SINUMERIK 3M

Basic Version 4C

Programming Guide

SINUMERIK

Edition 08.88

To: **Library (000000)**

DOCUMENT CHANGE

Date: 90/08/03

Change no: **001574**

Control Document no: **3.05.4**
SINUMERIK System 3M Basic version 4C. Programming gui

Please make the following changes to the above document and then sign and return the Change Confirmation below.

Change type:

O New issue of this document - please add to your collection

Ensure the top half of this form is either stapled into the document or filed behind the Amendment Record Sheet.

Please return the lower half after signing it.

SINUMERIK 3M
Basic Version 4C

Programming Guide

Edition 08.88

General 0

Program format 1

Path information 2

Preparatory functions 3

Miscellaneous and auxliary functions S, T, M, H 4

R Parameters 5

Programming aids 6

Cycles 7

Appendix 8

SINUMERIK®-Documentation

Key to Editions

The Editions listed below have been published prior to the current Edition.

The column headed "Amendments" lists the amended sections, at all times with reference to the previous Edition.

Edition	Order No.	Amendments
08.88	6ZB5 410-0AM02-0BA0	New Edition

Table of contents

		Page
0	General introduction to Programming Instructions	0-1
1	Program format	1-1
1.1	Punched tape code	1-1
1.2	Character set	1-2
1.3	Word structure	1-3
1.4	Block structure	1-5
1.5	Leader	1-7
1.6	Comments	1-7
1.7	Part program	1-9
1.8	Subroutines	1-10
1.9	Subroutine call, subroutine nesting	1-12
1.10	Punched tape formats	1-13
1.11	Punched tape format for erasing programs	1-15
2	Path information	2-1
2.1	Axis commands	2-1
2.2	Mirroring	2-3
3	Preparatory functions	3-1
3.1	G90/G91 Absolute and incremental dimension programming	3-2
3.2	G00 Rapid traverse	3-3
3.3	G01 Linear interpolation	3-4
3.4	G02/G03 Circular interpolation	3-5
3.4.1	Circular interpolation using interpolation parameters	3-6
3.4.2	Example for circular interpolation using interpolation parameters	3-8
3.4.3	Circular interpolation by specifying the radius	3-9
3.4.4	Example of circular interpolation by specifying the radius	3-9
3.4.5	Helical interpolation	3-10
3.5	G 33 Thread cutting	3-11
3.5.1	Constant lead tapered threads	3-13
3.5.2	Feed direction	3-14
3.5.3	Variable lead thread	3-15
3.5.3.1	Variable lead thread without G34, G35	3-15
3.5.3.2	Thread cutting G34 (increasing lead)	3-15
3.5.3.3	Thread cutting G35 (decreasing lead)	3-15
3.5.4	Multiple thread	3-16
3.5.4.1	Multiple thread via axial starting point offset	3-16
3.5.4.2	Multiple thread with spindle-specific starting angle offset (Basic control 4B, option E35)	3-16
3.5.5	Thread cuttting with a boring bar	3-17

3.6	G60 Exact stop (Reset state 10th G group)	3-19
3.7	G63 Tapping with a floating tap holder	3-20
3.8	G64 Continuous path control	3-20
3.8.1	G62 Contour Machining (rounding of block transition by reduced speed)	3-20
3.9	G04 Dwell time	3-21
3.10	G70/71 Input systems	3-22
3.11	Zero offsets (ZO)	3-24
3.11.1	G54/G55/G56/G57 Settable zero offset	3-25
3.11.2	G59 Programmable zero offset	3-27
3.11.3	G53 Cancelling the zero offsets	3-29
3.12	G94 /G95 Feed F	3-30
3.13	G17/G18/G19 Machining plane selection	3-34
3.14	Tool offset	3-35
3.15	G40/G41/G42 Intersectional cutter radius compensation	3-37
3.16	G43/G44 Tool length offset (positive, negative)	3-43
3.17	Milling of cylindrical contours G92 P	3-45
3.18	G10/G11 Polar coordinate programming	3-48
3.19	G36, G37, G38 TRANSMIT coordinate transformation	3-51
3.20	Programmable turning of coordinates	3-56
3.21	Dynamic smoothing exponent for threads	3-59
4	**Miscellaneous and auxiliary functions S, T, M, H**	4-1
4.1	S word	4-1
4.2	Tool function T	4-2
4.3	Miscellaneous functions M	4-2
4.4	Auxiliary functions H	4-4
4.5	Rapid switching and auxiliary functions	4-4
5	**R Parameters**	5-1
5.1	Parameter definition	5-2
5.2	Assigning parameters in a program	5-2
5.3	Operations with parameters	5-3
5.4	Parameter chaining	5-5
5.5	Programming examples using parameters	5-7
5.6	Empty buffer store, L99; @ 31	5-10
5.7	Branch conditions	5-11
5.7.1	@ 00 Unconditional jump	5-11
5.7.2	@ 01; @ 02; @ 03 Conditional jump	5-13
5.7.3	Concluding example for unconditional and conditional jumps	5-15
5.8	@ 10 "Square root"	5-16
5.9	@ 15 "Sine"	5-17
5.10	@ 18 "Arctan"	5-18
5.11	@ 20 "Load address parameter"	5-19
5.12	@ 90 - @ 93 "Address parameter"	5-21
5.13	@ 29 "Load/read from system memories"	5-22
5.14	@ 28 "Selecting the R parameter group" (only basic version 4C)	5-26

6	**Programming aids**	6-1
6.1	Automatic insertion of champfers and radii	6-2
6.1.1	Miscellaneous and auxiliary functions in linked blocks	6-3
6.1.2	Example: Rectangular pattern	6-4
6.2	Blueprint programming (Basic control 4B, 4C)	6-5
6.2.1	Contour cycle programming	6-6
6.2.2	Operation of the function F, S, T, H, M in contour cycles	6-11
6.2.3	Chaining of blocks	6-12
6.2.4	Examples	6-13
6.2.5	Miscellaneous and auxiliary functions in chained blocks	6-15
7	**Cycles**	7-1
7.1	Drilling/Boring cycles G81 - G89	7-1
7.2	Examples of limitations in calling boring cycles	7-8
7.2.1	Calling boring cycles in a subroutine	7-18
7.3	Listing of the boring cycles	7-19
7.3.1	Boring cycles G80 - G89 for boring axis Z (basic controls 0,1,2)	7-19
7.3.2	Boring cycles G80 - G89 for boring axis Z (basic controls 3, 4, 4B, 4C)	7-21
7.3.3	Boring cycles G80 - G89 with variable boring axes (basic controls 3, 4, 4B, 4C)	7-23
7.4	Keyway milling pattern L94	7-25
7.5	Milling rectangular shape, subroutine L95	7-26
7.6	Milling circular shape	7-28
7.7	Drilling/boring patterns	7-30
7.7.1	Subroutine: Boring pattern L97	7-30
7.7.2	Subroutine: Boring/drilling pattern L93 (up to basic control 4B) Subroutine: Boring/drilling pattern L904 (basic control 4C)	7-31
7.8	Subroutine: Milling pattern for elongated hole L98	7-32
8	**Appendix**	8-1
8.1	Cutter radius compensation	8-1
8.1.1	Selecting the CRC	8-1
8.1.2	CRC in the program	8-2
8.1.3	Repetition of G function already selected (G41/G42) with same offset number	8-6
8.1.4	Cutter radius compensation deselection	8-7
8.1.5	M00, M02 and M30 with CRC selected	8-9
8.1.6	CRC with combination of different block types	8-10
8.1.7	Special cases for CRC	8-14

8.2	Input systems, diagrams and tables	8-17
8.2.1	Incorrect input of the interpolation parameters I, J, K	8-17
8.2.2	Reference points	8-18
8.2.3	Path calculation	8-19
8.2.4	Rotational feedrate limits	8-20
8.2.5	Input formats	8-21
8.2.6	Code tables	8-22
8.3	Program key	8-23
8.4	Special cases	8-26
8.4.1	Special case "Delete distance to go"	8-26

0.1 General introduction to the programming guide

This programming guide applies to SINUMERIK 3M in the basic versions 0, 1, 2, 3, 4, 4B and 4C.

The following assumptions are made in the programs used in this programming guide.

1. The decimal point is written even when it is automatically generated.

2. Block construction is in accordance with 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 section 8.2.5 and 3.10.

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. This programming guide is designed for the maximum functional range of the control basic version 4. Restrictions in the functions for the basic versions 0, 1, 2 and 3 are given in the respective footnotes.

7. For better understanding preparatory functions are programmed even if these are commands with reset position.

8. The contents of this programming guide are included in the fold-out program key as an aid to finding the respective section.

9. Functions extending beyond the scope of this guide may be capable of operating in the controller. However, no claims are made with respect to such functions when first delivered or on servicing.

We reserve the right to alter this guide on the basis of technical modifications.

1. Program format

1.1 Punched tape code

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

DIN 66025 (ISO)
EIA-RS 244-B

The control automatically recognizes the correct format. The coding format is determined on reading the first % or EOR. Individual punched 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 error.

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

ISO even number of holes
EIA odd number of holes

The criterion of even or odd number of holes is used from the second character of the program onwards for character parity check. Simple errors can be recognized 100%.

The block parity check 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 check can be deselected.

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 assignment is defined by DIN 66025 (ISO).

7. For better programm positi ... e punched tape code are read by the ... executable block is assembled using only ... ters.

8. Th

Address characters	A, B, C, D, F, G, H, [1]I, J, K, L, M, N, P, R, S, T, U, V, W, X, Y, Z
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alphanumeric character	O (for inputting TO = tool offset)
Printable special characters	%, (,), +, -, /, ., @
Non-printable characters	HT Tabulator SP Space DEL Delete CR Carriage return LF Line feed

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

LF is displayed as an *

1) for basic control 4

1.3 Word structure

A Word consists of an address letter and a series of digits with or without a sign (Address notation).
The word structure and therefore the input format are exactly and precisely defined.

Address notation in accordance with ISO/R1056,
ISO R1057, ISO R 1058, DIN 66027,
DIN 66025, DIN 66217, EIA R 274,
variable length of words and blocks

The metric values are as follows:

Basic control 0, 1, 2:	%04 N04 G02 D02 XL+043 YL+043 ZL+043 ID043 JD043 KD043 F05 S04 T04 R2 RL+043 L4 PD033 M02
Basic control 3:	%04 N04 G02 D02 XL+053 YL+053 ZL+053 ID053 JD053 KD053 F05 S04 T04 R2 RL+053 L4 PD033 M02
Basic control 4, 4B:	%04 N04 G02 D02 XL+053 YL+053 ZL+053 ID053 JD053 KD053 F05 S04 T04 H04 R2 RL+053 L5 PD033 M02 AL035
Basic control 4C:	%04, N04 G02 D02 XL+053 YL+053 ZL+053 ID053 JD053 KD053 F05 S04 T04 H+04 R2 RL+053 L5 PD033 M+02 AL035

Explanations:

First character	Address	
Second character	L	absolute/incremental
Second character	D	incremental
Symbol	+	Absolute dimension with positive or negative sign
First digit	0	leading zeros not required: variable word length
Second digit	decade	part of number
Second and third digit	decade	part of number (Coordinate values X, Y, Z, I, J, K in mm)
Symbol	*	Block end

Word examples:

X + 2345.531

X	Address
+	Sign
2345	Digits
.	Decimal point
531	Digits

G60

G	Address
60	Digits

Value	Decimal point entry:
1 µm	0.001 or .001
10 µm	0.01 or .01
100 µm	0.1 or .1
1000 µm	1. or 1
10200 µm	10.2
100000 µm	100. or 100

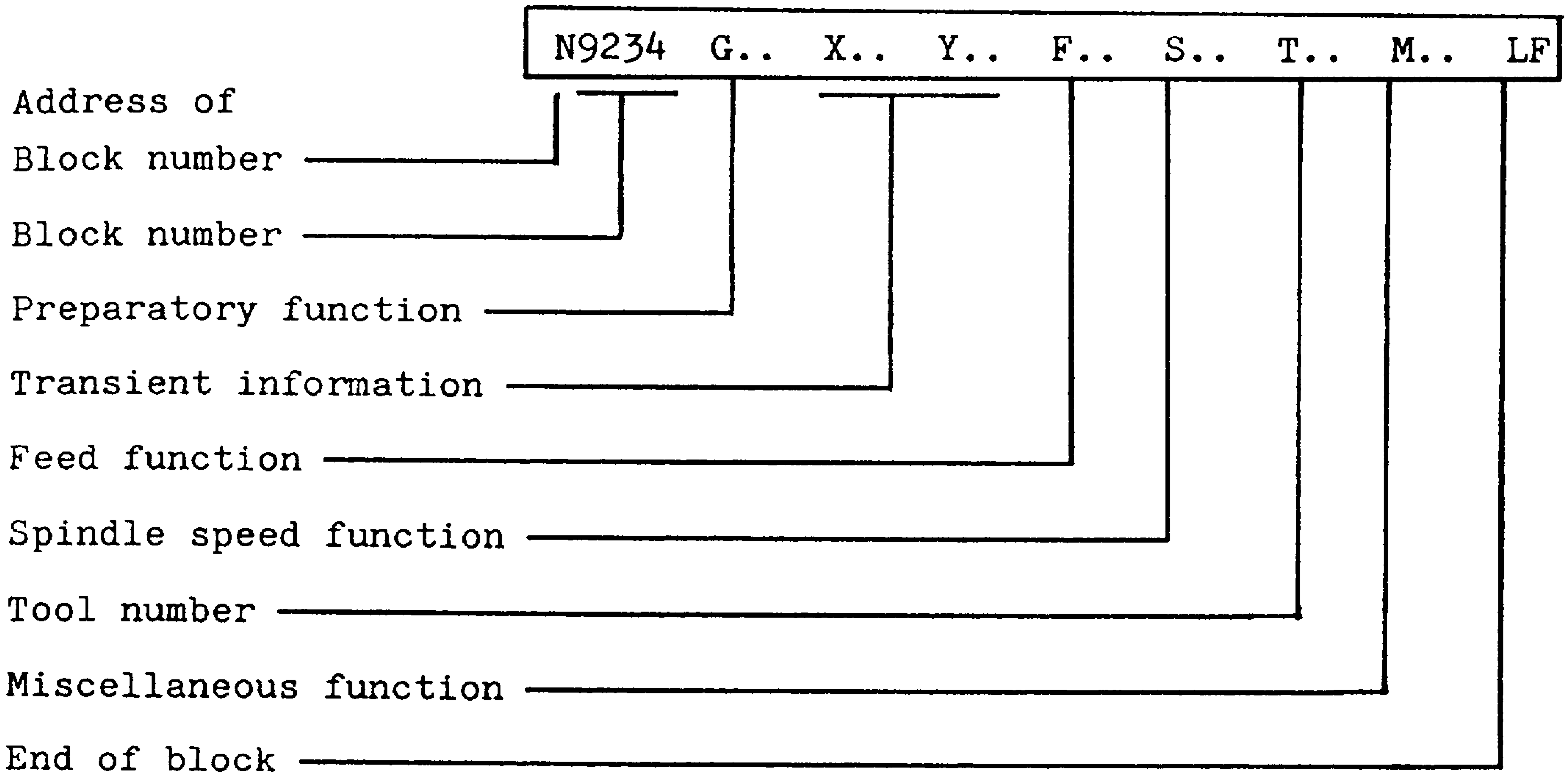
Decimal point entry is possible using the following addresses:
X, Y, Z, A, B, C, U, V, W, I, J, K, P, R, F (mm/rev) see 8.2.5.

1.4 Block structure

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 guaging, may be identified by the block delete symbol "/". These program sections will be ignored by the control when the "Block Skip" switch has been activated.

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

/N....   Skippable block

The blocks skipped must have the same start and end points, otherwise the program will execute incorrectly, when activating or deactivating the "block skip" switch.

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 screen/display.

Within a comment all characters except % and LF are legal. But no block number used in the program may be included, because the bracket contents are read for jump functions and for block search.
A comment may contain a maximum of 29 characters. If this is insufficient, several comments can be programmed one after another.

Example:

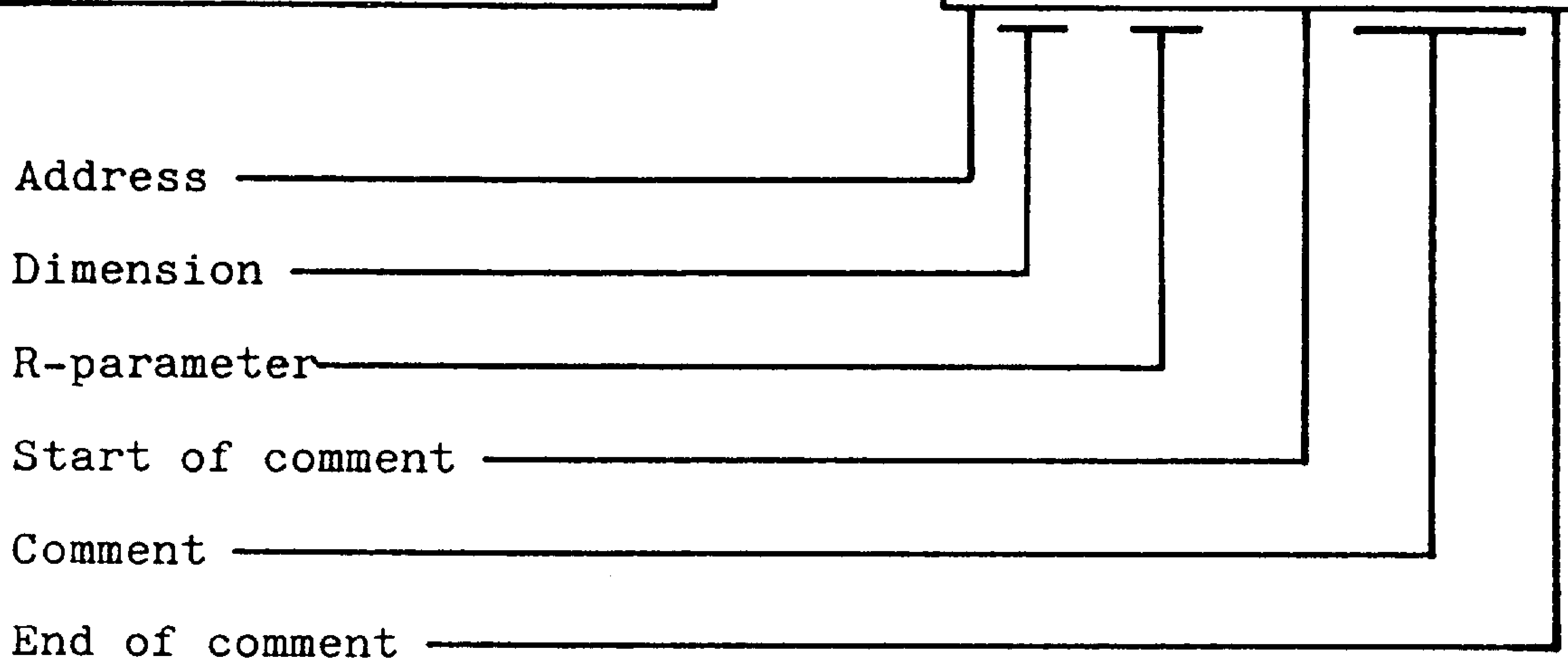
N25 G70 (all following)
(Geometric values in inches)

Incorrect

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

Correct

```
X100.         ( FLANGE ) Z 200.
X100.  R01    ( FLANGE ) Z 200.
```



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

1.7 Part programs

A part program describes the execution of a machining process and contains the part program itself with possible subroutines and/or cycles called.

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

The program number may contain a maximum of 4 digits.

*) With basic control 3: max. 100 part programs and subroutines

With basic controls 0, 1, 2: max. 20 part programs and subroutines

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 decade number and 2 trailing zeros. Part programs and subroutines can be filed in memory simultaneously in the following quantities:

Basic controls 0, 1, 2:	20 part programs and subroutines
Basic control 3:	100 part programs and subroutines
Basic control 4:	200 part programs and subroutines

L 41200					- Subroutine 412 Always without block no. and M17
N1	G91	G01 Y-10.	F100.	LF	Definition of path information, dimensions, directions, and feeds.
N5		Y... X10.		LF	
.					
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. The subroutines can be nested three-deep in the part program.

The cycle call is as follows for basic control 4:

L 412 05 — 5-digit call with number of passes

Number of passes must be input as 2 digits (01...99).
Subroutine number can be input as 3 digits (001...999) or <u>must</u> be input as 2 digits (01...99).

L 412 __ — 3-digit call without number of passes

Omission of the number of passes means that the subroutine is only executed once.
Subroutine number 2-digit or 3-digit

The cycle call is as follows for basic controls 0, 1, 2, 3:

L 41 05 — 4-digit call with number of passes

Number of passes <u>must</u> be input as 2 digits (01...99).
Subroutine number <u>must</u> be input as 2 digits (01...99).

L 41 __ — 2-digit call without number of passes

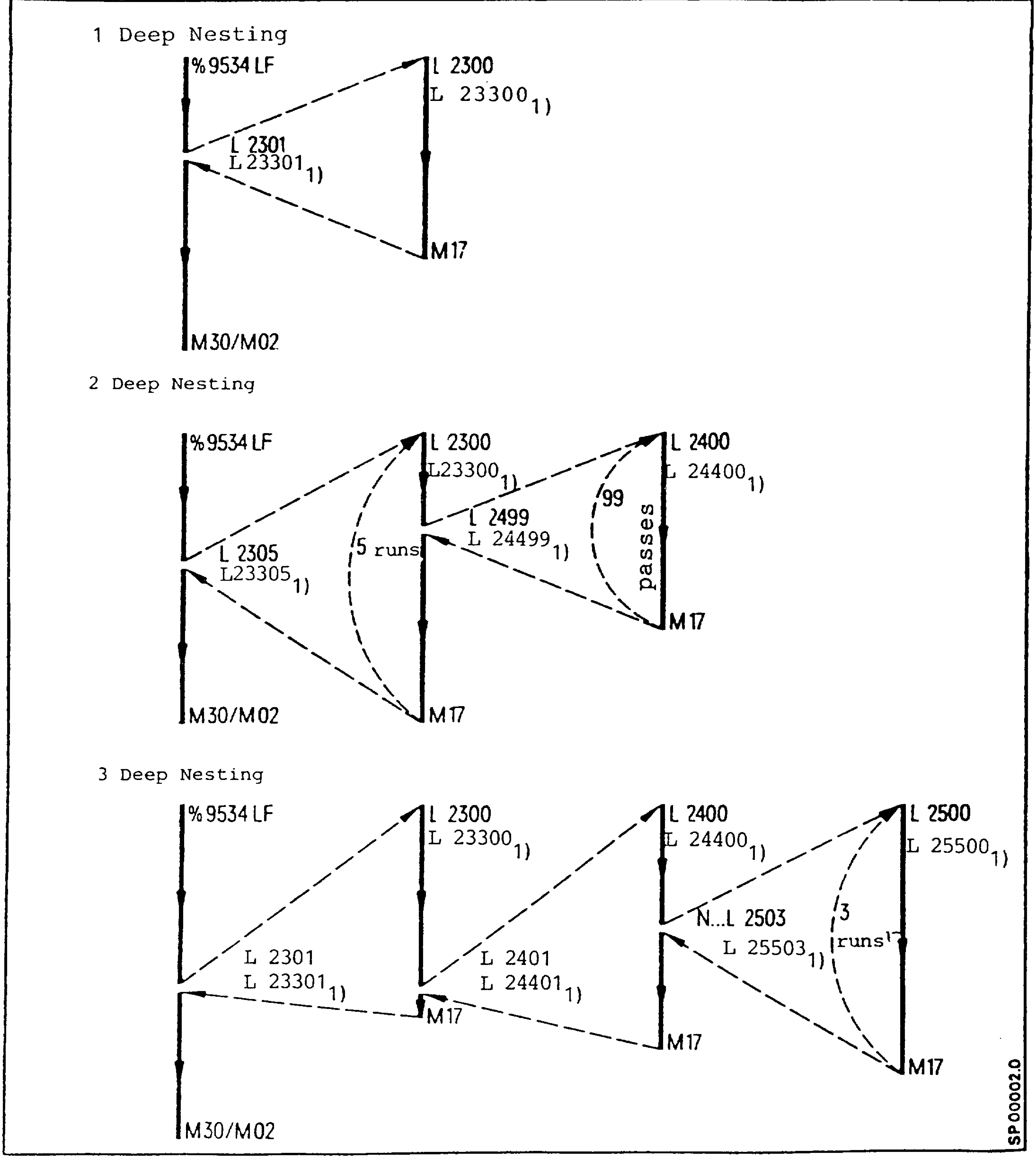
Omission of the number of passes means that the subroutine is only executed once.
Subroutine number 2-digit (01...99)

The subroutine call must not appear in the same block as M02, M30 or M17. Subroutines L80-L99 and L900-L999 [1] are reserved for cycles. Automatic block search is possible to the 1st subroutine level.

1) In the case of basic controls 0, 1, 2, 3: Only L91 - L99

1.9 Subroutine call, subroutine nesting

Part program % 9534 LF	Subroutine L 2300 1)	Subroutine L 2400 1)	Subroutine L 2500 1)



1) only with basic control 4

1.10 Punched tape format

Subroutine % {SR} LF / L 2300 LF / N1 ....LF / N2 M17 LF

Leader — Rewind stop and subroutine 23 Block start — End of subroutine

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

Program shaft / % {1234} LF /(Gauging sequence)

Leader — Part program 1234 — (Comment)

N.. ... LF / N.. ... LF / N200 M02 or M30 LF

Part program — Part program end

% TO LF / G92 D01 .... LF / G92 D02 .... LF / M02 or M30 LF

Tool offsets — End of tool offset load

% TE LF / N... S... LF / N... S... LF / M02 or M30 LF

Machine data Test data — End of machine data block

characters between brackets may be omitted.

SR Subroutine

% R LF | R00 12345.678 LF | R01 5LF ... | M02 or M30 LF

R parameter [1]

R parameter block end

% H LF | H00 12345.678 LF | H01 5LF ... | M02 or M30 LF

Background memory [1]

Background memory block end

The input sequence for the above tape is arbitrary.

The classification of the memory in part programs and subroutines takes place automatically.

The tool offset is inserted in the corresponding memory area with the TO (Tool Offset) identifier.

1) only with basic control 4C

1.11 Punched tape format for erasing programs

With this function, main programs and subroutines can be erased in any sequence via the universal input/output interface.

ERASE PROGRAM	- Leader
% CL LF	- Identifier (CLear)
% 1234 LF	- Erase parts program %1234
% 1 % 1200 LF	- Erase parts programs %1 to %1200
L10 LF	- Erase subroutine L10
L11 L79 LF	- Erase subroutines L11 to L79
L81 LF	- Erase subroutine L81
M30 or M02 LF	

Example:

% CL LF	% 1 LF	L55 LF	% 1 % 1200 LF	L11 L79 LF
erase programs	erase program % 1	erase subroutine L55	erase programs % 1 to % 1200	erase subroutines L11 to L79

L81 LF	M30 or M02 LF
erase subroutine L81	End of erase block

Caution:

Subroutines L80 - L99 and L900-L999 1) must be erased individually.
These subroutines cannot be erased if cycle inhibit is on 1).

1) with basic control 4, 4B, 4C

2. Path Information

2.1 Axis commands

The address of the axis command specifies the axis (X, Y, Z) in which the dimension should be traversed. The addresses A, B, C, U, V and W can be used for the 4th axis.

The 4th axis may also be used as a rotary axis. This is set via a machine parameter.

The dimension values for the rotary axis must always be programmed 3 positions to the right of the decimal, even though the rest of the dimension input is in the 10^{-4} inch system.

The rotary axis can be programmed to max. ± 27 x 360 degrees (9720.000°). 1)

The rotary axis can be programmed to max. ± 256 x 360 degrees (± 92159.999°). 2)

With the "Modulo 256" function the rotary axis can be turned endlessly. The actual value is reset to zero after ± 256 x 360 degrees (± 92159.999 degrees). In a single block the rotary axis can be programmed to a max. value of ± 256 x 360 degrees.

Several axes can be defined as endless rotary axes simultaneously. 3)

With the "Modulo 1 rev" function, rotary axes can turn endlessly. The actual value is reset to zero degrees after ±1 times 360 °. Several axes can be defined as endless rotary axes simultanously. 4)

1) with basic controls 0, 1, 2
2) with basic controls 3, 4
3) with basic control 4
4) With basic control 4B, 4C

Fourth axis

The fourth axis may be defined as being parallel to the X, Y, or Z axis via a machine parameter.

Which of the parallel axes is the secondary motion axis (e.g. Z or 4th axis) is determined by a signal from the interface. This signal may not change state after the start of the program.

Circular interpolation between parallel axes is not possible. The secondary axis can be used in place of its primary axis for circular interpolation.

Only the 4th axis can be used as a rotary axis. [1)]

1) with basic controls 0, 1, 2, 3

2.2 Mirroring

The axes defined as primary axes during commissioning can be mirrored by acknowledging programmed auxiliary functions via the interface. The following values are inverted or exchanged during block preparation.

For the primary motion axes the following holds true

- Sign of the programmed values
- Rotation direction G02 → G03; G03 → G02
- Single axis mirrored in the CRC plane; G41 → G42; G42 → G41.
- Two axes mirrored in the CRC plane: G41 → G41; G42 → G42,
  G02 → G02, G03 → G03

The following values are not mirrored:

- Length offset dimensions
- Zero offsets

The workpiece is always mirrored when mirroring the primary motion axes.

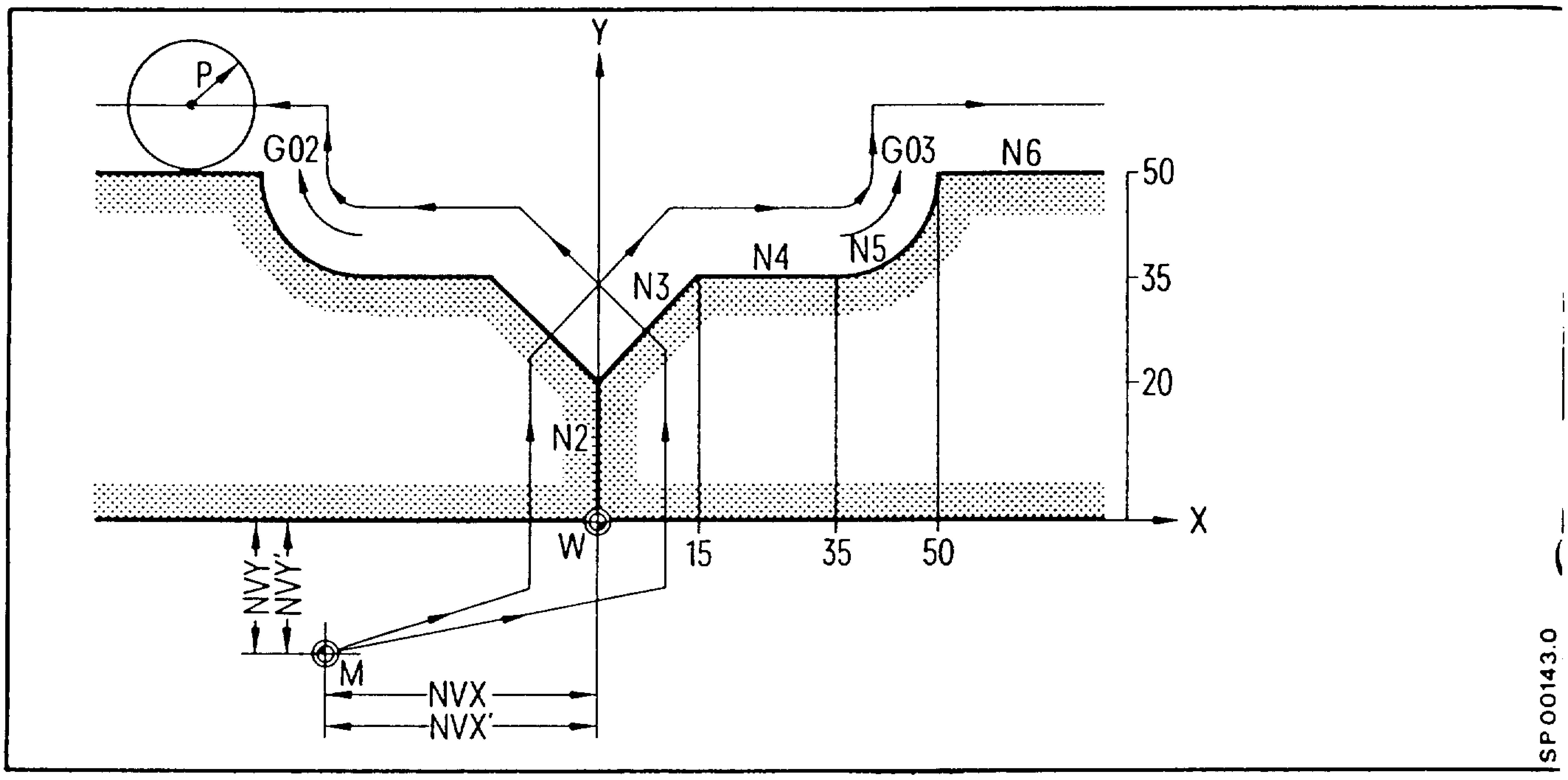
The "mirror" interface signal is transmitted to the NC via the buffer memory (see section 5.6).

Example: Mirroring selection

```
N10 G90 G54 G00 X0 Y0
N20 Z30.
    G1 Z0.F500
N35 M..  (X-axis mirroring)
N36 G04 F.. (dwell time longer than 1.2 times max. PLC cycle time)
N37  @ 31
N40 X50. Y50.
N45 M.. (deselect axis mirroring)
N46 G04 F..
N47  @ 31
N50 G0 G53 Z0.
N60 G53 X0. Y0.M30
```

Example: Mirroring the X axis

Mirrored workpiece   Programmed workpiece



M = machine zero

W = workpiece zero

P = cutter radius

NV = zero offset

——— workpiece contour

– – – the cutter centre path

If the control is equipped with the cutter radius compensation function (option), only the workpiece contour need be programmed.

```
N1  G00  G90  G17   G64  G41  D01  X0.  Y0.  M03  S56  LF
N2  G01       Y20.                                     LF
N3       X15. Y35.                                     LF
N4       X35.                                          LF
N5  G03  X50. Y50.  I0.  J15.                          LF
N6  G01  X...
```

3. Preparatory functions

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

The preparatory functions are divided into groups (see program key).

A programmed block may contain only one preparatory function from each group, otherwise the last programmed function of a group is valid.

The default states are valid when the control is switched on, reset, or at the end of program. It is not necessary to program the default preparatory functions.

Modal preparatory functions can only be overwritten by other preparatory functions from the same function group. Hence they remain effective until another code from the same group is selected.

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 workpiece zero.
The dimension denotes the end position in the coordinate system.

Incremental dimensioning G91

An incremental dimension defines the path to be traversed with respect to the present position. Incremental dimensioning is advantageous in subroutine programming.

A zero offset is always active with absolute and incremental programming.

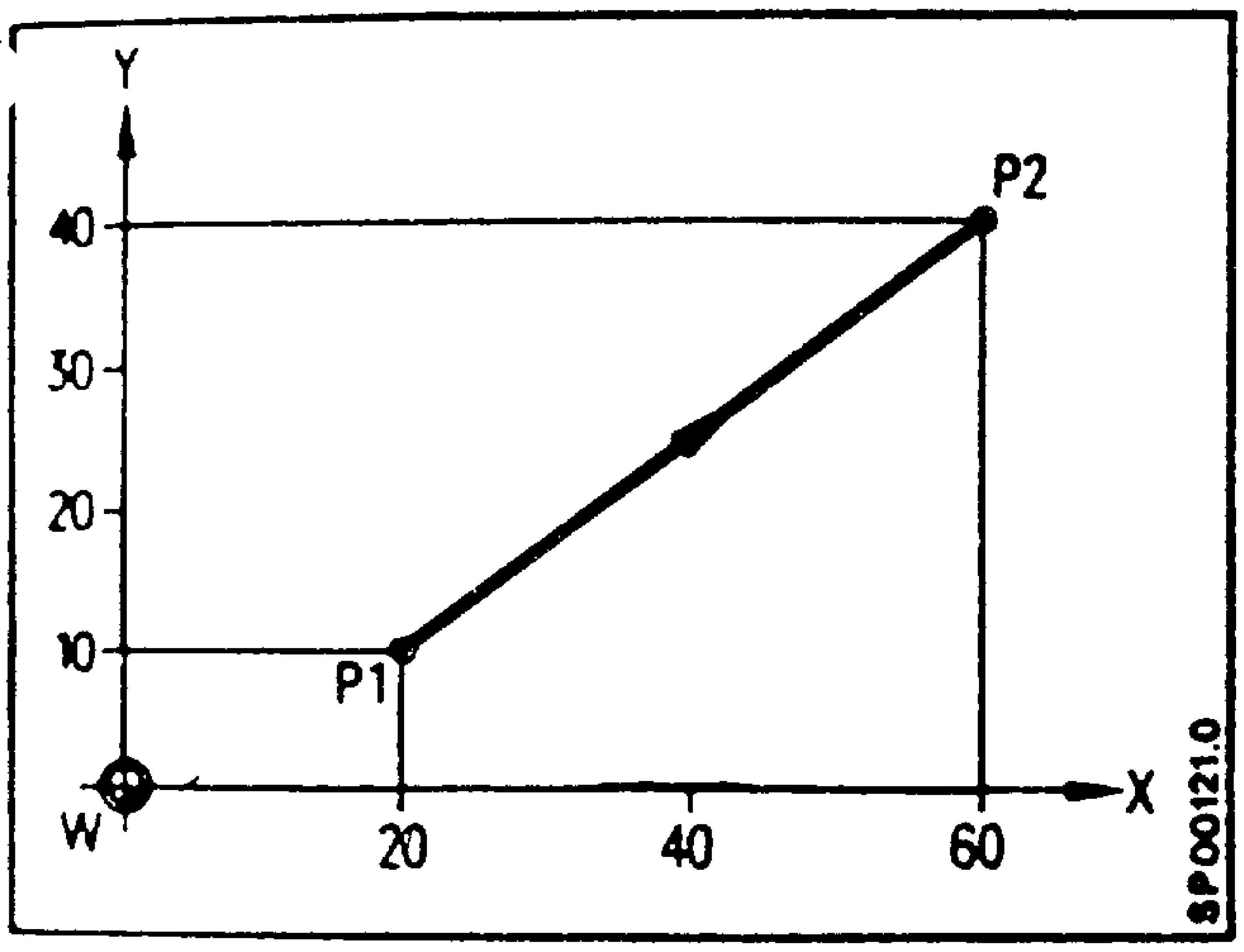
(See also section 8.2.3).

3.2 G00 Rapid traverse

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

The preparatory function, rapid traverse (G00), includes automatic exact stop.

Programming G00 does not cancel the feed function. The feed function will still be active when programming a G01 following a G00.



W = workpiece zero

Absolute dimension programming

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

Incremental dimension programming

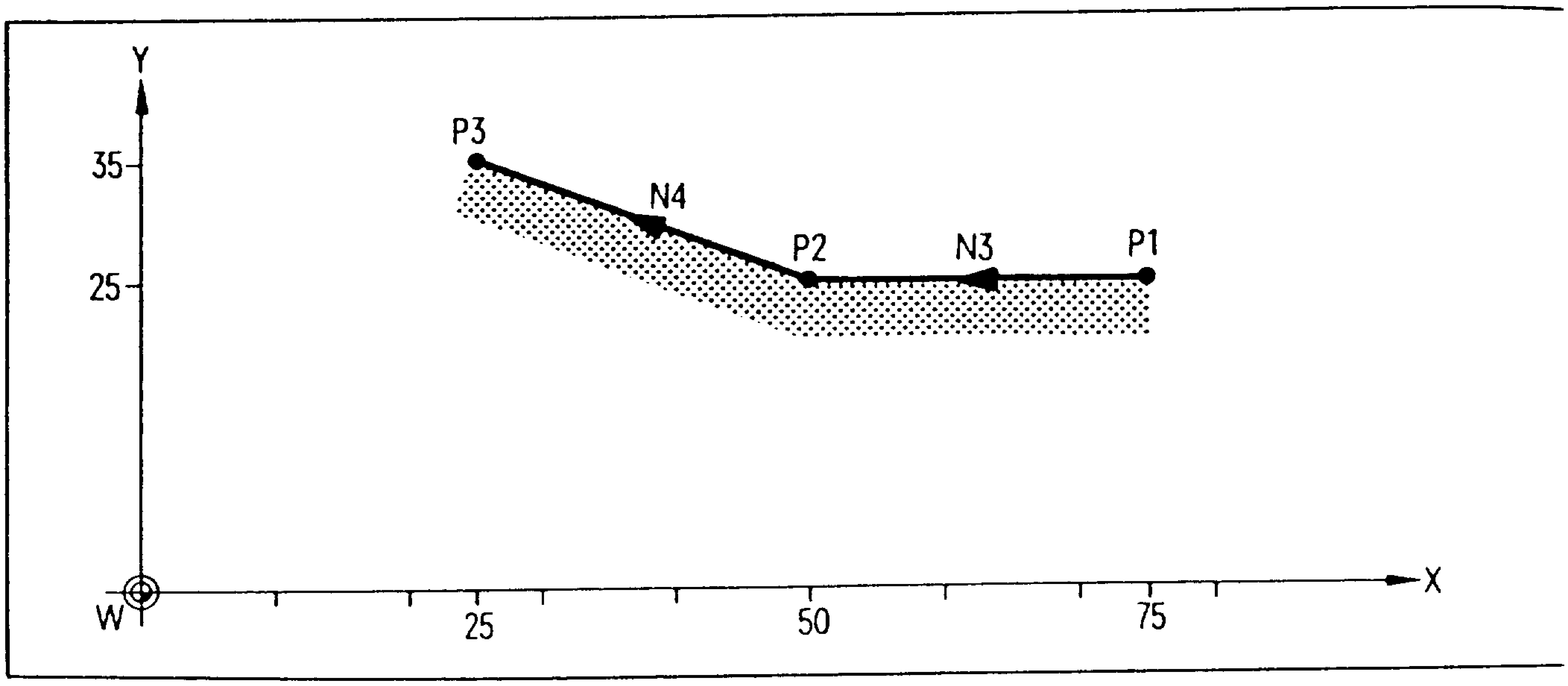
N... G00 G91 X40. Y30. LF Tool traverses from P1 to P2.

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 2 out of 4, or with 3D interpolation 3 out of 4 axes can be traversed [1].



Incremental dimension programming

N3	G91	G94	G01	X-25.	F1000	LF	Feedrate programming
N4				X-25.	Y10.	LF	at 1000 mm/min

Absolute dimension programming

N3	G90	G94	G01	X50.	Y25.	F1000	LF
N4				X25.	Y35.		LF

1) with basic controls 3, 4

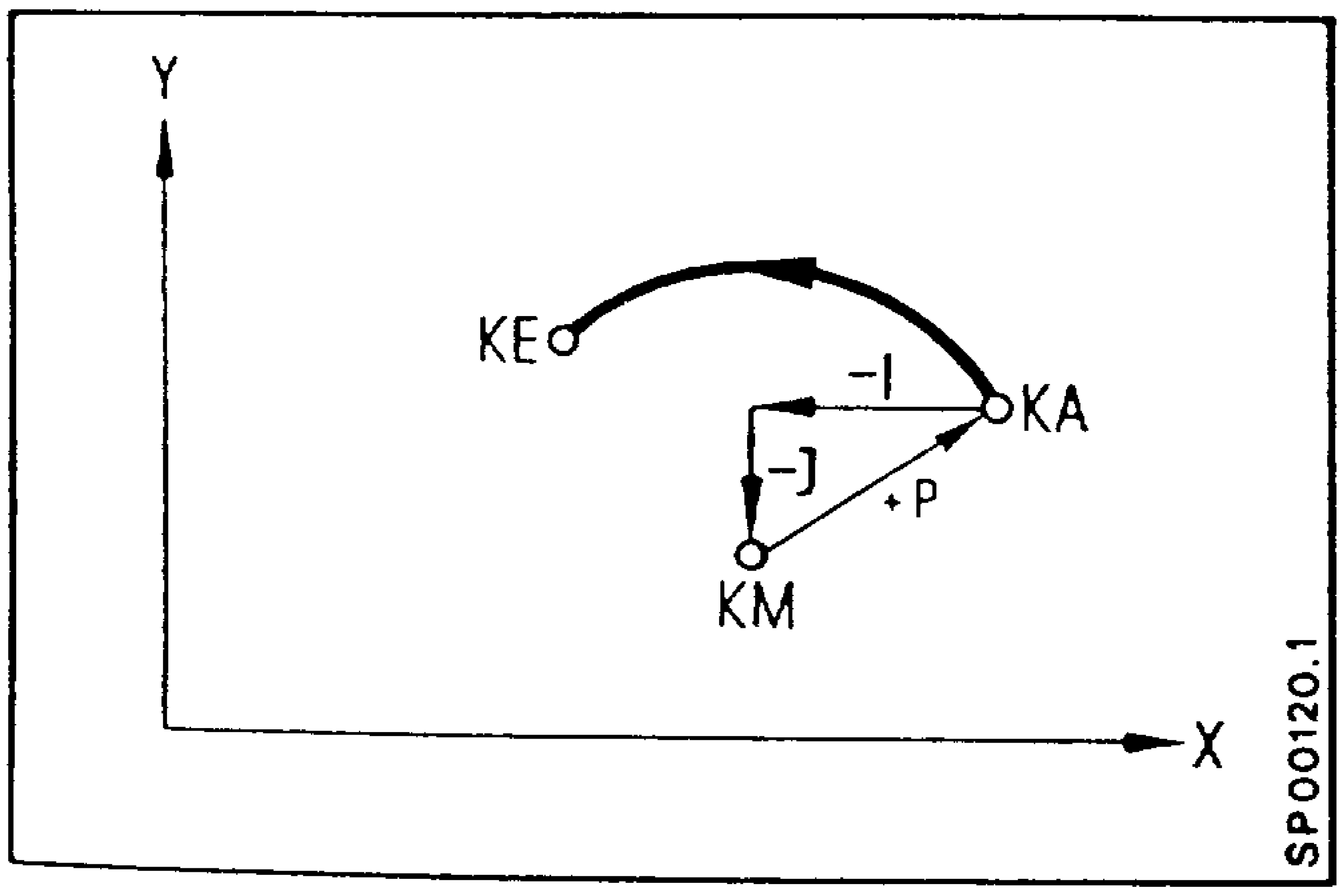
3.4 G02/G03 Circular interpolation

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

a) either through the I, J and K vectors, sign dependent, from a range of 0 to 360°. The sign results from the coordinate direction from the start point to the centre point.

b) or directly through the radius P 1)
+P Angle less than or equal to 180°
-P Angle greater than 180°

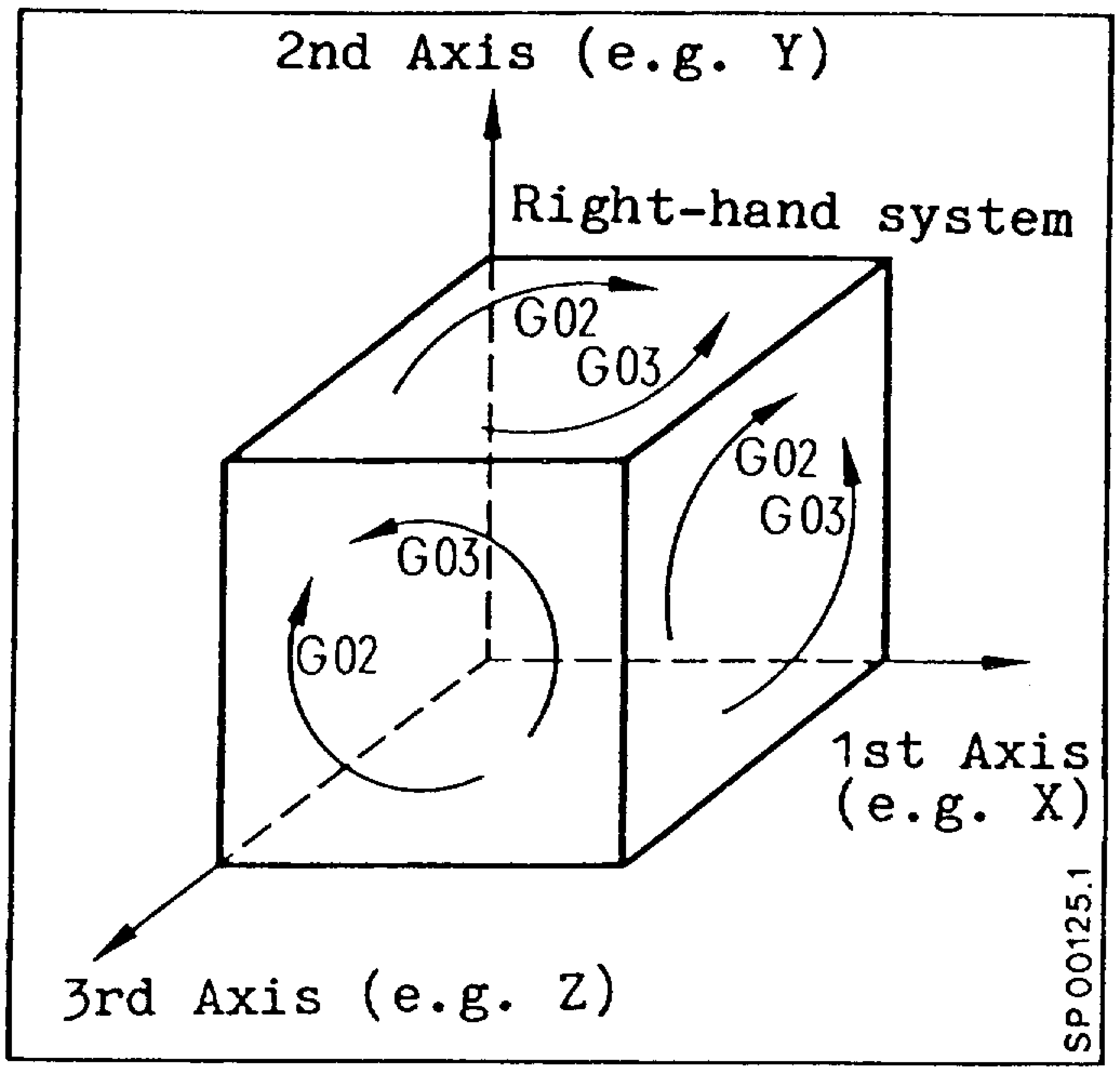
Radii should not be programmed when the angle to be traversed is 0° or 360°. In these cases the full circles must be programmed using the interpolation parameters I, J, and K.



Circular interpolation is possible in 2 out of 4 axes

1) with basic control 4

The direction in which the arc is traversed is determined by G01 or G03.



A right-hand system in the 3 primary axes is achieved with the following axis combinations:

```
X...  Y...   ->  G17
Z...  X...   ->  G18
Y...  Z...   ->  G19
```

3.4.1 Circular interpolation using interpolation parameters

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

The circle centre is determined through the interpolation parameters.

Increment (+ sign) from circle start point to circle centre point	I parallel to X axis J parallel to Y axis K parallel to Z axis

If only one axis coordinate is programmed, the value belonging to the primary axis of the selected plane (G17, G18, G19) is used as secondary axis coordinate. If the signal "4th axis = primary axis" is active, then this value is used according to the selected plane.

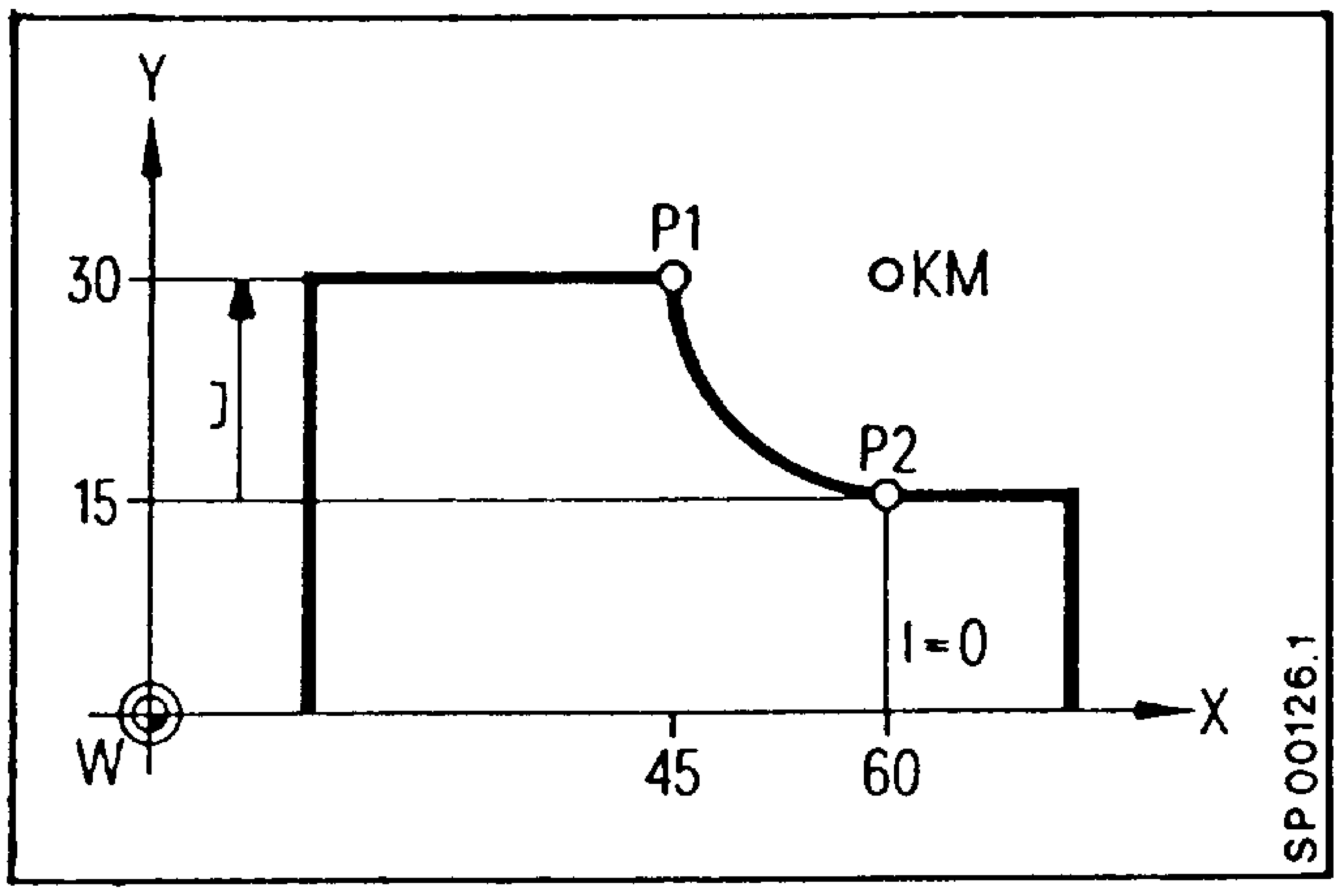
- The 4th axis may be defined using machine parameters as being parallel to the X, Y, or Z axis.
- The address of the circular interpolation parameter for the 4th axis is then equal to that of the associated parallel primary axis.
- If an interpolation parameter is not programmed, zero is automatically generated by the control.

Example:

N5 G17 G42 D03 ... LF	Plane and tool offset selection
N10 G03 X17 Y30. I-9. J8. LF . . .	Complete definition of the 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 generates the following:

N25 G17 G03 X17. Y30. I-9. J0. LF

3.4.2 Example for circular interpolation using interpolation parameters



Absolute dimension programming

N5 G02 G90 X45. Y30. IO. J15. LF
- The tool moves from point 2 to point 1.

N5 G03 X60. Y15. I15. JO. LF
- The tool moves from point 1 to point 2.

Incremental dimension programming

N10 G02 G91 X-15. Y15. IO. J15. LF
- The tool moves from point 2 to point 1.

N10 G03 X15. Y-15. I15.JO. LF
- The tool moves from point 1 to point 2.

3.4.3 Circular interpolation by specifying the radius 1)

The starting point of the circle or arc is determined by the previous block. The end point is given by both of the axis values (e.g. X and Y). The circle centre is defined by the signed radius.

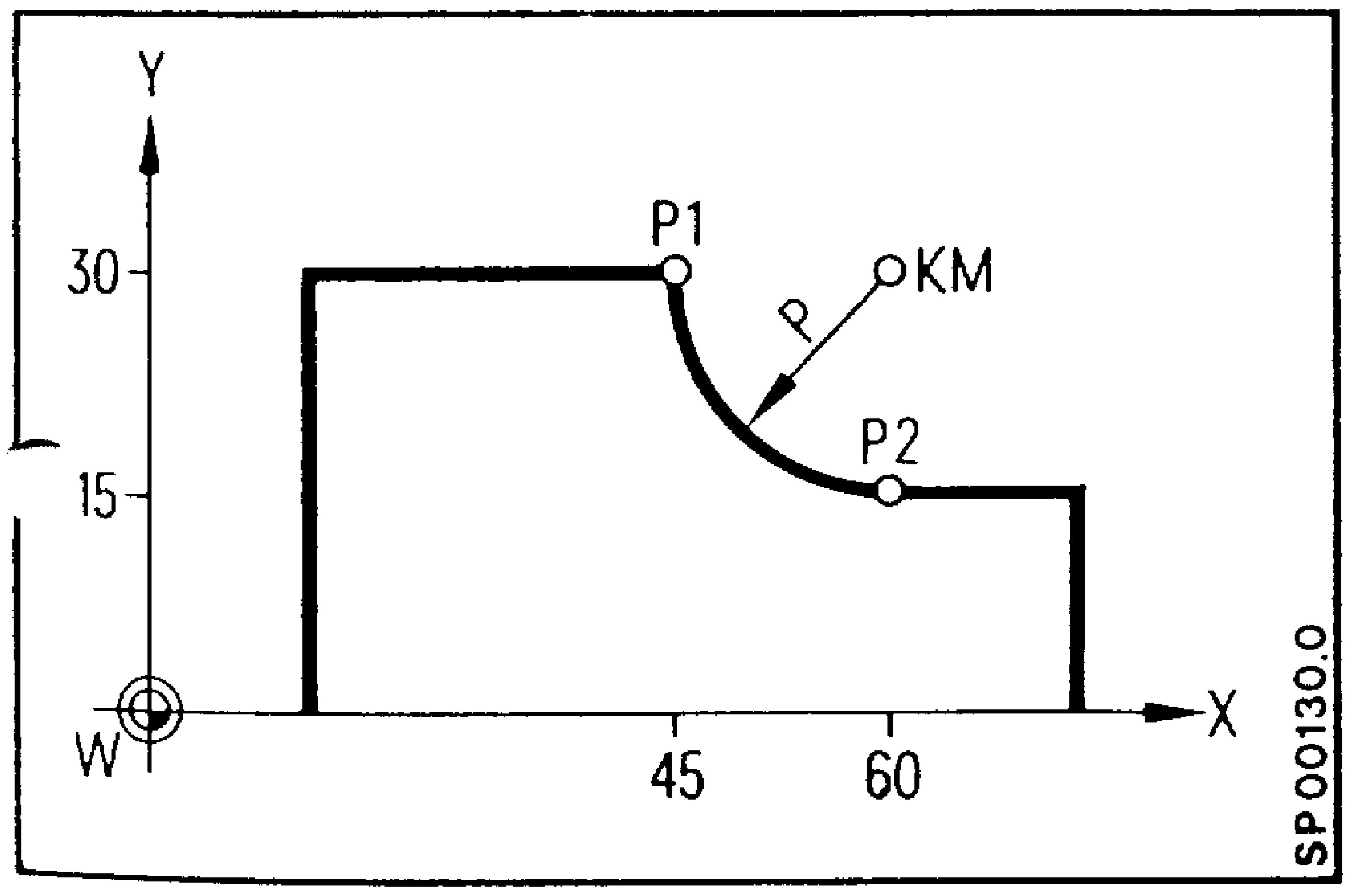
The sign of the radius value is given according to the size of the traversing angle.

less than or equal to	180°	P+
greater than	180°	P-

No radii may be programmed, when the distance between the circle end point and circle start point is less than 10 µm, i.e. a complete circle must be programmed using the interpolation parameters I, J or K.

3.4.4 Example of circular interpolation by specifying the radius

The circle centre point is determined by the signed radius.



N5	G03	G90	X60.	Y15.	P15.	LF	The tool moves from point 1 to point 2.
N10	G02	X45.	Y30.	P15.	LF		The tool moves from point 2 to point 1.

1) Only with basic control 4

3.4.5 Helical Interpolation 1)

(Requirement: 3D-interpolation) 1)

Helical interpolation is possible between 3 linear axes which are perpendicular to each other. A circle and a straight line are programmed in one block. The straight line is at right angles to the plane with the circular interpolation. The programmed feed rate is maintained along the circular path.
Example: Semicircle with 100 mm radius

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

- X200. Y0. — Plane of circular interpolation (using G17, G18 or G19). All axes have to be programmed
- Z200. — Axis for linear interpolation can be before or after the interpolation parameters
- I100. J0. — Interpolation parameters

The following combinations are possible if the 4th axis was declared a parallel axis to the X-axis:

Straight	Circle		Interpolation parameter	
X	Y	Z	J	K
Y	X	Z	I	K
Z	X	Y	I	J
4	Y	Z	J	K
Y	4	Z	I	K
Z	4	Y	I	J

The interpolation parameter of the corresponding parallel axis applies to the 4th axis.

1) with basic controls 3 and 4

3.5 G33 Thread Cutting

With boring and milling machines, threads can be cut by using a boring tool or a facing head.
G33 establishes a relationship between main spindle speed and feedrate speed. A pulse encoder generates 1024 pulses for each rotation of the spindle. These are converted by the control and the value defaulted to the feed arrives.

The feed is synchronized with the rotating spindle (i.e. the feedrate need not be programmed). However the feedrate programmed in address F is retained. In order to produce a thread in several passes, the axis will feed when the zero marker pulse initiates the thread cutting cycle. This ensures that the same angle between tool and part is maintained throughout the threading cycle. All threading cuts must be made with the same feed to avoid variations in the following error.
The spindle speed function and the rotation direction should be programmed prior to the thread cutting block proper to allow the spindle to reach the desired speed.
Right-handed and left-handed threads are programmed with M03 and M04 respectively.
The traverse direction depends on the programmed M03/M04.
The programmed thread length should take into consideration the machine axis acceleration and deceleration time; instantaneous acceleration and deceleration is not possible.

The thread length including acceleration and deceleration distance is programmed under the appropriate dimension address. In addition, the tool width must be taken into consideration.

The thread lead is specified under addresses I, J, K.

I, J, K parameters are incremental dimensions.
The dimension value is unsigned. The input resolution for the thread lead is 0.001 mm/rev.

The feed override, feed hold, spindle speed override, and single block switches are disabled when thread cutting.

See Section 8.2.4 for the assignment of thread lead to spindle speed.

Thread cutting with different speeds
(Basic control 4B, Option E35)

With this option, no further operator inputs are required. The speed must remain constant during a cut. This function ensures that the same thread part is cut at higher cutting speed as with the lower speed. An offset due to following error no longer occurs.

3.5.1 Constant lead tapered threads

For constant lead tapered threads, the thread lead is programmed for the leading axis.

The leading axis is defined as the axis traversing the longest distance.

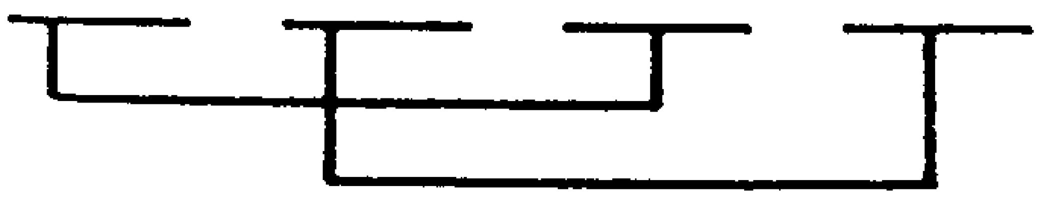
For equidistant traverse in all axes, the leading axis is defined in the order: X, Y, Z.

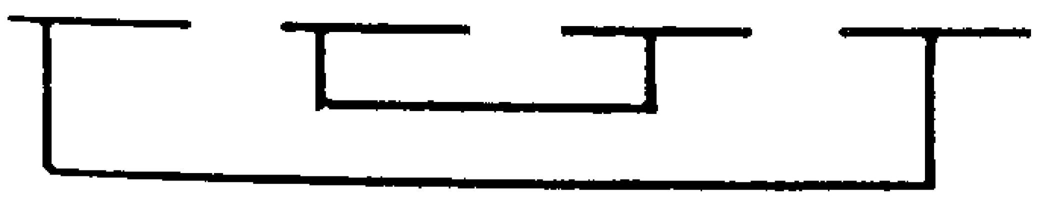
The address pairing for thread lead is XI, YJ, ZK.

The 4th axis can take the place of X, Y, or Z. If, for example, the 4th axis is parallel to the Z axis (machine parameter), the 4th axis thread lead is programmed under address K.

Example:

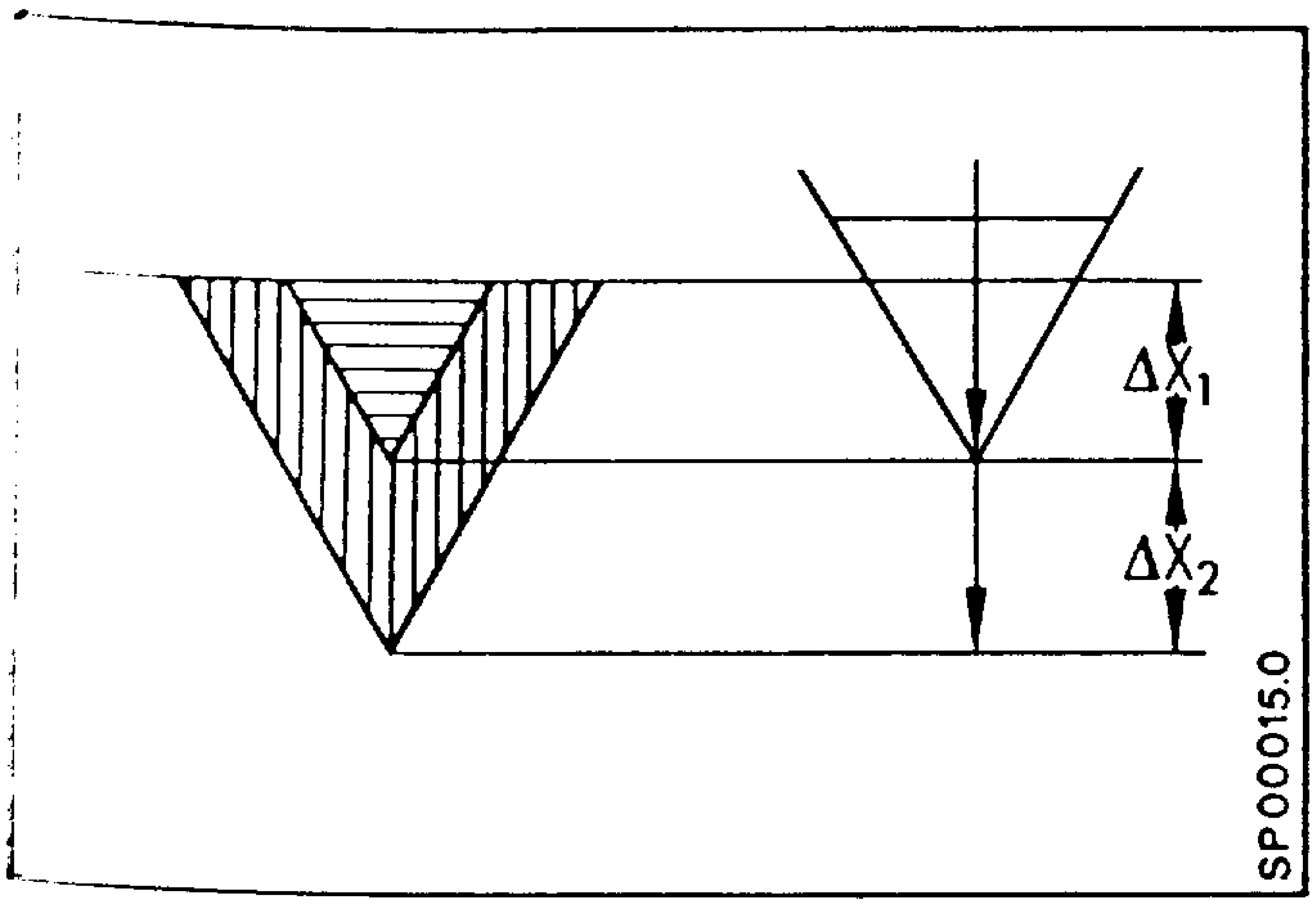
```
G33  X20.  Z10.  I0.2               Thread lead = 0.2 mm/rev

G33  X10.  Z20.  I0.2               Incorrect programming
G33  X10.  Z20.  J0.2               Z is the leading axis
                                    The thread lead must be
                                    programmed using K
G33  X10.  Z10.  I0.2          =  0.2 mm/rev.
G33  X10.  Z10.  I0.2  K0.2    =  0.2 mm/rev.  )
                                               )
                                               )  Permissible
                                               )
G33  Z10.  X10.  I0.2  K0.2    =  0.2 mm/rev.  )
```





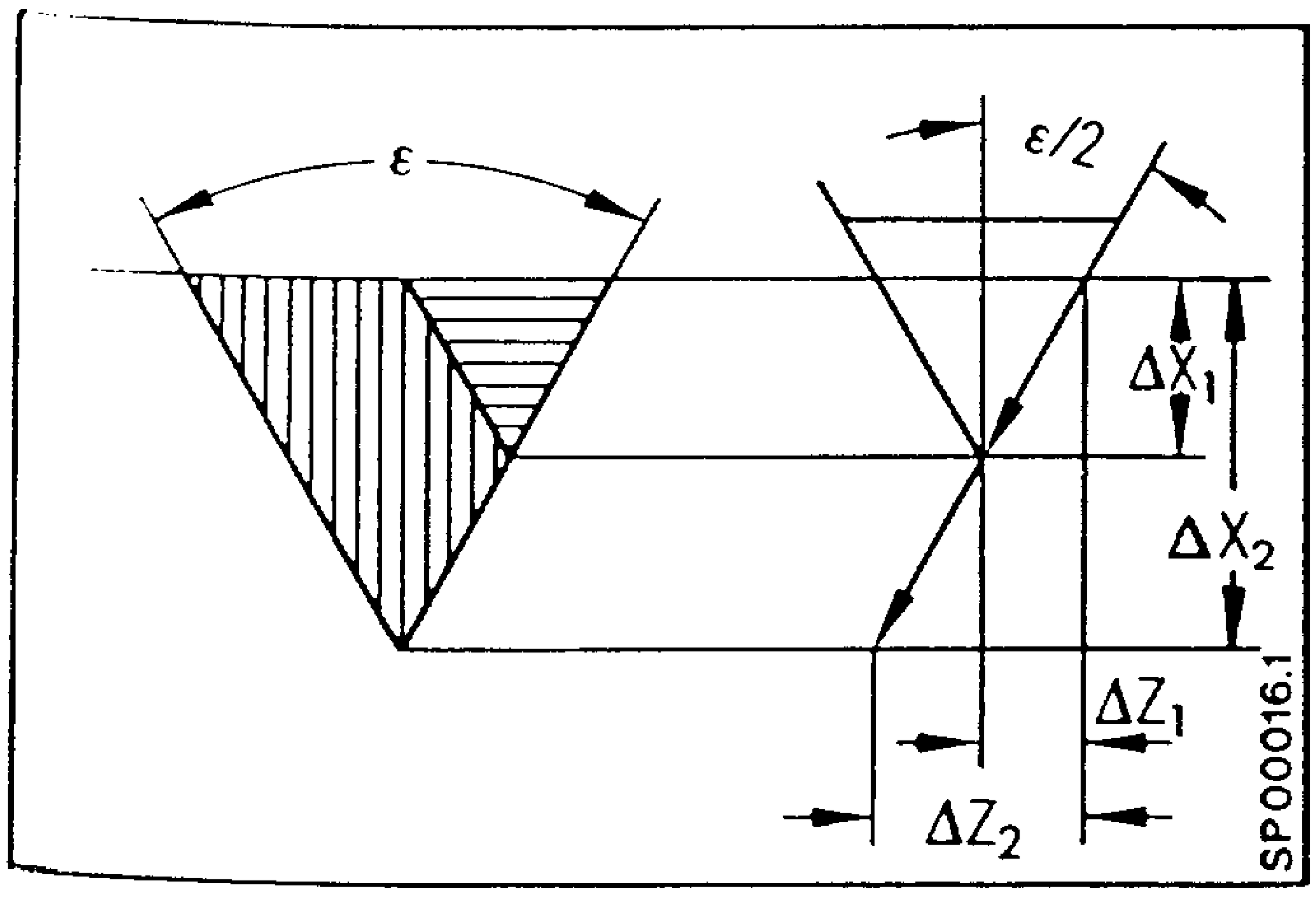
2.3.2 Feed Direction

Two methods can be used to thread cut. The tool can feed perpendicular to the cutting direction or parallel to the cutting direction.



"Perpendicular to the cutting direction"

When only one edge of the cutting tool is to cut, both axes must feed. The tool is fed in the direction of cut and perpendicular to the cutting direction before the start of the next threading pass.



"Cutter Edge Feed"

$$\Delta Z = \Delta X \cdot \tan \frac{\varepsilon}{2}$$

3.5.3 Variable lead thread

3.5.3.1 Variable lead thread without G34, G35

The thread lead can be modified by programming several contiguous thread cutting blocks. Within a block, the thread lead is constant. The region of constant thread lead can, if desired, be less than a single revolution. Subsequent thread cutting blocks will execute without waiting for the next zero marker pulse of the pulse encoder.

3.5.3.2 Thread cutting G34 (increasing lead) 1)

The thread lead increases per thread by the value programmed in F up to the maximum possible lead.

```
N...    G34    G90    Z 17.    K 2.    F 0.1    LF

Initial lead 2 mm  (K 2.)
Lead change +0.1 mm per thread,  (F 0.1)
e.g after 5 threads the thread lead is:
2 mm + (5·0.1 mm) = 2.5 mm
```

3.5.3.3 Thread cutting G35 (decreasing lead) 1)

The thread lead decreases per thread by the value programmed in F up to the minimum possible lead.

```
N...    G35    G90    Z 17.    K 200    F 0.5    LF

Initial lead 200 mm  (K 200)
Lead change -0.5 mm per thread,  (F 0.5)
e.g after 10 threads the thread lead is:
200 mm - (10 · 0.5 mm) = 195 mm
```

1) Basic control 4B, 4C

Examples for G34/G35 as for G33. Only the thread cutting blocks must be programmed differently if the thread lead is to be increasing or decreasing. The maximum lead modification is 16 mm. It must be programmed without sign. The value should be calculated as follows if the initial and final lead are known:

$$F = \left| \frac{\text{Initial lead}^2 - \text{Final lead}^2}{2 \cdot \text{Thread length}} \right|$$

3.5.4 Multiple thread

3.5.4.1 Multiple thread via axial starting point offset

A multiple thread is programmed in the same manner as a single thread. After the first thread is cut, the threading start point is displaced by an amount equal to the pitch circle before the thread cut sequence is repeated.

3.5.4.2 Multiple thread with spindle-specific starting angle offset (Basic control 4B, option E35)

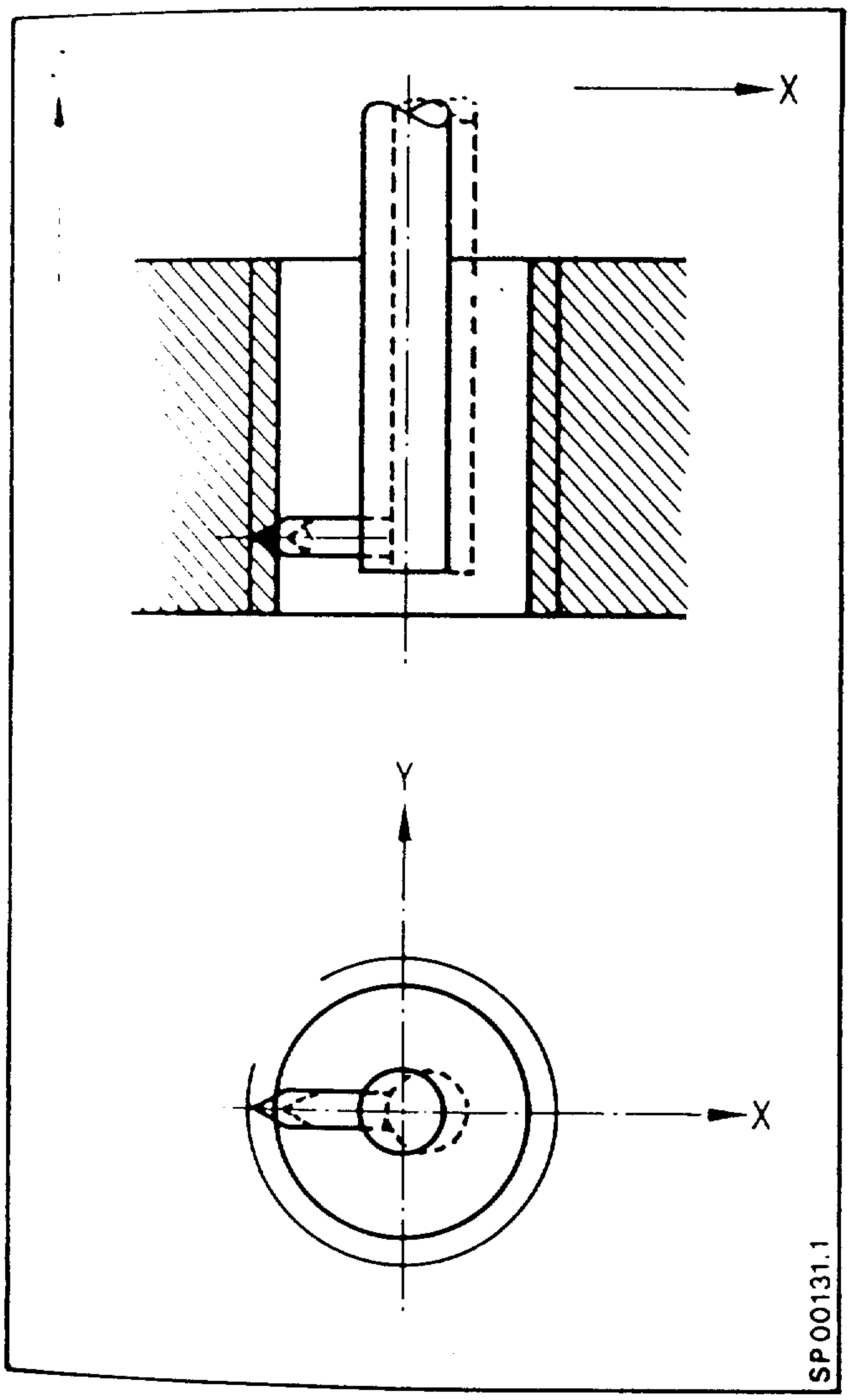
This function makes it possible to cut multiple threads. The spindle-specific starting angle for the respective thread can be programmed via G92 A... . "A" stands for the letter of the angle.

Example: triple thread (0°, 120°, 240°)
- 1st thread with G92 A0.
- 2nd thread with G92 A120.
- 3rd thread with G92 A240.

3.5.5 Thread Cutting with a Boring Bar

With the workpiece stationary, a thread can be cut by simultaneously rotating and feeding the boring tool. It is necessary to program the bar to retract to the start point:

Before the bar is retracted, the spindle must be stopped in an oriented position (see 4.3). Auxiliary function M19 is executed in the interface control.[1]. The bar is moved out of the cut and with the spindle stopped is retracted to the start position.



1) with basic controls 0, 1 and 2

Example:

Threading a blind hole with a boring bar

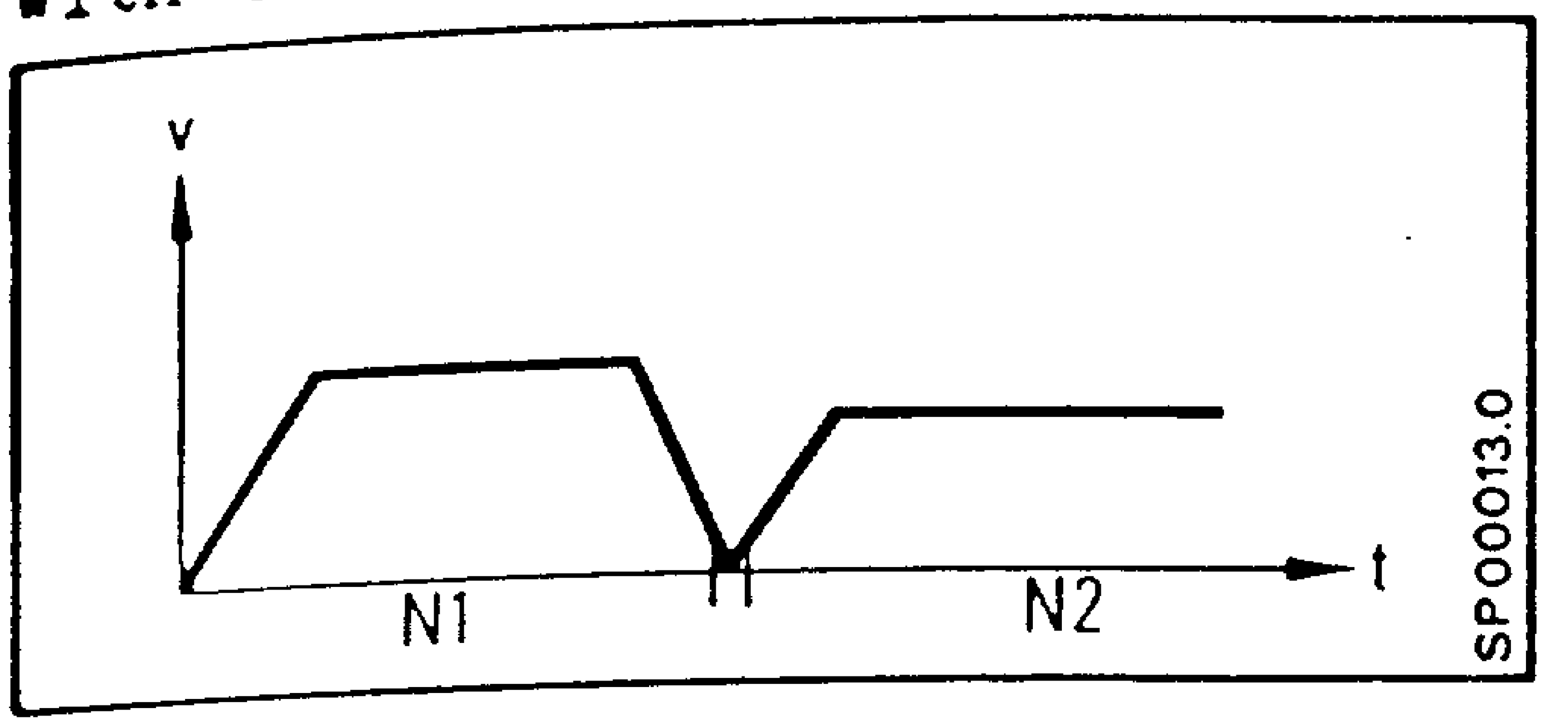
Absolute dimension programming

```
N20  G90  G00  X100.  Y...  S45  M03       LF
N25                     Z200.              LF
N30  G33                Z120.  K10.        LF
N35                                 M19    LF
N40  G00       X105.                       LF
N45                     Z200.       M00    LF
N50            X100.                M03    LF
N55  G04                F2                 LF
N60  G33                Z120.  K10.        LF
```

Block 20, 25:	The boring bar is centred over the drilled hole. The spindle is turned on. Direction M03.
Block 30:	The first threading cut is made. The thread end position (e.g. in absolute dimensions) is programmed under address Z. The thread lead is programmed under address K.
Block 35:	The spindle is brought to an oriented stop.
Block 40:	The boring bar is moved excentrically out of the cut in the X direction.
Block 45:	The boring bar is moved out of the hole in the Z direction. It is possible with a programmed stop (M00) to dress the boring bar for a second cut (e.g. manual feed).
Block 50:	The boring bar is centered over the drilled hole. At the same time the spindle is turned on.
Block 55:	If the positioning time in block 50 is shorter than the time it takes the spindle to accelerate to the correct speed, a dwell time of sufficient length must be programmed in block 55. This ensures that the spindle has reached the desired speed before beginning the next threading pass.
Block 60:	A second threading cut begins.

3.6 G60 Exact stop (Reset state 10th G group)

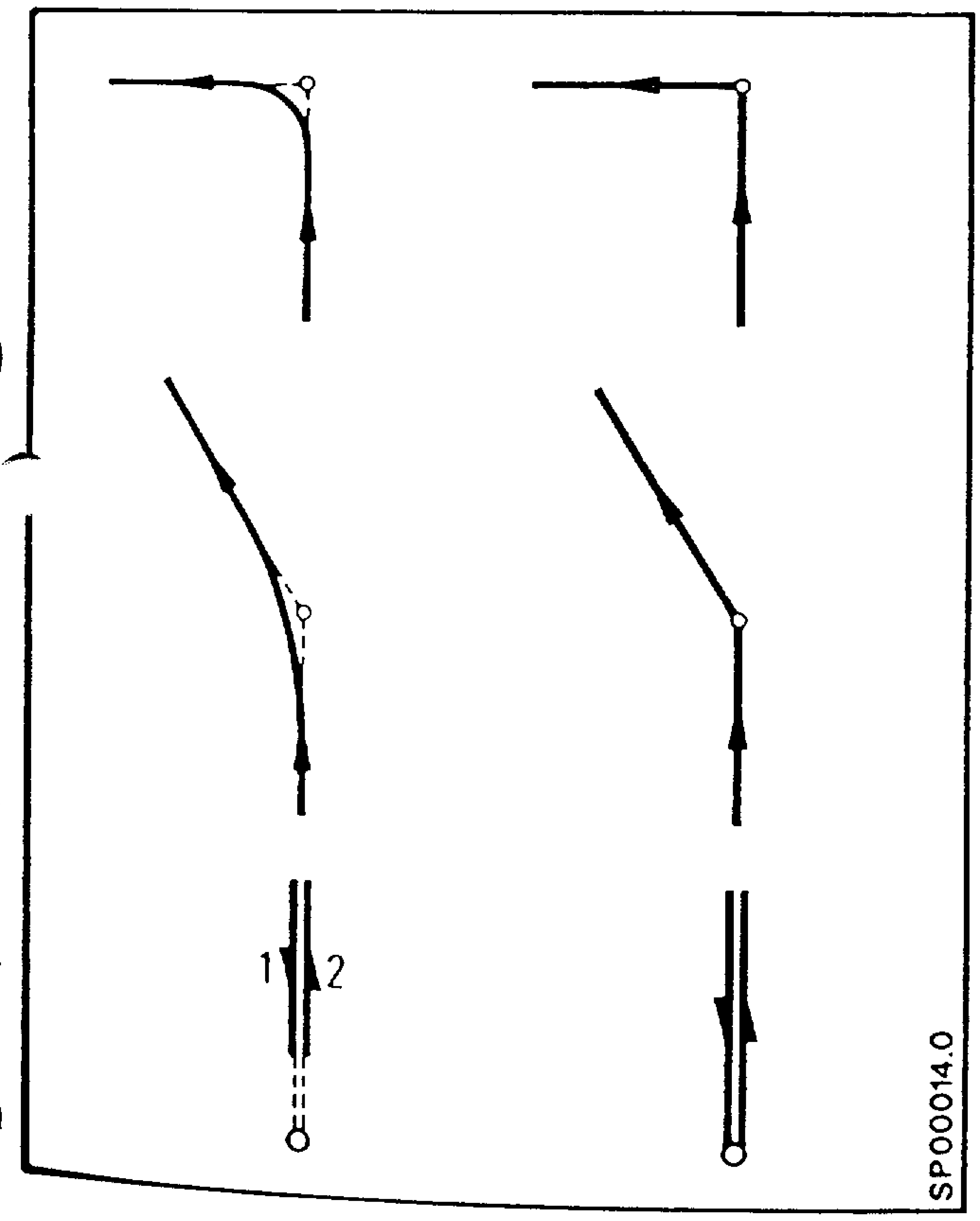
With the preparatory function G60 it is possible to position exactly to a target position (within the "exact stop tolerance"). The feed velocity is reduced to zero. The current following error is reduced to zero.



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 as G00 contains exact stop. G60 is modal and is cancelled with G64 (continuous path control) or G63 (tapping with compensation chuck).

The example shows direction reversal with and without G60:

without G60 with G60



3.7 G63 Tapping with a floating tap holder

The preparatory function G63 is programmed for 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. Sufficient length compensation must be provided on reaching the programmed position to allow for overshoot due to spindle speed run down.

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

G63 may only be used with linear interpolation G01. G60 will cancel G63.

3.8 G64 Continuous path control

The preparatory function G64 is programmed to assure smooth path transitions between contiguous blocks containing path movements, however, a tangential direction change will result in a rounded corner.

3.8.1 G62 Contour Machining (rounding of block transition by reduced speed)

This function is self-holding like G64. There is deceleration towards the end of the block to the reduced speed input under machine parameter 347. The minimum value of the reduced speed is 5 mm/min.

3.9 G04 Dwell Time

The dwell time is programmed under address F.

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

A block programmed with G04 may not contain other functions.

e.g.
N.. G04 F11.5 LF

Dwell time 11.5 sec always an **unsigned number**

When necessary, several contiguous 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 only valid for the block in which it is programmed.

3.10 G70/71 Input Systems

G70 Inch input system

G71 Metric input system

The reset state (basic state) is determined by a machine parameter during commissioning. The control operates internally only with this predetermined inch or metric system, regardless of whether the input is in inches (G70) or metric (G71).

If a value is input to the system, which does not correspond to the basic state, it must be preceded by the associated G-function *). The control converts the input value into the other system defined in the basic state, so that the converted value is displayed in the system defined in basic state when the block is executed.

A different input system to that defined in the basic state can be programmed for one or more blocks or for a complete program. The required G-function must be programmed in the first block and the reset state must be programmed after the last block (the reset state is automatically selected after M02, M30 program end).

The following are dependent on the reset state of the input system:

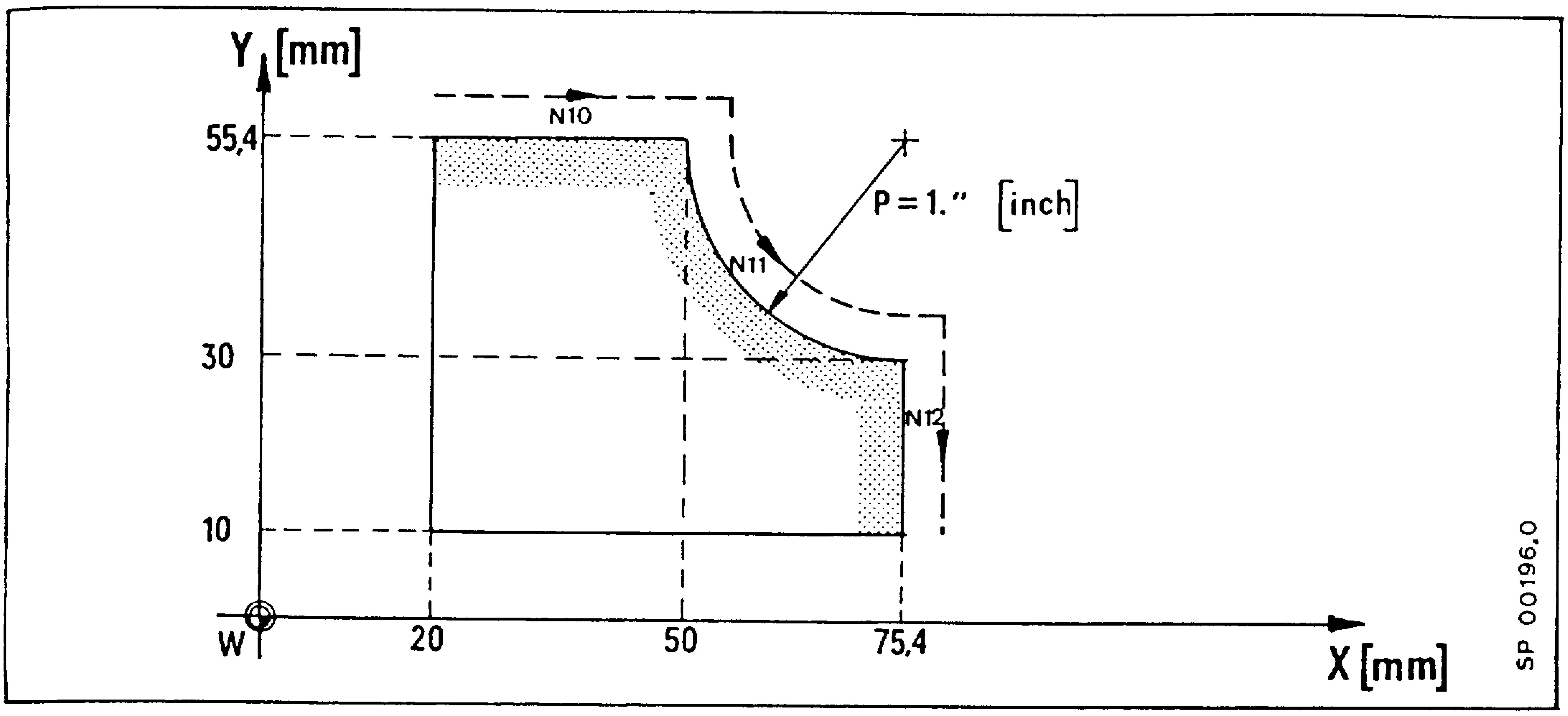
- Actual value display (also setpoint/actual value difference)
- Zero shifts
- Feedrate speed G94, G95
- Tool offset

The following are dependent on programmed G70 or G71 functions:

- Path information X, Y, Z
- Interpolation parameters I, J, K
- Chamfers/Radii P-/P
- Parameters, which are not included under path information, interpolation parameters, or chamfers/radii

*) G70/G71 must be the first function programmed in the block.

Example: G71 = reset state (metric)



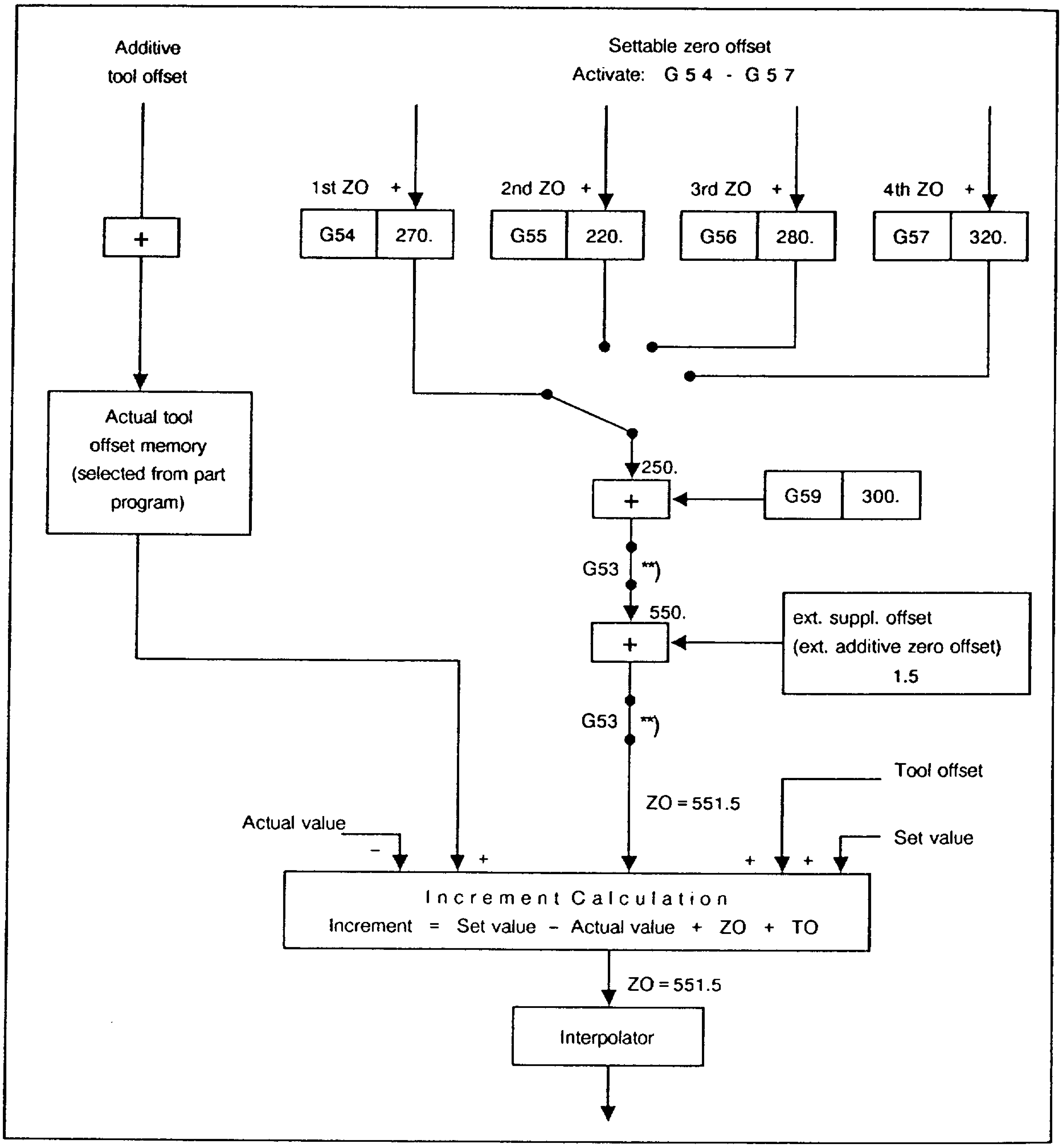

```
.
.
.
N10 G91 X30.                    LF
N11 G03 G70 X1. Y-1. I1. J0.    LF
N12 G01 G71 Y-20.               LF
.
.
.
```

3.11 Zero offsets (ZO)

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

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

Zero offset using one axis / and a selected group* as an example:



*) Basic control 4B, 4C

**) Can be selected via machine datum

3.11.1 G54/G55/G56/G57 Settable Zero Offset

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

Values for the zero offset for each axis can be entered into the control via the operator panel.
The offset is included when calculating the block end point in absolute and incremental data blocks, when the associated axis is programmed
With incremental data blocks (G91) any change in zero offset is taken into account.

Example:

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

Four or twelve* settable zero offsets per axis can be selected.

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

*) Basic control 4B, 4C

Selection of settable zero offset (from basic version 4B, 4C)

The settable zero offset (Z01-Z012) is selected by G54 to G59 and additional interface signals for groups 1 to 3.

	Group 1	Group 2	Group 3
G54	Z01	Z05	Z09
G55	Z02	Z06	Z010
G56	Z03	Z07	Z011
G57	Z04	Z08	Z012

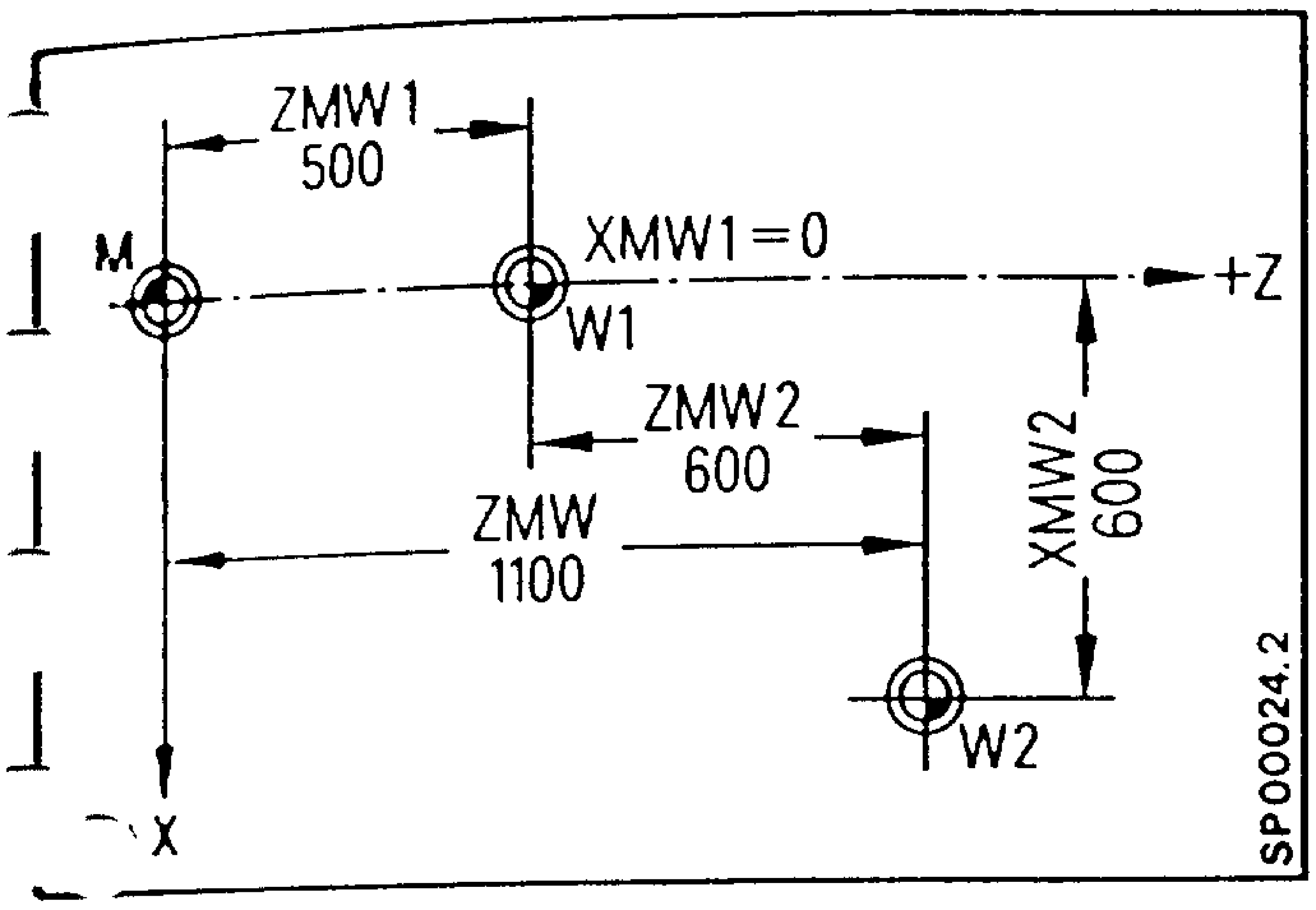
Release of the interface signals for instance via M-function. For reasons of clarity, L99 "Delete buffer store" must be programmed and the cutter radius compensation cancelled.

Example:

```
N10  G40  X.....  LF
N15  M..  ......  LF   Selection of zero offset groups
                       1 to 3 using an M function
N20  L99  ......  LF   Delete buffer store (see section 5.6.)
N25  G56  ......  LF   Selection of zero offset
```

3.11.2 G59 Programmable Zero Offset

An additional zero offset can be programmed with G59 under addresses X, Y, Z or 4th axis (if main axis). The programmed value is added to the settable zero offset and the external suppl. offset.



Settable Zero Offset:

Input Value:	XMW_1	=	0
	ZMW_1	=	500

Programmed Additive Zero Offset:

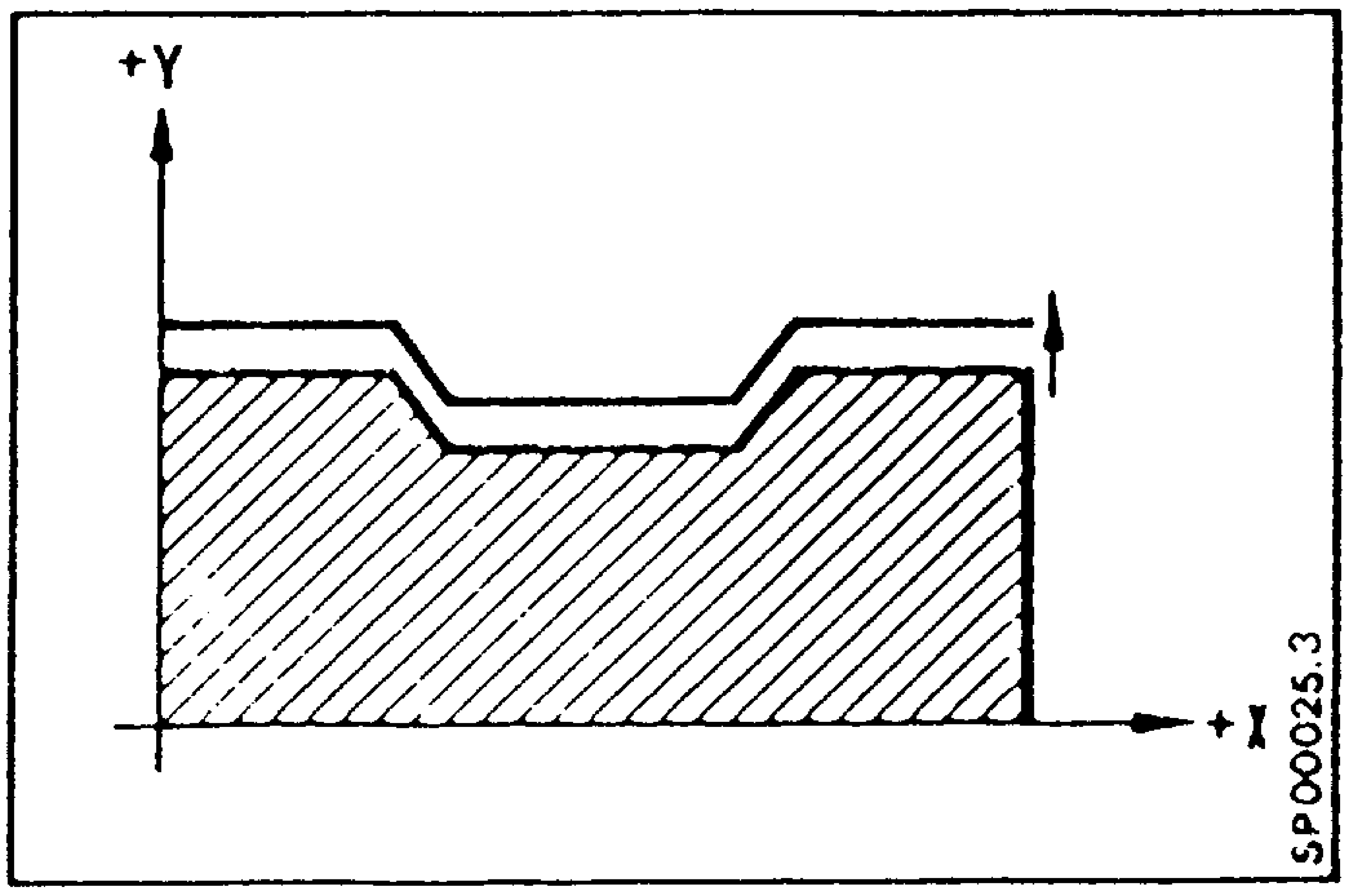
Input Value:	XMW_2	=	600
	ZMW_2	=	600

Resultant Zero Offset:	XMW	=	600
	ZMW	=	1100

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 YO. LF	(Value deleted)



Programmable additive zero offset in Y

With M02/M30 (program end) or on program abortion, 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 <u>blockwise</u> the coordinate displacement achieved by

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

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...  Y...  LF   - cancellation of tool offset
N1233  D00  Z...        LF   - cancellation of length compensation
N1234  G53  X...  Y...  LF   - cancellation of all ZOs
```

3.12 G94/G95 Feed F

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

- G94 feedrate F in mm/min
- G95 feedrate F in mm/rev.

For the relationship between the rotational feedrate and the spindle speed, see section 8.2.4.

With rotary axes, 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

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

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:

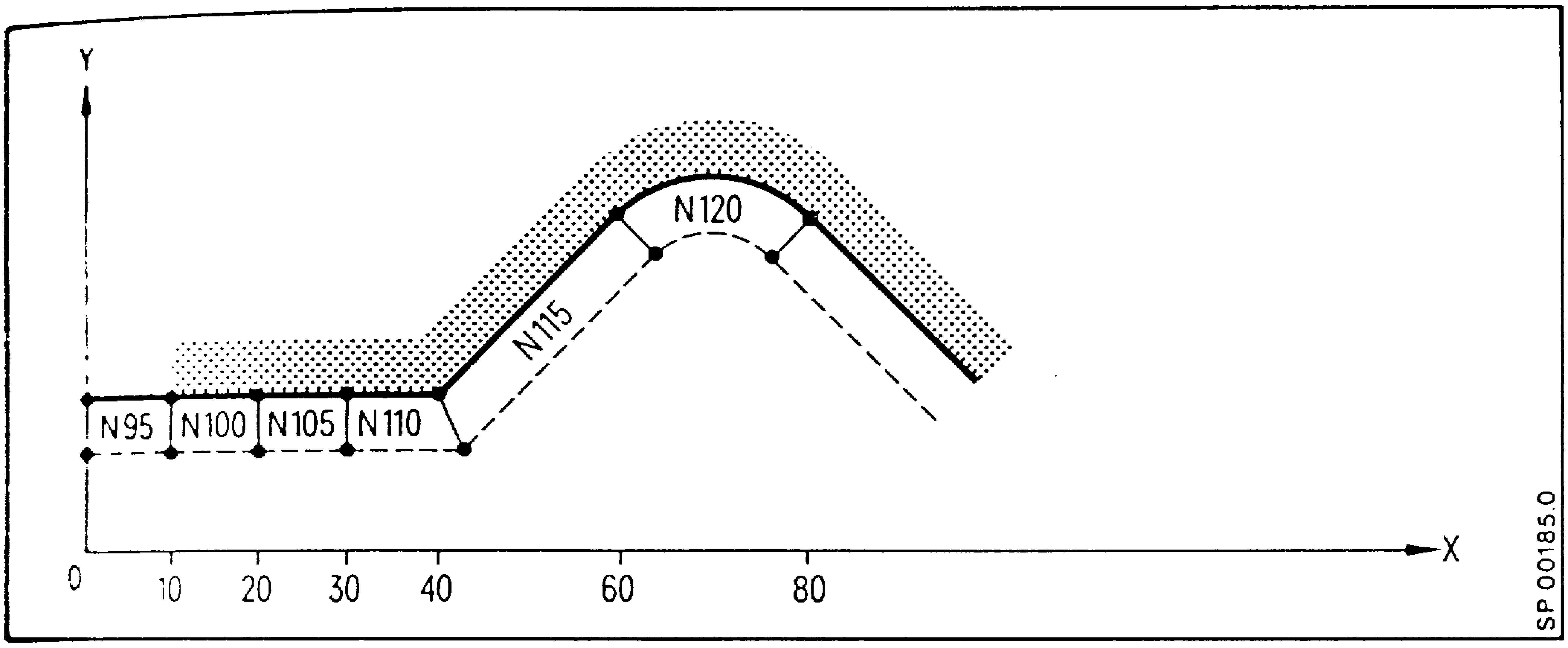
$$V_{tool} \text{ (mm/min)} = \frac{D}{Do} \times V_{programmed} \text{ (deg./min)}$$

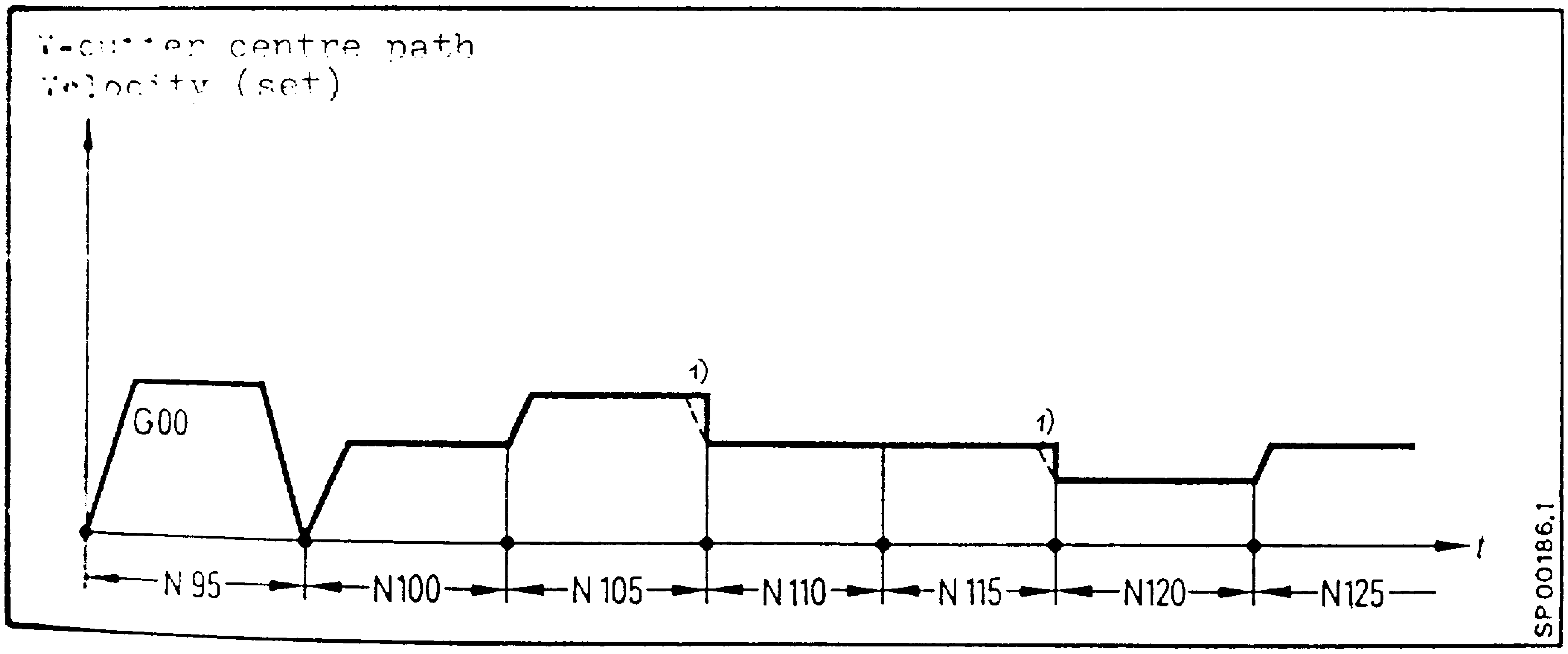
$$V_{programmed} \text{ (deg./min)} = \frac{Do}{D} \times V_{tool} \text{ (mm/min)}$$

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

Velocity Transitions

```
.
.
N95    G91   G42   G00   X10.                              LF
N100               G01   X10.      F200                    LF
N105                     X10.      F300                    LF
N110                     X10.      F200                    LF
N115                     X20.      Y20.                    LF
N120               G02   X ...     Y ...   I ...   J ...   LF
N125               G01   X ...     Y ...                   LF
```



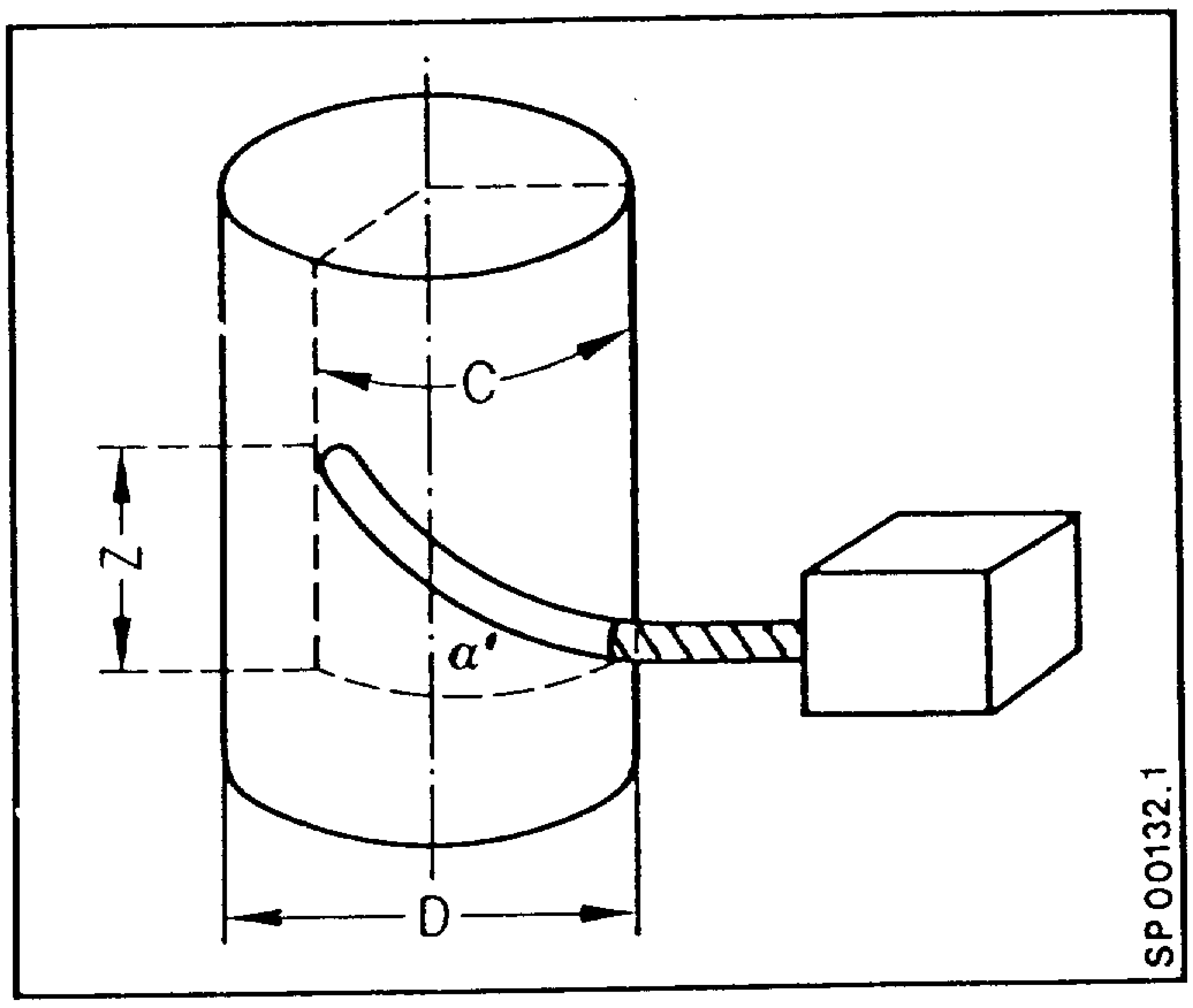


In block N120 the feedrate changes with respect to the cutter centre path in relation to the two radii (cutter radius, contour radius).

1) Speed transition can be selected as a ramp or jump function (machine data)

The following holds true when <u>simultaneously</u> moving a linear and rotary axis:

Whenever the distance between the tool tip contact point and rotary axis remains constant, the speed of the tool at the workpiece 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.



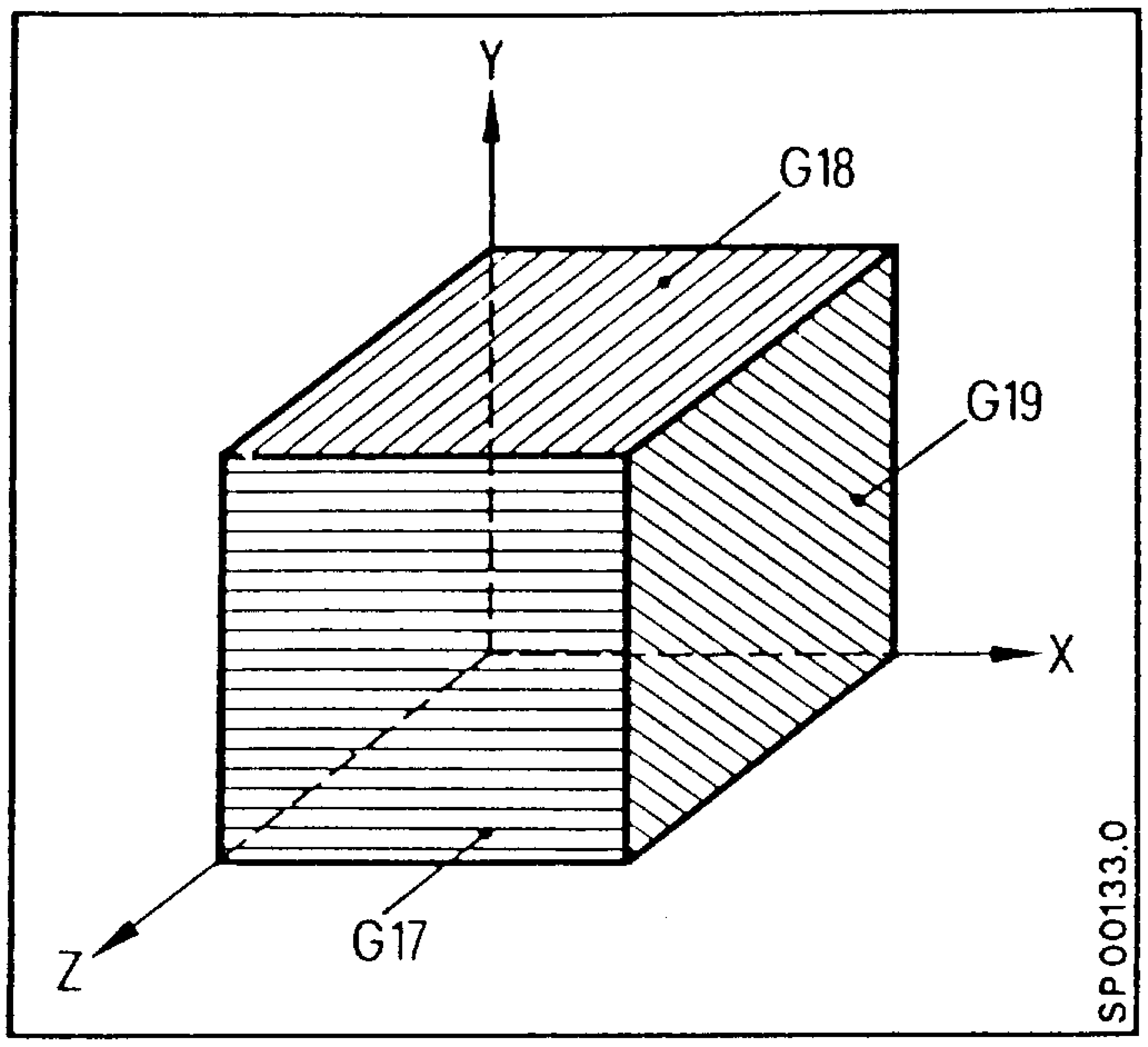
$$V_{tool} = V_{programmed} \sqrt{1 + \frac{D^2 - Do^2}{Do^2} \cos^2 \alpha}$$

$V_{programmed}$	= programmed path velocity in deg./min
D	= helix diamter in mm
Do	= unit circle diameter = 114.592 mm
α	= arc tan $\frac{Z}{C}$ (slope angle of the helix with D = I
Z	= programmed departure (mm)
C	= programmed angle in degrees
α'	= arc tan $\frac{Z}{C \times \frac{D}{Do}}$ (slope angle of the helix with D ≠ Do).

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.

3.13 G17, G18, G19 Machining plane selection



By programming G17 - G19, the plane is defined in which cutter radius compensation is available. If a plane has not been selected at the start of the program, the default plane will be defined by G17 (default setting of the 4th G group).

With a 4th axis the cutter radius compensation plane is defined as follows:

4th axis / G code	parallel to X	parallel to Y	parallel to Z	machine without a 4th axis
G17	X-Y *4-Y	X-Y *X-4	X-Y	X-Y
G18	Z-X *Z-4	Z-X	Z-X *4-X	Z-X
G19	Y-Z	Y-Z *4-Z	Y-Z *Y-4	Y-Z

(4th axis - see Section 2.1)

*) 4th axis equals main axis

3.14 Tool Offset

Two geometric values are stored under a tool offset number:

1st length ± 999.999 mm
2nd length ± 999.999 mm (radius)

A total of 64 offset pairs are available.

A tool offset is called via the 2 digit designator D01 ... D64. 1)

- Tool offset input via tape

```
%TO                     LF  (TOOL OFFSET)
G92  D01  D...  P...    LF
.
G92  D64  D...  P...    LF
M02  or   M30           LF
```

The tool length is stored under D..., the radius under P... .

Selecting and cancelling the length compensation

The selection is only possible, when G00 or G01 are active. The plane perpendicular to the plane in which the compensation should act must be selected.

N5 G00 G17 D... Z...

Only the length compensation is used from the store D... . The compensation value contained in the D word is always used in the calculation with its sign according to the programmed axis.

The length compensation is cancelled via D00. The compensation is only executed when the respective axis is programmed.

1) Basic controls 0, 1, 2, 3
max. 32 tool offset pairs (D01...D32)

1. Length compensation without CRC

N5	G90	G00	G17	D01	Z...	Selection of the length compensation (e.g. boring tool)
.						
.						
N50				D00	Z...	Cancellation of the length compensation via D00

2. Length compensation with CRC (3.15)

N5	G90	G00	G17	G41	D02	X...	Y...	Automatic selection of the CRC with length compensation
N10						Z...		
.								
.								
N50				G40		X...		Cancellation of the CRC only. The length compensation is not cancelled.
N51					D00	Z...		Cancellation of the length compensation.

3.15 G40/G41/G42 Intersectional Cutter Radius Compensation (Option)

G40 Cutter compensation off
G41 Tool left of the part - in traverse direction -
G42 Tool right of the part - in traverse direction -

The cutter radius compensation is active in the selected plane (G17 to G19). The length compensation of the cutter is always active perpendicular to the selected plane, in this case to the third axis.

When mirror imaging is used and the sign is considered, the traversed path is as follows:

Both axes are mirrored or Neither axis is mirrored		one axis is mirrored	
Sign for the radius compensation value of the cutter			
G41 + left	- right	+ right	- left
G41 right	left	left	right

Selecting and cancelling the intersectional cutter radius compensation.

The selection is only possible when G00 or G01 is programmed. G40, G41, and G42 may be programmed in a block without axis motion. However, at least one axis motion must be programmed in order for them to become active. In the selection block only the axes of the selected plane may be programmed.

Example for the selection

N10 G01 G17 G41 D07 X... Y...	LF	- When the end position of this block is reached, the path in the selected plane is correctly compensated. Only the radius is compensated.
N15 Z...	LF	- The length compensation is activated.

or

N10 G17	LF	- Plane selection
N15 G41 D07	LF	- Compensation selection
N20 G01 X... Y...	LF	- When the end position of this block is reached, the path in the selected plane is correctly compensated. Only the radius is compensated.
N25 Z...	LF	- The length compensation is activated.

With G40 the compensation (G41 or G42) is cancelled. At least one axis motion must be programmed in the selcted plane in order to restore the tool to its uncompensated path. The length compensation is cancelled by D00. The uncompensated path is obtained when the axis parallel to the length compensation is programmed.

Example for the cancellation:

N30 G40		X...	LF	-	Cancellation of the radius compensation
N35	D00	Z...	LF	-	The length compensation is cancelled.

Or:

N30 G41 D00		X...	LF	-	Cancellation of the radius compensation via D00.
N35		Z...	LF	-	The length compensation is cancelled.

Change from G41 to G42

N10	G01	G17	G41	D12	X...	Y...	LF	-	Radius compensation
N15						Z...	LF	-	Length compensation
N20			G42		X...	Y...	LF	-	Change in the radius compensation (e.g. change in direction relative to the workpiece; straight milling)
N25						Z...	LF	-	No change in the length compensation.

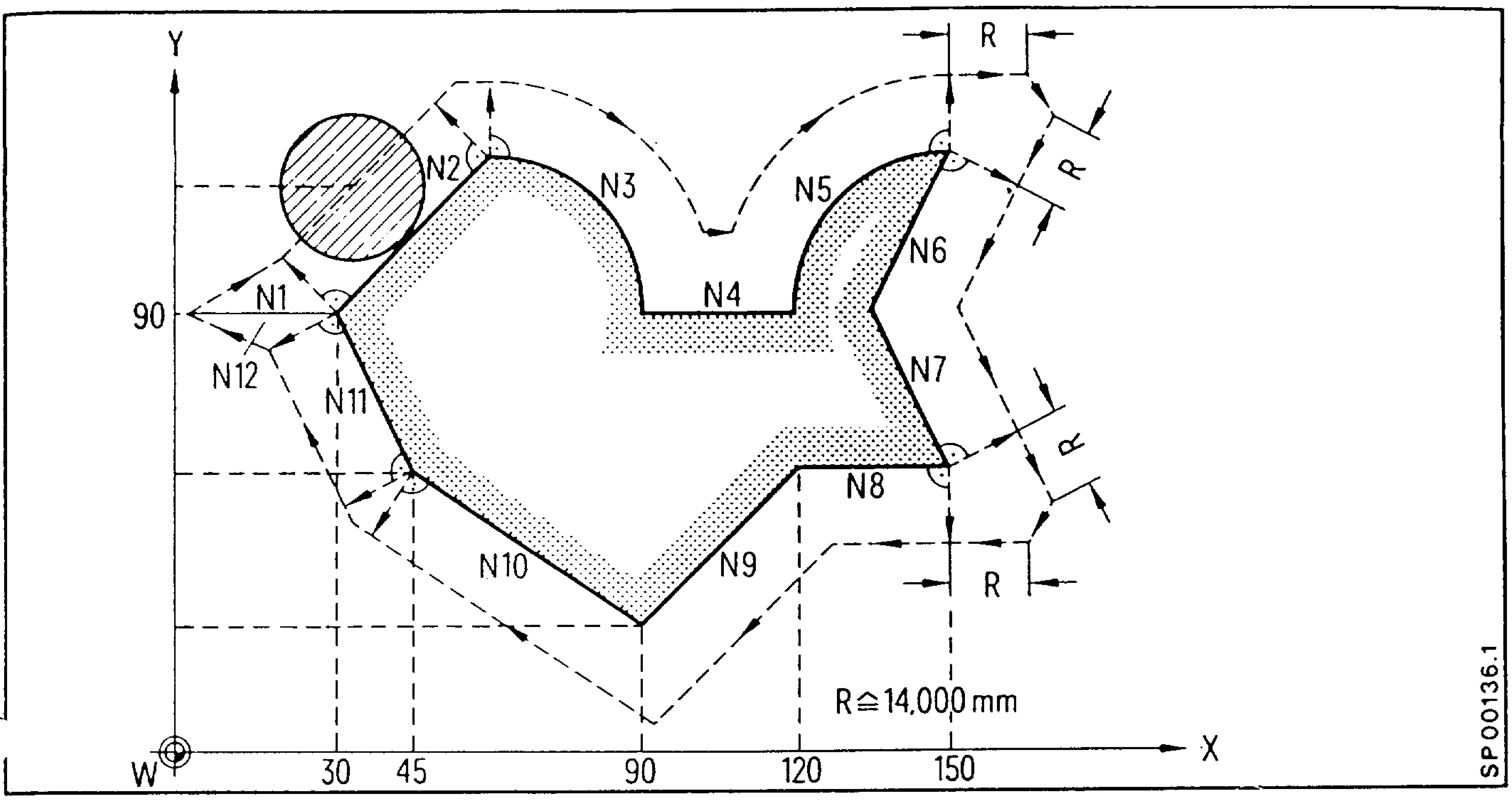
Change in the tool offset function

The G function does not have to be reprogrammed.

```
N10 G01 G17 G41 D12 X... Y... LF
N15                 Z...      LF
N20             D10 Z...      LF - Change in the length compensation
N25                 X... Y... LF - Change in the length compensation
```

With radius compensation selected, G92, G59, and G33 must not be programmed. Remedy: Program the functions (G92, G59, G33) before selecting radius compensation, or switch off the radius compensation, select G92, G59 and G33, and then again select radius compensation. With radius compensation selected (including G40 block) the effective zero offset value may not be changed.

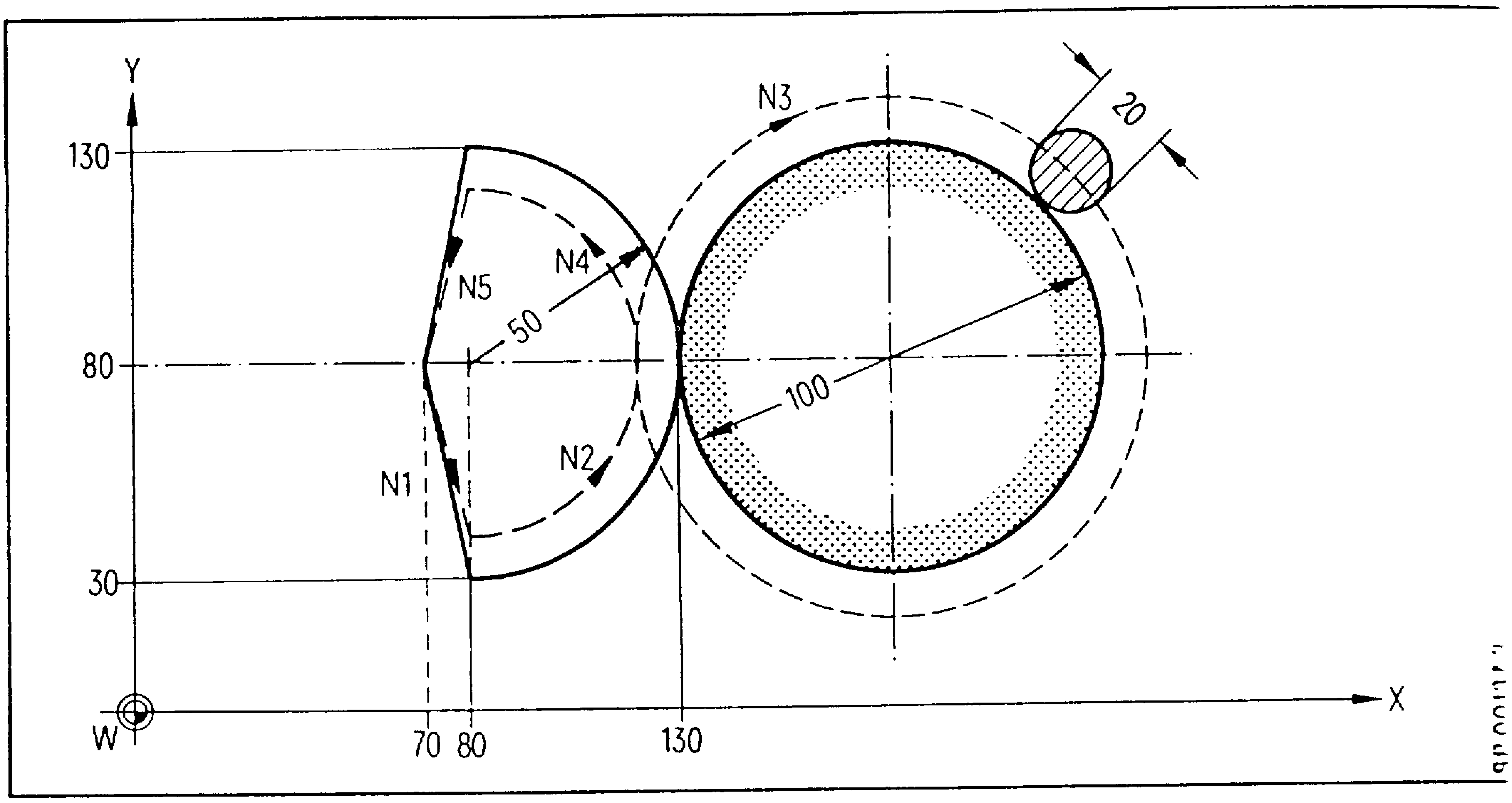
Example: Milling with CRC




```
N1   G01  G41  D01  G90  G17  X30.  Y90.  F5000  S56  M03  LF
N2   G91                      X30.  Y30.                   LF
N3   G02                      X30. Y-30.  I0.   J-30.      LF
N4   G01                      X30.                         LF
N5   G02                      X30.  Y30.  I30.   J0.       LF
N6   G01                      X-15.Y-30.                   LF
N7                            X+15.Y-30.                   LF
N8                            X-30.                        LF
N9                            X-30.Y-30.                   LF
N10                           X-45.  Y30.                  LF
N11                           X-15.  Y30.                  LF
N12  G40  G90                 X0.   Y90.                   LF
N13 ...
```

The cutter used has a radius of 14.000 mm. The radius dimension is stored under address D01.

Example: Full circle programming with CRC



- - - - - - - - - - - Cutter Centre Path

——————— Part Contour

```
N1   G90   G00   G17   G41   D01   X80.   Y30.                 LF
N2   G03                           X130.  Y80.   I0.   J50.    LF
N3   G91   G02                     X0.    Y0.    I50.  J0.     LF
N4   G90   G03                     X80.   Y130.  I-50.J0.      LF
N5   G00   G40                     X70.   Y80.                 LF
```

3.16 G43/G44 Tool Length Offset (positive, negative) 1)

G43 Positive tool length offset (reset state)
G44 Negative tool length offset

When selecting the tool length offset G43 is active, if G44 has not been previously programmed. The tool length offset is active in the axis perpendicular to the CRC-plane.

The tool length offset is cancelled with DOO. The un-compensated path is traversed when the respective axis is programmed (s. Sections 3.14 and 3.15).

1) with basic control 4

Sign Convention

A plus sign is input by the operator, when the actual dimension of the tool is greater than the programmed value. A minus sign is input, when the actual dimension of the tool is smaller than the programmed value.

For example:

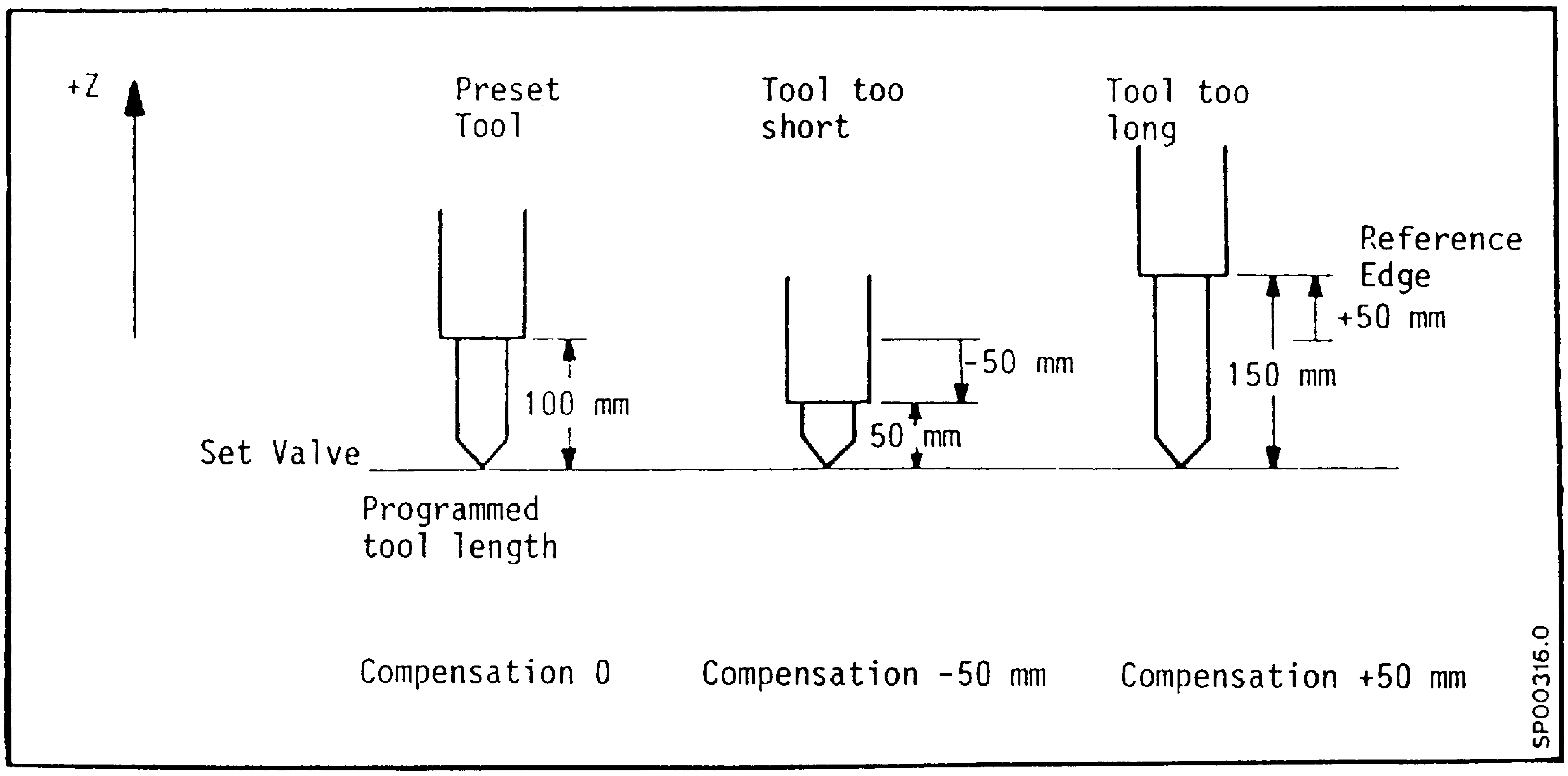
The actual drill length is longer than the programmed drill length	+ offset
The actual drill length is smaller than the programmed radius	- offset

G43 positive tool length offset

The offset called by the D word is calculated with its sign to the associated axis.

G44 negative tool length offset

The offset called by the D word is calculated with its opposite sign to the associated axis.



3.17 Milling of cylindrical contours G92 P 1)

(Prerequisite: Option B61)

With cylindrical interpolation it is possible to machine cylindrical contours by coordinating the motion of a rotary axis with a linear axis while the rotary axis diameter is held constant. Straight contour paths as well as arc contour paths using intersectional cutter compensation can be programmed.
The rotary axis angle is dimensioned in degrees:
The circumferential dimension is calculated by the control.
The diameter is defined as

$$P = \frac{\text{part contour diameter}}{\text{unit circle diameter}}$$

and programmed with G92 P. This ratio can range between 0.00001 and 999.99999.

Input system	Unit circle diameter
metric	114.592 mm
inch (10^{-4})	11.4592 inch

The unit circle diameter is derived from the relation $\pi \times d = 60$.

$$\text{Unit circle diameter} = \frac{360}{\pi} \text{ in mm or inch}$$

Apart from the names of the axes, no other signs may be written in a block with G92 P.. .

1) with basic control 4

N.. G92 P... C LF

- P... : Factor for normalized circle $= \frac{\text{part contour diameter}}{\text{unit circle diameter}}$
- C : Rotary axis (e.g. C) associated with the normalized circle

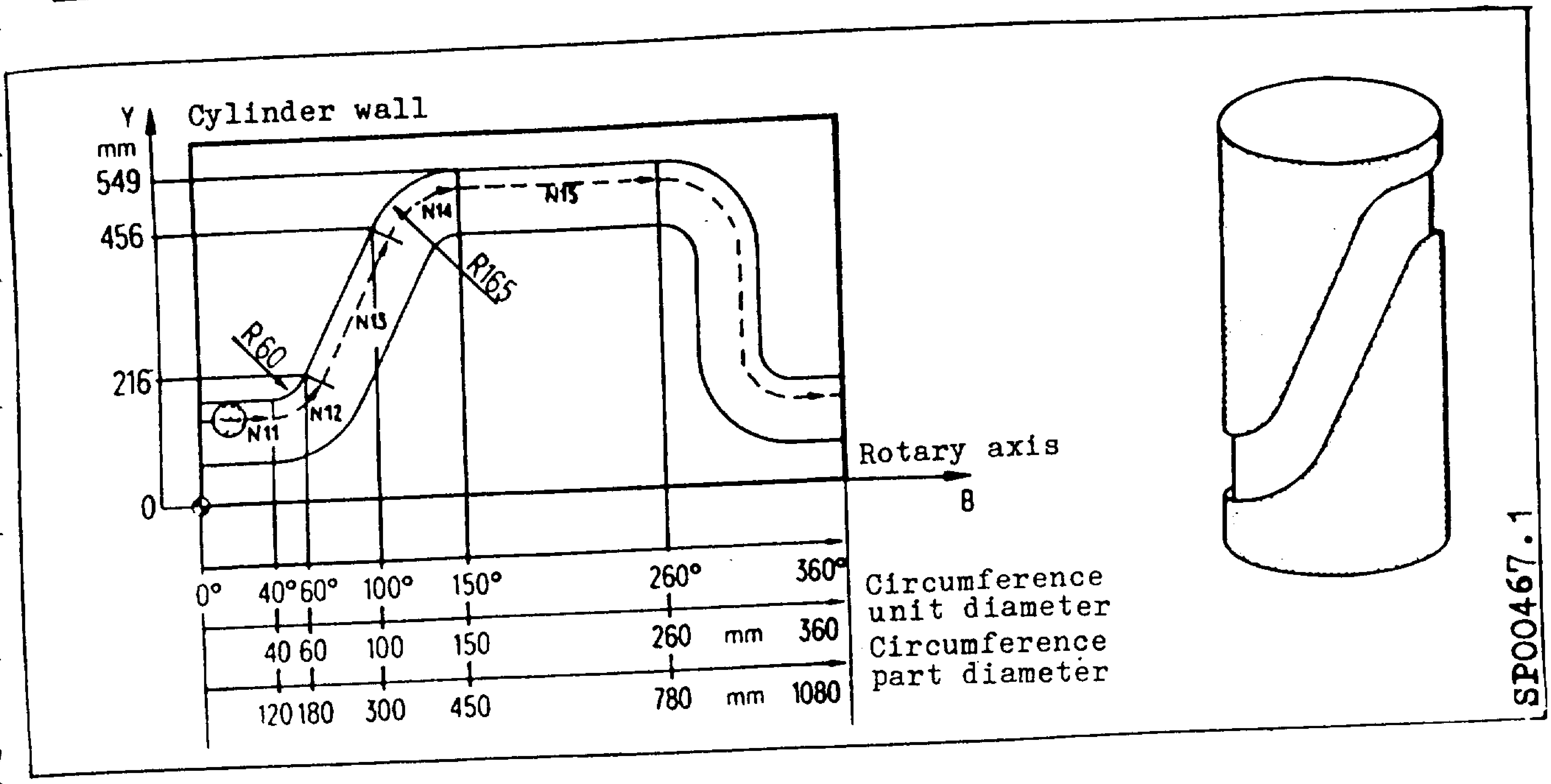
- The factor for the normalized circle is modal but can be redefined in subsequent blocks. The value is reset with M02/M30.

- The programmed feedrate is maintained on the centre path.

- As long as the factor is ≠ 1, this axis (e.g. C)can only interpolate with one other axis; i.e. interpolation with more than 2 axes is possible only after the factor is set to 1.

- Cutter radius compensation cannot be used with cylindrical interpolation!

Example:



.

.

.

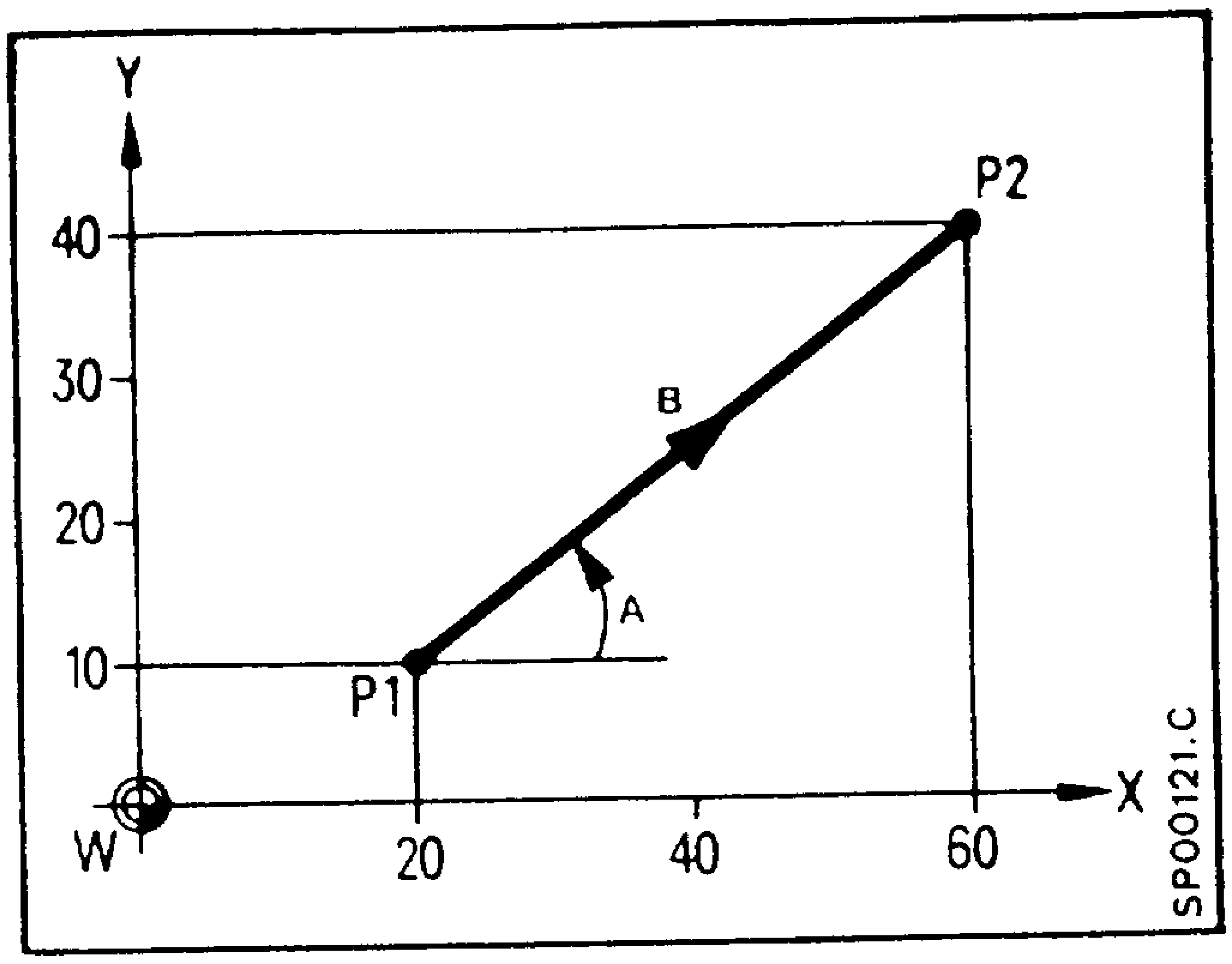
```
N10  G92  P3  B                          LF - Select cylindrical inter-
                                              polation
N11  G01        B40.                     LF
N12  G03        B60. Y216.   P+60.       LF
N13  G01        B100.Y456.               LF
N14  G02        B150.Y549.  P+165.       LF
N15  G01        B260.                    LF
.
.
.
N30  G92  P1.B                           LF - Delete cylindrical inter-
                                              polation
.
.
.
```

- 4th axis (B) parallel to X-axis
- 4th axis (B) = main axis

3.18 G10/G11 polar coordinate programming 1)

G10 Linear interpolation rapid traverse

G11 Linear interpolation feedrate (F)



W = Workpiece zero

P1= Centre point in the polar coordinate system

A = Angle

P = Radius

Example traverse from P1 to P2.

N... G.. X20. Y10. P50. A40.

- With rapid traverse G10 or with feedrate G11 → G..
- Centre point in the polar coordinate system → X20. Y10.
- Radius → P50.
- Angle → A40.

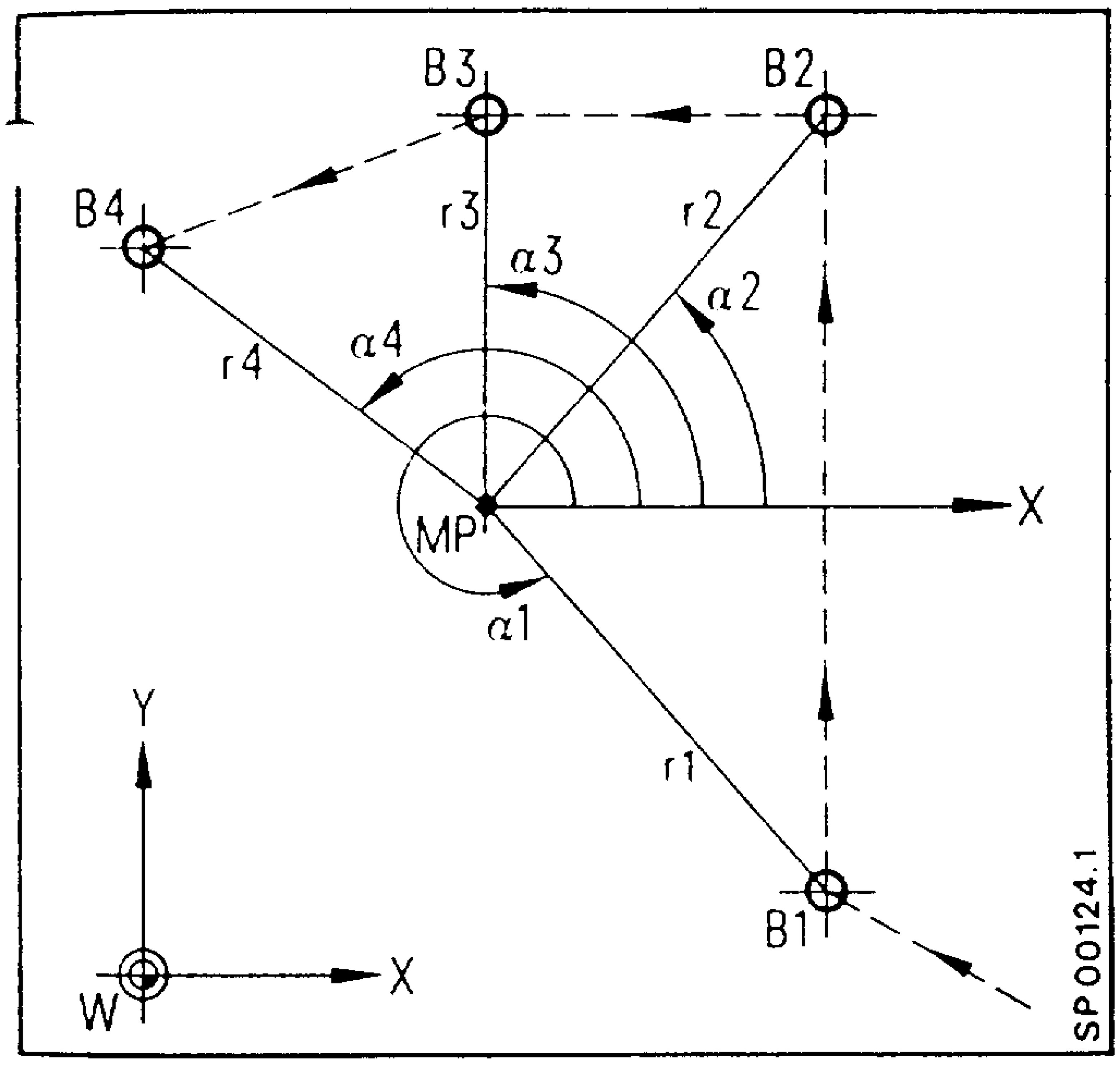
- The angle always refers to the positive axis programmed first (in this case X-axis). The positive direction of the first programmed axis is equal to an angle of 0 degrees.
  The positive direction of the second axis programmed is equal to an angle of 90 degrees.
- The angle value is always absolute and positive. Input resolution 10^{-5} degrees.

1) with basic control 4

- When first programming the polar coordinates, both centre point coordinates must be programmed in absolute dimensions. It is recommended to program both centre point coordinates.
- The centre point is modal and can be reprogrammed. On "End of Program" (M02/M30), the centre point dimension is cleared.
- The incremental programming of the centre point (with G91) is always referenced to the previously programmed centre point.

Application

E.g. for multiple hole drilling where all hole dimensions are with respect to a common centre point.



The angles are with reference to the X-axis since the centre point coordinate in X is programmed first.

Programming:

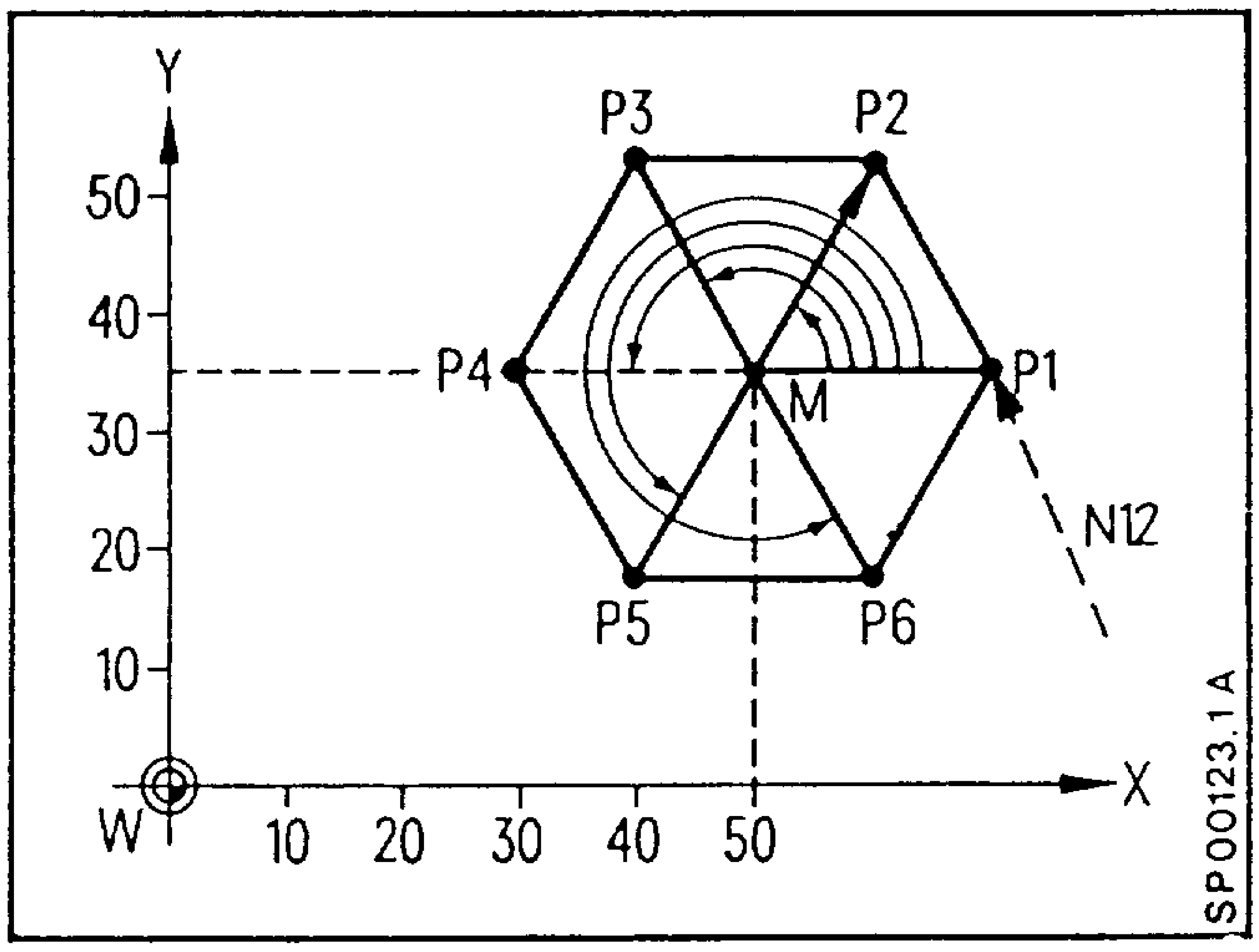
```
N10 G90 G10 G81 X... Y... A... P(r1) R01... R02... R11... LF
N15     G10                A... P(r2) LF
N20     G10                A... P(r3) LF
N25     G10                A... P(r4) LF
N30         G80                       LF
```

A... P(r.) = Hole position in polar coordinates
X... Y... = Centre point of the polar coordinate system
G81 = Drilling cycle selection
G80 = Drilling cycle cancellation

Program execution:
- approach drilling location B1
- perform the desired cycle
- approach drilling location B2
- perform the desired cycle
- etc.
- G10 must be programmed in each block as G81 terminates with rapid traverse

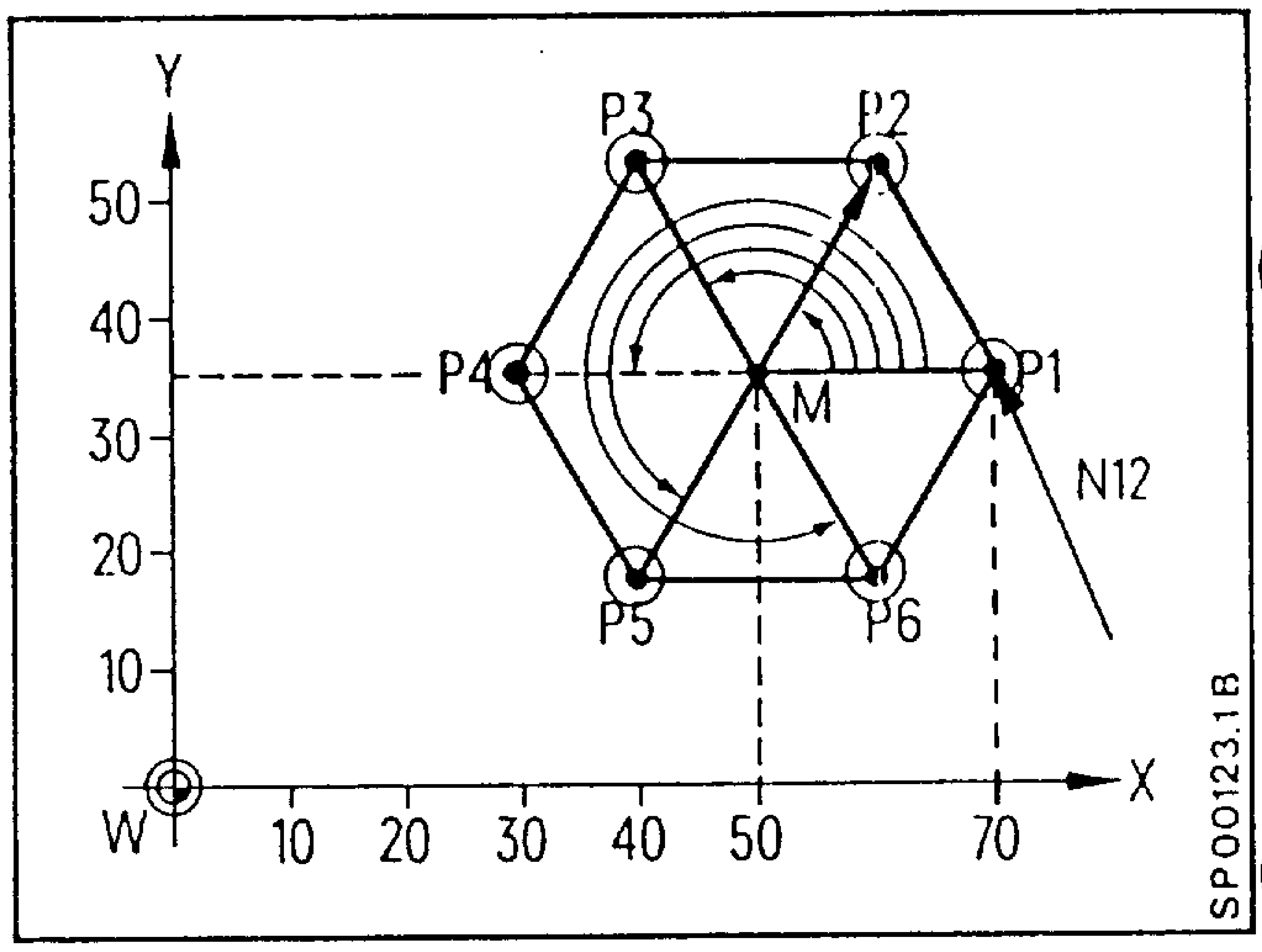
Application

E.g. Milling of a hexagon head




```
N11 F1000                             LF
N12 G90 G11 X50.Y35.P20.A   0.LF (P1)
N13                         A  60.LF (P2)
N14                         A120.LF (P3)
N15                         A180.LF (P4)
N16                         A240.LF (P5)
N17                         A300.LF (P6)
N18                         A   0.LF (P1)
```

e.g. Approaching drilling positions




```
N11 G81 F1000                         LF
N12 G90 G10 X50.Y35.P20.A     0.LF
N13 G10                         A  60.LF
N14 G10                         A120.LF
N15 G10                         A180.LF
N16 G10                         A240.LF
N17 G10                         A300.LF
N18 G80                               LF
```

P... A... = Position in the polar coordinate system

X... Y... = Centre point in the polar coordinate system

G81 = Drilling cycle selection

G80 = Drilling cycle cancellation

3.19 G36/G37, "TRANSMIT" coordinate transformation (Option B65)

Milling of rotary parts on rotating tables whereby the desired contour is achieved by interpolation of the rotary axis with a linear axis (application with special machines). The TRANSMIT function permits contour programming in a "fictitious" Cartesian coordinate system while the machine moves in a real polar coordinate system.

The fictitious Cartesian coordinate system is formed by the 1st X axis and the 4th axis (rotary axis).

The rotary axis is then labelled with the address "C" and the fictitious axis with the symbolic address "Cf". The transformation is selected and cancelled in the program using G-functions.

G36 coordinate transformation cancelled (reset position)

Programming is as normal in the real Cartesian system (machine coordinate system).
Rotary axis C in degrees, speed in degrees/min.

G37 TRANSMIT coordinate transformation selected

Programming is in the fictitious cartesian coordinate system.

For further information see also SINUMERIK 3T/3TT programming guide.

- With G17/18/19 the missing axis of the current fictitious plane is provided when only one axis in the block is programmed and this axis is an axis of the selected plane (example a and b).

- If only one axis of the fictitious plane and one axis which does not belong to the fictitious plane is programmed, the controller outputs Alarm 504 (example c).

- Both axes of the fictitious plane must be programmed if other axes in addition to the axes of the plane are programmed in this block (example d).

Example: G37 active, X - Cf = fictitious plane = G 17 - plane

a)	G17 X.. LF	= Cf is provided
b)	G17 Cf.. LF	= X is provided
c)	G17 X.. Z.. LF	= Alarm 504
d)	G17 X.. Cf.. Z.. LF..	= no axis provided no Alarm
e)	G17 Z.. LF	= - " -

- The cutter centre must be on the X axis in the centre of rotation.

- The controller cannot compensate for an offset in the direction of Cf.

- The workpiece zero is in the rotary axis centre.

- If the cutter radius path correction (G41/42) is selected the transformation must neither be switched on or off (changeover from G36/G37).

- Within a sequence of contour definition blocks the transformation must not be switched on or off (changeover from G36/G37).

- Rapid traverse movements are to be programmed with the appropriate F value under G01 or G11.

- When changing from G36 to G37 the actual value of the C axis is set to the value 0 and the actual value of the X axis to the machine actual value regardless of any offsets to be carried out (ZO). Zero offsets are calculated according to the Cartesian system.

The external zero offsets for the C-axis must be zero for G37 in the nch input system, as the position of the decimal point is different.

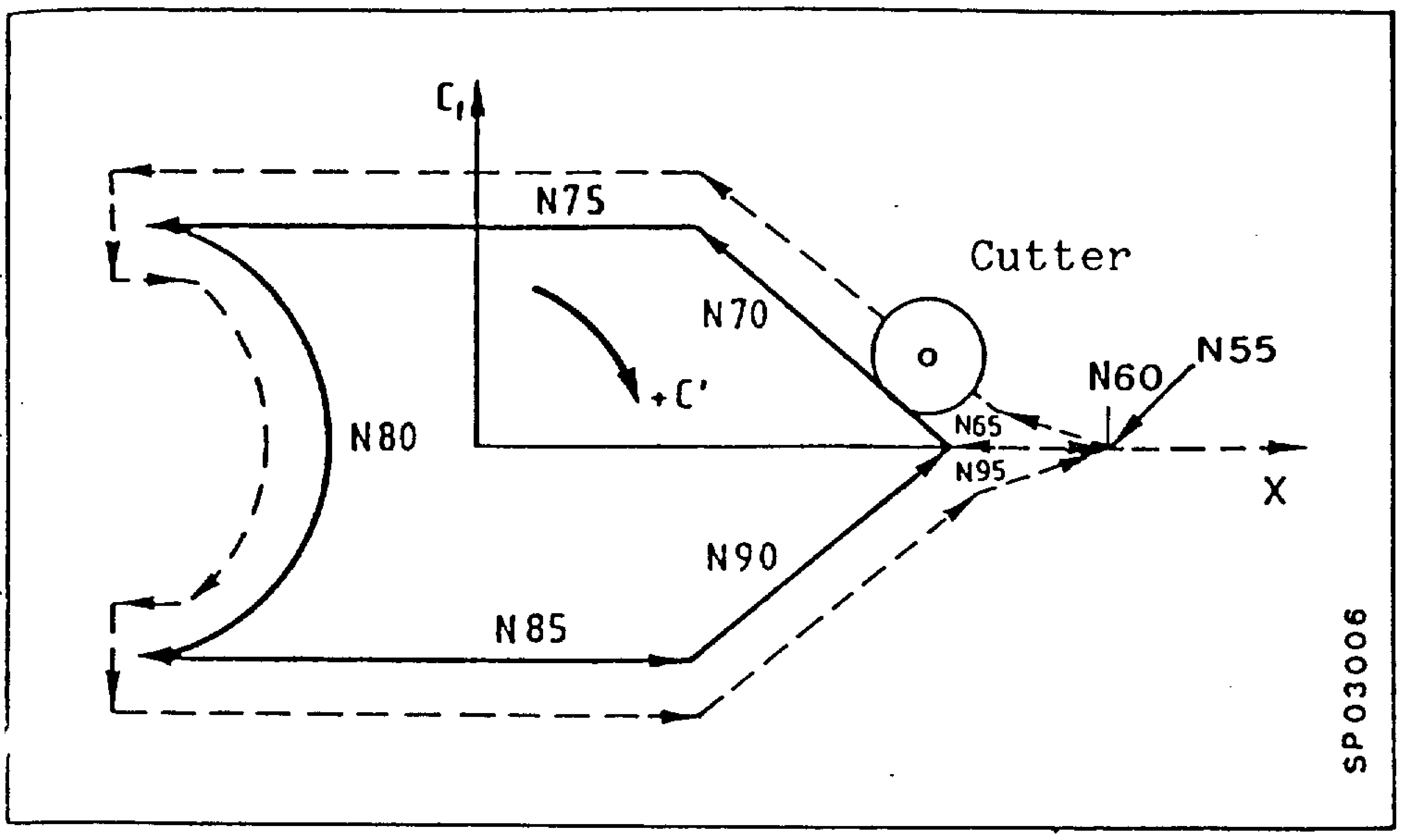
The tool path feed rate is programmed in the X Cf coordinate system where it is constantly retained. In G37 the tool path feed rate is monitored to prevent the max. permissible C axis speed from being exceeded.

- The precision which can be achieved on the part when the C axis is used depends on the current working radius (control in degrees).

- A block search via G37 blocks is not permitted.

- If automatic mode is interrupted, the system programmed with G36/G37 is retained. Traversing is thus possible in the Cartesian system in job mode.

Programming example: Cutting a "transverse contour" with TRANSMIT




```
%1234
  X.. C.. Z..
    *
    *                                 Real coordinate system
N50 X.. C.. Z..
N55 G0 x 120 C0 Z100 D50              Approach starting point
                                      Select length correction
N60 G37 G01 F200 Z90                  Activate transformation module with
                                      G37.
                                      Program in fictitious coordinate system
N65 G42 X90 C0                        Selection of cutter radius correction
                                      (C corresponds to fictitious axis Cf)
N70 X40 C40
N75 X-60
N80 G02 C-40 J-40                     Semi-circle
N85 G01 X40
N90 X90 C0
N95 G40 X120                          Cancellation of cutter radius compen-
                                      sation
N100 Z100                             Retraction in Z
N105 G36                              Deactivate transformation
N110 X.. C.. Z..                      Real coordinate system
    *
    *
```

3.20 Programmable coordinate rotation [1]

Programmable turning of coordinates permits a part program or section of a part program to be rotated by a certain angle. The rotation point is the sum of all zero offsets.

```
N... G59 X... Y... Z... 4... A...
```

G function for programmable additive zero offset (G59)

Values of programmable additive zero offset (X... Y... Z... 4...)

Angle for rotation of coordinates (A...)

- The rotation angle must not be negative or greater than 360 °.
- The rotation angle can also be programmed by chaining R parameters.
- The rotation angle is automatically deleted at the end of the program.
- The level can only be changed when the rotation angle is 0.
- The rotation angle is only displayed with G59.
- The main axes in the level selected can be rotated.

1) Option only basic control 4B, 4C

- After a G59 block with rotation angle, both axes to be rotated must be traversed to give the NC a new reference point.

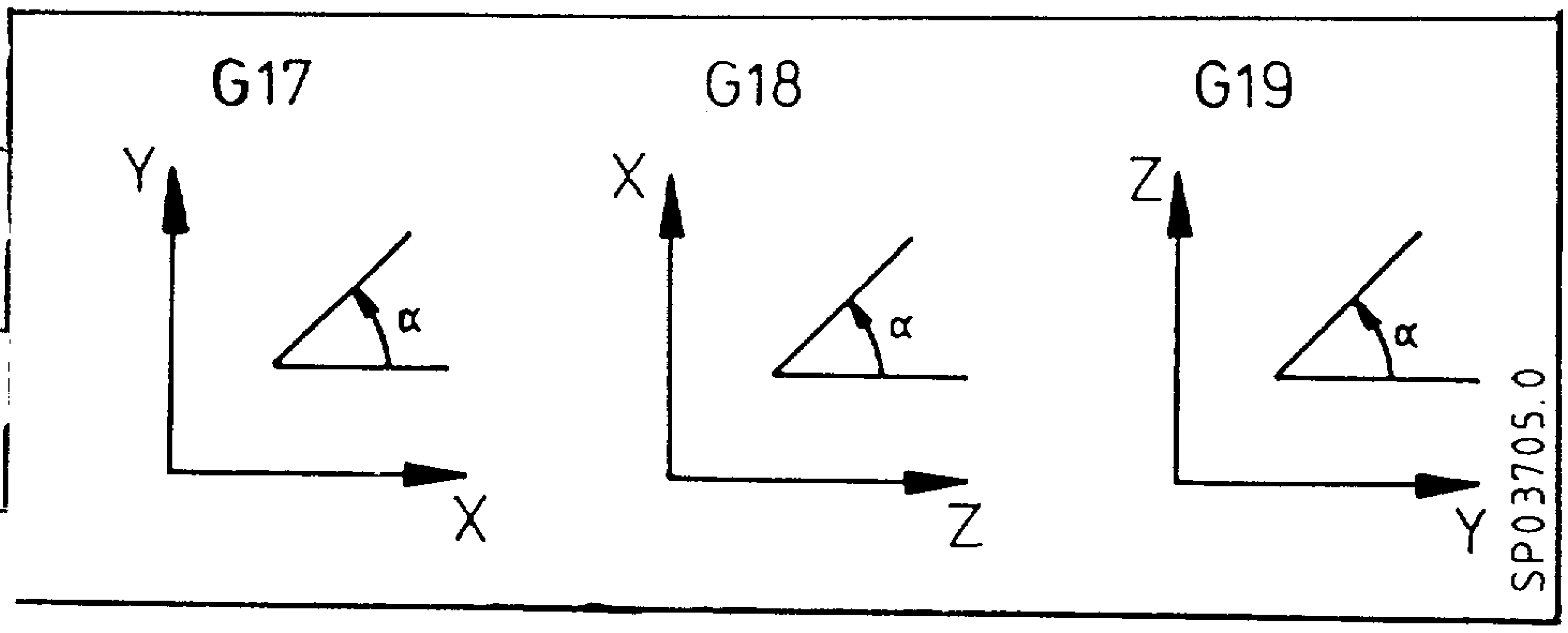
- In the 3 M system, rotation can change a 2 D block to a 3 D block. If the 3 D interpolation option is not available, an alarm is output.

- When an axis is mirrored, the block to be traversed is first rotated and then mirrored.

- In a G53 block, coordinate rotation is not effective.

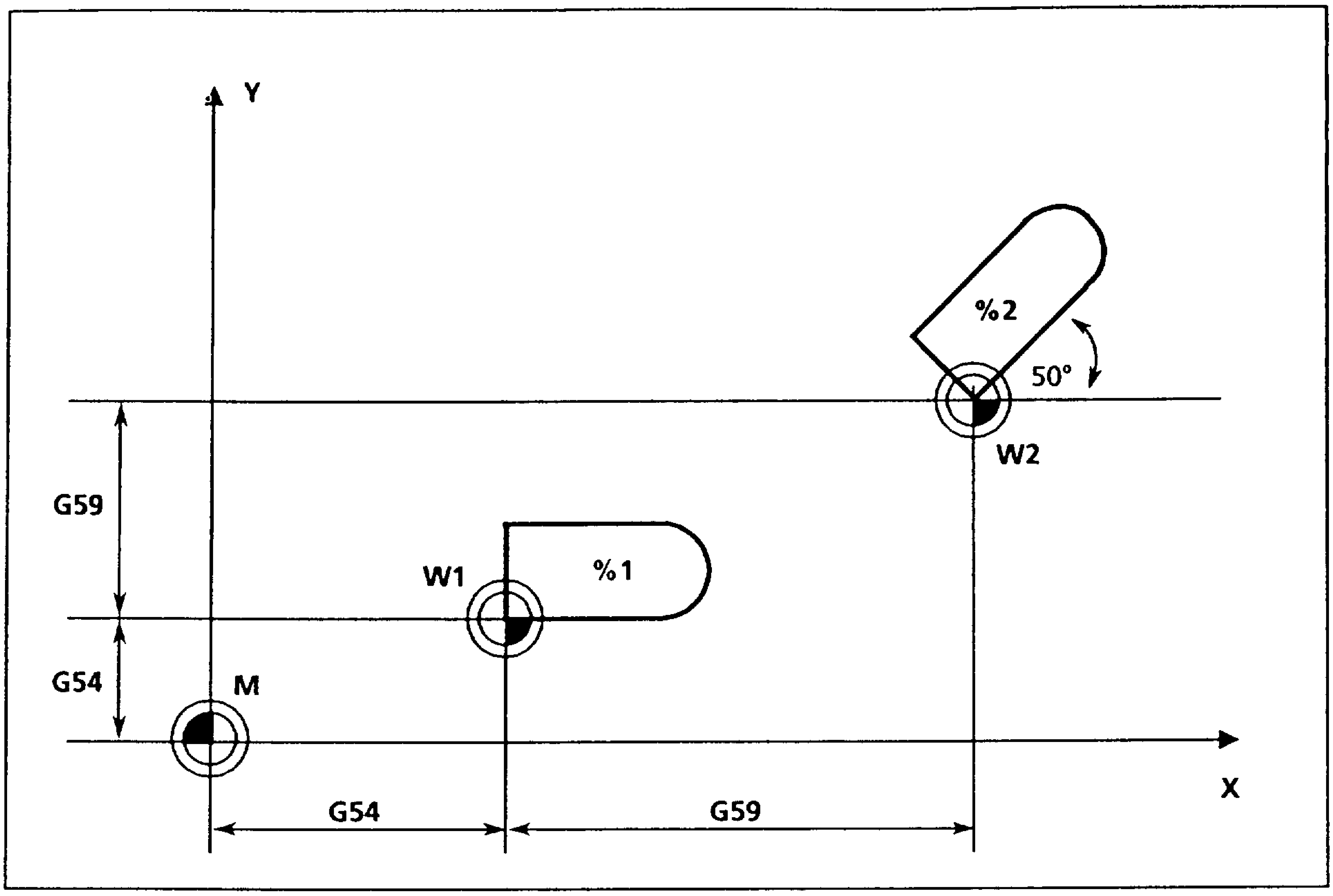
- If a G37 (Transmit) is programmed while coordinate rotation is active, both axes must be traversed in the next block.

Rotation angle direction in plane selected.



- If the rotation angle is written via @ 29, a @ 31 block must be programmed before @ 29, so that the rotation angle is effective from the next block.

Example:



M = Machine zero

W1 = Workpiece zero of % 1 (without programmable coordinate rotation)

W2 = Workpiece zero of % 2 (with programmable coordinate rotation)

% 1					% 2						
:					:						
N5	G54	LF			N5	G54	LF				
N10	G59	X0	Y0	LF	N10	G59	X25	Y30	A50	LF	Selection of coordinate rotation
N15	G00	X0	Y0	LF	N15	G00	X0	Y0	LF		
N20	G01	X20	LF		N20	G01	X20	LF			
N25	G03	Y10	O5	LF	N25	G03	Y10	P5	LF		
N30	G01	X0			N30	G01	Y0				
N35	G01	Y0			N35	G01	Y0				
N40	M30	LF			N40	G59	A0	LF			Deselection of coordinate rotation

3.21 Dynamic smoothing exponent for threads

This influences the run-up time of feed drives for threads as machine data 120 - 123 (acceleration) are ineffective when G33 is selected. The number of spindle actual value pulses for the speed of the feed axis are smoothed.

The smoothing and runup time is programmed as a dedicated block.

N.. G92 T. LF

The value is written into machine data item 358. The exponent can have the value 0 to 5. The value can also be programmed with R parameters.

Programmed value G92 T..	0	1	2	3	4	5
Cycle time times	0	1	3	7	15	31

Use the actual value scanning cycle time as time-base.

4. Miscellaneous and auxiliary functions S, T, M, H

The miscellaneous and auxiliary functions are output when the program block in which they are programmed is executed. A maximum of one M, one S, one T and one H function may be programmed in one block. The functions are output to the interface control in the following sequence:

- M
- S
- T
- H

A machine parameter is used to define whether the function is output before or while the programmed axis is in motion. See the machine tool maker's data.

If the functions are output while the axis is in motion, the following will hold true:

If a new function is to be in effect before an axis is traversed, the function must be programmed in the preceding block.

4.1 S Word

The S-function may be selected to specify:

- Spindle speed as a coded number
- Spindle speed in rpm. or 0.1 rpm
  (defined by machine parameter)

4.2 Tool function T

The tool function designates the tool necessary for a machining operation.

T1234

Address — T
Tool number:
2 decade field 1) — 12
4 decade field — 1234

4.3 Miscellaneous function M

M00 Programmed stop (unconditional)

M00 is used to stop an executing program to enable an operation e.g. a measurement, to be performed and when completed, program execution can recommence by pressing the "cycle start" key. Stored information is not affected. The miscellaneous function M00 functions in all automatic modes. Whether or not the spindle is stopped depends on the machine tool and is specified in the machine Programming Instructions.

M00 is effective in blocks programmed with or without path information. "M0" is recognized as an M00.

M02 End of program

M02 with reset to the first program block is programmed in the last program block. M02 may be programmed alone or together with other functions in a block.
The read-in procedure is stopped with M02.
M2 is recognized as an M02.
The control will revert to its default state (see program key).

1) with basic control 0

M17 End of subroutine

M17 may be programmed alone or together with other functions in the last subroutine block. M17 only signals the end of a subroutine. A subroutine call and M17 (e.g. with nesting) may not be programmed in the same block.

M30 Program end

M30 is similar to M02; automatic tape rewind is not possible. This can only be carried out by operating the tape reader manually.

M03, M04, M05 2) Main Spindle Control

(M19 only in conjunction with encoder on main spindle)

The following M words are defined for analog spindle control with analog spindle speed output:

M03 Spindle rotation clockwise
M04 Spindle rotation counter-clockwise
M05 Spindle stop
M19 Oriented spindle stop 2)

It is possible to stop the main spindle with orientation using M19 S. The angle can be programmed under address S in degrees (distance to the zero mark in M03 direction). The programmed angle under address S is retentive. When M19 is programmed without S, the stored value for the angle is active. M19 does not deselect M03 or M04.

Programming M19 S123 LF, whereby S can be omitted.

Unassigned additional functions

All additional functions except for M00, M02, M03, M04, M05, M17, M19, M30 are available for use.

2) with basic controls 3, 4

Exact information about the use of the individual functions is given in the respective program key of the particular machine. These functions are defined partially in DIN 66025.

4.4 Auxiliary Function H [1)]

The address H is available for switching functions or motions not controlled by the NC. One auxiliary function is available per block, which can be used for programming a maximum of 4 decades. Please refer to the Programming Instructions of the machine tool maker for further details.

4.5 Rapid switching and auxiliary functions [2)]

When programming M 12, H 1234, block processing in automatic operation is speeded up as the system does not wait for PLC acknowledgement. All functions are output rapidly with S and T functions (can be selected via machine datum). In the last block of a main program no rapid auxiliary functions may be programmed.

1) with basic control 4
2) with basic control 4C

5. R Parameters

Part programs may be programmed with parameters instead of numerical values for the dimension data. Parameters R00 - R99 1) can be assigned to all addresses exept N.

A parameter is set equal to a numerical value in the part program or in the subroutine. R parameters must always be programmed as a 2-digit number.A maximum of 10 parameters are allowed in a block.

From basic control 4C on, a total of 500 R parameters are available. They are divided into 5 groups of 100 R parameters each in groups 0 to 4. The parameters can be addressed within each group via addresses R00 to R99. The groups can be selected via the functicn @28x (x = 0 to 4, depending on the group). The following examples and sections 5.1 to 5.5 are valid for one R parameter group in basic version 4C.

Note: When using cycles (L95 to L903) only R parameters R00 to R49 are freely available. The R parameters from R50 onwards are used for cycle calculation.

Example:

```
L5100
N1   X-R05   YR01   LF
N2   ZR29           LF
N3   XR05   Y-R01   LF
.
N50  M17            LF
```

Parameters R01, R05, and R29 are used in subroutine.

```
%    4081           LF
N1   . . .          LF
.
.
N37  R01   10.   R29-20.05   R05   500.   LF
```

Value assignments

```
N38  L5102  LF
```

Subroutine L51 is called, it will run twice

R01 = 10.
R05 = 500.
R29 = -20.05

1) with basic controls 0, 1, 2: 50 parameters R00 to R49

5.1 Parameter definition

With parameter definition, individual R parameters are assigned to signed numerical values. The R parameters are assigned in part programs or subroutines. Up to 10 parameter definitions may be programmed in one block.

Programmed operation	Execution	Result
R01 10.78 R02 95.34 R03 -555.1	R01 +10.78 R02 +95.34 R03 -555.1	R01 = +10.78 R02 = +95.34 R03 = -555.1

5.2 Assigning parameters in a program

Direct assignment for dimension data only. An address is given the value defined by the R parameter.

Programmed operation	Execution	Result
XR01 ZR X-R03	XR01 ZR03 X-R03	X = +10.78 Z = -555.1 X = +555.1

Arithmetic assignment

To the numerical value of an address, the control performs a signed addition or subtraction with the parameter value.

Programmed operation	Execution	Result
X 20.78 - R01 Z 44.9 - R03 X 10.1 R02	X = (20.78 - 10.78) Z = 44.9 - (-555.1) X = + 10.1 +95.34	X = 10 Z = 600 X = 105.44

The sequence address, numerical value, parameter must be maintained.
An unsigned parameter or number is assumed positive.

5.3 Operations with parameters

Arithmetic function	Programmed operation	Execution	Result is stored in
Addition	R01 R02	R01 + R02	R01
Subtraction	R01 - R02	R01 - R02	R01
Multiplication	R01 · R02	R01 · R02	R01
Division	R01/R02	R01 : R02	R01
Parameter Definition and Addition	R01 10 R02	R01 = 10 R01 + R02	R01
Parameter Definition and Subtraction	R01 -10 - R02	R01 = -10 R01 - R02	R01
Square root 1) Sine 1) Arctan 1)	@ 10R01 @ 15R01 @ 18R01	$\sqrt{R01}$ sinR01 arctan (R01:R02)	R01 R01 R01

The @ symbol is produced by pressing the [7/%] key and then the [÷/R] key.

1) Basic controls 3 and 4

Only R parameters may be multiplied or divided with one another; i.e. a parameter and a number may not be multiplied or divided together. The decimal point defines the operation as multiplication. The block skip character "/" defines the operation as division.
The sequence determines the order in which the expression is evaluated. @10, @15, @18 must be programmed in one block.

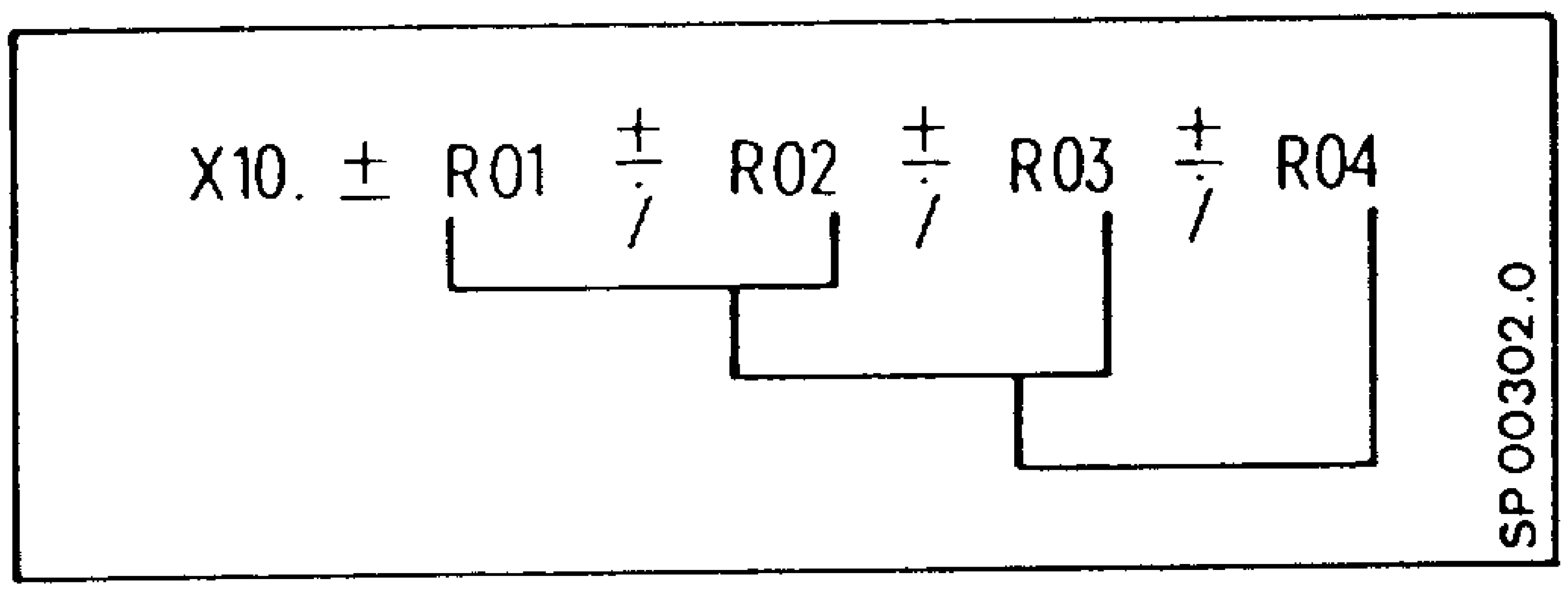
The control calculating time is app. 10 msec per operation. Only one operation per block may be programmed.

<u>Range:</u>	smallest value:	$1 \cdot 10^{-8}$
	largest value:	$2^{27} -1$
<u>Display:</u>	floating decimal point (± .8) to (± 8.) [1)]	

1) Basic controls 0, 1, 2 (±.7) to (±7.)

5.4 Parameter chaining

Through parameter chaining a parameter value is altered continually as it loops through a section of program or a subroutine. A calculation is performed whenever chaining parameters are encountered in a running program.
The last parameter of the chain remains unchanged.
A maximum of 4 parameters may be chained.



Conventions for evaluating an expression

A new R parameter is calculated from the chaining of two parameters and the sign between them. The parameter value is sign-dependent (i.e. it can be positive or negative).

An example of chaining 2 parameters

R01 + R02	$R01_{new}$ = R01 + R02
-R01 + R02	$R01_{new}$ = R01 + R02
R01 - R02	$R01_{new}$ = R01 - R02
-R01 - R02	$R01_{new}$ = R01 - R02
R01 · R02	$R01_{new}$ = R01 · R02
R01 / R02	$R01_{new}$ = R01 / R02

An example of chaining 4 parameters

-R01 + R02.R03-R04	$R01_{new}$ = R01 + R02
	$R02_{new}$ = R02 · R03
	$R03_{new}$ = R03 - R04
	$R04_{new}$ = R04

The parameters as well as the parameter value may be a signed number.

Example:

%9534LF N1 L0105 R01-10. R02 81. R03 3. LF N6 L0204 R04-1. R05 2. R06 4. LF . N100 M30 LF	A subroutine call to loop 5 times and 4 times resp. Parameters are defined prior to subroutine entry as: R01 = -10. R02 = 81. R03 = 3. R04 = -1. R05 = 2. R06 = 4. R07 = -1.
L0100 (3-parameter chaining) N5 X1000. -R01 +R02/R03 LF N10 M17 LF	
L0200 (4-parameter chaining) N17 Y500. +R04. R05 +R06 +R07 LF N20 M17 LF	

The following numerical values are taken by the motion axes and the parameters.

L0100		X -	R01 +	R02 /	R03
Pass	Definition	1000.	-10.	81.	3.
1	value ass.	1010.	71	27.	3.
2	value ass.	929.	98.	9.	3.
3	value ass.	902.	107.	3.	3.
4	value ass.	893.	110.	1.	3.
5	value ass.	890.	111.	0.333	3.

L0200		Y +	R04 ·	R05 +	R06 +	R07
Pass	Definition	500.	-1.	2.	7.	-1
1	value ass.	499.	-2.	6.	3.	-1.
2	value ass.	498.	-12.	9.	2.	-1.
3	value ass.	488.	-108.	11.	1.	-1.
4	value ass.	392.	-1188.	12.	0.	-1.

At the end of a program, the parameters take on the values assigned to them in the last performed parameter manipulation. This value remains stored until the parameter is redefined or a parameter manipulation is done resulting in a new value.

5.5 Programming examples using parameters

Example: a rectangle

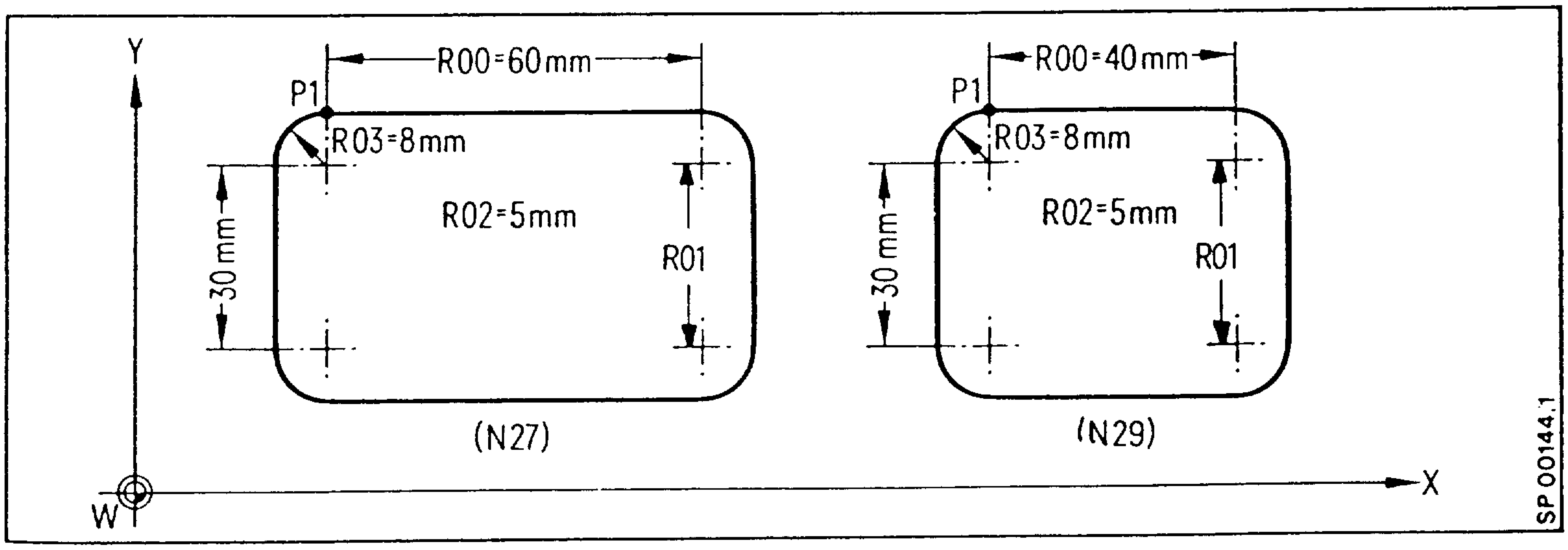
The following subroutine illustrates the machining of a rectangle in the X, Y plane whose sides vary dimensionally. The rectangle sides are assumed parallel to the machine axes.

Subroutine

```
L4600
N5    G01   G64   G91   Z-R02                           LF
N10                     X R00                           LF
N15   G02               X R03   Y-R03   I0   J-R03      LF
N20   G01                       Y-R01                   LF
N25   G02               X-R03   Y-R03   I-R03   J0      LF
N30   G01               X-R00                           LF
N35   G02               X-R03   Y R03   I0   J R03      LF
N40   G01                       Y R01                   LF
N45   G02               X R03   Y R03   I R03   J0      LF
N50   G01               Z R03                           LF
N55   M17                                               LF
```

Subroutine call:

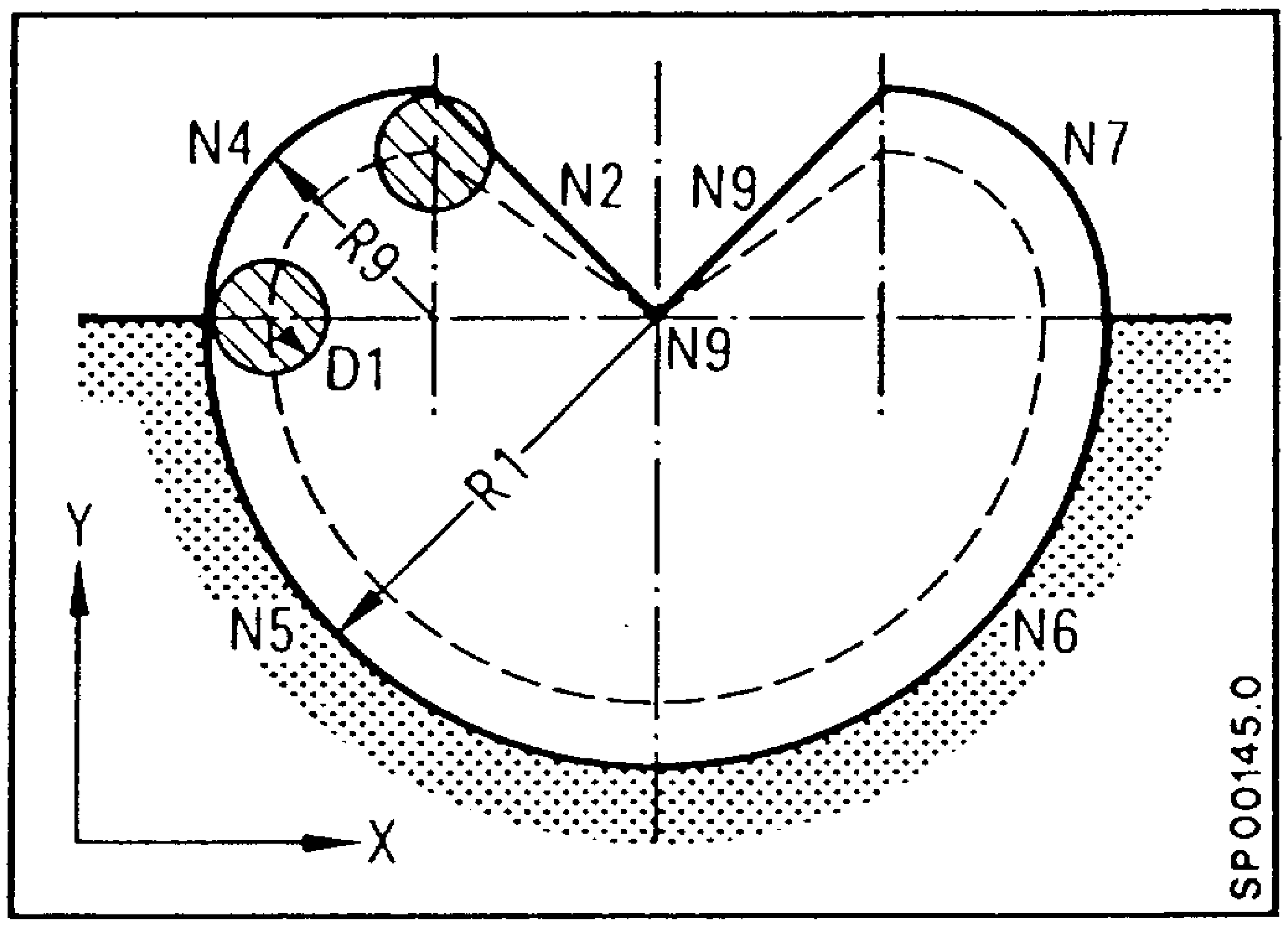
```
N26   G90   X . . .   Y . . .                   LF   1st subroutine
N27   L46   R00 60   R01 30.   R02 5   R03 3    LF   entry
N28   G90   X . . .   Y . . .                   LF   2nd subroutine
N29   L46   R00 40.                             LF   entry
```



R02 = Dressing depth in the Z axis direction
P1 = Start and end point of the subroutine

Example: Machine an internal semi-circle surface

The following subroutine illustrates a stock removal and finishing operation for a semi-circle. The contour radius and the cutter entry radius are written as parameter variables. Each time the subroutine is run, the actual part dimension can be compared to the programmed dimension. The resulting difference is stored as a additive tool wear offset.



Subroutine:

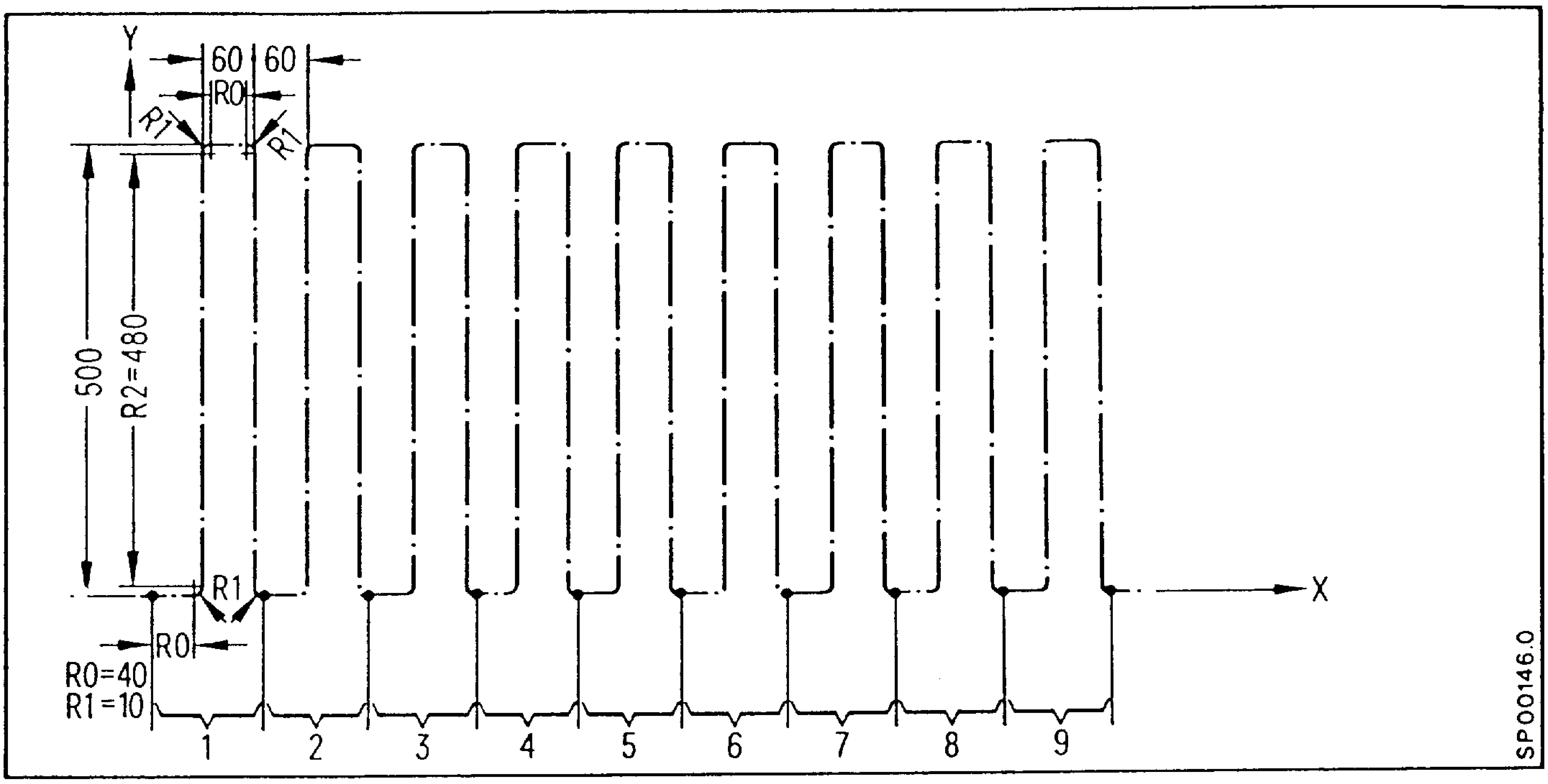
```
L0100
N1 R01-R09                            LF   Calculate cutter entry radius
N2 G00 G64 G91 G17 G41 D01                 Position to cutter entry point
X-R01 YR09                            LF
N3 R01+R09                            LF   Reset R01 to original value
N4 G03 X-R09 Y-R09 I0. J-R09          LF   Position to contour
N5 XR01 Y-R01 IR01 J0.                LF   Machining
N6 XR01 YR01 I0. JR01                 LF   Machining
N7 X-R09 YR09 I-R09 J0.               LF   Position from contour
N8 R01-R09                            LF   Calculate workpiece centre
N9 G00 G40 X-R01 Y-R09                LF   Positioning
N10 R01+R09 M01                       LF   Reset R01 to original value
N11 M17                               LF   End of subroutine
```

Subroutine call:

```
% 5873                                LF
N1 ...                                LF
N2  L0101  R01 50.  R09 10.           LF
N3  ...
.
.
```

Example: Straight Milling

The path transitions are programmed with radii to avoid a reduction in the feed rate. In this manner dwell marks are avoided during a path direction change.



L34 is called by another program:

```
.
N15 L3409 R00 40. R01 10. R02 480. F200 LF
.
```

Subroutine:

```
L3400
N1 G01 G64 G91 XR00                   LF
N2 G03         XR01 YR01 IO. JR01     LF
N3 G01              YR02              LF
N4 G02         XR01 YR01 IR01 JO.     LF
N5 G01         XR00                   LF
N6 G02         XR01 Y-R01 IO J-R01    LF
N7 G01              Y-R02             LF
N8 G03         XR01 Y-R01 IR01 JO.    LF
N9 M17                                LF
```

5.6 Empty buffer store, L99; @ 31

A series of control signals from the interface control (parallel interface or PLC) are registered in the active store of the NC indirectly via buffer stores. Associated with these signals are:

- external tool offset
- mirror image
- external additive TO

If these functions which are actuated in the active program are to be effective in the block following their selection, the block buffer store must be emptied. Alternatively the selected control signal only becomes active several blocks later.

In each program the buffer store can be emptied by a single call-up of the subroutine L99. The subroutine L99 must be defined as follows:

```
L9900      LF
@ 31 M17   LF
```

Example:

Activation of external tool offset, e.g. after measurement of the tool.

N15 M...	- PLC takes away input enable with the help of M function and transmits the external tool compensation.
	- Afterwards PLC gives input enable again
N20 G04 X...	- Dwell time is bigger than 1,2 x max. PLC cycle time
N25 L99	- Empty buffer store

5.7 Branch Conditions

The @ symbol is produced by pressing the [7 / %] key and the [· / R] key.

5.7.1 @ 00 Unconditional jump

Application: Program sections may be omitted using the unconditional jump. Blocks skipped are not processed.

Programming: @ 00 ± 1234

- @ 00 — Operation: unconditional jump
- ± — Branch destination is located before (-) or after (+) the branch operation: The control searches for the branch destination in the given direction
- 1234 — Block no. of the branch destination: max. 4 digits

Different branch destinations can be achieved by using an R parameter with sign. An application of this special case is shown as follows:

Branch destination located before (-) branch operation

Standard case	Special case
.	R01 has, for example, the value 1
.	
N98 X.. Y..	N98 X.. Y..
.	N99
.	.
.	.
.	.
.	.
N215 @00 - 98	N215 @00 - 98 R01
.	.

Branch destination located after (+) branch operation

Standard case	Special case
.	R01 has, for example, the value 1
.	
N215 @00 + 280	N215 @00 + 280 R01
.	.
.	.
.	.
.	.
N280 X.. Y..	N280 X.. Y..
.	N281 G04...
.	.
.	.

Note: Branch destinations must always be blocks with a block number. This also applies, when the branch destination is varied by the R parameter.

Branch operations require time (max. 10 msec. per block skipped).

5.7.2 @01, @02, and @03 Conditional jump

Application:

Branch operations may be executed according to the following conditions:

equal	@ 01
greater	@ 02
greater or equal	@ 03

Programming: @.. ± 1234 R.. R..

- @.. — Operation with conditional jump
- ± — Branch destination is located before (-) or after (+) the branch operation: The control searches for the branch destination in the given direction
- 1234 — Block no. of the branch destination: max. 4 digits
- R.. R.. — R parameter comparison for the conditional jump:

1st parameter equal to 2nd parameter	@ 01
1st parameter greater than 2nd parameter	@ 02
1st parameter greater or equal to 2nd parameter	@ 03

1st parameter (first R..)
2nd parameter (second R..)

Different branch destinations can be achieved by using an R parameter with sign. An application of this special case is shown as follows:

Branch destination located before (-) branch operation

Standard case	Special case
	R10 has, for example, the value 2
.	.
.	.
.	
N98 X.. Y..	N98 X.. Y..
.	N99 M..
.	N100 X.. Z..
.	.
N215 @ 01 - 98 R01 R02 (= above 01)	N215 @ 02 - 98 R10 R01 R02 (≥ above 02)
N216 G.. M..	N216 G.. M..
.	.
.	.

Branch destination located after (+) branch operation

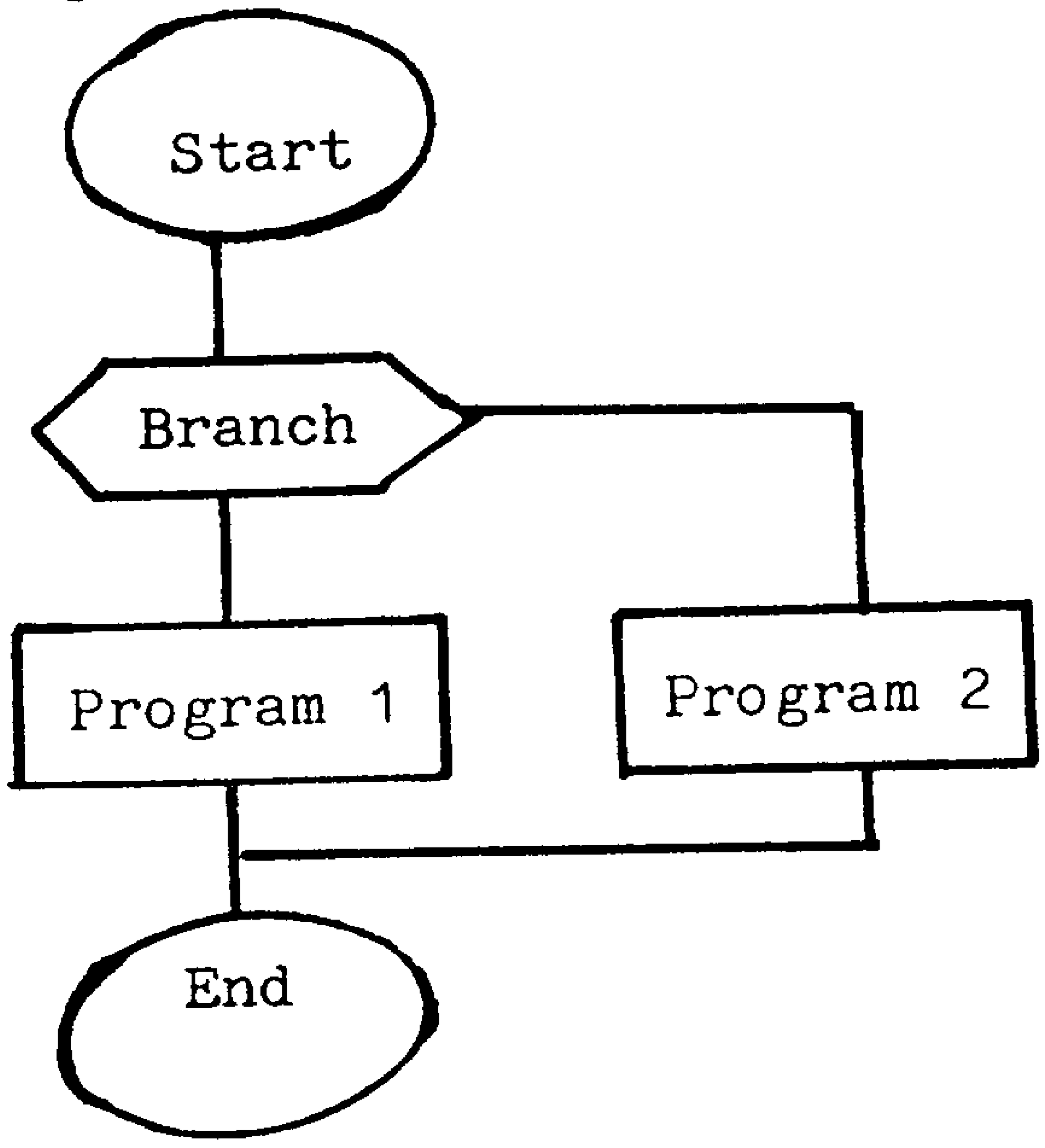
Standard case	Special case
	R15 has, for example, the value 5
.	.
.	.
.	
N215 @ 03 + 280 R20 R25 (≥ above 03)	N215 @ 01 + 280 R15 R20 R25 (= above 01)
N220 G.. M..	N220 G.. M..
.	.
.	.
.	.
N280 X.. Y..	N280 X.. Y..
.	N285 G04
.	.

Note: Branch destinations must always be blocks with a block number. This also applies, when the branch destination is varied by the R parameter.

5.7.3 Concluding example for unconditional and conditional jumps

Requirement: A branch to 2 programs within a program

Flow chart:



Programming:

```
N005  G01  X..
N010  @01 045  R10  R20     Conditional jump to program 2, if
                            R10 = R20
N015       T
N020   Program 1
N025       |
N030       |
N035       ⊥
N040   @  00 070            Unconditional jump to block 70, in
N045       T                order to skip program 2
N050       |
N055   Program 2
N060       |
N065       ⊥
N070  G00  X..
```

5.8 @ 10 "Square root" 1)

Application: Extract square root

Programming: @ 10 R..

Operation: extract square root — @ 10

R parameter for value assignment and result — R..

Example:

N10 R10 25 R10 is loaded with 25

.

.

.

N75 @ 10 R10 Extract square root of the value defined in R10

N80 R10 has the value 5 from the next program block (here N80)

.

Note:

- Only positive values are to be entered
- Largest value 99999999.
- Smallest value .00000001

1) Only with basic control 3, 4

5.9 @ 15 "Sine" 1)

Application: Calculate the sine of an angle

Programming: @ 15 R..

Operation: calculate sine (→ @ 15)

R parameter for value assignment and result (→ R..)

Example:

N10 R17 45	R17 is loaded with 45
.	
.	
.	
N75 @ 15 R17	Calculate sine of value stored in R17.
N80 .	R17 contains the value .7071067 from the next program block (here N80)
.	
.	

Note:

- Positive and negative values are permitted
- Largest value +359.99999
- Smallest value -359.99999

1) Only with basic control 3, 4

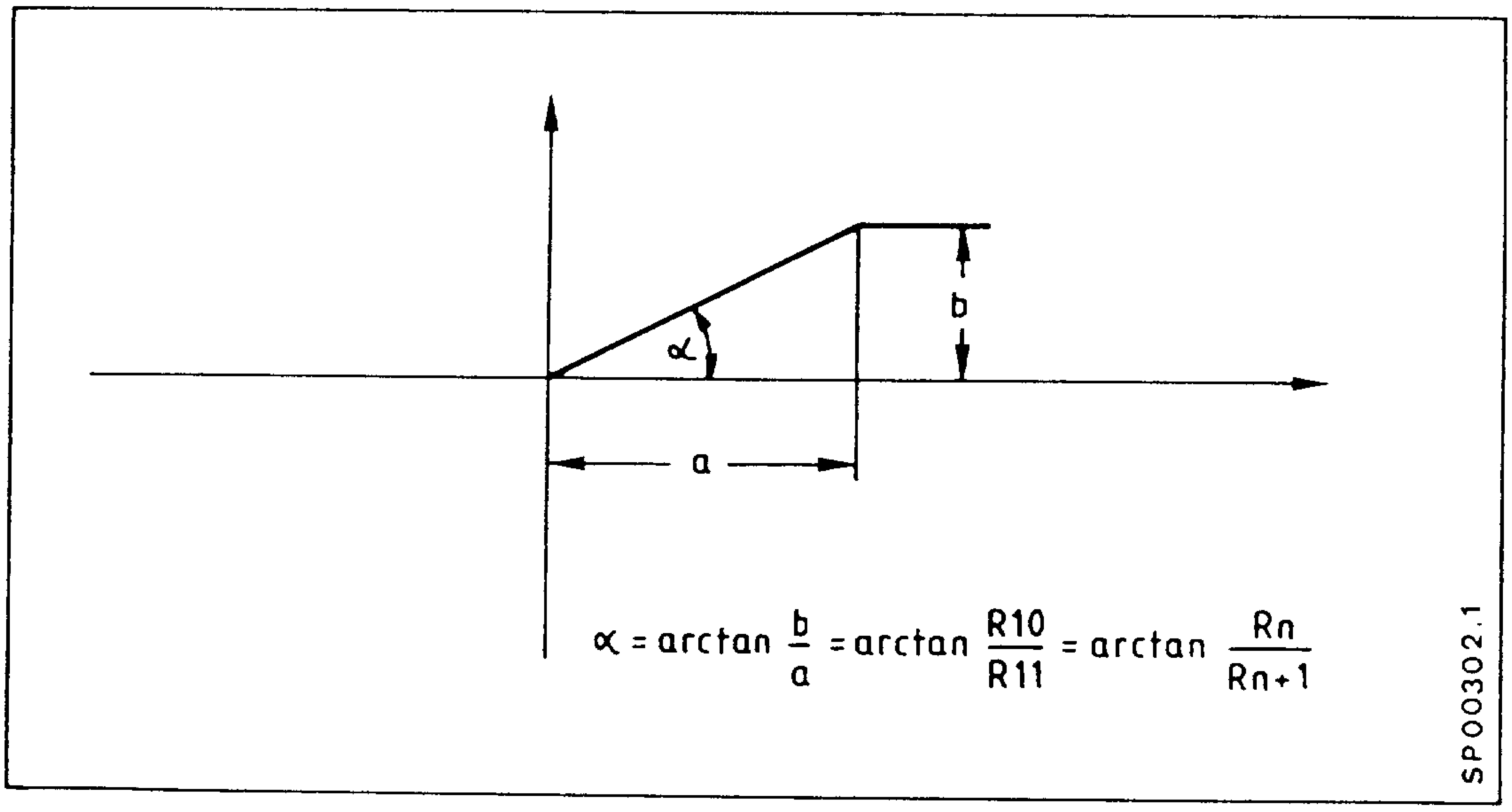
5.10 @ 18 "Arctan" 1)

Application: Determination of an angle using the arctan function

Programming: @ 18 R ...

Operation: calculate arctan

R parameter (Rn) for value assignment b and result
R parameter (Rn+1) for value assignment



n = Parameter number

Example:

```
N10 R10   20    load 20 in R10 (b)
N15 R11   30    load 30 in R11 (a)
N20 @ 18 R10    Arctan calculation
R10 = + 33.69007° Result is in R10
```

1) Only with basic control 3, 4

5.11 a 20 "load address parameter" [1]

Application: During measurement cycles for instance, it is necessary not only to program the numerical values of words as variables, but also the axis addresses. By varying the axis addresses the measurement cycle can operate in all NC-axes. Similar values for the number values (R00 to R99) of several address parameters @ 90 to @ 93 are available. @ 20 issues the statement to the NC, to load and address the parameter with the address code (see table below).

Axis	Address code
X	1
Y	2
Z	3
4.	4

Programming:

a) Load address parameters @ 90 to @ 93 by direct specification of address code via the axis No. (machine data)

N... @ 20 @ 93 ± 2 LF

- @ 20 — Load address parameter with address code (axis No.)
- @ 93 — Address parameter
- ± — Sign + or - allows direction changes when exchanged
- 2 — Axis No. for Z e.g.

1) Only with basic control 3, 4

b) Load address parameters @90 to @93 by indirect specification of the address via R parameter

N... @20 @93 ± R49 LF

R49 is loaded with the address code of the required axis before execution of this block.

The address is allocated to the specified address parameter (@90 to @93) using this program. The axis direction can be reversed if required using the sign.

5.12 @ 90 to @ 93 "address parameter" 1)

Application: Parameter for axis addresses, so that not only numerical values but also addresses can be programmed as variables. Instead of axes X and Z, the address parameters are programmed.

Programming in the user program.

Example:

a) Address parameter without R parameter assignment.

N... @ 90 ± 12345.678 LF

- Address parameter defined e.g. as Y axis. — @ 90
- Sign for the axis value — ±
- Axis value — 12345.678

b) Address parameter with R parameter assignment.
Instead of a fixed numerical value, the axis value can be entered via an R parameter.

N... R01 12345.678 LF

- Value assignment — R01 12345.678

1) Only with basic controls 3, 4

5.13 @ 29 "load/read from system memories" [1)]

Application: Flexible access to the system memory locations of the NC control are possible from the user program with this function. The contents of the system memory can be read and other values can be entered into the system memory in some cases (loaded).

Access to following system memory locations is possible:

- Actual tool offset no.
- Tool offset
- Adjustable zero offset
- Programmable additive zero offset
- Current offset
- R parameters
- Machine data
- Machine data bit
- Actual values
- Background memory
- Special-purpose bit
- External zero offset (additional compensation)
- Measuring values

Note: If, for example, the programmable zero offset (G 59) is to be changed, @ 31 must have been programmed previously. This is always necessary if @ 29 is used to change system data which influence the paths to be traversed.

1) Only with basic controls 3, 4

Programming: N... @ 29 1 63 01 R17 LF (own block)

Operation: load/read from the system memory — @ 29

1st position (1):
- 1 = read system memory location
- 2 = load system memory location
- 3 = read machine data bit

2nd + 3rd position (63): No. of R parameter (here R63). The value read is stored in this R parameter or the value from this R parameter is loaded in the system memory location.

4th + 5th position (01): Memory area coding
- 01 = Tool geometry
- 03 = adjustable zero offset
- 04 = programmable additive zero offset
- 08 = current offset
- 09 = R parameter
- 10 = Machine data
- 12 = Actual value
- 18 = Background memory
- 19 = Special bit
- 20 = Current tool correction No.
- 13 = External zero offset (additional compensation)
- 34 = Value store for measuring data

R17: R parameter No. (here R17). The contents of this R parameter are a 4-decade coded number and specify the required memory from the memory area defined by @ 29, positions 4 + 5.

1st + 2nd position: e.g. axis number (for further explanations see overview p. 5 - 26)

3rd + 4th + 5th position: Identifying number e.g. number of the settable zero offset (for further explanations, see p. 5 - 26).

Example:

Application: The R parameter R01 is loaded during a gauging cycle with zero offset values for axis Z. This measured value is to be loaded into the system memory of the 3rd zero offset group in the SINUMERIK 3T.

Programming:

```
.
.
.
N50 R07 1 @29 10409 R07
N55 R05 1003 @29 20403 R05
.
.
.
```

Explanation of the R parameters

R01	165.015	New zero offset value of Z axis
R04		contains the value to be transferred
R05	1003	Coding @29 (1st axis, 3rd zero offset group)
R07	1	Coding for @29 (R parameter No.) (leading zeros can be suppressed).

Overview of possible memory access operations (max. values)

e.g. N100 RAB 1 2 3 4 5 LF

N110 @29 1 2 3 4 5 RAB LF

Position 1	Meaning	Position 2 & 3	Meaning	Position 4 & 5	Meaning: System memory area for	Position 1 & 2	Meaning	Position 3 & 4 & 5	Meaning
1 or 2	Read location Write location	00 to 99	max. 100 R pars.	01	Tool compensation	01 02	Length Radius	001 to 064 1)	max.64 groups 1)
1 2	Read Write	00 to 99	max. 100 R pars.	03	Settable zero offset	01 02 03 04	1st axis 2nd axis 3rd axis 4th axis	001 to 004 012	4 groups for 4B\4C 12 groups
1 2	Read Write	00 to 99	max. 100 R pars.	04	Programmable additive zero offset	01 02 03 04	1st axis 2nd axis 3rd axis 4th axis	001	1 group
1	Read	00 to 99	max. 100 R pars.	08	Current offset (Tool offset) (also mirrored) + zero offset	01 02 03 04	1st axis 2nd axis 3rd axis 4th axis	001	1 group
1 2	Read Write	00 to 99	max. 100 R pars.	09	R parameters	00 00 to 04	No significance (up to basic version 4B) R parameter Group No. (from basic version 4C)	000 to 099	100 numbers
1	Read	00 to 99	max. 100 R pars.	10	Machine data	00	no signif.	100 to 479 2)	380 2) numbers
3	Read bit	00 to 99	max. 100 R pars.	10	Machine data bit	00 to 07	Bit no.	400 to 479 3)	80 3) numbers
1	Read	00 to 99	max. 100 R pars.	12	Actual value	01 02 03 04	1st axis 2nd axis 3rd axis 4th axis	001	1 group
1 2	Read Write	00 to 99	max. 100 R pars.	18	Background memory	00	no signif.	000 to 099	100 numbers
3	Read bit	00 to 99	max. 100 R pars.	19	Special bit	00 to 07	Bit no.	001	1 group
1	Read	00 to 99	max. 100 R pars.	20	Current tool offset number	00	no signif.	001	1 group
1 2	Read Load	0 0 to 9 9	max 100 R pars.	1 3	External zero offset (additional compens.)	01 02 03 04	1st axis 2nd axis 3rd axis 4th axis	0 0 1	1 group
1 2	Read Load	0 0 to 9 9	max. 100 R pars.	3 4	Value store	00	no signif.	0 0 0 to 0 9 9	100 value store numbers

* Bit 0 : 1 = block active
Bit 1 : 1 = trial run active
Bit 2 : 1 = switching sensor swung out
Bit 3 : 1 = G70 active
Bit 4 : 1 - Simulation active (from 4C)
Bit 5 to 7: Unassigned

RAB is a freely selectable R parameter (AB = 00 to 99). RAB defines the system memory location to be read or to be loaded.

1) Basic control 3: 32 possible groups
2) Basic control 3: 100 ... 419 = 320 numbers
3) Basic control 3: 400 ... 419 = 20 numbers

5.14 @28 Selecting the R parameter group (only basic version 4C)

Application: This function can be used to select one of the groups 0 to 4 for the R parameters. Each group has 100 R parameters (R 00 to R 99).

Programming:

```
                           @28   1    LF
                           _|_  _|_
Selecting the               |    |
R parameter group __________|    |
                                 |
Group No. (0 to 4)_______________|
e.g. group 1
```

With power On, reset and M02/M30, group 0 is automatically selected.

An exchange of values over group limits is possible with function @29.

Example: R66 from group 0 is to be loaded with the value from R77 from group 1.

N5 @28 0	Selection group 0
N10 R10 01077	RAB assignment (R10) with R77 from group 1
N15 @29 26609 R10	@29 Read R parameters Target = R66, source = R77 (Address for source in R10)

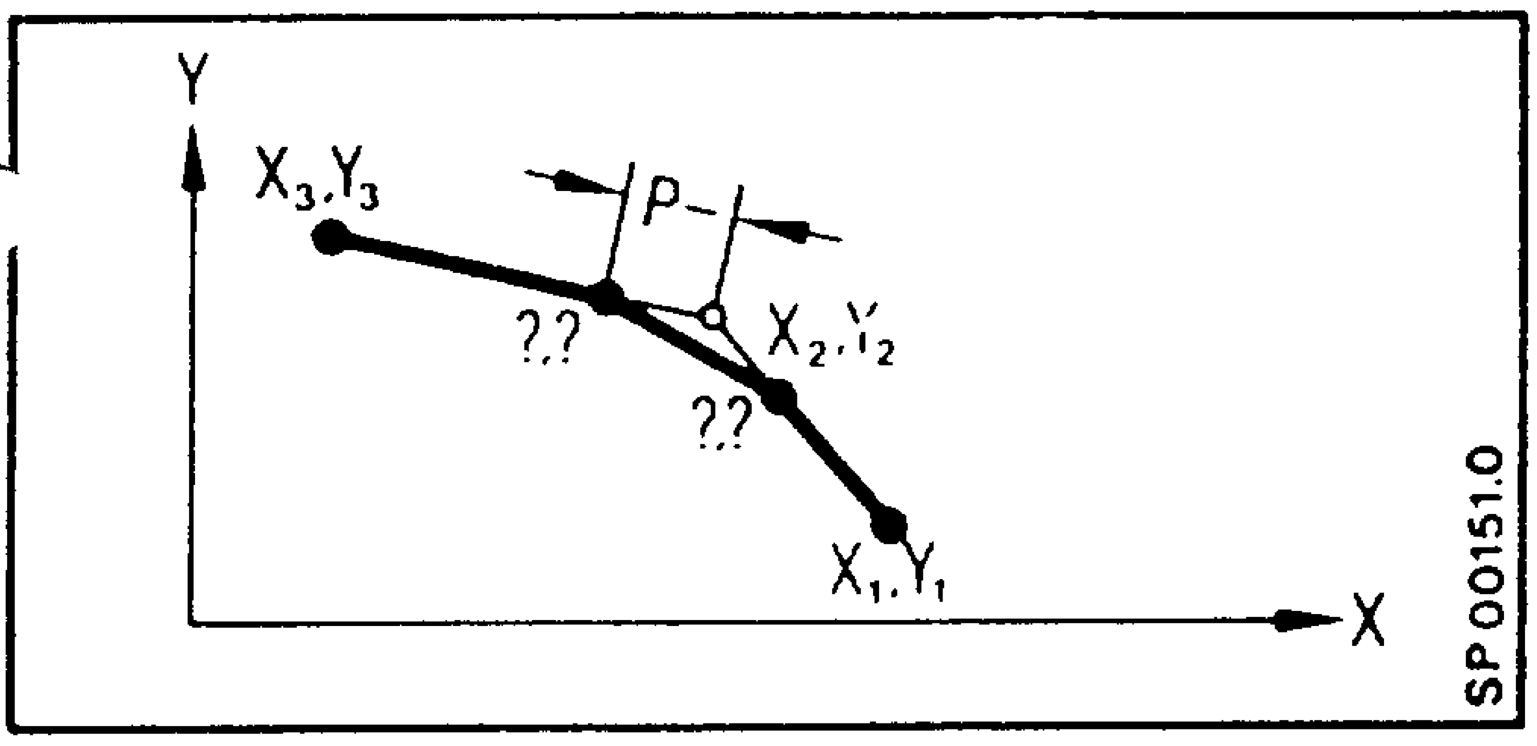
6 Programming aids

6.1 Automatic insertion of champfers and radii

At corners a chamfer or a radius can be inserted automatically. Examples 1 and 2 show the basic elements.

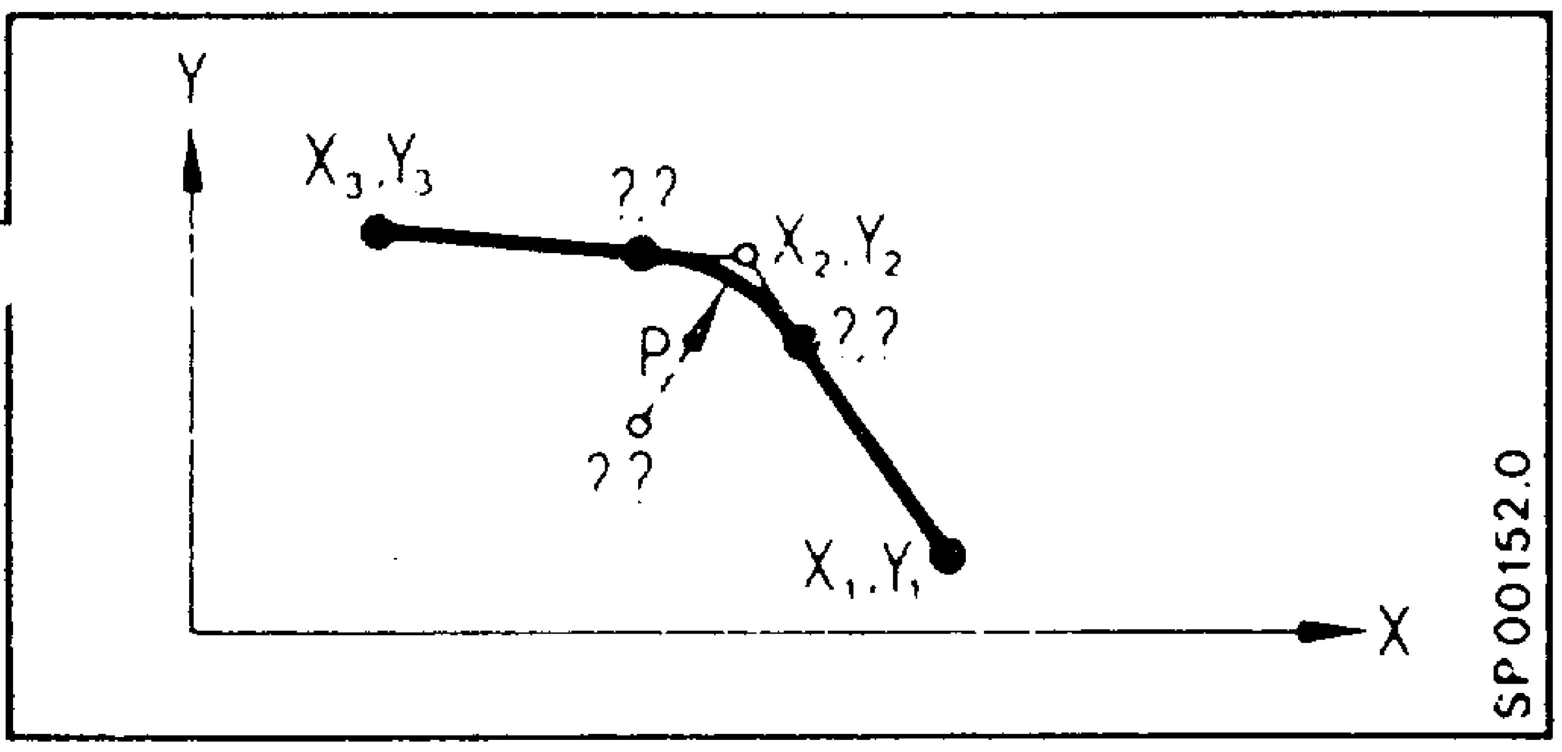
P-...	means the insertion of a chamfer
P...	means the insertion of a radius

1 Chamfer



N... X_2... Y_2... P-...
N... X_3... Y_3...
The inserted chamfer must not be bigger than the smaller one of both distances.

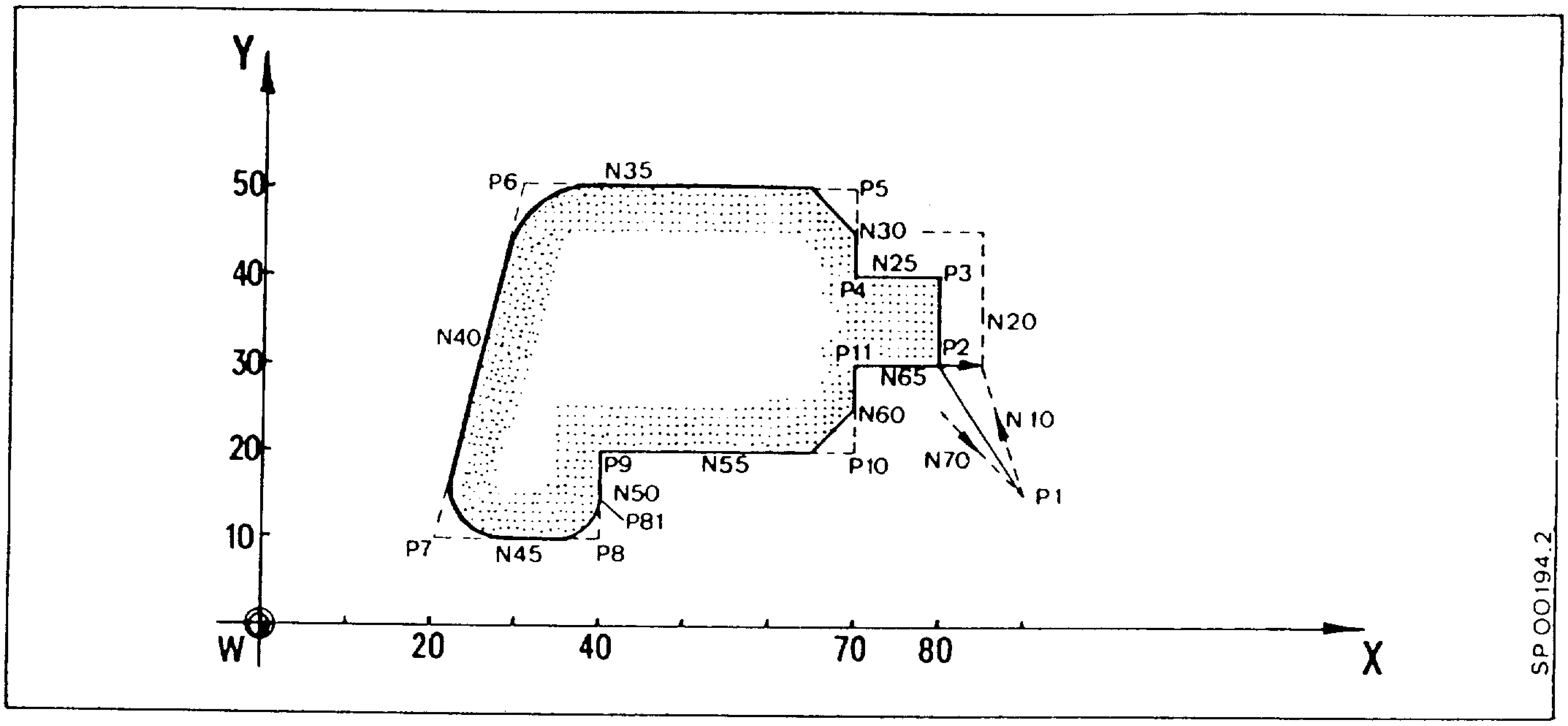
2 Radius



N... X_2... Y_2... P...
N... X_3... Y_3...
The inserted radius must not be bigger than the smaller one of both distances.

$\overline{X_2\ Y_2,\ X_3\ Y_3}$ or $\overline{X_1\ Y_1,\ X_2\ Y_2}$

Example



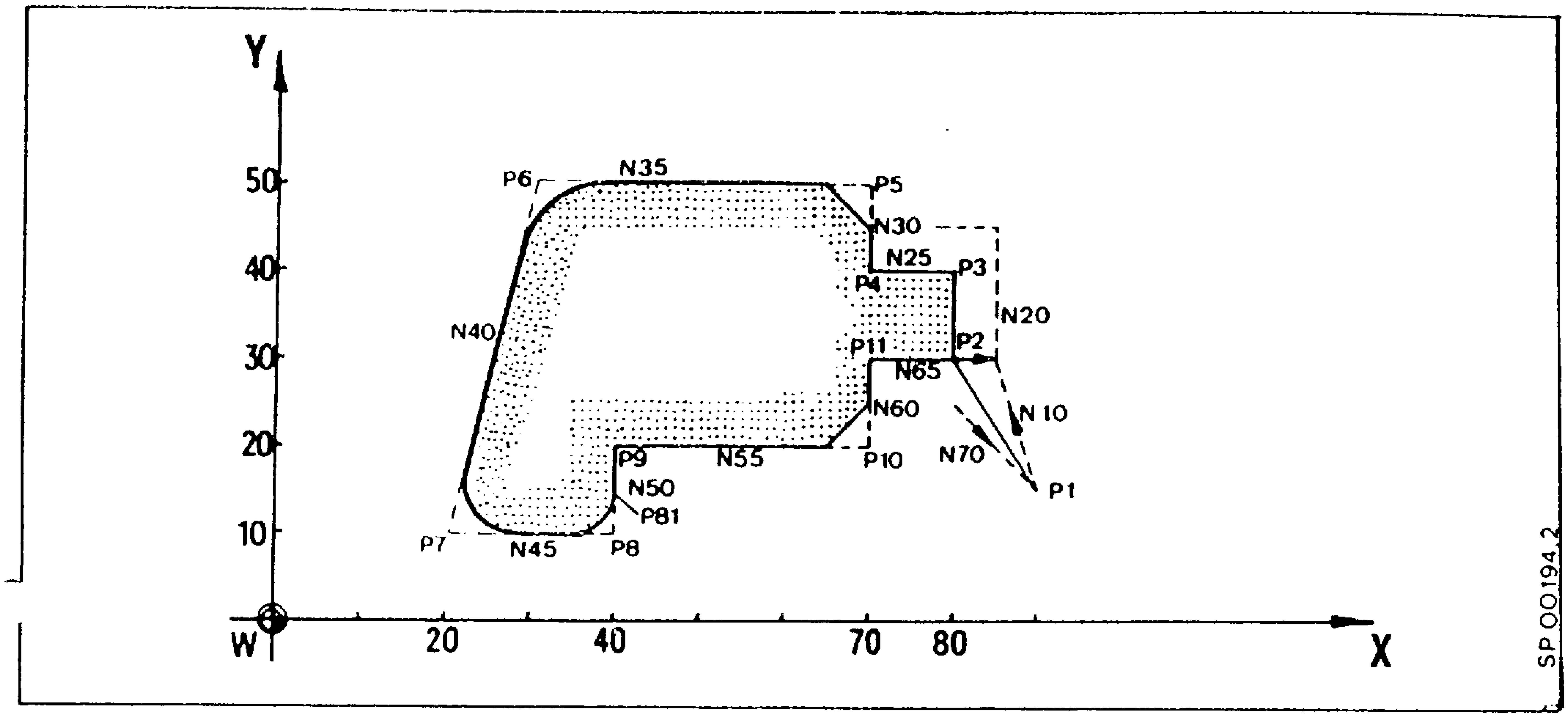
.
.
.

```
N10 G00 G90 G42 X80.  Y30.      LF (P2)
N15             Z-10.           LF (P2)
N20 G91 G01     Y10.            LF (P3)
N25             X-10.           LF (P4)
N30             Y.10  P-5.      LF (P5)
N35 G64         X-40. P18.      LF (P6)
N40        X-10.Y-40. P5.       LF (P7)
N45             X20.  P5.       LF (P8)
N50 G60         Y10.            LF (P9)
N55             X30.  P-5.      LF (P10)
N60             Y10.            LF (P11)
N65 X10.                        LF
N70 G00 G90 G40 X90. Y15.       LF (P1)
N75             Z10.            LF
N80 M02                         LF
```

The value of P- or p must be < than the programmed dimension of the block in which it is to become active.

6.1.1 Miscellaneous and auxiliary functions in linked blocks

A block is considered to be linked to an adjacent block when a radius or chamfer is used to connect the two blocks.



A block containing miscellaneous and auxiliary functions may be programmed between linked blocks.

Example: see above and example 6 - 1

```
N45 X20 P5 LF      (contour definition P7 - P8)
N46 M..    LF
N50 Y10
```

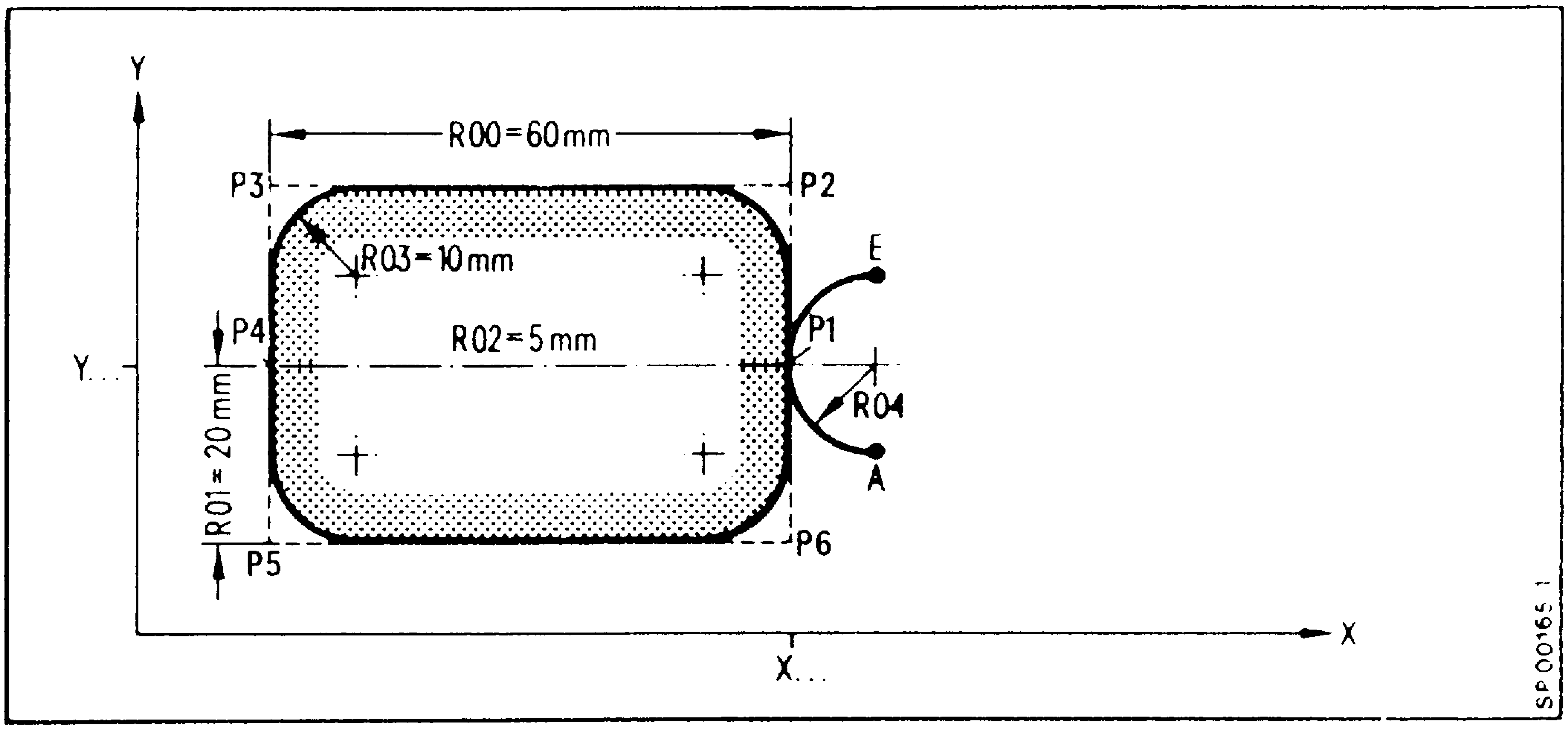
The miscellaneous functions become effective at the end of the block (i.e. P8)(see above).
Free-cutting will result in P81.

Within the contour definition, G60 is automatically generated by the control at abrupt transitions (inserted chamfer).
This inserted G60 is not active as the reset position of the 10th G-group.

6.1.2 Example: Rectangular pattern

The following subroutine describes a rectangular pattern with variable rectangle sides, corner radii and depth advance. The radius parameter R03 must be smaller than R01, or one half of R00.

Example:



Subroutine:

```
L6 100
G42 D18 X... Y....              LF        (A)
N0 G01 G91 G64 Z-R02            LF        (A)
N1 G02 X-R04 YR04 I0. JR04      LF        (P1)
N2 G01 YR01 PR03                LF        (P2)
N3     X-R00 PR03               LF        (P3)
N4     Y-R01                    LF        (P4)
N5     Y-R01 PR03               LF        (P5)
N6     X R00 PR03               LF        (P6)
N7     Y R01                    LF        (P1)
N8 G02 X R04 Y R04 I R04 J0.    LF        (E)
N9 G01 Z R02                    LF        (E)
N10 M17
```

Subroutine call:

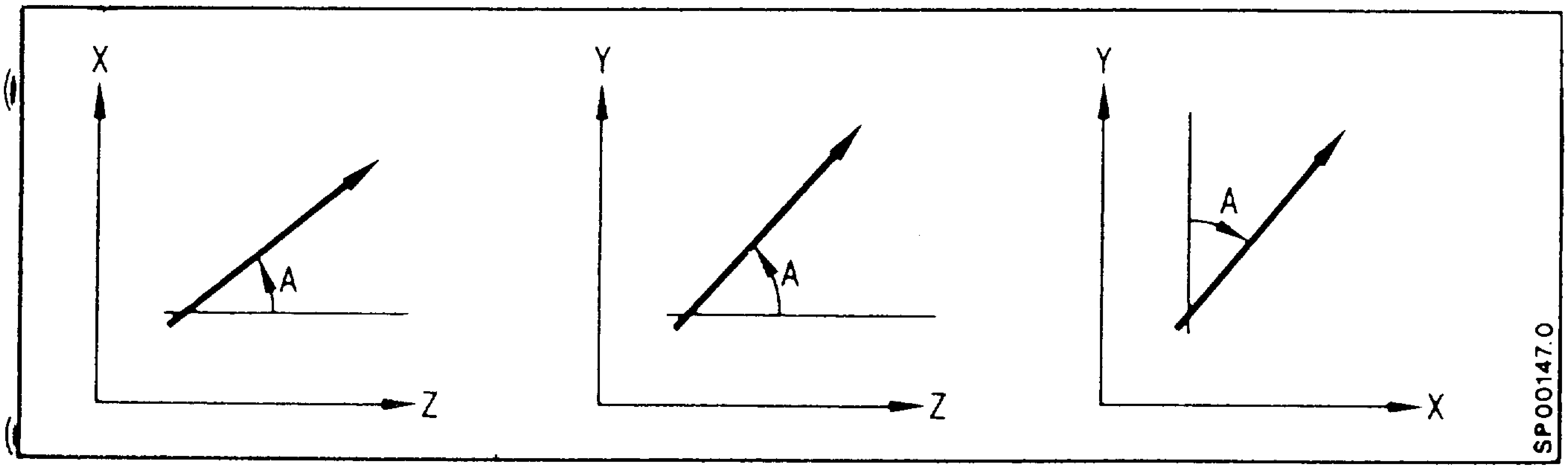
```
N25 G90 LF
N30 L6101 R00 60. R01 20. R02 5. R03 10. R04 10. LF
```

6.2 Blueprint programming (basic control 4B, 4C)

Blueprint programming means to directly input multi-point cycles according to the workpiece drawing. Intersections of straight lines are input as coordinate values or as angles.

The individual linear motions can have a direct transition in the form of a corner, they can be rounded by radii or chamfered. Chamfer and transition radii are input only using their size. The geometrical calculation is made by the control. The end point coordinates are programmed in absolute or incremental dimensions.

Angle (A): Input unit 0.00001°
The angle (max. 359.99999°) is always related to the positive axis vector with the highest address and to the positive direction of the other axis.
Address sequence: Z - Y - X

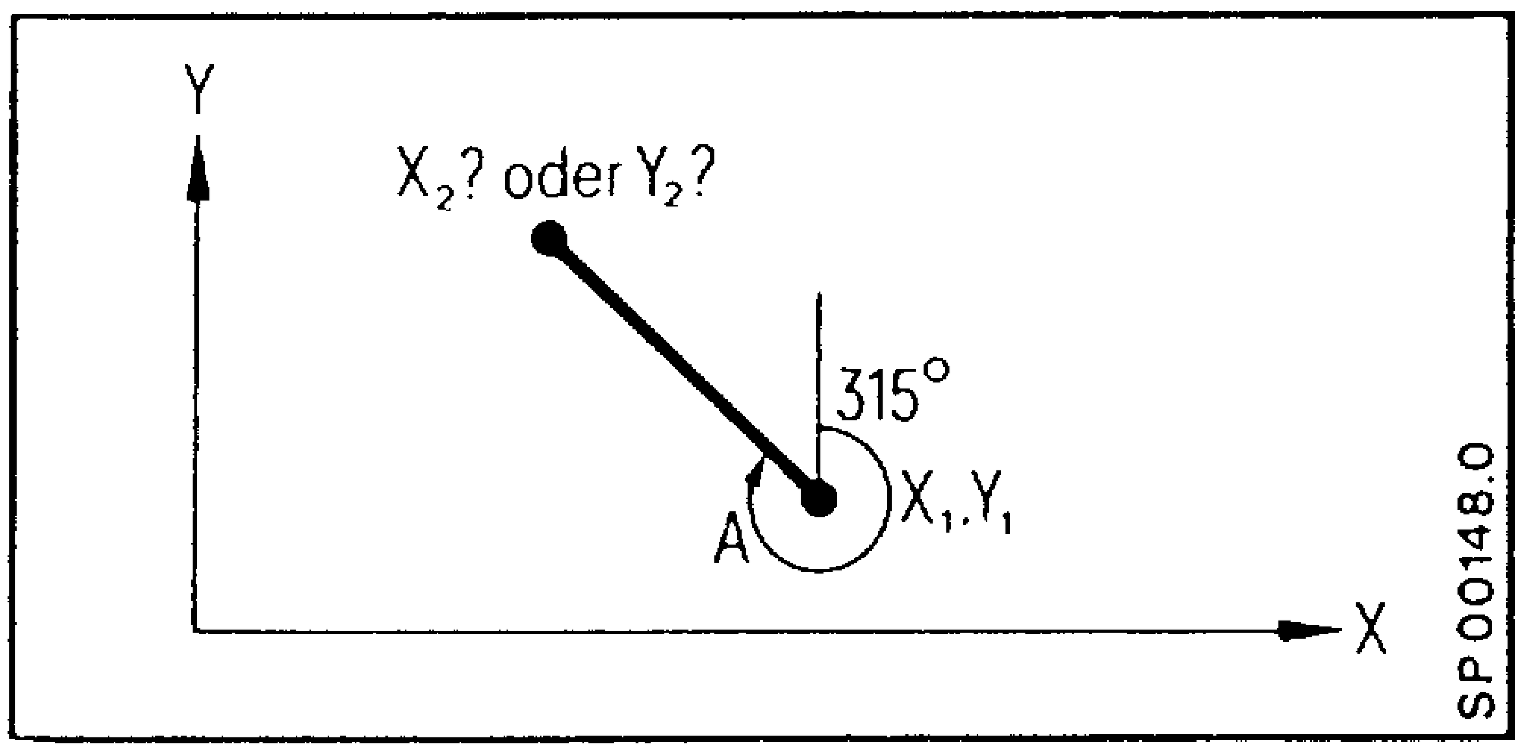


Caution:
The blueprint programming is effective in the selected plane. 3D-machining is not possible. If there are four axes, the programmed axis must be the main axis.

6.2.1 Contour cycle programming

Examples 1 to 8 represent basic elements of the contouring cycle programming. These basic elements can be combined.

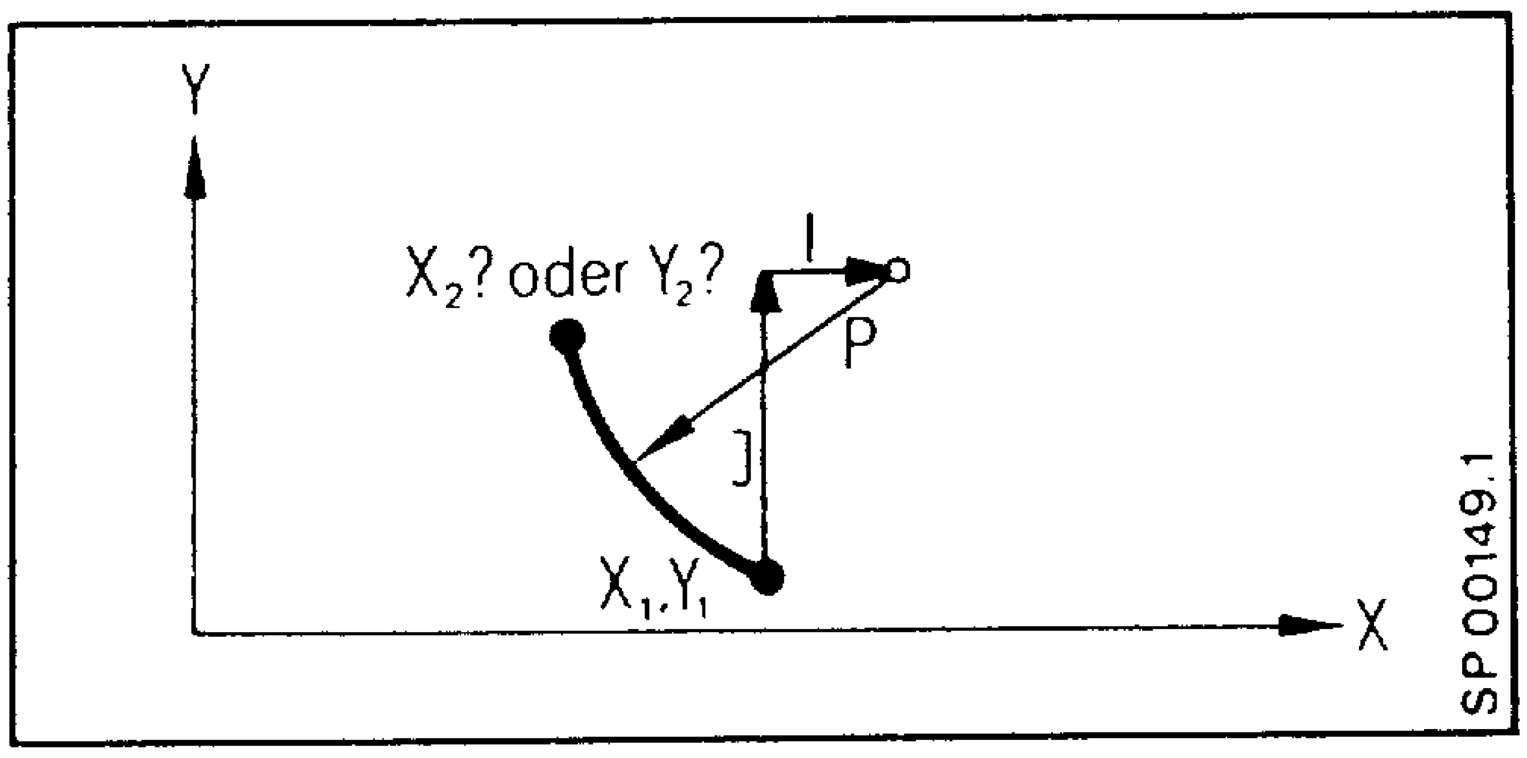
1. Two-point cycle



N...A...X_2... (or Y_2)

The second end position coordinate is calculated by the control

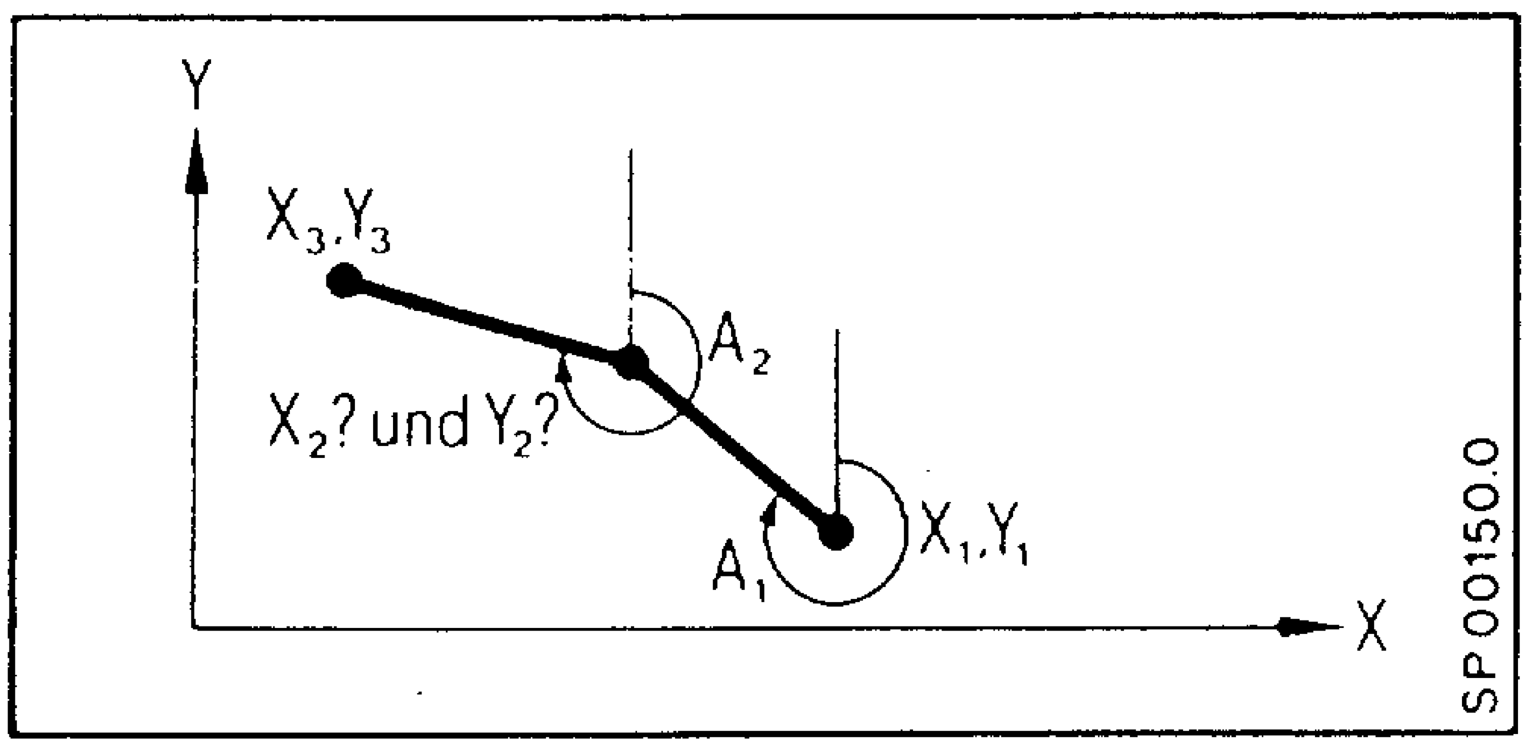
2. Circular arc



N..G02 (or G03) I..J..P..X_2 (or Y_2)

The circular arc is limited to one quadrant. The second end position coordinate is calculated by the control. In the contour cycle, both parameters I and J must be programmed even if the value is zero.

3. Three-point cycle

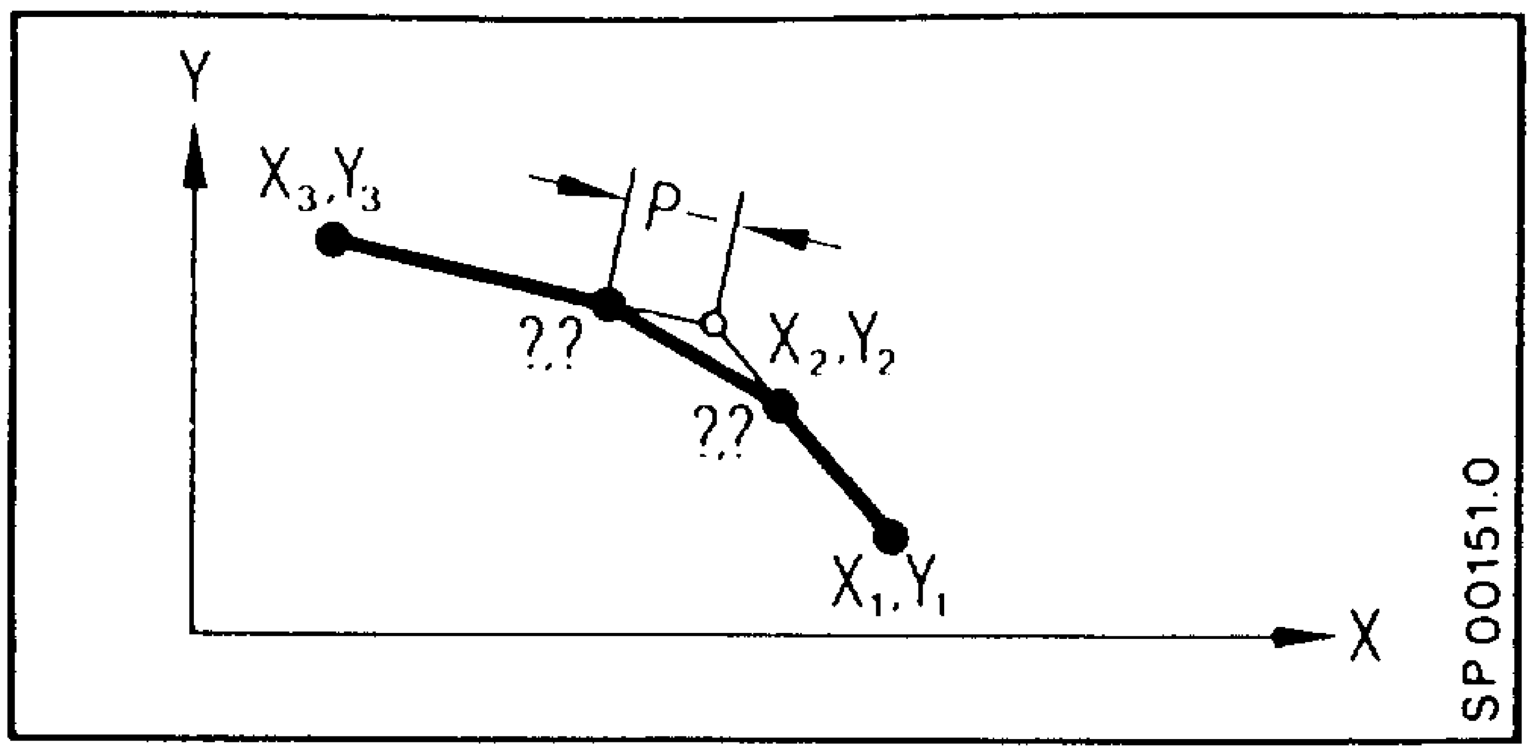


N...A_1...A_2...X_3...Y_3

The control calculates the intersection coordinates (X_2, Y_2) and generates 2 blocks. The angle A2 refers to the non-programmed support point (X_2, Y_2).

4. Chamfer

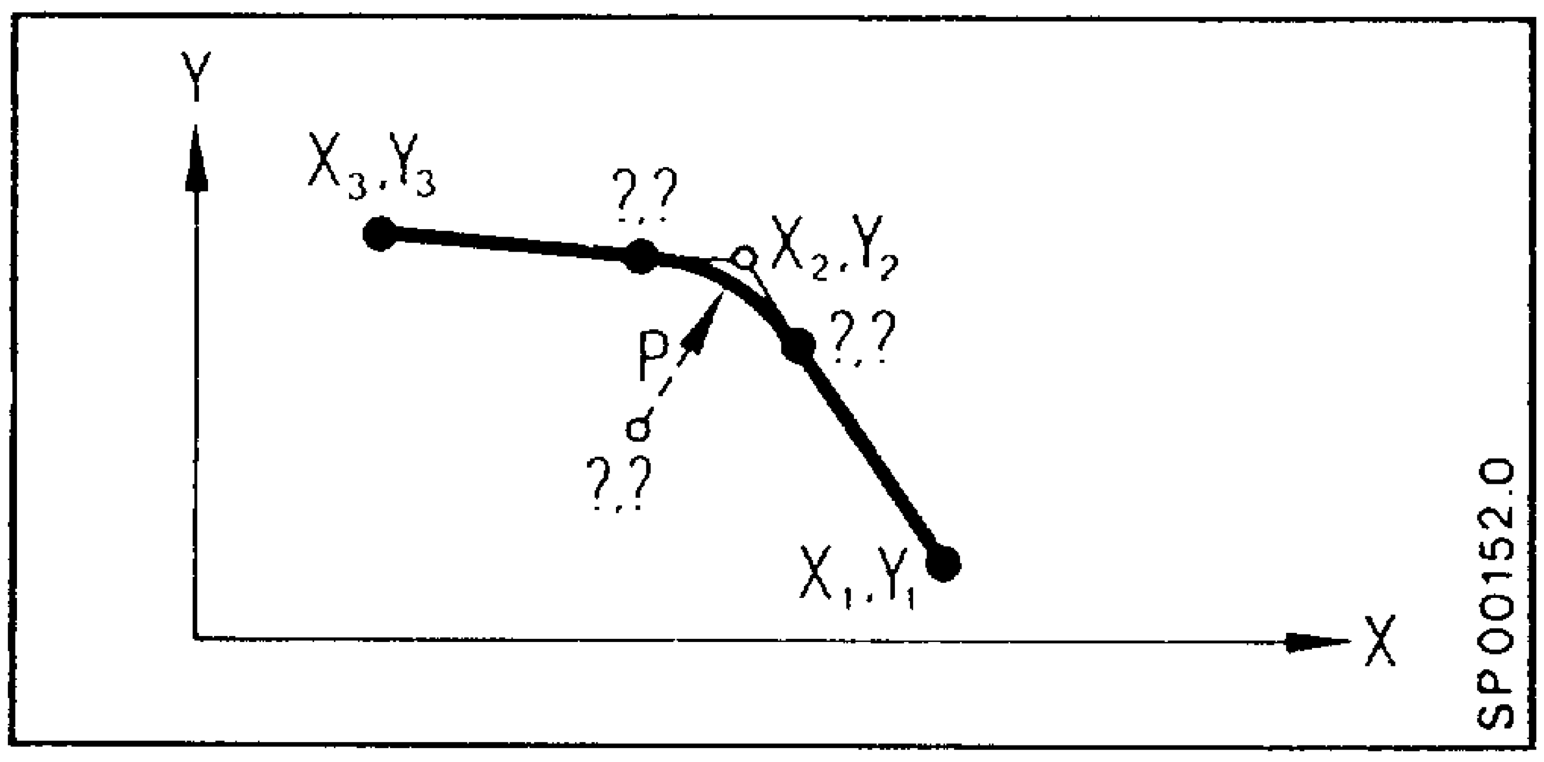
N... X_2... Y_2... P-...
N.. X_3... Y_3



P-... means insertion of a chamfer. The "Minus" character here does not mean a sign, but specifies P as chamfer.

5. Radius

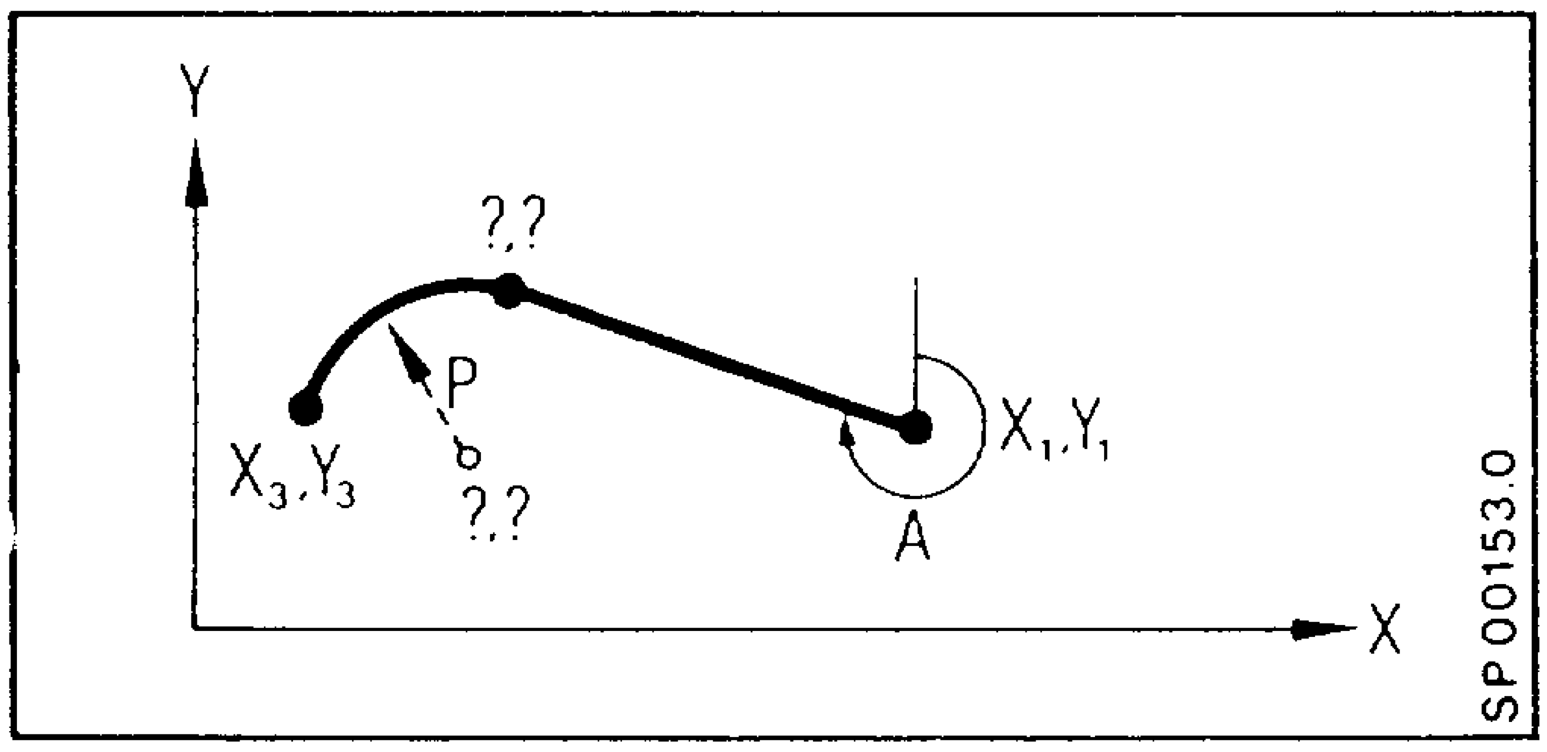
N... X_2... Y_2... P...
N... X_3 *



The radius inserted must be smaller than the smaller of the two distances

$\overline{X_2Y_2, X_3Y_3}$ or $\overline{X_1Y_1, X_2Y_2}$.

6. Straight line - circular arc (tangential)

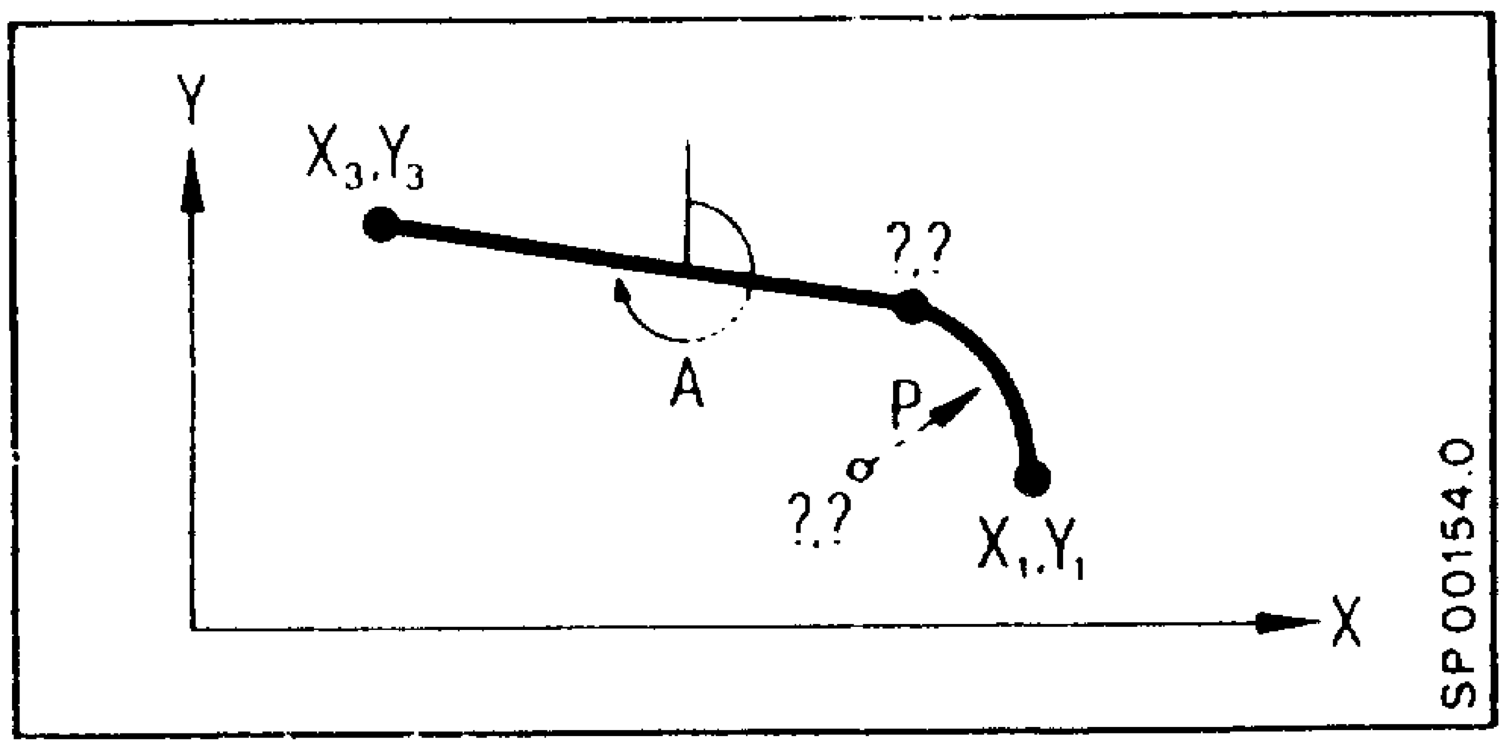


N..G02 (or G03) A..P..X_3..Y_3..

Circular arc below 180°.
The sequence A (angle) and P (radius) must be kept.

* Second block may also be a contour cycle.

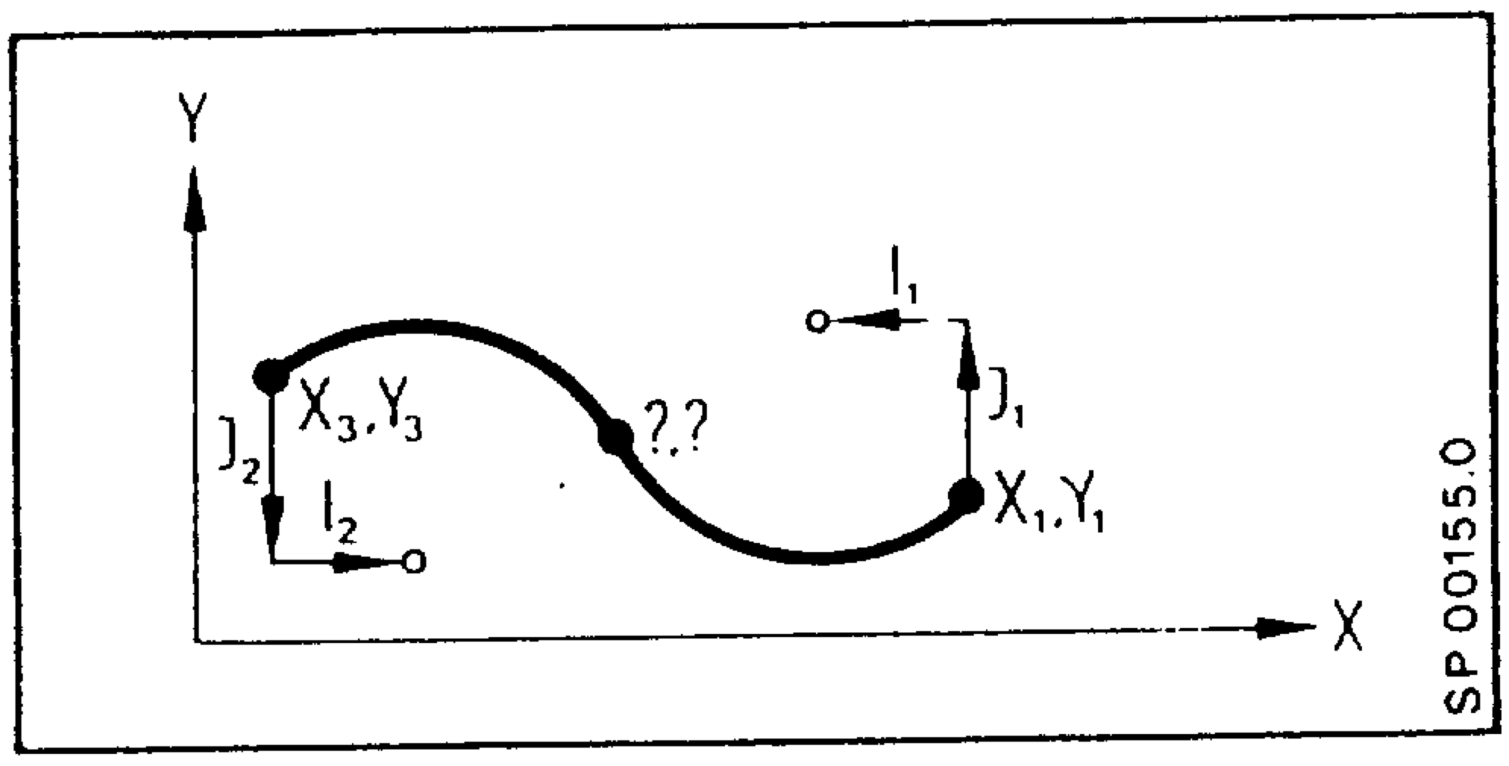
7. Circular arc - straight line (tangential)



N..G02 (or G03) P..A..X_3..Y_3..

Circular arc below 180°. The sequence P, A must be kept. No radius in X_3, Y_3.

8. Circular arc - circular arc (tangential)

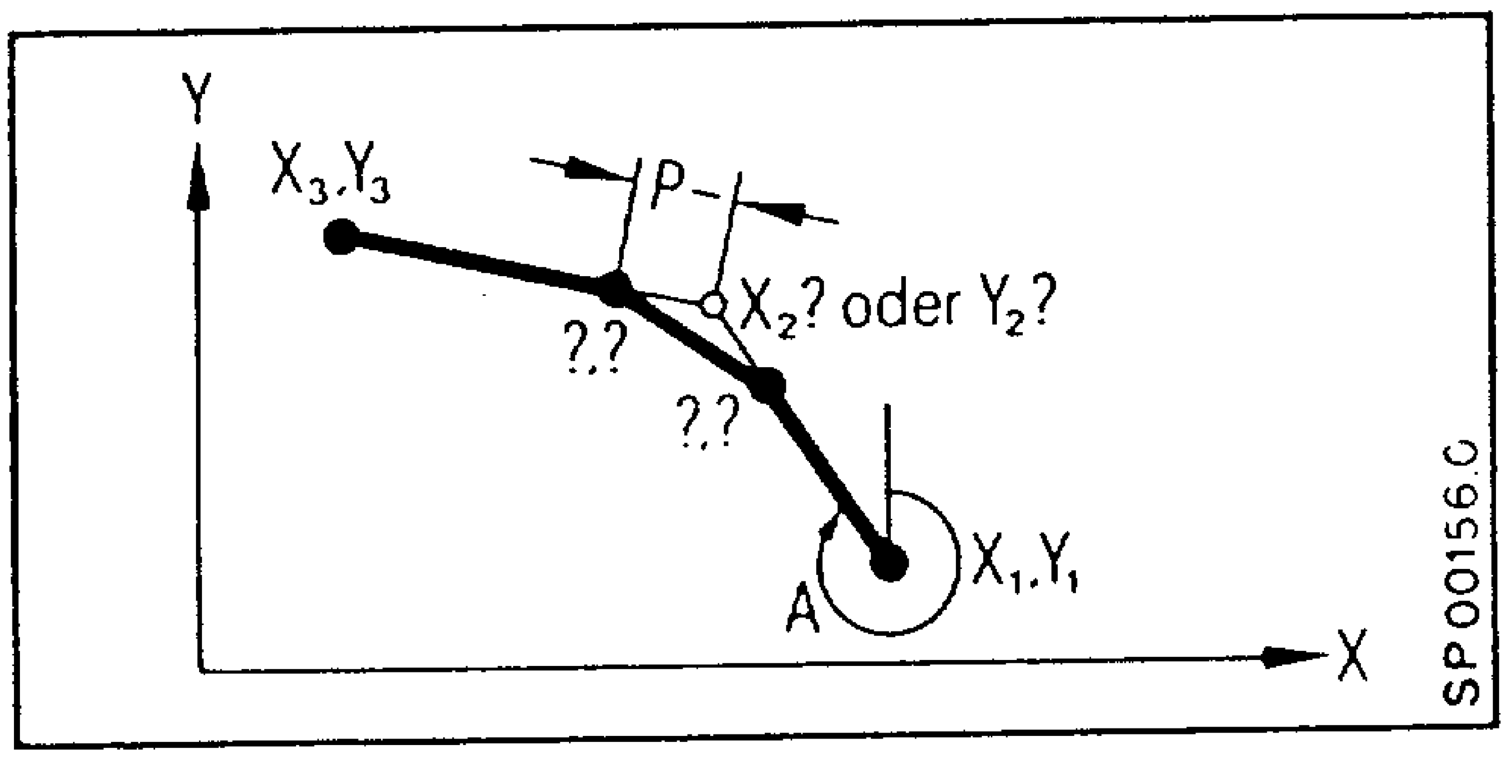


N..G02 (or G03) I_1..J_1..I_2..J_2 ..X_3..Y_3

circle 1

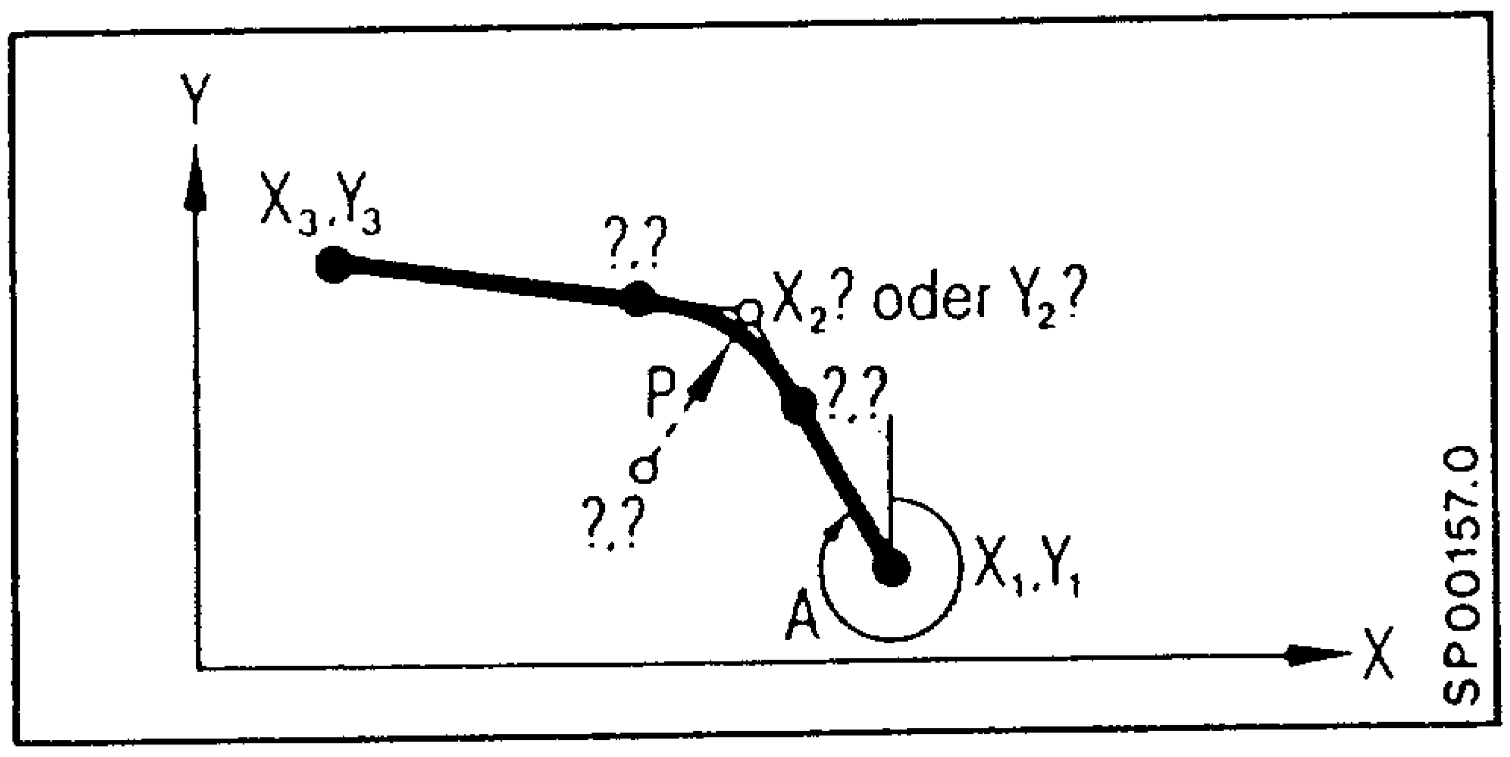
The preparatory function is always programmed just for the first circular arc. The second preparatory function is contrary to the first one.

(1.) + (4.) Two-point cycle + chamfer



N.. A..X_2... (or Y_2..)
P...N.. X_3..Y_3... *

(1.) + (5.) Two-point cycle + radius

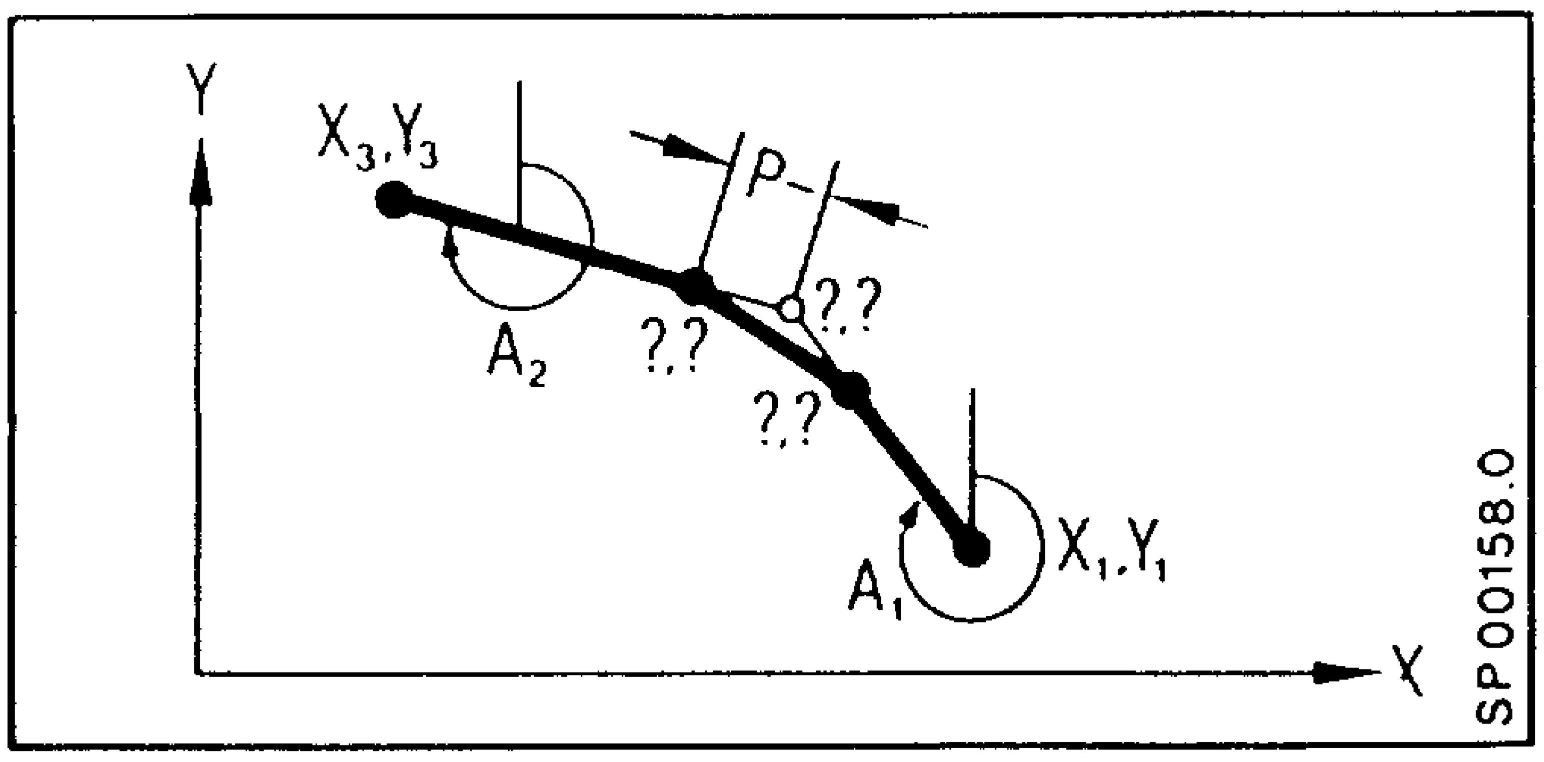


N.. A..X_2... (or Y_2...)
P...N.. X_3..Y_3...

The radius inserted must be smaller than the shorter one of the two movements.

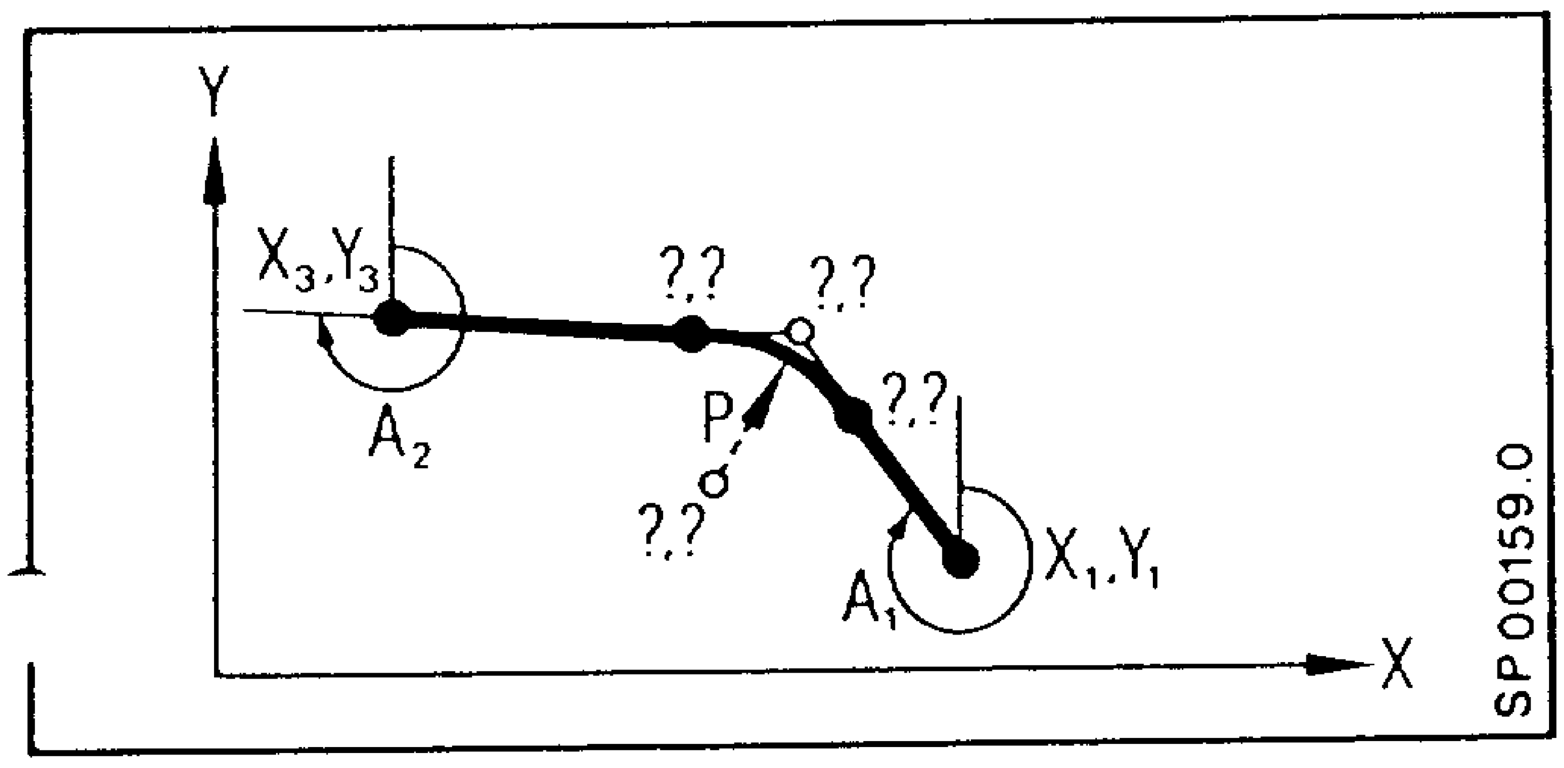
* Second block may be also a contour cycle.

(3.) + (4.) Three-point cycle + chamfer



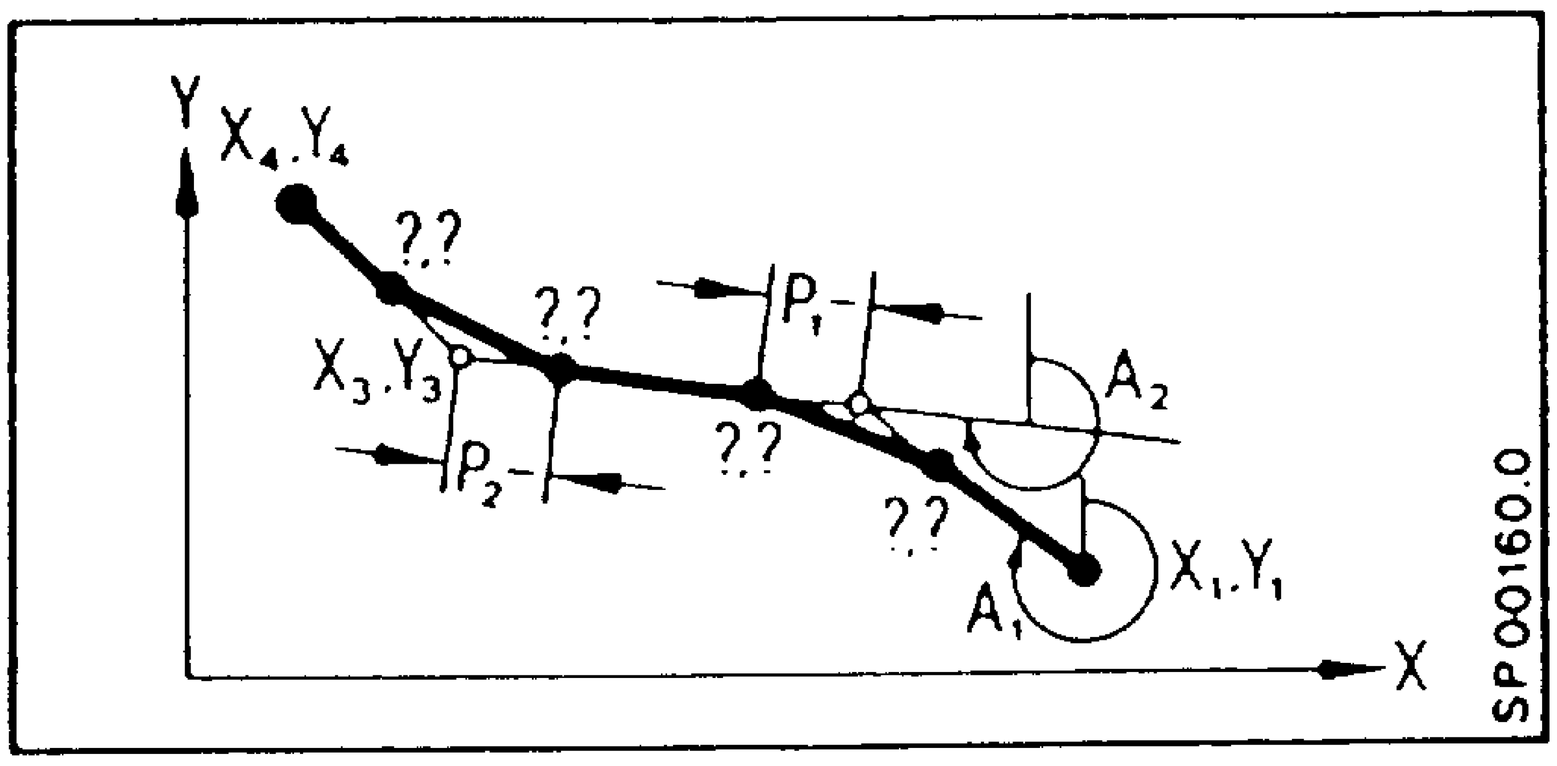
N.. A_1..A_2..X_3..Y_3..P-..

(3.) + (5.) Three-point cycle + radius



N.. A_1..A_2..X_3..Y_3..P...

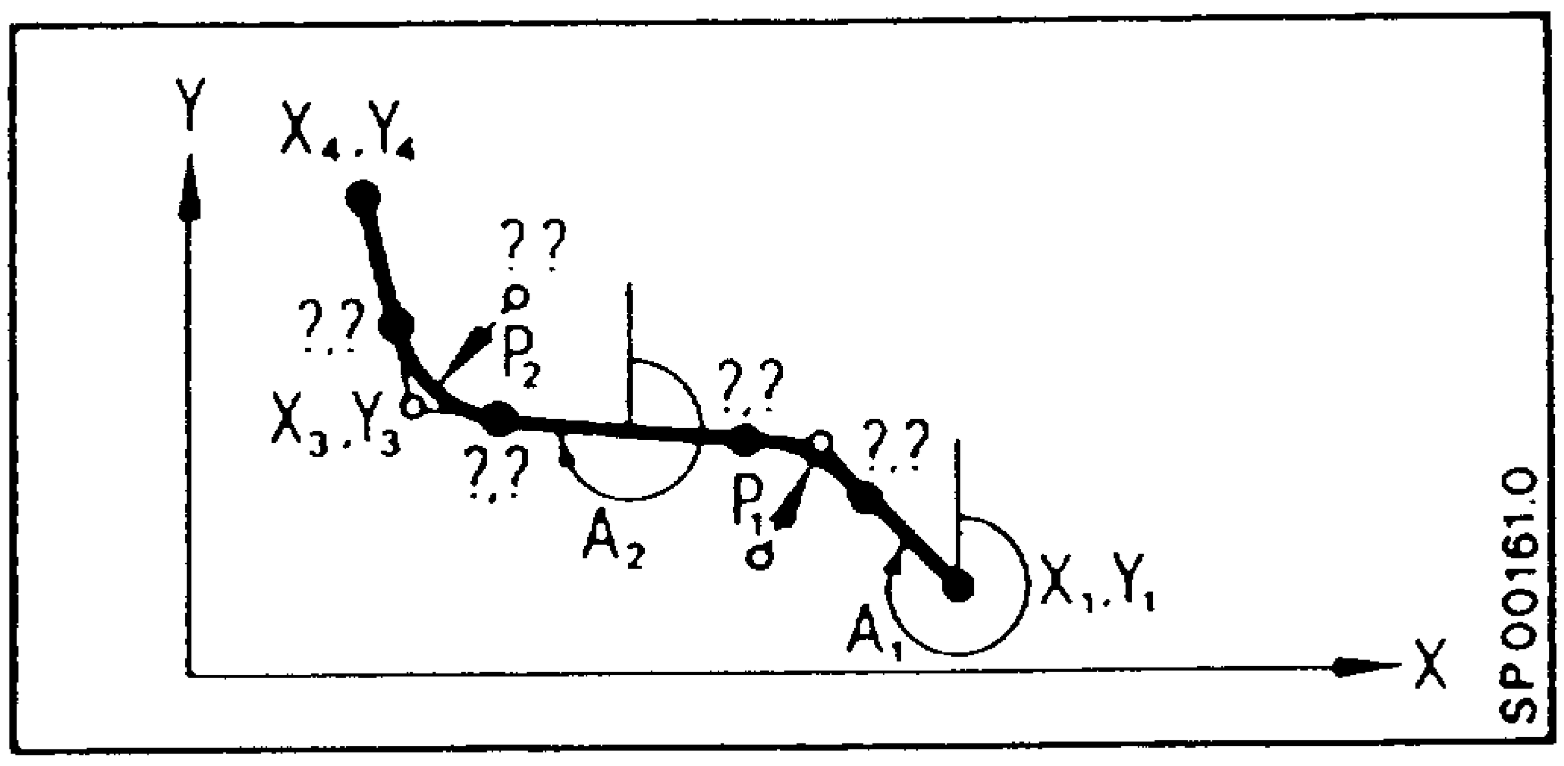
(3.) + (4.) + (5.) Three-point cycle + chamfer + chamfer



N15 A_1..A_2..X_3..Y_3.. P-..
P_2-..
N16 X_4.. Y_4.. *

Insertion of a second chamfer at the end position (X_3, Y_3).

(3.) + (5.) + (5.) Three-point cycle + radius + radius

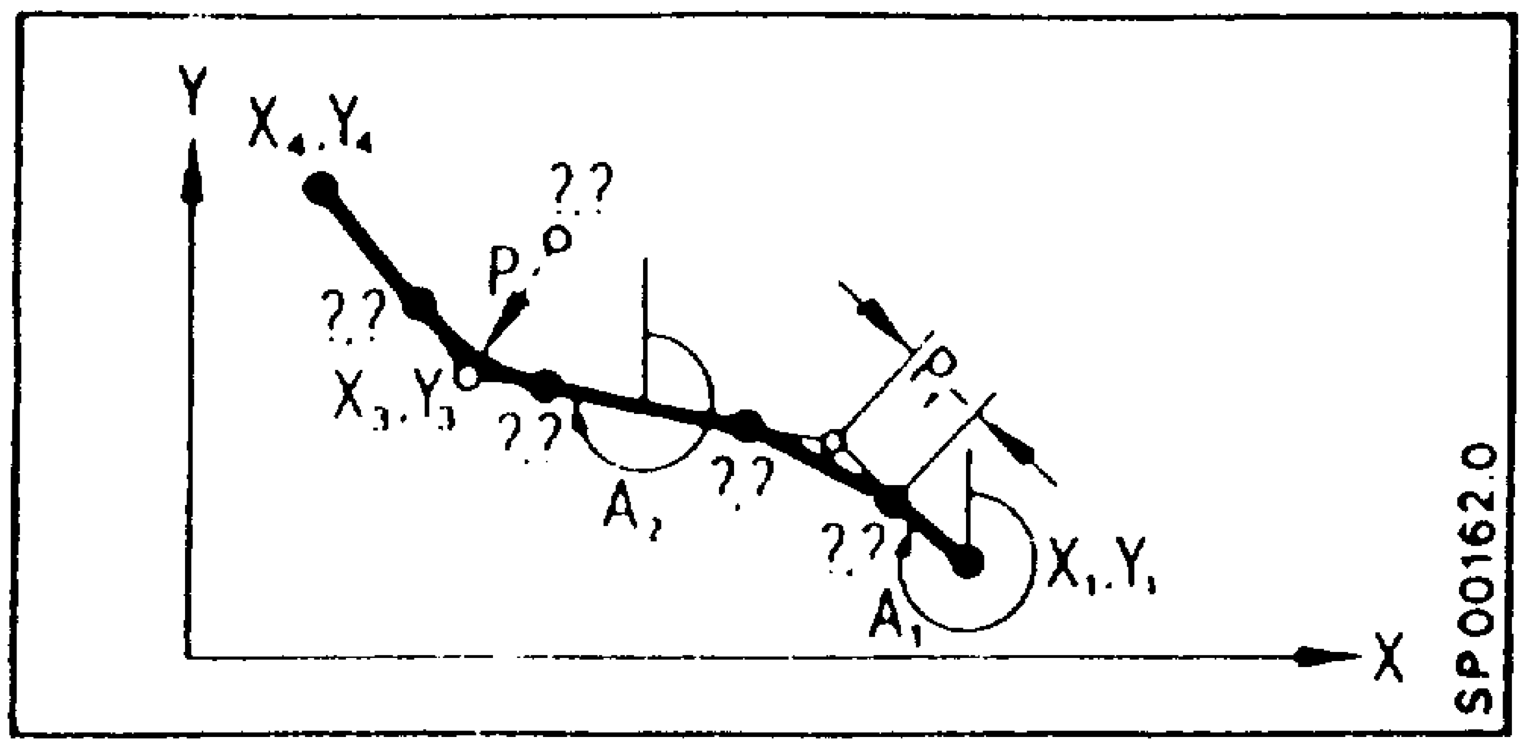


N15 A_1.. A_2.. X_3.. Y_3.. P..
P_2..
N16 X_4.. Y_4.. *

Insertion of a second radius at the end position X_3, Y_3.

* Second block may be also a contour cycle.

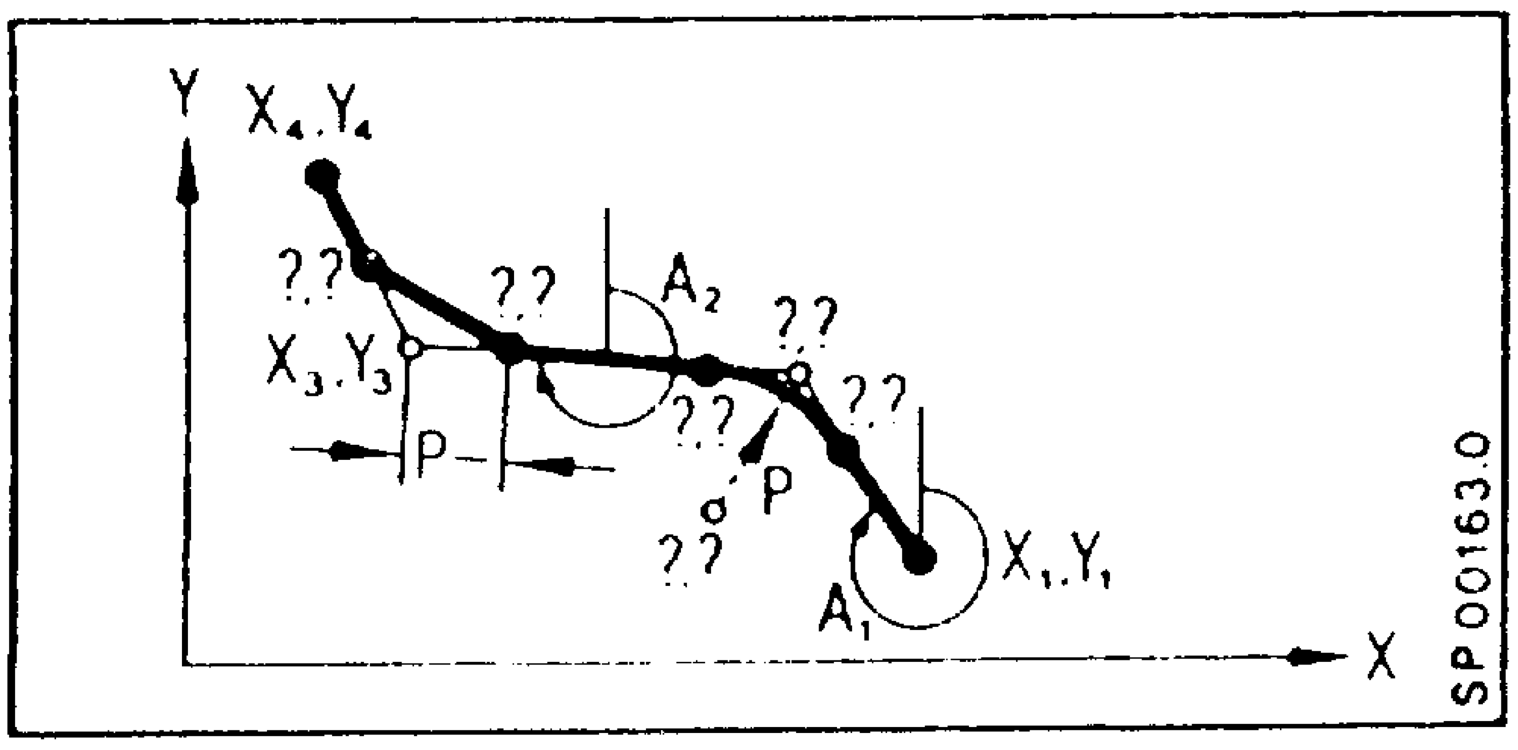
(3.) + (4.) + (5.) Three-point cycle + chamfer + radius



N.. A_1..A_2..X_3..Y_3..P-..P_2
..
N16 X_4..Y_4

Insertion of a radius at the end position X_3, Y_3.
The following block is automatically taken into account.

(3.) + (5.) + (4.) Three-point cycle + radius + chamfer



N15 A_1..A_2..X_3..Y_3..P..P-..

N16 X_4..Y_4..

Insertion of a chamfer (P-) at the end position.

* Second block may be also a contour cycle.

Corners without chamfer or radius must be programmed with 0 if there is another radius or chamfer in the contour cycle.

Caution:

In this programming, the control generates a block with the path = 0. This must be particularly observed if the cutter radius compensation is active.

B-0 is interpreted as B 0.
A radius or a chamfer may be inserted if the following block is no circular block.
The sequence of the addresses A, X, Y, P, F etc. can be freely selected, however, angles and radii must exist in the before-mentioned sequence (first angle before second angle, first radius before second radius in machining direction).

6.2.2 Operation of the function F, S, T, H, M in contour cycles

Within a contour cycle, the control automatically generates G60 at corners, edges etc. Here, the automatically inserted G60 function does not mean a reset of group 10.

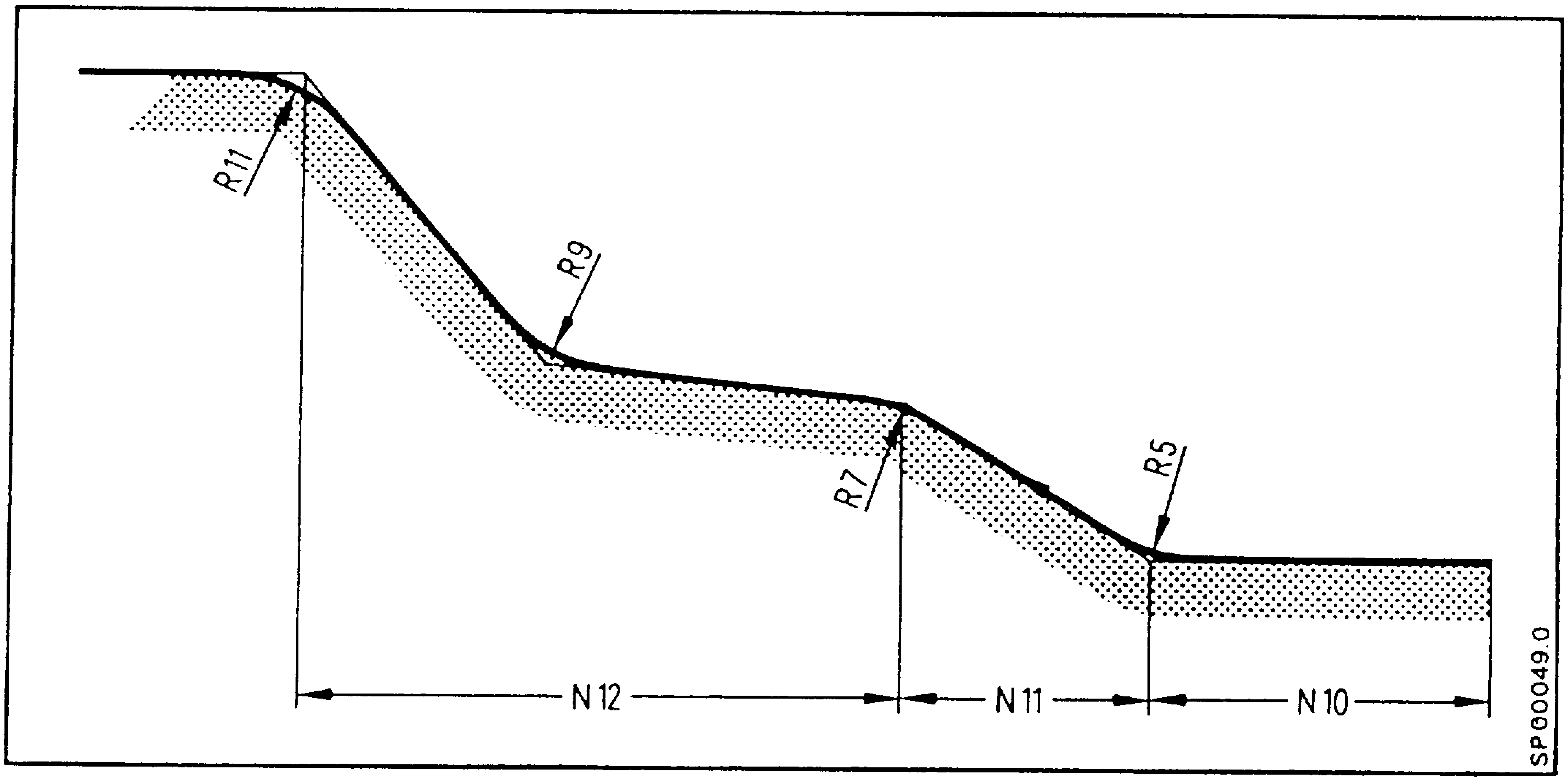
If F, S, T, H, M are programmed in a contour cycle, they are effective at block start.

If M00, M02, M17, M30 are programmed in a contour cycle, they are effective at block end.

6.2.3 Chaining of Blocks

The chaining of blocks with and without angular input as well as with inserted radii or chamfer can be made in arbitrary sequence.

Example:

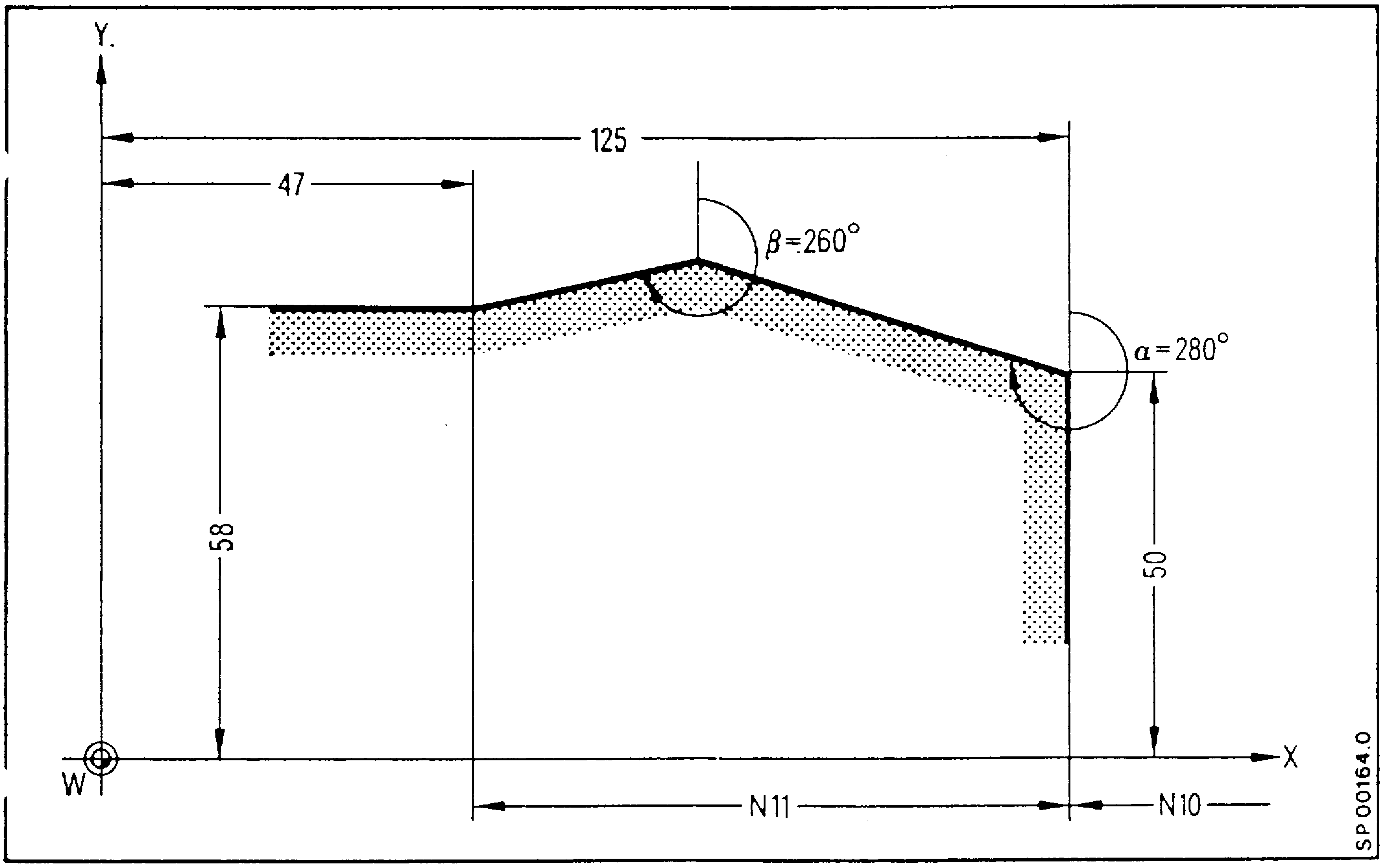


N10	Z...	P5	LF			
N11	A...	X...	P7.	LF		
N12	A...	A...	X...	Y...	P9.	P11. LF

6.2.4 Examples

The angle a refers to the startingposition, the angle ß applies to the missing support point.
The end position can be programmed in absolute dimensions (G90) or in incremental dimensions (G91). Both end point coordinates must be started. The control then computes the support point using the starting position, the two angles and the end position.

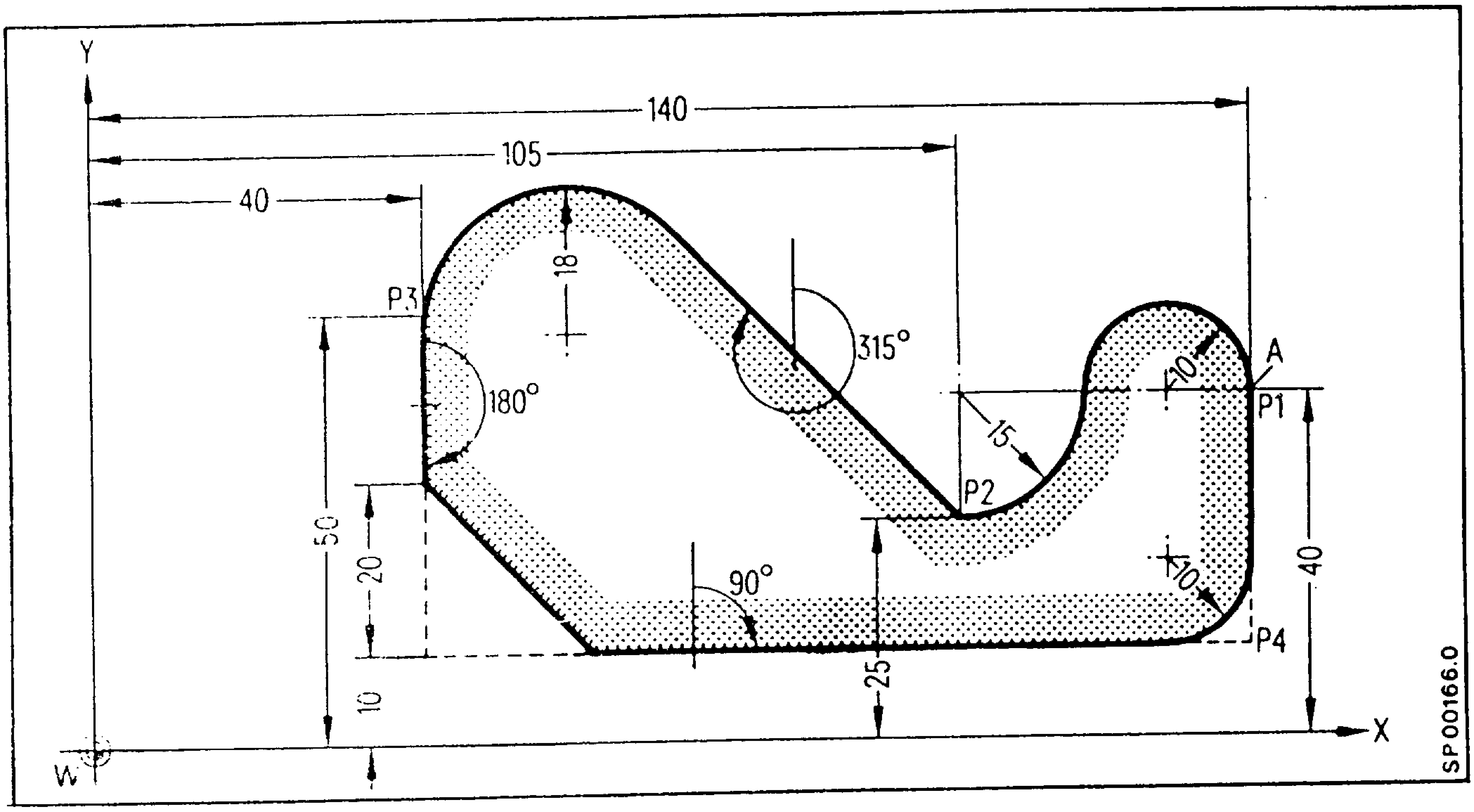
Example:




```
:
N10   G00   G90    X125.  Y50.               LF
N11   G01   A280.  A260.  X47.   Y58.   F... LF
```

Example: Contour cycle programming

In this example, the following contour cycles are used: Circular arc - circular arc, straight line - circular arc and a three-point cycle + chamfer + radius.

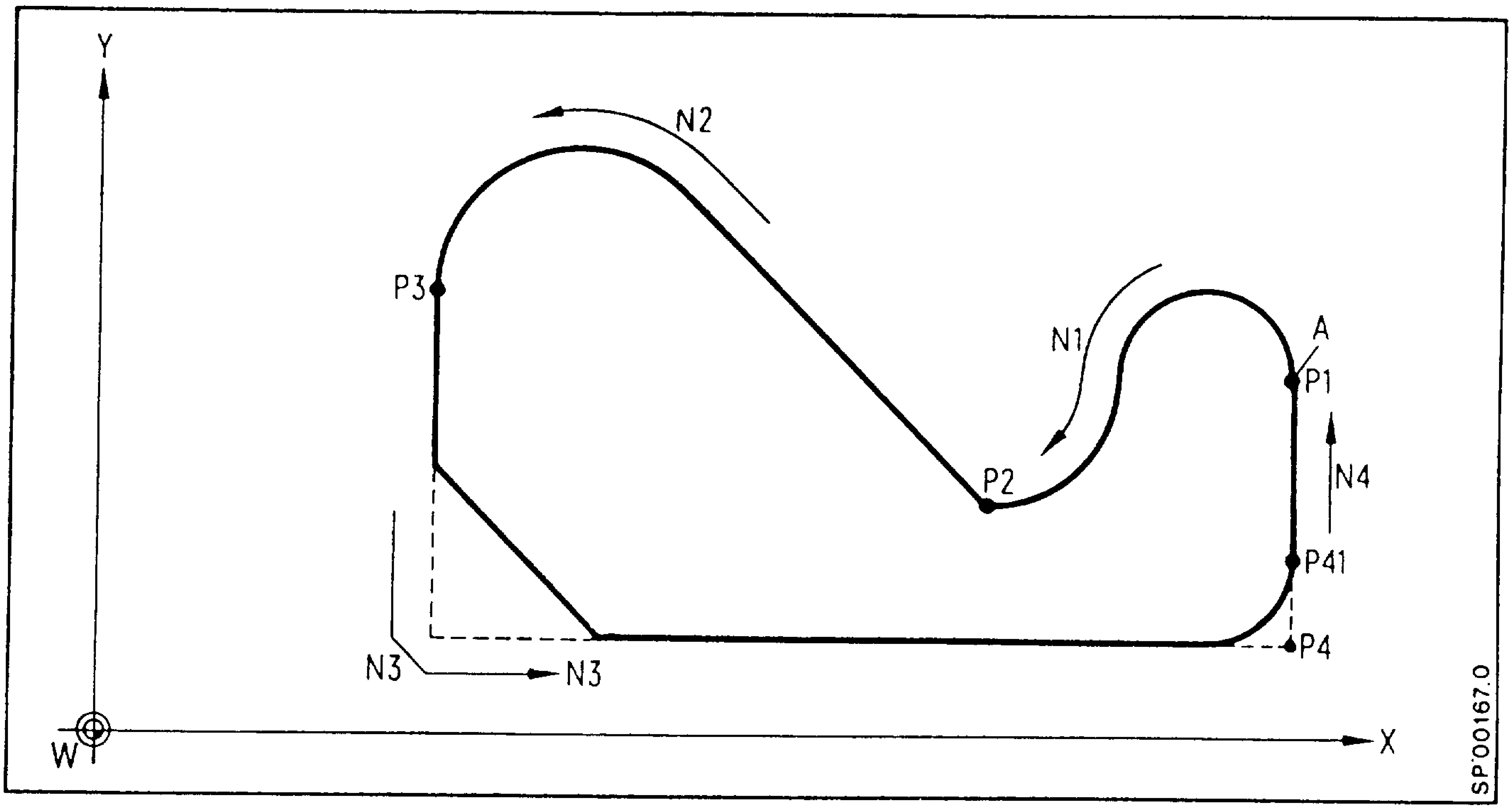



```
L16800
N1 G90 G03 I-10. J0. I0. J15. X105. Y25. LF (P2) Circular arc
                                                  - circular arc
N2     G03 A315. P18. X40. Y50. LF           (P3) Straight line
                                                  - circular arc
N3     G01 A180.A90.X140.Y10.P-20.P10.  LF   (P4) three-point cycle
                                                  + chamfer + radius
N4                Y40.   LF                  (P1) straight line
N5 M17 LF
```

In block N2 G03 must be programmed since the second circular direction in the transition: circular arc - circular arc is opposed to the direction of the first circle (contour cycle 8).

6.2.5 Miscellaneous and Auxiliary Functions in Chained Blocks

Chained blocks are all blocks which share transitions with radii or chamfers.



Between chained blocks, it is possible to input a block with miscellaneous and auxiliary functions.
Example: see above and page 6 - 10.

```
N3  A180. A90. X140. Y10. P-20. P10 LF  (contour cycle P3-P4)
N4  M... H...   ....   LF
N50 Y40. LF
```

The auxiliary functions are activated in point P4 (see above). Accordingly, release cutting is done at point P41.

7. Cycles

7.1 Drilling/Boring Cycles G81 - G89

A boring cycle (working cycle) defines in accordance with DIN 66025) a series of machine movements necessary for drilling, boring, tapping, etc..
The boring cycles G81 to G89 are stored in the control as subroutines L81 to L89.

Canned cycle No.	Canned cycle Subroutine	Traverse rate into the part after positioning to the reference plane	At hole bottom Dwell	At hole bottom Spindle	Retract to the reference plane	Application example
0	L800	-	-	-	-	Cancels L81-L89
1	L8100	in feed	-	-	in rapid trav.	Drilling, centering
2	L8200	in feed	yes	-	in rapid trav.	Drilling, counter-sinking
3	L8300	in interrupted feed	-	-	in rapid trav.	Deep hole drilling, chip breaking
4	L8400	feed per revolution	-	rev.	in rapid trav.	Tapping
5	L8500	in feed	-	-	in feed	Boring 1
6	L8600	Spindle on, in feed	-	stop	in rapid trav.	Boring 2
7	L8700	Spindle on, in feed	-	stop	manual retr.	Boring 3
8	L8800	Spindle on, in feed	yes	stop	manual retr.	Boring 4
9	L8900	in feed	yes	-	in feed	Boring 5

The user may deviate from a standard fixed cycle and re-define it to suit his specific machine or tooling requirements. The parameters R00 to R11 are used by the subroutine to define the variable values necessary to correctly execute a fixed cycle (e.g. reference plane coordinates, the hole depth, feed rate, dwell time, etc.). Prior to a subroutine call, all necessary parameters must be defined in the main program.
A fixed cycle call is initiated with G80 to G89. G81 to G89 are modal fixed cycles that are cancelled with G80. A boring cycle can be called with L81-L89, however, L81 to L89 are not modal. L81-L89 is performed only once in the block in which it is programmed. At the end of a fixed cycle the tool is repositioned at the starting point.

The following R parameters are used in cycles L81 - L89

R00 Dwell time at the start point (deburr hole)
R01 First depth advance (incremental) entered without sign
R02 Reference plane - retract plane (absolute)
R03 Final depth
R04 Dwell time at hole bottom (break chips)
R05 Depth advance modifier entered without sign
R06 Reverse spindle rotation direction
R07 Return to the original spindle rotation direction used in the calling program (after R06 or M05)

R09 Depth advance or thread lead modifier.
R10 Retract plane
R11 Boring axis (axis numbers X=1; Y=2; Z=3) 1)

The parameters are changed at the end of the cycle. The cutter must be positioned to the correct location in the plane perpendicular to the Z axis or to R11[1)] before calling. The appropriate feed, spindle speed, and rotation direction must be programmed for absolute dimensions. After a return from a fixed cycle incremental dimensions must be programmed again in the calling program.

Subroutine L80: (cancels G81 - G89)

G80 is an internal control function call. No parameter definitions are required.

1) with basic controls 3 and 4

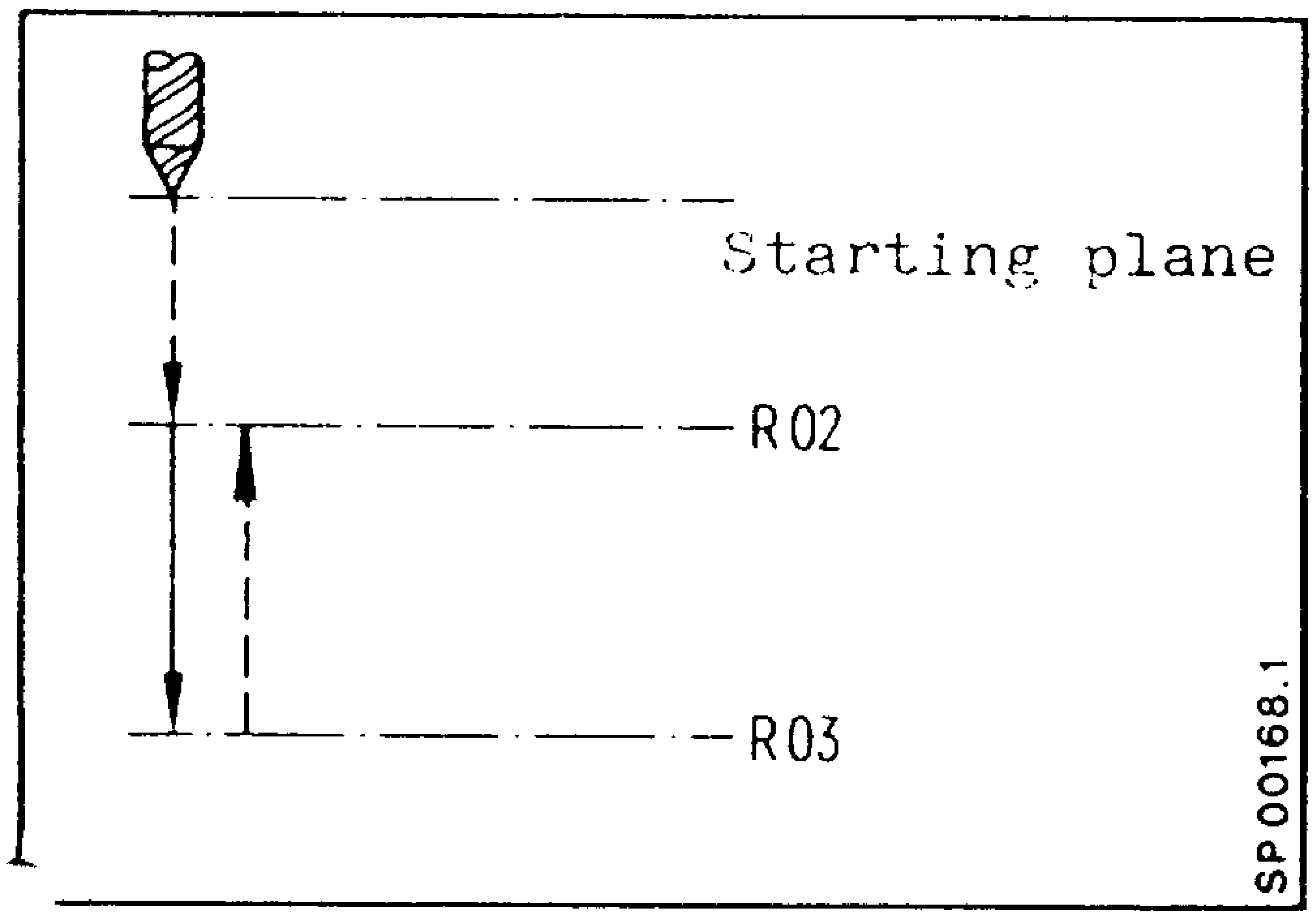
Subroutine L81: (Drilling, centering)

The following parameters must be defined:

R02 Reference plane (retract plane).

R03 Final hole depth

R11 Boring axis



Subroutine L82: (Drilling, counter sinking)

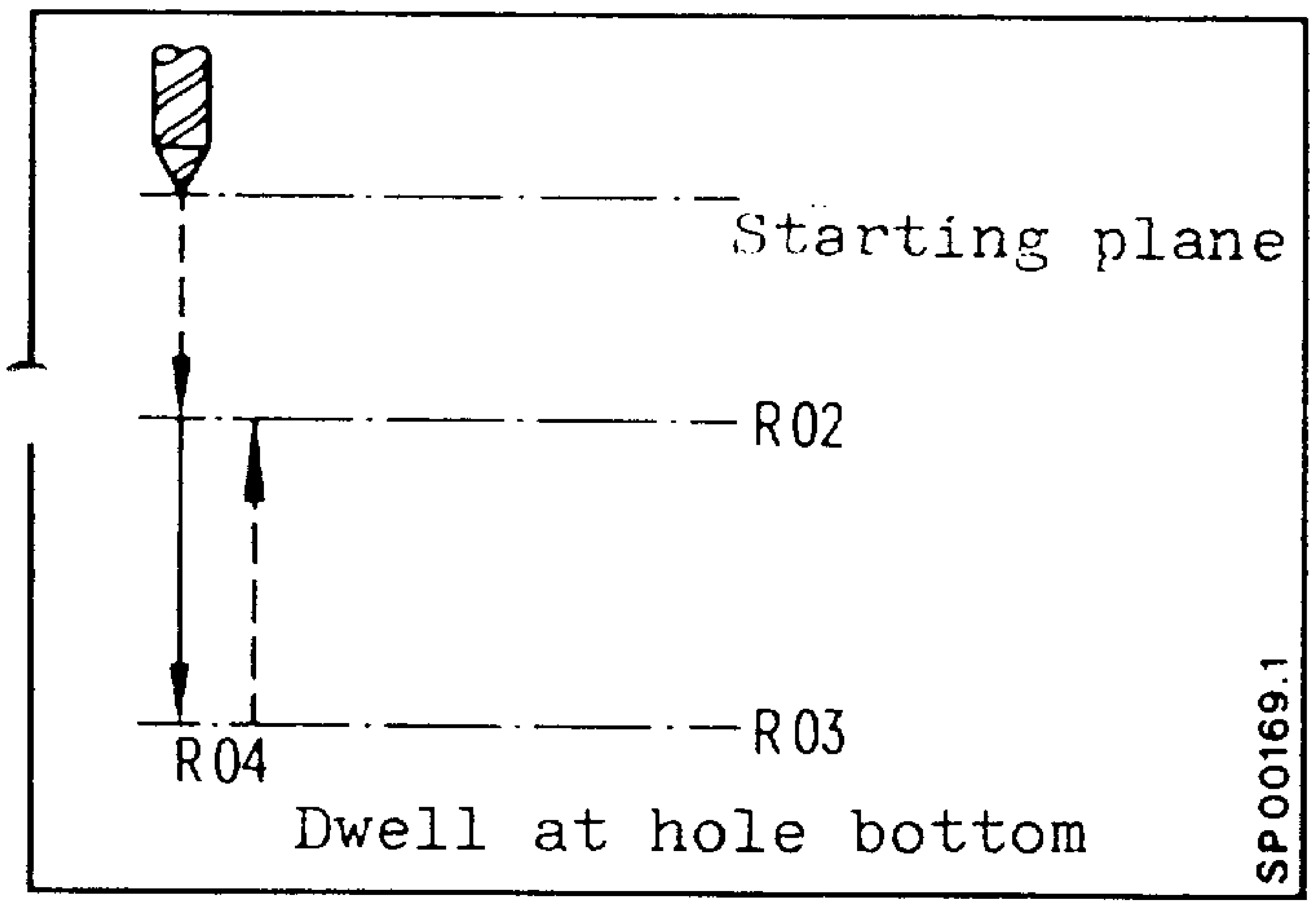
The following parameters must be defined:

R02 Reference plane (retract plane)

R03 Final hole depth

R04 Dwell time

R11 Boring axis



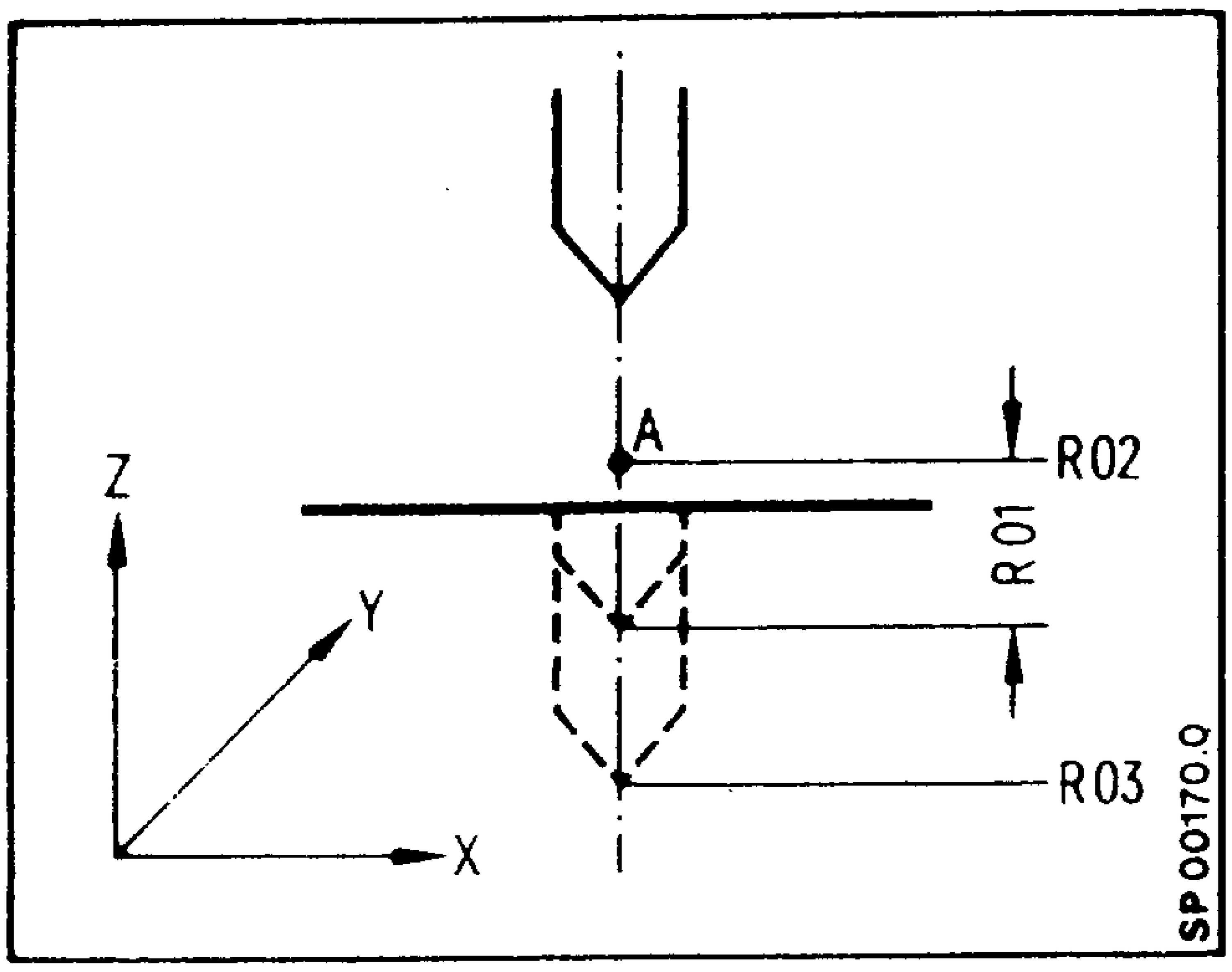
------- Rapid traverse

——— Feedrate

Subroutine L83: (Deep hole drilling)

The following R parameters must be defined prior to calling canned cycle L83:

R00 Dwell at the start position (to deburr hole)
R01 First depth advance (incremental) entered without sign
R02 Reference plane = retract plane (absolute) "A"
R03 Final hole depth (absolute)
R04 Dwell time at hole bottom (break chips)
R05 Incremental depth advance modifier entered without sign
R11 Boring axis



R03 Final hole depth:

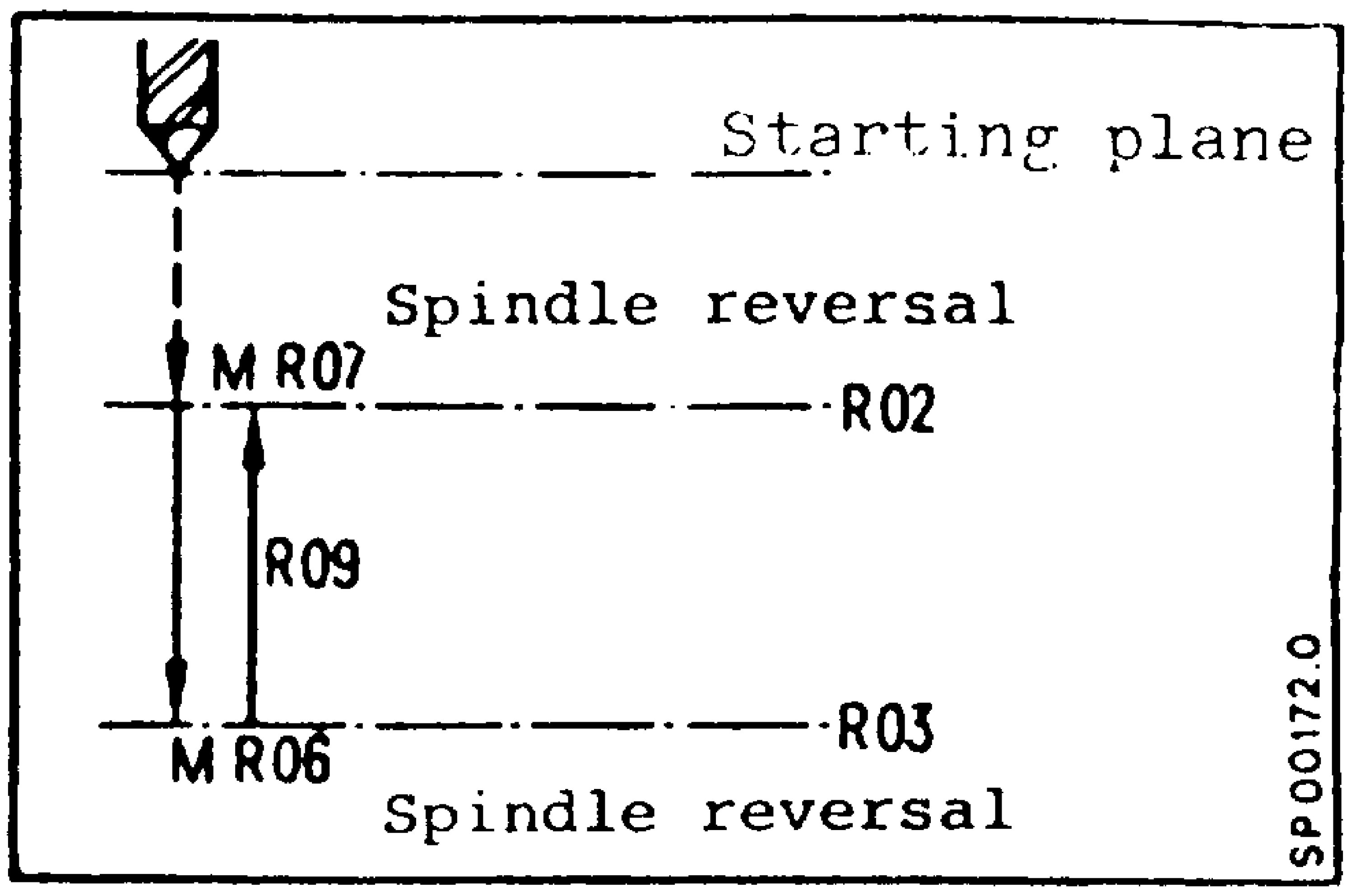
The incremental depth **diminishes with each successive drill** amount till the final hole depth R03 is reached. If the incremental depth advance modifier exceeds the actual drill advance, succeeding drill advances will be held constant. At the end of the drilling cycle the drill is brought to starting point A.

If the remaining depth is greater than R05 and less than 2 times R05, it is divided into 2 drilling strokes.

$$R05 < a < 2\ R05$$

a = remaining depth

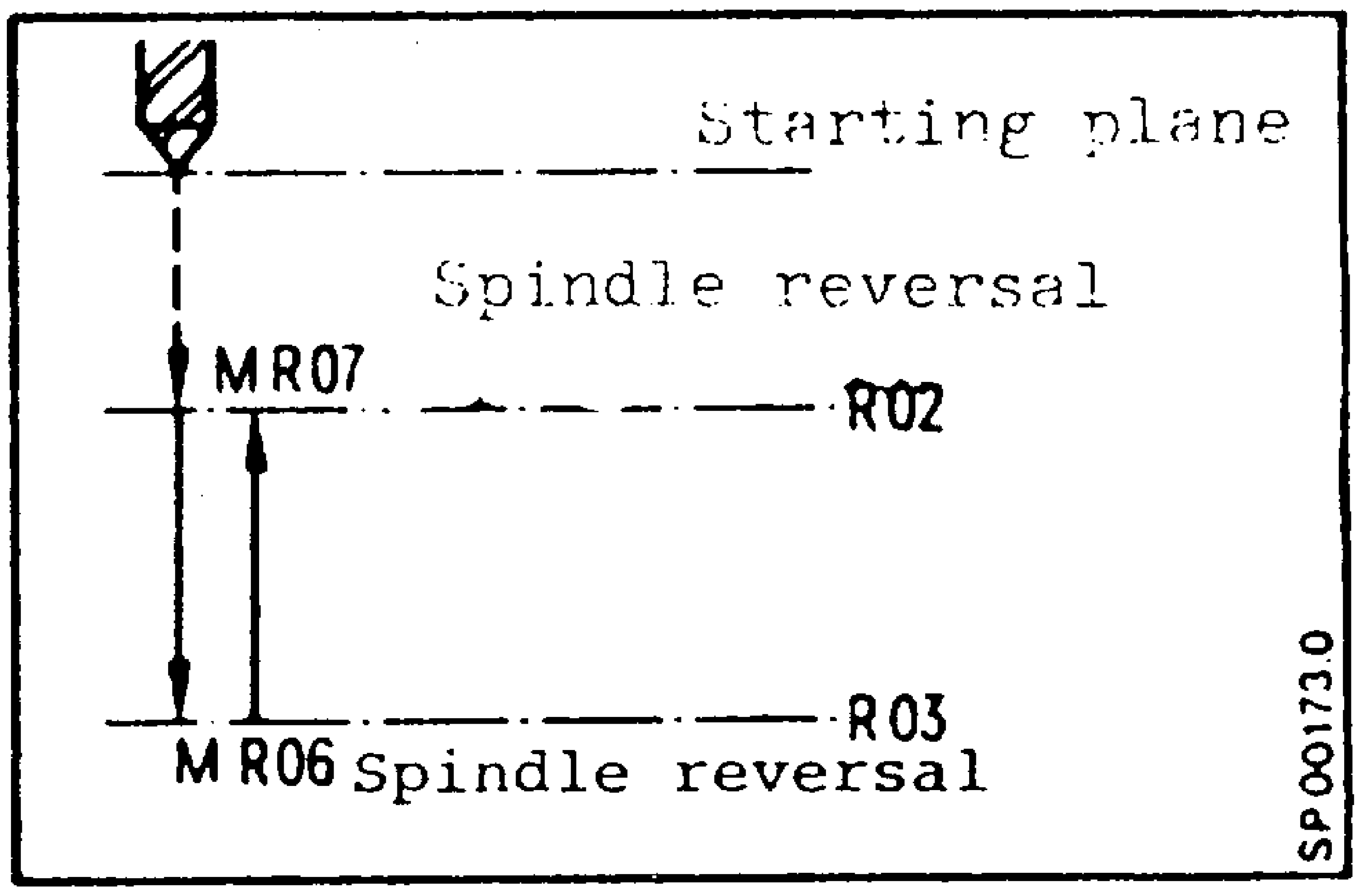
Subroutine L84:
(Tapping with spindle encoder)



The following R parameters must be defined:

R02 Reference plane (retract plane)
R03 Final depth
R06 Spindle rotation reversal
R07 Return to original spindle rotation direction.
R09 Thread lead dimension
R11 Boring axis

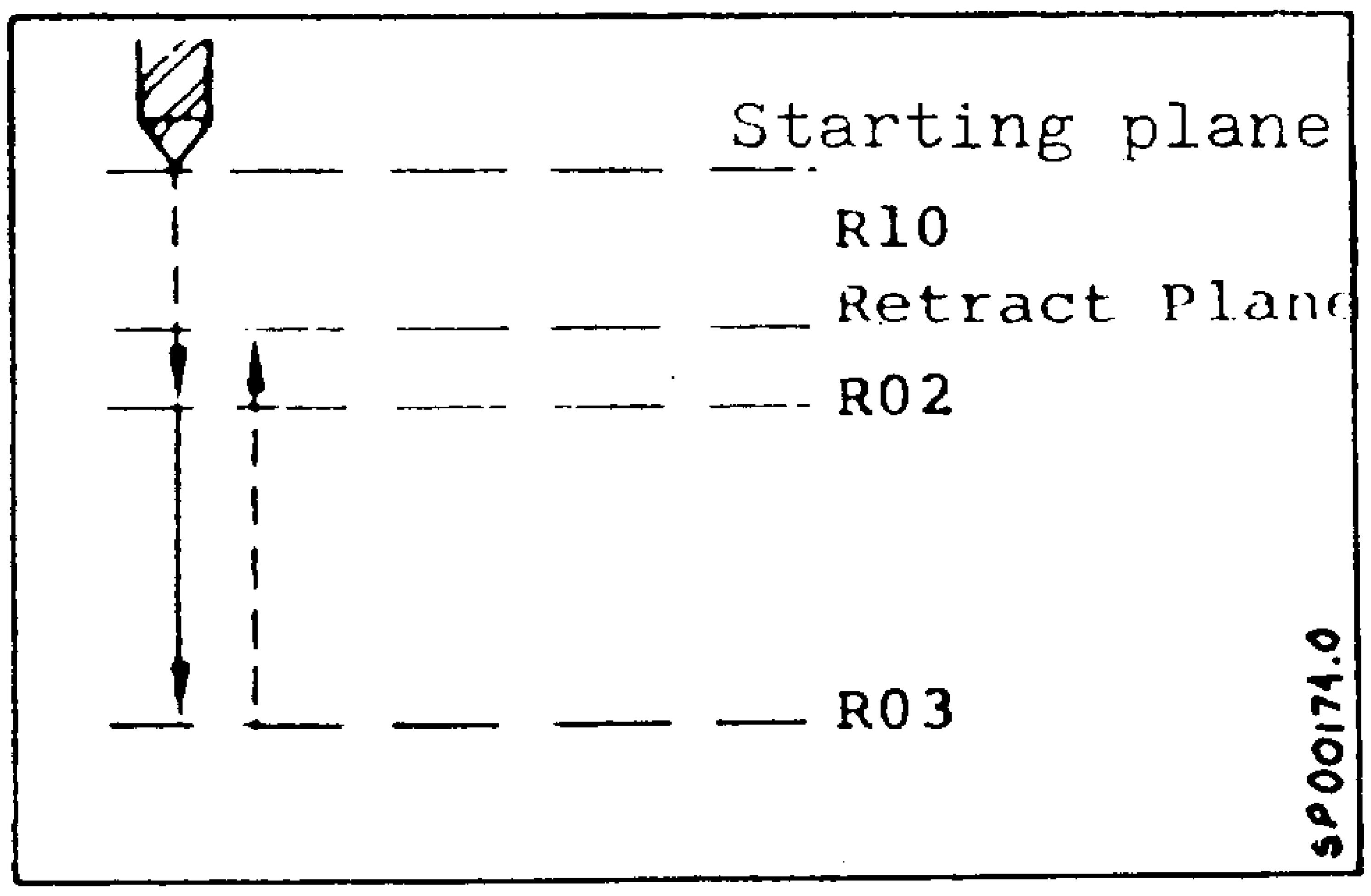
Subroutine L84:
(Tapping without spindle encoder)



The following R parameters must be defined:

R02 Reference plane (retract plane)
R03 Final depth
R06 Spindle rotation reversal
R07 Return to original spindle rotation direction.
R11 Boring axis

Subroutine L85:
(Boring 1)



The following R parameters must be defined:

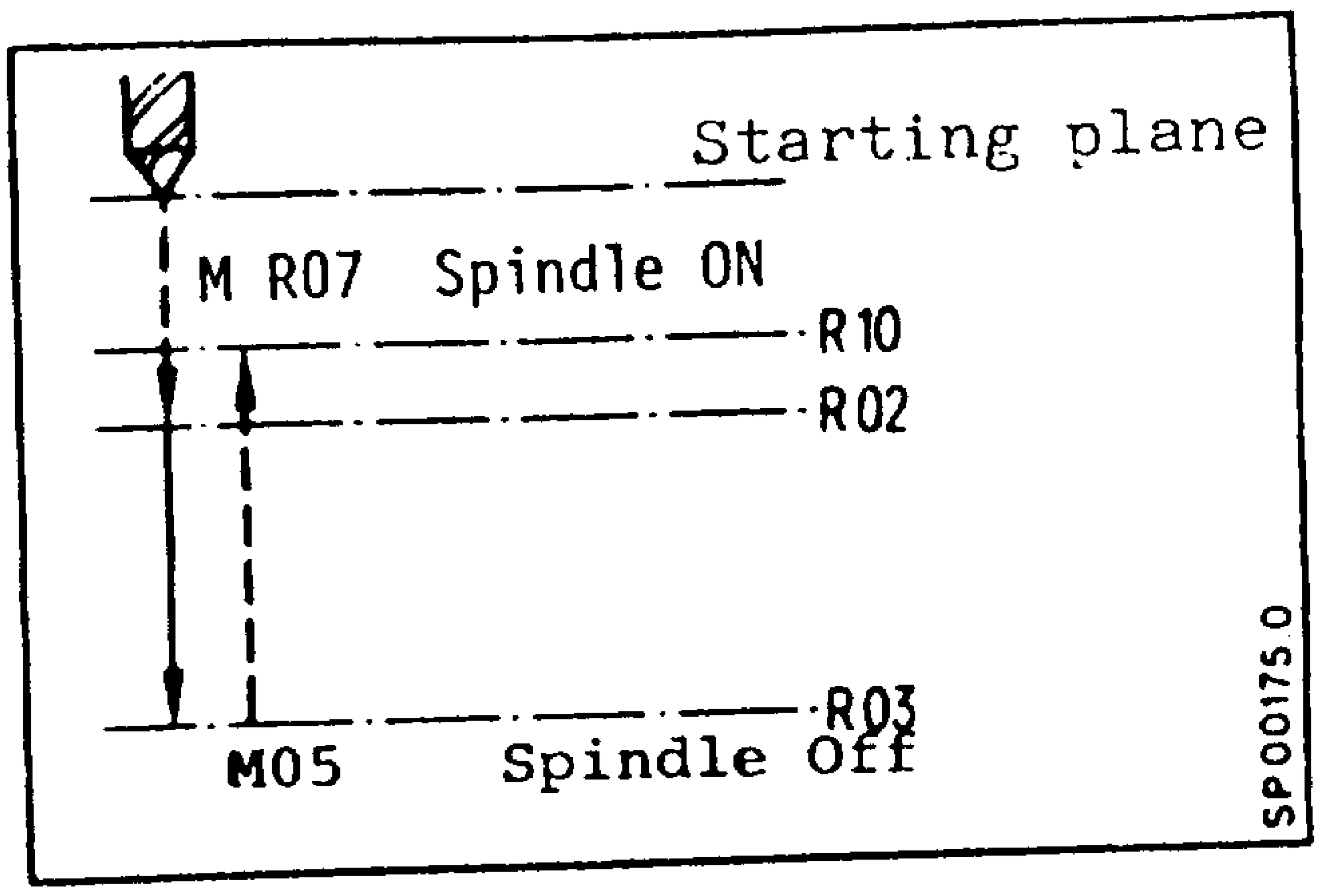
R02 Reference plane
R03 Final depth
R10 Retract plane
R11 Boring axis

----- Rapid traverse
—— Feedrate

Subroutine L86: (Boring 2)

The following R parameters must be defined:

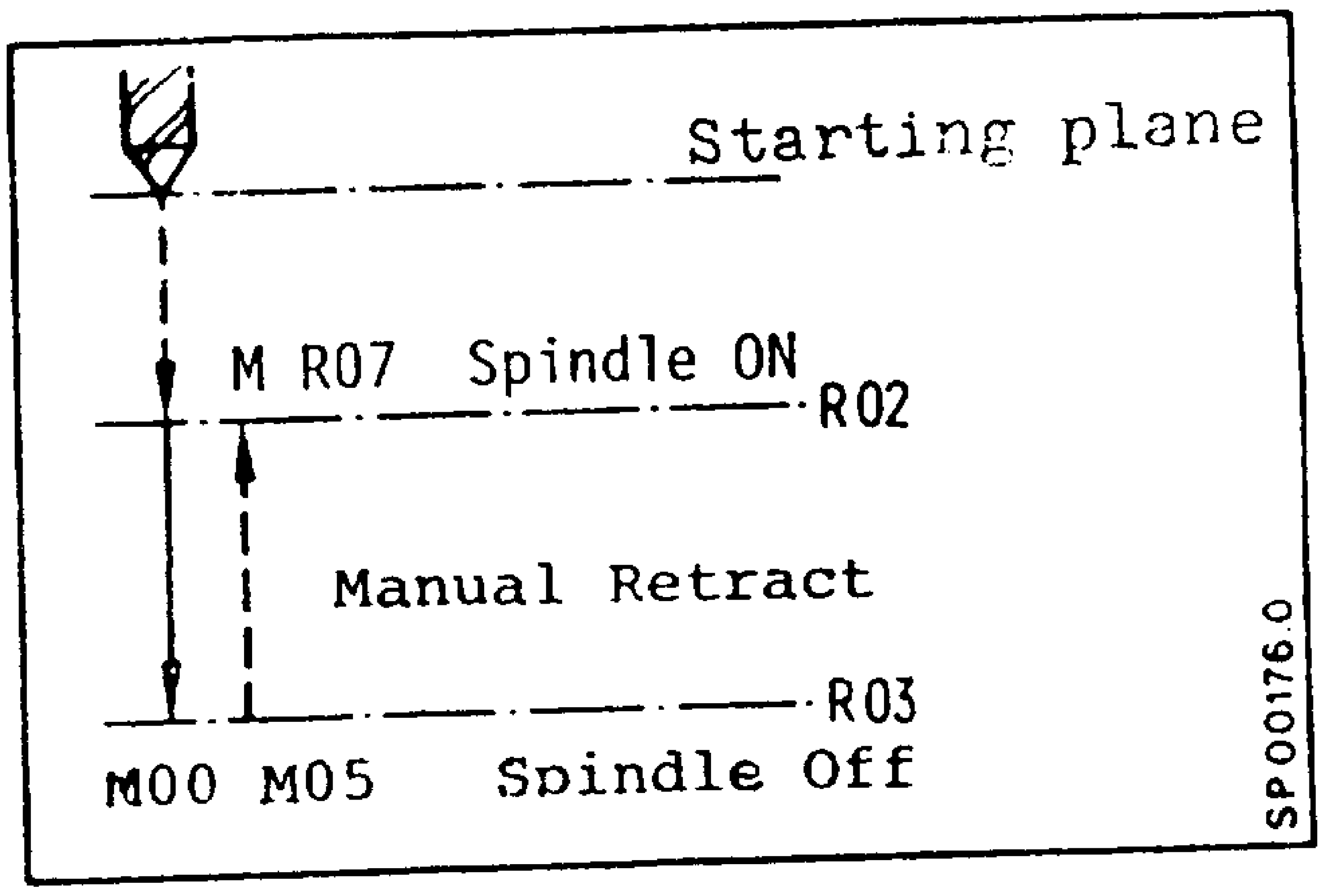
R02 Reference plane
R03 Final depth
R07 Spindle on (after M05)
R10 Retract plane
R11 Boring axis



Subroutine L87: (Boring 3)

The following R parameters must be defined:

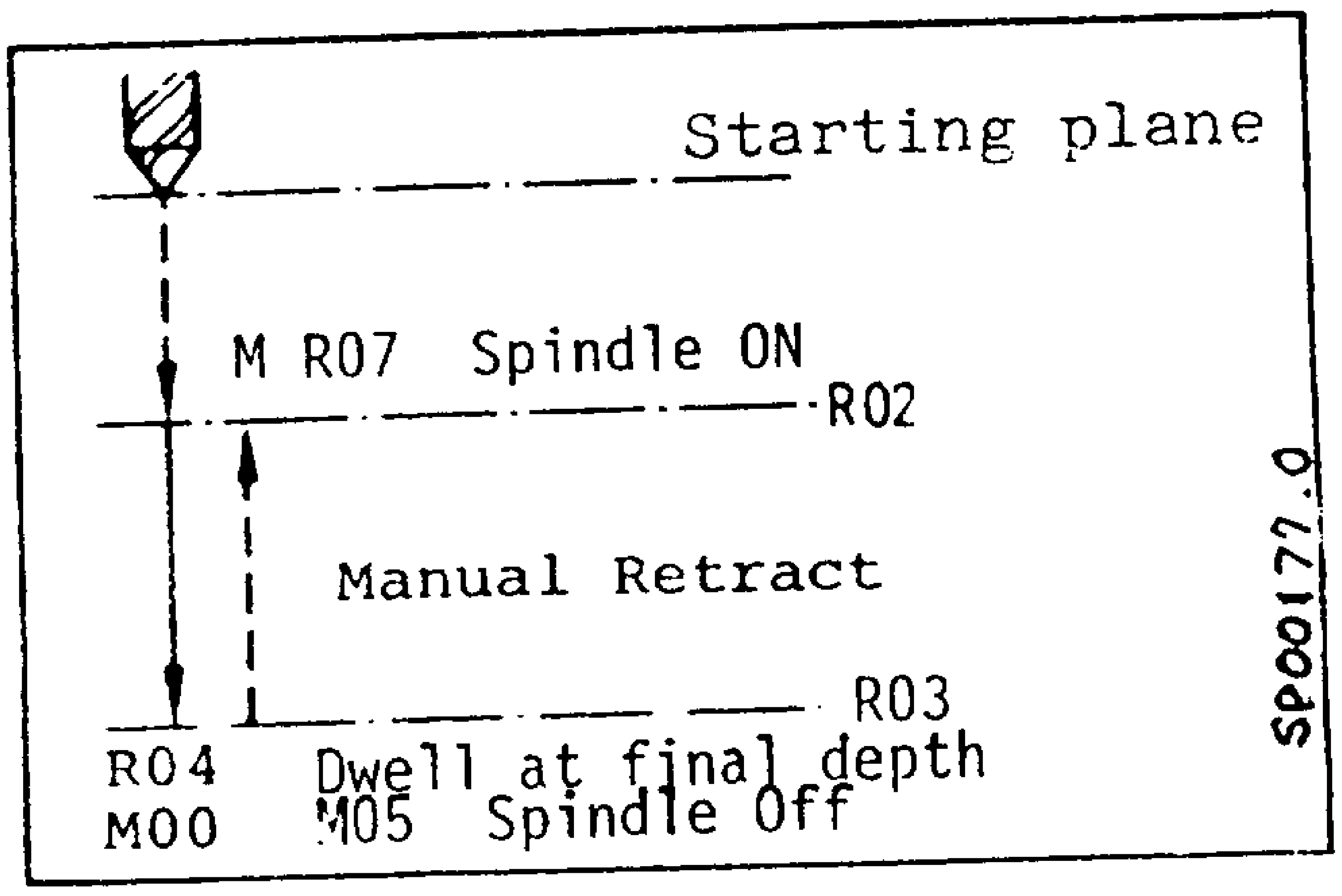
R02 Reference plane (retract plane)
R03 Final depth
R07 Spindle on (after M05)
R11 Boring axis



Subroutine L88: (Boring 4)

The following R parameters must be defined:

R02 Reference plane (retract plane)
R03 Final depth
R04 Dwell time
R07 Spindle on (after M05)
R11 Boring axis



Fixed cycle L88 (G88) is similar to L87 (G87), however, a dwell is performed at the bottom of the hole.

------- Rapid traverse
——— Feed rate

Subroutine L89: (Boring 5)

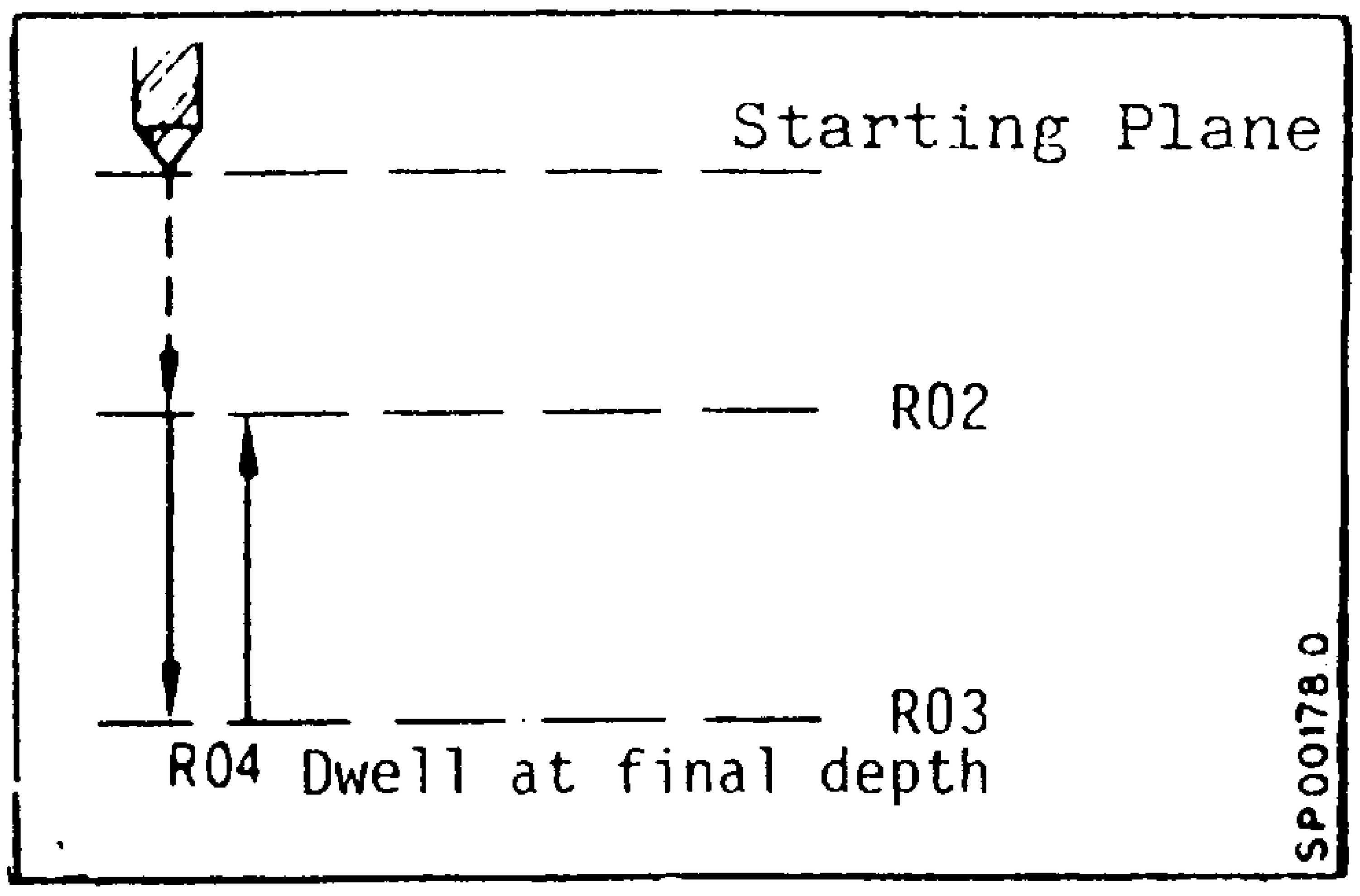
The following R parameters must be defined:

R02 Reference plane (retract position)

R03 Final depth

R04 Dwell time length

R11 Boring axis



----- Rapid traverse

——— Feedrate

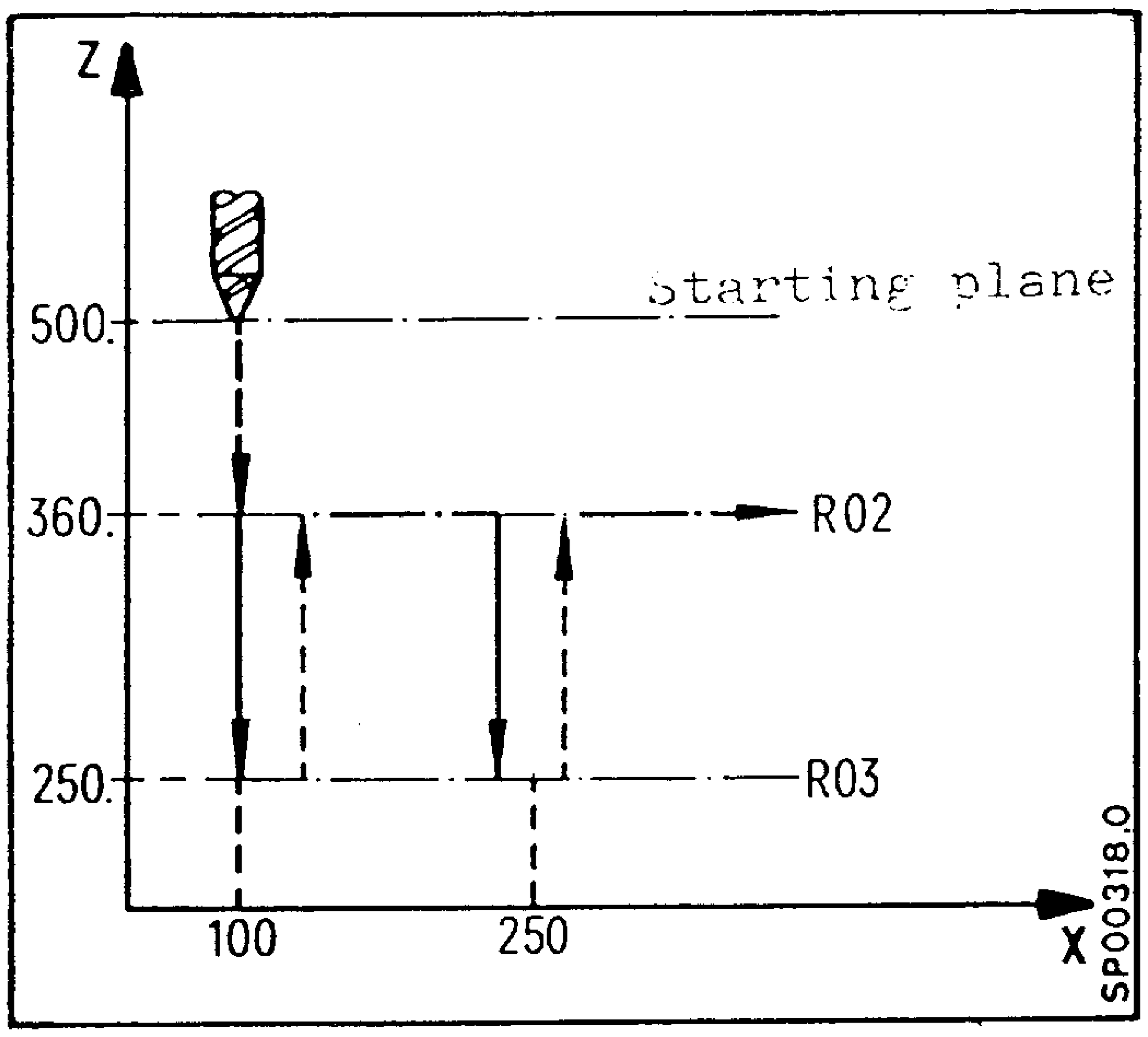
7.2 Examples of limitations in calling boring cycles

The drill cycle for every hole to be machined is called up only after the drill position has been reached.

The preparatory functions G81 to G89 can call up the subroutines L8100 to L8900 for one cycle pass. At every drill position a called up cycle is activated; these cycles can be cancelled with G80. If comments are required for the cycles G81 - G89, these comments have to be written in blocks with position data. If one of these comments is written alone between 2 LF characters, a boring cycle will also be executed.

Call-up G81 (Drilling, boring, centering, boring axis Z)

```
N8101   G90   S48   M03   F460                LF  - Spindle ON
N8102   G00   D01   Z500.                     LF  - Activate tool offset
N8103   X100.  Y150.                          LF  - 1st drill position
N8104   G81   R02   360.   R03 250.  R11 3.   LF  - Call cycle
N8105   X250.  Y300.                          LF  - 2nd drill position and
                                                    automatic G81 call
.
.
N8110   G80         Z500.                     LF  - Cancelling G81 and re-
                                                    turning to starting plane
```



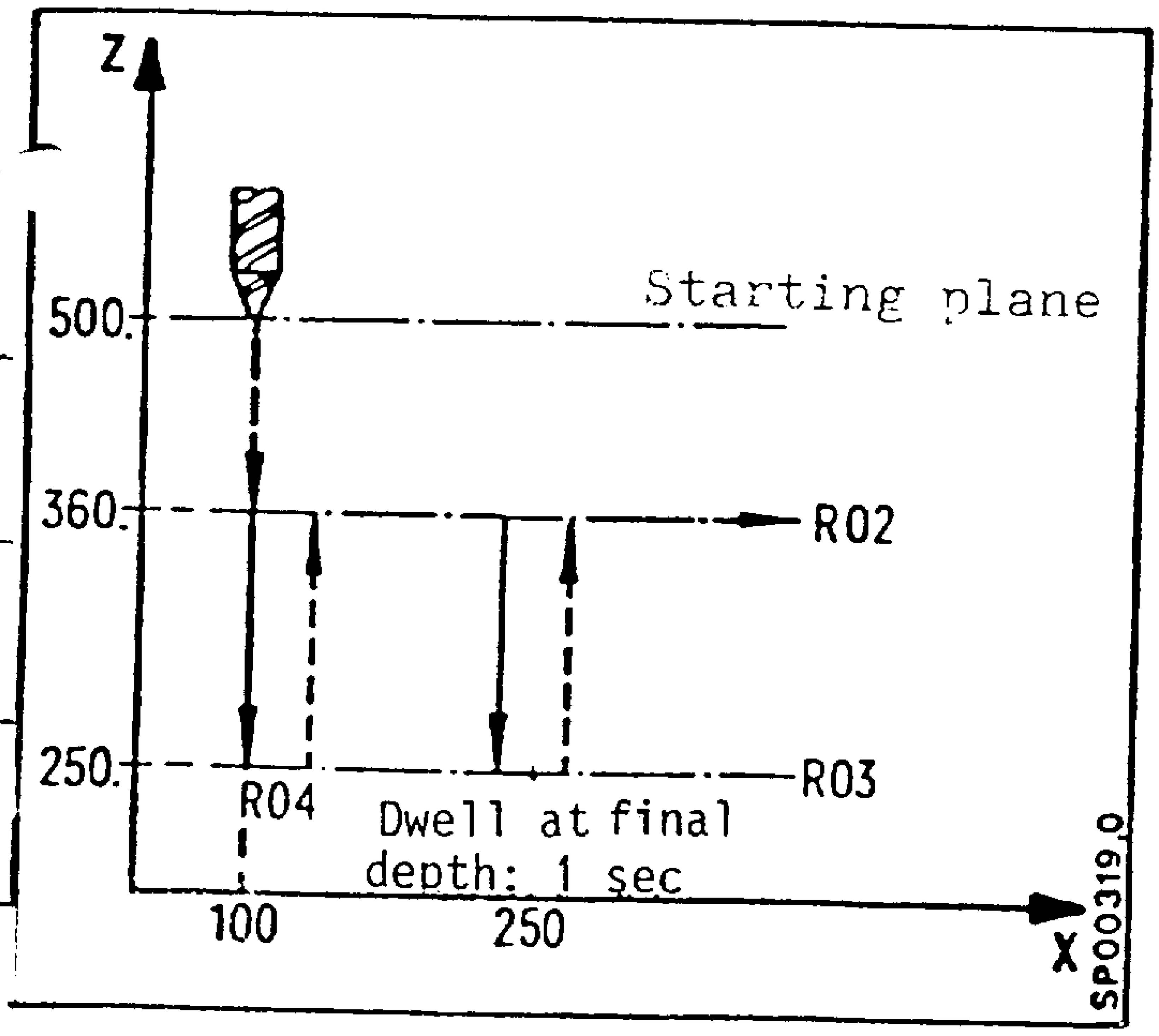
Call with L81

```
N8101   G90   S48   M03   F460.                          LF
N8102   G00   D01   Z500.                                LF
N8103   X100.   Y150.                                    LF
N8104   L81   R02   360.   R03   250.   R11 3.           LF   - Call-up drilling
                                                                cycle, 1st hole
N8105   X250.   Y300.                                    LF
N8106   L81   R02 . . .                                  LF   - Call-up drilling
.                                                               cycle, 2nd hole
.
N81..   Z500                                             LF
```

As opposed to the call-up with G81, here the drilling cycle must be called up anew at every new drilling position.

Call-up G82 (Drilling, countersinking, e.g. with variable boring axis)

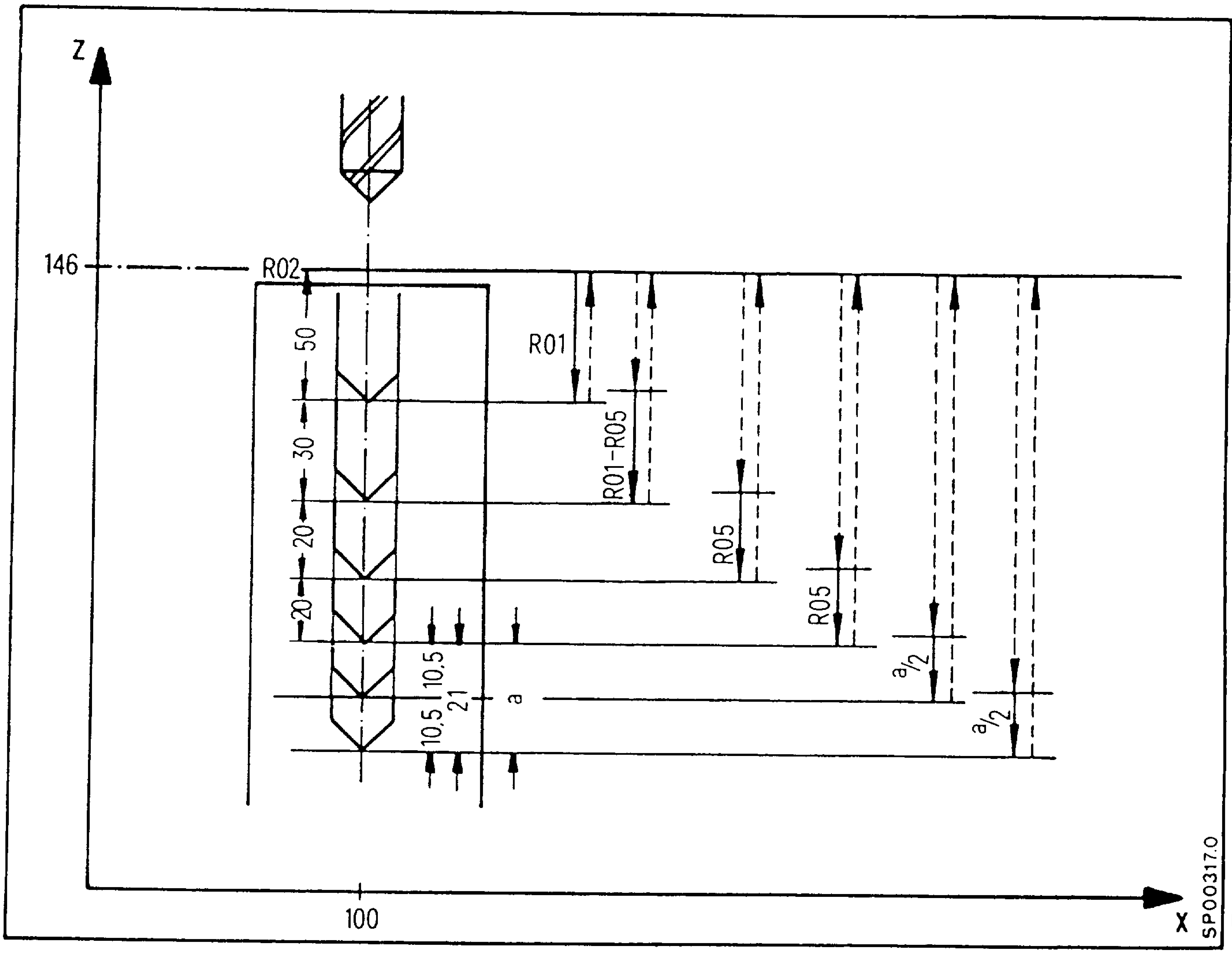
```
N8201   ...   M03   F460.                                        LF
N8202   G00   D01   Z500.                                        LF
N8203   X100.   Y150.                                            LF
N8204   G82   R02   360.   R03   250.   R04   1.   R11 3.        LF
N8205   X250.   Y300.                                            LF
.
.
N82..   G80   Z500.                                              LF
```



Call L83 (Deep hole drilling)

First drilling depth	50 mm	R01 50.
Reference plane = retract plane	146 mm	R02 146.
Final drilling depth	5 mm	R03 5.
Dwell at starting point	5 s	R00 5.
Dwell at final depth	1 s	R04 1.
Degression value	20 mm	R05 20.
Drilling axis (Z)	3	R11 3.

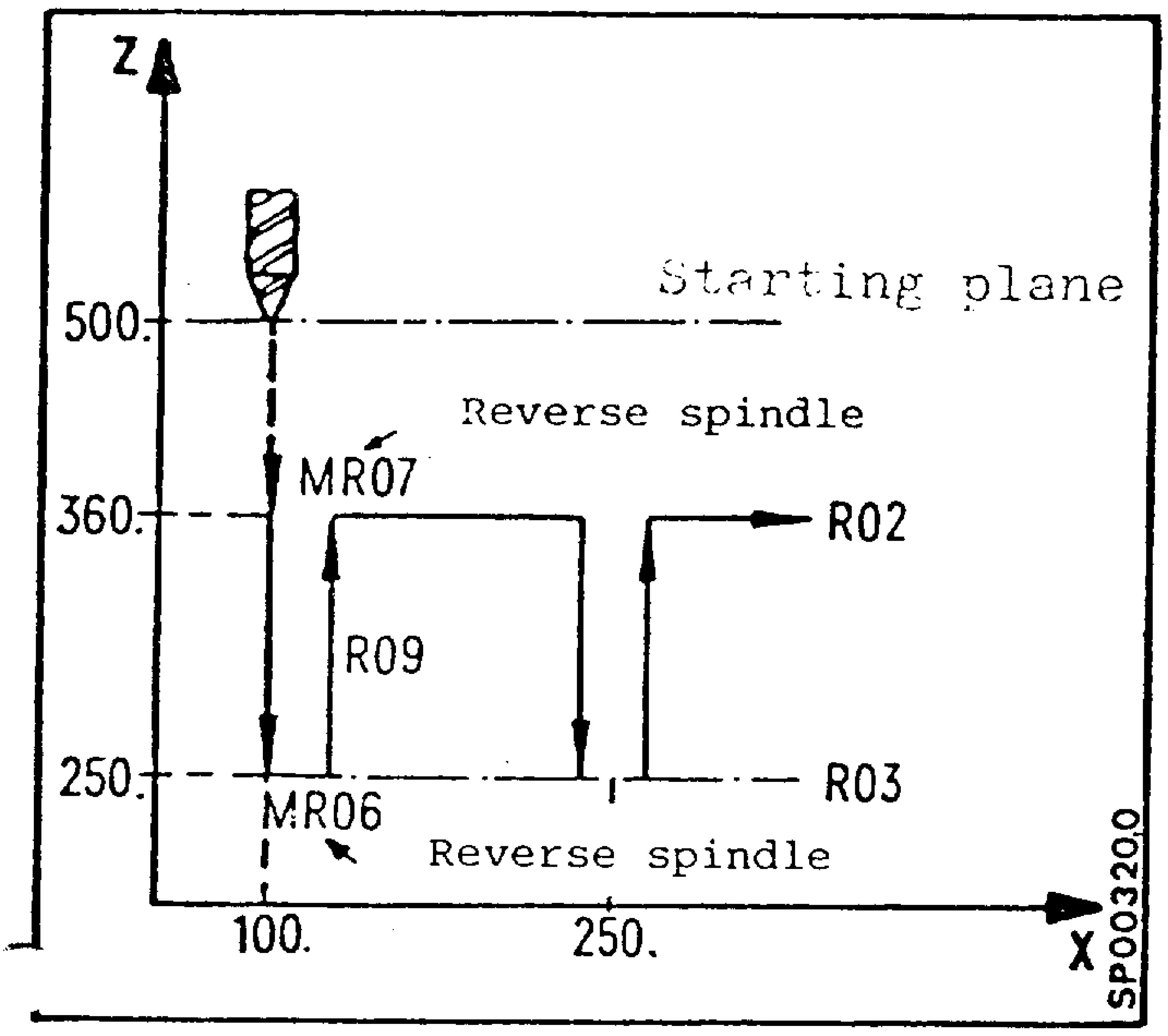
```
N8301   ...   S48   M03   F460.                                              LF
N8303   G00   D01   Z500.                                                    LF
N8304   X100.   Y150.                                                        LF
N8305   G83   R01  50.  R02  146.  R03  5.  R00  5.  R04  1.  R05  20.  R11  3.  LF
N8306   X250.   Y300.                                                        LF
.
.
N83..   G80   Z500.
```



At the rapid traverse advance respective to the new drilling depth, a safety distance of 1 mm is kept (on account of the chips still remaining in the hole).
With the inch system (G70) the safety distance must be changed accordingly.

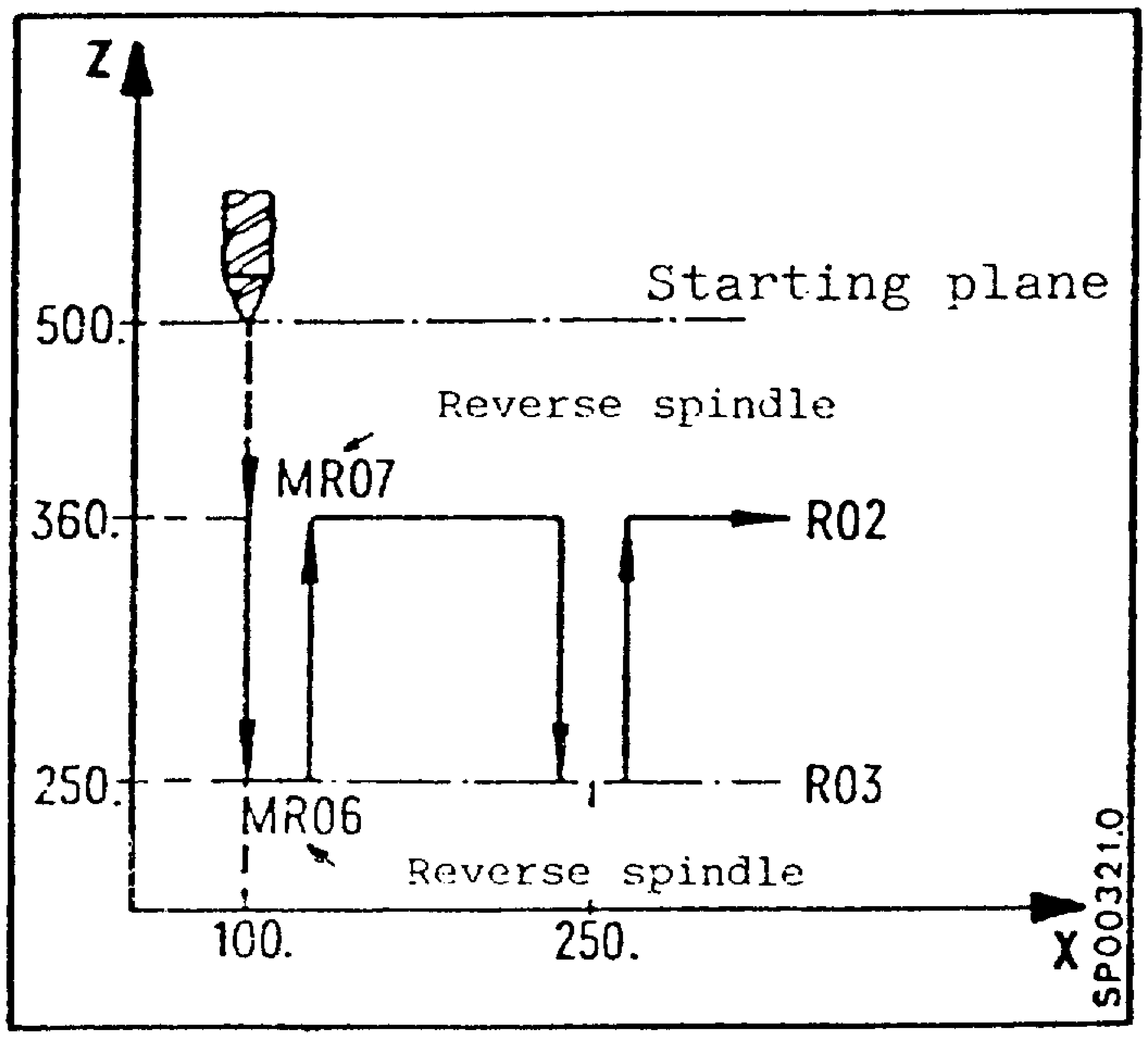
Call-up G84 (Tapping for machines with spindle encoder)

```
N8401  ...   S48   M03   F460.                                        LF
N8402  G00   D01   Z500.                                              LF
N8403  X100.  Y150.                                                   LF
N8404  G84   R02 360.  R03 250.  R06 04  R07 03  R09 5. R11 3.        LF
N8405  X250.  Y300.                                                   LF
.
.
N84..  G80   Z500.                                                    LF
```



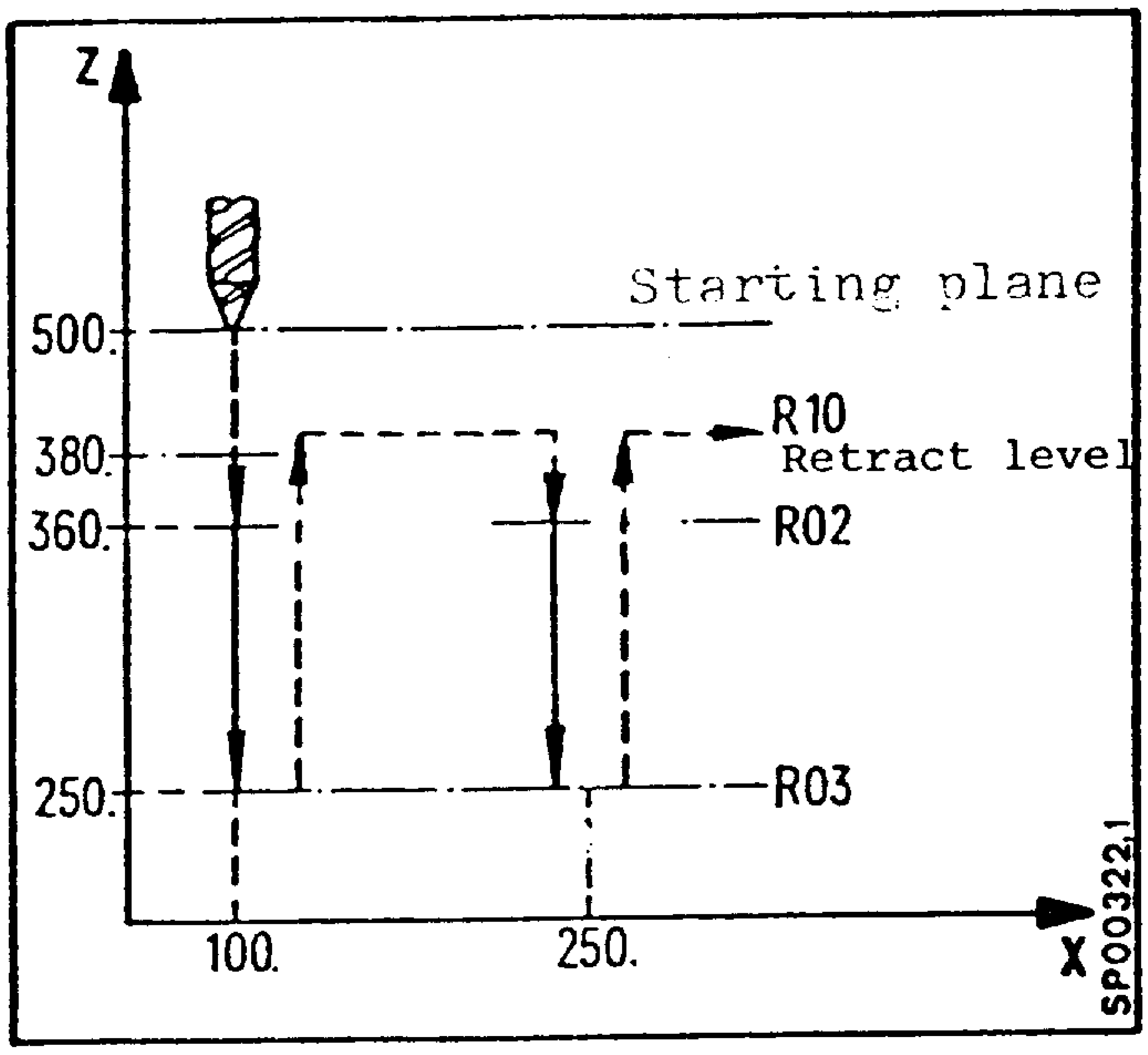
Call G84 (Tapping for machines without spindle encoder)

```
N8401   ...   S48   M03   F460.                                    LF
N8402   G00   D01   Z500.                                           LF
N8403   X100.   Y150.                                               LF
N8404   G84   R02 360.   R03 250.   R06 04   R07 03   R11 3.         LF
N8405   X250.   Y300.                                               LF
.
.
N84..   G80   Z500.                                                 LF
```



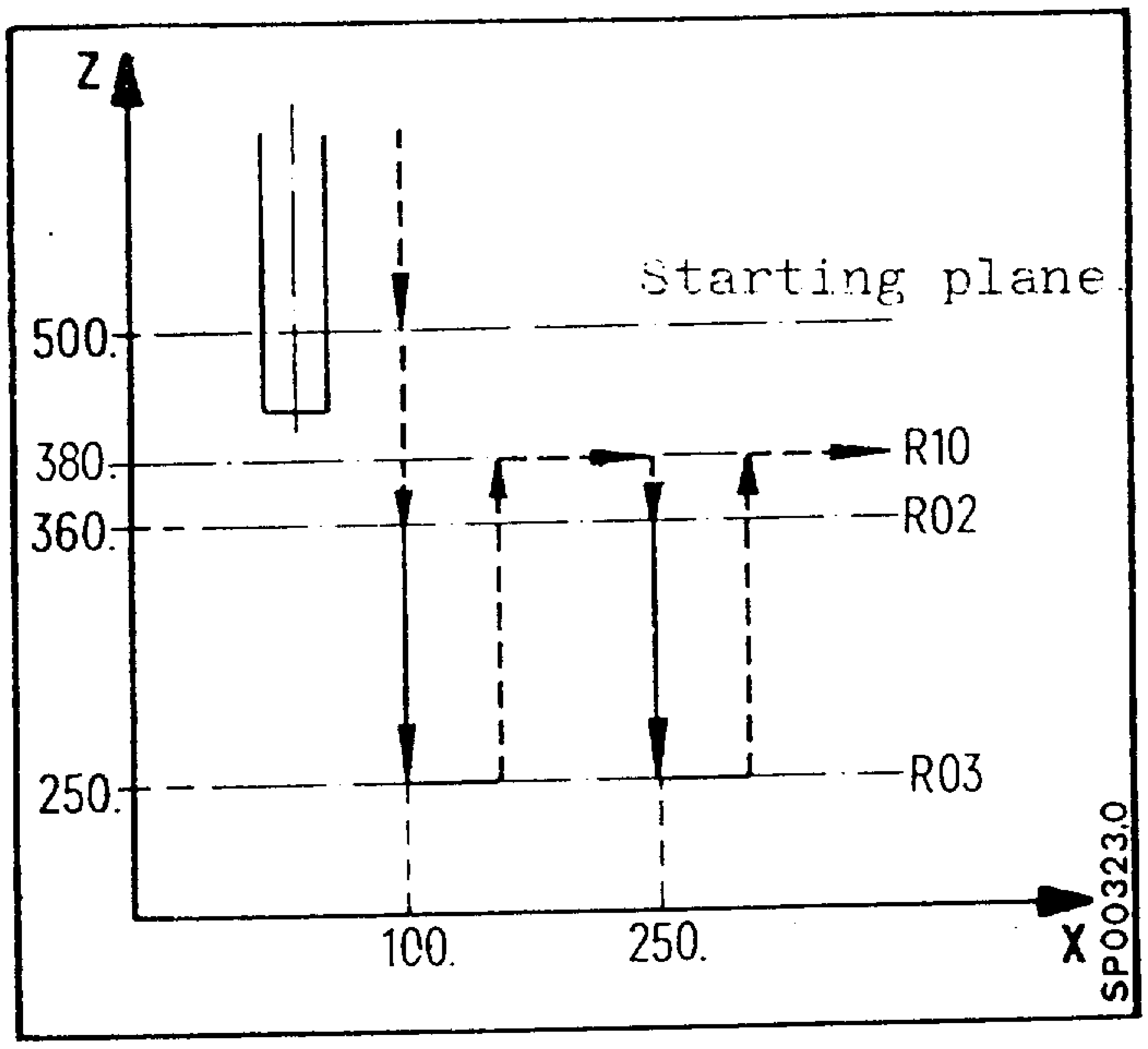
Call G85 (Boring 1)

```
N8501   ...   S48   M03   F460.                          LF
N8502   G00   D01   Z500.                                LF
N8503   X100.   Y150.                                    LF
N8504   G85   R02 360   R03 250.   R10 380.   R11 3.     LF
N8505   X250.   Y300.                                    LF
.
.
N85..   G80   Z500.                                      LF
```



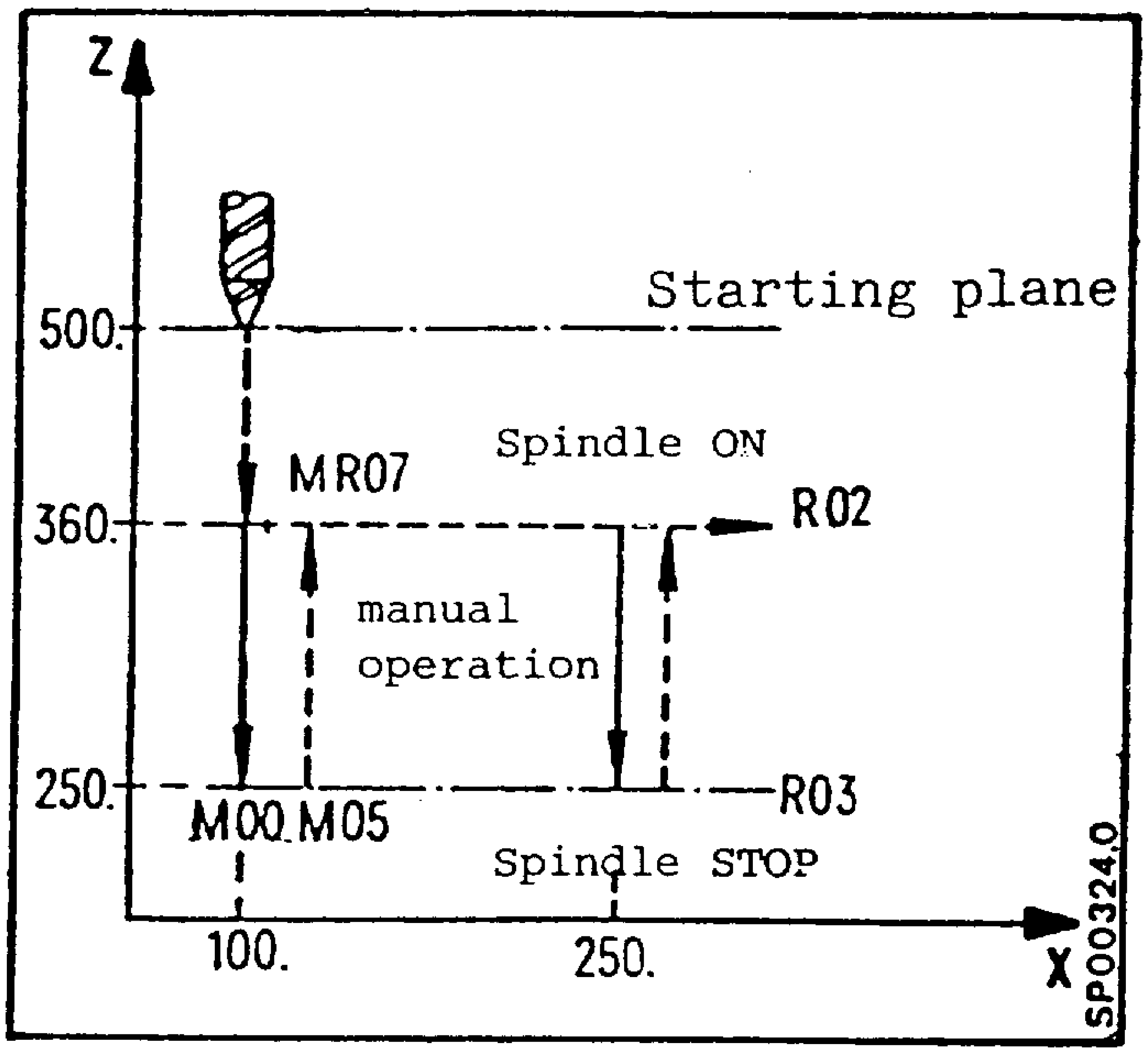
Call G86 (Boring 2)

```
N8601   ...   S48   M03   F460.                                         LF
N8602   G00   D01   Z500.                                               LF
N8603   X100.   Y150.                                                   LF
N8604   G86   R02 360.   R03 250.   R07 04.   R10 380.   R11 3.         LF
N8605   X250.   Y300.                                                   LF
.
.
N86..  G80   Z500.                                                      LF
```



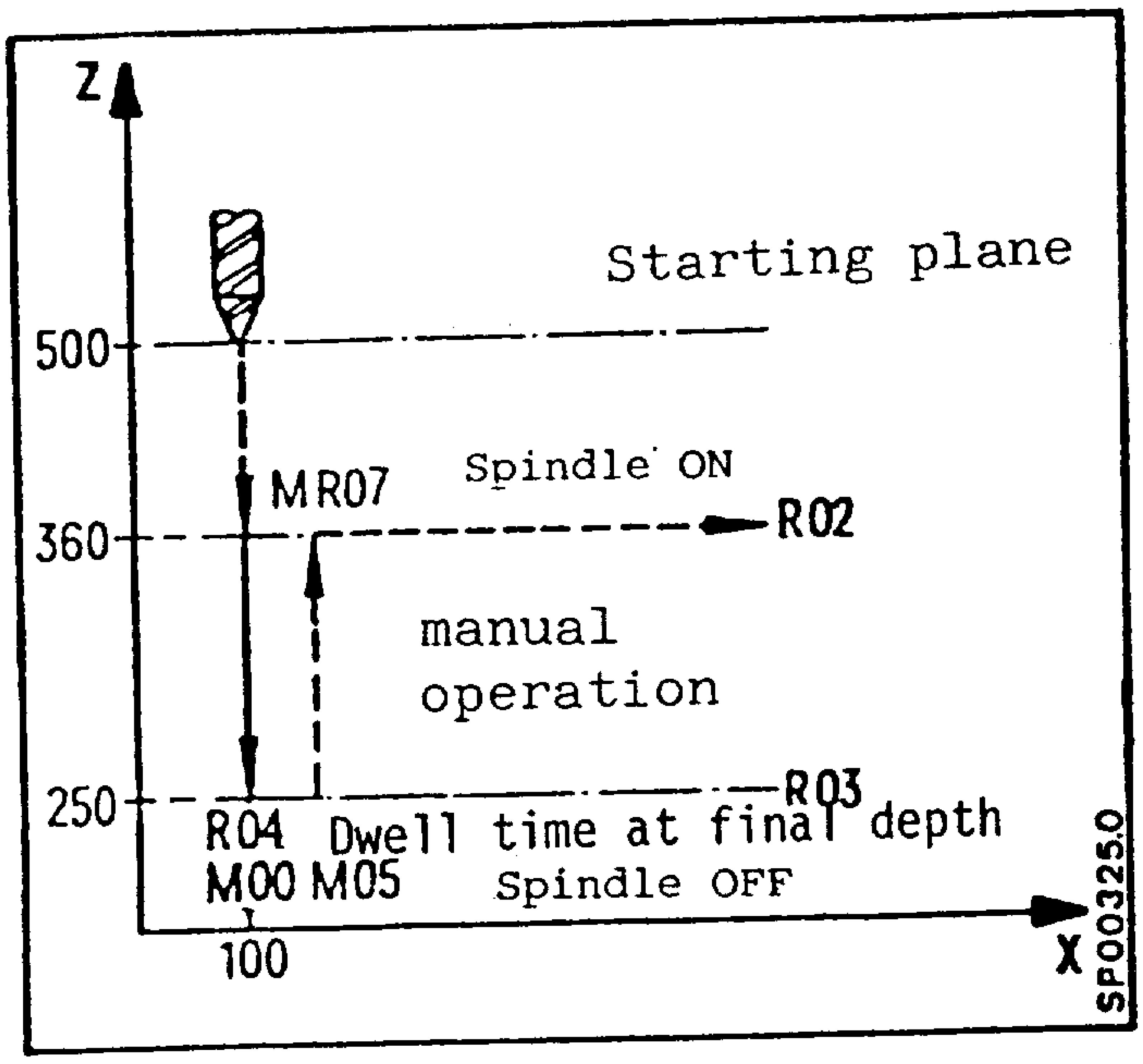
Call G87 (Boring 3)

```
N8701   ...   S48   M03   F460.                        LF
N8702   G00   D01   Z500.                              LF
N8703   X100.  Y150.                                   LF
N8704   G87   R02 360.   R03 250.   R07 04   R11 3.    LF
N8705   X250.  Y300.                                   LF
.
.
N87..   G80   Z500.                                    LF
```



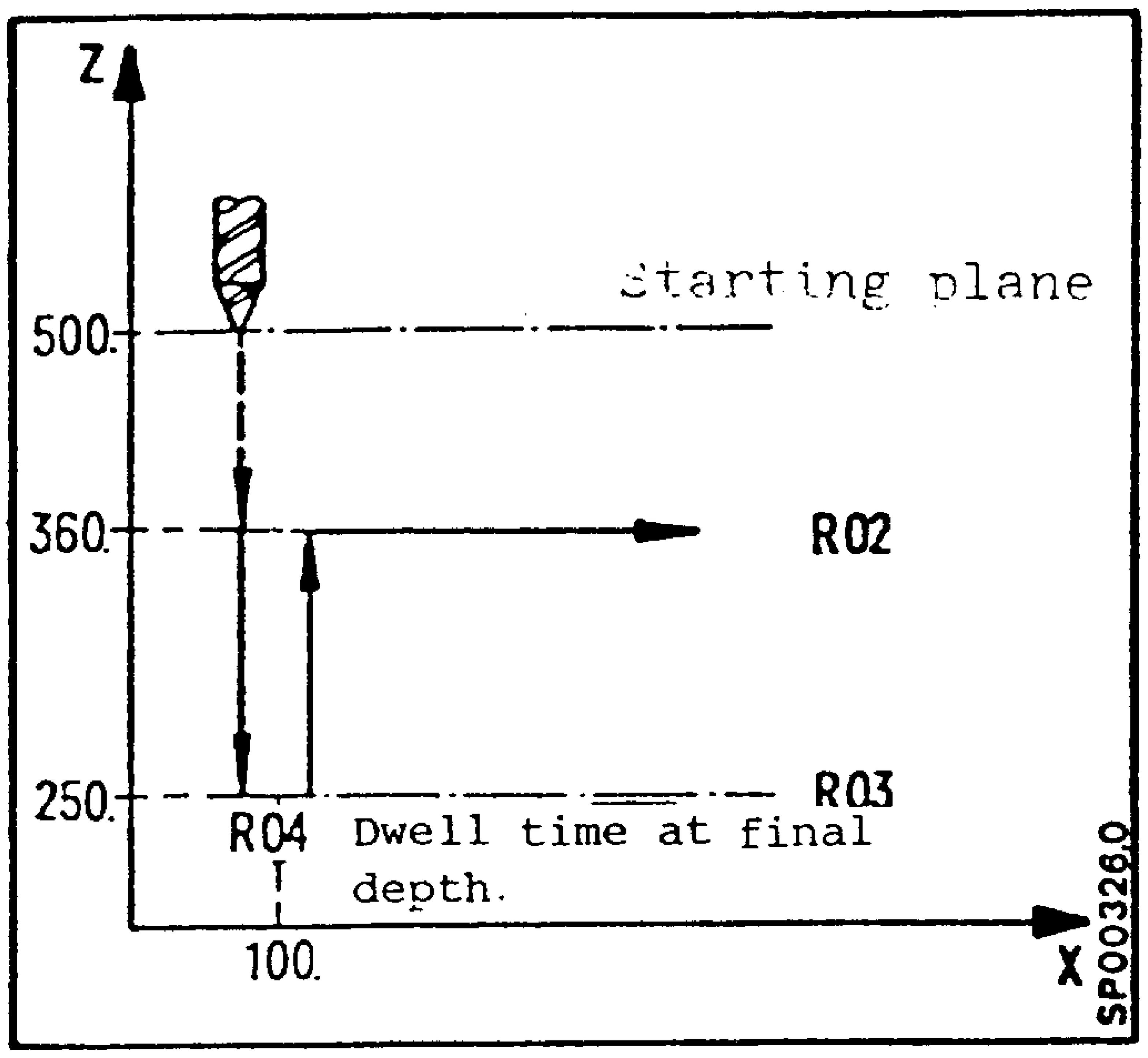
Call G88 (Boring 4)

```
N8801   ...   S48   M03   F460.                                  LF
N8802   G00   D01   Z500.                                          LF
N8803   X100.   Y150.                                              LF
N8804   G88   R02 360.   R03 250.   R04 1.   R07 04   R11 3.       LF
N8805   X250.   Y300.                                              LF
.
.
N88..   G80   Z500.                                                LF
```



Call G89 (Boring 5)

```
N8901   ...   S48   M03   F460.                         LF
N8902   G00   D01   Z500.                               LF
N8903   X100.   Y150.                                   LF
N8904   G89   R02 360.   R03 250.   R04 1.   R11 3.     LF
N8905   X250.   Y300.                                   LF
.
.
N89..   G80   Z500.                                     LF
```



7.2.1 Calling boring cycles in a subroutine

If boring cycles are called in a subroutine, the following procedure is necessary:

```
%
.
.
.
R02 360. R03 250. R00 81. R11 3.  - Supply boring cycle parameters

L0101                             - Boring positions
M30

L0101  (Boring positions)

G R00 X1 Y1                       - 1st Boring position
X2 Y2                             - 2nd Boring position
X5 Y5                             - 3rd Boring position
X10                               - 4th Boring position
G80 M17                           - Deselect boring cycle and
                                    end of subroutine
```

7.3 Listing of the boring cycles

7.3.1 Boring cycles G80-G89 for boring axis Z (basic controls 0, 1, 2)

E321 3M 0-2 GE 548811.2090.01 (old designation)

E822 3M 0-2 GE 548811.9102.01

```
% S P
L8100 G G60 G90 Z R02 (10.03.83 [Z])
G1 Z R03
G Z R02 M17
L8200 G G60 G90 Z R02
G1 Z R03
G4 F R04
G Z R02 M17
L8300
R28 1
G G60 G90 Z R02
R01 R05
R02 -R03
@03 4 R01 R02
R01 -R05
R02 R03
R23 0 R01
R24 0 R03
R24 R05
R22 0 R02
N2 R22 -R23
@03 3 R24 R22
G1 Z R22
G4 F R04
G Z R02
G4 F R00
R22 R28
G Z R22
R22 -R28
R23 -R05
@03-2 R23 R05
R23 0 R05
@00-2
N3 R25 2
R22 R23
R22 -R03
R22 /R25
R03 R22
G1 Z R03
G4 F R04
G Z R02
G4 F R00
R03 R28
G Z R03
R03 -R28
R03 -R22
G1 Z R03
@00 5
```

```
N4 R02 R03
R01 -R05
R02 -R01
R26 0 R02
G1 Z R26
R02 R01
G4 F R04
G Z R02
G4 F R00
R26 R28
G Z R26
R26 0
G1 Z R03
N5 G4 F R04
G Z R02 M17
L8400 G G60 G90 Z R02
G1 G63 Z R03
M R06
Z R02
G G60 M R07
M17
L8500 G G60 G90 Z R02
G1 Z R03
G Z R10 M17
L8600 M R07
G G60 G90 Z R02
G1 Z R03
M5
G Z R10 M17
L8700 M R07
G G60 G90 Z R02
G1 Z R03
M5
M
G Z R02 M17
L8800 M R07
G G60 G90 Z R02
G1 Z R03
G4 F R04
M5
M
G Z R02 M17
L8900 G G60 G90 Z R02
G1 Z R03
G4 F R04
Z R02
G M17
L9000 G G60 G90 Z R02
G33 Z R03 K R09
Z R02 K R09 M R06
G M R07
M17
L9900
@31 M17
M02
```

7.3.2 Boring cycles G80 - G89 for boring axis Z

(basic controls 3, 4, 4B, 4C)

E321	3M	3-4	GE	548815.2090.01	(old designation)
E822	3M	3-4B	GE	548815.9101.01	

```
%SP
L8100 G G60 G90 Z R02 (10.03.83[Z13)
G1 Z R03
G Z R02 M17
L8200 G G60 G90 Z R02
G1 Z R03
G4 F R04
G Z R02 M17
L8300 R78 1
R63 0 R02 R64 0 R01 R65 0 R67 2
R63 -R03
R67 .R05
G G60 G90 Z R02
N3 R63 -R64 R62 0 R03
@03 4 R65 R63
R62 R63
G1 Z R62
G4 F R04
G Z R02
G4 F R00
R62 R78
Z R62 -R78
@03 4 R05 R63
R64 -R05
@02-3 R64 R05
R64 0 R05
@03-3 R63 R67
R64 0 R63 R62 2
R64 /R62
@00-3
N4 G1 Z R03
G4 F R04
G Z R02 M17
L8400 G G60 G90 Z R02
G1 G63 Z R03
M R06
Z R02
G G60 M R07
M17
L8500 G G60 G90 Z R02
G1 Z R03
G Z R10 M17
L8600 M R07
G G60 G90 Z R02
G1 Z R03
M5
G Z R10 M17
L8700 M R07
G G60 G90 Z R02
```

```
G1 Z R03
M5
M
G Z R02 M17
L8800 M R07
G G60 G90 Z R02
G1 Z R03
G4 F R04
M5
M
G Z R02 M17
L8900 G G60 G90 Z R02
G1 Z R03
G4 F R04
Z R02
G M17
L9000 G G60 G90 Z R02
G33 Z R03 K R09
Z R02 K R09 M R06
G M R07
M17
L9900
@31 M17
M02
```

7.3.3 Boring cycles G80/G89 with variable boring axes

(Basic controls 3, 4, 4B, 4C)

E321	3M	3-4	GE	548815.2091.01	(old designation)
E822	3M	3-4B	GE	548815.9101.01	

```
%SP
L8100 (10.03.83[VAR])
@20 @92 R11
G G60 G90 @92 R02
G1 @92 R03
G @97 R02 M17
L8200
@20 @92 R11
G G60 G90 @92 R02
G1 @92 R03
G4 F R04
G @92 R02 M17
L8300 R78 1 R65 0 R67 2
R64 0 R01 R63 0 R02 R66 1
@20 @92 R11
R67 .R05
R63 -R03
@03 2 R63 R65
R66-1
N2 R63 .R66
R78 .R66
G G60 G90 @92 R02
N3 R63 -R64
@03 4 R65 R63
R63 .R66 R62 0 R03
R62 R63
G1 @92 R62
G4 F R04
G @92 R02
G4 F.R00
R62 R78
@92 R62 -R78
R63 .R66
@03 4 R05 R63
R64 -R05
@02-3 R64 R05
R64 0 R05
@03-3 R63 R67
R64 0 R63 R62 2
R64 /R62
@00-3
N4 G1 @92 R03
G4 F R04
G @92 R02 M17
L8400
@20 @92 R11
G G60 G90 @92 R02
G1 G63 @92 R03
M R06
```

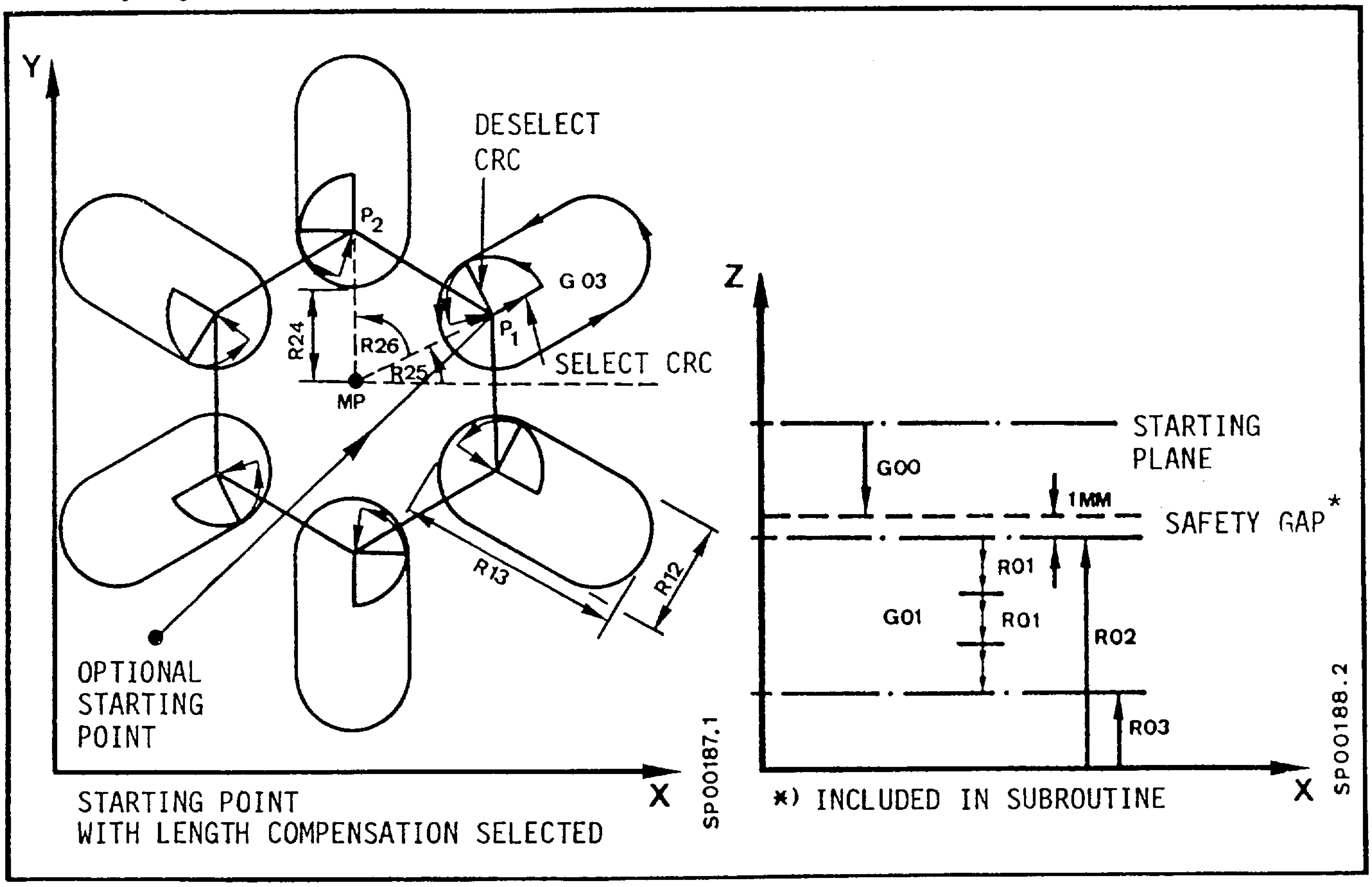
```
@92 R02
G G60 M R07
M17
L8500
@20 @92 R11
G G60 G90 @92 R02
G1 @92 R03
G @92 R10 M17
L8600
@20 @92 R11
M R07
G G60 G90 @92 R02
G1 @92 R03
M5
G @92 R10 M17
L8700
@20 @92 R11
M R07
G G60 G90 @92 R02
G1 @92 R03
M5
M
G @92 R02 M17
L8800
@20 @92 R11
M R07
G G60 G90 @92 R02
G1 @92 R03
G4 F R04
M5
M
G @92 R02 M17
L8900
@20 @92 R11
G G60 G90 @92 R02
G1 @92 R03
G4 F R04
@92 R02
G M17
L9000 R77 3
@20 @92 R11
G G60 G90 @92 R02
@02 3 R11 R77
@00 R11
N1 G33 @92 R03 I R09
@92 R02 I R09 M R06
@00 5
N2 G33 @92 R03 J R09
@92 R02 J R09 M R06
@00 5
N3 G33 @92 R03 K R09
@92 R02 K R09 M R06
N5 G M R07
M17
L9900
@31 M17
M02
```

7.4 Keyway milling pattern L94

(Requirement: polar coordinates, Option B63)
(Reading/loading from, system memories, option B76)
When programming only the subroutine L94 is called and the subsequent parameters are assigned values.

R04 **	Feed (depth)
R15 **	Feed (surface)
R01	Infeed (incremental without sign)
R02	Reference plane in absolute dimensions
R03	Keyway depth in absolute dimensions
R22, R23	MP-Midpoint of the slot referred to tool zero
R24	Radius
R25	Starting angle (referred to the horizontal axis)
R26	Indexing angle, if zero is selected as the indexing angle, then the number of keyways is subdivided accordingly
R27	Number of keyways
R12, R13	Parameter for "keyways"

The milling cutter diameter may not exceed a maximum of 0.9 of the keyway width!



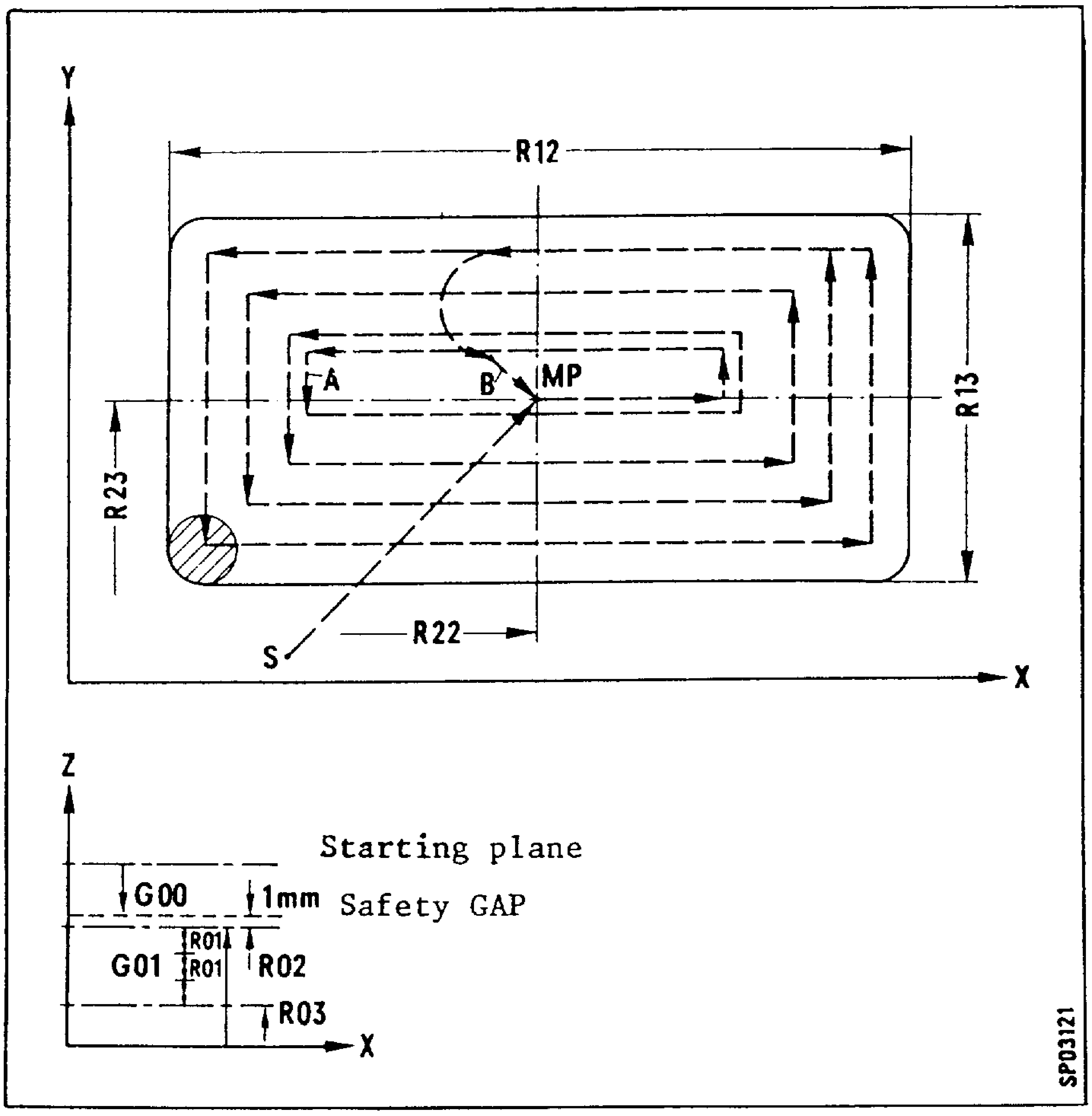
** Basic control 4B, 4C

7.5 Milling rectangular shape, subroutine L95

(Requirement reading/ loading from system memories,Option B76
Insertion of chamfers and radii, Option C33 or B75)

When programming, only subroutine L95 is called and the subsequent parameters are assigned values. The machining plane must be defined in the main program.

** R11	Boring axis (variable plane)
*R04	Feed (depth of pocket)
*R15	Feed (surface of pocket)
R01	Infeed (incremental without sign)
R02	Reference depth in absolute dimensions
R03	Depth of pocket in absolute dimensions
R06	Milling direction cw/ccw (02/03)
R22/R23	Midpoint of the shape referred to tool zero
R12/R13	Dimensions of shape X/Y
**R24	Radius



MP = Mid-point of pocket
* Basic control 4 from cycle revision .02 onwards
** Basic control 4B from NC-Software version 03 onwards

The tool follows the path shown after infeed in the part (optional clockwise or anticlockwise milling R06 = 02/03). The path is parallel to the outer edges of the shape and adds a maximum of $\sqrt{2} \cdot r$ (r = cutter radius) to these edges.

If the shape cannot be milled due to a too large cutting force requirement for infeed in the tool axis, then the infeed depth must be selected. The milling process repeats itself until the required depth has been reached. If R01 = 0, then the maximum infeed is immediately selected.

Subroutine L95 automatically selects the cutter radius compensation when required. This can lead to contour deviations during single block operation.

If the milling cutter diameter exceeds half the length of the shorter side of the pocket, the subroutine is not executed. There is a jump to the end of the program.
Subroutine L95 operates without CRC. The shorter side of the pocket must be at least 1 mm greater than the cutter diameter. Radius R24 must be at least 0.5 mm smaller than half the length of the shorter side (R12, R13). **

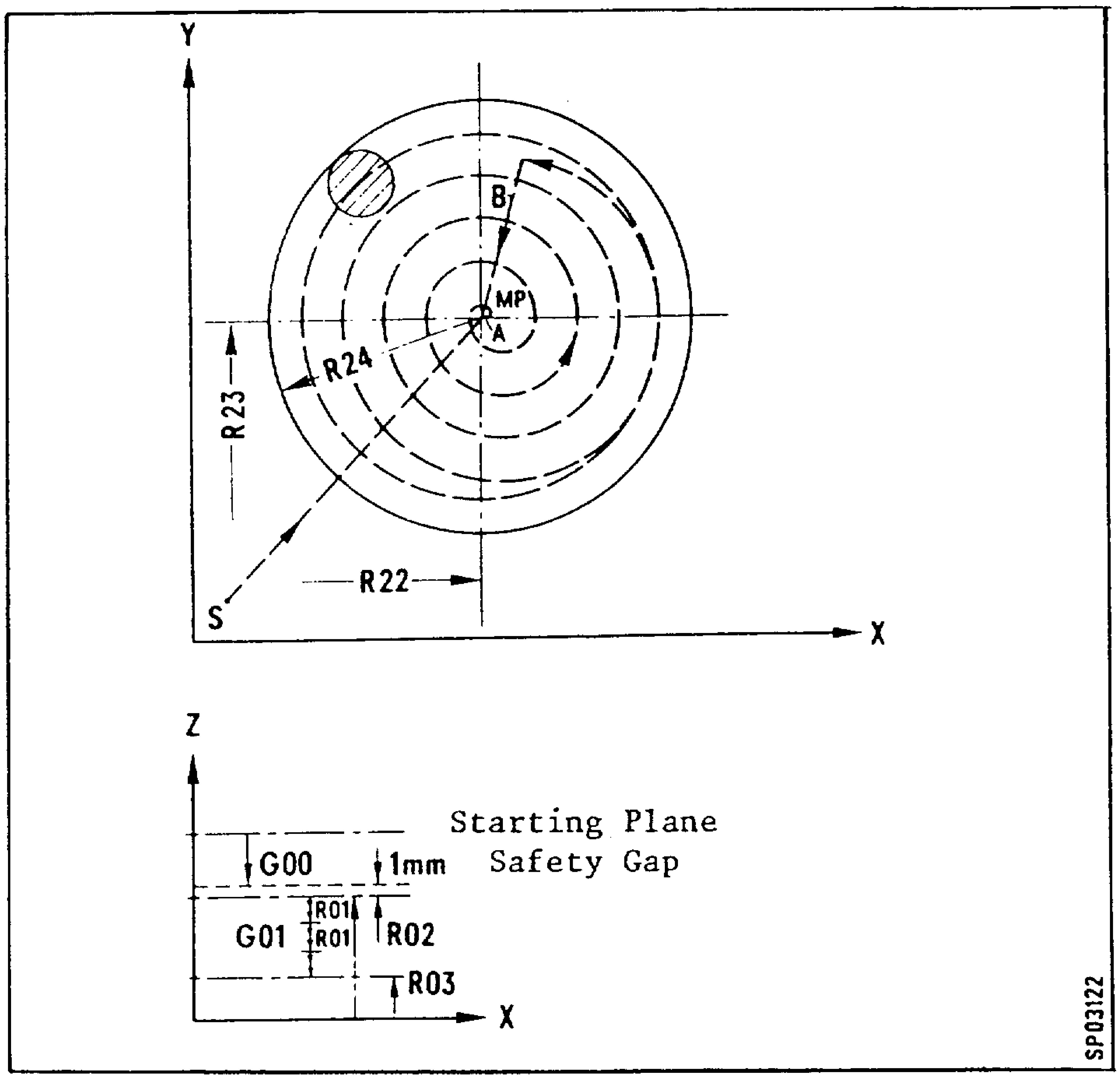
From basic control 4 onwards, the "cycle lock" signal on the interface must be active, and the limit of the R parameter display (machine parameter 382) must be set to ≥ 50.

* Basic control 4
** Basic control 4B from NC-software version 03 onwards, 4C

7.6 Milling circular shape

(Requirement: reading/loading from system memories; Option B76)
When programming, only the subroutine L96 is called and the subsequent parameters are assigned values.

R04	Feed (depth of pocket)
R15	Feed (surface of pocket)
R01	Infeed (incremental without sign)
R02	Reference plane in absolute dimensions
R03	Depth of pocket in absolute dimensions
R06	Milling direction cw/ccw (02/03)
R22/R23	MP midpoint of the shape referred to tool zero
R24	Shape radius
**R11	Boring axis (variable plane)



MP = Mid-point of pocket
A = CRC select
B = CRC cancel
S = Any desired starting point with selection of tool length compensation
** Basic control 4B, 4C

The tool follows the spiral path shown after boring in to the part. The path spirals outwards at infeed $\sqrt{2} \cdot r$ (r = cutter radius) (optional clockwise or counter-clockwise milling R06 = 02/03). If the shape cannot be milled due to a too large cutting force requirement for infeed in the tool axis, then the infeed depth must be selected. The milling process repeats itself until the required depth has been reached.

If R01 = 0, then the maximum infeed is immediately selected.

Subroutine L96 automatically selects the cutter radius compensation when required. This can lead to contour deviations during individual block operation. If the product of the milling cutter radius multiplied by 1.414 exceeds the radius of the pocket, the subroutine is not executed. There is a jump to the end of the program. *

Subroutine L96 operates without CRC. The radius of the pocket must be 1 mm greater than the cutter radius. **

From basic control 4 onwards, the "cycle lock" signal on the interface must be active and the limit of the R parameter display (machine parameter 382) must be set to ≧ 50.

* Basic control 4

** Basic control 4B from NC-software version 03 onwards, 4C

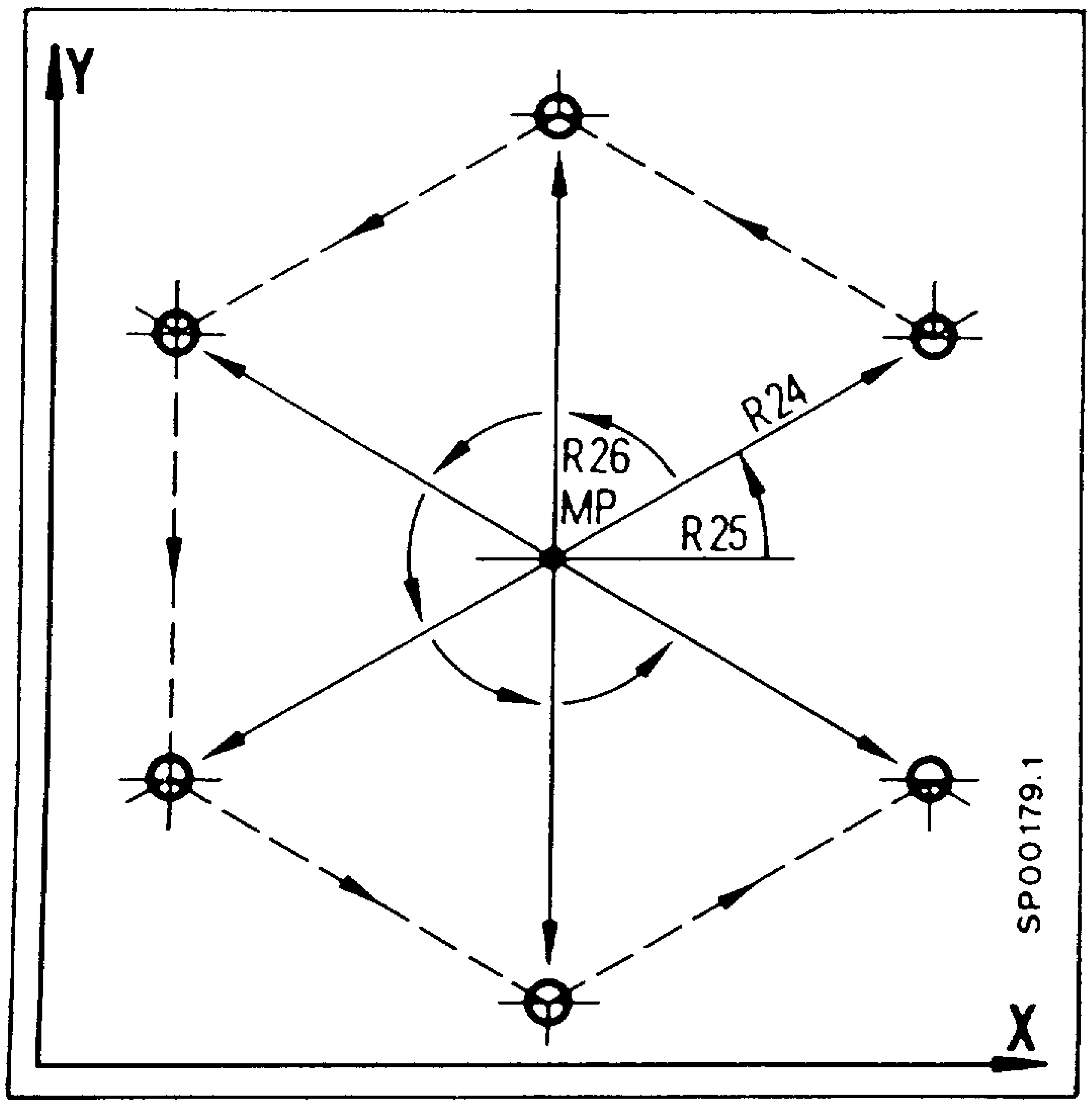
7.7 Drilling/boring patterns

7.7.1 Subroutine: Boring pattern L97

(Requirement: polar coordinates, Option B63)

When programming, only the subroutine L97 is called and the subsequent parameters are assigned values after the parameters for the desired boring cycle have been programmed.

R22, R23	MP midpoint of the boring pattern referred to workpiece zero
R24	Radius
R25	Starting angle (referred to the horizontal axis)
R26	Indexing angle, if zero is selected as the indexing angle, then the number of holes is subdivided accordingly
R27	Number of holes
R28	Number of the boring cycle required (81-89)
**R11	Boring axis (variable plane)



** Basic control 4B, 4C

Subroutine call

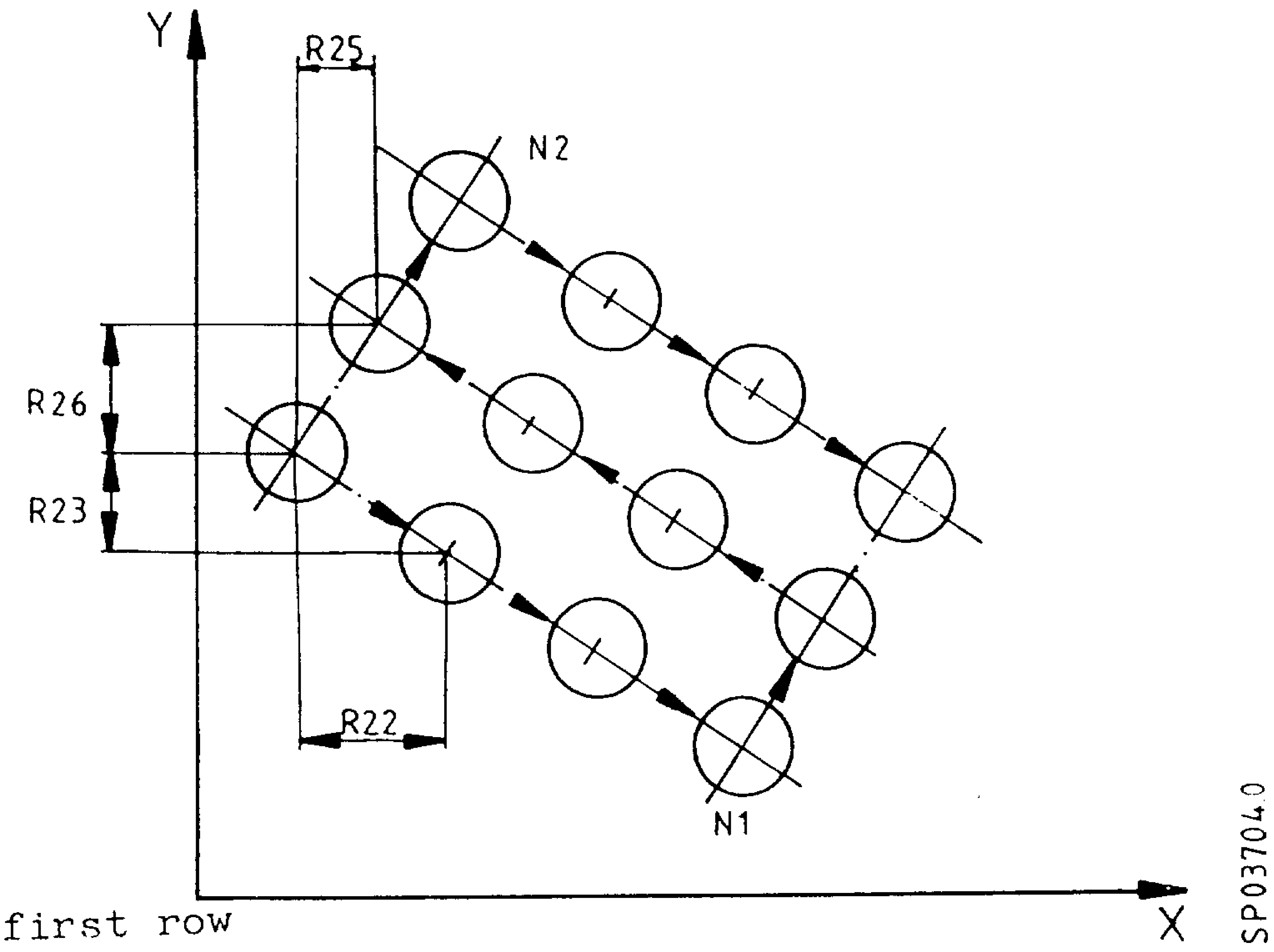
```
N1900 L97  R22.. R23.. R24.. R25..
           R26.. R27.. R28..        LF
N1901...
```

7.7.2 Subroutine: Boring/Drilling pattern L93 (up to basic control 4B)
Subroutine: Boring/Milling pattern L904 (basic control 4C)

If identical operations are required at a large number of points at regular intervals on a workpiece (boring/drilling patterns, cycles) they can be entered as a pattern in a simple block. The points are then approached in sequence. There the boring/drilling cycle containing the required operation is called. The tool moves from point to point in a straight line. The point pattern can be parallel or at an angle to the coordinate axes.

When programming, only subroutine L93 is called and values assigned to the parameters after the parameters have been programmed for the boring/drilling cycle.

R22/R23	Distance
R24	Number of bores (first row)
R25/R26	Distance
R27	Number of bores (second row)
R28	Required boring/drilling cycle (81-89)
R11	Boring axis (variable plane)



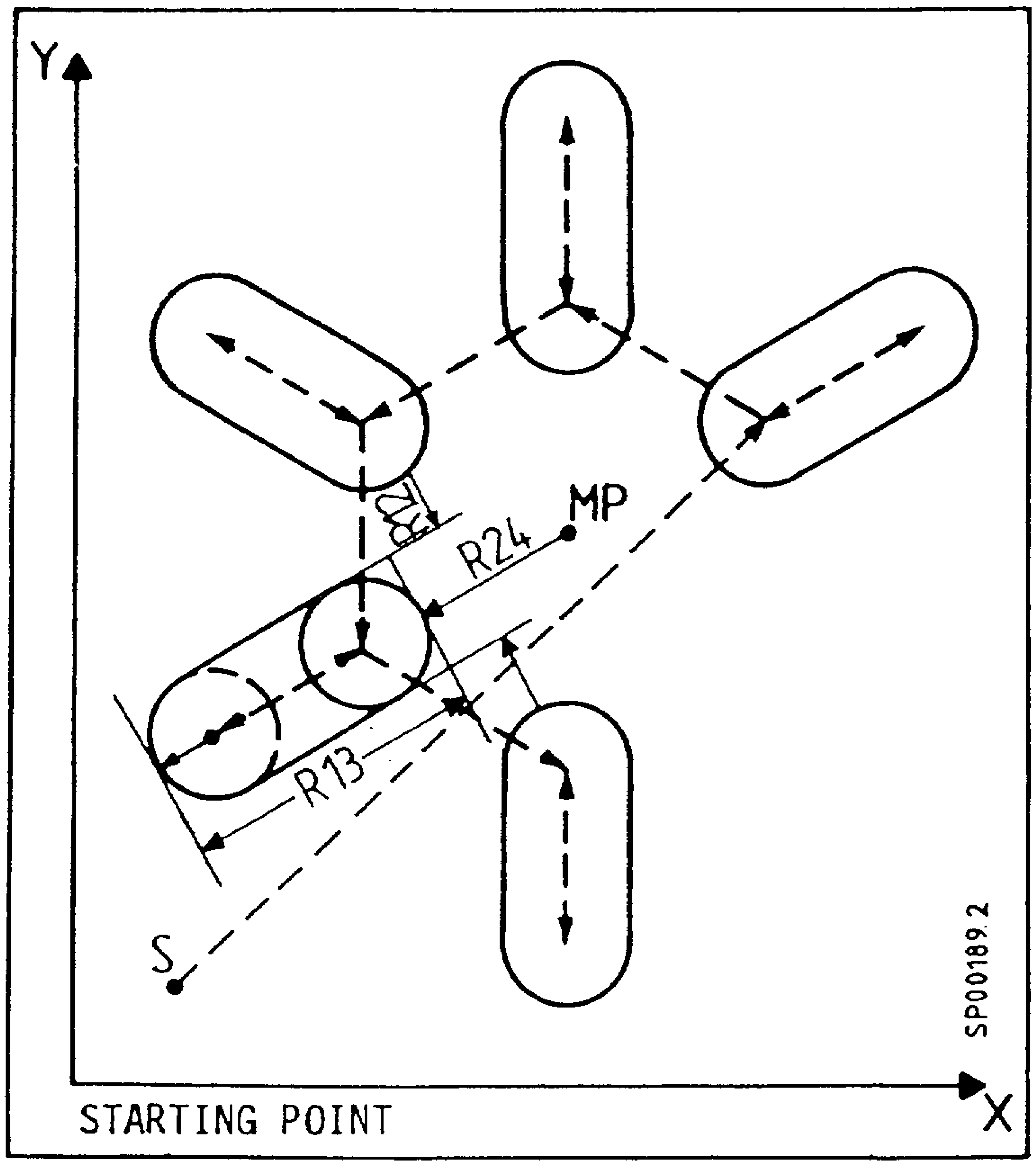
N1 = first row
N2 = second row

7.8 Subroutine Milling pattern for elongated hole L98

(Requirement: polar coordinates, Option B63)
(reading/loading from system memories; Option B76)

When programming only the subroutine L98 is called and the subsequent parameters are assigned values.

**R04	Feed (depth)
*R15	Feed (surface)
R01	**Infeed (incremental, without sign)**
R02	Reference plane
R03	Depth of elongated hole
R22,R23	MP midpoint of milling pattern referred to workpiece zero
R24	Radius
R25	Starting angle (referred to the horizontal axis)
R26	Indexing angle, if zero is selected as the indexing angle, then the number of elongated holes is subdivided accordingly.
R27	Number of elongated holes
R12	Diameter of tool
R13	Total length of elongated hole
*R11	Boring axis (variable plane)



* Basic control 4
** Basic controls 4B, 4C

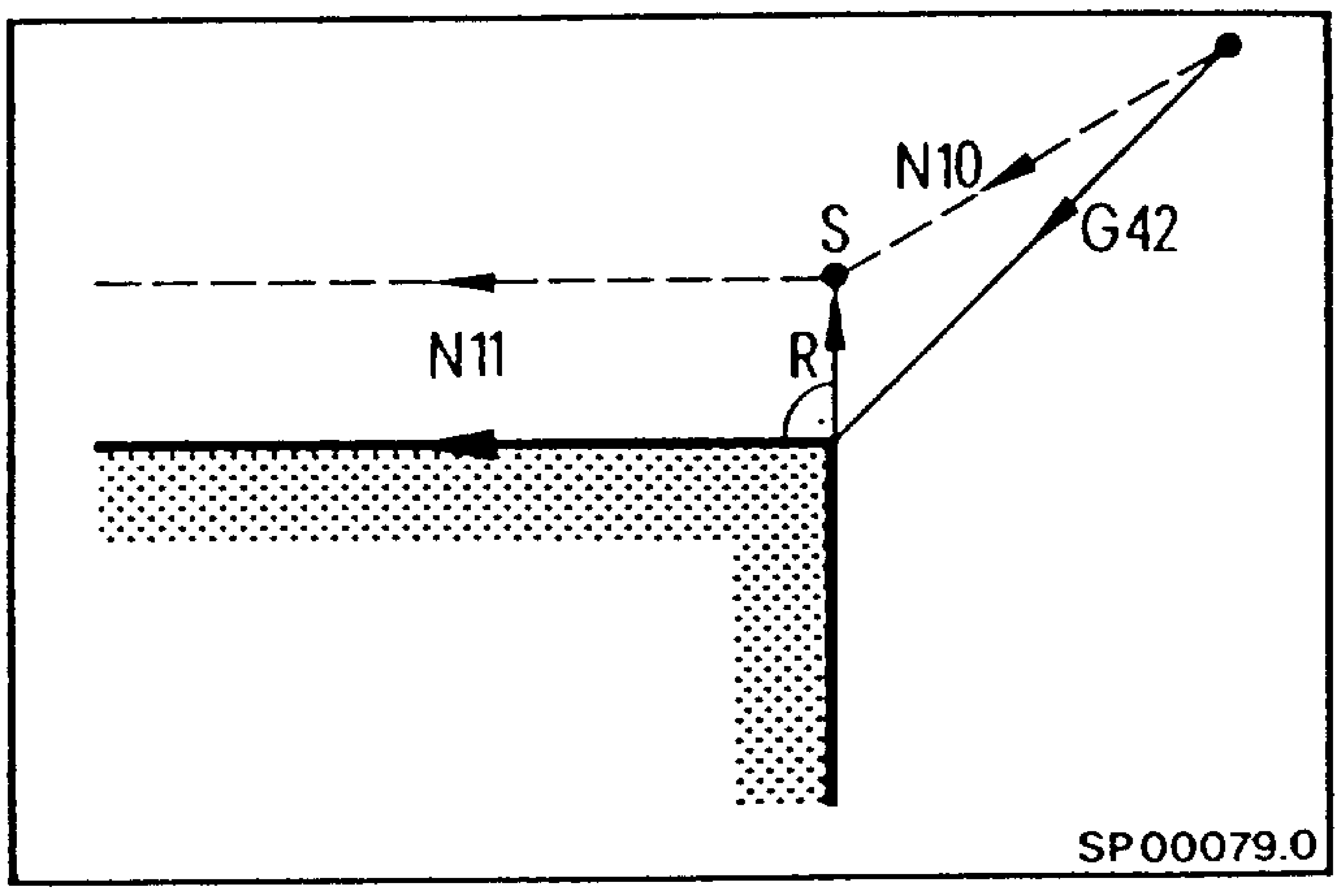
8. Appendix

8.1 Cutter radius compensation

In the following all stop points are designated by an S.

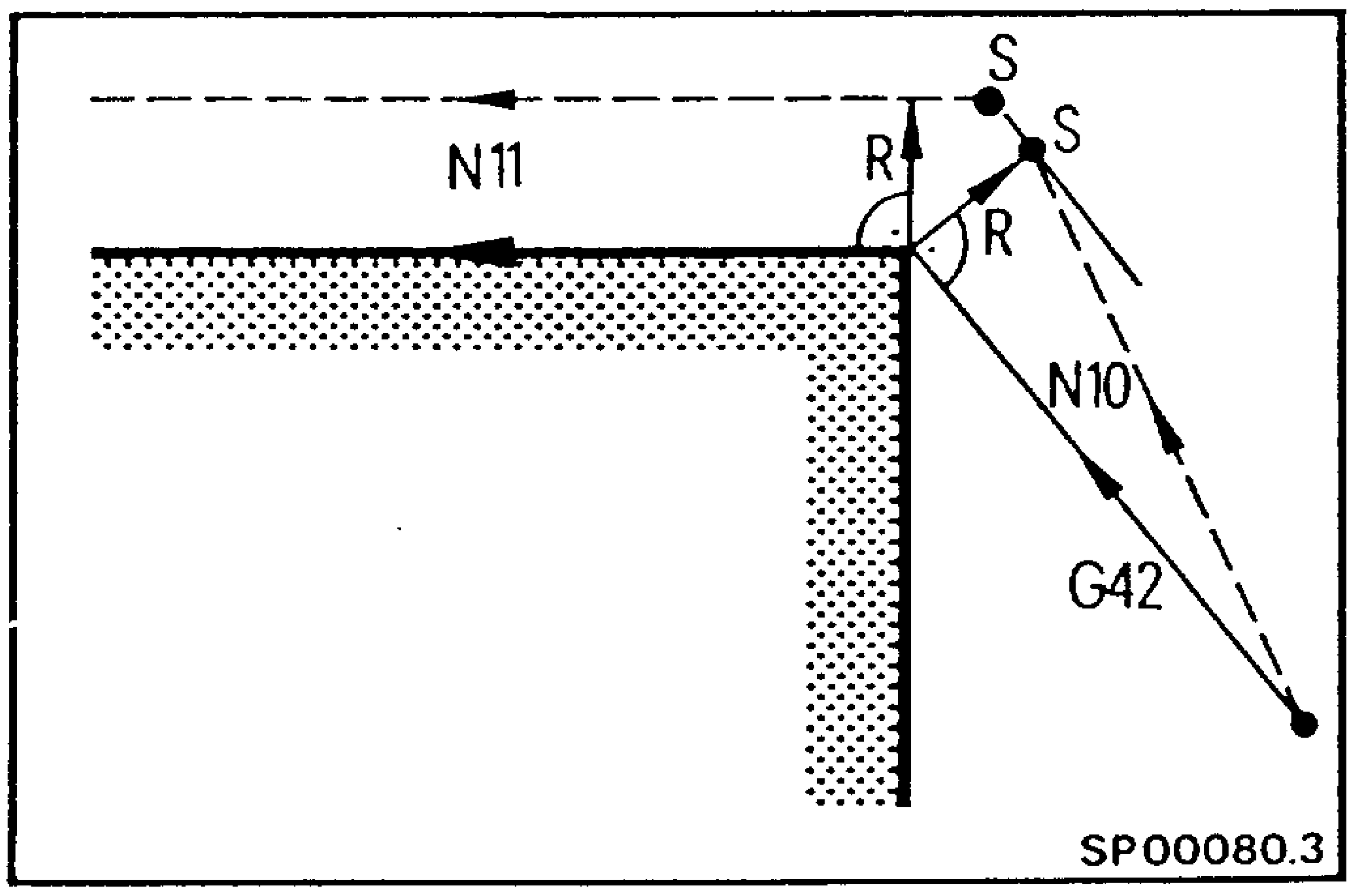
8.1.1 Selecting the CRC

- Inside contours (the angle between blocks N10 and N11 is less than 180°).

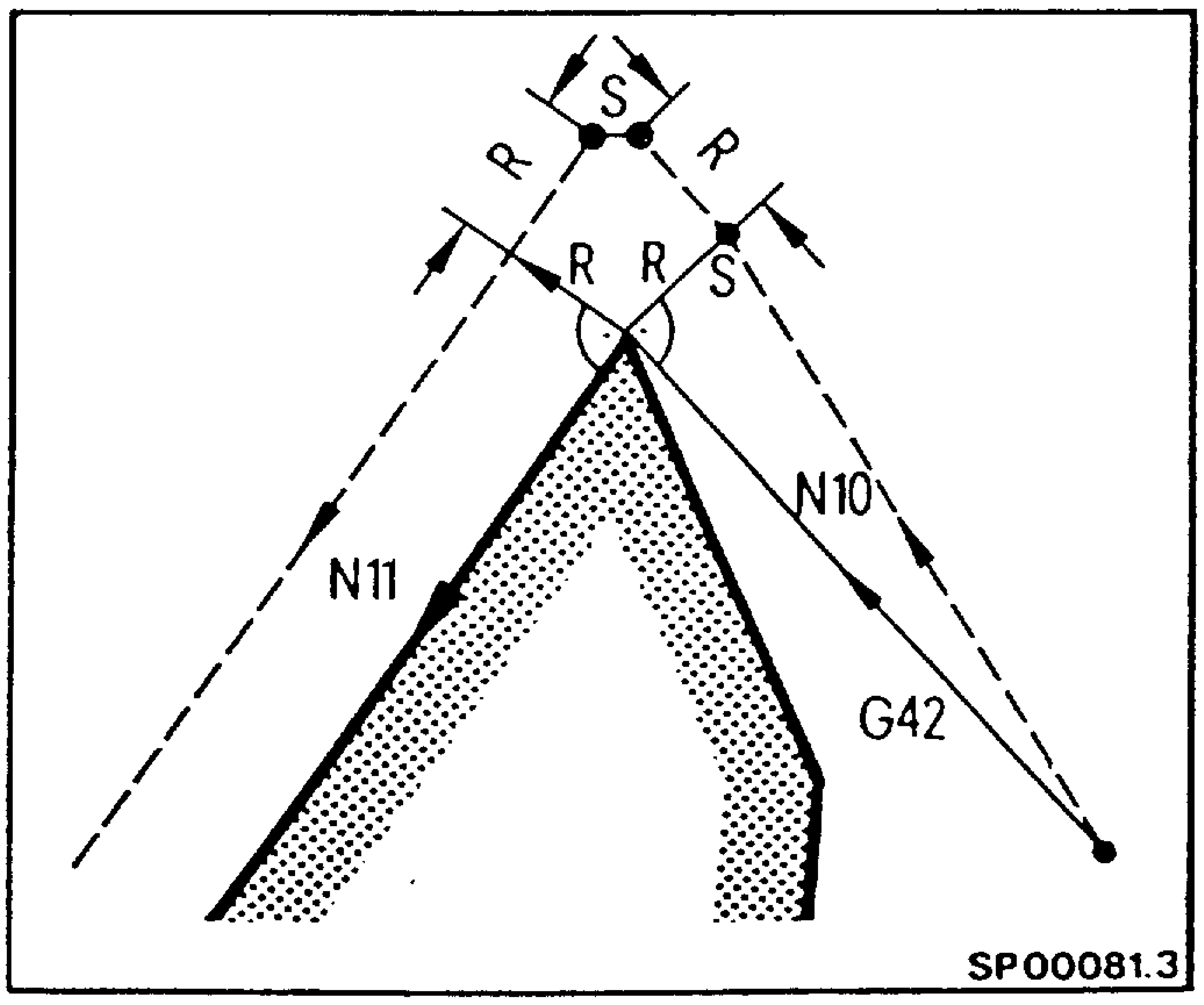


In a block following a block which selects the CRC, a vector of length R perpendicular to the programmed path is calculated.

- Outside contours (the angle between blocks N10 and N11 is less than 270° and greater than 180°).



- Outside contours (the angle between blocks N10 and N11 is greater than 270°).

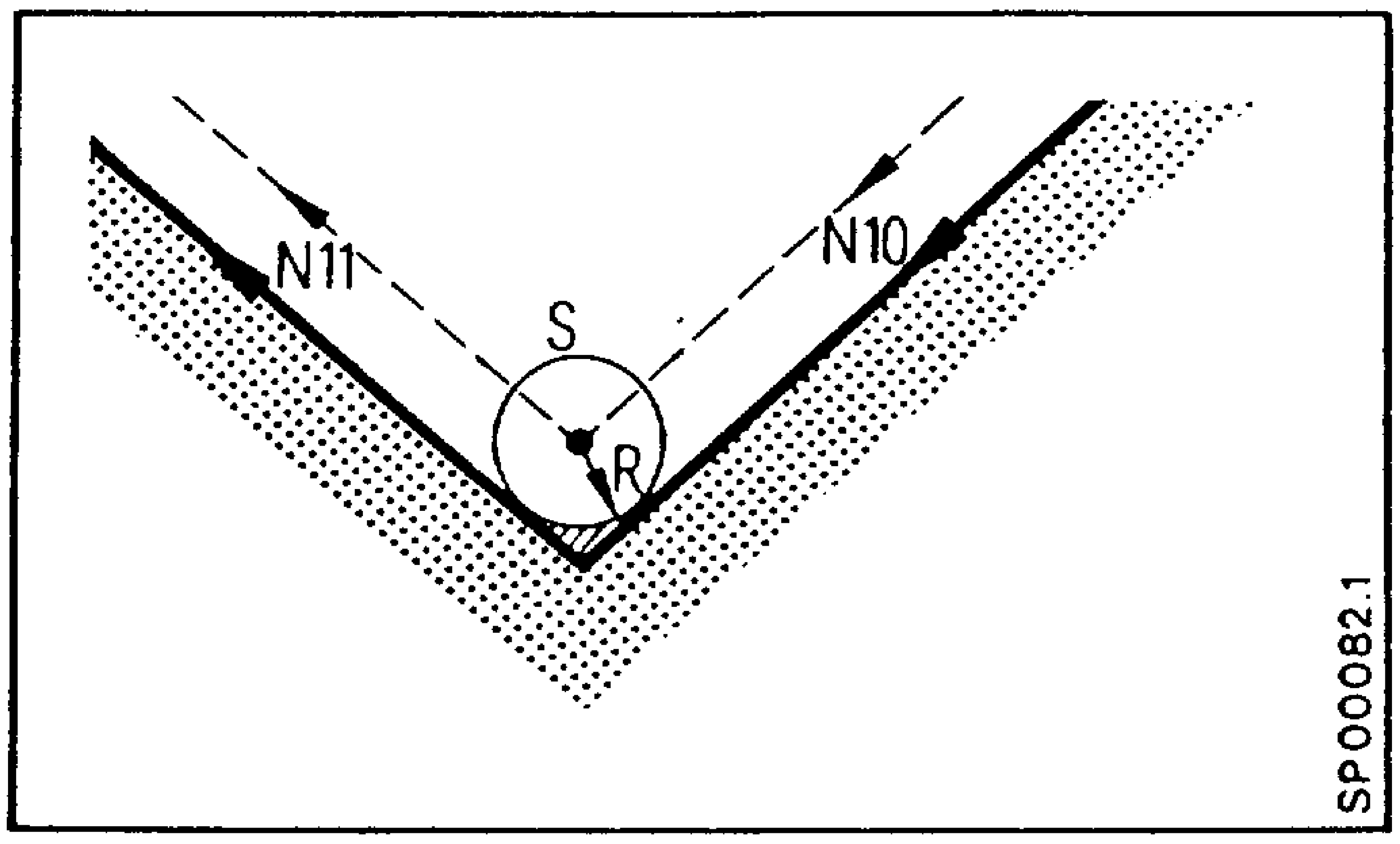


8.1.2 CRC in the program

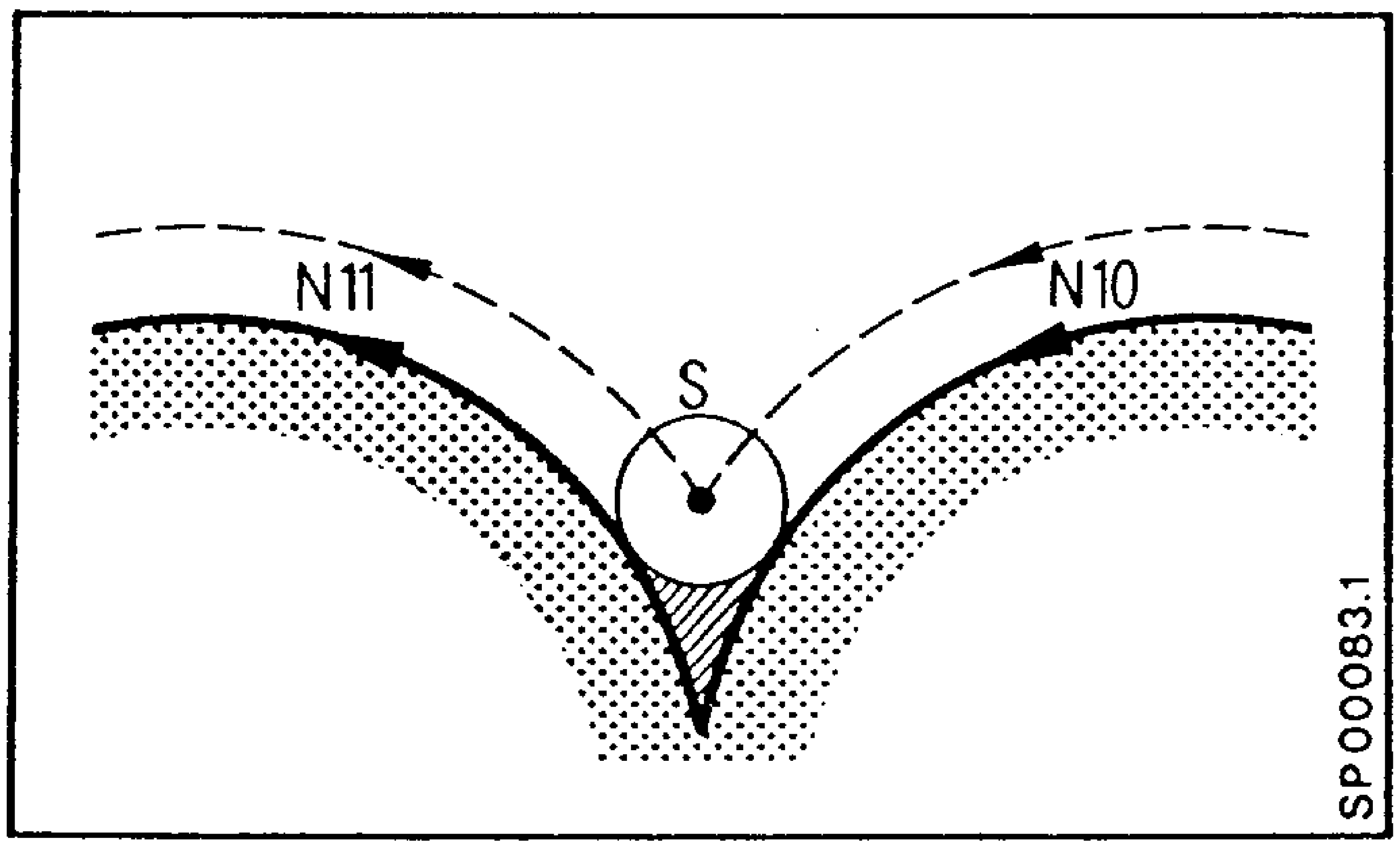
- Inside contour (the angle between two blocks is less than 180°)

Straight line - Straight line

The intersection of the cutter compensation path is calculated.



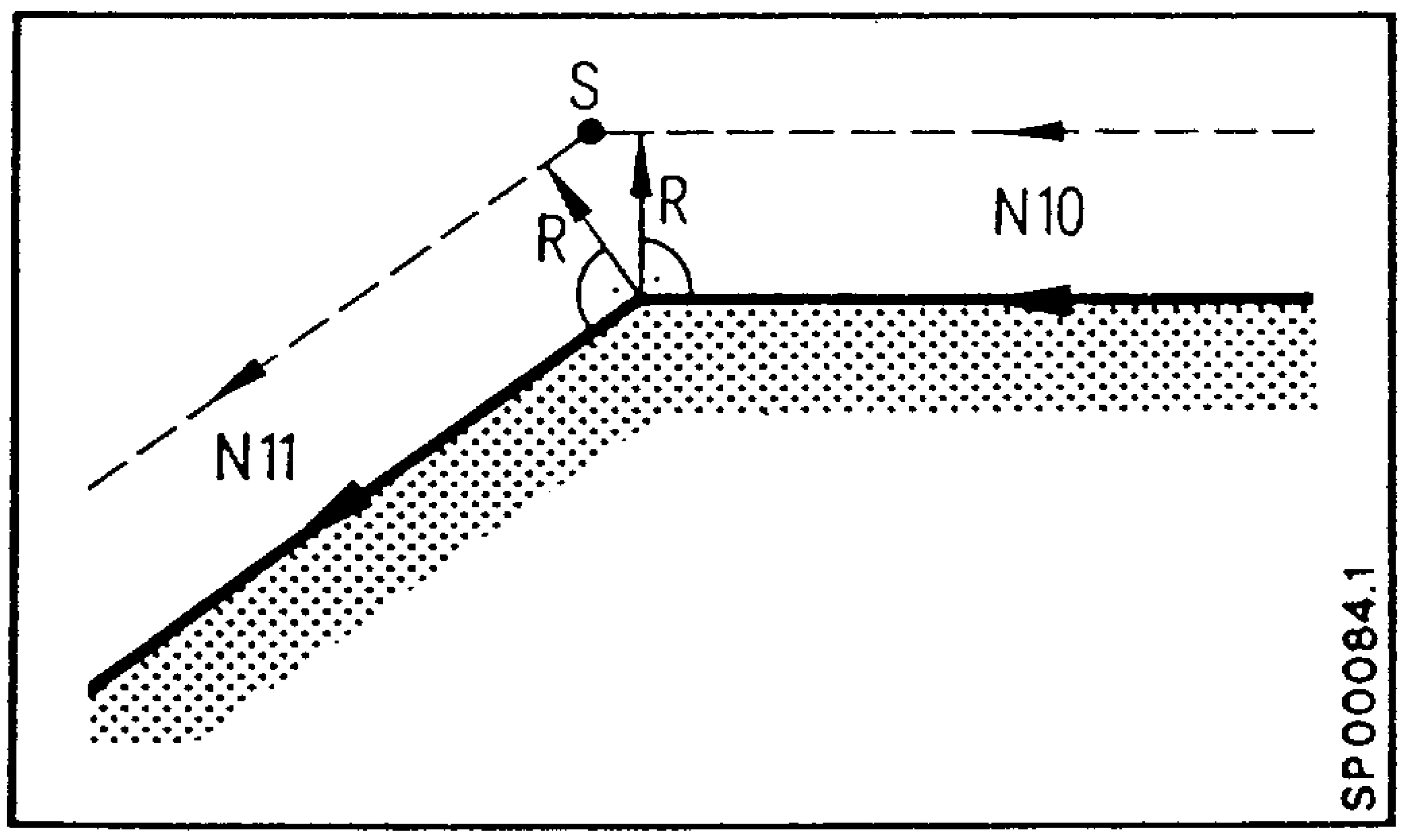
Circle - circle



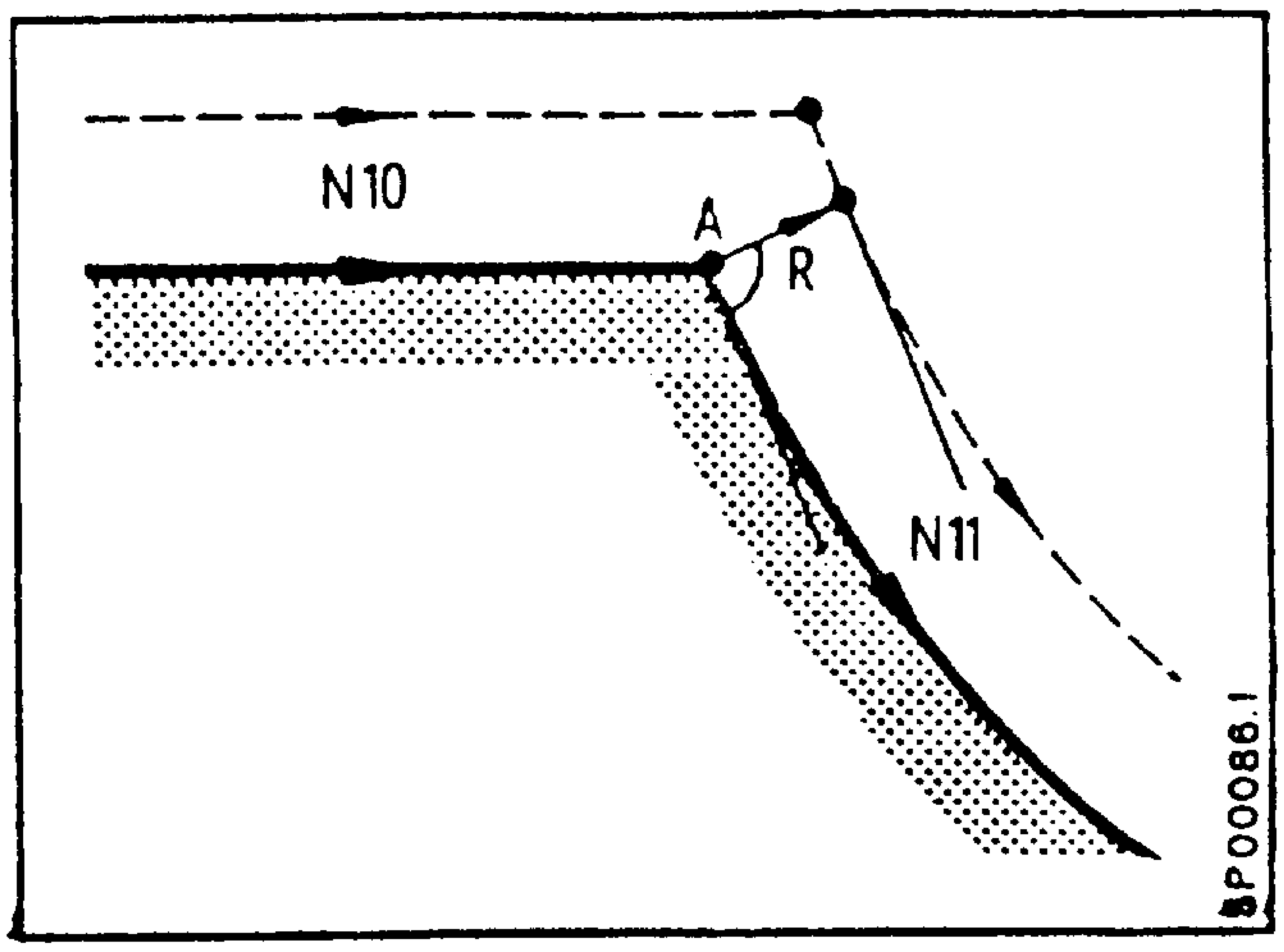
- Outer contour (the angle between two blocks is less than 270° and greater than 180°)

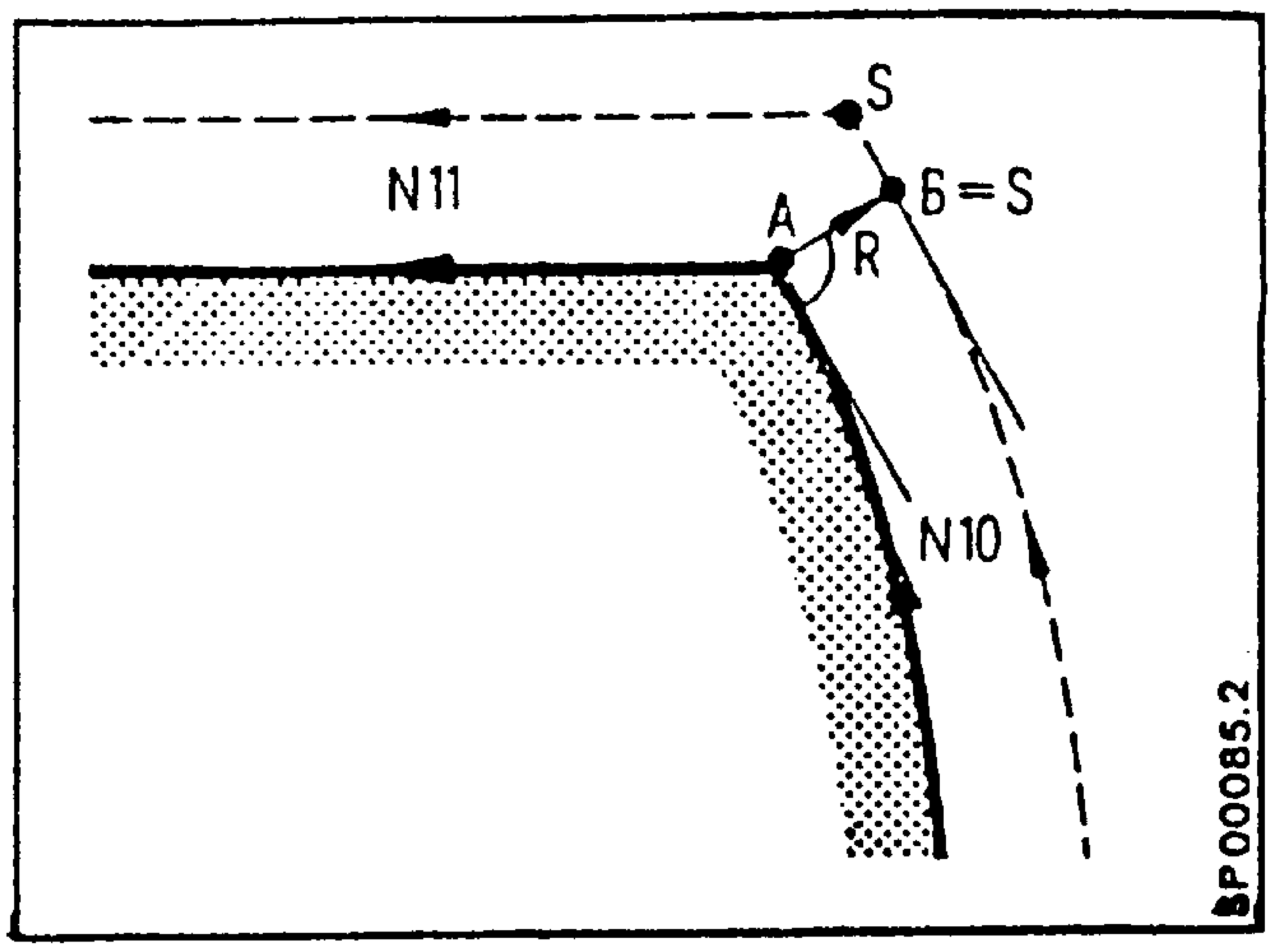
Staight line - straight line

The intersection of the cutter compensation path is calculated.



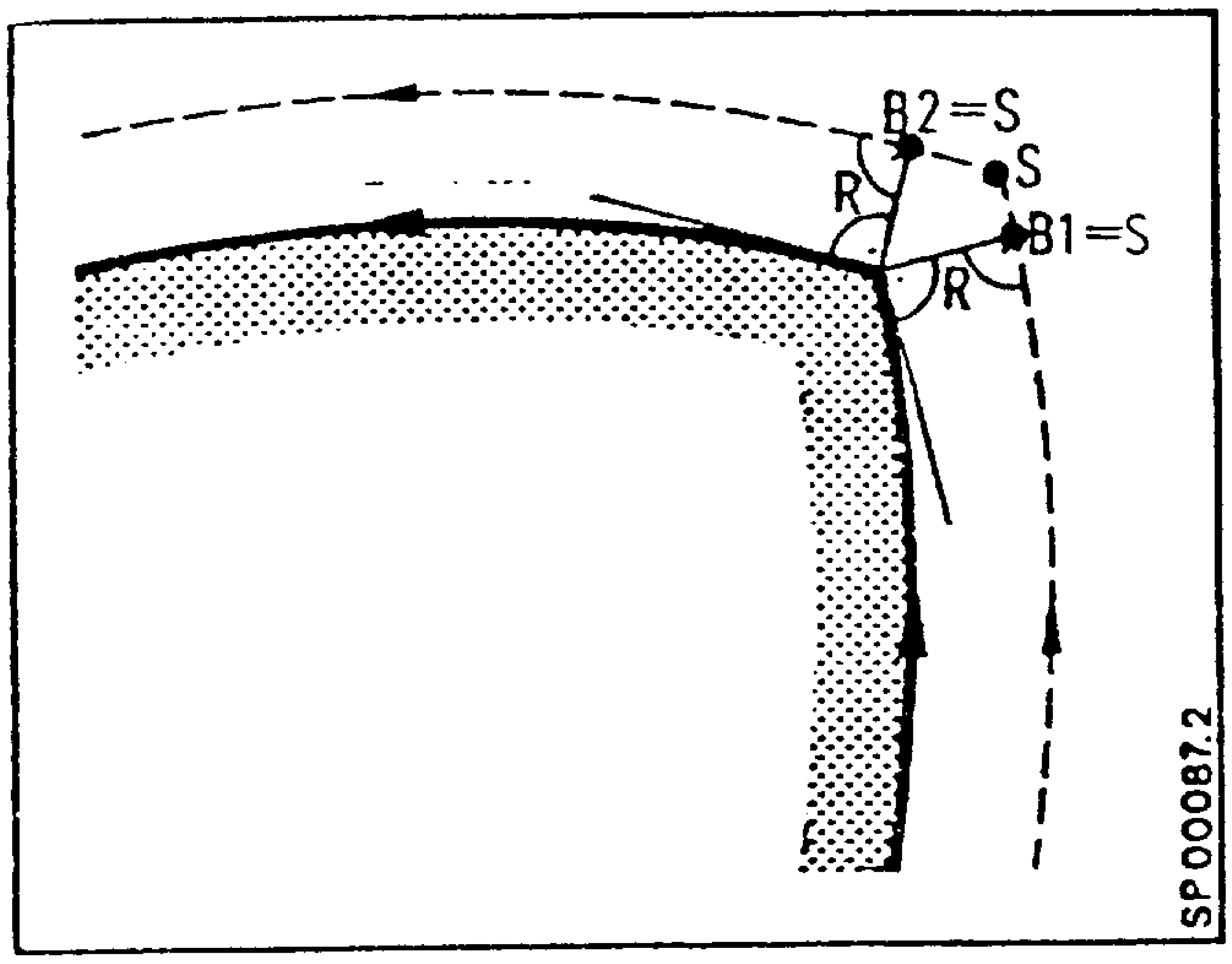
Straight line - circle





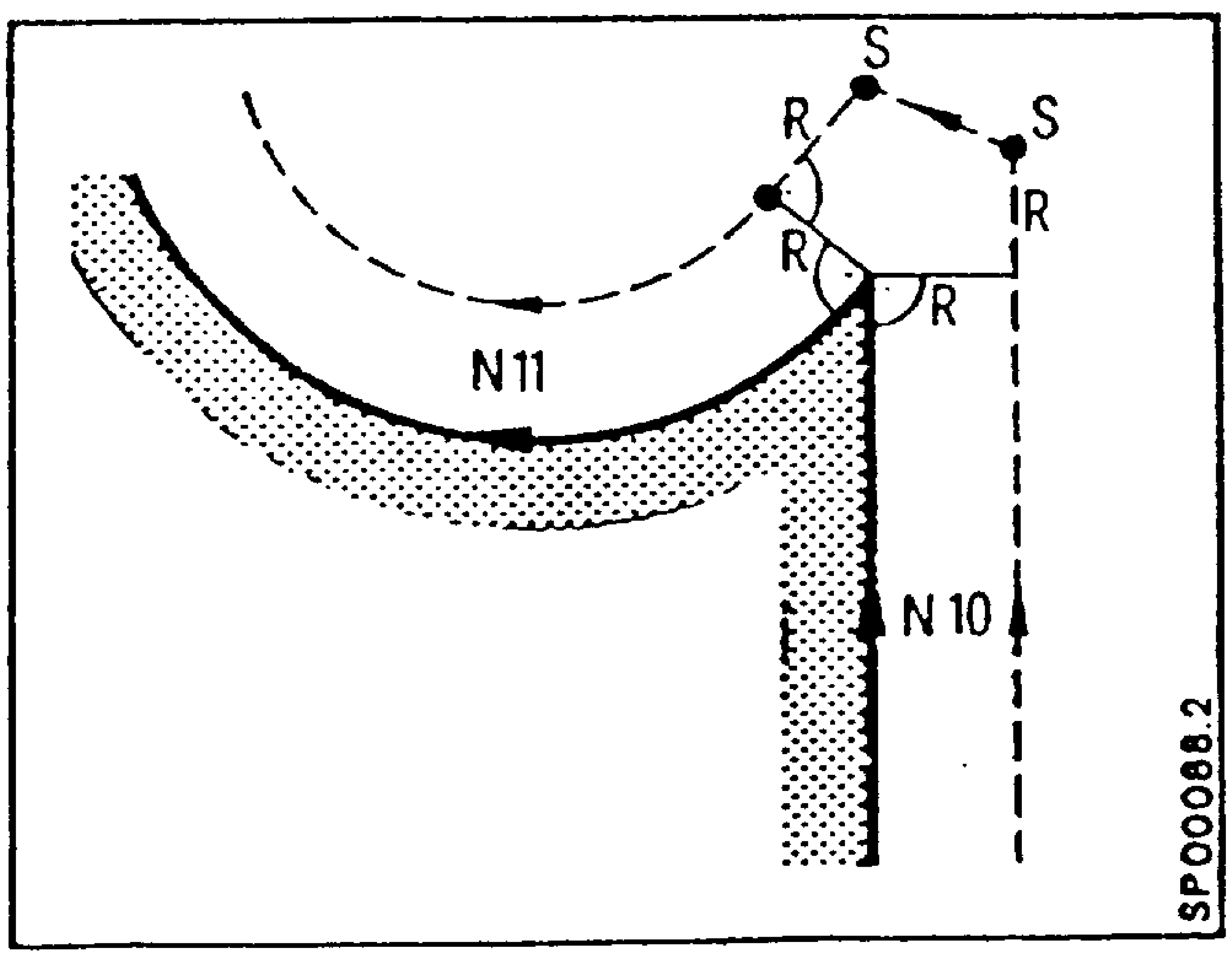
At the arc end point A (or arc start point), a normal vector of length R is calculated. The intersection point is calculated from the tangent at point B and the cutter compensated path of N11 (or N10).

Circle - circle



At the arc end point (or arc start point), a normal vector of length R is created for both arcs. The tangent to point B1 - B2 is determined and the tangent - tangent intersection calculated.

Straight line - circle (the angle is greater than 270°)

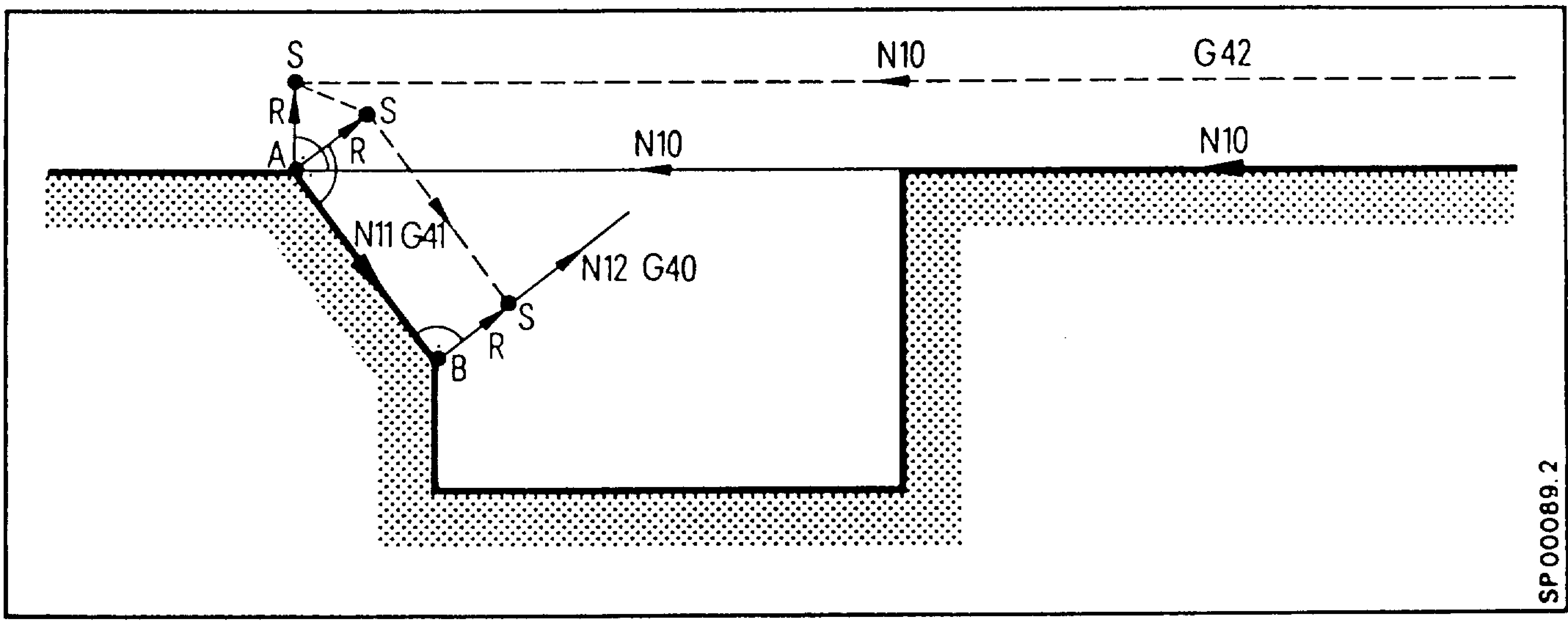


At the end point and start point respectively of blocks N10 and N11, a normal vector of length R is calculated for each path. A compensating movement is made, whereby the distance of radius R is traversed past the programmed contour to prevent damage to the workpiece.

Changing the cutter compensation direction

At the block end point the old compensation direction (G41/G42) is changed to start the next block (G41/G42). The compensation direction is switched in the following manner:

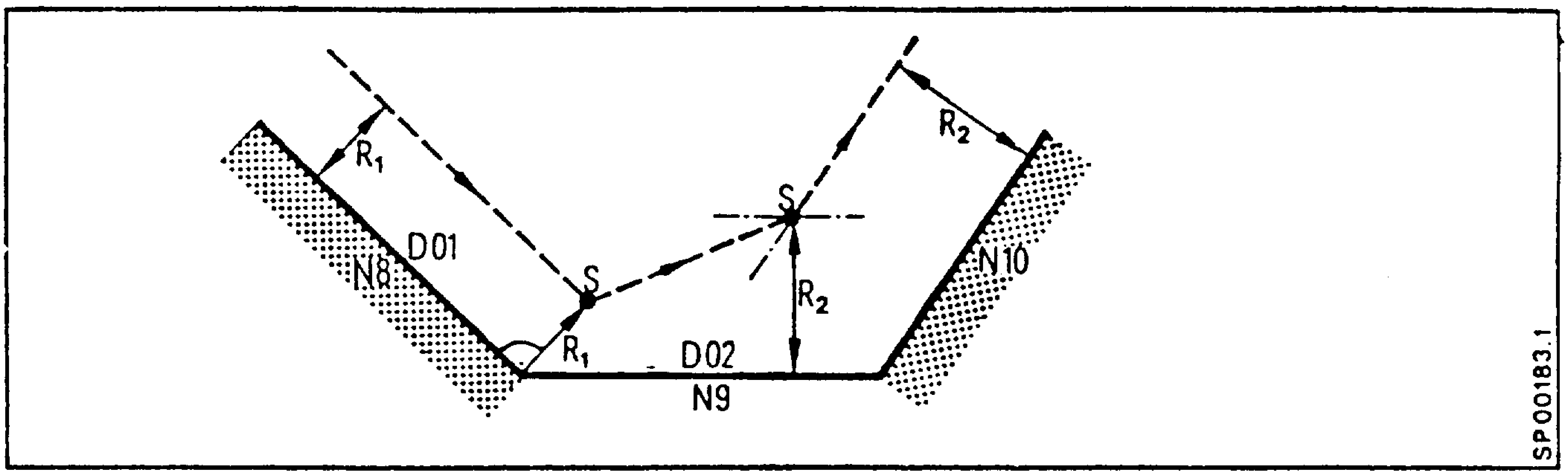
Normal vectors of length R are calculated at the end point and start point of the new blocks respectively.



Changing the offset number (G41 D.., G41 D..)

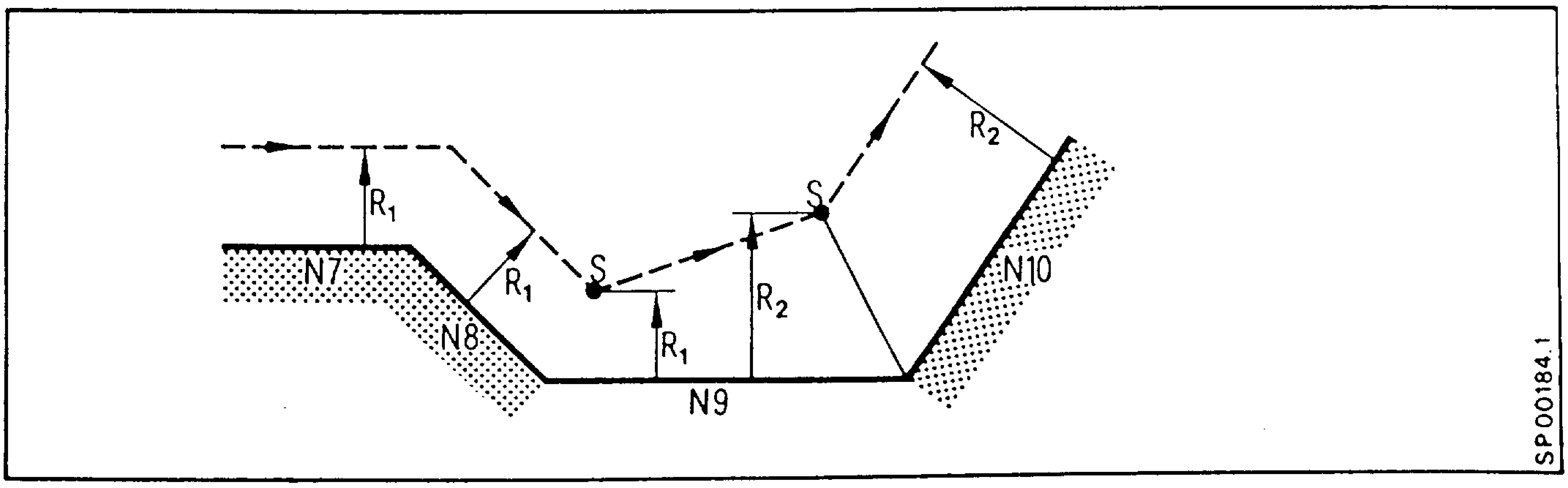
When changing the offset number, the following applies:

No block start intersection is calculated with the old offset value.
At the end point of the block with the old offset number, a normal vector of length R1 is calculated.
The end point of the block is calculated with the new offset.



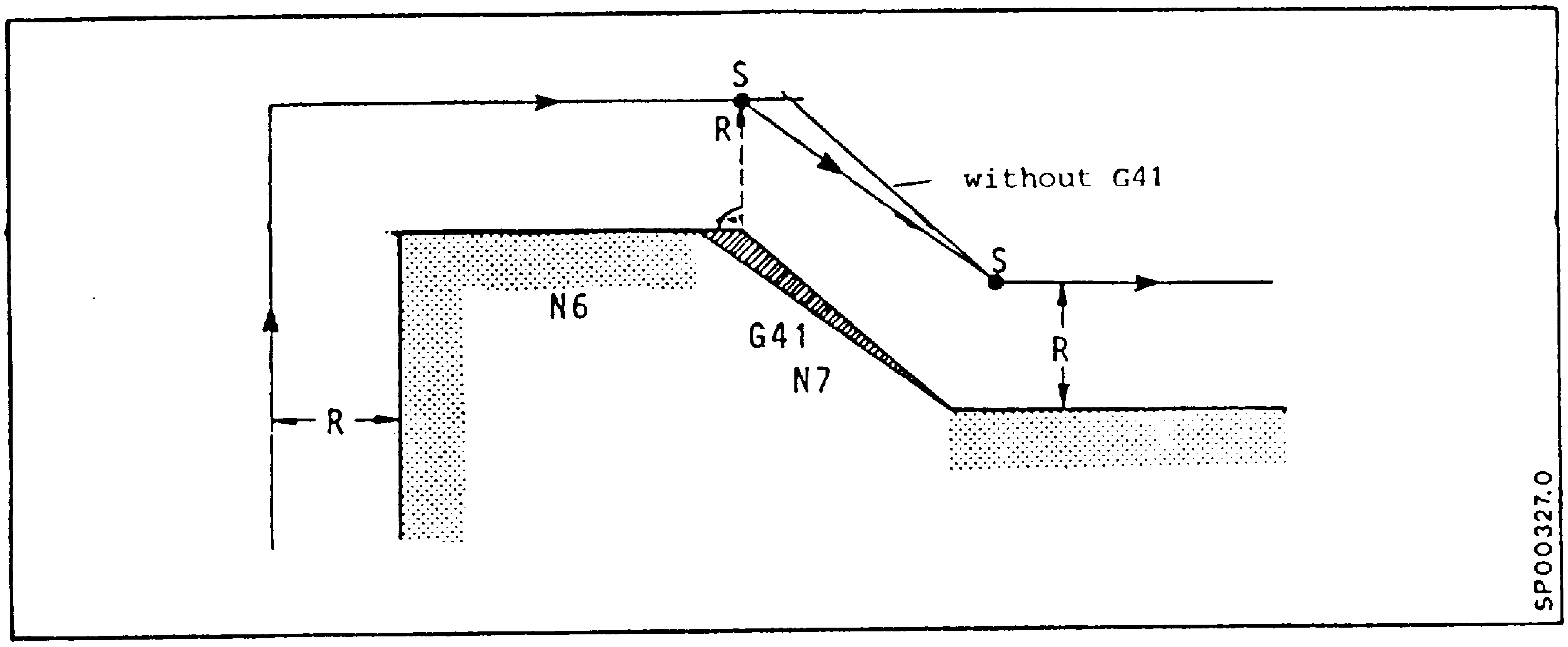
Changing the offset values

The offset values can be changed via the operator panel, punched tape, external tool offset or in the parts program.
The new offset value is effective in the next block.



8.1.3 Repetition of G function already selected (G41, G42) with same offset number

If already programmed G41, G42 is repeated, a normal vector of length R is erected for the programmed path at the end point of the previous block.



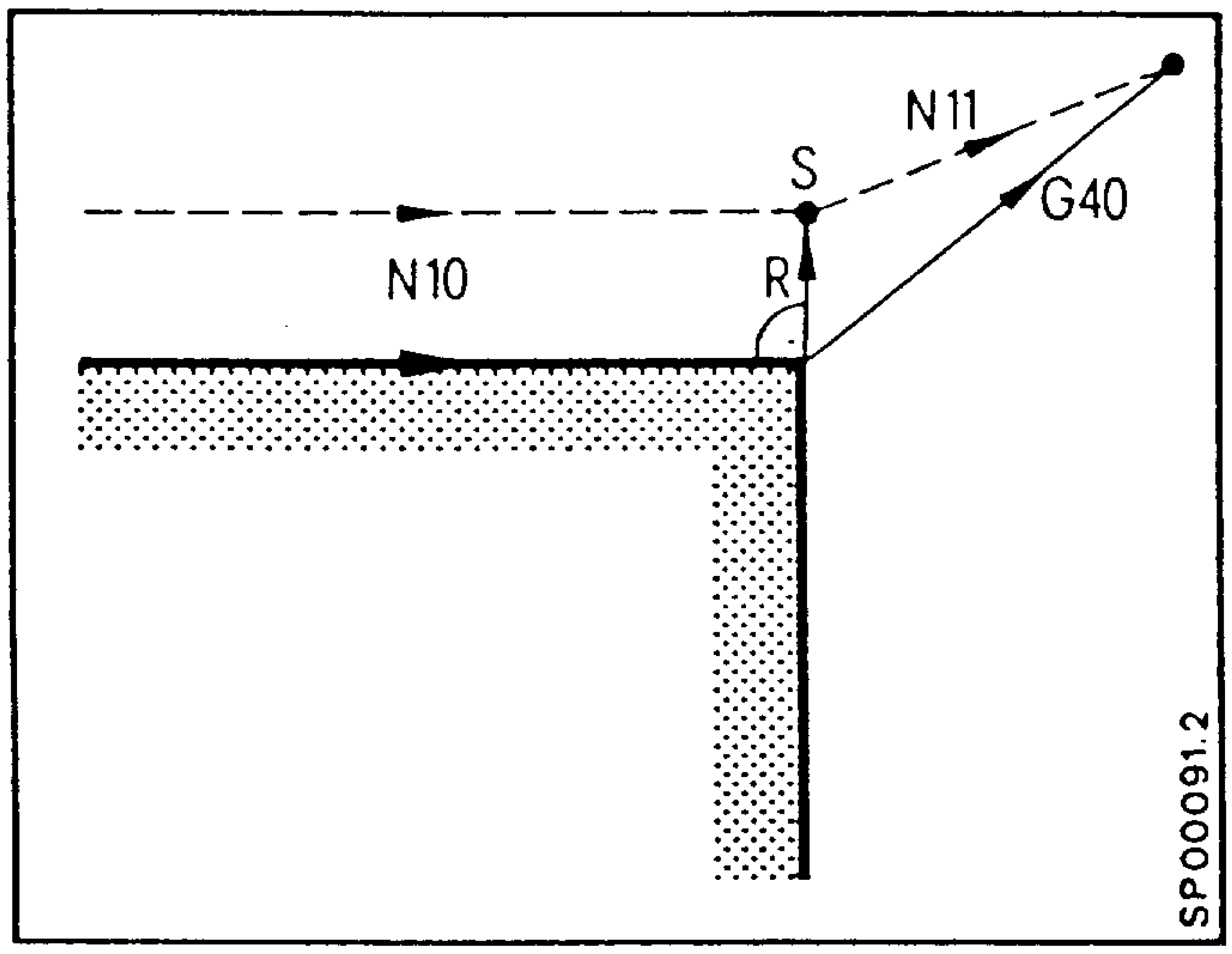
For the following block, the block start intersection is calculated:

N4	G91	D10	G41	X....		LF	
N5				X....		LF	
N6				Y....		LF	
N7	G41			X....	Y....	LF	Error: G41 repeated!
N8				Y....		LF	

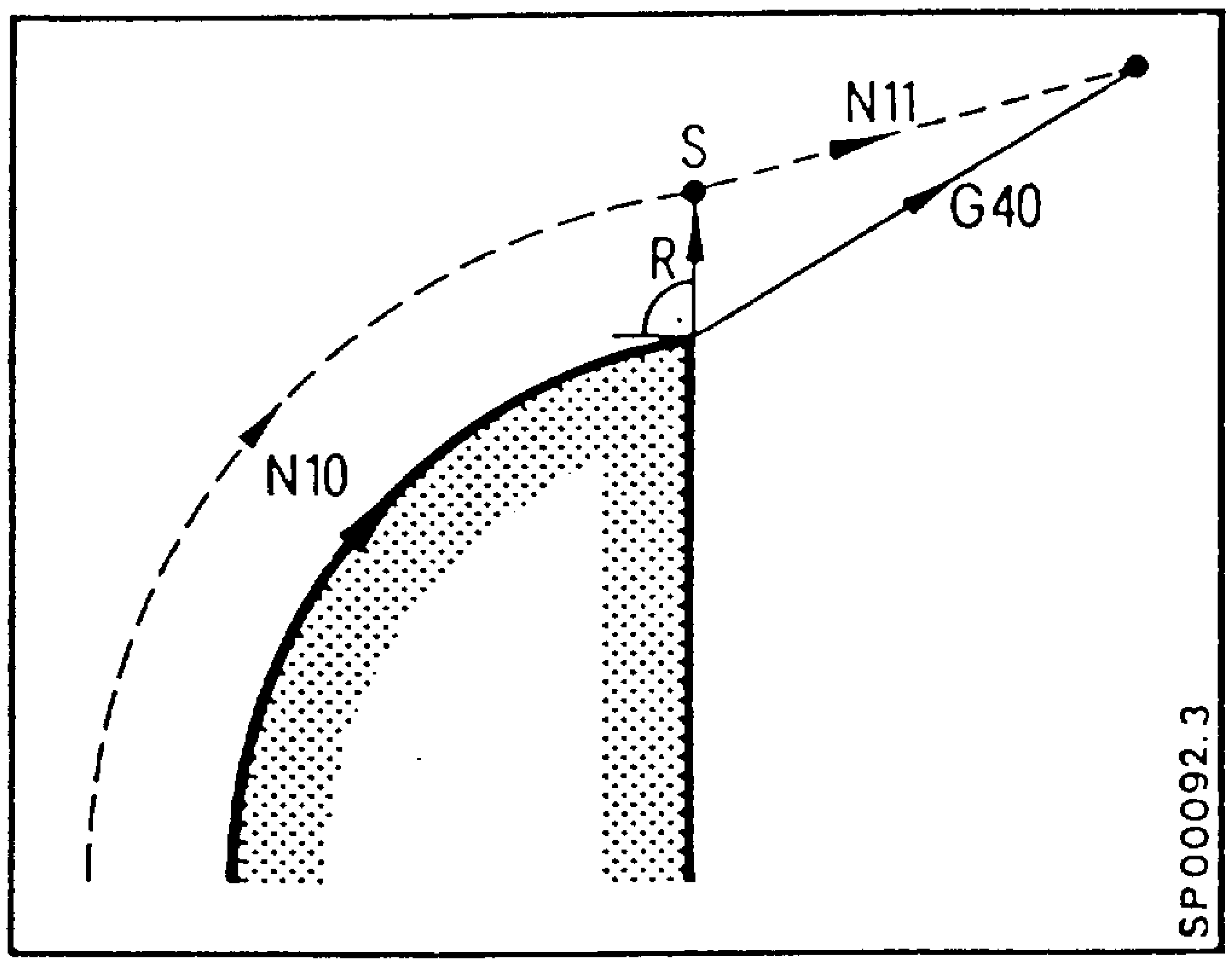
8.1.4 Cutter radius compensation deselection

- Inside contour (the angle between blocks N10 and N11 is less than 180°).

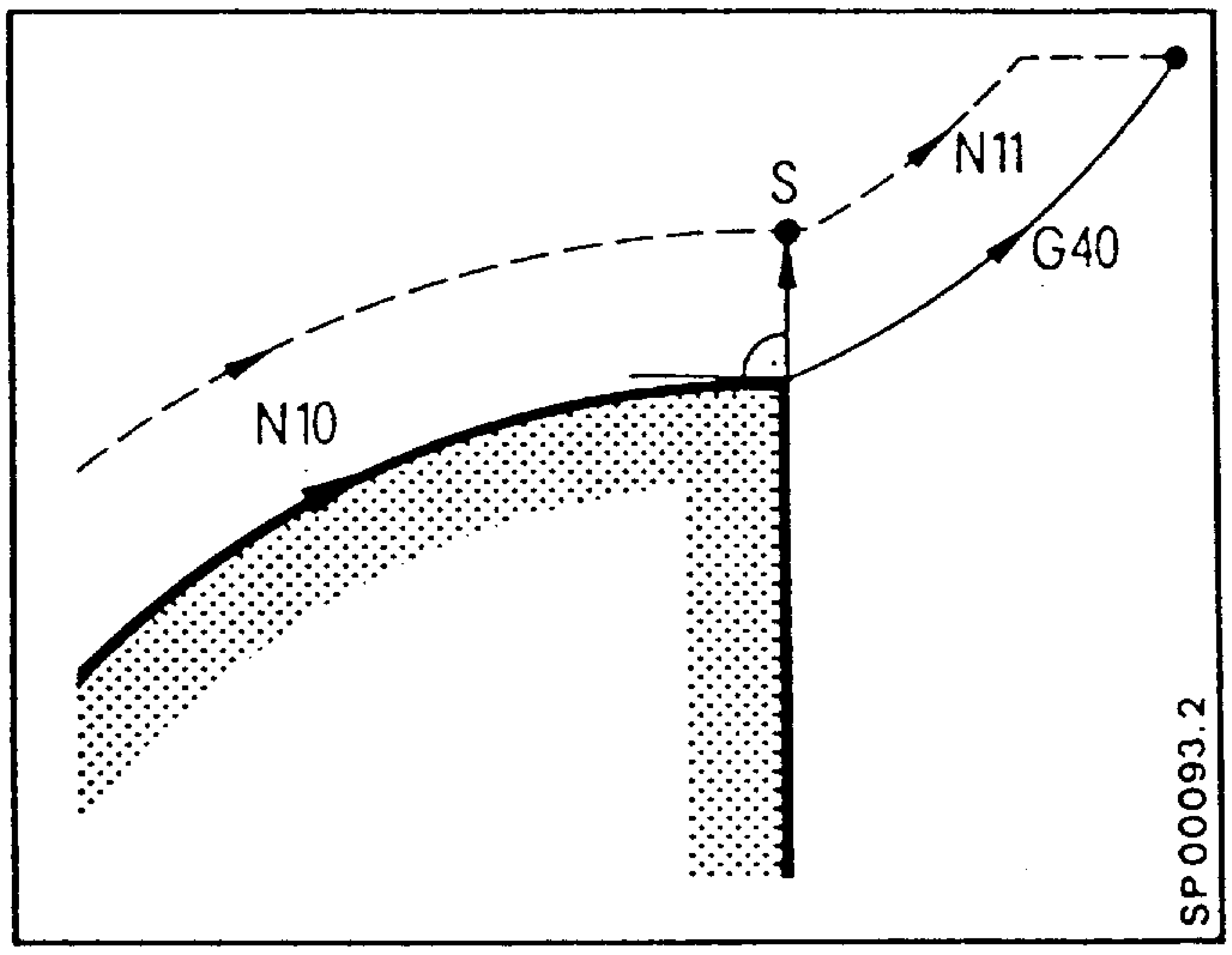
Straight line - straight line



Circle - straight line



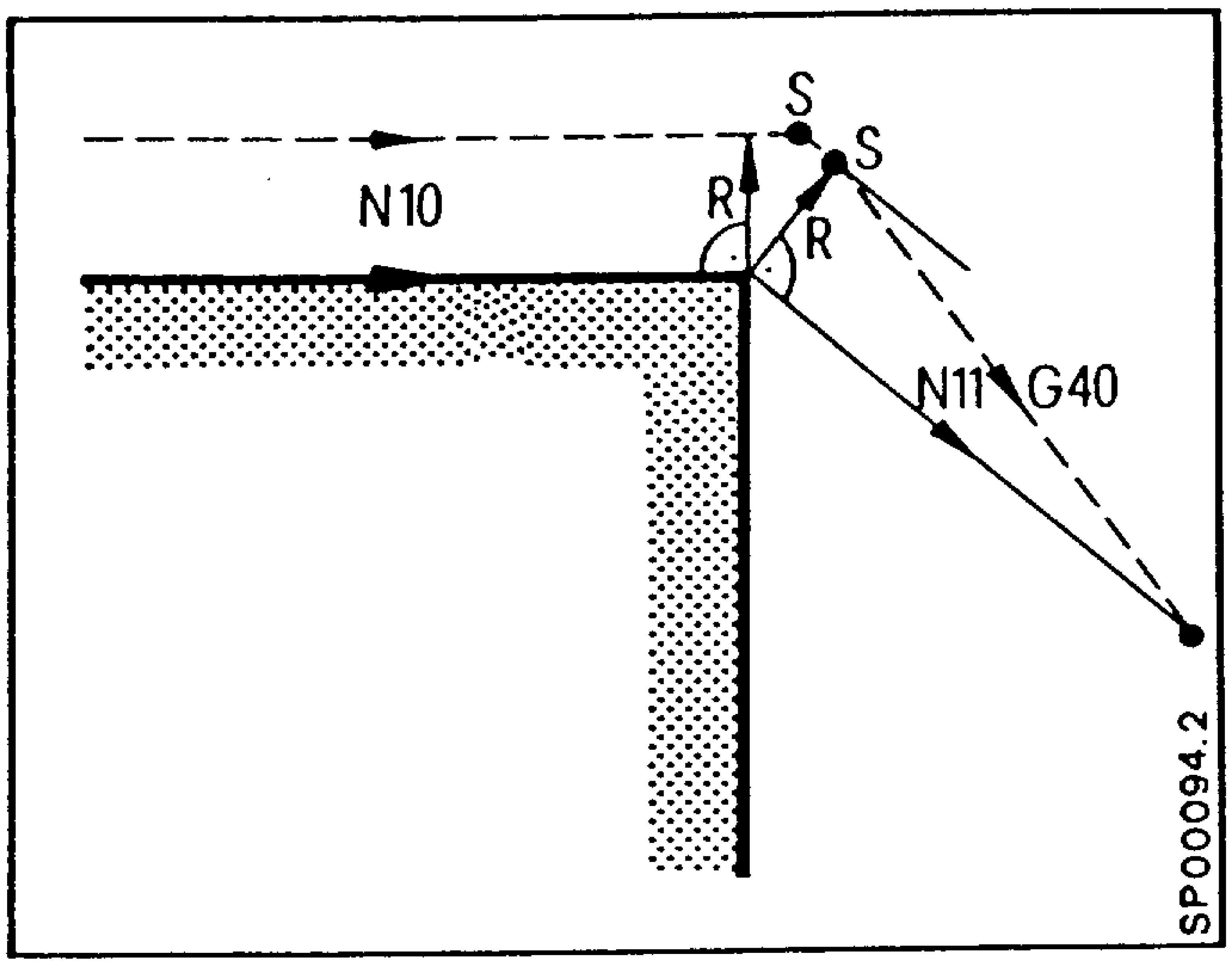
Circle - circle



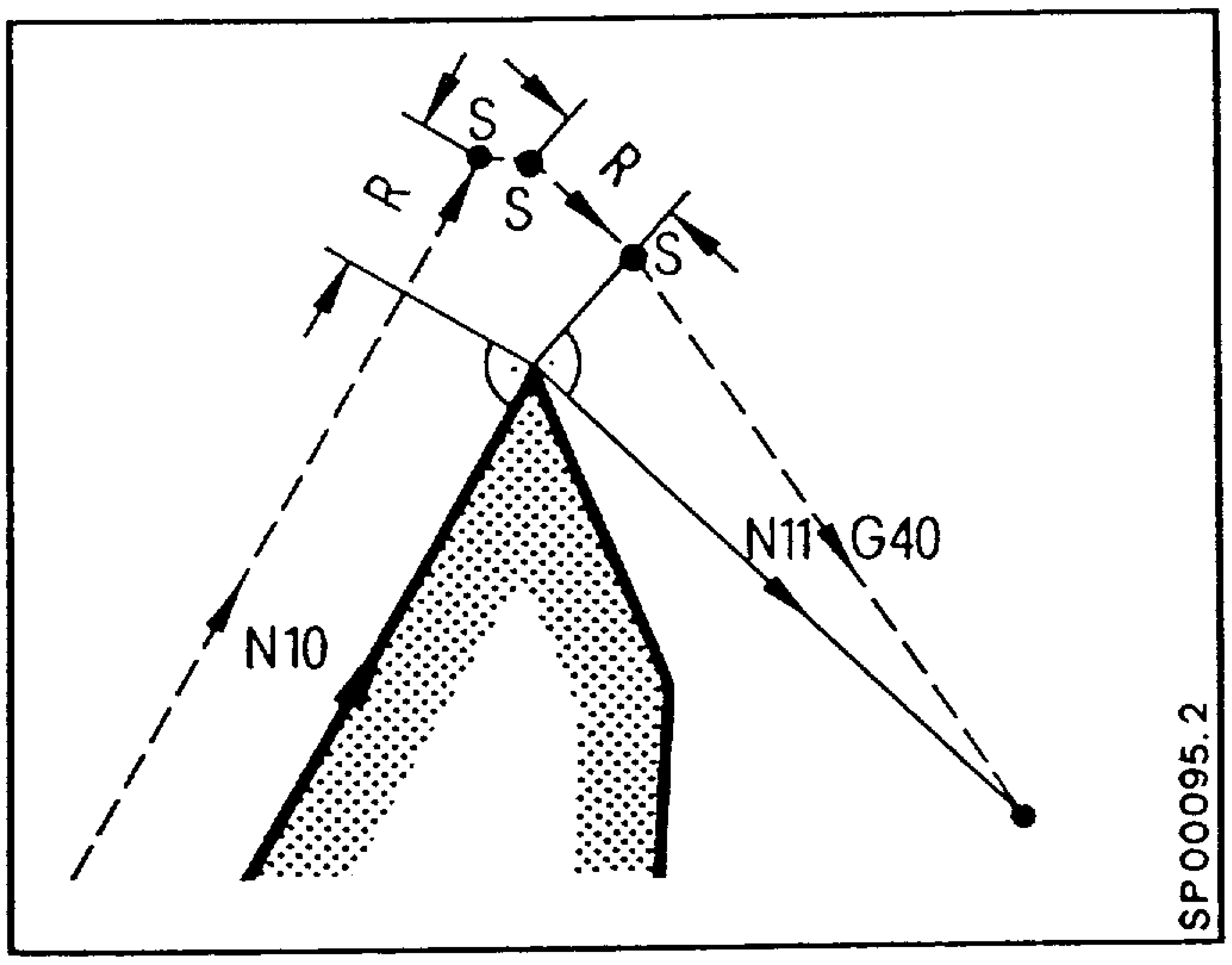
The last block in which CRC is active, a normal vector of length R is erected for the programmed path.

When a transition is made to a linear path, the programmed end point is approached directly.
When a transition is made to a circular path, a displaced arc path is traversed to the horizontal (or vertical) intersection point. The remaining distance is traversed along the horizontal (or vertical) to the end point.
If prior to reaching the horizontal (vertical) intersection path an end point coordinate is reached, the other coordinate is approached directly (see figure above).

- Outside contours (the angle between blocks N10 and N11 is less than 270° and greater than 180°)



(The angle between blocks N10 and N11 is greater than 270°)



The compensated path is calculated, the tool traverses to the next calculated intersection point of the block in which CRC is cancelled.

8.1.5 M00, M02, and M30 with CRC selected

M00: The NC stops at point S in the single block mode. (The positions are shown in the figures).

M02, M30:

- The CRC is cancelled, if it is deleted with G40, and at least one axis is programmed.

```
N150  X..  Y..        LF
N200  G40  X..  M30   LF
```

- The CRC is not cancelled, if no path is programmed.

Example 1:

```
N150  X..  Y..  LF
N200  G40       LF
N250  M30       LF
```

Example 2:

```
N150  X..  Y..  LF
N200  M30  G40  LF
```

8.1.6 CRC with combination of different block types

The examples refer to the X-Y-plane.

Type:

- Distances in the CRC - plane

  Example:

  N... G91 X1000. LF

- "Distance = 0"

  Preparatory functions are programmed, no movements take place, because the distance is zero.
  Example:

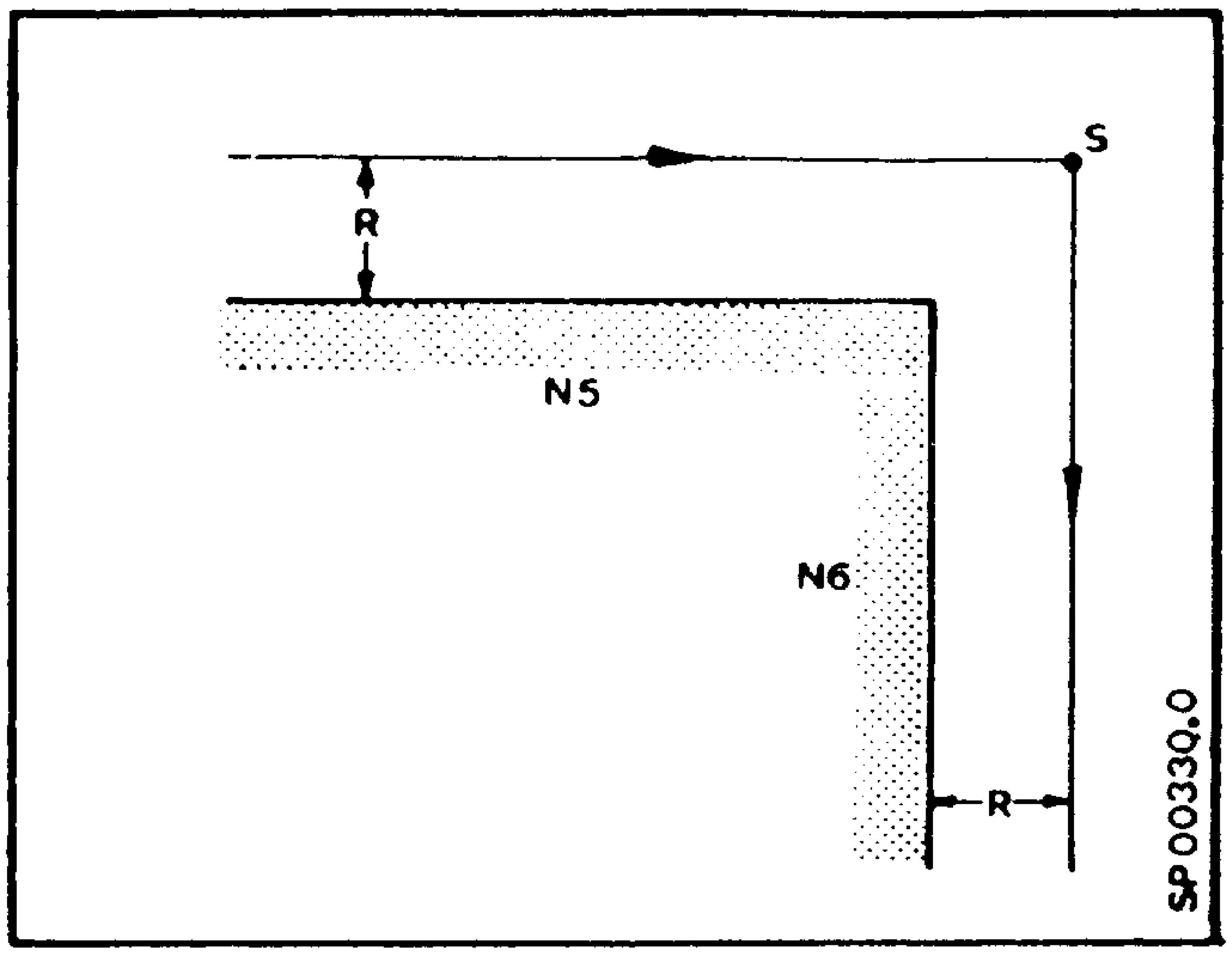
  N... G91 X0. LF

- "Auxiliary functions"

  No preparatory functions of the CRC-plane are programmed; **instead only auxiliary functions, dwell** or a zero offset are in the block.
  Example:

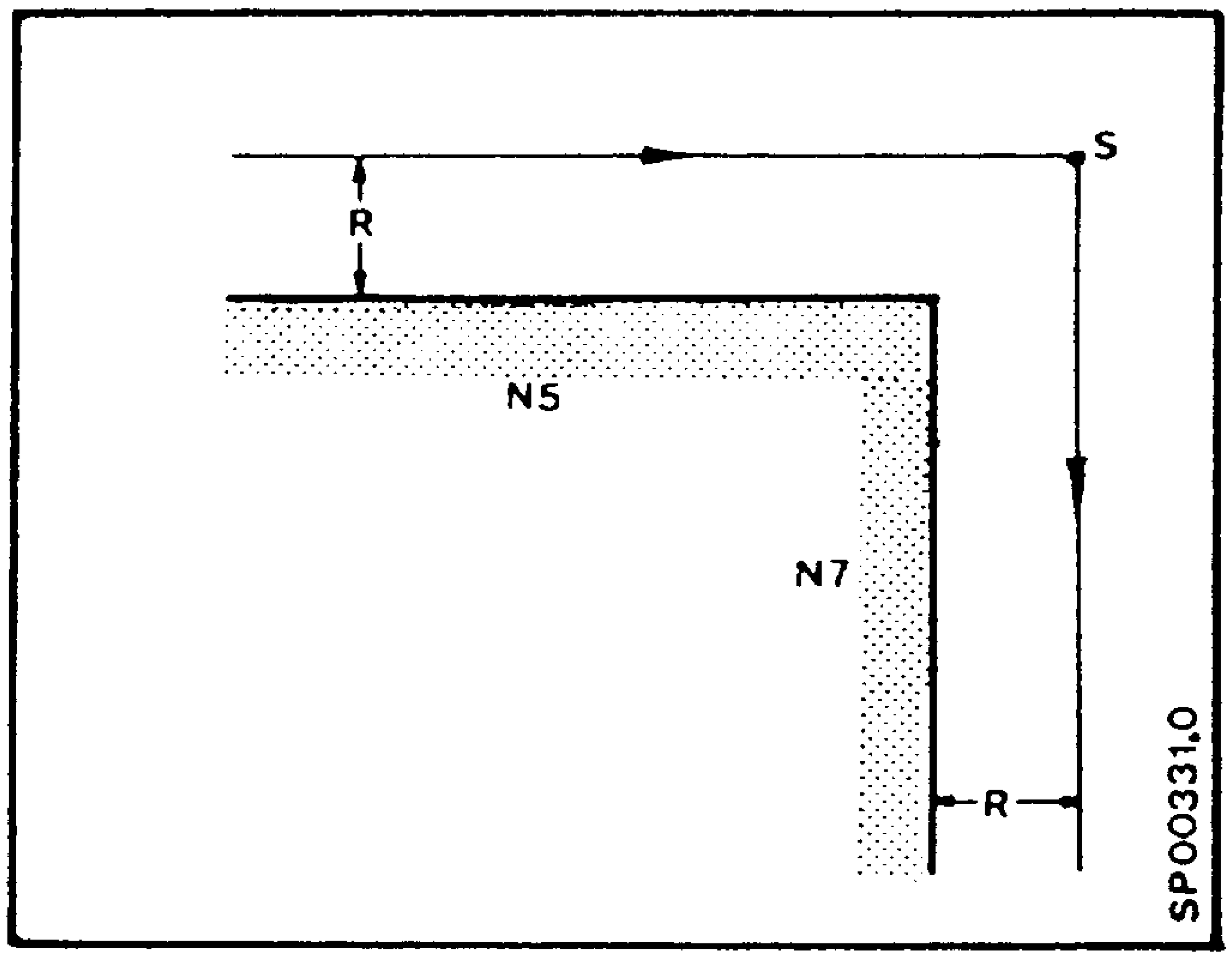
N...	Y1000		LF
N...	M08		LF
N...	G04	F10.	LF
N...	G59	X10.	LF
N...	T0101		LF

Two distances in the CRC-plane




```
N5  G91  X1000.   LF
N6       Y-1000.  LF
```

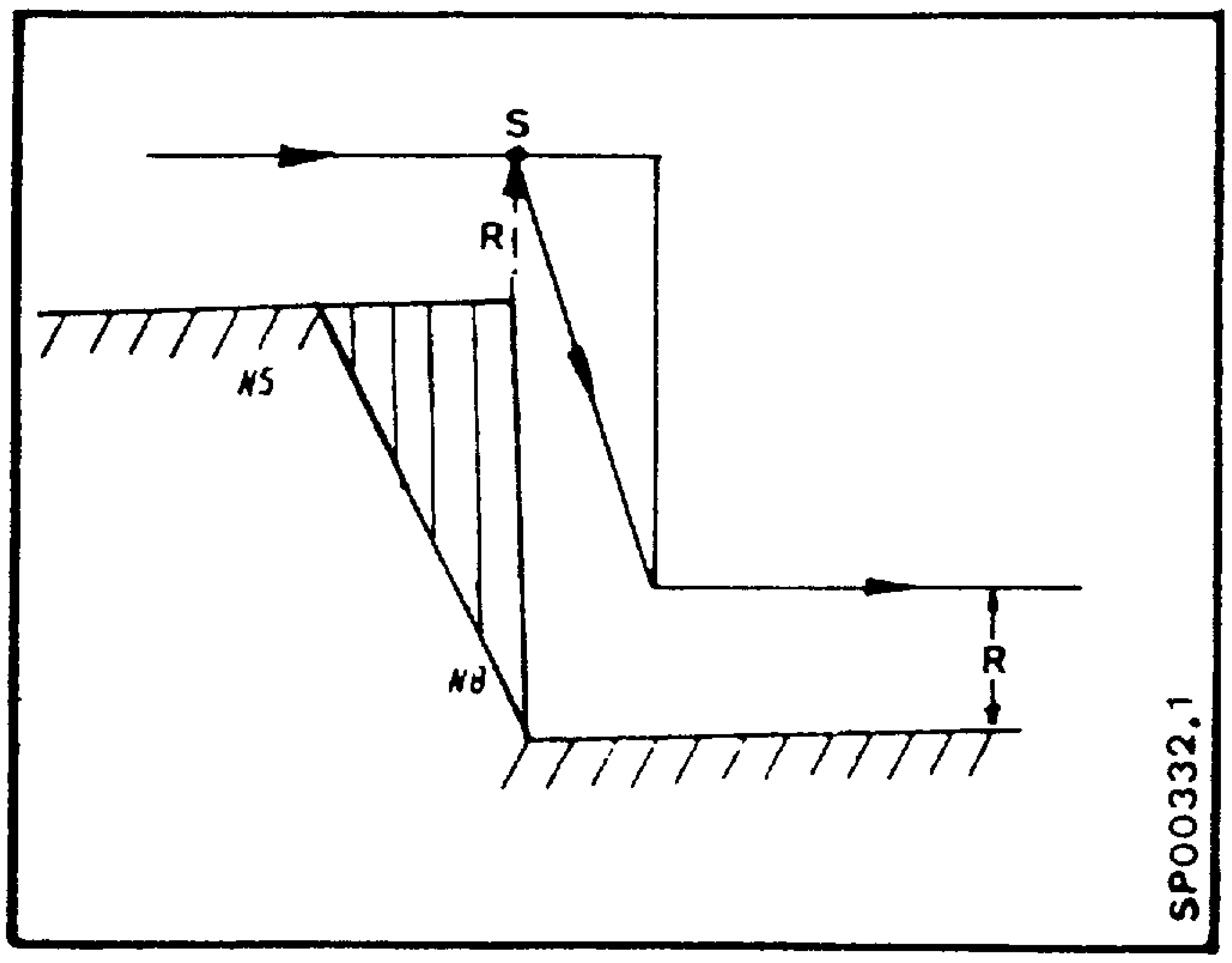
One "aux. function block" between distances in the CRC-plane




```
N5  G91  X10000.  LF
N6  M08           LF
N7       Y-1000.  LF
```

The block N6 is executed at point S.

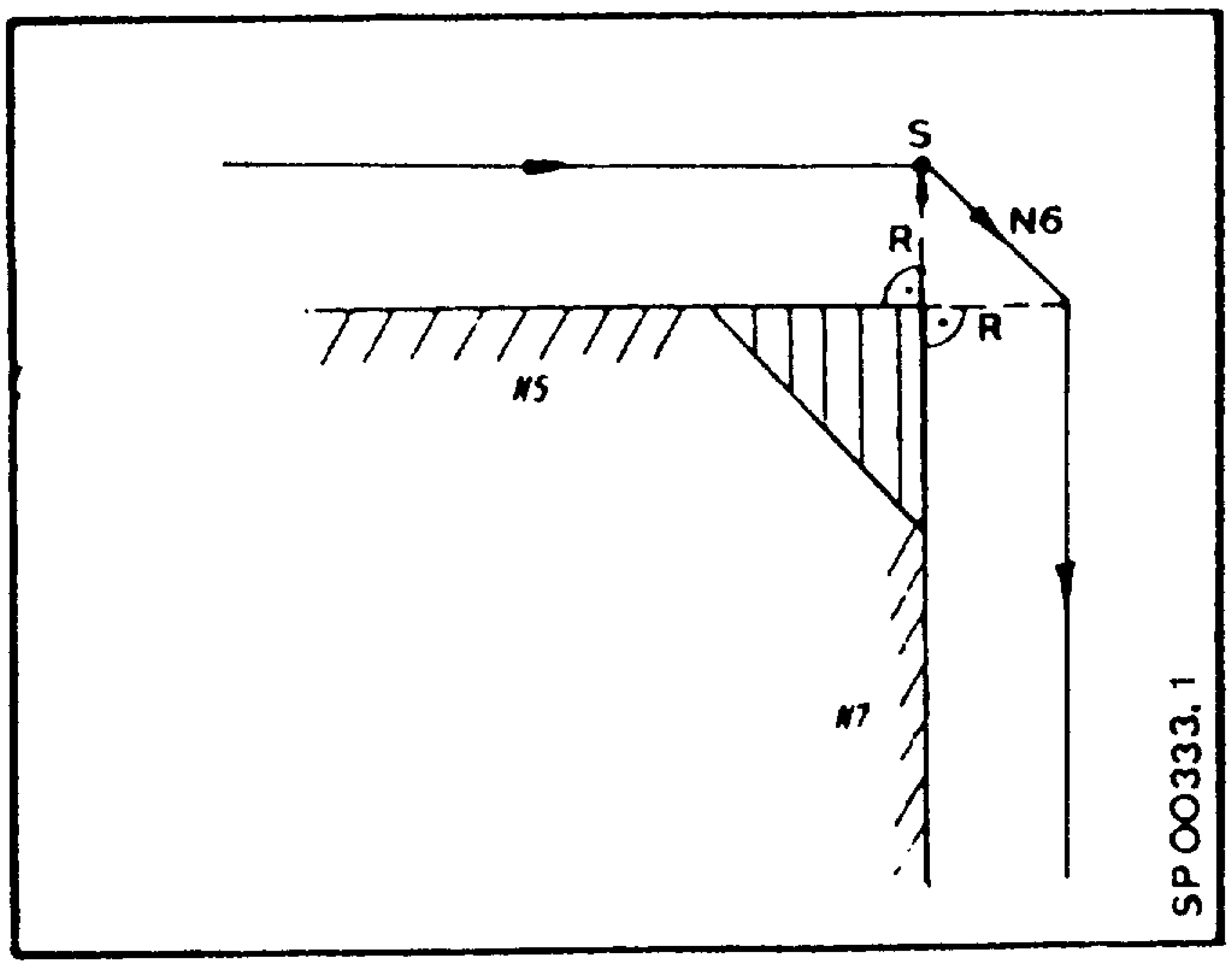
Two "miscellaneous blocks" between distances in the CRC-plane




```
N5  G91  X10000.  LF
N6  M08           LF
N7  M09           LF
N8       X-10000. LF
N9       X1000.   LF
```

The blocks N6 and N7 are executed at point S. With the exception of tangential transitions, contour errors result.

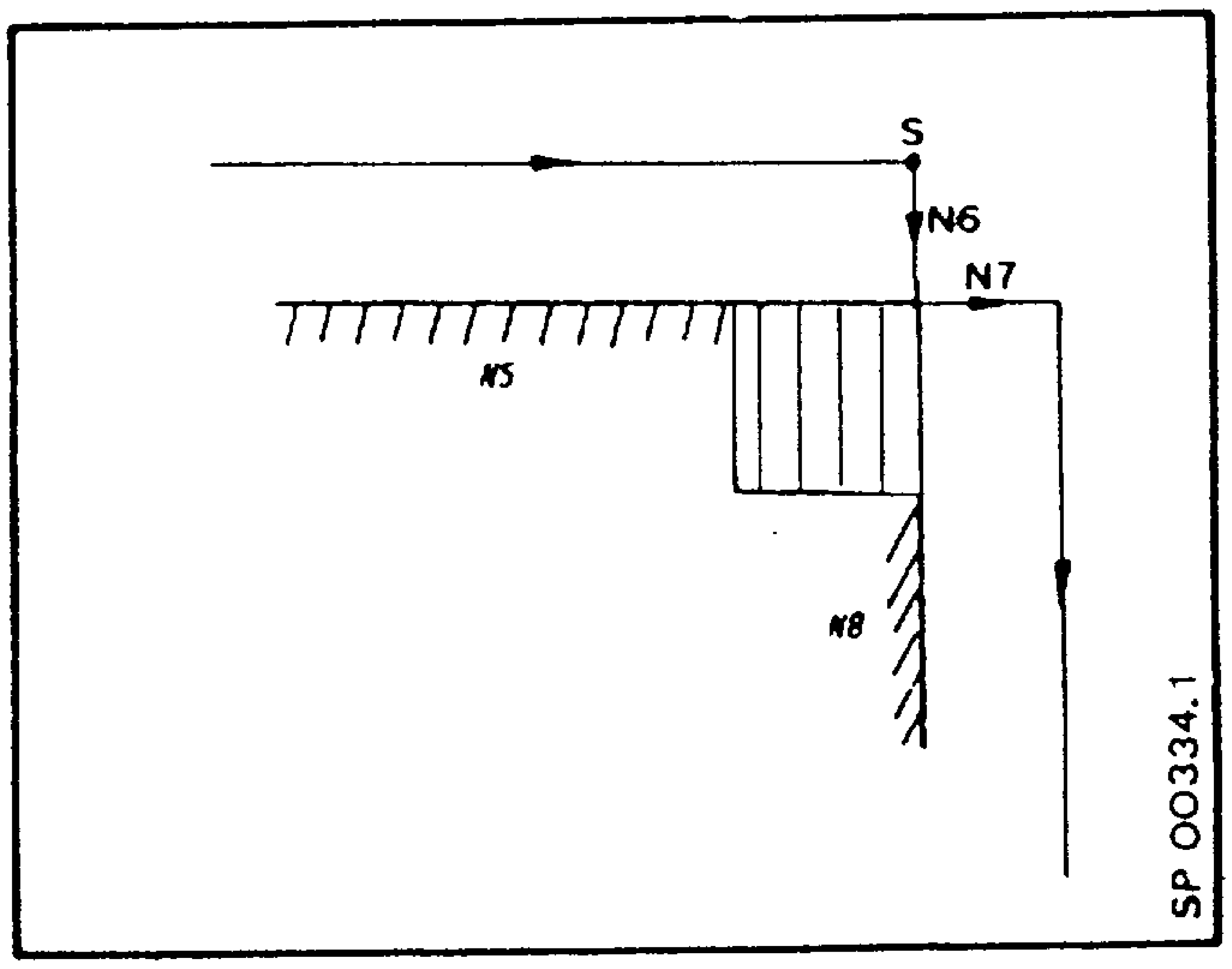
One block "distance = 0" between distances in the CRC-plane




```
N5   G91   X10000.   LF
N6         X0.       LF
N7         Y-1000.   LF
```

With the exception of tangential transitions, contour errors result.

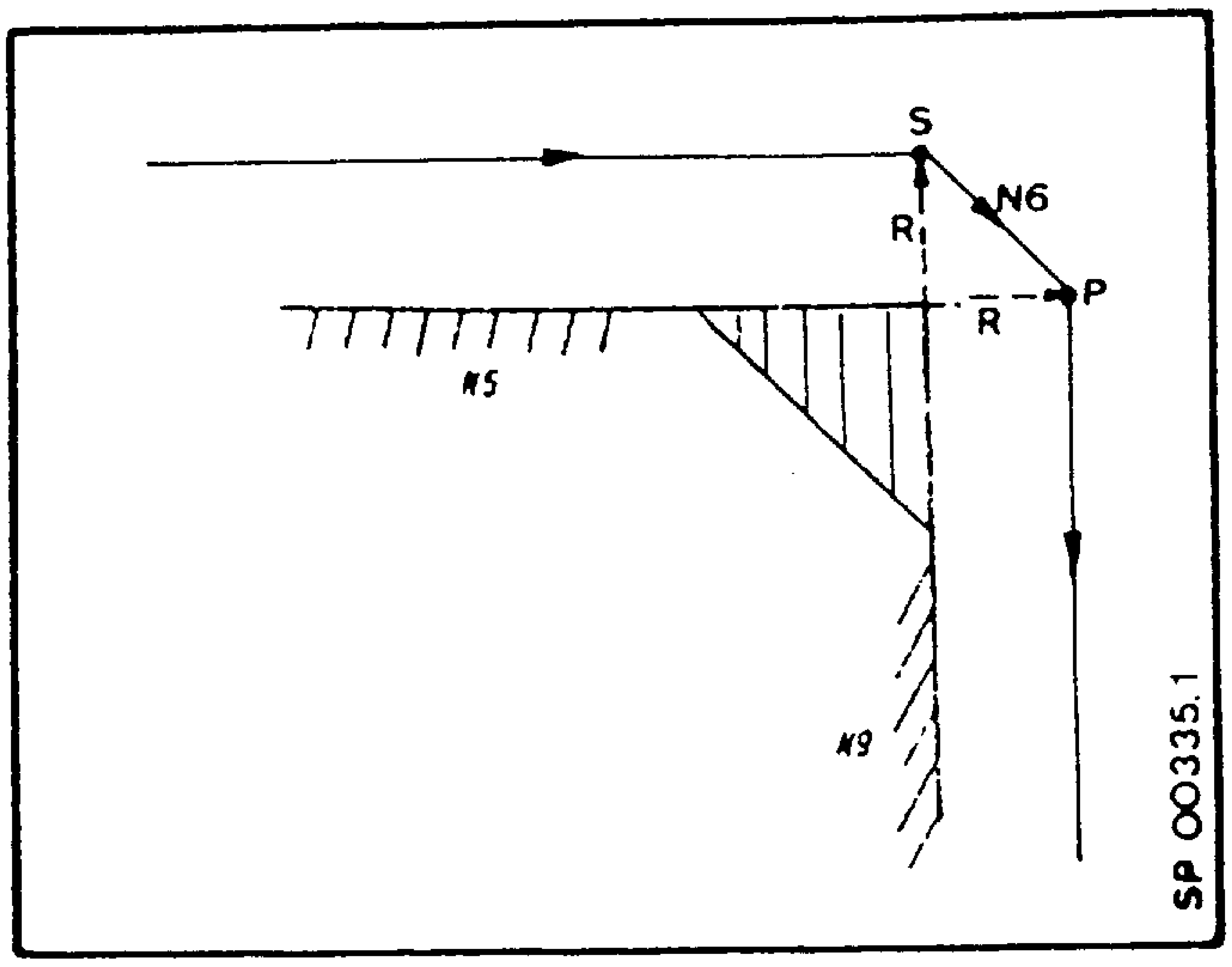
Two blocks "distance = 0" between distances in the CRC-plane




```
N5   G91   X1000.    LF
N6         X0.       LF
N7         X0.       LF
N8         Y-1000.   LF
```

With the exception of tangential transitions, contour errors result.

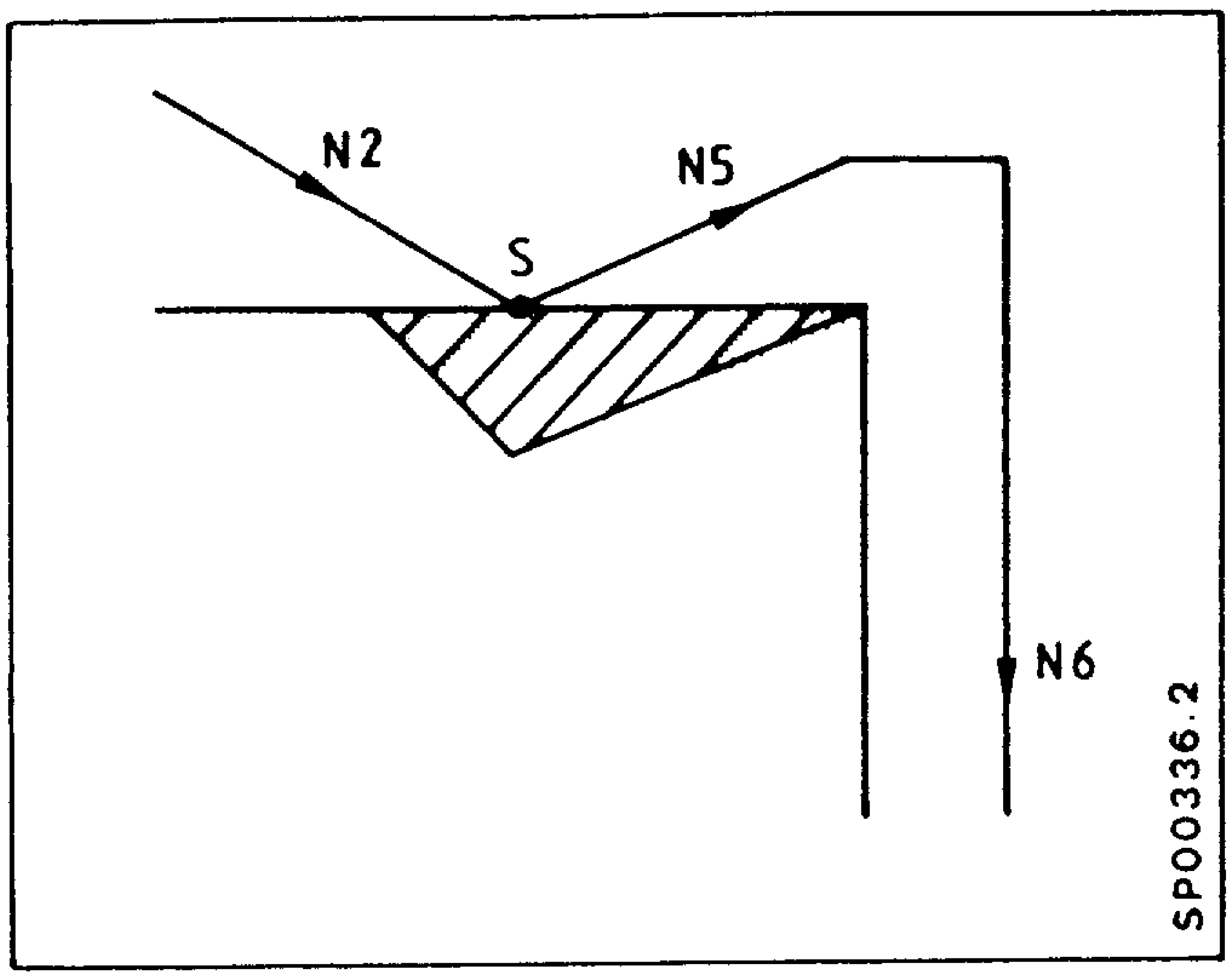
One block "distance = 0" and one "aux. function block" between distances in the CRC-plane




```
N5   G91   X1000.    LF
N6         X0.       LF
N7   M08             LF
N8         Y-1000.   LF
```

With block N7 is executed at point P.
With the exception of tangential transitions, contour errors result.

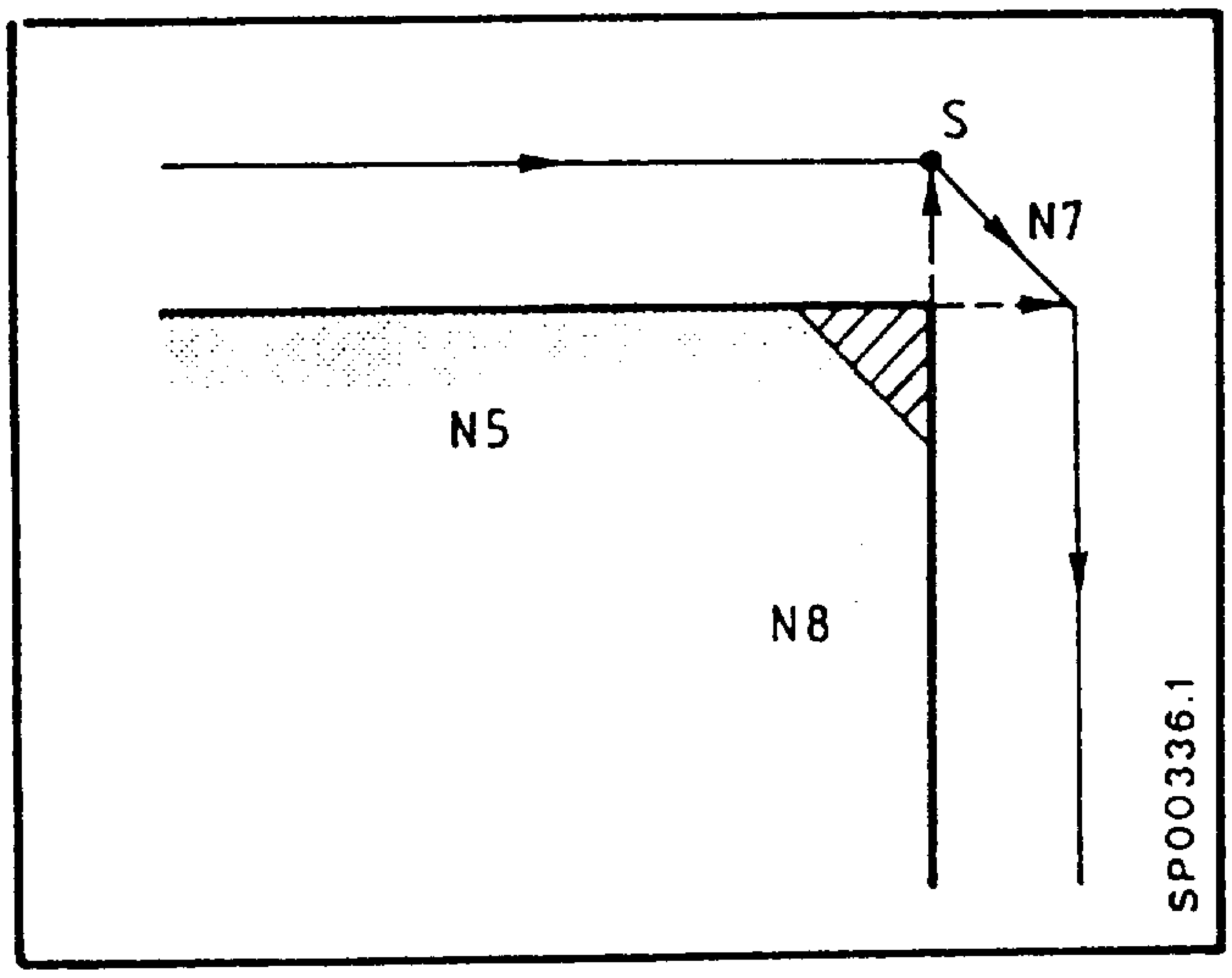
One "aux. function block" and one block "distance = 0" between distances in the CRC-plane



N5 G91 X1000. LF
N6 M08 LF
N7 X0. LF
N8 Y-1000. LF

The block N6 is executed at point S.
With the exception of tangential transitions, contour errors result.

Two blocks between distances in the CRC plane

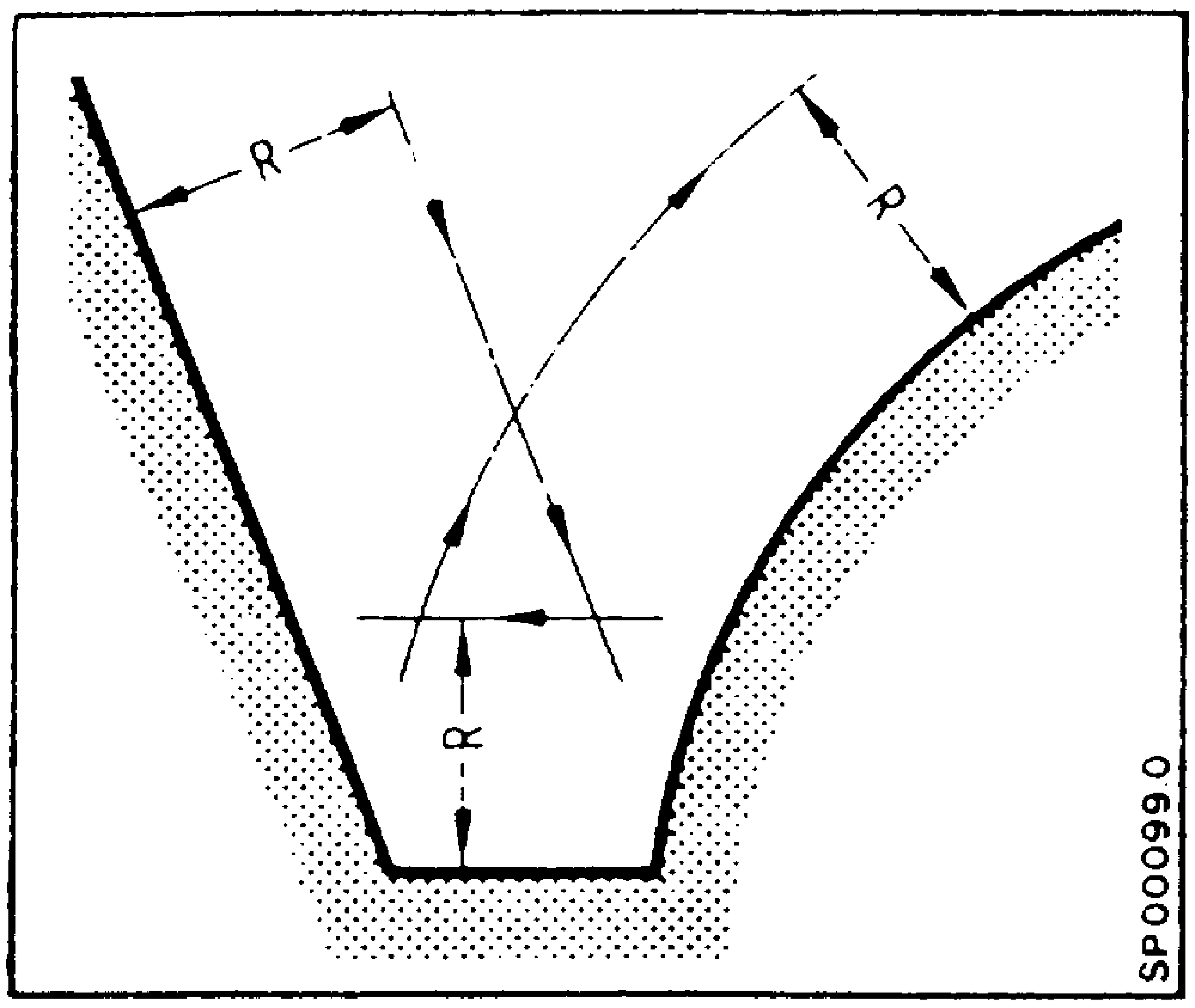


N1 G0 X0 Y0 Z100
N2 G41 X50 Y-50
N3 Z10
N4 Z0
N5 X75
N6 Y-100

The blocks N3 and N4 are executed at point S. Contour errors result, except on tangential transitions.

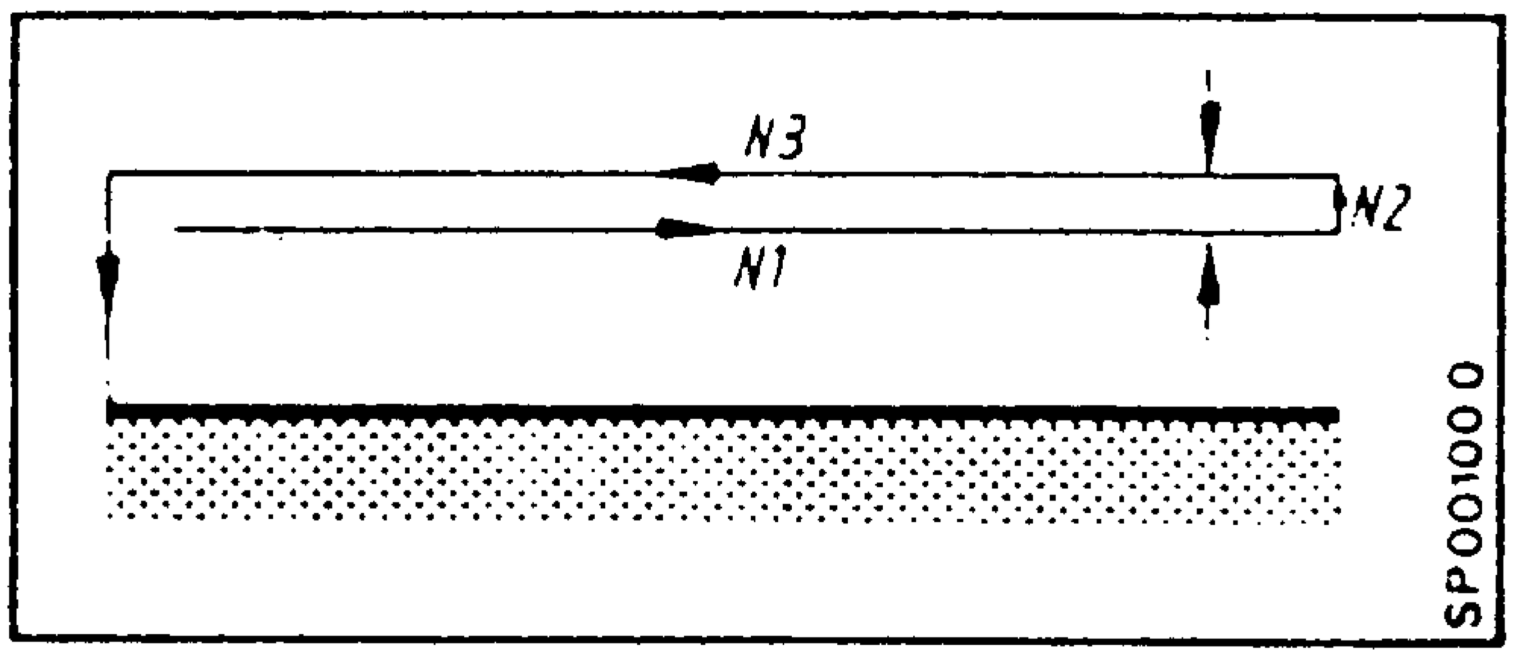
8.1.7 Special cases for CRC

Since the control always uses the information of the next block to calculate the intersectional path, a contour distortion will result under the following circumstances.



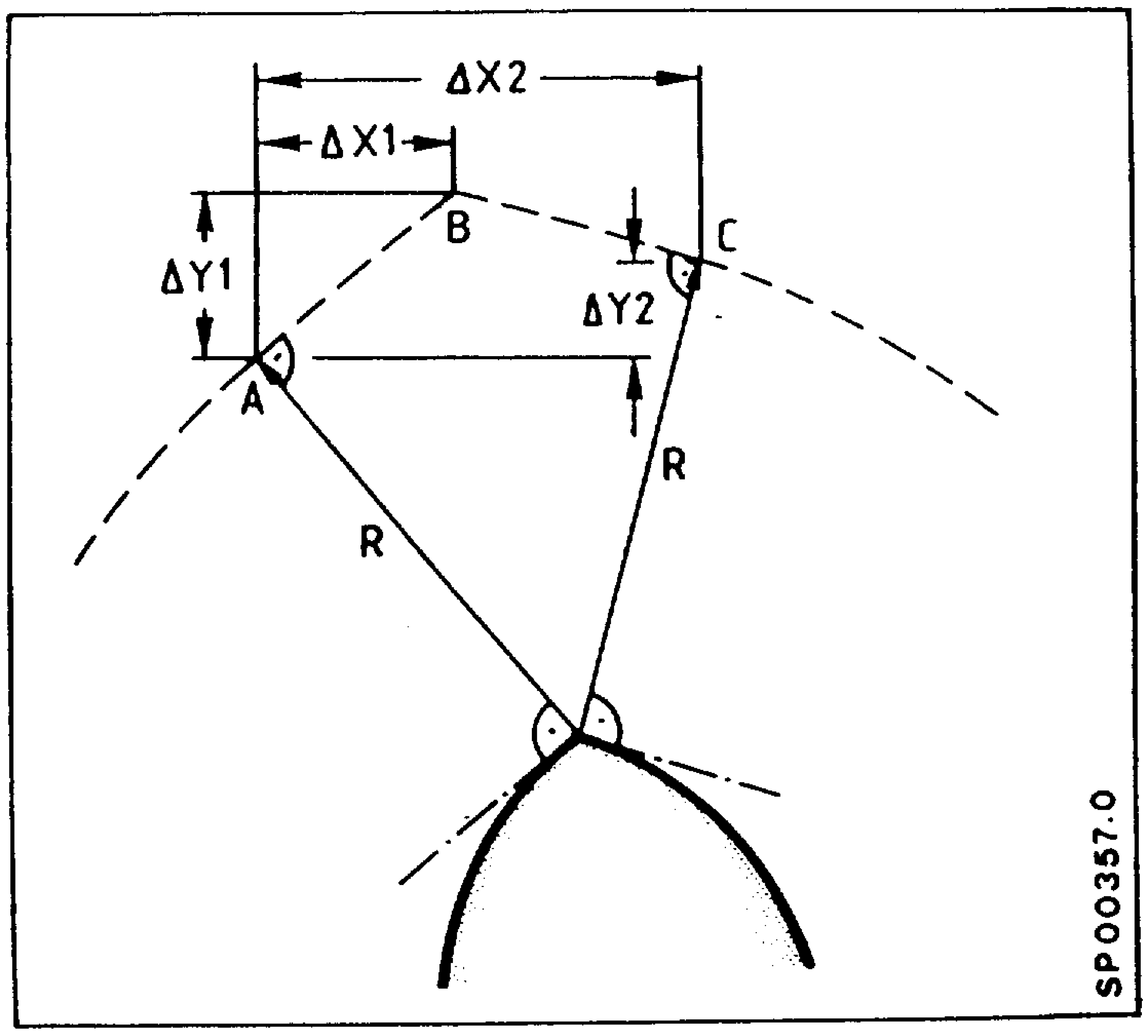
The tool offset dimension is larger than the distance between two paths.
Machining is not interrupted, however, an alarm is signalled and again cancelled at the end of the program.

CRC direction is not changed but the traverse direction of the cutter is reversed.

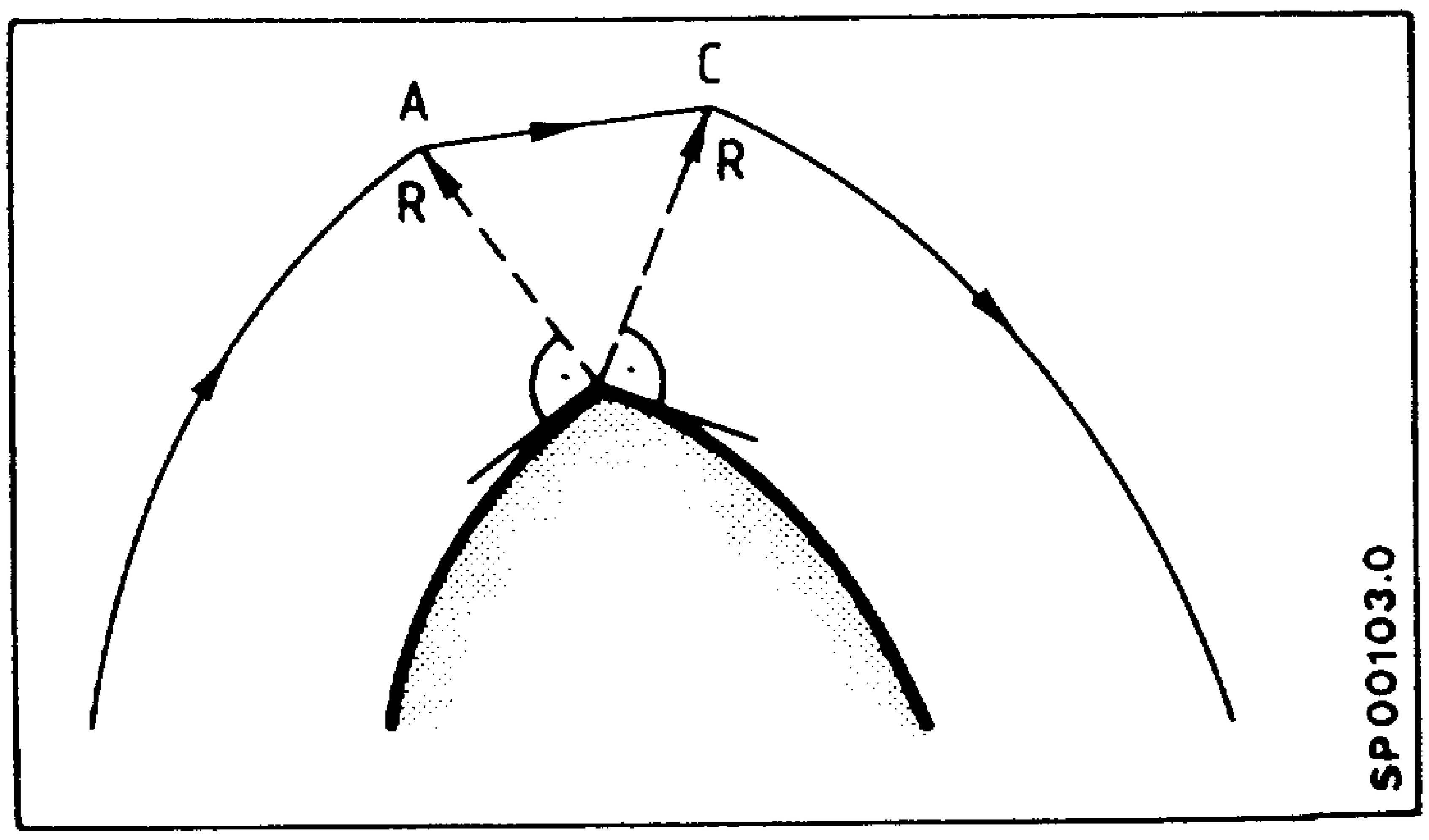


The retract dimension N2 must be programmed at least twice the cutter radius amount, since the cutter's position relative to the part has been incorrectly defined.
When the direction reversal takes place, the control will compensate by an amount twice the radius value in a direction towards the part.

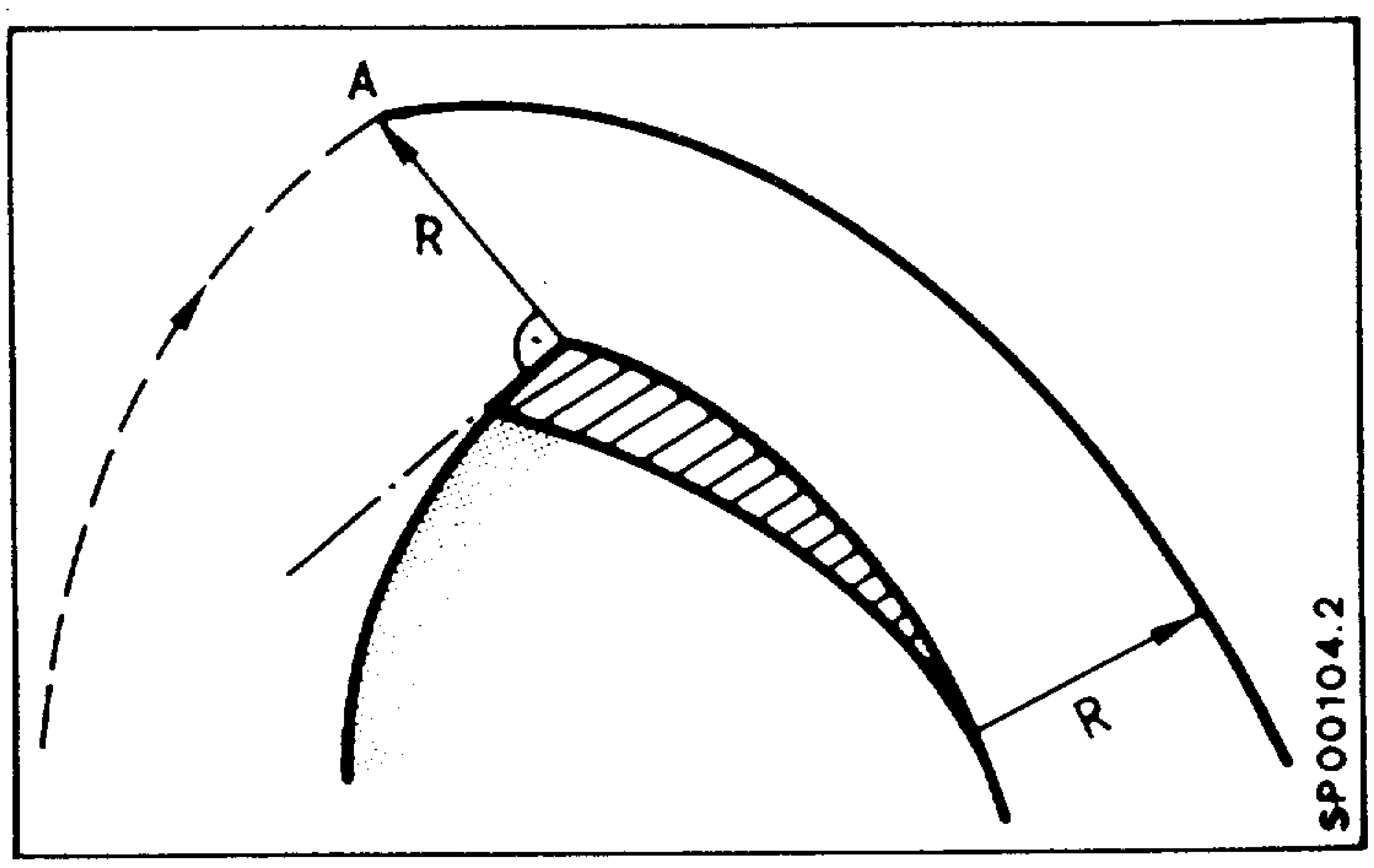
For an outside contour with a circle to circle transition at an obtuse angle, the following applies:



To avoid transition paths, control generated, that are of such short time duration that axis motion is momentarily halted, paths AB and BC may be omitted.
Dependent on a dimension tolerance defined during set up (max. 32000 μm) the resultant generated path is as shown.

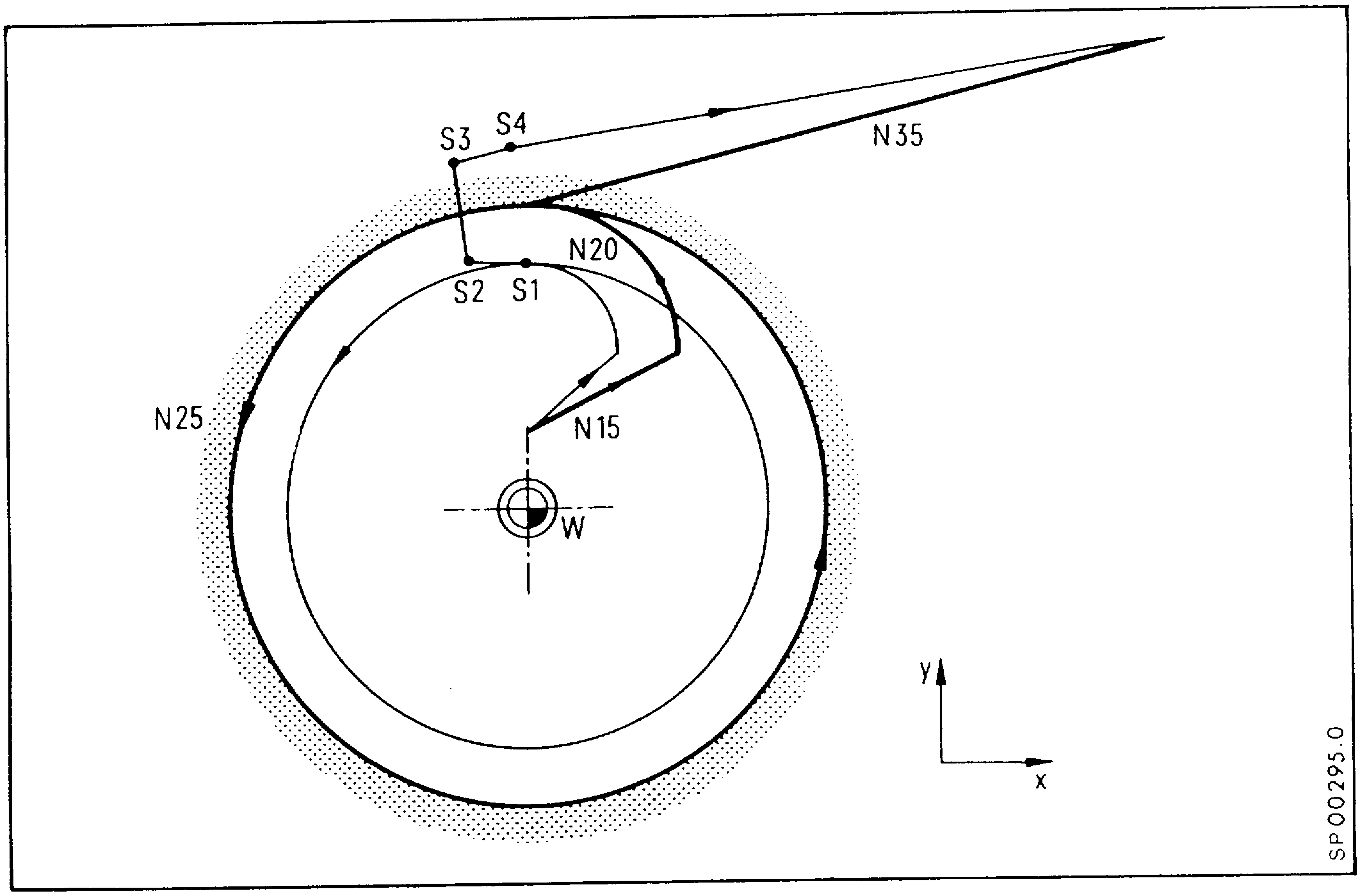


If ΔX1, and ΔY1 is less than d, the traversing is directly from A to C.



If ΔX1, ΔY1, ΔX2, and ΔY2 are less than d, no path is generated.
Traversing is continued from A using the new radius.

When machining inner contours with selected cutter radius compensation, an acute-angular contour transition (programmed) and an axis which moves in the meantime but does not lie in the cutter radius compensation plane, a virtual change of the block numbers results on the display.
With this method the workpiece will not be damaged.




```
N5   G00  Z100.
N10  X0.  Y10.
N15  G41  D01  X20.  Y20.
N20  G03  X0.  Y40.  I-20.  J0.
N25       X0.  Y40.  I0.    J-40.
N30  G01  Z0.
N35  G40  X80. Y60.
```

Logically, the points S1, S2, S3, S4 belong to block N25.
The sequence of machining (which is visible in the single block) is: ..., N20, N25 (S1), N30 (retracting tool from the workpiece), N25 (S2), N25 (S3), N25 (S4), N35... .
This method is also applied, if N25 is a linear block.

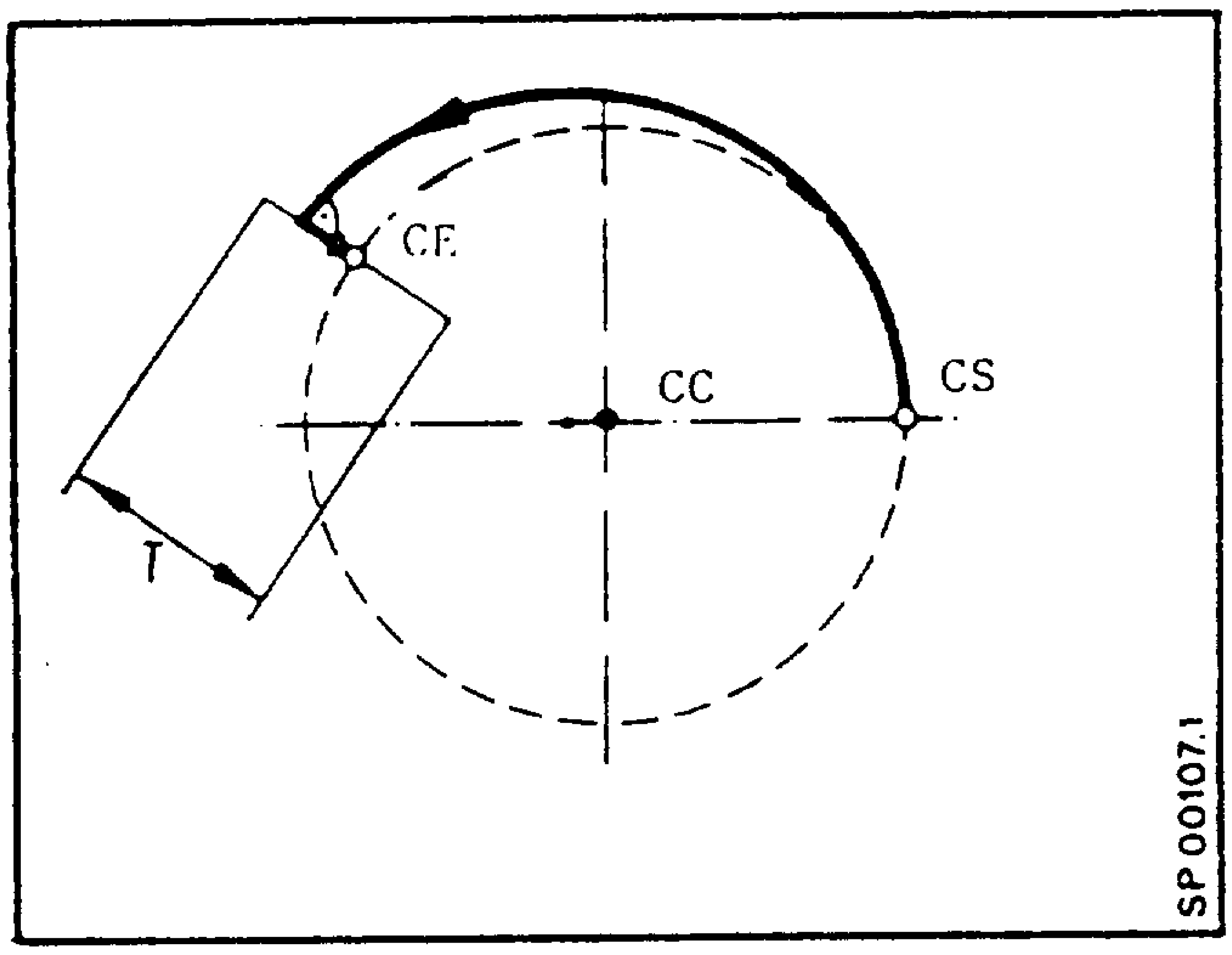
8.2 Input system, digrams, and tables

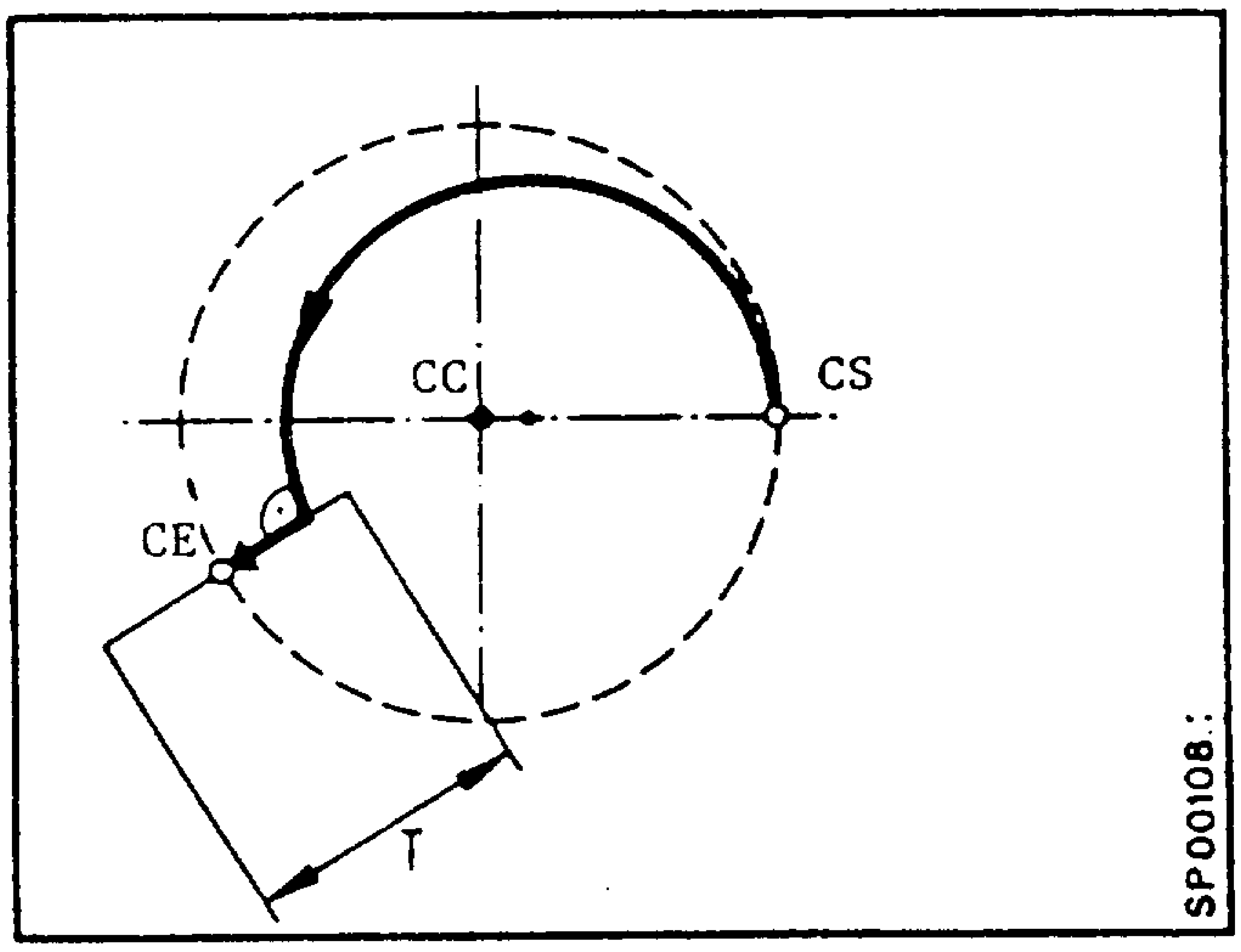
8.2.1 Incorrect input of the interpolation parameters I, J, K

An arc end point programming error is recognized by the control (assuming the tolerance window is exceeded). Circular interpolation will not begin, instead, an alarm is signalled.

If the programming error lies within the arc tolerance window, the control will position accurately to the end point, however, the path will deviate from the desired arc as shown.

Interpolation parameter

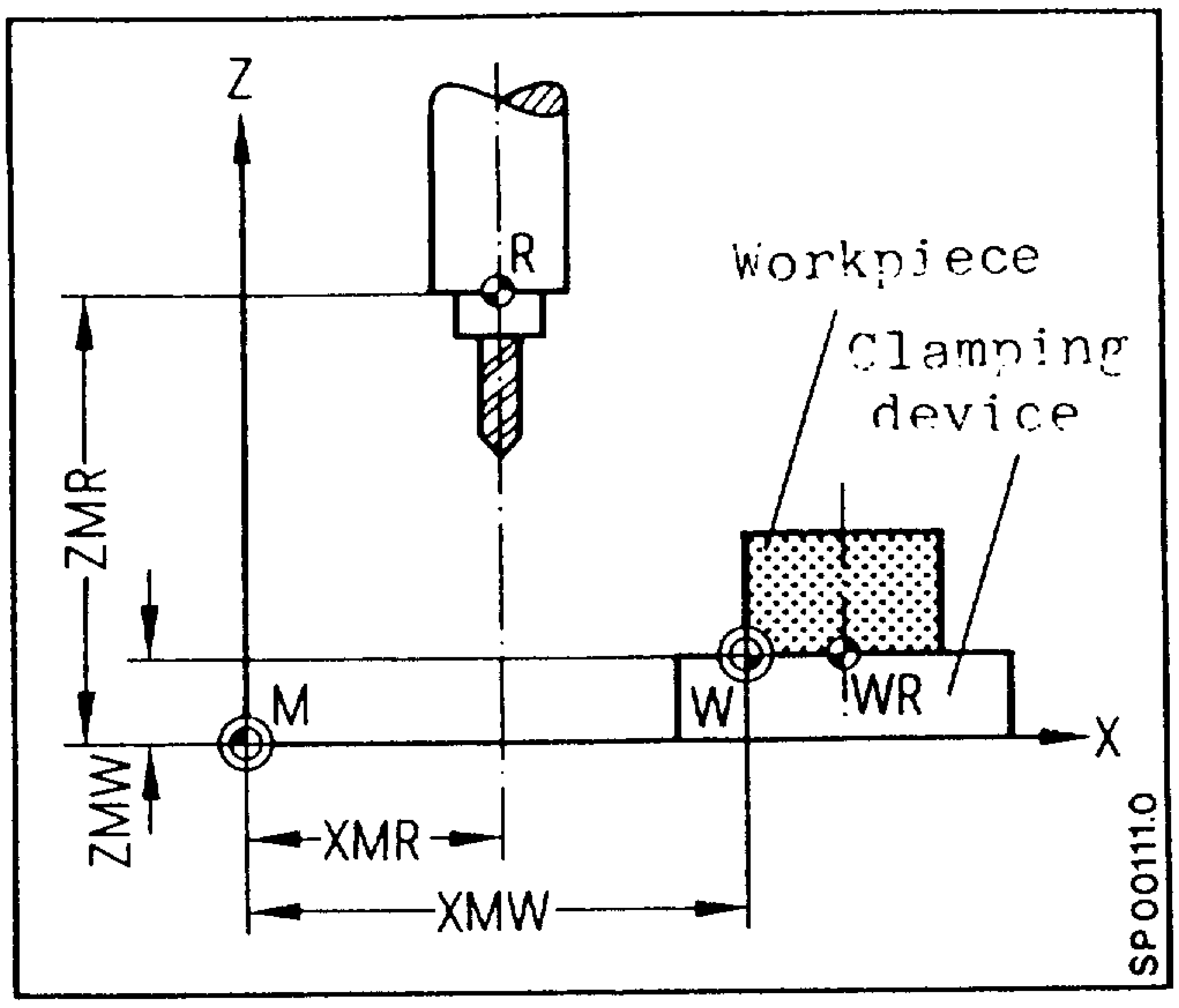




The tolerance window T about the arc end point CE is adjustable from ± 1 µm to ± 32000 µm.

The monitor can be suppressed by setting a large dimensional value. The tolerance window is input as an unsigned dimension stored under a machine parameter address.

8.2.2 Reference points



M = Machine zero

W = Workpiece zero

R = Machine reference point

WR = Workpiece reference point

XMR, ZMR etc. = Reference point coordinates for each axis

XMW, ZMW etc. = Sum of all zero offsets for each axis

Total zero offset = settable offset (G54..G57) + additive zero offset (G59) + external additive zero offset (ext. suppl. offset)

8.2.3 Path calculation

G91 in the first program block with a motion dimension

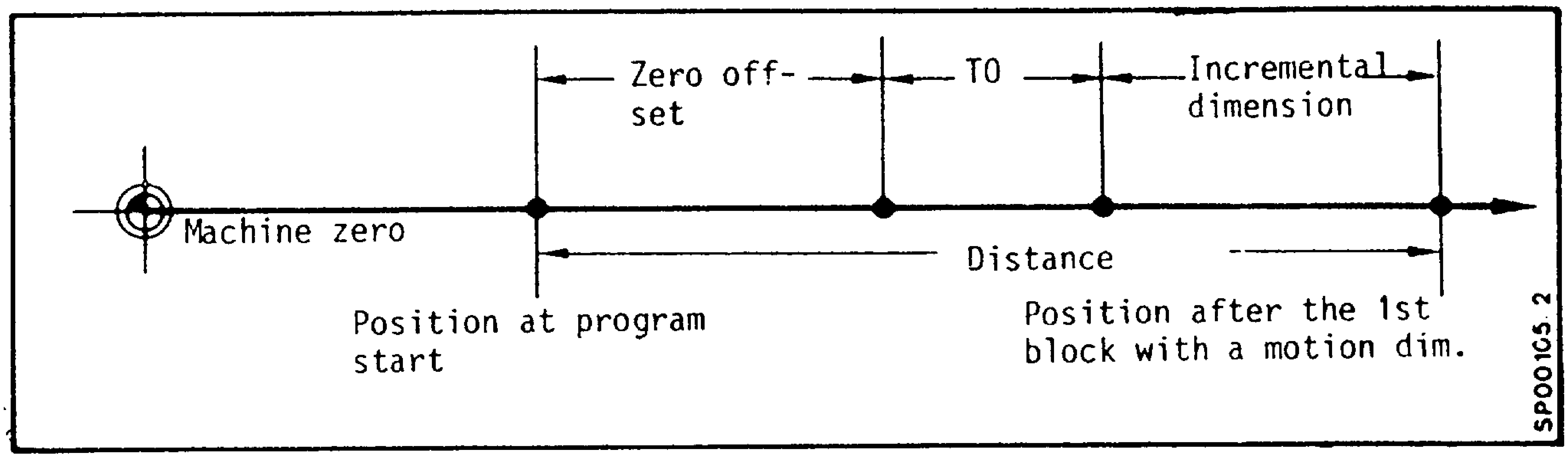
Distance = Incremental dimension + ZO + TO

G91 from the second program block with a motion dimension

Distance = Incremental dimension + ZO_{new} - ZO_{old} + TO_{new} - TO_{old}

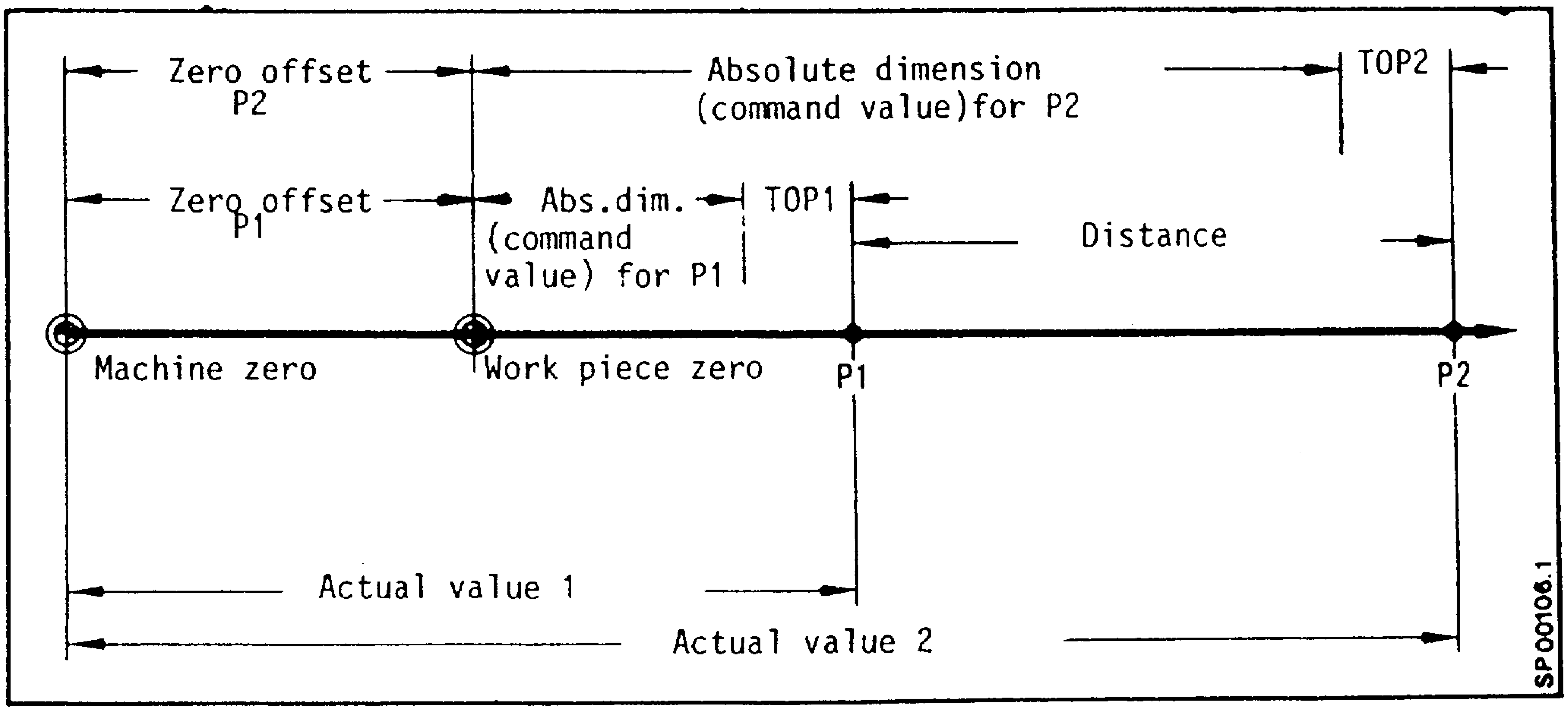
If the ZO and TO are not changed, the following applies:

Distance = Incremental dimension

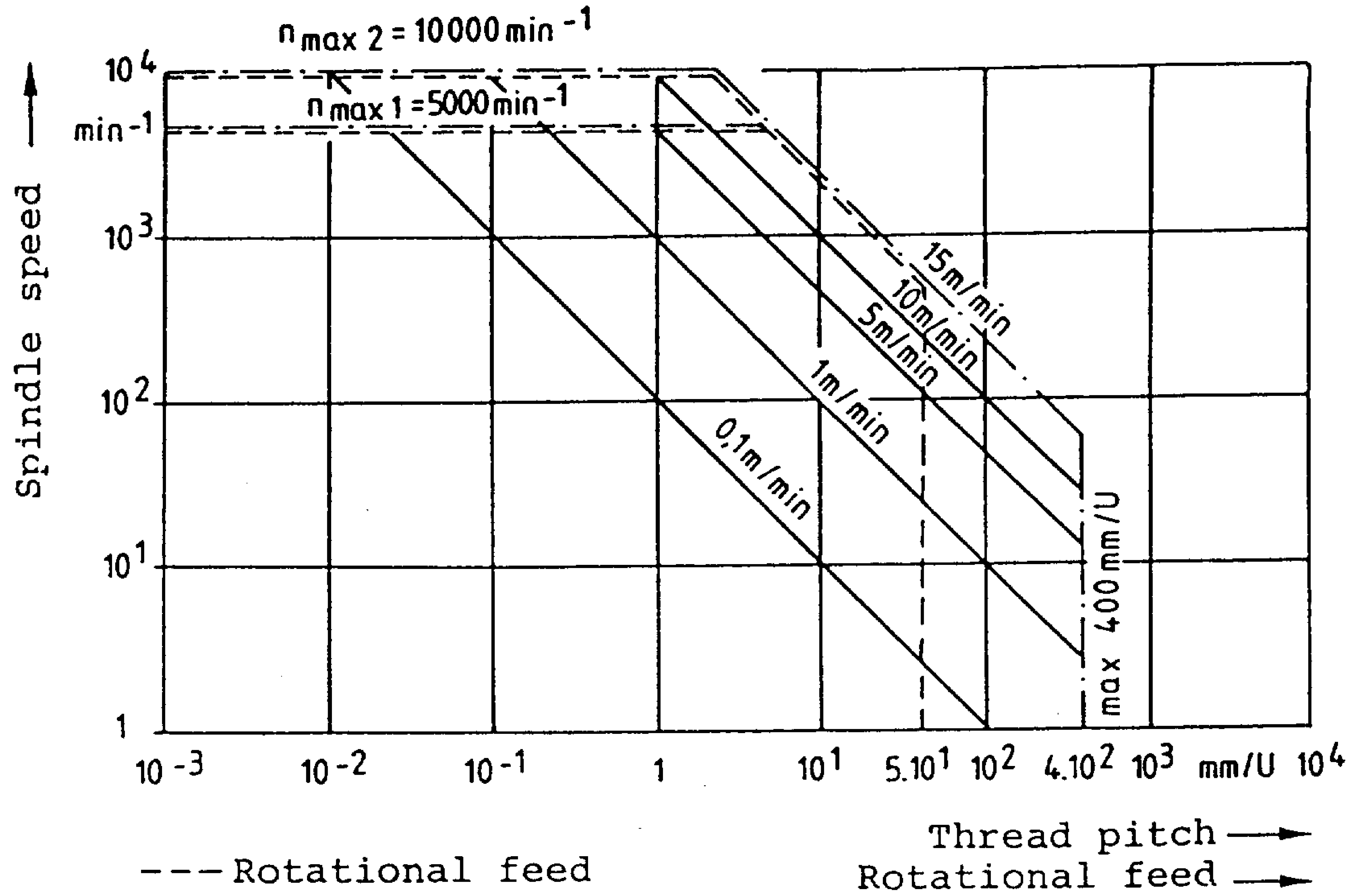


G90 in an arbitrary program block with a motion dimension

Distance = Absolute dimension$_{(new)}$ - absolute dimension$_{(old)}$ + $ZO_{(new)}$ - $ZO_{(old)}$ + $TO_{(new)}$ - $TO_{(old)}$



8.2.4 Rotational feedrate limits



--- Rotational feed

—·— Thread pitch

—— Feed rate in mm/min

Relationship between pitch and spindle speed (thread cutting G33):

Relationship between rotational feedrate and spindle speed

SP00197.3

$n_{max.\ 1}$ can be achieved with encoder connected 1:1

$n_{max.\ 2}$ can be achieved with encoder connected 1:2

8.2.5 Input formats

Address definition	Metric		Inch		Degrees	
	Decades	Smallest increment	Decades	Smallest increment	Decades	Smallest increment
Path data (linear axes) Interpolation parameter	± 5.3 1)		± 4.4 1)		-	
Path data (rotary axes)	-		-		± 5.3 1)	
Chamfer (P-); Radius (P)	3.3	10^{-3}mm	2.4	10^{-4}in	-	10^{-3} degrees
Zero offset	± 5.3		± 4.4 1)		± 5.3 1)	
Thread pitch	3.3		2.4		-	
Spindle speed S (Weighting factor set during commissioning)	4.0	1 or 0.1 min^{-1}	4.0	1 or 0.1 min^{-1}		
Linear feedrate (F)	5.0	mm/min	3.1	10^{-1}	5.0	degrees/min
Rotary feedrate (F)	2.3	10^{-3}mm/rev	1.4	10^{-4}in/rev		
Tool offset Length	±3.3	10^{-3}mm	± 2.4	10^{-4} in		
Tool offset Radius	±3.3		± 2.4			
Dwell X						
Dwell F	2.3	10^{-3}sec	2.3	10^{-3}sec		
Angle					3.5	10^{-5} degrees
Angle for oriented spindle stop 3)					3.1	10^{-1} degrees
R parameter	Dimension depends on associated address (internal floating point) all combinations (2 decades for call)					
G preparatory function	2		2			
M functions	2		2			
Block number	1 to 4		1 to 4			
H function	4		4			
Special functions @	2		2			
Factor P	3.5	10^{-5}	3.5	10^{-5}		

The parameters (R00-R99) 2) are always to be written with 2 decades. For all other functions (except address L) the leading zeros can be omitted.

1) For basic control 0, 1, 2: 1 decade less before the decimal point
2) For basic control 0, 1, 2: R00-R49
3) Not with basic controls 0, 1, 2

8.2.6 Code tables

	ISO DIN 66024									Code Table	
	P	7	6	5	4		3	2	1	Bit-No. (P testbit)	
	6	7	6	5	4	T	3	2	1	Track-No. (T sprocket)	
Character	Hole combination									Allowed in Program	Allowed in Leader
NUL						•				x	x
BS	●				●	•				–	x
HT					●	•			●	x	x
LF					●	•		●		x	-
CR	●				●	•	●		●	x	x
SP	●		●			•				x	x
(			●		●	•				x	x
)	●		●		●	•			●	x	x
%	●		●			•	●		●	x	–
.			●	●	●	•		●		–	x
/	●		●		●	•	●	●	●	x	x
+			●		●	•		●	●	x	x
–			●		●	•	●		●	x	x
.			●		●	•	●	●		x	x
0			●	●		•				x	x
1	●		●	●		•			●	x	x
2	●		●	●		•		●		x	x
3			●	●		•		●	●	x	x
4	●		●	●		•	●			x	x
5			●	●		•	●		●	x	x
6			●	●		•	●	●		x	x
7	●		●	●		•	●	●	●	x	x
8	●		●	●	●	•				x	x
9			●	●	●	•			●	x	x
@	●	●				•				x	x
A		●				•			●	x	x
B		●				•		●		x	x
C	●	●				•		●	●	x	x
D		●				•	●			x	x
E	●	●				•	●		●	–	x
F	●	●				•	●	●		x	x
G		●				•	●	●	●	x	x
H		●			●	•					x
I	●	●			●	•			●	x	x
J	●	●			●	•		●		x	x
K		●			●	•		●	●	x	x
L	●	●			●	•	●			x	x
M		●			●	•	●		●	x	x
N		●			●	•	●	●		x	x
O	●	●			●	•	●	●	●	–	x
P		●		●		•				x	x
Q	●	●		●		•			●	—	x
R	●	●		●		•		●		x	x
S		●		●		•		●	●	x	x
T	●	●		●		•	●			x	x
U		●		●		•	●		●	x	x
V		●		●		•	●	●		x	x
W	●	●		●		•	●	●	●	x	x
X	●	●		●	●	•				x	x
Y		●		●	●	•			●	x	x
Z		●		●	●	•		●		x	x
DEL	●	●	●	●	●	•	●	●	●	x	x

	EIA 244B										
	8	7	6	P	4		3	2	1		
	8	7	6	5	4	T	3	2	1		
Character	Hole combination									Allowed in Program	Allowed in Leader
No hole						•				-	x
RT			●		●	•		●		-	x
TAB			●	●	●	•	●	●		x	x
EOB	●					•				x	-
LC)		●	●	●	●	•		●		-	x
ZWR				●		•				x	x
1) (				●	●	•		●		x	x
1))		●			●	•		●		x	x
EOR					●	•		●	●	x	–
		●	●		●	•		●	●	x	x
/			●	●		•			●	x	x
+		●	●	●		•				x	x
–		●				•				x	x
0			●			•				x	x
1						•			●	x	x
2						•		●		x	x
3				●		•		●	●	x	x
4						•	●			x	x
5				●		•	●		●	x	x
6				●		•	●	●		x	x
7						•	●	●	●	x	x
8					●	•				x	x
9				●	●	•			●	x	x
2) @		●	●		●	•	●		●	x	x
a		●	●			•			●	x	x
b		●	●			•		●		x	x
c		●	●	●		•		●	●	x	x
d		●	●			•	●			x	x
e		●	●	●		•	●		●	—	x
f		●	●	●		•	●	●		x	x
g		●	●			•	●	●	●	x	x
h		●	●		●	•				x	x
i		●	●	●	●	•			●	x	x
j		●		●		•			●	x	x
k		●		●		•		●		x	x
l		●				•		●	●	x	x
m		●		●		•	●			x	x
n		●				•	●		●	x	x
o		●				•	●	●		—	x
p		●		●		•	●	●	●	x	x
q		●		●	●	•				—	x
r		●			●	•			●	x	x
s			●	●		•		●		x	x
t			●			•		●	●	x	x
u			●	●		•	●			x	x
v			●			•	●		●	x	x
w			●			•	●	●		x	x
x			●	●		•	●	●	●	x	x
y			●	●	●	•				x	x
z			●		●	•			●	x	x
IRR		●	●	●	●	•	●	●	●	x	x
UC		●	●	●	●		●			-	x

1) Punched track for beginning and end of a comment

2) Code definable via machine data

8.3 Program key

Group	EIA	ISO	Code for basic control 0,1,2	3	4 / 4B,4C	Section	Function and meaning
	EOR	o/o				1.	Program start
	EOR ... EOB	o/o ... LF	1 ... 9999	1 ... 9999	1 ... 9999	1.7	Program number
	n /n	N /N	1 .. 9999	1 ... 9999	1 ... 9999	1.4 1.4	Block number Skippable block
	. + -	. + -				1.3	Decimal point Positive sign (can be omitted) Negative sign
G1	g	G	00 01• - - 02 03 33	00 01• - - 02 03 33	00 01• 10 11 02 03 33	3.2 3.3 3.18 3.18 3.4 3.4 3.5	Rapid traverse Linear interpolation Polar coordinate programming rapid traverse Polar coordinate programming linear interpolation Circular interpolation clockwise Circular interpolation counter- clockwise Thread cutting
					– 34 – 35	3.5.3.2 3.5.3.3	Thread cutting, pitch increasing linearly Thread cutting, pitch decreasing linearly
G2	g	G	■04*	■04*	■04*	3.9	Dwell time, time under F address
G4	g	G	17• 18 19	17• 18 19	17• 18 19	3.13 3.13 3.13	Plane selection XY-plane XZ-plane YZ-plane
G5	g	G	40• 41 42	40• 41 42	40• 41 42	3.15 3.15 3.15	Cancel cutter radius compensation Cutter radius compensation, left Cutter radius compensation, right
G6	g	G	- -	- -	43• 44	3.16 3.16	Tool length offset positive Tool length offset negative
G7	g	G	■53	■53	■53	3.11.3	Suppress the zero offsets
G8	g	G	54• 55 56 57	54• 55 56 57	54• 55 56 57	3.11.1 3.11.1 3.11.1 3.11.1	Select zero offset 1 Select zero offset 2 Select zero offset 3 Select zero offset 4
G9	g	G	■59*▲	■59*▲	■59*▲	3.11.2	Programmable additive zero offset
G10	g	G	60• - 63 64	60• - 63 64	60• 62 63 64	3.6 3.8.1 3.7 3.8	Exact stop Continuous path mode via reduced speed Tapping with a compensation chuck Continuous path mode
G11	g	G	70 71	70 71	70 71	3.10 3.10	Input system in inch Reset state via Input system in metric machine data
G12	g	G	90• 91	90• 91	90• 91	3.1 3.1	Absolute dimension programming Incremental dimension programming
G13	g	G	-	-	92	3.17	Factor for the unit circle with P
			-	-	– ▲92* – ▲92*	3.5.4.2 3.21	Spindle-related starting angle offset under A Smothing exponent for thread under T
G14	g	G	94• 95	94• 95	94• 95	3.12 3.12	Feedrate under address F in .../min Feedrate under address F in .../rev.
G15	g	G	80• 81 ... 89	80• 81 ... 89	80• 81 ... 89	7.	No boring cycle Boring cycle
G16	g	G	- -	- -	– 36• – 37 – 38	3.19 3.19 3.19	Coordinate transformation OFF Coordinate transformation TRANSMIT "ON" Coordinate transformation "DOUBLE- TRANSMIT" ON

Program key (continued)

Group	EIA	ISO	Code for basic control 0,1,2	 3	 4	Section	Function and meaning
	s	S	1...99 1... 9999	1...99 1... 9999	1...99 1... 9999	4.1	Spindle speed (geometrical series) Spindle speed in rev/min or 0.1 rev/min
	t	T	1...99 1... 9999	1.... ...9999	1.... 9999	4.2 4.2	Tool-No. Tool-No.
	d	D	01..32 00	01...32 00	01...32 01...64 1) 00	3.14 3.14	Tool offset number Tool length 999.999 mm 99.9999 inch Cutter radius 999.999 mm 99.9999 inch Suppress tool offset
	h	H	-	-	1...9999	4.4	Auxiliary functions
M1	m m	M M	00 02 17 30	00 02 17 30	00 02 17 30	4.3 4.3 4.3 4.3	Programmed stop, unconditional End of program End of subroutine, written in the last block of the subroutine End of program (see M02)
M2	m	M	03 04 05•	03 04 05• 19■	03 04 05• 19■	4.3 4.3 4.3 4.3	Spindle start, clockwise Spindle start, counter clockwise Spindle stop Oriented spindle stop, angle under S in degrees
M3	m	M	00.. 99	00... 99	00... 99	4.3	Miscellaneous functions, unassigned except groups M1, M2
	l	L	01.. .. 99.. - - ..01.. ..99 ..01... ..99	01.. 99.. - - ..01... ..99 ..01... ..99	01.. ... 99.. 001.. 2) 999.. ...010... ...99 ...01... ...99	1.8 1.8	Subroutine numbers (L80-99, L900-L999) for Cycles Number of repetitions
	@	@	00 01 02 03 31 - - - - - -	00 01 02 03 31 10 15 18 20 90-93 29	00 01 02 03 31 10 15 18 20 90-93 29	5.7 5.6 5.8 5.9 5.10 5.11 5.12 5.13	Unconditional jump Conditional jump, equal Conditional jump, greater Conditional jump, greater or equal Empty buffer store Square root Sine Arctan Load address parameters Address parameters Load/read from system memory
	EOB	LF				1.4	End of block

1) With basic control 4 32 additional compensations as option

2) Only with basic control 4

8.4 Special cases

8.4.1 Special case "Delete distance to go"

The remaining setpoint-actual value difference (distance to go) of a programmed axis motion can be reset (erased) via an interface. The machine manufacturer determines which M function is used for activating and disabling the "delete distance to go" signal.

The axis, whose traversing motion was interrupted by the "delete distance to go" function, must have a G90 programmed in the subsequent block (absolute dimension programming). The programming procedure given below must be followed:

- empty buffer with @ 31
- load actual value with @ 29
- travel to actual position with G90

Programming example

```
:
M..                          - activate the "delete distance to go"
                               function
G90  G01  X100               - Delete distance to go  for X60
M..                          - Deselect "delete distance to go"
                               function
@31                          - empty buffer
R00 1001 @29 19312 R00       - load actual value
G90 XR93                     - travel to actual position
G00 G91 X 60                 - travel to new position e.g. X 120
:
```

To
Siemens AG

E 885
Postfach 48 48
D-8500 Nürnberg 1

Suggestions

Corrections

For Publication/Manual:

SINUMERIK 3M
Basic Version 4C

Programming Guide

Order No.: 6ZB5 410-0AM02-0BA0
Edition: 08.88

Sender:

Name

Company/Dept.

Address

Telephone /

Should you come across any printing errors when reading this publication, we would ask you to inform us accordingly, using this form. We would also welcome any suggestions you may have in the way of improvement.

Suggestions and/or corrections