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

PROGRAMMING AND OPERATOR'S MANUAL

Sankt-Petersburg
2008

PUBLICATION ISSUED BY:

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

Fax: +7(812)744 70 59
Tel: +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.

CONTENTS

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
- Simultaneous control of up to 3 axes
- 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).
- Configurable 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. During cycle execution you can select as many as 5 alphanumeric screens, displaying the program name, the cycle duration, operator messages, current and programmed dimensions, G, T, S, M, H functions, origins, offsets, power absorption, current block number, repeat cycles and subroutines.

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 though in a smaller body size.
During machining, the remaining part of the screen displays the cartesian axes, programmed dimensions, profiles, the points on which fixed cycles are due and the movements of the axis perpendicular to the active 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 64,000 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.

Operating modes

You can select
the desired operating mode by means of key on the operator panel.
The allowable options are:
- execution of blocks input via keyboard
- automatic program execution
- semiautomatic program execution
- manual continuous jog
- manual incremental jog
- automatic restart after a cycle stop followed by manual jogging.

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 nigh 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 256 compensation points on each axis.

Transducers

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

Absolute origins

The code AXO allows you to define absolute origins. The format is:
AXO,n,X..,Y..,Z..

where:
n origin number (0÷99)
X,Y,Z machine axes.

These origins can be recalled from a program by means of the UAO code. Origins can be specified in either the

current measuring unit of the machine or the alternative unit.
To select the alternative unit, input the origin number with a negative sign.
For example, AXO,-n,X..,Y..,Z...

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.

Tool offsets

Tool offsets must be defined during installation.
You can define an unlimited number of tool offsets. The maximum values are:
- Z=±9999.999 mm for length offsets
- K= 999.999 mm for diameter offsets

The tool length offset can be applied to any axis. Length values can be either input via keyboard or automatically calculated by the control (tool presetting).
Diameter offset values must be input via keyboard.
Current offset values can be displayed or modified at any time. They can be modified from a program, after a probing cycle.
Offsets can be punched on tape or recorded on magnetic cassette, and afterwards re-loaded from tape or magnetic cassette.

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=0). In either

case, the axis repeats the path performed performed during the removal (RAP=1).
The maximum number of displacements is 32.

Reaming and facing heads

These heads are mounted on the spindle and are handled as simultaneous and coordinated axis.
The head is programmed in diameters.

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 radius and centre of a circle in the plane (with G73).
The resulting values are stored in the E variables contained in the measuring block.
With the G74 function, a tool mounted in the spindle and a fixed probe, you can program the survey of deltas between theoretical and actual points.
You can use a G74 cycle for tool requalification or monitoring.

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 ife
- minimum theoretical life
- remaining life
- tool condition

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, even 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

System ambient parameters, as well as machine tool parameters (e.g. speed, acceleration, etc.), tool length and diameter offset values, origins and part programs, are stored on EPROM (Erasable Programmable Read Only Memory).
The EPROM contents can be modified or updated by Balt-System Ltd. engineers only.

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

Part programs can be stored in read/write CMOS RAM memory. In case of power failure, the CMOS RAM contents remain unchanged for 1000 hours.

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 disfunction to be corrected has been noticed.
Diagnostic messages are stored in the system 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. Example: X-20.275
Tape code	EIA RS-244, ISO 840 with automatic acknowledge.
Programming format	N4, G2, X/Y/Z/A/B/C/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 +O.OOO1 to ±99999.9999.
I J coordinates	They allow you to progam 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 o 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

G00 Rapid axis positioning
G01 Linear interpolation
G02 Circular interpolation CW
G03 Circular interpolation CCW
G04 Dwell at end of block
G09 Deceleration at end of block
G17 Selects the interpolation plane defined by the first and second configured axes.
G18 Selects the interpolation plane defined by the first and third configured axes.
G19 Selects the interpolation plane defined by the second and third configured axes.
G20 Closes 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
G73	Measure of hole parameters
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
G85	Reaming cycle
G86	Boring cycle
G89	Boring cycle with dwell
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

Auxiliary H user functions

These functions allow you to expand the set mf auxiliary functions provided by the ISO Standard (M00...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.
The typical fixed cycle movements are:

1) rapid positioning on the hole axis

2) rapid to the work plane (R)

3) machining feedrate up to the programmed Z

4) machining functions at the bottom of the hole

5) rapid or feedrate return to R. You can set a final point R2 different from R (this having two R's within the block)

The available fixed cycles are listed in Table 1.1.

Table 1.1. - Fixed cycles

Fixed cycle	Approach	Bottom end funct. Dwell	Bottom end funct. Spindle rotation	Return
G81 drilling	machining	no	feedrate	rapid
G92 spot-facing	machining	yes	feedrate	rapid
G83 deep drilling (with chips discharge)	intermittent machining	no	feedrate	rapid
G84 tapping	machining spindle rotat.	no	rotation reversal	machin.
G85 reaming or tapmatic	machining	no	feedrate	machin.
G86 boring	machin. - spindle rotat.	no	stop	rapid
G89 boring with spot-facing	machining	yes	feedrate	machin.

Fixed tapping cycle with spindle transducer

In this cycle, the control automatically calculates feedrate according to the spindle speed and the tap pitch (K). Therefore, you need not specify F.

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
(SCF,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 characterize the type of thread.
The allowable formats are:

G33 Z..K..	cylindrical thread with constant pitch
G33 Z..U..K..	conical 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
Z,U	coordinates of the final point
K	thread pitch
Ic	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=0

Tool radius paraxial offset

By programming tool offset factors u, v, w within the machining block, you can obtain the final point offset for the machine cartesian axes.
The final point is calculated as follows:

Pi = Qi + r * Fi

where:
Qi = programmed axis coordinates
r = tool radius
Fi = offset factor; it can be:
u for axis 1
v for axis 2
w for axis 3

Mirror machining

MIR allows you to program mirror machining for all the coordinated axes.

Example:

```
..........
(MIR,X)
..........
(MIR,Y)
..........
(MIR,X,Y)
```

Rotation in the plane

URT allows you to 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 1.2

Table 1.2. - E parameters

Format	Parameters	Min/max value
BY (byte)	E0..E9	0 to 255
IN (integer)	E10..E19	-32768 to +32768
LI (longinteger)	E20..E24	-2.147.483.647 to +2.147.483.647
RE (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)	comon logariphm 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 customized 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.) characterizing the working cycle can be replaced by parameters, the values of which must have been defined in the main program, before calling the subroutine.

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 1.3.

Table 1.3. - Branching codes

Code	Meaning
(BNC,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 E5 N40
(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-G73-G74, you can program 3 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 Y50 E32
  the values calculated for X and Y are stored in E32 and E33

- G73 measures the parameters of a hole in the current interpolation plane (coordinates of the center and radius) and stores the results in the sequence of E variables defined by the block.

This function includes probe radius compensation.
Example:
G73 R100 E35
the abscissa, ordinate and radius of the circle are respectively stored in E35-E36-E37

- 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.
E.g. (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 RIF starting from byte No.7 of the K buffer
SA3.BY=255	give value 255 to byte No.3 of the R 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 beeing exceeded.

Operating field limits

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

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+,Y- Y+)

where:
n = number of the protected area (1 to 3)

X- = lower limit for X
X+ = upper limit for X
Y- = lower limit for Y
Y+ = upper limit for Y

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,Y5 Y100)
(DSA,2,X-100 X-50, Y-20 Y150)
(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 radii 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 1.4.

Table 1.4. - GTL elements

Element	Definition	Description
Reference origins	on=XYa	
Points	pn=[om] X Y	point with cartesian coordinates
	pn=[om] m a	point with polar coordinates
	pn=clm,clp	intersection of two straight lines
	pn=clm,ccp[,s2]	intersection line-circle
	pn=ccm,clp[,s2]	intersection circle-line
	pn=ccm,ccp[,s2]	intersection of two circles
Straight lines	ln=[om]I J r,[op]I J r	line tangent to two circles
	ln=[om]X Y,[op]X Y	line through two points
	ln=[om]I J r,[op]X Y	line tangent to a circle and through a point
	ln=[om]X Y,[op]I J r	line through a point and tangent to a circle
	ln=[om]I J r,a	line tangent to 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=ccm,ccp	line tangent to two circles
	ln=ccp,pm	line tangent to a circle and through a point
	ln=pm,ccp	line through a point and tangent to a circle
	ln=pm,pq	line through two points
	ln=ccm,a	line tangent to 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 1.4. continued

Element	Definition	Description
Circles	cn=[om]I J r	circle with cartesian coordinates
	cn=[om]m a r	circle with polar coordinates
	cn=clm clp,r	circle of given radius tangent to two lines
	cn=clm,ccp,r	circle tangent to one line and one circle of given radius
	cn=ccp,clm,r	circle of given radius tangent to one circle and one line
	cn=pm,clp,r	circle of given radius through a point and tangent to a line
	cn=clp,pn,r	circle of given radius tangent to a line and through a point
	cn=ccm,ccp,r	circle of given radius tangent to two circles
	cn=pm,ccp,r	circle of given radius through a point. tangent to a circle
	cn=ccp,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,clp	circle with center 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=ccm,+d	concentric circles at a given distance
	cn=pm,ccp[,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 I50 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,XC,PV,r)

where:

X real linear axis
C real rotary axis
P virtual abscissa axis
v virtual ordinate axis
r minimum radius

The minimum radius establishes the area the tool is not allowed to enter.

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,B,W,r)

Where:

B real rotary axis
W virtual axis
r radius of the cylinder

Dual Axes

The UAV code also allows you to establish dual axes, i.e. define "masters" and "slaves". To move dual axes, you must only program the motion of the master: the slave will automatically move after the master. With dual axes you can also program mirror machining.

You establish dual axes by programming UAV with modality 3. The allowable format is:

(UAV,3,SLAVE NAME,MASTER NAME,MASTER-SLAVE CORRESPONDENCE,MIRROR)

For example:

(UAV,3,VWU,XYZ,123,124)

where:

3 dual mode
VWU slaves (from 1 to 7 characters)
XYZ masters (from 1 to 4 characters)
123 numeric string that defines the correspondence between the axes. The value of the digit defines the master and the position defines the slave
121 numeric string that characterizes the type of movement:
1 - normal movement
2 - mirror movement
The position of the digit defines the slave.

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

Either a parallel or serial line are available.

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.

Table 1.5. - 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	Rotation of the machining plane around the current origin
SCF	Scale factor
RQO	Reapply origin
RQU	Reapply
RQP	tool
DPI	Define interpolation plane
DTL	Define positioning tolerance
DLO	Define operating limits
DIS	Display variable
TOF	Tool Off
UCG	Define graphic scale
CLG	Reset videographic 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/disable virtual modality

Table 1.5. 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

Table 1.6. - Assignment blocks 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	Meal 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

Table 1.7. - 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 the videographic initialization parameters
CLG	Videographic field reset
DCG	Videographic disable
CAC	Delete tool offset / the whole offset file
SPG	Program selection
REL	Release a selected program
MBR	Enable/disable retracing
RCM	Enable controlled search
ERM	Disable controlled search
DPT	Probe parameters definition
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 initialization
CRE	Create a file with fixed length
FOR	Create file of offsets, tool life, origins, with fixed length, and formatted fields
DIS	Display a file
ATT	Protect a program
DIF	Check a stored program

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 EPROMs to be installed on the memory modules that contain the system software.

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

This programming technique allows you to easily customize, modify and update the interface. The result is a much more reliable 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)

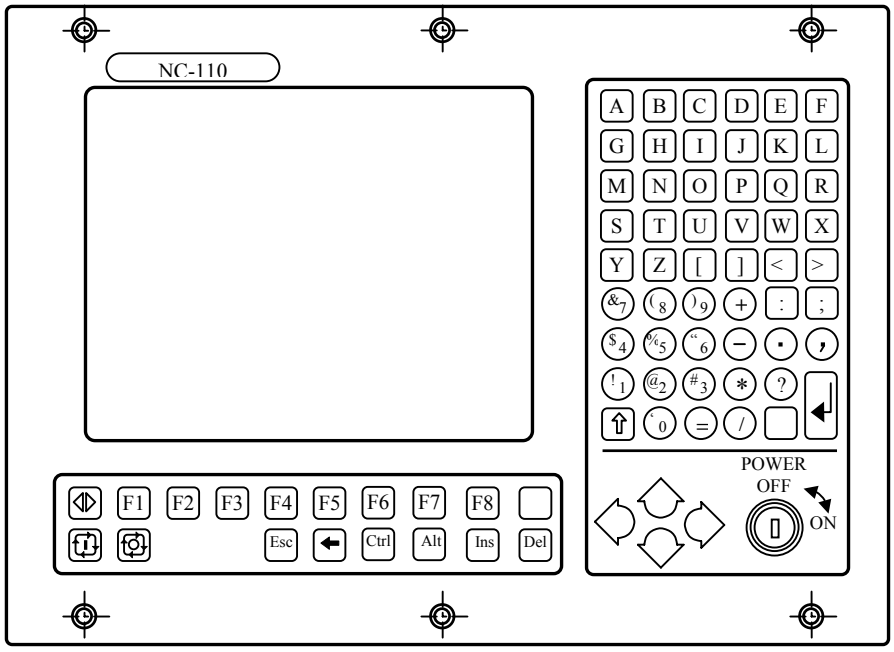


Fig. 2.2. – THE CONSOLE PANEL WITH CRT

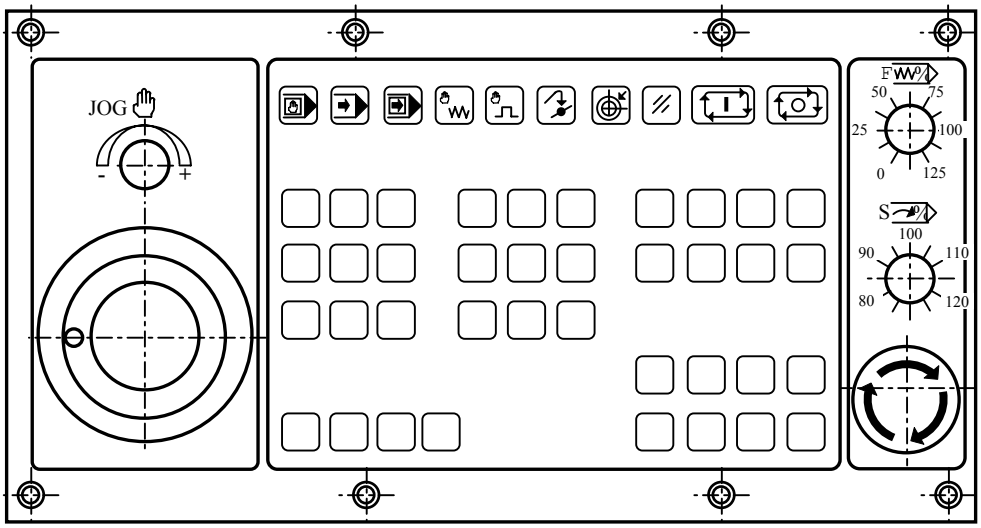


Fig. 2.3. – THE OPERATOR PANEL FOR NC-110 (TFT)

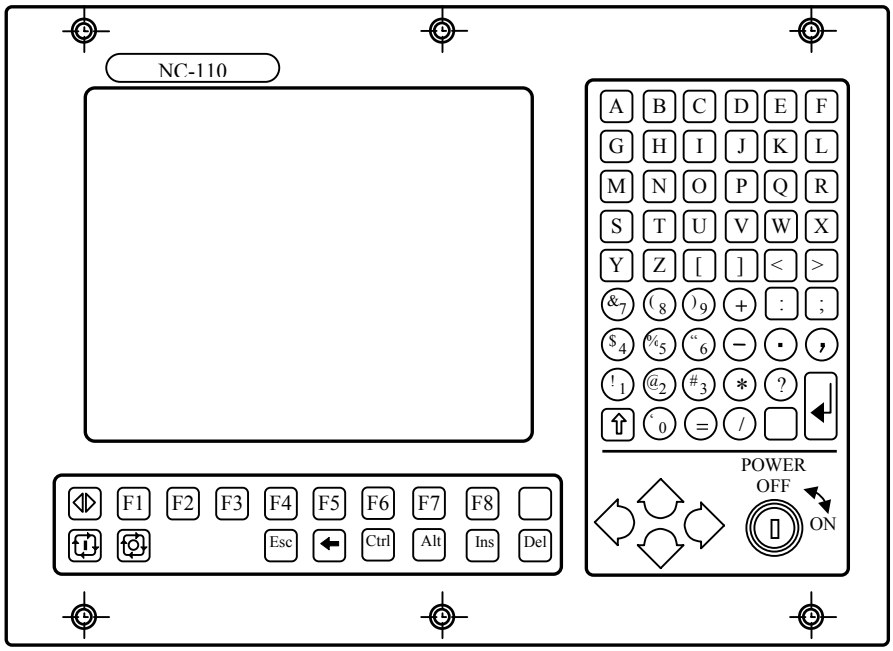
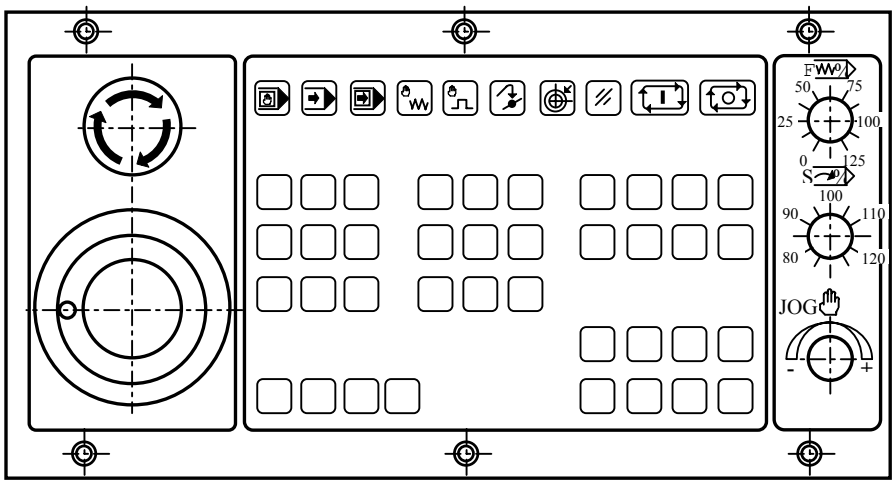


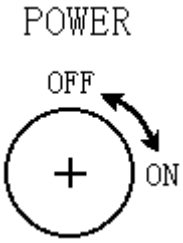
Fig. 2.4. – THE CONSOLE PANEL FOR NC-110 (WITH TFT)



2.2. Functional purpose of keyboard

2.2.1. Functional purpose of operator panel pushbuttons

The functions of SOFT KEYs **F1÷F8** are presentend 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.

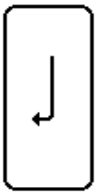
LINE BACK

- 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.

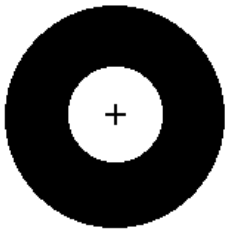
Alt

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

Ins

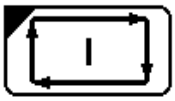
In CNC it does not used.

2.2.2. The pushbuttons of console



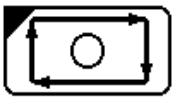
EMERGENCY STOP (red pushbutton)

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

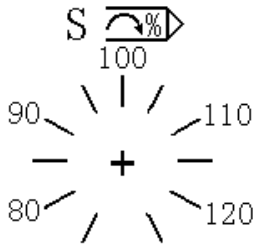
Initia tesapart program execution in the AUTO and STEP modes or separate Block execution in the MDI mode, and axes motion in the MANJ, MANU, PROF and HOME modes. Executes reset of system when control is in the RESE mode.



HOLD

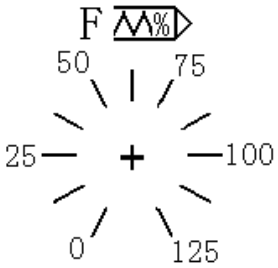
Establishes a stop after a controlled deceleration. To resume the cycle, press HOLD and the CYCLE START. This pushbutton does not function during a threading cycle.

2.2.3. Switches of a console



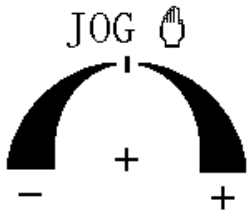
SPEED OVERRIDE - potentiometer

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

```
MACHINE TOOL CONTROL                00:00:00                                  #1
Input-editing string
   FACT            PROGRAM     AT Ofs    NAME PROGRAM     RPT        EPP       BLOCK №
X+xxxxx.xxxx     +xxxxx.xxxx  A  00A    XXXXXX/MPx    xx xx xx     xxxxxx      xxxx
Y+xxxxx.xxxx     +xxxxx.xxxx  O  00A    XXXXXX/MPx    xx xx xx     xxxxxx      xxxx
Z+xxxxx.xxxx     +xxxxx.xxxx  S  00A    XXXXXX/MPx    xx xx xx     xxxxxx      xxxx
A+xxxxx.xxxx     +xxxxx.xxxx     00A
B+xxxxx.xxxx     +xxxxx.xxxx     00A
C+xxxxx.xxxx     +xxxxx.xxxx     00A
D+xxxxx.xxxx     +xxxxx.xxxx     00A
Executed block indication PP                 The text PP with a «running» line
F xxxxx.xxxx   000.0%  xxxxx.xxx
              +000.0%
S xxxxx.xxxx   000.0%  xxxxx.xxxx
T xxxx  ↓ xxxx   T xxxx  ↓ xxxx
L +xxxxx.xxxx  K  +xxxxx.xxxx
P +xxxxx.xxxx                        ESE=        MBR UAS USB  UAV=0  MUSP  IDLE
                                     URT=+0.000  RAP URL UVR  UEP    COMU  [I]
                                     URP=+0.000  RCM USO VOL  UCV=0  CEFA  [O]
G xx xx xx xx xx xx xx xx            Films from PP
  xx xx xx xx xx xx xx xx
M xx xx xx xx xx xx xx xx            Films 4__xx
  xx xx xx xx xx xx xx xx            Films 5__xx
JOG=xxxxx.xxxx   D=xxxxx.xxxx        Films 5__xx
1 | 2 Video page | 3 Displace-ment correct | 4 Input correct | 5 Send to logic | 6 Move block | 7 | 8
```

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 disconnnected
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	memorized 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

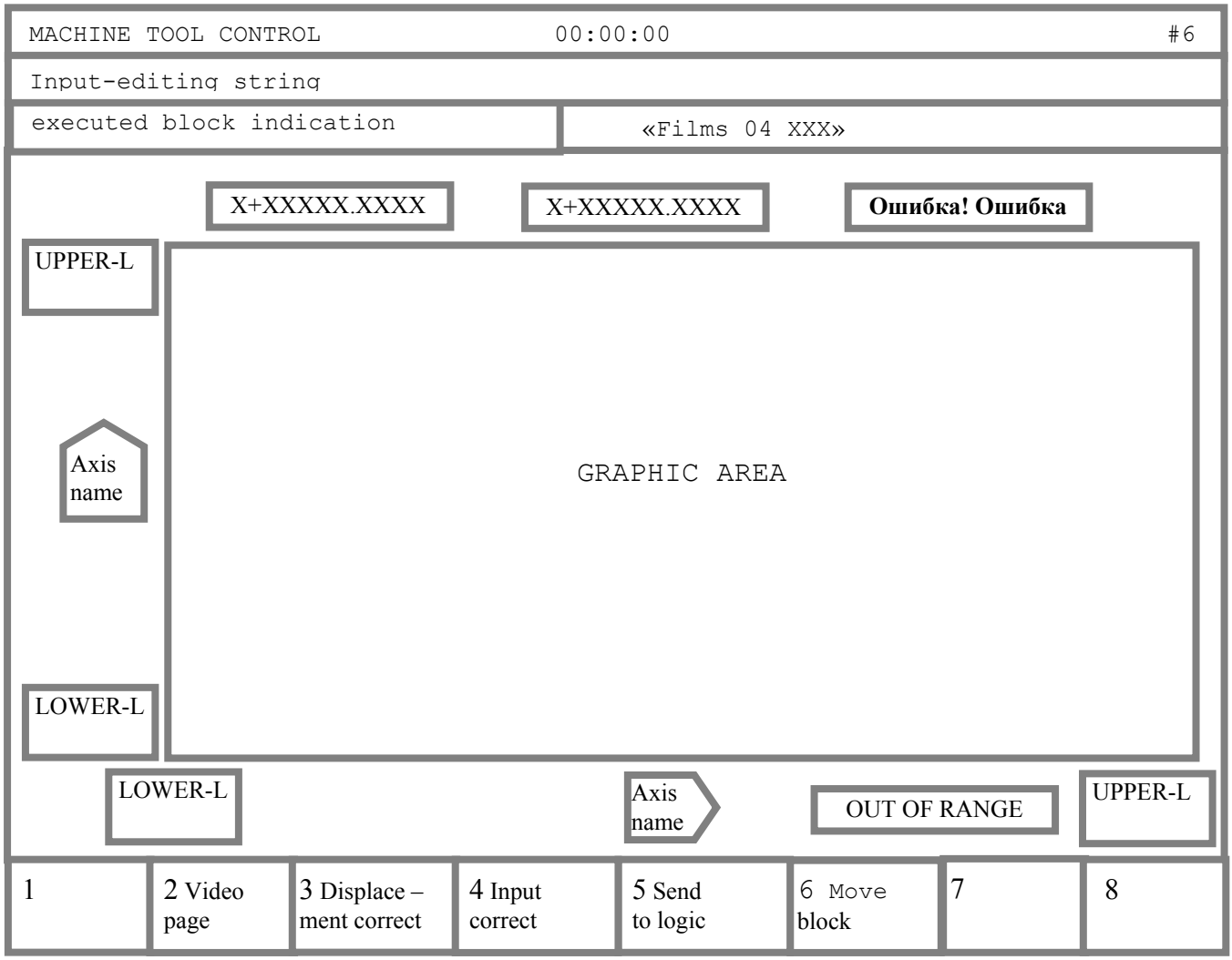


Fig. 2.7.

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 describtion 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 writted 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 writting 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 vizualization 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 OF LINE	2 END OF LINE	3 FIND UP	4 FIND DOWN	5 DEL LINE	6 GO TO LINE N	7 SAVE & EXIT	8 QUIT 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 derection 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 writting of editing file (pushbutton **F7**);
QUIT NO SAVE – exit from EDIT session without the writting of editing file (pushbutton **F8**).

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

1 MARK BLOCK	2 COPY BLOCK	3 MOVE BLOCK	4 DELETE BLOCK	5 UNMARK	6 GO TO LINE N	7 SAVE & EXIT	8 QUIT 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 writting 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 posible 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 appeas 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	Initialize magnetic cassette
	INI,NAME,/MP1	Initialize memory boards
DIS	DIS,NAME	Display a file
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	Protect a program
	ATT,NAME,0	Disable protection
DIF	DIF,NAME/MP1,/PR	Check the difference between a program in memory and on punched tape, or
	DIF,NAME/MP1,NAME/HD	between two programs resident on different devices

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) N.B.: Code UAS cannot be used by program, but only from keyboard
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 following 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..,Y..,Z..	Define absolute axes origin n: origin number To select the alternative measuring unit, enter -n
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,Y12.127 (the coordinates are stored in record 5 of the file of origins)
URP	URP,n	Define workpiece rotation angle n: rotation angle (in degrees)
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. - CODES FOR MACHINE TOOL CONTROL

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 abscissa AXIS1S: more positive abscissa AXIS2I: more negative ordinate AXIS2S: more positive ordinate 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 for editing operations on the current machining program or on a subroutine recalled from it.
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)
PTM	PTM,h:min.:[sec.]	Set the system clock

Table 4.4. continued

Code	Format	Function
MBR	MBR=0	Disable retracing
	MBR=1	Enable retracing
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. ESE,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,SQR(144)+20/2 Press [SEND] displays 22 EVA,(SQR(144)+20)/2 Press [SEND] displays 16 To calculate expressions that contain an E parameter, use LR formats (from E30 to max). E.g. EVA,E30/E31 Press [SEND]
UCA	UCA,n,Z,K	Incremental modification of offset n by value Z/K

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 VAR1 Ш VAR2
BGE	(BGE,VAR1,VAR2, LABEL)	Branch if VAR1 т VAR2
BLE	(BLE,VAR1,VAR2, LABEL)	Branch if VAR1 y VAR2
EPP	(EPP,LABEL1,LABEL2)	Execute the part of program between LABEL1 and LABEL2
RPT	(RPT,n)	Repeat a part of program n times (ny99). 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..,Y..,Z..)	Define and apply a temporary origin for the current axes n: No. of the absolute origin
UIO	(UIO,X..,Y..)	Declare the incremental origin with reference to the current absolute origin

Table 4.5. continued

Code	Format	Function
MIR	(MIR,X,Y) 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.
RQO	(RQO,n,axis..)	Medefine axis origin n: origin number
RQU	(RQU,Ntool,Noffset, Z..,K..)	Modify current Z and/or K offset and update offset file Ntool: tool number Noffset: offset number
RQP	(RQP,Ntool,Noffset, Z..,K..)	Modify current 2 and/or K offset without updating correction values
DPI	(DPI,axis1,axis2)	Define interpolation plane Axis1, Axis2: name of the axes defining the plane
DTL	(DTL,axis 1 value, axis n value)	Define a positioning tolerance for the programmed axes (differ-ent from the value declared in the characterization file)
DLO	(DLO,AXISn+ AXISn-)	Define sw overtravel limits E.g.: (DLO,X100 X-80)

Table 4.5. continued

Code	Format	Function
DIS	(DIS,variable)	Display a variable, i.e. an E parameter, an input value, a geometric element or a message for the operator.
DLY	(DLY,time)	Define a waiting time
CTL	(CTL,T) (CTL)	Select the lathe mode Re-establish the starting mode
TOF	(TOF,n)	Declare a tool out of use
UCG	(UCG,n,AXIS1I AXIS1S, AXIS2I AXIS2S,[AXIS3]	Define graphic screen n:1=non-coordinated axes n: 2=coordinated axes Example: (UCG,1,X-100 X100,Y-100 Y100,Z)
CLG	(CLG)	Reset the graphic screen
DCG	(DCG)	Disable graphic screen
DSA	(DSA,n,AXIS1I,AXIS1S) AXIS2I,AXIS2S)	Define protected zone n: zone number (1,2,3) AXIS1I:more negative abscissa AXIS1S:more positive abscissa AXIS2I:more negative ordinate AXIS2S:more positive ordinate E.g.:(DSA,2,x2OO x300,Y-60 Y100)
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
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
UAV	(UAV,1.xc,pv,r)	Enable virtual axes p and v r: minimum radius
	(UAV,2,b,w,r)	Enable virtual axis w r: radius of the cylinder
	(UAV,0)	Disable virtual mode
	(UAV,3,slaves,master slave/master offset/mirror)	Enable dual axes: slave/master offset: establishes the slave of each master mirror: enable/disable slave mirror

Table 4.5. continued

Code	Format	Function
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 field: field number value: value to be displayed A negative number displays the field in reverse mode

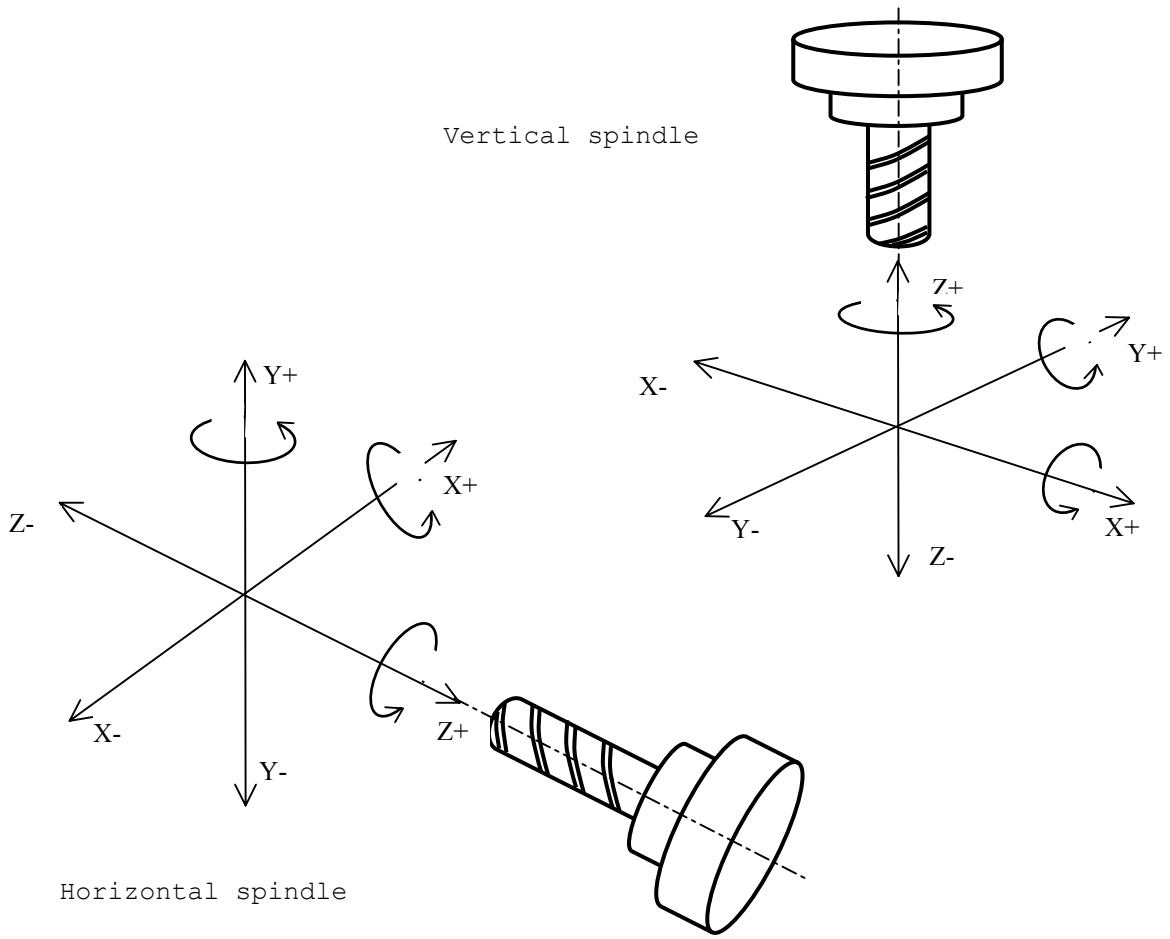
5. PROGRAMMING

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, independent of which --axes or workpiece-- is actually moving, as shown in figure 5.1.

Fig. 5.1. - Vertical and Horizontal Spindle



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 drawinq 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 proqrams 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 identifyies the type of instruction. For example: G, X, Y, 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;
- #: synchronization marker.

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
"INIZIO" N125
/"FINE" N125

The block synchronization field is marked by the "#" character.
If this field is present, the system starts analyzing the block only after all the other current operations have been been terminated, i.e. after all the preceding blocks have been read, calculated and executed.

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	% (IS0) er (EIA)
End of block	L.F. (LINE FEED) (ISO) 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 this 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 Y20 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 Y25
• 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,"TWIST DRILL D6")
N2   T1.1  M6 S800
N3   G X80 Y80 M13
............
............
N236 G X250 Y50 M5
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: A B C U V W X Y Z P Q D
- R
- I J
- K
- S
- T
- F
- G
- M
- H

A B C U W X Y Z P Q D - Format: ;±0.0001 to ±99999.9999 mm or in.
To characterize 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.
In circular interpolation. R defines the radius of the circumference.

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
K defines the tool diameter offset.
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.
In a helicoid interpolation, K defines the helix pitch.

S - Format: 0.01 to 99999.99
S defines either spindle speed in rpm (with G97) or --if characterized-- cutting speed in m/min (with G96)

T - Format: 1.0 to 9999.9999
T defines the desired tool (integer portion) and the corresponding tool offset (decimal portion).
Tool ooffset is enabled by M06. Both the tool length and diameter (K) can be offset.
Tool length offset can be defined for any axis. To specify your selection, enter the axis word. For example:
Z55, X20, Y60

Offset does not require any other preparatory function. The control automatically adds the offset value to the programmed coordinate.
Tool diameter offset is simultaneous with length offset and is enabled by preparatory functions G41/G42. RESET and POWER-OFF do not cancel active offsets.

If you omit the tool offset number, the current offset will be deactivated.

Important. If the tool is multiple (i.e. has several cutters), it is possible to define a T function with an offset for each cutter. In this case, a PROGRAMMED TOOL-TOOL POCKET correspondence file must be created.

F - Format: 0.01 to 99999.99
With G93, F defines the inverse mf 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/min (with G71) or in/min (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 m/rev (with G71) or in/rev (with G70).

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

M - M functions define on/off operations of the machine tool components. M codes (M0 to M99) are normally assigned by the machine tool manufacturer. The most frequent M functions are listed in Table 5.3. For further details, refer to the documentation provided by the manufacturer of your machine.

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
G17	b	no	Circular interpolation and depth compensation on XY plane (1st-2nd axes)	yes
G18	b	no	Circular interpolation and depth compensation on XZ plane (3rd-1st axes)	no
G19	b	no	Circular interpolation and depth compensation on YZ plane (2nd-3rd axes)	no
G27	c	no	Continuous path mode with automatic deceleration on corners	yes
G28	c	no	Continuous path mode without 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

Table 5.2. continued

Code	Modal Group	Only one block?	Description	At power-up
G41	e	no	Tool tip rad.compensation (tool left of part)	no
G42	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
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
G86	g	no	Fixed boring cycle	no
G89	g	no	Fixed boring cycle with dwell	no
G90	h	no	Absolute programming	yes
G91	h	no	Incremental programming	no
G79	i	yes	Programming referred to machine zero	no
G04	j	yes	Dwell at end of block	no

Table 5.2. continued

Code	Modal Group	Only one block?	Description	At power-up
G09	j	yes	beceleration at block end	no
G72	k	yes	Measure of a point with radius compensation	no
G73	k	yes	Measure of hole parameters	no
G74	k	yes	Measure of variance from nominal coordinates without radius compensation	no
G93	l	no	Feedrate as inverse of time	no
G94	l	no	Feedrate mm/min or inch/min	yes
G95	l	no	Feedrate mm/rev or inch/rev	no
G96	m	no	Cutting speed m/min or foot/min	no
G97	m	no	Spindle speed in rpm	yes

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 US0=0, M01 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
M10	x		M11	Lock axes not involved
M11	x			Unlock axes (cancel M10)
M12	x		M11	Lock rotary axes
M13	x		M4-M5-M14-M19	Spindle rotation CW and coolant ON

Table 5.3. continued

Code	Active at Motion start	Active at Motion end	Can be cancelled by	Meaning
M14	x		M3-M5-M13-M19	Spindle rotation CCW and coolant ON
M19	x		M3-M4-M5-M13-M14	Spindle stop and angle orient
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
M60		x		Part 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 available 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.

A block can include several G functions, provided they are mutually compatible. Compatibility between the different groups is dealt with in table 5.5.

Table 5.4. - MODAL GROUPS

GROUP	G FUNCTIONS	DESCRIPTION
a	G00-G01-G02-G03-G33	Definition of the movement
b	G17-G18-G19	Definition interpolation plane
c	G27-G28-G29	Definition dynamic mode (point-to-point or continuous)
d	G21-G20	Open and close the GTL programming ambient
e	G40-G41-G42	Tool radius offset enable and disable
f	G70-G71	Programming in alternative unit
g	G81..G86-G89-G80	Standard fixed cycles
h	G90-G91	Incremental/absolute programm.
i	G79	Programming referred to machine zero
j	G04-G09	Attributes of the dynamic mode
k	G72-G73-G74	Measuring cycles
l	G93-G94-G95	V/D feedrate coding
m	G96-G97	Spindle speed

Table 5.5. - COMPATIBILITY BETWEEN G FUNCTIONS

1=non compatibility

G	00	01	02 03	33	81 86 89	80	72 73 74	21	20	41 42	40	27 28	29	04	09	90 91	79	70 71	17 18 19
G00	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1
G01	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1
G02	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
G03	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
G04	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	1
G09	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	1
G17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G20	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	1
G21	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	1
G27	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	1
G28	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	0	0	0	1
G29	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1
G33	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1
G40	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	1
G41	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1
G42	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1
G70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
G71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
G72	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G73	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G74	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G79	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	1	1	0	1
G80	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	1
G81	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G82	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G83	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G84	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G85	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G86	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G89	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
G91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
G93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
G94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
G97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

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

The allowable functions are:

G00 rapid axes positioning
G01 linear interpolation
G02 circular interpolation CW
G03 circular interpolation CCW
G33 constant or variable pitch threading

```
.......................................................
: 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] [OFFSET OPERANDS] [FEED] [AUXILIARY FUNCTIONS]

where:

[OTHER G CODES]	all the compatible G functions (refer to table 5.6.).
[AXES]	the axes words followed by a numerical value, explicit or implicit (E parameter). Max. 3 axes. The specified axes are not interchangeable. To implicitly define the axes, you must first define a point according to the current abscissa and ordinate.
[OFFSET OPERANDS]	offset factors on profile (u,v,w)

[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 G00 is the vectorial composition of the rapids of the programmed axes. It is stored in the characterization file.

[AUXILIARY FUNCTIONS] 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 fedrate.

The allowable format is:

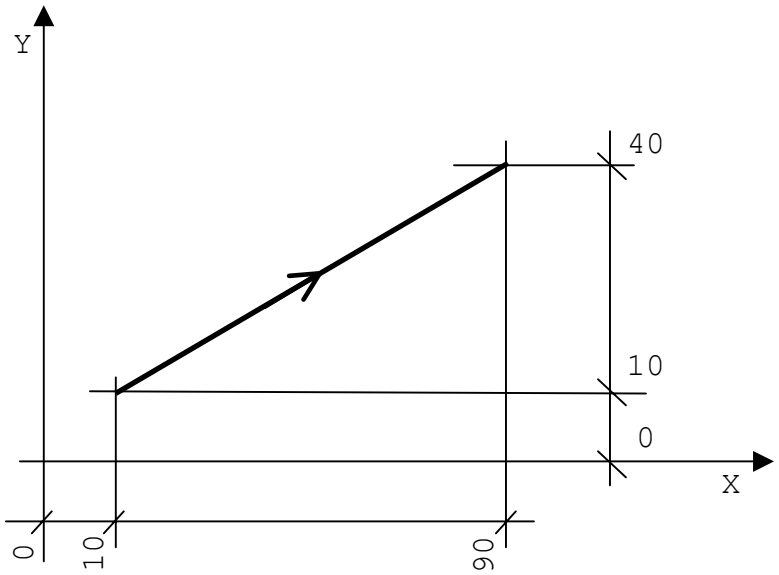
G01 [OTHER G CODES] [AXES] [OFFSET OPERANDS] [FEED] [AUXILIARY FUNCTIONS]

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 G00.

Fig. 5.2. - Linear interpolation




```
N79     X10 Y10
N80 G01 X90 Y40 F200
```

G02 G03 - CIRCULAR INTERPOLATION

Use these codes to program circular movements, respectively CW (G02) or CCW (G03).
The programmed axes move along the circular path at programmed feedrate.

The allowable formats are:

1) for circular interpolation with center coordinates (I and J)
{G02}[OTHER G CODES][AXES] I J [F][OFF.OP.][AUX.FUNCT.]
{G03}

2) for circular interpolation with the circle radius
{G02}[OTHER G CODES][AXES] c R [F][OFF.OP.][AUX.FUNCT.]
{G03}

where:

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

- [FEED] has been explained in G01.

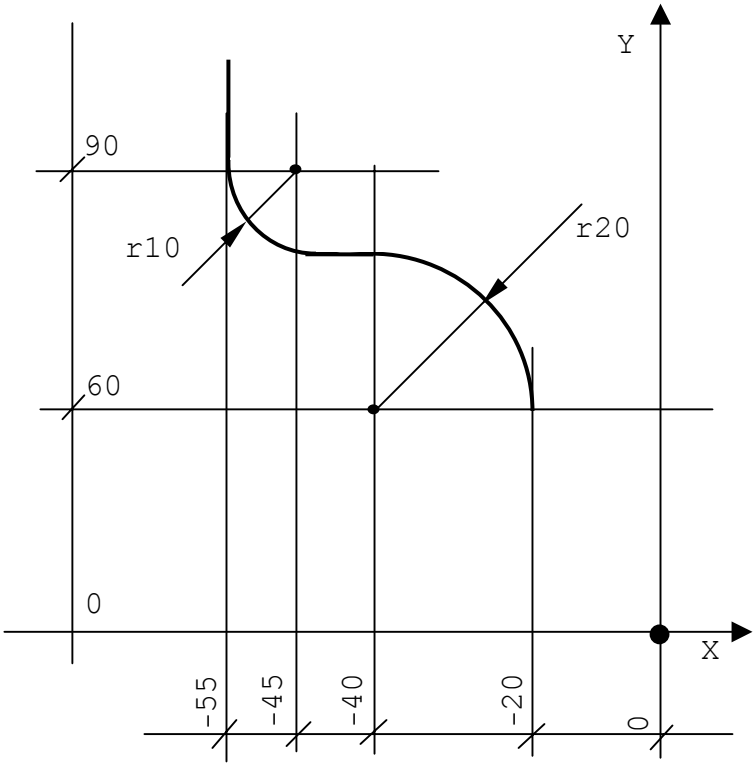
- the axes are represented by the corresponding word followed by a numerical value, either explicit or implicit (E parameter).
  If no axis is specified. The code generates a complete circular movement in the interpolation plane.
  You can define axes implicitly by specifying a geometric item (i.e. point).

- 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.

Circular interpolation with center coordinates (I and J)

```
N10 G1 X-20 Y60 F200
N20 G3 X-40 Y80 I-40 J60
N30 G1 X-45
N40 G2 X-55 Y90 I-45J90
N50 G1 Y...
```



Example:

Circular interpolation with the R circle radius

```
N10 G1 X-20 Y60 F200
N20 G3 G91 X-20 Y20 I-20 J0
N30 G1 X-5
N40 G2 X-10 Y10 I0J10
N50 G1 Y...
```

Notes:

- The maximum programmable arc is 360 degrees.

- Full circles must be programmed with I and J parameters.

- 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.

G02 G03 HELICOIDAL INTERPOLATION

Use these codes to program, in a single block, a helicoidal path to be carried out with simultaneous circular movement of the axes defining the interpolation plane and linear movement of the perpendicular axis.

To program a helicoidal path, add the depth value and the helix pitch (K) to the block describing a circular interpolation.

The allowable format is:

for helicoidal interpolation with center coordinates (I and J)
{G02}[OTHER G CODES][AXES] I J K [F][OFF.OP.][AUX.FUNCT.]
{G03}

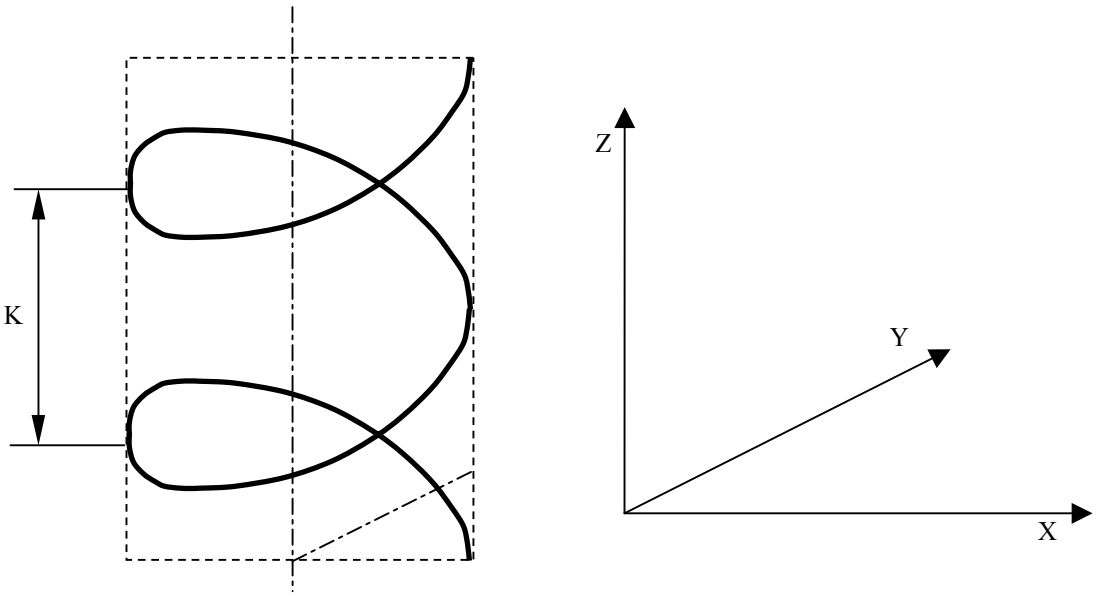
Example:

G2 (or G3) X.. Y.. Z.. I.. J.. K.. F..

where:

G2 X..Y..I..J..	circle coordinates
Z	depth dimension
K	helix pitch (K can be omitted if the helix depth is less than 1 pitch)

Fig. 5.3. - Helicoidal interpolation



Notes:

- The path length is equal to n K.
  If n is an integer number, program a complete circle (360 degrees).
  If n is a decimal number, program an arc proportional to n. For example, if the path ength is 2.7 K
  program the following arc:
  360 x 0.7 = 252 degrees

- As in the case of circular interpolation, you can specify the plane of helicoidal interpolation with G17-G18-G19 or with three-letter code DPI.

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:

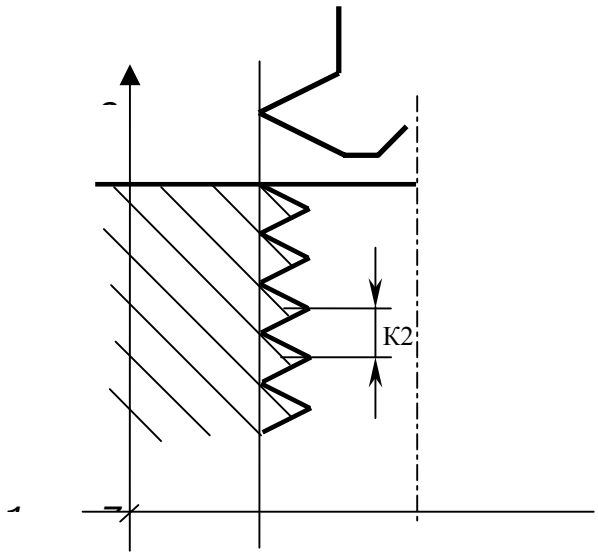
- During threading, HOLD, FEEDRATE OVERRIDE and SPINDLE SPEED OVERRIDE are disabled.

- You can only program G33 if your spindle features a transducer.

Here are some examples of constant and variable pitch threading:

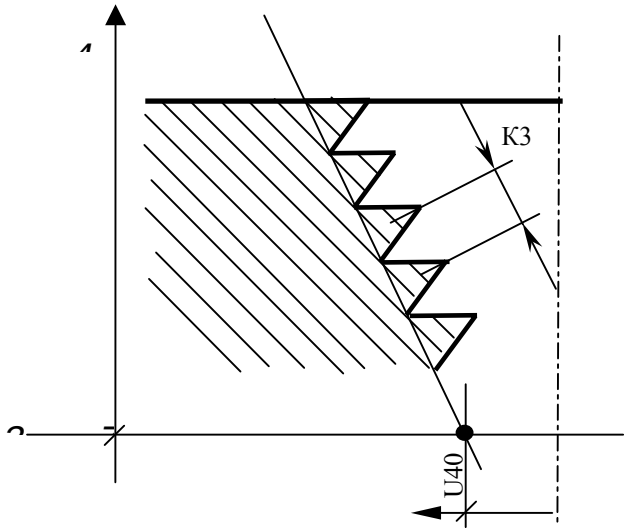
Fig. 5.4. - Constant pitch threading

a) Cylindrical thread



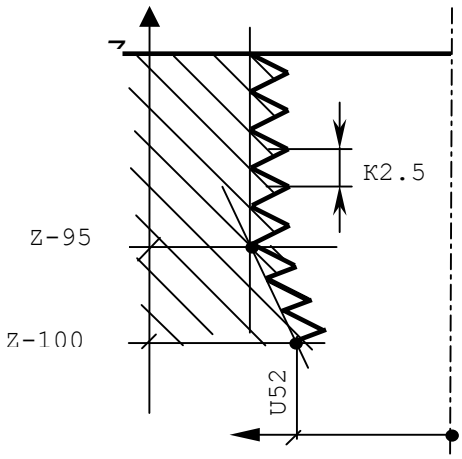
N15 G33 Z-100 K2

b) Conical thread



N22 G33 Z-80 U40 K3

c) Cylindrical-conical thread

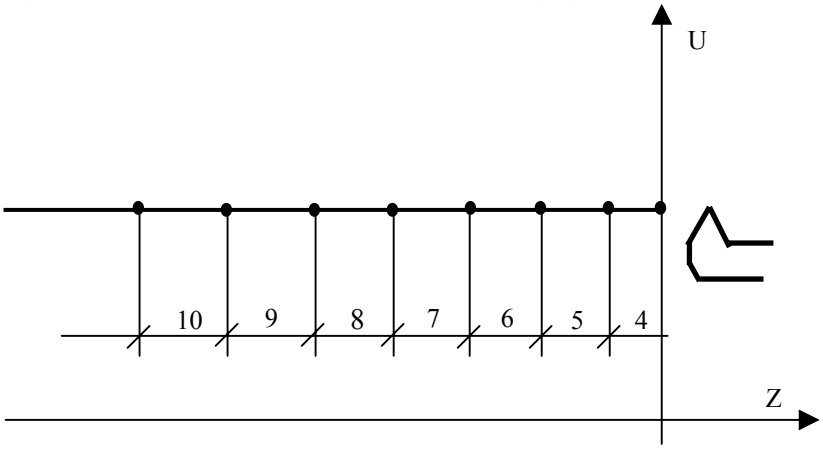


N35 G33 Z-95 K2.5
N36 Z-100 U52 K2.5

Notes:
- U is a diametric axis.
- All the parameters can be expressed with either an explicit or an implicit numeric value.

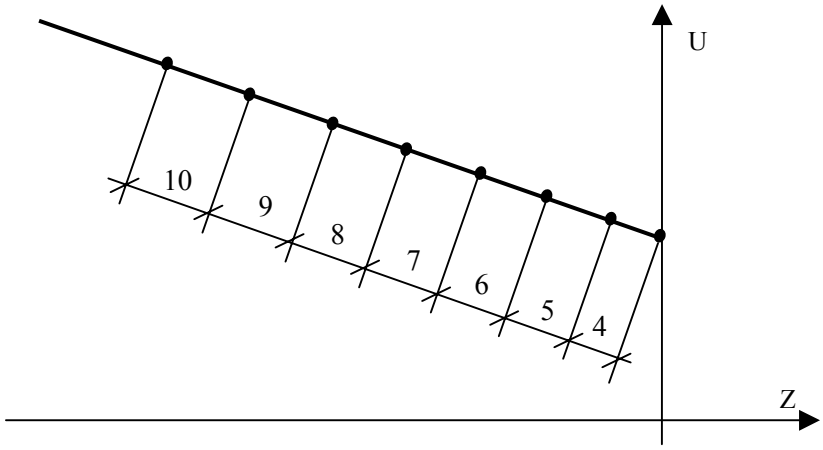
Fig. 5.5. - Variable pitch thread:

a) Cylindrical thread with increasing pitch



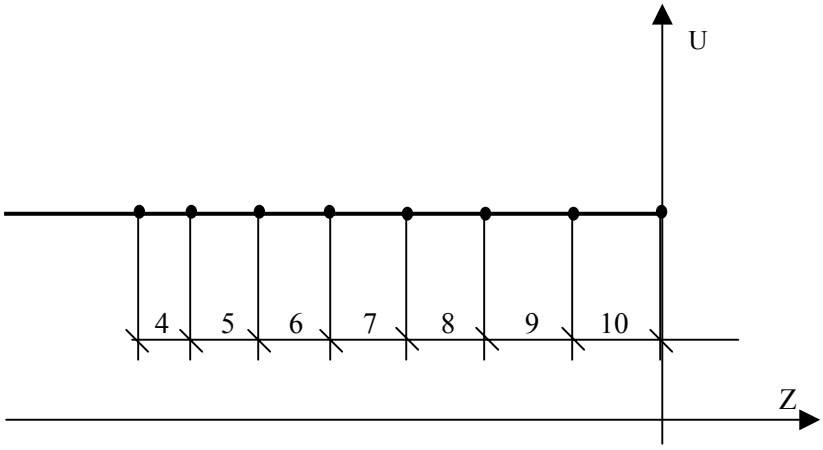
N24 G33 Z-50 K4 I1

b) Conical thread with increasing pitch



N24 G33 U50 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)

<u>Example of triple-start thread</u>

```
..........
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: G17 G18 G19 - DEFINITION OF THE INTERPOLATION PLANE

To specify the interpolation plane, use:

G17 if the interpolation plane is defined by axes 1 and 2 (X and Y)

G18 if the interpolation plane is defined by axes 3 and 1 (Z and X)

G19 if the interpolation plane is defined by axes 2 and 3 (Y and Z).

Axes 1-2-3 are the first three axes declared in the characterization file (by default, X-Y-Z respectively).

The allowwable format is:

{G17}
{G18}
{G19}

Note:

- These functions must be declared by themselves. Do not enter any other parameter in the same block.

5.8.3. MODAL GROUP C: DEFINITION OF THE DYNAMIC MODE

The functions belonging to this class define the execution mode.

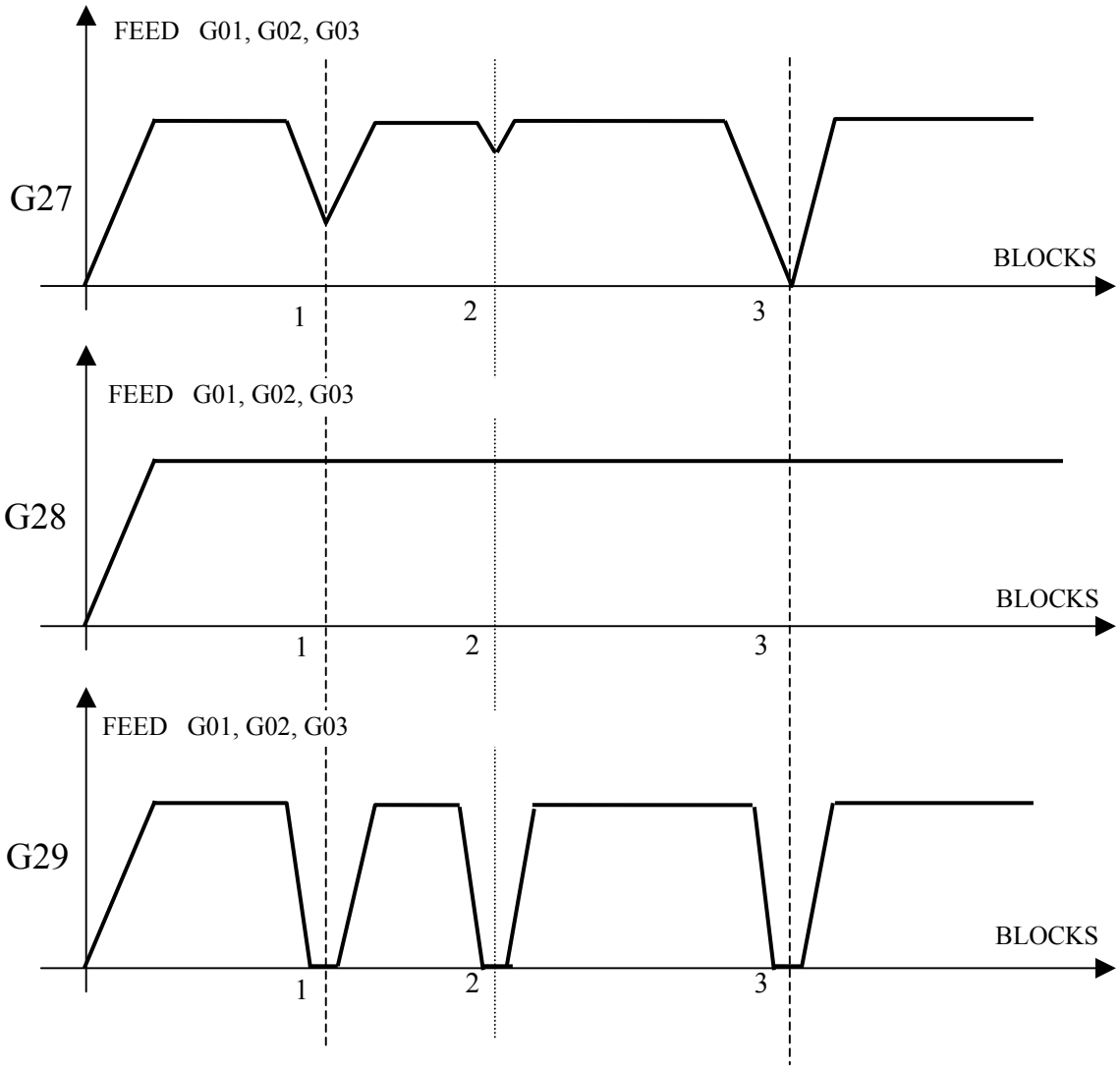
The allowable formats are:

{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. 5.6. - Graphic representation of the dynamic mode



- 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).

- 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 le block. Therefore, within a G27-G28 path, you cannot program M, H, S nor T functions.
  Since tne continuous mode is temporarily closed by a G00 stil belonging to the profile, program the desired auxiliary functions after G00.

To execute the profile shown in, program the following sequences:

a) Continuous mode

```
...........
N9  (DIS,"MILL DIA.  16")
N10 S800 T4.4 M6
N11 G X-235 Y-230 M13
N12 Z-10
N13 G1  X75 F500              -Continuous mode starts  (G27)
N14 Y
N15 G3 X-70.477 Y25.651  I J
N16 G1  X-187 Y-295
N17 G Z5 M5                   -Continuous mode temporarily
N18  (DIS,"MILL DIA. 28")      closed (G), spindle stop, tool
N19 T5.5 M6 S1200              change, S functions
N20 G X..  Y.. M13
N21 Z-..
N22 G1  X..  Y..              -Continuous mode restarts
...........
```

Important. If function GZ9 were programmed in block N17, the continuous mode would be definitely closed and the subsequent movements specified with G1, G2, G3 would be carried out in point-to-point mode.

b) Point-to-point mode

```
...........
N9  (DIS,"MILL DIA.  16")
N10 S800 T4.4 M6
N11 G29 G X-235 Y-230 M13     -Point-to-point starts
N12 Z-10
N13 G1  X75 F500 M5           -Spindle stop
N14 Y S1200 M13               -Change S, spindle speed
N15 G3 X-70.477 Y25.651  I J   coolant
N16 TMR=2
N17 G1  G4 X-187 Y-295         -Dwell at the end of the item
N18 G Z5 M5
N19 (DIS,"MILL DIA. 28")
N20 S1200 T5.5 M6 S1200
N21  G X..  Y.. Ml13
N22 Z-..
N23 G1  X..  Y..
...........
```

Important. By programming point-to-point mode in N11 (with G29), you can enter special functions S and M within the profile (blocks N13 and N14.
You can also program a dwell at the end of the item in continuous mode (with G4).

5.8.4. MODAL GROUP C: 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 pn to obtain an open profile; never program pn 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.

For further examples and details, refer to section 5.15.

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

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

G41 enable compensation, tool left of part
G42 enable compensation, toll riqht of part
G40 disable compensation

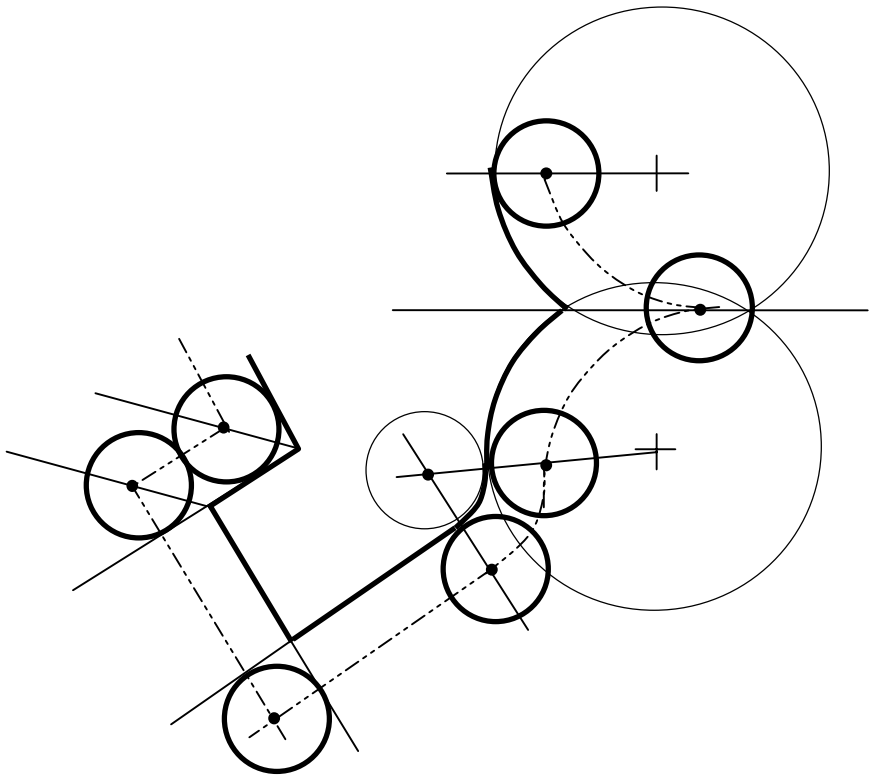
The allowable format is:

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

Before programming a tool radius compensation, you must specify the interpolation plane. As we have already seen, the interpolation plane can be defined with G17 (at power-up), G18, G19 or with the DPI code.

The tool radius compensation, which is perpendicular to the tool path, enables the control to position the tool at the exact intersection point.

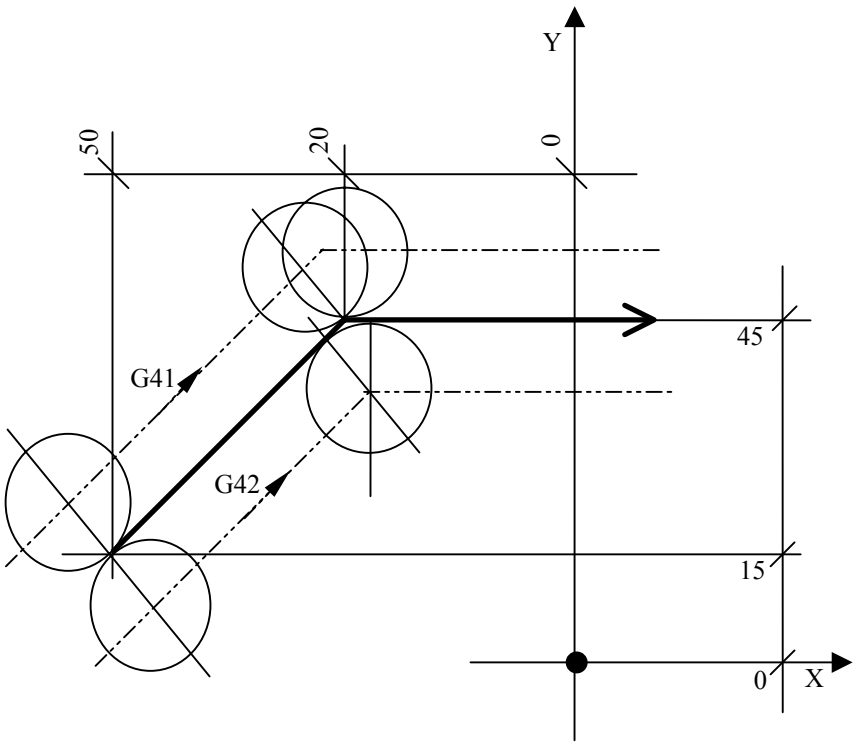
Fig. 5.7. - Tool radius compensation



To activate tool radius compensation, program G41 or G42. The tool must reach the first point with a linear interpolation (G00-G01).
The control compensates all tool motion programmed in the block that contains G41 or G42 up to but not including the block that contains G40.

Here are two examples of programming for tool radius compensation:

Fig. 5.8. - 1st. linear item



Route covered by tool bit

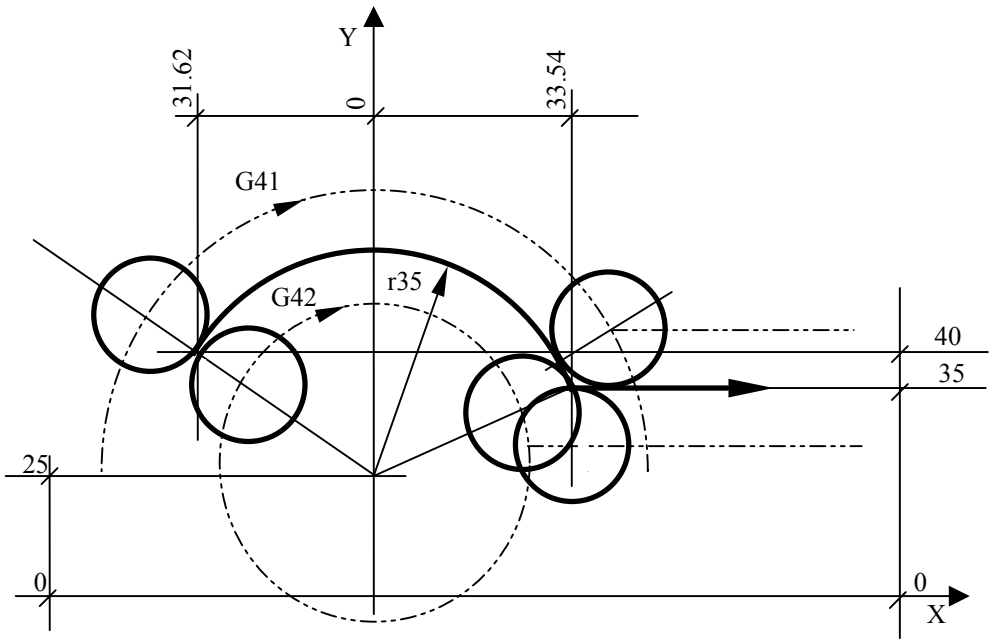
Tool right of part:

```
G1 G42 X-50 Y15 F200
X-20 Y45
```

Tool left of part:

```
G1 G41  X-50 Y15 F200
X-20 Y45
```

Fig. 5.9. - 1st circular item



Tool right of part:

```
G1  G42 X-31.622 Y40 F200
G2  X33.541 Y35 I J25
```

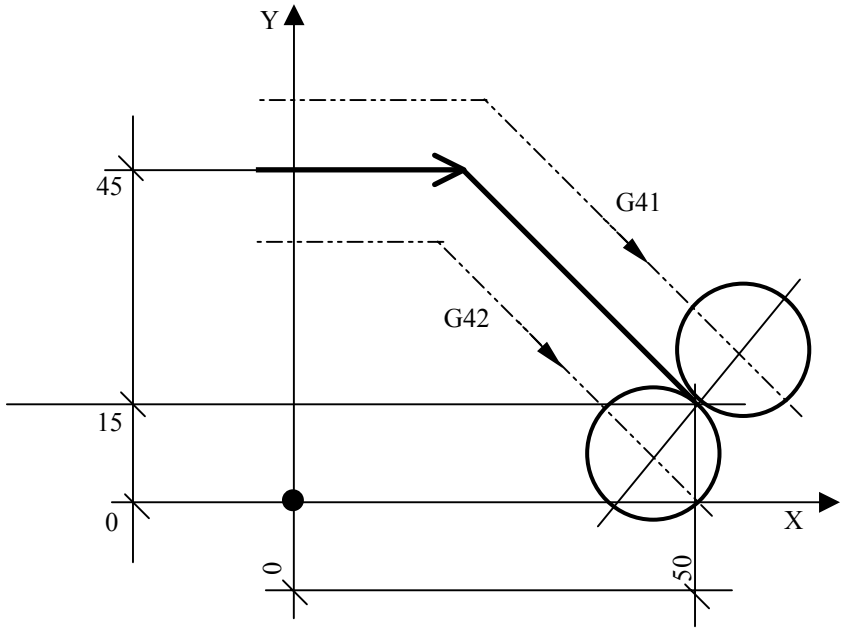
Tool left of part:

```
G1  G41 X-31.622 Y40 F200
G2  X33.541 Y35 I J25
```

To disable tool radius compensation, program G40. The feature will be disabled from the first block following G40.

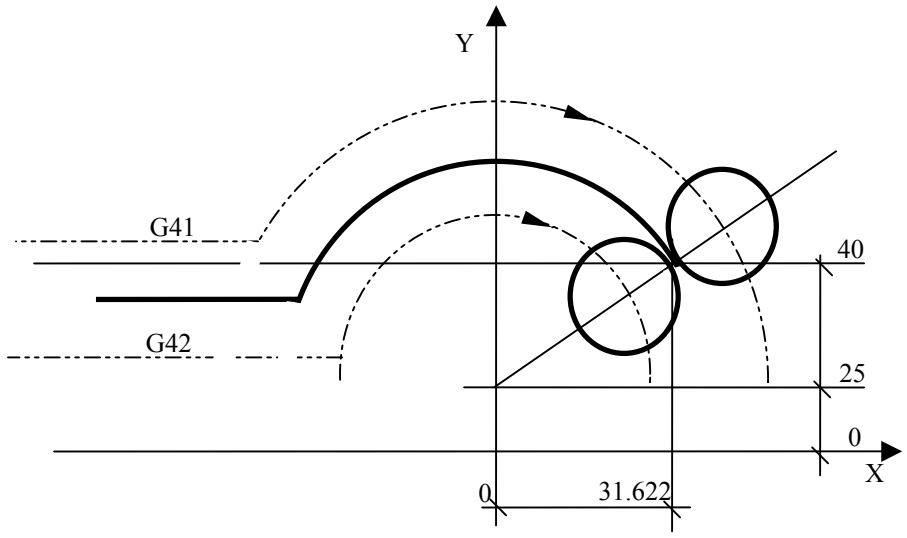
Here are some examples of G40:

Fig. 5.10. - Last linear item




```
...........
N88  G1 G40 X50 Y15
N100 G X..  Y..
..........
```

Fig. 5.11. - Last circular item

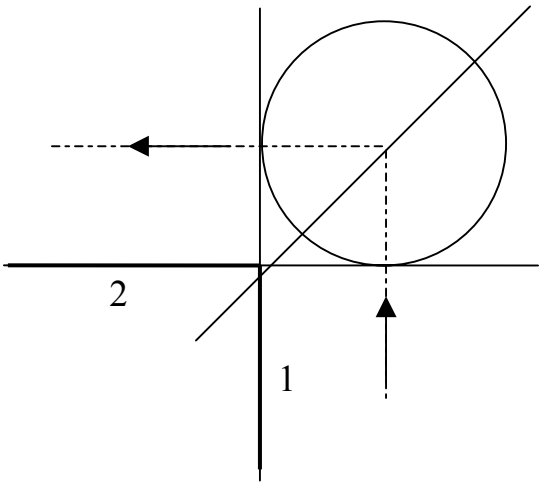


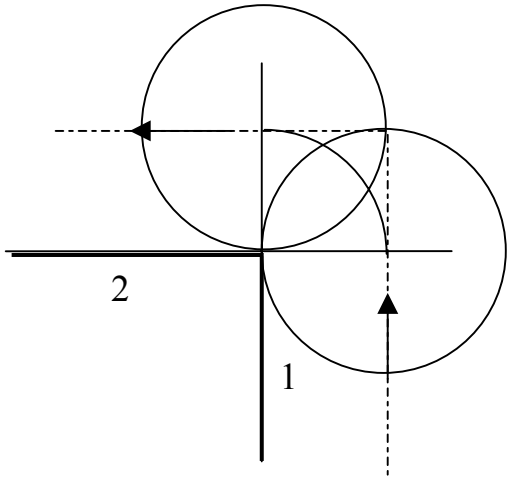

```
..........
N99  G2 G40 X31.62 Y40 I J25
N100 G X..  Y..
..........
```

Notes:

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

  Examples:



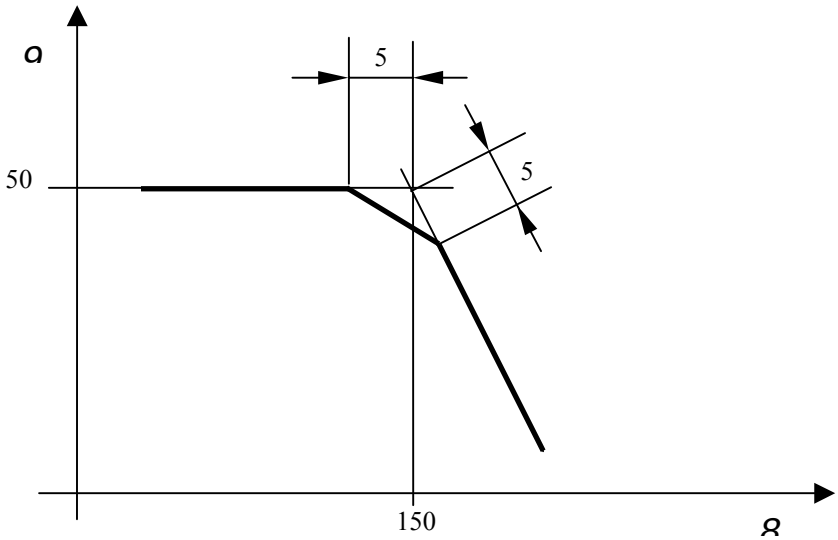



```
Without radius
...........
1)  N20 G1 X100 Y100
2)  N21 X-100
    ..........
```

```
With radius
 ...........
 1)  N20 G1  X100 Y100
     N21 r-1
 2)  N22 X-100
```

A radius r=0 optimizes the tool path by generating a radius equal to zero mn 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:




```
1) N10 G1 X50 Y
   N11 X150 Y50
   N12 b5
   N13 X100 Y50
```

- When using tool radius compensation, you cannot program any of the following furctions:
  . G21-G20
  . G related to fixed cycles
  . G79
  . G17-G18-G19
  . G33
  . G related to probing cycles.

- Tool compensation affects all programmed motions, either at rapid or machining speed.

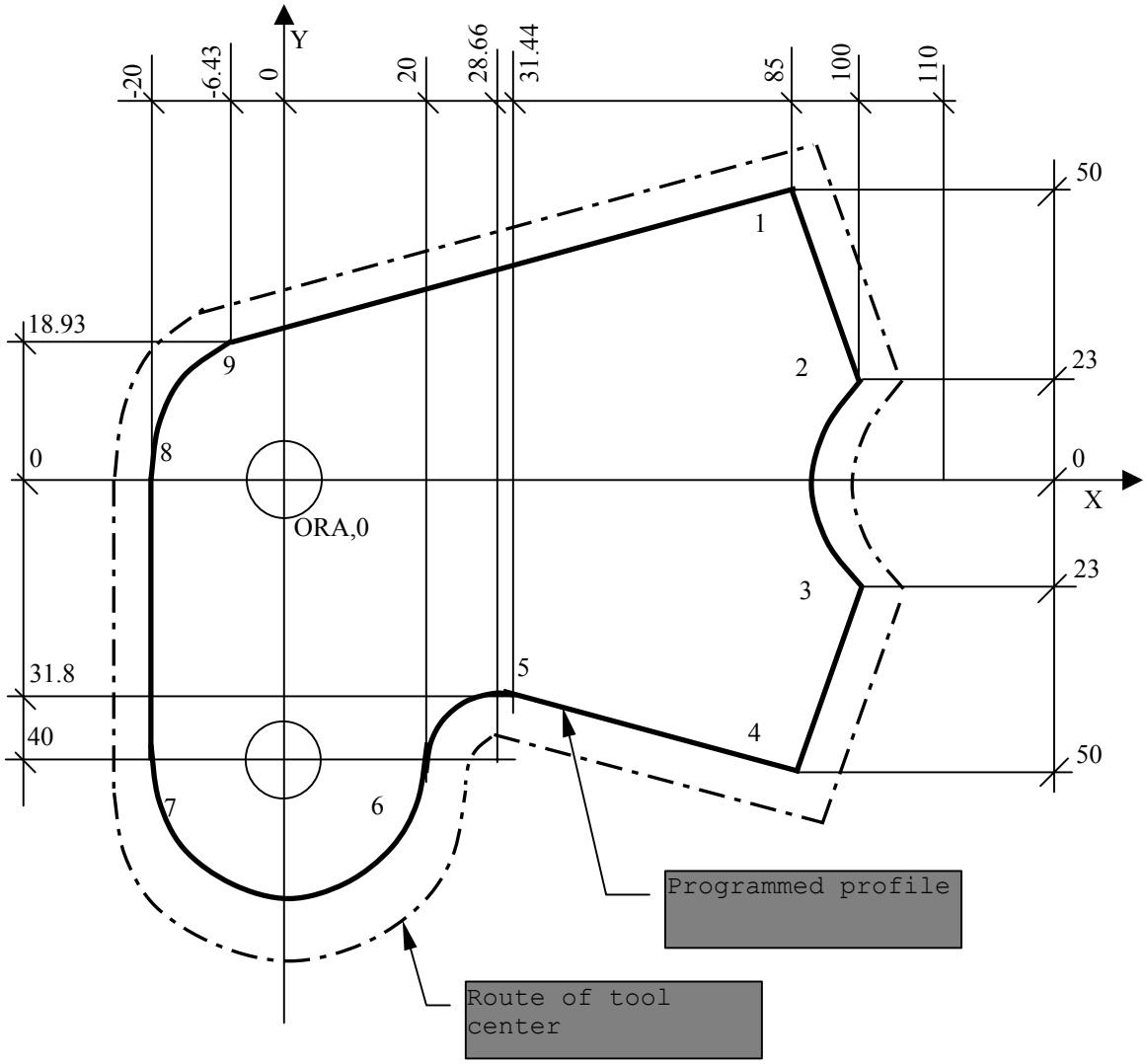
- When tool compensation is enabled, both a change of plane and a radius less than the tool r generate an error message.

- Within a contouring operation with G41-G42, you can program the motion of axes not belonging to the profile (i.e., axes perpendicular to the interpolation plane). Such a motion is not allowed after an automatic radius.

- In GTL ambient, you can program in a single block:
  . G21 and G41 or G42 to enamle tool compensation;
  . G20 and G40 to disable tool compensation.

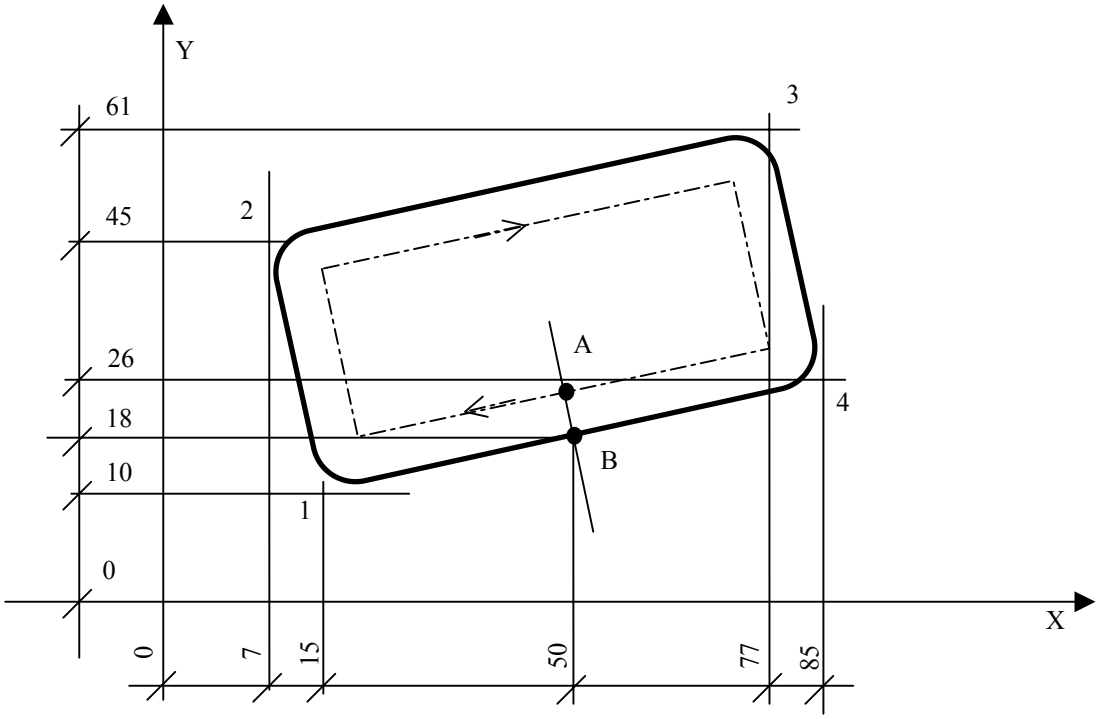
- When programming compensation in a concave path, an error message will be displayed if the compensated profile does not coincide with the programmed profile.

Fig. 5.12. - External path with radius compensation




```
   N1  S1500 T1.1  M6
   N2  G X85 Y60 M3
   N3  Z-12
1  N4  G1  G41  X85 Y50 F220
2  N5  X100 Y23
   N6  r0
3  N7  G3 Y-23 I110 J  F180
   N8  r-2
4  N9  G1  X85 Y-50
5  N10 X31.441  Y-31.803
6  N11 G3 X20 Y-40 I28.657 J-40 F100
7  N12 G2 X-20 I J-40 F250
8  N13 G1  Y F220
9  N14 GZ X-6.433 Y18.937 I J  F250
1  N15 G1  G40 X85 Y50 F220
   N16 G X85 Y60
   N17 Z2
```

Fig. 5.13. - How to program a slot




```
   N84  (DIS,"END MILL MD D=12")
   N85  S1100 F170 T9.9 M6
A  N86  G X50 Y32 M3
   N87  Z-305
8  N88  G1  G42 X50 Y18 F170
1  N89  X15 Y10
2  N90  X7 Y45
3  N91  X77 Y61
4  N92  X85 Y26
B  N93  G40 X50 Y18
A  N94  G X50 Y32
   N95  Z M5
```

Notice that, to prevent the mill cutter from damaging the piece, we had to introduce point B. Otherwise, machining would have started directly from segment 1-2.

5.8.6. 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]

Notes:

- If you do not specify any unit of measure, the control automatically defaults the characterized unit.

- Both the tool requalification value and the probed coordinate are expressed in the characterized unit.

5.8.7. 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] R1 [R2] SPINDLE AXIS [MISCELLANEOUS OPERANDS] [FEED] [AUXILIARY FUNCTIONS]

where:

[OTHER G CODES]	other preparatory functions compatible with the fixed cycle.
R1 [R2]	these spindle coordinates define, respectively, the start machining point (reached by the tool at rapid) and the final point. They can be specified in explicit or implicit mode (E parameter). If R2 is not specified, the control assumes R1 as final coordinate.
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.

Table 5.6. - FIXED CYCLES

Code Function	Approach	Bottom end		Return
		Dwell	Spindle rotation	
G81 drilling	feedrate	no	normal	rapid to R1 or R2*
G82 spot-facing	feedrate	yes	normal	rapid to R1 or R2*
G83 deep drill. (with chip discharge)	intermittent feedrate: down at feedr. retracts rapid	no	normal	rapid
G84 tapping	feedrate - rotation starts	no	rotation reversal	feedr. to R1 rapid to R2*
G85 reaming or tapping by Tapmatic	feedrate	no	normal	feedr. to R1 - rapid to R2*
G86 boring	feedrate - spindle rot. starts	no	stop	rapid to R1 or R2*
G89 boring with spot-facing	feedrate	yes	normal	feedr. to R1 - rapid to R2*
G80 cancels fixed cycles				

* if R2 has been specified.

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

Notes:

- The block programming the fixed cycle must not include any cther 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.

- To program subsequent cycles identical to the first one, simply program the coordinates of the points on which the new fixed cycles must be executed.
  To define these points, either enter their cartesian coordinates (i.e. X and Y) or program the definition (i.e. with the P code).

- Program dwells with TMR.

- The fixed cycle is attributed to the axis specified in the declaration block. For example,

  G81 R Y-20 the fixed cycle is attributed to Y.

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

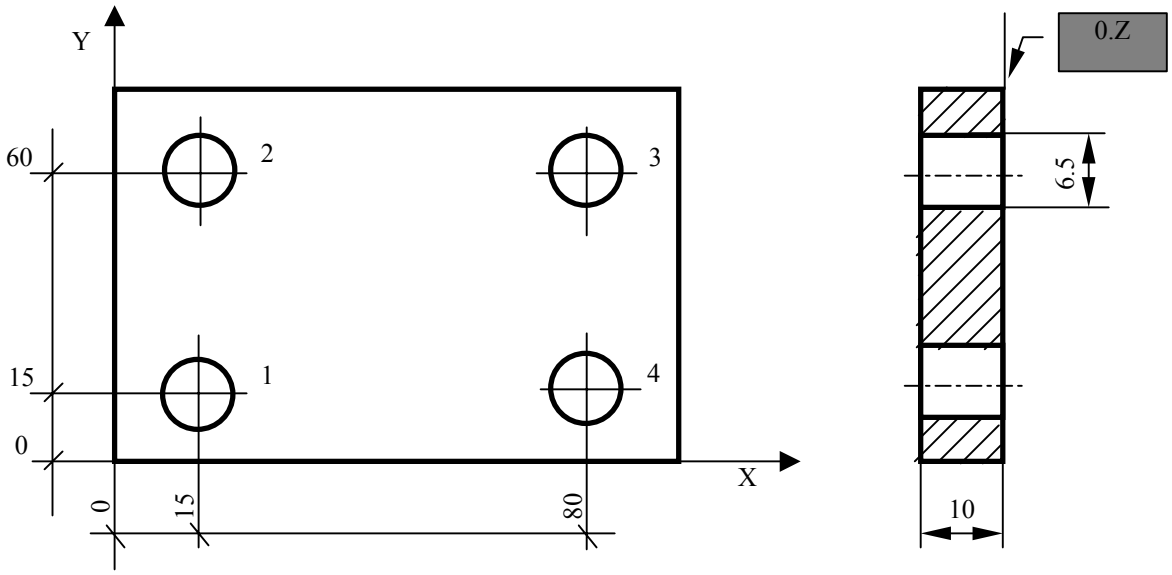
- To only update the preceding values without generating any motion, program the R coordinate by itself.

G81 Drilling cycle

The format of the programming block is:

G81 R1..[R2] Z..

Fig. 5.14. - Drilling cycle




```
   N31 (DIS,"TWIST DRILL D=6.5)
   N32 S1100 F95 T3.3 M6
   N33 G81 R 3  Z-15 M3
1  N34 X15 Y15
2  N35 Y60
3  N36 X80
4  N37 Y15
   N38 G80 Z50 M5
```

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 TMR=2
N34 G82 R-3  Z-5   T6 M13
N35 X35 Y150
```

G83 Deep drilling

The programminq block hasthe following format:

G83 R1.. [R2..] Z..I..J..K..

where:

R1 hole start point (as for G81)
[R2..] return point (as for G81)
Z bottom end (as for G81)
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 R1
  c. macnining Teemrate to R1+I
  d. rapid approach to R1 (chip discharge)
  e. calculation OT tne new R1 value
     R1 = R1 + I - 1
  f. calculation of the new I value
     I = I * K if I * Kт J
     I = J if I * K < J

  Steps b, c, d, e, f, follow one another until depth reaches the programmed value (step b is always carried out at the initial R1; if present, R2 only serves for the last discharge).

  To keep I invariable, that is, to get a constant increment, program K=1 and omit J.

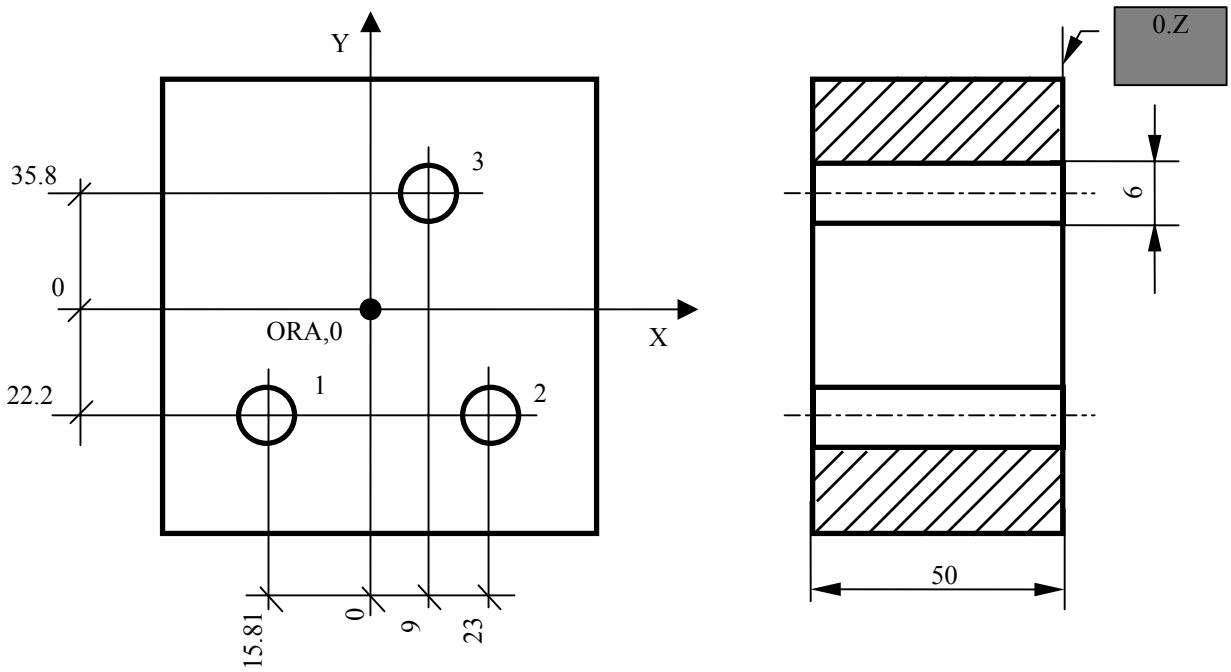
- if K and J have not been specified (woodpecking without discharge), the cycle proceeds as follows:

  a. rapid approach to the hole axis
  b. rapid approach to R1
  c. machining feedrate to R1=R1+I
  d. dwell programmed with TMR
  e. approach hy another I

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

  f. rapid to R1 or R2 (if programmed).

Fig. 5.15.




```
    N65 (DIS,"TWIST DRILL D=6")
    N66 S930 F65 T6.6 M6
    N67 G83 M3 Z-55 I20 K.8 J6 M13
1   N68 X-15.81 Y-22.2
2   N69 X23
3   N70 X9 Y35.8
    N71 G80 Z50 M5
```

G84 Tapping cycle

This cycle offers you two variants:

- if the spindle does not feature a transducer:

  the programming block is:

  G84 [R1.. [R2..]]Z..

  where:

  G84 code for fixed tapping cycle
  [R1] rapid approach and feedrate return dimension
  [R2] rapid return dimension
  Z end of tapping dimension

  When programming this block, bear in mind that:

  . Rapid approach (R1) to part should always reach a distance equal to 5 tap pitches (if depth ≤ 3 diameters or 7 tapX012pitches (if depth > 3 diameters).

  . To calculate feedrate (F) use the following formula follows:

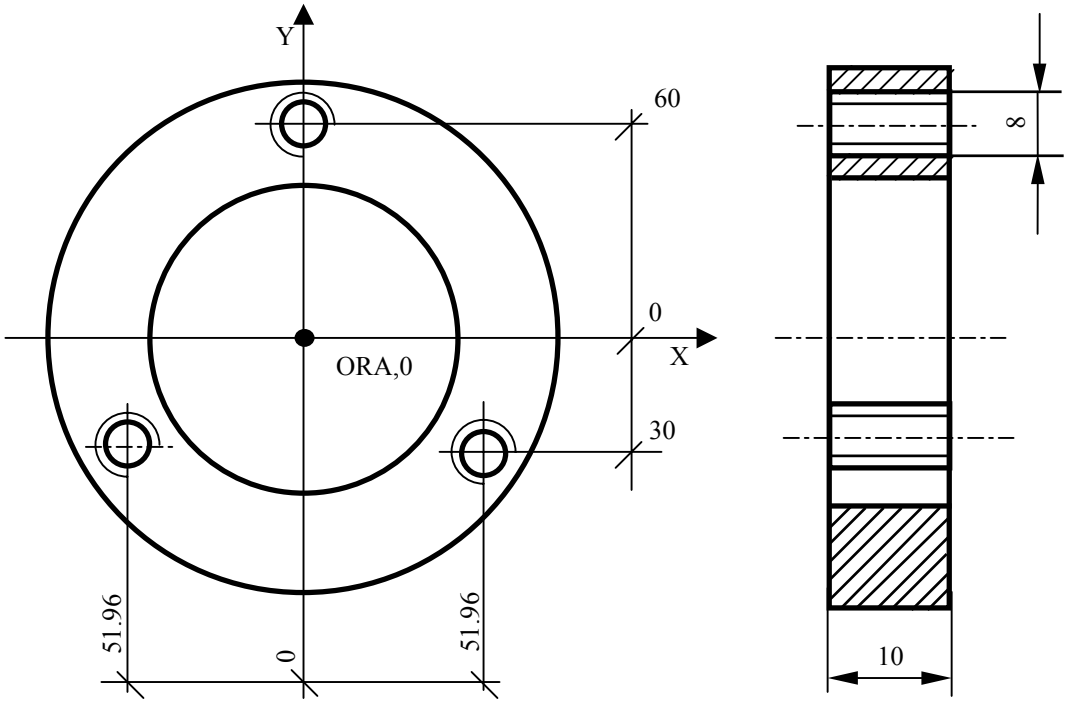
  $$F = S \times p \times 0.9$$

  where:

  S spindle speed
  p tap pitch
  0.9 feed decrease (necessary to keep under tension the spring compensator of the toolholder)

  . The final Z dimension must be reduced by a quantity equal to 10% of the actual tap working travel.

Fig. 5.20. - Tapping cycle without transducer




```
N90 (DIS,"TAP M8-TRACTION COMPENSATOR")
N91 S280 F315 T8.8 M6 M13
N92 G84 R-7   Z-15
N93 X-51.96 Y-30
N94 X51.96
N95 X Y60
N96 G80 Z50 M5
```

Use the sequence in the example to program a right-hand tapping operation (with M3 programmed in block N91).
To obtain left-hand tapping, program (or M04) instead of M13 (or M03).

N.B.: If the Z work travel is not long enough to allow the axis to reach the programmed feedrate and to stop with controlled deceleration (calculated on the spindle stop time), an error message will be displayed.

- if the spindle features a transducer, you can either program G84 with F (as in the previous example) or specify K (tap pitch). In this case, the control automatically calculates feedrate (as the product of K by the spindle's rpm).

  The programming block has the following format:

  N.. G84 [R1.. [R2..] Z..[K..]

  where:

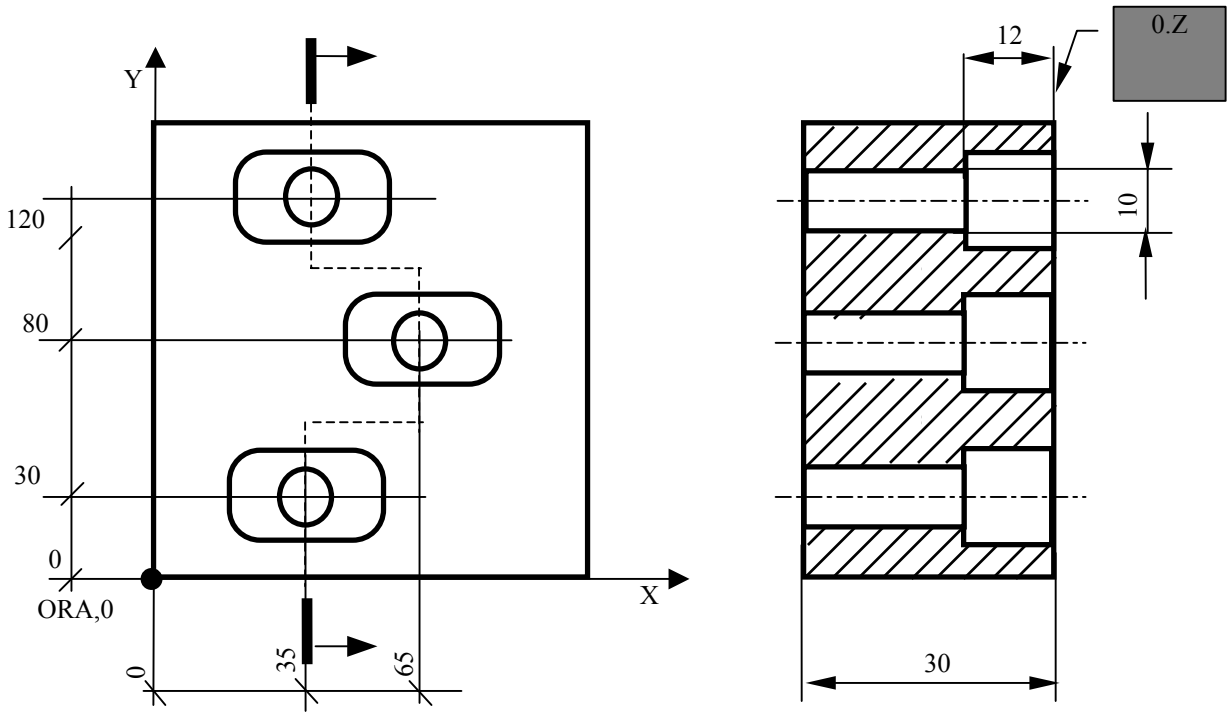
  G84 fixed tapping cycle
  [R1] rapid approach dimension and machining return
  [R2] rapid return dimension
  Z X012 end of tapping dimension
  [K] thread tap pitch

  Example:

```
N90  (DIS,"TAP M8")
N91  S280 T8.8 M6 M3
N92  G84 R 7 Z-15 K1
N93  X-51.96 Y-30
N94  X51 .96
N95  X Y60
N96  G80 Z M5
```

Example of fixed cycle programming with two R dimensions (R1-R2)

Fig. 5.21. - Drilling With R1-R2




```
N42 (DIS,"TWIST DRILL D=10")
N43 S850 F100 T4.4 M6
N44 G81 R-10 R2 Z36 M3
N45 X35 Y40
N46 X65 Y80
N47 X35 Y120
N48 G80 Z50 M5
```

CHARACTERISTICS OF FIXED CYCLES

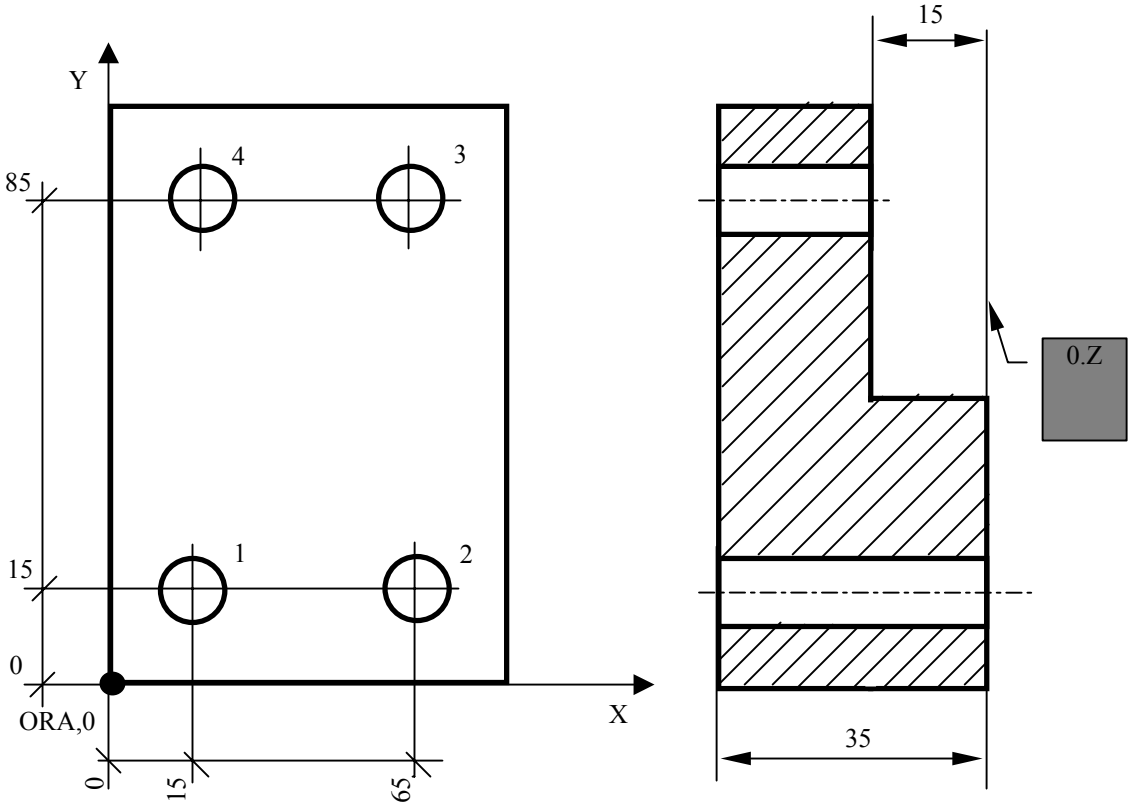
- If within a fixed cycle you program a block such as:
  X, Y, R or
  X, Y, R and/or Z

  the cycle will be executed in this order:

  . X and Y
  . R updating
  . Z updating

  This allows you to to vary the hole depth or the working plane (i.e. pass to a lower plane) without previously closing the fixed cycle with G80. For example,

Fig. 5.22.




```
    N35 (DIS,"TWIST DRILL D=8")
    N36 S1000 F100 T4.4 M6
    N37 G81 R3 Z-42 M3
1   N38 X15 Y15
2   N39 X65
3   N40 Y85 R-13
4   N41 X15
    N42 G80 Z50 M5
```

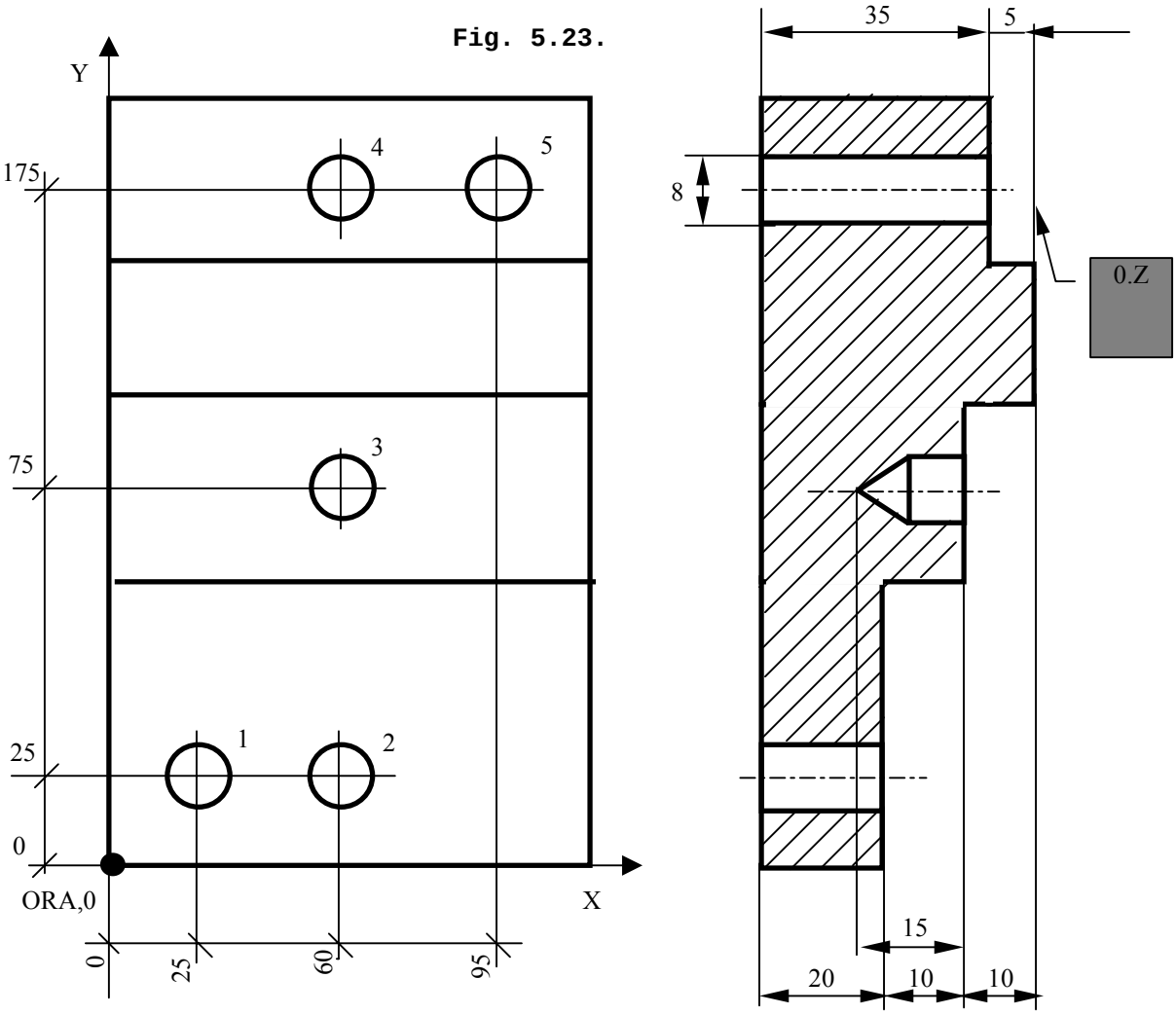
- If you program a block such as:
  X Y R1 R2 (last hole of the lower plan)

  the cycle will proceed in this order:
  . X and Y move to the point
  . the fixed cycle is executed and R1-R2 updated (at cycle end the spindle rapids to the new return R2)

  This allows you to change the working plane (i.e. pass to a higher plane) without first closing the fixed cycle with G80. To obtain the first hole on the higher plane, program the following block:
  X Y R2

  Here are two examples of this feature:

Fig. 5.23.



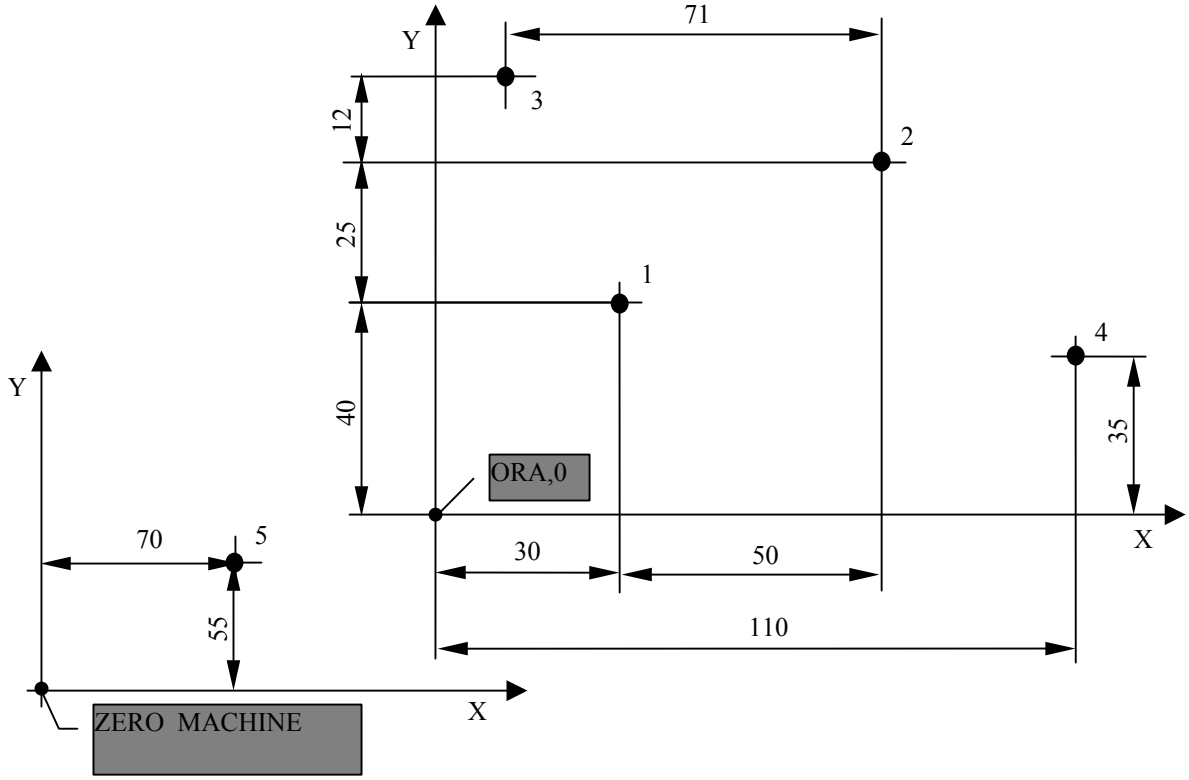

```
    N42 (DIS,"TWIST DRILL D=8")
    N43 S1000 F100 T5.5 M6
    N44 G81 M-18 Z-46 M13
1   N45 X25 Y25
2   N46 X60 R-18 R-8
3   N47 Y75 M-8 R200 Z-25
4   N48 Y175 R-3 Z-46
5   N49 X95
    N50 G80 Z50 M5
```

5.8.8. 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. 5.24.



N1 G X Y -X and Y position on the origin
N2 X30 Y40 -X and Y posdition on point 1
N3 G91 X50 Y25 -X and Y posdition in incremental mode on point 2 (X+50, Y+25 with resdpect to point 1)
N4 X-71 Y12 -X and Y position in incremental mode on point 3 (X-71, Y+12 with resdpect to point 2)
N5 G90 X110 Y35 -X and Y posdition in absdylute mode on point 4 (X+110, Y+35 with resdpect to the origin)
N6 G79 X70 Y55 -X and Y posdition on point 5 referred to machine zero (X+70, Y+55 with resdpect to machine zero).

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.

5.8.9. 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.
  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.

5.8.10. MODAL GROUP K: PROBING CYCLES (G72-G73-G74)

To establish a probing cycle, program the following blocks:

G72 measurement of a point with linear movement and radius compensation
G73 measurement of the parameters of a hole
G74 measurement of the variance between a nominal and a real coordinate (without radius compensation)

G72 probes the coordinates of a point in the space with a linear movement and radius compensation. The calculated values are stored them in the E parameters previuosly defined in the cycle, starting from the programmed variable. The allowable format is:

G72 axis [axis] [axis] En

where:

[axis] programmed axes (max. 3); movements are executed with nominal dimensions.

En starting variable for storing probed dimensions

Example:

G72 X100 Y50 E32 the dimensions calculated for X and Y are stored respectively in E32 and E33 independent from the order in which they have be programmed.

Notes:

- G72 can be used with rotation instruction URT. In this case, both axis names must be programmed in the block.

- If one of the coordinates is zero, it must be programmed with its value before the E variable. For example:
  G72 X100 Y0 E32

- G73 probes (with tool radius compensation) the parameters (center coordinates and radius) of a hole lying on the current interpolation plane. The results are stored in the previoulsy defined E variables.
  The axes must already be on the center of the hole.

The allowable format is:

G73 r En

where:
r theoretical hole radius
En initial variable for storing hole parameters

Example:
G73 r 100 E55 center abscissa. Ordinate and circle radius will be respectively stored in E55-E56-E57

- G74 constitutes a tool requalification cycle. The control checks the variance between nominal dimensions (i.e. measured by an electronic gauge) and the dimensions measured by the tool on the spindle.
  G74 can also be used for tool wear check. Since measures are taken without radius compensation, the resulting values are regarded as the actual "tool" dimensions.
  The allowable format is:

G74 axis [axis] [axis] En

where:
[axis] max. 3 simultaneous axes.
En starting variable for storing the variances

Example:
G74 X60 E41 E41 = Pm - Pt
where:
Pm = measured point
Pt = theoretical point

The steps in this cycle are similar to those executed with G72. The main differences between the two cycles are the absence of radius compensation and the fact that variances (and not dimensions) are stored in the variables.

Important:

- In all of the measure cycles, the parameters stored by the system in E variables are expressed in the characterized unit. If your program includes the alternative unit, you

must introduce all the necessary conversion factors in the calculations that use those parameters.

- G72 and G73 can be progammed in the same block as URT (axes rotation command) and MIR (axes inversion command).

  If both URT and MIR are present in a block, URT must always be programmed first.

  URT and MIR can also be programmed in the same block as URP (part rotation command).

5.8.11. MODAL GROUP L: V/D FEEDRATE CODING (G93)

G93 allows you to express feedrate as inverse time.

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

If you know the velocity and the distance of a move, you can use the following formulas to calculate F word value:

Linear interpolation:

$$F = \frac{\text{Velocity}}{\text{Distance}}$$

Circular interpolation:

$$F = \frac{\text{Velocity}}{\text{Arc}}$$

where:

Velocity	linear or circular velocity in mm/min (G71) or in/min (G70)
Distance	vectorial distance of linear programmed axis motion in mm or in
Arc	development of the programmed arc in mm or in

With G93 active, the F word applies only to the block in which it is programmed. For example:

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

The programmed F indicates the number of blocks that can be executed in 1 min. For example:

```
X100 F1          -the element is executed in 1 min.
X100 F10         -the element is executed in 6 sec.

                  that is. 10 blocks are executed in 1 min
```

The feedrate is displayed in mm/min and cannot be higher than the value established during characterization.

5.9. M19: SPINDLE ANGLE ORIENTATION

With M19 you can program the spindle to move to a specified angle and stop.

This feature can be used for back spot-facing operations, in which you must position the spindle, move Y axis (or X axis according to blade orientation), enter the hole, position the spindle with reference to the axis and then start machining.

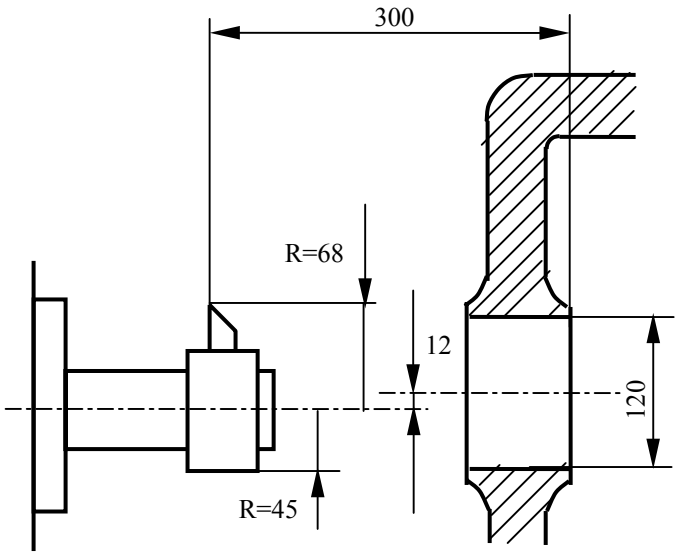
Spindle anqle orientation can also be used for high-accuracy reaming operations, to avoid scoring the reamed surface during the Z axis return. In this case, you finish the hole, then orient the spindle, move X or Y axis (according to the blade orientation), and return the Z axis.

To cancel M19 program M03, M04, M13 or M14.

In a programming block, M19 is always executed before any other movement instruction.

Here's an example of back spot-facing with M19.

Fig. 5.25. - Spot facing with M19




```
N32 (DIS,"BACK SPOT-FACING BAR D=136")
N33 S115 F20 T7.7 M6
N34 G X250 Y-12 M19
N35 Z-306
N36 Y M3
N37 TMR=2
N38 G1 G4 Z-300
N39 G Z-302 M5
N40 Y-12 M19
N41  Z
```

In N34, the position of X and Y and the spindle orientation have been programmed so as to enable the tool to go through the hole having a diameter 0=120.
N35 programs Z poisitioned at spot-facing start.
In N36, the tool axis moves to overlap with the spot-facing axis.
N38 programs spot-facing. In the subsequent blocks, the tool is oriented, drawn from the part and eventually positioned along Y axis, so that it can go through the hole of 0=120.

5.10. M10: AXES LOCKING

M10 allows you to lock axes not used in the current program.

In continuous mode (with G27-G28), you must both program M10 and specify the axes that must remain active; otherwise, they will be locked as well and, upon a motion request, the followinq error message will be displayed: SERVO ERROR

In point-to-point mode (with G29), you can simply program M10 without specifying any axis.

Examples:

```
a)   N9  G G29 X100 Y100
     N10 G1 X-100 M10 F250
     N11 Y-100
     N12 X100
     N13 Y100
     N14 G X.. Y.. M11
     ..........

b)   N9  G G27 X100 Y100
     N10 G1 X-100 Y100 M10 FZ50
     N11 Y-100
     N12 X100
     N13 Y100
     N14 G X.. Y.. M11
     ..........
```

In the first example, only tne moving axis is unlocked, whereas in the second example, X and Y -- recalled in block N10 -- are always unlocked.
To cancel M10 program M11.
The same criteria apply to M12 (rotary axes locking).

5.11. PARAXIAL COMPENSATION FACTORS (u,v,w)

Paraxial compensation factors u,v,and w are applied to proqrammed coordinates according to the following formulas:

```
X= programmed X +  (tool  radius * u)
Y= programmed Y +  (tool  radius * v)
Z= programmed Z +  (tool  radius * w)
```

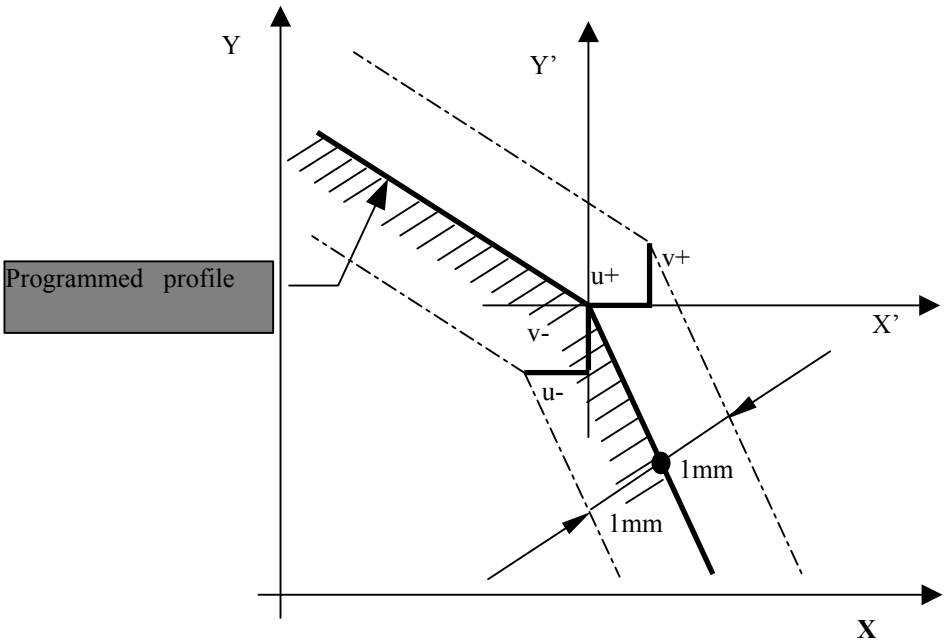
You can use them in simple profiles (contours parallel to axes) or to mill three-dimensional surfaces.

Compensation factors u, v and w can be calculated respectively as X, Y and Z coordinates of a cartesiao system parallel to the machine axes, whose center is the point to be compensated.

To establish a negative compensation factor, program a negative number: the positive sign is ellided.

Paraxial compensation factors are not compatible either with tool radius compensation (G41-G42) or with the GTL ambient.

Fig. 5.26. - How to determine compensation factors in a profile



This method allows you, to correct translating profiles in which the coordinate increment is directly proportional to the translation step. The control calculates the coordinate increment as follows:

$$\text{increment} = \frac{\text{compensation factor} * \text{tool diameter}}{2}$$

Paraxial compensation factors can be applied to program:

- profiles made up of straight segments, parallel or 00000 to the machine axes;

- profiles made up of straight lines tangent to arcs;

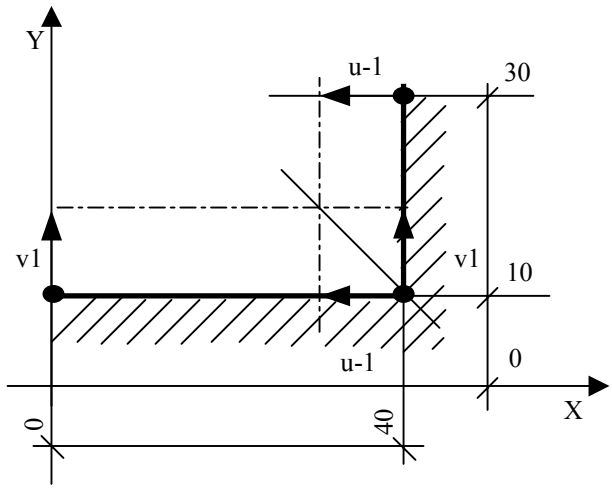
- profiles made up of tanqent arcs, on the condition that they continue to be tangent after translation;

Important. Paraxial compensation factors are only active in the block in which they have been programmed. Also they must always be accompanied by the coordinates to which they apply, that is:

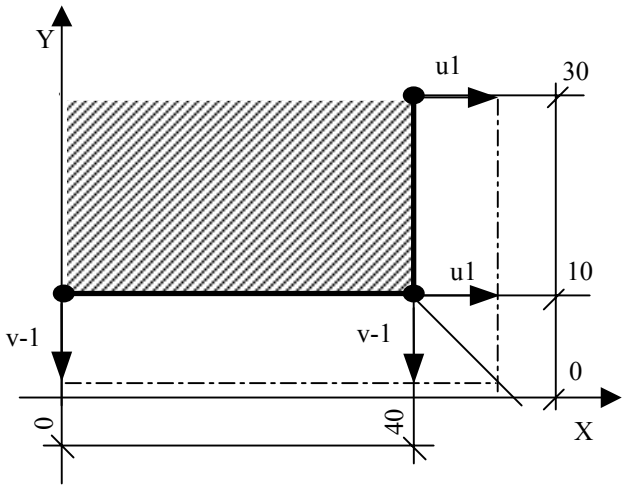
- u with X coordinate (axis 1 or equivalent)
- v with Y coordinate (axis 2 or equivalent)
- w with Z coordinate (axis 3 or equivalent).

Fig. 5.27. - Examples of programming sequences with u-w-w

a)

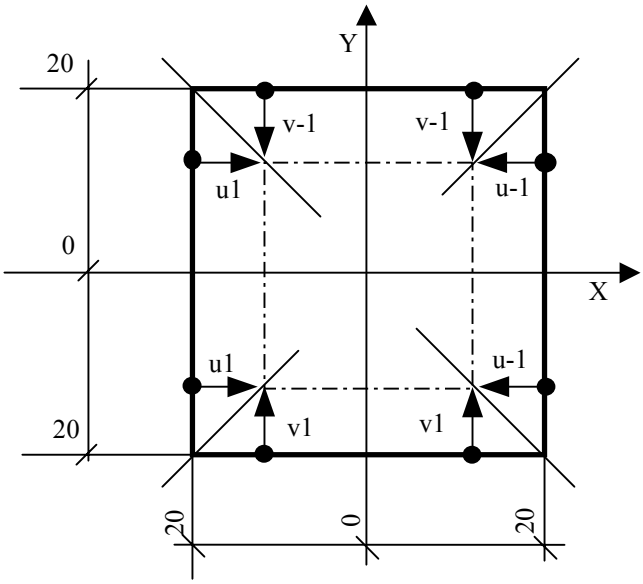



```
N5  T1.1 M6 S.. F..
N6  G X Y30
N7  G1  Y10 v1
N8  X40 u-1
N9  Y30
N10 G X.. Y..
```



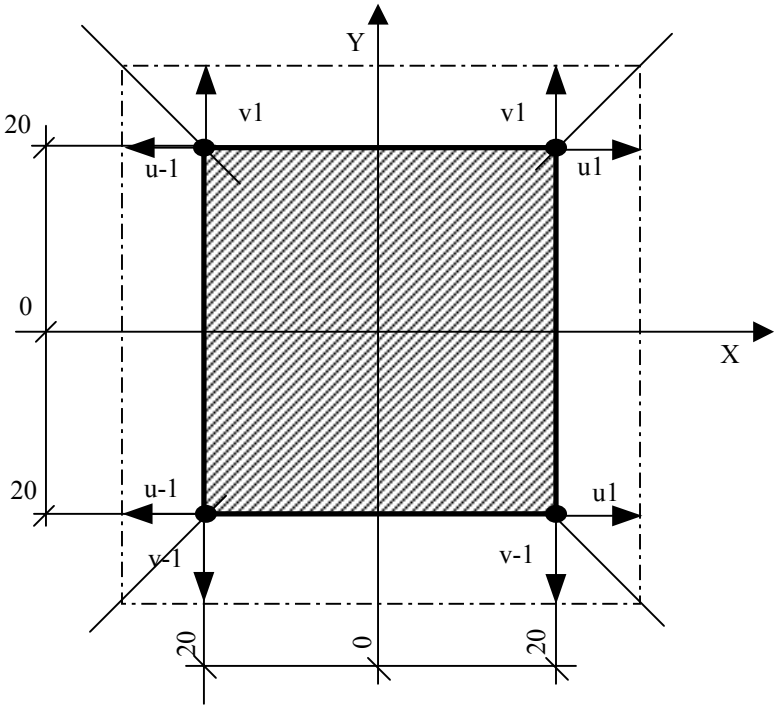
b)

```
N13 G X Y
N14 Y10 v-1
N15 X40 u1
N16 Y30
N17 G X.. Y..
```



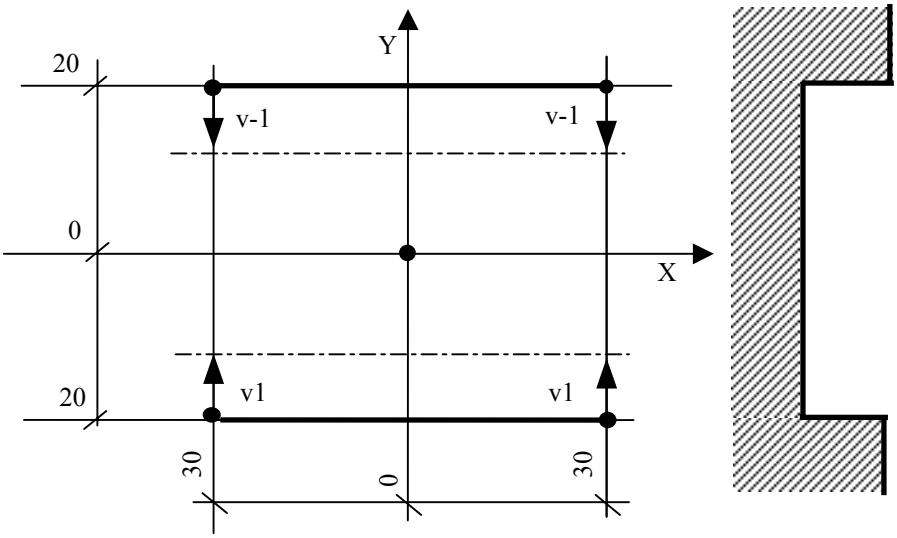
c)

```
N13 G X Y
N14 G1  Z-10
N15 X-20 Y-20 u1 v1
N16 X20 u-1
N17 Y20 v-1
N18 X-20 u1
N19 Y-20 v1
N20 G X Y
```



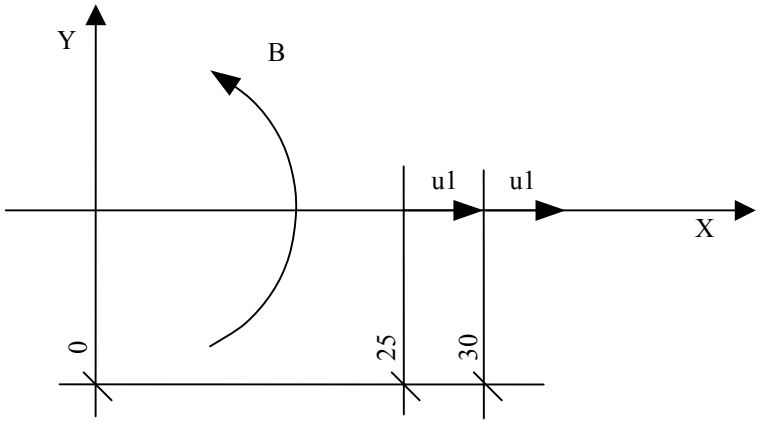
d)

```
N13 G X-35 Y-35
N14 Z-10
N15 G1  X-20 Y-20 u-1 v-1
N16 X20 u1
N17 Y20 v1
N18 X-20 u-1
N19 Y-20 v-1
N20 GZ
```



e)

```
N12 .........
N13 G X-30 Y
N14 G1 Y20 v-1
N15 X30
N16 Y-20 v1
N17 X-30
N18 .........
```



f)

```
N12 G X40 Y B0
N13 G1 X30 u1
N14 X25 B360 u1
N15 G X40
```

5.12. INDEXING TABLE

To program the rotation of an indexing table, enter the name of the characterized axis followed by the number of indexes. An indexing table can be programmed in absolute or incremental mode (with G90 or G91).

Example:

```
..........
..........
N24 B10
..........
..........
N41 G91 P2
..........
..........
N80 G90 B0
..........
..........
```

The command for the rotation of the table is a starting instruction; therefore, linear movements are executed only after the table has shifted to the programmed angle.

When programming movements of an indexing table remember the following:

- at each power-up, the indexing table must be reset to zero via keyboard;
- the axes symbols are those of the machine axes;
- NC-110 can handle as many as 3 indexing axes;
- the number of indexes can be programmed in explicit or implicit mode. To program it implicitly, use an E real parameter (E25÷E29). The upper limit is 99999.

5.13. ROTARY AXIS - CONTINUOUS ROTARY TABLE

You can declare a rotary axis when you configure the system.

To program a continuous rotary table (i.e., simultaneous and coordinated with the axes programmed in the same block), remember that:

- movements must be programmed in degrees (integer or decimal), from ±0.0001 to ±99999.9999, referred to a preset origin;

- you can select either rapid (G00) or machining rates (G01). Program velocity in degrees/min (with word F and as many as 2 decimal digits)

For example, F75.5 programs a velocity of 75.f degrees/min.

When programming milling along a circumference. Calculate angular velocity with the following formula:

$$F = \frac{360}{¶} * \frac{A}{D} = 114.64 * \frac{A}{D}$$

where:

F angular velocity in degrees/minute
A linear velocity along the circumference mm/min
D milling diameter (in mm).

When rotary axes move with linear axes, calculate velocity with the following formulas:

with G94

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

with G93

$$F = \frac{A}{L}$$

where:

F	velocity
A	feedrate (in mm/min) on part
X Y	actual travel performed by each axis (in mm for linear axes, in degrees for rotary axes)
L	length of the resulting path (in mm).

The resulting path can be an arc (if only the rotary axes moves) or a complex curve (an Archimedean spiral, a cylindrical helix, etc.), according to the linear axes coordinated to the rotary axis.

Note:

- In G93 and G94 you can enter the time of block execution, thus avoiding to calculate the feedrate. The time is indicated with a lowercase "t" and is expressed in seconds.
  E.g.: G1 Y100 B20 t6
  (We remind you that to enter a lowercase letter you must press key and then the desired letter).

5.13.1. ROLLOVER ROTARY AXES

A rotary coordinate axis can be characterized as having rollover.

With this feature, the displayed position of movements programmed via part program or keyboard will always remain within the 0÷359.9999 range.

It can be used either with absolute (G90) or incremental programming (G91).

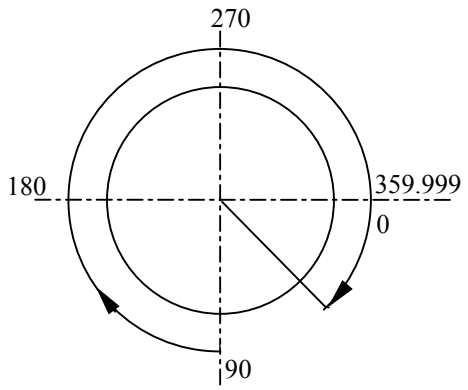
With G90, the allowable range is ± 0÷359.9999. The programmed angle corresponds to the desired position and the sign indicates the sense of rotation.

Example 1:

Let's asssume that rotary axis B is positioned to 90 degrees and that the following block is programmed or input:

8 45

The axis rotates by +315 degrees and positions to 45 degrees, while the dimension B45 appears on the CRT

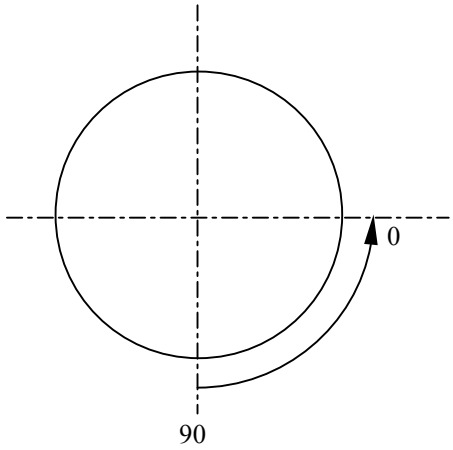


Example 2:

Let's asssume that rotary axis B is positioned to 90 degrees and that the following block is programmed or input:

B-0

The axis rotates by -90 degrees and positions to 0 degrees, while the dimension B0 appears on the CRT



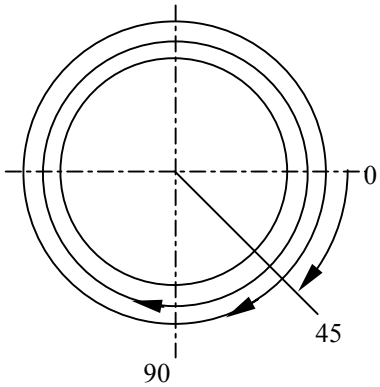
With G91 the programmable range is ±0.0001 ÷ 99999.9999, the sense of rotation is determined by the increment sign.

Example 3:

Let's assume that rotary axis B is positioned to 0 degrees and that the following block is programmed or input:

G91 B765

The axis rotates by two complete turns plus 45 degrees, while the dimension B45 appears on the CRT:.



Notes:

- When programming a real rollover rotary axis to execute a profile, select the increment mode (with G91). If you program virtual axes (refer to section 5.22.), make sure the profile does not exceed from 360 degrees.

- When switching a rotary axis from rollover to normal or from normal to rollover, you must always re-establish tne axis absolute reference.

5.14. 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.

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 assignnent 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.

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 NC-110: 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 initialization, 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 whith the access format of he variable.

Examples:

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

When characterizing 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. 5.28. -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="12">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="6">S3.LI</td><td rowspan="6">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 adress a SYVAR variable starting from SYVAR0.BY:

SYVAR0.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 (TIM0) 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 synchronization (TIM preceded by #).

Example:

```
..........
N9  G0 ...
N10#TIM1=TIM0       -assign the system time (clock) to TIM1
..........
..........
N89 G0 ...
N90#TIM2=TIM0-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 TOT0 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
N20# TIM1=TIM0
N30  T2.2 M6
N40  XY S2000 F500 M13
...........
...........
N200 TOT0=TIM0-TIM1
N210 TOT1=TOT0*E1
N220 TOT2=TOT1/3600
N230 (DIS,TOT2)         -display the time required to execute
...........              a batch of 75 parts (in hours)
```

MAXIMUM FORM ERROR - ERF

ERF defines the maximum form error allowable with the NC-110. The format of the instruction is as follows:

ERF=value

Where:

value is a distance in inches or mm (the measure unit can be selected with G70 and G71).

The control executes a theoretical deceleration profile in G27 mode. This profile causes axis deceleration and does not allow the error to overflow the limit programmed with ERF.

Important: This code can be assigned in section *4 of the PGCFIL file. The default value (0) can be changed from part program or from keyboard. If ERF=0, the axis will decelerate to 0 at the end of each block.

MAXIMUM DEVIATION OF DIRECTORY COSINES - MCD

This instruction defines the maximum angle for an axis when G27 is active. The allowable value ranges from 0 to 2 and selects an angle that represents the limit of the G27 function. If the actual angle is greater than this value, the motion will be performed in G29 or in point-to-point mode.

Important: This code can be assigned in section *4 of the PGCFIL file. The default value (0) can be changed from part program or from keyboard.

The allowable format is as follows:

MCD=x

Where:

x is a real number from 0 to 2. This number represents the maximum desirable angle for the axis when G27 is active.

If G27 is active, the desired angle is ≤ 90 degrees. MCD is calculated as the sine of the maximum angle. For example,

MCD= sine of 90 degrees = 1

If G27 is active, 180 degrees ≥ desired angle > 90 degrees. Then MCD is calculated as follows:

MCD = 1 + [sine (angle - 90)]

For example, if the maximum desired angle is 180 degrees, MCD will be:

MCD = 1 + [sine (180-90)]
MCD = 1 + 1
MCD = 2

The NC-110 will force deceleration of the programmed axis to zero if the angle displacement is greater than the angle defined by the MCD instruction. Conversely, the control will calculate a deceleration ramp for the programmed axis if the angle displacement is less than or equal to the angle defined by the MCD instruction.

5.15. 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":

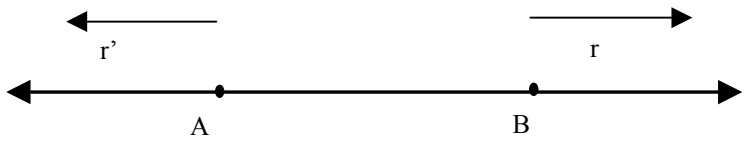
- reference origins
- points
- straight lines
- circles

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 assignement of a direction of motion.

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

Fig. 5.29. - Straight lines r and r' with orientation

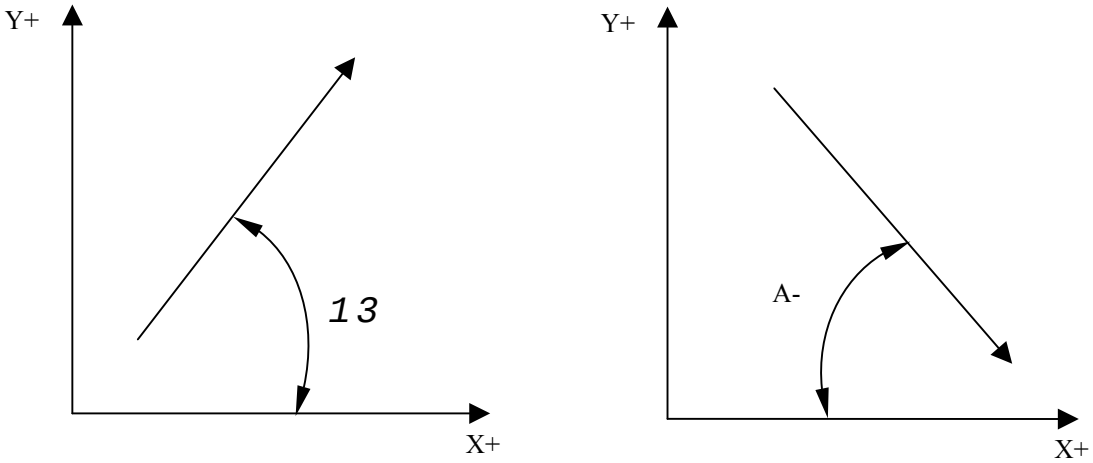


In oriented geometry, r and r' are two different straight lines that have opposite directions.

With GTL and oriented qeometry, 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 abscissa axis.

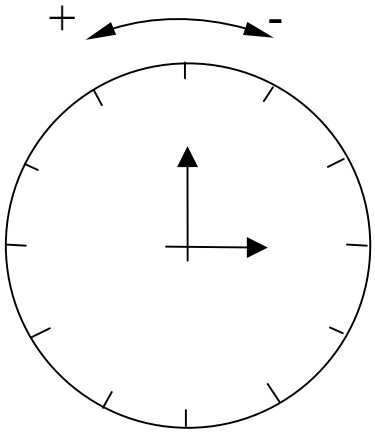
To determine the direction of a straight line, rotate the positive abscissa axis until it coincides with the straight line. The direction will be positive if you rotated CCW, or negative if you rotated CW.

Fig. 5.30. - Angles convention



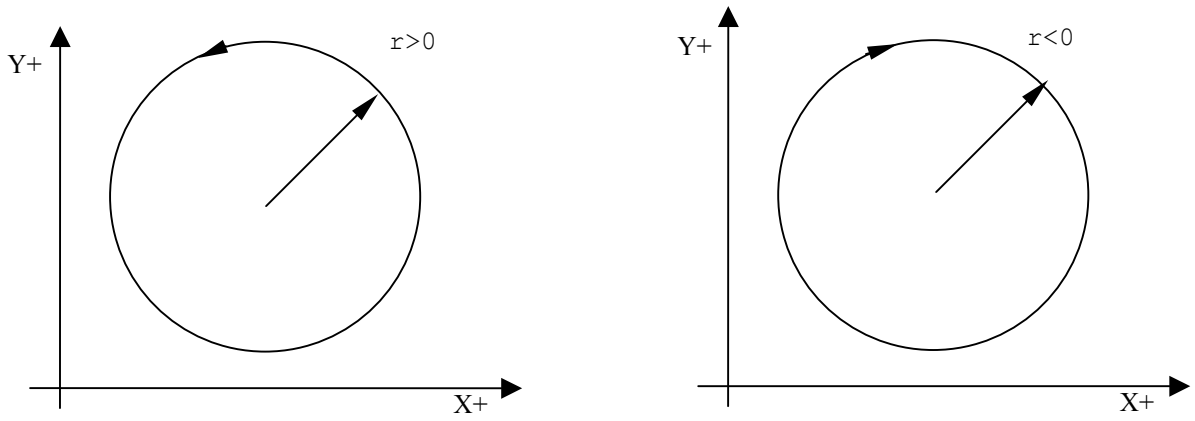
Circles must also have directions associated with them. The direction of a circle is positive if movement is CCW, or negative if movement is CW.

Fig. 5.31. - 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. 5.32. - 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, radii, 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 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 orimins. 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 nunber between 0 and 255 inclusive.
The max. limit is defined during sysem configuration.

expression — contains all the information necessary to describe tihe geonetric element.

You can define elements in:

- direct (explicit) mode, i.e., programm 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  = X30 Y30 a45
p1  = o1  X15 Y15
p2 = X60 Y30
l1  = p1, p2
l2 = X30 Y50, a45
c1 = l1,  l2,  r15
l3 = X0 Y0, X100 Y60
p3 = l3, c1
c2 = p3, r8
..........
```

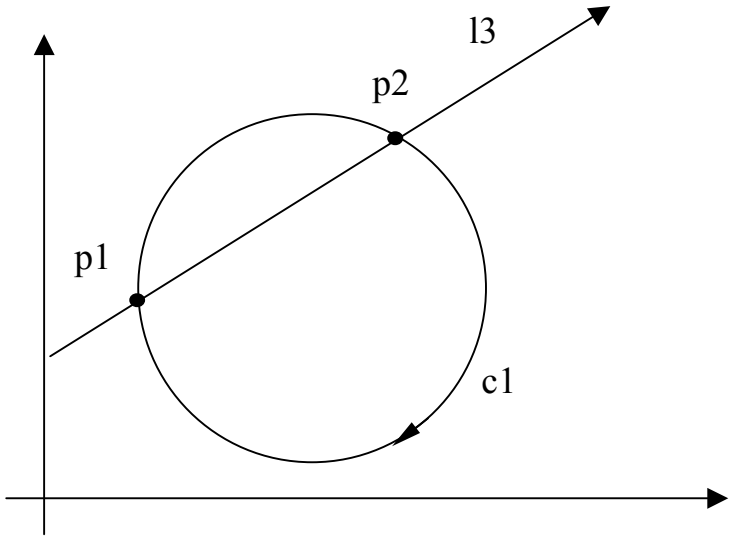
The number of geometric elements you can store is defined during system configuration.

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

Examples:

```
p1 = X30 Y30                the separator
c1 = I10 J20 r30            is not required


l1  = X20 Y20, X100 Y-10
        |          |
      point      point


l2 = I30 J20 r10, X80 Y80
         |          |
       circle      point


l3 = X100 Y100, a45
        |       |
      point   angle


p5 = c1, c2
     |   |
     |  circle
    circle


c3 = l1, l2, r18
     |   |   |
     |   |  radius
     |  line
     |
    line
```

In figure 5.33., s2 selects the second intersection.




```
p1 = c1, l3

p2 = c1, l3, s2
```

Figure 5.33.

The allowed formats for defining geometric elements are shown in Table 5.7.

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 5.7. - GTL DEFINITIONS

Reference origins	on = X.. Y.. a..
Points	pn = [om] X.. Y.. pn = [om] m.. a.. pn = lm, lp pn = [-]lm, cp [,S2] pn = cm, [-]lp [,S2] pn = cm, cp [,S2]
Straight lines	ln = [om] X.. Y.., [op] X.. Y.. ln = [om] X.. Y... a.. ln = [om] I.. J.. r.., [op] I.. J.. r.. ln = [om] I.. J.. r.., a.. ln = [om] I.. J.. r.., [op] X.. Y.. ln = [om] X.. Y.., [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 = X.. Y.. a..

where:

on — identifies the name and index of the reference origin

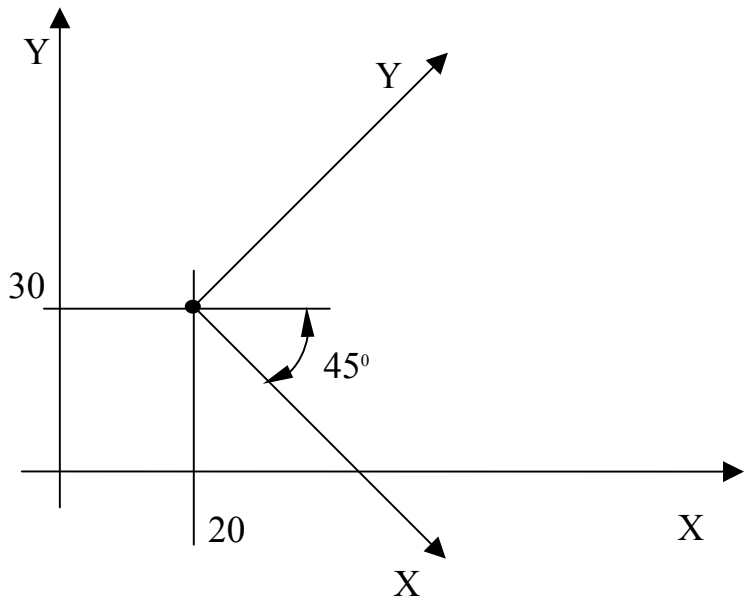
X.. Y.. — coordinates of the new origin

a.. — rotation angle (positive CCW)

Example:

Fig. 5.34.

O4 = X20 Y30 a-45

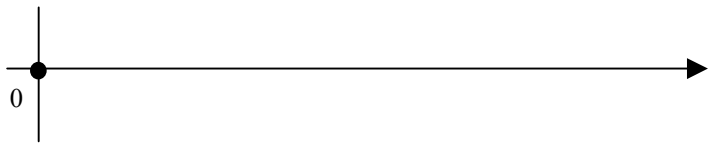


DEFINING POINTS

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 oriqin called pole that originates the X axis, which is called polar axis (Figure 5.35.).

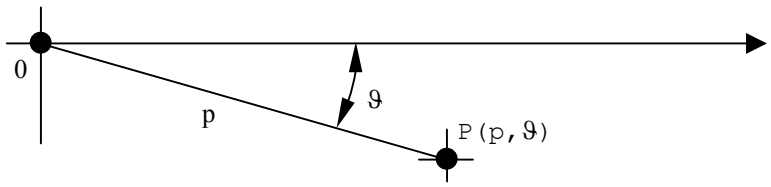
Fig. 5.35. - Polar axis



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 5.36.).

Fig. 5.36. - Polar coordinates



DIRECT PROGRAMMING FORMATS FOR POINTS

Point with cartesian coordinates (figure 5.39.)
pn = [on] X.. Y..

Point with polar coordinates (figure 5.41.)
pn = [on] m.. a..

INDIRECT PROGRAMMING FORMATS FOR POINTS

Intersection point of two predefined lines (figure 5.42.)
pn = lm,lp

Intersection point of a predefined line and a predefined circle (figure 5.43.)
pn = [-]lm,cp[,s2]
pn = cm,[-]lp[,s2]

Intersection point of two circles (figure 5.44.)
pn = cm,cp[,s2]

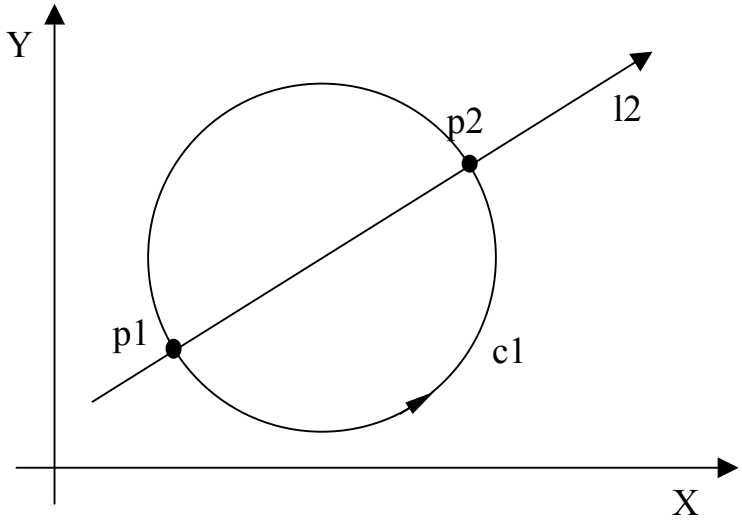
where:

pn	identifies the name of the point having index n (n is a number between 1 and the maximum configured number)
X.. Y..	point coordinates
[om]	reference origin previously defined having index n, to which the X and Y 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 5.37.): circle c1 and line l2 intersect at points p1 and p2. By traveling in the specified direction of straiqht 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. 5.37. - Straight line-circle intersection



Two solutions are also mossible for circle-circle intersections: the circles c1 and c2 intersect at points p1 and p2 (Fig. 5.38.). 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 5.38. - Circle-circle intersection

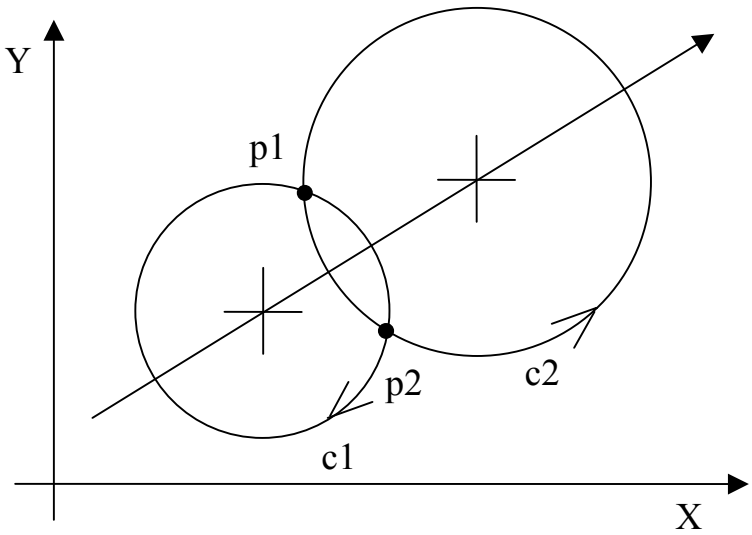
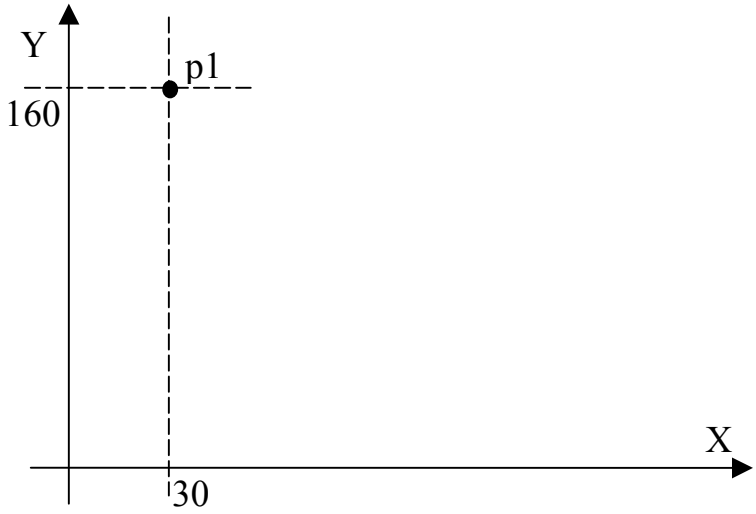
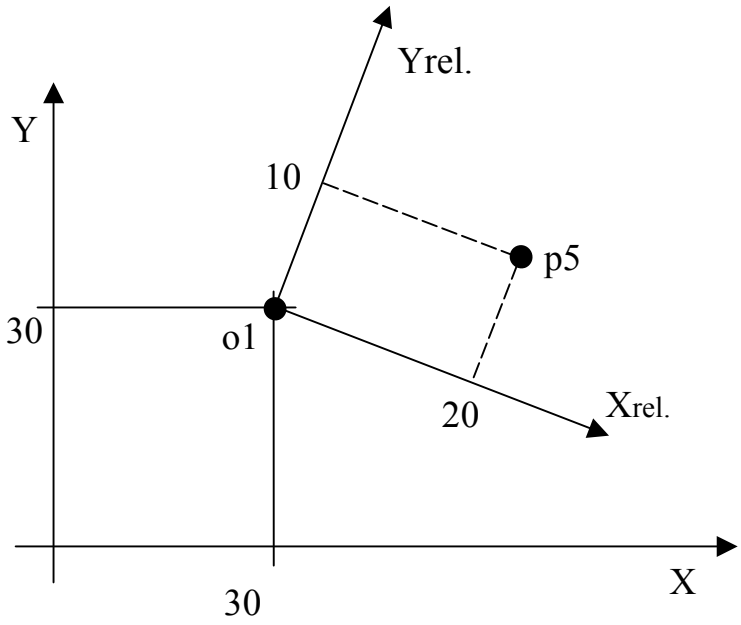


Fig. 5.39.



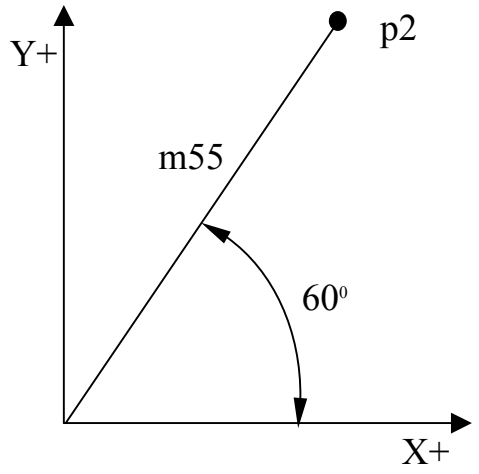

```
Pl = X30 Y160
```

Fig. 5.40.



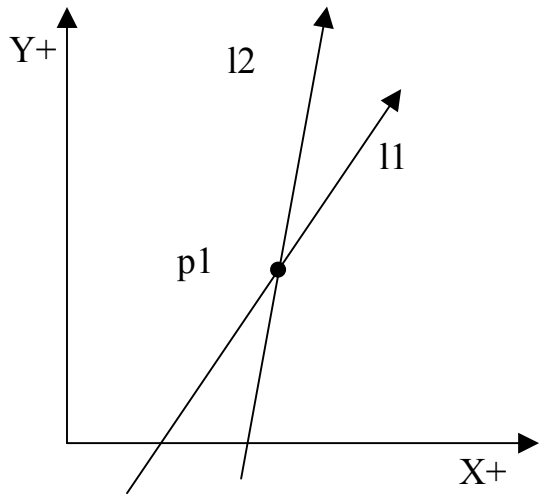

```
o1 = X30 Y29 a-20
P5 = o1 X20 Y10
```

Fig. 5.41.



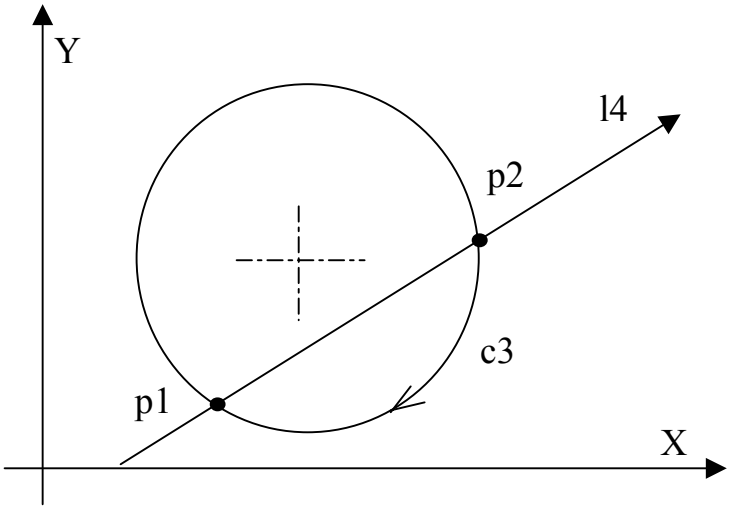

```
p2 = m55 a60
```

Fig. 5.42.



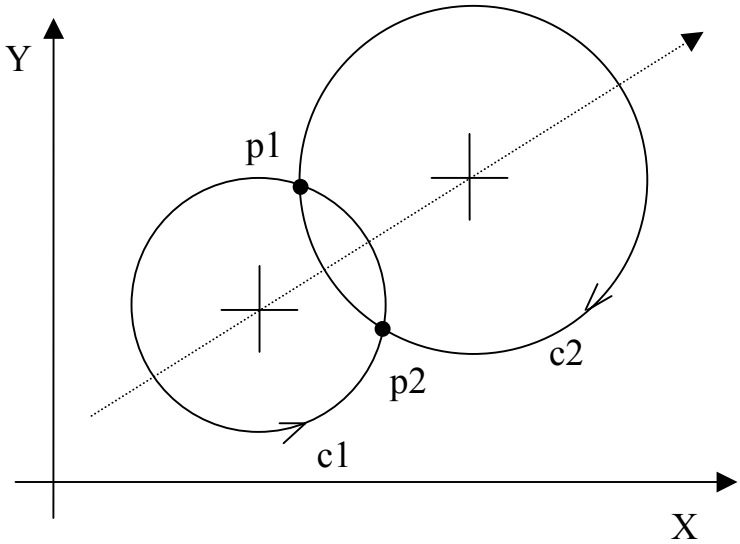

```
p1 = l1, l2
```

Fig. 5.43.




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

Fig. 5.44.




```
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. 5.45. - Direction Incompatibility of Geometric Elements

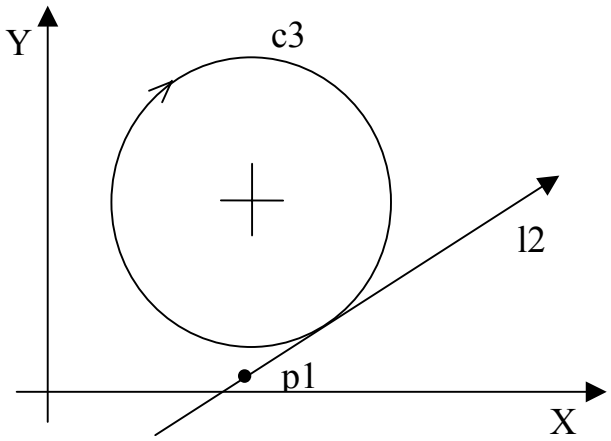
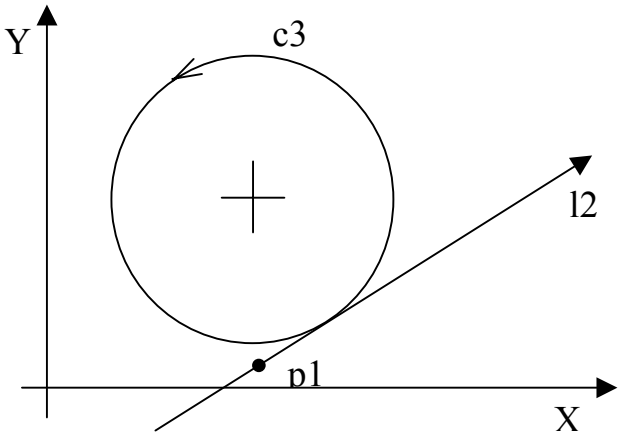


Fig. 5.46. - Direction Compatibility of Geometric Elements



Direct Programming Format

Straight line through two points (fig. 5.47.)
ln = [om] X.. Y..,[op]X.. Y..

Straight line through one point and forming an angle with the abscissa axis (figs. 5.49., 5.50.)
ln = [om] X.. Y..,a..

Straight line tangent to one circle and forming an angle with the abscissa axis (figs. 5.51., 5.52.)
ln = [on] I.. J.. r..,a..

Straight line tangent to two circles (fig. 5.53., 5.54.)
ln = [om] I.. J.. r..,[op] I.. J.. r..

Straight line tangent to one circle and through one point (fig. 5.48.)
ln = [om] I.. J.. r..,[op] X.. Y..
ln = [om] X.. Y..,[op] I.. J.. r..

Indirect Programming Format

Straight line through two points (fig. 5.55.)
ln = pm,pq

Straight line through one point and forming an angle with the abscissa axis (fig. 5.49.)
ln = pm,a..

Straight line tangent to two circles (figs. 5.57., 5.58.)
ln = [-]cm,[-]cp

Straight line tangent to one circle and forming an angle with the abscissa axis (fig. 5.60.)
ln = [-]cm,a..

Straight line tangent to one circle and through one point (fig. 5.56.)
ln = [-]cp,pm
ln = pm,[-]cp

Straight line parallel to one straight line at d distance (figs. 5.61., 5.62.)
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)
X.. Y..	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. 5.47.

```
l1 = X40 Y20,X60 Y70
```

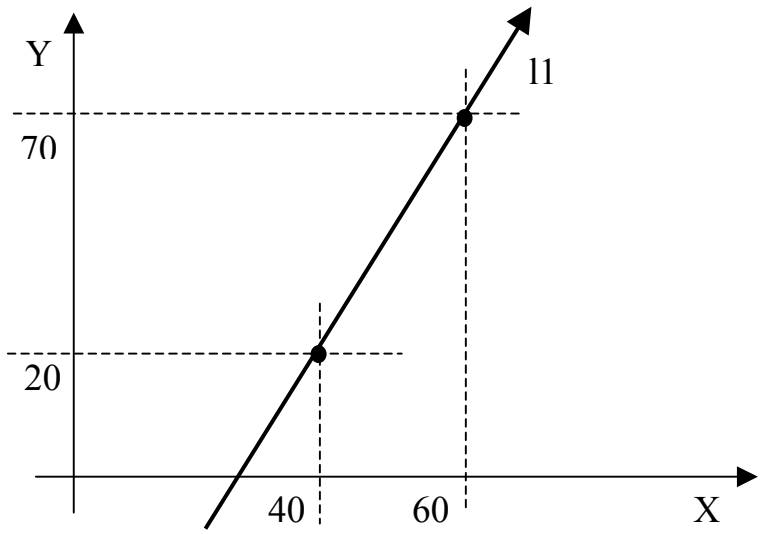


Fig. 5.48.

```
l1 = X10 Y15,I45 J30 r-15
l2 = X10 Y15,I45 J30 r15
```

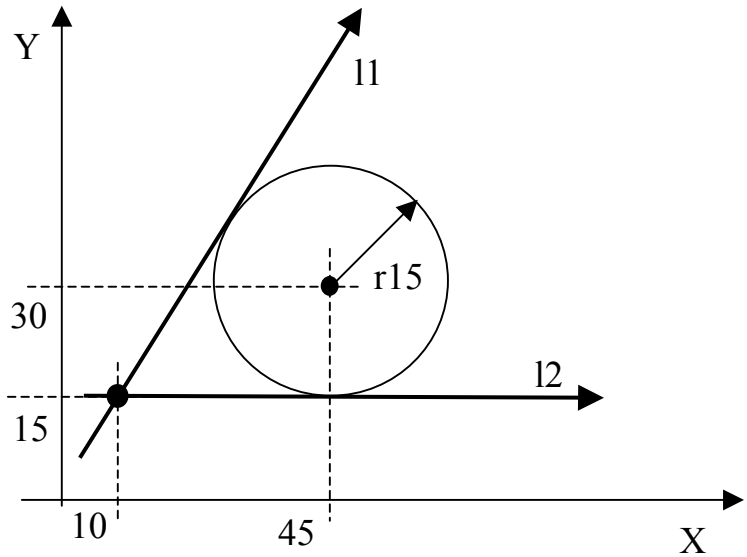


Fig 5.49.

```
l2 = X50 Y60,a-20
```

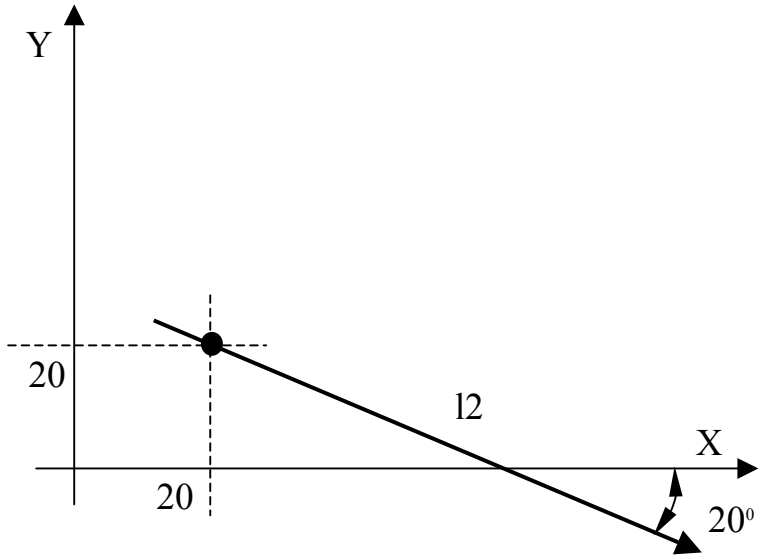


Fig. 5.50.

```
l5 = o1 X25 Y30,a60
```

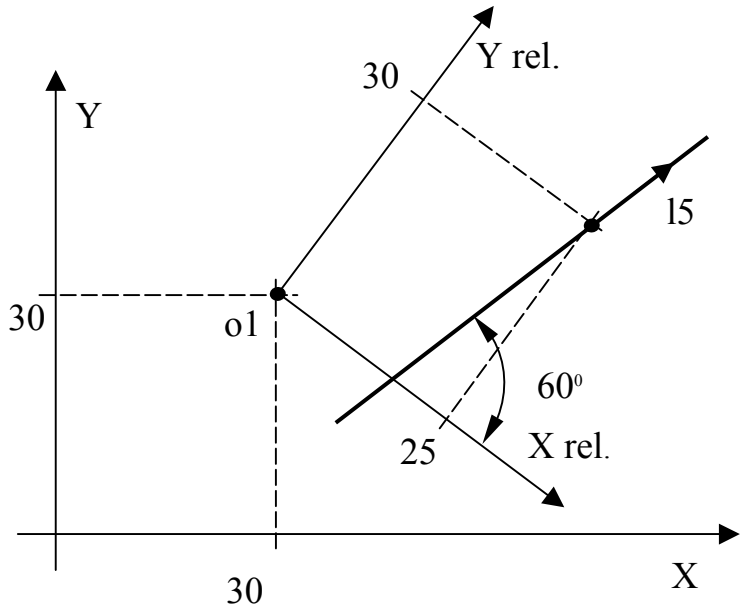


Fig. 5.51.

```
l1 = I60 J80 r40,a45
```

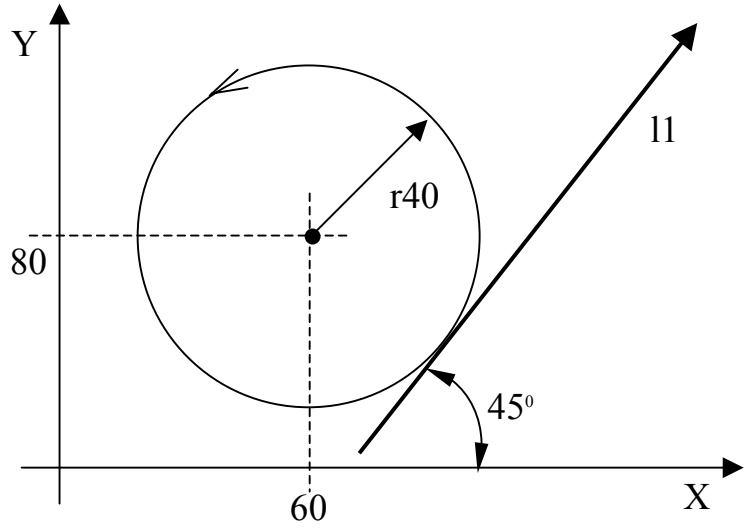


Fig. 5.52.

```
l4 = o3 I25 J15 r10,a115
```

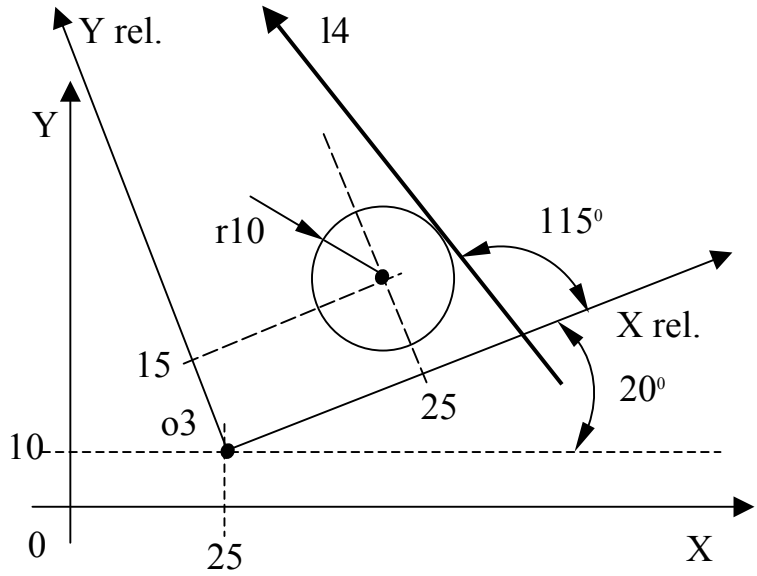


Fig 5.53.

```
l3 = I25 J35 r-17,I70 J20 r13
```

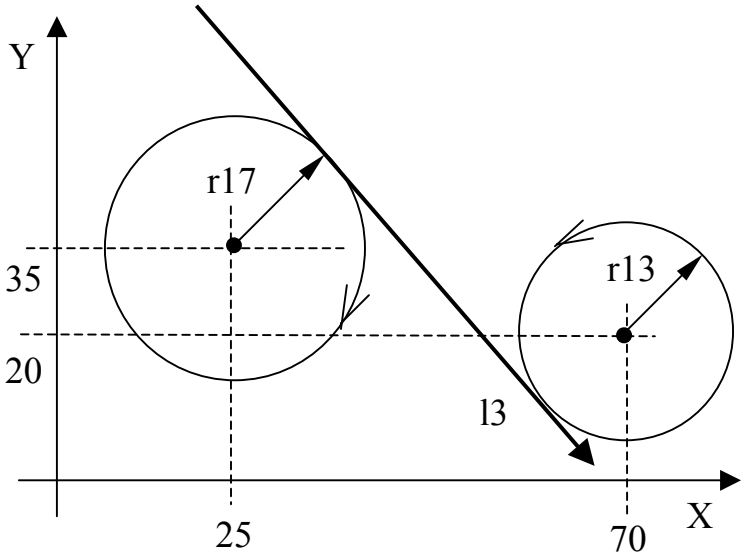


Fig 5.54.

```
l4 = I25 J35 r17,I70 J20 r13
```

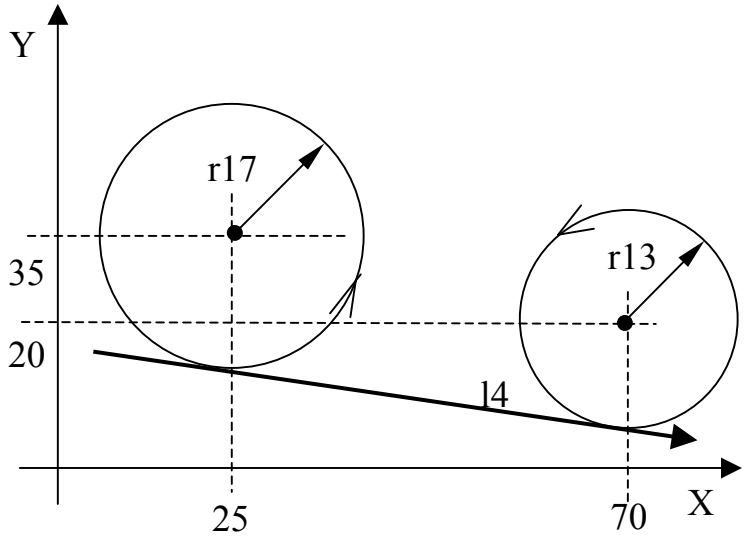


Fig. 5.55.

```
l9 = p7,p8
```

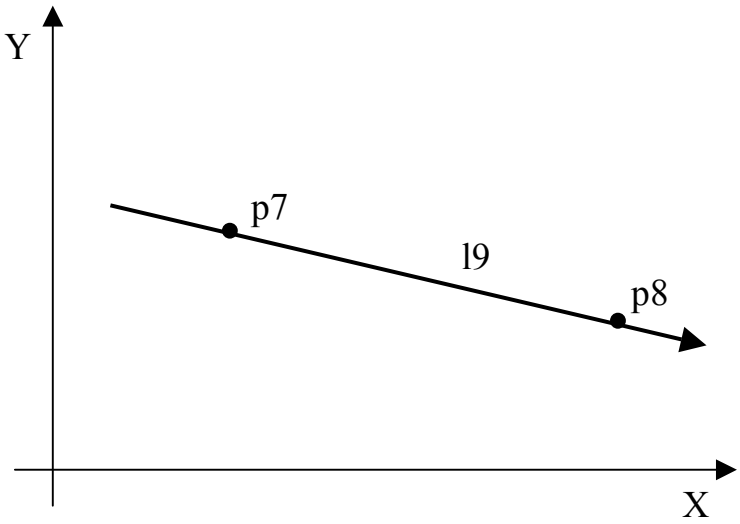


Fig. 5.56.

```
l1 = p1,c1
l2 = p1,-c1
```

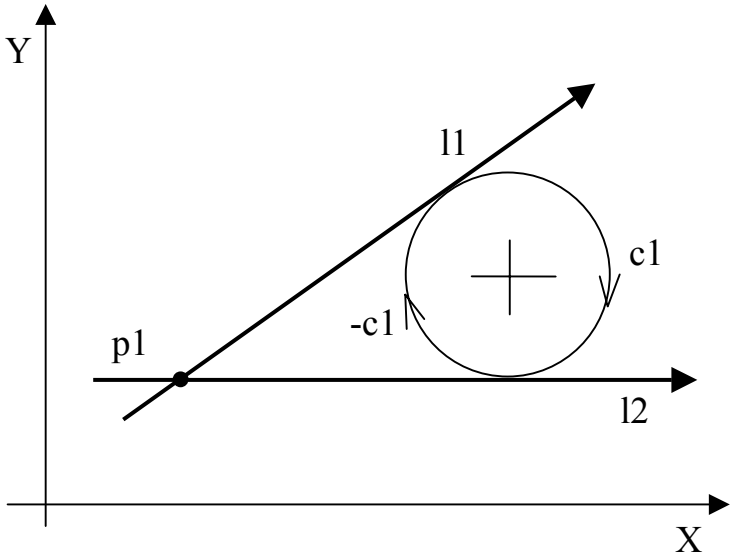


Fig. 5.57.

```
l3 = -c1,c2
```

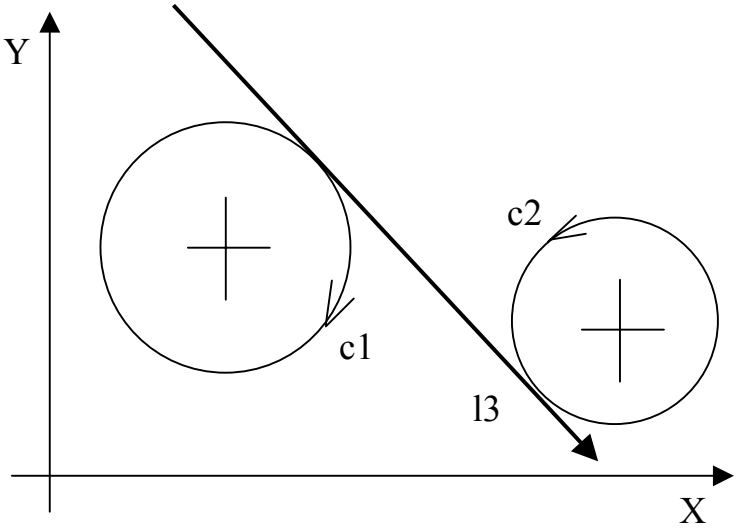


Fig. 5.58.

```
l4 = c1,c2
```

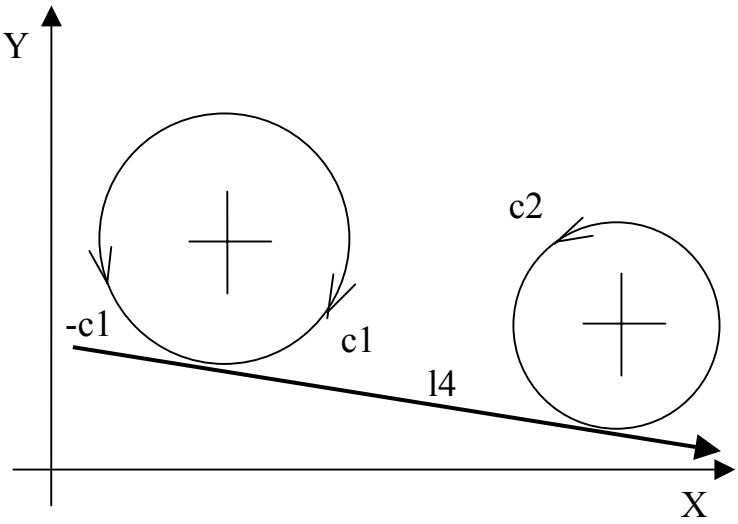


Fig. 5.59.

```
l3 = p1,a30
```

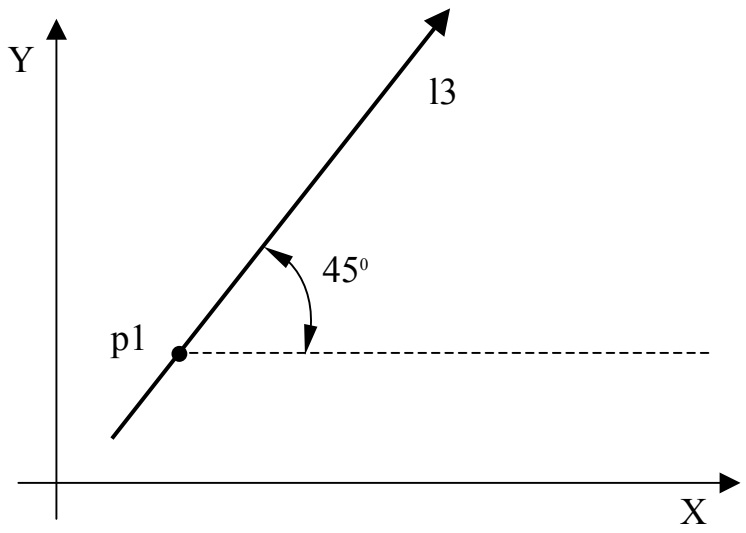


Fig. 5.60.

```
l1 = c1,a50
l2 = -c1,a50
```

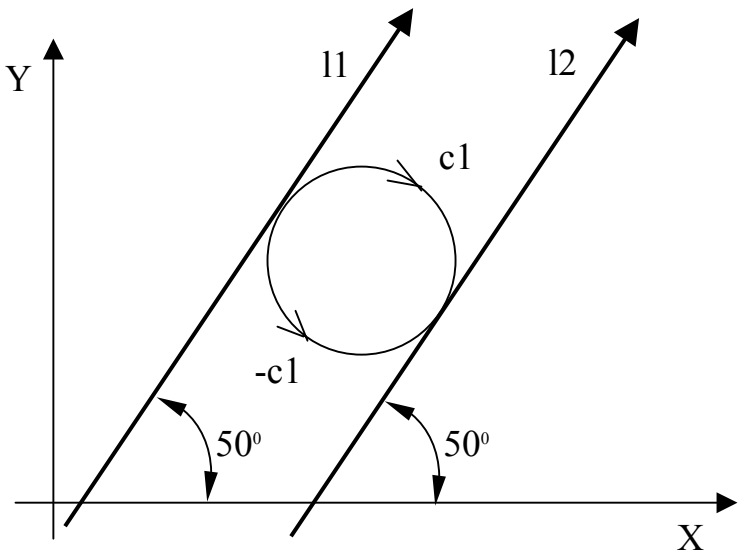


Fig. 5.61.

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

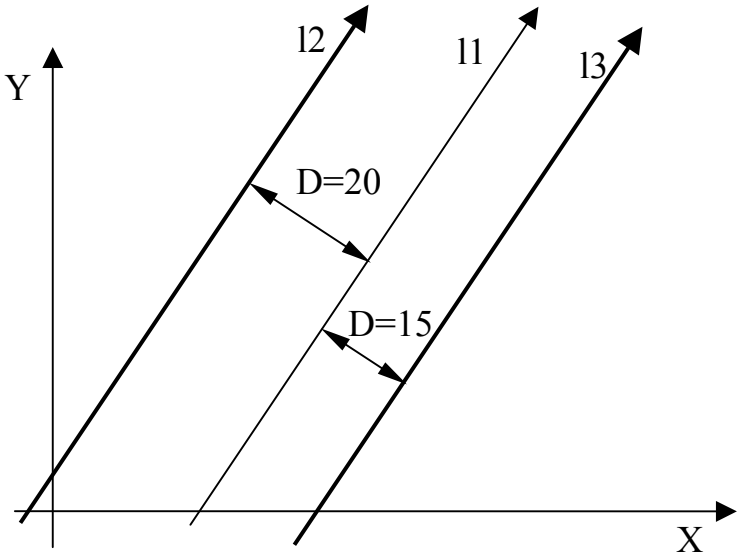
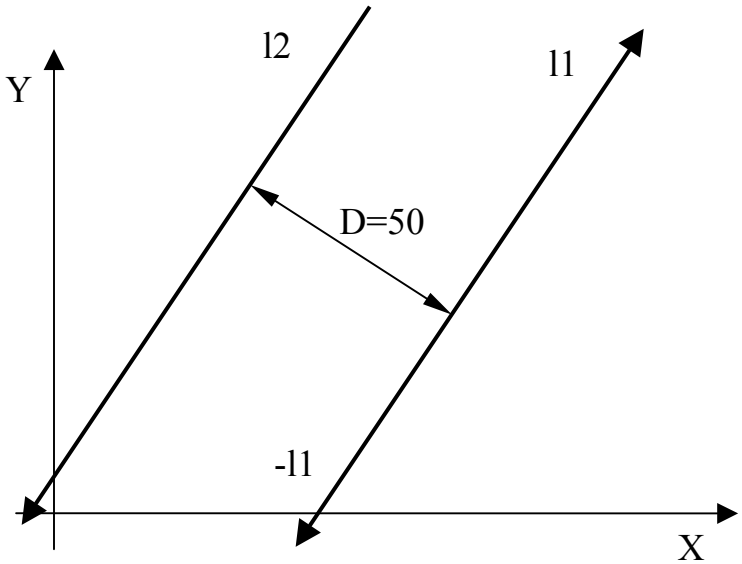


Fig. 5.62.

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

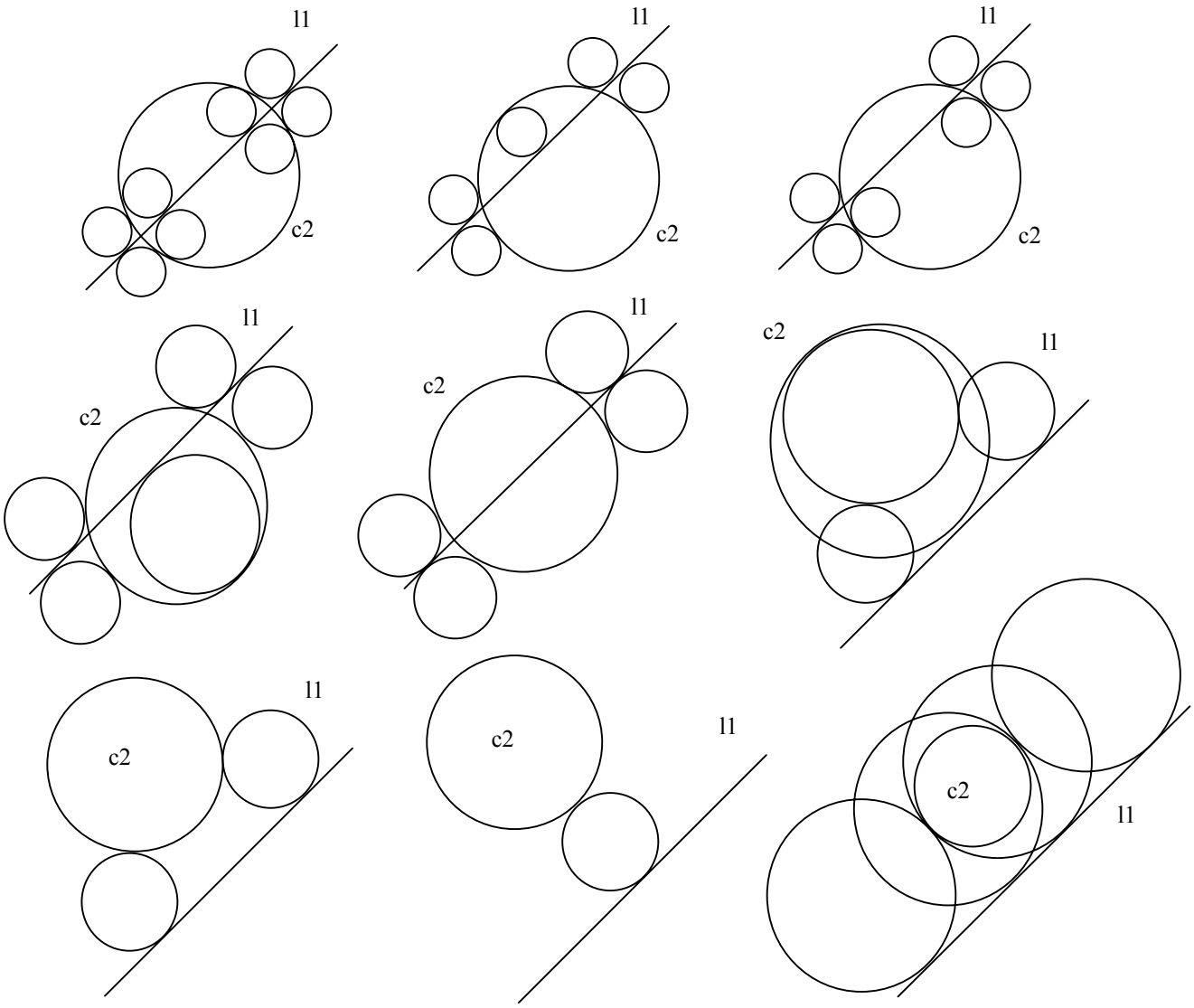


DEFINING CIRCLES

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 straiqht line can have 1 to 8 circles tangent to the straight line and the circle

Fig. 5.63. - Circles Tangent to One Straight Line and One Circle

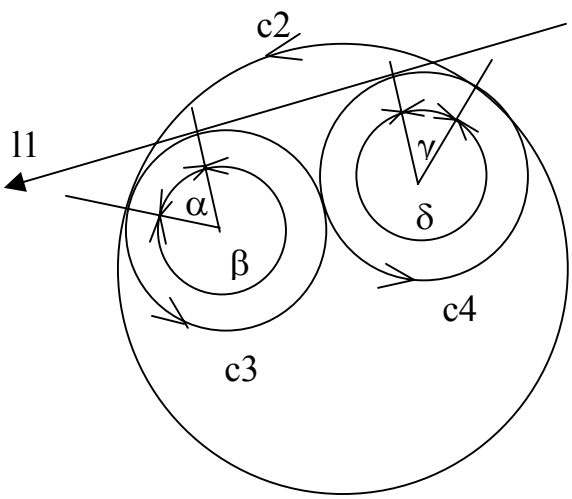


If you account for direction compatibility between a given circle and a line, there are only two possible tangent circles that have the same radius and direction. To distinguish between the two possible circles, consider:

- 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 tanqent circles.

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

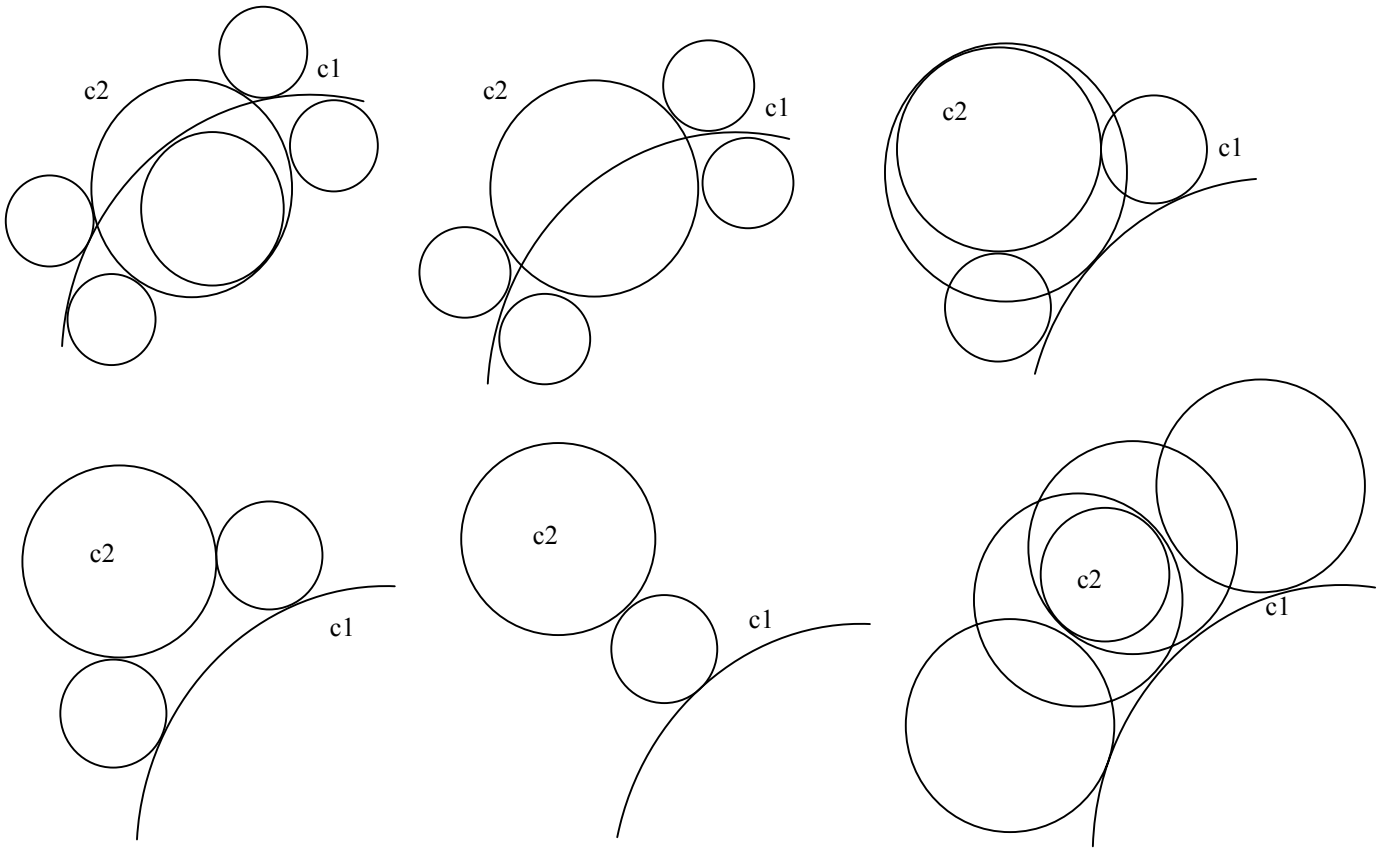
Fig. 5.64. - Tangent Circles with Minor Central Angles



In Fig. 5.64., 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 deiining circle c2 first and straight line l1 second. C4 allows the motion from circle c2 to line l1, and has a minor central angle.

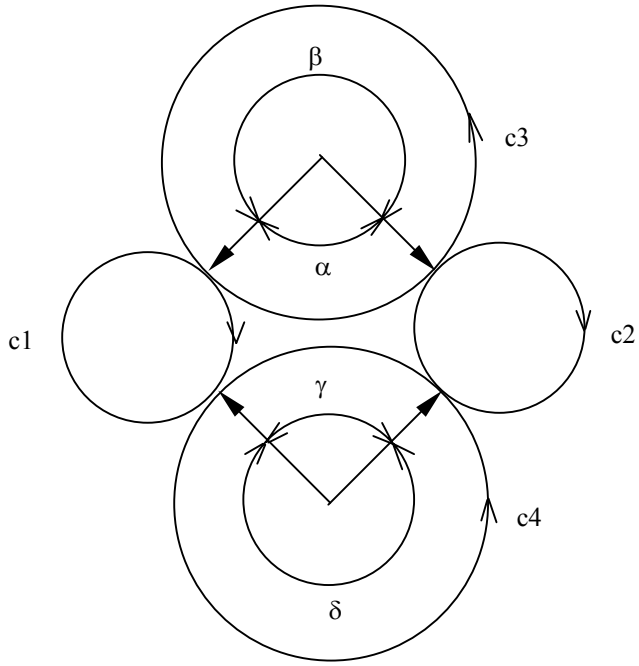
This method also applies to defining a circle that is tangent to two predefined circles. Unless you account for direction compatibility, order of definition and minor central angle, 1 to 8 solutions are possible (Fig. 5.65.).

Fig. 5.65. - 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. 5.66.).

Fig. 5.66. - Circles Tangent to Two Predefined Circles



In Fig. 5.66., 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. 5.67., 5.68.)
cn = [om] I.. J.. r..

Circle with polar centre coordinates and radius (Fig. 5.69.)
cn = [om] m.. a.. r..

Indirect Programming Format

Circle of given radius and tangent to two predefined straight lines (Fig. 5.70.)
cn = [-]lm,lp,r..

Circle tangent to a straight line and predefined circle of given radius (Figs. 5.71., 5.72., 5.73.)
cn = [-]lm,[-]cp,r..
cn = [-]cp,[-]lm,r..

Circle of given radius through a predefined point and tangent to a predefined straight line (Fig. 5.74.)
cn = pm,[-]lp,r..
cn = [-]lp,pm,r..

Circle of given radius tangent to two predefined circles (Figs. 5.75., 5.76.)
cn = [-]cm,[-]cp,r..

Circle of given radius through one predefined point, tangent to one predefined circle (Fig. 5.77.)
cn = pm,[-]cp,r..
cn = [-]cp,pm,r..

Circle of given radius through two predefined points (Fig. 5.78.)
cn = pm,pq,r..

Circle with centre at a predefined point and tangent to a predefined straight line (Fig. 5.79.)
cn = pm,[-]lp

Circle with centre at a predefined point and tangent to a predefined circle (Fig. 5.80.)
cn = pm,[-]cp[,s2]

Circle through three points (Fig. 5.81.)
cn = pm,pq,pr

Circle of given radius with centre at a point (Fig. 5.32.)
cn = pm,r..

Circle concentric to a predefined circle and at a given distance from it (Fig. 5.83.)
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. 5.67.

```
c2 = I50 J100 r-40
```

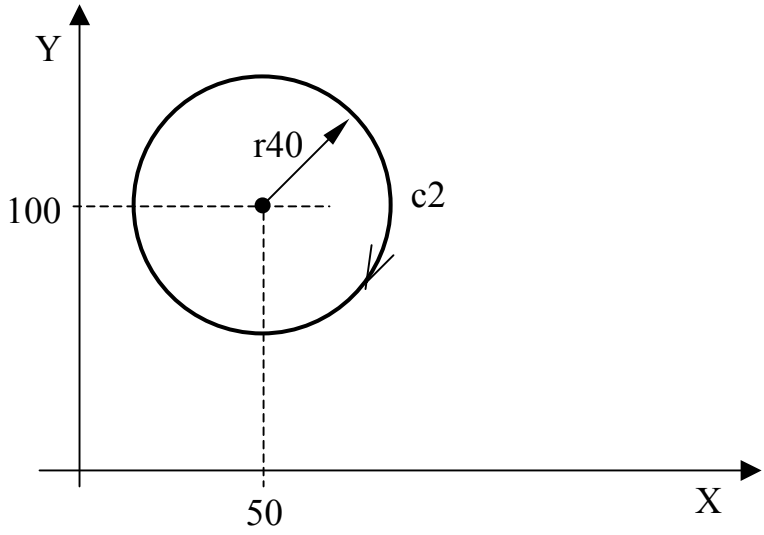


Fig. 5.68.

```
c1 = o1 I20 J20 r-15
```

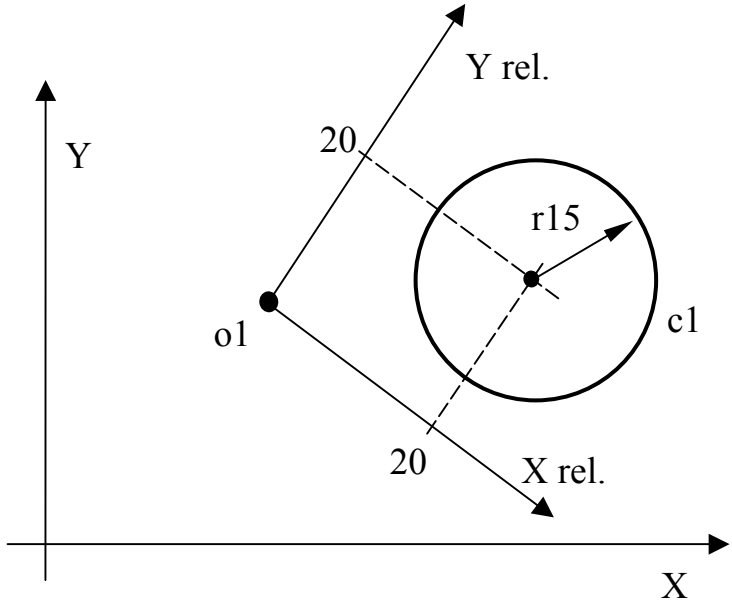


Fig. 5.69.

```
c2 = m70 a30 r15
```

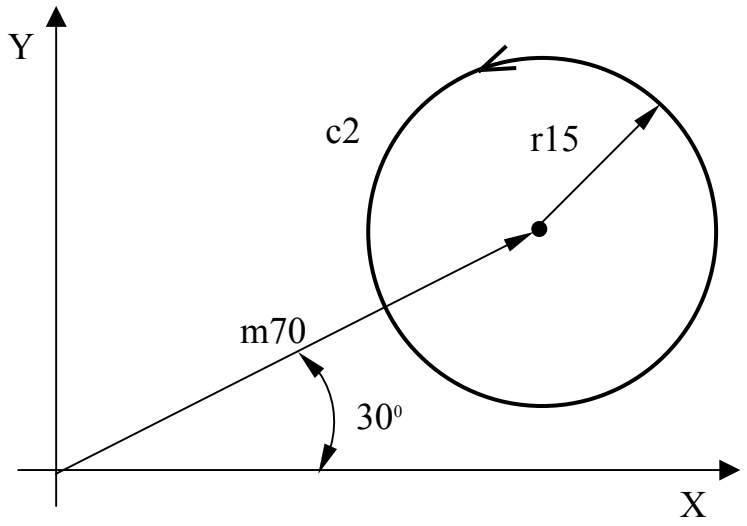


Fig. 5.70.

```
c3 = l1,l2,r-15
```

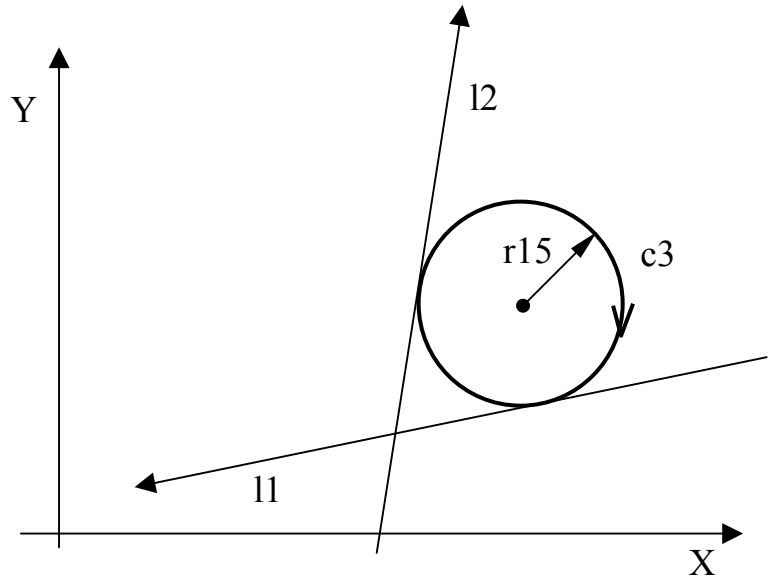


Fig. 5.71.

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

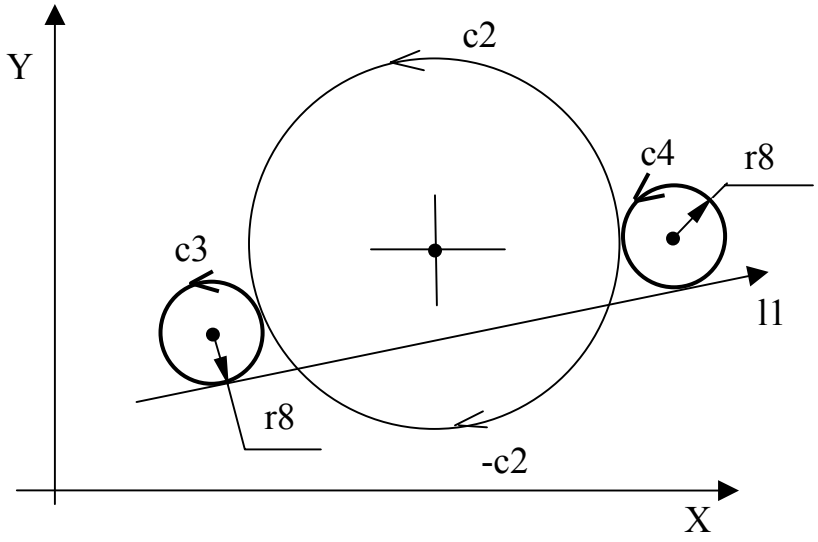


Fig. 5.72.

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

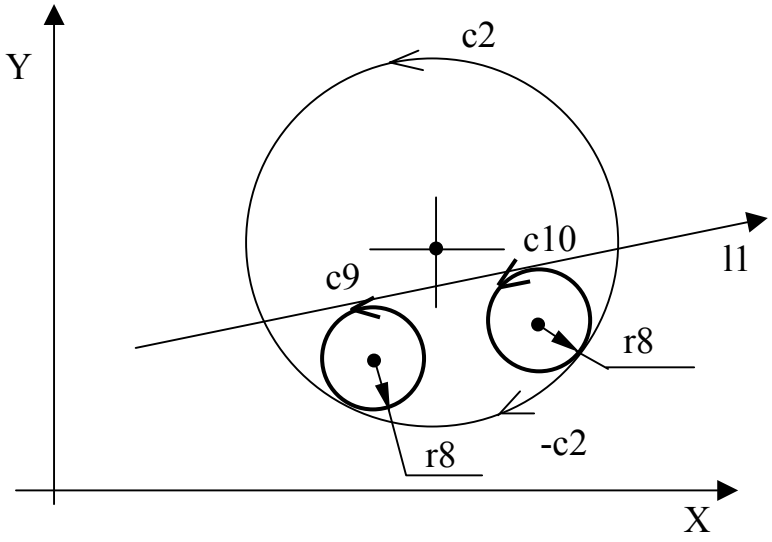


Fig. 5.73.

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

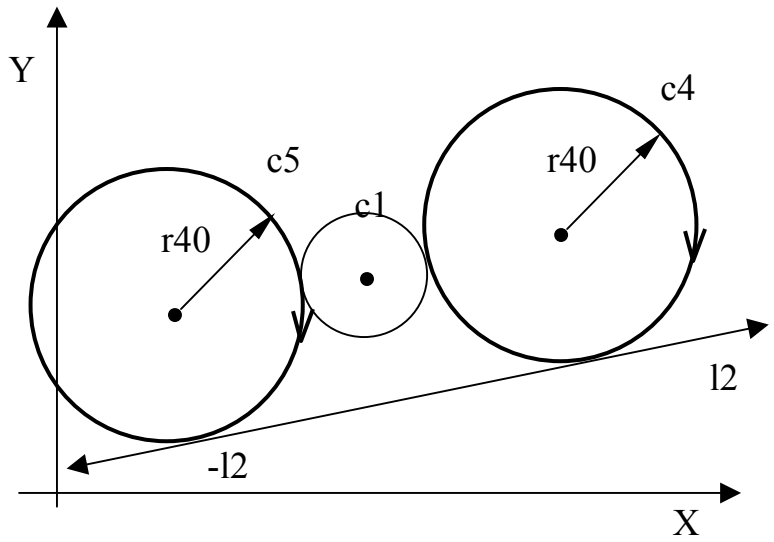


Fig. 5.74.

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

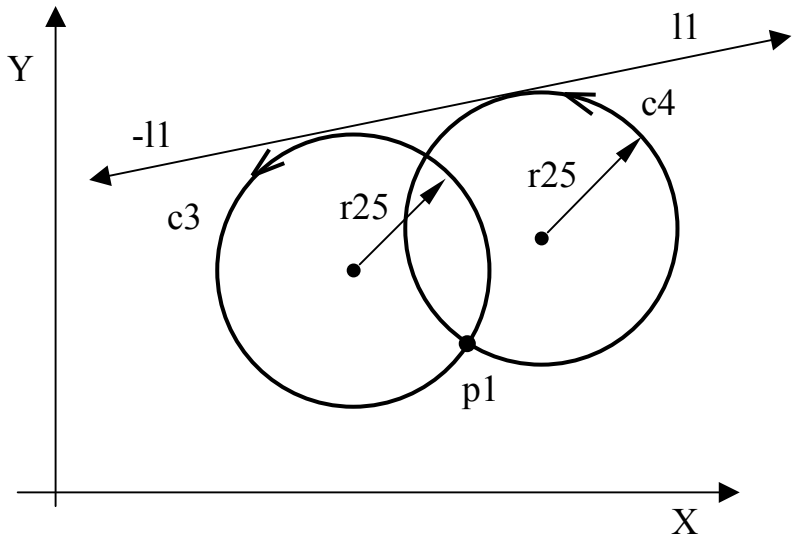


Fig. 5.75.

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

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

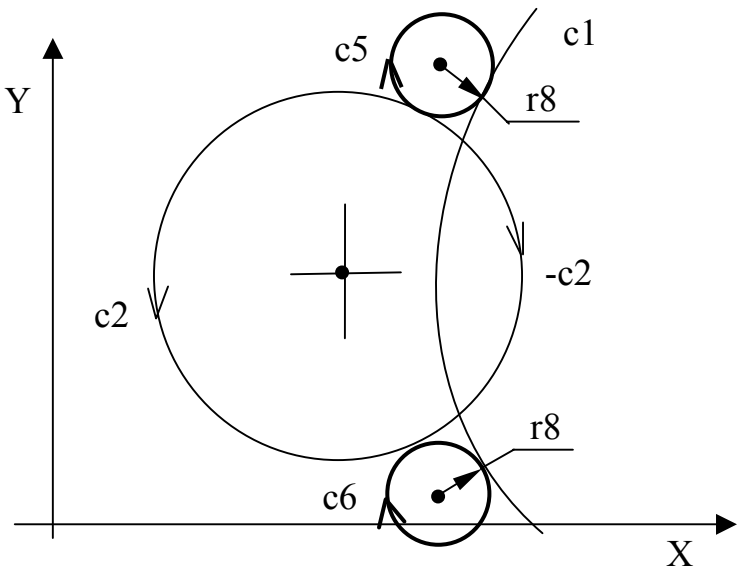


Fig. 5.76.

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

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

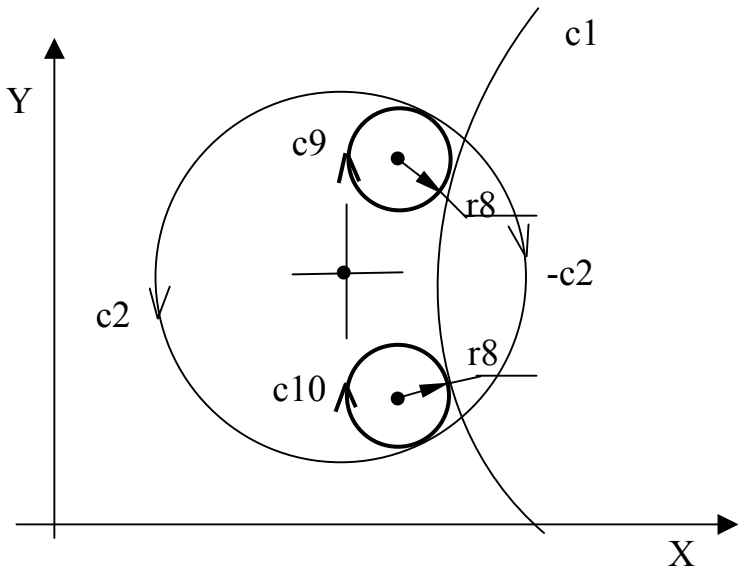


Fig. 5.77.

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

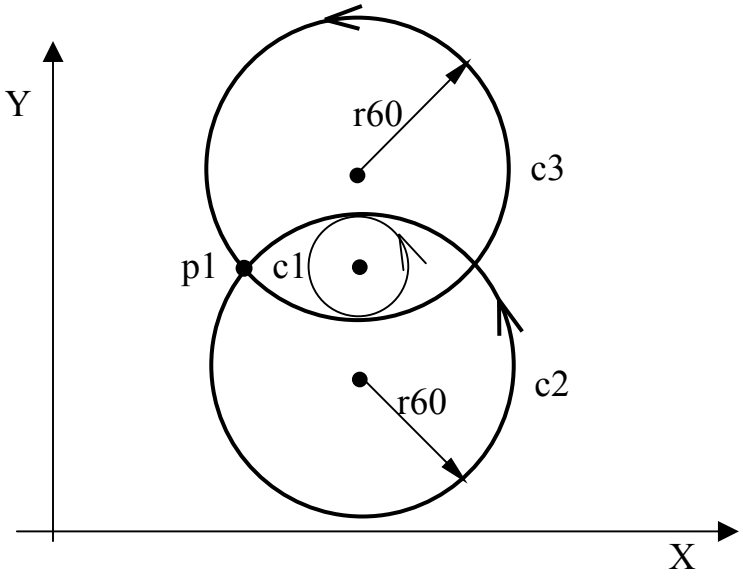


Fig. 5.78.

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

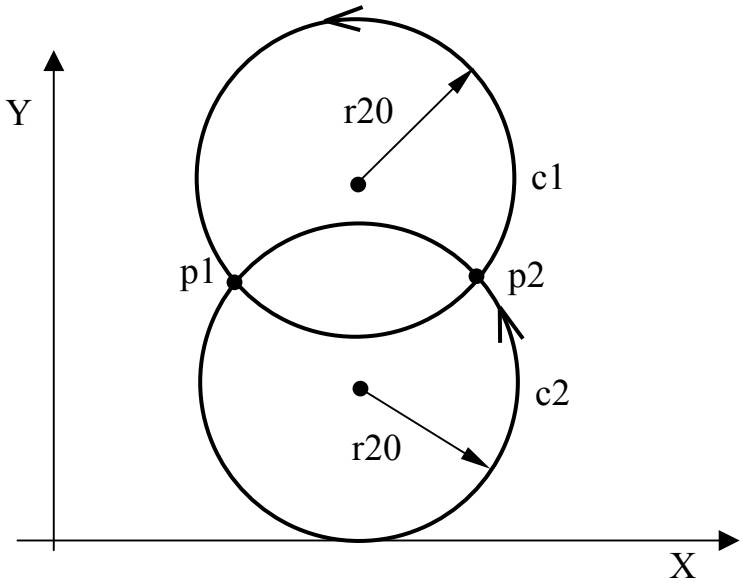


Fig. 5.79.

```
c3 = p1,l2
```

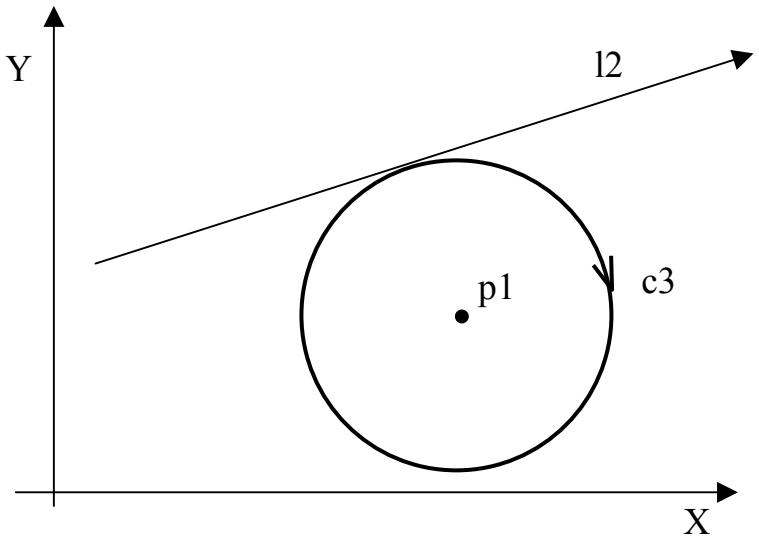


Fig. 5.80.

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

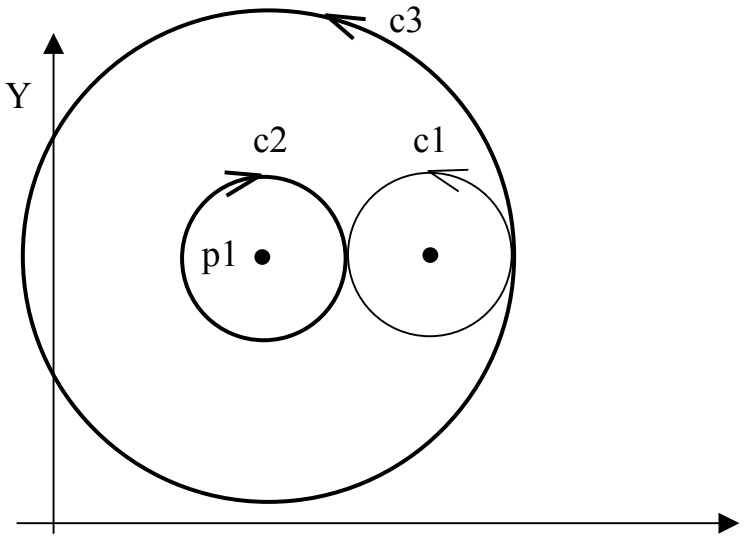
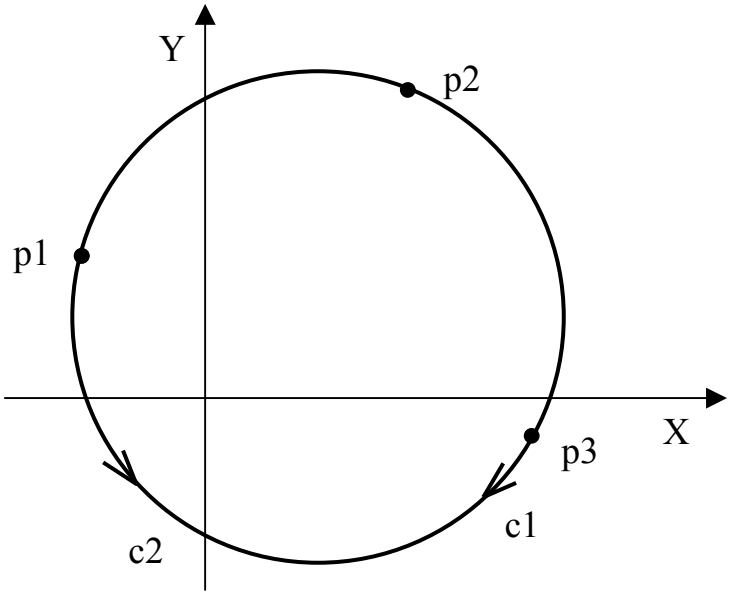


Fig. 5.81.

```
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. 5.82.

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

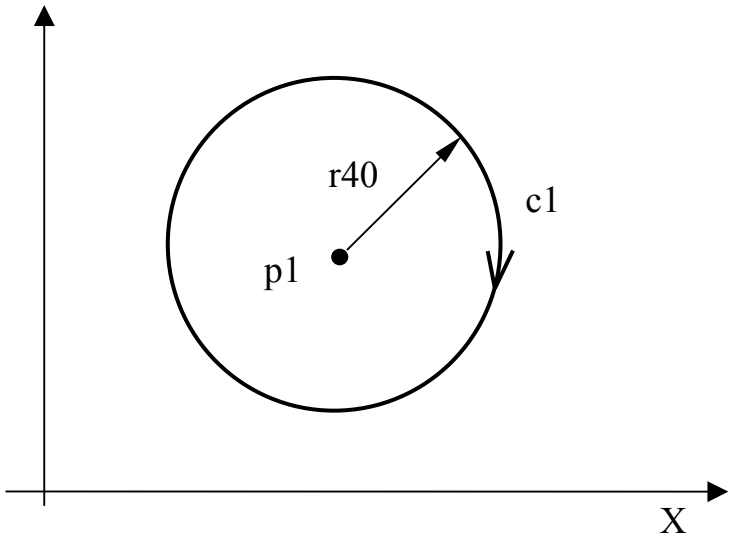
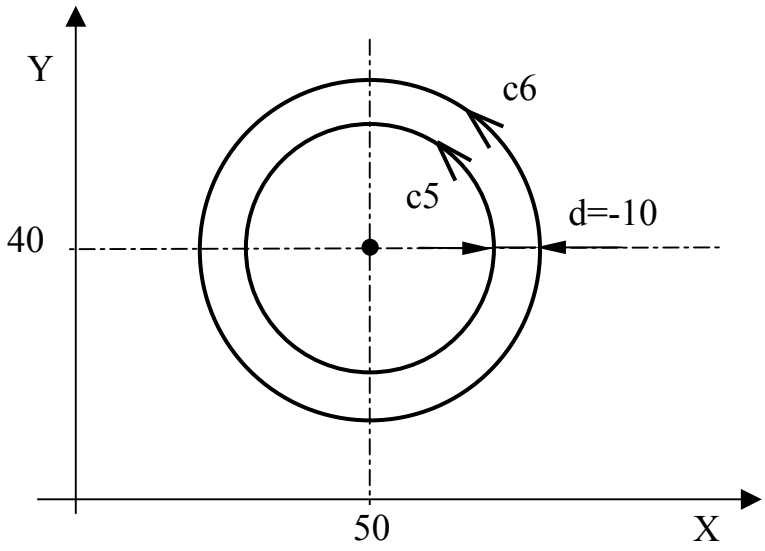


Fig. 5.83.

```
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 instructimns:

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

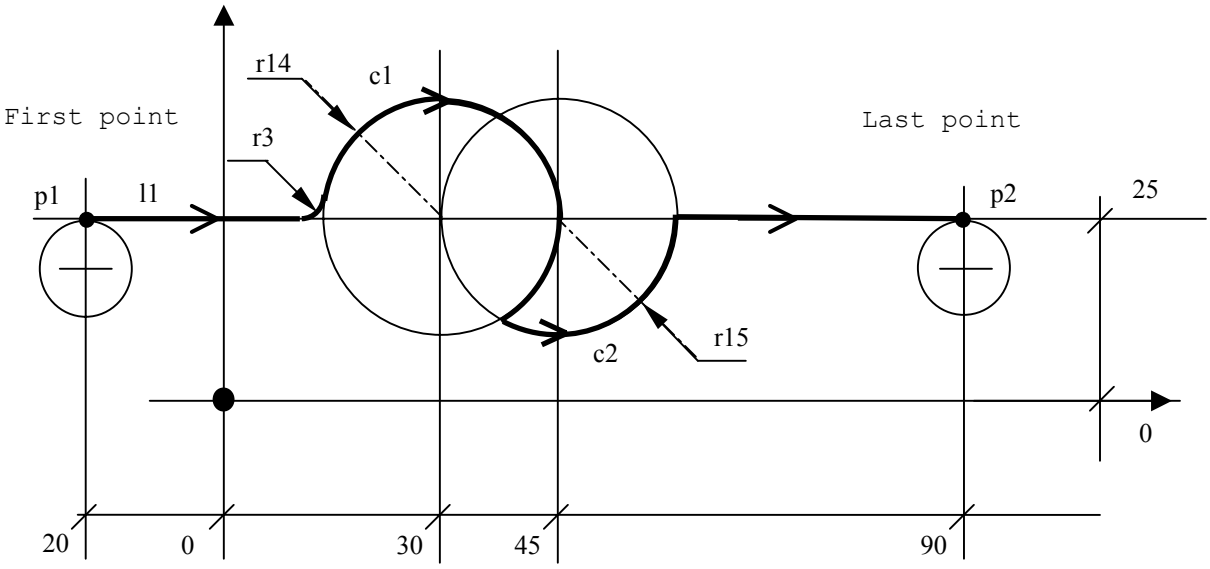
A profile must start with a point (pn) and end with a point different from the first one.

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 proqramming G21 and G41 or G42. Deactivate tool compensation on the last point with functions G20 and G40 (refer to Figure 5.84.).

Fig. 5.84. - Open profile

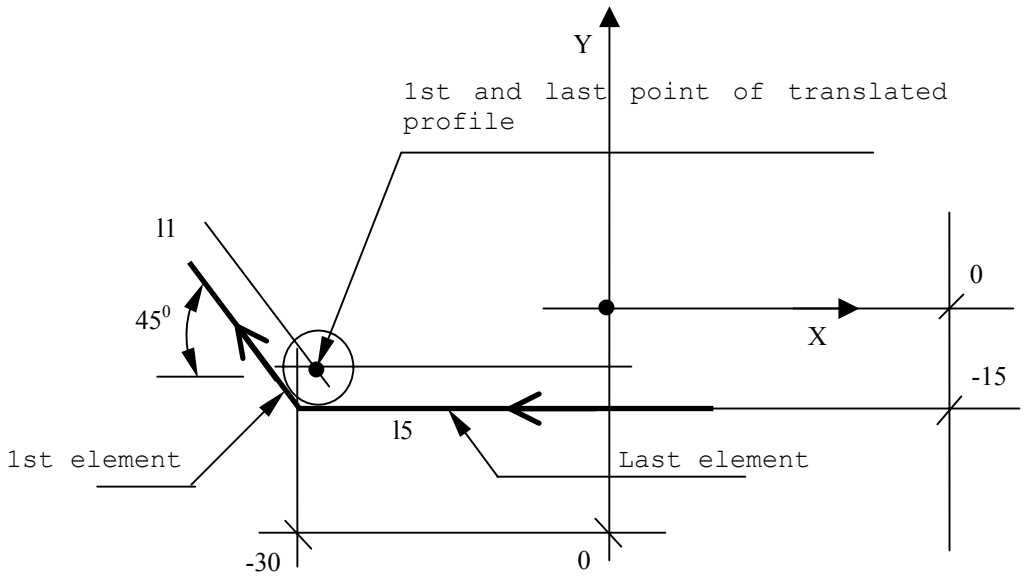



```
..............
l1 = X Y25,a
p1 = X-20 Y25
p2 = X90 Y25
c1 = I30 J25 r-14
c2 = I45 J25 r15
..............
G21 G42 p1                  -first point
l1
r3
c1 s2
c2 s2
l1
G20 G40 p2                  -last point
..............
```

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.

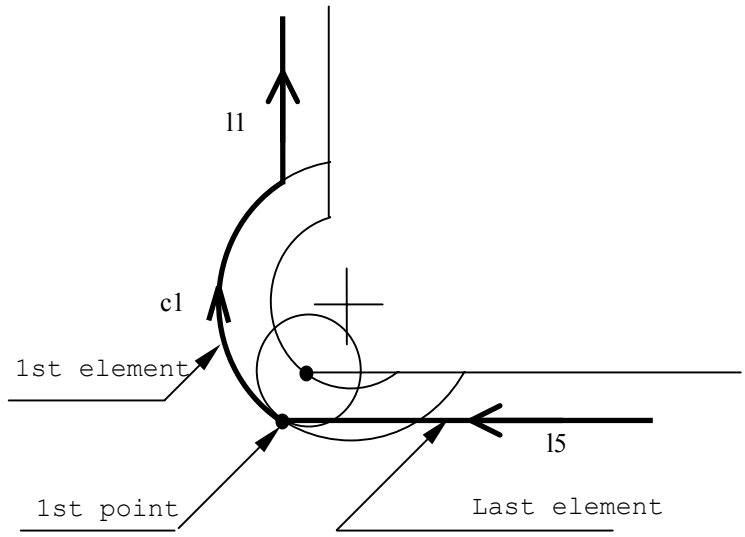
To program a closed profile, you must program the last element first and the first element second (refer to Figures 5.86. and 5.87.).
To activate tool offset, proqram G21 and G41 or G42 in the block containing the last element. To deactivate tool offset, program G20 and G40 at the end of the profile, 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. 5.85. - Closed Profile




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

Fig. 5.86. - Closed profile




```
c1 = I.. J.. r
.............
l1  = X.. Y..,a90
.............
l5 = X.. Y.., 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.

Motion of the Spindle Axis

On any point of the profile it is possible to move the axes not involved in the contouring operation.
To move the axis on the first point of an open profile, program the novement inmediately after the first point. To program a motion on the first point of a dlosed profile, program the motion block between the last and the first element in the profile. For example,

```
..........                G21 G42 l5          -last element
G21 G42 p1                Z-10
Z-10                      l1                  -first element
l1                        ..........
..........                ..........
```

Connecting GTL elements

You can connedt the elements of a profile by using:

- tangency or intersection conditions
- automatic radii

a) Connections with Intersections

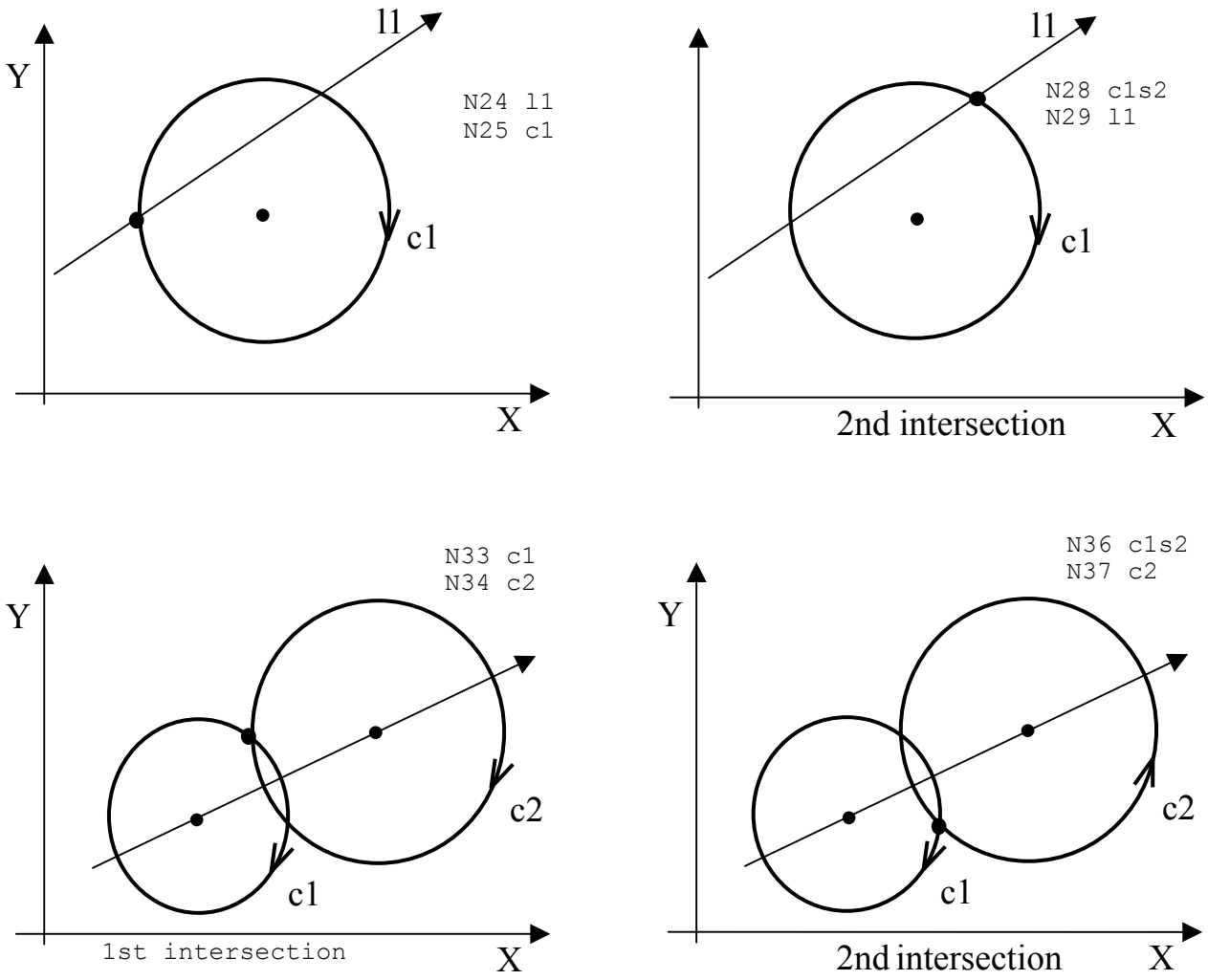
There is only one possible solution for the intersection of two straight lines.

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

The control automatidally 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 5.87.)

Fig. 5.87. - Line-Circle Intersections

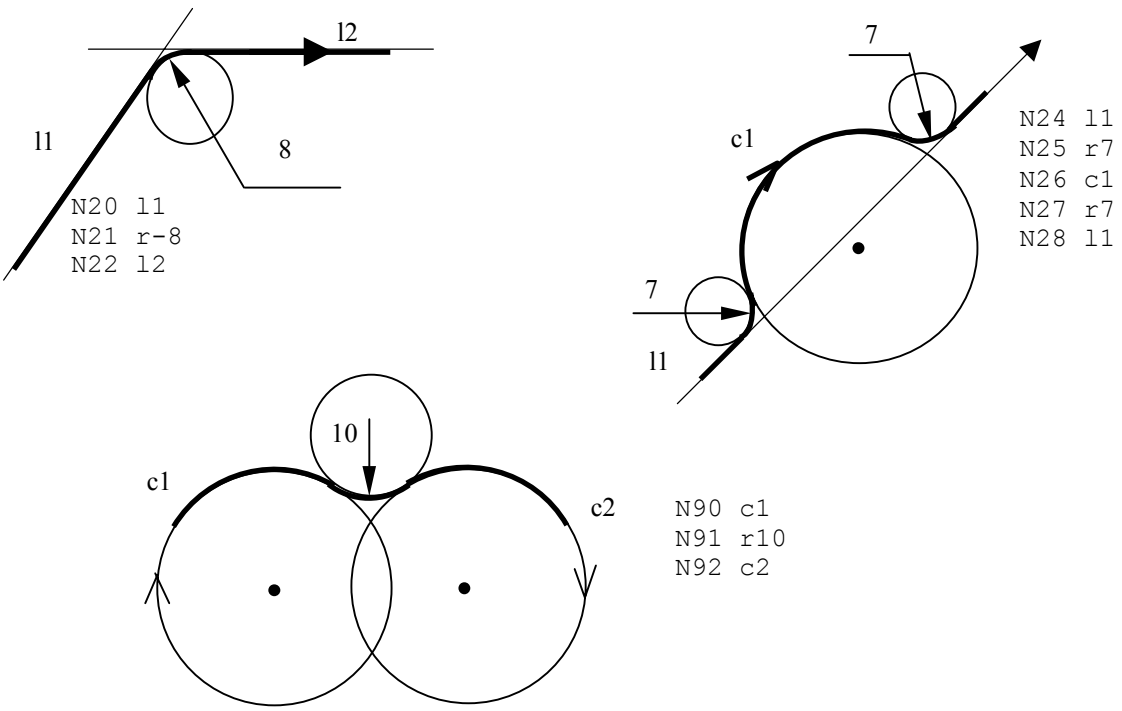


For straight line-circle intersections, the direction of movement of the straight line determines the first and second intersections.
For circle-circle intersections, the first intersectimn is the one on the left of the straight line that connects the centre of the first circle with the centre of the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

b) Connection with automatic radii

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).

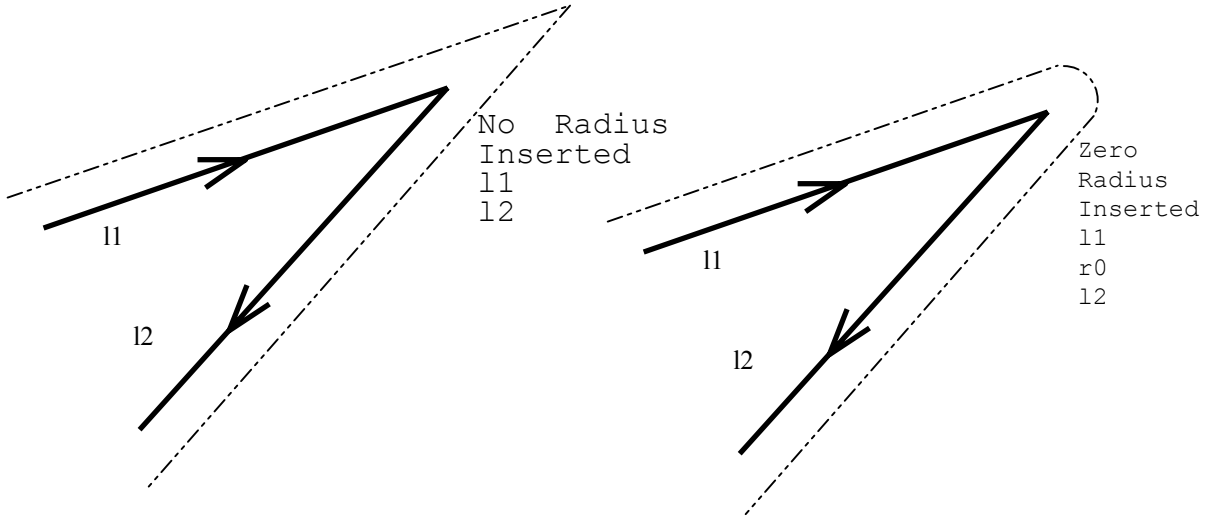
Fig. 5.88. - Connecting Elements with Automatic Radius



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 mr 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 5.89.

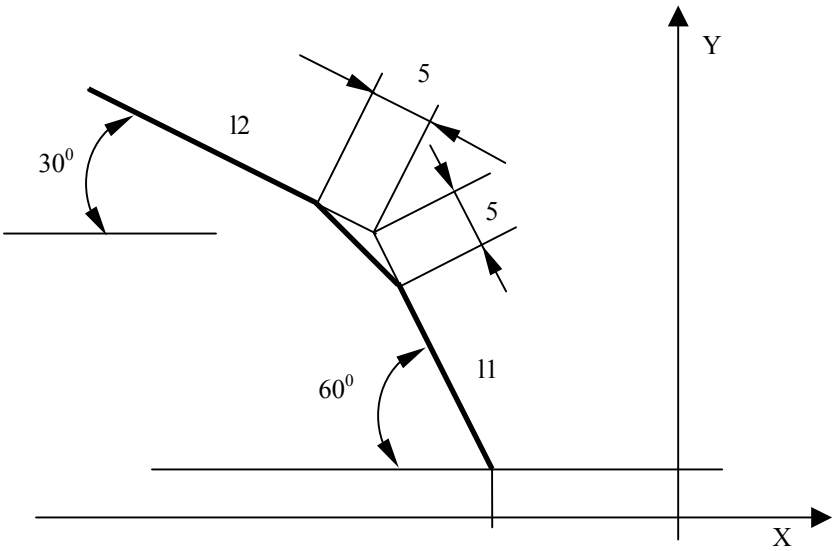
Fig. 5.89. – Zero Radius with Intersections and Tool Compensation



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. 5.90. - Bevel




```
N30  l1
N31  b3
N32  l2
```

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.

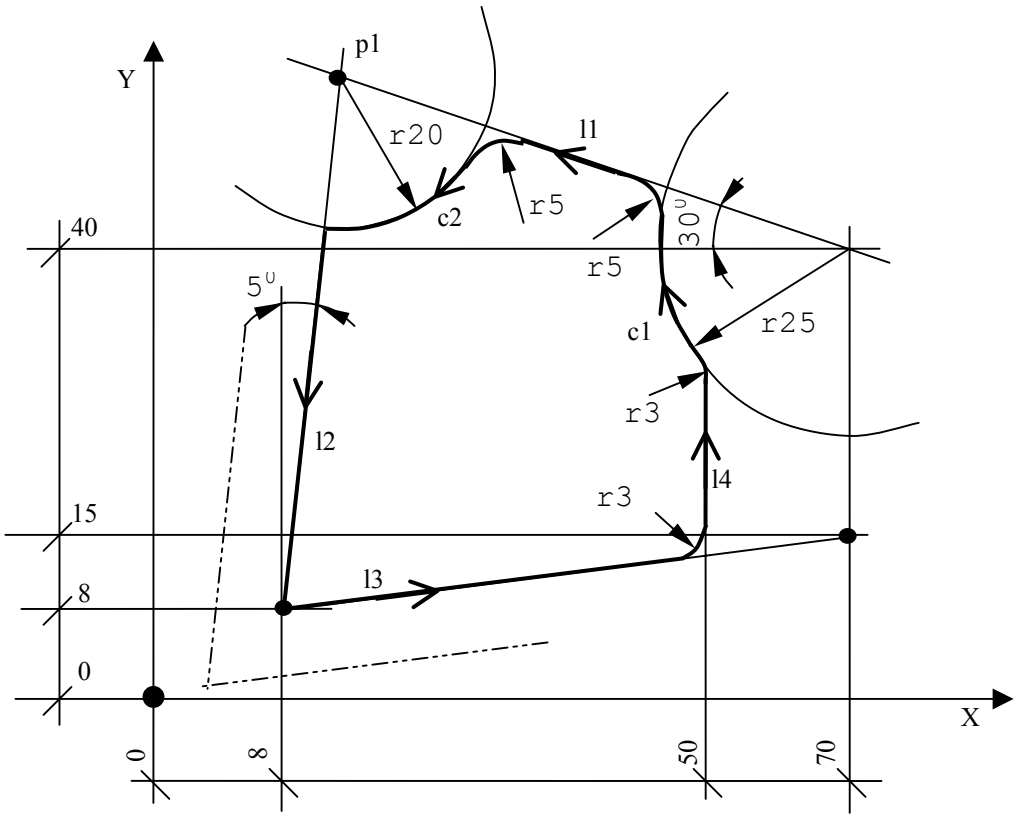
In GTL programming, all movements are performed at feedrate. To proqram a movement at rapid rate, program a high feedrate.

If the interpolation plane is not the one defined by the Y and X axis, you must first define the plane and afterwords store the profile elements. The current abscissa and ordinate are those of the new interpolation plane. For example:

```
N1 (DPI,B,Y)
N2 l1=B70,Y40,a150
N3 l2=B8,Y8,a-95
N4 p1=l1,l2
N5 l3=B8,Y8,B70,Y15
N6 c1=170,J40,r-25
.
.
N12 G21 G42 l2
..............
```

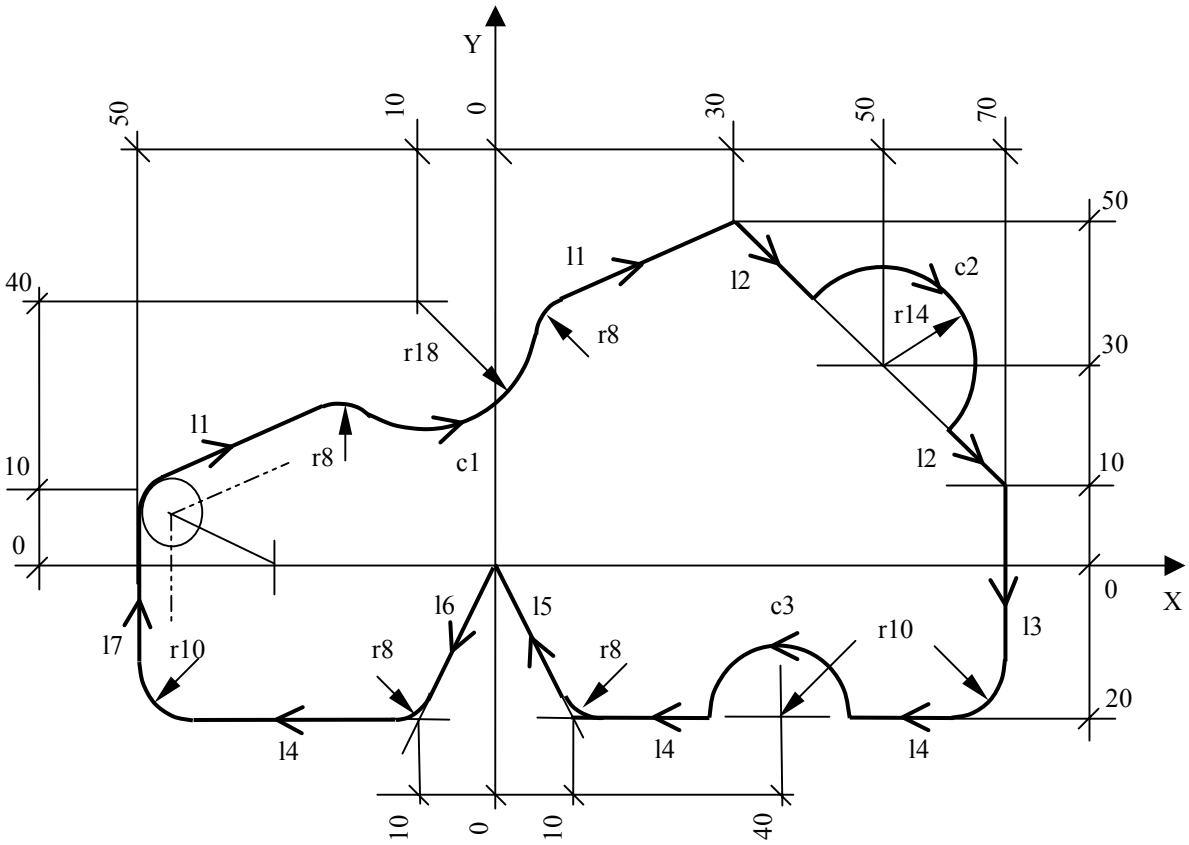
EXAMPLES OF GTL PROGRAMMING

Fig. 5.91. - Example 1



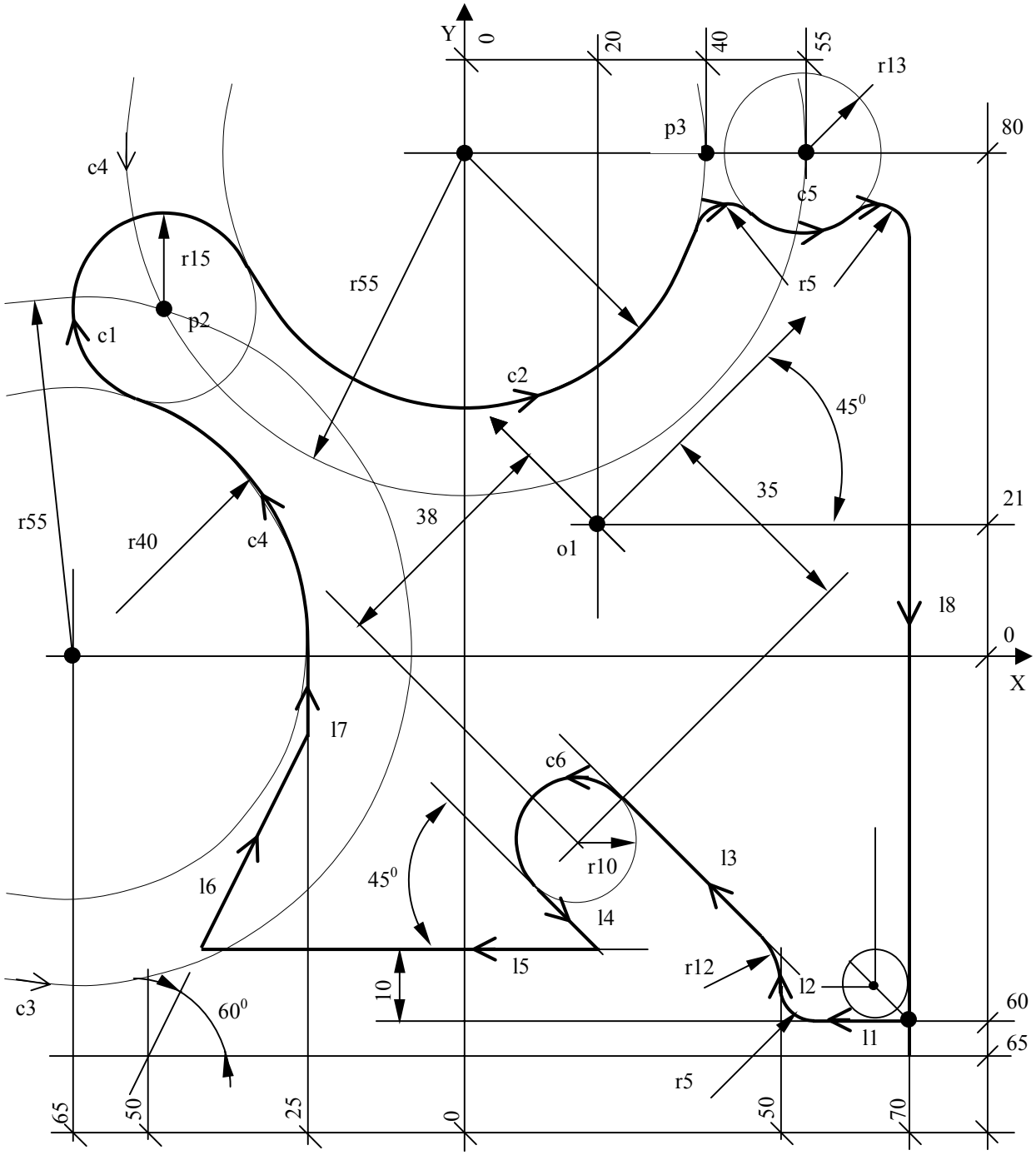

```
N1  (DIS,"EXAMPLE 1)
N2  l1=X70Y40,a150
N3  l2=X8Y8,a-95
N4  p1=l1,l2
N5  l3=X8Y8,X70Y15
N6  l4=X50Y,a90
N7  c1=I70J40 r-25
N8  c2=p1,r-20
N9  F250 S800 T1.1  M6M3M7
N10 GXY
N11 Z-10
N12 G21G42l2
N13 l3
N14 r3
N15 l4
N16 r3
N17 c1
N18 r5
N19 l1
N20 r5
N21 c2 s2
N22 l2
N23 G20 G40 l3
NZ4 GZ2
N25 XY M30
```

Fig. 5.92. - Example 2




```
N1  (DIS,"EXAMPLE 2)
N2  l1=X-50Y10,X30Y50
N3  l2=X30Y50,X70Y10
N4  l3=X70Y0,a-90
N5  l4=X0Y-20,a180
N6  l5=X10Y-20,X0Y0
N7  l6=XOY0.X-10Y-20
N8  l7=X-50Y0.a90
N9  c1=I-10J40r18
N10 c2=I50J30r-14
N11 c3=I40J-20r10
N12 S..M...T3.3M6M..
N13 G0X-30Y0
N14 Z-10
N15 G21G42l7
N16 l1
N17 r-8
N18 c1
N19 r-8
N2O l1
N21 l2
N22 c2s2
N23 l2
N24 l3
N25 r-10
N26 l4
N27 c3s2
N28 l4
NZ9 r-8
N30 l5
N31 l6
N32 r-8
N33 l4
N34 r-10
N35 l7
N36 G20G40l1
N37 GOZO
..........
```

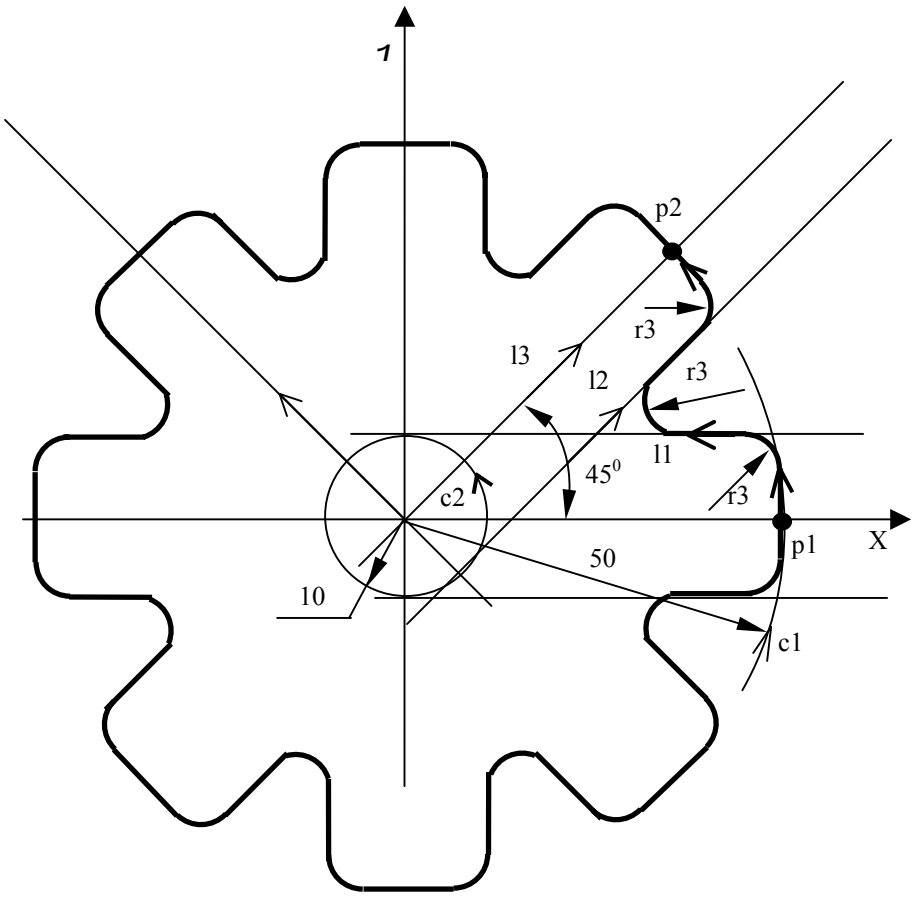
Fig. 5.93. – Example 3



```
N1  (DIS,"EXAMPLE 3")
N2  S..F..T1.1 M6 M..
N3  o1 = X20 Y21 a45
N4  l1 = X0 Y-60a180
N5  l2 = X50 Y0,a90
N6  c6 = o1  I-38 J-35 r10
N7  l3 = c6,a135
N8  l4 = c6,a-45
N9  l5 = X0 Y-50,a180
N10 l6 = X-50 Y-65,a60
N11 l7 = X-25 Y0,a90
N12 c3 = I-65 J0 r55
N13 c4 = I0 J80 r55
N14 p2 = c3,c4
N15 c1 = p2,r-15
N16 p3 = X40 Y80
N17 c2 = c1,p3,r40
N18 c5 = I55 J80 r13
N19 l8 = X70 Y0,a-90
N20 G21 G42 l8
N21 Z-10
H22 l1
N23 r5
N24 l2
N25 r12
N26 l3
N27 c6
N28 l4
N29 l5
N30 l6
N31 l7
N32 r40
N33 c1
N34 c2
N35 r-5
N36 c5
N37 r-5
N38 l8
N39 G20 G40 l1
N40 G Z
N41 X.. Y.. M30
```

Fig. 5.94. - Example 4

Profile repeated 8 times (executed in 2 cuts)




```
N1  (DES,"EXAMPLE 4 WITH ROTATION")
N2  F..S..T2.2 M6..
N3  UOV=2
N4  p1=X50 Y0
N5  c1=I0 J0 r50
N6  c2=I0 J0 r10
N7  l1=c2,a180
N8  l3=X0 Y0,a45
N9  l2=c2,a45
N10 p2=l3,c1,s2
N11 GX60 Y0
N12 Z-20
"START" N13 E25=0
N14 (RPT,8)
N15 (URT,E25)
N16 G21 G42 p1
N17 c1
N18 r3
N19 l1
N20 r-3
N21 l2
N22 r3
N23 c1
N24 G20 G40 p2
N25 E25=E25+45
N26 (ERP)
N27 (URT,0)
"END" N28
N29 UOV=0
N30 (EPP,START,END)
N31 GZ20
N32 XY M30
```

5.16. 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.

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

(*) The number of E parameters (longreal format) is defined during configuration.

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

En = <expression>

The format for an assignment block for version K14 is:

E(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 on formed by arithmetic operators, functions and operands (E parameters or numerical constants).

Arithmetic operators are:

+ (addition)
- (subtraction)
* (multiplication)
/ (division)

Possible functions are:

SIN(A)	calculates 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 of A
NEG(A)	inverts the sign of A
LOG(A)	comon logariphm for version K14
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)	chanqes 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 E2 the content of byte 396 from the K buffer
E8=SYVAR1	assigns to E8 the value of variable SYVAR 1.

Example for version K14:

Nш1 E3=17	assigns the value 17 to parametr E3
Nш2 E(E3)=10	assigns the value 10 to parametr E17

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 utilization of E parameters is summarized in table 5.9.

Table 5.9. - Utilization of E parameters

Parameters (Format)	Data (geometric-technical)	Programming examples
E0..9 (BY)	G functions	GE1
	M functions	ME3
	RPT code	(RPT,E9)
E10..E19 (IN)	Absolute origin No.	(UAO,E10)
		(UOT,E11,X..,Y..)
	S functions	SE15
E20..E24 (LI)		
E25..E29 (RE)	F functions	FE27
	URT code	(URT,E25)
	SCF code	(SCF,E26)
	Indexed axes	PE29
E30...(*) (LR)	A B C X Y Z U W	
	axes coordinates	XE32
	R coordinate	RE33
	I J K miscellaneous operators	KE34
	u v w compensation factors	E35
	Global system variables:	
	TMR	TMR=E38
	UOV	UOV=E40

5.17. 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 then for:

- modifying the reference system of the axes (section 5.17.1.);
- modifying the sequence of program execution (section 5.17.2.);
- performing miscellaneous commands (section 5.17.3.);
- performing I/O commands (section 5.17.4.);
- monitoring tool life (section 5.17.5.);
- managing the graphic display (section 5.17.6.);
- managing tool offsets (section 5.17.7.).

5.17.1. 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:

UAO - use absolute origins
UOT - use temporary origins
UIO - use incremental origins
MIR - mirror machining
URT - rotation of the plane
SCF - scale factor
RQO - requalify origins

USE ABSOLUTE ORIGINS - UAO

This command allows you to activate the absolute origins previously defined with ORA.
The allowable format is:

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

where:

n defines the number of the permanent origin to be selected. It can be either a numerical constant or an E parameter of the integer type (E10 to E19).

VAR i is a character that represents the name of the axis for which the origin n must be specified.
For undeclared axes, the current origin stays in effect.
If no axis name is specified, the origin n is activated for all those axes in which the n origin has been declared.

Example:

```
(UAO,1)      -activate absolute origin 1
..........   -program referred to origin 1 for all axes
(UAO,2,X,Y)  -activate absolute origin 2 for axes X and Y
(UAO,3,Z)    -activate origin 3 for axis Z
..........   -program referred to origin 2 for axes X and Y
..........    to origin 3 for axis Z, to origin 1 for all
..........    the other axes
(UAO,0)       -reactivate absolute origin zero for all axes
```

Notes:

- At power-on and after a reset, the control automatically establishes absolute origin zero for all the axes.
- You can specify as many as 3 axis names. An axis can only be declared once in each UAO command.
- To declare a different origin for each axis, program as many UAO blocks as the desired origins.
- If the selected origin had been stored in the file in the alternative measuring unit, i.e. preceded by a "-" sign, the system automatically converts its value dividing or multiplying it by the conversion factor.

USE TEMPORARY ORIGINS - UOT

Temporarily shifts the absolute origin by a programmed distance.
The allowable format is:

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

where:

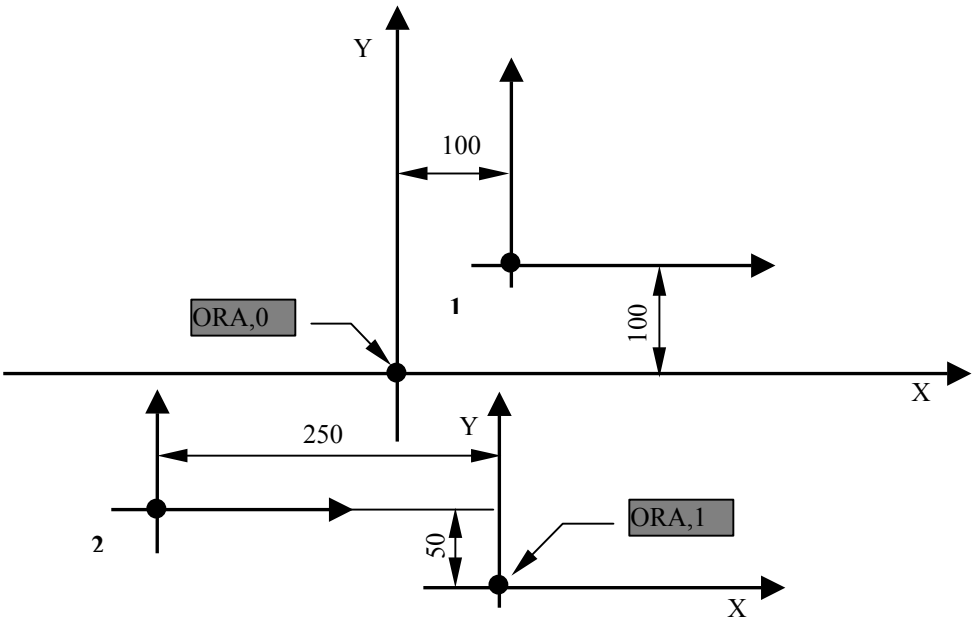
n defines the number of the absolute origin you want to temporarily modify

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. 5.95.):

```
(UAO,O)            -activates absolute origin 0
..........         -program referred to absolute origin 0
   ..........       for all axes
(UOT,O,X100,Y100)  -applies a temporary origin to origin 0,
                    with X100 and Y100 offset (Point 1)
..........         -this portion of the program uses the
..........          temporary origin
(UOT,1,X-250,Y50)  -applies a temporary origin to origin 1,
                    with X-250 and Y50 offset (Point 2)
..........         -this portion of the program uses
.........           X-250 Y50 temporary origin
(UAO,0)            -reactivates absolute origin 0 for all
                    axes
```

Fig. 5.95.



Notes:

- With the UOT command you must declare at least one axis and as many as 3 axes. An axis only be declared once in each UOT command.

- A temporary origin remains in effect until you redefine it with a new UOT or you restablish the absolute origin zith either UOA or RESET.

- You must program the dimension in the UOT command with the current measuring unit (G70/G71). If set, the control will apply the scale factor to the temporary origin.

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:

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

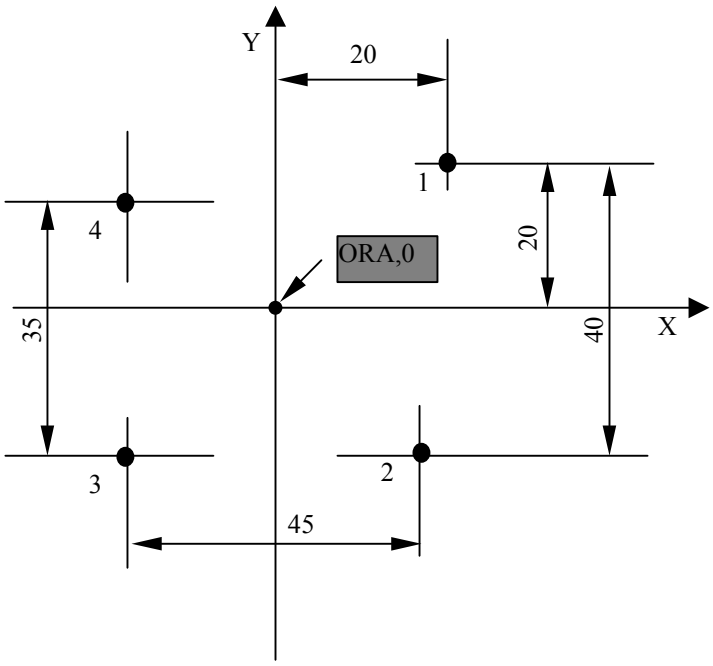
where:

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 oriqin stays in effect.

Example (Fig. 5.96.):

```
...........
N65   (UIO,X20,Y20)       -point 1
...........
N121  (UIO,Y-40)          -point 2
...........
N180  (UIO,X-45)          -point 3
...........
N230  (UIO,Y35)           -point 4
...........
N300  (UAO,0)
```

Fig. 5.96. - Using UIO command



Notes:

- With the UIO command you must declare at least one axis and as many as 3 axes. An axis can only be declared once in each UIO command.

- An incremental origin remains in effect until you redefine it with a new UIO or you restablish the absolute origin with either UOA or RESET.

- You must program the dimension in the UIO command with the current measuring unit (G70/G71). If set, the control will apply the scale factor to the temporary origin.

MIRROR MACHINING - MIR

The MIR command reverses (mirrors) the programmed direction of motion for the axes specifed in the command.
The allowable format is:

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

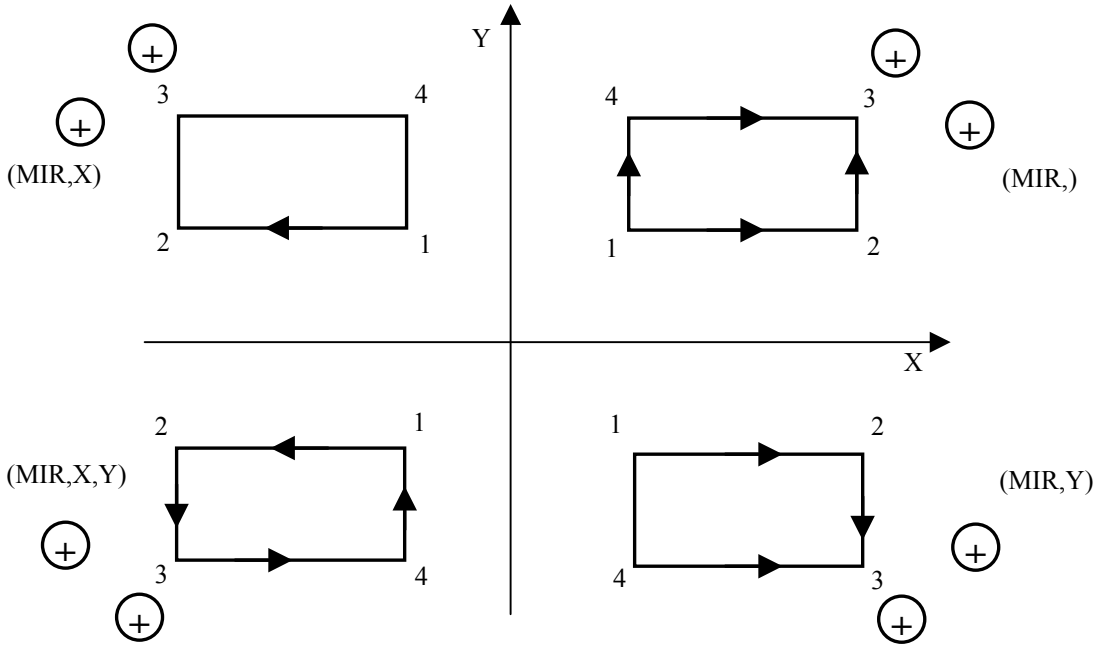
where:

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

Example (Fig. 5.97.):

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

Fig. 5.97. - Use of the MIR command



Notes:

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

- For the undeclared axis the preceding MIR command remains in effect.

- You can declare as many as 3 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: URT first, MIR second.

ROTATION OF THE PLANE - URT

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

(URT,Operand)

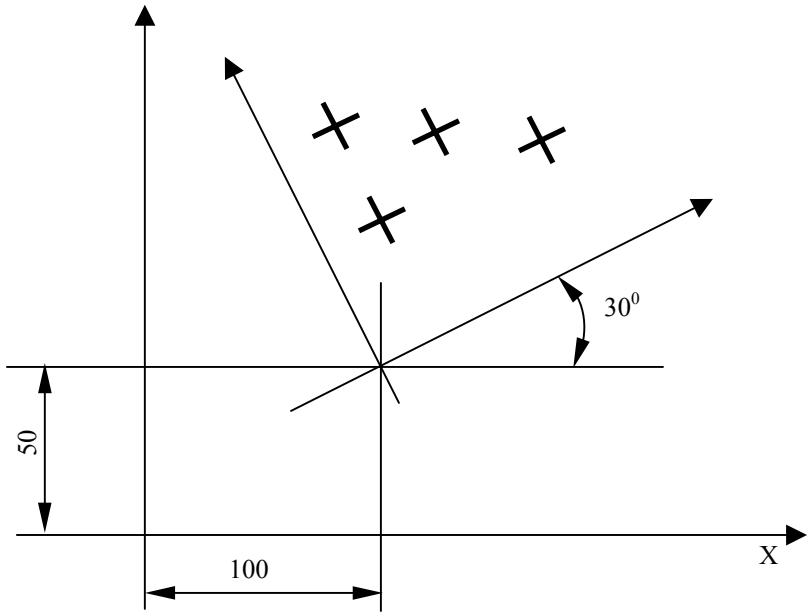
where:

OPERAND the value of the rotation angle in degrees (with decimals). It can be positive (CCW) or negative (CW) and 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. 5.98.):

```
(UOT,0,X100,Y50)
(URT,30)
..........
..........
..........
(UA0,0)
(URT,0)
```

Fig. 5.98. – 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: URT first, MIR second.

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 SCF 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 3 axis names.

REAPPLY ORIGIN - RQO

This command allows you to apply an incremental shift of the absolute origin for fll specified axes.
The allowable format is:

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

where:

n defines the number of the absolute origin you want to modify.
It can be a number from 0 to 99, depending on the number of records you specified when creating the file of origins.
You can also specify it with an integer E parameter (from E10 to E19).

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 current absolute origin for that axis.

Example:

(RQO,3,XE31) -modifies origin 3 of the X axis by applying the value stored in E31

Notes:

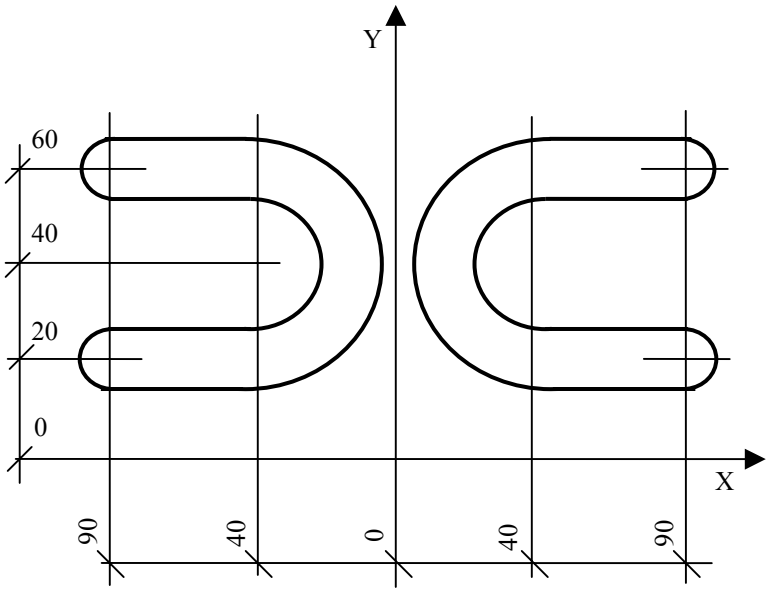
- With the RQO command you must declare at least one axis and as many as 3 axes. An axis can only be declared once in each RQO command.

- The new value of the absolute origin is premanently stored in the file of origins. If you modify the current absolute origin, the new value will also be stored in the user memory.

- In the file of origins, the requalification value is applied with the measuring unit in which the origin is recorded.
  However, you must specify the requalification value in the characterizated measuring unit. The control will apply no scale factor.

Examples of rotation and mirror machining are shown in Figures 5.99. to 5.102.

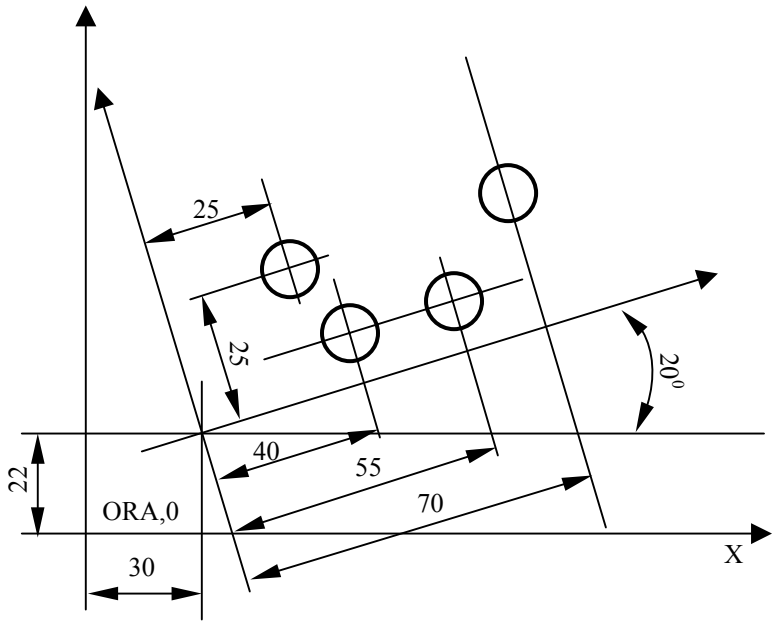
Fig. 5.99. - Example of MIRROR machining




```
N199 (DIS,"MILLING CUTTER D=16")
N200 S1500 T8.8 M6 M3
N201  (RPT,2)
N202 G X90 Y20
N203 Z2
N204 G1  Z-8 F150
N205 X40 F200
N206 G2 Y60 I40 J40
N207 G1  X90
N208 G Z-100
N209 (MIR,X)
N210 (ERP)
N211 (MIR)
N212 G Z50
...............
```

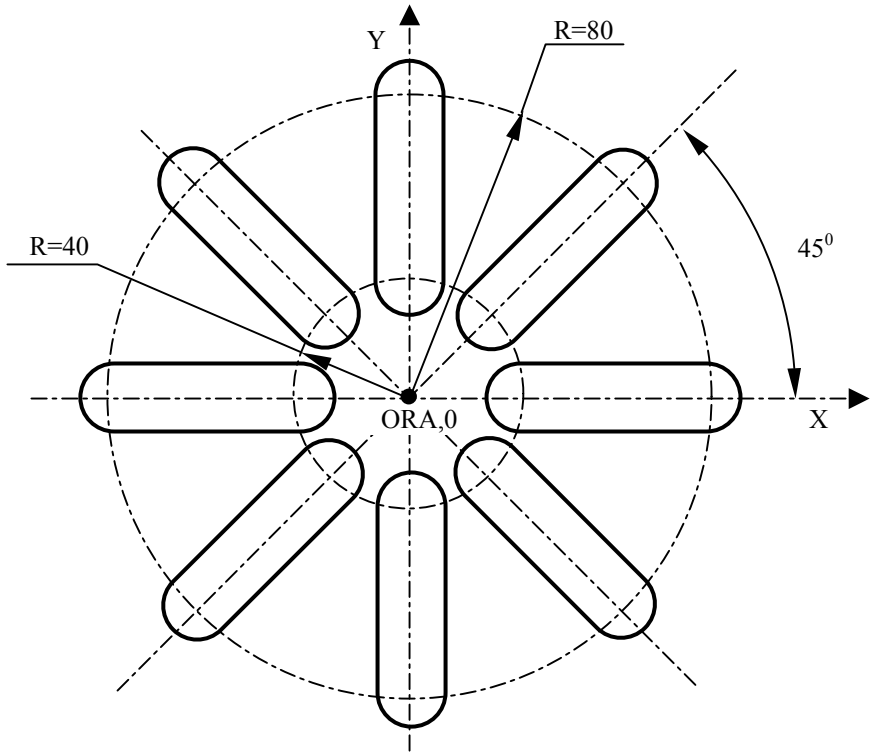
RPT and ERP are dealt with in section 5.17.2.

Fig. 5.100. - Use of the- URT command



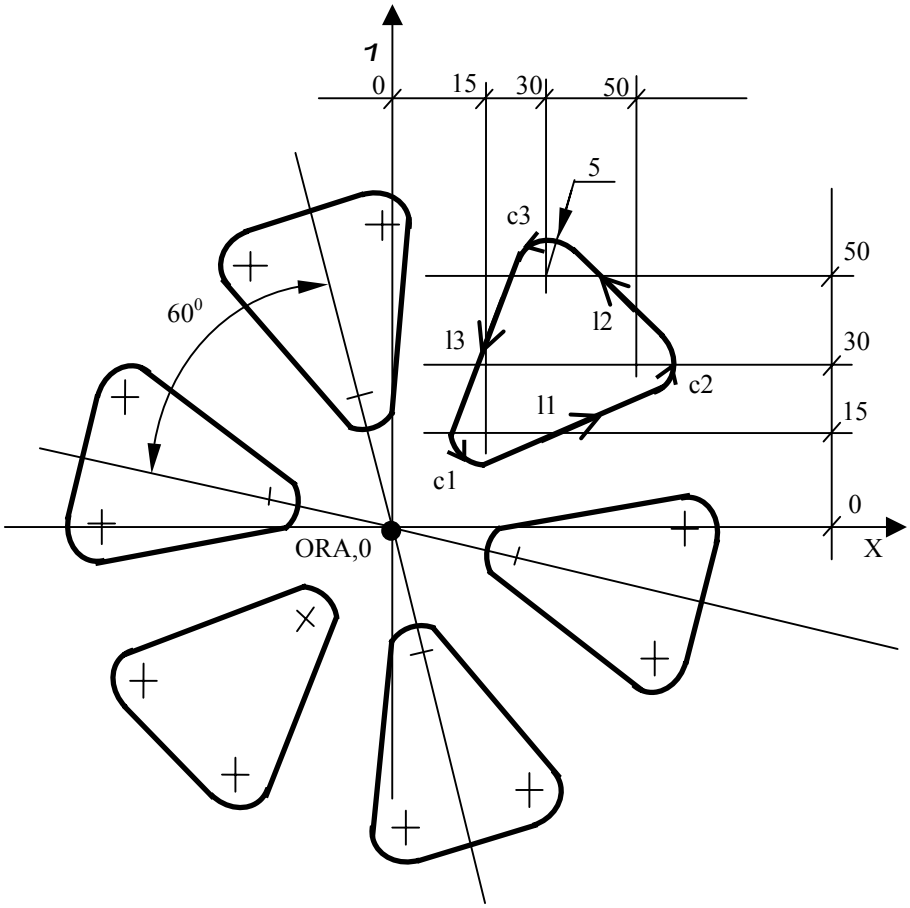

```
N99  (DIS,"DRILLING D=6")
N100 S2000 F200 T3.3 M6
N101 (UOT,0,X30,Y22)
N102 (URT,20)
N103 G81  R3 Z-25 M3
N104 X25 Y25
N105 X40 Y10
N106 X55
N107 X70 Y25
N108 G80 Z20
N109 (UAO,0)
N110 (URT,0)
N111 S1000 T4.4 M6
N112 .........
```

Fig. 5.101. - Use of URT associated to RPT for parametric programming




```
N 148 (DIS,"...")
N149 S1500 T5.5 M6 M3
N150 E25=0
N151 (RPT,8)
N152 (URT,E25)
N153 G X40 Y
N154 Z2
N155 G29 G1  Z-6 F150
N156 X80 F200
N157 Z-12 F150
N158 X40 F200
N159 G Z20
N160 E25=E25+45
N161 (ERP)
N162 (URT,0)
................
```

Fig. 5.102. - Rotation of a GTL profile




```
N80  (DIS,"END MILL D=8")
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  G21 G41  c1
N92  Z-10
N93  l1
N94  c2
N95  l2
N96  c3
N97  l3
N98  c1
N99  Z 2
N100 G20 G40 l1
N101 E26=E26+60
N102 (ERP)
N103 (URT,0)
N104 G X Y Z10
```

5.17.2. 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 proqram blocks
ERP - define the end of the repetition set
CLS - call a subroutine for execution
EPP - execute a subprogram
BNC
BGT
BLT
BEQ - branch commands
BNE
BGE
BLE

REPEAT A SET OF PROGRAM BLOCKS - RPT/ERP

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:

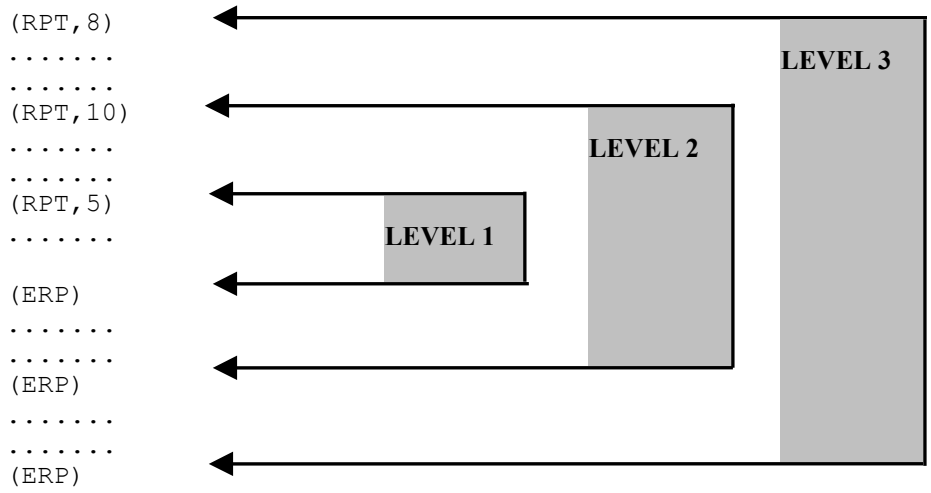
(RPT,n) -repetition command and number of times
.....
..... -set of blocks to be executed the specified number
..... of times
.....
(ERP) -defines the end of the set

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 epeat command (Figure 5.103.).

Fig. 5.103. - Repeat levels



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

USING SUBROUTINES - CLS

The CLS command allows you to call and execute a subroutine that is stored in memory. A subroutine is a sequence of blocks that define a machininq cycle.
The allowable format is:

(CLS,FILE NAME[/DEVICE])

where:

FILE NAME — is the name of the soubroutine 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 nane from the file nane.
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 menory MP1. If MP1 is the default memory, it does not need to be programmed in the CLS comnand.

<u>Example of a Subroutine Call:</u>

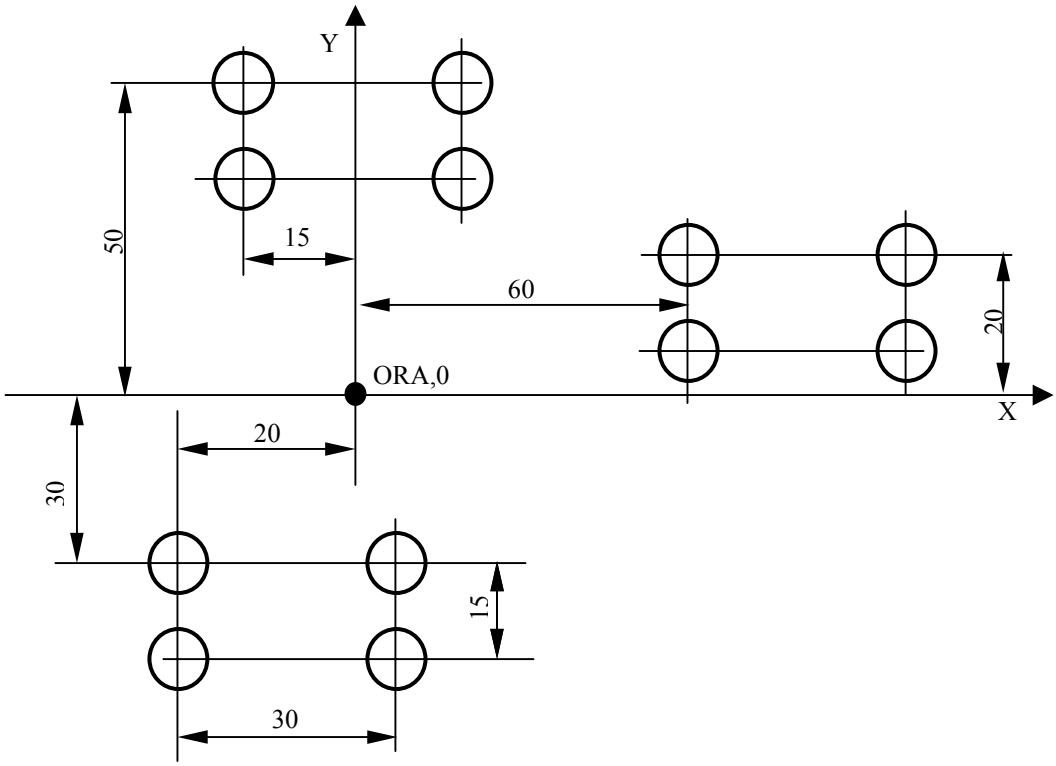
```
Main program
N16 ..........                    Subroutine P800
N17 (CLS,P800)  ---------------->  N500 ..........
N18 ..........                    N501 ..........
..............                    N502 ..........
N67 (CLS,P800)  ----------------  N503 ..........
N68 ..........                  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

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 um other programs.
- Subroutines can be parametric, the numeric values of the parameters are defined in main program during the recall.

Figure 5.104. shows another example for suhroutine programminq.

Fig. 5.104. - Drilling Jig (30x15)



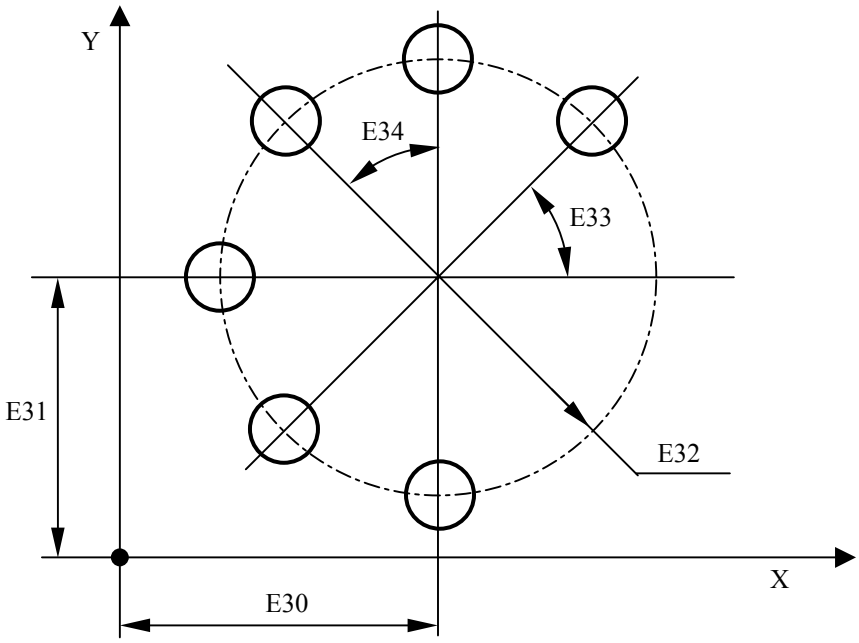

```
Main program
N19 (DIS,"...")
N20 S2000 F180 T2.2 M6
N21 (UOT,O,X-20,Y-25)
N22 (CLS,S600)
N23 (UOT,0,X-15,Y50)
N24 (CLS,S600)
N25 (UOT,0,X60,Y20)
N26 (CLS,S600)
N27 Z...
```

```
Subroutine S600
N501 G81  R.. Z..
N502 XY
N503 Y-15
N504 X30
N505 Y
N506 G80
```

PARAMETRIC SUBROUTINES

In a parametric subroutine, the numeric values of the parameters (G, S, X, Z, F, etc.) are defined in the main progran durinq the recall. Fiqure 5.105. shows an example of parametric subroutine.

Fig. 5.105. - Circular grid



E1 = No. of holes
E2 = fixed cycle G
E30 = X of the grid centre
E31 = Y of the grid centre
E32 = grid radius
E33 = starting angle
E34 = anqular pitch
E36 = R dimension
E37 = Z dimension

```
...............
N43 (DIS,"TWIST DRILL D12")
N44 T2.2 M6 S900 F70 M3
N45 E30=50
N46 E31=50
N47 E32=30
N48 E33=45
N49 E34=45
N50 E1=6
N51 E36=2
N52 E37=-10
N53 E2=81
N54 (CLS,SUB800)
N55 G Z20 M5
...............
```

```
Subroutine SUB800
N799 (DIS,"CIRCULAR GRIDS")
N800 (UOT,0,XE30,YE31)
N801 E25=E33
N802 (RPT,E1)
N803 (URT,E25)
N804 GE2 RE36 ZE37
N805 XE32 Y
N806 E25=E25+E34
N807 G80
N808 (ERP)
N809 (UAO,0)
N810 (URT,0)
```

For further details about parametric programming, refer to section 5.16.

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 alphanuneric 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 centering 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 6.10 shows the format and function of branch commands.

Table 5.10. 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,VAR1,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 a 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

Notes:

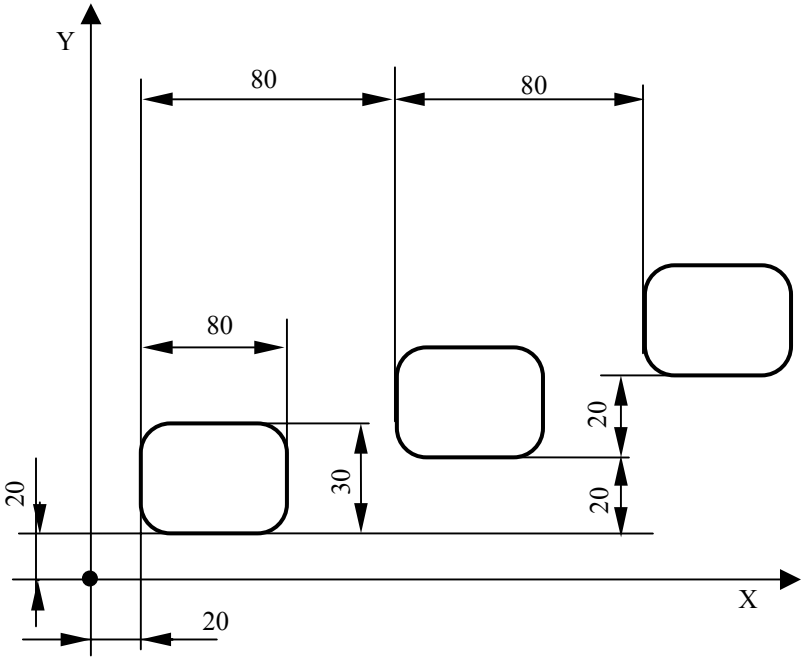
- If the variable has character format (CH), the control will check a string of characters, the length of which is indicated by the index preceding CH. If no index is specified, the control defaults it to 1. For example,

(BEQ,SYVAR2.3CH,"ABC",END)	branch to "END" if the 3 characters from SYVAR2 are A B C.

- If variables VAR1 and VAR2 have LR or RE format, any mathematical operation between them has an approximatian error typical of the processor that cumulates after each successive operation. This error must be taken into consideration when defining the decimal threshold below which both variables can be considered equal. For example, branch to LAB1 if E42=E41 by comparing (BLT) the difference between them to the threshold:

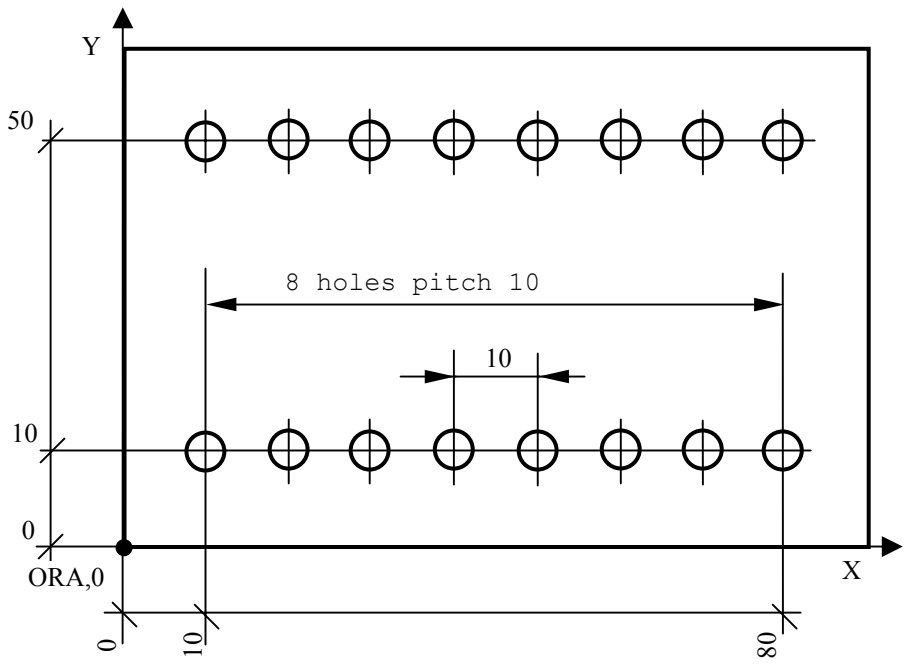
```
E42=0.021
E41=0.015
E47=0.0015
E43=0.0001              -threshold=0.0001
"LAB2" E41=E41+E47
E44=ABS(E42-E41)
(BLT,E44,E43,LAB1)      -branch to LAB1 if E44=E42-E41<E43
(BNC,LAB2)              -unconditioned branch to LAB2
"LAB1"(DIS,E41)
```

Fig. 5.106. - Utilization of RPT/ERP to repeat cycles at a constant distance



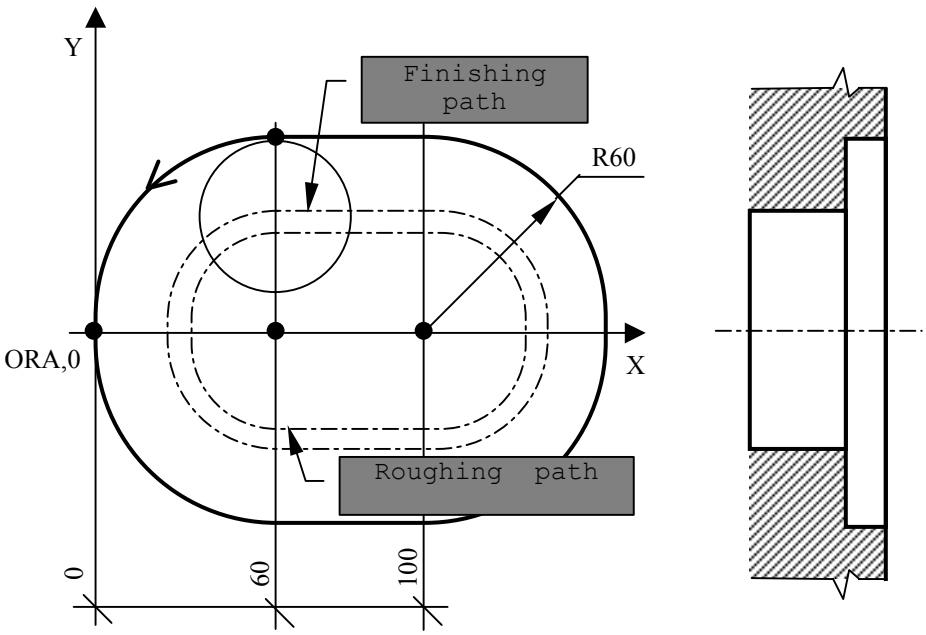

```
(DIS,"N.3 GROOVES")
(DIS,"MILL D12")
N1  S600 T6.6 M6 M3
N2  (RPT,3)
N3  X40 Y35
N4  Z2
N5  (RPT,2)
N6  G91  Z-8
N7  G90 G1  G41  X40 Y2O F300
N8  X60
N9  Y50
N10 X20
N11 Y20
N12 G40 X40
N13 Y35 F1000
N14 (ERP)
N15 G Z2
N16 (UIO,X80,Y2O)
N17 (ERP)
N18 (UAO,0)
N19 Z20
N20 XYM30
```

Fig. 5.107. - RPT/ERP for equidistant holes




```
(DIS,"EQUIDISTANT HOLES")
N1  F200 S900 T1.1  M6
N2  G81  R5 Z-10 M3
N3  X10 Y10
N4  (RPT,7)
N5  G91  X10
N6  (ERP)
N7  Y40
N8  (RPT,7)
N9  X-10
N10 (ERP)
N11 G80 G90 XY M5
.....................
```

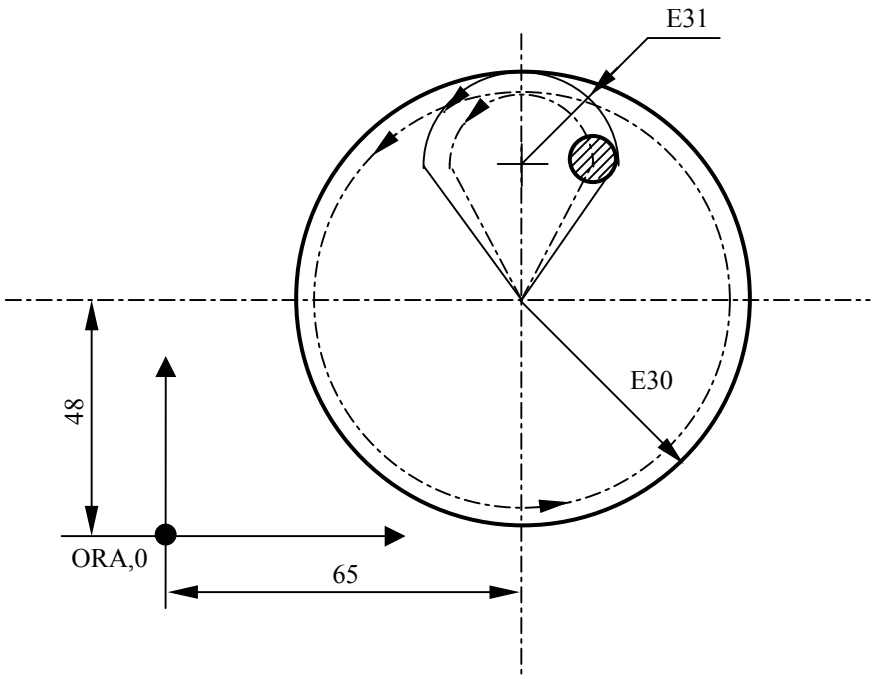
Fig. 5.108. - Roughing Cycle with Finish




```
(DIS,"DEFINITION OF STOCK ALLOWANCE")
N1  S350 T6.6 M6
N2  X60 Y M3
N3  Z-50
N4  UOV=0.5
N5  (RPT,2)
N6  G1 G41  X60 Y60 F500
N7  G3 Y-60 I60 J
N8  G1  X100
N9  G3 Y60 I100 J
N10 G1  G40 X60
N11 G Y
N12 UOV=0
N13 (ERP)
N14 Z M5
N15 X Y M30
```

EXAMPLES OF PARAMETRIC SUBROUTINES

Fig. 5.109. - Milling-Boring operation



E30 = hole radius
E31 = starting radius
E25 = machining feed

Main program

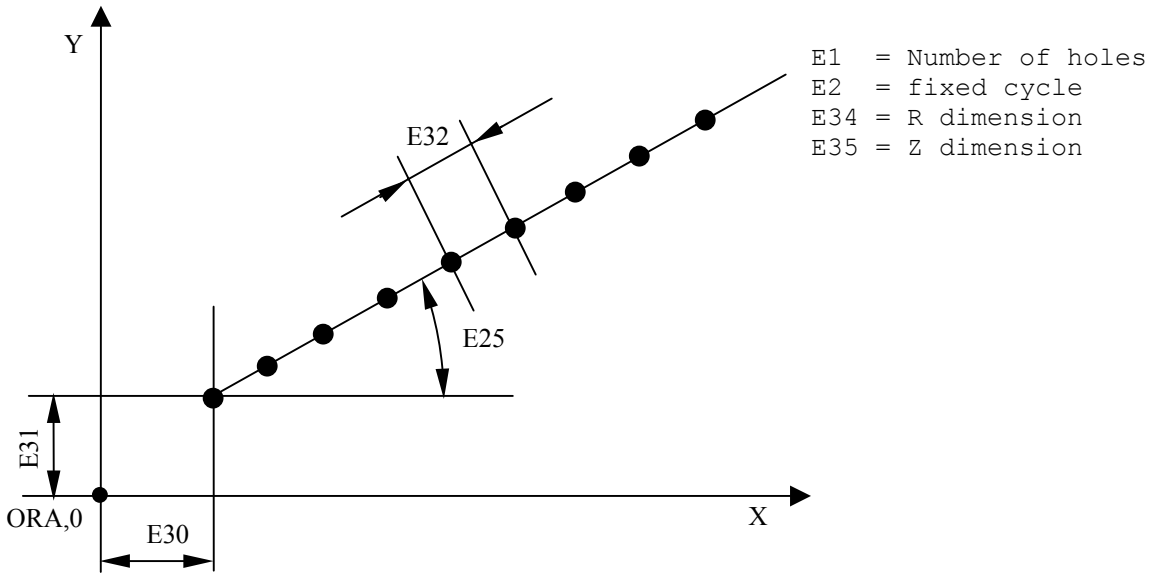
```
..........
N128 (DIS,"DISK MILL D=32")
N129 S330 T12.12 M6
N130 (UOT,O,X65,Y48)
N131 G X Y M3
N132 E30=62.5
N133 E31=25
N134 E25=160
N135 (CLS,SUB500)              -Subroutine call
N136 (UAO,0)
N137 G Z
..........
```

Subroutine SUB500

```
N500 E35=E30-E31
N501 G1  G41  X E31  YE35  F2000
N502 G3 X YE30 I JE35  FE25
N503 I J
N504 E31=NEG(E31)
N505 G40 XE31  YE35 I JE35
N506 G1  X Y F2000
```

Important. When you program the nominal profile, E30 is the hole radius. E31 must always be grater than the radius of the mill.

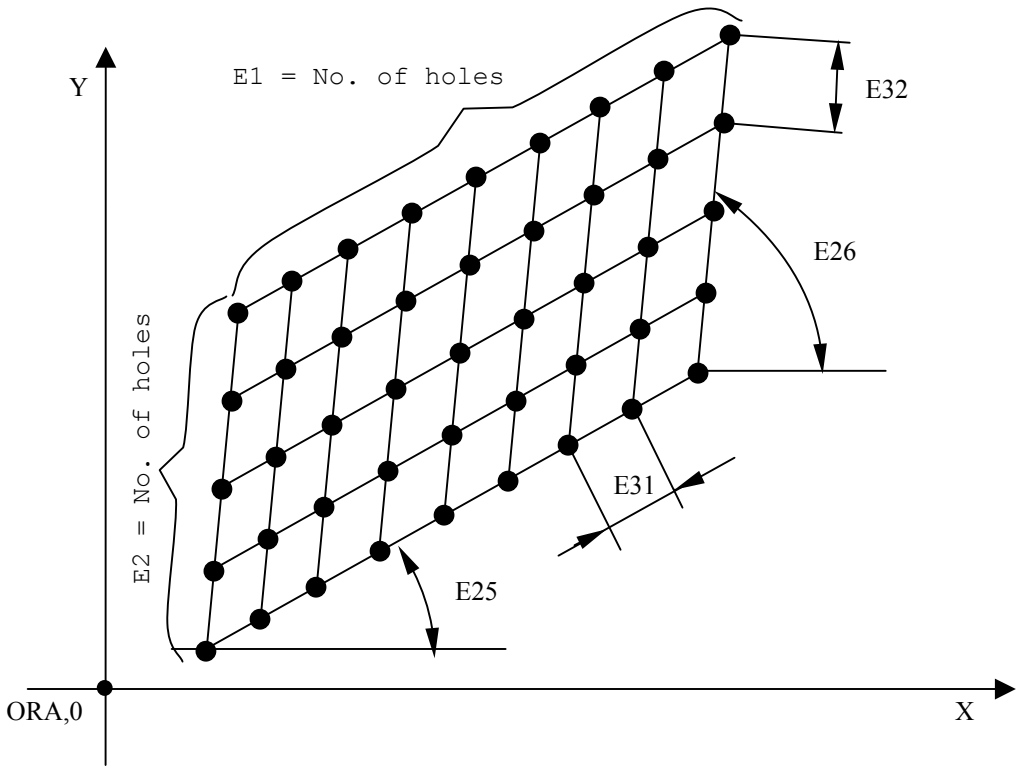
Fig. 5.110. - Linear grid




```
Main program                          Subroutine SUB600
N29 (DIS,"CENTRE BIT D=3")            N600 (UOT,0,XE30,YE31)
N30 S1000 F60 T4.4 M6 M3              N601 (URT,E25)
N31 E25=30                            N602 E3=E1-1
N32 E30=12                            N603 GE2 RE34 ZE35
N33 E31=12                            N604 XY
N34 E32=20                            N605 (RPT,E3
N35 E1=9                              N606 G91  XE32
N36 E34=2                             N607 (ERP)
N37 E35=-3                            N608 G90
N38 E2=81                             N609 G80
N39 (CLS,SUB600)                      N610 (UAO,0)
N40 (DIS,"TWIST DRILL D=9")           N611 (URT,0)
N41  S880 F100 T5.5 M6 M3
N42 E35=-20
N43 (CLS,SUB600)
..................
```

Blocks N31 to N38 assign a numeric value to each parameter for the centering operation.
Block N39 calls up subroutine SUB600.
Block N42 calls up subroutine SUB600 for a drilling operation.

Fig. 5.111. - Hole grid

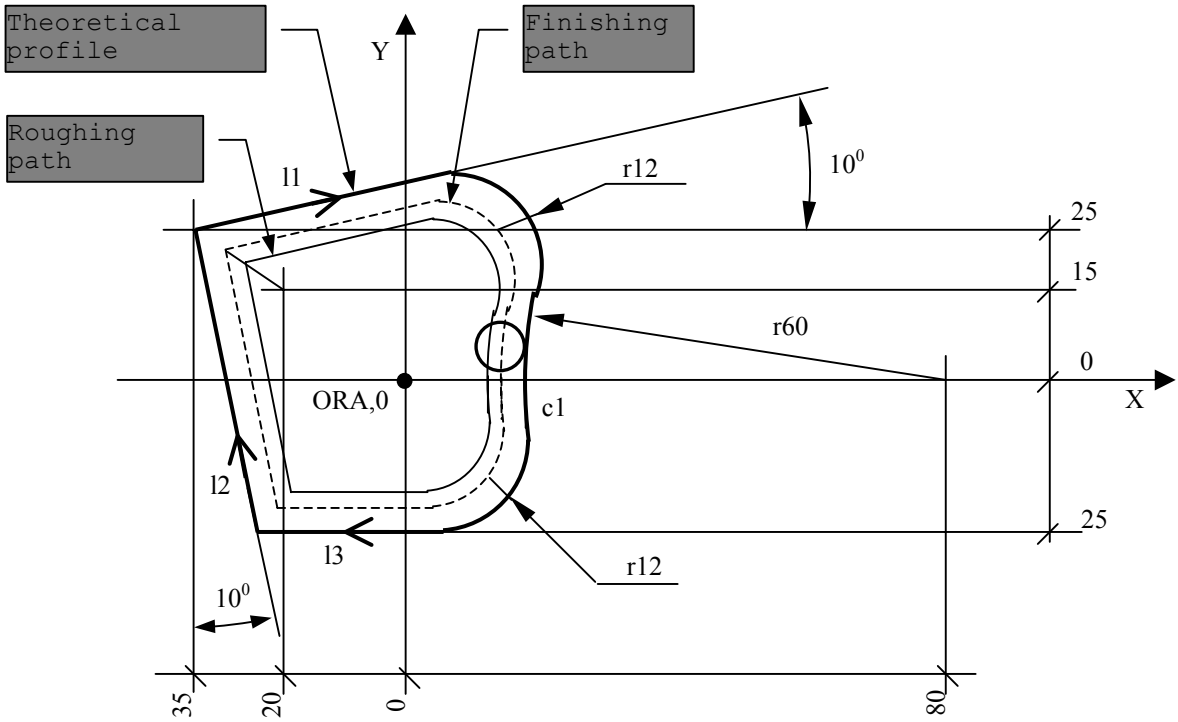



```
Main program
N20  (DIS,"TWIST DRILL D=5")
N30  F900 S3000 T2.2 M16
N40  G X20 Y25 M3
N50  G81 R3 Z-20
N60  E25=30
N70  E26=80
N80  E31=15
N90  E32=12
N100 E1=8
N101 E2=S
N102 (CLS,SUB700)
..........
```

```
Subroutine SUB700
N700 E33=E31*COS(E25)
N701 E34=E31*SIN(E25)
N702 E41=0
N703 E42=0
N704 E5=E1-1
N705 (RPT,E2)
N706 G91 XE41 YE42
N707 (RPT,E5)
N708 XE33 YE34
N709 (ERP)
N710 E41=E32*COS(E26)
N711 E42=E32*SIN(E26)
N712 E33=NEG(E33)
N713 E34=NEG(E34)
N714 (ERP)
N715 G80 G90
```

Blocks N60÷N101 assign a numeric value is ascribed to each parameter.
Block N102 calls up subroutine SUB700.

Fig. 5.112. - Utilization of EPP in a milling operation




```
N1  l1=X-35 Y25,a10
N2  l2=X-35 Y25,a100
N3  l3=X Y-25,a180
N4  c1=I80 J0 r60
N5  (DIS,"ROUGHING MILL D=8")
N6  F400 S2000 T4.4 M6 M3
N7  UOV=1
"START" N8
N9  G X-20 Y15
N10 Z-10
N11 G21 G42 l2
N12 l1
N13 r-12
N14 c1
N15 r-12
N16 l3
N17 l2
N18 G20 G40 l1
N19 G X-2O Y15
N20 Z
"END" N21
N22 (DIS,"FINISHING MILL D=8")
N23 F500 S2500 T5.5 M3 M6
N24 UOV=O
N25 (EPP,START,END)
```

Fig. 5.113. - Use of EPP in a point-to-point operation

```
N1  (DIS,"DRILLING CENTRE HOLES")
N2  F300 S200  T1.1  M3 M6
N3  G81 R0 Z-4
"D6" N4
N5  X100 Y100
N6  X-100
N7  Y-100
N8  X100
"D10" N9
N10 X40 Y40
N11 X-40
N12 Y-40
N13 X40
"END" N14
N15 G80
N16 (DIS,"BIT D6")
N17 F200 S1800 T2.2 M3 M6
N18 G81  R Z-22
N19 (EPP,D6,D10)
N20 G80
N21 (DIS,"BIT D10")
N22 F220 S1600 T3.3 M3 M6
N23 G81  R Z-24
N24 (EPP,D10,END)
N25 G80
N26 ........
............
```

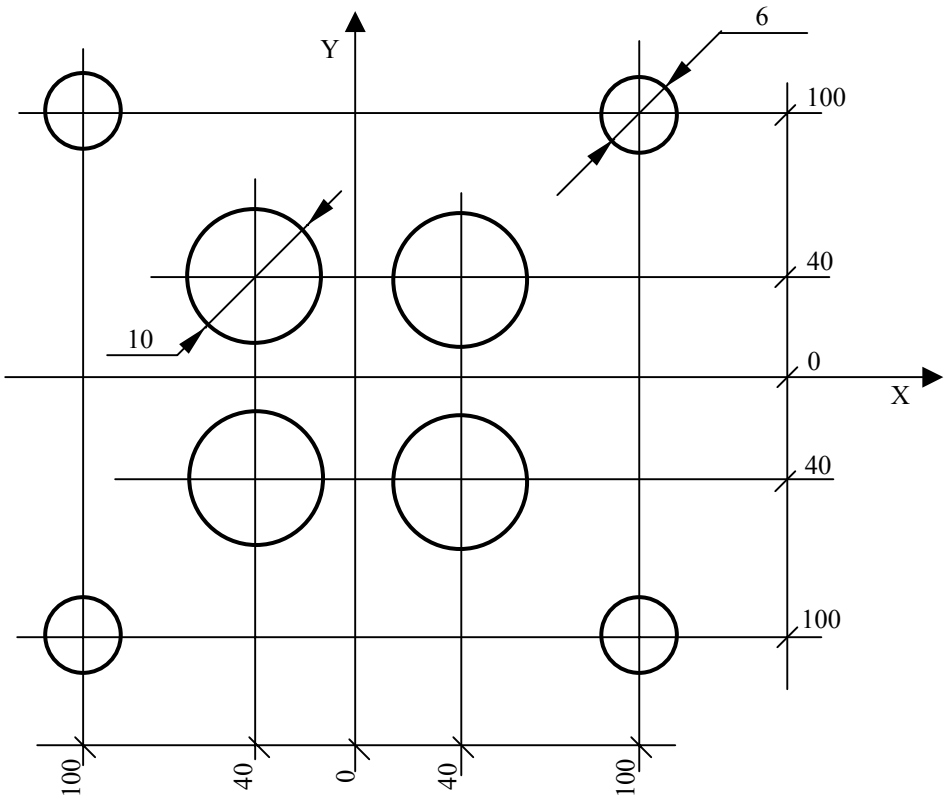
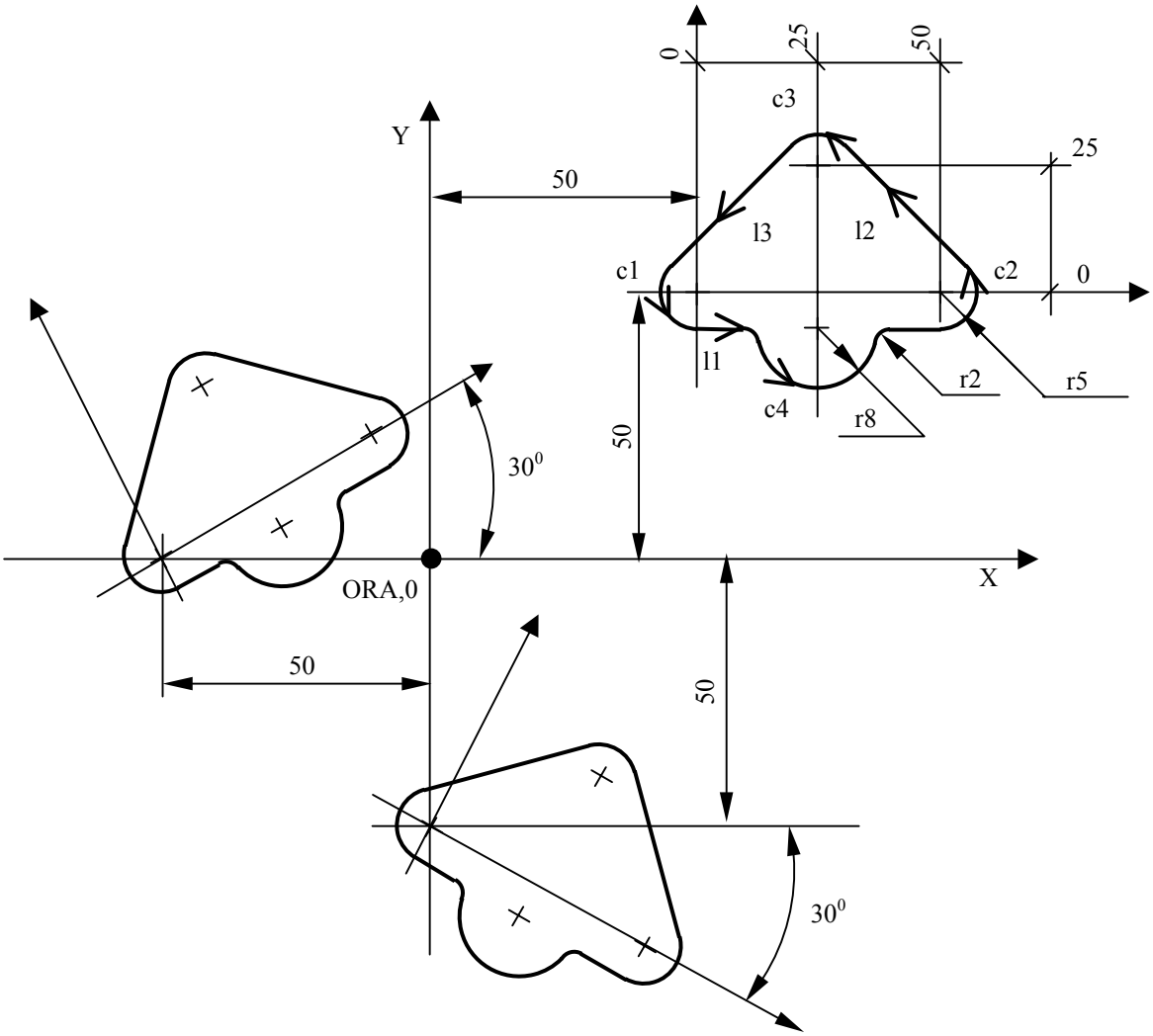


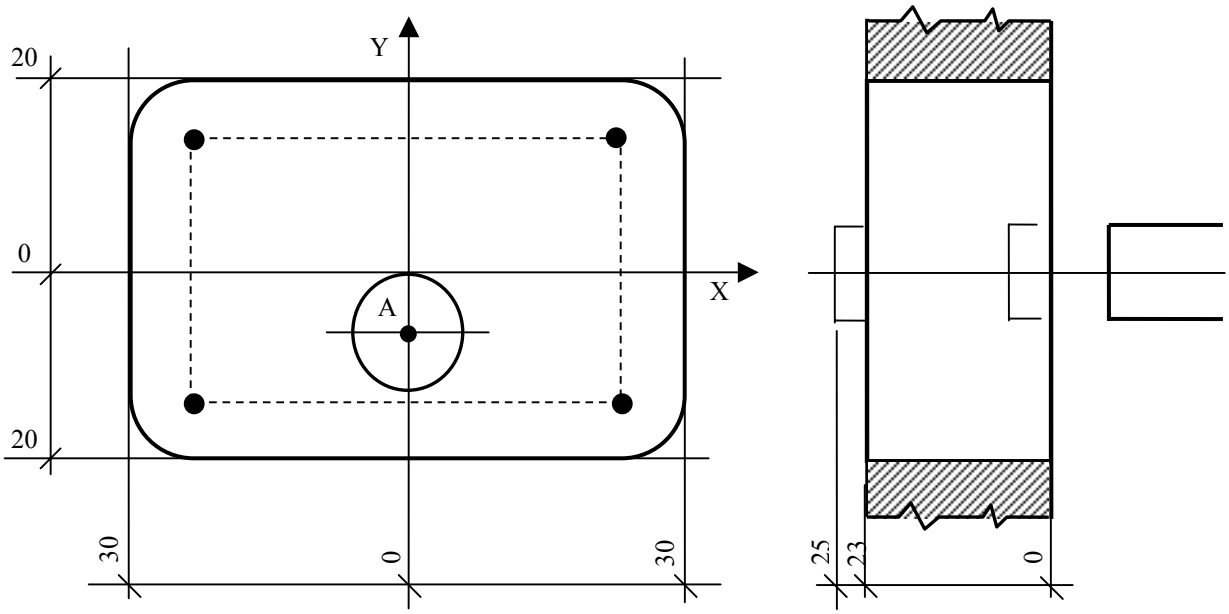
Fig. 5.114. - Use of EPP to repeat a given figure with different plane orientation




```
N32 F200 S1200 T1.1  M6 M3
N33 c1=I J  r5
N34 c2=I50 J  r5
N35 c3=I25 J25 r5
N36 l1=c1,c2
N37 l2=c2,c3
N38 l3=c3,c1
N39 c4=I25  J-5 r8
N40 (UOT,0,X50.Y50)
"1" N41
N42 G21  G41  c1
N43 Z-10
N44 l1
N45 r-2
N46 c4
N47 r-2
N48 l1
N49 c2
N50 l2
N51 c3
N52 l3
N53 c1
N54 Z2
N55 G20 G40 l1
"2" N56
N57 (UOT,0,X-50,Y)
N58 (URT,30)
N59 (EPP,1,2)
N60 (UOT,0,X,Y-50)
N61 (URT,-30)
N62 (EPP,1,2)
N63 (URT,)
N64 (UAO,)
N65 G Z
```

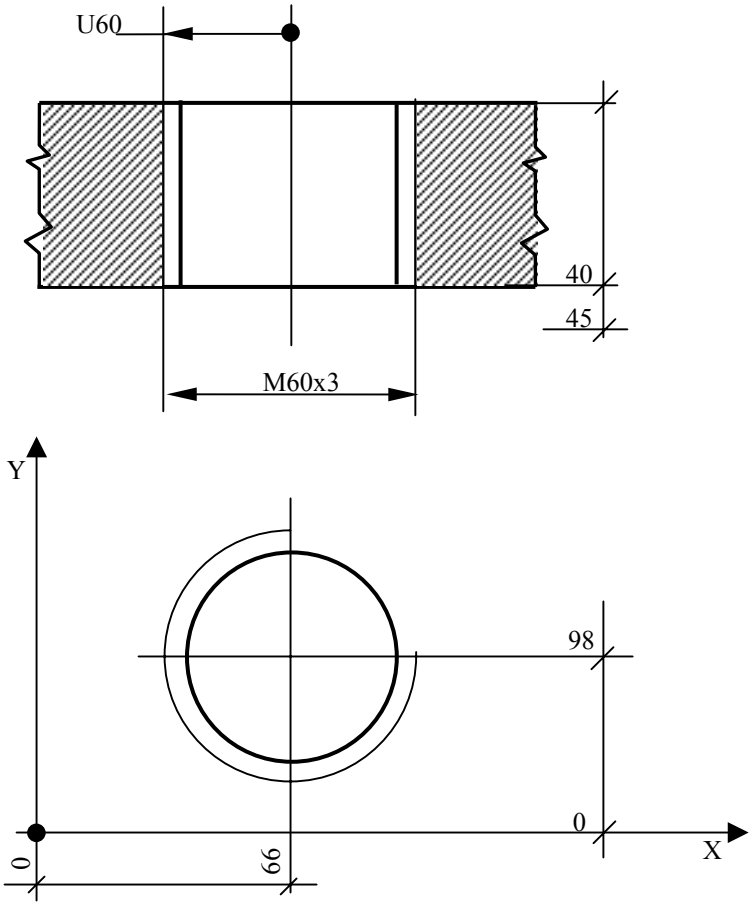
EXAMPLES OF CONDITIONED BRANCHING

Fig. 5.115. - Execution of a slot




```
N1  (DIS,"MILL A SLOT")
N2  F500 S2000 T1.1 M6 M3
N3  E31=-3.5
N4  E32=-24
"START" N5
N6  G X Y-10
N7  Z E31
N8  G1 G42 X Y-20
N9  X-30
N10 Y20
N11 X30
N12 Y-20
N13 G40 X
N14 Y-10
"END" N15
N16 E31=E31-3.5
N17 (BGT,E31,E32,START)
N18 E31=-25
N19 (EPP,START,END)
N20 G Z10
N21 X Y
.................
```

Fig. 5.116. - Example of Cylindrical Thread




```
E30 = diameter of the 1st.cut
E31 = diametric increment
      of cut depth
E32 = return diameter
E33 = final diameter
```

```
............
 N1   (DIS,"THREAD DIA 60")
 N2  S150 Y5.5 M6
 N3  G X66 Y98 Z5 M3
 N4  E30=56.8
 N5  E31=0.5
 N6  E32=50
 N7  E33=60
 "I" N8
 N9  G Z5
 N10 UE30
 N11 G33 Z-45 K3
 N12 GUE32
 N13 E30=E30+E31
 N14 (BGT,E30,E33,F)
 N15 (BNC,I)
 "F" N16
 N17 GUE32
 N18 Z5
 N19 UE33
 N20 G33 Z-45 K3
 N21 GUE32
 N22 Z20
 ..........
```

5.17.3. USING MISCELLANEOUS THREE-LETTER COMMANDS

This section deals with three-letter commands used for miscellaneous operations. The following commands belong to this class:

DPI - defines the plane of interpolation
DTL - defines the positioning tolerance level
DLO - defines the limits of the operating field
CTL - toggles the control between lathe/mill configurations

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,Z,U)	specifies the interpolation plane formed by axes Z and U

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 3 axes. Program the dimension in the measuring unit (G70/G71) active when the DTL command is executed.

For example:

(DTL,X.1,Y.O5) specifies a positioning tolerance of 0.1 for the X axis and 0.0.5 for the Y 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 SEPVO ERROR.
- The DTL command will cause an error if any of the followinq 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, an error message will be displayed and the control will use the value declared during characterization.
The format is:

(DLO,VAR-1)

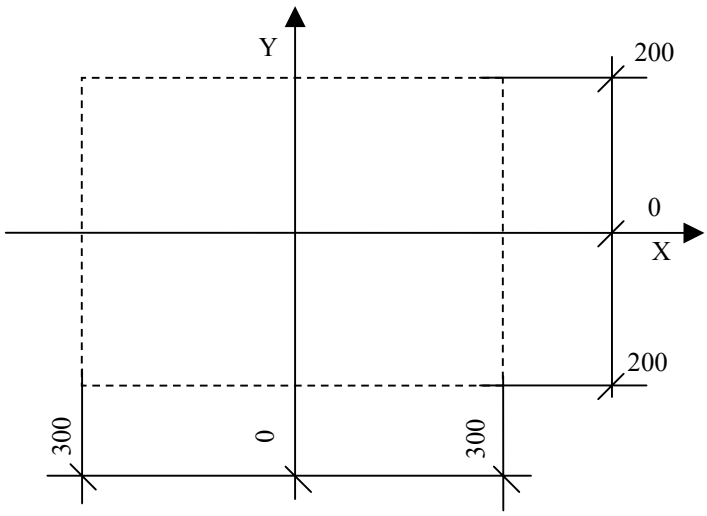
where:

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. 5.117.)

Fig. 5.117. - Definition of the Operating Field

```
(DLO,X300 X-300)
..........
(DLO,Y200 Y-200)
..........
```



Notes:

- You can program as many as 8 couples of different operating limits.
- Program dimensions in the measuring unit (G70/G71) active when DLO is executed.
- DLO is deactivated by RESET.

SWITCHING BETWEEN LATHE AND MILL CONFIGURATIONS - CTL

The CTL command allows you to select the lathe 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 lathe (refer "NC-110 USER'S MANUAL").
The allowable formats are:

(CTL,T) activates the LATHE configuration

or

(CTL) returns the control to the original MILL configuration

The RESET key also returns the control to the MILL configuration.

5.17.4. USING I/O COMMANDS

These commands allow you to perform I/O operations from inside a program. The commands in this class are:

DIS - display a variable
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 E parameters.

  The control will display the sequence VARIABLE NAME=value. For example, (DIS,EZ7)

. 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 oriqin 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 and decimal fractions of a second. 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 synchronization (refer to section 5.21.).

5.17.5. MONITORING TOOL LIFE

This class of commands allow you to control tool life parameters from inside a program. At present, there is only one command in this group:

TOF - declare tool out of useful life

DECLARING A TOOL OUT OF USEFUL LIFE - TOF

The TOF command allows you to declare the useful life of the specified block exceeded.
The allowable format is:

(TOF,n)

where:

n is the number of the tool to be declared "OUT OF USE". You can express it either directly with whole number or an integer E parameter (E10 to E19).

For example:

(TOF,22) declares tool 22 out of useful life.

(TOF,E14) declares tool in E22 out of useful life

Notes:

- TOF is not operant if the axes are disconnected or if memorized search has been activated.
- If tool life management is active, TOF declares out of useful life the main or alternative tool whose status is indicated by E or F (tool on spinde or already used) in the tool life file.

5.17.6. 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:

UCG - define the qraphic display field
CLG - clear the graphic display
DCG - disable the graphic display

DEFINING THE GRAPHIC DISPLAY FIELD - UCG

This command initializes 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,[AXIS3]])

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

AXIS3 (option) defines the third axis, considered normal to the plane defined by axis1 and axis2. It is mostly used to show the points where fixed cycles are performed.

(UCG,2,X100 X150,Y50 Y250,Z) - activates graphic display not coordinated with the axes movement (n=2). The graphic display shows moves between X100 and X150 for the abscissa axis and between Y50 and Y250 for the ordinate axis with respect to the current origin. Z axis movements are also displayed.

Notes:

- The graphic display takes into account the programmed absolute, temporary and incremental origins. Instruction UCG must be programmed after the origin the program refers to.
- With instruction UCG the control performs graphic calculations regardless of which EDP or graphic screen is active and displayed to the mperatmr.
  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 synchronization (refer to section 6.21).
- Field limits are always expressed in the default measuring unit.

CLEARING THE GRAPHIC DISPLAY - CLG

This command clears the currently displayed profile from the screen, leaving the coordinates system. The format is:

(CLG)

DISABLING THE GRAPHIC DISPLAY - DCG

This command disables the graphic display. The format is:

(DCG)

N.B. This command must be programmed after instruction CLG.

5.17.7. 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. Proqram 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 vahue 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)
..........
(RQU,10,10,ZE40)
..........
(RQU,7,7,KE41)
```

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, pogram RQP with the same format.

- The requalification value is always expressed in the machine default measuring unit. If the offset value in the file has a negative sign, this means that it is expressed in the alternative measuring unit. In this case, the requalification value is first converted, and then added to the value present in the file
  In other words, after a requalification:

  Metric system machine:
  New offset = Old offset + requalification value * 1/25.4

  Imperial system machine:
  New offset = Old offset + requalification value * 25.4

  The sign (positive or negative) makes part of the requalification value.

- 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 tool life management is active, RQU requalifies the offset of the main or alternative tool whose status is indicated by E or F (tool on spindle or already used) in the tool life file.

5.18. PROTECTED AREAS

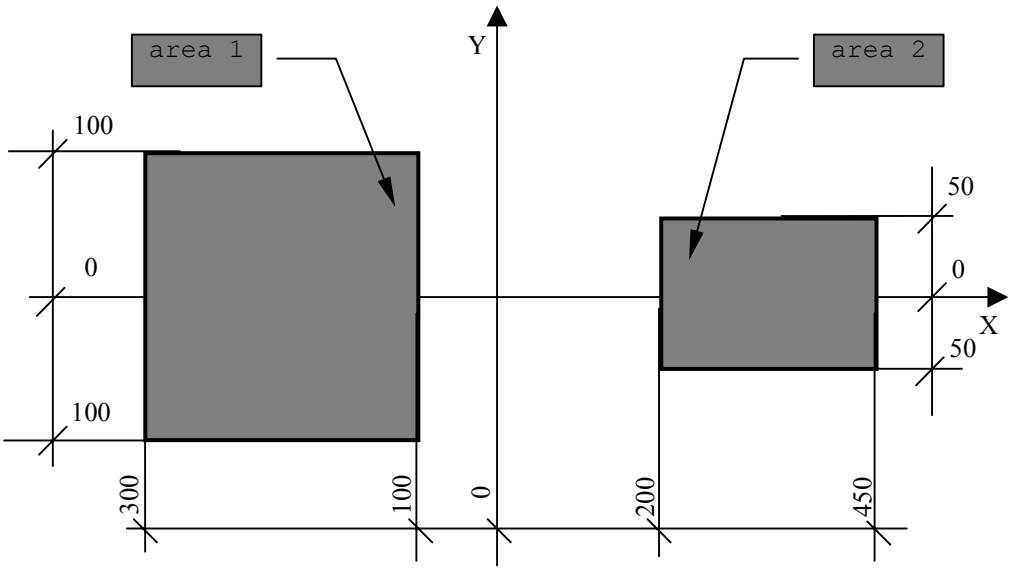
These commands allow you to activate or deactivate protected areas, i.e. areas the axes moving on a plane are not allowed to enter.The commands in this class are:

DSA - defines a protected area
ASC - activates a protected area
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,AXIS1I,AXIS1S,AXIS2I,AXIS2S
(ASC,n)
(DSC,n)

where:
n number of the area
AXIS1I more negative abscissa
AXIS1S more positive abscissa
AXIS2I more negative ordinate
AXIS2S more positive ordinate

Fig. 5.118. - Example of Protected Area Programming




```
N1   (DSA,1,X-300 X-100,Y-100 Y100)        N5  T1.1  M6
N2   (DSA,2,X200 X450,Y-50 Y50)            ...........
N3   (ASC,1)                               N80  (DSC,1)
N4   (ASC,2)                               ............
                                           N99 M30
```

Note. RESET cancels protected areas.

5.19. MANAGEMENT OF REAMING/FACING HEADS (DIAMETER AXIS)

The diameter or U axis aallows you to execute:

- boring operations on cylindrical or conical holes;
- circular radii (concave or convex);
- chamfers;
- grooves;
- facing operations;
- threads.

Programming the Diameter Axis

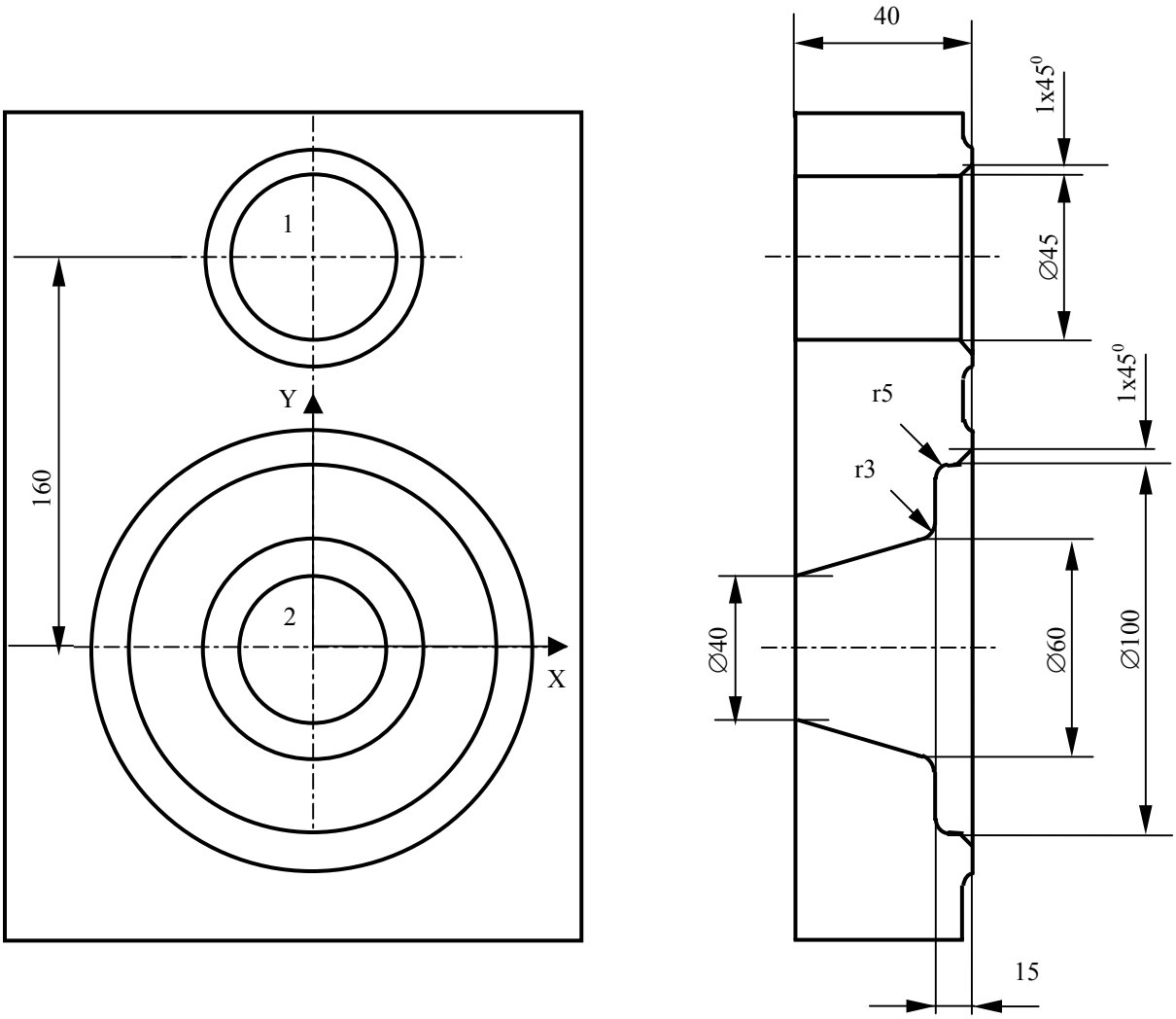
The dimension of the U axis is the diameter. Program the values with the current measuring unit, millimeter (G70) or inch (G71).
The movement of the U axis is simultaneous and coordinated with that of axes X, Y and Z programmed in the same block.
The allowable rates are: rapid rate (G00) and machining rate (G01, expressed in mm/min using function F).
Before programming the execution of a profile moving the U axis, define the interpolation plane with this command:

(DPI,Z,U)

When using a diameter axis, you can program the profile with standard or GTL programming, apply tool radius compensation (G41 or G42), and define the stock allowance (UOV command).
The values of the stock allowance and of temporary and incremental origins are applied to the radius.

Fig. 5.119. shows an example of programming with diameter axis.

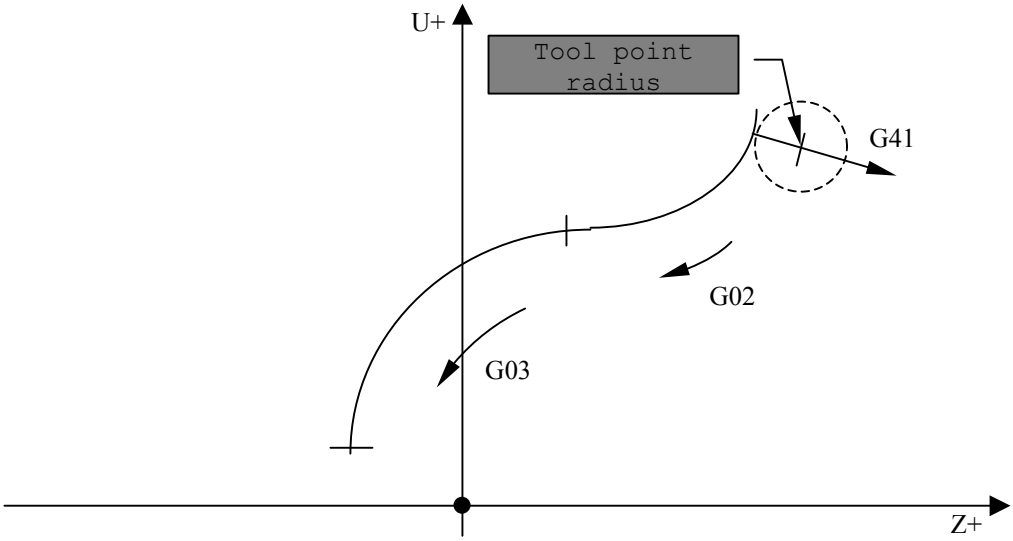
Fig. 5.119. - Finishing operation with diameter axis




```
N116  (DIS,"FINISHING  WITH REAMING/FACING HEAD")
N117 F60 S630 T9.9 M6
N118 (DPI,Z,U)            -define .interpolation plane
N119 (UAO,1)              -call the head absolute origin
N120 (UOT,0,Z-200)        -temporary Z origin
N121 X Y160 T10 M3        -positioning at hole 1
N122 G41 Z2 U51
N123 G1  Z-1  U44.98      -execution of the chamfer
N124 Z-44                 -execution of hole 0=45  0/-0.04
N125 G G40 U40
N126 Z2 F40 S380
N127 G41  Y U106          -positioning at hole 2
N128 G1  Z-1  U99.975     -execution of the chamfer
N129 Z-15                 -execution of hole 0=100  0/-0.05
N130 r5                   -execution of  radius R = 5
N131 U60                  -counterboring
N132 r-3                  -execution of  radius R = 3
N133 Z-40 U40             -tapering
N134 G40 Z-44             -travel of Z axis continues
N135 G U35
N136 Z100 M5
N137 (DPI,X,Y)
N138 (UAO,0)
```

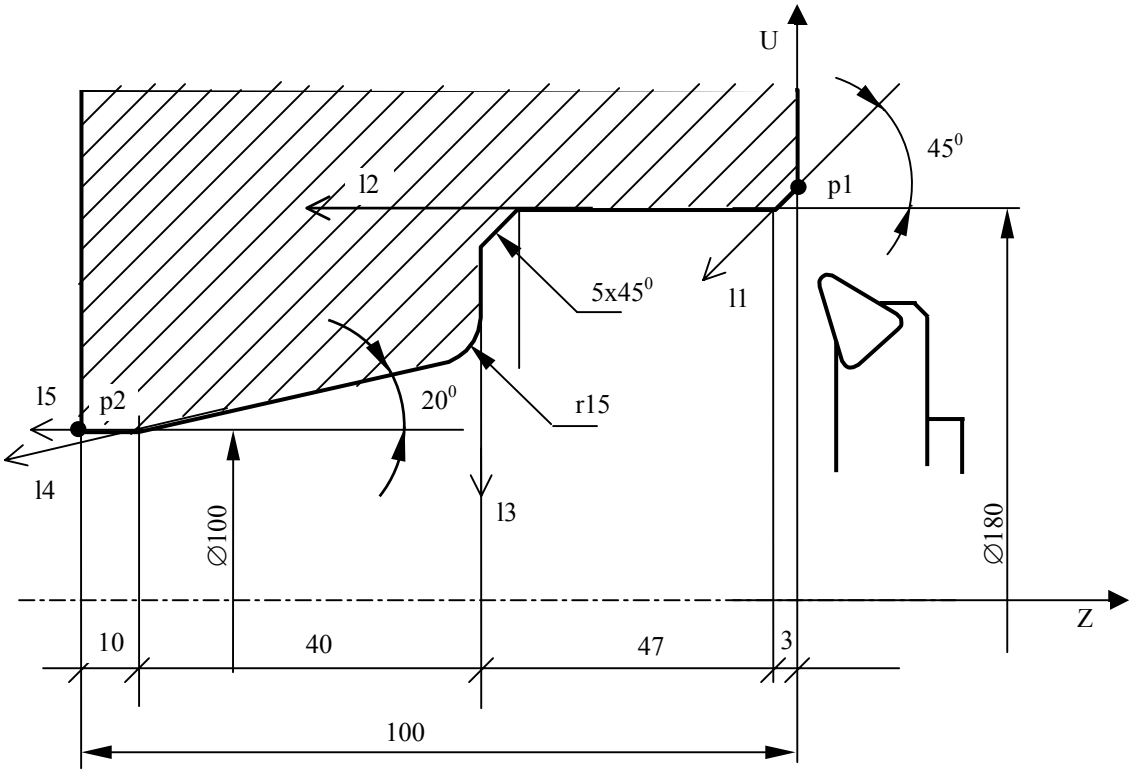
Important. The direction of the arcs (programmed with G02, G03 or r) and the tool offset must be selected with reference to the Z-U plane.

Example (Fig. 5.120.):



Negative diameters are usually not programmed. Therefore, select the first two quadrants of the Z-U plane.

Fig. 5.121. Example of GTL programming with U axis

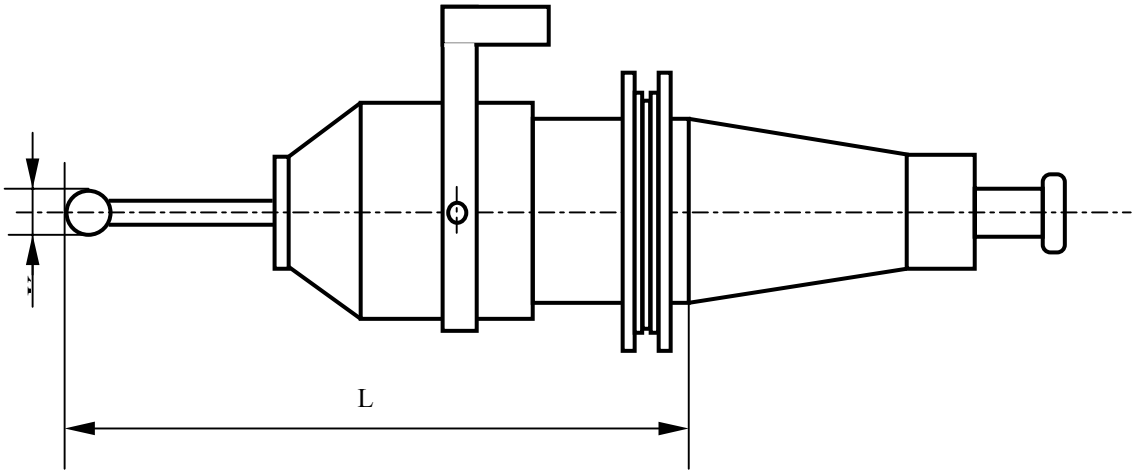



```
N1  (DIS,"U AXIS FINISHING"
N2  (DPI,Z,U)
N3  p1 = Z U186
N4  l1 = p1, a-135
N5  l2 =Z U180, a180
N6  l3 =Z-50 U, a-90
N7  l4 =Z-90 U100, a-160
N8  l5 =Z U100, a180
N9  p2 =Z-100 U100
N10 S..F..T20.20 M6 M3
N11 (UA0,15)
N12 (UIO,X..,Y..,Z..)
N13 G X Y B
N14 U190
N15 Z2
N16 G21  G41  p1
N17 l1
N18 l2
N19 b5
N20 l3
N21 r-15
N22 l4
N23 l5
N24 G20 G40 p2
N25 G U90
N26 Z2
N27 (DPI,X,Y)
N28 (UAO,0)
N29 ...
```

5.20. MANAGING AN ELECTRONIC PROBE

An electronic probe is a measuring device. You can mount it on lhe axis and manage it like a tool. The probe can have length and diameter offsets applied to it. You can also mount the probe in a fixed position and use it as an electronic gauge.

Fig. 5.122. - Probe



With G72, the probe measures the coordinates of a point in the space.

With G73, the probe measures the coordinates of the centre and radius of a circle.

With G74 and behaving as an electronic gauge, the probe measures the displacements from theoretical points.

The control stores measured values in E parameters that you defined in the probing cycles.

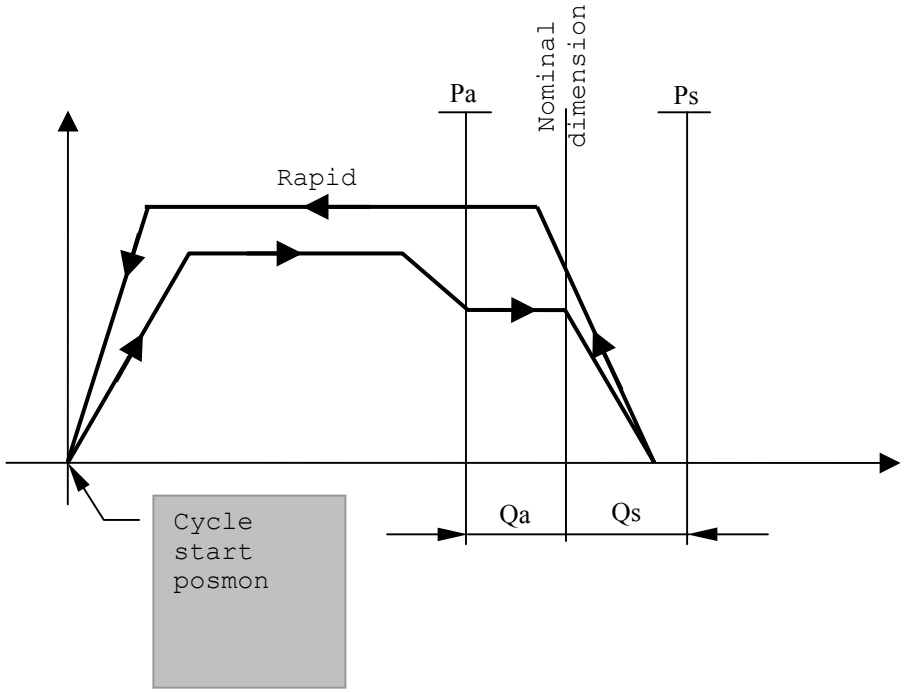
When the control executes a probing cycle, it performs the following sequence of moves to measure a point (Fig. 5.123.):

a) Move at rapid to the approach point (Pa).

b) Move at measuring speed (Vm) to the point where the probe triggers, them stop and store the dimensions.

c) Return at rapid to the start position of the probing cycle.

Fig. 5.123. – Sequence of moves to measure a point



5.20.1. PRESETTING A PROBING CYCLE

The first time that you use a probem or whenever probing cycle conditions change, you must:

- define the probing parameters;
- perform the dynamic measurement of the diameter of the probe ball;
- requalify the probe with respect to the spindle axis;
- perform the dynamic measurement of the prohe length (Z).

a) 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

Example:

DPT,10,12,1000 Press [SEND] from the keyboard
(DPT,10,12,1000) from the program

Important. To determine the measuring speed, consider that probing with Vm=1000 mm/min generates 1 um error.

b) DYNAMIC MEASUREMENT OF THE DIAMETER OF THE PROBE BALL (APPARENT DIAMETER)

To measure the diameter of the probe ball, use a sample ring, the center of which corresponds to the origin of the axes X and Y.

c) REQUALIFICATION OF THE PROBE WITH RESPECT TO SPINDLE AXIS

To correct the variance between the centre of tlie probe ball and the spindle axis, you must requalify the probe.
Refer the probe to the center of the sample ring, which corresponds to the origin of axes X and Y.

d) DYNAMIC MEASUREMENT OF THE PROBE LENGTH

To measure the length (Z) of the probe, use the reference point on the sample ring that corresponds to the origin of the Z axis.

The following sequence programs a complete presetting cycle.

We assume having:

- a sample reference ring;
- absolute origin 99 for rotary axis B (if the rotary axis is mounted onto an indexed table);
- absolute origin 99 for axes X and Y at the centre of the hole of the reference ring;
- absolute origin 99 for Z axis on the upper surface of the ring.

The initial values stored in the probe offset are:

Z = nominal length of the probe with respect to the axis of the ball
K = 0

EXAMPLE OF PROBE REQUALIFICATION

```
N1   (DIS,"DPT,RTA,RTO")
N2   T30.30 M6              -probe on spindle
N3   (UAO,99)
N4   (DPT,10,12,600)        -defines probing parameters
N5   RTA=0
N6   RTO=0
N7   E30=...                -assigns diameter to the hole of
                             the sample ring
N8   E31=E30/2
N9   E32=...                -assigns distance d from Z=0 to probing
                             surface on the Z axis  (usually d=0)
N10  E33=E31+10
N11  GBO                    -only if the ring is mounted on the
N12  XY                      indexing table
N13  Z-4
N14  M...                   -activates probe
N15  G73 rE31  E40          -measures hole coordinates (center and
                             radius)
N16  Z100
N17  (DIS,"RTA=",E40,"RTO=",E41)
N18  M0
N19  RTA=E40                -riqualifies probe abscissa
N20  RTO=E41                -riqualifica probe ordinate
N21  E34=(E30-E42*2)        -diameter of apparent ball
N22  (DIS,"DIAMETER="E34)
N23  M0
N24  (RQP,30,30,KE34)       -stores ball diameter on K offset
N25  T30.30 M6              -enables new offset
N26  GXYE33
N27  G72 ZE32 E43           -measures Z dimension on ring surface
N28  E35=E43-E32            -variance between nominal and real
                             value
N29  Z100
N30  (DIS"VARIANCE.Z=",E35)
N31  M0
N32  (RQP,30,30,ZE35)        -requalifies  length offset Z
N33  M30
```

5.20.2. OPERATIONS THAT USE A PROBE

When you use the G72-G73 probing cycles, you can:

- reapply origins;
- check dimensions.

With the G74 probing cycle, a fixed probe (like an electronic gauge) and a tool fitted into the spindle, you can:

- requalify tools;
- check tool wear.

REAPPLYING ORIGINS

The available methods are:

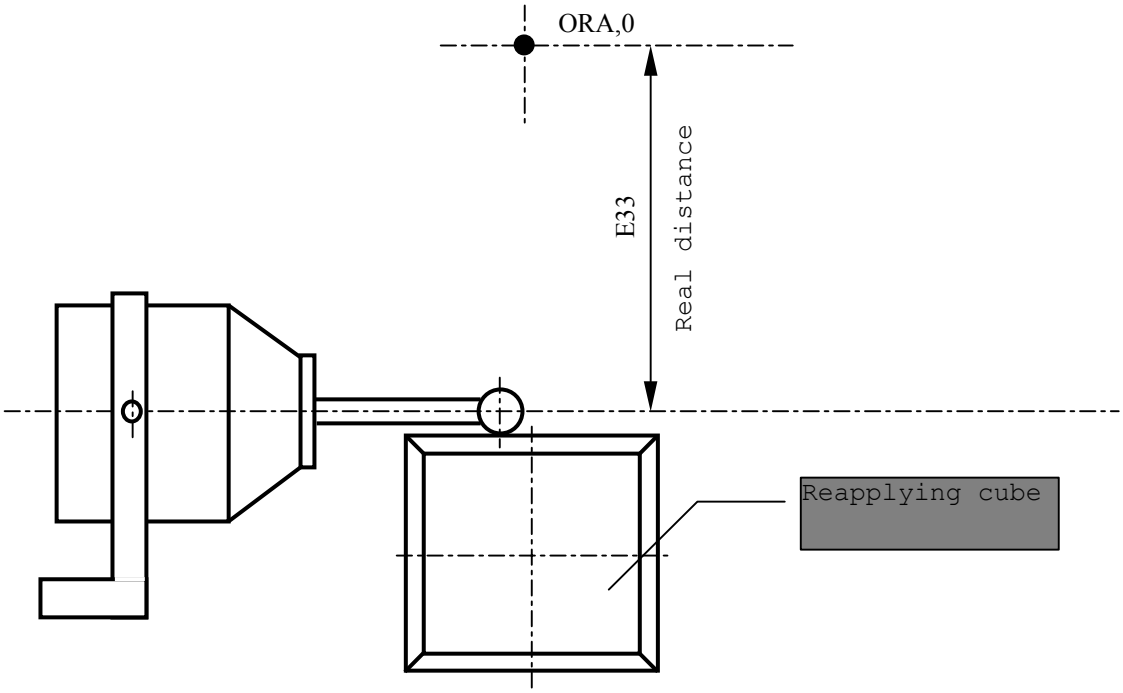
- probing the reference surface;
- positioning on the reference hole.

a) Probing reference surfaces - This method can be used for:

1. Origin reapplication due to thermal drift

The cubic gauge must be at a known dimension. For example (Fig. 5.124.):

Fig. 5.124. - Reapplying the origin of the X axis



Main program

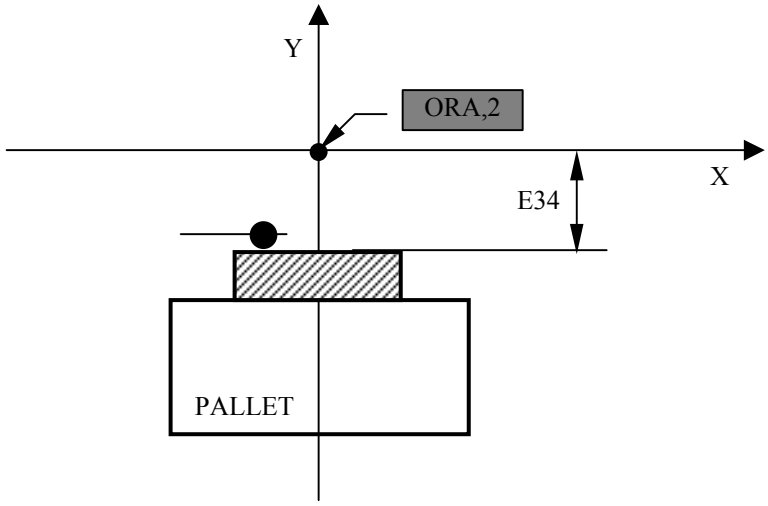
```
..........
..........
N99  E33=-300
..........
/N100  (CLS,TAST3)
..........
```

Subroutine TAST3

```
N500 (DIS,"RQO-DT")
N501 G72 YE33 E32   -measure distance stored in E32
N502 E32=E32-E33
N503 (RQO,0,YE32)   -reapplying origin 0 for Y axis
N504 (RQO,1,YE32)   -reapplying origin 1 for Y axis
N505 (RQO,2,YE32)   -reapplying origin 2 for Y axis
```

2. Reapplying origins on a new pallet

Fig. 5.125. - Reapplying origins for the Y axis (horizontal machining center)



Main program

```
..........
N10 (UAO,1)
..........
..........
N194 M.....        -pallet change
N195 T30.30 M6     -probe onto the spindle
N196 (UAO,2)
N197 GXY
N198 E10=2
N199 E34=-250
/N200 (CLS,TAST4)
...........
```

Subroutine TAST4

```
N500 G72 XE34 E30
N501 E31=E30-E34
N502  (RQ0,1,XE31)   -reapply origin2 for the Y axis
```

b) Positioning on the reference hole - Before programming the probing cycle, you must program the positioning of axes X and Y on the hole axis as well as the positioning of the probe inside the hole. For example:

```
..........
N200 (DIS,"ORIGIN REAPPLYING FOR AXES X AND Y")
N201  T11.11  M6
N202 GX180 Y60
N203 Z-130
N204 G73 r50 E35           - measuring cycle on hole 0
                             100. The X and Y coordinates
                             are stored in E35 and E36
N205 E35=E35-180
N206 E36=E36-60
N207 (RQO,0,XE35,YE36)     - reapplies origin 0 of X and Y
..........                   in E35 and E36
```

CHECKING DIMENSIONS

a) Checking diameters - The control compares the hole diameter to the nominal value. If the variance is beyond tolerance, the program branches to a label block.

Example 1 - Checking the diameter of a hole

Nominal diameter = 100 +0.02/-0.015

```
..........
"A1" N111
N112 GZ-150
N113  (DIS,"REAMING HOLE D=100")
N114 F..S..T13.13 M6
N115 GX-120 Y80 M13
..........
N129 (DIS,"HOLE TOLERANCE CHECK D=100")
N130 T14.14 M6
N131 GX-120 Y80         -XY positioning on hole centre
M132 Z-85               -Z positioning
N133 G73 r50 E30        -hole radius stored in E32
N134 E32=E32*2
N135  (DIS,E32)
N136 (BGT,E32,100.02,A3)
N137 (BLT,E32,99.985,A4)
N13B GZ150
M139 (DIS,"WORKPIECE IN TOLERANCE")
..........
..........
N2100 M30
"A3" N2101 (TOF,13)     -hole too big
N2102 M00
"A4" N2103 (TOF,13)     -hole too small
/N2104  (BNC,A1)
```

If variance is within the programmed tolerance, the program continues.

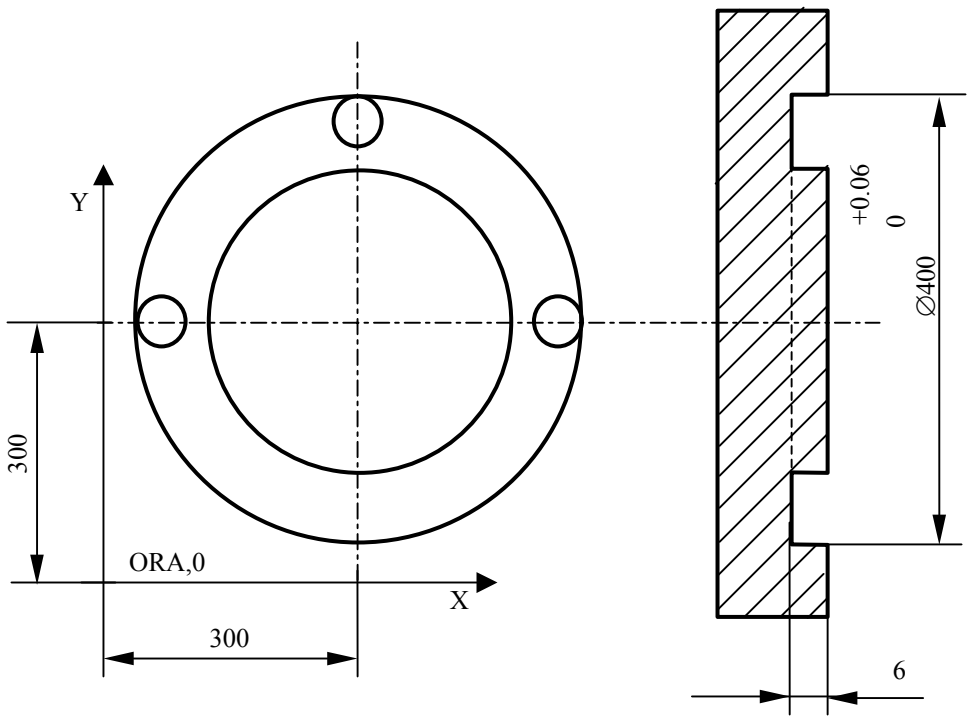
If variance exceeds the programmed tolerance, the "tool condition" (T13) is automatically reset (label A3) and the cycle stops (M00). The workpiece is rejected.

If variance is less than the programmed tolerance, the "tool condition" (T13) is automatically reset (label A4) and the program branches to label A1, which programs the

repetition of the cycle with the alternative tool.
If the alternative tool has not been programmed or is out of useful life, the cycle stops and the corresponding message is displayed.

Example 2

Fig. 5.126. - Checking the diameter of a groove through three points




```
..........
N154 (DIS,"CHECK TOLERANCE D=400")
N155 T16.16 M6
N156 GX496.5 Y300
N157 Z-4
N158 G72 X500 Y300 E31     -1st point in E31 and E32
N159 Z10
N160 X300 Y496.5
N161 Z-4
N162 G72 X300 Y500 E33     -2nd point in E33 and E34
N163 Z10
N164 X103.5 Y300
N165 Z-4
N166 G72 X100 Y300 E35     -3rd point in E35 and E36
N167 Z250
```

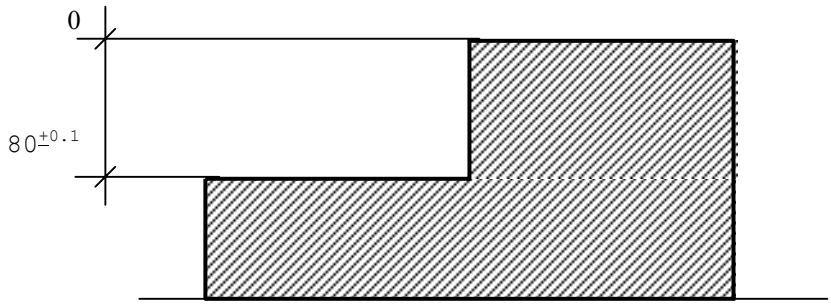
```
N168 p1=XE31 YE32
N169 p2=XE33 YE34
N170 p3=XE35 YE36
N171 c1=p1,p2,p3
N172 E31=FEC(1,3)*2-400   -variance between measured and
N173 (BGT,E31,0.6,B1)      nominal diameter
N174 (BLT,E31,0,B2)
..........

B1=hole too big
BZ=hole too smaLL
```

Notes:

- You can use a similar cycle to check external diameters or diameters of circular sectors.
- To maximize accuracy, the three points should be as far as possible from each other.

b) Checking the coordinates of planes and the depth of holes

Example:



Nominal depth:-80±0.1

```
.........
"C1" N252
N253 (DIS,".....")
N254 F..S..T23.23 M6
..........
N266 (DIS,"CHECK PLANE DIMENSION -80")
N267 T30.30 M6
N268 GX150 Y35         -X-Y positioning on the check point
N269 G72 Z-80 E30      -measured distance stored in E30
N270 E31=ABS(E30)
N271 (BGT,E31,80.1,C2) -branch to C2 if E30<min.coordinate
N272 (BLT,E31,79.9,C3) -branch to C3 if E30>max.coordinate
N273 (DIS,"WORKPIECE IN TOLERANCE")
..........
..........
N2100 M30
```

```
"C2" N2101 (DIS,E30)    -distance too long
N2102 M00
"C3" N2103 (DIS,E30)    -distance too short
N2104 E32=E30-80        -difference between nominal and
                         measured coordinate
N2105 (RQU,23,23,ZE32,K)   -requalifies  length offset 23
(BNC,C1)
```

If variance between nominal and real depth remains within the programmed tolerance, the cycle continues.

If variance exceeds the allowed tolerance, the cycle stops and the piece is rejected.

If variance is less than the allowed tolerance, the tool is requalified and the cycle is repeated from the block containig the C1 label.

TOOL REQUALIFICATION

You can program automatic tool requalification with code RQU and the probing cycle defined with G74.

The values for requalifying the tool are stored in the E parameters defined in the probing cycle.

The allowable format is:

(**RQU,Tool N.,Offset N,ZEn,KEn)**

Important. If you have created the tool offset file to store the current and maximum offset values, the tool offset file has the following format:

Offset number,Z..,K..,c..,m..,c..,m..

where:
c.. c.. are the current offset values for Z (length) and K (diameter)
m.. m.. are the maximum allowed values for Z and K,

The RQU command updates the current offset value (c). The tool is considered unusable if the current value exceeds the maximum allowed value (m).
RQU can only be programmed if the tool to be requalified has been declared in the tool life file.

Example 1 - Requalifying Tool Length Offset

```
..........
N170 G X100 Y100        -positioning on the measuring point
                         (probe position)
N180 G74 Z-50 E30       -deviation measured and value stored
                         in E30
N190 (RQU,10,1,ZE30)    -requalifying offset 1 in length (Z)
                         by E30
```

Example 2 - Requalifying Tool Length and Diameter Offsets

```
..........
N200 G X100 Y100        -positioning on the measuring point
                        (probe position)
N210 G74 Z-50 E30       -deviation measured (Z axis) and
                        value stored in E30
N220 G X150
N230 Z-60               -deviation measured (X axis) and
                        value stored in E31
N240 G74 X130 E31
N250 E31=31*2
N260 (RQU,10,1,ZE30,KE31)   -requalify offset 1 in length
                            (Z) by E30 and in diameter (K) by
                            E31
..........
```

CHECKING TOOL WEAR

When the tool life management feature is present. You can automatically declare a tool unusable because its life is exhausted, or because the sum of its offsets exceeds the maximum allowed value.

In the program, you can declare a tooL unusable, because the deviation measured by the probing cycle (G74) exceeds a defined vaLue. To accomplish this, use conditionaL branches and instructimn TOF (TOF, tool number).
See tool wear checking example on the following pages.

5.20.3. MANAGING ERRORS IN MEASURING CYCLES

The management of some errors during the execution of measuring cycles can be of two types:

1 - Automatic
2 - Via part program

To select either modality, set the boolean ERR variabLe using instruction ERR=1, ERR=0.
The control is configured and reset with ERR=0.
The instructions allowing the double error management are:

1) G72
2) G73
3) G74

Automatic error management

This is the type mf management normally used by the NC-110; it consists in stopping the execution of the part program that generated the error, with display of a message string relating to the error.
For macroinstructions that permit double error management, this modaLity is active when ERR=0.
The variable ERR is automatically set to zero during part program selection and after a RESET.

Error management via part program

If ERR=1, you select error management from part program.
When the control detects a program error, the system variable IOSTA acquires a value related to an error code.
The only errors treated with this modality are:

IOSTA = 1	MEASURING MISMATCH
IOSTA = 2	PROBE NOT RELEASED
IOSTA = 3	INCONSISTENT PROBE INPUT

Part program execution is continued as if the instruction had been carried out correctly, and it is up to the programmer to check the contents of IOSTA from part program in order to test the validity of the blocks.
IOSTA is set to zero if the instruction is executed with no errors.

EXAMPLES: Managing errors via part program testing variable IOSTA

1) Checking tool wear

```
N480  (DIS,"... TOOL D=10")
N490  T10.10 M6
...........
...........
N600  (DPT,10,5,500)
N610  (UAO,90)
N620  G X0 Y0
N630  ERR=1                 - Error managed by part program
N640  G74 Z0 E35            - Measurement of tool length
N650  (BEQ,IOSTA,1,A2)      - Branch to A2 if the tool is
                              broken (if IOSTA=1, because the
                              measurement did not occur
                              within the safety limit of 5 mm
                              declared in instruction DPT)
N660  ERR=0                 - Error managed by the system
N670  .....
...........
N1500 M30
...........
"A2" N2001 (TOF,10)         - Declare tool 10 out-of-use
N2002  (DIS,"TOOL KO")
N2003 M00
```

2) Checking workpiece presence

```
N100  T30.30 M6             - Probe in spindle
N110  (DPT,30,15,500)
N120  G Z50
N130  X80 Y100
N140  ERR=1
N150  G72 Z0 E30
N160  (BEQ,IOSTA,1,END)     - Branch to END if the probe
                              input did not occur (IOSTA=1)
                              within the 15 mm declared in
                              instruction DPT
N170  ERR=0
..........
..........
"END" N1002 (DIS,"WORKPIECE NOT FOUND")
N1003 M00
```

5.20.4. ERROR MESSAGES IN PROBING CYCLES

If the probing cycle is called, but the probe parameters have not been defined in the configuration file or part program, the probing cycle does not start and the following error message is displayed:

PROBING PARAMETERS UNDEFINED

If no contact is made before the seek limit is reached, the following error message is displayed:

MEASURING MISMATCH

If after contact the probe does not reseat, i.e. the status of the input does not change back to the normal state, the following error message is displayed:

PROBE NOT RELEASED

If the probe input is not congruent with the declaration in the characterization file, the following message is displayed:

INCONSISTENT PROBE INPUT

5.21. SYNCHRONOUS BLOCKS

Synchronous blocks are those the control executes only at certain moments or under certain conditions. To activate/deactivate synchronization, use the following symbols:

activates synchronization
& deactivates synchronization

Program these symbols after the number of the block and before the instruction.

Important. By default, the control executes instructions without synchronization. However, synchronization 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 synchronization 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=TIM0              -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 synchronization between calculation and axes move.

5.22. VIRTUAL AXES

To execute profiles on the plane or on cylinders using 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 you activate one of these modalities, the rotary axis is positioned to 0.
You can program the profile with ISO or GTL language, according to the axes (real or virtual) that define the cartesian plane.

When re-converting virtual axes back to real axes, you must re-define the interpolation plane with DPI.

With active virtualization you cannot program origin displacements.

PROGRAMMING WITH MODALITY 1

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

The allowable format is:

(UAV,1,linear real axis,rotary real axis,virtual abscissa axis,virtual ordinate axis,minimum radius)

i.e., (UAV,1,XC,PV,r)

where:
X real linear axis
C real rotary axis
P virtual abscissa axis
V virtual ordinate axis
r minimum radius

The minimum radius establishes the area the tool is not allowed to enter.

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

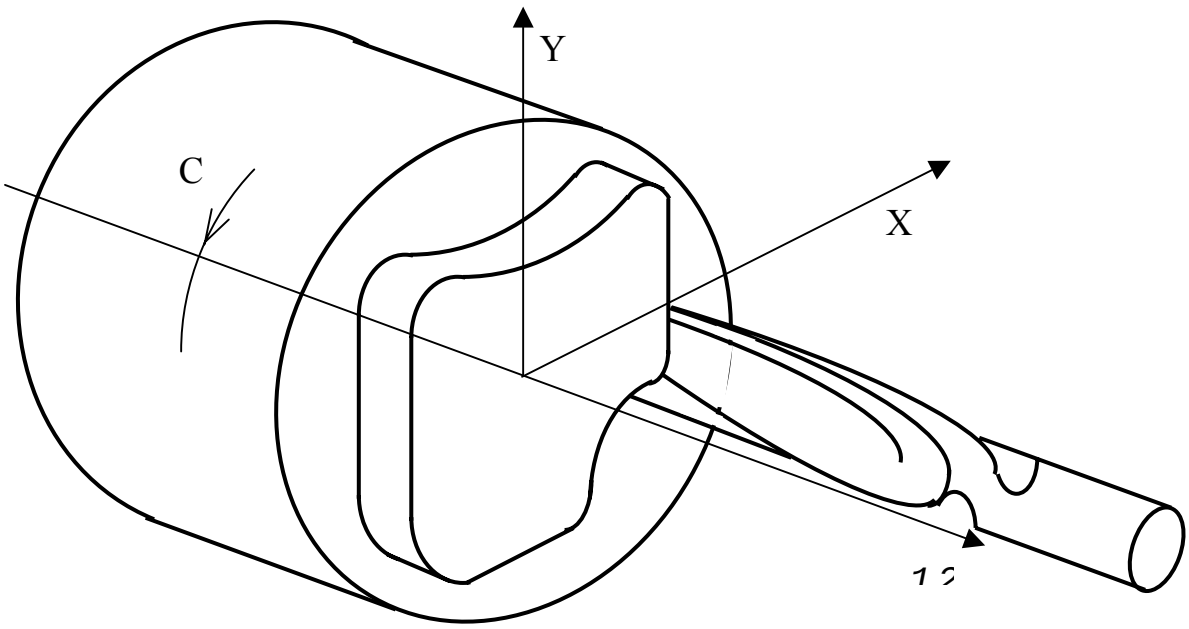
$$r = \frac{F}{Vc\ max} * \frac{360}{2¶}$$

where:

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

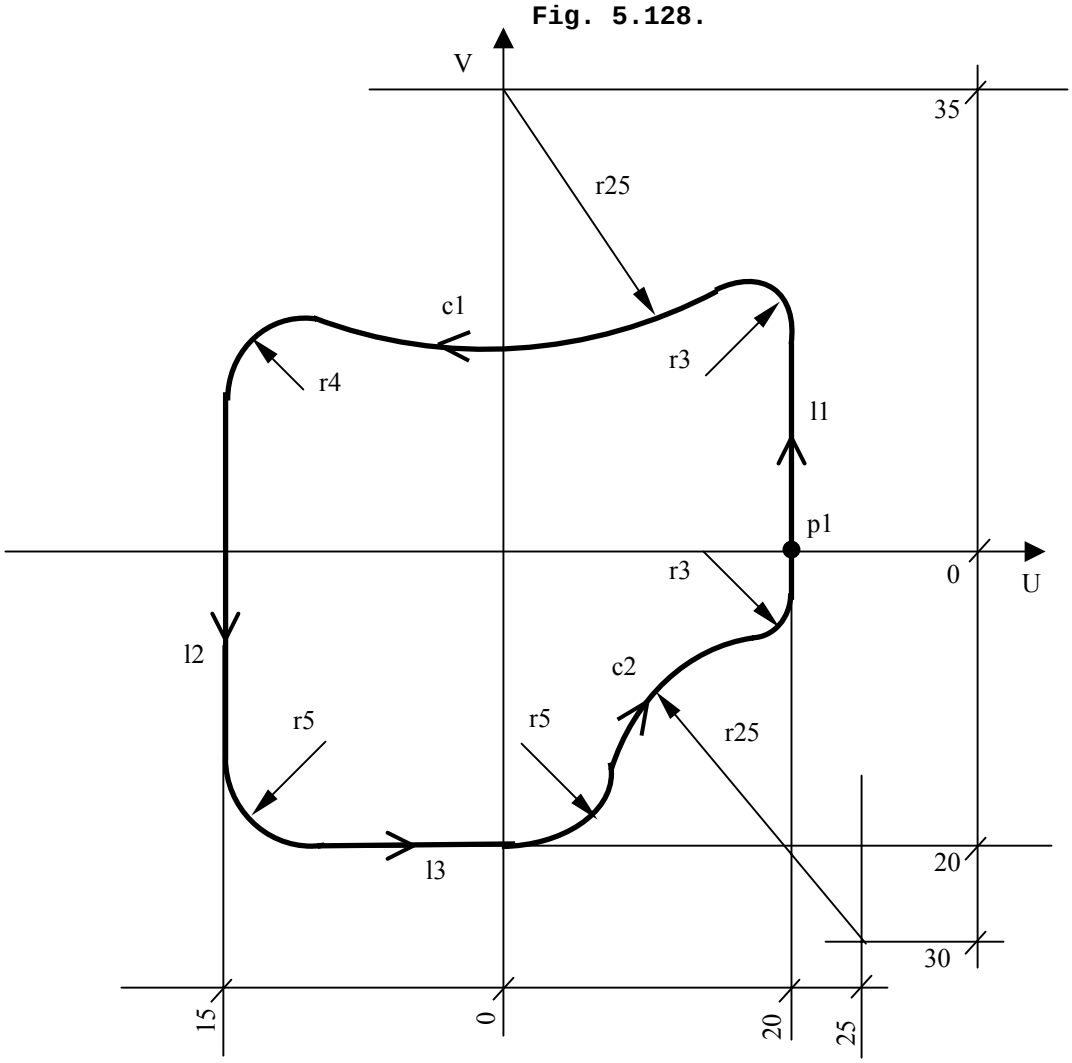
EXAMPLE OF PROGRAMMING WITH MODALITY 1 (Figs. 5.127., 5.128.)

Fig. 5.127.



PROGRAMMING VIRTUAL AXES IN GTL, MODALITY 1

Fig. 5.128.

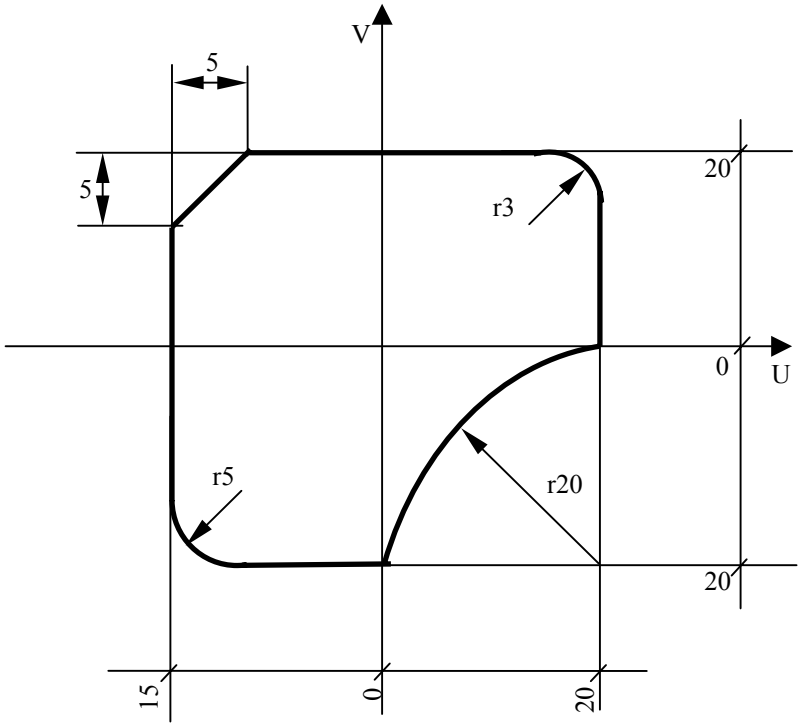



```
N1    T1.1 M6
N2    (DIS,"PROFILE MILLING")
N3    S2000 M3 F300 M8
N4    C0
N5    GX80Y0Z-5
N6    (UAV.1.XC.PV.10)
N7    (DPI,P,V)
N8    p1=P20 V0
N9    l1=p1,a90
N10   c1=I0 J35 r-25
N11   l2=P-15 V0,a-90
N12   l3=P0V-20,a0
N13   c2=I25 J-30 r-25
N14   G21 G42 p1
N15   l1
N16   r3
N17   c1
N18   r4
N19   l2
N20   r5
N21   l3
N22   r5
N23   c2
N24   r3
N25   l1
N26   G40 G20 p1
N27   (UAV,0)
(DPI,X,Y)
N28   G X80
...........
N99   M30
```

PROGRAMMING VIRTUAL AXES IN ISO, MODALITY 1

Fig. 5.128/1

```
%:V1ISO
;VIRTUALIZATION 1 IN ISO PROGR.
T0.1M6
S100M3
E40=110*180/(3.14159*800)
GC0X80Y
(UAV, 1,XC,PW,E40)
(DPI,P,W)
G1G42P20WF110
W20
r3
P-15
b5
W-20
r5
P0
G40G2P20W0I20J-20
(UAV,0)
GX80
M30
%
```



N.B. In this example the minimum radius (r) is calculated directly, by assigning the value to parameter E40 using the following formula:

$$r = \frac{F * 180}{3.14 * Vc\ max}$$

PROGRAMMING WITH MODALITY 2

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

To program a profile, use the following format:

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

i.e., (UAV,2,B,W,r)

where:
B real rmtary axis
W virtual axis
r radius of the cylinder on which the profile is to be executed

EXAMPLE OF PROGRAMMING WITH MODALITY 2 (Figs. 5.129., 5.130.)

Fig. 5.129.

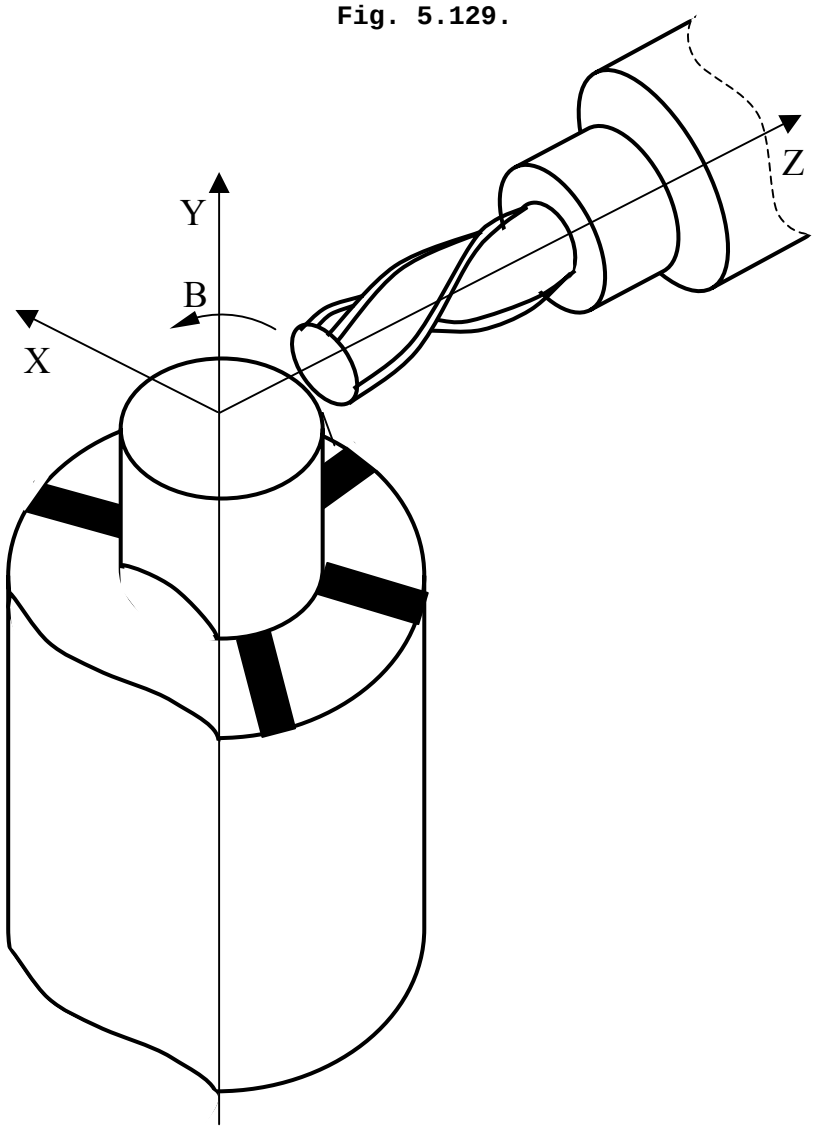
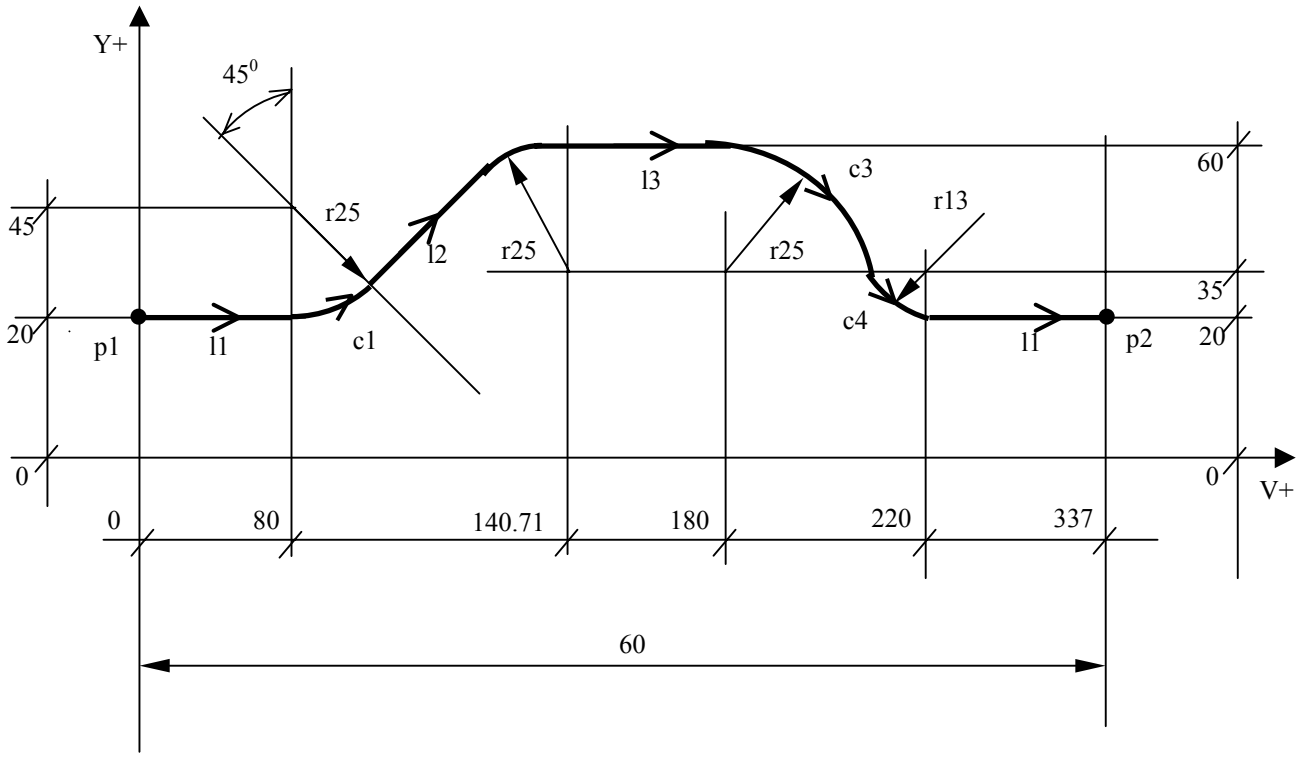


Fig. 5.130.




```
N2   (DIS,"EXAMPLE OF VIRTUAL AXES")
N3   T2.2M6
N4   S2000F300M3
N5   B0
N6   XY20Z10
N7   E30=60
N8   (INP,"CYLINDER RADIUS=",E30)
N9   (UAV,2,B,W,E30)
N10  (DPI,W,Y)
N11  p1=W0Y20
N12  E31=2*3.1415*E30
N13  p2=WE31Y20
N14  l1=p1,p2
N15  c1=I80J45r25
N16  c2=I140,71J35r-25
N17  l2=c1,c2
N18  c3=I180J35r-Z5
N19  l3=c2,c3
N20  c4=c3,l1,r15
N21  G21G41p1
N22  Z-12
N23  l1
N24  c1
N25  l2
N26  c2
N27  l3
N28  c3
N29  c4
N30  l1
N31  G20G40p2
N32  GZ 20
N33  (UAV,0)
N34  M30
```

5.22.1. DUAL AXES

It is possible to define one or several "master" axes to which one or several "slave" axes are associated. In this way, by only programming the master movements, you can obtain the movement of the slaves. In the same command you can also establish mirror machining for the slave.

To enable dual axes, use this command:

(UAV,3,slaves names, masters names, slaves-masters correspondence, mirror machining)

Example:

(UAV,3,VWU,XYZ,123,212)

where:

V W U are the slave axes (1-3)

X Y Z are the master axes (1-4)

123 string that establishes the master-slave correspondence. The value of each figure indicates the master, while the relative position of each figure indicates the slave. In the example, X, Y and Z are the masters of, respectively, V, W and U.

212 string that establishes the motion relationship between master and slave. 2 indicates mirror machining for the slave, whereas 1 indicates normal motion. The relative position of the figure indicates the slave axis. In the example, V and U perform a mirror movement.

To enable dual axes, both the master and the slave must be positioned to zero.

To disable dual axes, use this command:

(UAV,0)

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 [**EMERGENCY**] pushbutton until you hear a click.

3. Press the [**CONTROL ON**] (or [**P1**]) pushbutton.
   The control is supplied with power and starts the autodiagnostics. If no errors are detected, the last executed program appears on the CRT.

   When the message "M.T. OFF" appears, press [**P1**] again. The built-in lamp turns on and power is supplied to the machine tool auxiliaries.

6.2. POWER OFF

When you press (or turn) the [**EMERGENCY**] 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 TNE KEYBOARD

In order to edit or execute a program you must store it in the control's memory. Storying 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 [**P0**] 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 X00 Y50 Z100 Press **[SEND]**
etc.

When you reache the end of the program, press the [**ESCAPE**] key two times: the first tim disables the INS instruction and the second exits the Edit condition anc closes the file.

When vou press the [**ESCAPE**] key the second time, the following message appears on the CRT:

JOB EDI NAME

Press [**P0**] to go back tc the previously active screen.

6.3.2. MODIFYING PROGRAMS

Press [**P0**] to establish Edit.

Enter
EDI,PROG1
Press [**SEND**]

The allowable EDIT operations are:

RIM - modify a block
INS - insert a block
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 l>y 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 hlock. 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 JOB appears in the upper left hand corner of the CRT. For examle:

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.

With COP you can also display a program. For example:

COP,PR77
Press **SEND**

allows you to display program P77 page by page. Each page has seven blocks. To display the subsequent page, press the **LINE FORWARD** key. Yo close the display session, press **LINE FORWARD** when the last page appears on the CRT: the message FILE DISPLAYED will appear.

To set the control in the JOB status, press **RESET.**

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 - Initializing User Memory

For example:

INI,NAME,/MP1
Press **SEND**

This command initializes 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.

DIS - Display a program

For example,

DIS,P34
Press **SEND**

allows you to display program P34 even if it is currently being executed.

To establish the JOB status, press **ESCAPE.**

DIF - Check a stored program

A comparison between a program in memory and a punched tape can be carried out in order to check possible errors due to the external device.
It is also possible to compare two files resident in any storage device (MP/HD/FD) except magnetic cassette, mini-dnc, prom programmer, A-B cassette.

1- Comparison between a program on memory and a tape on external device (TY, PR).
   Instruction: DIF.NAME/DEVICE,/EXTERNAL DEVICE
   E.g.: DIF,PROG1/MP2,/PR
   Press **SEND**

2- Comparison between programs resident in different devices.
   Instruction: DIF,NAME/DEVICE.NAME/DEVICE
   E.g.: DIF,PROG1/MP2,PROG1/HD
   Press **SEND**

WILDCARD

This feature allows to personalize the COP, DIR, DEL and ATT commands by adding an asterisk to the standard format.

For example:

1) COP,AB*/MP2,/HDO
   Press **SEND**
   copies all the files whose names start by AB from the MP2 memory to the HDO memory.

2) COP,*FIL/MP2,/DD0
   Press **SEND**
   copies all the files whose names end by FIL from the MP2 memory to the HD0 memory.

3) COP,AB*F*/HD0,/MP2
   Press **SEND**
   copies all the files whose names start by AB and contain an F character from the HDO memory to the MP2 memory.

4) COP,AB*F*L*/HD0,/MP2
   Press **SEND**
   copies all the files whose names start by AB, contain the F character and end by L from the HDO memory to the MP2 memory.

5) COP,*/MP2,/HD0
   Press **SEND**
   copies all the files stored in the MP2 memory to the HD0 memory. It is equivalent to the following copy command:

   COP,/MP2,/HD0

Notes:

- The file name can be up to 6 alphanumerical characters long. The first character must be a letter.

- When copying files from one memory into another using the asterisk. It is not possible to change the name of the copy file. For instance:

  COP,A*/MP,B*/HD0 generates FILMS1 10 error
  COP,A*/MP,B/HD0 generates FILMS1 14 error

- The above rules also apply to instructions DIR, DEL and ATT.

Recalling program management commands

This feature allows to recall and execute the last eight instructions already entered in the JOB partition using the **LINE SKIP** and **LINE BACK** keys.

Since the search always starts from the last command entered in the JOB partition, it is always necessary to press the **LINE BACK** key first.

After having pressed **LINE BACK**, you can press **LINE SKIP** in order to search for subsequent commands.

To enable the re-called command, press **SEND.**

Example:

```
1)  EDI,NAME1            ┐
2)  COP,NAME1,NAME2      │
3)  DIS,NAME2            │
4)  EDI,NAME3            │  Last 8 commands in the
5)  MOU,/PRO,1,3         │  JOB partition
6)  EDI,NAME4            │
7)  COP,NAME4,/CTO       │
8)  COP,/PRO,NAME 5      ┘
```

To enable the DIS,NAME 2 command, press **LINE BACK** 6 times until DIS,NAME2 appears on the display. To execute DIS,NAME2 press **SEND.**

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 FI1COR, that is stored in memory MP3. FI1COR 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 FORMAT file using the command
   EDI,FORMAT/MP3
   Press **SEND**

3. If the offsets file only contains legth and diameter values, write the following sequence mf characters:

   IA1L3A1L3
   Press **SEND**

   If the offset file also contains information concerning the electronic probe (c..,m..), type in:

   IA1L3A1L3A1L3A1L3A1L3A1L3
   Press **SEND**

4. Exit from Edit by pressing the **ESCAPE** key twice.

5. Create the offset file using the command.

FOR,FI1COR/MP3,xx
Press **SEND**

where xx specifies the number of offsets required for the file.

6. Press **PO** to call for the process display partition (machining screen) and enter:

   REC
   Press **SEND**

   the control replies with CONFIRM? (Y/N)

   Type Y Press **SEND**

   Now, the offset file is ready to be initialized. Refer to section 6.5.

Important. There are two ways of managing tool offsets:

1) By default - In this mode, the offset number corresponds to the record number in the offsets file. For example, in a 3-record offsets file you can only have the following sequence of offsets: 1, 2 and 3.

   1Z+0.000K+0.000
   2Z+0.000K+0.000
   3Z+0.000K+0.000

   During initialization, the record number of the corresponding offset is written with the CAC instruction.

2) Configurable - By setting the CWP code in PGCFIL section *5 (refer to the characterization manual), the offset number can be different from the number of records in the file. This allows to handle a great number of records with a relatively small file. For example, a 3-record offsets file can have offsets 2, 46 and 998.

   2Z+0.000K+0.000
   46Z+0.000K+0.000
   998Z+0.000K+0.000

   With this mode, however, file access time is longer than with default management.

6.4.2. ORIGINS FILE

The origins file is a formatted file called FI1EOR and stored in memory MP3. To create FI1EOR, 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:

   IA1L4A1L4A1L4..

   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,FI1EOR/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:

   CA0
   Press **SEND**

   the control replies with

   CONFIRM? (Y/N)

   Type Y Press **SEND**

   Now, the origins file is ready to be initialized (refer to section 6.10.1).

6.4.3. TOOL LIFE FILE

The tool life management file is a formatted file called GE1TOL 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 FORMAT file using the command
   EDI,FORMAT/MP3
   Press **SEND**

3. Type in the following sequence of characters:
   UAUAUAUAUAR3A
   Press **SEND**

   where:
   U integer format without a sign
   A 1-character ASCII field
   R3 real format with three decimals

4. Exit from Edit by pressing the **ESCAPE** key twice.

5. Create the tool life file using the command
   FOR,GE1TOL/MP3,xx
   Press **SEND**

   where xx specifies the number of tools monitored by the file.

6. Press **PO** to call for 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 initialized by entering a command like:

   VTU,1,T11,[TOOL COMP] 11,t12O,t5,t120,B
   Press **SEND**

Notes:

- Like all formatted files, the TOOL LIFE file can be created from program using the ASSET uti lity. In such a case, the allowable format of the record is:

  IAIAIAIAIA R3A

  where:

  I integer format
  A ASCII format
  R3 real format with three decimals

- It is possible to access the TOOL LIFE file from program using the ASSET uti lity only if the TOOL LIFE file has been created with ASSET.

6.5. STORING AND EDITING TOOL OFFSETS

Before executing a program, you must store tool offsets required by the program. The offsets values can input from keyboard or read from a peripheral.

6.5.1. Storing tool offsets from keyboard

To input tool offsets from keyboard follow this procedure:

1. Specify the address of the tool offset by pressing the **TOOL COMP** key and typing the offset number followed by a comma. For example,

   [TOOL COMP] 1,

2. Next, enter the Z length, a comma and K diameter value, followed by press **SEND**. For example,

   [TOOL COMP] 1, Z-2.15,K20

3. In the ECHO area of the CRT, the following command will be displayed:

   1,Z-2.15,K20

   Press **SEND**

Repeat these steps for each offset.

Notes:

- You can specify the length (Z) and diameter (K) offset values in the alternative measuring unit by entering a - sign before the offset number. For example,

  [TOOL COMP] -2,Z5..,K..
  Press [SEND]

  5mm if the machine is configured in inches, 5" if the machine is configured in mm

- Tool offset compensation can be applied to any axis (Z, X or Y) which is specified by the address associated to the offset value.

- If the offset file contains the current and maximum tool requalifying values (c.. m..), first input the length and diameter. Then display these values by typing **TOOL COMP** and the offset numbers and press **SEND.**

  The offset file for that offset number will appear on line 2 of the CRT.

  Use **BACKSPACE, FORWARD SPACE, DELETE** and ASCII keys to enter the maximum offset values (m's). Press **SEND** to store the m values.
  Values for c must remain at zero. The control updates them after the requalifying command RQU.

  (n,Z..,K..,c..,m..,c..,m..) will be displayed

- When setting offset values, addresses Z and K (or X and K - Y and K) must always be displayed (even when one of them is zero).

6.5.2. Storing tool offsets from peripherals

To store tool offsets from a peripheral, use this command:

COP,name/peripheral,FI1COR/MP3
Press **SEND**

where

COP	is the transfer mnemonics;
name	is the name of the peripheral file (if you are reading from a tape, the file name can be omitted);
peripheral	is the name of the peripheral device (refer to Table 5.2);
FI1COR	is the name of the file to be loaded (already formatted and created with this name);
MP3	is the device on which the offsets file will be stored

The writing format of the support is either:

```
% 1 Z.0K.0                  -record with Z=0 and K=0
  2 Z200.0K25.0             -record with Z=200 and K=25
%
```

or

```
% 1 Z.0K.0c.0m.0c.0m.0
  2 Z200.0K25.0c.0m.0c.0m.0
%
```

if the file has been created to store c and m.

Notes:

You must write all the records of the tool offsets file written with the above format even when all the addresses are zero. For instance, in order to store offsets 2, 4 and 7, write the following records:

```
% 1 Z.0 K.0
  2 Z210.0K.0
  3 Z.0K.0
  4 Z180.0 K20.0
  5 Z.0 K.0
  6 Z.0 K.0
  7 Z190.0 K18.0
```

Example:

To transfer the tool offsets file from tape via teletype, use this command:

COP,/TY1,FI1COR/MP3
Press **SEND** (the control will ask for confirmation)
Enter Y
Press **SEND**

To transfer the offsets file using a tape reader, substitute PR for TY1.

To transfer the file from the control memory to a peripheral, use the procedure described in Chapter 7.

6.5.3. DISPLAYING AND MODIFYING TOOL OFFSETS

To display the value of a stored offset, enter

[TOOL COMP] 3
Press **SEND**

where 3 is the offset number. The control displays the values in the echo area of the CRT. For example,

[TOOL COMP] 3,Z5.612,K20.2

Once the offset is displayed, there are two ways to modify it:

- Clear the offset with the RESET key and enter the new value. For example,

  [TOOL COMP] 3,Z4.37,K20.2
  Press **SEND**

- Use **FORWARD SPACE** and **BACKSPACE** keys to position the cursor on the right of the character you want to modify. Cancel the character with the **DELETE** key and type in the new character. Repeat these steps for each character to be modified.

Important. When you finished mmdifying the offset values, press **SEND** to store the new offsets.

Notes:

- You can display and modify tool offests while machining, but the new offset will only be enabled when the control reads a new T function followed by M6.

- The difference between the offset old values and the newly stored values cannot exceed the threshold established during configuration.
  The default value for this threshold is 1.0001.
  If you must enter a difference that is greater than the configured value, delete the offset and store a new value.

  Use the following procedure:

  1. Press **PO** to select the process display partition.

  2. Enter

     CAC,n
     Press **SEND**

     to cancel a given offset or

     CAC
     Press **SEND**

     to cancel the offset file. The control replies with:

     CONFIRM? (Y/N)

     Type Y press **SEND** or N press **SEND.**

To display the entire tool offsets file on the EDP partition (which you can obtain by pressing the P0 key), key in the following command:

DIS,FI1COR/MP3
Press **SEND**

6.5.4. INCREMENTAL TOOL OFFSET MODIFICATION

Tool offsets can also be modified in incremental mode using code UCA and typing directly the offset value, that will be algebraically summed to the offset dimension.
To modify input:

UCA, offset n., axis name +/- length offset value, K +/- diameter offset value

Example:

UCA,3,Z-0.02
Press **SEND**

There are two possibilities:

1) Stored offset modification
   If the offset to be modified is not the current one and the machine is at a standstill (but not in Hold), the modification will be stored and then applied when the offset is called up by an M6.

2) Active offset modification
   If the offset to be modified is the active one (i.e. the displayed one) and the machine is at a standstill but not in HOLD status, the modification is stored and applied immediately. It is not necessary to re-call it with an M6.

N.B. The system checks that the modification does not exceed the value declared during configuration (default value = 1 mm).

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 doea 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. Table updating takes place at the end of each tool machining cycle.

You can enter and edit tool life data from the keyboard with the system in process display partition (machining screen). Use the following operations:

1) Initialize the Tool Life File

For each record, the file has the following format:

VTU,tool number, T field 2, [TOOLCOMP] 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 (declared with TOF)
B = usable tool (whose life time is managed)
C = usable tool (whose condition is managed by adding corrections and with G74)
D = tool out of useful life
E = usable tool on spindle (originally B). Since partially used, it will have the highest priority
F = usable tool currently on spindle (originally C) to be managed by adding corrections. Since partially used, it will have the highest priority.

For example, you can initialize a tool life file by inserting a record like:

VTU,2,T12, [TOOL GOMP] 12,t60,t2,t60,B
Press **SEND**

Notes:

- In the above example, the alternative tool (i.e. T12) must be declared in the table as follows:
  VTU,12,T12,[TOOL COMP] 12,t60,t2,t60,B
  Press **SEND**

- Even if a tool has not an actual alternative tool, it must be initialized as if it had one. The allowable modalities are:

  1) tool declared as alternative to itself. Example:
     VTU,5,T5,[TOOL COMP]5,t60,t2,t60,B

     When the usable life of the main tool is over, the following locking error will be displayed: FILMS4 72

  2) tool declared as alternative to TO. Example:
     VTU,5,T5,[TOOL COMP]5,t60,t2,t60,B

     When the usable life of the main tool is over, the system communicates it to the machine logic via interface (by means of the FUTKO strobe).

2. Display and Edit a Record in the Tool Life File

   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 _i.

3. Delete a Tool from the Tool Life File

   CTU,n -cancels tool n
   Press **SEND**

4. Delete Tool Life File

   CTU -cancels the tool life file
   Press **SEND**

5. Display Tool Life File
   To display the entire tool life file on the EDP partition (selectable with the P0 key), key in the following command:
   DIS,GE1TOL/MP3
   Press **SEND**

- You can update the condition of tools out of useful life in the table even when not all of the tools in the family are in that condition. The system will automatically seletcs the tools that are still usable (in E or F condition).

- If K buffer signal ABTVU is set to 1 by the machine logic or from program (bit 340 with SK340.BL=1), the lifetime of the tool mounted on the spindle will be counted even if spindle rotation and machining rate are not active. The tool life count will remain active as long as bit 340 remains high.

- If the machining time to be counted is programmed in dynamic mode (with G27/G28), the motion function (G01, G02 or G03) must be programmed in the block with the profile start coordinates.

6.7. RANDOM TOOL MANAGEMENT

To enable random tool management, you must first create a formatted file (typically called FI1RAN) using the CRE instruction with the control in the EDP mode selectable with the P0 key. The allowable format is:

CRE,FI1RAN/MPO,4,xx
Press **SEND**

where:

4 (mandatory) is a 4-digit record format
xx number of pockets in the turret

Then you must edit FI1RAN, i.e. declare the tools managed by each record.

Each records corresponds sequentially to a tool station. When a tool updating is requested, the interface associates each tool with the corresponding station.

For both "normal" and "special" tools (special tools are described in the note), the format has always 4 digits. For example:

```
0001═══════════════════════════════╗
0002═══════════════════════════════╗
9003═══╗                          ╚═══════› normal tools
9003   ╚═══════› special tool      ╚
9003═══╝                          ╚
0004═══════════════════════════════╝
```

If random management is active, the interface checks whether the programmed T is valid and, if necessary, looks for a tool fixture. In case of manual tool change, the programmed T does not require an automatic search for the tool fixture.

Note on special tools. Special tools occupy 3 stations. They are identified by the 9000 code.

6.8. 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 hmme 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:

1. Move the mode selector to the Automatic Home Position (7).

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.

6.8.1. JOGGING

There are two ways of jogging the machine axis:

- continuous jog;
- incremental jog.

a) **Continuous jogging**

1. Move the mode selector to the CONT JOG position (4).

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. Move the mode selector to the Incremental Jog position (5).

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. Select continuous jog or incremental jog mode.
2. Press **LINE FORWARD** or **LINE BACK** to select the axis to be moved.
3. At each complete turn of the handwheel, the selected axis moves as follows:

characterized unit	continuous jog	incremental jog
mm	1 mm	0.1 mm
with G70	0.039"	0.0039"
inch	0.1"	0.01"
with G71	2.54 mm	0.254 mm

Notes:

- With diameter axes. The above distances double.
- If the axis reaches a software travel limit, it stops but no message appears on the CRT.

6.10. AXES ORIGINS

Before you start machininq a workpiece, you must define the part zero with respect to the absolute machine zero. Ahsolute machine zero is a system of coordinates in which each axis has its origin on the home position. In order to define the part zero, you must store the distaruce between part zero and absolute machine zero, for each axis, in the file of origins. Follow these steps:

1. Fit a fixture in the nachine, and fit each tool that must be referenced. Position each axis to the reference point;

2. Define the absolute origin for each axis.

6.10.1. ABSOLUTE ORIGINS

Each absmlute origin can be assigned to 1 to 3 axes. You can define as many as 100 (0÷99) absolute origins.

Absolute origins are contained in the file of origins that you created and stored in the control memory (usually MP3).

To define the absolute origins for each axis, type the following command:

ORA,n,X.. - assigns 1 axis to each origin
Press **SEND**

or

ORA,n,X..,Y..,Z.. -assigns 3 axes to each origin
Press **SEND**

You can modify, display and delete these origins.

DEFINING ABSOLUTE ORIGINS

Home the axes (limit switches), Jog the axes to a known reference point. Then enter the definition command from keyboard with the following format:

ORA,origin number,X..,Y..,..
Press **SEND**

where

X..,Y..,.., is the position of the specified with respect to respect to the current position of the axes

Fig. 6.1. - Reference point coinciding with the origin

Enter ORA,0,X,Y
Press **SEND**

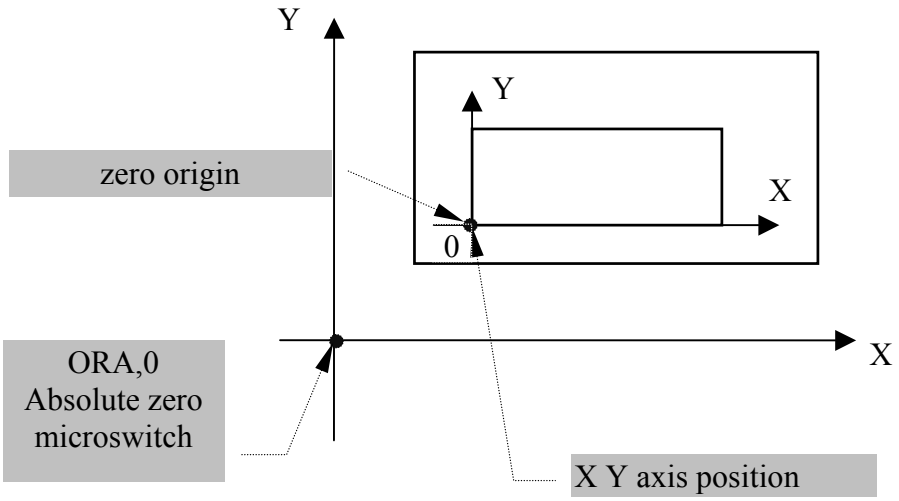
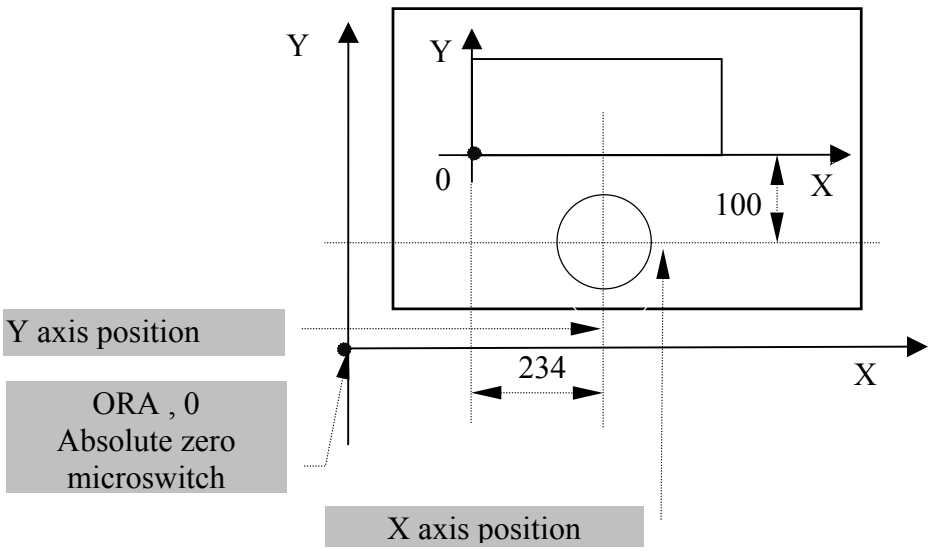


Fig. 6.2. - Reference point not coinciding with the origin

Enter ORA,0,X-234,Y100
Press **SEND**



Notes:

- If you do not define an axis origin as an absolute origin, the control continues to use the absolute origin enabled during the last home operation.
- If you use a reference fixture, the definition of the absolute origin must only be performed for the first workpiece.
- After power-on, for each homed axis, the control enables the absolute origin that you have previously defined. The coordinates displayed on the CRT are referenced to this point.
- To call for define oriqins from inside a program, use the code UAO.
- If no absolute origin is called in a program, the control automatically enables origin 0.
- To express absolute origins in the alternative measuring unit, specify thie origin number with the negative sign. **For example:**

  ORA,-1,X10 -10" if the machine is configured in mm;
  10mm if the machine is configured in in.

MODIFYING ABSOLUTE ORIGINS

After defininn and storing absolute origins, you can modify the origins in one of the following ways:

– specifying new values. **For example:**
  ORA,O,X..,Y..,..
  Press **SEND**

– entering the keyboard command
  ORA,origin numher
  Press **SEND**

  This displays the distance from the current position of the axes to the specified absolute origin. Modify the displayed dimensions and then store the result with press **SEND**

Important. Defining and modifying origins musr not he performed if **START** and **HOLD** pushbuttons are on.

DISPLAYING ABSOLUTE ORIGINS

To display the distance of the absolute origin from the machine zero, enter the following command:

VOA,n,
Pres **SEND**

Example:

VOA,5 displays VOA,5,X878,25,Y12,127
Press **SEND**

With the EDP screen on the CRT, you can display the entire origins file by entering the following command:

EDI,FILEOR/MP3
Press **SEND**

DELETING ABSOLUTE ORIGINS

To delete a given origin, enter:

CAO,origin number
Press **SEND**

To delete all the stored origins, enter:
CAO
Press **SEND**

6.11. ZEROING THE Z AXIS AND ESTABLISHING TOOL LENGTH OFFSETS

You can set the zero point for the Z axis and establish tool length compensations by using three methods:

- preset tool on the workpiece;
- set tool offset to zero;
- include tool length in tool offset.

To reset the Z axis wwith tool presetting on the machine, use the following procedure:

1. Cancel any existing current tool offset from keyboard.
   For example: TM6.

2. Jog the axes to the absolute zero points.

3. set the origin of the Z axis. For example: ORA,0,z
   Press **SEND**

4. Fit the tool to be preset on the machine.

5. Jog the Z axis skimming the piece surface, up to the point where the origin of the Z axis must be set.

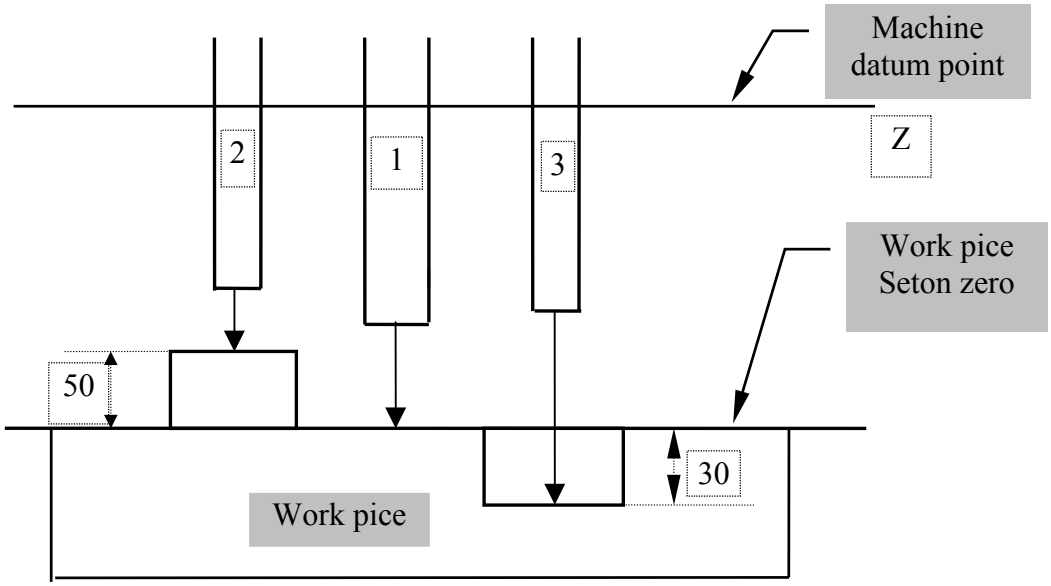
6. Enter the following instruction:
   [TOOL LENGTH] 1,Z0
   Press **SEND**

   The control automatically calculates the value of the tool compensation which relates the absolute machine zero with part data, considering the lengths of the tool.

7. Repeat steps 4, 5, 6 for all tools.

   To preset the tool during machining, you must repeat steps 4, 5, 6 again.

Fig. 6.3. - Presetting Tools on the Workpiece



Enter:

[TOOL LENGTH] 1,Z0
Press **[SEND]**
[TOOL LENGTH] 2,Z50
Press **[SEND]**
[TOOL LENGTH] 3,Z-30
Press **[SEND]**

In order to set tool offsets equal to zero, you must know:

- the exact length of each tool;

- the distance between the piece and the longest tool you want to use;

Use the following procedure:

1. Home the Z axis.

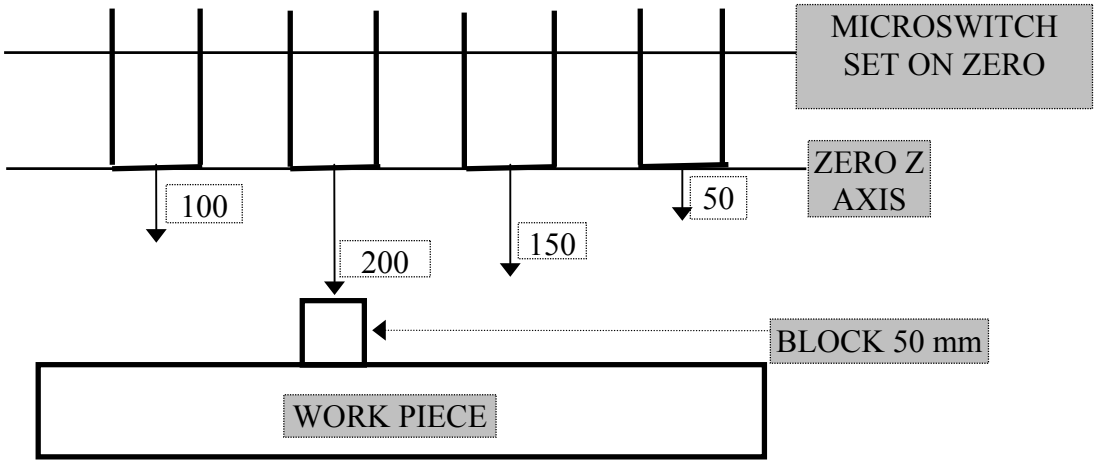
2. Establish a reference height above the surface of the workpiece that allows you to change the longest tool. For example, use a reference block 50 mm thick.

3. Set the Z axis on the absolute origin. Activate the absolute origin with the following command:

   ORA,0,Z
   Press **SEND**

You must program the Z axis with different dimensions for each tool. With tool offsets set to zero you can use offsets to compensate for the difference in length between actual and theoretical tools.

Fig. 6.4. – Setting Tool Offsets to Zero



If you include tool length in offsets, use the following procedure:

1. Home the Z axis.

2. Load a tool whose length is known and has already been stored. Activate the length offset (with Tn.m M6).

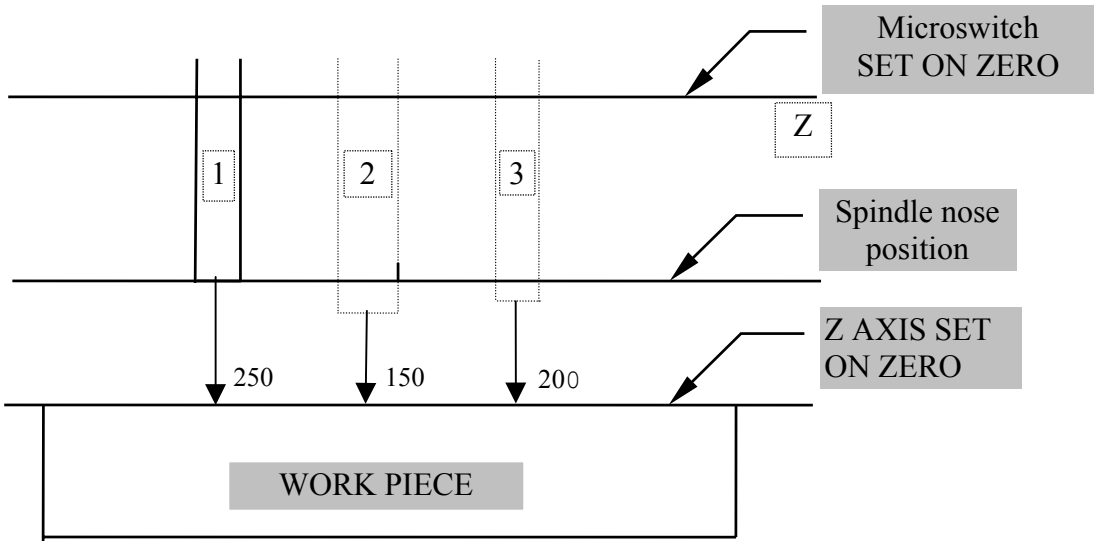
3. Jog the Z axis to the point where the tool tip skims the surface of the piece where the Z zero point should be set.

4. Reference the Z axis to the absolute origin. Activate the absolute origin with the following command:

   ORA,O,Z
   Press **SEND**

Important. Store the length of each tool in the tool offset table.

Fig. 6.5. - Including Tool Length in Tool Offset



In the example, store the length values by using these commands:

[TOOL COMP] 1, Z250, K..
Press **SEND**
[TOOL COMP] 2, Z150, K..
Press **SEND**
[TOOL COMP] 3, Z200, K..
Press **SEND**

6.12. ZEROING DIAMETER AXES (REAMING/FACING HEADS)

Before you start machining, you must define the zero point for the diameter axis (U). Use the following procedure:

1. Using the keyboard, input the X and Y coordinates of a hole to be reamed.

2. Home the U axis.

3. Jog the U axis to the position required for reaming. The specific diameter position is not important.

4. Enable the reaming execution for a hole. For example, use the following command:

   G81 R..Z..F..S..M13
   2X.. Y..

5. If the hole has no witness marks, remove the tool from the hole without moving the U axis.

6. Measure the exact hole dimensions. For example: 0=98.73 mm

7. Define the origin for the U axis by entering this command from the keyboard:

   ORA,n,U-measured diameter
   Press **SEND**

   For example: ORA,O,U-98.73
   Press **SEND**

If the tool exceeds the allowable tolerances during machining, update the U axis origin by entering the diameter of the last reamed hole. For example: ORA,0,-119.95
Press **SEND**

Important. Do not move the tool radially from the diameter of the reamed hole.

6.13. EXECUTING A PROGRAM

1. Activate the program for execution with the SPG command.
   For example:

   SPG,PROG1 -selects program PROG1 from memory
   Press **SEND**

2. Move the mode selector to the AUTO or BLOCK-BY-BLOCK position.

3. Press the **START** pushbutton.

Notes:

- If you select BLOCK BY BLOCK 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 **SEND**
  For example:

  ESE,22 -the program stops when block 22 has been
  Press **[SEND]** executed. To resume execution, press **START**

ESE cannot be programmed inside a continuous profile (G27-G28).

6.14. PROGRAM TESTING

You can test programs by using 3 methods:

- axes standstill program test;
- dry rur 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 (shown in Streen 1).
Press **SEND**

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. Move the mode selector to the AUTO or BLOCK-BY-BLOCK position.

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,X..,X..,Y..,Y.. -defines the workpiece area and the plane you want to display.
Press **SEND**

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 methodl 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. Move the mode selector to position 2 (Auto) or 3 (Semiauto)

4. Press the **[START**] pushbutton.

In SEMIAUTO, you must press **[START]** to execute each block.
You can vary the exedution 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 progran to be tested:

   SPG,TEST
   Press **[SEND]**

2. Input the command URL=1 press **[SEND],** which appears on display 1 only.

3. Select the semiauto mode by moving the mode selector to position 3.

4. 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 nachine. 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 (selectmr S4 on position 1).

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. 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 nust repeat a sequence of operations after checkinq the piece or adjusting the tool offset, search for the starting block. If you nodified offsets, search for the tool change block (refer to section 6.15.).

6.15. 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)

1. Enter RAP=0 press **[SEND]** from the keyboard.
2. Move the mode selector to the JOG RETURN position.
3. Select the desired axis with **[LINE FORWARD]** or **[LINE BACK].**
4. Set the JOG potentiometer to the desired rate and direction for the returning axis.
5. Press and hold the **[START]** pushbutton. 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 controlautomatically retraces in reverse order the sequence of manual moves performed during jogging. The control stores an many as 32 jog moves.

1. Enter RAP=1 press **[SEND]** from the keyboard.
2. Move the mode selector to the JOG RETURN position.
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.

6.16. MULTIBLOCK RETRACE

This feature allows you retrace the profile up to a given position. Yhe 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].**

6.17. 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 characterize 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 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].**

   Alternatively, you can press **[CYCLE START]** until the interrupt block appears on the CRT and then enter RCM press **[SEND].**

AUTOMATIC SEARCH OF A GIVEN BLOCK

Use the following procedure:

1. With the mode selector 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. Input ERM
   Press **[SEND]**

To search the block where the interruption took place, it is possible, before entering instruction ERM press **[SEND],** to have the search go automatically forward by one or more blocks, pressing **CYCLE START** until the desired block is displayed.

When searching a given block, it is possible, before entering instruction ERM press **[SEND],** to bring the search to block n inputting ESE,n press **[SEND]** and then pressing **[START].**
Now, to restart the machining cycle, after entering instruction ERM, you must:

1. Press **(START).**
   The system outputs the auxiliary functions and goes into Hold status. The auxiliary functions are performed in the following order: S, T in spindle, indexing axes, M/H by class of search, programmed T.

2. Jog the axes from the stop points.

3. Select the Auto or Semiauto mode.

4. Press **(HOLD)**. The Hold indicator turns OFF.

5. Press **(START)** to resume execution.

Example:

The listing that follows illustrates a profile made up of 8 parts repeated 8 times with 2 cuts each. Execution re-starts from block N170 (second repeat of the second cut).

The procedure is as follows:

1. Select automatic mode.

2. Enter **RCM**
   Press [SEND]

3. Enter **ESE,330** (This EPP block recalls the second cut)
   Press **[SEND]**

4. Press **[CYCLE START]**

5. Execute the INP blocks. When you press **SEND**, the control emulates execution up to block N330.

6. Enter **ESE,280**
   Press **[SEND]**

7. Press **[CYCLE START].** The control emulates the first repeat.

8. Enter **ERM** to close the memorized search.
   Press **[SEND]**

```
N10   (DIS, "WHEEL WITH ASSET")
N11   E = 8
N12   (INP, "N OF RADIUS=",E)
N13   E30=360/E
N14   E31=100
N15   (INP,"WHEEL DIAMETER=",E31)
N16   E31=E31/2
N17   E32=20
N18   (INP,"RADIUS DIAMETER=",E32)
N19   E32=E32/2
N30   F1000S1000T2.2M6
N50   p1=XE31Y
N60   c1=IJrE31
N70   c2=IJrE32
N80   l1=c2,a180
N90   l3=XY,aE30
N100  l2=c2,aE30
N110  p2=l3,c1,s2
N111  E33=E31+10
N112  E34=NEG(E33)
N113  (UCG,2,XE34XE33,YE34YE33,Z)
N120  XE33Y
N130  Z-20
N131  UOV=2
"IN"
N150  E25=0
N160  (RPT,E)
N170  (URT,E25)
N180  G21G42P1
N190  c1
N200  r3
N210  l1
N220  r-3
N230  l2s2
N240  r3
N250  c1
N260  G20G40p2
N270  E25=E25+E30
N280  (ERP)
N290  (URT,)
N300  Z
"OUT"
N320  UOV=0
N330  (EPP,IN,OUT)
N340  GXE33Y
N350  M30
```

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 RCM for the interrupt block after an REL (=program release) command. After REL you can only activate RCM for a given block.

- Automatic search for the interrupt block is only allowable if cycle execution has started from the beginning of the program or after an automatic search. You cannot use automatic search if execution has been activated from the keyboard.

- [RESET] does not disable an automatic search status.

- No automatic search can be carried out on virtual axes.

- The E parameters, receiving a value in measuring cycles, are set to zero by the "disconnected axes" condition forced by an automatic search; if necessary, the operator shall restore the measured values before resuming the cycle.

6.18. 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.

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 contrml 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 fourth 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 fourth 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.

6.19. 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 correections 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.

6.20. EXECUTING FROM THE KEYBOARD

To input and execute data from the keyboard:

1. Select the Input by keayboard mode (with mode selector in position 1).
2. Enter the block you want tm execute. For example,

   G1 X20 Y10.3 F500

   The block will appear on the second line of the CRT.
3. Press **[START].**

6.21. USING RESET

The reset function stops axes movement, spindle rotation and coolant flow. It clears the control's buffer except tool offsets, restores the zero absolute origin and causes a tape rewind.

To use the reset function:

1. Select the Reset mode (with mode selector in position 8).

2. Press **[START].**

6.22. USING GRAPHICS

The graphic page allows you to show:

- theoretical profiles and profiles translated by the cutter radhus
- 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,[AXIS3]
Press **[SEND]**

where:

n defines the display mode
n = 1 display not coordhnated 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

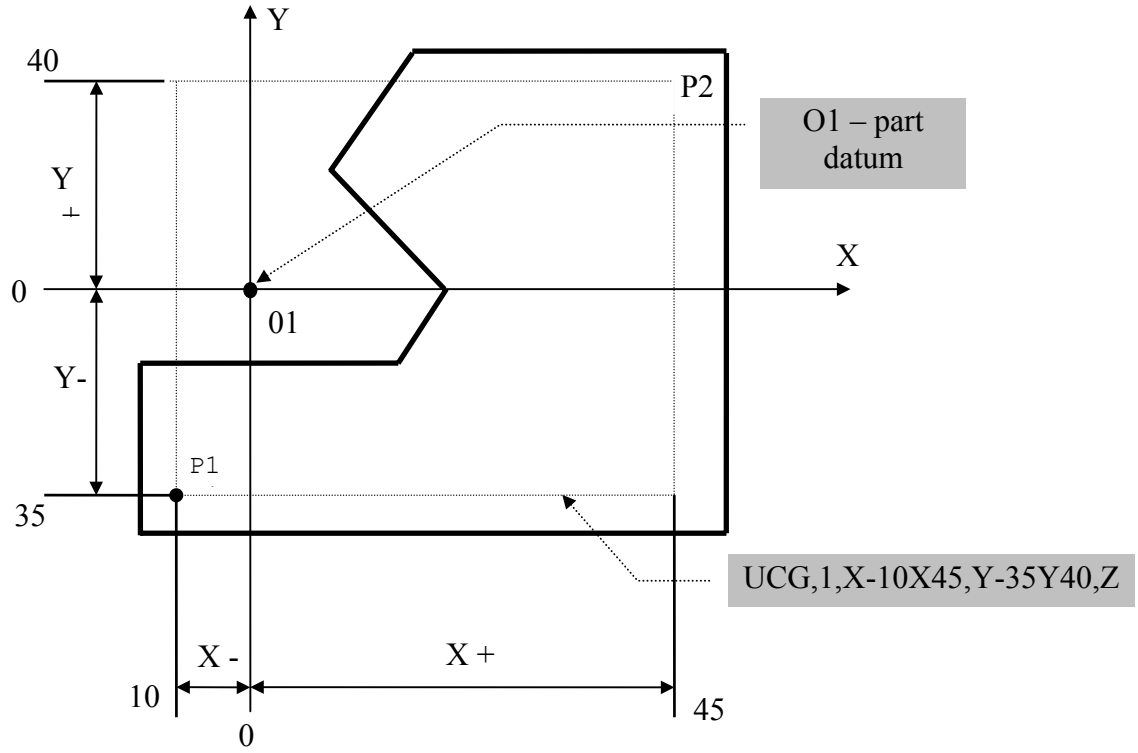
AXIS2I defines the name and the lower limit of the display for the ordinate axis

AXIS2S defines the name and the upper limit of the display for the ordinate axis

AXIS3 defines the nane of the third axis normal to axis 1 and axis 2. The third axis is used during fixed cycles and can be omitted. Motion of this axis will be indicated by a special grahics symbol on the profile, at the intersection points of the three axes.

The lowest field limit is 1. The graphic display considers the absolute, temporary or incremental origins.
The graphic field must be defined after selecting any origin.

Fig. 6.6. - Definition of the graphic field



UCG,1,X-10 X45,Y-35 Y40,Z
Press **[SEND]**

The graphic field is defined with respect to the active origin.
The zero point in the graphic field corresponds to the zero defined in the program.
The name of the axes can be changed according to the programmed interpolation plane.

Example:

UCG,1,X-100 X100,Z0 Z50,Y
Press **[SEND]**

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,X..X..,Y..Y..,Z -defines graphic field limits with display not coordinated with the axes
   Press **[SEND]**

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 prformed 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 cancell 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,X..X..,Y..Y..,Z -defines the limits of the graphic field with the display coordinated to the axes moves
Press **[SEND]**

The graphic display shows the aachining 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. PERIPHERALS

Chapter Objectives

This chapter tells you how to use peripheral devices connected to your NC-110.

We discuss the use of the following devices:

- TEAC magnetic cassette (7.1.)
- printer (7.2.)
- CRT (7.3.)
- teletype (7.4.)
- serial tape reader (7.5.)
- serial tape punch (7.6.)
- parallel devices (7.7.)
- M24 personal computer (7.8.)

Important: You can perform input/output operations using serial or parallel devices under any normal system conditions, including moving axes.
Before starting an input/output operation, press **[P0]** to establish the Edit mode.
If the serial line is used by an ASSET program, a possible transfer request in J0B status generates error FILMS3 29. and the command is aborted.
If command MOU (peripheral driver initialization) is activated, error FILMS1 14 is generated.

7.1. TEAC_MAGNETIC_CASSETTE

You can use this device for program storage onto/from the user memory.

The allowable instructions are:

COP,PROG1,/CT Press **[SEND]**	loads file PROG1 onto the cassette under the name of PROG1
COP,PROG1,PROG2/CT Press **[SEND]**	loads PROG1 onto the cassette, under the name of PROG2

COP,PROG1/CT,PROG1 Press **[SEND]**	loads file PROG1 from cassette onto user memory under the name of PROG1
COP,PROG1/CT,PROG2 Press **[SEND]**	loads file PROG1 from cassette onto user memory under the name of PROG2
DIR,/CT Press **[SEND]**	displays the directory of the magnetic cassette (name and length of the files and number of free sectors)
INI,NAME,/CT Press **[SEND]**	initializes a new cassette. The name can be up to 6 characters long. The first character must be a letter.
COP,PROG1/CT Press **[SEND]**	displays a program stored in the cassette on the CRT.

Important: before performing any operation with the cassette, check that the cassette drive is ON and select a baud rate of 1200.

7.2. PRINTER

If your control is connected to a printer, use the following commands:

COP,PROG1,/LP Press **[SEND]**	lists the file stored in the user memory under the name of PROG1.
COP,PROG1/CT,/LP Press **[SEND]**	lists the file stored in the magnetic under the name of PROG1.

7.3. CRT

The CRT is considered the default peripheral device of the control. Use the following commands:

COP,PROG1 Press **[SEND]**	displays the listing of program PROG1 stored in user memory

COP,PROG1/CT Press **[SEND]**	displays the listing mf program PROG1 stored in the cassette.

Each page of the listing is 7 blocks long. To see subsequent pages, press **[LINE FORWARD].**

To interrupt the display, press **[RESET].**

If a file is left open (i.e., power off while the file is being edited), you can close it after power up by using this command.

7.4. TELETAPE

If your control is connected to a teletype, you can use the following commands:

COP,PROG3,/TY1 Press **[SEND]**	punches program PROG3 on the teletype and prints the listing
COP,/TY1,PROG4 Press **[SEND]**	read the program from the tape and store it on memory as PROG4.
COP,/TY1 Press **[SEND]**	displays the program read from the tape. Each page is 7 blocks long. To display subsequent pages, press **[LINE FORWARD].**
COP,PROG5,/TY0 Press **[SEND]**	lists PROG5 on the teletype printer.

7.5. SERIAL_TAPE_READER

If your control is connected to a serial tape reader, use the following command:

COP,/PR,PROG1 Press **[SEND]**	reads the tape from the reader and stores the program in memory under the name of PROG1.

If there is a parity error or the character is not recognized, at the end of the transfer session the following message will be displayed:

TRANSFER ERROR

Program errors to be corrected will be displayed as question marks (?).

7.6. SERIAL TAPE PUNCH

To punch a program with a serial tape punch, use the following command:

COP,PROG3,/PP
Press **[SEND]**

Section 4 of the system characterization file FCRSYS usually specifies the ISO standard. To punch a program with a different standard, enter the following commands:

COP,PROG4,EIA/PP selects EIA standard
COP,PROG5,ASCII/PP selects ASCII standard

7.7. USING PARALLEL PERIPHERALS

A parallel/serial adapter is available to connect your control to any serial peripheral devices, such as the ADDMASTER portable reader and the FACIT 4070 punch.

Before using parallel devices, you must initialize them by using the following commands:

MOU,/PR,1,2 for tape reader
Press **[SEND]**
MOU,/PP,1,2 for punch
Press **[SEND]**

The allowable peripherals and the configuration of devices (baud rate, terminal, etc.) must be specified in the system characterization file FCRSYS.

Once initialized, parallel peripheral devices can be managed with the commands used for serial devices.

Sometimes, due to erroneus commands, communication with a peripheral device must be reactivated by initializing the peripheral driver. You can follow two alternative proceddures:

- switch the control off and then back on, or

- enter an initialization command using the keyboard. The allowable format is:

  MOU,PERIPHERAL NAME,BOARD No.,BAUD RATE
  Press **[SEND]**

For example:

MOU,/TY1,0.1 Press **[SEND]**	for teletype
MOU,/PR,1,2 Press **[SEND]**	for serial or parallel tape reader
MOU,/PP,1.2 Press **[SEND]**	for serial or parallel punch
MOU,/LP,1.3 Press **[SEND]**	for printer

7.8. USING_THE_M24_PERSONAL_COMPUTER

The COMUN, applicable to release 1.0 of the NC-110, you can connect your control to the M24 personal computer. This facility allows you to perform the following operations:

- input/output operations (using "DOWNLOAD" and "UPLOAD" menus):

- program editing.

8. MESSAGES AND ERROR SIGNALS

This appendix helps you find the meaning of the messages and error signals that appear on the screen of the CRT.

Section 8.1. lists messages and commands displayed in Edit status

Section 8.2. lists error signals and commands displayed in EDP status

Section 8.3. lists messages related to I/O errors

Section 8.4. lists messages displayed during machining

Section 8.5. lists machine logc messages.

8.1. MESSAGES AND COMMANDS DURING EDIT OPERATIONS

During editing operations. the 4th. line of the CRT displays messages that point out errors. Edit errors are shovn in Table 8.1. Table 8.2. shows the allowable edit commands.

These messages and commands are for the present English version of the control. They can be changed for other languages by modifying the system characterization file FCRSYS.

Table 8.1. - Messages in Edit status

MESSAGE	DESCRIPTION
COMMAND ERROR	The entered command moes not have tne allowable format or is not recognised by the system
MEMORY OVERFLOW	The available user memory cannot store the new program or the modified program
RECORD OVERFLOW	The entered block exceeds the maximum length for that type of file
WRONG KEY	Mequest for search for a sequence of characters without inputting $ before the sequence. e.g. only 1.01
KEY OVERFLOW	Request for search for a record number greater than the max. allowed for that type of file. e.g. 999999
WRONG DIRECTION	The search direction selected with (LINE FORWARD) and (LINE BACK) is incompatible with the current cursor position. Example: 18 (LINE BACK) with cursor on position 20.
FORMAT ERROR	The record format is not compatible with the type of file
EOF	The cursor is on the last program block
BOF	The cursor is on the first program block
NEW	A new program is being stored
OLD	An old program is being modified
I/O ERROR	Error when trying to access a memory or a peripheral device

Table 8.2. - Commands in Edit status

COMMAND	MEANING
DIR	List all programs stored in user memory
EDI	Editor call for storing a new program or modifying an old one via keyboard
COP	Transfer a program from memory to peripheral or viceversa. or from memory to memory to make a copy. and display a stored program
DIS	Display a file
REN	Rename a stored program
DEL	Delete a stored program
INI	Initialize a peripheral or the user memory
CRE	Create a fixed length file
FOR	Create the file of offsets. tool life. origins. with fixed length and formatted fields
DIF	Check a stored program
RIM	EDI subinstruction: modify a block
INS	EDI subinstruction: insert a block
CAN	EDI subinstruction: delete a block

8.2. ERRORS AND MESSAGES IN EDP STATUS

Yhis section deals with messages and erros displayed in EDP status. EDP errors are are shown in Table 8.3. EDP messages are listed in Table 8.4.

The texts of these messages are stored in two separate files. FILMS1 and FILMS2. If these files are not stored. the control only displays the message number preceded by the code FILMS1 or FILMS2. For example. FILMS1 20 or FILMS2 03

Table 8.3. - Ressages in EDP Status

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 BUBBLE OR CMOS
FILMS1 06	ILLEGAL RECORD
FILMS1 07	LOAD-GO WITHOUT START ADDRESS
FILMS1 08	INVALID COMMAND
FILMS1 09	FORMAT ERROR
FILMS1 10	SYNTAX ERROR
FILRS1 11	UTILITY NOT AVAILABLE
FILMS1 12	LINE ERROR
FILRS1 13	TIME-OUT ERROR
FILMS1 14	ILLEGAL REQUEST
FILMS1 15	FORMAT ERROR IN SECT. 2 FCRSYS
FILMS1 16	BOOTSTRAP ERROR IN SECT. 2 FCRSYS
FILM51 17	DRIVER NOT PRESENT IN EPROM
FILMS1 18	PERIPHERAL NUMBER DOES NOT MATCH
FILMS1 19	EOF RECORD MISSING
FILMS1 20	FCRSYS FILE NOT FOUND
FILMS1 21	POWER-UP IN EMERGENCY-STOP MODE
FILMS1 22	FORMAT ERROR IN SECT. 1 FCRSYS
FILMS1 23	HDWR AND FCRSYS FILE DO NOT MATCH
FILMS1 24	EPROR DIRECTORY ERROR
FILMS1 25	SECT. 3 OF FCRSYS EMPTY
FILMS1 26	FORMAT ERROR IN SECT. 3 FCRSYS
FILMS1 27	UOOTSTMAP ERROM IN SECT. 3 FCRSYS
FILMS1 28	FILE NOT PRESENT IN EPROM
FILMS1 29	OUT OF MEMORY
FILMS1 30	COMMAND INHIBITED

DESCRIPTION OF EDP 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 enetred is not assigned.

FILMS1 04 PERIPHERAL ERROR
The device specified in the command does not exist. does nmt answer. is not ready or is not initialized.

FILMS1 05 FILE NOT FOUND IN BUBBLE OR CMOS
The file specified in your command does not exist in bubble or CMOS memory.

FILMS1 06 ILLEGAL RECORD
The remuired program cannot be carried out with the present format.

FILMS1 07 LOAD-GO WITHOUT.START ADDRESSED
It concerns the files without start address (non-main files).

FILMS1 08 INVALID COMMAND
The control does not recognized the command you entered.

FILRS1 09 FORMAT ERROR
The control recognizes your command but the format is incorrect

FILMS1 10 SYNTAX ERROR
The control recognizes 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 13 TIME OUT ERROR
Reserved for the system.

FILMS1 14 ILLEGAL REQUESY
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 SECY. 2 FCRSYS
During the power-up initialization sequence. the control detected format or syntax errors in section 2 of FCRSYS.

FILMS1 16 BOOTSTARP ERROR IN SECT. 2 FCRSYS
The declared driver present on bubble or CMOS memory cannot be loaded.

FILMS1 17 DRIVER NOT PRESENT IN EPROM
The driver declared in section 2 of FGRSYS is not present in EPROM.

FILMS1 18 PERIPHERAL NUMBER DOES NOT MATCH
The logic number does not match the driver declared in section 2 of FCRSYS.

FILMS1 19 EOF RECORD MISSING
You have not entered the EOF code.

FILMS1 20 FCRSYS NOT FOUND
The FCRSYS file has not been stored in memory. or the control cannot access FCRSYS because of a memory fault.

FILMS1 21 POWER-UP IN EMERGENCY MODE
You held CSTART3 or CHOLD3 pushbuttons during the power-up initialization. Therefore, the control could not check FCRSYS.

FILMS1 22 FORMAT ERROR SECT. 1 FCRSYS
The control detects a format or syntax error in section 1 of FCRSYS.

FILMS1 23 HRDW AND FCRSYS DO NOT MATCH
The modules installed in your control do not match the configuration seclared in section 1 of FCRSYS.

FILMS1 24 EPROM DIRECTORY ERROR
Reserved for the system.

FILMS1 25 SECT. 3 FCRSYS EMPTY
Section 3 of FGRSYS was not programmed or has no records.

FILMS1 26 FORMAT ERROR IN SECT. 3 FCRSYS
The control detects a format or syntax error in section 3 of FGRSYS.

FILMS1 27 BOOTSTRAP ERROR IN SECT. 3 FCRSYS
The file declared in section 3 of FCRSYS to be on bubble or CMOS memory cannot be loaded from the program.

FILMS1 28 FILE NOT PRESENT IN EPROM
A program declared in section 3 of FCRSYS is not in EPROM. Or. a program specified in a RUN command is not in EPROM.

FILMS1 29 OUT OF MEMORY
Cannot display. memory not available.

FILMS1 30 COMMAND INHIBITED
The utilities that can be disabled are: EDITOR. INI. DELETE.

Table 8.4. - EDP messages

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 DISPLAYED
FILMS2 07	WAIT: JCL BUSY
FILMS2 08	TRANSFER ERROR
FILMS2 09	DO YOU CONTINUE? (Y/N)
FILMS2 10	UNIT FULL-INSERT NEW DISX- PRESS Y

DESCRIPTION OF EDP MESSAGES

FILMS2 01 NOT USEO

FILMS2 02 D0 YOU CONFIRM? (Y/N)
The message appears when the whole memory must be cleared. Enter Y press **[SEND]** to delete all files. Enter N press **[SEND]** to return to the Job status.

FILMS2 03 COMMANO 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 **[SEND]** to update all records starting with 1. If you type N press **[SEND],** 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 **[SEND]** to cancel the old file and copy the new one. If you type N press **[SEND],** the new file is not copied.

FILMS2 06 FILE DISPLAYED
You copied a file to the CRT display.

FILMS2 07 WARNING JCL BUSY
Yhis message appears when you are operating with the MINI DNC and enter a command using the keyboard or viceversa.

FILMS2 08 TRANSFER ERROR
The transfer has been achieved but either parity errors or unknown characters have been detected.

FILMS2 09 D0 YOU CONTINUE? (Y/N)
This message is generated by utility DIF. when a difference between two files is detected.
If you press Y press **[SEND]** the comparison continues, if you press N press **[SEND]** it stops.

FILMS2 10 UNIT FULL - INSERT NEW DISX - PRESS Y
This message appears when the contents of the device (CMOS, bubble memory. hard disk or wild card) overflow the capacity of the floppy disk. The system will interrupt the downloading operatimn and cancel the file that has been downloaded only partially.

To resume downloading. use this procedure:

- Insert a new diskette;
- Press Y press **SEND**to continue. Downloading will be resumed from the interrupted file.
  or
  Press N press **[SEND]** to abort downloading.

8.3. I/O ERROR MESSAGESS

The list of available I/O errors is shown in Table 8.5.

The text of these messages is stored in a file associatedd to the logic name FILMlS3. 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 debugginq control software, we describe only the messages that appear during operation.

Table 8.5. - 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 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	DEVICE NOT FORMATTED
FILMS3 35	HARDWARE ERROR
FILMS3 36	INVISIBLE FILE
FILMS3 38	LIHE ERROR
FILMS3 39	FORMAT ERROR ON TRACK O HD/FD
FILMS3 40	RW ERROR ON TRACK HD/FD
FILMS3 41	FD NOT PRESENT

DESCRIPTION OF I/O ERROR MESSAGES

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.

FILMS3 12 FILE OPEN
Indicates that you must close the editor before trying to select or execute a program.

FILMS3 15 NO FREE SECTORS
The specified destination device is full.

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. Select a program before pressing **[CYCLE START].**

FILMS3 24 FILE NOT FOUND
The file associated to a logic name in section 7 of FCRSYS does not exist.

FILMS3 26 WRITE PROTECTED FILE
You cannot write on this file vith the ATT command.

FILMS3 30 DEVICE NOT READY
The specified device is not ready for use.

FILMS3 34 DEVICE NOT FORMATTED
The required support (FD or HD) has not been initialized yet.

FILMS3 35 HARDWARE ERROR
A bubble memory module has a hardware fault.

FILMS3 36 INVISIBLE FILE
Yhe specified file cannot be displayed on the CRT.

FILMS3 38 LINE ERROR
The required peripheral is not properly connected.

FILMS3 41 NO FD PRESENT
The FD has not been inserted in its drive yet.

8.4. MESSAGES DISPLAYED DURING MACHINING

The 4th.line of the GRT shows messages that indicate particular conditions or faults of the system (machine tool and control) during machining.

Messages related to program errors ar: dismlayed in reverse.

The maximum message length is 32 characters.

All errors displayed in reverse are block:ind errors: you must execute a reset to exit from the error concition.

Machining messages (Table 8.6.) are storec in a separate file called FILMS4. If this file is not stored. the control displays only the message number, precedec by code FILMS4. For example. FILMS4 72.

Table 8.6. - 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	UNlDEFINED 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	DRTA 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	FUNGTI0Nl UNDEFINED
FILRS4 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
FILRS4 29	AXIS ORIGIN/OFFSET NOT DEFINED
FILMS4 30	PROBING PARAMETERS UNDEFINED
FILMS4 31	RPT NEST >3 OR ERP WIITHOUT RPT
FILRS4 32	SUBROUTINE NEST >2
FILMS4 33	EPP NEST >1
FILMS4 34	UMNDEFINED 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 8.6. 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	RLTERNATIVE TOOL NOT IN FILE
FILMS4 58	END OF PROGRAM
FILRS4 59	BLOCK NOT ALLOWED
FILMS4 60	SPINDLE AXIS CHANGE NOT ALLOWED
FILMS4 61	NOT USED
FILRS4 62	NOT USED
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 NEASURING 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 8.6. 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
FILMS6 94	INCOMPLETE PROGRAM
FILMS4 95	PROBING OPERANDS DEACTIVATED
FILMS6 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
FILMS6 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
FILMS6.110	PROBE INPUT SIGNAL NOT COHERENT

DESCRIPTION OF PROCESS MESSAGES

FILMS4 01 SYSTEN ERROR
This message appears:
. when an axis has been reset without having declared the PGCFIL origins file;
. when you move a spindle axis that has been declared in MAS.
In all other cases, call the assistance service and fully describe the coditions in which the error occurred.

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 configuration file.

FILMS4 03 OUT OF OPERATING LIMITS
This message appears when you programmed a movement greater than Allowed in the configuration file.

FILMS4 04 HARDWARE OVERTRAVEL
One of the end of stroke microswitches has been pressed.

FILMS4 05 AUTO RETURN TO PROFILE ILLEGAL
Using three-letter come RAP=i, 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 manual jog selector, 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 following 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 messaqe appears if, in a part program block or in a block entered from tlhe 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.

FILMS4 16 STRIN6 TOO LONG
Nlumber 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.

FILMS4 19 PARAMETER OVERFLOW
This message appears in case of division by zero.

FILMS4 20 UHDEFINED FUNCTION
Call for synchronization not compatible with system status.

FILMS4 21 ILLEGAL NUMBEM OF OPERATORS
This message appears if more than 8 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 (pointto-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 UNDEFINEO
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 withmut closing any open RPT and/or EPP cycle.

FILMS6 36 WMONG 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 ddefined 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

FILMS6 46 ROTATION RANGE NOT DEFINED
Spindle rotation range not inserted.

FILMS4 47 TO0 MANAEXSPEDITED 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 50 INDEXING AXIS NOT HOMED
There is an attempt to move an indexing axis without homing it first.

FILMS4 51 UNKNOWN M mr 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/SEMIAUTO 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 NOT USED

FILMS4 62 NOT USED

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.

FILRM4 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 OVERFLOW
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 progammed 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 tne 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 transduser.

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 open an existent file

FILMS4 100 WRONG CHANNEL
This message appears if:
a) the specified channel is busy,
b) the specified channel is
c) all the channels specified in OPN are busy
d) you programmed PUT, GET or DCL without first programming SOP
e) the serial line is busy

FILMS4 101 ILLEGAL OPERATION ON FILE
This message appears if:
a) the specified record does not exist
b) the specified record structure has not been defined

FILMS4 102 USER SCREEN ERROR
This message appears if:
a) the field specified in OUT has not been defined
b) the format of the variable is not compatible with the format of the specified field
c) the screen has not been declared with SCR

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 thar 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 a block that exceeds the limit declared in PGCFIL.

FILMS4 110 PROBE INPUT SIGNAL ILLEGAL
The status of the probe input is not compatible With the characterized status.

FILMS4 111 HANDWHEEL ENABLED
Handwheel enabled for the selected axis.

8.5. MACHINE LOGIC MESSAGES

SIPROM allows you to customize the interface between the control and the machine tool.

With SIPROM, you can display as many as 255 machine logic messages having 32 characters each. Machine logic messages are displayed on line 4 of screen 1. 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 0ц63. 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.