

P R O G R A M M I N G

Contents

1. GENERAL.....	4
1.1 TOOL MOVEMENT ALONG WORKPIECE - INTERPOLATION.....	4
1.2 FEED.....	6
1.3 PART DRAWING AND TOOL MOVEMENT.....	7
1.3.1 Reference (Zero) Point of the Machine.....	7
1.3.2 Coordinate System on Part Drawing and Coordinate System on CNC.....	8
1.3.3 Commands for Moving the Tool.....	11
1.4 CUTTING SPEED - SPINDLE SPEED FUNCTION.....	12
1.5 SELECTION OF TOOL.....	13
1.6 SPECIFIC MACHINE OPERATIONS - MISCELLANEOUS FUNCTION.....	13
1.7 PROGRAM CONFIGURATION.....	14
1.8 TOOL FIGURE AND TOOL MOTION.....	17
1.9 TOOL MOVEMENT RANGE - STROKE.....	18
2. CONTROLLED AXES.....	20
2.1 CONTROLLED AXES.....	20
2.2 NAMES OF AXES.....	20
2.3 INCREMENT SYSTEM.....	20
2.4 MAXIMUM STROKE.....	20
4. INTERPOLATION FUNCTIONS.....	24
4.1 POSITIONING (G00).....	24
4.2 SINGLE DIRECTION POSITIONING (G60).....	24
4.3 LINEAR INTERPOLATION.....	26
4.4 CIRCULAR INTERPOLATION (G02, G03).....	29
4.5 SKIP FUNCTION.....	33
5. FEED FUNCTIONS.....	37
5.1 GENERAL.....	37
5.2 RAPID TRAVERSE.....	40
5.3 CUTTING FEED.....	40
5.4 CUTTING FEEDRATE CONTROL.....	43
5.4.1 Exact Stop (G04).....	44
5.5 DWELL (G04).....	44
6. REFERENCE POSITION.....	45
7. COORDINATE SYSTEM.....	50

7.1 MACHINE COORDINATE SYSTEM	50
7.2 WORKPIECE COORDINATE SYSTEM.....	51
7.2.1 Setting a Workpiece Coordinate System	51
7.2.2 Selecting a Workpiece Coordinate System	52
7.2.3 Workpiece Coordinate System Change	54
7.3 PLANE SELECTION.....	57
8. COORDINATE VALUES AND DIMENSIONS.....	58
8.1 ABSOLUTE AND INCREMENTAL PROGRAMMING (G90 AND G91)	58
8.2 INCH/METRIC CONVERSION (G20 AND G21)	60
8.3 DECIMAL POINT PROGRAMMING.....	61
9. SPINDLE SPEED FUNCTION (S FUNCTION)	63
9.1 SPECIFYING THE SPINDLE SPEED WITH A BINARY CODE.....	63
9.2 DIRECT SPECIFYING THE SPINDLE SPEED	63
10. TOOL FUNCTION (T FUNCTION).....	64
10.1 TOOL CHANGE FUNCTION	64
11. AUXILIARY FUNCTION	65
11.1 AUXILIARY FUNCTION (M FUNCTION)	65
11.2 SECOND AUXILIARY FUNCTION (B CODE)	67
12.1 PROGRAM COMPONENTS OTHER THAN PROGRAM SECTIONS	70
12.2 PROGRAM SECTION CONFIGURATION	71
12.3 SUBPROGRAM	76
13. FUNCTIONS TO SIMPLIFY PROGRAMMING	81
13.1 CANNED CYCLE	81
13.1.1 High-speed Peck Drilling Cycle (G73).....	88
13.1.2 Left-handed Tapping Cycle(G74)	91
13.1.3 Fine Boring Cycle (G76).....	94
13.1.4 Drilling Cycle, Spot Drilling Cycle (G81)	97
13.1.5 Drilling Cycle, Counter Boring Cycle (G82)	100
13.1.6 Peck Drilling Cycle (G83)	102
13.1.7 Tapping Cycle (G84).....	106
13.1.8 Boring Cycle (G85)	109
13.1.9 Boring Cycle (G86)	111
13.1.10 Boring Cycle, Back Boring Cycle (G87).....	115
13.1.11 Boring Cycle (G88)	118
13.1.12 Boring Cycle (G89)	120

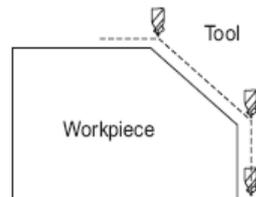
13.1.13 Canned Cycle Cancel	124
13.2 EXTERNAL MOVEMENT FUNCTION (G81)	127
14. COMPENSATION FUNCTION	128
14.1 TOOL LENGTH OFFSET	128
14.2 OVERVIEW OF CUTTER COMPENSATION	133
14.3 DETAILS ON CUTTER COMPENSATION.....	140
14.3.1 General.....	140
14.3.2 Tool Movement in Start-up	141
14.3.3 Tool Movement in Offset Mode	144
14.3.4 Tool Movement in Offset Mode Cancel	161
14.3.5 Interference Check.....	167
14.3.6 Overcutting by Cutter Compensation.....	173
14.3.7 Input Command From MDI	176
14.4 TOOL COMPENSATION VALUES, NUMBER OF COMPENSATION VALUES AND ENTERING VALUES FROM THE PROGRAM (G10)	177
14.5 Read and write parameters from program (G10).....	180
15. CUSTOM MACRO	180
15.1 CUSTOM MACRO COMMAND	182
15.1.1 Single Call.....	182
15.1.2 Modal Call	182
15.1.3 Argument Specification	183
15.2 CUSTOM MACRO BODY	184
15.2.1 Variables	185
15.2.2 Kind of Variables	186
15.2.3 Macro Instructions type A	189
15.2.4 Macro instructions type B	190
15.2.5 Macro instructions commands type A and type B.....	192
15.2.4 Notes and Cautions Concerning the User Macro	201

1. GENERAL

1.1 TOOL MOVEMENT ALONG WORKPIECE - INTERPOLATION

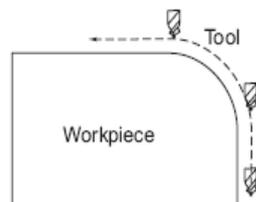
The tool counters straight lines and arcs following the programmed trajectory along the workpiece.

◆ Tool movement along a straight line



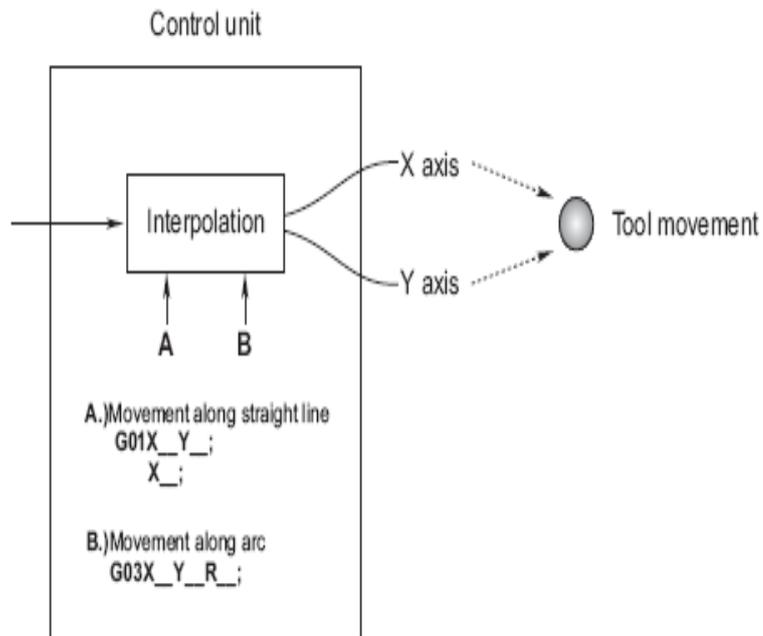
```
Program  
G01X__Y__;  
X__;
```

◆ Tool movement along an arc



```
Program  
G03X__Y__R__;
```

Symbols of the programmed commands **G01**, **G02**, ... are called preparatory functions. They specify the type of interpolation **CNC** executes between points set by the user.

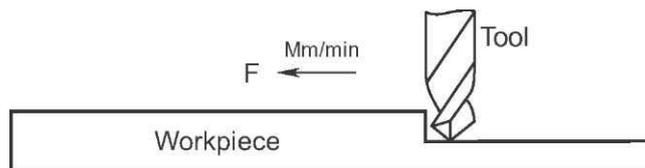


Note:

Some machines request physical movement of the table instead, but this manual assumes that tool is moving always along the workpiece.

1.2 FEED

Movement of the tool at a specified speed for cutting a workpiece is called feed.



Feedrate is specified with values. For example, if a feedrate of 150 mm/min must be specified, the program is as follows:

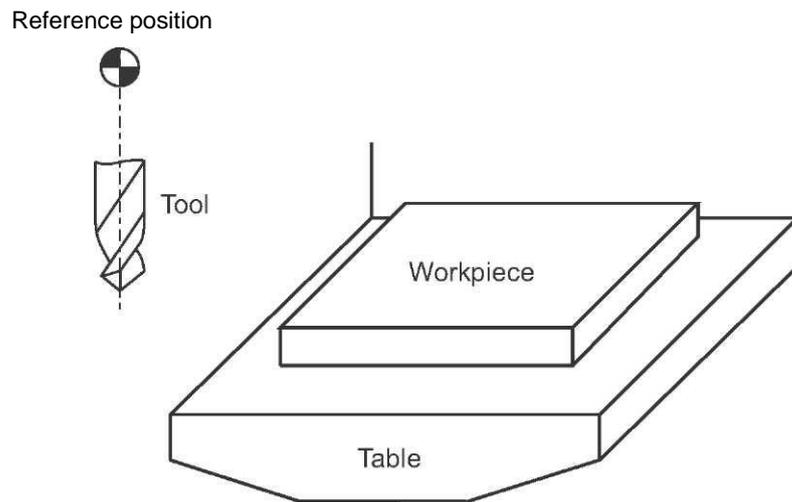
F150.0

This function determines feedrate.

1.3 PART DRAWING AND TOOL MOVEMENT

1.3.1 Reference (Zero) Point of the Machine.

Any machine equipped with **CNC** unit provides a fixed position which is called reference (zero) point. Generally, tool change and absolute zero point programming is performed in this position.



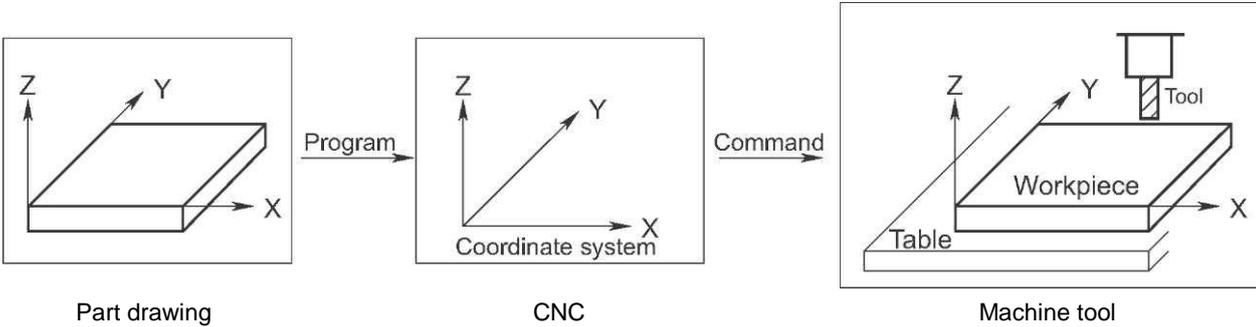
The tool can be moved to the reference position in one of the following ways:

1) Manual reference position return - reference position return is performed manually usually by buttons of the panel

2) Automatic reference position return - in general, manual reference position return is performed right after switching the power on

If it is necessary to be performed reference position return (for example for tool change), this is executed thoroughly automatically by the CNC control after specifying a command.

1.3.2 Coordinate System on Part Drawing and Coordinate System on CNC



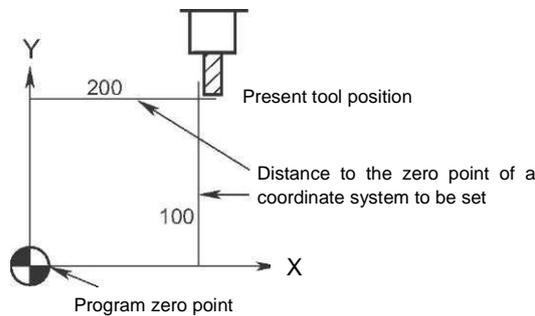
The following two coordinate systems specify two different positions:

1) Coordinate system on part drawing

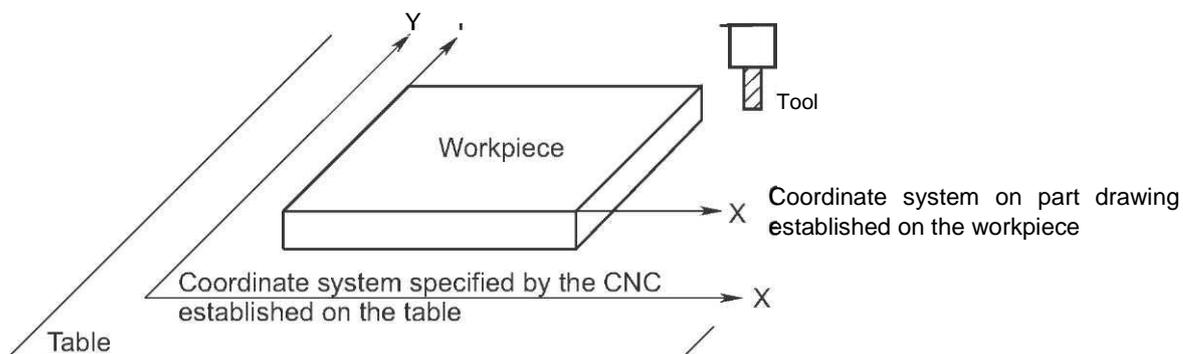
This is a coordinate system according to which has been designed the detail. The coordinate values from this drawing are used as input data when creating the program.

2) Coordinate system specified by CNC

The coordinate system should be prepared on the actual machine tool table. This could be done by programming the distance from the current position of the tool to the reference point of the desired coordinate system to be set.



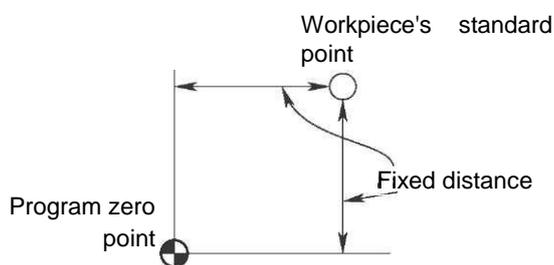
The relation between these two coordinate systems depends on the position of the workpiece on the table.



The tool moves in the coordinate system specified by the CNC control, following the commands of the program, which has been created according to the coordinate system on part drawing. That is the way of cutting a workpiece into a shape on the drawing. Having in mind all this, in order to correctly cut the workpiece as specified on the drawing, it is necessary the two coordinate systems be set at the same position.

The setting of the coordinate systems in the same position could be done using one of the following methods:

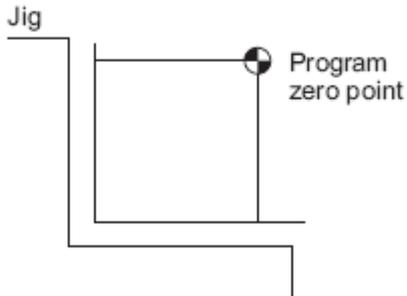
1) Using a standart plane and point of the workpiece



Bring the tool center to the workpiece standard point.
And set the coordinate system specified by the CNC at the position.

The center of the tool is used as standard point. The coordinate system of the CNC is set in that position.

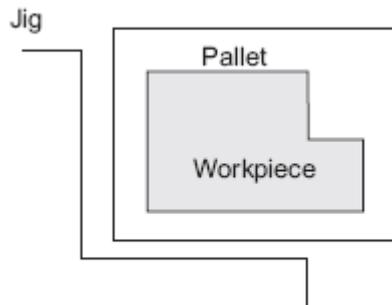
2) Mounting the workpiece directly against the jig



Meet the tool center to the reference position.
And set the coordinate system specified by CNC
at this position. (Jig shall be mounted on the
predetermined point from the reference position.)

Set the coordinate system specified by CNC in this position. Jig should be mounted in a point predetermined from the reference point.

3) Mounting the workpiece in the pallet and mounting the workpiece and the pallet on the jig



Jig and coordinate system shall be specified by the same as shown above

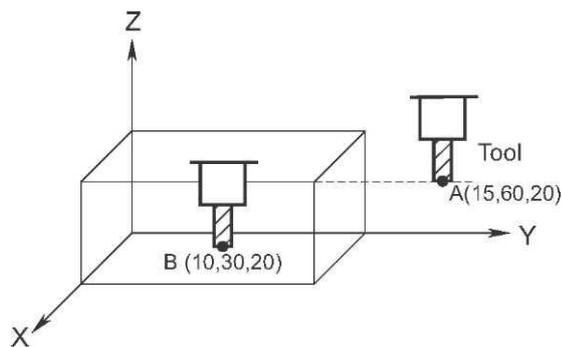
Jig position and coordinate system are specified as in (2).

1.3.3 Commands for Moving the Tool

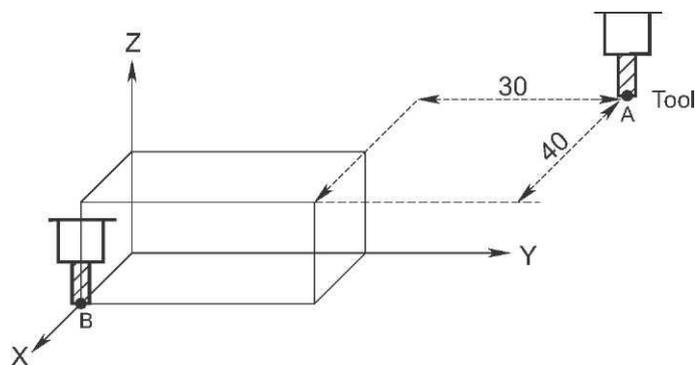
ABSOLUTE AND INCREMENTAL COMMANDS

Coordinate values in a command for moving the tool could be absolute or incremental.

- ◆ the tool moves to the point which is at a distance according to the origin of the coordinate system as specified in the coordinate values, i.e. coordinate values specify directly point in the coordinate system.



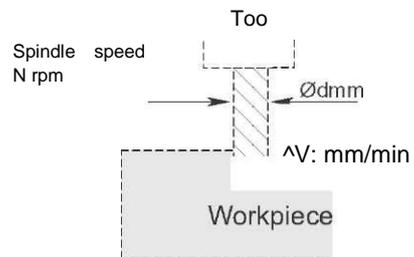
- ◆ specify the distance from the current position of the tool to the desired end position.



General

1.4 CUTTING SPEED - SPINDLE SPEED FUNCTION

The speed of the tool with respect to the workpiece during cutting is called cutting speed. **CNC** control specifies the cutting speed by determining the spindle speed in rpm.



Example:

The workpiece have to be machined with a tool 100 mm in diameter and cutting speed of 80 mm/min.

Spindle speed have to be approximately 255 rpm and could be obtained from:

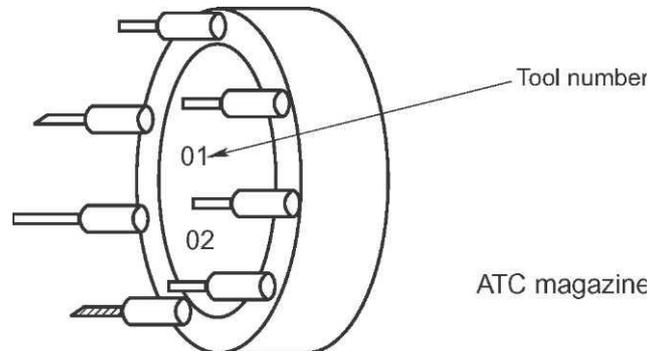
$$N = \frac{1000 \cdot V}{\pi \cdot D}$$

The command required is: S255;

This function specifies the spindle speed.

1.5 SELECTION OF TOOL

When drilling, tapping, boring, milling, etc. is performed, it is necessary to select a suitable tool. Each tool is assigned a number. When the program needs it, it just calls the needed one.



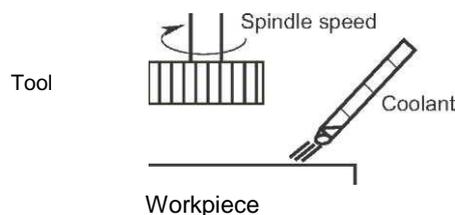
Example:

If the tool needed is in location 3 of the magazine, it will be selected by specifying the following command: T03;

This is a tool choosing command.

1.6 SPECIFIC MACHINE OPERATIONS - MISCELLANEOUS FUNCTION

When the machining itself starts, it is necessary to rotate the spindle, to feed coolant, etc. For this purpose, some kind of control is needed (frequently on/off operation), so that these processes could be controlled.



General

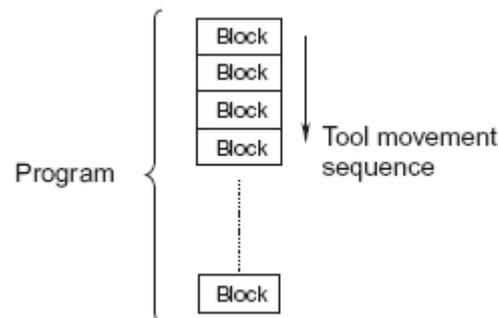
Controlling of these processes is realized by miscellaneous function. This function is specified by an **M** code. The actual process staying behind this function is requested by the machine builder in the machine control program. For more details on the number and functions of this code and its use refer to the manual provided by the machine builder.

Example:

Specifying **M03** causes rotating the spindle clockwise at a commanded speed.

1.7 PROGRAM CONFIGURATION

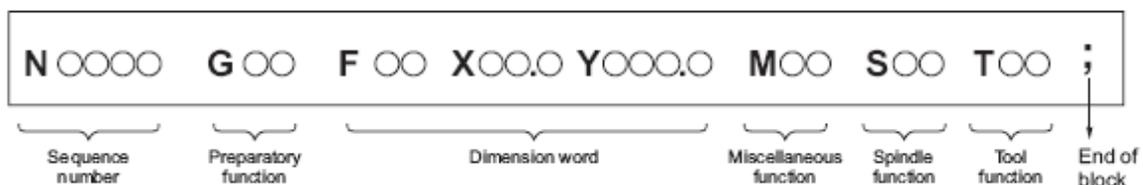
Program is a group of commands to the **CNC** unit for controlling the machine. A group of commands is called block. Therefore the program consists of blocks, which are executed one after another in a predetermined sequence. Each block specifies movement of the tool, a specific machine operation, etc.



Each block may have its own number. Each program has a unique number in the memory. The block and the program have the following configuration:

◆ Block

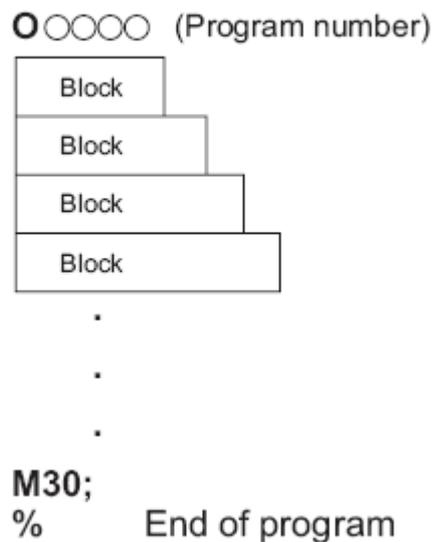
1 block



Any block has the following fields:

- *sequence number*
- *preparatory function*
- *feedrate specifying function*
- *coordinates of end point*
- *miscellaneous function*
- *spindle function*
- *tool function*
- *end of block*

◆ **Program**

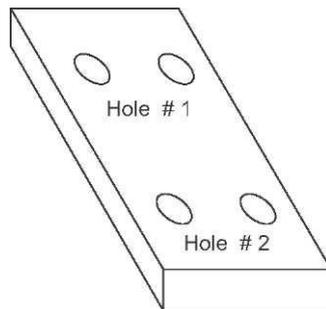
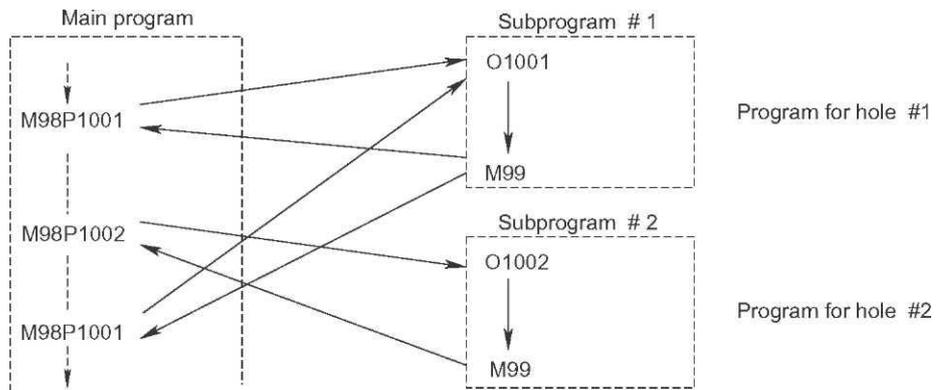


Any program starts with a block containing **○** code followed by the program number and an optional comment for the purpose of the program. The blocks specifying machine operations follow. Generally the programs end with a miscellaneous function, specifying the end of machining and the program.

◆ **Main program and subprograms**

General

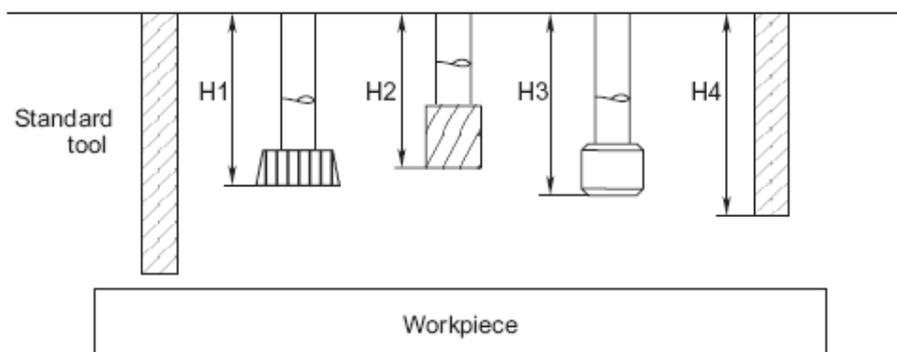
When certain kind of operations have to be repeated several times, in order to make small programs and write clear and understandable code, it is very convenient to create subprograms for executing repeatable operations. When such an operation has to be performed, the main program calls the subprogram which returns the control back to the main program after executing the operation.



1.8 TOOL FIGURE AND TOOL MOTION

◆ **Machining using the end of cutter - tool length compensation function**

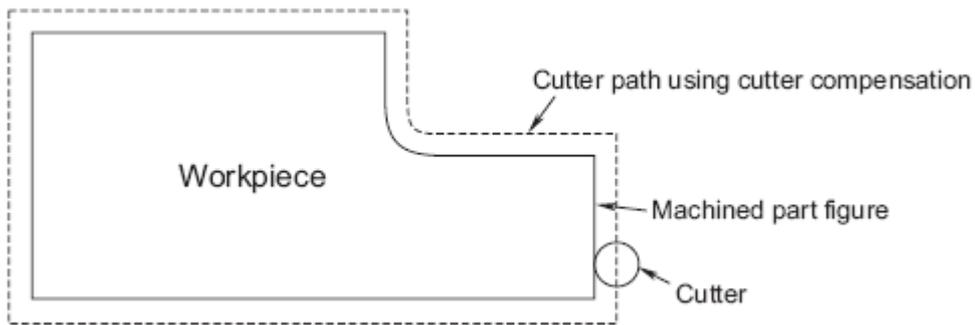
Usually, a set of tools is used when machining a workpiece and all of the tools have different length. Changing the program in accordance with the length of the tool is very time consuming process.



This problem could be eliminated if tool length compensation function is used. Each tool is measured to a reference point and the differences in length is stored in the memory of the CNC control unit. For each tool a compensation value is specified afterwards and the program should not be changed.

◆ **Machining using the side of the cutter - cutter compensation function**

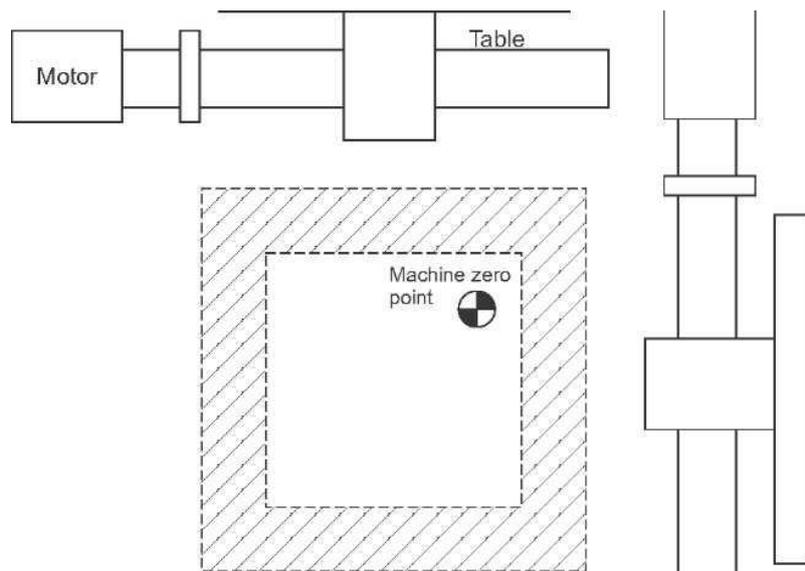
Because the cutter has a radius, the center of the tool moves along a trajectory different from the figure to be machined on the workpiece.



If the radius of each tool is stored in the memory of the CNC unit, the tool could be moved by the cutter compensation function along the trajectory needed in such a way that the distance to the workpiece will be the given radius. In this case as well, the program should not be corrected when changing the tool.

1.9 TOOL MOVEMENT RANGE - STROKE

Limit switches are installed at the ends of each axis on the machine to prevent tool moving beyond a predetermined region. They are used to keep the machine safe. Sensors are installed right before them, so that an alarm occurs. When choosing the place for their installation there have to be kept in mind the maximum speed of movement along an axis and the inertia.



These limits define a zone of work for the machine in which the tool moves. Beside that "hard" definition, the operator can define by parameters of the **CNC** control a working area. This could be done by specifying a distance to the reference point of the machine, which could be changed depending on the needs. For the limits of your machine refer to the manual supplied by the machine builder.

2. CONTROLLED AXES

2.1 CONTROLLED AXES

Number of controlled axes

SYSTEM 20 M - to **8**

Number of simultaneously controlled axes

SYSTEM 20 M - to **6**

Note:

When using the option for direct controlling of the spindle, the number of controlled axes decreases by one.

2.2 NAMES OF AXES

The basic axes are with names respectively **X**, **Y** and **Z**. Names of the additional axes can be selected by parameters.

2.3 INCREMENT SYSTEM

Name of increment system	Leastcommand increment	Maximum stroke
0 . 001 mm 0.0001 inch 0.001 deg	0 . 001 m m 0.0001 inch 0.001 deg	9999.999 mm 999.9999 inch 999.9999 deg

Combined use of inch metric system is not permitted. There are functions that cannot be used between axes with different unit systems (circular interpolation, cutter compensation, etc.). For more details on the increment system refer to the manual of the machine builder.

2.4 MAXIMUM STROKE

Maximum stroke = Least input increment x 9999999

3. PREPARATORY FUNCTIONS (G FUNCTIONS)

By specifying **G** code and numeric value after it, the operator sets the mode of the **CNC** unit, determines the way of interpreting the blocks in the programs, etc.

Generally G functions can be divided into the following two types:

One-shot G code - the function is valid only in the block in which is specified

Modal G code - the function is valid until another **G** code of the same group is specified

G codes are divided in groups and each group can have only one modal function.

Example: **G00** and **G01** are modal **G** codes in group 1

```
G01X__ ; i
      Y__ ; > G01 is valid only in this range
      X__ ;
G00Z__ ;
```

EXPLANATIONS:

1. After the power is on each group has one **G** code which is modal by default. In the following table these functions are marked with (*).

- **G00** or **G01** is specified by parameter
- **G20** or **G21** retains its last state, which can be checked in the field **INCH** on the screen **SETTINGS**

2. All **G** codes in group **00** are one-shot.

3. If a **G** code that does not appear in the table or which is not optional and is not supported, an alarm is displayed.

4. There could be used multiple **G** codes from different groups in a single block. When multiple **G** codes from one group is specified, valid is the **G** code specified last.

5. If a **G** code from group **01** is specified in a canned cycle, the cycle is automatically cancelled and **G80** becomes modal by default. On the other side, **G** code from group **01** is not affected by **G** codes for canned cycles.

6. The screen can show all current modal **G** codes for each block.

G Code List

G code	Group	Function
G00*	01	Positioning in rapid traverse
G01*		Linear interpolation
G02		Circular interpolation CW
G03		Circular interpolation CCW
G04*	00	Dwell. Exact stop
G10	02	Data setting
G17		X Y plane selection
G18		Z X plane selection
G19		Y Z plane selection
G20	06	Input in inch
G21*		Input in mm
G27	00	Reference position check
G28		Reference position return
G29		Return from reference position
G30		Second reference position return
G31		SKIP function
G33		01
G39	00	Angular circular interpolation
G40*	07	Cutter compensation cancel
G41		Cutter compensation on left
G42		Cutter compensation on right
G43	08	Tool length compensation + direction
G44		Tool length compensation - direction
G49*		Tool length compensation cancel
G54*	14	Workpiece coordinate system 1 selection
G55		Workpiece coordinate system 2 selection
G56		Workpiece coordinate system 3 selection
G57		Workpiece coordinate system 4 selection
G58		Workpiece coordinate system 5 selection
G59		Workpiece coordinate system 6 selection
G60	00	Single direction positioning

Continue

G code	Group	Function
G65	00	Macro call
G66	11	Macro modal call
G67*		Macro modal call cancel
G73	09	Peck drilling cycle
G74		Counter tapping cycle
G76		Fine boring cycle
G80*		Canned cycle cancel
G81		Drilling cycle/spot boring cycle
G82		Drilling cycle/counter boring cycle
G83		Peck drilling cycle
G84		Tapping cycle
G85		Boring cycle
G86		Boring cycle
G87		Back boring cycle
G88		Boring cycle
G89		Boring cycle
G90*	03	Absolute command
G91		Increment command
G92	00	Setting of absolute reference position
G94*	05	Fit per minute
G95		Fit per rotation
G98*	10	Return to initial point in canned cycle
G99		Return to point R in canned cycle

Interpolation Functions

4. INTERPOLATION FUNCTIONS

4.1 POSITIONING (G00)

The **G00** command moves the tool to a defined absolute or incremental position at a rapid traverse rate.

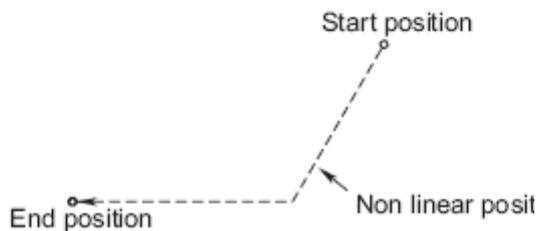
In absolute command, the operator specifies the coordinates of the end point. In incremental command, the operator specifies the distance to the point desired.

Format:

G00 IP_;

IP_ : position in absolute command or distance in incremental Tool path

generally is not a straight line.



The rapid traverse rate in command **G00** is set with parameters for each axis independently by the machine builder. In this positioning mode the tool is accelerated in the beginning of the block, moved along the least path and decelerated at the end of the block. The execution of the program continues after confirming "**in-position**" check.

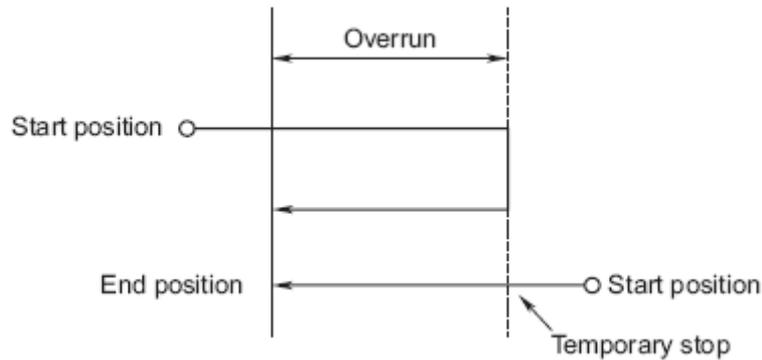
"**In-position**" means that deviation in servo system is in the predetermined range. The range is set with corresponding parameters by the machine builder and generally there is no need of changing it.

RESTRICTIONS:

The rapid traverse rate cannot be specified by **F** code.

4.2 SINGLE DIRECTION POSITIONING (G60)

For accurate positioning without backlash, positioning from one direction is available.

**Format:****G60 IP_n;****IP_n :**

In absolute command these are the coordinates of the end point, in incremental command that is the direction of movement of the tool.

The overrun and positioning direction are set by the corresponding parameters. Even when the command for positioning direction coincides with that set for the parameter, the tool stops once before the end point.

RESTRICTIONS:

- During drilling canned cycle positioning along **Z** axis does not take effect.
- No single direction positioning is done for an axis for which no overrun has been set by the parameter.
- When the command for move distance is **0**, no single direction positioning is performed.
- The direction set by the parameter is not affected by mirror image.
- Single direction positioning should not be performed with commands **G76** and **G87** for shifting in canned cycles.

4.3 LINEAR INTERPOLATION

The tool moves along a straight line.

Format:

G01 IP_F_;

In absolute command these are the coordinates of the end point, in incremental command that is the direction of movement of the tool.

F_ : Speed of tool speed

The tool moves along a straight line to the predefined position at the feedrate specified in **F**. The speed set in **F** is effective until a new value is specified. It is not necessary to set it in each block. The feedrate set in **F** is measured along the direction of movement of the tool. If **F** is not specified, the feedrate is regarded as zero. The feedrate along each axis is as follows:

G01 α β γ ζ Ff;

Feedrate along axis:

$$\alpha: F\alpha = \frac{\alpha}{L} \cdot f$$

$$\beta: F\beta = \frac{\beta}{L} \cdot f$$

$$\gamma: F\gamma = \frac{\gamma}{L} \cdot f$$

$$\zeta: F\zeta = \frac{\zeta}{L} \cdot f$$

$$L = \sqrt{\alpha^2 + \beta^2 + \gamma^2 + \zeta^2}$$

Interpolation Functions

The feedrate of the rotary axes is specified in the unit degrees/minute (the value is with decimal point).

When the axis **a** for linear movement (as **X**, **Y** or **Z**) and axis **b** for rotational movement (as **A**, **B** or **C**) are linear interpolated, the feedrate is that in which the tangential feedrate in the **a** and **b** Cartesian coordinate system is commanded by **F** (mm/min).

The feedrate along axis **b** is obtained when first calculate the necessary time for movement according the latter formula and then the feedrate along axis **b** is converted in degrees per minute.

Example:

G91 G01 X20.0 B40.0 F300.0

This changes the unit of axis B from 40.0 degrees in 40 mm with metric input. The time required for distribution is calculated as follows:

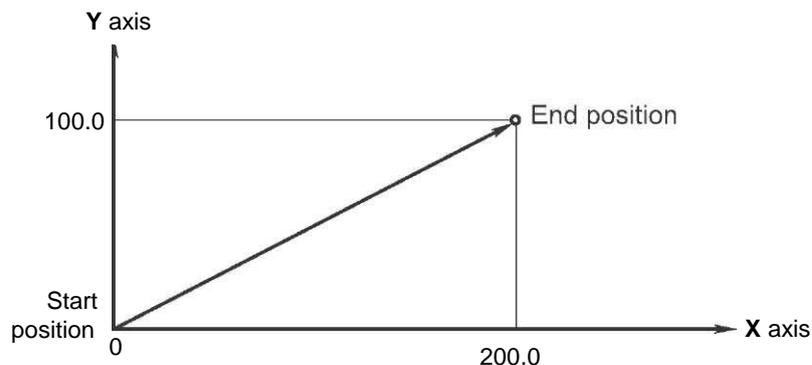
$$\frac{\sqrt{20^2 + 40^2}}{300} = 0.14907(\text{min})$$

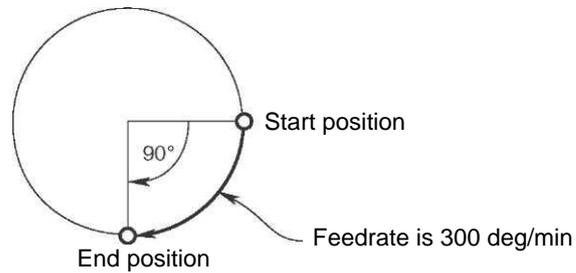
The feedrate for axis B is as follows:

$$\frac{40}{0.14907} = 268.3\text{deg/min}$$

Examples:

Linear interpolation



Feedrate for rotation axis**4.4 CIRCULAR INTERPOLATION (G02, G03)**

The command specified moves the tool along a circular arc.

Format:

The arc is in **XY** plane:

$$G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Xp_ Yp_ \begin{Bmatrix} I_ J_ \\ R_ \end{Bmatrix} F_;$$

The arc is in **ZX** plane:

$$G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Xp_ Zp_ \begin{Bmatrix} I_ K_ \\ R_ \end{Bmatrix} F_;$$

The arc is in **YZ** plane:

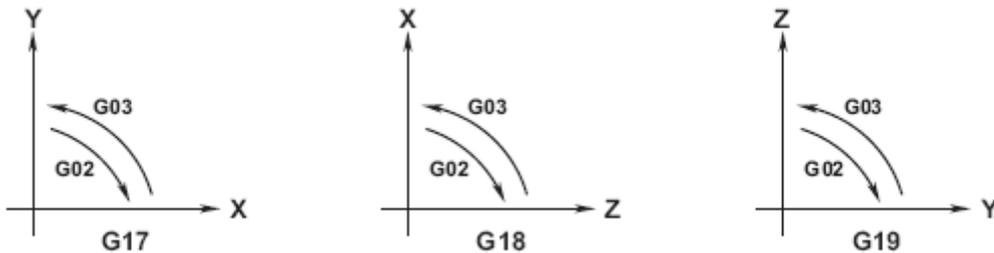
$$G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Yp_ Zp_ \begin{Bmatrix} I_ K_ \\ R_ \end{Bmatrix} F_;$$

Description of the Command and Format

Command	Description
G17	Specification of arc in X Y plane
G18	Specification of arc in Z X plane
G19	Specification of arc in Y Z plane
G02	Circular interpolation CW
G03	Circular interpolation CCW
X_	Command values X axis
Y_	Command values Y axis
Z_	Command values Z axis
I_	X axis distance from the start point to the center of the arc with sign
J_	Y axis distance from the start point to the center of the arc with sign
K_	Z axis distance from the start point to the center of the arc with sign
R_	Radius of the arc, depending or the designation
F_	Feedrate along an arc

- Direction of the circular interpolation

The terms "**CW**" (**G02**) and "**CCW**" (**G03**) in the pane **XY** (**ZX** or **YZ**) are defined when viewing the plane **XY** over the **Z** axis (**Y** or **X** respectively) in the positive-to-negative direction in the Cartesian coordinate system. See the figure below.



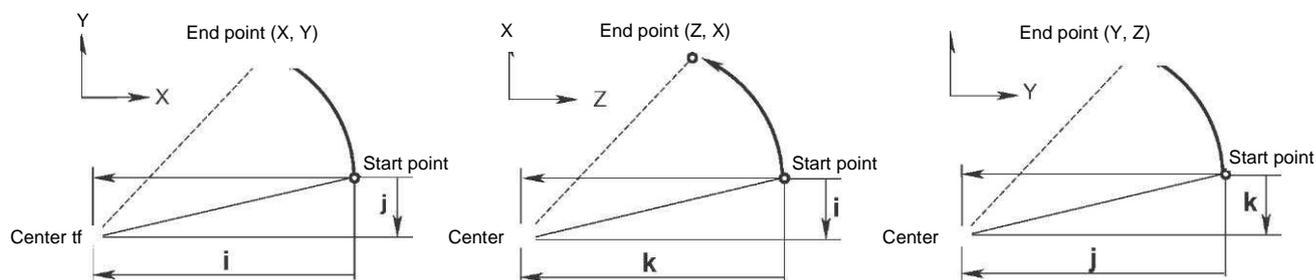
Clock wise and counter clock wise

- Distance moved on an arc

The end point of the arc is specified with **X**, **Y** and **Z** codes and is expressed with absolute or incremental value depending on G90 or G91. Incremental value specifies the distance between start and end point of the arc.

- Distance from the start point to the center of the arc

The center of the arc is specified with **I**, **J** and **K** codes, for axes **X**, **Y** and **Z**, respectively. The number following **I**, **J** or **K**, however, is a vector component, in which the center of the arc is seen from the start point and is determined always as an incremental value irrespective to **G90** or **G91**, as shown below. **I**, **J** and **K** must be with sign depending on the direction.



Circular interpolation programming

I0, **J0** and **K0** can be omitted. When **X**, **Y** and **Z** are omitted (the end point coincides the start one) and the center is specified with **I**, **J** and **K**, then an arc with 360 degrees is specified (circle).

Example:

G02I_; Circle command.

- Arc radius

The distance between the arc and the center of a circle which contains the arc can be specified directly using the **R** of the circle instead **I**, **J** or **K**. In this case it is determined that one arc is less than 180 degrees and the other is more than 180 degrees. When an arc with more than 180 degrees is commanded, the radius should be negative. When **X**, **Y** and **Z** are omitted and if the start point coincides the end one and **R** is used, then an arc with 0 degrees is commanded.

Interpolation Functions

Example:

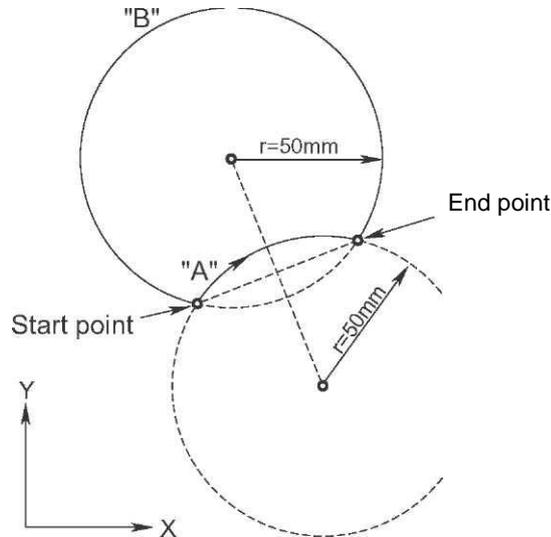
G02R_; The cutter is not moving.

For arc "A" less than 180°:

G91 G02 X60.0 Y20.0 R50.0 F300.0;

For arc "B" greater than 180°:

G91 G02 X60.0 Y20.0 R-50.0 F300.0;



◆ Feedrate

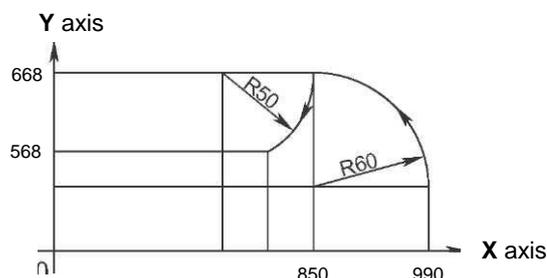
The feedrate in circular interpolation is the same as the feedrate specified in **F** code. The value to be controlled and specified is the feedrate along the arc (tangential feedrate). The error between feedrate specified and the actual tool feedrate is $\pm 2\%$ or less. However, the feedrate is measured along the arc after cutter compensation is applied.

RESTRICTION:

If in the code exist simultaneously **I**, **J**, **K** and **R**, the arc specified in **R** has a priority and the others are ignored. If an axis which is not in the specified plane is commanded, an alarm is displayed.

When arc with a center angle close to 180 degrees is commanded with **R**, it is possible that the system could not calculate correctly the center of the arc. Therefore specify arcs with **I**, **J** and **K**.

Examples:



The tool path as shown on the figure can be programmed as follows:

(1) ***Absolute programming***

```
G92X990.0 Y568.0 Z0;
G90 G02 X850.0 Y668.0 R60.0 F300.;
```

or

```
G92X990.0 Y568.0 Z0;
G90 G02 X850.0 Y668.0 I-60.0 F300.;
```

(2) ***Incremental programming***

```
G91 G03 X-60.0 Y-60.0 R60.0 F300;
G02 X-20.0 Y40.0 R50.0;
```

or

```
G91 G03 X-60.0 Y-60.0 I-60.0 F300; G02
X-20.0 Y40.0 I-50.0;
```

4.5 SKIP FUNCTION

Linear interpolation can be commanded as an axial move following the **G31** command, like **G01**. If an external signal cancelling the operation occurs during the execution of that command, the command is interrupted and the work resumes with the next block. Skip function is used when the end of machining is not specified but is determined by signal from the machine itself, as for example in grinding. Other application is for measuring the dimensions of the workpiece.

Format:

```
G31 IP_;
```

G31 : One-shot **G** code (effective only in the block in which is specified)

Coordinate values when function skip is commanded can be used in user macro, as they are kept in the user macro system - variables from #5061. The addresses are as follows:

```
#5061 : X axis coordinate Y
#5062 : axis coordinate Z
#5063 : axis coordinate
```

Interpolation Functions

WARNING:

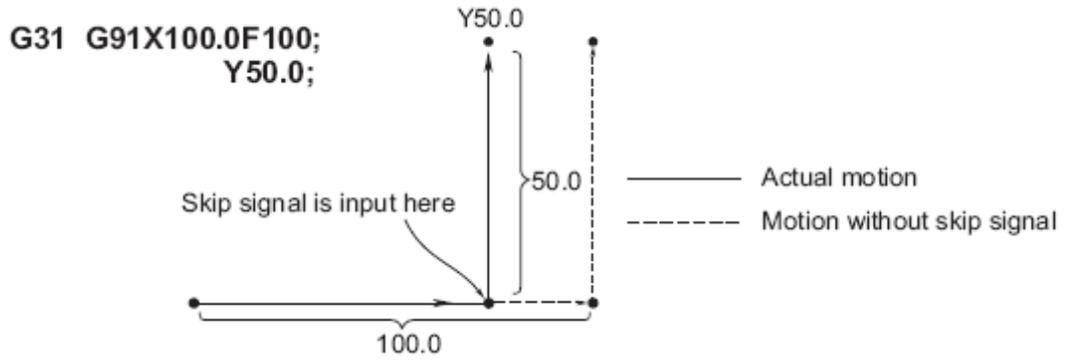
Disable feedrate override, dry run and automatic acceleration and deceleration (These operations could be performed by setting parameter **SKPF** to 1) when feed per minute is specified, so that you can receive the error in tool position when skip function is commanded. The above functions are enabled when specifying feed per rotation.

Note:

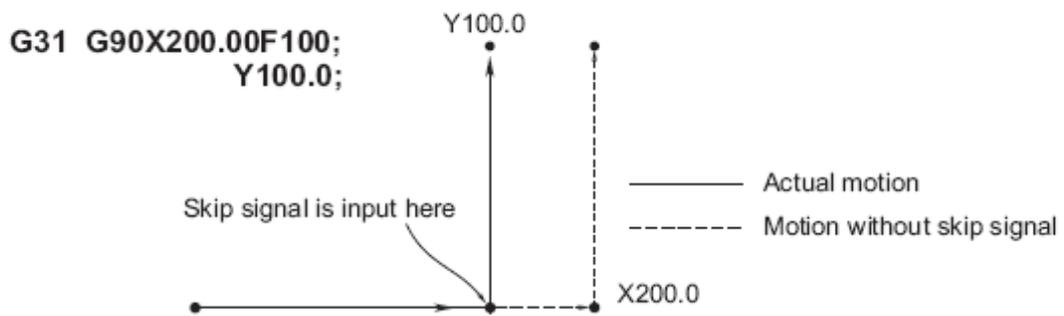
*If **G31** is commanded in cutter compensation, an alarm **P/S036** is displayed. Before using **G31** command cancel cutter compensation with **G40**.*

Examples:

- **The next block to G31 is an incremental command**

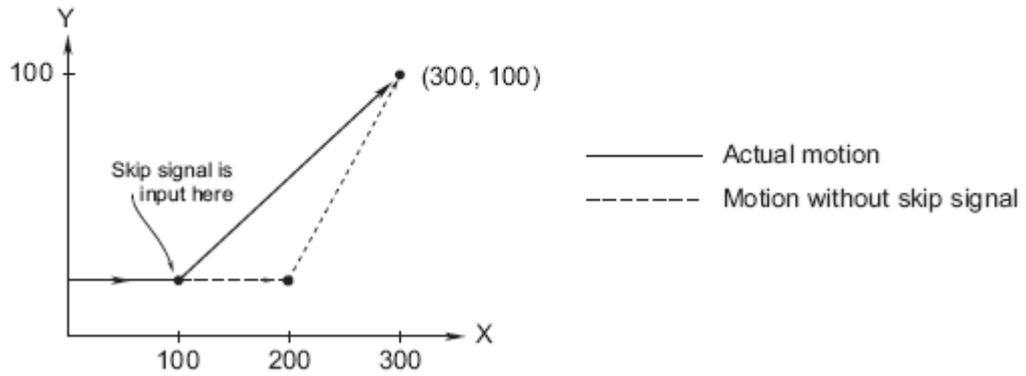


◆ The next block to G31 is an absolute command for one axis



- ◆ The next block to G31 is an absolute command for two axes

G31 G90X200.0F100;
X300.0 Y100.0;



5. FEED FUNCTIONS

5.1 GENERAL

Feed functions control the feedrate of the tool. The following two feed functions are used:

◆ Feed functions

1. *Rapid traverse*

When a positioning command is specified (**G00**), the tool moves at rapid traverse, set in the **CNC** by parameters.

2. *Cutting feed*

The tool moves with predetermined cutting feed.

◆ Override

Override can be applied to a rapid traverse rate or cutting feed using switches on the machine operator's panel.

◆ Automatic acceleration/deceleration

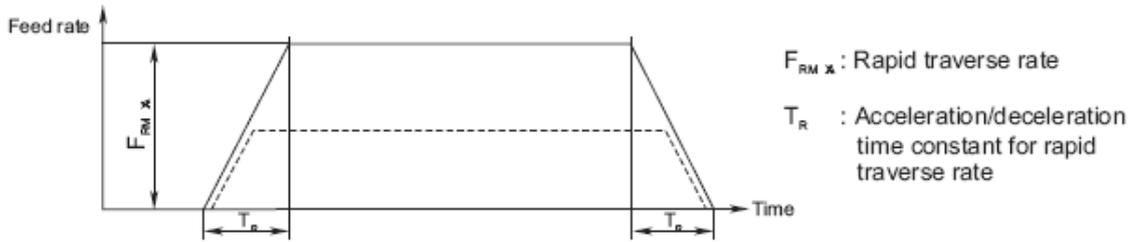
The automatic acceleration/deceleration is used at the beginning and at the end of tool movement in order to avoid mechanical shock.

Rapid traverse rate

Linear acceleration/deceleration (constant acceleration)

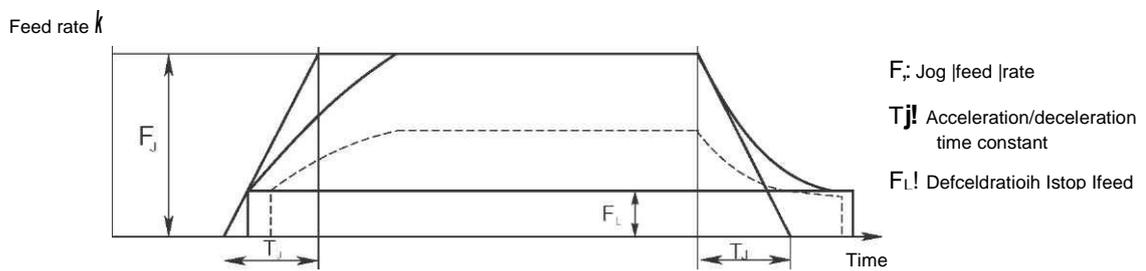
Feed rate

Linear acceleration/deceleration (constant acceleration)



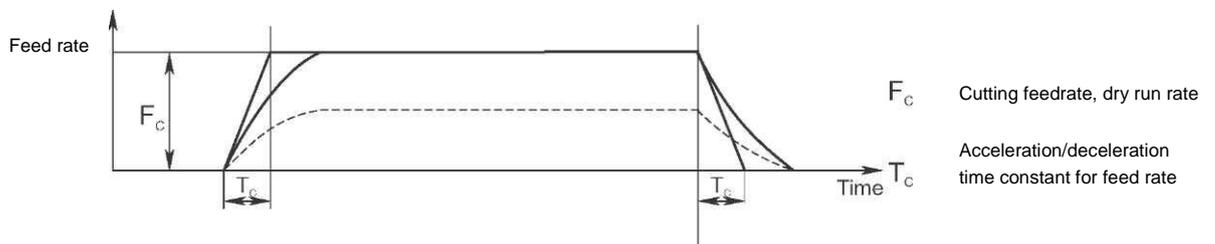
Jog feed

Exponential acceleration/deceleration (constant time)



Cutting feed, dry run Exponential

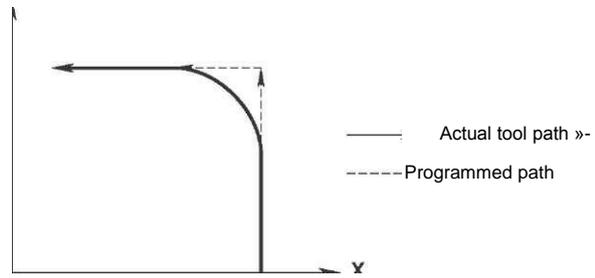
acceleration/deceleration (constant time)



◆ **Tool path in a cutting feed**

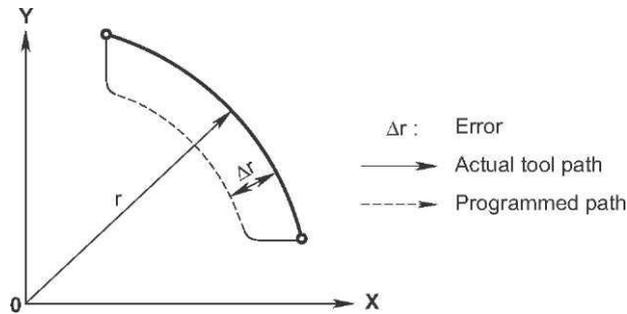
If the tool changes its direction when the program runs from one block to another during machining, a rounded corner path may result.

Here is an example of tool path between two blocks of the program.
0



Feed Functions

In circular interpolation a radial error occurs.



The rounded corner path shown on the first figure and the error shown on the second one depend on the feed rate. Therefore that feedrate must be controlled, so that the tool to move as being programmed.

5.2 RAPID TRAVERSE

Format:

G00 IP_z;

G00 : G code (group 01) for positioning (rapid traverse) **IP_z** ; Dimension word specifying the end point

The positioning command (G00) positions the tool at a rapid traverse. In rapid traverse the next block is executed after feedrate specified becomes 0 and the servo motor reaches the predetermined position in a certain range set by parameters.

Position check. (The position check can be disabled for each block by corresponding parameter.) Rapid traverse rate is specified for each axis by parameters. The following overrides can be set using the switch on the machine operator's panel: F0, 25, 50, 100%.

F0: Allows setting a fixed feedrate of each axis by parameter. For detailed information refer to the appropriate manual supplied by the machine builder.

5.3 CUTTING FEED

Feedrate in linear interpolation (**G01**), circular interpolation (**G02, G03**), etc. is specified by a number after **F** code. In cutting feed the block is executed in such a way that the feedrate change compared to the previous block is minimized.

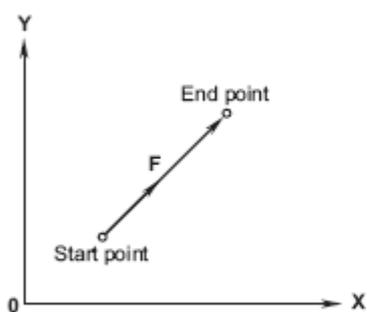
Format:***Feed per minute***

G94; **G** code (group 05) for feed per minute

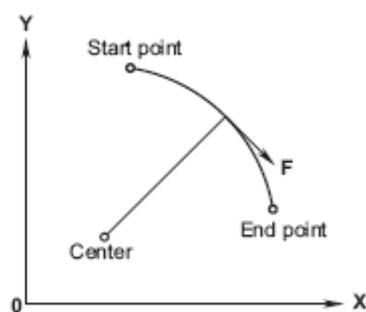
F_; Feedrate command (mm/min or inch/min)

◆ **Constant control of tangential speed**

Cutting feed is constantly monitored so that the tangential feedrate is always a predetermined value.



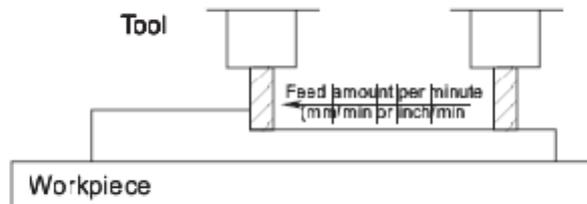
Linear interpolation



Circular interpolation

Feed per minute

After specifying **G94** (in the feed per minute mode), the amount of feed of the tool per minute is to be directly determined by writing a number after **F** code. **G94** is a modal code. After the power is on a value for feed per minute is specified. An override of 0 to 150% can be applied to the set value with a switch on the machine operator's panel. For more details refer to the manual supplied by the machine builder.



WARNING:

No override can be applied for some of the commands such as threading.

◆ **Cutting feedrate clamp**

A common limit for the cutting feedrate can be set by parameter. If the actual feedrate (including override) exceeds the upper limit specified, then the feedrate is clamped to that value.

Note:

The upper limit is set in mm/min or inch/min. The calculation of the CNC can lead to an error of $\pm 2\%$ with respect to the specified value. However, this is not true for the acceleration and deceleration. To be more detailed, this error appears because of the measure of time the tool takes to move 500 mm or more after steady state.

5.4 CUTTING FEEDRATE CONTROL

The feedrate can be controlled as shown in the table below.

Functionname	Control	Validity	Description
Exact stop	G04	Valid only for the specified block	The tool is decelerated at the end of the block and then an in-position check is made. The next block is executed afterwards
Exact stop	Operator's panel		The tool is decelerated at the end of the block and then an in-position check is made. The next block is executed afterwards.
Cutting mode	Operator's panel		The tool is not decelerated at the end of the block but the next block is executed.
Tapping mode	Operator's panel		The tool is not decelerated at the end of the block but the next block is executed. When is specified feedrate override and feed hold are invalid.

Notes:

1. *The aim of in-position check is confirming the range of the servo motor (specified by the corresponding parameter).
The in-position check can be disabled by the corresponding parameter.*
2. *Cutting feed is set by default. Exact stop or tapping mode setting is performed according to the controller program by the machine builder. Generally, this can be done by buttons. For details refer to the manual supplied by the machine builder.*

Format:

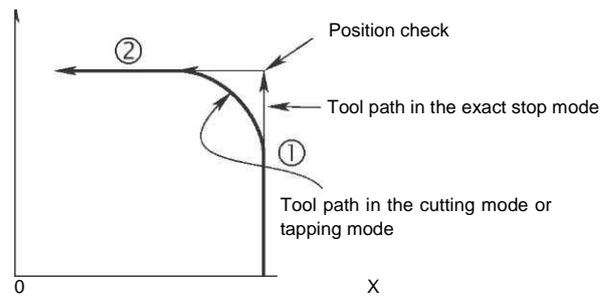
Exact stop **G04;**

Feed Functions

5.4.1 Exact Stop (G04)

Cutting Mode
Tapping Mode

The paths tool follow in the block itself in exact stop and tapping mode are different. Here is an example of a tool path from block 1 to block 2.



WARNING:

After switching the power on and after reset the default is cutting mode.

5.5 DWELL (G04)

Format:

Dwell G04 X_;; or **G04 P_;**;

X_; : specifies time (decimal point allowed)

P_; : specifies time (decimal point not allowed)

When dwell is commanded, the execution of the next block is delayed specified time. In addition to that, dwell could be commanded to be made check for the mode. When the command is without parameters **X_;** or **P_;** an exact stop is performed.

6. REFERENCE POSITION

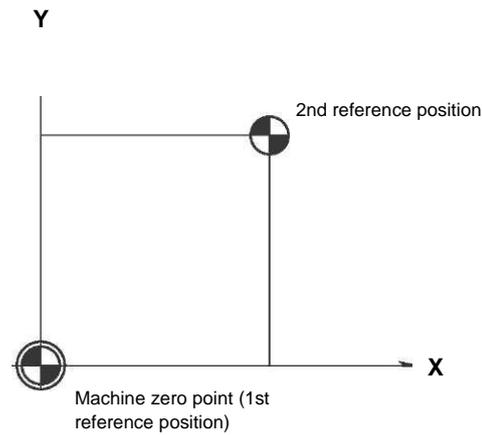
GENERAL:

◆ Reference position

The reference position is a fixed position of the machine tool to which the tool is easily moved using the function for reference position return.

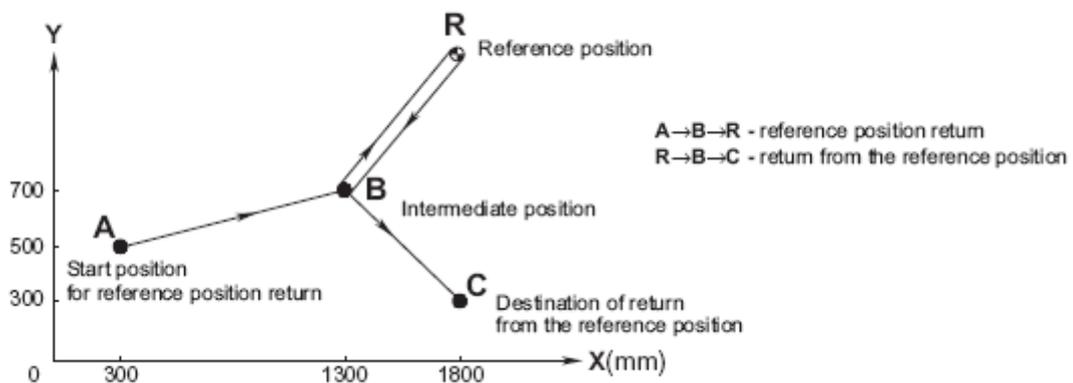
For example, the reference point is used as a position at which the tools are automatically changed. Up to two reference positions with their corresponding coordinates in the machine coordinate system can be set by parameters. The first reference position should be the machine zero point.

Reference Position



◆ Reference position return and movement from the reference position

The tools automatically return in the reference position via an intermediate position along a specified axis. The tools are automatically moved from the reference position to a specified position via an intermediate position along an axis. When the reference position return operation is completed, it is indicated by a lamp on the operator's panel. (The indication varies with different manufacturers.)



◆ **Reference position return check** Reference position return check is a function that checks whether the tool is correctly positioned in the reference point as specified by the program. If the tool has correctly returned to the reference position along the axis specified, the lamp for indicating the completion of return goes on.

Format:

◆ **Reference position return**

G28 IP_; Reference position return

G30 IP_; Second reference position return

IP_ : A command specifying intermediate position (Absolute/incremental command)

◆ **Return from reference position**

G29 IP_;

IP_ : A command specifying the destination to go from the reference position. (Absolute/incremental command)

◆ **Reference position return check**

G27 IP_;

IP_ : A command specifying the reference position (Absolute/incremental command)

EXPLANATIONS:

◆ **Reference position return**

Positioning in intermediate or reference position is performed at a rapid traverse along each axis.

For that reason, for safety, the cutter compensation must be disabled before using that command. The coordinates of the intermediate point are stored in the **CNC** unit only for the axes for which **G28** command is specified in the block. The other axes use predetermined coordinates.

Reference Position

Example:

```
N1 G28 X40.0;   Intermediate position (X40.0) Intermediate
N2 G28 Y60.0;   position (X40.0, Y60.0)
```

▪ **Second reference position return**

In a system without absolute-position detector, functions for second reference position return can be used only after execution of command for return to first reference position (**G28**) or manual reference position return. The **G30** command is generally used when the automatic tool change position is different from the reference position.

▪ **Return from the reference position**

Generally, this command follows immediately **G28** or **G30** commands. In incremental mode the value of the command specifies the incremental value from the intermediate position.

Positioning in the intermediate or reference points is performed at rapid traverse rate for each of the axis.

When the workpiece coordinate system is changed with the command **G28** after the tool reaches the reference point through intermediate point, the intermediate point is also shifted in the new coordinate system. If **G29** is commanded afterwards, the tool moves to the specified position through the intermediate point which has been shifted in the new coordinate system.

The same operations are performed after **G30** is commanded.

▪ **Reference position return check (G27)**

The **G27** command positions the tool at rapid traverse rate. If the tool reaches the reference point, a light indicates the completion of the operation.

If, however, the position reached by the tool is not the reference position, an alarm is displayed.

RESTRICTIONS:**- Status when the machine lock is turned on**

The lamp for indicating the completion of the positioning in the reference position does not go on when the machine lock is turned on even when the tool automatically returns to the reference position. In that case, it is not checked whether the tool has returned to the reference point even when **G27** is commanded.

- First return to the reference position after the power is switched on

When **G28** is commanded and when no manual reference position return is performed, after the power is on, the movement from the intermediate point is the same as manual return to the reference position.

In that case the tool moves in the direction specified for the reference position in the corresponding parameter. For that reason the intermediate point specified should be a position from which return to the reference point is possible.

- Reference position return check in offset mode

In an offset mode, the position that should be reached by the tool when **G27** is commanded is the position obtained by adding the offset value. For that reason if the position value added to the shift value is not the reference position, the lamp does not go on but an alarm is displayed. Therefore it is better to disable shifts before **G27** is commanded.

- Lighting the lamp when the programmed position does not coincide the reference position

When the machine tool system is as an inch one with metric input, the lamp indicating reference position return can go on even when the programmed position is shifted from the actual reference position with 1 micron. This occurs because the least input increment of the machine is smaller than the least command increment.

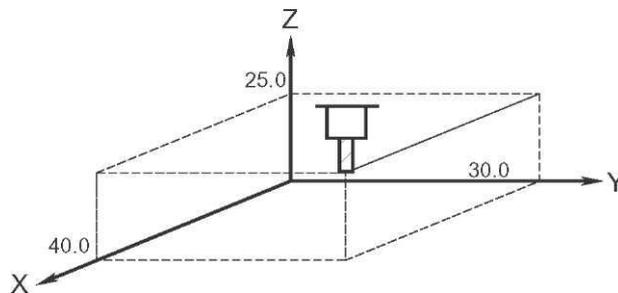
7. COORDINATE SYSTEM

The tool can be moved to a specified position after teaching **CNC** to do such an operation. Such a positioning of the tool is represented in coordinates in a specified coordinate system. The coordinates are set by program axes.

When three coordinate axes are used (**X**, **Y** and **Z**), the coordinates are set in the following way:

X_Y_Z_

This command is referred as a word specifying dimension. *Example:*
Tool positioning commanded with **X40.0 Y30.0 Z25.0**.



The coordinates are set in one of the following coordinate systems:

- 1) **Machine coordinate system**
- 2) **Workpiece coordinate system**

The number of axes of a coordinate system varies with different machines. Therefore in this manual the dimension word is represented with **IP_**.

7.1 MACHINE COORDINATE SYSTEM

*The point that is specific to the machine and acts as a reference point is called **machine zero point**.* The machine zero point coincides the first reference point.

*The coordinate system with origin the machine zero point is called **machine coordinate system**.*

The machine coordinate system is set by manual return to the reference position after the power is on. Once set, the machine coordinate system stays unchanged till the power is switched off.

7.2 WORKPIECE COORDINATE SYSTEM

*The coordinate system used when machining the workpiece is called **workpiece coordinate system**. A proper coordinate system must be set before the actual machining begins (workpiece coordinate system).*

The once set workpiece coordinate system can be changed by shifting the center (workpiece coordinate system change).

7.2.1 Setting a Workpiece Coordinate System

The workpiece coordinate system can be set in one of the following three methods:

1) By commanding G92

The workpiece coordinate system is set by specifying a value after G92 code in the program.

2) Automatic setting

The workpiece coordinate system is automatically set after performing a manual reference position return if the corresponding parameter has been properly set before.

3) Input using the CRT/MDI panel

Six workpiece coordinate systems can be set using the **CRT/MDI** panel.

Set the workpiece coordinate system by one of the methods described above for that you can use absolute programming.

Format:

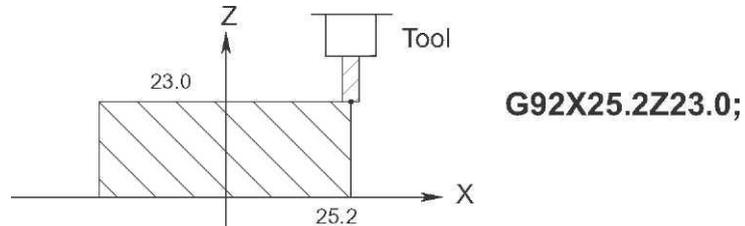
◆ Workpiece coordinate system setting by G92 command G92 IP_;

The workpiece coordinate system is set in such a way that the leading point of the tool, as for example its tip, is at the coordinates specified. If the coordinate system is set using **G92** in shift mode for compensation the length of the tool, such a coordinate system is set that the position before shifting coincides the position specified by **G92** command.

Coordinate system

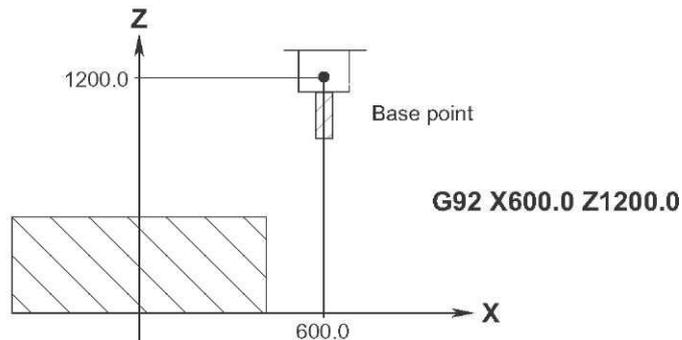
When **G92** is commanded cutter compensation is temporarily canceled

Example 1:



Setting the coordinate system with the command **G92 X25.2 Z23.** (The tool tip is the start point for the program)

Example 2:



Setting the coordinate system with the command **G92 X600.0 Z1200.0.** (The base point of the tool holder is the start point for the program)

If an absolute command is issued, the base point is moved to the position specified. In order to move the tool tip to the specified position, there must be set a tool length compensation adjusting the difference.

7.2.2 Selecting a Workpiece Coordinate System

The user can choose one of the predetermined coordinate systems described below. (For more details on setting methods see the previous chapter.)

1) Workpiece coordinate system selecting by **G92** command or automatic workpiece coordinate system setting.

Once the workpiece coordinate system has been set, the absolute commands may use it.

2) Choosing from six workpiece coordinate systems set by the TFT/MDI panel. By specifying **G** code from **G54** to **G59** there can be chosen one of the following six coordinate systems:

G54 . . . Workpiece coordinate system number 1

G55 . . . Workpiece coordinate system number 2

G56 . . . Workpiece coordinate system number 3

G57 . . . Workpiece coordinate system number 4

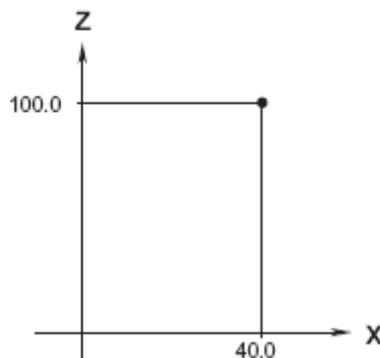
G58 . . . Workpiece coordinate system number 5

G59 . . . Workpiece coordinate system number 6

The coordinate systems from 1 to 6 are set after reference position return after the power is switched on. When the power is on the default coordinate system is **G54**.

Examples:

G90 G55 G00 X40.0 Y100.0; Z



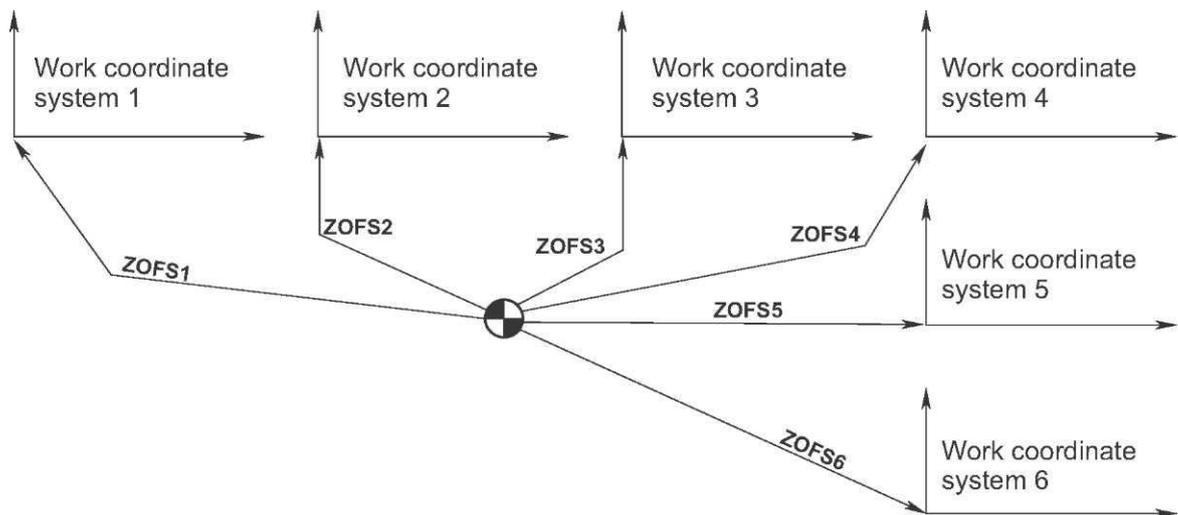
In this example the positioning is in coordinates **(X40.0 Y100.0)** in workpiece coordinate system number 2.

7.2.3 Workpiece Coordinate System Change

The six workpiece coordinate systems specified by **G54** to **G59** can be changed by changing the offset value of the external workpiece zero point or the offset value of the workpiece zero point.

There can be used three methods for changing the external offset value of the workpiece zero point.

- 1) TFT/MDI *panel input*
- 2) *Programming by G10 or G92*
- 3) *Change of the offset value of the external workpiece zero point* (for more details refer to the machine builder programming manual).



EXOFS : External offset value of the workpiece zero point
ZOFS1 to ZOFS6 : Offset value of the workpiece zero point

Format:

- Changing by G10

G10 L2 Pp IP_;

- p = 0 :** External offset value of the workpiece zero point
- p = 1 to 6 :** The offset value of the workpiece zero point correspond to the workpiece coordinate system from 1 to 6
- IP_; :** Offset value of the workpiece zero point for each of the axis

◆ **Changing by G92**

G92 IP_;

EXPLANATIONS:

◆ **Changing by G10**

Each workpiece coordinate system can be changed independently by commanding **G10**.

When an absolute offset value of the workpiece zero point has been specified, the value specified is the new offset value.

When an incremental offset value of the workpiece zero point has been specified, the value specified is added to the current offset value thus giving the new offset value.

◆ **Changing by G92**

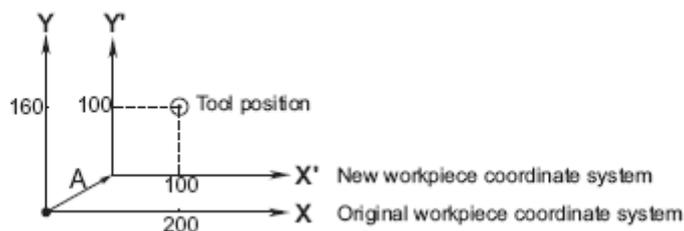
By specifying the command **G92 IP_;** the workpiece coordinate system (chosen with some of the commands from **G54** to **G59**) is shifted thus giving a new coordinate system in which the current tool position coincides the coordinates specified in **IP_**. Then the amount of coordinate system shift is added to all shift values of workpiece zero points. This means that all workpiece coordinate systems are shifted by the same amount.

WARNING:

When the coordinate system is set by **G92** after setting the shift value of the workpiece zero point, the coordinate system is not affected by the external zero point shift value. When for example **G92 X100.0 Z80.0;** is commanded, this means that a coordinate system with momentary values **X = 100.0** and **Z = 80.0** of the tool reference point is set.

Examples:

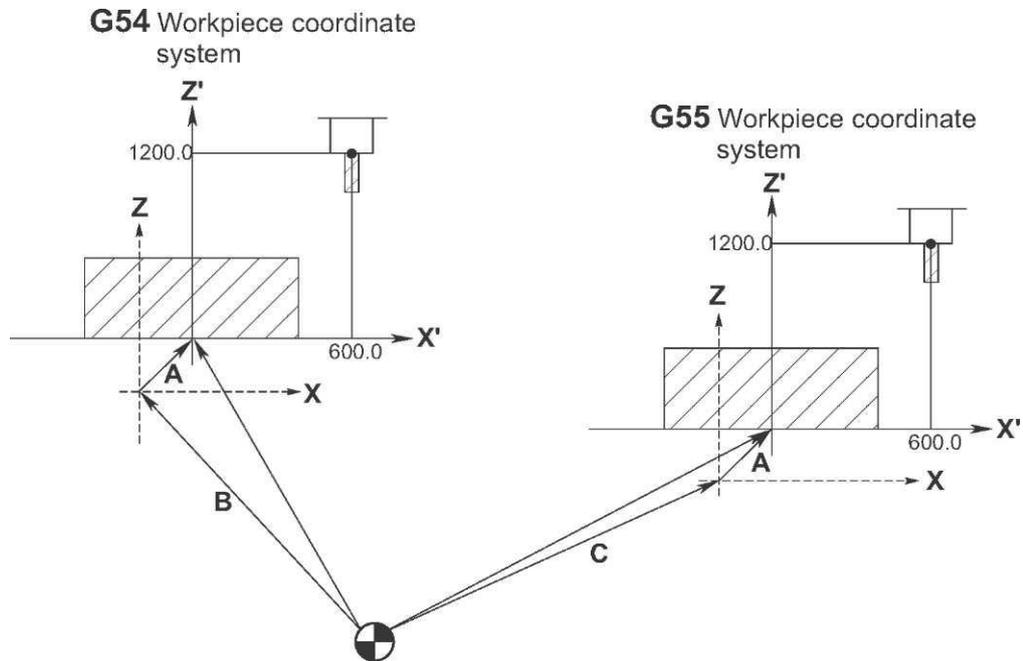
G54 workpiece coordinate system



Coordinate system

If the command **G92 X100.0 Y100.0;** is specified when the tool is in position 200,160 in **G54** mode, a workpiece coordinate system 1 (**X' - Y'**) shifted with vector **A** is set.

A: Offset value created by **G92**



X' - Z' New workpiece coordinate system
X - Z Original workpiece coordinate system

A: Offset value created by **G92**
B: **G54** Workpiece zero point offset value
C: **G55** Workpiece zero point offset value

Suppose that a workpiece coordinate system **G54** is specified. Then a **G55** workpiece coordinate system where the black circle of the tool (figure at the left) is in position 600.0 1200.0 can be set with the following command if the relative relationship between the **G54** workpiece coordinate system and **G55** workpiece coordinate system is set correctly: **G92 X600.0 Z1200.0;** Also suppose that the pallets are loaded at two different positions. If the relationship between the coordinate systems of the pallets in the two positions is correctly set by handling the coordinate systems as the **G54** workpiece coordinate system and **G55** workpiece coordinate system, a coordinate system shift with **G92** in one pallet causes the same coordinate system shift in the other pallet. This means that the workpieces on two pallets can be machined by one and the same program just by specifying **G54** or **G55**.

7.3 PLANE SELECTION

Changing the plane for circular interpolation, cutter compensation and drilling can be set by **G** code.

Plane selection by **G** code

G code	Chosen plane
G17	X Y plane
G18	Z X plane
G19	Y Z plane

The plane is not changed in a block in which

G17, G18 or **G19** do not appear.

Note:

*When the system is switched or in reset **G17** becomes modal.*

8. COORDINATE VALUES AND DIMENSIONS

This chapter consists of the following sections:

ABSOLUTE AND INCREMENTAL PROGRAMMING (G90 AND G91) INCH/METRIC
CONVERSION (G20 AND G21) DECIMAL POINT PROGRAMMING

8.1 ABSOLUTE AND INCREMENTAL PROGRAMMING (G90 AND G91)

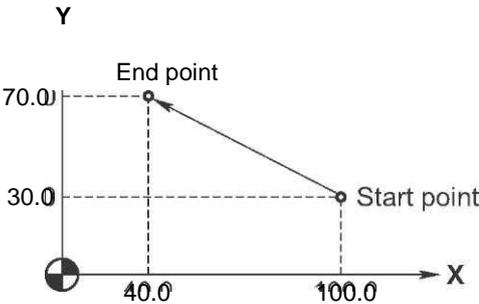
There are two ways for commanding tool movement - absolute and incremental command. In absolute programming the coordinates of the end position are specified. In incremental programming the direction of tool movement to the end point is specified. The **G90** and **G91** commands are used for absolute and incremental programming, respectively.

Format:

Absolute command **G90 IP_;**
Incremental command **G91 IP_;**

Examples:

G90 X40.0 Y70.0; Absolute command **G91 X-60.0
Y40.0;** Incremental command



Coordinate Values and Dimensions

8.2 INCH/METRIC CONVERSION (G20 AND G21)

Either inch or metric input can be selected by **G** command. **Format:**

G20; Inch input **G21;**

Metric input

This **G** code should be specified in a separate block before setting the coordinate system at the beginning of the program. After specifying the **G** code for inch/metric conversion, the unit of input data is switched to the least inch or metric input increment of the corresponding increment system. The unit of input data in degrees stays unchanged. The unit systems for the following values are changed after inch/metric conversion:

- FEEDRATE COMMANDED BY F CODE
- POSITION COMMAND
- WORK ZERO POINT OFFSET VALUE
- TOOL COMPENSATION VALUE
- UNIT OF SCALE FOR MANUAL PULSE GENERATOR
- MOVEMENT DISTANCE IN INCREMENTAL FEED
- SOME PARAMETERS

After the power is on the **G** code is the same as that held before switching off.

WARNING:

1. The **G20** and **G21** commands should not be specified during program execution.
2. When switching from inch input (**G20**) to millimeters input (**G21**) and vice versa, the tool compensation value should be re-set according to the least input increment.

Notes:

1. *When the least input increment and the least command increment systems are different, the maximum error is half of the least command increment. The error is not accumulated.*
2. *Switching between inch and metric input can be done manually by the **INCH** field in the **SETTINGS** screen.*

8.3 DECIMAL POINT PROGRAMMING

The numeric values can be input with decimal point. The decimal point can be used when inputting distance, time and speed. *The decimal point can be specified with the following addresses:*

X, Y, Z, U, V, W, A, B, C, I, J, K, Q, R, F.

There are *two ways for specifying the decimal point: **calculator type and standart type.***

When **calculator** type decimal notation is used it is considered that a value without decimal point is in millimeters, inches or degrees.

When **standart** decimal notation is used, such a value is considered a least input increment.

Choosing between the above types is made by parameter. The values can be specified with or without decimal point in a program.

Examples:

Program command	Pocket calculator type decimal point programming	Standard type decimal point programming
X1000 Command value without decimal point	1000mm Unit: mm	1mm Unit: Least input increment (0.001mm)
X1000.0 Command value with decimal point	1000mm Unit: mm	1000mm Unit: mm

WARNING:

Before entering a value specify the **G** command in a single block. The decimal point position may depend on the command itself.

Coordinate Values and Dimensions

Examples:

G20; Inch input

X1.0 G04; X1.0 is considered to be distance and is processed as X10000. This command is equal to G04 X10000. The tool dwells for 10 seconds.

G04 X1.0; Equal to G04 X1000. The tool dwells for one second.

1. *Fractions less than the least input increment are truncated.*

Notes:

Examples:

X1.2345 The value is truncated to X1.234 when the least input increment is 0.001 mm. Executed as X1.2345 when the least input increment is 0.0001 inch.

2. *When more than eight digits are specified, an alarm occurs. If the value is with decimal point, the number of digits is also checked after the value is converted in an integer value according to the least input increment.*

Examples:

X1.23456789; An alarm occurs because of specifying more than eight digits.

X123456.7 If the least input increment is 0.001 mm, the value is converted to the integer value 123456700. For the value has more than eight digits, an alarm occurs.

9. SPINDLE SPEED FUNCTION (S FUNCTION)

The spindle speed can be controlled with a value following the **S** code.

9.1 SPECIFYING THE SPINDLE SPEED WITH A BINARY CODE

A two digit code can be specified with the **S** code. For more details concerning the **S** codes application, their sequential execution in the block in which they are specified, spindle speed, move command and **S** codes themselves, refer to the manual provided by the machine builder.

9.2 DIRECT SPECIFYING THE SPINDLE SPEED (**S5 - DIGITAL COMMAND**)

The spindle speed can be set directly with **S** code followed by a five digit number (rounds per minute). The unit for specifying the spindle speed can be different for different machines and depends on the machine builder. For more details refer to the manual provided by the machine builder.

10. TOOL FUNCTION (T FUNCTION)

The tool is changed using the tool function.

10.1 TOOL CHANGE FUNCTION

The tools of a machine can be selected by specifying a two- or four-digit number after **T** code.

There can be used only one **T** code in a single block. For the number of digits available for address **T** and the correspondence between **T** codes and the machine operation, refer to the manual provided by the machine tool builder. When a move command and **T** code are specified in a single block, the commands are executed in one of the following ways:

- 1) Simultaneous execution of move command and **T** function.
- 2) Executing **T** function after completion of move command.

The choice **1)** or **2)** depends on the machine tool builder specifications. For more information refer to the manual provided by the machine tool builder.

11. AUXILIARY FUNCTION

There are *two types of auxiliary functions*:

miscellaneous function (M code) for specifying spindle start, spindle stop, end of program, etc. and

secondary auxiliary function (B code) for specifying index table positioning.

When a move command and auxiliary are specified in a single block, the commands are executed in one of the following ways:

- 1) Simultaneous execution of move command and auxiliary function.
- 2) Executing the auxiliary function after completion of move command.

The choice **(1 or 2)** depends on the machine tool builder specifications. For more information refer to the manual provided by the machine tool builder.

11.1 AUXILIARY FUNCTION (M FUNCTION)

When a two-digit number follows the **M** code, code signal and a strobe signal are sent to the machine. The machine uses these signals for turning its functions on and off.

Only one **M** code can be specified in a single block.

The correspondence between the machine function and the **M** code is set by the machine builder.

All **M** codes except **M98** and **M99** - the **M** code for calling a subprogram and the **M** code for calling user macro are processed in the machine. For more information refer to the instruction manual provided by the machine builder.

THE FOLLOWING M CODES HAVE SPECIAL MEANING:

◆ **M02, M03 (End of program)**

This code specifies the end of the program.

The automatic operation is cancelled and the **CNC** device is reset. This can be different with different machine builders.

After executing a block specifying end of program, the control is returned to the beginning of the program.

Return to the beginning of the program can be disabled by parameter.

◆ **M00 (Program stop)**

The automatic operation is stopped when a block containing **M00** is reached. When the program stops, all current modal data stays unchanged. The automatic operation can be resumed by actuating cycle operation. This differs with different machine builders.

◆ **M01 (Optional stop)**

Like **M00**, the automatic operation is stopped after executing a block containing **M00**. This code is effective only when the switch OPTIONAL STOP on the operator's panel is turned on.

◆ **M98 (Calling subprogram)**

This code is used for calling a subprogram. The code and strobe signals are not sent. For more details refer to the subprograms chapter.

◆ **M99 (End of subprogram)**

This code specifies the end of a subprogram. The **M99** code returns the control to the main program. For more details refer to the subprogram chapter.

Note:

*If a block is specified after **M00**, **M01**, **M02** and **M03**, it is not read in the input buffer register. Similarly, two **M** codes which do not buffer can be set by parameters. Refer to the machine builder instruction manual for these **M** codes.*

11.2 SECOND AUXILIARY FUNCTION (B CODE)

Indexing of the table is performed by address **B** and a three- or six-digit number afterwards. The relationship between **B** codes and the corresponding indexing differs between machine builders. For more information refer to the instruction manual provided by the machine builder.

RESTRICTIONS:

This function cannot be used when axis named **B** is used.

12. PROGRAM CONFIGURATION

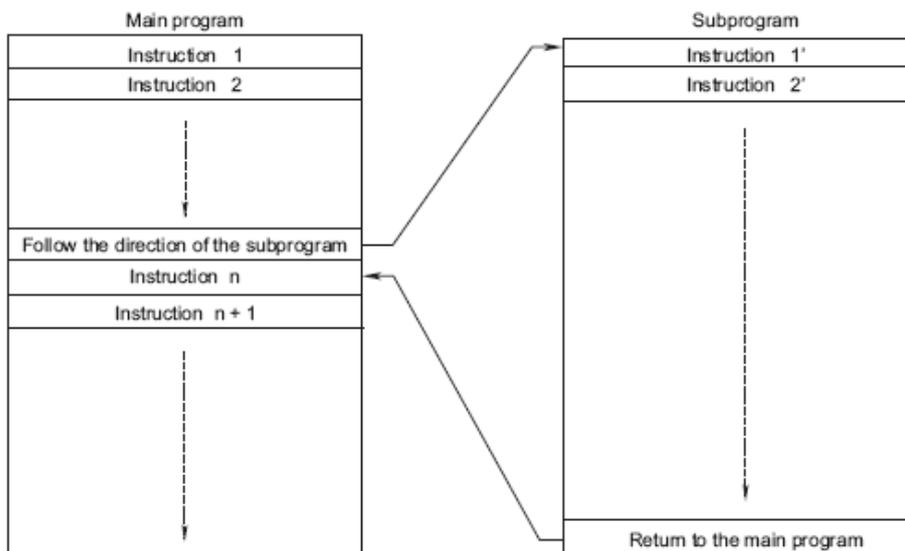
GENERAL

◆ **Main program and subprogram**

There are **two program types** - **main program** and **subprogram**.

Generally, **CNC** executes the main program. When in the main program a subprogram call is encountered, the control is passed to the subprogram. When a return command is reached in the subprogram, the control is returned to the main program.

Program Configuration



The **CNC** memory can hold up to 512 main programs and subprograms. The main program for operating the machine can be selected from the stored main programs. For more information about the methods of registering and selecting the programs refer to the corresponding chapter.

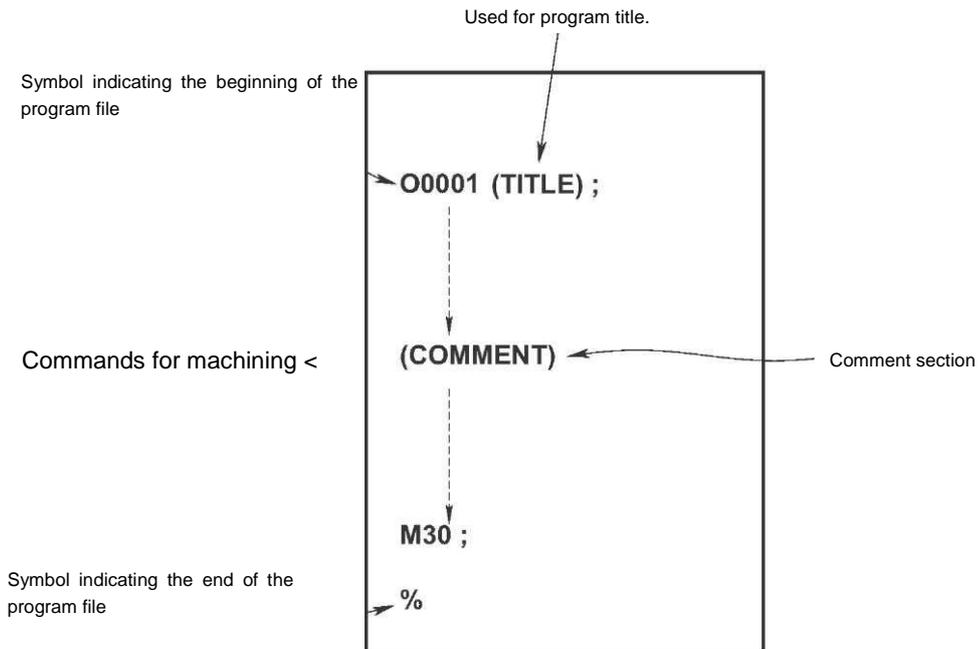
◆ Program components

A program consists of the following components:

Program components

Components	Descriptions
Program start	Symbol indicating the start of a program
Program section	Commands for machining
End	Symbol indicating the end of a program file

◆ Program configuration



Program Configuration

Beginning of the program section

The number of the program is specified by **O** code and four digits afterwards. The comment following is the program name and is used for fast find in the program list. The block ends with a symbol; which indicates end of a block.

◆ Program section configuration

A program section consists of several blocks. It starts with program number and ends with a code indicating the end of the program.

Program section configuration

Program selection

Number of the program	O0001 (Test program)
Block 1	N1 G91 G00 X120.0 Y80.0;
Block 2	N2 G43 Z-32.0 H01;
...	
...	
Block n	Nn M20;
End of the program	M30;

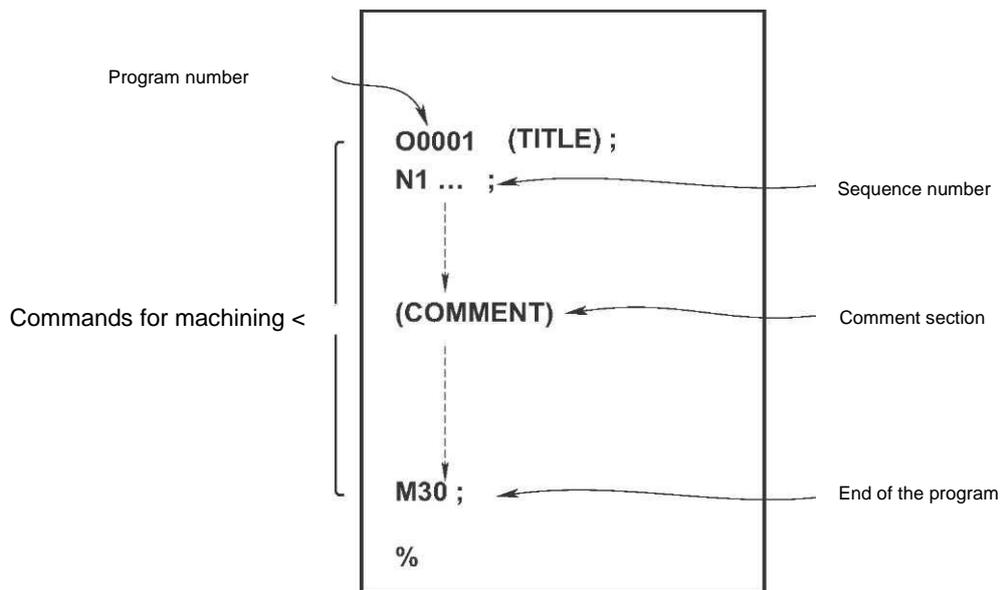
A block contains all the necessary for machining information as a move command or coolant on/off command. Specifying a slash / at the beginning of the block disables the execution of some of the blocks depending on the mode.

12.1 PROGRAM COMPONENTS OTHER THAN PROGRAM SECTIONS

This chapter describes program components other than program sections.

12.2 PROGRAM SECTION CONFIGURATION

This section describes the elements of a program section.



◆ **Program number**

The number of the program consists of four digits specifying the number of the program in the **CNC** memory, so that it can be found for sequential use.

Program Configuration

◆ **Sequence number and block**

A program consists of some commands. Each command unit is called block. The blocks are separated from each other with the symbol “;” which stays at the end of the block.

At the beginning of the block can be placed a sequence number **N** followed by a number with no more than four digits (1 to 9999). Sequence numbers can be specified in random order and some numbers can be omitted. Sequence numbers can be specified on all blocks or just on some of them. In general case, however, it is convenient to put the numbers in ascending order according to the steps for machining (for example when a new tool is used after the change of the old one and the machine continues operation on a new surface with table indexing).

N300 X200.0 Z300.0; Sequence number is underlined.

Note:

***NO** should not be used because of the need of compatibility with other **CNC** machines. There cannot be used a program number **0**. So **0** should not be used as a sequence and as a program number.*

◆ **Block configuration**

A block consists of one or more words. The words consist of address followed by a number with some digits. There could be placed + or - in front of the number.

Word = Address + Number

Example:

The address is specified by one of the letters from **A** to **Z**. The address specifies the meaning of the number following. The table below indicates used addresses and their meanings.

One address can have different meanings depending on the preparatory function specification.

Main functions and addresses

Function	Address	Meaning
Program number	O	Program number
Sequence number	N	Sequence number
Preparatory function	G	Specifies a motion mode (linear, arc, etc)
Dimension word	X, Y, Z, U, V, W, A, B, C	Coordinate axis move command
	I, J, K	Coordinate of the arc center
	R	Radius
Feed function	F	Rate of feed per minute Rate of feed per revolution
Spindle speed function	S	Spindle speed
Tool function	T	Tool number
Auxiliary function	M	On/off control on the machine functions
	B	Table indexing
Offset number	D, H	Offset number
Dwell	P, X	Dwell
Program number designation	P	Subprogram number
Number of repetitions	P	Number of subprogram repetitions
Parameter	P Q	Canned cycle parameters

Example for a block

N_ G_ X_ Y_ F_ S_ T_ M_;

- N_** - Sequence number
- G_** - Preparatory function
- X_ Y_** - Dimension word
- F_** - Feed function
- S_** - Spindle speed function
- T_** - Tool function
- M_** - Miscellaneous function

◆ **Major addresses and ranges of command values**

Major addresses and ranges of values specified for the addresses are shown below. Note that these figures represent limits on the **CNC** side which are totally different from the limits of the machine and the tool.

For example, **CNC** allows tool movement up to 24 meters (millimeter input) along **X** axis. However, the actual stroke along **X** axis may be limited to 2 meters for a specific tool.

Similarly, **CNC** can execute cutting feedrate of up to 15 meters/min. When the program is created, the user must read carefully the machine tools manual together with this manual to be well known the limits while programming.

Major addresses and ranges of command values

Function	Address	Input in mm	Input in inch
Program number	O	0001 - 9999	0001 - 9999
Sequence number	N	1 - 9999	1 - 9999
Preparatory function	G	0 - 99	0 - 99
Dimension word	X, Y, Z, U, V, W, A, B, C, I, J, K, R	± 9999.999mm	± 999.9999inch
Feed function	F	15000mm/min	600inch/min
Spindle speed function	S	0 - 20000	0 - 20000
Tool function	T	0 - 9999	0 - 9999
Auxiliary function	M	0 - 99	0 - 99
	B	0 - 999999	0 - 999999
Offset number	D, H	0 - 99	0 - 99
Dwell	P, X	0 -9999.999	0 - 9999.999
Program number designation	P	1 - 9999	1 - 9999
Number of repetitions	P	1 - 999	1 - 999

◆ Optional block skip

When a slash is specified at the beginning of the block and the switch for block skip is in position on, all the data in the block is ignored. When the switch for block skip is in off position, the data in the block with slash at the head is valid. This actually means that the operator may specify including or excluding of each block separately. Programs held in the memory can be output regardless of the switch position.

WARNING:

1. Position of the slash

The slash / must be put at the beginning of the block.

2. Optional skip block switch disable

Optional skip block operation is performed when blocks are read from the memory in a buffer. Even if the switch is in on position, after the blocks have been read in the buffer, the blocks are ignored.

◆ Program end

End of a program is specified using one of the following codes at the end of the program:
If one of codes for program end is executed during machining, CNC stops

Program end code

Code	Meaning
M02	For main program
M30	For main program
M99	For subprogram

program execution and reset state is set. When a code for end of a subprogram is executed, the control is returned to the program which has called the subprogram.

WARNING:

Block containing optional skip block code such as **/M02**; **/M30**; or **/M99**; is not regarded as program end if the optional block switch on the machine control panel is in position on.

Program Configuration

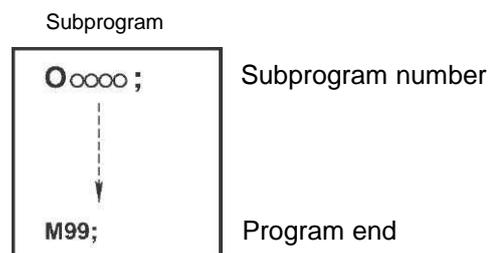
12.3 SUBPROGRAM

If a program contains a fixed sequence of commands or frequently used pattern, such a sequence can be held in the memory as a subprogram for simplifying the main program.

The subprogram can be called by the main program. The called program can call another program.

Format:

Subprogram configuration

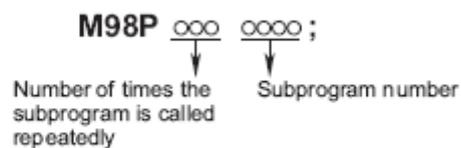


It is not necessary **M99** to be a separate block as shown below:

Example:

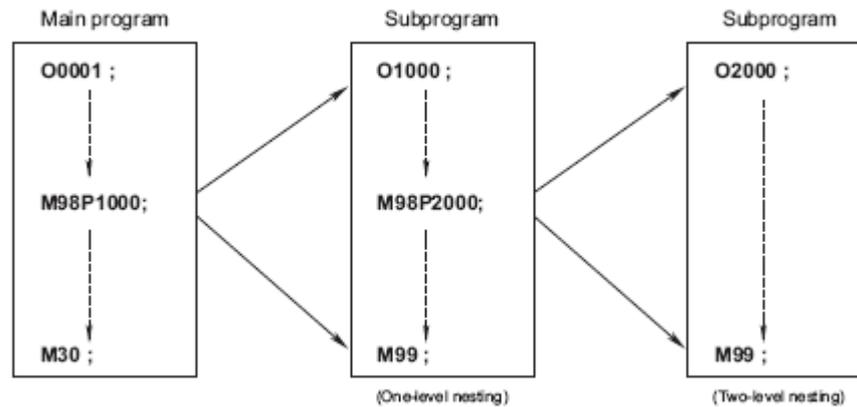
X100.0 Y100.0 M99;

◆ Subprogram call



When no repetition data is specified for subprogram execution, the subprogram is called just once.

When the main program call a subprogram, it is regarded as one level subprogram call. Thus subprogram calls can be nested as shown below:



Maximum nested level is 8.

One call command can call a subprogram up to 999 times.

Note:

1. The **M98** and **M99** signals are not output to the tool of the machine.
2. If the subprogram number specified with **P** address cannot be found, an alarm is issued.

Examples:

- **M98 P51002;**

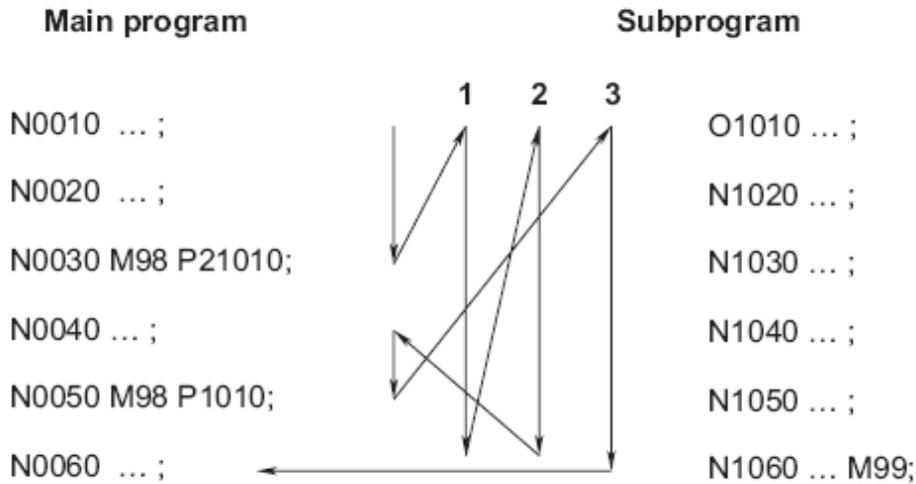
This command specifies call of subprogram 1002 five times in succession. The subprogram call command **M98P_** can be specified in the same block as the move command.

- **X1000.0 M98 P1200;**

This example calls a subprogram 1200 after movement along **X** axis.

Program Configuration

- ◆ Execution sequence of subprograms called by the main program



A subprogram can call another subprogram in the same way as the main program calls a subprogram.

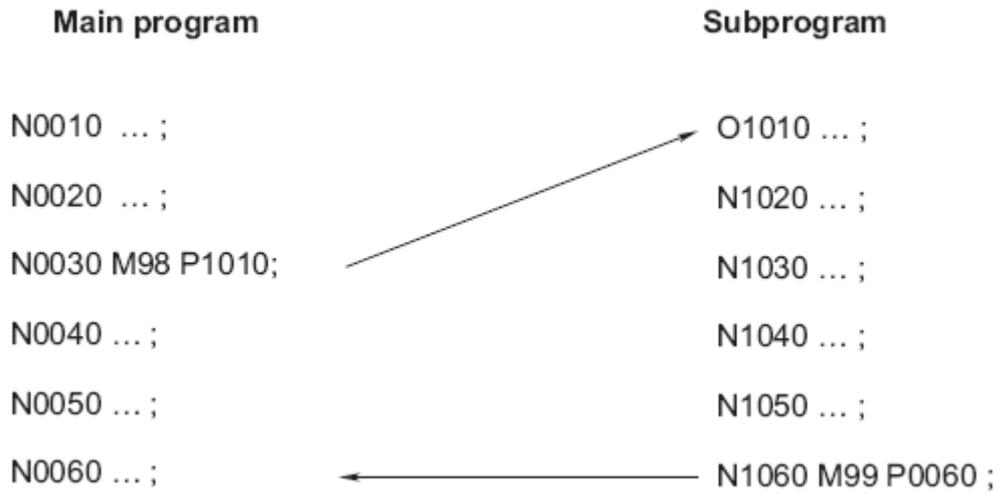
SPECIAL ; **USAGE:**

- ◆ Specifying the sequence number for the return destination in the main program

If **P** is used for specifying the sequence number when a subprogram is terminated, the control is not returned to the block after the block called the subprogram but is returned to the block with sequence number specified in **P**.

Note, however, that **P** is ignored if the main program is executed in a mode different than memory operation.

This method consumes much more time than the usual one for returning to the main program.



◆ **Using M99 in the main program**

If **M99** is executed in the main program the control is passed to the beginning of the main program.

For example **M99** can be executed when **/M99;** is put in an appropriate position in the main program and the optional skip block function is off when the main program is executed.

When **M99** is executed, the control is returned to the main program and the operations are repeated from the beginning of the main program afterwards.

These operations are repeated while the optional skip block function is off. If the optional skip block function is on, block **/M99;** is ignored and the control is passed to the next block for continued execution.

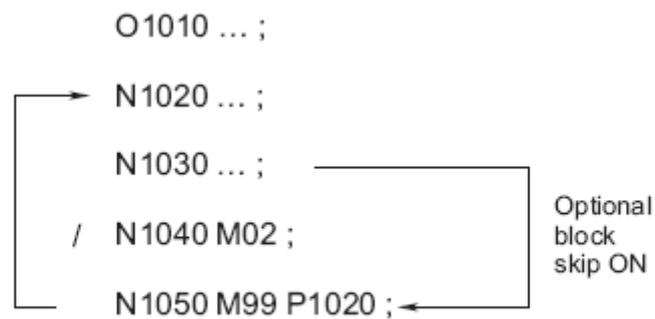
If **/M99Pn;** is specified, the control is passed not to the beginning of the main program but to the sequence number n. In this case, much longer time is needed for returning to sequence number n.



◆ **Using a subprogram only**

A subprogram can be executed as a main program by searching its start with the keyboard.

In this case, if a block containing **M99** is executed, the control is passed to the beginning of the subprogram for repeated execution. If a block containing **M99Pn** is executed, the control is passed to the block with sequence number n in the subprogram for repeated execution. To interrupt the program you must specify a block with **/M02**; or **/M30**; on a proper position and the optional skip block must be off; the switch must be on in the beginning.



13. FUNCTIONS TO SIMPLIFY PROGRAMMING

This chapter explains the following items: CANNED
CYCLE RIGID TAPPING
EXTERNAL MOTION FUNCTION

13.1 CANNED CYCLE

Canned cycles simplify programmer's work while creating programs. In canned cycles the operation which is used most frequently may be specified in one block with G function. Without canned cycles, usually, more than one block is required. More than that, the user who uses canned cycles makes the program shorter and saves memory.

The table shown below lists all canned cycles.

Canned Cycles

G code	Drilling (-Z direction)	Operation at the bottom of a hole	Retraction (+Z direction)	Application
G73	Intermittent feed	-	Rapid traverse	High-speed peck drilling cycle
G74	Feed	Dwell - Spindle CW	Feed	Left hand tapping cycle
G76	Feed	Spindle orientation	Rapid traverse	Fine boring cycle
G80	-	-	-	Cancel
G81	Feed	-	Rapid traverse	Drilling cycle, spot drilling cycle
G82	Feed	Dwell	Rapid traverse	Drilling cycle, counter boring cycle
G83	Intermittent feed	-	Rapid traverse	Peck drilling cycle
G84	Feed	Dwell-Spindle CCW	Feed	Tapping cycle
G85	Feed	-	Feed	Boring cycle
G86	Feed	Spindle stop	Rapid traverse	Boring cycle
G87	Feed	Spindle CW	Rapid traverse	Back boring cycle
G88	Feed	Dwell-spindle stop	Manual	Boring cycle
G89	Feed	Dwell	Feed	Boring cycle

A canned cycle consists of six sequential operations:

Operation 1 - Positioning X and Y axes (including other axes)

Operation 2 - Rapid traverse up to point R level

Operation 3 - Hole machining

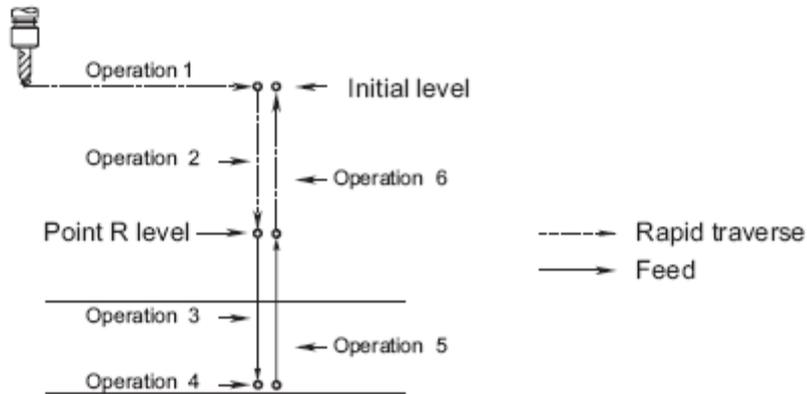
Operation 4 - Operation at the bottom of the hole

Operation 5 - Retraction to point R level

Operation 6 - Rapid traverse up to the initial point

Canned cycle operation sequence

Functions to Simplify Programming

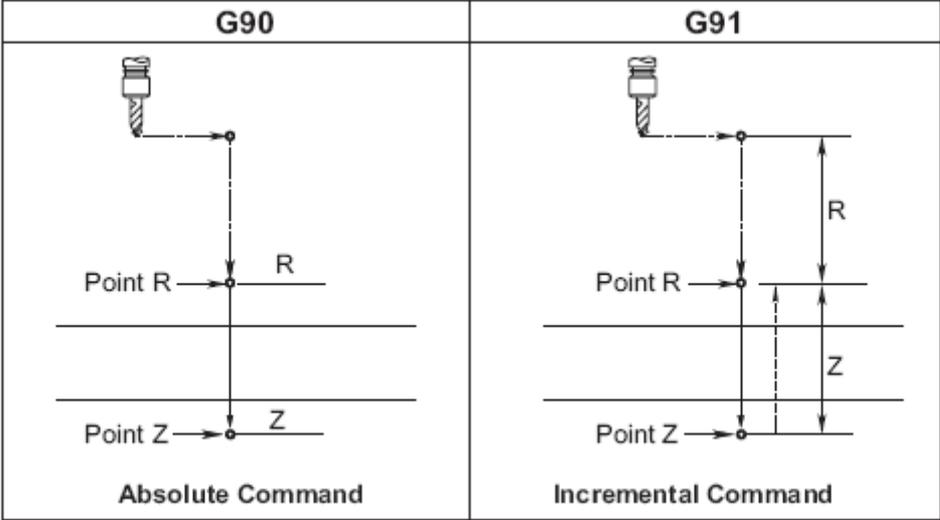


◆ Positioning plane

The positioning plane is **XY** and **Z** is the drilling axis. **WARNING:**
Switch the drilling axis after cancelling a canned cycle.

◆ Travel distance along the drilling axis G90/G91

The travel distance along the drilling axis is different while using **G90** or **G91** and is as follows:



◆ **Drilling mode**

G73, G74, G76 and **G81** to **G89** are **modal codes** and stay valid until cancelled. When they are valid, the current state is drilling mode.

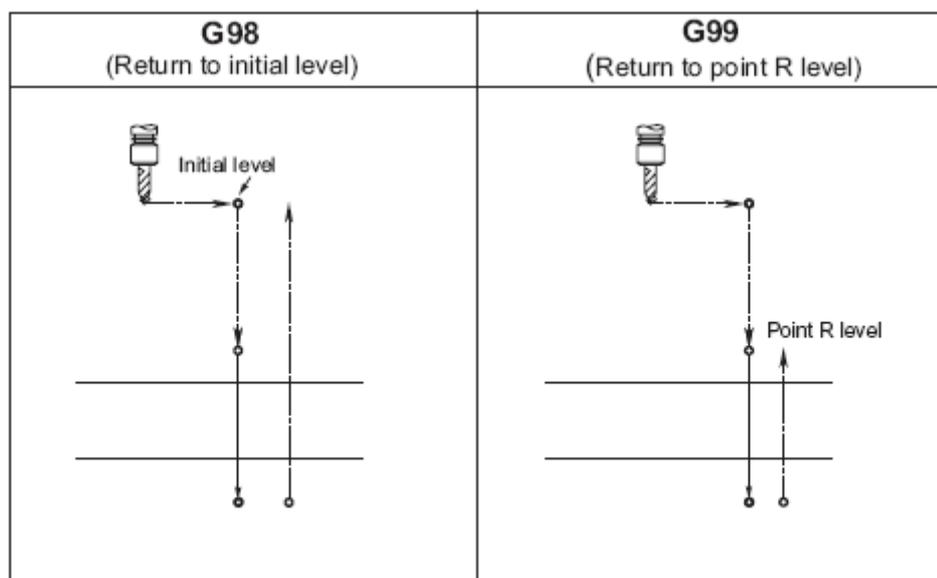
Once the data is set for drilling mode, it stays unchanged until modified or cancelled

Specify all necessary data for drilling mode at the beginning of the canned cycle. While executing the canned cycle modify only the parameters in use.

◆ **Return point level**

When the tool reaches the bottom of the hole, it can return to point R or to the initial level. These operations are specified with **G98** or **G99** commands. Figures below show the tool path when **G98** or **G99** is commanded. Normally, **G99** is used for the first drilling operation while **G98** is used for the last one.

The initial level does not change even when drilling is performed in **G99** mode.



◆ **Repetitions**

To repeat drilling for equally-spaced hole specify the number of repeats in **K_**;
K is effective only in the block in which has been specified.

Specify the first hole position in incremental mode (**G91**).

If it is set in absolute mode (**G90**) the dwell is performed at the same position.

Number of repeats K: Maximum command value - 9999

If **K0** is specified, all the drilling data is stored but drilling operation is not performed.

◆ **Cancel**

To cancel the cycle use **G80** command or **G** code from group **01**.

G codes from group 01:

G00: Positioning (rapid traverse)

G01: Linear interpolation

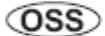
G03: Circular interpolation (CW)

G04: Circular interpolation (CCW)

Symbols in the figures

The next section illustrates all different kinds of canned cycles. Figures in these illustrations use the following symbols:

Functions to Simplify Programming

	Positioning (rapid traverse G00)
	Cutting feedrate (linear interpolation G01)
	Manual feed
	Spindle orientation (spindle stops at a fixed rotation position)
	Shift (rapid traverse G00)
P	Dwell

13.1.1 High-speed Peck Drilling Cycle (G73)

This cycle performs high-speed peck drilling. It performs intermittent feed to the bottom of the hole while removing the chips.

Format:

G73 X_ Y_ Z_ R_ Q_ F_ K_ ;

X_ Y_ : *Hole position data*

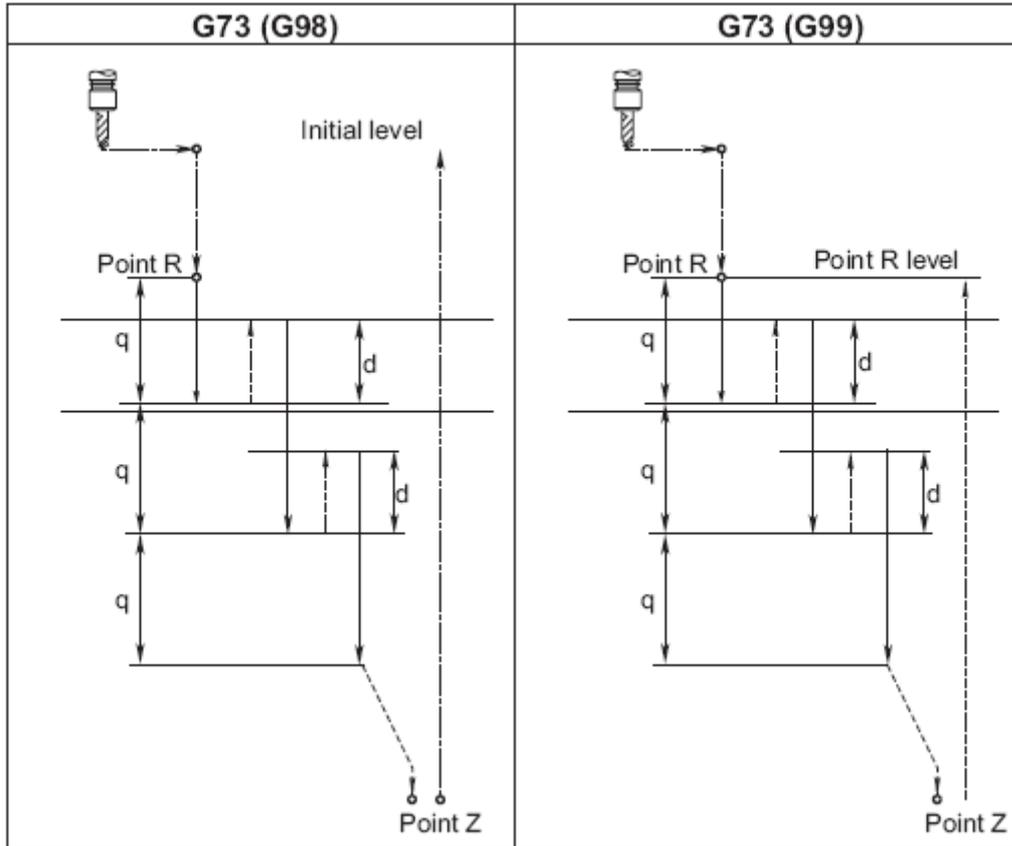
Z_ : *The distance from point R to the bottom of the hole*

R_ : *The distance from the initial level to the point R level*

Q_ : *Depth of cut for each cutting feed*

F_ : *Cutting federate*

K_ : *Number of repeats*



High-speed peck drilling cycle is performed with intermittent feed of the tool along **Z** axis. When this cycle is used, chips can be easily removed and the retraction value can be smaller. This allows more effective drilling. Set the clearance in the corresponding parameter. The tool is retracted in rapid traverse.

Before specifying **G73** rotate the spindle using the auxiliary function (**M** code). When **G73** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in a canned cycle, the shift is performed during positioning in **R** point.

RESTRICTIONS:

♦ **Drilling**

Drilling is not performed in a block which do not contain X, Y,Z, R or other axes.

♦ **Q/R**

Specify **Q** or **R** in blocks that perform drilling. If they are not set in a block performing drilling operation, they cannot be stored as a modal data.

♦ **Cancel**

Do not specify codes from group **01 (G00 to G03)** and **G73** in one block. If they are used together, **G73** is cancelled

Examples:

M3 S2000;	Cause the spindle to start rotating
G90 G99 G73 X300. Y-250. Z-150. R-100. Q15. F120.;	Positioning, drill hole 1 and return to point R
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position Cause the
M5;	spindle to stop rotating

13.1.2 Left-handed Tapping Cycle(G74)

This cycle performs left-handed tapping cycle. In this cycle, when the bottom of the hole is reached, the spindle is rotated clockwise.

Format:

G74 X_ Y_ Z_ R_ P_ F_ K_ ;

X_ Y_ : *Hole position data*

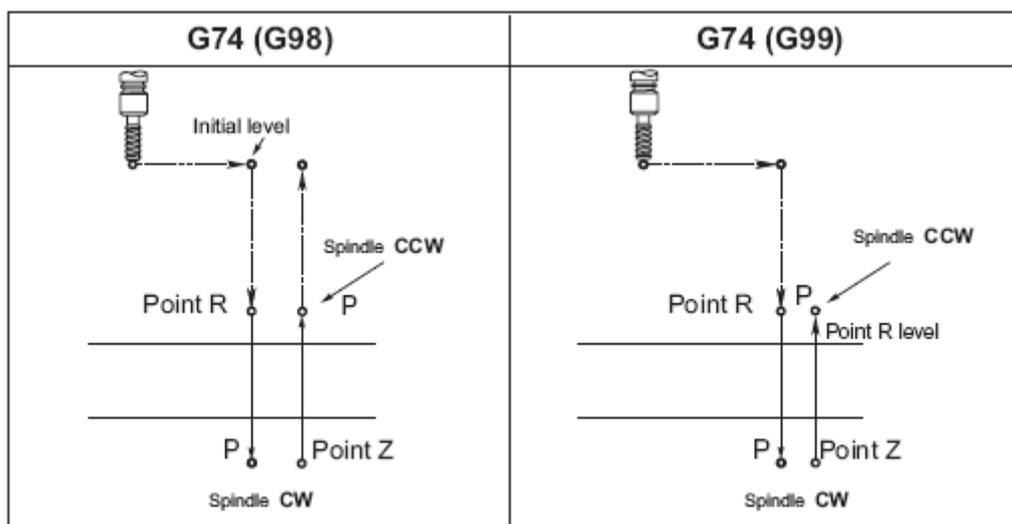
Z_ : *The distance from point R to the bottom of the hole*

R_ : *The distance from the initial level to the point R level*

P_ : *Dwell time*

F_ : *Cutting feedrate*

K_ : *Number of repeats*



Functions to Simplify Programming

Tapping cycle is performed by turning the spindle counter clockwise. When the bottom of the hole is reached, the spindle rotates clockwise for retraction. This creates a reverse thread.

Feedrate overrides are ignored during this operation. A feed hold does not stop the machine until retraction operation is completed.

Before specifying **G74** rotate the spindle counter clockwise using the auxiliary function (**M** code). When **G74** and **M** code are specified in a block , the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified, the shift is performed during positioning in R point.

RESTRICTIONS:

◆ Drilling

Drilling is not performed in a block which do not contain X, Y,Z, R or other axes.

◆ R

Specify R in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ Cancel

Do not specify codes from group 01 (**G00** to **G03**) and **G74** in one block. If they are used together, **G74** is cancelled

Examples:

M4 S100;	Cause the spindle to start rotating
G90 G99 G74 X300. Y-250. Z-150. R-120. F120.;	Positioning, tapping hole 1 and return to point R
Y-550.;	Positioning, tapping hole 2 and return to point R
Y-750.;	Positioning, tapping hole 3 and return to point R

Functions to Simplify Programming

X1000.;	Positioning, tapping hole 4 and return to point R
Y-550.;	Positioning, tapping hole 5 and return to point R
G98 Y-750.;	Positioning, tapping hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.3 Fine Boring Cycle (G76)

Fine boring cycle bores a hole precisely. When the bottom of the hole is reached the spindle stops and the tool is moved slowly from the machined surface of the workpiece and retraced.

Format:

G76 X_ Y_ Z_ R_ Q_ P_ F_ K_ ;

X_ Y_ : Hole position data

Z_ : The distance from point R to the bottom of the hole

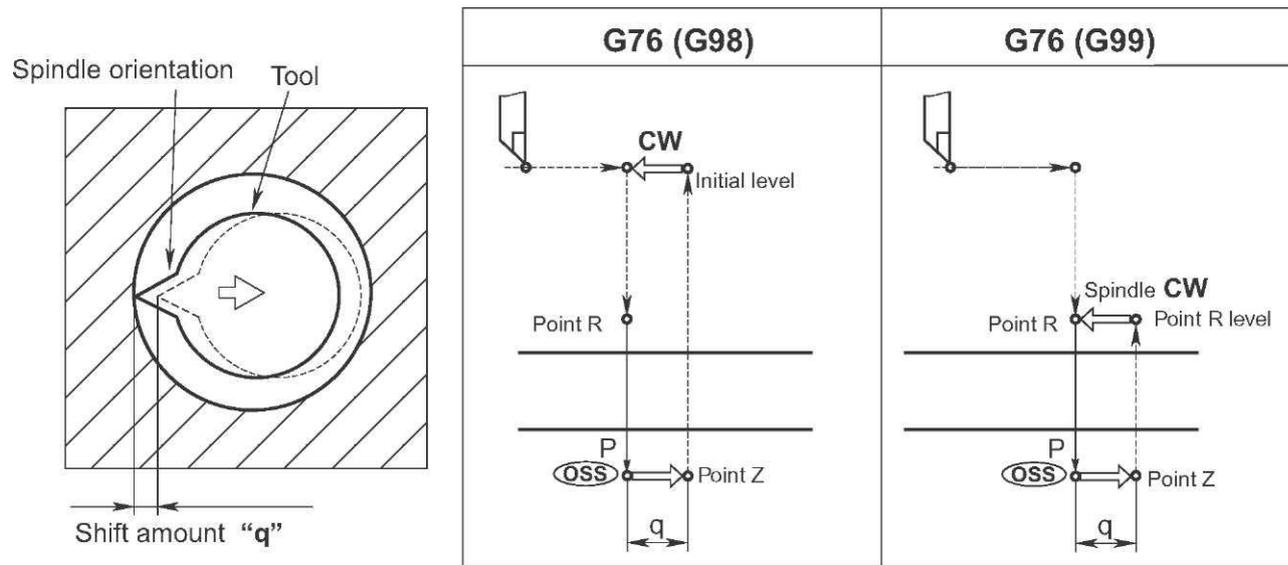
R_ : The distance from the initial level to the point R level

Q_ : Shift amount at the bottom of the hole

P_ : Dwell time

F_ : Cutting feedrate

K_ : Number of repeats



WARNING:

Q (shift at the bottom of the hole) is a **modal value** which retains in the canned cycle. It has to be set very carefully for it is used as the depth of cut for **G73** and **G83**.

When the bottom of the hole is reached, the spindle stops at a fixed rotation position and the tool is shifted in direction opposite to its tip and retracted. This ensures that the machined surface of the workpiece is not damaged and execution of efficient boring.

Before specifying **G76** rotate the spindle using the auxiliary function (M code). When **G76** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in R point.

Functions to Simplify Programming

RESTRICTIONS:

◆ Boring

Boring is not performed in a block which do not contain **X, Y, Z, R** or **other** axes.

◆ Q/R

Be sure to specify a positive value for **Q**. If the value is negative, the sign is ignored. Shift direction is set by parameter. Specify **Q** and **R** in blocks that perform drilling. If they are not set in a block performing drilling operation, they cannot be stored as a modal data.

◆ Cancel

Do not specify codes from group **01 (G00 to G03)** and **G76** in one block. If they are used together, **G76** is cancelled

Examples:

M3 S500;	Cause the spindle to start rotating
G90 G99 G76 X300. Y-250.	Positioning, bore hole 1 and return to point R
Z-150. R-120. Q5;	Orient at the bottom of the hole and shift by 5 mm
P1000 F120;	Stop at the bottom of the hole for 1 second
Y-550.;	Positioning, bore hole 2 and return to point R
Y-750.;	Positioning, bore hole 3 and return to point R
X1000.;	Positioning, bore hole 4 and return to point R
Y-550.;	Positioning, bore hole 5 and return to point R

G98 Y-750.;	Positioning, bore hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.4 Drilling Cycle, Spot Drilling Cycle (G81)

This cycle is used for normal drilling. Cutting feed is performed to the bottom of the hole. Then the tool is retraced from the bottom of the hole an rapid traverse.

Format:

G81 X_ Y_ Z_ R_ F_ K_ ;

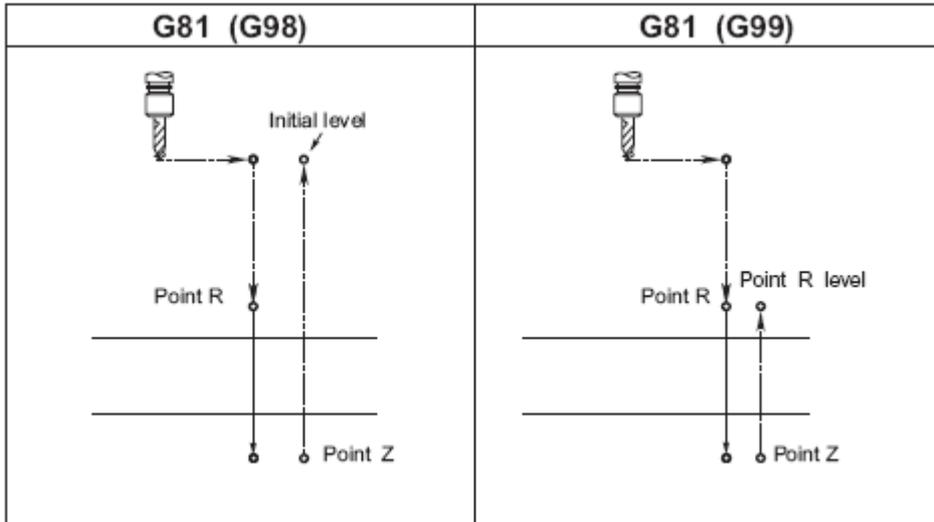
X_ Y_: *Hole position data*

Z_ : *The distance from point R to the bottom of the hole*

R_ : *The distance from the initial level to the point R level*

F_ : *Cutting feedrate*

K_ : *Number of repeats*



After positioning along **X** and **Y** axes, rapid traverse is performed to point **R**. Drilling is executed from point **R** to point **Z**.

Before specifying **G81** rotate the spindle using the auxiliary function (**M** code). When **G81** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in **R** point.

RESTRICTIONS:

Drilling

Drilling is not performed in a block which do not contain **X**, **Y**, **Z**, **R** or other axes.

◆ **R**

Specify **R** in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ **Cancel**

Do not specify codes from group 01 (**G00** to **G03**) and **G81** in one block. If they are used together, **G81** is cancelled

Examples:

M3 S2000;	Cause the spindle to start rotating
G90 G99 G81 X300. Y-250. Z-150. R-100. F120.;	Positioning, drill hole 1 and return to point R
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R

Functions to Simplify Programming

G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.5 Drilling Cycle, Counter Boring Cycle (G82)

This cycle is used for normal drilling. Cutting feed is performed to the bottom of the hole. A dwell is performed at the bottom and then the tool is retraced from the bottom of the hole an rapid traverse. This cycle is used to drill holes more accurately with respect to depth.

Format:

G82 X_ Y_ Z_ R_ P_ F_ K_ ;

X_ Y_ : *Hole position data*

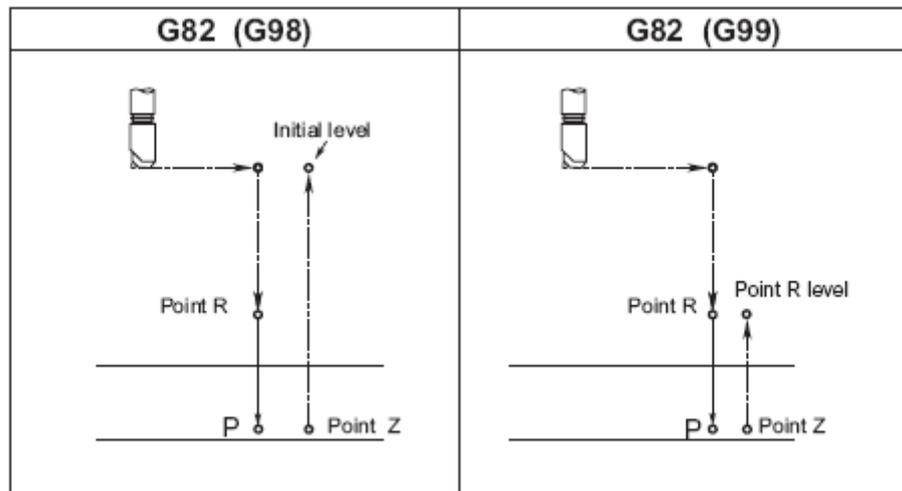
Z_ : *The distance from point R to the bottom of the hole*

R_ : *The distance from the initial level to the point R level*

P_ : *Dwell time at the bottom of the hole*

F_ : *Cutting feedrate*

K_ : *Number of repeats*



After positioning along **X** and **Y** axes, rapid traverse is performed to point **R**. Drilling is executed from point **R** to point **Z**.

Before specifying **G82** rotate the spindle using the auxiliary function (**M** code). When **G82** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in **R** point.

RESTRICTIONS:

◆ Drilling

Drilling is not performed in a block which do not contain **X**, **Y**, **Z**, **R** or other axes.

◆ **R**

Specify **R** in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ **Cancel**

Do not specify codes from group 01 (**G00** to **G03**) and **G82** in one block. If they are used together, **G82** is cancelled

Examples:

M3 S2000;	Cause the spindle to start rotating
G90 G99 G82 X300. Y-250. Z-150. R-100. P1000 F120.;	Positioning, drill hole 1, dwell for 1 second at the bottom of the hole and return to point R
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.; G98	Positioning, drill hole 5 and return to point R Positioning, drill
Y-750.;	hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.6 Peck Drilling Cycle (G83)

This cycle performs peck drilling. It performs intermittent feed to the bottom of the hole while removing the chips.

Format:

G83 X_ Y_ Z_ R_ Q_ F_ K_ ;

X_ Y_: *Hole position data*

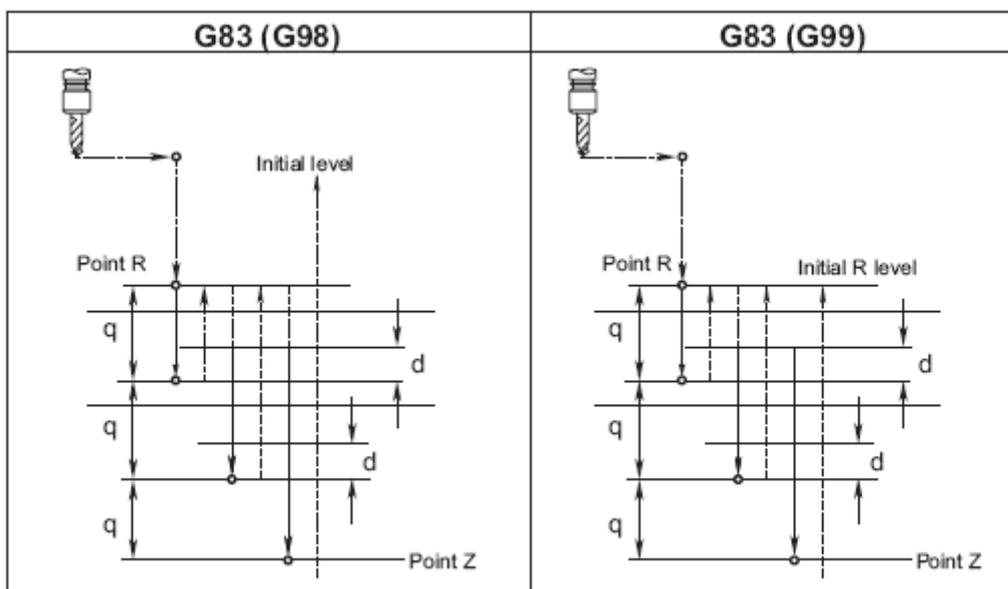
Z_ : *The distance from point R to the bottom of the hole*

R_ : *The distance from the initial level to the point R level*

Q_ : *Depth of cut for each cutting feed*

F_ : *Cutting federate*

K_ : *Number of repeats*



Q is depth of cut for each cutting feed. It is always specified as an incremental value.

During the second and the following feeds, rapid traverse is performed to the point just before where the last drilling ended and cutting feed is performed again. The number of returns is set in a parameter. Be sure to specify a positive value for **Q**. The negative values are ignored.

Functions to Simplify Programming

Before specifying **G83** rotate the spindle using the auxiliary function (**M** code). When **G83** and **M** code are specified in a block , the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43, G44** or **G49**) is specified in a canned cycle, the shift is performed during positioning in R point.

RESTRICTIONS:♦ **Drilling**

Drilling is not performed in a block which do not contain **X, Y, Z, R** or other axes.

♦ **Q/R**

Specify **Q** or **R** in blocks that perform drilling. If they are not set in a block performing drilling operation, they cannot be stored as a modal data.

♦ **Cancel**

Do not specify codes from group 01 (**G00** to **G03**) and **G83** in one block. If they are used together, **G83** is cancelled.

Examples:

M3 S2000;	Cause the spindle to start rotating
G90 G99 G83 X300. Y-250. Z-150. R-100. Q15. F120.;	Positioning, drill hole 1 and return to point R
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

Functions to Simplify Programming

13.1.7 Tapping Cycle (G84)

This cycle performs tapping.

In this tapping cycle the spindle is rotated in the reverse direction when the bottom of the hole is reached.

Format:

G84 X_ Y_ Z_ R_ P_ F_ K_ ;

X_ Y_ : Hole position data

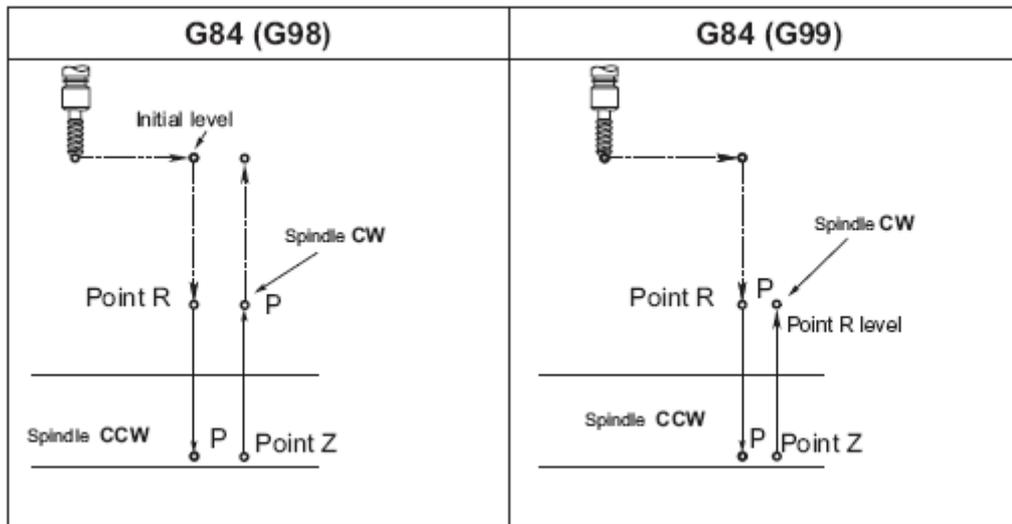
Z_ : The distance from point R to the bottom of the hole

R_ : The distance from the initial level to the point R level

P_ : Dwell time

F_ : Cutting feedrate

K_ : Number of repeats



Tapping cycle is performed by rotating the spindle clockwise. When the bottom of the hole is reached, the spindle is rotated in the opposite direction and retracted. This operation creates threads.

All feedrate overrides are ignored during this cycle. A feed hold does not stop the machine until the return operation is completed.

Before specifying **G84** rotate the spindle using the auxiliary function (**M** code). When **G84** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43, G44 or G49**) is specified in the canned cycle, the shift is performed during positioning in R point.

RESTRICTIONS:

◆ **Drilling**

Drilling is not performed in a block which do not contain **X, Y, Z, R** or other axes.

◆ **R**

Specify **R** in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ **Cancel**

Functions to Simplify Programming

Do not specify codes from group 01 (**G00** to G03) and G84 in one block. If they are used together, G84 is cancelled.

Examples:

M3 S2000;	Cause the spindle to start rotating
G90 G99 G84 X300. Y-250. Z-150. R-120. P300 F120.;	Positioning, drill hole 1 and return to point R
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.8 Boring Cycle (G85)

This cycle is used to bore a hole.

Format:

G85 X_ Y_ Z_ R_ F_ K_ ;

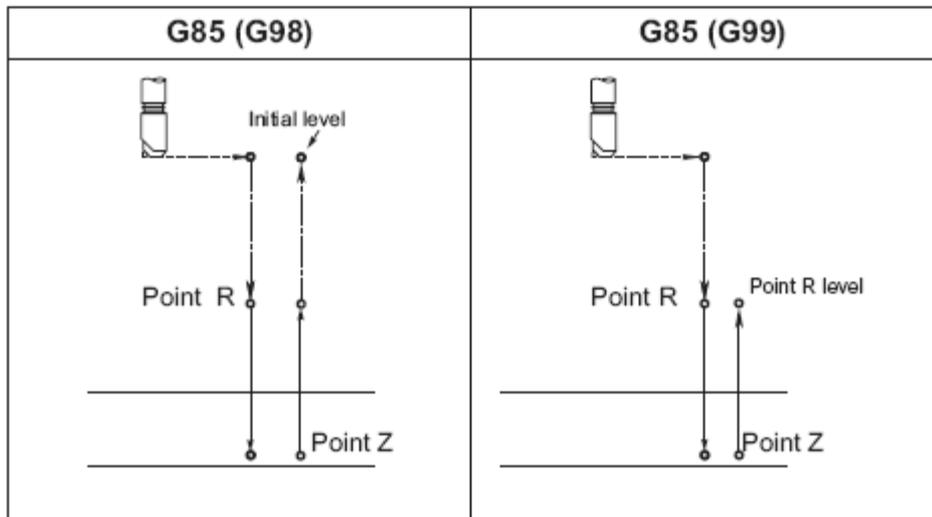
X_ Y_ : Hole position data

Z_ : The distance from point R to the bottom of the hole

R_ : The distance from the initial level to the point R level

F_ : Cutting feedrate

K_ : Number of repeats



After positioning along **X** and **Y** axes, rapid traverse is performed to point **R**.

Drilling is executed from point **R** to point **Z**. When point **Z** is reached, cutting feed is performed to return to point **R**.

Before specifying **G85** rotate the spindle using the auxiliary function (**M** code). When **G85** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in **R** point.

RESTRICTIONS:

◆ **Drilling**

Drilling is not performed in a block which do not contain **X**, **Y**, **Z**, **R** or other axes.

◆ **R**

Specify **R** in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ **Cancel**

Do not specify codes from group 01 (**G00** to **G03**) and **G85** in one block. If they are used together, **G85** is cancelled.

Examples:

M3 S100;	Cause the spindle to start rotating
G90 G99 G85 X300. Y-250. Z-150. R-120. F120.;	Positioning, drill hole 1 and return to point R
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.9 Boring Cycle (G86)

Functions to Simplify Programming

This cycle is used to bore a hole.

Format:

G86 X_ Y_ Z_ R_ F_ K_ ;

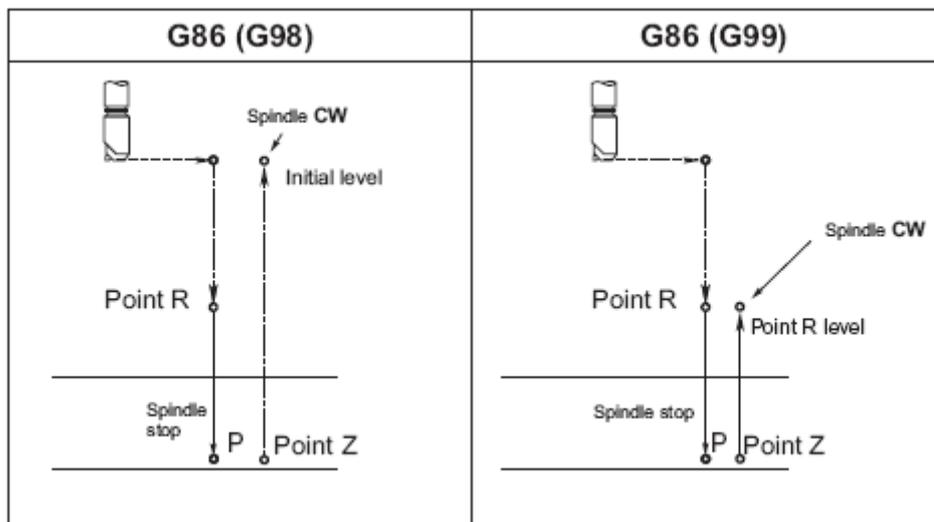
X_ Y_: *Hole position data*

Z_ : *The distance from point R to the bottom of the hole*

R_ : *The distance from the initial level to the point R level*

F_ : *Cutting federate*

K_ : *Number of repeats*



After positioning along **X** and **Y** axes, rapid traverse is performed to point **R**. Drilling is executed from point **R** to point **Z**.

The spindle stop at the bottom of the hole and the tool is returned in rapid traverse.

Before specifying **G86** rotate the spindle using the auxiliary function (**M** code). When **G86** and **M** code are specified in a block , the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the

Functions to Simplify Programming

first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in R point.

RESTRICTIONS:

◆ Drilling

Drilling is not performed in a block which do not contain **X**, **Y**, **Z**, **R** or other axes.

◆ R

Specify R in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ Cancel

Do not specify codes from group 01 (**G00** to **G03**) and **G86** in one block. If they are used together, **G86** is cancelled.

Examples:

M3 S2000;	Cause the spindle to start rotating
G90 G99 G86 X300. Y-250. Z-150. R-100. F120.;	
	Positioning, drill hole 1 and return to point R
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position

Functions to Simplify Programming

K_ : *Number of repeats*

WARNING:

Q (shift at the bottom of the hole) is a modal value which retains in the canned cycle. It has to be set very carefully for it is used as the depth of cut for **G73** and **G83**.

After positioning along **X** and **Y** axes, the spindle stops at a predetermined fixed

rotation position. The tool is moved in direction opposite to its tip and the positioning (in rapid traverse) is performed at the bottom of the hole (point R).

The tool is shifted in the direction of its tip afterwards and the spindle is rotated clockwise. Boring is performed in positive direction along **Z** axis until point **Z** is reached. The spindle is stopped at a fixed rotation position, the tool is shifted in direction opposite to its tip and is returned to the initial level afterwards. Then the tool is shifted in direction opposite to its tip and the spindle starts rotating clockwise and continues performing the operations of the next block.

Before specifying **G87** rotate the spindle using the auxiliary function (M code). When **G87** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in R point.

RESTRICTIONS:

◆ Boring

Boring is not performed in a block which do not contain **X**, **Y**, **Z**, **R** or other axes.

◆ Q/R

Be sure to specify a positive value for **Q**. If the value is negative, the sign is ignored. Shift direction is set by parameter. Specify **Q** and **R** in blocks that perform drilling. If they are not set in a block performing drilling operation, they cannot be stored as a modal data.

◆ Cancel

Do not specify codes from group 01 (**G00** to **G03**) and **G87** in one block. If they are used together, **G87** is cancelled.

Examples:

Functions to Simplify Programming

M3 S500;	Cause the spindle to start rotating
G90 G87 X300. Y-250. Z-150 R-120. Q5 P1000 F130;	Positioning, bore hole 1
	Orient at the initial level and shift by 5 mm
	Stop at point Z for 1 second
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.11 Boring Cycle (G88)

This cycle is used to bore a hole.

Format:

G88 X_ Y_ Z_ R_ P_ F_ K_ ;

X_ Y_ : Hole position data

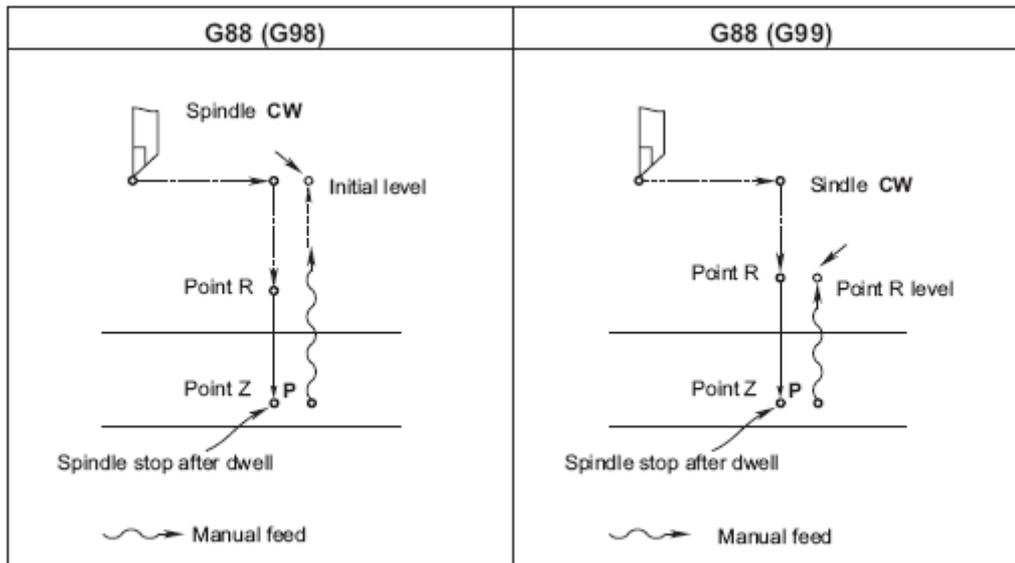
Z_ : The distance from point R to the bottom of the hole

R_ : The distance from the initial level to the point R level

P_ : Dwell time at the bottom of the hole

F_ : Cutting federate

K_ : *Number of repeats*



After positioning along **X** and **Y** axes, rapid traverse is performed to point R. Drilling is executed from point R to point **Z**.

When boring is completed, dwell is performed and then the spindle stops. The tool is returned manually (point Z) to point R. In point R the spindle is rotated clockwise and is returned to the initial position at rapid traverse.

Before specifying **G88** rotate the spindle using the auxiliary function (M code). When **G88** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in R point.

RESTRICTIONS:

Functions to Simplify Programming

◆ Drilling

Drilling is not performed in a block which do not contain **X, Y, Z, R** or other axes.

◆ R

Specify **R** in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ Cancel

Do not specify codes from group 01 (**G00 to G03**) and **G88** in one block. If they are used together, **G88** is cancelled.

Examples:

M3 S2000;	Cause the spindle to start rotating
G90 G99 G88 X300. Y-250. Z-150. R-100. P1000 F120.;	Positioning, drill hole 1, return to point R and dwell for 1 second at the bottom of the hole
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position
G80 G28 G91 X0 Y0 Z0;	Return to the reference position
M5;	Cause the spindle to stop rotating

13.1.12 Boring Cycle (G89)

This cycle is used to bore a hole.

Format:

G89 X_ Y_ Z_ R_ P_ F_ K_ ;

X_ Y_ : *Hole position data*

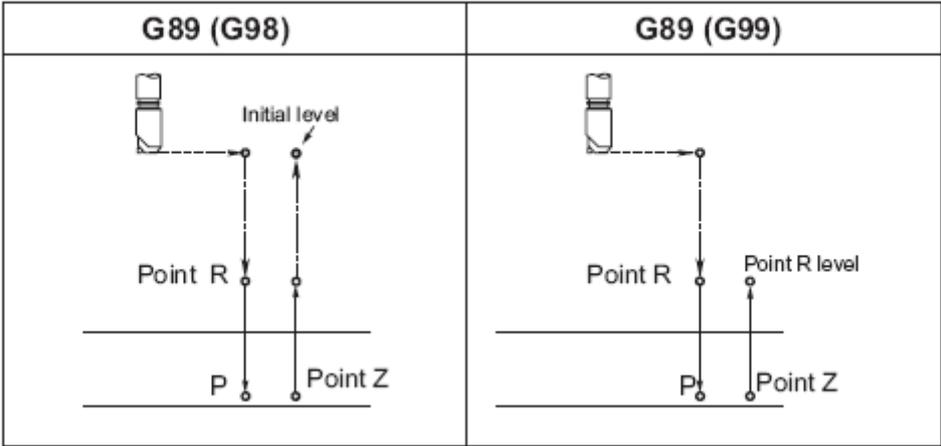
Z_ : *The distance from point R to the bottom of the hole*

R_ : *The distance from the initial level to the point R level*

P_ : *Dwell time at the bottom of the hole*

F_ : *Cutting feedrate*

K_ : *Number of repeats*



This cycle is almost the same as **G85**. The difference is that dwell is performed at the bottom of the hole here.

Before specifying **G89** rotate the spindle using the auxiliary function (**M** code). When **G88** and **M** code are specified in a block, the **M** code is executed during the first positioning operation. The system continues work with the next operation afterwards.

When **K** is used for the number of repeats, the **M** code is executed only for the

first hole; for the second and following ones it is not performed.

When a shift compensating the tool length (**G43**, **G44** or **G49**) is specified in the canned cycle, the shift is performed during positioning in R point.

RESTRICTIONS:

◆ Drilling

Drilling is not performed in a block which do not contain **X**, **Y**, **Z**, **R** or other axes.

◆ R

Specify R in blocks that perform drilling. If it is not set in a block performing drilling operation, it cannot be stored as a modal data.

◆ Cancel

Do not specify codes from group 01 (**G00** to **G03**) and **G89** in one block. If they are used together, **G89** is cancelled.

Examples:

M3 S100;	Cause the spindle to start rotating
G90 G99 G89 X300. Y-250. Z-150. R-120. P1000 F120.;	Positioning, drill hole 1, return to point R and dwell for 1 second at the bottom of the hole
Y-550.;	Positioning, drill hole 2 and return to point R
Y-750.;	Positioning, drill hole 3 and return to point R
X1000.;	Positioning, drill hole 4 and return to point R
Y-550.;	Positioning, drill hole 5 and return to point R
G98 Y-750.;	Positioning, drill hole 6 and return to the initial position

G80 G28 G91 X0 Y0 Z0; Return to the reference position

M5; Cause the spindle to stop rotating

13.1.13 Canned Cycle Cancel

The **G80** code cancels canned cycles.

Format:

G80;

All canned cycles are cancelled and return to normal operation is performed. The R and Z points are cleared. This means that in incremental mode R=0 and Z=0. All other drilling data is also cancelled (cleared).

Examples:

M3 S100; Cause the spindle to start rotating

G90 G99 G88 X300. Y-250. Z-150. R-120. F120.;
Positioning, drill hole 1 and return to point R

Y-550.; Positioning, drill hole 2 and return to point R

Y-750.; Positioning, drill hole 3 and return to point R

X1000.; Positioning, drill hole 4 and return to point R

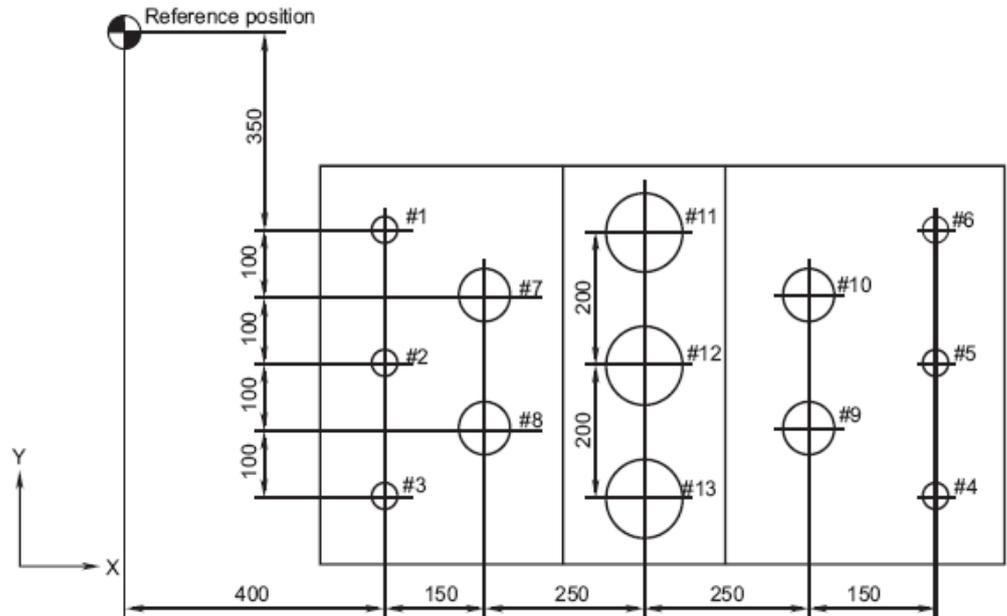
Y-550.; G98 Positioning, drill hole 5 and return to point R

Y-750.; Positioning, drill hole 6 and return to the initial position

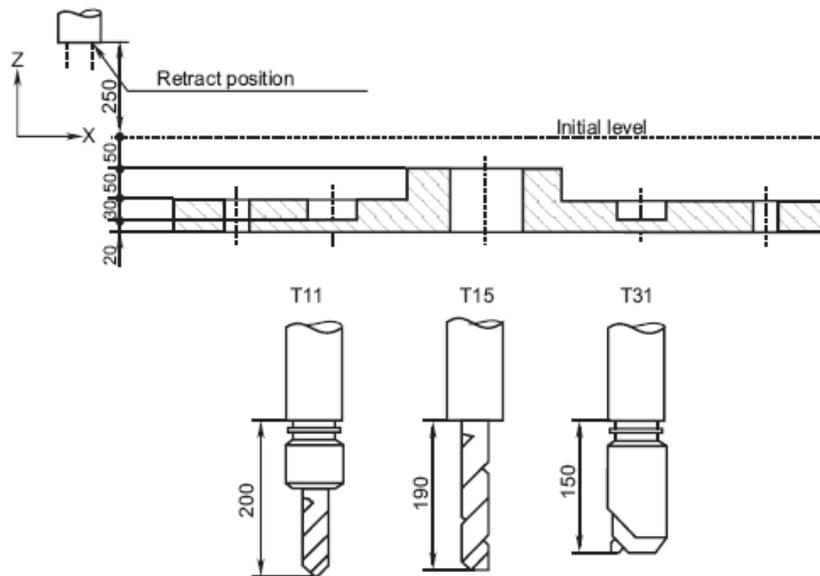
G80 G28 G91 X0 Y0 Z0; Return to the reference position and canned cycle cancel

M5; Cause the spindle to stop rotating

Program example using tool length offset and canned cycles



- # 1 ~ 6 --- Drilling of a 10mm diameter hole
- # 7 ~ 10 --- Drilling of a 20mm diameter hole
- #11~ 13 --- Boring of a 95mm diameter hole (depth 50mm)



Offset value +200.0 is set in offset No. 11
 +190. Is set in offset No.15
 And +150. Is set in offset No. 31

Functions to Simplify Programming

N001 G92X0Y0Z500.0;	Coordinate setting at reference position
N002 G90 G00 Z250.0 T11 M6;	Tool change
N003 G43 Z0 H11;	Initial level, tool length offset
N004 S30 M3;	Spindle start
N005 G99 G81 X400.0 R Y-350.0 Z-153.0 R-97.0 F120;	Positioning and #1 drilling
N006 Y-550.0;	Positioning, #2 drilling and return to point R level
N007 G98 Y-750.0;	Positioning, #3 drilling and return to point R level
N008 G99 X1200.0;	Positioning, #4 drilling and return to point R level
N009 Y-550.0;	Positioning, #5 drilling and return to point R level
N010 G98 Y-350.0;	Positioning, #3 drilling and return to initial level
N011 G00 X0 Y0 M5;	Reference point return, spindle stop
N012 G49 Z250.0 T15 M6;	Tool length offset cancel, tool change
N013 G43 Z0 H15;	Initial level, tool length offset
N014 S20 M3;	Spindle start
N015 G99 G82 X550.0 Y-450.0 Z-130.0 R-97.0 P300 F70;	Positioning, #7 drilling and return to point R level
N016 G98 Y-650.0;	Positioning, #8 drilling and return to point R level
N017 G99 X1050.0;	Positioning, #9 drilling and return to point R level
N018 G98 Y-450.0;	Positioning, #10 drilling and return to point R level
N019 G00 X0 Y0 M5;	Reference position return, spindle stop

N020 G49 Z250.0 T31 M6;	Tool length offset cancel, tool change
N021 G43 Z0 H31;	Initial level, tool length offset
N022 S10 M3;	Spindle start
N023 G85 G99 X800.0 Y-350.0 Z-153.0 R47.0 F50;	Positioning, #11 drilling and return to point R level
N024 G91 Y-200.0 K2;	Positioning, #12 and #13 drilling and return to point R level
N025 G28 X0 Y0 M5;	Reference position return, spindle stop
N026 G49 Z500.0;	Tool length offset cancel
N027 M0;	End of program

13.2 EXTERNAL MOVEMENT FUNCTION (G81)

After positioning has been completed, an external signal can be output to each block of the program, so that the machine can perform a specific operation.

For more details on the operation itself refer to the manual provided by the machine tool builder.

Format:

G81 IP_; (IP_ Movement along an axis command)

Each time the positioning for **IP_** is completed, **CNC** sends a signal to the machine. The external signal is sent for each positioning operation until cancelled with **G80** or **G** code from group 01.

RESTRICTIONS:

A signal for external operation is not sent after a block which does not contain **X** and **Y** coordinates.

The **G81** code is used as a **G** code in canned drilling cycle. Use the corresponding parameter to specify whether **G81** is used for the latter cycle or for a canned cycle.

14. COMPENSATION FUNCTION

GENERAL

This chapter describes the following compensation functions:

TOOL LENGTH OFFSET (G43, G44, G49) CUTTER COMPENSATION (G40 - G42)

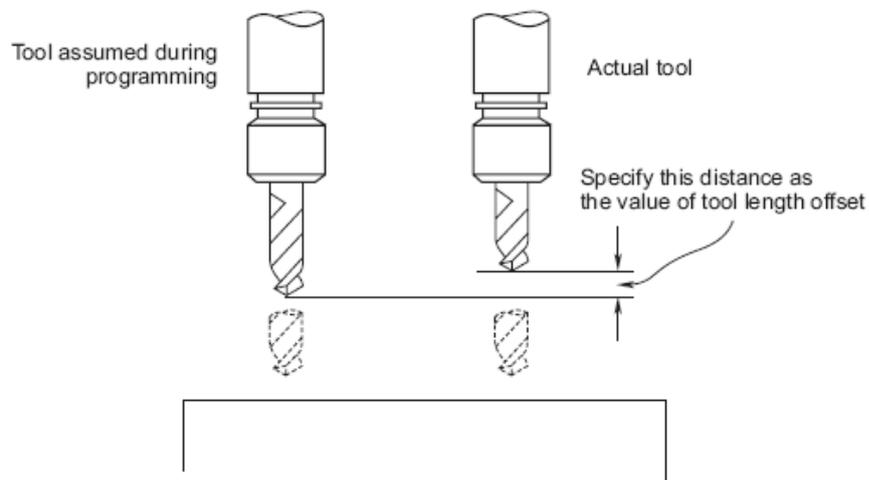
TOOL COMPENSATION VALUES, NUMBER OF COMPENSATION VALUES AND ENTERING VALUES FROM THE PROGRAM (G10)

14.1 TOOL LENGTH OFFSET

This function can be used for setting the difference between the tool length entered as a data in the program and its actual value. The value is stored in the offset memory. The difference can be compensated without the necessity of modifying the program.

The sign of the offset is specified with **G43** and **G44**. Choose a value for tool length compensation from the offset memory by entering the corresponding address and a number (**H code**).

Tool length compensation:



The following two methods for compensating the tool length can be used depending on the axis along which the corresponding compensation is made:

Tool length compensation **A**

Compensates the difference along **Z** axis.

Tool length compensation **B**

Compensates the difference along **X, Y** or **Z** axis.

Format:

Tool length offset A	G43 Z_H_ G44 Z_H_;	G43: Positive offset G44: Negative offset G17: XY plane selection G18: ZX plane selection G19: YZ plane selection G19: YZ plane selection H: Code for specifying tool length offset value
Tool length offset B	G17 G43 Z_H_ G17 G44 Z_H_ G18 G43 Y_H_ G18 G44 Y_H_ G19 G43 X_H_ G19 G44 X_H_;	
Tool length offset cancel	G49; or H0;	

EXPLANATIONS:

◆ **Selection of tool length offset**

The tool length compensation **A** or **B** is selected by setting the corresponding parameter.

◆ **Direction of the offset**

When **G43** is specified, the tool length compensation value (stored in the offset memory) specified with **H** code is added to the coordinates of the end position set by a command in the program. When **G44** is specified the same value is subtracted from the coordinates of the end position. The resulting coordinates specify the end position after the compensation regardless of the mode - absolute or incremental.

If no movement along the axes is specified, the system assumes such a situation a movement command without actual move.

Compensation Function

When a positive value is specified for tool length compensation with **G43**, the tool moves in the corresponding positive direction.

When a negative value is specified for tool length compensation with **G44**, the tool moves in the corresponding negative direction.

When a negative value is specified to the offset itself, the tool is moved in the opposite direction.

G43 and **G44** are modal codes. They are valid until another **G** code from the same group is reached.

◆ Specifying a tool length offset value

The tool length offset value which actually is a number (offset number) specified with **H** is loaded from the offset memory and is added or subtracted to the program command. This value can be set through TFT/MDI panel.

The range of the values which can be set as tool length compensation values is as follows:

Metric input	Inch input
0 to ±999.999mm	0 to ±99.9999inch

WARNING:

When the tool length compensation value changes due to a change in the number specifying the offset, the offset value changes to a new compensation value and the new value is not added to the old one.

H1 : the compensation value is 20.0

H2 : the compensation value is 30.0

G90 G43 Z100.0 H1; Z will move to 120.0

G90 G43 Z100.0 H2; Z will move to 130.0

Note:

The tool length compensation value in offset No0 is 0, i.e. **H0** always means 0. It is not possible to specify another compensation value by **H0**.

◆ Tool length compensation along two or more axes

The offset **B** can be performed along two or more axes when the axes are in two or more blocks.

Offset in X and Y axes

G19 G43 H_; Offset in **X** axis

G18 G43 H_; Offset in **Y** axis
(Offset in **X** and **Y** axes is performed)

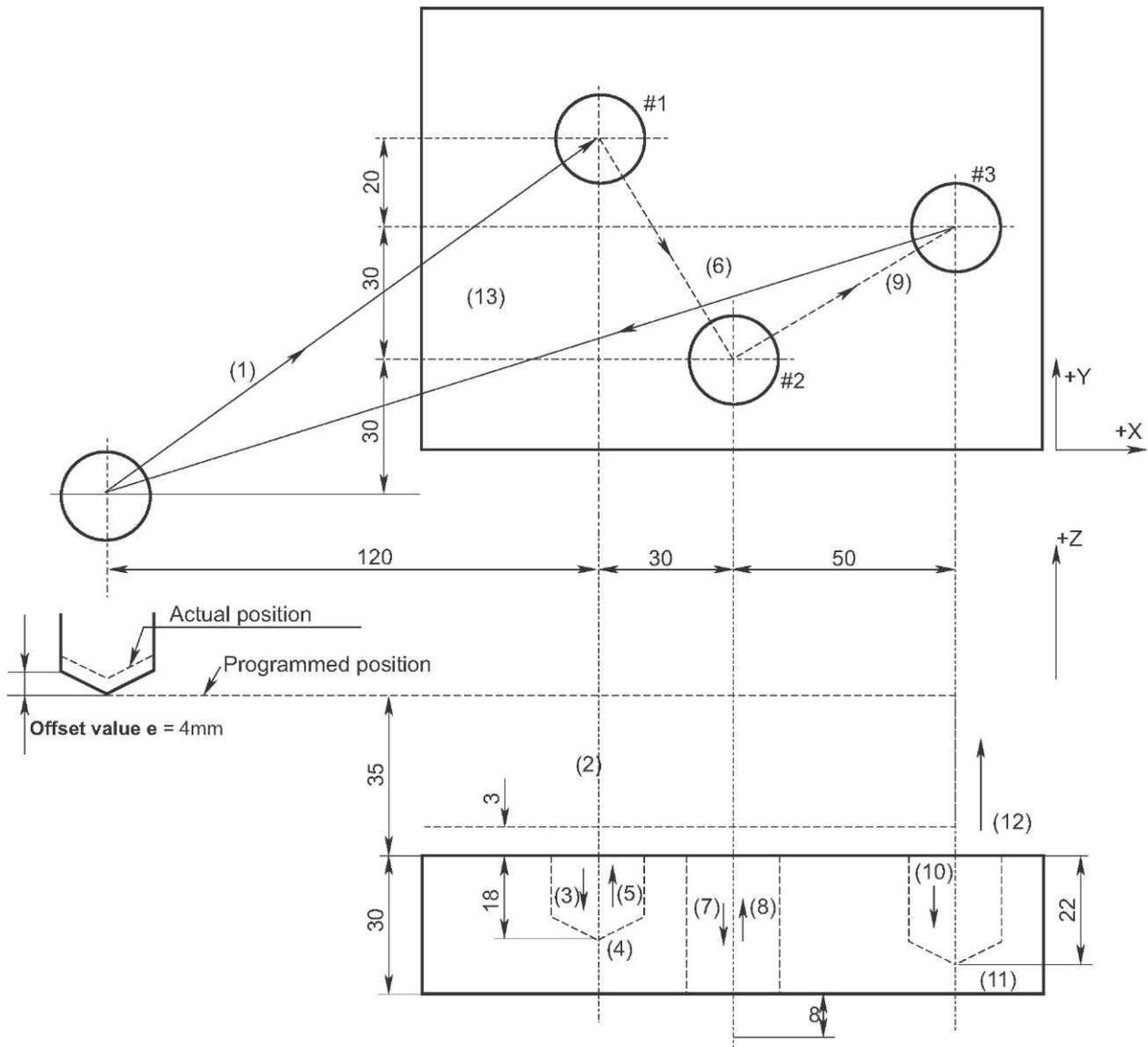
◆ Tool length compensation cancel

To cancel the tool length compensation specify **G49** or **H0**. After **G49** or **H0** the system cancels the compensation mode automatically.

WARNING:

After compensation **B** along two or more axes is performed, the offset along all axes is canceled by specifying **G49**. **H0** code cancels the offset only along the axis perpendicular to the chosen plane.

Compensation Function



Program:

```

H1=-4.0 (Tool length compensation value)

N1 G91 G00 X120.0 Y80.0; (1)

N2 G43 Z-32.0 H1; (2)

N3 G01 Z-21..0 F1000; (3)

N4 G04 P2000; (4)
  
```

N5 G00 Z21.0;	(5)
N6 X30.0 Y-50.0;	(6)
N7 G01 Z-41.0;	(7)
N8 G00 Z41.0;	(8)
N9 X50.0 Y30.0;	(9)
N10 G01 Z-25.0;	(10)
N11 G04 P2000;	(11)
N12 G00 Z57.0 H0;	(12)
N13 X-200.0 Y-60.0;	(13)
N14 M2;	

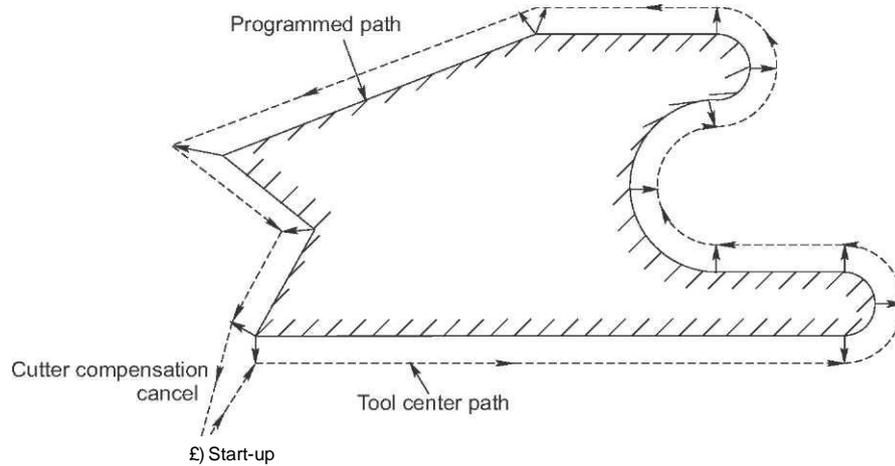
14.2 OVERVIEW OF CUTTER COMPENSATION

When the tool moves, the tool path can be shifted by its radius. To make an offset as large as the tool radius, **CNC** first creates an offset vector with length equal to the tool radius. The offset vector is perpendicular to the tool path. The tail of the vector is on the workpiece side and the head positions to the center of the tool.

If a linear or circular interpolation is commanded after the initial position, the tool path can be shifted by the length of the offset vector during machining.

To return the tool in the initial position at the end of machining, cancel the compensation mode. Tool path in cutter compensation:

Compensation Function



In cutter compensation the proper offset of tool path can be obtained both for outer and inner corners just by specifying the offset direction. Unlike cutter compensation **B**, the programs can be created easily without having to consider the difference between inner and outer corners.

Format:

◆ Start up (Tool compensation start)

G00 (or G01) G41 (or G42) IP_ H_;

G41: Cutter compensation left (Group 07)

G42: Cutter compensation right (Group 07)

IP_: Axis movement command

H_: Code for specifying the cutter compensation value

◆ Cutter compensation cancel (offset mode cancel)

G40 IP_;

G40: Cutter compensation cancel (Group 07) (Offset mode cancel)

IP_: Axis movement command

◆ **Offset plane selection**

Offset plane	Plane selection command	IP_
XY	G17 ;	X_Y_
ZX	G18 ;	Z_X_
YZ	G19 ;	Y_Z_

◆ **Offset cancel mode**

At the beginning when the power is switched on, the cutter compensation mode is canceled. In cancel mode the offset vector is always 0 and the tool center path coincides the programmed path.

◆ **Start-up**

When a compensation command (non zero value for offset in the plane and a **H** code different from **H0**) is issued in offset cancel mode, **CNC** enters in offset mode. The tool shift with this command is called start-up.

Specify positioning (**G00**) or linear interpolation (**G01**) for start-up. If circular interpolation (**G02** or **G03**) is specified, an alarm is displayed.

When the start and next blocks are processed, **CNC** prereads two blocks ahead. The second one is not displayed on the screen.

◆ **Offset mode**

A compensation is performed when positioning (**G00**), in linear interpolation (**G01**) or in circular interpolation (**G02, G03**) in offset mode. If two or more blocks which do not move the tool (miscellaneous function, dwell, etc.) are specified in offset mode, the tool will make either insufficient or excessive cut. If the offset plane is included in offset mode, an alarm is displayed and the tool stops.

◆ **Offset mode cancel**

In offset mode when a block satisfying one of the following conditions is executed, the equipment enters the offset cancel mode and the action of this block is called offset cancel.

1. **G40** is commanded

2. The offset cutter compensation value specified is 0.

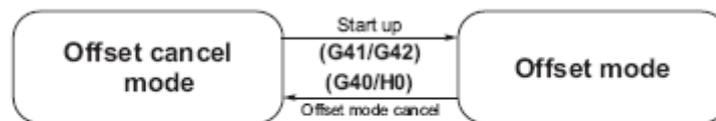
When offset cancel is performed, the arc commands (**G02** and **G03**) are not valid. If an arc is commanded, an alarm is displayed and the tool stops.

Compensation Function

In offset mode cancel, **CNC** control executes the instructions in that block and in the block found in the tool cutter compensation buffer. Meanwhile, in the case of a single block mode, the control executes it and stops. After pushing the cycle start button, the block is executed without reading the next one.

In that case the control is in cancel mode and usually the block which will be executed afterwards will be stored in a buffer register and the next block will not be read in the cutter compensation buffer.

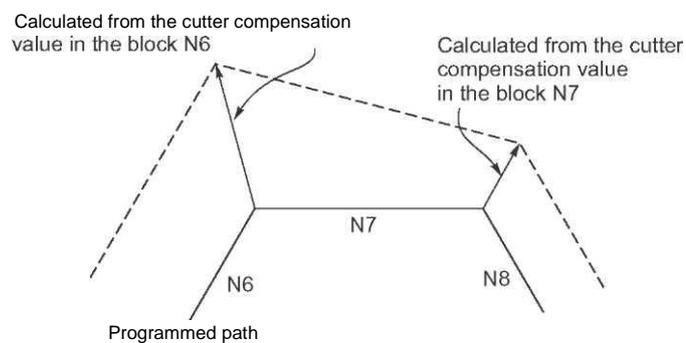
Changing the offset mode:



◆ Change of the cutter compensation value

In general, the cutter compensation value will be changed in cancel mode, when the tool is to be changed. If the cutter compensation value is changed in offset mode, the vector at the end point of the block is calculated for the new cutter compensation value.

Changing the cutter compensation value:



◆ **Positive/negative cutter compensation value and tool center path**

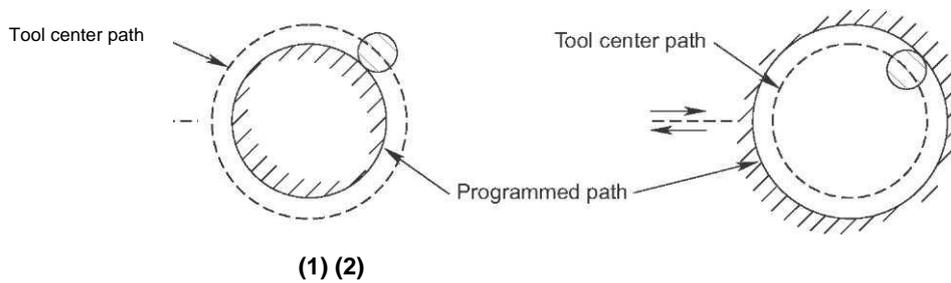
If the compensation is negative, distribution is made for a figure in which the codes **G42** and **G43** can be replaced in the program. Therefore, if the tool center passes outside the workpiece, it can pass the inside and vice versa.

The figure below shows an example. Normally, the compensation is positive. When the tool path is specified as in **(1)**, if the compensation is set negative, the tool center moves as in **(2)** and vice versa.

Consequently, one and the same program allows cutting outer and inner surfaces. The gap between them can be adjusted by the sign of the compensation. This is applicable if the start-up and the cancel are of A type.

Tool center paths when positive and negative cutter compensation values are specified:

◆ **Cutter compensation values setting**



The cutter compensation values are set with **H** codes on the TFT/MDI panel. The table below shows the range in which cutter compensation values can be set:

Metric input	Inch input
0 to ±999.999mm	0 to ±99.9999inch

Note:

1. The cutter compensation value specified with **H0** corresponds to **0**, i.e. **H0** always mean **0**. It is not possible to specify **H0** with another offset.

◆ **Offset vector**

The offset vector is a two-dimensional vector which is equal to the compensation value specified in H code. It is calculated in the control unit and its direction is updated in accordance of tool movement along each block.

The offset vector is deleted by reset.

◆ **Specifying a cutter compensation value**

The tool cutter compensation is specified by a number. The number consists from one to two digits following the address (**H** code). The **H** code is valid until another **H** code is specified. The **H** code is used both for tool offset and compensation value.

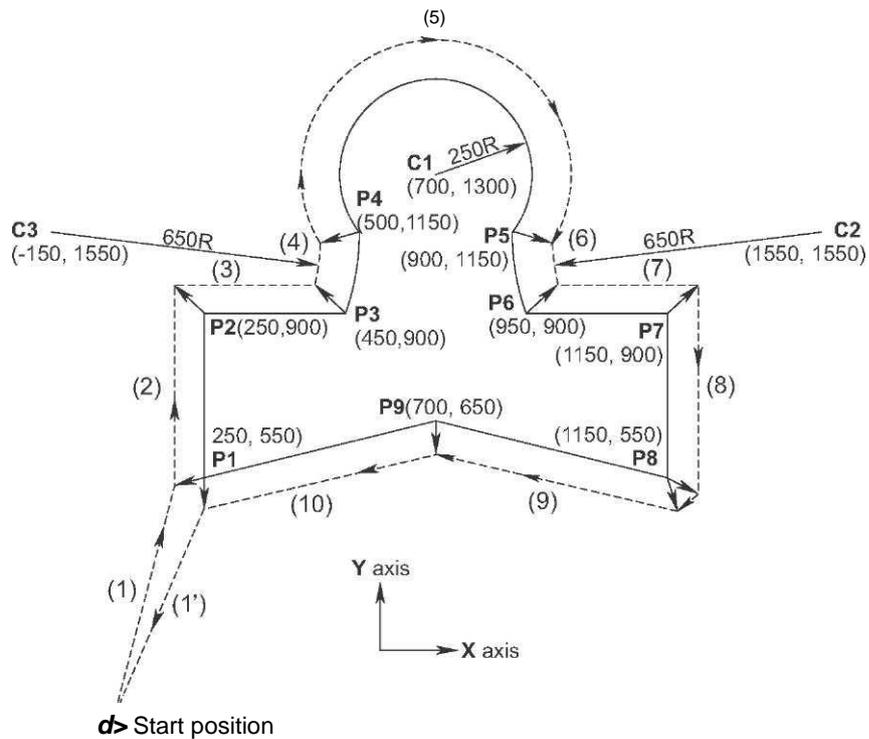
When the **H** code for the compensation is specified during tool length compensation, the length compensation stays unchanged. When **H** code for tool length compensation is specified during tool cutter compensation, both the cutter compensation and the tool length compensation are changed. Do not change tool length compensation in cutter compensation mode.

◆ **Plane selection and vector**

Offset calculation is performed in the plane specified by **G17**, **G18** and **G19** (**G** codes for plane selection). This plane is called offset plane. The compensation is not performed for coordinates of a position which is not in the specified plane. In simultaneous three axes control, the tool path projected on the offset plane is compensated.

The offset plane is changed during offset cancel mode. If it is executed in offset mode, an alarm is displayed and the machine stops.

Examples:



G92 X0 Y0 Z0;	Specifies absolute coordinates. The tool is positioned at the start position (X0, Y0, Z0)
N1 G90 G117 G00 G41 H07 X250.0 Y550.0;	Starts cutter compensation. The tool is moved to the left of the programmed path at a distance specified in H07 . In other words, the tool path is shifted by its radius (offset mode), because the value in H07 is 15 (the tool radius is 15 mm).
N2 G01 Y900.0 F150;	Specifies machining from P1 to P2.
N3 X450.0;	Specifies machining from P2 to P3.
N4 G03 X500.0 Y1150.0 R650.0;	Specifies machining from P3 to P4.
N5 G02 X900 R-250.0;	Specifies machining from P4 to P5.
N6 G03 X950.0 Y900.0 R650.0;	Specifies machining from P5 to P6.
N7 G01 X1150.0;	Specifies machining from P6 to P7.
N8 Y550.0;	Specifies machining from P7 to P8.
N9 X700.0 Y650.0;	Specifies machining from P8 to P9.
N10 X250.0 Y550.0;	Specifies machining from P9 to P1.
N11 G00 G40 X0 Y0;	Cancels the offset mode. The tool is returned to the start position (X0, Y0, Z0).

Compensation Function

14.3 DETAILS ON CUTTER COMPENSATION

This chapter provides a detailed explanation for the tool movement when cutter compensation is active.

The chapter consists of the following sections:

GENERAL

TOOL MOVEMENT IN START-UP

TOOL MOVEMENT IN OFFSET MODE

TOOL MOVEMENT IN OFFSET MODE CANCEL

INTERFERENCE CHECK

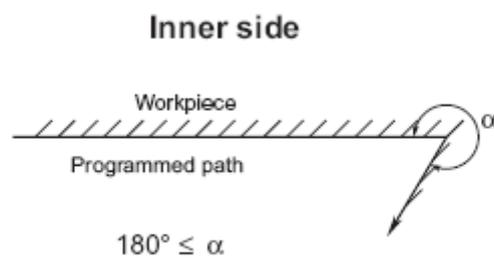
OVERCUTTING IN CUTTER COMPENSATION

INPUT COMMAND FROM MDI

14.3.1 General

Inner and outer side

When the intersection angle created by the tool path with two subsequent move commands is over 180 degrees, it is referred as "**inner side**".



When the angle is between 0 and 180 degrees, it is referred as "**outer side**".

◆ Meaning of symbols

The following symbols are used in the subsequent figures:

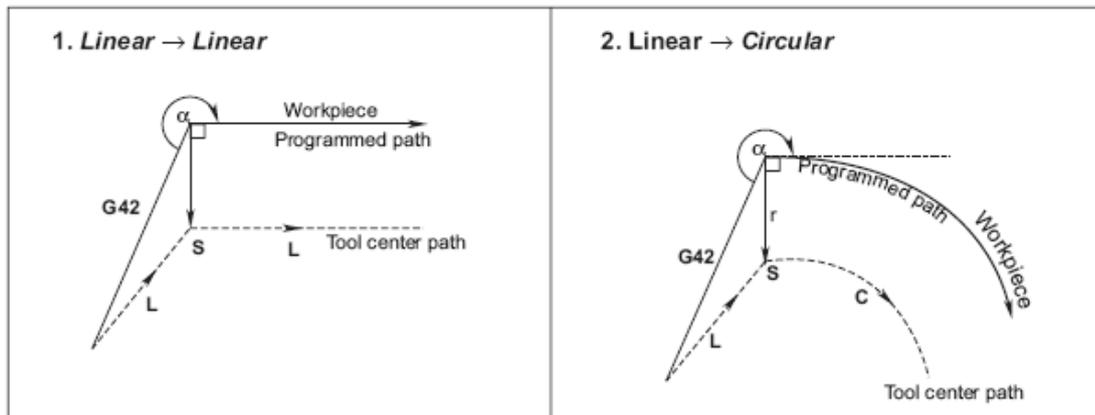
S - indicates a position at which a single block is executed once

- SS** - indicates a position at which a single block is executed twice
 - SSS** - indicates a position at which a single block is executed three times
 - L** - indicates that the tool moves along a straight line
 - C** - indicates that the tool moves along an arc
 - r** - indicates the cutter compensation value
- The **intersection** is the position at which two program paths of two blocks intersect, after they have been compensated with **r**.
 - The **circle** indicates the tool center

14.3.2 Tool Movement in Start-up

When an offset cancel mode is changed to offset mode, the tool moves as shown below:

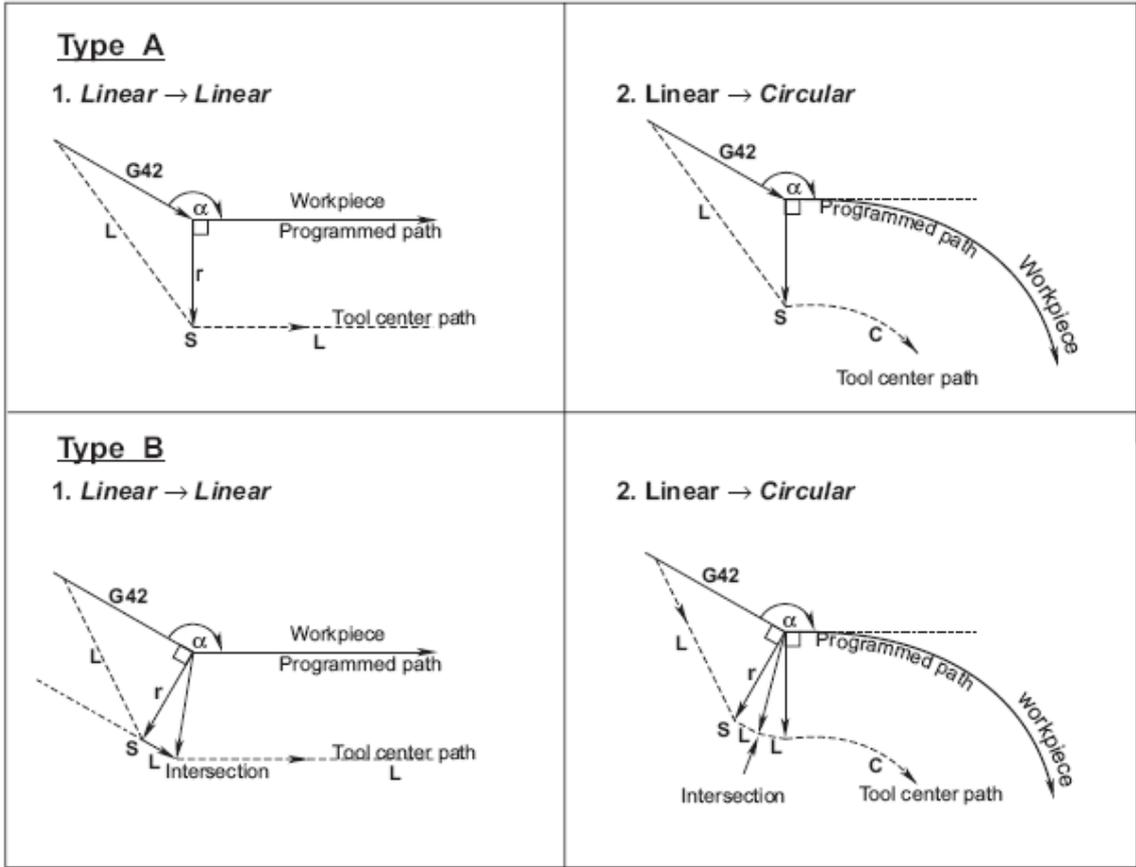
◆ Tool movement around the inner side of a corner



◆ Tool movement around the outside of a corner in the range of 90 to 180 degrees

In start-up the tool path has two alternatives - type **A** and type **B** which are chosen by parameter.

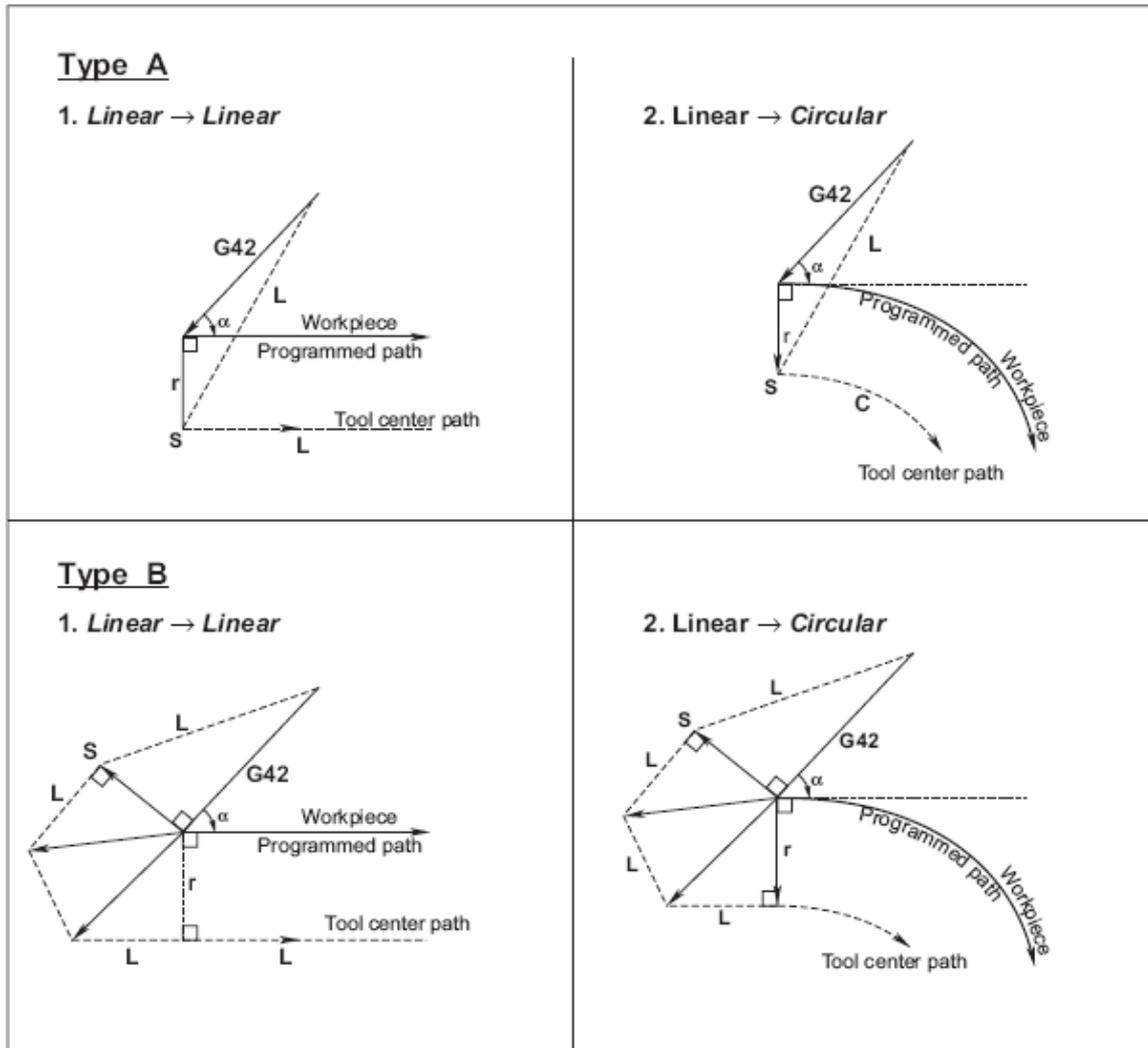
Compensation Function



The tool moves around an obtuse angle ($90 \text{ degrees} < \alpha < 180 \text{ degrees}$). The tool path may be type **A** or type **B** which are chose by parameter SUPM (No 0016).

◆ Tool movement around the outside of an acute angle

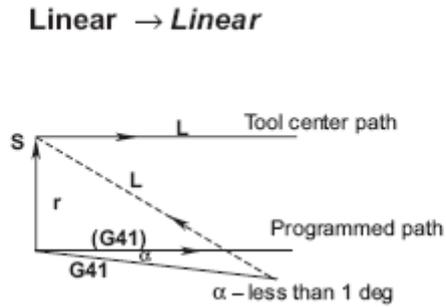
In start-up the tool path has two alternatives - type **A** and type **B** which are chosen by parameter.



The tool moves around the outside of an acute angle ($\alpha < 90$ degrees).

Compensation Function

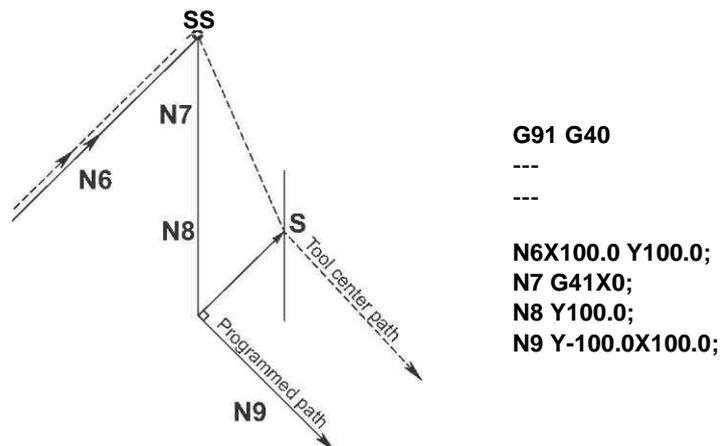
- ◆ Tool movement **linear @ linear** around the outside of an acute angle less than 1 degree



The tool moves around an acute angle less than 1 degree, performs **linear@linear** interpolation and the compensation is as follows:

- ◆ **A block without tool movement specified at start-up**

If the command is specified at the start-up, the offset vector is not created.



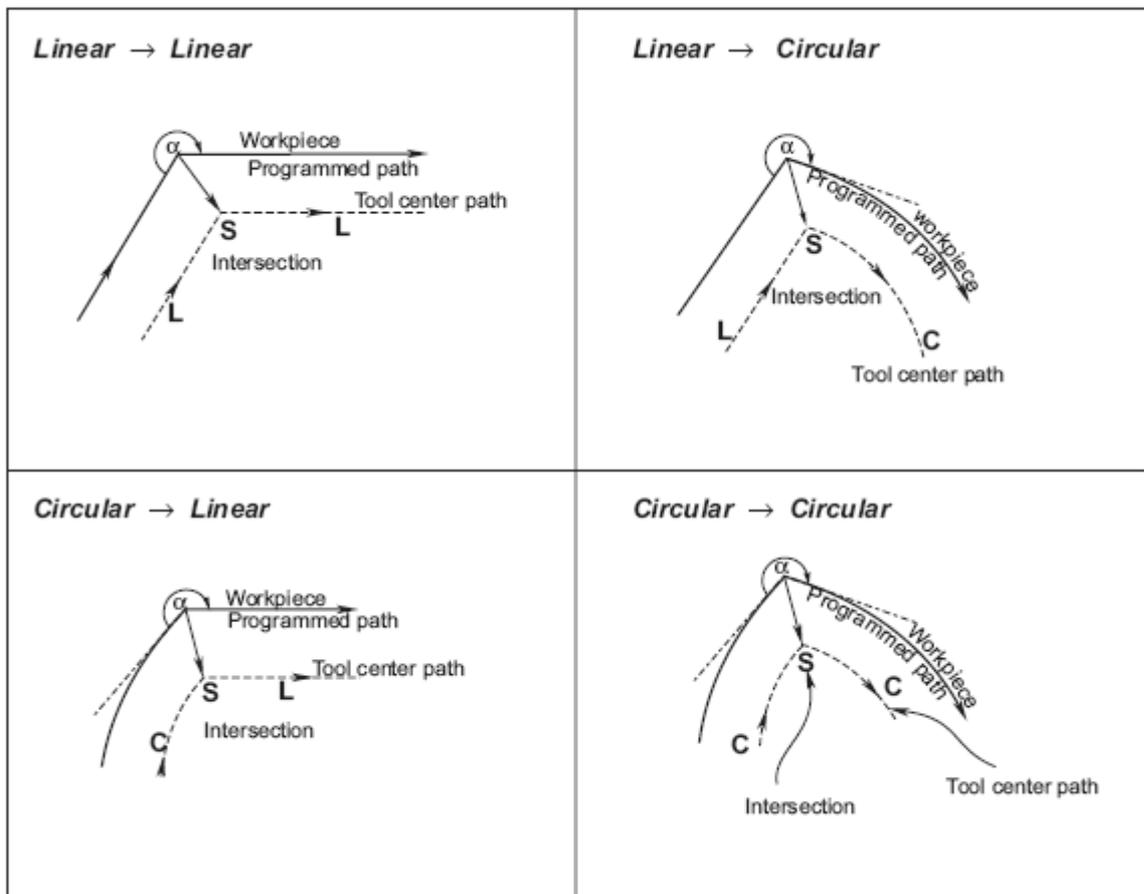
Note:

For the definition of blocks that do not move the tool see the next chapter.

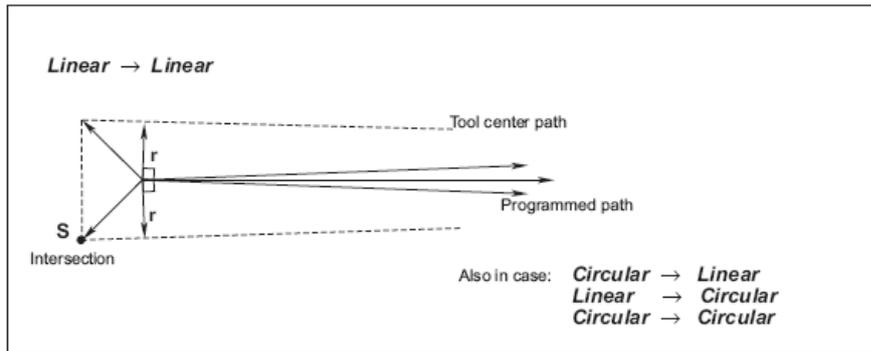
14.3.3 Tool Movement in Offset Mode

In offset mode the tool moves as shown below:

- ◆ **Tool movement around the inside of a corner**
($\alpha < 180$ degree)



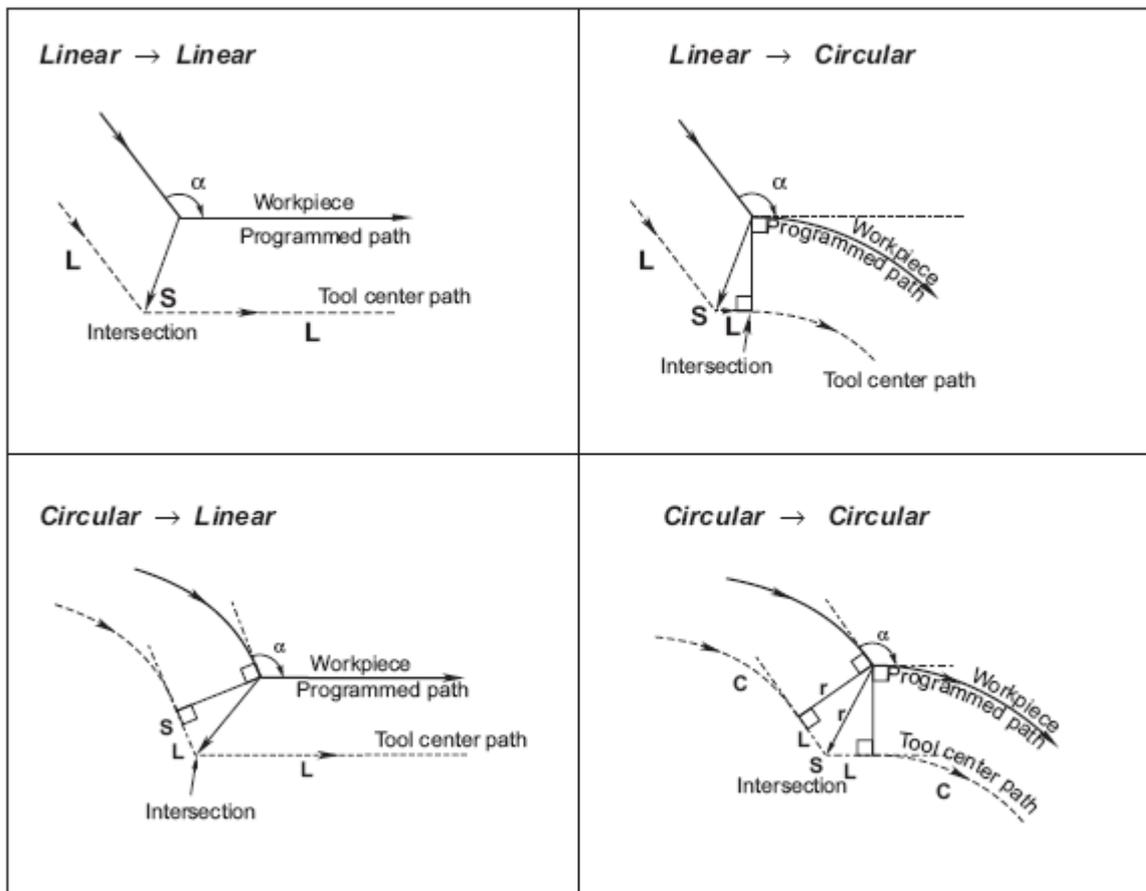
- ◆ **Tool movement linear to linear around the inside of an angle less than 1 degree with abnormally long vector**



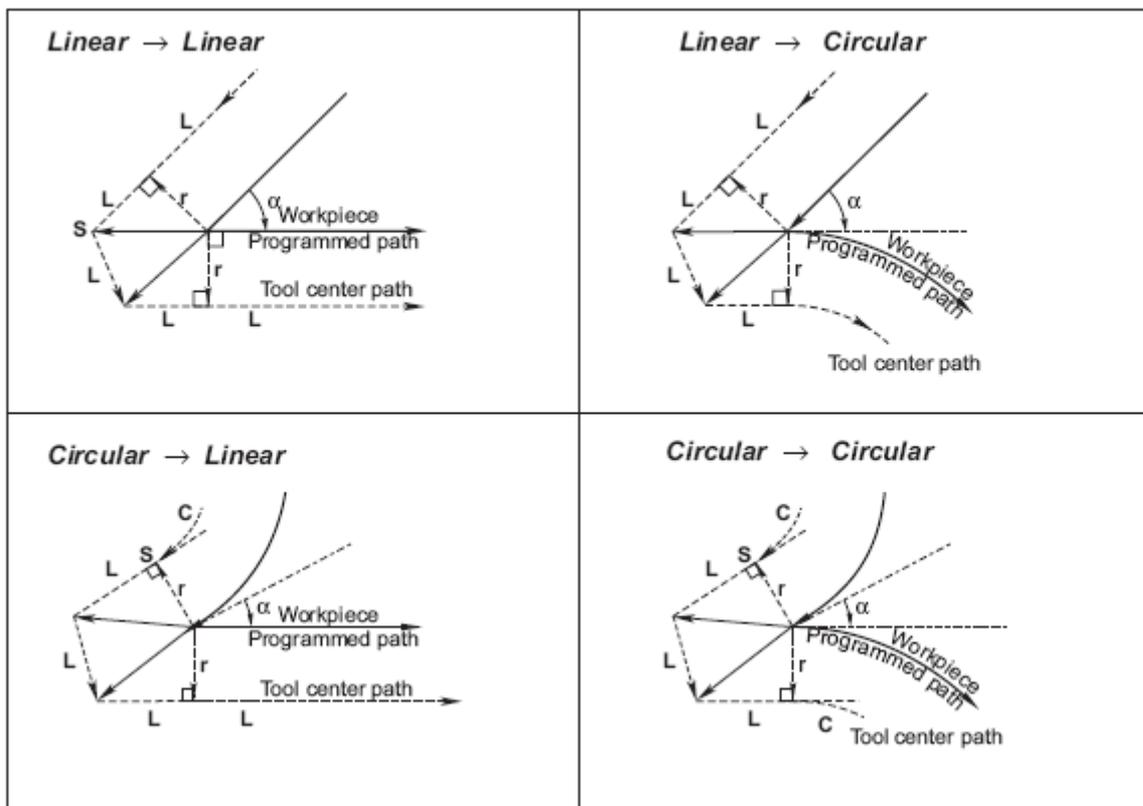
When machining the inner side of an angle less than 1 degree, the value of the compensation vector is increased.

The same procedure should be inferred in the case of intersection from a straight line to an arc, from an arc to a straight line, and from an arc to an arc.

◆ **Tool movement around the outside of a corner in the range from 90 to 180 degrees**



◆ Tool movement around the outside of an acute angle < 90 degrees

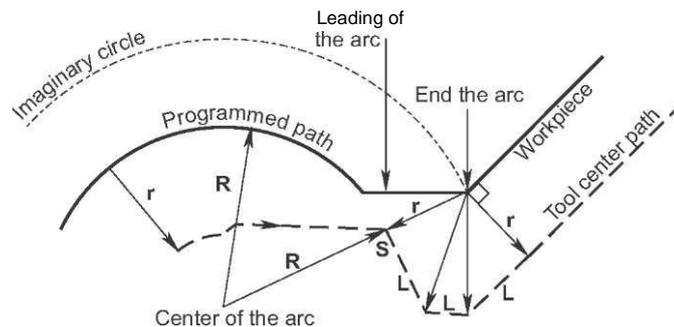


EXCEPTIONAL MOVEMENT

◆ The end position of the arc is not on the arc

If the end of a line, leading to an arc, is programmed as an arc end by mistake, as shown below, the system assumes that cutter compensation has been executed with respect to an imaginary circle that has the same center as the arc and passes the specified end position. Based on this assumption the system creates the vector and carries out the compensation. The resulting tool center path is different than that created by applying cutter compensation to the programmed path in which the line leading to the arc is considered straight.

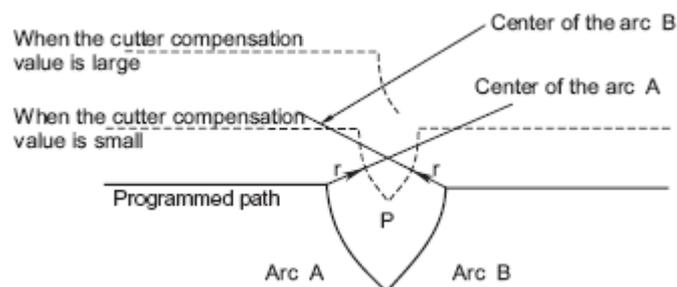
Compensation Function



The same description is valid for tool movement from one arc to another.

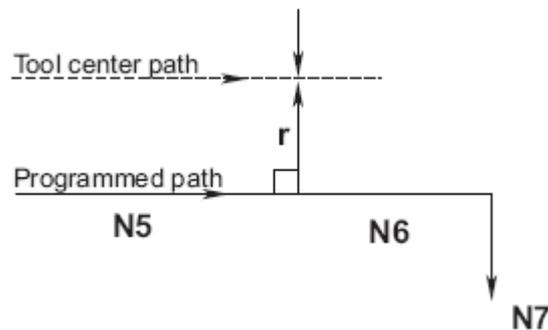
◆ There is no inner intersection

If the tool cutter compensation value is sufficiently small, the two circular tool center paths calculated after the compensation intersect at position **P**. Intersection **P** may not occur if an excessively large value for cutter compensation is specified. When such a situation is predicted, an alarm is displayed at the end of the previous block and the tool is stopped. In the example shown below, tool center paths along arcs **A** and **B** intersect in **P** when a very small cutter compensation value is specified. If an excessively large value is specified, this intersection does not occur.



◆ The center of the arc coincides the start position or the end position

If the center of the arc coincides the start position or the end position, an alarm is displayed and the tool stops at the end position of the previous block.



```
(G41)
N5 G01 X100.0;
N6 G02 X100.0 I0 J0;
N7 G03 Y-100.0 J-100.0;
```

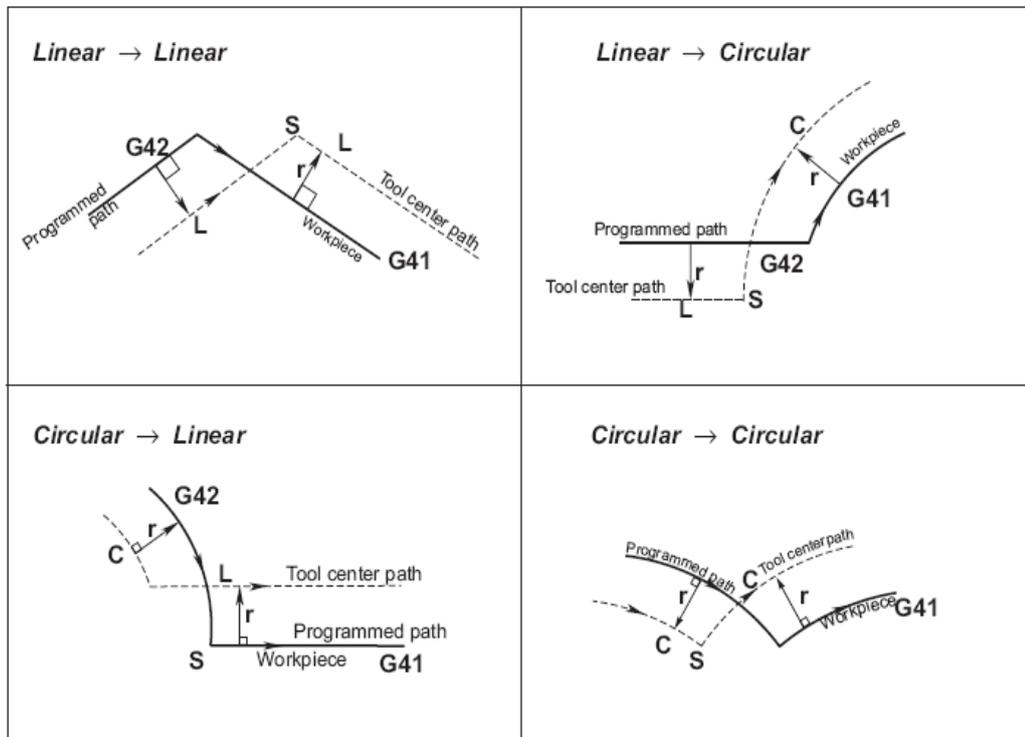
◆ Changing the offset direction in offset mode

The offset direction is specified with **G** code (**G41** and **G42**) for the radius and the sign of the compensation as follows:

G code	Sign of offset amount	
	+	-
G41	Left side offset	Right side offset
G42	Right side offset	Left side offset

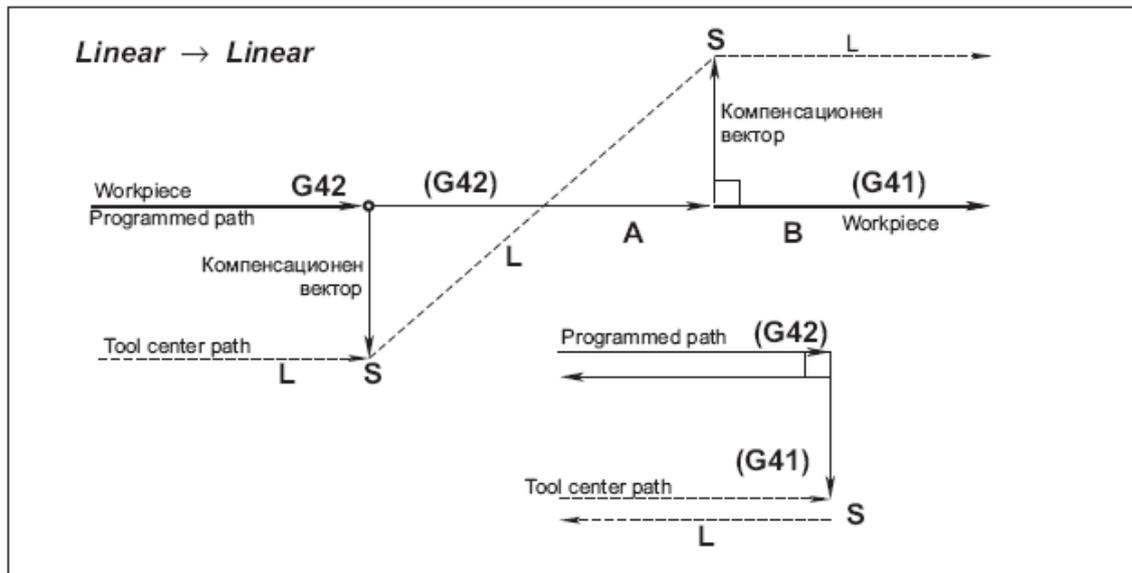
The offset direction can be changed in offset mode. If the offset direction is changed in the block, the vector is created at the intersection of the tool center path of that block and the tool center path of the previous one. Such a change, however, cannot be applied to the first block and the block immediately following it.

◆ Tool center path in an intersection

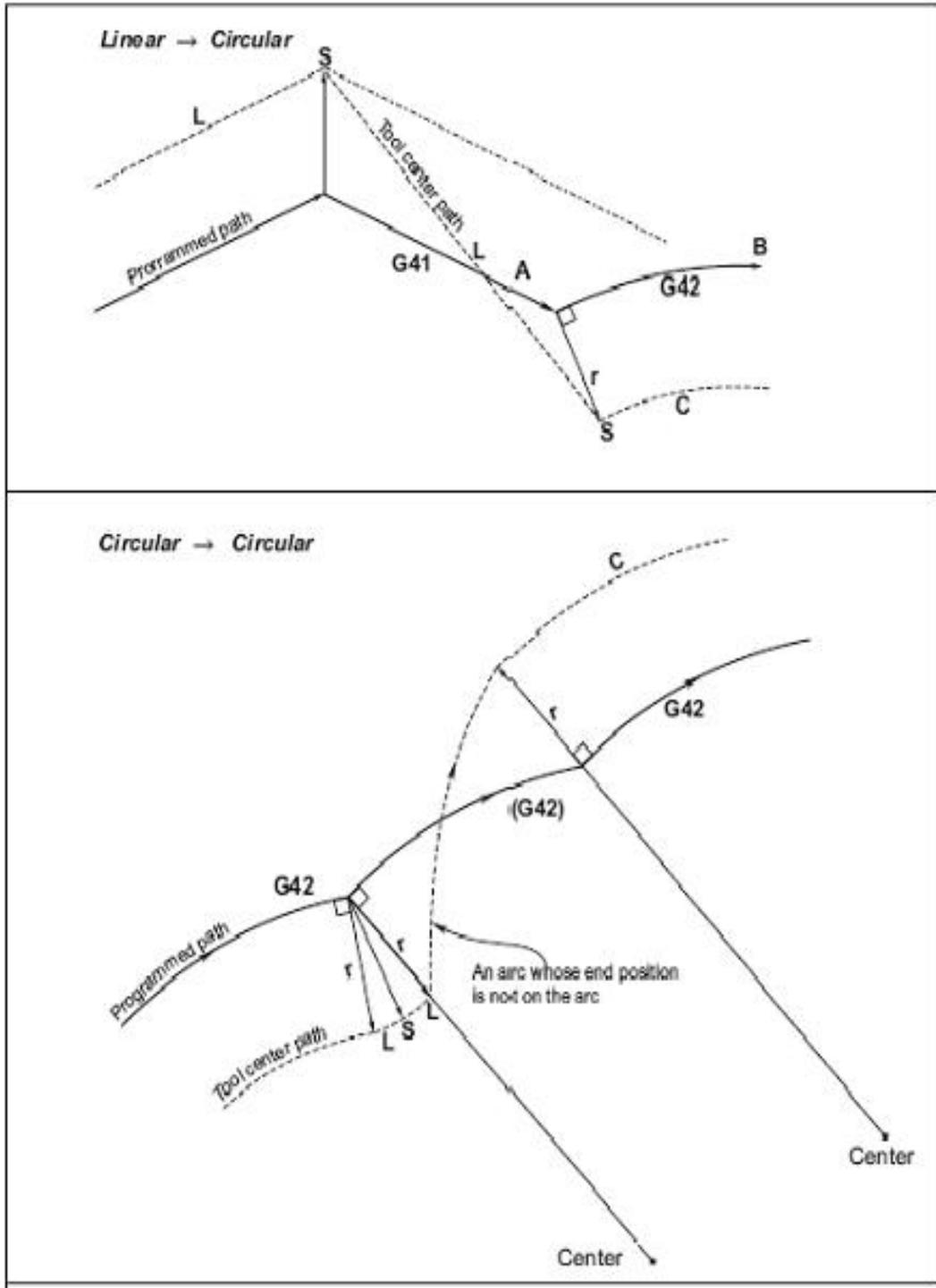


◆ Tool center path without an intersection

When the offset direction is changed from block **A** to block **B** by **G41** and **G42**, intersection with the offset path is not required, the vector which is normal for block **B** is created at the beginning of block **B**.



Compensation Function

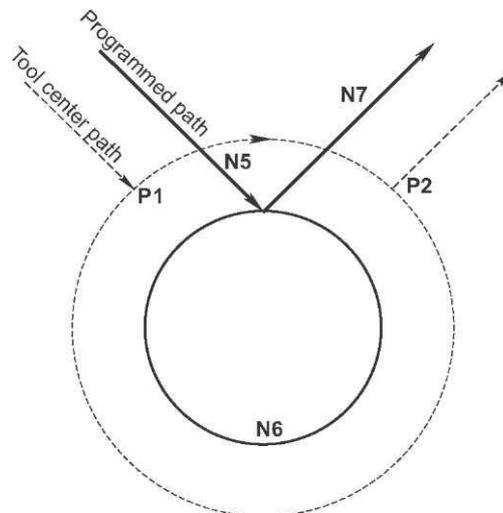


◆ **The length of tool center path larger than the circumference of a circle**

Normally, there is almost no possibility of generating such a situation. However, when **G41** and **G42** are changed or when **G40** is commanded with addresses **I**, **J** and **K**, this situation may occur.

In this case, shown on the figure, the tool cutter compensation is not performed for more than one circumference of the circle.

The arc is formed from **P1** to **P2** as shown below. Depending on the situation, an alarm may be displayed due to the "**Interference check**" described later. To make a circle with more than one circumference, it has to be specified in segments.



```
(G42)
N5 G01 G91 X50.0 Y-70.0; N6
G41 G02 J-50.0
N7 G42 G01 X50.0 Y70.0;
```

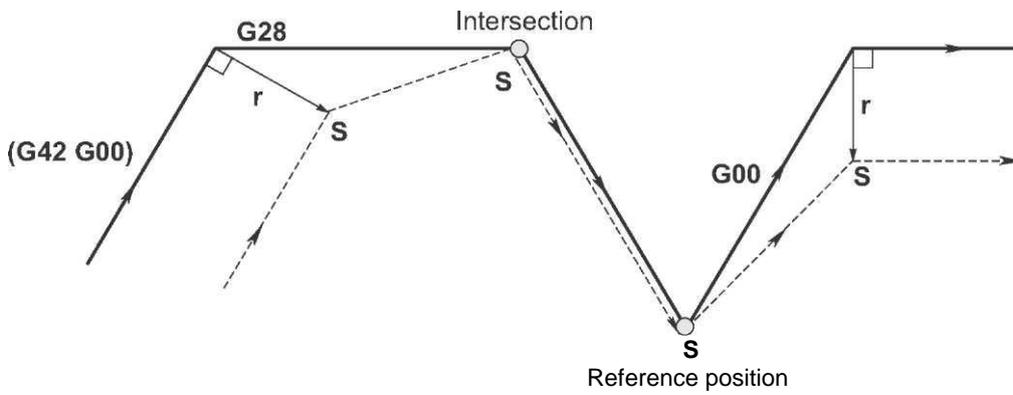
◆ **Temporary cutter compensation cancel**

If the following commands are specified in offset mode, the offset mode is canceled temporarily and then automatically restored. The offset mode can be canceled and restored as described in the corresponding chapters.

Compensation Function

Specifying **G28** (automatic return to the reference position) in offset mode

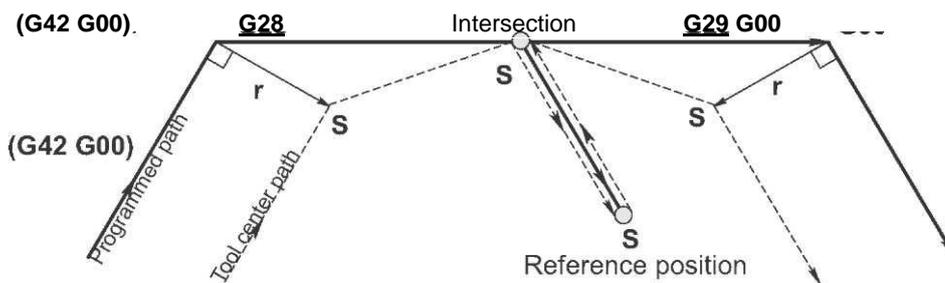
If **G28** is specified in offset mode, it is canceled in the intermediate position. If the vector remains after the tool has returned to the reference position, its components are zeroed for each axis, along which reference position return has been performed.

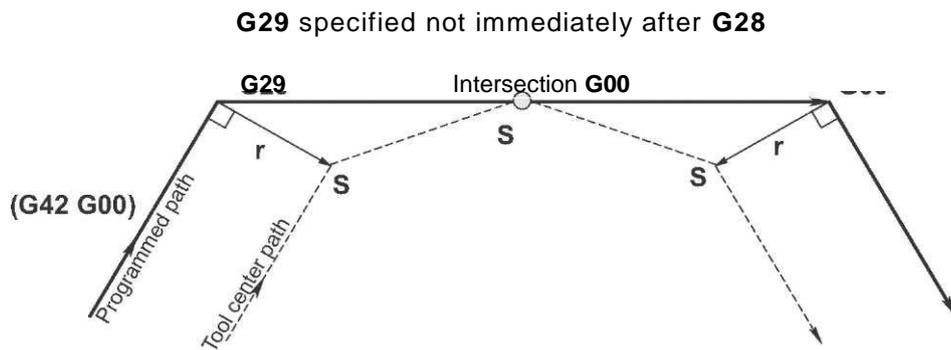


Specifying **G29** (automatic return from the reference position) in offset mode

If **G29** is specified in offset mode, it is canceled in the intermediate position and the offset mode is restored automatically in the next block.

G29 specified immediately after **G28**

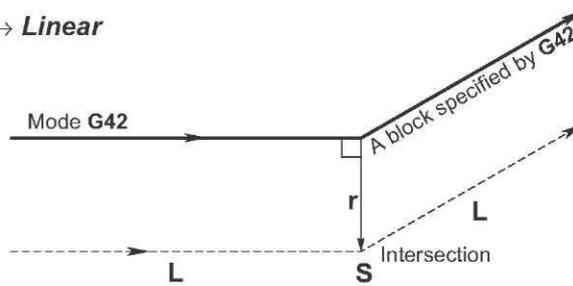




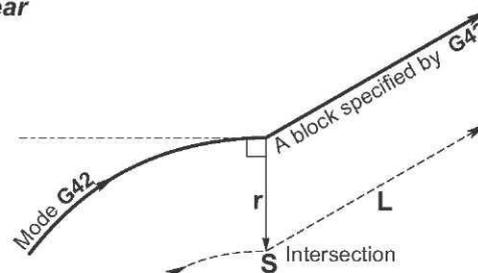
◆ **Cutter compensation G code in offset mode**

The offset vector can be set in such a way that it will form a right angle to the moving direction, specified in the previous block, irrespective of machining inner or outer side, by commanding cutter compensation **G** code (**G41** or **G42**) in offset mode. If this code is specified in an arc command, there will be not obtained a correct circular motion.

Linear → Linear



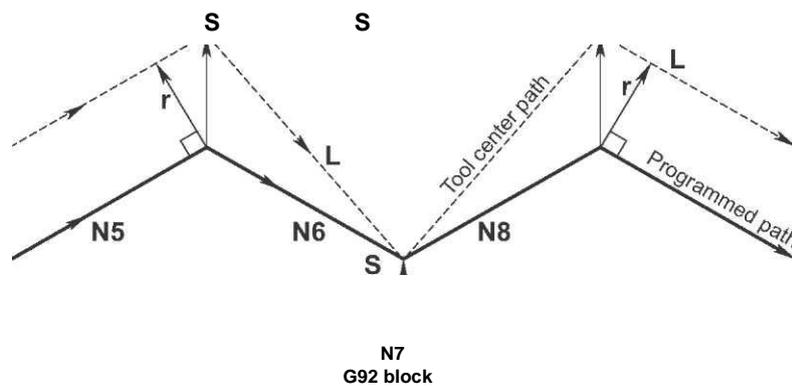
Circular → Linear



Compensation Function

◆ G92 command in offset mode

If **G92** is commanded in offset mode (absolute zero point programming), the offset vector is temporarily canceled. Then the offset mode is automatically restored. In this case, without disabling the offset cancel, the tool is directly moved to the specified position where the offset vector is canceled. Also when the offset mode is restored, the tool is moved directly to the intersection point.



(G41)

```
N5 G91 G01 X70.0 Y30.0; N6  
X60.0 Y-30.0;  
N7 G92 X20.0 Y10.0;  
N8 G90 G01 X80.0 Y40.0;
```

◆ A block without tool movement

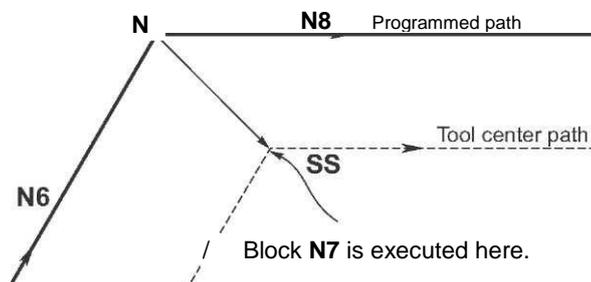
The blocks following do not move the tool. In these blocks, the tool will not be moved even if tool cutter compensation is applied.:

```
M05; M code output  
S21; S code output G04  
X10.0; Dwell G10 P01  
R10.0; (G17) Z200.0;  
G90; G91 X0;
```

Cutter compensation value setting The move command is not included in the offset plane **G** code only Move distance is zero

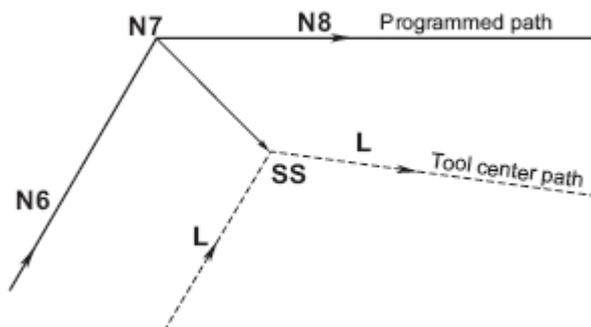
A block without tool movement specified in offset mode

When a block without tool movement is reached in cutter compensation mode, the vector and the tool center path remain the same. This block is executed in the stop point.



```
N6 G91 X100.0 Y100.0;  
N7 G04 X10.0; N8  
X100.0;
```

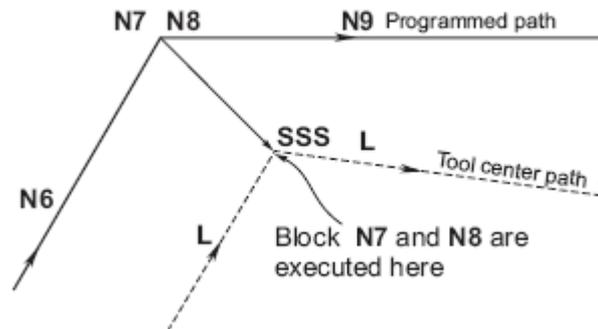
When, however, the move distance is zero even when the block is specified as single, the tool movement is the same as that when more than one block is executed without tool movement which will be described subsequently.



```
N6 G91 X100.0 Y100.0;  
N7 X0; N8  
X100.0;
```

Two blocks without tool movement should not be specified consecutively. If such a situation occurs, a vector is created with length equal to the offset value and direction which is the same as that in the previous block. So overcutting may result.

Compensation Function

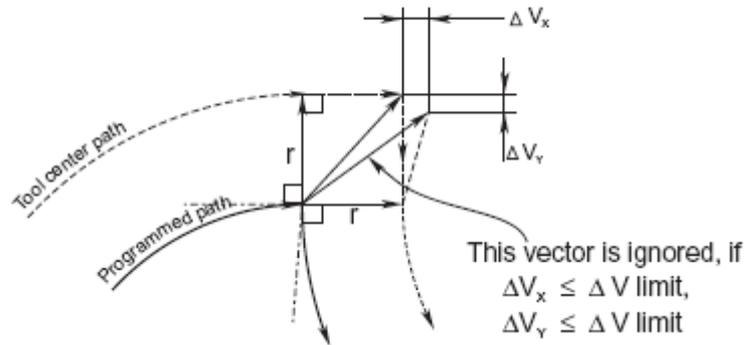


```
N6 G91 X100.0 Y100.0;  
N7 S21;  
N8 G04 X10.0;  
N9 X100.0;
```

◆ Corner movement

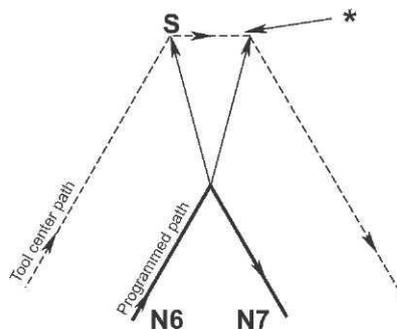
When two or more vectors are created at the end of the block, the tool performs linear movement from the first vector to the second. This movement is called corner movement.

If these vectors coincide with each other, the corner movement is not performed and the last vector is ignored.



If the conditions $\Delta V_x \leq \Delta V$ and $\Delta V_y \leq \Delta V$ are true, the last vector is ignored. The ΔV limit is set by parameter.

If these vectors do not coincide, a move is performed to turn around the corner. The move belongs to the latter block.



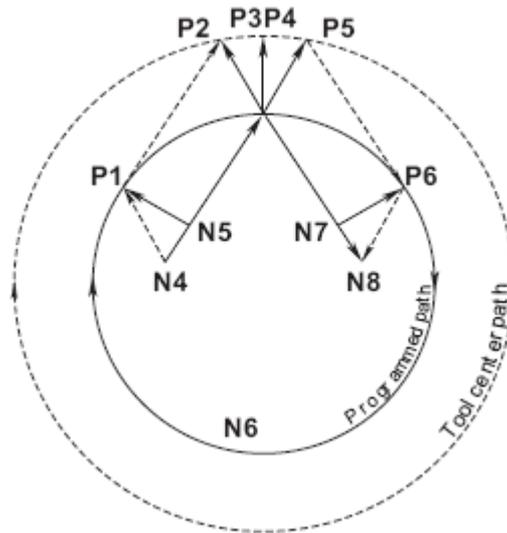
This move belongs to block **N7** and thus the feedrate is equal to that in block **N7**.

If block **N7** is **G00** mode, the tool moves according **G00** mode. If, however, it is **G01**, **G02** or **G03**, the tool moves in **G01** mode.

If the path specified in the next block is semicircular or more, the latter function is not executed.

The reason for this is as follows:

Compensation Function



N4 G41 G91 G01 X15.0 Y20.0;

N5 X15.0 Y20.0;

N6 G02 J-60.0;

N7 G01 X15.0 Y-20.0;

N8 G40 X15.0 Y-20.0;

If the vector is not ignored, the tool path is as follows:

P1 → P2 → P3 → (Circle) → P4 → P5 → P6

If the distance between **P2** and **P3** is negligible, **P3** is ignored. Thus the tool path is as follows:

P2 → P4

In other words, the circle cut by block N6 is ignored.

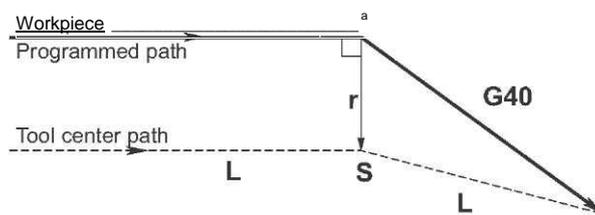
◆ **Interruption for manual operation**

For manual operation during cutter compensation mode refer to the corresponding chapter.

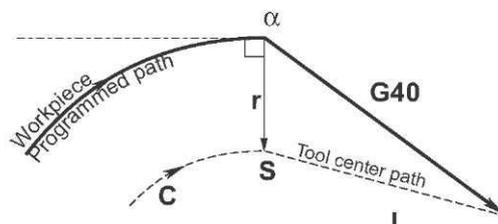
14.3.4 Tool Movement in Offset Mode Cancel

◆ **Tool movement around the inside of a corner**

Linear ® Linear



Circular → Linear

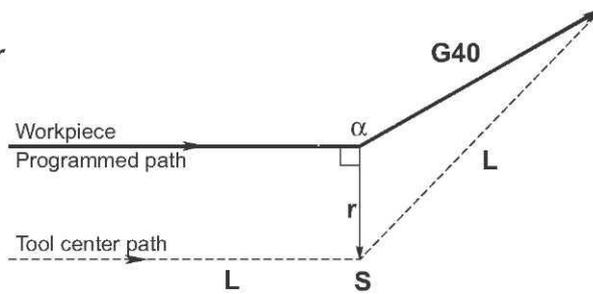


◆ Tool movement around the outside corner of an angle from 90 to 180 degrees

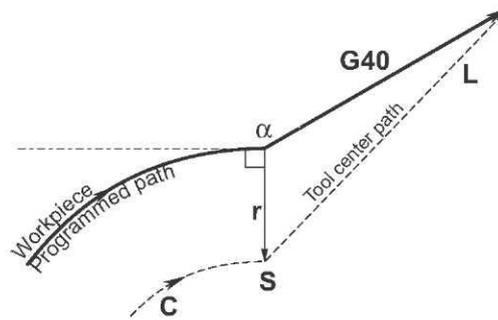
In start-up the tool path has two alternatives - type **A** and type **B**. They are selected by parameter.

Type A

Linear → Linear

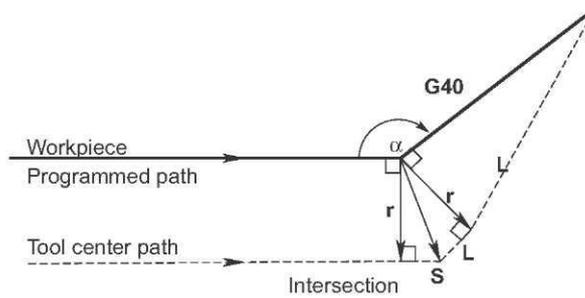


Circular → Linear

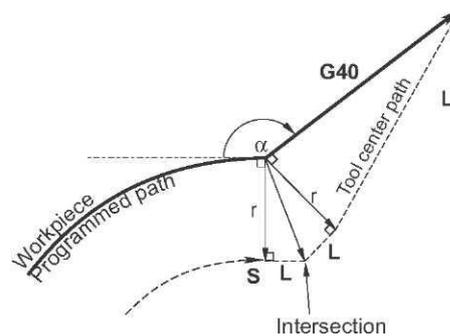


Type B

Linear → Linear



Circular → Linear

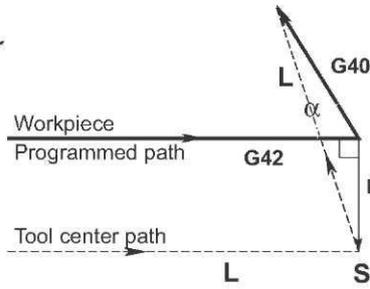


◆ **Tool movement around the outside corner of an acute angle**

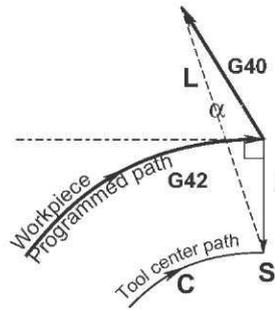
In start-up the tool path has two alternatives - type **A** and type **B**. They are selected by parameter.

Type A

Linear ® *inear*

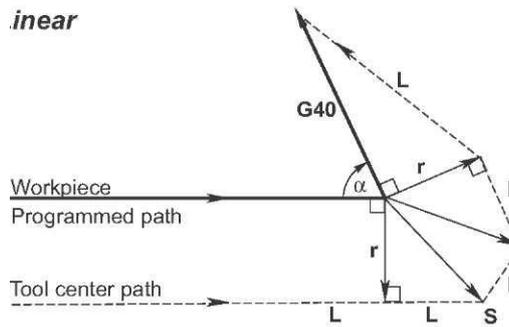


Circular ® *Linear*

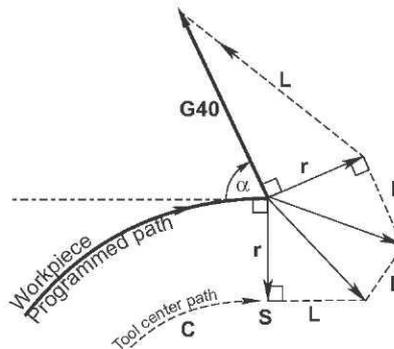


Type B

Linear ® *inear*

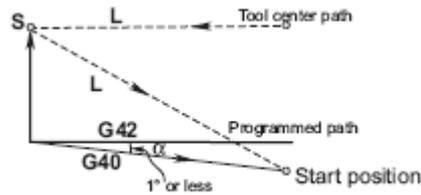


Circular ® *Linear*



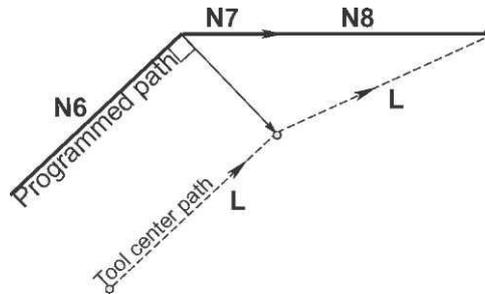
Compensation Function

◆ Tool movement type linear-linear around the outside of an acute angle less than 1 degree



◆ A block without tool movement specified together with offset cancel

When a block without tool movement is executed together with offset cancel, a vector is created with length equal to the offset value in the previous block. The vector is canceled with the next move command.



```
N6 G91 X100.0 Y100.0;
```

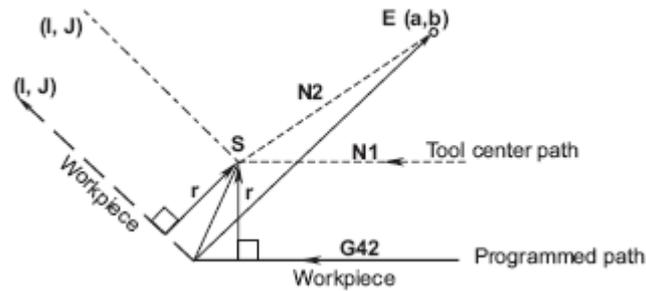
```
N7 G40;
```

```
N8 X100.0 Z0;
```

◆ Block containing G40 and I_J_K_

The previous block contains **G41** or **G42**

If a block containing **G41** or **G42** precedes a block containing **G40** and **I_J_K_**, the system assumes that the programmed path is the path from the end position, defined by the previous block to the vector defined by **(I,J)**, **(I,K)** or **(J,K)**. The offset direction is inherited from the previous block.



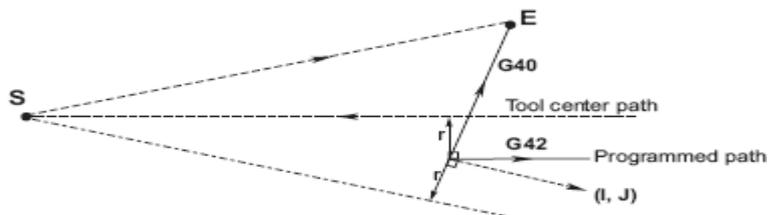
N1 (G42 mode)

In a block containing **G42** the tool center is moved towards **X**. **N2 G40**
Xa Yb I_J_;

In a block containing **G40** the tool moves towards **E**.

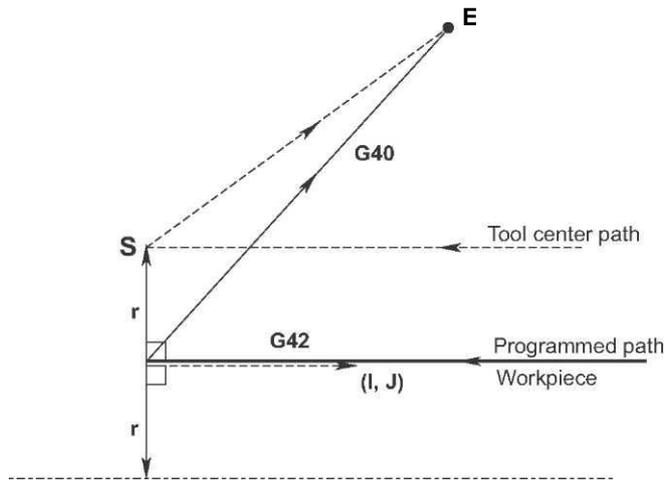
In a block containing **G42** the tool center is moved towards **X**.

In this case, note that **CNC** performs intersection of the tool path irrespective of whether inner or outer side machining is specified.



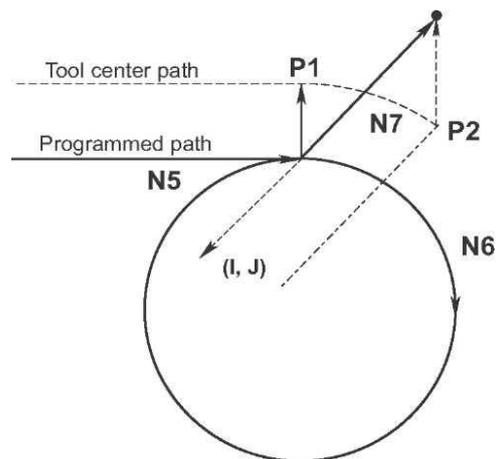
When intersection does not occur, the tool moves to a normal for the previous block position at the end of its execution.

Compensation Function



Length of the tool center path larger than the circumference of the circle

In the example shown below the tool does not trace the circle more than once. It moves along the arc from **P1** to **P2**. The interference check function described in the next chapter may issue an alarm.



```
(G41)
N5 G01 G91 X100.0; N6
G02 J-60.0;
N7 G40 G01 X50.0 Y50.0 I-10.0 J-10.0;
```

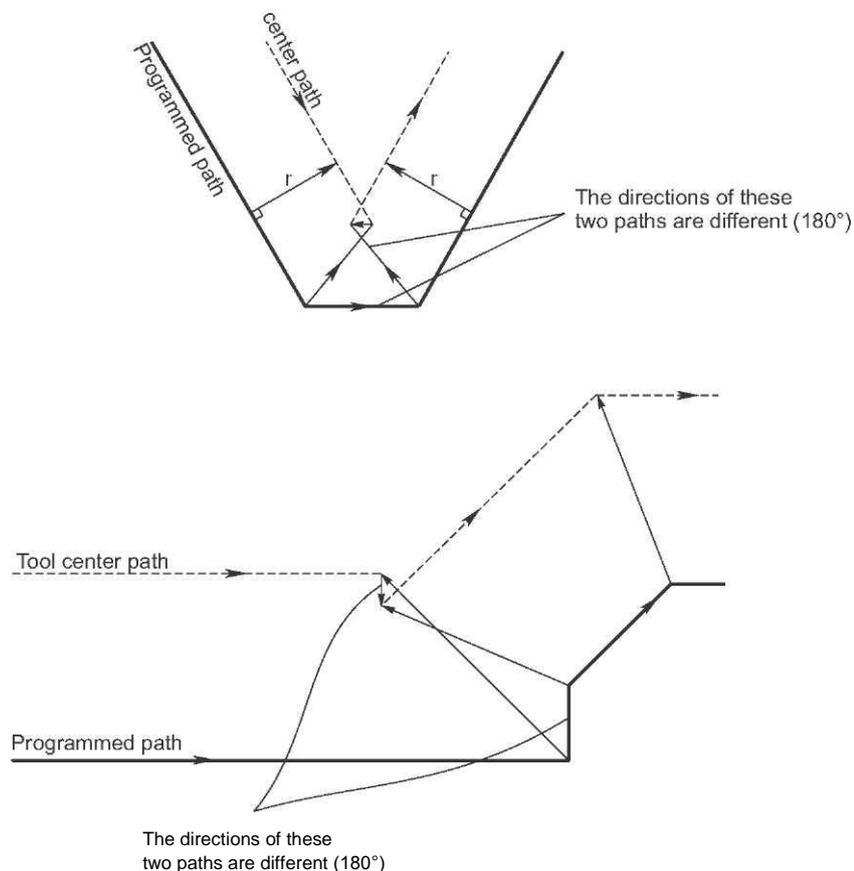
To move the tool along the circle several times program two or more arcs.

14.3.5 Interference Check

Tool overcutting is called interference. The interference check function checks in advance whether such an event can occur. However, not all the interferences can be checked by this function. The interference check is performed even if there is no overcutting.

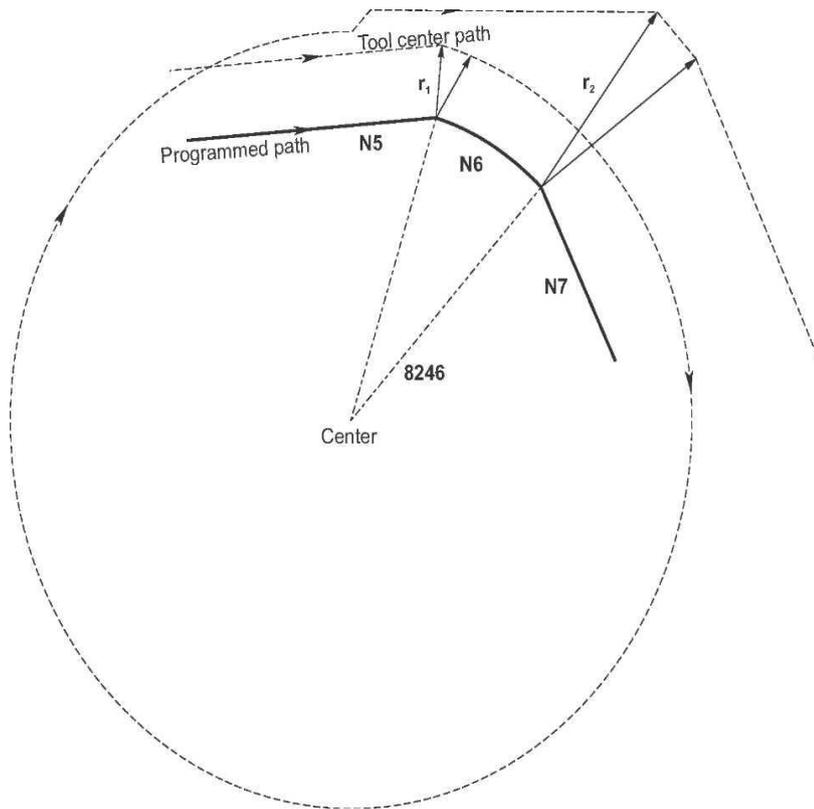
◆ **Criteria for detecting interference**

- (1) The tool path direction is different than the programmed one (from 90 to 270 degrees between these paths).



- (2) In addition to condition (1), the angle between the start and the end point of the tool center path is different than that defined between the start and the end point of the programmed path in circular machining (more than 180 degrees).

Compensation Function



(G41)

N5 G01 G91 X80.0 Y20.0 H1;

N6 G02 X32.0 Y-16.0 I-20.0 J-80.0 H2;

N7 G01 X20.0 Y-50.0;

(Tool compensation value corresponding to **H1** : $r1 = 20.0$)

(Tool compensation value corresponding to **H2** : $r2 = 60.0$)

In the example above, the arc in block N6 is in the first quadrant. After the compensation, it is placed in the four quadrants.

Correction of interference in advance

(1) Removal of the vector causing the interference

When a tool cutter compensation is performed for blocks **A**, **B** and **C**, between the blocks **A** and **B** vectors V_1 , V_2 , V_3 and V_4 are created and between blocks **B** and **C** the vectors are V_5 , V_6 , V_7 and V_8 . The nearest vectors are checked first. If interference conditions occur, the vectors are ignored.

But if the vectors that have to be ignored because of an intersection are the last ones in the angle, they cannot be ignored.

Check between vectors V4 and V5

Interference - V₄ and V₅ are ignored

Check between vectors V3 and V6

Interference - V₃ and V₆ are ignored

Check between vectors V2 and V7

Interference - V₂ and V₇ are ignored

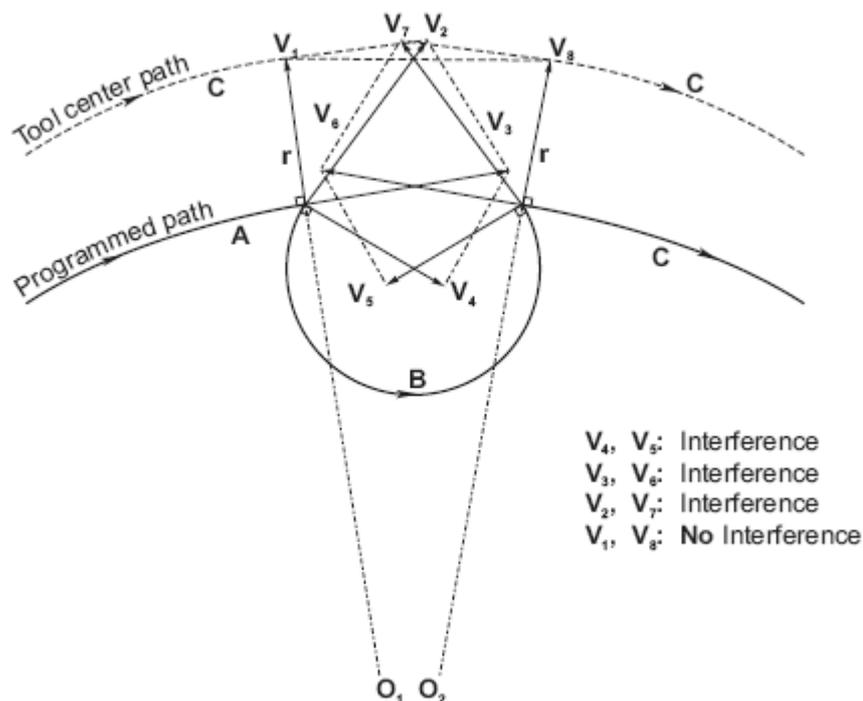
Check between vectors V1 and V8

Interference - V₁ and V₈ cannot be ignored

If a vector without interference is detected during the check, the following vectors are not checked. If block **B** is a movement along a circle, linear movement is performed when the vectors interfere.

Example 1:

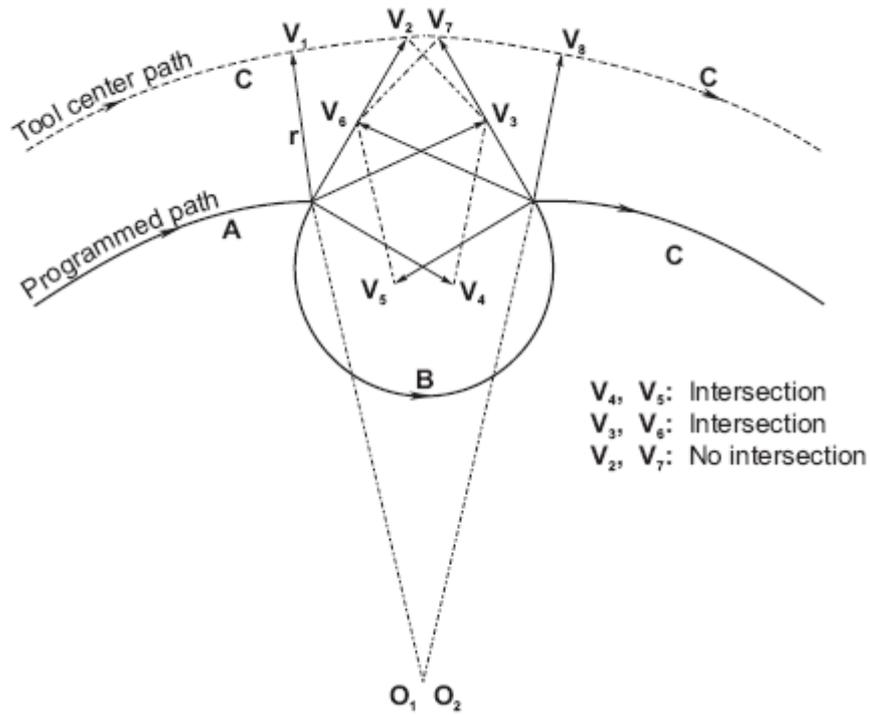
The tool moves linear from V₁ to V₈



Example 2:

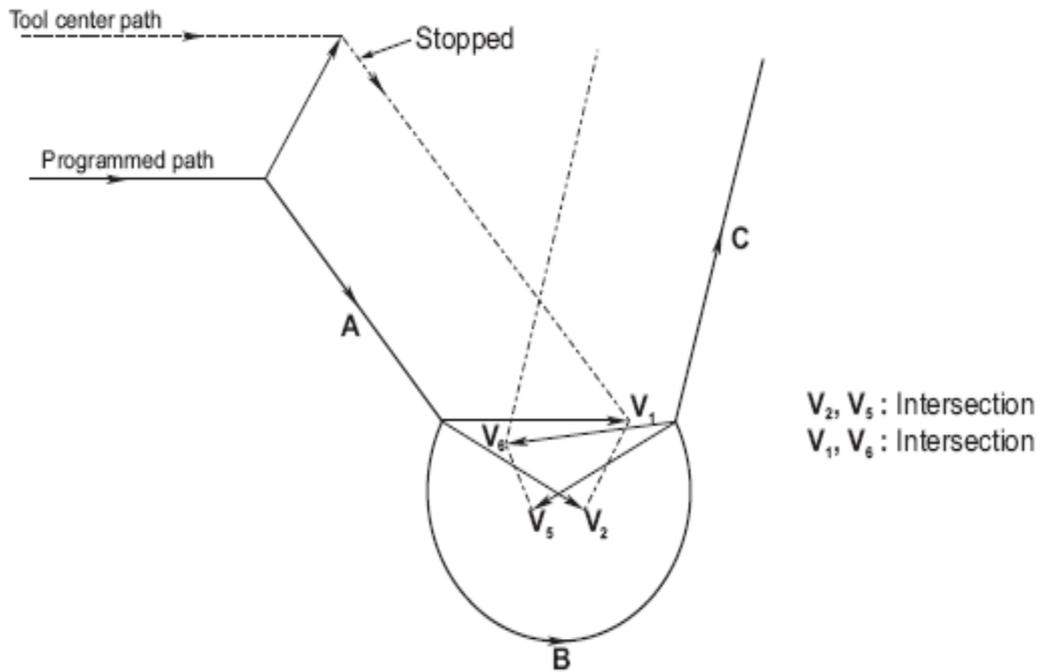
The tool moves linear from V₁, V₂, V₇ to V₈

Compensation Function



(2) If an interference occurs after correction (1), the tool is stopped and an alarm is displayed.

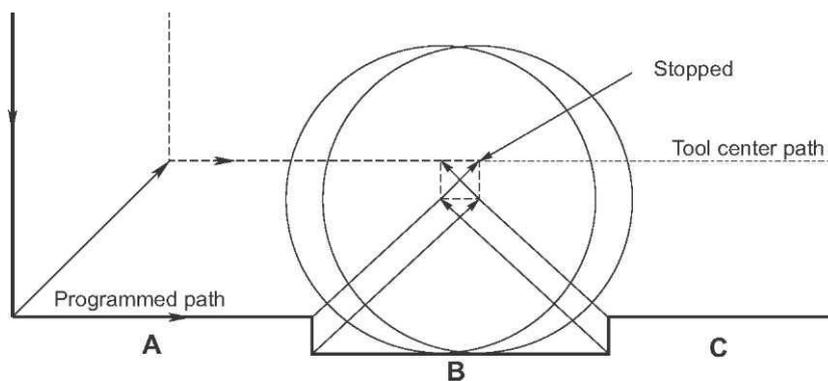
If an interference occurs after correction (1) or if just a couple of vectors exist at the beginning of the check and these vectors interfere, an alarm is displayed and the tool is stopped immediately after the execution of the previous block. If the block is executed as a single block operation, the tool is stopped at its end.



After ignoring vectors V_2 and V_5 because of an intersection, an intersection occurs between V_1 and V_6 vectors. An alarm is displayed and the tool is stopped.

◆ When an interference is assumed though actual interference does not exist

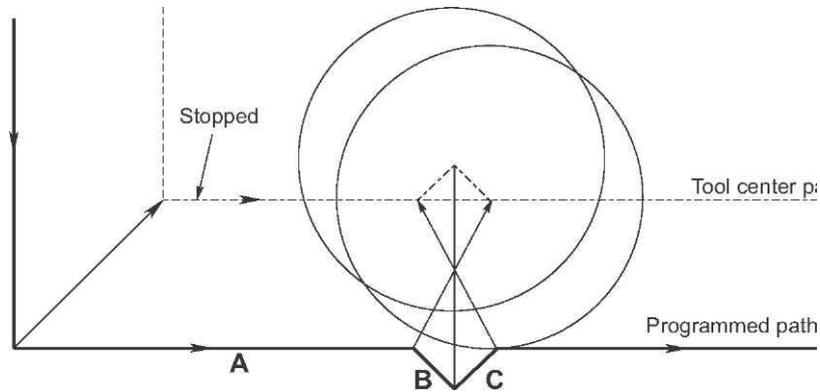
- (1) Depression which is smaller than the tool cutter compensation value.



Compensation Function

Actually there is no interference but since the direction specified in block **B** is opposite to that of the path after the compensation, the tool is stopped and an alarm is displayed.

(2) Groove which is smaller than the tool cutter compensation value.

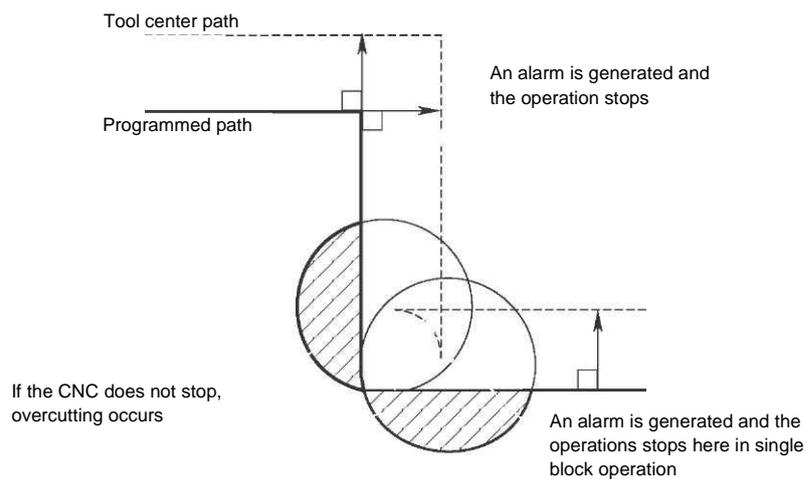


The situation is the same as in **(1)**. The programmed path will be opposite to the tool path after the compensation is applied. The situation is assumed as interference. An alarm is displayed and the machine is stopped.

14.3.6 Overcutting by Cutter Compensation

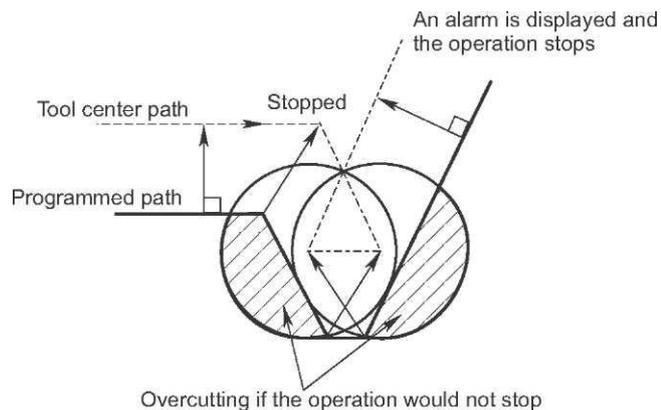
◆ Machining the inside of a corner with radius smaller than the cutter radius

When the radius of a corner is smaller than the compensation radius, as the inner offsetting of the cutter will lead to overcutting, an alarm is displayed and the CNC stops at the beginning of the block. When just one block is machined, the overcutting is executed cause the tool stops after the execution of the block.



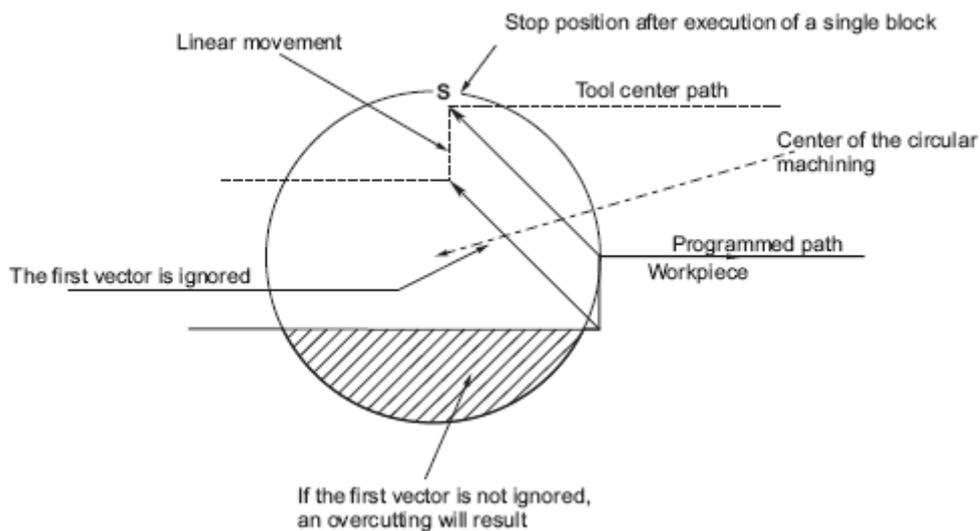
◆ Machining a groove smaller than the tool radius

As the tool cutter compensation corrects the tool center path in direction opposite to the programmed one, an overcutting will result. In this case an alarm is displayed and the **CNC** stops at the beginning of the block.



◆ Machining a step smaller than the tool radius

If machining of one step is commanded when machining an arc by steps smaller than the tool radius, the path of the tool center with the ordinary offset becomes reverse to the programmed one. In this case the first vector is ignored and the tool is moved linearly to the second vector position. When a single block is executed the machining is stopped in that point. If the machine is not in a single block operation, the cycle operation continues. If the step is linear, no alarm will be displayed and the cutting will be correct. Otherwise uncut part will remain.

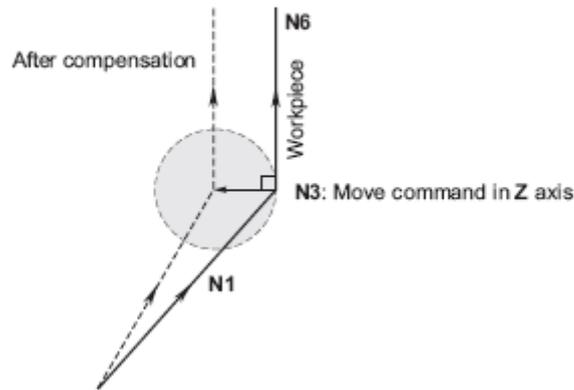


If the first vector is not ignored, overcutting occurs. Otherwise the tool moves linearly.

◆ Starting compensation and cutting along the Z axis

Normally, such a method is used according to which the tool is moved along the **Z** axis, after performing a cutter compensation at some distance from the workpiece at the beginning of the machining.

In the case above, if you wish to divide the motion along **Z** axis at rapid traverse and the feedrate, follow the procedure described below.



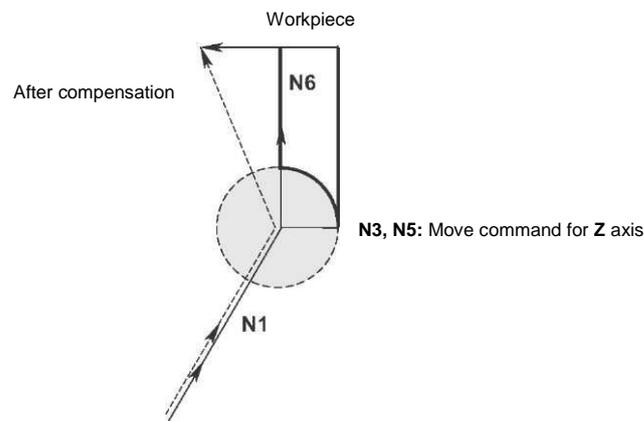
```

N1 G91 G00 G41 X50.0 Y50.00 H1;
N3 G01 Z-300.0 F100;
N6 Y100.0 F200;
    
```

In the example program above, when block **N1** is executed, blocks **N3** and **N6** are also loaded in the buffer and depending on their relationship, a proper compensation is performed, as shown on the figure above.

Then block **N3** (a move command along the **Z** axis) is divided as follows: as there are two blocks with move commands which are not included in the selected plane and block **N6** cannot be loaded into the buffer, the tool center path is calculated according the data in block **N1** from the figure above. In this case the offset vector is not calculated at the start position and overcutting may result.

The example above should be modified as follows:



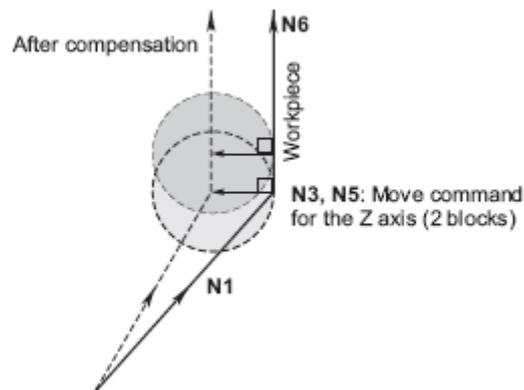
```

N1 G91 G00 G41 X50.0 Y50.00 H1;
    
```

Compensation Function

```
N3 G01 Z-250.0;  
N5 G01 Z-50.0 F100;  
N6 Y100.0 F200;
```

A move command in the same direction as the move command after the shift along the **Z** axis should be specified.



```
N1 G91 G00 G41 X50.0 Y40.00 H1;  
N2 Y10.0;  
N3 Z-250.0;  
N5 G01 Z-50.0 F100;  
N6 Y100.0 F200;
```

As block **N2** performs a move command in the same direction as that in block **N6**, a correct compensation is performed.

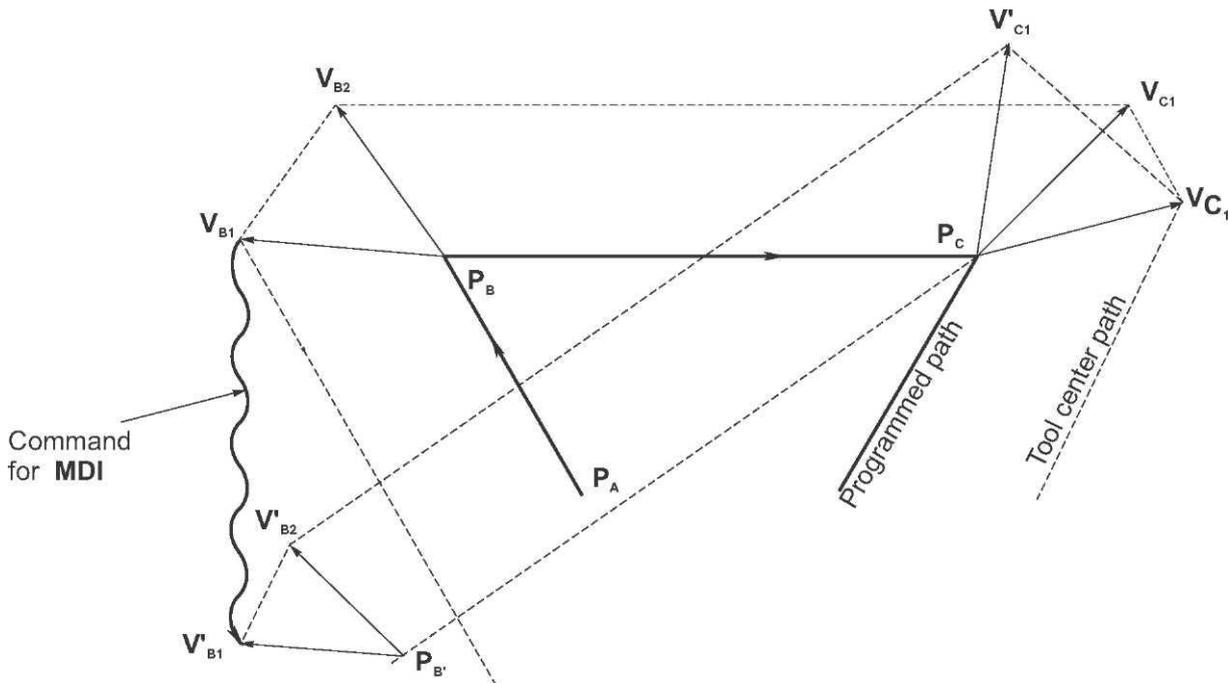
14.3.7 Input Command From MDI

Cutter compensation **C** is not performed for commands input from the **MDI** panel.

However, when the automatic operation with absolute commands is temporarily stopped by a single block execution function, the **MDI** operation is performed, the automatic operation is restored again and the tool path is as follows:

In this case the start position vectors of the next block are translated and other vectors are generated for the next two blocks. For this reason the next block performs correct cutter compensation **C**.

Compensation Function

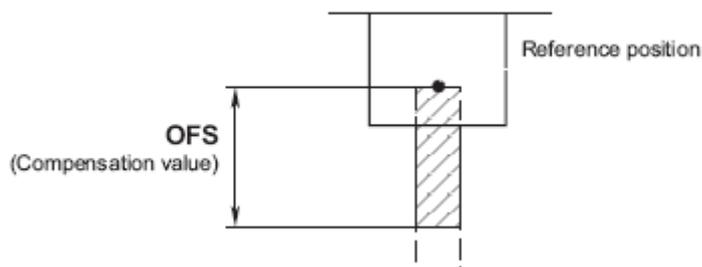


When P_A , P_B and P_C positions are programmed in an absolute command, the tool is stopped by a single block function after executing blocks from P_A to P_B and the tool is moved by MDI operation. The vectors V_{B1} and V_{B2} are translated to V'_{B1} and V'_{B2} and the offset vectors are recalculated for vectors V_{C1} and V_{C2} between blocks P_B - P_C and P - P_{CD}

The compensation is performed correctly by P_C until vector V_{B2} is not recalculated.

14.4 TOOL COMPENSATION VALUES, NUMBER OF COMPENSATION VALUES AND ENTERING VALUES FROM THE PROGRAM (G10)

Tool compensation



Compensation Function

The tool compensation values can be input in the **CNC** memory from the TFT/ **MDI** panel or from the program.

The tool compensation value is loaded from the **CNC** memory by specifying the corresponding code after the **H** address in the program.

The value is used for compensation of the tool length, cutter compensation of tool shift.

The table below shows the valid range of tool compensation values.

◆ Valid range of tool compensation values

Valid input range of tool compensation value

Tool compensation value	
Metric input	Inch input
0 to ±999.999mm	0 to ±99.9999inch

◆ Number of tool compensation values and the addresses to be specified

The memory can store up to 99 tool compensation values. Address **H** is used in the program. The used address depends on function that is used: tool length compensation or cutter compensation.

◆ Input of tool compensation value by programming

Setting the format of tool compensation memory and the tool compensation value.

Format:

G10P_R_;

P: Number of tool compensation

R: Tool compensation value in absolute command mode (**G90**) or value which should be added to the specified tool compensation in incremental mode (**G91**).

Note:

To provide compatibility with the format of older **CNC** programs, the system allows specifying **L1**:

Compensation Function

G10 L1 P_ R_;

14.5 Read and write parameters from program (G10)

Format:

G10 L50 N_ R_ ; write value R at parameter number N

G10 L50 N_ Q_ ; read parameters number N to variable Q

G10 L55 N_ R_ ; write value R at offset number N

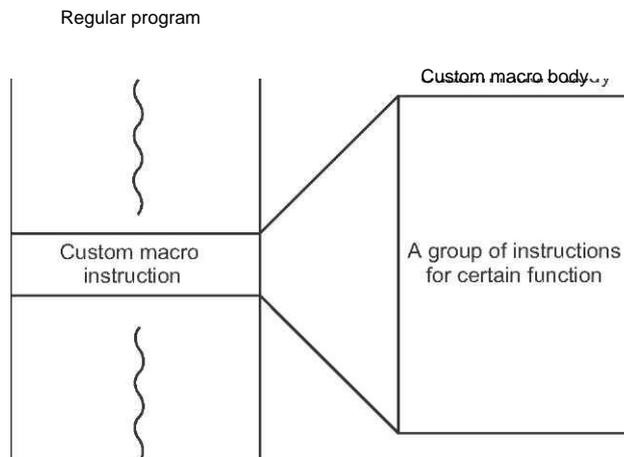
G10 L55 N_ Q_ ; read offset value at variable number Q

15. CUSTOM MACRO

A function which performs a group of instructions can be stored in the memory just like a subprogram. The stored function is represented by one instruction. So the programmer have to specify only that instruction and all the commands that are within it will be executed.

A group of named instructions is called "body of a custom macro" and the name for calling the macro is called "instruction of a custom macro". The body of the custom macro can be called only macro and the instruction for calling it is named a command for calling macro.

The programmers have to learn just the representative macro instructions without the



necessity of remembering all the instructions in the macro body. The three most important points of the custom macro are: there can be used variables in the macro body, different operations can be performed over the variables and actual values can be assigned to the variables.

Note:

*Machine builders are requested to attach a list with already made macros to the **CNC** manual. If it is necessary to replace some of the macros because of failures, you have to call the people in charge who know the contents of the custom macros and who will replace them immediately.*

15.1 CUSTOM MACRO COMMAND

The custom macro command is that command which call the corresponding macro.

15.1.1 Single Call

The format of the command is as follows:

M98 P_;
↑ Called macro No.

With the above command, the macro body specified by is called.

15.1.2 Modal Call

Format:

G66 P_;
↑ Called macro body

The above command selects a macro modal call mode for **CNC**. In other words, each time a block is executed following the above command, the macro specified by **P** is executed. More, the block following the above command can point to an argument. For more details on this argument refer to the next chapter. The macro modal call mode can be canceled with the command shown below:

G67;

CAUTION:

1. Blocks containing **G66** and **G67** cannot call macros.
2. In **MDI** mode, the **G66** command specifies a macro modal call mode. The **G67** command cancels that mode. However, other commands do not call a macro even when they are called between **G66** and **G67**. They are executed as normal commands **instead**.
3. Addresses different from **O**, **N** and **P** are ignored in blocks with **G66** and **G67**.
4. A repetition can be set in macro modal calling. Only the four low-order digits of the value specified by **P** in **G66** block are valid.

5. The maximum nesting level in macro modal call is 1. The maximum nesting level in subprogram call is 8. However, the maximum nesting level for macro modal call and subprogram call is 8.

15.1.3 Argument Specification

An argument means an actual value given to a variable which has been used in a called macro. The argument can be specified for all usable addresses excluding O. The format of the argument specification is the same as in normal CNC command. The limitation for each address valid for each CNC command as decimal point, sign, maximum number of digits, etc. are applicable to this format too.

The table below specifies the correspondence between the addresses of the specified arguments and the number of the variables.

Correspondence between addresses and variable numbers

Correspondence between addresses and variable numbers

Variable number (value)	Variable number (flag)	Address	Remarks
#8004	#8104	I	
#8005	#8105	J	
#8006	#8106	K	
#8009	#8109	F	
#8010	#8110	G	
#8011	#8111	H	
#8013	#8113	M	
#8014	#8114	N	
#8016	#8116	P	
#8017	#8117	Q	
#8018	#8118	R	
#8019	#8119	S	
#8020	#8120	T	
#8024	#8124	X	
#8025	#8125	Y	
#8026	#8126	Z	

Custom Macro

#8000's variables specify the value if an argument has been set. If, however, an argument has not been set, they are initialized as shown below:

(a) Reference in CNC command

The address is neglected.

(b) Reference in macro command and branch command.

The value is undefined. Use it after confirming #8100's numbers.

Correspondence between **G** codes of the argument specification and variable numbers

Corresponding between **G** codes of the argument specification and variable number

Variable number (value)	Variable number (flag)	G code group number	G codes of the argument specification
#8030	#8130	00	One shot and others
#8031	#8131	01	G00, G01, G02, G03
#8032	#8132	02	G17, G18, G19
#8033	#8133	03	G90, G91
#8035	#8135	05	G94, G95
#8036	#8136	06	G20, G21
#8037	#8137	07	G40, G41, G42
#8038	#8138	08	G43, G44, G49
#8039	#8139	09	G73, G74, G76, G80 - G89
#8040	#8140	10	G98, G99
#8041	#8141	11	G66, G67

15.2 CUSTOM MACRO BODY

There can be used CNC commands which work with normal **CNC** variables, calculations and branch commands in the custom macro body. The custom macro body starts from the program No specified immediately after O and ends with **M99**.

Example of a construction of a custom macro body:

O;	Program No
G65 H01 . . . ;	Calculation command
G90 G00 X#101 ;	CNC command using variables
G65 H82 . . . ;	Branch command
M99 ;	End of custom macro

15.2.1 Variables

The macro can be made more flexible and convenient if a variable which gets a value when the macro is called or which is used during macro execution is specified. The variables can be identified from each other by variable numbers.

(1) How to express a variable

The variable can be expressed by the number following symbol # as shown below:

#i (i = 1, 2, 3, 4 . . .)

Example: #5, #109, #1005

(2) How to use variables

The number following the address can be replaced by a variable. Imagine that you specify **<Address> #i** or **<Address> -#i**. It means that the variable value or its negative value serves as a value for the command in the address.

Example:

F#103	- F15 is specified when #103 = 15
Z-#110	- Z-250 is specified when #110 = 250
G#130	- G3 is specified when #130 = 3

When replacing a variable number with another variable, this is not expressed with **##100** for example but with **#9100**. The digit (after the symbol # specifies replacement of the variable number which is specified with the number following **9**).

Example:

If #100 = 105 and #105 = -500,
X#9100 specifies that the command is X-500 and X-#9100 specifies command X500.

Note:

1. There cannot be used variables for addresses N and O. A command N#120 or O#100 cannot be specified.
2. It is not possible to specify a value which exceeds the maximum value for a given address. When #30 = 120, the command G#30 exceeds the maximum allowable value.

15.2.2 Kind of Variables

The variables are divided into common variables and system variables as this depends on the variable number and its application. The variable characters differ from each other.

(1) Common variables from #100 to #131 and from #500 to #599

The common variables are common for the whole program and for all macros which are called from the main program. It means that #i in one macro is equal to #i in another macro.

The common variables from #100 to #130 are cleared when the power is switched off and are set to 0 right after the power is on. The common variables from #500 to #599 are not zeroed even when the power is off. Their values stay unchanged.

(2) System variables

The system variables are defined as variables whose application remain fixed.

(a) Tool offset amount variables from #1 to #99.

The offset can be read from the system variables from #1 to #99. They concern the tool offset amount and their values can be modified. There are such variables which are not used for offset among them and they can be used as common variables (like these from #500 to #599).

(b) Interface input signals from #1000 to #1015, #1032

The interface signals can be read from the system variables from #1000 to #1015.

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
UI15	UI14	UI13	UI12	UI11	UI10	UI9	UI8	UI7	UI6	UI5	UI4	UI3	UI2	UI1	UI0
#1015	#1014	#1013	#1012	#1011	#1010	#1009	#1008	#1007	#1006	#1005	#1004	#1003	#1002	#1001	#1000

Input signal	Variable value
Contact closed	1
Contact opened	0

When system variable #1032 is read, all input signals can be read at once:

$$\#1032 = \sum_{i=0}^{15} \#(1000+i) \times 2^i$$

Notes:

1. The values of system variables from #1000 to #1032 cannot be changed.
2. The system variables from #1000 to #1015 can be shown with the diagnostic function.
DGN No. 130 UI0 to UI7 DGN
No. 131 UI8 to UI15
3. The system variables from #1000 to #1032 can be used only with PMC.

(c) Interface output signals from #1100 to #1115, #1132

Values can be written in the system variables from #1000 to #1015 which can be used as output interface signals afterwards.

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
UO15	UO14	UO13	UO12	UO11	UO10	UO9	UO8	UO7	UO6	UO5	UO4	UO3	UO2	UO1	UO0
#1115	#1114	#1113	#1112	#1111	#1110	#1109	#1108	#1107	#1106	#1105	#1104	#1103	#1102	#1101	#1100

Output signal	Variable value
Contact closed	1
Contact opened	0

When system variable #1032 is written, all output signals can be sent at once:

$$\#1132 = \sum_{i=0}^{15} \#(1100+i) \times 2^i$$

CAUTION:

If a value different from 0 or 1 is written in the system variables from #1100 to #1115, it is assumed as 1.

Notes:

1. The values of system variables from #1100 to #1132 can be read.
2. The system variables from #1100 to #1132 can be shown with the diagnostic function.

**DGN No. 162 U00 to U07 DGN No.
163 U08 to U015**

3. The system variables from #1100 to #1132 can be used only with PMC.

(d) Positional information

The information for the position can be obtained from the system variables from #5000 to #6044. The unit for the positional information is 0.001 mm in metric system and 0.0001 inch in inch system.

System variables	Position information	Reading while moving	Compensations
#5001 #5002 #5003 #5004	Position at the end of a block along X Position at the end of a block along Y Position at the end of a block along Z Position at the end of a block along 4th	Possible	Not considered. Position of tool nose (Program command position)
#5041 #5042 #5043 #5044	Cutter position along X axis Cutter position along Y axis Cutter position along Z axis Cutter position along 4th axis	Impossible	Considered. Position of tool reference point. (Absolute indication)
#5061 #5062 #5063 #5064	Skip signal position of X axis Skip signal position of Y axis Skip signal position of Z axis Skip signal position of 4th axis	Possible	Considered. Position of tool reference point.
#5080 #5081 #5082 #5083	Value of cutter compensation Value of tool length compensation along X axis Value of tool length compensation along Y axis Value of tool length compensation along Z axis	Possible	
#6041 #6042 #6043 #6044	Machine position X axis Machine position Y axis Machine position Z axis Machine position 4th axis	Possible	.

Note:

1. It is not possible to write values in the system variables from #5001 to #5083.

- When the skip signal does not turn on when G31 is commanded, the skip signal position is the end point of this block.

15.2.3 Macro Instructions type A

General form:

G65HmPi Q#j R#k;

m: specifies a macro function from 01 to 99

#i: name of a variable to load the arithmetic result

#j: name of the first variable to be operated. Using a constant is possible. **#k:** name of the second variable to be operated. Using a constant is possible.

Meaning: #i = #j E #k

↑ Operator specified with Hm

Example:

P#100 Q#101 R#102 #100 = #101 E #102
 P#100 Q#101 R#15 #100 = #101 E 15
 P#100 Q-100 R#102 #100 = -100 E #102
 P#100 Q120 R-50 #100 = 120 E -50
 P#100 Q-#101 R#102 #100 = -#101 E #102

CAUTION:

- There cannot be put a decimal point in the variable value. Therefore, the meaning of each value is the same as that without decimal point when used with the corresponding address.

Example:

#100 = 10
 X#100 - - 0.01 mm (metric input)

- The variables noting an angle should be specified in degrees and the least increment is 1/1000 degrees.

Example:

100----- 0.1 degrees

Note:

H code specified with G65 does not affect the selection of offset.

Macro instructions type A

G code	H code	Function	Definition
G65	H01	Definition, substitution	#i = #j
G65	H02	Addition	#i = #j + #k
G65	H03	Substraction	#i = #j - #k
G65	H04	Product	#i = #j x #k
G65	H05	Division	#i = #j / #k
G65	H11	Logical sum	#i = #j . OR . #k
G65	H12	Logical product	#i = #j . AND . #k
G65	H13	Exclusive OR	#i = #j . XOR . #k
G65	H21	Square root	#i = $\sqrt{\#j}$
G65	H22	Absolute value	#i = #j
G65	H23	Remainder	#i = #j - trunc (#j / #k)x #k
G65	H24	Conversion from BCD to binary	#i = BIN
G65	H25	Conversion from binary to BCD	#i = BCD
G65	H26	Combined multiplication/division	#i = BCD (#j)
G65	H27	Combined square root 1	#i = $\sqrt{\#j^2 + \#k^2}$
G65	H28	Combined square root 2	#i = $\sqrt{\#j^2 - \#k^2}$
G65	H31	Sine	#i = #j . SIN (#k)
G65	H32	Cosine	#i = #j . COS (#k)
G65	H33	Tangent	#i = #j . TAN (#k)
G65	H34	Arctangent	#i = #j . ATAN (#k)
G65	H80	Unconditional divergence	GOTO n
G65	H81	Conditional divergence 1	IF #j = #k, GOTO n
G65	H82	Conditional divergence 2	IF #j = #k, GOTO n
G65	H83	Conditional divergence 3	IF #j > #k, GOTO n
G65	H84	Conditional divergence 4	IF #j < #k, GOTO n
G65	H85	Conditional divergence 5	IF #j > #k, GOTO n
G65	H86	Conditional divergence 6	IF #j £ #k, GOTO n
G65	H99	P/S alarm occurrence	P/S alarm number 500+n occurence

15.2.4 Macro instructions type B

General form:

(* macro instruction type B)

TYPE A	TYPE B
G65 H01 P#100 Q#101	(* #100 = #101)
G65 H02 P#100 Q#101 R#102	(* #100 = #101 + #102)
G65 H03 P#100 Q#101 R#102	(* #100 = #101 - #102)
G65 H04 P#100 Q#101 R#102	(* #100 = #101 * #102)
G65 H05 P#100 Q#101 R#102	(* #100 = #101 / #102)
G65 H11 P#100 Q#101 R#102	(* #100 = #101 OR #102)

G65 H12 P#100 Q#101 R#102	(* #100 = #101 AND #102)
G65 H13 P#100 Q#101 R#102	(* #100 = #101 XOR #102)
G65 H21 P#100 Q#101	(* #100 = SQRT #101)
G65 H22 P#100 Q#101	(* #100 = ABS #101)
G65 H23 P#100 Q#101 R#102	(* #100 = #101 TRNC #102)
G65 H24 P#100 Q#101	(* #100 = BIN #101)
G65 H25 P#100 Q#101	(* #100 = BCD #101)
G65 H26 P#100 Q#101 R#102	(* #100 = #101 CMDV #102)
G65 H27 P#100 Q#101 R#102	(* #100 = #101 SQRA #102)
G65 H28 P#100 Q#101 R#102	(* #100 = #101 SQRS #102)
G65 H31 P#100 Q10000 R#102	(* #100 = 10000 * SIN #102)
G65 H32 P#100 Q10000 R#102	(* #100 = 10000 * COS #102)
G65 H33 P#100 Q10000 R#102	(* #100 = 10000 * TAN #102)
G65 H34 P#100 Q#101 R#102	(* #100 = 10000 * ATAN #102)
G65 H80 P101	(* GOTO 101)
G65 H81 P102 Q#101 R#102	(* IF #101 EQ #102 GOTO 102)
G65 H82 P103 Q#101 R#102	(* IF #101 NE #102 GOTO 103)
G65 H83 P104 Q#101 R#102	(* IF #101 GT #102 GOTO 104)
G65 H84 P105 Q#101 R#102	(* IF #101 LT #102 GOTO 105)
G65 H85 P106 Q#101 R#102	(* IF #101 GE #102 GOTO 106)
G65 H86 P107 Q#101 R#102	(* IF #101 LE #102 GOTO 107)

Note. Macro instructions can be written in mode **[EDIT]** by the functional button **[CMNT]**

15.2.5 Macro instructions commands type A and type B

(a) Definition and substitution of a variable $\#i = \#j$

G65 H01 P $\#i$ Q $\#j$;

(Example):

type A - G65 H01 P#101 Q1055 ; (#101 = 1055)
type B - (* #101 = 1055)

Addition $\#i = \#j + \#k$ G65 H02 P $\#i$ Q $\#j$ R $\#k$;

(Example):

type A - G65 H02 P#101 Q#102 R15 ; (#101 = #102 + 15)
type B - (* #101 = #102 + 15)

Subtraction $\#i = \#j - \#k$ G65 H03 P $\#i$ Q $\#j$ R $\#k$;

(Example):

type A - G65 H03 P#101 Q#102 R#103 ; (#101 = #102 - #103)
type B - (* #101 = #102 - #103)

(d) Multiplication $\#i = \#j \times \#k$ G65

H04 P $\#i$ Q $\#j$ R $\#k$;

(Example):

type A - G65 H04 P#101 Q#102 R#103 ; (#101 = #102 x #103)
Type B - (* #101 = #102 * #103)

Division $\#i = \#j / \#k$ G65 H05 P $\#i$ Q $\#j$ R $\#k$;

(Example):

type A - G65 H05 P#101 Q#102 R#103 ; (#101 = #102 / #103)

type B - (* #101 = #102 / #103)

(f) Logical sum #i = #j OR #k G65

H11 P#i Q#j R#k;

(Example):

type A - G65 H11 P#101 Q#102 R#103 ; (#101 = #102 OR #103)

type B - (* #101 = #102 OR #103)

(g) Logical product #i = #j AND #k G65 H12 P#i

Q#j R#k;

(Example):

type A - G65 H12 P#101 Q#102 R#103 ; (#101 = #102 AND #103)

type B - (* #101 = #102 AND #103)

(h) Exclusive OR #i = #j XOR #k

(i) G65 H13 P#i Q#j R#k;

(Example):

type A - G65 H13 P#101 Q#102 R#103 ; (#101 = #102 XOR #103)

type B - (* #101 = #102 XOR #103)

(i) Square root #i = $\sqrt{\#j}$

G65 H21 P#i Q#j;

(Example):

type A - G65 H21 P#101 Q#102 ; (#101 = $\sqrt{\#102}$)
type B - (* #101 = SQRT #102)

(j) Absolute value #i = $|\#j|$ G65 H22 P#i Q#j;

(Example):

type A - G65 H22 P#101 Q#102 ; (#101 = $|\#102|$)
type B - (* #101 = ABS #102)

(k) Remainder #i = $\#j - \text{trunc}(\#j / \#k) \times \#k$

G65 H23 P#i Q#j R#k;

(Example):

type A - G65 H23 P#101 Q#102 R#103; (#101 = $\#102 - \text{trunc}(\#102 / \#103) \times \#103$)
type B - (* #101 = #102 TRNC #103)

(l) Conversion from BCD to binary #i = BIN (#j)

G65 H24 P#i Q#j;

(Example):

type A - G65 H24 P#101 Q#102 ; (#101 = BIN (#102))
type B - (* #101 = BIN #102)

(m) Conversion from binary to BCD #i = BCD (#j)

G65 H25 P#i Q#j;

(Example):

type A - G65 H25 P#101 Q#102 ; (#101 = BCD (#102))

type B - (* #101 BCD #102)

(n) Combined multiplication/division $\#i = (\#i \times \#j) / \#k$

G65 H26 P#i Q#j R#k;

(Example):

type A - G65 H26 P#101 Q#102 R#103; (#101 = (#101 x #102) / #103)

type B (* #101 = #102 CMDV #103)

(o) Combined square root 1 $\#i = \sqrt{\#j^2 + \#k^2}$

G65 H27 P#i Q#j R#k;

(Example):

type A - G65 H27 P#101 Q#102 R#103; (#101 = $\sqrt{\#102^2 + \#103^2}$)

type B (* #101 = #102 SQRA #103)

(p) Combined square root 2 $\#i = \sqrt{\#j^2 - \#k^2}$

G65 H28 P#i Q#j R#k;

(Example):

type A - G65 H28 P#101 Q#102 R#103; (#101 = $\sqrt{\#102^2 - \#103^2}$)

type B - (* #101 = #102 SQRS #103)

(q) Sine #i = #j.SIN(#k)

G65 H31 P#i Q#j R#k;

(Example):

Type A - G65 H31 P#101 Q#102 R#103; (#101 = #102.SIN(#103))

type B - (* #101 = #102 SIN #103)

(r) Cosine #i = j.COS(#k)

G65 H32 P#i Q#j R#k;

(Example):

type A – G65 H32 P#101 Q#102 R#103; (#101 =#102.COS(#103))

type B - (* #101 = #102 COS #103)

(s) _Tangent #i = #j.TAN(#k)

G65 H33 P#i Q#j R#k;

(Example):

type A -G65 H33 P#101 Q#102 R#103; (#101 = #102.TAN(#103))

type B - (* #101 = #102 TAN #103)

(t) Arc tangent #i = ATAN(#j / #k)

G65 H34 P#i Q#j R#k;

(Example):

type A - G65 H34 P#101 Q#102 R#103; (#101 = ATAN(#102 / #103))

type B - (* #101 = #102 ATAN #103)

CAUTION: The angle in **(q)** to **(t)** must be in degrees and the little input increment is 1/1000

degrees.

Note:

1. If either **Q** or **R** is requested but they are not specified, their value is assumed as zero.
2. All digits following the decimal point are truncated if each arithmetic result includes decimal point.

◆ **Branch commands**

(a) Unconditional branch

G65 H80 Pn ; n : Sequence number

(Example):

type A - G65 H80 P120 ; (Branch to N120)

type B - (* GOTO 120)

(b) Conditional branch 1 #j EQ #k (branch if equal)

Custom Macro

G65 H81 Pn Q#j R#k ; n : Sequence number

(Example):

type A - G65 H81 P1000 Q#101 R#102;
IF #101 = #102, branch to N1000
else next command
type B - (* IF #101 EQ #102 GOTO 120)

(c) Conditional branch 2 #j NE #k (branch if not equal)

G65 H82 Pn Q#j R#k ; n : Sequence number

(Example):

type A - G65 H82 P1000 Q#101 R#102;
IF #101 not equal to #102, branch to N1000
else next command

type B - (* IF #101 NE #102 GOTO 1000)

(d) Conditional branch 3 #j GT #k (branch if higher)

G65 H83 Pn Q#j R#k ; n : Sequence number

(Example):

type A - G65 H83 P1000 Q#101 R#102;
#101 > #102, branch to N1000
else next command

type B - (* IF #101 GT #102 GOTO 1000)

(e) Conditional branch 4 #j LT #k (branch if lower)

G65 H84 Pn Q#j R#k ; n : Sequence number

(Example):

type A - G65 H84 P1000 Q#101 R#102;
#101 < #102, branch to N1000
else next command

type B - (* IF #101 LT #102 GOTO 1000)

(f) Conditional branch 5 #j GE #k (branch if higher or equal)

G65 H85 Pn Q#j R#k ; n : Sequence number

(Example):

type A - G65 H85 P1000 Q#101 R#102;
IF #101 \geq #102, branch to N1000
else next command
type B - (* IF #101 GE #102 GOTO 1000)

(g) Conditional branch 6 #j LE #k (branch if lower or equal)

G65 H86 Pn Q#j R#k ; n : Sequence number

(Example):

type A - G65 H86 P1000 Q#101 R#102;
IF #101 \leq #102, branch to N1000
else next command
type B - (* IF #101 LE #102 GOTO 1000)

(h) P/S alarm code

G65 H99 Pn ;

Alarm code No. 500 + n

(Example): G65 H99 P15; Alarm code 515 in P/S

Note:

Custom Macro

- 1. If positive numbers are specified as sequence numbers when branching, they are searched forwards first and backwards after that. If negative numbers are specified, search is made backwards first and forwards then.*
- 2. The sequence number of the block can be specified by a variable.*

(Example) **G65 H81 P#100 Q#101 R#102 ;**

When these conditions are specified, processing branches to a point specified in #100.

15.2.4 Notes and Cautions Concerning the User Macro

CAUTION:

As variable values are only integer, in case of returning a number with decimal point, all the digits following the decimal point are truncated.

You have to be very careful when using these arithmetic operations.

(Example):

When #100 = 35, #101 = 10 и #102 = 5, the following operations give the following results.

#110 = #100 / #101 (=3) #111 =
 #110 x #102 (=15) #120 = #100 x
 #102 (=175) #121 = #120 / #101
 (=17)

Notes:

1. How to enter the "#".
 For a standart MDI keyboard when "/#;" is pressed after addresses **G, X, Y, Z, R, I, J, K, F, H, M, S, T or P**, # is entered.
2. A macro instruction can be entered in **MDI** mode. Address data, however, different than **G65** is not displayed when entered by keys.
3. **H, P, Q** and **R** addresses of the macro instruction have always to be entered after **G65**. **N** and **O** addresses can be entered before **G65**.

H02 G65 P#100 Q#101 R#102; Error N100 G65 H01
P#100 Q10; Correct

4. Single block.
 Normally, the block with macro instructions is not interrupted even if a single block is executed. If, however, a parameter is set, the single block can be made valid. This is used for macro tests.
5. The range of the variable numbers is from -2^{31} to $2^{31} - 1$ but the number is not displayed correctly except if it is not from **-99999999** to **99999999**. If the number exceeds the limit, it is displayed with *********.
6. The subprograms can be nested up to 8 levels.

