 G82 R44 X30
N13 Z-20
N14 Z-35
N15 Z-50
N16 Z-70
N17 G80 X.. Z..
..............
```

G83 Deep drilling

The programming block hasthe following format:

G83 R.. Z..I..[K..][J..]

where:

R	hole start point (as for G81)
Z	bottom end (as for G81i)
I	Z increment after each pull-out cycle for chip discharge
J	minimum increment of the pull-out cycle increment; when J reaches the programmed value the cycle proceeds by constant increments
K	reduction coefficient of I (until I reaches J)

The presence or absence of these parameters generates two different cycles:

- if I, K and J have been specified the cycle proceeds as follows:

a) rapid approach to the hole axis;
b) rapid approach to R;
c) machining feedrate to R+I;
d) rapid approach to R (chip discharge);
e) calculation of the new R value
   R = R + I - 1
f) calculation of the new I value
   I = I * x if I * K ≥ J
   I = J   if I * K < J

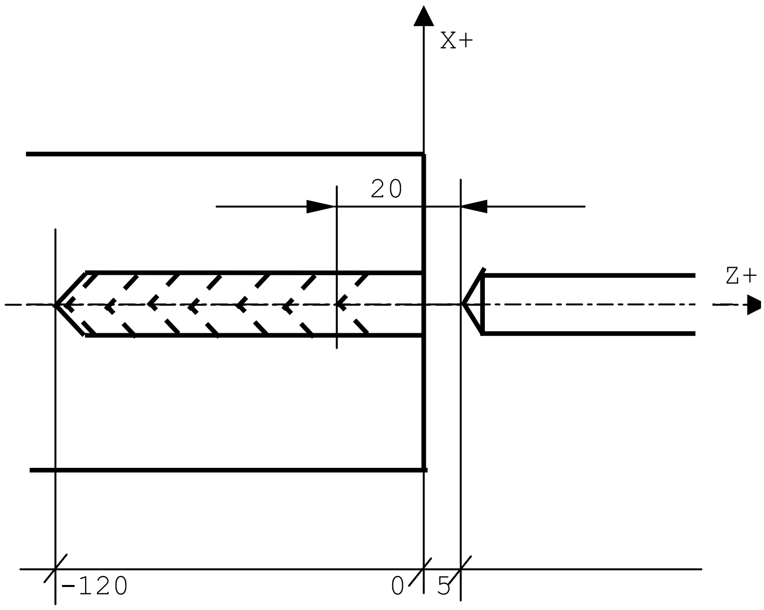
Steps b, c, d, e, f, follow one another until depth reaches the programmed value.
To keep I invariable, that is, to get a constant increment, program K=1 and omit J;

- if K and J have not been specified (woodpecking without discharge), the cycle proceeds as follows:

a) rapid approach to the hole axis;
b) rapid approach to R;
c) machining feedrate to R= R + i;
d) dwell programmed with TMR;
e) approach by another I.

Steps c, d, e, follow one another until the tool reaches the programmed depth.

Fig. 5.19.




```
N65 (DIS."TWIST DRILL D=6")
N66 G97 S1000 T3.3 M6 M3 M7
N67 G83 R5 Z-120 I2O K.8 J6
N68 XO
N69 G80
..........
..........
```

G84 Tapping cycle

The programming block is:

G84 R.. Z.. K..

where:

G84	code for fixed tapping cycle;
R	rapid approach and feedrate return dimension;
Z	end of tapping dimension;
K	tapping pitch.

Example:

```
N90 (DIS,"TAP M8")
N91 G97 S280 T4.4 M6 M3
N92 G84 R5 Z-15 K2
N93 XO
N94 G80
```

The example refers to a tapping with standstill tap and rotating workpiece with the transducer on the spindle.

If the spindle has a motorised tool, you can –through auxiliary M functions-- switch the spindle with the motorised tool on the turret and tap with motorised tool and standstill workpiece. In this case, there are two possibilities: motorised tool with or without transducer.

If the motorised toml has the transducer, the tapping cycle is programmed as above.

If the motorised tool does not have a transducer, the programming block becomes:

G84 R.. Z.. F..

where:

G84	fixed tapping cycle;
R	rapid approach and feedrate return dimension;
Z	end of tapping dimension;
F	feedrate.

5.8.7. MODAL GROUPS H-I: ABSOLUTE/INCREMENTAL PROGRAMMING (G90-G91-G79)

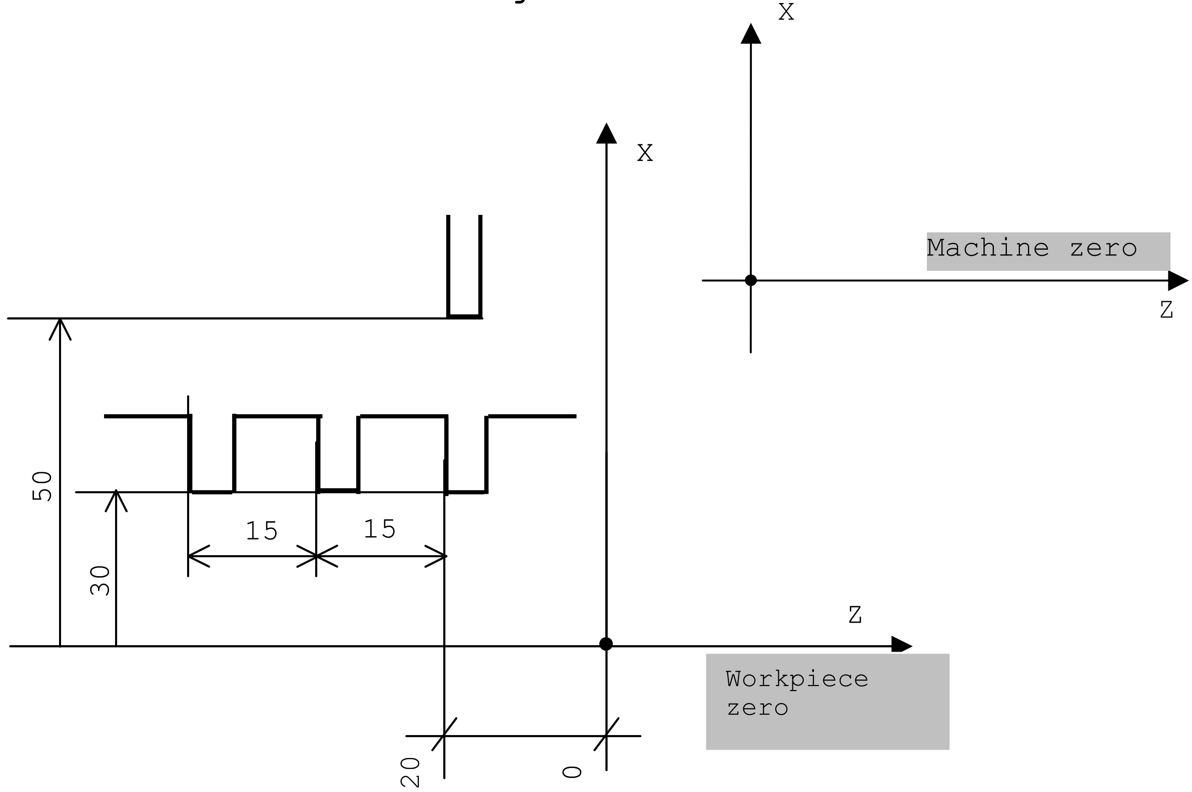
To establish the desired reference system program:

- G90 movement referred to absolute zero point;
- G91 movement referred to the last position of the axes (incremental programming);
- G79 movement referred to machine zero (valid only in the block where it has been programmed).

The allowable format is:

{G90}
{G91} [OTHER G CODES] [OPERANDS]
{G79}

Fig. 5.20.




```
N12 TMR=2
N21 X20
N13 G0 X50 Z-20
N14 G1 G91 G94 X-20
N15 X20
N16 Z-15
N17 X-20
N18 X20
H19 Z-15
N20 X-20
N21 X20
N22 GO G90 X80
N23 G79 X0 Z0
```

Notes:

• If you program none of these functions, the control automatically defaults the absolute zero point.

• G90 and G91 are modal functions, whereas G79 is not. A programmed mode remains active until the control executes G79.

• Incremental programming is not compatible with the GTL ambient.

5.8.8. MODAL GROUP J: ATTRIBUTES OF THE DYNAMIC MODE (G04-G09)

The allowable options are:

- G04 dwell at end of block;
- G09 deceleration at end of block.

The programming blocks have the following format:

{G04}
{G09} [OTHER G CODES] [OPERANDS]

- The duration of the dwell at item end is specified in the assignment block with TMR (in turns or seconds, with G95/G94).
The dwell is only executed in the G04 block.

- G09 sets feedrate = 0 at the end of the item where it has been programmed, but does not vary the current execution mode (either end of profile or point-to-point).
It is only valid in the block in which it is programmed.

5.9. ASSIGNMENT BLOCKS OF GLOBAL SYSTEM VARIABLES

According to the type of output variable, assignment blocks are grouped into three classes:

- assignment blocks of calculation variables (see Parametric Programming);
- assignment blocks of geometric variables (see High Level Geometric Programming);
- assignment blocks of global system variables.

This section deals with the assignment blocks of global system variables normally used from the program. For a complete list of the assignment codes from the program or the keyboard, refer to table 4.5.

Global system variables, which can receive a value from the program, typically define the parameters used for a machining cycle.

This class includes the following blocks:

TMR - dwell time definition
UOV - stock allowance definition
RMS - variation of the tap return speed
SSL - spindle speed limit
SA,SK - definition of buffer structures
SYVAR - definition of a group of variables
TIM - system time definition
TOT - time totalization definition

DWELL TIME DEFINITION - TMR

Allows you to assign a dwell time at block end, enabled in the blocks with G04 and/or fixed cycles.

The format is:

TMR = VALUE

where:

VALUE can be programmed in explicit and/or implicit mode (E parameter of LR format)

For example:	TMR = 12.5	assign a dwell time of 12.5 s
	E32 = 13.4	
	TMR = E32	assign a dwell time of 13.4 s

Note. TMR can be present in any part of the program.

STOCK ALLOWANCE DEFINITION - UOV

Defines the value of the stock allowance to be left along the profile. It is used in roughing and pre-finishing cycles.

The format is:

UOV = VALUE

where:

VALUE can be programmed in explicit or implicit mode (E parameter with LR format), and is expressed in the same measuring units as dimensions.

For example:	UOV = 0.5	assign a 0.5 stock allowance
	E30 = 1.5	
	UOV = E30	assign a 1.5 stock allowance

Note. UOV is normally programmed, but can also be used in the assignment blocks from the keyboard.

VARIATION OF THE TAPPING RETURN SPEED - RMS

Defines the speed variation percentage in the return phase of the transducer tapping cycle.

The format is:

RMS = VALUE

where:

VALUE can be a constant or an E variable of BY format.

For example:	RMS = 110	(+10% of the programmed F)
	RMS = 10	(-90% of the programmed F)

Note. RMS is normally programmed, but can also be used in the blocks from the keyboard.

SPINDLE SPEED LIMIT - SSL

SSL allows you to define the spindle speed limit. It is mandatory if the control has been set to constant surface speed (G96).

The allowable format is:

SSL = VALUE

where:

VALUE can be programmed either as an E parameter or a LR format.

Example:

SSL = 200	establish a spindle speed of 2000 rpm
E32 = 1500	
SSL = E32	establish a spindle speed of 1500 rpm

DEFINITION OF THE A BUFFER AND K BUFFER STRUCTURES - SA, SK

The assignment functions of I/O variables towards the interface logic make it possible to transfer a value directly from the program to the data structures for the communication between application software and interface logic.

To this purpose two data structures are available in the system 8600 MC: the A buffer and the K buffer.

The A buffer defines all the physical (electrical) signals of ON/OFF type, connecting the CU (Control Unit) to the MT (Machine Tool). It is a structure of 1024 bytes, allocated in the memory at the physical address 1000:0.

The K buffer defines all the communication variables between application software and interface logic to the MT. It is allocated in a memory address defined on system initialisation, which can be found in the "System Address Table".

For further information about the functions of the A buffer and K buffer structures, refer to the description of the data area 2 "Siprom system".

The allowable format is:

SA [index] = VALUE

where:

[index]	a value defining the assignment variable. Its range depends on the format.
VALUE	a constant or a symbolic variable or a sequence of characters.

Examples:

SA12=SK	assign to bit No.12 of the A buffer the value contained in the first bit of the K buffer
SK5=SK7	assign to byte No.5 of the K buffer the byte No.7 of the same structure
SA128=1	set the signal (bit) No.128 of the A buffer to ON
SK7="RIF"	write sentence RIF starting from byte No.7 of the K buffer
SA3.BY=255	assign value 255 to the byte No.3 of the A buffer

DEFINITION OF A GROUP OF VARIABLES - SYVAR

SYVAR defines a group of variables at the programmer's disposal.

The allowable types associated to the single variable are all those provided for the symbolic variables of the language NC-110.

The format is:

SYVAR [index] [format] = VALUE

where:

[index]	a value defining the variable to which a value has to be transferred. Its range depends on the format. If not specified, the byte format is assumed by default.
VALUE	can be a constant, an E parameter, a system variable or a string of characters, as long as it is compatible which the access format of he variable.

Examples:

```
SYVAR = E4
SYVAR1 = E3+E4
SYVAR.IN = 268
E4 = SYVAR
E35 = SYVAR2.LR
SYVAR16.3CH = "ABC"
```

When characterising the system, you can have SYVAR variables reset at each power-off.

By default, you can define 200 SYVAR of byte format.

HOW TO ADDRESS GLOBAL SYSTEM VARIABLES SA-SK-SYVAR

In order to address global variables, you must consider two aspects:
- the programming index of the variable;
- the format with which you want to access the variable.

The allowable formats are:

FORMAT	RANGE
B L = 1/8 bytes	0/1
BY = 1 bytes	0÷255
IN = 2 bytes	-32768 ÷ +32767
LI = 4 bytes	-2.147.483.647 ÷ +2.147.483.647
RE = 4 bytes	±7 significant digits (whole or decimal)
LR = 8 bytes	±16 significant digits (whole or decimal) ±13 whole digits
CH = 1 byte per character	

Notes:

• If the variable has a CH format, the elements preceding the format indicate the number of characters addressed by the variable. The default value is 1. The upper limit is 32. For example: SYVAR 1.4CH addresses 4 characters starting from SYVAR1.

In version K14 may be programmed ISO blocks using the variables SYVAR.
For example:
SYVAR0.12CH="G1X30Z20F100" - addresses 12 characters starting from SYVAR0.
SYVAR12.8CH="G0X0C0Z0" - addresses 8 characters starting from SYVAR12.
@ SYVAR0.12CH - runs ISO blocks G1X30Z20F100
@ SYVAR12.CH - runs ISO blocks G0X0C0Z0

• When programming the index and format of a variable, you must take into account the size of the preceding variable, in order to avoid overlapping.

Fig. 5.21. - Address table for SYVAR/SA/SK variables

<table>
<tr><td>S0.BL
S7.BL</td><td>S0.BY</td><td>S0.CH</td><td rowspan="2">S0.IN</td><td rowspan="4">S0.LI</td><td rowspan="4">S0.RE</td><td rowspan="8">S0.LR</td></tr>
<tr><td rowspan="14"></td><td>S1.BY</td><td>S1.CH</td></tr>
<tr><td>S2.BY</td><td>S2.CH</td><td rowspan="2">S1.IN</td></tr>
<tr><td>S3.BY</td><td>S3.CH</td></tr>
<tr><td>S4.BY</td><td>S4.CH</td><td rowspan="2">S2.IN</td><td rowspan="4">S1.LI</td><td rowspan="4">S1.RE</td></tr>
<tr><td>S5.BY</td><td>S5.CH</td></tr>
<tr><td>S6.BY</td><td>S6.CH</td><td rowspan="2">S3.IN</td></tr>
<tr><td>S7.BY</td><td>S7.CH</td></tr>
<tr><td>S8.BY</td><td>S8.CH</td><td rowspan="2">S4.IN</td><td rowspan="4">S2.LI</td><td rowspan="4">S2.RE</td><td rowspan="8">S1.LR</td></tr>
<tr><td>S9.BY</td><td>S9.CH</td></tr>
<tr><td>S10.BY</td><td>S19.CH</td><td rowspan="2">S5.IN</td></tr>
<tr><td>S11.BY</td><td>S11.CH</td></tr>
<tr><td>S12.BY</td><td>S12.CH</td><td rowspan="2">S6.IN</td><td rowspan="4">S3.LI</td><td rowspan="4">S3.RE</td></tr>
<tr><td>S13.BY</td><td>S13.CH</td></tr>
<tr><td>S14.BY</td><td>S14.CH</td><td rowspan="2">S7.IN</td></tr>
<tr><td>S120.BL
S127.BL</td><td>S15.BY</td><td>S15.CH</td></tr>
<tr><td>S128.BL</td><td></td><td></td><td></td><td></td><td></td><td>S2.LR</td></tr>
</table>

Note. S Stands for SYVAR/SA/SK variables.

Here's an example of how to address a SYVAR variable starting from SYVARO.BY:

SYVARO.BY
SYVAR1.CH
SYVAR1.IN
SYVAR1.LI
SYVAR1.RE
SYVAR2.RE
SYVAR2.LR

SYSTEM TIME DEFINITION - TIM

Defines a group of variables used by the programmer to strobe, in certain machining phases, the hour indicated by the system timer. The group covers seven timers (from 0 to 6). The measuring unit is the second.

The first timer (TIMO) is reserved. It receives a value not by means of a simple assignment, but by inputting a three-letter code (PTM) via keyboard.

The format is:

TIM [index] = VALUE

where:

VALUE can be a constant or an E parameter (E25 to E29).

Notes:

• The assignments to TIM variables can only be programmed in a program block.

• The variable TIM must be programmed in G00 mode with synchronisation (TIM preceded by #).

Example:

```
..........
N9  GO ...
N10 TIM1=TIMO           - assign the system time (clock) to TIM1
...........
...........
N89 GO  ...
N90 TIM2=TIMO-TIM1      - calculate the machining time from block
..........                N10 to block N90
..........
(DIS,TIM2)              - display the calculated time on line 2
```

TIME TOTALIZATION DEFINITION - TOT

Defines a group of variables used by the programmer to totalize partial times of machining cycles obtained in certain points of the program (difference between TOTO and the TIM detected at machining cycle start).

By using the same number as index of TIM and TOT, the programmer can dispose of 6 variables to measure times and 6 to totalize partial times.

The only allowable format for variable dimensions is the "real" format (RE):

TOT [index] = VALUE

where:

VALUE can be a constant or an E parameter (E25 to E29).

Note. The assignments to TOT variables can only be programmed in a program block.

Example:

```
N10  E31=75                     - number of parts
N2O# TIM1=TIMO
N30  T2.2 M6
N40  XZ S2000 F500 M3
..........
..........
N200 TOTO=TIMO-TIM1
N210 TOT1=TOTO*E1
N220 TOT2=TOT1/3600
N230  (DIS.TOT2)                - display the time required to execute
..........                        a batch of 75 parts (in hours)
```

5.10. HIGH LEVEL GEOMETRIC PROGRAMMING (GTL)

With the NC-110 you can program a geometric profile using the standard programming language (G1-G2-G3) or a high level programming language, GTL.

GTL allows you to program a profile made of straight lines and circles from the information on the part drawing. The control automatically calculates intersections and tangency points between geometric elements.

GTL and the standard programming language can be used in the same program but not in the same profile. GTL only works with absolute mode (G90).

ORIENTED GEOMETRY

The definition of a profile with GTL is based on four types of "geometric elements":

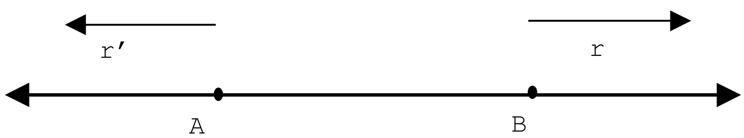
- reference origins;
- points;
- straight lines;
- circles.

Geometric elements and their directions define a particular type of geometry in the GTL language: oriented geometry.

Oriented geometry requires parameters that identify the position of an element in the active plane and the assignment of a direction of motion.

For example, in the below figure, consider a straight line r. Dassing through points A and 8 and running from A to B. Or, consider a straight line r', lying on r, but running from B to A.

Fig. 5.22. - Straight lines r and r' with orientation

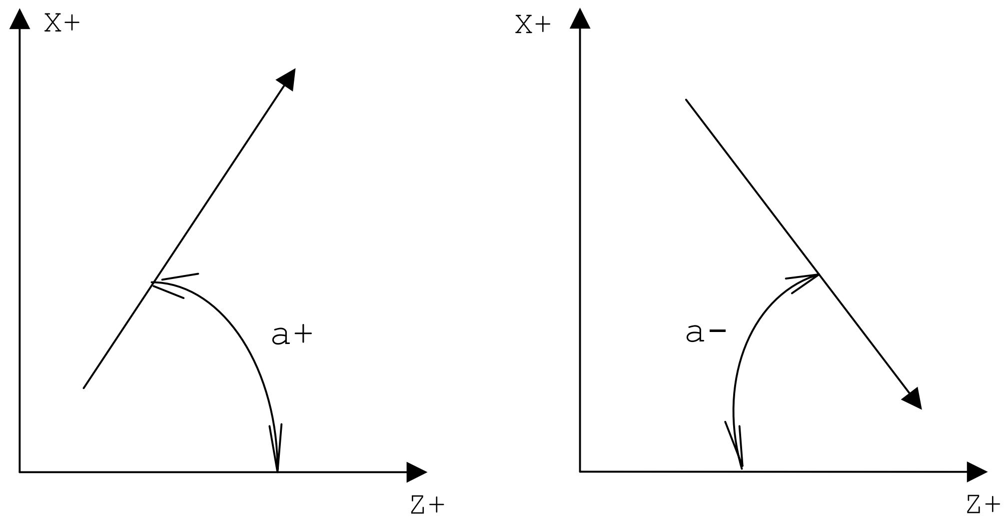


In oriented geometry, r and r' are two different straight lines that have opposite directions.

With GTL and oriented geometry, you must assign a direction of motion to each straight line. By convention, the direction of a straight line is the angle it forms with the positive z axis.

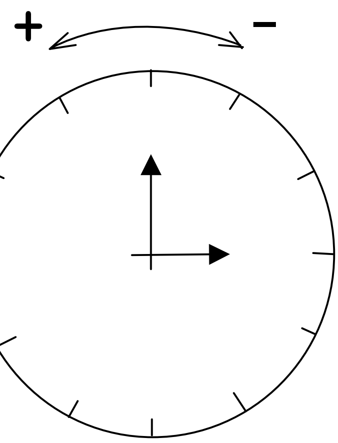
To determine the direction of a straight line, rotate the positive z axis until it coincides with the straight line.
The direction will be positive if you rotated CCW, or negative if you rotated CW.

Fig. 5.23. - Angles convention



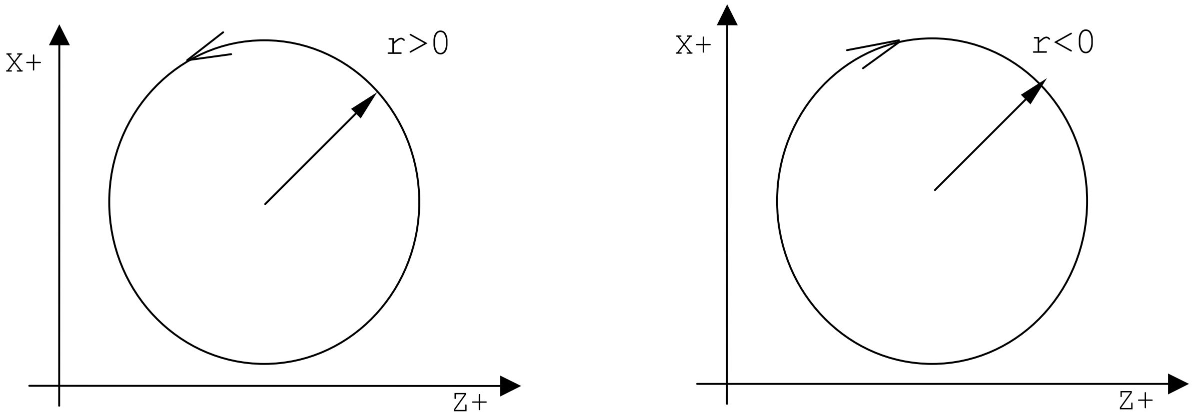
Circles must also have directions associated with them. The direction of a circle is positive if movement is CCW, or negative if movement is CW.

Fig. 5.24. - CW and CCW directions for circles



You assign a positive radius value to circles that are CCW, and a negative radius value to circles that are CW.

Fig. 5.25. - Circle radius and direction of motion



The direction you assign to an element should usually correspond to the direction of movement on the profile. However, when you define the profile it is possible to change the direction of an element if it runs counter to the direction of the rest of the profile.

STORING GEOMETRIC ITEMS

GTL language requires lower case characters (a-l-c-d-m-o-r-p-s) to define angles, circles, distances, radius vectors, reference origins, radius, points and
intersection numbers.

Lower case characters are needed because the control uses the same, upper case, characters for other functions.

To enter a lower case character, first press the **LOWERCASE** key, then press the character key from the keyboard.

You must store geometric elements before you define the profile.

GTL deals with: straight lines, circles, points, reference origins. You can think of them as geometric variables. They are identified by a NAME and an INDEX defined in an assignment block.

The format of the block is:

NAME INDEX = <expression>

where:

NAME — is one of the four symbolic names provided for geometric elements:
o - reference origins;
p – points;
l - straight lines;
c – circles.

INDEX — defines the number of the geometric variable NAME (element). It is a number between 0 and 255 inclusive. The max. limit is defined during system configuration.

Expression — contains all the information necessary to describe the geometric element.

You can define elements in:

- direct (explicit) mode, i.e., program in a block all the information required to identify the geometrical element in the block;

- indirect (implicit) mode, i.e. referencing in the block other geometric elements that were previously defined.

Here are some examples for storing elements:

```
o1 = Z30 Y30 a45
p1 = o1  Z15 X15
p2 = Z60 X10
l1 = p1, p2
l2 = Z30 X50, a45
c1 = l1, l2, r15
l3 = ZO XO, Z100 X60
p3 = l3, c1
c2 = p3, r8
..........
```

The number of geometric elements you can store is defined during system configuration.

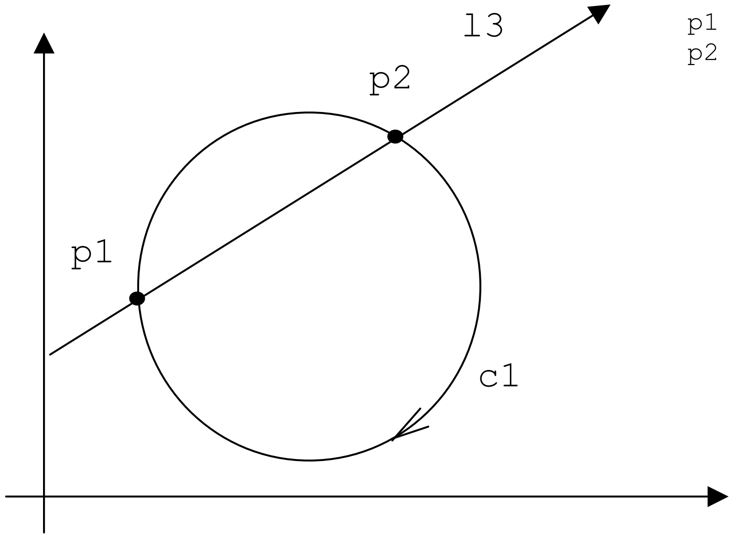
The format of geometric definitions requires a "," (comma) as element separator (straight lines, points, circles) from the next item or the next information (such as radius, r, or angle, a).

Examples:

```
p1  = 230 X30                the separator
c1  = I10 J20 r30            is not required


L1 = Z20 X20, Z100 X-10
        |          |
      point      point


L2 = I30 J2O r10, Z80 X80
          |           |
       circle      point


L3 = Z100 X100, a45
         |       |
      point   angle


p5 = c1, c2
       |   |
       | circle
    circle


c3 =  L1,  L2,  r18
       |    |    |
       |    |  radius
       |   line
     line
```

In figure 5.26., s2 selects the second intersection.

Figure 5.26.



p1 = c1, l3
p2 = c1, l3, s2

The allowed formats for defining geometric elements are shown in Table 5.6.

Important. A sequence of two points (..) indicates that you must declare numerical values. The elements in square brackets [] are optional and can be omitted.

Table 5.6.- GTL DEFINITIONS

Reference origins	on = Z.. X.. a..
Points	pn = [om] Z.. X.. pn = [om] m.. a.. pn = lm, lp pn = [-]lm, cp [,s2] pn = cm, [-]lp [,s2] pn = cm, cp [,s2]
Straight lines	ln = [om] Z.. X.., [op] Z.. X.. ln = [om] Z.. X.., a.. ln = [om] I.. J.. r.., [op] I.. J.. r.. ln = [om] I.. J.. r.., a.. ln = [om] I.. J.. r.., [op] Z.. X.. ln = [om] Z.. X.., [op] I.. J.. r.. ln = pm, pq ln = pm, a.. ln = [-]cm. [-]cp ln = [-]cm, a.. ln = [-]cp, pm ln = pm, [-]cp ln = [-]lm, d..
Circles	cn = [om] I.. J.. r.. cn = [om] m.. a.. r.. cn = [-]lm, [-]lp, r.. cn = [-]lm, [-]cp, r.. cn = [-]cp, [-]lm, r.. cn = pm, [-]lp, r.. cn = [-]lp, pm, r.. cn = [-]cm, [-]cp, r.. cn = pm, [-]cp, r.. cn = [-]cp, pm, r.. cn = pm, pq, r.. cn = pm, [-]lp cn = pm, [-]cp [,s2] cn = pm, pq, pr cn = pm, r.. cn = [-]cm. [-]d..

DEFINING REFERENCE ORIGINS

GTL makes it possible to define reference origins in direct (explicit) mode.

Usually, the information contained in the NC tape refers to one system of axes that coincides with the machine axes.
However, the part could have been designed using various cartesian systems, i.e., the absolute system, or other reference systems (origins) generated from absolute system by rotating and translating the axes.

GTL geometry can be defined with any reference system.

Use the direct format only:

on = Z.. X.. a..

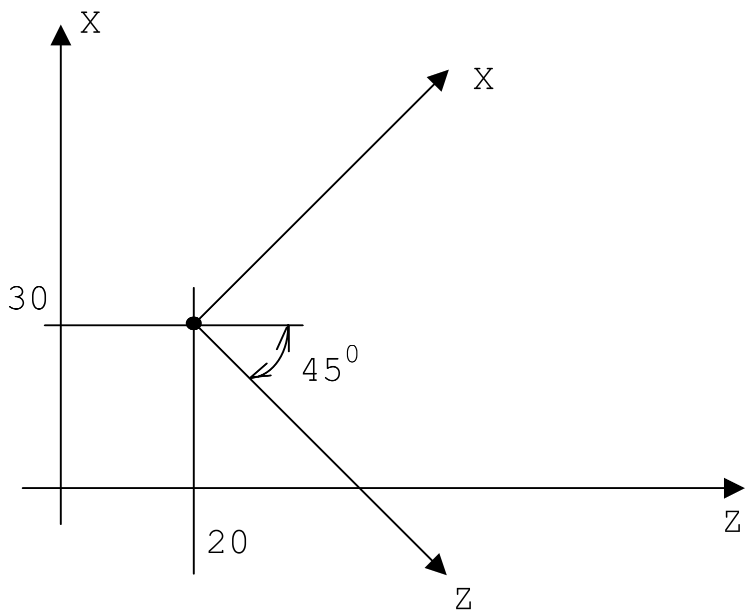
where:

on	identifies the name and index of the reference origin
Z..X..	coordinates of the new origin
a..	rotation angle (positive CCW)

Example:

Fig. 5.27.

o4 = Z20 X30 a-45

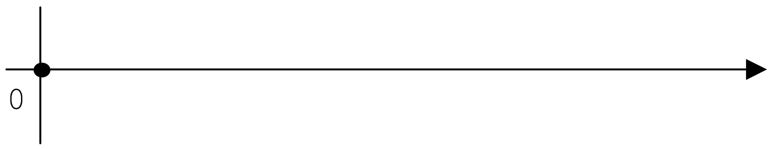


DEFINING POINTS

GTL makes it possible to define the points in direct (explicit) mode or in indirect (implicit) mode. A point can be defined with either cartesian or polar coordinates.

The polar reference system consists of an origin called pole that originates the X axis, which is called polar axis (Figure 5.28.).

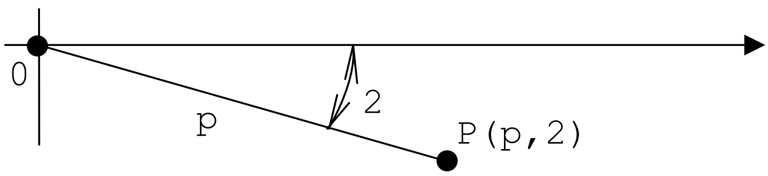
Fig. 5.28. - Polar axis



Any point of the axis can be identified by:

- a segment running from the pole to the point (called radius vector);
- an angle formed by the radius vector with the polar axis (Figure 5.29.).

Fig. 5.29. - Polar coordinates



DIRECT PROGRAMMING FORMATS FOR POINTS

Point with cartesian coordinates (figure 5.30.)
pn = [on] Z.. X..

Point with polar coordinates (figure 5.34.)
pn = [on] m.. a..

INDIRECT PROGRAMMING FORMATS FOR POINTS

Intersection point of two predefined lines
pn = lm,lp

Intersection point of a predefined line and a predefined circle
pn = [-]lm,cp[,s2]
pn = cm,[-]lp[,s2]

Intersection point of two circles
pn = cm,cp[,s2]

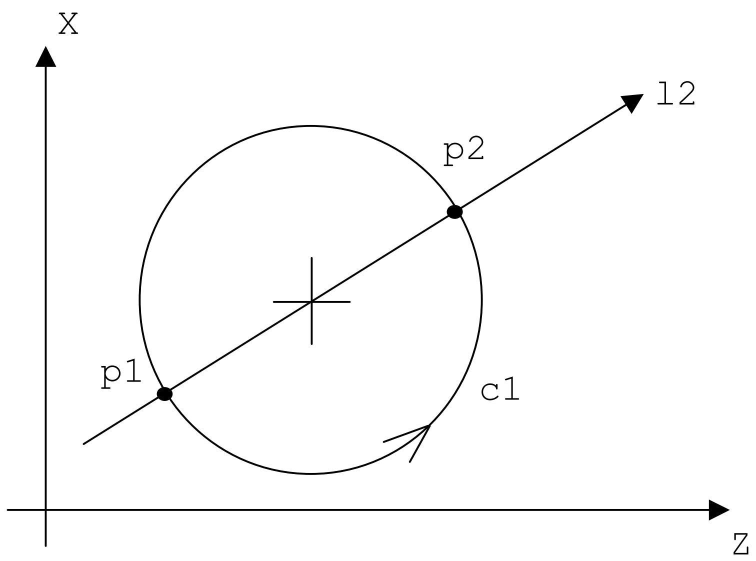
where:

pn	identifies the name of the point having index n (n is a number between 1 and the maximum configured number)
Z.. X..	point coordinates
[om]	reference origin previously defined having index n, to which the Z and X coordinates are referred
m..	radius vector of the polar coordinate system
a..	angle between the radius vector and the polar axis
cm cp	predefined circle elements having index m and p
[-]lm [-]lp	predefined line elements of having index m and p The direction can be changed by introducing a "-" sign
[,s2]	indicates the second intersection

Important. For line-circle intersections, two solutions are possible (figure 5.30.): circle c1 and line l2 intersect at points p1 and p2. By travelling in the specified direction of straight line l2, you come to the point p1 (1st intersection) first, and then to the point p2 (2nd intersection).

To select the second intersection (p2), you must declare the indicator, s2. If you omit s2, the control will select the first intersection (p1).

Fig. 5.30. - Straight line-circle intersection



Two solutions are also possible for circle-circle intersections: the circles c1 and c2 intersect at points p1 and p2 (Fig. 5.31.). Consider the oriented line that joins the centre of the 1st circle with the centre of the 2nd circle. This lines divides the plane into two half planes. To select the point in the right half plane (p2), use the s2 indicator. If you omit s2, the control automatically selects the point in the left half plane (p1).

Figure 5.31. - Circle-circle intersection

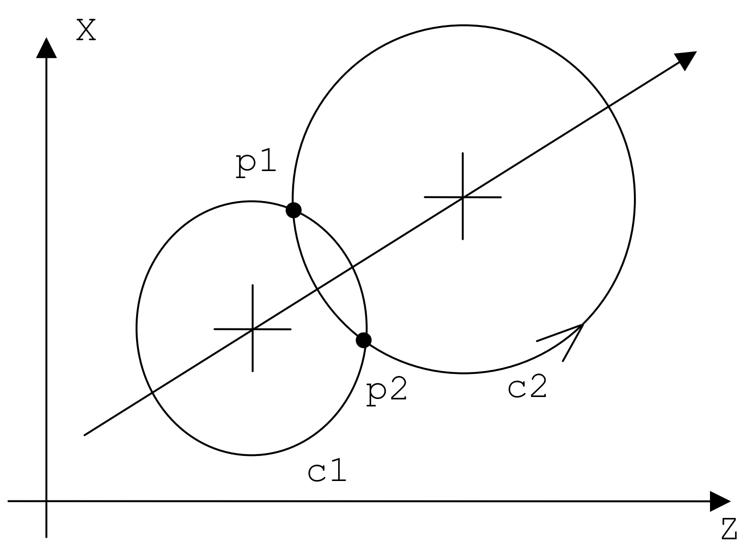
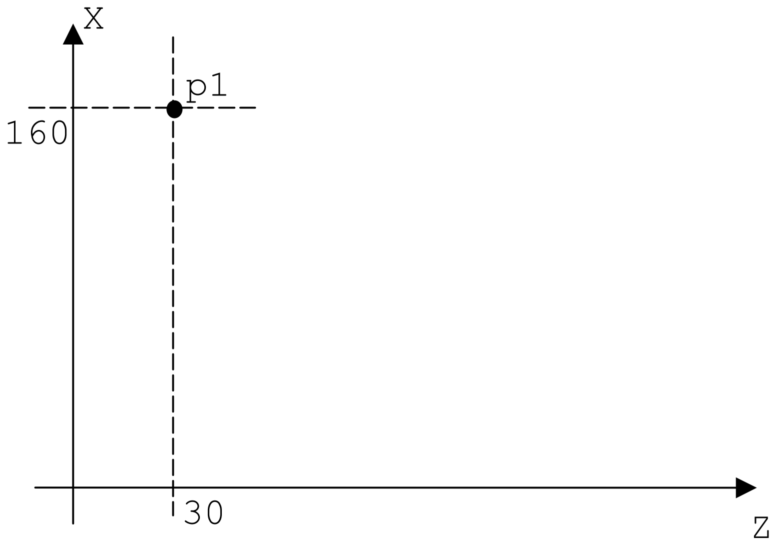
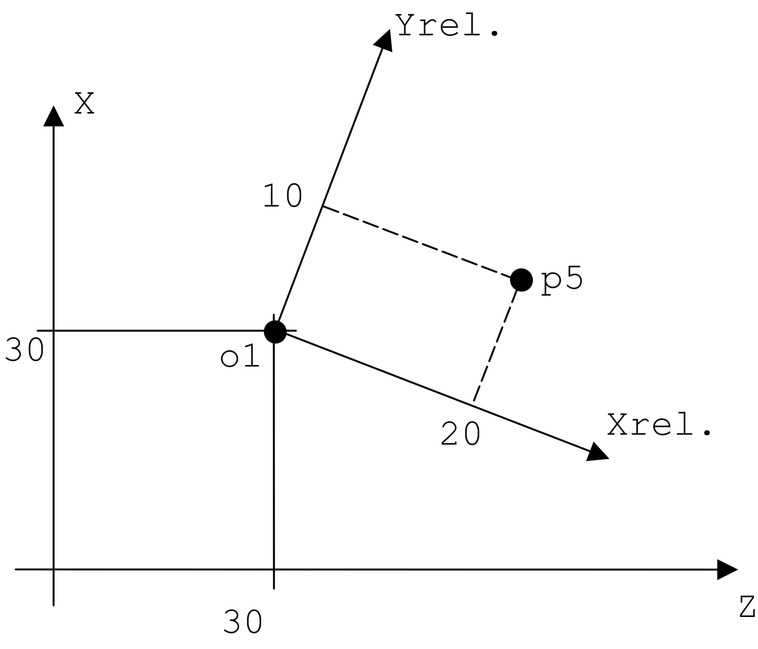


Fig. 5.32.



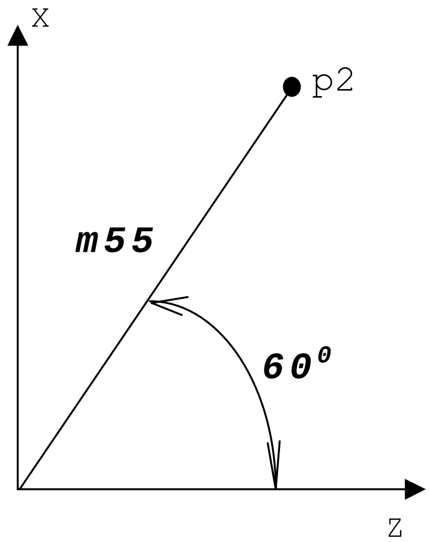
p1 = X30 Y160

Fig. 5.33.



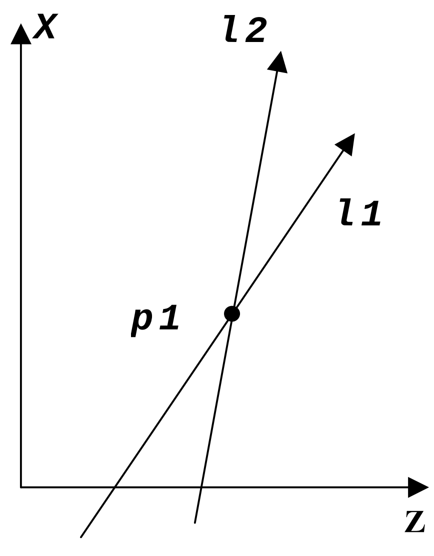
o1 = Z30 X30 a-20
p5 = o1 Z20 X10

Fig. 5.34.



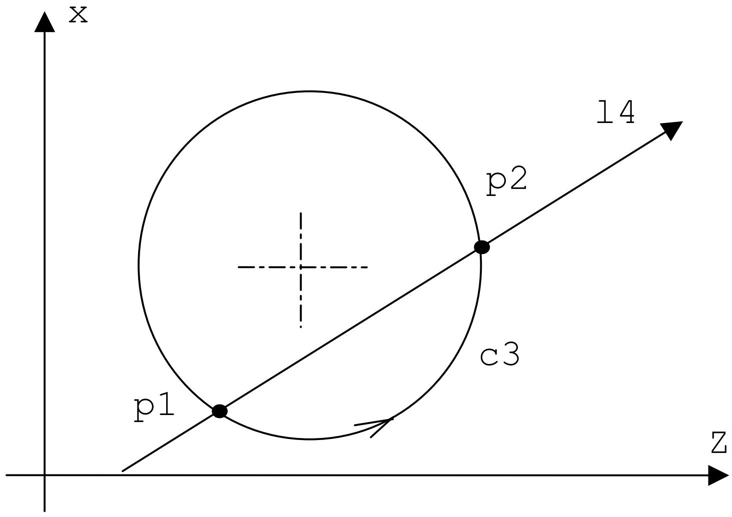
p2 = m55 a60

Fig. 5.35.



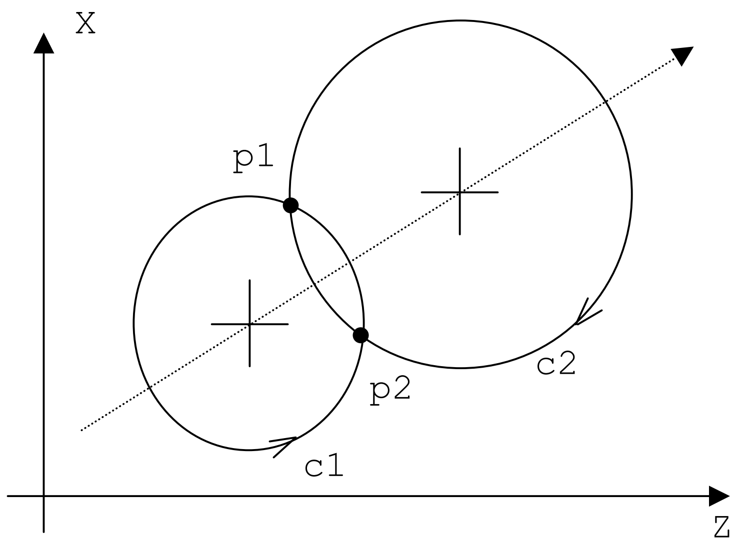
p1 = l1, l2

Fig. 5.36.




```
p1 = l4, c3
p2 = l4, c3, s2
p1 = -l4, c3, s2
```

Fig. 5.37.




```
p1 = c1, c2
p2 = c1, c2, s2
p1 = c2, c1, s2
```

DEFINING STRAIGHT LINES

GTL allows you to define straight lines in direct (explicit) or indirect (implicit) mode.

The direction of the straight line is always from the first to the second element you define.

Two solutions are possible for a straight line tangent to one circle. The straight line can be tangent to one side of the circle or to the other side. GTL will select the tangency point at which the circle and the straight line have the same direction.

Fig. 5.38. - Direction Incompatibility of Geometric Elements

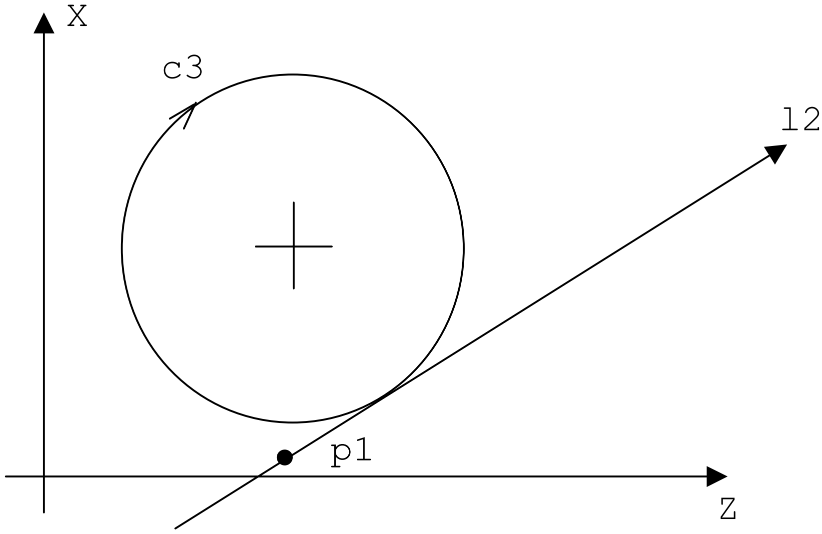
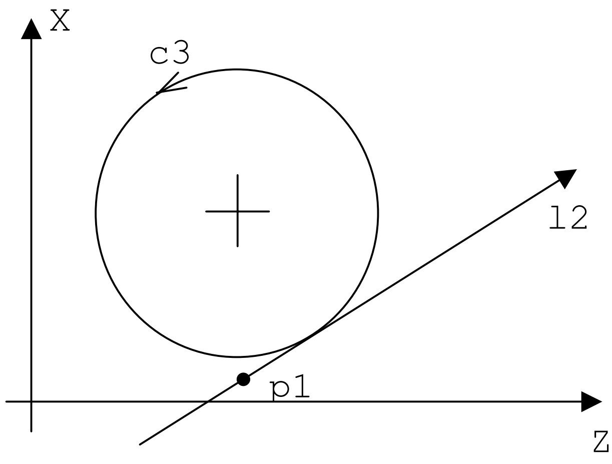


Fig. 5.39. - Direction Compatibility of Geometric Elements



Direct Programming Format

Straight line through two points (fig. 5.40.)
ln = [om] Z.. X..,[op]Z.. X..

Straight line through one point and forming an angle with the abscissa axis (figs. 5.42., 5.43.)
ln = [om] Z.. X.., a..

Straight line tangent to one circle and forming an angle with the abscissa axis (figs. 5.44., 5.45.)
ln = [om] I.. J.. r.., a..

Straight line tangent tm two circles (fig. 5.46., 5.47.)
ln = [om] I.. J.. r..,[op] I.. J.. r..

Straight line tangent to one circle and through one point (fig. 5.41.)
ln = [om] I.. J.. r..,[op] Z.. X..
ln = [om] Z.. X..,[op] I.. J.. r..

Indirect Programming Format

Straight line through two points (fig. 5.48.)
ln = pm, pq

Straight line through one point and forming an angle with the abscissa axis (fig. 5.42.)
ln = pm, a..

Straight line tangent to two circles (figs. 5.50., 5.51.)
ln = [-]cm,[-]cp

Straight line tangent to one circle and forming an angle with the abscissa axis (fig. 5.53.)
ln = [-]cm, a..

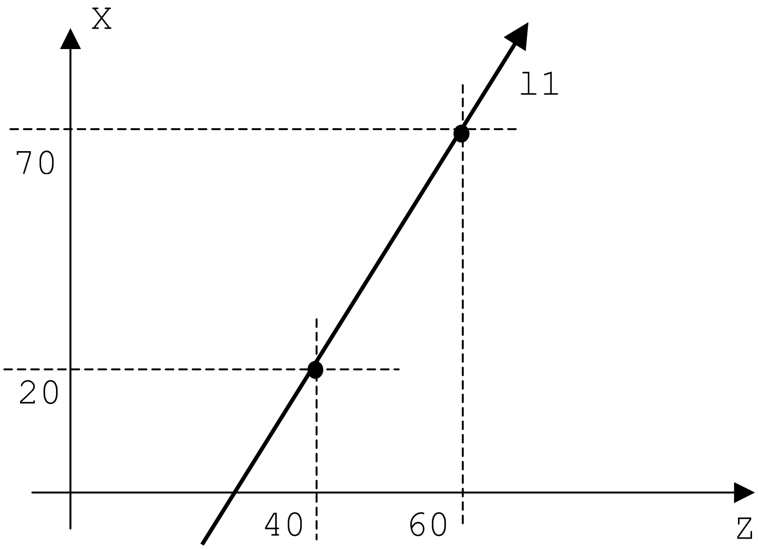
Straight line tangent to one circle and through one point (fig. 5.49.)
ln = [-]cp, pm
ln = pm,[-]cp

Straight line parallel to one straight line at d distance (figs. 5.54., 5.55.)
ln = [-]lm, d..

where:

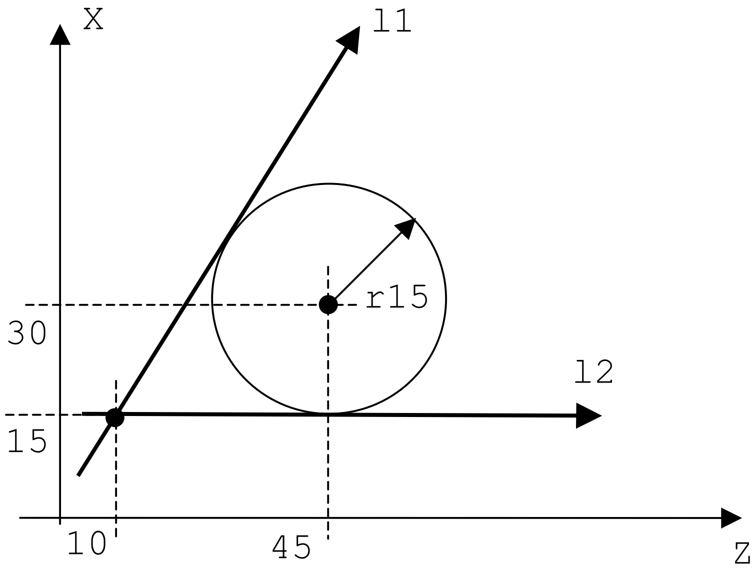
ln	identifies the name of the line of n index (n is a number between 1 and the max. configured number)
Z.. X..	point coordinates
a..	angle formed by abscissa axis and straight line (positive CCW)
r..	circle radius (positive CCW)
pm pq	predefined point elements having index m and q
[-]cm [-]cp	predefined circle elements having index m and p. You can change the circle direction by using a negative sign, in order to assure direction compatibility between a straight line and the circle in the tangency point
[-]lm	predefined straight line element
d..	distance between two straight lines, positive if the straight line is on the left, negative if it is on the right (looking in the direction of the predefined straight line).

Fig. 5.40.



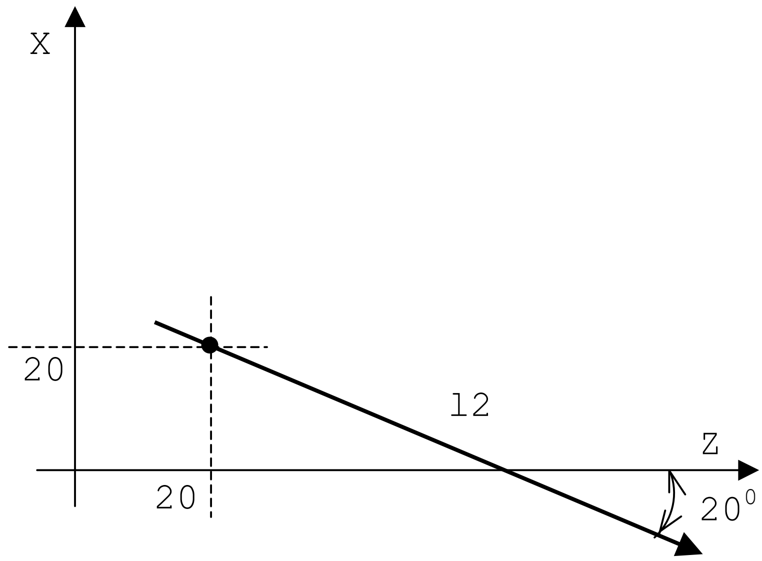

```
l1 = Z40 X20, Z60 X70
```

Fig. 5.41.



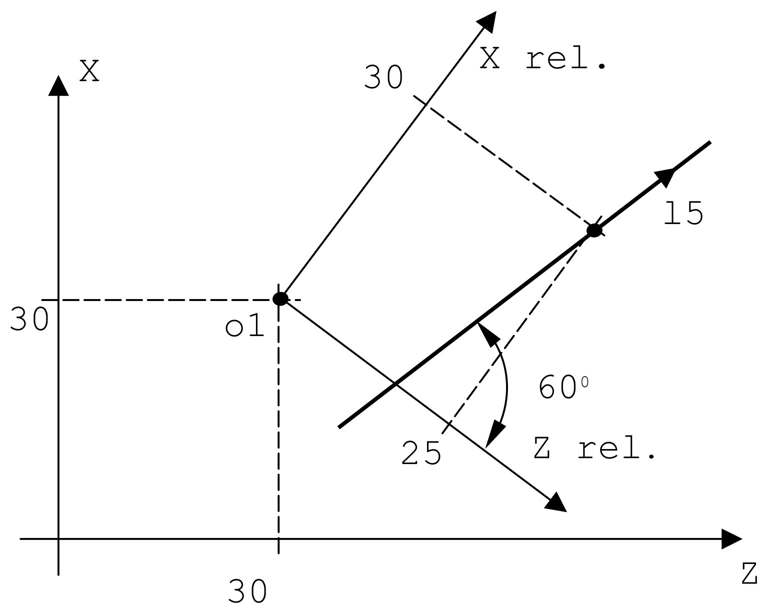

```
l1 = Z10 X15, I45 J30 r-15
l2 = Z10 X15, I45 J30 r15
```

Fig. 5.42.



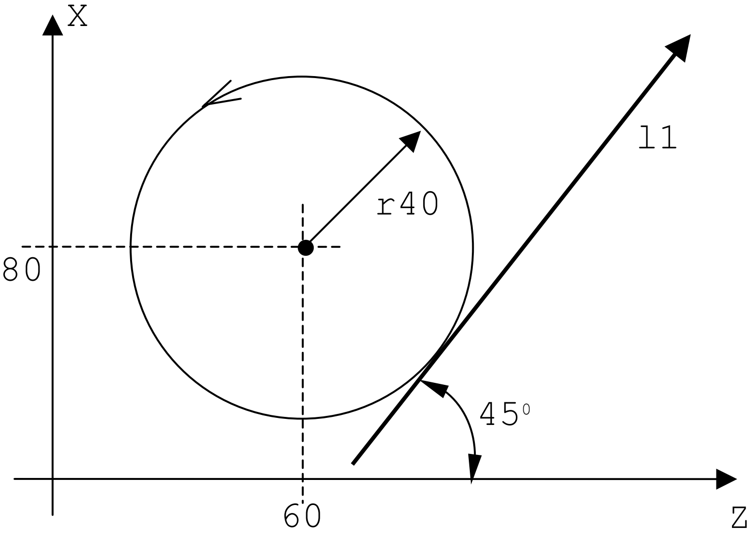
l2 = Z20 X20, a-20

Fig. 5.43.



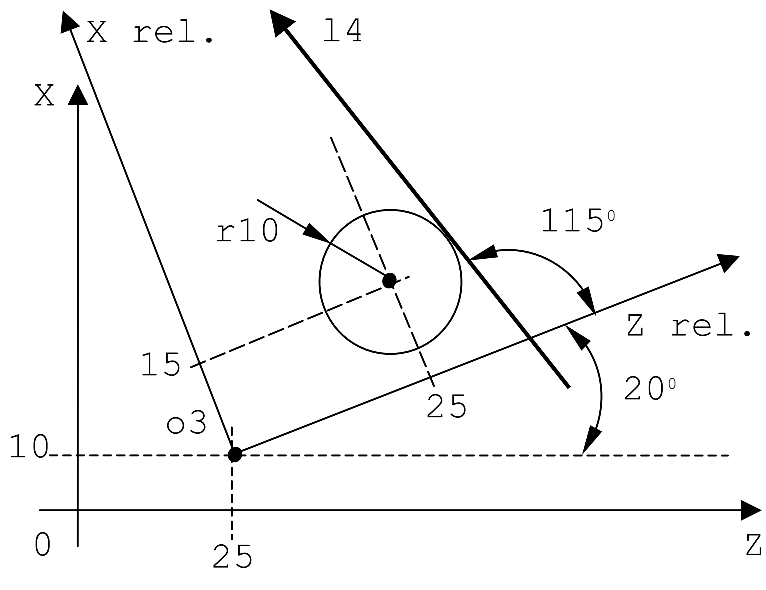
l5 = o1 Z25 X30, a60

Fig. 5.44.



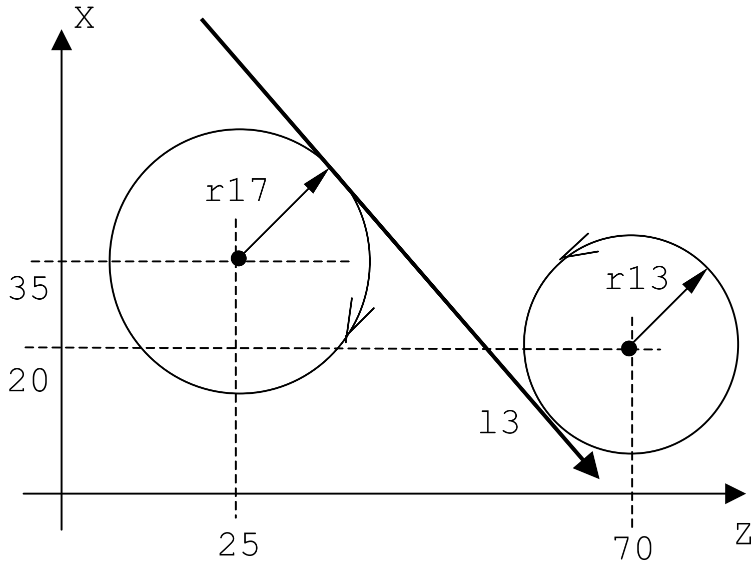
l1 = I60 J80 r40, a45

Fig. 5.45.



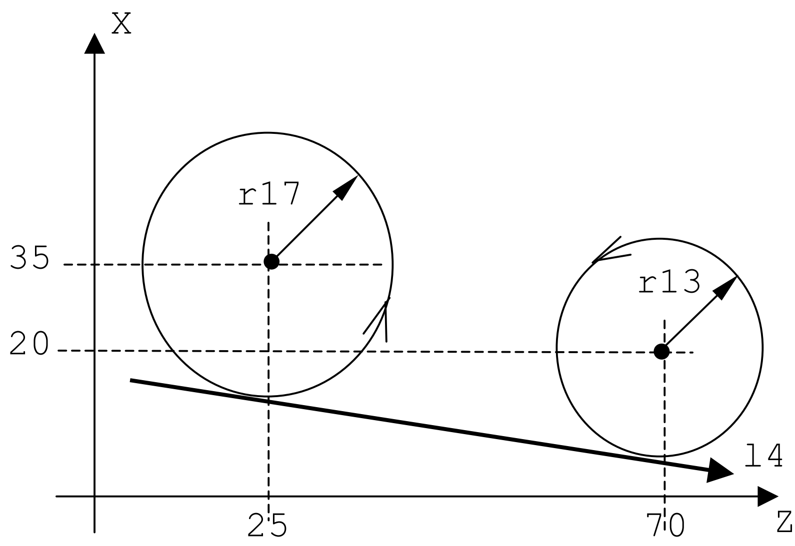
l4 = o3 I25 J15 r10, a115

Fig. 5.46.



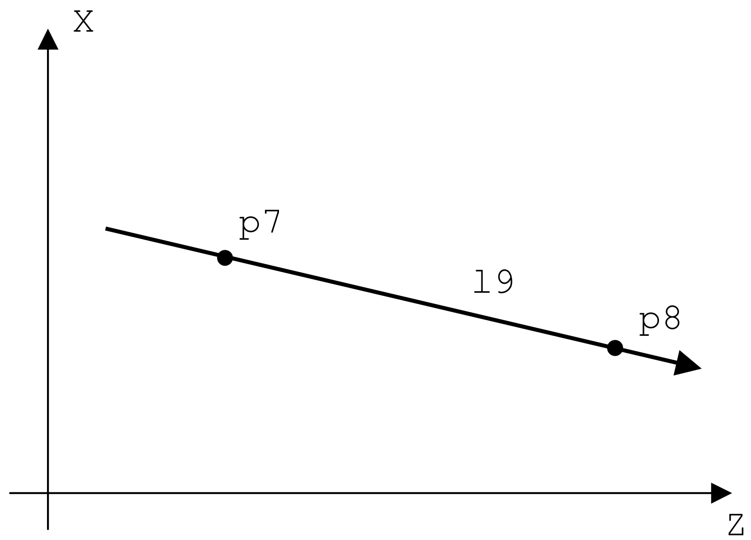
l3 = I25 J35 r-17, I70 J20 r13

Fig. 5.47.



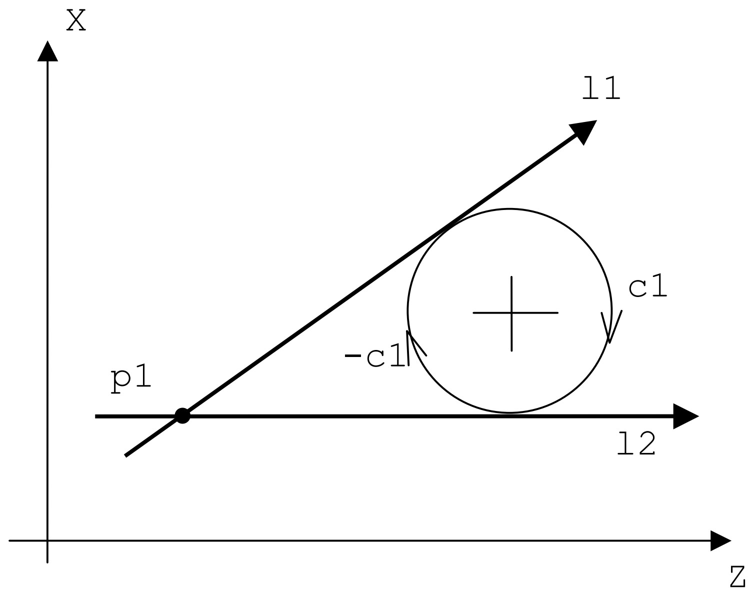
l4 = I25 J35 r17, I70 J20 r13

Fig. 5.48.



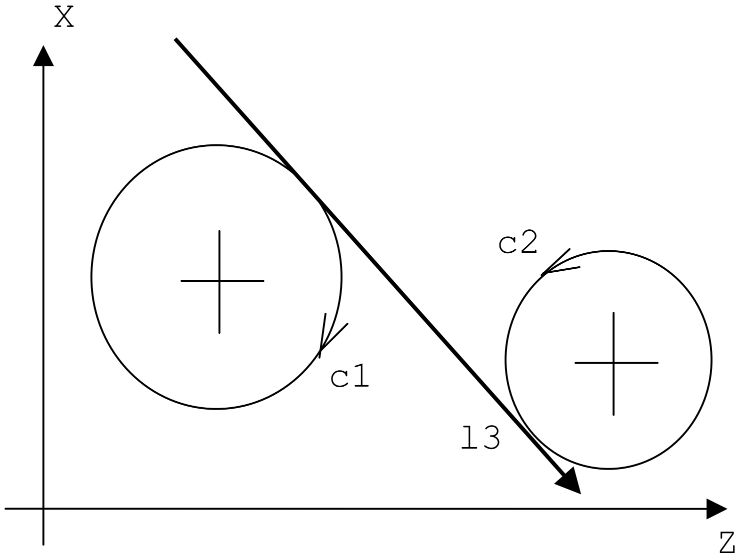
l9 = p7, p8

Fig. 5.49.



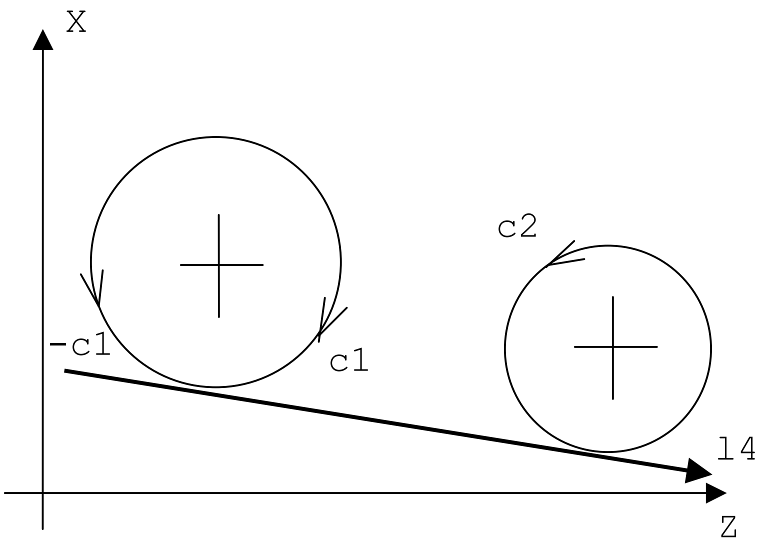
l1 = p1, c1
l2 = p1, -c1

Fig. 5.50.



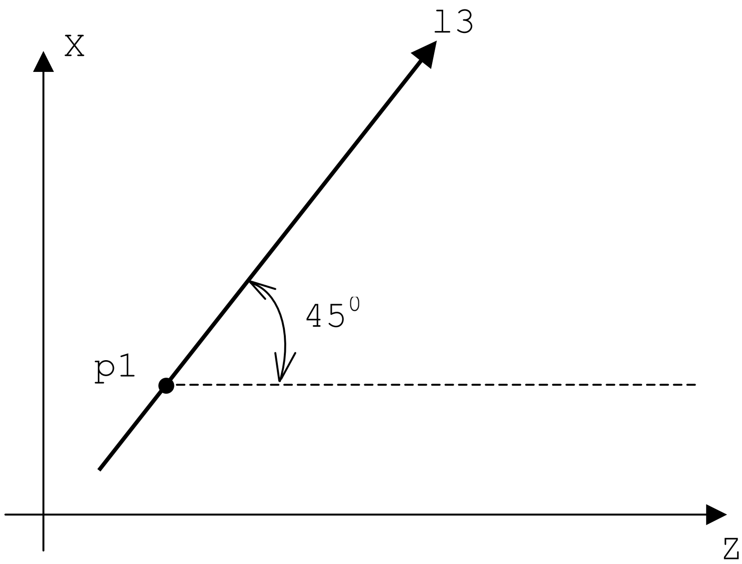
l3 = c1, c2

Fig. 5.51.



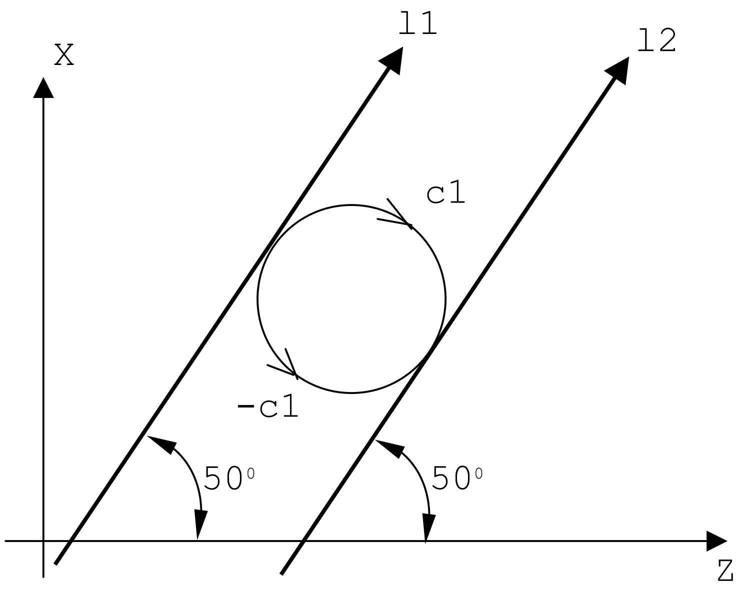
l4 = -c1, c2

Fig. 5.52.



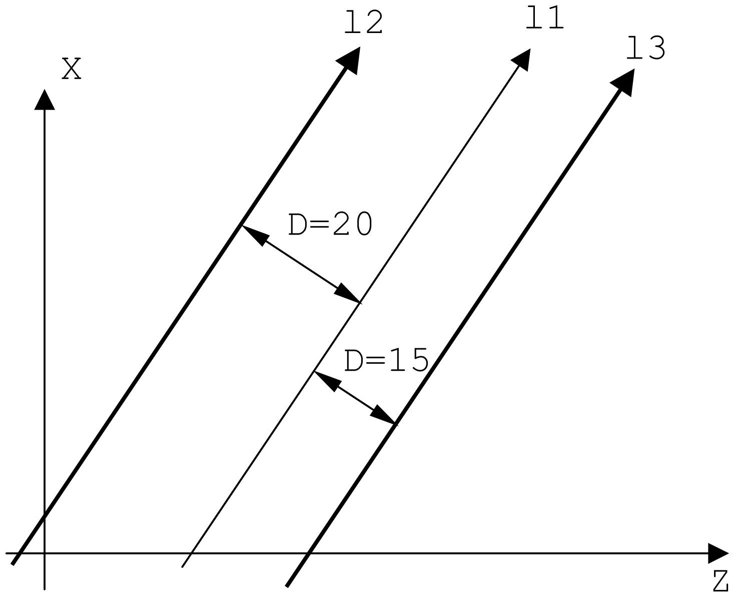
l3 = p1, a45

Fig. 5.53.



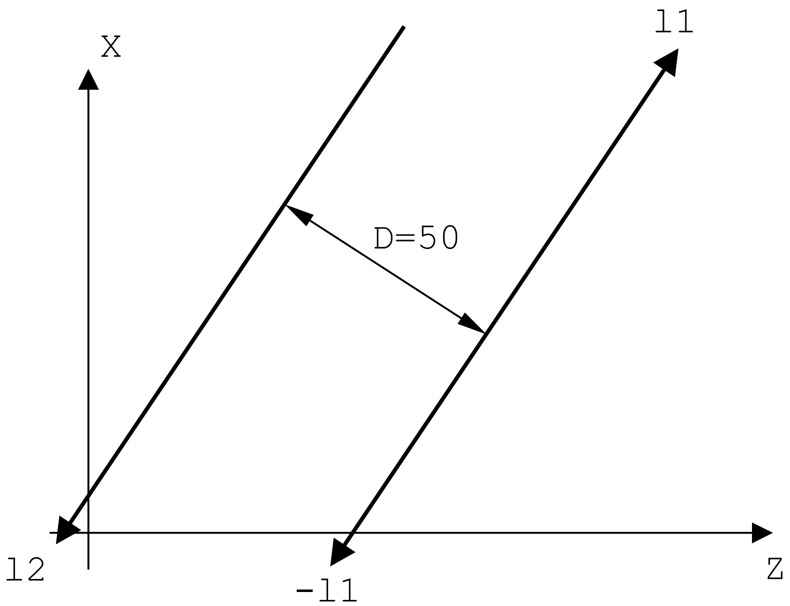
l1 = c1, a50
l2 = -c1, a50

Fig. 5.54.




```
l2 = l1, d20
l3 = l1, d-15
```

Fig. 5.55.




```
l2 = -l1, d-50
```

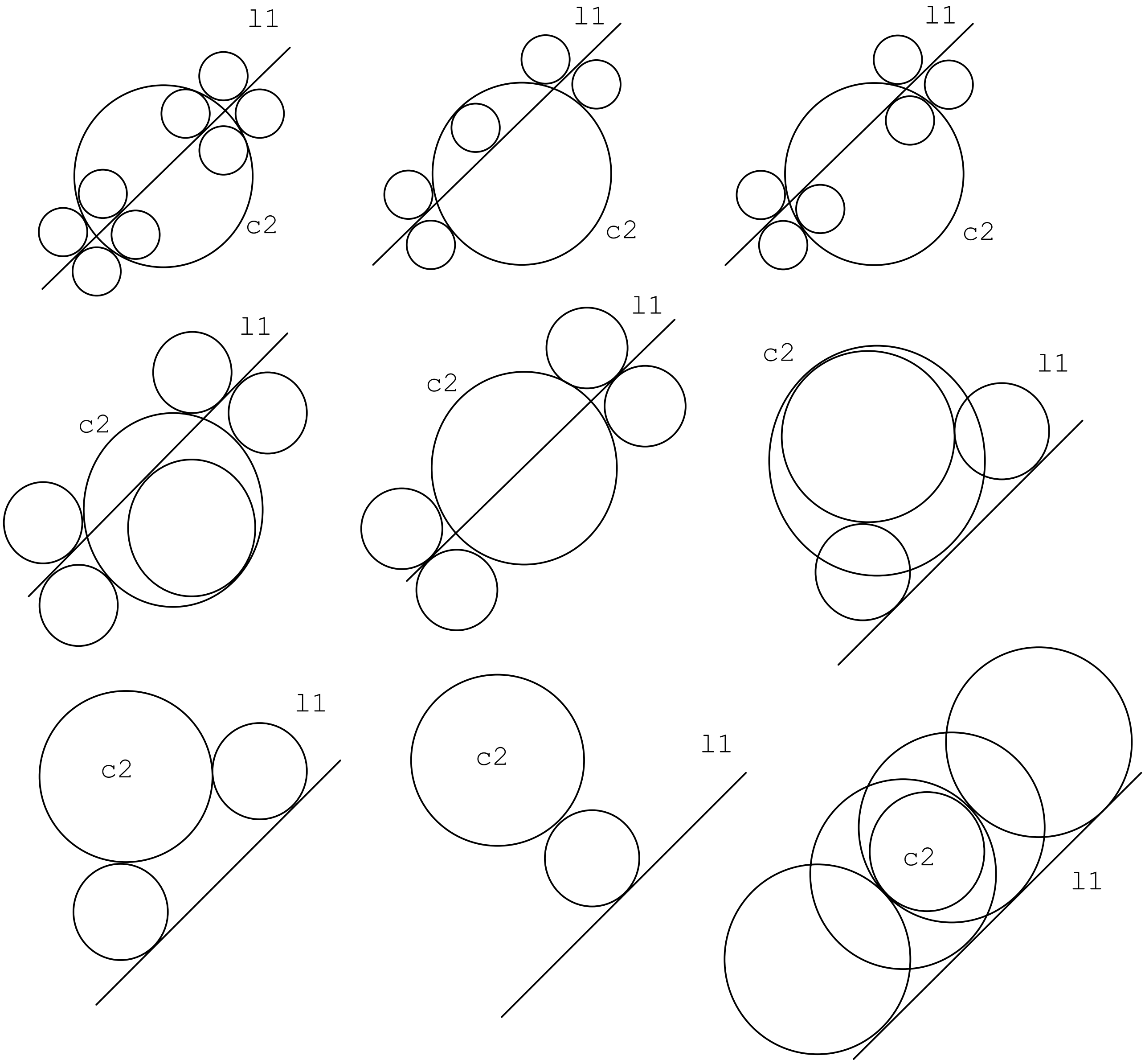
DEFINING CIRCLES

GTL also allows you to define circles in direct (explicit) or indirect (implicit) mode.

When defining circles in indirect mode, you should consider the direction compatibility of the elements (a "-" sign can change the direction of predefined circles).

If you do not account for the direction of the circle, a circle of known radius and a straight line can have 1 to 8 circles tangent to the straight line and the circle.

Fig. 5.56. - Circles Tangent to One Straight Line and One Circle



If you account for direction compatibility between a given circle and a line, there are only two possible tangent circles that have the same radius and direction.

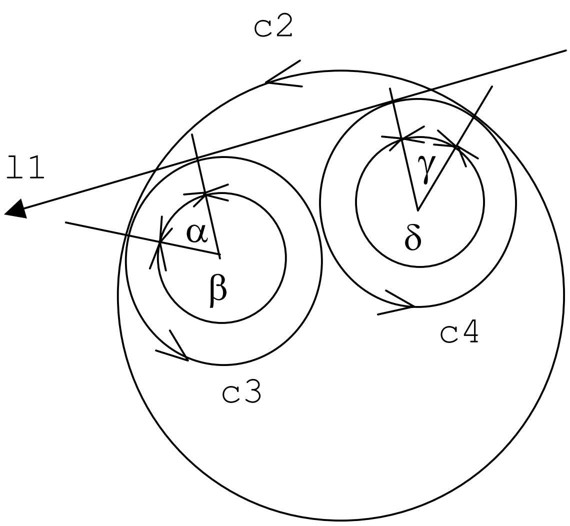
To distinguish between the two possible circles, consider:

– the order in which the circle and line are defined (line-circle or circle-line);

– the length of the central angles of the two possible tangent circles.

GTL always generates a circle with its direction going from the first to the second element (circle or line). The tangent circle will have a minor central angle.

Fig. 5.57. - Tangent Circles with Minor Central Angles

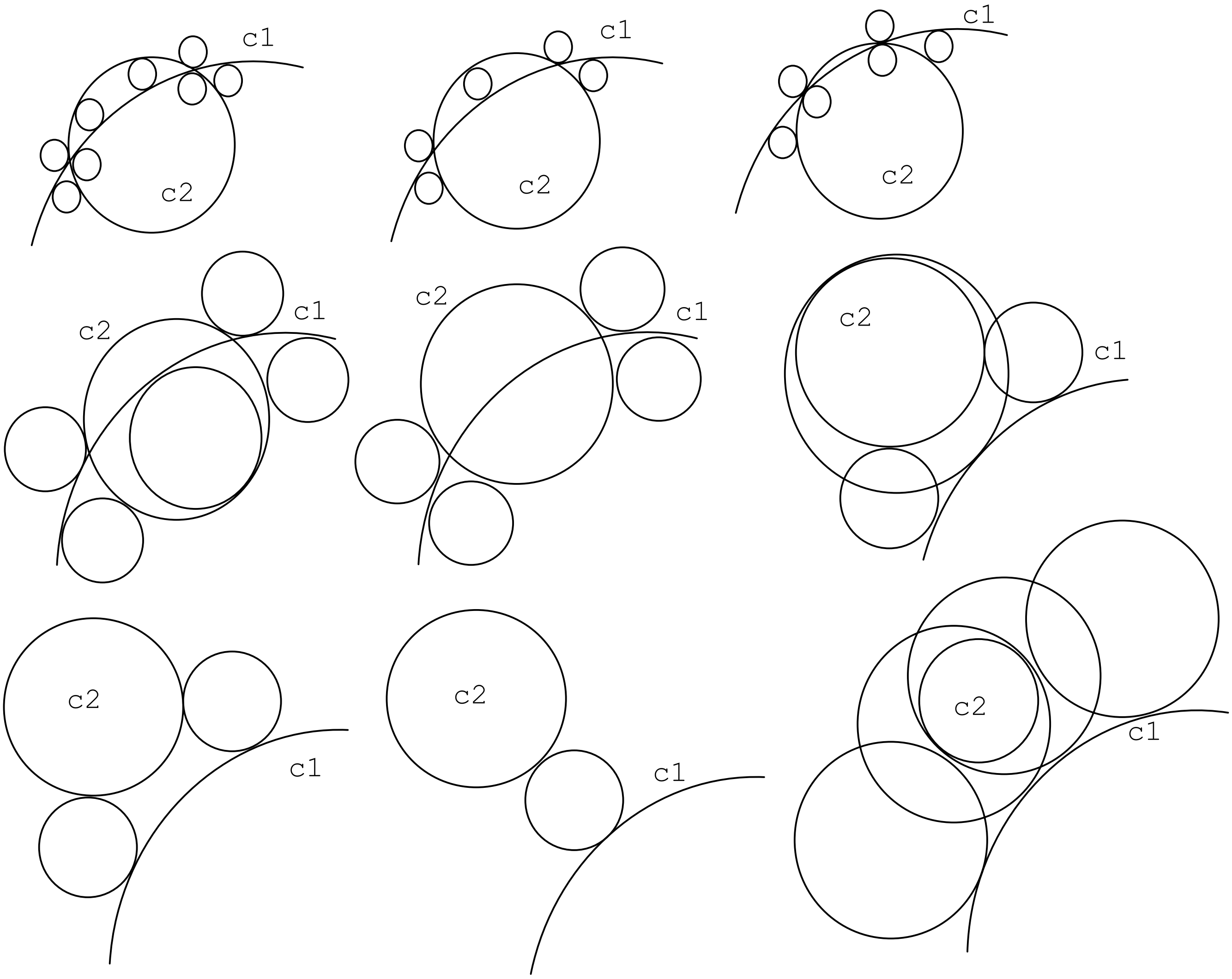


In Fig. 5.57. circle c3 is obtained by defining straight line l1 first and the circle c2 second. Circle c3 allows motion from line l1 to circle c2, and has a minor central angle.

Similarly, circle c4 is obtained by defining circle c2 first and straight line l1 second. C4 allows the motion from circle c2 to line l1, and has a minor central angle.

This method also applies to defining a circle that is tangent to two predefined circles. Unless you account for direction compatibility, order of definition and minor central angle, 1 to 8 solutions are possible (Fig. 5.58.).

Fig. 5.58. - Circles Tangent to Two Predefined Circles

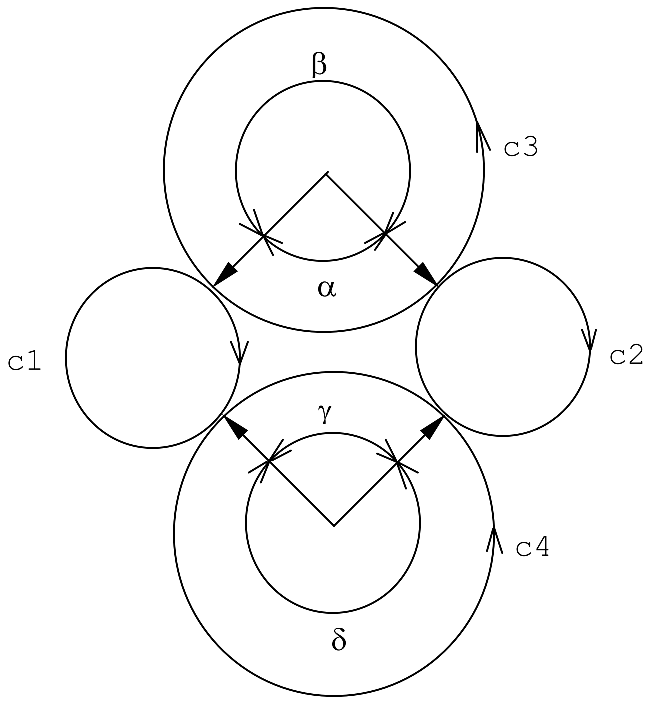


If you account for direction compatibility between the predefined circles and the tangent circle you want to define, there are only two possible solutions.

To distinguish between two tangent circles that have the same direction and the same radius, consider the order in which you defined the circle and the arcs of the tangent circles.

GTL generates the circle moving from the first to the second predefined circle, with the tangent circle having a minor central angle (Fig. 5.59.).

Fig. 5.59. - Circles Tangent to Two Predefined Circles



In Fig. 5.59., to obtain circle c3, you must define circle c1 first and circle c2 second. To obtain circle c4, you must define circle c2 first and circle c1 second.

Direct Programming Format

Circle with cartesian centre coordinates and radius (Figs. 5.60., 5.61.)
cn = [om] I.. J.. r..

Circle with polar centre coordinates and radius (Fig. 5.62.)
cn = [om] m.. a.. r..

Indirect Programming Format

Circle of given radius and tangent to two predefined straight lines (Fig. 5.63.)
cn = [-]lm,lp,r..

Circle tangent to a straight line and predefined circle of given radius (Figs. 5.64., 5.65., 5.66.)
cn = [-]lm,[-]cp,r..
cn = [-]cp,[-]lm,r..

Circle of given radius through a predefined point and tangent to a predefined straight line (Fig. 5.67.)
cn = pm,[-]lp,r..
cn = [-]lp,pm,r..

Circle of given radius tangent to two predefined circles (Figs. 5.68., 5.69.)
cn = [-]cm,[-]cp,r..

Circle of given radius through one predefined point, tangent to one predefined circle (Fig. 5.70.)
cn = pm,[-]cp,r..
cn = [-]cp,pm,r..

Circle of given radius through two predefined points (Fig. 5.71.)
cn = pm,pq,r..

Circle with centre at a predefined point and tangent to a predefined straight line (Fig. 5.72.)
cn = pm,[-]lp

Circle with centre at a predefined point and tangent to a predefined circle (Fig. 5.73.)
cn = pm,[-]cp[,s2]

Circle through three points (Fig. 5.74.)
cn = pm,pq,pr

Circle of given radius with centre at a point (Fig. 5.75.)
cn = pm,r..

Circle concentric to a predefined circle and at a given distance from it (Fig. 5.76.)
cn = [-]cm,d..

where:

cn	identifies the name of the circle of index n (n is a number between 1 and the max. configured number)
I.. J.	circle centre coordinates
r..	circle radius (positive if CCW, negative if CW)
[-]lm **[-]lp**	predefined straight lines of indexes m and p They can have opposite direction if sign "-" is used.
pm pq pr	predefined points of indexes m, q and r
[-]cm **[-]cp**	predefined circles of index m and p. They can have opposite direction if sign "-" is used.
[s2]	attribute for the greater of the two possible circles
d..	distance between two circles: it is positive if, looking at [-]cm, cn is on its left; it is negative if it is on its right.

Fig. 5.60.

c2 = I50 J100 r-40

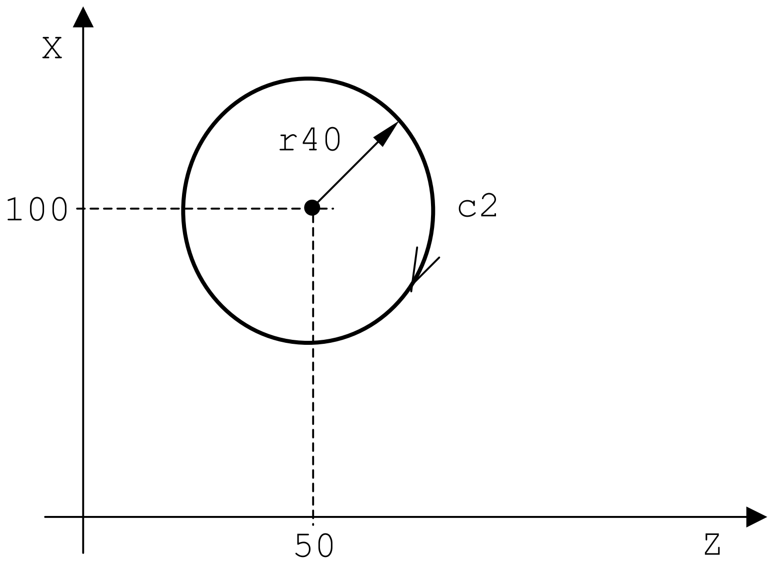


Fig. 5.61.

c1 = o1 I20 J20 r-15

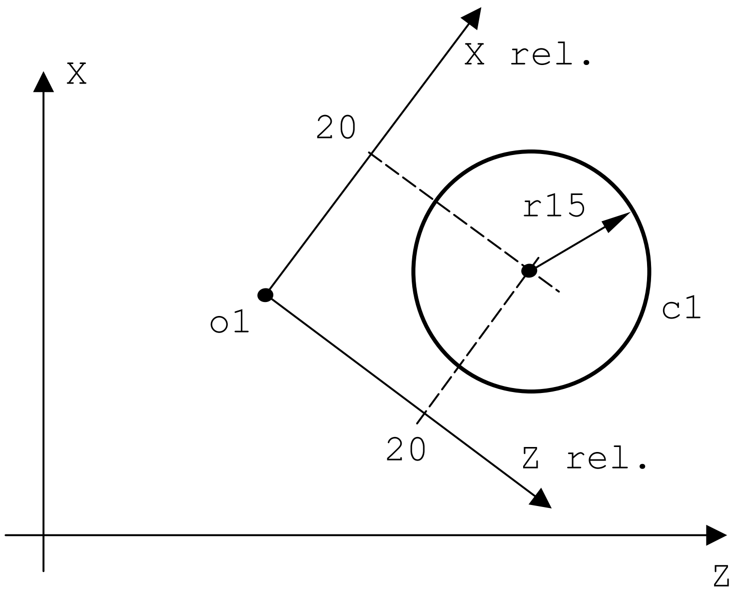


Fig. 5.62.

c2 = m70 a30 r15

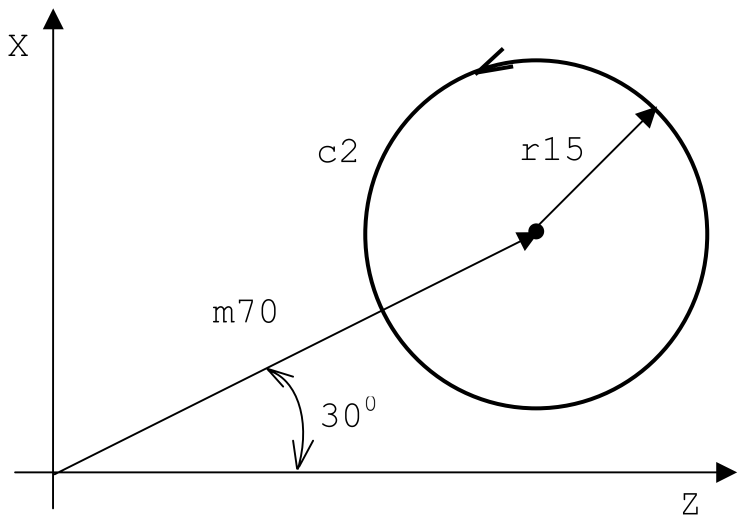


Fig. 5.63.

c3 = l1, l2, r-15

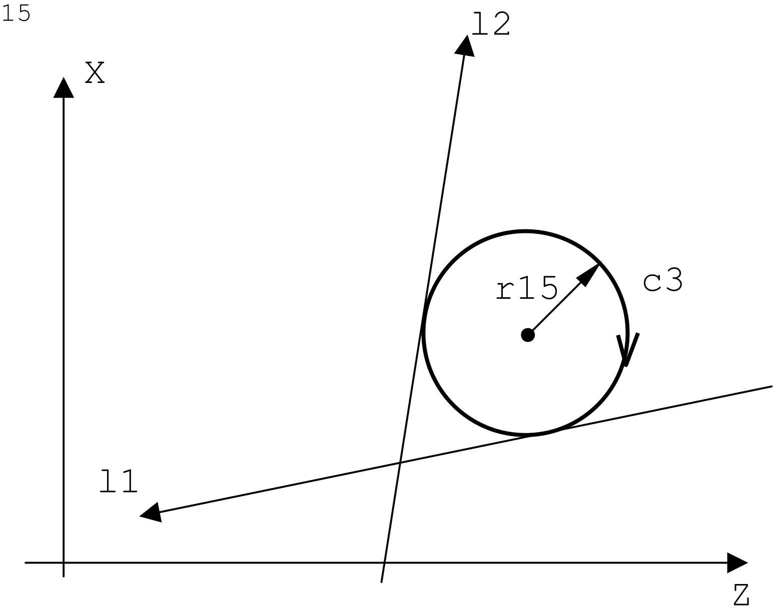


Fig. 5.64.

```
c3 = l1, -c2, r8
c4 = -c2, l1, r8
```

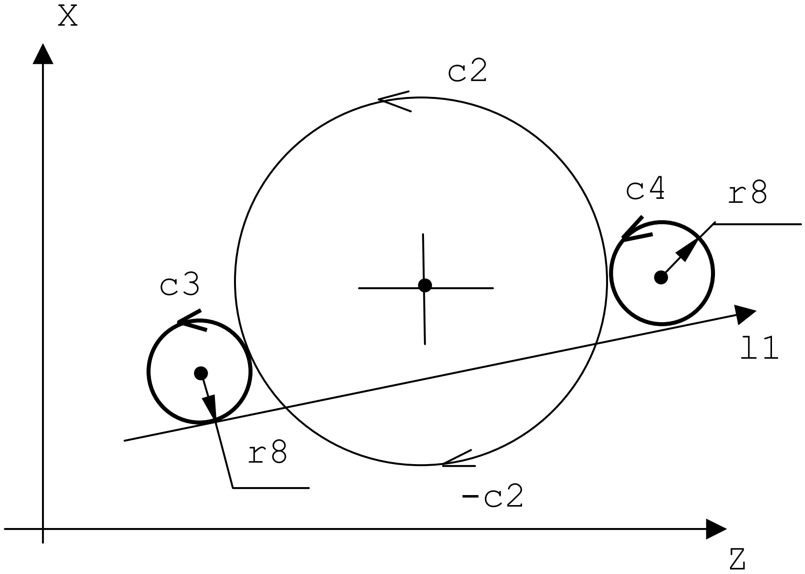


Fig. 5.65.

```
c9 = -c2, l1, r-8
c10 = l1, -c2, r-8
```

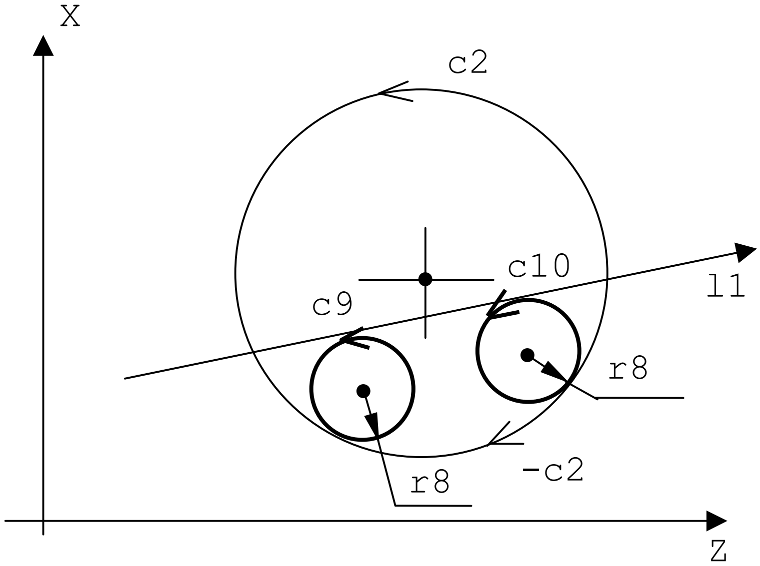


Fig. 5.66.

```
c4 = -l2, c1, r-40
c5 = c1, -l2, r-40
```

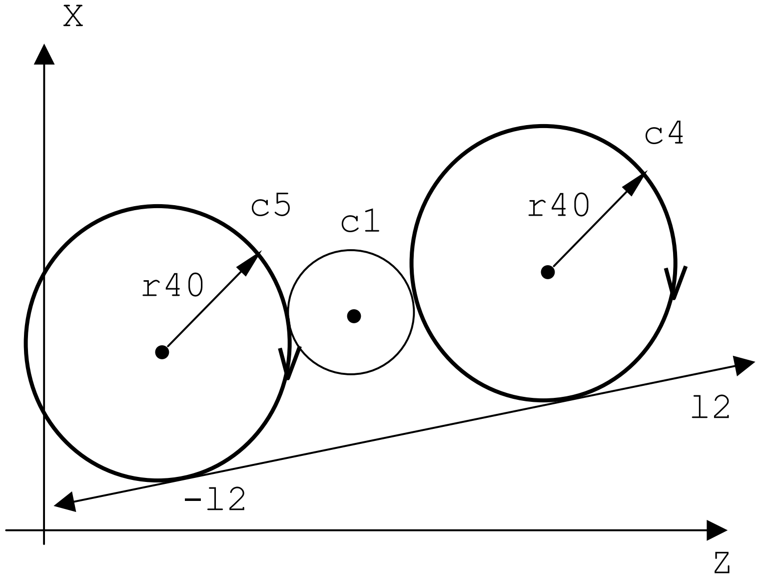


Fig. 5.67.

```
c3 = p1, -l1, r25
c4 = -l1, p1, r25
```

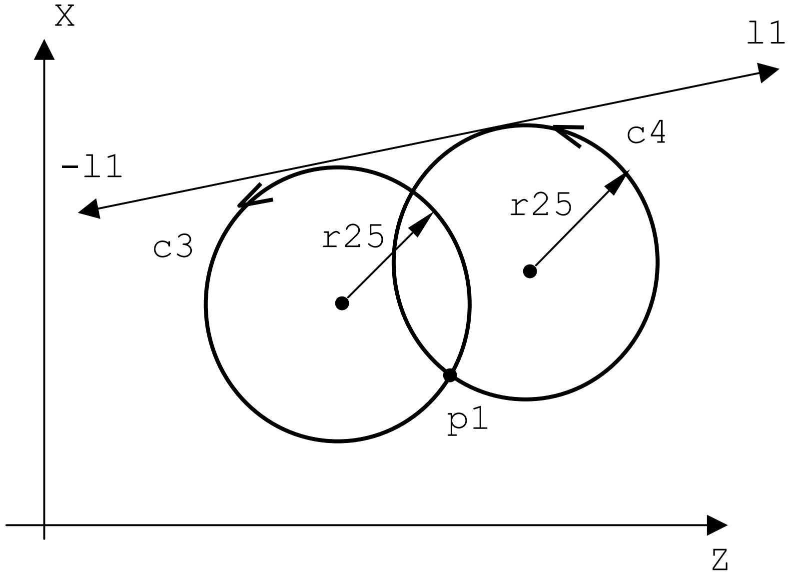


Fig. 5.68.

```
c5 = c1, c2, r-8
c6 = c2, c1, r-8
```

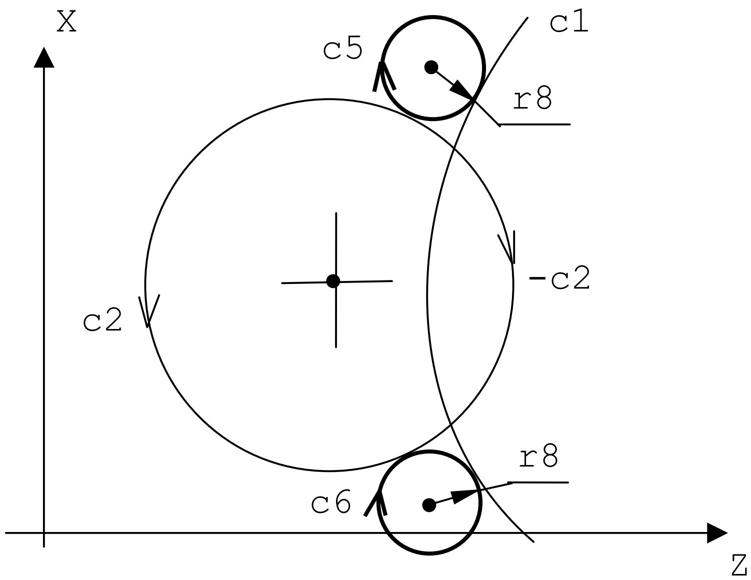


Fig. 5.69.

```
c9 = -c2, c1, r-8
c10 = c1, -c2, r-8
```

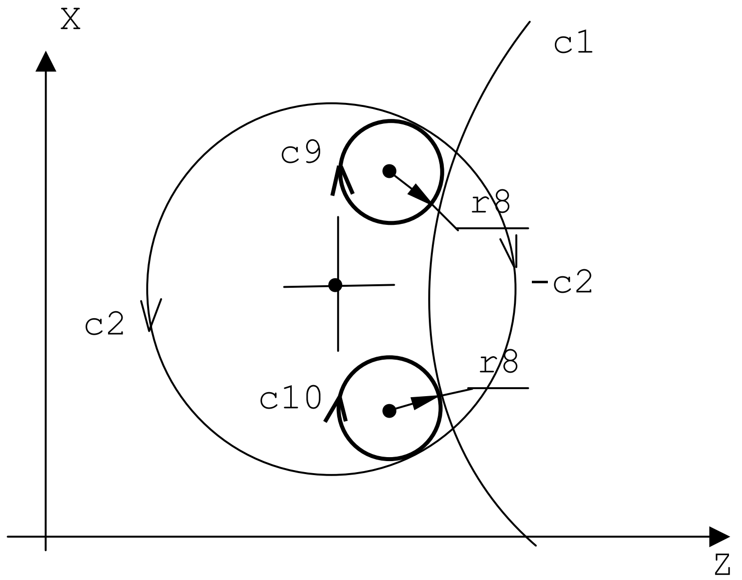


Fig. 5.70.

```
c2 = c1, p1, r60
c3 = p1, c1, r60
```

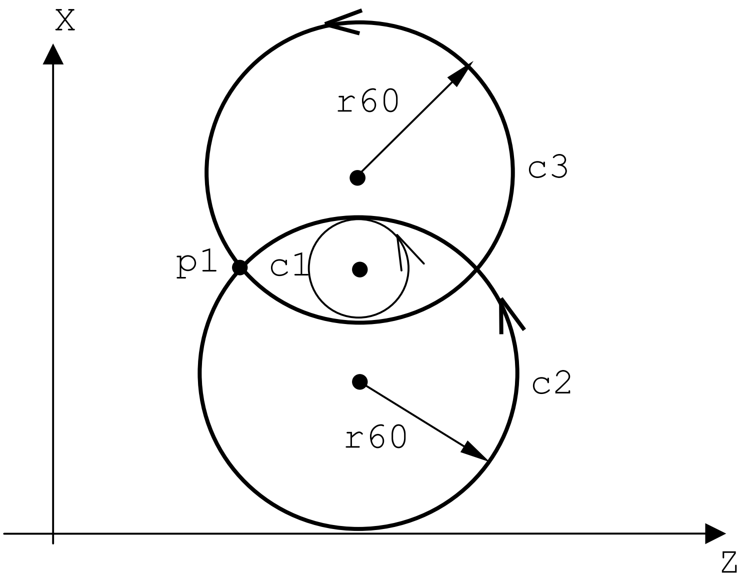


Fig. 5.71.

```
c1 = p1, p2, r20
c2 = p2, p1, r20
```

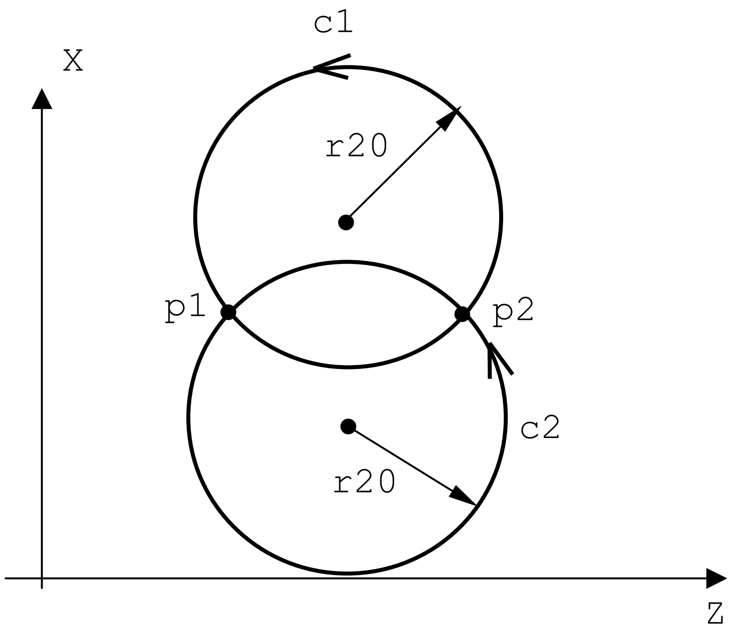


Fig. 5.72.

```
c3 = p1, l2
```

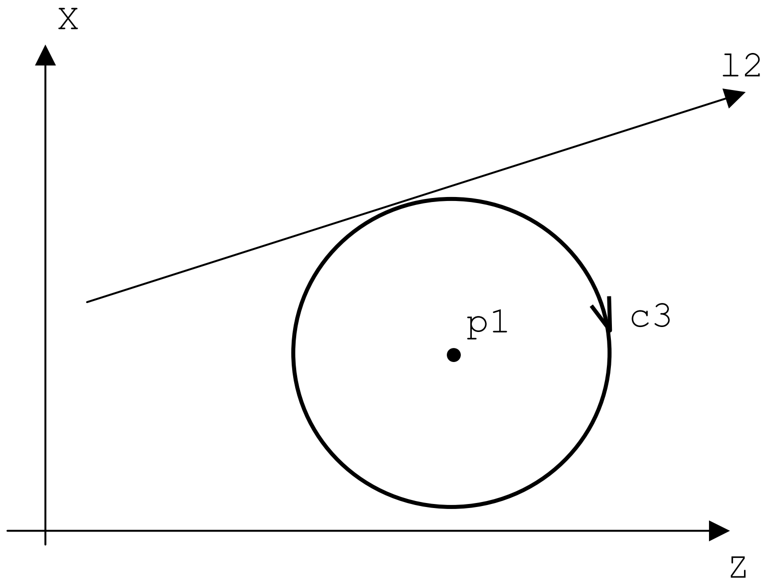


Fig. 5.73.

```
c2 = p1, c1
c3 = p1, c1, s2
```

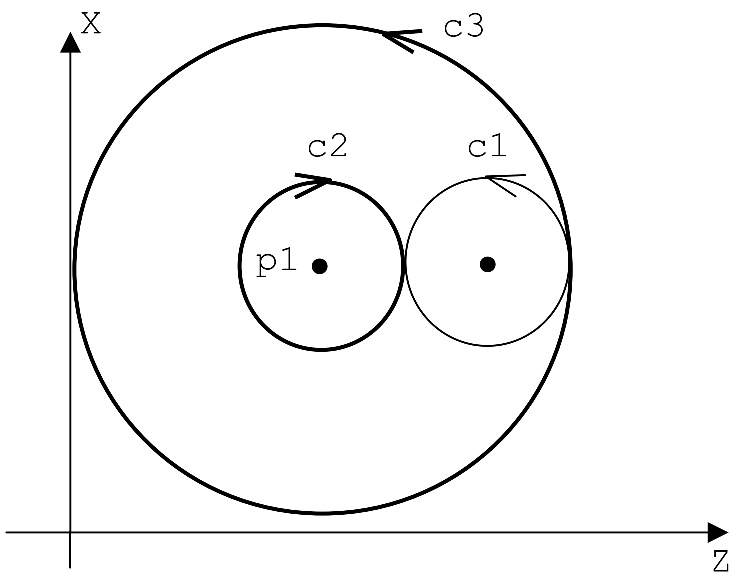
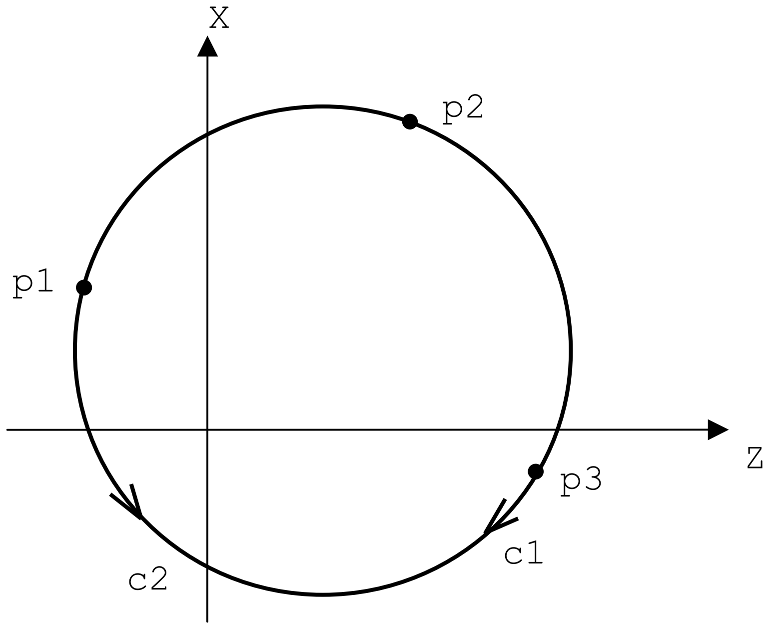


Fig. 5.74.

```
c1 = p1, p2, p3
c2 = p3, p2, p1
```



Important. The direction of the circle is from the first to the second and third defined point.

Fig. 5.75.

```
c1 = p1, r-40
```

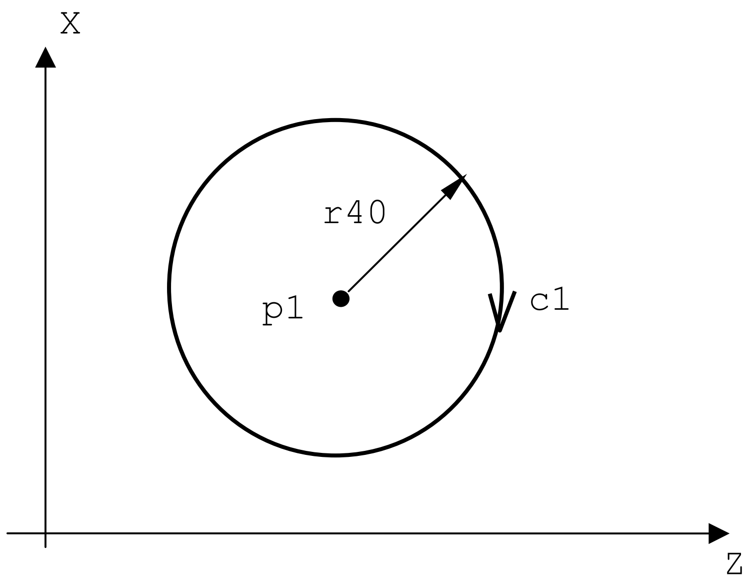
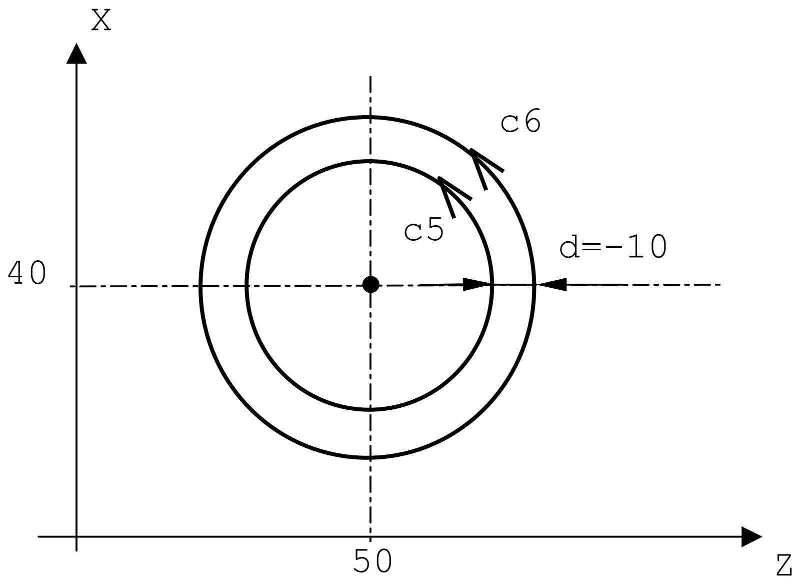


Fig. 5.76.

c6 = c5, d-10



DEFINING A PROFILE

A profile is a sequence of geometrical elements that have been stored before machining begins.

Starting and Ending a Profile

To program the start and end of a GTL profile, use the following instructions:

G21 identifies profile start;
G20 identifies profile end.

The only G functions allowed inside a GTL profile are G27-G28-G04-G41-G42.

Open and Closed Profiles

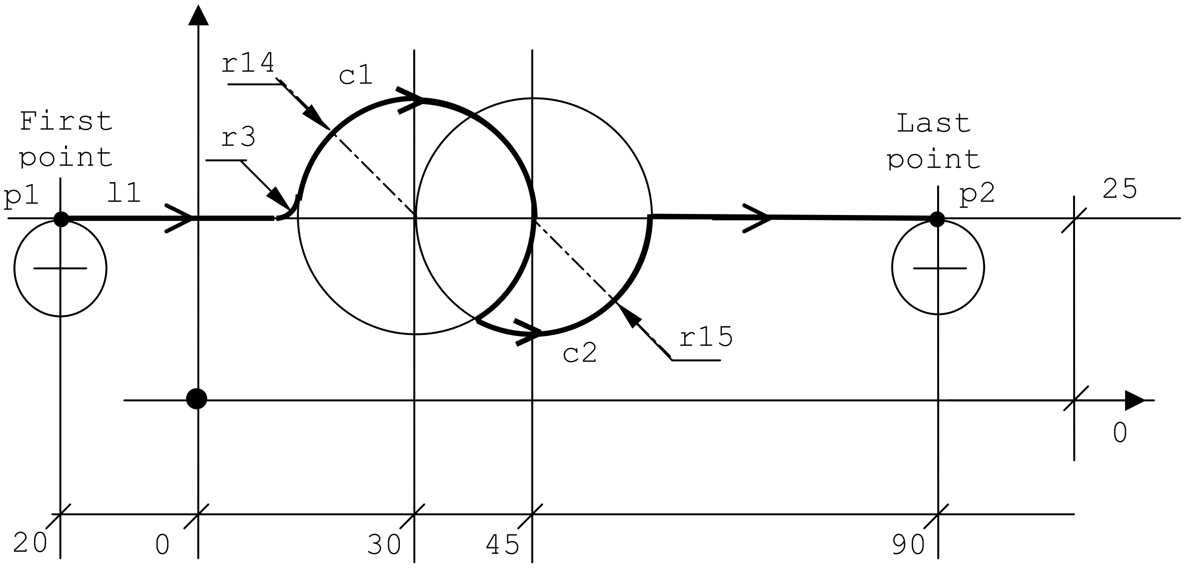
A profile must start with a point (pn) and end with a point different from the first one.

If point pn is programmed only in the starting block of the profile, the profile is open. Point pn cannot be programmed in the profile.

Cutter compensation acts normally to the first elements at the profile start point and normally to the last element at the final point.

Activate tool radius compensation on the first point of the profile by programming G21 and G41 or G42. Deactivate tool compensation on the last point with functions G20 and G40 (refer to Figure 5.77.).

Fig. 5.77. - Open profile




```
.............
l1   = Z X25,a
p1   = Z-20 XZ5
p2   = Z90 X25
c1   = I30 J25 r-14
c2   = I45 J25 r15
.............
G21  G42 p1               -first point
l1
r3
c1  s2
c2  s2
l1
G20 G40 p2                -last point
.............
```

Important. The cutter radius offset must be activated on the first profile point and deactivated on the last profile point.

Offset is deactivated on the first motion block following G40.

To program a closed profile, you must program the last element first and the first element second (refer to figures 5.79. and 5.80.).

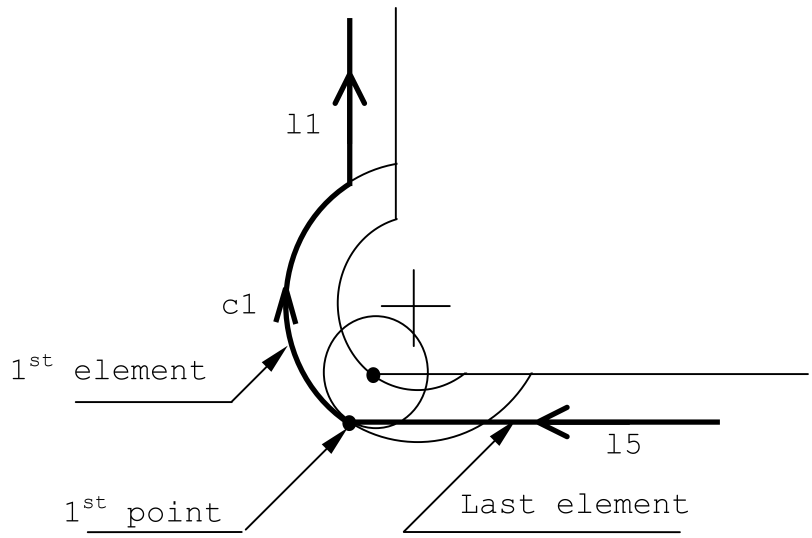
To activate tool offset, program G21 and G41 or G42 in the block containing the last element. To deactivate tool offset, program G20 and G40 at the end of the profile, in the block recalling the first element.

The first offset point is the intersection between the first and the last element.

For straight line-circle or circle'circle intersections, two solutions are possible.

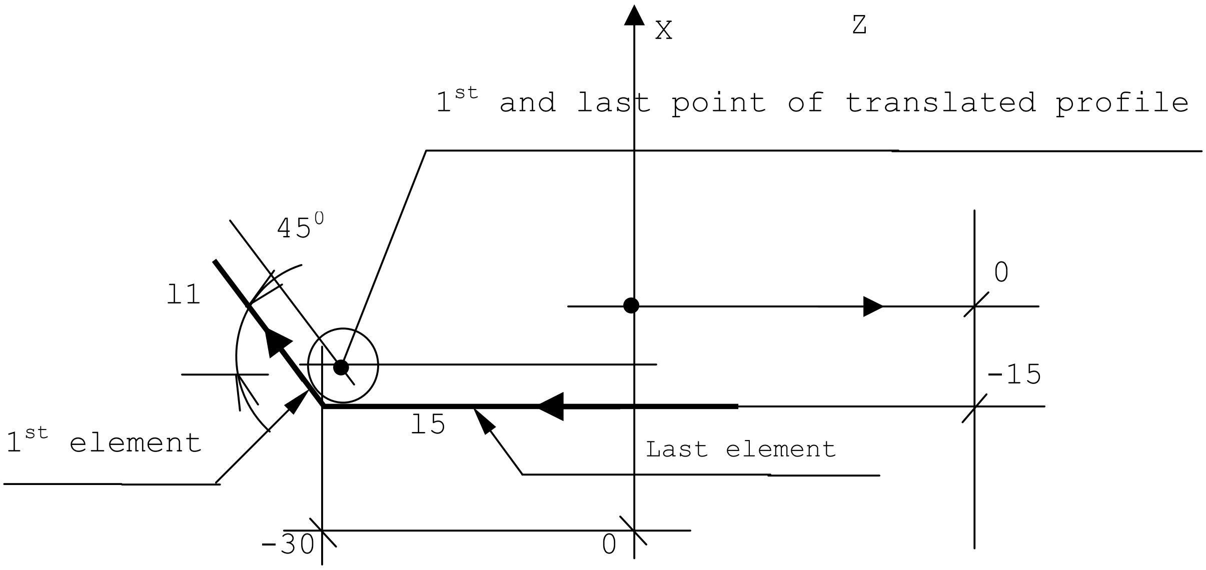
The control automatically selects the first intersection. If you need the second intersection, you must program the intersection selector s2, after defining the first element.

Fig. 5.78. - Closed Profile




```
.............
l5 = Z X-15,a180
.............
l1 = Z-30 X-15,a135
.............
G21  G42  l5                  -last element
l1                            -first element
.............
l5                            -last element
G20 G40 l1                    -first element
```

Fig. 5.79. - Closed profile




```
c1 = I..  J..  r
.............
l1 = Z.. X..,a90
.............
l5 = Z.. X.., a180
.............
G21  G42  l5 s2              -last element
c1  s2                       -first element
l1
.............
l5 s2                        -last element
G40 c1                       -first element
.............
```

Important. The cutter radius offset must be programmed at profile start, in the block specifying the last item. Offset must be deactivated at profile end, in the block recalling the first item.

Offset is disabled in the first movement block following G40.

Connecting GTL elements

You can connect the elements of a profile by using:

- tangency or intersection conditions;
- automatic radius.

a) Connections with Intersections

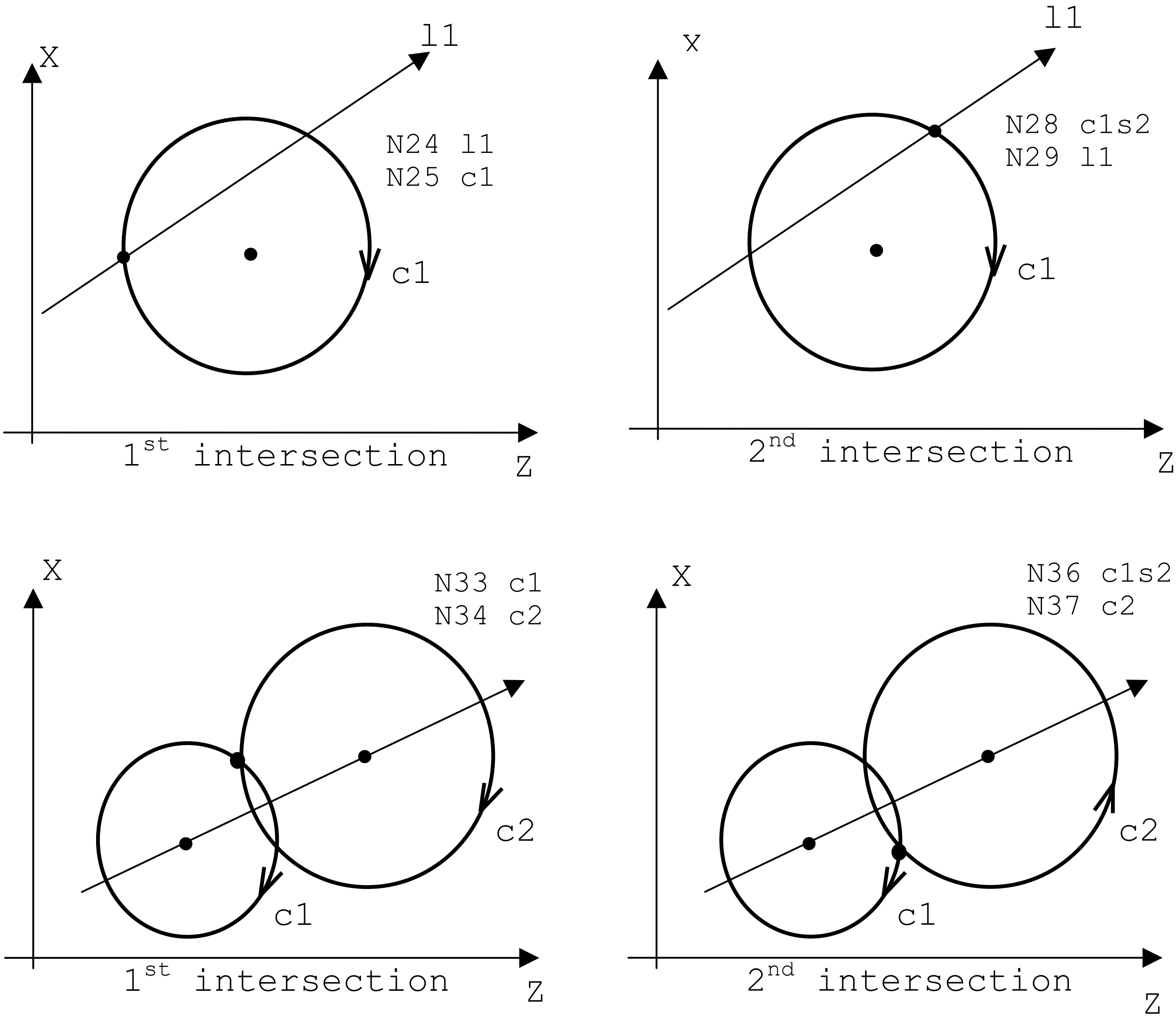
There is only one possible solution for the intersection of two straight lines.

For straight line-circle or circle-circle intersections, however, two solutions are possible.

The control automatically selects the first intersection. To select the second intersection, program selector s2 after defining the first element.

Examples of line-circle intersections are shown in Figure 5.80.

Fig. 5.80. - Line-Circle Intersections



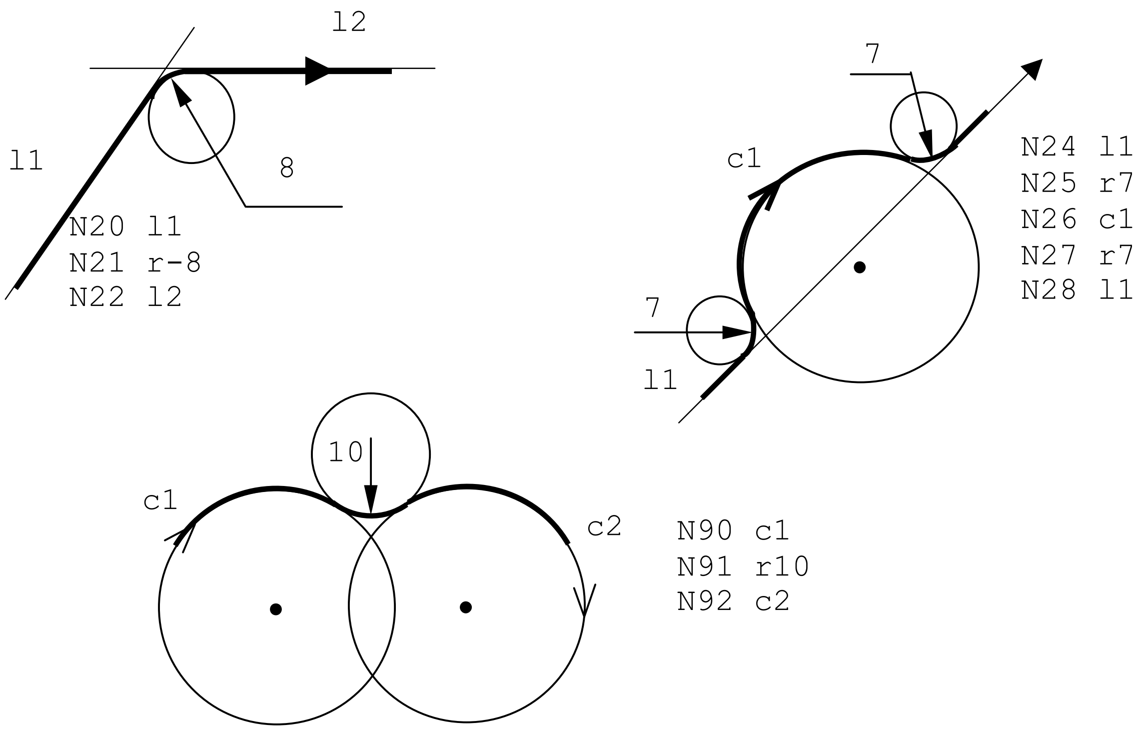
For straight line-circle intersections, the direction of movement of the straight line determines the first and second intersections.

For circle-circle intersections, the first intersection is the one on the left of the straight line that connects the centre of the first circle with the centre of the second circle. The second intersection is the one on the right of the same straight line.

b) Connection with automatic radius

You can define a radius between straight lines or circles that intersect by programming the radius value (positive for CCW profiles, negative for CW profiles).

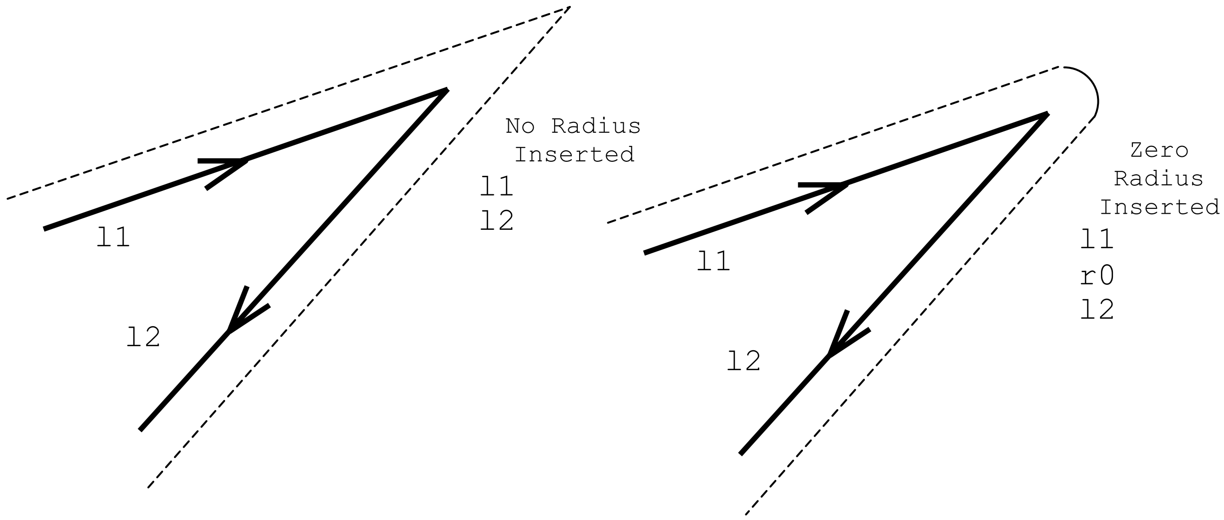
Fig. 5.81. - Connecting Elements with Automatic Radius



Important. A profile can neither start nor end with a radius, i.e. you should not program the radius value in the block immediately following the G21 block or in the block preceding the G20 block.

With active cutter radius offset, the tool positions on the intersection of the two geometric elements, translated by cutter radius. If a radius must be inserted between the two elements, program a 0 radius. An example is shown in Figure 5.82.

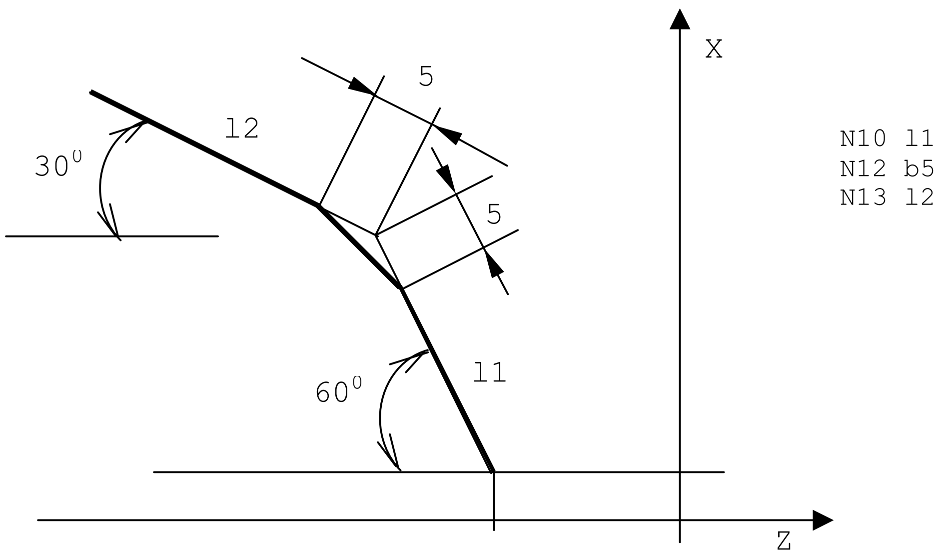
Fig. 5.82. – Zero Radius with Intersections and Tool Compensation



Bevels

To program a bevel between two straight lines, program the value of the bevel without a sign, i.e., as a distance from the intersection point. For example:

Fig. 5.83. - Bevel

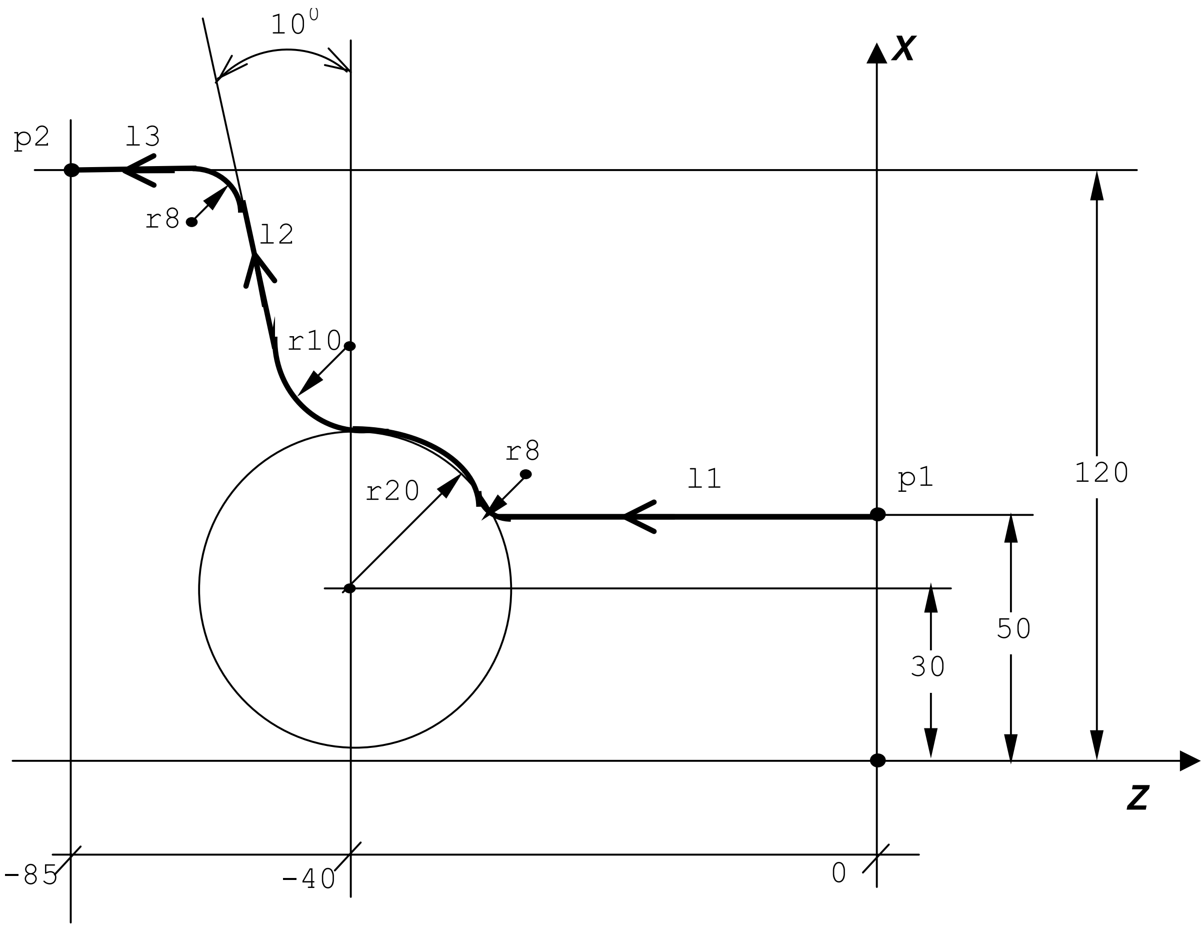


A profile can neither start nor end with a bevel, i.e. you should not program the bevel value in the block immediately following the G21 block or in the block preceding the G20 block.

In GTL programming, all movements are performed at feedrate. To program a movement at rapid rate, program a high feedrate.

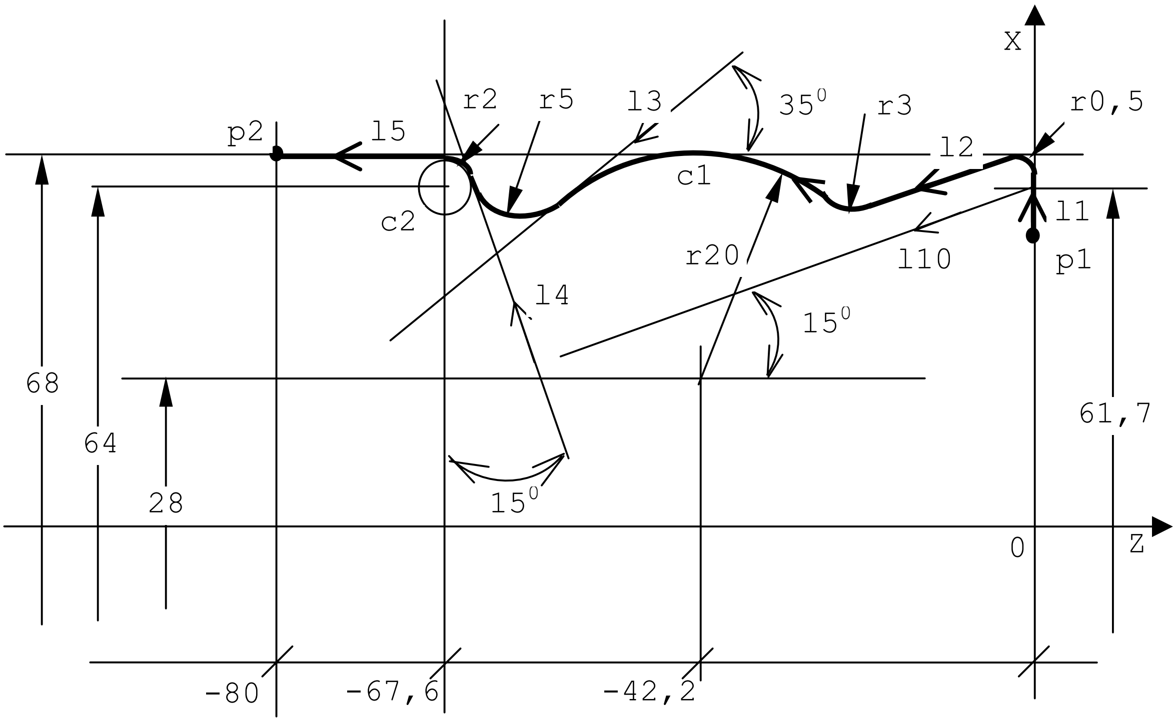
EXAMPLES OF GTL PROGRAMMING

Fig. 5.84. - Example 1




```
N1  (DIS, "EXAMPLE GTL")
N2  p1=Z0 X50
N3  l1=p1,a180
N4  c1=I-40 J30 r20
N5  l2=Z-40 X30,a100
N6  p2=Z-85 X120
N7  l3=p2,a180
N8  T1.1  M6 S200 M3
N9  G X100 Z10
N10 G21  G42 p1
N11 l1
N12 r-8
N13 c1
N14 r-10
N15 l2
N16 r8
N17 l3
N18 G20 G40 p2
N19 G X150 Z..
N20 .........
```

Fig. 5.85. - Example 2




```
p1=Z0 X60
l1=p1,a90
l10=Z0 X61.7,a195
l2=l10,d3.5
c1=I-42.2 J28 r20
l3=c1,a-215
c2=I-67.6 J64 r2
l4=c2,a105
p2=Z-80 X68
l5=p2,a180
T1.1  M6 S200 M3
GO X65 Z2
G21  G42 p1
l1
r0.5
l2
r-3
c1
l3
r-5
l4
c2
l5
G20 G40 p2
GO X100
Z.........
..........
```

5.11. PARAMETRIC PROGRAMMING

You can use E parameters for the geometrical and technological data of a machining cycle. E parameters allow mathematical and trigonometric operations and using calculations of expressions as well.

The maximum number of E parameters must be defined during system configuration.

E parameters require different indexes for variables having different format. The allowable formats are shown in Table 5.8.

Table 5.8. - E Parameters and their Formats

Format	Parameters	Min/max value
BY (byte)	E0..E9	0 to 255
IN (integer)	E10..E19	-32768 to +32767
LI (long integer)	E20..E24	-2.147.483.647 to +2.147.483.647
RE (real)	E25..E29	±7 whole or decimal digits
LR (longreal)	E30..(*)	±16 significant whole and decimal digits ±13 whole digits

(*) Maximum number of E parameters defined during configuration.

The E parameters receive values in special assignment blocks. The format for an assignment block is:

En = <expression>

where <expression> can be a numerical value or a mathematical expression whose result will be stored in the E parameter having index n. n can be a number or an E parameter (of either byte or integer type).

EXPRESSIONS

An <expression> is a mathematical equation formed by arithmetic operators, functions and operands (E parameters or numerical constants).

Arithmetic operators are:

1) + (addition);
2) - (subtraction);
3) * (multiplication);
4) / (division).

Possible functions are:

- SIN(A) - calculates sine of A;
- COS(A) - calculates cosine of A;
- TAN(A) - calculates tangent of A;
- ARS(A) - calculates arcsine of A;
- ARC(A) - calculates arccosine of A;
- ART(A) - calculates arctangent of A;
- SQR(A) - calculates square root of A;
- ABS(A) - calculates absolute value of A;
- INT(A) - calculates integer of A;
- NEG(A) - inverts the sign of A;
- MOD(A,B)- calculates the remainder of the A to B ratio;
- FEL(A,B)- finds the value associated with a line with index A that is defined in GTL (ln, where n=A). B is the value to find (B=1,2 or 3, where 1=sine of the angle, 2=cosine of the angle, 3=distance of the straight line from the origin). For example, with E30=FEL(5,1), E30 receives the value of the sine of the angle generated by the abscissa and the straight line l5;
- FEP(A,B)- finds the value associated with a point with index A. B specifies the value to find. (B = 1 or 2, where 1=abscissa of the point, 2=ordinate of the point). For example, with E34=FEP(4,2), E34 receives the value of the ordinate of point p4;
- FEC(A,B)- finds the value associated with a circle with A index that is defined in GTL. B specifies the value to find. (B=1,2 or 3, where 1=abscissa of the center, 2=ordinate of the center, 3=radius of the circle). For example, with E42=FEC(8,3), E42 receives the value of the radius of circle r8.

For the values of (A) and (A.B) you can use E parameters or numerical constants.

The control solves mathematical equations by considering the priority of brackets and signs. If allowed, the result is converted into the format of the E parameter on the left of the = sign.

Important. You must express the values of trigonometric functions (SIN, COS, TAN) in degrees. The results of inverse trigonometric functions (ARS, ARC, ART) must also be expressed in degrees.

The following are examples of assignment blocks for calculating parameters:

N1 E37=(E31*SIN(E30)+123.4567)/SQR(16)	solves the expression and assigns the result to parameter E37
"LAB1"E51=-0.00000124+5	calculates the expression and assigns the result to parameter E51
E40=TAN(35)	calculates the tangent of 35 degrees and assigns the result to parameter E40
/E35=FEP(37,1)	calculates the abscissa value of point p37, previously defined, and assigns the result to parameter E35
E31=NEG(E31)	changes the sign of parameter E31
E7=81	assigns the value 81 to Parameter E7
E25=E25+30	adds 30 to the current value of E25 and assigns the result to E25
E2=SK396	assigns to E1 the content of Byte 396 from the K buffer
E8=SYVAR1	assigns to E8 the value of variable SYVAR 1

You can use E parameters either inside a program or inside a subroutine.

E parameters normally remain stored after a power down. If needed, you can select the automatic reset of E parameters during configuration.

To display the current value of an E parameter, use the DIS command. For example, (DIS,E54) displays the value of E54.

The utilisation of E parameters is summarised in table 5.9.

Table 5.9. - Utilisation of E parameters

Parameters (Format)	Data (geometric-technical)	Programming examples
EO..E9 (BY)	G functions M functions RPT code	GE1 ME3 (RPT,E9)
E10..E19 (IN)	Absolute origin No. S functions	(UAO,E10) (UOT,E11,Z..,X..) SE15
E20..E24 (LI)		
E25..E29 (RE)	F functions URT code SCF code Indexed axes	FE27 (URT,E25) (SCF,E26) PE29
E3O...(*) (LR)	C X Z axes coordinates R coordinate I J K miscellaneous operators u v w compensation factors Global system variables: TMR UOV	 XE32 RE33 KE34 E35 TMR=E38 UOV=E40

5.12. PROGRAM BLOCKS WITH THREE-LETTER COMMANDS

This section describes the function and syntax of program blocks that use three-letter commands.

We have grouped three-letter commands in seven classes. You can use them for:

- modifying the reference system of the axes (section 5.12.1.)
- modifying the sequence of program execution (section 5.12.2.)
- performing miscellaneous commands (section 5.12.3.)
- performing I/O commands (section 5.12.4.)
- managing the graphic display (section 5.12.5.)
- monitoring tool life (section 5.12.6.)
- managing a probe (section 5.12.7.)
- managing tool offsets (section 5.12.8.)

5.12.1. MODIFYING THE REFERENCE SYSTEM OF THE AXES

The commands in this class allow you to change the cartesian reference system in which you programmed a profile.

The following commands belong to this class:

1) UOT - use temporary origins;
2) UIO - use incremental origins;
3) MIR - mirror machining;
4) URT - rotation of the plane;
5) SCF - scale factor.

USE TEMPORARY ORIGINS - UOT

This command allows an incremental shift of the current origin for each axis specified in the command.
The allowable format is:

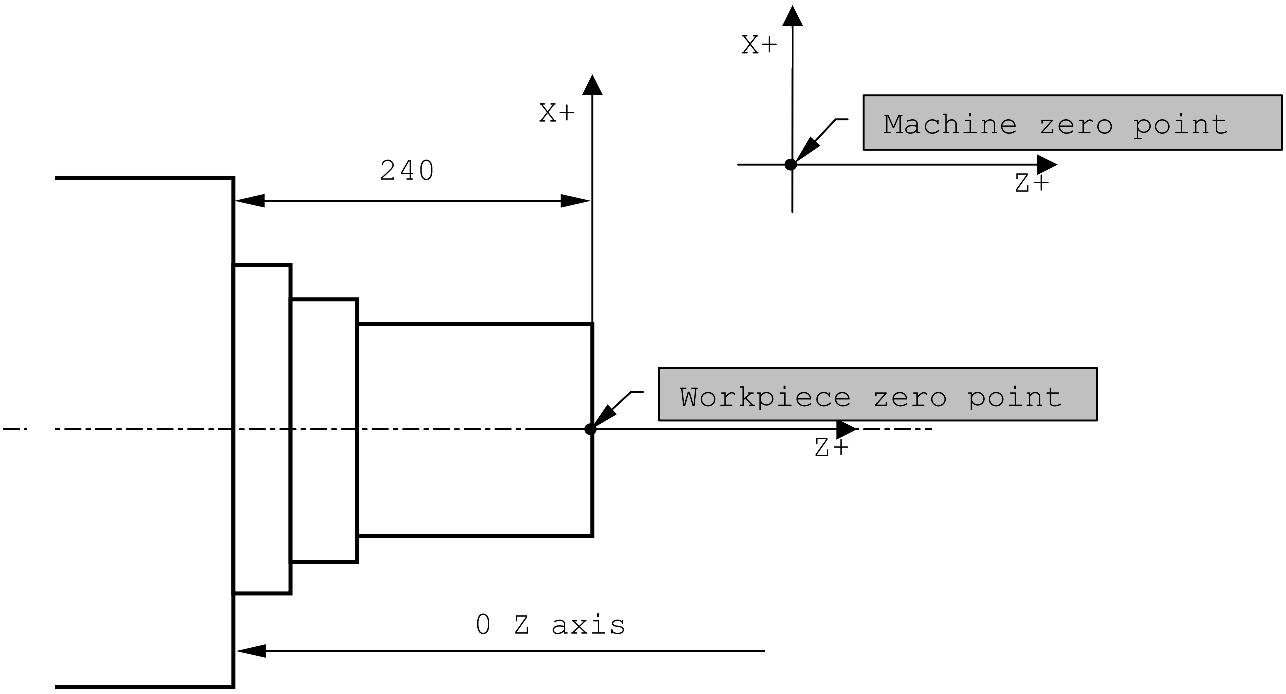
(UOT,0,VAR-1[,VAR-2])

where:

VAR-i — is an axis and a dimension.
The control takes the dimension as an absolute offset and adds it to the value of the absolute origin for that axis.
For undeclared axes, the current origin stays in effect.

Example (Fig. 5.86.):

Fig. 5.86. - Using UOT command




```
..............
N1 (UOT,0,Z240)
N2 T3.3 M06
..............
```

Notes. A temporary origin remains in effect until you redefine it with a new UOT or you restablish the absolute origin with either (UOA,0) or **RESET**.

USE INCREMENTAL ORIGINS - UIO

This command allows an incremental shift of the current origin for each axis specified in the command.
The allowable format is:

(UIO,VAR-1[,VAR-2...VAR-n])

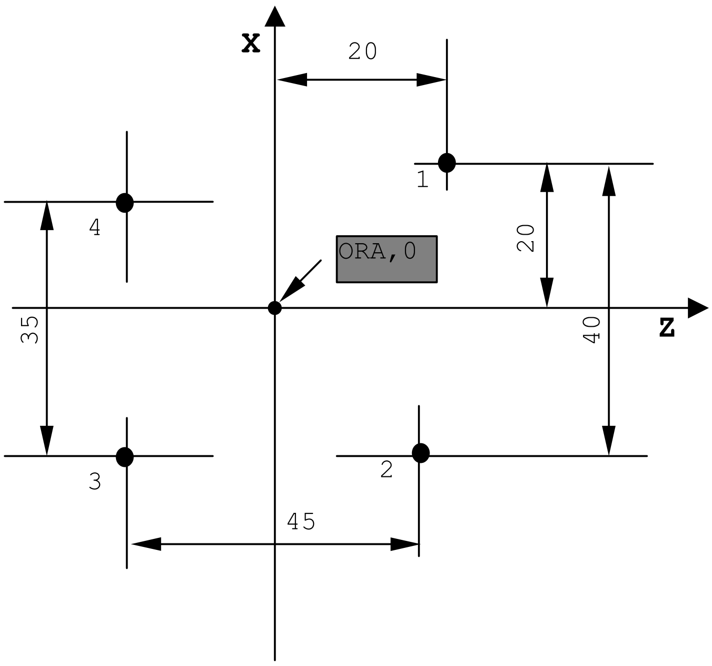
where:

VAR-i — is an axis and a dimension.
The control takes the dimension as an absolute offset and adds it to the value of the absolute origin for that axis. For undeclared axes, the current origin stays in effect.

Example (Fig. 5.87.):

```
............
N65    (UIO,Z20,X20)      -point 1
............
N121   (UIO,X-40)         -point 2
............
N180   (UIO,Z-45)         -point 3
............
N230   (UIO,X35)          -point 4
............
N300   (UA0,0)
```

Fig. 5.87. - Using UIO command



Notes. An incremental origin remains in effect until you redefine it with a new UIO or you restablish the absolute origin with either (UOA,0) or **RESET**.

MIRROR MACHINING - MIR

The MIR command reverses (mirrors) the programmed direction of motion for the axes specified in the command.

The allowable format is:

(MIR[,VAR-1,...,VAR-n])

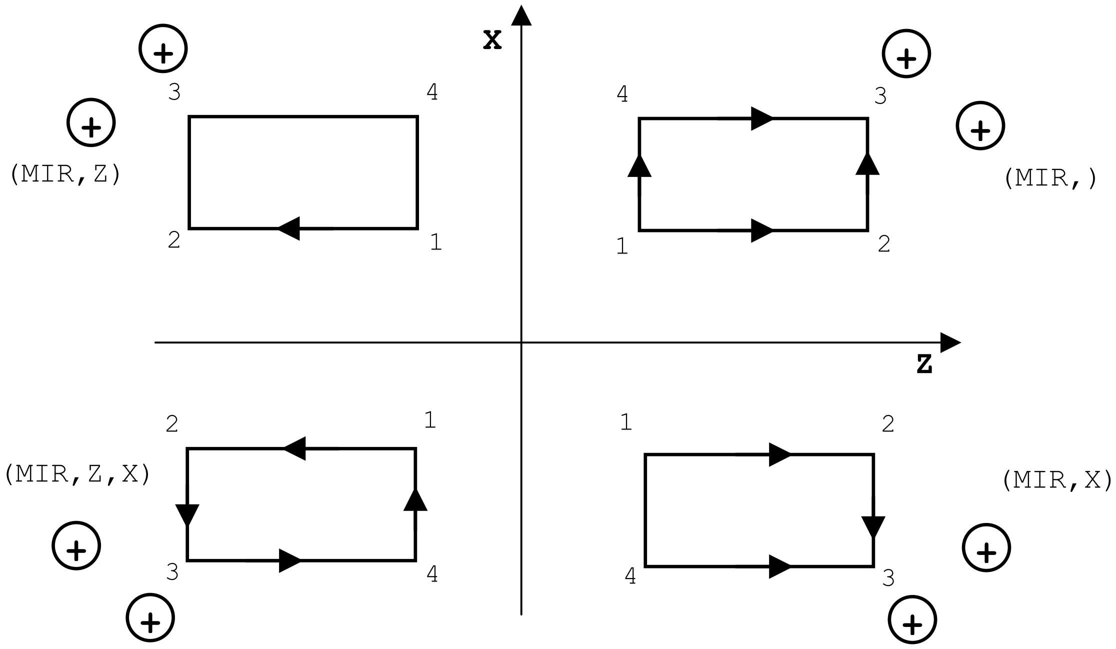
where:

VAR-n is a letter that corresponds to a configured axis in the system

Example (Fig. 5.88.):

```
N24   (MIR,Z)
..........
N42   (MIR,Z,X)
..........
N84   (MIR,X)
..........
N99   (MIR)
```

Fig. 5.88. - Use of the MIR command



Notes:

- The control mirrors programmed axis move with respect to the current origin.

- You can declare as many as 8 axes. If no axes are programmed in the MIR command, the mirror function is deactivated for all configured axes.

- The control applies the mirror function to an axis beginning with the first movement of that axis after the MIR command.

- Rotation and mirror commands (respectively, URT and MIR) are applied in the following order: MIM first, URT second.

ROTATION OF TNE PLANE - URT

This command allows you to rotate the interpolation plane with respect to the currently active origin.
The allowable format is:

(URT, OPERAND)

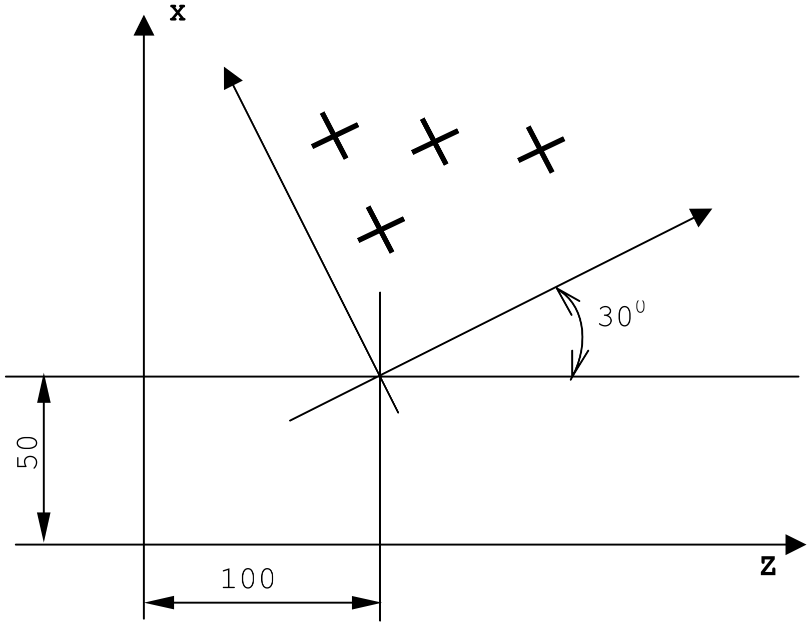
where:

OPERAND the value of the rotation angle in degrees (with decimals). It can be specified in explicit or implicit mode (real E parameter from E25 to E29). The angle must always be declared. If it is 0 the function is deactivated.

Example (Fig. 5.89.):

```
(UOT,0,Z100,X50)
(URT,30)
..........
..........
..........
(UAO,0)
(URT,0)
```

Fig. 5.89. - Using URT command



Notes:

• After an URT block, the control applies rotation to the programmed coordinates. However, coordinates referred to machine zero (G79) are not rotated.

• If programmed in the same block, MIR and URT are applied in the following order: MIR first, URT second.

Fig. 5.90. - Rotation of a GTL profile

```
N76  (DIS, "MILL D=8")
N77  M21
N78  G CO Z5
N79  (UAV.1,XC,UV.10)
N80  (DPI,U,V,)
N81  c1=I15 J15 r5
N82  c2=I50 J30 r5
N83  c3=I30 J50 r5
N84  l1=c1,c2
N85  l2=c2,c3
N86  l3=c3,c1
N87  F170 S800 T1.1  M6 M3
N88  E26=0
N89  (RPT,6)
N90  (URT,E26)
N91  G U15 V15
N02  GZ1  G41  c1
N93  Z-10
N94  l1
N95  c2
N96  l2
N97  c3
N98  l3
N99  c1
N100 Z
N1O1 G20 G40 l1
N102 E26=E26+60
N103 (ERP)
N104 (URT,0)
N105 (UAV,0)
N106 M20
N107 G G79 X Z M30
```

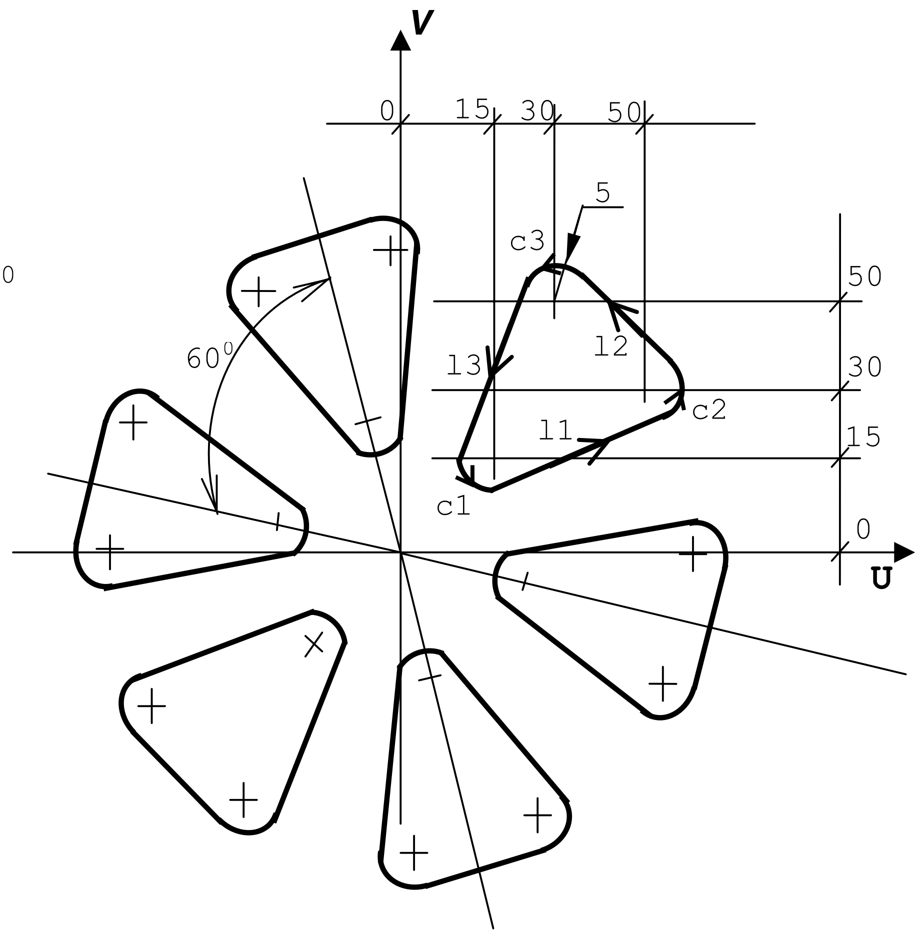
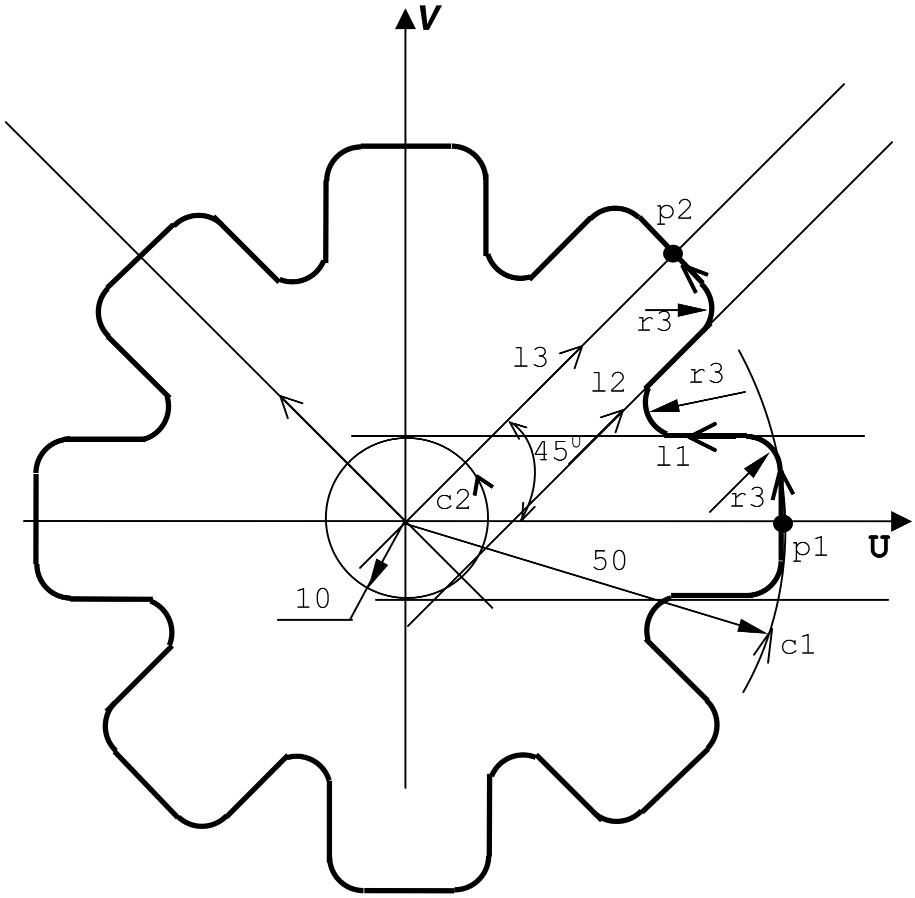


Fig. 5.91. - Profile repeated 8 times (2 passes)




```
N1  (DIS."GTL WITH ROTATION")
N2  M21
N3  GO CO Z5
N4  (UAV,1,XC,UV,10)
N5  (DPI,U,V)
N6  F..  S.. T2.2 M6
N7  UOV=2
N8  p1=X50 Y0
N9  c1=I0 J0 r50
N10 c2=I0 J0 r10
N11 l1=c2,a180
N12 l3=X0 Y0,a45
N13 l2=c2,a45
N14 p2=l3.c1.s2
N15 U60 V0
N16 Z-10
N17 "START" E25=0
N18 (RPT,8)
N19 (URT,E25)
N20 G21  G42 p1
N21  c1
N22  r3
N23  r3
N24 l2 s2
N25 r3
N26 c1
N27 G20 G40 p2
N28 E25=E25+45
N29 (ERP)
N30 (URT,O)
N31 "END" G
N32 UOV=O
N33 (EPP,START,END)
N34 (UAV,0)
N35 GO M2O
N36 Z5
N37 G79 X Z M30
```

USING SCALE FACTORS - SCF

The USF command assigns a scale factor to programmed axes dimensions. The control applies the scale factor to the axes specified in the SGF command.
The allowable format is:

(SCF[,n[,VAR-1,...,VAR-m]])

where:

n	defines the scale factor. It can be programmed explicitly (with a decimal number format) or implicitly (a real E parameter from E25 to E29).
VAR-i	is a character that represents one of the configured axes.

The control cancels scale factors for axes not specified in the command.

If the scale factor is not specified, the SCF command cancels scaling for all axes.

Example:

```
..........
(SCF,3)         -applies scale factor 3 to programmed
..........       for all configured axes
(SCF,2,X)       -applies scale factor 2 to X axis and
..........       deactivates scaling for all the other axes
..........
(SCF)           -deactivates scaling for all axes
```

Note. With SCF you can program as many as 8 axis names.

5.12.2. MODIFYING THE SEQUENCE OF PROGRAM EXECUTION

This section deals with codes that allow you to modify the order of execution of a part program. The commands in this class are:

RPT - repeat a set of program blocks
ERP - define the end of the repetition set
CLS - call a subroutine for execution
EPP - execute a subprogram
BNC, BGT, BLT, BEQ, BNE, BGE, BLE - branch commands

REPEAT A SET OF PROGRAM BLOCKS - RPT/ER

RPT and ERP delimit a set of program blocks that should be executed a specified number of times. The set begins with RPT and ends with ERP.
The format of the full command is:

```
(RPT,n)      -repetition command and number of times
.........
.........    -set of blocks to be executed the specified number
.........     of times
.........
(ERP)        -defines the end of the set
```

where:

n — is the number of executions. n must be a whole number from 1 to 99. You can program n explicitly or implicitly, with a byte type E parameter (E0 to E9).

The control allows 3 repeat levels. You can program up to 2 repeat commands inside another repeat command (Figure 5.92.).

Fig. 5.92. - Repeat levels

```
(RPT,8)      <-------------------------------------
.......                                            |
.......                                            |
(RPT,10)     <------------------------         |
.......                               |         |
.......                               |         |
(RPT,5)      <---------------         |         |
.......                     |  LEVEL  |  LEVEL  |LEVEL
             <---------------    1    |    2    |  3
(ERP)                                 |         |
.......                               |         |
.......      <------------------------         |
(ERP)                                           |
.......                                         |
.......      <-------------------------------------
(ERP)
```

The control allows 3 repeat levels. You can program up to 2 repeat commands inside another repeat command.

USING SUBROUTINES - CLS

The GLS command allows you to call and execute a subroutine that is stored in memory. A subroutine is a sequence of blocks that define a machining cycle.

The allowable format is:

(CLS,FILE NAME[/DEVICE])

where:

FILE NAME	is the name of the subroutine file to be recalled. It can have as many as 6 alphanumeric characters. The first character must be a letter. All letters must be capital letters.
/DEVICE	is the name of the device containing the program. Use / to separate the device name from the file name. The device name can have 2 or 3 alphanumeric characters. The first character must be a letter. All letters must be capital letters. If the device is not specified, the control defaults the device declared during configuration.

For example:

N1 (CLS,P800/MP1)	calls and executes subroutine P800 allocated on memory MP1. If MP1 is the default memory, it does not need to be programmed in the GLS command.

Example of a Subroutine Call:

```
Main program
N16 ..........                       Subroutine P800
N17  (CLS,P800)   ------------>      N500 ..........
N18  ..........              |       N501  .........
...............              |       N502  .........
N67  (GLS,P800)   -----------        N503  .........
N68  ..........                      ...............
```

Notes:

• Only two call levels are available, i.e., the program recalled with CLS can call up other programs but the called programs cannot call up other programs.

• Subroutines can be parametric, the numeric values of the parameters are defined in main program during the recall.

EXECUTING A PORTION OF A PROGRAM - EPP

EPP allows you to execute a portion of a part program contained between two blocks that have label fields.

The allowable format is:

(EPP,LABEL1,LABEL2)

where:

LABEL1 and LABEL2	are, respectively, the label fields of the first and last block of the portion. A label is a sequence of up to 6 alphanumeric characters. Program each label between quotes in the block of program, do not include the quotes in the branch block. Program the label before any sequence number in the block. If you use block delete code (/), program it before the label.

For example:

```
..........
"START"N25              -first block with  label
..........
..........
"END"N100               -last block with label
..........
N150  (EPP,START,END)   -EPP command that specifies the labels.
..........               The control will execute blocks N25 to N100. At
                         this point it will resume execution with the
                         block that follows N150.
```

Notes:

• An EPP command cannot occur during current EPP command execution.

• In contouring operations, you can use the EPP command to finish mill with the same blocks you programmed for roughing. During the roughing phase, program a stock allowance with the UOV command.

• In positioning operations, you can program points that have a centring operation, and then use EPP to call for different tools to execute separate operations at each hole if required.

USING BRANCHES INSIDE A PROGRAM

The branch commands cause program execution to jump to a block that contains a corresponding label field.

Branch commands can be unconditional or conditioned by E parameters, machine logic signals or numerical values. Table 5.9. shows the format and function of branch commands.

Table 5.9. - Branch Commands

Format	**Function**
(BNC,LABEL)	Branch to the label block unconditionally
(BGT,VAR1,VAR2,LABEL)	Branch to the label block if VAR1>VAR2
(BLT,VAR1,VAR2,LABEL)	Branch to the label block if VAR1<VAR2
(BEQ,VAR1,VAR2,LABEL)	Branch to the label block if VAR1=VAR2
(BNE,VAR1,VAR2,LABEL)	Branch to the label block if VAR1≠VAR2
(BGE.VAR1.VAR2.LABEL)	Branch to the label block if VAR1≥VAR2
(BLE,VRR1.VAR2.LABEL)	Branch to the label block if VAR1≤VAR2

where:

VAR1 and VAR2 — variables whose values determine the conditional branching. They can be E parameters, machine logic signals, global system variables, numerical values.

LABEL — a sequence of up to 6 alphanumeric characters. Program each label between quotes in the block of program. Do not include the quotes in the branch block.
Program the label before any sequence number in the block. If you use block delete code (/), program it before the label.

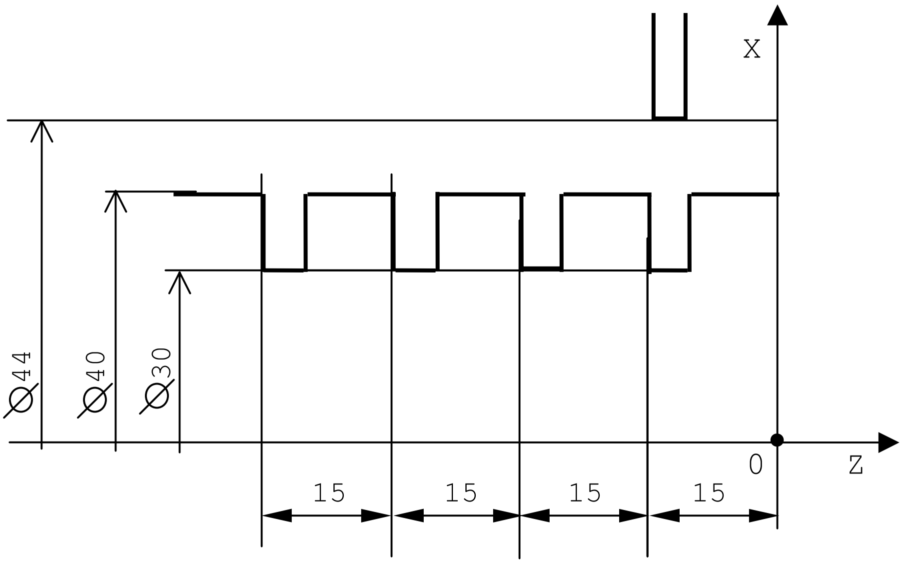
Examples:

N10 (BGT.E1.123,END)	branch to "END" if the value of E1 is greater than 123
N20 (BEQ,SA3,1,LAB1)	branch to "LAB1" if the Boolean variable SA3 is ON
N30 (BNE,E1,E5,START)	branch to "START" if the value of E1 is different from that of E5
N40 (BEQ,SYVAR1.2CH,"OK",LAB1)	branch to "LAB1" if the sequence of characters from SYVAR1 is OK

Notice that, if the variable has character format (CH), the control will check a string of characters, the length of which is indicated by the index preceding CH. If no index is specified, the control defaults it to 1. For example,

(BEQ,SYVAR2.3CH,"ABC",END)	branch to "END" if the 3 characters from SYVAR2 are A B C.

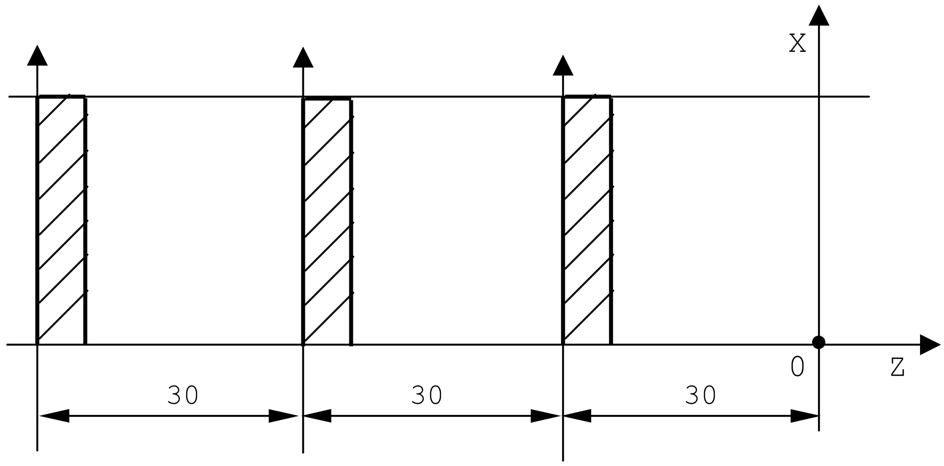
Fig. 5.93. - Example of Repetition




```
..........
N24 GO X44 Z0
(RPT,3)
N25 G91 Z-15
N26 G90 G1 G4 X30 F0.2
N27 GO X44
(ERP)
..........
```

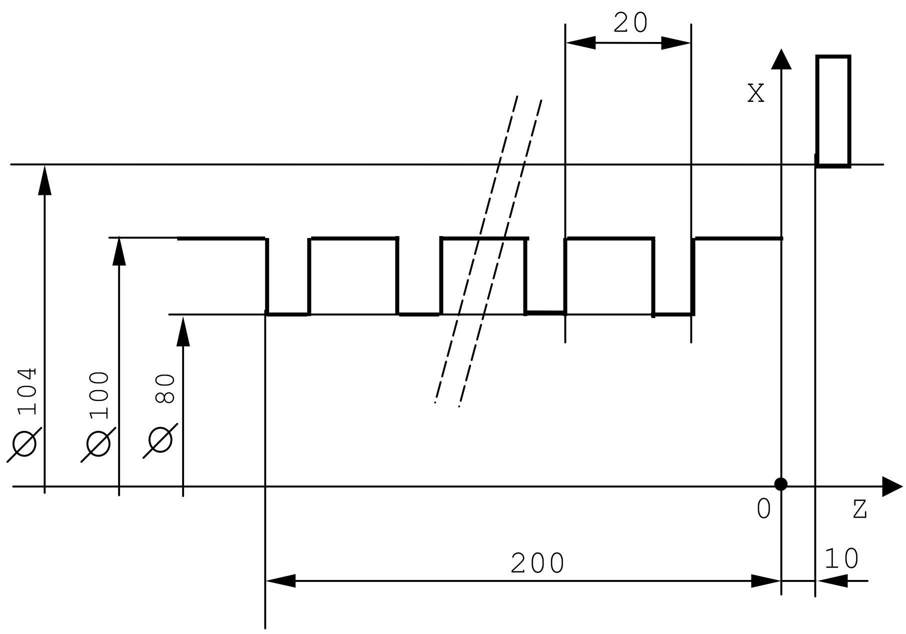
With RPT associated to an incremental origin UIO, you can several equal pieces simply by programming the first piece referred to its own zero point.

Fig. 5.94. - Reusing a program subroutine




```
..........
(RPT,3)
..........          -Complete program for the 1st part,
..........           the parting-off included
(UI0,Z-30)
(ERP)
M30
```

Fig. 5.95. - Using parametric programming and RPT code



Execution of a square head 10 mm deep by 10 1mm cuts

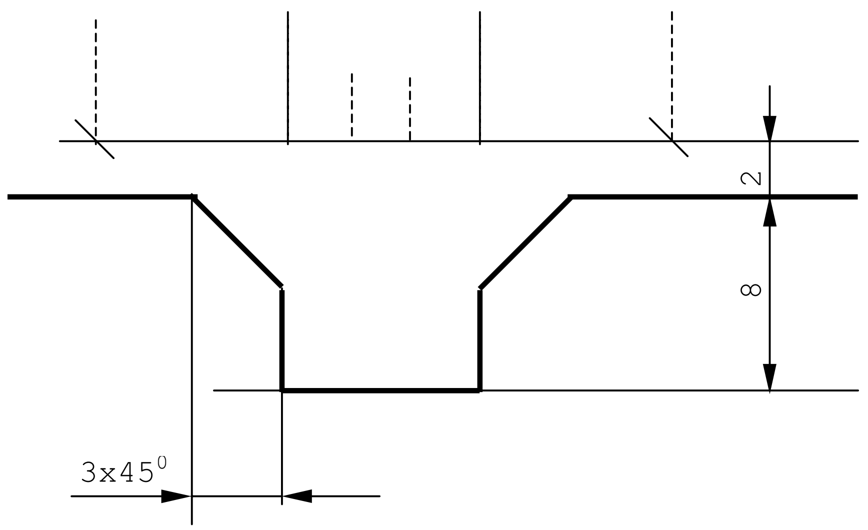
```
..........
N18 ( DIS, "TEST RPT")
N19 G97 S300 T2.2 M6 M3 M8
N20 E31=98
N21 GO X104 Z10
N22 (RPT,10)
N23 G0 XE31
N24 G33 Z-200 K20
N25 G0 X104
N26 Z10
N27 E31=E31-2
N28 (ERP)
..........
```

This cycle can also be programmed using conditional branching instructions, attributing the current diameter to E31, the final diameter to E32 and the cut depth to E33.

```
..........
N18 (DIS,"TEST CONDITIONAL JUMPING")
N19 G97 S300 T2.2 M6 M3 M7
N20 E31=98
N21 E32=80
N22 E33=1.8
N23 G Z10
"CONT" N24 XE31
N25 G33 Z-200 K20
N26 GO X104
N27 Z10
N28 E31=E31-E33
N29 (BGT,E31,E32,CONT)
N30 XE32
N31 G33 Z-200 K20
N32 GO X104
N33 Z..
..........
```

Fig. 5.96. – Subroutine without parameters

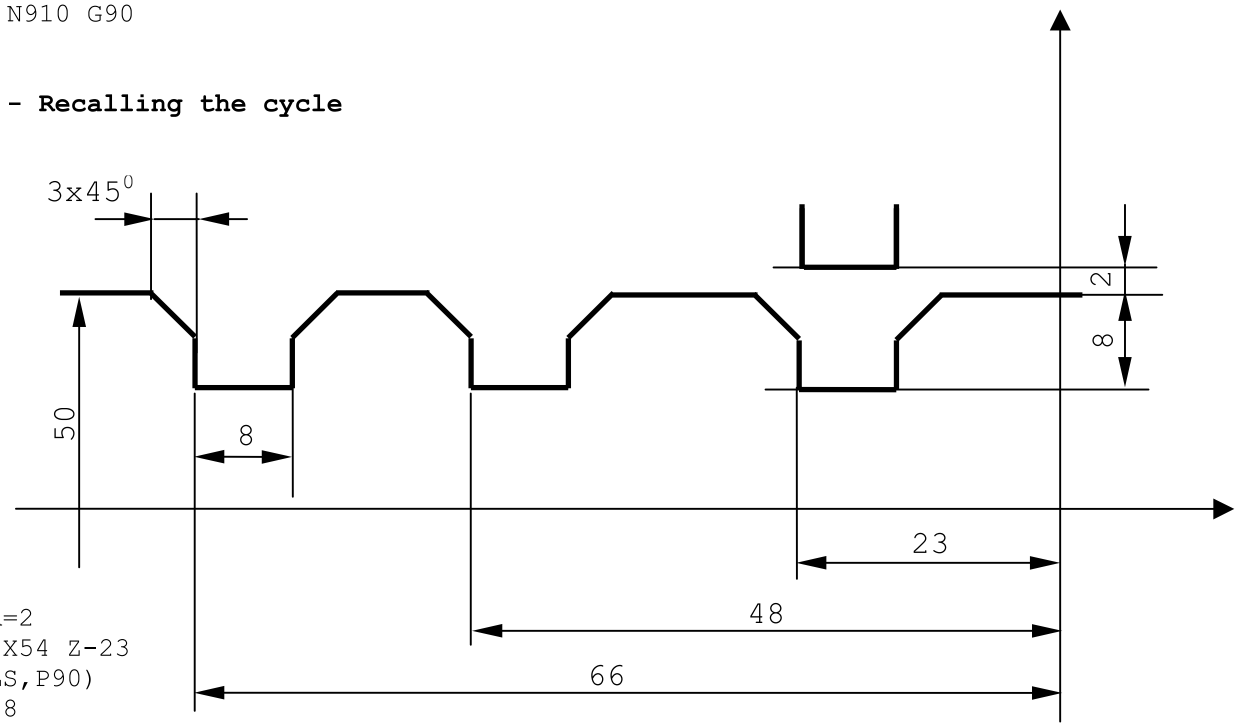
- Defining a grooving cycle



Subroutine P90

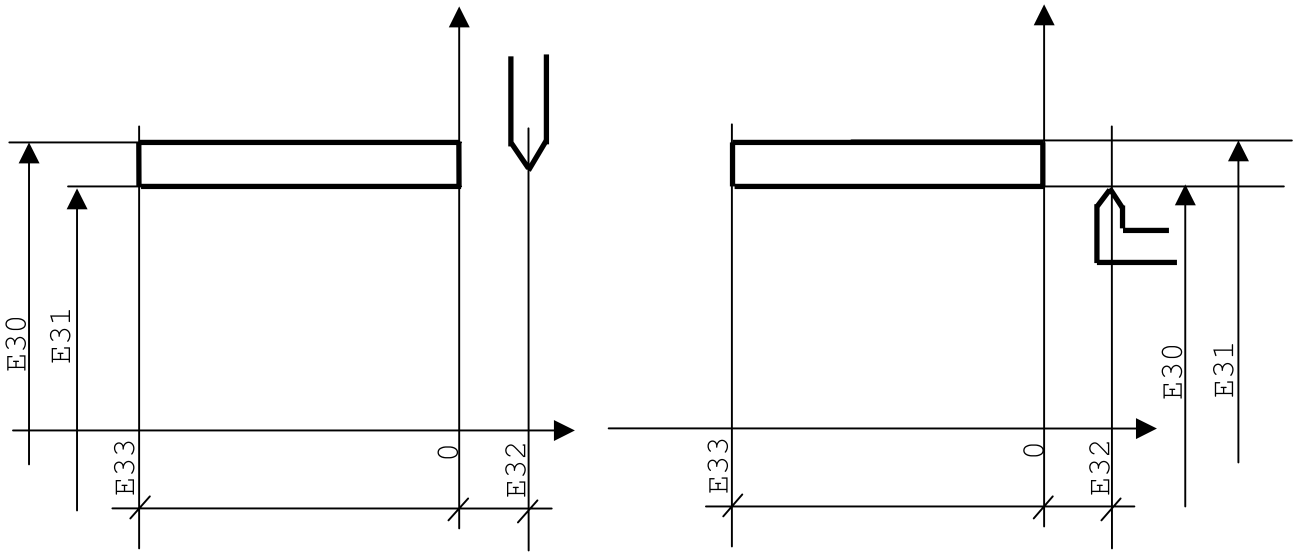
```
N902 G91 G1 G4 X-20 F0.1
N903 G0 X20
N904 Z-5
N905 G1 X-10 Z-5
N906 G0 X10
N907 Z5
N908 G1 X-10 Z-5
N909 G0 X10
N910 G90
```

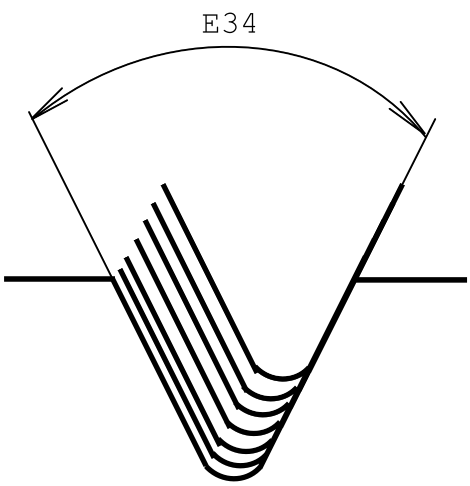
- Recalling the cycle




```
N17 TMR=2
N18 G0 X54 Z-23
N19 (CLS,P90)
N20 Z-48
N21 (CLS,P90)
...............
```

Fig. 5.97. – Subroutine for standard threading





Parameters:
E30 = external diameter
E31 = internal diameter
E32 = starting Z
E33 = ending Z
E34 = tool angle
E35 = number of cuts (negative for internal threading)
E36 = pitch
E37 = safety distance

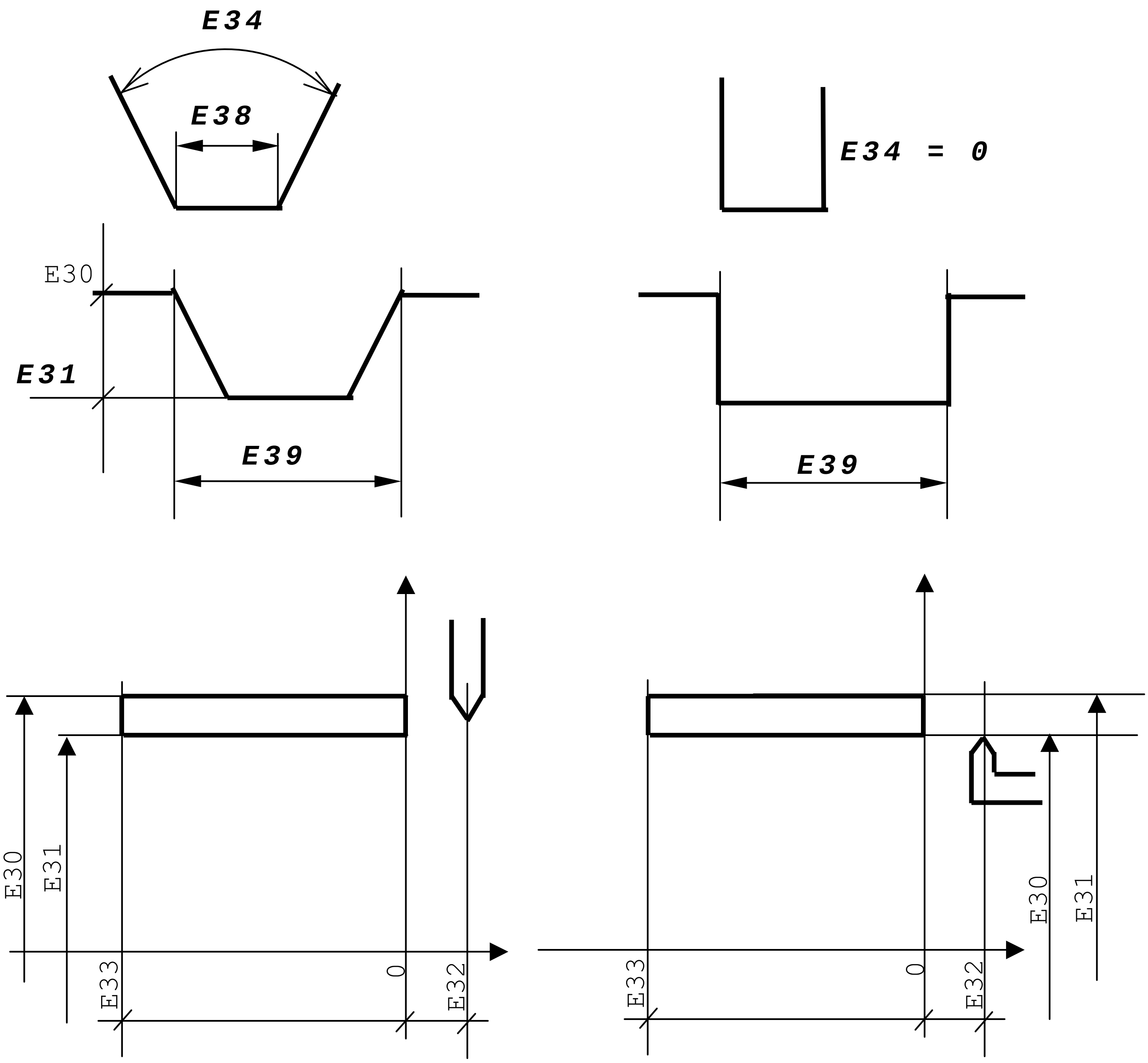
DEFINING THREADING SUBROUTINE

```
E40=(E30-E31/2)
;E47=RETURN DIAMETER
E47=E30+2*E37*E35/ABS(E35)
E44=TAN(E34/2)
E1=ABS(E35)
;E41=DEPTH OF THE 1ST CUT
E41=E40/SQR(E1)
E42=0
(RPT,E1)
E42=E42+1
E43=E41*SQR(E42)
;E45= X REAL   E46 = Z REAL
E45=E30-2*E43
E46=E32-E43*E44
GO ZE46
XE45
G33 ZE33 KE36
GO XE47
(ERP)
GO ZE32
```

RECALLING THREADING SUBROUTINE

```
..........
..........
..........
N24 T3.3 M6
N25 G97 S1200 M3 M7
N26 E30=E40
N27 E31=37.4
N28 E32=3
N29 E33=-30
N30 E34=60
N31 E35=8
N32 E36=2
N33 E37=1
N34 (CLS.THREA)
..........
..........
```

Fig. 5.98. – Parametric Subroutine for Acme or Square Threading



Parameters:
E30 = external diameter
E31 = internal diameter
E32 = starting Z
E33 = ending Z
E34 = tool angle
E35 = number of cuts (negative for internal threading)
E36 = pitch
E37 = safety distance (radial value)
E38 = tool width
E39 = thread width

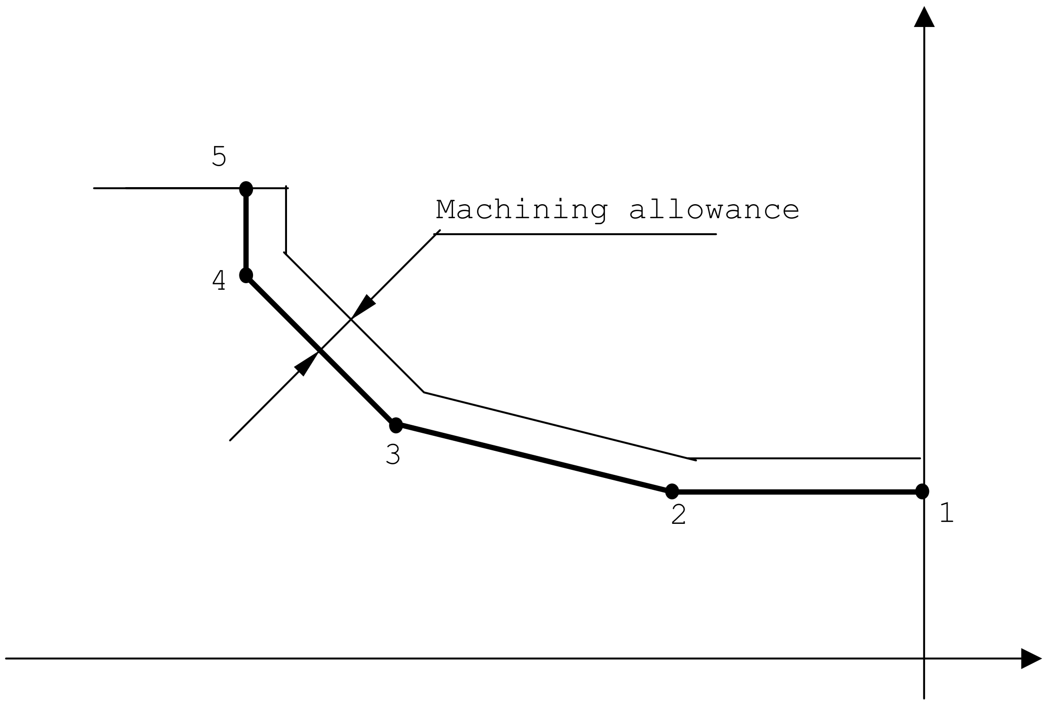
DEFINING SUBROUTINE "TRAPEZ"

```
;E40=THREAD DEPTH
E40=E30-E31
;PARAMETERS CONTROL
E11=(E30-E31)*E35
(BLT,E11,0,ERR)
E45=E39-ABS(E40)*TAN(E34/2)
(BLT,E45,E38,ERR)
E41=E40/ABS(E35)
;E42=RETURN DIAMETER
E42=E30+2*E37*E35/ABS(E35)
E43=ABS(E41)*TAN(E34/2)
E1=ABS(E35)
(RPT,E1)
E30=E30-E41
E39=E39-E43
;E2=No. OF CUTS TO COMPLETE
THE THREAD
E2=INT((E39/E38)-0.001)+1
E44=(E39-E38)/(E2-1)
;E32= REAL STARTING Z
E32=E32+E43/2
;E45= REAL Z
E45=E32
(RPT,E2)
GO ZE45
XE30
G33 ZE33 KE36
GO XE42
E45=E45+E44
(ERP)
(ERP)
GO ZE32
(BNC,END)
"ERR" (DIS,"WRONG PARAMETERS")
MO
"END"
```

CALLING "TRAPEZ"

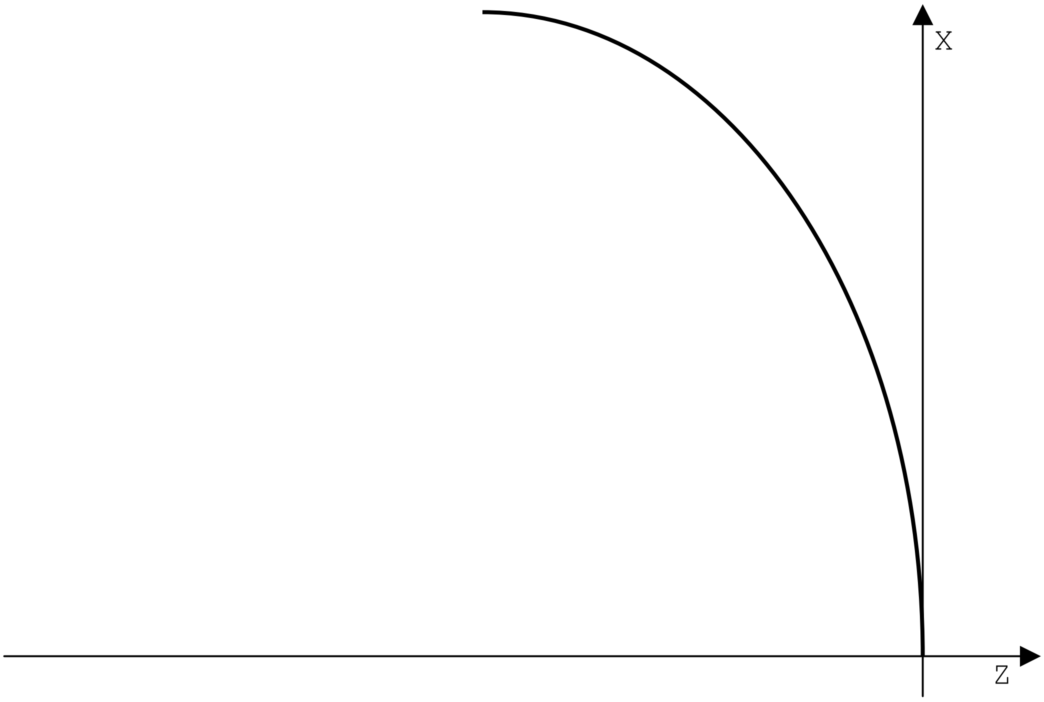
```
..........
..........
N10 (DIS,"ACME THREAD")
N11 T4.4 M6
N12 G97 S300 M3 M7
N13 E30=100
N14 E31=90
N15 E32=5
N16 E33=-100
N17 E34=30
N18 E35=12
N19 E36=15
N20 E37=1
N21 E38=8
N22 E39=15
N23 E40(CLS,TRAPEZ)
..........
..........
```

Fig. 5.99. - Using EPP code for profile roughing and finishing




```
N0 (DIS,"PREFINISHING")
N10 S150 T1.1 M6 M3 M7 F0.3
UOV=O.3
"START" GO G42 X.. Z..  (point 1)
..........
..........
"END" G40 X..  Z..      (point 5)
..........
..........
N70 (DIS,"FINISHING")
N80 S200 T2.2 M6 M3 M7 F0.15
N81 UOV=0
N82 (EPP,START,END)
```

Fig. 5.100. - Using Parametric Programming and Subroutines to machine Parabolic Profiles



Parameters:

E31 = focus distance (twice the focus)
E32 = Z increment
E33 = starting Z
E34 = final Z

Subroutine PARAB

```
G1  G42 X Z E33
"START" E33=E33-E32
(BLT,E33,E34,END)
E35=2*SQR(2*E31*ABS(E33))
XE35 ZE33
(BNC,START)
"END"E35=2*SQR(2*E31*ABS(E34))
G40 XE35 ZE34
E35=E35+10
G XE35
```

MAIN program

```
N.. T1.1 M6 S..  F..
N.. G X Z5
E31=52
E32=2
E33=0
E34=-168.8
(CLS,PARAB)
G Z ...
..........
..........
```

5.12.3. USING MISCELLANEOUS THREE-LETTER COMMANDS

This section deals with three-letter commands used for miscellaneous operations. The following commands belong to this class:

1) DPI - defines the plane of interpolation;
2) DTL - defines the positioning tolerance level;
3) DLO - defines the limits of the operating field;
4) DSA - defines protected areas;
5) CTL - switches the control to the lathe/mill configuration.

DEFINITION OF THE INTERPOLATION PLANE - DPI

The DPI command allows you to define the abscissa and ordinate axes of the interpolation plane.
The allowable format is:

(DPI,VAR-1,VAR-2)

where:

VAR-1 and VAR-2	letters for the abscissa and the ordinate of the interpolation plane. They must be configured axes in the system.

Example:

(DPI,U,V)	specifies the interpolation plane formed by axes U and V.

Notes:

• The two letters for the axes you specify in DPI must be different, i.e. VAR-1≠VAR-2.

• In a DPI command, you cannot use axes that are alternative one another (i.e. functionally equivalent).

• You cannot use the UIP command if any of the following conditions are active:
. GTL (G21);
. cutter compensation (G41-G42);
. standard fixed cycles (G81÷G89);
. continuous operation (G27-G28).

DEFINING THE POSITIONING TOLERANCE LEVEL - DTL

This command allows you to define the tolerance for the positioning of the axes from inside a program.
The allowable format is:

(DTL,VAR-1[,VAR-2...,VAR-n])

where:

VAR-i	is an axis and a dimension. You can program as many as 8 axes. Program the dimension in the measuring unit (G70/G71) active when the DTL command is executed.

For example:

(DTL,Z.1,X.05)	specifies a positioning tolerance of 0.1 for the Z axis and 0.0.5 for the X axis

Notes:

• If you program a value of 0 in DTL, the control defaults to the positioning tolerance specified during system configuration.

• If you do not program a specific axis in the DTL command, the control uses the positioning tolerance that is currently active for that axis.

• You cannot specify the same axis twice in one DTL command.

• The programmed positioning tolerance must not exceed the current SERVO ERROR.

• The DTL command will cause an error if any of the following conditions are active:
. GTL (G21);
. cutter compensation (G41-G42);
. continuous operation (G27-G28).

DEFINING THE OPERATING FIELD - DLO

The DLO command allows you to define the operating field for the axis programmed in the record, with respect to the present applied origin. For not programmed axes, the previously active value is maintained.

If the programmed value exceeds the limit declared in the characterization file, it will be ignored. The control uses the value declared during characterization.

The format is:

(DLO,VAR-1)

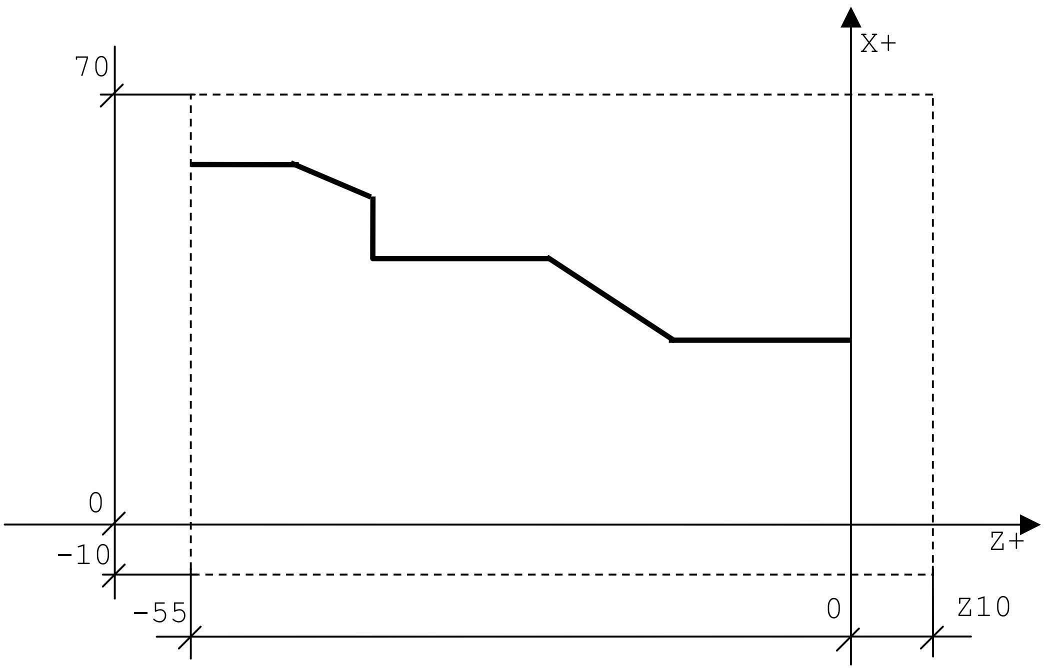
where:

VAR-i is a couple of words of type "axis-dimension", having the same "axis name" each; they represent respectively the upper and lower limit of the operating field with respect to the current origin.

Example (Fig. 5.101.)

Fig. 5.101. - Definition of the Operating Field

```
(DLO,Z-10 X-70)
..........
(DLO,Z-55 X10)
..........
```



Notes:

• You can program as many as 8 different couples of operating limits.

• Program dimensions in the measuring unit (G70/G71) active when DLO is executed.

SWITCHING BETWEEN LATHE AND MILL CONFIGURATIONS - CTL

The CTL command allows you to select the mill configuration when both the mill and lathe options have been established in the PGCFIL.

After a CTL, the control features all the characteristics of a mill (refer "8600MC USER'S MANUAL").
The allowable formats are:

(CTL,T) activates the MILL configuration

or

(CTL,O) returns the control to the original configuration

PROTECTED AREAS - DSA, ASC, DSC

These commands allow you to activate or deactivate protected areas, i.e. areas the axes are not allowed to enter.
The commands in this class are:

1) DSA - defines a protected area;
2) ASC - activates a protected area;
3) DSC - deactivates a protected area.

The control checks for protected areas before starting the movement. From inside a program you can define up to 3 protected areas referred to the active origin.
The allowable formats are:

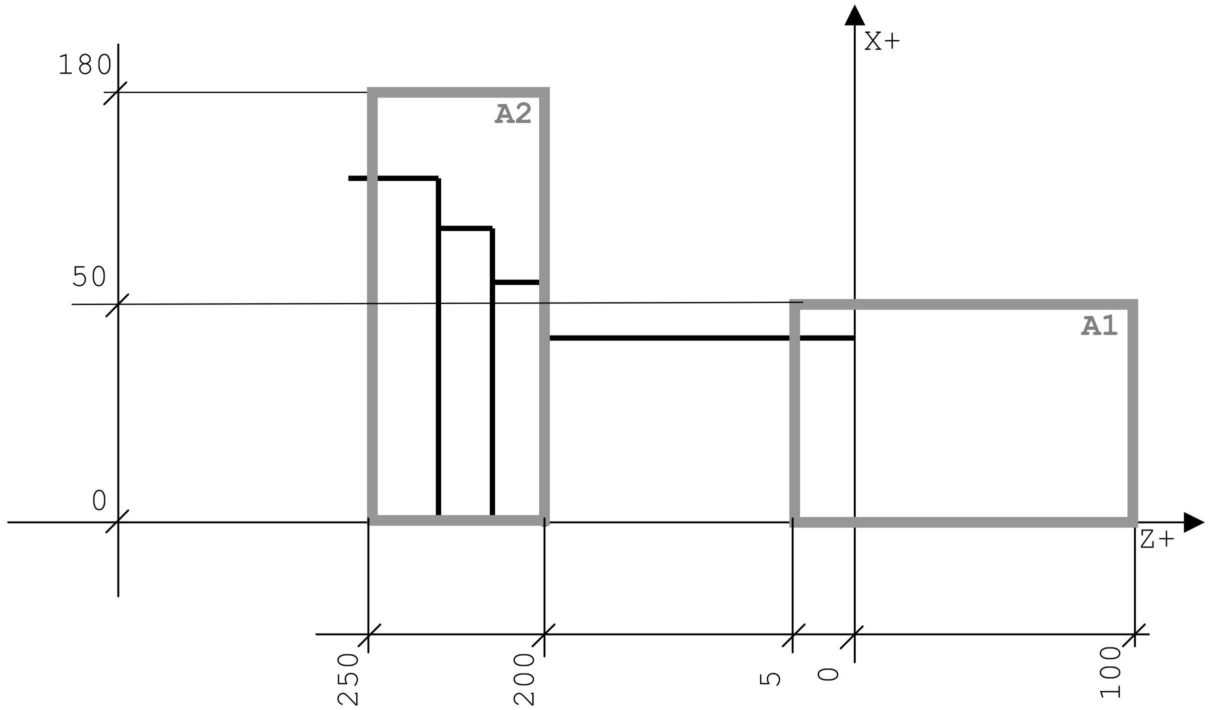
(DSA,n,Z- Z+, X- X+)
(ASC,n)
(DSC,n)

where:

n number of the area
Z- more negative Z
Z+ more positive Z
X- more negative X
X+ more positive X

Fig. 5.102. shows an application of protected area programming.

Fig. 5.102. – Example of Protected Area Programming




```
N1 (DSA,1,Z-5 Z100,XO X50)
N2 (DSA,2,Z250 Z-200,X0 X180)
N3 (ASC,1)
N4 (ASC,2)
N5 T1.1  M6
..........
N80 (DSC,1)
..........
N99 M30
```

5.12.4. USING I/O COMMANDS

These commands allow you to perform I/O operations from inside a program. The commands in this class are:

1) DIS - display a variable;
2) DLY - establish a delay.

DISPLAY A VARIABLE - DIS

The DIS command allows you to display the value of the specified variable. The desired value appears in the screen area reserved for communicating with the operator.

The format is:

(DIS,var)

where:

var can be:

. any code used in assignment blocks for global system variables, GTL elements or parameters. The control will display the sequence VARIABLE NAME=value. For example, (DIS,E27);

. a message for the operator.
The message can be up to 32 characters long. Program the message in quotes in the DIS command. For example, (DIS."THIS IS AN EXAMPLE");

. a numerical constant. For example, (DIS,100).

Examples:

(DIS,"c1=",c1)	displays the coordinates of the center and radius of the circle
(DIS,"l2=",l2)	displays the distance between the origin and the straight line and the angle formed by the straight line and the abscissa axis.

ESTABLISH A DELAY - DLY

The DLY command allows you to program a delay interval of the specified duration.

The allowable format is:

(DLY,time)

where:

time — is the duration of the delay interval in seconds. It can be up to 32 seconds long. You can program it as a numerical value or longreal E parameter (E30 to En).

Example:

```
(DLY,2)

E48=2

(DLY,E48)
```

Important. When using this command, you need synchronisation (refer to section 5.16.).

5.12.5. MANAGING THE GRAPHIC DISPLAY

This class of commands allows you to control the graphic display from inside a part program. The commands in this class are:

1) UCG - define the graphic display field;
2) CLG - clear the graphic display;
3) DCG - disable the graphic display.

DEFINING THE GRAPHIC DISPLAY FIELD - UCG

This command initialises the graphic display and establishes the limits, the scale and the mode of the display.
The allowable format is:

(UCG,n[,AXIS1I AXIS1S,AXIS2I AXIS2S])

where:

n	defines the display mode: . n = 1 display not coordinated with axis movement; . n = 2 display coordinated with axis movement. n can be programmed either explicitly or implicitly (with an E parameter of the byte type)
AXIS1I	defines the axis and dimension for the lower limit of the abscissa axis on the display
AXIS1S	defines the axis and dimension for the upper limit of the abscissa axis on the display
AXIS2I	defines the axis and dimension for the lower limit of the ordinate axis on the display
AXIS2S	defines the axis and dimension for the upper limit of the ordinate axis on the display

For example:

(UCG,2,X100 Z150,X50 X250,Z)

Activates graphic display not coordinated with the axes movement (n=2). The graphic display shows moves between Z100 and Z150 for the abscissa axis and between X50 and X250 for the ordinate axis with respect to the current absolute origin.

Notes:

• With the UCG command, the control performs graphic calculations regardless of which EDP or graphic screen is active and displayed to the operator.
However, since graphic calculations can determine unwanted dwells between part program blocks, we recommend that production runs be performed with graphic display enabled only when strictly necessary.

• When programming an UCG block, use synchronisation (refer to section 5.16).

CLEARING THE GRAPHIC DISPLAY - CLG

This command clears the currently displayed profile from the screen, leaving the coordinates system.
The allowable format is:

(CLG)

DISABLING THE GRAPHIC DISPLAY - DCG

This command disables the graphic display and removes the displayed profile as well as the coordinate system from the screen.
The allowable format is:

(DCG)

5.12.6. MANAGING TOOL OFFSETS

This class of commands allows you to manage tool offset values from inside a part program. At present, the only command in this class is:

RQU - requalifies tool offset

REQUALIFYING TOOL OFFSETS - RQU

The RQU command requalifies (updates and modifies) a specific tool offset according to the programmed length and dimensions.
The allowable format is:

(RQU,n.tool,n.offset,Z..,X...)

where:

n.tool	is the number of the tool. Program it directly or as an integer type E parameter (E10÷E19).
n.offset	is the number of the offset to requalify. You can program it directly or as an integer type E parameter. The offset number is a value between 1 and 9999. The upper limit depends on the number of records defined in the offset file.
Z	defines the increment of length to apply to the Z axis offset. You can program this value either directly (decimal number) or as a long real E parameter (E30÷En). If you program 0, nothing is added to the current length value of the offset.
X	defines the increment of length to apply to the X axis offset. You can program this value either directly (decimal number) or as a long real E parameter (E30÷En). If you program 0, nothing is added to the current length value of the offset.

For example:

```
..........
(RQU,10,1.ZE40.XE41)    -requalifies tool 10, offset 1.
..........              The Z increment is contained in E40.
                        The X increment is contained in E41.
```

Notes:

• If the offset file has been created to manage current (c) and maximum (m) offset values, the RQU command will update the current (c) values. If you do not want to update the current values, program RQP with the same format.

• The length values are added to the values in the tool offset file according to the unit (mm or in) of the offset.

• You must program increments in the RQP command with the measuring unit configured for the system (G70/G71). No scale factor (SCF) is applied to them.

• If you requalify an offset for a diameter axis, the control will divide the values you specify by 2 before adding them to the offset file values.

• If the incremental offset value is not declared in the probing cycles, the RQU command requires a synchronisation indicator (#).

5.12.7. MANAGING A PROBE

The probe allows you to execute measure cycles on the piece in order to update the length offset file.

PROBING G FUNCTIONS

To define measuring cycles you can use functions G72 and G74.

G72 measures the point coordinates with a linear interpolation and stores the value in the E parameters defined for the cycle. The allowable format is:

G72 axis En

where:

axis	is the axis to be moved
En	is the parameter in which the measured value must be stored

Example:

G72 X100.2 E31	the measured value is stored in E31

G74 measures the variance with respect to the theoretical point and stores it in the E parameter. The allowable format is:

G74 axis En

where:

axis	is the axis to be moved to the theoretical coordinate
En	is the parameter in which the measured variance must be stored

Example:

G74 X100.2 E31	the variance between the nominal and the real coordinate is stored in E31.

DEFINING THE PROBING PARAMETERS - DPT

The DPT command allows you to define probing parameters from the keyboard or in a program. The parameters you must define are:

- approach dimension (Qa) expressed in mm;
- safety dimension (Qs) expressed in mm;
- measuring speed (Vm) expressed in mm/min.

The format of the DPT command is:

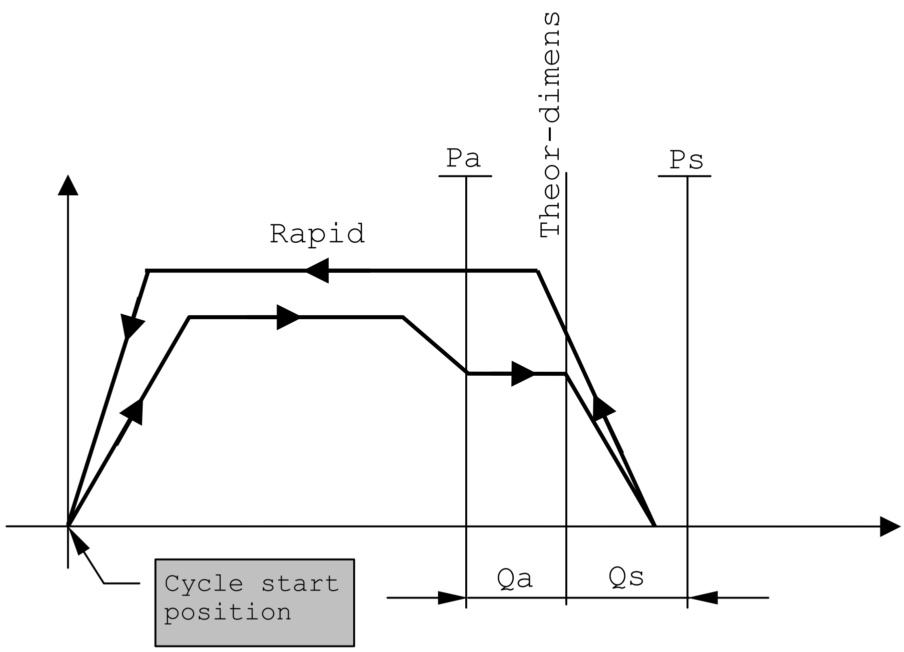
DPT,Qa,Qs,Vm	from the program
(DPT,Qa,Qs,Vm)	from the keyboard

Example:

(DPT,10,12,1000)	from the keyboard

When the control executes a probing cycle, it executes the following sequence of moves (Fig. 5.103.).

Fig. 5.103. - Moves and Dimensions of a Probing Cycle



1. Move at rapid to the approach point (Ps).

2. Move at measuring speed up to the measuring point or the safety distance, then store the dimensions.

3. Return at rapid to the start position of the probing cycle.

Important. If the probe does not trigger before reaching the safety point, the probe returns to the start point of the cycle and the following message is displayed: NO PROBE CONTACT.

5.12.8. MANAGING TOOL OFFSETS

DECLARING A TOOL OUT OF USE - TOF

To declare a tool out of use, program the following format:

(TOF,n)

where:

n — is the number of the tool to be declared out of use. It can be either a number or an E parameter (E10÷E19).

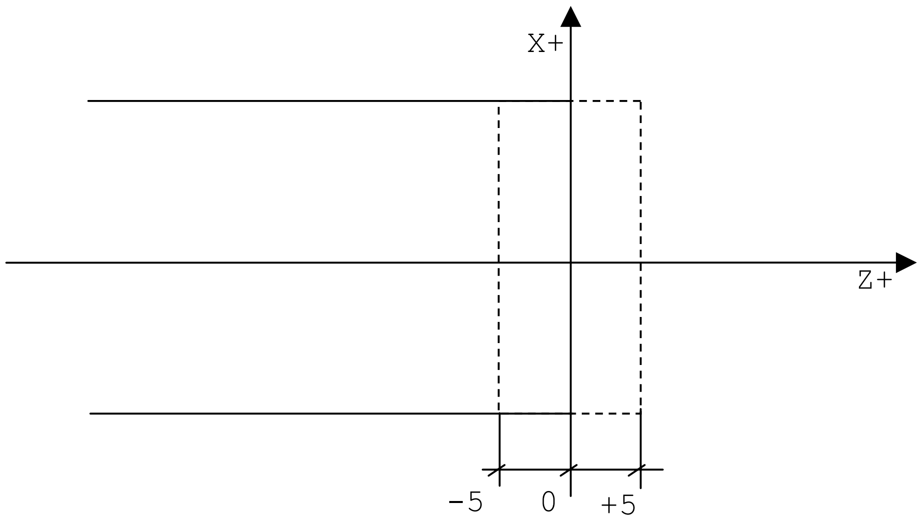
Example:

```
(TOF,22)
(TOF,E11)
```

PROBING TO DEFINE ZERO ON A WORKPIECE

Fig. 5.104. - Probing for the Rough Turn

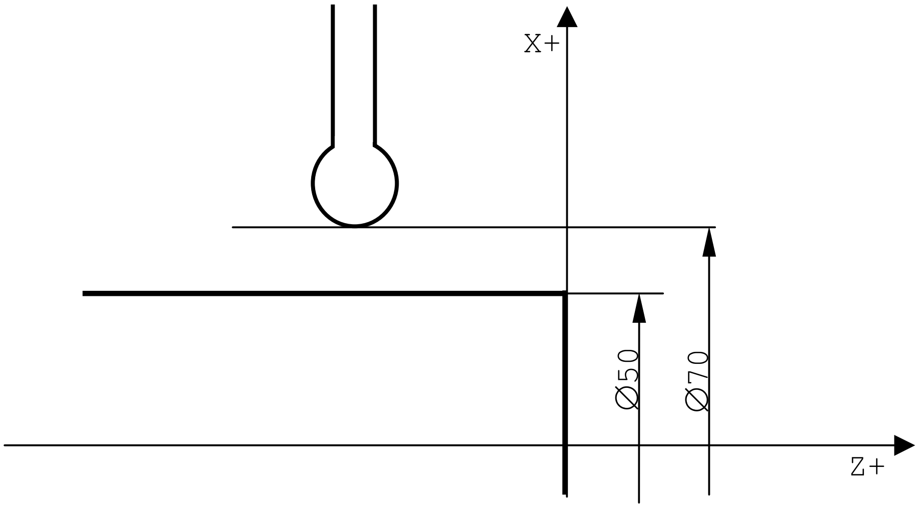
In the example, the difference between nominal and real workpiece zero can be c 5 mm.




```
N21 T1.1 M6
N22 (DPT,6,6,300)
N23 G X0 Z10
N24 G74 Z0 E30
N25 E31=ABS(E30)
N26 (BGT,E31,5,K0)
N27 (UOT,0,ZE30)
N28 T2.2 M6
..........
N44 T3.3 M6
..........
"KO"(DIS,"PIECE OUT OF TOLERANCE")
..........
..........
```

EXAMPLE OF TOOL OFFSET REAPPLYING

Fig. 5.105. - Tool offset Reapplying




```
N1 T1.1 M6
N2 G X70 Z-20
N3 G74 X50 E30
N4 (RQU,4,4,XE30)
N5 ..........
.............
```

<u>Example of control of the displacement before reapplying</u>

```
N1 T1.1 M6
N2 G X70 Z-20
N3 G74 X50 E30
N4 E31=ABS(E30)
N5 (BLT,E31,0.04,OK)
N6 (TOF,4)
N7 (BNC,KO)
"OK"(RQU,4,4,XE30)
..........
..........
"KO"(DIS,"TOOL OUT OF USE")
..........
```

5.13. COORDINATED SPINDLE AXIS

By programming the M function provided by the machine logic, e.g. M21, the system controls the spindle axis as an axis coordinated with X and Z axes.

The spindle axis is identified by the C function espressed in degrees and programmable in absolute or incremental mode from ± 0.0001 to ± 9999.9999.

Axes motion can take place at rapid or machining rate, with circular or linear interpolation. The F machining rate is expressed in degrees/minute and must be programmed when G94 is present.

Functions S (to be programmed in G97), M3, M4 and M5 are attributed to the motorised tool fitted on the turret.

To deactivate M21, use the **RESET** key or another M function provided by the machine logic. For example, M21.

The spindle enable/disable functions must be programmed in a separate block, when GO is present.

Example:

```
..........
N24 M21
N25 G97 S2000 M3 M8
N26 GO C90 X100 Z-100
N27 G1  Z-105 G94 F100
N28 C180 F150
N29 G Z
N30 M20
..........
```

By combining the motions of the X, Z and C axes you can obtain spirals and cylindrical or conical helixes.

Since working speed F has been programmed in degrees/minute, you must calculate its value according to the diameter and the speed in mm/minute to be obtained on the workpiece.

To execute a milling cycle along a circumference on a D diameter, calculate the angular speed with the following formula:

$$F = \frac{AV}{D} \cdot \frac{360}{\P} = AV \cdot \frac{114,64}{D}$$

where:

F - angular speed in degrees/min;
AV - speed along the circumference in mm/min;
D - diameter of circumference in mm.

If the rotary C axis moves together with the longitudinal W axis, a cylindrical helix is obtained. In this case, calculate the angular speed with the following formula:

$$F = \frac{AV \cdot \sqrt{dZ^2 + dC^2}}{\sqrt{dZ^2 + (\pi D/360 \cdot dC)^2}}$$

where:

F - angular speed in degrees/min;
AV - speed along the helix in mm/min;
dZ - actual displacement along the Z axis in mm;
dC - actual displacement along the C axis in degrees.

5.14. VIRTUAL AXES

With the NC-110, the machining of polygonal profiles –made up of straight lines and circles-- is performed by introducing the concept of virtual axes.

The programmer programs the profile on a virtual cartesian planne u-v, as if the workpiece had to be milled on a milling machine. The NC-110 transforms the profile into motions of the real C, X axes and, with the suitable modifications, handles and modifies the F of the virtual profile.

To enable virtual axes management, program the UAV code, followed by a number defining the type of transformation. The allowable modalities are 1 (to transform cartesian coordinates into polar coordinates) and 2 (to transform cartesian coordinates into cylindrical coordinates).

Virtual axes management can only be activated if the spindle axis is coordinated and positioned to 0.

To deactivate virtual axes, program **(UAV,O)**.

Modality 1

To program the transformation of cartesian coordinates into polar coordinates, use the following format:

(UAV,1,XC,UV,r)

where:

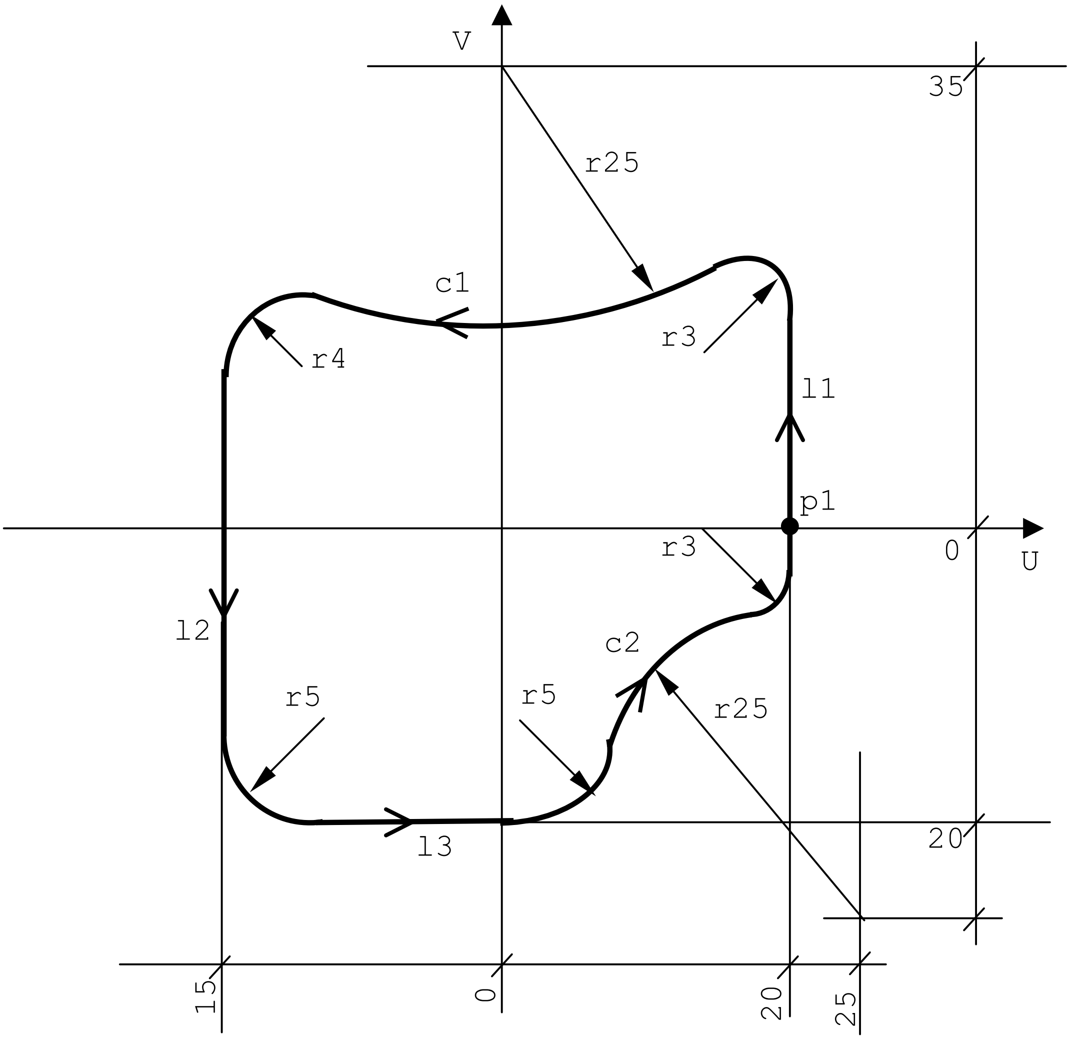
XC are the real axes;
UV are the virtual axes;
r is the minimum radius.

The minimum radius must be compatible with the programmed F, to prevent the rotation speed of the C axis from exceeding the rapid. To calculate the minimum radius, use this formula:

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

where:
r - minimum radius;
F - feedrate in mm/min;
VCmax - rapid of the C axis.

Fig. 5.106. - Example of Modality 1




```
N1   T1.1 M6
N2   (DIS,"PROFILE MILLING")
N3   G X.. Z..
N4   M21
N5   GO C0 X80 S2000 M3 M8
N6   (UAV,1,XC,UV,10)
N7   (DPI,U,V)
N8   p1=U20 VO
N9   l1=p1,a90
N10  c1=IO J35 r-25
N11  L2=U-15 VO,a-90
N12  L3=U0 V-20,a0
N13  c2=I25 J-30 r-25
N14  G21 G42 p1 G94 F300
N15  l1
N16  r3
N17  c1
N18  r4
N19  l2
N20  r5
N21  l3
N22  r5
N23  c2
N24 r3
N25 L1
N26 G40 G20 p1
N27 (UAV,0)
N28 G X80
N29 M20
..........
..........
N99 M30
```

Modality 2

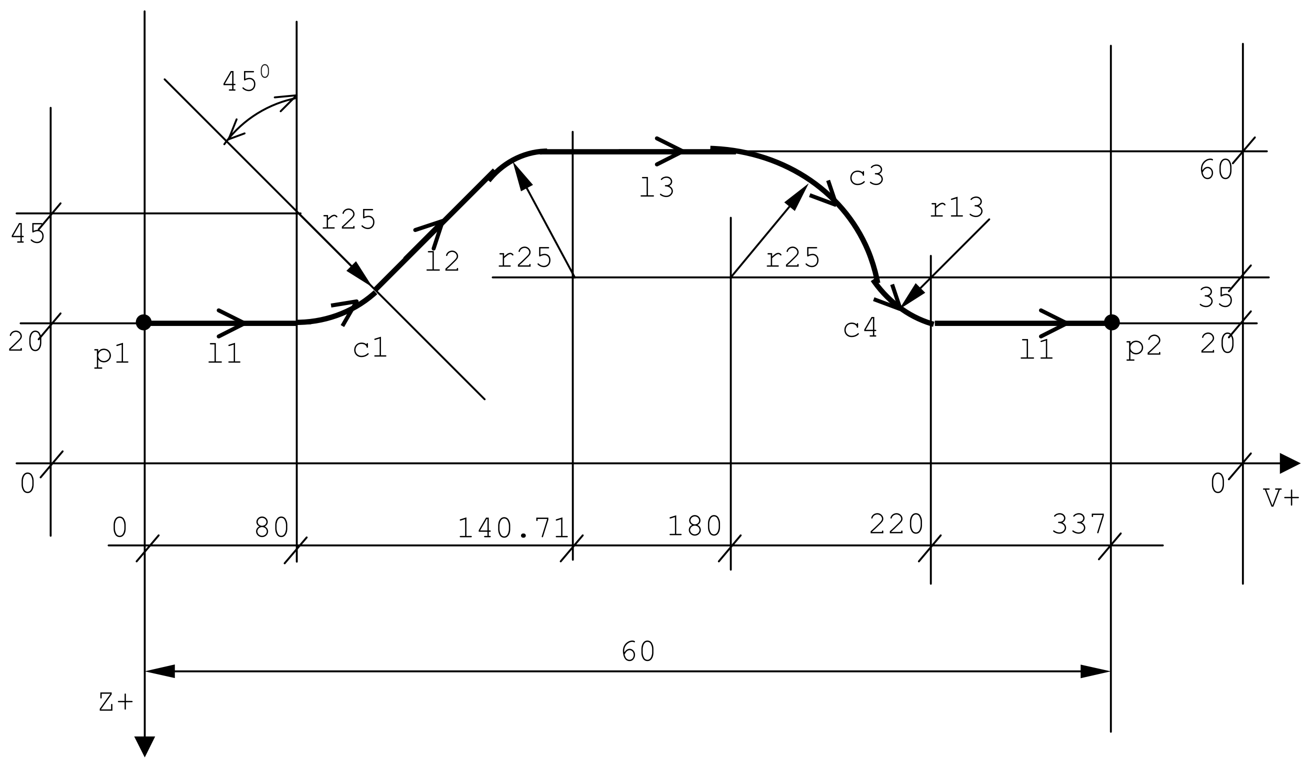
To program the conversion of cartesian coordinates into cylindrical coordinates, use the following format:

(UAV,2,C,V,n)

where:

C V are the cylindrical coordinates
n is the radius of the cylinder

Fig. 5.107. - Example of Modality 2




```
N1   ("DIS", EXAMPLE)
N2   T1,1M6
N3   GX120 Z-20
N4   M21
N5   GO C0 S... M...
N6   E30=60
N7   (UAV,2,C,V,E30)
N8   (DPI,V,Z)
N9   P1=V0 Z-20
N10  E31=(2*3.1415)*E30
N11  P2 = VE31 Z-20
N12  L1 = P1, P2
N13  C1  = I-45 J 80 r 25
N14  C2 = I-35 J 140.71 r-25
N15  C3 = i-35 J 180 r-25
N16  L2 = C1, C2
N17  L3 = C2, C3
N18  C4 = C3,L1,r15
N19  G21 = G41 P1
N20  L1
N21  C1
N22  L2
N23  C2
N24  L3
N25  C3
N26  C4
N27  L1
N28  G20 G40 P2
N29  (UAV,0)
N30  G X 130
```

5.15. CODES FOR MACHINING CYCLE PROGRAMMING

This group includes the following macroinstructions:

1) TGL - grooving cycle;
2) FIL - threading cycle;
3) DFP - profile definition;
4) SPA - paraxial roughing without prefinishing;
5) SPF - paraxial roughing with prefinishing;
6) SPP - roughing parallel to the profile;
7) CLP - profile finishing.

GROOVING CYCLE

This cycle generates either an external or an internal groove, parallel to the axes X or Z.

To obtain a groove parallel to the Z axis, use this format:

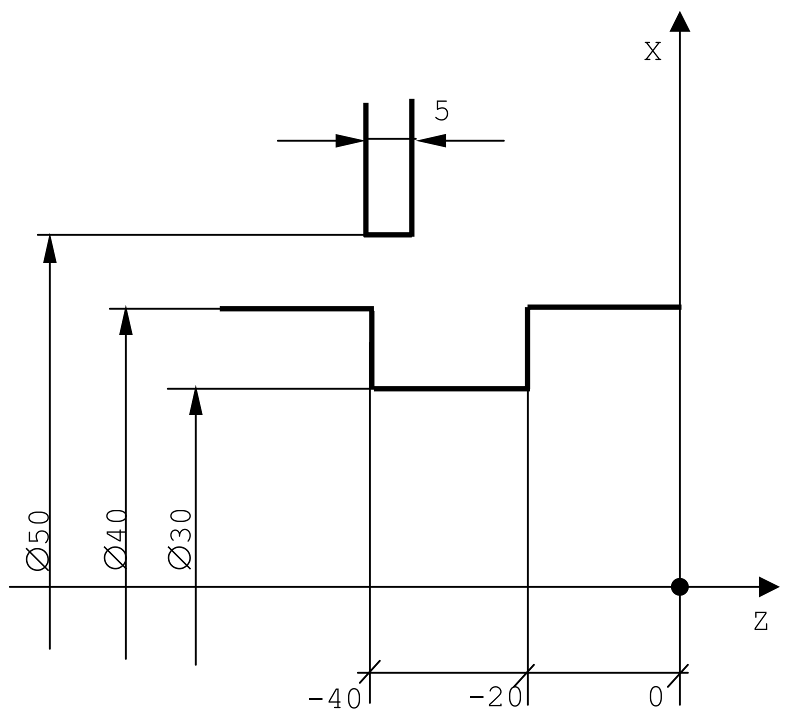
(TGL,Z value,X value,K value)

where:

Z   is the final groove dimension;
X   is the internal diameter;
K   is the tool width.

TGL must be preceded by a move block with G0/G1 on the cycle starting point.

Fig. 5.108. - Example of Groove Parallel to the Z Axis




```
N1 T1.1 M6 S.. F..
N2 G X50 Z-40
N3 TMR=2
N4 (TGL,Z-20,X30,K5)
N5 G X.. Z..
```

To program a groove parallel to the X axis, use the following format:

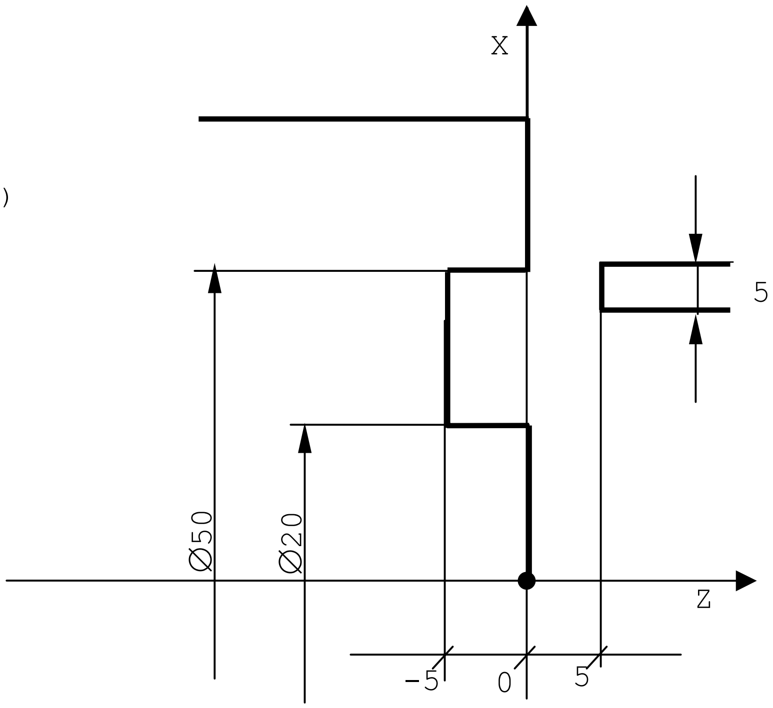
(TGL,X value,Z value,K value)

where:

X   is the final groove dimension;
Z   is the internal groove dimension;
K   is the tool width.

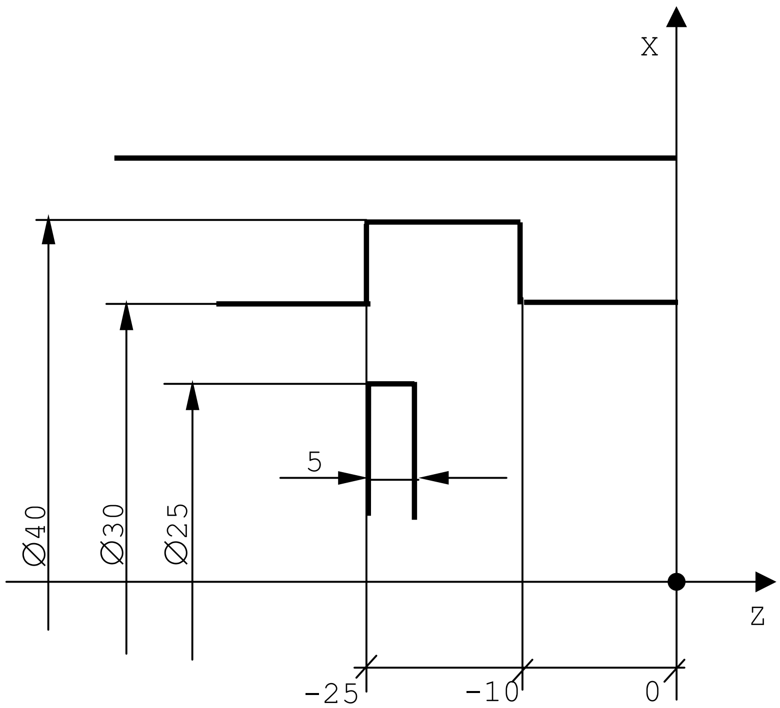
Fig. 5.109. - Example of Groove Parallel to the X Axis

```
N1 T1.1 M6
N2 G X20 Z5
N3 TMR=2
N4 (TGL,X50,Z-5,K5)
```



The control automatically establishes a pause at the end of the groove. The duration of the pause is defined by the TMR code. To disable the automatic pause, program TMR=0 before starting the grooving cycle.

Fig. 5.110. - Internal Groove




```
N1 T1.1 M6 S... F...
N2 G X25
N3 Z-25
N4 TMR=2
N5 (TGL,Z-10,X40,K5)
N6 G X... Z...
```

At groove end, the tool returns to the starting point specified in the previous block.

THREADING CYCLE

The threading cycle allows you to program in a single block a cylindrical or conical threading to be performed at several passes. The allowable format is:

(FIL,Z..,X..,K..,L..,R..,T..,P..,a..,b..)

where:

Z..	final Z dimension;
X..	final X dimension;
K..	pitch;
L..	number of roughing and finishing passes, i.e. L11.2;
R..	distance between the tool and the piece (by default, R=1);
T..	three-figure code defining the type of threading (by default, T000):

Figure 1: 0 = threading with final groove
1 = threading without final groove

Figure 2: 0 = external threading
1 = internal threading

Figure 3: 0 = metric threading
1 = witworth threading
2 = non-standard threading with depth and angle defined by parameters a and b

P..	number of principles (by default, P = 1);	
a..	thread angle	only for non-standard
b..	thread depth	threading

Notes:

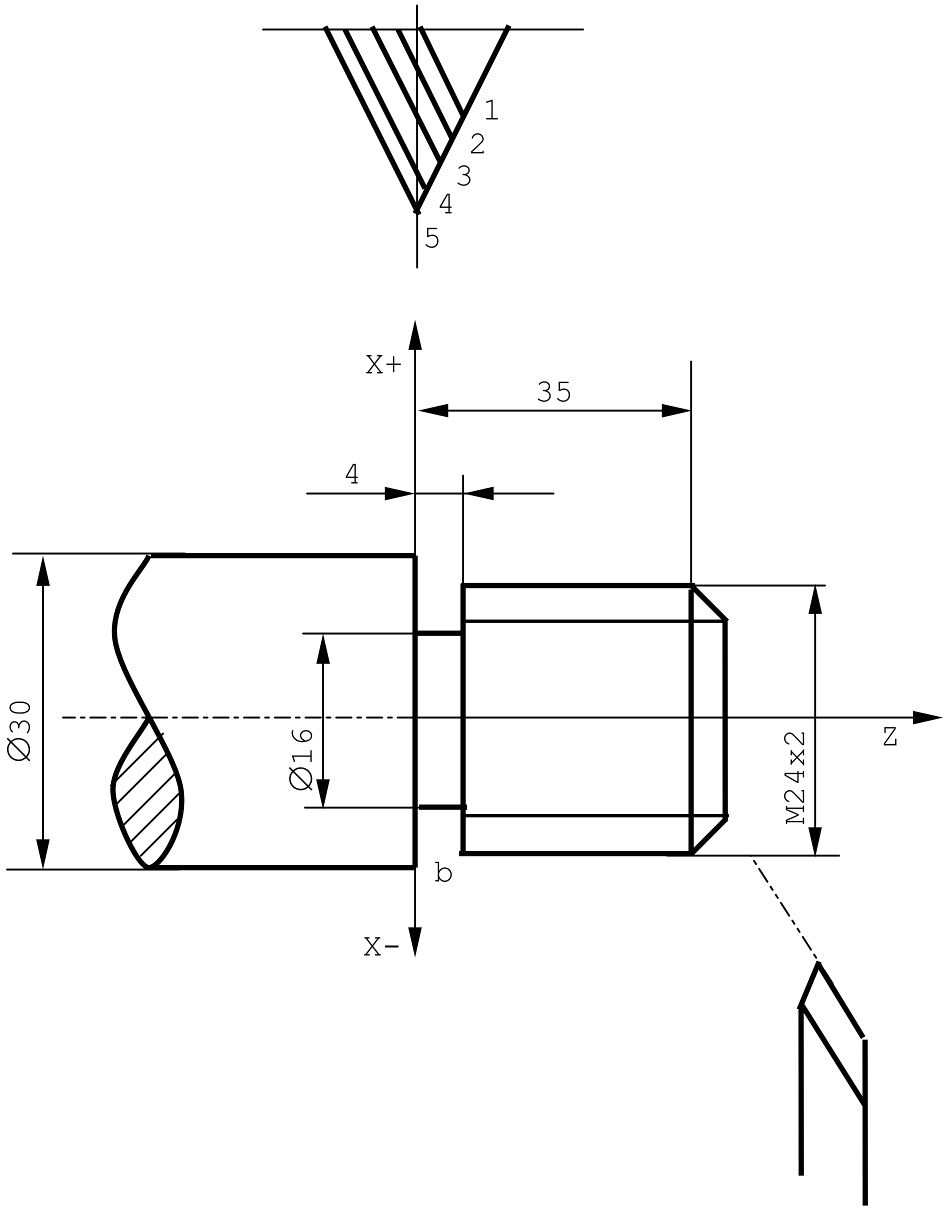
• The control automatically calculates the positions by sliding along the thread side, so that the section of the resulting chip remains constant. For threads with several principles, you must only specify the pitch of each spire. The control executes each pass for every principle before performing the subsequent pass. Each principle is executed without displacing the starting point of each thread, but at a distance from the angle zero of the spindle.

• For threads with final groove, you must program a theoretical final Z, since the fixed cycle provides a stroke increase equal to half the pitch.

• In threads without final groove, the tool reaches the programmed dimension and then moves away with conical threading along the return diameter.

A thread without final groove cannot be obtained in SEMIAUTO.

Fig. 5.111. - Example of Threading Cycle




```
N35 T5.5 M6
N36 GO G97 X24 Z37 S250 M3 M8
N37 (FIL,Z4,K2,L5.1,R2)
N38 GO X250 Z215
```

DEFINITION OF A PROFILE - DFP

This code allows you to define and store as many as 8 profiles. Inside each profile you can define as many as 16 elements, with ISO or GTL standards.

The stored profiles can be recalled from roughing or finishing cycles.

Examples of profile definition with ISO standard:

```
(DFP,1)
G1  X40 Z
Z-2O
X60 Z-30
Z-50
X80
(EPF)
```

```
(DFP,1)
G21 p1
l1
r-5
c1
l2
r-3
l3
G20 p2
(EPF)
```

When describing a profile, consider that:

- With ISO standard, all the profile blocks must contain contouring codes (G1, G2, G3). The rapid code G0 can only appear on the first block.

- Although F functions can be programmed inside the profile, they will only be activated during the profile finishing cycle.

- DFP must always precede the corresponding machining cycle.

- Description errors are signalled only during the execution of machining cycles.

- The block number inside a DFP cycle will only be displayed during the execution of the finishing cycle (GLP). In all the other cycles (roughing parallel to the X or Z axis, etc.), the CRT displays the block that contains the macroinstructions recalling the profile defined by DFP.

- To apply tool radius compensation, program G40/G41/G42 inside the DFP cycle. For example:

```
(DFP,1)
G42 G1 X100 Z0
Z-50
..........
..........
G40 X200
(EPF)
```

The control will automatically apply the compensation during both roughing and finishing cycles.

PARAXIAL ROUGHING WITHOUT PREFINISHING

To program profile roughing parallel to the X axis, use the following format:

(SPA,X,n,L..,X..,Z..)

To program profile roughing parallel to the Z axis, use the following format:

(SPA,Z,n,L..,X..,Z..)

where:

n	is the number of the profile previously store with DFP. It is mandatory and can vary from 1 to 8.
X	is the radial stock allowance on X.
Z	is the stock allowance on Z.
L	is the number of roughing passes. It can vary from 1 to 255.

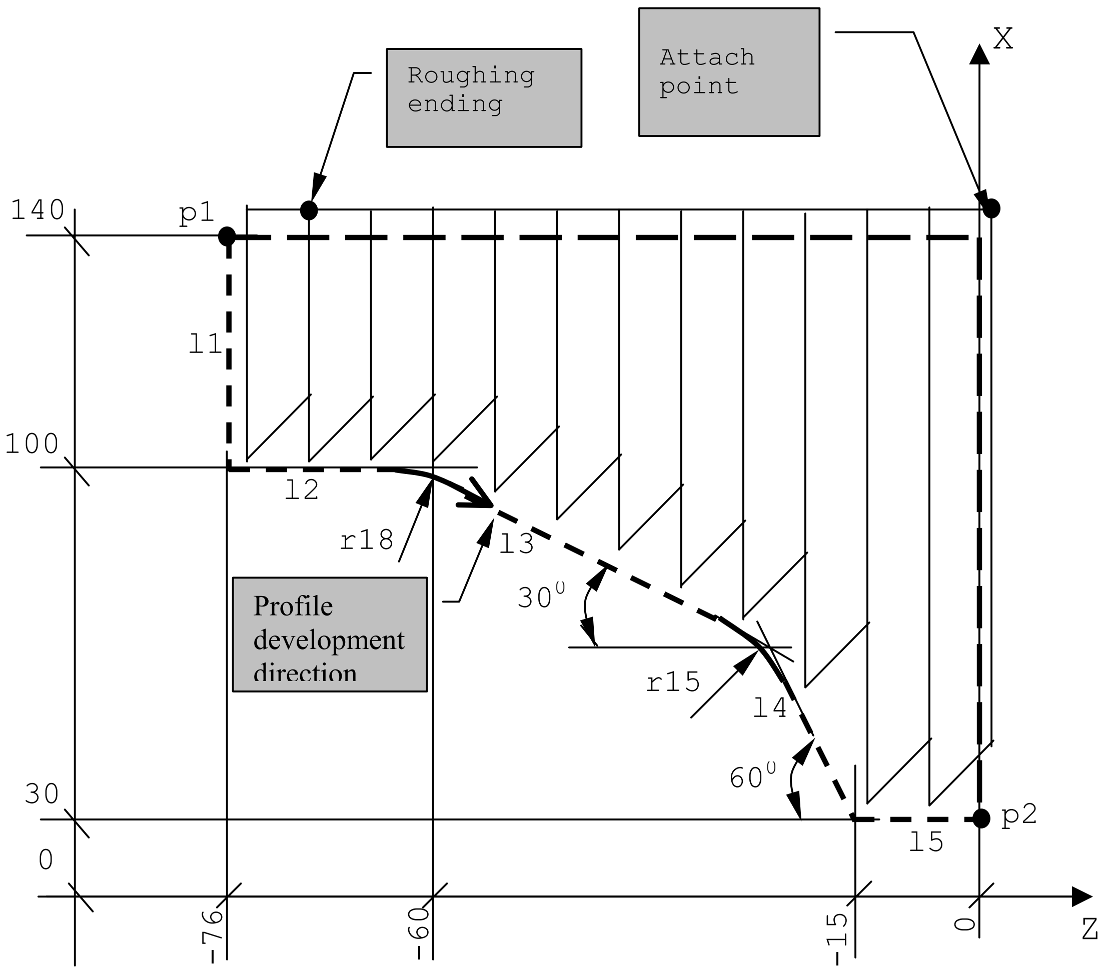
X and Z can be omitted. If present, they must always have a positive value.

From the clamping point and the profile direction, the control automatically decides whether the roughing must be internal or external and attributes the corresponding sign to the stock allowance.

The clamping point must be external to the roughing field, by at least the programmed stock allowance.

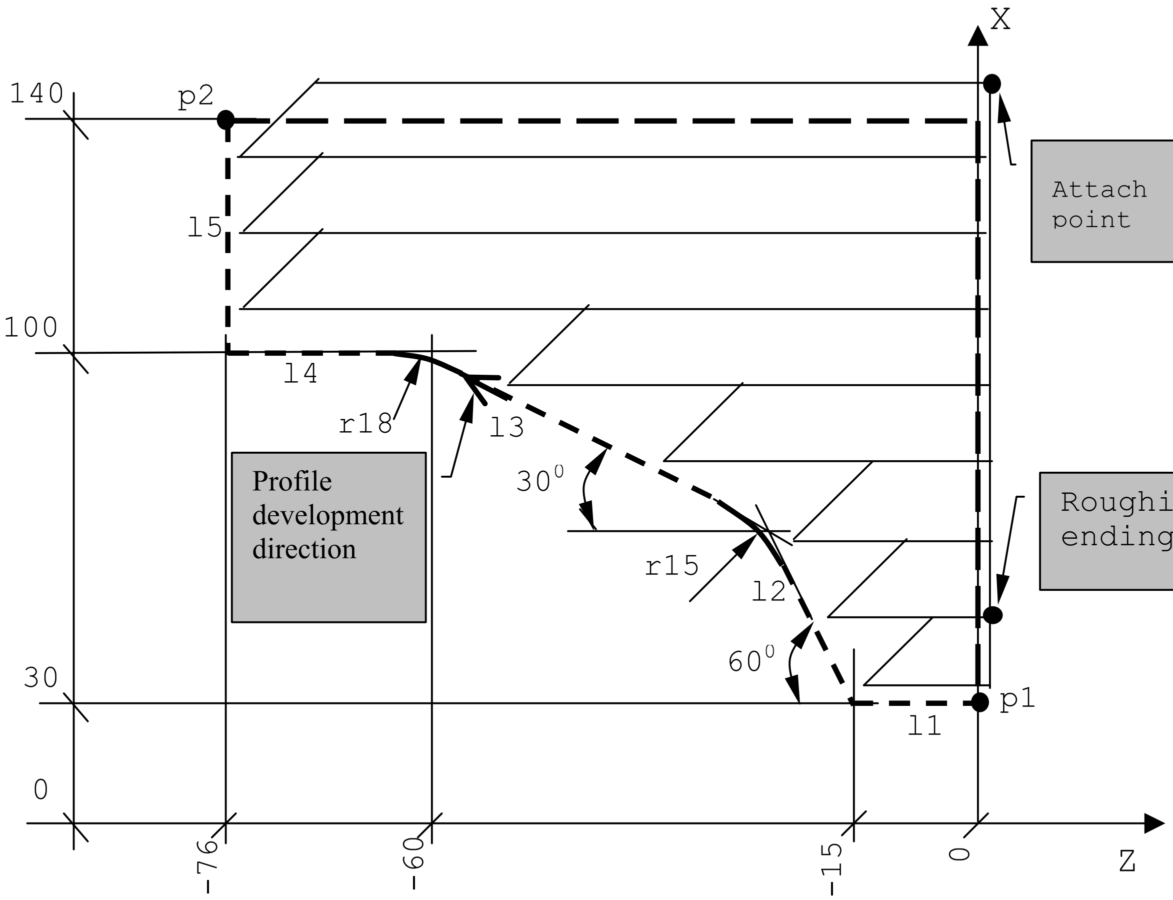
If the profile is not monotonous, i.e. if it includes pockets, the tool automatically skips the pockets during roughing.

Fig. 5.112. - Example of Roughing Parallel to the X axis




```
p1=Z-76 X140
l1=p1,a-90
l2=ZO X100 a0
l3=Z-60 X100,a-30
l4=Z-15 X30,a-60
p2=ZO X30
l5=p2,a0
(DFP,1)
G21 p1
l1
l2
r-18
l3
r-15
l4
l5
G20p2
(EPF)
T1.1 M6 S.., F..
G X143 Z1.5
(SPA,X,1,L12,X1,Z1)
..........
..........
```

Fig. 5.113. - Roughing Parallel to the Z Axis




```
p1=ZO X30
l1=p1,a180
l2=Z-15,X30,a120
l3=Z-60 X100,a150
l4=Z-75 X100,a180
p2=Z-75 X140
l5=p2,a90
(DFP,1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T1.1 M6 S..,P..
G X143 Z1.5
(SPA,Z,1,L9,X1,Z1)
..........
..........
```

PARAXIAL ROUGHING WITH PREFINISHING

To program roughing parallel to the X axis, with a final cut along the profile, use the following format:

(SPF,X,n,L..,X..,Z..)

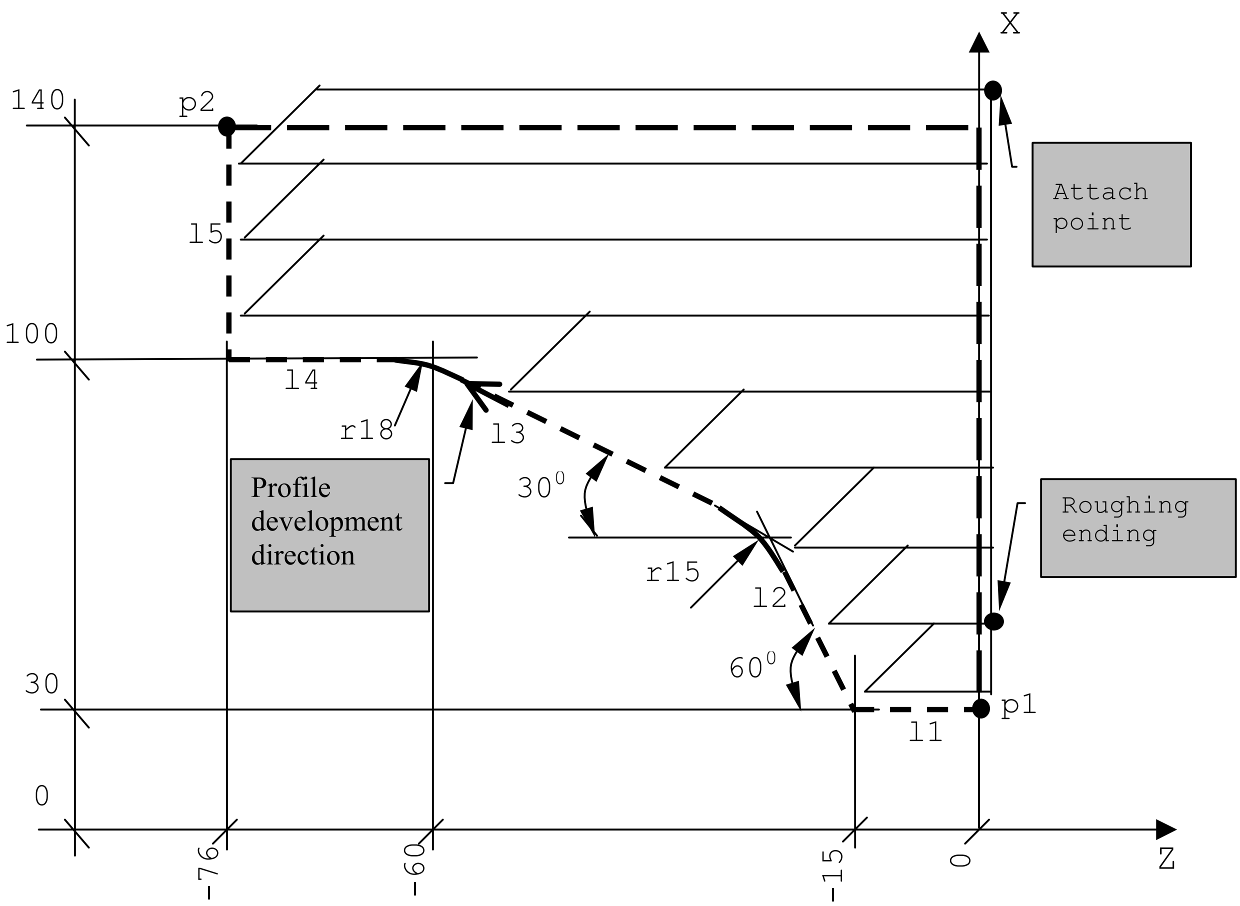
To program profile roughing parallel to the Z axis, use the following format:

(SPF,Z,n,L..,X..,Z..)

These parameters are defined in the previous section.

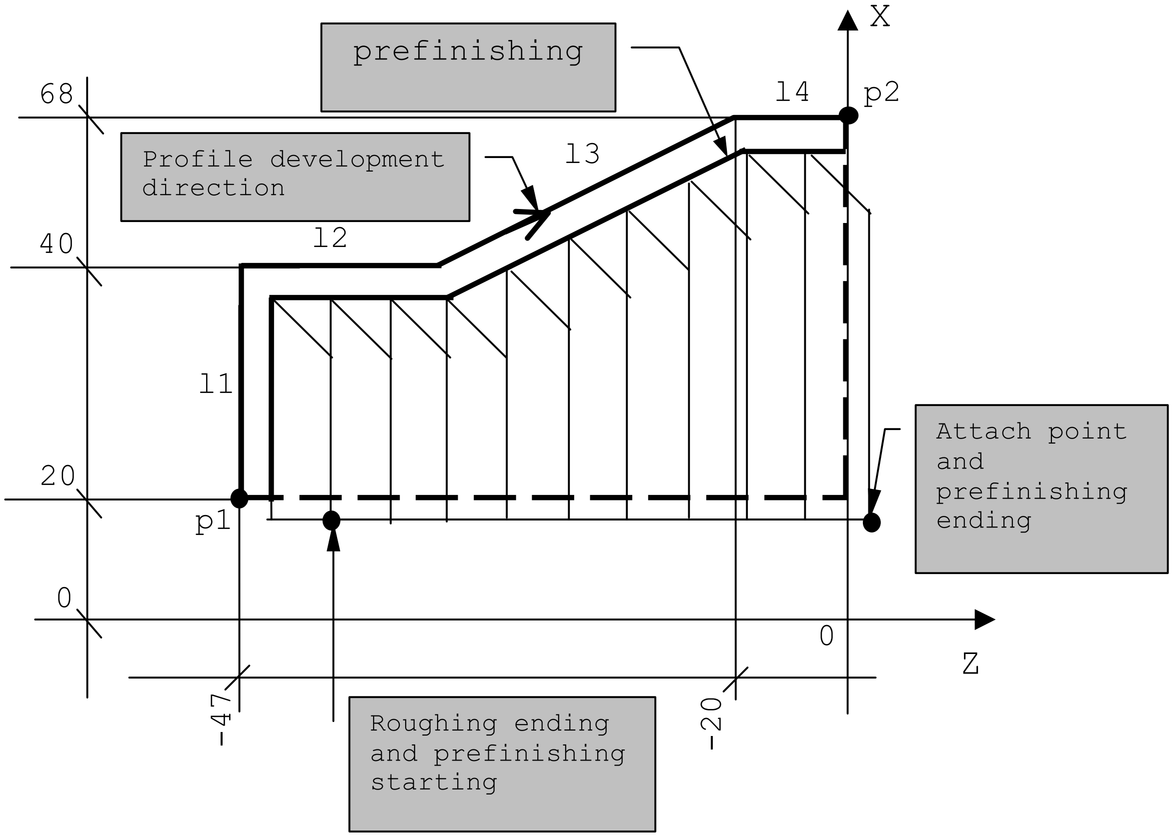
Important. The profile must be monotonous. Otherwise, an error message will be displayed.

Fig. 5.114. - Use of the SPF code



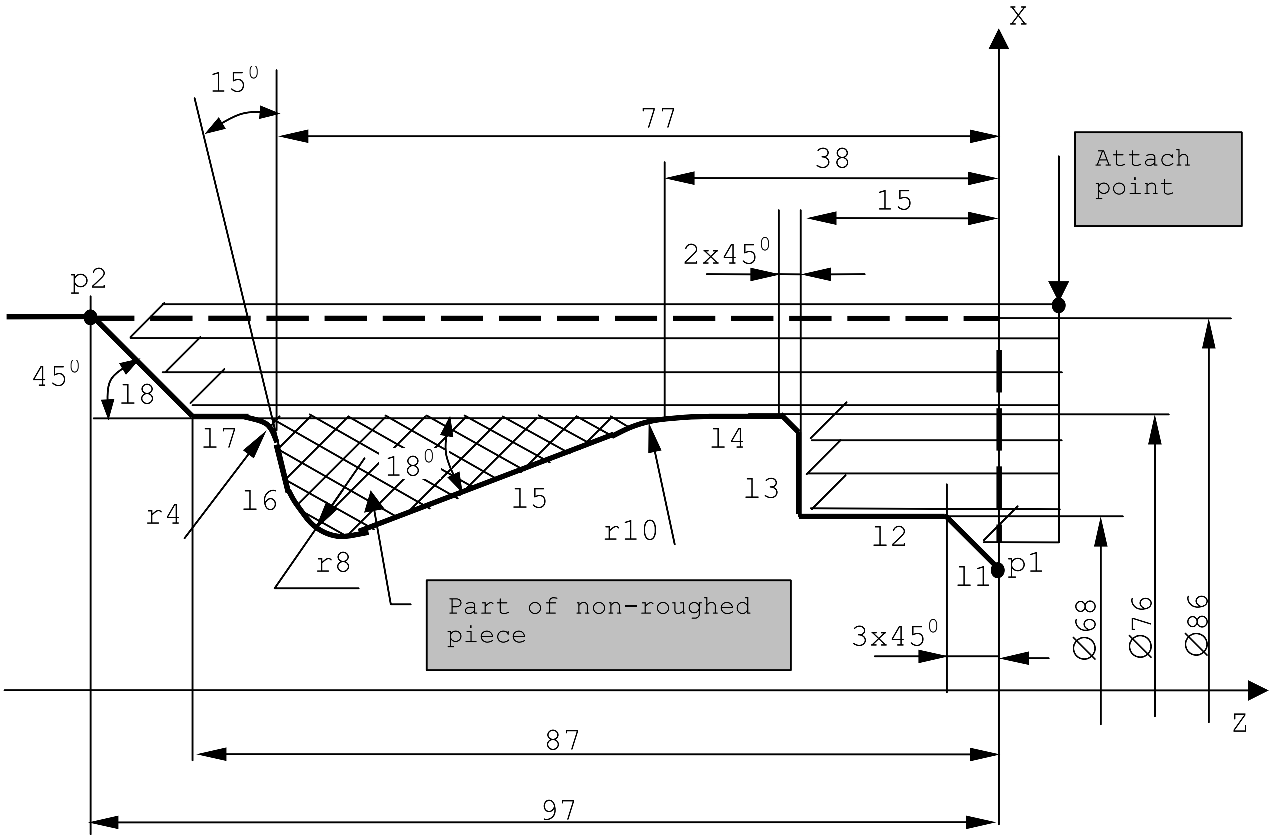
```
p1=ZO X30
l1=p1,a180
l2=Z-15 X30,a120
l3=Z-60 X100,a150
l4=Z-75 X100,a180
p2=Z-75 X140
l5=p2,a90
(DPF,1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T1.1 M6 S..,P..
G X143 Z1.5
(SPF,Z,1,L9,X1,Z1)
..........
..........
(CLP,1)
G G79 X Z M30
```

Fig. 5.115. - Internal Paraxial Roughing with Prefinishing



```
p1=Z-47 X2O          l1
l1=p1,a90            l2
l2=ZO X40,a0         l3
l3=Z-20 X68,a45      l4
p2=ZO X68            G20 p2
l4=p2,a0             (EPF)
(DFP,1)              T.. S.. F..
G21 p1               G X15 Z2.5
                     (SPF,X,1,L10,X2,Z2)
```

Fig. 5.116. - Non-monotonous profile with roughing

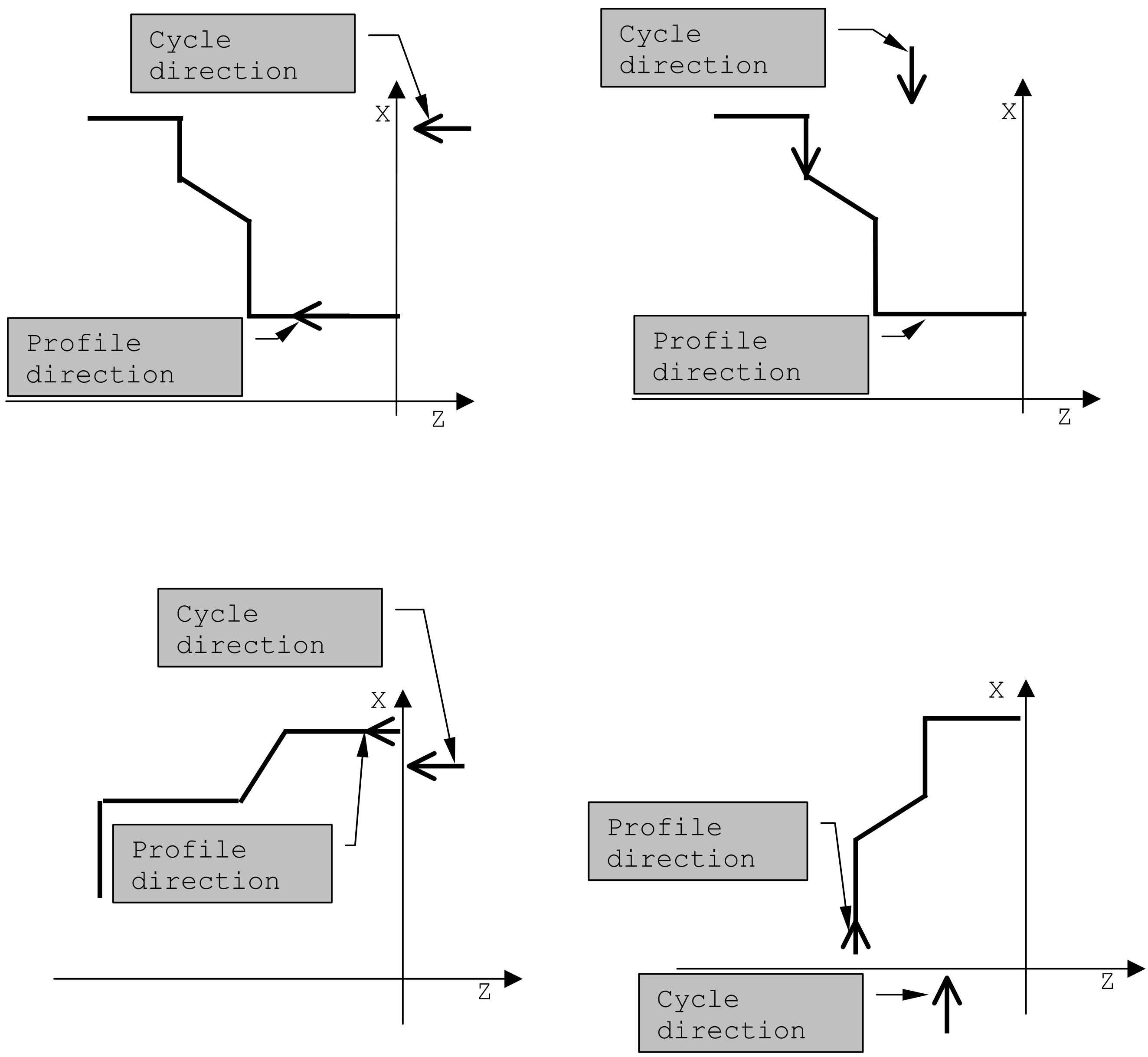



```
p1=Z0 X62            r2
l1=p1,a135           l4
l2=Z0 X68,a180       r10
l3=Z-15 X0,a90       l5
l4=Z0 X76,a180       r-8
l5=Z-38 X76,a198     l6
l6=Z-77 X76,a105     r4
p2=Z-97 X86          l4
l7=p2,a135           l7
(DFP,1)              G20 p2
G21 p1               (EPF)
l1                   T.. S.. F.. M
l2                   G X90 Z2
l3                   (SPA,Z,1,L7,X1,Z1)
```

In the above example, an SPA code has been programmed. The control executes the roughing cycle and skips the pocket. The remaining chips can be removed with standard motions. Then, program a CLP code to finish the profile.

The direction of the profile must coincide with the direction of the tool in the roughing cycle. Figure 5.117. shows an example of profile and tool direction.

Fig. 5.117. - Profile and Tool Direction



ROUGHING PARALLEL TO THE PROFILE

This code allows you to execute a roughing cycle on workpiece with an average stock allowance. The allowable format is:

(SPP,n,L,X1. X2.,Z1. Z2.)

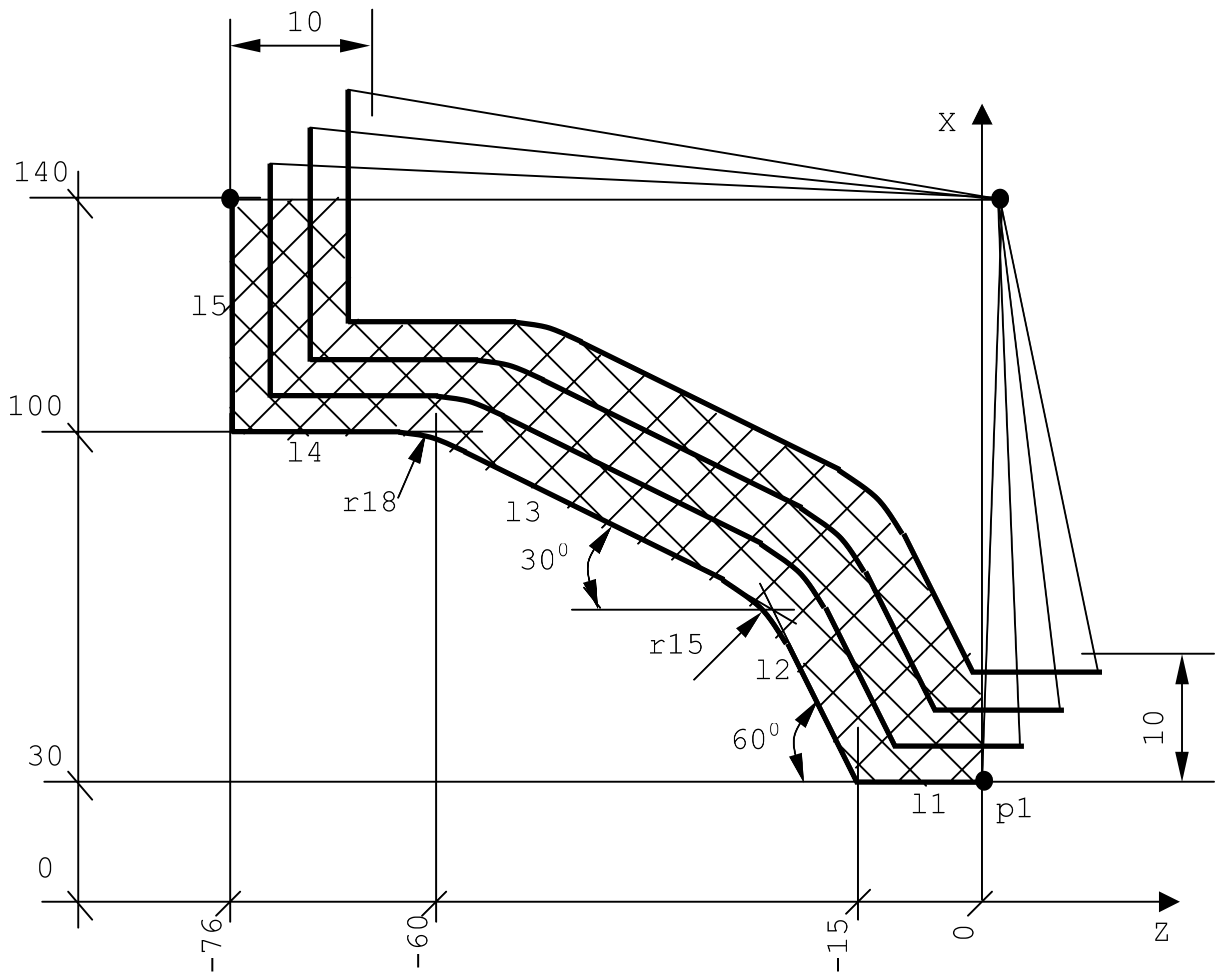
where:

n	is the profile number;
L	is the number of passes;
X1	X stock allowance to be left on the finished piece;
X2	X stock allowance on the raw piece;
Z1	Z stock allowance to be left on the finished piece;
Z2	Z stock allowance on the raw piece.

X1 and Z1 are mandatory, even if their value is zero.

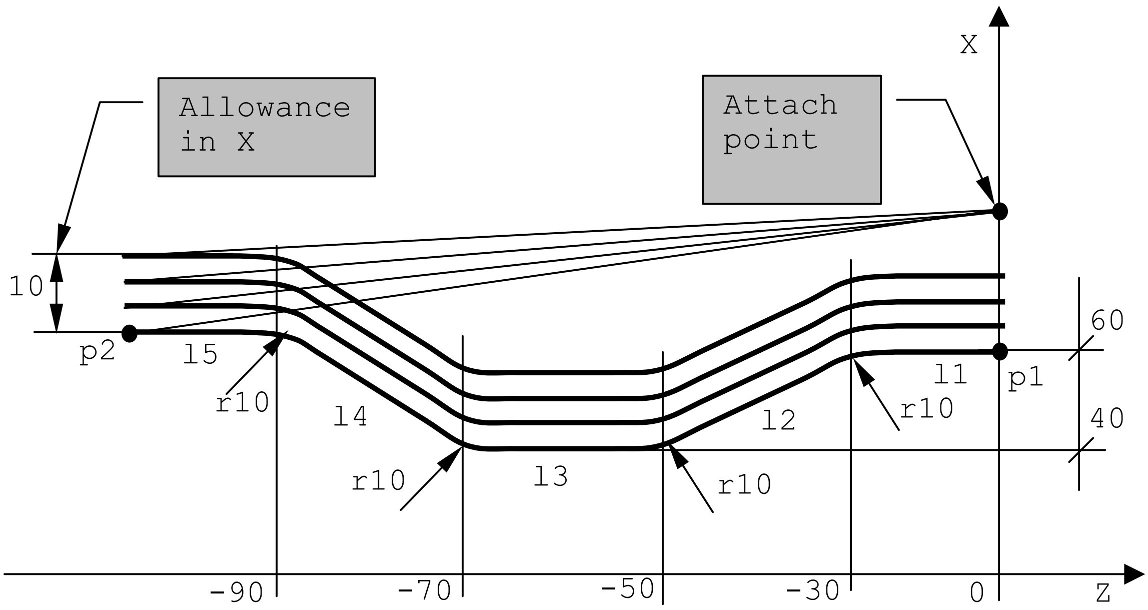
As for the clamping point, refer to the previous sections (SPA - SPF).

Fig. 5.118. - Use of SPP



```
p1=Z0 X30
l1=p1,a180
l2=Z-15 X30,a120
l3=Z-60 X100,a150
l4=Z X100,a180
p2=Z-75 X140
l5=p2,a90
(DFP,1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T..  S..  F.. M..
G X143 Z1.5
(SPP,1,L4,Z1  Z10,X1  X10)
..........
```

Example (Fig. 5.119.).




```
p1=Z0 X60
l1=p1,a180
l2=Z-30 X60,Z-50 X40
l3=Z0 X40,a180
l4=Z-70 X40,Z-90 X60
p2=Z-110 X60
l5=p2,a180
(DFP,1)
G21  p1
l1
r10
l2
r-10
l3
r-10
l4
r10
l5
G20 p2
(EPF)
T.. S.. F..
G X84 Z1
(SPP,1,L4,X X90)
```

PROFILE FINISHING

To program profile finishing, use the following format:

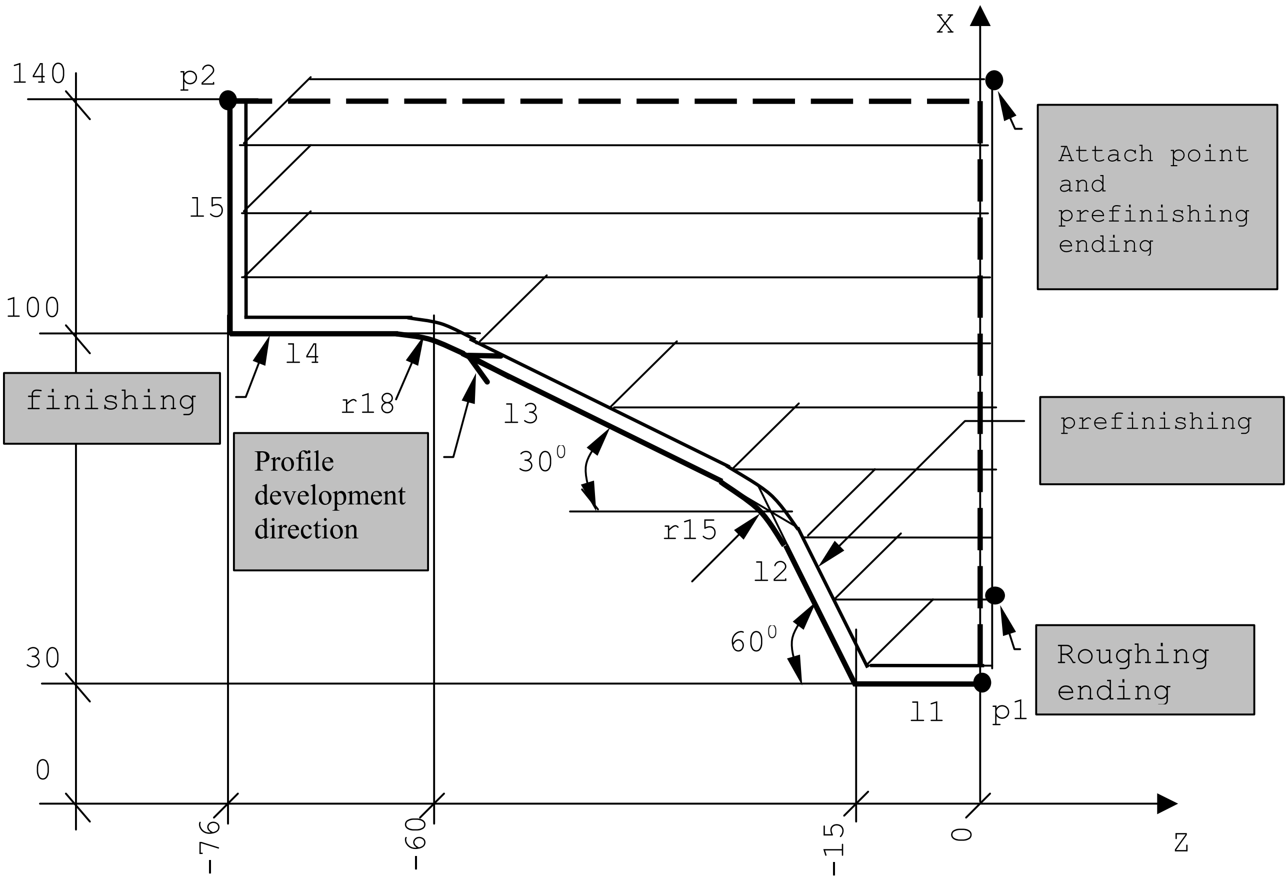
(CLP,n)

where:

n name of the profile previously defined with DFP.

CLP is the only machining cycle during which the F functions programmed inside a DFP can be activated.

Fig. 5.120. - Paraxial roughing with prefinishing and finishing




```
p1=Z0 X30
l1=p1,a180
l2=Z-15 X30,a120
l3=Z-60 X100,a150
l4=Z-75 X100,a180
p2=Z-75 X140
l5=p2,a90
(DFP,1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T1.1 M6 S.. P..
G X143 Z1.5
(SPF,Z,1,L9,X1,Z1)
X300 Z200
T2.2 M6 S.. F..
G X30 Z2
(CLP,1)
G X300 Z200
..........
```

5.16. SYNCHRONOUS BLOCKS

Synchronous blocks are those the control executes only a: certain moments or under certain conditions. To activate/deactivate synchronisation, use the following symbols:

1) # activates synchronisation;
2) & deactivates synchronisation.

Program these symbols after the number of the block and before the instruction.

Important. By default, the control executes instructions without synchronisation. However, synchronisation is the default modality for blocks containing SA and SK variables.

ACTIVATING SYNCHRONIZATION

When you program # before a block, the specified axes move after the control has executed the calculation (1 block in point-to-point mode, n blocks in continuous operations).

Use synchronisation to program instructions that must be conditioned by the result of calculations or that assign a value to a variable at the end of the programmed move. For example:

```
N9  GX100Y80
N10 #TIM1=TIMO                    -adopts the time indicated by the
.                                 system clock at the end of the
.                                 move programmed by N9
.
N29 GXY
N30 #(UCG,2,X-50 Y100,Y-20,Y80)   -defines graphic field at
.                                 the end of the move
.                                 programmed by GXY
.
N50 GX200
N51 #(BEQ,SA126,1,LAB)            -branches to LAB if Bit 126 from
.                                 SA buffer is 1. after the X axis
.                                 has reached 200
N59 GX50
```

N60 #(DLY,10)	-establishes a 10" delay at the end of the movement programmed by N59
.	
.	
N87 E30=0.2	
N88 #(RQU,1,1,ZE30)	-tool requalification when E30 reaches the desired value (0.2 in block N88, 0.3 in block N95)
.	
.	
N94 E30=0.3	
N95 #(RQU,1,1,ZE30)	

DEACTIVATING SYNCHRONIZATION

Use & to deactivate default synchronisation between calculation and axes move.

6. BASIC SYSTEM OPERATION

Chapter Objectives This chapter describes the basic system operation with the NC-110.

6.1. POWER UP

1. Turn the main switch in the electrical cabinet to the ON position.
2. Pull (or turn) the **EMERGENGY** pushbutton until you hear a click.
3. Press the **CONTROL ON** (or **P1**) pushbutton.

The control should turn on and the indicator should lit up. If no errors are detected by the diagnostics, the last executed program appears on the CRT.

When the message "M.T. OFF" appears, press **P1** again: the machine tool interface will turn on.

6.2. POWER OFF

When you press (or turn) the **EMERGENGY** pushbutton, you deactivate the machine tool and the control and delete stored information, including axes positions.

Nevertheless, programs, tool offsets and absolute origins remain stored in memory when the control is off.

6.3. STORING AND MODIFYING PROGRAMS USING THE KEYBOARD

In order to edit or execute a program you must store it in the control's memory. Storing programs can be done using the keyboard or a peripheral device (i.e., tape reader, teletype, magnetic cassette, etc. Refer to Chapter 7).

6.3.1. STORYING PROGRAMS

Press **PO** to establish Edit. Then input the command:

EDI,PROG1
Press **SEND**

The program name can have as many as 6 alphanumeric characters. If the program is already in memory, it appears on the CRT to be modified. If the program is not already in memory, the control prepares itself to enter a new program and displays the message "NEW", activating the insert instruction (INS).

You can now enter program blocks. Close each entry with press **SEND.**
For example:

N1 T1.1 M6 S200 M3 M7
Press **SEND**

N2 G0 Z100 X50 Z100
Press **SEND**
etc.

When you reach the end of the program, press the **ESCAPE** key two times: the first time disables the INS instruction and the second exits the Edit condition and closes the file.

When you press the **ESCAPE** key the second time, the following message appears on the CRT:

JOB EDI NAME

Press **P0** to go back to the previously active screen.

6.3.2. MODIFYING PROGRAMS

Press **P0** to establish Edit.

Enter

EDI,PROG1
Press **SEND**

The allowable EDIT operations are:

1) RIM - modify a block;
2) INS - insert a block;
3) CAN - delete a block.

RIM - Modify a block in a stored program

To modify a block in a stored program, use the following procedure:

1. Display the program for editing by using the command

EDI,PROG1
Press **SEND**

2. Use the **LINE FORWARD** or **LINE BACK** keys to position the cursor on the block you want to modify. The block with the cursor will be displayed in reverse mode.

3. Enter RIM press **SEND**
The block will be shown on the second line of the display.

4. Move the cursor in the block with the **FORWARD SPACE** or **BACKSPACE** keys until you create a space to the right of the characters you want to modify. To remove a character, press the **DELETE** key. To insert characters, position the cursor on the desired position and type in the new characters.

5. Press **SEND** to close the entry. The new block, on the second line of the screen, replaces the old one in the display area.

To modify additional blocks, repeat steps 2, 4, 5. The RPL instruction remains active.

CAN - Delete a block

To delete a block use the following procedure:

1. Position the cursor on the block to be deleted.

2. Enter.

CAN
Press **SEND**

3. Press **SEND** to delete the block.

Whenever the **SEND** key is pressed, the next block in the program is deleted. To delete blocks in other parts of the program, repeat the procedure.

INS - Inserting a block

To insert a block in a stored program, use the following procedure:

1. Use the **LINE FORWARD** or **LINE BACK** keys to position to the block that should come just before the new block. The block will be displayed in reverse.
2. Enter

INS
Press **SEND**

3. Input the block (or blocks). Press **SEND** after each block. The block or blocks that you input will be inserted after the block in reverse.

To insert additional blocks, repeat the procedure.

$ - Searching for Characters.

If the control is in the Edit mode and no input instructions (RIM-INS-CAN) are active, it is possible to search inside the displayed program for a sequence of characters (maximum 32).

1. Press the **$** key.

2. Input the sequence and then press **LINE FORWARD** or **LINE BACK** to search forward or back through the program.

For example:

```
$ T1.1      LINE FORWARD
$ G1 X100   LINE FORWARD
$ F         LINE BACK
```

If the control does not find the character sequence, the editor positions to the beginning of the program (BOF) for search reverse or to the end of the program (EOF) for search forward.

DIR - Listing the Directory of Stored Programs

Enter

DIR
Press **SEND**

This command displays all programs that are stored in the memory. Each program is shown with its name and length in memory sectors (1 sector = 128 bytes).

Each page of the directory contains 5 programs. To display the next page, press the **LINE FORWARD** key.

The message on the last line of the page ("FREE SECTORS:") and the number indicate the number of memory sectors that are still free. The word J0B appears in the upper left hand corner of the GRT. For example:

```
JOB DIR

DIR/MP1     NAME
NAME                  SECT        LREC        ATTR
PROG1                    4
P99                     12
PR24                     8
FREE SECTORS           258
```

If the control is in JOB condition, it is waiting for new instructions. Press the Keyboard Reset key to restore the JOB condition anytime.

COP - Copying a program in memory

For example:

COP,P9,P24
Press **SEND**

This control copies program P9 and stores it with name P24.

REN - Renaming a program

For example:

REN,P99,PR1
Press **SEND**

This command renames program P99 with name PR1 in the memory of the control.

DEL - Delete a program

For example:

DEL,PROG1
Press **SEND**

This command deletes program PROG1 from memory.

INI - Initialising User Memory

For example:

INI,NAME,/MP1
Press **SEND**

This command initialises and completely clears the user memory. NAME is a memory identification designator (6 characters max.). The first character must always be a letter.

Important. This command cancels the offset file, the origin file and the tool life management file. In this case, you must create them again.

6.4. CREATING OFFSETS, ORIGINS AND TOOL LIFE FILES

You must store offsets, absolute origins and tool life parameters in suitable files. To create these files, use the procedures described in the following sections.

6.4.1. OFFSETS FILE

The offsets file is a formatted file called FILCOR, that is stored in memory MP3. FILCOR usually contains the tool length along the Z axis and the tool diameter (K).

If the control monitors an electronic probe, the offsets file can contain additional information such as the current and the maximum offset values for both tool length and tool diameter.

When the current offset value exceed the maximum allowed value, the tool is considered out of useful life.

To create the offsets file, perform the following steps with the system in EDP:

1. Delete the FORMAT file, if it exists, using the command
DEL,FORMAT/MP3
Press **SEND**

2. Create a new FOMMAT file using the command
EDI,FORMAT/MP3
Press **SEND**

3. If the offsets file only contains length and diameter values, write the following sequence of characters:

I2A1L3A1L3A1L3A1I1
Press **SEND**

If the offset file also contains information concerning the electronic probe (c.., m..), type in:

I2A1L3A1L3A1L3A1I1A1L3A1L3A1L3A1L3
Press **SEND**

4. Exit from Edit by pressing the **ESCAPE** key twice.

5. Create the offset file using the command
FOR,FILCOR/MP3,xx
Press **SEND**

where: xx specifies the number of offsets required for the file.

6. Press **PO** to call the process display partition (machining screen) and enter:

CAC
Press **SEND**

the control replies with

CONFIRM? (Y/N)

Type Y
Press SEND

Now, the offset file is ready to be initialised. Refer to section 6.5.

6.4.2. ORIGINS FILE

The origins file is a formatted file called FILORI and stored in memory MP3. To create FILORI, perform the following steps with the system in EDP:

1. Delete the FORMAT file, if it exists, using the command

DEL,FORMAT/MP3
Press **SEND**

2. Create a new FORMAT file using the command

EDI,FORMAT/MP3
Press **SEND**

3. Write the following sequence of characters:

I2A1L4A1L4A1L4..

Repeat the characters sequence A1L4 once for each interpolated axis in the system.

4. Exit from Edit by pressing the **ESCAPE** key twice.

5. Create the origins file using the command

FOR,FILEOR/MP3,xx
Press **SEND**

where: xx specifies the number of origins required for the file.

6. Press **PO** to call for the process display partition (machining screen) and enter:

CAO
Press **SEND**

the control replies with

CONFIRM? (Y/N)

Type Y
Press **SEND**

Now, the origins file is ready to be initialised.

6.4.3. TOOL LIFE FILE

The tool life management file is a formatted file called VITUT and stored in memory MP3. To create this file, execute the following procedure, with the system in EDP:

1. Delete the FORMAT file, if it exists, using the command
DEL,FORMAT/MP3
Press **SEND**

2. Create a new FORMRT file using the command
EDI,FORMAT/MP3
Press **SEND**

3. Type in the following sequence of characters:
UAUAUAURUAR3A
Press **SEND**

4. Exit from Edit by pressing the **ESGAPE** key twice.

5. Create the tool life file using the command
FOR,GETOOL/MP3,xx
Press **SEND**

where: xx specifies the number of tools monitored by the file.

6. Press **PO** to call the process display partition (machining screen) and enter:

CTU
Press **SEND**

the control replies with

CONFIRM? (Y/N)

Type Y
Press **SEND**

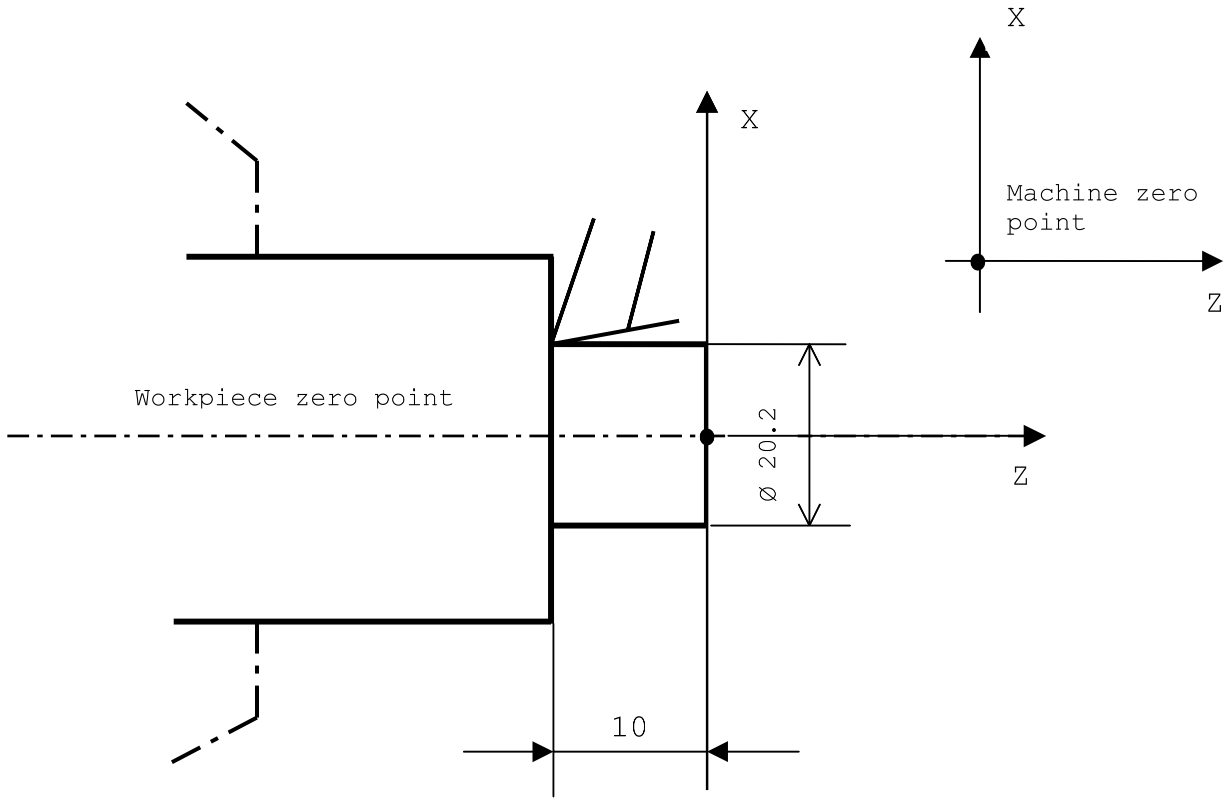
Now, the offset file is ready to be initialised by entering a command like:

VTU,1,T11, 11,t120,t5,t120,B

Press **SEND**

6.5. TOOL PRESETTING

Fig. 6.1. - Workpiece and Tool Zero



To present the tools on the machine, use the following procedure:

1. Fit each tool on the machine and call the desired program with SPG.

2. Home the axes.

3. Manually set the turret on a position that allows free rotation.

4. Call a tool from the keyboard. Example: T1.1 M6

5. Manually set the tool tip on a point having known coordinates and stop the spindle.

6. Keep the axes in this position and type in the tool offset, the diameter and the Z coordinate referenced to the workpiece zero.

Example:

3,X20.2,Z-10
Press **SEND**

The control automatically calculates the value of the length offset, by establishing the relationship between the machine zero and the workpiece zero according to the length of the various tools.

7. Repeat steps 3-4-5-6 for each tool.

8. Deposition the turret on a point that allows free rotation.

Important. Your program must begin with a tool change block. Before any tool change and at program end. You must program the moves that allow the turret to rotate and then call the subsequent tool.

Example:

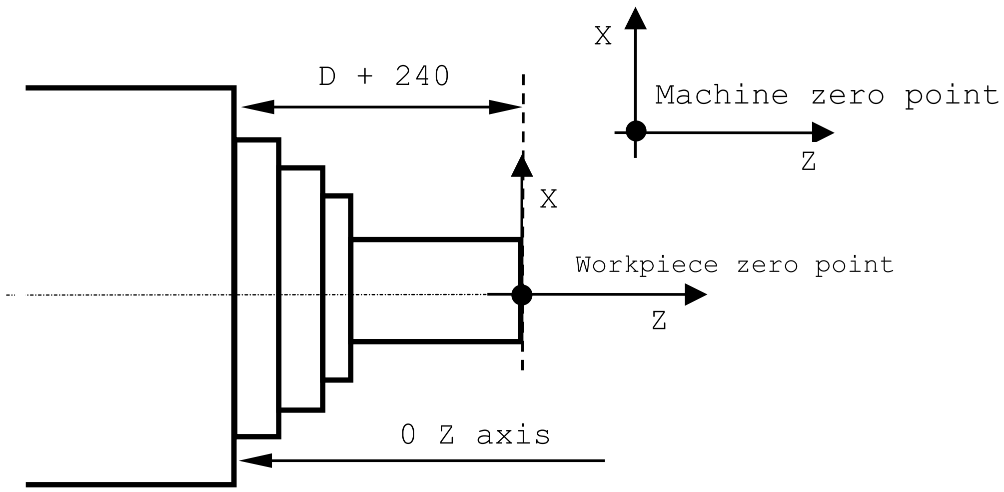
```
N1  T1.1 M6
............
............
N18 G X.. Z..
N19 T2.2 M6
............
............
N99 G X.. Z.. M30
```

For tools that remain fit on the turret, i.e. tools required by all the machining cycles. You can establish the x reference once. When you first fit the tool on the machine. Since the distance between machine and workpiece absolute zero is a constant, you don't need to repeat the operation.

Similar criteria apply to the Z axis when you reference the tool to another fixed point in the machine. For example. When your machine features a self'centering spindle. In this case, skim the spindle border with the tool tip and then type in:

n,Z0
Press **SEND**

Fig. 6.2. - How to reference the Z axis



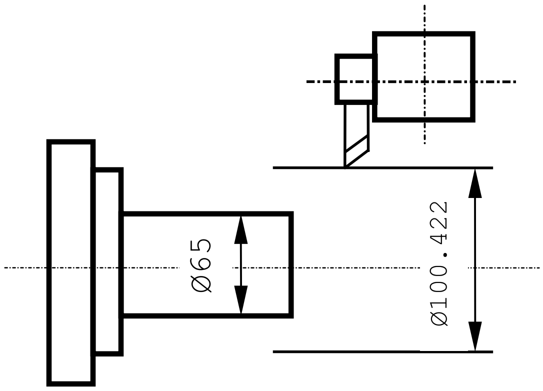
Since the Z dimensions are referenced to the actual workpiece zero. You must program the distance between spindle border and workpiece zero with the UOT (Use Temporary Origins) code. In theory, you can program only one UOT block at the beginning of the sequence. Yet, it is advisable to insert one UOT block after each tool change.

```
N1 (UOT,0,Z240)
N2 T1.1 M6
...........
...........
N20 (UOT,0,Z240)
N21  T2.2 M6
...........
...........
N30 (UOT,0,Z240)
N31 T3.3 M6
...........
```

MODIFYING TOOL OFFSETS

During a lathing cycle, the control can detect a variance between the nominal and the measured diameters. For example (Fig. 6.3.):

Fig. 6.3.



nominal diameter = 65 mm
measured diameter = 65,4 mm

If the current tool and offset are not on the CRT, call them from keyboard: T1.1 M6.

The control displays the current positions of the axes X and Z. In the example: X +0100.422   Z+0018.964

Since the measured variance is 0.4 mm the actual lathed diameter will be X = 100.822 mm

You must therefore reset the offset. Key in the following instruction:

1, X100.822
Press **SEND**

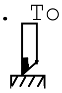
The control will display the new current position of the X axis (X=100.822) and the new offset value corrected by 0.2 mm. The correction will be stored and applied at the subsequent axis motion.

Analogous criteria can be applied to the Z axis.

Alternatively, you can modify the tool offsets by correcting the displayed values. Use this procedure:

1. Press the **CLEAR** key to clear the CRT.

2. To display offset 1, type in the following instruction:

 1

Press **SEND**

The offset and the preceding correction value appear on the CRT. For example:

01, X-0150.032,Z-0120.367,R1.2,03

3. Use the **BACK SPACE** and **FORWARD SPACE** keys to position the cursor on the digit you want to modify and make the necessary correction. Then press the CSEND3 key.

For example,

01,X-150.232,Z-0120.367,R1.2,03

The control checks whether the correction exceeds the allowable limit established in characterization. The default limit value is 1 mm. If you want to enter a greater value, you must first cancel the current offset (with CAC,n Press **SEND**) and then replace it with the desired value.

4. Select the INPUT VIA KEYBOARD mode selector.

5. Enter a T function with the corresponding tool and offset and M6 function. Example: T1.1 M6.

6. Press **START**. The correction will be activated and displayed at the first axis motion.

Use step 2 in order to specify the tool radius offset and the orientation code 0 for tools to which vectorial radius compensation must be applied. Also use step 2 if you must specify or modify the current and maximum correction values (i.e. c.., m..).

INCREMENTAL TOOL OFFSET MODIFICATION

By programming the UGA code, you can specify an incremental correction for the tool offset.

Use the following format:

UCA, ± offset number, X ± diameter correction, Z ± length correction

where:

n — offset for tool fit on the secondary turret

\+ — enter a positive value if the control measured a negative variance (in diameter or length)

\- — enter a negative value if the control measured a positive variance (in diameter or length)

Example:

UCA,3,X-0.02
Press **SEND**

The control stores the correction and displays the offset on top of the CRT.

The correction will be applied from the subsequent call of the offset with M6.

If you correct the currently displayed offset and the machine is not in the Hold status, the correction will be stored and applied at once.

6.6. TOOL LIFE MANAGEMENT

Tool life management makes it possible to automatically check the machining time of each tool and make sure that it doe not exceed the expected life.

You can assign to each tool a machining time up to 9999 minutes.

If the programs calls a tool having a remaining life that is less than the minimum life allowed, the tool is automatically replaced by an alternative one. If the remaining life of the alternative is also below the allowable limit or if there is no alternative tool, the program stops and a message from the machine logic is displayed.

You can program a "cascade" of alternative tools, i.e. each alternative tool can have another alternative tool and so on.

The control calculates tool machining time while it is actually being used for workpiece machining, either in auto or semiauto.

You can enter and edit tool life data from the keyboard with the system in process display partition (machining screen). Use the following operations:

- Initialise the Tool Life File

For each record, the file has the following format:

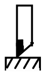
```
VTU,tool number, T field 2, field 3, t field 4,
                 t field 5, t field 6, field 7
```

where:

tool number	a whole number without sign. The maximum limit is the number of tools specified in the tool life file
field 2	number of the alternative tool. It must be a whole number without sign
field 3	offset to be applied to the alternative tool. It must be a whole number without sign
field 4	maximum theoretical tool life in minutes. It must be a whole number without sign
field 5	minimum theoretical tool life in minutes. It must be a whole number without sign
field 6	remaining working time in minutes. This field is decremented during the machining operations using the tool. It will begin with the value specified in field 4.

field 7 tool status. Identified by a letter:
A = broken tool
B = usable tool (whose life time must be managed)
C = usable tool (whose condition is to be monitored by with G74)
D = tool having a working time under the minimum level

For example, you can initialise a tool life file by inserting a record like:

VTU,2,T12,  12,t60,t2,t60,B
Press **SEND**

▪ <u>Display and Edit a Record in the Tool Life File</u>

Enter

VTU,n
Press **SEND**

Line n of the tool life file is displayed in the Echo area of the CRT. The displayed parameters of tool n can be modified and then stored by pressing **SEND**.

▪ <u>Delete a Tool from the Tool Life File</u>

CTU,n
Press **SEND**

cancels tool n.

▪ <u>Delete Tool Life File</u>

CTU
Press **SEND**
The whole tool life file is deleted.

6.7. HOMING THE AXES

Whenever power is removed from the system (control + machine tool) and the system is re-powered-up, you must home the axes. You home the axes by positioning them on the absolute zero points. Each machine axis has an absolute zero close to one of the limit stops.

To home the axes to the absolute zero points use the following procedure. Depending on the operator panel of your system:

Operator panel

1. Press the **T7** key to select the Automatic Home Position.

2. Select the axis to be homed by using the **LINE FORWARD** or **LINE BACK** keys. The axis you select is displayed in reverse.

3. Turn the JOG potentiometer to the desired speed and direction.

4. Press and hold the **START** pushbutton.

The selected axis will automatically home to the point defining the absolute machine zero. Repeat the same operation for the other axes.

To home the indexed table, use the following procedure:

1. Move the Mode selector to position 1 (Input from keyboard).

2. Type in BO and then press the **START** pushbutton.

Notes:

• If the JOG potentiometer is not set in the proper direction, the axis will not move.

• If you input RAP=1 press **SEND** from the keyboard in step 3, the control automatically selects the rapid speed and the correct direction for homing the axis. It is not necessary to hold the **START** pushbutton.

• If specified in the machine logic, you can automatically home the axes, one by one.

6.8. JOGGING

There are two ways of jogging the machine axis:

- continuous jog;
- incremental jog.

Operator panel

a) Continuous jogging

1. Press the **T4** pushbutton.

2. Select the axis to jog by using the **LINE FORWARD** or **LINE BACK** keys. The axis you select is displayed in reverse.

3. Turn the JOG potentiometer to the desired speed and direction.

4. Press and hold the **START** pushbutton. The axis moves at the selected rate and direction until you release **START**.

b) Incremental jogging

1. Press the **T5** pushbutton.

2. Select the axis to be jogged by using the **LINE FORWARD** or **LINE BACK** keys. Type in the increment value. For example: JOG = 0.55 press **SEND**

3. Turn the JOG potentiometer to the desired speed and direction.

4. Press and hold the **START** pushbutton. When you release **START**, the axis stops and the remaining distance is displayed on the CRT.

6.9. ELECTRONIC HANDWHEEL

The electronic handwheel allows you to move the axes from keyboard. The allowable entries are:

VOL = 1 enables the electronic handwheel
Press **SEND**
VOL = 0 disables the electronic handwheel
Press **SEND**

To move the axes, use the following procedure:

1. Type in VOL = 1 press **SEND**

2. In control panel A. Move the mode selector to the MANUAL position.

In control panel B, press the **T4** pushbutton.

3. Press **LINE FORWARD** or **LINE BACK** keys to select the axis to be moved.

4. At each complete turn of the handwheel, the selected axis moves by 1mm.

For greater accuracy, select the MANUAL INCREMENTAL Position in step 2 (or press **T5** in panel B). At each turn of the handwheel, the axis moves by 0.1 mm.

6.10. EXECUTING A PROGRAM

Operator panel

1. Activate the program for execution with the SPG command.

For example: SPG,PROG1 -selects program PROG1 from memory
Press **ENTER**

2. Select the execution mode by pressing **T2** or **T3**.

3. Press the **START** pushbutton.

Notes:

• If you select STEP execution, you must press **START** to execute each block.

• To perform a portion of a program up to a given block number, use the following command:

ESE,n
Press **ENTER**

For example:

ESE,22 -the program stops when block 22 has been
Press **ENTER** executed. To resume execution, press **START**

ESE cannot be programmed inside a continuous profile (G27-G28).

6.11. PROGRAM TESTING

You can test programs by using 3 methods:

- axes standstill program test;
- "dry run" program test;
- limited block-by-block program test.

AXES STANDSTILL PROGRAM TEST

You can execute this test with the axes in hold from any of the available screens, in auto or block-by-block mode.

In either case. Activate the axes standstill by entering UAS=1. Press **SEND** (shown in Screen 1).

Use the following procedure:

1. Input UAS=1 press **SEND**

2. Activate the program you want to test with the SPG command. For example: SPG,PROG1 press **SEND**

3. With panel A. Move the mode selector to the AUTO or BLOCK-BY-BLOCK position. With panel B, select the execution mode by pressing **T2** or **T3**.

4. Press the **START** pushbutton.

If you want to run the test with the graphic display, press **P1** after step 2, then input the UCG command to create the graphic field. For example,

UCG,1,Z..,Z..,X..,X.. Press **SEND**	-defines the workpiece area and the plane you want to display.

If the control detects syntax or geometry errors, it signals them and stops the execution. Edit the program (after pressing **P0** to quit the graphic screen), then repeat the test.

DRY RUN PROGRAM TEST

This method allows you to run a program at calibrated rapid rates with no workpiece in place.

The dry run test must be performed with the axis in motion; therefore, if UAS=1 is active, enable motion with UAS=0.

Use the following procedure:

1. Select the program to be tested:

SPG,TEST
Press **SEND**

2. Input the command UVR,1 press **SEND**, which appears on display 1 only.

3. Press the **START** pushbutton.
If you selected the auto mode, the program will run until the end block. In Semiauto, you must press **START** to execute each block.

You can vary the execution rate of movements programmed at feedrate with potentiometer S2 (Feedrate Override).

If you want to vary the execution rate of movements programmed at rapid rate, input URL=1 press **SEND.** You can then control rapid rates with potentiometer S3. To reset the programmed rapid, input URL=1 press **SEND.**

LIMITED RAPID, BLOCK BY BLOCK PROGRAM TEST

This method allows you to execute a first run piece in semiauto. You can vary programmed rapids with potentiometer S3 and reduce rapid movements to 0%.

Use the following procedure:

1. Select the program to be tested:

SPG,TEST
Press **SEND**

2. Input the command URL=1 Press **SEND**, which appears on display 1 only.

3. Press the **START** pushbutton.
At the end of each block, the subsequent block appears on line 4 of the CRT. When you press **START** the block is executed.

If the block includes a rapid move. You can vary the programmed value with the S3 selector. The current dimension is displayed on line 1 and the programmed dimension is displayed on line 3. You can check the actual position of the tool on the machine. If the programmed dimensions are not correct, press the **RESET** key, correct the erroneous block and then restart the execution. At the end of the test, change the block permanently with an EDIT procedure.

To execute the modified block use the following procedure:

1. Select the Input Via Keyboard mode (the mode selector **T1**).

2. Press **START**. At the end of the block. Select the Semiauto mode. Then **START** to execute the subsequent block.

You can only modify blocks that do not depend on previous or following blocks. Blocks with direct programming (GTL), circular interpolation, active tool radius offset, etc. cannot be modified and executed from keyboard with this procedure.

If you must include slashed blocks. Enter USB=1 press **SEND** before pressing **START**.

When the test is completed. Correct the remaining erroneous blocks, cancel URL with URL=0 press **SEND** and USB with USB=0 press **SEND**. Then resume ordinary cycle execution.

To display a previous block, press **LINE BACK**. To display the following block. Press **LINE FORWARD**. Press **ESCAPE** to display the block after the last executed one.

Important. If you must repeat a sequence of operations after checking the piece or adjusting the tool offset, search for the starting block. If you modified offsets, search for the tool change block (refer to section 6.12.).

6.12. JOG RETURN

This feature allows you to establish a hold during a machining operation in auto (with **HOLD**), jog the machine axes to inspect the tool or the piece, and then automatically return the axes to the position the machine had when you established hold.

You can return the axes in two ways:

- one axis at a time (with RAP=0);

- return along the jog path (retrace axis by axis, enter RAP=1).

After pressing **HOLD** and jogging the axes to a given position, use the following procedures:

SELECTED AXIS JOG RETURN (RAP=0)

Operator panel

1. Enter RAP=0 press **SEND** from the keyboard.

2. Select the JOG RETURN mode with **T6**.

3. Select the desired axis with **LINE FORWARD** or **LINE BACK** keys.

4. Set the JOG potentiometer to the desired rate and direction for the returning axis.

5. Press and hold the **START** key (**T9**). The selected axis starts to return. When the return is completed, the START led turns off. If you release START before the return is completed, the axis stops.

AUTOMATIC JOG RETURN (RAP = 1)

Important. When RAP=1 is active, the control automatically retraces in reverse order the sequence of manual moves performed during jogging. The control stores as many as 32 jog moves.

Operator panel

1. Enter RAP=1 press **SEND** from the keyboard.

2. Select the JOG RETURN mode with **T6**.

3. Set the JOG potentiometer to the desired rate and return direction.

4. Press and hold the **START** pushbutton. The selected axis starts to return. If you release **START** before the return is completed, the axis stops.

6.13. MULTIBLOCK RETRACE

This feature allows you retrace the profile up to a given position. The axes can be drawn back by as many as 64 blocks (this limit must be established during characterization).

To retrace the axes, use the following procedure.

1. Stop the axes with **HOLD**.

2. Enter MBR=1 press **SEND** to activate retracing.

3. Press **CYCLE START**:

- if the Semiautomatic mode has been selected, the axes will move back by only one block;
- if the Automatic mode has been selected, the axes will move back by n (1 to 64 blocks).

To vary retracing speed, use **FEEDRATE OVERRIDE**.

4. Stop the retracing movement with **HOLD**.

5. To deactivate retracing, enter MBR=0 press **SEND**.

6. To resume execution in the current mode (Auto or Semiauto), press **CYCLE START**.

6.14. BLOCK SEARCH FOR EXECUTION

To search for a block or sequence of characters in order to start execution from a specific block, use this procedure:

1. Select the BLOCK-BY-BLOCK mode (the mode selector **T3**).

2. Using the keyboard, enter the block number you want the control to search for. For example, N20.

3. Press **LINE FORWARD** to search forward in the program, or **LINE BACK** to search back through the program.

If you do not input the n number, the control will only move one line forward or backward each time you press **LINE FORWARD** or **LINE BACK.**

When the control finds the block number, it displays the block on the third line of the CRT. If the control does not find the number, it stops at the beginning or at the end of the block and displays an error message.

The block that appears on the third line of the CRT will be executed when you press **START.** The control does not process the information coming before this block. Therefore, the rest of the program must contain all the information needed for machining.

6.15. AUTOMATIC SEARCH

After an interruption or a reset of the cycle, this feature allows you to search for the block in which the interruption took place and automatically resume the cycle.

The control automatically updates and stores the parameters that characterise a machining cycle while the cycle is being executed. Using these parameters, the control can also automatically search for the interrupt block.

You can perform automatic search in two ways:

- automatic search after reset or power-off;
- automatic search of a given block.

AUTOMATIC SEARCH AFTER RESET OR POWER OFF

Use the following procedure:

1. With the mode selector (**T2**), select the Auto mode.

2. Activate the automatic search feature with RCM press **SEND.**

3. Press **START**. When the control finds the interrupt block, the following message is displayed: END OF MEMORIZED SEARCH.

4. Deactivate the automatic search feature with ERM press **SEND.**

AUTOMATIC SEARCH OF A GIVEN BLOCK

Use the following procedure:

1. With the mode selector (**T2**), select the Auto mode.

2. Activate the automatic search feature with RCM press **SEND.**

3. Enter the following command:
ESE,n
Press **SEND**

where n is the number of the desired block.

Important. You cannot search for a block belonging to a subroutine.

4. Press **START**. When the control finds the block, the following message is displayed: END OF PROCESS.

5. Deactivate the automatic search feature with ERM press **SEND**.

To resume execution, follow the procedure described in section 6.16.

Notes:

• With RCM press **SEND**, the cycle is resumed from the block where the interruption took place. With RCM press **SEND**+ESE,n press **SEND**, the cycle is resumed from block n+1.

• You cannot enter MGM after an REL (=program release) command.

• Automatic search is only allowable when cycle execution starts from the beginning of the program or after an automatic search. You cannot use automatic search if execution has been activated by using the keyboard.

6.16. RESUMING EXECUTION

According to the system status, the allowable options are:

- resuming execution after general power off (control + M.T.);
- resuming execution after M.T. power off or reset;
- resuming execution after **HOLD**.

AFTER GENERAL POWER OFF (CONTROL + MACHINE TOOL)

To resume the cycle after a general power off, use the following procedure:

1. Home the axes.

2. Establish the Automatic mode.

3. If needed, activate automatic search for a given block with RCM press **SEND**.

4. To enable automatic search, press **START**.

5. Deactivate automatic search with ERM press **SEND**.

6. Press **START**. The control performs the auxiliary functions (S-T-M) related to the current status and displays the coordinates of the stop point for each axis. The system goes into Hold status (Hold indicator ON).

To exit from Hold status:

7. Jog the axes from the stop points.

8. Select the Auto mode.

9. Press **HOLD**. The Hold indicator turns OFF.

10. Press **START** to resume execution.

AFTER M.T. POWER OFF AND RESET

If only the machine tool has been turned off or reset, the control clears all the auxiliary functions and the corrections but maintains the axes origins. Therefore, you need not home the axes.

To resume the cycle, repeat steps 2-10 in the above procedure.

AFTER HOLD

To resume the cycle after HOLD, use steps 7-10 in the above procedure.

Important. If no jog move was performed during Hold, you need not return the axis to the stop point. Therefore, only execute steps 9 and 10.

6.17. EXECUTING FROM THE KEYBOARD

To input and execute data from the keyboard:

1. Select the Input by keyboard mode (with mode selector B).

2. Enter the block you want to execute. For example,
G1 X20 Y10.3 F500

The block will appear on the second line of the CRT.

3. Press **START**.

6.18. USING RESET

The reset function stops axes movement, spindle rotation and coolant flow. It clears the control's buffer, restores the absolute origin and causes a tape rewind.

To use the reset function:

1. Select the Reset mode (with mode selector **T9**).

2. Press **START**.

6.19. USING GRAPHICS

The graphic page allows you to show:

- theoretical profiles and profiles translated by the cutter radius;
- points at which fixed cycles are performed;
- points at which axes moves perpendicular to the interpolation plane are performed.

To display the graphic screen, press **P1** while any process partition appears on the CRT.

The graphic display occurs on the lower 2/3 of the page, on a rectangular area that you define with the UCG command. With UCG, you select the display scale that defines the limits of the graphic field.

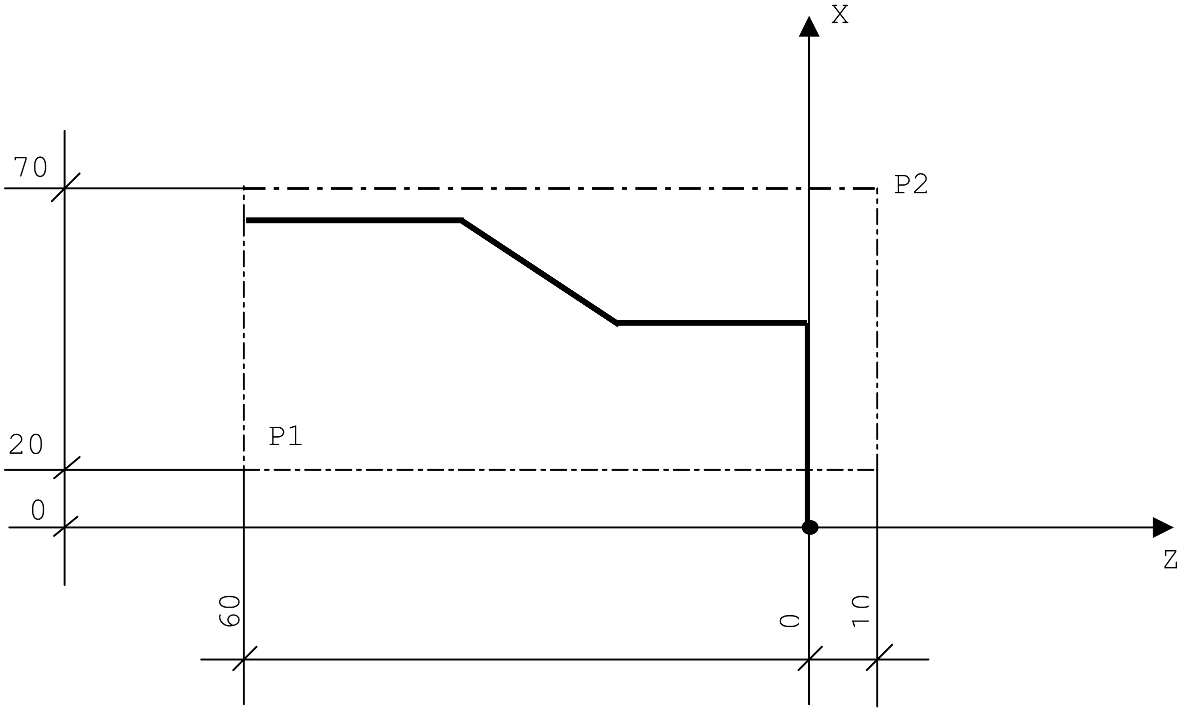
The allowable format is:

UCG,n,AXIS1I AXIS1S,AXIS2I AXIS2S
Press **SEND**

where:

n	defines the display mode n = 1 display not coordinated with axes n = 2 display coordinated with axes
AXIS1I	defines the name and the lower limit of the display for the abscissa axis
AXIS1S	defines the name and the upper limit of the display for the abscissa axis
AXIS2I	defines the name and the lower limit of the display for the ordinate axis
AXIS2S	defines the name and the upper limit of the display for the ordinate axis

Fig. 6.4. - Definition of the graphic field




```
UCG,1,Z-60 Z10,X20 X70
```

Press **SEND**

The zero point in the graphic field corresponds to the lower angle (P1) of the workpiece area to be displayed. The higher angle in the graphic screen depends on the coordinates of P2.

You can generate the graphic display and see the programmed motion in two different modes:

- standstill axes (for program tests);

- moving axes.

GRAPHICS WITH STANDSTILL AXES

To generate this mode, use the following procedure:

1. Enter UAS=1
Press **SEND**

2. Enter this command:

UCG,1,Z..Z..,X..X.., Press **SEND**	-defines graphic field limits with display not coordinated with the axes

You can execute the test in AUTO or SEMIAUTO modes and control feedrate with the feedrate potentiometer.

Machining moves are traced with a continuous line. Rapid moves are dashed lines. The points at which fixed cycles are performed are shown with a diamond.

If there are points in the program that are outside the area defined by UCG, the following message will be displayed: "OUT OF RANGE"

To cancel standstill mode after the program test, enter UAS=0 press **SEND.**

GRAPHIC WITH MOVING AXES

To generate this mode, enter the following command:

UCG,2,Z..Z..,X..X..,Z press **SEND**	-defines the limits of the graphic field with the display coordinated to the axes moves

The graphic display shows the machining rate moves.

Programmed dimensions of the axes appear in the upper part of the screen.

Notes:

• During program testing (UAS=1) and during machining (UAS=0), you can change from graphic to alphanumeric display and back by pressing **P1.**

• To erase the graphic display, enter CLG press **SEND.**

• To disable the graphic display, enter DCG press **SEND.**

7. MESSAGES AND ERROR SIGNALS

This appendix helps you find the meaning of the messages and error signals that appear on the screen. The messages are stored in 4 files:

FILMS1 lists messages that define the errors of operator.
FILMS2 lists messages signals about the errors that are appeared in the execution of commands.
FILMS3 .lists messages related to I/O errors.
FILMS4 lists messages that defined the errors of programming.

7.1. MESSAGES FILMS1 ABOUT THE FCRSYS FILE ERRORS

Table 7.1. - Messages from FILMS1

NUMBER	TEXT
FILMS1 01	LOGIC NAME TABLE IS FULL
FILMS1 02	LOGIC NAME ALREADY DEFINED
FILMS1 03	UNDEFINED NAME
FILMS1 04	PERIPHERAL ERROR
FILMS1 05	FILE NOT FOUND IN MEMORY
FILMS1 06	ILLEGAL RECORD
FILMS1 07	DO NOT USE
FILMS1 08	INVALID COMMAND
FILMS1 09	FORMAT ERROR
FILMS1 10	SYNTAX ERROR
FILMS1 11	UTILITY NOT AVAILABLE
FILMS1 12	LINE ERROR
FILMS1 13	DOES NOT USE
FILMS1 14	ILLEGAL REQUEST
FILMS1 15	FORMAT ERROR IN SECT. 2 FCRSYS
FILMS1 16	DOES NOT USE
FILMS1 17	DOES NOT USE
FILMS1 18	MPX NOT EXIST
FILMS1 19	PATH FOR MPX HAS MORE 40 SYMBOLS (SECT.1 FCRSYS)
FILMS1 20	SECTION 1 DOES NOT CONTAIN MPX MEMORY
FILMS1 21	FCRSYS FILE NOT FOUND
FILMS1 22	NOT EXIST SECTION 1 IN FCRSYS FILE
FILMS1 23	DOES NOT USE
FILMS1 24	MPX ALREADY EXISTS
FILMS1 25	DOES NOT USE
FILMS1 26	MPX NUMBER HAS ERROR
FILMS1 27	DOES NOT USE
FILMS1 28	PROGRAM NOT PRESENT
FILMS1 29	OUT OF MEMORY
FILMS1 30	UTILITY INHIBITED

DESCRIPTION OF ERRORS

FILMS1 01 LOGIC NAME TABLE FULL
The table of logic name assignments already contains 15 logic names.

FILMS1 02 LOGIC HAME ALREADY DEFINED
You tried to enter a logic name already defined in the table.

FILMS1 03 UNDEFINED NAME
The name you entered is not assigned.

FILMS1 04 DOES NOT DECLARE MP1

FILMS1 05 FILE NOT FOUND IN MEMORY
The file specified in your command does not exist in memory.

FILMS1 06 ILLEGAL RECORD
The required program cannot be carried out with the present format.

FILMS1 08 INVALID COMMAND
The control does not recognised the command you entered.

FILMS1 09 FORMAT ERROR
The control recognises your command but the format is incorrect

FILMS1 10 SYNTAX ERROR
The control recognises your command but the syntax is incorrect (the command is displayed in reverse up to the point of error).

FILMS1 11 UTILITY NOT AVAILABLE
The utility you specified is not available in your control.

FILMS1 12 LINE ERROR
Reserved for the system.

FILMS1 14 ILLEGAL REQUEST
In a copy command you tried to copy all the files from a device onto a single file (no source file name)

FILMS1 15 FORMAT ERROR SECTION 2 FCRSYS FILE
Section 2 FCRSYS file does not exist.

FILMS1 18 MEMORY MPX DOES NOT EXIST

FILMS1 19 PATH FOR MPX HAS MORE 40 SYMBOLS
In section 1 FCRSYS file path for MPx contains more 40 symbols

FILMS1 20 SECTION 1 FCRSYS DOES NOT CONTAIN MPX MEMORY
The FCRSYS file has not been declared MPx memory. Memory MP0 and MP1 must be declared always.

FILMS1 21 FCRSYS FILE NOT FOUND
FCRSYS file does not exist

FILMS1 22 DO NOT EXIST SECTION 1 IN FCRSYS FILE

FILMS1 24 MPX EXISTS ALREADY
Number of MPx exists already.

FILMS1 25 DOES NOT USE

FILMS1 26 MPX NUMBER HAS ERROR
The control detects a number of MPx (x=0-6)

FILMS1 27 DOES NOT USE

FILMS1 28 PROGRAM NOT PRESENT
A declared program does not present

FILMS1 29 OUT OF MEMORY
The address specified is not available with <DEBUG>.

FILMS1 30 UTILITY INHIBITED

Table 7.2. – MESSAGES FILMS2 FILE ABOUT EXECUTION ACTS WITH PROGRAM

NUMBER	MESSAGE
FILMS2 01	NOT USED
FILMS2 02	DO YOU CONFIRM? (Y/N)
FILMS2 03	COMMAND EXECUTED
FILMS2 04	EXISTS UPDATE??? (Y/N)
FILMS2 05	EXISTS DELETE??? (Y/N)
FILMS2 06	FILE COPIED
FILMS2 07	WAIT: LINE BUSY

DESCRIPTION OF EDP MESSAGES

FILMS2 01 NOT USED

FILMS2 02 D0 YOU CONFIRM? (Y/N)
The message appears when the whole memory MPx or file must be cleared. Enter Y press **ENTER** to delete all files of MPx (file). Enter N press **ENTER** to return to the COMAND mode. In the MACHINE TOOL CONTROL mode is displayed on the screen after entering the command: CAO, CAC,CTU to deleting of records (record) or initialisation of offsets, origins and tool life files.

FILMS2 03 COMMAND EXECUTED
This message indicates that the control has executed the command you entered.

FILMS2 04 EXISTS UPDATE? (Y/N)
This message appears when you try to copy to a file with preallocated space and fixed length. Type Y press **ENTER** to update all records starting with 1. If you type N press **ENTER**, the new file is not copied.

FILMS2 05 EXISTS DELETE? (Y/N)
The message appears during a copy operation to indicate that the destination memory already has a file with the specified name. Type Y press **ENTER** to cancel the old file and copy the new one. If you type N press **ENTER**, the new file is not copied.

FILMS2 06 FILE COPIED
You copied a file to the display.

FILMS2 07 LINE BUSY
This message appears when RS232 line busy.

7.2. I/O ERROR MESSAGESS FROM FILMS3 FILE

The list of available I/O errors is shown in Table 7.3.

The text of these messages is stored in a file associated to the logic name FILMS3. If this file is not stored the control only displays the error number preceded by the coded FILMS3. For instance, FILMS3 25.

Since these messages are intended mainly for debugging control software, we describe only the messages that appear during operation.

Table 7.3. - I/O errors

NUMBER	MESSAGE
FILMS3 01	INVALID OPERATION
FILMS3 02	PARAMETERS DO NOT MATCH
FILMS3 03	INVALID DEVICE NAME
FILMS3 04	INVALID MECORD NUMBER
FILMS3 05	RECORD OVERFLOW
FILMS3 06	BUFFER OVERFLOW
FILMS3 07	INVALID LOGIC CHANNEL
FILMS3 08	INVALID FLAG NUMBER
FILMS3 09	INVALID FUNCTION CODE
FILMS3 10	UNDEFINED LOGIC NAME
FILMS3 11	CHANNEL ALREADY FREE
FILMS3 12	FILE OPEN
FILMS3 13	FILE CLOSED
FILMS3 14	RECORD ALREADY WRITTEN
FILMS3 15	NO FREE SECTORS
FILMS3 16	FILE ALREADY DEFINED
FILMS3 17	MEMORY OVERFLOW
FILMS3 18	FILE EXISTS ALREADY
FILMS3 19	INVALID ASSIGNMENT OF DATES
FILMS3 20	INVALID OPERATION
FILMS3 21	INVALID ACCES METHOD
FILMS3 22	CHANNEL NOT AVAILABLE
FILMS3 23	RECORD NOT FOUND
FILMS3 24	FILE NOT FOUND
FILMS3 25	PROTECTED FILE
FILMS3 26	WRITE PROTECTED FILE
FILMS3 27	END OF FILE
FILMS3 28	BEGINNING OF FILE
FILMS3 29	DEVICE ALREADY USED
FILMS3 30	DEVICE NOT READY
FILMS3 31	WRITE PROTECTED DEVICE
FILMS3 32	PARITY ERROR
FILMS3 33	BUFFER OVERFLOW
FILMS3 34	DINAMIC MEMORY OVERFLOW
FILMS3 35	HARDWARE ERROR
FILMS3 36	INVISIBLE FILE
FILMS3 38	FORMAT ERROR
FILMS3 39	LINE ERROR

DESCRIPTION OF I/O ERROR MESSAGES

These messages serves to debugging of software, therefore in this document will be presents the messages for operator of machine tool only.

FILMS3 03 INVALID DEVICE
The peripheral specified in the command does not have file-type structure.

FILMS3 05 RECORD OVERFLOW
The input record exceeds the maximum allowable length.

FILMS3 10 UNDEFINED NAME
The logic name you entered is unassigned (section 2 FCRSYS file).

FILMS3 12 FILE OPEN
Indicates that file specified in editor to modifying has been select by the command SPG.

FILMS3 15 NO FREE SECTORS
The specified destination device is full. There is not enough memory to editing of specified file.

FILMS3 17 MEMORY OVERFLOW
There is not enough memory available for the new program or the modified program.

FILMS3 20 INVALID OPERATION
An invalid operation was executed. Example, has been select a program before pressing **START**.

FILMS3 24 FILE NOT FOUND
The file associated to a logic name in section 2 of FCRSYS file does not exist.

FILMS3 26 WRITE PROTECTED FILE
You cannot write on this file because it protected with the ATT command.

FILMS3 30 DEVICE NOT READY
The specified device is not ready for use. The program has been written to the memory that in

FILMS3 35 HARDWARE ERROR
A bubble memory module has a hardware fault.

FILMS3 36 INVISIBLE FILE
The specified file cannot be displayed on the screen for user.

FILMS3 38 LINE ERROR
The required peripheral (RS232) is not properly connected.

7.3. MESSAGES DISPLAYED DURING MACHINING

On the screen are displayed messages that indicate particular conditions or faults of the system (machine tool and control) during machining.

Messages related to program errors are: displayed in reverse. The maximum message length is 32 characters. All errors displayed in reverse are block: in errors: you must execute a reset to exit from the error condition.

Machining messages (Table 7.4.) are stored in a separate file called FILMS4. If this file is not stored, the control displays only the message number, precedes by code FILMS4. For example: FILMS4 72.

Table 7.4. - Machining messages

MESSAGE	TEXT
FILMS4 01	SYSTEM ERRROR
FILMS4 02	WRONG HOMING DIRECTION
FILMS4 03	OUT OF OPERATING LIMITS
FILHS4 04	HARDWARE OVERTRAVEL
FILMS4 05	AUTO RETURN TO PROFILE ILLEGAL
FILMS4 06	UNDEFINED JOG
FILMS4 07	AXIS ON PROFILE
FILMS4 08	MEASUREMENT MIS-MATCH
FILMS4 09	PROBE NOT RELEASED
FILMS4 10	THREADING/TAPPING ERROR
FILMS4 11	SYNTAX ERROR
FILMS4 12	DATA MISSING
FILMS4 13	FORMAT ERROR
FILMS4 14	SYMBOL NAME ERROR
FILMS4 15	UNDEFINED SYMBOL
FILMS4 16	STRING T00 LONG
FILMS4 17	NON EXISTENT FUNCTION
FILMS4 18	INDEX OUT OF RANGE
FILMS4 19	PARAMETER OVERFLOW
FILMS4 20	FUNCTI0N UNDEFINED
FILMS4 21	ILLEGAL NUMBER DF OPERATORS
FILMS4 22	OPERAND NOT PROVIDED
FILMS4 23	ILLEGAL NUMBER OF OPERANDS
FILMS4 24	OPERAND NOT ALLOWED
FILMS4 25	BLOGK DOES NOT MATCH
FILMS4 26	ORIGINS FILE NOT FOUND
FILMS4 27	ORIGIN NOT EXISTENT
FILMS4 28	AXIS NOT NOMED YET
FILMS4 29	AXIS ORIGIN/OFFSET NOT DEFINED
FILMS4 30	PROBING PARAMETERS UNDEFINED
FILMS4 31	RPT NEST >3 OR ERP WITHOUT RPT
FILMS4 32	SUBROUTINE NEST >2
FILMS4 33	EPP NEST >1
FILMS4 34	UNDEFINED LABEL
FILMS4 35	RPT OR EPP CYCLE NOT CLOSED
FILMS4 36	WRONG DATA SET
FILMS4 37	DUPLICATED LABEL
FILMS4 38	TOO MANY PROGRAMS
FILMS4 39	T00 MANY LABELS
FILMS4 40	END OF FILE

Table 7.4. continued

MESSAGE	TEXT
FILMS4 41	TOP OF FILE
FILMS4 42	I/O LOGIC ERROR
FILMS4 43	OFFSET FILE NOT FOUND
FILMS4 44	OFFSET NON EXISTENT
FILMS4 45	S EXCEEDS MAX. RANGE
FILMS4 46	ROTATION RANGE NOT DEFINED
FILMS4 47	TOO MANY EXPEDITED M OR H CODES
FILMS4 48	S NOT DEFINED
FILMS4 49	T NOT DEFINED
FILMS6 50	INDEXING AXIS NOT YET HOMED
FILMS4 51	UNKNOWN M OR H CODES
FILMS4 52	OPERATOR PANEL DISABLED
FILMS4 53	HOLD DISABLED
FILMS4 54	NO PROGRAM SELECTED
FILMS6 55	TOOL LIFE FILE NOT FOUND
FILMS4 56	NO ALTERNATIVE TOOL DECLARED
FILMS4 57	ALTERNATIVE TOOL NOT IN FILE
FILMS4 58	END OF PROGRAM
FILMS4 59	BLOCK NOT ALLOWED
FILMS4 60	SPINDLE AXIS CHANGE NOT ALLOWED
FILMS4 61	MACHINE ON
FILMS4 62	MACHINE OFF
FILMS4 63	COMPUTATION EXCEPTION
FILMS4 64	AXIS OUT OF TOLERANCE
FILMS4 65	SERVO ERROR
FILMS4 66	SKEW ERROR
FILMS4 67	NO LONGER ON HRDW OVERTRAVEL
FILMS4 68	TRANSDUCER ERROR
FILMS4 69	TOO MANY AXES DEFINED
FILMS4 70	DIFFERENT MEASURING UNITS
FILMS4 71	ERROR IN RECORD FORMAT
FILMS4 72	TOOL MONITORING LOOP
FILMS4 73	TOOL MONITORING FILE OVERFLOW
FILMS4 74	AXIS HOMED
FILMS4 75	BLOCK NOT EXECUTABLE IN HOLD
FILMS4 76	M OR H NOT EXECUTABLE IN HOLD
FILMS4 77	UNDEFINED PROFILE
FILMS4 78	PROGRAM TOO COMPLEX TO CALCULATE
FILMS6 79	AXIS DISABLED
FILMS4 80	EXIT FROM HOLD NOT ALLOWED
FILMS4 81	AXES NOT ON PROFILE
FILMS4 82	T NOT ALL0WED

Table 7.4. continued

MESSAGE	TEXT
FILMS4 83	RANDOM TOOL FILE NOT FOUND
FILMS4 84	VIOLATION OF PROTECTED AREA
FILMS4 85	EXPEDITE M OR H NOT ALLOWED
FILMS4 86	FEEDRATE NOT PROGRAMMED
FILMS4 87	DPT PARAMETERS MISSING
FILMS4 88	UNDEFINED CANNED CYCLE
FILMS4 89	END OF CONTROLLED SEARCH
FILMS4 90	MIXED TOOL CHANGE
FILMS4 91	WRONG OFFSET PARAMETERS
FILMS4 92	MAS FILE NOT FOUND
FILMS4 93	MAS RECORD NOT FOUND
FILMS4 94	INCOMPLETE PROGRAM
FILMS4 95	PROBING OPERANDS DEACTIVATED
FILMS4 96	AXIS NOT BELONGING TO WORK PLANE
FILMS4 97	G CODE NOT ALLOWED
FILMS4 98	G CODE NOT ALLOWED DURING G95
FILMS4 99	WRONG OPERATION ON FILE
FILMS4 100	WRONG CHANNEL
FILMS4 101	ILLEGAL OPERATION ON FILE
FILMS4 102	USER DISPLAY ERROR
FILMS4 103	FUNGTION NOT CONFIGURED
FILMS4 104	INSUFFICIENT MEMORY
FILMS4 105	CONTROL NOT AVAILABLE OR OFF
FILMS4 106	ERROR IN SERIAL TRANSMISSION/RECEPTION
FILMS4 107	I/O SERIAL RECORD TOOL LONG
FILMS4 108	SELECTED MODE ILLEGAL
FILMS4 109	NO BLOCKS BEHIND MBR
FILMS4 110	PROBE INPUT SIGNAL NOT COHERENT

DESCRIPTION OF PROCESS MESSAGES

FILMS4 01 SYSTEM ERROR
Call the Field Service, specifying the exact machining conditions in which the error has been detected.

FILMS4 02 WRONG HOMING DIRECTION
With the mode selector on home position search. You have tried to home an axis in the opposite direction to that declared in the 5 section of AXCFIL file.

FILMS4 03 OUT OF OPERATING LIMITS
This message appears when you programmed a movement greater than allowed in the LOP (Lox) instruction of AXCFIL file or have been programmed in work (instruction DLO).

FILMS4 04 HARDWARE OVERTRAVEL
One of the end of stroke microswitches has been pressed.

FILMS4 05 AUTO RETURN TO PROFILE ILLEGAL
Using three-letter code RAP=1, there is an attempt to return to the profile, even if more than 32 elementary movements in manual have already been carried out.

FILMS4 06 UNDEFINED JOG
With the potentiometr JOG, there is an attempt to carry out a movement greater than allowed by the operating limits.

FILMS4 07 AXIS ON PROFILE
Operation of return to profile carried out for the selected axis.

FILMS4 08 MEASUREMENT MISMATCH
During a measuring cycle, there has been no probe contact.

FILMS4 09 PROBE NOT RELEASED
The probe has not been released during a measuring cycle.

FILMS4 10 THREADING/TAPPING ERROR
This message appears during a threading or tapping cycle for one of the allowing reasons:
a) idle spindle;
b) too short thread length;
c) incongruent spindle rotation direction.

FILMS4 11 SYNTAX ERROR
This message appears if, in a part program block or in a block entered from the keyboard, the following conditions are present:
a) illegal character;
b) illegal geometric definition;
c) you have programmed a switched-over axis, that during configuration had been declared as a direct axis;
d) during a probing cycle (G72), more than one axis of the interpolation plane has been specified;
e) closing bracket -)- redundant or missing.

FILMS4 12 OPERANDS OR OPERATORS MISSING

This message appears if, in a part program block or in a block entered from the keyboard, a necessary operand or operator is missing.

FILMS4 13 FORMAT ERROR

This message appears when the format of the numerical value associated to an operand or an operator is not allowable one.

FILMS4 14 SYMBOL ERROR

FILMS4 15 UNDEFINED SYMBOL

Symbol not present in the system table or pressed the ENTER pushbutton but must be pressed START.

FILMS4 16 STRIN6 TOO LONG

Number of characters in the string greater than allowed.

FILMS4 17 NONEXISTING FUNCTION

Functions G. H. M. not coded.

FILMS4 18 INDEX OUT OF RANGE

This message appears if:

a) tool number is greater than 9999;

b) offset number is greater than 9999;

c) G, H, M is greater than 99;

d) in the part program number of points, lines and circles are greater than have been defined in characterization (section 2 of PGCFIL file).

FILMS4 19 PARAMETER OVERFLOW

This message appears when number of G, M is greater than 99.

FILMS4 20 UHDEFINED FUNCTION

Call for synchronisation not compatible with system status. For example: in one block are programmed G-functions from one class.

FILMS4 21 ILLEGAL NUMBEM OF OPERATORS

This message appears if more than 9 G functions are present in the block.

FILMS4 22 OPERAND NOT PROVIDED

This message appears if:

a) there are inhibited operands in a fixed measuring or tool wear check cycle;

b) K is inhibited in cycle G84 with spindle without transducer;

c) R is present in a cylindrical thread.

FILMS4 23 ILLEGAL NUMBEM OF OPERANDS

This message appears if:

a) more than two axes are programmed in a threading cycle;

b) the number of items associated to a function is different from the required;

c) too many M are programmed in a block (max. 4);

d) too many indexing axes are programmed (max. 3).

FILMS4 24 OPERAND NOT ALLOWED

This message appears if:

a) a wrong item is programmed in a function;

b) an axis is specified together with and the switched-over one;

c) an operand not congruent with the previous operands has been programmed;

d) an operand inhibited by the G function class present in the block has been programmed;

e) an axis operand is duplicated.

FILMS4 25 UNDEFINED COMMAND

This message appears if:

a) a G not congruent with the active program status is present (IS0, OFFSET, GTL, ...);
b) a G not congruent with the active dynamic mode is present (point-to-point, continuous mode etc.);
c) the active program status is not congruent with the interpolation mode (G00, GO1, GO2,G03, G33);
d) the active program status is not congruent with the programming mode (absolute, incremental, referred to machine zero);
e) auxiliary functions not congruent with the active dynamic mode are present;
f) auxiliary functions not congruent with the active program status are present;
g) auxiliary functions not congruent with the interpolation mode are present (GOO, GO1, GO2, G03, G33);
h) the program status (ISO, GTL, fixed cycles, tool wear check cycle. measuring cycle) is not congruent with the OFFSET mode (G41, G42, G40).

FILMS4 26 ORIGINS FILE NOT FOUND

This message appears when there is an attempt to refer an axis without editing the file of origins first.

FILMS4 27 ORIGIN NONEXISTENT

This message appears when an origin not stored in the file of origins is recalled.

FILMS4 28 COMMAND AXIS NOT HOMED YET

This message appears when there is an attempt to move an axis that has not been homed yet.

FILMS4 29 UNDEFINED AXIS IN ORIGIN/OFFSET

This message appears when you specify an axis whose origin and or offset has not been defined yet.

FILMS4 30 PROBING PARAMETERS UNDEFINED

This message appears when you execute a probing cycle without entering the parameters associated to the DPT command first.

FILMS4 31 RPT NESY >3 OR ERP WITHOUT RPT

This message appears if you program more than 3 repeat levels or a repeat end without repeat start.

FILMS4 32 SUBROUTINE NEST >2

This message appears if the subroutine nesting level in a part program is greater than 2.

FILMS4 33 EPP NEST >1

This message appears in case of EPP nesting in a part program.

FILMS4 34 UNDEFINED LABEL

This message appears when the label has not been defined.

FILMS4 35 RPT OR EPP CYCLE NOT CLOSED

This message appears if in a part program the end of file is attained without closing any open RPT and/or EPP cycle.

FILMS6 36 WRONG DATA SET

This message appears if:

a) you specified a non existent device;
b) you specified a non existent program name or subroutine name.

FILMS6 37 DUPLICATED LABEL
A label has been defined more than once.

FILMS6 38 TOO MANY PROGRAMS
You have defined a number of programs greater than declared in the configuration.

FILMS6 39 T00 MANY LABELS
You have defined a number of label declared in the configuration.

FILMS4 40 END OF FILE
An EOF has been found during part program scanning or execut

FILMS4 41 TOP OF FILE
A TOF has been found during part program scanning.

FILMS4 42 I/O LOGIC ERROR
Call the Field Service specifying the exact machining conditions in which the error has been detected.

FILMS4 43 OFFSET FILE NOT FOUND
This message appears when an offset is recalled without creating the offset file first

FILMS4 44 OFFSET NONEXISTENT
This message appears when the specified offset is not stored in the offset file.

FILMS4 45 S EXCEEDS MAX RANGE
The S programmed with USS exceeds the max. Number of revolutions for the specified range.

FILMS4 46 ROTATION RANGE NOT DEFINED
You have not specified the range of spindle rotation.

FILMS4 47 TO0 MANY EXSPEDITED M or H CODE:
You have programmed too many expedited M or H functions.

FILMS4 48 S NOT DEFINED
When programming spindle rotation you have not specified S.

FILMS4 49 T NOT DEFINED
When programming M06 you have not specified T.

FILMS4 5O INDEXING AXIS NOT HOMED
There is an attempt to move an indexing axis without homing it first.

FILMS4 51 UNKNOWN M or H code
You have programmed uncoded M or H functions.

FILMS4 52 O.P. CONSOLE DISABLED
This message appears when trying to operate on pushbuttons or selectors that are temporarily deactivated by the machine logic.

FILMS4 53 HOLD DISADLED
This message appears when trying to enter/exit from the Hold status and the Hold pushbutton is temporarily deactivated by the machine logic.

FILMS4 54 NO PROGRAM HAS BEEN SELECTED
There is an attempt to execute a part program in AUTO/STEP without entering an SPG command first.

FILMS4 55 TOOL LIFE FILE NOT FOUND
The specified tool life file has not been created yet.

FILMS4 56 NO ALTERNATIVE TOOL DECLARED
The specified alternative tool has not been declared in the tool life file.

FILMS4 57 ALTERNATIVE TOOL INEXISTENT
Alternative tool not present in the tool magazine.

FILMS4 58 END OF PROGRAM
This messages appears when program execution has been accomplished.

FILMS4 59 BLOCK NOT ALLOWED
You are trying to execute a block not allowed from keyboard or part program.

FILMS4 60 SPINDLE AXIS CHANGE NOT ALLOWED
There is an attempt to change the spindle axis with a not allowed program mode.

FILMS4 61 MACHINE ON
This message appears on line 0 at machine tool power-up.

FILMS4 62 MACHINE OFF
This message appears on line 0 at machine tool power-off.

FILMS4 63 COMPUTATION EXCEPTION
The control has detected a computation exception

FILMS4 64 AXIS OUT OF TOLERANCE
The specified axis is out of position tolerance.

FILMS4 65 AXIS SERVO ERROR
This message indicates the name of the axis in servo error.

FILMS4 66 SKEW ERROR
This message indicates the name of the axis in skew error.

FILMS4 67 NO LONGER ON HARDWARE OVERTRAVEL
This message appears when the limit switch is released.

FILMS4 68 TRANSDUCER ERROR
A count error has been detected in transducers.

FILMS4 69 TOO MANY AXES DEFINED
A number of axes greater than allowed has been associated to an origin.

FILMS4 70 DIFFERENT MEASURING UNITS
This message appears when there is an attempt to modify a record of the origins or offsets file with values in measuring units different from those declared in the record definition.

FILMS4 71 ERROR IN RECORD FORMAT
This message appears when the record format of the origins or offsets file is not allowed.

FILMS4 72 TOOL MONITORING LOOP
This message appears when in the tool life file a pair of tools alternative to each other have been declared. For example, main tool T1, alternative tool T2, main tool T2, alternative tool T1.

FILMS4 73 TOOL MONITORING FILE 0VERFLOW
This message appears if there is an attempt to insert a record in the tool monitoring file, which is already completely defined.

FILMS4 74 AXIS HOMED
This message appears when the search for the zero microswitch has been completed.

FILMS4 75 BLOCK NOT EXECUTABLE IN HOLD
There is an attempt to execute a program block that cannot be performed in Hold status.

FILMS4 76 M OR H NOT EXECUTABLE IN HOLD
There is an attempt to execute a logic function that cannot be carried out in Hold status.

FILMS4 77 UNDEFINED PROFILE
This appears if:
a) a geometry incongruence has been programmed;
b) there is no intersection between geometric items;
c) there is a geometry reversal;
d) the GTL or OFFSET closure is not congruent;
e) there are wrong probing parameters;
f) the fixed cycle is not correct.

FILMS4 78 PROGRAM TOO COMPLEX
You have programmed a profile including more than five elements.

FILMS4 79 AXES DISABLED
There is an attempt to manually move an axis using UAS=1.

FILMS4 80 EXIT FROM HOLD NOT ALLOWED
This message appears when exit from the Hold status is not allowed by the machine logic.

FILMS4 81 AXES NOT ON PROFILE
There is an attempt to exit from the Hold status without manualy moving the axes to the profile.

FILMS4 82 T NOT ALLOWED
This message appears when you program a tool not stored in the tool monitoring file.

FILMS4 83 RANDOM TOOL FILE NOT FOUND
This message appears when you program a tool without storing the random tool file.

FILMS4 84 PROTECTED AREA VIOLATION
You programmed a profile that violates the protected area.

FILMS4 85 EXPEDITED M OR H NOT ALLOWED
An expedited auxiliary function has been programmed in a block without movement.

FILMS4 86 FEEDRATE NOT PROGRAMMED
There is an attempt to move the axes at machining rate without programming feedrate.

FILMS4 87 DPT PARAMETERS MISSING
There is an attempt to execute a probing cycle without entering code DPT first.

FILMS4 88 UNDEFINED FIXED CYCLE
This message appears if a fixed cycle is programmed on an axis that has not been declared as spindle.

FILMS4 89 END OF CONTROLLED SEARCH
This message appears when the system is in controlled search and the cursor is positioned on the latest executed block.

FILMS4 90 INCORRECT TOOL CHAHGE
This message appears if a standard tool has been programmed while a special tool is on the spindle (a special tool is a tool occupying more than one station of the magazine).

FILMS4 91 WRONG OFFSET PARARETERS
This message appears if you try to operate on the offset file with parameters not present in the record associated to the offset.
For example, UCA, offset No.,X10 and X is not stored in the record associated to the offset number

FILMS4 92 MAS FILE NOT FOUND
This message appears when trying to execute a record without creating the MAS file first.

FILMS4 93 MAS RECORD NOT FOUND
This message appears when trying to execute a record not present in the MAS file.

FILMS4 94 INCOMPLETE PROGRAM
This message appears when the end of file is attained, but GTL, OFFSET, fixed cycles, etc. Are still open.

FILMS4 95 NOT USED

FILMS4 96 AXIS NOT BELONGING TO WORK PLANE
This message appears when a you program an helix without programming the axis perpendicular to the machining axis.

FILMS4 97 G CODE NOT ALLOWED
This message appears if you program G95 or G96 for a spindle without transducer.

FILMS4 98 G CODE NOT ALLOWED WITH G95
This message appears if you program G86 or G89 while G85 is active.

FILMS4 99 WRONG OPERATION ON FILE
This message appears if:
a) the specified file name already exists;
b) you try to cancel a non existing file;
c) you try to call for a file without having closed the Edit mode.

FILMS4 100 WRONG CHANNEL
This message appears if the specified channel is not configured.

FILMS4 101 ILLEGAL OPERATION ON FILE
This message appears if the required operation on a channel or a data structure is not allowable.

FILMS4 102 USER SCREEN ERROR
This message appears if:
a) the format of the command is not allowable;
b) the screen area is not defined;
c) the screen has not been specified.

FILMS4 103 FUNCTION NOT CONFIGURED
This message appears if you enter an ASSET command without having configured the
corresponding PGCFIL area.

FILMS4 104 INSUFFICIENT MEMORY
The available free sectors do not allow you to create the specified program.

FILMS4 105 CONTROL NOT AVAILABLE 0R OFF
The peripheral selected for transmission/reception is off or not available.

FILMS4 106 ERROR IN SERIAL TRANSMISSION/RECEPTION
The characters transmitted/received exceed the programmed number or there is a transmission/reception error.

FILRS4 107 I/O SERIAL RECORD TOO LONG
The DER instruction programs more than 240 bytes or more than 240 characters

FILMS4 108 SELECTED MODE ILLEGAL
The selected mode is not compatible with the displayed process.

FILMS4 109 NO BLOCKS BEHIND MBR
The operator calls an MBR from the first program block.

FILMS4 110 PROBE INPUT SIGNAL NOT COHERENT
The status of the probe input (OFF) is not coherent with the characterized status.

7.4. MACHINE LOGIC MESSAGES

PLC allows you to customise the interface between the control and the machine tool.

With PLC, you can display as many as 255 machine logic messages having 32 characters each. Machine logic messages are displayed on “FILMS5” lines (see fig. 2.6.) of videopages 1, 7. The text of these messages is stored in FILMS5 file. In the pages that follow, we provide you with forms for listing and describing messages 0÷63. When filling in these modules, the MTM or the end user must follow the documentation supplied with the control.

The end user can also implement specific M functions to be used with a particular system. The M functions described in this manual are typical. You may use specific M functions differently.