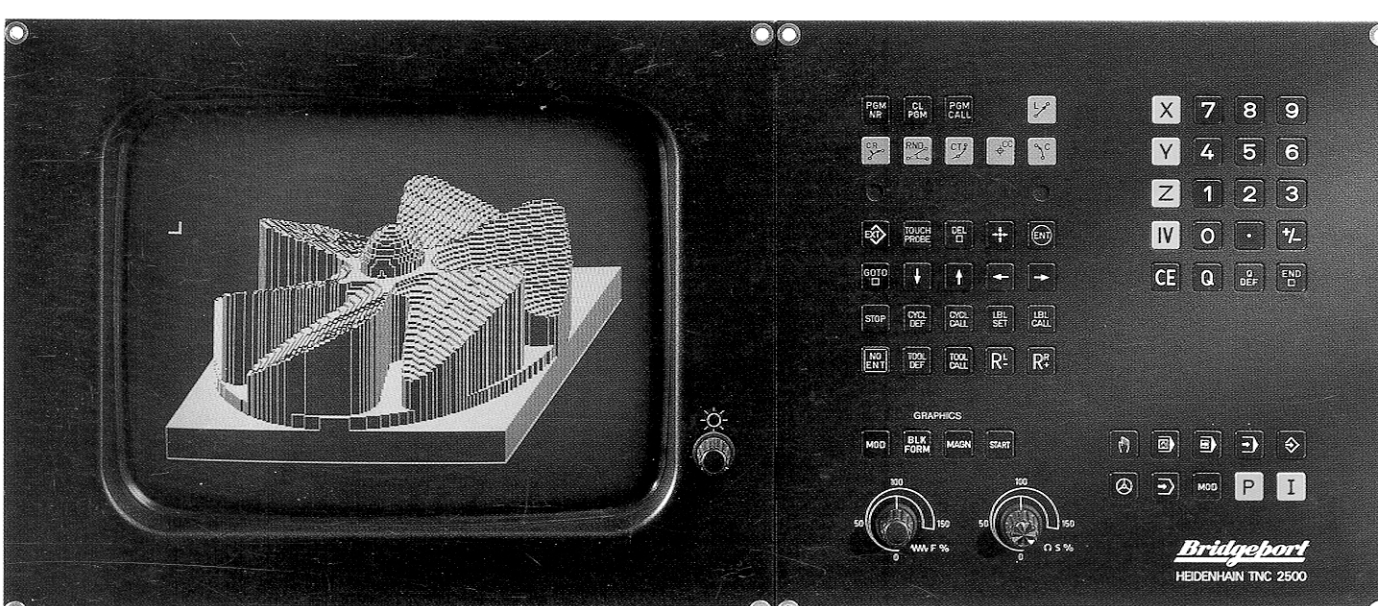
HEIDENHAIN

Operating Manual

Conversational Programming

TNC 2500 B
Contouring Control



October '91

Screen displays

```
PROGRAMMING AND EDITING

1    BLK FORM 0.1   Z   X+0
         Y+0            Z-20
--------------------------------
2    BLK FORM 0.2       X+100
         Y+100          Z+0
--------------------------------
3    TOOL DEF 1         L+1
                        R+1
4    TOOL CALL 1            Z
             S 1000
- - - - - - - - - - - - - - - - -

ACTL. XSN+   98,354  YN -   37,580
      Z +    32,000  C  +   82,600


                   ROT +    37,000
                   SCL  0,750000
T1    Z              F        M5/9
```

Operating mode
Error messages

Preceding block

Current block

Next block

Block after next

Status display

Status display:

ACTL.: Type of position display, switchable with MOD
(further displays: NOML, DIST., LAG – see index "General Information")

X …
Y …
Z …
etc. } Position coordinates

✱: "control is started" display
D: Datum shift, shown as an index on the shifted axis.
M: Mirror image, shown as an index on the mirrored axis.
ROT: Basic rotation of the coordinate system
SCL: Scaling
CC: Circle center or pole

T …: Called tool
Z: Spindle axis
S: Spindle speed

F: Feed rate
M: Spindle status (M03, M04, M05, M13, M14)

Guideline for procedure
from preliminary operations to workpiece machining

Sequence	Action	Operating mode	Cross reference	Page
1	Select tools	–	Workpiece drawing	–
2	Set datum for workpiece machining	–	Workpiece coordinates	A15
3	Determine speeds and feed rates	–	Spindle speed, feed rate diagrams	A20
4	Switch on machine	–	Machine operating manual	–
5	Traverse reference points (homing the machine)	–	Switch on	M1
6	Clamp workpiece	–	Clamping instructions	–
7a	With 3D Touch Probe: datum setting and compensation of workpiece misalignment	Manual	Workpiece setup with the 3D Touch Probe	M3
or				
7b	Align workpiece, insert zero tool, mark workpiece and set datum	Manual	Manual operation Machine handbook: Tool change	M13
8	Enter program – by keying in or from external storage device	Programming and editing	Back fold-out page, program example; Programming and editing	P1
9	Test program (without axis movements)	Test run	Programming, Test run	P135
10	Graphic program simulation (without axis movements)	Program run	Programming, Graphic simulation	P136
11	Test run without tool in single block mode	Program run, Single block	Program run	M20
12	Optimize program if necessary	Programming and editing	Programming and editing	P3
13	Insert tool and machine workpiece automatic program run	Program run, Full sequence	Program run	M20

Operating Panel TNC 2500 B

Machine Operating Modes

	Manual operation
	Electronic handwheel
	Positioning with manual data input
	Program run, Single block
	Program run, Full sequence

Programming Modes

	Programming and editing
	Test run with graphic simulation

Program Management

PGM NR	Naming/selecting a program
CL PGM	Clear program
PGM CALL	Programmable program call
EXT	External program input and output
MOD	Supplementary operating modes

Graphics

MOD	Graphic operating modes
BLK FORM	Define blank form, reset blank form
MAGN	Magnify detail
START	Start graphic simulation

Override

S%	Feed rate override
F%	Spindle speed override

Programming

Entering the Workpiece Contour

L	Straight line
CC C	Circle with known center
CR	Circle with known radius
CT	Circle with tangential transition
RND	Round corners/ Tangential contour approach and departure
TOOL DEF, TOOL CALL	Define/Call a tool
RL, RR	Specify mode tool radius compensation
CYCL DEF, CYCL CALL	Define/Call a cycle
LBL SET, LBL CALL	Label/Call a subprogram and program section repeats
STOP	Programmed stop/Terminate program
TOUCH PROBE	Touch probe functions

Entering and Editing Values

X . . . IV	Axis keys
0 . . . 9	Number keys
. +/	Decimal point, sign change
P	Key for polar coordinates
I	Key for incremental dimensions
Q, Q DEF	Enter parameter instead of a number, Define parameter
	Transfer actual position to memory
GOTO	Cursor keys, Jump to a certain block or cycle
NO ENT, ENT, END	No entry, Enter data, Terminate block entry
CE	Clear entry
DEL	Delete block

Contents

General Information (A)

Introduction

Description

The TNC 2500B from HEIDENHAIN is a shop-floor programmable contouring control with up to 4 axes for milling and boring machines as well as for machining centers. It is conceived for the "man at the machine", featuring conversational programming and graphic simulation of workpiece machining. Fixed cycles, coordinate transformations and parameter programming are available, as well as functions for 3D touch probes. Its "parallel operation" feature permits a new program to be created (or a program located in the control memory to be edited) while another program is being executed.

Programs can be output to peripheral devices and read into the control via the RS-232-C data interface, allowing programs to be created and stored externally.

Conversational or ISO programming

In addition to programs written in conversational format, ISO programs can also be entered, either via the snap-on keyboard or via the data interface. Both interactive format and ISO format programs can reside in memory at the same time.

Compatibility

This control can execute programs from other HEIDENHAIN controls, provided they contain only the functions described in this manual.

Structure of manual

This manual addresses the skilled machine operator and requires appropriate knowledge of non-NC-controlled boring and milling.

TNC beginners are advised to work through this manual and the examples systematically. If you have already worked with a HEIDENHAIN TNC, you can skip familiar topics.

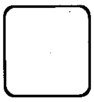
This manual deals with **programming in HEIDENHAIN format.** ISO programming is described in detail in a separate operating manual for the TNC 2500B.

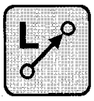
The sequence of chapters in this operating manual is according to control operating modes and key functions, as well as according to the logical working order:

- **Machine operating modes:**
  Switch-on – manual setup – set display value – machine workpiece.
- **Programming modes:**
  Enter program – test program.

Symbols for keys

The following symbols are used in this manual:

Empty square:  ... keys for numerical input on the TNC operating panel

Square with symbol, e.g.  ... other keys on the TNC operating panel

Circle with symbol, e.g.  ... keys on the machine operating panel

The pages of this manual are distinctly marked with the relevant key symbols.

Typeface for screen displays

Program blocks and TNC screen dialogs are printed in this SPECIAL TYPE.

Introduction

Program Examples

The example programs in this manual are based on a uniform blank size and can be displayed on the screen by adding the following blank definition (see index "Programming Modes", Program Selection):

BLK FORM 0.1 Z X+0 Y+0 Z–40
BLK FORM 0.2 X+100 Y+100 Z+0

The examples can be executed on machine tools with tool axis Z and machining plane XY. If your machine uses a different axis as the tool axis, this axis must be programmed instead of Z and likewise the corresponding axes for the machining plane.

Beware of collisions when executing the example programs!

Buffer batteries in the control

Buffer batteries protect the stored programs and machine parameters against loss due to power interruption.

When the message
EXCHANGE BUFFER BATTERY
appears, you must change the batteries.

The batteries should be replaced once a year.

Battery type:
3 AA-size batteries, leak-proof
IEC designation "LR6"

Changing the battery

Battery replacement is described in the manual of the machine manufacturer.

Error messages

The TNC checks input data and status of the control and machine.

	Cause and reaction of the control:	Remedy:
Input range exceeded	The permitted range of values is exceeded: e.g. feed rate too high. The value is not accepted and an error message appears.	Clear the value with the "CE" key, enter and confirm the correct value.
Incompatible/ contradictory inputs	E.g. L X+50 X+100 During "TEST" or during program execution, the TNC stops with an error message before executing the corresponding block and displays the block number in which an error was found.	Change to the "Programming" operating mode. The error can normally be found either in the block with the displayed block number or in a previously executed block. Then: correct the error. Operating mode "Full sequence" and restart.
Malfunction of the machine or control	Malfunctions that affect operating safety cause blinking error messages. Note down the error message!	Switch off the machine or the control. Remove the fault if possible. Attempt to restart. If the program then runs correctly, the problem was only a spurious malfunction. If the same error message comes up again, contact the customer service of the machine manufacturer.

TNC 2500 B
Brief description

Control type	Contouring control for 4 axes
Traversing possibilities	Straight lines in 3 axes Circles in 2 axes Helix
Parallel operation	Programming and program execution simultaneously
Graphics	**Test graphics** in the "Program run" operating modes
Program input	In HEIDENHAIN format or according to ISO
Input resolution	Max. 0.001 mm or 0.0001 inch or 0.001°
Program memory	For 32 programs, battery buffered: 4000 program blocks
Tools	Up to 254 tool definitions in a program Up to 99 tools in the central tool file

Programmable functions

Contour	Straight line, chamfer Circle (input: center and end point of the arc or radius and end point of the arc), circle connected tangentially to the contour (input: arc end point) Corner rounding (input: radius) Tangential approach and departure from a contour
Program jumps	Subprograms, program section repeats, call of other programs
Fixed cycles	Drilling cycles for pecking, tapping Milling cycles for rectangular pocket, circular pocket, slot "Subcontour List" cycles for milling pockets and islands with irregular contours
Coordinate transformations	Move and rotate the coordinate system, mirror image, scaling
Probing functions	For 3-D touch trigger probe
Digitizing	With TS 120 and software expansion option Optional evaluation software for PC
Parameter programming	Mathematical functions ($=$ / $+$ / $-$ / x / $\div$ / sin / cos / angle α from axis sections / $\sqrt{a}$ / $\sqrt{a^2+b^2}$); parameter comparison ($=$ / $\neq$ / $>$ / $<$)
Traversing range	Max. ± 30000 mm or 1181 inches
Cutting data	Traversing speed: max. 30 m/min or 1181 inches/min Spindle speed: max. 99999 rpm

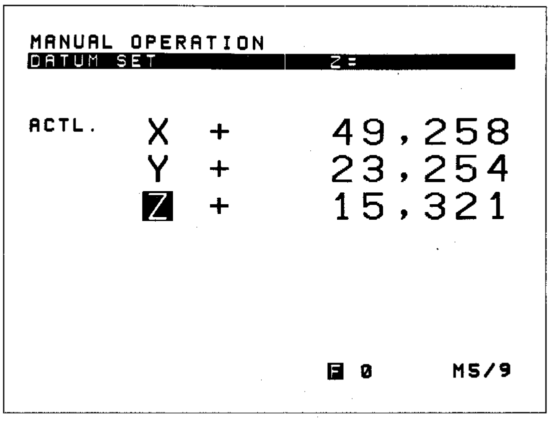
Hardware

Component units	Logic unit, control panel and monochrome screen
Block processing time	1500 blocks/min (40 ms)
Control loop cycle time	6 ms
Data interface	RS-232-C/V.24 Data transfer speed: max. 19200 baud
Ambient temperature	Operation: 0° C to 45° C (32° F to 113° F) Storage: –30° C to 70° C (–22° F to 158° F)

Machine operating modes

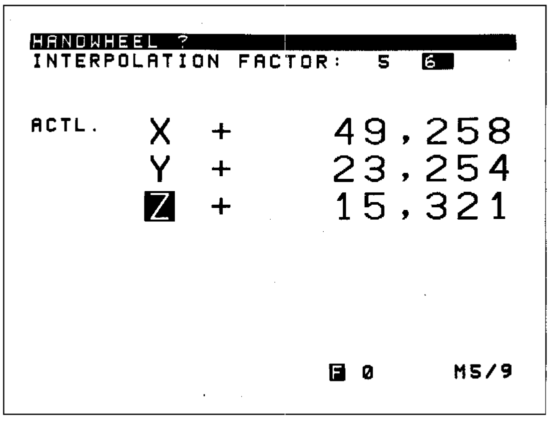
Manual operation

The axes can be moved via the external axis direction buttons. Workpiece datum can be set as desired.
The TNC functions as a conventional numerical position readout (DRO).



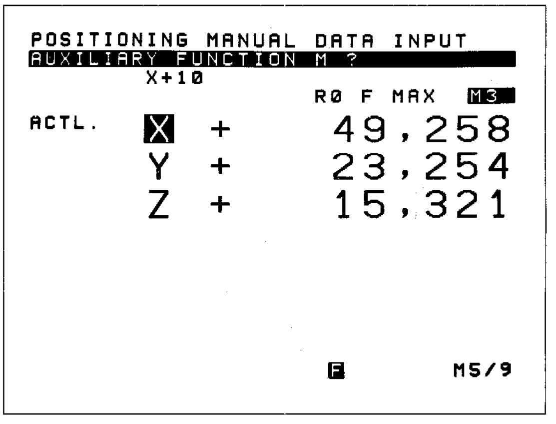
Electronic Handwheel

The axes can be moved either via an electronic handwheel or via the external axis direction buttons. It is also possible to position by defined jog increments.



Positioning with manual data input (MDI)

The axes are positioned according to the data keyed in. These data are not stored.



Program run

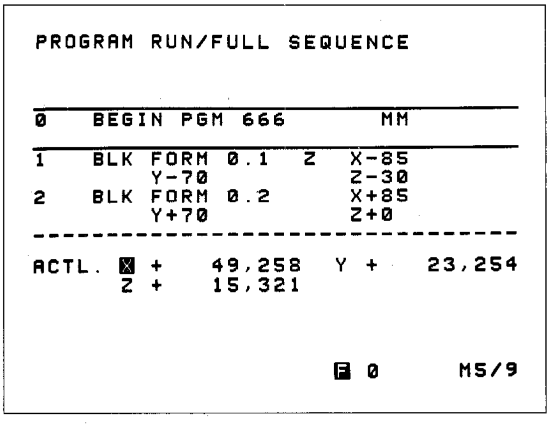
A part program in the memory of the control is executed by the machine.

Full sequence

After starting via the machine START button, the program is automatically executed until the end or a STOP is reached.

Single block

Each block is started separately with the machine START button.



Programming and editing operating modes

Programming and editing

Part programs can be entered, looked over and altered in the "Programming and editing" operating mode.

In addition, programs can be read in and output via the RS-232-C data interface.

```
PROGRAMMING AND EDITING
TOOL RADIUS COMP. RL/RR/NO COMP. ?
17  TOOL CALL 2                Z
             S 300
18  L  Z+2
                       R  F15998 M03
19  CYCL DEF 6.0 ROUGH-OUT

20  CYCL DEF 6.1  SET UP-2
                  DEPTH -20
---------------------------------
ACTL. X +    49,258   Y +    23,254
      Z +    15,321


                       F 0      M5/9
```

Test run

In the "Test run" operating mode, machining programs are analyzed for logical programming errors, e.g. exceeding the traversing range of the machine, redundant programming of axes, certain geometrical incompatibilities etc.

```
TEST RUN
     TO BLOCK NUMBER =

0   BEGIN PGM 666       MM

1   BLK FORM 0.1  Z   X-85
         Y-70         Z-30
2   BLK FORM 0.2      X+85
         Y+70         Z+0
---------------------------------
ACTL. X +    49,258   Y +    23,254
      Z +    15,321   C +    84,000


                       F 0      M5/9
```

Test graphics

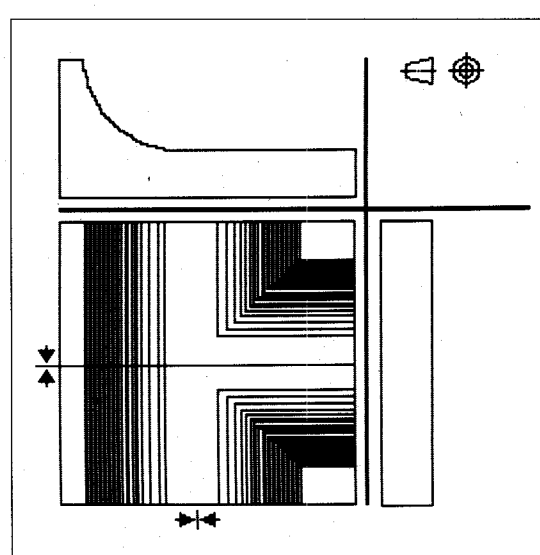
GRAPHICS

BLK FORM MAGN START

MOD

In the "Program run" operating modes "full sequence" and "single block", you can graphically simulate machining programs via the "GRAPHICS" keys.

Display modes:
- plan view with depth indication
- view in three planes
- 3-D view



External data transfer

EXT

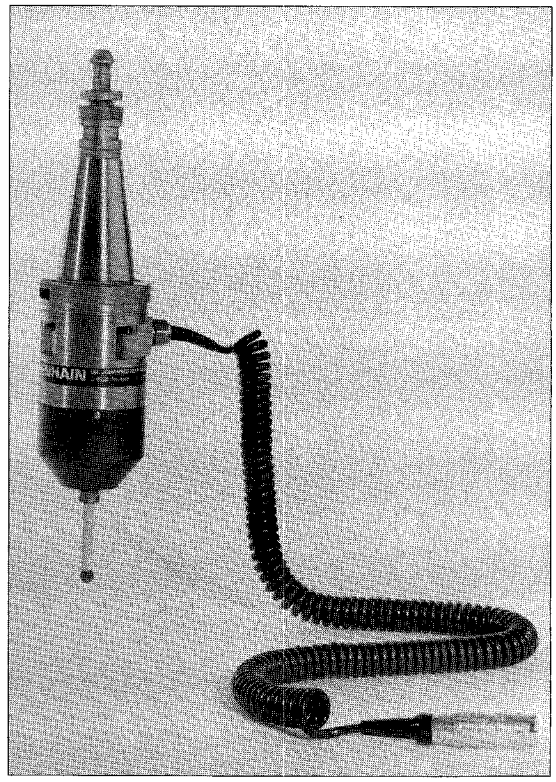
In the "Programming and editing" mode, programs can be read-in from an external storage medium, such as the FE 401 Floppy Disk Unit, and read-out to an external unit (data storage, printer). Data transfer takes place via the RS-232-C data interface.

In the "Program run single block" and "Program run full sequence" modes of operation it is possible to read-in programs whose size exceeds the control's memory block by block for simultaneous execution.

TOUCH PROBE

Accessories
3D Touch Probe Systems

The TNC software incorporates measuring cycles for the application of a HEIDENHAIN 3-D Touch Probe in the "Manual", "Handwheel" and "Program run" operating modes.



TS 120

Manual use

The following measurements can be performed in the "Manual" and "Handwheel" operating modes:

- position
- line
- angle
- corner point
- circle radius and circle center.

The probing functions allow compensation of workpiece misalignment and automatic setting of the position displays.
Thus, they help you setup workpieces more easily, quickly and accurately.

The probing functions can also be used for measurements on the workpiece.

Program run

You can program position measurements in the "Programming and editing" operating mode. This feature can be used with Q parameter programming to execute measurements before, during and after machining a piece (see index "Programming and Editing", Programmable probing function and Parameter programming).

HEIDENHAIN offers touch probes in various versions. There are different clamping shafts to affix the probe head in the spindle like a tool. The stylus is replaceable.
Standard versions are:

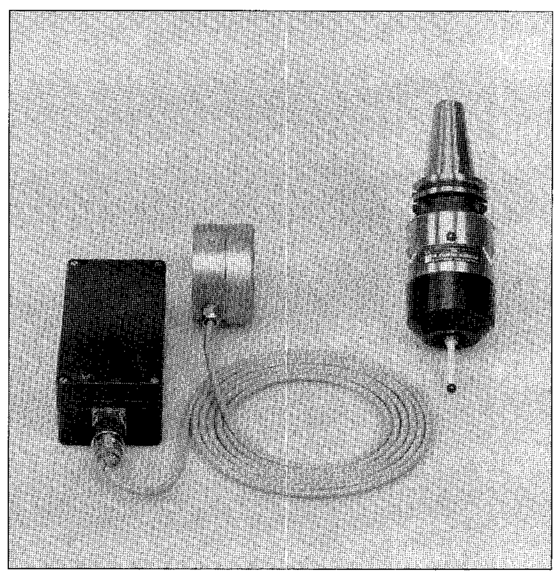
TS 120

Touch Probe System 120
with cable connection and interface electronics, incorporated into probe.

TS 511

Touch Probe System 511
with infrared transmission, separate interface electronics and transmitter/receiver unit.

This probe head has a transmitter and receiver window (for the triggering signal) on one side and another transmitter window offset by 180°. The side with the transmitter and receiver window must be pointed towards the transmitter/receiver unit during measurement.



TS 511

Certain provisions are required by the machine tool manufacturer for the connection of a touch probe system.

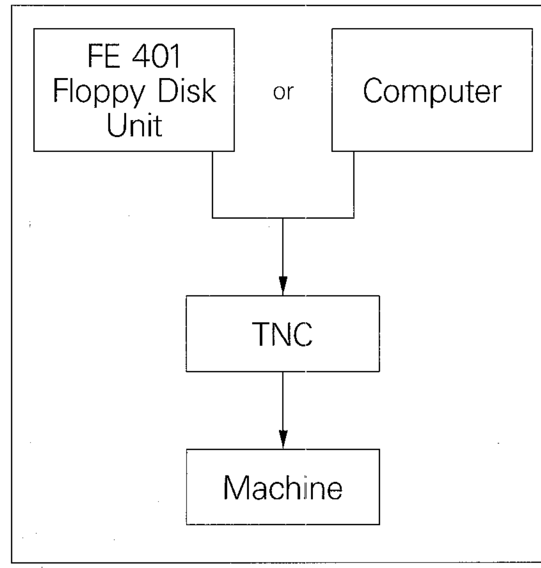
Accessories
FE 401 Floppy Disk Unit
HR 130/HR 330 Electronic Handwheels

FE 401 Floppy Disk Unit

Part programs which do not have to reside permanently in the control memory can be stored with the FE 401 Floppy Disk Unit.

The storage medium is a normal 3 1/2 inch diskette, capable of storing up to 256 programs and a total of approximately 25 000 program blocks.

Programs can be transferred from the TNC to diskette or vice-versa.



Programs written at off-line programming stations can also be stored on diskette with the FE 401 and read into the control as needed.

In the case of extremely long programs which exceed the storage capacity of the TNC, the FE 401 can be used to transfer a program blockwise into the control while simultaneously executing it.

A second diskette drive is provided for backing up stored programs and for copying purposes.

Technical Data

	FE 401 Floppy Disk Unit with two drives
Data medium	3 1/2 inch diskette, double-sided, 135 TPI
Storage capacity	approx. 790 KB (25 000 blocks); max. 256 programs
Data interface	Two RS-232-C data interfaces
Transfer rate	"TNC" interface: 2400/9600/19 200/38 400 baud "PRT" interface: 110/150/300/600/1200/2400/4800/9600 baud

Handwheel

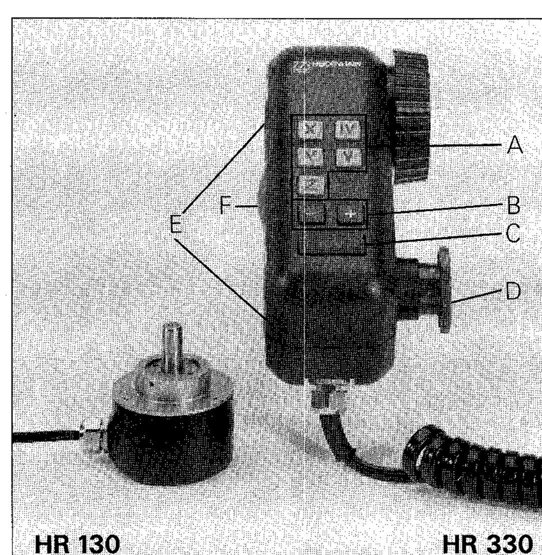
The control can be equipped with an electronic handwheel for better machine setup.
Two versions of the electronic handwheel are available:

HR 130

The HR 130 electronic handwheel is designed to be incorporated into the machine control unit. The axis of control is selected at the machine control panel.

HR 330

The portable HR 330 electronic handwheel includes keys for axis selection, axis direction, rapid traverse and emergency stop.



MOD Functions

MOD

In addition to the main operating modes, the TNC has supplementary operating modes or so-called MOD functions. These permit additional displays and settings.

Initiate the dialog MOD

Selecting

VACANT MEMORY 160044	↓ ↑ MOD	Select MOD functions either via arrow keys or via the MOD key (only paging forward possible).

Terminating

LIMIT X+ = + 350.000	END	Terminate supplementary operating mode.

Transfer numerical inputs with the "ENT" key before terminating the MOD functions.

Vacant memory

The number of free characters in the program memory is displayed with the MOD function "VACANT MEMORY".

Programming and editing

You can use this MOD function to switch the control between conversational format (**HEIDENHAIN**) and ISO format (**ISO**). Switchover is performed with the "ENT" key.

Baud rate

The transfer rate for the data interface is specified with "BAUD RATE".

RS-232-C interface

The data interfaces can be switched via "RS-232-C interface" to the following operating modes with the "ENT" key:

- ME operation
- FE operation
- EXT operation: operation with other external devices.

NC software number

The software number of the TNC control is displayed with this MOD function.

PLC software number

The software number of the integrated PLC is displayed with this MOD function.

User parameters

Up to 16 machine parameters can be accessed by the machine operator with this MOD function. These user parameters are defined by the machine manufacturer – he may be contacted for more information.

Code number

A code number can be entered with this MOD function:

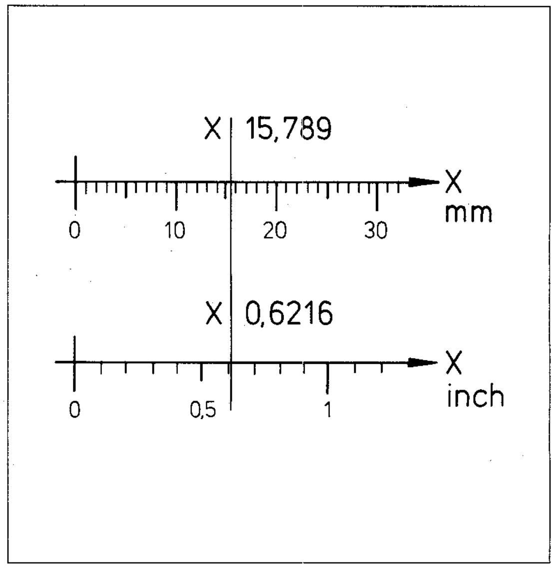
- **86357:** cancel "erase and edit protection"
- **123:** select the user parameters.
  These user parameters are accessible on all controls (see User parameters).

MOD Functions
Position displays

Change mm/inch

The MOD function "Change mm/inch" determines whether the control displays positions in the metric system (mm) or in the inch system. You switch between the mm and inch systems via the "ENT" key. After pressing this key the control switches to the other system.

You can recognize whether the control is displaying in mm or inches by the number of digits behind the decimal point:
X15.789 mm display
X 0.6216 inch display.



Position displays

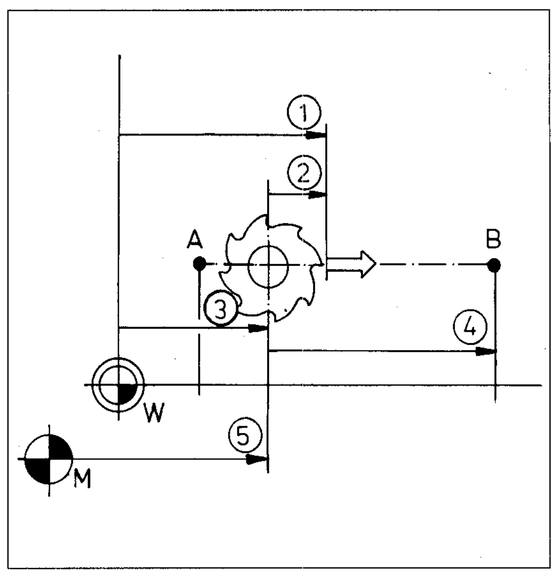
The following position displays can be selected:

① nominal position of the control	NOML
② difference nominal/actual position (lag distance)	LAG
③ actual position	ACTL.
④ remaining distance to programmed position	DIST.
⑤ position based on the scale datum	REF

A = last programmed position (starting position)
B = new (programmed) target position, which is presently targeted

W = Workpiece datum for the part program
M = scale datum (machine-based)

Switchover is with the "ENT" key.



Position display large/small

The character height of the position display can be changed in the operating modes "Program run/single block" or "Program run/full sequence". The position display shows 11 program blocks with small characters, two with large characters.

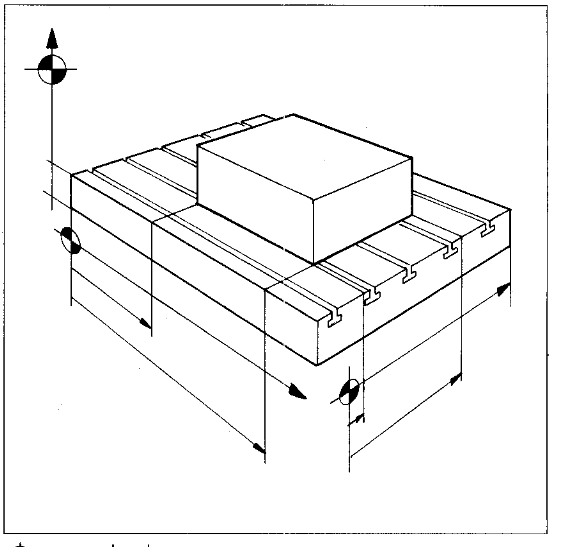
Switchover is with the "ENT" key.

MOD Functions
Traverse range limits

MOD

Limits

The maximum displacements are preset by fixed software limits.
The MOD function "Limits" enables you to specify additional software limits for a "safety range" within the limits set by the fixed software limits. Thus you can, for example, protect against collision when clamping a dividing attachment. The displacements are limited on each axis successively in both directions based on the **scale datum** (reference marks). The position display must be switched to REF before specifying the limit positions of the position display.
To work without safety limits, enter the maximum values +30 000.000 or –30 000.000 for the corresponding axes.



= scale datum

Effectiveness

The entered limits do not account for tool compensations. Like the software limit switches, they are only effective after you traverse the reference points. They are reactivated with the last entered values after a power interruption.

Determine values

To determine the input values, switch the position display to REF.

X Y Z Traverse to the end positions of the axis/axes which is/are to be limited.
Note the appropriate REF displays (with signs).

Enter values

Select

MOD ↓ Continue pressing until LIMIT appears.

Enter the limit(s)

ENT Enter value, or

↓ select the next limit

.
.
.

END terminate the input.

User Parameters
General Information

MOD

Machine parameters

The TNC contouring controls are individualized and adapted to the machine via machine parameters (MP). These parameters consist of important data which determine the behavior and performance of the machine.

Parameters accessible for the user

Certain machine parameters which determine functions dealing only with operation, programming and displays are accessible for the user.

Examples

- Scaling factor only effective on X, Y or on X, Y, Z.
- Adapting the data interface to different external devices.
- Display possibilities of the screen.

Accessibility

The user can access these machine parameters in two ways:

- Access by entering the **code number 123.**
  This access is possible on every control (see code number 123).
- Access to additional parameters via the MOD function **User parameters.**
  You can only access via the MOD function if the manufacturer has made the machine parameters accessible for this purpose.

The machine manufacturer can inform you about the sequence, meaning, texts etc. of any user parameters.

Only these machine parameters may be changed by the user. In no case should the user change any non-accessible machine parameters.

Selection

MOD ↑ ENT Select the user parameter.

↓ Continue pressing until the desired USER PARAMETER or dialog appears.

Enter numbers.

END END Terminate or select further user parameters with ↓ and then terminate.

User Parameters

After entering the code number **123** via MOD, the following machine parameters and the parameters for the data interface (see index "Programming Modes", "External data transfer") can be selected and changed.

Measuring with the 3D touch probe

Function	Parameter no.	Input	Input values
Probe system selection	6010	0 → Cable transmission 1 → Infrared transmission	
Probe system: feed rate for probing	6120	80 to 3000 [mm/min]	
Probe system: measuring distance	6130	0 to 30000.000 [mm]	
Probe system: set-up clearance over measuring point for automatic measurement	6140	0 to 30000.000 [mm]	
Probe system: rapid traverse for probing	6150	80 to 29998 [mm/min]	

Display and programming

Function	Parameter no.	Input	Input values
Programming station	7210	0 → Control 1 → Programming station: PLC active 2 → Programming station: PLC inactive	
Block number increment	7220	0 to 255	
Switching of dialog language German/English	7230	0 → First dialog language 1 → Second dialog language (English)	
Inhibit PGM input for PGM no. = user cycle no.	7240	0 → Inhibited 1 → Uninhibited	
Central tool file	7260	0 → No central tool file 1 to 99 = Central tool file Input value = Number of tools	
Display of the current feed rate before start in the **manual operating modes** (same feed rate in all axes, i.e smallest programmable feed rate)	7270	0 → No display 1 → Display	
Decimal character	7280	0 → Decimal comma 1 → Decimal point	
Display increment	7290	0 → 1 μm 1 → 5 μm	
Clearing the status display and the Q parameters with M02, M30 and end of program	7300	0 → Status display is not cleared 1 → Status display is cleared	
Graphics (display mode)	7310 Bit		
Switch over projection type "display in 3 planes"	0	+ 0 → German standard + 1 → American standard	
Rotate the coordinate system in the machining plane by 90°	1	+ 0 → No rotation + 2 → Coordinate system rotated by +90°	

Machining and program run

Function	Parameter no.	Input	Input values
"Scaling" cycle is effective on 2 axes or 3 axes	7410	0 → 3 axes 1 → in the machining plane	
SL cycles for milling pockets with irregular contour	7420		
	Bit		
"Rough out" cycle: direction for pilot milling of contour	0	+ 0 → Pilot milling of contour for pockets counterclockwise, for islands clockwise + 1 → Pilot milling of contour for pockets clockwise, for islands counterclockwise	
"Rough out" cycle: sequence for rough out and pilot milling	1	+ 0 → First mill a channel around the contour, then rough out the pocket + 2 → First rough out the pocket, then mill a channel around the contour	
Joining compensated or uncompensated contours	2	+ 0 → Joining compensated contours + 4 → Joining uncompensated contours	
"Rough out" and "pilot milling" to pocket depth or for every infeed	3	+ 0 → "Rough out" and "pilot milling" are performed continuously over all infeeds + 8 → "Pilot milling" and then "rough out" are performed for every infeed (depending on bit 1) prior to the next infeed	
Overlap factor for pocket milling	7430	0.1 to 1.414	
Output of M functions	7440		
	Bit		
Programmed stop at M06	0	+ 0 → Programmed stop at M06 + 1 → No programmed stop at M06	
Output of M89, modal cycle call	1	+ 0 → No cycle call, normal output of M89 at start of block + 2 → Modal cycle call at end of block	
Constant path speed at corners	7460	0 to 179.999	
Display mode for rotary axis	7470	0 → 0 to 359.999 1 → ± 30000.000	

Hardware

Function	Parameter no.	Input	Input values
Feed rate and spindle override	7620		
	Bit		
Feed rate override, if rapid traverse key is pressed in operating mode "Program run"	0	+ 0 → Override inactive + 1 → Override active	
Feed rate override in 2% increments or 1% increments	1	+ 0 → 2% increments + 2 → 1% increments	
Feed rate override, if rapid traverse key and external direction buttons are pressed	2	+ 0 → Override inactive + 4 → Override active	
Handwheel	7640	0 = Machine with electronic handwheel 1 = Machine without electronic handwheel	

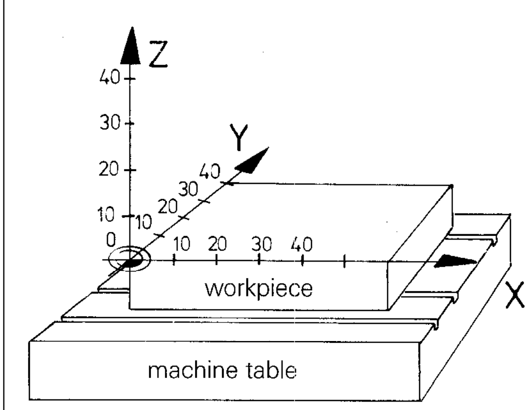
Coordinates
Coordinate system

In a part program, the **nominal positions** of the tool (or of the tool cutting edge) are defined in relation to the workpiece; encoders on the machine axes continuously deliver the signals needed by the control for determining the current **actual position.**

A reference system is always required for determining position. In the present case, such a system must be **workpiece-based.**

Cartesian coordinates

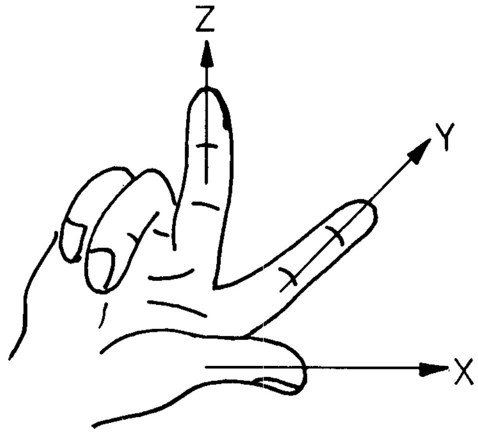
The reference system normally used is the **rectangular** or **Cartesian* coordinate system** (coordinates are those values which define a unique point in a reference system). The system consists of three coordinate axes, perpendicular to each other and lying parallel to the machine axes, which intersect each other at the so-called origin or (absolute) zero point. The coordinate axes represent mathematically ideal straight lines with divisions; the axes are termed X, Y and Z.



Right-hand rule

You can easily remember the traversing directions with the **right-hand rule:** the positive direction of the X axis is assigned to the thumb, that of the Y axis to the index finger, and that of the Z axis to the middle finger.

ISO 841 specifies that the **Z axis** should be defined according to the **direction of the tool spindle,** whereby the positive Z direction always points **from the workpiece to the tool.**



*) after the French mathematician René Descartes, in Latin Renatus Cartesius (1596 – 1650).

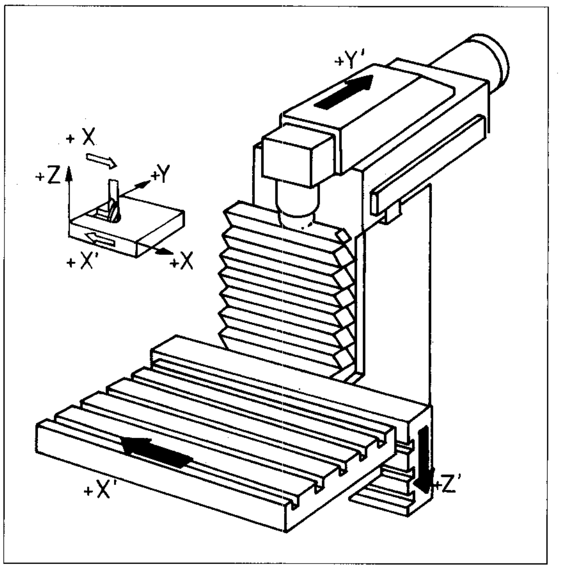
Coordinates
Datum

Relative tool movement

Part programs are always written with workpiece-based coordinates X, Y, Z. That is, they are written as if the tool moves and the workpiece remains still, independent of the machine type.

If, however, the work support on a given machine actually moves in any axis, then the direction of the coordinate axis and the direction of traverse will be opposite.

In such a case the machine axes are designated as X', Y' and Z'.



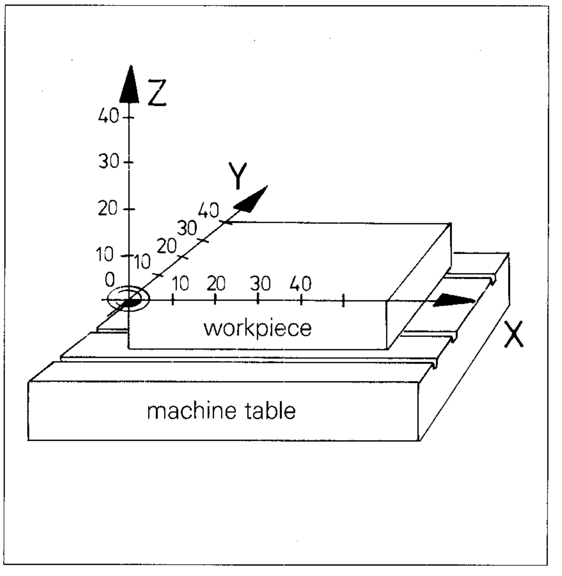
Zero point of the coordinate system

For the **zero point of the coordinate system,** the position on the workpiece which corresponds to the datum of the part drawing is generally chosen – that is, the point to which the part dimensioning is referenced.

For reasons of safety, the workpiece datum in the Z axis is almost always positioned at the highest point on the workpiece.

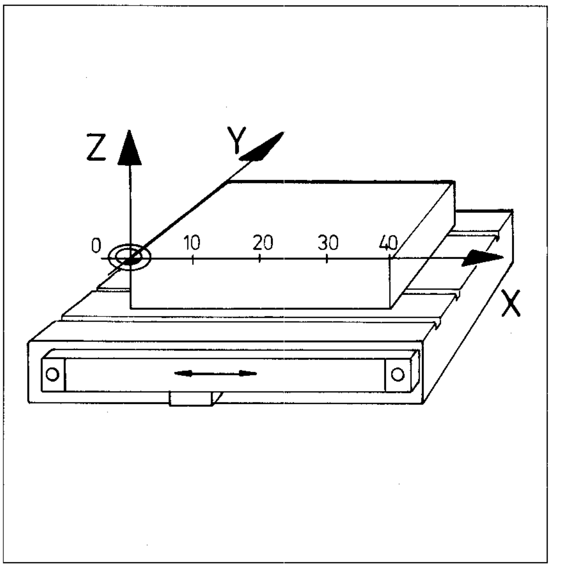
The datum position indicated in the drawing to the right is valid for all programming examples in this manual.

Machining operations in a horizontal plane require freedom of movement mainly in the positive X and Y directions. Infeeds starting from the upper edge of the workpiece Z = 0 correspond to negative position values.



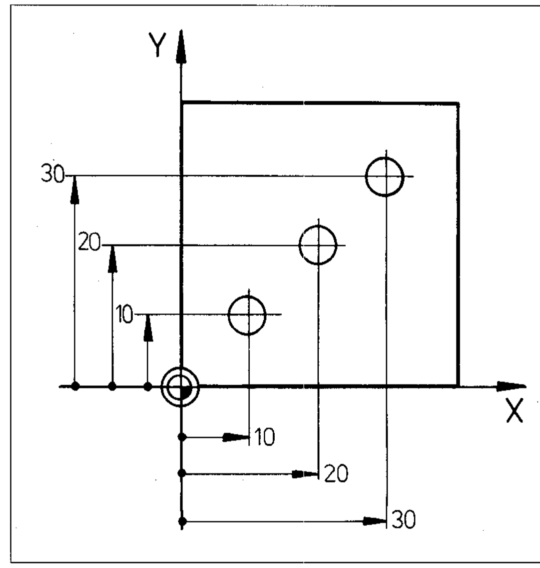
Datum Setting

The workpiece-based rectangular **coordinate system** is defined when the coordinates of any **datum P** are known – that is, when the tool is moved to the datum position and the control "sets" the corresponding coordinates (datum setting).



Absolute and incremental coordinates

If a given point on the workpiece is referenced to the **datum,** then one speaks of **absolute coordinates** or absolute dimensions. It is also possible to indicate a position which is referenced to **another known workpiece position:** in this case one speaks of **incremental coordinates** or incremental dimensions.

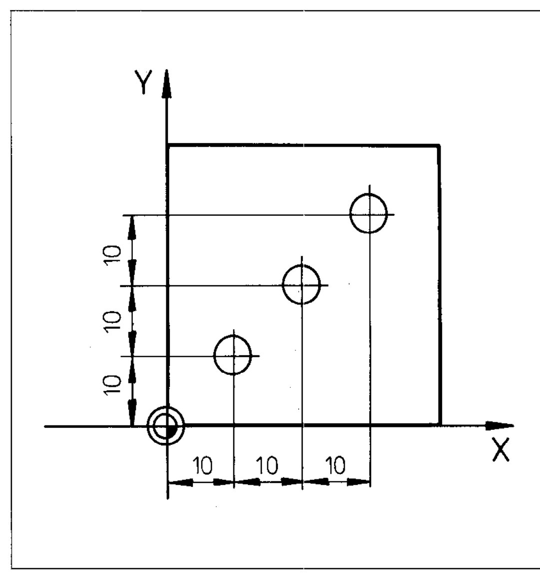


Absolute dimensions

The machine is to be moved **to** a certain position or **to** certain nominal coordinates.

Example: X+30 Y+30

Dimensions in this manual are given as **absolute Cartesian dimensions** unless otherwise indicated.



Incremental dimensions

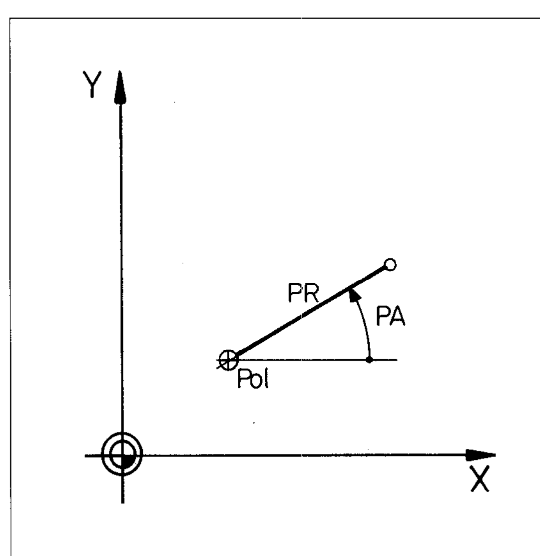
Incremental dimensions in a part program always refer to the **immediately preceding nominal position.** Incremental dimensions are indicated by the letter I.
The machine is to be moved **by** a certain distance: it moves from the previous position along a distance given by the incremental nominal coordinate values.

Example: IX+10 IY+10

Mixing absolute and incremental dimensions

It is possible to mix absolute and incremental coordinates within the same program block.

Example: L IX+10 Y+30



Polar coordinates

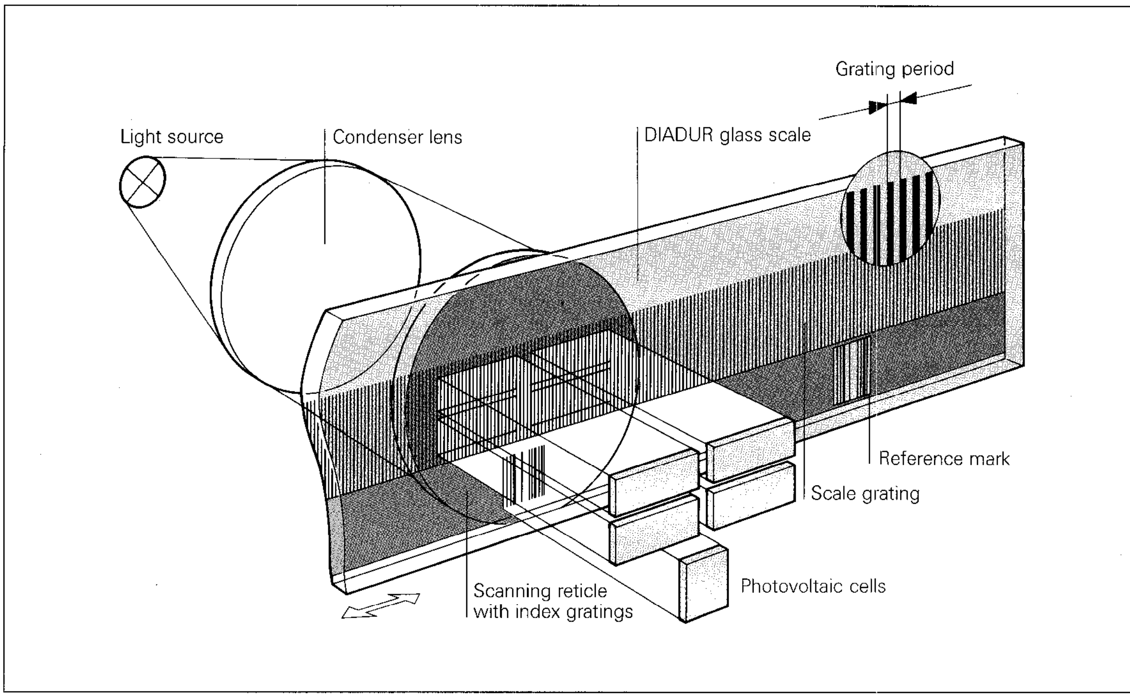
Positions on the workpiece can also be programmed by entering the radius and the direction angle referenced to a pole (see index Programming Modes, Polar coordinates).

CC = Pole
PR = Polar radius (distance from pole)
PA = Polar angle (direction angle)

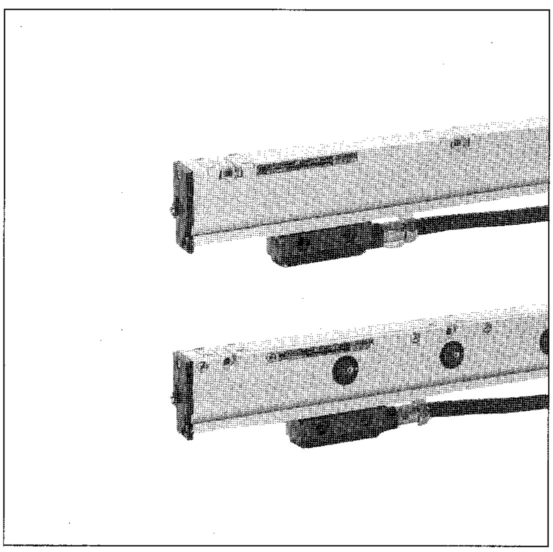
Coordinates
Linear and angle encoders

Linear and angle encoders in machine tools

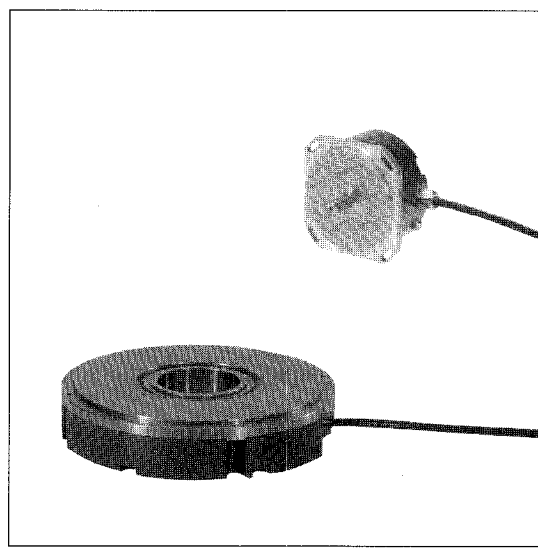
Each machine axis requires a measuring system to provide the control with information on the actual position: linear encoders for linear axes, angle encoders for rotary axes.



Principle of photoelectric scanning of fine gratings



LS 101C, LS 107C



RON 706C, ROD 250C

With **linear axes,** position measurement is generally based on either

- a photoelectrically scanned **steel or glass scale,** or
- the **high-precision spindle,** which also functions as the moving element (the electrical signals are then produced by a rotary encoder coupled to the spindle).

With **rotary axes,** a graduated disk permanently attached to the axis is photoelectrically scanned. The TNC forms the position value by counting the generated impulses.

Coordinates
Linear and angle encoders

Linear and angle encoders are **machine-based:**

Datum

The datum for determination of the nominal and actual position must correspond to the workpiece datum, or be brought into correspondence by **setting** the correct position value (= the position value determined by the workpiece datum) in any axis position. This procedure is called datum setting (or datum presetting).

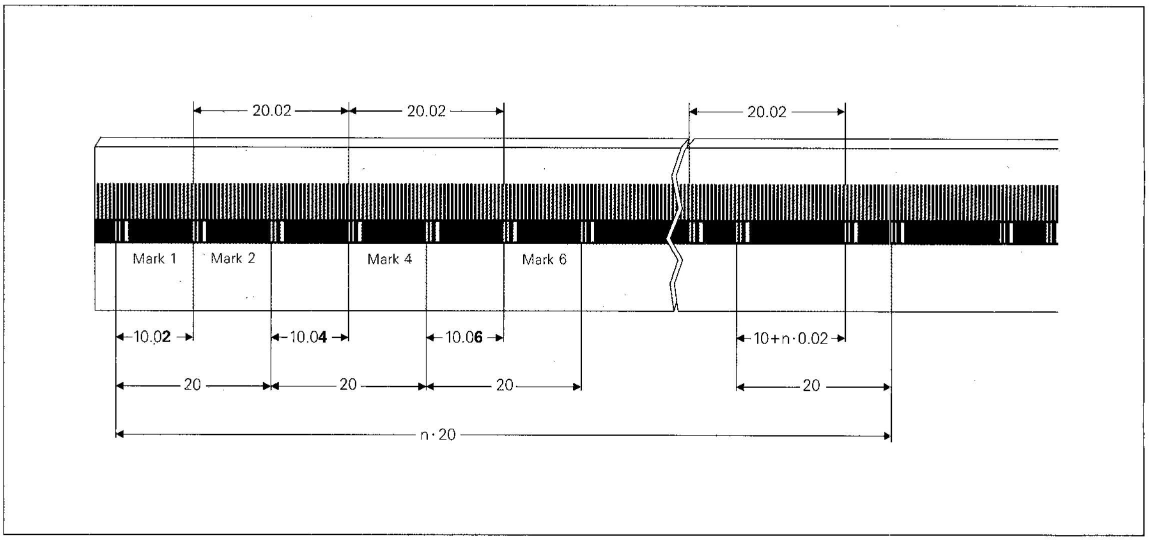
Reference marks

After the control has been switched off or after a power interruption, it is necessary to set the datum again. To simplify this task, the encoders possess **reference marks,** which in a sense also represent datum points.

The relationship between axis positions and position values which were established by the last setting of the workpiece datum (datum setting), are **automatically** retrieved by traversing over the reference marks after switch-on. This also re-establishes the machine-based references such as the software limit switch or tool change position.

In the case of linear encoders with distance-coded reference marks, the machine axes need only be traversed by a maximum of 20 mm. For angle encoders with distance-coded reference marks, a rotation of just 20° is required.

Linear encoders with only one reference mark have an **"RM"** label which indicates the position of the reference mark, while angle encoders with one reference mark indicate the position with a notch on the shaft.



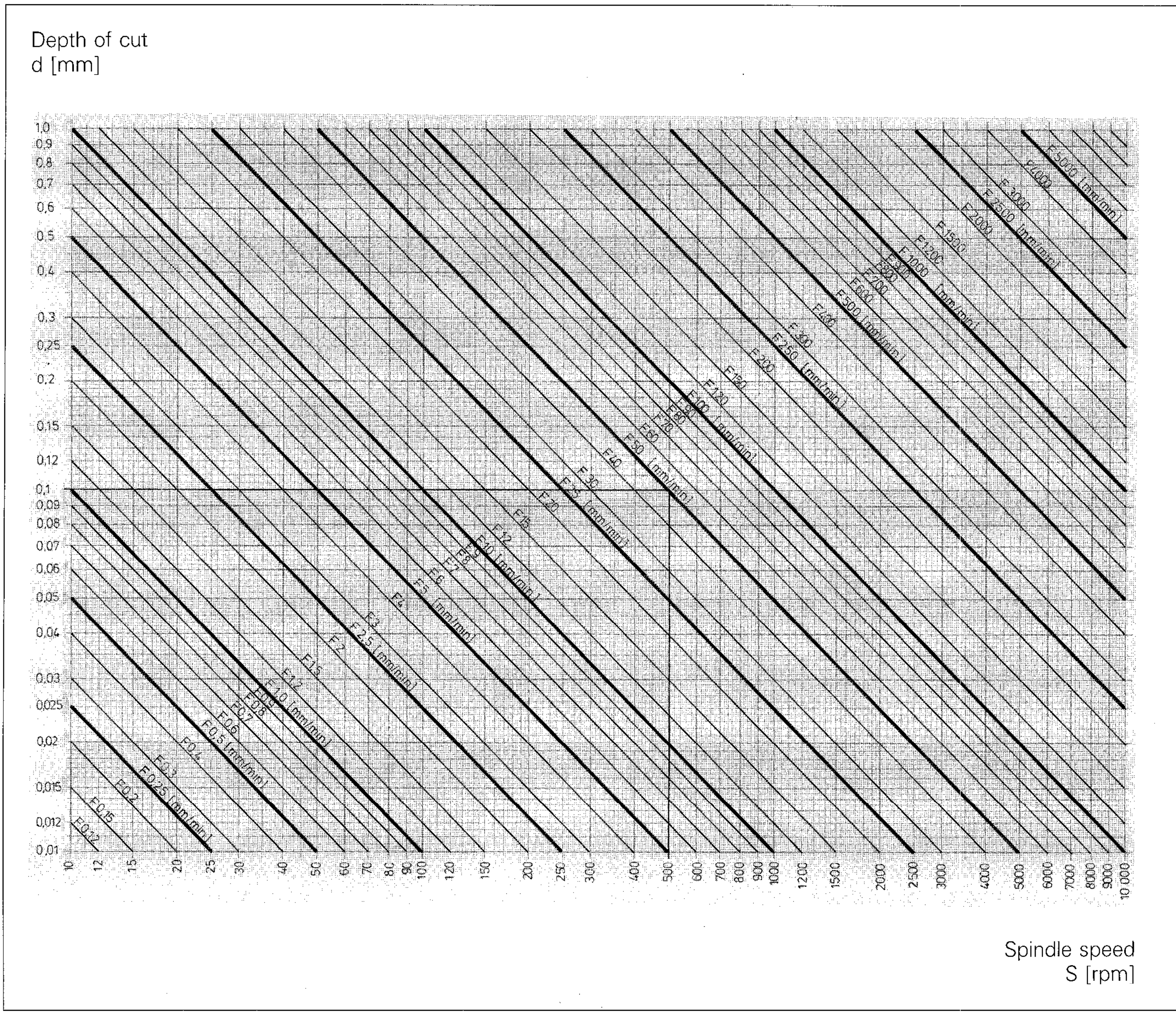
Schematic of scale with distance-coded reference marks

Cutting Data
Feed rate diagram

The feed rate F must be defined in [mm/min] in the program. Usually, the number of teeth n on the tool, the permitted chip thickness d per tooth and the previously determined spindle speed S are given. The diagram below helps you determine the feed rate F.

Determine the required **feed rate F in [mm/min]**

		Example
Given:	n = number of teeth	6
	d = permitted depth of cut per tooth	0.1 [mm]
Selected:	S = spindle speed	500 [rpm]
Find:	F = feed rate	



Calculation

Horizontal line through depth of cut 0.1 mm
Vertical line through cutting speed 500 m/min

At the point of intersection, read off the feed rate F = 50 [mm/min]; this is multiplied by the number of teeth n = 6: F = 300 mm/min

Formula

$d = \frac{F}{S \cdot n}$ or $F = d \cdot S \cdot n$

Prerequisites:

The feed rate determination assumes that

- the tool axis infeed = 1/2 tool radius

or

- the lateral infeed = 1/4 tool radius and the downfeed is selected equal to the tool radius.

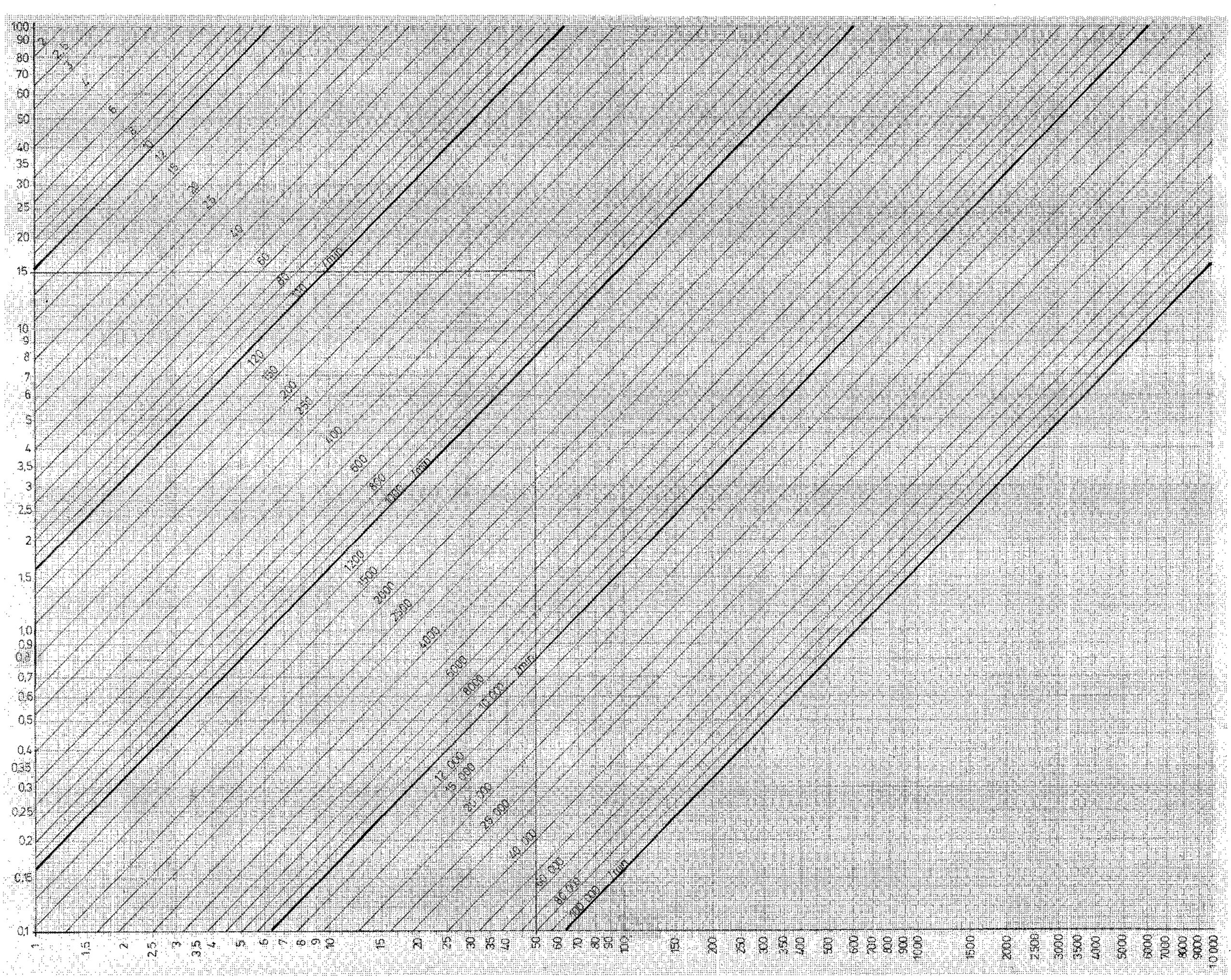
Cutting Data
Spindle speed diagram

The spindle speed S must be defined in [rpm] in the program. Usually the tool radius R is given in [mm] and the cutting speed V in [m/min]. The diagram below helps you determine the spindle speed S.

Determining the required **spindle speed S in [rpm]**

	Example
Given: R = tool radius	16 [mm]
V = cutting speed	50 [m/min]
Find: S = spindle speed	

Tool radius
R [mm]



Cutting speed
V [m/min]

Calculation Horizontal line through the tool radius R = 16 mm
Vertical line through the cutting speed V = 50 m/min

Read off the value at the point of intersection: approx. 500 rpm (calculated: 497 rpm)

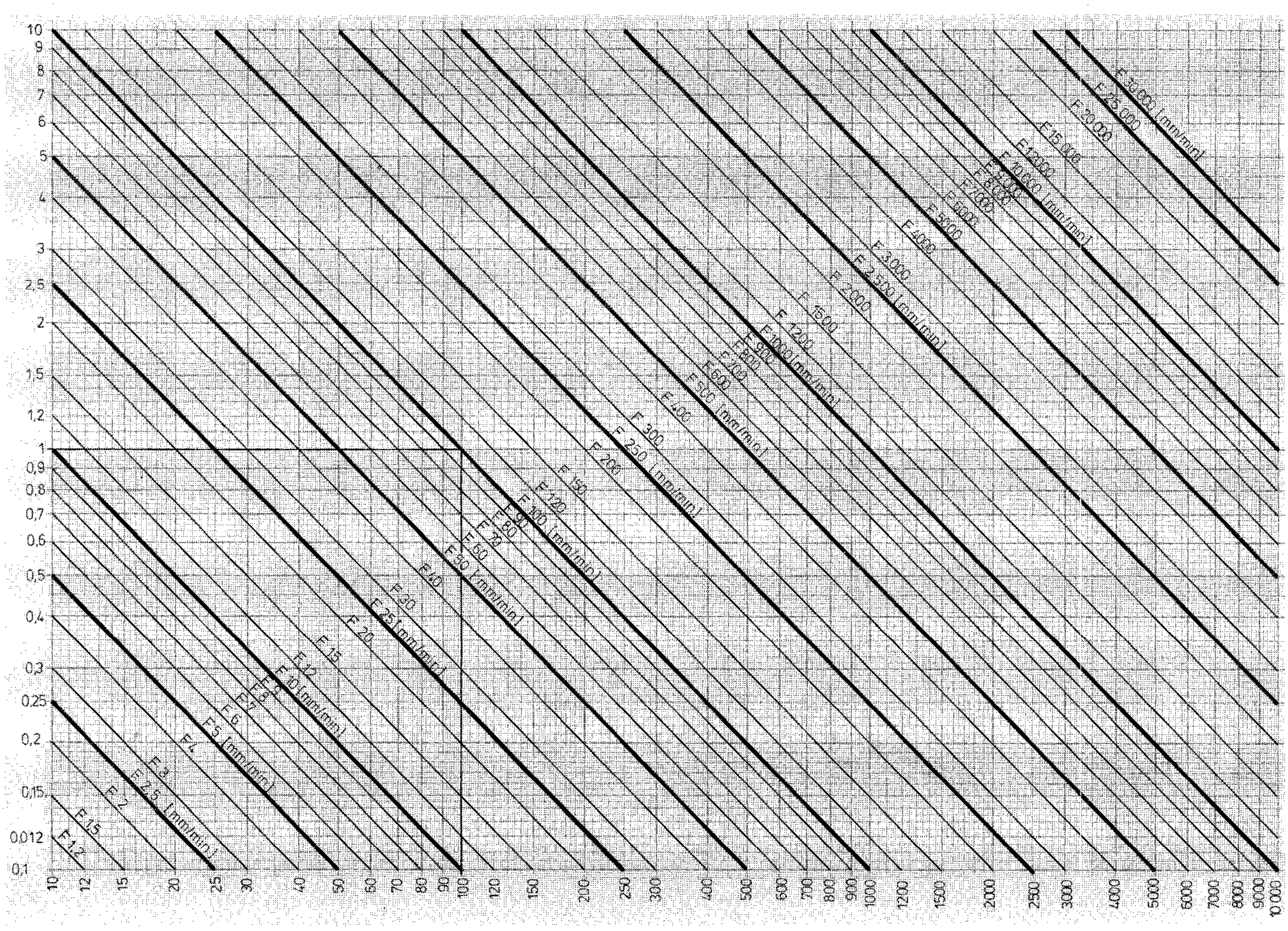
Formula $V = 2R \cdot \pi \cdot S; \qquad S = \frac{V}{2R\,\pi}$

Cutting Data
Feed rate diagram for tapping

When tapping a thread, the pitch P is given [mm/rev]. The spindle speed S and the feed rate F must be defined in the program. First, the spindle speed is determined in the appropriate diagram, then the feed rate is found in the diagram below.

Determine the required **feed rate F in [mm/min]**

		Example
Given:	p = pitch [mm/rev]	1 [mm/rev]
Selected:	S = spindle speed [rpm]	100 [rpm]
Find:	F = feed rate [mm/min]	



Calculation

Horizontal line through pitch p = 1.0 mm/rev
Vertical line through spindle speed S = 100 rpm

Read off feed rate at point of intersection:
F = 100 mm/min

Formula

$p = \frac{F}{S}$ or $F = p \cdot S$

Machine Operating Modes (M)

Switch-On
Traversing the reference points

Switch-On

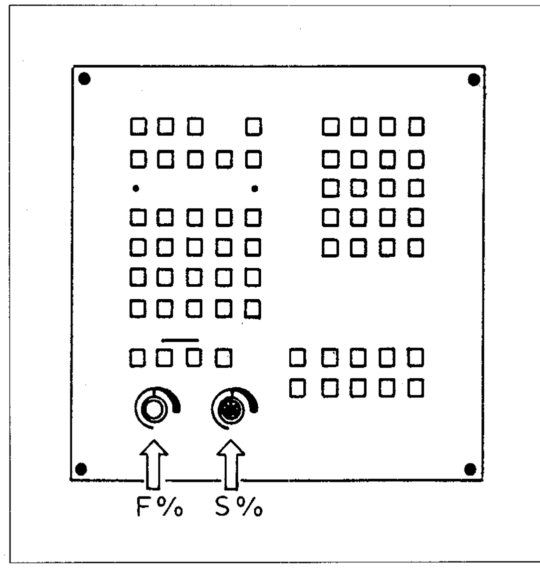
	Switch power on.
MEMORY TEST	The TNC tests the internal control electronics. The display is automatically cleared.
POWER INTERRUPTED	CE Delete the message. The control then tests the EMERGENCY STOP circuit.
RELAY EXT. DC VOLTAGE MISSING	I Switch on the control DC voltage.
MANUAL OPERATION TRAVERSE REFERENCE POINTS Z AXIS X AXIS Y AXIS 4th AXIS	START Traverse the axes over the reference points in the displayed sequence. Start each axis separately or X Y Z ... move the axes with the external direction keys. The sequence of the axes is determined by the machine manufacturer.
MANUAL OPERATION	"Manual operation" is now selected automatically.

Encoders

The required traversing distance for linear and angle encoders with distance-coded reference marks is max. 10 mm or 20 mm/10° or 20°.
If the encoder has only one reference mark, it must be traversed.

Manual Operation
Traversing with the axis direction buttons/ Spindle speed S/Miscellaneous functions M

The machine axes can be moved and the datum set in the "Manual" operating mode.



Jog mode

X

The machine axis moves as long as the corresponding external axis direction button is held down. Several axes can be driven simultaneously in the jog mode.

Continuous operation

X START STOP

If the machine "START" button is pressed simultaneously with an axis direction button, the selected machine axis continues to move after the two buttons are released. Movement is stopped with the external "STOP" button.

Feed rate override

The traverse speed (feed rate) is preset by machine parameters and can be varied with the feed rate override (F %) of the control.

Spindle speed S

TOOL CALL

The spindle speed can be selected with "TOOL CALL".

Spindle override

On machines with continuously variable spindle drives, the speed can also be varied with the spindle override (S %).

Initiate the dialog TOOL CALL

SPINDLE SPEED S RPM ?	Key in the spindle speed.
	ENT Confirm entry.
	START Switch on the spindle.

Miscellaneous function M

Use the "STOP" key to enter a miscellaneous function:

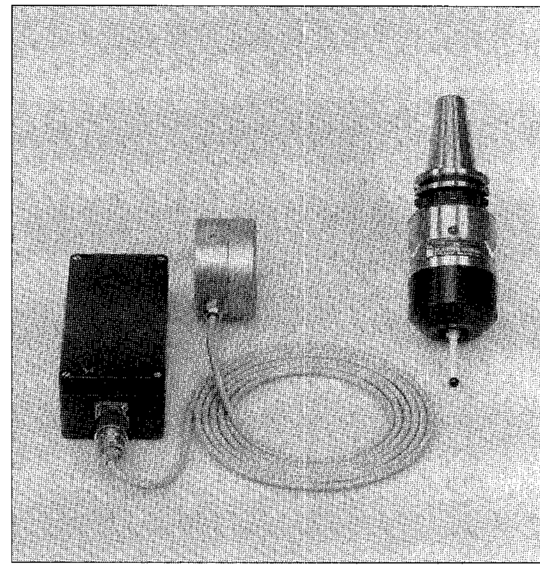
Initiate the dialog STOP

MISCELLANEOUS FUNCTION M ?	Key in the M function.
	ENT Confirm entry.
	START Activate the miscellaneous function.

3D Touch Probe
Datum setting with probe system

Using the touch probe for setup

For workpiece setup the 3D touch probe systems from HEIDENHAIN in association with TNC software offer considerable benefits. One is that the workpiece does not have to be aligned precisely to the machine axes: The TNC will determine and compensate misalignment automatically ("basic rotation"). Another important benefit of the 3D touch probe systems is significantly faster and more accurate datum setting.



TS 511

Probing functions

TOUCH PROBE

The touch probe functions described below can also be employed in the "electronic handwheel" operating mode.

Pressing the "TOUCH PROBE" key calls the menu shown here to the right. The probing function is selected with the cursor keys and entered with the "ENT" key.

```
CALIBRATION EFFECTIVE LENGTH
CALIBRATION EFFECTIVE RADIUS
BASIC ROTATION
SURFACE = DATUM
CORNER = DATUM
CIRCLE CENTER = DATUM
```

Calibration

The effective length of the probe and the effective radius of the probing ball must be calibrated once, before beginning touch probe work. Both dimensions are determined by CALIBRATION routines and stored in the control.

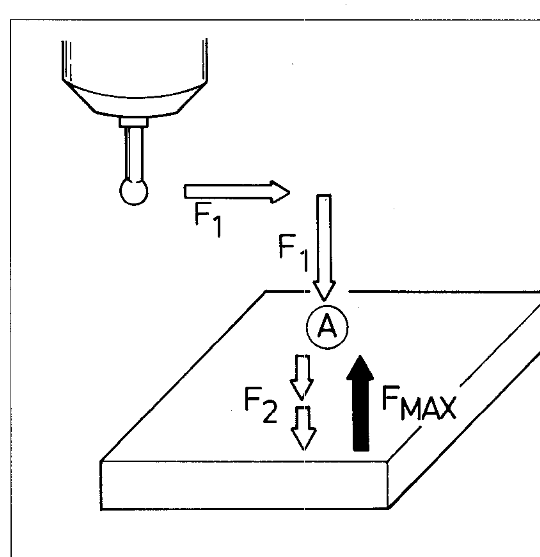
Terminating the probing functions

The probing functions can be terminated at any time with "END □".

Calibrating/working procedure

The probe head traverses to the side or upper surface of the work. The feed rate during measurement and the maximum measuring distance are set by the machine manufacturer via machine parameters.

The touch probe system signals contact with the workpiece to the control. The control stores the coordinates of the contacted points. The probing axis is stopped and retracted to the starting point. Overrun caused by braking does not affect the measured result.



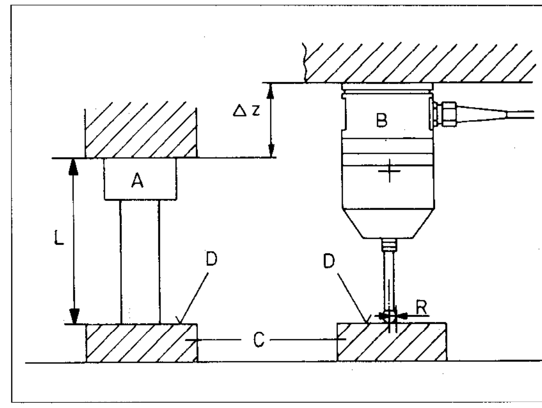
Ⓐ = prepositioning with the external axis direction buttons.
F1 = feed rate for prepositioning.
F2 = feed rate for probing.
FMAX = retraction in rapid traverse.

3D Touch Probe
Calibrating effective length

Work aid: ring gauge

For calibration of the effective length, a ring gauge of known height and known internal radius is clamped to the machine table.

A = zero tool
B = 3D touch probe
C = ring gauge
D = reference plane (surface)
L = length of the zero tool
R = ball tip radius

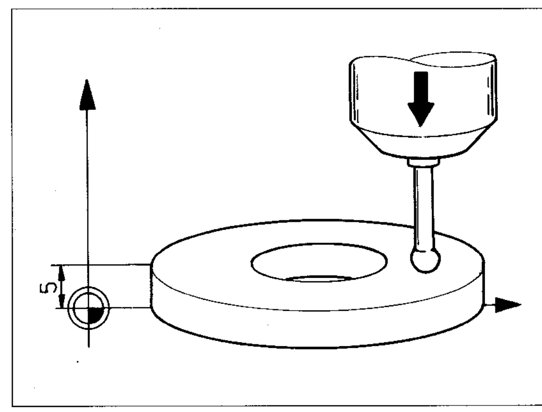


Procedure

The reference plane is set with the zero tool prior to calibration.

To determine the effective length of the stylus, the probe head touches the reference plane. After contacting the surface, the probe head is retracted in rapid traverse to the starting position.

The length L is stored by the control and automatically compensated during the measurements.



Initiate the dialog — TOUCH PROBE

CALIBRATION EFFECTIVE LENGTH — ↑ ENT Select probing function and enter.

TOOL AXIS = Z	Enter a different tool axis if required.
	↓ Select the "Datum".
DATUM +5	ENT Enter the datum in the tool axis, e.g. +5.0 mm.

Z+ Z–	X Y Z Move the touch probe to the vicinity of the reference plane. ← → Select the direction of probe movement, here Z–. START The probe head moves in negative Z direction. After touching the surface and returning to the starting position, the control automatically switches to the "Manual operation" or "Handwheel" operating mode.

Display

The value for effective length can be displayed by selecting "Calibration effective length" again.

3D Touch Probe
Calibrating effective radius

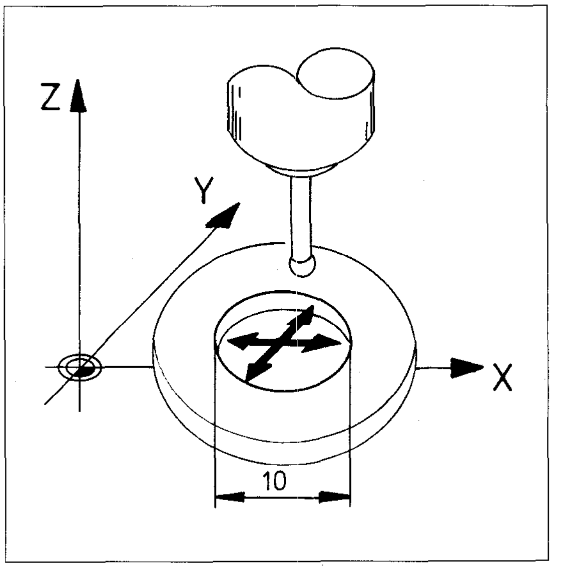
TOUCH PROBE

Procedure

The probe ball is lowered into the bore of the ring gauge. 4 points on the wall must be touched to determine the effective radius of the stylus ball. The traverse directions are determined by the control, e.g. X+, X–, Y+, Y– (tool axis = Z).

The probe head is retracted in rapid traverse to the starting position after every deflection.

The radius R is stored by the control and automatically compensated during the measurements.



Initiate the dialog — TOUCH PROBE

CALIBRATION EFFECTIVE RADIUS — ↑ ENT Select probing function and enter.

TOOL AXIS = Z	☐ Enter another tool axis if required.
	↓ Select "Radius ring gauge".
RADIUS RING GAUGE = 10	☐ ENT Enter the radius of the ring gauge, e.g. 10.0 mm.

X+ X– Y+ Y–	X Y Z Traverse approximately to the center of the ring gauge. ← → Select the traversing direction of the probe head (only necessary if you prefer a certain sequence or the exclusion of one probing direction). START Probe a total of 4 times. After contacting the wall of the ring gauge four times, the control automatically switches to the "Manual operation" or "Handwheel" operating modes.

Display

You can display the value for effective radius by selecting "Calibration effective radius" again.

Error messages

All touch probe systems:

TOUCH POINT INACCESSIBLE
The stylus was not deflected within the measuring distance (machine parameter).

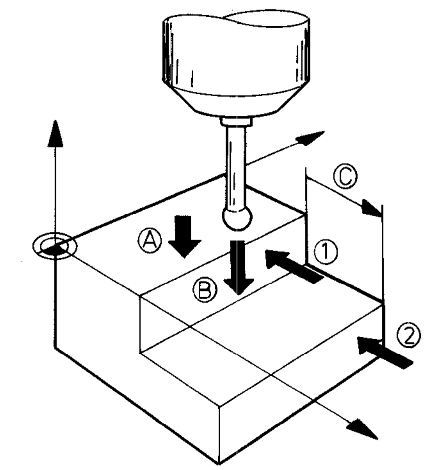
STYLUS ALREADY IN CONTACT
The stylus was already deflected at the start.

Touch probe system TS 511:

PROBE SYSTEM NOT READY
Probe system not set up correctly, or transmission path was interrupted.
The transmitter and receiver window (i.e. the side with two windows) must be pointed towards the transmitter/receiver unit.

3D Touch Probe
Reference surface, Position measurement

The position of a surface on the clamped workpiece is determined with the probing function "Surface = datum".



Functions

- Setting the reference plane Ⓐ
- Measuring positions Ⓑ
- Measuring distances Ⓒ

Measuring positions

Initiate the dialog — TOUCH PROBE

SURFACE = DATUM — ↑ ENT Select probing function and enter.

X+ X– Y+ Y– Z+ Z– C+ C–	X Y Z Move to the starting position. ← → Select the traversing direction, e.g. Z–. START Move the probe head in negative Z direction. The probe head is retracted in rapid traverse to the starting position after touching the surface.

Measured value

DATUM Z–18,125	The control displays the measured value.

Setting the reference plane

DATUM Z+0	Enter a new value if required, e.g. 0 mm. ENT Confirm entry.

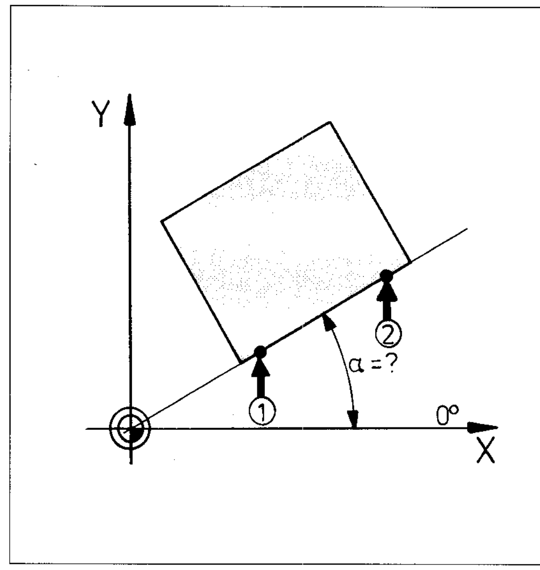
Measuring distances

You can also measure distances on an aligned workpiece.

- Probe the first position and set the datum (e.g. 0 mm).
- Probe the second position.
  The distance can be read in the "Datum" display.

3D Touch Probe
Basic rotation, Angular measurement

The probing function "Basic rotation" determines the angle of deviation of a plane surface from a nominal direction. The angle is determined in the machining plane.



Functions

- Basic rotation
  (the control compensates for an angular misalignment)
- Correct an angular misalignment
  (on a machine tool with rotary axis)
- Measure an angle.

Basic rotation

Initiate the dialog — TOUCH PROBE

BASIC ROTATION — Select probing function and enter.

ROTATION ANGLE = 0

- Select the "Rotation angle".
- Enter the nominal direction of the surface to be probed, e.g. 0°.

X+ X– Y+ Y–

- X Y Z: Move the probe head to the starting position ①.
- Select the probing direction, e.g. Y+.
- START: The probe head travels in the selected direction, e.g. Y+. The probe head returns to the starting position after touching the side surface.

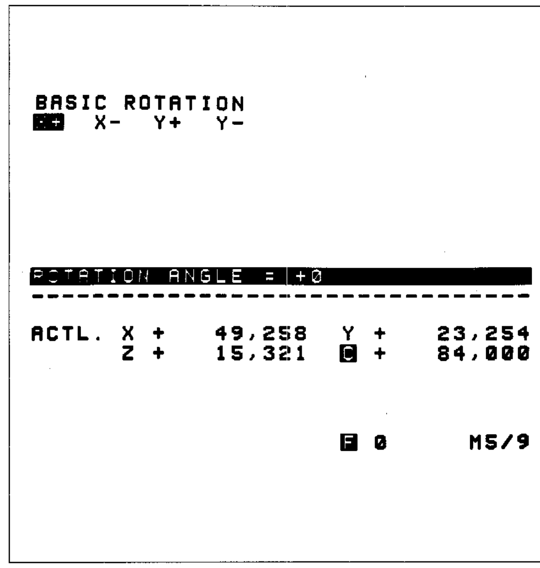
- X Y: Move the probe head to the starting position ②.
- START: The probe head travels in the selected direction, e.g. Y+. The probe head returns to the second starting position after making contact. The control automatically switches to the "Manual operation" or "Handwheel" operating mode.

3D Touch Probe
Basic rotation, Angular measurement

Displaying the rotation angle

The measured rotation angle is displayed by selecting the probing function "Basic rotation".

Compensation of angular misalignment is registered on the screen with "ROT" in the status display. It also remains stored after a power interruption.



Cancelling the basic rotation (rotation angle 0°)

The basic rotation is cancelled by selecting the probing function "Basic rotation" and entering a 0° rotation angle.
The "ROT" display is cleared.

Once basic rotation is activated, all subsequent programs are executed with rotation and shown rotated in the graphic simulation.

Measuring angles

In addition to basic rotation, angle measurements can also be performed on aligned workpieces.

Carry out the following procedure:

- Execute a basic rotation.
- Display the rotation angle.
- Cancel the basic rotation.

Compensating for misalignment

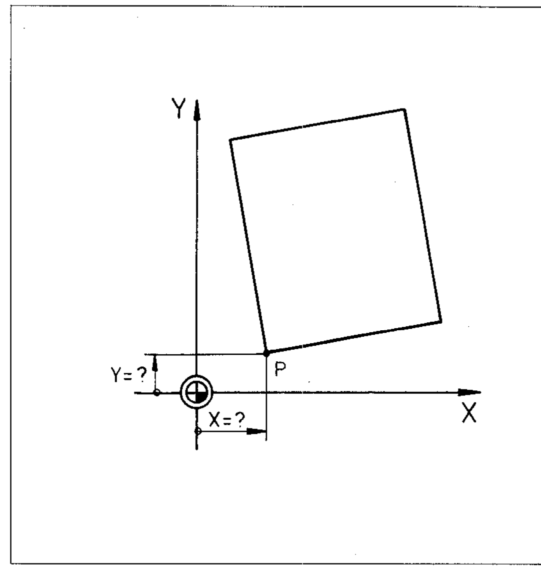
On machine tools with a rotary axis, you can also correct misalignment of a workpiece by rotating the axis.

Carry out the following procedure:

- Execute a basic rotation.
- Display and note the rotation angle.
- Cancel the basic rotation.
- Enter the noted value for the rotary axis **incrementally** in the "Positioning with MDI" operating mode (cf. Positioning at the entered position without radius compensation) and start the rotation with the machine "START" button.

3D Touch Probe
Corner = datum/ Determining corner coordinates

With the probing function "Corner = datum", the control computes the coordinates of a corner on the clamped workpiece. The computed value can be taken as datum for subsequent machining. All nominal positions then refer to this point.

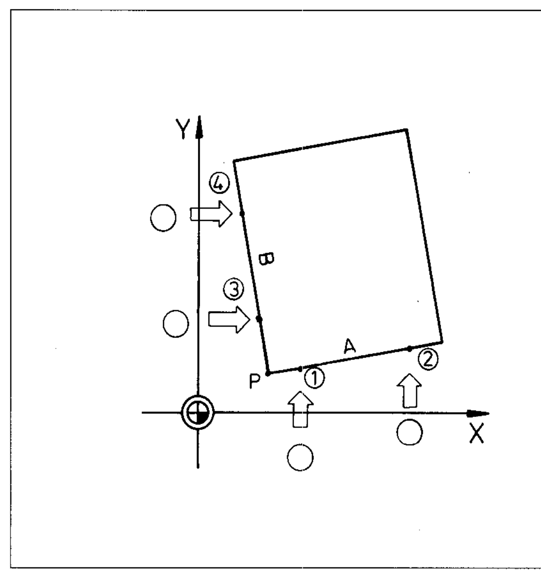


The probing function "Basic rotation" should be performed before "Corner = datum".

Procedure

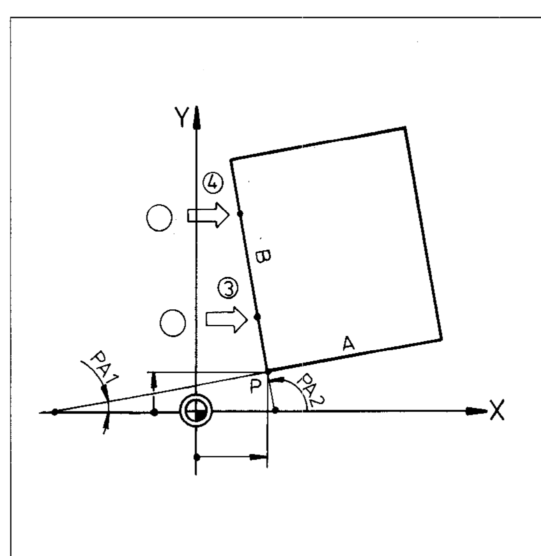
The probe head touches two side surfaces (see figure) from two different starting positions per side.

The corner point P is computed by the control as the intersection of straight line A (contact points ① and ②) with straight line B (contact points ③ and ④).



After performing a basic rotation

If the probing function "Corner = datum" is called after performing a basic rotation (straight line A), the first side need not be contacted.



3D Touch Probe
Corner = datum/
Determining corner coordinates

To transfer the direction of the first side face from the routine "basic rotation", simply respond to the dialog query TOUCH POINTS OF BASIC ROTATION ? by pressing the "ENT" key (otherwise "NO ENT").

> If only the probing function "CORNER = DATUM" is performed, then it does not contain a basic rotation.

Initiate the dialog — TOUCH PROBE

CORNER = DATUM — Select probing function and enter.

First side face

X+ X– Y+ Y–

- X Y Z: Move the probe head to the first starting position.
- ← →: Select the probing direction, e.g. Y+.
- START: The probe head travels in the selected direction. After touching the side face, the probe head is retracted to the starting position.

Traverse to the second starting position and probe in the same probing direction as described above.

Second side face

X+ X– Y+ Y–

- X Y Z: Move the probe head to the third starting position.
- ← →: Select the probing direction, e.g. X+.
- START: The probe head travels in the selected direction. After touching the side face, the probe head is retracted to the starting position.

Traverse to the fourth starting position and probe in the same probing direction as described above.

Display corner coordinates/ Setting the datum

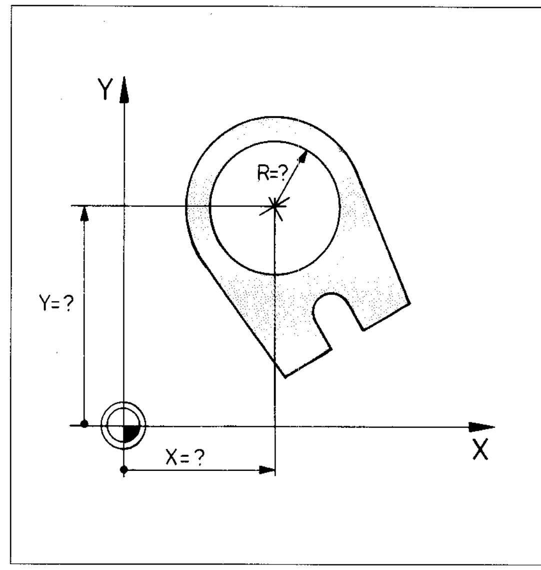
DATUM X+0

DATUM Y+0

- Enter the corner coordinates for X and Y if required, e.g. X+0, Y+0.
- ENT: Confirm entries.

3D Touch Probe
Circle center = datum/ Determining the circle radius

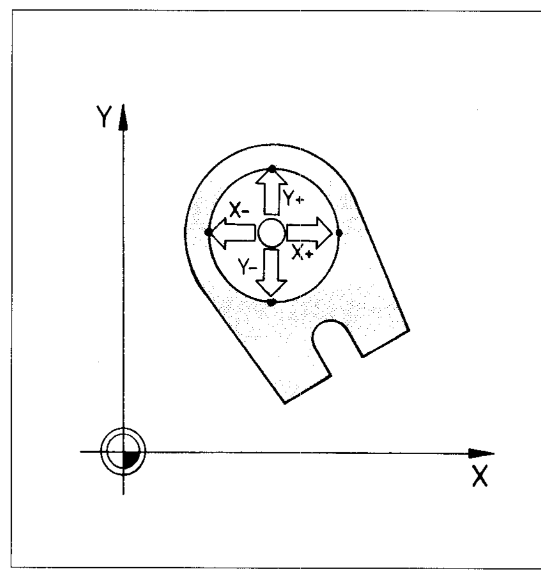
In the probing function "Circle center = datum", the control computes the coordinates of the circle center and the circle radius on a clamped workpiece with cylindrical surfaces. The coordinates of the center can be used as the datum for subsequent machining. All nominal positions are then referenced to this point.



The "Basic rotation" probing function must be carried out prior to "Circle center = datum".

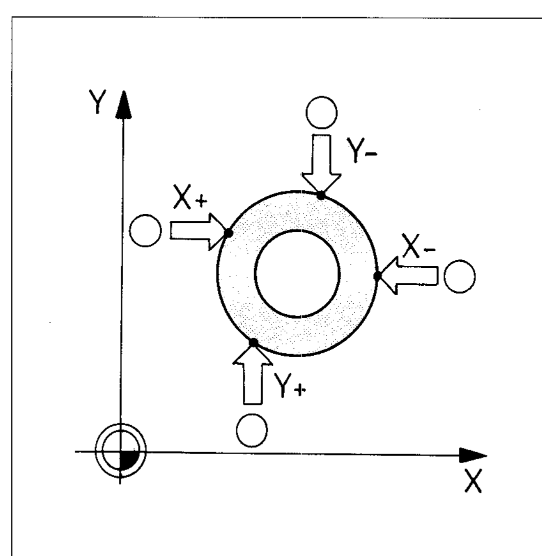
Bore, Circular pocket

Position the probe head in the bore with the remote axis direction keys. 4 different positions are then touched by pressing the machine START button.



Outer cylinder

On workpieces with cylindrical outer surfaces, the probing directions must be specified for each of the four points.



3D Touch Probe
Circle center = datum/ Determining the circle radius

Initiate the dialog	TOUCH PROBE
CIRCLE CENTER = DATUM	↓ ENT Select the probing function and enter.
X+ X– Y+ Y–	X Y Z Move the probe head to the first starting position. ← → Select the probing direction if required, e.g. X–. START Probe head travels in the selected direction. After touching face, the probe head is retracted to the starting position.
	Traverse to the second and third starting positions and probe in different directions as described above.
X+ X– Y+ Y–	X Y Z Move the probe head to the fourth starting position. ← → Select the probing direction if required, e.g. Y–. START The probe head travels in the selected direction. The probe head is retracted to the starting position after touching the side face.

Display

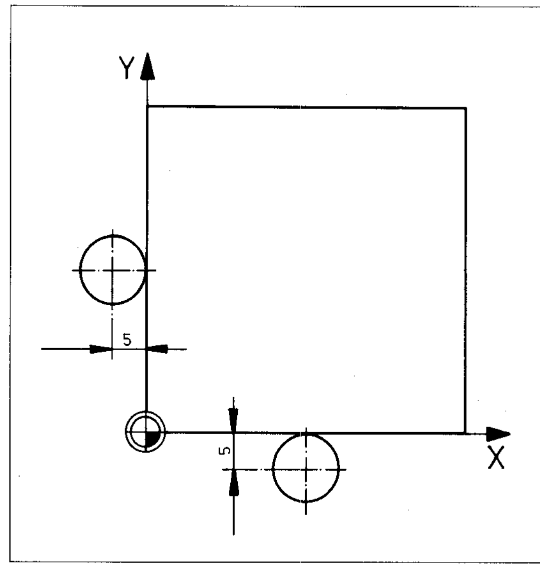
X+54.3 Y+21.576	Coordinates of the circle center.
PR+20	Circle radius.

Datum setting

DATUM X+40	Enter the X and Y coordinates of the circle center if necessary, e.g. X+40, Y+30.
DATUM Y+30	↓
	ENT Confirm entries.

Align workpiece and set datum

First align the workpiece parallel to the machine axes in the conventional way. For datum setting the machine is then moved to a known position relative to the workpiece and the relevant position values are set with the axis keys.

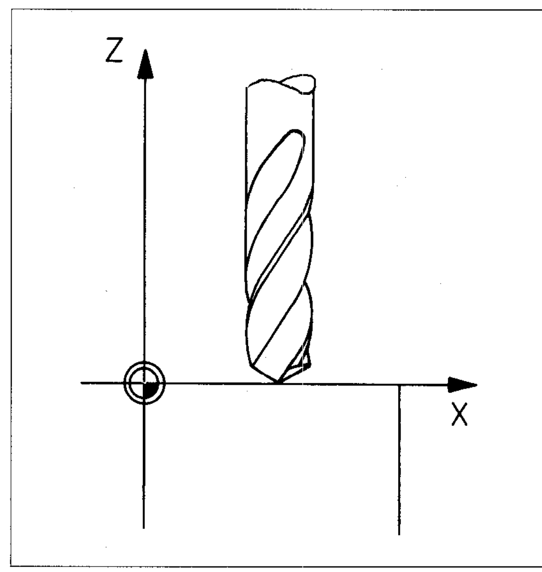


Touching in the machining plane

Touch both sides of the workpiece with a tool or edge finder and, at contact, set the actual position display of the associated axis to the tool radius or the ball tip radius of the edge finder with a **negative** sign (here e.g. X = –5 mm, Y = –5 mm).

Touching in the feed axis (spindle axis)

The actual position display is set to zero when the zero tool touches the work surface.
If contact with the work surface is not allowed, you can lay a metal shim of known thickness (e.g. 0.1 mm) on it. Then enter the thickness of the metal when contact is made (e.g. Z = +0.1 mm).

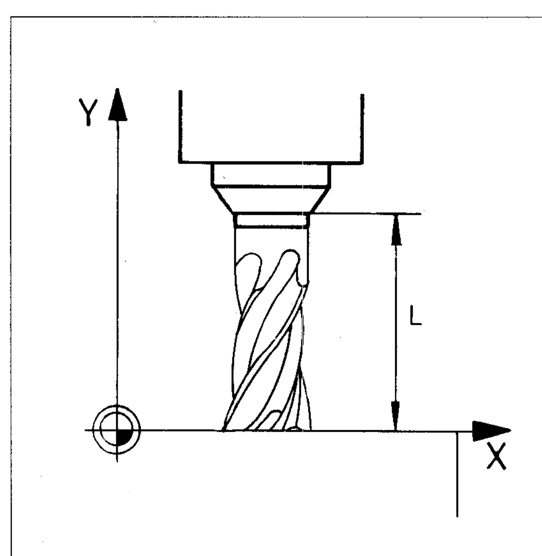


Preset tools

When using preset tools, i.e. when the tool lengths are already known, touch the work surface with any tool. To assign the value 0 to the surface, enter the length L of the inserted tool with a positive sign as the actual value for the infeed axis. If the work surface has a value other than 0, enter the following actual value:
(actual value Z) = (tool length L) + (surface position)

Example:
tool length L: 100 mm
position of the work surface: +50 mm

actual value to be entered:
Z = 100 mm + 50 mm = 150 mm



Manual Operation
Datum setting without probe system

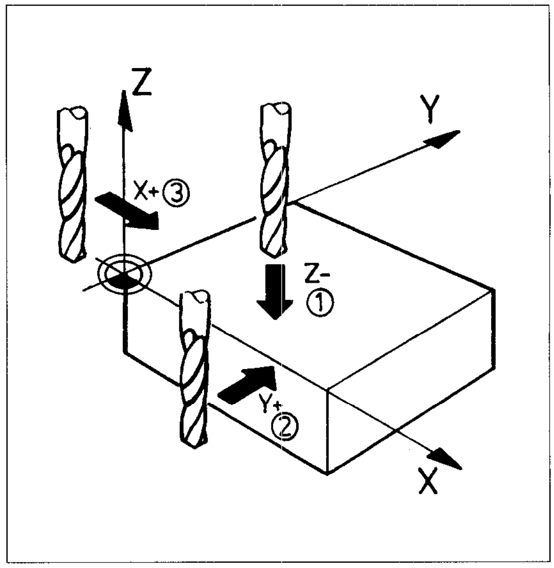
Example: Setting the datum

The datum is to be set with a drill (tool radius R = 5 mm) as shown to the right.

① Touch the workpiece surface.

② Touch side by moving the Y axis.

③ Touch side by moving the X axis.



Touching with Z axis

Initiate the dialog — Z, after surface ① is touched.

DATUM SET Z =	▢ Enter the value for the Z axis, e.g. 0 mm. ENT Confirm entry. The Z display reads: 0.000

Y axis

Initiate the dialog — Y, after surface ② is touched.

DATUM SET Y =	▢ Enter the value for the Y axis, e.g. 5 mm. +/– Here with a negative sign. ENT Confirm entry. The Y display reads: –5.000

X axis

Initiate the dialog — X, after surface ③ is touched.

DATUM SET X =	▢ Enter the value for the X axis, e.g. 5 mm. +/– Here with a negative sign. ENT Confirm entry. The X display reads: –5.000

The datum for the fourth axis can be set in a similar way.

If the dialog DATUM SET was opened by mistake, the dialog can be cleared with "NO ENT" or "END □".

The set datum is only shown in the "ACTUAL" position display.
This display may have to be selected with "MOD" (see index General Information/MOD Functions – Position displays).

Electronic Handwheel/Jog Increment

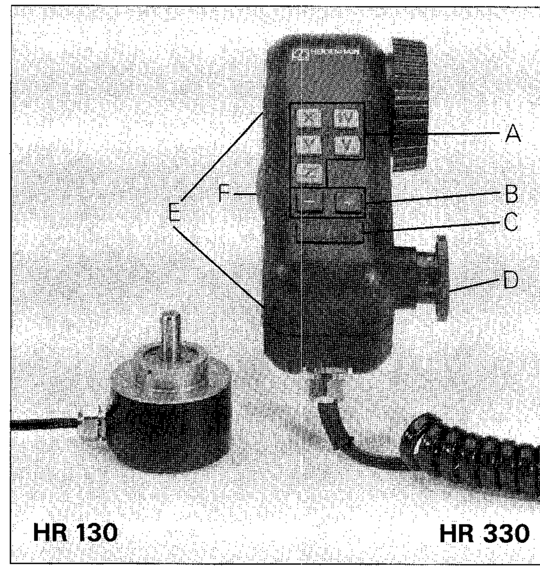
Versions

The control is usually equipped with an electronic handwheel. It can be used, for example, to set up the machine.

There are two versions of the electronic handwheel:

HR 130: to be incorporated into machine operating panel

HR 330: portable version with
axis selection keys,
axis direction keys,
rapid traverse key,
EMERGENCY STOP button.



Interpolation factor

The displacement per handwheel turn is determined by the interpolation factor (see table to the right).

Interpolation factor	Displacement in mm per turn
0	20.0
1	10.0
2	5.0
3	2.5
4	1.25
5	0.625
6	0.313
7	0.156
8	0.078
9	0.039
10	0.020

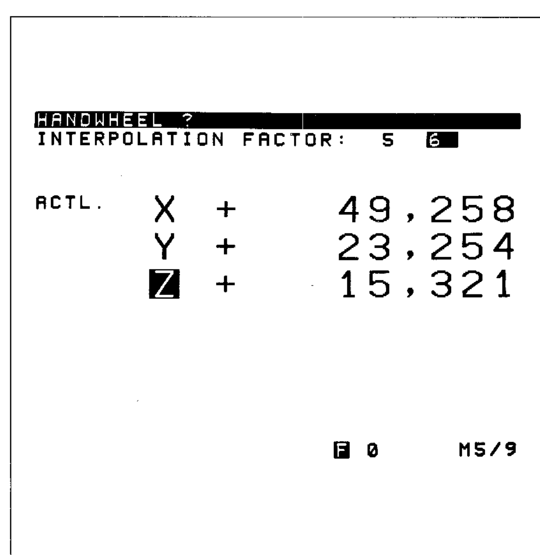
Operating the HR 130

The handwheel is switched to the required machine axis with the axis keys of the control.

Operating the HR 330

The axis is selected on the handwheel.
The axis to be driven by the electronic handwheel is highlighted in the screen display.

In the "Electronic handwheel" operating mode, the machine axes can also be driven with the external axis direction buttons.



Operating the HR 130/330

Set operating mode and initiate the dialog

INTERPOLATION FACTOR: 3	Enter the desired interpolation factor, e.g. 4. Confirm entry.

INTERPOLATION FACTOR: 4	Select the axis: on the control (HR 130) or on the handwheel (HR 330)

The tool can now be moved in a positive or negative Y direction with the electronic handwheel.

Jog positioning

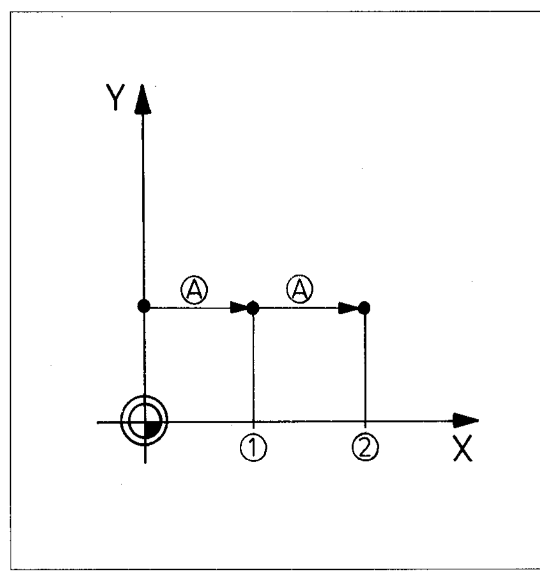
The machine manufacturer can activate jog positioning via the integral PLC. In this case, a traversing increment can be entered in this operating mode.

The axis is moved by the entered increment when you press an external axis button. This can be repeated as often as desired. Only single-axis movements are possible.

Ⓐ Jog increment: e.g. 2 mm.

① External axis button (e.g. X) pressed once.

② External axis button pressed twice.



Entering the jog increment

Set operating mode and initiate the dialog

JOG-INCREMENT: 1.000	Enter the jog increment, e.g. 2 mm. Confirm the entry.

JOG-INCREMENT: 2.000	X or another remote axis key. The axis is driven by the entered jog increment.

Positioning with Manual Data Input
Tool call/Spindle axis/Spindle speed

You must first define (i.e. enter the dimensions of) a tool before you can call it with "TOOL CALL" in the "Positioning with MDI" operating mode. You can define a tool either in the central tool file or in the part program. If in the operating modes "Program run/full sequence" or "Program run/single block" you are working without a central tool file, you must define each tool with "TOOL DEF".

The concepts "TOOL DEF" and "TOOL CALL" are defined in the "Programming and Editing" section under "Tool definition".

Initiate dialog — TOOL CALL

Example: Tool call

TOOL NUMBER ?		Enter tool number.
	ENT	Confirm entry.

Select the spindle axis

WORKING SPINDLE AXIS X/Y/Z ?	Z	Enter spindle axis, e.g. Z.

Select the spindle speed

SPINDLE SPEED S RPM ?		Enter spindle speed.
	ENT	Confirm entry.

BLOCK COMPLETE	START	Start tool call.

Positioning with Manual Data Input
Positioning to entered position

In the operating mode "Positioning with manual data input", single-axis positioning blocks can be entered and executed (the entered positioning blocks are not stored).

Traversing to position

Initiate the dialog — X or another axis key.

POSITION VALUE ?	I Incremental – absolute?
	ENT Enter a numerical value for the selected axis. Confirm the entry.

Radius compensation

TOOL RADIUS COMP.: R+/R–/NO COMP. ?	R+ R– Enter either radius compensation or
	ENT no radius compensation.

FEED RATE ? F = / FMAX = ENT	ENT Enter either the feed rate or
	ENT no value for rapid traverse.

MISCELLANEOUS FUNCTION M ?	ENT Either enter a miscellaneous function, e.g. M03 or
	NO ENT choose no miscellaneous function.

BLOCK COMPLETE	START Start the positioning block.

Terminate block entry

END

Direct termination of input. Data entered previously such as radius compensation, feed rate, or direction of spindle rotation then remain permanently effective.

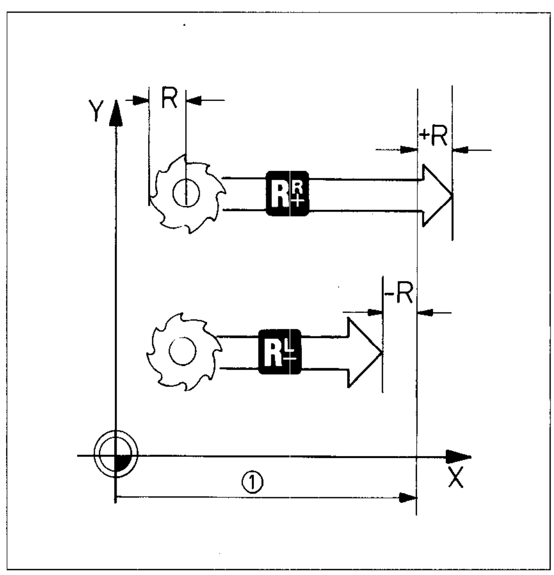
Single-axis radius compensation

For single-axis positioning blocks, you only have to consider whether the tool path is lengthened or shortened by the tool.

R+ tool path to be increased.
R– tool path to be reduced.

> If a radius compensation R+/R– is also entered to position the spindle axis, this axis is not compensated.
>
> A radius compensation is also ignored when the 4th axis is used for a rotary table.

① Nominal position



Program Run
Single block, Full sequence

Stored programs are executed in the operating modes "Program run single block" and "Program run full sequence".

The workpiece datum must be set before machining the work!
See: Datum setting with/without probe system.

Program run single block

In this operating mode, the control executes the part program block by block. The program must be restarted after every block.

Program run single block is best used for program test and for the first program run.

Operating mode — Single block

Selecting the program

	PGM NR	Select the program or, if the program was already selected:
	GOTO 0 ENT	select block 0.

0 BEGIN PGM 7225	The first program block is shown in the current line of the program.

Starting the run

	START	Each program block must be started with the machine START button.

Program run full sequence

In this operating mode, the control executes the machining program until a programmed stop or end of program occurs.

Stop functions:
M02, M30, M00 (M06 "STOP", if assigned a stop function via machine parameter).

The program run is also stopped if an error message appears.

You must restart the program to continue after a programmed stop.

Selecting the program

Operating mode — Full sequence
Select the program and block number as described above.

Starting the run

	START	The program runs continuously until a programmed stop or end of program occurs.

Feed rate

The programmed feed rate can be varied via the feed rate override.

Spindle speed

The programmed spindle speed can be varied via the spindle override (if output is analog).

Program Run
Interrupting the program run

Stop

STOP

	STOP Stop program run: Stop axis movements with the machine STOP button. The block currently being processed is not completed. The "Control in operation" (✱) display blinks.

Abort

STOP

	STOP Interrupt program run. The "Control in operation" (✱) display is cleared.

The control stores:

- the last tool called
- coordinate transformations
- the last valid circle center/pol CC
- the current program section repeat
- the return jump label for subprograms

Switching to single block

In the "Program run full sequence" operating mode, you can interrupt the program run by switching to "Single block".

The block currently being processed is completed.

Program run is to be discontinued after execution of the current block.	To continue, either start each block separately or reactivate "Program run full sequence".

EMERGENCY STOP

The machine can be switched off in an emergency by hitting one of the EMERGENCY STOP buttons.

The control acknowledges this with the message
EMERGENCY STOP

To continue working, release the emergency stop key by turning it clockwise, then

1. Remove the cause of error
2. Switch on the control power again
3. Clear the message EMERGENCY STOP with the "CE" key
4. Restart the program run.

Program Run
Checking/Changing Q parameters

Q parameters You can check and, if necessary, change Q parameters after interrupting the program run.

Interrupt program run

STOP	Stop program run by pressing the machine STOP button.
STOP	Interrupt program run.

Check parameter

Q ENT	Select and check the desired parameter.

Change parameter

ENT	Terminate Q parameter display or
ENT	change the parameter and confirm.

Programming during program execution

While a part program is being executed in the "Program run full sequence" operating mode, **another** program can, in the "Programming and editing" mode, be **simultaneously** either edited or transferred via the data interface RS-232-C/V.24.

This parallel operation is especially advantageous for long programs with little operator activity.

A program cannot be run and edited at the same time.

Starting the part program

Operating mode

Initiate the dialog

PROGRAM NUMBER =	Select part program.

Start machining.

Parallel operating mode: programming and editing

Operating mode

Select and edit the program

or

transfer a program via the RS-232-C/V.24 data interface.

Screen display

The screen is divided into two halves during parallel operation: The program to be edited is shown in the upper half. The program currently in process appears in the lower half: program number, current block number and current status are displayed.

Terminating the parallel operating mode

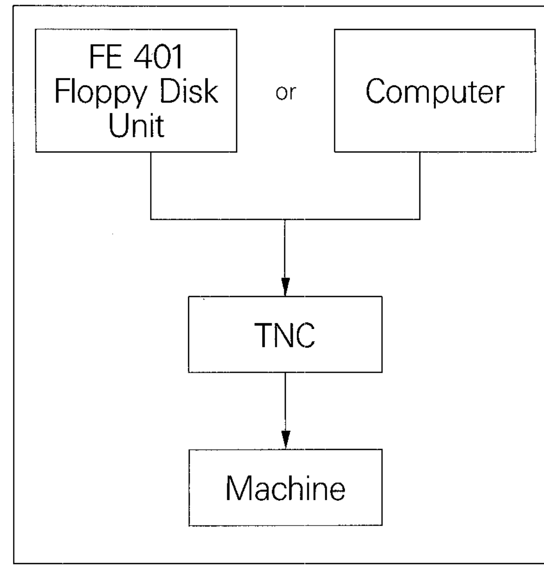
Operating mode

Parallel operating is terminated by pressing the "Program run full sequence" key.

Program Run
Blockwise transfer
(Reloading operation)

Execution from external storage

In the "Program run full sequence" or "Single block" operating mode, part programs can be "transferred blockwise" from a remote computer, a storage medium or a HEIDENHAIN FE unit via the RS-232-C/V.24 serial data interface. This allows execution of part programs which exceed the storage capacity of the control.



Data interface

The data interface is programmable via machine parameters (see index "Programming Modes", External data transfer).

The RS-232-C interface of the TNC must be set for external transfer or FE operation!

Program structure

Only linear programs can be executed with "Blockwise transfer".

- Program calls, subprogram calls, program section repeats and conditional program jumps cannot be executed.
- Unless the control operates with a central tool file, only the tool last defined can be called.

Block numbers (sequence numbers)

The program to be transferred can have block numbers (sequence numbers) exceeding 999.

The blocks do not have to be numbered sequentially; however, no block number may exceed the number 65534.

High sequence numbers are displayed on the screen with 2 lines.

Starting "blockwise transfer"

START

Data transfer from an external storage device can be started in the operating modes "Program run full sequence/single block" with the "EXT" key.

The control stores the transferred program blocks in available memory and interrupts data transfer when the storage capacity is full.

No program blocks are displayed until the available memory is full or the program is completely transferred.

The program run can be started with the machine "START" button even when no program block is displayed.

To avoid unnecessary interruptions of the program run, you should already have a number of stored program blocks as a buffer before starting. Therefore, it is advantageous to wait until the available memory is full.

After starting, the processed blocks are cleared and further blocks are continuously called from the external storage device.

Skipping over program blocks

GOTO □

If, in "Blockwise transfer" operation, you press the "GOTO □" key before starting and enter a sequence number, all blocks preceding this number will be ignored.

Notes

Programming Modes (P)

Programming Modes (P)

Programming Modes (P)

CYCL DEF CYCL CALL **Standard Cycles**

CYCL DEF **Coordinate Transformations**

CYCL DEF **Other Cycles**

CYCL DEF CYCL CALL

Q Q DEF **Parametric Programming**

• New function

Programming Modes (P)

● New function

Conversational Programming
General information

Introduction

The individual work steps on a conventional machine tool must be initiated by the operator. On an NC machine, the numerical control assumes computation of the tool path, coordination of the feed movements on the machine slides and generally also monitors the spindle speed. The control receives the information for this in form of a program in which the machining of the workpiece is described.

Program start and specification of blank

Define and call a tool, move to the tool change position.

Move to the workpiece contour, machine the workpiece contour, depart from the workpiece contour.

Traverse to the tool change position.

End of program

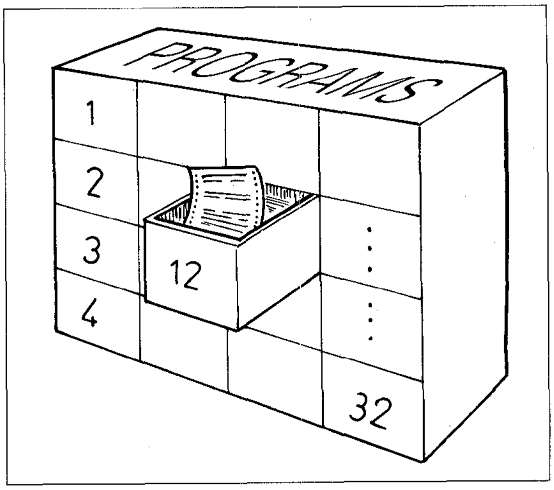
Program scheme

Programs

The control can store up to 32 programs (HEIDENHAIN or ISO) with a total of 4000 blocks (HEIDENHAIN dialog).

One part program can contain up to 4000 blocks.

Individual programs are identified by program numbers.



Switching between conversational and DIN/ISO programming

The control is switched to conversational or ISO programming via the MOD functions (see: index General Information, MOD functions).

A program consists of individual lines, called program blocks (symbol: ☐).

Blocks

Every block in a program corresponds to one work step, e.g.
L X+20 Y+30 Z+50 R0 F1000 M03.

```
7   L    Z–20   R0 FMAX         M03
8   L    X–12   Y+60 R0 FMAX
9   L    X+20   Y+60 RR F40
10  RND  R+5    F20
11  L    X+50   Y+20 RR F40
12  CC   X–10   Y+80
13  C    X+70   Y+51,715 DR+ RR
```

Block numbers (Sequence numbers)

The block number (also called the sequence number) identifies the program block in a part program. The control assigns a unique number to each block.

Words

Each block is composed of words, e.g. X+20;

Address Values

A word is composed of an address letter, e.g. X and a value, e.g. +20.

Abbreviations used above:

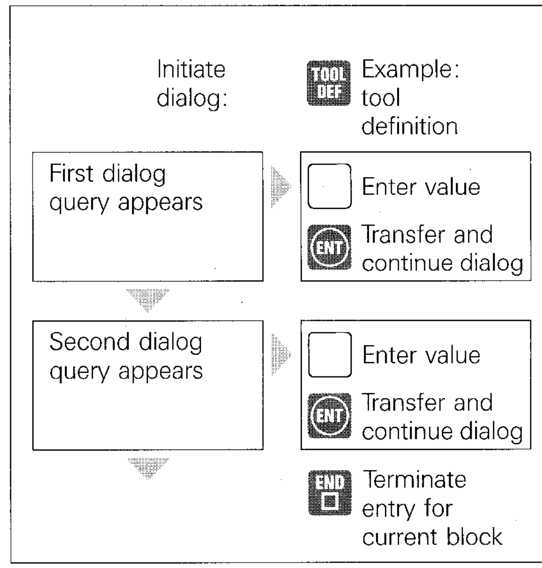
L = linear interpolation
X, Y, Z = coordinates
R0 = no tool radius compensation
F = feed rate
M = miscellaneous function

Conversational Programming
Responding to dialog queries

The dialog principle

Program input is dialog guided, i.e. the control requests the required data. The corresponding dialog sequence for each program block is started with a dialog initiation key, e.g. "TOOL DEF" (the control subsequently requests the tool number, then the tool length, etc.).

Errors are displayed in plain language during program input. False entries can be corrected immediately – while entering the program.



Responding to dialog queries/ Continuing the dialog

After you press a dialog initiation key, the control requests the necessary data. You must give a response to every dialog query. The response is shown in the highlighted field on the screen. After answering the dialog query, the entry is transferred to the memory with the "ENT" key. "ENT" is short for "enter" (i.e. confirm entry, transfer, store). The control then displays the next dialog query.

Skipping dialog queries

To make the entries in the preceding block modal, that means valid for the current block, (e.g. feed rate or spindle speed), do not respond to the associated dialog queries; skip them with the "NO ENT" key.

Entries already displayed in the highlighted field or already included in the program are deleted with "NO ENT"; the next dialog query appears on the screen.

During program run, the previously programmed values are valid for the associated address.

```
PROGRAMMING AND EDITING
FEED RATE ?            F= / F MAX = ENT
43   CYCL DEF 1.5 F50
44   L   X-70            Y-55
         Z-2           R0 F100      M99
45   L   X+70
                       R   F        M99
46   L   Y+55
                       R   F        M99
--------------------------------------
ACTL.  X +     49,258  Y +     23,254
       Z +     15,321  C +     84,000

                       F 0          M5/9
```

Directly terminating a block

If you have programmed all the desired information in a block, you can directly terminate the block with "END □".

The control saves the entered data, and no more queries for this block appear.
Data not programmed in this block remain effective as programmed in previous blocks.
Certain routines, such as "Read-in program", are also terminated with this key.

Entering numerical values

Numerical values are entered with the numeric keypad – with a decimal point or decimal comma (selectable via machine parameter) and sign key. You need not enter preceding or succeeding zeroes. You can enter the sign before, during or after the entering the number.

Conversational Programming
Editing functions

Editing

The term **editing** means entering, changing, supplementing and checking programs.

The editing functions are helpful in selecting and changing program blocks and words, and they become effective at the touch of a key.

Selecting a block

GOTO □

The current block stands between two horizontal lines.

A specific block is selected with "GOTO □".

Initiate the dialog — GOTO □

GOTO: NUMBER =	□ ENT	Key in and confirm the block number.

Paging through the program

↓ ↑

Vertical cursor keys:

Select the next lower or next higher block number.

Hold down a vertical cursor key to continuously run through the program lines.

Inserting a block

You can insert new blocks anywhere in existing programs. Just call the block which is to precede the new block. The block numbers of the subsequent blocks are automatically increased.

If the program storage capacity is exceeded, this is reported at dialog initiation with the error message: = PROGRAM MEMORY EXCEEDED =.

This error message also appears if program end (PGM END block) is selected. You should then select a lower block number.

Editing words

← →

Horizontal cursor keys:

The highlighted field is moved within the current block and can be placed on the program word to be changed.

One word in the current program block is to be changed:	→	Move the highlighted field to the word to be changed.

The dialog query appears for the highlighted word, e.g.		
COORDINATES ?	X	
	□	Change the value.

To change another word:	← →	Move the highlighted field to the word to be changed.
If all corrections have been made:	END □	Transfer the block (or move the highlighted field to the right or left off the screen).

Conversational Programming
Editing functions

Searching for certain addresses

You can use the vertical cursor keys to search for blocks containing a certain address in the program.

Use the horizontal cursor keys to place the highlighted field on a word having the search address, and then page in the program with the vertical cursor keys:
only those blocks having the desired address are displayed.

Example

All blocks with the address M are to be displayed:	▶ ↓ ↑	Select one block with the desired address.
	▶ →	Place the highlighted field on a word with the required address.
MISCELLANEOUS FUNCTION M ?	▶ ↓ ↑	Call blocks with the desired address.

Conversational Programming
Clearing/deleting functions

Clear program

CL PGM

The dialog for clearing a program is initiated with the **CL PGM** key.

Initiate the dialog CL PGM

ERASE = ENT/END = NOENT	
Program is to be cleared:	↓ ↑ or GOTO □ select a program number.
	ENT Erase the program.
Program is not to be cleared:	NO ENT

Delete block

DEL □

The current block (in a program) is deleted with **DEL □.**
The block to be deleted is selected with **GOTO** □ or a cursor key.
Program blocks can only be deleted in the PROGRAMMING AND EDITING operating mode.
After deletion, the block with the next lower sequence number appears in the current program line.
The following sequence numbers are corrected automatically.

The current program block is to be deleted:	DEL □ Delete block.

Delete program section

To delete program sections, call the last block of the program section.
Then continue pressing **DEL** □ until all blocks in the definition or program section are deleted.

Clear entry, error message

CE

You can clear numerical inputs with the "CE" key. A zero appears in the highlighted field after pressing the "CE" key.
Non-blinking **error messages can also be cleared with the "CE" key.**

NO ENT

An entered value and the address are completely cleared with "NO ENT".

Program Selection
Opening a program
Selecting an existing program

You open a program and select a stored program by first pressing the "PGM NR" key (program number).

A table with the HEIDENHAIN dialog programs and ISO programs stored in the TNC appears on the screen. The program number last selected is highlighted. The program length in characters is given after the program number. ISO programs are designated by "ISO" after the program number.

You can select the desired program either
- via the cursor keys

or
- by entering its number.

If the selected program number does not yet exist, a new program is opened.

```
PROGRAM SELECTION
PROGRAM NUMBER =
       1        360
   10001        756
   10002        1440
     111        1548
   11111  ISO   44
      13        450
      14        450
       2        900
--------------------------------
ACTL.  X +    49,258  Y +    23,254
       Z +    15,321  C +    84,000

                      F 0       M5/9
```

Opening a program

Depending on the selected program type, HEIDENHAIN dialog programs or ISO programs can be opened (see index General Information, MOD Functions).

Initiate the dialog

PROGRAM SELECTION PROGRAM NUMBER =	Enter the program number (maximum 8 characters). Confirm entry.
MM = ENT / INCH = NO ENT	for dimensions in mm, or
	for dimensions in inches.

Example display

0 BEGIN PGM 96231 MM
1 END PGM 96231 MM

Selecting an existing program

All existing programs (HEIDENHAIN format and ISO) can be edited, tested, displayed graphically and executed, regardless of the selected type of programming.

Initiate the dialog

PROGRAM SELECTION PROGRAM NUMBER =	Place the highlighted field on the desired program number.
or	Enter the program number.

Example display

0 BEGIN PGM 7645 MM
1 BLK FORM Z X+0
Y+0 Z–40
2 BLK FORM X+100
Y+100 Z+0

Program Selection
Program protection

Edit protection

After creating a program, you can designate it as erase- and edit-protected.

The program is then marked with a **P** ("protected") at the start and end of the program.

Protected programs can be executed and viewed, but not changed.

A protected program can only be erased or changed if the erase/edit protection is removed beforehand. This is done by selecting the program and entering the code number **86357.**

Activating edit protection

Initiate the dialog PGM NR

PROGRAM NUMBER =	Enter and transfer the number of the program to be protected. (ENT)
0 BEGIN PGM 22 MM	Press the key until the dialog query "PGM protection" appears. (→)
PGM PROTECTION ?	(ENT) Protect the program.
0 BEGIN PGM 22 MM P	(→) Erase/edit protection is programmed. **P** appears at the end of the line.

Removing edit protection

Initiate the dialog PGM NR

PROGRAM NUMBER =	(ENT) Enter the number of the program whose edit protection is to be removed.
0 BEGIN PGM 22 MM P	MOD Select the auxiliary operating mode.
VACANT MEMORY: 148330 BYTE	(↑) Select the MOD function "Code number".
CODE NUMBER =	Enter code number **86357**.
0 BEGIN PGM 22 MM	(ENT) Erase/edit protection is removed. The **P** is deleted.

Code number 86357

Program Selection
Blank form definition

Test graphics

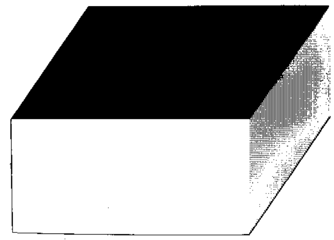
A blank form definition must be programmed before the machining program can be simulated graphically.

Blank

For the graphic displays, the blank dimensions of the workpiece (BLK FORM = BLANK FORM) must be entered at the start of program.

The blank form must always be programmed as a cuboid, aligned with the machine axes.

Maximum dimensions:
14000 mm x 14000 mm x 14000 mm.

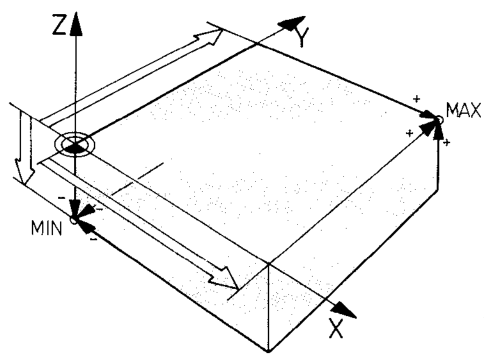


Minimum point
Maximum point

The cuboid is defined with the minimum point **(MIN)** and maximum point **(MAX)** (points with "minimum" and "maximum" coordinates).

MIN can only be entered in absolute dimensions; MAX may also be incremental.

The blank data are stored in the associated machining program and are available after program call.



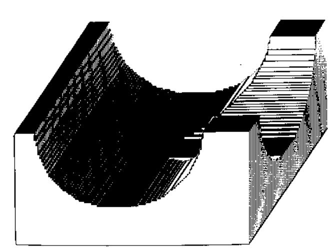
Graphic display

Machining can be simulated in the three main axes – with a fixed tool axis.

Tool form

Machining is correctly displayed with a cylindrical tool in the graphic view.

The graphic must be interpreted accordingly when using form tools.

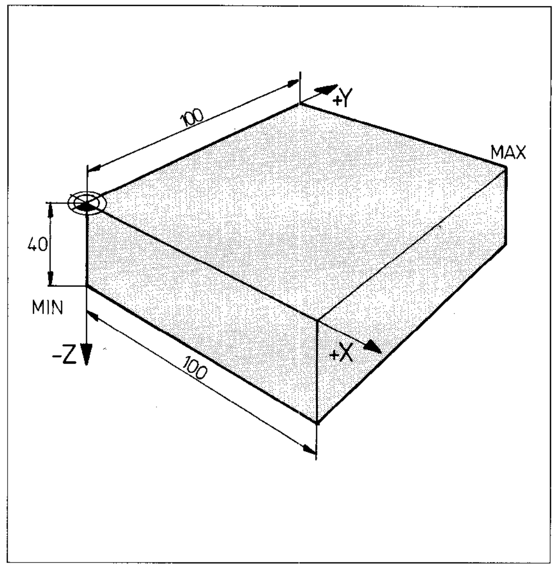


Program Selection
Blank form definition

Example

The **MIN** point has the coordinates X0, Y0 and Z–40.

The **MAX** point has the coordinates X100, Y100 and Z0.



To define a blank, a program must be selected in the "Programming and editing" operating mode.

Entering the cuboid corner points

Initiate the dialog BLK FORM

WORKING SPINDLE AXIS X/Y/Z ?	Z Enter the spindle axis, e.g. Z.

MIN

DEF BLK FORM: MIN-CORNER ?	0 ENT	X coordinate.
	0 ENT	Y coordinate.
	+/– 4 0 ENT	Z coordinate.

MAX

DEF BLK FORM: MAX-CORNER ?	1 0 0 ENT	X coordinate.
	1 0 0 ENT	Y coordinate.
	0 ENT	Z coordinate.

Example display

1 BLK FORM 0.1 Z X+0
Y+0 Z–40

2 BLK FORM 0.2 X+100
Y+100 Z+0

Error messages

BLK FORM DEFINITION INCORRECT
The MIN and MAX points are incorrectly defined, or the machining program contains more than one blank definition, or the side proportions differ too greatly.

PGM SECTION CANNOT BE SHOWN
Wrong spindle axis is programmed.

Tool Definition
Tool definition in part program

Tool definition

TOOL DEF

The control requires the tool length and tool radius to enable it to compute the tool path from the given work contour.

These data are programmed in the tool definition.

Whether the tools are defined decentralized in the appropriate part program or in a central tool file (program 0) is determined by a machine parameter.

```
PROGRAMMING AND EDITING
2    BLK FORM 0.2           X+100
          Y+100             Z+0
3    TOOL DEF 1             L+0
                            R+5
4    TOOL CALL 1               Z
               S 125
5    END PGM 55             MM
-----------------------------------
ACTL. X +    49,258  Y +    23,254
      Z +    15,321  C +    84,000

                     F 0      M5/9
```

Tool number

Compensation values always refer to a certain tool which is designated by a number.

Valid tool numbers:

with automatic tool change or in program 0:
1 to 99
without automatic tool change or in the machining program: 1 to 254.

Tool definition in the part program

If tools required in a program are defined in that program, a program printout will include the specifications of the tool dimensions.

Input

Initiate the dialog TOOL DEF

TOOL NUMBER ?	ENT Enter the tool number. The tool number 0 cannot be programmed under TOOL DEF. Tool 0 is internally defined with L = 0 and R = 0.
TOOL LENGTH L ?	+/- ENT Enter the tool length or the difference to the zero tool.
TOOL RADIUS R ?	ENT Enter the tool radius.

Central tool file

If the central tool file (program 0) is activated by machine parameters, the tools must always be defined there.

They then only have to be called in any program.

The central tool file is programmed, changed, output and read in the "Programming and editing" operating mode.

Every tool is entered with the tool number, length, radius and pocket number. Tool 0 must be defined with L = 0 and R = 0.

```
PROGRAMMING AND EDITING
T2          L+5,3          R+6
T3          L+12,45        R+7,75
T4          L+25,21        R+3,5
T5          L+52,52        R+2,5
T6          L+85           R+2
T7          L+32,71        R+8
T8          L+147,1        R+13
T9          L+0            R+15,5
--------------------------------
ACTL. X +    49,258   Y +    23,254
      Z +    15,321   C +    84,000

                      F 0      M5/9
```

Example

Tool 3 is to be defined with L = 5, R = 7

Initiate the dialog PGM NR

PGM NAME ?	0 ENT	Select the central tool file.
BEGIN TOOL MM	GOTO 3 ENT	Select the tool.
T3 L0 R0	→ 5	Enter the length.
	→ 7 END	Enter the radius.

Tool changer with flexible pocket coding

On machines with a tool magazine and flexible pocket coding, the tools can be returned to a different magazine pocket than they were taken from.

The control memorizes which tool number is stored in which pocket.

"TOOL DEF" functions like a tool pre-selection here, i.e. the tool search is programmed with "TOOL DEF". In this case, only the query for the tool number appears.

Oversize tools

Oversize tools occupying three pockets are to be designated as "special tools". A special tool is always returned to the same pocket.

Program by placing the highlighted field on the dialog query

SPECIAL TOOL ?

and respond with the "ENTER" key.

The preceeding and succeeding pocket numbers should be deleted by positioning the highlighted field and pressing the "NO ENT" key. A ✱ is displayed in place of the erased pocket number.

"S" for special tool and "P" for pocket number only appear if this function was selected via machine parameters.

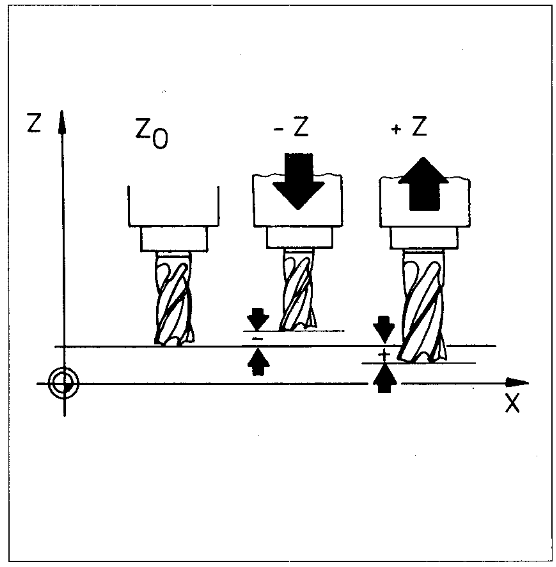
P0 (spindle) or another pocket must be free in the magazine.

Tool Definition

Tool length L

The tool length is compensated with a single adjustment of the spindle axis by the length compensation.

Compensation becomes effective after tool call and subsequent movement of the tool axis.



Zero tool

Compensation ends after a tool is called or with T_0 (T_0 is called the zero tool and has a length of 0).

The correct compensation value for the tool length can be determined on a tool pre-setter or on the machine.

If the compensation value is to be determined on the machine, then you must first enter the workpiece datum.

Length differences

When the compensation values are determined on the machine, the zero tool serves as a reference.

The length differences –Z or +Z of the other clamped tools to this zero tool are programmed as tool length compensations.

If a tool is **shorter** than the zero tool, the difference is entered as a **negative** tool length compensation. If a tool is **longer** than the zero tool, the difference is entered as a **positive** tool length compensation.

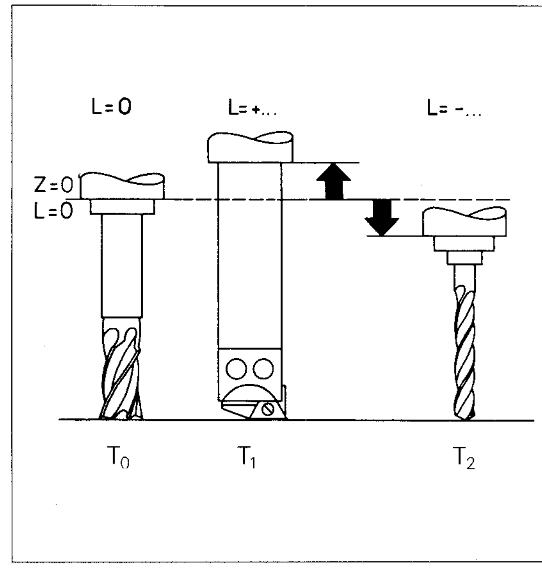
Preset tools

If a tool pre-setter is used, all tool lengths are already known. The effective compensation values correspond to the tool length and are entered with the correct signs according to a list.

Tool Definition
Transferring tool length

Tool lengths can be easily and quickly entered with the actual position transfer function.

1. Move the zero tool T_0 to the work surface and set the spindle axis to zero.
2. After exchanging, move the tools T_1 or T_2 to the work surface.
3. Transfer each display value in this position to the tool length definition. This gives you the length compensation to the zero tool.



Input

Operating mode — or

X Y Z Touch the surface with the zero tool.

Initiate the dialog — Z Spindle axis, e.g. Z.

DATUM SET	ENT Reset to zero.

X Y Z Also touch the surface with the new tools T_1 or T_2.

Operating mode

Either
1. call a tool definition in a program and initiate the dialog "TOOL LENGTH L ?",

or
2. select a tool in the central tool file and initiate the dialog "TOOL LENGTH L ?".

TOOL LENGTH L ?	Z Select the spindle axis to transfer the tool length. ENT Transfer the length compensation.

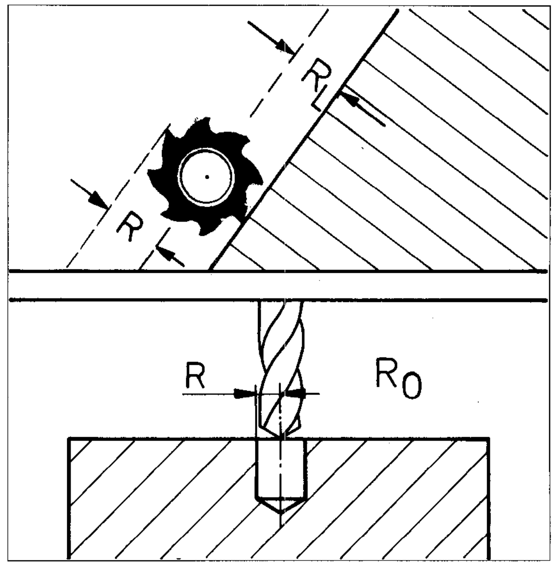
TOOL RADIUS R ?	ENT Enter the radius.

Tool Definition
Tool radius

Tool radius R

The tool radius is entered as a positive number (exception: radius compensation when programming the cutter center path).

A tool radius must always be programmed before a machining program can be checked with test graphics.

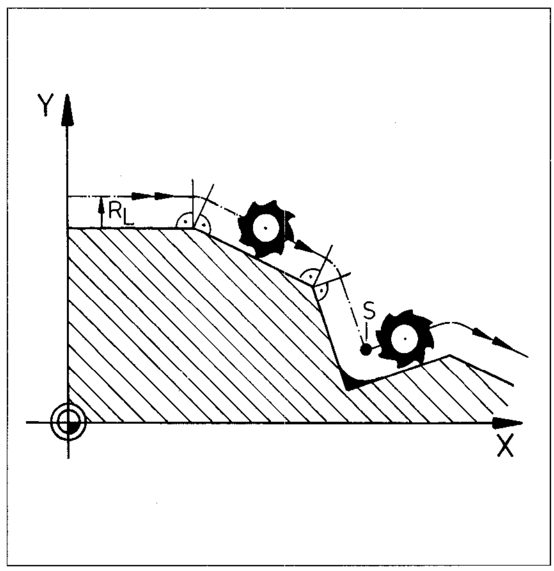


Tool radius compensation

Drilling work is programmed without radius compensation (R0), while milling jobs are usually programmed with radius compensation (R_L/R_R).

Compensation is effective after a tool call, programming with RL or RR in a positioning block (L, C etc.), or a movement in the active interpolation plane. Compensation ends with a positioning block which contains R0.

If the tool travels with path compensation, i.e. the tool center path is offset by the programmed tool radius, the tool follows a path parallel to the contour at the distance of the tool radius. The programmed feed rate applies to the center path.



Outside corners

The control inserts a transition curve for the center path of the tool at outside corners, so the tool rolls around the corner.

In most cases, the tool is thus guided at a constant path speed around the outside corner.

Automatic decleration at corners

If the programmed feed rate is too high for the transition curve, the path speed is reduced (which produces a more precise corner). The limit value is permanently programmed in the control (machine parameter).

Inside corners

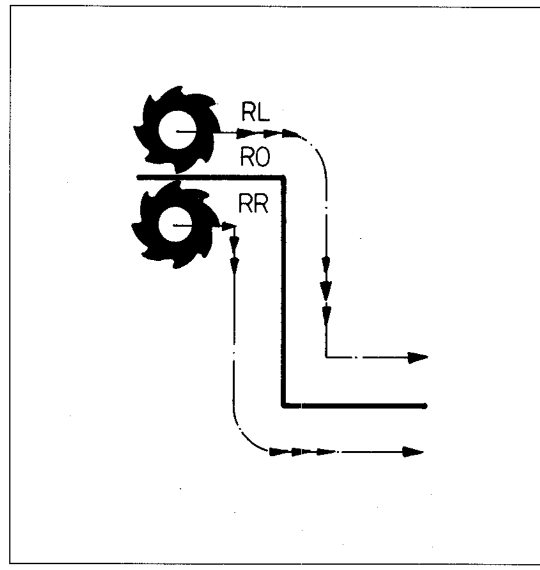
The control automatically determines the intersection S of the two cutter paths parallel to the contour (equidistant) at inside corners.

This prevents back-cutting in the contour; the work is not damaged. The control thus shortens traversing distances according to the tool radius in use.

The radius of the tool must always be chosen so that every contour element – even when shortened – can be machined.

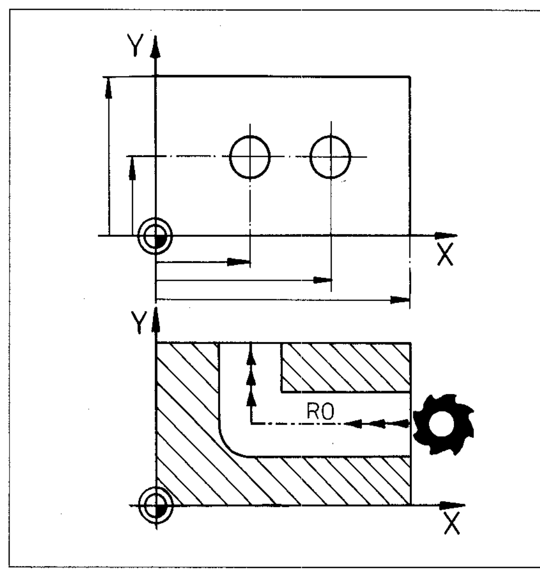
Cutter Path Compensation
Entering RL/RR

To automatically compensate for the tool radius – as entered in the TOOL DEF blocks – the control must be informed whether the tool travels to the left of, to the right of, or directly on the programmed contour.



R0

If the tool is to travel **on** the programmed contour, no radius compensation should be programmed in the positioning block.
At the dialog query
TOOL RADIUS COMP.: RL/RR/NO COMP. ?
press the "ENT" key.
Screen display: R0



Programming radius compensation

The radius compensation is entered in positioning blocks (L, C etc.) with the "RL" and "RR" keys at the dialog query
TOOL RADIUS COMP.: RL/RR/NO COMP. ?

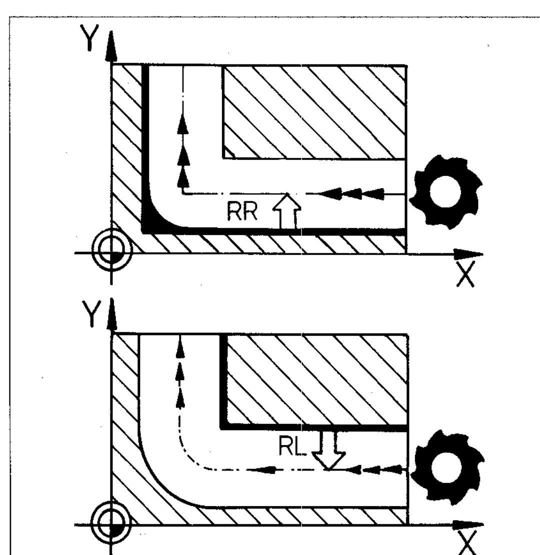
"Left" or "right" should be understood as looking in the direction of movement.

RR

If the tool is to travel at the distance of the radius to the **right** of the programmed contour, press the "RR" key.
Display: RR

RL

If the tool is to travel at the distance of the radius to the **left** of the programmed contour, press the "RL" key.
Display: RL



R

If the previous compensation should remain effective (modal):
press the "NO ENT" key.
Display: R

Cutter Path Compensation
Working with radius compensation

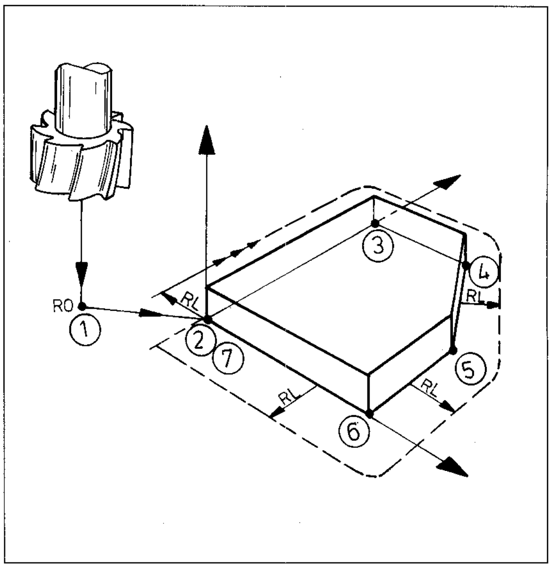
Starting point R0

Change the tool and call the compensation values with "TOOL CALL".

Traverse rapidly to the starting point ①.

At the same time lower Z to the working depth (if danger of collision, first traverse in X/Y, then separately in Z!). This compensates for the tool length.

The radius compensation still remains switched off with "R0".



1st contour point RL/RR

Traverse to contour point ② with radius compensation RL/RR at reduced feed rate.

Machining around the contour

Program the following contour points to ⑦ at milling feed rate.

Since the RL/RR asignment remains constant, the associated dialog queries can be skipped with "NO ENT" or "END □".

Last contour point RL/RR

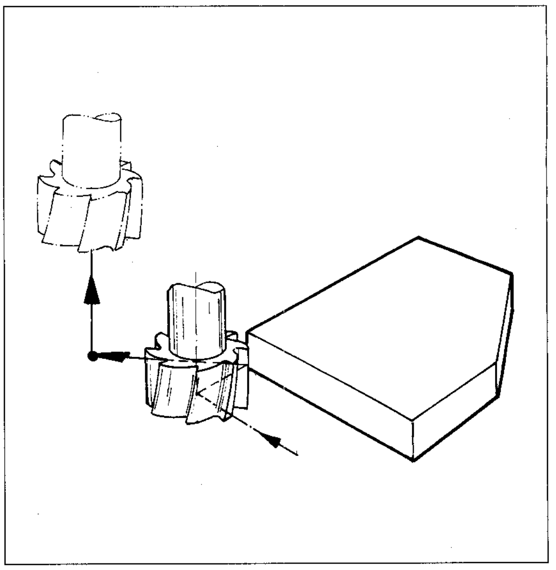
After a complete circulation, the last contour point ⑦ is identical to the first contor point ② and is still radius compensated.

End point R0

The end point (outside the contour) must be programmed without compensation R0 for complete machining.

To prevent collisions, only retract in the machining plane to cancel the radius compensation.

Then back-off the tool axis separately.

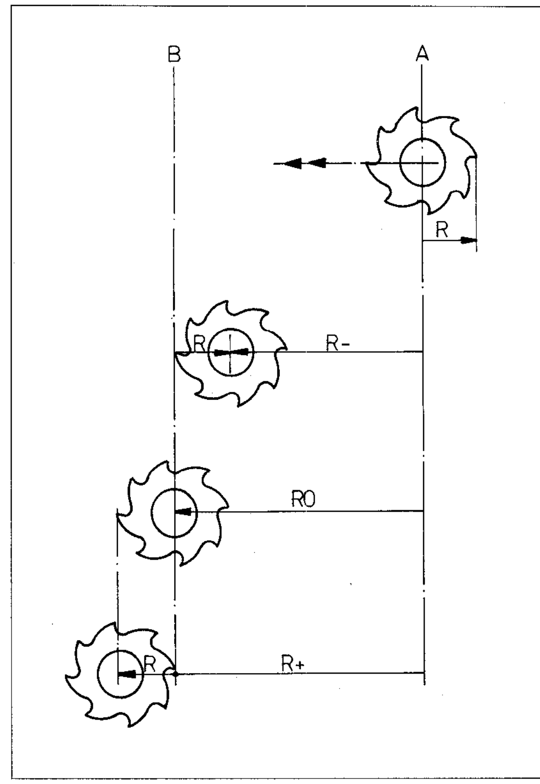


Cutter Path Compensation
Radius compensation R+, R–

Initiating the dialog

X Y ...

By pressing "R+" or "R–", you can lengthen or shorten a single-axis displacement by the tool radius.



This simplifies:
- positioning with manual data input,
- single-axis machining
- prepositioning for the "Slot" cycle.

The input dialog may be initiated – as on the point-to-point and straight cut controls TNC 131/ TNC 135 – directly via the corresponding yellow axis key.

Effect

This radius compensation has the following effect:

- The displacement is **shortened** by the tool radius: display **R–**.
- The tool traverses **to** the programmed nominal position: display **R0**.
- The displacement is **lengthened** by the tool radius: display **R+**.

R+/R– do not affect the spindle axis.

Example

The tool is to traverse from initial position X = 0 to X = (46 + tool radius).
Application example:
prepositioning for the "Slot" cycle.

Initiate the dialog X

POSITION VALUE ?	4 6 ENT
TOOL RADIUS COMP.: R+/R–/NO COMP. ?	R+ END Display: X+46 R+

Mixing

L and X

Uncompensated blocks (e.g. L X+20 R0) and single-axis blocks (e.g. X+20 R0 or X+20 R+) can be mixed in a part program.
Single-axis compensated positioning blocks (R+/R–) and radius compensated positioning blocks (RR/RL) are not to be entered in succession!

Correct:

```
L X+15 Y+20 R0
  Y+50      R0
  X+40      R+
  Y+70      R0
```

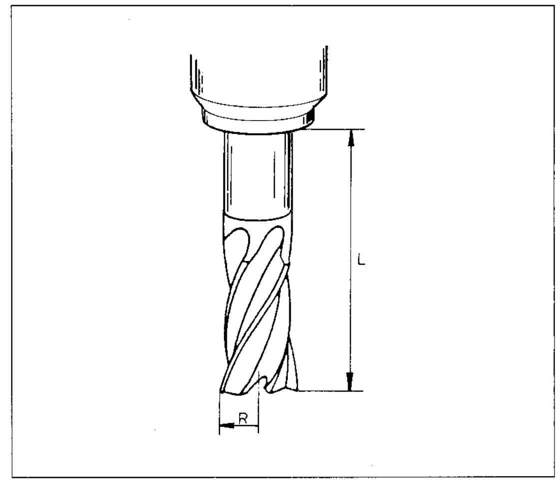
Incorrect:

```
L X+15 Y+20 RR
  Y+50      R+
L X+50 Y+57 RR
```

Tools
Tool call

Tool call

With TOOL CALL a new tool and the associated compensation values for length and radius are called up.



Spindle axis

In addition to the tool number, the control also needs to know the spindle axis to carry out length compensation in the correct axis or radius compensation in the correct plane.

Compensation effect

The spindle axis also defines the plane (e.g. XY) for circular movements: It is identical to the "radius compensation" plane.
This is also the plane for "coordinate rotation" and "mirror image".

Spindle axis	Length compensation	Radius compensation
Z	**Z**	**XY**
Y	Y	ZX
X	X	YZ

Spindle speed

The spindle speed is entered directly after the spindle axis.
Input range of the control: 0 to 99999 rpm.
If the speed exceeds the valid range for the machine, the following error message appears at program run

WRONG RPM

Activating compensation

A tool call activates length compensation.
It first becomes effective when the next tool axis movement is programmed.
It can be seen as a single infeed height movement.
Radius compensation first becomes effective when the compensation direction "RL" or "RR" is programmed in a positioning block.

Ending compensation

A "TOOL CALL" block ends the "old" tool length and tool radius compensation and calls the new compensation values.
Example: TOOL CALL 12 Z S 300

Tool radius compensation is also ended by programming "R0" in the positioning block.

If only the spindle speed is entered with "TOOL CALL", the compensations remain valid.
Example: TOOL CALL S 300

Tool call

Initiate the dialog

TOOL NUMBER ?	ENT Enter the tool number.

Spindle axis

WORKING SPINDLE AXIS X/Y/Z ?	Z Enter the spindle axis, e.g. Z.

Spindle speed

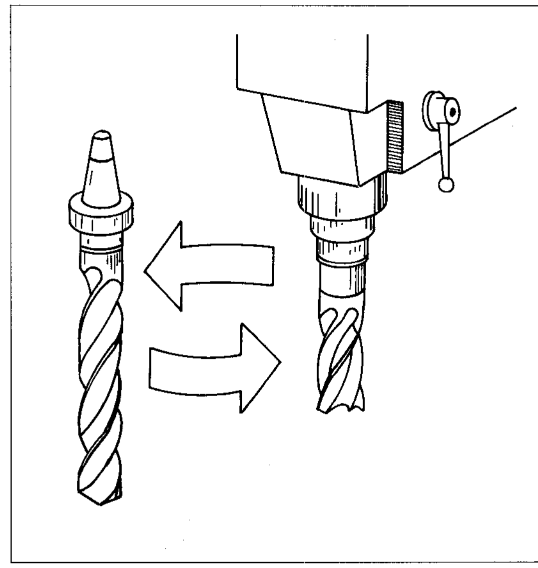
SPINDLE SPEED S RPM ?	ENT Enter the spindle speed (rpm).

Tools
Tool change

Tool change position

To change the tool, the main spindle must be stopped and the tool retracted in the spindle axis.

We recommend the insertion of an additional block in which the axes of the machining plane are likewise backed-off.



Workpiece-related change position

The tool moves to a **workpiece-related** position if no additional measures are taken.
Example: L Z+100 FMAX M06
The tool is driven 100 mm over the work surface if the tool length is 0 or TOOL CALL 0 was programmed.

T0 reduces the distance to the workpiece (danger of collision!) if a positive length compensation was effective prior to TOOL CALL 0.

Machine-related change position

You can use M91, M92 or a PLC positioning to traverse to a **machine-related** tool change position.
Example: L Z+100 FMAX M92
(see Machine-related coordinates M91/M92).

Manual tool change

The program must be stopped for a manual tool change. Therefore, enter a program STOP before the TOOL CALL. M6 has this stop effect when the control is set accordingly via machine parameters. The program is then stopped until the external START button is pressed.

The program STOP can only be omitted when a tool call is programmed solely to change the spindle speed.

Automatic tool change

The tool is changed at a defined change position. The control must therefore move the tool to a machine-specific change position. The program run is not interrupted.

```
1 BLK FORM 0.1 Z X+0 Y+0 Z–40
2 BLK FORM 0.2 X+100 Y+100 Z+0
3 TOOL DEF 1 L+0 R+5
4 TOOL DEF 2 L–2,4 R+3
5 TOOL CALL 0 Z
6 L Z+200 R0 FMAX M06
7 TOOL CALL 1 Z S 1000
8 L X+25 Y+30 FMAX
9 L Z+2 FMAX M3
.
.
.
```

Feed Rate F/Spindle Speed S/ Miscellaneous Functions M

Feed rate
F

The feed rate F, i.e. the traversing speed of the tool in its path, is programmed in positioning blocks in mm/min or 0.1 inch/min. The current feed rate is shown in the status display on the lower right of the screen.

Feed rate override

The feed rate can be varied within a range of 0% to 150% with the feed rate override on the control operating panel. The effective range of the potentiometer for tapping is limited by machine parameters!

Rapid traverse

The maximum input value (rapid traverse) on the control for positioning is:
- 29 998 mm/min or
- 11 800/10 inch/min.

The maximum operating speeds are set for each axis.
FMAX or the max. input is programmed for rapid traverse.
The control automatically limits rapid traverse to the permissible values.

FMAX is only effective **blockwise.**

If the F display is highlighted and the axes do not move, this means the feed rate was not enabled at the control interface. In this case, you must contact your machine manufacturer.

Spindle speed
S

The spindle speeds are set with "TOOL CALL".

Spindle override

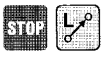
On machines with continuous spindle drive, the speed can be varied from 0% to 150% using the spindle override.

Miscellaneous functions
M

Miscellaneous functions can be programed to regulate certain machine functions (e.g. spindle "on"), to control program run and to influence tool movements. The miscellaneous functions are comprised of the address M and a code number according to ISO 6983. All of the M functions from M00 to M99 can be used.

Certain M functions become effective at the start of block (e.g. M03: spindle "on" clockwise), i.e. before movement, and others become effective at the end of block (e.g. M05: spindle "stop"). A list of all M functions with their effects as determined by the control can be found inside the back cover.

Only a certain number of these M functions are effective on a given machine.
Some machines may employ additional, non-standard M functions not defined by the control.
M functions are normally programmed in positioning blocks (L, C etc.).
However, M functions can also be programmed without positioning:



- Via the "STOP" key or
- by initiating the dialog with the "L" key and skipping the queries with "NO ENT" up to address M.

Programmable Stop/Dwell Time

Stopping program run

STOP

Program run can be stopped by one of the following functions.
Restart by pressing the external START button.

Initiate the dialog STOP

MISCELLANEOUS FUNCTIONS M ?	
Miscellaneous function is desired:	ENT Enter the miscellaneous function.
No miscellaneous function desired:	NO ENT No entry.

Example

18 STOP	M	Program run is stopped at block 18. No miscellaneous function.

M02/M30
- Program stop and (according to ISO) also spindle stop and coolant off.
  Return to block 1 of the program.

M00
- Program stop and (according to ISO) also spindle stop and coolant off.

M06
- Program stop and (according to ISO) also spindle stop, coolant off and tool change.

Program stops only when set accordingly by machine parameter!

Dwell time

Cycle 9 "Dwell time" can be used during program run to delay execution of the next block for the programmed time period (see "Other cycles").

Note:
The program continues running after the dwell time runs out!

Path Movements
Entry

The control/operator dialog for entering positioning blocks is illustrated below using the example of a straight line movement.

Operating mode	Programs can only be input in "PROGRAMMING AND EDITING".
Initiate the dialog	Select the type of movement, e.g. straight line.

Example

	Enter the end point of movement:
COORDINATES ?	X Select the axis, e.g. X.
	I Incremental – absolute ?
	Enter numbers with sign.
	Y Enter further coordinates.
	⋮
	ENT If all endpoint coordinates are entered, confirm the entry.
TOOL RADIUS COMP.: RL/RR/NO COMP. ?	RL RR Enter the radius compensation or
	ENT no radius compensation (R0).
FEED RATE ? F =	ENT Enter the feed rate or press only
	ENT for FMAX = rapid traverse.
MISCELLANEOUS FUNCTION M ?	ENT Enter a miscellaneous function if desired.

Abbreviated input

Subsequent blocks can be ended immediately with "END □", e.g. after entering the corner point coordinates.

In these cases, the last entries remain valid for non-programmed addresses.

Addresses may be skipped with "NO ENT".

Path Movements
Initiating the dialog

Contour elements

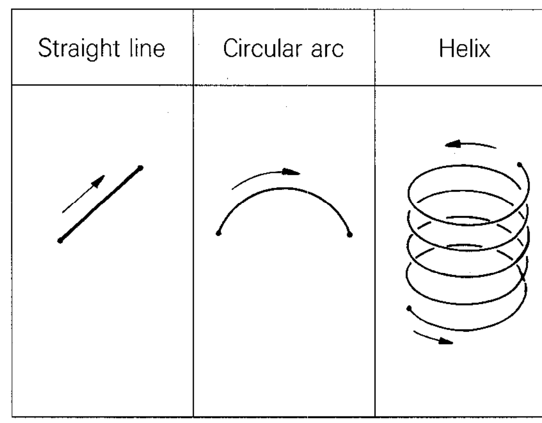
The shape of the workpiece is programmed without considering the tool. You program as though the tool were constantly being moved, regardless of the machine design. The programmable contours are composed of the contour elements **straight line** and **circle.** Using tool radius compensation, the control computes the tool-dependent path for the cutter center along which the tool is guided.

Generating the workpiece contour

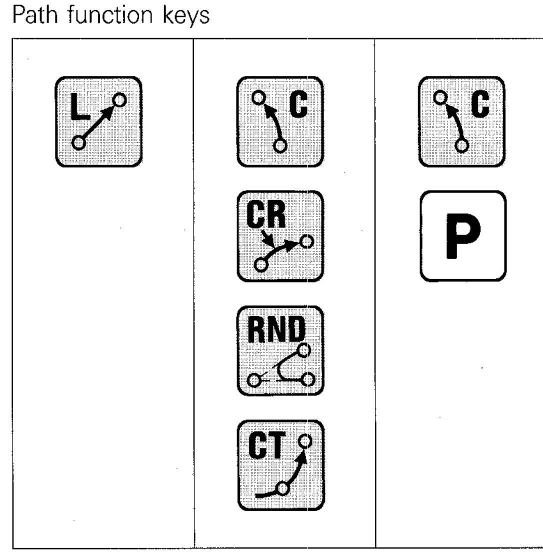
To be able to generate the workpiece contour, the control must be given the individual contour elements. Since each program block specifies the next step, the following information is required:

- straight line or circle
- the coordinates of each endpoint or other geometrical data such as the circle center and contour radius.

Contour elements



Path function keys



Initiating the dialog

To program a contour element, always begin with one of the gray path function keys. The type of movement is then defined for the contour element in question.

Coordinates

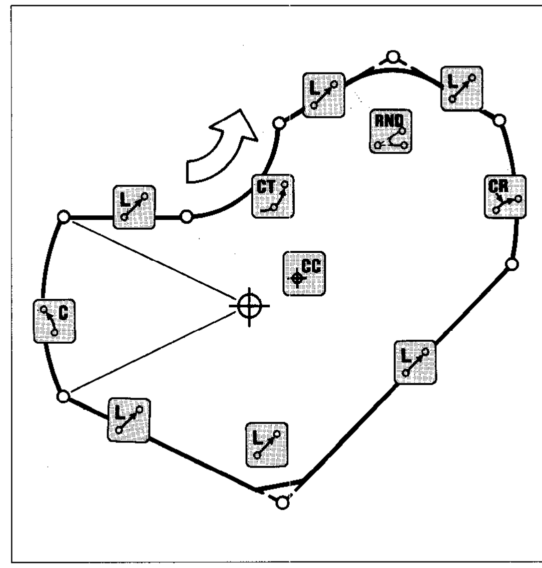
Point coordinates can only be input after selecting the path function.

Incremental/ Absolute

I

To enter the point coordinates incrementally, press the key for incremental inputs.

Path Movements
Overview of path functions



Straight lines

Straight line (L):
The tool moves in a straight line.
The endpoint of the straight line must be programmed.

Chamfer
A chamfer is inserted **between two straight lines.**

Circles

Circle center (CC) –
also the pole for polar coordinates:
Used to program the circle center for a circular arc with the "C" key, or to program the pole for polar coordinates.
CC generates no movement!

Circular movement (C):
The tool is moved in a circular arc. Program the endpoint of the arc. The circle center must be specified beforehand.

Corner rounding (RND):
An arc with tangential connections is inserted between two contour elements.
Program the arc radius and (in other blocks) the contour elements of the corner to be rounded.

Circular arc (CT) = "circle tangential":
A circular arc is tangentially connected to the preceding contour element.
Only the endpoint of the arc is programmed.

Circular arc (CR) = „circle per radius":
The tool is moved on a circular path.
Program the circle radius and the endpoint of the arc (but not the circle center).

Multi-axis movements

A maximum of three axes can be programmed for straight lines and a maximum of two axes for circles.

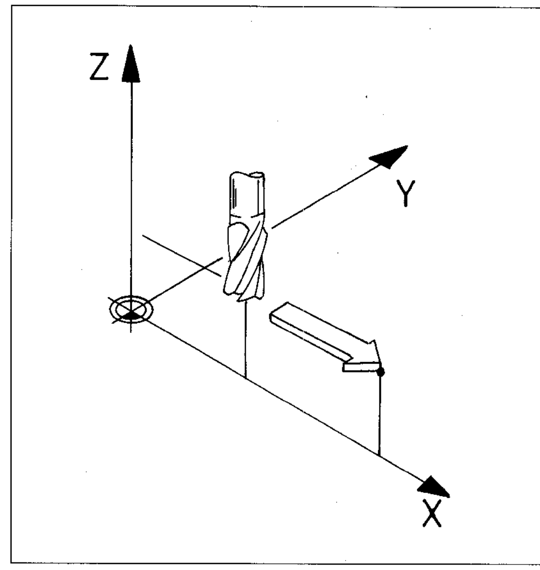
Graphics

The examples on the following pages must be supplemented with a uniform BLK FORM if a graphic display is wanted:

```
BLK FORM 0.1  Z  X+0    Y+0    Z-40
BLK FORM 0.2     X+100  Y+100  Z+0
```

Path Movements
1D/2D/3D movements

Movements are referred to – depending on the number of simultaneously traversed axes – as 1D, 2D or 3D movements.



Single-axis traverse: 1D movements

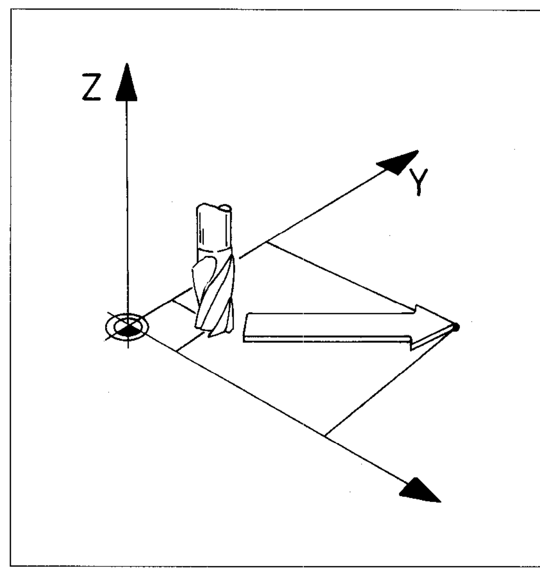
If the tool is moved relative to the work on a straight line along the direction of a machine axis, this is called single-axis positioning or machining.

Single-axis movements can also be programmed without using the grey path function keys. Only the radius compensation R+/R– is then available (see Radius compensation R+/R–).

2D movements

Movement in a main plane (XY, YZ, ZX) is called 2D movement.

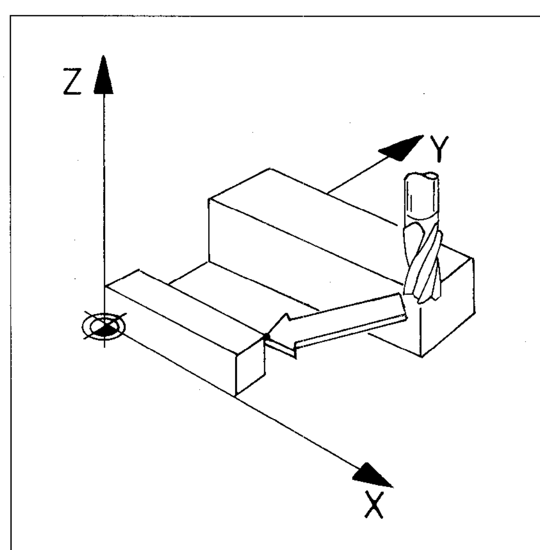
Straight lines and circles can be generated in the main planes with 2D movements.



3D movements

If the tool is moved relative to the workpiece on a straight line with simultaneous movement of all three machine axes, it is called a 3D straight line.

3D movements are required to generate oblique planes and bodies.



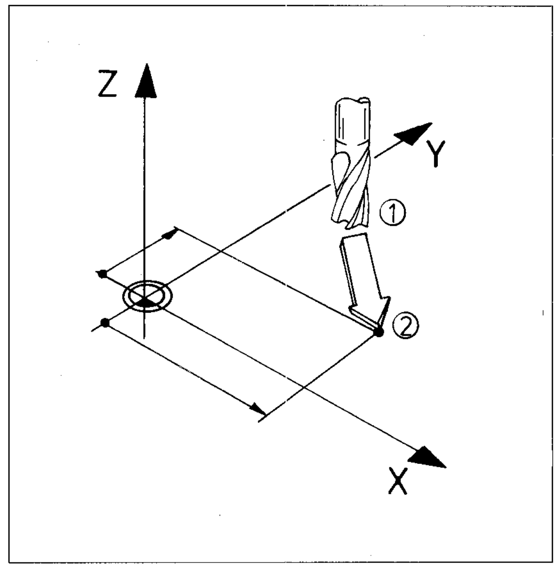
Linear Movement/Cartesian
Positioning in rapid traverse

Positioning

The tool is at the starting point ① and must travel on a straight line to target point ②.
You always program the target point ② (nominal position) of straight lines.

Position ② can be entered in Cartesian or polar coordinates.

The first position in a program must always be entered as an absolute value. The following positions can also be incremental values.



Example tool definition/ call

TOOL DEF 1 L+10	ENT R5 ENT	Tool 1 has length 10 mm and radius 5 mm.
TOOL CALL 1	ENT Z	Tool 1 is called in the spindle axis Z.
S200	END □	Spindle speed is 200 rpm.

Positioning block: complete input (main block)

L X+50 Y+30 Z0 ENT — Z is traversed with length compensation. Only press "ENT" after all simultaneously traversing axes are entered!

R0 ENT — "R0" is only programmed via "ENT"!

FMAX ENT M3 ENT — Rapid traverse movement "FMAX", spindle clockwise.

L X+50 Y+30 Z+0 R0 FMAX M3

Re-entry at tool calls is especially easy if you enter a main block (= complete positioning block) after a tool call.

Rapid traverse can also be entered as a machine-specific value (e.g. F 6000), if known.

Abbreviated input

L X+50 Y+30 END □ — Positioning in the XY plane without radius compensation. The tool center is driven to the programmed position (if R0 was programmed in preceding blocks).

After entering the desired values, program blocks may be shortened with the "END □" key if remaining data is unchanged.

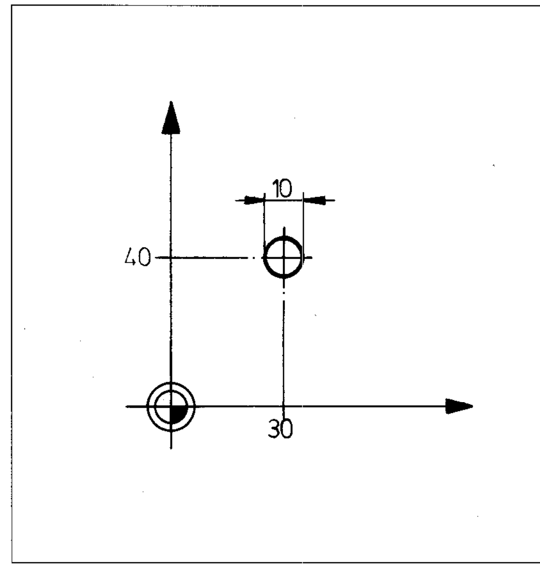
Linear Movement/Cartesian
Drilling

Absolute Cartesian coordinates

L X 30 Y 40 Z 2 END

L X+30 Y+40 Z+2

Multidimensional contour elements can only be entered after initiating with a gray path function key!



Incremental Cartesian dimensions

L I X 20 END

L IX+20

Only incremental entry.

Mixed entries

L I X 20 Y 30 END

L IX+20 Y+30

The position for X is entered in incremental dimensions, for Y in absolute dimensions.

Example drilling

A bore is programmed below for illustration purposes without cycles.

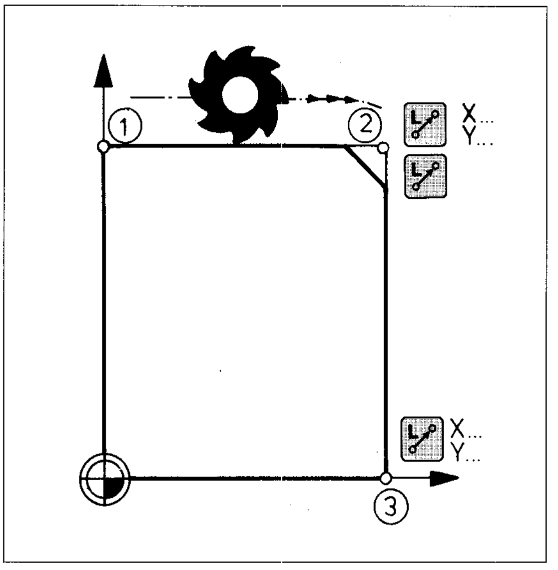
Program

0 BEGIN PGM 5231 MM	
1 BLK FORM 0.1 Z X+0 Y+0 Z–20	Blank form definition (only if graphic workpiece simulation desired)
2 BLK FORM 0.2 X+100 Y+100 Z+0	
3 TOOL DEF 1 L+0 R5	Tool definition
4 TOOL CALL 1 Z S1250	Tool call
5 L Z+200 R0 FMAX M6	Retract in Z, tool change
6 L X+20 Y+30 R0 FMAX M3	Positioning to 1st hole in X/Y, rapid traverse, switch on spindle
7 L Z+2 FMAX	Pilot positioning in Z
8 L Z–10 F80	Drilling at programmed feed rate
9 L Z+2 F1000	Retract in Z
10 L X+50 Y+70 R0 FMAX	Positioning to 2nd hole in X/Y
11 L Z–10 F80	Drilling at programmed feed rate
12 L Z+2 F1000	Retract in Z
13 L X+75 Y+30 R0 FMAX	Positioning to 3rd hole in X/Y
14 L Z–10 F80	Drilling at programmed feed rate
15 L Z+200 FMAX M2	Retract in Z
16 END PGM 5231 MM	End of program

Linear Movement/Cartesian
Chamfer

Chamfer

A chamfer can be programmed for contour corners formed by the intersection of two straight lines. The angle between the two straight lines can be arbitrary.



Prerequisites

A chamfer is completely defined by the points ① ② ③ and the chamfer block. A positioning block containing both coordinates of the machining plane should be programmed before and after a chamfer block. The compensation RL/RR/R0 must be identical before and after the chamfer block. A contour cannot be started with a chamfer.

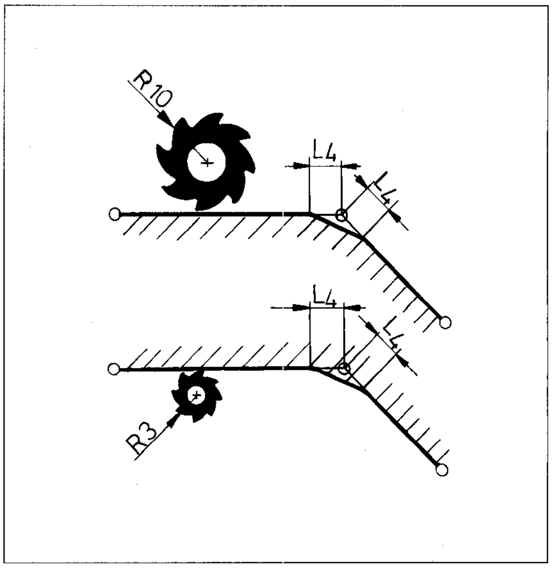
A chamfer can only be executed in the machining plane. The machining plane in the positioning block before and after the chamfer block must therefore be the same.

The chamfer length must not be too long or too short at inside corners: the chamfer must "fit between the contour elements" and also be machineable with the chosen tool.

The previously programmed feed rate remains effective for the chamfer.

Programming

Program a chamfer as a separate block.
Only enter the chamfer length – **no coordinates.**
The "corner point" itself is not traversed!



L = chamfer length

Entering the chamfer

[L] [4] [ENT]

Program block

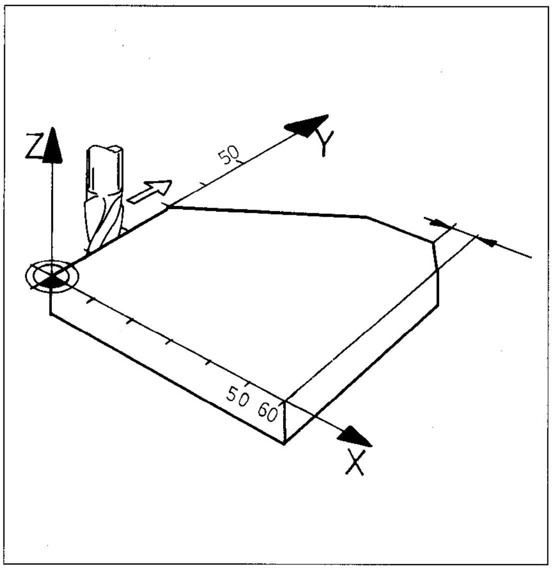
L 4

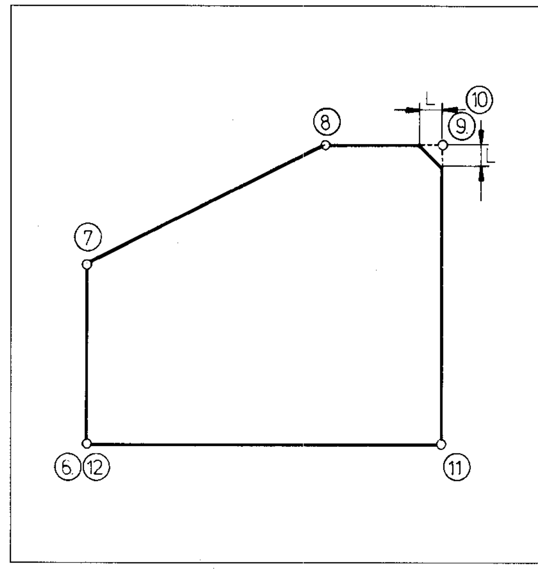
Example

TOOL DEF 1 L+0 R10	
TOOL CALL 1 Z S200	
L X+0 Y+50 RL F300 M3	Position ① (cf. figure above)
L X+50 Y+50	Position ②
L 4	Chamfer
L X+50 Y+0	Position ③

Linear Movement/Cartesian
Example

Example: milling straight lines





The block numbers are shown in the figure to aid you in following the sequence.

Program

1	TOOL DEF 1 L+0 R5	Tool definition
2	TOOL CALL 1 Z S500	Tool call
3	L Z+200 R0 FMAX M6	Tool change
4	L X–10 Y–20 **R0** FMAX M3	Pilot position (tool is up)
5	L Z–20 R F80	Plunge at downfeed rate
6	L X+0 Y+0 **RL** F200	Approach the contour, call radius compensation
7	L X+0 Y+30 RL F400	Machine the contour
8	L X+30 Y+50 RL	.
9	L X+60 Y+50 RL	.
10	L 2	Chamfer block
11	L X+60 Y+0 RL	
12	L X+0 Y+0 **RL**	Last block with radius compensation
13	L X–20 Y–10 **R0**	Cancel radius compensation
14	L Z+200 R FMAX M2	Back-off Z

Linear axes U, V, W

Linear interpolation can be performed simultaneously with a maximum of 3 axes – even when using additional axes.

For linear interpolation with an additional linear axis, this axis must be programmed with the corresponding coordinate **in every NC block.** This requirement holds even when the coordinate remains unchanged from one block to the next. If the additional axis is not specified, the control traverses the main axes of the machining plane again.

Example: linear interpolation with X and IV, tool axis Z.

```
11  L  X+0    IV+0   RR F100
12  L  X+100  IV+0
13  L  X+150  IV+70
```

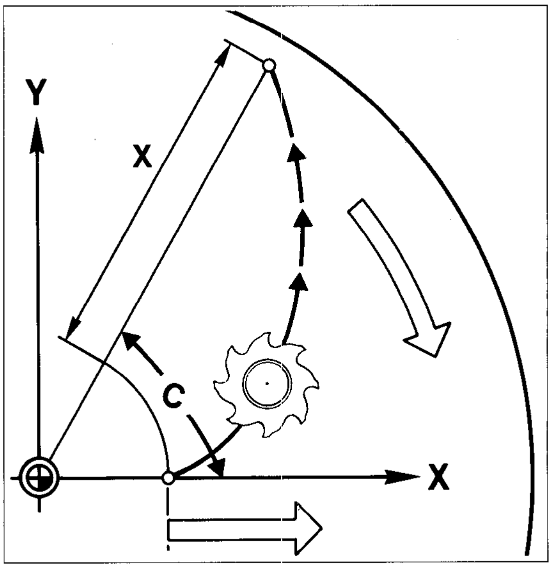
Rotary axes A, B, C

If the additional axis is a rotary axis (A, B or C axis), the control registers the entered value in angular degrees.

During linear interpolation with one linear and one rotary axis, the TNC interprets the programmed feed rate as the path feed rate. That is, the feed rate is based on the relative speed between the workpiece and the tool. Thus, for every point on the path, the control computes a feed rate for the linear axis F_L and a feed rate for the angular axis F_W:

$$F_L = \frac{F \cdot \Delta L}{\sqrt{(\Delta L)^2 + (\Delta W)^2}}$$

$$F_W = \frac{F \cdot \Delta W}{\sqrt{(\Delta L)^2 + (\Delta W)^2}}$$



where:
F = programmed feed rate
F_L = linear component of the feed rate (axis slides)
F_W = angular component of the feed rate (rotary table)
ΔL = linear axis displacement
ΔW = angular axis displacement

M94 for rotary axes

The position display for rotary axes can be set via machine parameters for either:

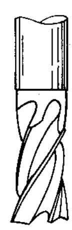
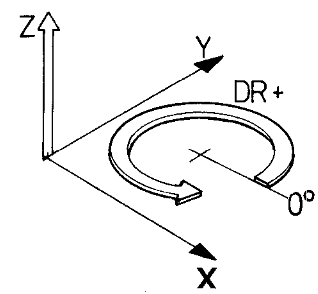
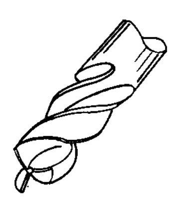
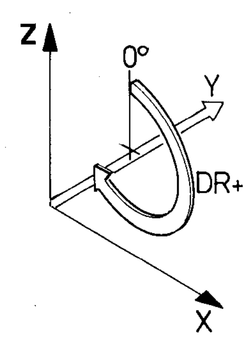
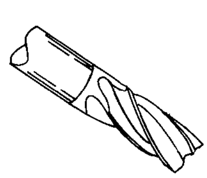
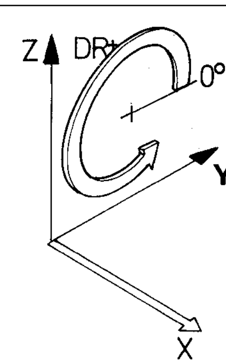
- ± 360° or
- ± ∞ (i.e. ± max. display value).

If ± ∞ is chosen as the measuring range, the position display for rotary axes can be reduced to values below 360° with M94.

Main planes Circular arcs can be directly programmed in the main planes XY, YZ, ZX.

TOOL CALL The circular interpolation plane is selected by defining the spindle axis in the "TOOL CALL" block. This also allocates the tool compensations.
The axis printed bold below (e.g. **X**) is identical in its positive direction with the angle 0° (leading axis).

Interpolation planes

	Spindle axis parallel to	Circular interpolation plane
Standard for milling machines	Z 	**X**Y 
Standard for horizontal borers	Y 	**Z**X 
	X 	**Y**Z 

Oblique circles in space Circular arcs which are not parallel to a main plane can be programmed via Q parameters and executed as a sequence of multiple short straight lines (L blocks).

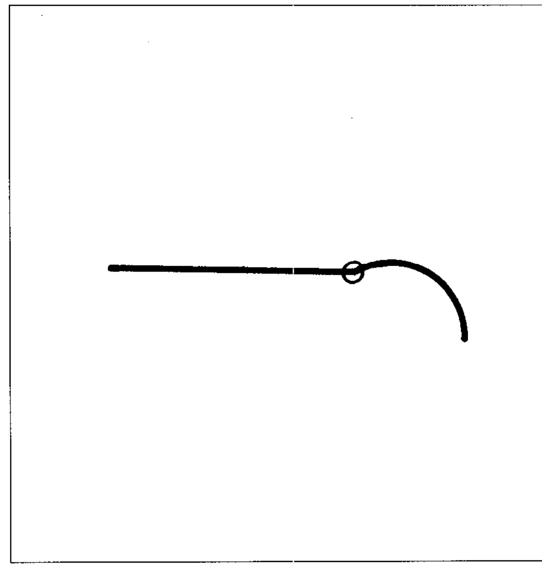
Circular Movement/Cartesian
Selection guide: Arbitrary transitions

Circular movement

The control moves two axes simultaneously, so the tool describes a circular arc relative to the workpiece.

Arbitrary transitions

The functions C and CR define – together with the preceding block – arbitrary transitions (i.e. tangential and non-tangential transitions) at the beginning and end of the arc.



Prerequisite

The starting point ① of the circular movement must be approached in the immediately preceding block.

Circle endpoint

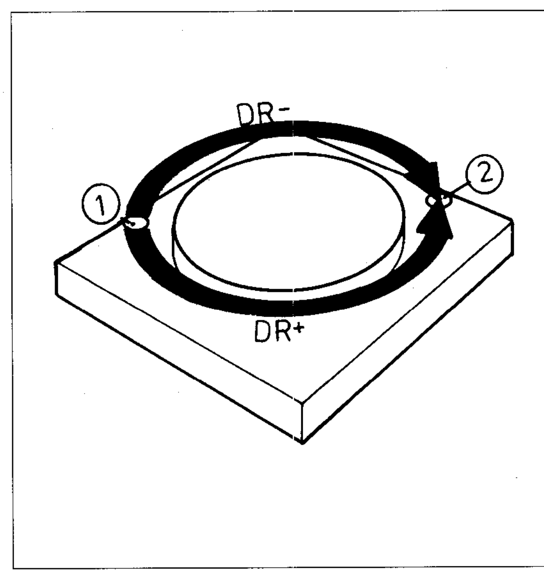
The circle endpoint ② is programmed in a C or CR block.

Rotating direction DR+/DR–

Both definitions also contain the direction of rotation.

Positive rotating direction (in mathematical terms) is **counterclockwise.**

Negative rotating direction is **clockwise.**



Radius

The radius is indirectly given for "C" as the distance from the position programmed in the immediately preceding C block (start of arc) to the circle center CC.

Full circles

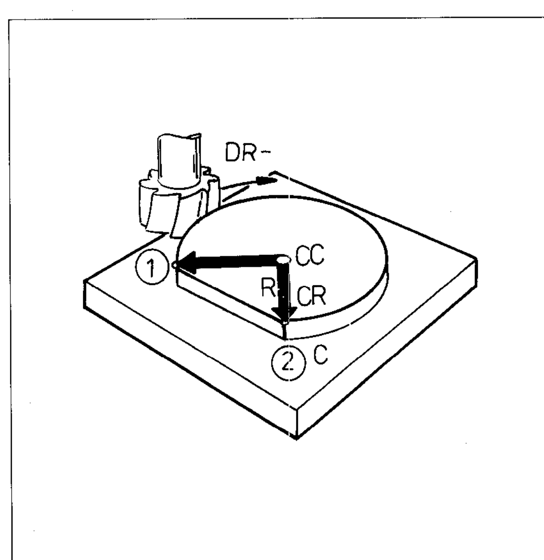
Full circles can only be programmed in one block, with "C".

CR

With CR the radius can be entered directly (CC not required).

Selecting

Given	Select
Starting point of arc ①	e.g. Approach starting point
Circle center	
End point of arc ②	
Starting point of arc ①	e.g. Approach starting point
Radius + end point of arc ②	



Circular Movement/Cartesian
Selection guide: Tangential transitions

Tangential transitions

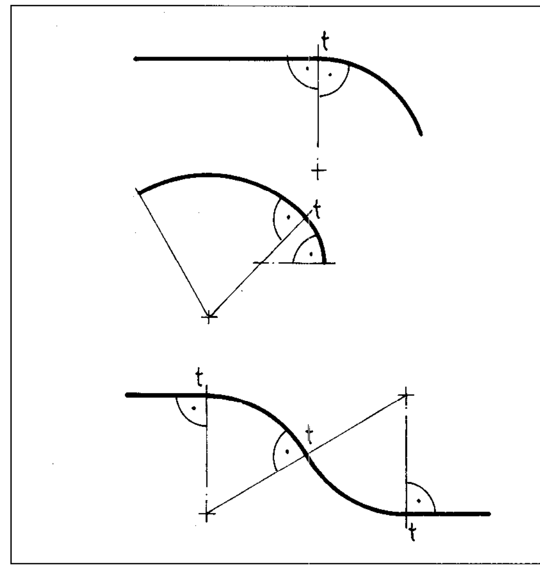
The "RND" and "CT" functions automatically produce a tangential (soft) entry into the arc. Departure from the arc is also tangential with "RND", and arbitrary with "CT". The direction of movement when entering the circle thus also determines the shape of the arc.

Direction of rotation

The direction of rotation need therefore not be given.

Center

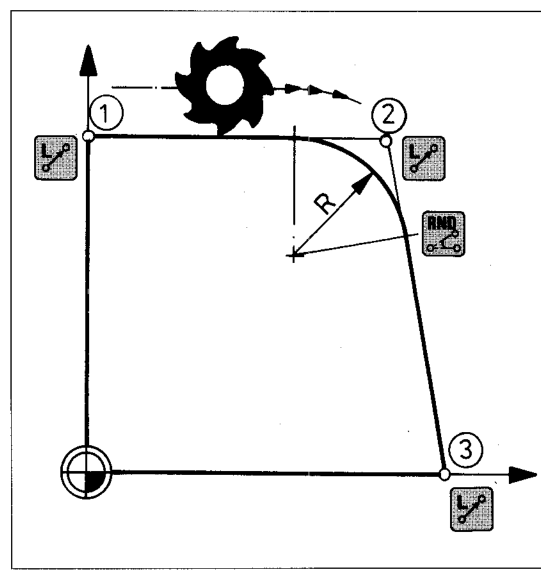
The circle center is not required for either function.



RND

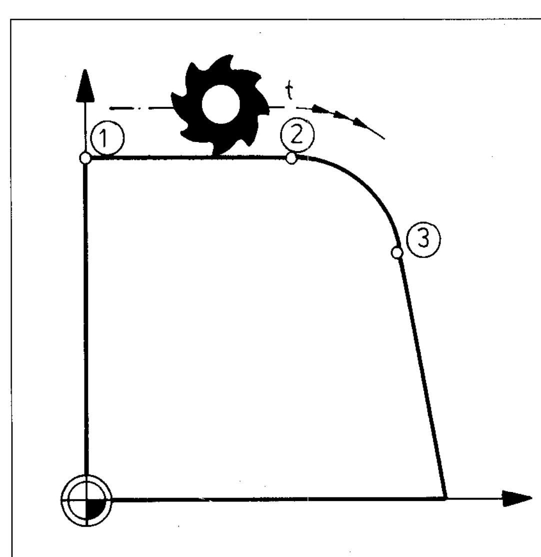
The "RND" rounding is inserted between two contour elements which can be either straight lines or arcs.

Program the **corner point** ② that is not approached and directly thereafter a separate rounding block "RND" with the rounding radius R. Entry and departure from the rounding is necessarily tangential and is automatically computed by the control.



CT

With "CT" **only the arc endpoint** ③ is to be programmed.



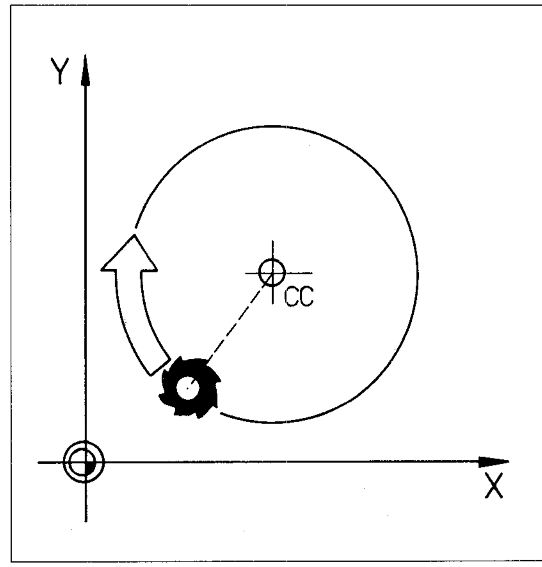
Selecting

Given	Select
Point ①	e.g. approach with L
Corner ②	e.g. approach with L
Rounding radius	RND
Point ③	e.g. approach with L
Tangent-forming point ①	e.g. approach with L
Tangential entry ②	e.g. approach with L
End point of circular arc ③	CT

Circular Movement/Cartesian CC + C

CC has two functions:

1. Specifying the circle center for circular arcs (to be programmed with "C").
2. Defining the **pole** for polar coordinates.



Circle center CC

The circle center CC must be programmed before circular interpolation with "C". The CC coordinates remain valid until changed by new CC coordinates.

There are three methods for programming CC:

- The circle center CC is directly defined by Cartesian coordinates.
- The coordinates last programmed in a CC block define the circle center.
- The current position is taken as CC with "NO ENT" or "END □" (without numerical input).

  This is also possible for positions programmed in polar coordinates.

The dialog for the circle center is initiated with the "CC" key.

CC absolute: the circle center is based on the work datum.
CC incremental: the circle center is based on the tool position last programmed.
"CC" produces no movement!

Approaching the starting point

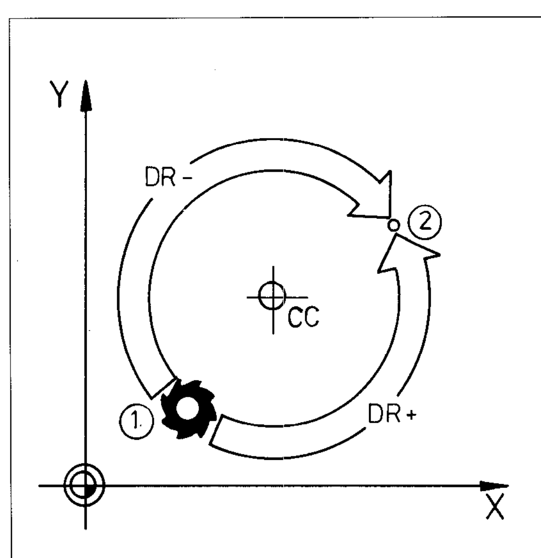
Approach the starting point for the circular arc before the C block.

Radius

The distance from the starting point to the circle center determines the radius.

Circle C

The tool is to travel from position ① to target point ② in a circular path. Only program ② in the C block. Position ② can be entered in Cartesian or polar coordinates.



Direction of rotation

The direction of rotation DR must be defined for circular movement:
rotation in positive direction **DR+** (counterclockwise)
rotation in negative direction **DR–** (clockwise).

Any tool radius compensation must begin before a circular arc.

Circular Movement/Cartesian CC + C

Input tolerance

The starting and endpoint must lie on the same circular path, i.e. they must be at the same distance from the circle center CC. The tolerance of position inputs for the starting position, end position and circle center is ± 8 µm.

Input CC CC X 50 Y 50 ENT — Circle center

Input C C X 15 Y 50 ENT — Arc endpoint

DR +/− END — Specify the rotating direction with the "+/−" key:
press once for −
press twice for +

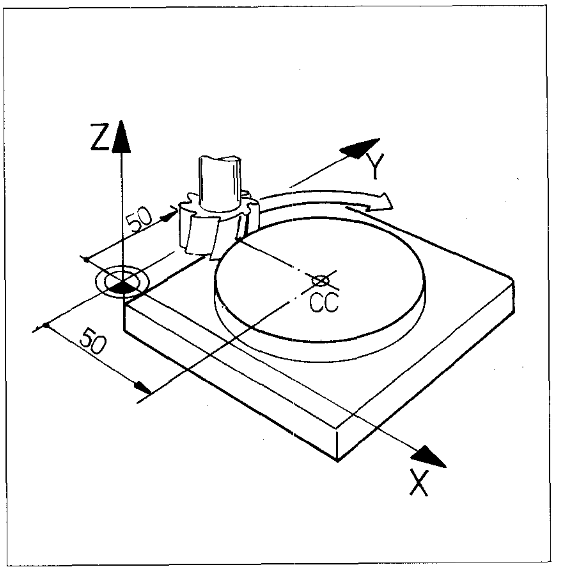
Program blocks

```
CC X+50 Y+50
C X+15 Y+50 DR-
```

Enter R, F and M as for straight lines.
Input is only needed to change earlier specifications.

Example full circle

Full circle in the XY plane (outer circle) around center X+50, Y+50 with 35 mm radius.



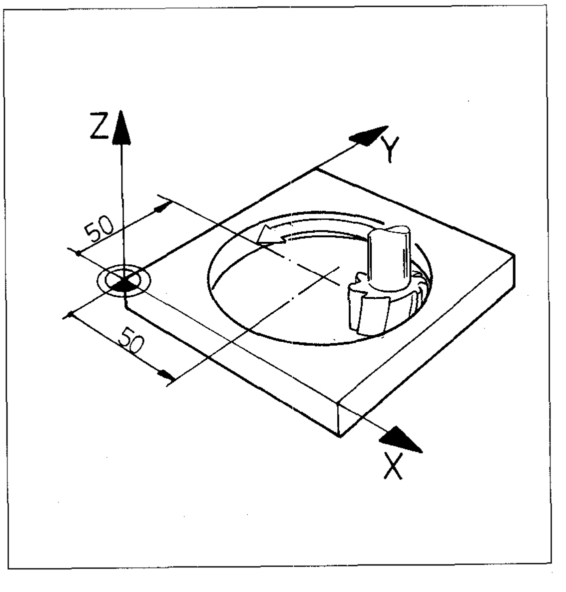
Program

```
TOOL DEF 1 L+0 R5
TOOL CALL 1 Z S200

L X+15 Y+50 RL F300 M3

CC X+50 Y+50

C X+15 Y+50     DR- RL
```

Full circles can be programmed with "C" in one block.
The circle starting point and the circle endpoint are identical.

Example arc

Semicircle in the XY plane (inside circle) around center X+50 Y+50 with 35 mm radius.



Program

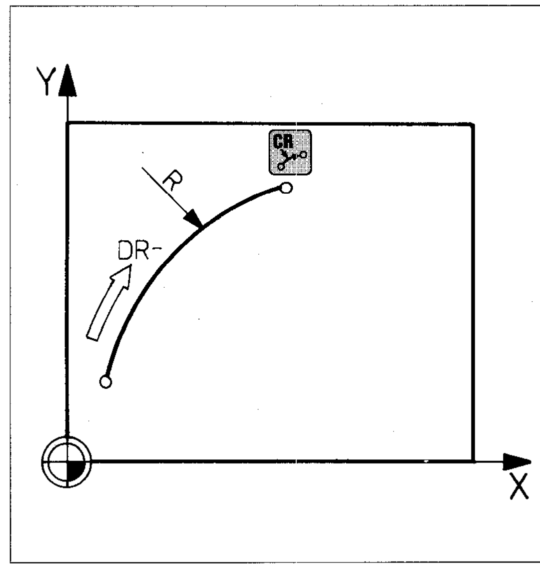
```
L X+85 Y+50 RL F300 M3

CC X+50 Y+50

C X+15 Y+50 DR+ RL
```

Circular Movement/Cartesian CR

Circle CR

If the contour radius is given in the drawing, but no circle center, the circle can be defined via the "CR" key with the
- endpoint of the circular arc
- radius and
- direction of rotation.

R, F and M are entered as for straight lines and are only required when changing earlier specifications.



Starting point

The starting point of the arc must be approached in the preceding block.

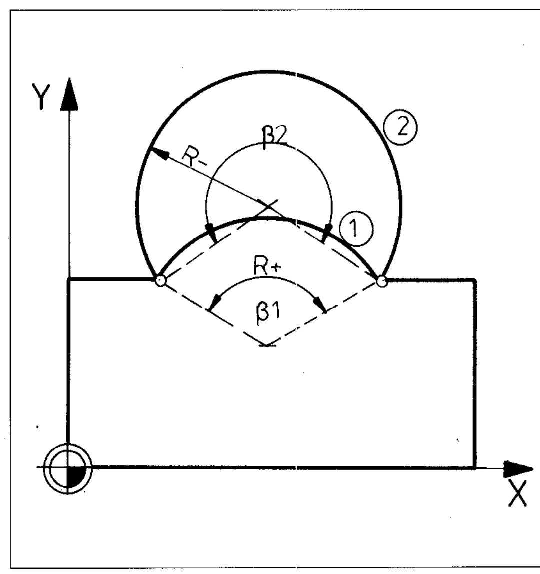
Endpoint

In the CR block the endpoint can only be programmed with Cartesian coordinates.

The distance between starting and end point of the arc must not exceed 2 x R. With CR, full circles can be programmed in 2 blocks.

Central angle

There are two geometric solutions for connecting two points with a defined radius (see figure), depending on the size of the central angle β:
The smaller arc 1 has a central angle $\beta_1 < 180°$, the larger arc 2 has a central angle $\beta_2 > 180°$.



Contour radius

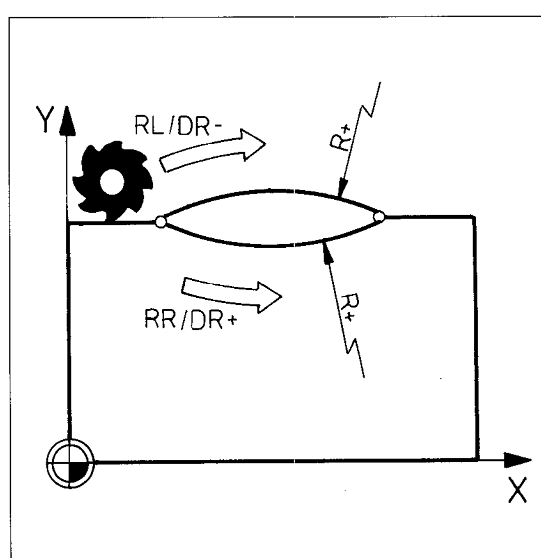
Enter a **positive** radius to program the smaller arc ($\beta < 180°$).
(The + sign is automatically generated.)

To program the larger arc ($\beta > 180°$), enter the radius as a **negative** value.

The maximum definable radius = 30 m.
Arcs up to 99 m can be produced with parametric programming.

Rotating direction

Depending on the allocation of radius compensation RL/RR, the rotating direction determines whether the circle curves inward (= concave) or outward (= convex).
In the adjacent figure, DR– produces a convex contour element, DR+ a concave contour element.



Circular Movement/Cartesian CR

Input CR	CR X 80 Y 40 ENT	Endpoint of arc

Input CR	CR X 80 Y 40 ENT	Endpoint of arc
	R+100 ENT	Radius, positive sign
	DR +/− ENT	Direction of rotation specified with the change sign key.

Program block CR X+80 Y+40 R+100 DR–

Examples:

TOOL DEF 1 L+0 R5
TOOL CALL 1 Z S200

Arc A

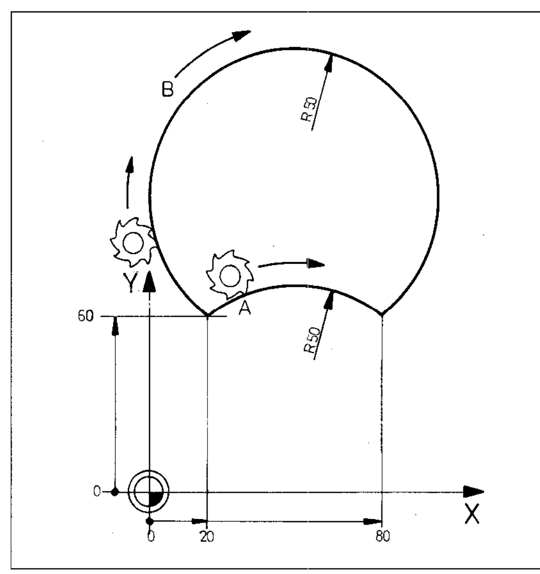
L X+20 Y+60 RL F300 M3

CR X+80 Y+60 **R+50** **DR–**

Arc B

L X+20 Y+60 RL F300 M3

CR X+80 Y+60 **R–50** **DR–**



Arc C

L X+20 Y+60 RL F300 M3

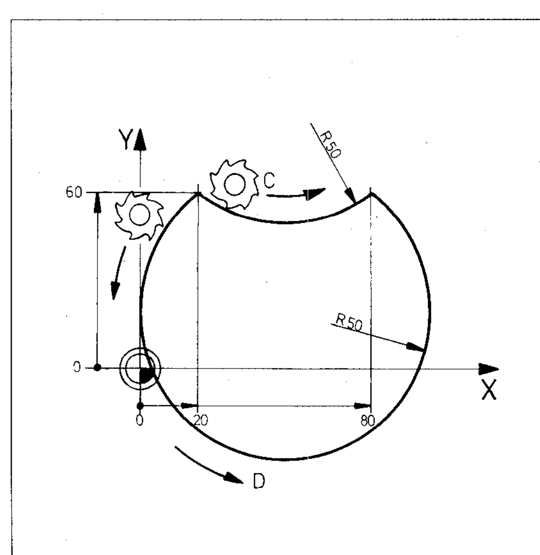
CR X+80 Y+60 **R+50** **DR+**

Arc D

L X+20 Y+60 RL F300 M3

CR X+80 Y+60 **R–50** **DR+**

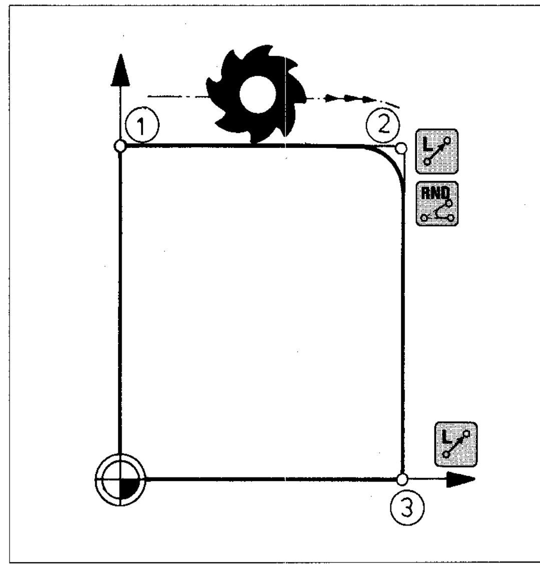
The position X+20 Y+60 is the start of arc in the examples; the position X+80 Y+60 is the end of arc.



Circular Movement/Cartesian
Corner rounding RND

"RND" has two functions:

- rounding of corners, if RND is "in the contour",
- soft approach and departure from the contour, if RND is at the start or end of the contour.



Circular arc

Contour corners can be rounded with arcs. The circle connects tangentially with the preceding and succeeding contour.

A rounding arc can be inserted at any corner formed by the intersection of the following contour elements:

- straight line – straight line,
- straight line – circle, or circle – straight line,
- circle – circle.

Prerequisites

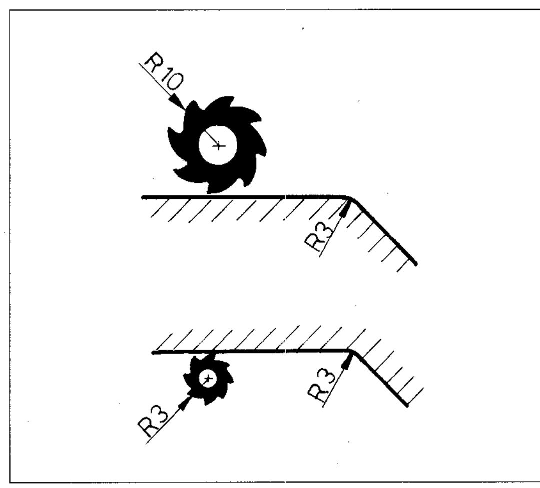
Rounding is completely defined by the RND block and the points ① ② ③. A positioning block containing both coordinates of the machining plane should be programmed before and after the RND block. The RL/RR/R0 compensation must be identical before and after the RND block.

Note

The rounding arc can only be executed in the machining plane. The machining plane must be the same in the positioning block before and after the rounding block.

The rounding radius cannot be too large or too small for inside corners – it must "fit between the contour elements" and be machinable with the current tool.

The feed rate for corner rounding is effective blockwise. The previously programmed feed rate is reactivated after the RND block.



The tool radius can be larger than the rounding radius on outside corners.

The tool radius must be smaller than or equal to the rounding radius on inside corners.

Programming

The rounding arc is programmed as a separate block following the corner to be rounded. Enter the rounding radius and a reduced feed rate F, if needed. The "corner point" itself is not approached!

Error messages

PLANE WRONGLY DEFINED
The machining planes are not identical before and after the RND block.

ROUNDING RADIUS TOO LARGE
The rounding is geometrically impossible.

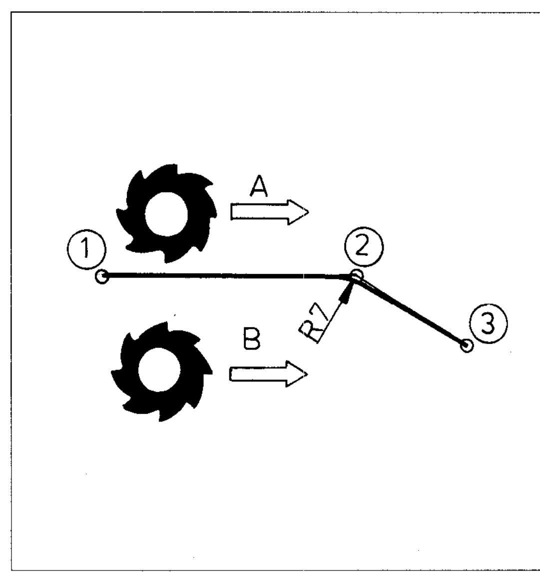
Circular Movement/Cartesian
Corner rounding RND

Input RND	RND 8	ENT or END	Rounding radius
	F100	ENT	A separate feed rate can be entered and is only effective for this rounding.

Program block RND 8 F100

Examples: TOOL DEF 1 L+0 R5
TOOL CALL 1 Z S200

Sequence A	L X+10 Y+60 RL F300 M3	Position ①
	L X+50 Y+60	"Corner point" ②
	RND 7	Rounding
	L X+90 Y+50	Position ③
Sequence B	L X+10 Y+60 RR F300 M3	Position ①
	L X+50 Y+60	"Corner point" ②
	RND 7	Rounding
	L X+90 Y+50	Position ③



Circular arc CT

A circular arc can be programmed more easily if it connects tangentially to the preceding contour. The circular arc is defined by **merely entering the arc endpoint** with the "CT" key.

Geometry

An arc with tangential connection to the contour is exactly defined by its endpoint.

This arc has a specific radius, a specific direction of rotation and a specific center. This data need not therefore be programmed.

Prerequisites

The contour element which connects tangentially to the circle is programmed immediately before the tangential arc. Both coordinates of the same machining plane must be programmed in the block for the tangential arc and in the preceding block.

Tangent

The tangent is specified by **both** positions ① and ② directly preceding the CT block. Therefore, the first CT block can appear no earlier than the third block in a program.

Circular arc CT

The tool is to travel a circle connecting tangentially to ① and ② to target point ③. Only ③ is programmed in the CT block.

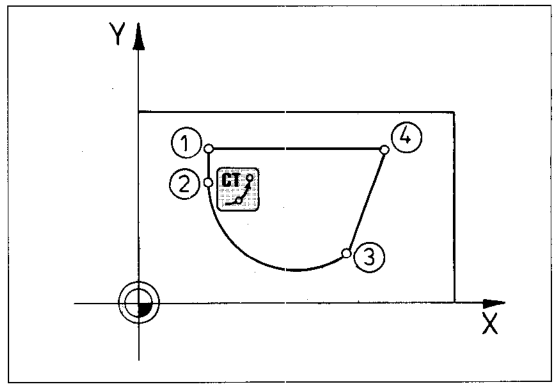
Coordinates

The endpoint of the circular path can be programmed in either Cartesian or polar coordinates.

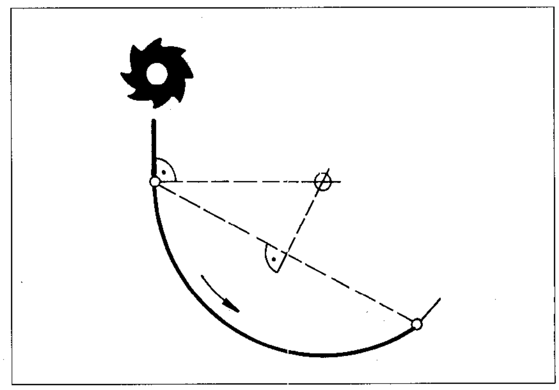
Error messages

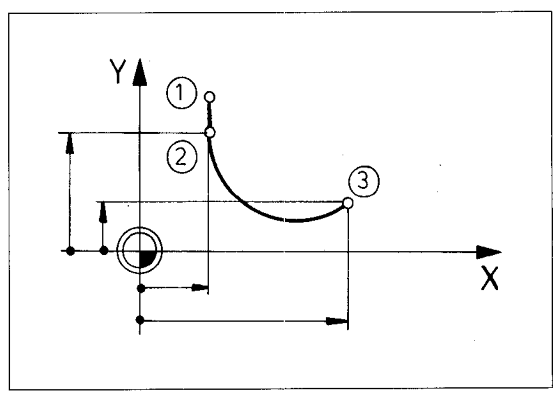
WRONG CIRCLE DATA
The required minimum 2 positions before the CT block were not programmed.

ANGLE REFERENCE MISSING
Both coordinates of the machining plane are not given in the CT block and the preceding block.

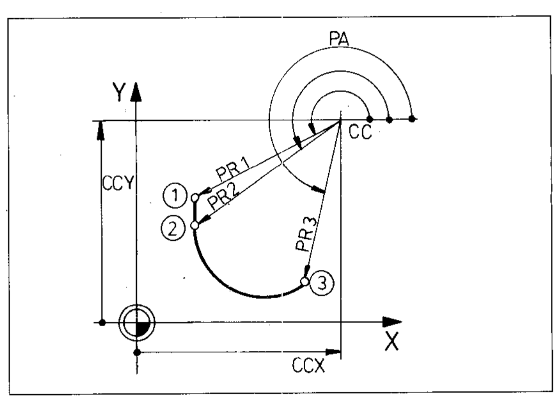


Machining sequence


Geometry



Cartesian coordinates



Polar coordinates

Circular Movement/Cartesian
Tangential arc CT

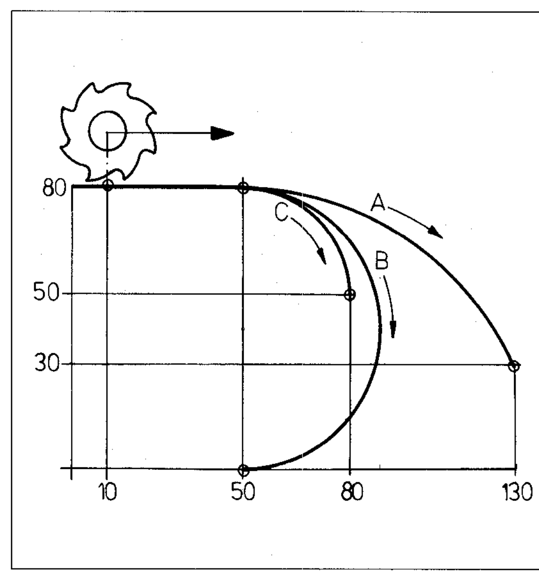
Input CT [CT] [X] 90 [Y] 40 [ENT] Arc endpoint

Program block CT X+90 Y+40

Enter R, F and M as for straight lines.
Input is only necessary to change earlier definitions.

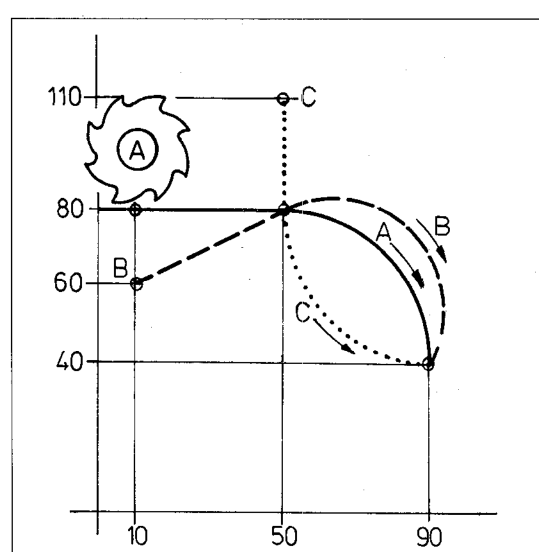
Examples: different endpoints

TOOL DEF 1 L+0 R10
TOOL CALL 1 Z S200



Arc A	L X+10 Y+80 RL F300 M3	1st tangent point
	L X+50 Y+80	Start of arc
	CT X+130 Y+30	End of arc
Arc B semicircle	L X+10 Y+80 RL F300 M3	1st tangent point
	L X+50 Y+80	Start of arc
	CT X+50 Y+0	End of arc. A semicircle with R = 40 is formed.
Arc C quarter circle	L X+10 Y+80 RL F300 M3	1st tangent point
	L X+50 Y+80	Start of arc
	CT X+80 Y+50	End of arc. A quarter circle with R = 30 is formed.

Different tangents



Arc A

L X+10 Y+80 RL F300 M3
L X+50 Y+80
CT X+90 Y+40

Arc B

L X+10 Y+60 RL F300 M3
L X+50 Y+80
CT X+90 Y+40

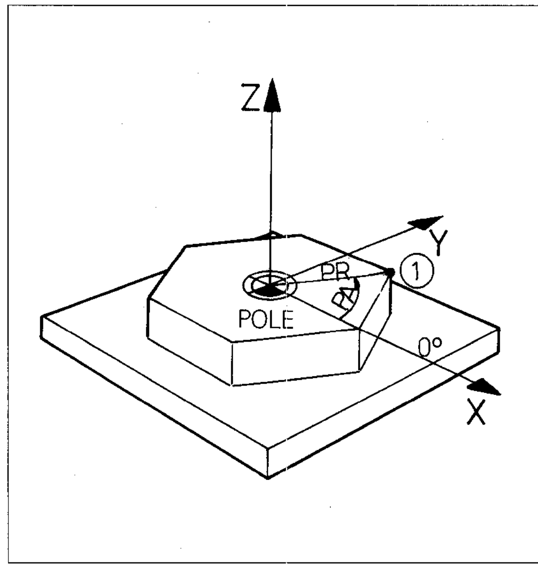
Arc C

L X+50 Y+110 RL F300 M3
L X+50 Y+80
CT X+90 Y+40

Polar Coordinates
Fundamentals

The control allows you to enter nominal positions in either Cartesian or polar coordinates.
In polar coordinates, the points in a plane are specified by the
polar radius PR (distance to the pole), and the **polar angle PA** (angular direction).

The pole position is entered with the "CC" key in Cartesian coordinates based on the workpiece datum.



Marking

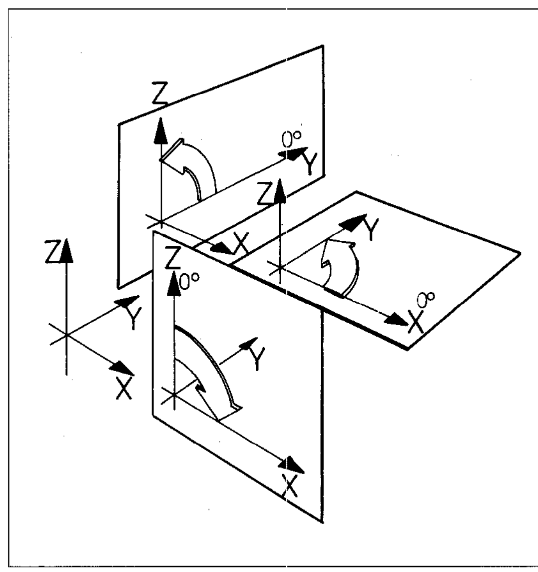
Blocks in polar coordinates are marked by a P (LP, CP etc.).

Angle reference axis

The angle reference axis (0° axis) is the
+X axis in the XY plane,
+Y axis in the YZ plane,
+Z axis in the ZX plane.

The machining plane (e.g. XY plane) is determined by a tool call.

The sign of the angle PA can be seen in the adjacent figure.



Absolute polar coordinates

Absolute dimensions are based on the current pole.
Example: LP PR+50 PA+40

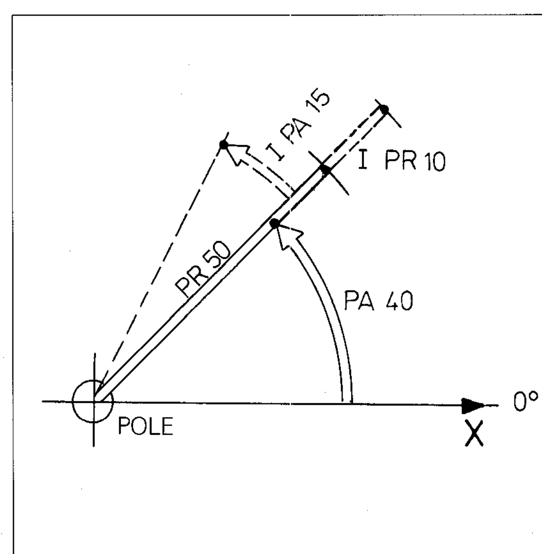
Incremental polar coordinates

A polar coordinate radius entered incrementally changes the last radius.
Example: LP IPR+10

An incremental polar coordinate angle IPA refers to the last direction angle.
Example: LP IPA+15

Mixing

Absolute and incremental coordinates may be mixed within one block.
Example: LP PR+50 IPA+15



Polar Coordinates
Pole

Pole

The pole must be specified with "CC" before entering polar coordinates. The pole can be set anywhere in the program prior to using polar coordinates.

The pole is programmed in absolute or incremental Cartesian coordinates.

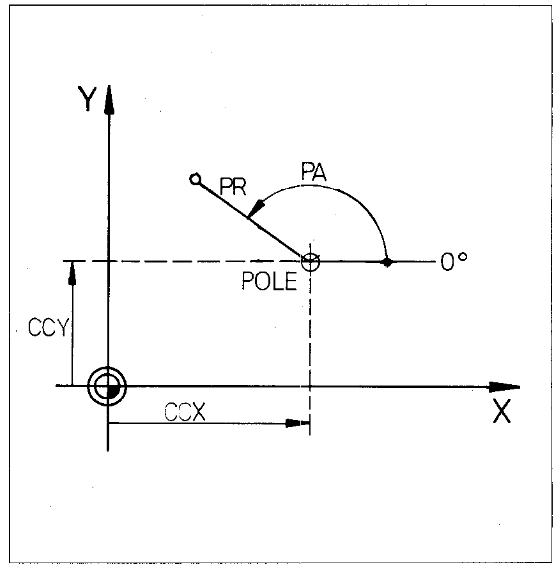
CC absolute: the pole is referenced to the workpiece datum.

CC incremental: the pole is referenced to the last programmed nominal tool position.

A CC block is programmed with the coordinates of the machining plane.

Example

CC X+60 Y+30



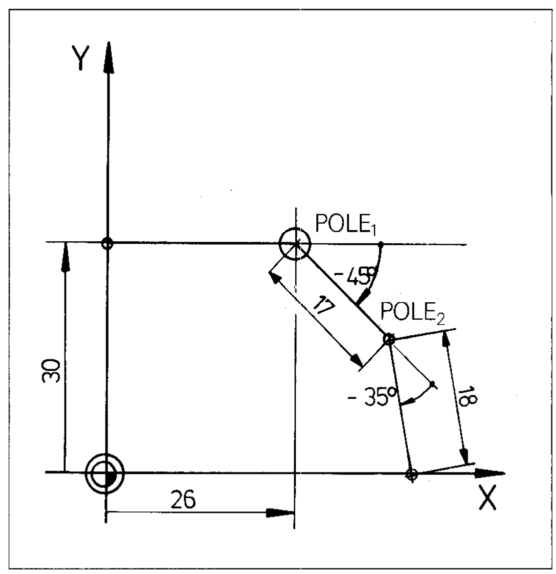
Transferring the pole

The last **programmed position** is **transferred** as the pole. Program an empty CC block.

Directly transferring the pole in this manner is especially well suited for polygon shapes with polar dimensions (see illustration below).

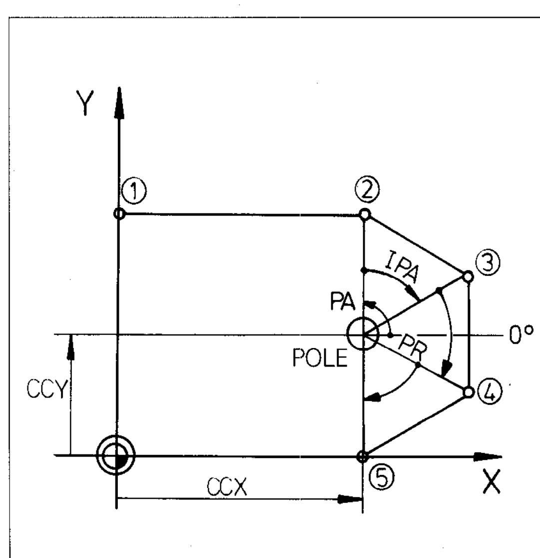
Example

L X+26 Y+30	
CC	POLE 1
LP PR+17 PA–45	
CC	POLE 2
LP PR+18 IPA–35	



Modal effect

A pole definition remains valid in a program until it is overwritten with another definition. The same pole therefore need not be programmed repeatedly.



Polar Coordinates
Straight line LP

After opening with the "L" key, you must press the "P" key to enter positions in polar coordinates.

For dimensions which are referenced to a rotational axis in some way, such as bolt hole circles or cams, programming is usually easier in polar coordinates than in Cartesian coordinates because calculations are avoided.

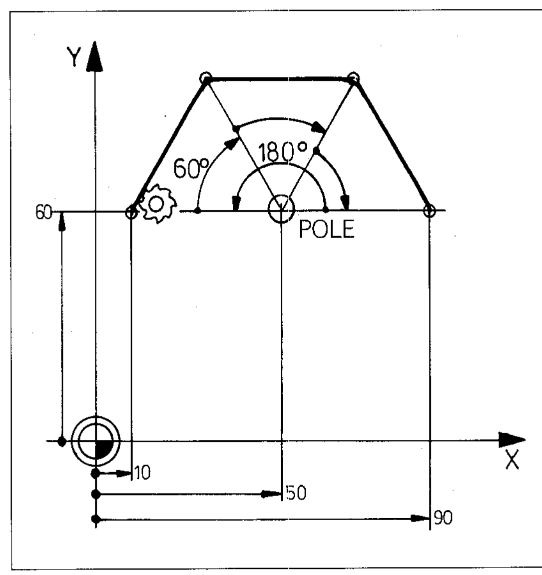
Range for polar angle PA

Input range for linear interpolation: absolute or incremental –360° to +360°.

PA positive: counterclockwise angle.
PA negative: clockwise angle.

Example

Milling an inside contour:

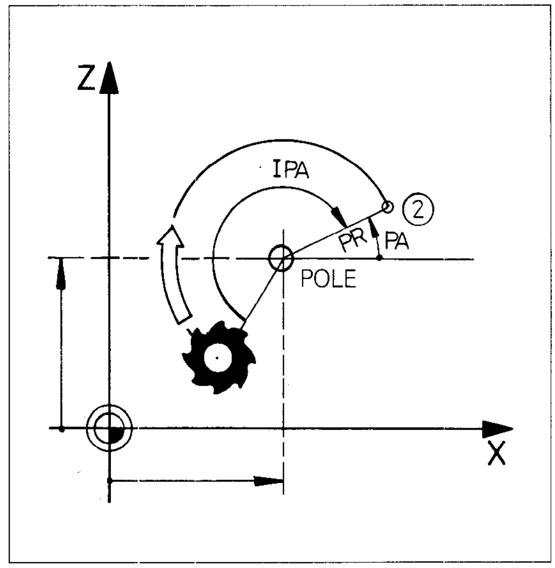


Program

TOOL DEF 2 L+0 R2 TOOL CALL 2 Z S200	
CC X+50 Y+60	Set POLE
L X+15 Y+50 R0 F1000 M3	Approach starting point externally (Cartesian coordinates)
L Z–5 F100	Plunge
LP PR+40 PA+180 RR F200	Approach 1st contour point with compensation (polar coordinates)
LP IPA–60	2nd contour point
LP IPA–60	
LP IPA–60	Last contour point
L X+85 Y+50 R0	Depart from contour, cancel compensation
L Z+50 R0 F9999 M2	Retract

Circular arc

If the target point on the arc is programmed in polar coordinates, you only have to enter the polar angle PA to define the endpoint. The radius is defined by the distance from the starting point of the arc to the programmed circle center CC.



When programming a circle in polar coordinates, the angle PA and the rotating direction DR can be entered positively or negatively. The angle PA determines the endpoint of the arc.

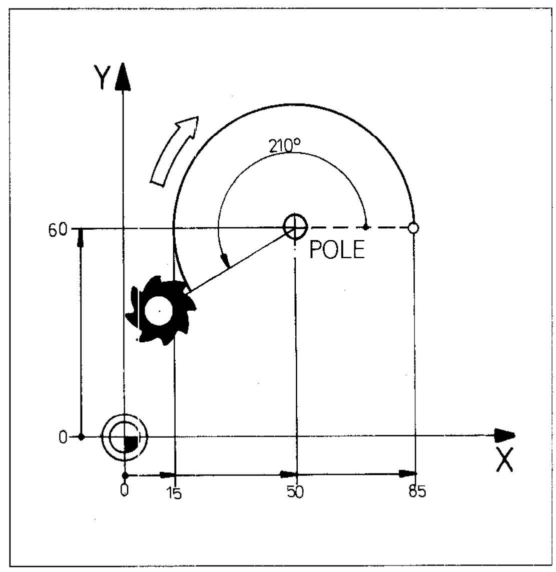
If the angle PA is entered incrementally, the sign of the angle and the sign of the rotating direction should be the same. In the figure to the right, this means that IPA is negative and DR is also negative.

Range for polar angle PA

Input range for circle interpolation:
absolute or incremental –5400° to +5400°.

Example

An arc with radius 35 and circle center X+50 Y+60 is to be milled.
Rotating direction is clockwise.



Program

TOOL DEF 1 L+0 R5
TOOL CALL 1 Z S200

CC X+50 Y+60	Coordinates of circle center
LP PR+35 PA+210 RL F200 M3	Approach circle (circle radius is 35 mm)
C PA+0 DR- F300	Circular movement clockwise

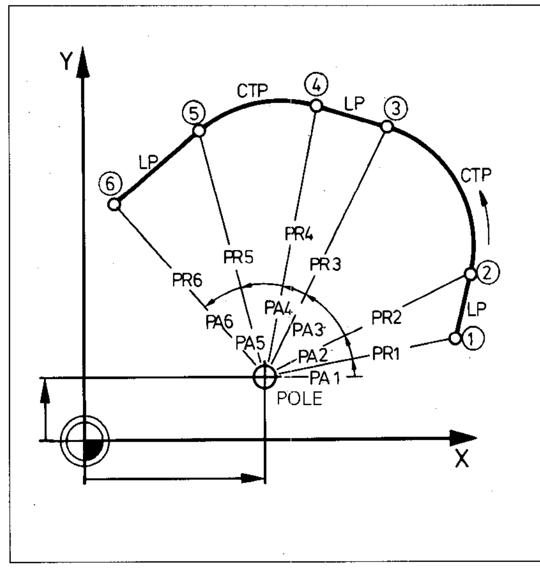
In the example, a contour radius of 35 mm is obtained from the distance between the POLE and the approach point on the circle.

Polar Coordinates
Tangential arc CTP
Corner rounding RND

Tangential arc

The endpoints of tangential arcs may be entered in polar coordinates to simplify the programming of, for example, cams.

The start of the arc is automatically tangential when programming with CT.

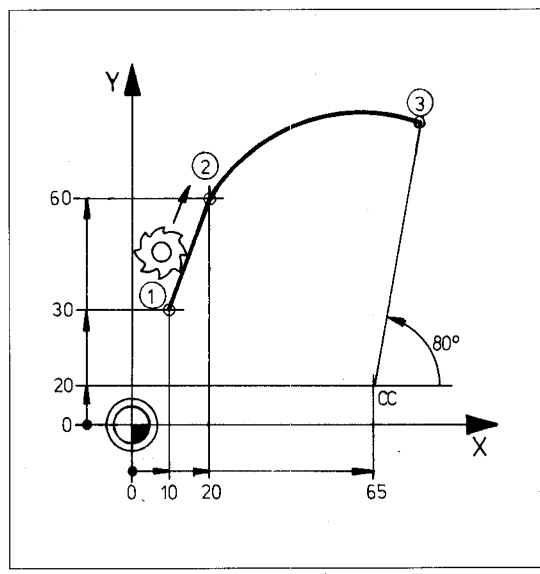


> If the transition point are not calculated exactly, the arc elements could become "jagged".

Specify the pole CC before programming in polar coordinates.

Example

A straight line through ① and ② is to tangentially meet the arc to ③. The radius and direction angle of ③ with respect to CC are known.



Program

```
TOOL DEF 1 L+0 R4
TOOL CALL 1 Z S200

CC X+65 Y+20

L X+10 Y+30 RL F500 M3

L X+20 Y+60

CTP   PR+70   PA+80
```

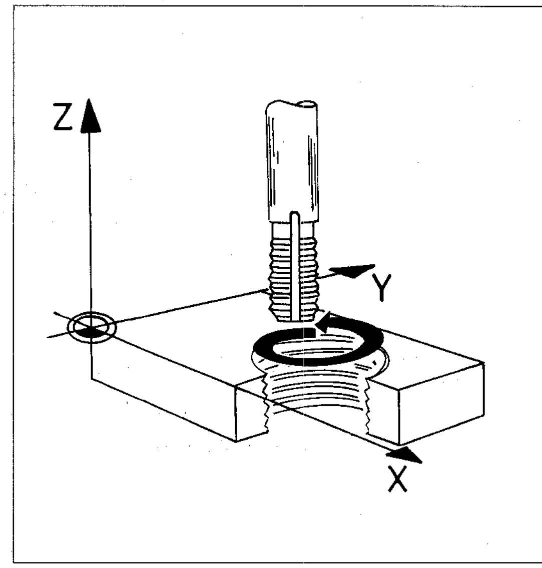
Polar "corners" can also be rounded with the "corner rounding" function (see Circular Movement/Cartesian, Corner rounding RND).

Polar Coordinates
Helical interpolation (CC + CP) + Z

+ Z

Helix

If 2 axes are moved simultaneously to describe a circle in a main plane (XY, YZ, ZX), and a uniform linear motion of the tool axis is superimposed, then the tool moves along a helix (helical interpolation).



Applications

Helical interpolation can be used to advantage with form cutters for producing internal and external threads with large diameters, or for lubricating grooves. This can save you substantial tool costs.

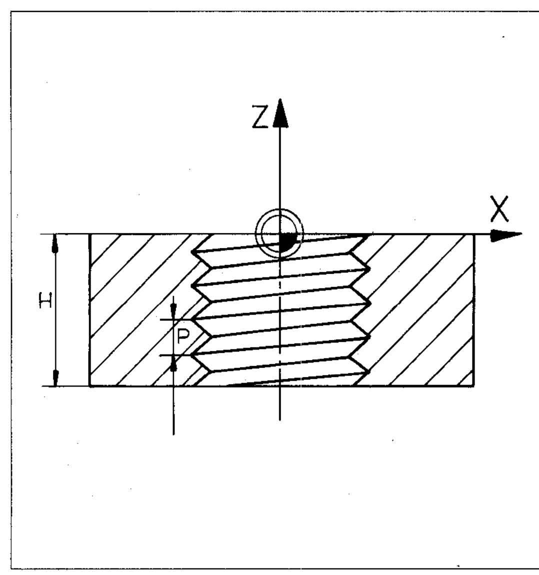
Input data

The helix is programmed in polar coordinates.

First specify the POLE or circle center CC.

Angle range

Enter the total angle of tool rotation for the polar angle IPA in degrees:
IPA = number of rotations x 360°
Maximum angle of rotation: ± 5400° (15 complete rotations).



Height

The total height H (= IZ) is entered for the tool axis (Z) at the query "Coordinates".

Calculate the value from the thread pitch and the required number of tool rotations.
IZ = P · n, IZ = total height/depth to be entered
P = pitch
n = number of threads
The total height/depth can be entered in absolute or incremental dimensions.

Thread

A complete thread can be programmed quite easily with IZ and IPA; the number of threads is then specified with a program section repeat REP.

Radius compensation

The radius compensation depends upon the

- rotating direction (right/left),
- type of thread (internal/external),
- milling direction (positive/negative axis direction)
(see table to the right).

Internal thread	Working direction	Rotating direction	Radius compensation
right-hand	Z+	DR+	RL
left-hand	Z+	DR–	RR
right-hand	Z–	DR–	RR
left-hand	Z–	DR+	RL

External thread	Working direction	Rotating direction	Radius compensation
right-hand	Z+	DR+	RR
left-hand	Z+	DR–	RL
right-hand	Z–	DR–	RL
left-hand	Z–	DR+	RR

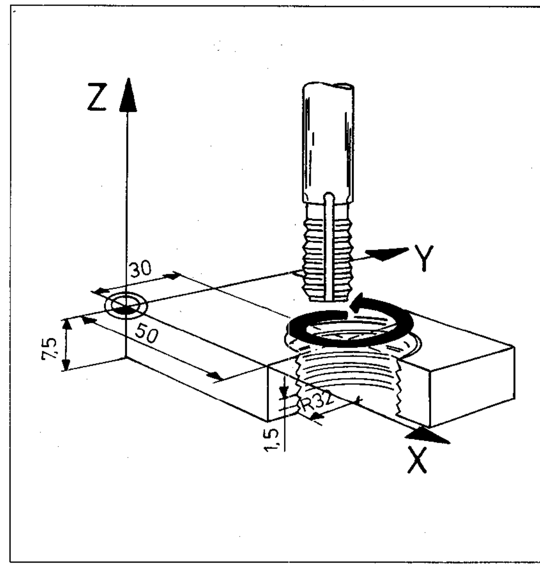
Polar Coordinates
Helical interpolation (CC + CP) + Z

Input example

CC P I 360 I Z 2 — Endpoint

+/– +/– END — Rotating direction

CP IPA+360 IZ+2 DR+



Task

A right-hand internal thread M64 x 1.5 is to be produced in one cut with a multi-cutter tool.

Thread

Thread data:

pitch	$P = 1.5$ mm
start	$\alpha_a = 0°$
end	$\alpha_e = 0° = 360°$
Number of threads	$n_0 = 5$
Overrun of threads:	
at start	$n_1 = 1/2$
at end	$n_2 = 1/2$

Calculations

Total height:
$IZ = P \cdot n = 1.5\ \text{mm} \cdot [5 + (2 \cdot 1/2)] = 9\ \text{mm}$

Incremental polar angle:
$IPA = 360° \cdot n = 360° \cdot [5 + (2 \cdot 1/2)] = 2160°$

Due to overrun of 1/2 thread, the start of thread is advanced by 180°:
starting angle $\alpha_s = \alpha_a + (-180°) = 0° + (-180°) = -180°$

The overrun of 1/2 thread at the start of thread gives the following initial value for Z:
$Z = -P \cdot n = -1.5\ \text{mm} \cdot [5 + 1/2] = -8.25\ \text{mm}$

Program

TOOL DEF 1 L+0 R20
TOOL CALL 1 Z S500

L X+50 Y+30	Approach the hole center
CC	Take the position as pole
L Z–8.25 R0 FMAX M3	Downfeed at center to **initial value Z**
LP PR+32 PA–180 RL F100	Approach the wall with radius R and **starting angle** α_s
CP IPA+2160 IZ+9 DR+ RL F200	Helical movement with **incremental angle IPA** and **total height IZ**
L X+50 Y+30 M05	Retract in XY
L Z+100 FMAX	Retract in Z

Note

Helical interpolation cannot be graphically displayed.

Contour Approach and Departure
Starting and end position

Selecting the 1st contour point

Before beginning contour programming, specify the first contour point at which machining **with radius compensation** is to begin.

Starting point

In the vicinity of the first contour point, define an **uncompensated starting point** that can be approached in rapid traverse, and be sure to consider the tool in use. The starting point must fulfill the following criteria:

- approachable without collision
- near the first contour point
- outside the material
- the contour will not be damaged when approaching the first contour point.

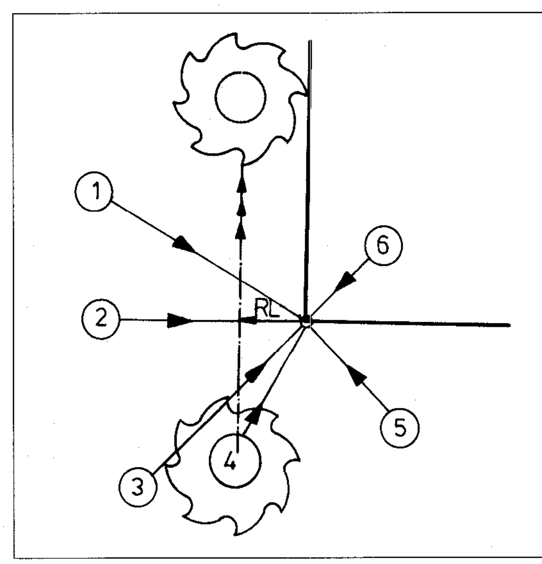
Direct approach

When working on a circle (RND) without the TNC approach/departure function, also check that the tool does not blemish the contour due to a direction change.

Starting points

① Not recommended	Surface blemish due to change of Y-axis direction
② Not recommended	
③ **Suitable**	**Also for end point**
④ **Optimal**	Lies on the extension of the compensated path
⑤ Not recommended	Contour damage
⑥ Not permitted!	

Radius compensation must remain switched off for the starting position (R0).



End points

The same prerequisites apply for selecting the **uncompensated end point** as for the starting point.

The ideal end point ④ lies on the extension of the last contour element RL.

①, ② Not recommended	Surface blemish due to change of the X-axis direction
③ **Suitable**	**Also for the starting point**
④ **Optimal**	Lies on the extension of the compensated path
⑤ Not recommended	Contour damage
⑥ Not permitted!	

Radius compensation must be switched off after departure from the contour (R0).

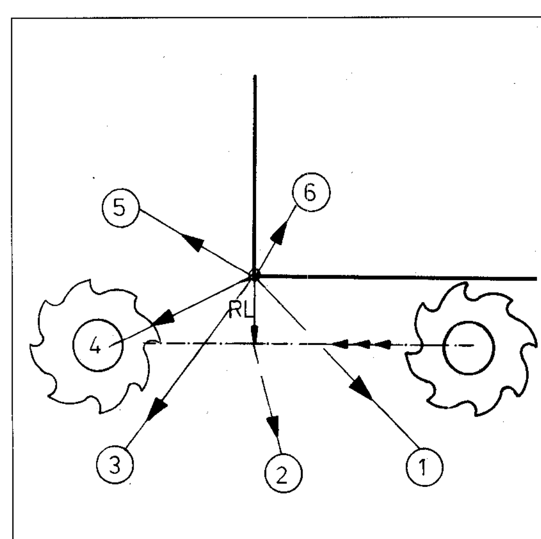


Illustration
——— programmed path
– · – · – traversed cutter center path

Common starting and end point

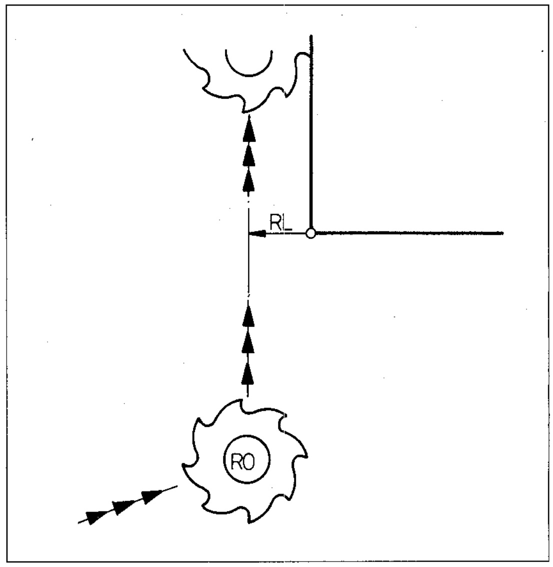
For a **common starting and end point**, select point ③ on the bisecting line of the angle between the first and last contour element.

Contour Approach and Departure
Starting and end position

Approach

The starting position Ⓢ must be programmed without radius compensation, i.e. with R0.

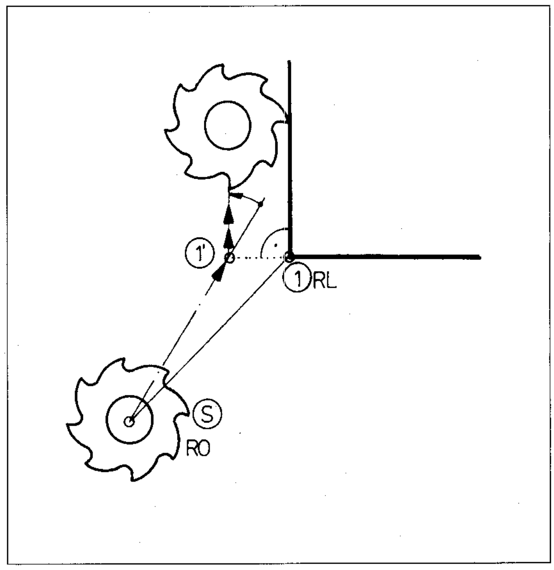
The control guides the tool in a straight line from the uncompensated position Ⓢ to the compensated position ①' of contour point ①. The tool center is then located perpendicular to the start of the first radius-compensated contour element.



Departure

At a transition from RL/RR to R0, the control positions the tool center in the last radius compensated block (RL) perpendicular to the end of the last contour section.

Then the next uncompensated position is approached with R0.

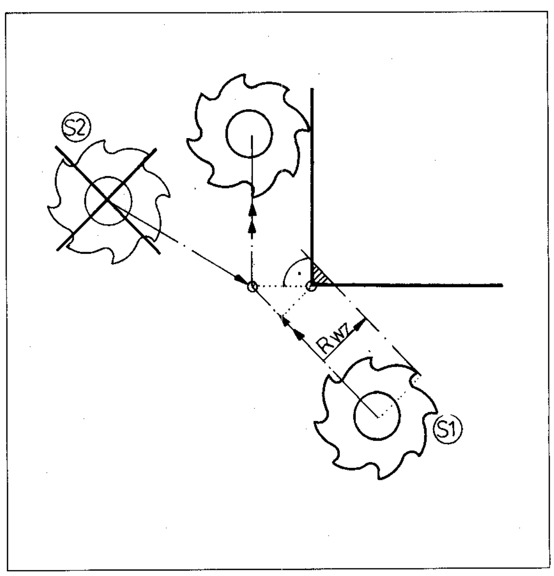


Approaching from an unsuitable position

If radius compensation is begun from S1, the tool will damage the contour at the first contour point if no extra measures are taken!

Departure

The same applies when departing from the contour.



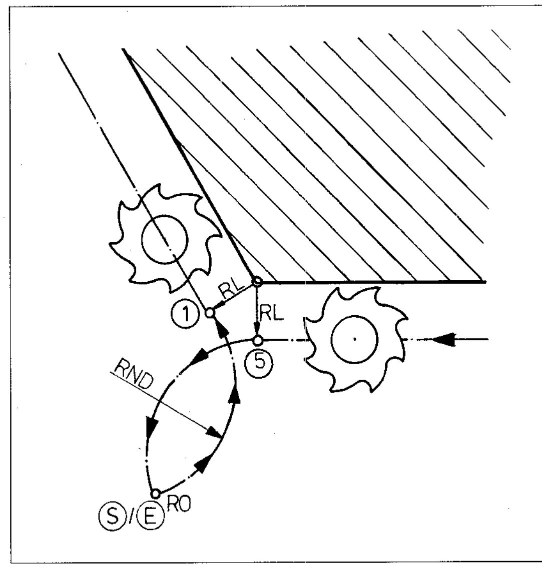
Contour Approach and Departure
on an arc

Approach and departure on an arc

The TNC enables you to automatically approach and depart from contours on a circular path.

RND

Begin programming with the "RND" key.



Approach

The tool traverses from the starting position Ⓢ initially on a straight line and then on a tangentially connected arc to the programmed contour.

The starting point can be selected as desired, and is approached without radius compensation (with R0).

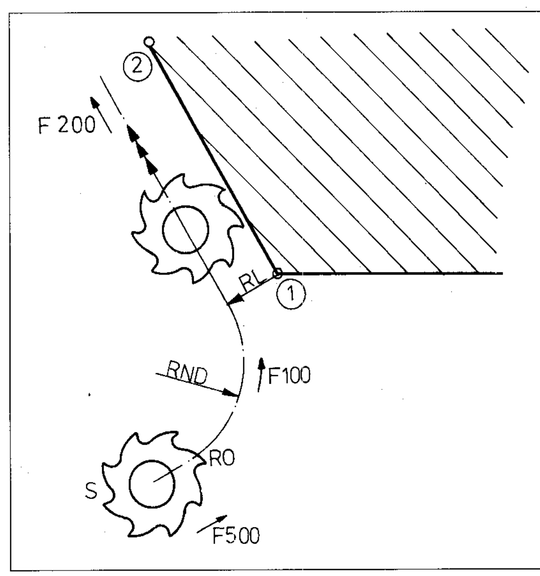
The straight line positioning block to contour point ① must contain radius compensation (RL or RR).

Then program a RND block.

Departure

The tool travels from the last contour point ⑤ on a tangentially connecting arc and then on a tangentially connecting straight line to the end position Ⓔ.

The positioning block for Ⓔ should not contain radius compensation (i.e. R0).



Approach arc/ departure arc

The radius R can be substantially less than the tool radius. It must be small enough to fit between Ⓢ and ① or ⑤ and Ⓔ.

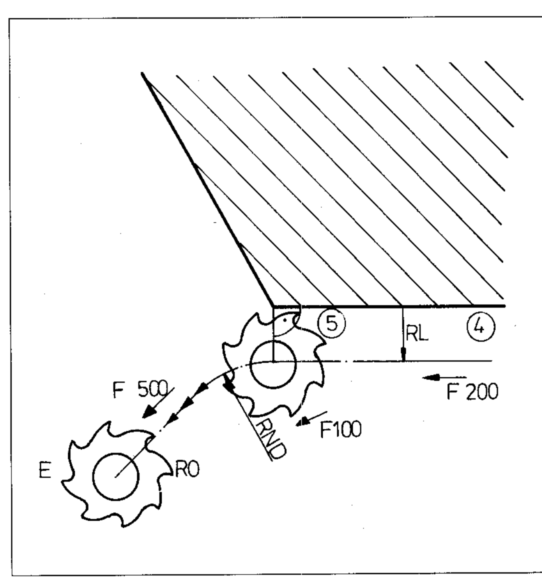
Feed rate

A feed rate exclusively for the approach and departure arc can be programmed separately in the RND block.

Program scheme

L X_S Y_S Z_S **R0 FMAX**
L X_1 Y_1 **RL F500**
RND 2.5 F100
L X_2 Y_2 F200
:

:
L X_5 Y_5 **RL F200**
RND 2.5 F100
L X_E Y_E **R0 F500**
Z200 FMAX



Notes

A positioning block containing both coordinates of the machining plane must be programmed before and after the RND block.

Approach on an arc:

Program a RND block after the first radius compensated position (RL/RR).

Departure on an arc:

Program a RND block after the last radius compensated position (RL/RR), or before the first uncompensated position following machining.

Predetermined M Functions
Constant contour speed: M90

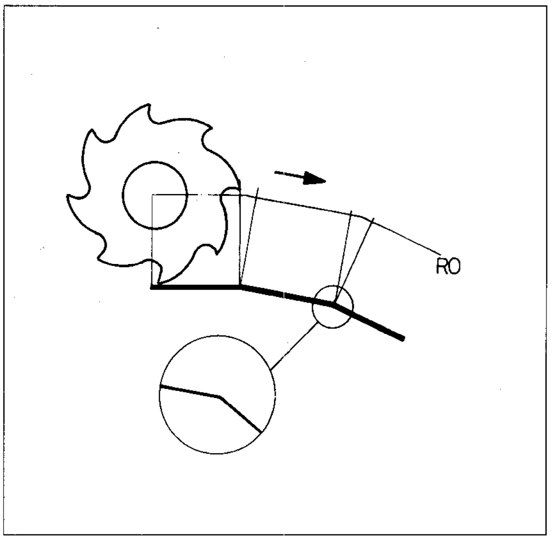
Standard practice: automatic deceleration at corners

For angular transitions such as internal corners and contours with R0, the axes are stopped briefly because an abrupt change of direction is not mechanically possible.

This protects the machine and results in sharp definition of corners.

For some tasks it is advantageous not to stop at corners.

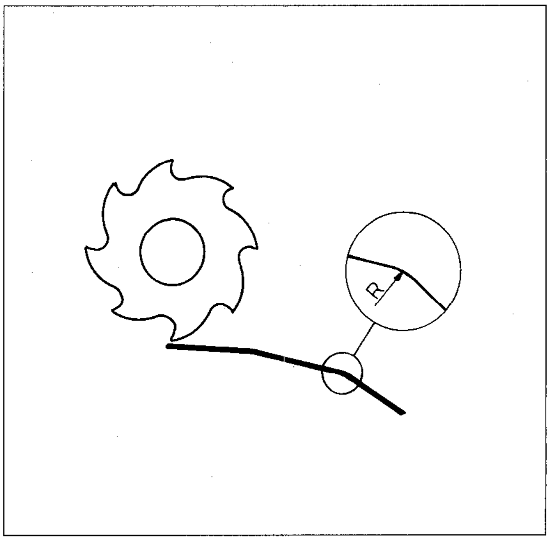
Example:
The contour of a free-form surface produced with a large number of short linear movements. Here it is desireable to smooth the corners.



Without M90

M90

The corners are **smoothed** if **M90** is programmed in every block. The workpiece is smoother and can be machined faster. M90 prevents stoppage of the axes blockwise for R0 or internal corners.



With M90

Drawbacks

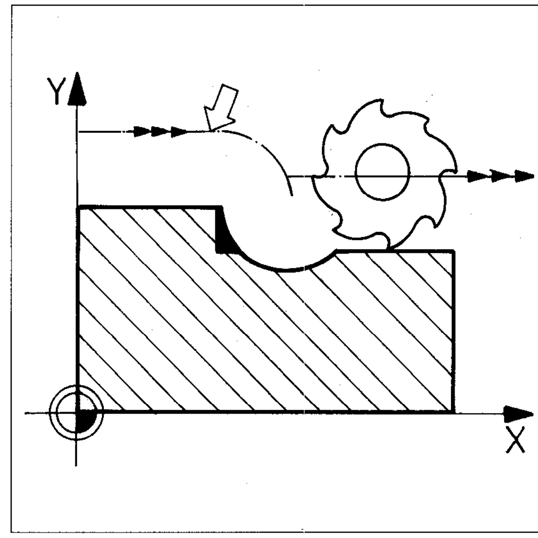
Greater strain on the machine at sharper changes of direction, until safety limit is reached (specified by the machine manufacturer).

Note

The exact execution depends on the machine parameters. Contact the machine manufacturer for more information.

Small contour steps: M97

If there is a step in the contour which is **smaller** than the tool radius, the standard transition arc would cause contour damage. The control therefore issues an error message and does not execute the corresponding positioning block.



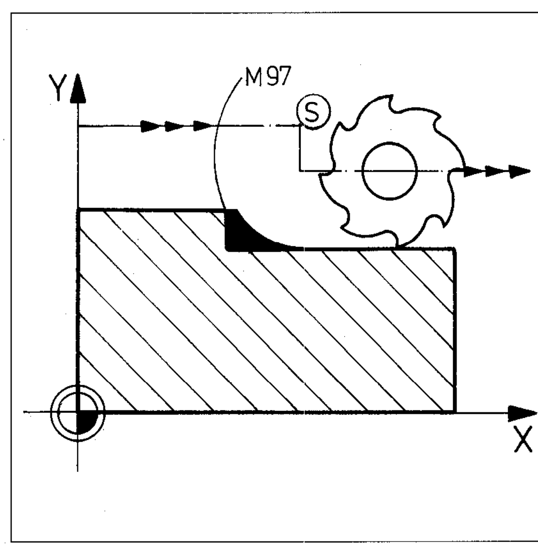
Without M97

M97

M97 prevents insertion of the transition arc. The control then determines a contour intersection Ⓢ as at inside corners and guides the tool over this point. The contour is not damaged.

However, machining is then incomplete and the corner may have to be reworked. A smaller tool may help.

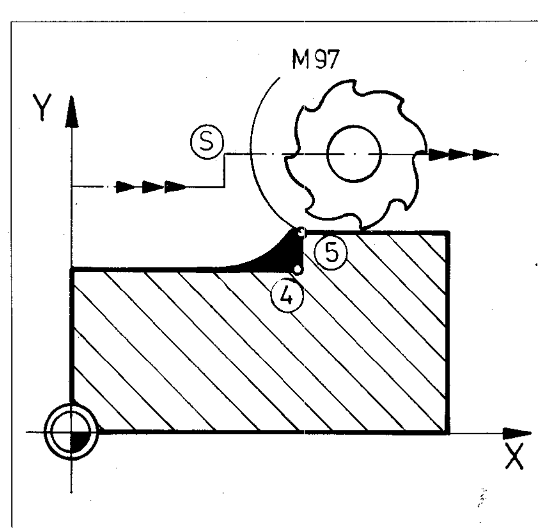
M97 is effective blockwise and must be programmed in the block containing the outside corner point.



With M97

Example

```
TOOL DEF 1 L+0 R10
TOOL CALL 1 Z S 100

L X+0 Y+30 RL F200 M3      ①
L X+40 Y+30 M97            ②
L X+40 Y+28                ③
L X+80 Y+28                ④
L X+80 Y+30 M97            ⑤
L X+100 Y+30
```

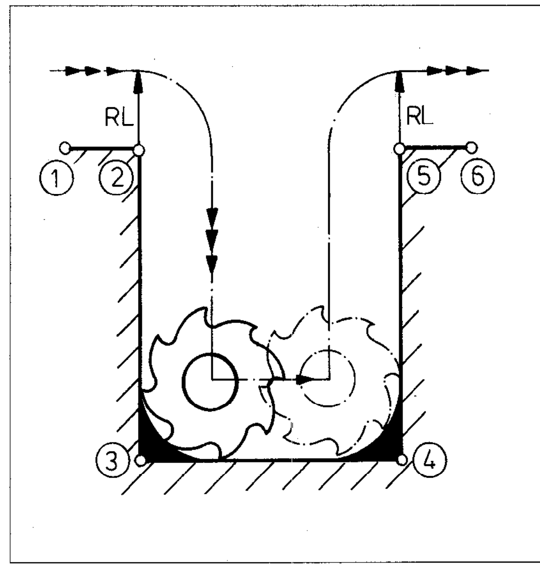


With M97

Predetermined M Functions
End of compensation: M98

Standard inside corner compensation

On inside corners in a continuously radius-compensated contour, the tool moves only to the intersection of the equidistants (see top figure). The work cannot be completely machined at positions ③ and ④.



M98

The middle figure shows two independent workpieces. Positions ③ and ④ are not connected. The tool must therefore be guided to positions ③' and ④'.

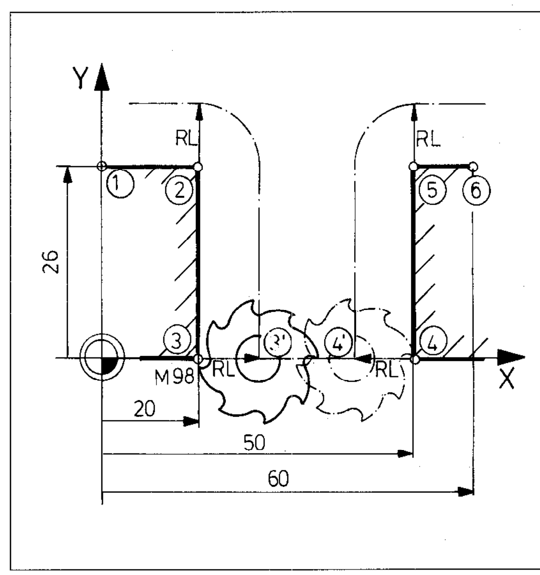
If you program a position with M98, the path offset remains valid until the end of this element and is ended there **for this block.**

No intersection is computed and **no transition arc is generated** for the end position, so the tool is always traversed perpendicular to the contour end point.

The previous compensation is reactivated automatically in the following block ④.

Position ④' is approached perpendicularly to ④.

The contour is thus completely machined at ③ and ④.

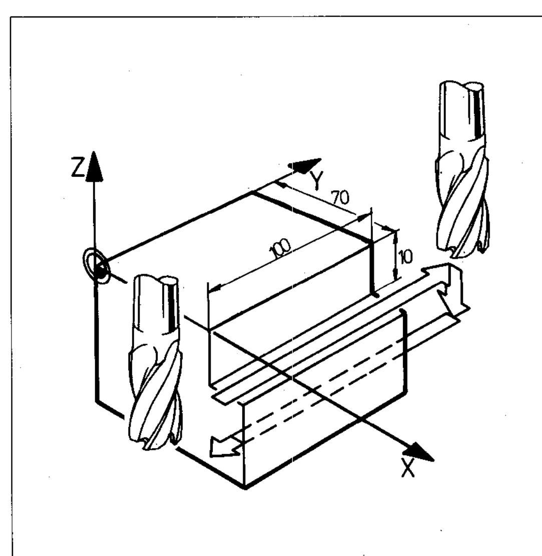


Example

L X+0 Y+26 RL F100	①
L X+20 Y+26	②
L X+20 Y+0 M98	③
L X+50 Y+0	④
L X+50 Y+26	⑤
L X60 Y26	⑥

Stepover milling with M98

Stepover milling with infeeds in Z.



Example

TOOL DEF 1 L+0 R5
TOOL CALL 1 Z S200

L X+70 Y–10 RR FMAX M3	Prepositioning
L Z–10 FMAX	Plunge
L Y+110 F200 **M98**	Stepover
L Z–20 FMAX	Second infeed
L Y+110 RL F200	Prepositioning
L Y–10 **M98**	Stepover

Predetermined M Functions
Machine-based coordinates: M91/M92

Coordinates programmed with M91 and M92 are independent of the manually set workpiece datum.

M91

Positions programmed with M91 are referenced to the datum of the linear or angle encoders. The datum is located at the negative end of the measuring range on linear encoders with distance-coded reference marks. On encoders with a single reference mark, the datum is set by this reference mark (the position of the reference mark is indicated by the **RM** sticker).

M92

When programming M92, nominal positions refer to the machine datum.

Applications

The miscellaneous functions M91 and M92 are used, for example, to

- approach fixed machine points, or
- approach the tool change position.

Displaying fixed machine coordinates

You may use the "MOD" key to display the coordinates referenced to the datum of the encoders (see index "General Information", MOD Functions).

Program Jumps
Overview

Jumping within a program

The following jumps can be made within a program:

- **Program section repeat**
- **Subprogram call**
- **Conditional jump**
- **Unconditional jump**

Nesting:
A program section repeat or a subprogram can also be called from within another program section repeat or subprogram.
(Maximum nesting depth: 8 levels)

Examples:

```
CALL LBL 4 REP 3/3

CALL LBL 7

IF Q5 GT0 GOTO LBL 12

IF 0 EQU 0 GOTO LBL 8
```

Jumping to another program

You can jump from a part program to any other program which is stored in the control.

Program a jump to another program with a

- **program call**

  or with

- **cycle 12: PGM CALL**

Nesting:
You can call further programs from a called program.
(Maximum nesting depth: 4 levels)

Examples:

```
CALL PGM 3

CYCL DEF PGM CALL PGM 3
.
.
.
L X+50 M99
```

Jumping Within a Program
Program markers (labels)

Labels

Labels (program markers) can be set during programming to mark the beginning of a subprogram or program section repeat.

These labels can be jumped to during program run (e.g. to execute the appropriate subprogram).

```
0 BEGIN PGM 1 MM
1 BLK FORM 0.1 Z X+0 Y+0 Z–40
2 BLK FORM 0.2 X+100 Y+100 Z+0
3 TOOL DEF 1 L+0 R+3
4 TOOL CALL 1 Z S 500
5 CYCL DEF 1.0 PECKING
6 CYCL DEF 1.1 SET UP –2
7 CYCL DEF 1.2 DEPTH –20
8 CYCL DEF 1.3 PECKG. –6
9 CYCL DEF 1.4 DWELL 0
10 CYCL DEF 1.5 F120
11 L Z+50 R0 FMAX M06
12 L X+10 Y+20 R FMAX M03
13 L Z+2 FMAX
14 CALL LBL 1 REP
15 L X+20 Y+50 FMAX
16 CALL LBL 1
17 L X+10 Y+80 FMAX
18 CALL LBL 1
19 L Z+50 R0 FMAX M02

20 LBL 1
21 CYCL CALL M
22 LBL 2
23 L IX+10 R FMAX M99
24 CALL LBL 2 REP 5 /5
25 LBL 0
26 END PGM 1 MM
```

Pecking cycle
Refer to "Fixed cycles" for explanation

Setting a label

LBL SET

A label is set with the "LBL SET" key. The label numbers 1 to 254 can be set only once in a program.

Label 0

Label number 0 always marks the end of a subprogram (cf. "Subprogram") and is therefore the return jump marker. It can thus occur more than once in a program.

Calling a label number

LBL CALL

The dialog is initiated with the "LBL CALL" key.

With LBL CALL you can:

- call subprograms
- create program section repeats.

Label numbers (1 to 254) can be called as often as desired.
Do not call label 0!

Program section repeats

For program section repeats, respond to the query REPEAT REP ? by entering the number of required repetitions.

Subprograms

For subprogram calls, respond to this query with the "NO ENT" key.

Conditional jumps

You can make the call of a program label be dependent on a mathematical condition (see Parametric Programming, Overview).

Error messages

JUMP TO LABEL 0 NOT PERMITTED
This jump (CALL LBL 0) is not allowed.

LABEL NUMBER ALLOCATED
Each label number – except LBL 0 – can be allocated (set) **only once** in a given program.

LBL CALL

Jumping Within a Program
Program section repeats

Program section repeats

An executed program section can be executed again immediately. This is called a program loop or program section repeat.

LBL SET

A label number marks the beginning of the program section which is to be repeated.

LBL CALL REP with number

The end of the program section to be repeated is designated by a call LBL CALL with the number of repetitions REP.

A program section can be repeated up to 65534 times.

```
.
.
.
22 LBL 2
23 L IX+10   FMAX M99
24 CALL LBL 2 REP 5 /5
.
.
.
```

Jump direction

A called program section repeat is always executed completely, i.e. until LBL CALL.
A program jump is therefore only meaningful if it is a return jump. In other words, the called label (LBL SET) must have a smaller block number than the calling block (LBL CALL).

Program run

The control executes the main program (along with the associated program section) until the label number is called. Then the return jump is carried out to the called program label and the program section is repeated.

The number of remaining repetitions on the display is reduced by 1: REP 2/1.

After another return jump, the program section is repeated a second time.

When all programmed repetitions have been performed (display: REP 2/0), the main program is resumed.

The total number of times a program section is executed is always one more than the programmed number of repeats.

```
.
.
.
22 LBL 2

23 L IX+10
   FMAX M99

24 CALL LBL 2
   REP 5 /5
.
.
.
```

Error message

EXCESSIVE SUBPROGRAMMING

A jump was programmed incorrectly:

1. No REP value was entered for a program section repeat.
   If no response is given to the query REP (by pressing the "NO ENT" key), the program section is treated like a subprogram without a correct ending (LBL 0): the label number is called 8 times. During program run or a test run, the error message appears on the screen after the 8th repetition.
2. The subprogram was programmed without LBL 0 for an intended subprogram call.

Jumping Within a Program
Program section repeats

Setting the program label

Example:

LBL SET 1 ENT

Program label 1 is set.

⋮

Repeating a program section after LBL

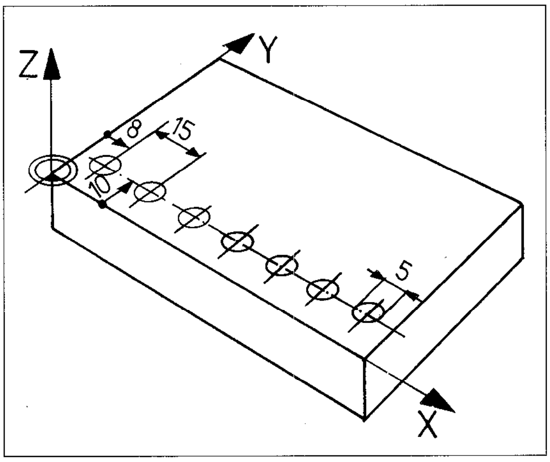
LBL CALL 1 ENT 6 ENT

6 repetitions after LBL 1.
The program section between LBL 1 and CALL LBL 1 is executed a total of 7 times.

Example bolt-hole row

The illustrated bolt-hole row with 7 identical bores is to be drilled with a program section repeat.

The tool is pilot positioned (offset to the left by the bore center distance) before starting the repeat to simplify programming.



Program

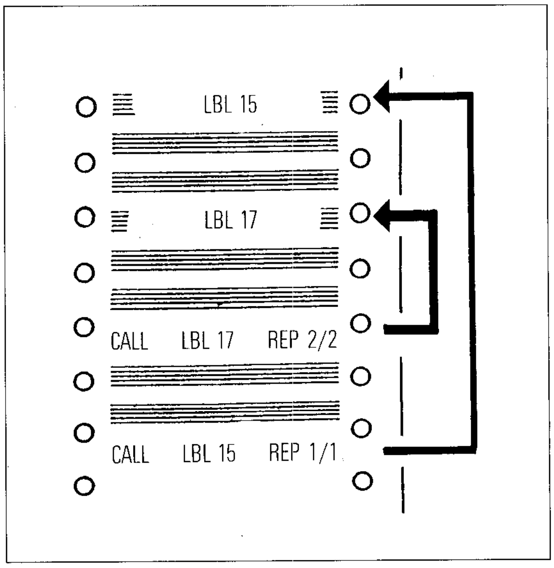
TOOL DEF 1 L+0 R2.5	Tool definition
TOOL CALL 1 Z S200	Tool call
L X–7 Y+10 Z+2 R0 FMAX M3	Pilot positioning
LBL 1	Start of the program section repeat
L IX+15 FMAX	Incremental distance between the bores, rapid traverse
L Z–10 F100	Absolute drilling depth, drilling feed rate
L Z+2 FMAX	Absolut retraction height, rapid traverse
CALL LBL 1 REP 6	Call for repeats

Nesting of repetitions

The main program is executed until the jump to LBL 17 (CALL LBL 17).
The program section between LBL 17 and CALL LBL 17 is repeated twice.

The control then resumes the main program run until the jump to LBL 15 (CALL LBL 15).

The program section until CALL LBL 17 REP 2/2 is repeated once and the nested program section also two more times. Then the program run is resumed.



Jumping Within a Program
Subprograms

Subprograms

If a program section occurs several times in the same program, it can be designated as a subprogram and called whenever required. This speeds up programming.

Start of subprogram

The start of subprogram is marked with a **label number** (can be any number).

End of subprogram

The end of the subprogram is always marked by **label 0.**

The different subprograms are then called in the main program as often as wanted and in any sequence.

```
.
.
.
14 CALL LBL 1
15 L X+20 Y+50
16 CALL LBL 1
17 L X+10 Y+80
18 CALL LBL 1
19 L Z+50 R0 FMAX M02

20 LBL 1
21 CYCL CALL M
22 LBL 2
23 L IX+10 R FMAX M99
24 CALL LBL 2 REP 5 /5
25 LBL 0

26 END PGM 1 MM
```

No repetitions Reply to REPEAT REP with NO ENT

When the subprogram is called with LBL CALL, the "NO ENT" key must be pressed after the dialog query REPEAT REP ? appears. A subprogram can be called at any point in the main program (but not from within the same subprogram).

Program run

The control executes the main program until the subprogram call ①.

A jump to the called program label ② is then performed.

Subprogram 1 is processed until label 0 (③) (end of subprogram).

Then the return jump to the main program follows. The main program is resumed with the block ④ following the subprogram call.

```
  .
  .
  .
① CALL LBL 1
④ L X ... Y ...
  .
  .
  .
  .
              M02

② LBL 1
  .
  .
  .
③ LBL 0
```

Subprograms should be placed after the main program (behind M2 or M30) for the sake of clarity. If a subprogram is placed within the main program, it is also executed once during program run without being called.

Error messages

If a subprogram call is programmed incorrectly (e.g. an end of subprogram lacks LBL 0, or a value for REPEAT REP ? was entered), the error message

EXCESSIVE SUBPROGRAMMING

appears.

Jumping Within a Program
Subprograms

Entry example: Subprogam 2

BEGIN PGM 1 MM

:

LBL CALL 2 ENT NO ENT — Subprogram 2 is called from within the main program. Conclude with "NO ENT"

.
.
.

L Z100 FMAX M2 — Retract and return jump to start

LBL 2 — Start of subprogram 2

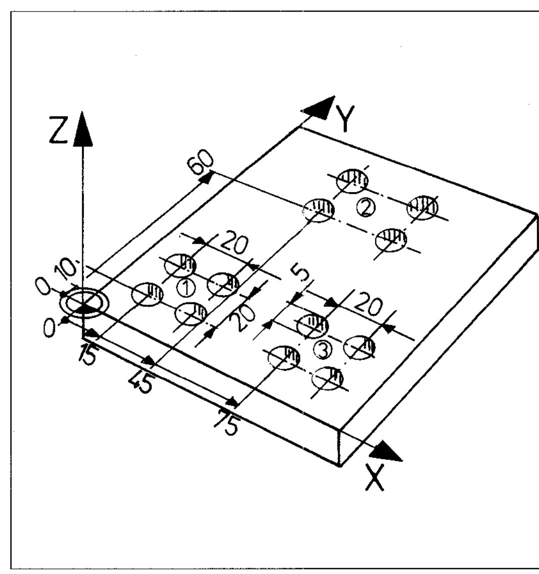
.
.
.

LBL 0 — End of subprogram 2

END PGM 1 MM — End of main program

Example

A group of four bores is to be programmed as subprogram 2 and executed at three different positions.



Program

TOOL DEF 1 L+0 R2.5
TOOL CALL 1 Z S200

CYCL DEF PECKING
SET UP –2
DEPTH –20
PECKG. –10
DWELL 0
FEEDRATE F100

L X+15 Y+10 R0 FMAX M3	Approach bore group ①
L Z+2 FMAX CALL LBL 2	 Subprogram call
L X+45 Y+60 FMAX CALL LBL 2	Approach bore group ② Subprogram call
L X+75 Y+10 FMAX CALL LBL 2	Approach bore group ③ Subprogram call
L Z+50 FMAX M2	Retract tool axis
LBL 2	Start of subprogram
CYCL CALL	Call peck drilling cycle
L IX+20 FMAX M99	Incremental traverse, drill
L IY+20 FMAX M99	Incremental traverse, drill
L IX–20 FMAX M99	Incremental traverse, drill
LBL 0	End of subprogram

M99 = blockwise cycle call

Cross-reference

You will find an explanation of the peck drilling cycle in the section "Fixed cycles".

Jumping Within a Program
Nesting subprograms

Nesting subprograms

The main program is executed until the jump command CALL LBL 17 is reached.

The subprogram beginning with LBL 17 is subsequently executed until the next CALL LBL 20, which is then run until CALL LBL 53.

The lowest nested subprogram 53 is run through until its LBL 0.

At the end (LBL 0) of the last subprogram (53), return jumps are made to the preceding subprograms (20 and 17), until the main program is finally reached.

The main program is then taken up again at the point immediately following CALL LBL 17.

A subprogram call is considered executed when the first LBL 0 is reached!

```
BEGIN PGM 12 MM
:
CALL LBL 17
:
M2

LBL 17
:
CALL LBL 20
:
LBL 0

LBL 20
:
CALL LBL 53
:
LBL 0

LBL 53
:
LBL 0

END PGM 12 MM
```

Repeating subprograms

You can execute subprograms repeatedly with the nesting technique:

Subprogram 50 is called in a program section repeat. This subprogram call is the only block in the program section repeat.

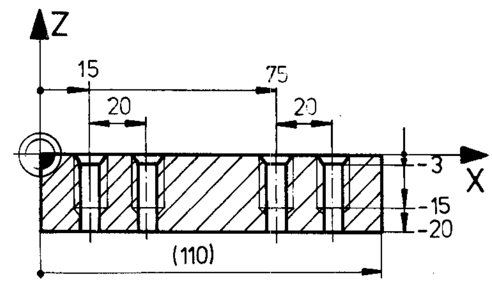
Remember: the subprogram will be executed one more time than the programmed number of repeats.

```
.
.
.
LBL 5
CALL LBL 50
CALL LBL 5 REP 9
.
.
.
M2

LBL 50
.
.
LBL 0
```

Jumping Within a Program
Example: Hole pattern with several tools

Task Based on the example "group of four holes at three different positions" (see the section "Subprograms" on the previous pages), three different tools and drilling operations will be added to the program.

Note You will find an explanation of the pecking and tapping cycles in the section "Fixed cycles".




```
0 BEGIN PGM 183 MM
1 BLK FORM 0.1 Z X+0 Y+0 Z–20
2 BLK FORM 0.2 X+110 Y+100 Z+0
3 TOOL DEF 25 L+0 R+2.5
4 TOOL DEF 30 L+0 R+3
5 TOOL DEF 35 L+0 R+3.5
```

Countersink

```
6 CYCL DEF 1.0 PECKING
7 CYCL DEF 1.1 SET UP –2
8 CYCL DEF 1.2 DEPTH –3
9 CYCL DEF 1.3 PECKG –3
10 CYCL DEF 1.4 DWELL 0
11 CYCL DEF 1.5 F100
12 TOOL CALL 35 Z S 500
13 CALL LBL 1                          Call subprogram 1
```

Pecking

```
14 CYCL DEF 1.0 PECKING
15 CYCL DEF 1.1 SET UP –2
16 CYCL DEF 1.2 DEPTH –25
17 CYCL DEF 1.3 PECKG –6
18 CYCL DEF 1.4 DWELL 0
19 CYCL DEF 1.5 F50
20 TOOL CALL 25 Z S 1000
21 CALL LBL 1
```

Tapping

```
22 CYCL DEF 2.0 TAPPING
23 CYCL DEF 2.1 SET UP –2
24 CYCL DEF 2.2 DEPTH –15
25 CYCL DEF 2.3 DWELL 0
26 CYCL DEF 2.4 F100
27 TOOL CALL 30 Z S 250
28 CALL LBL 1
29 L Z+50    R0 FMAX M2               Retract spindle axis, jump to start of program
```

Subprogram 1

```
30 LBL 1
31 L X+15 Y+10 R0 FMAX M3             Approach bore group ①
32 L Z+2  FMAX                        Traverse to safety clearance
33 CALL LBL 2                         Call subprogram 2
34 L X+45 Y+60   FMAX                 Approach bore group ②
35 CALL LBL 2
36 L X+75 Y+10   FMAX                 Approach bore group ③
37 CALL LBL 2
38 LBL 0
```

Subprogram 2

```
39 LBL 2
40 CYCL CALL                          Cycle call (countersink, pecking, tapping)
41 L IX+20          M99
42 L IY+20          M99
43 L IX–20          M99
44 LBL 0
45 END PGM 183 MM                     M99 = blockwise cycle call
```

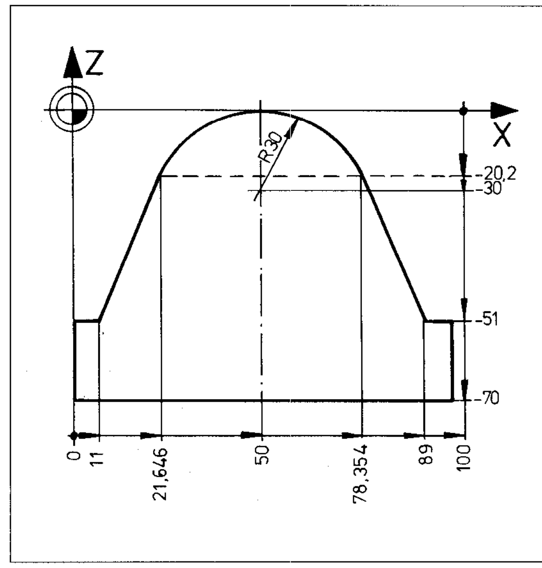
Jumping Within a Program
Example: Horizontal geometric form

Task

The adjacent geometric contour is to be machined from a cuboid with an end mill which is to be continuously advanced in the Y direction by a program section repeat.

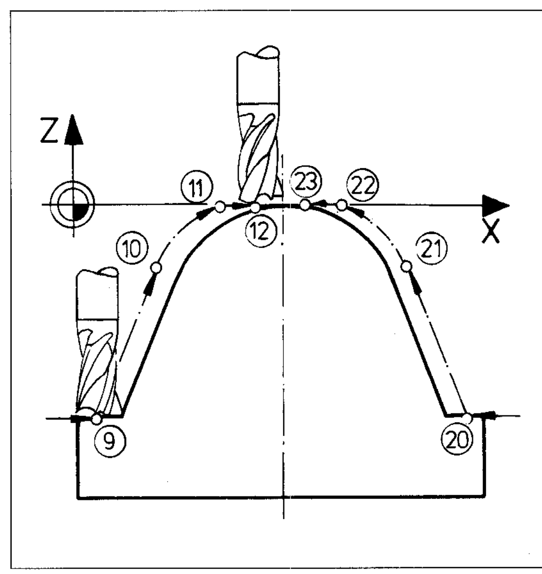
The contour is divided into two halves along the line of symmetry to simplify the working procedure. The contour is to be machined upwards.

In addition to the adjacent dimensions, the cuboid length is specified with:
Y = 100 mm.



Program procedure

The adjacent figure schematically shows the cutter center path and the associated program blocks. The entire contour is divided into a "left" and "right" half and is machined in the two program section repeats LBL 1 and LBL 2. The program runs **without radius compensation,** i.e. the cutter center path is programmed. To obtain the desired contour, the tool radius must be **subtracted** on the **left** side and **added** on the **right** side (all X coordinates).




```
0 BEGIN PGM 90007685 MM
1 BLK FORM 0.1 Z X+0 Y+0 Z–70
2 BLK FORM 0.2 X+100 Y+100 Z+0
3 TOOL DEF 1 L+0 R+10
4 TOOL CALL 1 Z S1000
5 L X–20 Y–1 FMAX M3
```

Approach starting point for "left side"

Program section repeat 1

```
6 LBL1
7 L Z–51
8 L X+1 F100
9 L X+11,646 Z–20.2
10 CT X+40 Z+0
11 L X+41
12 L Z+10 FMAX
13 L X–20 IY+2.5
14 CALL LBL1 REP40/40
```

Infeed in Y axis
Program section is executed 41 times

```
15 L Z+20 FMAX
16 L X+120 Y–1
```

Retract spindle axis
Approach starting point for "right side"

Program section repeat 2

```
17 LBL2
18 L Z–51
19 L X+99 F100
20 L X+88.354 Z–20.2
21 CT X+60 Z+0
22 L X+59
23 L Z+10 FMAX
24 L X+120 IY+2.5
25 CALL LBL2 REP40/40
```

Infeed in Y axis
Program section is executed 41 times

```
26 L Z+20 FMAX M2
27 END PGM 90007685 MM
```

Retract spindle axis, jump to start of program

Program Calls

Jumping to another main program

You can call another program which is stored in the control from any machining program.
This allows you to create your own fixed cycles with parametric programming.
Program the call with a "PGM CALL" key.

Calling criteria

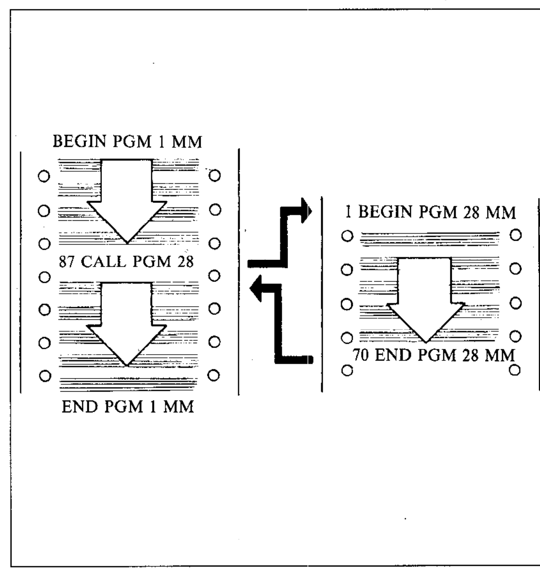
The program to be called cannot contain M02 or M30. In the called program, do not program a jump back to the original program (creates an endless loop). Only one BLK FORM can exist.

Process

The control executes main program 1 until CALL PGM 28. Then a jump is made to main program 28.

Main program 28 is executed from beginning to end.

Then a return jump is made to main program 1. Main program 1 is resumed with the block following the program call.



Example 1

PGM CALL

PGM CALL 10 ENT

CALL PGM 10	Call with a separate program line

Example 2

CYCL DEF

The program to be called can also be specified with a cycle definition. The call then functions like a fixed cycle.

CYCL DEF GO TO 1 2 ENT

CYCL DEF 12.0 PGM CALL CYCL DEF 12.1 PGM 20	Call e.g. via M99 (see Other Cycles, Cycle 12)

CYCL DEF CYCL CALL

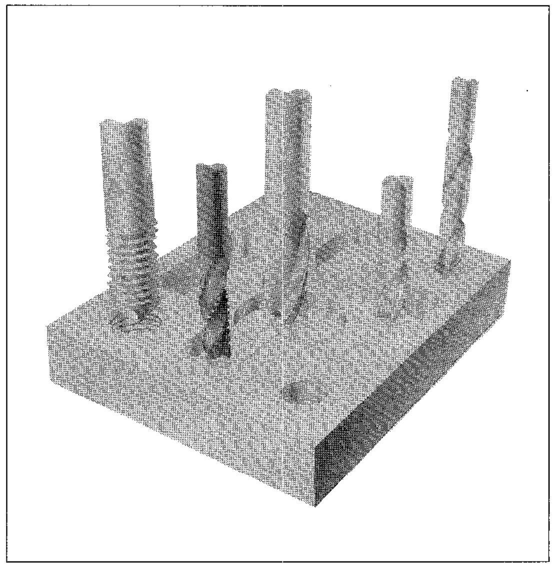
Standard Cycles
Introduction, Overview

Standard cycles

To facilitate programming, frequently recurring machining sequences (drilling and milling jobs) and certain coordinate transformations are pre-programmed as standard cycles.

Machine builder cycles

The machine manufacturer can also store his own programs as cycles in the control. These cycles can be called under the cycle numbers 68 to 99. Contact the machine manufacturer for more information.



Selecting a cycle

CYCL DEF

After pressing the "CYCLE DEF" key, data for the cycles shown to the right can be entered and also any programmed user cycles can be selected. The desired cycle can be selected with the vertical cursor keys or with "GOTO □".

Defining a cycle

The cycle definitions can be entered in the dialog after pressing "ENT".

Fixed cycles

No.	Cycle	Effective after call	Effective immediately
1	Pecking	●	
2	Tapping	●	
17	Rigid tapping	●	
3	Slot milling	●	
4	Pocket milling	●	
5	Circular pocket	●	
12	Program call	●	
14	Contour geometry		●
15	Pilot drilling	●	
6	Rough-out	●	
16	Contour milling	●	

Calling a fixed cycle

Cycles must be called after moving the tool to the appropriate position – only then will the last defined cycle be executed.

Coordinate transformations

No.	Cycle	Effective after call	Effective immediately
7	Datum shift		●
8	Mirror image		●
10	Rotating the coordinate system		●
11	Scaling		●
9	Dwell time		●

There are three ways to call a cycle:

CYCL CALL

- With a separate CYCL CALL block

M99

- Via the miscellaneous function M99. "CYCL CALL" and M99 are only effective blockwise and must therefore be reprogrammed for every execution.

M89

- Via the miscellaneous function M89 (depending on machine parameters).

M89 is effective modally, i.e. the last programmed cycle is called at every subsequent positioning block. M89 is cancelled or cleared by M99 or by CYCL CALL.

Coordinate transformations

Coordinate transformations and the dwell time are effective immediately and remain effective until changed.

Fixed Cycles
Preparatory measures

Prerequisites

The following must be programmed before a cycle call (e.g. M99).

- Tool call: to specify the spindle axis and the spindle speed
- Positioning block to the starting position in the plane with miscellaneous function for specifying the rotating direction of the spindle
- Cycle definition (CYCL DEF)
- Positioning the spindle axis

TOOL CALL

L X . . . Y . . . M3

CYCL DEF

L Z . . .   M99

Dimensions

In the cycle definition, dimensions for the tool axes are to be entered incrementally, referenced to the tool position at cycle call.

The "incremental" key does not have to be pressed!
All infeeds must have the same sign (usually negative).

Selecting a cycle

Initiate the dialog

CYCL DEF 1 PECKING	Search for the cycle by name: Select the desired cycle Confirm cycle.

or

CYCL DEF 1 PECKING	Select the cycle by its number: Enter the cycle number with "GOTO □" Confirm cycle.

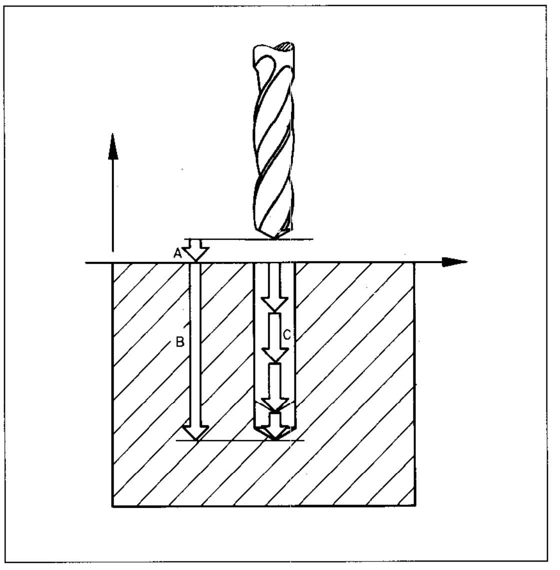
Entering values

Enter all values as requested and confirm entry with "ENT".
You must respond to every dialog query by entering a value!

Fixed Cycles
Cycle 1: Pecking

The cycle

A hole is drilled with multiple infeeds, each followed by a complete retraction.



Input data

Infeed value signs:

- – for negative working direction
- + for positive working direction

All infeeds must have the same sign.

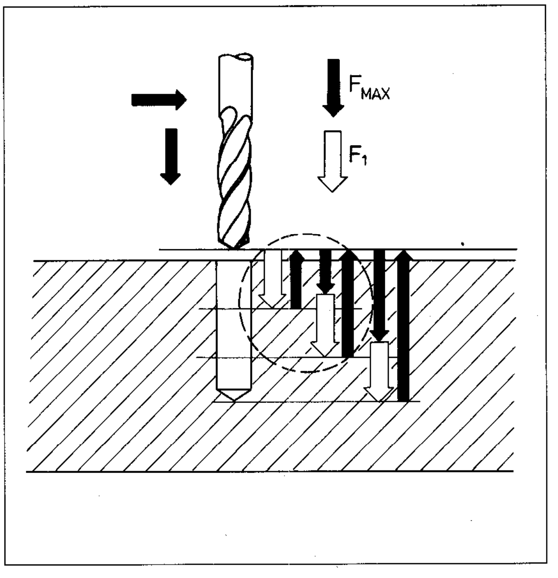
Setup clearance A: distance between tool tip (starting position) and workpiece surface.

Total hole depth B: distance between the workpiece surface and the bottom of the hole (tip of the drill taper).

Pecking depth C: the infeed per cut.

Dwell time: the time the tool remains at the bottom of the bore hole for chip breaking.

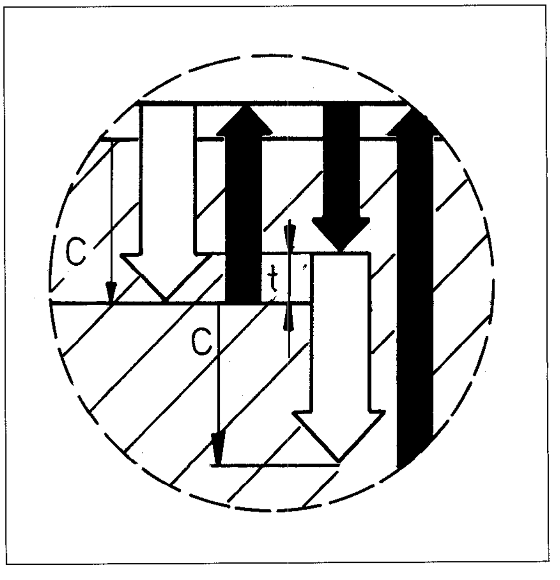
Feed rate F: traversing speed of the tool during infeed in mm per minute.



Process

- The tool must be positioned to the setup clearance with a **separate block,** before the cycle call.
- The tool drills from the starting position to the first pecking depth at the programmed feed rate.
- After reaching the first pecking depth the tool is retracted in rapid traverse FMAX to the starting position and advanced again to the first pecking depth, minus the advanced stop distance t.
- The tool then advances by another infeed at the programmed feed rate, returns again to the starting position etc.

- Drilling and retraction are performed alternately until the programed total hole depth is reached.
- After the dwell time at the hole bottom, the tool is retracted to the starting position in rapid traverse.



Advanced stop distance

The advanced stop distance t is automatically computed by the control:

- For a total hole depth up to 30 mm: t = 0.6 mm;
- For a total hole depth over 30 mm: t = total hole depth/50, whereby the maximum advanced stop distance is limited to $t_{max} = 7$ mm.

Fixed Cycles
Cycle 1: Pecking

Defining the cycle

Operating mode

Initiate the dialog

CYCL DEF 1 PECKING	ENT Select the cycle.
SET UP CLEARANCE ?	Specify setup clearance +/- Enter the sign correctly (normally positive) ENT Confirm entry
TOTAL HOLE DEPTH ?	Specify hole depth +/- Enter the sign correctly (normally negative) ENT Confirm entry
PECKING DEPTH ?	Specify pecking depth +/- Enter the sign correctly (normally negative) ENT Confirm entry
DWELL TIME IN SECS. ?	Enter the dwell time at the bottom of the hole (zero for no dwell time) ENT Confirm entry
FEED RATE ? F =	Enter the feed rate for pecking ENT Confirm entry

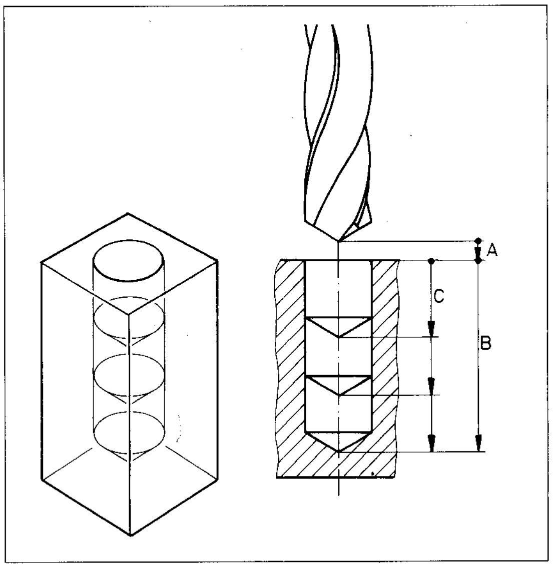
The signs for setup clearance, total hole depth and pecking depth are all the same (normally negative)!

Fixed Cycles
Cycle 1: Pecking

Remarks

- The total hole depth can be programmed equal to the pecking depth. The tool then traverses in **one** work step to the programmed depth (e.g. for centering).
- The pecking depth need not be a multiple of the total depth. In the last work step, the tool will only be advanced the remaining distance to the programmed hole depth.
- If the specified pecking depth is greater than the total hole depth, drilling is only performed to the programmed total hole depth.

The above also applies to other fixed cycles.



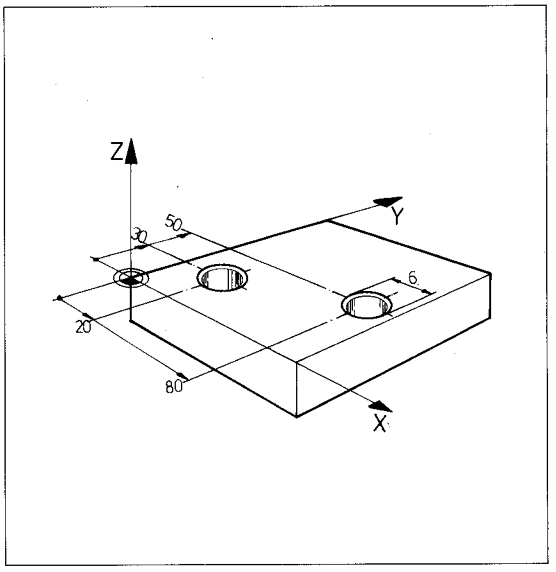
Example

Drill 2 holes with the standard pecking cycle.

TOOL DEF 1 L+0 R3	Tool definition
TOOL CALL Z S200	and call

The definition occupies 6 program blocks

CYCL DEF 1.0 PECKING	
CYCL DEF 1.1 SET UP –2	Setup clearance
CYCL DEF 1.2 DEPTH –20	Total depth
CYCL DEF 1.3 PECKG –10	Pecking depth
CYCL DEF 1.4 DWELL 2	Dwell time
CYCL DEF 1.5 F 80	Feed rate



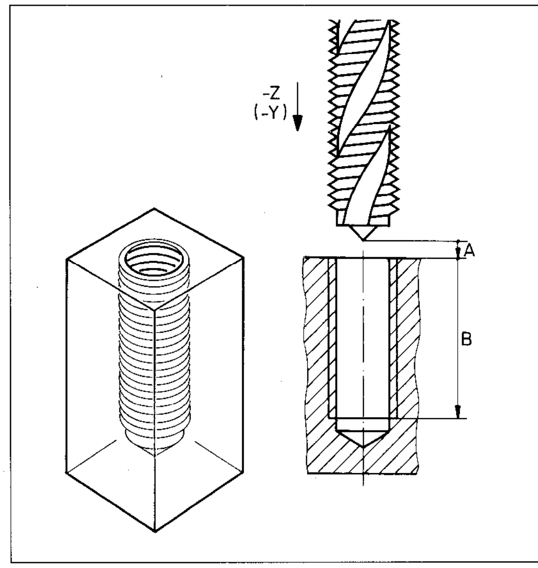
L X+20 Y+30 R0 FMAX M3	Pilot positioning, spindle on
L Z+2 FMAX M99	1st hole, cycle call
L X+80 Y+50 FMAX M99	2nd hole, cycle call

Fixed Cycles
Cycle 2: Tapping with floating tap holder

The cycle

The thread is cut in **one** operation.
A **floating tap holder** is required for tapping. It must compensate for the tolerances between the feed rate, speed and the tool geometry as well as spindle run out after the position is reached.

Spindle speed override is inactive after a cycle call; the feed rate override is only active over a limited range (set by the machine manufacturer via machine parameters).



Input data

Setup clearance A: distance between tool tip (starting position) and workpiece surface (standard value: approx. 4 x thread pitch). The preceding sign depends on the working direction.

Total hole depth B (= thread length): distance between workpiece surface and end of thread. The signs for setup clearance and total hole depth are the same (usually negative).

Dwell time: enter either the time between reversing the direction of spindle rotation and retracting the tool, or 0. This time is machine-dependent.

Feed rate/ thread pitch

Feed rate F: traversing speed of the tool during tapping.

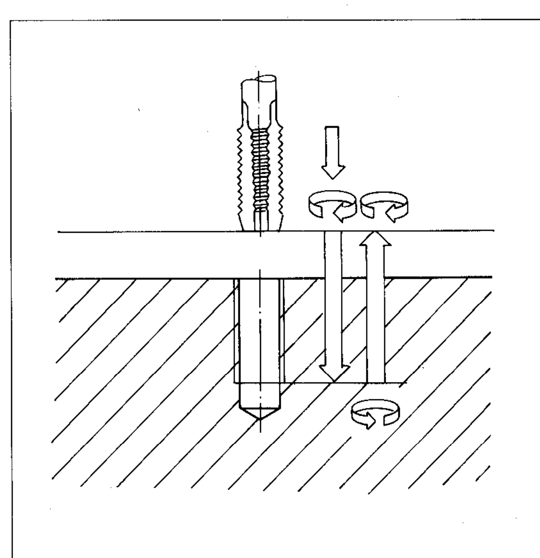
Determining the required feed rate:
F = S x P
F: feed rate
S: spindle speed
P: thread pitch

The thread pitch is determined indirectly by the spindle speed specified in the tool call and the feed rate of the cycle (see "Cutting data").

Process

Once the tool has reached the total hole depth, the direction of spindle rotation is reversed within a time period set by machine parameters.

At the end of the programmed dwell time, the tool is retracted to the starting position. The spindle direction is reversed again in the retracted position.



Input

Analogous to "Pecking".

Example

Tap an M6 hole with 0.75 mm pitch at 100 rpm.

TOOL DEF 1 L+0 R3 TOOL CALL 1 Z S100	Tool definition and call

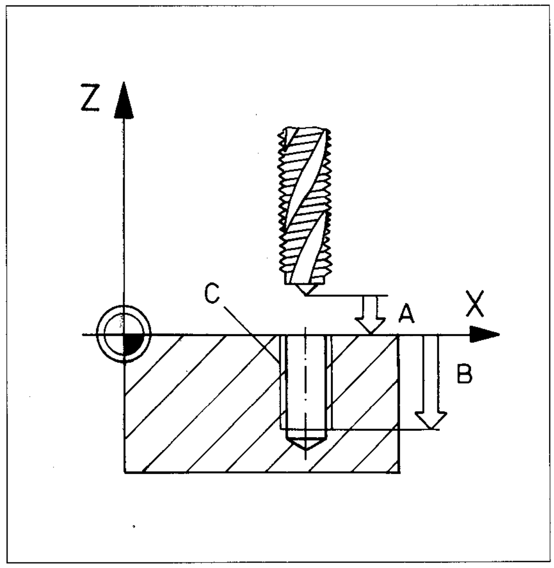
The definition occupies 5 program blocks.

CYCL DEF 2.0 TAPPING	
SET UP –3	Setup clearance
DEPTH –20	Thread depth
DWELL 0.4	Dwell time
F 75	Feed rate
L X+50 Y+20 R0 FMAX M3	Pilot positioning, spindle right
L Z+3 FMAX **M99**	Cycle call

Fixed Cycles
Cycle 17: Rigid tapping

The cycle

Threads are cut without a floating tap holder in one or – if necessary – several passes (example: removing chips in a blind hole).



Rigid tapping offers the following advantages over tapping using a floating tap holder:

- Higher machining speeds possible
- Repeated tapping of the same thread; repetitions are made possible by means of a spindle orientation referenced to the 0° position during cycle call
- Simpler input data
- Increased usable traverse range of the spindle axis

Input data

Setup clearance A: distance between tool tip (starting position) and workpiece surface. Preceding sign depends on the working direction.

Thread depth B: distance between workpiece surface and end of thread. The signs for setup clearance and the thread depth are always the same (usually negative).

Pitch C: thread pitch. The preceding sign differentiates between right-hand and left-hand threads:
pitch + = right-hand thread
pitch – = left-hand thread

Feed rate

The control calculates the feed rate F from the active spindle speed S and the pitch P (F = S x P). If spindle speed override is activated during tapping, the feed rate is automatically adjusted.

Input

Similar to the pecking cycle (cycle 1).

Example

Producing a right-hand M6 thread with 0.75 mm pitch at 100 rpm:

TOOL DEF 1 L+0 R3	Tool definition
TOOL CALL 1 Z S100	Tool call

The definition occupies 4 program blocks

CYCL DEF 17.0 RIGID TAPPING	
CYCL DEF 17.1 SET UP –10	Setup clearance
CYCL DEF 17.2 DEPTH –30	Thread depth
CYCL DEF 17.3 PITCH +0.75	Thread pitch

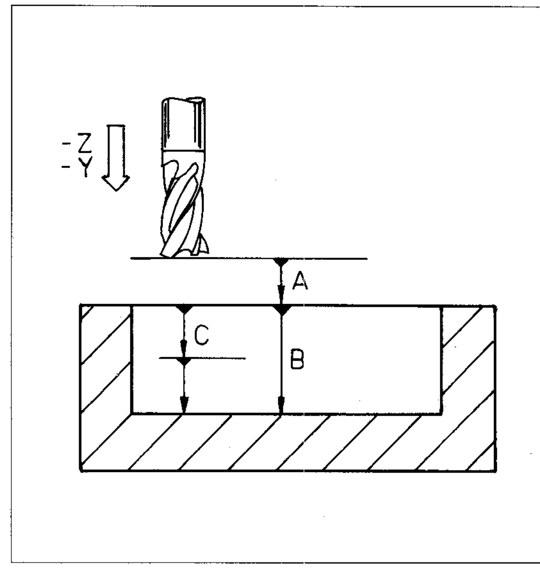
The control and the machine must be specially prepared by the machine manufacturer to enable rigid tapping.

Fixed Cycles
Cycle 3: Slot milling

The cycle

The slot milling cycle is a combined roughing/ finishing cycle.

The slot is parallel to one axis of the current coordinate system (rotation with cycle 10, if desired).



Tool required

The cycle requires a center-cut end mill. The cutter diameter must be slightly smaller than the slot width.

Input data

Setup clearance A: distance between tool tip (starting position) and workpiece surface.

Milling depth B: (= slot depth): distance between work surface and bottom of slot.

Pecking depth C: penetrating distance of the tool into the workpiece.

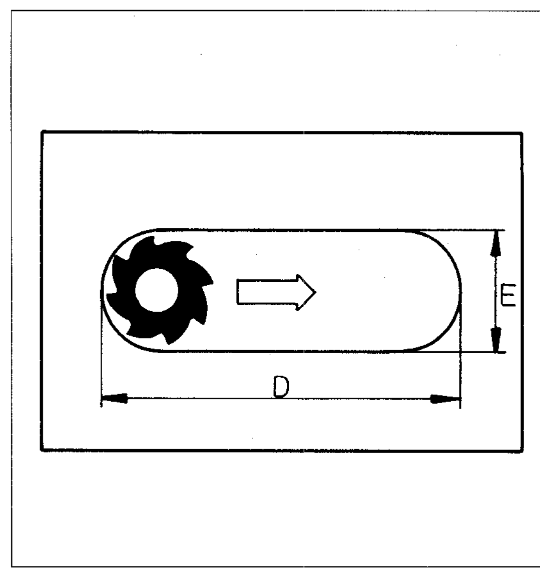
The signs for setup clearance, milling depth and pecking depth are all the same (usually negative).

Feed rate for pecking: traversing speed of the tool during penetration.

1st side length D: slot length (finished size). Sign depends on the first direction of cut parallel to the longitudinal axis of the slot.

2nd side length E: slot width, maximum 4 times the tool radius (finished size).

Feed rate: traversing speed of the tool in the machining plane.



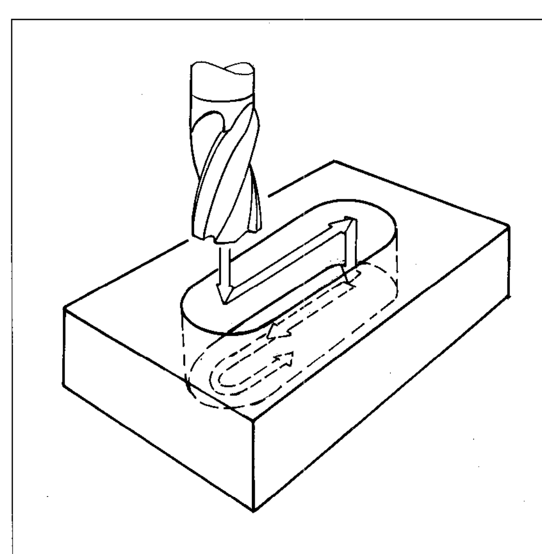
Roughing process

- The tool penetrates the work from the starting position.
- Subsequent milling is in the longitudinal direction of the slot. After downfeed at the end of the slot, milling is in the opposite direction.
- The procedure is repeated until the programmed milling depth is reached.

Finishing process

The control advances the tool in a semicircle at the bottom of the slot by the remaining finishing depth and down-cut mills the contour (with M3). The tool is subsequently retracted in rapid traverse to the setup clearance.

If the number of infeeds was odd, the cutter returns along the slot at the setup clearance to the starting position in the main plane.

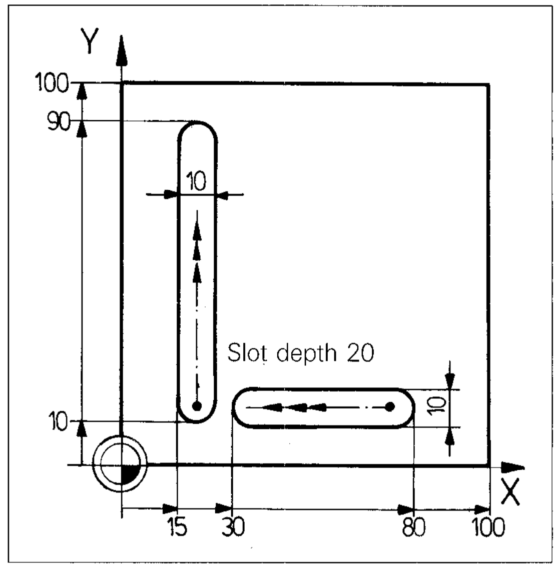


CYCL DEF CYCL CALL

Fixed Cycles
Cycle 3: Slot milling

Example

A horizontal slot with length 50 mm and width 10 mm as well as a vertical slot with length 80 mm and width 10 mm are to be milled.




```
0 BEGIN PGM SLOTS MM

1 BLK FORM 0.1 Z X+0 Y+0 Z–40
2 BLK FORM 0.2 X+100 Y+100 Z+0

3 TOOL DEF 1 L+0 R+4
4 TOOL CALL 1 Z S1800

5 L Z+50 R0 FMAX
```

	Program	
Cycle definition	6 CYCL DEF 3.0 SLOT MILLING	Definition of the **horizontal** slot
	7 CYCL DEF 3.1 SET UP –2	Setup clearance
	8 CYCL DEF 3.2 DEPTH –20	Milling depth
	9 CYCL DEF 3.3 PECKG –5 F80	Pecking depth, feed rate for pecking
	10 CYCL DEF 3.4 X–50	Length of slot and first **milling direction (–)**
	11 CYCL DEF 3.5 Y+10	Slot width
	12 CYCL DEF 3.6 F120	Feed rate
Starting position	13 L X+76 Y+15 R0 FMAX M3	Approach starting position **without compensation,** taking the tool radius into account in the longitudinal direction of the slot; spindle on
	14 L Z+2 R0 F1000 M99	Pilot positioning in Z, cycle call
	15 CYCL DEF 3.0 SLOT MILLING	Definition of the **vertical** slot
	16 CYCL DEF 3.1 SET UP –2	Setup clearance
	17 CYCL DEF 3.2 DEPTH –20	Milling depth
	18 CYCL DEF 3.3 PECKG –5 F80	Pecking depth, feed rate for pecking
	19 CYCL DEF 3.4 Y+80	Slot length and first **milling direction (+)**
	20 CYCL DEF 3.5 X+10	Slot width
	21 CYCL DEF 3.6 F120	Feed rate
	22 L X+20 Y+14 R0 FMAX M99	Approach starting position, cycle call
	23 L Z+50 R0 FMAX M2	Retract in tool axis
	24 END PGM SLOTS MM	End of program

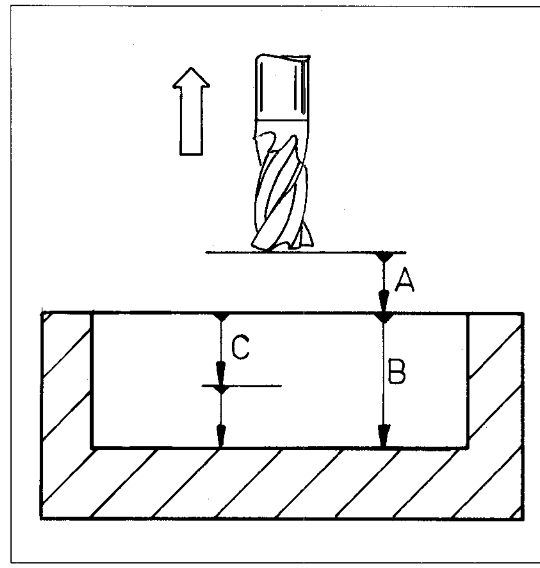
Fixed Cycles
Cycle 4: Rectangular pocket milling

The cycle
The rectangular pocket milling cycle is a roughing cycle.

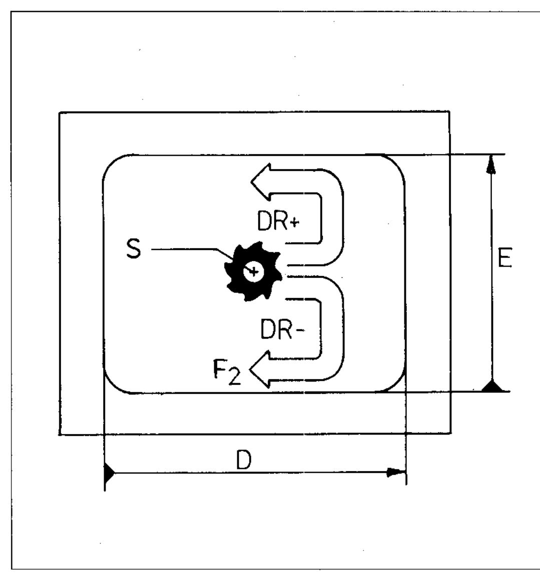
Tool required
The cycle requires a center-cut end mill, or pilot drilling at the pocket center.

> The tool determines the radius at the pocket corners. There is no circular movement in the pocket corners.

Position
The pocket sides are parallel to the coordinate system axis; the coordinate system may have to be rotated (see Cycle 10: Rotating the coordinate system).



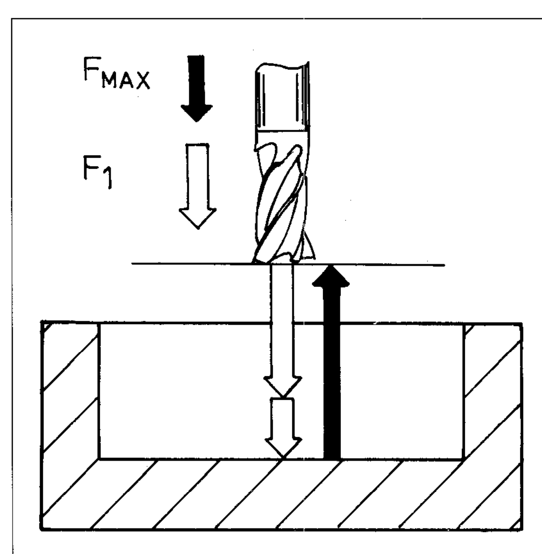
Input data
Setup clearance A: distance between tool tip (starting position) and workpiece surface.
Milling depth B (= pocket depth): distance between workpiece surface and bottom of pocket.
Pecking depth C: amount by which the tool penetrates the workpiece.
The signs for setup clearance, milling depth and pecking depth are all the same (usually negative).
Feed rate for pecking F_1: traversing speed of the tool at penetration.
1st side length D: pocket length parallel to the first main axis of the machining plane. The sign is always positive.
2nd side length E: pocket width; the sign is always positive.
Feed rate F_2: traversing speed of the tool in the machining plane.



Direction of the milling path:

Climb milling (down cut)
DR+: counterclockwise, down-cut milling with M3

Conventional milling (up cut)
DR–: clockwise, up-cut milling with M3



Starting position
The starting position S (pocket center) must be approached without radius compensation in a preceding positioning block.

Process
- The tool penetrates the work from the starting position (pocket center).
- The cutter then follows the programmed path at feed rate F_2.

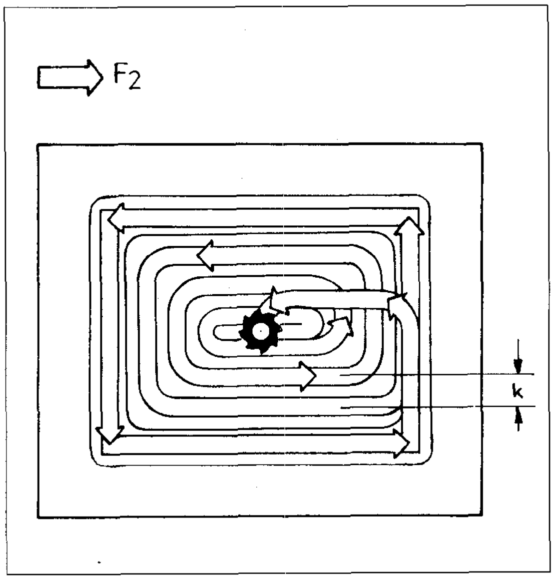
The starting direction of the cutter is the positive axis direction of the longer side, i.e. if this longer side is parallel to the X axis, the cutter starts in the positive X direction.
The cutter always starts in the positive Y direction on square pockets.

Fixed Cycles
Cycle 4: Rectangular pocket milling

Process

The milling direction depends on the programming (here, DR+). The maximum stepover is k.

The process is repeated until the programmed milling depth is reached.
On completion, the tool is withdrawn to the starting position.

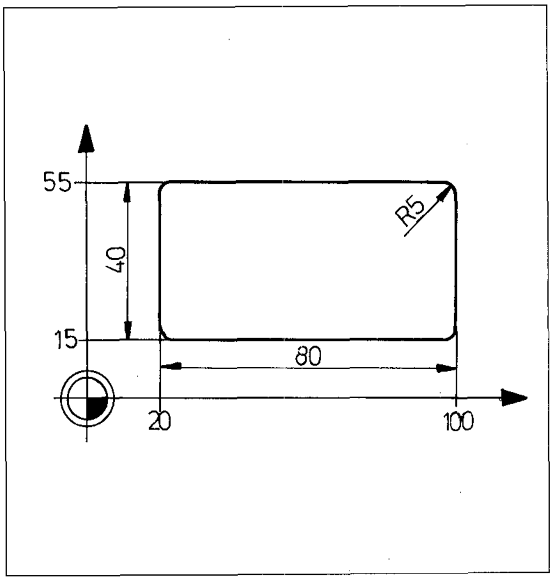


Stepover

Stepover k is computed by the control according to the following formula:

k = F x R

k: stepover
F: the overlap factor specified by the machine manufacturer
(depends upon a machine parameter, see index General Information, MOD Functions, User parameters)
R: cutter radius



Example

TOOL DEF 1 L+0 R5	
TOOL CALL 1 Z S200	
CYCL DEF 4.0 POCKET MILLING	The definition occupies 7 program blocks
CYCL DEF 4.1 SET UP –2	Setup clearance
CYCL DEF 4.2 DEPTH –30	Milling depth
CYCL DEF 4.3 PECKG –10	Pecking depth
F 80	Feed rate for pecking
CYCL DEF 4.4 X+80	1st side length of the pocket
CYCL DEF 4.5 Y+40	2nd side length of the pocket
CYCL DEF 4.6 F 100 DR+	Feed rate and rotating direction of the cutter path
L X+60 Y+35 R0 FMAX M3	Pilot positioning in X, Y, spindle on
L Z+2 FMAX M99	Pilot positioning in Z, cycle call

Fixed Cycles
Cycle 5: Circular pocket milling

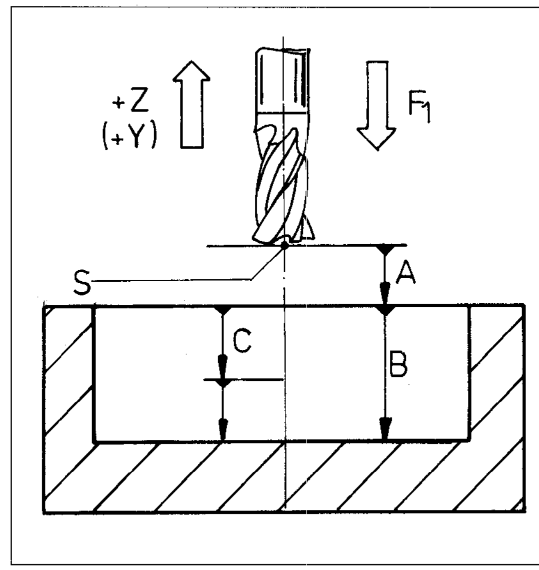
The cycle

The circular pocket milling cycle is a roughing cycle.

Tool required

The cycle requires a center-cut end mill or pilot drilling at the pocket center S.

Input data



Setup clearance A: distance between tool tip (starting position) and workpiece surface.

Milling depth B (= pocket depth): distance between workpiece surface and bottom of pocket.

Pecking depth C: amount by which the tool penetrates the workpiece.

The signs for setup clearance, milling depth and pecking depth are all the same (usually negative).

Feed rate for pecking F_1: traversing speed of the tool at penetration.

Circle radius R: radius of the circular pocket.

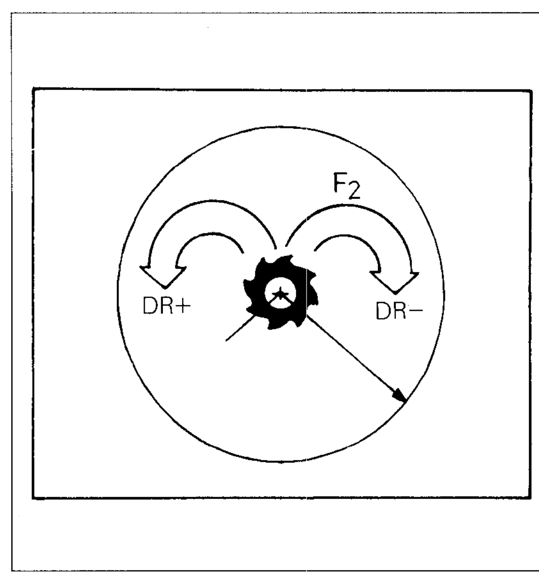
Feed rate F_2: traversing speed of the tool in the machining plane.

Direction of the milling path:
Climb milling (down cut)

DR+: counterclockwise, down-cut milling with M3

Conventional milling (up cut)

DR–: clockwise, up-cut milling with M3

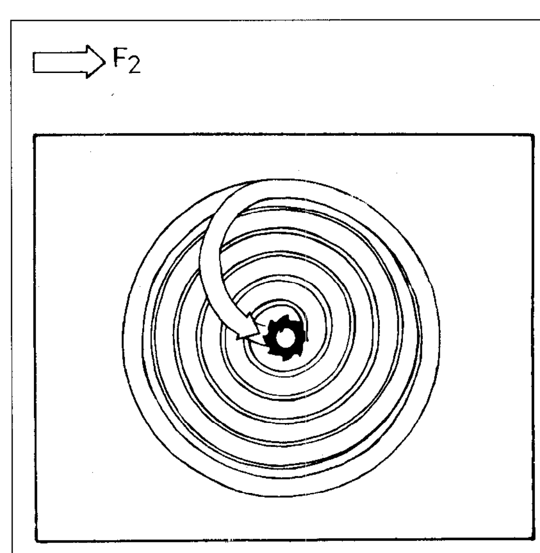


Starting position

The starting position S (pocket center) must be approached without radius compensation in a preceding positioning block.

Process

- The tool penetrates the work from the starting position (pocket center) at the "feed rate for pecking".
- The cutter then follows the programmed spiral path at feed rate F_2. The direction of the path depends upon the programming (here, DR+).



The starting direction of the cutter is for the

- XY plane the Y+ direction,
- ZX plane the X+ direction,
- YZ plane the Z+ direction.

The maximum stepover is the value k (see Rectangular Pocket Milling cycle).

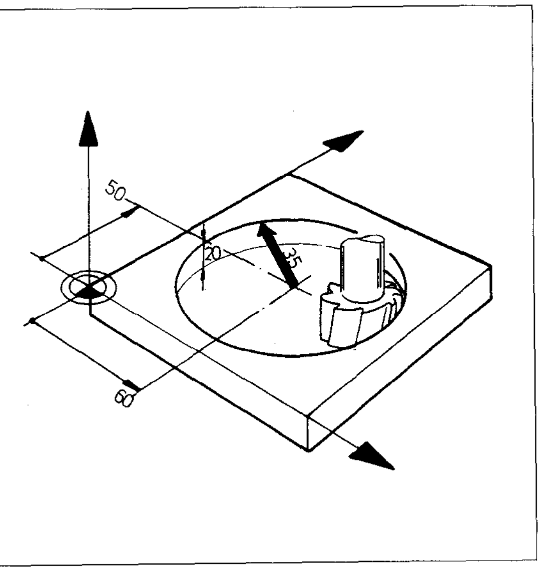
The process is repeated until the programmed milling depth is reached.

When milling is completed, the tool is withdrawn to the starting position.

Fixed Cycles
Cycle 5: Circular pocket milling

Example

A circular pocket with radius 35 mm and depth 20 mm is to be milled at position X+60 Y+50.




```
TOOL DEF 1 L+0 R10
TOOL CALL 1 Z S200
```

CYCL DEF 5.0 CIRCULAR POCKET	
CYCL DEF 5.1 SET UP −2	Setup clearance
CYCL DEF 5.2 DEPTH −20	Milling depth
CYCL DEF 5.3 PECKG −6	Pecking depth
F 80	Feed rate for pecking
CYCL DEF 5.4 RADIUS 35	Circle radius
CYCL DEF 5.5 F100 DR−	Milling feed rate and direction of the cutter path
L X+60 Y+50 R0 FMAX M03	Pilot positioning in X and Y
L Z+2 FMAX M99	Starting position in Z, cycle call

SL Cycles
Fundamentals

CYCL DEF 14
CONTOUR
GEOMETRY

CYCL DEF 15
PILOT DRILLING

CYCL DEF 6
ROUGH-OUT

CYCL DEF 16
CONTOUR
MILLING

M2

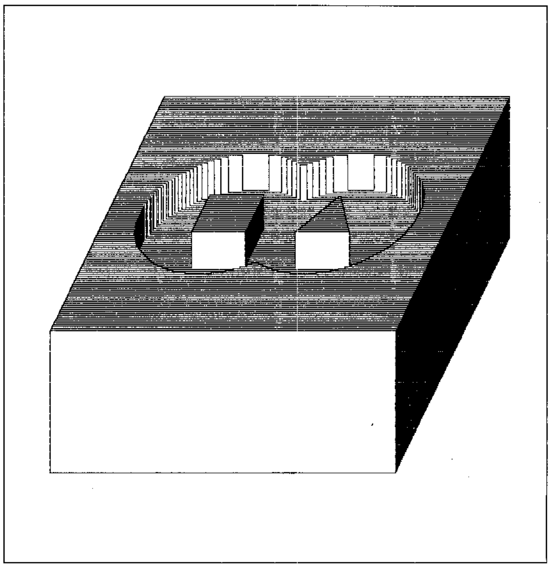
Subprograms of
the individual
subcontours

Scheme of a program with SL cycles

The group of cycles that we categorize as SL cycles is designed for efficient programming and milling of contours with one or more tools. The contour can be composed of several overlapping subcontours which are defined in separate subprograms.

The term **SL cycles** is derived from the characteristic **Subcontour List** of cycle 14: The control superimposes the separate contours to form a single whole. The programmer need not calculate the points of intersection!

The subcontours are defined in the form of subprograms. Cycle 14, **Contour geometry,** is more or less a list of the corresponding label numbers and can store up to 12 islands or pockets. Each subcontour is programmed as a closed pattern.



To be able to work with several tools, the machining task is described in cycle 14 without tool-specific data or feed values; those are entered in the individual cycles:

Cycle 15 Pilot drilling (if required)
Cycle 6 Rough-out
Cycle 16 Contour milling (finishing)

Each subprogram must specify whether **RL or RR radius compensation** applies and in which **direction** the contour is to be machined. The control deduces from these data whether the specific subprogram describes a **pocket** or an **island.**

The control recognizes a pocket if the tool path lies inside the contour.
The control recognizes an island if the tool path lies outside the contour.

Please be sure to run a graphic simulation before executing a program to see whether the contour was computed by the control as desired.
All coordinate transformations are allowed in programming the contours.

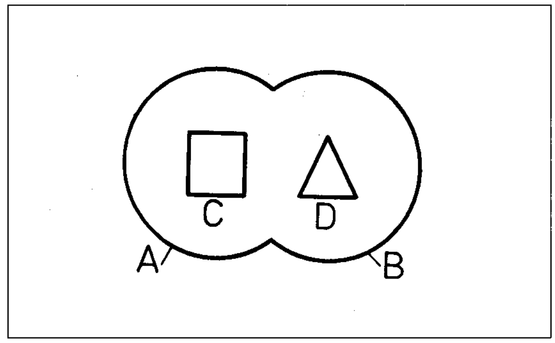
Not all of the SL cycles are always required.

For easier familiarization, the following examples begin with only the rough-out cycle and then proceed progressively to the full range of functions.

SL Cycles
Cycle 14: Contour geometry
Cycle 6: Rough-out

Cycle 14 Contour geometry

The label numbers (subprograms) of the subcontours are specified in cycle 14.
Up to 12 label numbers can be entered.
The TNC computes the intersections of the resulting contour from the subcontours.
Cycle 14 is immediately effective after definition (this cycle cannot be called).
The list of subcontours in cycle 14 should begin with a pocket.



A, B = Pockets
C, D = Islands

Example

```
5 CYCL DEF 14.0 CONTOUR GEOM.
6 CYCL DEF 14.1 CONTOUR LABEL 11 / 12 / 13
```

The definition occupies up to 3 program blocks. The subprograms 11, 12 and 13 define the complete contour in the example.

Cycle 6 Rough-out

Cycle 6 specifies the cutting path and partitioning. It must be called, and can be executed separately.

Tool required

Cycle 6 requires a center-cut end mill (ISO 1641) if no pilot drilling is desired and if the tool must repeatedly jump over contours and plunge to the milling depth.

Input data

Setup clearance (A), milling depth (B), pecking depth (C) are incremental with the same signs (usually negative).

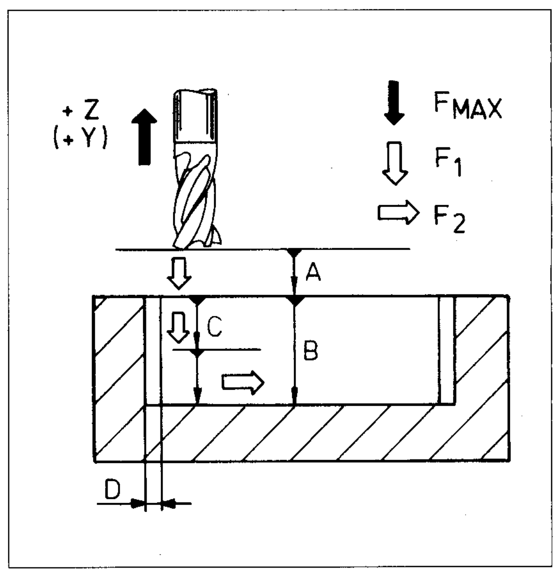
Feed rate for pecking: traversing speed of the tool at penetration (F1).

Finishing allowance: allowance in the machining plane (positive value) (D).

Rough-out angle: roughing out direction relative to the reference axis of the machining plane.

Feed rate: traversing speed of the tool in the machining plane (F2).

The tool must be positioned at the setup clearance (A) before the cycle call.



Example

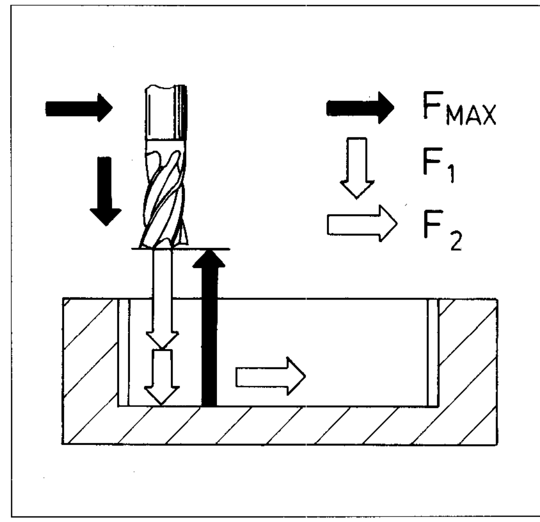
```
16 CYCL DEF 6.0 ROUGH-OUT
17 CYCL DEF 6.1 SET UP –2
                DEPTH –20
18 CYCL DEF 6.2 PECKG –10
   F40          ALLOW +1
19 CYCL DEF 6.3 ANGLE +0
   F60
```

Setup clearance
Milling depth
Pecking depth
Feed rate for penetration and finishing allowance
Rough-out angle
Feed rate in the machining plane

SL Cycles
Cycle 6: Rough-out

Process

The tool is automatically positioned over the first penetration point (with finishing allowance). It may be necessary to pilot position the tool before the call to prevent collision. The tool penetrates at the feed rate for pecking.



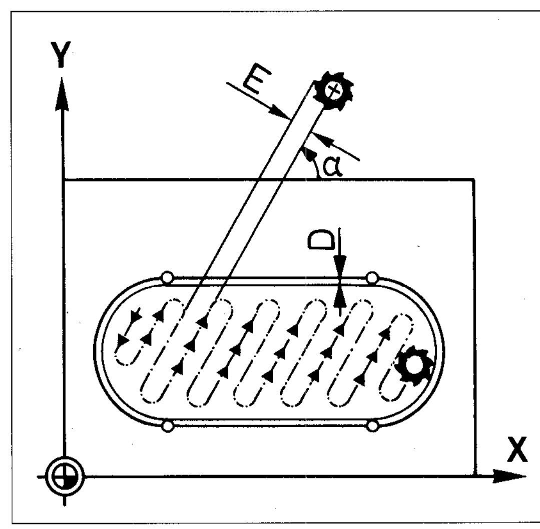
Milling the contour

After reaching the first pecking depth, the tool mills the first subcontour at the programmed milling feed rate with the finishing allowance.
At the penetration point, the tool is advanced to the next pecking depth. This process is repeated until the programmed milling depth is attained. Further subcontours are milled in the same manner.

Clearing the surface

The surface is then roughed out. The feed direction corresponds to the programmed rough-out angle and can be set, so the resulting cuts are as long as possible with few cutting movements. The stepover equals the tool radius. Clearing out can be performed with multiple downfeeds.

The tool is retracted to the setup clearance at the end of the cycle.

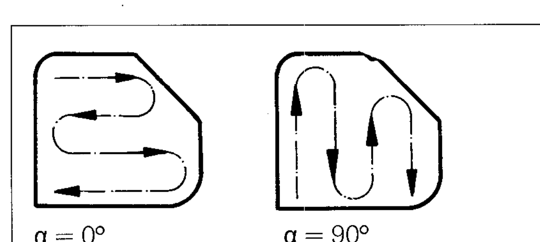


D = Finishing allowance
E = Stepover
α = Rough-out angle

Sequence contour milling/ surface machining

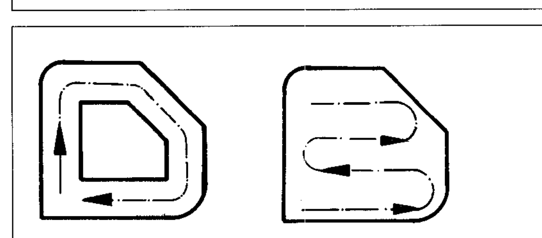
A machine parameter determines whether the contour is milled first and then surface machined or vice versa.

In the same way is specified whether contour milling or roughing out is performed continuously over all infeeds, or for each infeed in the specified sequence.



Climb/ conventional

A machine parameter also determines whether the contour is milled conventionally or by climb cutting (see index General Information, MOD Functions, User parameter MP 7420).



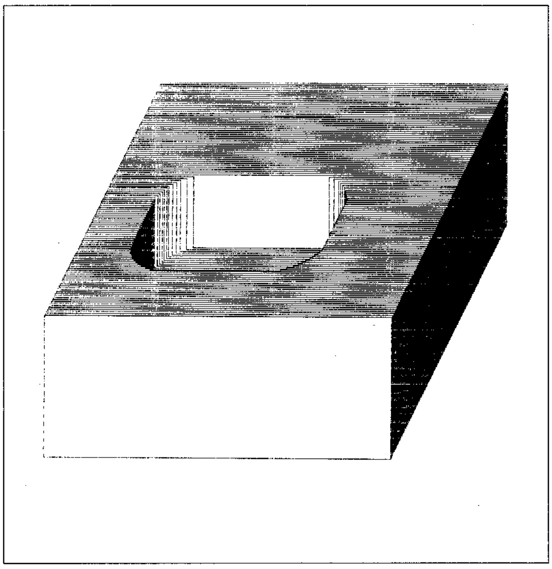
Begin with contour milling

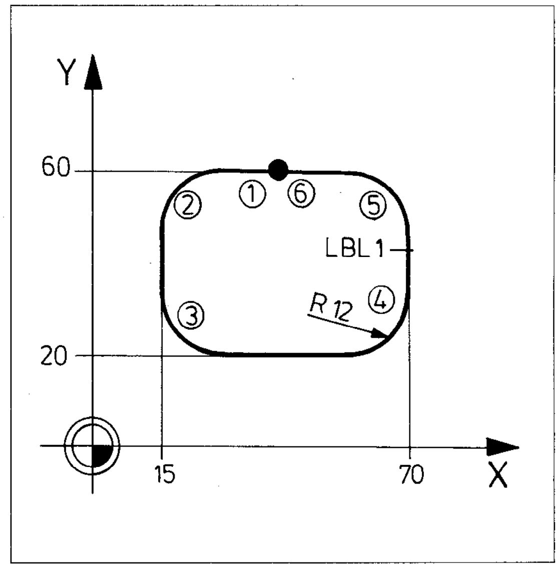
Begin with surface clearing

SL Cycles
Roughing-out a rectangular pocket

Task

Interior machining of rectangular pocket with rounded corners, with a center-cut end mill, tool radius 5 mm.





PGM 7206

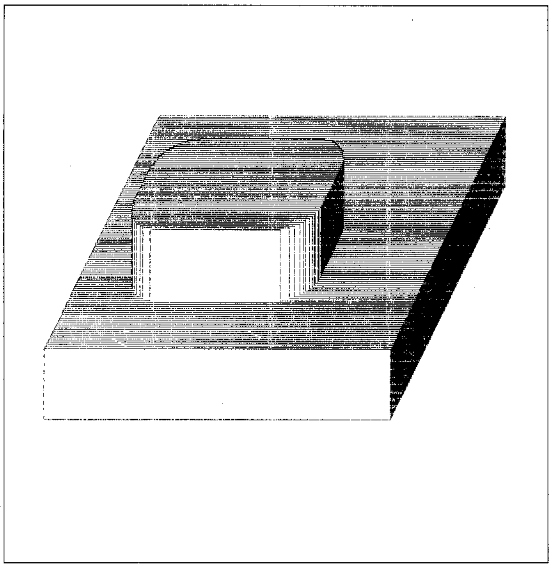
0 BEGIN PGM 7206 MM	
1 BLK FORM 0.1 Z X+0 Y+0 Z–40	Blank min. point
2 BLK FORM 0.2 X+100 Y+100 Z+0	Blank max. point
3 TOOL DEF 1 L+0 R+5	Tool definition
4 TOOL CALL 1 Z S 111	Tool call
5 L Z+100 R0 FMAX M03	
6 CYCL DEF 14.0 CONTOUR GEOM.	
7 CYCL DEF 14.1 CONTOUR LABEL 1	"List" of contour subprograms
8 CYCL DEF 6.0 ROUGH-OUT	Definition for "rough-out"
9 CYCL DEF 6.1 SET UP –2 DEPTH –20	
10 CYCL DEF 6.2 PECKG –8 F100 ALLOW +0	
11 CYCL DEF 6.3 ANGLE +0 F500	
12 L X+40 Y+50 Z+2 R0 FMAX M99	Pilot positioning, cycle call
13 L Z+20 FMAX M02	Retract, return jump to start of program
14 LBL 1	Contour subprogram
15 L X+40 Y+60 **RL**	
16 L X+15	From radius compensation RL and counterclockwise machining direction, the control deduces a **pocket.**
17 RND R12	
18 L Y+20	
19 RND R12	
20 L X+70	
21 RND R12	
22 L Y+60	
23 RND R12	
24 L X+40	
25 LBL 0	
26 END PGM 7206 MM	

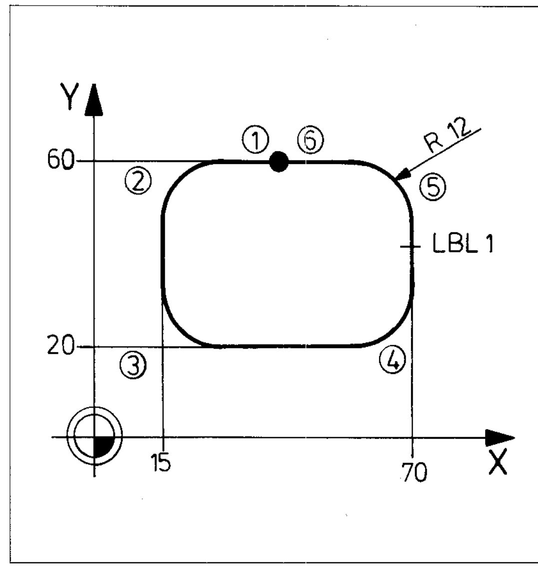
PGM 7207 creates a contour island with identical dimensions.

SL Cycles
Roughing-out a rectangular island

Task

Exterior machining of rectangular island with rounded corners, with a center-cut end mill, tool radius 5 mm.





PGM 7207

0 BEGIN PGM 7207 MM	
1 BLK FORM 0.1 Z X+0 Y+0 Z–40	Blank
2 BLK FORM 0.2 X+100 Y+100 Z+0	
3 TOOL DEF 1 L+0 R+5	Tool
4 TOOL CALL 1 Z S 111	
5 L Z+100 R0 FMAX M03	
6 CYCL DEF 14.0 CONTOUR GEOM.	
7 CYCL DEF 14.1 CONTOUR LABEL 2 / 1	"List" of contour subprograms (sequence!)
8 CYCL DEF 6.0 ROUGH-OUT	Definition for "rough-out"
9 CYCL DEF 6.1 SET UP –2 DEPTH –20	
10 CYCL DEF 6.2 PECKG –8 F100 ALLOW +0	
11 CYCL DEF 6.3 ANGLE +0 F500	
12 L X+40 Y+50 Z+2 R0 FMAX M99	Pilot positioning, cycle call
13 L Z+20 FMAX M02	Retract, return jump to start of program
14 LBL 1	From radius compensation RR and the counterclockwise machining direction, the control deduces an **island.**
15 L X+40 Y+60 **RR**	
16 L X+15	
17 RND R12	
18 L Y+20	
19 RND R12	
20 L X+70	
21 RND R12	
22 L Y+60	
23 RND R12	
24 L X+40	
25 LBL 0	
26 LBL 2	Auxiliary pocket to externally limit the machined surface
27 L X–5 Y–5 **RL**	
28 L X+105	
29 L Y+105	
30 L X–5	
31 L Y–5	
32 LBL 0	
33 END PGM 7207 MM	

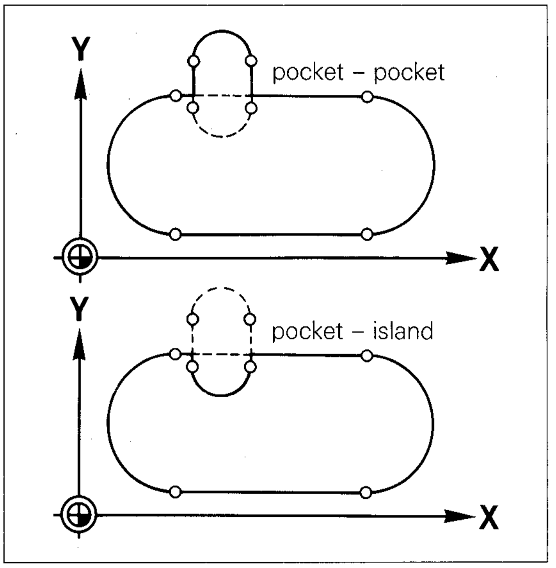
PGM 7206 creates a contour pocket with identical dimensions.

SL Cycles
Overlaps

Overlapping pockets and islands

Pockets and islands can be overlapped (superimposed). The resulting contour is computed by the TNC.

The area of a pocket can for example be enlarged by an another pocket or reduced by an island.



Starting position

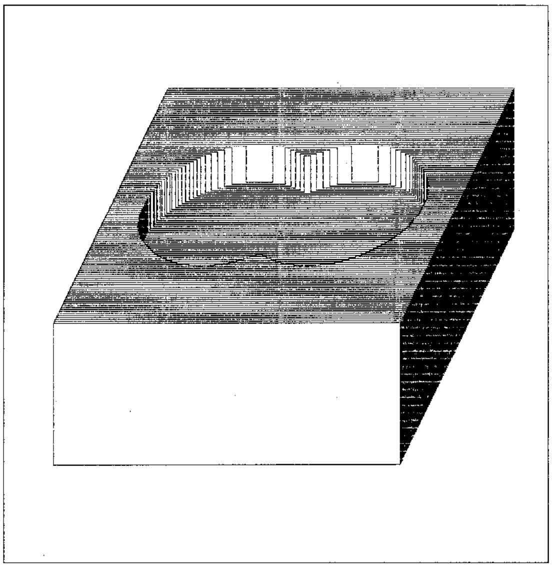
Machining begins at the starting position of the first contour label of cycle 14. The starting positions should be located as far as possible from the superimposed contours.

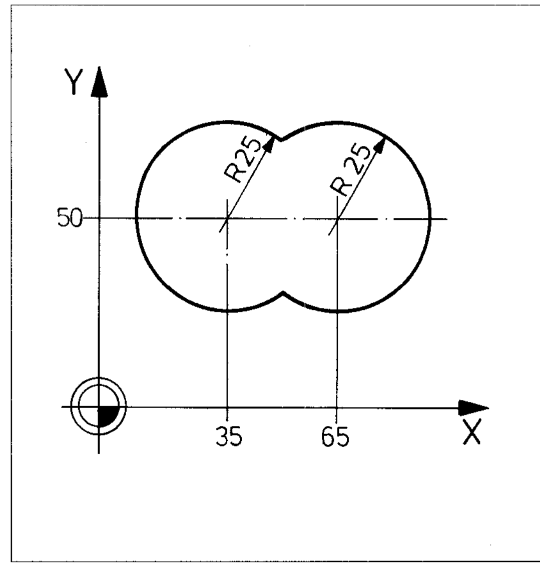
If the subcontours are always defined in the same working direction, then for example with a positive working direction pockets can be easily recognized by the "RL" compensation, and islands by the "RR".

SL Cycles
Overlapping pockets

Task

Interior machining of overlapping pockets with a center-cut end mill, tool radius 3 mm.





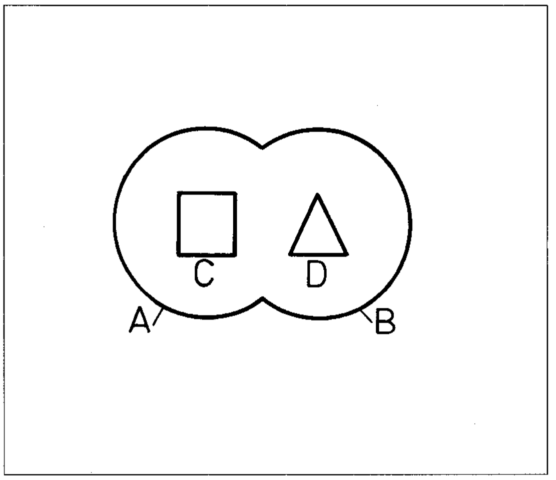
PGM 7208

0 BEGIN PGM 7208 MM	
1 BLK FORM 0.1 Z X+0 Y+0 Z–40	Blank, tool axis
2 BLK FORM 0.2 X+100 Y+100 Z+0	
3 TOOL DEF 2 L+0 R+3	Tool
4 TOOL CALL 2 Z S 100	
5 L Z+200 R0 FMAX	
6 L X+50 Y+50 FMAX M03	Pilot position X and Y, spindle on
7 CYCL DEF 14.0 CONTOUR GEOM.	"List" of contour subprograms
8 CYCL DEF 14.1 CONTOUR LABEL 1 / 2	
9 CYCL DEF 6.0 ROUGH-OUT	Definition for "rough-out"
10 CYCL DEF 6.1 SET UP –2 DEPTH –10	
11 CYCL DEF 6.2 PECKG –10 F500 ALLOW +0	
12 CYCL DEF 6.3 ANGLE +0 F500	
13 L Z+2 R0 FMAX M99	Setup clearance Z, cycle call
14 L Z+200 R0 FMAX M02	Retract return jump to start of program

Note

Machining begins with the first contour label defined in block 8!
The first pocket must begin outside the second pocket.

SL Cycles
Overlapping pockets



Points of intersection

The pocket elements A and B overlap each other. They are programmed as full circles.
Since the control automatically computes the points of intersection S1 and S2, these points need not be programmed.

```
15 LBL 1
16 L X+10 Y+50 RL
17 CC X+35 Y+50
18 C X+10 Y+50 DR+
19 LBL 0
```

} A Left pocket

```
20 LBL 2
21 L X+90 Y+50 RL
22 CC X+65 Y+50
23 C X+90 Y+50 DR+
24 LBL 0
```

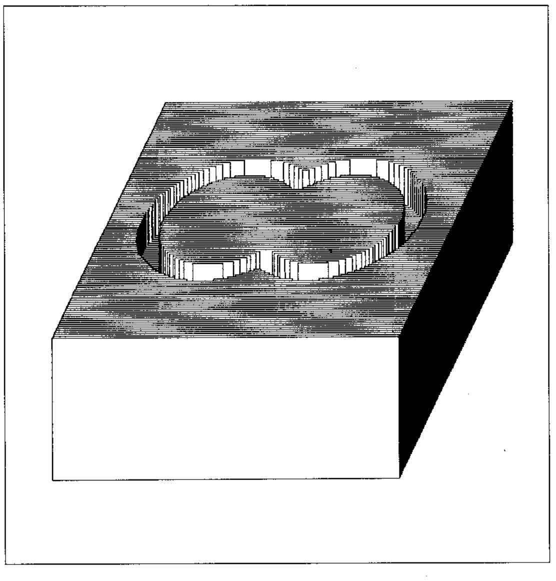
} B Right pocket

```
25 END PGM 7208 MM
```

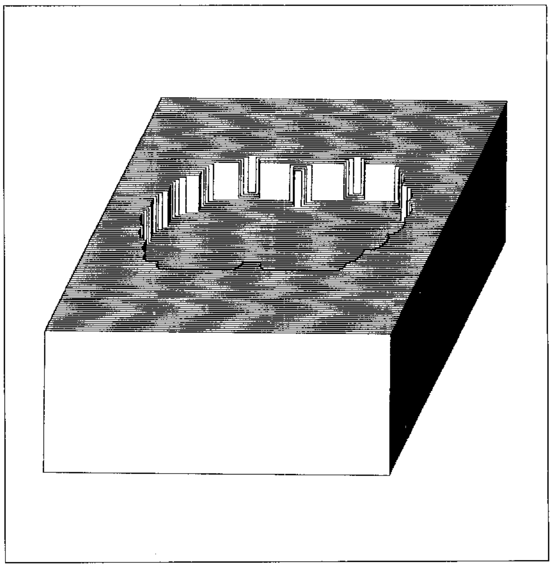
Execution

Depending on the control setup (machine parameters), machining begins either with the contour edge or the surface.

Contour edge is machined first



Surface is machined first



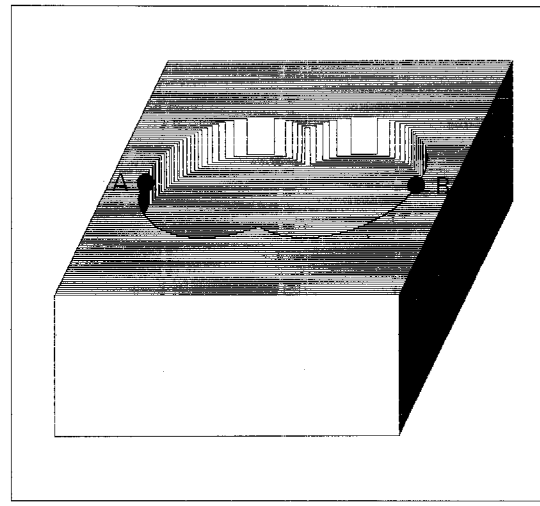
SL Cycles
Overlapping pockets

"Sum" area

Both areas (element A and element B) along with the common overlapping area are to be machined.
- A and B must be pockets.
- the first pocket (in cycle 14) must begin outside the second.

16 LBL 1
17 L X+10 Y+50 **RL**
18 CC X+35 Y+50
19 C X+10 Y+50 DR+
20 LBL 0

21 LBL 2
22 L X+90 Y+50 **RL**
23 CC X+65 Y+50
24 C X+90 Y+50 DR+
25 LBL 0



● A and ● B are the starting points of the contour labels.

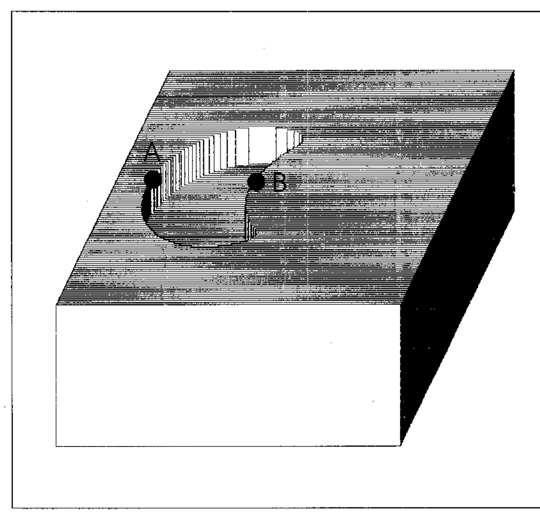
"Difference" area

Area A is to be machined without the portion overlapped by B:

- A must be a pocket and B an island.
- A must begin outside of B.

15 LBL 1
16 L X+10 Y+50 **RL**
17 CC X+35 Y+50
18 C X+10 Y+50 DR+
19 LBL 0

20 LBL 2
21 L X+90 Y+50 **RR**
22 CC X+65 Y+50
23 C X+90 Y+50 DR+
24 LBL 0



An island can also reduce several pocket areas. The starting points of the pocket contours must all be outside the island.

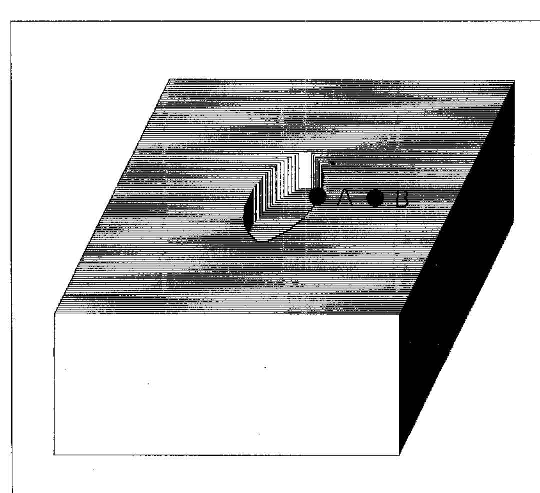
"Intersecting" area

Only the area overlapped commonly by A and B is to be machined.

- A and B must be pockets.
- A must begin inside of B.

15 LBL 1
16 L X+60 Y+50 **RL**
17 CC X+35 Y+50
18 C X+60 Y+50 DR+
19 LBL 0

20 LBL 2
21 L X+90 Y+50 **RL**
22 CC X+65 Y+50
23 C X+90 Y+50 DR+
24 LBL 0



SL Cycles
Overlapping islands

Expanding PGM 7208

7 CYCL DEF 14.0 CONTOUR GEOM.
8 CYCL DEF 14.1 CONTOUR LABEL 1 / 2 / 5

25 LBL 5
26 L X+5 Y+5 **RL**
27 L X+95
28 L Y+95
29 L X+5
30 L Y+5
31 LBL 0

An island always requires an additional outer limit = pocket (here, LBL 5).

A pocket can also reduce several island areas. This pocket must begin inside the first island. The starting points of the remaining intersected island contours must be outside the pocket.

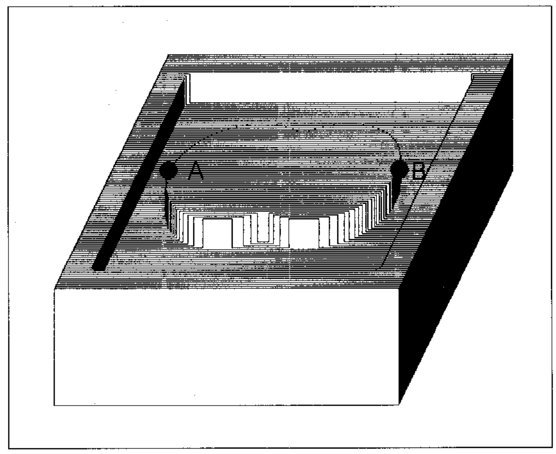
"Sum" area

Both areas (element A and element B) along with the common overlapping area are to remain unmachined.

- A and B must be islands.
- The first island must begin outside the second.

16 LBL 1
17 L X+10 Y+50 **RR**
18 CC X+35 Y+50
19 C X+10 Y+50 DR+
20 LBL 0

21 LBL 2
22 L X+90 Y+50 **RR** F500
23 CC X+65 Y+50
24 C X+90 Y+50 DR+
25 LBL 0



● A, ● B are the starting points of the subcontours.

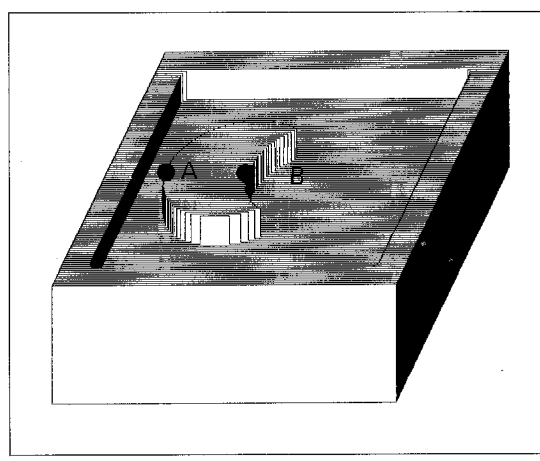
"Difference" area

Area A is to remain unmachined except that portion overlapped by B.

- A must be an island and B a pocket.
- A must begin outside of B.

15 LBL 1
16 L X+10 Y+50 **RR**
17 CC X+35 Y+50
18 C X+10 Y+50 DR+
19 LBL 0

20 LBL 2
21 L X+40 Y+50 **RL**
22 CC X+65 Y+50
23 C X+40 Y+50 DR+
24 LBL 0



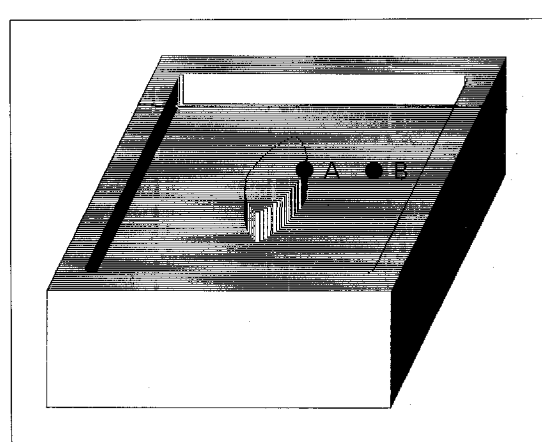
"Intersecting" area

Only the area overlapped commonly by A and B remains unmachined.

- A and B must be islands.
- A must begin inside of B.

15 LBL 1
16 L X+60 Y+50 **RR**
17 CC X+35 Y+50
18 C X+60 Y+50 DR+
19 LBL 0

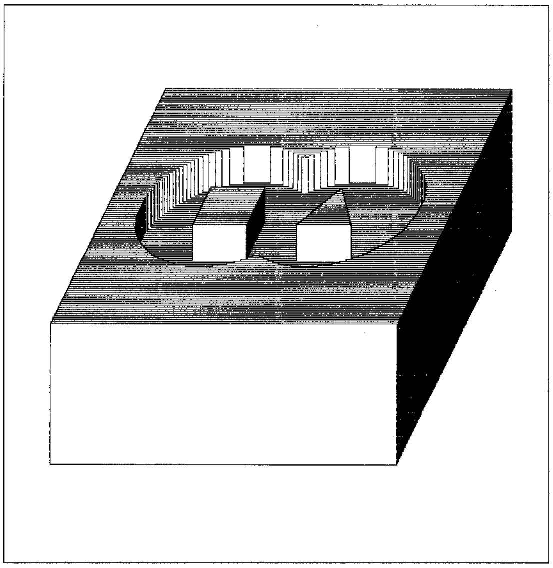
20 LBL 2
21 L X+90 Y+50 **RR**
22 CC X+65 Y+50
23 C X+90 Y+50 DR+
24 LBL 0

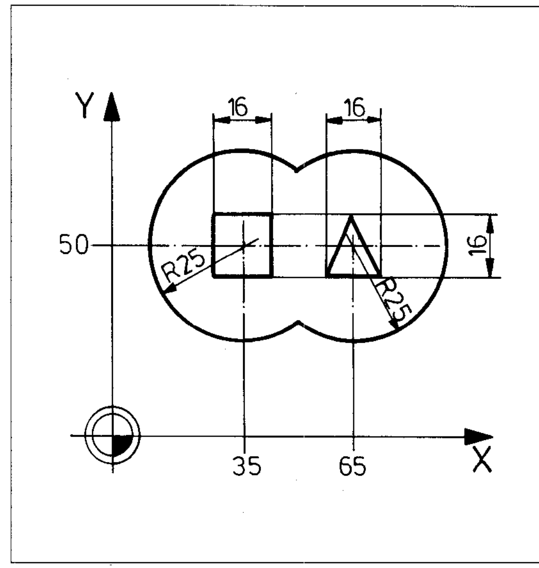


SL Cycles
Overlapping pockets and islands

Task

Interior machining of overlapping pockets and islands with a center-cut end mill, tool radius 3 mm. Islands are located within a pocket area.





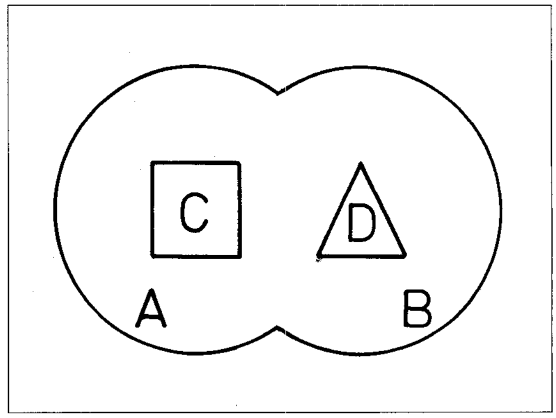
Main PGM 7209

```
0 BEGIN PGM 7209 MM
1 BLK FORM 0.1 Z X+0 Y+0 Z–40
2 BLK FORM 0.2 X+100 Y+100 Z+0
3 TOOL DEF 2 L+0 R+3
4 CYCL DEF 14.0 CONTOUR GEOM.
5 CYCL DEF 14.1 CONTOUR LABEL 1 / 2 / 3 / 4        List of contour elements
6 LBL 10
7 TOOL CALL 0 Z S 0
8 L Z+20 R0 FMAX
9 L X–20 Y–20 FMAX
10 LBL 0
11 STOP M06
12 TOOL CALL 2 Z S 100

13 CYCL DEF 6.0 ROUGH-OUT
14 CYCL DEF 6.1 SET UP –2 DEPTH –10
15 CYCL DEF 6.2 PECKG –5 F500 ALLOW +0
16 CYCL DEF 6.3 ANGLE +0 F500
17 L Z+2 R0 FMAX
18 CYCL CALL M03
19 CALL LBL 10
20 L Z+20 R0 FMAX M02
```

PGM 7209 is an expansion of PGM 7208: the interior islands are added (subprograms 3 and 4).

SL Cycles
Overlapping pockets and islands

The entire contour is composed of the elements A and B, i.e. two overlapping pockets and C and D, i.e. two islands within these pockets.



Contour subprograms for PGM 7209

```
21 LBL 1
22 L X+35 Y+25 RL
23 CC X+35 Y+50
24 C X+35 Y+25 DR+
25 LBL 0
```
A Left pocket

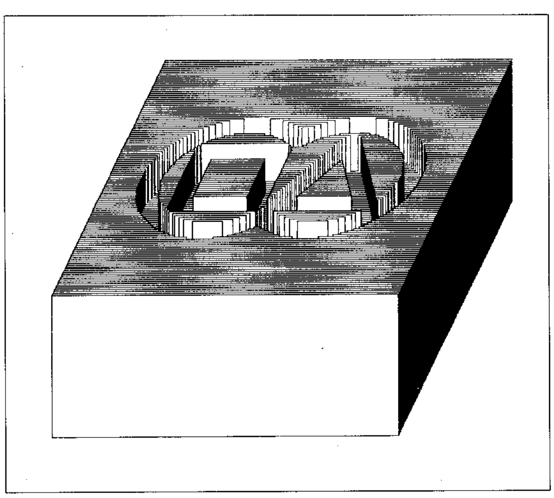
```
26 LBL 2
27 L X+65 Y+25 RL
28 CC X+65 Y+50
29 C X+65 Y+25 DR+
30 LBL 0
```
B Right pocket

```
31 LBL 3
32 L X+35 Y+42 RR
33 L X+43
34 L Y+58
35 L X+27
36 L Y+42
37 L X+35
38 LBL 0
```
C Square island

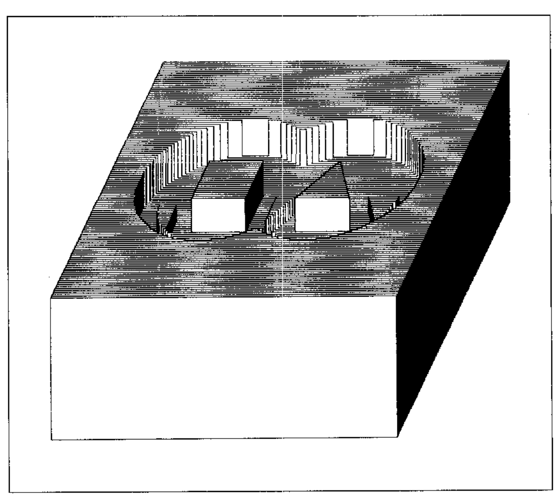
```
39 LBL 4
40 L X+65 Y+42 RR
41 L X+73
42 L X+65 Y+58
43 L X+57 Y+42 RR
44 L X+65
45 LBL 0
46 END PGM 7209 MM
```
D Triangular island

Execution

Machining of the contour edges



Surface machining (unfinished)



CYCL DEF CYCL CALL

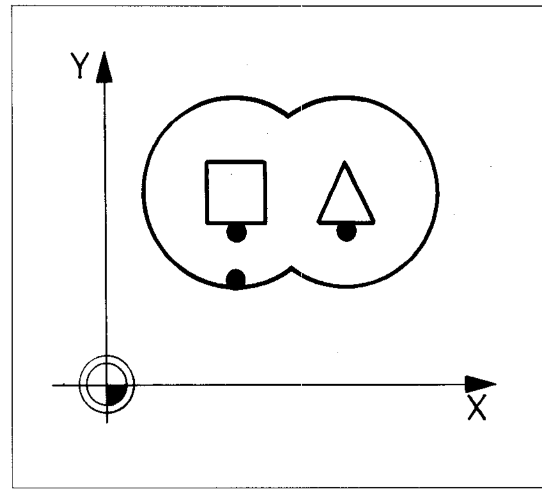
SL Cycles
Cycle 15: Pilot drilling

The cycle

Pilot drill the cutter infeed points at the starting points of the subcontours, compensated by the finishing allowance.

For closed contour sequences resulting from multiple superimposed pockets and islands, the infeed point is the starting point of the first subcontour.

This cycle must be called!



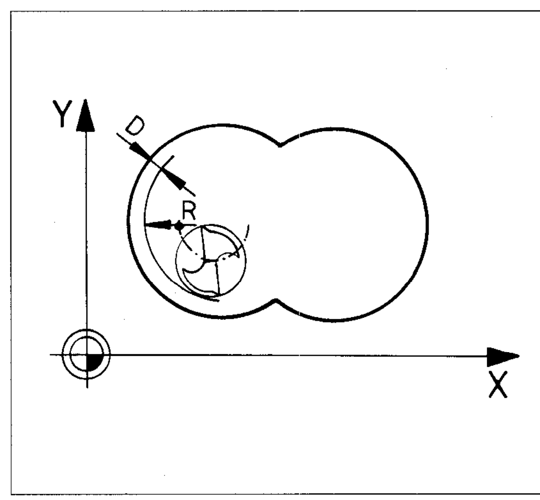
● Cutter infeed point

Input data

The input values are identical to pecking; enter a **finishing allowance** in addition.

Finishing allowance: allowance for drilling (positive value), effective in the working plane.
The sum of the tool radius and the finishing allowance should be the same for pilot drilling and roughing-out.

The tool must be at the setup clearance before calling the cycle!



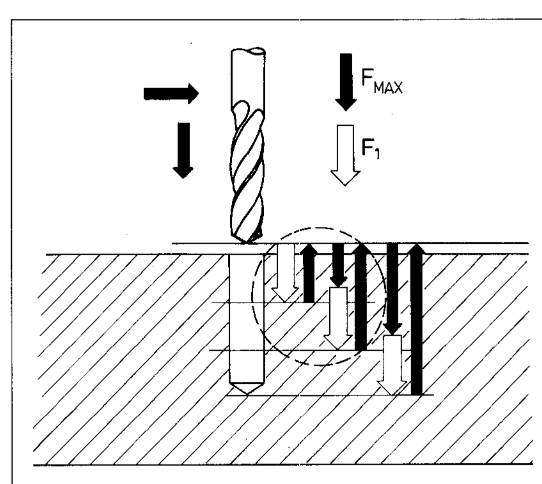
D = Finishing allowance
R = Tool radius

Process

The tool is automatically positioned over the first infeed point, offset by the allowance.
The tool may have to be pilot positioned to prevent collision!

The drilling process is identical to the fixed cycle "pecking" (cycle 1).

Subsequently, the tool is positioned over the second infeed point at the programmed setup clearance, and the drilling procedure is repeated.



Example

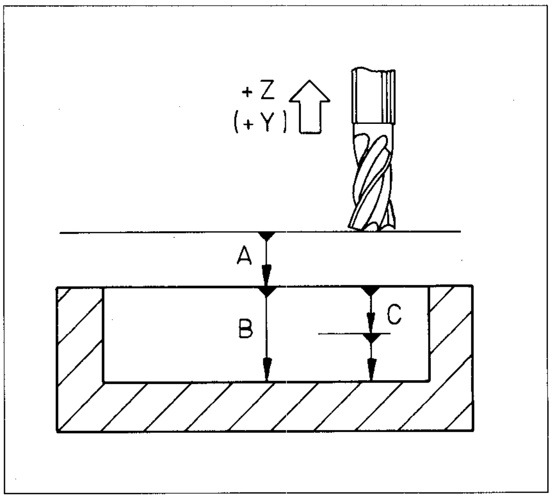
18 CYCL DEF 15.0 PILOT DRILL	
19 CYCL DEF 15.1 SET UP –2	Setup clearance
DEPTH –20	Drilling depth
20 CYCL DEF 15.2 PECKG –10	Pecking depth
F40 ALLOW +1	Feed rate for infeed and finishing allowance

SL Cycles
Cycle 16: Contour milling (finishing)

The cycle

Cycle 16 "contour milling" is used for finishing the contour pocket.

The cycle can also be generally used to mill contours made up of subcontours.
This offers the following benefits:
- contour intersections are computed,
- collisions are avoided.



Tool required

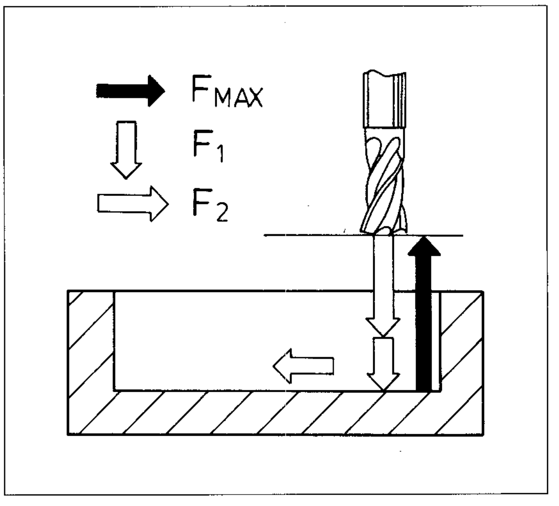
The cycle requires a center cutting tool.

The cycle must be called!
The setup clearance A, milling depth B and pecking depth C are identical to pecking.
The signs must be the same (normally negative).

Input data

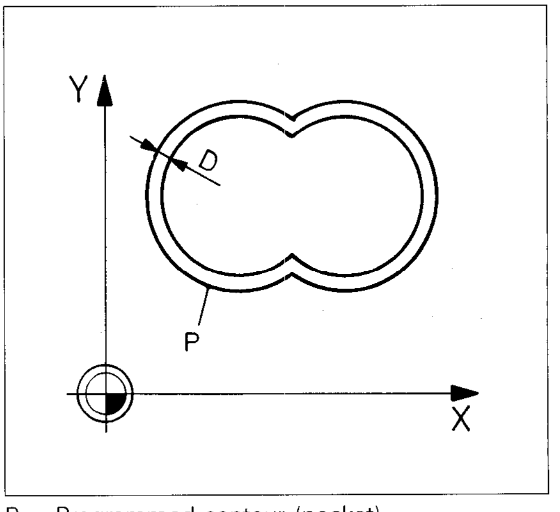
Feed rate for pecking: tool traversing speed at infeed (F_1).
Rotating direction for contour milling: milling direction along the pocket contour (island contours: opposite milling direction).
For the following directions, M3 means
DR+: down-cut milling for pocket and island,
DR–: up-cut milling for pocket and island.
Feed rate: tool traversing speed in the machining plane (F_2).

The tool must be at the setup clearance (A) prior to the cycle call.



Process

The tool is automatically positioned over the first contour point.
Beware of collisions with clamping devices!
The tool then penetrates the workpiece at the programmed **feed rate** to the first **pecking depth.**
After reaching the first pecking depth, the tool mills the first contour at the programmed **feed rate** in the specified **rotating direction.**
At the infeed point, the tool is advanced to the next pecking depth. The procedure is repeated until the programmed **milling depth** is attained.
The next subcontours are milled in the same manner.



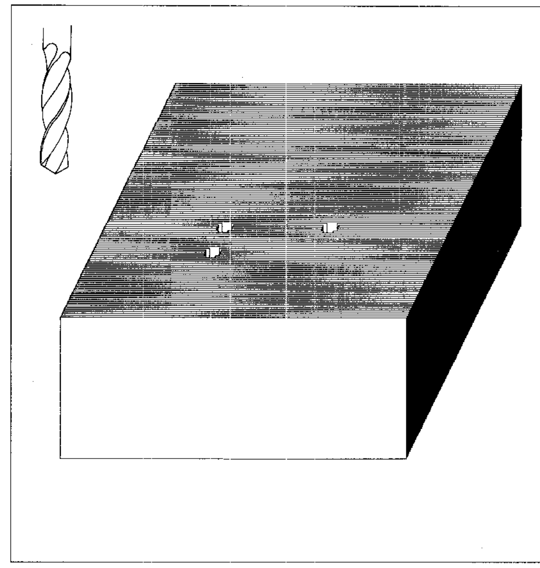
P = Programmed contour (pocket)
D = Finishing allowance from cycle 6 rough-out

Example

25 CYCL DEF 16.0 CONTOUR MILLG.	
26 CYCL DEF 16.1 SET UP –2	Setup clearance
DEPTH –20	Milling depth
27 CYCL DEF 16.2 PECKG –10	Pecking depth
F40 DR– F60	Feed rate for infeed, milling direction and feed rate in the working plane

SL Cycles
Machining with several tools

The following scheme illustrates the application of the SL cycles pilot drilling, rough-out, and contour milling in one program:

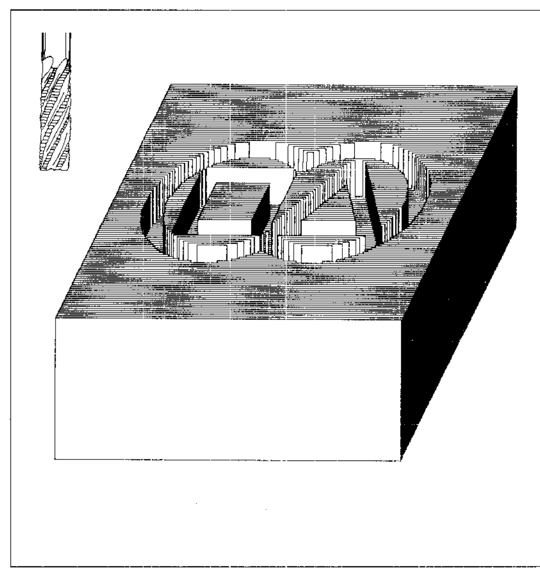


List of contour subprograms

Cycle definition:
CYCL DEF 14.0 CONTOUR GEOM.
No call!

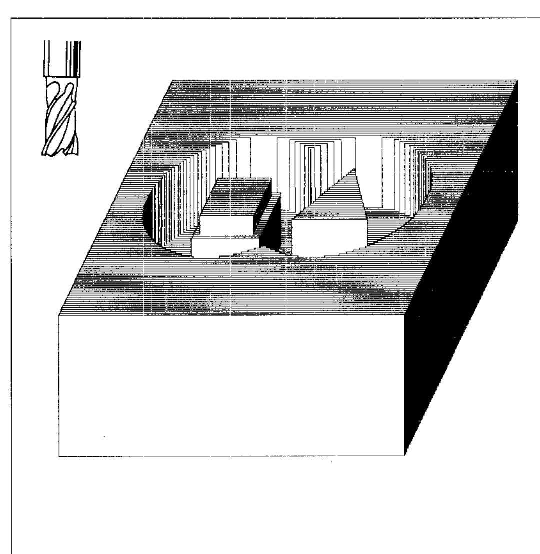
Drilling

Define and call the drill
Cycle definition:
CYCL DEF 15.0 PILOT DRILL
Pilot positioning,
Cycle call!



Rough-out

Define and call the roughing cutter
Cycle definition:
CYCL DEF 6.0 ROUGH-OUT
Pilot positioning,
Cycle call!



Finishing

Define and call the finishing cutter
Cycle definition:
CYCL DEF 16.0 CONTOUR MILLG.
Pilot positioning,
Cycle call!

STOP M02

Contour subprograms

Subprograms for the subcontours

SL Cycles
Machining with several tools

Task

Overlapping pockets with islands.

Interior machining with pilot drilling, roughing, finishing.

Main PGM 7210

0 BEGIN PGM 7210 MM	
1 BLK FORM 0.1 Z X+0 Y+0 Z–40	
2 BLK FORM 0.2 X+100 Y+100 Z+0	
3 TOOL DEF 1 L+0 R+2.2	Drill
4 TOOL DEF 2 L+0 R+3	Roughing cutter
5 TOOL DEF 3 L+0 R+2.5	Finishing cutter
6 CYCL DEF 14.0 CONTOUR GEOM.	
7 CYCL DEF 14.1 CONTOUR LABEL 1 / 2 / 3 / 4	
8 LBL 10	
9 TOOL CALL 0 Z S 0	Tool change
10 L Z+20 R0 FMAX	
11 L X–20 Y–20 R0 FMAX	
12 LBL 0	
13 STOP M06	
14 TOOL CALL 1 Z S 100	
15 CYCL DEF 15.0 PILOT DRILL	Pilot drilling
16 CYCL DEF 15.1 SET UP –2 DEPTH –20	
17 CYCL DEF 15.2 PECKG –5 F500 ALLOW +2	
18 L Z+2 R0 FMAX	
19 CYCL CALL M03	
20 CALL LBL 10	
21 STOP M06	
22 TOOL CALL 2 Z S 100	
23 CYCL DEF 6.0 ROUGH-OUT	Roughing-out
24 CYCL DEF 6.1 SET UP –2 DEPTH –20	
25 CYCL DEF 6.2 PECKG –5 F100 ALLOW +2	
26 CYCL DEF 6.3 ANGLE +0 F500	
27 L Z+2 R0 FMAX	
28 CYCL CALL M03	
29 CALL LBL 10	
30 STOP M06	
31 TOOL CALL 3 Z S 500	
32 CYCL DEF 16.0 CONTOUR MILLG.	Finishing
33 CYCL DEF 16.1 SET UP –2 DEPTH –20	
34 CYCL DEF 16.2 PECKG –5 F100 DR– F500	
35 L Z+2 R0 FMAX	
36 CYCL CALL M03	
37 CALL LBL 10	
38 L Z+20 R0 FMAX M02	Retract and return jump

PGM 7210 builds upon PGM 7209:

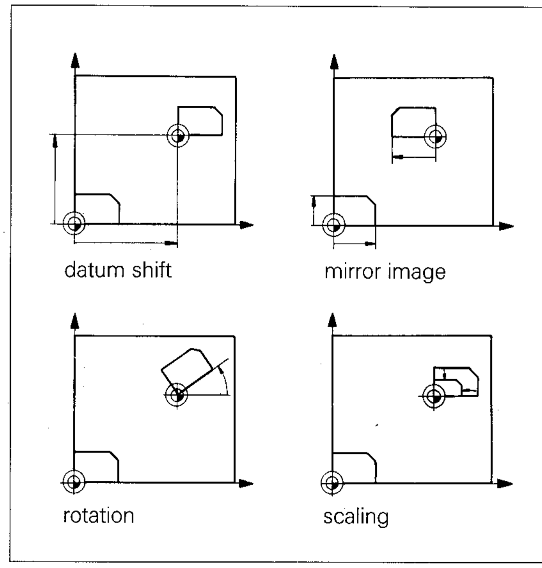
The main program section is expanded by the cycle definitions and calls for pilot drilling and finishing.

The contour subprograms 1 to 4 are identical to those in PGM 7209 and are to be added after block 38.

Coordinate Transformations
Overview

The following cycles serve for coordinate transformations:

7.0 Datum shift
8.0 Mirror image
10.0 Rotation
11.0 Scaling



Original

With the help of coordinate transformations, a program section can be executed as a variant of the "original".
In the following descriptions, subprogram 1 is always the "original" subprogram (identified by the grey background).

Immediate activation

Every transformation is immediately valid – without being called.

Duration of activation

A coordinate transformation remains valid until it is changed or cancelled.

Its effect is not impaired by interrupting and aborting program run. This is also true when the same program is restarted from another location with "GOTO □".

End of activation

You can cancel coordinate transformations in the following ways:

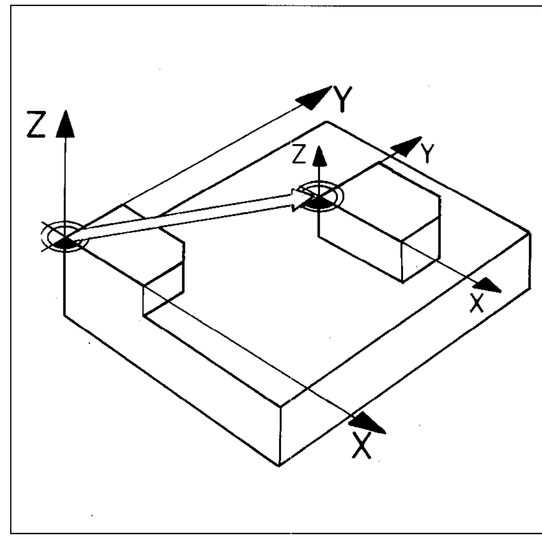
- Cycle definition for basic condition (e.g.: scaling factor 1.0);
- Programming of miscellaneous functions M02 or M30, or END PGM ... (depending on the machine parameters);
- Selecting another program with "PGM NR" in the operating mode program run "full sequence" or "single block".

Error message

CYCL INCOMPLETE

This error message is displayed if a fixed cycle is called after defining a transformation but no machining cycle was defined. Otherwise the control executes the fixed cycle which was last defined.

Coordinate Transformations
Cycle 7: Datum shift



The cycle

You can program a datum shift (also called a zero offset) to any point within a program. The manually set absolute workpiece datum remains unchanged.

Thus, identical machining steps (e.g. subprograms) can be executed at different positions on the workpiece without having to reenter the program section each time.

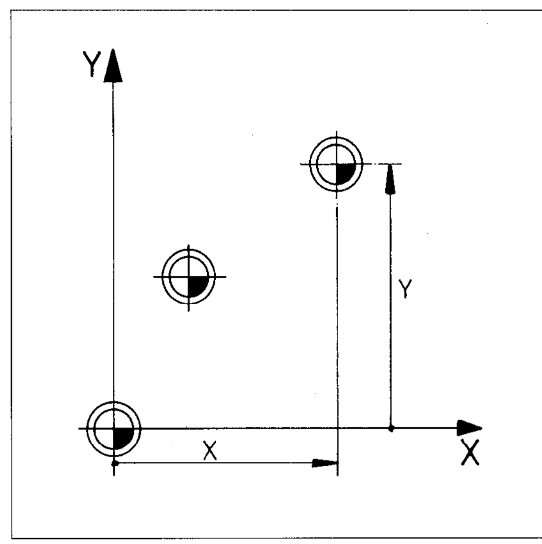
Combining with other coordinate transformations

If a datum shift is to be combined with other transformations, the shift has to be made **before** the other transformations!

Effect

For a datum shift definition, only the coordinates of the new datum are to be entered.

An active datum shift is displayed in the status field. All coordinate inputs then refer to the new datum.



Absolute datum shift

Incremental/ absolute

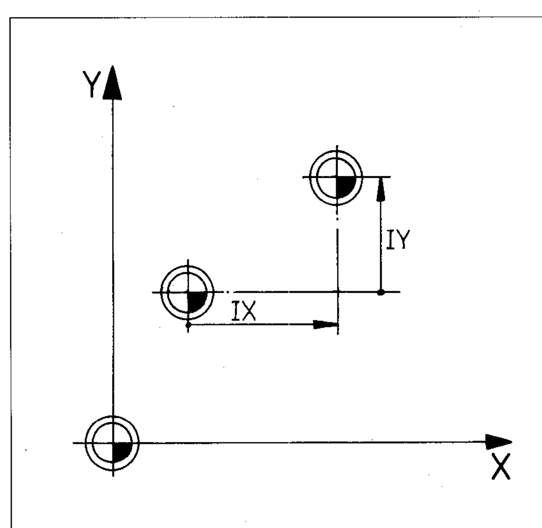
In the cycle definition the coordinates can be entered as absolute or incremental dimensions:

- **Absolute:** The coordinates of the new datum refer to the manually set workpiece datum. Refer to the center figure.
- **Incremental:** The coordinates of the new datum refer to the last valid datum, which can itself be shifted. Refer to the lower figure.

Cancelling the shift

A datum shift is cancelled by entering the datum shift X0/Y0/Z0 ...
Only the "shifted" axes have to be entered.

```
CYCL DEF 7.0 DATUM SHIFT
CYCL DEF 7.1 X+0
CYCL DEF 7.2 Y+0
```



Incremental datum shift

Coordinate Transformations
Cycle 7: Datum shift

Selecting the cycle

Initiate the dialog — CYCL DEF ↓ or GOTO □ 7 ENT

CYCL DEF 7 DATUM SHIFT	ENT Confirm the selected cycle.

Entering the value

SHIFT ?	X Select the axis. Enter the coordinates of the new datum. Y The datum shift is possible in all 5 axes. ENT When shifting in several axes, only confirm entry after entering all the coordinates!

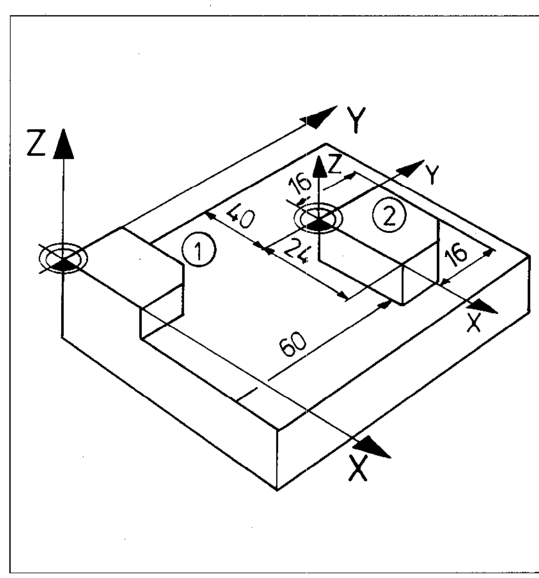
Example

A machining task is to be carried out as a subprogram

a) referred to the set datum X+0/Y+0 and

b) additionally referred to the shifted datum X+40/Y+60.

```
TOOL DEF 1 L0 R5
TOOL CALL 1 Z S200

CALL LBL 1              Without datum shift ①

CYCL DEF 7.0 DATUM SHIFT
CYCL DEF 7.1 X+40
CYCL DEF 7.2 Y+60

CALL LBL 1              With datum shift ②

CYCL DEF 7.0 DATUM SHIFT
CYCL DEF 7.1 X+0        Datum shift reset
CYCL DEF 7.2 Y+0

L Z+50 FMAX M02
```



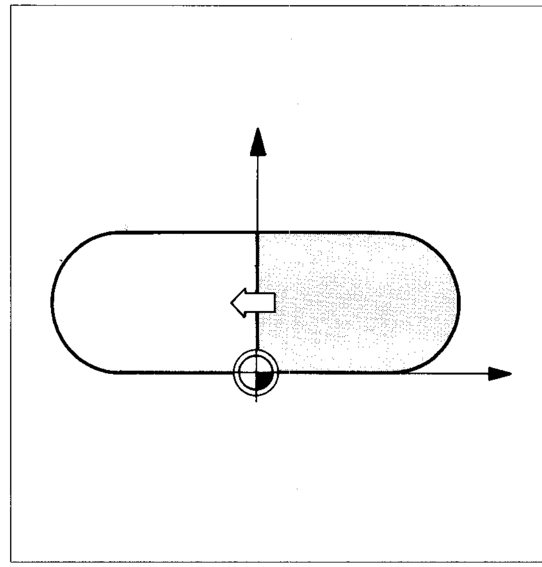
Subprogram

```
LBL 1
L X–10 Y–10 R0 FMAX M03
L Z+2 FMAX
L Z–5 F100
L X+0 Y+0 RL F500
L Y+20
L X+25
L X+30 Y+15
L Y+0
L X+0
L X–10 Y–10 R0
L Z+2 FMAX
LBL 0
```

Coordinate Transformations
Cycle 8: Mirror image

The cycle

The direction of an axis is reversed when it is mirrored. The sign is reversed for all coordinates of this axis. The result is a mirror image of a progammed contour or of a hole pattern.
Mirroring is only possible in the machining plane. You can mirror in one axis or both axes simultaneously.



Activation

The mirror image is immediately valid upon definition. The mirrored axes can be recognized by the highlighted axis designations in the status display for the datum shift.

Mirroring is performed at the current datum. The datum must therefore be shifted to the required position before a "mirror image" cycle definition.

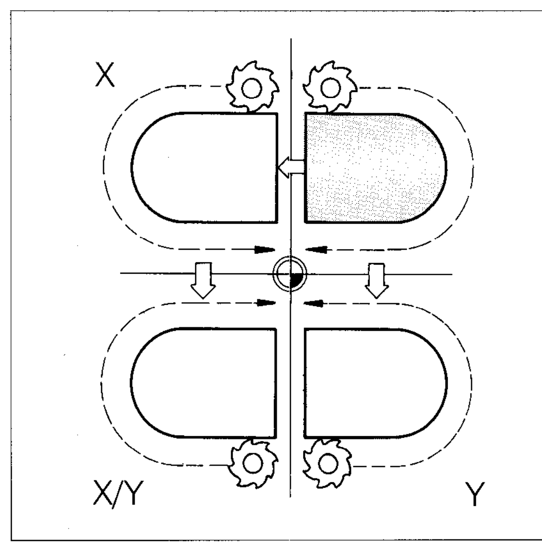
Mirrored axes

Enter the axis or axes to be mirrored.
The tool axis cannot be mirrored.

Climb and conventional milling

Mirroring one axis: The rotating direction is changed with the coordinate signs, so climb milling becomes conventional and vice versa.
The milling direction remains unchanged for fixed cycles.

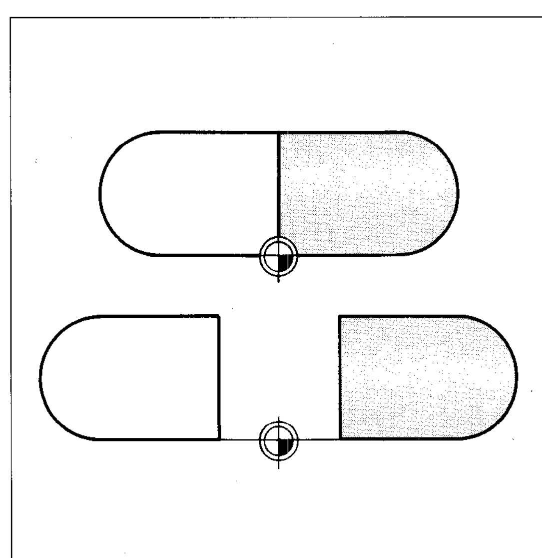
Mirroring two axes: The contour which was mirrored in one axis is mirrored a second time – in the other axis. The direction of rotation and e.g. climb milling remains the same.



X, Y = Axes to be mirrored

Datum position

1. If the datum is on the part contour, the part is only mirrored across the axis.
2. If the datum is outside the contour, the part is also moved!



Cancelling the mirror image

The mirror image cycle is cancelled by entering the mirror image cycle and responding to the dialog query "mirror image axis" with "NO ENT":

```
CYCL DEF 8.0 MIRROR IMAGE
CYCL DEF 8.1
```

Coordinate Transformations
Cycle 8: Mirror image

Selecting the cycle

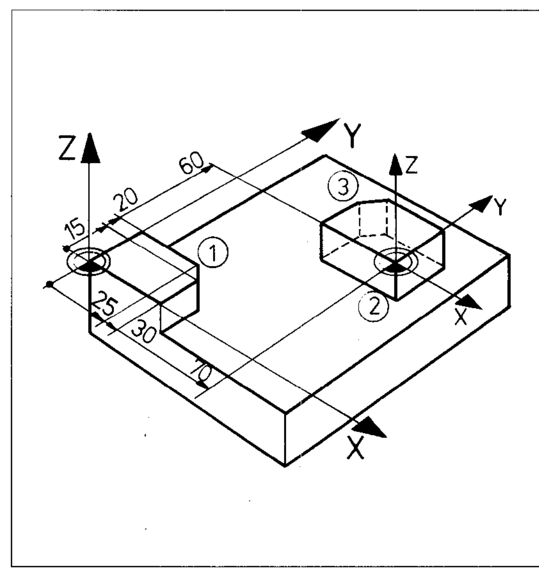
Initiate the dialog — CYCL DEF ↓ or GOTO □ 8 ENT

CYCL DEF 8 MIRROR IMAGE	ENT Confirm the selected cycle.

MIRROR IMAGE AXIS ?	X Enter the axis to be mirrored, e.g. X. Y Enter the second axis to be mirrored if applicable, e.g. Y. END □ Confirm the axes and always terminate the input with "END □".

Example

A program section (subprogram 1) is to be executed once – at position X+0/Y+0, and also mirrored once in X at position X+70/Y+60.



TOOL DEF 1 L+0 R5 TOOL CALL 1 Z S200	
CALL LBL 1 REP	Not mirrored ①
	Mirrored execution: sequence
CYCL DEF 7.0 DATUM SHIFT CYCL DEF 7.1 X+70 CYCL DEF 7.2 Y+60	Datum shift ②
CYCL DEF 8.0 MIRROR IMAGE CYCL DEF 8.1 X	Mirror image ③
CALL LBL 1	Subprogram call
CYCL DEF 8.0 MIRROR IMAGE CYCL DEF 8.1	Reset mirror image
CYCL DEF 7.0 DATUM SHIFT CYCL DEF 7.1 X+0 CYCL DEF 7.2 Y+0	Cancel datum shift
L Z+50 FMAX M02	Retract, return jump

Subprogram:

```
LBL 1
L X–10 Y–10 R0 FMAX M03
L Z+2 FMAX
L Z–5 F100
L X+0 Y+0 RL F200
L Y+20
L X+25
L X+30 Y+15
L Y+0
L X+0
L X–10 Y–10 R0
L Z+2 FMAX
LBL 0
```

Note

For correct machining according to the drawing, it is absolutely necessary that the sequence of cycles shown in the above execution be retained!

Coordinate Transformations
Cycle 10: Coordinate system rotation

The cycle

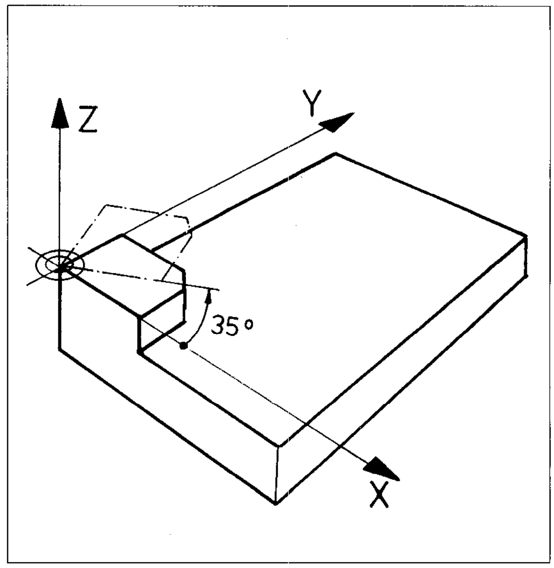
The coordinate system can be rotated in the machining plane about the current datum in a program.

Activation

Rotation is effective without being called and is also active in the operating mode "Positioning with MDI".

Rotation

To rotate the coordinate system, you only have to enter the rotation angle ROT.

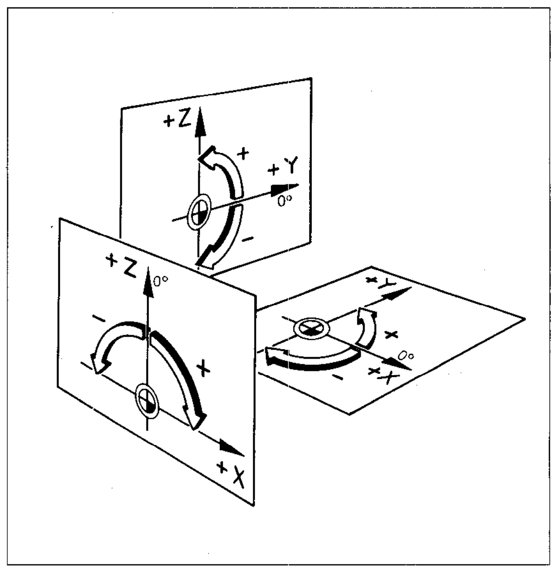


Planes

XY plane: +X axis = 0° (standard)
YZ plane: +Y axis = 0°
ZX plane: +Z axis = 0°

All coordinate inputs following the rotation are then referenced to the rotated coordinate system.

The rotation angle is entered in degrees (°). Input range: –360° to +360° (absolute or incremental).



Activating the rotation

CYCL DEF 10.0 ROTATION
CYCL DEF 10.1 ROT+35

The active rotation angle is indicated by "ROT" in the status display.

Cancelling the rotation

A rotation is cancelled by entering the rotation angle 0°.

CYCL DEF 10.0 ROTATION
CYCL DEF 10.1 ROT+0

Coordinate Transformations
Cycle 10: Coordinate system rotation

Selecting the cycle

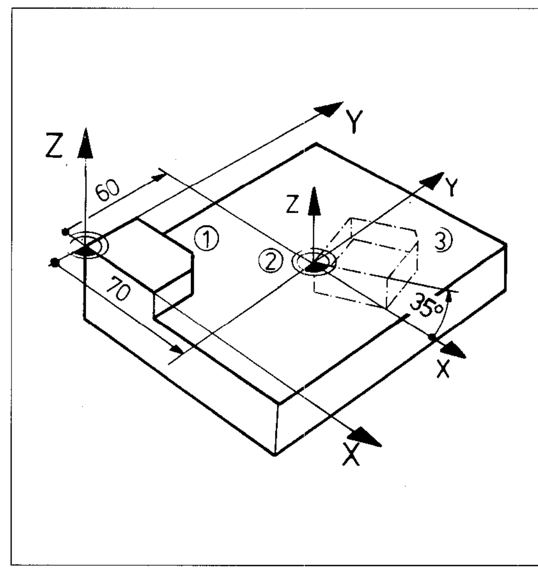
Initiate the dialog — CYCL DEF, ↑ or GOTO □ 1 0 ENT

CYCL DEF 10 ROTATION	ENT Confirm the selected cycle.

ROTATION ANGLE ?	Enter the rotation angle.
	I Incremental/absolute?
	ENT Confirm entry.

Example

A program section (subprogram 1) is to be executed:
once based on datum X+0/Y+0,
a second time based on datum X+70 Y+60.



TOOL DEF 1 L0 R5 TOOL CALL 1 Z S200	
CALL LBL 1	Non-rotated execution ①
CYCL DEF 7.0 DATUM SHIFT CYCL DEF 7.1 X+70 CYCL DEF 7.2 Y+60	Rotated execution. Sequence: 1. Datum shift ②
CYCL DEF 10.0 ROTATION CYCL DEF 10.1 ROT+35	2. Rotation ③
CALL LBL 1	3. Subprogram call
CYCL DEF 10.0 ROTATION CYCL DEF 10.1 ROT 0	Reset rotation
CYCL DEF 7.0 DATUM SHIFT CYCL DEF 7.1 X+0 CYCL DEF 7.2 Y+0	Cancel datum shift
L Z+200 FMAX M02	Return jump to first block of the main program

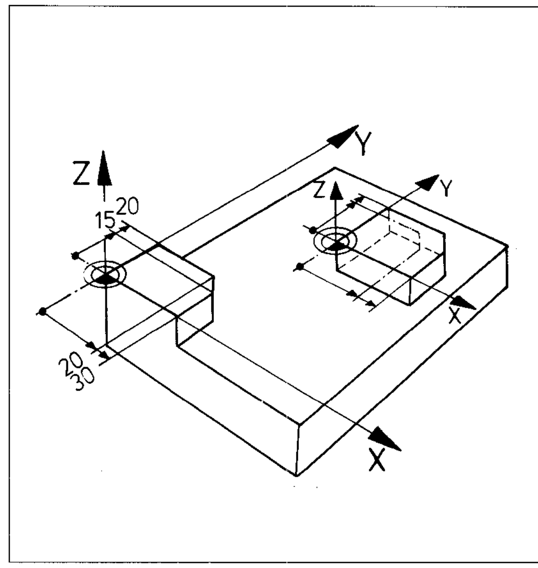
Subprogram

The associated subprogram (see cycle 7, Datum shift) is programmed after M02.

Coordinate Transformations
Cycle 11: Scaling

The cycle

Contours can be enlarged or reduced with this cycle. This permits generation of contours geometrically similar to an original without reprogramming, and also use of shrinkage and growth allowances.



Scaling is effective – depending on the specified machine parameters – either in the machining plane or in the three main axes (see index General Information, MOD Functions, User parameters).

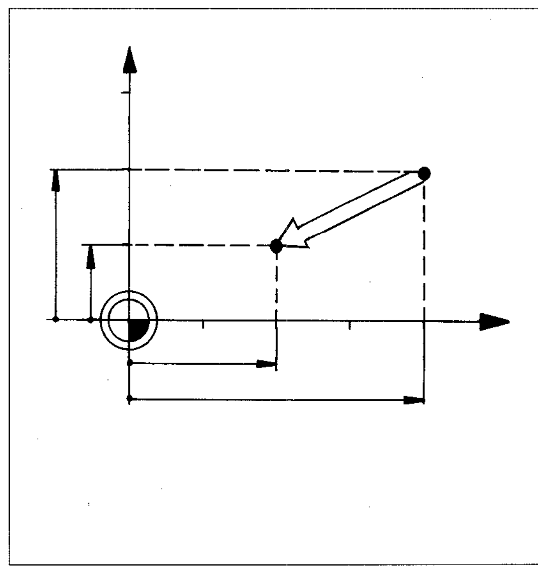
Activation

Scaling is effective immediately, without being called. Scaling factors greater than 1 result in magnification, factors between 0 and 1 result in reduction.

SCL factor

The scaling factor SCL (scaling) is entered to magnify or reduce a contour.
The control applies this factor to all coordinates and radii either in the machining plane or (depending on MP 7410; see index General Information, MOD Functions, User parameters) in all three axes X, Y and Z. The factor also affects dimensions in cycles.

Input range: 0.000001 to 99.999999.



Datum position

It is helpful to locate the datum on an edge of the subcontour. This way, the datum of the coordinate system is retained during a reduction or magnification as long as it is not subsequently moved or if the move is programmed before the scaling factor.

Activating scaling

```
CYCL DEF 11.0 SCALING
CYCL DEF 11.1 SCL 0.8
```

Cancelling scaling

The scaling cycle is cancelled by entering the factor 1 in the scaling cycle:

```
CYCL DEF 11.0 SCALING
CYCL DEF 11.1 SCL 1.0
```

Coordinate Transformations
Cycle 11: Scaling

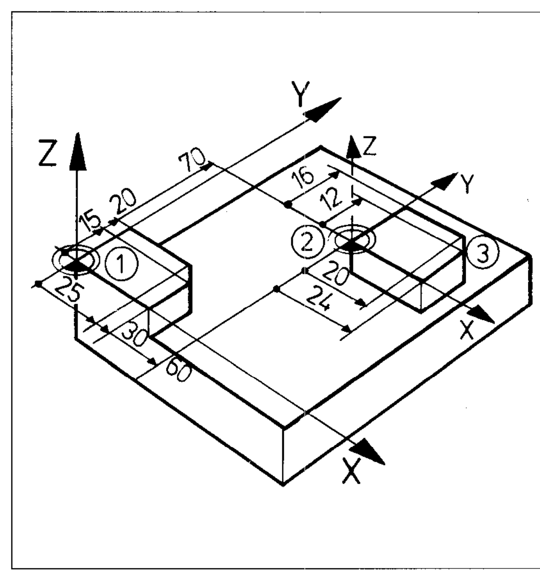
Selecting the cycle

Initiate the dialog — CYCL DEF ↑ or GOTO □ 1 1 ENT

CYCL DEF 11.0 SCALING	ENT Confirm the selected cycle.
FACTOR ?	□ Enter the scaling factor. ENT Confirm entry.

Example

A program section (subprogram 1) is to be executed one time based on the manually set datum X+0/Y+0, and one time based on X+60/Y+70 with the scaling factor 0.8.




```
TOOL DEF 1 L+0 R5
TOOL CALL 1 Z S200
```

CALL LBL 1	Execution in original size ①
CYCL DEF 7.0 DATUM	Execution with scaling factor. Sequence:
CYCL DEF 7.1 X+60 CYCL DEF 7.2 Y+70	1. Shift datum ②
CYCL DEF 11.0 SCALING CYCL DEF 11.1 SCL 1.0	2. Define scaling factor ③
CALL LBL 1	3. Call subprogram (scaling factor effective)
CYCL DEF 11.0 SCALING CYCL DEF 11.1 SCL 1.0 CYCL DEF 7.0 DATUM CYCL DEF 7.1 X+0 CYCL DEF 7.2 Y+0	Cancel transformations
L Z+200 FMAX M02	Retract, return jump

Subprogram

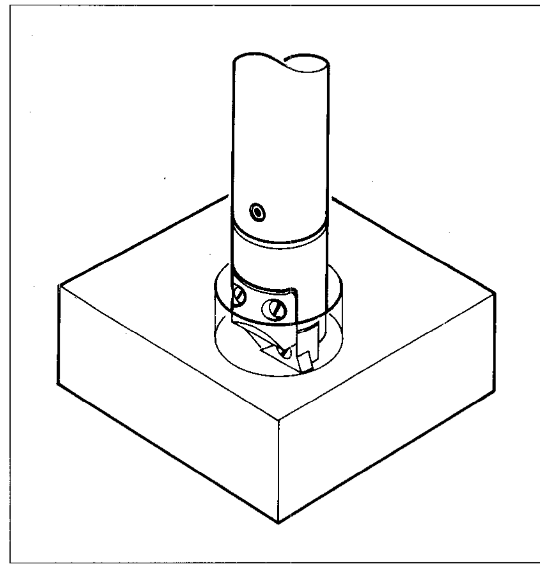
The corresponding subprogram (see cycle 7, Datum shift) is programmed after M02.

CYCL DEF

Other Cycles
Cycle 9: Dwell time

The cycle

In a program which is being run, the next block will be executed only after the end of the programmed dwell time. Modal conditions, such as spindle rotation, are not affected.



Activation

The dwell cycle is valid immediately upon definition, without being called.

CYCL DEF 9.0 DWELL TIME
CYCL DEF 9.1 DWELL 0.500

Possible applications

For example, chip breaking can easily be programmed with a dwell cycle after every drilling step.

Input range

The dwell time is specified in seconds.
Input range: 0 to 30000 s (≙ 8.3 hours)

Cycle definition

Initiate the dialog — CYCL DEF ↑ or GOTO □ 9 ENT

CYCL DEF 9.0 DWELL TIME	ENT Confirm the selected cycle.
DWELL TIME IN SECS. ?	Enter desired dwell time, in seconds. ENT Confirm entry.

Other Cycles
Cycle 12: Program call

The cycles

Machining procedures that you have programmed – such as special drilling cycles, curve milling, or geometry modules – can be created as callable main programs and be used like fixed cycles. They can be called from any program with a cycle call. They can thus help speed up programming and improve safety, since you are using proven modules.

Cycle 12 PGM CALL

A callable program defined as a cycle becomes in essence a fixed cycle.

It can be called with

CYCL CALL (separate block) or

M99 (blockwise) or

M89 (modally).

Initiate the dialog CYCL DEF ↑ or GOTO □ 1 2 ENT

Entering the cycle selection

CYCL DEF 12 PGM CALL	ENT	Confirm the selected cycle.

PROGRAM NUMBER ?	□ ENT	Program number

Example

The callable program 50 is to be called from program 5.

Program:
BEGIN PGM 5 MM

.
.

CYCL DEF

CYCL DEF 12.0 PGM CALL
CYCL DEF 12.1 PGM 50

Definition:
"Program 50 is a cycle"

CYCL CALL

L X+20 Y+50 FMAX M99

Call program 50

.
.

END PGM 5 MM

Cross-reference Drilling with chip breaking

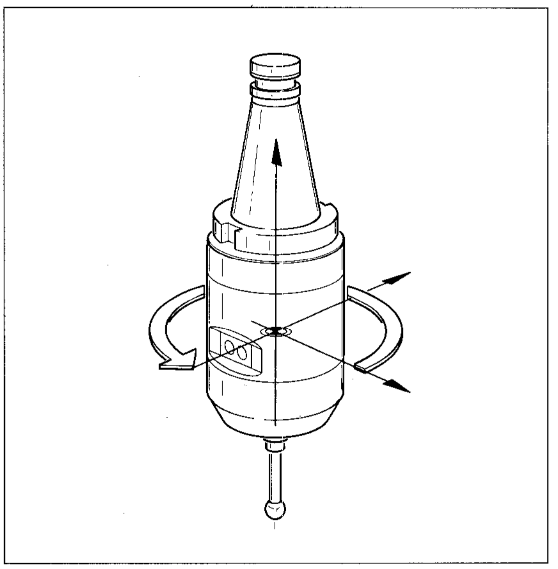
A realistic example of a program call with cycle 12 can be taken from the drilling example (Parameter programming PGM 7445):

1. Subprogram 1 is written separately as PGM 7444 (without LBL 1 or LBL 0).
2. PGM 7444 now exists as a callable, additional drilling procedure.
   This PGM can remain stored in the control and be called by any other program, e.g. 7445.
3. Subprogram 1 is deleted in the main program 7445.
4. Instead of CALL LBL 1, write in PGM 7445:
   CYCL DEF 12 PGM CALL 7444, and M99 in a subsequent positioning block.

Other Cycles
Cycle 13: Oriented spindle stop

The cycle

The control can address the machine tool spindle as a 4 or 5^{th} axis and turn it to a certain angular position.



Applications:

- Tool changing systems with defined change position for tool.
- Orientation of the transmitter/receiver window of the TS 511 3D-touch probe system from HEIDENHAIN.

Activation M19

The cycle – if provided on the machine – is executed through **M19.** The spindle orientation is activated either through

- machine parameter or
- cycle 13: spindle orientation.

If the cycle is called without prior definition, the spindle will be oriented to an angle which has been set in the machine parameters. Further information is available from the machine tool builder.

Input range

The angle of orientation is entered according to the reference axis of the working plane.
Input range: 0 to 360°.
Input resolution: 0.1°.

Cycle definition

Initiate dialog — CYCL DEF, ↑ or GOTO □, 1, 3, ENT

CYCL DEF 13.0 ORIENTATION	ENT Confirm the selected cycle.
ORIENTATION ANGLE ?	□ Enter new angle for spindle.
	ENT Confirm entry.

Example

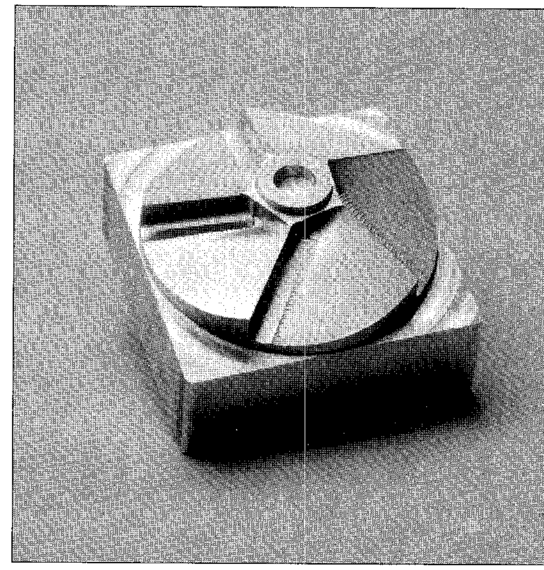
CYCL DEF 13.0 ORIENTATION
CYCL DEF 13.1 ANGLE 45

Parametric Programming
Overview

Parametric programming

Parametric programming expands the capabilities of the control enormously and offers features such as:

- Variable drilling programs
- Processing of mathematical curves (e.g.: sine wave, ellipse, parabola, hyperbola)
- Programs for machining families of parts
- 3D programming for mold making



Basic functions

The mathematical and logical functions listed at the right are available for programming.

FN 0:	ASSIGN
FN 1:	ADDITION
FN 2:	SUBTRACTION
FN 3:	MULTIPLICATION
FN 4:	DIVISION
FN 5:	SQUARE ROOT
FN 6:	SINE
FN 7:	COSINE
FN 8:	ROOT-SUM OF SQUARES
FN 9:	IF EQUAL, JUMP
FN 10:	IF UNEQUAL, JUMP
FN 11:	IF GREATER, JUMP
FN 12:	IF LESS, JUMP
FN 13:	ANGLE
FN 14:	ERROR CODE

Computation time

The time required for one computing step – depending on the workload on the processor – can reach the millisecond range.

For this reason, very many computations and very small displacements may cause the machine axes to be halted. In this case you have to make a compromise between high surface definition (many computations, small displacements) and efficient machining.

Variable addresses with parameters

The program data shown at the right can be kept variable by using the Q parameters:
Enter a Q parameter instead of a specific number.

Example for variable positioning:
instead of L X+20.25 you write L X+Q21.

The parameter value for Q21 must be computed in the program or be defined before it is called.

Nominal positions	L X+Q21 Y+Q22
Circle data	CC X+Q1 Y+Q2 C X+Q10 Y+Q20 CT X+Q11 Y+Q21 RND Q1 CR X+Q21 Y+Q22 R Q62
Feed rate	F Q10
Tool data	TOOL DEF 1 L+Q1 R Q2 TOOL CALL Q5 Z S Q6
Conditional jump	IF+Q10 GT+0 GOTO LBL Q30
Cycle data	CYCL DEF 1.0 PECKING SET UP –Q1/DEPTH –Q2 PECKG –Q3 DWELL Q4/F Q5

Inch dimensions

Programs using parameters as jump address (e.g. GOTO LBL Q10) are not to be switched from mm to inches or vice versa, because the contents of the Q parameters are also converted during switchover, which would result in false jump addresses.

Parametric Programming
Selection

Selecting basic functions
Q

After pressing "Q", the functions can be selected either with the vertical cursor keys or with "GOTO □", the associated function number and "ENT".

Defining parameters

A parameter is designated by the letter Q and any number between 0 and 99.

Specific numerical values (contents) can be allocated to the parameter either directly or with mathematical and logical functions. Parameter contents can also have a negative sign. Positive signs need not be programmed.

Starting values

Parameters must be defined before they can be used. When program run is started, all parameters are automatically assigned the value 0 if machine parameter MP 7300 = 0. If the parameters are to be assigned values before program start, set MP 7300 = 1. The parameter values are then not deleted at program start.

Examples of defined parameters:
Q1 = +1.5
Q5 = +Q1
Q9 = +Q1 * +Q5

Notation

The notation corresponds to the standard computer format:
The operands and the operator are on the right, the desired result on the left. Consider the entire line as a mathematical operation and not as an equation!

Here also use the "ENT" key to continue the dialog within one program line.

Example

The following multiplication is to be entered:

$Q10 = Q5 \cdot 1.7$

Initiate the dialog Q

FN 0: = ASSIGN	Q ↓ or GO TO 3 ENT	Select multiplication (operation).
FN 3: MULTIPLICATION	ENT	Entry into the function.
PARAMETER NUMBER FOR RESULT?	10 ENT	Parameter for result.
FIRST VALUE OR PARAMETER?	Q 5 ENT	1st operand (parameter).
SECOND VALUE OR PARAMETER?	1.7 ENT	2nd operand.
FN 3: Q10 = +Q5 * +1.7		Finished program line.

Q10 is assigned the result when the operation is executed; the contents of Q5 are retained.

Parametric Programming
Algebraic functions

FN 0: Assignment	This function assigns a parameter either a numerical value or another parameter. The assignment corresponds to an equal sign.	Example: FN 0: Q5 = +65.432 Q5 = +Q12 Q5 = −Q13
FN 1: Addition	This function defines a certain parameter to be the sum of two parameters, two numbers or one parameter and one number.	FN 1: Q17 = +Q2 + +5 Q17 = +5 + +7 Q17 = +5 + −Q12 Q17 = −Q4 + +Q8 Q17 = +Q17 + +Q17
FN 2: Subtraction	This function defines a certain parameter to be the difference between two parameters, two numbers or one parameter and one number.	FN 2: Q11 = +5 − +Q34 Q11 = +5 − +7 Q11 = +5 − −Q12 Q11 = +Q4 − +Q8 Q11 = +Q11 − −Q11
FN 3: Multiplication	This function defines a certain parameter to be the product of two parameters, two numbers or one parameter and one number.	FN 3: Q21 = +Q1 * +60 Q21 = +5 * +7 Q21 = +5 * −Q12 Q21 = +Q4 * −Q8 Q21 = +Q21 * +Q21
FN 4: Division	This function defines a certain parameter to be the quotient of two parameters, two numbers or one parameter and one number. **Division by 0 is not permitted!**	FN 4: Q12 = +Q2 DIV +62 Q17 = +5 DIV +7 Q17 = +5 + DIV −Q12 Q17 = +Q4 DIV +Q8
FN 5: Square root	This function defines a certain parameter to be the square root of one parameter or one number. The operand must be positive.	FN 5: Q98 = SQRT +2 Q98 = SQRT +Q12 Q98 = SQRT −Q70

Sign for operands

Parameters with negative signs can be used in equations.

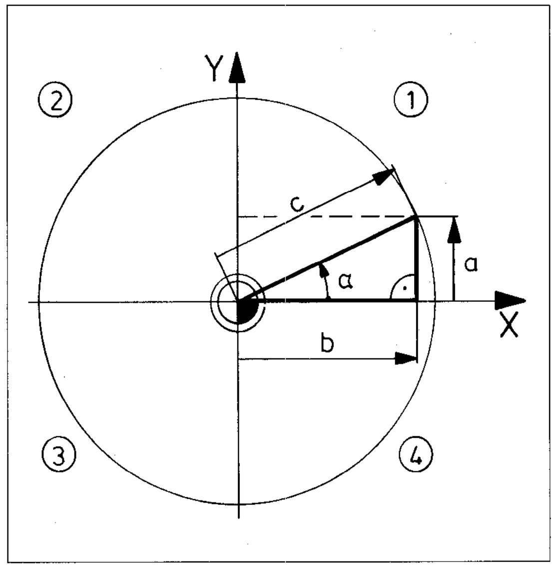
Q11 = +5 − −Q34

E.G. subtraction can be obtained from an addition and vice versa. This also applies for other operations.

Parametric Programming
Trigonometric functions

Basics of trigonometry

A circle with radius c is divided symmetrically into four quadrants ① to ④ by the two axes X and Y. If the radius c forms the angle α with the X-axis, the two components a and b of the right-angled triangle depend upon angle α.



Defining the trigonometric functions

$$\sin\alpha = \frac{\text{opposite side}}{\text{hypotenuse}} = \frac{a}{c} \quad \text{or } a = c \cdot \sin\alpha$$

$$\cos\alpha = \frac{\text{adjacent side}}{\text{hypotenuse}} = \frac{b}{c} \quad \text{or } b = c \cdot \cos\alpha$$

$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha} = \frac{\text{opposite side}}{\text{adjacent side}} = \frac{a}{b}$$

Length of one side

According to the Pythagorean theorem:
$c^2 = a^2 + b^2$ or $c = \sqrt{a^2 + b^2}$

Table for preceding sign and angle range

Function	Quadrant ①	②	③	④
sin α	+	+	−	−
cos α	+	−	−	+
tan α	+	−	+	−
Angle	0° – 90°	90° – 180°	180° – 270°	270° – 360°

FN 6: Sine

A parameter is defined as the **sine** of an angle, whereby the angle can be a number or a parameter (unit of measurement of the angle: degrees).
Q44 = sin Q11

FN 6: Q44 = SIN + Q11

FN 7: Cosine

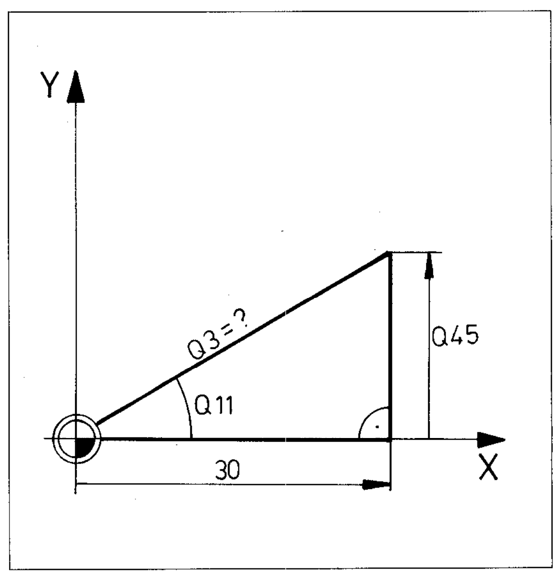
A parameter is defined as the **cosine** of an angle, whereby the angle can be a number or a parameter (unit of measurement of the angle: degrees).
Q81 = cos Q11

FN 7: Q81 = cos + Q11

FN 8: Root sum of squares

A parameter is computed as the **square root** of the sum of squares of two numbers or parameters (LEN = length).
$Q3 = \sqrt{Q45^2 + 30^2}$

FN 8: Q3 = +Q45 LEN+30



Parametric Programming
Trigonometric functions

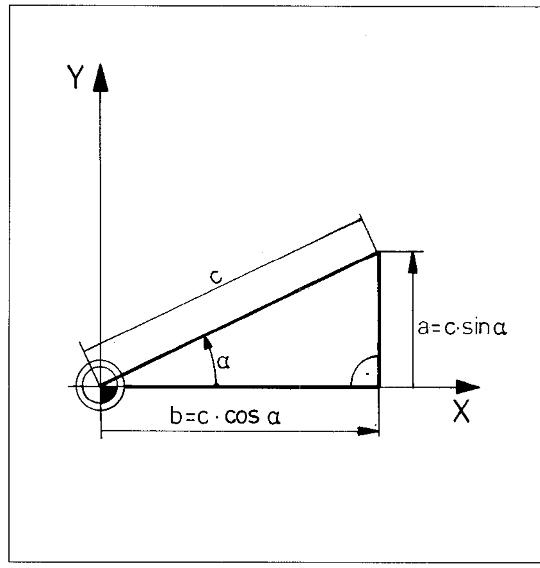
Angles from line segments or trigonometric functions

According to the definitions of the angular functions, either the angular functions sin α and cos α, or the lengths of sides a and b can be used to determine tan α:

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha} = \frac{a}{b} \ .$$

The angle α is therefore

$$\alpha = \arctan \frac{\sin \alpha}{\cos \alpha} = \arctan \frac{a}{b} \ .$$



Unambiguous angle

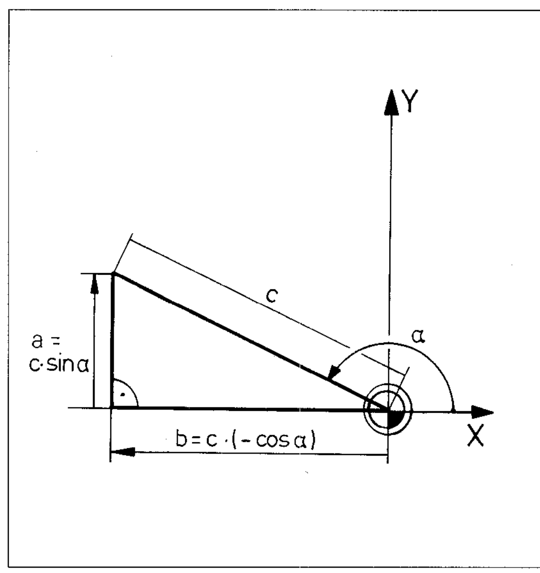
If the value of sin α or the side a is known, two possible angles always result:

Example: $\sin \alpha = 0.5$
$\alpha_1 = +30°$ and $\alpha_2 = +150°$.

To determine angle α unambiguously, the value for cos α or side b is required. If this value is known, an **unambiguous** angle α is the result:

Example: $\sin \alpha = 0.5$ and $\cos \alpha = 0.866$
$\alpha = +30°$,

$\sin \alpha = 0.5$ and $\cos \alpha = -0.866$
$\alpha = +150°$.



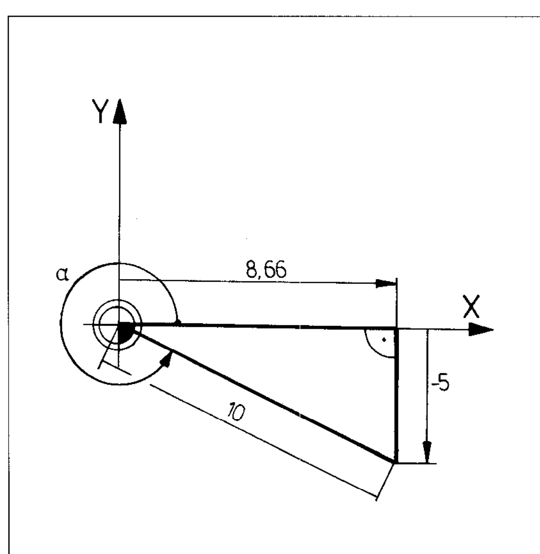
FN 13: Angle

This function assigns a parameter the **angle** from a sine and cosine function, or from the two legs of the right-angled triangle.

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha} = \frac{a}{b} = \frac{-5}{8.66}$$

$$\alpha = \arctan \left[\frac{-5}{8.66}\right]$$

FN 13: Q11 = –5 ANG +8.66



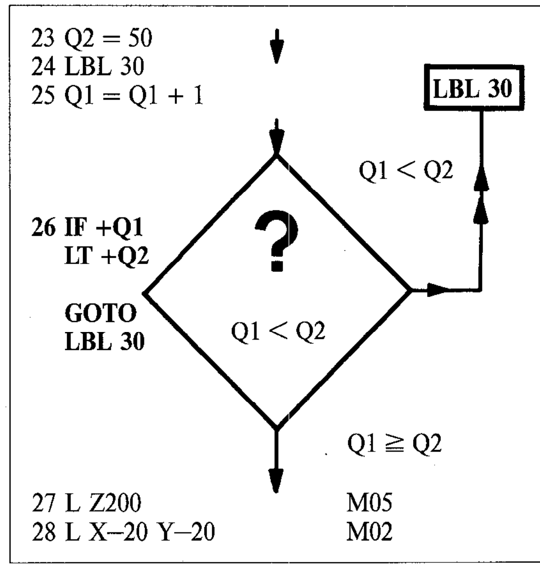
Parametric Programming
Conditional/unconditional jumps

IF:
IF-THEN
jump

With the parameter functions FN 9 to FN 12, you can compare one parameter with another parameter or with a given number (e.g. a maximum value).

Depending on the result of this comparison, a jump to a certain label in the program can be programmed (conditional jump):

If the programmed IF condition is fulfilled, a jump is performed; if the condition is not fulfilled, the next block (following IF ...) will be executed.



Program call

If you write a program call behind the called program label, a jump can be made to another program.
(Program calls are for example PGM CALL or cycle 12).

	Examples:	Decision criteria:
Equation FN 9: =	FN 9: IF + Q1 EQU + 360 GOTO LBL 30	A parameter is equal to a value or a second parameter, e.g. Q1 = Q2 or in the example: Q1 has the value 360.000.
Inequalities FN 10: ≠	FN 10: IF + Q1 NE + Q2 GOTO LBL 2	A parameter is not equal to a value or a second parameter, e.g. Q1 ≠ Q2
FN 11: >	FN 11: IF + Q1 GT + 360 GOTO LBL 17	A parameter is greater than a value or a second parameter, e.g. Q1 > Q2. Also possible: greater than zero, i.e. positive.
FN 12: <	FN 12: IF + Q1 LT + Q2 GOTO LBL 3	A parameter is less than a value or a second parameter, e.g. Q1 < Q2. Also possible: less than zero, i.e. negative.

Unconditional jumps

You can also program **unconditional jumps** to a label with the parameter functions FN 9 to FN 12.

Example:	Decision criterion:
FN 9: IF 0 EQU 0 GOTO LBL 30	The condition is **always** fulfilled, i.e. an **unconditional** jump is performed.

Abbreviations

EQU: equal to
NE: not equal to
GT: greater than
LT: less than

Parametric Programming
Special functions

FN 14: Error code

You can call error messages and dialog texts of the machine manufacturer from the PLC EPROM with FN 14. To call, enter the error code number between 0 and 499.
The error message terminates program run.
The program must be restarted after the error has been corrected.

The messages are allocated as follows:

Error code	Screen display
0 ... 299	ERROR 0 ... ERROR 299
300 ... 399	PLC ERROR 01 ... PLC ERROR 99 (or dialog determined by the machine tool manufacturer).
400 ... 483	DIALOG 1 ... 83 (or dialog determined by the machine tool manufacturer).
484 ... 499	USER PARAMETER 15 ... 0 (or dialog determined by the machine tool manufacturer).

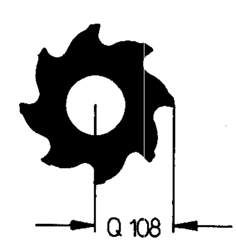
Example:
FN 14: ERROR = 100

Q100 – Q107

The control can transfer Q parameter values from the integrated PLC to a NC program. The parameters Q100 to Q107 are used for this.

Q108 Tool radius

The control always stores the tool radius of the last called tool in parameter Q108.
Thus, the active tool radius can be used for the radius compensation in parameter computations and comparisons.



Q109 Tool axis

The control stores the current tool axis in parameter Q109:
Different machines alternately use the X,Y or Z axis as the tool axis.
On these machines it is helpful when the current tool axis can be requested in the machining program; this makes program branching in user cycles possible.

Current tool axis	Parameter
no tool axis called	Q109 = –1
X axis is called	Q109 = 0
Y axis is called	Q109 = 1
Z axis is called	Q109 = 2

Parametric programming
Special functions

Q110 Spindle on/off

The value in parameter Q110 specifies the last M function issued for the direction of spindle rotation:

M function	Parameter
no M spindle function	Q110 = –1
M03 spindle on clockwise	Q110 = 0
M04 spindle on counterclockwise	Q110 = 1
M05, if M03 was previously issued	Q110 = 2
M05, if M04 was previously issued	Q110 = 3

Q111 Coolant on/off

Parameter Q111 indicates whether the coolant was switched on or off.

Meaning:	Parameter
M08 coolant switched on	Q111 = 1
M09 coolant switched off	Q111 = 0

Q112 Overlap factor

Parameter Q112 contains the overlap factor for pocket milling (see index General Information, MOD Functions, User parameters, MP 7430).

The overlap factor for pocket milling can be used to advantage in milling programs.

Q113 mm/inch dimensions

Parameter Q113 specifies whether the NC program at the highest program level (in cases of sub-programming with PGM CALL) contains mm or inch dimensions.

Meaning:	Parameter
mm dimensions	Q113 = 0
inch dimensions	Q113 = 1

Parameters for programmable probing function: Q115 to Q118

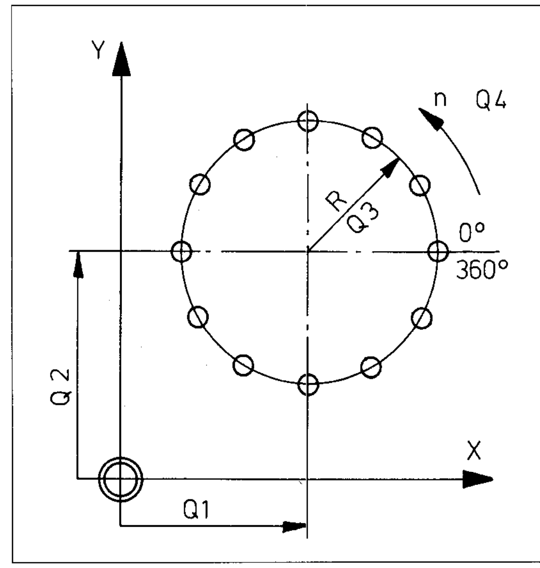
Parameters Q115 to Q118 contain the **uncompensated** position measurements (i.e. length and radius of the stylus are neglected) that were acquired with the programmable probing function "surface = datum":

Measurement:	Parameter
X axis	Q115
Y axis	Q116
Z axis	Q117
4^{th} axis	Q118

Parametric Programming
Example: Bolt-hole circle

Task

A bolt-hole circle is to be drilled using the pecking cycle in the XY plane.



Example

Radius R of the bolt-hole circle:
Q3 = 35 mm.

Number n of bore holes:
Q4 = 12.

X coordinate of the bolt circle center:
Q1 = 50 mm.

Y coordinate of the bolt circle center:
Q2 = 50 mm.

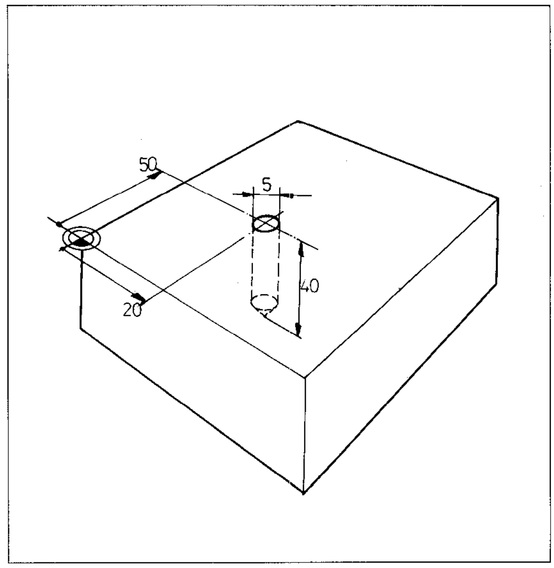
Program

Assigning values	FN0: Q1 = +50 FN0: Q2 = +50 FN0: Q3 = +35 FN0: Q4 = +12	Center in X Center in Y Bolt circle radius Number of bore holes
	TOOL DEF / TOOL CALL	Define and call tool
	CYCL DEF PECKING	Select and load drilling cycle
	FN0: Q10 = +0	Set starting angle
Computation	FN4: Q14 = +360 DIV+Q4	Compute angle increment
	L Z+2 R0 FMAX M03	Approach setup clearance and switch on spindle
	CC X+Q1 Y+Q2	
Execution	LP PR Q3 PA+Q10 FMAX M99	1st bore
	LBL 1	Start of loop
	FN1: Q10 = +Q10 + Q14	Angle increment
	FN9: IF+Q10 EQU+360 GOTO LBL 2	
	LP PA+Q10 FMAX M99	Further bores
	FN12: IF+Q10 LT+360 GOTO LBL 1	If not all holes are drilled, jump to the start of the loop.
	LBL 2	
	L Z+50 R0 FMAX M02	End of program

Parametric Programming
Example: Drilling with chip breaking

Example

Interruptable drilling procedure with automatic approach to the setup clearance and raising of the tool to break the chip for longer tool life.

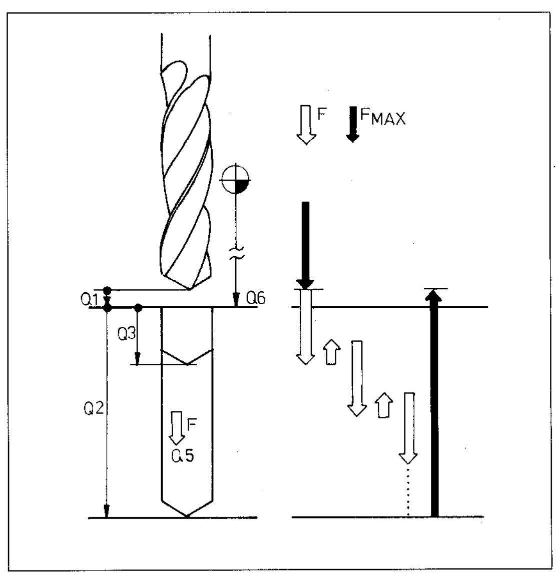


Main program

BEGIN PGM 7445 MM	
Q1 = –1	Setup clearance (incremental)
Q2 = –40	Depth (incremental)
Q3 = –5	Infeed (incremental)
Q4 = +0.5	Dwell time
Q5 = +200	Drilling feed rate
Q6 = +0	Work surface (absolute)
TOOL DEF 1 L+0 R2.5	Define tool
TOOL CALL 1 Z S200	Call tool, spindle speed
L X+20 Y+50 R0 FMAX M03	Approach drilling position
CALL LBL 1	Drilling
L Z+300 FMAX M02	End of main program

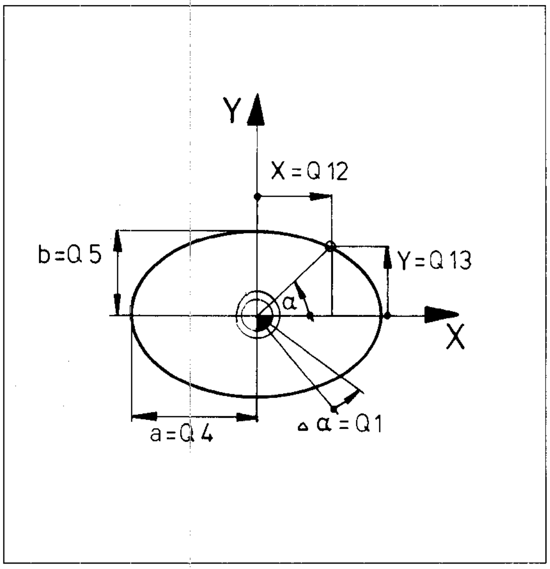
Subprogram 1: Drilling procedure

LBL 1	
FN 1: Q21 = +Q6 + –Q1	Setup clearance (absolute)
FN 0: Q23 = +Q6	Current work surface (absolute)
FN 1: Q24 = +Q6 + +Q2	Final drilling depth (absolute)
L Z+Q21 R0 FMAX	Approach setup clearance in rapid traverse
LBL 10	
FN 1: Q23 = +Q23 + +Q3	Compute (new) drilling depth
FN 1: Q22 = +Q23 + –Q1	Compute (new) chip breaking height
FN 12: IF+Q23 LT+Q24 GOTO LBL 99	Drilling depth would not be attained
L Z+Q23 F Q5	Drilling
L Z+Q22	Chip breaking
FN 11: IF+Q23 GT+Q24 GOTO LBL 10	Another drilling step required?
LBL 99	
L Z+Q24 F Q5	Drill directly to final depth
CYCL DEF DWELL Q4	Clear base of bore
L Z+Q21 FMAX	Return to setup clearance
LBL 0	
END PGM 7445 MM	



Parametric Programming
Example: Ellipse as an SL cycle

Programming of a mathematical curve will be illustrated with an ellipse.

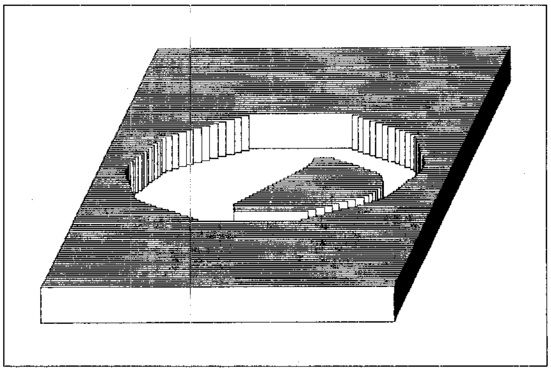


Geometry

An ellipse is defined according to the following formula (parameter form of the ellipse):
$X = a \cdot \cos \alpha$
$Y = b \cdot \sin \alpha$
a and b are referred to as the semiaxes of the ellipse.
Starting at 0° (Q2 = starting angle α_a) and increasing α in small increments (Q1 = incremental angles $\Delta\alpha$) to 360° (Q3 = end angle α_e), a multitude of points on an ellipse results. If these points are connected by short straight lines (see machining program, block 38), a closed contour is produced.

Process

The machining direction of the ellipse (counterclockwise) and the selected radius compensation **RL** produce an inside contour (pocket). The contour is contained in the subprogram with program section repeat.



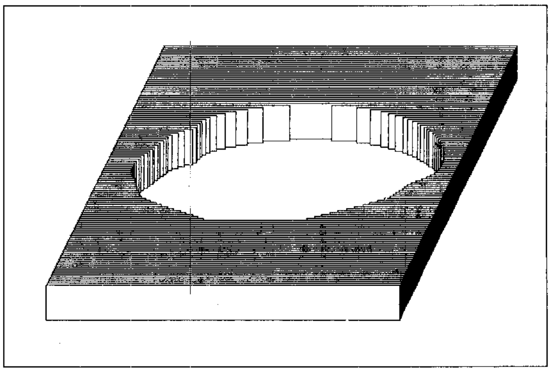
Roughing out

With the SL cycle "Contour", you can write a parameter program as an SL subprogram and execute this with the SL cycle "rough-out" by selecting an appropriate incremental angle.

Error message

TOO MANY SUBCONTOURS

If the incremental angle ($\Delta\alpha$ = Q0) selected for roughing out is too small, the control calculates too many short straight lines, which are interpreted as excessive subcontours.



Remedy

A relatively large incremental angle (e.g. Q0 = 10°) suffices for roughing out.

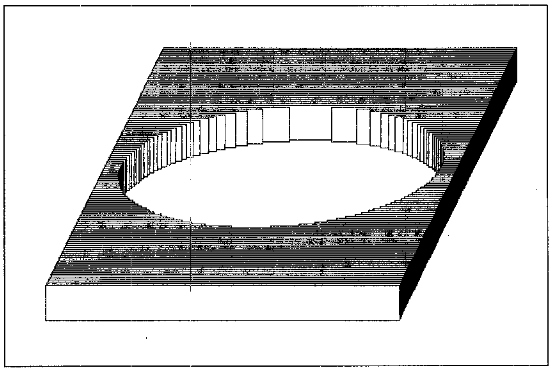
Finishing

For subsequent finishing, the subprogram is executed in the conventional manner with a finer incremental angle (e.g. Q1 = 1°).

Note

This program works with only one tool. It can be expanded to use a roughing cutter for "roughing out" and a finishing cutter for "finishing".

Also, a center-cut end mill (ISO 1641) is required or cycle 15 is to be applied for pilot drilling.



Parametric Programming
Example: Ellipse as an SL cycle

	0 BEGIN PGM 94152500 MM	
Parameter definition	1 FN 0: Q0 = +10	Incremental angle $\Delta\alpha$ for contour roughing
	2 FN 0: Q1 = +1	Incremental angle $\Delta\alpha$ for contour finishing
	3 FN 0: Q2 = +0	Starting angle α_a
	4 FN 0: Q3 = +370	End angle α_e*)
	5 FN 0: Q4 = +45	Semiaxis a
	6 FN 0: Q5 = +25	Semiaxis b
	7 FN 0: Q6 = +50	X coordinate for the datum shift
	8 FN 0: Q7 = +50	Y coordinate for the datum shift
	9 FN 0: Q8 = +2	Setup clearance Z
	10 FN 0: Q9 = –5	Pecking depth Z
	11 BLK FORM 0.1 Z X+0 Y+0 Z–10	
	12 BLK FORM 0.2 X+100 Y+100 Z+0	
	13 TOOL DEF 25 L+0 R+5	
	14 TOOL CALL 25 Z S1000	
	15 L Z+50 R0 FMAX M6	
	16 L Z+Q8 R0 FMAX M3	
	17 FN: 0 Q14 = Q2	Copy starting angle for counter
	18 CYCL DEF 7.0 DATUM	Datum shift
	19 CYCL DEF 7.1 X+Q6	
	20 CYCL DEF 7.2 Y+Q7	
	21 CYCL DEF 14.0 CONTOUR GEOM.	
	22 CYCL DEF 14.1 CONTOUR LABEL2	Define subprogram 2 – as contour label
	23 CYCL DEF 6.0 ROUGH-OUT	SL cycle rough-out
	24 CYCL DEF 6.1 SET UP –Q8 DEPTH +Q9	(for more information, see SL Cycles)
	25 CYCL DEF 6.2 PECKG –5 F100 ALLOW +2	
	26 CYCL DEF 6.3 ANGLE +45 F100	
Roughing out	27 CYCL CALL	Cycle call
	28 FN 0: Q0 = +Q1	Copy incremental angle for finishing
	29 FN 0: Q14 = Q2	Copy starting angle for counter
	30 L Z+Q9 F100	Drive tool to milling depth Z
Finishing	31 CALL LBL 2	Call subprogram 2
	32 L Z+50 R0 FMAX M2	Retract spindle axis, jump to start of program
Subprogram with program section repeat	33 LBL 2	
	34 FN 7: Q10 = COS+Q2	Computation of the X and Y positions on the elliptical path
	35 FN 6: Q11 = SIN+Q2	
	36 FN 3: Q12 = +Q10 * +Q4	
	37 FN 3: Q13 = +Q11 * +Q5	
	38 L X+Q12 Y+Q13 **RL** F200	Feed rate for finishing
	39 FN 1: Q = +Q + +Q0	Increase angle
	40 FN 12: IF +Q LT +Q3 GOTO LBL 2	If angle not attained, jump to LBL 2
	41 LBL 0	
	42 END PGM 94152500 MM	

*) End angle α_e is greater than 360°, so the contour is safely completed with the cutter.

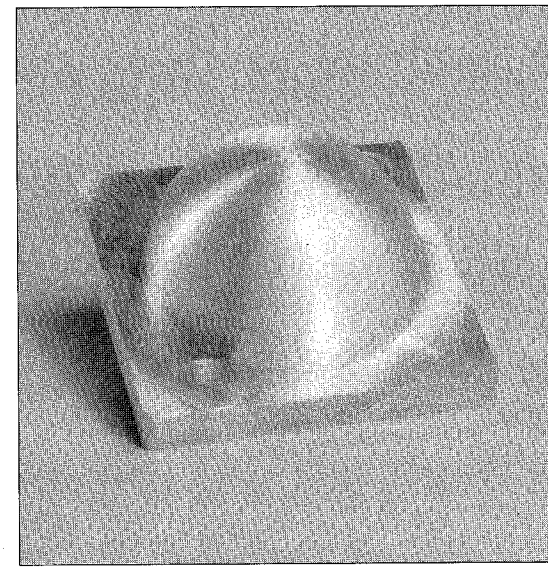
Modified program

If only the curve of the ellipse is to be milled, lines 1 and 21 – 27 are not needed.
Line 30 (drive tool to milling depth Z) is inserted behind line 38.

Parametric Programming
Example: Sphere

Task

Program 7513 machines a convex segment of a sphere using concentric circular movements in the horizontal plane.



Geometry

The size and location of the sphere can be entered.

You obtain a hemisphere when you select:

Starting 3D angle	Q1 = 0°
End 3D angle	Q2 = 90°
Starting plane angle	Q6 = 0°
End plane angle	Q7 = 360°

Cutting conditions

Cutting is performed during both the advance and return movements.
The following can be selected:
3D angle increment Q3
Downfeed rate Q11
Milling feed rate Q12

Note

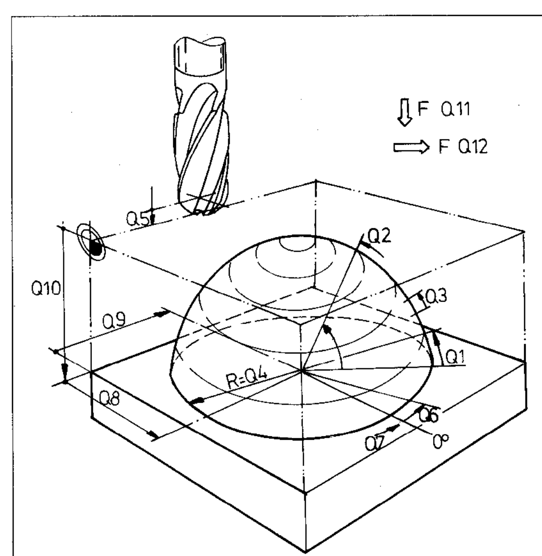
When selecting the 3D incremental angle, you have to make a compromise between the desired surface quality and the machining time. Small 3D incremental angles must be selected for high surface quality, but they require correspondingly long machining times.

Tool required

A spherical cutter is used for finishing.

Assigning values

0 BEGIN PGM 7816 MM	
1 FN 0 : Q1 = +10	Starting 3D angle
2 FN 0 : Q2 = +55	End 3D angle
3 FN 0 : Q3 = +2	3D incremental angle
4 FN 0 : Q4 = +50	Sphere radius
5 FN 0 : Q5 = +55	Setup clearance in Z
6 FN 0 : Q6 = +300	Starting plane angle
7 FN 0 : Q7 = +20	End plane angle
8 FN 0 : Q8 = +50	X sphere center
9 FN 0 : Q9 = +50	Y sphere center
10 FN 0 : Q10 = –40	Z sphere center
11 FN 0 : Q11 = +100	Downfeed rate
12 FN 0 : Q12 = +500	Milling feed rate



Blank

13 BLK FORM 0.1 Z X+0 Y+0 Z–50
14 BLK FORM 0.2 X+100 Y+100 Z+0

Tool

15 TOOL DEF 1 L+0 R+5
16 TOOL CALL 0 Z S 0

Change/ Start position

17 L Z+100 R0 F9999 M06
18 TOOL CALL 1 Z S 800

Subprogram call

19 CALL LBL 2

20 L Z+100 F9999 M02

Roughing

If roughing is required, an end mill can be used with a correspondingly larger sphere radius (Q4).

Parametric Programming
Example: Sphere

Setting the starting values	21 LBL 2	
	22 CYCL DEF 7.0 DATUM	Move datum to the sphere center
	23 CYCL DEF 7.1 X+Q8	
	24 CYCL DEF 7.2 Y+Q9	
	25 CYCL DEF 7.3 Z+Q10	
	26 CC X+0 Y+0	Set circle center
	27 FN 0 : Q20 = +Q1	Starting and current 3D angle
	28 FN 1 : Q31 = +Q4 + +Q108	Compensate sphere radius (with tool radius)
Starting position	29 CALL LBL 3	Compute starting position
	30 LP PR+Q17 PA+Q6 R0 F9999 M03	Approach starting position
	31 L Z+Q5	Approach setup clearance
	32 L Z+Q15 FQ11	Plunge cut at downfeed rate
	33 CP PA+Q7 DR+ FQ12	Circle segment to plane end angle
Program loop	34 LBL 1	
	35 FN 1 : Q20 = +Q20 + +Q3	3D angle increment
	36 FN 11 : IF +Q20 GT +Q2 GOTO LBL 99	If condition* is fulfilled, then jump to end
	37 CALL LBL 3	Position computation
	38 L Z+Q15 FQ11	Pilot positioning for withdrawal
	39 LP PR+Q17 PA Q20 FQ12	
	40 CP PA+Q6 DR– R0 FQ12	Return to plane starting angle
	41 FN 1 : Q20 = +Q20 + +Q3	3D angle increment
	42 FN 11 : IF +Q20 GT +Q2 GOTO LBL 99	If condition* is fulfilled, then jump to end
	43 CALL LBL 3	Position computation
	44 L Z+Q15 FQ11	Pilot positioning
	45 LP PR+Q17 PA Q20 R0 FQ12	
	46 CP PA+Q7 DR+ R0 FQ12	Arc to plane end angle
	47 FN 12 : IF +Q20 LT +Q2 GOTO LBL 1	If condition* is fulfilled, then jump to start of loop
End	48 LBL 99	
	49 L Z+Q5 R0 F9999	Finished, retract
	50 CYCL DEF 7.0 DATUM	Reset datum
	51 CYCL DEF 7.1 X+0	
	52 CYCL DEF 7.2 Y+0	
	53 CYCL DEF 7.3 Z+0	
	54 LBL 0	
Position computations	55 LBL 3	Computations
	56 FN 6 : Q14 = SIN +Q20	Z components
	57 FN 3 : Q15 = +Q14 * +Q31	
	58 FN 7 : Q16 = COS +Q20	Radius components
	59 FN 3 : Q17 = +Q16 * +Q31	
	60 LBL 0	
	61 END PGM 7816 MM	

* Condition: if current 3D angle Q20 is greater than or less than end 3D angle Q2, then jump to ...

Computation values

Q15: Current Z height
Q17: Current radius (polar radius)
Q20: Current 3D angle
Q31: Compensated contour radius
Q108: Current tool radius

Cycle sphere

The program can be used as a cycle:

1. Subprogram 2 (blocks 21 to 53) is written as a separate program.
2. Lines 21 and 54 are not required. Subprogram 3 (blocks 55 to 60) is written in place of block 29.
3. The user must only write the surrounding program (blocks 1 to 20) and then call the cycle in block 19 (PGM CALL).

Parametric Programming
Example: Sphere

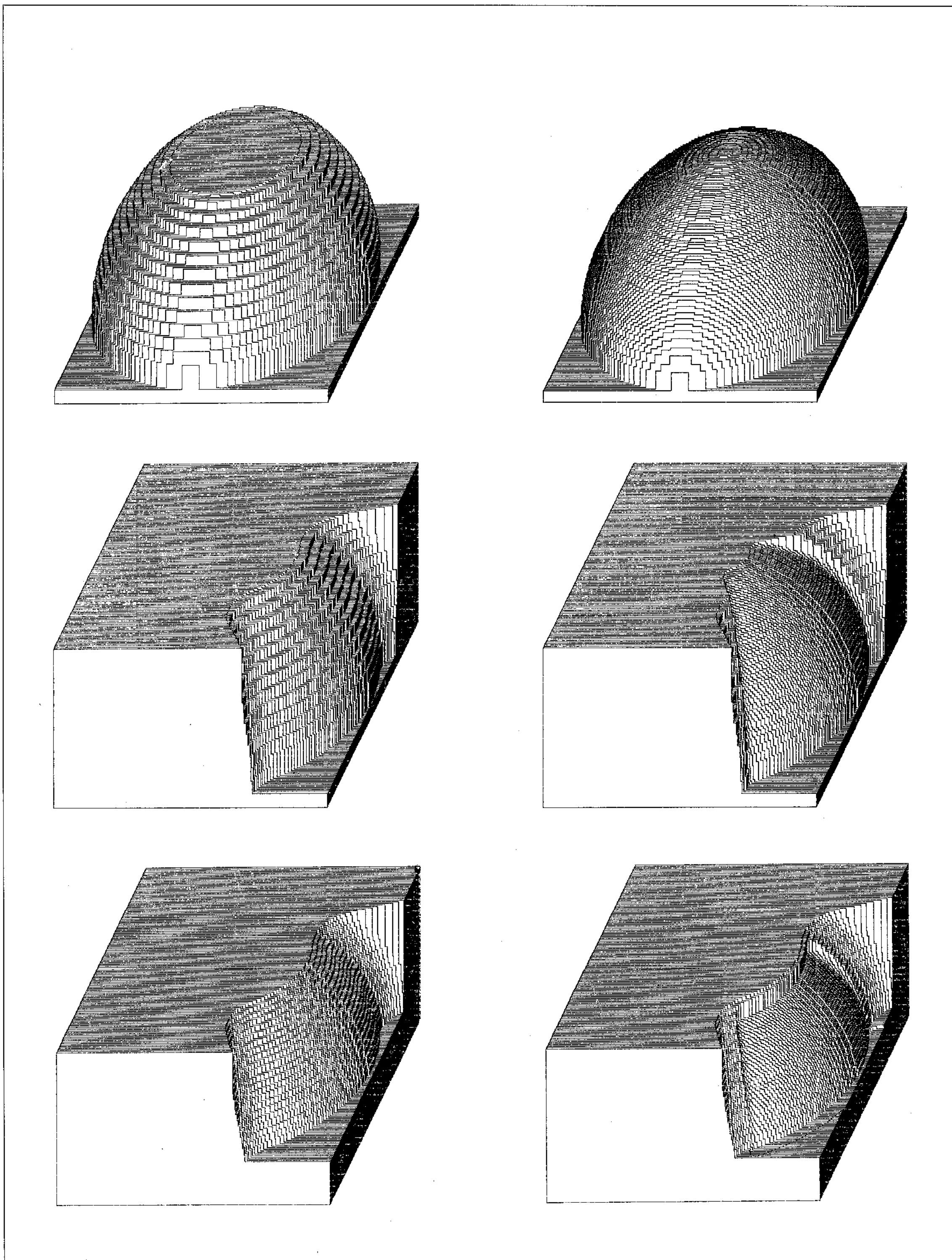
Machining sections of a hemisphere

Program 7816 can also be used to machine sections of a hemisphere by limiting the plane angles and 3D angles.

The graphics always shows the typical section for a cylindrical end mill.

Roughing
End mill, R = 12 mm,
3D angle increment 4°

Finishing
Spherical cutter, R = 3 mm,
3D angle increment 1°



Hemisphere:
3D angle
0° to 90°
Plane angle
0° to 360°

3D angle
0° to 90°
Plane angle
–60° to 20°

3D angle
10° to 55°
Plane angle
–60° to 20°

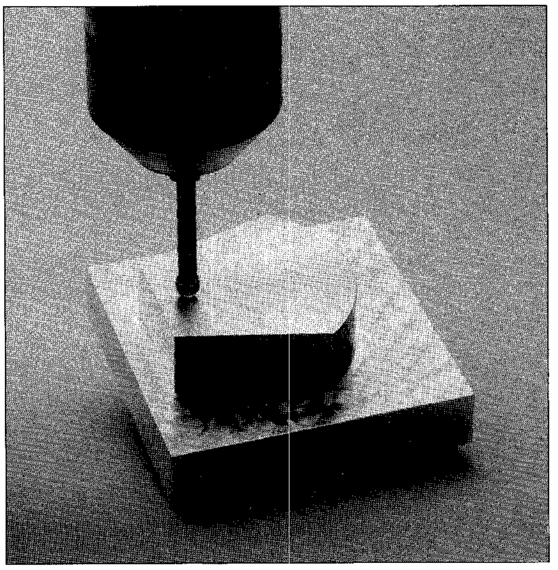
TOUCH PROBE

Q Q DEF

Programmed Probing
Overview

The programmable probing function allows you to perform measurements before or during machining. For example, the surfaces of cast parts with different heights can be probed, so that the correct depth is always attained during subsequent machining.

In addition, thermally-induced position deviations of the machine can be determined at selected time intervals and compensated.



Process

First, pilot position in rapid traverse while maintaining the setup clearance (machine parameter). Then probe with the probing axis at the measuring feed rate, transfer the probing position and retract to the setup clearance in rapid traverse. If the stylus is not deflected before reaching the maximum probing depth (machine parameter), the probing operation is aborted.

Initiate the dialog TOUCH PROBE

Input

PARAMETER NUMBER FOR RESULT ?	▢ ENT Parameter number
PROBING AXIS/PROBING DIRECTION ?	X +/− ENT Probing axis and probing direction
POSITION VALUE ?	X ▢ All coordinates of the pilot position Y ▢ ENT or incremental

Example

The probe is first to be pilot positioned to X–10, Y+20 and Z–20, and then probing begun with the X axis in positive direction. The probed result (X position) is to be stored in Q10.

Program

TOOL CALL 0 Z S0	
L Z+200 R0 FMAX M06	Tool change position
TCH PROBE 0.0 REF.PLANE Q10 X+	Probing with the X axis in positive direction, measuring result in Q10
TCH PROBE 0.1 X–10 Y+20 Z–20	Pilot positioning Q10 contains the compensated X axis measurement after probing

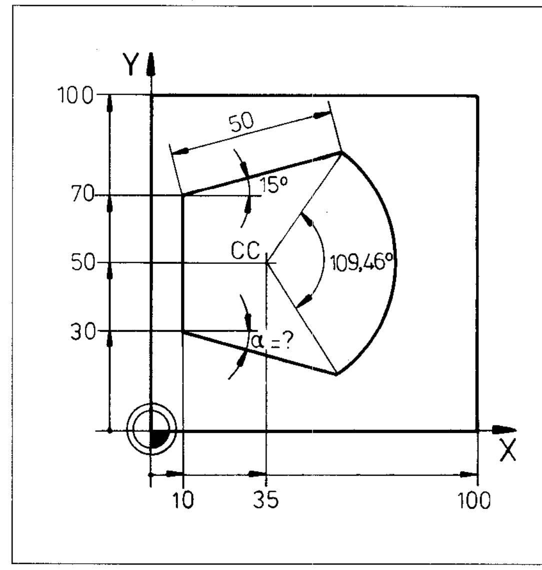
Q115 to Q118

Q115 to Q118 contain the actual, uncompensated values for X, Y, Z etc. after probing.

Programmed Probing
Example: Measuring length and angle

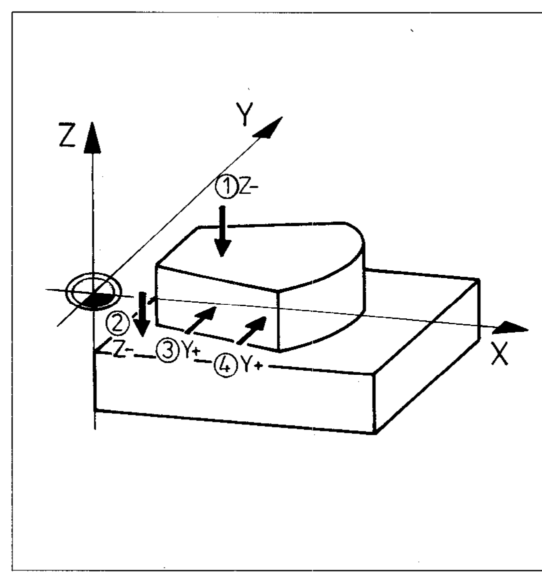
Task

A height (from the probing points ① and ②) and an angle (from the probing points ③ and ④) are to be measured with parameter programming.



Main program: Definition of probing points (pilot positioning)

0 BEGIN PGM 3DPROBE MM	
1 FN 0: Q11 = +20	Probing point ①
2 FN 0: Q12 = +50	X, Y, Z coordinates for
3 FN 0: Q13 = +10	pilot positioning
4 FN 0: Q21 = +20	Probing point ②
5 FN 0: Q22 = +15	
6 FN 0: Q23 = +0	
7 FN 0: Q31 = +20	Probing point ③
8 FN 0: Q32 = +15	Z coordinate Q33 valid
9 FN 0: Q33 = –10	for probing point ④
10 FN 0: Q41 = +50	Probing point ④
11 FN 0: Q42 = +10	
12 TOOL CALL 0 Z	
13 L Z+100 R0 F1000 M6	Retract, insert probe system



Measure length

14 TCH PROBE 0.0 REF.PLANE Q10 Z–	① Probe
15 TCH PROBE 0.1 X+Q11 Y+Q12 Z+Q13	
16 L Y+Q22	Approach auxiliary point
17 TCH PROBE 0.0 PROBE Q20 Z–	② Probe
18 TCH PROBE 0.1 X+Q21 Y+Q22 Z+Q23	
19 CALL LBL 1	Call subprogram 1

Measure angle

20 TCH PROBE 0.0 REF.PLANE Q30 Y+	③ Probe
21 TCH PROBE 0.1 X+Q31 Y+Q32 Z+Q33	
22 TCH PROBE 0.0 REF.PLANE Q40 Y–	④ Probe
23 TCH PROBE 0.1 X+Q41 Y+Q42 Z+Q33	
24 CALL LBL 2	Call subprogram 2
25 STOP	Check result parameter (see index "Machine Operating Modes", chapter "Program run", Checking/Changing the Q Parameters)
26 L Z+100 R0 F1000 M2	Retract, jump to start of program

TOUCH PROBE

Q | Q DEF

Programmed Probing
Example: Measuring length and angle

Subprogram 1: measure length

27 LBL 1
28 FN 2: Q1 = +Q20 – +Q10 — Measured **height** in parameter **Q1.**
29 LBL 0

Subprogram 2: measure angle

30 LBL 2
31 FN 2: Q34 = +Q40 – +Q30
32 FN 2: Q35 = +Q41 – +Q31
33 FN 13: Q2 = +Q34 ANG+Q35
34 FN 1: Q2 = –360 + +Q2 — Measured **angle** in parameter **Q2.**
35 LBL 0
36 END PGM 3DPROBE MM

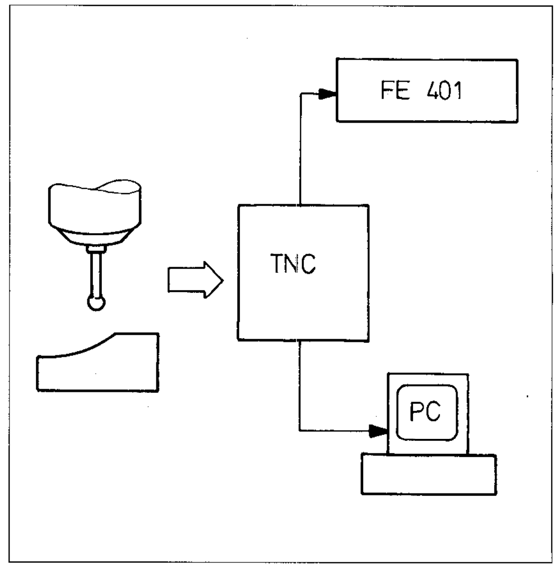
Digitizing 3D Contours
Overview

HEIDENHAIN digitizing software enables you to reduce a three-dimensional part model into discrete digital information by scanning it with the TS 120 triggering 3D touch probe. Thanks to the ability of this software to immediately access the position control loop of the TNC control, a series of 3 to 5 positions or more can be measured per second, depending on the machine. At a programmed point interval of 1 mm this results in scanning speeds of 180 to 300 mm/min (7 to 12 ipm).

Technical prerequisites

The following components are required to digitize a 3D contour with the HEIDENHAIN touch probe system:

- TS 120 triggering 3D touch probe
- Option for "digitizing with TS 120"
- External data storage medium: HEIDENHAIN FE 401 floppy disk unit or PC (IBM or IBM compatible) with HEIDENHAIN data transfer software **TNC.EXE**
- Evaluation software



The control must be specially prepared by the machine tool builder to enable the use of the 3D touch probe system.

The digitized positions are output in straight-line blocks in HEIDENHAIN format over the RS-232-C/V.24 data interface. Shorter programs can be stored in the FE 401 floppy disk unit, while longer programs must be stored in a PC.

Evaluation software for PCs (positive/negative forms, tool radius compensation etc.) is now being developed. Through the digitizing process, however, NC programs are immediately produced which are usable without additional evaluation if the cutter radius is equal to the probing ball radius.

Limitations

- The scanning cycles are programmed in the HEIDENHAIN conversational programming language. Programming in ISO is not planned.
- Only the main axes X, Y, Z may be used in the scanning cycles (no parallel axes U, V, W).
- Basic rotation and coordinate transformation (datum shift, mirror image, rotation, scaling) must be inactive when the scanning cycles are called.

Digitizing 3D Contours
Defining the scanning range

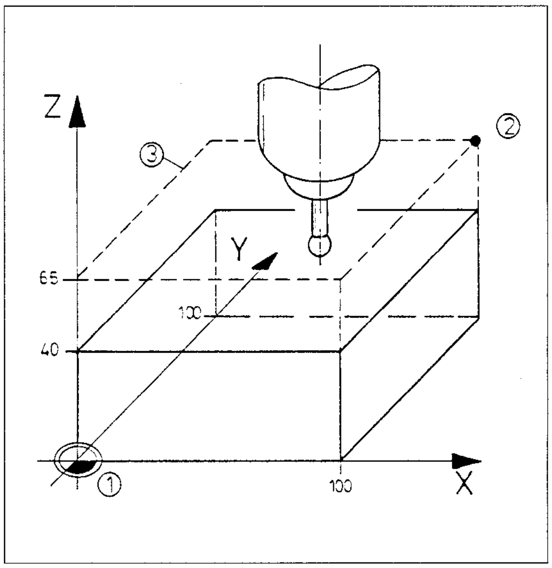
The digitizing functions comprise the following three cycles:

TOUCH PROBE 5.0: RANGE
TOUCH PROBE 6.0: MEANDER
TOUCH PROBE 7.0: CONTOUR LINES

Axis
MIN point
MAX point

With the scanning cycle **RANGE,** a cuboid is defined by entering coordinates of two points, the **minimum point (MIN)** and the **maximum point (MAX).** The contour to be scanned lies within this space.

This range is programmed similarly to the BLK FORM, but with additional input of a **program name** and the **clearance height.**



PGM NAME

The program name (max. 8 characters) – which is also the file name for the external memory – indicates the NC program in which the measured positions are stored in the form of straight-line blocks. The program name **must** be entered. If different names are assigned in several range definitions within one scanning program, then a corresponding number of NC programs will automatically be generated for storage of the measured positions.

Clearance height

Input of a height, in absolute dimensions, at which the stylus cannot collide with the contour to be scanned.

Digitizing 3D Contours
Defining the scanning range

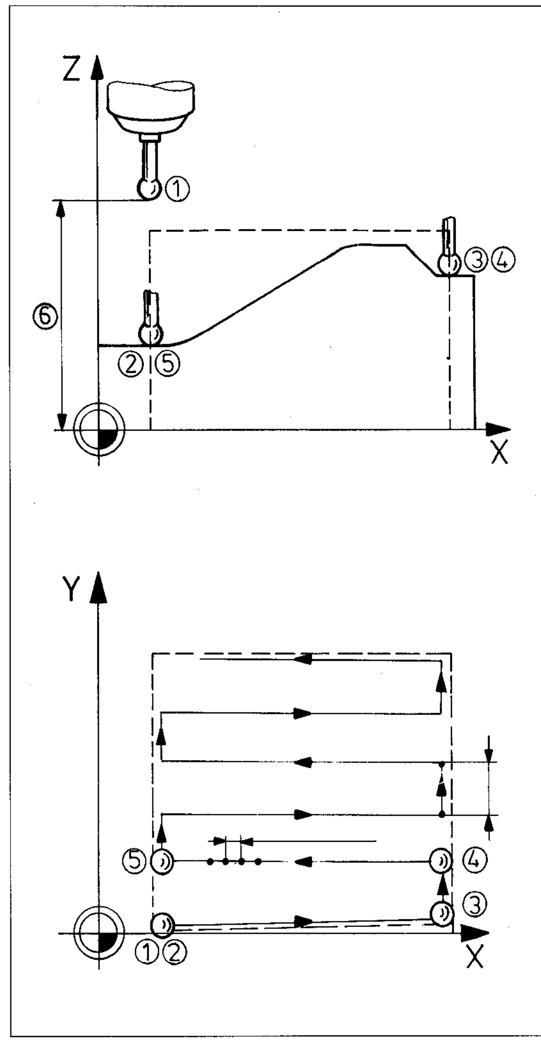
	Initiate dialog	TOUCH PROBE	
	TOUCH PROBE 5.0: RANGE	↓ ENT	Select the RANGE cycle and enter.
PGM NAME	PGM NAME ?	☐ ENT	Enter the program name in which the measured positions are filed.
MIN	TCH PROBE AXIS ?	Z	Enter axis parallel to touch probe, e.g. Z.
	MIN. POINT RANGE ?	0 ENT	X coordinate
		0 ENT	Y coordinate
		0 ENT	Z coordinate
MAX	MAX. POINT RANGE ?	1 0 0 ENT	X coordinate
		1 0 0 ENT	Y coordinate
		4 0 ENT	Z coordinate
CLEARANCE HEIGHT	CLEARANCE HEIGHT ?	6 0 . 5 ENT	Enter the **clearance height**

Display

```
1 TCH PROBE 5.0 RANGE
2 TCH PROBE 5.1
        PGM NAME: 12345678
3 TCH PROBE 5.2 Z X+0.000
        Y+0.000      Z+0.000
4 TCH PROBE 5.3    X+100.000
        Y+100.000    Z+40.000
5 TCH PROBE 5.4
        HEIGHT: +60.5
```

Digitizing 3D Contours
Line-by-line digitizing

The **MEANDER** scanning cycle digitizes a 3D contour in a meandering pattern (in a series of parallel lines) within the **RANGE** which was previously defined.



Starting position

The starting point is approached automatically with an approach logic Z to clearance height, then X, Y. The starting point coordinates are obtained from the definition of the **RANGE** scanning cycle:

X, Y: from the MIN corner
Z: clearance height

Approaching the contour

From the starting point ① the touch probe moves in negative direction (Z–) toward the contour surface. At the first contact, the position ② is stored.

Scanning

The touch probe moves in positive **line direction** while probing the surface in the probe point intervals (**PP INT**). The actual point intervals can be smaller than the programmed intervals (depending on the machine parameters).

Ascending, descending contour

If the control recognizes an ascending or descending contour, the probe retracts in the vertical direction (normal to the current position). The length of the path of retraction in normal direction is referred to as **TRAVEL.**

At the positive end of the range ③ the touch probe moves by the programmed line spacing (**L.SPAC**) in the positive **column direction** ④.

The touch probe now moves in negative line direction back to the negative end of the range ⑤.

This procedure is repeated **line by line** until the entire range is scanned.

The contour is departed automatically through retraction to the **clearance height.**

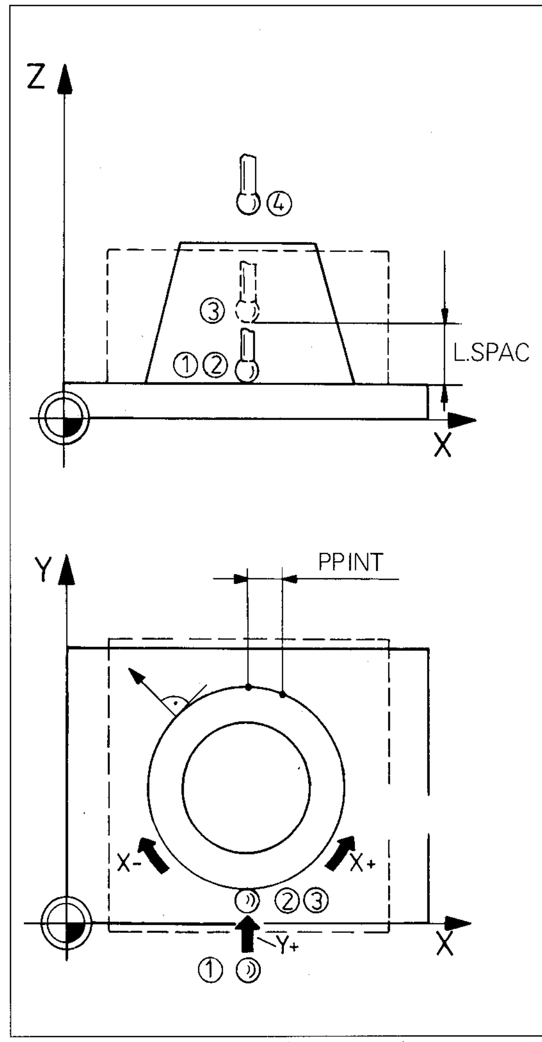
Digitizing 3D Contours
Defining the MEANDER scanning cycle

Initiate dialog	TOUCH PROBE	
TOUCH PROBE 6.0: MEANDER	↓ ENT	Select the MEANDER cycle and enter.
LINE DIRECTION	X ENT	Key in the line direction, e.g. X, and enter.
LIMIT IN NORMAL LINES DIRECTION ?	ENT	Enter the travel, e.g. 0.5, and confirm entry.
LINE SPACING ?	ENT	Enter the line spacing, e.g. 0.5, and confirm entry.
MAX. PROBE POINT INTERVAL ?	ENT	Enter the point spacing, e.g. 0.5, and confirm entry.

```
0 BEGIN PGM 1 MM
1 TCH PROBE 5.0 RANGE
2 TCH PROBE 5.1
        PGM NAME: 12345678
3 TCH PROBE 5.2 Z X+0.000
        Y+0.000       Z+0.000
4 TCH PROBE 5.3    X+100.000
        Y+100.000     Z+40.000
5 TCH PROBE 5.4
        HEIGHT: +60.5
6 TCH PROBE 6.0    MEANDER
7 TCH PROBE 6.1    DIRECTN: X
8 TCH PROBE 6.2    TRAVEL: 0.5
       L.SPAC: 0.5    PP INT 0.5
9 END PGM 1
```

Digitizing 3D Contours
Level-by-level digitizing

The **CONTOUR LINES** cycle scans a 3D contour by circling around the model in a series of upwardly successive levels within the **RANGE** which was previously defined.



Starting position

The starting point ① is approached automatically with an approach logic Z to clearance height, then X, Y. The starting point coordinates are obtained as follows:

X, Y: input in the **CONTOUR LINES** cycle
Z: from the MIN point given in the **RANGE** cycle

Approaching the contour

From the starting point ① the touch probe moves in the programmed direction (here Y+) toward the contour surface. At the first contact, the position ② is stored (1st touch point = target point for the respective contour line).

Scanning

The touch probe moves in the programmed direction of the starting axis along the contour while the probe positions are digitized in the programmed point interval (**PP INT**). The actual point interval may be less than the programmed interval (depending on the machine parameters).

Once the first level has been rounded, i.e. the probe has returned to the first probe point ②, the probe is raised by the line spacing (**L.SPAC**) to the next level (③). This process is repeated until the range limit has been reached.

Departing the contour

The contour is departed automatically through retraction to the safety clearance ④.

For the CONTOUR LINES scanning cycle, the RANGE cycle must be defined as follows:

- In the touch probe axis (e.g. Z axis) the Z MAX point coordinates should be smaller than the maximum Z coordinate of the contour by at least the radius of the ball tip of the touch probe stylus.
- In the working plane the range should be larger than the contour by at least the radius of the ball tip of the touch probe stylus.

Time limit

If the first probe point (②) is not reached within the programmed time, the digitizing process is terminated with an error message.

Digitizing 3D Contours
Defining the CONTOUR LINES scanning cycle

Initiate dialog — TOUCH PROBE

TOUCH PROBE 7.0: CONTOUR LINES — ↓ ENT Select CONTOUR LINES and enter.

TIME LIMIT ?	ENT Enter time, e.g. 200 seconds, and confirm entry.
STARTING POINT ?	X Enter coordinates of starting point, e.g. X+50
	Y ENT Y+100, and confirm entry.

AXIS AND DIRECTION OF APPROACH ?	Y +/- ENT Enter appropriate order of axes, e.g. Y+, X+
STARTING PROBE AXIS AND DIRECTN ?	X +/- ENT and confirm entry.

LIMIT IN NORMAL LINES DIRECTION ?	ENT Enter travel, e.g. 0.5, and confirm entry.
LINE SPACING ?	ENT Enter line spacing, e.g. 0.5, and confirm entry.
MAX. PROBE POINT INTERVAL ?	ENT Enter point interval, e.g. 0.5, and confirm entry.

Example

```
0 BEGIN PGM 2 MM
1 TCH PROBE 5.0 RANGE
2 TCH PROBE 5.1
        PGM NAME: 12345678
3 TCH PROBE 5.2 Z X+0.000
        Y+0.000      Z+2.000
4 TCH PROBE 5.3   X+100.000
        Y+100.000    Z+10.000
5 TCH PROBE 5.4
        HEIGHT: +60.5
6 TCH PROBE 7.0   CONTOUR LINES
7 TCH PROBE 7.1   TIME: 200
        X+50.000     Y+100.000
8 TCH PROBE 7.2
                ORDER: Y+/X+
9 TCH PROBE 7.3   TRAVEL: 0.5
       L.SPAC: 0.5  PP INT: 0.5
10 END PGM 2 MM
```

Digitizing 3D Contours
Executing a program with digitized positions

An additional short program section permits the scanned program stored by the FE 401 floppy disk unit or by a computer to be executed without radius compensation (R0) or any other changes, provided that the tool radius is equal to the effective ball tip radius of the 3D touch probe.

If a milling cutter with a different radius is used, an incorrect contour or a contour with oversize will result!

Example

Program of a 3D contour digitized with the CONTOUR LINES scanning cycle.

0 BEGIN PGM 1 MM	
1 L Z+0 FMAX	Starting point in Z
2 L X+0 Y–25 FMAX	Starting point in X, Y
3 L X+0 Y–8.849	
4 L X+2.003 Y–8619	
.	
.	
.	
29 L X–1.808 Y–8.669	
30 L X+0 Y–8.85	
31 L Z+1 X+0 Y–6.292	New contour line
32 L X+2.004 Y–5.965	
.	
.	
.	
53 L X+0 Y–6.293	
54 L X+0 Y–25 FMAX	End point = starting point in X, Y
55 L Z+40 FMAX	End point = clearance height in Z
56 END PGM 1 MM	

Before this program can be run, the following short program must be entered and executed:

0 BEGIN PGM 2 MM	
1 TOOL DEF 1 L+30 R+5	Tool definition: The tool radius **must** equal the radius of the ball tip of the 3D touch probe stylus as determined during calibration.
2 TOOL CALL 1 Z S 2250	Tool call
3 L R0 F525 M03	**No** radius compensation, feed rate, spindle on with clockwise rotation
4 L Mxy	M function determined by the machine tool manufacturer with which the tool, feed rate and direction of spindle rotation remain modally effective – even if a new program is selected.
5 END PGM 2 MM	

Digitizing 3D Contours
Error messages

WRONG AXIS PROGRAMMED

The touch probe axis in the **RANGE** scanning cycle is not the same as the touch probe axis used for calibration of the touch probe system.

FAULTY RANGE DATA

- A MIN coordinate value is larger than or equal to the corresponding MAX coordinate value in the **RANGE** scanning cycle.
- One or more coordinates in the **RANGE** scanning cycle are outside of the software limit switch range.
- The **RANGE** scanning cycle was not defined before calling the **MEANDER** or **CONTOUR LINES** scanning cycle.

MIRRORING NOT PERMITTED
ROTATION NOT PERMITTED
SCALING FACTOR NOT PERMITTED

Mirroring, rotation or scaling factor is active when the **RANGE, MEANDER** or **CONTOUR LINES** scanning cycle is called.

RANGE EXCEEDED

The probe moved beyond the borders of the range during probing, i.e. the 3D contour lies partially outside of the range.

CYCL-PARAMETER INCORRECT

The programmed travel, line spacing or probe point spacing is negative or larger than 65,535 mm (only possible via Q-parameter programming).

TOUCH POINT INACCESSIBLE

- Contact occurred during approach before the range was reached.
- In the **CONTOUR LINES** cycle no contact occurred between approach to the 3D contour and departure from the range.

STYLUS ALREADY IN CONTACT

The stylus is not at rest position during the beginning of the scanning cycle.

PLANE WRONGLY DEFINED

One of the starting point coordinates in the **CONTOUR LINES** scanning cycle is assigned to the touch probe axis.

START POSITION INCORRECT

The starting point coordinate is outside of the range in the starting probe axis.

AXIS DOUBLE PROGRAMMED

The starting point coordinates in the **CONTOUR LINES** scanning cycle were both assigned to the same axis.

TIME LIMIT EXCEEDED

During the **CONTOUR LINES** scanning cycle the first point of the line was not returned to within the programmed time limit.

Teach-In
Transferring tool compensation values

Tool compensation

Position values (coordinates) acquired via "Transfer actual position" contain the length and radius of compensation for the tool in use.

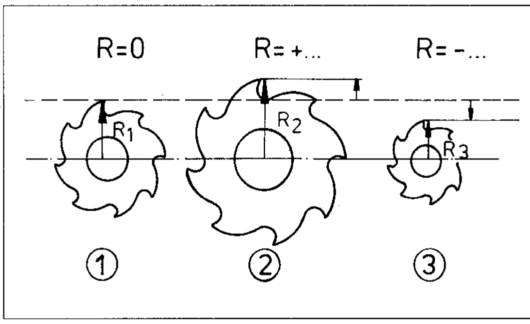
TOOL DEF

Therefore it is advisable when programming with "Transfer actual position" to enter the correct radius compensation (RL, RR or R+, R–) and use L = 0 and R = 0 in the tool definition.

If the tool breaks or another tool is selected instead of the original, then a different length and radius can be taken into account.

The new compensation values for the tool radius are differences:

1. Tool definition in the machining program
3 TOOL DEF 1 L+0 R+0
2. Tool definition in the central tool file
T1 L+0 R+0



Radius compensation

$R = R_2 - R_1$ or
$R = R_3 - R_1$

R = radius compensation for the tool definition
R_1 = tool radius of the original tool ①
R_2 = tool radius of a new tool ②
R_3 = tool radius of a new tool ③

The compensation R can be **positive or negative,** depending on whether the tool radius of the newly inserted tool is larger (+) or smaller (–) than the original tool.

Length compensation

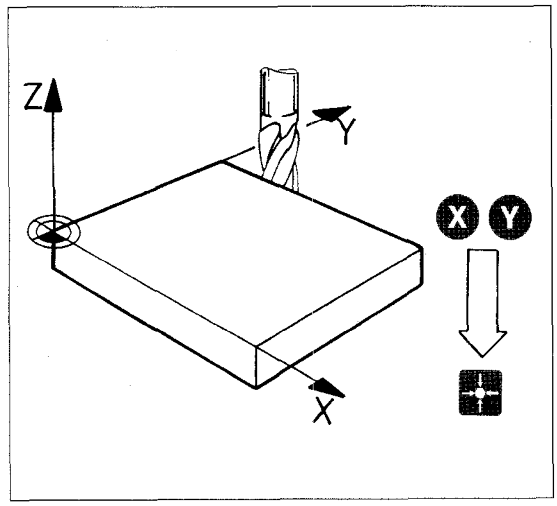
The compensation for the new tool length is also determined as the difference to the originally used tool (see Tool definition, Transferring tool lengths).

The new compensations are entered in the tool definition of the original tool (R = 0, L = 0).

Teach-In
Transferring Actual Positions to Program

Transfer actual position

The actual tool position can be transferred to the machining program with the "Transfer actual position" key.



Application possibilities

or

X R+ Z

or

TOOL DEF

In this way you can transfer:
- positions
- tool dimensions (see Tool Definition).

Process

Traverse the tool to the desired position.

Open a program block (e.g. for a straight line) in the "Programming and editing" operating mode. Select the axis from which the actual value is to be transferred.

This axis position is transferred by pressing the "Transfer actual position" key.

```
PROGRAMMING AND EDITING
COORDINATES ?
4     TOOL CALL 1                  Z
                  S 125
5     L   X+12,545           Y+45,12
                           R   F          M
6     END PGM 55                 MM

--------------------------------------------

ACTL.  X +    49,258    Y +     23,254
       Z +    15,321    C +     84,000

                        F 0         M5/9
```

Example

X Y Z Move the axis or axes via the axis keys.

Input

Initiate the dialog

COORDINATES ?

X Y ... Transfer the axis positions individually.

ENT Terminate position input.

RADIUS COMP.: RL/RR/NO COMP. ?

FEED RATE ? F =

MISCELLANEOUS FUNCTIONS M ?

RL Enter the radius compensation if needed.

ENT Enter the feed rate, if needed, and confirm entry.

ENT Enter miscellaneous function if needed.

You can skip dialog queries with "NO ENT" or "END □".

Test Run

In the "Test run" operating mode, a machining program is checked for the following errors without machine movement:

- Overrunning the traversing range of the machine
- Exceeding the spindle speed range
- Illogical entries, e.g. redundant input of one axis
- Failure to comply with elementary programming rules e.g. cycle call without a cycle definition
- Certain geometrical incompatibilities

```
TEST RUN
     TO BLOCK NUMBER =

0    BEGIN PGM 666          MM
1    BLK FORM 0.1   Z   X-85
          Y-70          Z-30
2    BLK FORM 0.2       X+85
          Y+70          Z+0
----------------------------------
ACTL. X +     49,258   Y +    23,254
      Z +     15,321   C +    84,000

                       F 0       M5/9
```

Testing the program

Initiate the dialog — PGM NR

PROGAM SELECTION PROGRAM NUMBER =	ENT	Select the program to be tested.
TO BLOCK NUMBER =	ENT	Key in and transfer the block number up to which the test is to run.
	or	
	NO ENT	Test the complete program.

No apparent errors

If the program contains no apparent errors, the program test runs until the entered block number is reached, or a jump is made back to the start of program if no STOP or M06 was programmed.

STOP/M06

If a STOP or M06 was programmed, the test can be continued by entering a new block number or by pressing the "NO ENT" key.

Error

If an error is found, the program test is stopped. The error is usually located in or before the stopped block. An error message is displayed on the screen.

STOP

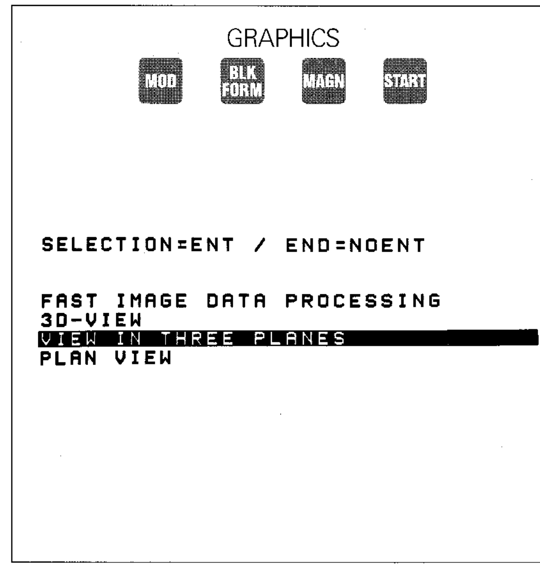
The program test can be halted with the "STOP" key and aborted at any time.

Test Graphics

GRAPHICS MOD BLK FORM MAGN START

Machining programs can be simulated graphically and checked in the "Program run" operating modes "Full sequence" and "Single block", if a blank has been previously defined (BLK FORM).

More information on the definition of the blank can be found in the section Program Selection, Blank form definition.



MOD GRAPHICS

After selecting a program, the "menu" shown at the right is displayed by pressing the GRAPHICS "MOD" key twice.

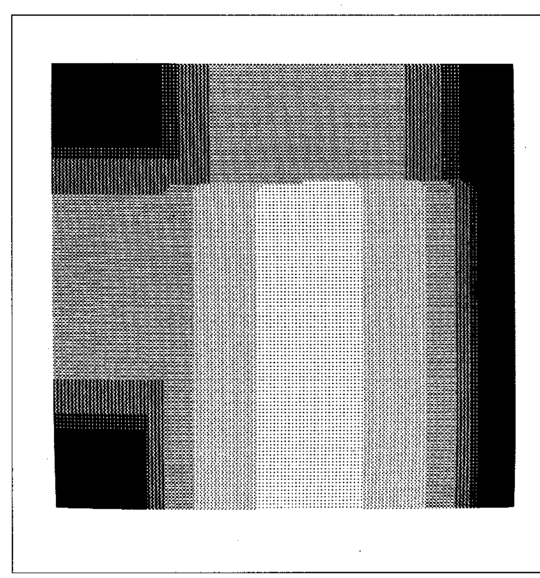
One of the versions of the graphic presentations can be selected with the vertical cursor keys and entered with the "ENT" key.

START

The graphic simulation or internal computation is started with the "START" key.

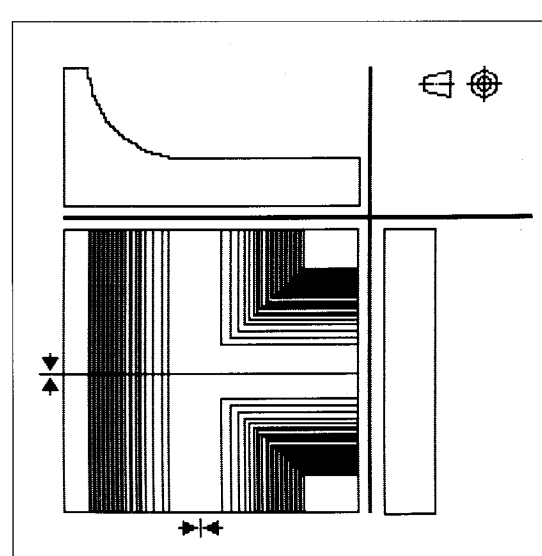
Fast data image processing

With "Fast data image processing" only the current block number is displayed on the screen and the internal computing also indicated by an asterisk (* = control is started).
When the program has been processed, the "machined" workpiece can be displayed in plan view, view in three planes or 3D view.



Plan view with depth indication

The workpiece center is shown in the plan view with up to 7 different shades: the lower the darker.



View in three planes

The workpiece is shown – like in drafting – with a plan view and two sections.

The sectional planes can be moved via the cursor keys.

The view in three planes can be switched from the German to the American projection standard via a machine parameter. A symbol (in conformance to ISO 6433) indicates the type of projection:

German standard (DIN/ISO)

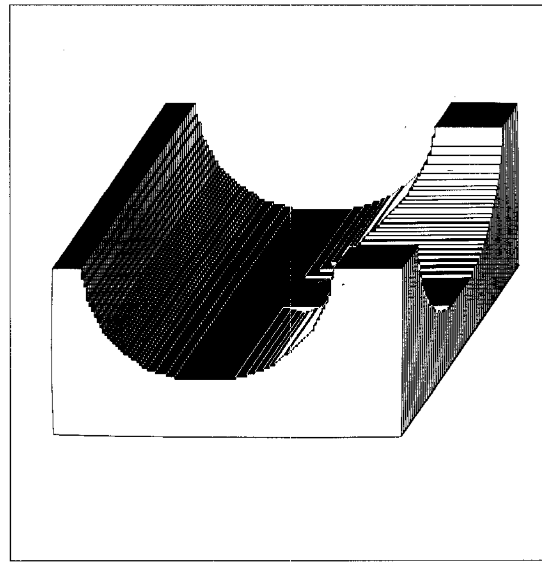
American standard

Test Graphics

GRAPHICS MOD BLK FORM MAGN START

3D view

The program is simulated in a three-dimensional view.



← →

The displayed workpiece can be rotated by 90° with each activation of the horizontal cursor keys. The orientation is indicated by an angle.
└ = 0° ┐ = 180°
┘ = 90° ┌ = 270°

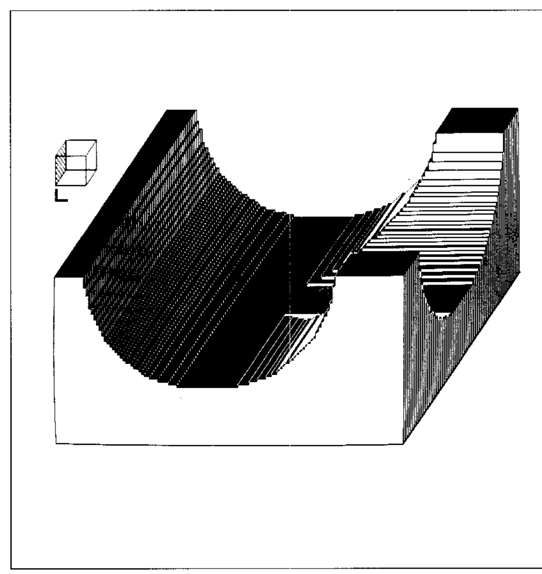
↑ ↓

If the height to side proportion is between 0.5 and 50, the type of display can be changed with the vertical cursor keys. You can switch between a scaled and non-scaled view. The short height or side is shown with a better resolution in the nonscaled view. The dimensions of the angle indicator change to a show the disproportion.

Magnifying

MAGN

You can magnify a detail of the displayed work with the "MAGN" key. A wire model with a hatched surface appears next to the graphic. This marks the sectional plane.



Selecting the sectional plane

↑ ↓

You can select a different sectional plane with the horizontal cursor keys.

Trimming

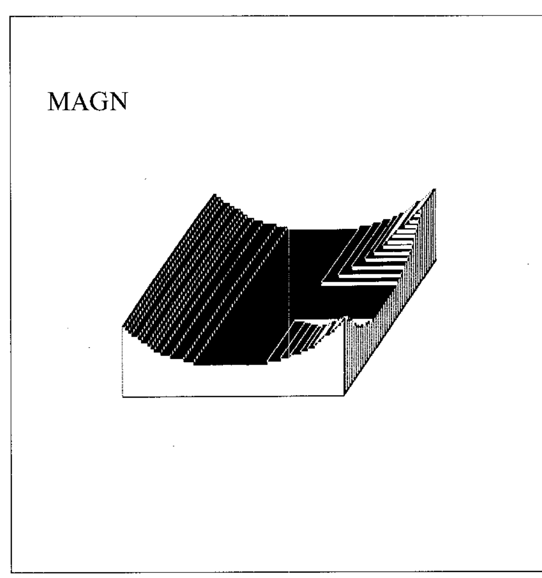
← →

You can trim the selected plane or cancel the section with the horizontal cursor keys.

Transferring the detail

↓ ENT

Once the desired detail is displayed, select the dialog "TRANSFER DETAIL = ENT" with the vertical cursor keys and confirm with the "ENT" key.



Magnification

START

The "remaining workpiece" is displayed on the screen with "MAGN".

Another graphic simulation of the magnified detail can be executed in the plan view, the view in three planes or the 3D view via the "START" key.

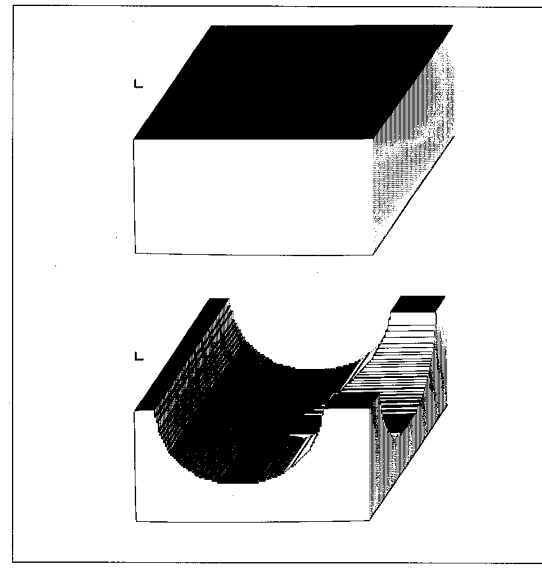
BLK FORM

Test Graphics

GRAPHICS MOD BLK FORM MAGN START

BLK FORM

You can revert to the blank with the "BLK FORM" key and restart simulation with "START".



Tips

The "3D view" and "View in three planes" require extensive computing. For long programs, we therefore recommend displaying the workpiece with "Fast data image processing" or in the quicker "Plan view with depth indication" first, and then switching to the "3D view" or the "View in three planes".

Displaying details

The following aids are available if fine details are to be examined:

- Trim the blank and magnify in an additional graphic program run.
- Restrict the blank detail to the section of interest.

Tool call

One "TOOL CALL" must be programmed prior to the first axis movement to designate the tool axis. Specifying the spindle axis in the BLK FORM definition does not suffice for the graphic program run. Both entries for the axis must be the same.

If the tool axis is not given, an error message appears after starting the graphics.

External Data Transfer
General information

EXT

The control has one data interface for read-in or output of programs. The data format complies with the following standard:

- RS-232-C (ISO)

The data interface can function in two different manners:

Blockwise transfer for
the HEIDENHAIN FE 401 Floppy Disk Unit and compatible computers.

Standard data transfer for
the HEIDENHAIN ME magnetic tape unit*, or for a printer, punch, reader, etc.

* (no longer in production).

Device adaptation

The TNC can be adapted to various peripheral devices via machine parameters, which can be accessed as user parameters.

The settings for three different peripheral devices are permanently stored in the TNC (selectable via "MOD"):

- FE = for HEIDENHAIN FE 401 Floppy Disk Unit.
- ME = for HEIDENHAIN ME magnetic tape unit.
- EXT = external devices.
  Interface defined by the machine manufacturer or user via machine parameters to connect a non-HEIDENHAIN device such as a printer, computer, etc.

External programming

Programs can also be written externally.

Observe the programming rules in this manual and the following instructions.

- At the start of program and after every program block, CR LF or LF or CR FF or FF must be programmed.[1)]
- After the end of program block, CR LF or LF or CR FF or FF[1)] and also ETX (Control C) must be programmed.
  Any character can be substituted for ETX.
- Spaces between single words can be omitted.
- Zeroes between single words can be omitted.
- During read-in of NC programs, comments that are marked with "*" or ";" are overread.

[1)] CR, LF at the start of program and CR, LF or LF or FF after every block are not required for "blockwise transfer". This function is assumed by the control characters.

External Data Transfer
Transfer menu

Read-in/ read-out

Machining programs can be read-out or read-in by the control. For example, the "Read-in program" display on the control means: data is entered from the floppy disk station and received by the control. Program transfer in the "Programming and editing" operating mode must be initiated from the control.

Transfer menu

The transfer mode is selected via a menu, which offers different read-in and read-out alternatives.

```
PROGRAMMING AND EDITING
SELECTION = ENT/END = NOENT

PROGRAM DIRECTORY
READ-IN ALL PROGRAMS
READ-IN PROGRAM OFFERED
READ-IN SELECTED PROGRAM
READ-OUT SELECTED PROGRAM
READ-OUT ALL PROGRAMS
```

Selections

Read-in to the TNC	Read-out from the TNC
PROGRAM DIRECTORY The list of program numbers on the data medium is displayed. The programs are not transferred. READ-IN ALL PROGRAMS All programs are read-in from the data medium. READ-IN PROGRAM OFFERED The programs are offered in the sequence in which they were externally stored and, if desired, can be read-in. READ-IN SELECTED PROGRAM A single, selected program is read-in.	READ-OUT SELECTED PROGRAM A single, selected program is read-out. READ-OUT ALL PROGRAMS The entire NC program memory is read-out.

Interrupting the data transfer

A started data transfer can be interrupted on the TNC by pressing the "END □" key.
After interruption of the data transfer, the following error message appears:

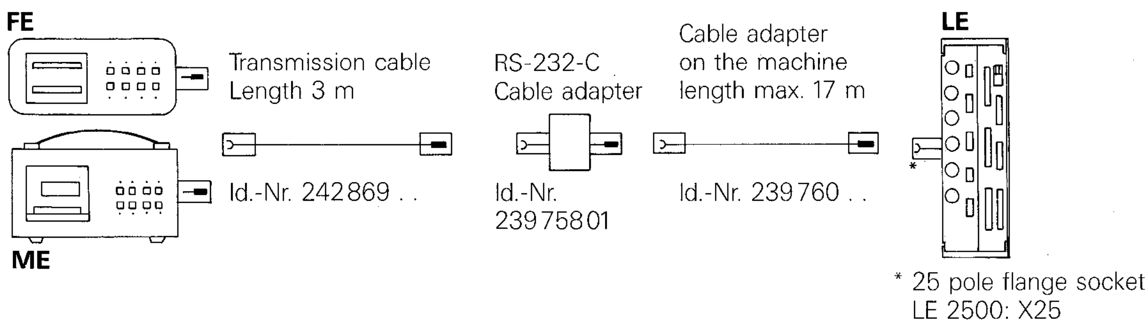
PROGRAM INCOMPLETE

Transfer TNC – TNC

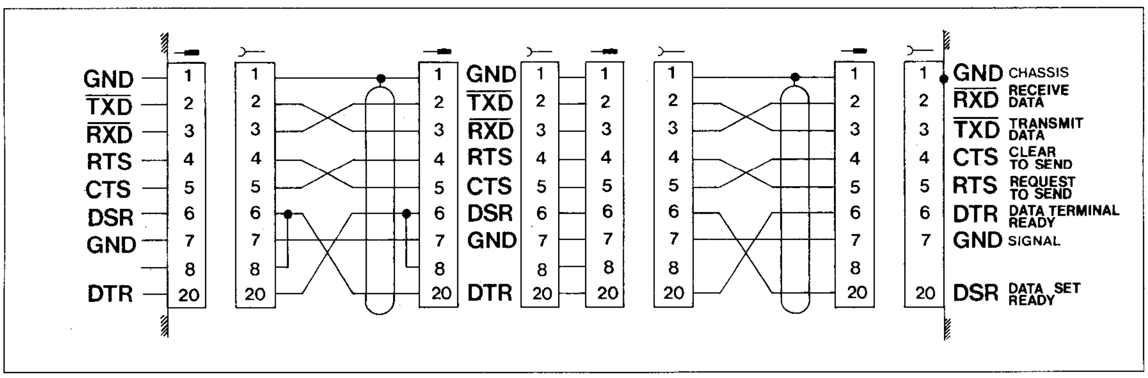
Data can also be transferred directly between two controls. The receiving control must be started first.

External Data Transfer
Connecting cable/Pin assignment for RS-232-C

HEIDENHAIN devices

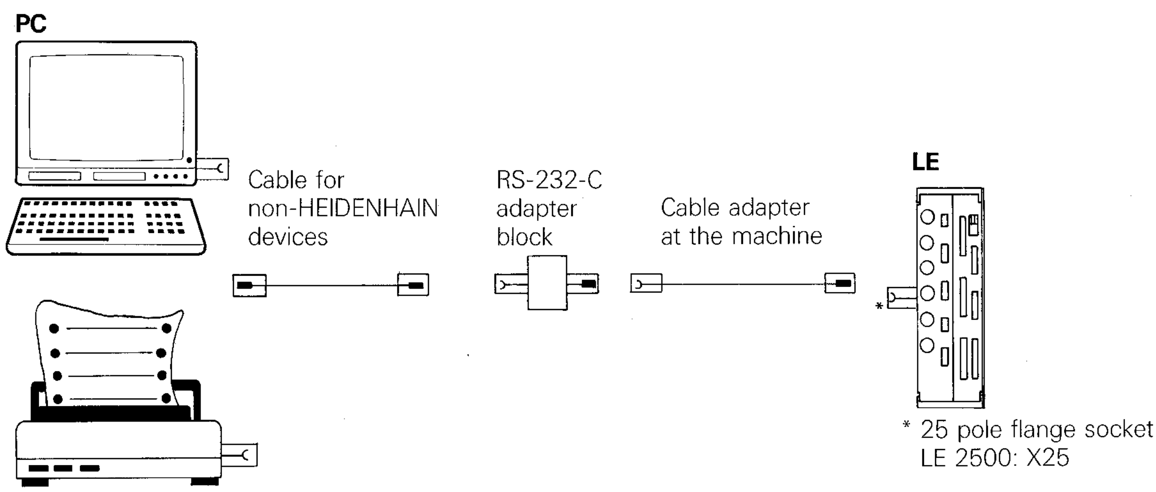


HEIDENHAIN-standard cable

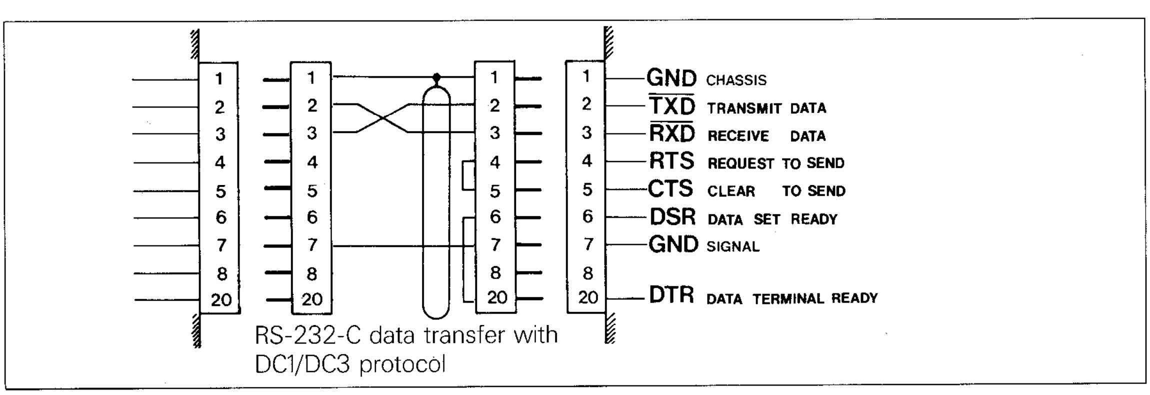


The RS-232-C data interface has a different pin layout at the LE and at the adapter block.

Non-HEIDENHAIN devices



Recommended pin layout for non-HEIDENHAIN devices



External Data Transfer
Peripheral devices

Adaptation The control interface must be set for the specific device which is to be connected.

HEIDENHAIN devices HEIDENHAIN devices are mated with the TNC controls and are therefore especially easy to put into operation:

FE, ME The adaptation for FE or ME can be selected via "MOD". The suitable standard cable can be ordered. The transfer rate can be altered for the FE 401B.

Connections In built-in controls, peripheral devices can usually be connected via a cable adapter on the operating panel or another accessible location on the machine.

Non-HEIDENHAIN devices Non-HEIDENHAIN devices must be individually adapted. This also includes:

- Adapting the control via machine parameters.
  These settings are stored after input and are automatically effective by selecting EXT.
- Adapting the peripheral device, e.g. via switches.
- Setting the baud rates for both devices.
- Wiring the data transfer cable.

Please remember: Both sides must be set identically.
You should always document the settings!

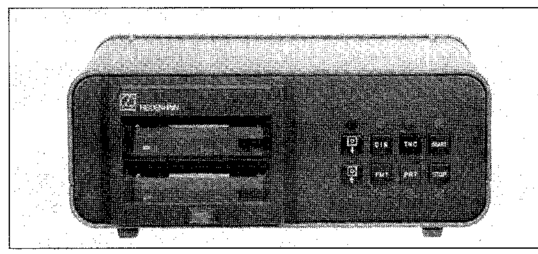
External Data Transfer
HEIDENHAIN FE 401 Floppy Disk Unit

Preparing the FE *)

Connect the FE to the mains, plug in the data cable, switch on, insert floppy disk in the upper drive, select the baud rate if necessary.

Please note when writing a diskette:

- You must format the diskette before writing for the first time.
- Do not write-protect the diskette.



Setting the TNC

Select at the TNC

MOD ↓ Continue pressing until RS-232-C INTERFACE appears.

RS-232-C INTERFACE =	ENT	Continue pressing until the FE setting appears.
	END	Terminate the MOD operating mode.

Examples for using the FE
Read-out selected program

Select EXT ↓

READ-OUT SELECTED PROGRAM	ENT	Confirm the function.
OUTPUT = ENT/END = NOENT		
1 13	↓ ↑	Select the program, e.g. program 14.
14 24	ENT	Output the program.
EXTERNAL DATA OUTPUT		The FE is started and stopped after program transfer.
OUTPUT = ENT/END = NOENT		The next program number is then highlighted.
		Select and output the next program or
	NO ENT	**terminate output. Very important!**

Read-in selected program

Select EXT ↓

READ-IN SELECTED PROGRAM	ENT	Confirm the function.
PROGRAM NUMBER =	☐ ENT	Enter the number, read-in.
EXTERNAL DATA INPUT		

The FE generally functions with "blockwise transfer" and can be switched over on the rear to operate like an ME.

*) The entire range of functions for the FE is described in the operating manual for the FE.

External Data Transfer
Non-HEIDENHAIN devices

EXT

EXT

After setting the TNC data interface to EXT, the following modes can be selected via machine parameter:

Standard data transfer for
printer, reader, puncher etc.

Blockwise transfer for
computer.

To transfer data from the control to non-HEIDENHAIN devices, the control must be adapted by machine parameters.

The transfer rate is set via the MOD function BAUD RATE.

Resetting the TNC to EXT

Select at the TNC

MOD ↓ Continue pressing until RS-232-C interface appears.

RS-232-C INTERFACE	ENT Continue pressing until the EXT setting appears. END Terminate the MOD operating mode.

Standard data transfer

For standard data transfer (e.g. to a printer), you only have to enter the following machine parameters at the control:

MP 5030 = 0 (standard data transfer is selected).

MP 5020 = e.g. 168 (data format)
(see External Data Transfer, Machine parameters).

Blockwise transfer

For "blockwise transfer" from a computer, transfer software is required, e.g. the data transfer software from HEIDENHAIN for personal computers.
For this operating mode, you must set the following machine parameters:

MP 5030 = 1 (blockwise transfer is selected).

MP 5020 = e.g. 168 (data format).

The following machine parameters determine the control character (for description see External Data Transfer, Machine parameters) and are valid for the data transfer software from HEIDENHAIN. If different transfer software is used, the machine parameters must be adapted correspondingly.

MP 5010 = 515
MP 5010.1 = 17736
MP 5010.2 = 16712
MP 5010.3 = 279
MP 5010.4 = 5382
MP 5010.5 = 4

When using the transfer software from HEIDENHAIN, the data interface is normally set to "FE". Then the above machine parameters need not be entered.

Adapting to non-HEIDENHAIN devices

Compare the interface descriptions of both devices.
Then proceed as follows:

- Determine the common settings (data format, baud rate).
  The peripheral device is usually set via internal switches.
- Determine the pin layout for the data transfer cable, and wire the cable.
- Plug in the data transfer cable.
- Plug in the power cord of the peripheral device.
- Switch on the power.
- Start the transfer software on computers, if required.
- Select the transfer menu on the TNC with the "EXT" key and start the desired transfer.

The following settings are only effective when operating the data interface in the "EXT" operating mode. To select the machine parameters, see index General Information, MOD Functions, User parameters.

MP 5010 Control characters for blockwise transfer

MP	Bit	Function	Input values[1]
5010.0	0 ... 7 8 ... 15	ETX or any ASCII character. Character for end of program. STX or any ASCII character. Character for start of program.	ETX and STX: 515
5010.1	0 ... 7 8 ... 15	H or any ASCII character. It is sent in the command block for **data input** prior to the program number. E or any ASCII character. It is sent in the command block for **data input** after the program number.	H and E: 17736
5010.2	0 ... 7 8 ... 15	H or any ASCII character. It is sent in the command block for **data output** prior to the program number. A or any ASCII character. It is sent in the command block for **data output** after the program number.	H and A: 16712
5010.3	0 ... 7 8 ... 15	ETB or substitute character (decimal code 1–47) is sent at the **end** of the **command block.** SOH or substitute character (decimal code 1–47) is sent at the **beginning** of the **command block.**	ETB and SOH: 279
5010.4	0 ... 7 8 ... 15	ACK or substitute character (decimal code 1–47): positive acknowledgement. It is sent when the data block is **correctly** received. NAK or substitute character (decimal code 1–47): negative acknowledgement. It is sent when the data block is **incorrectly** transferred.	ACK and NAK: 5382
5010.5	0 ... 7	EOT or substitute character (decimal code 1–47) is sent at the **end** of the **data transfer.**	EOT: 4

[1] The input values apply for the data transfer software from HEIDENHAIN

MP 5010.0
This defines one character from the ASCII character code for the end of program and one for the start of program for external programming. ASCII characters 1–47 are accepted.
"End of program" is sent at "standard data interface" and "blockwise transfer".
"Start of program" is only sent at "blockwise transfer".

Determining bit significance MP 5010.0

Example: End of program: ETX BINARY code 00000011
Start of program: STX BINARY code 00000010

Bits 0 – 7	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Significance of bit	128	64	32	16	8	4	2	1
Enter 0 or 1 accordingly	0	0	0	0	0	0	1	1

Bits 8 – 15	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Significance of bit	32768	16384	8192	4096	2048	1024	512	256
Enter 0 or 1 accordingly	0	0	0	0	0	0	1	0

Determine input value:

$$\begin{array}{r} 1 \\ 2 \\ +\ 512 \\ \hline 515 \end{array}$$

The input value for MP 5010.0 is thus 515.

The data format and the type of transfer stop are determined by MP 5020. Bit 1 is only set for "blockwise transfer". 0 is entered for standard data interface.

MP 5020
Data format

Function	Bit	Input	Input values
7 or 8 data bits	0	+ 0 → 7 data bits (ASCII code with 8th bit = parity) + 1 → 8 data bits (ASCII code with 8th bit = 0 and 9th bit = parity)	1
Block Check Character (BCC)	1	+ 0 → any BCC character + 2 → BCC character no control character	–
Transfer stop due to RTS	2	+ 0 → inactive + 4 → active	–
Transfer stop due to DC3	3	+ 0 → inactive + 8 → active	8
Character parity even or odd Character parity required	4 5	+ 0 → even + 16 → odd + 0 → not required + 32 → required	32
Number of stop bits	7 | 6 0 | 0 0 | 1 1 | 0 1 | 1	 1 1/2 Stop bits 2 Stop bits bit 6: + 64 1 Stop bit bit 7: + 128 1 Stop bit	128
		value to be entered for MP 5020	169

Notes on bit 1

Input value does not contain the significance 2:
The BCC can accept an arbitrary character (also control character) in "blockwise transfer".

Input value contains the significance 2:
If the computation of the BCC during "blockwise transfer" results in a number less than 20 HEX[1] (control character), then a "space" character (20 HEX) is additionally sent prior to ETB. In this case, the BCC is always greater than 20 HEX and therefore not a control character.

[1] HEX = Hexadecimal

Example of value determination

Standard data format: 7 data bits (ASCII code with 7 bits, even parity) Transfer stop due to DC3, 1 stop bit

Bits 0 – 7	7	6	5	4	3	2	1	0
Significance of bit	128	64	32	16	8	4	2	1
Enter 0 or 1 accordingly	1	0	1	0	1	0	0	0

After adding the significances, you obtain the input value for machine parameter 5020.
In our example: 168.

MP 5030
Operating mode of the interface

Operating mode data interface RS-232-C
This parameter determines the function of the data interface.

0 ≙ **"standard data interface"** (normally for printer, reader, puncher)
1 ≙ **"blockwise transfer"** (normally for computer link)

Miscellaneous functions M

Miscellaneous functions with predetermined function.

	Function	Effective at Begin of block	Effective at End of block	Reference Page
M00	Stop program run/stop spindle/coolant off		●	
M02	Stop program run/stop spindle/coolant off/ or clear the status display/return jump to block 1		●	P63
M03	Spindle on clockwise	●		
M04	Spindle on counterclockwise	●		
M05	Spindle stop		●	
M06	Tool change/or stop program run/stop spindle		●	P20
M08	Coolant on	●		
M09	Coolant off		●	
M13	Spindle on clockwise/coolant on	●		
M14	Spindle on counterclockwise/coolant on	●		
M30	like M02		●	P63
M89	Vacant miscellaneous function or	●		
M89	Cycle call, modal (depends on machine parameters)		●	P69
M90	Constant path speed on inside corners and uncompensated corners	●		P55
M91	**in the positioning block:** Coordinates refer to the scale reference point	●		P58
M92	**in the positioning block:** Coordinates refer to a position defined by the machine manufacturer (machine datum), e.g. tool change position	●		P58
M93	Reserved	●		
M94	Reduce the display of the rotary axis to a value under 360°	●		P31
M95	Reserved		●	
M96	Reserved		●	
M97	Path offset on outside corners: Intersection instead of tangential circle		●	P56
M98	Blockwise end of path offset		●	P57
M99	Cycle call effective blockwise		●	P69

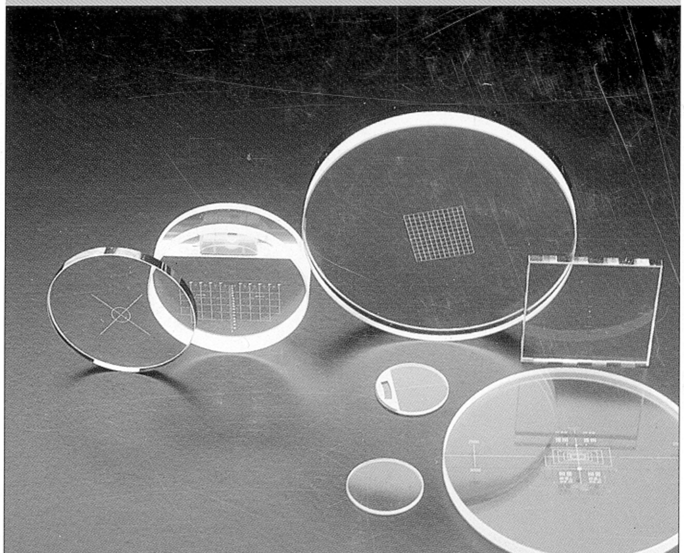
Program example: milling

Dialog initiation key	Function	Example values
PGM NR	Program number, mm/inch	BEGIN PGM 729 MM
BLK FORM	Blank definition: spindle axis Minimum point Maximum point	Z X+0 Y+0 Z–40 X+100 Y+100 Z+0
TOOL DEF	Tool definition Tool number Tool length Tool radius	TOOL DEF 1 L+0 R 7.5
TOOL CALL	Tool call Tool number Spindle axis (e.g. Z), speed (S)	TOOL CALL 1 Z S 100
L	Tool change Retract tool axis, length compensated, Radius uncompensated, rapid traverse, tool change	L Z+200 R0 FMAX M06
L	Starting position: Approach the workpiece No radius compensation, rapid traverse, Spindle rotates to the right	L X–20 Y–20 R0 FMAX M03
L	Move tool axis to working depth	L Z–20
L	**Machine the work, approach 1st contour point with radius compensation, machining feed rate**	L X+0 Y+0 RL F200
	. . **MACHINING** . .	
L	**Last contour point (with radius compensation)**	L X+0 Y+0 RL
L	After machining Retract in the machining plane Deselect radius compensation, spindle stop	L X–20 Y–20 R0 F500 M05
L	Retract the spindle axis, Return jump to the 1st block	L Z+200 M02

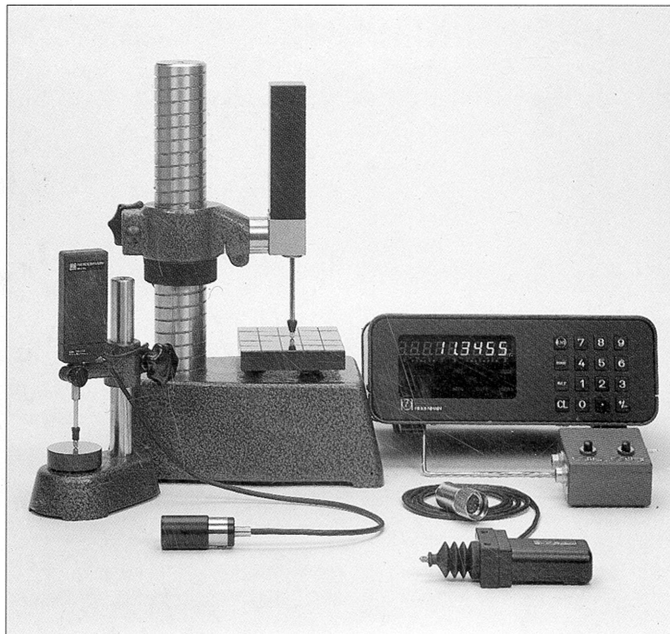
HEIDENHAIN

253 588 20 · 10 · 10/91 · H · Printed in Germany · Subject to alteration

HEIDENHAIN Products

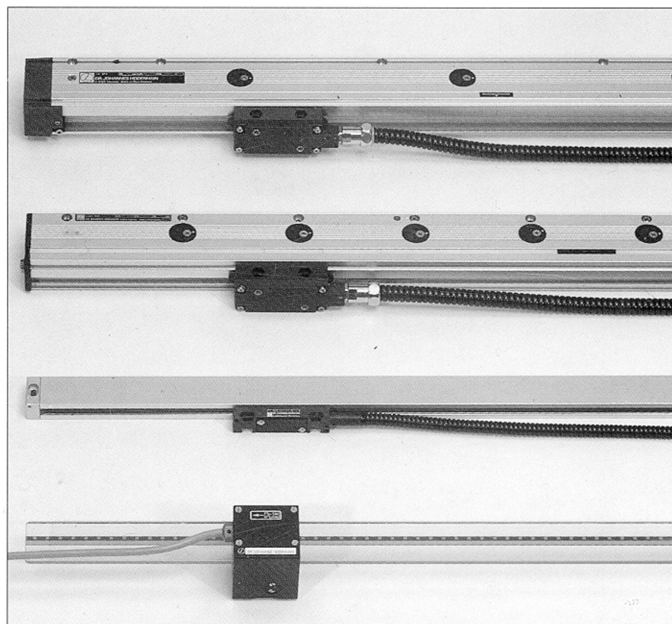


Precision graduations and reticles

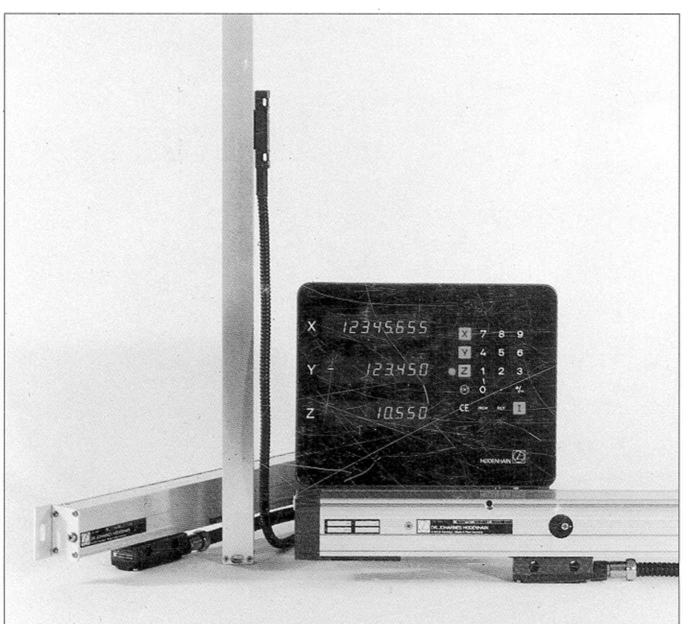


Length gages and display units

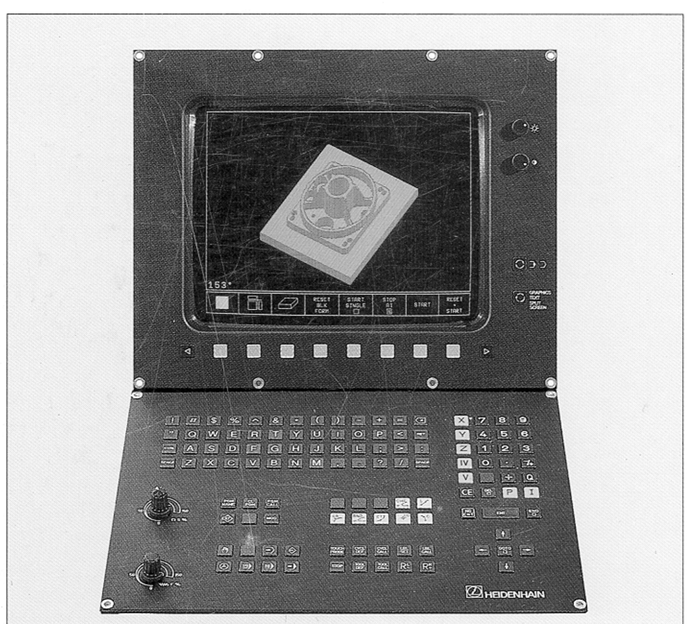
Incremental rotary and angle encoders
Absolute rotary encoders



Incremental linear encoders



Digital readouts for retrofitting machine tools



TNC contouring controls