SINUMERIK 3T/3TT
Basic Version 4C

Programming Guide

SINUMERIK

Edition 08.88

SINUMERIK 3T/3TT
Basic Version 4C

Programming Guide

Edition 08.88

Subject to change without prior notice

Functions extending beyond the scope of this Description may be capable of operating on the controller. However, we accept no responsibility for such functions for new equipment or equipment which has been serviced.

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

© Siemens AG 1989

General 0

Program format 1

Path information 2

Preparatory functions 3

Switching and miscellaneous functions M, S, T, H 4

Parameters 5

Automatic insertion of chamfers and radii 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 410-0AC02-0BA0	New Edition

Contents

		Page
0	**General**	0-1
0.1	General Introduction to Programming Instructions	0-1
1.0	**Program Format**	1-1
1.1	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	Remark	1-7
1.7	Part programs	1-8
1.8	Subroutines	1-9
1.9	Subroutine call, Subroutine nesting	1-10
1.10	Tape format	1-12
1.11	Paper tape format for erasing programs	1-14
2.0	**Path Information**	2-1
2.1	X, Z axis commands	2-1
2.2	Mirror image	2-3
3.0	**Preparatory Functions**	3-1
3.1	G90/G91 Absolute and incremental data programming	3-2
3.2	G00 Rapid traverse	3-4
3.3	G01 Linear interpolation	3-5
3.4	G02/G03 Circular interpolation	3-6

3.5	**G33 Thread cutting with constant pitch**	3-8
3.5.1	Thread cutting with constant pitch on variable angle tapers	3-10
3.5.2	Multi-start threads	3-10
3.5.2.1	Multi-start threads with axial starting point offset	3-10
3.5.2.2	Multi-start threads using spindle-related starting angle offset	3-10
3.5.3	Infeed directions	3-11
3.5.4	Single start facing thread with constant pitch	3-12
3.5.5	Single start logitudinal thread with constant pitch	3-13
3.5.6	Single start facing thread with constant pitch	3-14
3.5.7	Multi-start thread with constant pitch	3-15
3.5.8	Single start taper thread with constant pitch	3-16
3.6	**Variable pitch threads**	3-17
3.6.1	Variable pitch thread without G34, G35	3-17
3.6.2	G34 Thread cutting (increasing pitch)	3-18
3.6.3	G35 Thread cutting (decreasing pitch)	3-18
3.7	**G09 Speed decrease**	3-19
3.8	**G04 Dwell time**	3-20
3.9	**G70/71 Input systems**	3-21
3.10	**Zero offsets (ZO)**	3-23
3.10.1	G54 to G57 Adjustable zero offset	3-24
3.10.2	G59 Programmable additive zero offset	3-26
3.10.3	G53 Cancelling the zero offset	3-28
3.11	**G94/G95/G96 Feedrate F**	3-29
3.12	**G96 S.. Constant cutting speed**	3-30
	G97 Constant speed	3-31
3.13	**G92 S.. Limitiation of spindle speed when using G96**	3-32
3.14	**G40/G41/G42 Tool nose radius compensation (CRC)**	3-33
3.15	**G10/G11 Programming in polar coordinates**	3-35
3.16	**Dynamic smoothing exponent for threads**	3-37
3.17	**Turning/milling operation**	3-38
3.17.1	The turning machine in turning/milling operation	3-39
3.17.2	Arrangement and structure	3-40
3.17.3	G36, G37 TRANSMIT coordinate transformation (Option B65)	3-42
3.17.4	Display	3-52
3.18	**Programmable coordinate rotation**	3-53
3.19	**Coordinate rotation for inclined workpieces and tools (G39)**	3-55
3.19.1	Machine data	3-56
3.19.2	Actual values, reference points	3-57
3.19.3	Selection and deselection of coordinate rotation (G39/G36)	3-57
3.19.4	Switching on the control, reference point approach	3-58
3.19.5	General information on coordinate rotation X/Z	3-58

4.0	**Switching and Miscellaneous functions M,S,T,H**	4-1
4.1	**S Word**	4-1
4.2	**T Word tool command**	4-2
4.2.1	Tool offset without using CRC	4-3
4.2.2	Tool offset using tool nose radius compensation	4-5
4.2.3	Tape format for input of tool track coding	4-7
4.2.4	Tool offset with tool track coding	4-8
4.3	**Miscellaneous functions M**	4-9
4.4	**Auxiliary function H**	4-11
4.5	**Rapid switching and auxiliary functions**	4-11
5.0	**R-Parameters**	5-1
5.1	**Parameter definition**	5-3
5.2	**Assignment of the parameters in the program**	5-3
5.3	**Parameter linking**	5-4
5.4	**Parameter chaining**	5-5
5.5	**Branch conditions**	5-7
5.5.1	@ 00 Unconditional jump	5-7
5.5.2	@ 01, @ 02, and @ 03 Conditional jump	5-9
5.5.3	Conclusive example for unconditional and conditional jumps	5-11
5.6	**@ 31 Clearing the buffer**	5-12
5.7	**@ 10 "Square root"**	5-13
5.8	**@ 15 "Sine"**	5-14
5.9	**@ 18 "Arctan"**	5-15
5.10	**@ 20 "Load address parameter"**	5-16
5.11	**@ 90 to @ 93 "Address parameter"**	5-18
5.12	**@ 22 "Intersection calculation"**	5-20
5.13	**@ 29 "Load/read system memory"**	5-23
5.14	**@ 28 Selection of the R parameter group (only Basic Version 4C)**	5-27
5.15	**@ 21 Reference Editing**	5-28

6.0	**Automatic insertion of chamfers and radii**	6-1
6.1	**Linking of blocks**	6-2
6.2	**Example**	6-3
6.3	**Switching and auxiliary functions in linked blocks**	6-4
6.4	**Blueprint programming**	6-5
6.5	**Operation of the function G09. F, S, T, H, M in the contour**	6-10
6.6	**Linking of blocks**	6-11
6.7	**Examples**	6-12
6.8	**Switching and auxiliary functions in linked blocks**	6-15
7.0	**Cycles**	7-1
7.1	**Machining cycles**	7-1
7.1.1	L94 Stock removal cycle (paraxial roughing)	7-1
7.1.2	L96 Thread cutting cycle	7-10
7.1.3	L95 Stock removal cycle (Basic Control 4B/4C)	7-19
7.1.4	L97 Thread cutting cycle (Basic Control 4B)	7-28
7.1.5	L902 Four-point thread cutting cycle	7-35
7.1.6	L903 Plunge-cutting cycle	7-41
7.1.7	L98 Deep-boring cycle	7-46
7.2	**L99 Clearing the buffer**	7-68
8.0	**Appendix**	8-1
8.1	**Tool nose compensation (CRC)**	8-2
8.1.1.	Activation of CRC	8-2
8.1.2	CRC in the program	8-3
8.1.3	Cancellation of CRC	8-6
8.1.4	M00, M02 and M03 with CRC selected	8-8
8.1.5	Special cases with CRC	8-9
8.1.6	CRC in combination with various types of block	8-11
8.1.7	Repetition of previously activated G functions (G41, G42) with the same offset number	8-14
8.2	**Input systems, diagrams and tables**	8-15
8.2.1	Inexact input of interpolation parameters I and K	8-15
8.2.2	Reference points	8-16
8.2.3	Calculation of block increment	8-17
8.2.4	Limit data for rotational feedrate	8-18
8.2.5	Spindle speed as a function of the turning radius with v = constant	8-19
8.2.6	Input formats	8-20
8.2.7	Code Table	8-21
8.3	**Program key**	8-22
8.4	**Special cases**	8-25
8.4.1	Special case "Delete distance to go"	8-25

0.1 General introduction to Programming Instructions

This programming guide applies to SINUMERIK 3T in basic controls 0, 1, 2, 3, 4, 4B and 4C for turning machines with one slide and for SINUMERIK 3TT for double-slide or double-spindle turning machines.

The appropriate sections of the 3M Programming Instructions apply to controller types 3T/3TT with the spindle as circular axis (C axis).

With the SINUMERIK 3TT, each carriage is assigned to its own control system. The machining programs, including all switching and auxiliary functions are programmed separately for each control system (carriage) but with the same addresses. The machining program assigned to the carriage is stored in the program memory of the corresponding control system.

The functions described in these Programming Instructions are fully available to each control system of the SINUMERIK 3TT.

The programs in these Programming Instructions are based on the following preconditions:

1. Machining with only one carriage is described (SINUMERIK 3TT).

2. Machining always takes place behing the turning centre.

3. The decimal point is written even if it is automatically generated by the controller.

4. Block structure is in accordance with DIN 66024, DIN 66025, DIN 66217, ISO R 1056, ISO R 1057 and ISO R 1058.

5. The programming examples are written in ISO code.

6. For absolute dimensions, X values are entered in diameter (machine data set).

7. All geometric values are metric.
For conversion to inches, see Sections 8.2.6 and 3.9.

8. For the sake of clarity, preparatory functions are programmed even if they are commands with reset positions.

9. The specified maximum values are controller limits. They may be restricted during operation by the machine, interface and input/output devices.

10. These Programming Instructions are intended for the entire range of functions of the controller in basic version 4 and of the SINUMERIK 3TT. Functional limitations affecting basic versions 0, 1, 2 and 3 are indicated by a footnote.

11. The Table of Contents of these Programming Instructions can also be found in the fold-out program key.

Functions beyond the scope of this manual may be executable in the controller. However, no guarantee is made in regard to these functions for new deliveries or maintenace.

We reserve the right to make changes to this manual as a result of technical modifications.

1.0 Program format

1.1 Tape code

The data on the tape is coded in accordance with defined standards, i.e., a given combination of holes always represents the same character. Two code structures are used.

DIN 66025 (ISO)
EIA-RS 244-B

The controller automatically recognises the correct code. Code recognition results from the % character read or EOR. Each tape must be written in one of the permitted codes. A change of codes on a single tape or joining together of tapes with different codes is not permitted and results in activation of the character parity check.

The characters of a code have a common feature:

ISO The number of holes is always even.
EIA The number of holes is always odd.

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 to 100 %.

The block parity check monitors to ensure that the number of characters in a block is even. Uneven character counts are completed with "HT" or "SP". This test can be de-activated.

As an additional test program is read into the internal program memory twice and a complete program comparison made.

When the error is detected, the read operation is stopped and the error is indicated on the operator panel.

Assignment of the word addresses is in accordance with DIN 66025 (ISO).

1.2 Character set

The controller reads every character contained in the tape code. However, only certain characters may be used for defining the programm - related, geometric and technological statements in the parts program.

ISO - Code

Address letters	A,B,F,G,H,I,K,L,M,N,R,S,T,X,Z	
Digits	0,1,2,3,4,5,6,7,8,9	
Letters	O (Tool offset input TO = TOOL OFFSET)	
Printable special characters	%, (,), +, -, /, ., @	
Non-printable special characters	HT	Tabulator
	SP	Space
	DEL	Delete
	CR	Carriage return
	LF	Line feed

READ IN the following characters are not stored and not executed	PUNCHING, PRINTING The following characters are generated
HT	
SP	SP (after each word)
DEL	
CR (sequence CR LF optional	CR generated twice following LF
LF is indicated in the display as *.	

1.3 Word construction

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 defined by DIN 66025, Page 4.

The metric values are as follows:

Basic version 0,1,2:	%04 N04 G02 XL+43 ZL+43 ID043 KD043 F05 LF S04 T04 R2 RL+043 BD033 M02
Basic version 3:	%04 N04 G02 XL+053 Z1+053 ID053 KD053 F05 L4 S04 T04 R2 RL+053 BD033 M02
Basic Version 4, 4B, 3TT:	%04 N04 G02 XL+053 ZL+053 AL035 ID053 KD053 F05 L5 S04 T04 R2 RL+053 BD033 M02 H04
Basic Version 4C, 3TT:	%04 N04 G02 XL+053 ZL+053 AL035 ID053 KD053 F05 L5 S04 T04 H+04 R2 RL+053 BD033 M+02

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	positions in
Second and third	decade	series of digits (Coordinate values X,Z,I,K in mm)
Symbol	*	Block end.

Word examples:

X ± 2345.531		G09	
X	Address	G	Address
±	Sign		
2	Digits	09	Digits
5.5	Decimal point		
1	Digits		

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

Decimal point entry is possible using the following addresses:

R, X, Z, I, K, R, B, F (mm/rev) see 8.2.6

1.4 Block structure

A block consists of two or more words and the "End of Block" character.

The block has a maximum possible length of 120 characters, but should be limited to 80 because this is the maximum number which can be displayed.

Example of a block:
N9234 G.. X.. Z.. F.. S.. T.. M.. LF

N	Block number address
9234	Block number
G	Preparatory function
X.. Z..	Path data
F..	Feedrate
S..	Speed
T..	Tool number (offset number
M..	Miscellaneous functions
LF	End of block

Jumps can occur in the sequence of block numbers, e.g. modified or newly introduced blocks can be designated by introducing a digit into the thousands position.

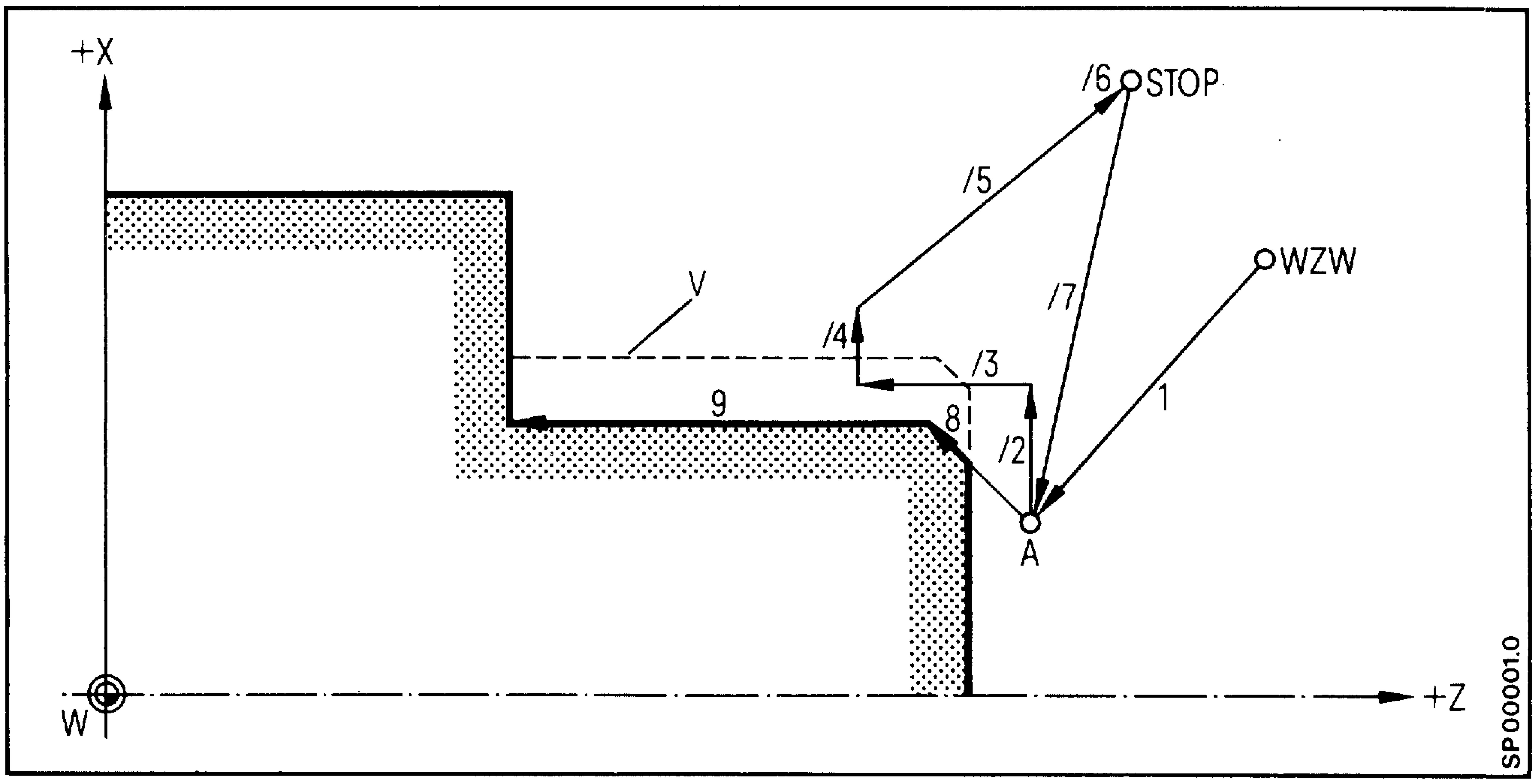
By placing the "/" character in front of the block number the block so designated may be skipped by the controller. This function is dependent on the "Skip" - delete block - switch.

/N... skippable block

Thus it is possible to skip certain machining sections such as test cuts, dry runs for threadcutting or part cuts necessitated by inaccurately defined offsets, i.e. their execution is not required for ech workpiece of the series.

It is important to ensure that deletable blocks form a closed loop (i.e. with the same starting and finishing point).

A = Starting Point; V = Pre-machined contour; WZW = Tool change point



Blocks 2 - 7 can be deleted if necessary.

1.5 Leader

The leader is used to differentiate between different tapes.

All characters

- except %, because the automatic code recognition is initiated by % (ISO code)
- except EOR, because the automatic code recognition is initiated by EOR (EIA code)

are permitted in the tape reader. During execution of the program the leader is skipped by the controller. The leader is not stored.

1.6 Remark

Program blocks can be more clearly defined using remarks. In this way it is also possible to display operator instructions on the CRT. A remark must not contain the character % or LF or a block number used in the program as the text in brackets is read in branching functions and block search.

The maximum length for a remark is 20 characters. If more text is required, several consecutive remarks can be programmed.

A comment can be up to 29 characters long. If more text is required, several consecutive comments can be programmed.

Example:

N 25 G70 (all subsequent)
(geometric values in inches)

Correct			Incorrect	
X100.		(remark) Z200.	Z (remark)	100.
X100.	R01	(remark) Z200.	Z 100 (remark)	R01

X	Address
100.	Digits
R01	R-parameter
(	Start of remark
Remark	remark
)	End of remark

No remark may be inserted between the address and digits or between a word and its associated parameters.

1.7 Part programs

A part program describes the sequence of a machining process and consists of the part program itself and any subroutine and/or cycles which may be called within it.

The program memory has space for a maximum of 200 parts programs and subroutines *) in total.

Program start when there is only one part program in the memory.

```
% LF
N5 G91 G01 X50. F100  LF
N10          Z100.    LF
N15          X-30.    LF
N20          Z-10.    LF
N25          M30      LF
```

Program start when there are two or more part programs in the memory (max. 4 decades 0 ... 9999)

```
% 1357 LF
N5 G91 G01 X50. F100  LF   Determination of preparatory function, path
N10        X-30.      LF   data, speed, direction etc.
N15        Z-10.      LF
N20 M30               LF   M30 or M02 program end with return to program
                           start.
```

*) with basic control, 0,1,2: max. of 20 parts programs and subroutines
with basic control 3: max. of 100 parts programs and subroutines

1.8 Subroutines

Repetitive sequences of functions and movements can be entered as subroutines and called up at any time in the part program or by manual data input. Subroutines, which have to be called up at any time (e.g. grooving) must be programmed incrementally. The subroutine is defined by the subroutine number in two decades or 3 decades with 2 trailing zeros.

L 41200		LF	-	Subroutine 412 Always without block number and M17
NO G91 GO Z-10. F100		LF	-	Designation of preparatory functions,
N5 Z... X 10.		LF		traverse paths, directions and speeds.
.				
N10 X...		LF		
N15	M17	LF	-	End of program with M17 occurs in the last block of the subroutine.

The subroutine is called from a part program or subroutine using the address L. Subroutines may be nested up to 3 times within the part program.

L 412 01

The number of the subroutine must be 2 or 3 decades (001 ...999)[1].

Call, 3 to 5 decades

The number of passes must be programmed in two decades.

No data specification signifies a single pass.

The subroutine call must not be programmed in the same block as M02, M30 or M17.

Cycles have been assigned to L91-L99 and L900-L999 [2].

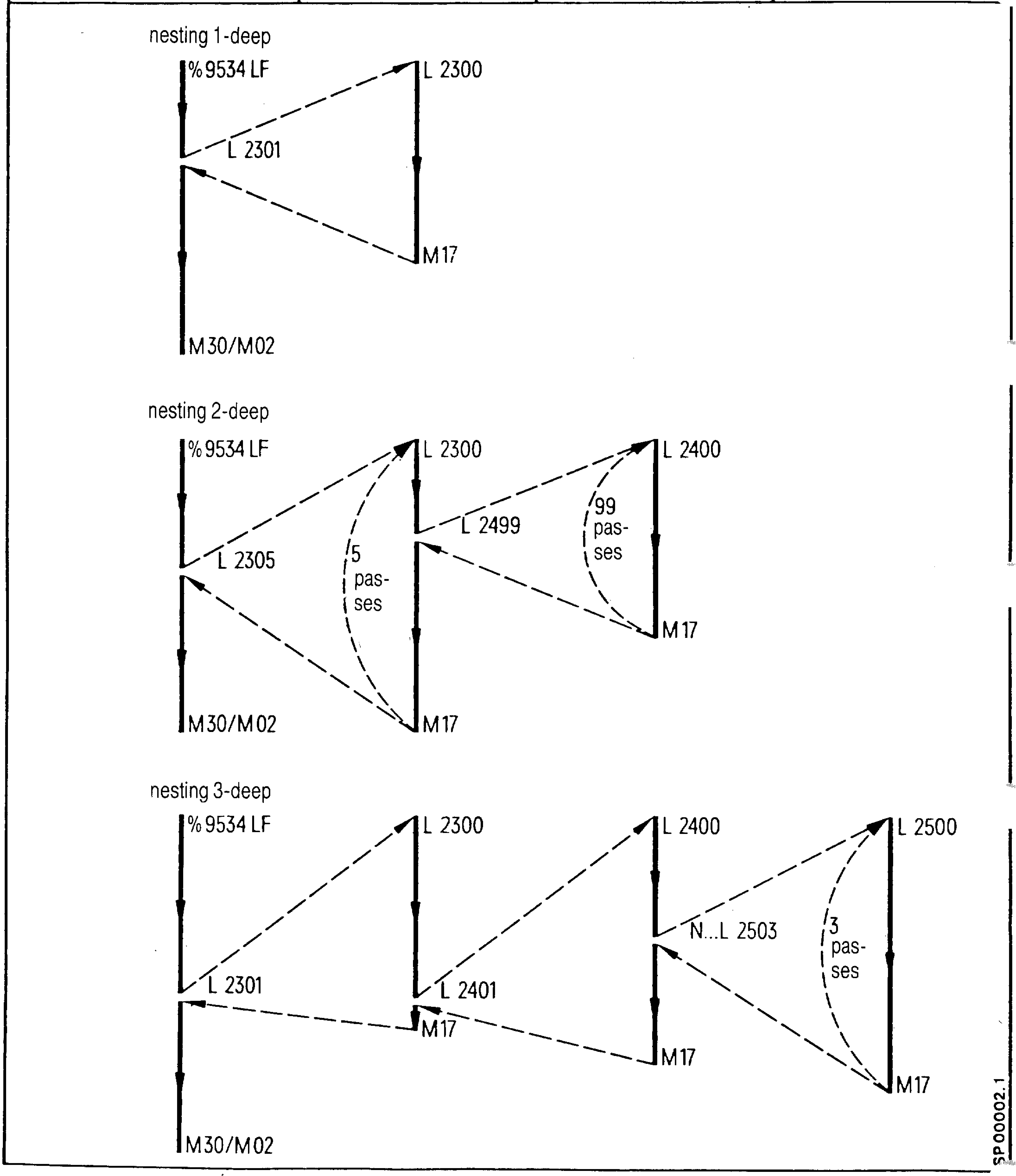
Automatic block search can be made up to the first subroutine.

1) with basic version 0,1,2,3: only 2 decades (01...99)

2) with basic version 0,1,2,3: only L91-L99

1.9 Subroutine call, Subroutine nesting

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



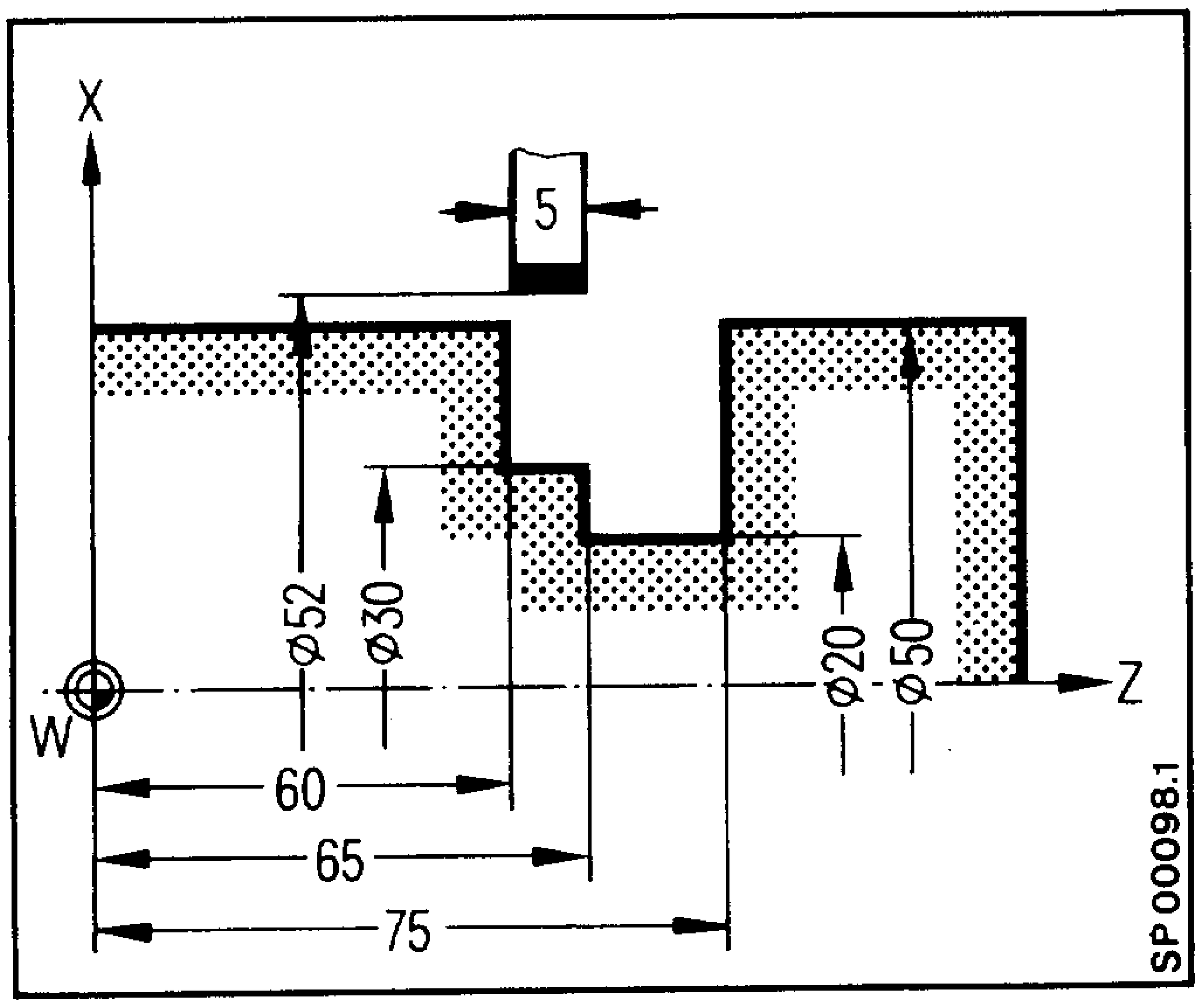
Subroutine nesting

```
% 4011 LF                                  - Part program 4011
N1 G90 G94 F.. S.. T1501 M.. LF
N2 G00 X52. Z60. LF
N3 L2301 LF                                - Call, subroutine 23,
.                                            single pass (L2301)
N90 M30 LF

L2300                                      - Subroutine 23
N1 G91 G01 X-11.LF
N2     G00 X11. LF
N3 L2402        LF                         - Call, subroutine 24,
                                             two passes (L2402)
M17             LF                         - End of subroutine

L2400                                      - Subroutine 24
N1 G91 G00 Z5.  LF
N2 G01     X-16.LF
N3 G00     X 16.LF
N4 M17          LF                         - End of subroutine.
```

Program execution



1.10 Tape formats

```
SUBROUTINES   % {SP} LF | L 2300 LF | N1 .... LF | N2 M17 LF
```

leader — rewind stop and block start — Subroutine 23 — end of subroutine

```
L 2400 LF | N1 .... LF | N2 (DRILLING CYCLE) .... LF | N.. M17 LF
```

Subroutine 24 — (remark) — end of subroutine

```
L 2500 LF | N.. .... LF | N.. .... LF | N.. M17 LF | N.. M02 oder M30 LF
```

Subroutine 25 — end of subroutine — end of subroutine block

```
PROGRAM SHAFT | % {1234} LF | (CARRY OUT MEASUREMENT)
```

leader — part program (remark) 1234 — (remark)

```
N.. .... LF | N.. .... LF | N200 M02 oder M30 LF
```

end of part program — part program

```
% T0 LF | G92 T1 .... LF | G92 T13 .... LF | M02   M30 LF
```

tool offset (TOOL Offset) — end of tool offset block

```
% TE LF | S.....   LF | S.....   LF | M02   M30 LF
```

machine data (TESTING-DATA) — machine data block end

{ }   Characters in brackets can be omitted

SP-subroutine (Sub-Program)

SP 00338.1

% R1 LF | R00 12345.678 LF | R01 5 LF ... | M02 or M30 LF

R parameter 1) — R parameter block end

% H LF | H00 12345.678 LF | H01 678 LF ... | M02 or M30 LF

Background memory 1) — Background memory block end

The sequence in which the punched tapes are input is optional.

Division of the memory into the area for part programs and the area for subroutines is automatic.

Tool offsets are entered in the appropriate memory areas using the designation TO (Tool Offset).

1) Basic Version 4C

1.11 Punched tape format for erasing programs

Main programs and subroutines can be erased in any order via the universal input/output interface with this program.

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	- End identifier M30 or M02

Example:

```
% CL LF % 1 LF        L55 LF    % 1 1200 LF
L11  L79  LF          L81 LF        M30(or M02) LF
```

% CL LF	Erase programs
% 1 LF	Erase program % 1
L55 LF	Erase subroutine L55
% 1 % 1200 LF	Erase programs % 1 to % 1200
L11 L79 LF	Erase subroutines L11 to L79
L81 LF	Erase subroutine L81
M30 (or M02) LF	End of erase block

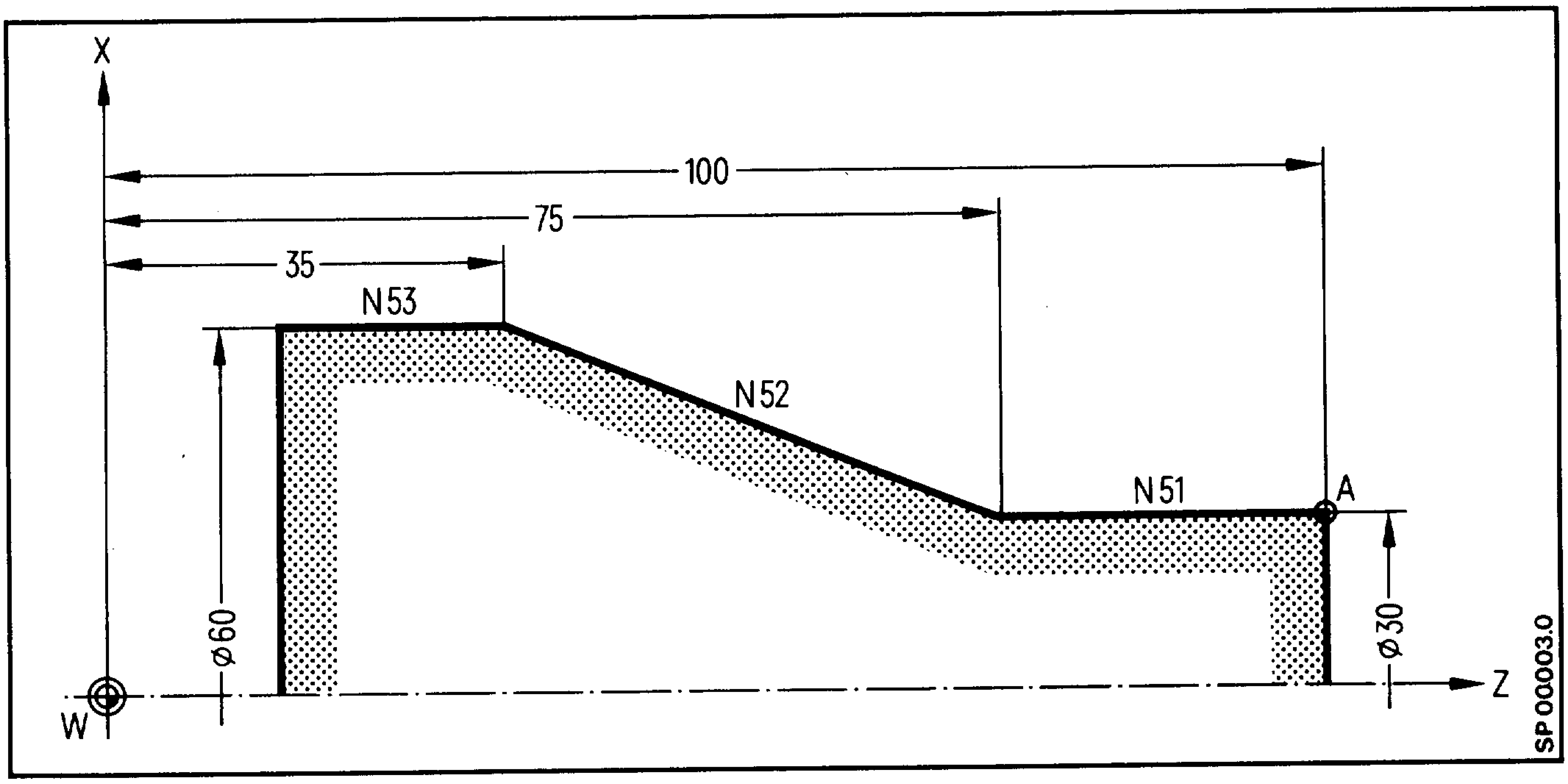
Caution:

Basic version 0,1,2,3:	Subroutines L80 - L99 must be erased individually
Basic version 4 and SINUMERIK 3TT:	Subroutines L80-L99 and L900-L999 must be erased individually. They cannot be erased if cycle lock is on.

2.0 Path data

2.1 X, Z axis commands

The address for the axis commands X and/or Z determines the axis which is to be traversed in accordance with the associated numerical value. When using absolute data input (G90), the values for the X axis are diameter values or radius values (machine parameter). With G91 the values always refer to the radius.



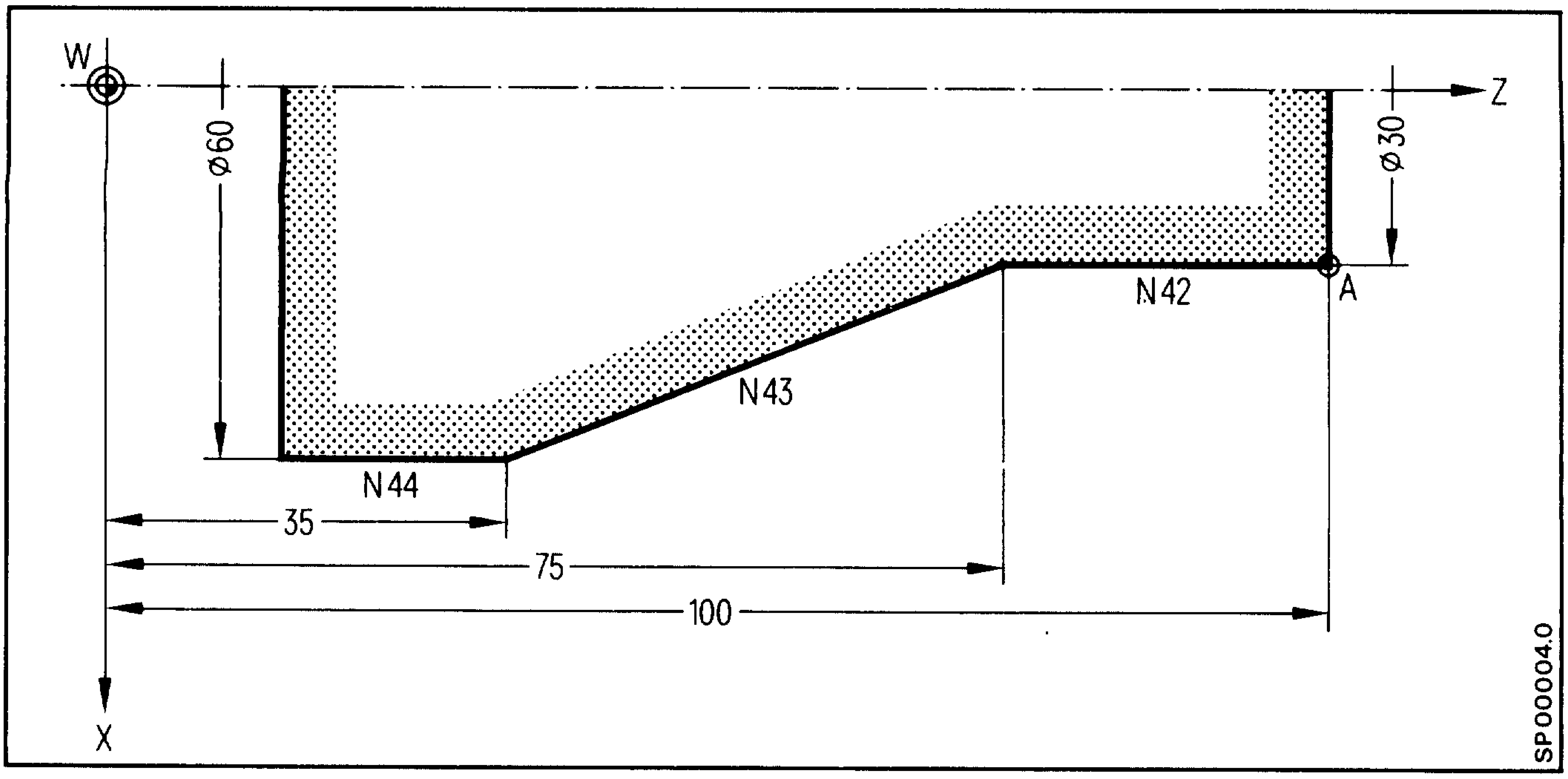
Machining behind the turning axis.

Absolute data input

```
N51  G01  G90  Z75.LF
N52       X60.Z35.LF        X values denote diameter
N53 ...            LF
```

Incremental data input

```
N51 G01 G91 Z-25. LF
N52 X15. Z-40.    LF        X values denote radius
N53 ...           LF
```



Machining in front of the turning axis.

<u>Absolute data input</u>

```
N42 G01 G90 Z75.  LF
N43     X60.Z35.  LF        X values denote diameter
N44 ...           LF
```

<u>Incremental data input</u>

```
N42 G01 G91 Z-25.LF
N43 X15.    Z-40. LF        X values denote radius
N44 ...           LF
```

2.2 Mirror image

Using the input signals "Mirror Image X" and/or "Mirror Image Z" the following values are inverted or interchanged in the controller:

X Axis

Mirror image of

- Programmed axis commands with sign (X values)
- Direction of rotation G02 - G03; G03 - G02
- Tool nose radius compensation G41 - G42 or G42 - G44 (see Section 4.2.2)
- Tool length compensation (see Section 4.2.1)(optional)
- Position of tool cutter point (see Section 4.2.2)

There is no mirror image of

- Zero offsets.
- Tool length compensation (optional)

Z Axis

Mirror image of

- Programmed axis commands with sign (Z values)
- Tool nose radius compensation G41 - G42, G42 - G41
- Direction of rotation G02 - G03; G03 - G02

no mirror image of

- Zero offsets
- Tool length compensation
- Position of the tool cutter point

The mirror image for the X axis is always the axis
The mirror image for the Z axis always refers to the workpiece.

The machine tool manufacturer defines which M function initiates mirror image.

3.0 Preparatory functions

The preparatory functions describe the movement of the machine slides, type of interpolation, type of dimensioning, time-related influences and activate certain operating conditions within the controller.

The preparatory functions are assigned to the groups G1 to G14 (see program key).

Only one preparatory function from each of the 14 groups may be included otherwise only the last value programmed is valid. Preparatory functions designated by * in the program key, must not be included in a block with another function.

The reset data are only effective after switching on the controller, resetting or at the end of the program. They do not have to be programmed.

Modal preparatory functions can only be overwritten by words from the same group.

3.1 G90/G91 Absolute and incremental data programming

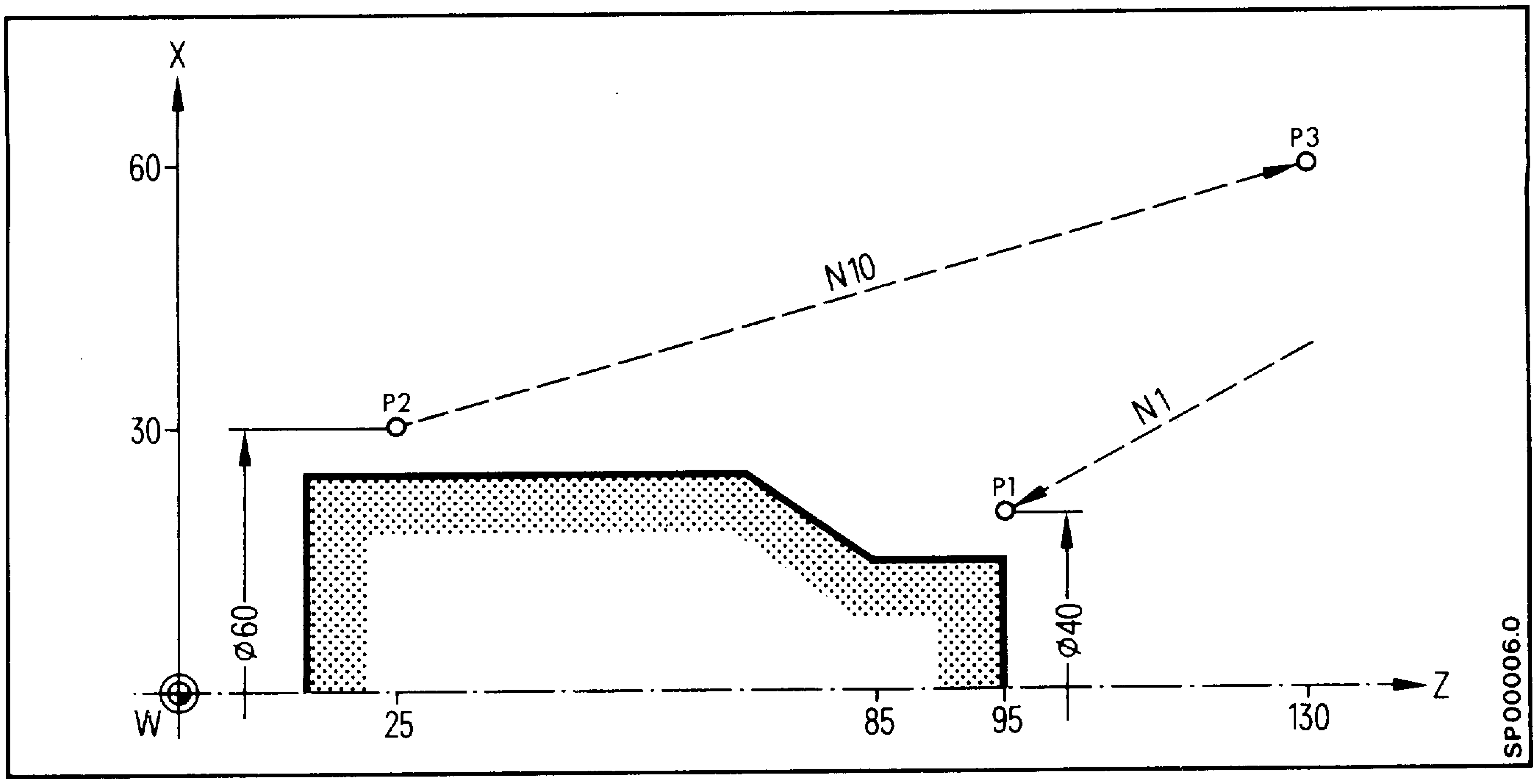
Absolute data input G90

When using absolute data input all data are referred to the fixed zero point of the workpiece.
X values are effective in diameter or radius (machine parameter).

Incremental data input G91

Incremental position data means that each dimension corresponds to the distance to be traversed. For this reason one refers to incremental dimensions or incremental data input. The numerical value of the axis command indicates the distance to be traversed in order to reach the end position. Incremental data are preferred for use with subroutines which have to be called up in different operating positions of the machine (recessing, etc.).

Tool offsets are taken into account for both incremental and absolute dimension programming, even with G91 in the first block. The first sentence after coordinate transformation within a program has the same effect as the first block in the program (see also block increment calculation in Section 8.2.3).



Absolute data input G90

.

.

N 1 G90 G00 X 40. Z95. ... Tool travels from any position to the point P1

N10 ...

Incremental data input G91

.

.

N1 G90

N10 G91 G00 X30. Z105. .... Tool travels from P_2 to P_3.

3.2 G00 Rapid traverse

The distance programmed in a block with G00 is traversed at the highest possible rate, i.e. at rapid traverse, in a straight line. At the same time the controller monitors the axes in order to prevent them exceeding the maximum permissible rate (machine parameter).

The rapid traverse preparatory function (G00) initiates automatic exact stop (G09). When G00 is programmed, the value of the feedrate programmed under address F remains stored and can be reactivated with G01, for example.

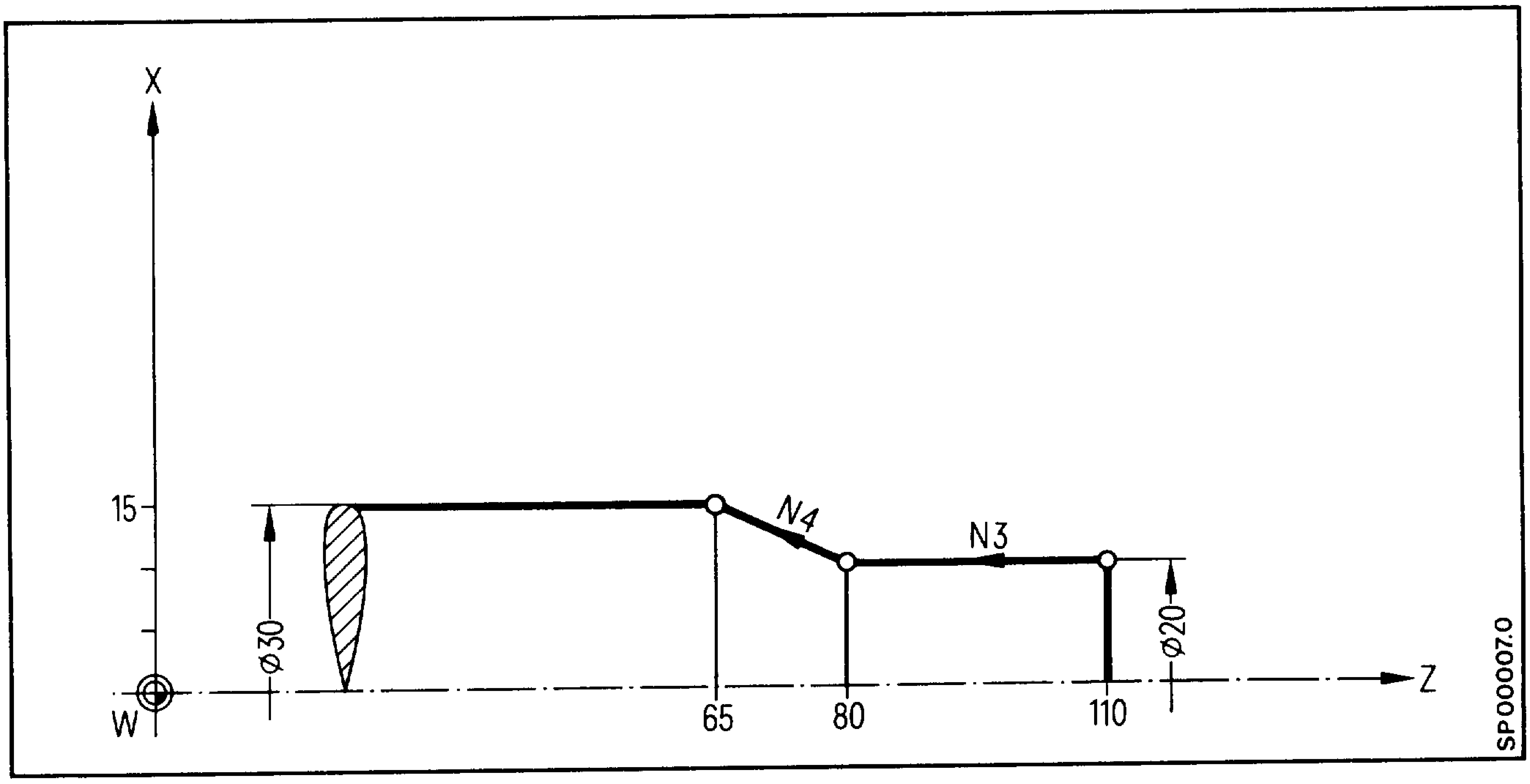
Example: Position approach using rapid traverse. (see G90/G91).

```
% 1234
N1  G90  G00  X20.  Z95.  LF
```

N1	Block number
G90	Absolute data input
G00	Rapid traverse
X20.Z95.	Position (target)
LF	End of block

3.3 G01 Linear interpolation

The tool traverses at the preset feedrate along a straight line to the required final position.
Paraxial and traverse movements at any angle may be executed.



Absolute data input

```
N2 ...
N3 G01 G90 Z80. F10.   LF
N4     X30. Z65        LF
.
```

Incremental data input

```
N2 ...
N3 G01 G91 Z-30.F10.      LF
N4            X5. Z-15.   LF
```

3.4 G02/G03 Circular interpolation

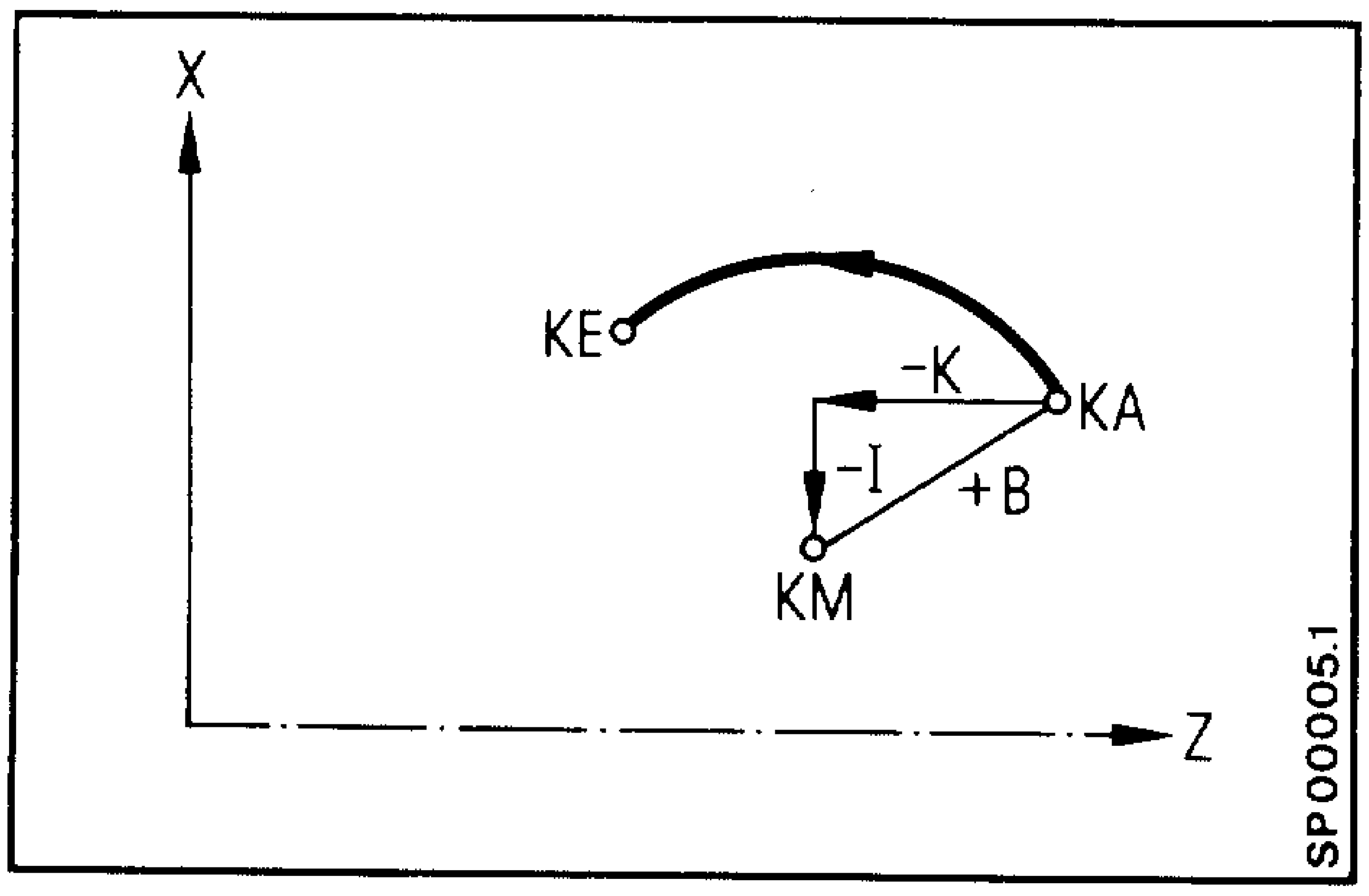
Together with axis commands, the interpolation parameters determine the circle or arc. The starting point "KA" of the circle or arc is determined by the previous block and the end point "KE" is fixed by the axis values X and Z. The circle centre "KM" is determined:

a) Either by using the vectors I and K with sign over a range from 0 to 360°. I in X direction, K in Z direction: The sign is determined by the coordinate direction from the starting point to the centre point.

b) Or directly by using the radius B (basic version 4)

+B angle less than or equal to 180°
-B angle greater than 180°

Do not program radii if the traverse angle is 0° or 360°. Thus circles must be programmed using the interpolation parameters I and K.



An interpolation parameter I or K with value 0 need not be programmed. If the end point coordinate has not changed with respect to the circle start, it does not have to be programmed. For a full circle, at least one axis must be programmed (X0 or Z0).

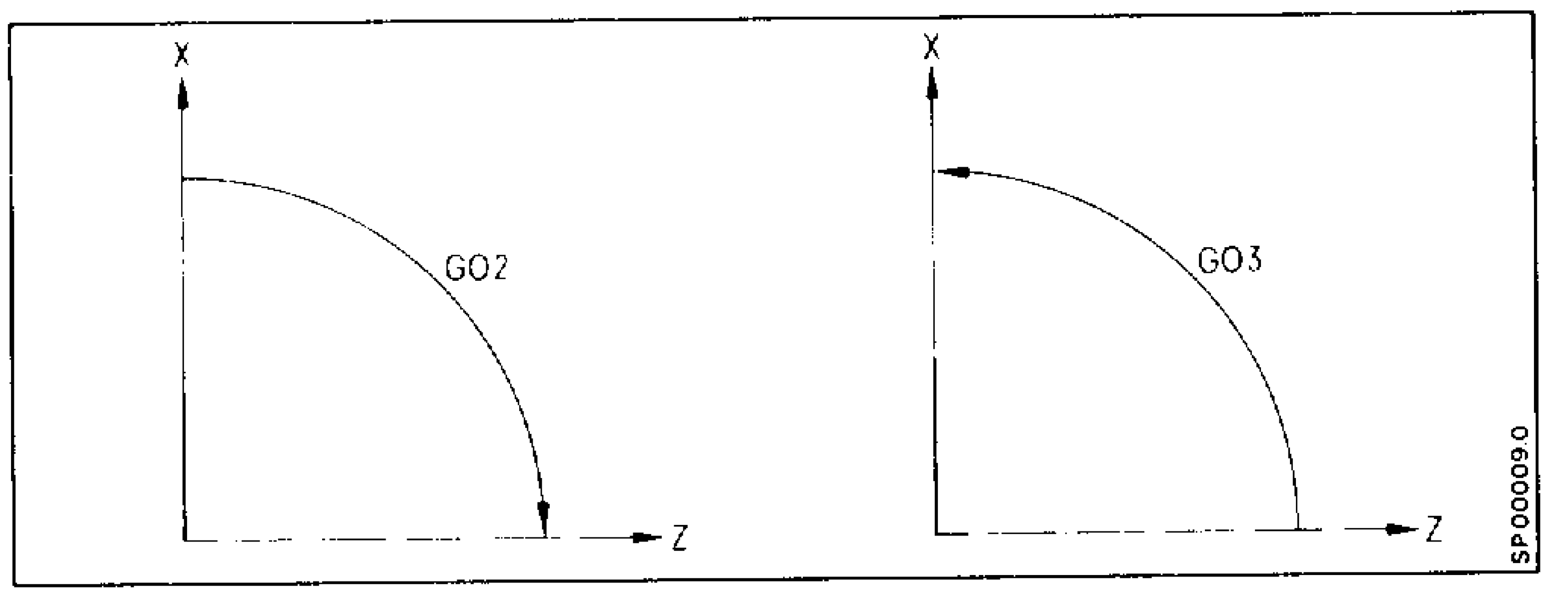
G02/G03 determine the traverse direction of the circle fixed by X, Z, I, K, or R.

Right-hand co-ordinate system (DIN 66025)

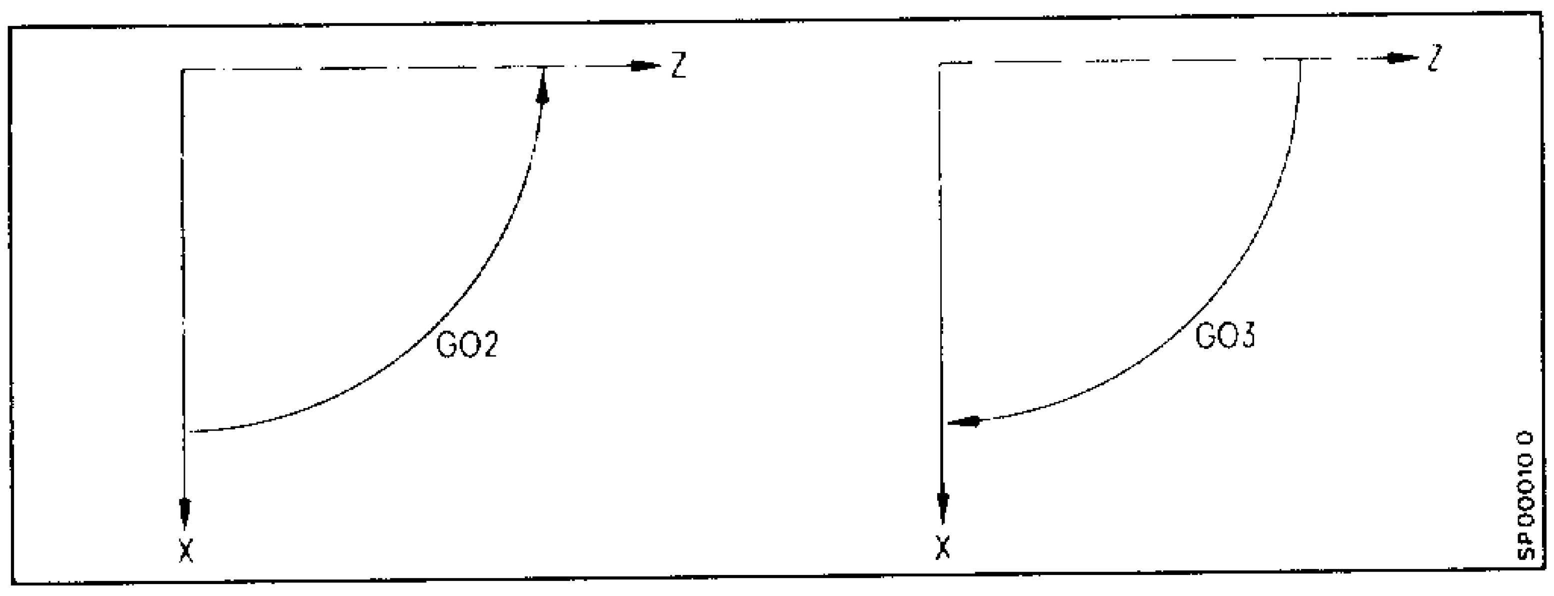
G02 in clockwise direction

G03 in counter-clockwise direction

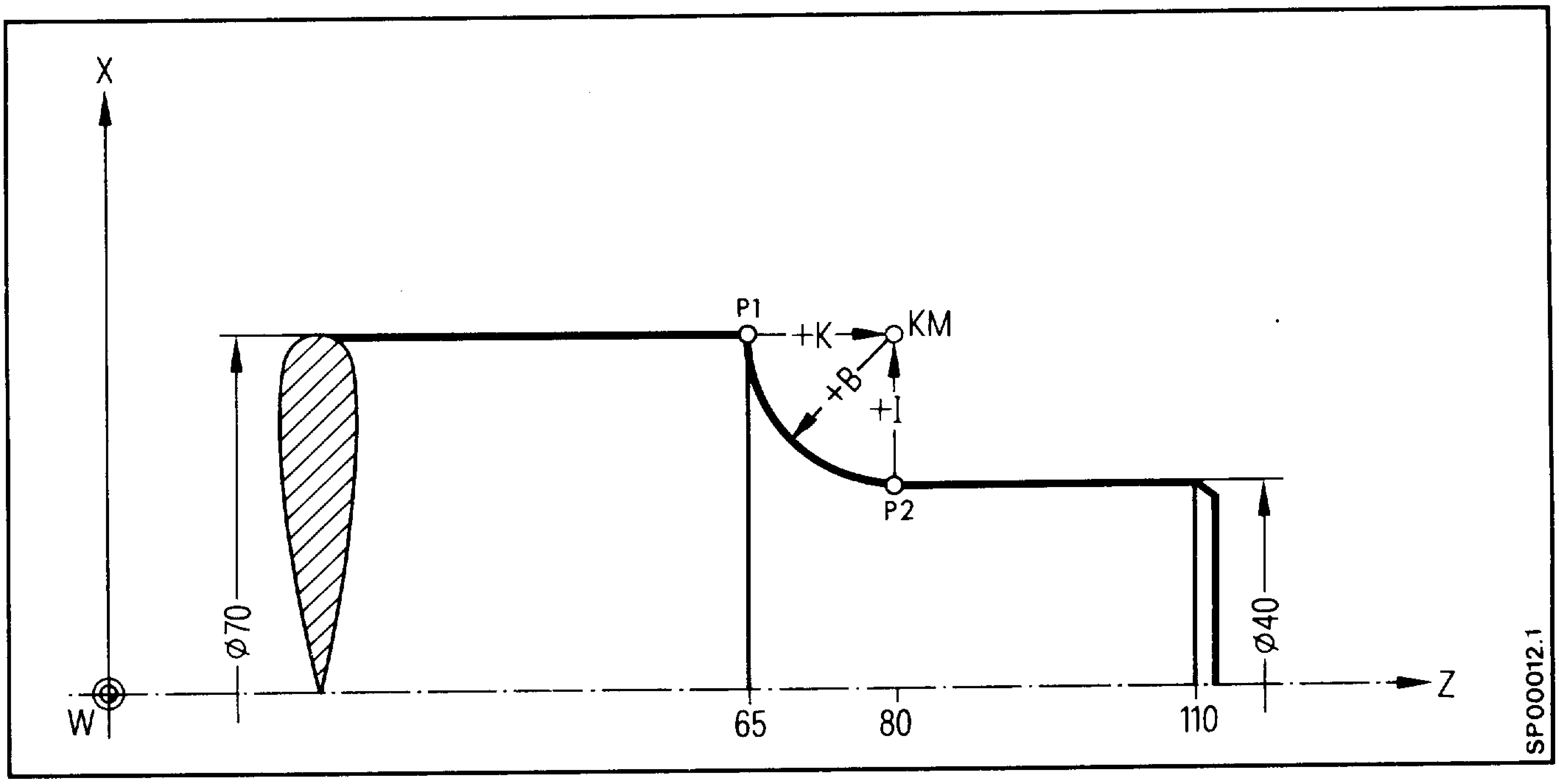
Operating area



Behind the turning axis



In front of the turning axis



N5 G03 G90 X40.Z80.K15.I0. LF - Tool traverses from P1 to P2

N10 G02 X70.Z65.K0. I15.LF - Tool traverses from P2 to P1

or

N5 G03 G90 X40.Z80.B+15.LF - Tool traverses from P1 to P2

N10 G02 X70.Z65.B+15.LF - Tool traverses from P2 to P1

3.5 G33 Thread cutting with constant pitch

Single pass or multi-pass turning, facing and conical threads with constant pitch can be machined using G33.

G33 links the main spindle speed to the feedrate. An encoder generates 4096 pulses per spindle revolution and these are decoded in the controller and fed to the feed drives. Thus the feedrate is directly linked with the spindle speed and no feedrate programming is required.

During threadcutting the feedrate override switch, the "Feed off" button, the spindle speed override switch, and the "Single Block" mode are all non-operational. However, the feedrate under F remains stored and is re-activated the next time G01, G02 and G03 are programmed.
In order to produce threads requiring several passes the feedstart is initiated by the zero mark of the encoder. This ensures that the threadcutting is always started at the same angular position of the workpiece relative to the tool. All cuts must be carried out at the same spindle speed in order to avoid following errors.

The direction of rotation and spindle must be programmed before the threadcutting block in order to allow the spindle to reach its operating speed.

The thread length is entered under path addresses X and/or Z, taking into account the acceleration distance and overrun distance in which the feedrate is increased or decreased.

The values can be entered as absolute or incremental dimensions.

The thread pitch is entered under addresses I and K. For longitudinal threads the pitch is entered under K, for facing threads under I and for taper threads under I and K. I and K values must always be entered as incremental dimensions without sign.
Thread pitch: 0.001 mm to 400.000 mm, up to a maximum of
100000 rpm.

Right or left-hand threads are programmed by specifying spindle rotation direction M03, M04.

Thread cutting with different speeds
(Basic version 4B OPTION E35)

This function requires no additional operator input.
The speed should be constant within a cut. This function ensures that the thread cut at high speed is the same as for a cut a low speed.
There is no longer an offset caused by the following error.

3.5.1 Thread cutting with constant pitch on variable angle tapers

The angle of the taper on which the thread is to be cut can be changed in steps. In the case of longitudinal threads, this permits a smooth run-out. Each taper cut is programmed as in Section 3.5.8.

3.5.2 Multi-start threads

3.5.2.1 Multi-start threads with axial starting point offset

Thread cutting always begins with the synchronization point of the encoder zero mark. Feed is only enabled when this signal is received from the digital encoder. An offset can be programmed for the position of the starting point for the thread. It is thus possible to cut threads using the right or left flank of the cutter as well as varying the offset to cut single, double or multi-start threads. A single pass for a multi-start thread is programmed in the same way as the single start thread.

When the fist pass has been completed, the starting point is offset by h and the next pass is initiated.

$$h = \frac{\text{Thread pitch}}{\text{Number of passes}}$$

See also the example for "Multi-start threads" (3.5.7).
The individual passes must be executed at the same spindle speed in order avoid different following errors.

3.5.2.2 Multi-start threads using spindle-related starting angle offset (basic version 4B, Option E35)

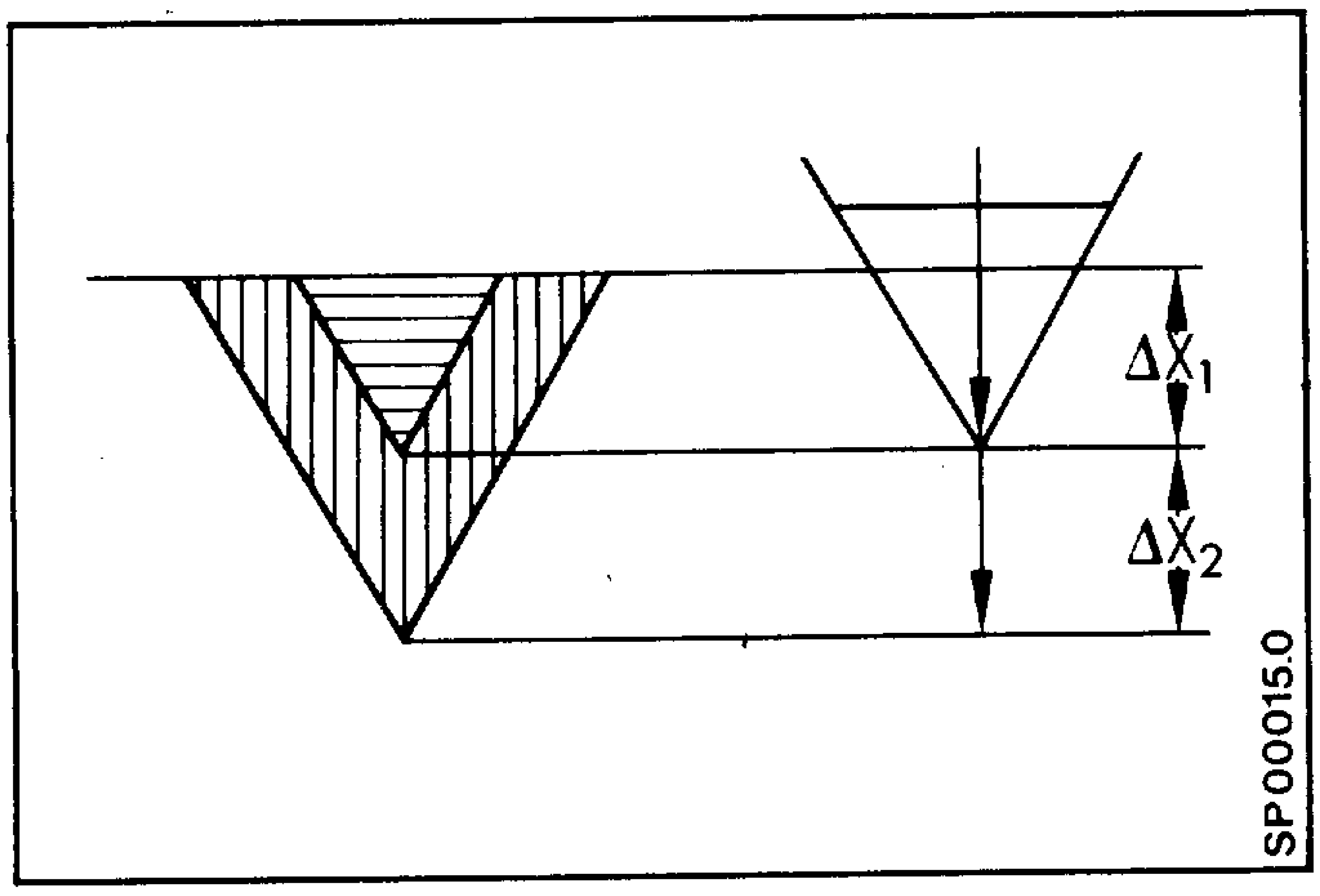
A multi-start thread can be cut with this function. The spindle-related starting angle for the pass should be programmed with G92 A.. . "A" represents the letter for the angle.

Example: Three-start thread (0°, 120°, 240°)

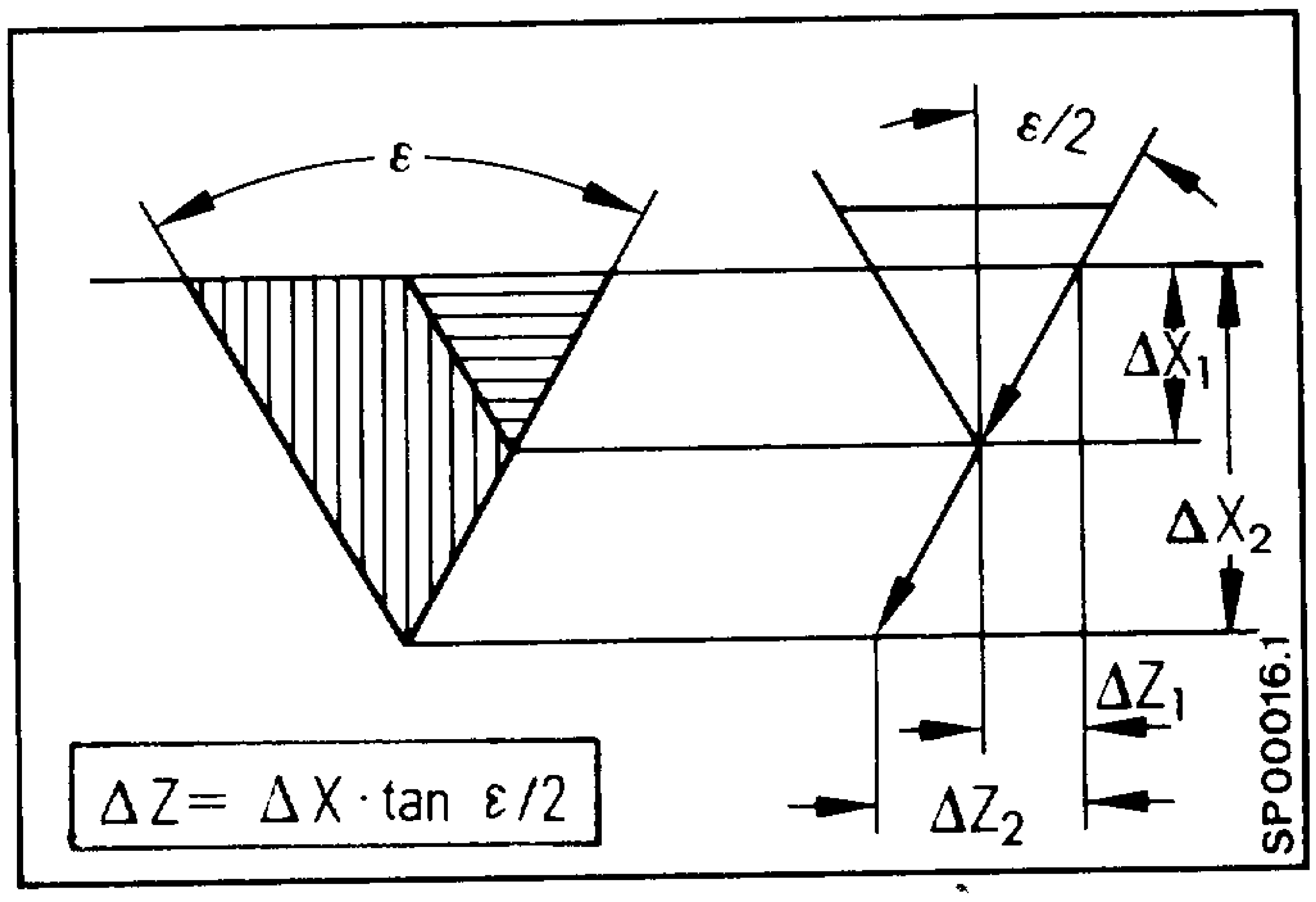
1st pass with G92 A0
2nd pass with G92 A120
3rd pass with G92 A240

3.5.3 Infeed directions

The tool can approach at right angles to the cutting direction or along the flank.



Infeed direction "at right angles to the cutting direction"



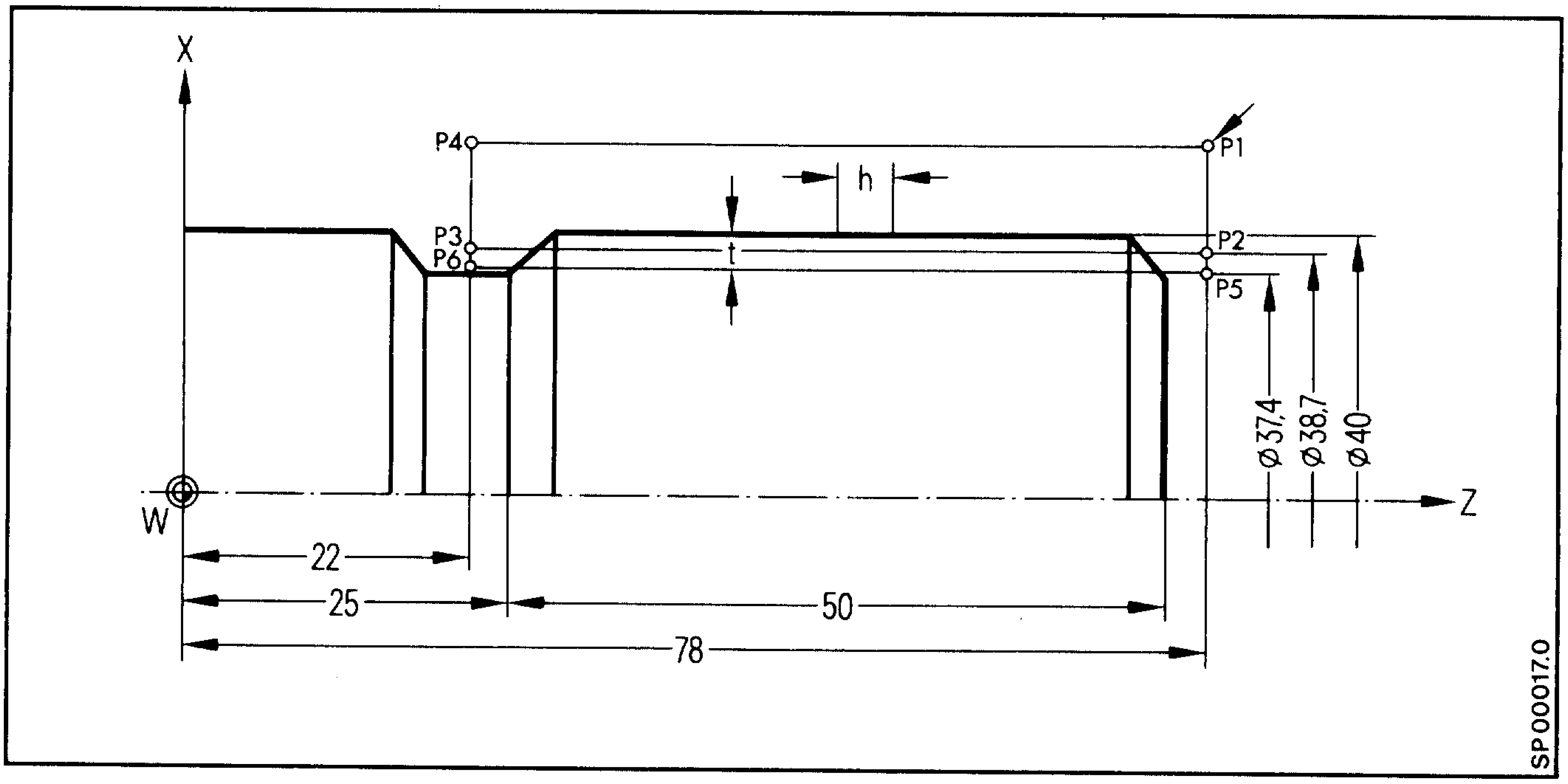
"Infeed along flank".

Z = X . tan E/2

3.5.4 Single start, cylindrical thread

Pitch h = 2 mm; thread depth t = 1.3 mm

Radial infeed direction



Absolute data input

N20	G90		S...		LF	
N21	GOO	X 46.	Z 78.		LF	(P1)
N22		X38.7			LF	(P2)
N23	G33		Z 22.	K 2.	LF	(P3)
N24	GOO	X 46.			LF	(P4)
N25			Z 78.		LF	(P1)
N26		X 37.4			LF	(P5)
N27	G33		Z 22.	K 2.	LF	(P6)
N28	GOO	X 46.			LF	(P4)

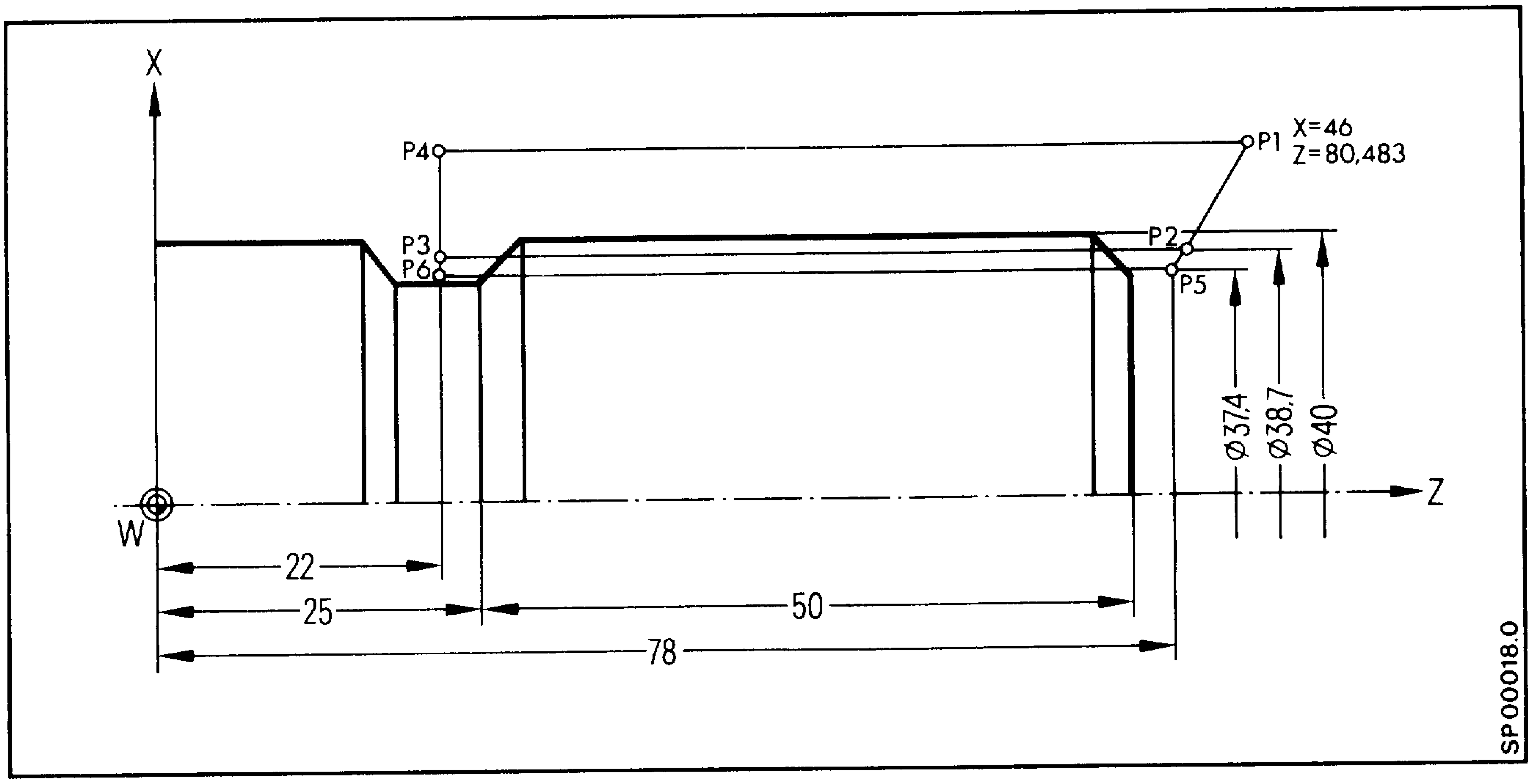
Incremental data input

N20	G91		S...		LF	
N21	GOO	X-...	Z-...		LF	(P1)
N22		X-3.65			LF	(P2)
N23	G33		Z-56.	K 2.	LF	(P3)
N24	GOO	X 3.65			LF	(P4)
N25			Z 56.		LF	(P1)
N26		X-4.3			LF	(P5)
N27	G33		Z-56	K 2.	LF	(P6)
N28	GOO	X 4.3			LF	(P4)

3.5.5 Single start longitudinal thread with constant pitch

Pitch h = 2 mm; Thread depth t = 1.3 mm; infeed angle = 60^{o}

Infeed direction along flank.



X = (46-38.7)/2 = 3.7	X = 4.3 3.65 + 1.3/2
Z = 3.7 x tan /2	Z = 4.3 . tan /2
Z = 3.7 . tan 30^{o}	Z = 4.3 . tan 30^{o}
Z = 3.7 . 0.5774	Z = 4.3 . 0.5774
Z = 2.136	Z = 2.483

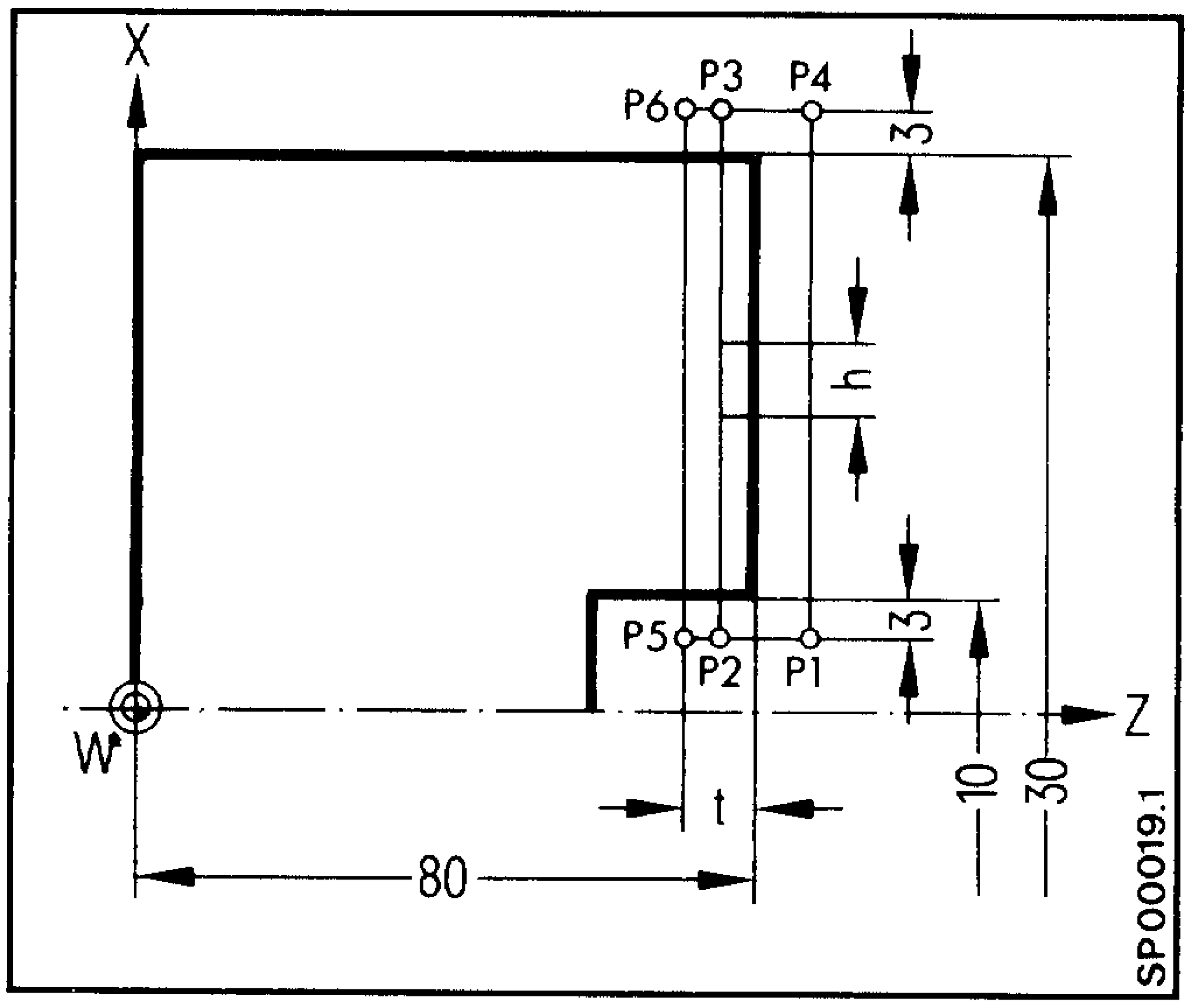
Absolute data input

```
N33 G90          S...            LF
N34 GOO X 46.    Z 80.483        LF (P1)
N35     X 38.7   Z 78.347        LF (P2)
N36 G33          Z 22.     K 2.  LF (P3)
N37 GOO X 46.                    LF (P4)
N38              Z 80.483        LF (P1)
N39     X 37.4   Z 78.           LF (P5)
N40 G33          Z 22.     K 2.  LF (P6)
N41 GOO X 46.                    LF (P4)
```

3.5.6 Single start facing thread with constant pitch

Pitch h = 2 mm; thread depth t = 1.3 mm

Infeed direction at right angles to cutting direction.



Absolute data input

```
N41 G90           S...
N42 G00 X  4.     Z 82.              LF (P1)
N43               Z 79.35            LF (P2)
N44 G33 X 36.              I 2.      LF (P3)
N45 G00           Z 82.              LF (P4)
N46     X  4.                        LF (P1)
N47               Z 78.7             LF (P5)
N48 G33 X 36.              I 2.      LF (P6)
N49 G00           Z 82.              LF (P4)

N50 etc.
```

Incremental data input

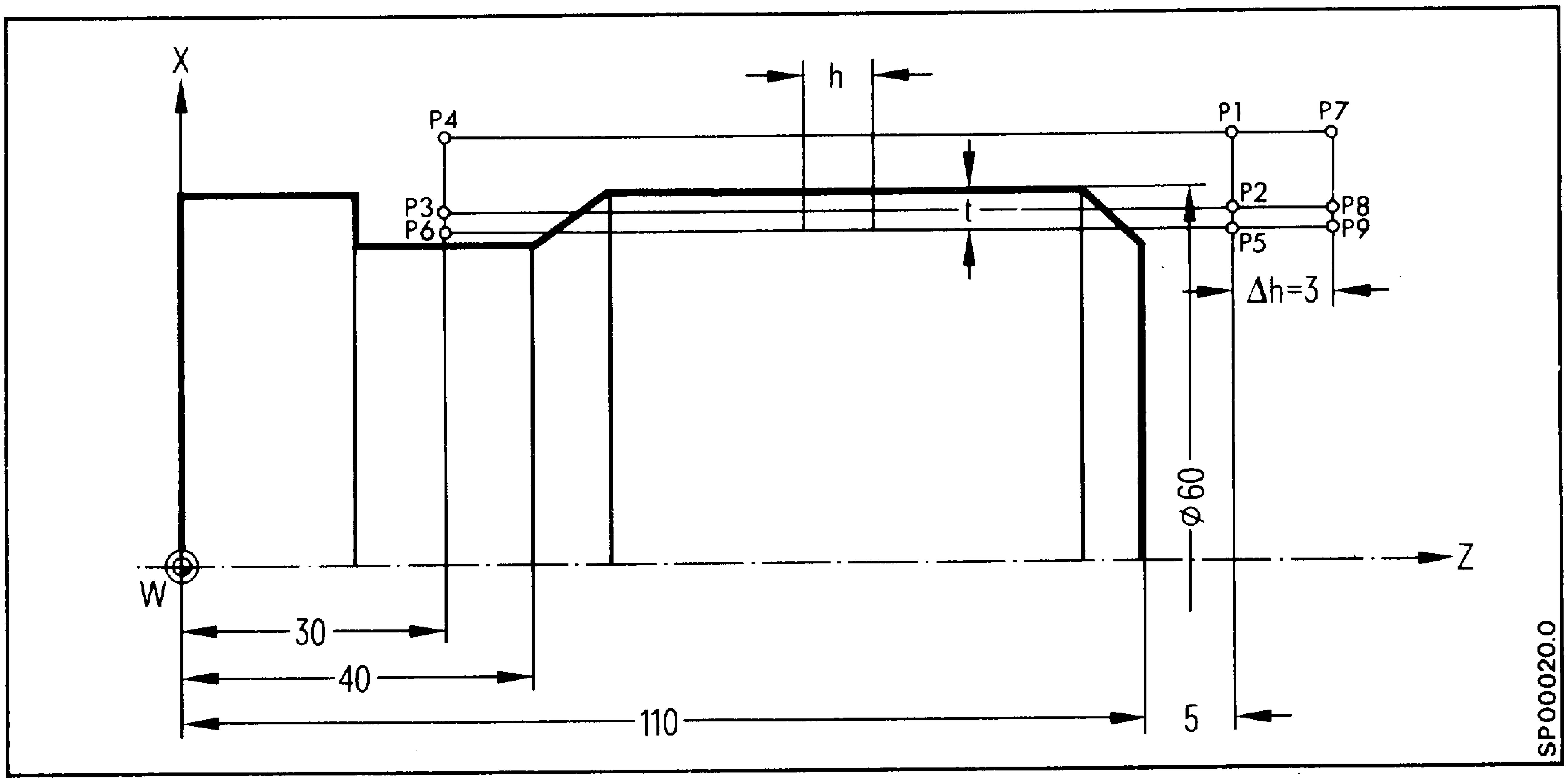
```
N41 G91                              LF
N42 G00 X-...     Z-....             LF (P1)
N43               Z-2.65             LF (P2)
N44 G33 X 16.              I 2.      LF (P3)
N45 G00           Z 2.65             LF (P4)
N46     X-16.                        LF (P1)
N47               Z-3.3              LF (P5)
N48 G33 X 16.              I 2.      LF (P6)
N49 G00           Z 3.3              LF (P4)
N50 etc.
```

3.5.7 Multi-start thread with constant pitch

Radial infeed direction; pitch h = 6 mm; thread depth t = 3.9 mm; double start.

In the example, each thread start is machined in two passes. After the first start has been machined the second start is machined with the starting point displaced by h.

h = thread pitch/number of starts = 6/2 = 3 mm



Absolute data input

```
N35 G90 S...
N36 G00 X 66    Z 115.         LF (P1)
N37     X 56.                  LF (P2)
N38 G33         Z 30.    K 6.  LF (P3)
N39 G00 X 66.                  LF (P4)
N40             Z 115.         LF (P1)
N41     X 52.2                 LF (P5)
N42 G33         Z 30.    K 6.  LF (P6)
N43 G00 X 66.                  LF (P4)
N44             Z 118.         LF (P7)
N45     X 56.                  LF (P8)
N46 G33         Z 30.    K 6.  LF (P3)
N47 G00 X 66.                  LF (P4)
N48             Z 118.         LF (P7)
N49     X 52.2                 LF (P9)
N50 G33         Z 30.    K 6.  LF (P6)
N51 G00 X 66.                  LF (P4)
```

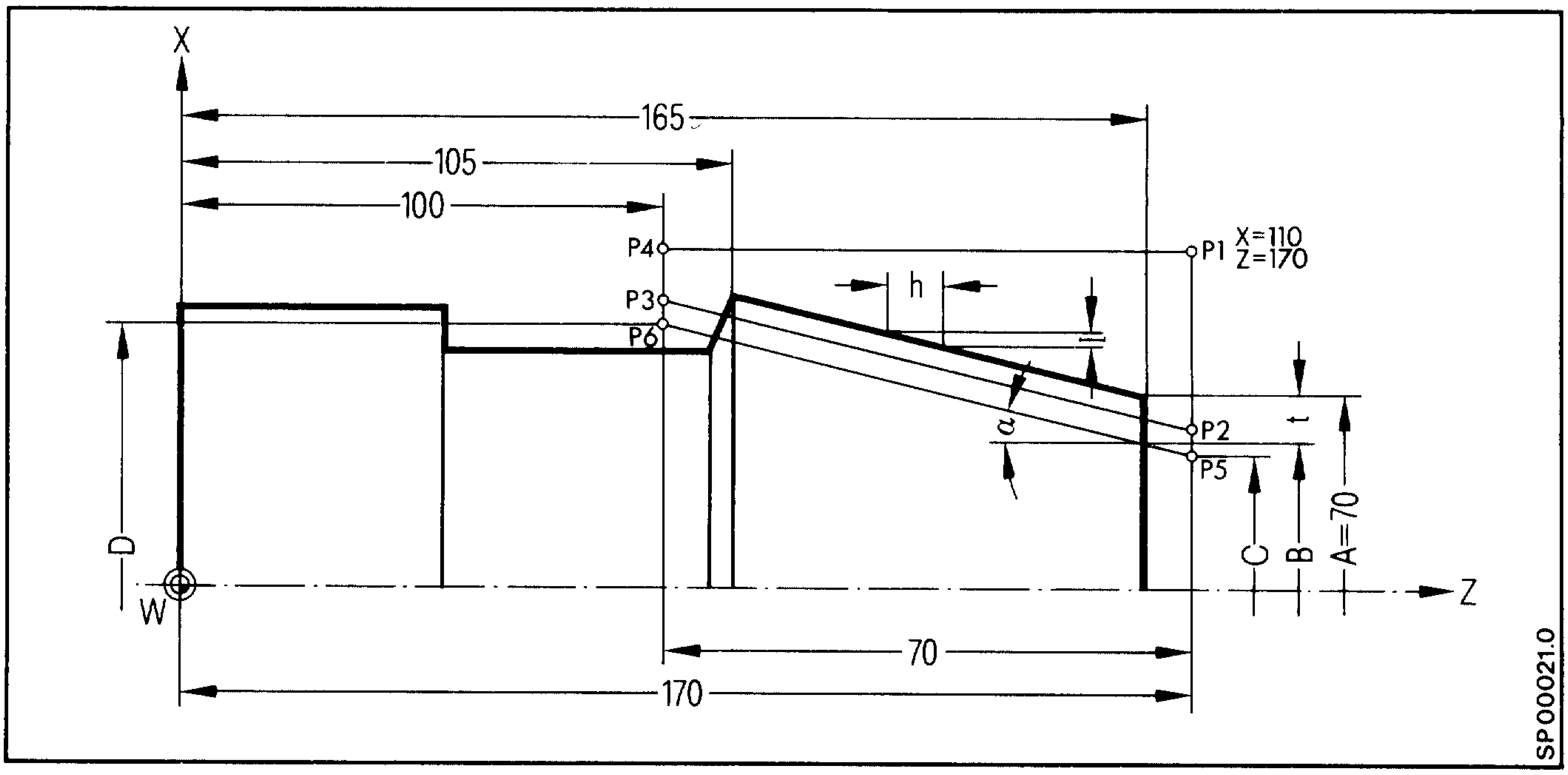
3.5.8 Single start taper thread with constant pitch

Radial infeed direction

Pitch h = 5 mm; thread depth = 1.73 mm; = 15°

Both end point coordinates must be written.

The pitch h is entered using K.



Calculation of the start and end point coordinates for the thread

A = 70

B = A - 1.73 . 2

B = 70 - 3.46

B = 66.54 mm

C = B - (5. tan) . 2

C = 66.54 - (5 . tan) . 2

C = 63.86 mm

D = C + (70 . tan) . 2

D = C + (70 . 0.2679) . 2

D = 63.86 + 37.506

D = 101.366 mm

h = K = 5 mm

I = K . tan

I = 5 . 0.2679

I = 1.34 mm

Absolute data input

```
N31 G90 S...                                  LF
N32 G00 X 110        Z 170.                   LF (P1)
N33     X 65.86                               LF (P2)
N34 G33 X 103.366    Z 100. I 1.34 K 5.       LF (P3)
N35 G00 X 110.                                LF (P4)
N36                  Z 170.                   LF (P1)
N37     X 63.86                               LF (P5)
N38 G33 X 101.366    Z 100. I 1.34 K 5.       LF (P6)
N39 G00 X 110.                                LF (P4)
```

Calculation of the points P2 and P3

X(P2) = C + 2 mm

X(P2) = 63.86 mm + 2 mm

X(P2) = 65.86 mm

X(P3) = D + 2 mm

X(P3) = 101.366 mm + 2 mm

X(P3) = 103.366 mm

3.6 Variable pitch threads

3.6.1 Variable pitch thread without G34, G35

The thread pitch can be varied in several consecutive blocks. In any one block the pitch is constant. The constant pitch range may be less than one revolution because blocks following consecutive thread cutting blocks do not wait for the marker pulse of the encoder.

3.6.2 G34 Thread cutting (increasing pitch) *

The thread pitch per turn increases by the value programmed under F, up to the maximum possible pitch.

Example:

```
| N ...   G 34  G90   Z 17. K 2 .  F 0.1    LF |
                          ---    -----
                            ]       ]
Initial pitch 2 mm     _____]       ]
```

Pitch variation + 0.1 mm per turn

i.e. after 5 turns the thread pitch is:

2 mm + 5 x 0.1 mm = 2.5 mm

3.6.3 G35 Thread cutting (decreasing pitch) *

The thread pitch per turn decreases by the value programmed under F, down to the minimum possible pitch.

```
| N ..  G 35   G90   Z 17.   K 200.  F 0.5   LF |
                              ------  -----
                                 ]      ]
Initial pitch 200 mm      _______]      ]
```

Pitch variation -0.5 mm per turn

i.e. after 10 revolutions the thread pitch is:

200 mm - 10 x 0.5 mm = 195 mm

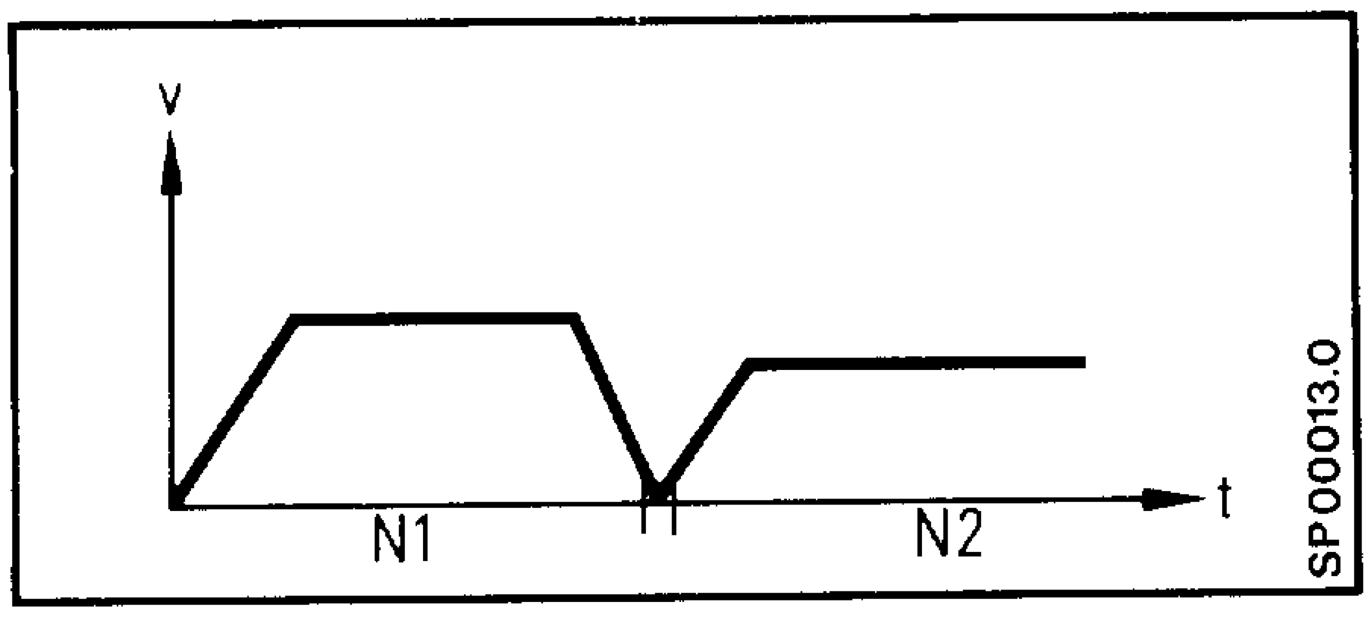
Examples for G34/G35 are analogous to those for G33. It is merely necessary to program the thread cutting blocks differently, if the pitch for the relevant thread is to increase or decrease. The maximum pitch variation is 16 mm. It should be programmed without sign.
With a known initial and final pitch, the value should be calculated as follows:

$$F = \frac{\text{Initial pitch}^2 - \text{final pitch}^2}{2 \times \text{thread length}}$$

*) With Basic Version 4B, 4C

3.7 G09 Speed Decrease

Using the function G09 it is possible to traverse exactly to a target

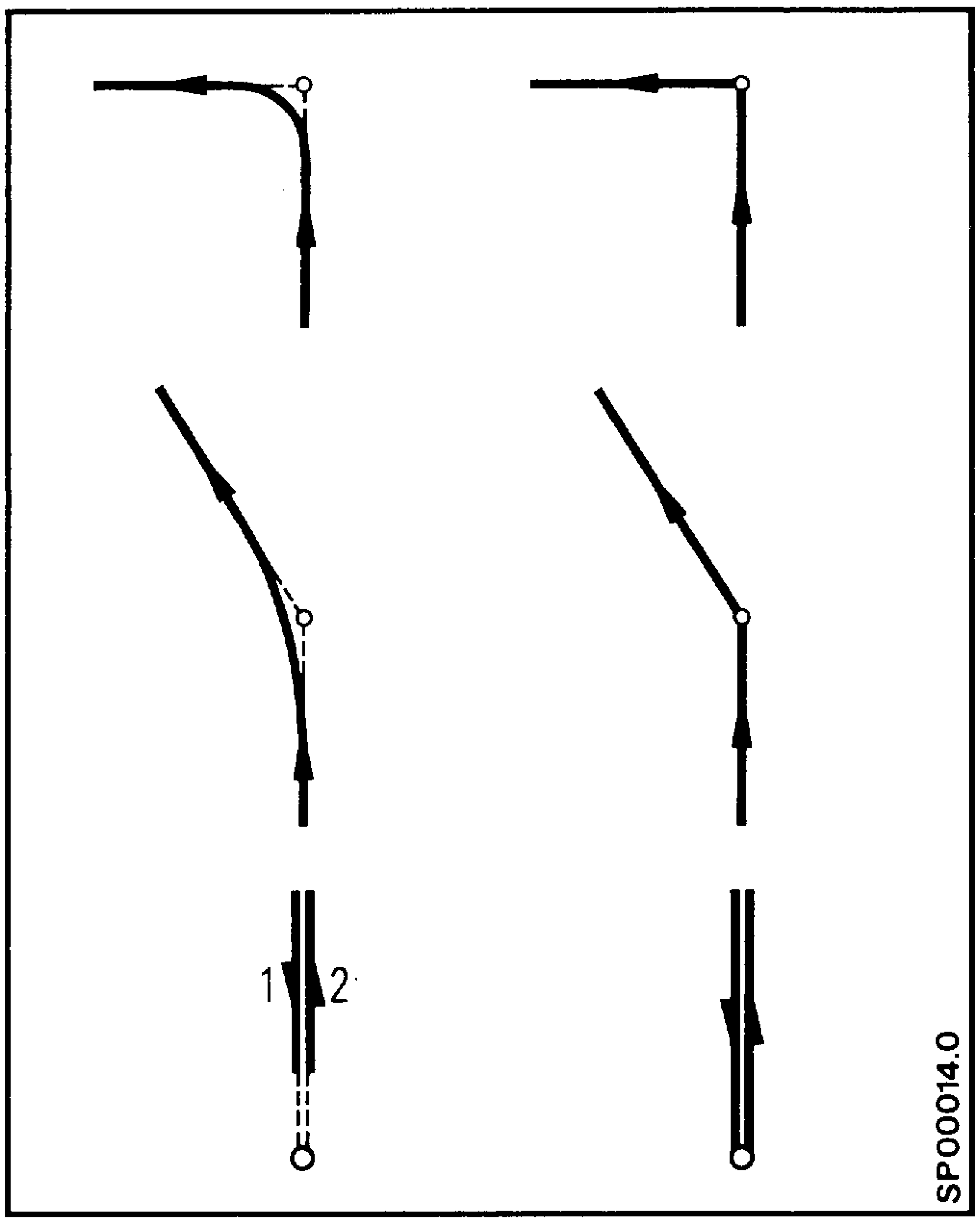


position (within the exact stop limit). In this case the feedrate and following error are reduced to zero.

The function G09 can be used, e.g. when sharp corners are to be machined or for cutting grooves or reversing direction. In blocks with G00 it is not necessary to write G09 since this is contained in G00. G09 is effective in blocks.

Examples of direction changes with and without speed reduction.

Without G09 With G09



Change from G64 to G00 without programming G09

Machine data MD428 bit 3 makes it possible to change from continuous path control G64 to rapid traverse G00 without programming an exact stop G09. Up to now G09 was mandatory. This does not apply to the transition from G33 to G00.

3.8 G04 Dwell Time

The dwell time is normally programmed using address X.
The range is: 0.001 to 99999.99 sec[1)]

With the exception of G04 no further preparatory functions may be written in a block containing a program dwell.

Example:

N.. G04   X11.5   LF
X11.5   Dwell   11.5 s always without sign

Dwells may be required for relief cuts or possibly during changes in speed and machine switching functions (steadies, tail stock etc.).
G04 is effective in blocks.

1) with basic version 0,1,2,: 0.001 and 9999.999 sec

3.9 G70/71 Input Sytems

G70 Inch input system

G71 Metric input system

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

If a value which does not correspond to the basic state, is input to the system, it must be preceded by the associated G function. The controller 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 the basic state when the block is executed.

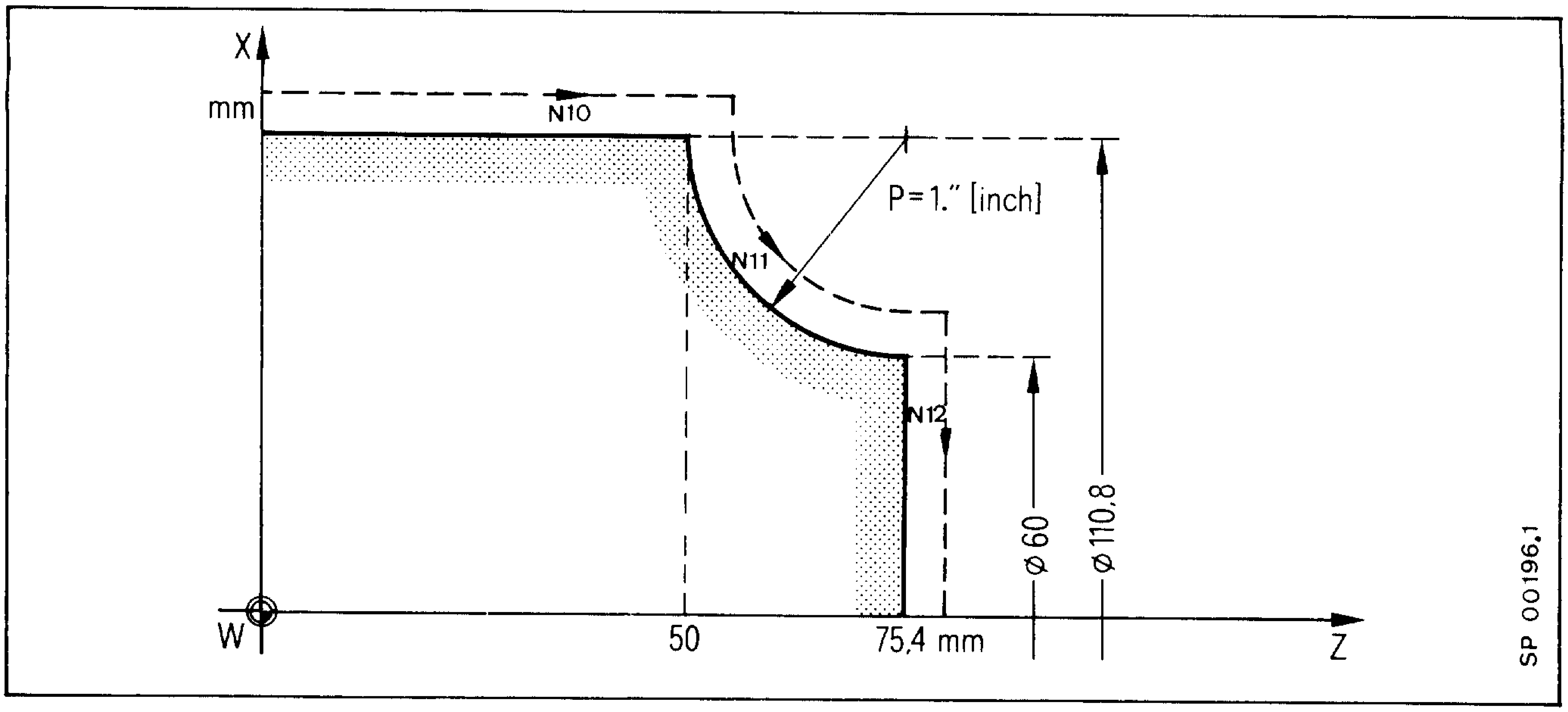
A different input system to that defined in the basic state can be programmed for one or more blocks or for a complete program. The required G function must be programmed in the first block and the reset state must be programmed after the last block (the reset state is automatically selected after Mo2, M30 program end).

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

Actual value display (also setpoint/actual value difference)
G54 to G57, G59 and external zero-point shift
Cutting speed G94, G95
Tool offset

The following are dependent on programmed G70 or G71 functions:
Path information X,Z
Interpolation parameters I,K
Chamfers/Radii B-/B
Parameters, which are not included under path information, interpolation parameters, or chamfers/radii.

Example: G71 - Reset state (metric)



.

```
N10 G91 Z50
N11 G03 G70 Z1. X-1. K1. I0. LF
N12 G01 G71 X-30             LF
```

.

.

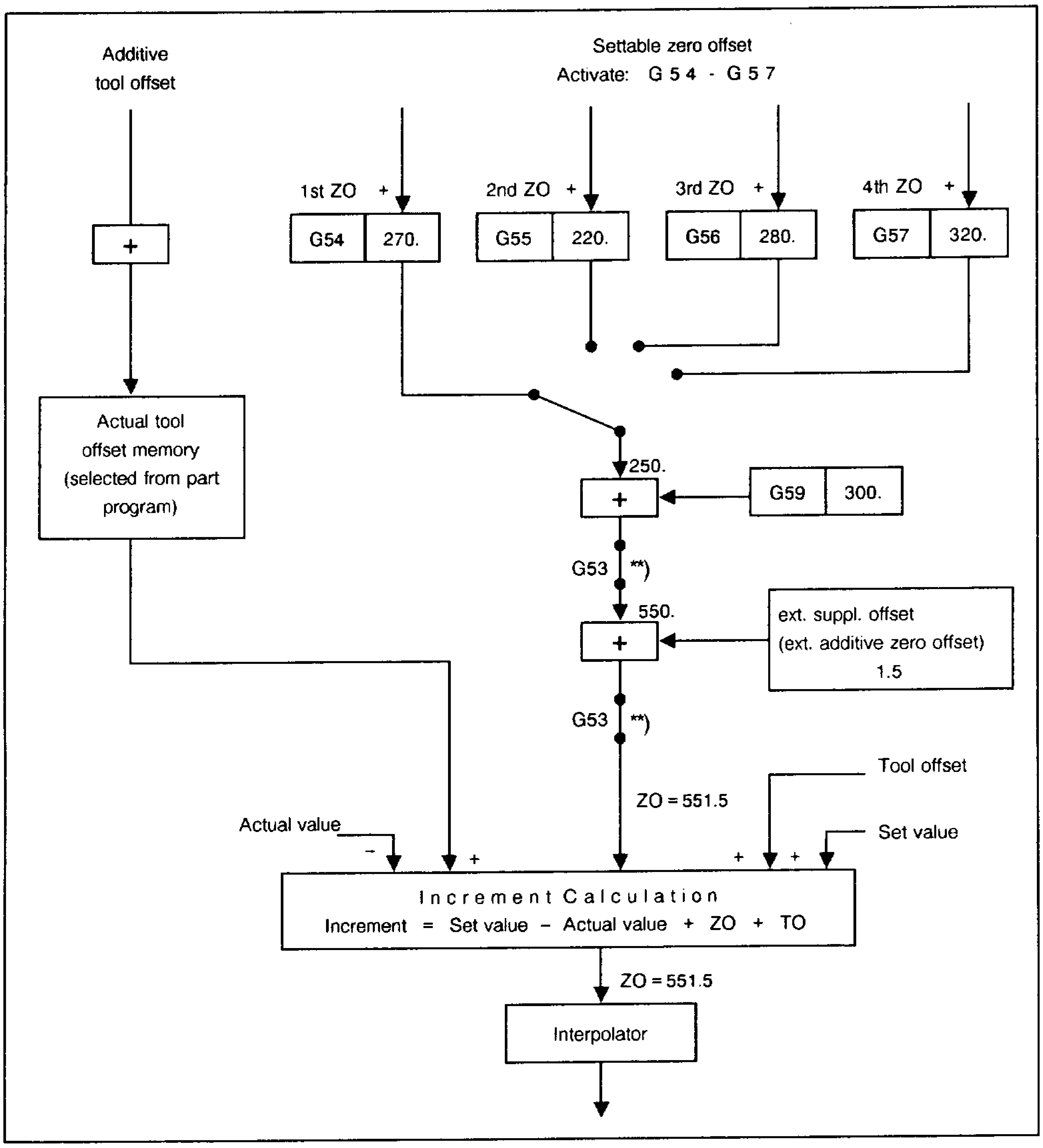
.

3.10 Zero offsets (ZO)

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

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

Example of zero offset in one axis and a selected group*:



*) Basic control 4B, 4C

**) Can be selected via machine datum

3.10.1 G54 to G57 Settable zero offset

Values for the zero offset for each axis are entered manually into the controller via the operator panel.
Absolute and incremental data blocks are used to calculate the block end point when the relevant axis is programmed. Any change in zero offset is also taken into account in incremental data blocks.
Example: Change fom 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).

4 or 12* settable zero offsets (G54 to G57) can be selected for the individual axes.
Any external additive zero offset applied from the interface controller (external additional compensation) is added to the value, for example, of the ZO memory selected via G54; the same applies to any programmed ZO.
The result corresponds to the entire zero offset.

Selecting the adjustable zero offset

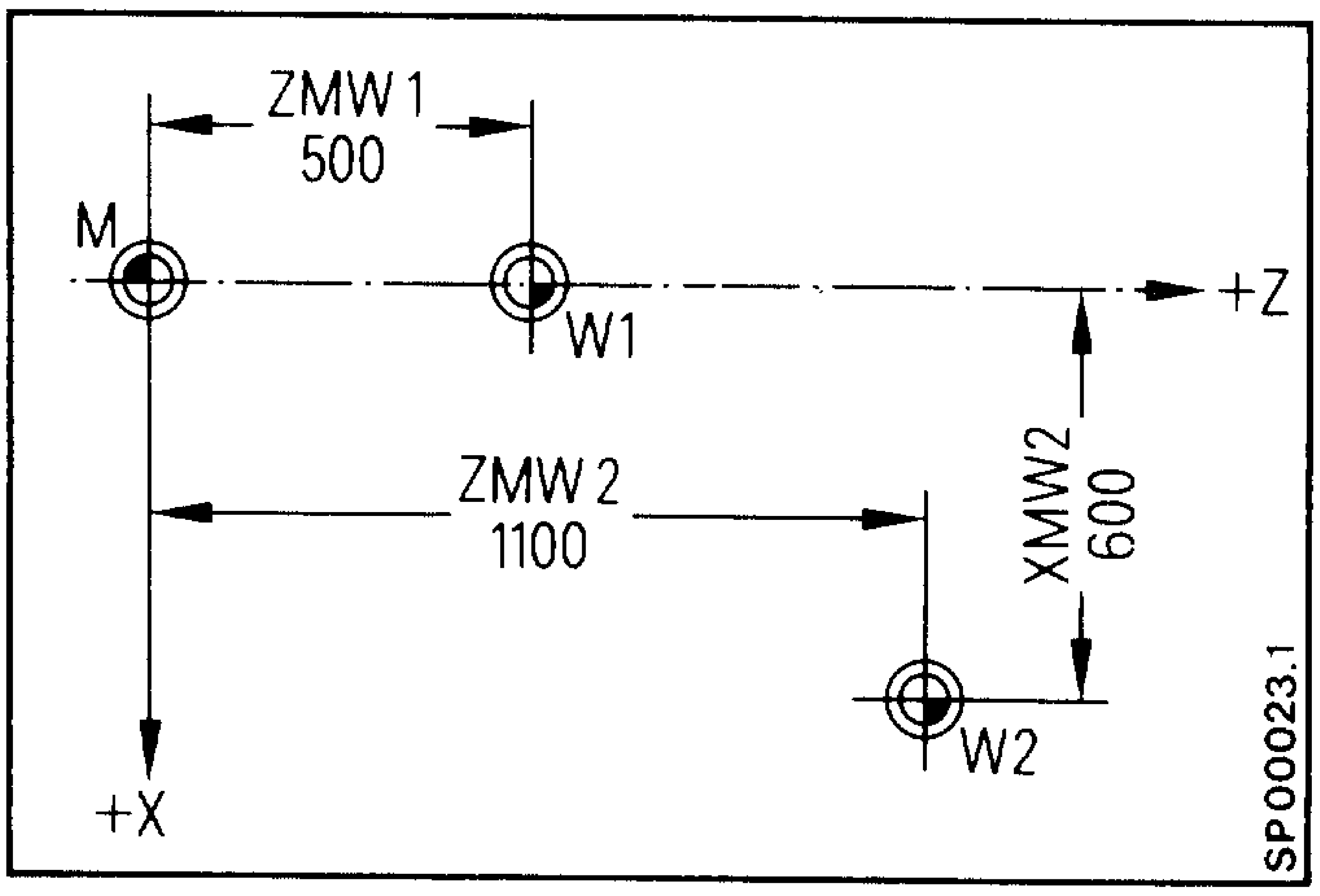
Example:

.

N150 G54 X... Z... LF - Select zero offset 1

.

N180 G55 X... Z... LF - Select zero offset 2



G54: Input values XMW = 0
ZMW = 500
G55: Input values XMW = 600
ZMW = 1100

*) Basic Version 4B, 4C

Selecting the Settable Zero Offset
(Basic version 4B)

Selection of the settable zero offsets (Z01-Z012) is carried out through G54 to G57 and through additional interface signals for the 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

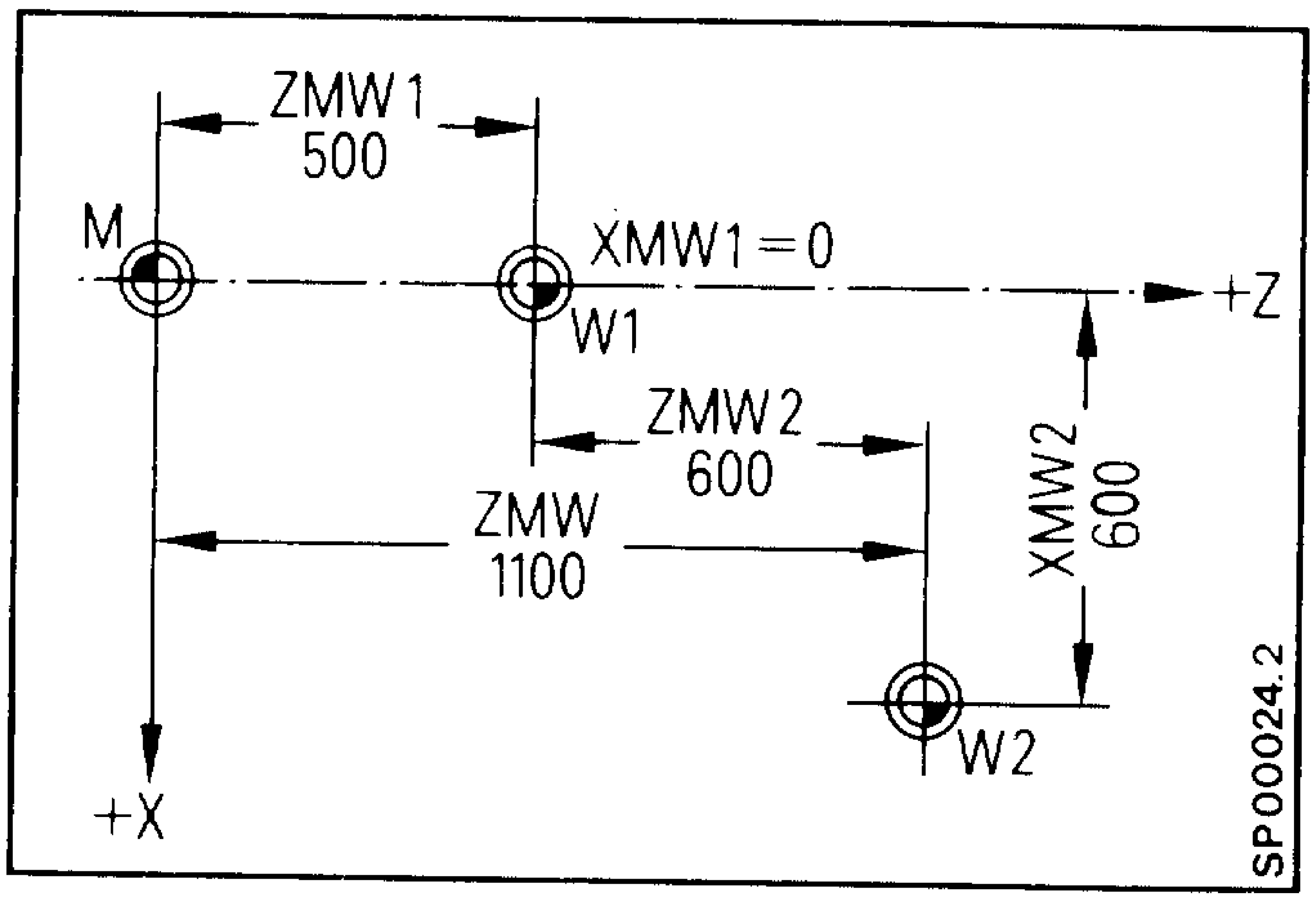
Activation of the interface signals, for example, via M function. For clarity, L99 (empty buffer) must be programmed and the cutter radius offset must be cancelled.

Example:

```
N10 G40 X... LF
N15 M..  ... LF   Selection of the zero offset groups 1 to 3
                  with an M function
N20 L99..... LF   Empty buffer
                  (see Sections 5.6 and 7.2)
N25 G56..... LF   Selection of the zero offset
```

3.10.2 G59 Programmable additive zero offset

G59 together with address X and Z can be used to program an additional zero offset. These programmed values are then added during the calculations to the values of the settable zero offset.



Settable zero offset:

Input values XMW_1 = 0

ZMW_1 = 500

Programmable additive zero offset:

Input values XMW_2 = 600

ZMW_2 = 600

Total zero offset:

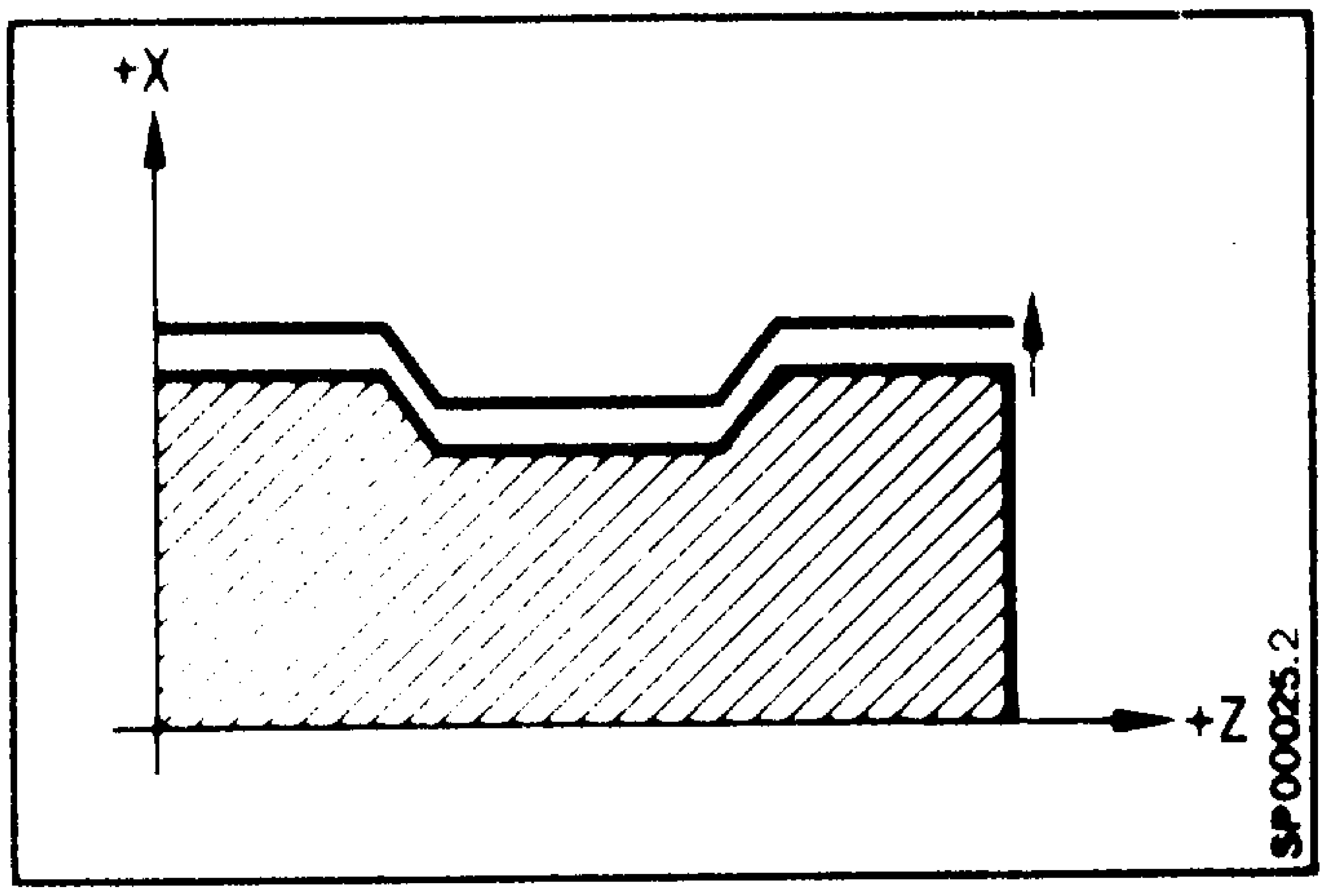
XMW = 600

ZMW = 1100

Application example

The contour has been programmed exclusively in absolute data. In order to achieve a finishing depth cut the entire contour can be offset in the coordinate direction X using programmable zero offsets (additive).

Select:	N .. G59 X.... LF
Cancel:	N .. G59 X 0 LF



Programmable additive zero offset only in X

Following end of program M02, M30 or program interrupt the values of the programmed zero offsets for X and Z are automatically cleared since they will be reset by the program upon program restart.

3.10.3 G53 Cancelling the zero offset

G53 suppresses in blocks the coordinate displacement from the machine zero point to the workpiece zero point achieved by

- settable ZO (G54 - G57)
- programmable additive ZO (G59)
- external additive ZO (external additional compensation, optional according to preset value) * Basic version 4 B

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	T1300	LF	- cancellation of tool offset
N1233	G53 X... Z...	LF	- cancellation of all ZOs and traverse to position

3.11 G94/G95/G96 Feedrate F

The feedrate F can be programmed in mm/min or in mm/rev.
G95 (F in mm/rev.) is always a reset state.

The programmed speed is maintained on the path of the cutter point "P" (imaginary tool point) also with tool nose radius compensation.

Using a feedrate override switch on the operator panel the programmed feedrate can be adjusted between 1% and 120%. The 100% position corresponds to the programmed value. During threadcutting the feedrate input is omitted and the feedrate override switch is non-operational.

Example:

N5... G94 F10. LF

G94 G94 Feedrate F in (mm/min)
G95 Feedrate F in (mm/rev)
G96 Feedrate F in (mm/rev) and
Constant cutting speed S in (m/min)

F10. Feedrate
LF

The relationship between the rotational feedrate and the spindle speed and limit values is shown in the figure entitled "Limit data for rotational feedrate" (see Section 8.2.4).

3.12 G96 S.. Constant cutting speed
G97 Constant speed

G96 S.. Constant cutting speed

Depending on the programmed cutting speed, the controller determines the appropriate spindle speed for the latest turning diameter.

N5 ... G96 S... LF

G96 Constant cutting speed

S... in (m/min)

LF

The correlation between turning diameter, spindle speed and feedrate ensures optimum matching of the program to the machine, the workpiece material and the tool.

Ther zero point for the X axis must be the turning axis. This is ensured by reference point approach.
When calculating the spindle speed for constant cutting speed, the following values are taken into account:

- Actual value
- Tool length compensation
- Zero offset in X direction
  - Settable zero offset G54, G55, G56, G57 (ZO)
  - Additive ZO G59
  - External additive compensation

The workpiece must not be shifted out of the turning axis by a zero offset in the X axis. The latter can be used to shift the tool holder.
G96 S... always refers to the workpiece-related actual value.

G97 Constant speed

The constant cutting speed is frozen with function G97.
The last actual speed value is used as the constant speed.

G97 is used to avoid undesirable changes in speed in the event of intermediate blocks in the X direction without machining. Additionally, an undesired change of speed, caused by the S word acknowledgement dependent on the PLC cycle time, can be avoided with G97.

Example of change from G95 to G96

```
.
.
.
N40 S120 M3
N45 G1 G95 X51 F1.2
N50 G97 S20
N55 G96
N60 X30 Z15
.
.
.
```

Change of gear

A gear is selected and used for constant cutting speed. Gear shifting at a suitable point in the program is possible at any time.

3.13 G92 S.. Limitation of spindle speed when using G96

It may become necessary (e.g. with constant cutting speed G96) to limit the rise in spindle speed at some point, i.e. to continue machining at a certain point with a constant spindle speed. The limitation is programmed in a separate block using address S in rev/min prior to the program section in which it is to be effective. The function G92 S... may be used more than once in the program.

```
N..   G92   S300   LF
```

G92

S300 Spindle speed limited to 300 rev/min.

LF

No further commands in the same block

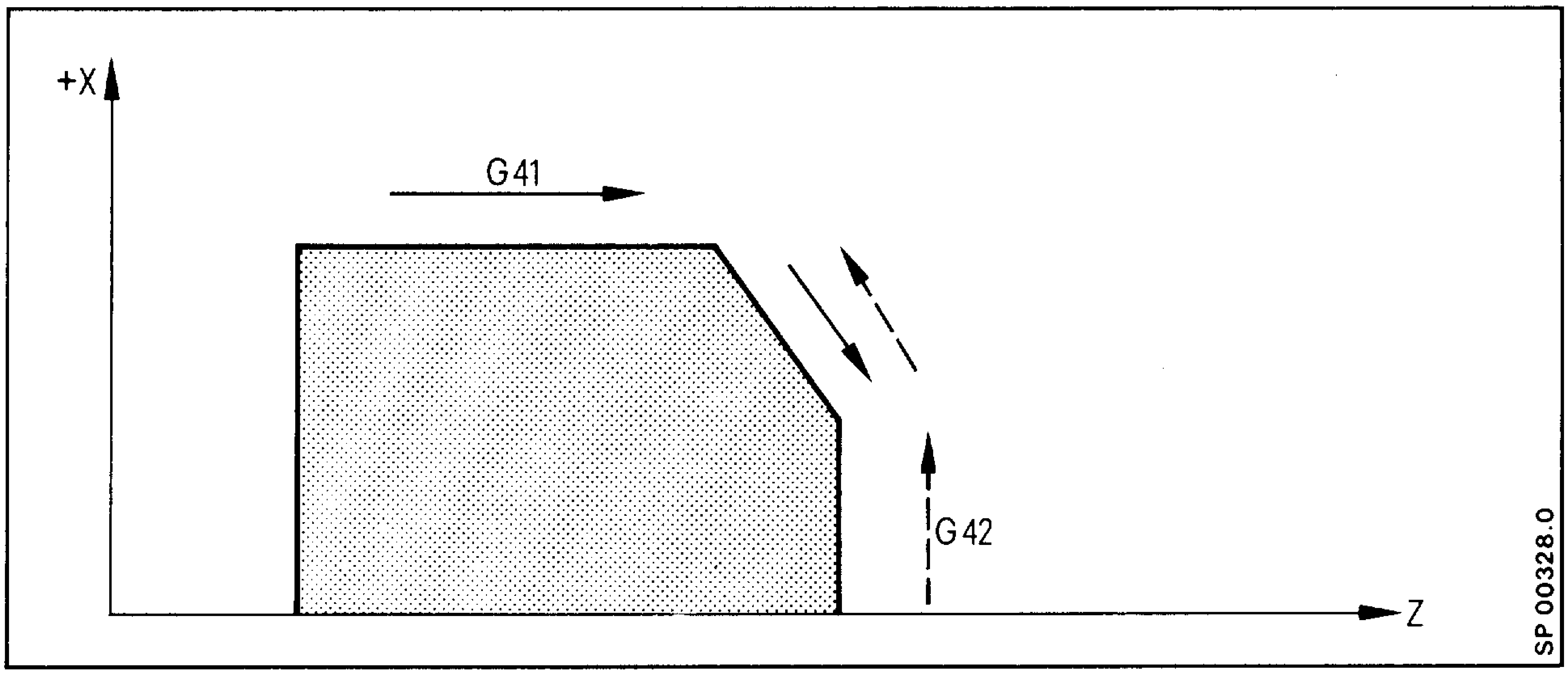
The limitation is not cancelled by G94 or G95 and remains effective only with G96. When constant cutting speed G96 S... is activated, the spindle speed is limited by the last value programmed using G92...

Cancellation of this limitation is also be achieved with G92 S..., whereby the maximum speed for the selected gear stage must be written under S. G92 S 0 will stop the spindle.

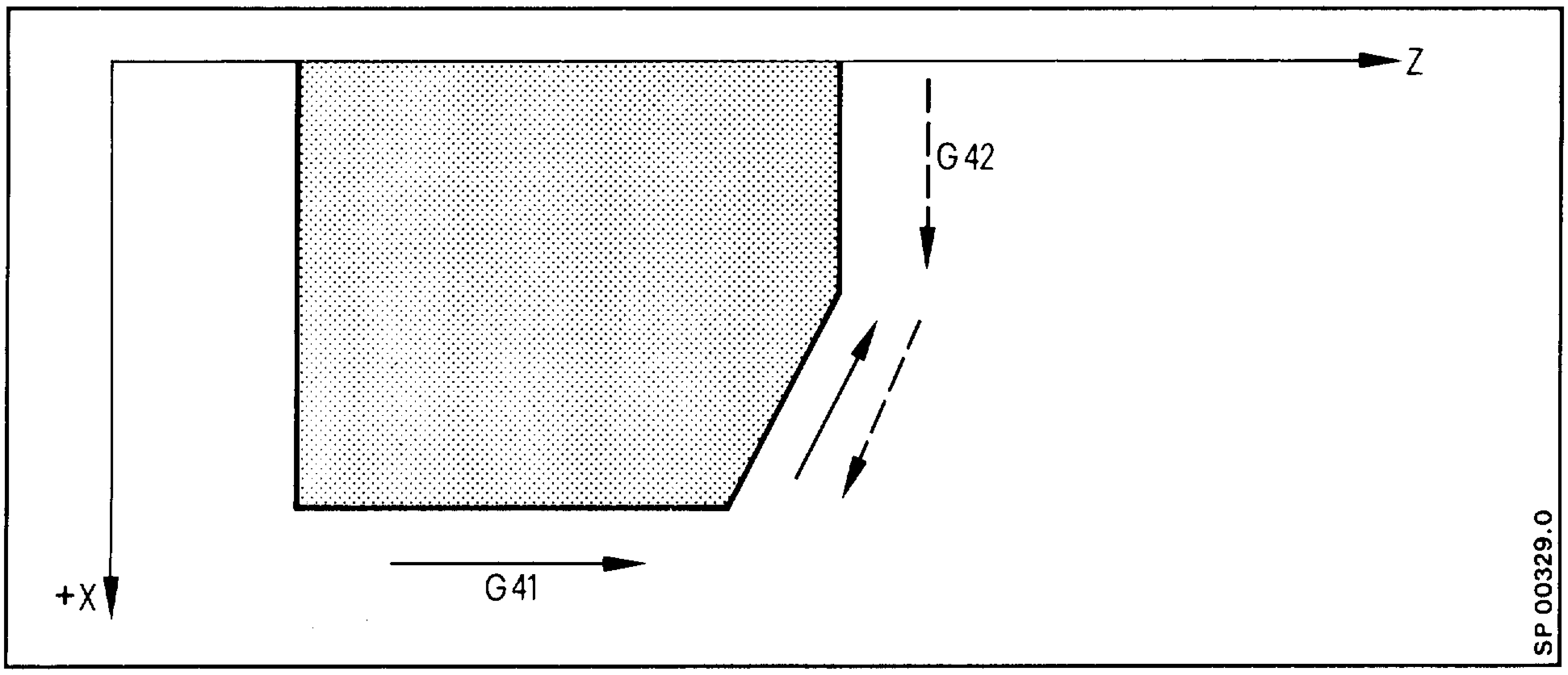
3.14 G40/G41/G42 Tool nose radius compensation (CRC)

	Machining behind the turning axis	Machining in front of the turning axis
G40	Cancel CRC	Cancel CRC
G41	Tool left of workpiece	Tool right of workpiece
G42	Tool right of workpiece	Tool left of workpiece

Machining behind the turning axis



Machining in front of the turning axis



The workpiece contour is programmed using CRC. The controller calculates the path for the tool nose radius centre (equidistant calculation) and the intersection points of these equidistant paths at acute transitions. CRC can be used for inside and outside contours, obtuse and acute angles and at acute and tangential transitions. There are no contouring errors.

Functions G40/41/42 can be entered with or without path data. After programming G41 or G42 the tool nose radius compensation becomes effective in the machining plane with the first movement (X and/or Z). Any tool length compensation is active in both axes for blocks containing G41, G42.
Activation is only effective in blocks with G00 or G01. Within the program (from block to block) a change from G41 to G42 or vice versa is possible. In this case the CRC does not have to be deactivated using G40.

Example: activate, change, deactivate

```
N180  G01  G41  X...  Z...   - activate
N190       G42  X...         - change
N200            X...  Z...   -
N210       G40  X...  Z...   - deactivate
```

Example: activation and deactivation of the tool nose radius compensation

```
% 5551 LF
N10                         T1212  M..  LF -  Select tool
N20  G00      X100.  Z212.  S190   M04  LF -  and offset
N30  G01  G41 X90.          F5.         LF -  Activation of CRC; the
                                              corrected path is reached
                                              at the end of this block
N80  G00  G40 X200.  Z350               LF -  Deactivation of CRC
N90                         T2222  M..  LF -  Tool change
N100 G00  X90.       Z21.3  S220   M04  LF
N110 G42  G01 X80.          F.5         LF -  Activation of CRC;
N120                 Z150.              LF    the corrected path is
                                              reached at the of this
                                              block.
N210 G00      X...   Z...   S00    T00  LF  - Cancellation of
N220 M30                                LF    tool offset using T..00
                                              or T00.
```

When CRC is selected, G92, G59, G33 must not be programmed.
Remedy: Program the functions (G92, G59, G33) before selecting CRC.
Alternatively cancel CRC, select G92, G59, G33, and then reselect CRC.
In the case of selection of CRC, including the G40 block, the active ZO value must not be changed.

3.15 G10/G11 Programming in polar coordinates

G10 Linear interpolation rapid traverse

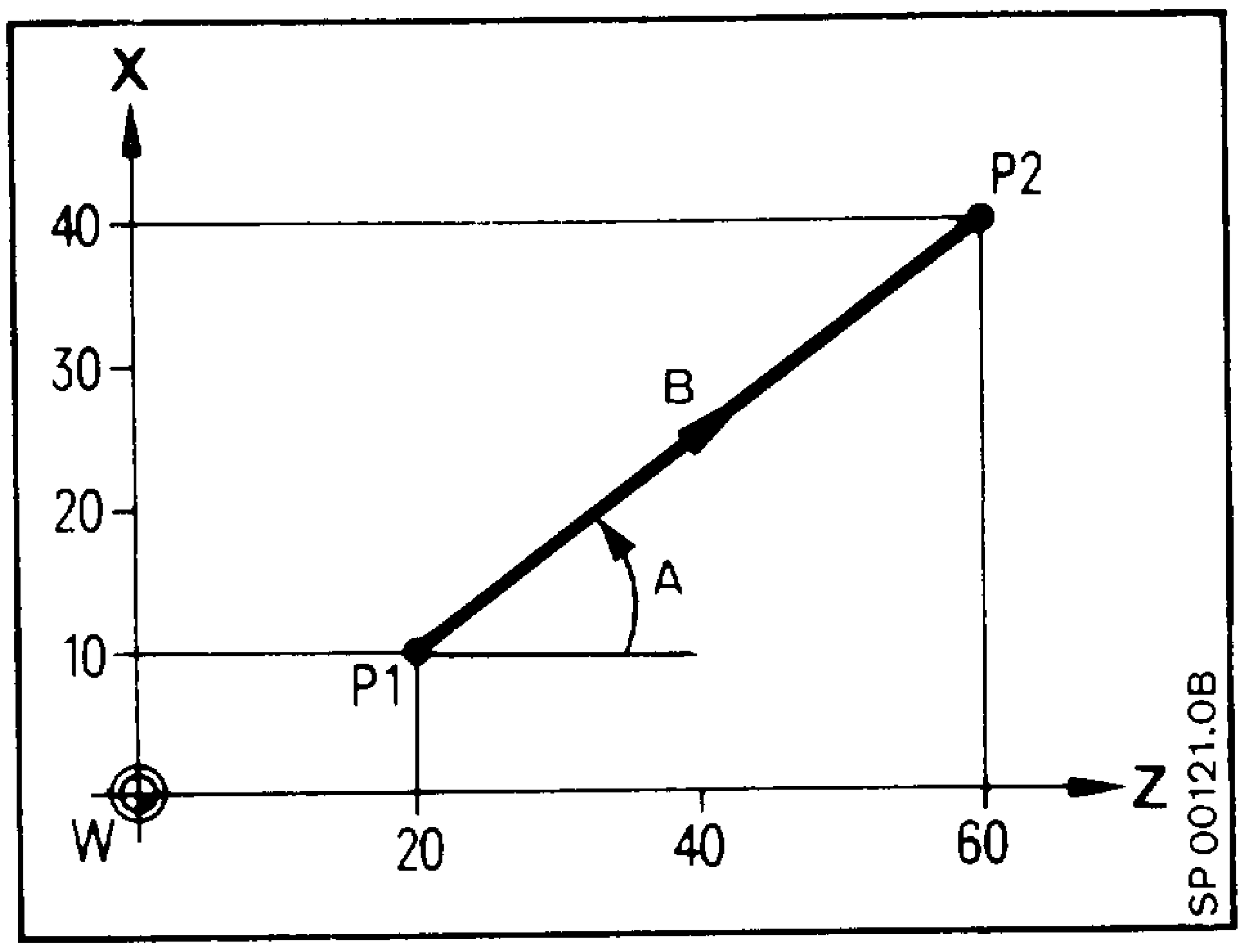
G11 Linear interpolation feedrate (F)

W Workpiece zero

P1 Polar coordinate system centre point

A Angle

B Radius



Example: traverse from P1 to P2

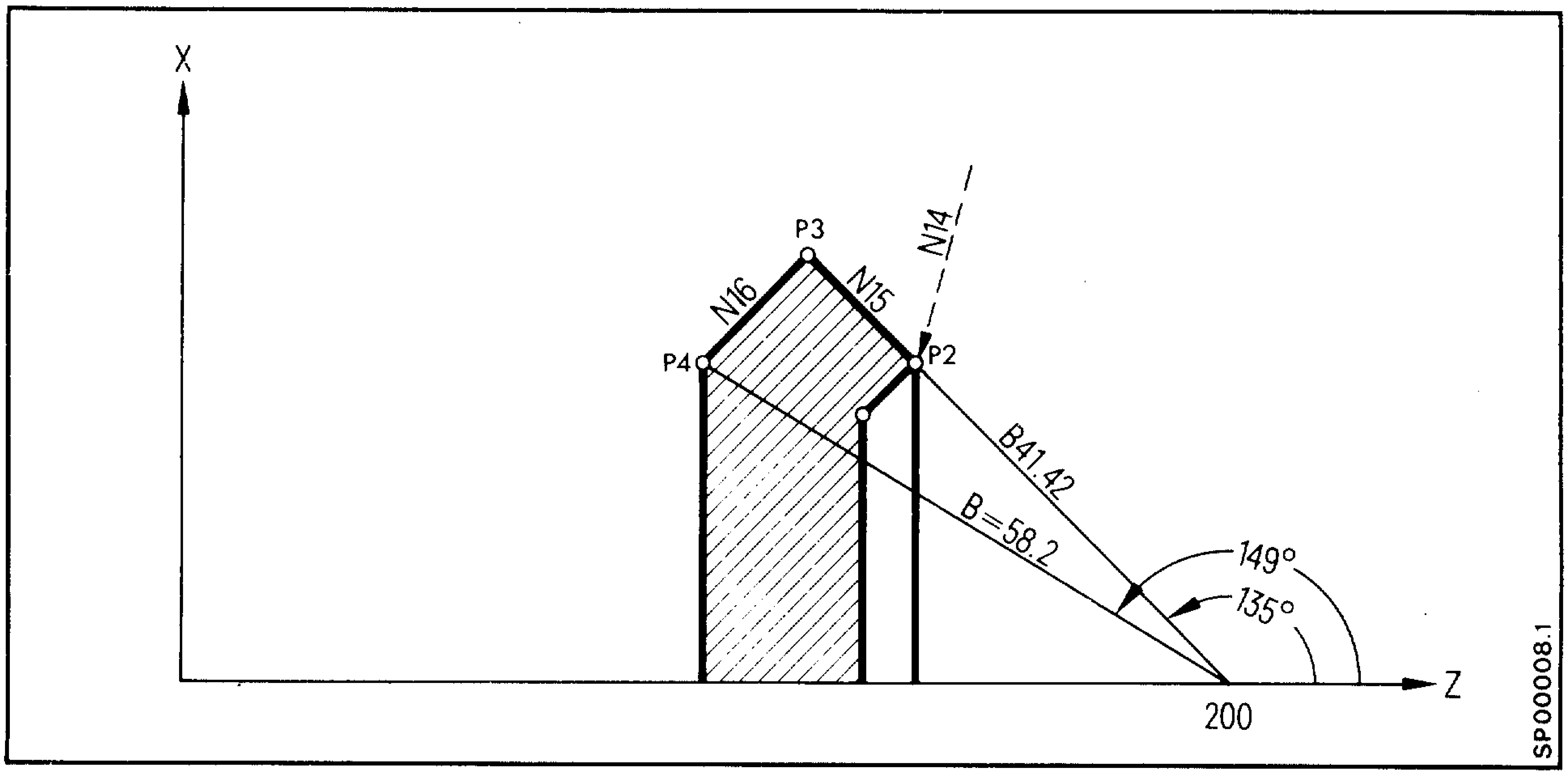
N... G.. Z20. X10. B50. A40.

G..	With rapid traverse G10 or feedrate G11
Z20.X10.	Polar coordinate system centre point
B50.	Radius
A40.	Angle

- The angle is always referred to the first axis programmed in the block in the positive direction (in this example, from +Z to +X in the shortest direction).
  The positive direction of the first programmed axis corresponds to an angle of 0°.
  The positive direction of the second programmed axis corresponds to an angle of 90°.
- The angular value is always absolute and positive. Input resolution 10^{-5} degrees.

- When programming in polar coordinates for the first time, both centre point coordinates must be entered as absolute data.
  It is recommended to always program both centre point coordinates.
- The centre point is modal until reprogrammed. At the end of the program (M02/M30) the centre point is reset.
- The incremental data input of the centre point (using G91) always relates to the last programmed centre point.

Example:



N14	G90	G10	Z200.	X0	B41.42	A+135	LF (P2)
N15		G11			B56.56		LF (P3)
N16					B58.2	A+149	LF (P4)

B... A... Position in the polar coordinate system
(A Angle, B Radius)

Z... X... Centre point of the polar coordinate system

3.16 Dynamic smoothing exponent for thread

The acceleration time of the feed drives for threads is overridden here, because machine data 120 - 123 are not effective when G33 is selected. The number of actual spindle value pulses for the speed of the feed axis is smoothed.

The smoothing and acceleration time is programmed as follows as a separate block:

```
N ..  G92  T.  LF
```

The value is written in machine parameter 358. The exponent may have the value 0 to 5. The value can also be programmed via an R parameter.

Programmed value

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

The actual value scan - cycle time should be used as the time-base.

3.17 Turning/milling operation

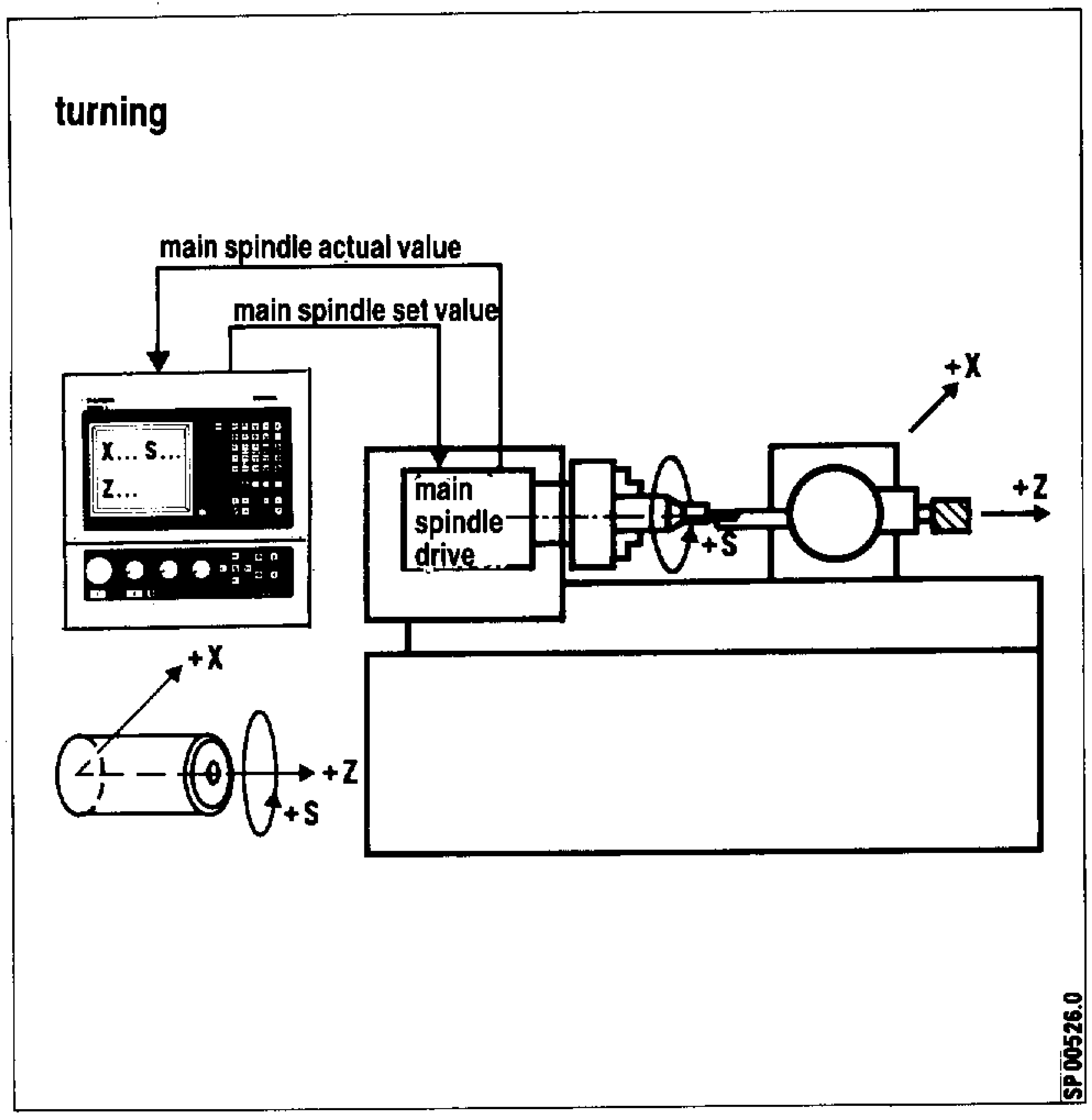
Advances in automation in the area of cutting machine tools have led to a far-reaching optimization of actual machining times. However, in mechanical production, many parts are clamped to a milling machine subsequently to a turning operation and, for example, given Vee-form surfaces or eccentrically related contours (outer polygon, eccentric, flange or cylindrical groove). In the course of this, considerable non-productive time is required for loading/unloading, clamping and, where necessary, alignment of the workpieces.
This control concept avoids this non-productive time and enables a considerably more cost-effective overall machining on one machine with one clamping operation by combining turning and milling operations:

The turning machine in milling operation can also make use of such milling functions as

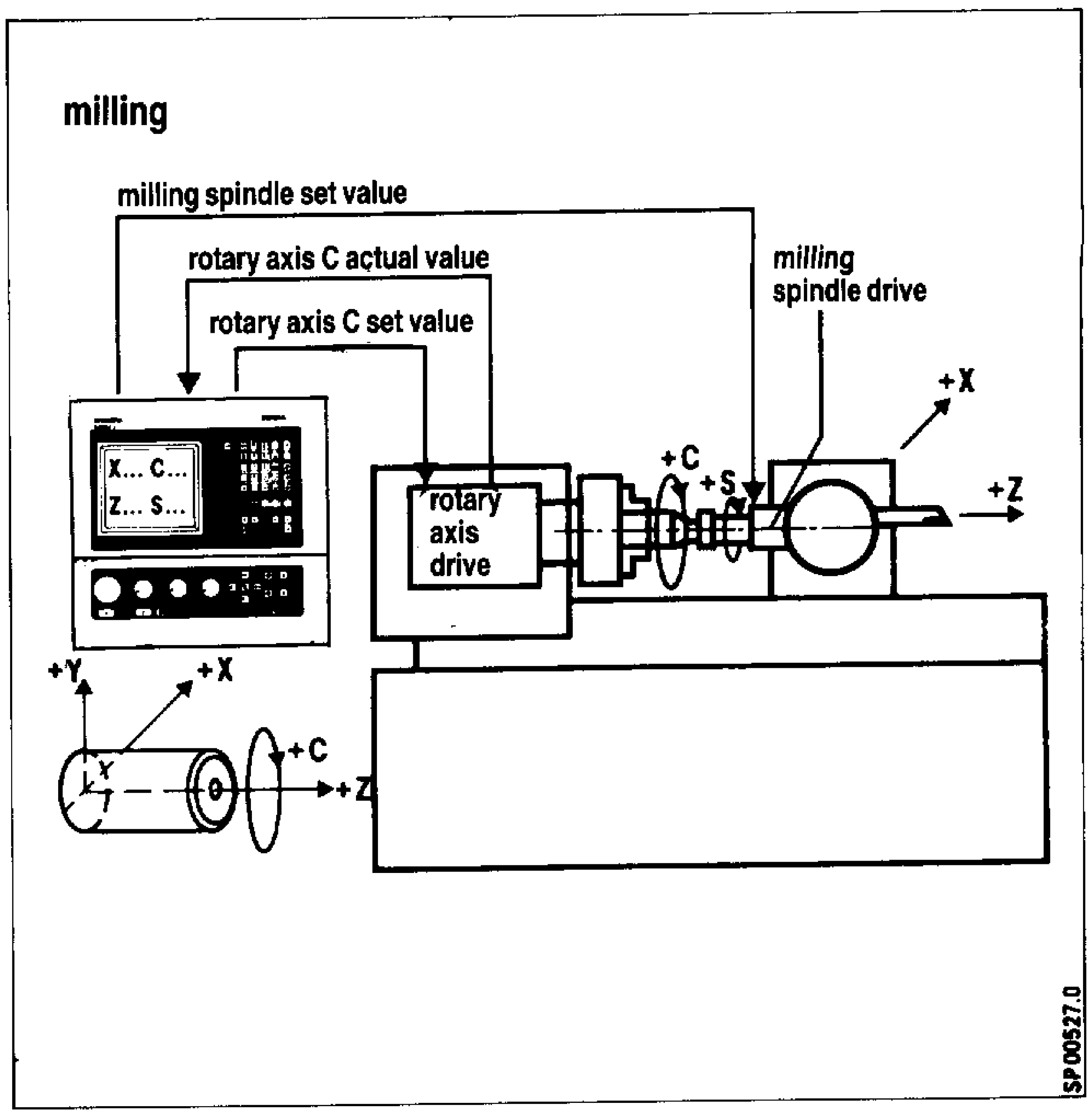
- 3 D interpolation (Option B61)
- Cylindrical interpolation (Option B73)
- TRANSMIT coordinate transformation (G37) (Option B65)
- Cutter radius compensation
- Machining cycles (drilling and milling patterns) (Option B70) 1)

1) If drilling and milling patterns are to be read into the parts program memory in addition to the 3T cycles, the subroutine numbers of the boring and milling patterns must be changed. This avoids assignments of M and T cycles to the same subroutine numbers.The drilling and milling patterns must be expanded for plane selection for C axis. Machining is only possible outside the turning center.

3.17.1 The lathe in turning and milling operation



Turning



Milling

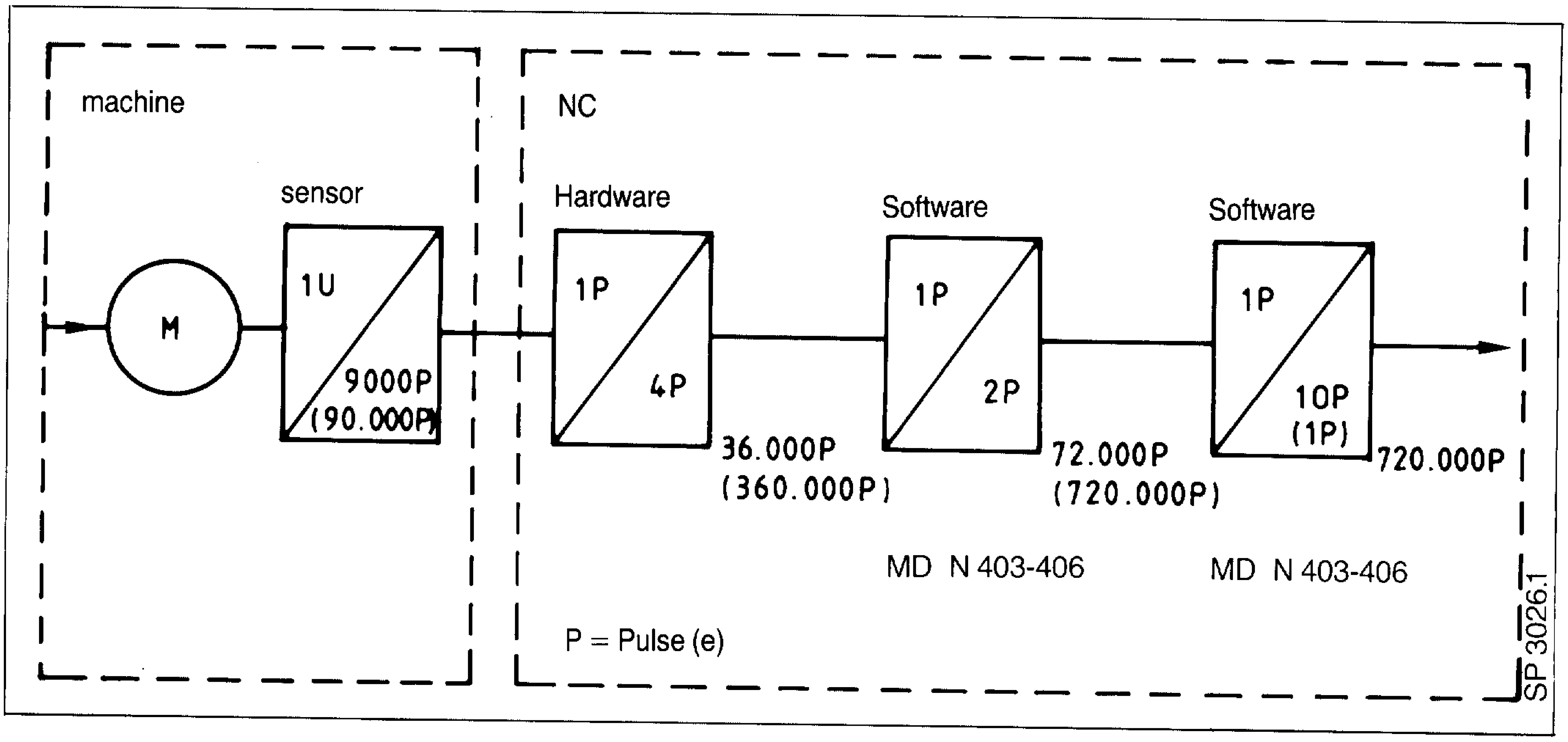
A standard lathe is characterized by the two linear axes X and Z and by a main spindle S . The missing Y-axis in milling operation is so-to-speak replaced by the main spindle as interpolating axis of rotation C.

The set speed value under address S is in this case preset for the cutter spindle drive.

The machine can be switched over by manual operation or programming from turning to milling operation - and vice versa. Programming is in the "Turning" mode according to the Programming Instructions for turning and has the programming features of a lathe control system, such as diameter programming, tool nose radius compensation and turning cycles. On switchover to milling operation, the main spindle functions as rotary axis C with interpolation. The set value under address S now belongs to the cutter spindle. The control system is now programmed in the milling mode according to the Programming Instructions with the following features: 3D linear interpolation, circular and helical interpolation, and drilling and milling patterns, cylindrical interpolation and cutter radius compensation.

3.17.2 Arrangement and structure

Drive and encoder, pulse value C-axis



Pick-up:

The double incremental shaft encoder 6FC9320-1EA is recommended for 0.01 ° resolution.

For resolution 0.001° we recommend an encoder with 18,000 pulses followed by a five-fold EXE. The maximum speed of the C axis is 150 rev/min at resolution 0.01°, 80 rev/min at 0.001° (54600 °/min).

Drive:

Main drive for turning/milling with feed controller properties. For more sophisticated demands, an additional feed drive is available, which can be engaged mechanically.

Setting the direction of rotation of the C axis

The spindle and workpiece turning direction shown defines the relative movement +C'. For workpiece-related programming direction, +C applies.

The following allocation of direction thus applies for the spindle:
+ C' corresponds to M04 or M19 S ...

Software switchover

The NC is commissioned as a 3M and is therefore a 3M in the basic state. Graphics for milling operation (3T + C axis) are not possible. Tool offset can be entered or deleted in the NC system selected (T or T + axis (M))

The switchover from turning to milling operation is effected via the NC-PLC interface with acknowledgment control The "Switchover to 3T" input signal is generated by the PLC. The switchover is only received by the NC in the "RESET" state (Warning 511). The NC acknowledges after switchover with the "Switched over to 3T" output signal.

Caution:
After switching from 3T + C axis (3M) to 3T, the "Reference point reached" signal in the third and fourth axis is removed (optionally via MD).

Machine control panel

It is recommended to enter the machine control panel signals into the appropriate flag area (Interface Description, Part 2, Section 5, Possibility 2).

When switching over from turning to milling operation, all direction keys in the machine control panel for the turning machine must be entered in the flag areas coded as a machine control panel for milling machines.

It is recommended to use the machine control panel of the 3M.

3.17.3 G36, G37 "TRANSMIT" coordinate transformation

From the standpoint of the lathe concept, with its rotary C axis (main spindle) and one linear X axis, it should be treated in terms of programming like a system of polar coordinates, This consideration led, particularly for machining the outside contour of turned parts, to upgrading the lathe for milling operation on the numerical control side so that it can be handled in terms of programming like a milling machine when used for milling operation, i.e. using the system of cartesian coordinates.
Cartesian programming and traversing in polar coordinates means transforming the coordinates.
Coordinate transformation then makes it possible, for example, to program an outside polygon on the lathe in four NC blocks as in a standard milling machine.

Rotary motions are programmed using the appropriate tool offsets T1 to T16 or T32 following 3T/3TT Programming Instructions.

Milling operation is programmed using tool offsets D33 to D64 following 3 M Programming Instructions.

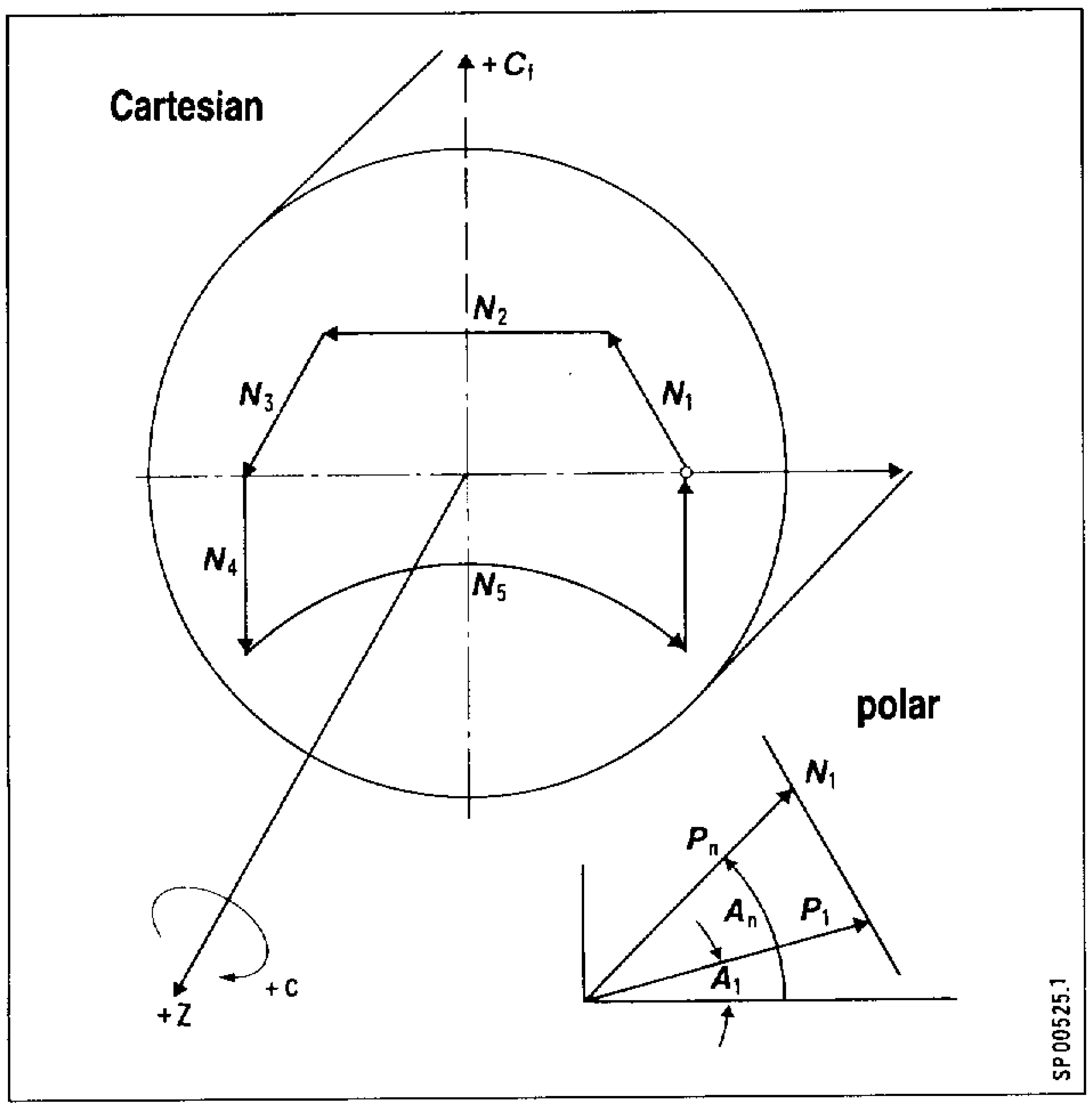
The "TRANSMIT" coordinate transformation is for milling turned part outer contours on faces (continuous path control, linear axis with rotary axis). The TRANSMIT function (Transformation Milling into Turning) permits contours to be programmed in a "fictitious" cartesian coordinate system while the machine movement is in the real coordinate system.
The fictitious cartesian coordinate system is implemented by the first axis X and by the fourth axis C as rotary axis. Subsequently, the rotary axis is designated with address "C" with the symbolic address "Cf".
The transformation is selected and deselected in the program via G functions. G37 is effective in the G17 plane.

G36 Cancel TRANSMIT coordinate transformation (Reset state)

Programming as normal in real cartesian system (machine coordinate, rotary axis C in degrees, speed in degrees/min)

G37 Selection of TRANSMIT coordinate transformation

Programming is in the fictional system of cartesian coordinates. The 4th axis must be defined as main axis parallel to the Y axis.



Plane definition

	Axis 1	Axis 2	Axis 3	Axis 4
	X	Z	Y	C
G17	X	-	-	C
G18	X	Z	-	-
G19	-	Z	-	C
Interpolation parameter	I	K	-	J

- The missing axis of the current fictional plane is supplemented with G17/18/19 if only one axis is programmed in the block and this axis is in the selected plane (example a and b).

- If only one axis in the fictional plane and one axis not belonging to the fictional plane are programmed, the control system generates Alarm 504 (example c).

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

Example: G37 active, X - CF = fictional plane = G17 - plane

a) G17 X..	LF	= Cf is supplemented
b) G17 Cf..	LF	= X is supplemented
c) G17 X.. Z..	LF	= Alarm 504
d) G17 X.. Cf.. Z..	LF	= No supplementing of axis No alarm
e) G17 Z..	LF	= No alarm

- The achievable accuracy at the part when the C axis is used depends on the current working radius (control in degrees) and the sensor resolution.

- Block search using G37 blocks is not permitted.

- The programmed G36/G37 system remains selected in automatic interrupt status. It is therefore possible to traverse manually using the cartesian system.

- The cutter centre must lie in the turning centre on the X axis.

- An offset in Cf direction cannot be balanced by the control system.

- The workpiece zero reference point lies in the centre of the plane axis.

- When G41/G42 cutter compensation has been selected, transformation may not be engaged or disengaged (change from G36/G37).

- Transformation may not be engaged or disengaged within a contour definition block sequence (change from G36/G37).

- Rapid traverse motions should be programmed under G01 or G11 with the appropriate F value.

- On changing from G36 to G37, the C axis actual value is set to 0 and the X axis actual value to the actual value of the machine, irrespective of the offsets present (zero offset). Zero offsets are included in cartesian calculations.

- In the inch input system the external zero offsets for the C axis must be 0 as the position of the decimal point is different.

- After selecting or deselecting G37, the first block must be programmed in G90. If programming is to be made in G91, the last position reached can be repeated in the G90 block. Alternatively, the zero offsets can be deselected before the change.

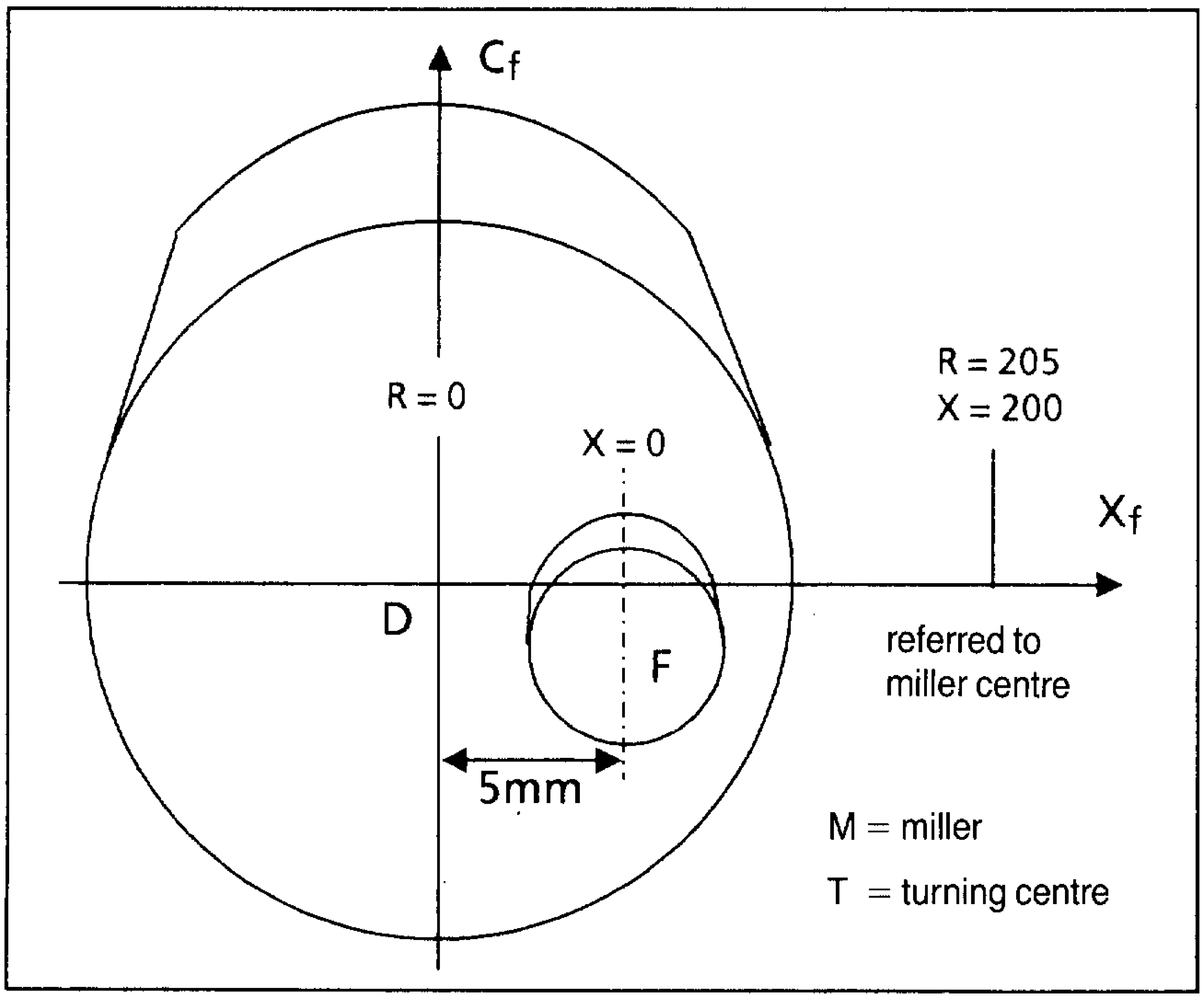
- Tool path feed rate is programmed and retained constant in the system of X-Cf coordinates. This feed rate is monitored in the case of G37 so that the max. permissible C axis turning speed is not exceeded.

- If necessary, the NC reduces the programmed speed. If the actual speed is 1 mm/min on the contour, an error message is output (FE 316).

Tool length compensation for "eccentric" cutter in TRANSMIT

An abitrary compensation number can be selected from the tool compensation memory by programming G37 D... .
The geometrical data of the given D number result in a compensation to the current position value of the X axis so that this X axis has a reference as radius axis with respect to the cutter center.
Programming of the G37 D... combination is binding.

Example:



Tool length compensation for eccentric cutter:
e.g. D48 D5. P0.
Offset = 5 mm

Programming:

.
.
.
N10
N15 G37 D48 *
N20 .... D35 (CRC)*
.
.
.

The compensation value is 5 mm.

Function G37 D... is comparable to setting an actual value.
The new actual value is equal to the old actual value for the total of the tool offset.

Compensation is not active when programming G37 D00. The G37 D.. function must not be programmed as first block in the program.

Program structure

The parts programs for turning operations are clearly separated from the programs for milling operation. By linking the programs for turning and milling operation, an overall program of operation is created which can be executed by means of a single start. The programmer sets the next program to be called in the parts program. A freely selectable M function at the beginning of a parts program defines whether the program is for turning or milling. M function M02 marks the end of a parts program and M30 defines the end of the overall machining program. This structure of the overall machining program enables re-entry into each parts program after tool breakage.

Example for program organization

Program for complete
machining

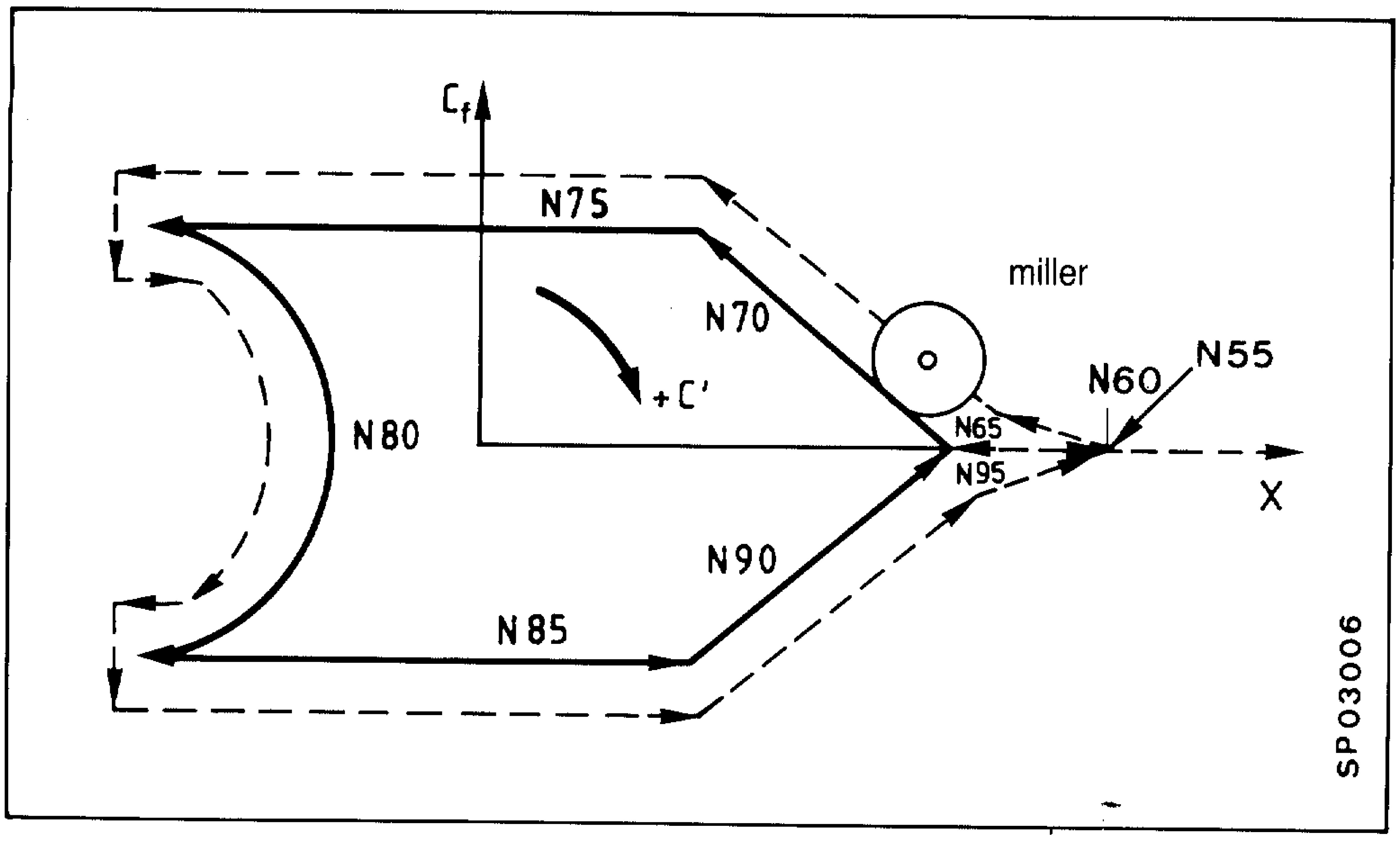
Turning	
% 10	Number of complete machining program turning and milling
M71	Start turning
N.. X.. Z..	
*	
*	
N.. X.. Z..	
% 20	Number of next part program
M70	End of part program
M02	
Milling	
% 20	Number of part program
M70	Start milling
N.. X.. Z.. C..	
*	
*	
N.. X.. Z.. C..	
% 30	Number of next part program
M71	End of part program
M02	
Turning	
% 30	Number of part program
M71	Start turning
N.. X.. Z..	
*	
*	
N.. X.. Z..	
% 10	Number of complete machining program for next pass
M30	End of complete machining program

Program structure with NC/PLC sequence description

NC programming	PLC functions	Remarks
% 100 (machining program) Turning operation N1 M71 N5 G.. G.. F.. X.. Z.. .		NC start by operator
		. Program section . . "Turning" N245 % 110 * Program No. of next program N250 M70
N255 M30	* M70 "RESET" * "AUT 3T" SWITCH-OVER = 0 signal	* Switchover with M70 from 3T 3T+ C axis (3M)
	* C axis not in follow-up operation * Ref. point C axis * S set value on drive appliance (C axis) * NC start	* % 110 program is started
% 110 (Milling operation) N1 M70 N260 G.. G.. F.. Z.. C.. .		. Program section . . "Milling"
N525 % 120	* M71 "RESET" * C axis in follow-up operation	* Program No. of next program
N530 M71 N535 M30	* "AUT 3T SWITCH-OVER" = 1 signal * S set value on main spindle * NC start	* Switchover with M71 from 3T+C axis (3M) 3T * % 120 program is started
% 120 (turning operation) N1 M71 N540 G.. G.. F.. X.. Z.. . .		. Program section . "Turning" .
N870 % 100		* Program No. of start program
N875 M30	* PLC sets the NC to the 3T basic state with M30 without M70/71	* End machining program (M30 without M70/71)

Other, freely selectable M functions can be used for M70/71. M02/M30 have the same effect in the NC. The end of this program must be evaluated in the PLC.

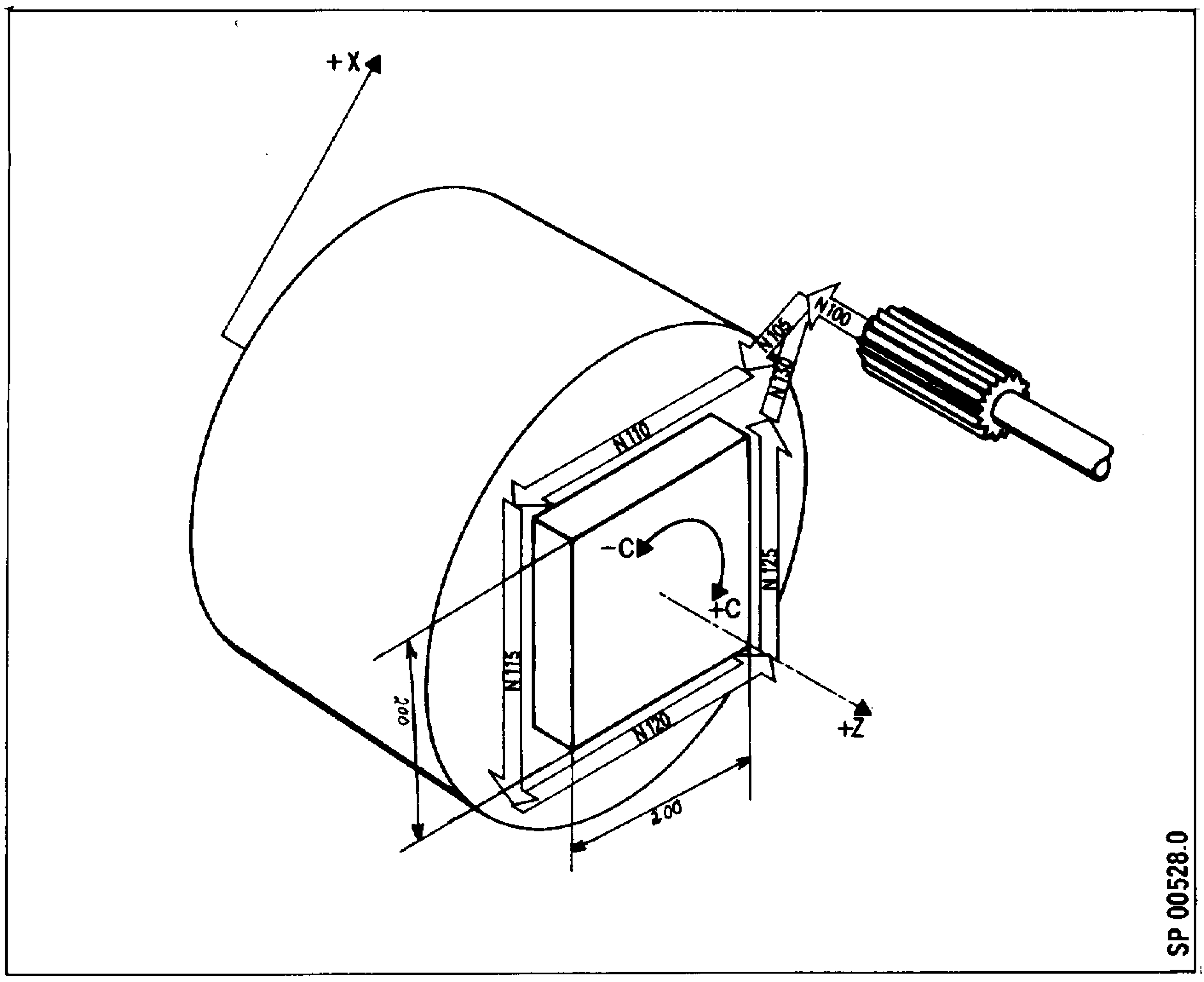
Example 1: "Milling of a contour on the plane surface" program




```
% 20                          Milling parts program
M70                           Engage milling operation
(a) 31
N5 X.. C.. Z..                System of real coordinates
   *
   *
N50 X.. C.. Z..
N55 G0 x 120 C0 Z100 D50      Approach starting point
                              Select length compensation
N60 G37 G01 F200 Z90          Engage TRANSMIT with G37,
                              Programming is in the system
                              of fictional coordinates
N65 G42 X90 C0                Selection of cutter radius compensation
                              (C corresponds to fictional axis Cf)

N70 X40 C40
N75 X-60
N80 G02 C-40 J-40             Semicircle
N85 G01 X40
N90 X90 C0
N95 G40 X120                  Cancelling of cutter radius compensation
N100 Z100                     Tool withdrawal in Z
N105 G36                      Disengage transformation
N110 X.. C.. Z..              System of real coordinates
     *
     *
```

Example 2: Milling a contour on a plane surface.




```
% 20                              Milling program section
     *
     *

N100 Z200 G1 F5000 C0
N105 G37 G42 D49 X100 C0          Engage coordinate transformation
N110 X-100 F100                   with G37
N115 C-100
N120 X+100
N125 C100
N130 G42 D0 X110 C110 F5000
N135 G36                          Disengage coordinate transformation
     *
     *
```

3.17.4 Display

Display of actual values can either be in the real machine system (SE 2, Bit 3 = 1) or in the fictional cartesian system.

Display real machine system

It is not possible to display the workpiece-related actual

Axis	Actual values			Distance to go		
	Input system inches	Input system metric		Input system inches	Input system metric	
X	10^{-4} Inches	10^{-3} mm	real	10^{-4} Inches	10^{-3} mm	fiktiv
Z	10^{-4} Inches	10^{-3} mm	real	10^{-4} Inches	10^{-3} mm	real
Y	10^{-3} Degrees		real	10^{-3} Degrees		real
C	10^{-3} Degrees		real	10^{-4} Inches	10^{-3} mm	fiktiv

Display fictional Kartesian programming system

It is possible to display the workpiece-related actual value system

Axis	Actual values			Distance to go		
	Input system inches	Input system metric		Input system inches	Input system metric	
X	10^{-4} Inches	10^{-3} mm	fiktiv	10^{-4} Inches	10^{-3} mm	fiktiv
Z	10^{-4} Inches	10^{-3} mm	real	10^{-4} Inches	10^{-3} mm	real
Y	10^{-3} Degrees Program not running		real	10^{-3} Degrees Program not running		real
	10^{-4} Inches Programm running	10^{-3} mm Programm running	fiktiv	10^{-4} Inches Programm running	10^{-3} mm Programm running	fiktiv
C	10^{-4} Inches	10^{-3} mm	real	10^{-4} Inches	10^{-3} mm	fiktiv

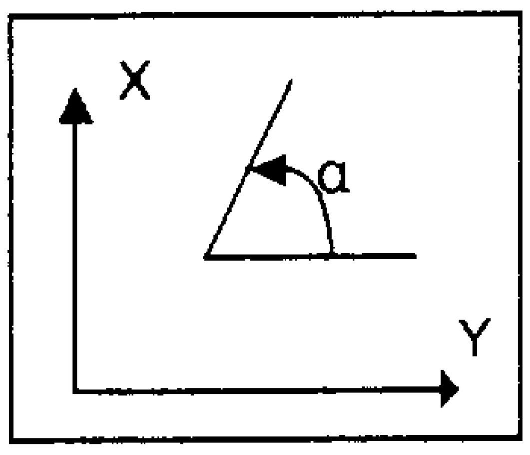
3.18 Programmable Coordinate Rotation [1)]

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

N... G59 X... Z... A...

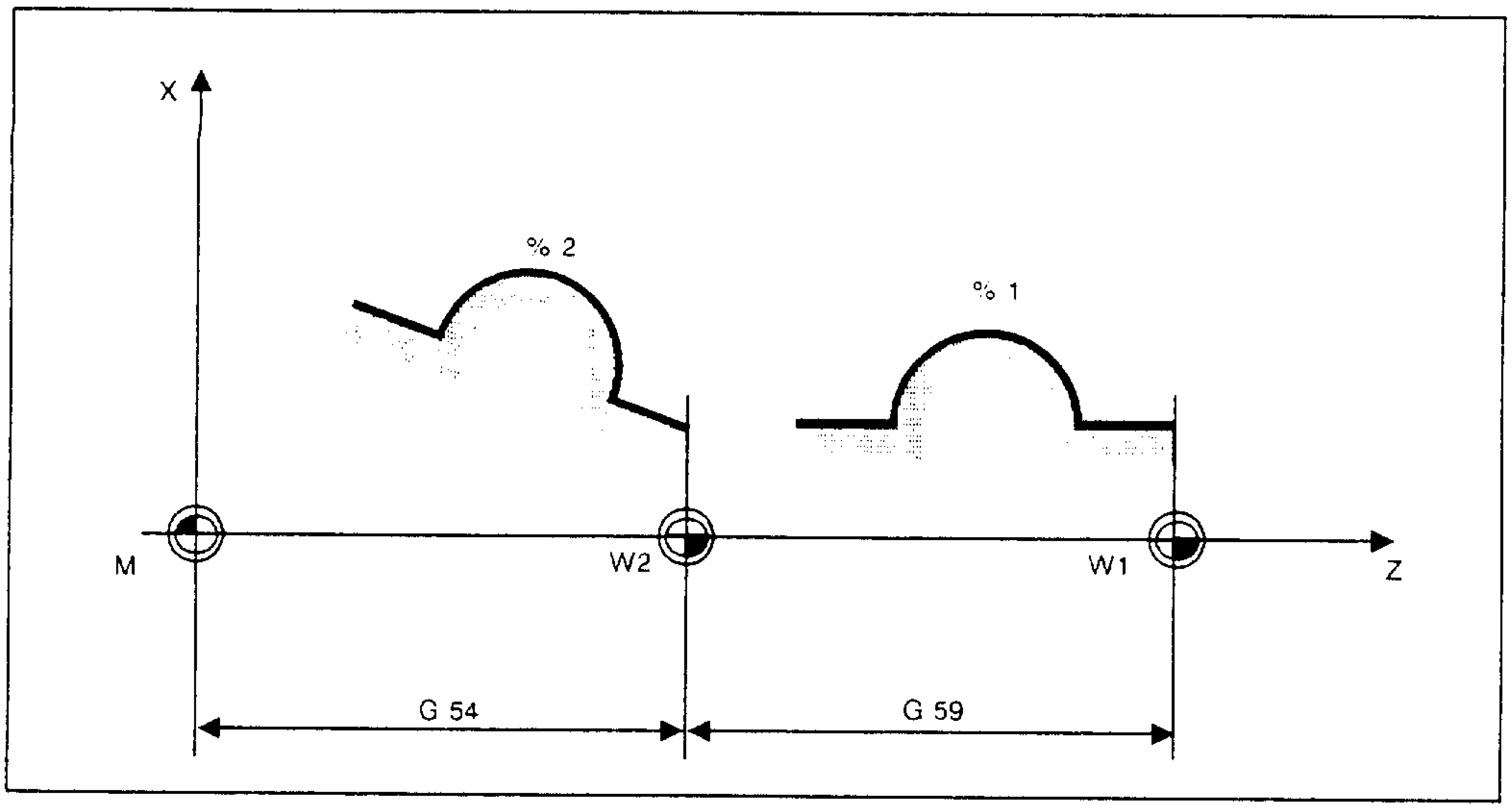
- G59: G function for programmable additive zero offset
- X... Z...: Value of the programmable additive zero offset
- A...: Turning angle for coordinate rotation

- The angle of rotation must not be negative or larger than 360 °.
- The angle of rotation can also be programmed by chaining R parameters.
- The angle of rotation is automatically deleted at the end of the program.
- The angle of rotation is displayed under G59.
- After a G59 block with an angle of rotation, traversing must be made in both axes to be rotated, to give the NC a new reference point.
- When mirroring an axis, the block to be traversed is first rotated and then mirrored.
- In a G53 block, coordinate rotation is not effective.
- If the angle of rotation is written via @29, @31 block must be programmed before the @29 block, so that the angle of rotation is effective from the following block.
- Direction of rotation



1) Option only Basic Version 4B, 4C

Example:

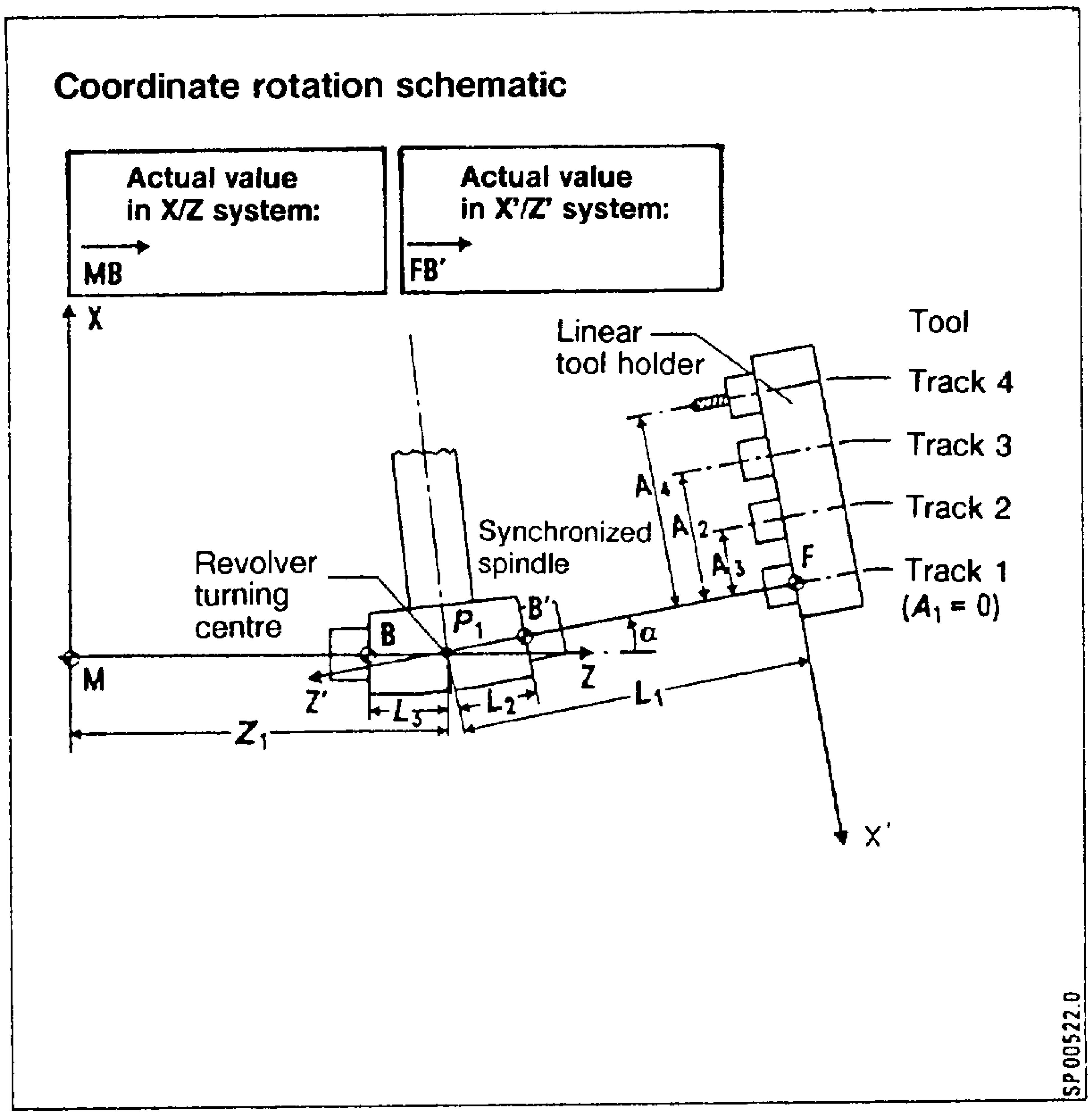


M = Machine zero

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

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

% 1 LF	% 2 LF	
.	.	
.	.	
.	.	
N5 G54 LF	N5 G54 Z55 F100 LF	
N10 G59 X0 Z55 LF	N10 G59 X0 Z0 A340 LF	Selection of co-
N15 G00 X10 Z0 LF	N15 G00 X10 Z0 LF	ordinate rotation
N20 G01 G91 Z-15 F100 LF	N20 G01 G91 Z-15 F100 LF	
N25 G03 Z-35 B10 LF	N25 G03 Z-35 B10 LF	
N30 G01 Z-45 LF	N30 G01 Z-45 LF	
N35 M30 LF	N35 G59 A0 LF	Deselection of co-
	N40 M30 LF	ordinate rotation



1) Option only Basic Version 4B, 4C

3.19.1 Machine data

See schematic

Mach. data No.	Input resolution	Max. value	Comments
426 bit 3	-	-	Activating the coordinate rotation option
325	0.00001° (10^{-5}°)	90 000 00 (90°)	Angle incrementation a Typical value: 10 000 00 (10°)
326	1 µ	9 999 999 (10 m)	Coordinate Z_1 of transformation center P_1 (X_1, Z_1) referred to machine zero M (coordinate X_1 = 0). Intersection of tool track 1 with center.
327	1 µ	9 999 999 (10 m)	Distance L_1 of transformation centre P_1 from tool zero F (tool track 1 of the rear tool holder)
328	1 µ	9 999 999	Revolver radius L_2, distance from revolver turning centre to reference surface B of the synchronous spindle.
317 . . . 324	1 µ	9 999 999	A_1 Distance from the tool . tracks to the Z or Z' axis. . Is effective as an additional zero offset in the . X or X' axis.

- Selection of the effective value via interface signal (3 bit code) (byte 10, bit 3-5).
- Can only be used in G39 mode.
- @31 must be programmed before each selection of a new machine datum (317-324) with a track change.
- Calculate like zero offset in the X axis.
- When programming G53, the offset is deselected.

3.19.2 Actual values, reference points

In a rotated system, the new "machine zero" (X' = 0, Z' = 0) is defined by point F (i.e., point B' of the synchronous spindle and point F of the first tool track coincide).

The actual value of the rotated Z' axis mirrors actual value Z.

3.19.3 Selection and deselection of coordinate rotation (G39/G36)

G function G39 selects coordinate rotation in automatic and MD mode, G36 deselects it.

In the Reset state of the control, it is possible to activate coordinate rotation for JOG and INC modes via an interface signal.

When AUTO is interrupted in JOG/INC, the mode selected by the program (G36 or G39) is always effective, i.e. it cannot be modified by the PLC.

The PLC, however, always receives confirmation of the current mode (G36/G39) via an interface signal.

In G39 mode, the same interface signals are used for the traversing keys as in G36 mode. The direction is based on the selected actual value system.

3.19.4 Switching on the control, reference point approach

After switching on the control in JOG and INC modes, traversing is possible in G39 mode via the PLC signal (G39) (without fixed reference point P_1).

Reference point approach (REF mode) must always be in G36 mode.

3.19.5 General information on coordinate rotation X/Z

- The "Define PATHS via PLC" function may only be executed in G36 mode, otherwise alarm 521.
- Traversing commands in G39 mode are always effective in both axes (X and Z).
- Machine data, axis speeds always refer to the non-rotated system.
- TNRC G41/G42 must be deselected with G40 before each mode change G36/G39 and vice versa.
- Block search is possible for blocks in G36 mode and also in G39 mode, as an unambiguous transformation centre is available.
- The display of actual values can be both in a fictitious coordinate system and in real machine coordinates (SE 2, bit 3 = 1).
- The distance to go is always displayed in fictitious coordinates.

4.0 Switching and miscellaneous functions M, S, T, H

The switching and miscellaneous functions are output in the block in which they are programmed. A maximum of one M, one S, one T and one H function can be programmed in a single block. The output to the interface is in the following sequence:

M - S - T - H [1)]

A machine parameter is used to determine whether or not the functions are output before or during axis movement. Exact specifications are given by the machine tool manufacturer. The following applies to output of the functions during axis movement:

If a new value is to become effective before axis movement, the new function must be written in the previous block.

4.1 S Word

The S word can be used in the following ways:

Spindle speed in coded values
Spindle speed in rev/min or 0.1 rev/min
(preset during commissioning).
Cutting speed in m/min or 0.1 m/min
(preset during commissioning).

The use of different forms of input for spindle speed and cutting speed is not possible.

1) H function with basic version 4 and SINUMERIK 3TT

4.2 T Word tool command

The tool command determines the tool (tool number) required for a particular machining section and also the relevant tool data (tool offset number).

T 12 16

T Tool command address

12 Tool number 00 .. 99
(1 or 2 decades)

16 Tool offset number
(Tool offset selection: 01..32)
(Tool offset deselected: 00)

Tool offset number

The tool data are stored under a tool offset number. The offset values for 32 tools can be stored.

Tool offset:

Each of the 32 tool offsets (T01...T32) comprises:

X... tool length offset X axis

Z... tool length offset Z axis

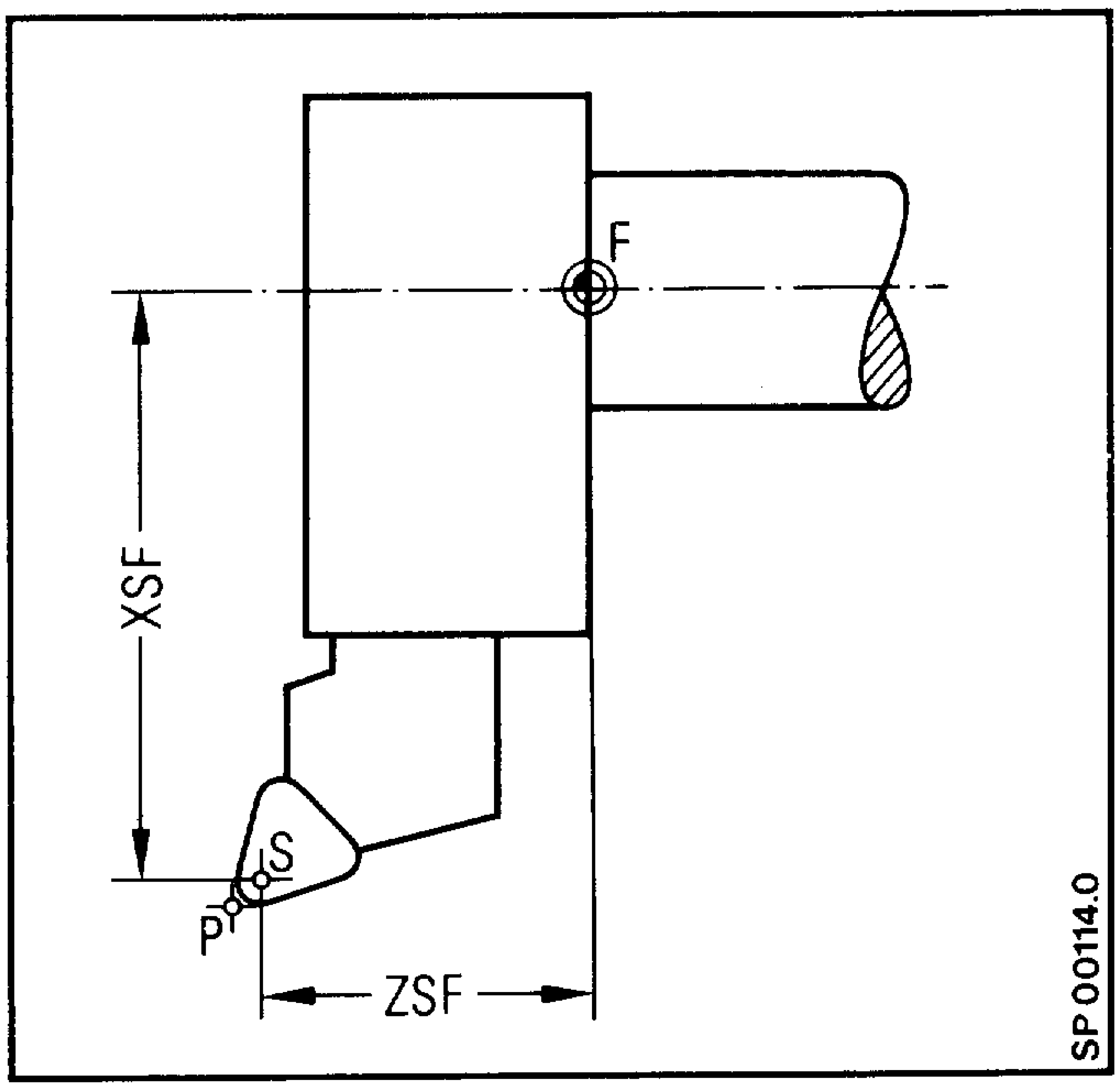
B... tool nose radius

A... position of tool cutter point

1) max. of 32 tool offsets with basic version 4, SINUMERIK 3TT and software version 02

4.2.1 Tool offset without using CRC

The effective tool offset is the sum of the tool length compensation and any external length compensation. This sum corresponds to the dimension XSF or ZSF.

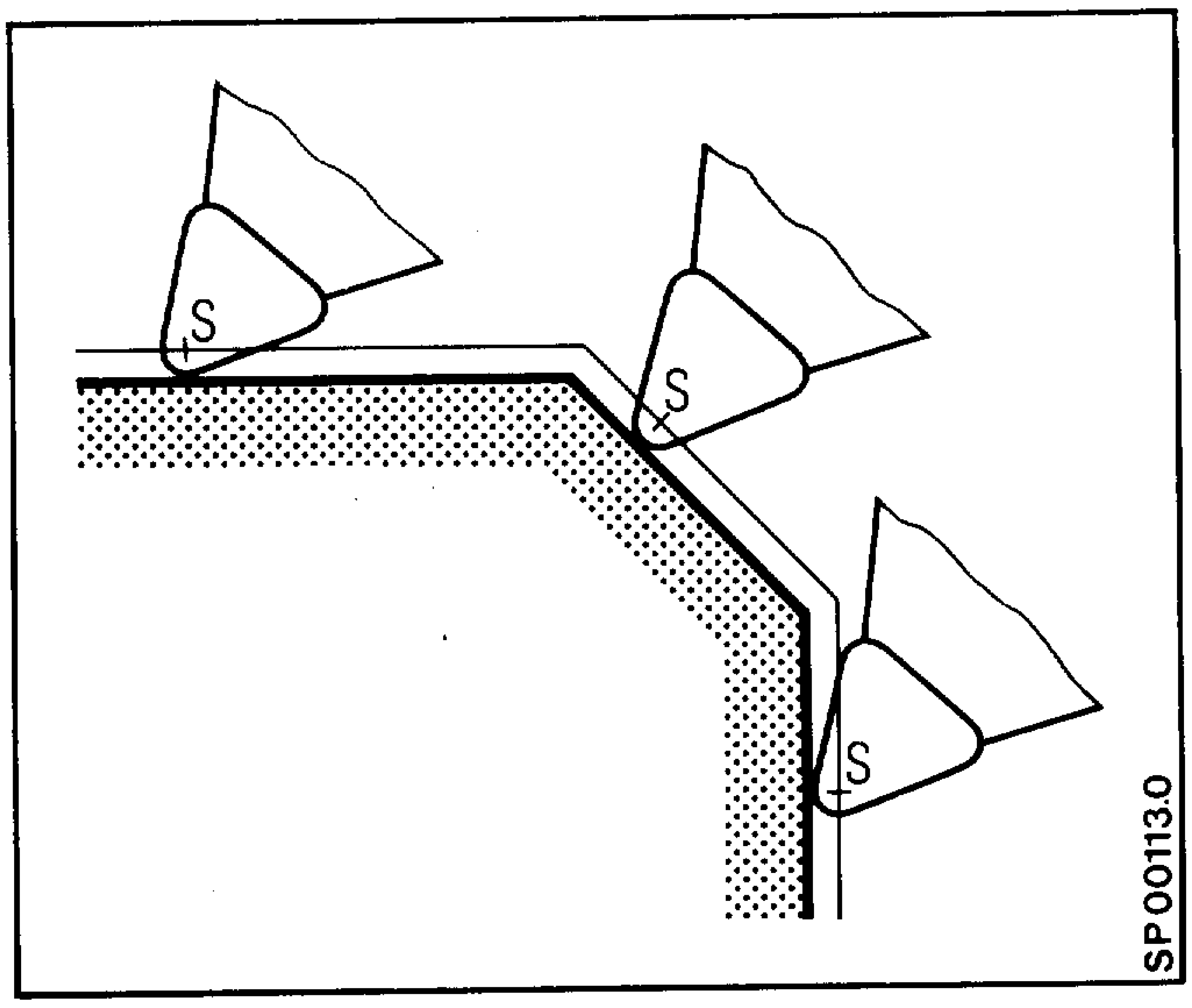


P = theoretical tool nose point

S = tool tip radius centre

F = slide reference point

The path of the tool tip radius centre is programmed. The length compensation refers to the tool nose radius centre.



Tool nose radius centre path

Workpiece contour.

Offset calculation:

When changing the tool offset number the difference between the old and new values is calculated.

The following is determined during commissioning:

The resulting difference

- is traversed directly after the change in offsets; no axis command is necessary for traverse of the tool offset or the difference;

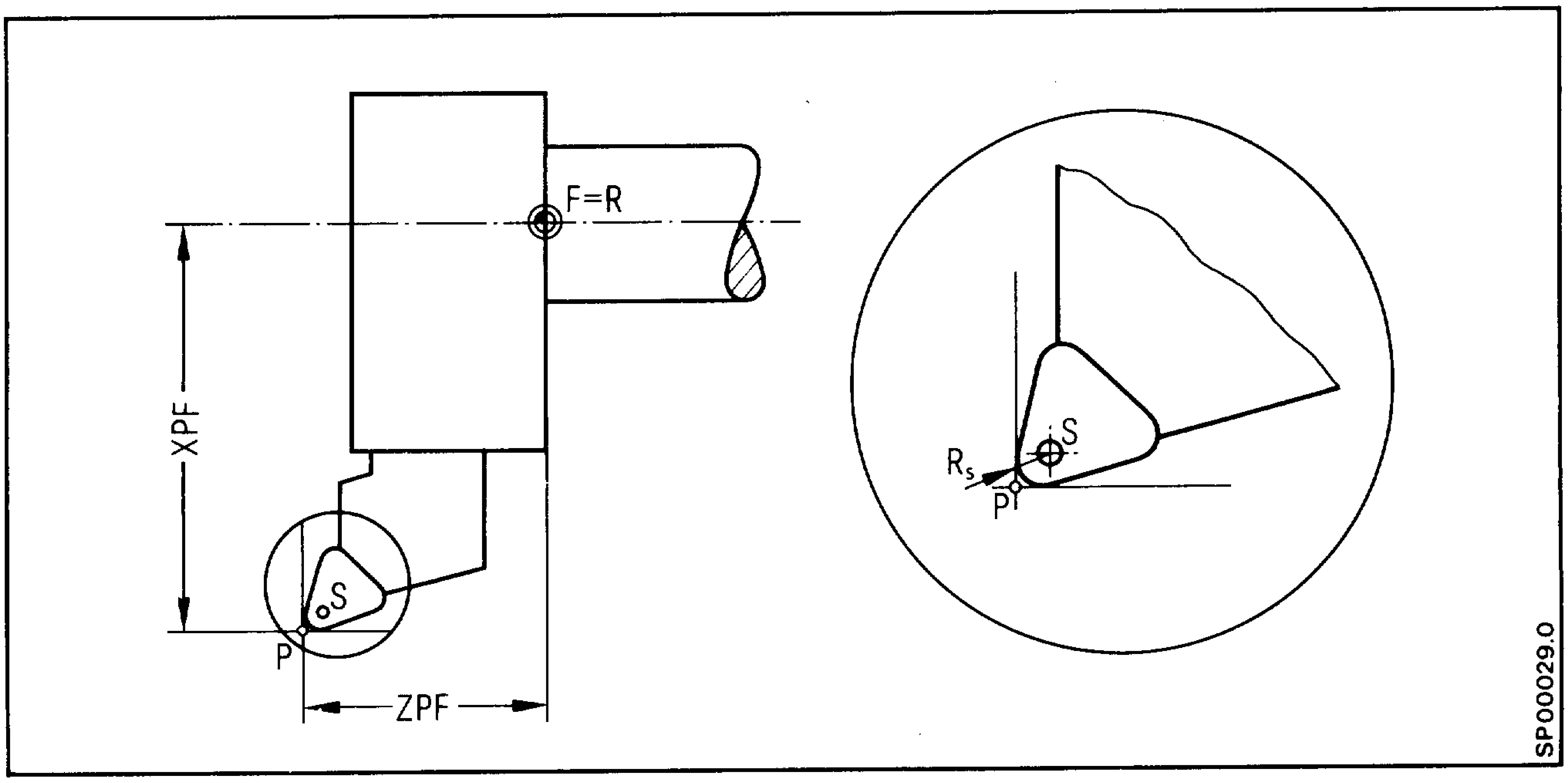
- is first considered when traversing of the appropriate axis occurs.

Note: With CRC (G41, G42) the difference is traversed in both axes in addition to the tool nose radius.

4.2.2 Tool offset using tool nose radius compensation

Using tool nose radius compensation the workpiece contour can be programmed. The tool length compensation to be input refers to the tool nose point "p". Both the tool nose radius and the position of the tool nose point must be entered. The controller then calculates the path to be traversed. No contour errors occur.

The tool nose radius compensation is effective after execution of the block in which it is programmed (G41, G42), i.e., the following block is traversed properly.

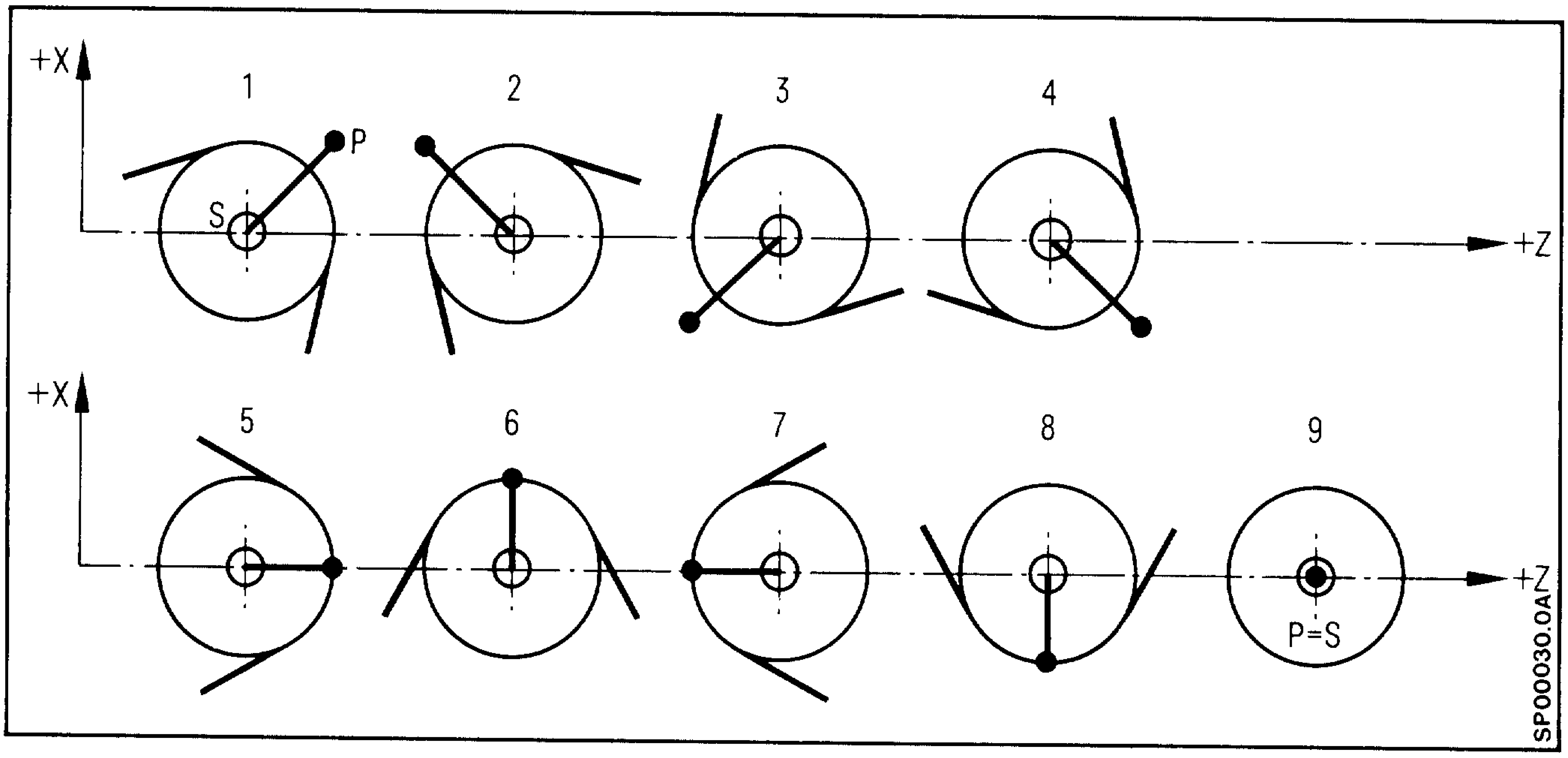


P = theoretical tool tip
S = tool nose radius centre
R_s = tool nose radius
F = slide reference point

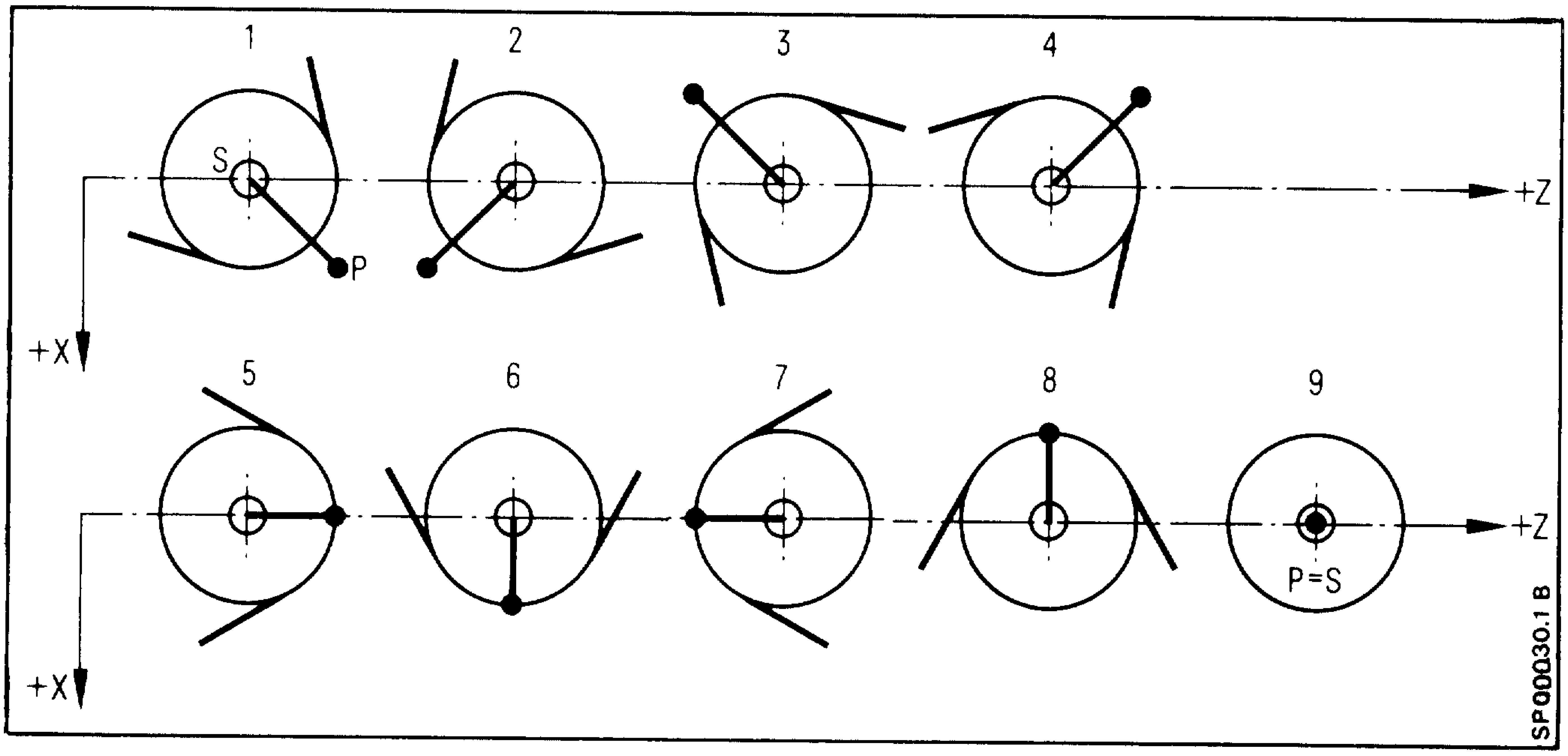
In the case where XSF and ZSF are selected as tool dimensions instead of XPF and ZPF (tool nose cutter point - slide reference point), the designation 9 must be used for each tool (see following page).

In order to calculate the tool radius compensation, the controller requires some indication of the position of the tool cutter point. A total of nine designations are used to represent the theoretical line between the tool nose "P" to the tool nose radius centre "S".

Viewing is always from S to P.



When the working area is in front of the turning axis the position identifications for the tool nose correspond to the new +X direction.



4.2.3 Tape format for input of tool offsets

Tool offset values can be entered not only via the operator panel but also via the data input interface. In this case block numbers must not be programmed.

```
TOOL OFFSETS FOR SHAFT          Leader
% TO LF (TOOL OFFSET)           Offset start
G91 T1  X.. Z.. B.. A.. LF
G92 T16 X.. Z.. B.. A.. LF
G92 T13 X.. Z.. B.. A.. LF
M02 or M30 LF
```

T = Tool offset number

X = Tool geometry in X axis

Z = Tool geometry in Z axis

B = Tool nose radius

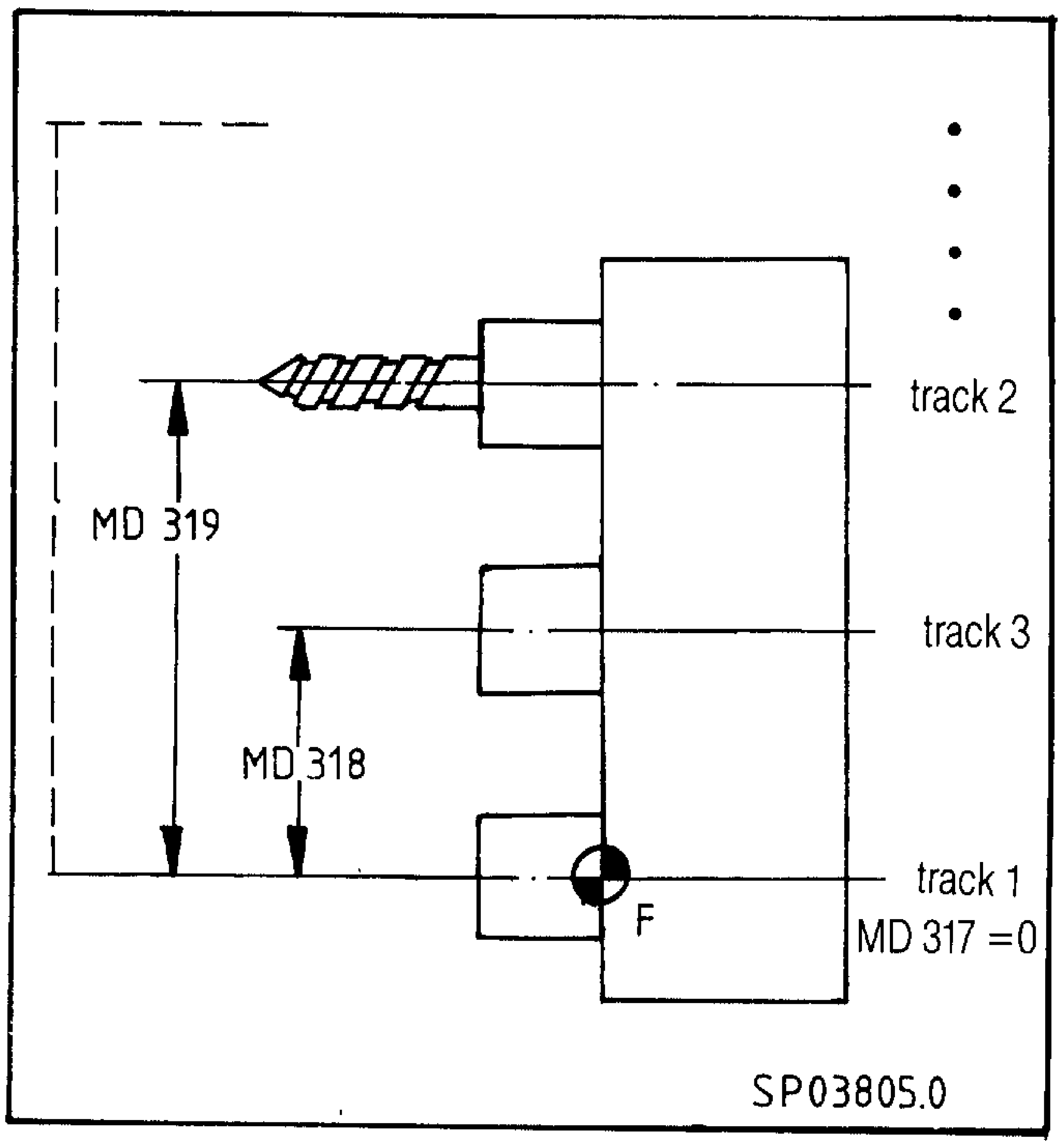
A = Tool nose position

4.2.3 Tool offset with tool track coding *

The tool track values stored in the machine data are calculated into the X axis like additional zero offsets. The PLC selects the track value (track 1 - 8).

- program @ 31 before each track change
- G53 deselects the offset
- track selection via the interface signals

Example:



Note:
For programs without tool track, a tool track with value 0 must be selected.

*) OPTION Basic Version 4B, 4C

4.3 Miscellaneous functions M

M00 Programmed stop (unconditional)

M00 makes it possible to interrupt the program in order to make a measurement or for some other similar purpose. After making the measurement the machine can be restarted by operating the "cycle" start key. The information stored in the controller is retained. The miscellaneous function M00 is operational in all automatic modes. Whether or not the spindle drive is stopped will be stated in the special Programming Instructions from the machine tool manufacturer. M00 is also effective in a block without position data. It is possible to program "M0".

M02 End of Program

M02 with program return to program start is written in the last block of the program.
M02 can be programmed on its own or together with other functions.
A read operation is stopped by M02.
M2 notation is also permitted. The control is placed in the reset state (see program key).

M17 End of subroutine

M17 is written in the last block (alone or with other functions) of a subroutine. M17 represents <u>only</u> the end of a subroutine. M17 may not be programmed in the same block in which a subroutine is called (when nesting).

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 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 [1)]

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 modal. When M19 is programmed without S, the stored value for the angle is active. M19 does not deselect M03 or M04.
The programming of M19 S123.4 LF is mandatory. M19 S must be alone in the block.

Unassigned auxiliary functions

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

1) Only with basic version 3, 4 and SINUMERIK 3TT

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 decaeds. Please refer to the Programming Instructions of the machine tool manufacturer for further details.

4.5 Rapid switching and auxiliary functions [2)]

If M-12, H-1234 is programmed, block processing in automatic mode is faster as PLC acknowledgement is not waited for. For S and T functions, all functions are output rapidly (can be selected via machine data). In the last block of a main program, no rapid auxiliary functions may be programmed.

1) Only with basic version 4 and SINUMERIK 3TT

2) Basic Version 4C

5.0 R parameters

In part programs, parameters R00 to R99 [1] can be assigned instead of numeric values to all addresses except N. For such a parameter, a certain numeric value is defined in the part programs or subroutines. R parameters must always be written in two decades. A maximum of 10 parameters may be written into a block.

From Basic Version 4C on, a maximum of 500 R parameters are available. They are divided into five groups of 100 R parameters each in groups 0 to 4. In each group, the parameters can be addressed via addresses R00 to R99. The groups can be selected via the 28x function (X = 0 to 4 depending on the group). The following examples and sections 5.1 to 5.5 apply for Basic Version 4C for one R parameter group.

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

Example:

```
L 5100              LF          Parameters R01, R05 and R29
N1  Z-R05  B-R01    LF          used in subroutine.
N2  XR29            LF
.
.
N50 M17             LF

% 5772  LF
N1 ...  LF
.
.
N37 R01 10. R29-20.05 R05 50. LF
N38 L5102                     LF   Call of subroutine 51
                                   2 passes
                                   R01 = 10.
                                   R05 = 50.
                                   R29 = 20.05
```

1) For basic versions 0,1,2: 50 parameters R00 to R49

5.1 Parameter definition

Defining the R parameter means that each R parameter is given a certain numeric value with sign.
The definition of the R parameter can be made in the part program and/ or subroutine. Up to 10 parameter definitions can be programmed in any one block.

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

5.2 Assignment of the parameter in the program

Direct assignment

An address is assigned directly to the value defined for an R parameter

Programmed Operation	Execution	Result
X R01	X R01	X = +10.78
Z R02	Z R02	Z = +95.34
X-R03	X R03	X = +555.1

Additive assignment

The defined value of an R parameter with the correct sign is assigned to the digital value of an address.

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

The sequence of address, numeric value, parameter must be observed.
No sign is interpreted as (+).

5.3 Parameter Linking

Type of calculation	Programmed calculation	Execution	Result found in
Adding	R01 R02	R01+R02	R01
Subtracting	R01-R02	R01-R02	R01
Multiplying	R01.R02	R01.R02	R01
Dividing	R01/R02	R01:R02	R01
Definition + addition	R01 10 R02	R01 = 10 R01+R02	R02
Definition + subtraction	R01 10 R02	R01-R02	R01
Square root 2)	@ 10 R01	R01	R01
Sine 2)	@ 15 R01	sin R01	R01
Arctan 2)	@ 18 R01	arctan R01 : R02	R01

7 .
The @ character is generated by pressing the – key and then the – key.
% R

During multiplication and division the linking of R parameters and absolute numbers is not permitted. The symbol for multiplication corresponds to that of the decimal point and the symbol for division corresponds to that for block delete. The sequence of links is decisive and not multiplication before addition. The calculation time per link is approx. 10 ms. One link may be programmed per block. @ 10, @ 15, and @ 18 must be in one block.

Range of values: smallest value: 1.10^{-8}
largest value: $22^{27}-1$

Display: Floating point (± .8) to (± 8.)

7 .
Note: The @ symbol is produced by pressing the – key and then the – key.
% R

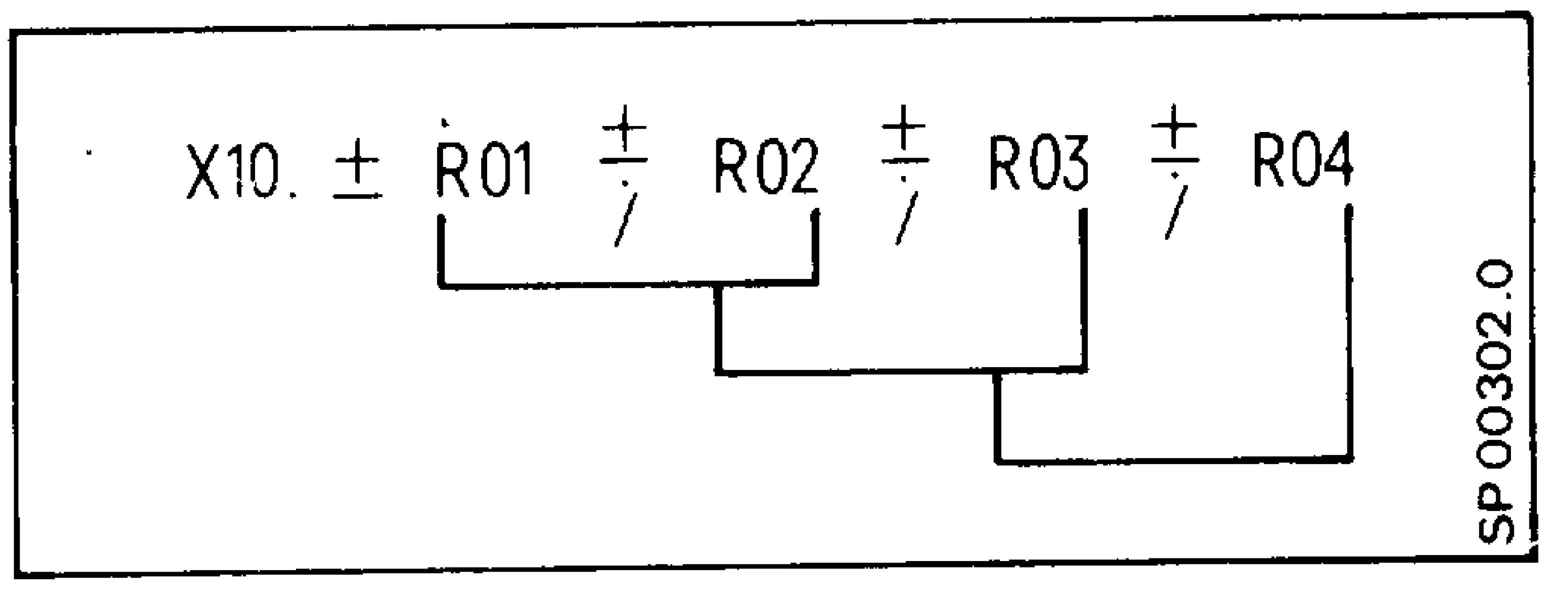
1) Basic versions 0, 1, 2 (± .7) to (± 7.)
2) Only for basic versions 3, 4 and SINUMERIK 3TT

5.4 Parameter chaining

Using parameter chaining certain values can be continually changed during multiple repetition of a program section or subroutine. A calculation is made during each programmed run through a chain. The last parameter of the chain is stored.

Four-fold chaining of parameters is the maximum possible.

Example:



Calculation:

During calculation of a new R parameter it is only the arithmetic sign between 2 parameters which is important. However, the parameter values do have a sign and can therefore be positive or negative.

Example of chaining 2 parameters:

R01 + R01	new R01 = R01+R02
-R01 + R02	new R01 = R01+R02
R01 - R02	new R01 = R01-R02
-R01 - R02	new R01 = R01-R02
R01 . R02	new R01 = R01.R02
-R01 / R02	new R01 = R01/R02

Example for chaining four parameters:

-R01+R02.R03-R04	new R01 = R01+R02
	new R02 = R02.R03
	new R03 = R03-R04
	new R04 = R04

Example:

```
% 9534 LF
.
N1 L0105 R01-10. R02 81. R03 3. LF
N6 L0204 R04-1.R05 4. R06-1. LF
.
N100 M30 LF
```

Subroutine call for the five or four passes and definition of the parameters.

R01=- 10. R02=81. R03=3.
R04=- 1. R05= 4. R06=-1.

```
L0100
N5  X1000.-R01+R02/R03 LF
N10 M17 LF
```

Using the parameters in the called subroutine.

```
L0200
N1  Z100.+R04.R05+R06 LF
N20 M17 LF
```

The following numerical values result

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

L0200		Z	-R04.	R05+	R06	
Pass	Definition	100.	-1.	4.	-1.	
1	Assigned value	99.	-4.	3.	-1.	
2	Assigned value	96.	-12.	2.	-1.	
3	Assigned value	88.	-24.	1.	-1.	
4	Assigned value	76.	-24.	0.	-1.	

At the end of the program, the parameters are given the last value to be determined. The value is retained until a new definition or a further change of parameters occurs.

5.5 Branch Conditions

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

5.5.1 @ 00 Unconditional jump

Application: Program sections may be omitted using the unconditional (absolute) jump.
Omitted blocks are not processed.

Programming: @ 00 ± 1234

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

It is possible to vary the branch destination using signed R parameters.
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.. Z..
N98 X.. Z..	N99
.	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
N21 @ 00 280	N215 @ 00 280 R01
.	.
.	.
.	.
.	.
N280 X.. Z..	N280 X.. Z..
.	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 skipped block).

5.5.2 @01, @02 and @03 Conditional jump

Application: Branch operations may be executed according to the following conditions:

equal to	@ 01
greater than	@ 02
greater than or equal to	@ 03

Programming: @ .. ± 1234 R.. R..

@ .. Operation with conditional jump

± Branch destination is located before (-) or after (+) the branch operation:
The operation is executed according to 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 than or equal to 2nd parameter @ 03

It is possible to vary the branch destination using signed R parameters.
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.. Z..	N98 X.. Z..
	N99 M..
	▶N100 X.. Z..
	.
└N215 @ 01 - 98 R01 R01	└N215 @ 02 - 98 R10 R01 R02
N216 G.. M..	N216 G.. M..
.	.
.	.

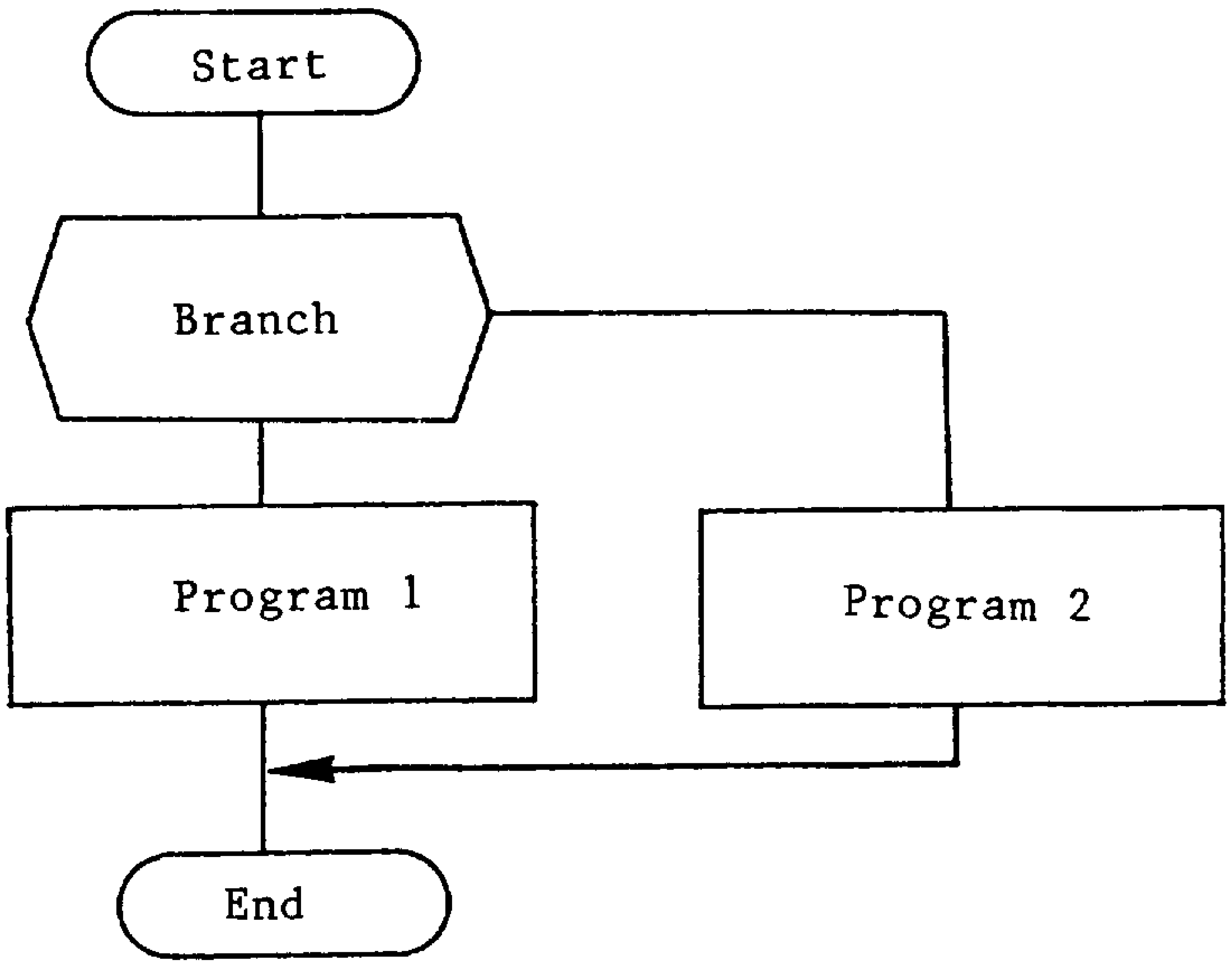
Branch destination located after (+) branch operation:

Standard case	Special case
	R16 has, for example, the value 5
┌N215 @ 03 + 280 R20 R25	┌N215 @ 01 + 280 R15 R20 R25
N220 G.. M..	N220 G.. M..
.	.
.	.
.	.
▶N280 X.. Z..	N280 X.. Z..
.	▶N285 G04
.	.

5.5.3 Conclusive 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 jumpt to program 2, if
│                                R10 = R20
│   N015  ┬
│   N020  Program 1
│   N025  │
│   N030  │
│   N035  ┴
│ ┌─N040  @  00 070
└─┼►N045  ┬
  │ N050  │
  │ N055  Program 2
  │ N060  │
  │ N065  ┴                      Unconditional jump to block 70,
  └►N070  G00 X...               in order to skip program 2
```

5.6 @31 Clearing the buffer

Application:

A series of control signals from the interface controller (parallel interface or PLC) are registered in the main memory of the NC indirectly via buffers. Associated with these control signals are the following functions:

- mirror image
- external zero offset
- R parameter input
- external tool offset

The control signals may be activated by using M functions.

If the functions actuated in the active program are to be effective in the block following their selection, the block buffers must be cleared, otherwise the selected control signal becomes active several blocks later.

The buffers may be cleared by using the function @31.

5.7 @ 10 "Square root"[1)]

Application:	Extract square root
Programming:	@ 10 R..

@10 Operation: extract square root

R.. R parameter for value entry and result

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 versions 3, 4 and SINUMERIK 3TT

5.8 @ 15 "Sine" [1)]

Application: Calculate the sine of an angle

Programming: @15 R..

@15 Operation: calculate sine

R.. R parameter for value entry and result

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 versions 3, 4 and SINUMERIK 3TT

5.9 @ 18 "Arctan" 1)

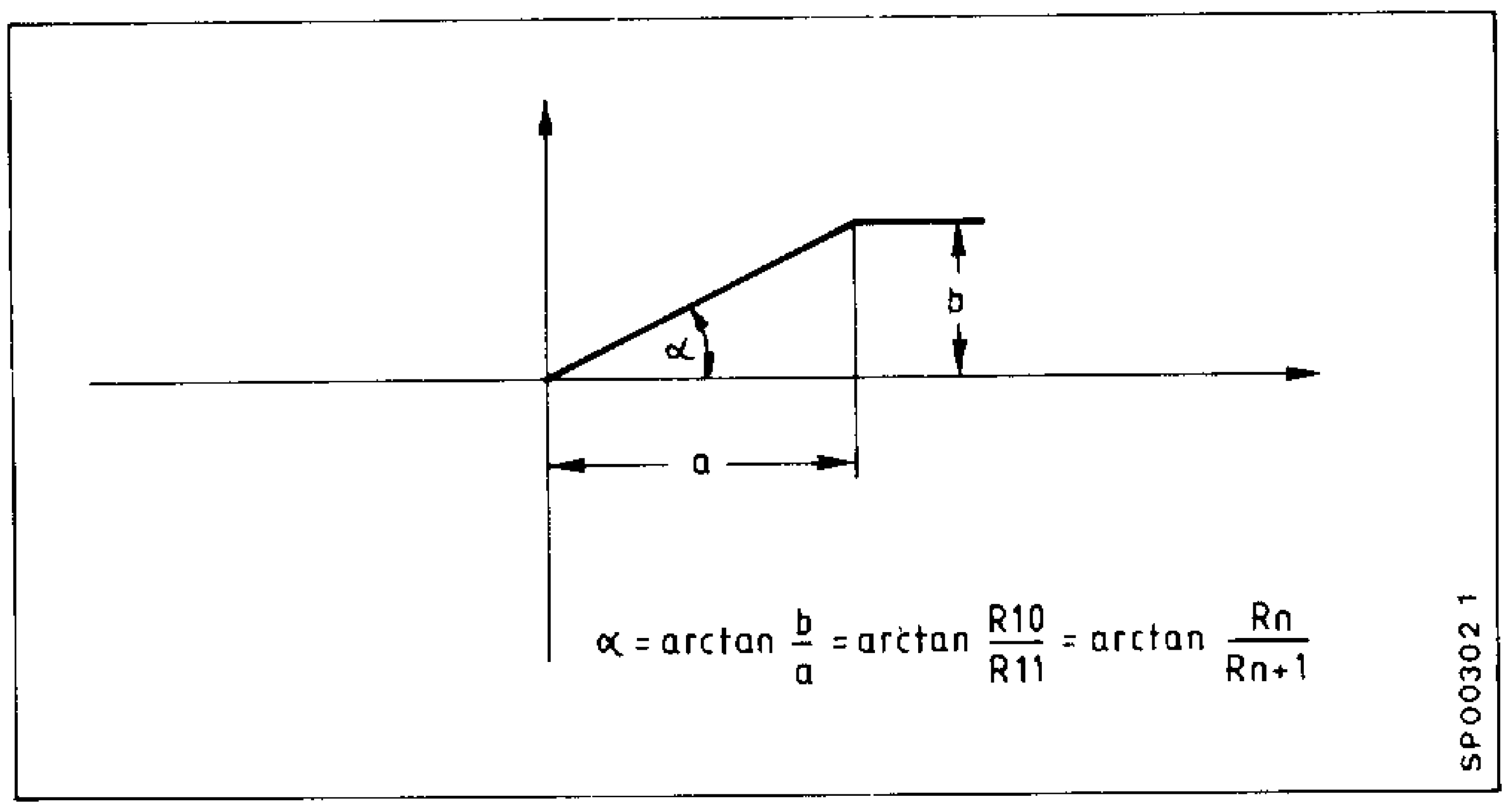
Application: Determination of an angle using the arctan function

Programming: @18 R..

@18 Operation: calculate arctan

R.. First R parameter for value entry b and result

Subsequent R parameters for value entry a



n Parameter No.

Example:

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

1) Only with basic versions 3, 4 and SINUMERIK 3TT

5.10 @ 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 adrresses. By varying the axis addresses the measuring cycle can operate in all NC-axes. Similar values for the numeric 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
Z	2

Programming:

a) Load address parameter @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.
LF	

1) Only with basic versions 3, 4 and SINUMERIK 3TT

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

N.. @20 @93 ± R49 LF

@20

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

LF

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.

5.11 @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

@90	Address parameter defined as X or Z axes.
±	Sign for the axis value
12345.678	Axis value
LF	

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

R01 12345.678 Value assignment

LF

1) Only with basic versions 3, 4 and SINUMERIK 3TT

N... @90 ± 0 ± R01 LF

@90	Address parameter defined as X or Z axis
±	Sign
0	Number value 0 cannot be eliminated
±	Sign
R01	R parameter is loaded with axis value

5.12 @22 Intersection calculation

The intersection calculation is a special function used for cutting cycles.
This function calculates the intersection between a circular interpolation block in R81 to R87 and a linear block which is programmed after @ 22. The result is stored in the R parameters R90, R91 and R92.

Input parameters

Basic Version 0 - 3	Basic Version 4 and 3TT	Parameter Assignment	
R10	R81	X	Block starting point
R11	R82	Z	(absolute)
R12	R83	X	Block end
R13	R84	Z	(absolute)
-	R85	I	Interpolation-
-	R86	K	parameters
-	R87	G	Function (00, 01, 02, 03)

Output parameters

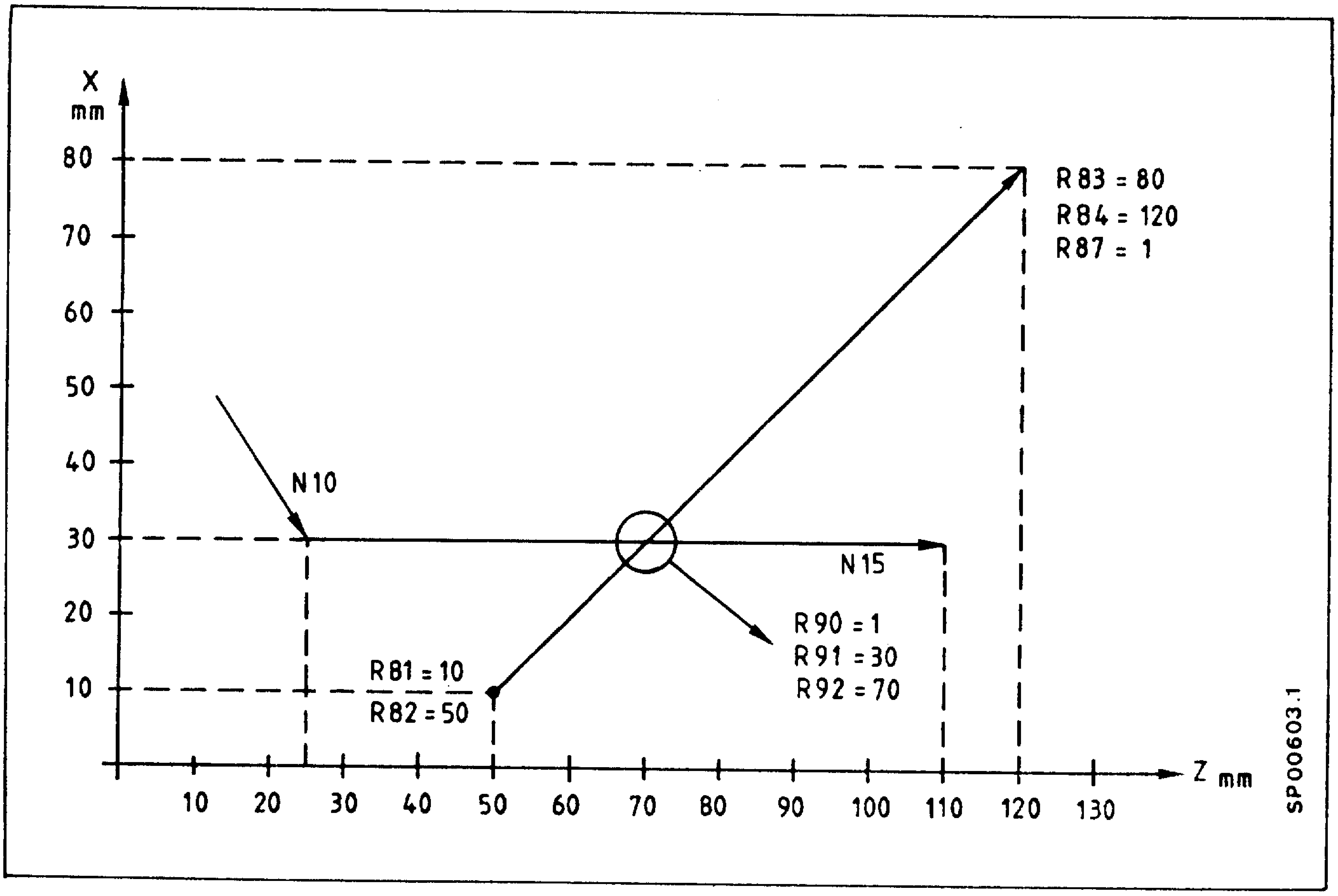
Basic Version 0 - 3	Basic Version 4 and 3TT	Parameter Assignment	
R47	R90	=0	No intersection found
		=1	Intersection found
R48	R91	X	Intersection (absolute)
R49	R92	Z	X value always in radius

Block Format:

N... G90 @ 22 G.. X.. Z..

G90	
@22	Intersection calculation
G..	G function (00, 01)
X.. Z..	Axis parameter (both axis parameters should always be programmed in absolute dimensions)

Example: Calculate intersection of two blocks
and store in R parameter



Programming:

```
.
.
.
N5   R81  10.  R82  50.  R83  80.  R84  120.  R87  1  LF
.
.
.
N10  G01  X30. Z25. LF         ← Traversing block
N15  @22 G01 X30. Z110. LF     ← Intersection is loaded
                                 in R91, R92
N20  X R91 Z R92 LF            ← Traversing movement
N25  X R83 Z R84 LF            ← Traversing movement
```

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

Application: Flexible access to the system memory locations of the NC control is 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 correction no.
- Tool correction
- Adjustable zero offset
- Programmable additive zero offset
- Current shift
- 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 (G59) is changed, @31 must be programmed beforehand. This is always necessary when system data which affect the paths traversed are changed by means of @29.

1) Only with basic versions 3, 4 and SINUMERIK 3TT

Programming:

```
N... @29  1   63   01   R17   LF
```

Operation: load/read the system memory

1st position:	1 = read system memory 2 = load system memory 3 = read machine data
2nd + 3rd position:	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:	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 values 13 = external zero offset (additional compensation) 18 = Background memory 19 = Special bit 20 = Current tool correction No. 34 = Value store for measuring data

R parameter (here R17).
This R parameter contains a 5-decade coded number and specifies the required memory from positions 4 + 5 of memory area defined by @ 29.

1st + 2nd position:	e.g. axis number (for further explanations see overview, Section 6.9)
3rd + 4th +5th position:	Identifying number e.g. number of the settable zero offset (further explanations in Section 6.9).

Example:

Application:	The R parameter R01 is loaded during a gauging cycle for axis X with zero offset value. This measured value should 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 R04	165.015	Z axis of a new zero offset value 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

@29						RAB			
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 parameters	01	Tool offset	01 02 03 04	Length X Z Radius Position	001 to 034	max. 34 groups
1 2	Read Write	00 to 99	max. 100 R parameters	03	Settable zero offset	01 02	Axis 1 Axis 2	001 to 012	max. 4 groups for basic control 3,4
1 2	Read Write	00 to 99	max. 100 R parameters	04	Programmable additive zero offset	01 02 08	Axis 1 Axis 2	001	1 group
1	Read	00 to 99	max. 100 R parameters	08	Current offset (Σ Tool offset) (also mirrored) + Σ zero offset	01 02	Axis 1 Axis 2	001	1 group
1 2	Read Write	00 to 99	max. 100 R parameters	09	R parameters	00	No identifier	000 to 099	100 numbers
1	Read	00 to 99	max. 100 R parameters	10	Machine data	00 00 to 04	No identifier R parameter group identifier**	100 to 479	380 numbers
3	Read bit	00 to 99	max. 100 R parameters	10	Machine data bit	00 to 07	Bit No.	400 to 419	19 numbers
1	Read	00 to 99	max. 100 R parameters	12	Actual value (axes)	01 02	Axis 2 Axis 1	001	1 group
1 2	Read Write	00 to 99	max. 100 R parameters	18	Buffer memory	00	No identifier	000 to 099	100 numbers
3	Read bit	00 to 99	max. 100 R parameters	19	Special bit	00 to 07	Bit No.	001	1 group
1	Read	00 to 99	max. 100 R parameters	20	Current tool offset number	00	No identifier	001	1 group
1	Read	00 to 99	max. 100 R parameters	12	Spindle actual value	05	Spindle	001	1 group
1 2	Read Write	00 to 99	max. 100 R parameters	13	External zero offset (external additive compensation)	01 02	Axis 1 Axis 2	001	1 group
1 2	Read Write	00 to 99	max. 100 R parameters	34	Value store	00	No identifier	000 to 099	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 4 C on)
Bit 5 to bit 7 : free

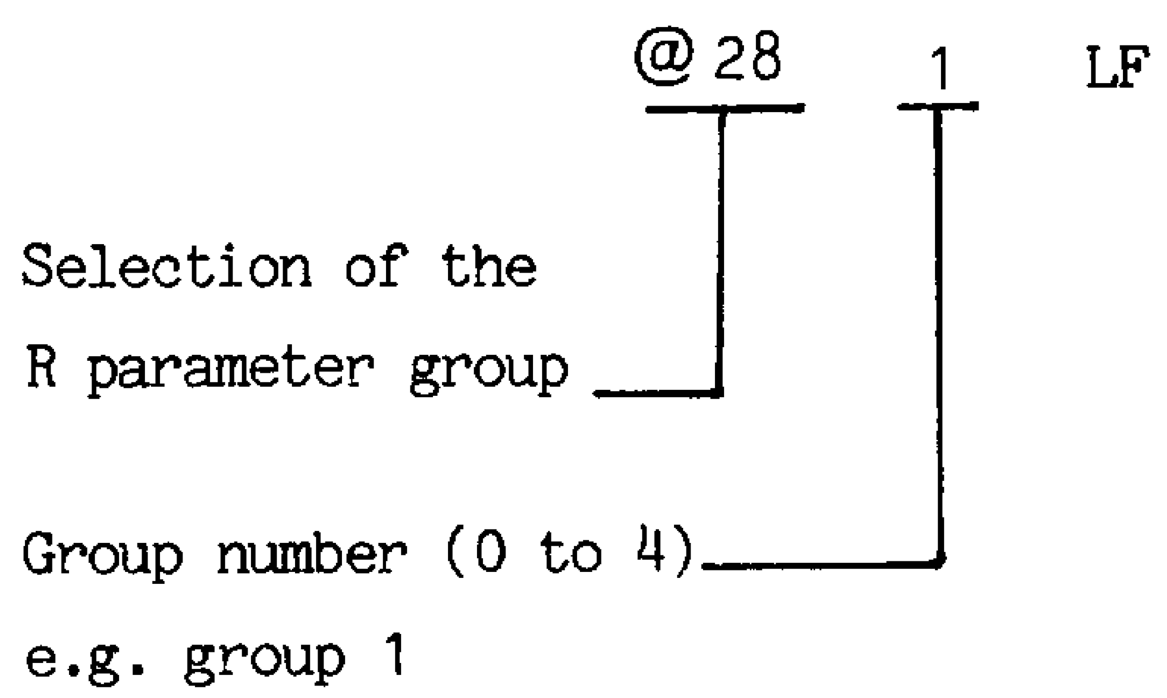
** from Basic Version 4 C

RAB is a freely selectable R parameter (AB = 00..99). In RAB the system memory location is defined from which or into which is to be loaded.

5.14 @28 Selection of the R parameter group (only Basic Version 4C)

Application: With this function, one of the groups 0 to 4 can be selected for the R parameters. Each group comprises 100 R parameters (R00 to R99).

Programming:



On Power ON, Reset and M02/M30, group 0 is automatically selected.

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

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 supply (R10) with R77 from group 1
N15 @29  16609 R10    @29 read R parameters
                      target=R66, source=R77
                      (address for source in R10)
```

5.15 @ 21 "Reference Editing"

Reference editing is a special function for the implementation of stock removal. With stock removal the contour is stored in a subroutine. The contour can be programmed with all auxiliary aids of the controller (blueprint programm, radius programming).

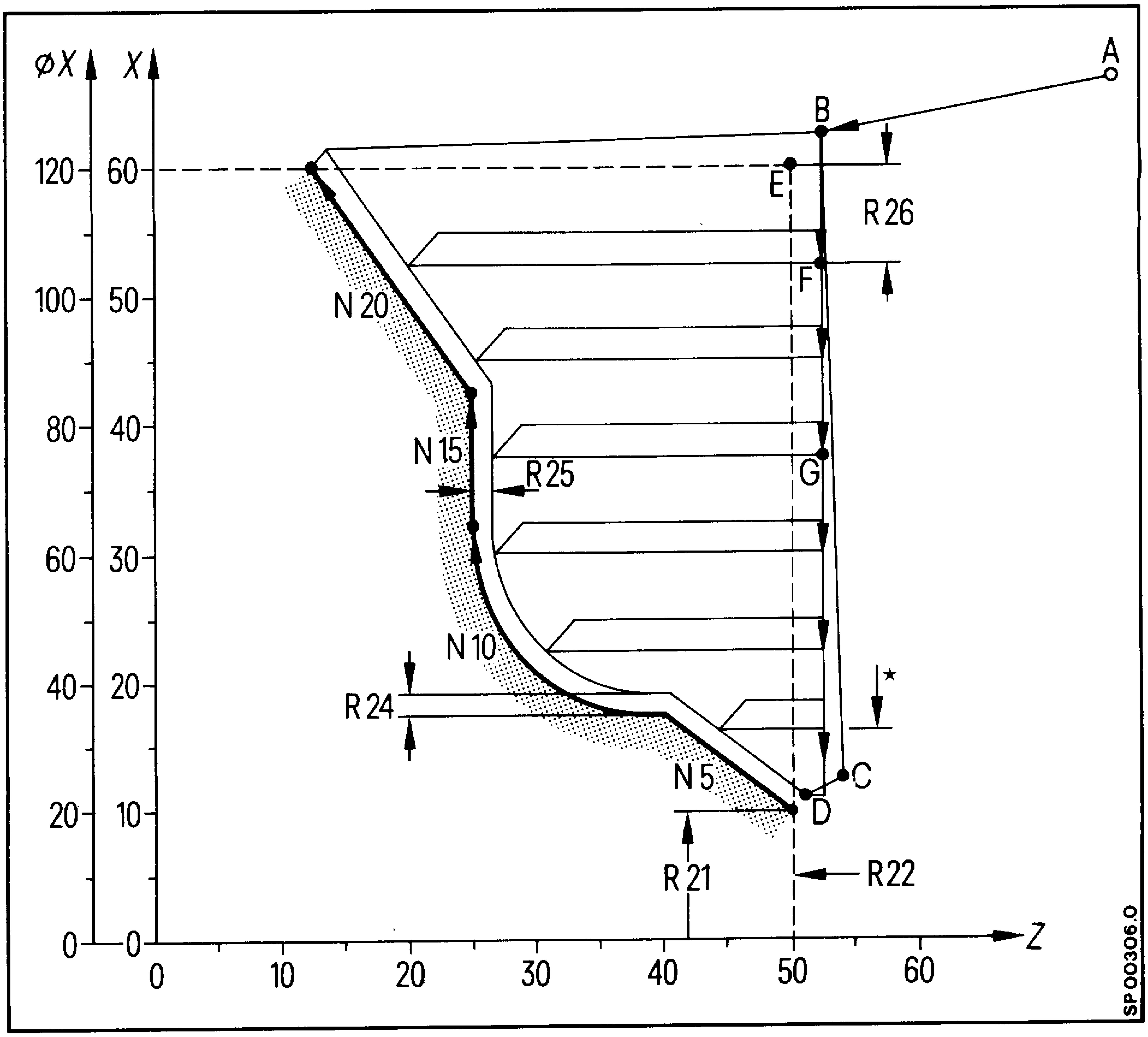
Since the stock removal cycle is programmed as a "normal subroutine", all block data are needed internally as R parameters. Now the reference editing makes certain that the programmed contour is broken down into single blocks and the data are deposited in fixed R parameters.

Programming: N...@ 21 LF (own block)

Instruction: reference editing

@21 is always programmed when new values must be provided for the interface calculation @22. This is always the case when a new contour element appears in the sequence of editing.

Example:



The contour is programmed through L50 as follows:

```
L5000
N5 G90 G01 X35 Z40
N10 G02 X65 Z25 I15 K0
N15 G01 X85
N20 X120 Z15 M17
```

When calling up the stock removal cycle (L95), the controller is informed:

via R20 50 of the subroutine number,
via R21 20 of the starting point X,
via R22 50 of the starting point Z.

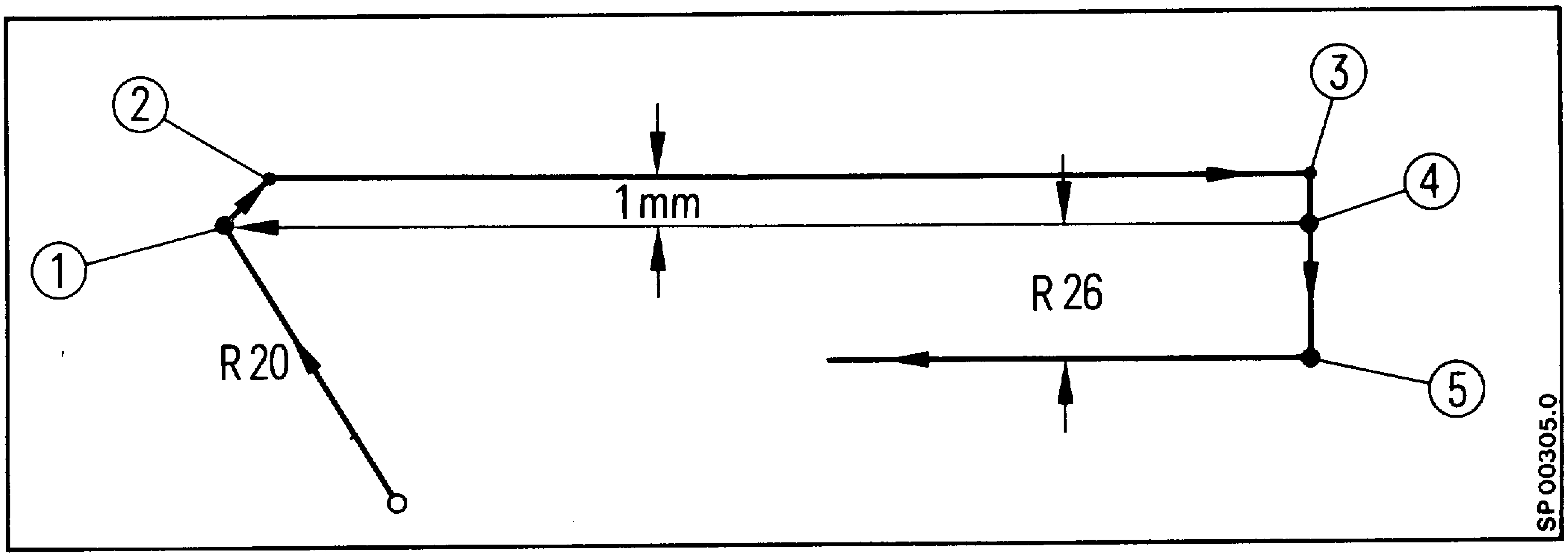
In the case of longitudinal stock removal the stock removal procedure begins from Point F. Reference editing @21 and the intersection calculation @22 are called repeatedly by the stock removal cycle until the intersection calculation finds a point of intersection.
In our example this is the case in block N20 of subroutine L50.

Program: Stock Removal Cycle

Sequence of the functions called	Block Starting point		Block end		Interpolation		G. function	Output control parameters
	X	Z	X	Z	I	K	G	
	R81	R82	R83	R84	R85	R86	R87	R88
@ a9	20	50	35	40	-	-	1	0
1. @ aa intersect.?no								
@ a9	35	40	65	25	15	0	2	0
2. @ aa intersect.?no								
@ a9	65	25	85	25	-	-	1	0
3. @ aa intersect.?no								
@ a9	85	25	120	15	-	-	1	0
4. @ aa intersection? yes								

The table shows how, through the reference editing function @21, the blocks of subroutine L50 are broken down into parameters. The intersection calculation @22 checks whether it comes to an intersection. If this is not the case, then the next block is broken down. In this example an intersection (1) is found at the 4th attempt. This point of intersection (possibly corrected by the depth of finishing cut) is started and thereafter withdrawn at an angle of 45^{o} to 1 mm safety distance (2) and Position (3) is approached at rapid traverse.

After infeed at the roughing depth R26 plus safety distance 1 mm) (5), the interception calculation @22 is then called again internally.



R26 Roughing depth

This sequence is repeated until no intersection in block N20 is found through the current infeed by the roughing depth. The following block is broken down by @21. This is M17. The control parameter R88 is set to 1. A logical check is made whether the cutting is finished. This is not the case. Via @21 the first block of the subroutine is again broken down. A check takes place again whether an intersection is present. In block N5, this is not the case. The control parameter R88 is set to 0. At block N15 the first new intersection is then found, etc.

Note:

Before calling up @21, the control parameter R88 must be set to 1. This ensures that the first block of the subroutine is broken down into parameters.

Further input parameters are:

R20 Number of the subroutine

R21 Starting point of the contour (absolute X value)

R22 Starting point of the contour (absolute Z value)

When @21 is used in the stock removal cycle, these parameters are automatically be set during the parameter definition of the cycle. R20 to R22 may not be changed in the cycle.

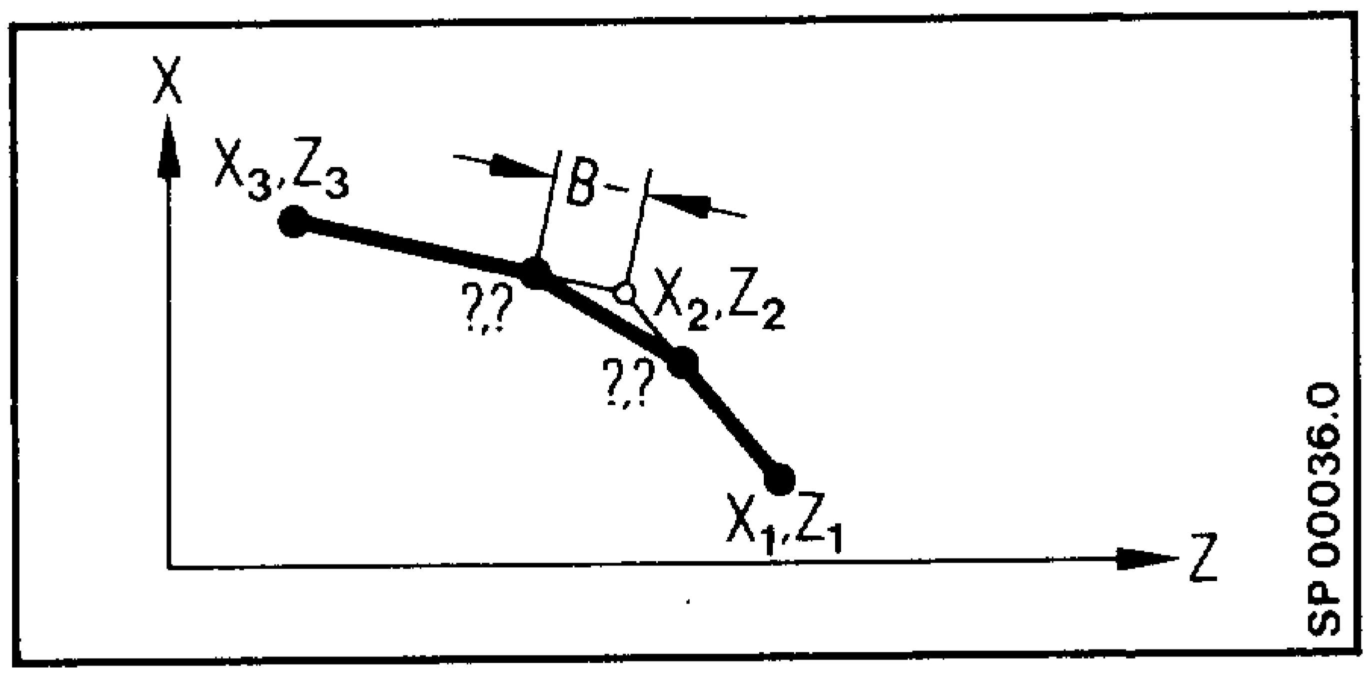
6.0 Automatic insertion of chamfers and radii

At corners a chamfer or a radius can be inserted automatically.
The examples 1 and 2 show these basic elements.

B-.. means insertion of a chamfer

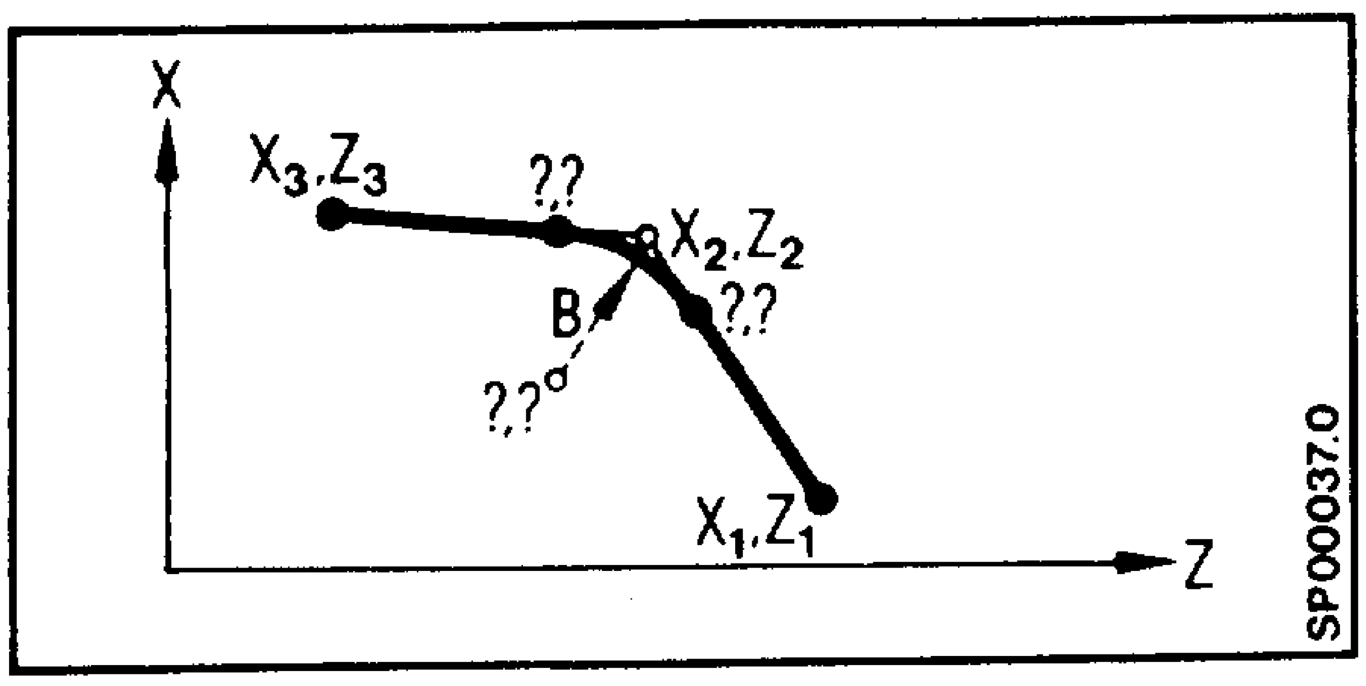
B.. means insertion of a radius

1 Chamfer



N.... X_2... Z_2 ... B-...
N152 X_3... Z_3...
The inserted chamfer must not be bigger than the shorter of the two paths.

2 Radius



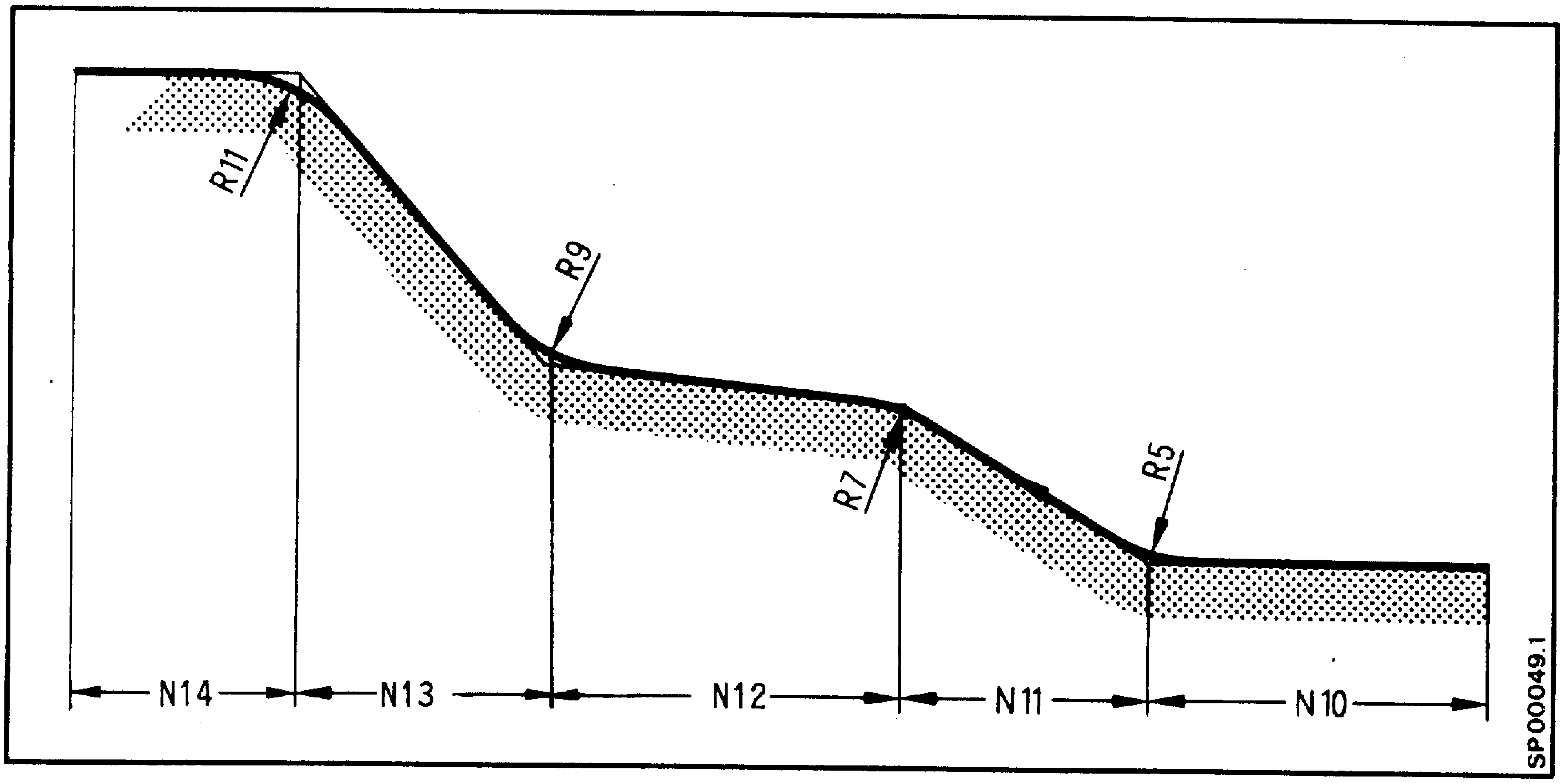
N... X_2... Z_2... B...
N... X_3... Z_3...

The inserted radius must not be bigger than the shorter of the two paths.

6.1 Linking of blocks

Blocks may be linked with inserted radii or chamfers in any order.

Example:



.

.

.

N10 Z... B5. LF

N11 Z... X... B7. LF

N12 Z... X... LF

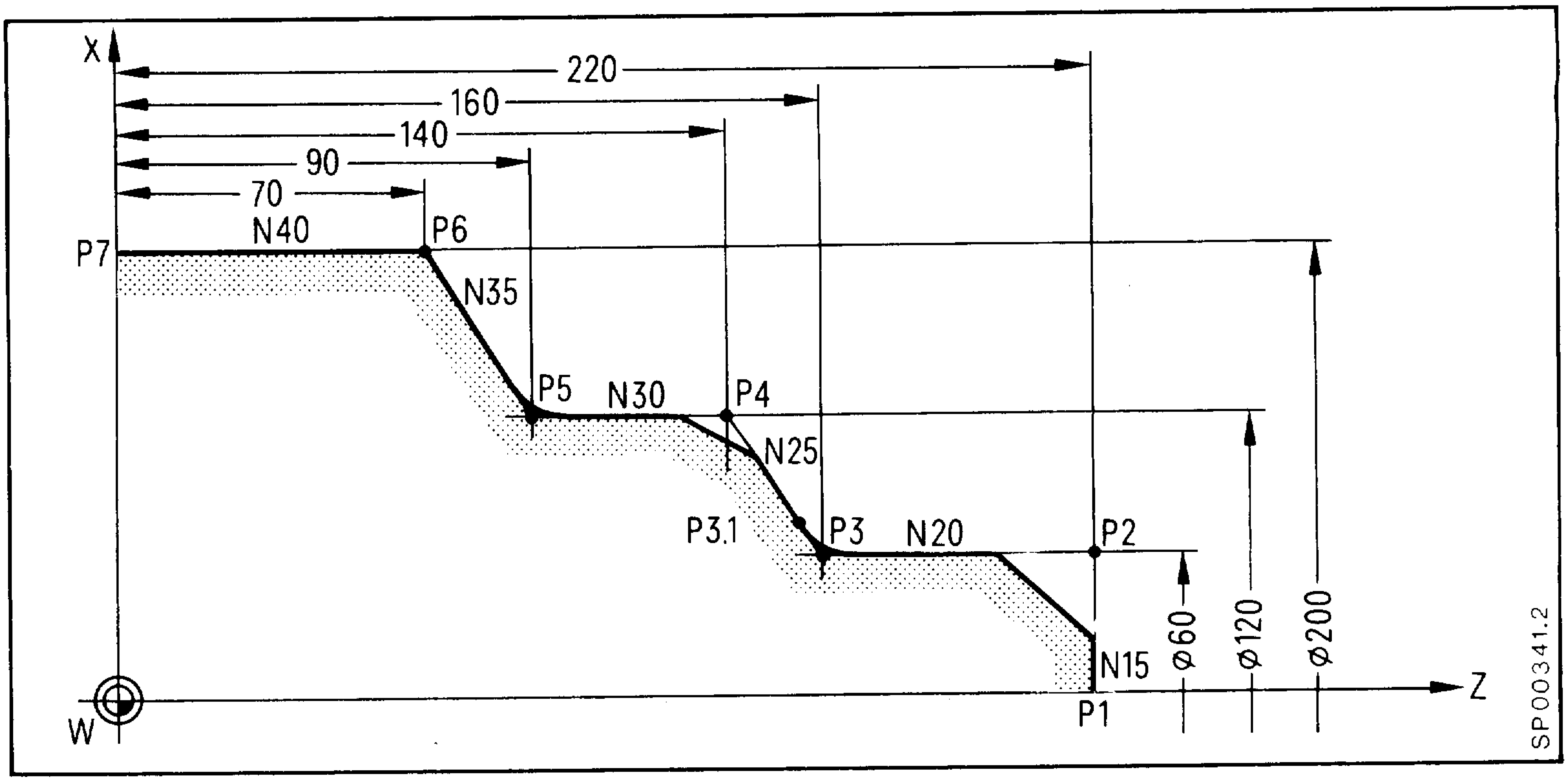
N13 Z...X... B11.LF

N14 Z...LF

.

.

6.2 Example



.

.

.

```
N10 G00 G90 G42 Z220. X0. LF (P1)
N15 G01 X60. B-20.        LF (P2)
N20     Z160. B5.         LF (P3)
N25     X120. Z140. B-5.  LF (P4)
N30     Z90.  B5.         LF (P5)
N35     X200. Z70.        LF (P6)
N40     Z0.               LF (P7)
```

.

.

.

6.3 Switching and auxiliary functions in linked blocks

Linked blocks result whenever blocks intersect due to radii or chamfers.
A block with a switching function may be inserted between linked blocks.

Example: see diagram on page 6-1

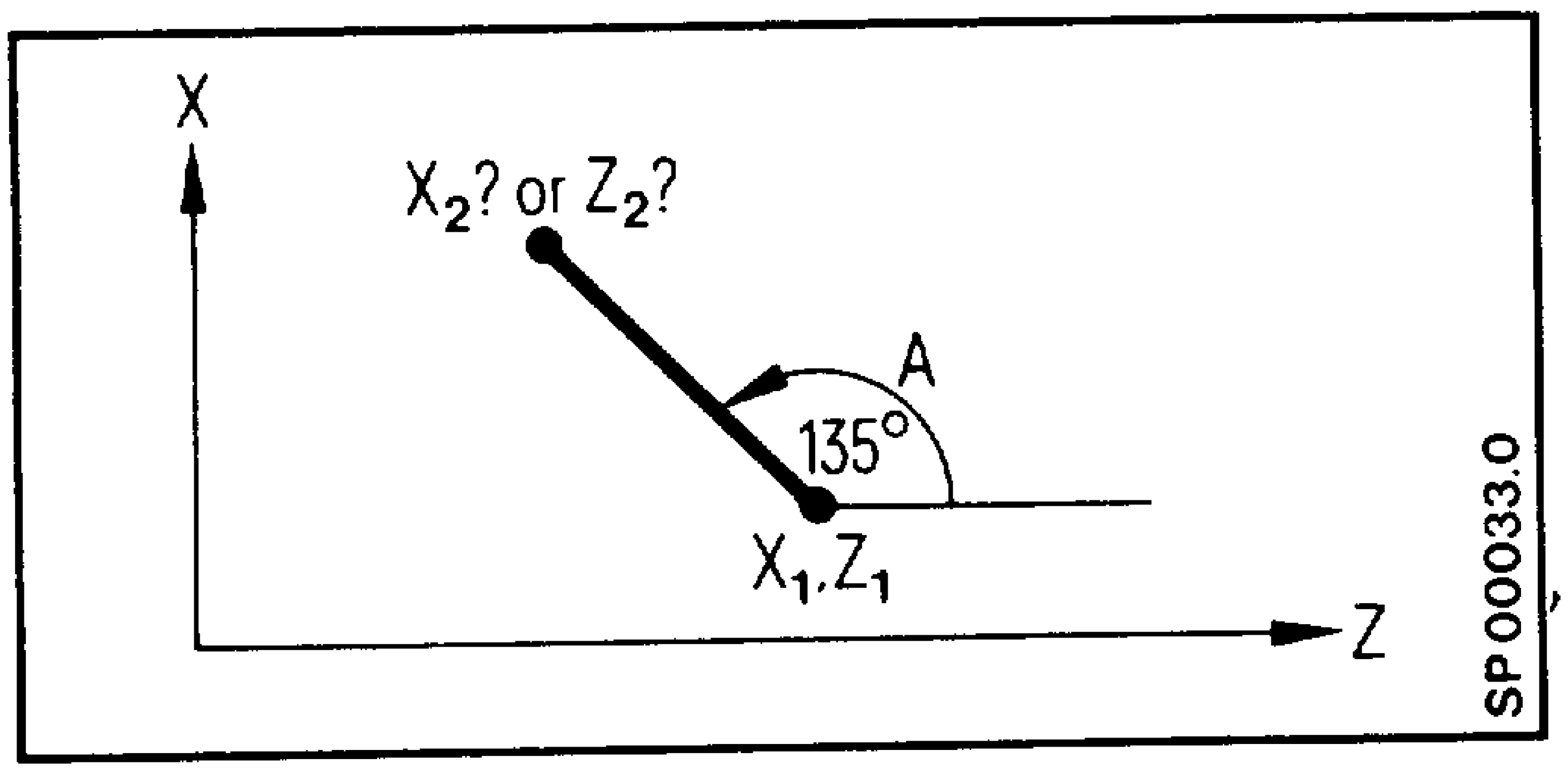
```
.
.
.
N20 Z160. B5.        LF (P3)
N21 M..              LF (P3.1)
N25 X120. Z140. B-5. LF (P4)
.
.
.
```

The switching function becomes effective at point P3.1. Undercutting therefore takes place at point P3.1.
At transition points (chamfer insertion) within a contour section, the controller automatically generates a G09.

6.4. Blueprint programming

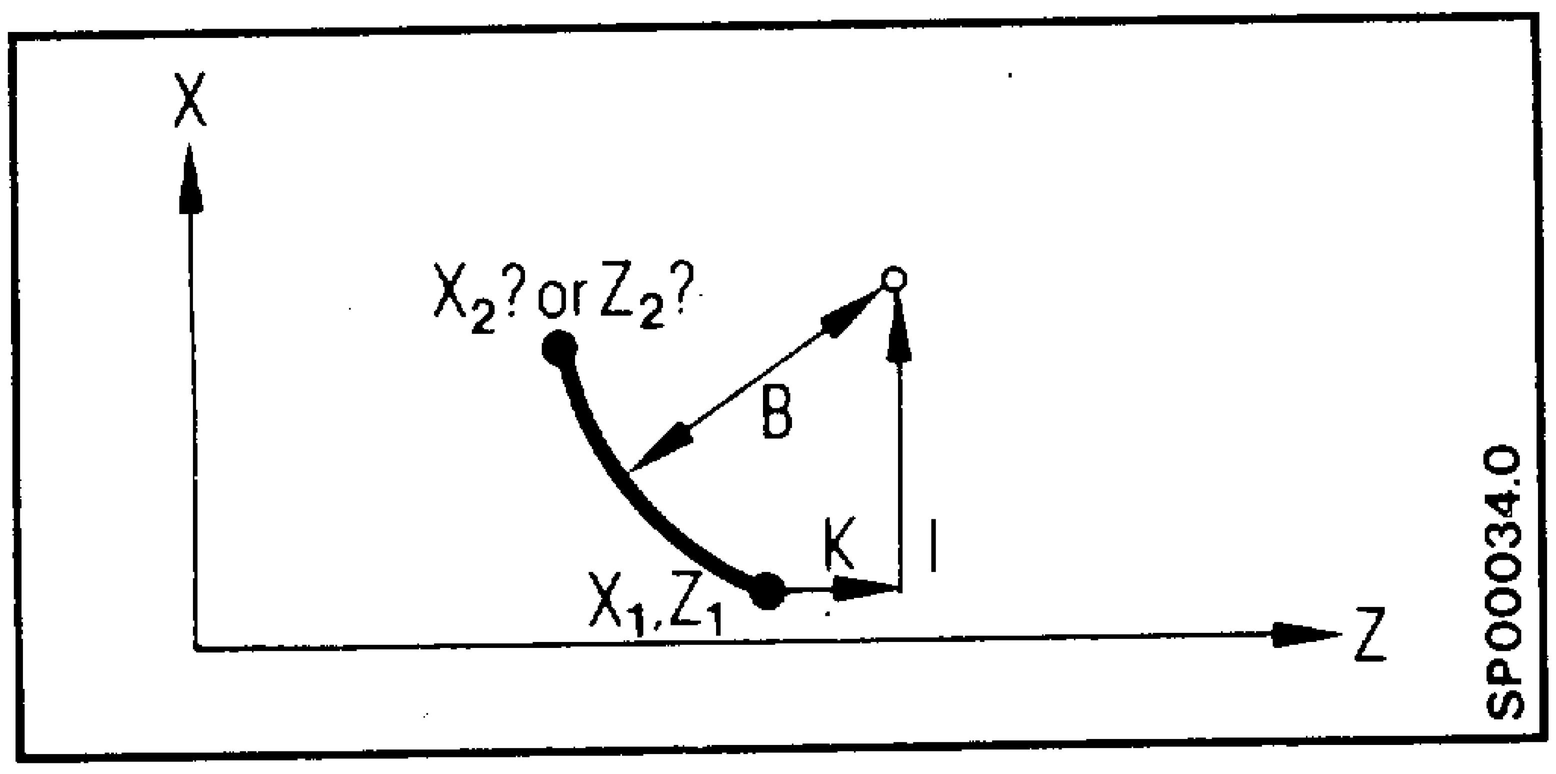
The examples (1) to (8) represent the basic elements of blueprint programming. These basic elements can be combined in various ways.

(1) 2 point definition



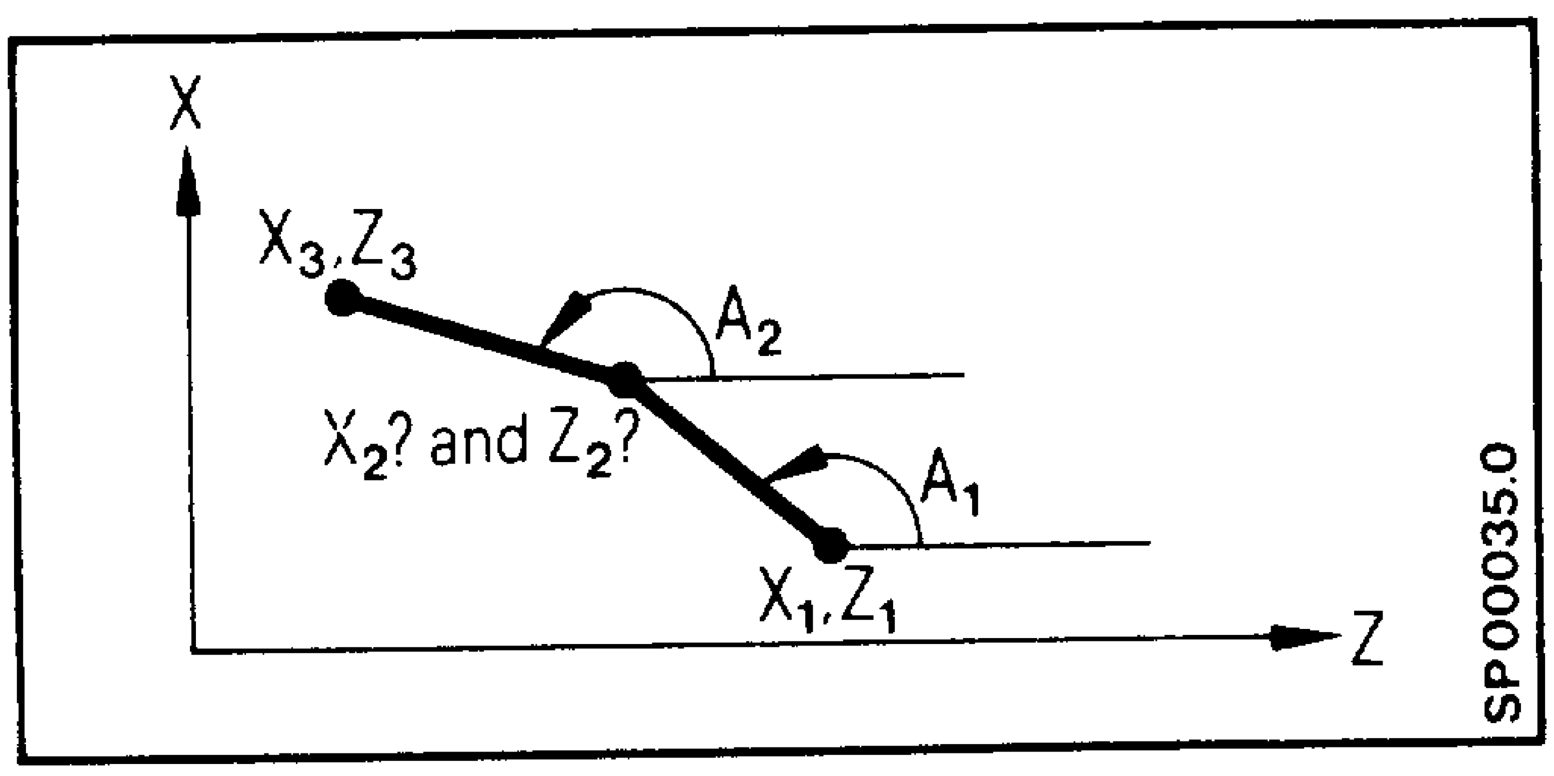
N...A...X_2 (or Z_2)
The second end coordinate is calculated by the controller.

(2) Circlular arc



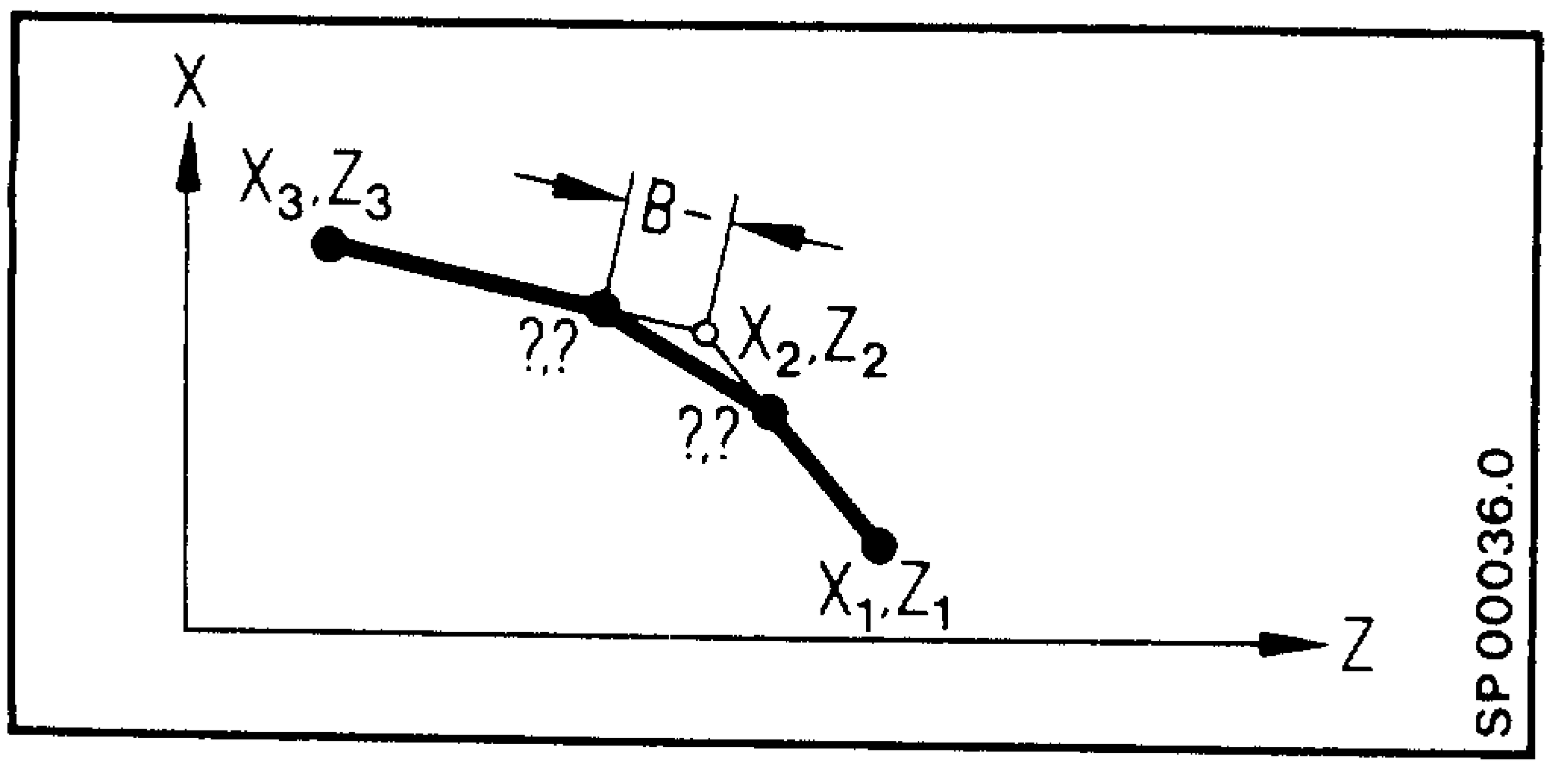
N...G02 (or G03) I..K..B..X_2 (or Z_2)
The circular arc is limited to one quadrant.
The second end coordinate is calculated by the controller.
In the contour section, both parameters I and K must be programmed, even if a value is zero.

(3) 3 point definition



N...A_1....A_2...X_3...Z_2
The controller calculates the coordinates (X_2, Z_2) and generates 2 blocks. The angle A_2 relates to the restart point which is not programmed (X_2,Z_2)

(4) Chamfer



N151 X_2...Z_2..B-...

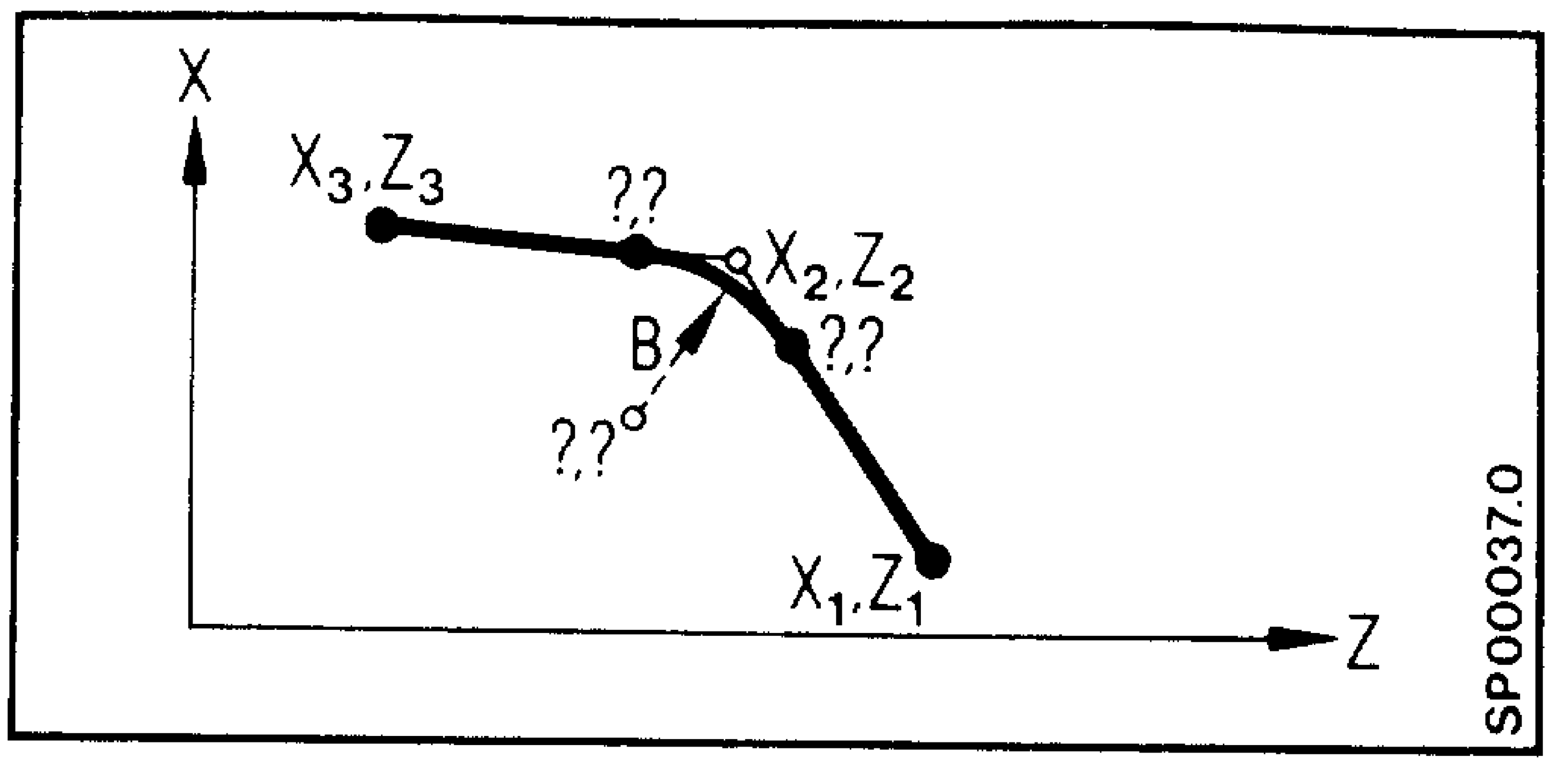
N152 X_3...Z_3 *

B-... means to insert a chamfer

B... means to insert a radius

(the "minus" sign has no meaning here as a sign, but it is a special identifier for B as chamfer).

(5) Radius

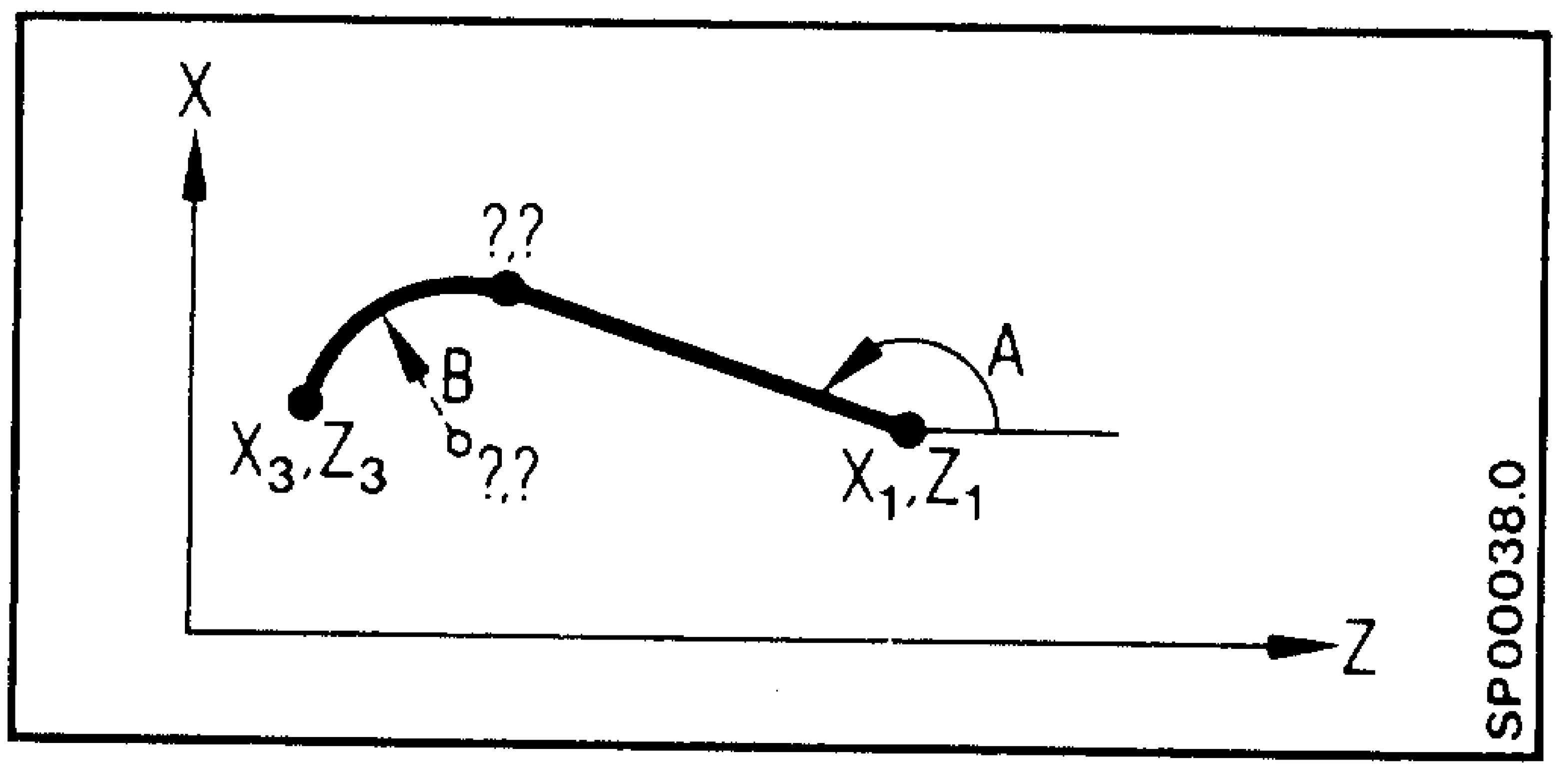


N...X_2...Z_2....B...

N...X_2...Z_2...   *

The inserted radius may not be larger than the shorter distance.

(6) Straight line - circular arc (tangential)



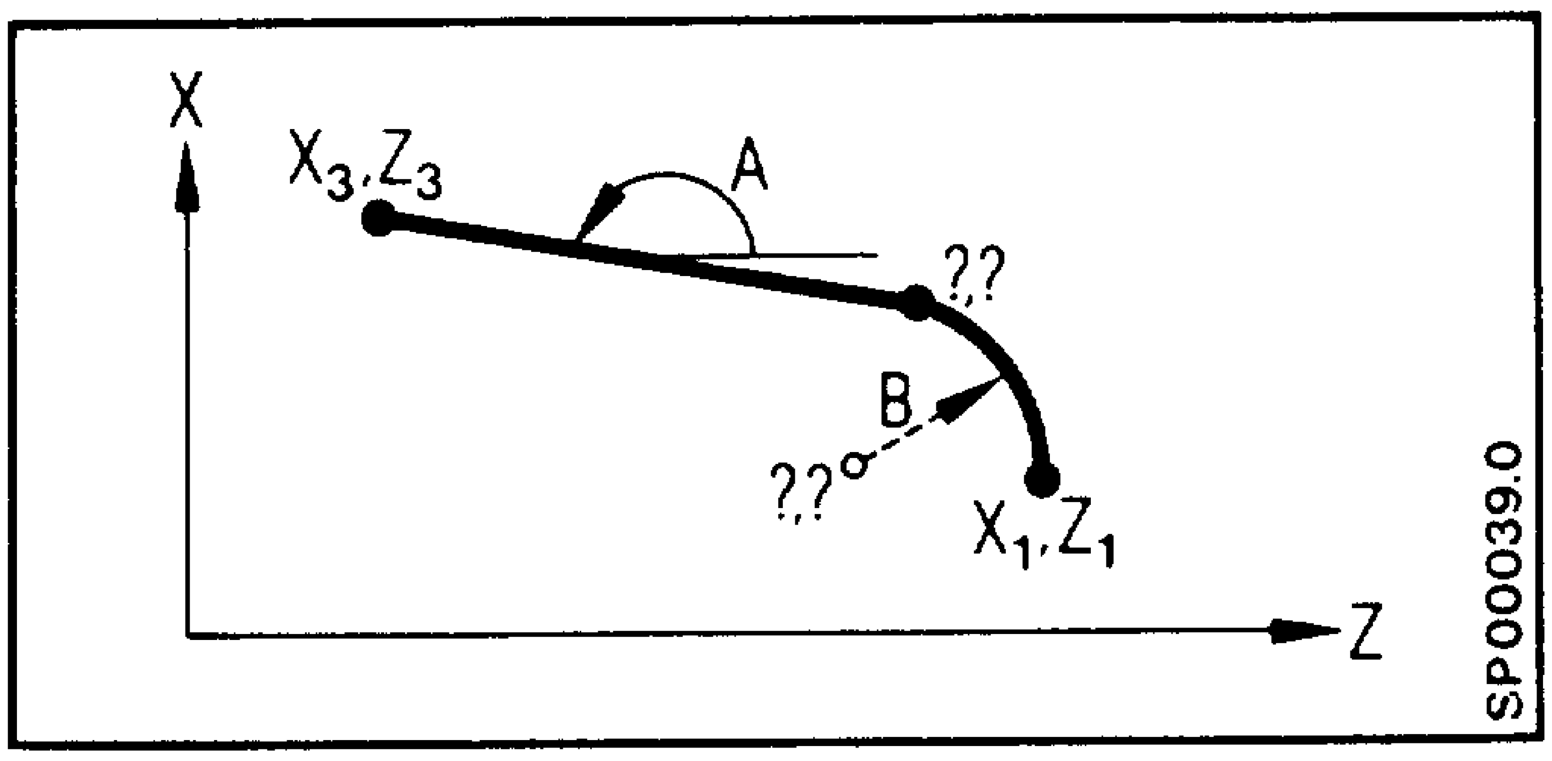
N..G02 (or G03) A.. B..X_3..Z_3..

circular arc not over 180°.

The sequence A (Angle) and B (Radius) must be observed.

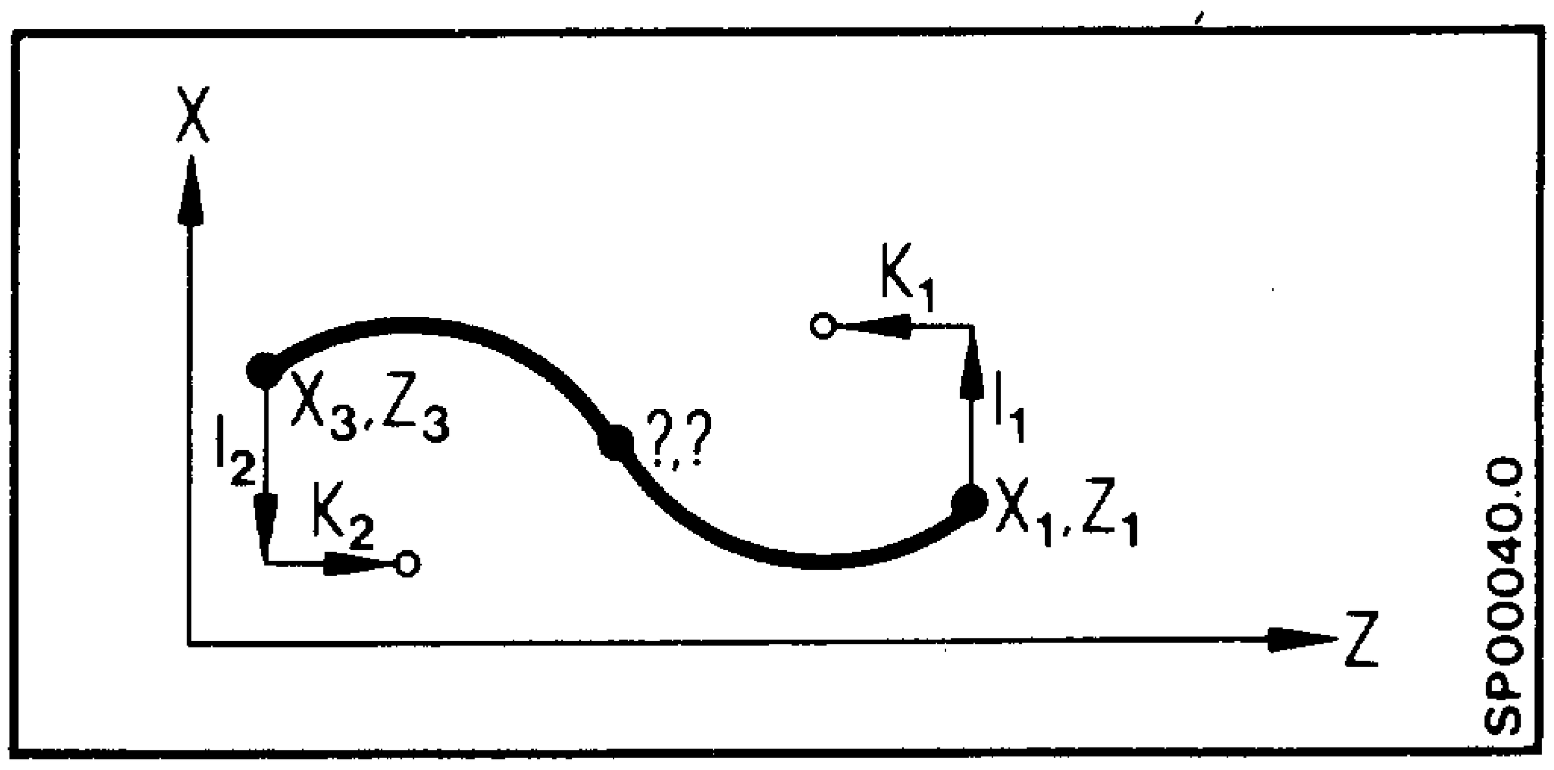
* Second block can also be a blueprint programmed block.

(7) Circular arc - straight line (tangential)



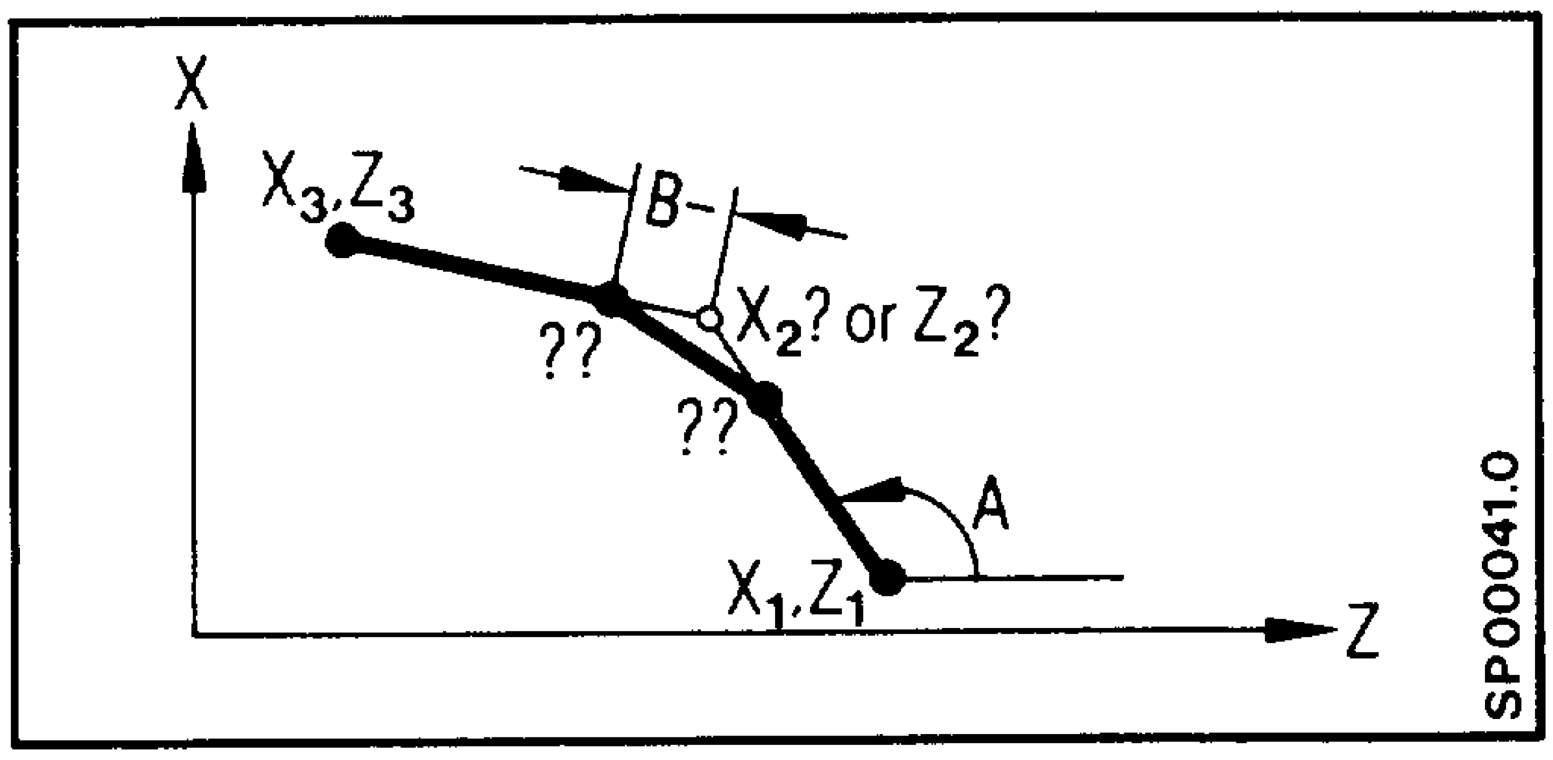
N..G02 (or G03) B..A..X_3..Z_3..
Circular arc not over 180°.
The sequence B, A must be observed.
No radius can be inserted in X_3, Z_3.

(8) Circular arc - Circular arc (tangential)



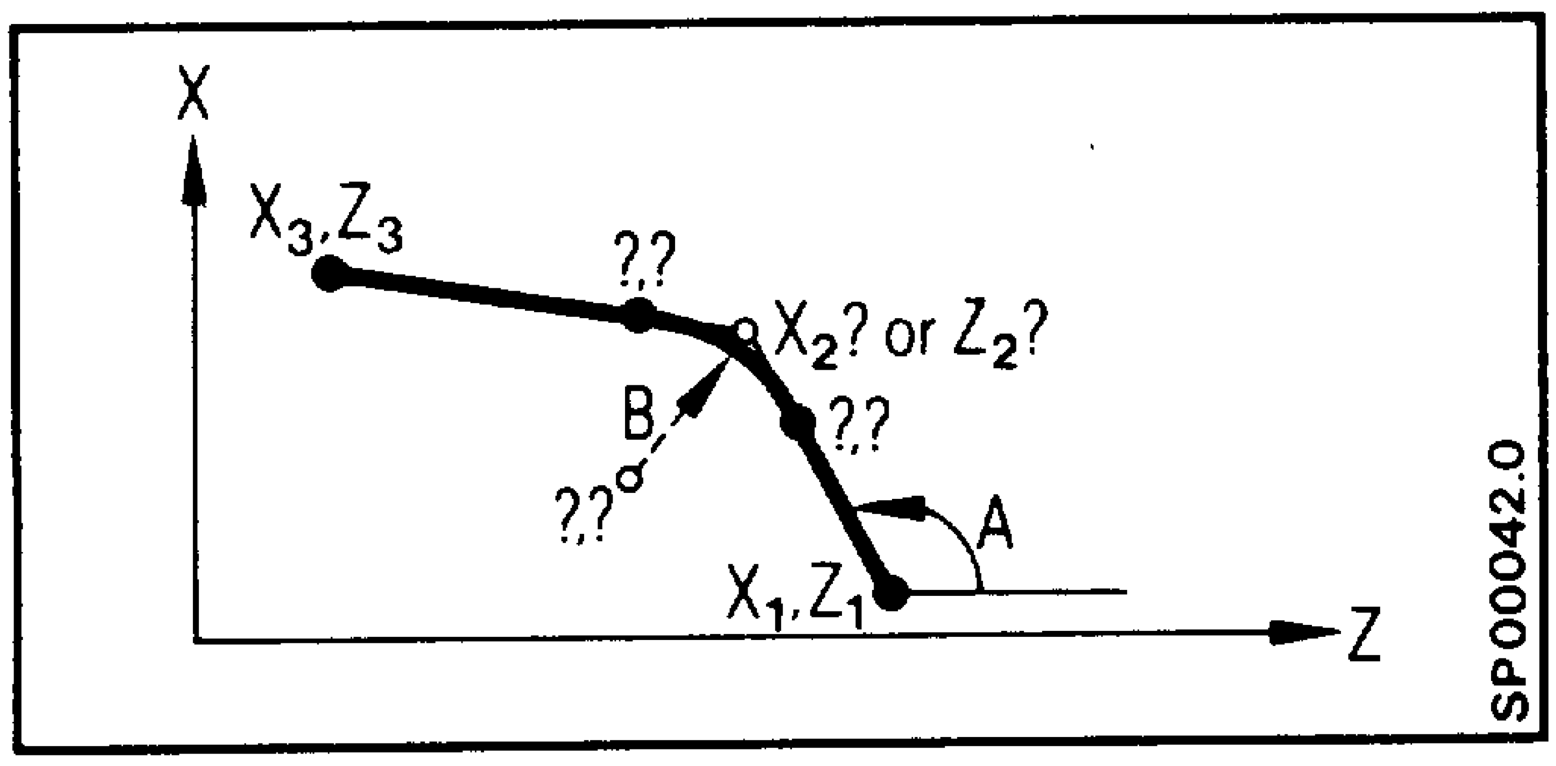
N...G02 (or G03) I_1..K_1..I_2..
K_2..X_3..Z_3
The preparatory function is programmed for the first arc. The second preparatory function is always opposite and is not programmed.

(1) + (4) 2-point-definition + chamfer



N15 A..X_2 (or Z_2..) B-..
N16 X_3..Z_3

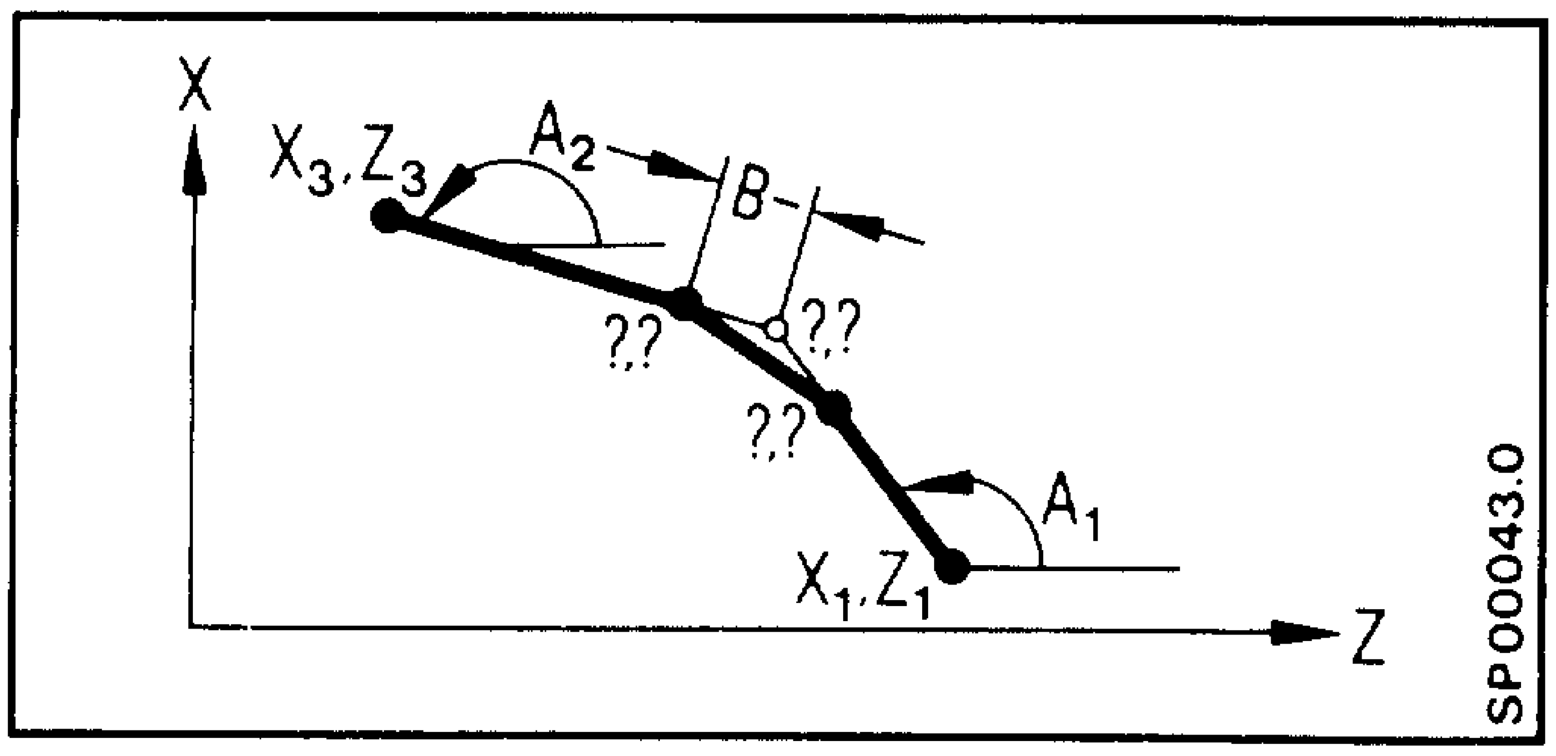
(1) + (5) 2-point-definition + radius



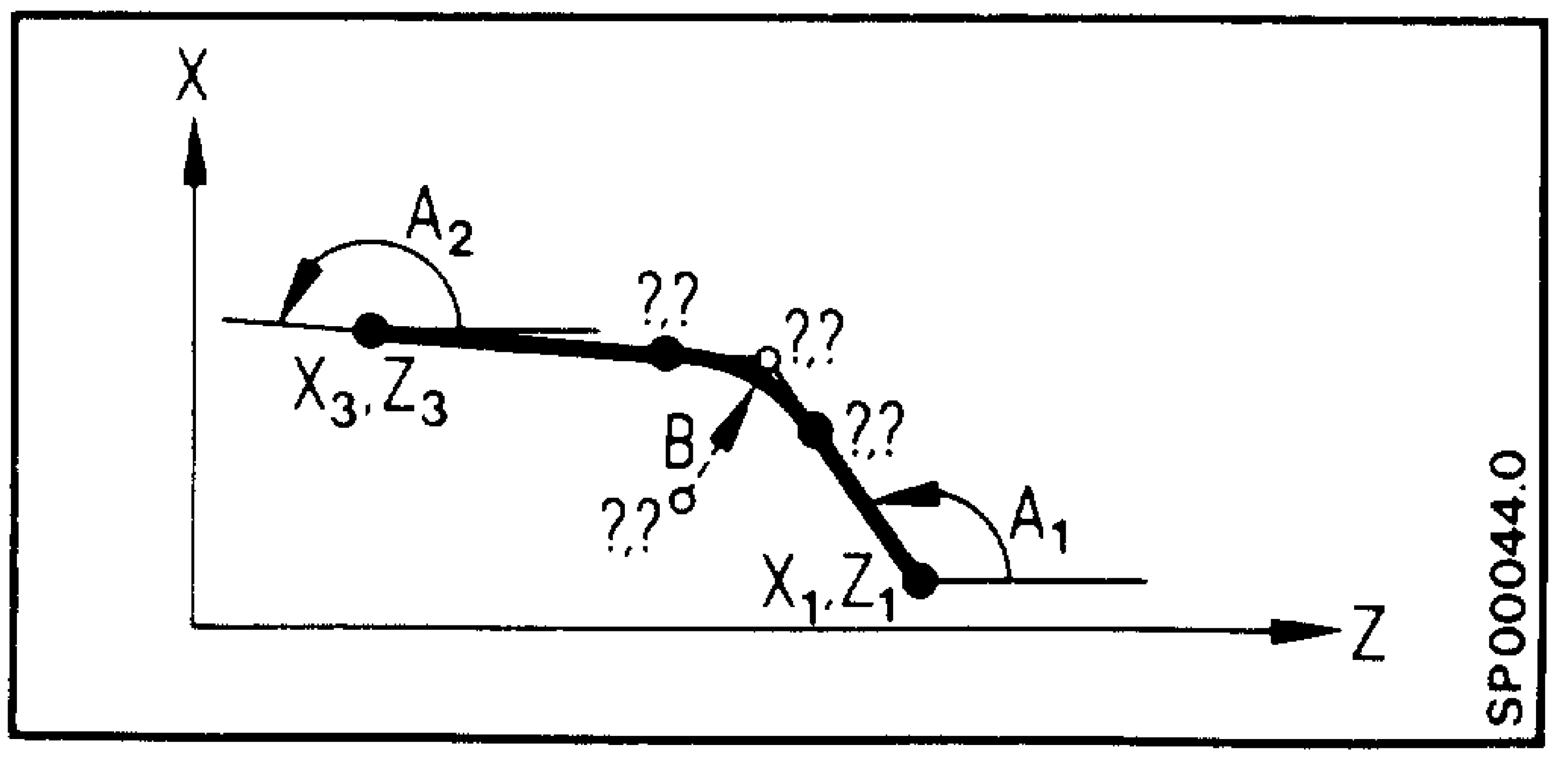
N15 A..X_2...(orZ_2...)B...
N16 X_3...Z_3..
The inserted radius may not be larger than the shorter distance.

* Second block can also be a blueprint programmed block.

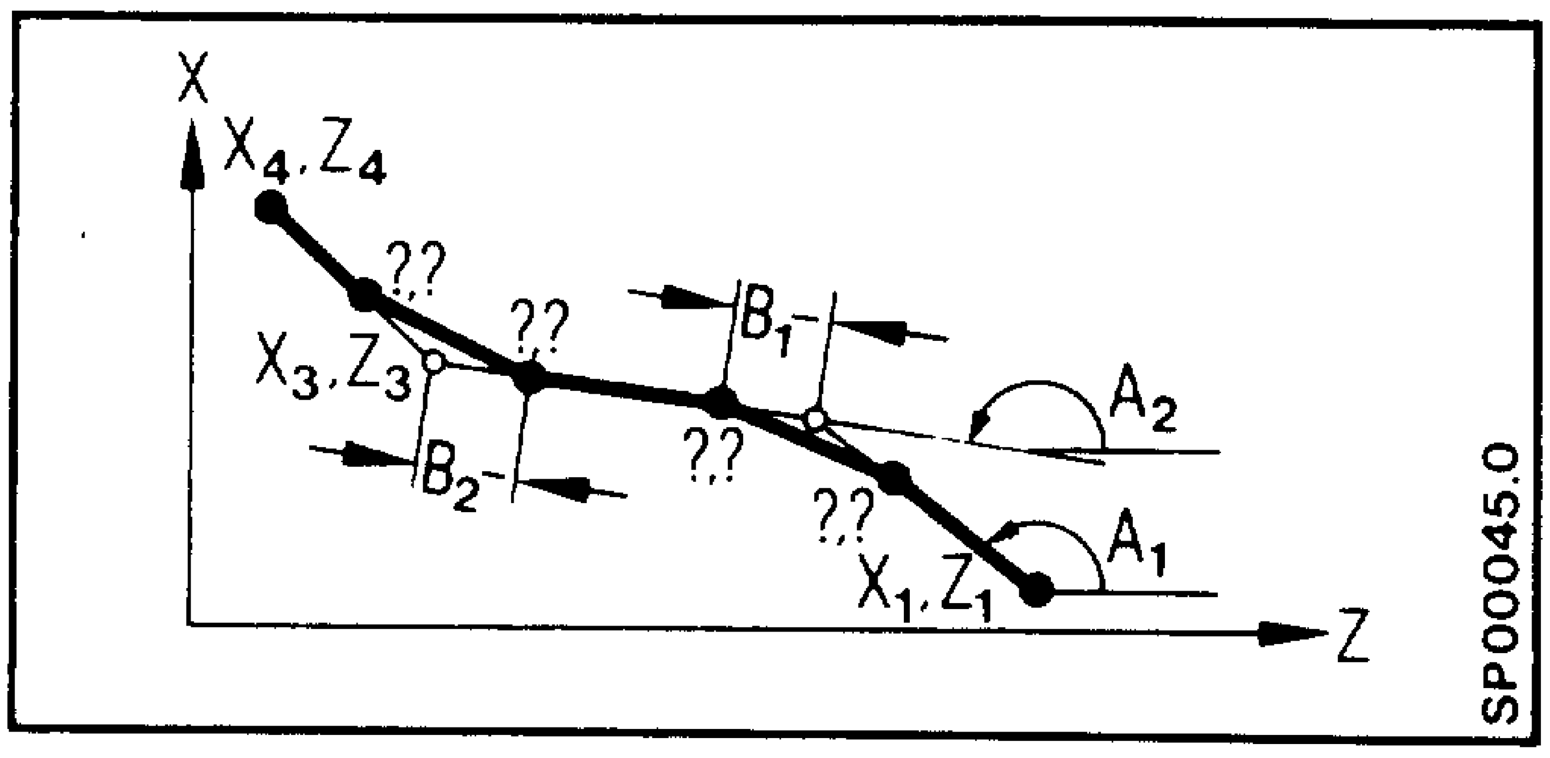
(3) + (4) 3-point-definition + chamfer N15 A_1...A_2..X_3..Z_3..B-..



(3) + (5) 3-point definition + radius N15 A_1..A_2..X_3..Z_3..B...

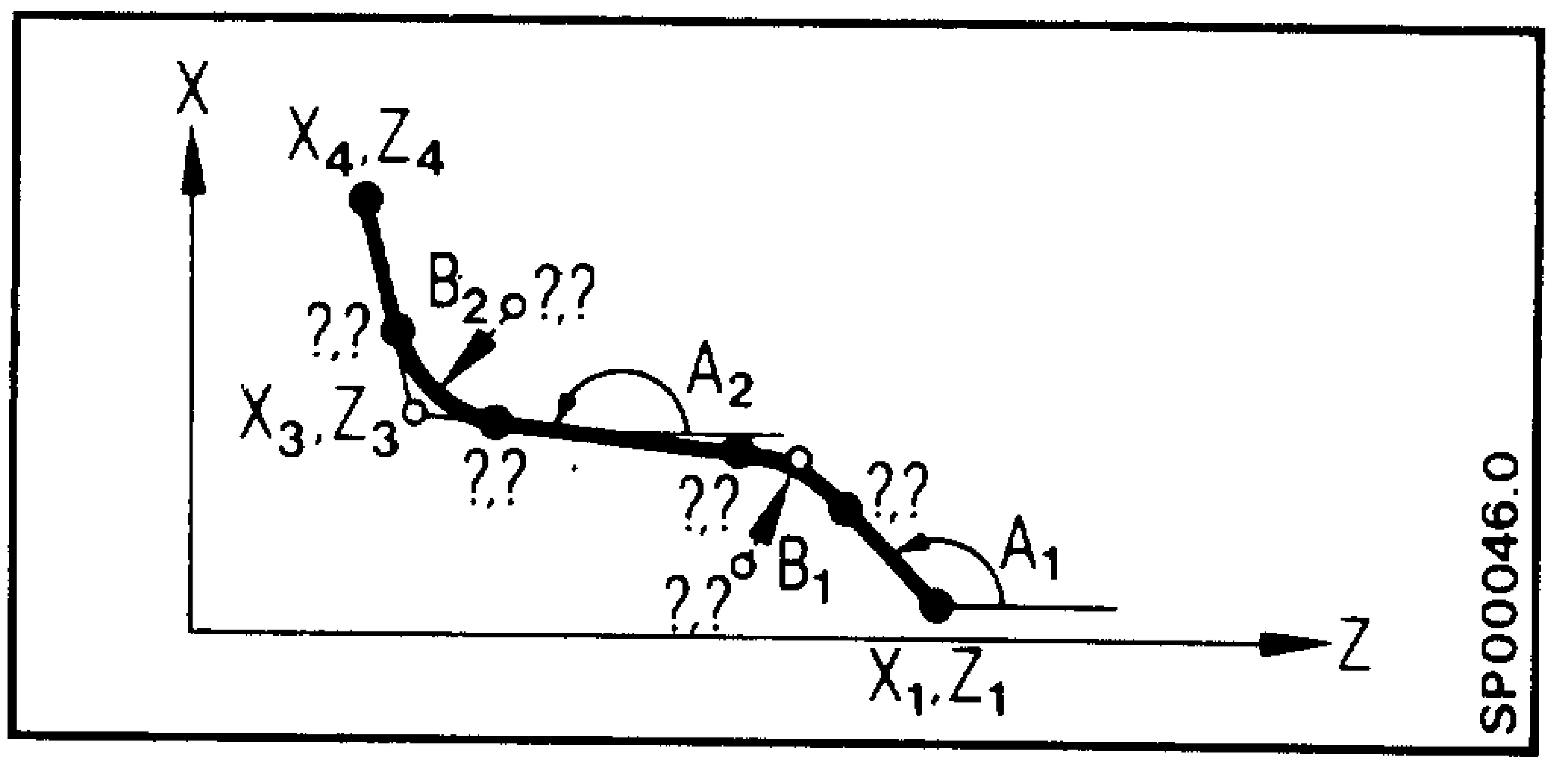


(3) + (4) + (5) 3-point-definition + chamfer + chamfer



N15 A_1..A_2..X_3..B_1-..B_2-.
N16 X_4..Z_4..
Addition of second chamfer at the end point (X_3, Z_3).

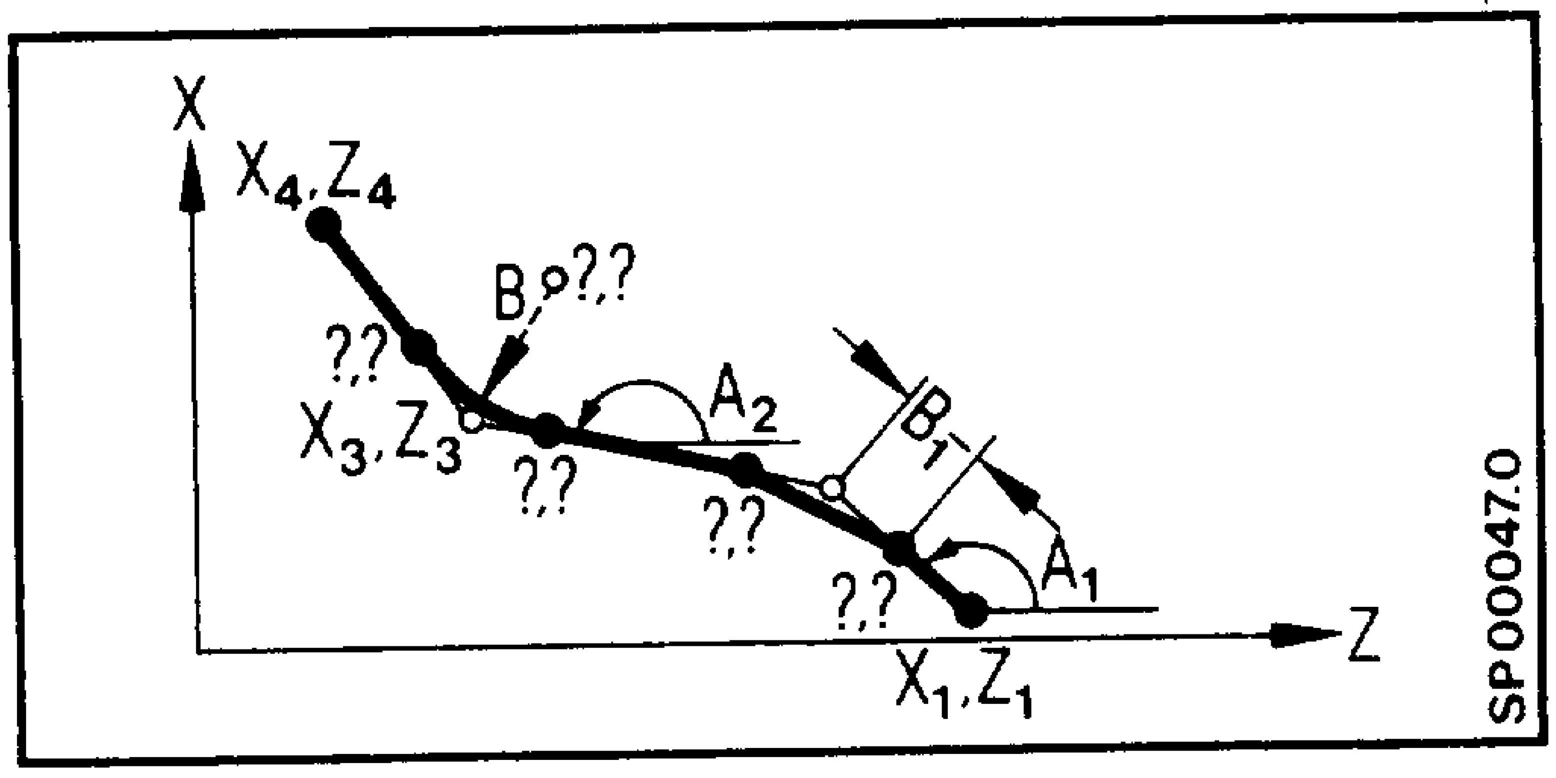
(3) + (5) + (3) 3-point-definition + radius + radius



N15 A_1..A_2..X_3..Z_3..B_1..
B_2.. N16 X_4..Z_4..
Addition of second radius at the end point X_3, Z_3.

* Second block can also be a blueprint programmed block.

(3) + (4) + (5) 3-point-definition + chamfer + radius

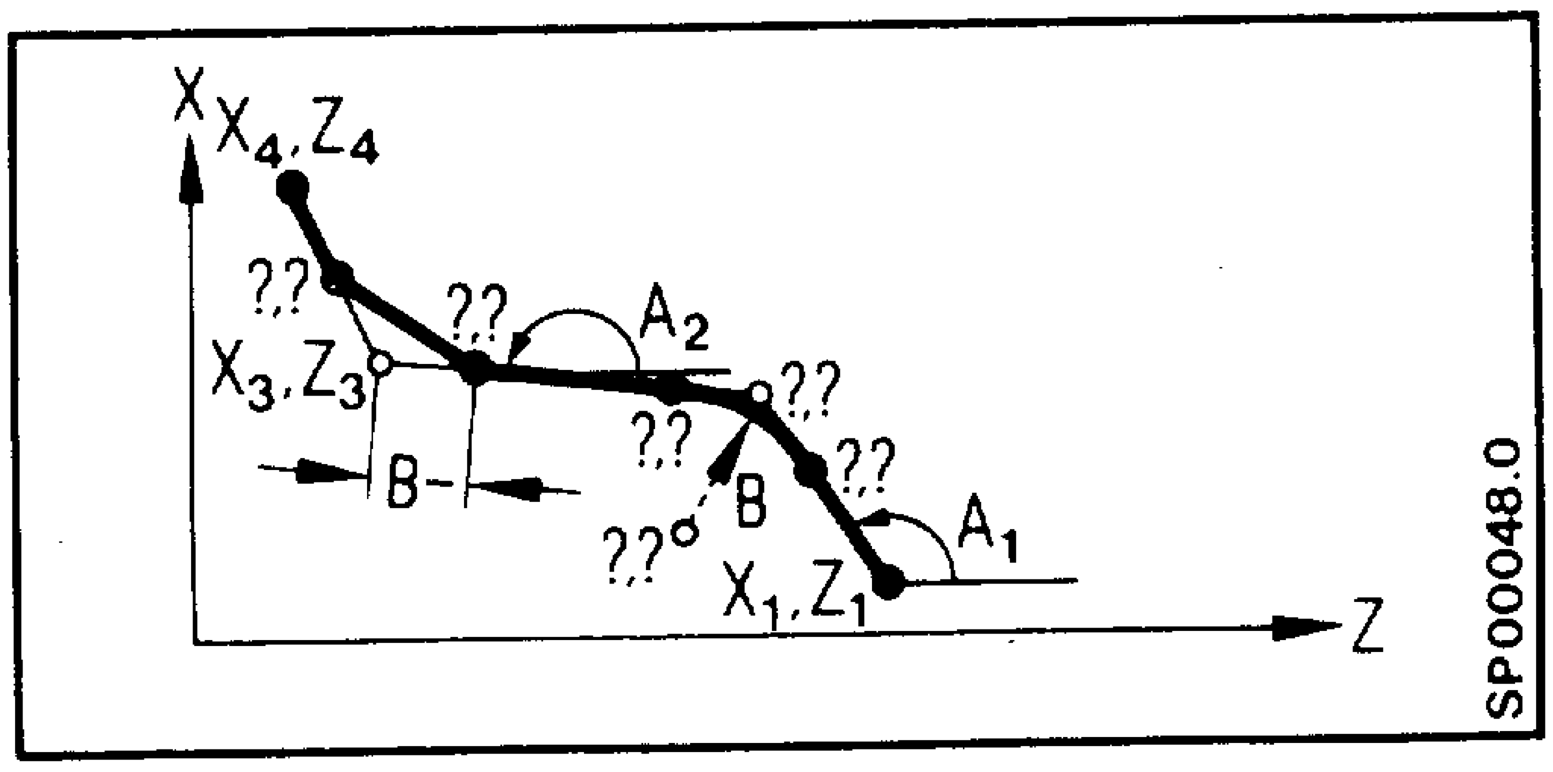


N15 A_1..A_2..X_3..Z_3.. B-..B+..

N16 X_4..Z_4..

Addition of a radius at the end point X_3, Z_3. The next block is automatically taken into consideration.

(3) + (5) + (4) 3-point-definition + radius + chamfer



N15 A_1..A_2..X_3..Z_3 B+..B-..

N16 X_4..Z_4..

Addition of a chamfer B- at the end point.

*Second block can also be a blueprint programmed block.

In the case of corners in which no chamfer or radius should be inserted, BO is to be programmed if a radius or chamfer follows in the contour section.

Note:

In this type of programming a block with a path = 0 is generated by the controller. This must be noted when the tool radius compensation is active.

B-0 is interpreted as B 0.

A radius or a chamfer can be inserted if the following block is not a circular block.

The sequence of the address A, X, Z, B. F etc. can be freely selected; however angles and radii must already be in the previously described sequence (first angle before second angle, first radius before second radius in the machining direction).

6.5 Operation of the G09 function , F, S, T, H, M in blueprint programming

If G09 is programmed in a block with contour definition, it is effective only at the end of the block, i.e. when the final position is reached.

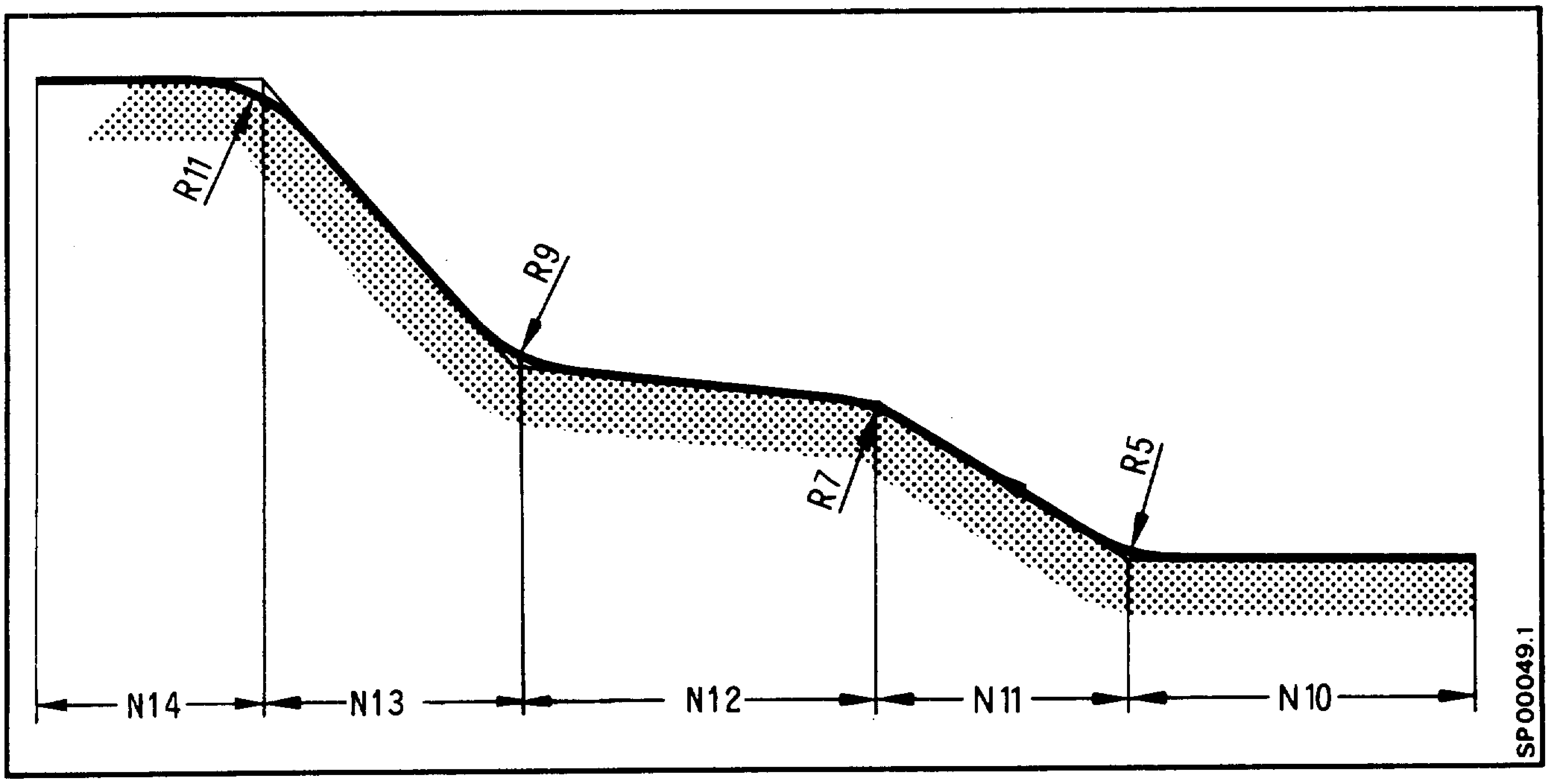
Within the block a G09 is generated automatically by the controller at transition points (edges, corners).
If F, S, T, H, M are programmed in a block with contour definition, they are effective at the beginning of the block.

If M00, M02, M17, M30 are programmed in a block with contour definition, these are effective at the end of the block.

6.6 Linking of blocks

The linking of blocks with and without angle inputs with inserted radii or chamfers is possible in any desired sequence.

Example:



N10 Z... B5 LF

N11 A... X... B7. LF

N12 A... A... X... Z... B9. B11. LF

N13 Z... LF

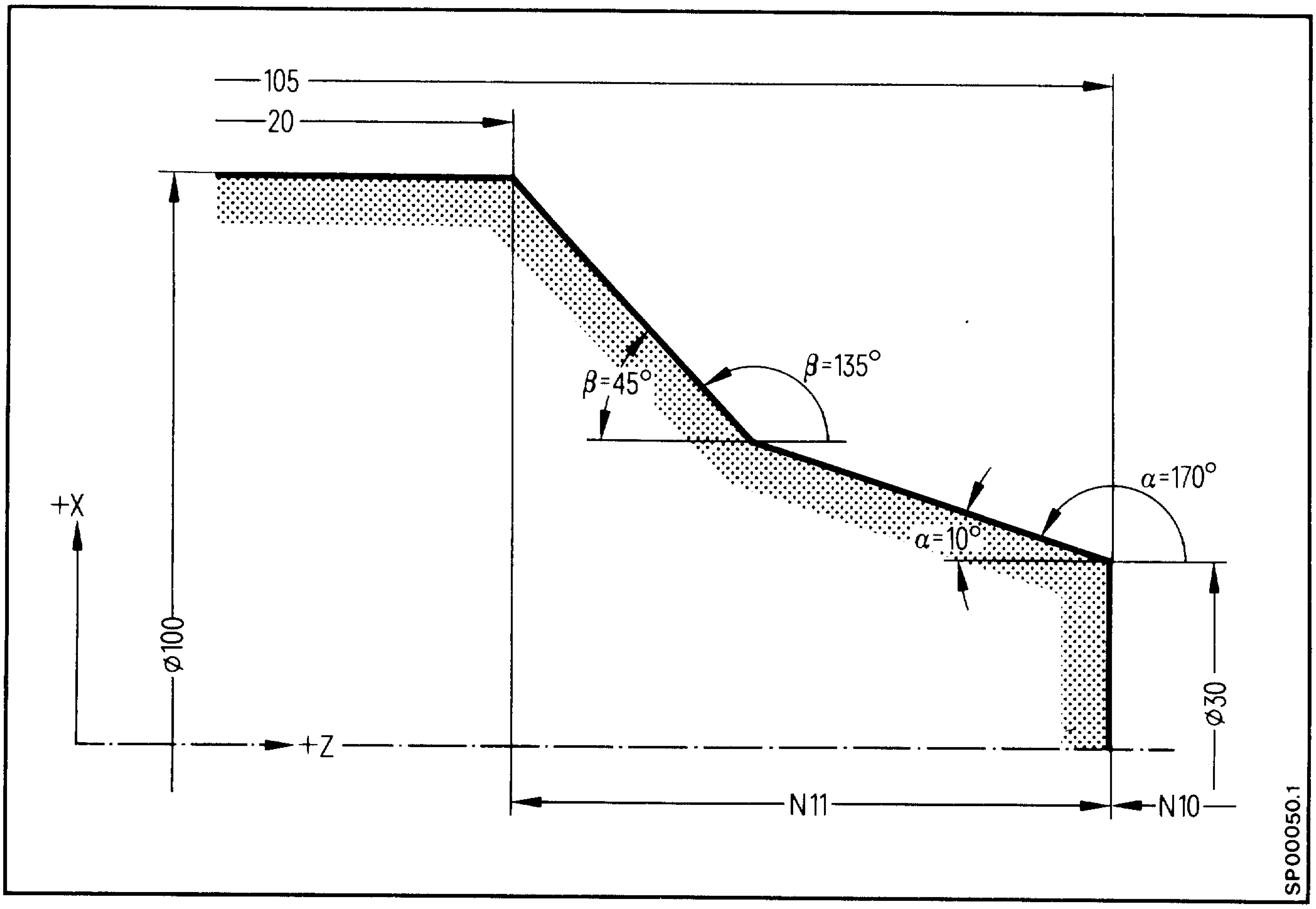
6.7 Examples

The angle X refers to the starting point, the angle ≙ to the missing restart point.

The end point can be programmed in the absolute dimensions G90 or in the incremental dimensions. Both end point coordinates are to be given.

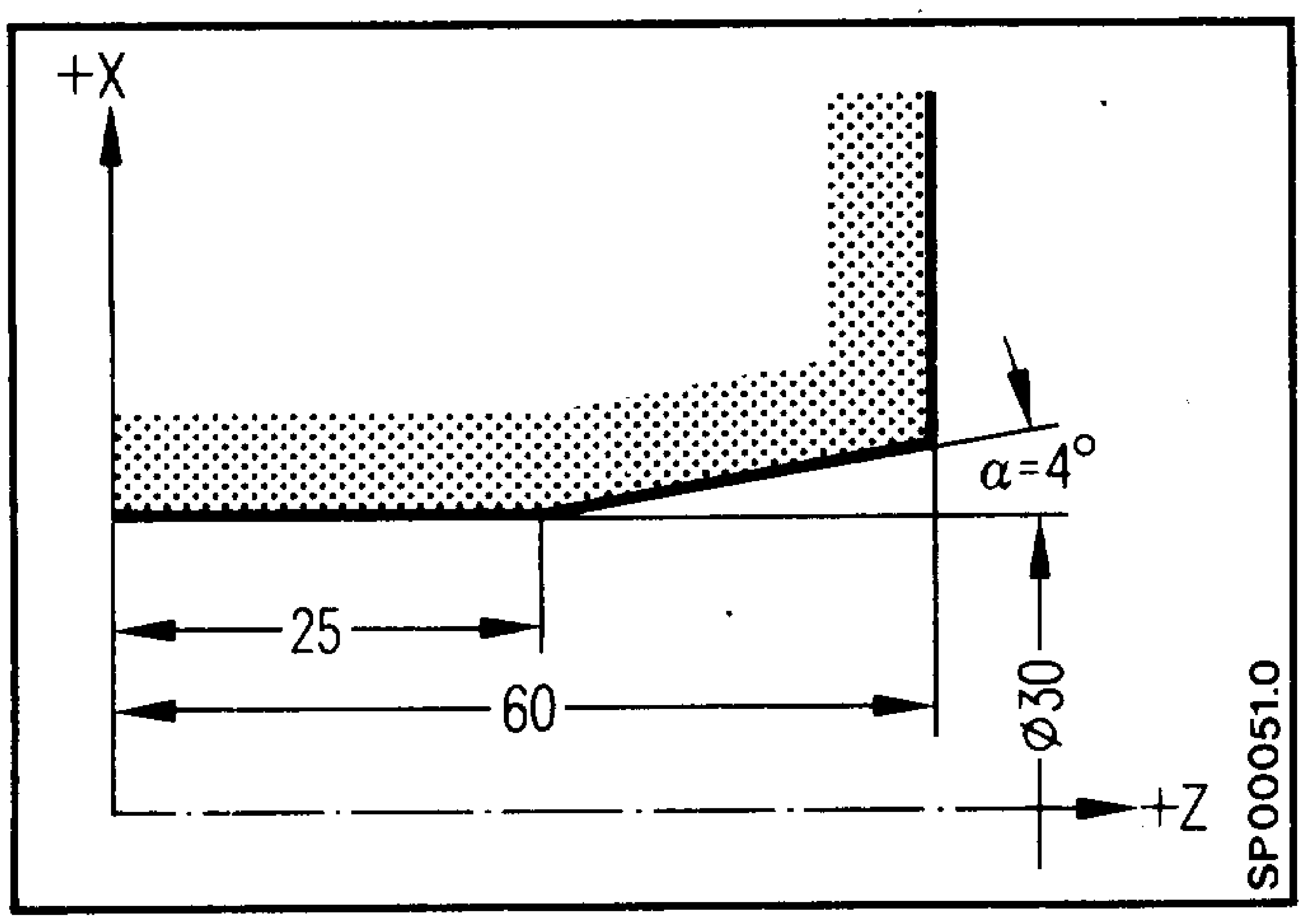
The control determines the restart point from the known beginning point, both angles and the end point.

Example: External Machining



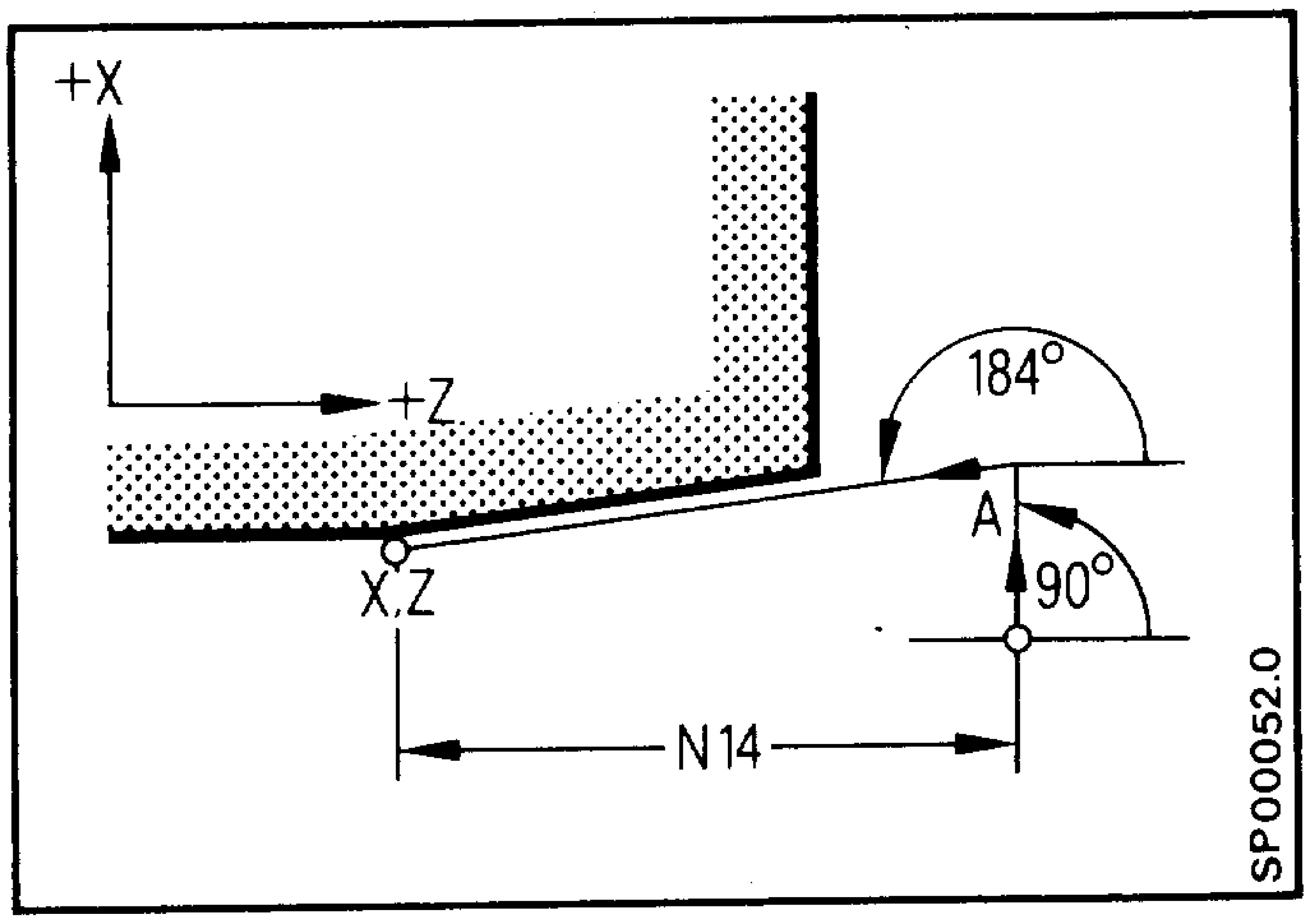

```
N10 G00 G90 X30. Z105. LF
N11 G01 A170. A135. X100. Z20. F... LF
```

Example: Internal machining



Drawing Dimensioning

The start point is determined freely outside the inner cone.

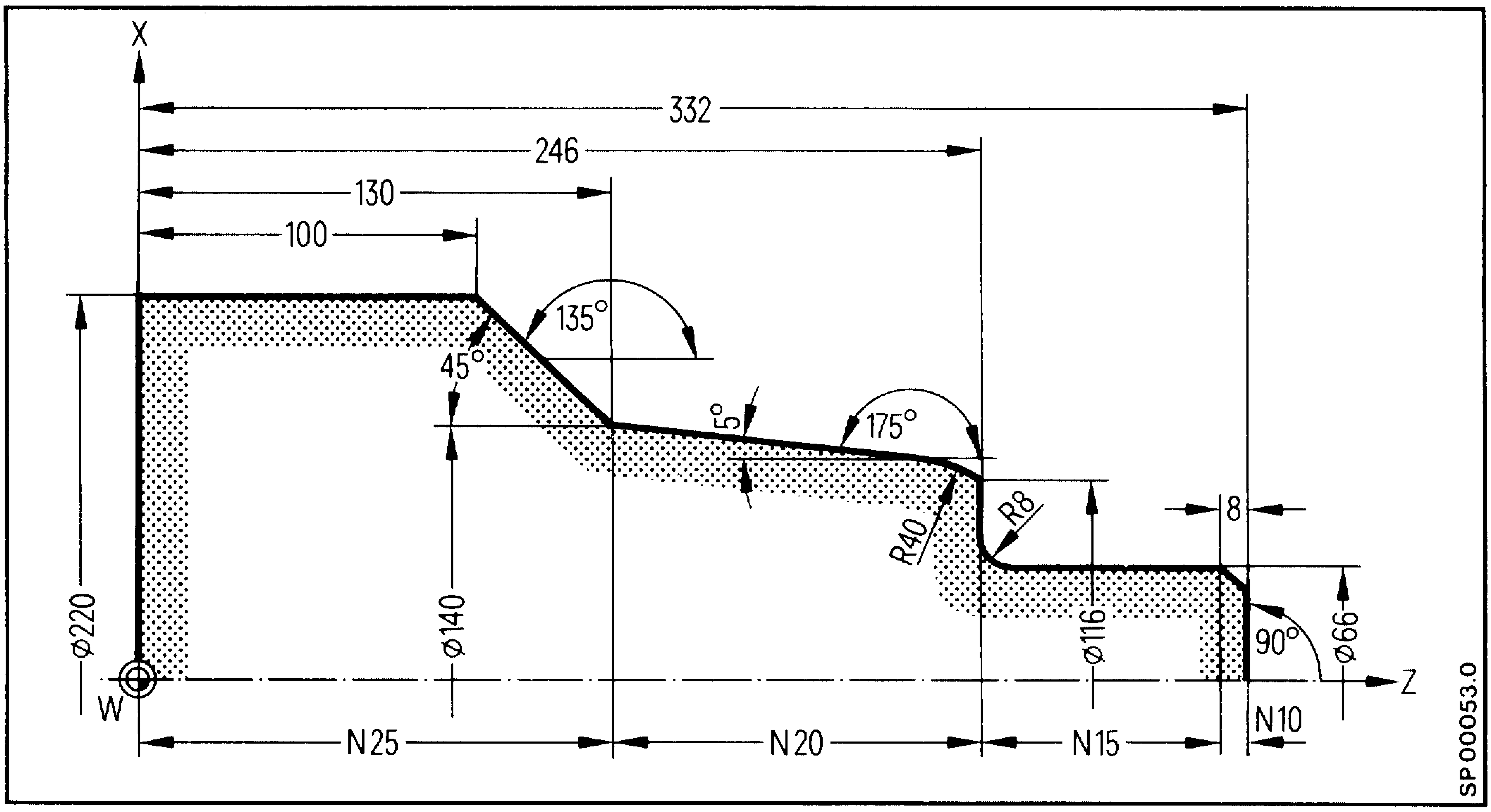


The vertical through the start point and the extension of the inner cone gives the intersection A.

The program is then as follows:

```
.
.
N13 G00 Xstart Zstart LF
N14 G01 A90. A184. X... Z... LF
```

Example: Programming with contour definition



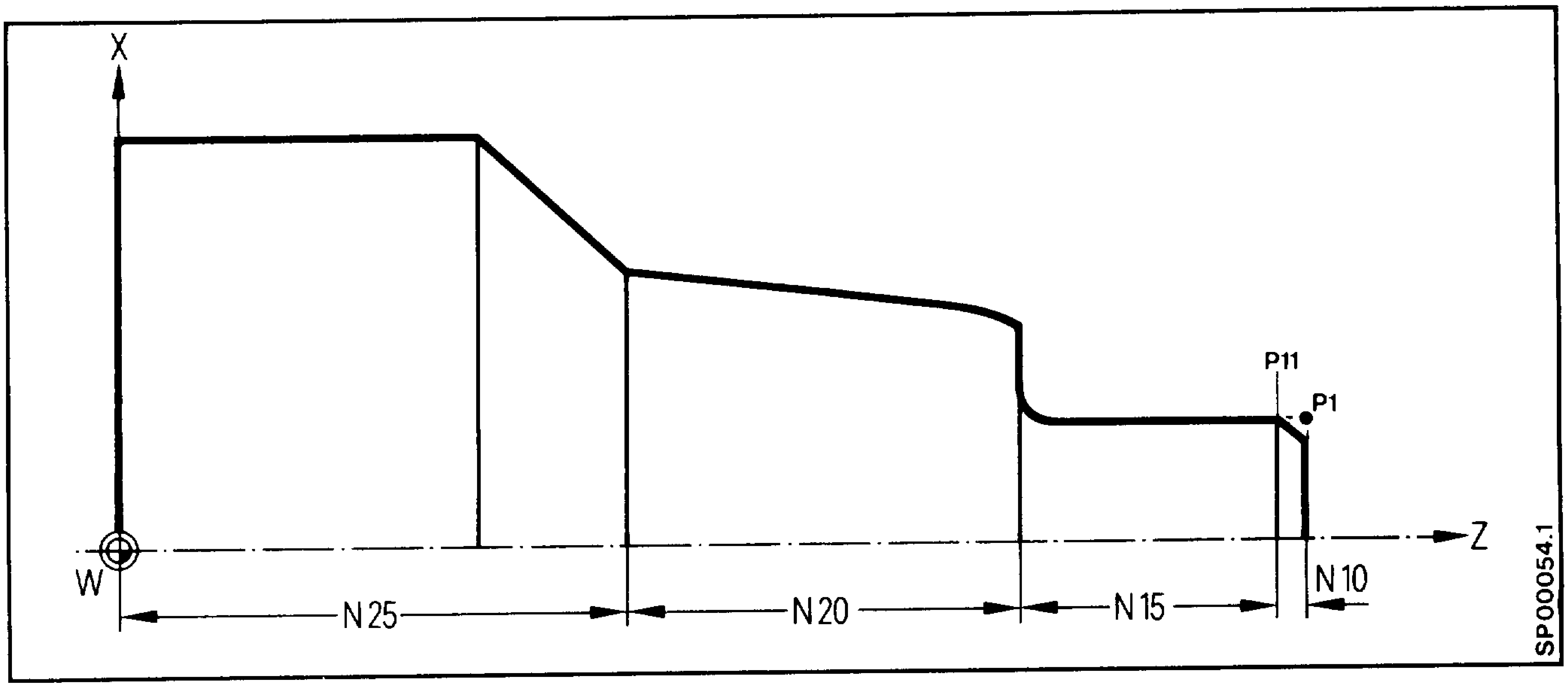
Blueprint programming

```
L10500
N5  G00 G90 X0. Z332. LF          Linking through B-8.
N10 G01 G09 A90. X66. B-8. F0.2  LF
N15 A180. A90. X116. Z246. B8.   LF
N20 G03 B40. A175. X140. Z130. LF
N25 G01 A135. A180. X220. Z0. LF
N30 M17 LF
```

6.8 Switching and auxiliary functions in concatenated blocks

Concatenated blocks are always present when blocks are combined with radii or chamfers.

Example:



For programming, see 6.14.

A block with switching and auxiliary functions can stand between concatenated blocks.

Example: see above and page 6-10.

N10 G01 G09 A90. X66. B-8.LF (P)

N101 M... H... ... LF

N15 A180. A90. X116. Z246. B8. LF

The switching and auxiliary functions are effective in point P11 (see above). Undercutting therefore takes place in point P11. The programmed F value in block N10 becomes effective at the beginning of the block N10 (p6 - 14).

7. Cycles

7.1 Machining cycles

7.1.1 L94 Stock removal cycle (paraxial roughing)

R10 Start point of the contour in X

R11 Start point of the contour in Z

R12 End point of the contour in X (absolute)

R13 End point of the contour in Z (absolute)

R14 Point B in X

R15 Point B in Z

R16 Depth of finishing cut in X (incremental)

R17 Depth of finishing cut in Z (incremental)

R18 Roughing depth in X or Z (incremental)

R20 Form determination for roughing

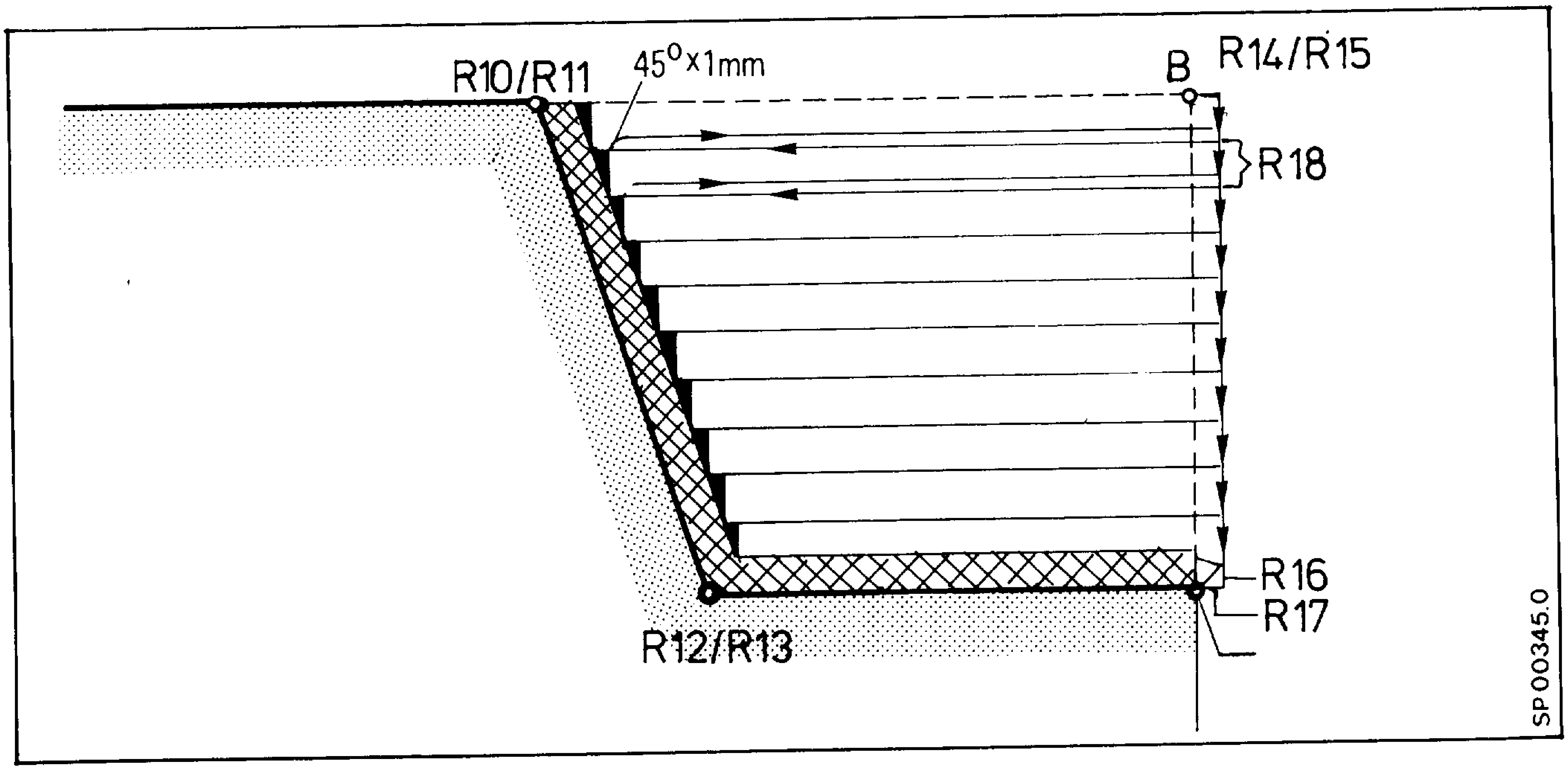
R21 Radius programming = 1, diameter programming = 2

*R22 Block type (0 = G00, 1 = G01, 2 = G02, 3 = G03)

*R23 Interpolation parameter I

*R24 Interpolation parameter K

The parameters to be entered are shown in the diagram below.



*Only with basic version 4 and SINUMERIK 3TT with software version 02 onwards

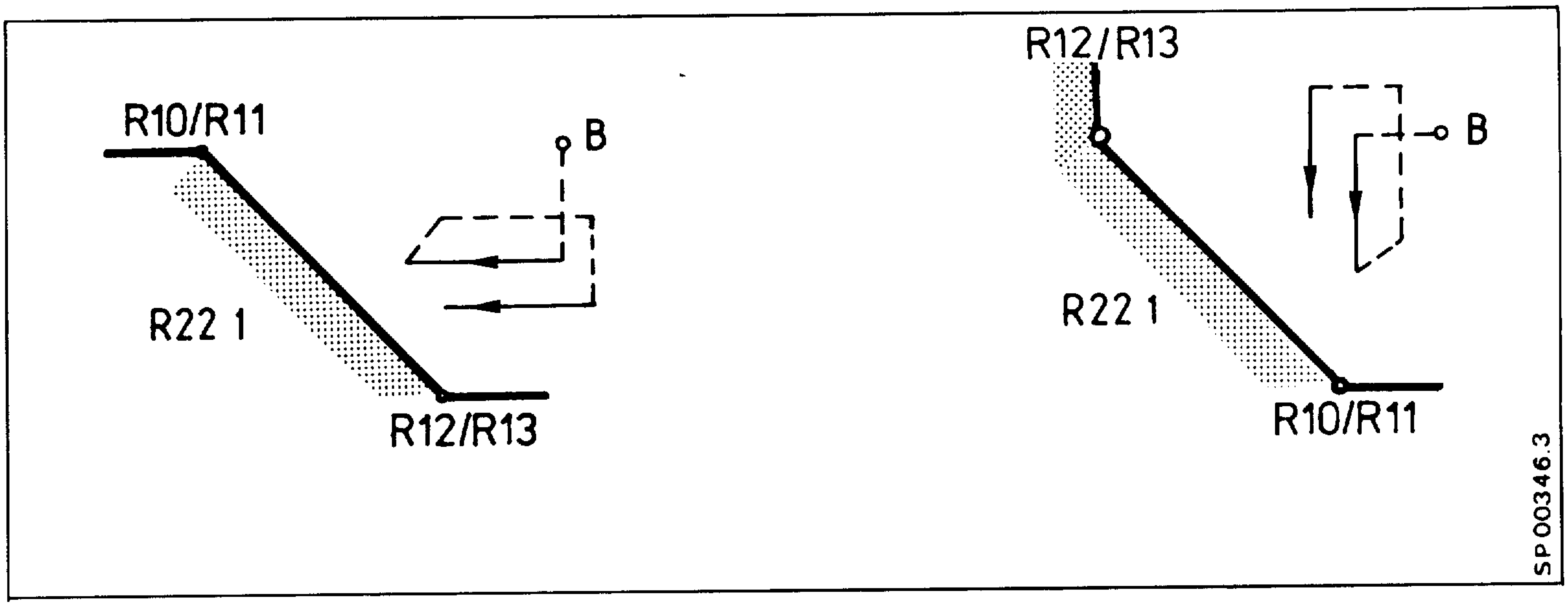
In the case of graphics with operator prompting, R23 and R24 are also stored during the cutting of a straight line.

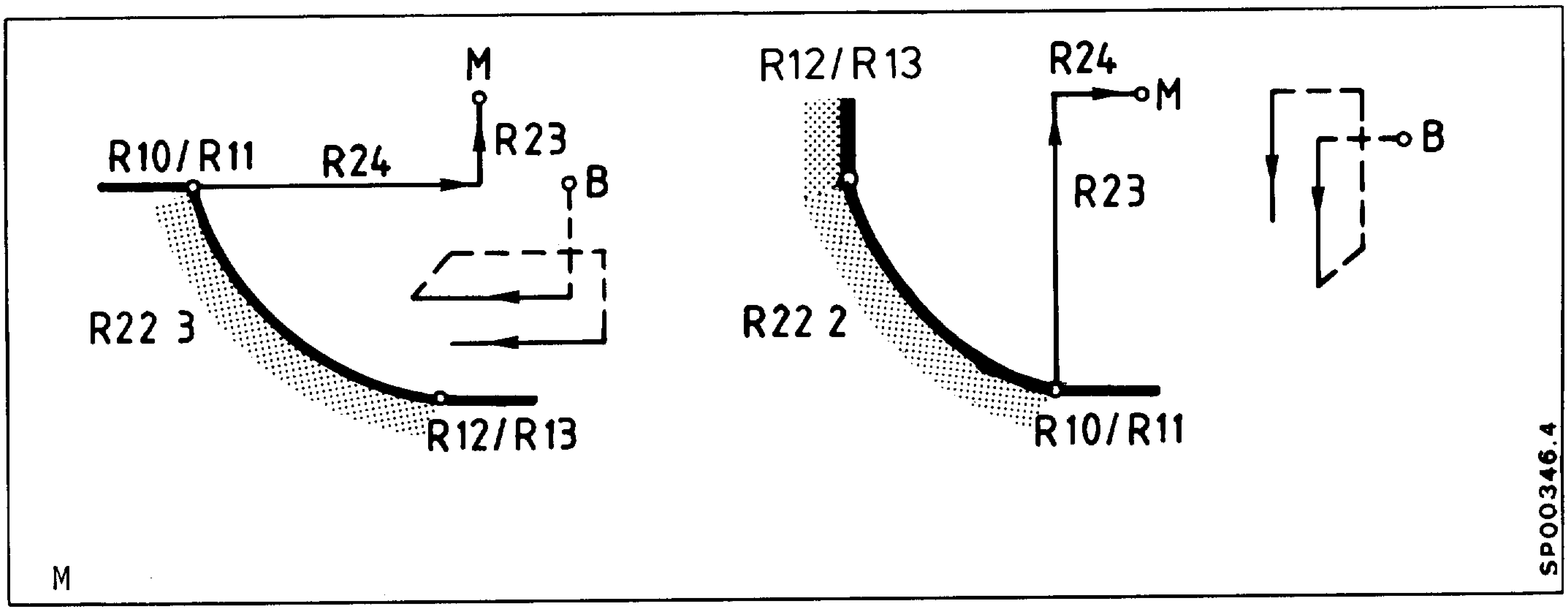
The machining cycle L94 can be called up from any collision-free position.

The R parameters previously referred to are modal, i.e., if several roughing cycles are called consecutively only the parameters R10-R15 (block coordinates) must be newly programmed. The finishing depth for consecutive cycles may not be changed, whereas the roughing depth may be varied in each cycle.

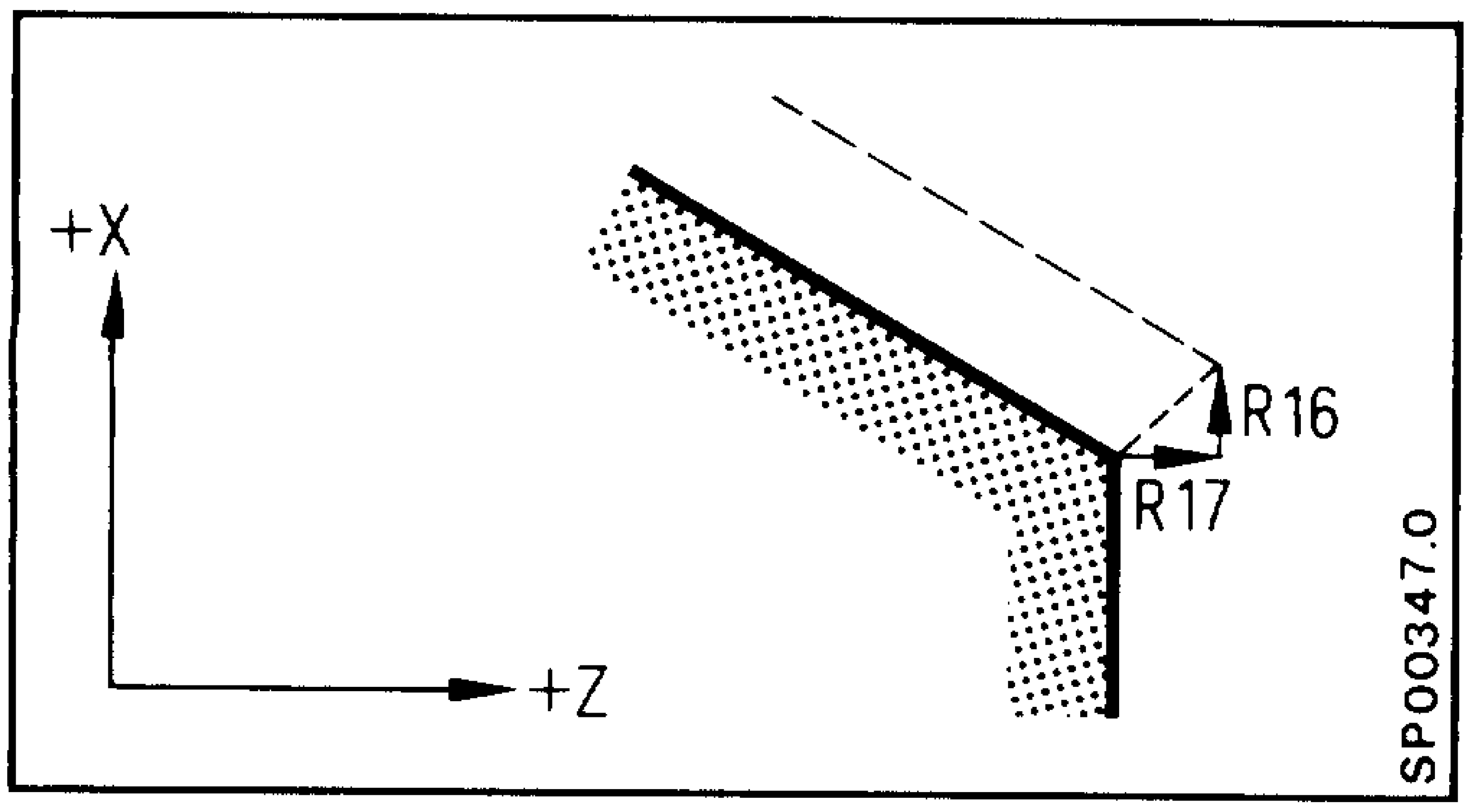
Coordinate inputs R10 to R15, R22, R23, R24

Stock removal is only possible in individual blocks, i.e. the contour to be machined must be subdivided. The machining cycles are then called one after the other. The start and end points of the blocks must be programmed in the direction of infeed.





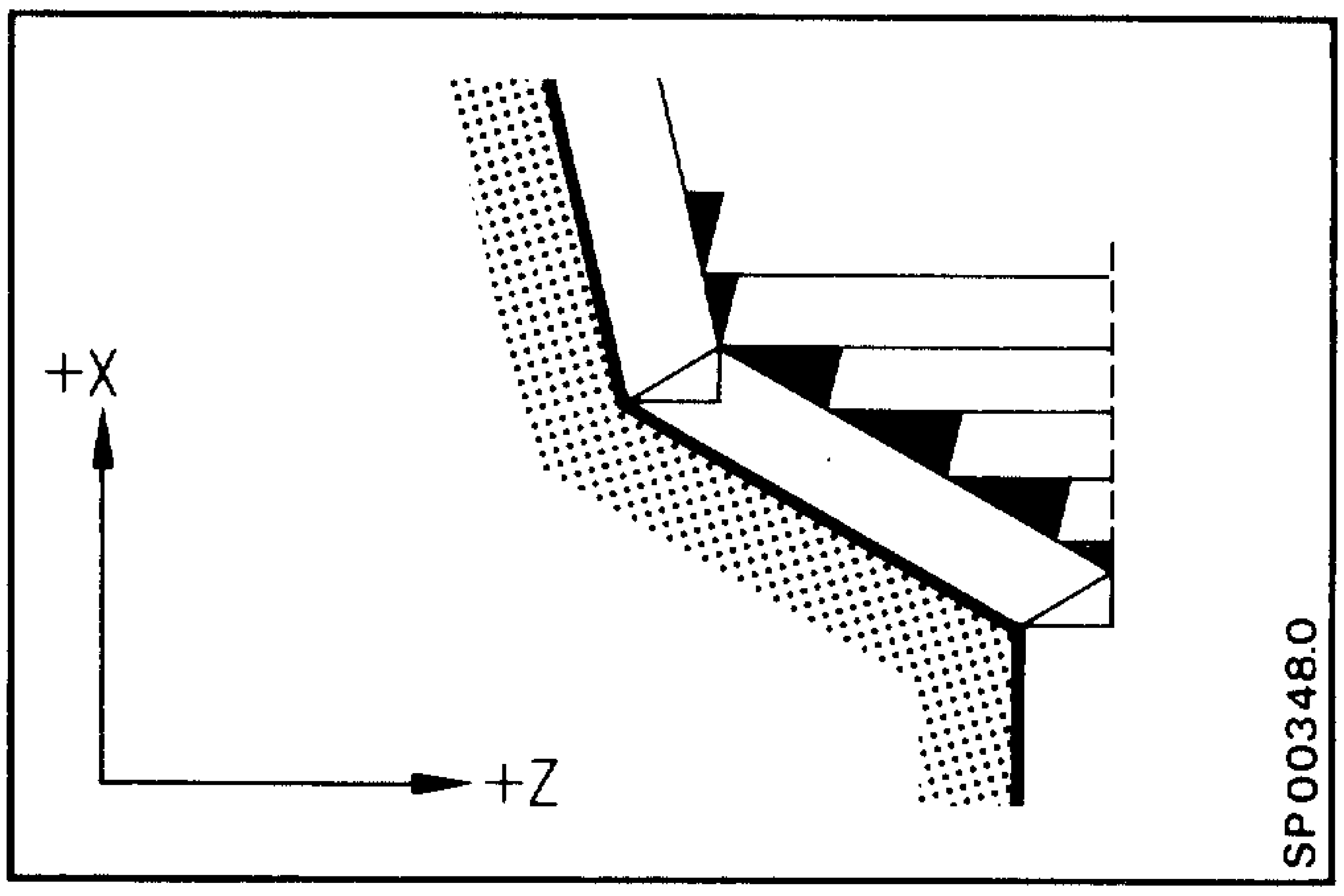
The contour-parallel stock removal of the remaining corners is carried out in opposite direction from the block end point to the block start point. In addition to block start and block end, type of block and interpolation parameters with the circle elements also point B must be programmed which may be shifted in paraxial direction to the block end point. Point B is the return point after each machining cycle.

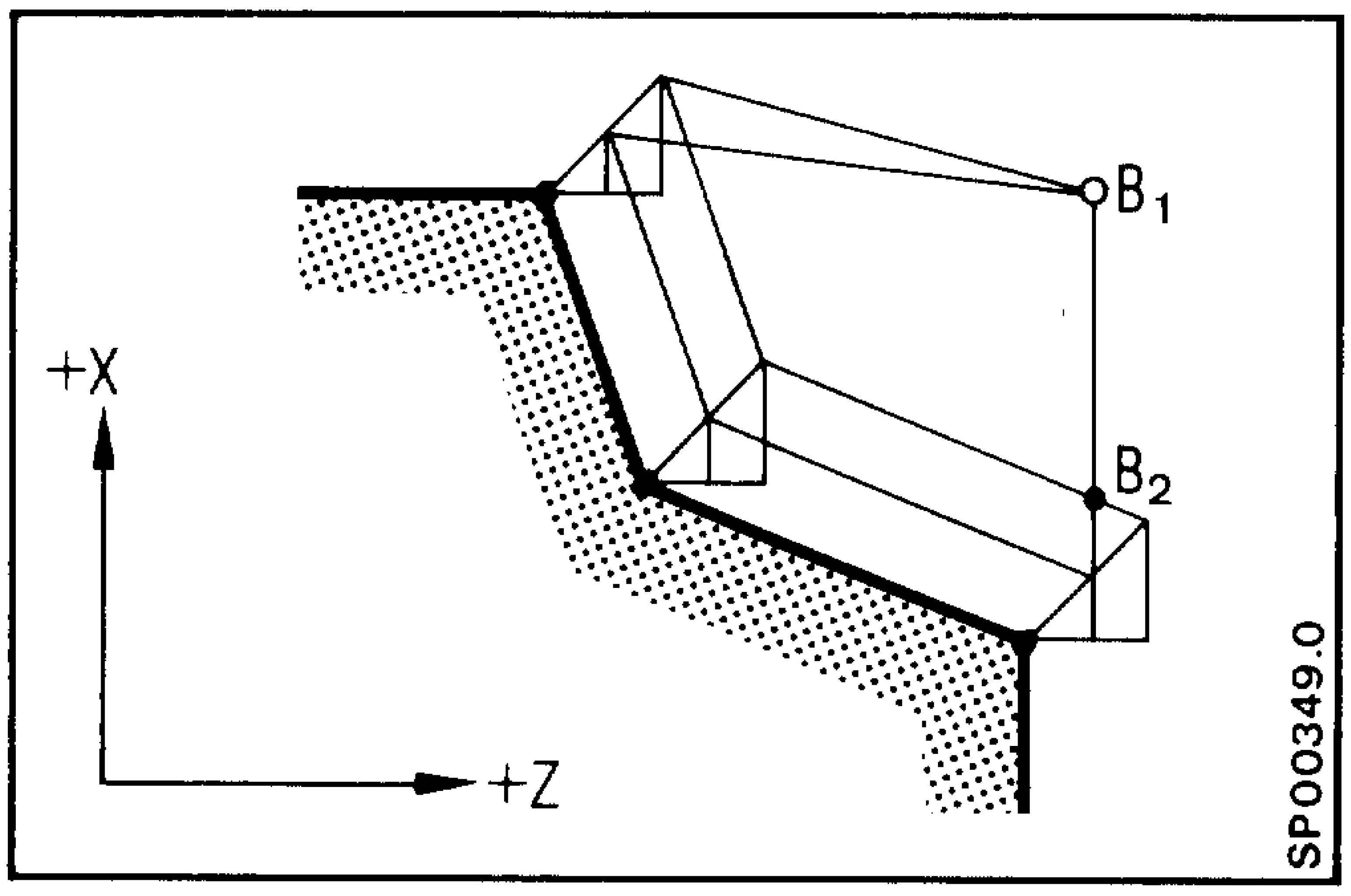


The contour is displaced by the depth of finishing cut programmed using R16 and R17.

e.g., R16 0.3
R17 0.3

During the roughing cycle, rough machining occurs down to this depth.





Finishing cut depth R16 0, R17 0 corresponds to the final contour.

Using several finishing cuts it is possible, to turn down for example, the corners left after the rough machining with the aid of a copy tool, in the event that the cycles R20 31 to R20 34 are not utilized.

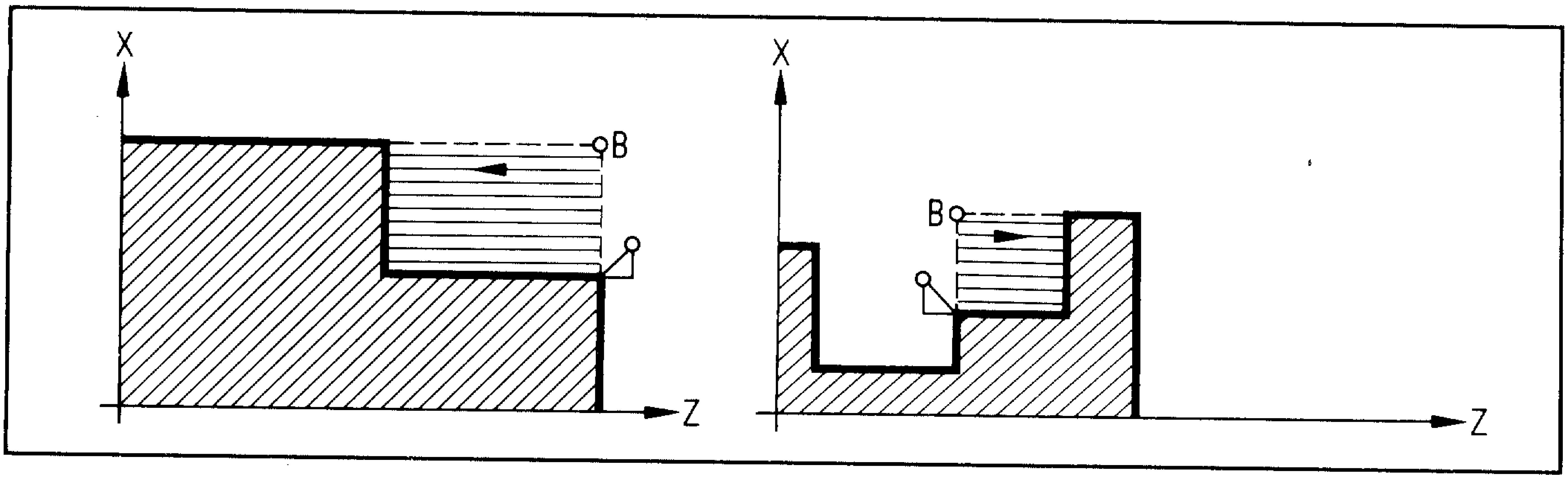
Type of machining (R20)

The parameter R20 describes the type of stock removal; roughing paraxial or parallel to the contour, whether inside or outside contour machining, longitudinal or facing.

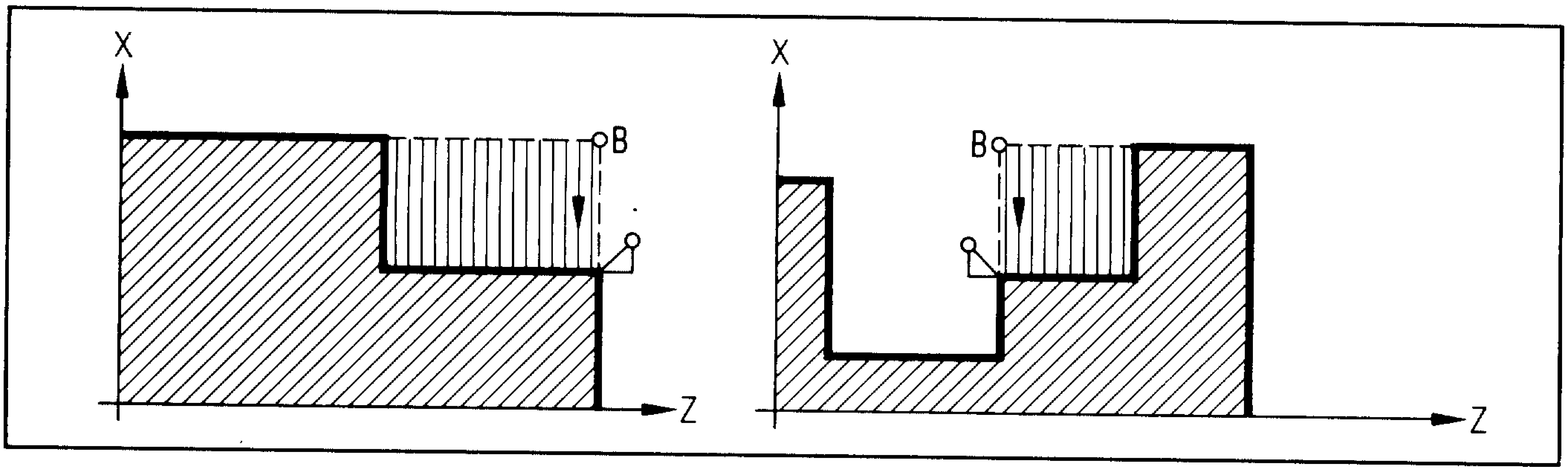
R20	11	longitudinal	(Z)	outside	Roughing, paraxial
R20	12	facing	(X)	outside	
R20	13	longitudinal	(Z)	inside	
R20	14	facing	(X)	inside	
R20	21	longitudinal		outside	1st roughing cut parallel to the contour down to finish cut depth
R20	22	facing			
R20	23	longitudinal		inside	
R20	24	facing			
R20	31	longitudinal	(Z)	outside	Paraxial roughing with one final cut parallel to the contour down to finish cut depth
R20	32	facing	(X)	outside	
R20	33	longitudinal	(Z)	inside	
R20	34	facing	(X)	inside	

Roughing paraxial or parallel to contour is always executed without cutter radius compensation. With cutter shape 9 (P=S), the finishing cut must be increased by the cutter radius.

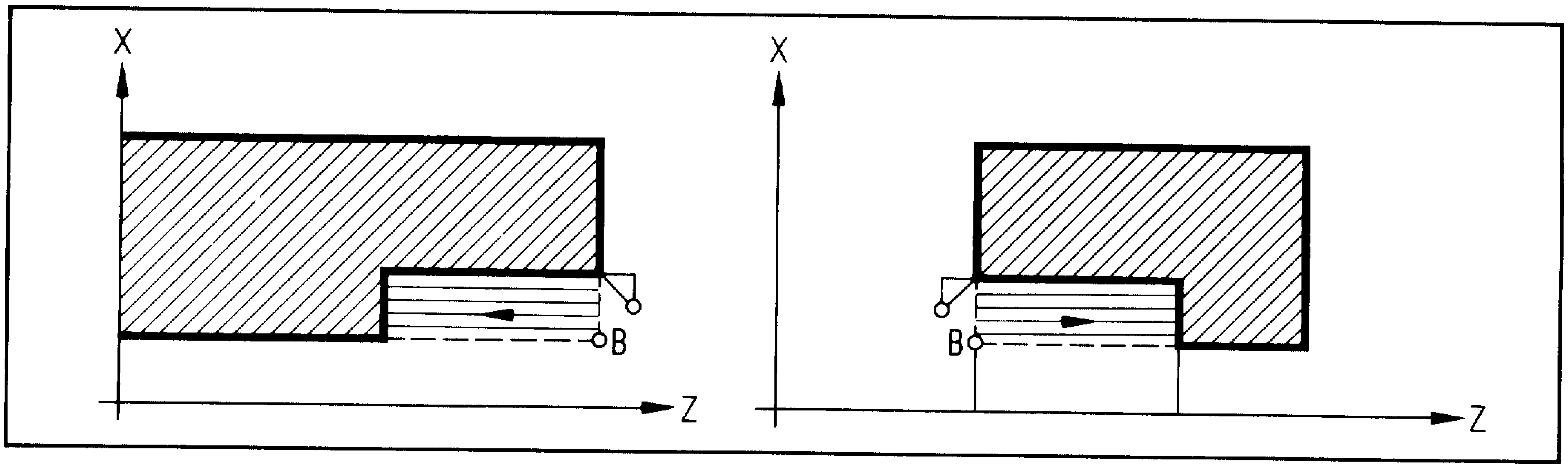
Example:



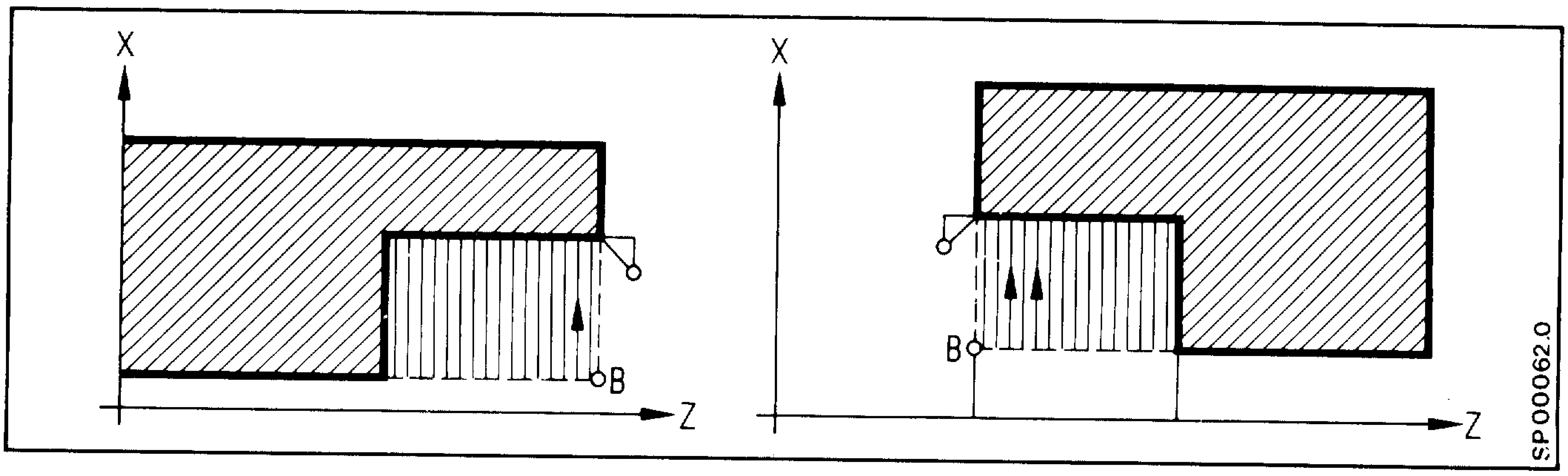
R20 11 External roughing, longitudinal (Z)



R20 12 External roughing, facing (X)



R20 13 Internal roughing, longitudinal (Z)



R20 14 - Internal roughing, facing (X)

Programming in X direction (R21)

Parameter R21 must be set to 1 for radius programming and to 2 for diameter programming.

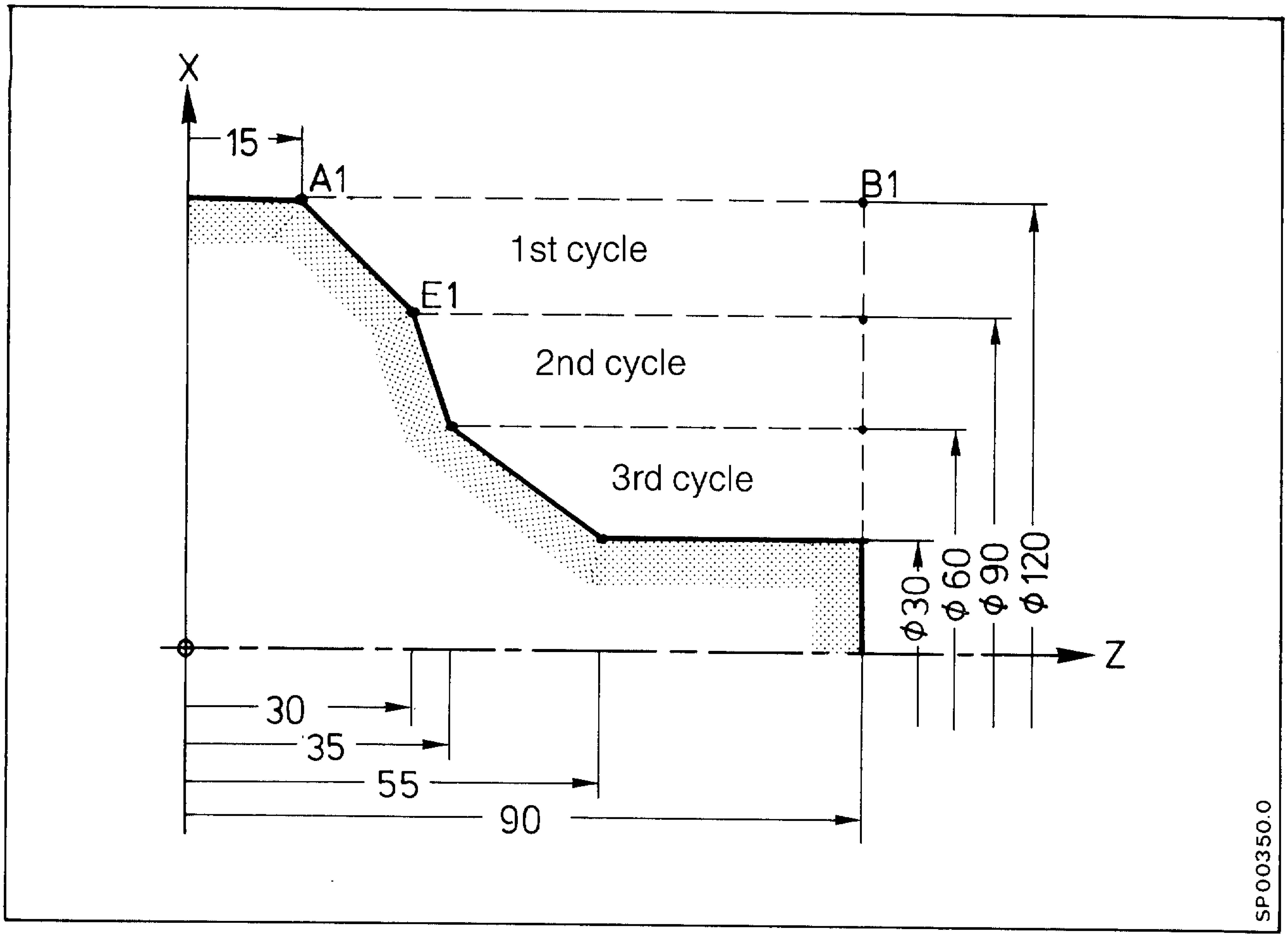
Stock removal at constant cutting speed

If the machining is to be executed with constant cutting speed, the "Constant cutting speed G96" function must be selected before the cycle is called.

e.g.

```
N.. G96  S180  LF
N.. R10 .. R1 .. R21 .. LF
N.. L94  F.. LF
```

Example:



Call in part program

```
N19 ... R000 R01 1                 LF        Auxiliary parameter

N20 R16 0.3  R17 0.3                         Finishing depth
    R18 5                                    Depth of cut
    R20 11                                   Roughing
    R21 2                          LF        Diameter programming

N21 R10 120 R11 15 R12 90 R13 30             Coordinates for the
    R14 120 R15 90 L94             LF        1st part block

N22 R10 90 R11 30 R12 60 R13 35
    R14 90 L94                     LF        2nd cycle

N23 R10 60 R11 35 R12 30 R13 55
    R14 60 L94                     LF        3rd cycle
```

		Stock removal of the remaining
N24 R20 21 R00 R01	LF	corners
N25 @01-21 R00 R01	LF	repeat the entire contour
N26 G01 G09 G42 X30 Z95	LF	
N27 G09 Z55	LF	pass along end contour
N28 G09 X60 Z35	LF	with CRC
N29 G09 X90 Z30	LF	
N30 G09 X120 Z15	LF	

.
.
.

or execute in stock removal type R 20 = 31 block N 20 to N23 and continue from N26 passing along end contour

7.1.2 L96 Thread cutting cycle

This cycle is used for longitudinal cutting of outside threads, inside threads and taper threads.
The tool infeed is automatic.
The infeed depth per cut decreases linearly.
The infeed is carried out at an angle of 90°.

Before calling up cycle L96 a value must be assigned to the following R parameters:

*R10	Start point of thread in X	absolute incl. approach path
*R11	Start point of thread in Z	
R12	End point of thread in X	absolute incl. run-out path
R13	End point of thread in Z	
R14	Thread pitch	
R15	Thread depth (incremental, sign required to define inside or outside thread, + = inside thread, - = outside thread)	
*R16	Finishing cut depth (incremental)	
*R17	Number of roughing cuts	
*R18	Number of idle passes	
R19	Radius programming = 1, diameter programming = 2	

* The parameters are changed by the cycle.

The individual parameter values are represented in the following diagrams.

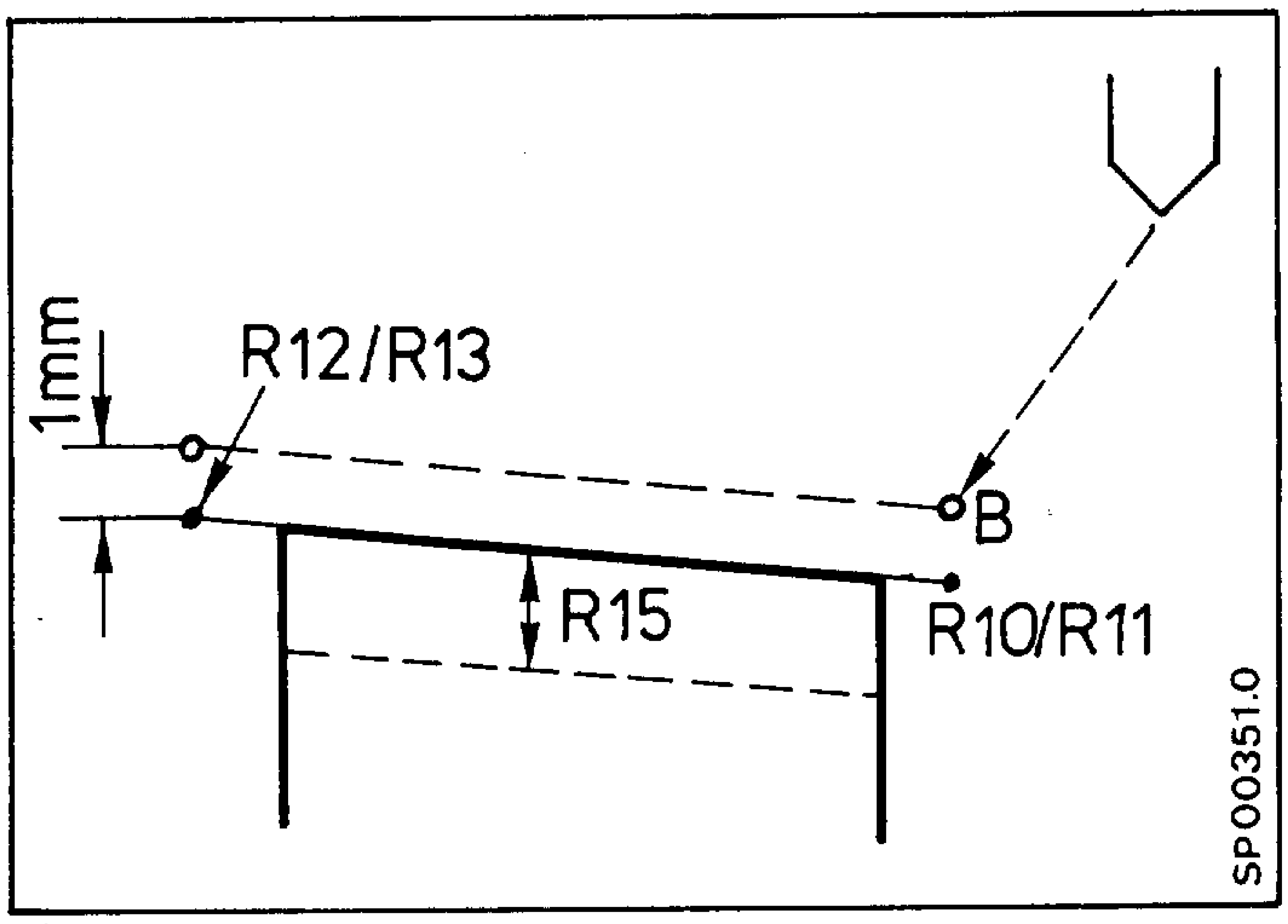
R14 Thread pitch

The parameter represents the value of the thread pitch. It is always written as a paraxial value without sign.

min. 0.001 mm

max.   400 mm

R10, R11 and R12, R13 Start point of thread, end point of thread



The coordinates of the start point or end point of thread must be programmed incl. approach path (start point) or run-out path (end point). In the X axis the starting point B is located 1 mm above the parameter value R10. This raised plane is generated automatically in the control. The thread cutting cycle can be called up independently of the tool position and infeed to point B is effected at rapid traverse rate.

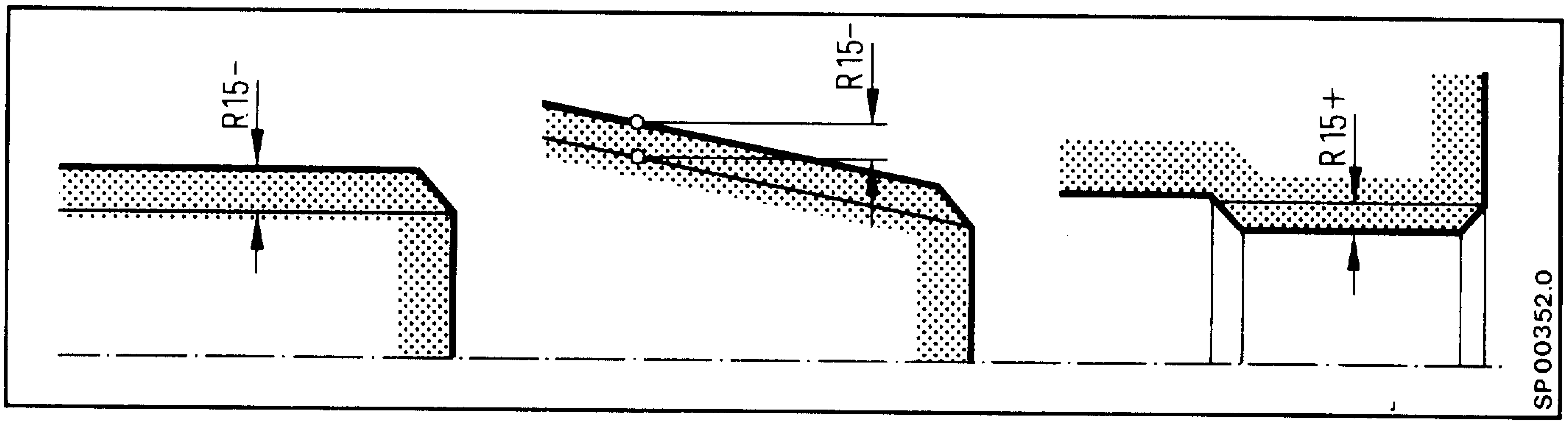
R18 Idle passes

Any number of idle passes can be selected. They are entered using parameter R18.

e.g., 3 idle passes R18 3.

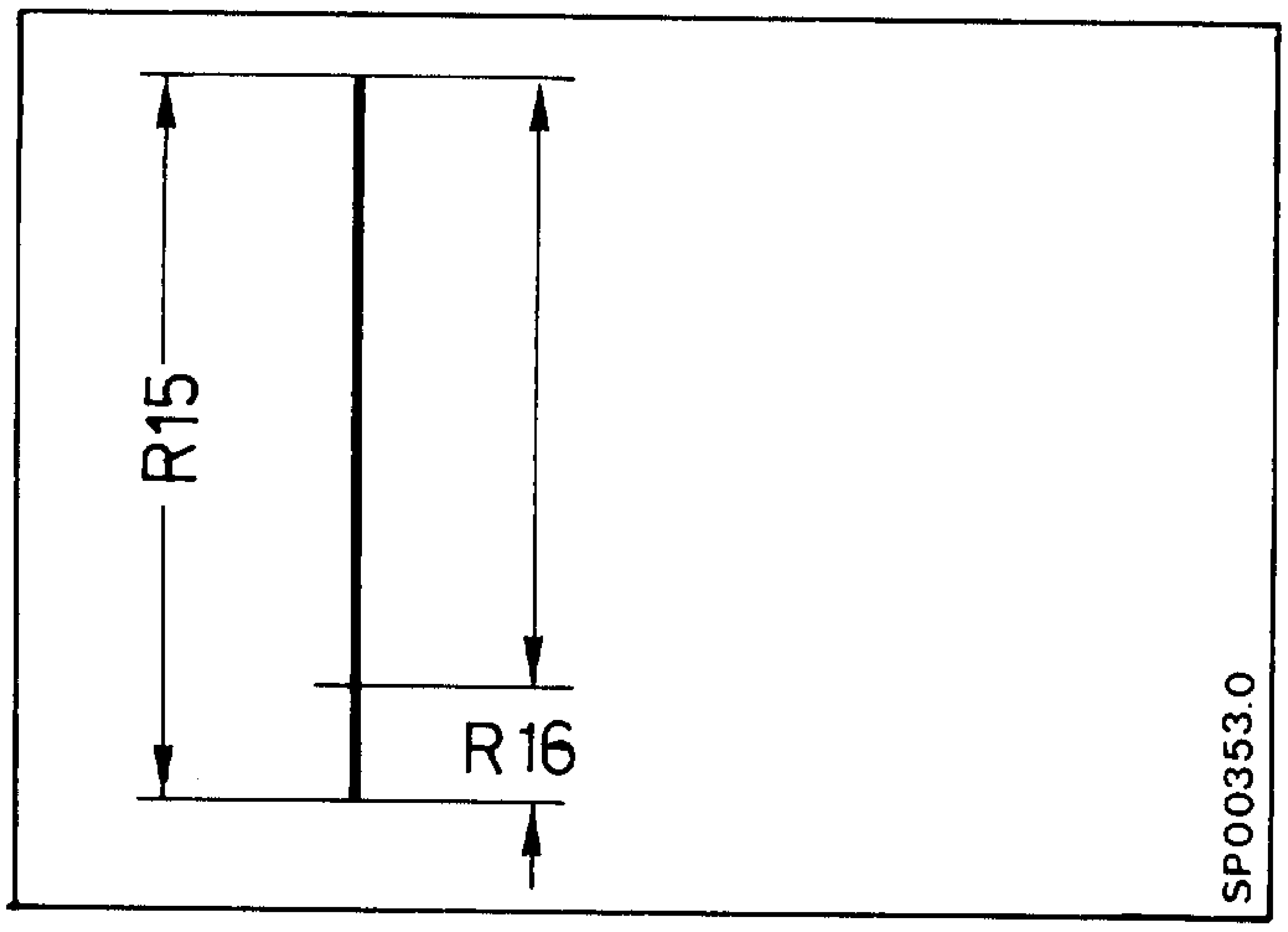
R15 Thread depth

The depth of thread is entered using parameter R15. The sign determines the infeed direction, i.e., whether it is an outside or inside thread (+ inside thread, - outside thread).



R16 Finishing cut depth

R16 gives the finishing cut depth. When a finishing cut depth is programmed, this depth is subtracted from the thread depth and the remaining value divided into roughing cuts. After the roughing cuts have been completed a finishing cut is made.



The roughing depth is automatically calculated and divided into roughing cuts.

Finish cut depth R16

R17 Number of roughing cuts

The number of thread roughing cuts is determined by the parameter value. The controller automatically calculates the individual infeed depth, decreasing it at each cut by a constant rate.

The depth of the first roughing cut is calculated as follows:

$$t_1 = \frac{2t}{n} = \frac{2\ (R15 - R16)}{(R17 + 1)}$$

and the constant difference:

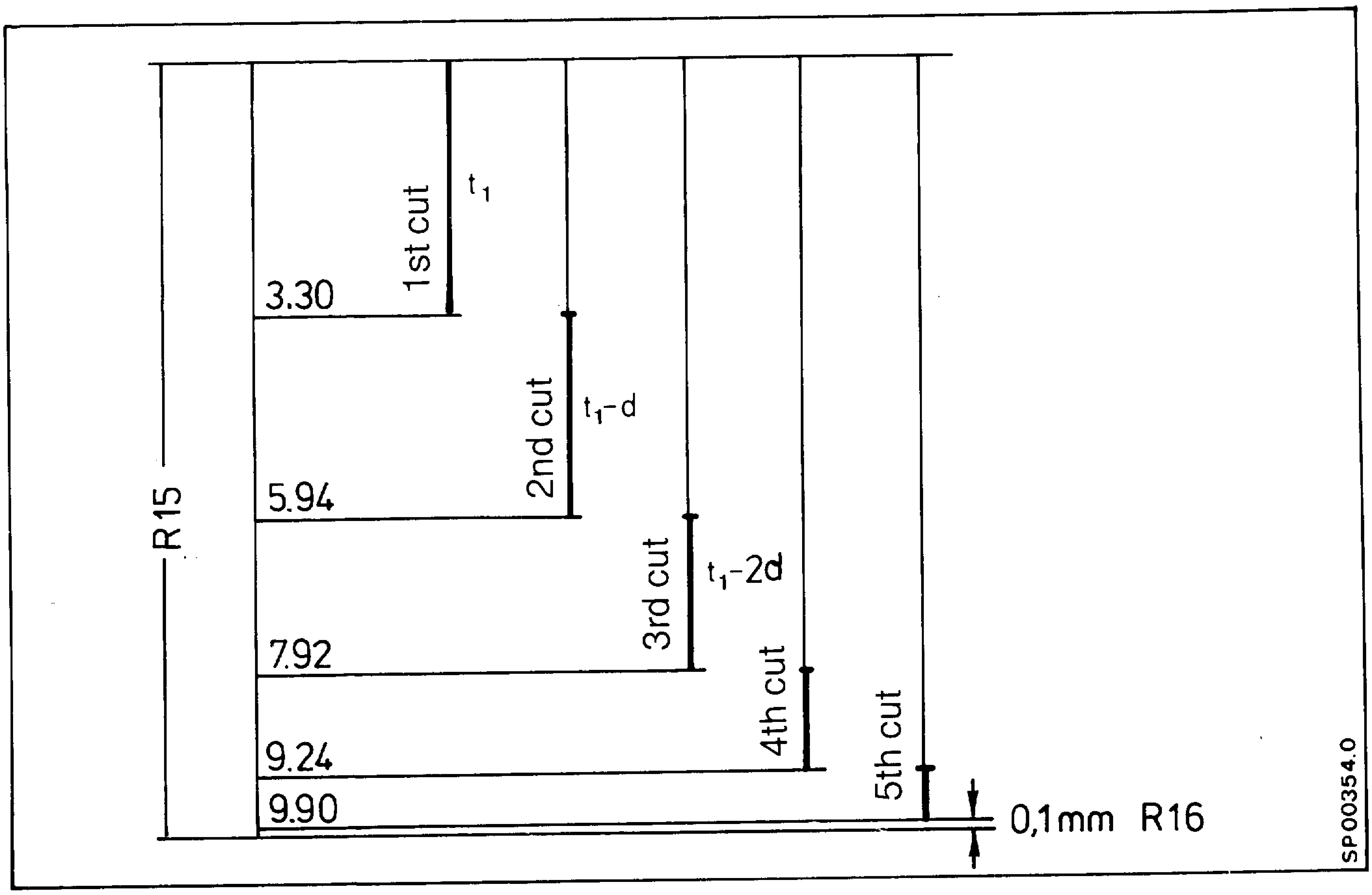
$$d = \frac{t_1}{n-1}$$

Example:

Thread depth t = 10

Number of roughing cuts = 5 t_1 = 3.3

Finishing cut depth = 01. d = 0.66



L94 Stock removal cycle, L96 Thread cutting cycle (Basic version 0, 1, 2, 3)

E 321 3T 0-3 GE548811.2091.01 (12.08.81) (old designation)

E 822 3T 0-3 GE548811.9101.01 (12.08.81)

```
% S P
L9400 (12.08.81..02)
G40 G90
R39 10 R25 0 R20
N4 R25 -R39
@02-4 R25 R39
R26 1 R27 1 R39 2
@02 5 R25 R39
R26-1
N5 @02 6 R11 R15
@02 6 R13 R15
R27-1
N6 R28 0 R26
R26 .R21
R29 0 R16
R29 .R26
R30 0 R17
R30 .R27
R33 0 R14
R34 0 R15
R33 -R29
R34 -R30
R10 -R29
R11 -R30
R12 -R29
R13 -R30
R40 9999 R39 1
R36 0 R18
R37 0 R18
@01 7 R39 R25
R39 3
@01 7 R39 R25
R31 0 R13
R35 0 R34
R37 .R27
R28 0 R27
R43 1
@00 8
N7 R31 0 R12
R35 0 R33
R36 .R21
R37 .R26
R43 0
N8 R39 1
R26 .R39
R27 .R39
R33 -R26
R34 -R27
R26 /R21
R32 20 R42 20 R39 0
@02 9 R32 R20
R32 30
@02 15 R32 R20
N9 R38 0 R41 2
G0 X R33 Z R34
N10 R32 0 R31
R32 -R35
R32 .R28
@03 15 R39 R32
R32 /R41
@03 11 R32 R36
R32 .R28
@02 11 R38 R39
R38 1 R37 0 R32
N11 R35 R37
@01 12 R43 R39
Z R35
@22 X R40 Z R35
G1 X R48 Z R49
G91 X -R26 Z -R27
G0 G90 X R33
@00 13
N12 X R35
@22 X R35 Z R40
G1 X R48 Z R49
G91 X -R26 Z -R27
G0 G90 Z R34
N13 @01-10 R38 R39
@02-10 R42 R20
N15 G0 X R33 Z R34
@02 18 R42 R20
@01 16 R43 R39
Z R13
@00 17
N16 X R12
N17 G1 X R12 Z R13
X R10 Z R11
G0 X R33 Z R34
N18 R10 R29
R11 R30
R12 R29
R13 R30
M17
L9600
G90 R25 1
R26 0 R15
R28 0 R29-1 R35-1
@03 1 R26 R28
R29 1
N1 R25 .R29
R31 0 R12
R31 -R10
R32 0 R13
R32 -R11
R31 /R19
R25 .R19
R10 R25
R25 /R19
R16 .R29
R26 R16
R17 -R35
R26 R26 /R17
R17 R35
R40 0 R26
R40 /R17
R39 0 R26 -R25
R39 -R40
G0 X R10 Z R11
N6 G91 X R26
G33 X R31 Z R32 K R14
G0 X -R26
G90 X R10 Z R11
R17 R35
@02 9 R17 R28
R26 0 R15
R26 -R25
@01 10 R16 R28
R16 0
@00-6
N9 R26 R39 -R40
@00-6
N10 @01 11 R18 R28
R18 R35
@00-6
N11 M17
L9900
@31 M17
M02
```

Safety clearance: L96 R25 = 1 for metric

R25 = 0.03937 inches

L94 in block N8 R39 = 1 for metric

R39 = 0.03937 for inches

L94 Stock removal cycle, L96 Thread cutting cycle

Basic version 4/3TT)

E 822 3T 4 GE548817.9101.04 (01.10.85)

```
%SP
L9400 (01.10.85..04 GERADE/KREIS)
R56 0 R66 1 R67 1 R74 0 R15
R82 0 R11 R84 0 R13 R85 0 R23 R86 0 R24
@31
R90 2409 @29 36110 R90
R61 R66 G40 G90
R79 1
@01 22 R21 R79
@02 23 R61 R79
R79 .5
@00 23
N22 @01 23 R61 R79
R79 2
N23 R10 .R79
R12 .R79
R14 .R79
R51 180 R55 360
@01 3 R66 R22
@01 3 R56 R22
R53 0 R12 R54 0 R13 R57 0 R23 R58 0 R24
R53 -R10
R54 -R11
@18 R53
@18 R57
R53 -R57
@03 2 R53 R56
R53 R55
N2 R22 2
@02 3 R51 R53
R22 3
N3 R87 0 R22 R79 10
R73 0 R14 R81 0 R10 R83 0 R12 R65 0 R20
N4 R65 -R79
@02-4 R65 R79
R79 1
@01 20 R65 R79
R79 3
@01 20 R65 R79
@02 21 R11 R13
@02 24 R74 R11
N25 R74 0 R11
@00 24
N21 @02-25 R74 R11
@00 24
N20 R79 3
@03 26 R65 R79
```

Safety clearance: L96 R65 = 1 for metric

R65 = 0.03937 for inches

```
@02 24 R10 R73
N27 R73 0 R10
@00 24
N26 @02-27 R10 R73
N24 R79 2
@02 5 R65 R79
R66-1
N5 @02 6 R82 R74
@02 6 R84 R74
R67-1
N6 R68 0 R66
R66 .R61 R69 0 R16
R69 .R66 R70 0 R17
R70 .R67
R73 -R69
R74 -R70
R81 -R69
R82 -R70
R83 -R69
R84 -R70 R79 1
R76 0 R18 R77 0 R18
@01 7 R79 R65
R79 3
@01 7 R79 R65
R71 0 R84 R75 0 R74
R77 .R67 R68 0 R67
R96 1
@00 8
N7 R71 0 R83 R75 0 R73
R76 .R61
R77 .R66 R96 0
N8 R79 1 R93 99999
@31
R90 6408 @29 39110 R90
@01 3 R91 R56
R79 .03937 R93 9999
N3 R66 .R79
R67 .R79
R73 -R66
R74 -R67
R66 /R61 R72 20 R95 20 R79 0
G X R73 Z R74
@02 9 R72 R20
R72 30
@02 15 R72 R20
N9 R78 0 R94 2
N10 R72 0 R71
R72 -R75
R72 .R68
@03 15 R79 R72
R72 /R94
@03 11 R72 R76
R72 .R68
@02 11 R78 R79
R78 1 R77 0 R72
N11 R75 R77
@01 12 R96 R79
Z R75
@22 X R93 Z R75
G1 X R91 Z R92
```

```
G91 X -R66 Z -R67
G G90 X R73
@00 13
N12 X R75
@22 X R75 Z R93
G1 X R91 Z R92
G91 X -R66 Z -R67
G G90 Z R74
N13 @01-10 R78 R79
@02-10 R95 R20
N15 @02 2 R95 R20
@01 16 R96 R79
Z R84
@00 17
N16 X R83
N17 G1 X R83 Z R84
R88 2
@02 19 R88 R87
@01 20 R88 R87
R88-1
@00 1
N20 R88 1
N1 R87 R88
R60 0 R81 R62 0 R83
R60 /R61
R62 /R61
R85 -R62
R85 R60
R86 -R84
R86 R82
G9 G R87 X R81 Z R82 I R85 K R86
@00 2
N19 G9 G R87 X R81 Z R82
N2 G X R73 Z R74
M17
L9600
R81 0 R10 R82 0 R11 R83 0 R12
R84 0 R13 R85 0 R14 R86 0 R15
R87 0 R16 R88 0 R17 R89 0 R18
R77 0 R65 1 R80 0 R19
@31
R90 6408 @29 39110 R90
@01 1 R91 R77
R65 .03937
N1 G90 R68 0 R69-1 R75-1 R66 0 R86
@03 1 R66 R68
R69 1
N1 R65 .R69
R71 0 R83
R71 -R81
R72 0 R84
R72 -R82
R71 /R80
R65 .R80
R81 R65
R65 /R80
R87 .R69
R66 R87
R88 -R75
R66 R66 /R88
```

```
R88 R75
R93 0 R66
R93 /R88
R79 0 R66 -R65
R79 -R93
G0 X R81 Z R82
N6 G91 X R66
G33 X R71 Z R72 K R85
G0 X -R66
G90 X R81 Z R82
R88 R75
@02 9 R88 R68
R66 0 R86
R66 -R65
@01 10 R87 R68
R87 0
@00-6
N9 R66 R79 -R93
@00-6
N10 @01 11 R89 R68
R89 R75
@00-6
N11 M17
L9900
@31 M17
M02
```

7.1.3 L95 Stock removal cycle (Basic Version 4B/4C)

R20 Subroutine number under which the contour is stored

R21 Start point of the contour in X (absolute)

R22 Start point of the contour in X (absolute)

R24 Finishing cut depth in X (incremental)

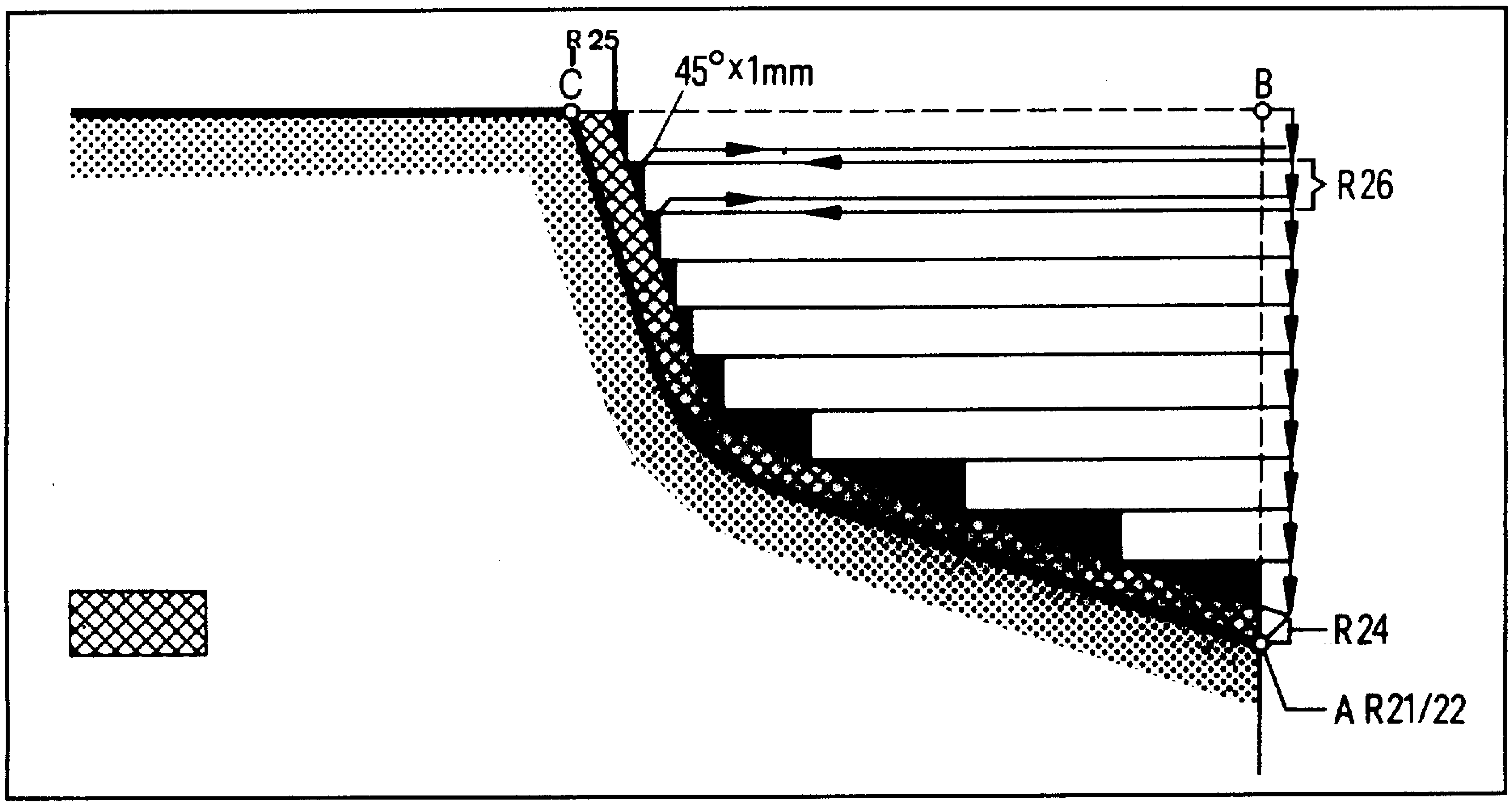
R25 Finishing cut depth in Z (incremental)

R26 Roughing depth in X or Z (incremental)
(not necessary for finishing with R29 21, R29 22)

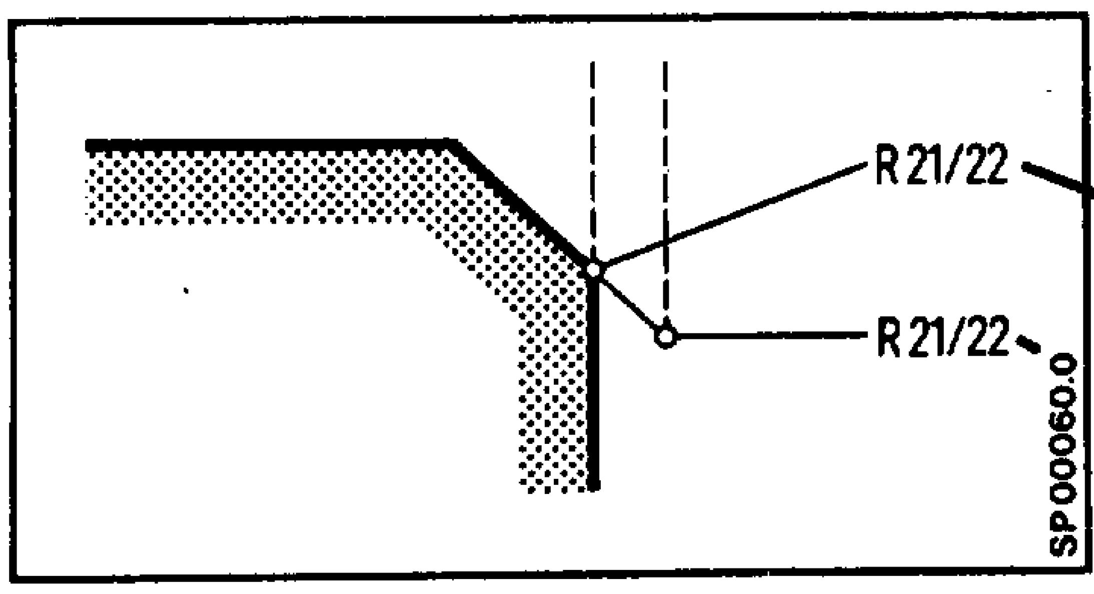
R27 Tool nose radius compensation (41, 42)

R29 Type determination for roughing and finishing

In the following diagram the input parameters are shown.



The start points R21 (X) and R22 (Z) are given referred to the contour. The control is offset automatically when roughing by the finishing cut depth R24, R25 + 1 mm safety margin. If this margin is not sufficient, then the contour start point R21, R22 must be correspondingly shifted.



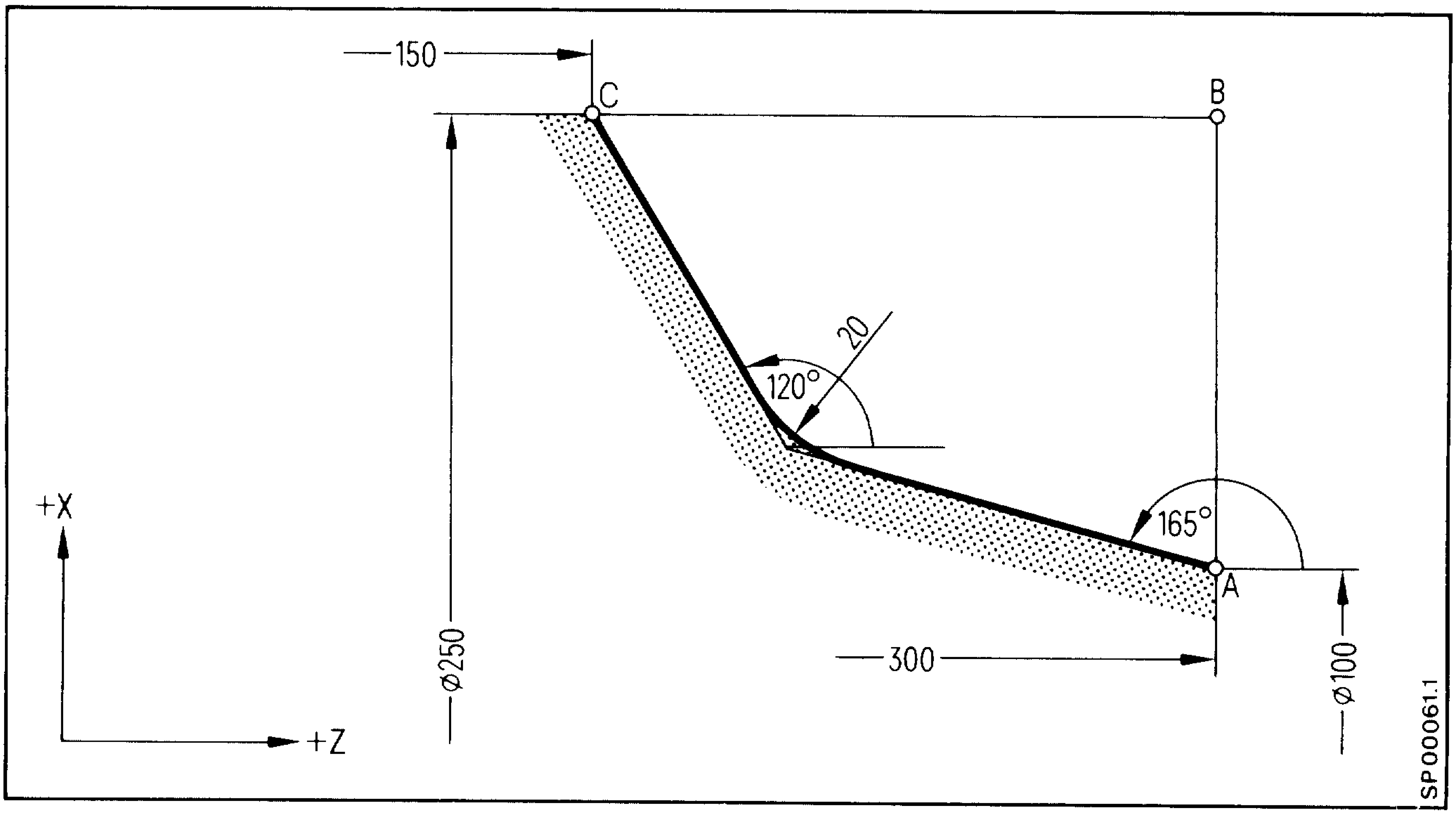
Original contour start point

Shifted start point

Contour (R20)

For automatic stock removal, the end contour of the finished part must be described. This is stored as a subroutine and called up within the stock removal cycle.

The subroutine can comprise any number of blocks. However, each block must contain a coordinate value, blocks without traversing movement are not permitted. Recess cutting in the roughing cycle is not permitted. The final contour description can be made via blueprint programming. The starting point, which is defined via R21, R22 must not be programmed in the first block of the contour.



L1000

N1 G90 A165. A120. X250. Z150. B20. M17 LF

The contour specification in the cycle call L95 results from R20 10. The corner point B represents also the reversing point in the finishing cycle. It is determined by the cycle from the values of the points A and C. End of the cycle is in point B.
Delete blocks in the contour are allowed.

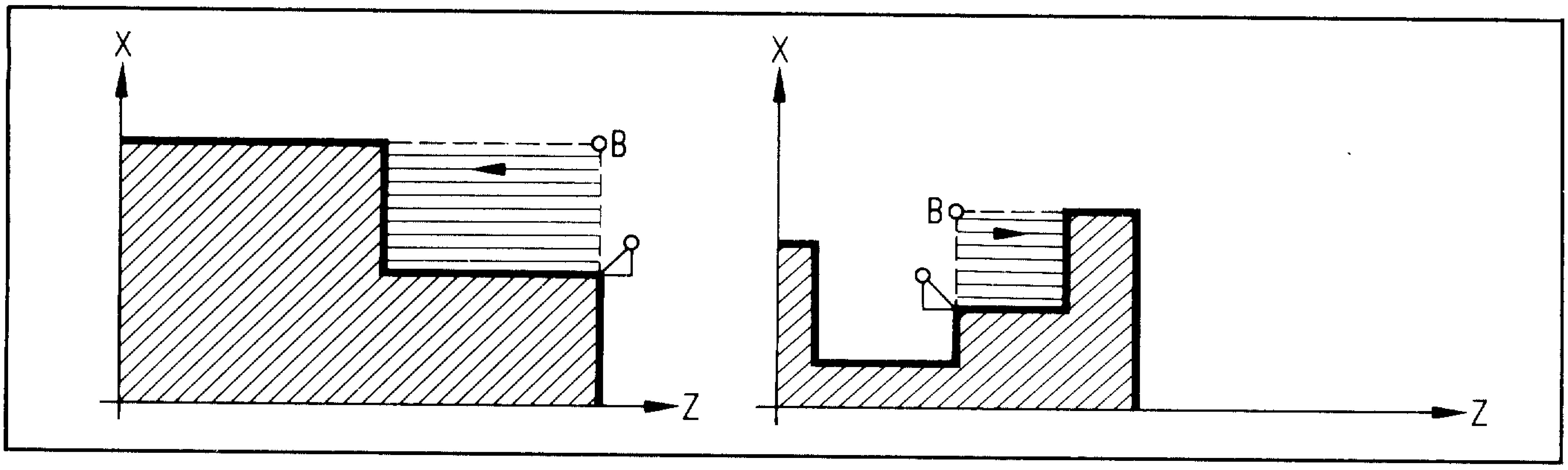
Type of machining (R29)

The parameter R29 gives the type of stock removal: roughing or finishing, whether external or internal machining, the type of cut distribution, whether longitudinal or facing.

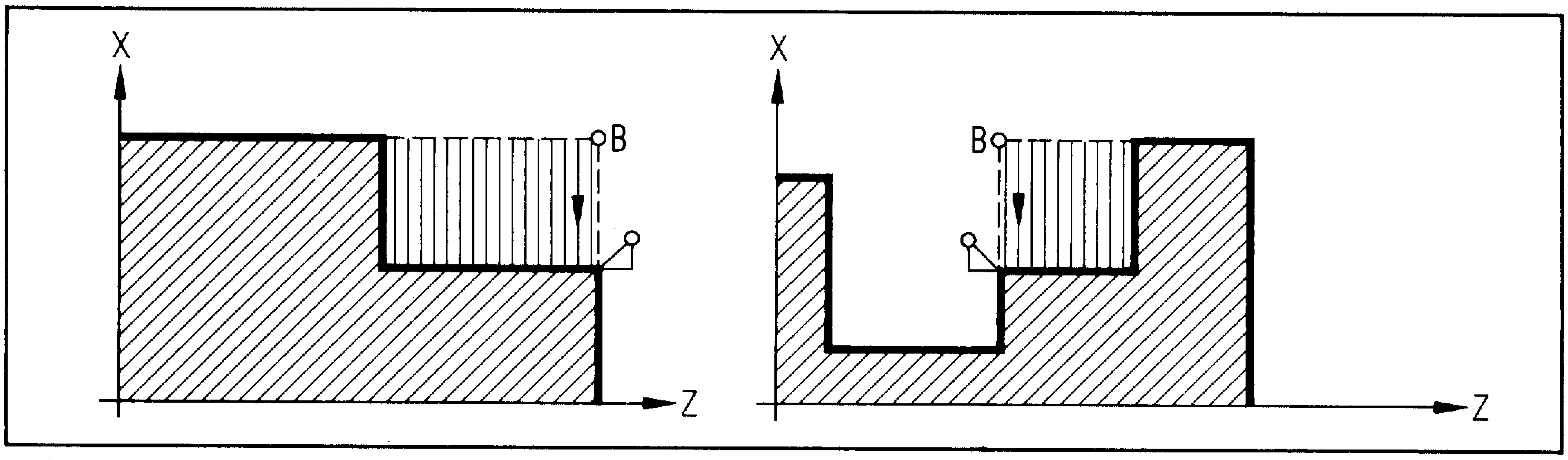
R29	11	longitudinal	(Z)	external	Roughing
R29	12	facing	(X)	external	
R29	13	longitudinal	(Z)	internal	
R29	14	facing	(X)	internal	
R29	21			external	*21, 23 finishing to finishing cut depth
R29	23			internal	
R29	31	longitudinal	(Z)	external	*Roughing paraxial, then 1 cut contour parallel to finishing cut depth
R29	32	facing	(X)	external	
R29	33	longitudinal	(Z)	internal	
R29	34	facing	(X)	internal	
R29	41	longitudinal	(Z)	external	*see R2931 - R2034 then 1 cut contour-parallel to end contour
R29	42	facing	(X)	external	
R29	43	longitudinal	(Z)	internal	
R29	44	facing	(X)	internal	

* In these cases the controller automatically activates the cutter radius compensation in the correct sequence if beforehand a selection has been made via R27 (41 or 42).
In the case of paraxial roughing, the cutter radius compensation is internally suppressed. At the end of the cycle it is deselected and must be programmed anew if necessary.
The selection or cancellation of the cutter radius compensation in the cycle is independently controlled by the cycle.
If cutter radius compensation (CRC) has been selected, the "Inhibit cycle" signal must be active at the interface, and the limit for R parameter display (machine parameter 382) must be set as less than or equal to 50.

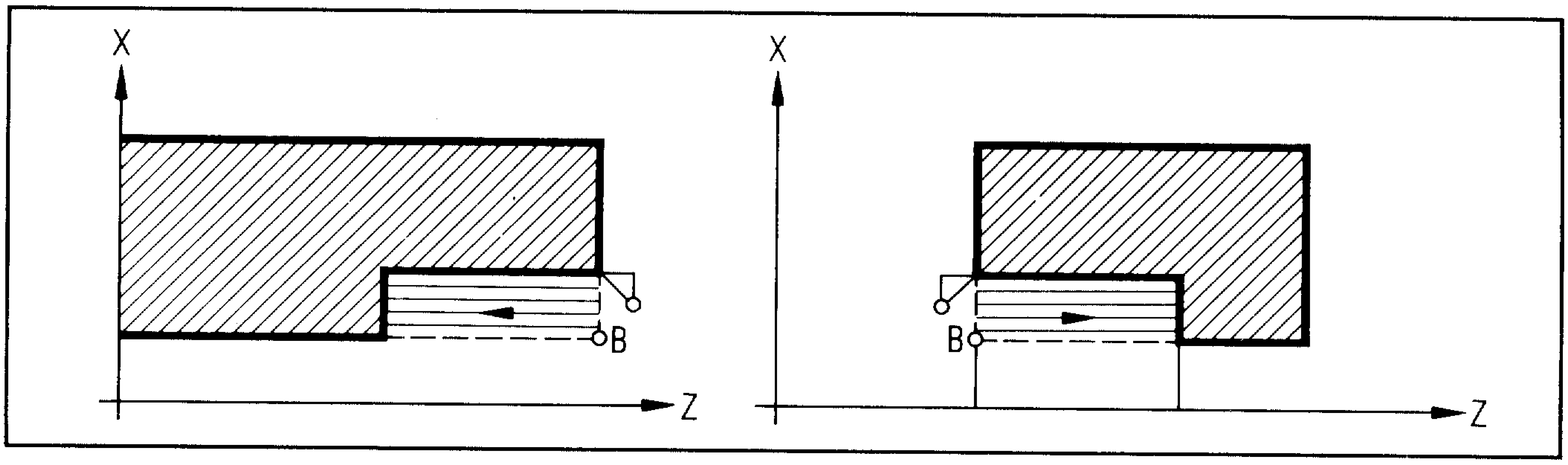
Examples:



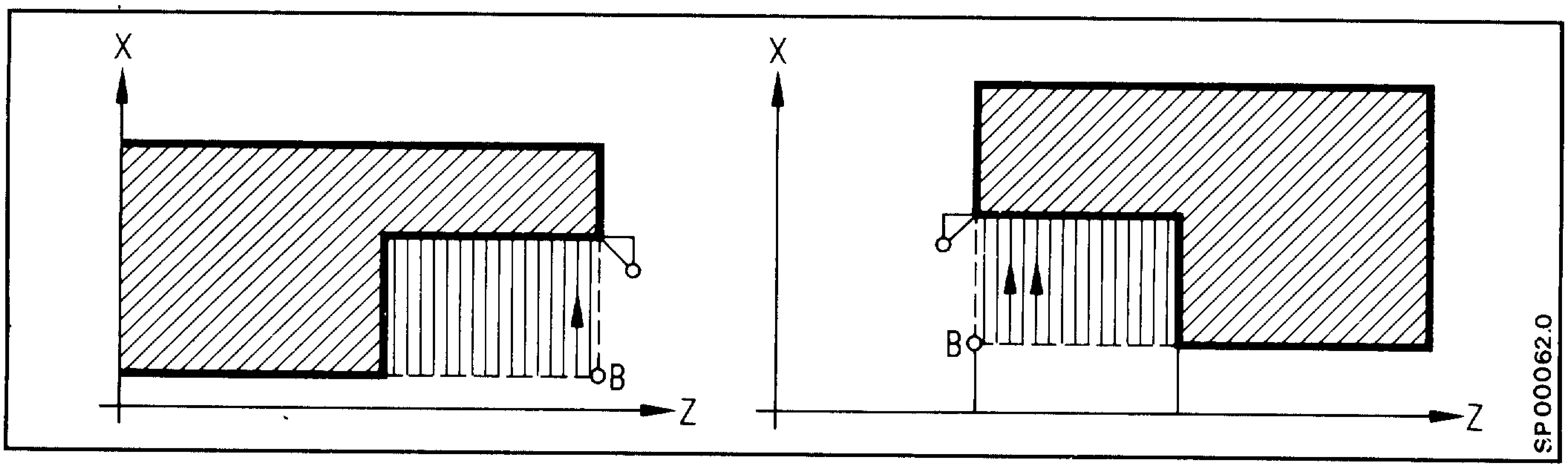
R29 11 External roughing, longitudinal (Z)



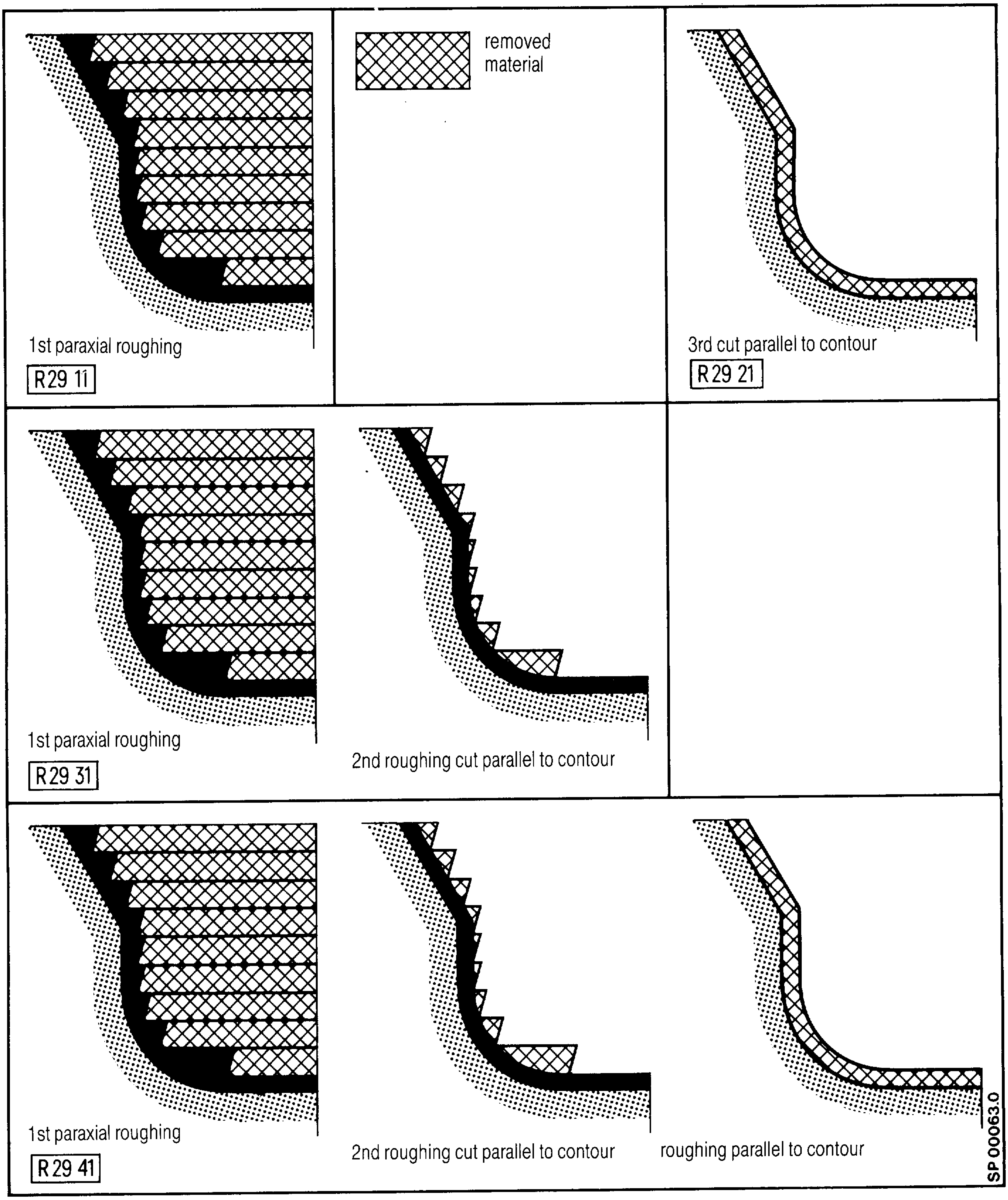
R29 12 External roughing, facing (X)



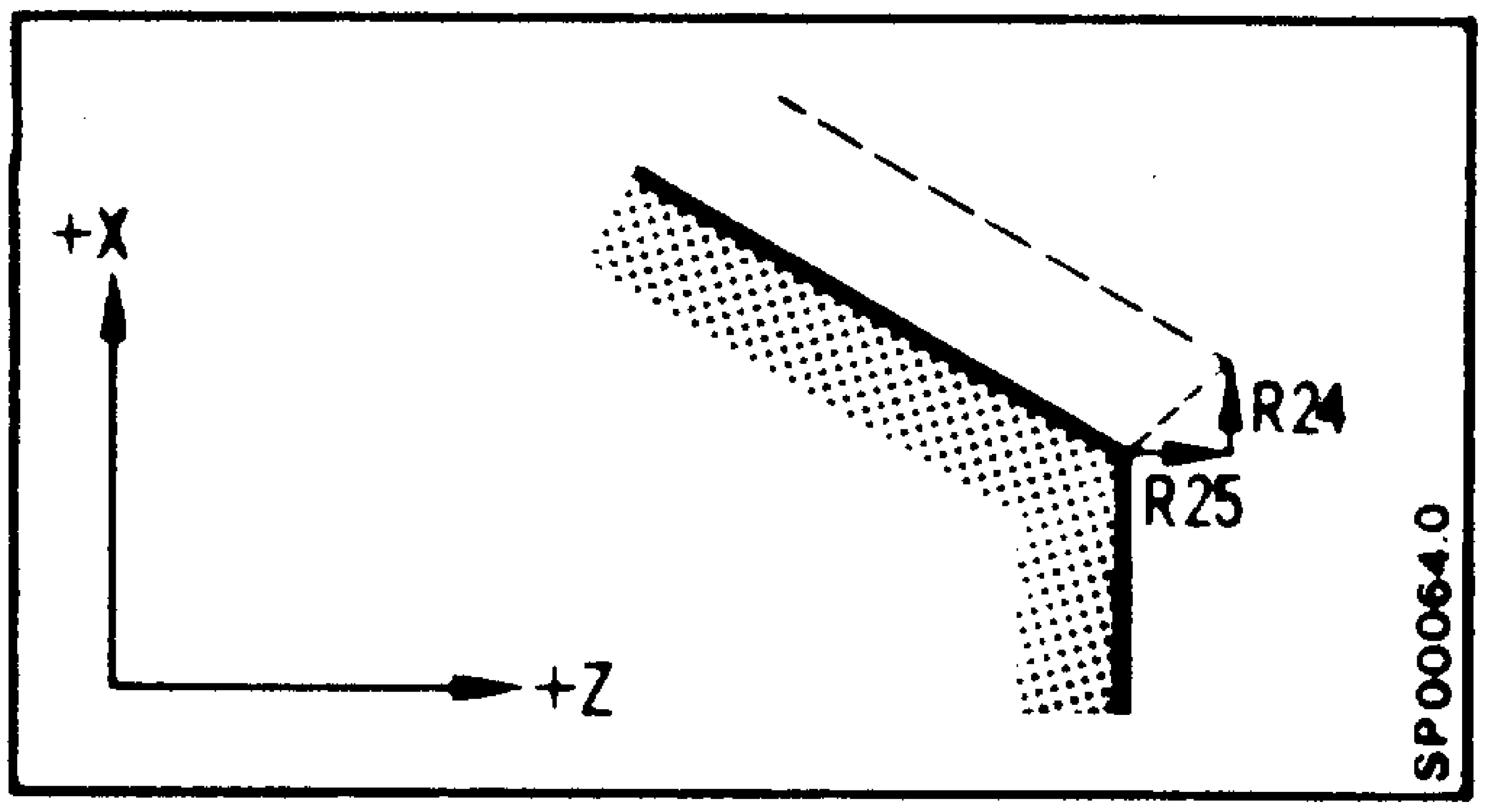
R29 13 Internal roughing, longitudinal (Z)



R29 14 Internal roughing, facing (X)

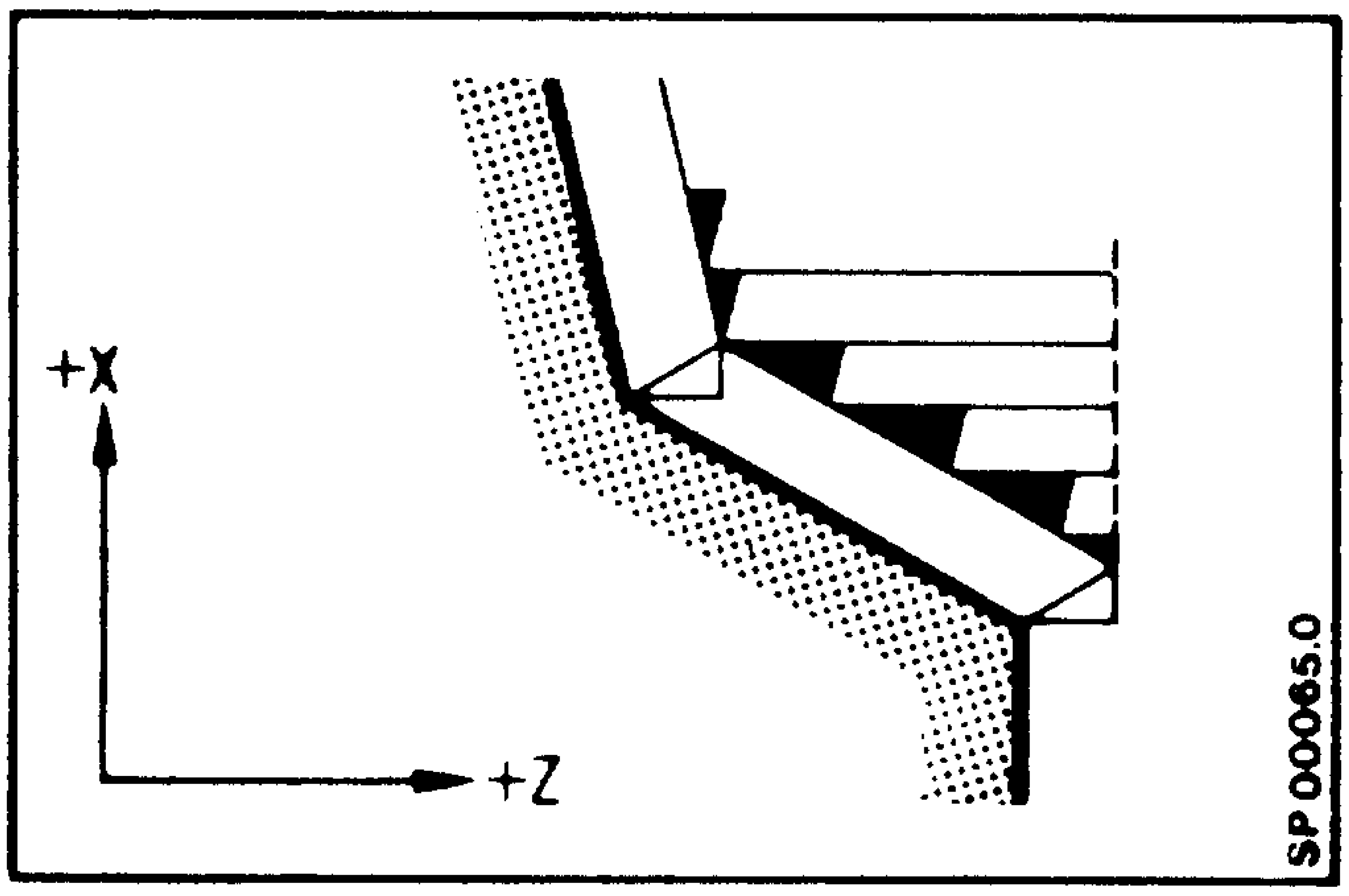

removed material
1st paraxial roughing
R29 11
3rd cut parallel to contour
R29 21
1st paraxial roughing
R29 31
2nd roughing cut parallel to contour
1st paraxial roughing
R29 41
2nd roughing cut parallel to contour
roughing parallel to contour
SP 00063.0

Finishing cut depth (R24, R25)

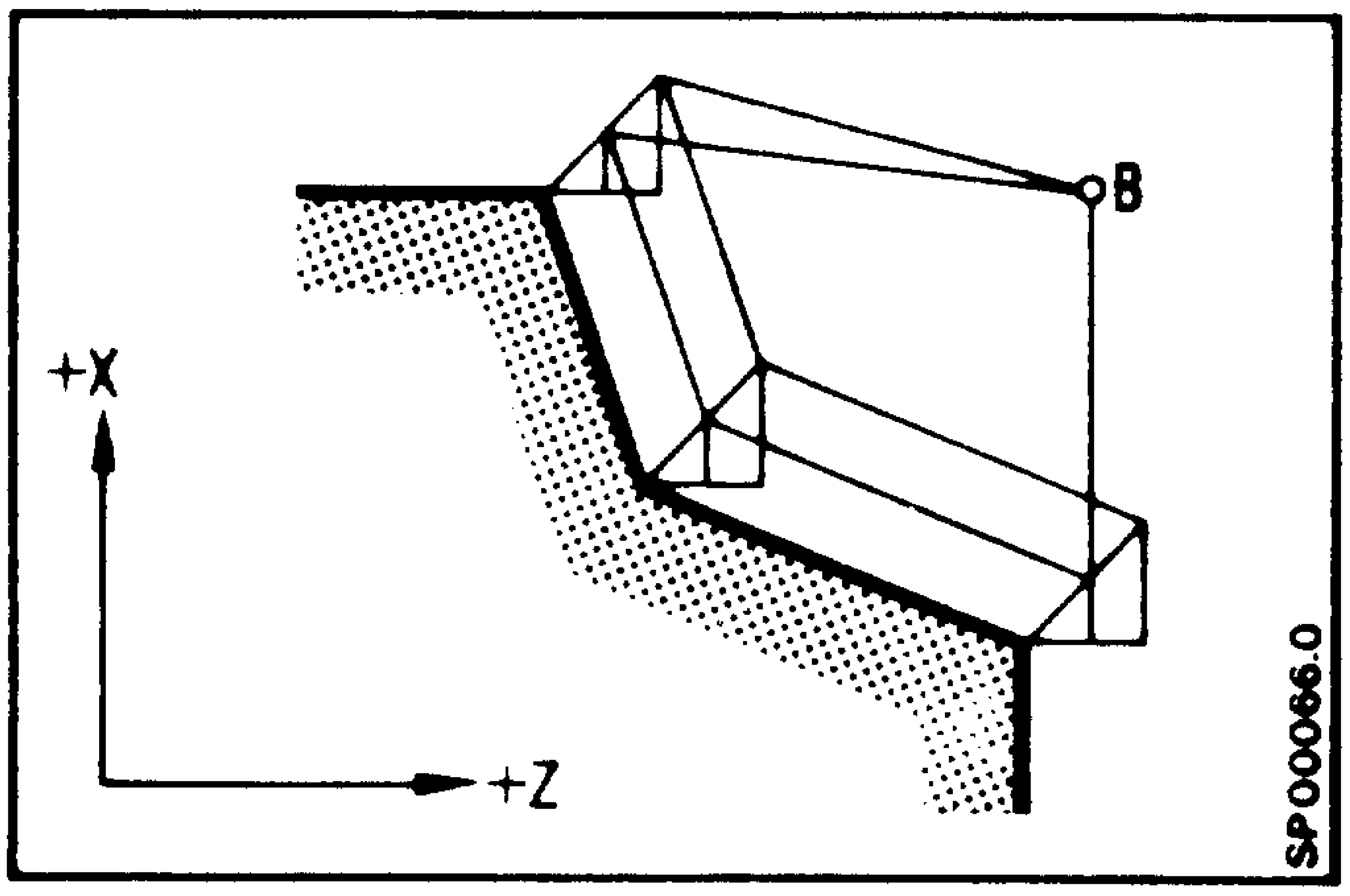


The contour is offset by the entered finishing cut depth with R24 and R25.
E.g. R24 0.3
R25 0.3

In the roughing cycle, the roughing is done up to this depth.



In the finishing cycle, finishing to the cut depth entered:



Finishing cut depth R24 0, R25 0 corresponds to the final contour.

Through several finishing cut depths, for example, it is possible to turn the corners left after roughing by using a copying tool, if the cycles R29 31 - R29 34 or R29 41 - R29 44 are not accessed.

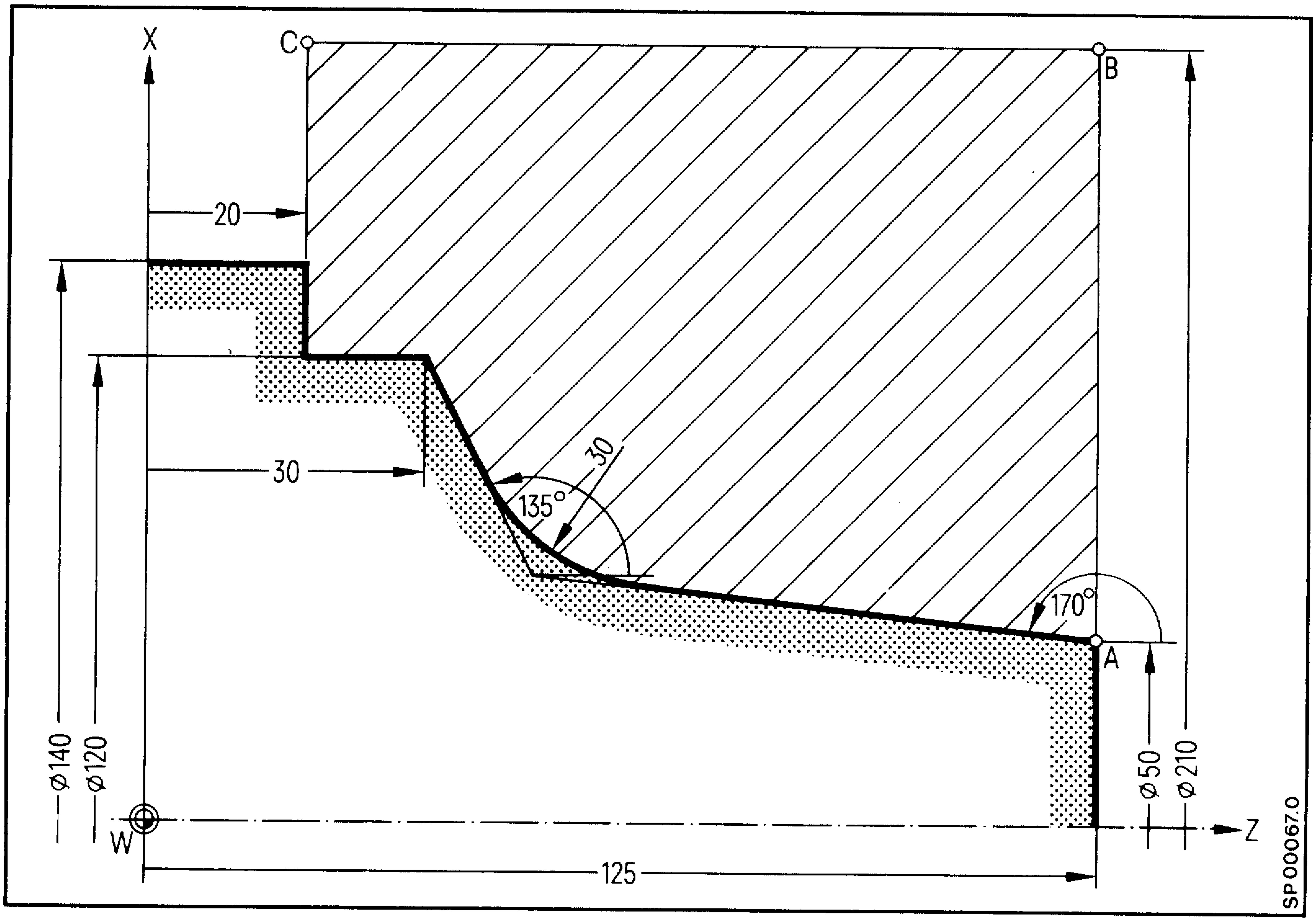
Stock removal at constant cutting speed

If the cycle is supposed to run at a constant speed, then the "Constant Cutting Speed G96" function is to be selected before calling up the cycle.

For example:

```
N.. G96   S180  LF
N.. R20 .. R21.. R22..  R24..  R25..  R26..  R27..  R29.. LF
N.. L95           F...    LF
N.. L91  LF
N.. T... LF
```

Example:



The stock removal contour is stored in a subroutine.

```
L7000
N1 G90 A170. A135. X120. Z30. B30. LF
N2                  Z20.          LF
N3           X210. M17            LF
```

Parameter Determination:

A Start point X	50 mm	R21	50
Z	125 mm	R22	125
Finishing Cut Depth X	3 mm	R24	3
Finishing Cut Depth Z	3 mm	R25	3
Depth of Cut	5 mm	R26	5
Contour L7000		R2070	

Call in Part Program:

```
N16   R20 70 R21  50. R22 125. R24  3
      R25  3 R26  5. R27 42 R29 11          R29 11 Roughing
          L95 F...                       LF
N17       L92                            LF  Withdraw for change of tools
N18       T..                            LF  Tool change
N19  R24  2.5 R25 2.5 R27 42 R29 21          R29 21 Finishing
          L95 F...                       LF  R24 2.5
                                             R25 2.5 Finishing cut
                                                     depth 2.5 mm
N20  R24  .05 R25 .05 R27 42 L95         LF  R24 .05
                                             R25 .05 Finishing cut depth
                                                     0.050 mm
N21  R24  0.  R25 0. R27.42 L95 F.. ..   LF  R24 0
                                             R25 0 Finishing cut depth 0=
                                                   final contour
```

Or roughing and finishing in one call:

```
N15  R20  70 R21 50 R22 125 R24 .3
     R25.3 R26 5 R27 42 R29 41  LF     Paraxial roughing
N16       T... L95 F...         LF     1 cut parallel to contour
                                       1 cut to final contour
```

7.1.4 L97 Thread cutting cycle (Basic Control 4 B)

With the aid of this cycle, outside threads, inside threads as well as taper threads can be cut.
The feed of the tool is automatic and decreases quadratically, the cut depth remains constant.

Before calling up the L97 cycle, the following R parameters must be assigned a value:
R20 Thread pitch
R21 Start point of thread in X (absolute)
R22 Start point of thread in Z (absolute)
R23 Number of compound feeds
R24 Thread depth (incremental), sign required to define inside or outside thread, + = inside thread/- = outside thread
R25 Finishing depth
R26 Approach path
R27 Run-out path
R28 Number of roughing cuts
R29 Infeed angle
R31 End point of thread in X (absolute)
R32 End point of thread in Z (absolute)
The values must be assigned in two blocks, since a maximum of 10 R parameters per block only are permissible.

Example

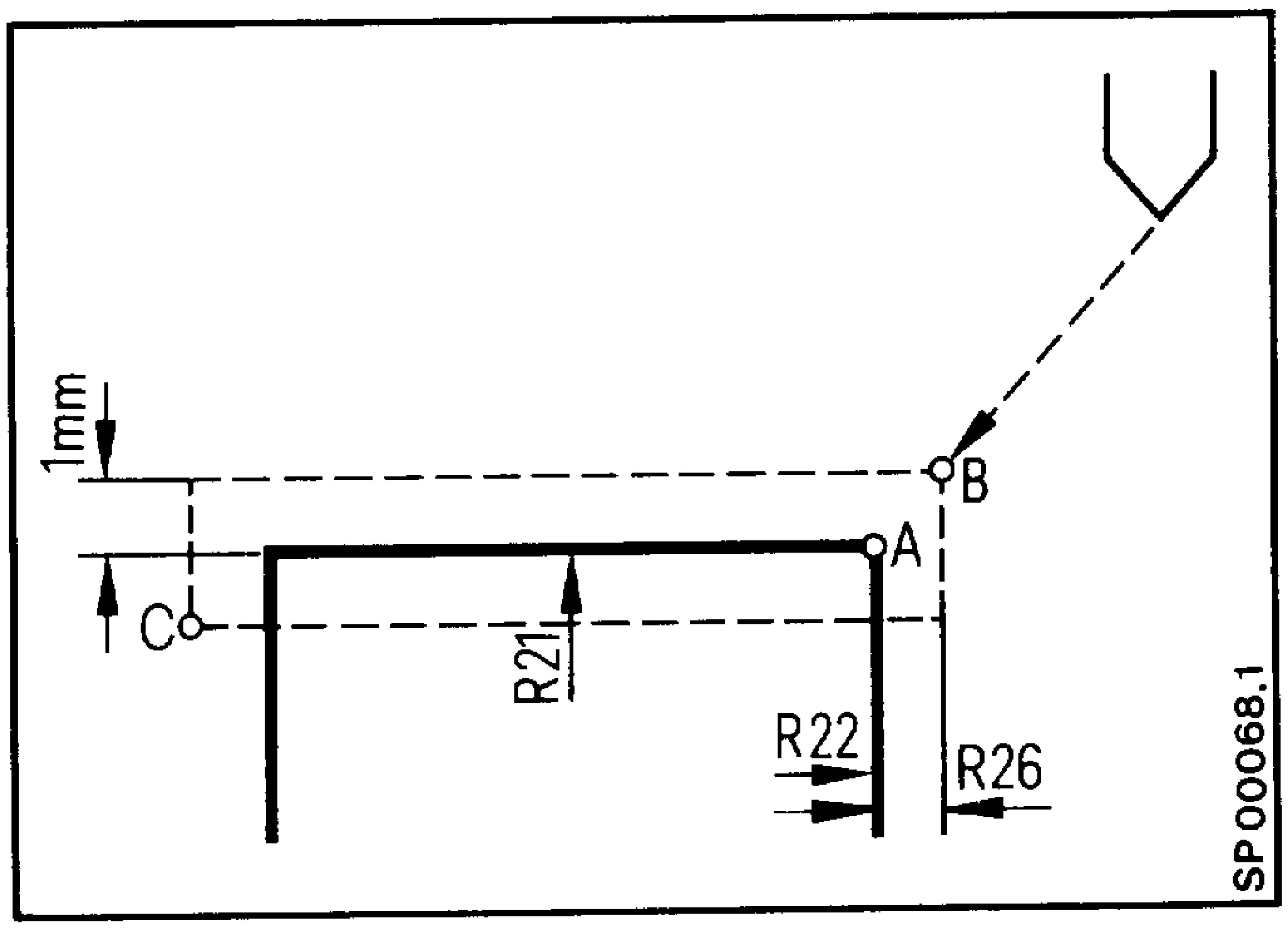
```
N10  R20... R21... R22... R23... R24... R26... R27... LF
N15  R28... R29... R31... R32... L97 LF
```

The following diagrams show the individual parameter values.

R20 Thread pitch

The parameter represents the value of the thread pitch. The paraxial value without a sign is always entered.
min. 0.001 mm; may. 400 mm

R21 and R22 Thread start point



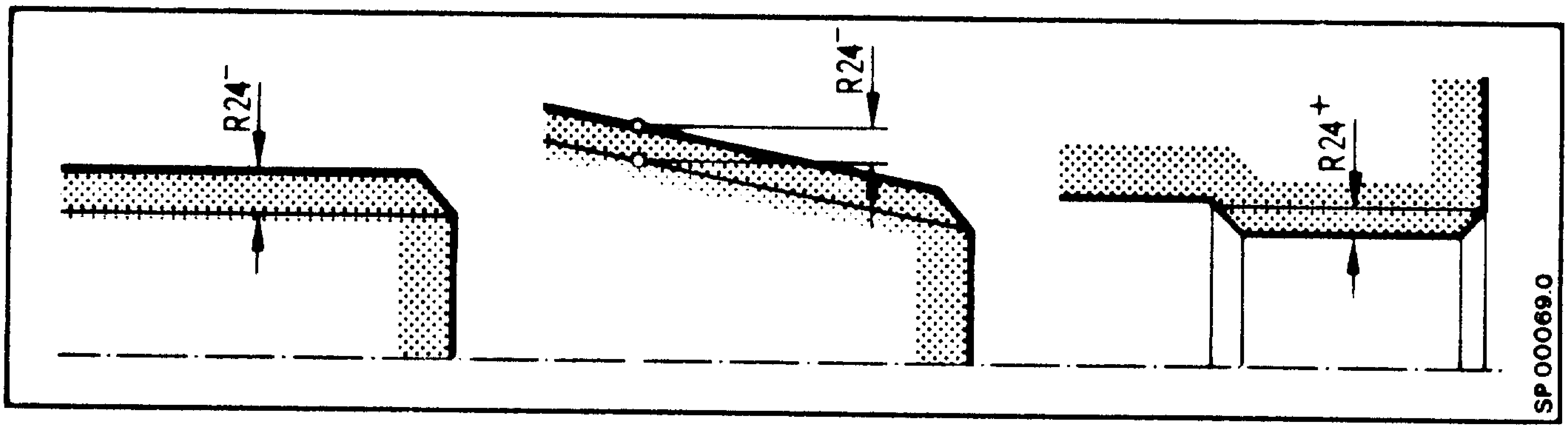
The parameters R21 and R22 represent the original starting points of the thread (A). The start point of the thread cycle is a Point B which lies at the parameter R26 (approach path) before the thread output point. The start point B lies in the X axis 1 mm over the parameter value R21. This raised plane is generated automatically in the controller. The thread cycle can be called up from any cutting position, the infeed to point B is effected at rapid tranverse rate.

R23 Idle passes

Any number of idle passes can be selected. They are entered using parameter R23.
e.g. 3 Idle passes R23 3.

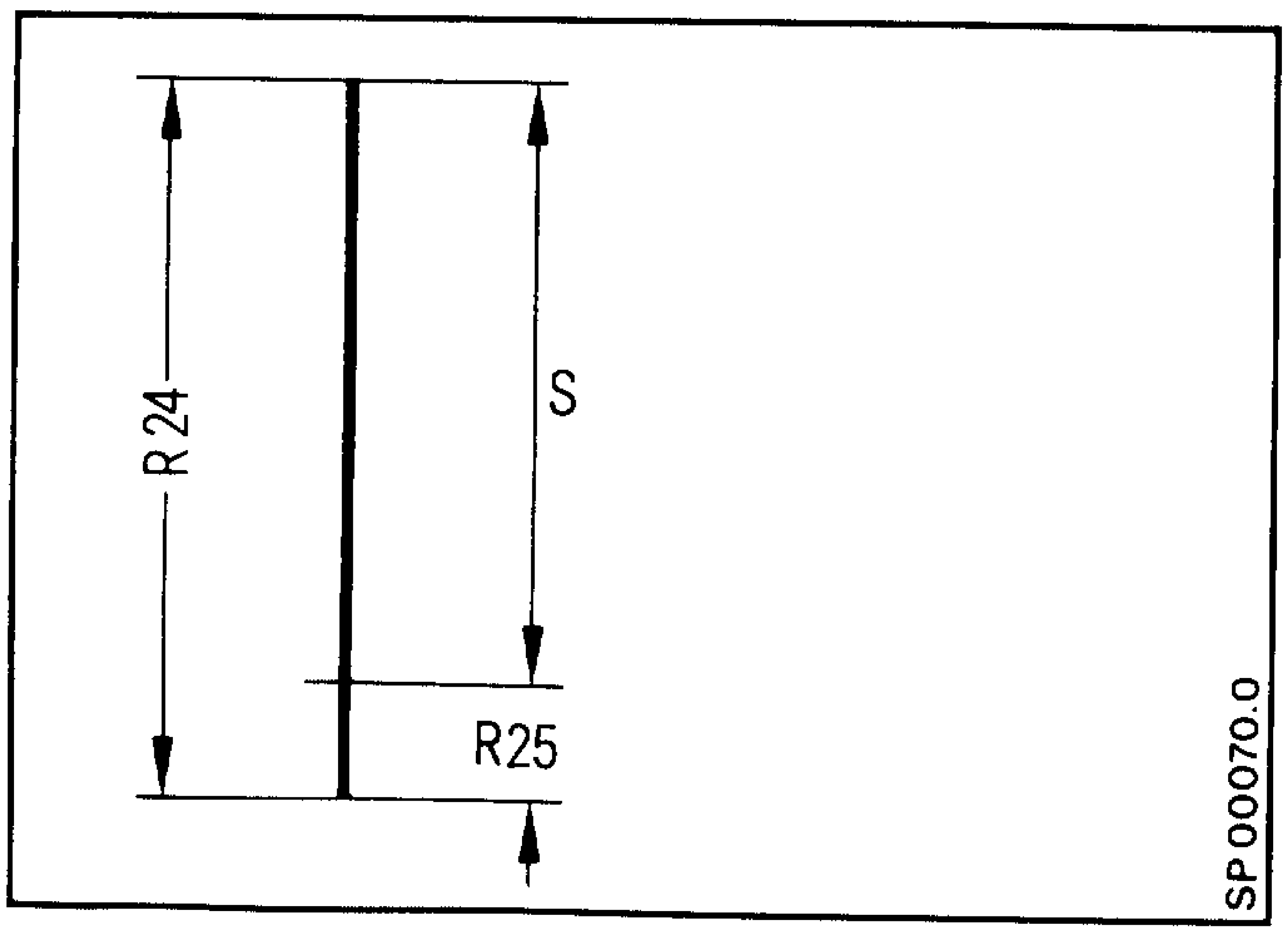
R24 Thread depth

The thread depth is entered using parameter R24. The sign determines the infeed direction, i.e. whether it is an outside or inside thread (+ inside thread, - outside thread).



R25 Finishing cut depth

R25 gives the finishing cut depth. When a finish cut depth is programmed, this depth is subtracted from the thread depth and the remaining value divided into roughing cuts. After the roughing cuts have been completed, first a finishing cut is made and then the number idle passes run as programmed under R23.



R24 = thread depth
R25 = finishing cut depth
S = Roughing depth, automatically calculated and divided into roughing cuts.
Finish cut depth R25

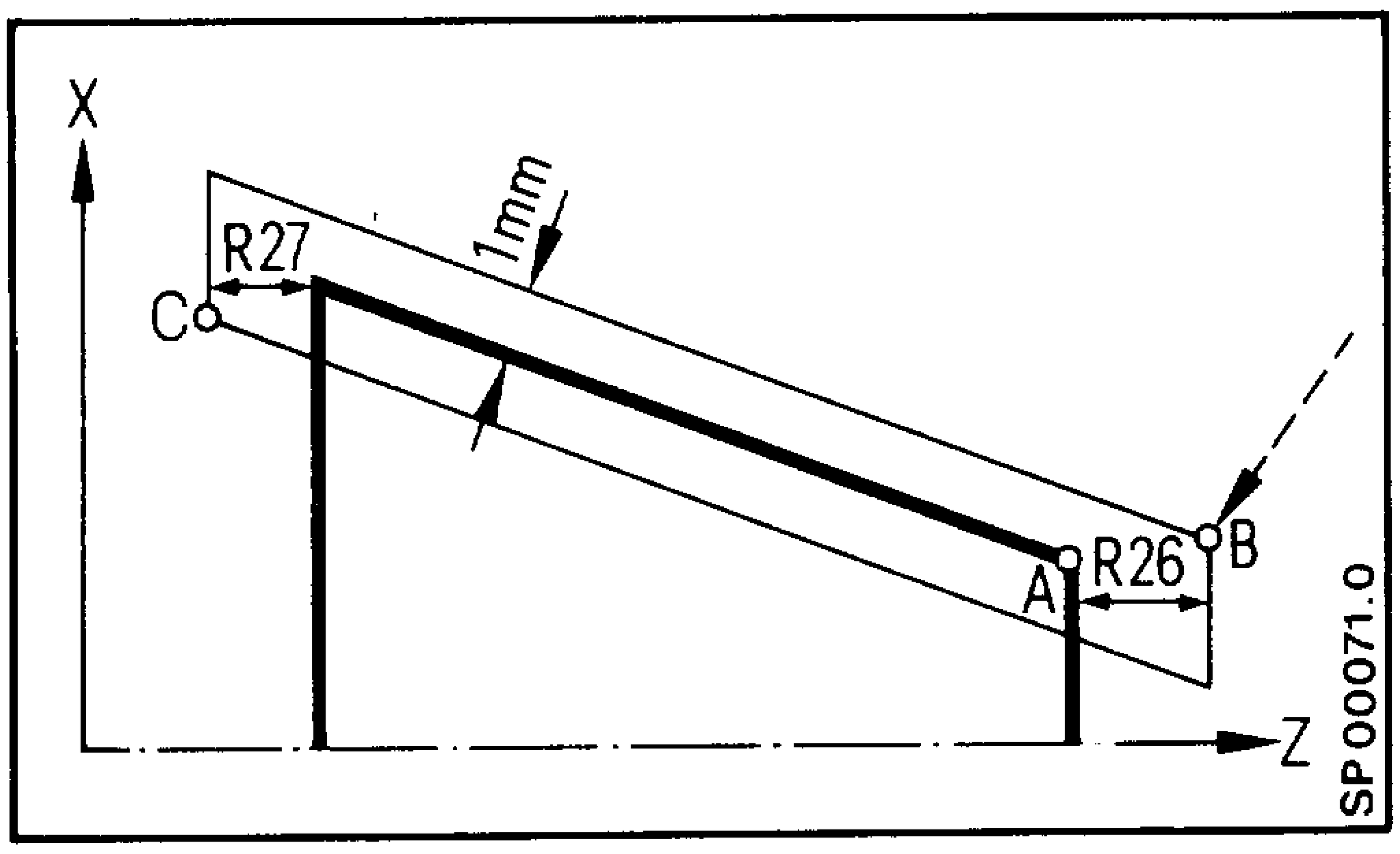
R26 Approach path

R27 Run-out path

The approach and run-out paths are entered without signs.

The parameters represent paraxial, incremental values.

In the case of taper threads, the control calculates the approach and run-out path distances in relationship to the taper and determines and corner points B and C.



R28 Number of Roughing Cuts

The parameter value determines the number of thread roughing cuts. The control automatically calculates the individual infeed depth. This ensures that the cut pressure from the first to the last roughing cut is the same.

The depth of the first roughing cut t is calculated as follows:

$$\Delta t = \frac{t}{\sqrt{R28}} \cdot \sqrt{i}$$

t = R24 - R25

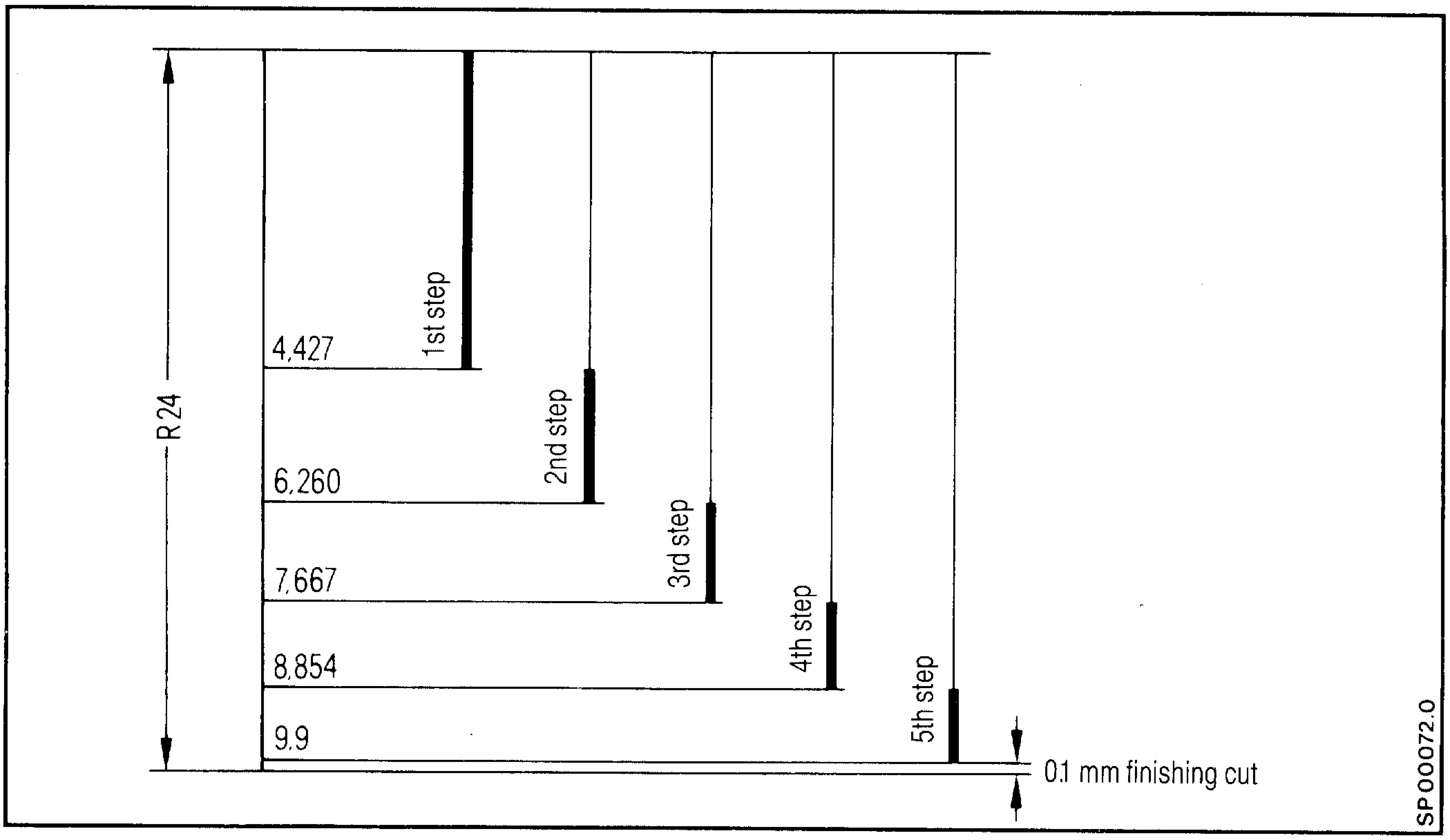
i = first cut

Example:

Thread depth t = 10

Number of roughing cuts = 0

Finishing cut depth = 0.1

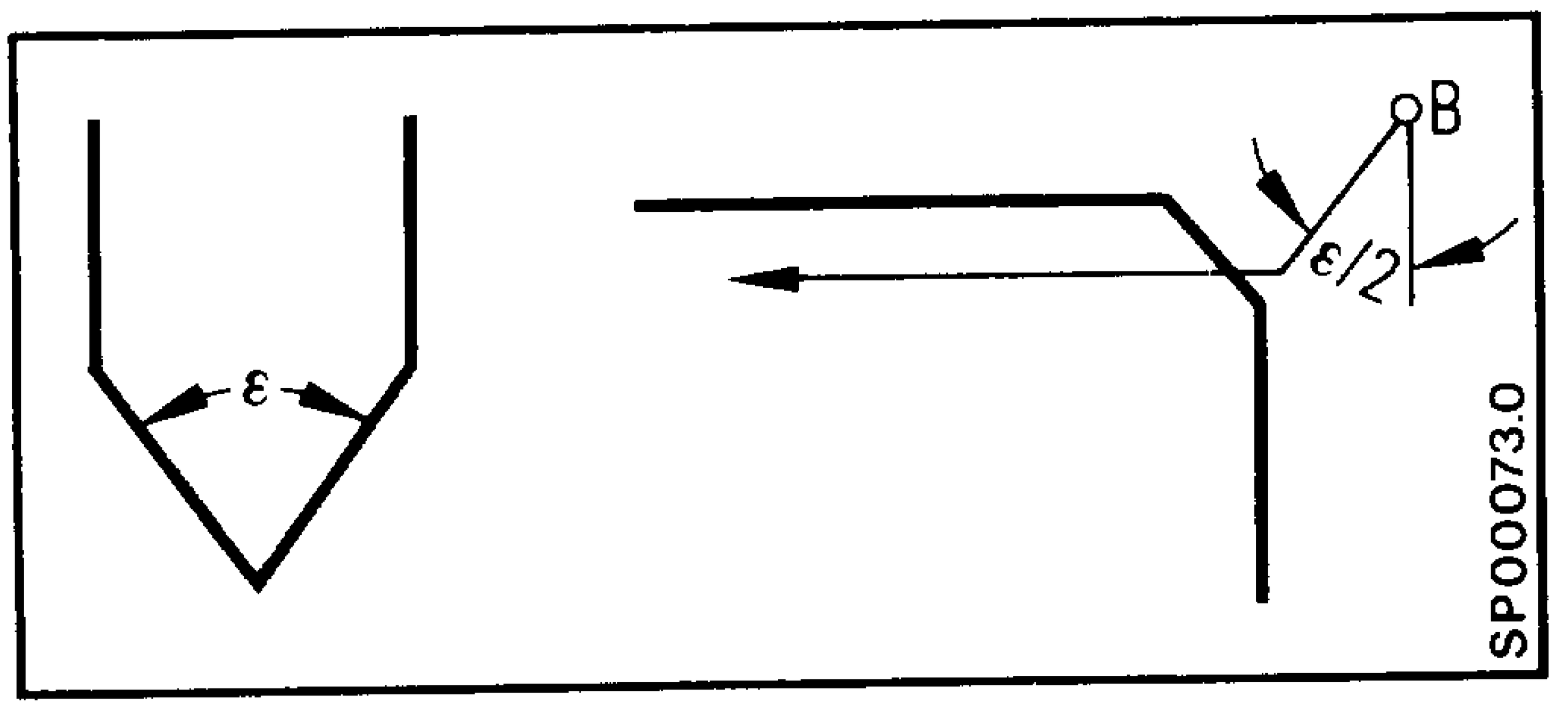


R29 Infeed angle for longitudinal or transverse thread

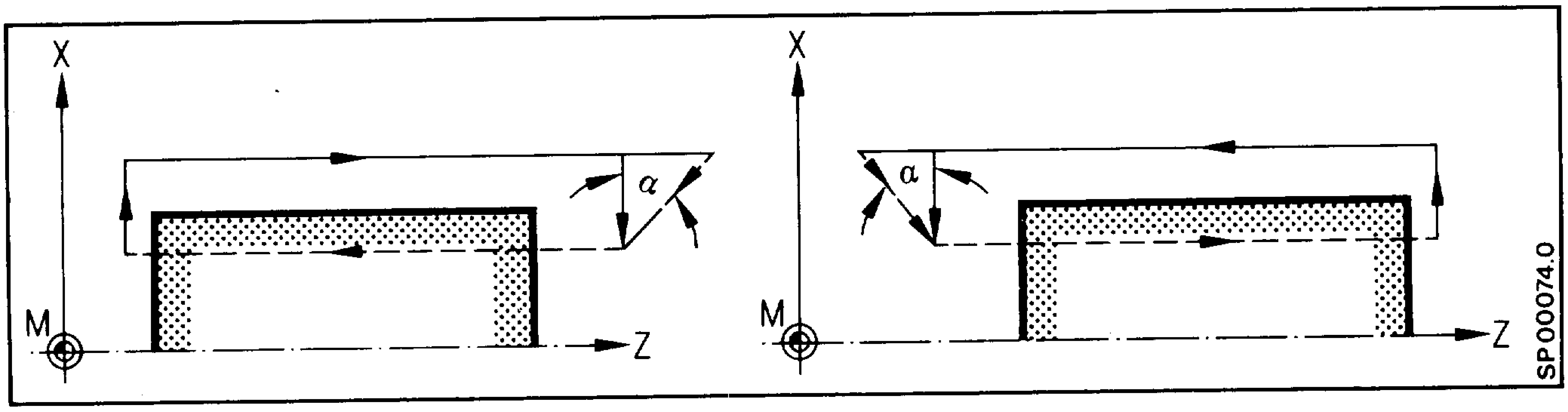
The infeed of the tool is at any desired angle possible, for longitudinal or transverse threads.

The angle input is made without sign.

e.g. /2 = 30^{o} R29 30



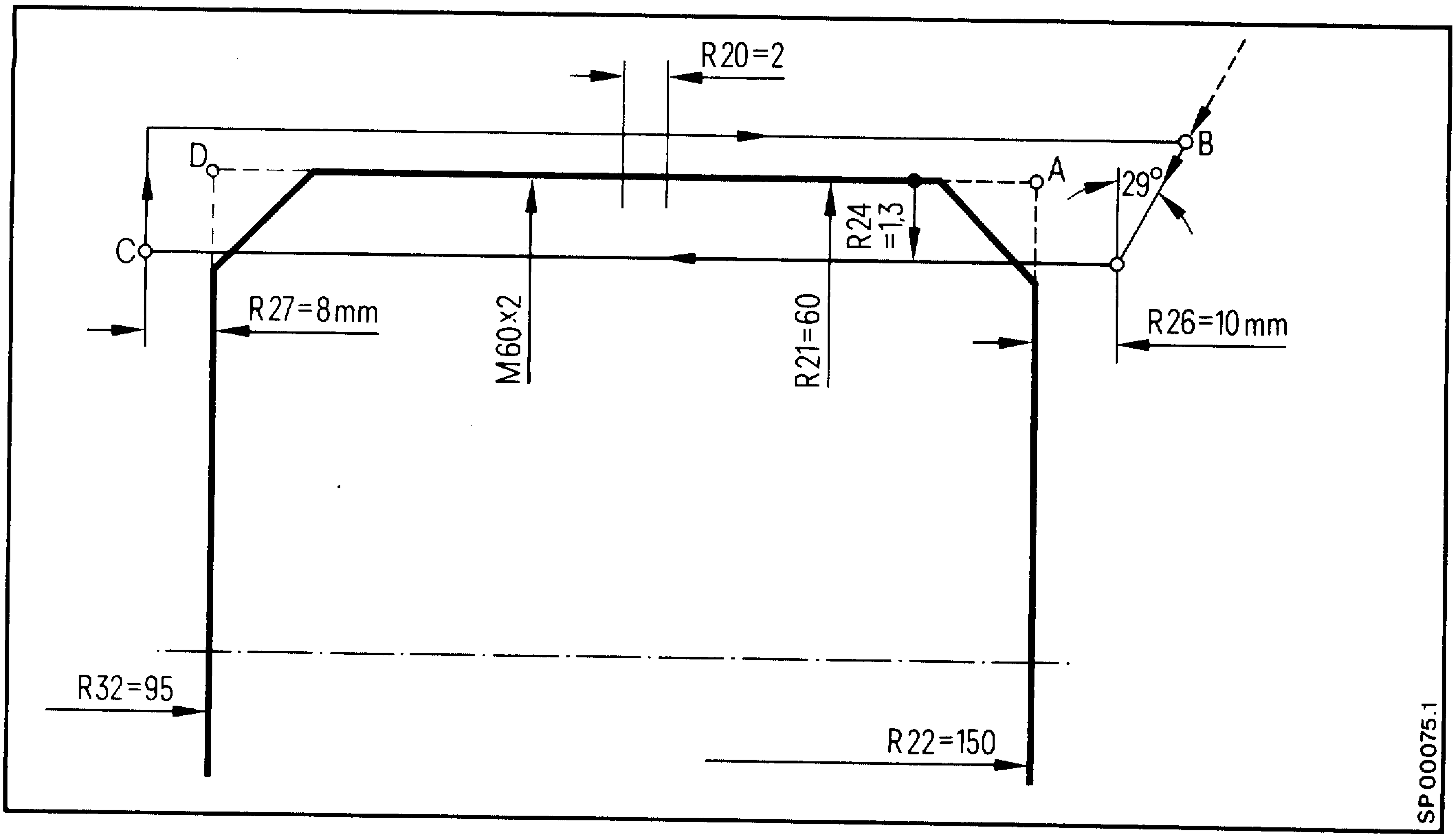
In the cycle the angle is used according to the machining direction.



R31 and R32 Thread end point

The parameters R31 and R32 represent the original end points of the thread.

Example: Outside thread



End Point X R31 = 60 mm = R21

End Point Z R32 = 95 mm

Number of idle passes R23 = 2

Finishing Cut Depth R25 = 0

Number of Roughing Cuts R28 = 7

Infeed Angle R29 = 29°

.

.

```
N130 T5   M15                          LF
N140 R20 2. R21 60. R22 150. R23 2. R24 -1.3
     R25 0. R26 10. R27 8.  R28 7. R29 29 LF
N150 R31 60. R32 95. L97                  LF
N160 GOO X...   Z...                      LF
```

7.1.5 L902 Four-point thread cutting cycle

With four-point thread cutting cycle L902, several chained thread blocks with different pitches can be produced at the same time.

Before cycle L902 is called, a value must be assigned to the following R parameters. Value assignment must take place in at least two blocks because a maximum of 10 R parameters per block are permitted:

R11: Starting point of the thread in X (absolute)
R12: First intermediate point of the thread in X (absolute)
R13: Second intermediate point of the thread in X (absolute)
R14: End point of the thread in X (absolute)

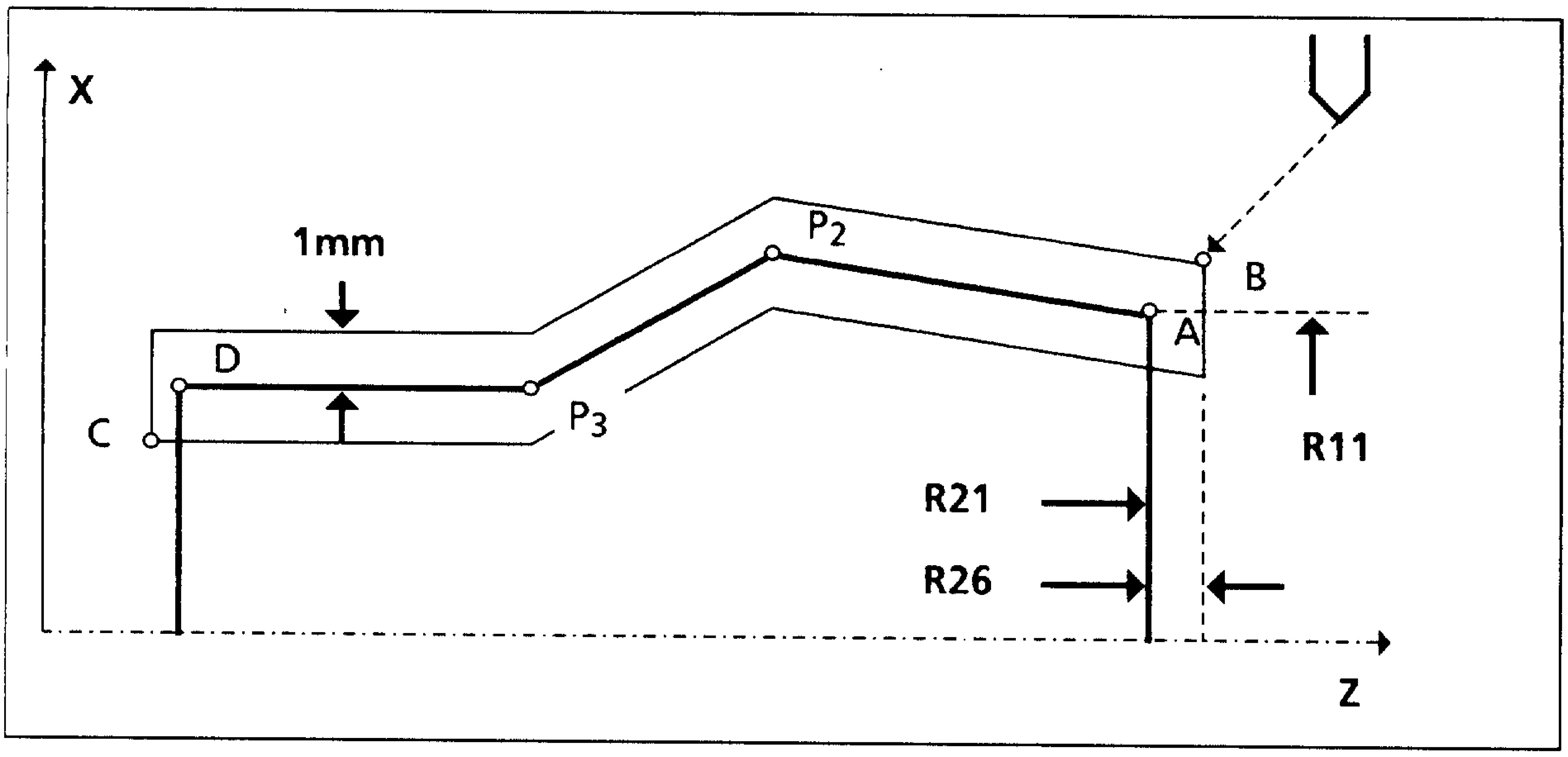
R21: Starting point of the thread in Z (absolute)
R22: First intermediate point of the thread in Z (absolute)
R23: Second intermediate point of the thread in Z (absolute)
R24: End point of the thread in Z (absolute)

R25: Finishing dimension
R26: Approach path
R27: Run-out path
R28: Number of roughing cuts
R29: Infeed angle

R35: Number of idle passes
R36: Thread depth (incremental) entered with sign, according to whether internal or external thread, + = internal thread, - = external thread

R41: Thread pitch 1
R42: Thread pitch 2
R43: Thread pitch 3

R11 and R21: Thread starting point



Parameters R11 and R21 represent the original starting point of the thread (A). The starting point of the thread cycle is at point B which precedes the thread starting point by parameter R26 (approach path). Starting point B is located in the X axis 1 mm above parameter value R11. This retraction plane is automatically created by the controller. The thread cycle can be called from any slide position; the infeed to point B takes place at rapid traverse.

R12 and R22: First intermediate point

Parameters R12 and R22 represent the first intermediate point of the thread (P1).

R13 and R23: Second intermediate point

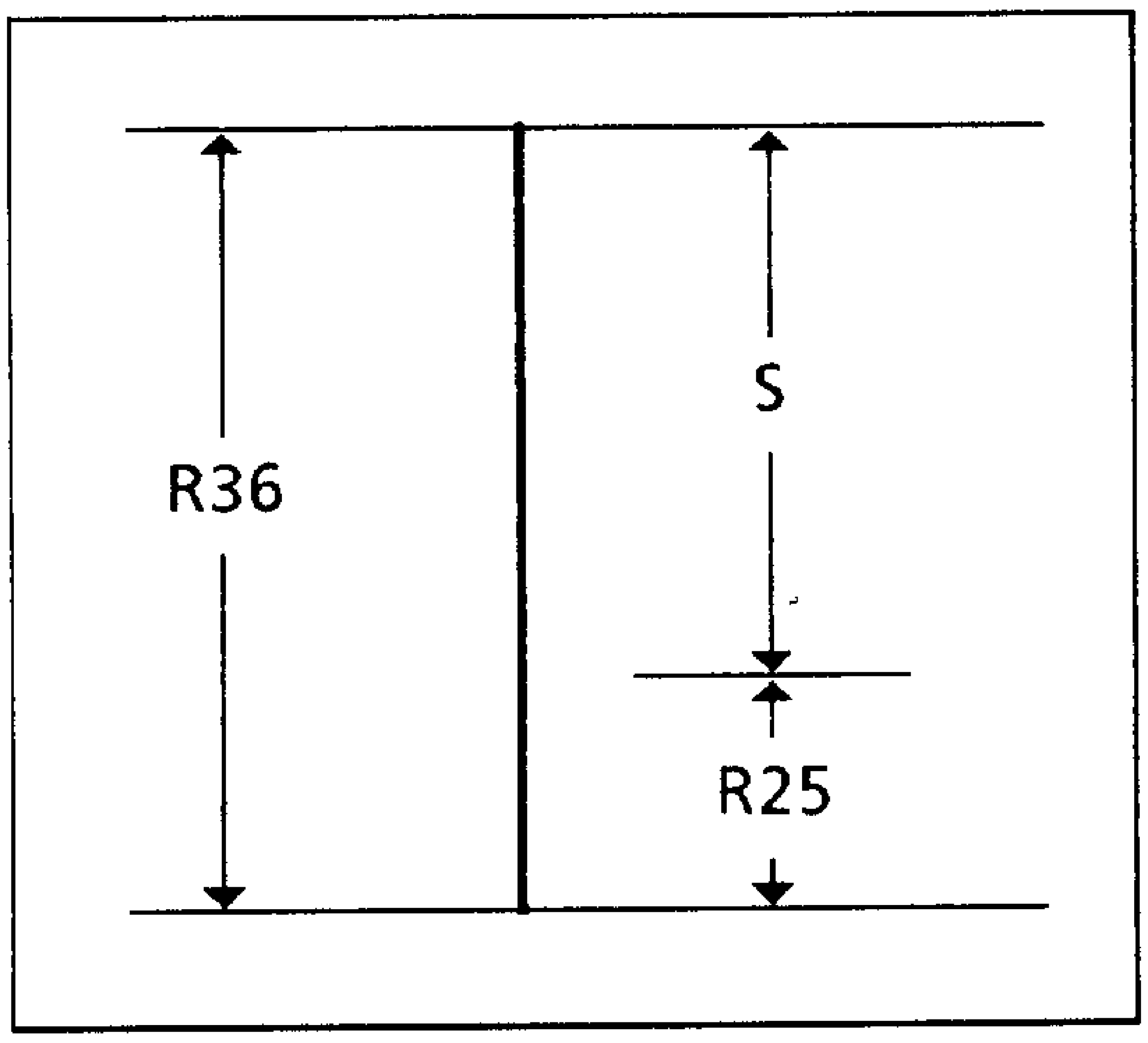
Parameters R13 and R23 represent the second intermediate point of the thread (P3). If a single intermediate point is to be entered, parameters R12 or R22 and R13 or R23 must be provided with the same values.

R14 and R24: Thread end point

Parameters R14 and R24 represent the original end point of the thread (D). If no intermediate point is to be entered, parameters R12 or R22, R13 or R23 and R14 or R24 must be provided with the same values.

R25: Finishing dimension

The finishing dimension is entered under R25. If a finishing dimension is programmed, the finishing dimension is subtracted from the thread depth and the remaining value is subdivided into roughing cuts. When the roughing cuts have ended, a finishing cut takes place first followed by the number of idle passes programmed under R36.



R36: Thread depth

R25: Finishing dimension

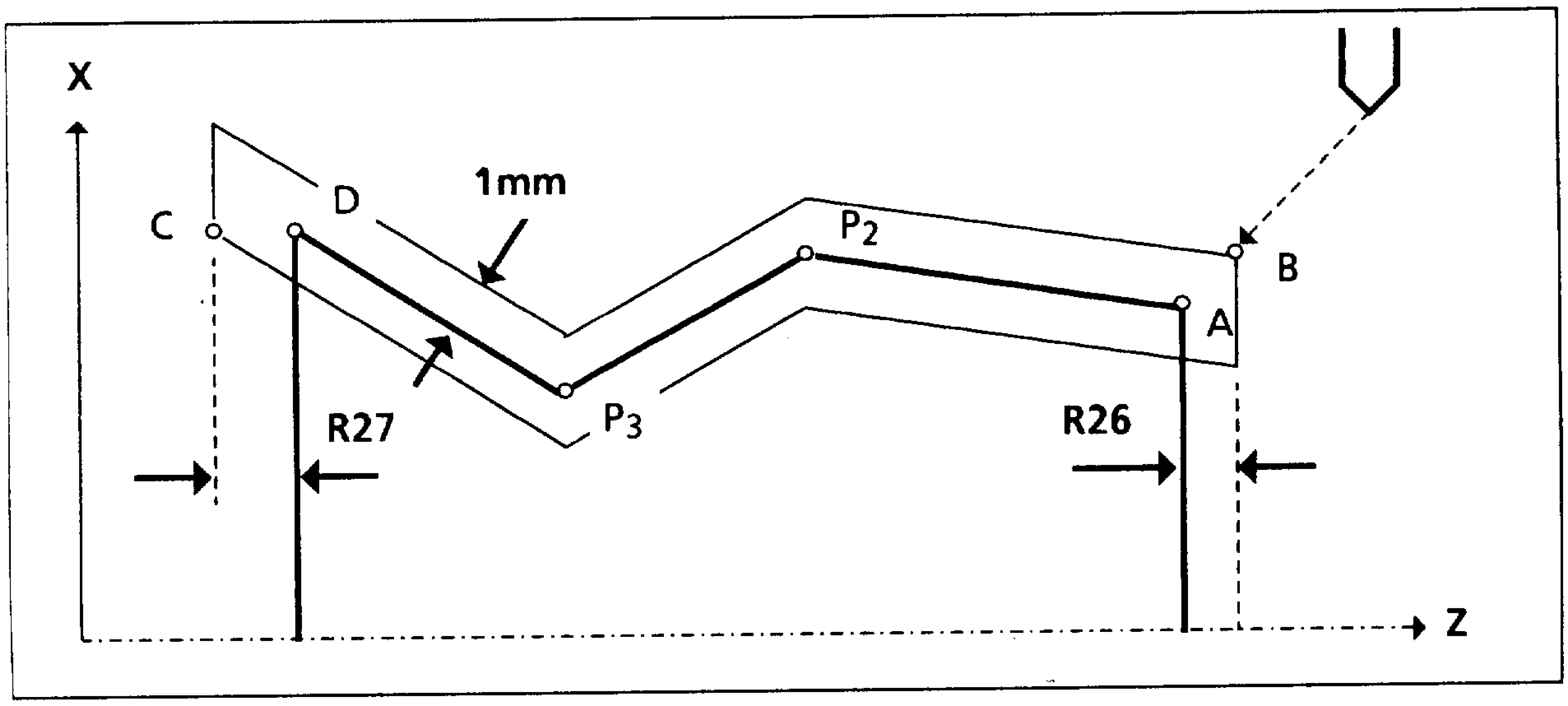
S : Roughing dimension

R26: Approach path and R27: Run-out path

The approach and run-out paths are entered without sign.
The parameters represent paraxial, incremental values:

- According to the Z axis for longitudinal threads;
- according to the X axis for transversal threads.

With a tapered thread, the controller converts the approach and run-out paths to the taper ratio and determines corner points B and C.



R28: Number of roughing cuts

The parameter value determines the number of thread roughing cuts.
The controller automatically computes the individual infeed depths with the same cut cross-section. This ensures that the cutting pressure remains the same from the first to the last roughing cut.
If R28 is entered with negative sign, the first roughing cut is halved.

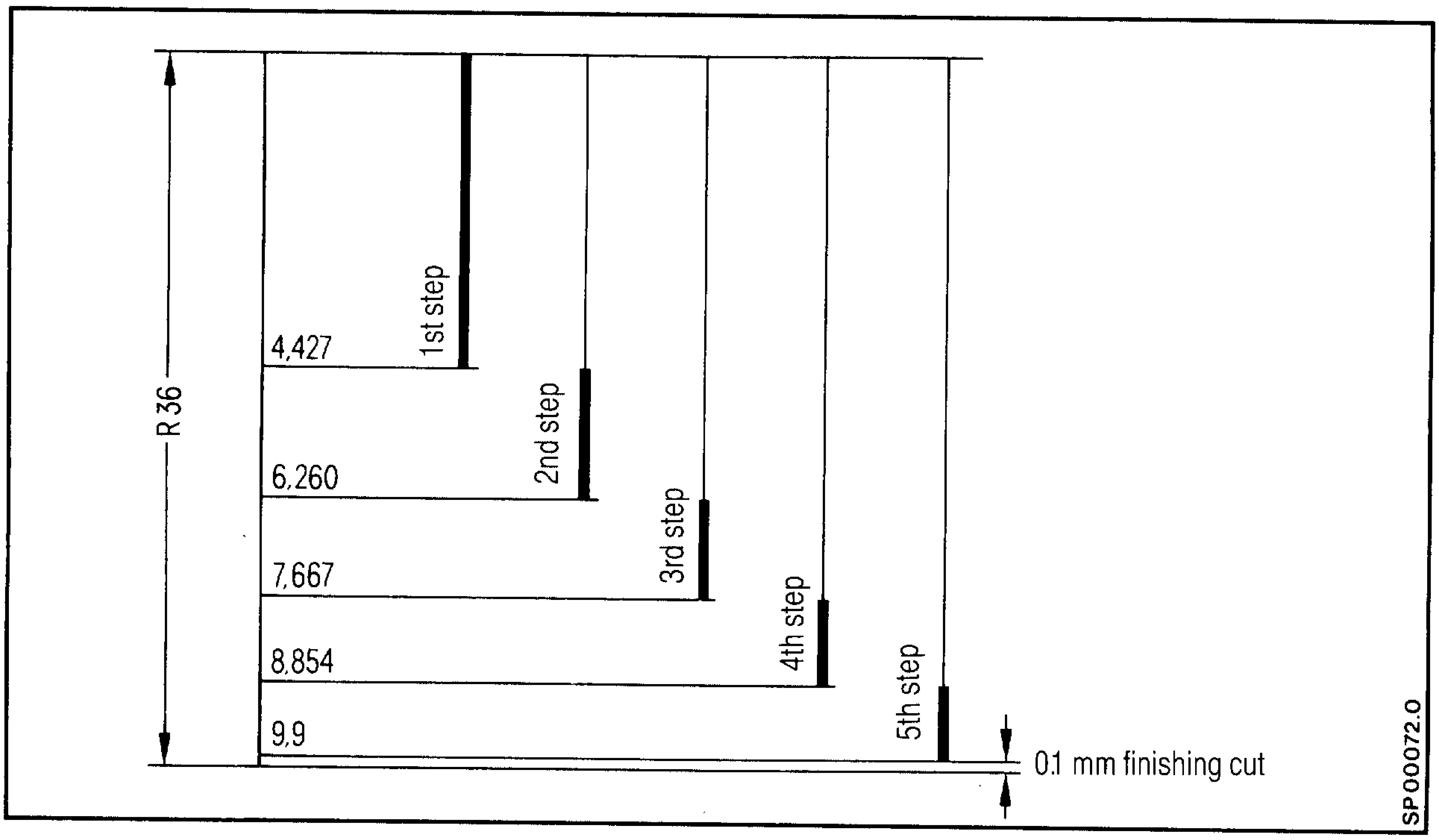
The current roughing cut t is computed with the following formula:

$$\Delta t = \frac{t}{\sqrt{R28}} \cdot \sqrt{i}$$

t = R 36 - R25
i = Current cut

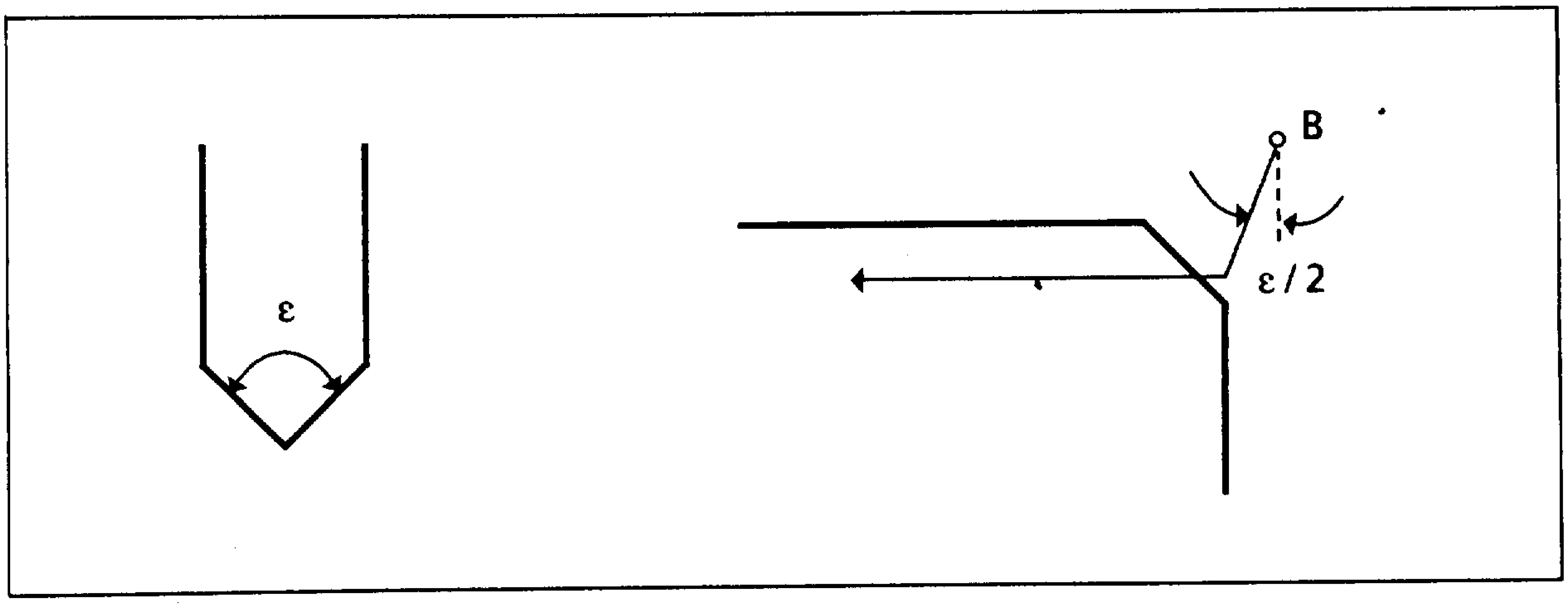
Example:

Thread depth t = 10
Number of roughing cuts = 5
Finishing dimension 0.1



R29: Infeed angle for longitudinal or transverse threads

Tool infeed is possible at any angle for longitudinal or transverse threads. The angle is entered without a sign.



R35: Number of idle passes

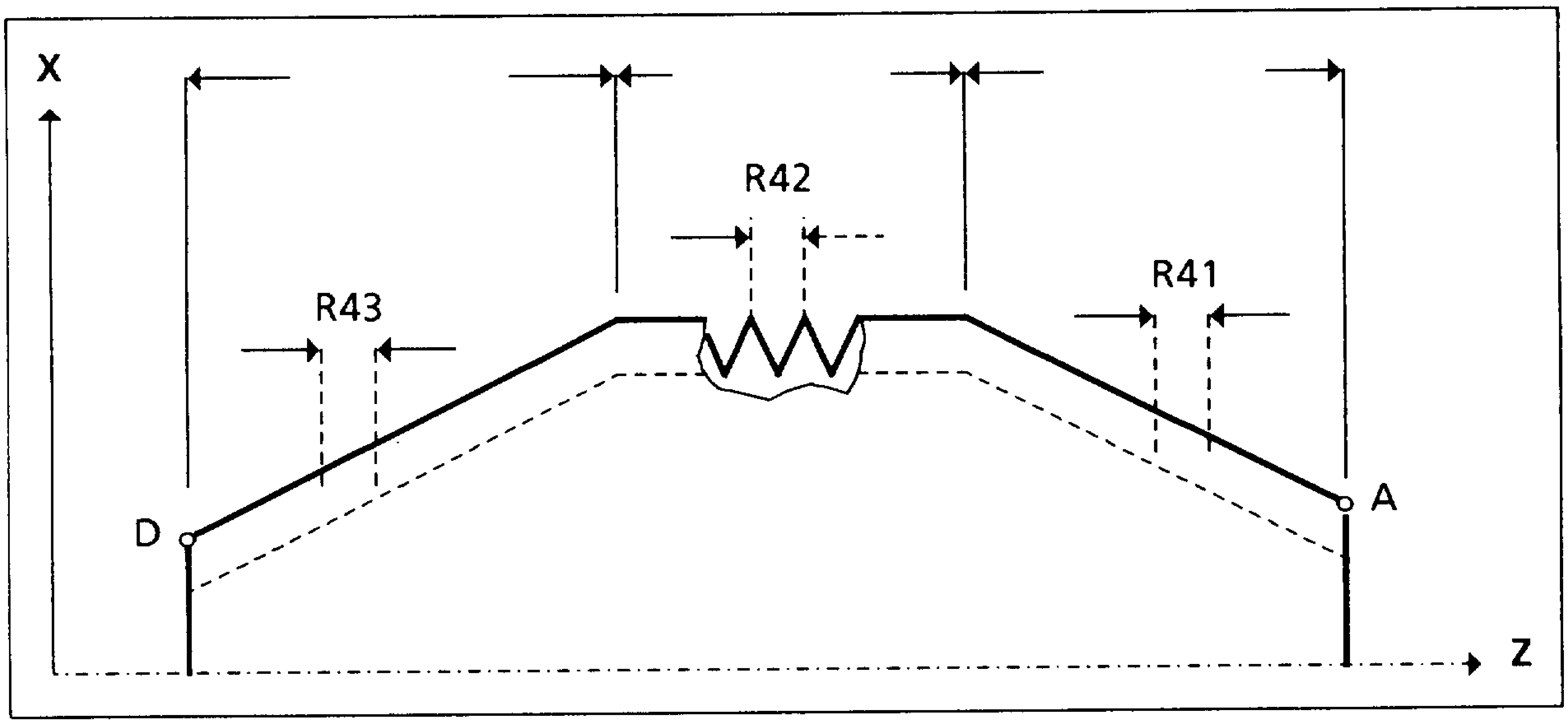
Any number of idle passes can be selected. The number is entered under parameter R35.

R36: Thread depth

The thread depth is entered under parameter R36. the sign determines the infeed direction, i.e. whether the thread is external or internal (+ internal thread, - external thread)

R41, R42, R43: Thread pitches

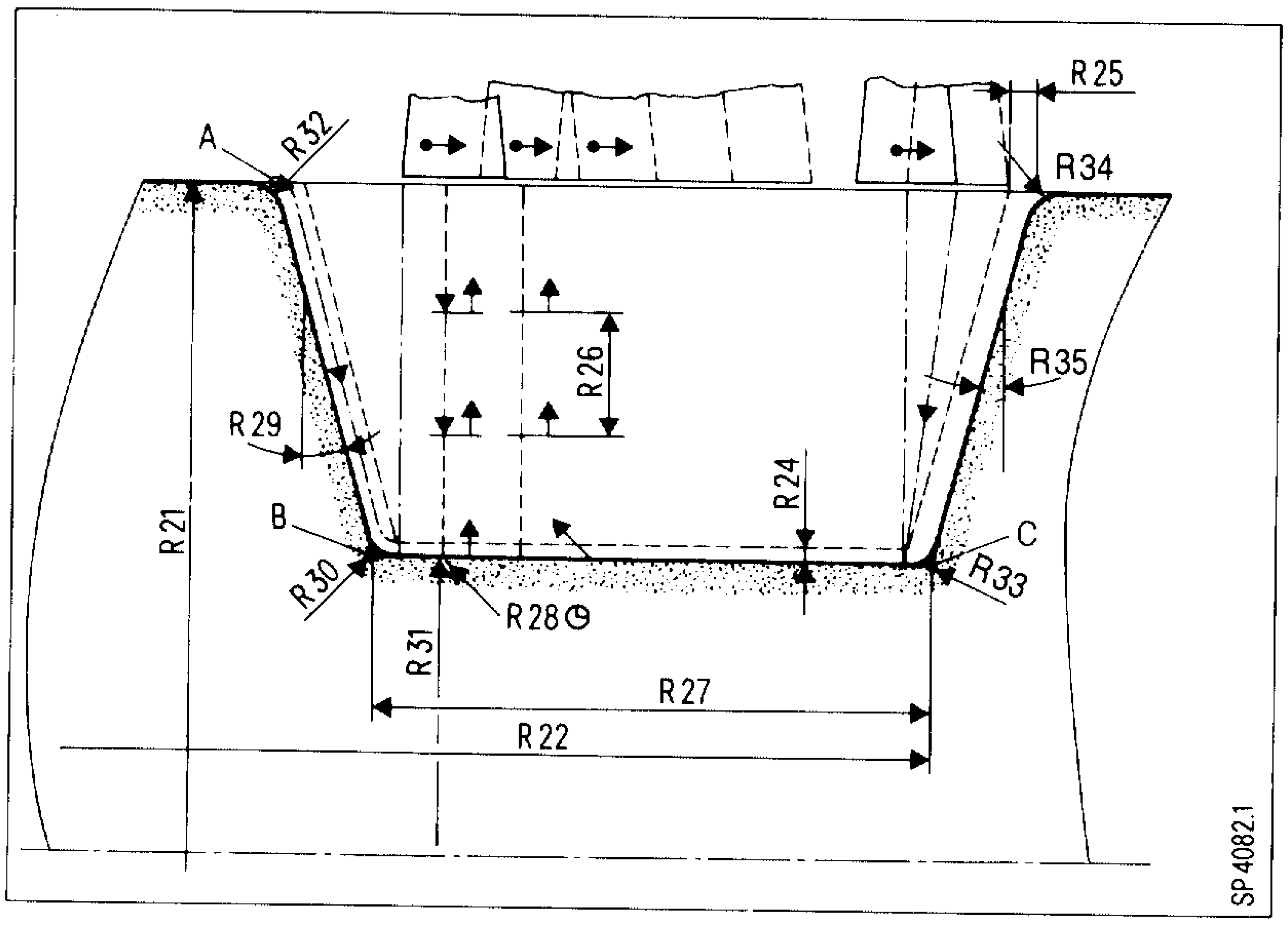
The parameters represent the value of pitch per element. The paraxial value is always entered without sign.



7.1.6 L903 Plunge-cutting cycle

The plunge-cutting cycle enables the machining of external and internal grooves. A value must be assigned to the following parameters before cycle L903 is called.

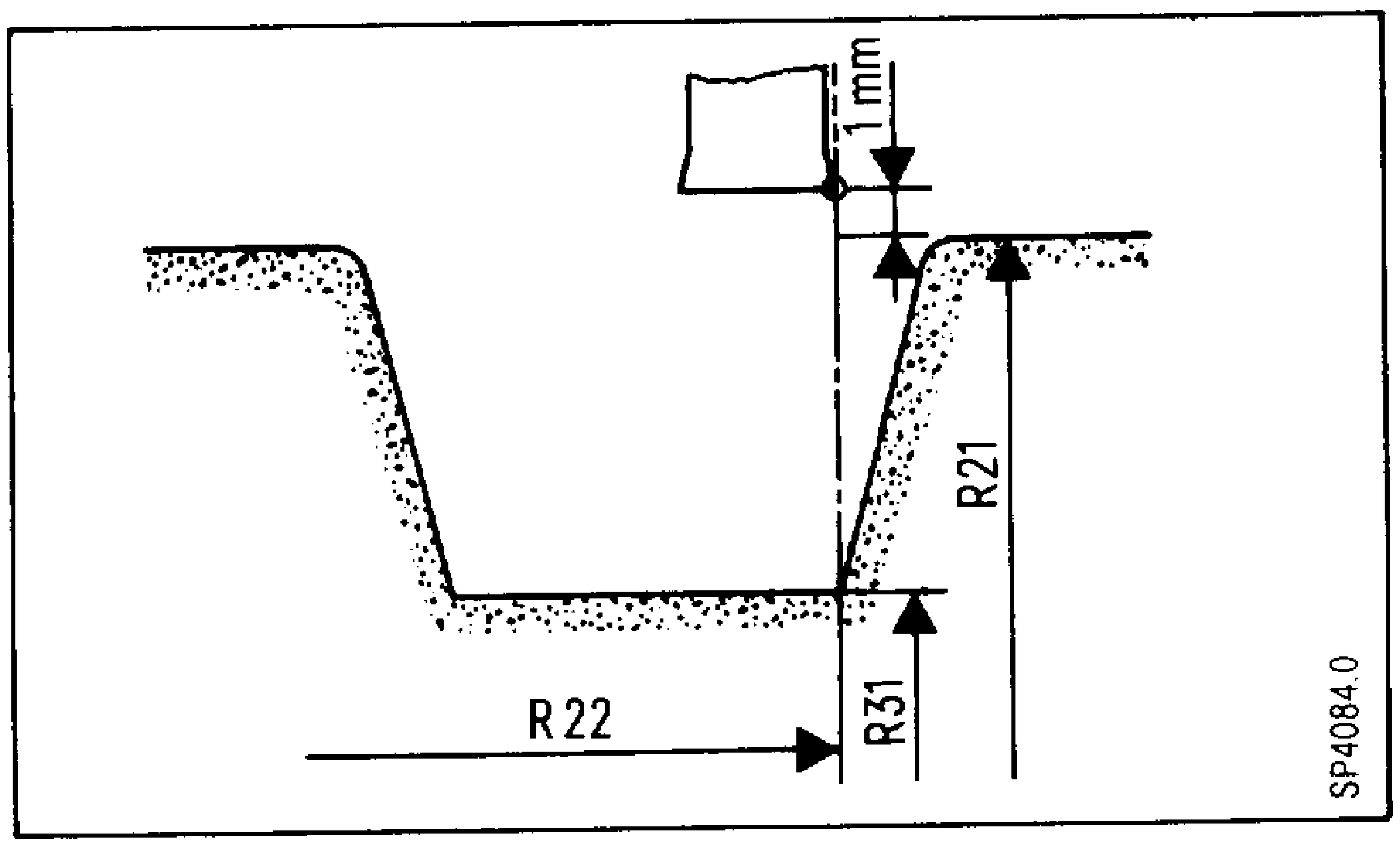
R29	Angle/left edge
R30	Radius or chamfer at base of cut/left edge
R31	Cutting diameter
R32	Radius or chamfer at edge of cut/left edge
R33	Radius or changer at base of cut/right edge
R34	Radius or chamfer at cutting edge/right edge
R35	Angle/right edge



Before the plunge-cutting cycle is called in a machining program, the tool offset of the first cutter must be selected. The tool offset for the second cutter of the plunge-cutting tool must be stored in the tool offset memory under the next higher offset number. If the tool offset for the first cutter is T 01, the tool offset for the second cutter must be T 02.

R21/R22 Diameter (outer or inner) / starting point

Parameters R21 and R22 determine the starting point. The controller automatically approaches the point programmed by R21 and R22. With an outer groove the Z direction is first traversed, and with an inner groove the X direction is first traversed. In the X direction a safety clearance of 1 mm is maintained.



R23 Control parameter

The control parameter determines the type of groove: outer or inner groove, starting right or left.

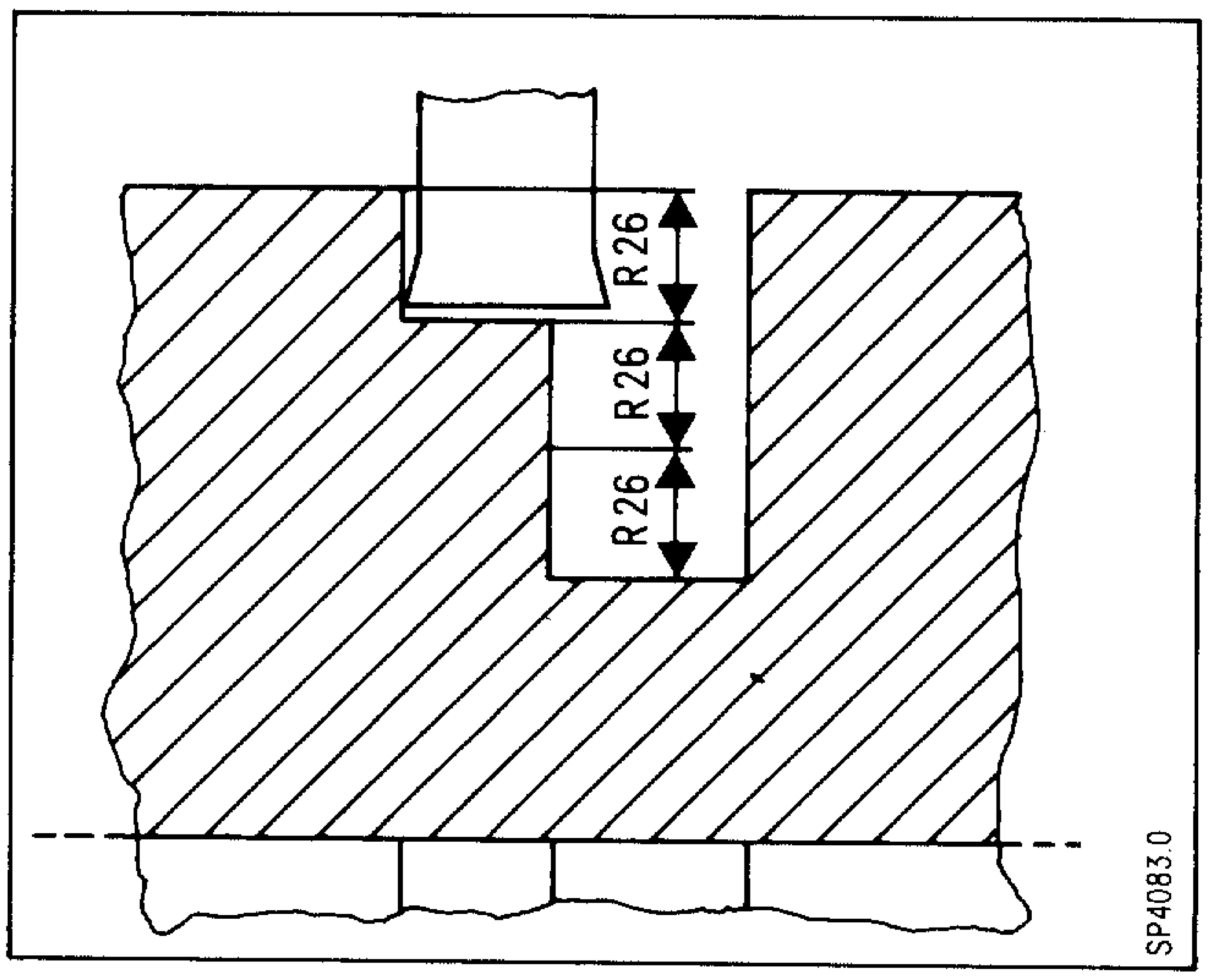
Longitudinal plunge cutting			Face plunge cutting		
R23	1	outside / left	R23	5	right / outside
R23	2	outside / right	R23	6	right / inside
R23	3	inside / left	R23	7	left / outside
R23	4	inside / right	R23	8	left / inside

R24/25 Finishing allowance

The finishing allowance entered can be of different magnitude for the X and Z directions. The same plunge-cutting tool is used for machining. The finishing allowance chosen should be great enough so that radii or chamfers at the groove base are not damaged during plunge-cutting.

R26 Infeed in X

The maximum incremental infeed path is determined with the infeed (R26). For chip breaking, the tool is retracted by 1 mm.



R27 Groove width

If the groove is wider than the plunge-cutting tool, the infeed in the Z direction is subdivided into sections of the same size. The maximum infeed in Z depends on the tool width. It is 95 % of the tool width after deduction of the cutter radii. This ensures an overlap in cutting.

R28 Dwell time at depth

The dwell time chosen must be great enough so that at least one spindle revolution takes place.

R29/R35 Angle

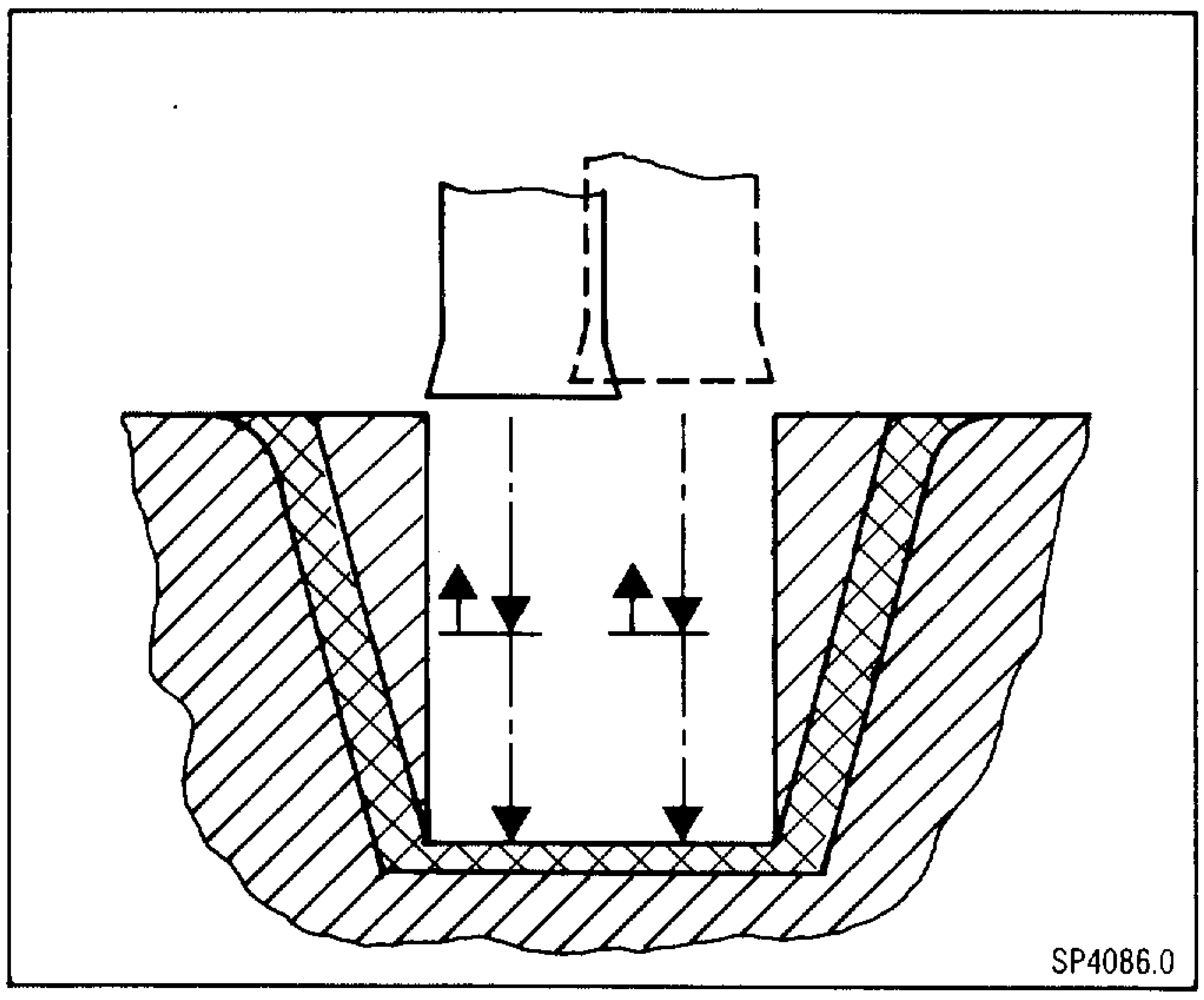
The edge angle R29/R35 can be any angle between 0° and 89°.

R30/R32/R33/R34 Radius or chamfer

Via parameters R30/R32, radii or chamfers at the groove base and/or groove edge can be inserted.
+ Sign radius
- Sign chamfer

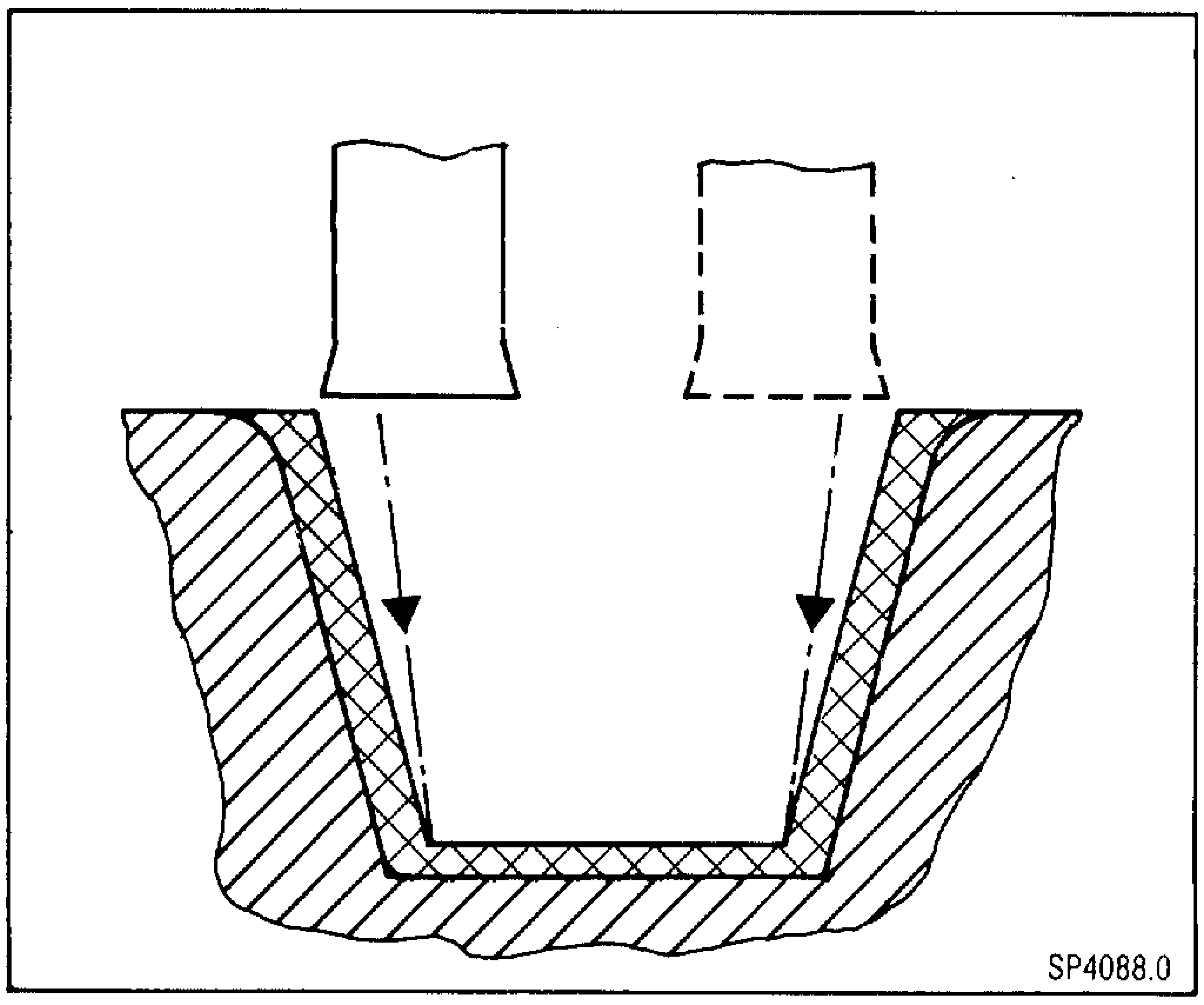
Machining sequence

1. Radial grooves
Groove perpendicular to the axis of rotation in one or more cuts. Before withdrawal from the groove, a clearance of 1 mm is allowed from the second cut in the Z direction.

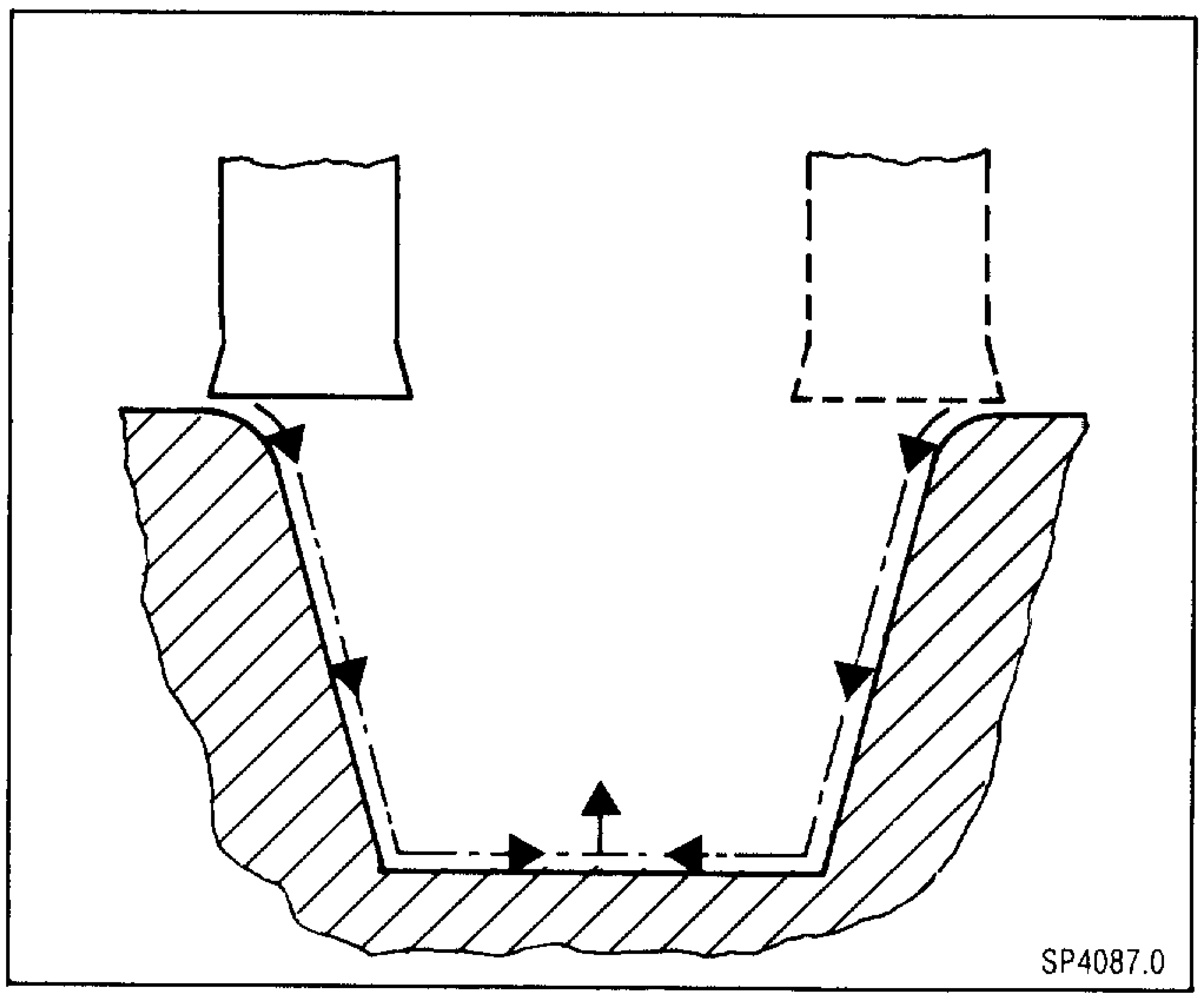


2. Machining flanks
Machining of flanks, where an angle has been programmed with R29.

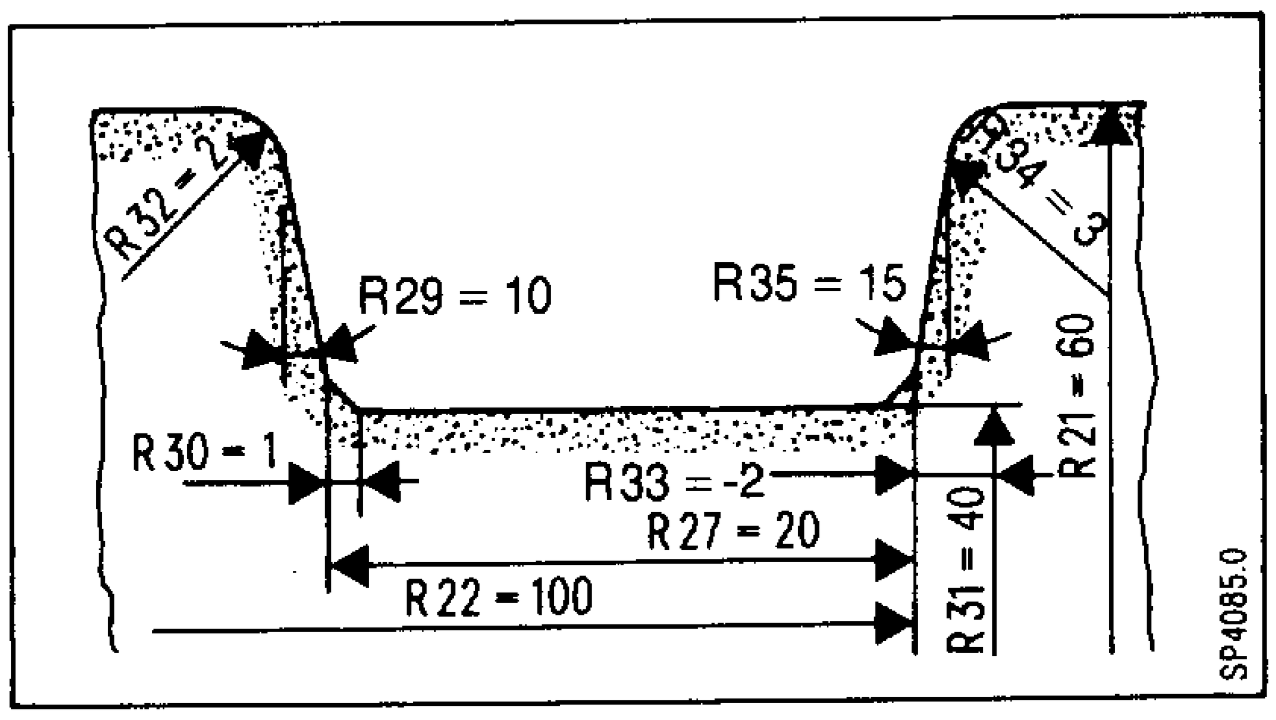
The infeed in the Z direction takes place in several steps if the tool width is less than the flank width.



3. Finishing
Machining the finishing allowance parallel to the contour, down to the midpoint of the groove.



Example:

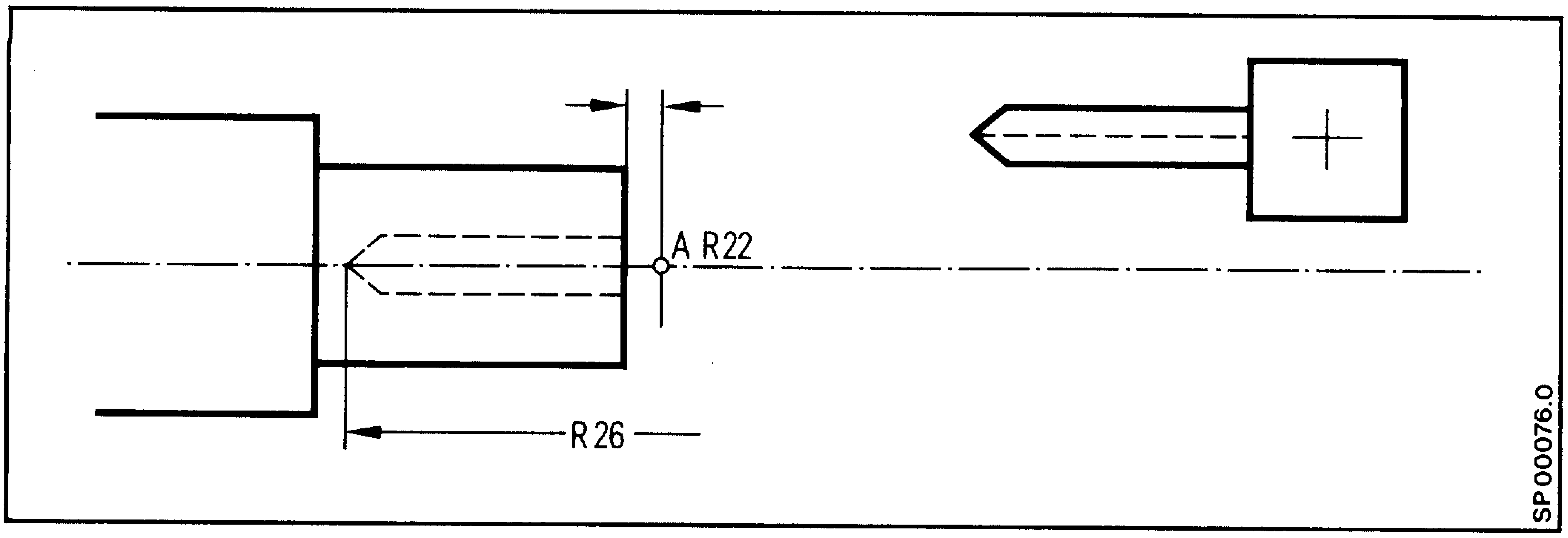


.
.
.

```
N5  R21 60. R22 100. R23 2 R24 1. R25 2.
    R26 10. R27 20. R28 0.5 R29 10. R30 -1. LF
N10 R31 40. R32 2. R33 -2. R34 3. R35 15. L90301 LF
```

.
.
.

7.1.7 L98 Deep-boring cycle



A value must be assigned to the following parameters before the call for cycle L98.

R22 Starting point in the Z direction, absolute values are entered.

R24 Amount of degression, entered incrementally without sign

R25 First boring depth, entered incrementally without sign

R26 Final boring depth (absolute)

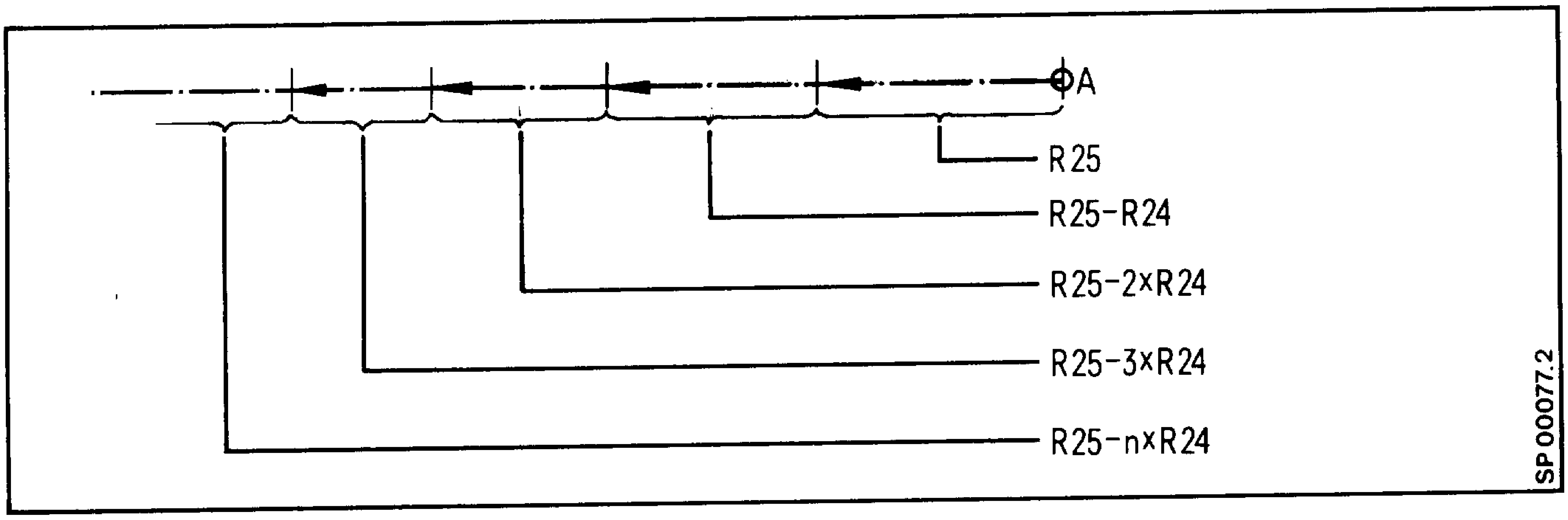
R27 Dwell time at starting point (for chip removal)

R28 Dwell time at boring depth (chip breakage)

R26 Final boring depth

The relevant boring depth is reduced degressively by the constant amount of degression, until final point R26 is reached. If, however, the boring depth reaches the calculated amount of degression, it is kept constant at this value.
If the residual infeed is less than twice the amount of degression, the residual amount is halved. The last two infeeds are executed with this halved value. This prevents the last infeed from being executed with an insufficient amount. This computation alway results in a minimum infeed of half the amount of degression.

The drill tip is at the starting point A again at the end of the cycle.



Example:

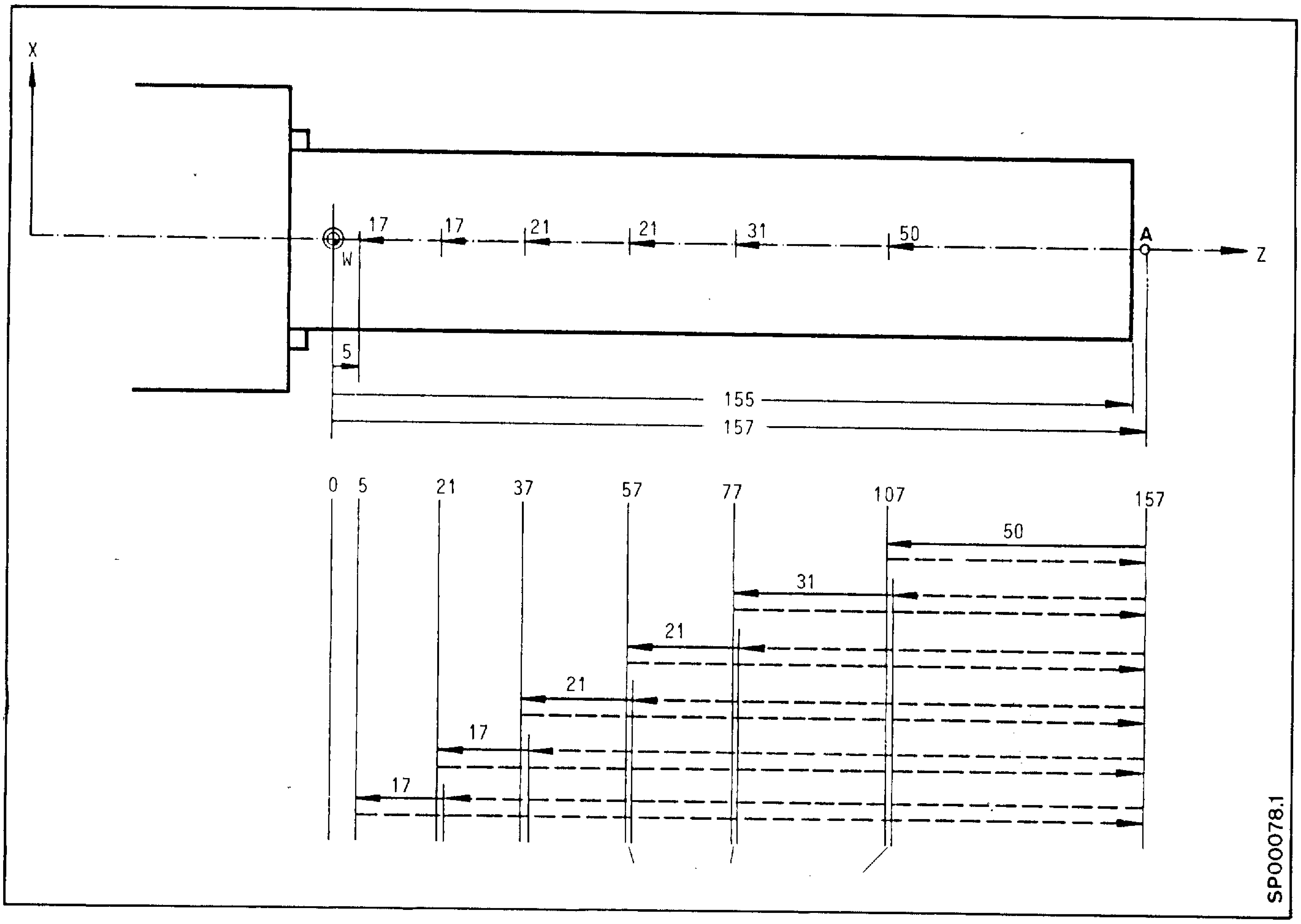
Starting point	Z = 157 mm	R 22 157.
Amount of degression	20 mm	R 24 20.
First boring depth	50 mm	R 25 50.
Final boring depth	5 mm	R 26 5.
Dwell time at starting point	2 s	R 27 2.
Dwell time at boring depth	1 s	R 28 1.

Call in the program:

.
.
.

```
N10  R22. 157. R24 20. R25 50. R26 5. R27 2. R28 1. LF
N15  L98         F...  S...  M...                   LF
```

.
.
.



___ Feed

--- Rapid traverse

L903 Plunge-cutting cycle

L94, L95 Machining cycle

L96, L97 Thread cycle

L98 Deep-hole drilling (Basic Version 4B, 4C)

E 822 3T4C GE548811.9202.10 (30.06.88)

```
%SP
L90300 (30.06.88 G4C.01 EINSTECHEN)
R99 8 R80 0 R81 1 R90 2 R86-1 R97 4 R48 1
@02 1 R23 R97
R48 0
N1 @02 780 R23 R99
@00 R23
N0 @00 780
N1 R97-1
@00 15
N2 R97 1
N15 @03 780 R31 R21
@00 17
N3 R97-1
@00 16
N4 R97 1
N16 @03 780 R21 R31
@00 17
N5 R97 1
@00 17
N6 R97-1
@00 17
N7 R97 1
@00 17
N8 R97-1
N17 R79 0 R25
R98 0 R25
R74 0 R24
R99 2409
@29 38910 R99
R89 R81
@31
R99 3001
@29 39619 R99
@01 10 R96 R80
R81 0.03937
N10 R99 6408 @29 39510 R99
R95 -R96 R94 1
@01 15 R95 R80
R94 25.4 R93 25.4
@02 15 R95 R80
R94 1
R94 /R93
N15 R99 1 @29 16020 R99
R66 0 R81
R59 0 R60
R81 1
R59 R81
R50 1
@01 20 R48 R80
R50 0 R89 R89 1
N20 R99 2000
@01 25 R48 R80
R99 1000
N25 R99 R60
@29 17501 R99
R99 2000
@01 30 R48 R80
R99 1000
N30 R99 R59
@29 15101 R99
```

```
@03 35 R51 R75
R52 0 R75
R52 -R51
R85 0 R60
R60 0 R59
R59 0 R85
@00 40
N35 R52 0 R51
R52 -R75
N40 R61 1 R92 2
R52 .R94
@01 45 R48 R80
R85 0 R60
R60 0 R59
R59 0 R85
R61 2 R92 1
N45 @20 @90 R61
@20 @91 R92
R84 90 R29
R84 /R90
R51 90 R84
@15 R84
R68 0 R30
R68 /R84
@15 R51
R68 .R51
@03 50 R30 R80
R68 0 R30
R68 .R86
N50 R67 0 R68
@03 65 R80 R24
@03 55 R30 R80
R72 0 R24
R72 /R51
R72 .R84
@00 60
N55 R72 0 R30
R72 .R30
R75 0 R30
R75 -R24
R51 0 R75
R75 .R51
R72 -R75
@10 R72
N60 R68 -R72
N65 @03 70 R25 R68
R79 0 R68
N70 R84 90 R35
R84 /R90
R51 90 R84
@15 R84
R77 0 R33
R77 /R84
@15 R51
R77 .R51
@03 75 R33 R80
R77 0 R33
R77 .R86
N75 R65 0 R77
@03 90 R80 R24
@03 80 R33 R80
```

```
R91 0 R24
R91 /R51
R91 .R84
@00 85
N80 R91 0 R33
R91 .R33
R75 0 R33
R75 -R24
R51 0 R75
R75 .R51
R91 -R75
@10 R91
N85 R77 -R91
N90 @03 95 R25 R77
R98 0 R77
N95 @01 100 R48 R81
@03 105 R89 R90
R21 /R90
R31 /R90
@00 105
N100 @03 105 R50 R90
R22 /R90
N105 R78 0 R31
R74 .R89
@03 110 R31 R21
R78 0 R31
R78 R74
R87 1
@00 115
N110 R78 0 R31
R78 -R74
R87-1
N115 R88 0 R27
R74 0 R79
R91 0 R98
@01 120 R48 R80
R88 .R86
R88 .R50
R91 .R86
R91 .R50
R74 .R86
R74 .R50
N120 R83 0 R22
R70 0 R22
@02 125 R97 R80
R83 R88
R70 R74
R71 0 R83
R71 -R91
@00 130
N125 R83 -R88
R70 -R91
R71 0 R83
R71 R74
N130 R53 0 R70
R53 -R71
@02 135 R53 R80
R53 .R86
N135 R99 3000
R99 R60
@29 17501 R99
```

```
R75 .R94
R85 0.95
R55 0 R52
R55 -R75
@03 770 R80 R55
R55 .R85
R51 0 R55
R56 0 R53
R56 /R50
R56 -R52
@02 770 R52 R27
@03 145 R80 R56
R57 0 R56
R57 /R55
R99 0
N140 R57 -R81
R99 R81
@02-140 R57 R80
R57 0 R99
R55 0 R56
R55 /R57
R55 .R50
@00 150
N145 R57 0
N150 R56 0 R21
R56 -R31
R56 .R87
R56 /R89
R58 0 R56
R58 /R26
R99 0
N155 R58 -R81
R99 R81
@02-155 R58 R80
R58 0 R99
R62 0 R21
R99 0 R66
R66 .R89
R63 0 R21
@03 160 R21 R31
R66 .R86
N160 R63 R66
R66 /R89
R54 0 R56
R54 /R58
R54 .R89
R64 0 R70
T R60
@03 165 R97 R80
T R59
N165 @03 170 R21 R31
G90 G @90 R63
R54 .R86
N170 G90 G @91 R64
@90 R63
R68 0
N175 @02 225 R68 R57
R69 0
@91 R64
N180 @03 205 R69 R58
R69 R81
```

```
R62 -R54
@03 185 R69 R58
G1 G90 @90 R62
@00 195
N185 G1 G90 @90 R78
G04 X R28
@01 195 R68 R80
@02 190 R68 R81
R99 .R97
@01 190 R48 R80
R99 .R86
N190 G91 G @90 R66 @91 R99
N195 @03 200 R69 R58
G91 G @90 R66
N200 @00-180
N205 G90 G @90 R63
R68 R81
@02 215 R68 R81
@01 210 R48 R80
R55 .R86
N210 R55 .R97
N215 R64 -R55
R62 0 R21
N220 @00-175
N225 R85 90 R29
@15 R85
R84 0 R29
@15 R84
R73 0 R84
R73 /R85
R77 0 R73
R88 0 R67
R88 .R85
@03 230 R88 R24
R88 0 R24
N230 R91 0 R25
R91 .R84
R91 R88
R77 .R88
R85 .R25
R49 0 R77
R49 -R85
@03 235 R24 R88
@03 240 R30 R80
N235 R49 0 R77
R49 -R25
R91 0 R88
N240 R88 0 R79
R88 R49
R93 0
@02 245 R80 R30
R93 0 R30
R93 -R24
N245 R92 0 R56
R92 .R73
R53 0 R92
R53 R79
R53 -R25
R85 90 R35
@15 R85
R84 0 R35
```

```
@15 R84
R73 O R84
R73 /R85
R77 O R73
R99 O R65
R99 .R85
@03 250 R99 R24
R99 O R24
N250 R74 O R25
R74 .R85
R74 R99
R77 .R99
R85 .R25
R61 O R77
R61 -R85
@03 255 R24 R99
@03 260 R33 R80
N255 R61 O R77
R61 -R25
R74 O R99
N260 R99 O R98
R99 R61
R85 O
@02 265 R80 R33
R85 O R33
R85 -R24
N265 R95 O R97
R84 O R56
R84 .R73
R78 O R84
R78 R98
R78 -R25
@01 270 R48 R80
R92 .R86
R92 .R50
R84 .R86
R84 .R50
R95 .R86
N270 R57 O R53
R57 /R51
R82 O
N275 R57 -R81
R82 R81
@02-275 R57 R80
R57 O R82
R64 O R71
R55 O R53
R55 /R57
R71 O R78
R71 /R51
R82 O
N280 R71 -R81
R82 R81
@02-280 R71 R80
R71 O R82
R82 O R78
R82 /R71
@02 295 R80 R97
N285 R78 O R92
@02 290 R78 R80
R78 .R86
```

```
N290 R78 R67
@03 340 R25 R78
R75 0 R93
R73 0 R67
R73 -R79
R72 0 R91
R69 0 R49
R66 0 R88
R62 0 R88
R58 0 R67
R56 0 R30
T R59
R78 0 R57
R54 0 R55
@00 305
N295 R78 0 R84
@02 300 R78 R80
R78 .R86
N300 R78 R65
@03 340 R25 R78
R72 0 R74
R73 0 R65
R73 -R98
R75 0 R85
R69 0 R61
R66 0 R99
R62 0 R99
R58 0 R65
R56 0 R33
T R60
R78 0 R71
R54 0 R82
R54 .R86
N305 @03 350 R80 R90
R68 0
R51 0 R64
R96 0 R54
@02 310 R96 R80
R96 .R86
N310 @01 315 R48 R80
R54 .R86
R54 .R50
N315 @03 340 R68 R78
R64 -R54
R66 -R96
@03 325 R80 R66
@02 320 R80 R56
R73 R96
R76 0 R73
R76 .R73
R77 0 R75
R77 .R75
R77 -R76
@10 R77
R77 .R86
R77 R56
@00 330
N320 R53 0 R72
R53 -R24
R73 R96
R77 0 R73
```

```
R77 /R62
R77 .R53
@00 330
N325 R53 0 R69
R53 .R95
R53 .R50
R51 0 R83
R51 -R53
R53 0 R72
R53 .R87
R53 .R89
R53 R31
@00 335
N330 @01 335 R58 R80
R51 -R54
R77 .R87
R77 .R89
R53 0 R31
R53 R77
N335 G G90 @91 R64
G1 @90 R21
@91 R51 @90 R53
G04 X R28
G G90 @90 R63
R68 R81
@00-315
N340 R83 0 R22
R95 .R86
R90 -R81
R64 0 R70
@03 350 R80 R90
@02-285 R80 R97
@00-295
N350 @02 355 R79 R80
@02 355 R24 R80
@01 352 R32 R80
@00 355
N352 @01 800 R34 R80
N355 @02 360 R31 R21
R85 42 R51 41
@00 365
N360 R85 41 R51 42
N365 R61 .001 R76 .1 R77 .1 R90 2
@03 370 R32 R80
R76 -R32
@00 375
N370 R76 R32
N375 @03 380 R34 R80
R77 -R34
@00 385
N380 R77 R34
N385 @01 390 R48 R80
R76 .R86
R61 .R86
R77 .R86
R77 .R50
R76 .R50
N390 @02 395 R24 R80
R54 0 R53 0
@00 415
N395 R53 0 R27
```

```
R53 /R90
R82 0 R65
@03 400 R82 R25
R82 0 R25
N400 R54 0 R53
R54 -R82
R82 0 R67
@03 405 R82 R25
R82 0 R25
N405 R53 -R82
R83 0 R53
R83 -R52
R82 0 R53
R53 0 R54
@03 410 R83 R80
R53 R83
N410 R83 0 R54
R83 -R52
R54 0 R82
@03 415 R83 R80
R54 R83
N415 @02 770 R80 R53
@02 770 R80 R54
R83 0 R27
R74 0 R75 0
R99 3000 R60
@29 19601 R99
R96 .R94
@02 425 R33 R80
R91 90 R35
R91 /R90
R78 90 R91
R78 /R90
@01 420 R33 R80
R91 0 R78
N420 R74 0 R96
R78 90 R91
@15 R91
@15 R78
R74 /R91
R74 .R78
R74 .R50
N425 R99 3000 R59
@29 19801 R99
R98 .R94
@02 435 R30 R80
R91 90 R29
R91 /R90
R78 90 R91
R78 /R90
@01 430 R30 R80
R91 0 R78
N430 R75 0 R98
R78 90 R91
@15 R91
@15 R78
R75 /R91
R75 .R78
R75 .F50
N435 R99 0
@02 436 R74 R53
```

```
R74 0
N436 @02 438 R75 R54
R75 0
N438 @01 440 R48 R80
R83 .R86
R53 .R86
R54 .R86
R74 .R86
R75 .R86
R67 .R86
R65 .R86
R83 .R50
N440 R70 0 R22 R71 0 R22
@02 445 R80 R97
R70 R99
R70 R84
R71 R99
R64 0 R70
R64 R77
R72 0 R71
R65 .R50
R72 -R65
T R60
R87 0 R85
R82 0 R61
R82 R53
R82 .R50
R72 -R82
R72 -R74
R95 0 R72
R95 R96
R88 0 R34
R89 0 R33
@00 450
N445 R70 -R99
R70 -R92
R71 -R99
R64 0 R70
R64 -R76
R72 0 R67
R72 .R50
R72 -R99
R72 R22
R82 0 R61
R82 R54
R82 .R50
R72 R82
R72 R75
R95 0 R72
R95 -R98
T R59
R87 0 R51
R88 0 R32
R89 0 R30
N450 @03 800 R80 R90
G40 G G90 @90 R63 @91 R64
G R87 G1 @90 R21
@90 R21 @91 R70 B R88
@90 R31 @91 R71 B R89
@91 R72
@02 455 R24 R80
```

```
G04 X R28
N455 G G40 @90 R63 @91 R95
R90 -R81
R99 0 R83
R97 .R86
@03 800 R80 R90
@00-440
N770 M00
@31
(TO DATA)
@00 800
N780 M00
@31
(INPUT R-PARAM.)
N800 M17
M02
```

```
%SP
L9400 (30.06.88 G4C.01 GERADE/KREIS)
R89 0 R56 0 R66 1 R67 1 R74 0 R15
R82 0 R11 R84 0 R13 R85 0 R23 R86 0 R24
@31
R90 2409 @29 36110 R90
R61 R66 G40 G90
R79 1
@01 22 R21 R79
@02 23 R61 R79
R79 .5
@00 23
N22 @01 23 R61 R79
R79 2
N23 R10 .R79
R12 .R79
R14 .R79
R51 180 R55 360
@01 3 R66 R22
@01 3 R56 R22
R53 0 R12 R54 0 R13 R57 0 R23 R58 0 R24
R53 -R10
R54 -R11
@18 R53
@18 R57
R53 -R57
@03 2 R53 R56
R53 R55
N2 R22 2
@02 3 R51 R53
R22 3
N3 R87 0 R22 R79 10
R73 0 R14 R81 0 R10 R83 0 R12 R65 0 R20
N4 R65 -R79
@02-4 R65 R79
R79 1
@01 20 R65 R79
R79 3
@01 20 R65 R79
@02 21 R11 R13
@02 24 R74 R11
N25 R74 0 R11
@00 24
N21 @02-25 R74 R11
@00 24
N20 R79 3
@03 26 R65 R79
@02 24 R10 R73
N27 R73 0 R10
@00 24
N26 @02-27 R10 R73
N24 R79 2
@02 5 R65 R79
R66-1
N5 @02 6 R82 R74
@02 6 R84 R74
R67-1
N6 R68 0 R66
R66 .R61 R69 0 R16
R69 .R66 R70 0 R17
R70 .R67
```

```
R73 -R69
R74 -R70
R81 -R69
R82 -R70
R83 -R69
R84 -R70 R79 1
R76 0 R18 R77 0 R18
@01 7 R79 R65
R79 3
@01 7 R79 R65
R71 0 R84 R75 0 R74
R77 .R67 R68 0 R67
R96 1
@00 8
N7 R71 0 R83 R75 0 R73
R76 .R61
R77 .R66 R96 0
N8 R79 1 R93 99999
@31
R90 3001 @29 39119 R90
@01 3 R91 R56
R79 .03937 R93 9999
N3 R66 .R79
R67 .R79
R73 -R66
R74 -R67
R66 /R61 R72 20 R95 20 R79 0
G X R73 Z R74
@02 9 R72 R20
R72 30
@02 15 R72 R20
N9 R78 0 R94 2
N10 R72 0 R71
R72 -R75
R72 .R68
@03 15 R79 R72
R72 /R94
@03 11 R72 R76
R72 .R68
@02 11 R78 R79
R78 1 R77 0 R72
N11 R75 R77
@01 12 R96 R79
Z R75
@22 X R93 Z R75
@02 99 R90 R89
@22 E R R O R
N99 G1 X R91 Z R92
G91 X -R66 Z -R67
G G90 X R73
@00 13
N12 X R75
@22 X R75 Z R93
@02 99 R90 R89
@22 E R R O R
N99 G1 X R91 Z R92
G91 X -R66 Z -R67
G G90 Z R74
N13 @01-10 R78 R79
@02-10 R95 R20
N15 @02 2 R95 R20
```

```
@01 16 R96 R79
Z R84
@00 17
N16 X R83
N17 G1 X R83 Z R84
R88 2
@02 19 R88 R87
@01 20 R88 R87
R88-1
@00 1
N20 R88 1
N1 R87 R88
R60 0 R81 R62 0 R83
R60 /R61
R62 /R61
R85 -R62
R85 R60
R86 -R84
R86 R82
G9 G R87 X R81 Z R82 I R85 K R86
@00 2
N19 G9 G R87 X R81 Z R82
N2 G X R73 Z R74
M17

L9500 (30.06.88 G4C.01 KONTUR)
G40 G90 R77 0 R64 1 R73 40 R88 1 R78 1 R99 5
@31
R90 3001 @29 39119 R90
@01 1 R91 R77
R78 .03937
N1 R90 2409 @29 37910 R90
R79 R88
N1 R50 1 R51 2
N2 @03 3 R73 R27
R73 0 R27
N3 R60 0 R26 R74 0 R26
N4 @21
@02-4 R64 R88
R65 0 R29 R64 10
N5 R65 -R64
@02-5 R65 R64
R64 2
@02 8 R65 R64
R62-1
@02 9 R21 R83
N6 R52 0 R83 R53 0 R22 R68 0 R21 R69 0 R84
@02 10 R84 R22
N7 R63-1
@00 11
N8 R62 1
@02-6 R21 R83
N9 R52 0 R21 R53 0 R84 R68 0 R83 R69 0 R22
@02-7 R84 R22
N10 R63 1
N11 R58 0 R62
R58 .R24
R58 .R79 R59 0 R63
R59 .R25
R68 -R52
R69 -R53
```

```
R52 -R58 R61 0 R71 0
R53 -R59 R64 9999
R64 .R78
R62 .R79 R76 0 R78
@00 11 R65
N12
N14 R60 .R79 R70 0 R50 R50 0 R51
R74 .R79 R51 0 R70
R76 .R63 R67 0 R62
R67 /R79 R56 0 R53 R57 0 R52
@00 16
N13
N15 R76 .R62 R67 0 R63 R70 0 R68 R68 0 R69 R69 0 R70
R56 0 R52 R57 0 R53
N16 @20 @90 R50
@20 @91 R51
R56 -R76
R60 .R67
R62 .R78
R63 .R78
R52 -R62
R53 -R63 R50 0 R21
R50 -R62 R51 0 R22
R51 -R63
R62 /R79
R77 0 R29 R70 20
R77 -R65
@01 21 R70 R77
G X R52 Z R53
R65 20 R72 2 R77 0
N17 R70 0 R68
R70 -R61
R70 .R67
@03 23 R71 R70
R70 /R72
@02 19 R70 R74
R70 .R67
@02 18 R77 R71
R60 0 R70 R77 1
N18 @02 19 R65 R29
R61 R60
N19 R57 R60
R61 R60
@90 R56 @91 R57
N20 @21
R81 -R58
R82 -R59
R83 -R58
R84 -R59
@22 G1 @90 R64 @91 R57
R83 R58
R84 R59
@01-20 R90 R71
G1 G9 X R91 Z R92
G G91 X -R62 Z -R63
G90 @90 R56
@00-17
N21 @01 22 R71 R24
@00 23
N22 @01 28 R71 R25
N23 @21
```

```
@01-23 R88 R71
R65 20 R70 1
G X R52 Z R53
@02 29 R65 R29
R50 -R58
R51 -R59
G X R50 Z R51
G1 G R73 G91 X R62 Z R63 R88 1
N24 @21
@02 27 R88 R70
R83 -R58
R84 -R59
@02 25 R87 R70
G1 G90 X R83 R58 Z R84 R59
@00 26
N25 G R87 G90 X R83 R58 Z R84 R59 I R85 K R86
N26 @02-24 R70 R88
N27 R65 40 G
@02 29 R65 R29
N28 R70 100 R71 1
R70 .R20
R70 R71
G40 X R52 Z R53
X R50 Z R51
G1 X R21 Z R22 G R73 L R70
N29 G G40 G90 X R52 Z R53
M17

L9600
R81 0 R10 R82 0 R11 R83 0 R12
R84 0 R13 R85 0 R14 R86 0 R15
R87 0 R16 R88 0 R17 R89 0 R18
R77 0 R65 1 R80 0 R19
@31
R90 3001 @29 39119 R90
@01 1 R91 R77
R65 .03937
N1 G90 R68 0 R69-1 R75-1 R66 0 R86
@03 1 R66 R68
R69 1
N1 R65 .R69
R71 0 R83
R71 -R81
R72 0 R84
R72 -R82
R71 /R80
R65 .R80
R81 R65
R65 /R80
R87 .R69
R66 R87
R88 -R75
R66 R66 /R88
R88 R75
R93 0 R66
R93 /R88
R79 0 R66 -R65
R79 -R93
G0 X R81 Z R82
N6 G91 X R66
G33 X R71 Z R72 K R85
```

```
G0 X -R66
G90 X R81 Z R82
R88 R75
@02 9 R88 R68
R66 0 R86
R66 -R65
@01 10 R87 R68
R87 0
@00-6
N9 R66 R79 -R93
@00-6
N10 @01 11 R89 R68
R89 R75
@00-6
N11 M17

L9700 R77 0 R75 1 R78 1
@31
R90 3001 @29 39119 R90
@01 1 R91 R77
R78 .03937
N1 R90 2409 @29 37910 R90
R79 R75
N1 R60 1 R61 2
N2 R72 0 R25 R56 0 R24 R62-1 R64 0 R65-1
@03 3 R24 R64
R62 1
N3 R50 1 R51 1 R52-1 R53-1
R78 .R62 R58 0 R31
R58 -R21
R58 /R79 R59 0 R32
R59 -R22
@03 4 R58 R64
R58 .R65 R50-1 R52 1
N4 @03 5 R59 R64
R59 .R65 R51-1 R53 1
N5 @03 6 R59 R58
R76 0 R60 R60 0 R61 R61 0 R76
R76 0 R50 R50 0 R51 R51 0 R76 R76 0 R58 R58 0 R59 R59 0 R76 R65 1
N6 @20 @90 R60
@20 @91 R61
R54 0 R58 R76 0 -R51
R54 /R59 R60 0 R26
R60 .R54 R61 0 R27
R61 .R54
R58 R60
R58 R61
R59 R26
R59 R27
R50 .R58
R51 .R59
@03 7 R65 R64
R60 .R79
R52 .R60
R78 .R79
R52 R78
R78 /R79
R53 .R26
@00 8
N7 R52 .R26
R52 .R79
```

```
R53 .R60
R53 R78
N8 R52 R21 R70 0 R71 0
R53 R22 R65 90 R29 R66 0 R29
@15 R66
@15 R65
R66 /R65 R68 0 R25
R68 .R62 R65 1 R61-1
R56 R68 R57 0 R28 R68 0 R28
@10 R57
R56 /R57
G G90 X R52 Z R53
@02 9 R64 R24
R75-1
N9 @03 10 R28 R64
R68 .R61 R57 1 R79 2
R57 /R79
@00 13
N10 R64 R65
@02 12 R68 R64
@02 11 R72 R71
R70 R65 R57 0 R24
@00 14
N11 R72 0
N12 R57 0 R64
@10 R57
N13 R57 .R56
N14 R67 0 R24
R67 -R57
R67 .R76
R67 .R66
R67 .R75 R54 0 R51
R54 R67
R57 -R78
G91 @91 -R67
@90 R57
G33 @90 R50 @91 R54 I R20 K R20
G @90 -R57
G90 X R52 Z R53
@03-10 R23 R70
M17

L9800 (30.06.88 G4C.01 TIEFBOHREN)
@31
R77 0 R78 1 R90 3001 @29 39119 R90
@01 1 R91 R77
R78 .03937
N1 R63 0 R22 R64 0 R25 R65 0 R67 2
R63 - R26
R67 . R24
G G64 G90 ZR22
N3 R63 - R64 R62 0 R26
@03 4 R65 R63
R62 R63
G1 ZR62
G4 XR28
G ZR22
G4 XR27
R62 R78
ZR62 - R78
@03 4 R24 R63
```

```
R64 - R24
@02 -3 R64 R24
R64 0 R24
@03 -3 R63 R67
R64 0 R63 R62 2
R64 / R62
@00 -3
N4 G1 ZR26
G4 XR28
G ZR22 M17

L9900
@31 M17
M02
```

7.2 Clearing the buffer, L99 (see Section 5.6)

A series of control signals from the interface controller (parallel interface or PLC) is registered in the main memory of the NC indirectly via buffers. Associated with these control signals are:

- external additive ZO
- mirror image
- external tool offset

These may be activated by using M functions, for example.

If these signals, which are actuated in the active program, are to be effective in the block following their selection, the block buffer must be cleared. Alternatively the selected control signal only becomes active a few blocks later.

In each program the buffer can be cleared by calling subroutine L99.

The control registers the status "Buffer empty" in the interface control and the selected control signal or the required external data input can be enabled.

The program L99 has to be defined as follows:

```
L9900
@ 31 M17.
```

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

```
.
.

N15 M..        - PLC removes the read-in enable with M function and
                 afterwards transfers the external tool offset
               - After acknowledgement of data transfer.
N20 G04 X..      The PLC enables reading in again.
N25 L99
.
.
.
```

8. Appendix

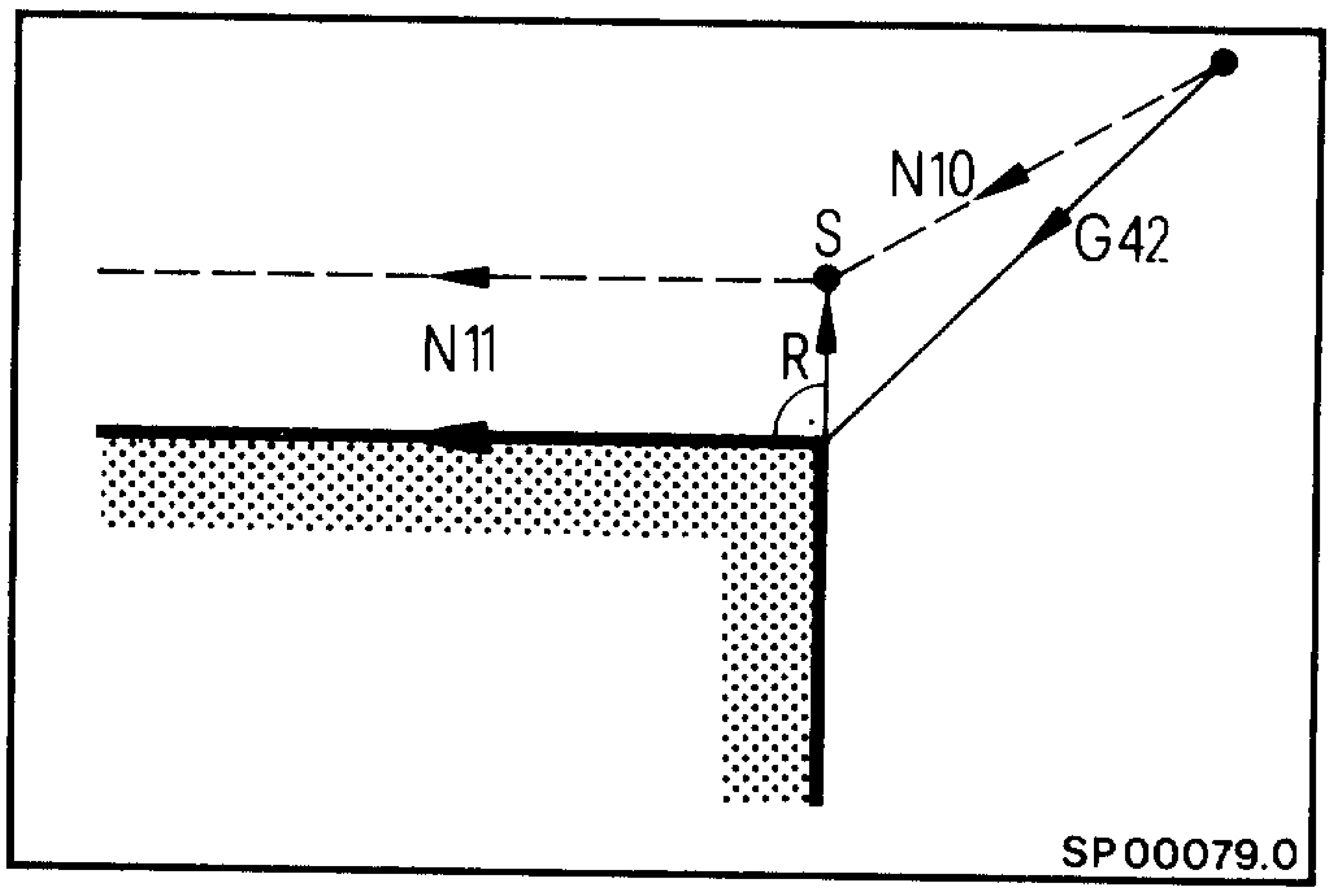
8.1	Tool nose radius compensation (CRC)
8.1.1	Activation of CRC
8.1.2	CRC in the program
8.1.3	Cancellation of CRC
8.1.4	M00, M02, M30 with CRC selected
8.1.5	Special cases with CRC
8.1.6	CRC in combination with various types of block
8.1.7	Repetition of previously activated G functions (G41, G42) with the same offset number
8.2	Input systems, diagrams and tables
8.2.1	Inexact input of interpolation parameters I and K
8.2.2	Reference points
8.2.3	Calculation of block increment
8.2.4	Limit data for rotational feed
8.2.5	Spindle speed as a function of turning radius with v = constant
8.2.6	Input formats
8.2.7	Code table
8.3	Program key
8.4	Special cases
8.4.1	Special case "Delete distance to go".

8.1 Tool nose radius compensation (CRC)

In the diagram below all stationary points for single block are designated S.

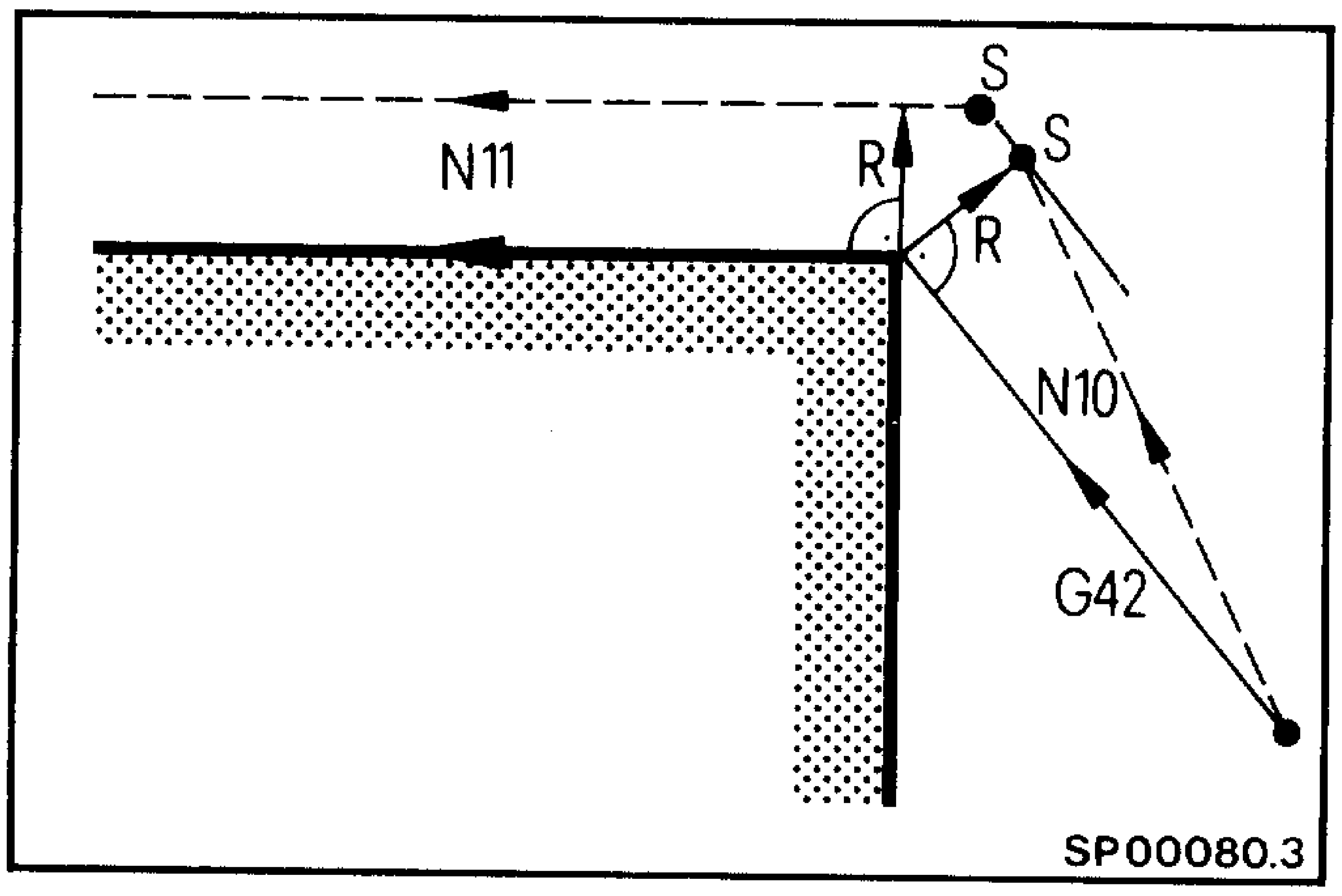
8.1.1 Activation of CRC

- for inside contours (angle between block N10 and N11 < 180°)



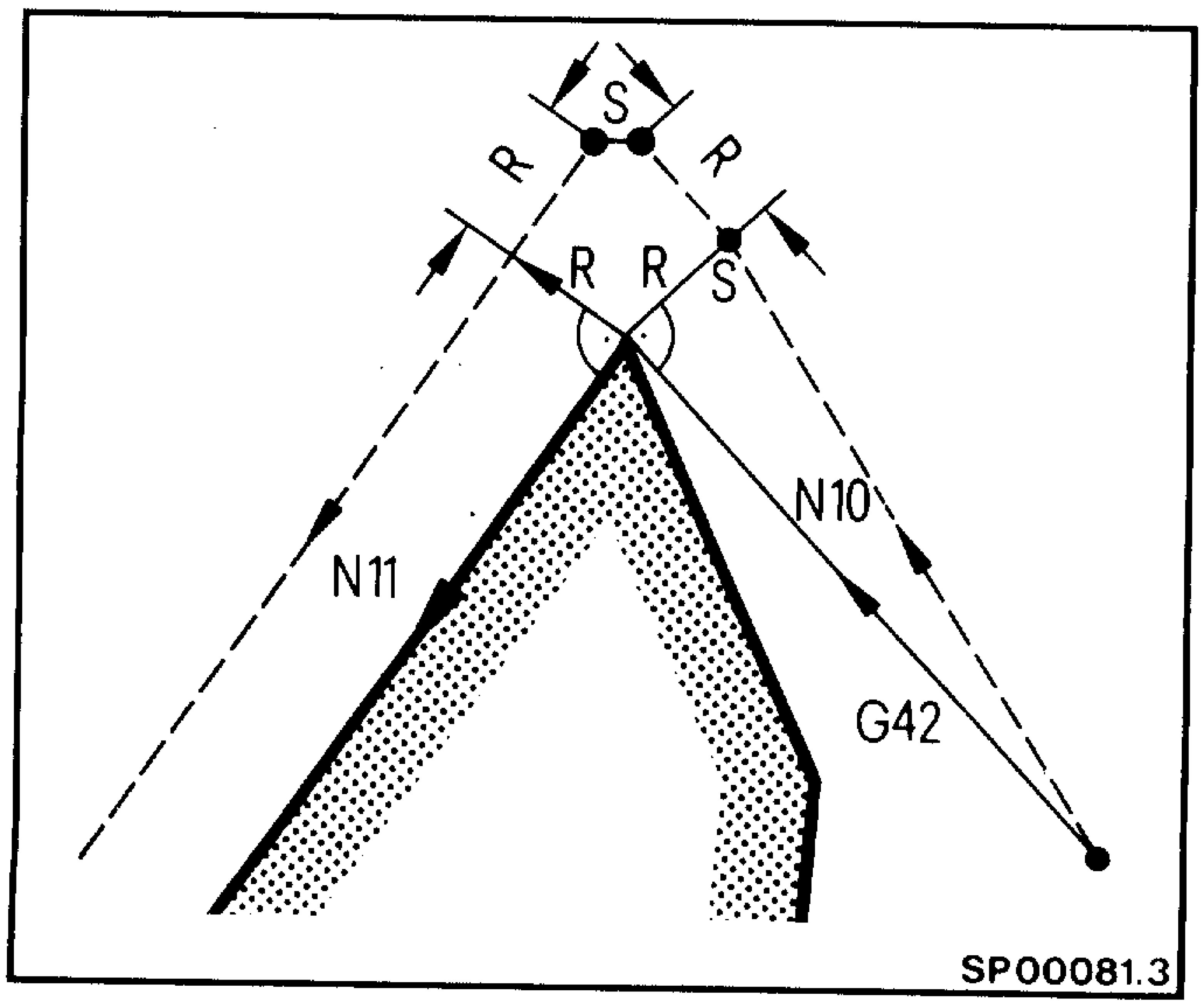
Start point.
In the block following activation a block start vector (length R) is set up perpendicular to the programmed path.
Block start vector.

- for outside contours (angle between block N10 and N11, 180° to 270°)



Start point

(Angle between block N10 and N11 < 270°)

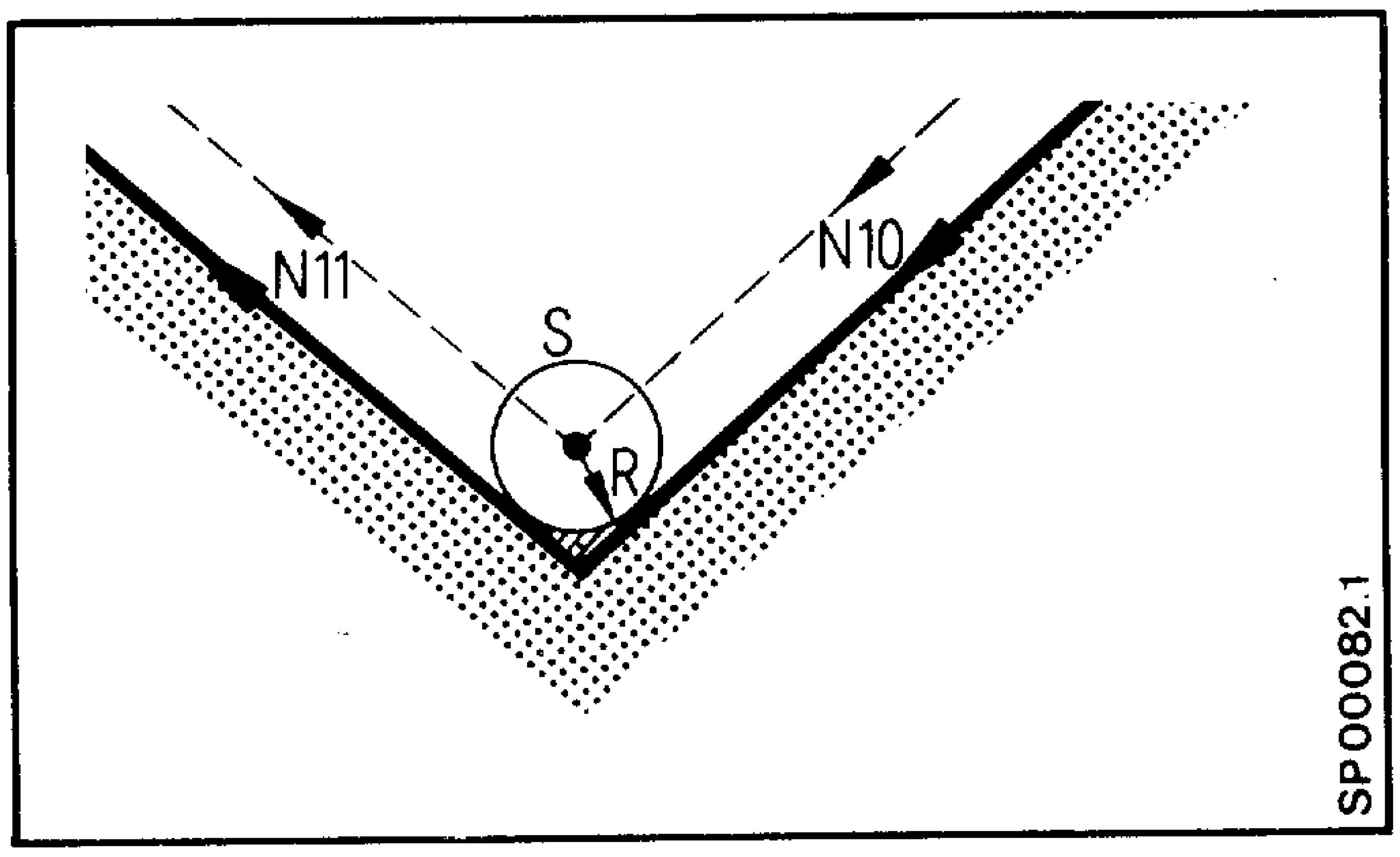


Start point

8.1.2 CRC in the program

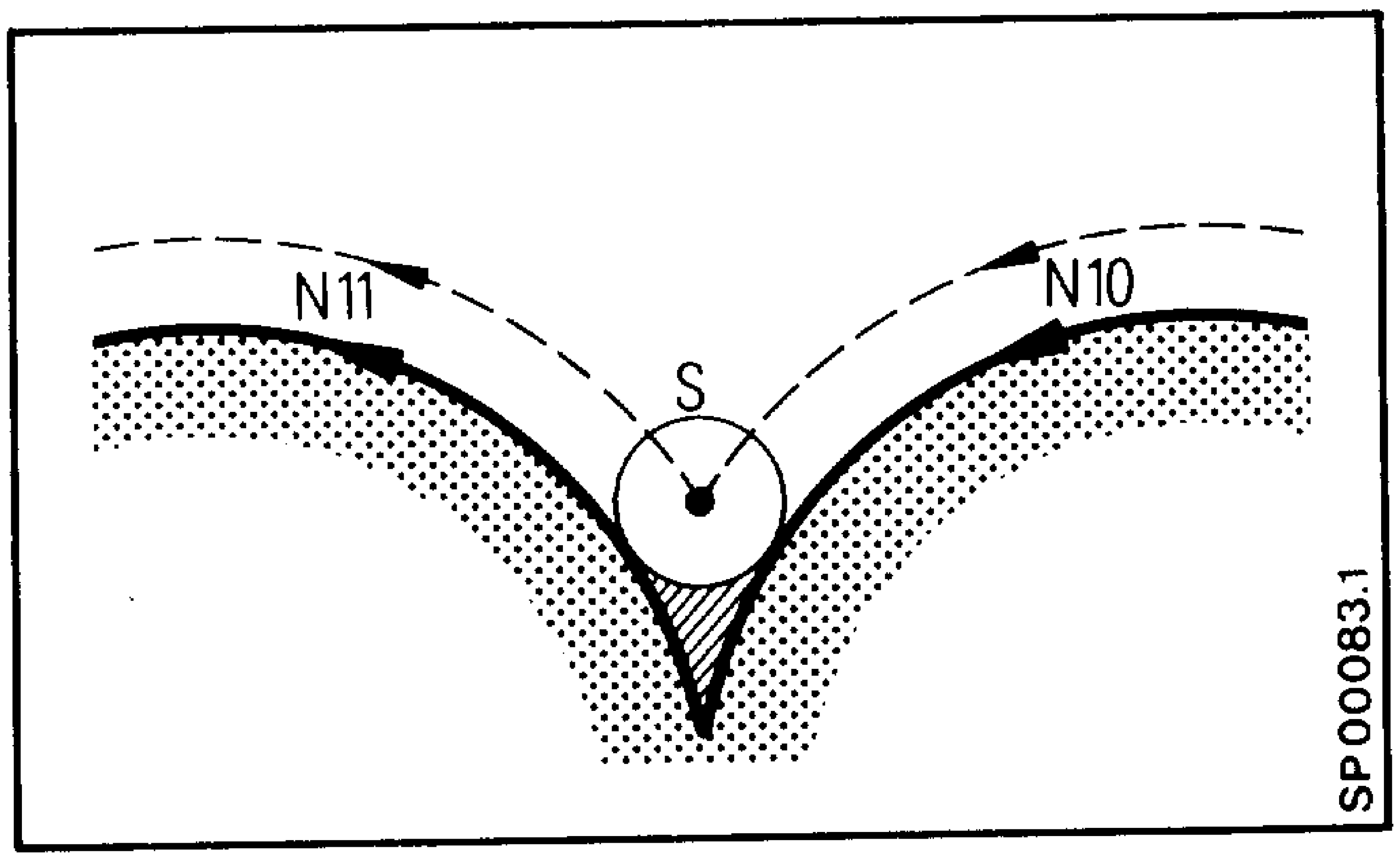
- inside contour (angle between 2 blocks < 180°)

Straight line - straight line



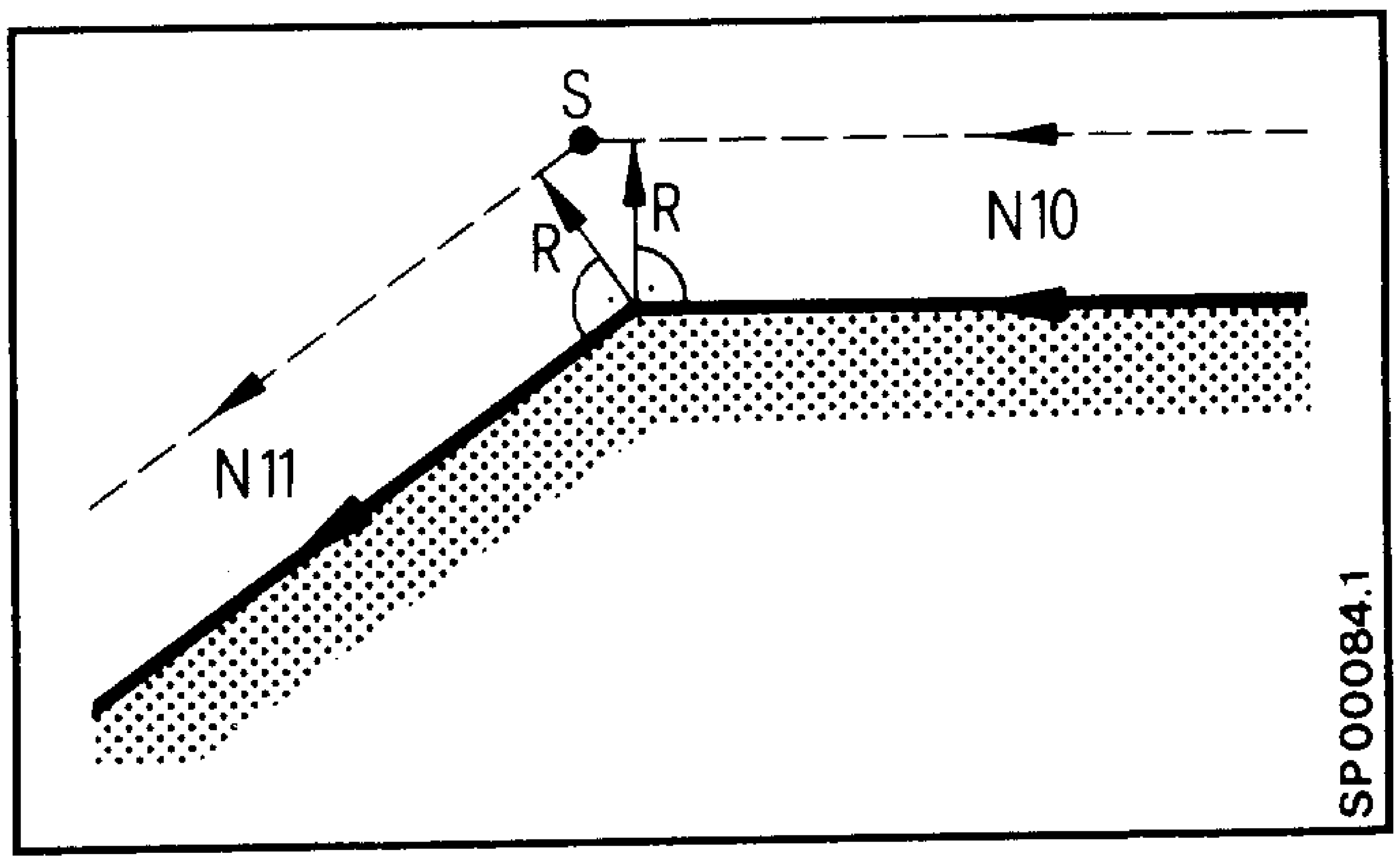
In each case the intersection of the corrected path is calculated

Circle - circle



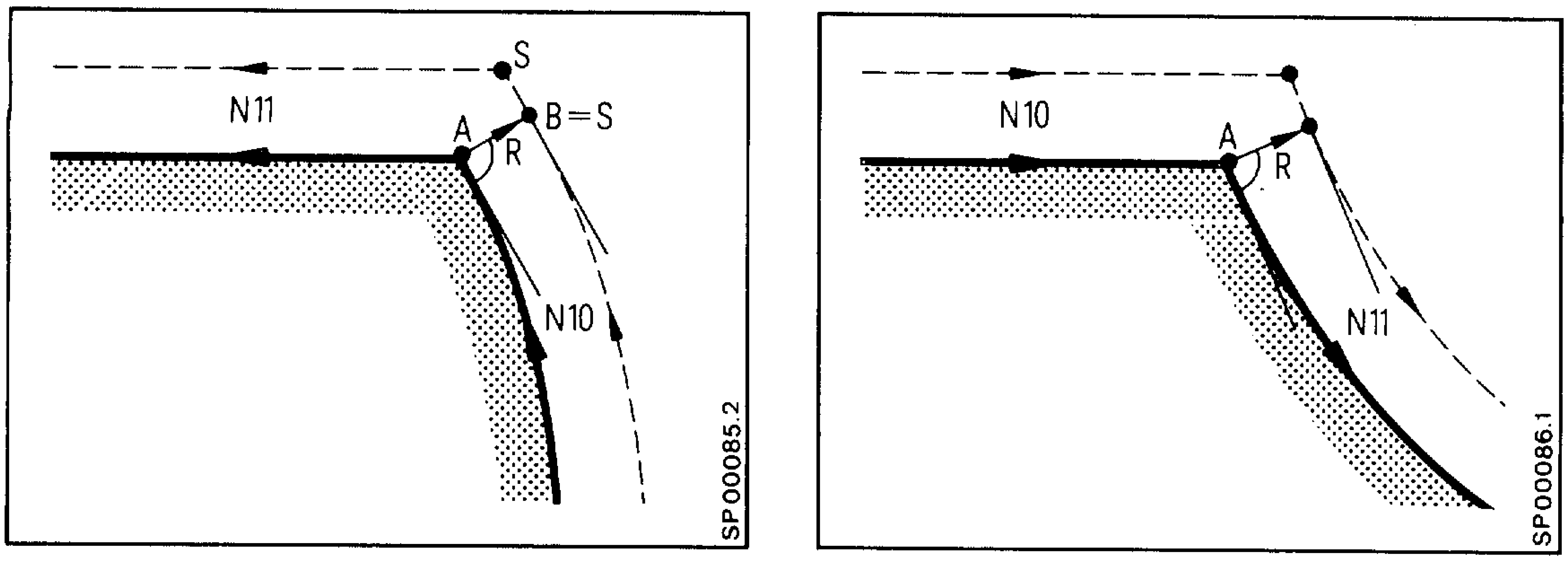
- outside contour (angle between 2 blocks, 180° to 270°)

Straight line - straight line



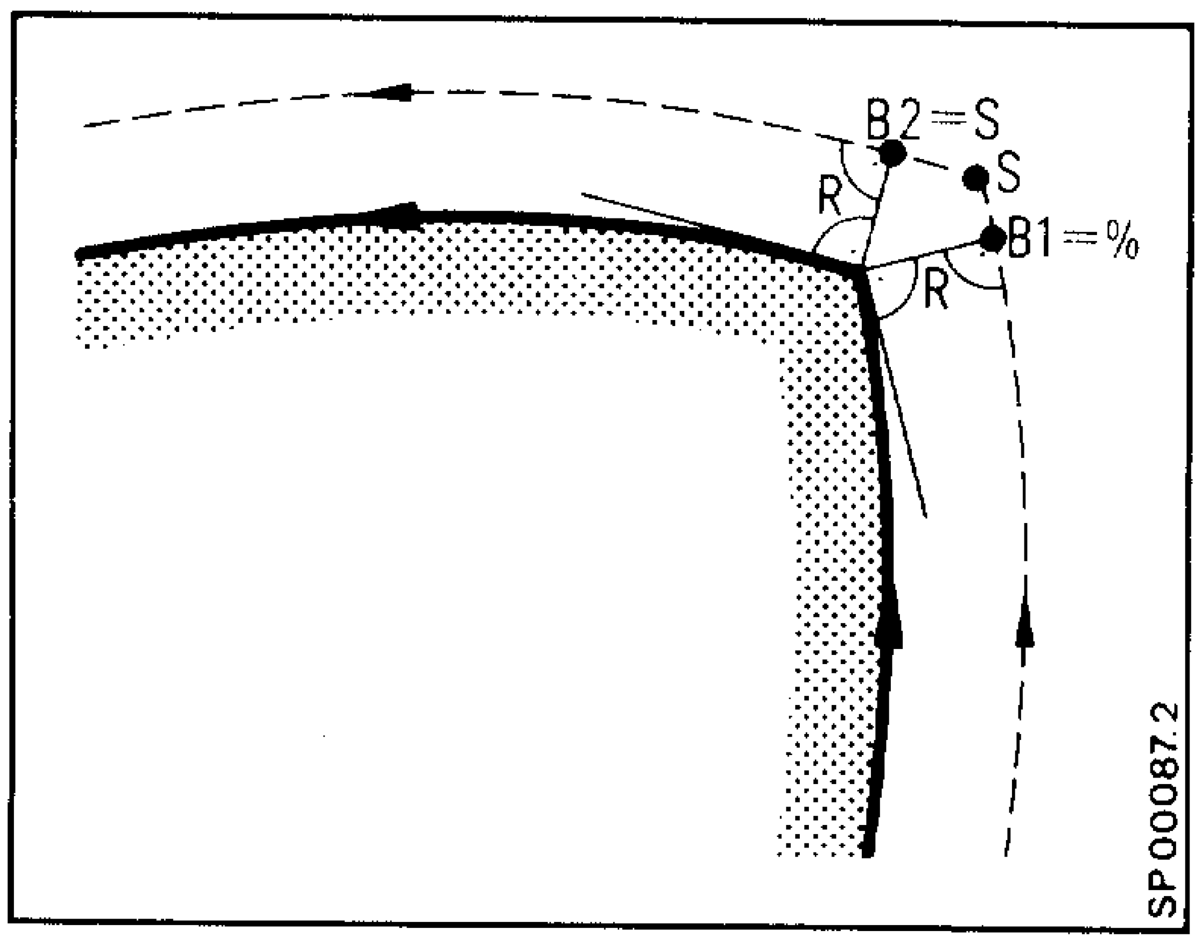
The intersection of the corrected path is calculated.

Straight line - circle



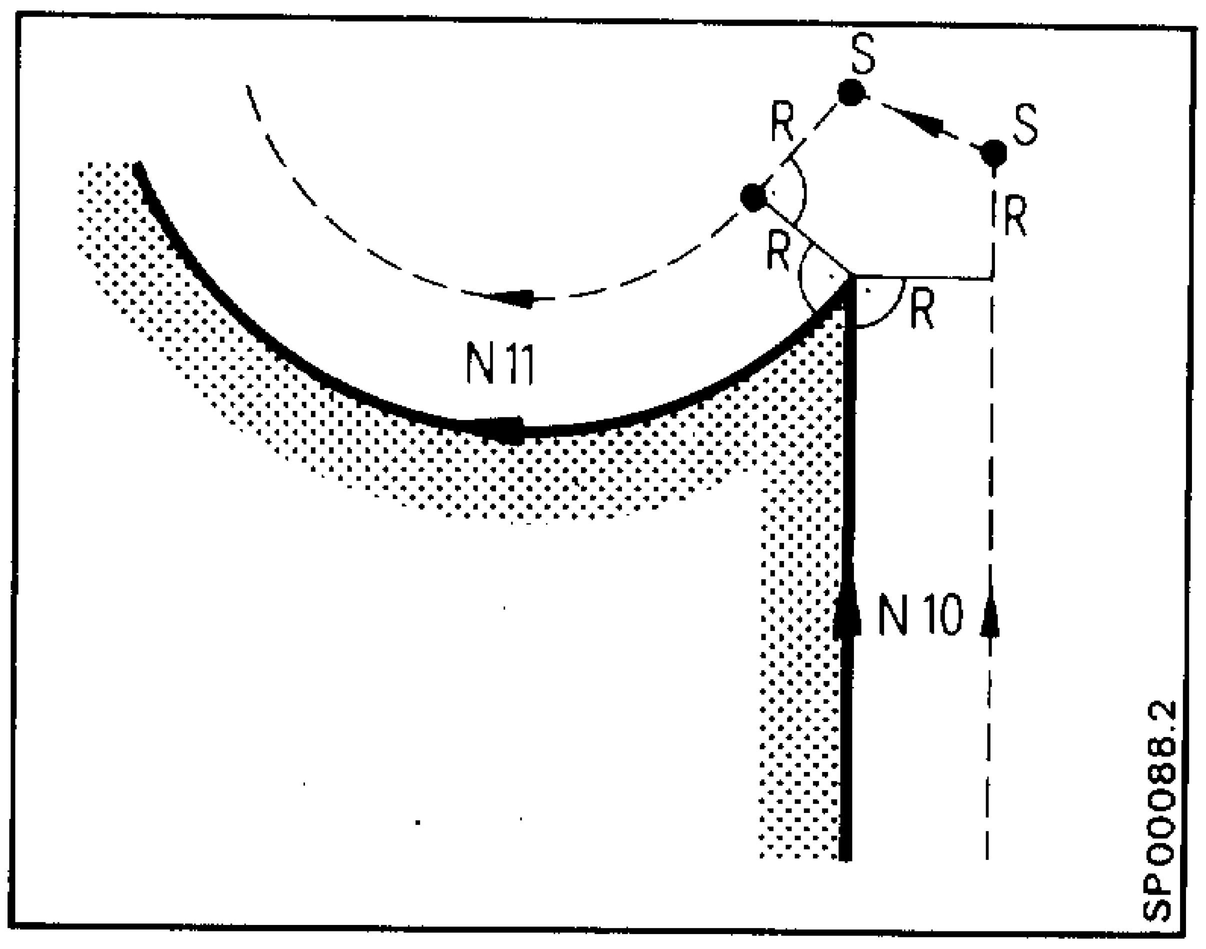
A vector (length R) is set up perpendicular to the end point of the circle (or start point) A. The intersection is calculated between the tangent at point B and the corrected path in block N11 (or N10).

Circle - circle



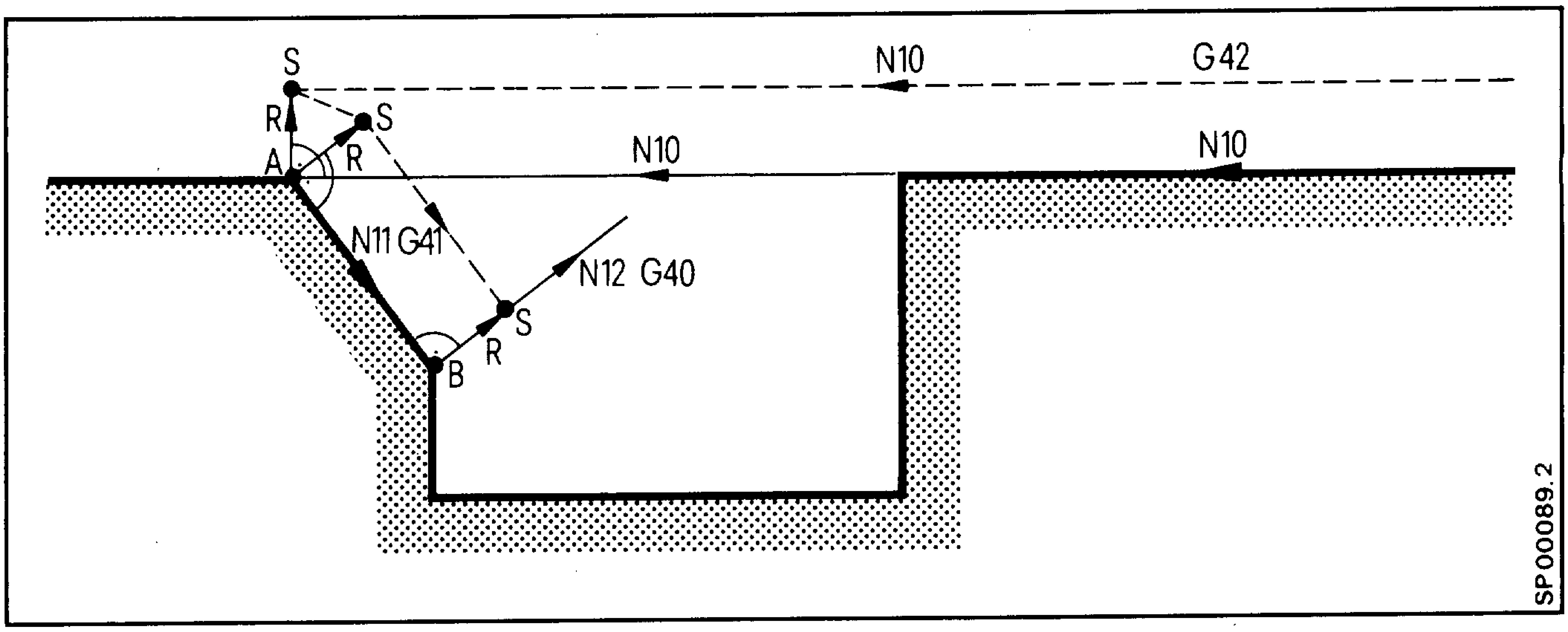
A vector of length R is set up at right angles to the block end point or start point
The cutting point is calculated between the tangents at point B1 and B2.

Straight line - circlular arc > 270°



A vector of length R is set up at right angles to the end point or start point of block N10 or N11.
A compensating movement is made beyond the programmed contour by the tool nose radius R to prevent damage to the workpiece.

Change of compensation direction



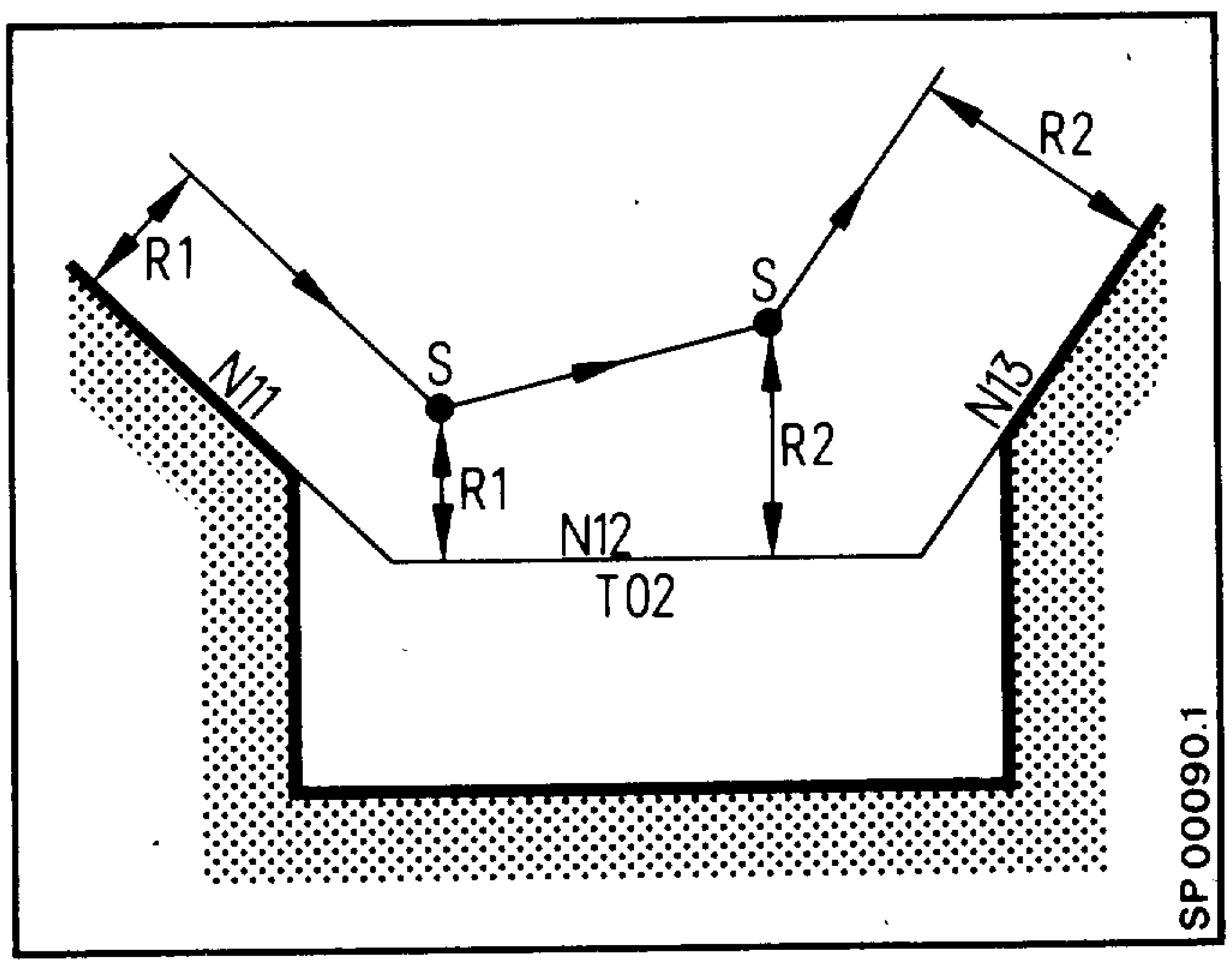
A perpendicular vector of length R is set up at the end point of the block with the old compensation direction (G42 here) and at the start point of the block with the new compensation (G41 here).

A small chamfer is made at point A because at this point no cutter point calculation is made.

In order to obtain exact machining at point B in this example, it would be necessary to deactivate the CRC in block N12 using G40.

Change of tool offset number (T.... T...)

The following applies when the tool offset is changed:

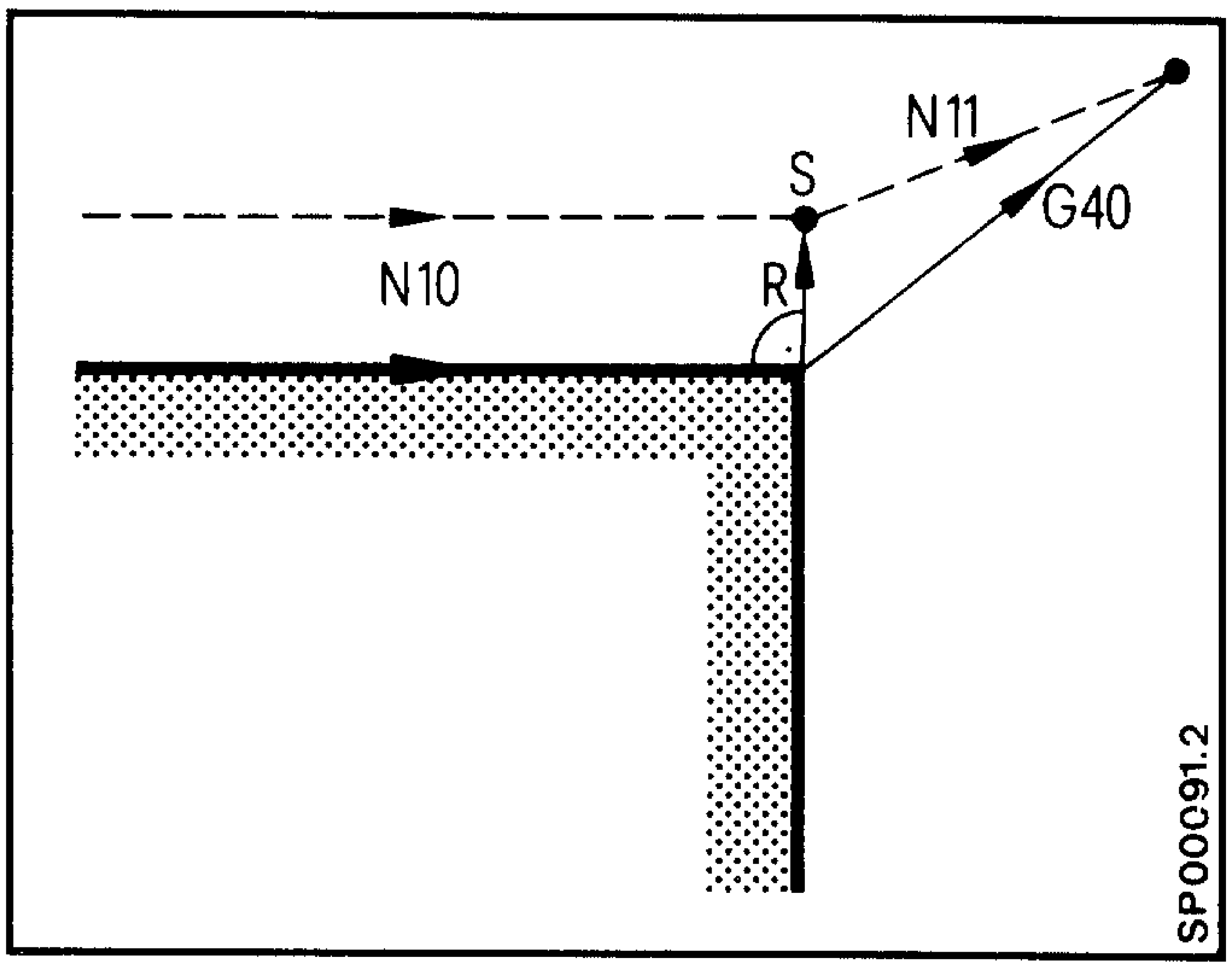


1. The cutting point at the start of the block is calculated using the old tool offset.
2. The cutting point at the end of the block is calculated using the new tool offset.

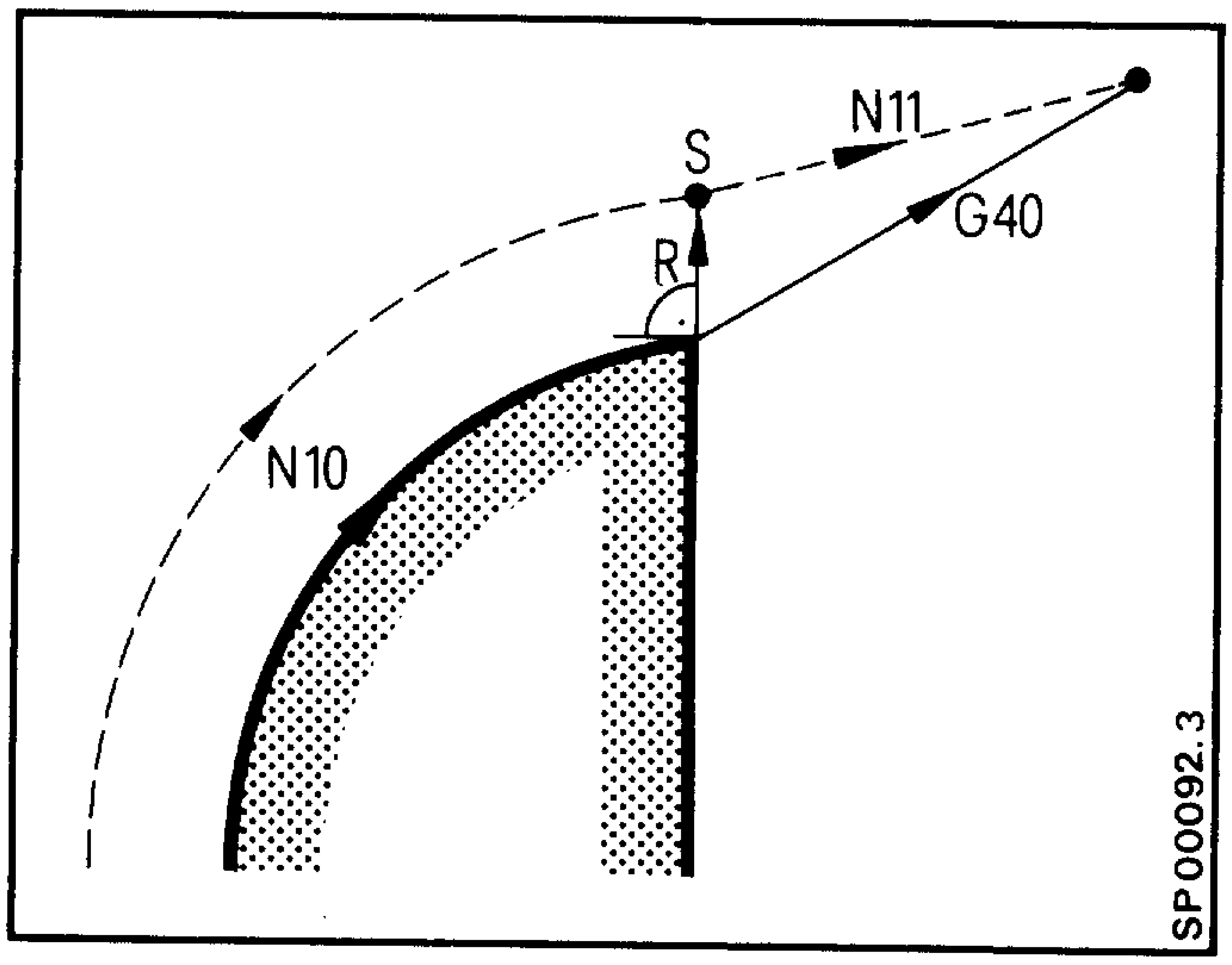
8.1.3 Cancellation of CRC

- for inside contours (angle between block N10 and N11 $< 180^{\circ}$)

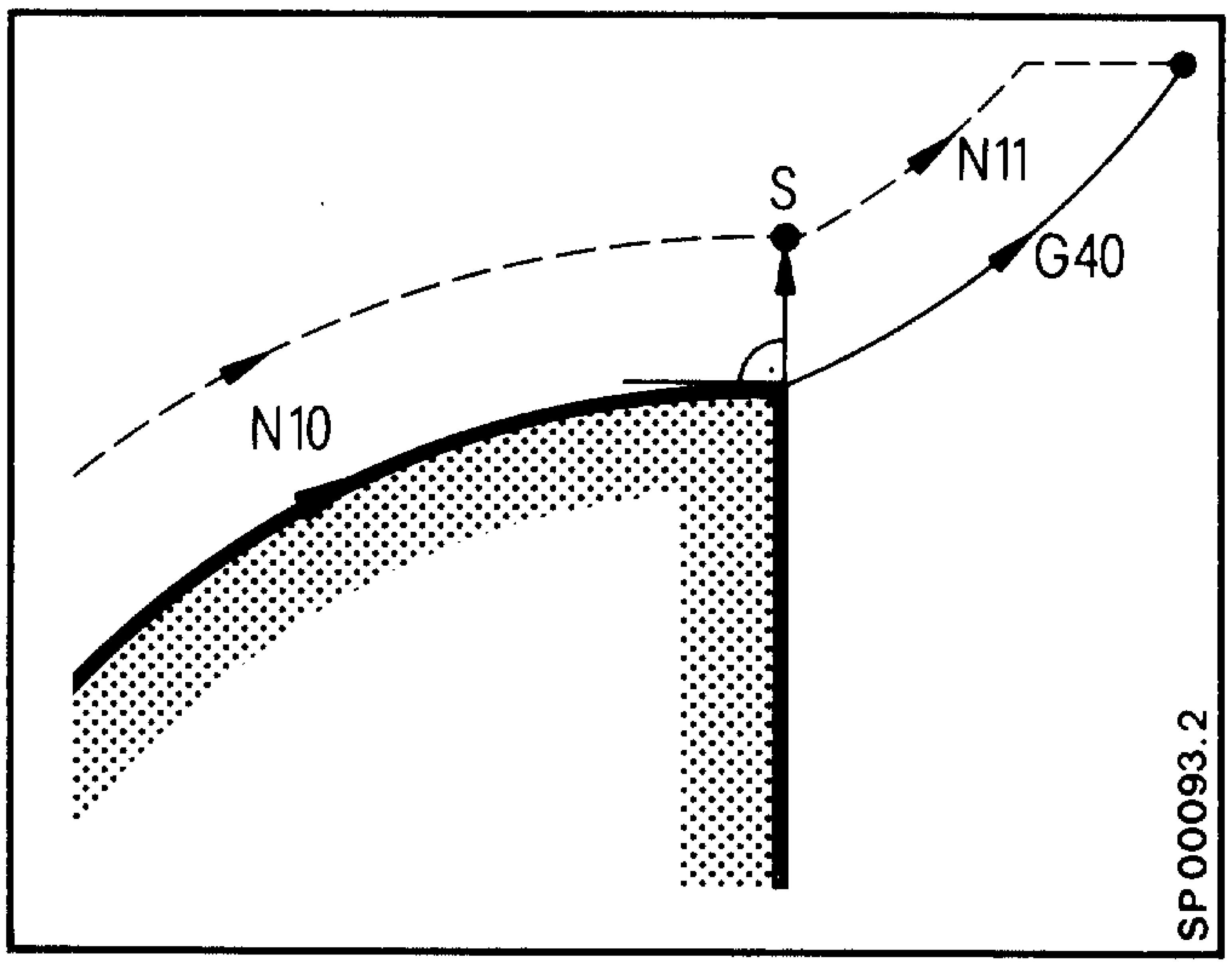
Straight line - straight line



Circle - straight line



Circle - circle

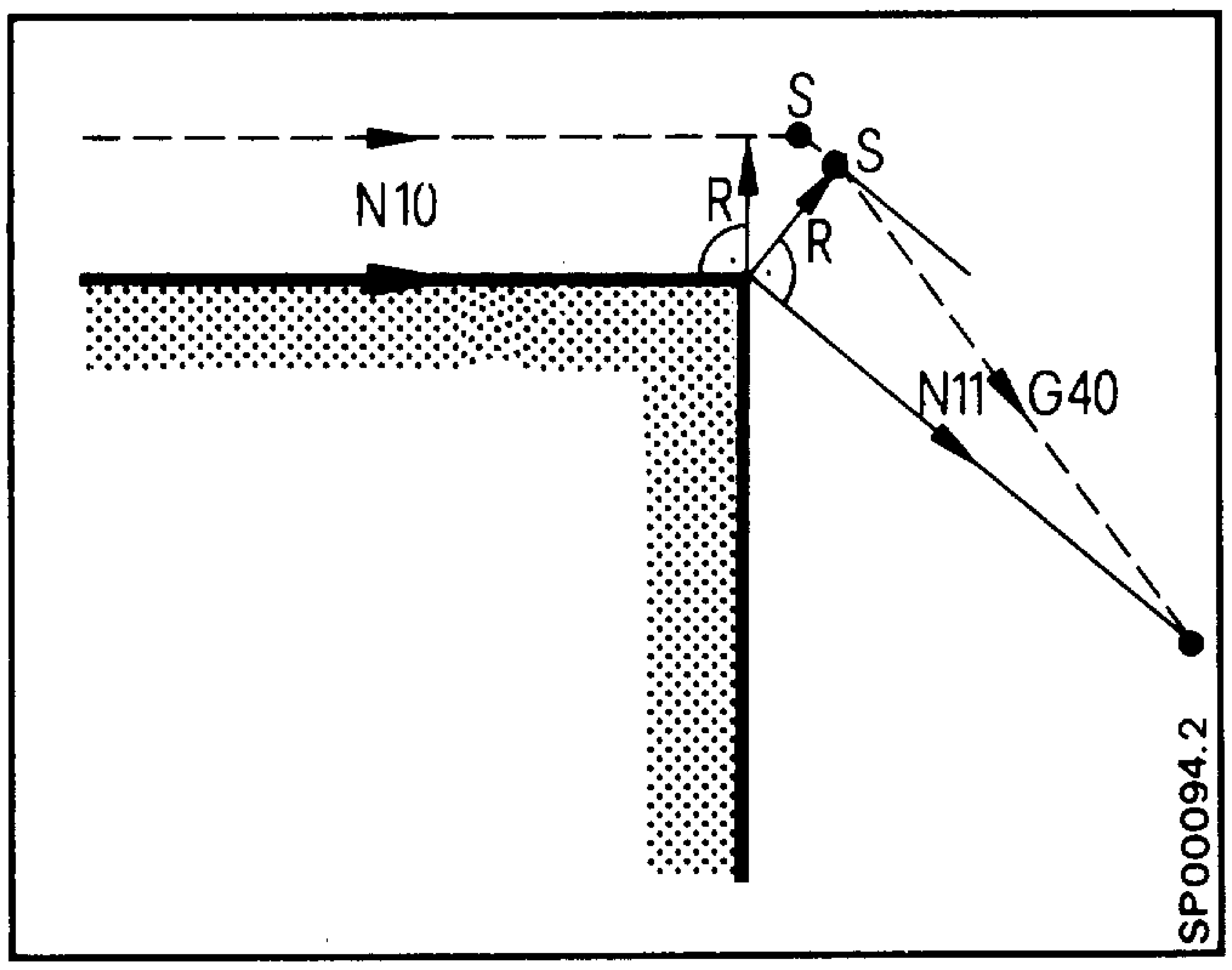


In the last block with CRC a vector of length R is set up at right angles to the programmed path.

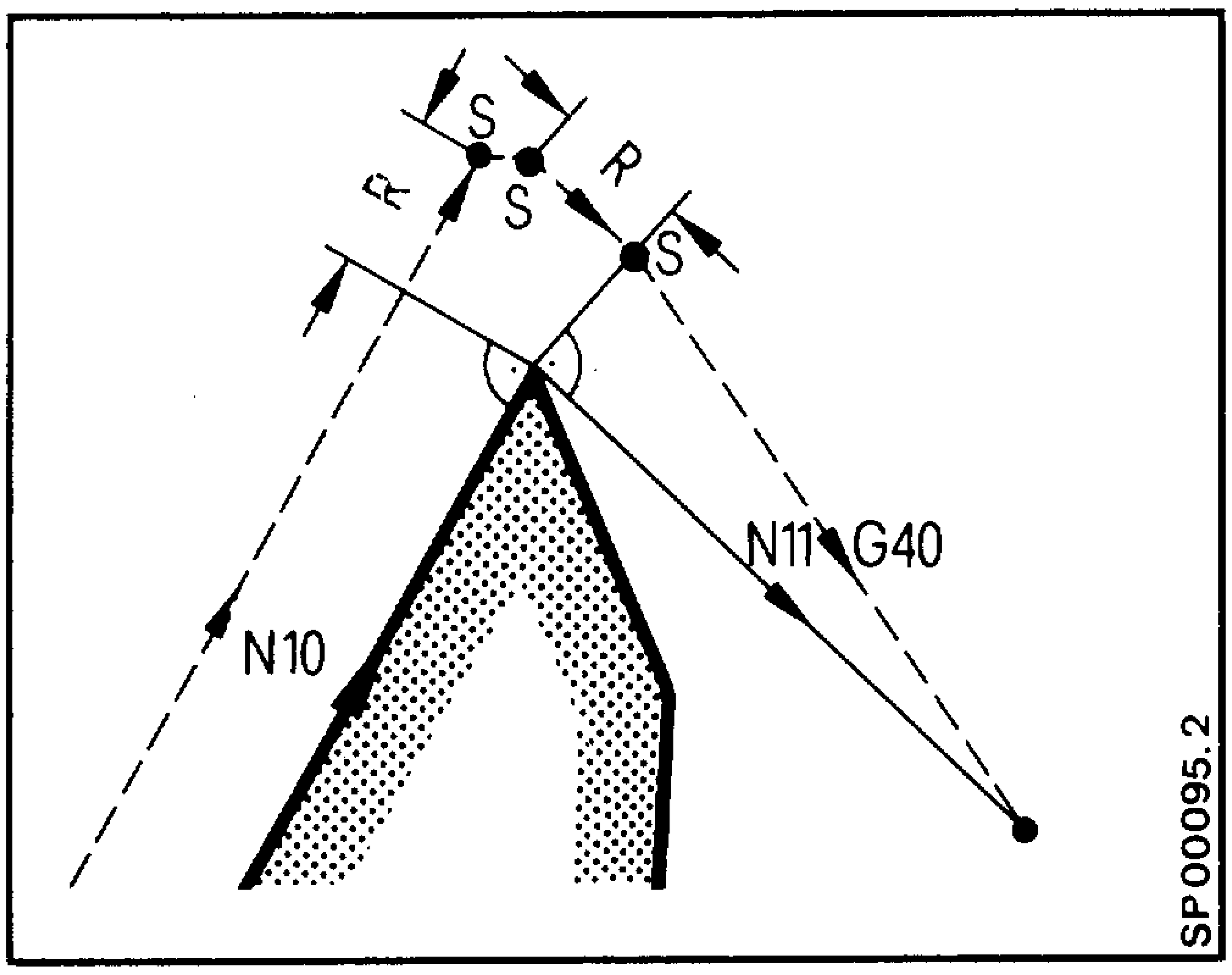
During transition to a straight line the programmed end point is approached directly. During the transition to a circle the programmed circle is approached with offsets until the perpendicular to the circle end point is reached. The approach is then made along the perpendicular to the end point.

If an end point coordinate (X or Z) is reached before the perpendiculars, the other coordinate is approached directly (see diagram above).

- for outside contours (angle between N10 and N11, 180^{o} to 270^{o})



(angle between N10 and N11 > 270^{o})



The corrected path is calculated and traversed up to the cutting point at the start of the block in which the CRC is deactivated.

G40: Cancellation of tool nose radius compensation (CRC)

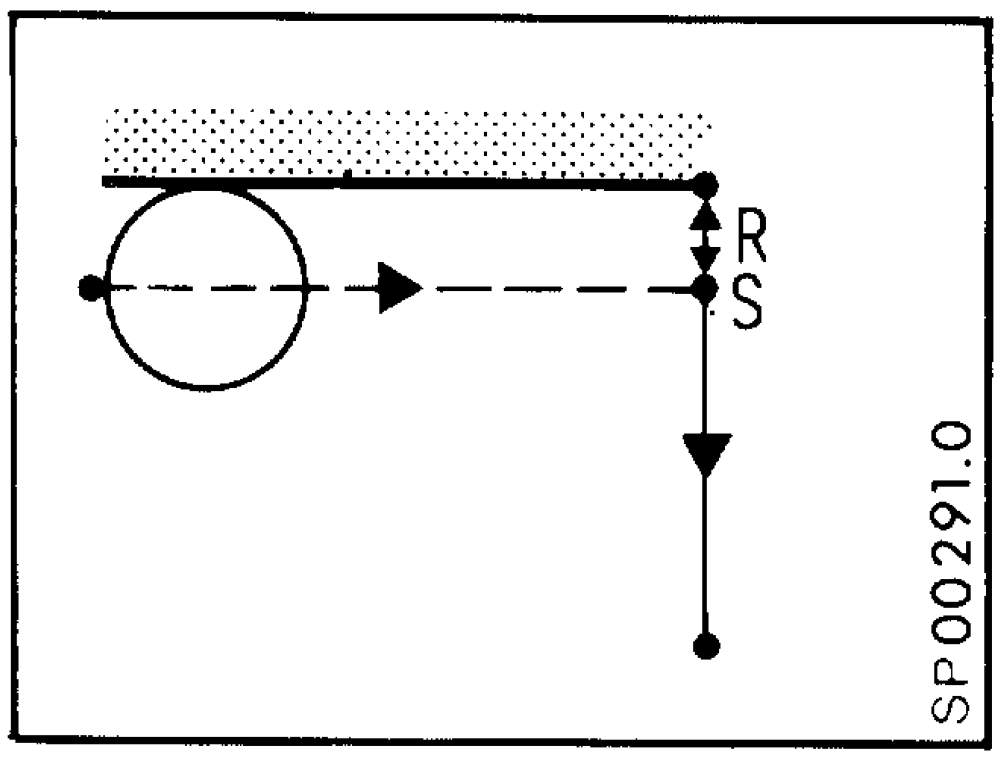
TOO Cancellation of tool nose radius compensation and tool length compensation.

The CRC should be cancelled before or in the last block.

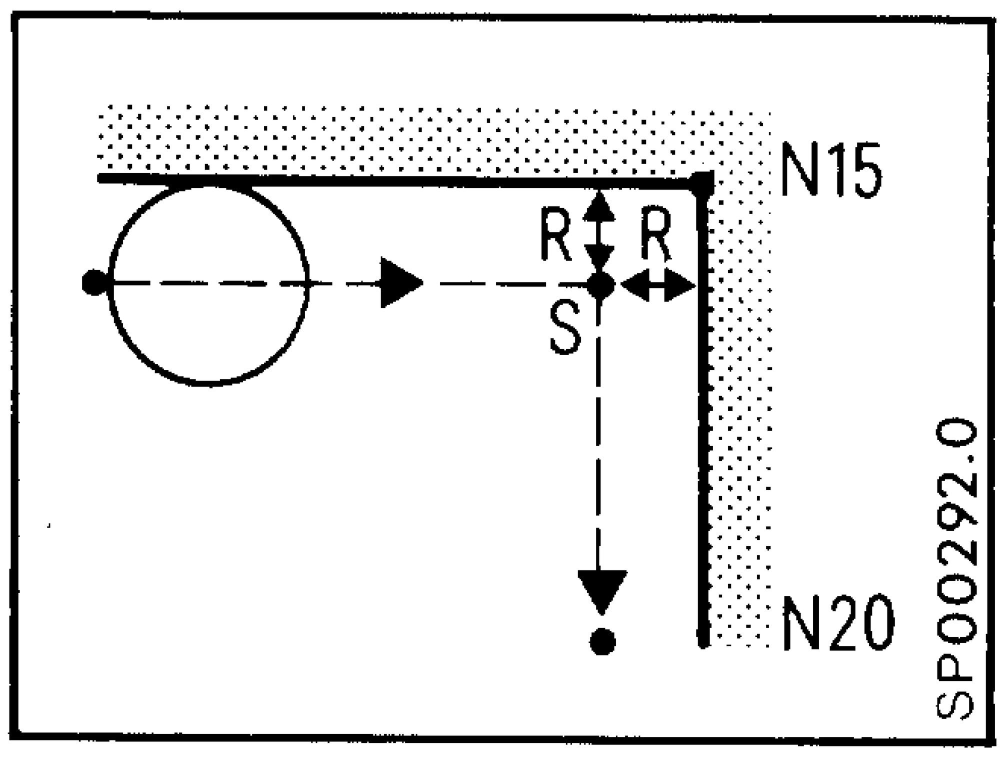
8.1.4 M00, M02, and M30 with CRC selected

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

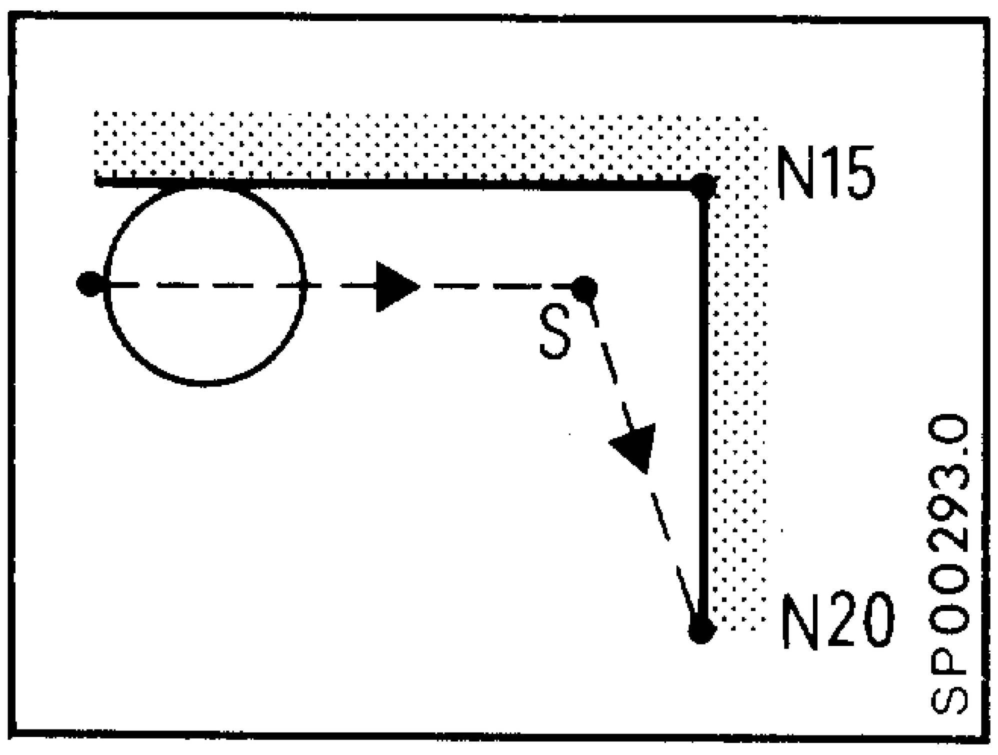
M02, M30:



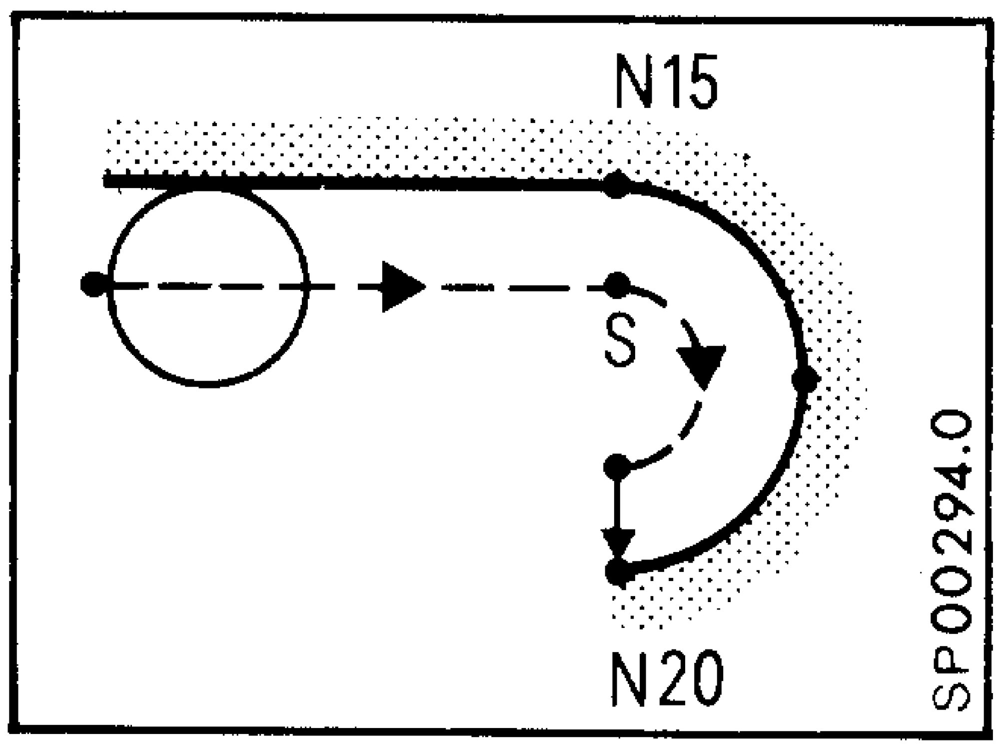
- Correct cancellation by G40 or G41/42 T00
  N10 X20 Z-30
  N15 Z0
  N20 X0 G40 M30



- Reaction when G40 or G41/42 are missing or when an axis is programmed in the last block
  N10 X20 Z-30
  N15 Z0
  N20 X0 M30
  The control calculates the last intersection S. The CRC is not executed.



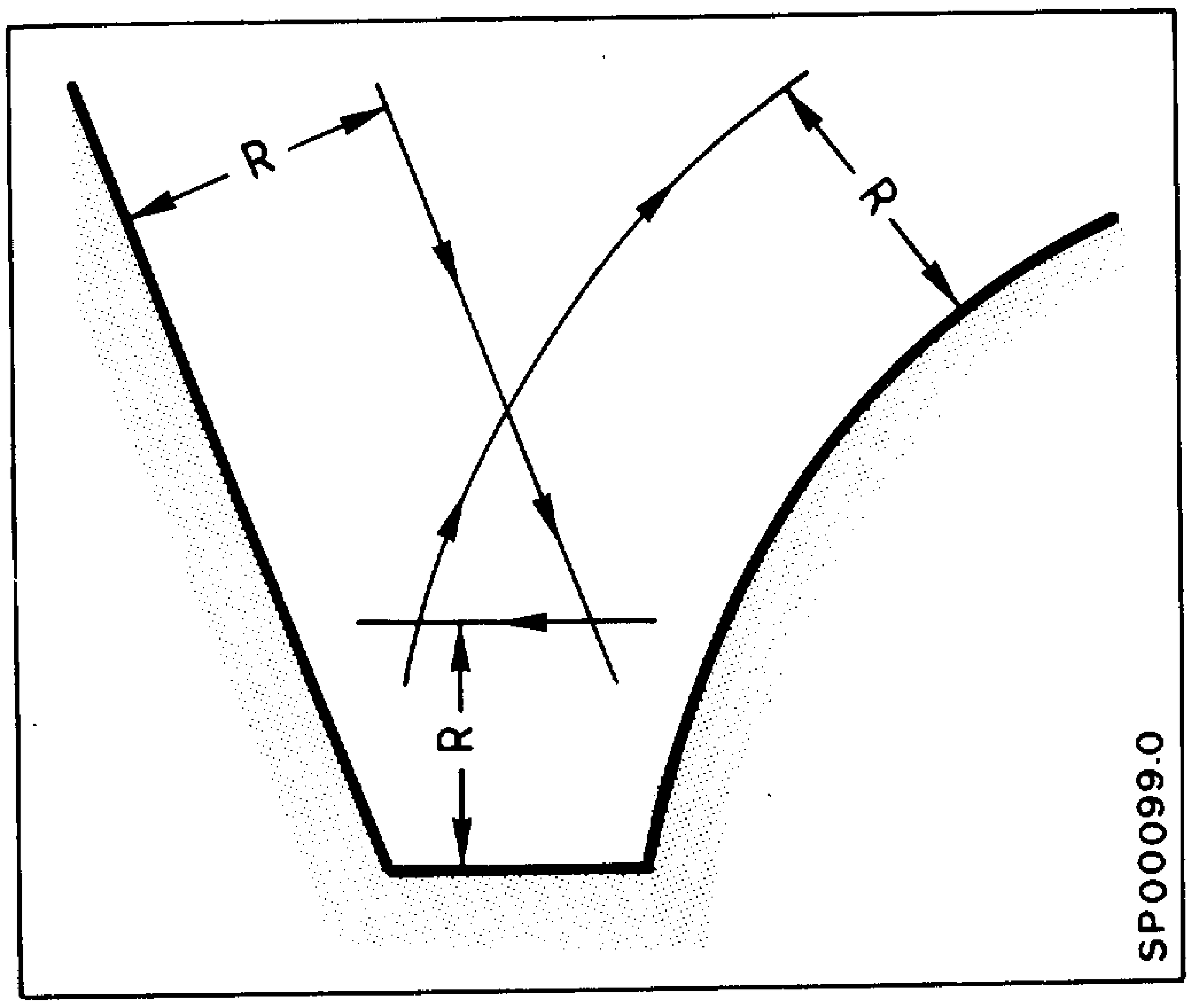
- Programming of both axes in the last block
  N10 X20 Z-30
  N15 Z0
  N20 X0 Z0 M30
  The CRC is executed in N20.



- Circle block at last block
  N10 X20 Z-30
  N15 Z0
  N20 G02 X0 I-10 M30
  The controller detects a circle end point error and stops at S with alarm 308. If the threshold for circle end point errors is great, the controller ends the program at N20, as it does for cancellation in the circle block.

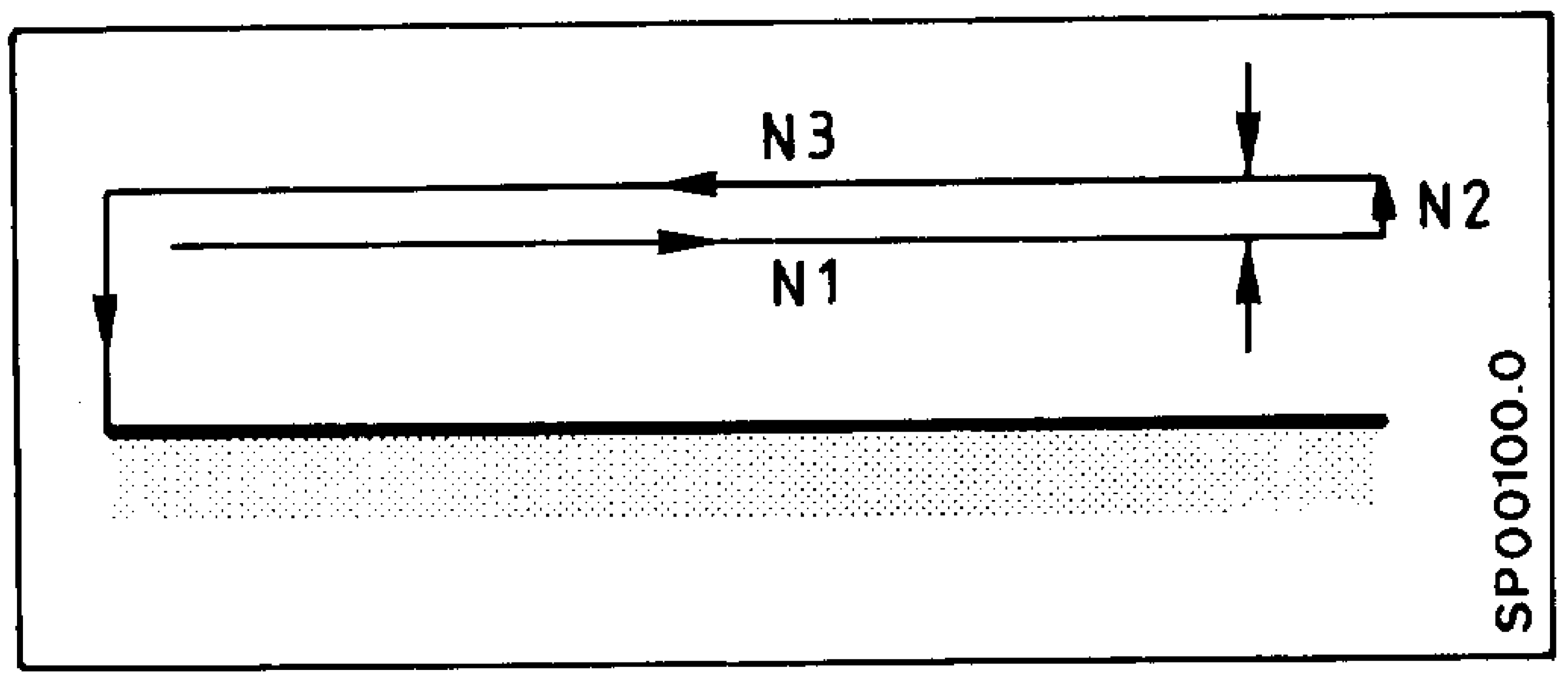
8.1.5 Special cases with CRC

Since the NC only uses the next block for calculation of the cutting point, the following contours errors can result with inside contours:



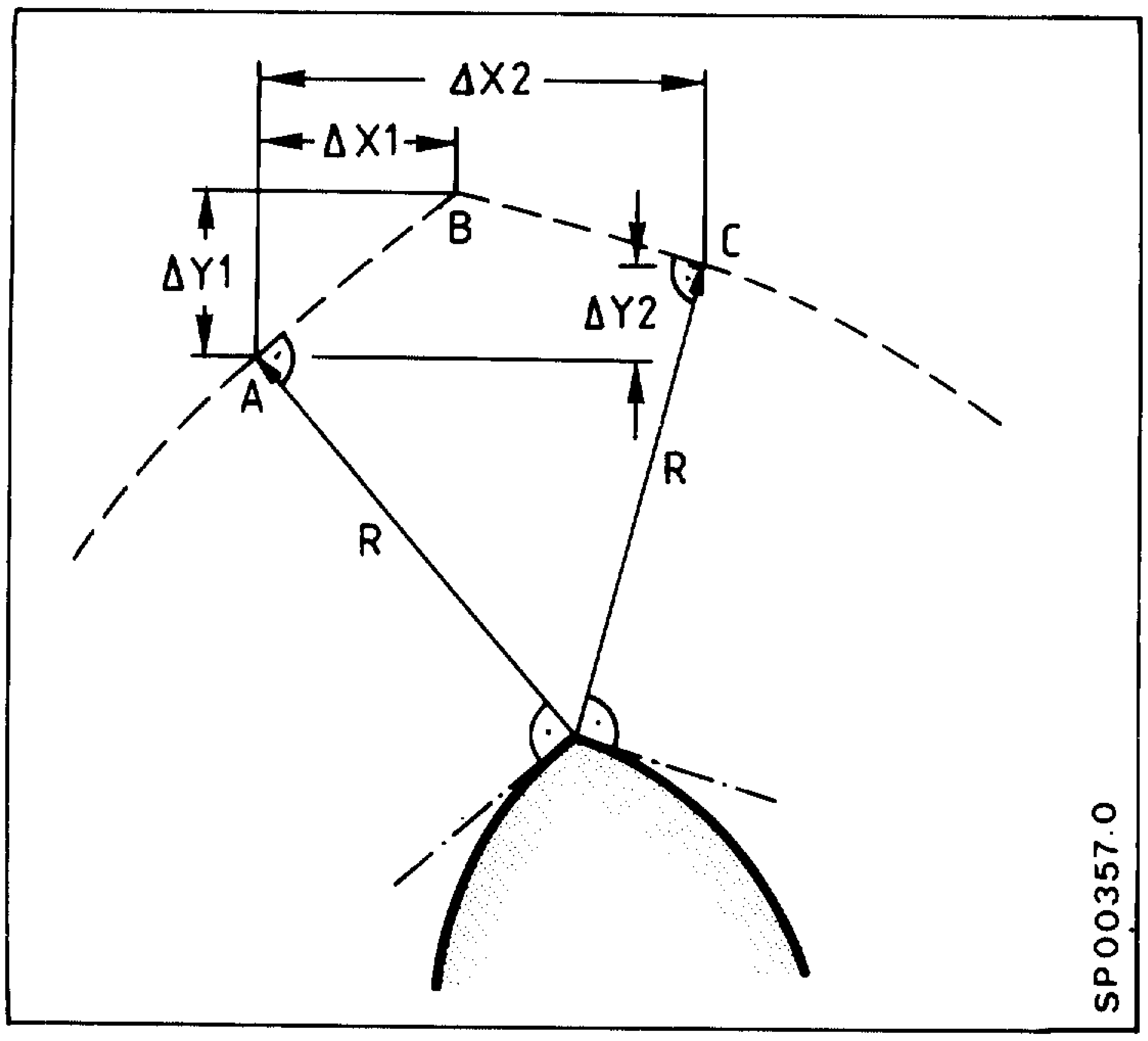
Intermediate block for the activated compensation is too small. Machining is not interrupted. An alarm (321) is displayed and cancelled at the end of the program.

The compensation direction for the CRC is maintained and the traversing direction reversed.



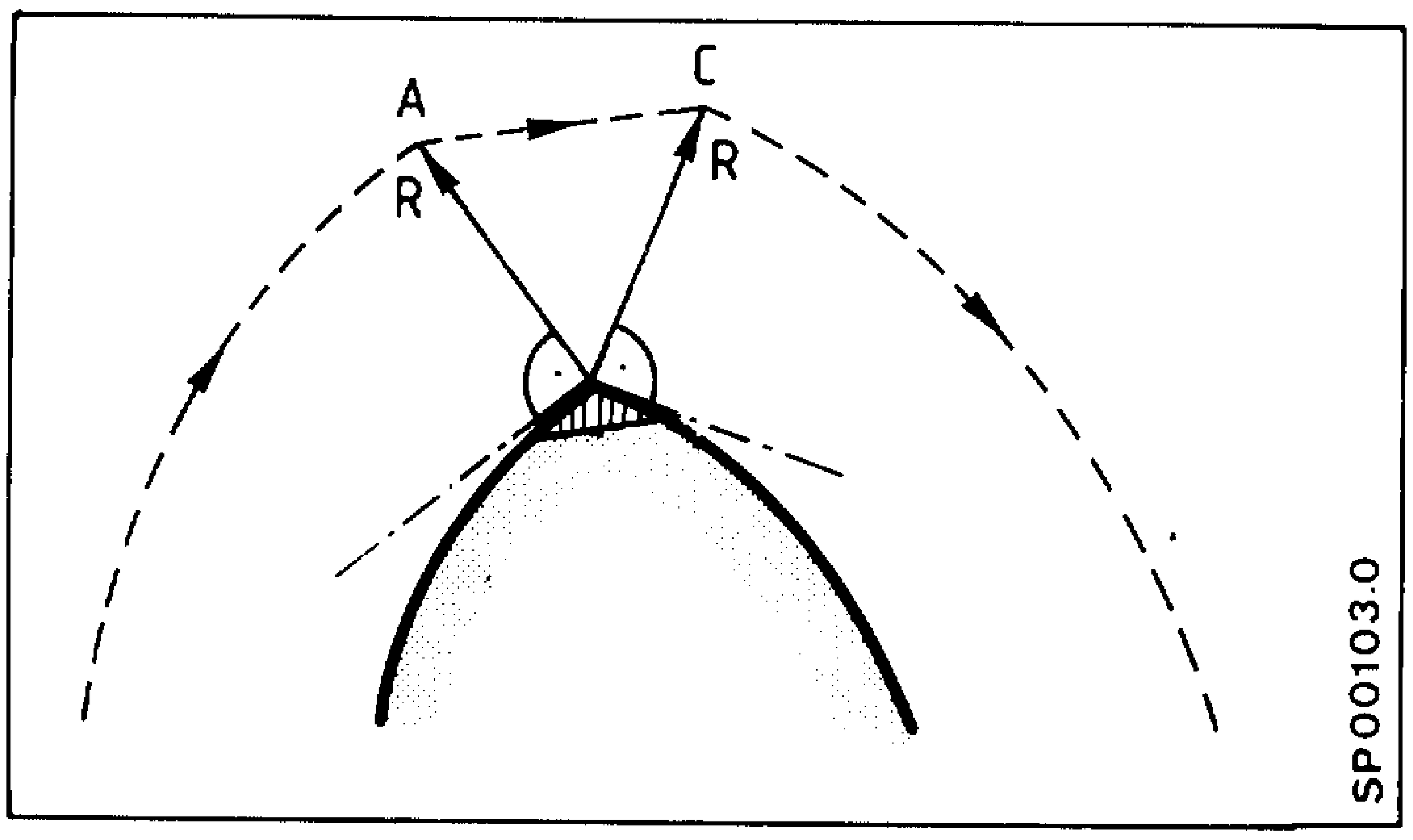
The retraction path in N2 must be greater than twice the tool nose radius (otherwise the tool would execute the movement in the wrong direction.

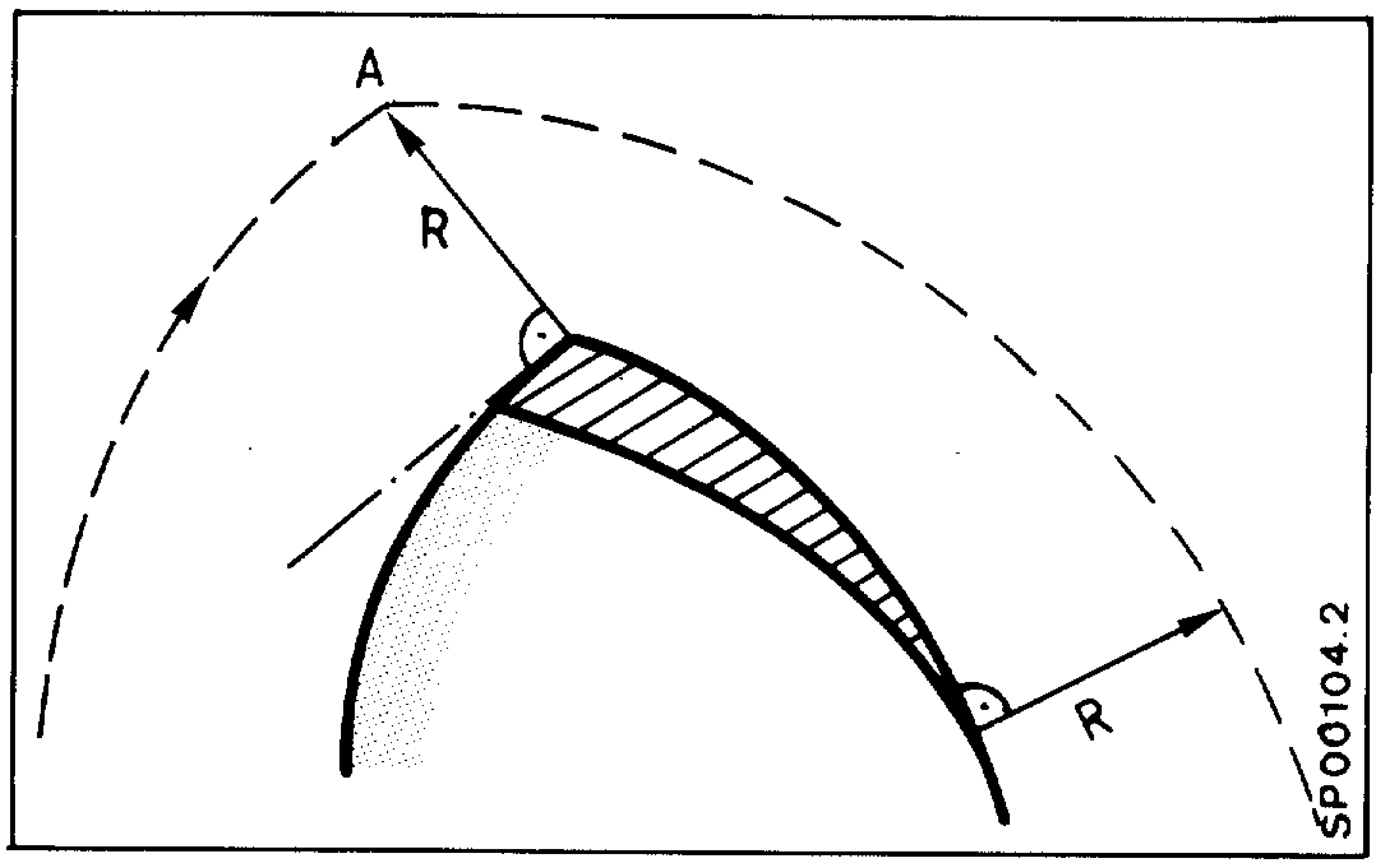
For an outside contour with a circle to circle transition at an obtuse angle, the following applies:



To avoid interruptions to continuous path control caused by intermediate blocks which are too small, paths AB and BC may be omitted by the NC. Dependent on a dimension tolerance d defined during set up (max. 32000 µm) the resultant path generated is as shown.

If X1 and Y1 is less than d, the traversing is direct from A to C.





If X1, Y1, X2, and Y2 are less than d, no path is generated. Traversing is continued from A using the new radius.

8.1.6 CRC in combination with various types of block

Type:

- "Path = 0"

The path addresses are programmed but there is no motion because the path is 0.

Example: N... G91 X0. LF

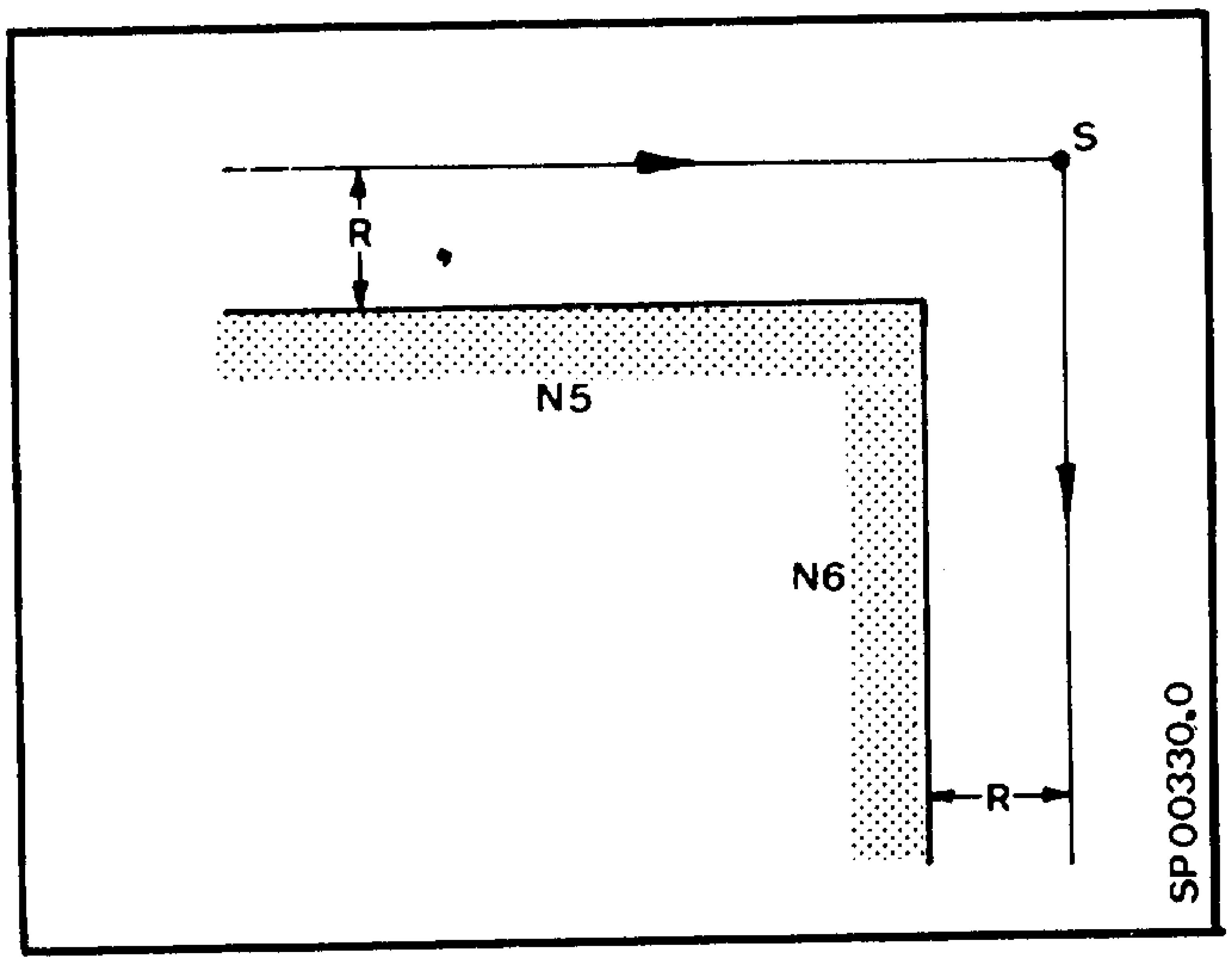
- "Auxiliary functions"

No path addresses programmed but instead auxiliary functions, dwell or a zero offset.

Example:

N... Z1000. LF
N... M08 LF
N... G04 X10. LF
N... G59 Z10. LF
N... T0101 LF

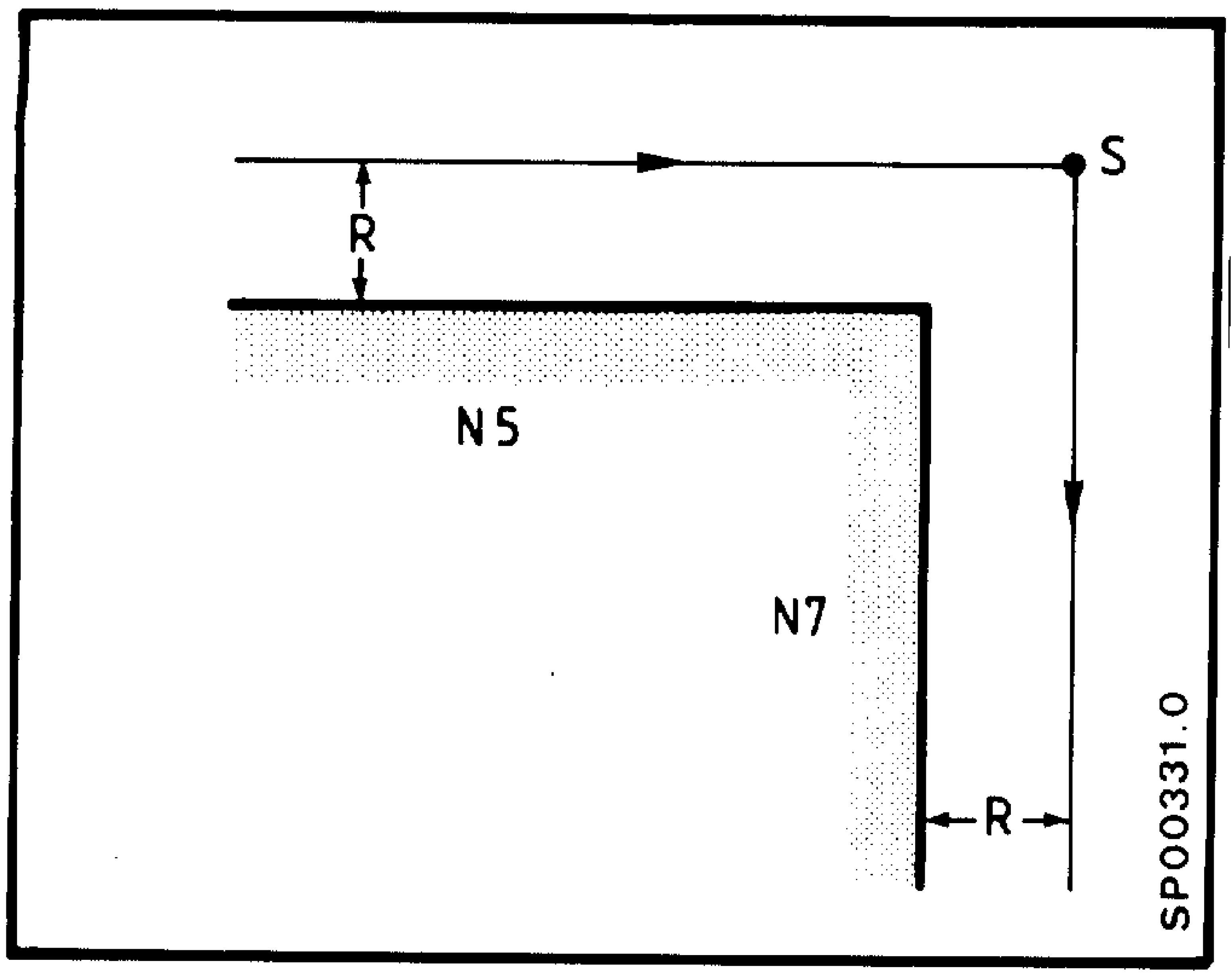
Examples:



Two contour sections

N5 G91 Z100. LF
N6 X-100. LF

One "auxiliary function block" between two contour sections



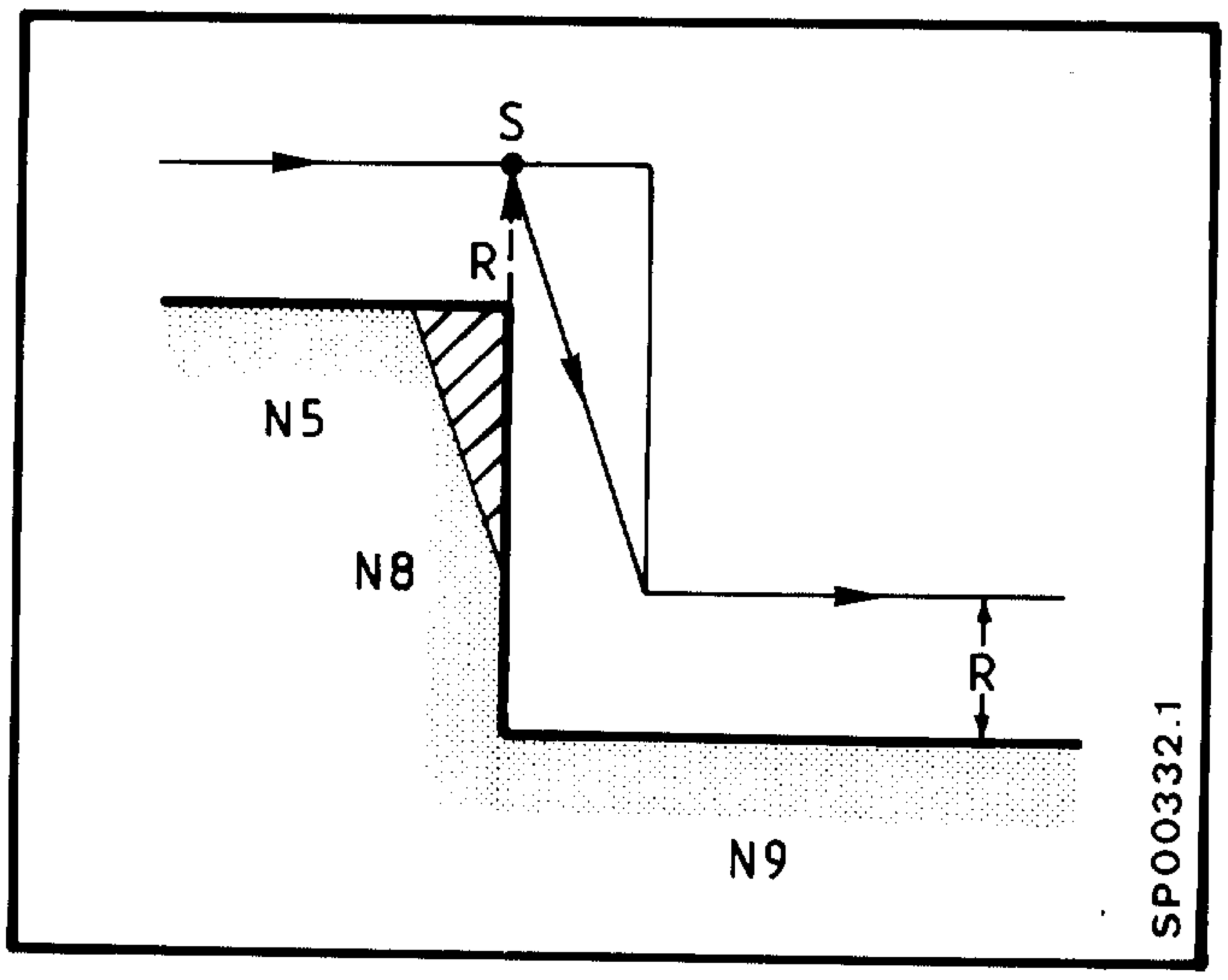
N5 G91 Z100. LF

N6 M08 LF

N7 X-100. LF

Block N6 is executed at point S.

Two "auxiliary function blocks" between two contour sections



N5 G91 Z100. LF

N6 M08 LF

N7 M09 LF

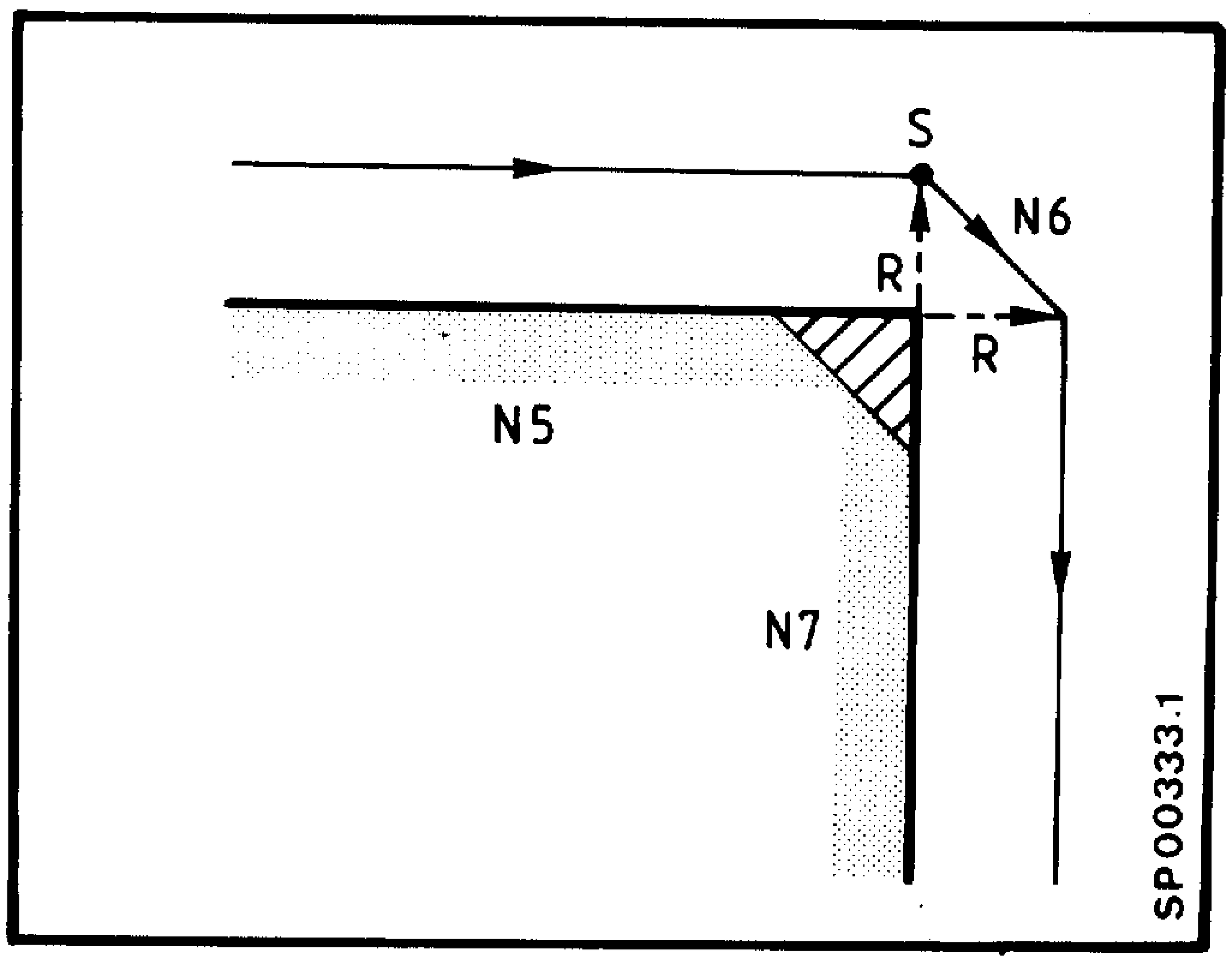
N8 X-100. LF

N9 Z100. LF

Blocks N6 and N7 are executed at point S.

There is a contour error, except at tangential passages.

A block with "path = 0" between two contour sections



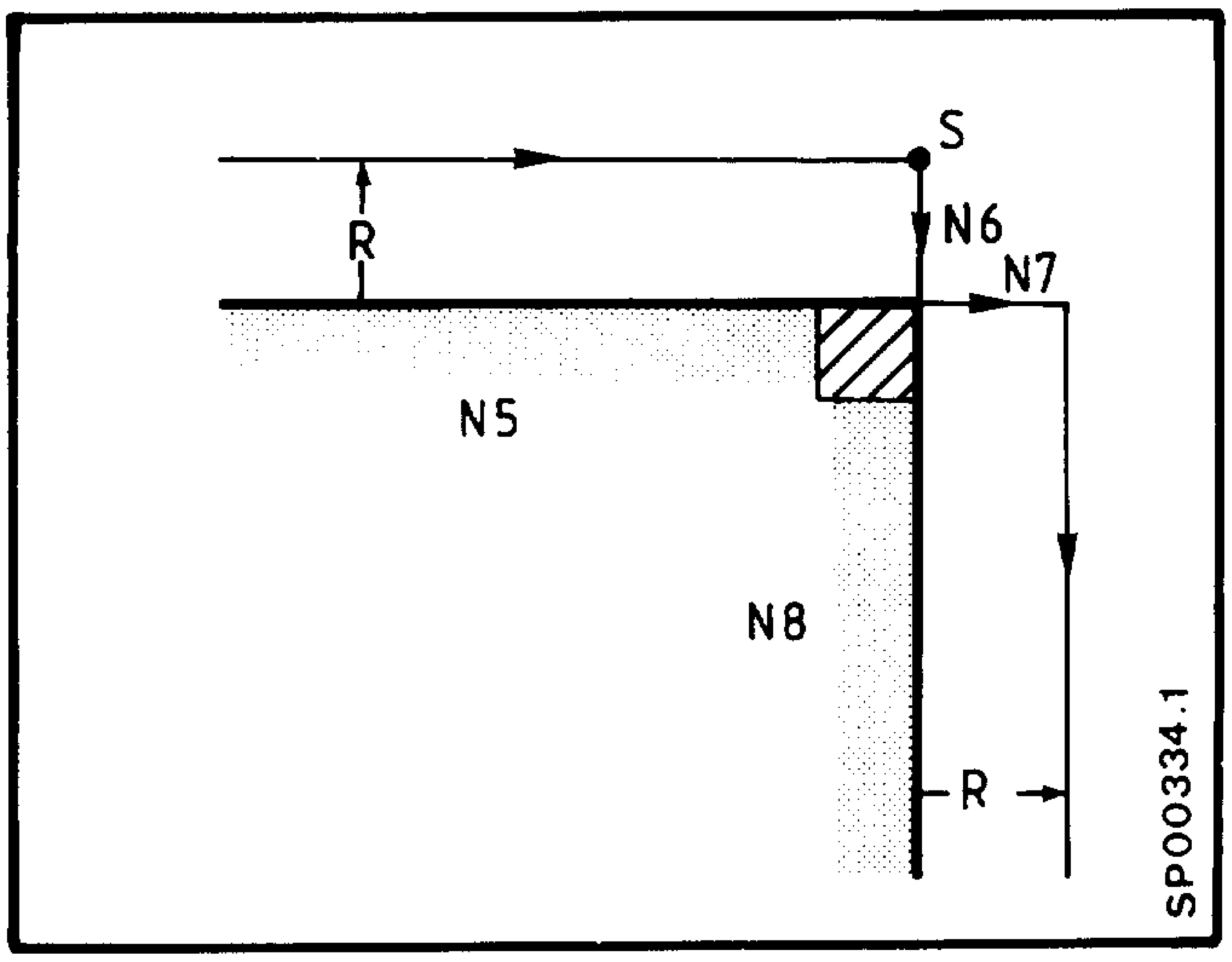
N5 G91 Z100. LF

N6 Z0 LF

N7 X-100 LF

There is a contour error, except at tangential passages.

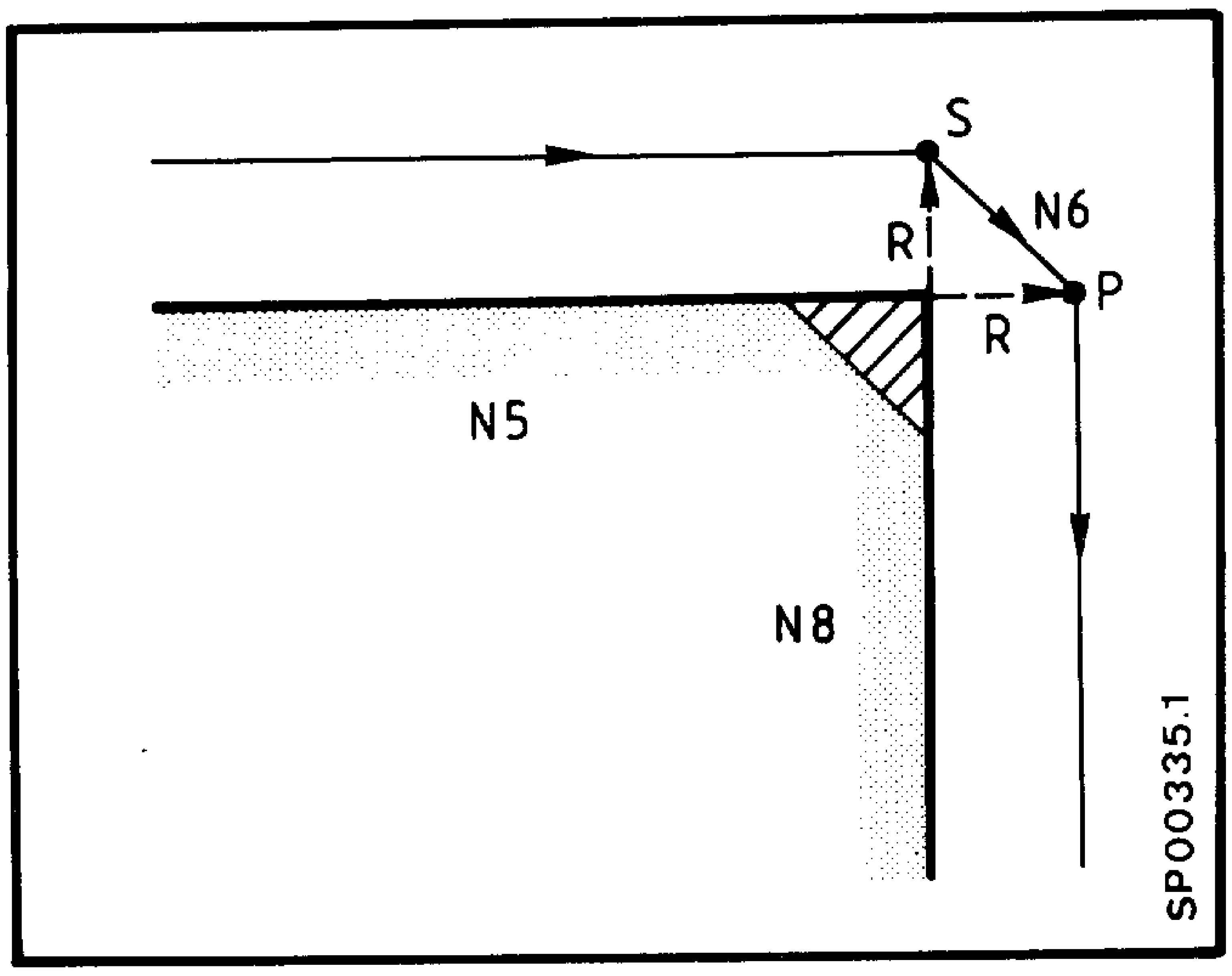
Two blocks with "path = 0" between two contour sections




```
N5 G91 Z100.   LF
N6     Z0.     LF
N7     Z0.     LF
N8     X-100.  LF
```

There is a contour error, except at tangential passages.

One block with "path = 0" and an "auxiliary function" block between two contour sections

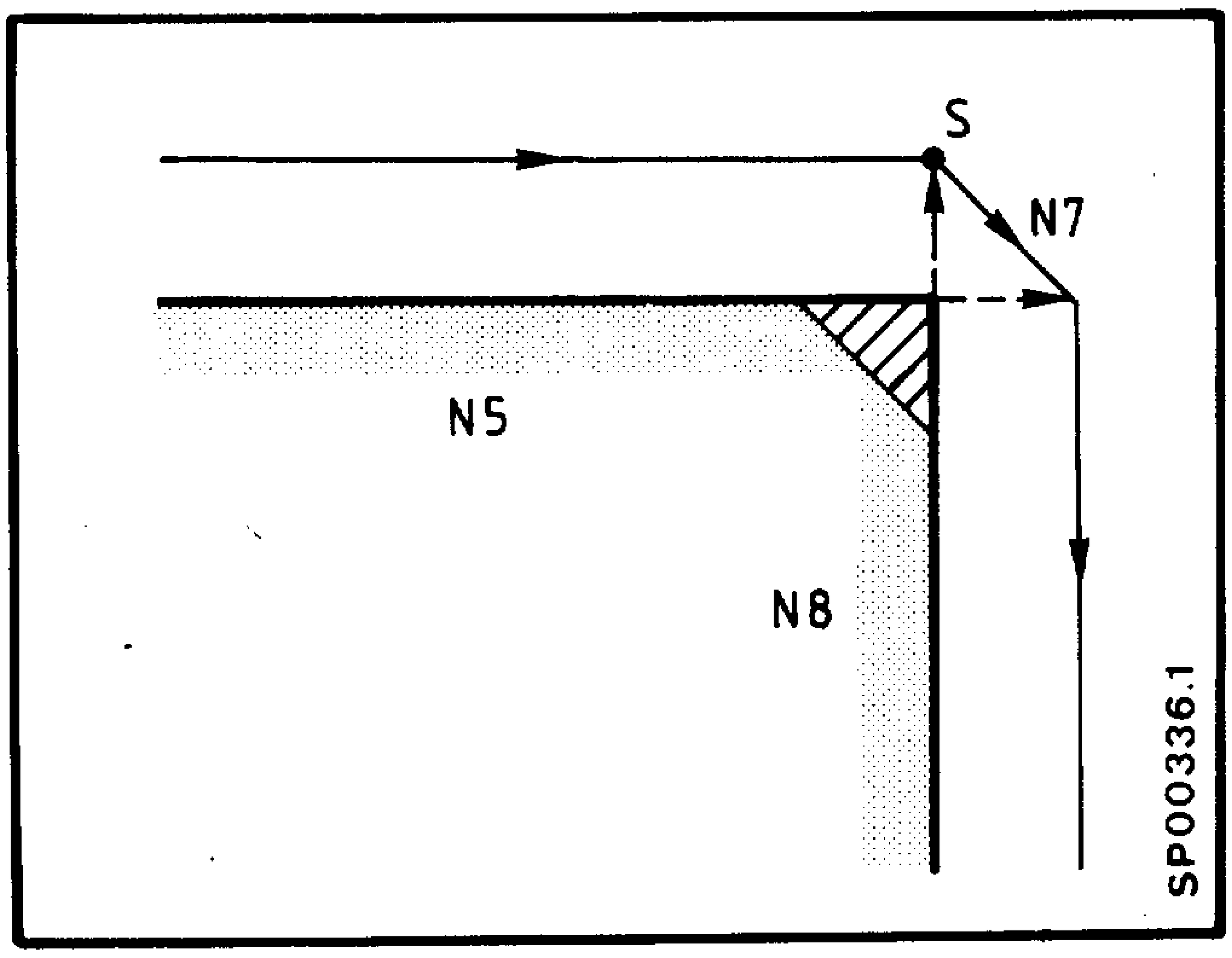



```
N5 G91 Z100.   LF
N6     Z0.     LF
N7 M08         LF
N8     X-100.  LF
```

Block N7 is executed at point P.

There is a contour error, except at tangential passages.

One "auxiliary function" block and one block with "path = 0" between two contour sections



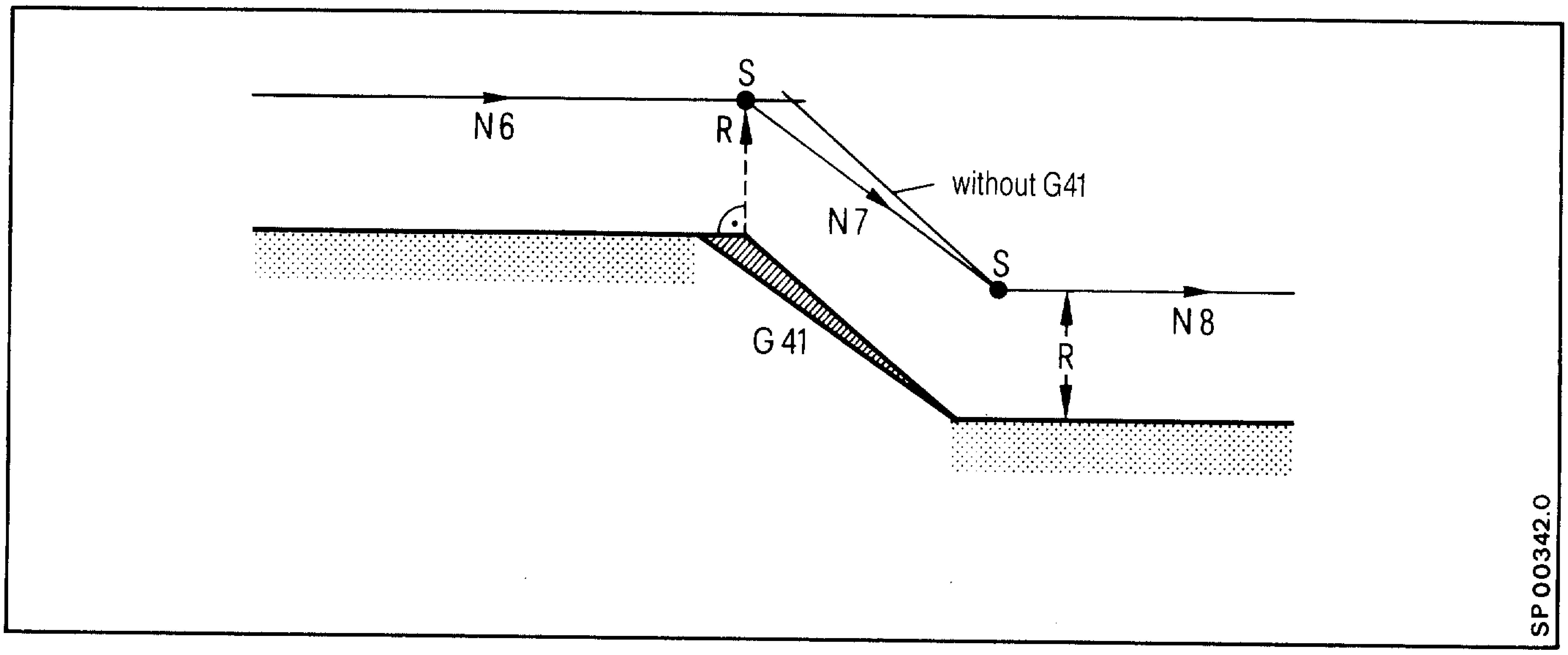

```
N5 G91 Z100.   LF
N6 M08         LF
N7     Z0. LF
N8     X-100   LF
```

Block N6 is executed at point S

There is a contour error, except at tangential passages.

8.1.7 Repetition of previously activated G function (G41, G42) with the same offset number

If a previously programmed G41, G42 is repeated, a vector with the length R perpendicular to the programmed path is set up at the starting point of the following block.



The block start intersection point is calculated for the following block:

```
N5  G91  G41  T1213  X-.... Z....  LF
N6                   Z-....        LF
N7  G41              X-.... Z....  LF  Error: G41 repeated
N8                   Z....         LF
```

8.2 Input systems, diagrams and tables

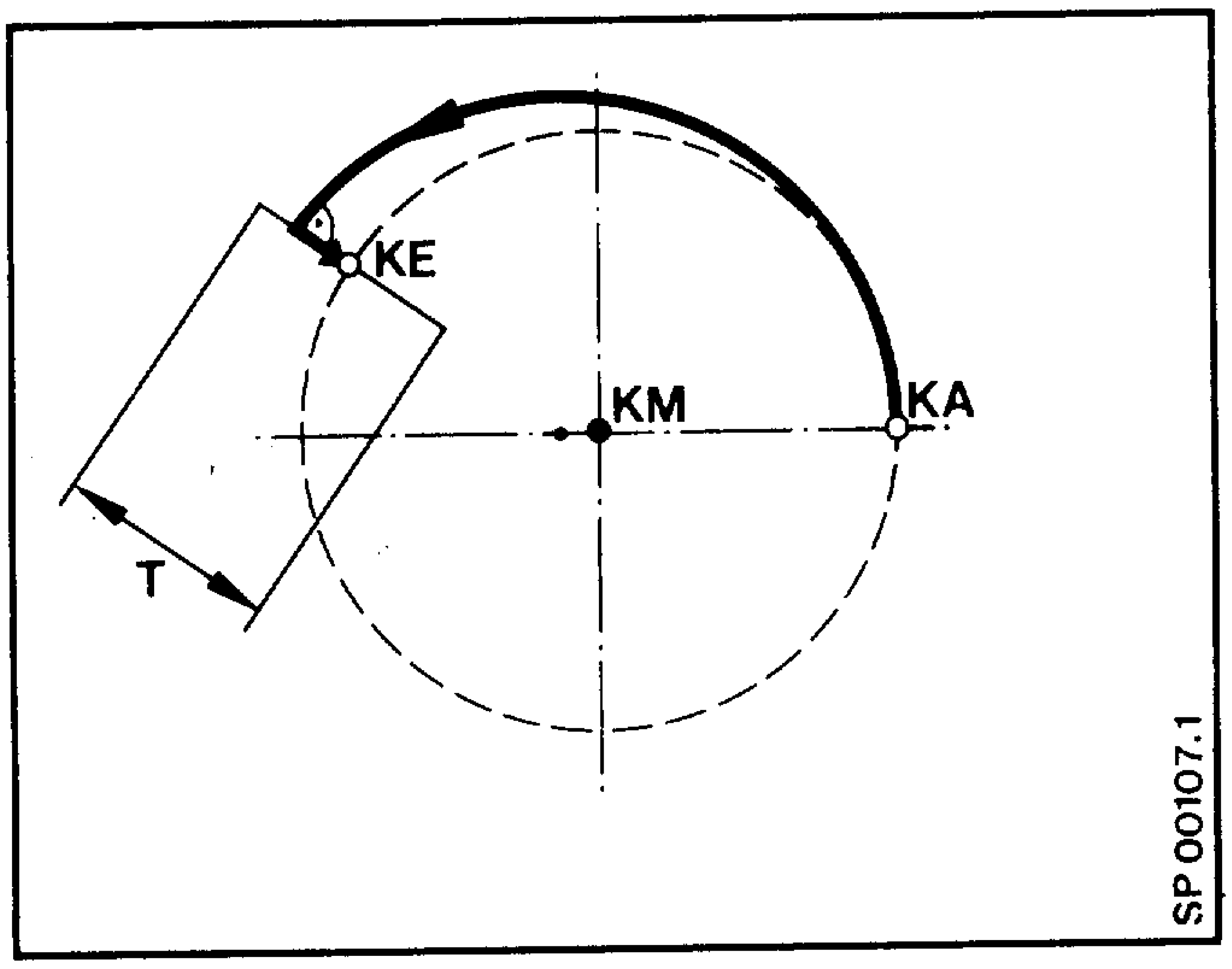
8.2.1 Inexact input of interpolation parameters I and K

The circle end point is monitored and any programming error (assuming it is outside the tolerance range) is recognised. Alarm 308 is displayed and circular interpolation cannot start.

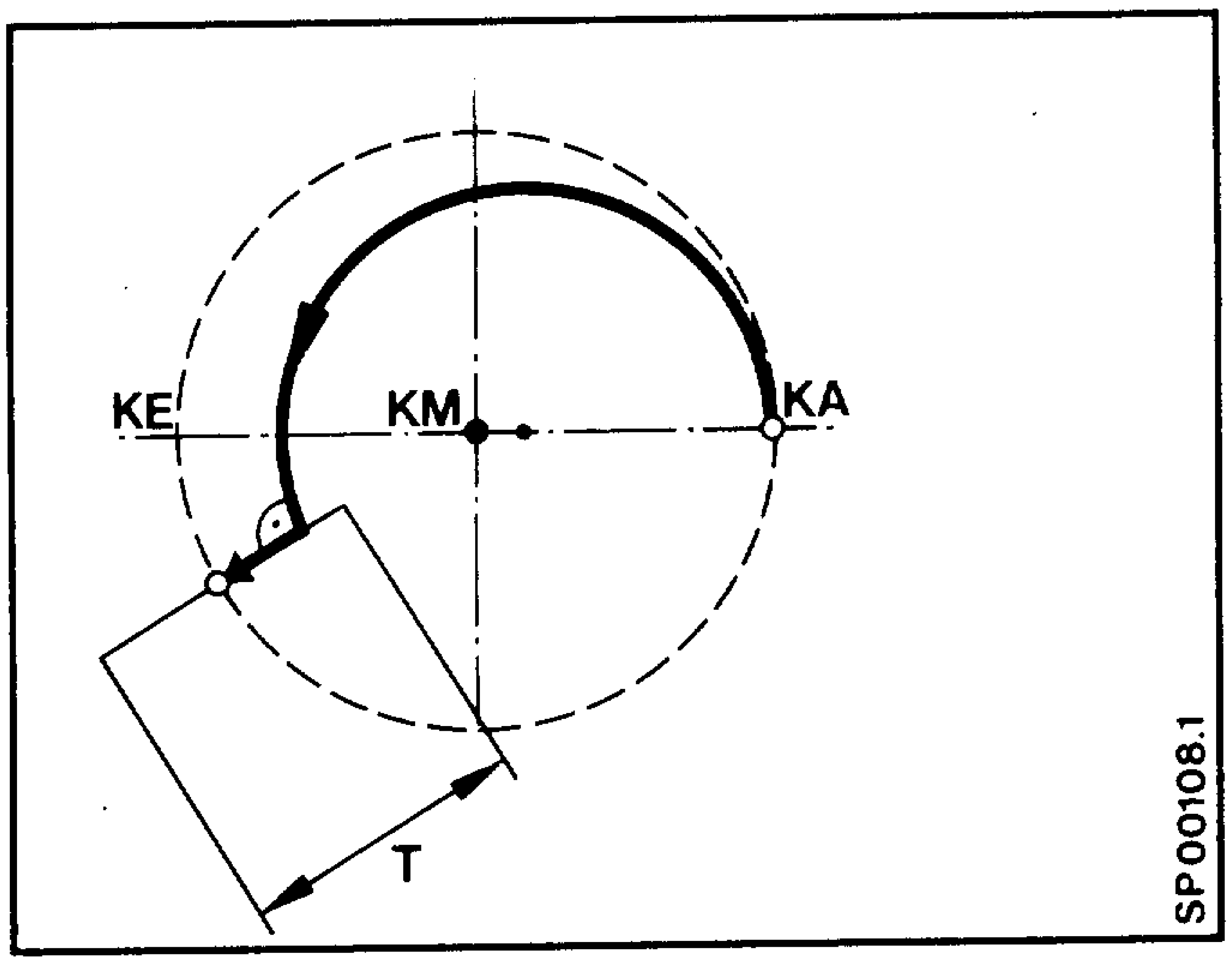
When the programming error lies within the tolerance range, the traverse is made exactly to the end point of the arc but then the path between start and end point is as follows:

Interpolation parameter

too large

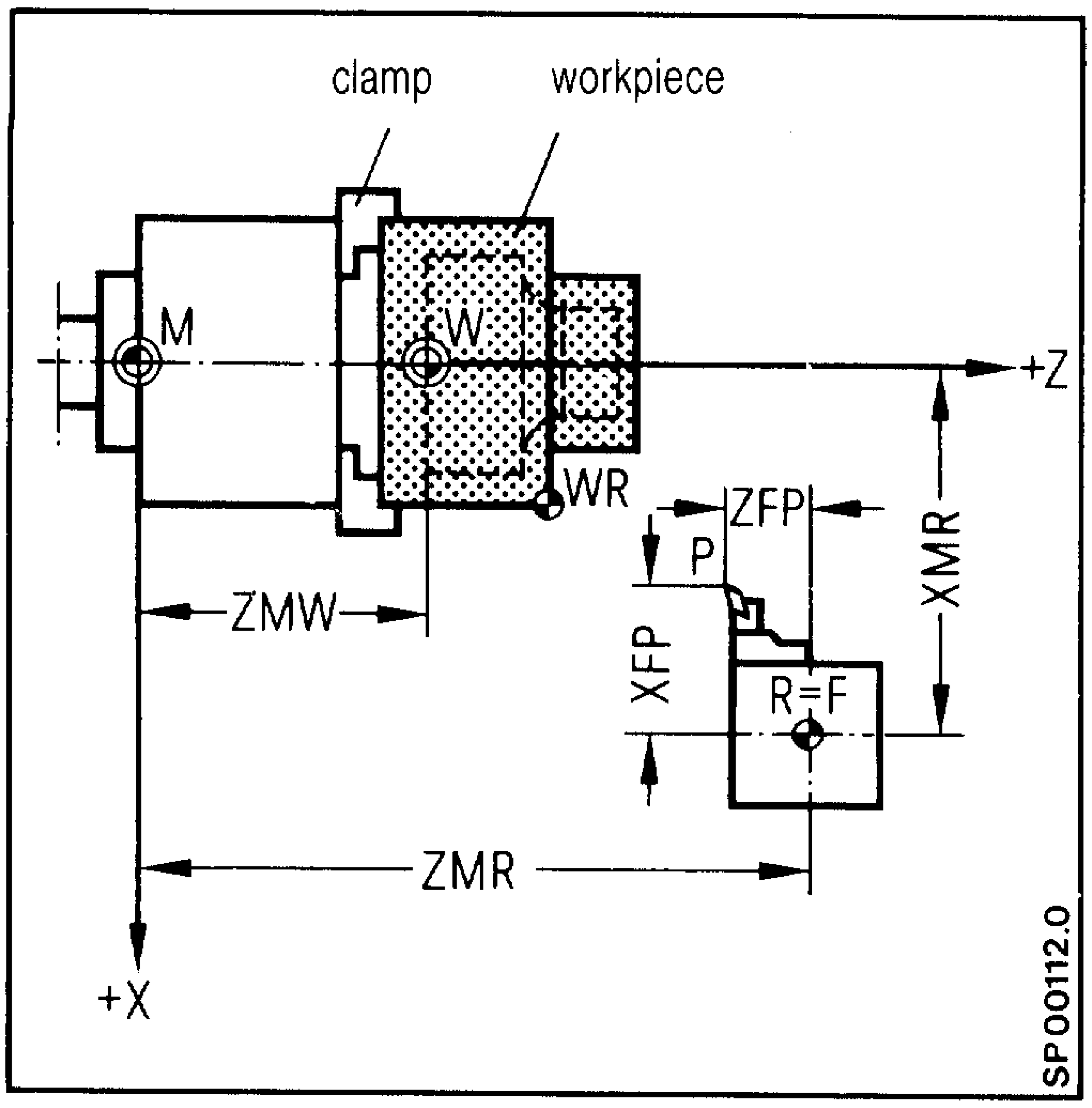


too small



The setting range (machine datum) for the tolerance "T" around the circle end point "KE" is ± 1 µm to ± 32000 µm. This monitoring of the circle end point can be suppressed by using a large value. The tolerance range is entered as a magnitude without sign.

8.2.2 Reference points



P	Tool setting point
M	Machine zero point
W	Workpiece zero point
R	Machine reference point
F	Machine slide reference point
WR	Workpiece reference point
XMR, ZMR, etc	Reference point coordinate for each axis
XMW, ZMW, etc	Sum of zero offsets for each axis

8.2.3 Calculation of the block increment

G91 in the first program block with a motion dimension

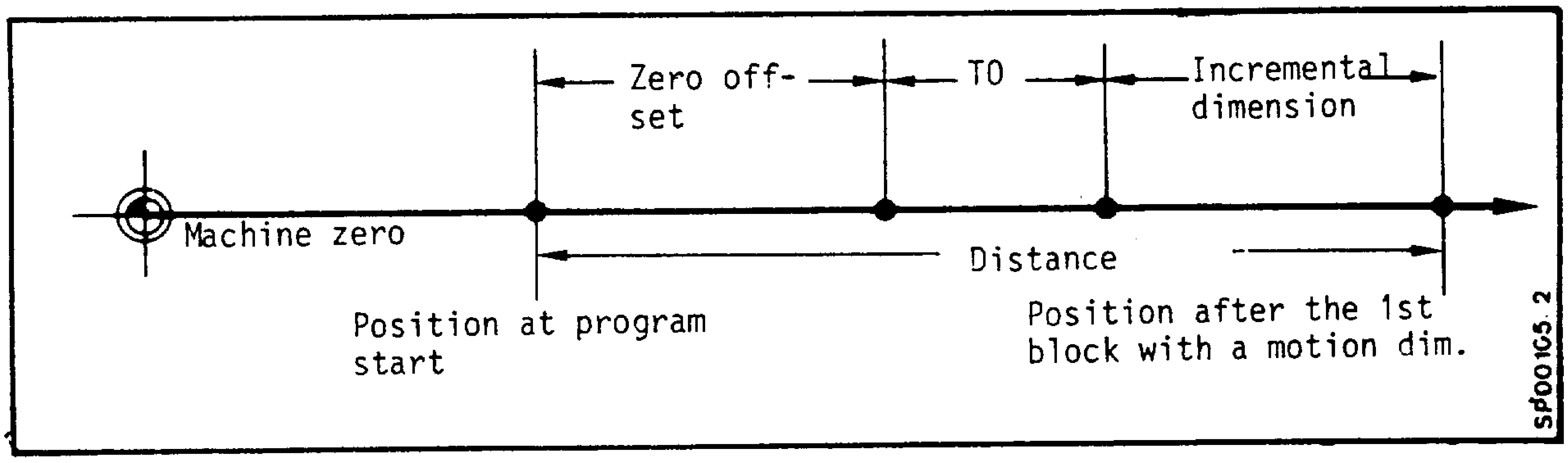
Distance = Incremental dimension + ZO + TO

G91 from the second program block with a motion dimension

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

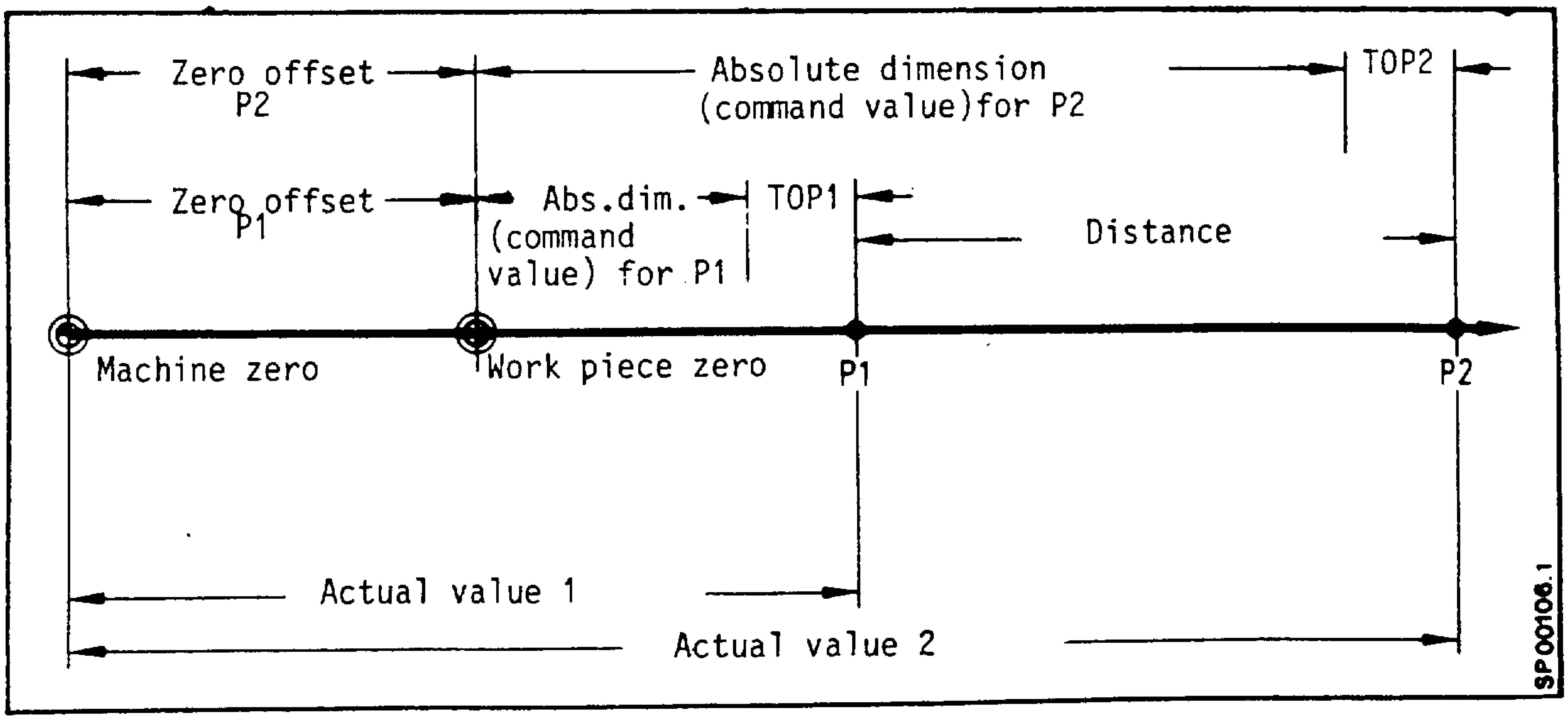
If the ZO and TO are not changed, the following applies:

Distance = Incremental dimension

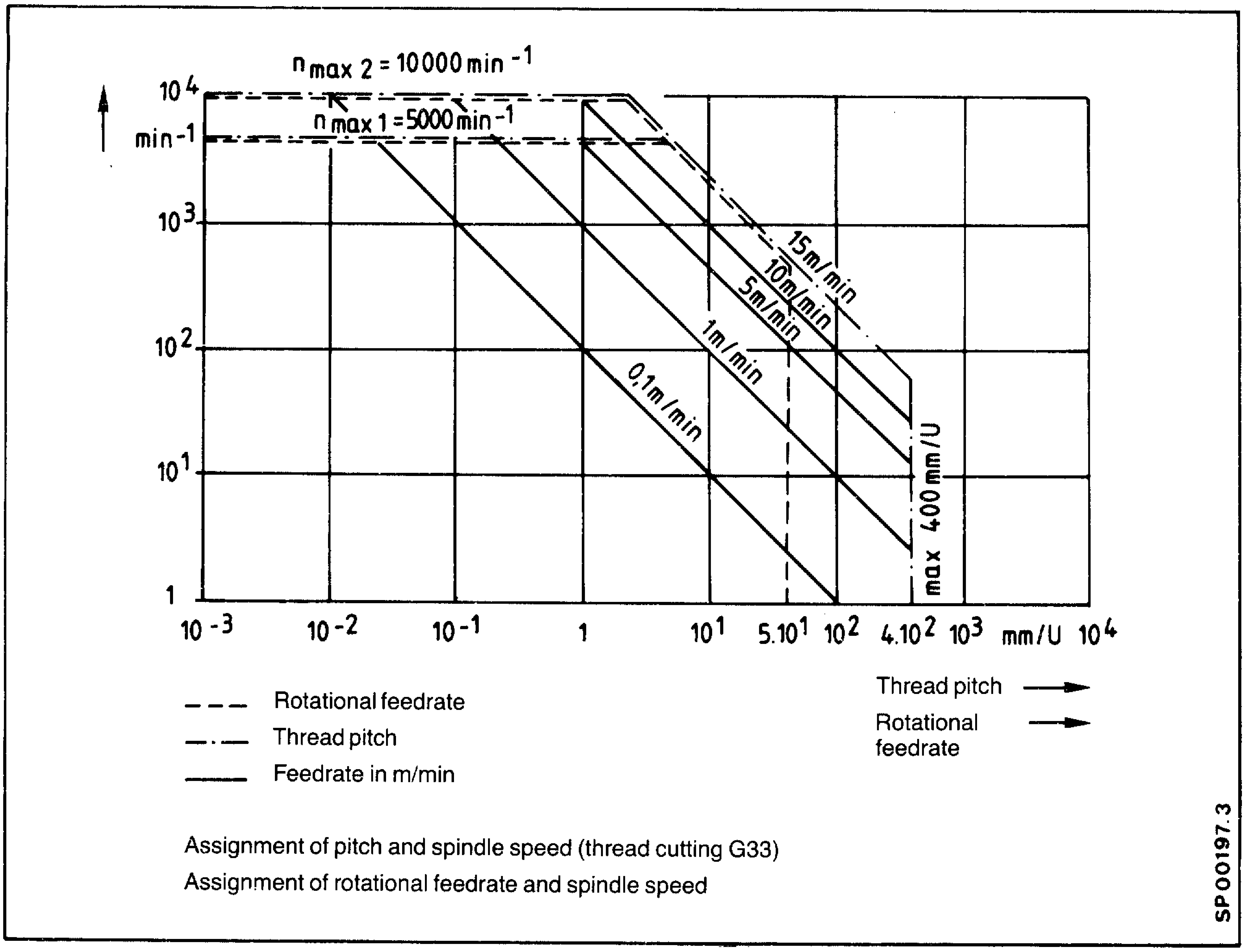


G90 in an arbitrary program block with a motion dimension

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



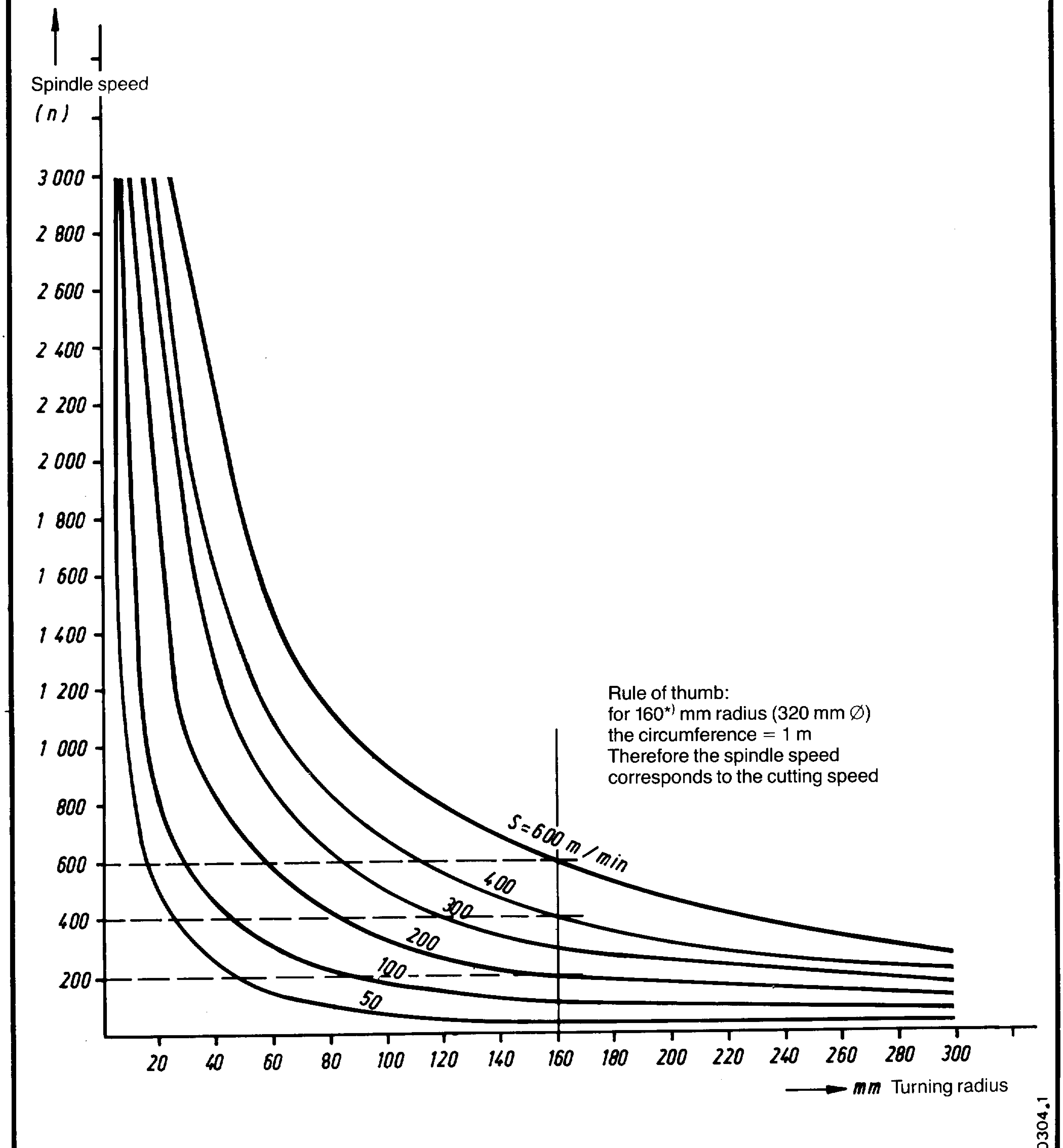
8.2.4 Limit data for rotational feedrate



$n\ max._1$ can be achieved with encoder connected 1:1

$n\ max._2$ can be achieved with encoder connected 1:2

8.2.5 Spindle speed as a function of turning radius with v = constant



Spindle speed and turning radius for v = constant

8.2.6 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)		-		-		± 4.3 [1]	
Chamfer (B-); Radius (B)		3.3	10^{-3} mm	2.4	10^{-4} inch	-	10^{-3} degrees
Zero offset		± 5.3 [1]		± 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 rev/min	4	1 or 0.1 rev/min		
Linear feedrate (F)		5	mm/min	3.1	10^{-1} inch/min		5 degrees/min
Rotary feedrate (F)		2.3	10^{-3}mm/rev	1.4	10^{-4}inch/rev		
Tool offset	Length	± 3.3	10^{-3} mm	± 2.4	10^{-4} inch		
	Radius	± 3.3		± 1.4			
Dwell time	X	5.3	10^{-3} sec		10^{-3} sec		
	F	X		2.3			
Angle						3.5	10^{-5} degrees
Angle for oriented spindle stop 3)						3.1	10^{-1} degrees
R Parameter		Dimension depends on association (internal floating point) all combinations (2 decades for call)					
G. preparatory functions		2		2			
M functions		2		2			
Block number		4		4			
H functions		4		4			
Special functions @		2		2			

The parameters (R00-R29) 2) must always be written as 2 decades. For other functions (except address L) the leading zeros can be omitted.

1) For basic control 0, 1, 2: decade less before the decimal point
2) For basic control 0, 1, 2: R00-R49
3) For basic control 3, 4 and 3TT

8.2.7 Code Table

Code table

ISO / DIN 66024

characters	P	7	6	5	4		3	2	1	bit No. (check bit P)	
	6	7	6	5	4	T	3	2	1	track No. (clock pulse track T)	
	hole combination									allowed in: program	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	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	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	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 244 B

	characters	8	7	6	P	4		3	2	1		
		8	7	6	5	4	T	3	2	1		
		hole combination									allowed in: program	leader
	no hole						•				–	x
	RT			●		●	•		●		–	x
	TAB			●	●	●	•	●	●		x	x
	< = EOB	●					•				x	–
	LC)		●	●	●	●	•		●		–	x
	ZWR				●		•				x	x
1)	(≡				●	●	•		●		x	x
1)	) ≡		●			●	•		●		x	x
	ER					●	•		●	●	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	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	x
	p		●		●		•	●	●	●	x	x
	q		●		●	●	•				x	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 tracks for beginning and end of comments.

2) Code can be defined in 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	%				1.	Program start
	EOR ... EOB	% ... 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
	. + -	. + -				1.1 1.2 1.2	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 3.15 3.15 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
G2	g	G	▶04*	▶04*	▶04*	3.8	Dwell time under the X address
G3	g	G	▶09	▶09	▶09	3.7	Reduce speed, exact stop
G5	g	G	40● 41 42	40● 41 42	40● 41 42	3.14 3.14 3.14	Cancel tool nose radius compensation Tool nose radius compensation left Tool nose radius compensation right
G7	g	G	▶53	▶53	▶53	3.10.3	Suppress the zero offsets
G8	g	G	54 55 56 57	54 55 56 57	54 55 56 57	3.10.1 3.10.1 3.10.1 3.10.1	Select zero offset 1 Select zero offset 2 Select zero offset 3 Select zero offset 4
G9	g	G	▶59*	▶59*	▶59*	3.10.2	Programmable zero offset
G11	g	G	70 71	70 71	70 71	3.9 3.9	Input inch Reset state via machine data Input metric
G12	g	G	90● 91	90● 91	90● 91	3.1 3.1	Absolute data input Incremental data input
G13	g	G	▶92*	▶92*	▶92*	3.13	Spindle speed limit under S
			- -	- -	▶92* ▶92*	3.5.22 3.16	Spindle-related starting angle offset under A Smoothing exponent for threads under T
G14	g	G	94 95 96 -	94 95 96 -	94 95 96 97	3.11 3.11 3.12 3.12	Feedrate F in .../min Feedrate F in .../min Constant cutting speed under S Deactivate G96, store last speed value of G96
G16	g	G	–	–	37 36 ● 39	3.17 3.17	Coordinate transformation "TRANSMIT" ON Coordinate transformation "TRANSMIT" OFF Coordinate rotation for inclined workpieces and tools
	x	X	0 to ±9999.999 0 to ±999.9999 0 to ±999.9999	0 to ±99999.999 0 to ±99999.999 0 to ±99999.999	0 to ±99999.999 0 to ±99999.999 0 to ±99999.999	2.1 3.8	Position information in mm Position information in inch Dwell time in seconds
	z	Z	0 to ±9999.999 0 to ±999.9999	0 to ±99999.999 0 to ±99999.999	0 to ±99999.999 0 to ±99999.999	2.1	Position information in mm Position information in inch

* No other preparatory functions can be written in this block
● Reset state (Reset state after RESET M02/M30 and switching on the control)
▶ Valid for the block, all others self-retaining (modal)

Group	EIA	ISO	Code for basic control 0,1,2	3	4/4B/4C	Section	Function and meaning
	i k	I K	0 to ±9999.999	0 to ±99999.999	0 to ±99999.999	3.4	Circle interpolation parameter in mm
			0 to ±999.9999	0 to ±99999.999	0 to ±99999.999		Circle interpolation parameter in inch (I for X axis, K for Z axis)
			0.001 to 400.000	0.001 to 400.000	0.001 to 400.000	3.5	Thread pitch in mm
			0.0001 to 15.0000	0.0001 to 15.0000	0.0001 to 15.0000		Thread pitch inch
		A	-	-	0 to 359.99999	3.15	Angle in degrees for polar coordinates
	b	B	+0.001.. +999.999	+0.001.. +999.999	+0.001.. +999.999	6.0	Radius of inserted circle in mm
			+0.0001.. +99.9999	+0.0001.. +99.9999	+0.0001.. +99.9999		Radius of inserted circle in inch
	b	B	-0.001.. -999.999	-0.001.. -999.999	-0.001.. -999.999	6.0	Chamfer in mm
			-0.0001.. -99.9999	-0.0001.. -99.9999	-0.0001.. -99.9999		Chamfer in inch
	r	R	00..49	00..99	00..99	5.0	Parameters
			0.. 9999 999	0.. 99999 999	0.. 99999 999		Parameter value in mm/inch, also ± Value allocation for all addresses except N _ with preparatory functions with sign
	f	F	1.. 15000	1... 15000	1... 15000	3.11	Feedrate in mm/min (G94)
			0.1.. 590	0.1... 590	0.1... 590		Feedrate in inch/min (G94)
			0.001.. 50.000	0.001... 50.000	0.001... 50.000		Feedrate in mm/rev (G95, G96)
			0.0001.. 2.0000	0.0001... 2.0000	0.0001... 2.0000		Feedrate in inch/rev (G95, G96)
	s	S	1..99	1..99	1..99	4.1	Spindle speed (geometric sequence)
			1..9999	1..9999	1..9999	4.1	Spindle speed in rev./min or 0.1 rev/min Spindle speed limitation
			1..9999	1..9999	1..9999	3.12	Constant cutting speed in ft/min or 0.1 ft/min
			-	0.5...359.5	0.5...359.5		Spindle stop in degrees, distance to zero mark of sensor
	t	T	01..... 99..	01..... 99..	01..... 99..	4.2	Tool number
			..01... ..16	..01... ..16	..01... ..16	4.2	Tool offset number
			..00	..00	..00	4.2	Suppress tool offset
	h	H	-	-	1....9999	4.2	Auxiliary functions
M1	m	M	00	00	00	4.3	Programmed stop unconditional
	m	M	02	02	02	4.3	End of program
			17	17	17	4.3	End of subroutine, in last block of subroutine
			30	30	30	4.3	End of program (see M02)
M2	m	M	03	03	03	4.3	Spindle direction clockwise
			04	04	04	4.3	Spindle direction counter-clockwise
			05●	05●	05●	4.3	Spindle stop
			-	▶19	▶19	4.3	Oriented spindle stop, sine angle in degrees

Group	EIA	ISO	Code for basic control 0,1,2	3	4/4B/4C	Section	Function and meaning
M3	m	M	00 ..	00 ..	00 ..	4.3	Miscellaneous function, unassigned (except for groups M1/M2)
	l	L	01 .. .. 99 ..	01 .. ... 99 ..	01 .. ... 999..	1.8	Subroutine number (91 to 99 for cycles)
			..01 99	..01 99	..01 99	1.8	Number of passes
	@	@	00	00	00	5.5	Unconditional jump
			01	01	01		Conditional jump-equal to
			02	02	02		Conditional jump-greater than
			03	03	03		Conditional jump-greater than or equal to
			31	31	31	5.6	Clear buffer
			-	10	10	5.7	Square root
			-	15	15	5.8	Sine
			-	18	18	5.9	Arctan
			-	20	20	5.10	Load address parameter
			-	90..93	90..93	5.11	Address parameter
			-	22	22	5.12	Intersection calculation
			-	29	29	5.13	Load/read system memory
					21*	5.14	Reference editing
	EOB	LF				1.4	End of block

***) 48**

8.4 Special cases

8.4.1 Special case "Delete distance to go"

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

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

- clear buffer with @ 31
- load actual value with @ 29
- traverse to actual position with G90

Programming example

```
.
.
M..                      - activate the "Delete distance to go" function
G90 G01 X100             - delete distance to go in X-axis e.g. at 60 mm
M..                      - disable "Delete distance to go" function
@31                      - clear buffer
R00 1001 @2919312R00 - load X-axis actual value
G90 XR93                 - traverse to actual position
G00 G91 X 60             - traverse 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 3T/3TT Basic Version 4C
Programming Guide **Order No.:** 6ZB5 410-0AC02-0BA0 **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

Energy and Automation Group
Department for Numerical Controls and
Drives for machine tools and Robots

Published by Information and Training Subdivision
Postfach 4848, D-8500 Nürnberg 1
Federal Republic of Germany

Subject to change without prior notice

Siemens Aktiengesellschaft

Order No. 6ZB5410-0AC02-0BA0
Printed in the Fed. Rep. of Germany
886353 PA 6891.

progress in automation: Siemens