

Handwritten notes:
1. 10. 83
2. 10. 83

FD 10.83

Programming instructions

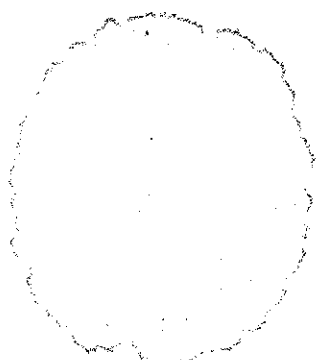
SINUMERIK
3M

Chapter 0	General
Chapter 1	Program format
Chapter 2	Path information
Chapter 3	Preparatory functions
Chapter 4	Miscellaneous and auxiliary functions S, T, M, H
Chapter 5	Parameters
Chapter 6	Insertion of chamfers and radii
Chapter 7	Canned cycles
Chapter 8	Appendix

Key to editions

Up to the present edition, the editions below have been issued. In the column "alterations" the chapters are listed which have been altered with respect to the preceding edition.

<u>Alterations</u>	<u>Order number</u>	<u>Edition</u>
Preliminary Edition 1-7,3-7,3-12,3-35, 4-3, 5-3, 5-5, 5-9.	E321/1820-101	E.4.81
Chapter 5.7, Chapter 5.8, Chapters 7, Page 8-4, 8-7, 8-10, 8-11, 8-12, 8-13, 8-14.	E321/1878-101	E.9.81



Page

Table of contents

0-1	Explanatory notes on programming manual	
1-1	<u>Program format</u>	1.
1-1	Perforated tape coding	1.1
1-2	Address characters	1.2
1-3	Word address system	1.3
1-5	Variable block format	1.4
1-7	Leader	1.5
1-7	Comments	1.6
1-9	Part program	1.7
1-10	Subroutines	1.8
1-11	Subroutine call, subroutine nesting	1.9
1-12	Perforated tape format	1.10
1-13	Tape format for program deletion	1.11
2-1	<u>Path information</u>	2.
2-1	Motion dimensions (axis commands)	2.1
2-2	Mirror image	2.2
3-1	<u>Preparatory functions</u>	3.
3-2	G90/G91 Absolute/incremental programming	3.1
3-3	G00 Rapid traverse	3.2
3-4	G01 Linear interpolation	3.3
3-5	G02/G03 Circular interpolation	3.4
3-8	Example: circular interpolation	3.4.1
3-9	Helical interpolation	3.4.2
3-17	G60 Exact positioning (reset state 10th G group)	3.6
3-18	G63 Tapping with floating tap holder	3.7
3-18	G64 Contour machining	3.8
3-19	G04 Dwell time	3.9

3-20	G70/G71 Input systems	3.10
3-22	zero point offset (Z0)	3.11
3-23	G54/55/56/57 Settable zero point offset	3.11.1
3-24	G59 Programmable additive zero offset	3.11.2
3-26	G53 Cancelling the zero offset	3.11.3
3-27	G94 Feed F	3.12
3-31	G17/18/19 Machining plane selection	3.13
3-32	Tool offset	3.14
3-34	G40/41/42 Intersectional (Look ahead) cutter radius compensation	3.15
4-1	Miscellaneous/auxiliary functions S/T/M	4.
4-1	Function S	4.1
4-2	Tool function T	4.2
4-2	Miscellaneous functions M	4.3
5-1	<u>Parameters</u>	5.
5-2	Parameter definition	5.1
5-2	Assigning parameters in a program	5.2
5-3	Parametric operations	5.3
5-5	Parameter chaining	5.4
5-7	Examples: use of parameters	5.5
5-10	Buffer store empty - E99	5.6
5-11	Branch operations	5.7
5-11	Unconditional jump @ 00	5.7.1
5-13	Conditional jump @ 01, @ 02, @ 03	5.7.2
5-15	Conclusive example of branch operations	5.7.3
5-16	Square root @ 10	5.8
5-17	Sine @ 15	5.9
5-18	Arc tangent @ 18	5.10
5-19	Loading of address parameters @ 20	5.11
5-21	Address parameters @ 90 to @ 93	5.12
5-23	Loading/reading of system stores	5.13
5-26	Access to stores at a glance	Table:

7-9	Drilling/boring with horizontal cutter spindle	7-32
7-8	Milling of oblong holes	7-31
7-7	Drilling pattern	7-30
7-6	Milling of circular pockets	7-28
7-5	Milling of rectangular pockets	7-26
7-4	Milling of grooves	7-25
7.3.3	Drilling/boring cycles G 80 to G 89 with variable drilling/boring axis	7-23
7.2.1	Cycle call in subroutine	7-18
7.2	Examples of limitations in cycle call-up	7-8
7.1	Drilling/boring cycles T81 to T89	7-1
7.	<u>Canned cycles</u>	7-1
6.3	Example of rectangular pattern programming	6-4
6.2	Miscellaneous and auxiliary functions in linked blocks	6-3
6.1	Example	6-2
6.	<u>Automatic insertion of chamfers and radii</u>	6-1

8-1	Appendix	8.
8-2	Intersectional cutter radius compensation (CRC)	8.1
8-2	Selecting the CRC	8.1.1
8-3	CRC used in a program	8.1.2
8-7	Repeating the already selected G code (G41/42) with the same offset number	8.1.3
8-8	CRC cancellation	8.1.4
8-10	M00, M02, M30 with CRC selected	8.1.5
8-11	Combination of different kinds of blocks	8.1.6
8-15	Special cases when using CRC	8.1.7
8-18	Input systems, diagrams, tables	8.2
8-18	Inaccurate specification of interpolation parameters	8.2.1
8-19	Reference point definitions	8.2.2
8-20	Path departure calculation	8.2.3
8-22	Input formats	8.2.5
8-23	Code tables	8.2.6
8-24	Programming key	8.3
8-27	Special cases	8.4
8-27	Special case: cancellation of remaining travel	8.4.1

The following assumptions are made in the programmes used for these programming instructions.

1. The decimal point is written even when it is automatically generated.
2. Block construction is in accordance to DIN 66024, DIN 66025, DIN 66217, ISO R 1056, ISO 1057, and ISO R 1058.
3. The programming examples are written in ISO code.
4. All geometric values are metric.
For conversion into inch see chapter 8.
5. The max. values given are limit values for the control. They can be limited in practice by the machine, interface, and input/output devices.
6. These programming instructions are designed for the max. functional range of the control. Functions to be realized by options are marked accordingly.
7. For better understanding preparatory functions are even programmed, if these are commands with reset position.
8. The contents of these programming instructions can be found in the fold-out program key.
9. Functions not included in this manual may be available in the control. However, this does not guarantee that these functions will be available with new equipment or in the case of service. We reserve the right to amend these instructions for technical reasons without prior notice.



1. Program format1.1 Perforated tape code

The data on the perforated tape is coded according to strictly defined guidelines. Each hole combination defines a unique character. Two perforated tape codes are permissible.

DIN 66025 (ISO)

EIA - RS 244-A

The control automatically recognizes the perforated tape format. The coding format is determined on reading the first & or EOR.

Individual perforated tapes must be coded in one of the allowable codes. It is not permissible to change codes within the same tape nor is it permissible to splice tapes together using different codes. Failure to observe the aforementioned will cause the control to signal a character parity alarm.

The characters in each code are defined to have even or odd parity.

ISO even number of holes
EIA odd number of holes

The even/odd criterion is used as a simple program check following the first character read. The block parity monitors for an even number of characters within a block of data. A block with an odd number of characters is made even by writing the characters "HT" or "SP". Block parity checking can be selected.

As an additional tape read check, the control compares the program previously read into the memory through a second read-in.

If a character mismatch occurs tape read is halted and a read error is displayed on the control operator panel. The word address tape format is defined by DIN 66025 (ISO).

1.2 Address characters

All characters listed in the perforated tape code are read by the control. However, only legal address characters are permissible for formulating the program commands and the geometric and technological data.

ISO code

Address words	C, D, F, G, I, J, K, L, M, N, P, S, T, X, Y, Z
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Letter	0, (Input of tool offset: TO=TOOL OFFSET)
Printable special characters	%, (,), +, -, /, ., @
Non-printable special characters	HT Tabulator SP Space DEL Delete CR Carriage return LF Line feed

INPUT READ	The following characters are neither processed nor stored
HT	
SP	
DEL	
CR (CR LF sequence arbitrary)	CR (generated twice after LF)
	SP (following every word)
OUTPUT TO PRINTER/PUNCH	The following characters are generated

LF is displayed as an * on CRT.

1.3 Word address system

A word consists of an address followed by a signed or unsigned digit sequence.

The word address format and thereby the input format is exactly defined.

Word address format according to ISO/R 1056, ISO R 1057, ISO 1058, DIN 66027, DIN 66025, DIN 66217, EIA R 274, variable word and block length.

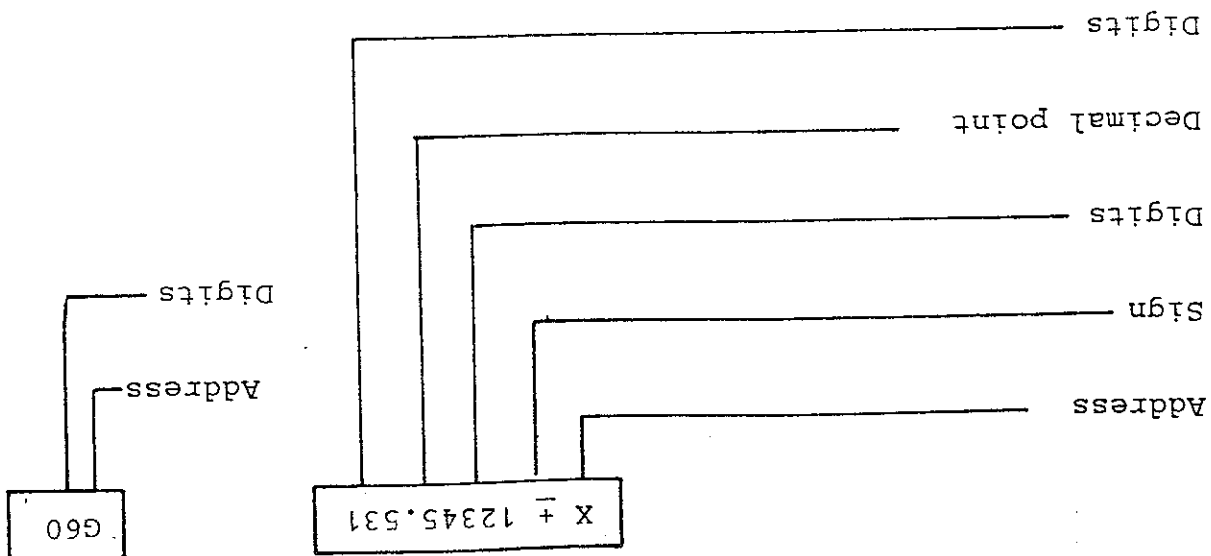
%04 N04 G02 D02 XL+053 YL+053 ZL+053
 ID053 JD053 KD053 F05 S04 T04
 R2 RL+053 L5 PD033 M02

If machine is equipped with NC rotary table or rotary index table: CL+043

Explanations:

1st address character	address				
2nd address character	L	absolute, incremental			
2nd address character	D	incremental			
Sign	+	absolute dimension value, signed positive or negative			
1st digit	0	leading zeros can be omitted: variable word length			
2nd digit	decades	adjust digit sequence			
2nd and 3rd digit	decades	adjust digit sequence			
Sign	*	End of block			

Example:



Word	1 μm 10 μm 100 μm 1000 μm 10200 μm 10000 μm
Using decimal point programming	0.001 or .001 0.01 or .01 0.1 or .1 1. or 1 10.2 100. or 100

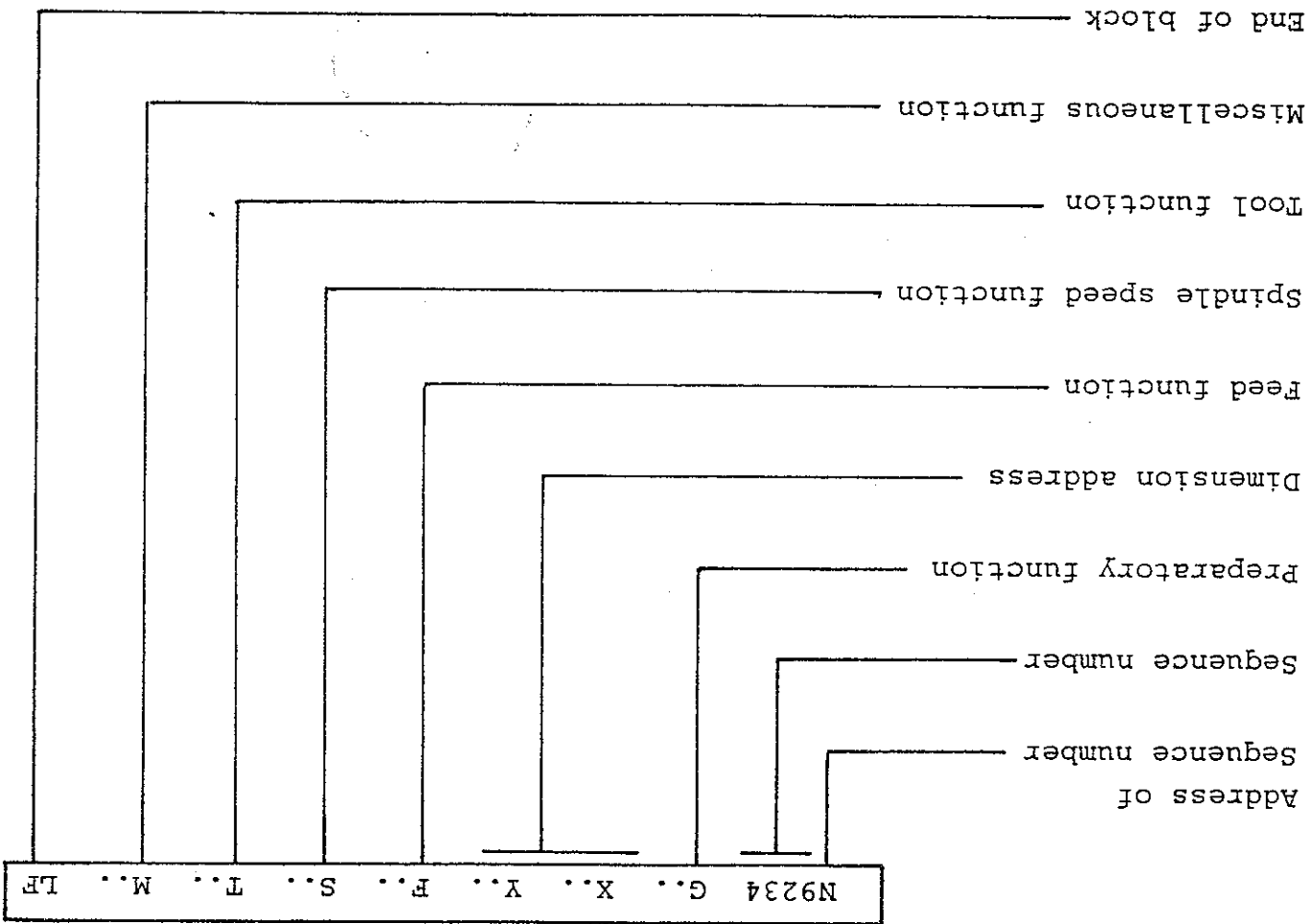
Decimal point can be programmed with the following addresses:
X, Y, Z, C, I, J, K, P, R, F

1.4 Variable block format

A block consists of several words terminated by the "End of Block" character.

Block length is variable and can have a max. of 120 characters. A total number of 80 characters can be displayed at one time.

An example of a block



The block sequence numbers need not necessarily be sequentially numbered. A numbering sequence can be interrupted arbitrarily, e.g. an edited or inserted block may have a sequence number several orders of magnitude higher than the preceding sequence number.

Program operations not necessary for every program run, such as gaging, may be identified by the block delete symbol "/". These program sections will be ignored by the control when the skip switch (Block delete) has been activated.

The block delete "/" is placed in front of the block sequence number:

/N... block delete

The block preceding the deleted block must agree with the block succeeding the deleted block. If the blocks (preceding and succeeding) do not agree, the program will execute in- correctly, when activating or deactivating the block deletion switch "skip".

1.5 Leader

The leader is used to differentiate between tapes. All characters are valid in the leader except:

- %; The automatic code recognition is initiated by % (ISO).
- EOR; The automatic code recognition is initiated by EOR (EIA).

During execution of the program, the leader is skipped by the control. The leader is not stored.

1.6 Comments

Program blocks may be clarified by using comments. It is possible for the operator to view comments on the display. Within a comment all characters except % and LF are legal. A comment may contain a maximum of 29 characters. Are more comments required, they can be programmed one after another.

Example:

N25 G70 (all following)

(Geometric values in inches)

Incorrect

Y	(FLANGE)	100.	
Y	100.	(FLANGE)	R01

Correct

X	100	(FLANGE)	Z	200.	
X	100.	R01	(FLANGE)	Z	200.

Address

Dimension

R-parameter

Comment delimiter

Comment

Comment delimiter

A comment may not be placed between an address and its dimension or between a word and its associated parameter.

1.7 Part program

A part program describes the execution of a work process and contains the part program itself with possible subroutines and/or cycle calls.

A total number of 200 part programs and subroutines can be stored in the program memory.

Program start with:

only one part program in the program memory	N25 M30 LF N20 Y-10. LF N15 X-30. LF N10 Y100. LF N5 G91 G01 X50. F100 S800 LF % LF
more than one part program in the program memory	N25 M30 LF N20 Y-10. LF N15 X-30. LF N10 Y100. LF N5 G91 G01 X50. F100 S800 LF % 1357 LF

The program number may contain a maximum of 4 digits.

1.8 Subroutines

Repetitive patterns and function cycles can be stored as subroutines which can be called arbitrarily by the part program or manual data input.

The subroutine definition is designated by 2 or 3 digits and 2 trailing zeros.

A total number of 200 subroutines and part programs¹ can be stored simultaneously in the program memory.

L41200

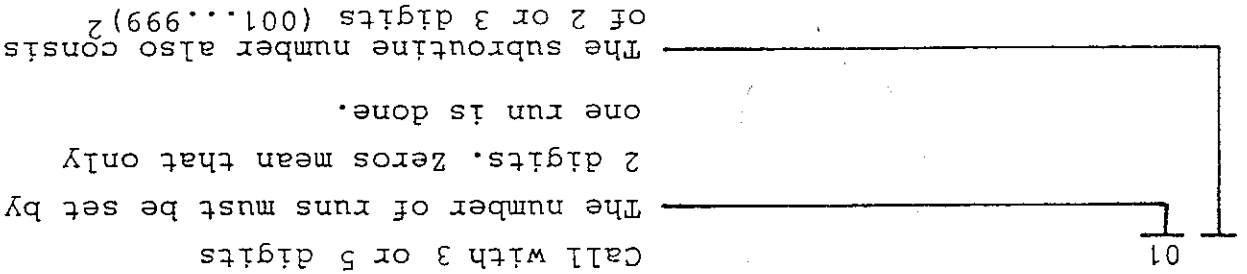
- Subroutine 412

N1 G91 G01 Y-10. F100. LF
Definition of path information,
dimensions, directions, and
feeds.

N10 X... LF

N15 M17 LF
- End of subroutine. M17 is
written in the last block.

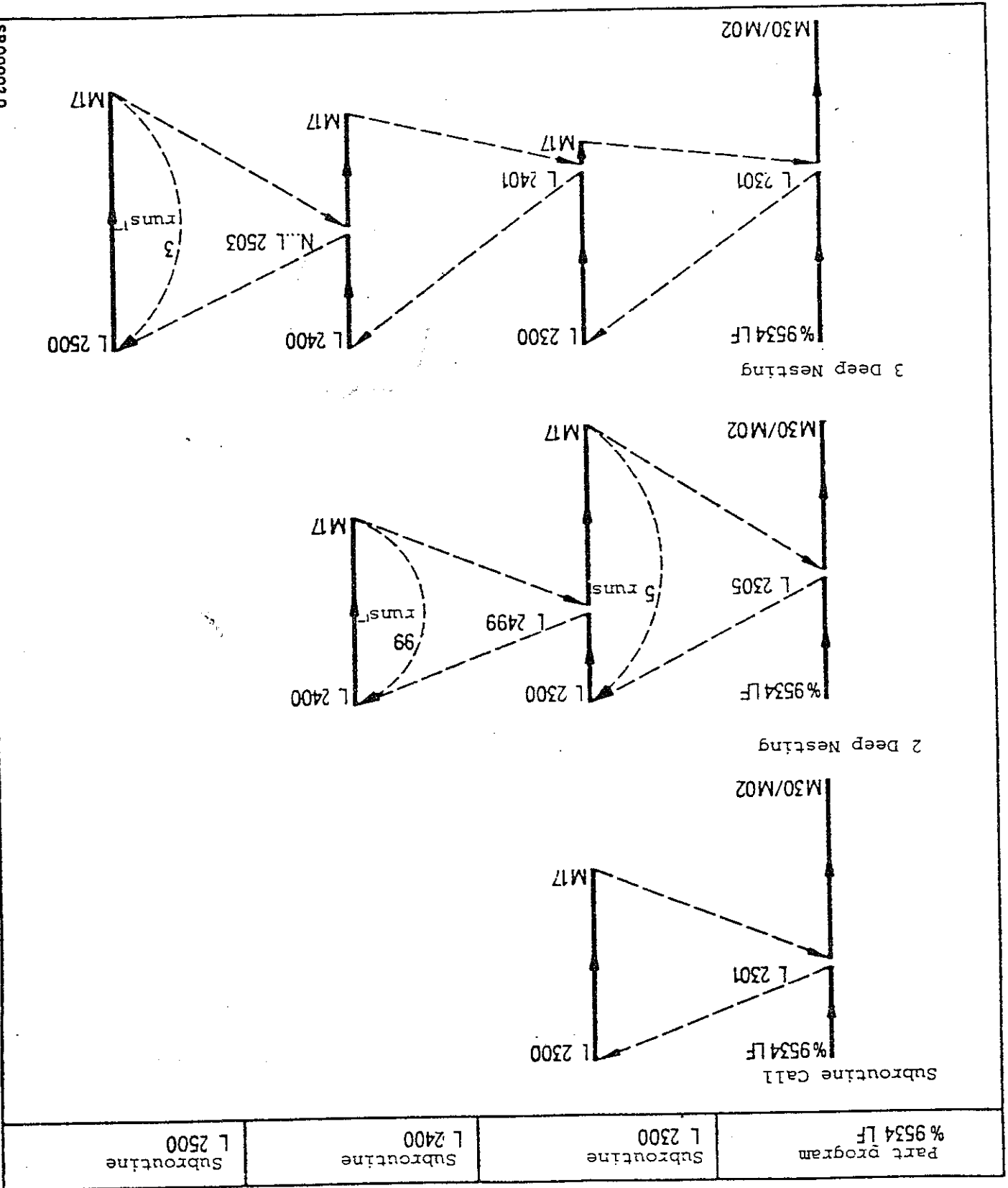
The subroutine call is made within a part program or a subroutine via the address L. A triple nesting of the subroutines is possible.



Subroutine calls are not permissible in blocks containing M02, M30 or M17. Subroutines L80 to 99 and L900 to 999³ are reserved for cycles. Automatic block search into the 1st subroutine is possible.

- (1) Basic version 3: max. 100 part programs and subroutines.
- Basic versions 0,1,2: max. 20 part programs and subroutines.
- (2) Basic versions 0,1,2,3: subroutine numbers only with 2 or 3 digits.
- (3) Basic versions 0,1,2,3: only L91 to L99.

1.9 Subroutine call, subroutine nesting



SP00002.D

1.10 Perforated tape format

Subroutine & {SR} LF | I 2300 LF | N1 LF | N2 M17 LF
 Leader Rewind stop and Subroutine 23 End of subroutine
 Block start

L 2400 LF | N1 LF | N2 (Drilling cycle) LF | N . . . M17 LF
 Subroutine 24 (Comment)
 End of subroutine

L 2500 LF | N LF | N LF | N . . . M17 LF | N . . . M02 or M30 LF
 Subroutine 25 End of subroutine End of subroutine block

Axle Program | & {1234} LF | (Gaging sequence)
 Leader Part program 1234 (Comment)

N LF | N LF | N200 M02 or M30 LF
 Part program end

& T0 LF | G92 D01 LF | G92 D02 LF | M02 or M30 LF
 Tool offsets End of tool offset load

& T1 LF S LF | S LF | M02 or M30 LF
 Testing data End of testing data load

{ } characters between brackets may be omitted.

SR Subroutine

The input sequence for the above tape is arbitrary.

The classification of the memory in part programs and sub-

rouines takes place automatically.

The tool offset is inserted in the corresponding memory zone under the code TO (Tool Offset)

1.11 Tape format for program deletion

This function can be used to delete main programs and subroutines in any sequence via the universal input/output interface.

```

DELETE PROGRAM
- Leader
%CL LF
- Identification (CLEAR)
% 1234 LF
- Delete part program % 1234
- Delete part program % 1 to % 1200
L10 LF
- Delete subroutine L10
L11 L79 LF
- Delete subroutines L11 to L79
L81 LF
- Delete subroutine L81
M30 or M02 LF
    
```

Example

% CL LF	% 1 LF	L55 LF	% 1 % 1200 LF	L11 L79 LF
Delete	Delete pro-	Delete sub-	Delete programs	Delete sub-
program	gram %01	routine L%	% 1 to 51200	outines
				L11 to L79

L81 LF M30 or M02 LF

Subroutine End of delete block

IMPORTANT

Subroutines L80 to L99 and L900 to L999¹ have to be deleted individually.

These subroutines are not deletable if cycle lock¹ is active.

(1) Basic version 4.

2. Path Information

2.1 Motion dimensions (axis commands)

The address of the motion dimension (axis command) specifies the axis in which the machine slide is to be traversed.

X, Y, Z and for the 4th axis the address C can be used.

(4th axis optional).

The 4th axis is used as a rotary axis.

This is defined via a machine parameter.

The dimension values for the rotary axis should always be programmed 3 positions to the right of the decimal point, even though input in the linear axes may be in the 10^{-4} inch system.

The rotary axis can be programmed to max $\pm 256 \times 360$ degrees (± 92159.999 degrees).

4th axis

Cutter radius compensation is not available in the rotary axis. Only the 4th axis can be used as a rotary axis.

2.2 Mirror Image

Input signals activated through the programmed auxiliary functions are used to designate which of the primary motion axis will be mirrored. The following values are inverted, or exchanged, during block processing.

Main axes

- Sign reversal of the dimension values
- Direction of rotation G02 - G03; G03 - G02
- Single axis mirrored in the CRC plane: G41 - G42; G42 - G41
- Dual axes mirrored in the CRC plane: G41 - G41; G42 - G42

Not mirrored:

- Length offset dimension
- Zero offset dimensions

Mirroring the primary motion axes (main axes) also mirrors the part contour.

The interface signal "mirror imaging" is one of the control signals supplied to the NC unit via the buffer store (see section 5.8).

Example: call of mirror imaging.

N10 G90 G54 G00 X0 Y0

N20 Z30

N30 G1 Z0 F500

N35 M22 (mirroring of X axis)

N36 G04 F... dwell greater than 1.2 times max PC cycle time

N37 a 31

N40 X50 Y50

N45 M29 (deletion of mirror image)

N46 G04 F...

N47 a 31

N50 G0 G53 Z0

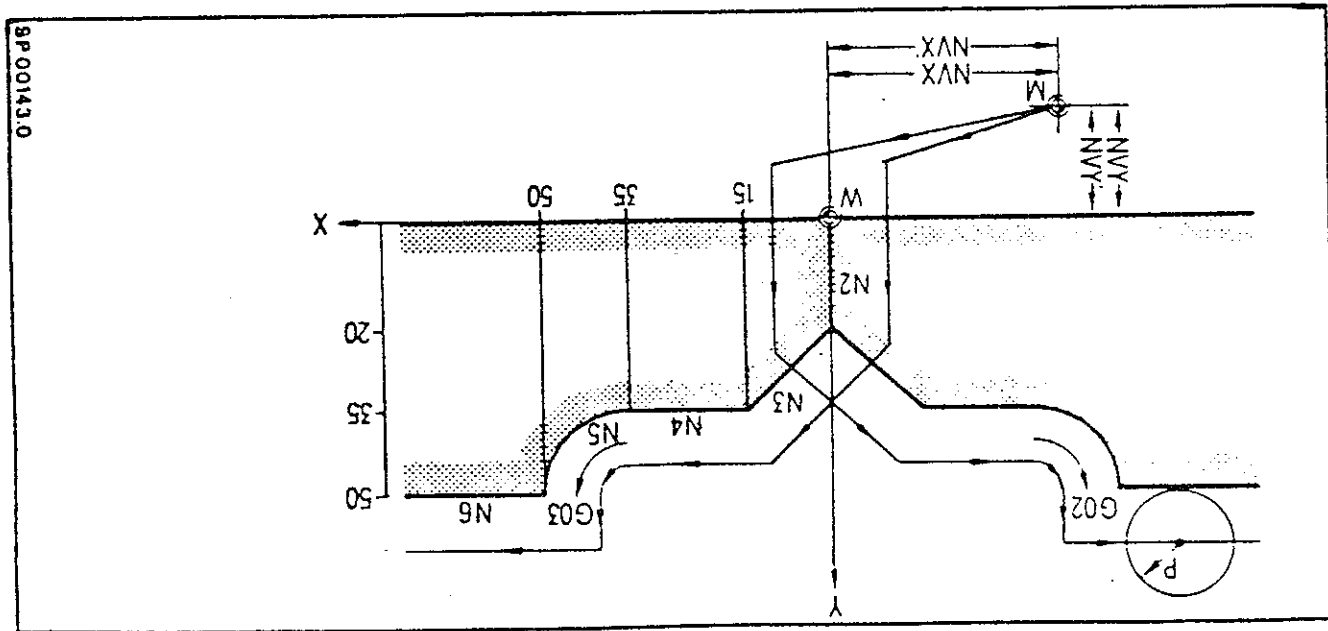
N60 G53 X0 Y0 M30

Indicating lamps (24) illuminated.

M22- mirroring of X axis
M23- mirroring of Y axis
M24- mirroring of Z axis
M29- deletion of mirror image

Example: Mirroring the X axis

Mirrored part Programmed part



M = machine zero point
 W = work piece zero point
 P = cutter radius
 NV = zero offset

 work piece contour

 the cutter centre path

If the control is equipped with the cutter radius compensation function (option), on ly the work piece contour need be programmed.

```

N1 G00 G17 G64 G41 D01 X0. Y0. S500 M03 LF
N2 G01 Y20. F100 LF
N3 X15. Y35. LF
N4 X35. LF
N5 G03 X50. Y50. I0. J15. LF
N6 G01 X...
    
```

9P001430

3. Preparatory Functions

The preparatory functions describe the manner in which the machine slide is to move, the method of interpolation, the dimensioning mode, the timed delay of program execution, and the activation of specific operational modes in the control.

The preparatory functions are categorized into groups G1 through G14 (see the programming key).

A programmed block may contain only one preparatory function from each of the 14 groups. When more than one preparatory function of the same group is programmed, the last programmed function is valid.

On control turn on, reset, or end of program, the control returns to its default state. It is not necessary to program the default preparatory functions.

Modal preparatory functions can only be altered by programming other preparatory functions from the same function group.

3.1 G90/G91 Absolute and incremental dimension programming

Absolute dimensioning G90 (Reset state 12th G-group)

In absolute dimensioning all dimensions are in reference to

the part zero dimension.

The dimension denotes the end position in the coordinate system.

Incremental dimensioning G91

An incremental dimension defines the path departure with

respect to the present position. Incremental dimensioning is

advantageous in subroutine programming.

A zero offset is always active with absolute and incremental

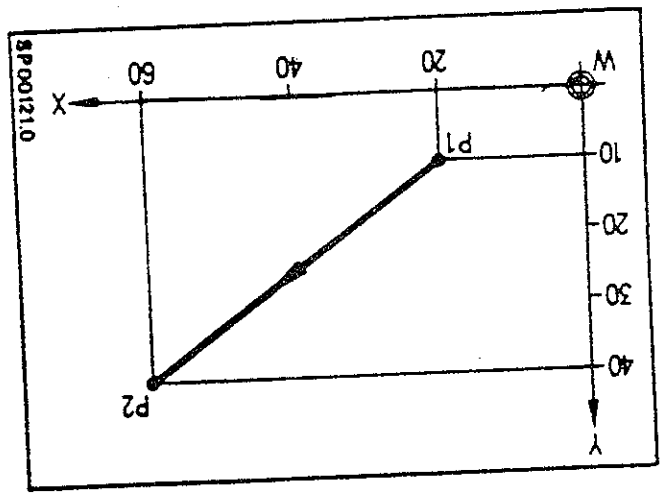
programming.

(See also chapter 8.2.3).

3.2 G00 Rapid traverse

A block programmed with G00 will traverse in a straight line at the highest possible rate to the programmed position. The control monitors each axis traverse rate so that the maximum allowable rate (machine parameter) is not exceeded.

Rapid traverse (G00) automatically includes exact positioning. 3 out of 4 axes can be moved simultaneously at rapid traverse rate. Programming G00 will not cancel the feed function. The feed function will still be active when programming a G01 following a G00.



W - part zero point

Absolute dimensioning

```
N . . . G00 G90 X60. Y40. LF
Tool traverses from P1 to P2.
```

Incremental dimensioning

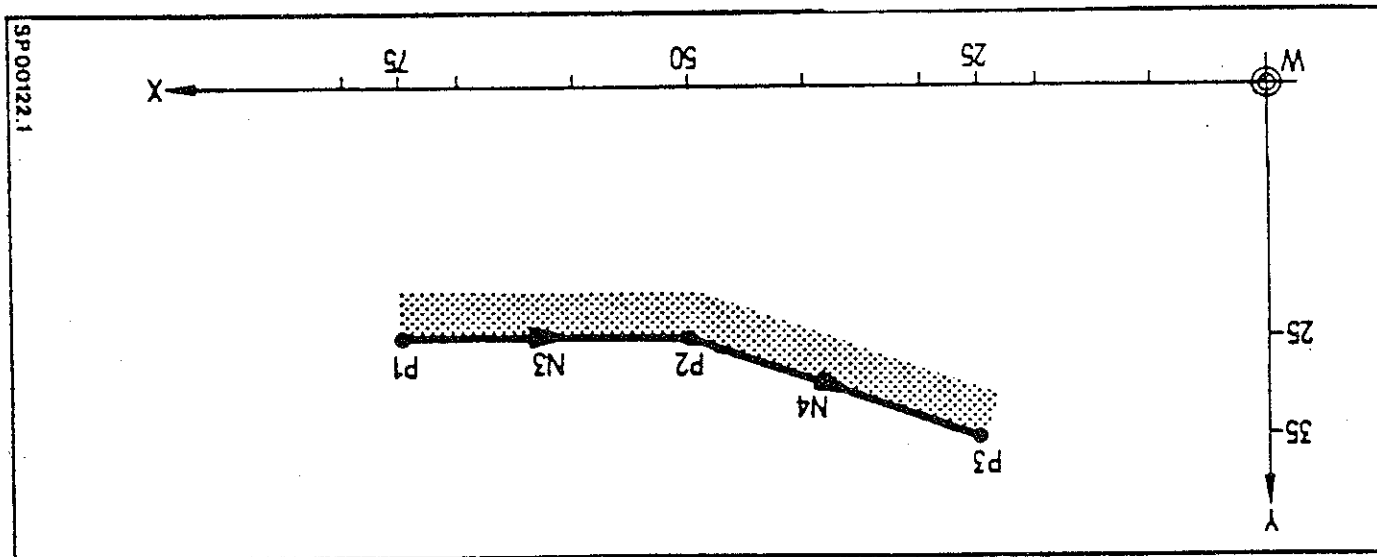
```
N . . . G00 G91 X40. Y30. LF
Tool traverses from P1 to P2.
```

3.3 G01 Linear interpolation (reset state 1st G-group)

The tool traverses with the programmed feed rate F in a straight line to the programmed end point.

Paraxial and straight line path movements at any angle are possible.

With linear interpolation 3 out of 4 axis can be simultaneously traversed (on machines with 3D interpolation).



Incremental dimensioning

```

N3 G91 G01 X-25. F1000 S2000 M3 LF Feed rate programming
N4 X-25. Y10. LF

```

at 1000 mm/min

Absolute dimensioning

```

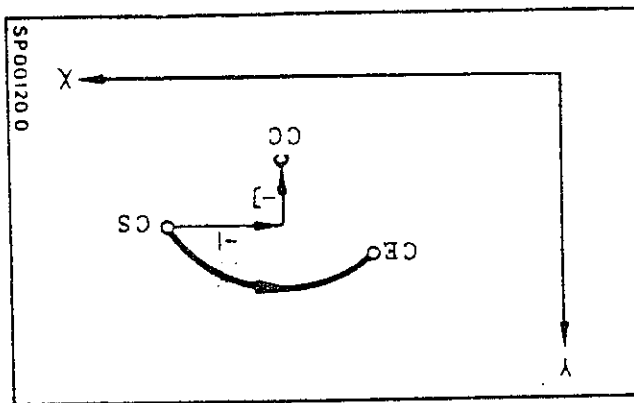
N3 G90 G01 X50. Y25. F1000 S2000 M3 LF
N4 X25. Y35. LF

```

3.4 G02/G03 Circular Interpolation

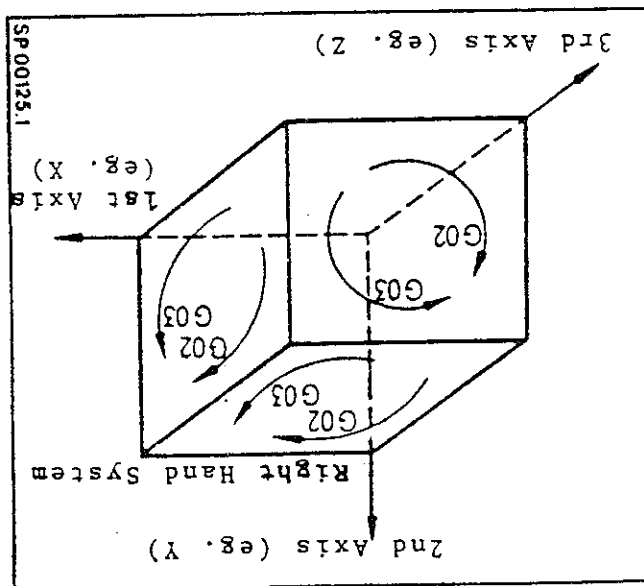
The interpolation parameters together with axis commands determine the circle or arc. The starting point "CS" is determined by the previous block. The end point "CF" is fixed by the axis values of the plane in which the circular interpolation is programmed.

The signed interpolation parameters I, J, and K determine the circle centre "CC".



Circular interpolation is possible in 2 out of 3 axes.

The direction in which the arc is traversed is determined by G0 or G03.



In order to obtain a right hand system in the 3 primary axes, they must be programmed in the following order:

X... Y...
 Z... X...
 and Y... Z...

Circular interpolation with interpolation parameters

The starting point of the circle or circular arc is determined by the previous block. The end point is fixed by the axis values programmed.

The circle centre is determined by the signed interpolation parameters.

Increment (signed)
from circle start point
to circle center point

I	parallel to X-axis
J	parallel to Y-axis
K	parallel to Z-axis

If only one axis value is programmed, the missing value is generated by the control. The value belonging to the primary axis of the selected plane (G17, G18, G19) is used.

If an interpolation parameter is not programmed, zero is assumed. Each block with G02/G03 must contain at least one axis value and one interpolation parameter.

Example

N5 G17 G42 D03.. LF Plane and tool offset selection
N10 G03 X17 Y30. I-9. J8 LF Complete definition of circle with
direction, circle end point coordinates
and interpolation parameters.

N25 G03 X17. I-9 LF Circle programming with missing addresses.
If no other plane and traverse distance
in the Y axis is programmed between
N10 and N25, the control will generate
the following:
N25 G17 G03 X17. Y30. I-9. J0. LF

3.4.2 Helical interpolation

Helical interpolation is obtainable between 3 mutually perpendicular linear axes. A block has to contain a circle and a straight line, with the straight line perpendicular to the plane in which circular interpolation is programmed. The programmed feed rate will be maintained along the circular path.

Example: semi-circle, radius 100 mm

```
N... G91 G02 X200. Y0. Z200. I100. J0. LF
```

Circular interpolation plane

(by G17, G18 or G19) -

all axes have to be programmed.

Axis for linear interpolation can be

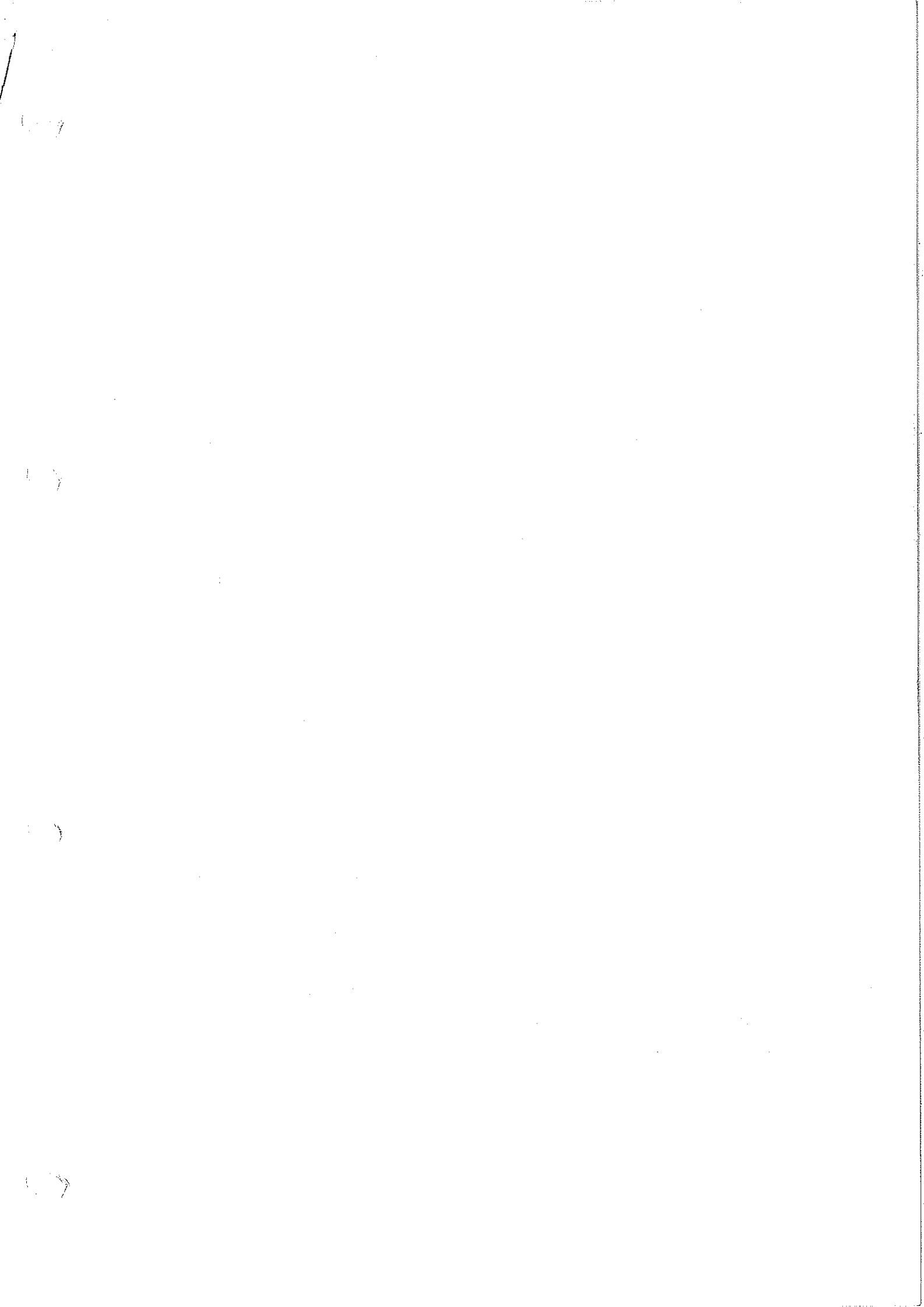
programmed before or behind the inter-

polation parameters.

Interpolation parameters

If, for example, the 4th axis has been defined as parallel to the X axis, the following combinations are obtainable:

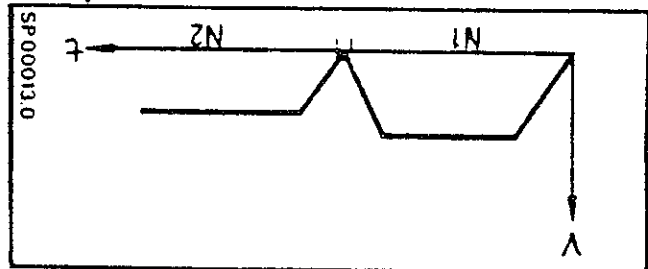
Interpolation parameters	Circle	Straight line
J I K	X Y Z	X Y Z



3.6 G60 Exact Positioning (Reset state 10th G-group)

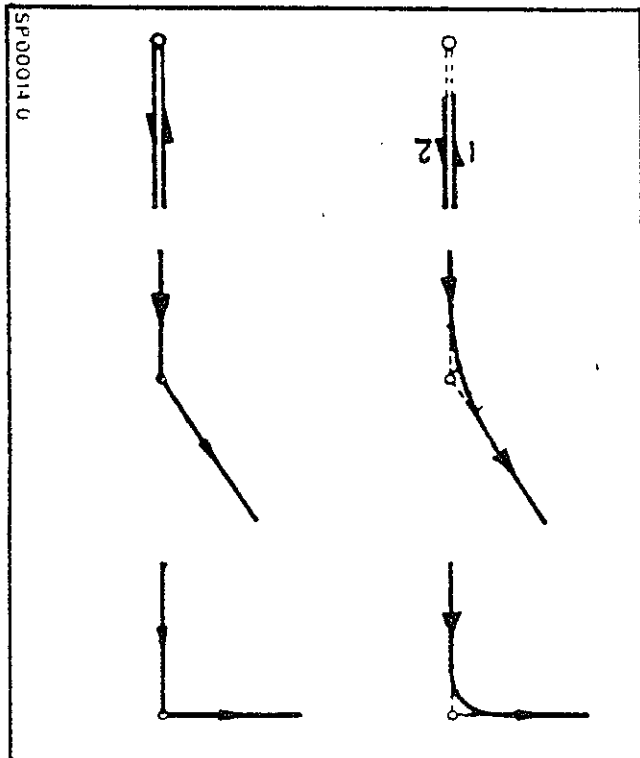
With the preparatory function G60 it is possible to position

exactly to a target position (within the "in position band tolerance"). The feed velocity is reduced to zero. The following error is worked to zero.



The preparatory function G60 is used, for example, to machine sharp corners, or when reversing direction. Blocks with G00 need not be programmed with G60. A G60 is automatically performed with G00. G60 is modal and is cancelled with G64 (contour machining), or G63 (tapping with compensation chuck).

The example shows direction reversal with and without G60:



3.7 Tapping with a floating tap holder

The preparatory function G63 is programmed when tapping drilled holes with a floating holder. The feed axis and spindle rotation are not synchronized.

Spindle speed is programmed under address S with the appropriate feed function programmed under address F. The floating tap must take up length variations resulting from the difference between the tap lead and the lead deviations due to feed rate and spindle speed fluctuations.

G63 inhibits the feed rate override switch and dependent on the interface design will shut the spindle down when "feed hold" is signalled.

G63 may only be used with linear interpolation G01.

G60 will cancel G63.

3.8 G64 Contour Machining

To prevent dwell marks, the preparatory function G64 is programmed. G64 assures smooth path transitions between contiguous blocks containing path movements, however, a tangential direction change will result in a rounded corner.

3.9 G04 Dwell Time

The dwell duration time is programmed under address F.

The dwell duration time lies between:
1ms and 99999 ms (F.001 ... F99.999)

A block containing a dwell may not contain other functions
except for G04.

Example

```
N . . G04 F11.5 LF
```

Dwell time 11.5 sec
always an unsigned number

When necessary, several continuous blocks containing dwell
functions may be programmed.

Dwell times are programmed when a tool is to cut free of the
part and may be used for speed change and machine switching
functions. G04 is not modal.

3.10 G70/G71 Input System

G70 Input dimensioning system is in inch
G71 Input dimensioning system is in metric

The default mode (reset state) is defined by a machine

parameter, set when commissioning the machine. The control
always functions internally in this set system, either in
the inch or in the metric system, independent of the input
data.

If the input data dimension is not equal to the reset state

dimension, then the respective G-function must be programmed
prior to the dimension. The control converts this dimension
into the other system. The converted value is displayed when
such a block is processed.

Data input may deviate from the reset state for one or several
blocks (or an entire program). This requires programming the
appropriate G-function in the first block and writing the reset
state again after the last block (end of program M02 or M30 will
automatically reactivate the reset state).

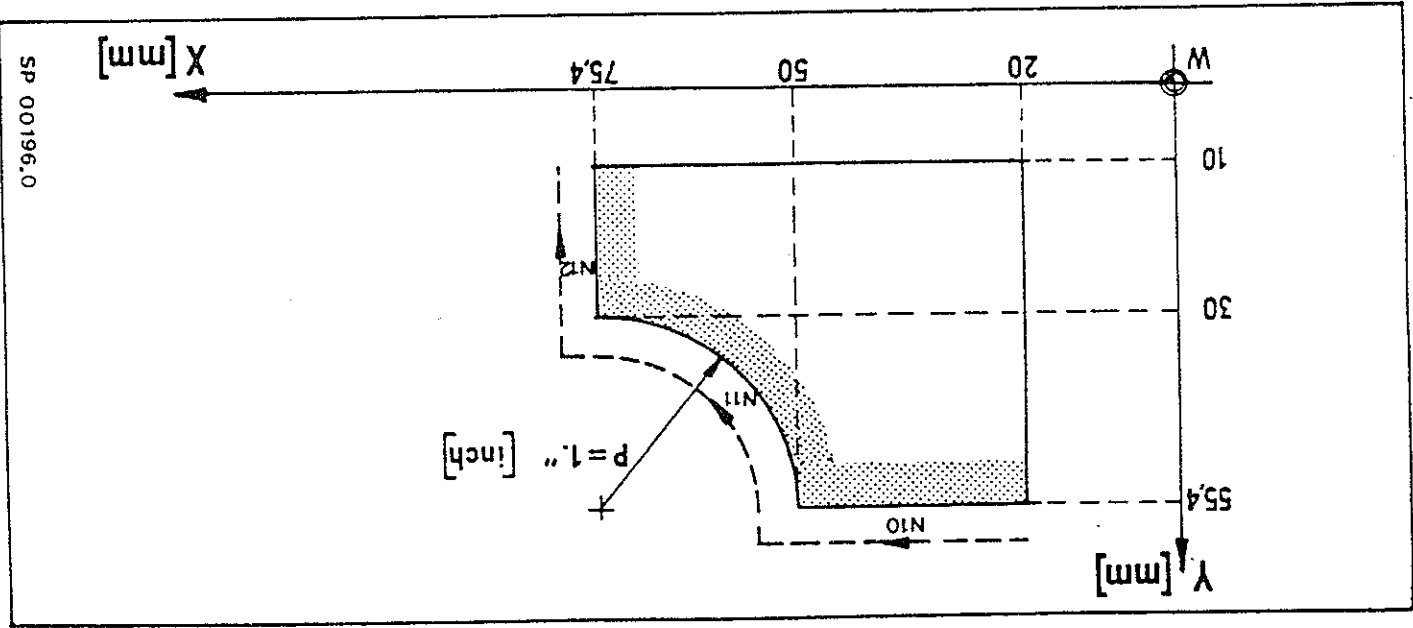
Items dependent on the reset state of the input system:
Actual-value readout (including difference actual/command value)
Zero offset
Feed rate
Tool offset

Items dependent on the programmed G70 or G71 function:

Axis values X/Y/Z
Interpolation parameters I/J/K
Radii/chamfers R/P-
Parameters associated with axis values, interpolation parameters
and radii/chamfers

.....

```
N10 G91 X30. LF
N11 G03 G70 X1. Y-1. I1. J0. LF
N12 G01 G71 Y-20. LF
```



SP 00196.0

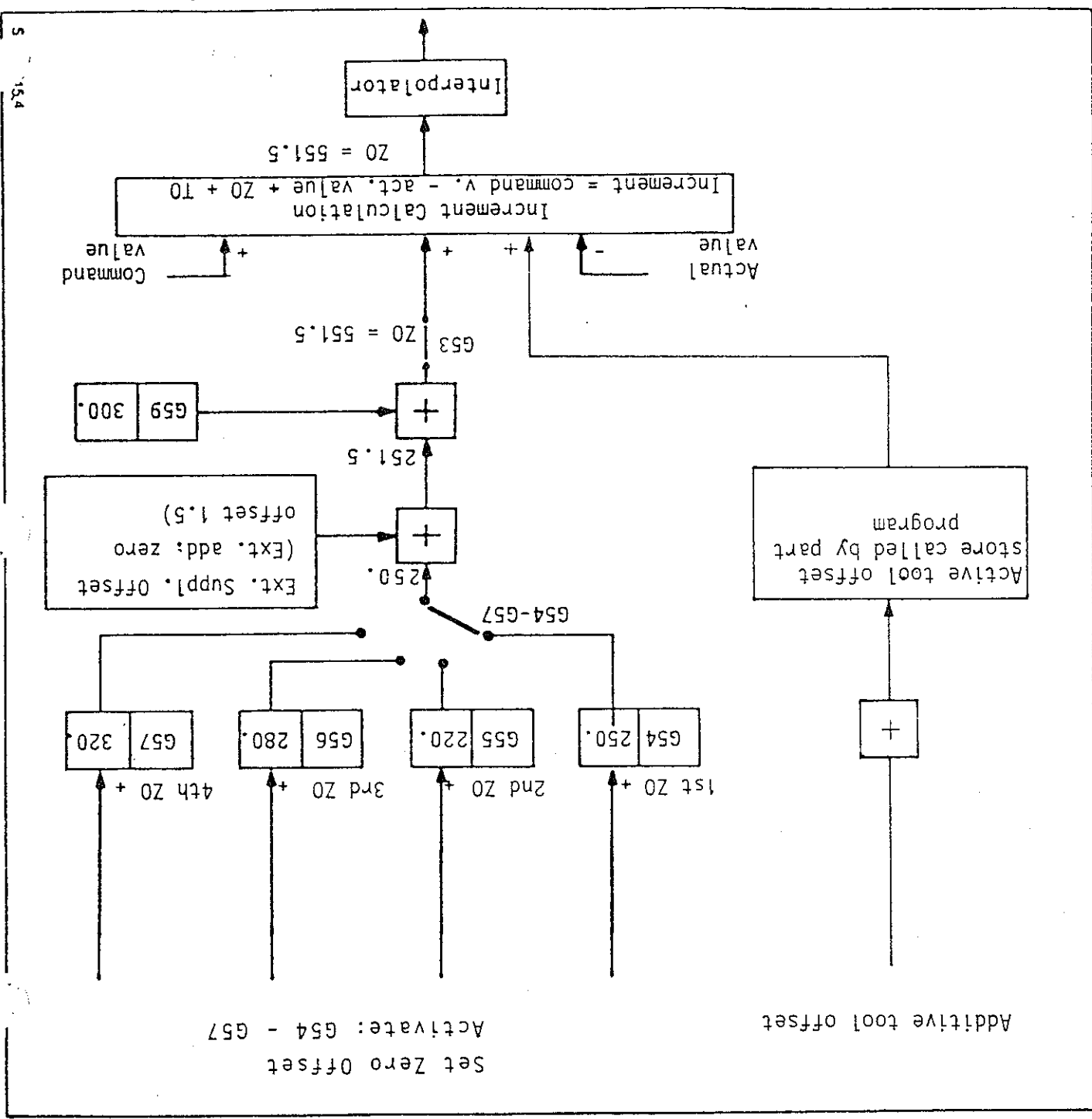
Example: G71 = reset state (metric)

3.11 Zero point offsets (ZO)

ZO = set. ZO (G54-57) + add. ZO (G59) + ext. suppl. ZO

The zero point offset is the difference between the workpiece zero point (to which the measurements are related) and the machine zero point.

Zero offset using one axis as an example:



3.11.1 G54/G55/G56/G57 Settable Zero Point Offset

(G54 is the reset state of the eighth G-group)

Values for the zero point offset for each axis can be entered into the control via the operator's panel. The offset is included when calculating the block end point in absolute and incremental data blocks, if the associated axis is programmed.

With incremental data blocks (G91) any change in zero point offset is taken into account.

Example:

Change from G54 to G55 in an incremental data block. The resulting difference between Z0 (G55) and Z0 (G54) is included in the calculation (see block increment calculation, chapter 8.2.3).

Four settable zero point offsets per axis can be selected.

An external supplementary zero point offset (ext. add. zero point offset) originating in the interface control is added to the value selected by G54 and the programmed Z0. The result is equal to the total zero offset.

Example:

The contour is programmed in absolute dimensions. To allow for finishing stock, the entire contour can be displaced in Y with a programmable (additive) zero offset.

Selected with

```
N . . . G59 Y . . . LF
```

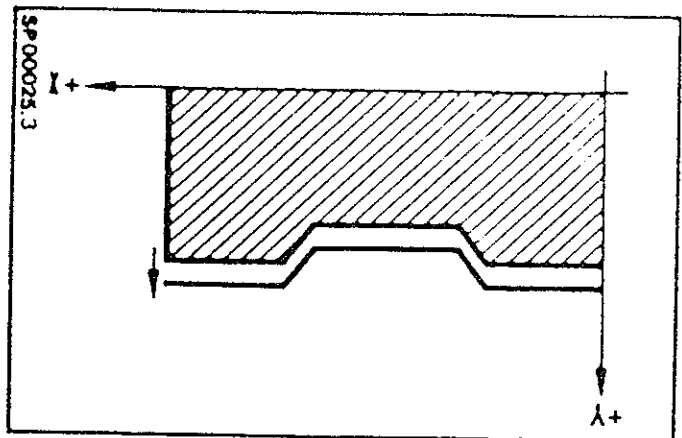
(Value input)

Cancelled with

```
N . . . G59 Y0. LF
```

(Value deleted)

Programmable additive zero offset in Y



With M02/M30 or on a program exit, the programmable zero offset is automatically cleared since a new program start will reload the offset value.

3.11.3 G53 Cancelling the zero offsets

G53 suppresses blockwise the coordinate displacement achieved by

- settable Z0 (G54 - G57)
- programmable additive Z0 (G59)
- external additive Z0

The tool offset must be separately cancelled.

In the block following G53 all zero offsets are again active.

Example: referred to machine zero point

- N1232 G40 X... cancellation of tool offset
- N1233 D00 Z... cancellation of length compensation
- N1234 G53 X... Y... cancellation of all Z0's

3.12 G94/G95 Feed F

The programmed feed rate when using cutter radius compensation is maintained on the contour surface.

- G94 feedrate F in mm/min

With a rotary axis the feed function is programmed under address F as an angular velocity in degrees/minute. The feed can be programmed in feed/minute instead of degrees/minute, however, the angular velocity and the part radius must be used to calculate circumferential velocity. For the unit circle diameter

$$D_o = 1 \text{ mm} \times \frac{\pi}{360} = 114.592 \text{ mm}$$

the resultant vectorial tangential velocity at 1°/min equals 1 mm/min.

If a rotating axis is only moving and the stationary tool tip contacts the part surface at a diameter equal to D, then the surface velocity of the tool tip relative to the part surface equals:

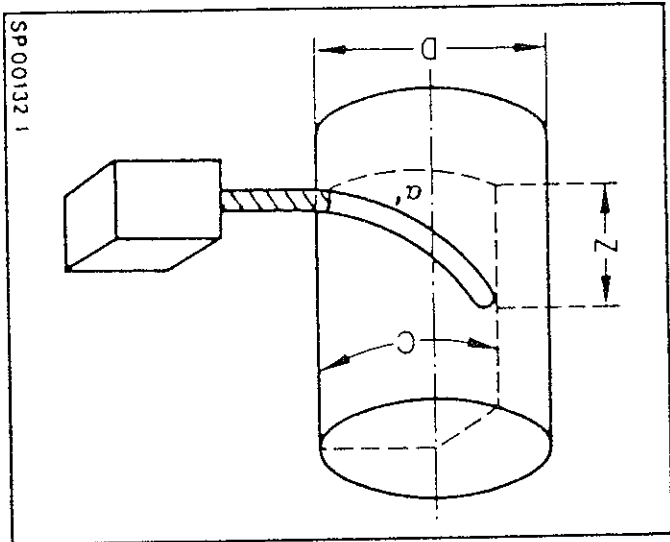
$$V_{\text{tool}}^{\text{mm/min}} = \frac{D_o}{D} \times V_{\text{programmed}}^{\text{deg./min}}$$

$$V_{\text{programmed}}^{\text{deg./min}} = \frac{D}{D_o} \times V_{\text{tool}}^{\text{mm/min}}$$

The feed rate override switch located on the operator panel can modify the programmed feed from 1% to 120%. The 100% setting corresponds to the programmed value.

The following holds true when simultaneously moving a linear and rotary axis:

Whenever the distance between the tool tip contact point and rotary axis remains constant, the magnitude of the surface tangential velocity will also be constant. A constant path velocity also results when linearly interpolating a rotary and linear axis in a path parallel to the axis of rotation (helical cutting on a cylinder). The resultant path velocity of the tool tip relative to the cylinder surface is a function of the programmed velocity, the cylinder diameter and the slope of the helix.



$$V_{\text{tool}} = V_{\text{programmed}} \sqrt{1 + \frac{D^2 - D_0^2}{D_0^2} \cos^2 \alpha}$$

$V_{\text{programmed}}$ = programmed path velocity in deg./min.

D = helix diameter in mm

D_0 = unit circle diameter = 114.592 mm

α = arc tan $\frac{C}{Z}$ (slope angle of the helix with $D = D_0$)

Z = programmed departure (mm)

C = programmed angle in degrees

$\alpha' = \text{arc tan} \frac{C \times \frac{D}{D_0}}{Z}$ (slope angle of the helix with $D \neq D_0$).

If the distance between tool tip, work surface, and the rotary axis is not held constant (e.g. spiral in a plane), the path velocity will not be constant. The path velocity will continually change as a function of the variable machining diameter.

A constant path velocity can be simulated by splitting the programmed block into several contiguous blocks in which the feed function is changed to approximately the desired velocity. A subroutine program using parameter chaining is a useful technique for velocity approximation.