CNC SYSTEM 10 T

Operator's manual

ETA-17
2004

GENERAL

This manual consists of the following parts:

I. GENERAL

Describes the manual itself and contains notes regarding its use.

II. PROGRAMMING

Describes each function: **NC** format of the function, characteristics and restrictions.

III. OPERATION

Describes the manual and automatic operation of the machine, procedures for input and output of data, editing programs, etc.

APPENDIX:

- Format of **NC** instructions;
- Valid data range;
- Accuracy deviation;
- Command status;
- List of error codes;
- Parameters

This manual does not concern the parameters of the **CNC** system. For the description of the latter refer to the appropriate manual.

All optional functions are described here. For the options which could be applied to your system, look up in the manual of the machine tool builder.

This manual regards the following **CNC** systems:

CNC SYSTEM - 10 **T, software version 3.00 and upper**

GENERAL FLOW OF OPERATIONS WHEN WORKING WITH CNC MACHINE

When you are to make a detail with a machine equipped with **CNC-ETA17**, you have to follow these steps:

1) Make the necessary program in **NC** format form a part drawing. This could be done manually or with appropriate **CAD** software.

*How to prepare the program is described in chapter **PROGRAMMING**.*

2) CNC system will read and execute the prepared program.
If the program could be stored in the free memory of **CNC**, you can load it (manual or through serial port) and execute it afterwards.

The execution could be done through serial port, in the so called DNC mode, if the program is too big or you do not wish the program to stay in program memory. In this case there are no restrictions to the size of the program. After mounting the workpiece and the tools needed, the operation can begin (program execution).

*For detailed information of how to operate the machine refer to chapter **OPERATION**.*

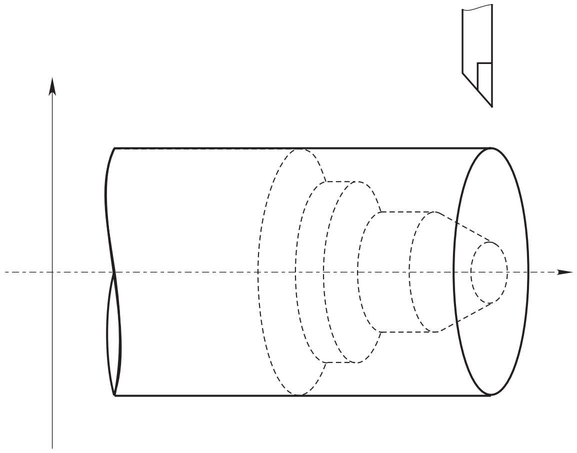
Before the actual programming it's good to make a plan for the machining of the workpiece:

- dimensions and material of the workpiece;

- method of mounting the workpiece on the machine;

- machining sequence in every cutting process;

- cutting tools and modes of operation.



Prepare a program describing the tool path and the conditions needed for the proper machining according to the plan.

NOTES ON USE OF THIS MANUAL

1) ***The abilities of a machine equipped with CNC*** depend not only of the **CNC** itself but on the machine, magnetic cabinet, servo regulators and servo motors, operator’s panel, the control program of the machine supplied by the machine builder, etc. For this reason it is difficult to describe functioning, programming and operating of the system cause of the vast number of combinations.

This manual describes these from the stand point of the **CNC**. So for detailed information regarding a machine equipped with our **CNC** system, refer to the manual issued by the machine tool builder, which should take precedence over this manual.

2) ***User’s programs for machining***, parameters, variables, tool overrides, etc. are stored in internal nonvolatile memory in the **CNC**.

In general the data is not lost when power is OFF. However it is possible to occure a state when all this data has to be deleted in case of failure restoration. In order to rapidly restore the functionality of the machine, it is recommended that you have a copy of this data.

3)***This manual tries to describe***, as it is possible, all situations. However it is impossible to describe all things that should not be attempted, or that could not be done, cause of the vast number of possibilities. For this reason operations that are not described as possible in this manual should be considered as impossible.

1. SAFETY PRECAUTIONS

This chapter describes the safety precautions related to the use of **CNC** devices. It is important these precautions to be observed by users to ensure safe operation of the machines equipped with **CNC** unit (this chapter covers such a configuration).

Note *that some precautions are related only to specific functions and thus they cannot be used with certain **CNC** units.*

Users must observe the safety precautions related to the machine, as described in the manual supplied by the machine tool builder. Before you attempt to operate the machine or create programs, you must be fully familiar with this manual and the manual supplied by the machine tool builder.

This chapter describes all necessary safety precautions, that should be observed, for protecting the operator and preventing damage to the machine.

Safety precautions are classified in the next chapters:

1. GENERAL SAFETY PRECAUTIONS

2. WARNINGS RELATED TO PROGRAMMING

3. WARNINGS RELATED TO HANDLING

4. WARNINGS RELATED TO DAILY MAINTENANCE

1.1 GENERAL SAFETY PRECAUTIONS

1) ***Never*** attempt machining a workpiece without first checking the operation of the machine. Before starting a machining cycle ensure that the machine is working correctly. You can use check functions such as single block, feedrate override or just starting the program without neither a tool nor a workpiece mounted.

If you miss that check, this may result in unexpected behaviour of the machine, possibly causing damage to the workpiece and/or the machine itself, or injury to the operator.

2) ***Before*** operating the machine check all the data entered.

The operation of the machine with improper data may result in unexpected behaviour of the machine, possibly causing damage to the workpiece and/or the machine itself, or injury to the operator.

3) ***Ensure*** that the specified feedrate is appropriate for the intended operation. For each machine there is a maximum allowable feedrate. The appropriate feedrate varies depending on the intended operation. Refer to the manual provided with the machine to determine the maximum allowable feedrate. ***Starting the machine with wrong value for feedrate may result in unexpected behaviour of the machine, possibly causing damage to the workpiece and/or the machine itself, or injury to the operator.***

4) ***When*** using a tool compensation function check very carefully the direction and the value of the compensation. ***Operating the machine with wrong data may result in unexpected behaviour of the machine, possibly causing damage to the workpiece and/or the machine itself, or injury to the operator.***

5) ***The parameters*** of the **CNC** and **PMC** are set by the machine builder. Generally their change is not needed. When however, there is no other alternative than changing a parameter, ensure that you thoroughly understand the parameter function, before making any changes. ***Entering a wrong value for a parameter may result in unexpected behaviour of the machine, possibly causing damage to the workpiece and/or the machine itself, or injury to the operator.***

6) ***Do not touch*** buttons on the **MDI** panel immediately after switching the power on, until **CNC** is not ready (showing a screen with coordinates or screen with errors). Some of the buttons on the **MDI** panel are used for service maintenance or have a specific function. Pushing these buttons may result in changing the mode of the machine to one that is different from normal state. Operation of the machine in such a state could not be predicted.

7) ***This manual*** supplied with a **CNC** unit contains general description of all the machine functions including all optional functions.

Note *that optional functions vary from one machine model to another. Therefore some of the functions described in this manual may not be actually available for a particular model. If in doubt check the specification of the machine.*

8) ***Some*** of the functions may have been implemented at the request of the machine tool builder. When using such functions refer to the manual supplied by the machine tool builder for details of their use and the related cautions needed.

Note:

Non volatile memory of the ***CNC*** *contains different data as programs, parameters, macro variables, etc. When the power is off, this data usually retains. Such data could be deleted inadvertently or because of a malfunction, or a situation requiring deleting it may occur (error recovery). To avoid such situations and assure quick restoration of functionality of the machine and* ***CNC*** *control, make a copy of all vital data and keep it in a safe place.*

1.2 WARNINGS RELATED TO PROGRAMMING

This section covers the main safety precautions related to programming. Before attempting to perform programming read carefully the manual to get known its contents.

1) *Coordinate system setting*

If the coordinate system is not set correctly, the machine may behave unexpectedly, though the program issues correct move commands. Such an operation may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.

2) *Positioning by nonlinear interpolation*

When positioning by nonlinear interpolation is performed (positioning by nonlinear movement between start and end points), the tool path must be carefully determined before performing programming. Positioning involves rapid traverse. If the tool reaches the workpiece while moving, this may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.

3) *Function involving a rotation axis*

Be very careful when programming the speed of the rotation axis. Improper programmed value may cause high speed spinning and thus the centrifugal force will lead the chuck to lose its grip on the workpiece if the latter is not mounted securely. This could bring a damage of the tool, the machine itself, the workpiece or injury to the operator.

4) *Inch/metric conversion*

Switching between inch and metric inputs noes not convert the measurement units of data such as the workpiece origin offset, parameters and current position. Before starting the machine define the measurement units to be used. The attempt to execute a command with incorrect data may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.

5) *Constant surface speed control*

When an axis with a constant surface speed reaches the workpiece, the spindle speed may become excessively high. Therefore it is necessary to determine a maximum allowable speed.

Specifying a incorrect value for this speed may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.

6) *Stroke check*

After turning the power on, the manual reference position return is obligatory. Stroke check can be turned off by pressing buttons **"CAN + P"** when you switch the system on. Note that when stroke check is disabled, an alarm is not issued even if the stroke limits are exceeded. This could lead to damage of the tool, the machine itself, the workpiece or injury to the operator.

7) *Mirror image programming*

Note that programming varies when mirror image programming function is activated.

8) *Compensation function*

If a command based on a machine coordinate system or reference position return is executed in active compensation function, the compensation is temporarily cancelled which leads to unexpected behaviour of the machine.

Before executing some of the following commands, always cancel compensation functions first.

1.3 WARNINGS RELATED TO HANDLING

This section presents safety precautions related to operating with the machine. Before attempting to operating the machine read carefully the manual to get known its contents.

1) *Manual operation*

When operating the machine manually be sure to get known the current position of the tool and the workpiece and ensure that the movement axis, direction and feedrate have been specified correctly. ***Incorrect operation of the machine may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.***

2) *Manual reference position return*

After turning the power on, manual reference position return is obligatory. If the machine is operated without reference position return first, it may behave unexpectedly. ***The unexpected behaviour of the machine may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.***

3) *Manual numeric command*

When issuing a manual numeric command be sure to get known the current position of the tool and the workpiece and ensure that the movement axis, direction and feedrate have been specified correctly. ***Incorrect input data may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.***

4) *Manual handle feed*

In manual handle feed rotating the handle with a large scale factor (for example 100) causes rapid movement of the tool or the table.

Careless use may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.

5) *Change of the coordinate systems*

Do not change the values of the coordinate systems when the machine is under the control of a program. Otherwise the machine may behave unexpectedly which may lead to damage of the tool, the machine itself, the workpiece or injury to the operator.

6) *Workpiece coordinate system shift*

Manual intervention, machine lock function or mirror imaging may shift the workpiece coordinate system. Before attempting to operate the machine under the control of a program, ensure that the coordinate system is set correctly. Otherwise there may occur damage of the tool, the machine itself, the workpiece or injury to the operator.

7) *Manual intervention*

If manual intervention is performed during the operation of the machine, the tool path may change after the programmed operation. Before continuing programmed operation after manual intervention, confirm that the settings of the current coordinates, manual switches, parameters, modal modes (for example absolute/incremental), etc. are correct.

8) *Dry run*

Dry run is used generally to confirm the operation of the machine. In this mode the machine works with speed set by the operator and which is constant. This feedrate differs from programmed feedrate.

Note that the dry run speed may be higher than programmed feedrate.

9) *Tool radius compensation in MDI mode*

Pay careful attention to the tool path set by command in **MDI** mode, because no actual tool radius compensation is performed. When a command which interrupts automatic operation is issued in **MDI** mode and in mode of tool radius compensation, pay attention to the tool path after restoring the automatic operation. For detailed information refer to the chapters describing corresponding functions.

10) *Program editing*

If the machine is stopped and the executed program is edited (with modification, insertion or deletion), the machine may behave unexpectedly after restoring automatic operation under the control of the same program. In other words do not change a program during execution.

1.4 WARNINGS RELATED TO DAILY MAINTENANCE

1) *Memory backup battery replacement.*

While replacing the memory backup batteries keep the power of the **CNC** on and activate the emergency stop of the machine. ***Cause this work is performed with power on and cabinet open, only qualified personnel should perform this work.***

When replacing the battery be careful not to touch high-voltage circuits. Touching uncovered high-voltage circuits presents an extremely dangerous electric shock hazard.

Note:

*The **CNC** uses battery to preserve the contents of its memory, because it must retain data such as programs, tool corrections, parameters, etc. while the external power is off. If the battery voltage drops, **CNC** displays corresponding message on the screen and if the machine tool builder has requested so, an alarm is issued. In this case the battery must be replaced at the moment. Otherwise the contents of the **CNC** memory will be lost. Details on how to perform the replacement refer to the appendix.*

2) *Fuse replacement*

Before replacing a blown fuse there have to be located and removed the cause for blowing. For this reason only qualified personnel can perform this work.

PROGRAMMING

1. INTRODUCTION

In this manual are described the manners for making programs for machining the workpieces using the CNC (Computer Numerical Control) machine tools ETA-17 SYSTEM10
This manual concerns about System Software model **T 3.00** or newer (the lathe's variant). Because the functions of the CNC machine tool do not depend only on the CNC, that's why some functions and operations, described in this manual, could not be execute in practice. In this case you may apply for help to the machine - building plant or to an ETA - 17's specialist.

The firm ETA -17 reserves the right for corrections and future improvements in this manual.

2. CONTROLLED AXES

2.1 Axes Controlled by the System

Mode	Axes	Destination
Manual (MDI)	2 + 1	X, Z + S
Automatic (AUTO)	2 + 1	X, Z + S
EDIT	0	-
Manual (JOG)	2	X, Z
Manual (HANDLE)	2	X, Z
TEACH	2	X, Z

where **MDI, AUTO, EDIT, JOG, HANDLE** and **TEACH** are the names of the machine's modes and **X, Z** and **S** are the names of the positioning axes and the spindle.

2.2 Least Input Increment

It is expressed in the smallest unit, whether mm or inches, of the amount of movement that is being programmed.

2.3 Least Output (command) Increment

It is expressed in the minimum unit of tool motion, either in mm or in inches. Any of the combinations given in the table below may be used.

Input / Output system	Least input increment	Least command increment
mm input / mm output	X: 0.001 mm (Diameter designation) Z: 0.001mm	X: 0.001 mm (Dia.) Z: 0.001 mm
	X: 0.001 mm (Radius designation) Z: 0.001mm	X: 0.001 mm Z: 0.001 mm
inch input / mm output	X: 0.0001 inch (Diameter designation) Z: 0.0001 inch	X: 0.001 mm (Dia.) Z: 0.001 mm
	X: 0.0001 inch (Radius designation) Z: 0.0001 inch	X: 0.001 mm Z: 0.001 mm
mm input / inch output	X: 0.001 mm (Diameter designation) Z: 0.001 mm	X: 0.0001 inch (Dia.) Z: 0.0001 inch
	X: 0.001 mm (Radius designation) Z: 0.001mm	X: 0.0001 inch Z: 0.0001 inch
inch input / inch output	X: 0.0001 inch (Diameter designation) Z: 0.0001 inch	X: 0.0001 inch (Dia.) Z: 0.0001 inch
	X: 0.0001 inch (Radius designation) Z: 0.0001 inch	X: 0.0001 inch Z: 0.0001 inch

For radius designation, select the parameter for **X** axis radius designation.

The increment units of the least command increment are depend on the machine. the choice of measurement units should be selected by establishing beforehand the parameter No. 001 (SCW). The choice of measurement units of the least input increment should be selected by the **G** code.

G20 - least input increment 0.0001 inches
G21 - least input increment 0.001 inches

The state of the system when the power is switched on will be the state of **G20** and **G21** at the time of the power was switched off.

2.4 Maximum strokes

The maximum strokes that can be commanded by this control system are shown in the table below:

mm input mm output	inch input mm output	mm input inch output	inch input inch output
±9999.999mm	±999.9999inches	±9999.999mm	±999.9999inches

3.PREPARATORY FUNCTIONS (G CODE)

A two - digit number following address **G** determines the meaning of the command of the block concerned. The **G** codes are divided into the following two types:

- One - shot **G** codes. The **G** code is valid only at the block in which it was specified.
- Modal **G** codes. The **G** code is valid until another **G** code in the same group is commanded.

Example:

G01 and **G00** are modal **G** codes in 01 group

```
G01 X10. Z10;   }
X20. Z20. F20;  } G01 is effective in this range
X0. Z0. F12.6;  }
G00 X20. 10Z;
```

There are two sets of **G** codes. One is the standard **G** codes and the other are the special **G** codes. Either standard or special **G** codes can be selected by parameter. In this manual, standard **G** code is used. The special **G** code and the corresponding standard **G** code as shown in the following table have the same functions, except **G90** and **G91**.

G90 command specifies absolute dimensions and **X** and **Z** used in **G90** mode are the same as **X** and **Z** respectively under standard **G** code.

G91 command specifies incremental dimensions and **X** and **Z** used in **G91** mode are the same as **U** and **W** respetively under standard **G** code.

In the special **G** code mode, addresses **U** and **W** specify incremental move distance even in **G90** mode.

In **MDI** operation, address **X** and **Z** is effective in absolute command, and **U** and **W** - in incremental command being irrespective of **G90/G91**.

Standard G code	Special G code	Group	Function
* G00	* G00	01	Positioning (rapid traverse)
G01	G01		Linear interpolation (cutting)
G02	G02		Circular interpolation CW
G03	G03		Circular interpolation CCW
G04	G04	00	Dwell
G10	G10		Data setting
G20	G20	06	Inch data input
G21	G21		Metric data input
* G25	* G25	08	Spindle speed fluctoation detect OFF
G26	G26		Spindle speed fluctoation detect ON
G27	G27	00	Reference point return check
G28	G28		Return to reference point
G30	G30		2nd reference point return
G31	G31		Skip cutting
G32	G33	01	Thread cutting
G36	G36	00	Automatic tool compensation X
G37	G37		Automatic tool compensation Z
* G40	* G40	07	Tool nose radius compensation cancel
G41	G41		Tool nose radius compensation left
G42	G42		Tool nose radius compensation right
G50	G92	00	Coordinate system setting, max. spindle speed setting
G65	G65		Custom macro call
G68	G68	04	Mirror image for double turrets ON
* G69	* G69		Mirror image for double turrets OFF

Continue

Standard G code	Special G code	Group	Function
G70	G70	00	Finishing cycle
G71	G71		Stock removal in turning
G72	G72		Stock removal in facing
G73	G73		Pattern repeating
G74	G74		Peck drilling on Z axis
G75	G75		Grooving on X axis
G76	G76		Multiple threading cycle
G90	G77	01	Outer diameter/internal diameter cutting cycle
G92	G78		Thread cutting cycle
G94	G79		Endface turning cycle
G96	G96	02	Constant surface speed control
* G97	* G97		Constant surface speed control cancel
G98	G94	05	Per minute speed
* G99	* G95		Per revolution feed
-	* G90	03	Absolute programming
-	G91		Incremental programming

Notes:

- Maximum spindle speed setting (**G50**) is valid when the constant surface speed control is provided;
- The **G** codes marked with ***** are set when the power is switched on;
- The **G** codes in the group **00** are not modal;
- An alarm occurs when a **G** code not listed in the above table is specified (**PS 010**);
- When a number of **G** codes of the same group are specified, the **G** code specified last is effective;
- **G** code from each group is displayed.

4. INTERPOLATION FUNCTIONS

4.1 Positioning (G00)

G00 specifies positioning. A tool moves to a certain position in the work coordinate system when absolute command or from its current position to the position in a certain distance when incremental command. In both cases positioning is accomplished at the rapid traverse rate.

Format: **G00 X(U)______Z(W)______;**

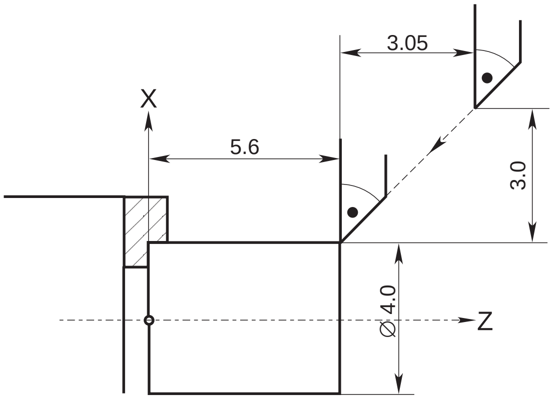
where:

- X(U) and **Z(W)** are absolute (incremental) addresses of axes. Absolute and incremental commands can be used at the same time;

- ";" means end of block (LF for ISO code, CR for EIA code).

The tool path is determined by non linear interpolation type positioning. Positioning is done for each axis separately. Tool path generally does not become a line.

An example of positioning:



(Radius programming)
G00X2.0Z5.6; (Absolute command) or
G00U-3.0W-3.05; (Incremental command)

The rapide traverse rate in the **G00** command is set for each axis independantly by the parameters No. 0518 and 0519. The rapid traverse rate can not be specified in the address **F** when programming.

In the positioning mode actuated by **G00**, the tool is accelerated at the start of the block and is decelerated at the end of the block. If the parameter specifying in-position checking has been set, execution proceeds to the next block after confirming the in-position.

"In-position" means that the axis is positioned in a given range around the programmed position.

4.2 Linear Interpolation (G01)

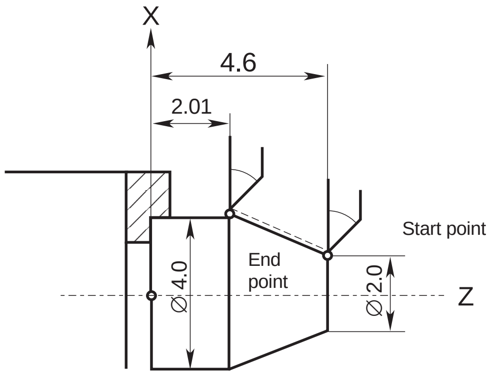
The linear interpolation can be performed by commanding **G01**.

Format: **G01 X(U)_____Z(W)_____F_____;**

X(U) and **Z(W)** are absolute (incremental) commands for movement along the axes.

The tool moves to a certain point in the selected coordinate system along the straight line at the feedrate specified by the F code. Since the feedrate remains effective until a new feedrate is commanded, it need not be respecified. The feedrate which has never been specified by the **F** code is regarded as zero.

Example:



(Diameter programming)

G01X4.0Z2.01 F20; (Absolute command)
G01U2.0W-2.59 F20; (Incremental command)

The feedrate in each axis directions is as follows (in case per minute feed):

G01 U$\underline{\alpha}$ W$\underline{\beta}$ F$\underline{f}$;

feedrate in **X** axis direction: **Fx= α.f/L**
feedrate in **Z** axis direction: **Fz= β.f/L**

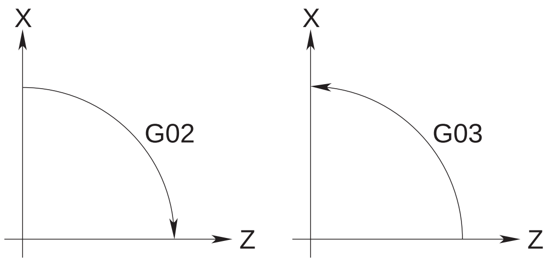
where: $L = \sqrt{\alpha^2 + \beta^2}$

4.3 Circular Interpolation (G02, G03)

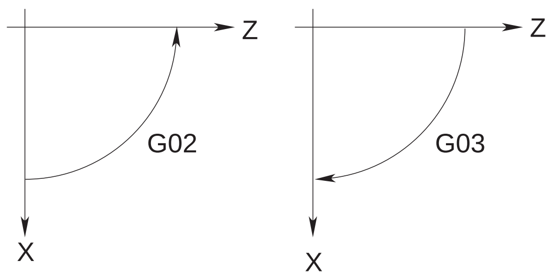
The following command will move the tool along a circular arc.

	Data to be begin		Command	Meaning
1	Rotation direction		G02	Clockwise direction (CW)
			G03	Counterclockwise direction (CCW)
2	End point position	Absolute command	X, Z	End point position in the work coordinate system
		Incremental command	U, W	Distance from start point to end point
3	Distance from start point to center		I, K	Distance from start point to center.
	Radius of arc.		R	Radius of arc..
4	Feedrate		F	Feedrate along the arc.

The clockwise or counterclockwise direction varies in right or left hand coordinate system.

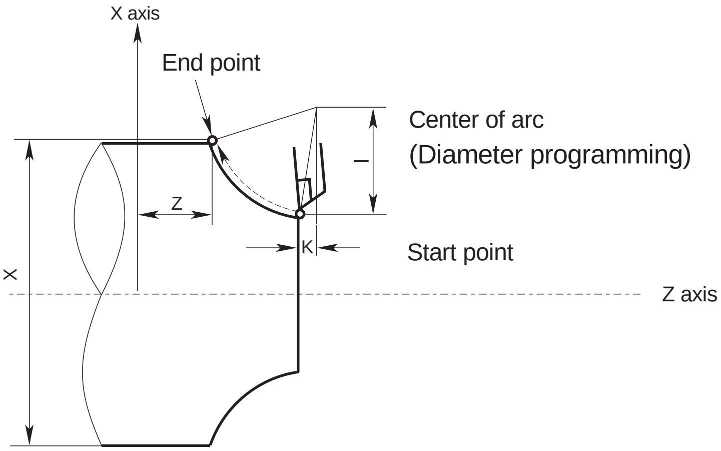


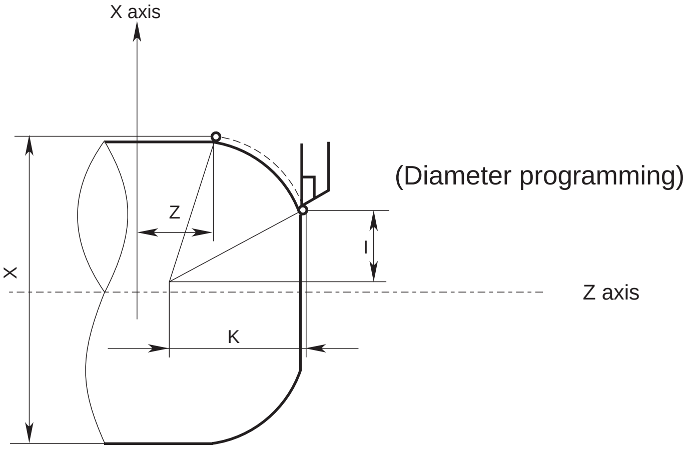
Right hand

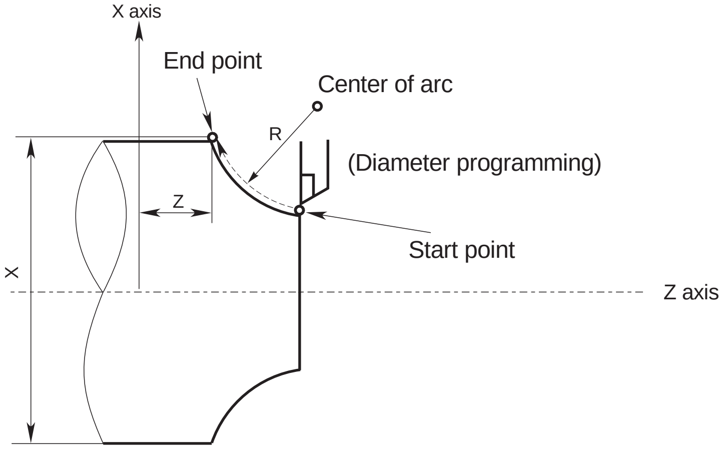


Left hand

Examples:


X axis
End point
Center of arc
(Diameter programming)
I
Z
K
Start point
Z axis
X

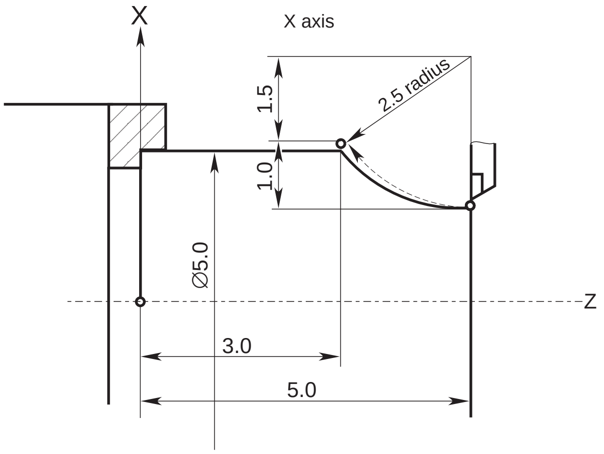

X axis
(Diameter programming)
Z
I
X
Z axis
K


X axis
End point
Center of arc
R
(Diameter programming)
Z
Start point
X
Z axis

(For absolute command)

(Diameter programming)

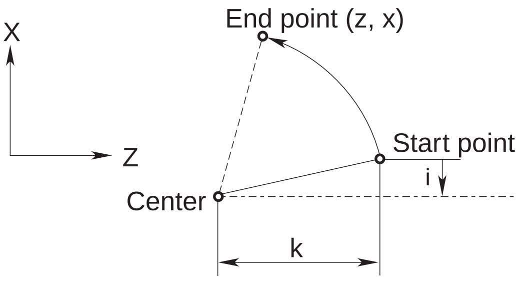
```
     G02X5.0Z3.0I2.5F0.03;
or   G02U2.0W-2.0I2.5F0.03
or   G02X5.0Z3.0R2.5F0.03
or   G02U2.0W-2.0R2.5F0.03
```



The feedrate for circular interpolation is specified by **F** code. The feedrate along an arc is controlled to maintain the specified feedrate.

The determination of the right and inverse direction is held in this way:

The right direction is the direction of rotation from the positive direction of the **X** axis to the positive one of the **Z** axis. The end point of an arc is specified by absolute (incremental) commands **X(U)** and **Z(W)**. The arc center is specified by addresses **I** and **K** for the **X** and **Z** axes respectively. The addresses **I** and **K** can be signed according to the direction. The radius can be specified with address **R** instead of specifying the center by **I** or **K**.



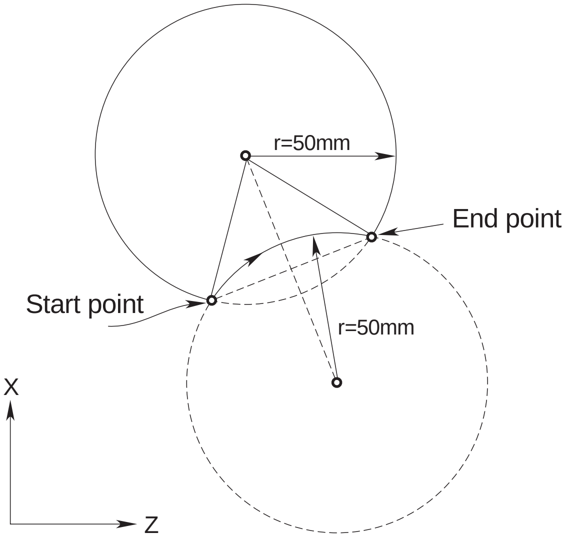
This radius is specified by address **R**. In this case an arc exceeding 180° can not be commanded.

Examples:

For arc (less than 180°)

G02Z60.0X20.0R50.0F300.0;

For arc (greater than 180°)
(Can not be specified in 1 block)



Note: **I0**and **K0** can be ommitted. **X(U)** and **Z(W)** can be omitted if the end point is located at the same position as the start point (a complete circle 360°).

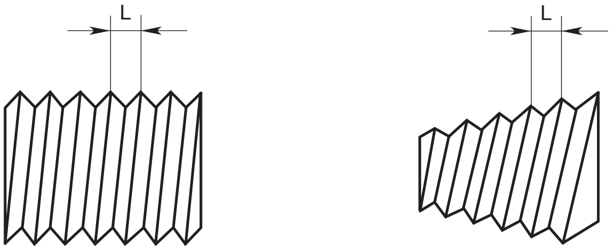
The error between the specified feedrate and the actual tool feedrate is ±2% or less.

If **I, K** and **R** addresses are specified simultaneously, the arc specified by address **R** takes precedence and the others are ignored.

If **I** or **K** is used, the difference in the radius values at the start and end points of an arc does not cause an alarm.

5. THREAD CUTTING (G32)

Straight thread, tapered thread and scroll thread can be cut by using the command **G32**.



Straight thread   Tapered screw

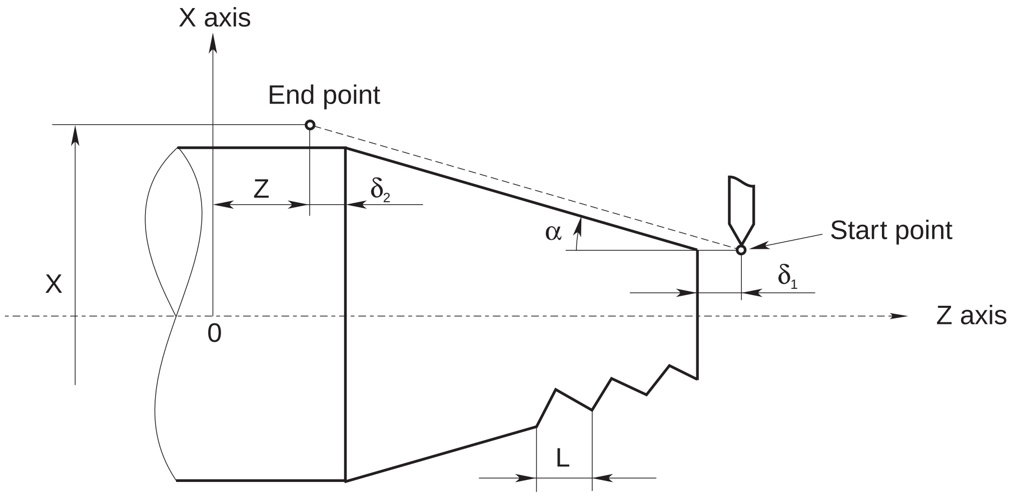
Format: **G32 X(U)_____ Z(W)_____ F_____;**

where:

X(U) and **Z(W)** are the commands of the end point **F** is the the thread lead

In general, thread cutting is made by executing several rough cutting and is ended by finishing cycle. Cutting begins when rotation signal is detected so, when the command is repeated the profile of the detail remains the same. In this case if spindle speed is changed incorrect thread is obtained.

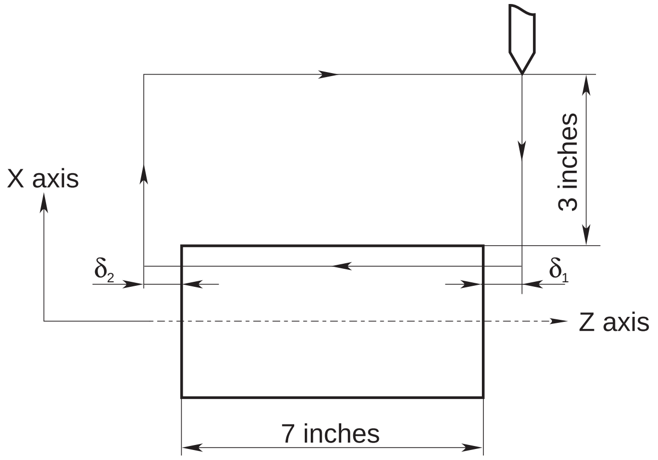
In case of taper thread cutting it is necessary to specify different values for **X** and **Z** axis.



The thread lead must be specified as a radius value.

In general, the lag of the servosystem will produce somewhat incorrect leads at the starting and ending points of the thread cut. Therefore, when specify the thread lenght, it is necessary to specify longer one than the thread to be machined.

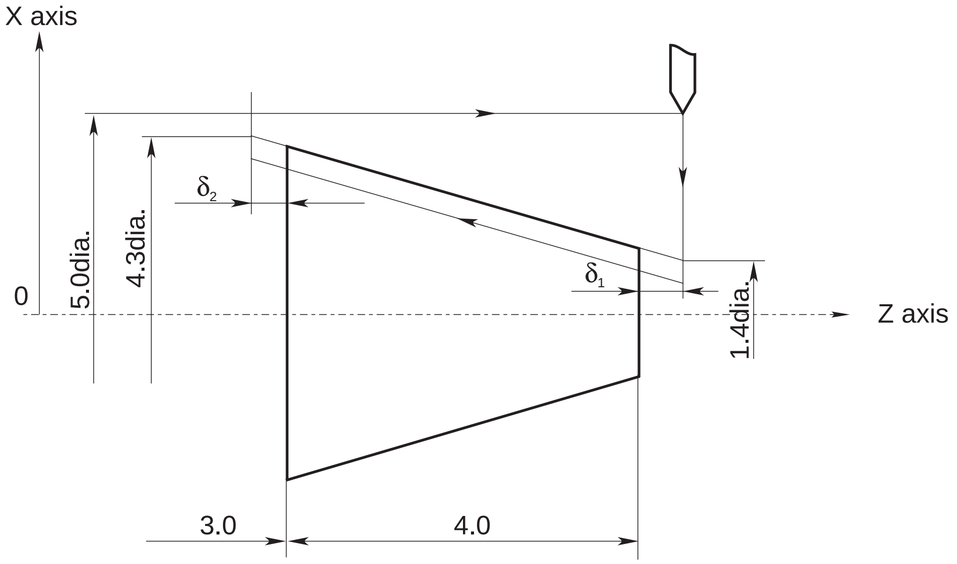
Example: Straight thread cutting



The following values are used in programming:

Thread lead : 0.4 inch
δ_1 : 0.3 inch
δ_2 : 0.15inch
Depth of cut : 0.1 inch (Cut twice)

Example: Taper thread cutting



During the thread cutting feedrate override is fixed at 100%. It is very dangerous to stop feeding the thread cutter without stopping the spindle. This will suddenly increase the cutting depth. That’s why during the thread cutting, the function FEED HOLD executes when there is a block that do not specify thread cutting. The same is valuable for execution block by block. When the mode is changed from automatical operation to manual operation during thread cutting, the tool stops at the first block not specifying thread cutting and then the mode can be changed.

When the previous block was a thread cutting block, it is not necessary during the cutting to check rotation signal from pulse-coder in the current block.

At a constant surface speed the thread cutting may be failured because of changes of spindle speed. Accordingly, constant surface speed control should not be used during thread cutting.

A block, preceding the thread cutting block must not specify chamfering or corner **R**.

Cancellation of the cutting is ineffective during the execution.

6. FEED FUNCTIONS

6.1 Rapid Traverse

Positioning is done in rapid motion by the positioning command **G00**. There is no need to program rapid traverse rate, because the rates are set in the parameters No.0518 and No. 0519.

Rapid traverse rate can be overridden by the switch on the machine operator's panel.

Possible values of the override are: **F0; 25%; 50%; 100%**

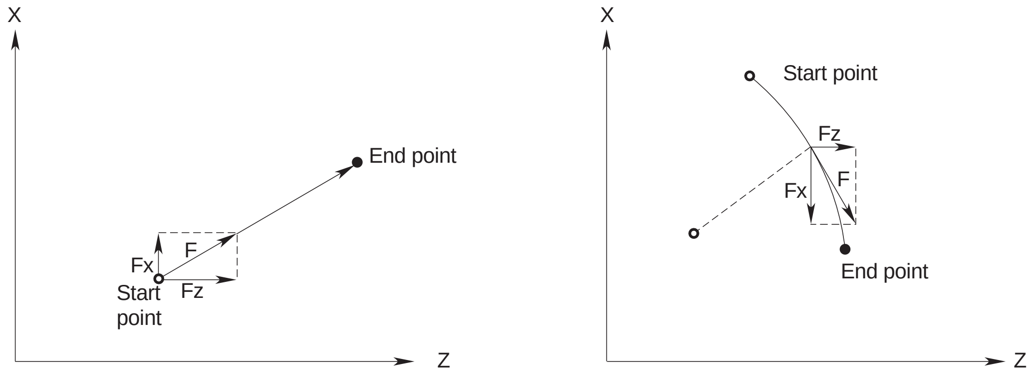
where: **F0** is a constant speed set by parameter No.0533.

6.2 Cutting Feed Rate

Feed speed of linear interpolation (**G01**) and circular interpolation (**G02, G03**) is specified by **F** code.

6.2.1 Tangential speed constant control

The tangential speed is the speed in a current point of the tool path. The speed is constant for the arc along which the movement of the tool is.



For linear

For arc

F : Feedrate of tangent direction
Fx: Feedrate component of X-axis direction
Fz: Feedrate component of Z-axis direction

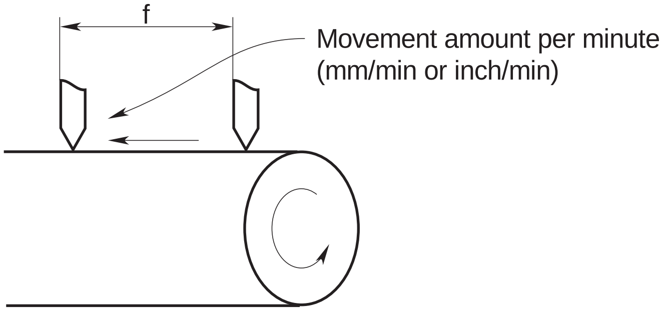
$$F = \sqrt{Fx^2 + Fz^2}$$

6.2.2 Cutting feed rate clamp

The upper limit of the cutting feed rate can be set as parameter No.0527. If the actual cutting feed rate is commanded exceeding the upper limit, it is clamped to a speed not exceeding the upper limit value.

6.2.3 Per minute feed (G98)

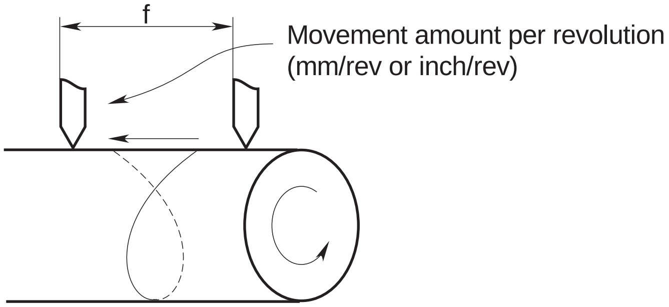
Per minute feed mode is specified by the code **G98**. The tool feed rate is directly commanded by code **F**. Once commanded **G98**, it is effective until **G99** (per revolution feed) is set.



Per minuts feed

6.2.4 Per revolution feed (G99)

Per revolution feed mode is specified by the code **G99**. Following **F**, directly specify the feed of tool per spindle revolution. In this case it is necessary to mount a position coder on the spindle. **G99** is modal. After **G99** is specified, it is effective until **G98** is specfied.



Movement per revolution

The link between per minute feed and per revolution feed is given by the following formula:

$$Fm=Fr . R$$

where:

Fm - per minute feedrate

Fr - per revolution feedrate

R - spindle speed in rpm

The error from the standpoint of the CNC operation with respect to the command value of the feedrate is ± 2%.

After the feedrate has attained its rated value, the time required to move over a distance exceeding 500 mm is measured and the error is calculated.

F code is possible up to a maximum of six digits. When a value exceeding the clamping value of the feedrate, it is clamped at this value.

6.3 Override

6.3.1 Feed rate override

The per minute feed (**G98**) or per revolution feed (**G99**) can be overridden by the switch on the operator’s panel: 0 to 150% (per every 10%).

Feed rate is not overridden to functions as thread cutting in which override is inhibited (it is fixed at 100%).

6.3.2 Rapid traverse override

Rapid traverse rate can be overridden by the switch on operator’s panel.

F0, 25%, 50%, 100%.

F0 - a constant speed can be set by parameter No.0533.

6.4 Automatic Acceleration/Deceleration

6.4.1 Automatic acceleration/deceleration after interpolation

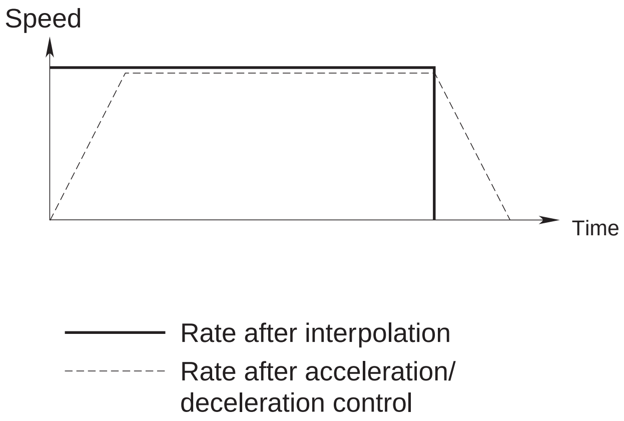
Automatic acceleration/deceleration is performed when starting and ending movement, resulting in smoth start and stop. Automatic acceleration/deceleration is performed also when feed rate changes.

Different kinds of acceleration/deceleration are used:

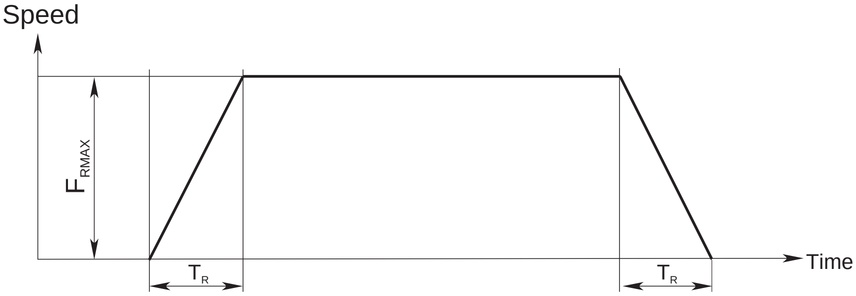
Rapid traverse: **-** Linear acceleration/deceleration . It is set by parameters No. 0522 and No. 0523.

Cutting feed: **-** Exponential acceleration/deceleration. It is set by parameter No. 0529.

Jogging: **-** The same as the Cutting feed.

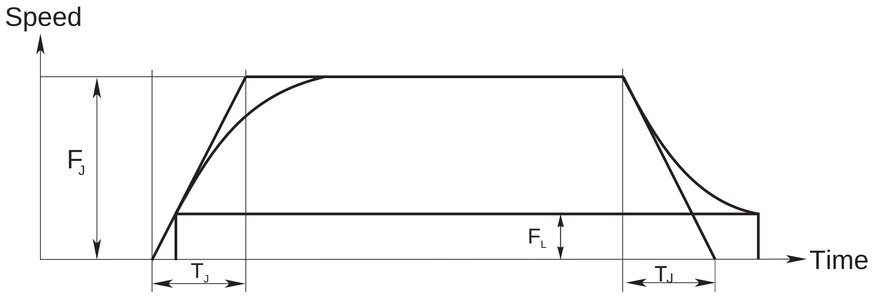


Rapid traverse



F_{RMAX} : Rapid traverse
T_R : Acceleration/deceleration
time constant
(Parameter No.No 522, 523)

JOG feed



F_J: Jog feed rate

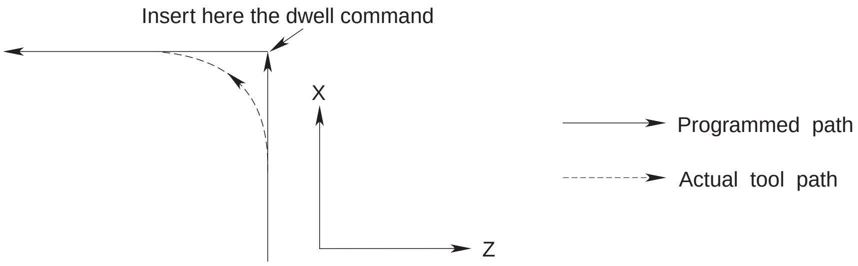
T_J: Jog feed time constant
(Parameter No.529)

F_L: Low feed rate after deceleration
(Parameter No.530)

6.5 Speed Command at Corners

After cutting feed acceleration or deceleration is applied automatically with a time constant so that the machine tool system is not jarred. Because of automatic acceleration and deceleration corners are not cut sharply. It must be inserted some dwell time (**G04**) between the blocks to cut a sharp corner.

In circular interpolation the actual arc radius is smaller than that of the programmed arc. This error can be minimized by making the acceleration/deceleration time constant or feedrate small.



The following chart shows feedrate changes between blocks of information specifying different types of movement.

New block	Previous block		
	Positioning	Feed	Not moving
Positioning	X	X	X
Feed	X	0	X
Not moving	X	X	X

X: The next block is executed after command rate has decelerated to zero.

O: The next block is executed sequentially so that the feedrate is not changed very much.

6.6 Dwell (G04)

Dwell is executed by the following commands:

G04 Xt
G04 Ut
G04 Pt

where: **t** is the dwell time in ms.

The error for for the time t is within 16 ms.

The maximum command time is 9999.999 s.

With address **P**, decimal point can not be used.

Dwell begins after the command feedrate of the previous block attains zero.

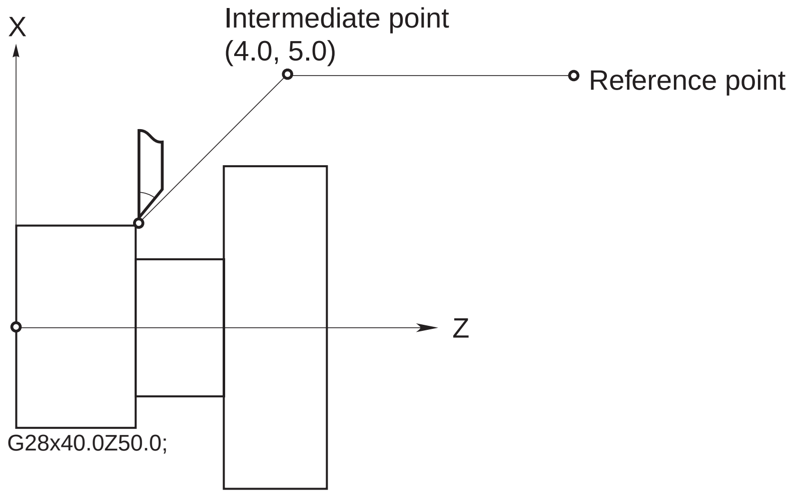
7. REFERENCE POINT

7.1 Automatic Reference Point Return (G28)

The command **G28 X(U)_____ Z(W)_____;**

specifies automatic return to the referent point (**RP**) for the specified axes. **X(U)** and **Z(W)** are intermediate coordinates and are commanded by absolute or incremental values.

Reference point positioning is done with rapid traverse rate of each axes. (Non linear positioning)



In general, this command is used for automatical tool changing (ATC). Therefore, for safety, the tool offset should be cancelled before executing this command.

When the command **G28** is specified and when manual return to the reference point has not been performed after the power has been turned on, the movement from the intermediate point is the same as in manual return to the reference point.

7.2 Reference Point Return Check (G27)

The **G27** command is used to confirm whether or not the tool has reached the reference point.

Format: **G27 X(U)_____ Z(W)_____;**

The tool moves to the specified position at the rapid traverse rate when the above command is used. When the tool reached the reference point the reference point return lamp goes on. If the reference point on the specified axis is not reached, an alarm is displayed.

If an offset has been specified, the position specified by the **G27** command will be shifted.

7.3 Second Reference Point Return (G30)

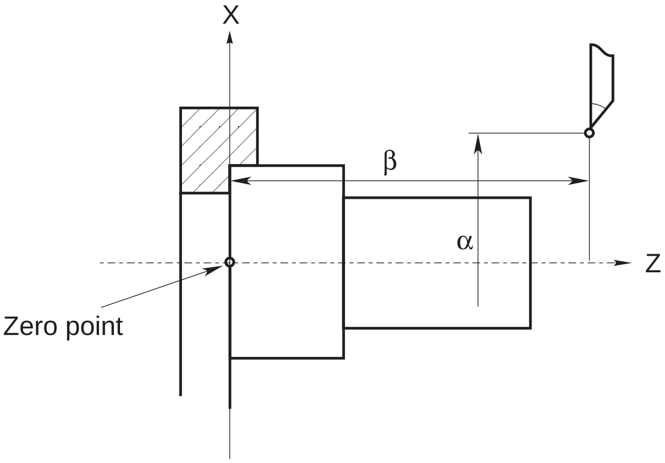
Format: **G30 X(U)____ Z(W)____;**

With the **G30** command, the commanded axis will be positioned to the second reference point. Set the second reference point position from the first reference point as parameters (No. 0735 and No. 0736). This function is available after power is turned on and reference point return is performed. It is the same as reference point return **G28** except tool returns to the second reference point. The **G28** command is usually used when the **ATC** position is different from reference point.

8. COORDINATE SYSTEMS

When tool movement is specified, the position, which must be reached, is designated by coordinate values in a coordinate system. This position is specified by values for each axis. Coordinate values for axes **X** and **Z** are specified as follows:

X___ Z___



Position of tool when Xα Zβ is commanded

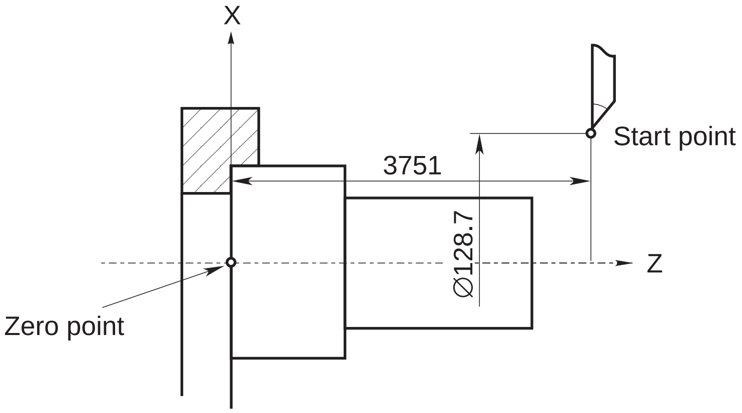
8.1 COORDINATE SYSTEM SETTING (G50)

8.1.1 Coordinate system setting

The setting of a new coordinate system is expressed as follows:

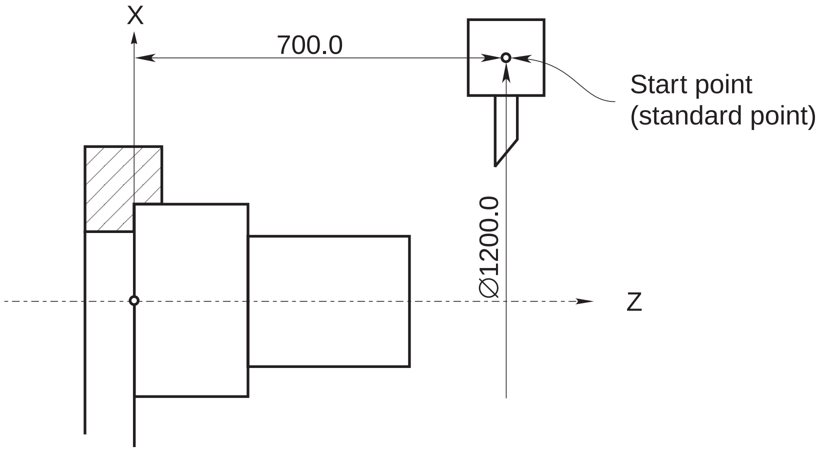
G50 X____ Z____;

After execution of this command, a certain point of the tool - ‘’ the current point ‘’ accepts assigned values for coordinates and all subsequent commands that are commanded become at the position of this new coordinate system. This coordinate system is reffered to as the work coordinate system. The value of **X** is the value of the diameter when diameter designation has been effected, and the value of the raduis when radius designation has been effected.



G50 X128.7 Z375.1; (diameter designation)

Ordinarily, the tip of the cutting edge is aligned with the start point as shown in the illustration above, and the work coordinate system is set in this position.



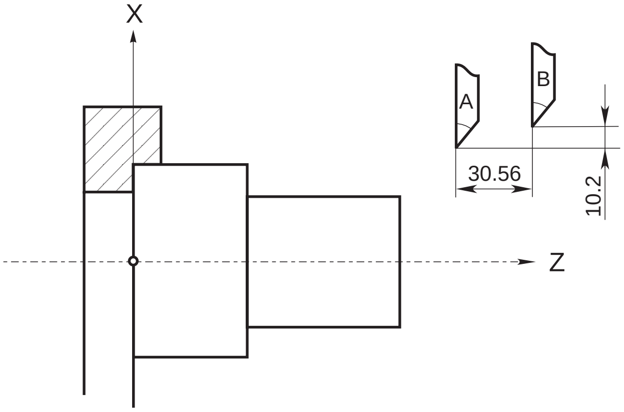
As shown in the illustration above, the reference point on the turret is aligned with the start point and the coordinate system is set at the head of the program. When an absolute command is assigned, the reference point will move to the position commanded. In order to move the tip of the cutting edge to the position commanded, the distance between the reference point and the tip of the cutting edge is compensated by tool offset.

8.1.2 Coordinate system shift

Format:

G50 U____ W____;

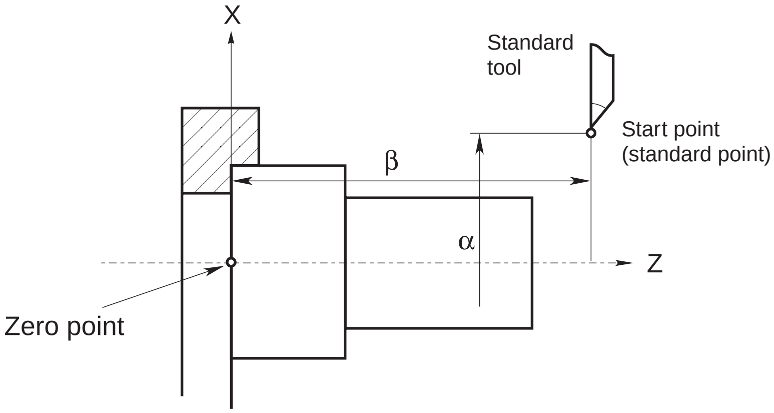
This command creates a new coordinate system which is shifted in comparison with the current one with translation given by **U** and **W**.



8.1.3 Automatic coordinate system setting

When parameter **APRS** (No. 010 bit 7) for automatic coordinate system setting is set in advance, a work is determined automatically at the time of reference point return. This work coordinate system is set by the parameters No. 0708 and No. 0709. The operation is the same as when the following command is designated at the reference point

G50 X____ Z____;



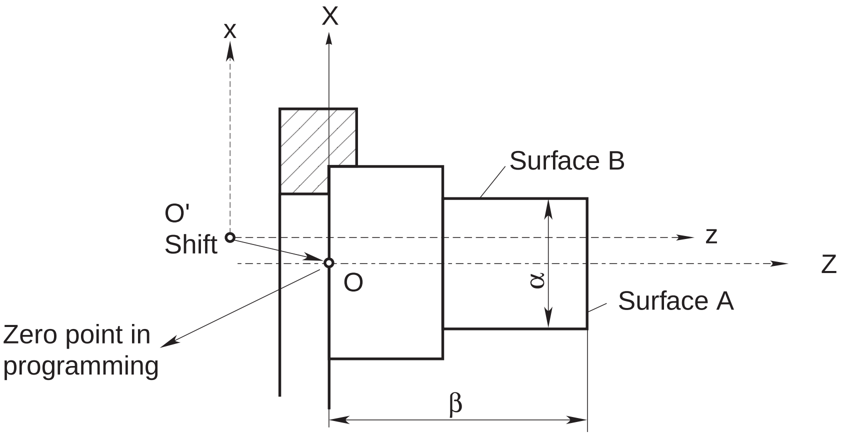
When the tool tip of the standard tool is set as the standard point

8.1.4 Automatic coordinate system shift

Except for the command **G50**, the coordinate system can be shifted by means of setting the values of the variables for shifting of the coordinate system. This kind of coordinate system shift is set by the parameter WSFT (No. 010 bit 6).

8.1.5 Direct measured value input for work coordinate system shift

In case of automatic coordinate system setting or **G50** setting, the coordinate system can be different from the coordinate system, used in a given program. Therefore, the difference can be measured directly as follows:



(1) *Cut the workpiece along the surface A using a standard tool in manual operation.*

(2) *Retract the tool only in the* ***X*** *direction without* ***Z*** *axis movement and stop the spindle.*

(3) *Measure distance* ***β*** *from the zero point in programming to surface* ***A****.*

(4) *Cut the workpiece along surface* ***B*** *by manual operation.*

(5) *Retract the tool only in* ***Z*** *direction without* ***X*** *axis movement and stop the spindle.*

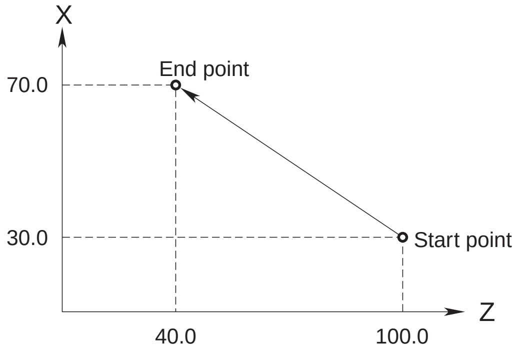
(6) *Measure the diameter* ***α*** *at surface* ***B****.*

(7) *Set the measured values α and β in the variables for coordinate system shift.*

9. COORDINATE VALUES

9.1 Absolute and Incremental Programming

There are two ways to command travels of the axes - the absolute command and the incremental command. Coordinate value of the end point is programmed in the absolute command. In the incremental command, move distance of the axis itself is programmed.



where: **X70.0 Z40.0** is the *absolute* command

and **U40.0 W- 60.0** is the *incremental* command

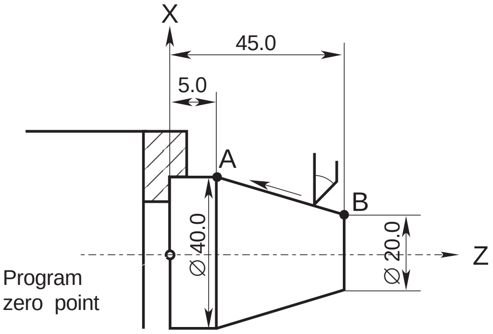
Absolute command	Incremental command	Notes
X	U	X axis move command
Z	W	Z axis move command

For special **G** code, either absolute command or incremental command is commanded in **G90 / G91**.

G90 - absolute command

G91 - incremental command

Example:



Absolute programming: **G90 X70.0 Z40.0;**

Incremental programming: **G91 X40.0 Z - 60.0;**

Command method		Address	Command specifying movement from B to A above
Absolute programming	Specifies an end point in the work coordinate system	X (Coordinate value on the X axis) Z (Coordinate value on the Z axis)	X40.0Z50.0;
Incremental programming	Specifies a distance from start point to end point	U (Distance along the X axis) W (Distance along the Z axis)	U20.0W - 40.0;

Absolute and incremental commands can be used together in a block.

When both **X** and **U** or **Z** and **W** are used together in a block, the one specified later is effective.

9.2 Inch/Metric System Setting.

Metric system can be set by **G21**, and inch system can be set by **G20**. These **G** codes must be specified in an independant block before setting the coordinate system at the beginning of the program.

The following unit systems are changed by **G** code;

Unit system	G code	Least input increment
Inch	G20	0.0001 inch
Millimetre	G21	0.001mm

(1) *Feedrate command by **F** code.*

(2) *Positioning command.*

(3) *Offset value.*

(4) *Unit of scale for manual pulse generator.*

(5) *Some parameters.*

When the power will be turned off, the **CNC** status remains the same. **G20** and **G21** must not be used in one program.

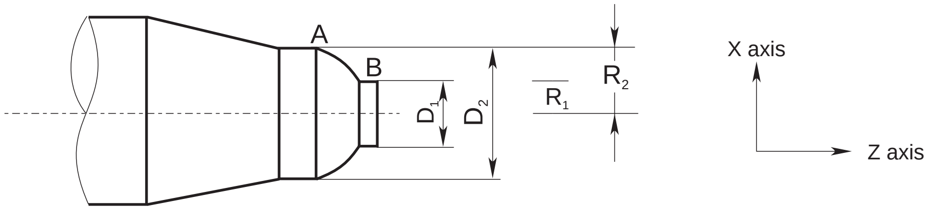
9.3 Decimal Point Programming

This system can input numerical values with a decimal point. However, some addresses can not use a decimal point. A decimal point may be used with mm, inches or second values. The decimal point means multiplication by 1000 for mm and seconds and multiplication by 10000 for inches.

Program command	Usual type decimal point input	Pocket calculator type decimal point input
X1000	1mm	1000mm
X1000.	1000mm	1000mm

9.4 Diameter and Radius Programming

Since the workpiece cross section is usually circular in **CNC** lathe control programming, its dimentions can be specified in two ways: Diameter and radius.



D_1, D_2 . . . . . . Diameter programming
R_1, R_2 . . . . . . Radius programming

The choice of Diameter or Radius Programming can be specified by parameter **XRC** (No. 019 bit 2). When using the diameter programming on the **X** axis, the conditions listed in the following table are valid:

Item	Notes
Z axis command	Specified independently of diameter or radius value
X axis command	Specified with a diameter value
Incremental command with address U	Specified with a diameter value. In the above figure, specifies from D_2 to D_1 for tool path B to A.
Coordinate system setting (G50)	Specifies a X axis coordinate value with a diameter
X component of tool offset value	Parameter setting (No. 0001, ORC) determines either diameter or radius value.
Parameters in G90-G94, such as cutting depth along X axis. (R)	Specifies a radius value.
Radius designation in circular interpolation (R, I, K)	Specifies a radius value.
Feedrate along X axis	Change of radius/rev Change of radius/min
Display of X-axis position	Displayed as diameter value.

10. SPINDLE SPEED FUNCTIONS (S FUNCTIONS)

10.1 Spindle Speed Command

The spindle speed is specified by BCD 2 - digit code signal for CNC spindle and by 5 - digit value for analogue control spindle. In both variants, the speed is specified by **S** - code.

When a move command and a **S**-code are specified in the same block, the commands can be executed in one of the following two ways, depending on the machine tool builder:

(1) *Simultaneous execution of the move command and S-code.*

(2) *Execution of the **S**-code begins after completion of the movement*.

Time constant for **S** code output can be set in parameter No.599. When setting **0** in this parameter **S** code is output immediately.

For details, refer to the manual issued by the machine tool builder.

10.2 Constant Surface Speed Control (G96, G97)

If surface speed (relative speed between workpiece and tool) is set after address **S**, the spindle speed is calculated so that the surface speed is always the specified value in correspondance with the tool position.

The units of surface speed are as follows:

- In case of metric system in m/min
- In case of inches system in feet/min

10.2.1 Command

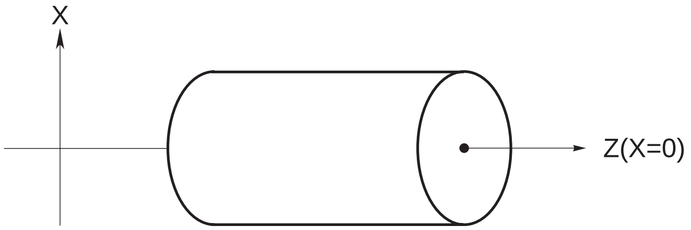
Constant Surface Speed Control is specified by the following command:

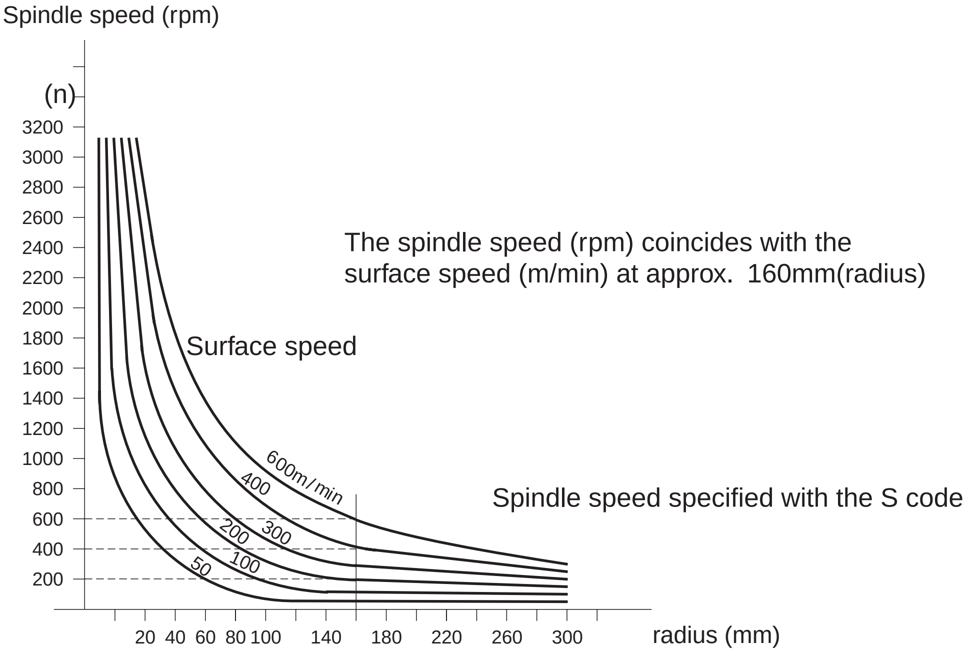
G96 S_____;

Constant Surface Speed Control is cancelled by the following command:

G97 S_____;

When constant surface speed control is used, the work coordinate system must be set so that the center of rotation coincides the **Z** - axis (**X=0**).





Work radius and surface speed in each surface speed

10.2.2 Spindle speed override

An override for the specified spindle speed or surface speed can be selected by the machine panel and it is as follows: 50, 60 , 70, 80, 90, 100, 110 or 120%.

10.2.3 Clamping maximum spindle speed (G50)

Maximum spindle speed is specified by the command:

G50 S____;

10.2.4 Rapid traverse in constant surface speed control

In block, including a **G00** command, the surface speed is not calculated according to the tool position because there is no cutting during the rapid traverse. The constant surface speed is calculated on the end position of the block only.

If the maximum spindle speed is not set when the power supply is switched on, the speed is not clamped.

If the maximum spindle speed is set by a **G92** command, clamping is effective for **G96** only, but it is not effective for **G97**.

G50 indicates that the spindle speed is clamped at 0 rpm.

The value for **S** specified in **G96** mode is not effected by **G97** and is restored when returning in **G96**.

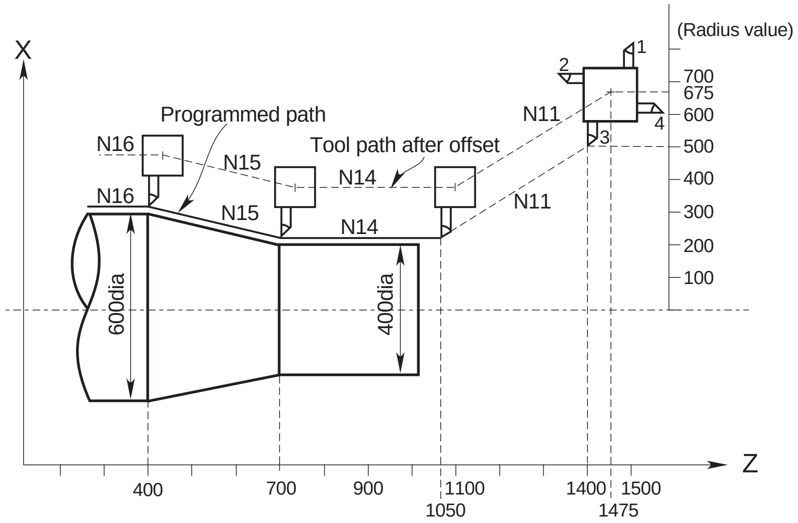
The constant surface speed is calculated even when a machine is operating in ‘’MACHINE LOCK” status.

The constant surface speed control is effective in thread cutting mode. Therefore, it is better to cancel constant surface speed speed control by **G97** before cutting.

In case of switching from **G96** to **G97** mode without specifying the **S** code in **G97**, the value of **S** in the last specifying command remains valid until setting a new **S** code.

In case of switching from **G97** to **G96** the last **S** code, specified in the previous block by **G96** is effective.

The constant surface speed is specified in the programmed path; not to the position where offset value is added to the programmed path.



10.3 Spindle Speed Detection

When the spindle speed deviates from the commanded speed, an overheat alarm is indicated.

The function has the following format:

G26 P____ Q____ R____;

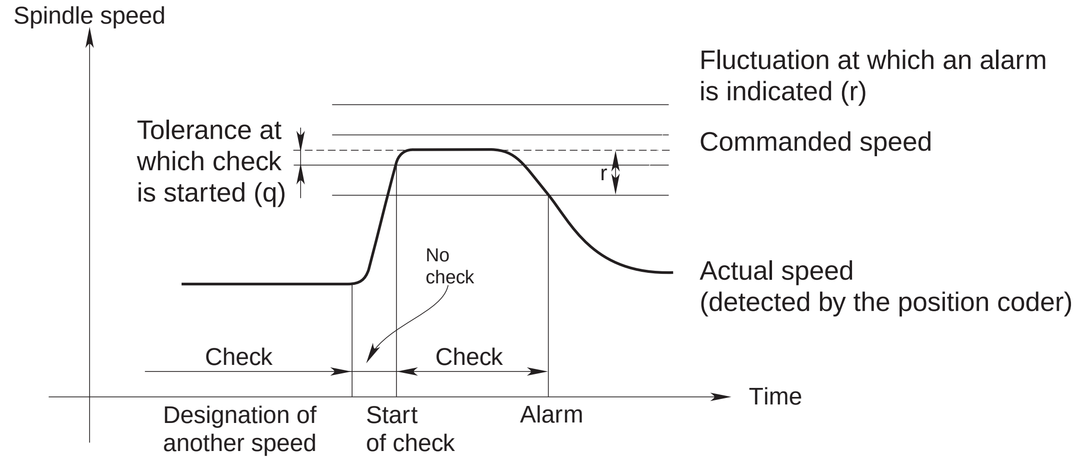
where: **P**: Time in ms for starting check when the commanded speed is not reached after a certain time

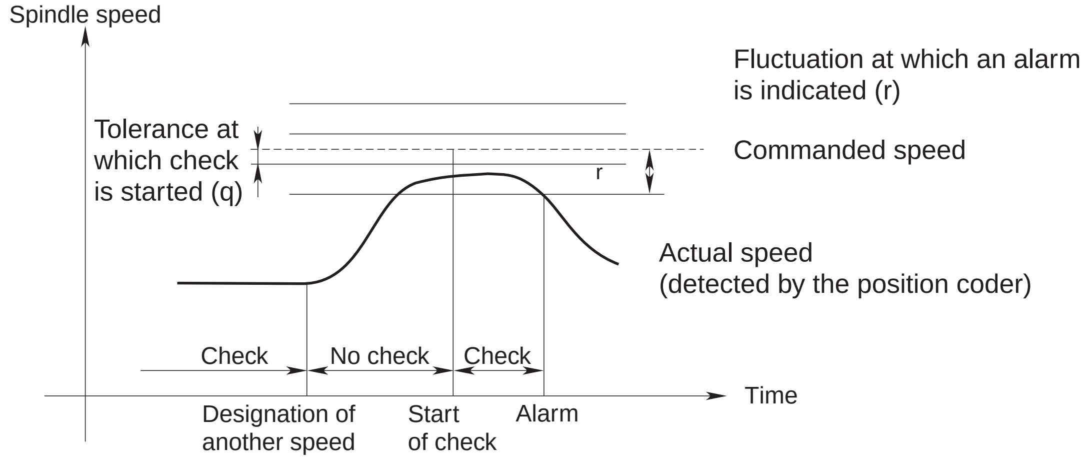
Q: Tolerance (%) at which the actual spindle speed is regarded to reach the command value

R: Spindle speed fluctuation (%) at which an alarm is indicated when the spindle speed changes beyond this value

G26 turns on the detection of the spindle speed fluctuation and **G25** turns it off.

Data **P, Q** and **R** retain even when **G25** is commanded.


Spindle speed
Fluctuation at which an alarm is indicated (r)
Tolerance at which check is started (q)
Commanded speed
r
No check
Actual speed (detected by the position coder)
Check
Check
Time
Designation of another speed
Start of check
Alarm


Spindle speed
Fluctuation at which an alarm is indicated (r)
Tolerance at which check is started (q)
Commanded speed
r
Actual speed (detected by the position coder)
Check
No check
Check
Time
Designation of another speed
Start of check
Alarm

This function is valid only when the constant surface speed control option is selected.

When an alarm occurs during automatic operation, ‘’SINGLE BLOCK STOP” is activated. The spindle overheat alarm is indicated on the TFT.

Even when the reset button is pressed after the alarm, the alarm is indicated again until the cause for the alarm is removed.

The check is not performed in ‘’SPINDLE STOP” state (*SSTP=0)

11. TOOL FUNCTIOS (T FUNCTIONS)

11.1 Tool Selection Function

The tool selection is accomplished by specifying a numerical value following address **T**. A BCD code signal and a strobe signal are transmitted to the CNC machine tool. In one block one **T** code can be commanded.

T code can be set by two or four digits depending on the parameter **T2D** (No. 014 bit 0). When a move command and a **T** code are specified in the same block, the commands are executed in one of the following two ways:

(1) *Simultaneous execution of the move command and **T** function*.

(2) *Execution of the **T** function is accomplished after completion of the command*.

One part of the value after the **T** code indicates the desired tooland the other the offset number. The following two kinds of specifications can be selected:

(1) T code is set by the two digit number and the last one digit designates the offset number

(T__)

(2) T code is set by the four digit number and the last two digits designate the offset number

(T____)

Example: **N1 G00 X1000 Z1400;**

N2 T0313; (select tool No.3 and offset value No.13)

N3 X400 Z1050;

Note: Selected tool corection are executed depend of modal **G** code.

11.2 Handy Tool Life Management

The system counts the number of parts by counting **M** code (**M30** or **M02**) at the program end. When the number of parts reaches a preset value (tool life), the system judges the tool has reached the life.The system counts up the life count, and clears the parts counter. It compensates for the **T** code which has been programmed according to the life count.

11.2.1 Display and setting of data required for tool life management

Select by the keyboard ‘’OFFSET” on the screen. The following parameters are displayed on the screen:

"TOOL LIFE" **-** nuber of parts; at this value the parts number counter is added by 1

"PARTS COUNT (LIFE)" - parts number counter

"LIFE COUNT" **-** Life count frequencies

"PARTS COUT (TOTAL)" - total parts count

11.2.2 Compensation for programmed T code

The programmed **T** code **T00 XX** cosists of tool selection number **00** and tool offset number **XX**. In this case the system executes **T** code **T00XX**.The designation of the **00** and **XX** depends on the Parts number counter.

(1) *Before reaching the first life*

Tool selection number **OO = oo**

Tool offset number **XX = xx**

(2) *After the first life*

Tool selection number **OO=oo+**(tool selection number compensation value)

Tool offset number **XX=xx+**(offset number compensation value)

(3) *After the **N** life*

Tool selection number **OO=oo+**(tool selection number compensation) x N

Tool offset number **XX=xx+**(offset number compensation value) x N

Example:

Parameters: Compensation value of offset number = 8
Maximum value of offset number = 16
Tool selection compensation =10
Maximum value of tool selection number = 99

Program	***After first life***	***After second life***
T0101	T1109	T2101
.	.	.
.	.	.
T0203	T1211	T2203
.	.	.
.	.	.
T0305	T1313	T2305
.	.	.
.	.	.
T0100	T1100	T2100
.	.	.
.	.	.
T0001	T0009	T0001

12. MISCELLANEOUS FUNCTIONS (M FUNCTIONS)

12.1 Miscellaneous functions

M functions are specified by a two digit number and are transmitted to the machine by BCD code. **M** codes are used for turning ON/OFF the control of a machine function. In one block of the program can set only one **M** code. The meaning of the **M** codes depends on the machine tool builder.

When a move command and **M** codes are specified in the same block, the commands are executed in one of the following two ways;

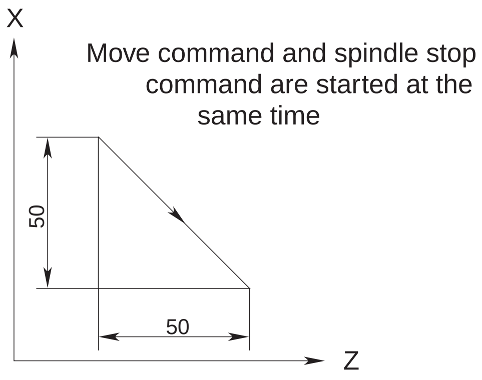
(1) *Simultaneous execution of the move command and **M** function*

(2) *Executing **M** function commands upon completion of the move command execution*

Example:

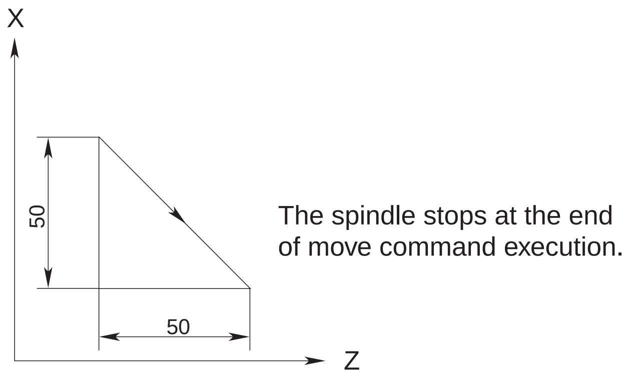
Diameter programming

N1 G01 X -100.0 Z50.0 M05;



Move command and spindle stop command are started at the same time

Sequence i



The spindle stops at the end of move command execution.

Sequence ii

The selection of either sequence depends on the machine tool builder's specifications.

The following **M** codes are used by the system:

(1) M02, M30 : *End of program*

These codes indicate the end of the main program.

(2) M00 : *Program stop*

Cycle operation stops after a block containing **M00**. When the operation is stopped, all executing modal information remains uncharged and the execution can be continued by pressing the key "START". The operation of the **M00** is the same as the key "SINGLE BLOCK".

(3) M01 : *Optional stop*

The operation of the **M01** is similarly to **M00**, with this difference that the execution depends on the position of the ''OPTIONAL STOP SWITCH" on the machine's panel.

(4) M08 : ***M90** keeps the machining down*

from the controller program from the **T-**code in the same sentens. This **M-**code depends on the parameter M90ENB(P014 bit 7). When **M90** is set, the **T-**code in the current sentence is processed only from the base software.

(5) M98 : *Calling of subprogram*

This code is used for calling of subprogram.

(6) M99 : *End of subprogram*

This code indicates the end of subprogram. Executing **M99** returns control to the main program.

If there is a block following **M00, M01, M02** or **M30**, it is not read into the buffer storage. Similarly, the blocks after **M** codes can not be entered into the buffer storage, set by parameters No.111 and No.112.

When executing **M90, M98** or **M99** a BCD code and a strobe signal are not transmitted.

All **M** codes except for **M90, M98** and **M99** are processed by the machine tool. For details refer to the manual issued by the machine tool builder.

13. PROGRAM CONFIGURATION

A list of commands to the **CNC** for controlling the machine is called a program. The list of commands is called a block (sentence). Blocks can be numbered. The program consists of blocks, which are executed one after another.

A program consists of the following parts:

(1) *Beginning of the program*
(2) *Programmed blocks*
(3) *End of program*

Examplary program stored in a file:

```
%
O0001;
G28 X0 Z0 (zero return);
G00 X10. Z10.
G98 F500 (work feedrate);
G01 W-20.;
U-5 ;
W20. ;
M02;
%
```

13.1 Beginning of the program

The beginning of the program is specified by the following symbol block:

% xxxxxLF CR

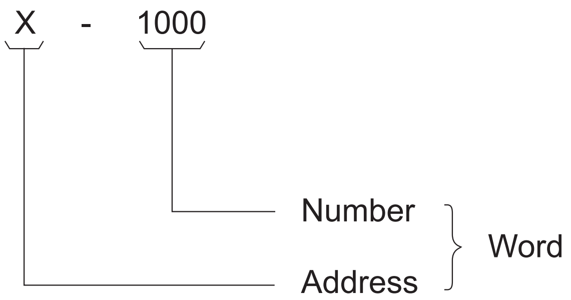
where:

LF is LINE FEED in ASCII standard
CR is CARRIAGE RETURN in ASCII standard

The symbols between **‘’%”** and **LF** are not valid and they are ignored (skipped) at the reading of the program.

13.2 Programmed Block

The blocks consist of valid programmed words and **/** or blocks of comments, completing with the symbol for end of block (**" ; "** in **ISO**).



The valid programmed codes are as follows:

Function	Address	Meaning
Program number	O	Program number
Seguence number	N	Sequence number
Preparatory function	G	Designates mode of function (such as straight line, etc)
	X, Z, U, W	Move command in each axis
	P	Canned cycle taper radius difference and radius in corner R.
	C	Chamfering amount
	I, K	Coordinate of arc center
Feed function	F	Designating of feedrate, designating of thread lead
Spindle speed function	S	Designating of spindle speed
Tool function	T	Designating of tool number, specifying of tool offset number
Miscellaneous function	M	Designating of on/off control at machine side
Dwell	P, U, X	Designating of dwell time
Designation of program number	P	Designating of subprogram number
Designation of sequence number	P, Q	Designating of sequence numbers of portions of program that are to be repeated
Repetitive count	P	Number of repetitions of a subprogram

One and the same code can have more of one meaning depending on the block in which it takes part or on the designation parameters.

Each code has a range of command values.

Basic addresses and range of command values.

Function	Address	Input in mm	Input in inch
Program number	O	1-9999	1-9999
Sequence number	N	1-9999	1-9999
Preparatory function	G	0-99	0-99
	W, Z, U W, R, C, A, I, K	±9999.999mm	±999.9999inch
Feed pat minute	F	1-15000mm/min	0.01-600.00inch/min
Feed per revolution, thread lead	F	0.0001-500.0000mm/rev	0.000001-9.999999inch/min
Spindle function	S	0-9999	0-9999
Tool functions	T	0-9932	0-9932
Miscellaneous function	M	0-99	0-99
Dwell	X, U, P	0-9999.999sec	0-9999.999sec
Designation of sequence number, number of repetetitions	P	1-9999999	1-9999999
Designation of sequence number	P, Q	1-9999	1-9999

Each block can have sequence numbers. The sequence number can be designated by the address N and four digit number, specified in the beginning of the block.

Example:

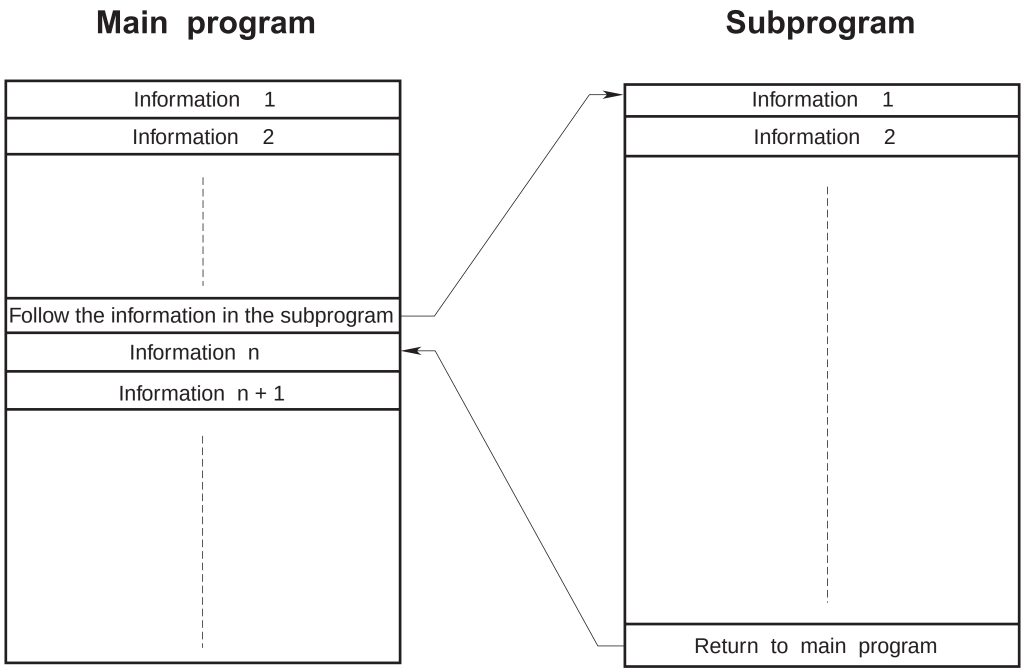
N0010 G01 X4. Z0.2;

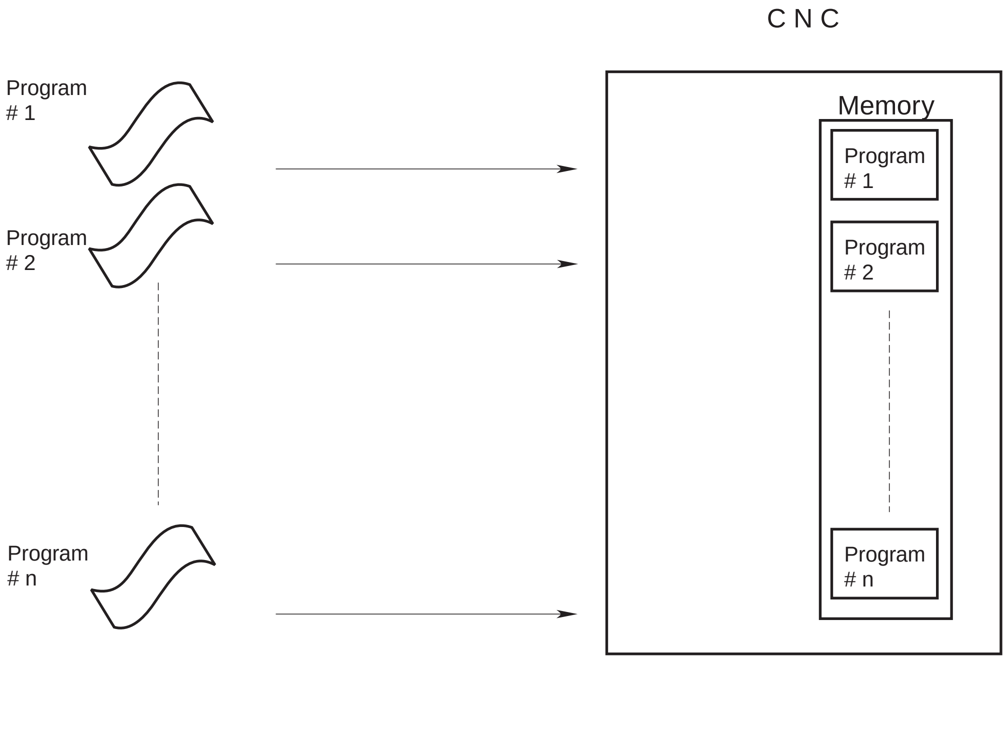
13.3 Disposition of the Programs in the Memory

Until 512 programs can be stored in the memory of the system. These programs can be main programs and subprograms. Each program begins with address **O** and the number of the program (four digit number):

Oxxxx

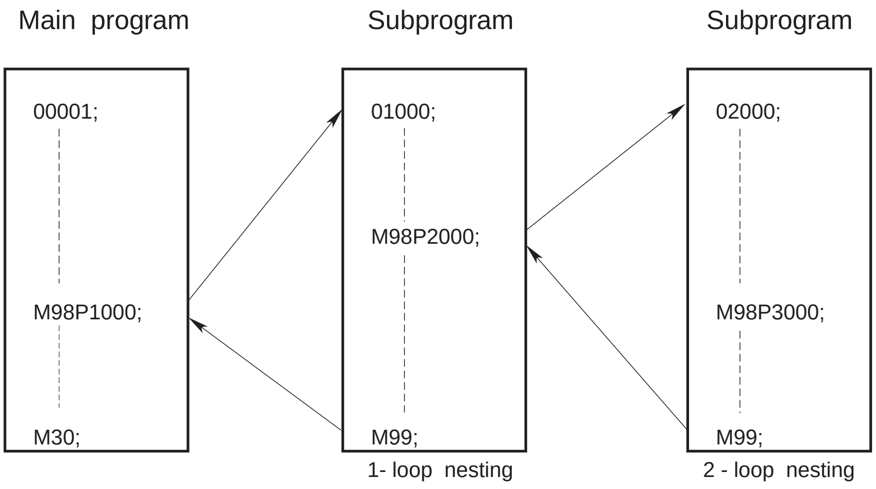
where: **xxxx is 1 – 9999 number**


Main program
Subprogram
Information 1
Information 2
Follow the information in the subprogram
Information n
Information n + 1
Information 1
Information 2
Return to main program


C N C
Program # 1
Program # 2
Program # n
Memory
Program # 1
Program # 2
Program # n

13.4 Subprogram

The subprograms are used to describe the frequently repeated actions or the executing of the one and the same operation with different parameters.



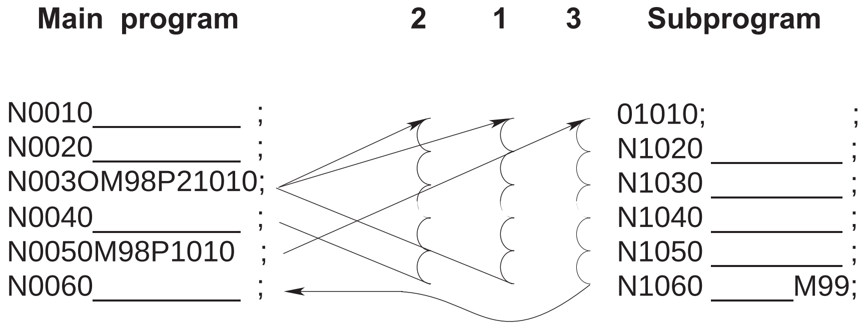
13.4.1 Subprogram calling

To call the certain subprogram, designate as follows:

```
. . .

M98 Pxxxnnn;

. . .
```

It must be specified the code **M98** and in the address **P** must be designate the number of the subprogram (**nnn**) and number of call repetitions (**xxx**) of the subprogram. If the repetition is not set, the subprogram is executed only once. Calling of the subprogram can be designated in the same block as the movement command. The subprogram can be called after the movement is finished. Another subprogram can be called from the parent subprogram in the same way as calling the parent subprogram from the main program. When the program number designated with address **P** is not found, alarm No.78 is indicated. It is impossible to call the subprogram by designating **M98 Pxxxx** from the **MDI** mode.

Example:



13.4.2 Subprogram return

The end of a subprogram is designated by the following command:

M99 Pnnnn;

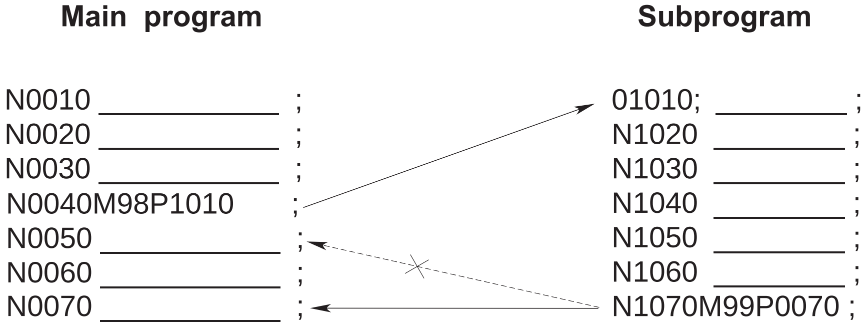
where the designation of the **P** code is not necessary

When **P** code is designated, control returns to the start of the main program and continues from the block whose number is **Nnnnn**. If **P** code is not designated, control returns to the main program and execution continues from the block just next after the calling of the subprogram.

When **M99** is specified in the main program, the execution starts from the beginning of the program.

When **M99 Pnnnn** is specified in the main program, the execution continues from the block whose sequence number is **Nnnnn** in the current program.

Example:



13.5 Comment

The program's comments are designated by the following format:

```
(This is a simple text);
```

The comments start by the symbol **"("** and finish by the symbol **")"**. The comnments can contain random symbols from **0** to **127** in **ASCII** standard.

The symbols in comments do not influence to the execution of the program and the status of the machine. They are used for more clearness in case of examination and checking of the programs. More than one comment can have in a block. Nested comments are not allaved.

13.6 Optional Block Skip

A block is subordinate when a slash followed by a number is specified at the beginning of a block.

A block with a slash (**/**) is executed or not, depending on the position of the "OPTIONAL BLOCK SKIP" switch. When the optional block switch is set **OFF**, the block is valid. If this switch is set **ON**, the block with a slash (**/**) is skipped.

Example:

```
/ N100 X100;
```

when the symbol **"/"** is not at the beginning of the sentance, the addresses before it are always valide, while the addresses after it become dependent on "OPTIONAL BLOCK SKIP".

13.7 End of program

The end of a program is indicated with the symbol **" % "**.

14. FUNCTION TO SIMPLIFY PROGRAMMING

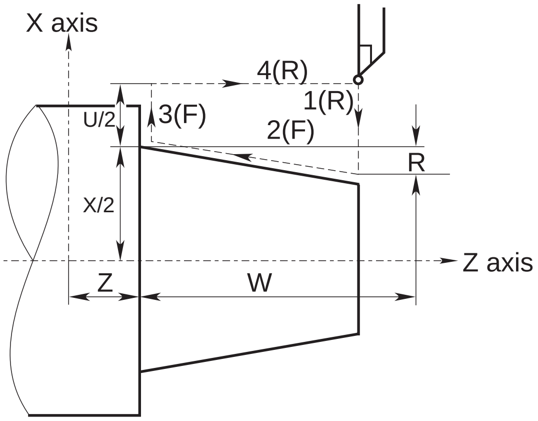
For repetitive machining peculiar to turning, such as metal removal in rough cutting, a series of paths usually specified in a range of several blocks can be specified in one block. For such operations can be used program cycles with suitable parameters.

14.1 Canned cycles

14.1.1 Outer/internal diameter cutting cycle

This cycle is specified with the following command:

G90 X(U)____ Z(W)____ R_____ F____;

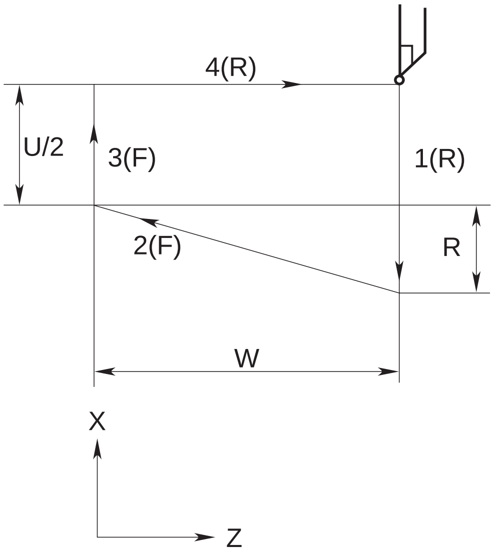


where: **R** - rapid traverse

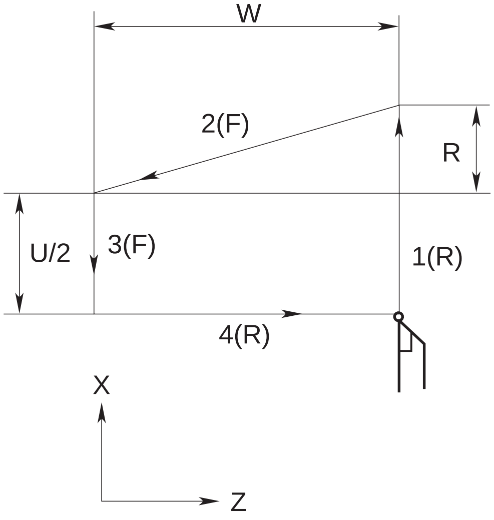
F - specified by **F** code

Depending on the signs of **U** and **W**, there are four cases.

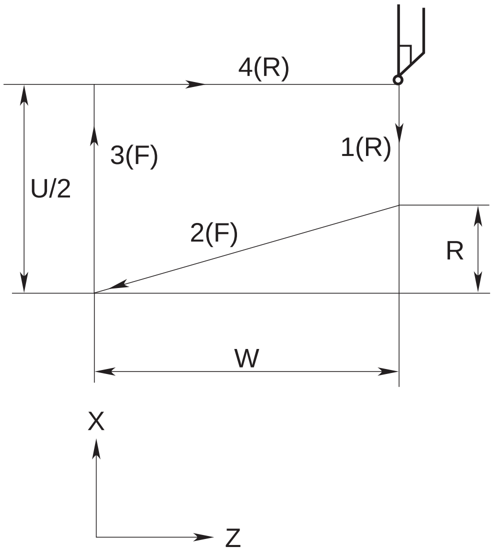
1) U < 0, W < 0, R < 0



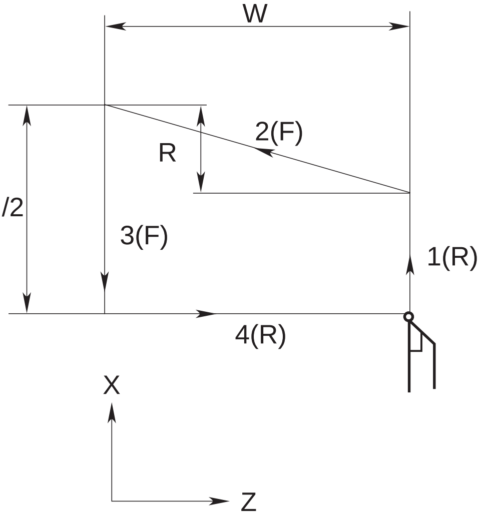
2) U > 0, W < 0, R > 0



$$3)\mathbf{U} < 0, \mathbf{W} < 0, \mathbf{R} > 0, \mathbf{at}|\mathbf{R}| \le \left|\frac{\mathbf{U}}{\mathbf{2}}\right|$$



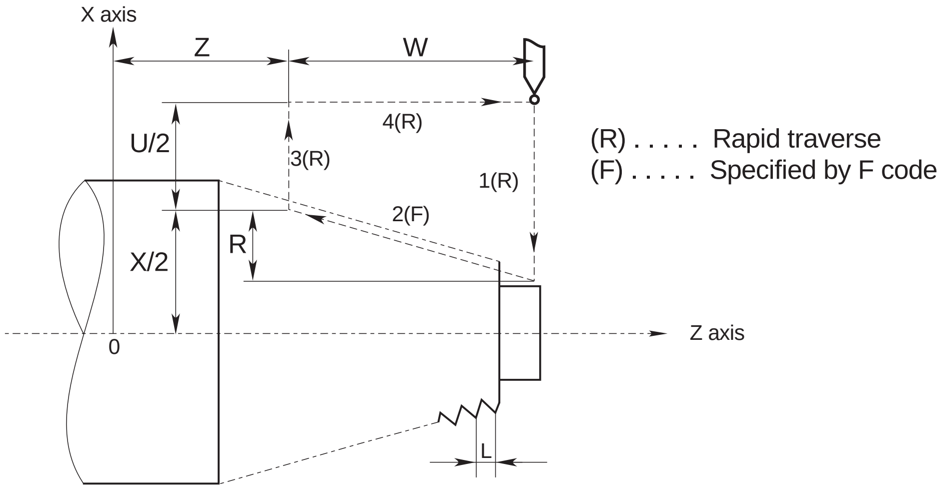
$$4)\mathbf{U} > 0, \mathbf{W} < 0, \mathbf{R} < 0, \mathbf{at}|\mathbf{R}| \le \left|\frac{\mathbf{U}}{\mathbf{2}}\right|$$



14.1.2 Thread cutting cycle (G92)

This cycle is specified by the following command:

G92 X(U)____ X(W)____R____ F____;


X axis
Z
W
U/2
4(R)
3(R)
1(R)
2(F)
R
X/2
0
Z axis
L
(R) . . . . . Rapid traverse
(F) . . . . . Specified by F code

where: **R** - rapid traverse

F - specified by **F** code

L - thread cutting

In incremental programming the cases are the same as in **G90**. The range of thread leads, limitation of spindle speed are the same as in **G32** (thread cutting). In the "SINGLE BLOCK" mode, the whole block is performed without break.

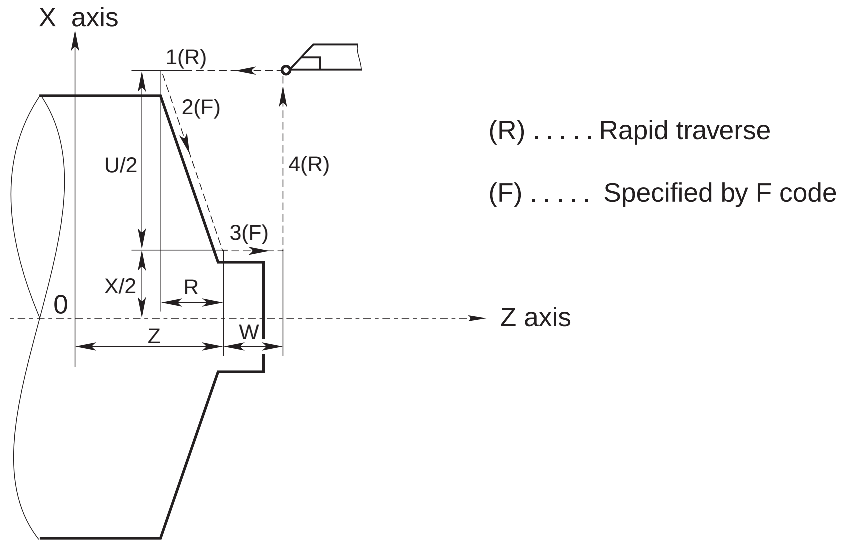
14.1.3 End face turning cycle (G94)

The face cutting cycle is specified by the following command:

G94 X(U)____ Z(W)____R____ F____;

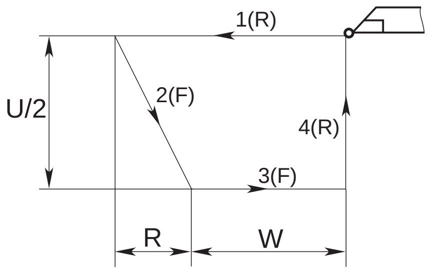
where: **R** - rapid traverse

F - specified by **F** code

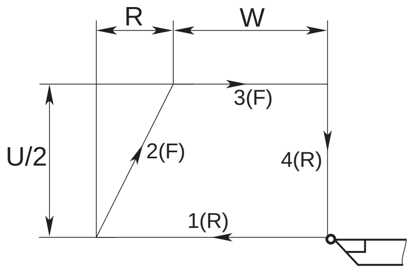

X axis
1(R)
2(F)
U/2
4(R)
3(F)
X/2
R
0
Z axis
Z
W
(R) . . . . . Rapid traverse
(F) . . . . . Specified by F code

In incremental programming the following cases are considered:

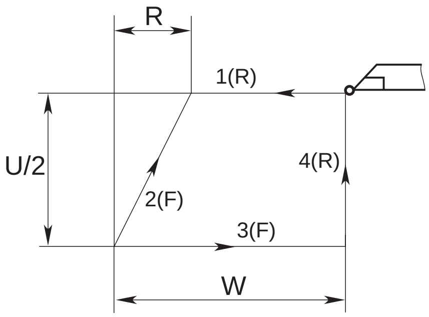
1) $U < 0, \; W < 0, \; R < 0$



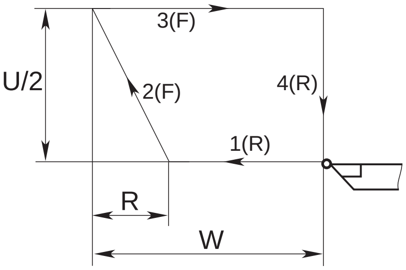
2) $U > 0, \; W < 0, \; R < 0$

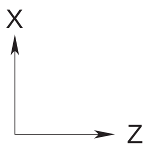


3)$U < 0, W < 0, R > 0, \text{at} |R| \leq \left|\frac{U}{2}\right|$



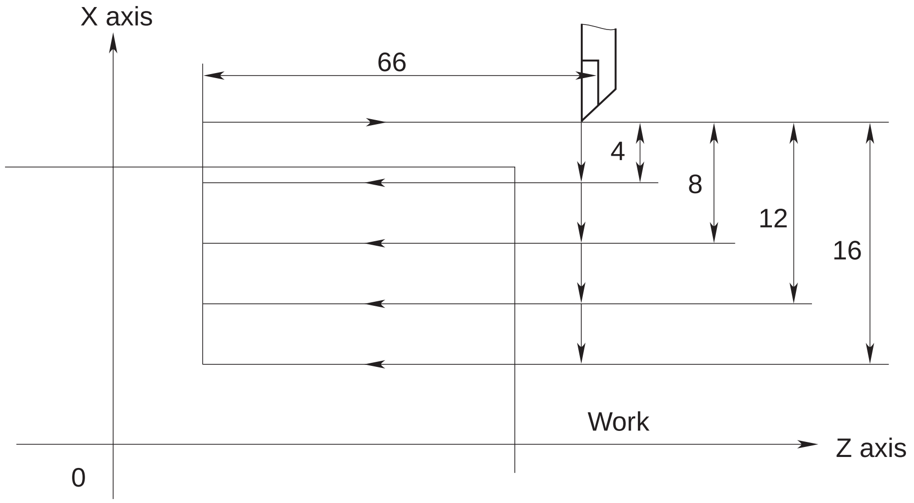
4)$U > 0, W < 0, R > 0, \text{at} |R| \leq \left|\frac{U}{2}\right|$





In general, for the canned cycles:

- when the button **"FEED HOLD"** is pushed, the canned cycle is not executed until its end, and the tool is taken out, returned to the start point and then the movement stopped.



- the data values of **X(U)** and **Z(W)** in case of modal **G90, G92** or **G94** are saved, so when repeating the cycle in one of the axes only it is not necessary to specify the rest addresses.

```
N030 G90 U-8000 W-66000 F400;
N031  U-16000;
N032  U-24000;
N033  U-32000;
```

- by specifying a canned cycle in the **MDI** mode, and pushing the [**INPUT**] button, the same cycle will be performed again.

- if the **M, S** or **T** function is commanded during the canned cycle mode, the **M, S** or **T** function and canned cycle can be performed simultaneously. If this is inadmissible, in specifying the **M, S** or **T** functions **G00** or **G01**are used to cancel the canned cycles.

Example:

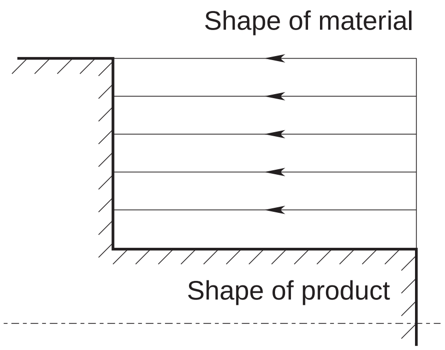
```
N010 G90 X20000 Z10000 F200;
N011 G00 T0201;
N012 G90 X20500 Z1000;
```

14.1.4 Usage of canned cycle

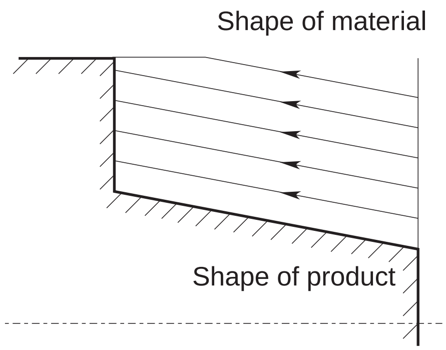
After ....

An appropriate canned cycle is selected according to the shape of the material and the shape of the product.

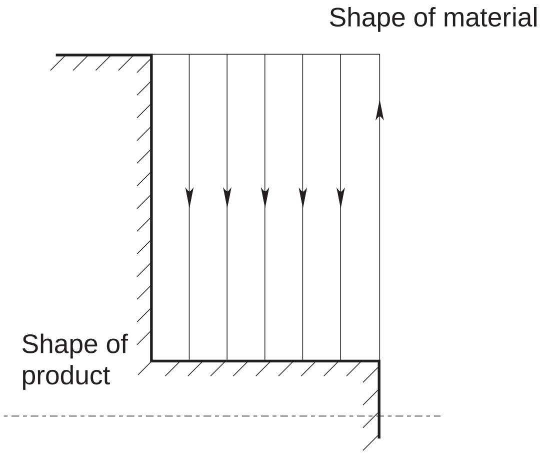
1. Straight cutting cycle



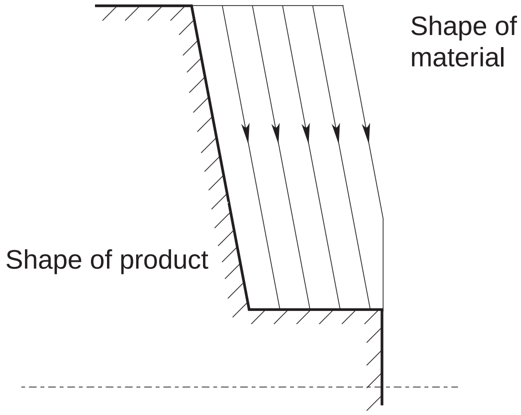
2. Taper cutting cycle



3. Face cutting cycle



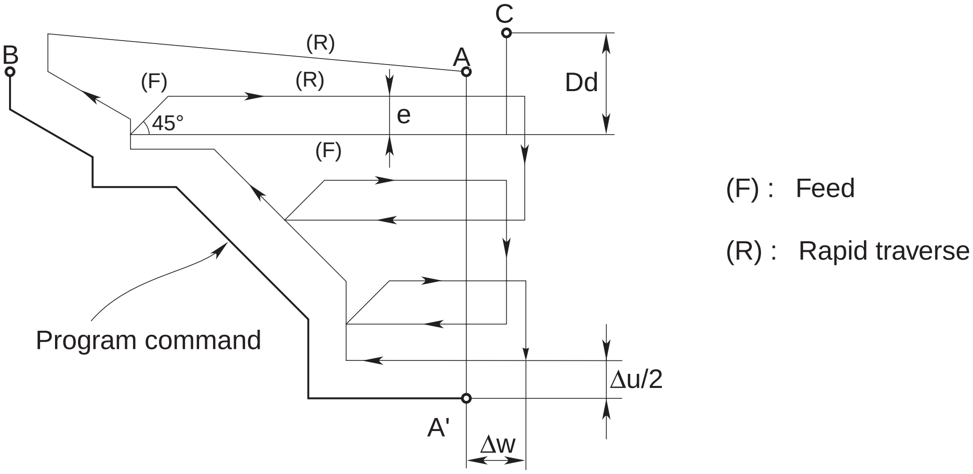
4. Face taper cutting cycle



14.2 Multiple Repetitive Cycle (G70 to G76)

14.2.1 Stock removal in turning (G71)

If a finished shape of **A** to **A'** to **B** is given by a program as in a figure below, the specified area is removed by **Δd** (depth of cut), with finishing allowance **Δu/2** and **Δw** left.



G71 U(Δd) R(e);

G71 P(ns) Q(nf) U(Δu) W(Δw) F(t) S(s) T(t);

where:

Δd: depth of cut. Designate without sign. This designation is modal and is valid until onother value is designated. Also this value can be specified by the parameter No.717, and the parameter is changed by the program command

e: escaping amount. This designation is modal and valid until onother value is designated. Also this value can be specified by the parameter No.718 and the parameter is changed by the program command.

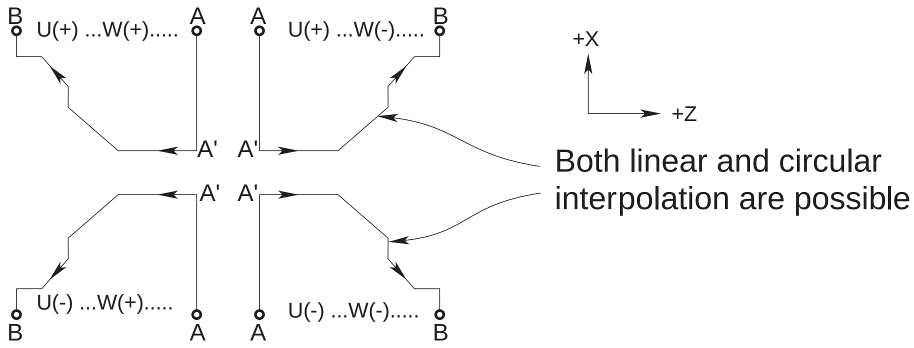
ns: sequence number of the first block from the program of finishing shape

nf: sequence number of the last block from the program of finishing shape

Δ**u**: distance and direction of finishing allowence in **X** direction

Δ**w**: distance and direction of finishing allowence in **Z** direction.

The following four cutting patterns are considered. All of these cutting cycles are made parallel to **Z** axis and the sign of Δ**u** and Δ**w** are as follows:



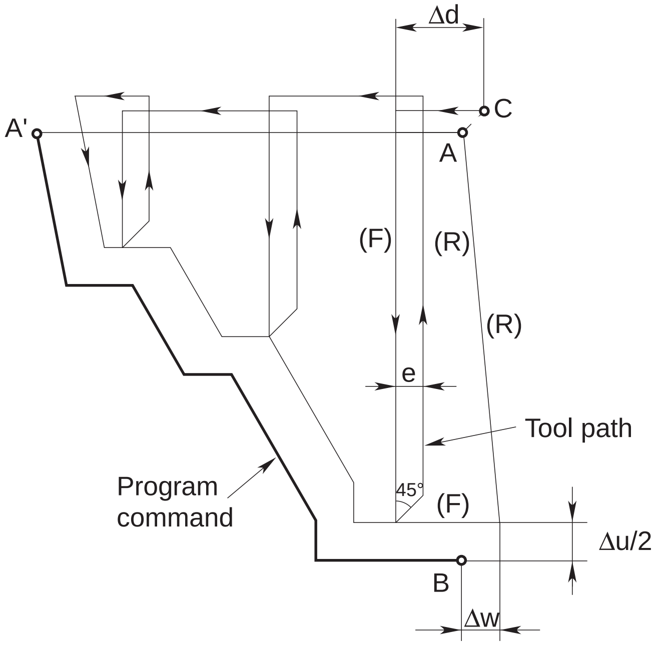
The tool path between **A** and **A'** is specified in the block with sequence number **“ns”** including **G00** or **G01**, and in this block a move command in the **Z** axis can not be specified.

The tool path between **A'** to **B** must be steadily increasing or decreasing pattern in both **X** and **Z** axes.

When the tool path between **A** and **A'** is programmed by **G00/G01**, cutting along **AA'** is performed in **G00/G01** mode respectively.

14.2.2 Stock removal in facing (G72)

This cycle is the same as **G71** except that cutting is made by operations parallel to **X** axis.

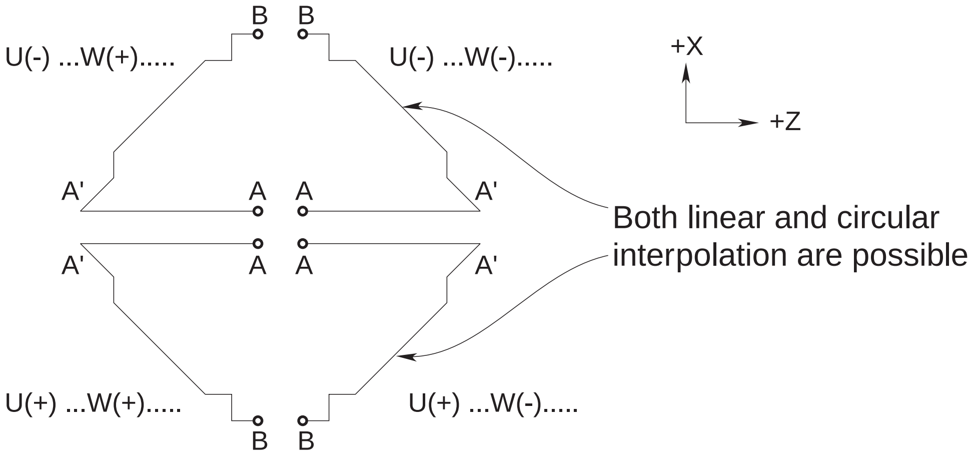


Format: **G72 W(Δd) R(e);**

G72 P(ns) Q(nf) U(Δu) W(Δw) F(f) S(s) T(t);

The meaning of **Δd, e, ns, nf, Δu, Δw, f, s** and **t** are the same as those in **G71**.

The following four cutting patterns can be considered. All of these cutting cycles are made parallel to **X** axis and the sign of **Δu** and **Δw** are as follows:



The tool path between **A** and **A'** is specified in the block with sequence number **“ns”** including **G00** or **G01**, and in this block a move command in the **X** axis can not be specified.

The tool path between **A'** to **B** must be steadily increasing or decreasing pattern in both **X** and **Z** axes.

Whether the cutting along **AA'** is **G00** or **G01** mode is determined by the command between **A** and **A'**, as described in item 14.2.1.

F, S and **T** functions in the blocks, which sequence number from **“ns”** to **“nf”**, are ignored. Also in this area the **G96** and **G97** codes are not effected end calling of the subprograms is not valid.

Notes:

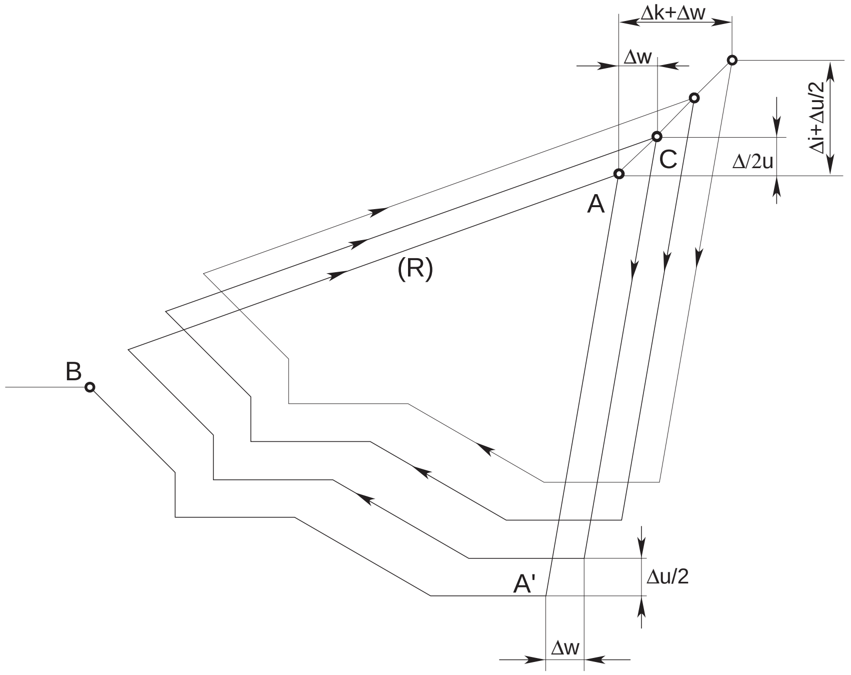
1. While both **Δd** and **Δu** are specified by address **U**, the meanings of them are determined by the presene of addresses **P** and **Q**.

2. The cycle machining is performed by **G71** command with **P** and **Q** specification. **F, S** and **T** functions which are specified in the move command between points **A** and **B** are ineffective. When an option of constant surface speed control is selected, **G96** or **G97** command specified in the move command between points **A** and **B** are ineffective, and that specified in **G71** block or the previous block is effective.

3. The subprogram can not be called from the block between sequence number **"ns"** and **"nf"**.

14.2.3 Pattern repeating (G73)

This function permits a fixed pattern repeatedly with a pattern being displaced bit by bit. This cycle is suitable for work whose shape has already been made by a rough machining or cutting method.



The format of this cycle should be as follows:

G73 U(Δi) W(Δk) R(d);

G73 P(ns) Q(nf) U(Δu) W(Δw) F(f) S(s) T(t);

where:

Δi: distance and direction of relief in **X** axis. This designation is modal and is not changed until onother value is designated. Also this value can be specified by the parameter No.719, and the parameter is changed by the program command.

Δk: distance and direction of relief in **Z** axis. This designation is modal and is not changed until other value is designated. Also this value can be specified by the parameter No.720, and the parameter is changed by the program command.

d: the number of repeats. This value is modal and is not changed until onother value is designated. Also this value can be specified by the parameter No.721 and the parameter is changed by the program command.

ns: sequence number of the first block from the program of finishing shape

nf: sequence number of the last block from the program of finishing shape

Δu: distance and direction of finishing allowence in **X** axis

Δw: distance and direction of finishing allowence in **Z** axis

14.2.4 Finishing cycle (G70)

After rough cutting by **G71, G72** or **G73**, the cycle **G70** permits finishing cutting. The format of the cycle is as follows:

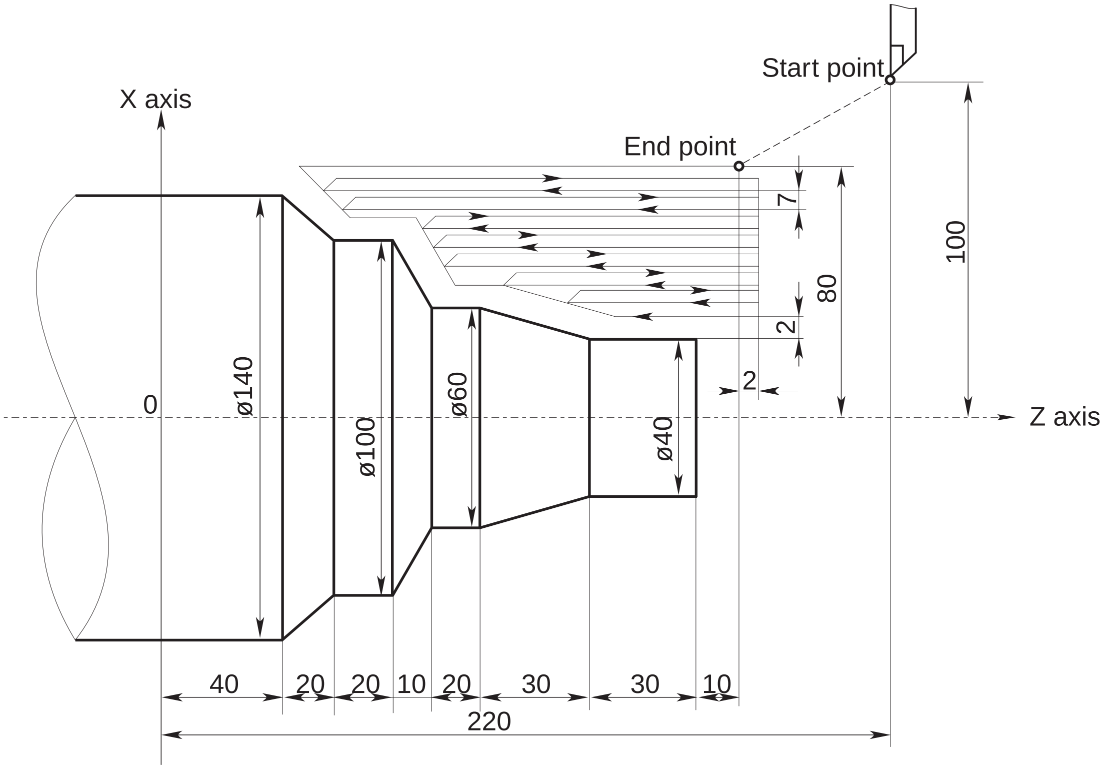
G70 <u>P(ns) Q(nf)</u>;

where:

ns: sequence number of the first block from the program of finishing shape

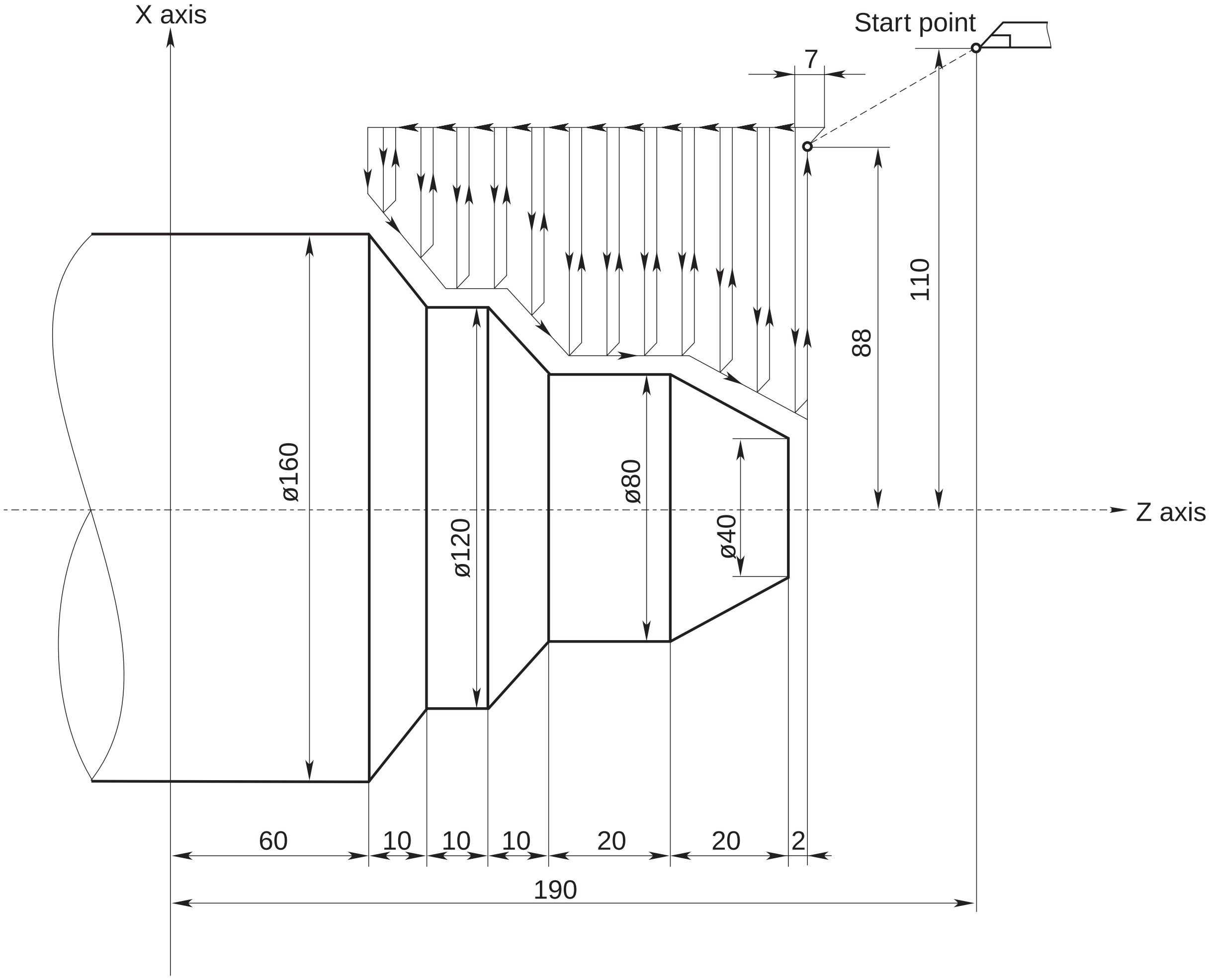
nf: sequence number of the last block from the program of finishing shape **M, S** and **T** functions can not be used in blocks, reffered in **G70** through **G73**.

Example of programming by multiple repetitive cycle **(G70, G71)**



```
N010 G50 X200.0 Z220.0;
N011 G00 X160.0 Z180.0;
N012 G71 U7.0 R1.0;
N013 G71 P014 Q020 U4.0 W2.0 F0.3 S55;
N014 G00 X40.0 F0.15 S58;
N015 G01 W-40.0;
N016 X60.0 W-30.0;
N017 W-20.0;
N018 X100.0 W-10.0;
N019 W-20.0;
N020 X140.0 W-20.0;
N021 G70 P014 Q020;
```

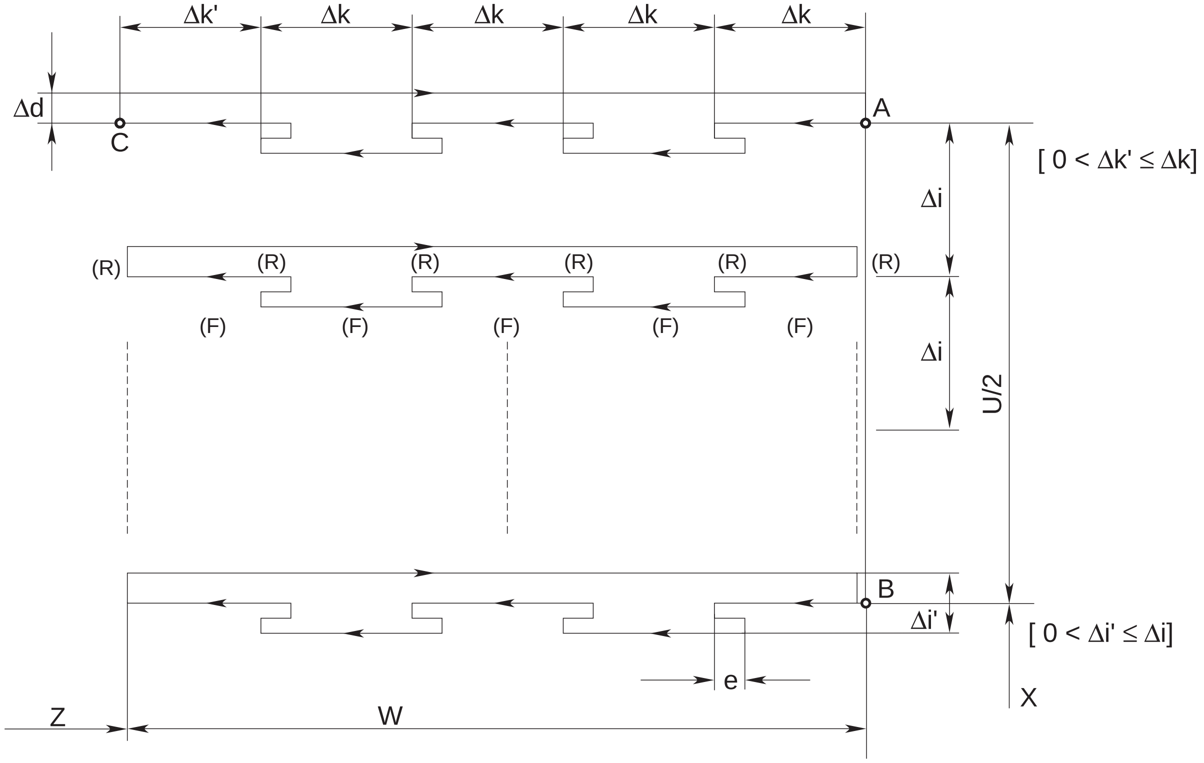
Example of programming by multiple repetitive cycle **(G70, G72)**



```
N010 G50 X220.0 Z 190.0;
N011 G00 X176.0 Z132.0;
N012 G72 W7.0 R1.0;
N013 G72 P014 Q019 U4.0 W2.0 Fo.3 S55;
N014 G00 Z58.0 S58;
N015 G01 X120.0 W12.0 F0.15;
N016 W10.0;
N017 X80.0 W-10.0;
N018 W20.0;
N019 X36.0 W-22.0;
N020 G70 P014 Q019;
```

14.2.5 End face peck drilling cycle (G74)

This cycle permits removal of the chip by the manner, shown in the figure below. If **X(U)** and **P** are omitted, operation only in **Z** axis results, to be used for drilling.



Format:

G74 R(e);

G74 X(U)___Z(W)___P(Δi) Q(Δk) R(Δd) F(f);

where:

e: return amount

This designation is modal and is valid until the other value is designated. Also this value can be specified by the parameter No.722 and the parameter is changed by the program command.

X: X component of point **B**

U: incremental amount from **A** to **B**

Z: **Z** component of point **C**

W: incremental amount from **A** to **C**

Δi: movement amount in **X** direction (without sign)

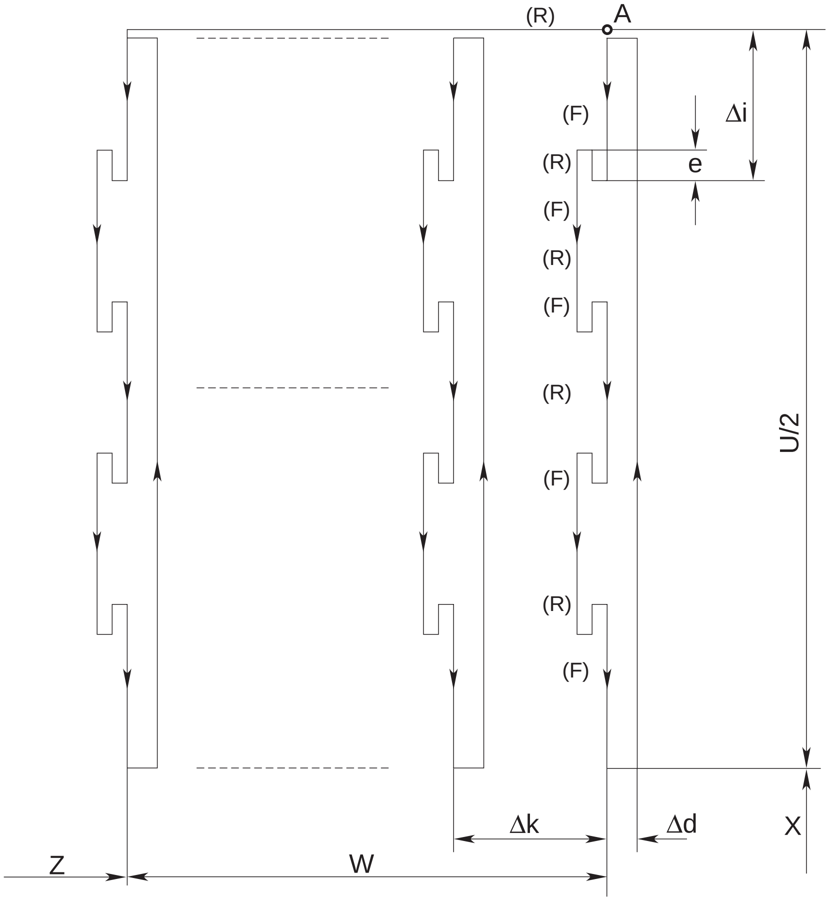
Δk: depth of cut in **Z** direction (without sign)

Δd: relief amount of the tool at the cutting bottom

f: feed rate

14.2.6 Outer/internal diameter drilling cycle (G75)

This cycle is equivalent to **G74** except that **X** is replaced by **Z** axis



Format:

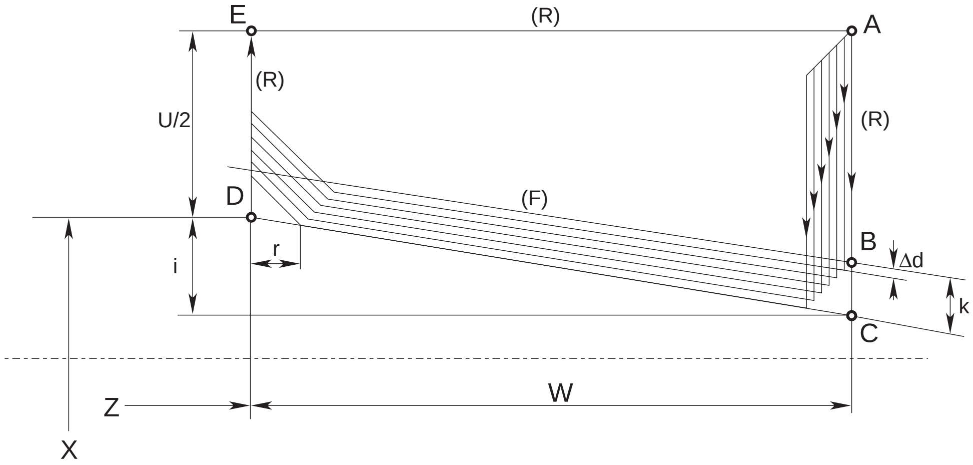
G75 R(e);

G75 X(U)___Z(W)___P(ΔDi) Q(Δk) R(Δd) F(f);

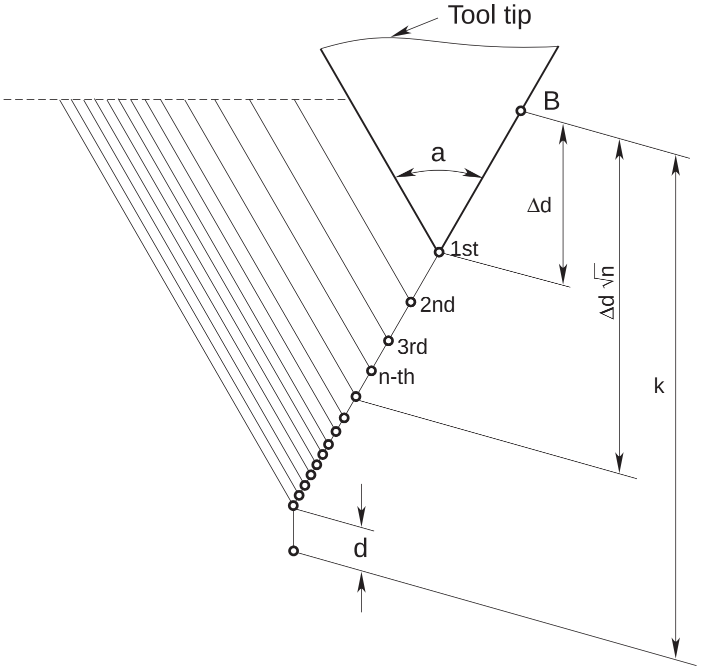
where: the parameters are the same as in the **G74**

14.2.7 Multiple thread cutting cycle (G76)

The thread cutting cycle can be programmed by the **G76** command as shown in the figure:



Detail of cutting:



Format: **G76 P (m) (r) (a) Q(Δd min) R(d);**

G76 X(U)___ Z(W)___ R(i) P(k) Q(Δd) F(l);

where:

m: repeat count in finishing (1 to 99)

This value is modal and is valid until onother value is designated. Also this value can be specified by the parameter No.723, and the parameter is changed by the program command.

r: chamfering amount

When the thread lead is expressed by l, the value of l can be set from **0.0l** to **9.9l** in 0.1 increment. This value can be specified by parameter No.109.

a: angle of tool tip

One of six the kinds of angle: 80°, 60°, 55°, 30°, 29°and 0° can be selected. This value is modal and can be specified by parameter No.724.

m, r and **a** are specified by address P at the same time.

Δdmin: minimum cutting depth

When the difference of the cutting depth in the previous and current operation becomes smaller than this value, the cutting depth is clamped at this value. This designation is modal and can be specified by parameter No.725.

d: finishing allowence

This designation is modal and can be specified by the parameter No.726.

i: difference of thread radius

If i ≠ 0, the taper thread can be made

k: height of the tread

This value is specified by the radius in X axis direction.

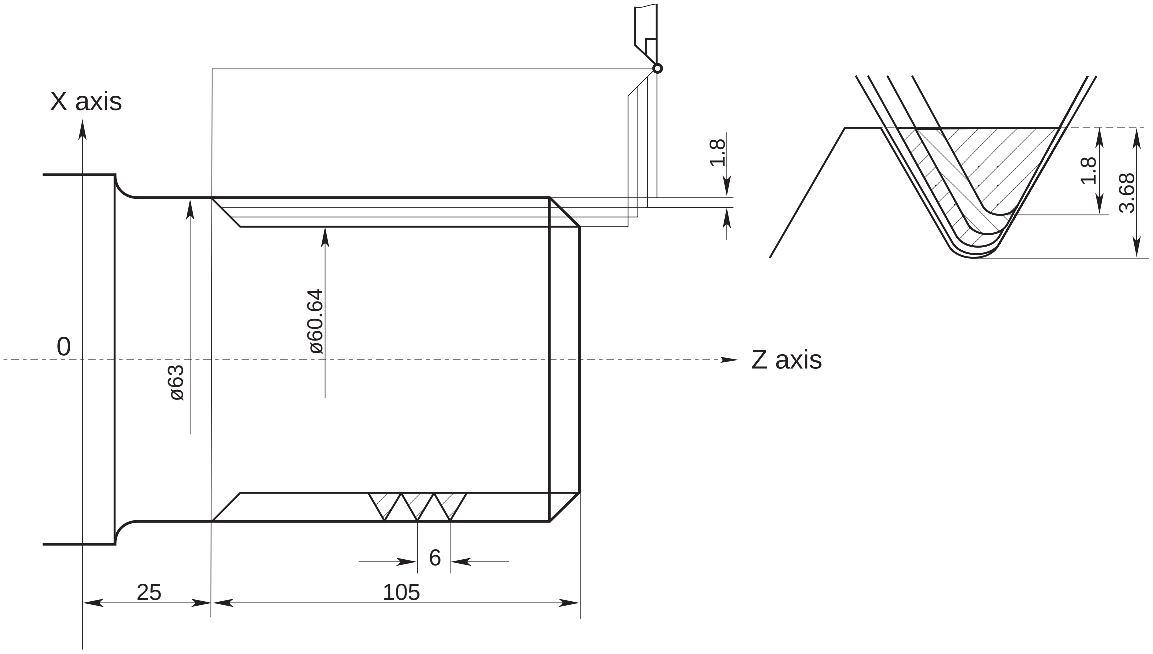
Δd: depth of cut in first cut

l: lead of cut

Example of programming by multiple repetitive cycle **(G76)**

G76 P011060 Q100 R200 ;

G76 X60640 Z25000 P3680 Q1800 F6.0 ;



14.2.8 Notes of multiple repetitive cycles (G70 to G76)

(1) *In the blocks, which are specified by address* ***P*** of **G71, G72** or **G73, G00** or **G01** of group should be commanded.

(2) *In* ***MDI*** *mode,* ***G71, G72*** *or* ***G73*** *can not be commanded.*

(3) *In the blocks between the sequence number specified by* ***P*** *and* ***Q***, the following commands can not be specified:

- one shot **G** code except for **G04**

- 01 group **G** code except for **G00, G01, G02** and **G03**

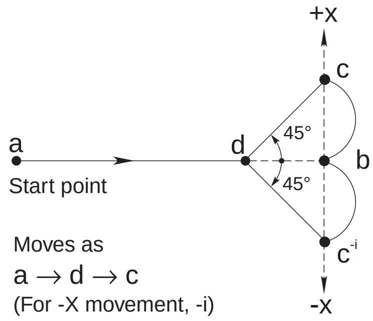
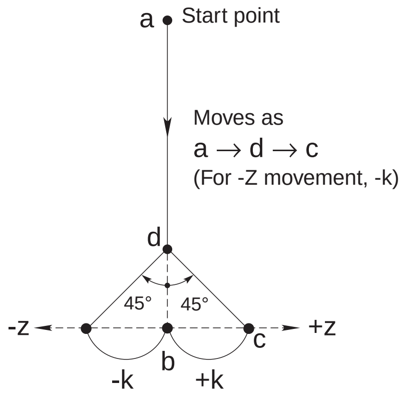
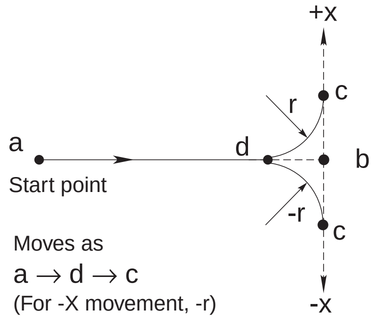
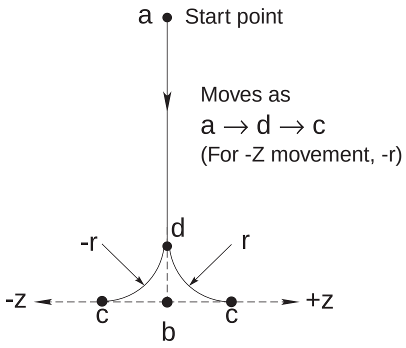
- 06 group **G** code

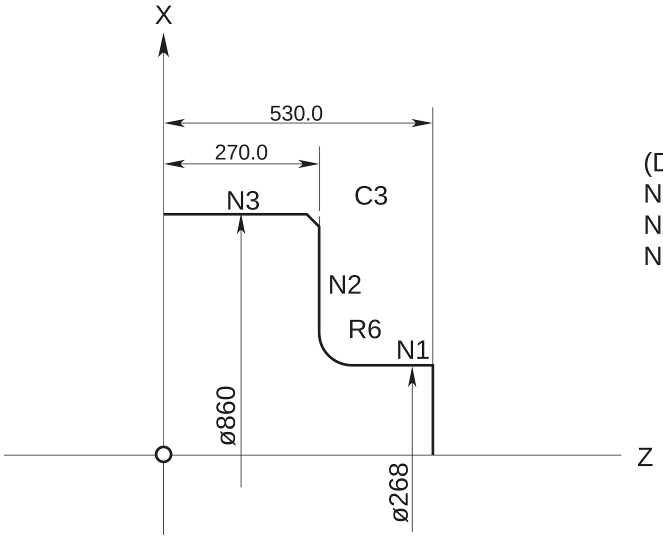
- M98 / M99 code

- chamfering and rounding **R**

14.3 Chamfering and Corner R

A chamfer or corner can be inserted between two blocks which intersect at a right angle as follows where **C** and **R** always specify a radius value.

Chamfering Z → X	G01Z(W) b/↑C ±i; Specifies movement to point b with an absolute or incremental command in the figure on the right.	 +x c a d 45° b Start point 45° Moves as a → d → c c-i (For -X movement, -i) -x
Chamfering X → Z	G01X(U) b/↑C ±k; Specifies movement to point b with an absolute or incremental command in the figure on the right.	 a Start point Moves as a → d → c (For -Z movement, -k) d 45° 45° -z +z c b -k +k
Corner R Z → X	G01ZbR↑ ±r; Specifies movement to point b with an absolute or incremental command in the figure on the right.	 +x c r a d b Start point -r c Moves as a → d → c (For -X movement, -r) -x
Corner R X → Z	G01XbR↑ ±r; Specifies movement to point b with an absolute or incremental command in the figure on the right.	 a Start point Moves as a → d → c (For -Z movement, -r) d -r r -z +z c c b



```
(Diameter programming)
N1   Z270.0   R6.0 ;
N2   X860.0   C-3.0 ;
N3   Z0 ;
```

The first movement for chamfering or corner **R** must be specified only along one axis. The second movement must be only along the axis perpendicular to the former movement.

Chamferings and corner **R** can not be used in a thread cutting block.

The following commands cause an alarm:

(1) *One **C** or **R** is commanded when **X** and **Z** are in a block*, contents addresses **G01**(PS054).

(2) *The move amount of **X** or **Z** is less than chamfering **C** value and corner **R** value (PS55).*

(3) *Next block to the block where chamfering and corner **R** were specified* has not **G01** command (PS051 or PS052).

When **C** and **R** are specified to the same block in **G01** code, the command that is specified later is valid.

14.4 Mirror Image for Double Turrets (G68, G69)

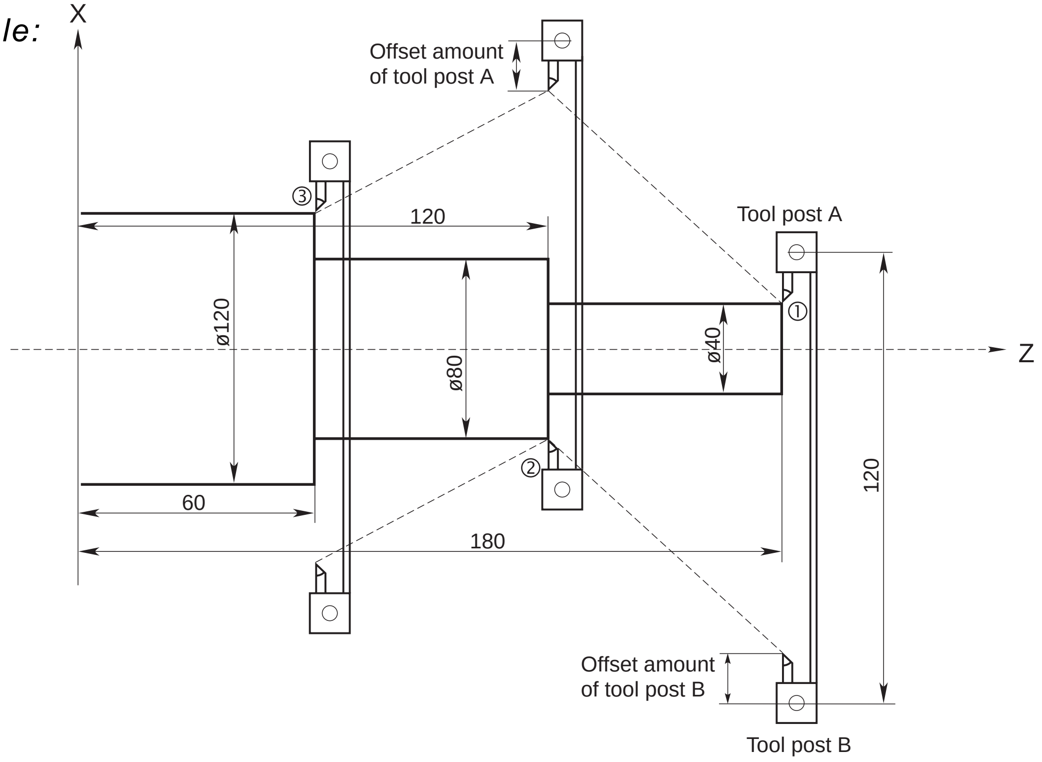
Mirror image can be applied to **X** axis by the following **G** codes:

G68: double turret mirror image on

G69: mirror image cancel

When **G68** is designated, the coordinate system is shifted to the mating turret side, and the **X** axis sign is reversed to perform symmetrical cutting. Before using this function, set the distance between two turrets to a parameter No.730.

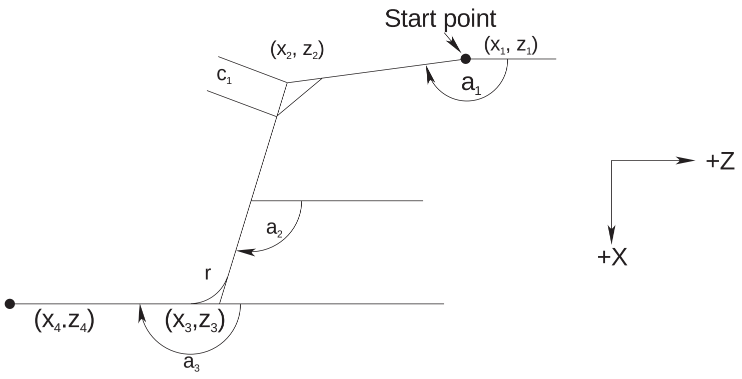
Example:



X40.0 Z180.0 T0101;	Position turret **A** at ①
G68;	Shift the coordinate system by the distance from **A** to **B** and turn mirror image on
X80.0 Z120.0 T0202;	Position turret **B** at ②
G69;	Shift the coordinate system by the distance from **B** to **A** and turn mirror image on
X120.0 Z60.0 T0101;	Position turret **A** at ③

14.5 Direct Drawing Dimension Programming

Angles of straight lines, chamfering value, corner rounding value and other dimensional values on machining drawings can be programmed by directly inputting these values.

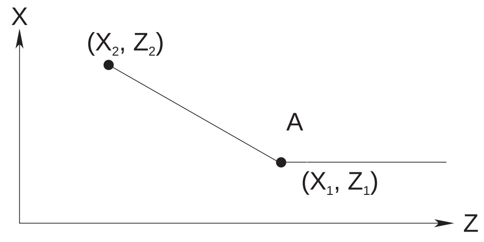
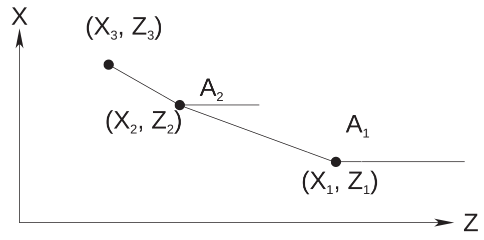
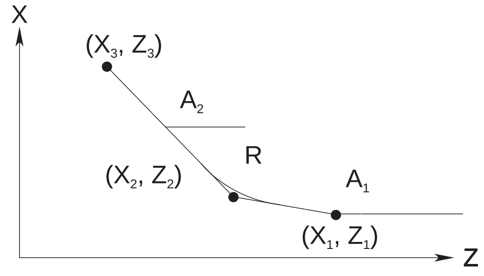


$X(x_2)\ Z(z_2)\ C(c_1);$
$X(x_3)\ Z(z_3)\ R(r_2)$
$X(x_4)\ Z(z_4);$

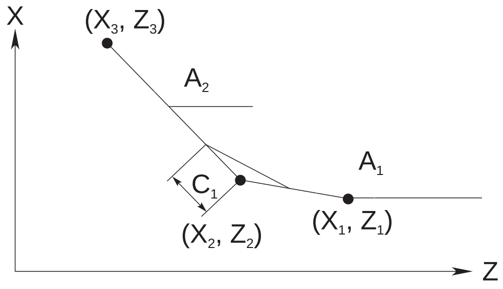
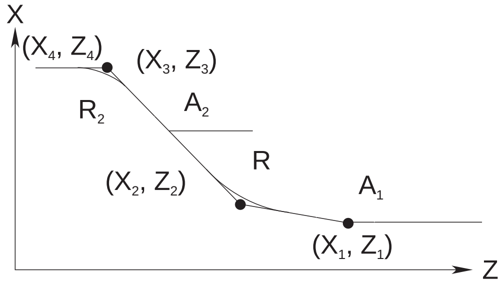
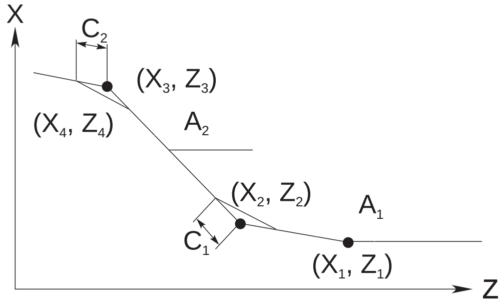
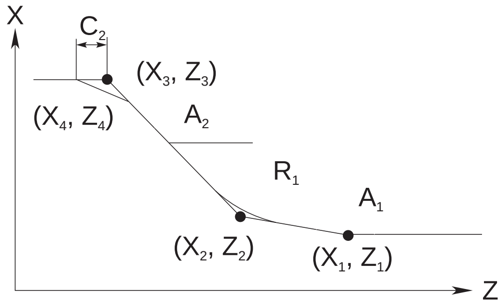
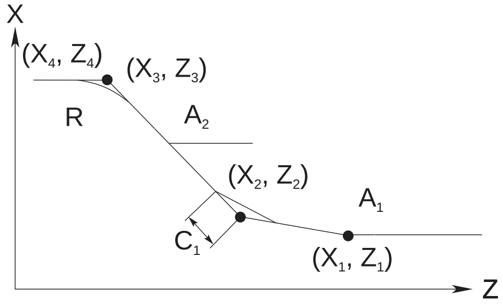
or

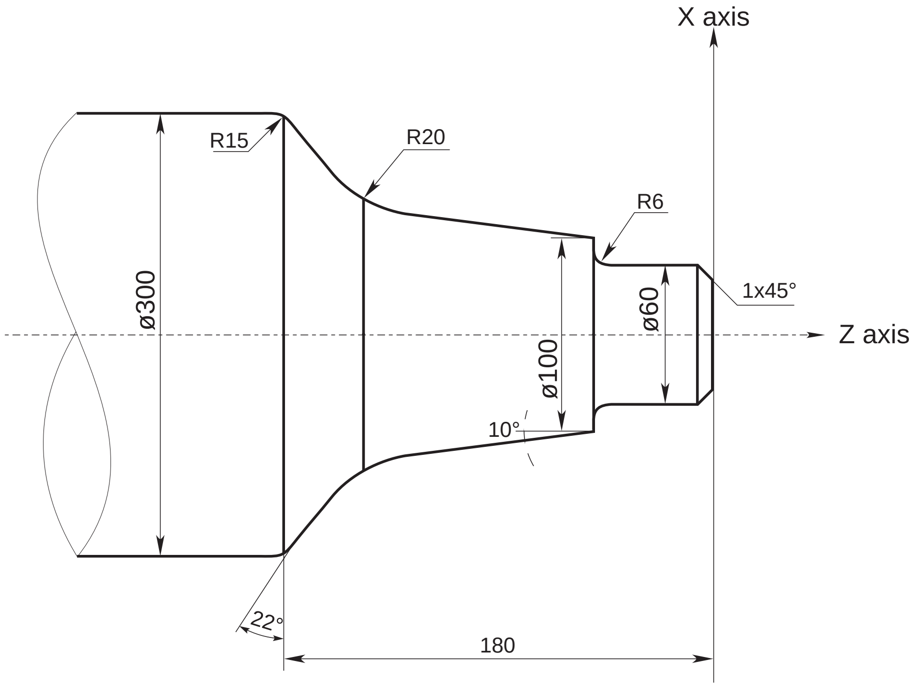
$A(a_1)\ C(c_1);$
$X(x_3)\ Z(z_3)\ A(a_2)\ R(r_2);$
$X(x_4)\ Z(z_4);$

Commands table

	Commands	Movement of tool
1	X_2___ (Z_2___) A___;	
2	A_1___; X_3___ Z_3___ A_2___;	 X (X_3, Z_3) A_2 (X_2, Z_2) A_1 (X_1, Z_1) Z
3	X_2___ Z_2___ R_1___; X_3___ Z_3___; or A_1___ R_1___; X_3___ Z_3___ A_2___;	 X (X_3, Z_3) A_2 R (X_2, Z_2) A_1 (X_1, Z_1) Z

Continued

	Commands	Movement of tool
4	X_2____ Z_2____ C_1____; X_3____ Z_3____; or A_1____ C_1____; X_3____ Z_3____ A_2____;	 X (X_3, Z_3) A_2 A_1 C_1 (X_2, Z_2) (X_1, Z_1) Z
5	X_2____ Z_2____ R_1____; X_3____ Z_3____ R_2____; X_4____ Z_4____; or A_1____ R_1____; X_3____ Z_3____ R_2____; X_4____ Z_4____;	 X (X_4, Z_4) (X_3, Z_3) R_2 A_2 R (X_2, Z_2) A_1 (X_1, Z_1) Z
6	X_2____ Z_2____ C_1____; X_3____ Z_3____ C_2____; X_4____ Z_4____; or A_1____ C_1____; X_3____ Z_3____ A_2____ C_2____; X_4____ Z_4____;	 X C_2 (X_3, Z_3) (X_4, Z_4) A_2 (X_2, Z_2) A_1 C_1 (X_1, Z_1) Z
7	X_2____ Z_2____ R_1____; X_3____ Z_3____ C_2____; X_4____ Z_4____; or A_1____ R_1____; X_3____ Z_3____ A_2____ C_2____; X_4____ Z_4____;	 X C_2 (X_3, Z_3) (X_4, Z_4) A_2 R_1 A_1 (X_2, Z_2) (X_1, Z_1) Z
8	X_2____ Z_2____ C_1____; X_3____ Z_3____ R_2____; X_4____ Z_4____; or A_1____ C_1____; X_3____ Z_3____ A_2____ R_2____; X_4____ Z_4____;	 X (X_4, Z_4) (X_3, Z_3) R A_2 (X_2, Z_2) A_1 C_1 (X_1, Z_1) Z



In the blocks, containing this kind of programming, is not permitted usage of:

(1) *thread cutting commands.*

(2) *canned cycles.*

(3) *non-modal **G** codes except **G04**.*

(4) ***G02** and **G03** codes.*

The angle values 0°, 90°, 180° and 270° occur an alarm.
Programming with angles is effective only in AUTO operation mode.

Example:

```
N001 G50 X0.0 Z0.0;
N02  G01 X60.0 A90.0 C1.0 F80;
N003 Z-30.0 A180.0 R6.0;
N004 X100.0 A90.0;
N005 A170.0 R20.0;
N006 X300.0 Z-180.0 A112.0 R15.0;
N007 Z-230.0 Z180.0;
          .
          .
          .
```

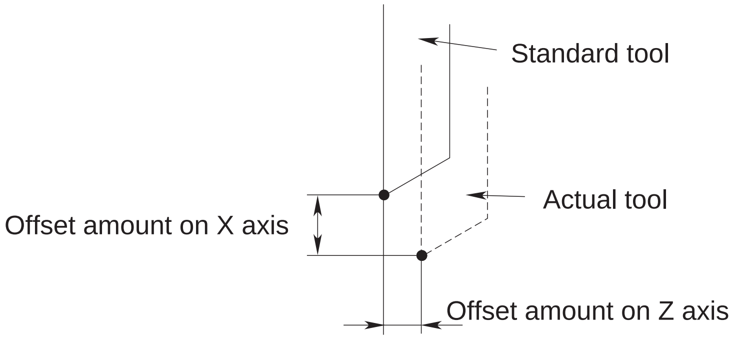
15.COMPENSATION FUNCTIONS

15.1 Tool Offset

The tool offset is specified by **T** code.

15.1.1 Basic Tool Offset

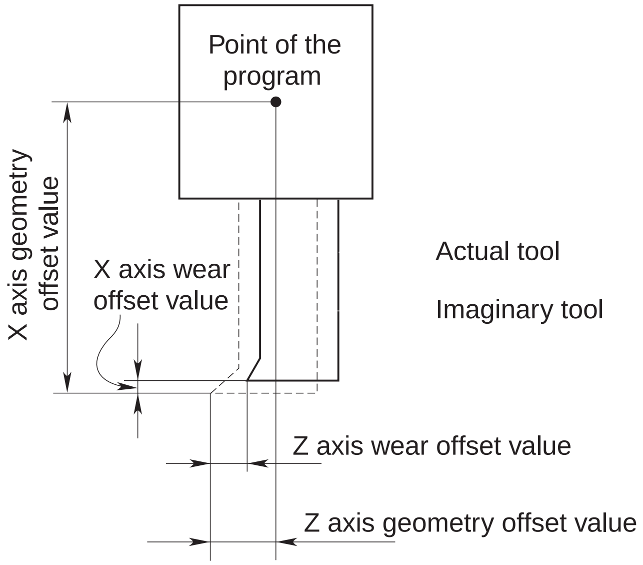
Tool offset is used to compensate for the difference when the tool actually used differs from the imagined tool used in programming (standard tool, usually)

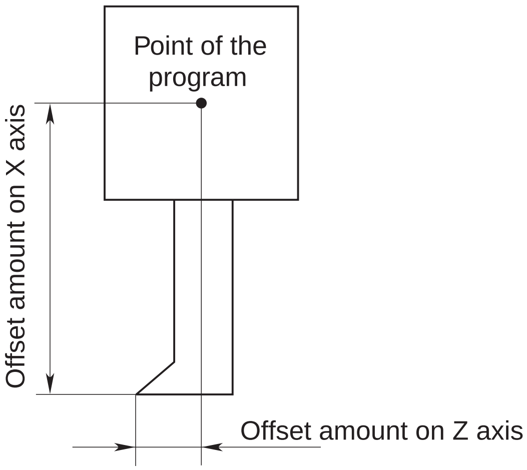


15.1.2 Tool geometry offset and tool wear offset

The tool geometry offset is used to compensate the tool shape. The tool wear offset is used to compensate the tool nose wear.

The tool geometry offset shifts the coordinate system without performing a movement. This offset is the same as the coordinate system shift with thw sign minus. The tool wear offset shifts the coordinate system and moves the tool.



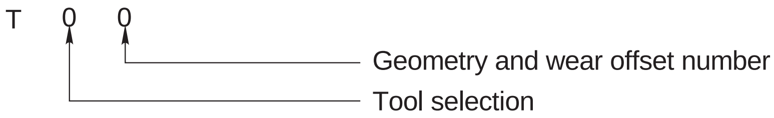


15.1.3 T code for tool offset

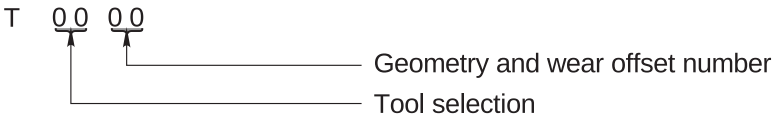
The specified **T** codes have the following meanings:

(1) *The geometry offset and wear offset numbers are specified by low order one or two digits* of the **T** code (parameter No.013 GOFU2=0)

For T(1 + 1) (Parameter No. 0014, T2D = 1)

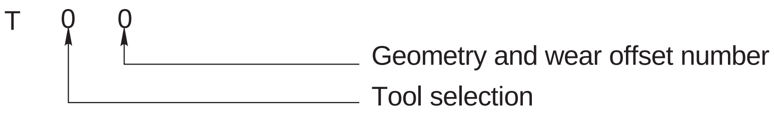


For T(2 + 2) (Parameter No. 0014, T2D = 0)

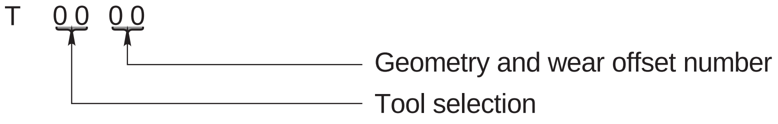


(2) *The wear offset is specified by junior part, and the geometry offset is specified by senior part* of the **T** code (parameter No.013 GOFU2=1).

For T(1 + 1) (Parameter No. 0014, T2D = 1)



For T(2 + 2) (Parameter No. 0014, T2D = 0)



15.1.4 Tool selection

The tool selection is made by specifying the **T** code. Refer to the machine tool builder’s manual for the relationship between the tool selection number and the tool.

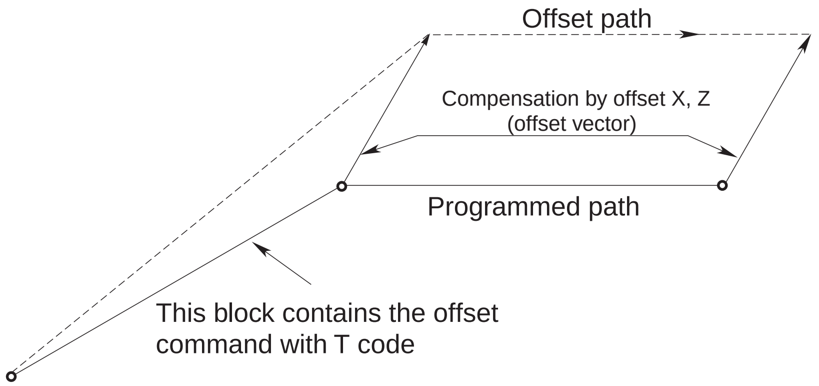
15.1.5 Offset number

The offset number corresponding to the definite distance which is stored into the system’s memory and this distance can be changed in **MDI** mode or by transfering in the serial channel. A tool offset **0** or **00** indicates that the offset amount is **0** and the offset is cancelled.

15.1.6 Offset

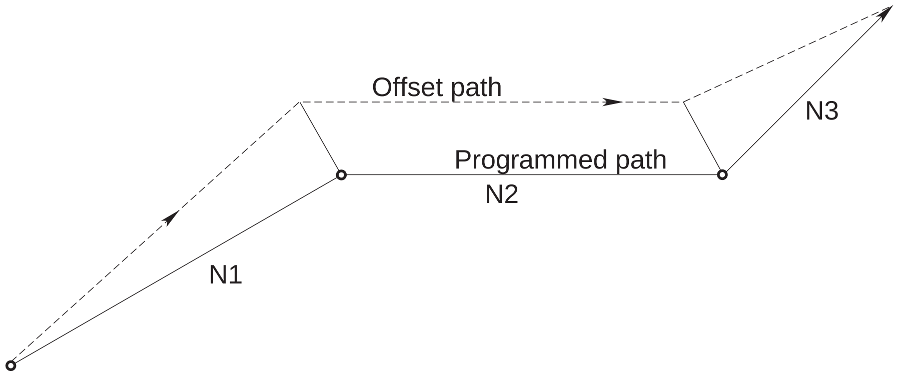
15.1.6.1 Wear offset

The tool path is offset by the **X** and **Z** offset values for the programmed path. The offset distance corresponding to the number specified by **T** code is added or subtracted from the end position of each programmed block.



Offset is cancelled when **T** code offset number **0** or **00** is selected. At the end of the cancelled block, the offset vector becomes zero.

```
N1 X50.0 Z100.0 T0202;
N2 Z200.0;
N3 X100.0 Z250.0 T0200;
```



(An offset value is assumed to have been entered in the 02 offset memory OFX, and OFZ, respectively.)

When the **RESET** key on the **MDI** unit is pushed or the rest signal is input to the **NC** from the machine tool, the offset is cancelled. Parameter No.001 TOC can be set so that the offset will not be cancelled by pressing the **RESET** key or by reset input.

When a **T** code is specified in a block only, the tool is moved by the wear offset value without a move command. The movement is performed at rapid traverse in G00 mode.

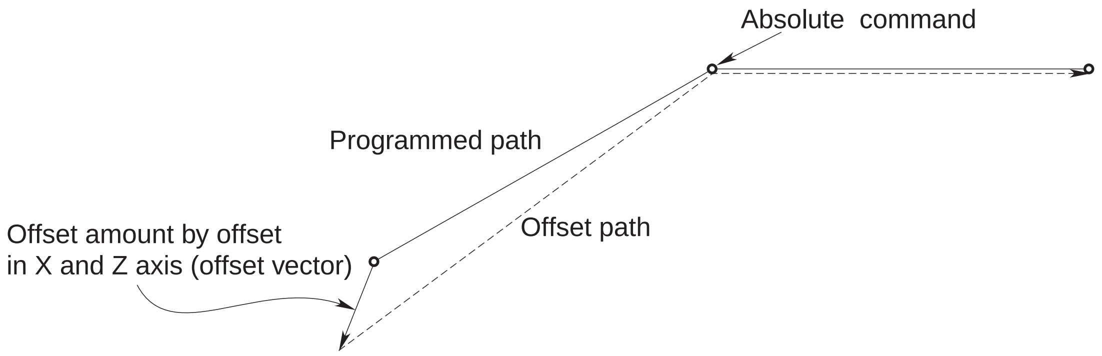
The tool will not move in the following block;

```
G50 X(x) Z(z) T____;
```

The coordinate system will be set in assigned coordinate **X** and **Z**. The tool position is obtained by subtracting the wear offset value corresponding to the offset number specified in the **T** code.

15.1.6.2 Geometry offset

With the geometry offset, the work coordinate system is shifted along the **X** and **Z** axes.



As well as wear offset, the geometry offset is determined by the parameter No.013 GMOFS whether to add or subtract the programmed end point of each block.

Offset cancel:

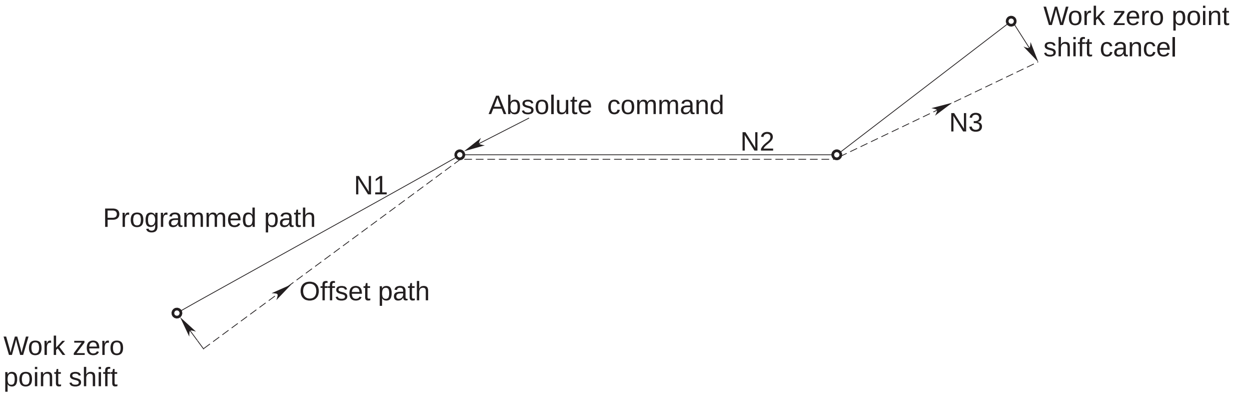
(1) *When designated wear and geometry offset number by last one or last two digits of **T** code* (parameter No.013 GOFU=0), the offset cancel is accomplished by specifying number **0**. The offset cancel is valid in case of parameter No.013 GOFC=1.

Example:

N1 X50.0 Z100.0 T0202;

N2 Z200.0;

N3 X100.0 Z250.0 T020;

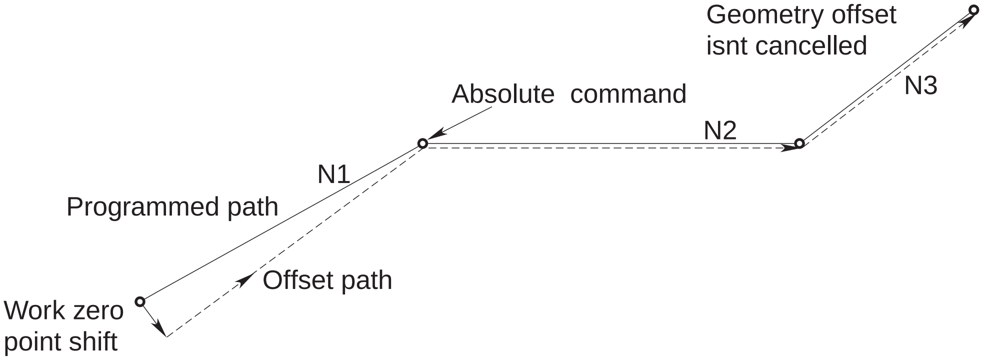


(Assume that there are offset amounts set at OFGX and OFGZ of the No. 02 geometry offset memory)

(2) *The geometry offset is designated by tool selection number* (parameter No.013GOFU2=1)

Example:

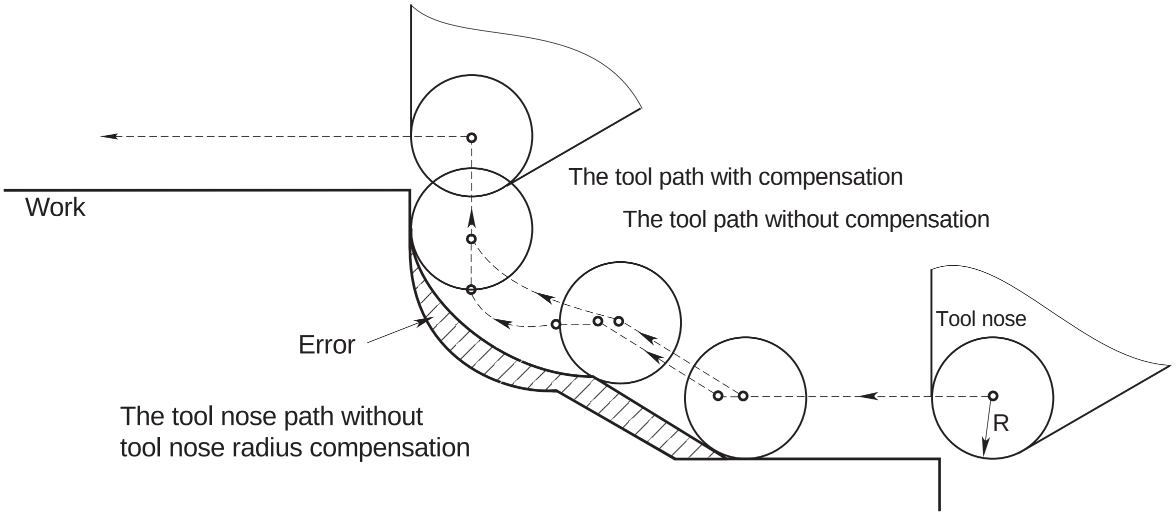
```
N1 X50.0 Z100.0     T0202;
N2 X200.0;
N3 X100.0 Z250.0 T0200;
```



(Assume that there are offset amounts set at OFGX and OFGZ of the No. 02 geometry offset memory)

15.2 Tool Nose Radius Compensation (G40 to G42)

To produce parts it is difficult to achieve big accuracy because of the tool nose roundness. In this case the tool nose radius compensation function automatically is used.

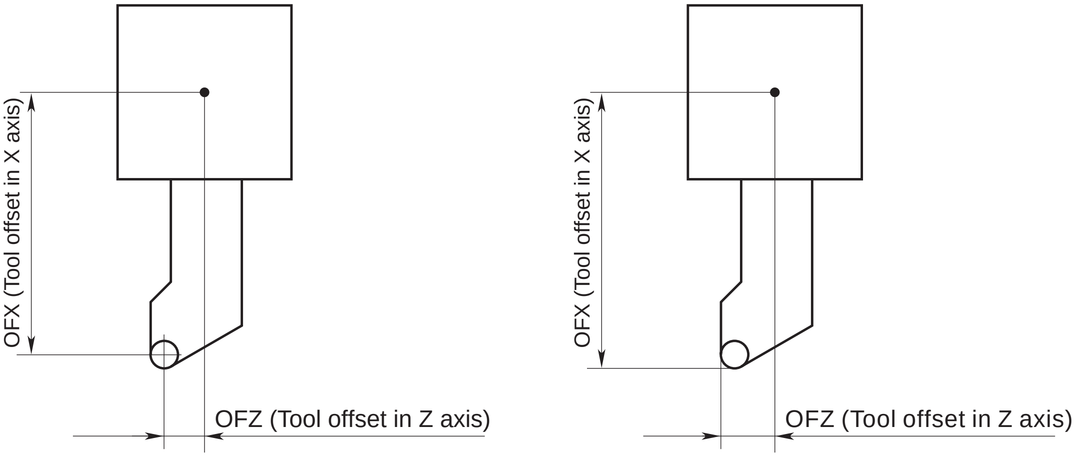

The tool path with compensation
The tool path without compensation
Work
Tool nose
R
Error
The tool nose path without
tool nose radius compensation

15.2.1 Imaginary tool nose

The tool nose at position **A** does not actually exists. the imaginary tool nose is required because it is more convenient to use than the real center of roundness of the tool nose. When imaginary tool nose is used, the tool nose radius need not be considered in programming.

In a machine with a reference point, a standard point like the turret center can be placed over the start point. The distance from this standard point to the nose radius center or the imaginary tool nose is set as the tool offset value. This offset is the same as placing the tool nose radius center over the start point.

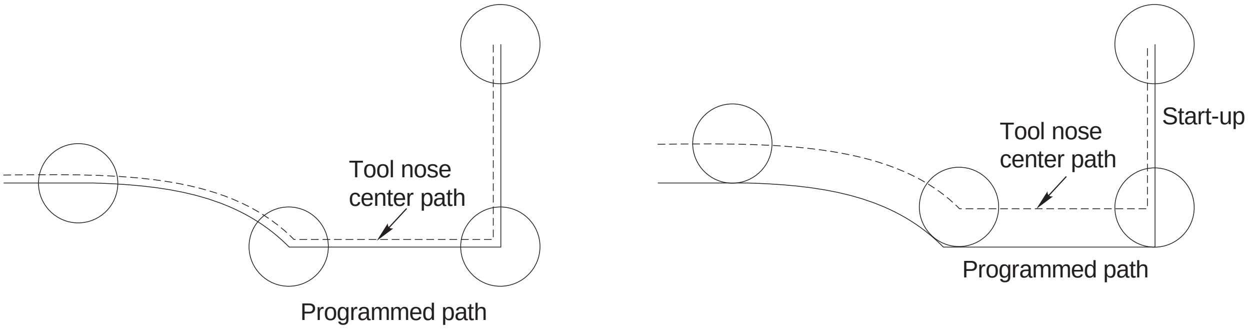
When the turret center is placed over the start point


OFX (Tool offset in X axis)
OFZ (Tool offset in Z axis)
OFX (Tool offset in X axis)
OFZ (Tool offset in Z axis)

(1) *programming using the tool nose center*

Unless tool nose radius compensation is performed, the tool nose center path is the same as the programmed path

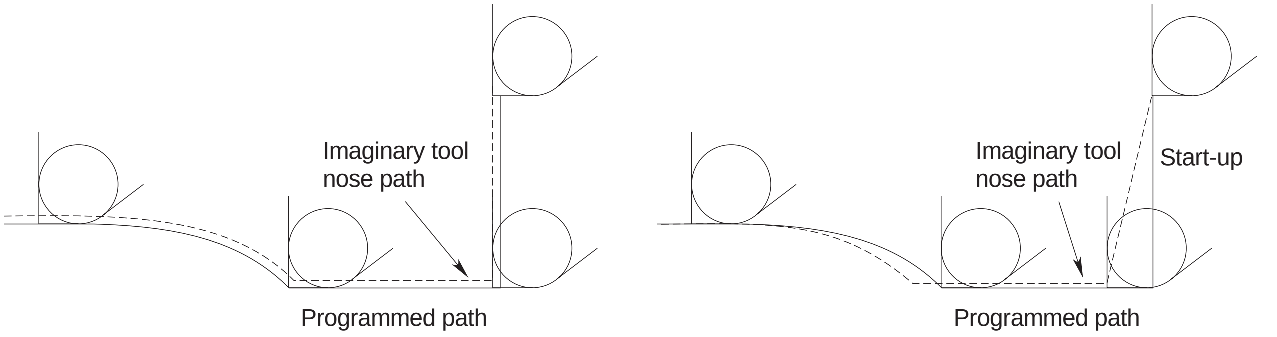
If tool nose radius compensation is used, accurate cutting will be performed



(2) *programming using imaginary tool nose*

Unless tool nose radius compensation is used, the imaginary tool nose path is the same as the programmed path

If tool nose radius compensation is used, accurate cutting will be performed



15.2.2 Direction of imaginary tool nose

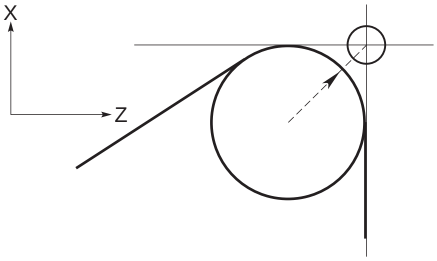
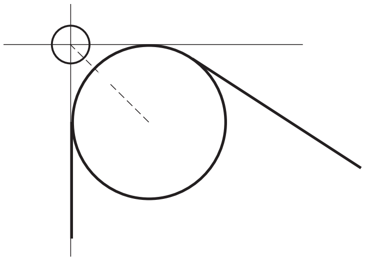
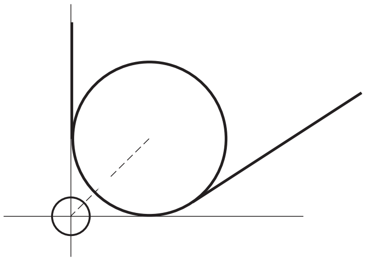
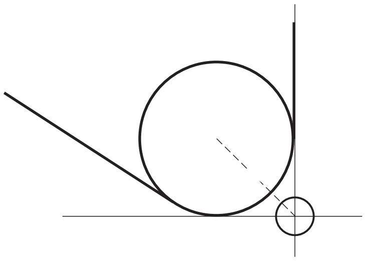
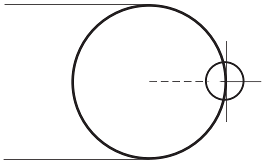
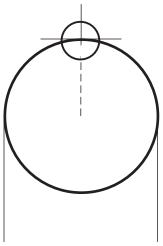
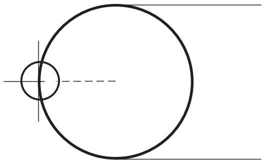
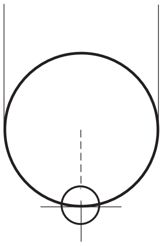
The direction of the imaginary tool nose viewed from the tool nose center is determined by the direction of the tool during cutting. This direction must be set in advance as well as the offset values.

The direction can be selected and specified by one of the following numbers:

In this case, the tool nose radius compensation amount is the sum of the geometry and wear offset amounts:

OFR=OFGR+OFWR

When the geometry offset is specified by the tool number and this number is different of those of the wear offset, the tool nose radius compensation is given by the geometry and wear offsets.

 Imaginary tool nose number 1	 Imaginary tool nose number 2
 Imaginary tool nose number 3	 Imaginary tool nose number 4
 Imaginary tool nose number 5	 Imaginary tool nose number 6
 Imaginary tool nose number 7	 Imaginary tool nose number 8

15.2.3 Offset number

The value is set by the keyboard.

Tool nose radius compensation

This value is set from the MDI according to the offset number

Offset number	OFX (Offset amount on X axis)	OFZ (Offset amount on Z axis)	OFR (Tool nose radius compensation amount)	OFT (Direction of imaginary tool nose)
01	0.040	0.020	0.20	1
02	0.060	0.030	0.25	2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
31	0.050	0.015	0.12	6
32	0.030	0.025	0.24	3
Max. 32 pairs				

Geometry Offset

Geometry offset number	OFGX (X-axis geometry offset amount)	OFGZ (Z-axis geometry offset amount)	OFGR (Tool nose radius geometry offset amount)	OFT (Iimaginary tool nose direction)
G 01	10.040	50.020	0	1
G 02	20.060	30.030	0	2
G 03	0	0	0.20	6
G 04	.	.	.	.
G 05	.	.	.	.
.	.	.	.	.

Wear Offset

Wear offset number	OFWX (X-axis wear offset amount)	OFWZ (Z-axis wear offset amount)	OFWR (Tool nose radius wear offset amount)	OFT (Iimaginary tool nose direction)
W 01	0.040	0.020	0	1
W 02	0.060	0.030	0	2
W 03	0	0	0.20	6
W 04	.	.	.	.
W 05	.	.	.	.
.	.	.	.	.

In this case, the tool nose radius compensation amount is the sum of the geometry and wear offset amounts:

OFR = OFGR + OFWR

When the geometry offset is specified by the tool number and this number is defferent of those of the wear offset, the tool nose radius compensation is given by the geometry and wear offsets.

Example:

T0102;
OFR=OFGR01+OFWR02;
OFT=OFT01;

The range of the offset values is:

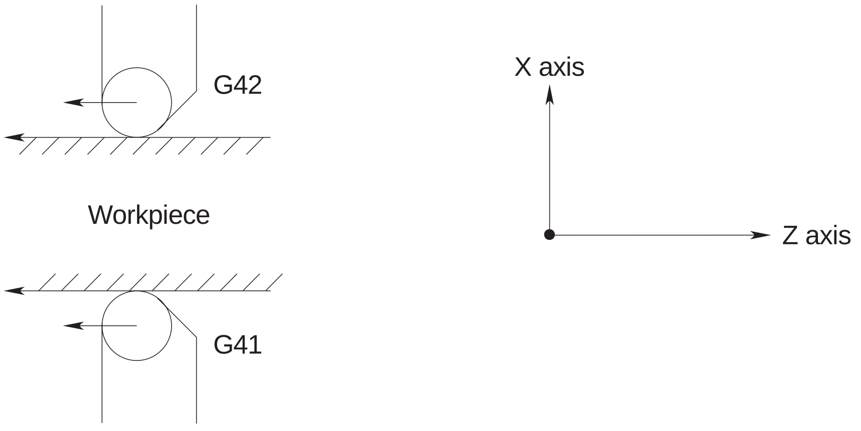
	mm input	inch system
Offset amount	0 - ±999.999mm	0 - ±99.9999inches

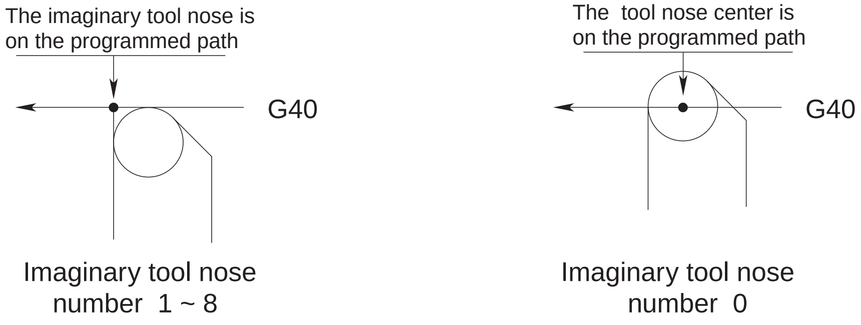
15.2.4 Work position and move command

In tool nose radius compensation, the position of the workpiece in respect to the tool must be specified.

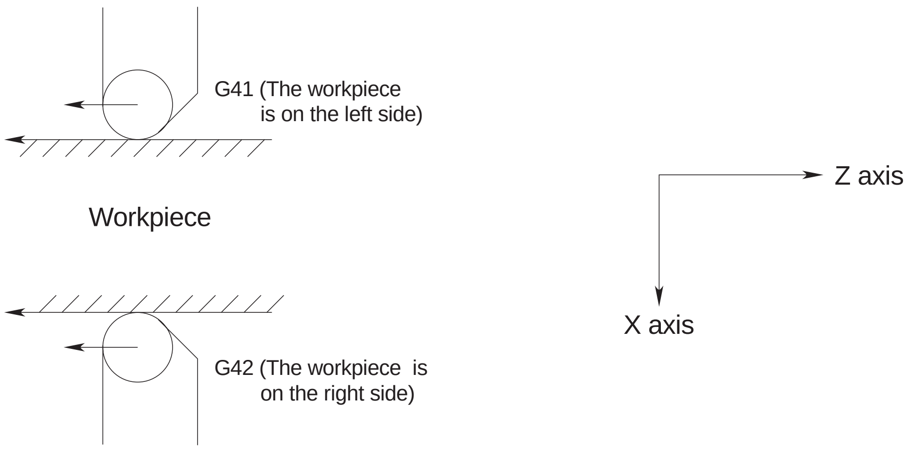
G code	Work position	Tool path
G 40	(Cancel)	Moving along the programmed path
G 41	Right side	Moving on the left side of the programmed path
G 42	Left side	Moving on the right side of the programmed path

The tool is offset to the side opposite the side of the workpiece.




The imaginary tool nose is on the programmed path
G40
Imaginary tool nose number 1 ~ 8
The tool nose center is on the programmed path
G40
Imaginary tool nose number 0

The position of the workpiece in respect to the tool can be changed by setting the coordinate system.


G41 (The workpiece is on the left side)
Workpiece
G42 (The workpiece is on the right side)
Z axis
X axis

If the tool nose radius compensation value is negative, the workpiece position is changed.

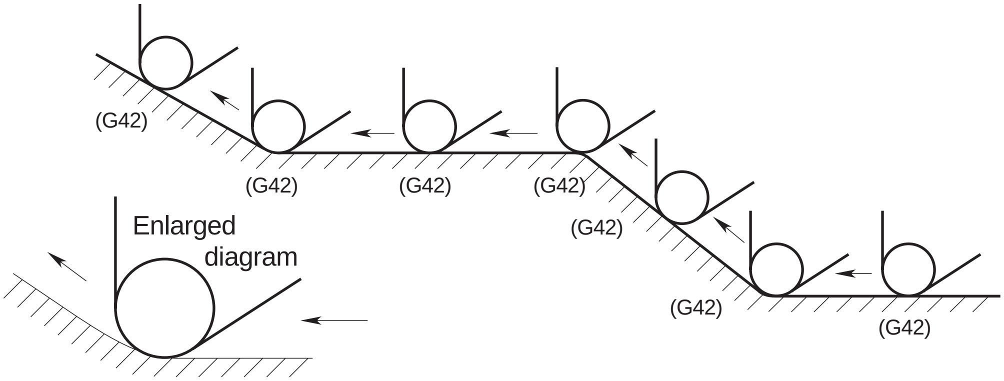
The codes **G40, G41** and **G42** are modal.

```
G41  X . . . . . . . . . .  Z . . . . . . . . . . . ;  }
     X . . . . . . . . . .  Z . . . . . . . . . . . ;  }  G41 mode
     X . . . . . . . . . .  Z . . . . . . . . . . . ;  }
G42  X . . . . . . . . . .  Z . . . . . . . . . . . ;  }  G42 mode
     X . . . . . . . . . .  Z . . . . . . . . . . . ;  }
G40  X . . . . . . . . . .  Z . . . . . . . . . . . ;  }  G40 mode
     X . . . . . . . . . .  Z . . . . . . . . . . . ;  }
```

You may not specify **G41** while in the **G41** mode. If you do, the compensation is not the same. For the **G42**, the same is valid.

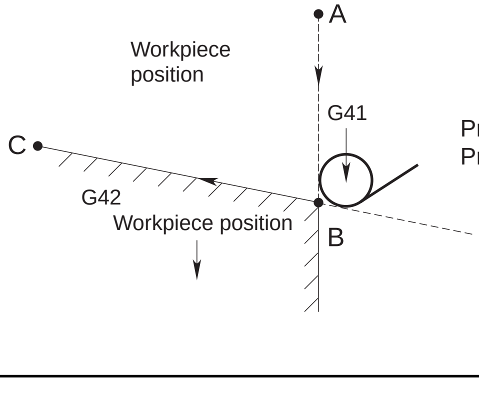
(1) *In the case of workpiece position does not change*

When the tool is moving, the tool nose maintains contact with the workpiece.



(2) *In case of workpiece position changes*

The workpiece position against the tool changes at the corner of the programmed path as shown in the figure below:



Programmed path from A to B: G41
Programmed path from B to C: G42

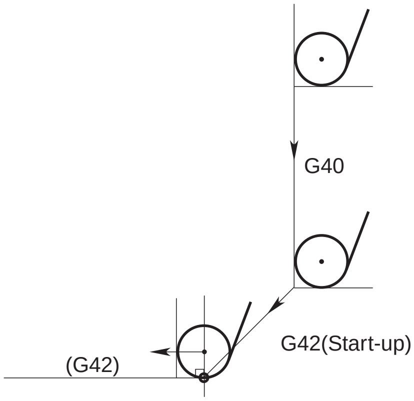
Although the workpiece does not exist on the right side of the programmed path in the above case, the existance of the workpiece is assumed in the movement from **A** to **B**. The workpiece position must not be changed in the block next to the start-up block. In the above example, if the block specifying motion from **A** to **B** was the start-up block, the tool path would not be the same as the one shown.

(3) *Start-up*

The block in which the mode changes to G41or G42 from G40 is called the start-up block.

```
G40____;
G41____;
   ____;
   ____;
```

Transient tool movements for offset are performed in the start-up block. In the block after the start-up block, the tool nose center is positioned vertically to the programmed path of the block at the start point.

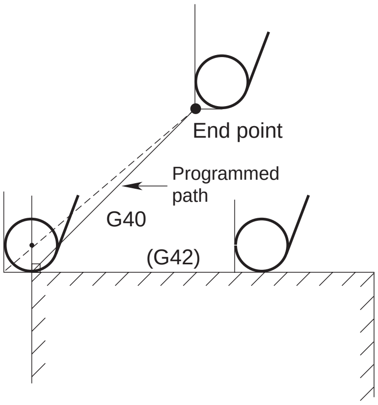


(4) *Offset cancel* (Start-up block)

The block in which the mode changes to **G40** from **G41** or **G42** is called the offset cancel block.

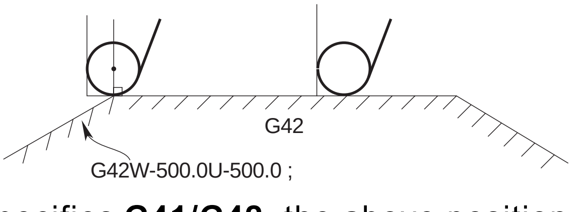
G41____;
____;
G40____;

The tool nose center moves to a position vertical to the programmed path in the block before the cancel block. The tool is positioned at the end point in the offset cancel block (**G40**), as shown below:



(5) *In case of* ***G41/G42*** *is specified again in* ***G41/G42*** *mode*

In this case the tool nose center is positioned vertically to the programmed path of the preceeding block at the end point of the preceding block.

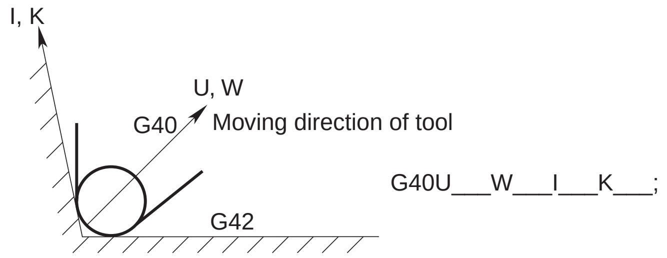


In the block that first specifies **G41/G42**, the above positioning of the tool nose center is not performed.

(6) *When moving direction of the tool in a block which includes a **G40** command is different from the direction of the workpiece*.

When you wish to retract the tool in the direction specified by **X(U)** and **Z(W)** cancelling the tool nose radius compensation, specify the following block:

G40 X(U)____Z(W)____I____K____;



The addresses **I** and **K** indicate the work position that must be specified with a **G40** command in a block. When they are specified with **G02** or **G03**, they are regarded as coordinate values of an arc center.

G40 X__Z__I__K__;	Tool nose radius compensation
G40 G02 X__Z__I__K__;	Circular interpolation

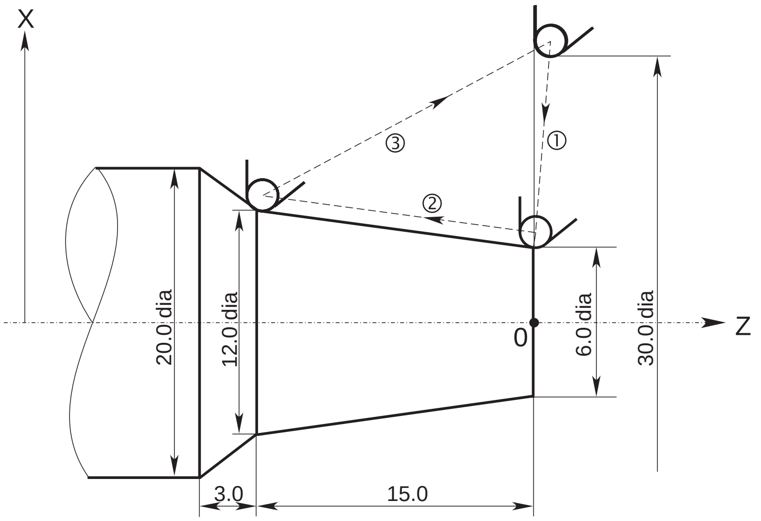
The workpiece position specified by addresses **I** and **K** is the same as that in the preceding block. If **I** and/or **K** is specified with **G40** in the cancel mode, the **I** and/or **K** is ignored.

G40 G01 X____Z____;

G40 G00 X____Z____I____K____;

The numerals following **I** and **K** should always be specified as radius values.

(7) *Example:*



(In G40 mode, radius programming)

```
① G42 G00 X3.0;
② G01 X6.0 W-15.0 F1;
③ G40 G00 X15.0 W15.0 I4.0 K-3.0;
```

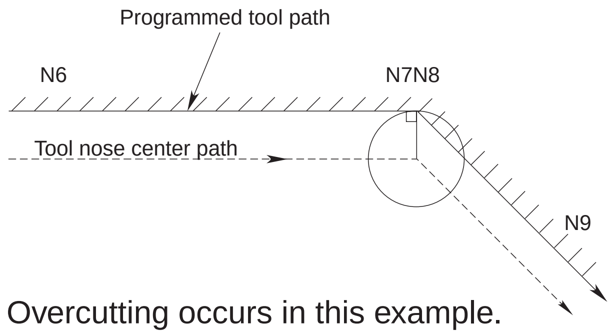
15.2.5 Notes on tool radius compensation

(1) *Two or more blocks without a move command should not be programmed consecutively*.

Blocks without a move commands are:

①	**M05;**	**M code**
②	**S21;**	**S code**
③	**G04 X1000;**	**dwell**
④	**G01 U0;**	**feed distance of zero**
⑤	**G98;**	**G code only**
⑥	**G10 P01 X100 Z200 R50 Q2;**	**offset change**

If two or more of the above blocks are specified consequentively, the tool nose center comes to a position vertical to the programmed path of the end of the preceding block. However, if no movement command is specified (4 above), the above tool motion is attained with one block only.



Overcutting occurs in this example.

(G42 mode)

N6W1000.0;
N7S21;
N8M04;
N9U-1000.0W1000.0;

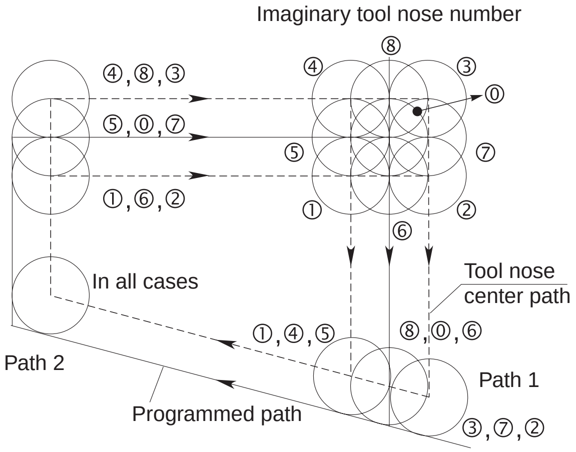
(2) *Compensation with **G90** or **G94***

Tool nose radius compensation with **G90/G94** is as follows:

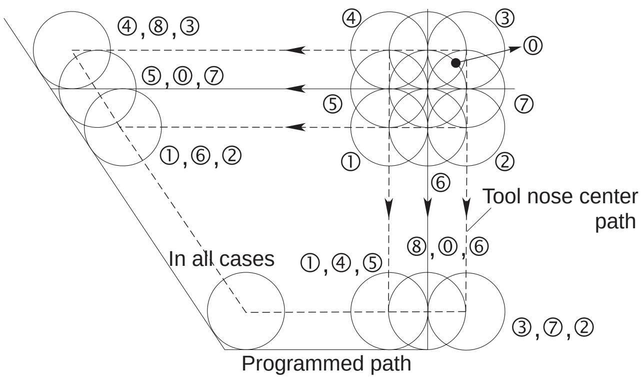
- motion of the imaginary tool nose

For each path in this cycle, the tool nose center path is parallel to the programmed path.

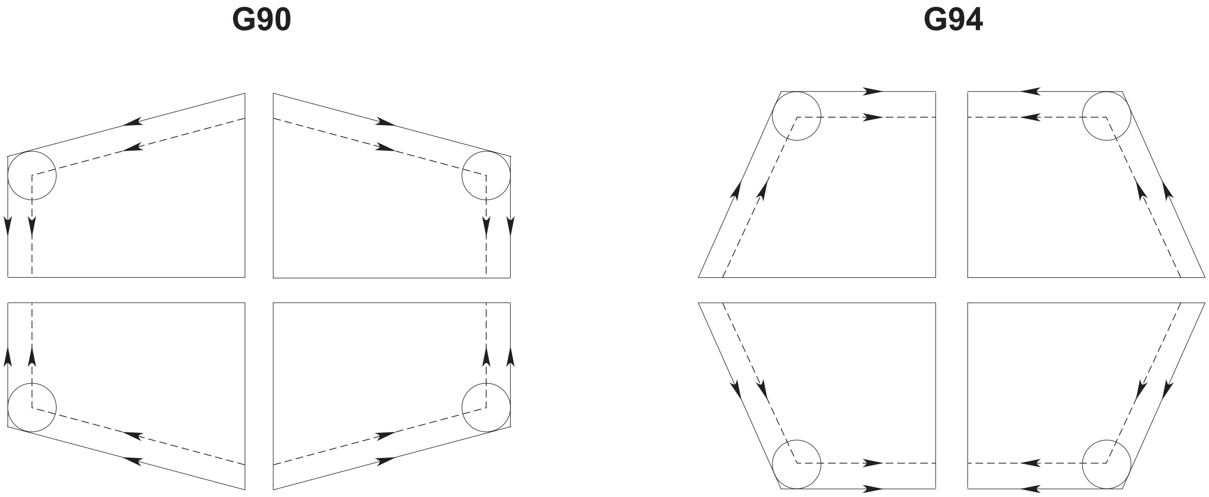
G90



G94



- *the offset direction* is indicated in the figure below regardless of the **G41/G42** mode



- compensation with G71, G72 or G73

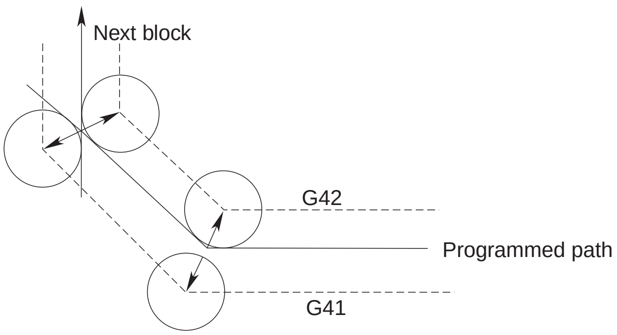
See 14.2.1.

- when G74 or G76 or G78 is specified

Tool nose radius compensation is not performed in this case.

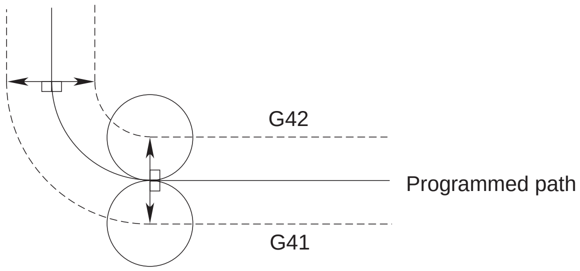
- when chamfering is performed

Movement after compensation is shown below.



- when a corner arc is inserted

Movement after compensation is as follows:

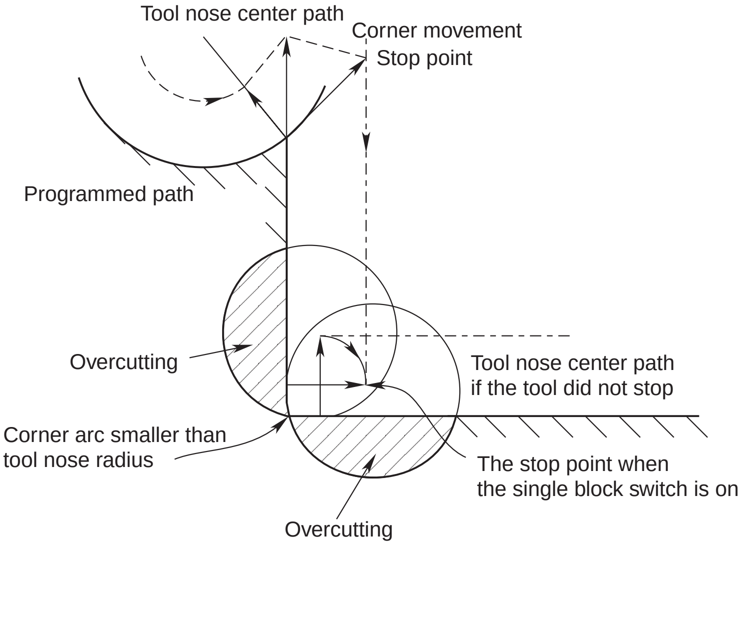


- when the block is specified from the MDI

Tool nose radius compensation is not performed in this case.

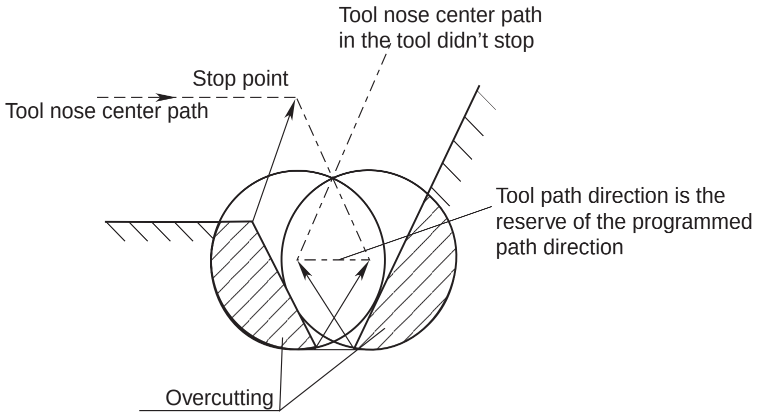
- when machining at an inside corner smaller than the tool nose radius

In this case, the inner offset of the tool will result in overcutting. The tool will stop and alarm (PS41) will be displayed just after starting the next block. If the SINGLE BLOCK SWITCH is on, the tool will stop at the end point of the preceding block.

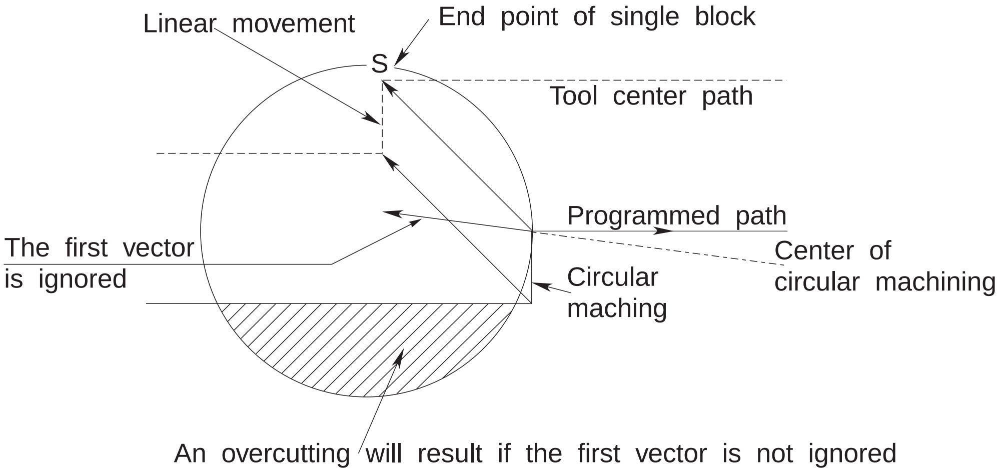


- machining a groove smaller than the tool nose radius

An overcutting will result when machining in a programmed path a groove smaller than the tool nose radius. In this case, alarm (**P/S41**) is displayed and the motion stops.



- when machining a step smaller than the tool radius and this step is an arc, the path of the center of the tool may travel in the reverse of the programmed direction. In this case, the first vector is ignored, and the tool moves linearly to the second vector position. The tool may stop at this point by the SINGLE BLOCK operation. If the step is specified with a line, the offset is properly performed without generating an alarm. (However, uncut parts remain).



15.2.6 Detailed description of tool nose radius compensation

(1) *tool nose **R** center offset vector*

This vector is a two dimensional vector equal to the offset value specified in a **T** code, and is calculated in the CNC. Its dimension changes block by block according to the tool movement. This offset vector (simply called vector hereinafter) is internally created by the control unit as required for proper offsetting and to calculate a tool path with exact offset (by tool nose radius) from the programmed path. This vector is deleted by resetting.

This vector always accompanies the tool as the tool advances. The proper understanding of the vector is essential to accurate programming.

(2) ***G40, G41, G42***

G40, G41 or **G42** is used to generate or to delete vectors. These codes are used together with **G00, G01, G02, G03** or **G33** to specify a mode for tool motion.

G code	Function	Workpiece position
G 40	Tool nose radius compensation cancel	Neither
G 41	Left offset along tool path	Right
G 42	Right offset along tool path	Left

G41 and **G42** specify an offset mode, while **G40** specifies cancellation of the off.

a) *Cancel mode*

The system enters the cancel mode immediately after the power turned on, when the RESET button is pushed or a program is forced to end by executing **M02** or **M30**. In this mode the vector is set to zero, and the path of the center of tool nose coincides with the programmed path.

Each program must end in cancel mode. If it ends in the offset mode, the tool cannot be positioned at the end point, and the tool stops at a location the vector length away from the end point.

b) *Start-up*

When a block satisfies all the following conditions is executed in cancel mode, the system enters the offset mode. This operation is called start-up.

- G41 or **G42** is contained in the block, or has been specified to set the system to **G41** or **G42** mode

- the offset number for tool nose radius compensation is not 00

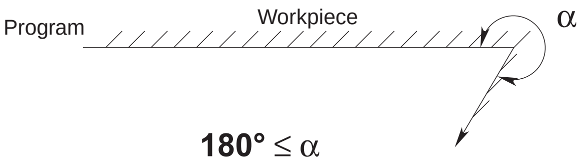
- X or **Z** moves are specified in the block and move distance is not zero

A circular interpolation is not allowed in start-up. If it is specified, alarm (PS34) will occur.

Two blocks are read during the start-up. The first block is executed, and the second block is entered into the tool nose radius compensation buffer. In the SINGLE BLOCK mode, two blocks are read and when the first one is executed, the machine stops.

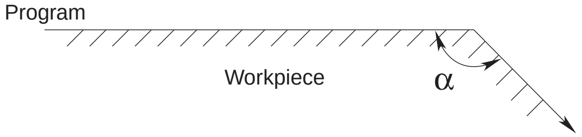
The meaning of *"inner-side"* and *"outer-side"* used later is as follows: An angle of intersection created by two blocks of move commands is referred to as *"inner-side"* when it is over 180° and *"outer-side"*, when it is from 0° to 180°.

(1) ***Inner-side***



180° $\leq \alpha$

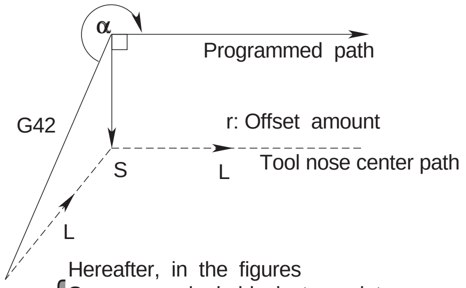
(2) ***outer-side***



0° $\leq \alpha <$ **180°**

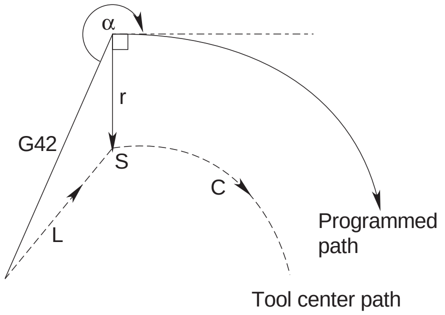
a) Machining an inner wall (180° ≥ α)

Linear → Linear



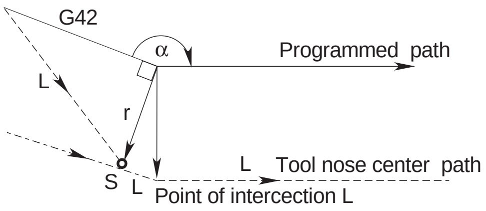
Hereafter, in the figures
S means a single block stop point.
L indicates a straight line.
C indicates circular interpolation..

Linear → Circular



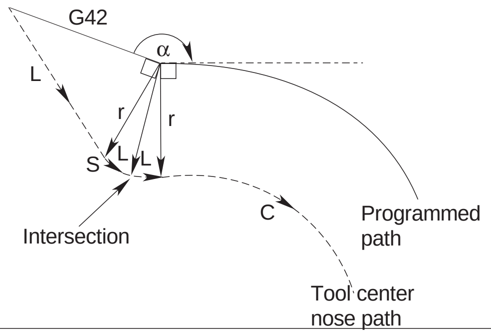
b) Machining an outer wall (at an obtuse angle 90° ≥ α < 180°)

Linear → Linear

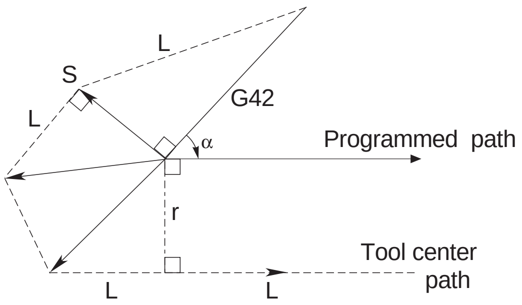


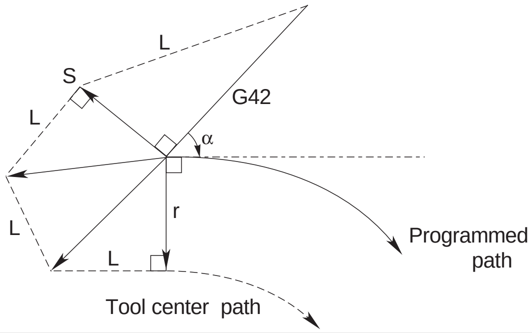
(Note) The intersection is the point of the offset paths of the two blocks where the programmed paths are offset by r.

Linear → Circular

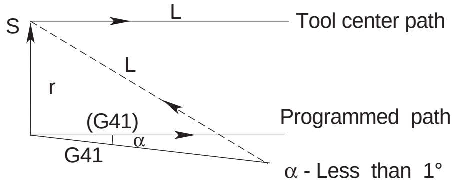


c) Machining an outer wall (at an acute angle α < 90°)





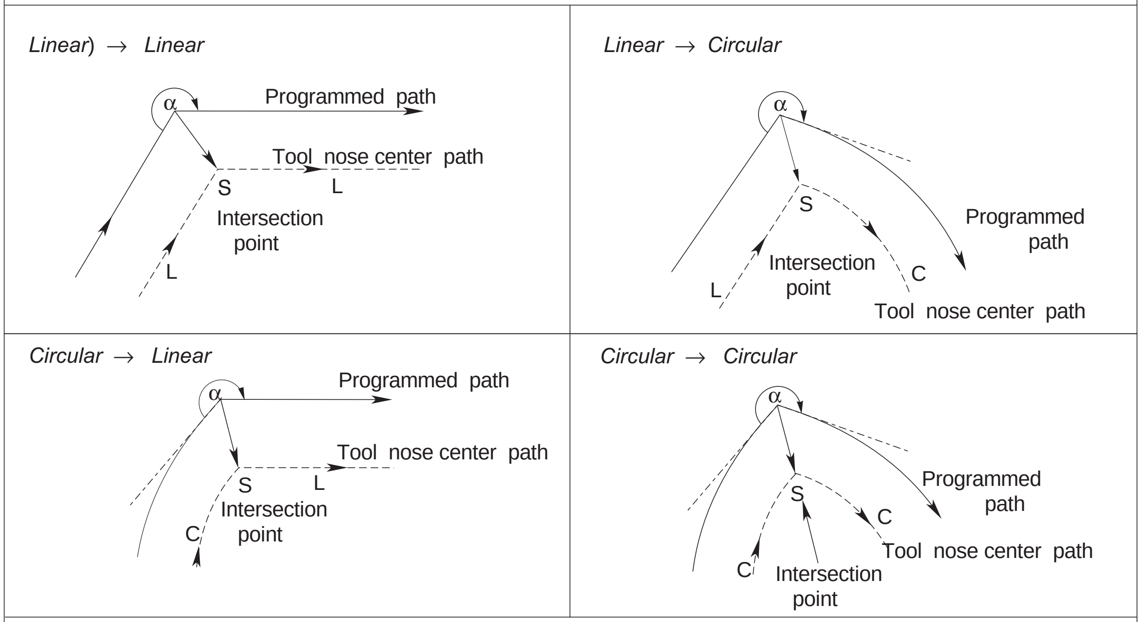
d) When machining an outer wall at a sharp corner (less than 1°) in linear-linear motion (α < 1°)



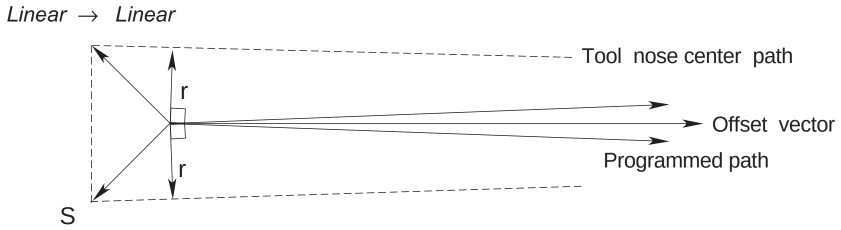
(3) *Offset mode*

In the offset mode, tool offset is provided even during positioning , as well as linear and circular interpolation. In this mode, blocks which do not specify tool movement (such as an **M** function or dwell block) must not be specified consequently). Otherways, overcutting or insufficient cutting will result.

a) Machining an inner wall (α ≤ 180°)



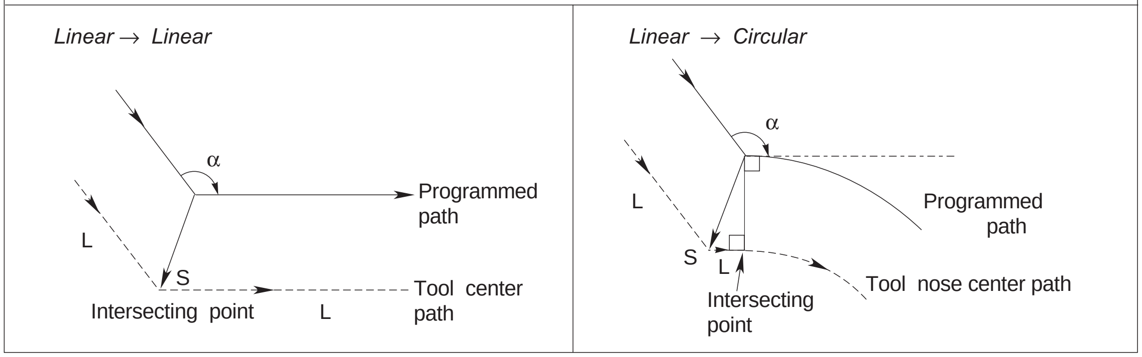
When machining the inner wall at a sharp corner (less than 1°) and the offset vector is abnormally enlarged:



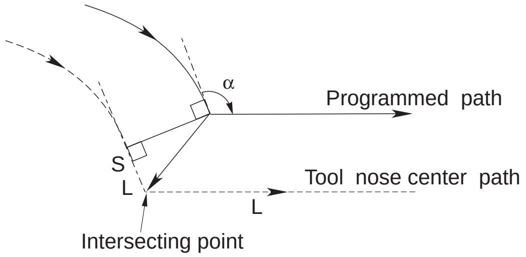
Consider the following in the same way:

Circular → *Linear*
Linear → *Circular*
Circular → *Circular*

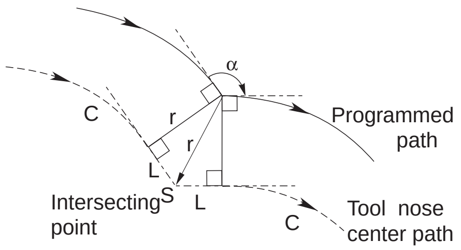
b) Machining an outer wall (at an obtuse angle, 90° ≤ α < 180°)



Circular → Linear

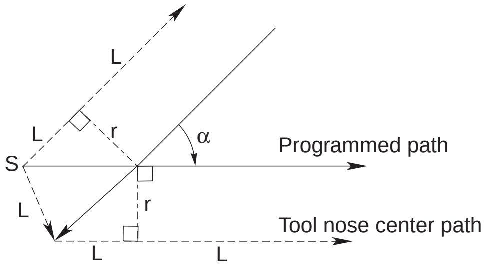


Circular → Circular

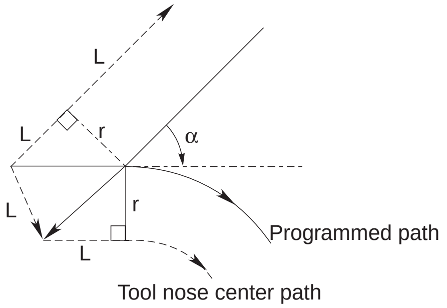


c) Machining an outer wall (at an acute angle, a < 90°

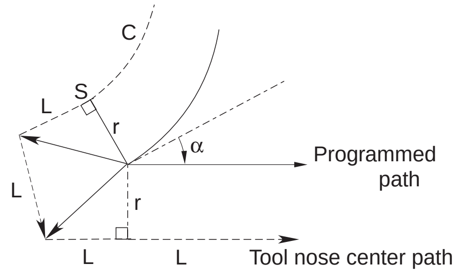
Linear → Linear



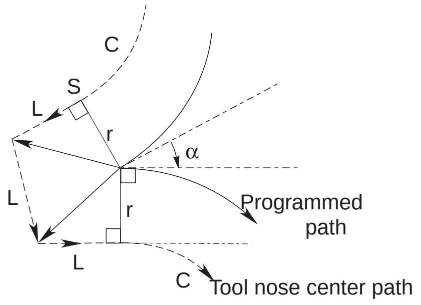
Linear → Circular



Circular → Linear

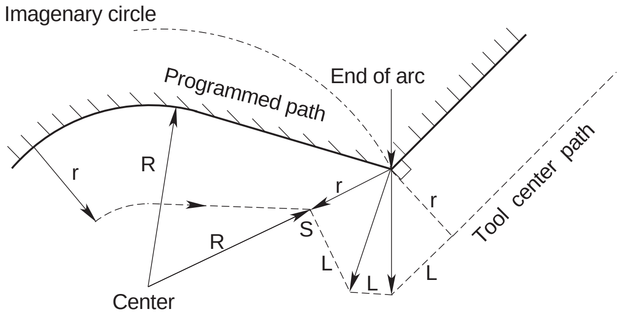


Circular → Circular



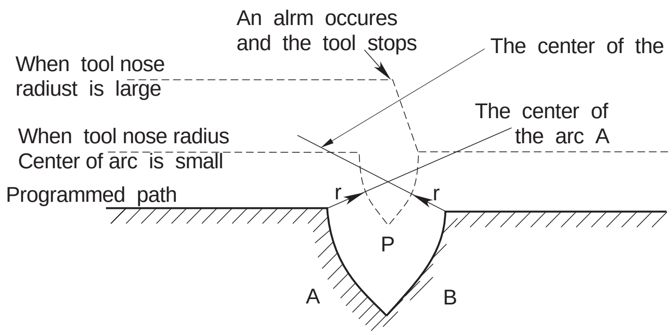
d) Exceptional case

When the end point is not on the arc



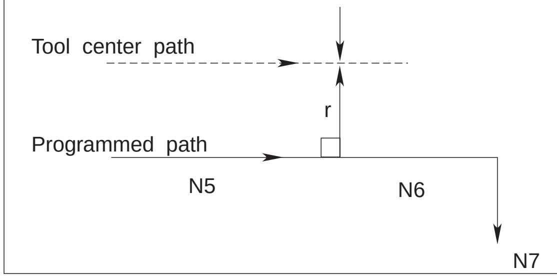
When the line drawn to the arc end point exists, the CNC assumes the imaginary arc shown in the above figure and an vector is created for the imaginary arc and is compensated.
Accordingly, the tool path differs from the compensated path parallel to the line drawn to the end of the arc.
The same concept can be considered in arc to arc motion.

When an inside intersection point is not present



In the figure to the left, when the tool nose radius value is small, the intersection of the offset path of an arc exists but when the radius becomes large, the intersection may not exist. The tool stops at the end of previous block and alarm will be indicated.

In the case of a circular arc in which the center and the start point or the end point coincide.



In the case, alarm No.38 is indicated and the tool stops at the end of the earlier block.

```
(G41)
N5  G01  W1000;
N6  G02  W1000 I0  K0;
N7  G03  U-100C  I-1000;
```

4) ***Offset cancel***

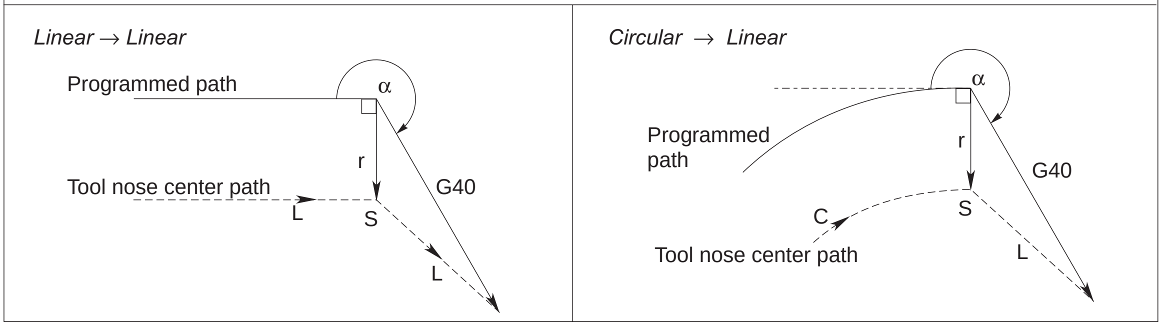
When a block which satisfies one of the following conditions is executed in the offset mode, the system enters the cancel mode. This operation is called the offset cancel.

- G40 is specified

- 0 is specified as the offset number of tool nose radius compensation

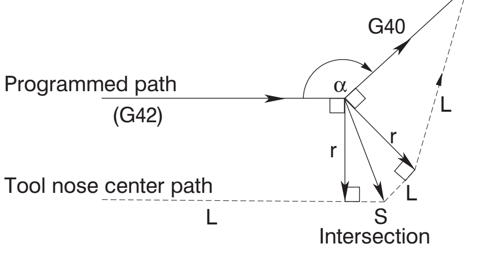
Offset cancellation must not be specified in a circular command (**G02, G03**). If specified, alarm No.34 will be indicated.

a) Machining an inner wall (α ≤ 180°)

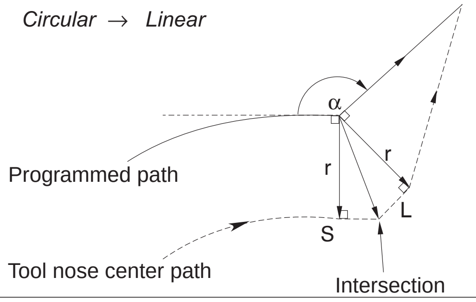


b) Machining an outer wall (at an abtuse angle 90° ≤ α < 180°)

Linear → Linear

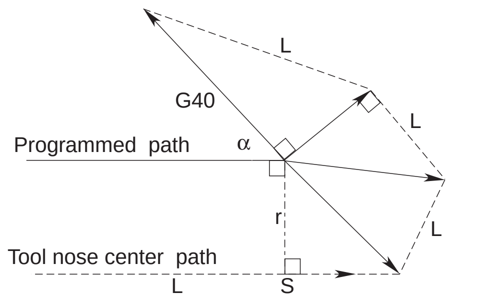


Circular → Linear

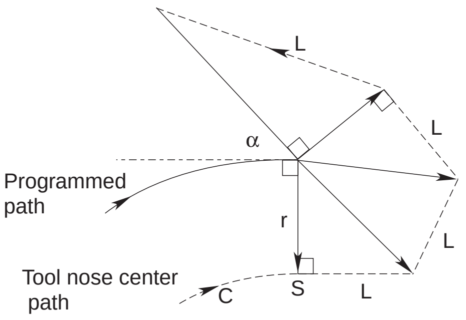


c) When machining an outer wall (at an acute angle α < 90°)

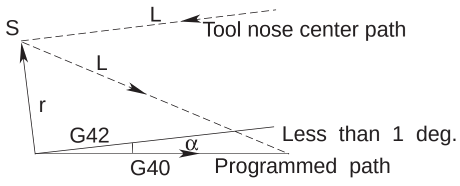
Linear → Linear



Circular → Linear



d) When machining is outer wall at sharp corner of less than 1degree linear in linear - linear motion (a < 1°), an inner wall is assumed.

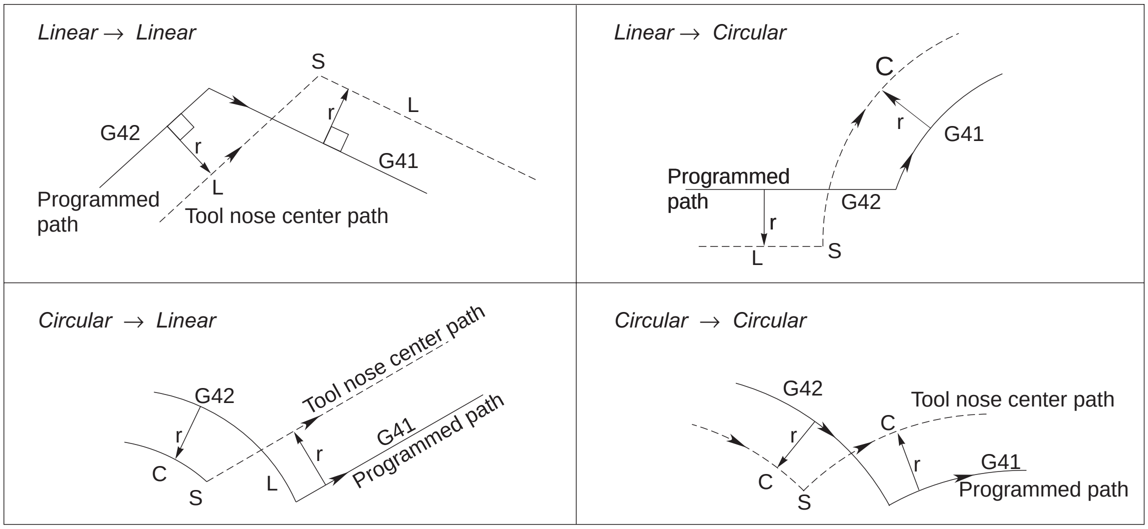


5)*Change of offset direction in offset mode*

The direction of offset is determined by the tool nose radius compensation **G** code (**G41, G42**) and the sign of the offset value.

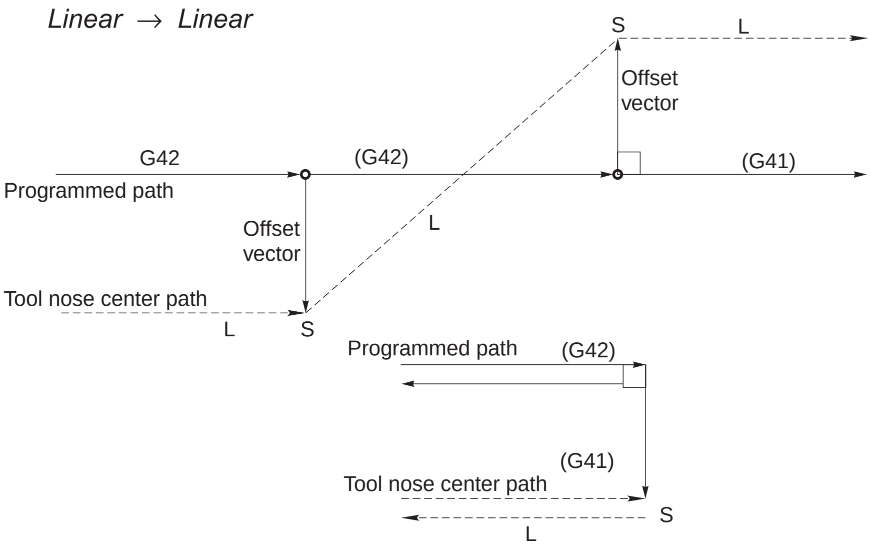
G code	Sign of offset value	
	+	-
G41	Left offset	Right offset
G42	Right offset	Left offset

The following drawinngs explain what happens when the offset direction is changed with **G41** or **G42**. In these examples, the sign of the value is assumed to be positive.

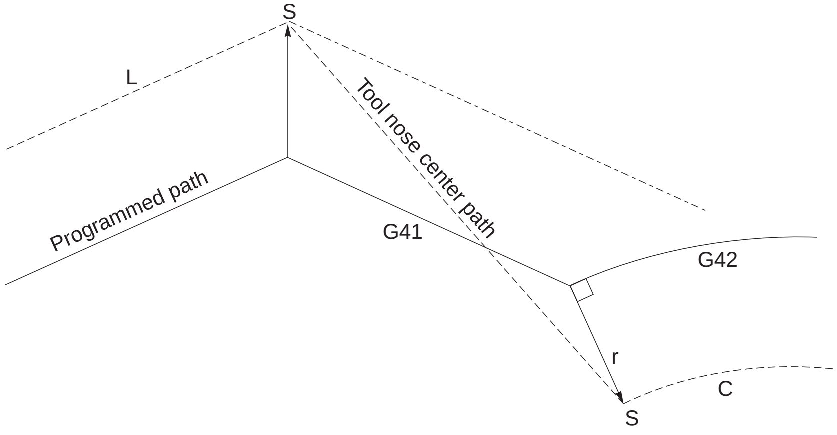


When an intersection is not obtained if offset is normally performed

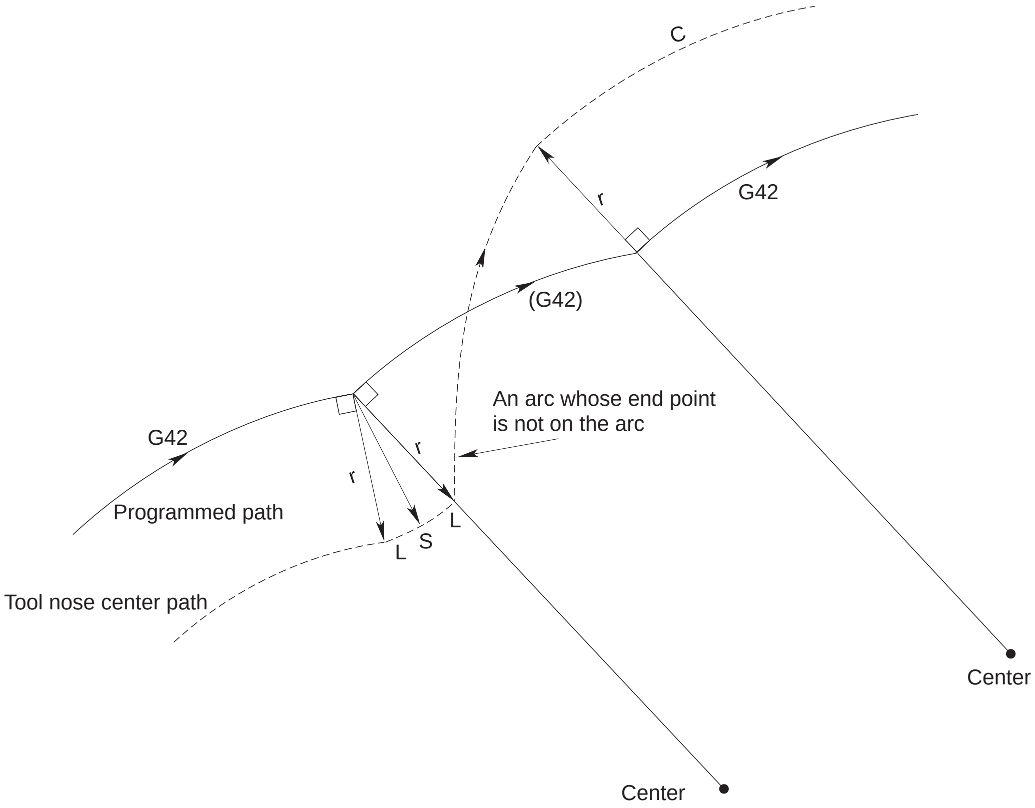
When changing the offset from block **A** to block **B** using **G41** and **G42**, if intersection with the offset path is not obtained, the vector normal to block **B** is created at the start point of block **B**.



Linear → Circular



Circular → Circular

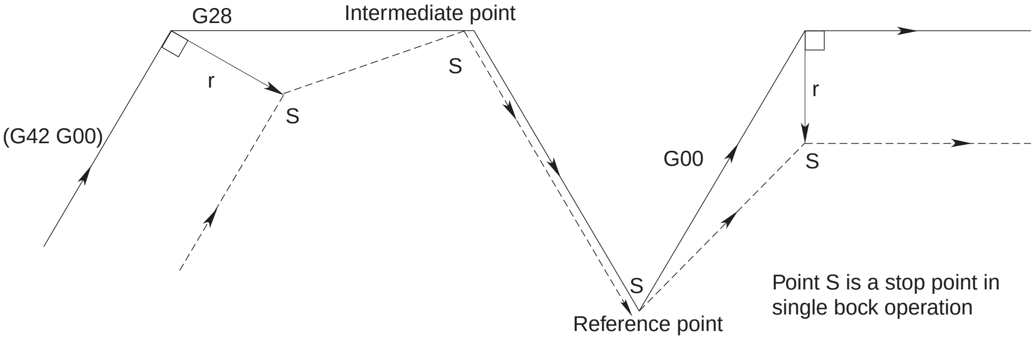


6) ***Temporary offset cancel***

If the command below is specified in the offset mode, a temporary offset cancel is executed and thereafter, the system will automatically restore the offset mode.

G 28 - Automatic return to reference point

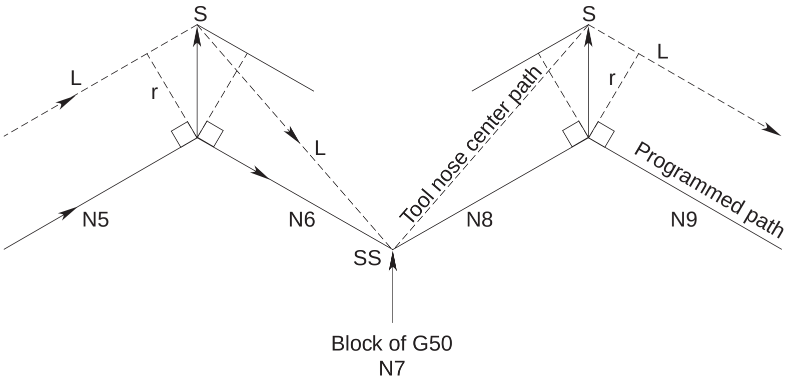
If **G28** is specified in offset mode, offset will be cancelled at the intermediate point, and the offset mode will be automatically restored after reaching the reference point.



7) ***Command to temporarily delete the vector***

When the following commands are specified in the offset in the offset mode, the offset vector is temporarily deleted, then the offset mode will be automatically restored. In this case, the offset cancel motion is not executed but the center of the tool nose goes to the programmed point from the top of the vector at the intersection of the offset point.

a) G50 - *programming of absolute zero point*



(G41)

N5 G01 U300.0 W700.0;

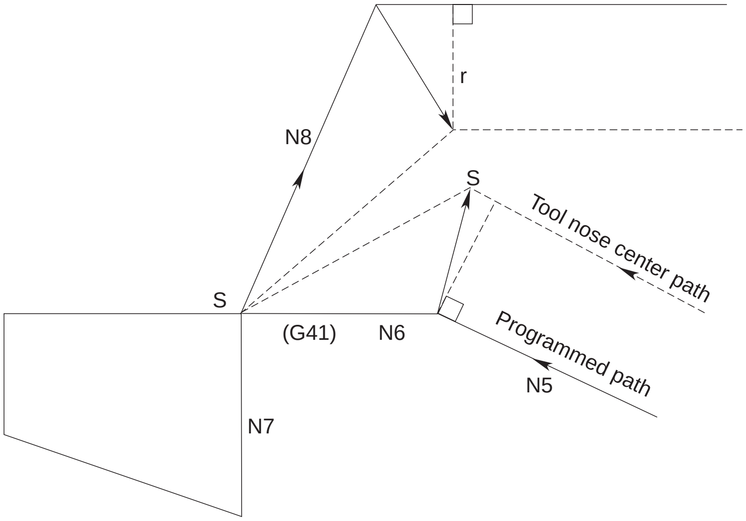
N6 U-300.0 W600.0;

N7 G50 U100.0 Z200.0;

N8 G01 U400.0 Z800.0;

b) G90, G92, G94 - *Canned cycles*

G71 - G76 Multiple repetitive cycles



(G42)

N5 G01 G91 U500.0 W600.0;

N6 W-800.0;

N7 G90 U-600.0 W-800.0 I-300.0;

N8 G01 U1200.0 W500.0;

c) T - *code commanded block*

d) *Double turret mirror image on / off (**G68 /G69**)*

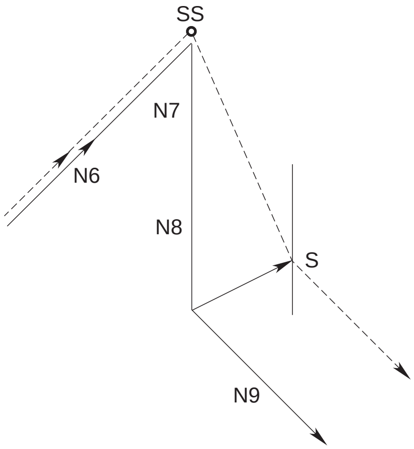
8) *A block not specifying tool movement*

The following blocks do not specify tool movement. In these blocks, a tool will not move even if tool nose radius compensation is actuated:

```
M05;
S21;
G04 X100;
G01 U0;
G98;
G10 P01 X10 Z20 R10 Q01;
```

***a)** when specified at start-up*

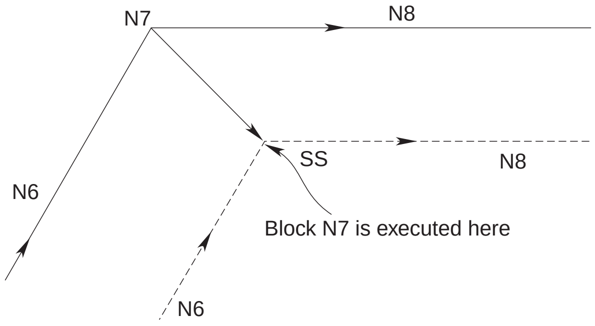
If a block not specifying toll movement is input at start-up, the offset vector is not produced.



```
G40....
.
.
.
N6 U1000.0 W1000.0;
N7 G41 U0;
N8 U1000.0;
N9 U-1000.0 W1000.0;
```

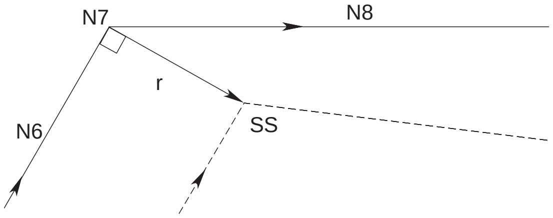
b) *when specified in offset mode*

When a block not specifying tool movement is input in the offset mode, the vector and tool nose center path are the same as if the block was not specified.



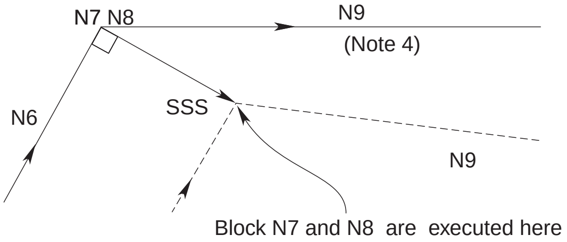
```
N6  G91U2000W1000;

N7  G04P1000;

N8  W1000;
```

When a block not specifying tool movement (the move distance is zero), even if the block is specified singly, tool motion is the same as if more than one block not specifying tool movement.



```
N6  G91U2000W1000;

N7  U0;

N8  W1000;
```

Two or more blocks not specifying tool movement should not be input consecutively. A vector whose lenght is equal to the offset value is produced in a normal direction to the tool motion in the preceding block. Therefore, an overcutting may be result.

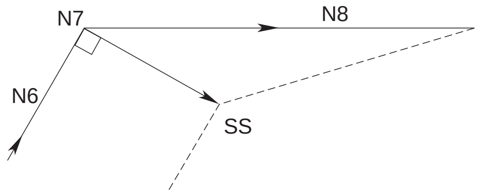


```
N6  G91  X100.0  Y200.0;

N7  S21;

N8  G04  X1.0;

N9  X100.0;
```

SSS means that tool stops three times by single block operation.

c) when specified with offset cancel command

When a block not specifying tool movement is input with an offset cancel command, a vector whose lenght is equal to the offset value is produced in a direction normal to the tool motion specified in the preceding block. This vector is cancelled when the next command is executed.



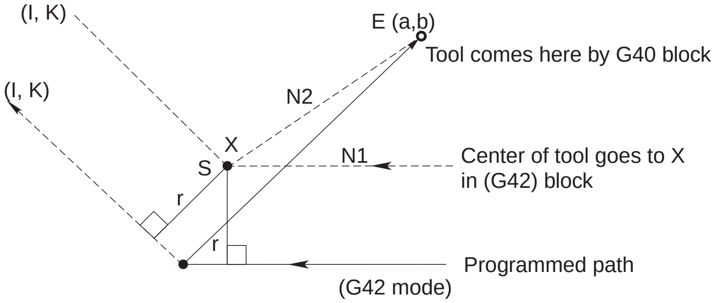
N6 U100.0W100.0;

N7 G40;

N8 U0W10.0;

9) When a block includes G40 and I___ K___;

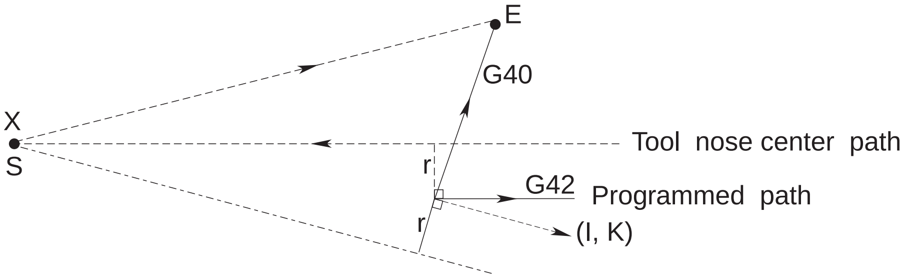
and the mode of the earlier block is **G41** or **G42, CNC** assumes that movement from the end point of the earlier block has been specified in the direction of (**I, K**).



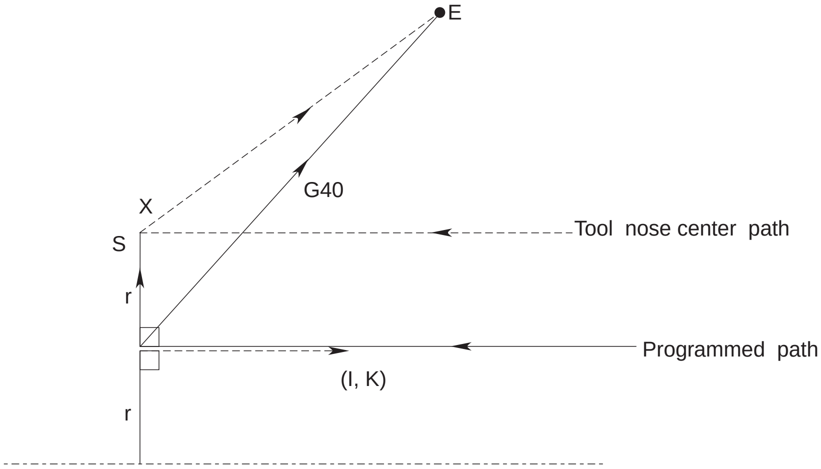
N1 (G42 mode)

N2 G40X a Z b I___ K___ ;

In this case, an intersection is obtained regardless of whether inner or outer wall machining is specified.



When an intersection can not be obtained, the tool moves to a position normal to the programmed path at the end of the earlier block.

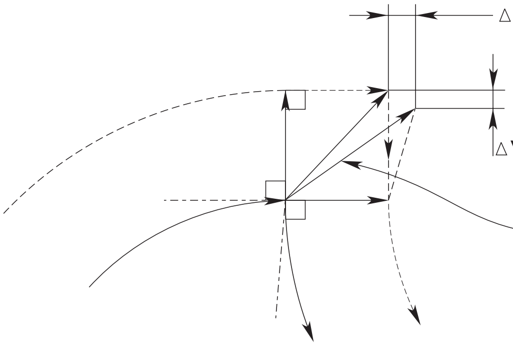


10) Corner movement

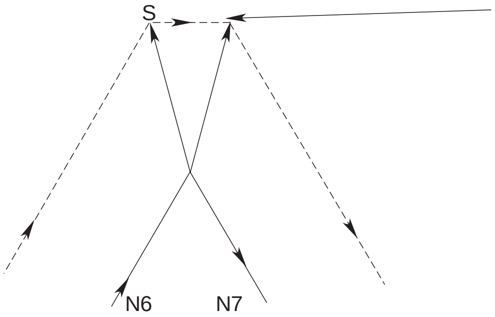
When two or more vectors are produced at the end of a block, the tool moves linaerily from one vector to another. If these vectors almost coincide, the corner movement is not performed and the latter vector is ignored.

If $\mathbf{V_x \leq DV_{limit}}$ **and** $\mathbf{V_z \leq DV_{limit}}$, the latter vector is ignored. The value of **DV$_{limit}$** is specified by parameter No.557 CRCDL. If these vectors do not overlap, a move is provided to turn the corner.

This move belongs to the later block.



This vector is ignored,
if ΔV$_x$ ≤ Δ V limit, ΔV$_z$ ≤ ΔV limit



This move belongs to block N7, thus,the feed rate is equal to that specified in block N7.

If the block N7 is G00 mode, the tool is moved at the rapid traverse rate, while if it is in G01, G02, G03, the tool is moved at the cutting feed rate.

11) Interference check

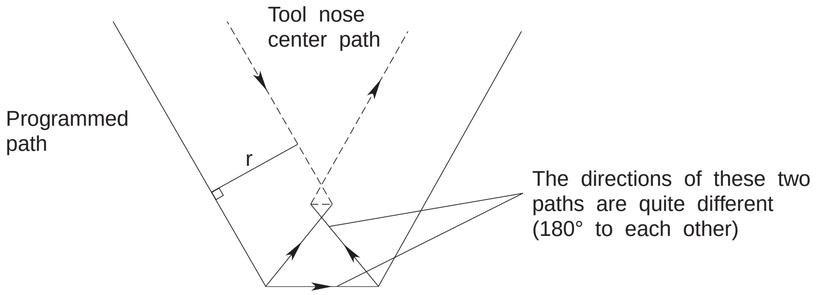
Tool overcutting is called **"interference "**. The interference check function checks for tool overcutting in advance. The interference check is performed even if overcutting does not occur.

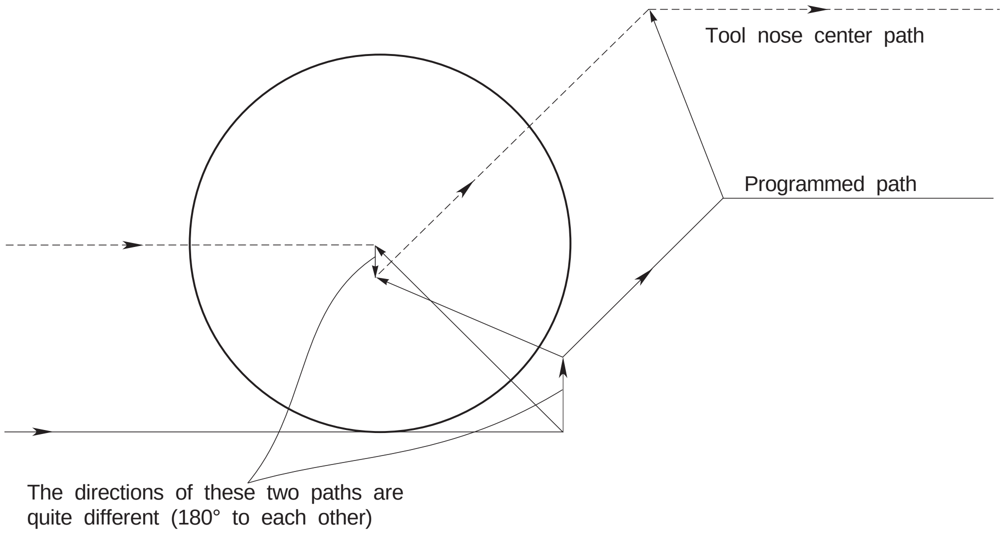
a) *Reference conditions for interference*

1) The direction of the tool nose center path in tool nose radius compensation is different from that of the programmed path.

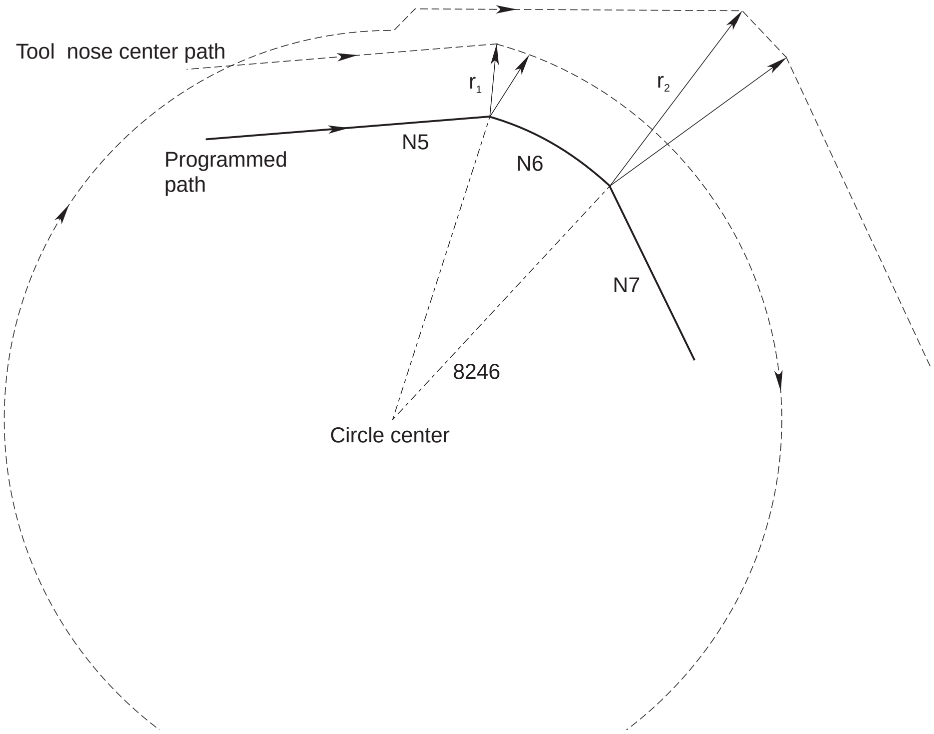
2) The angle between the start point and end point on the tool nose center path is quite different from that between the start point and end point on the programmed path in circular interpolation.

*Example of condition **1)**:*




Tool nose center path
Programmed path
The directions of these two paths are
quite different (180° to each other)

Example of condition ***2)****:*



(G41 mode)

N5 G01 U2000 W8000 T1;

N6 G02 U-1600 W3200 I-8000 K-2000 T2;

N7 G01 U-5000 W2000;

R1=2000 - tool nose radius compensation value for **T1**

R2=6000 - tool nose radius compensation value for **T2**

In the above example, the arc specified in block N6 is placed in the first quadrant. But after tool nose radius compensation, the arc passes through four quadrants.

b) *Correction of interference in advance*

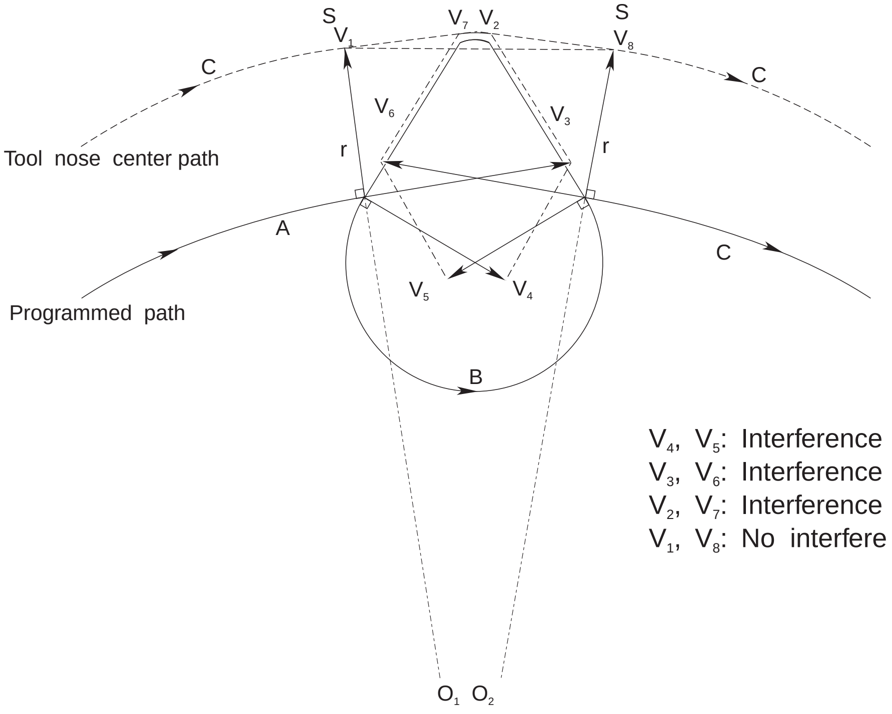
1) *Removal of the vector causing the interference*

When tool nose radius compensation is performed for blocks **A, B** and **C** - $\mathbf{V_1}$, $\mathbf{V_2}$, $\mathbf{V_3}$ and $\mathbf{V_4}$ and $\mathbf{V_5}$, $\mathbf{V_6}$, $\mathbf{V_7}$ and $\mathbf{V_8}$ are vectors between **B** and **C** are produced, the nearest vectors are checked first. If an interference occurs, they are ignored. But if the vectors to be ignored due to interference are the last vectors at the corner, they can not be ignored.

Interference check between vectors $\mathbf{V_4}$ and $\mathbf{V_5}$. Interference $\mathbf{V_4}$ and $\mathbf{V_5}$ are ignored.
Interference check between vectors $\mathbf{V_3}$ and $\mathbf{V_6}$. Interference $\mathbf{V_3}$ and $\mathbf{V_6}$ are ignored.
Interference check between vectors $\mathbf{V_2}$ and $\mathbf{V_7}$. Interference $\mathbf{V_2}$ and $\mathbf{V_7}$ are ignored.
Interference check between vectors $\mathbf{V_1}$ and $\mathbf{V_8}$. Interference can not be ignored.

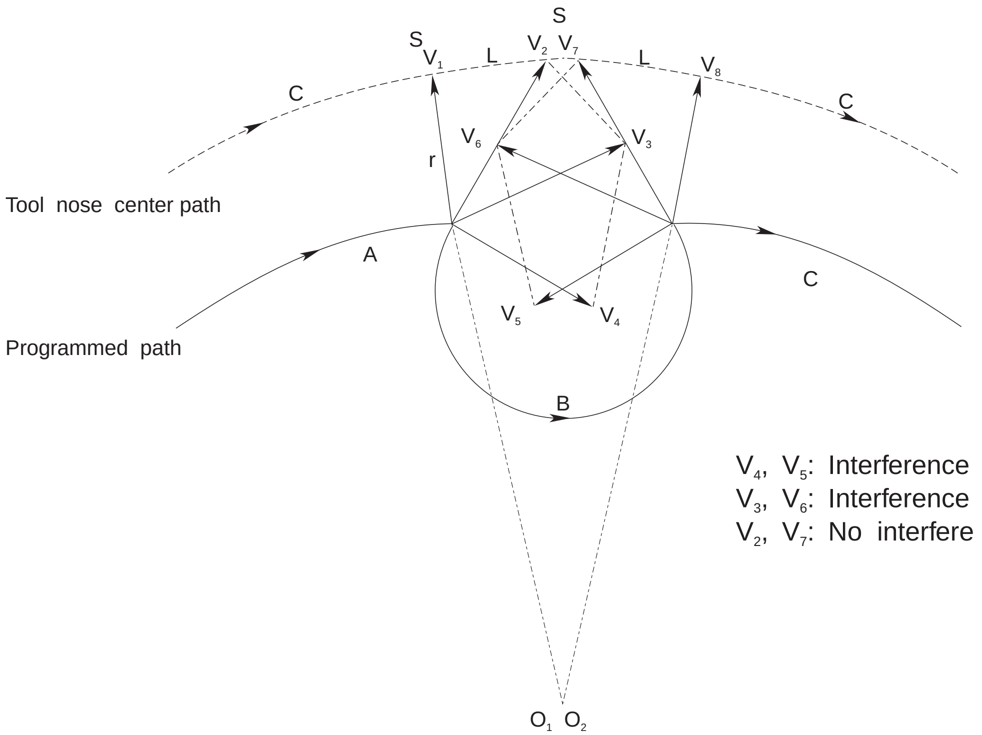
If while checking a vector with no interference is detected, subsequent vectors are not checked. If block **B** is a circular movement, a linear movement is produced if the vectors are interfered.

*Example **1)*** The tool moves linearly from $\mathbf{V_1}$ to $\mathbf{V_8}$


S
V_1
V_7 V_2
S
V_8
C
C
V_6
V_3
r
r
Tool nose center path
A
C
V_5
V_4
Programmed path
B
V_4, V_5: Interference
V_3, V_6: Interference
V_2, V_7: Interference
V_1, V_8: No interfere
O_1 O_2

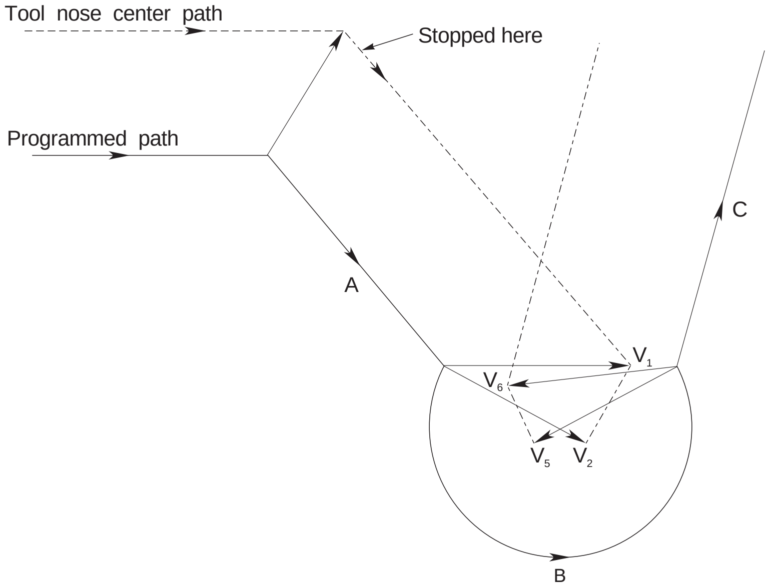
Example ***2)*** The tool moves linearly as follows;

tool path: $\mathbf{V_1 \rightarrow V_2 \rightarrow V_7 \rightarrow V_8}$



V_4, V_5: Interference
V_3, V_6: Interference
V_2, V_7: No interfere

2) *if the interference occurs after correction 1), the tool stops and alarm (PS41) is indicated*

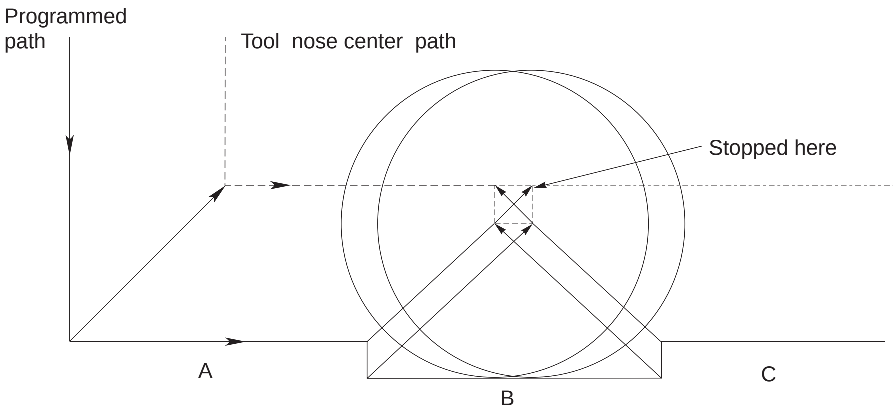


V_2, V_5 : Interference
V_1, V_6 : Interference

c) *Checking is performed although interference does not actually occurs*

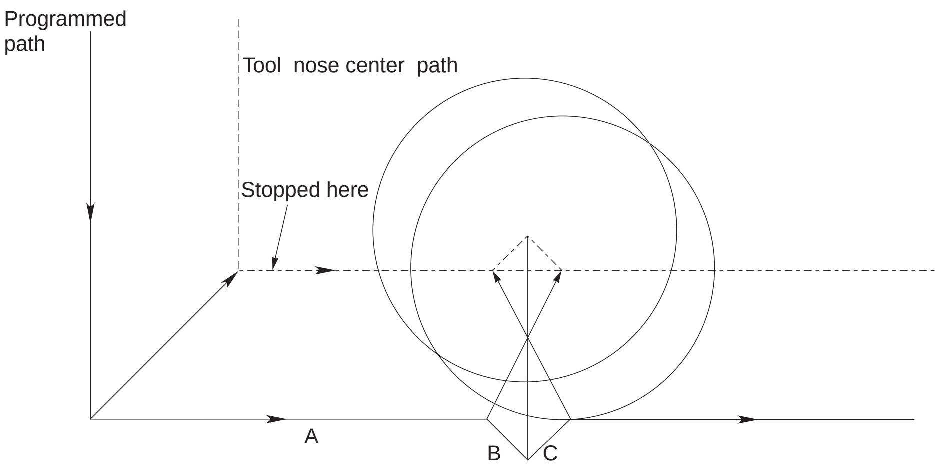
There are many examples, for instance the following:

1) A shallow depth, smaller than the tool nose radius



Although interference does not occur, the tool is stopped with alarm No.41 because the direction of the tool path is not the same as the programmed path.

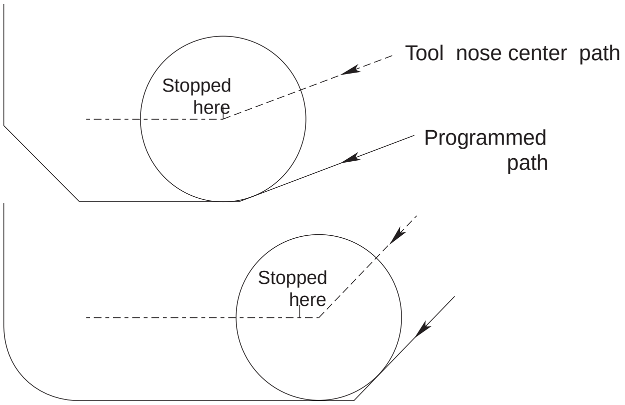
2) A groove depth, smaller than the tool nose radius



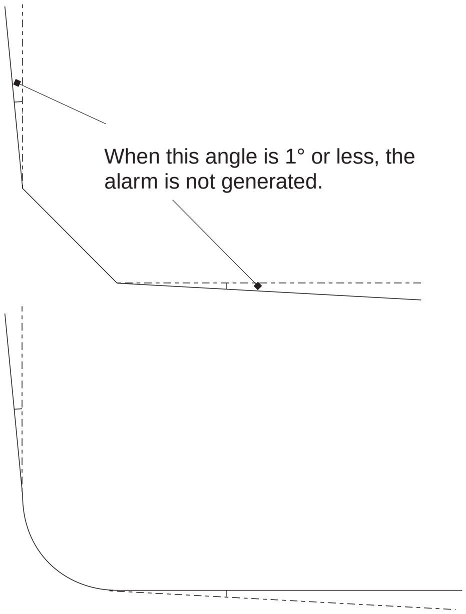
12) Correction in chamfering and corner arcs

a) *In chamfering or corner arcs, tool nose radius compensation* can be only be performed when an ordinary intersection exists at the corner. In offset cancel mode, a start-up block or when exchanging the offset direction, compensation can not be performed, an alarm No.39 is indicated and the tool is stopped.

b) *In inner chamfering or inner corner arcs*, if the chamfering value or corner arc value is smaller than the tool nose radius value, the tool is stopped with an alarm No.39.

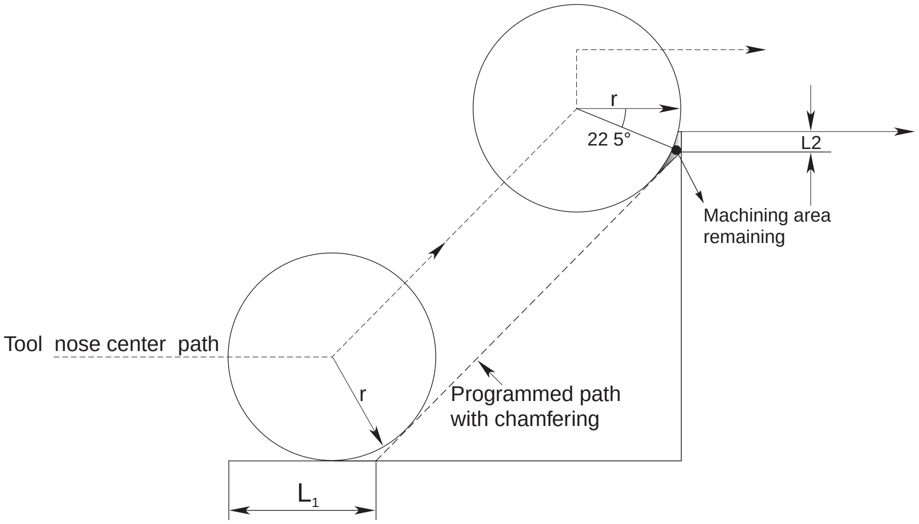


c) *The valid inclination angle of the programmed path in the blocks* before and after the corner is one degree or less so that the alarm does not occur.

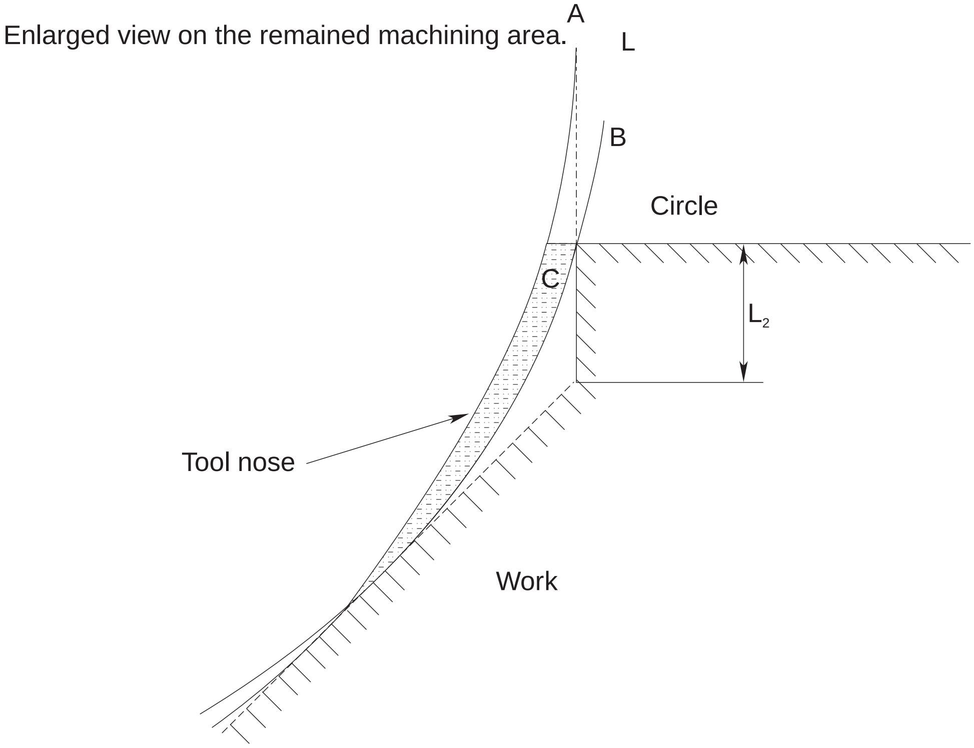


***d)** When machining area remains*

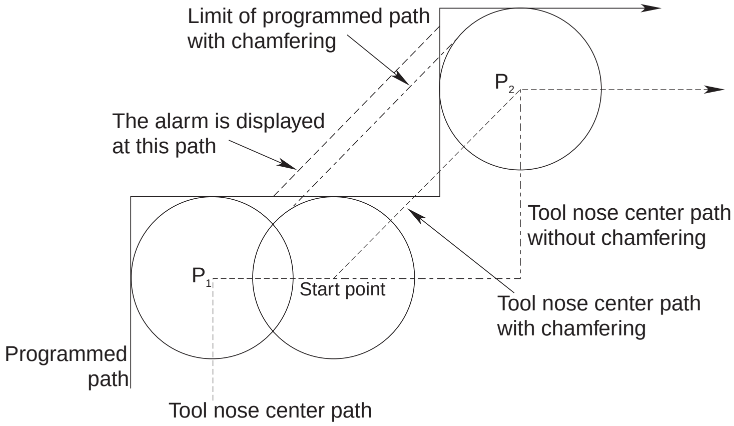
1) *The following example shows a machining area which can not be cut*



In inner chamfering, if the position of the programmed path is not a part of the chamfering but is in the following range, insufficiently cut area will remain.



2) *Alarm **PS52** or **55** is generated :*

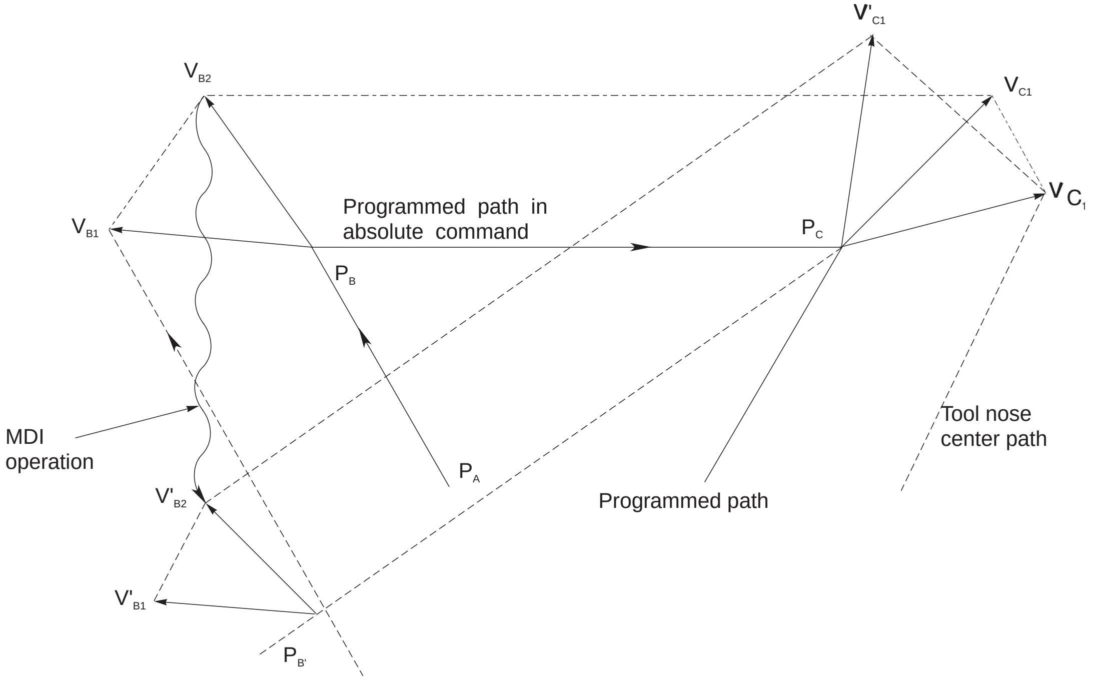

Limit of programmed path with chamfering
The alarm is displayed at this path
P2
P1
Start point
Tool nose center path without chamfering
Tool nose center path with chamfering
Programmed path
Tool nose center path

In outer chamfering with an offset, a limit is imposed on the programmed path. Path during chamfering coincides with the intersection points **P1** or **P2** without chamfering respectively, outer chamfering is limited. If the chamfering value is more than the limit value as specified, alarm **PS51** or **PS52** will be indicated.

13) Tool nose radius compensation for MDI input

Compensation is not performed in this case,. However, when automatic operation is temporarily stopped by the SINGLE BLOCK function, **MDI** operattion is performed, and the automatic operation starts again, the tool path is as follows:

In this case, the vectors at the start point of the next block are translated and the other vectors are produced by the next two blocks.

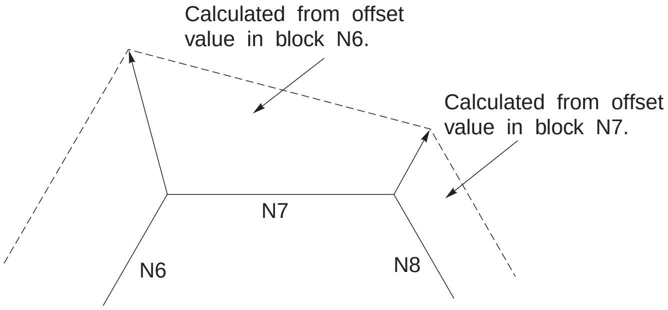


When points $\mathbf{P_A}$, $\mathbf{P_B}$ and $\mathbf{P_C}$ are programmed in absolute command, the tool is stopped by the SINGLE BLOCK function after executing the block from $\mathbf{P_A}$ to $\mathbf{P_B}$ and the tool is moved by **MDI** operation. Vectors $\mathbf{V_{B1}}$ and $\mathbf{V_{B2}}$ are translated to $\mathbf{V'_{B1}}$ and $\mathbf{V'_{B2}}$ and offset vectors are recalculated for the vectors $\mathbf{V_{C1}}$ and $\mathbf{V_{C2}}$ between block $\mathbf{P_B}$ **-** $\mathbf{P_C}$ and $\mathbf{P_C}$ **-** $\mathbf{P_{D.}}$ However, since vector $\mathbf{V'_{B2}}$ is not calculated again, compensation is accurately performed from point $\mathbf{P_C}$.

14) General precautions for offset operations

***a)** changing the offset value*

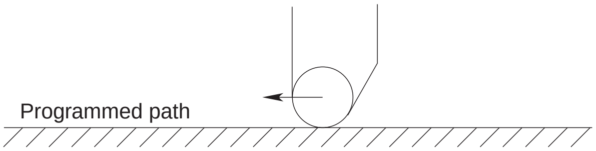
In general, the offset value is changed in cancel mode, or when changing tools. If the offset value is changed in offset mode, the vector at the end point of the block is calculated for the new offset value. The imaginary tool number and tool offset number are changed in the same way.


Calculated from offset value in block N6.
Calculated from offset value in block N7.
N7
N6
N8

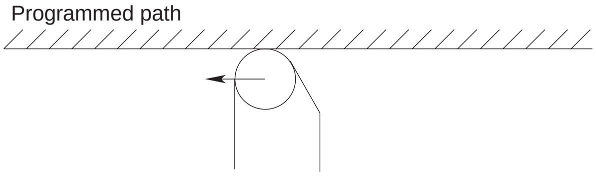
***b)** the polarity of the offset amount and tool nose center path*

When a negative offset value is specified, the program is executed by exchanging **G41** for **G42** or **G42** for **G41**. A tool machining an inner profile will machine the outher one, and vice versa.

When a program specifies a tool path as shown in **a)**, the tool will move as shown in **b)** if a negative offset is specified, and vice versa.



(a)



(b)

15.3 Changing of Tool Offset Amount (G10)

Offset values can be input by a program using the following command:

G10 P____X____Z____R____Q____;

or

G10 P____U____W____C____Q____;

where:

P - ofset number

For wear offset amount : P=wear offset number

For geometry offset amount : P=100+ geometry offset number

X **-** absolute offset value on **X** axis

Z **-** absolute offset value on **Z** axis

U **-** incremental offset value on **X** axis

W **-** incremental offset value on **Z** axis

R **-** tool nose radius offset value (absolute)

C **-** tool nose radius offset value (incremental)

Q **-** imaginary tool nose number

In an absolute command the values, specified in addresses **X** and **Z** are set as the offset value corresponding to the offset number specified by address **P**.

In an incremental command, the value specified in addresses **U** and **W** is added to the current offset value corresponding to the offset number.

16. MEASUREMENT

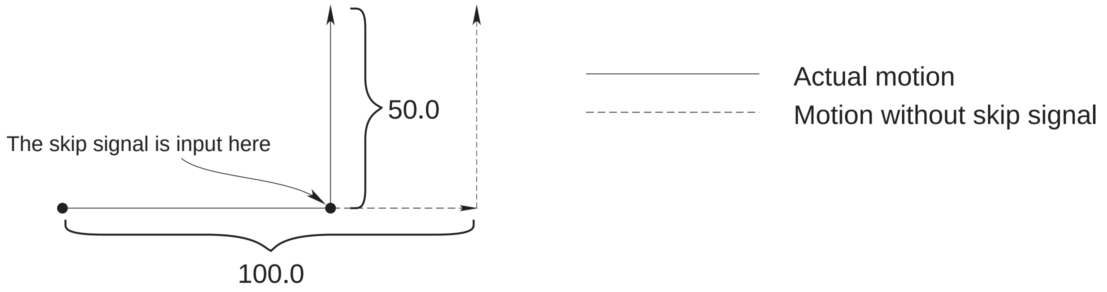
16.1 Skip function (G31)

Shift function following **G31** specifies linear interpolation as in **G01**. Input of the skip signal during execution of this command interrupts the rest of the block and executes the next block. **G31** is an one-shot command. The motion after input of the skip signal depends on whether the next block contains an incremental or absolute command.

a) *When the next block contains an incremental command*

The motion of the next block is incremental from the interrupted position.

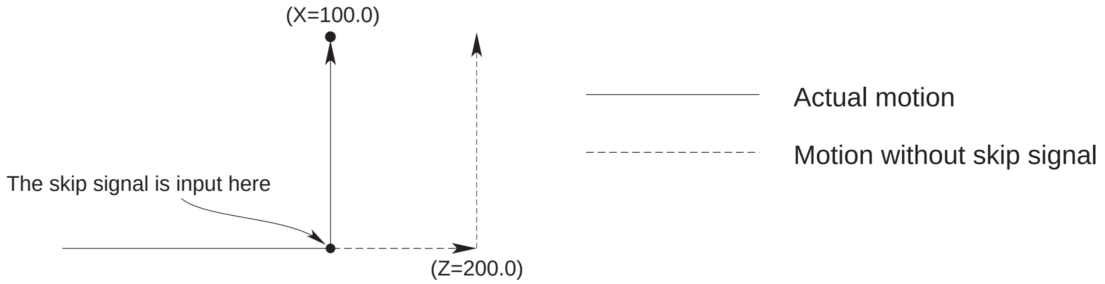
Example: **G98 G31 W100.0 F100;**
W50.0;



b) *When the next block contains an absolute command*

The tool moves along the specified axis to the specified position. The position of the other axis remains the same as when the skip signal was input.

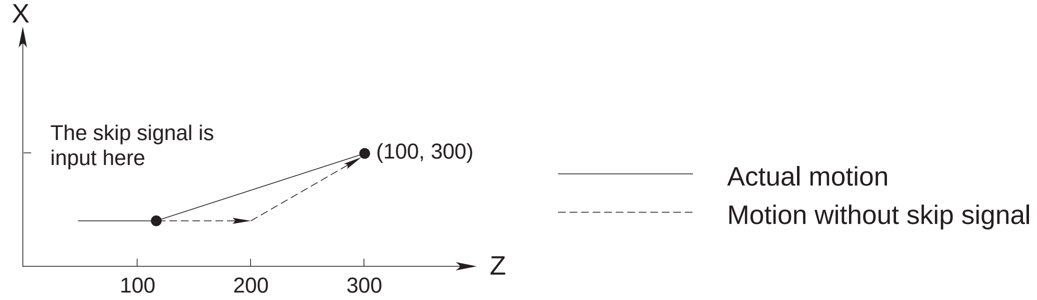
Example: **G31 Z200.0 F100.0;**
X100.0;



c) *When the next block contains an absolute command specifying two axes*

The tool moves to the specified position regardless of input of the skip signal.

Example: **G31 Z200.0 F100.0;**
X100.0 Z300.0;



The custom macro can use the coordinate values of the position where the skip signal was issued, since they are stored in system variables **#5061** and **#5062** of the custom macro.

#5061 - X coordinate value
#5062 - Z coordinate value

The **G31** can not be commanded when the tool nose radius compensation is used. When the feedrate specified in per minute feed set by a parameter, automatic acceleration/deceleration override and DRY RUN are invalid.

16.2 Automatic Tool Offset (G36, G37)

When a tool is moved to the measured position by the execution of a command given to CNC, the CNC automatically measures the difference between the current coordinate value and the coordinate value of the, measured position and uses it as the offset amount for the tool. This distance may be used as the tool offset value.

a) *Coordinate system*

When the tool moves to a position for a measurement, the coordinate system must be set in advance.

b) *Movement to a measured position*

A movement to a measured position is performed by specifying in the **MDI** or **AUTO** mode as follows:

G36 Xx_a;

or

G37 Zz_a;

In this case, the measured position should be $\mathbf{x_a}$ or $\mathbf{z_a}$ (absolute command)

Execution of this command moves the tool at the rapid traverse rate towards the measured position, lowers the feedrate halfway, then continues to move it until the approach end signal from the measuring instrument is issued. When the tool tip reaches the measurement position the measuring instrument sends a signal to the CNC which stops the tool.

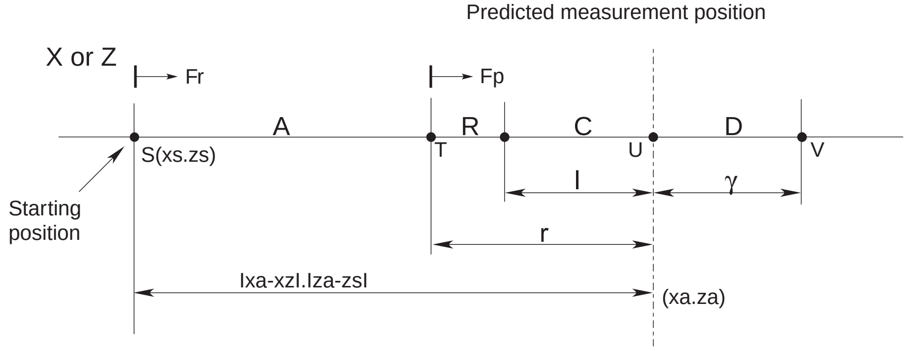
c) *Offset*

The new tool offset is the sum of the current tool offset and the difference between the coordinate value (**a** or **b**) when the tool has reached the measured position and the value of $\mathbf{x_a}$ or $\mathbf{z_a}$ specified in **G36 Xx_a** or **G37 Zz_a**.

Offset amount **X** = current offset amount $\mathbf{X + (\alpha - x_a)}$

Offset amount **Z** = current offset amount $\mathbf{Z + (\beta - z_a)}$

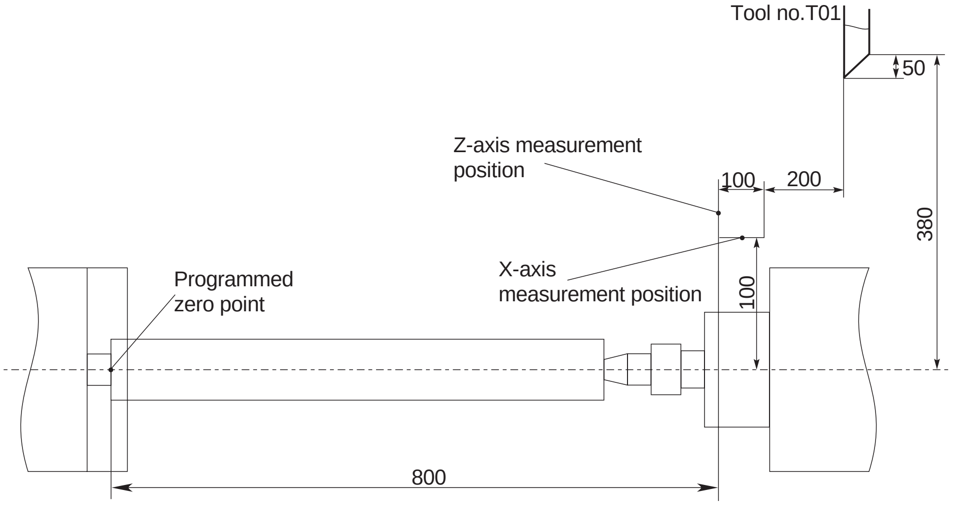
d) *Feedrate and alarm*



Fr: Rapid traverse rate
Fp: Measurement feedrate
(set by parameter)

The tool moves at the rapid traverse rate across area **A** from the starting position towards the measured position predicated by $\mathbf{x_a}$ or $\mathbf{z_a}$ in **G36** or **G37**. Then the tool stops at point $\mathbf{T(x_a - \gamma_x \text{ or } z_a - \gamma_z)}$ and moves at the measured feedrate set by a parameter across areas **B, C** and **D**. If the approach end signal turns on during movement across area **B**, an alarm is generated. If the approach end signal does not turn on before point **V**, an alarm is generated and tool stops at point **V**.

Example:



Offset amount *(before measurement)*		**Offset amount** *(after measurement)*
X	100000	98000
Z	0	4000

G50 X760000 Z110000; Programming of absolute zero point (coord. system setting)

S01 M03 T0101; Specifies tool **T1**, offset number **1**, and spindle revolution

Z850000; Moves some distance away from the measurement position

G36 X200000; Moves to the measured position. If the tool has reached the measured position at X19800: since the correct measurement position is 200 mm, the offset amount is altered by 198.0 - 200.0= - 2.0 mm

G00 X204000; Retracts a little along the **X** axis.

G37 Z800000; Moves to the **Z** axis measurement position. If the tool has reached the measured position at Z804000, the offset amount is altered by 804.0 - 800.0=4 mm

T0101; Further offset by the difference. The new amount becomes valid when the **T** code is specified again.

When there is no **T** code command before **G36** or **G37**, an alarm No.81 is generated.

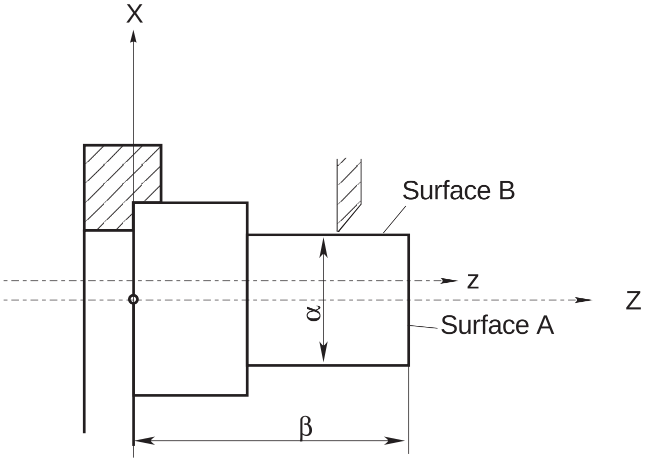
When a **T** code is specified in the same block as **G36** or **G37**, an alarm No.82 is generated.

Measurement speed, **γ** and **ε** are set by parameters. **ε** and **γ** must be positive numbers such as **$\gamma > \varepsilon > 0$**.

Before using the **G36** and **G37**, the tool nose radius compensation must be cancelled.

16.3 Direct setting of the tool compensation value

The following method describes the setting of the tool compensation value (the distance between the standard null-point in programming and the tool nose).



a) In manual mode move the tool to surface **A**;

b) Pull out the tool along the **X** axis without moving along the **Z** axis and stop the spindle;

c) Measure the distance **β** between the standard null-point and the surface **A**;

d) Select the screen "OFFSETS" and position the cursor to the appropriate offset. Press the **[Z]** key, enter the measured distance **β** and press the [INSERT] key;

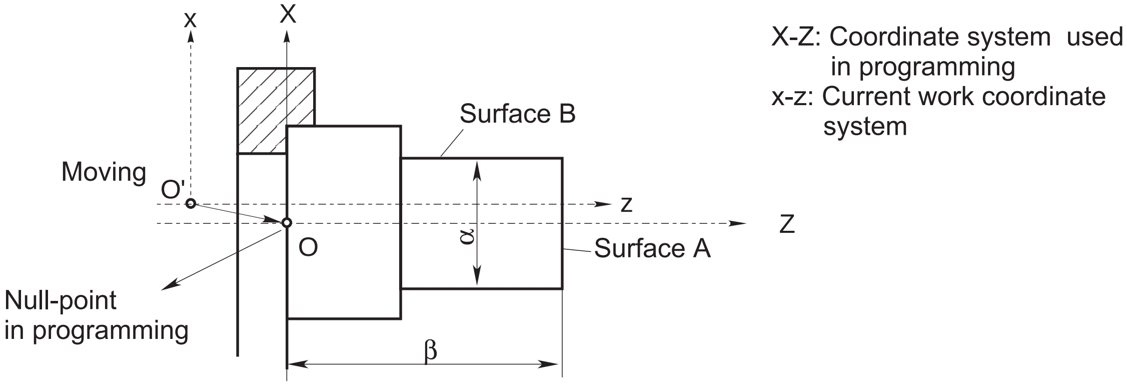
e) In manual mode move the cursor to surface **B**;

f) Pull out the tool along the **Z** axis without moving along the **X** axis and stop the spindle;

g) Measure the diameter a between the standard null-point and the surface B;

h) Select the screen "OFFSETS" and position the cursor to the appropriate offset; Press the **[X]** key, enter the measured distance **α** and press the [INSERT] key.

16.4. Direct setting of the coordinate system shift value.


x
X
X-Z: Coordinate system used in programming
x-z: Current work coordinate system
Surface B
Moving
O'
z
Z
α
O
Surface A
Null-point in programming
β

a) In manual mode move the tool to surface **A**;

b) Pull out the tool along the **X** axis without moving along the **Z** axis and stop the spindle;

c) Measure the distance **B** between the standard null-point and the surface **A**;

d) Select the screen "COORDINATE SYSTEM SHIFT"; Press the **[Z]** key, enter the measured distance **β** and press the [INSERT] key;

e) In manual mode move the tool to surface **B**;

f) Pull out the tool along the **Z** axis without moving along the **X** axis and stop the spindle;

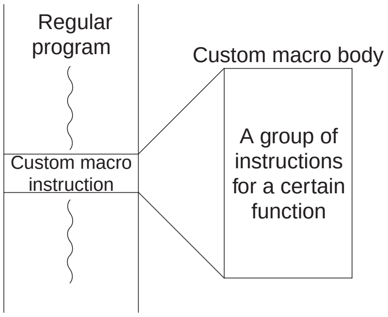
g) Measure the diameter **α** between the standard null-point and the surface **B**;

h) Select the screen "COORDINATE SYSTEM SHIFT"; Press the **[X]** key, enter the measured distance **α** and press the [INSERT] key.

17. CUSTOM MACRO

The custom macro instructions are functions which may be called out from the program by specifying of the definite parameters. It is important in case of using of the custom macros the usage of the variables, the operations which can be performed on variables and actual values can be assigned to the variables.

The custom macro instructions can be grouped in subprograms which can be calling by the command **M98**.



17.1 Variables

A variable can be designated at an address instead of a number. When the macro is executed, the calculated value of the variable is commanded. The variables which can be used are determined by the variable numbers.

17.1.1 Expression of variable

The variable is expressed by **#** followed by a variable number as follows:

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

Example: **#5, #109, #1005**

17.1.2 Reference of variables

The variables are used for substitution of the numbers specified in the addresses.

Examples:

F#103	equivalent to F13 when #103=13
Z- #110	equivalent to Z-250 when #110=250
G#130	equivalent to G03 when #130=03

To substitute the variable for the variable number, designate **#9100** instead of **##100.**

Example:

When #100=105 and #105 = - 500
X#9100 is equivalent to X - 500
X#- 9100 is equivalent to X500

Addresses **0** and **N** can not be used for the reference of the variable. It is impossible to designate a value exceeding the maximum command value specified for each address.

Example: **M#30** **when#30=120**

17.1.3 Display and setting of variable value

A variable value can be displayed on the screen or can be set for the variable by using the **MDI** keys. See the operator’s panel.

17.2 Kind of variable

Variables are divided into common variables and system variables according to the variable numbers.

17.2.1 Common variable #100 to #131 and #500 to #531

Application of the common variables is not determined in the system, but can be freely determined by the user.

Common variables **#100** to **131** are set to **“0”** immediately after the power is turned on.

Common variables **#500** to **#531** are retained when the power is cut off.

17.2.2 System variables

Application of the system variables are fixed in the system.

(1) *Interface input signals* ***#1000*** *to* ***1015****,* ***#1032***

	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	1^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
DI	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

All input signals can be read out by reading system variable **#1032.**

$$\#1032=\sum_{i=0}^{15}\#(100+i)x2^{i}$$

Values can be assigned to the system variables **#1000** to **1032**.

DGN 110: UI0 - UI7

DGN 111: UI8 - UI15

System variables **#1000** to **1032** can be displayed on the screen by diagnostic function.

(2) *Interface output signals* ***#1100*** *to* ***#1115*** *and* ***#1132***

	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	1^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
DO	U015	U014	U013	U012	U011	U010	U09	U08	U07	U06	U05	U04	U03	U02	U01	U00
	#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

All output signals can be sent by assigning a value to the system variable **#1132.**

$$\#1132 = \sum_{i=0}^{15} \#(1100+i)x2^{i}$$

When a value different from **“0”** or **“1”** is satisfied to the system variables **#1100** to **#1115**, the value is regarded as **“1”**. It is possible to read the values of system variables **#1100** to **#1132**. System variables **#1100** to **#1115** can be displayed by a diagnostic function.

DGN126 : U00 - U07
DGN127 : U08 - U015

(3) *Tool offset amount can be determined and specified by the system variables **#2001** to **#2932**.*

	Tool ofset number	Tool offset amount	Wear offset number	Geometry offset amount
X	1 to 32	#2001 to #2032	#2001 to #2032	#2701 to #2732
Z	1 to 32	#2101 to #2132	#2101 to #2132	#2801 to #2832
R	1 to 32	#2201 to #2232	#2201 to #2232	#2901 to #2932
T	1 to 32	#2301 to #2332	#2301 to #2332	#2301 to #2332

Example: **#30=#2005**

Assign the **X** axis tool offset amount of offset No.5 to variable **#30**. When the offset amount is 1.5 mm, **#30=1.5**

(4) *Position information **#5001** to **#5122***

The position information can be known by reading system variables **#5001** to **5122**. The unit is 0.001 mm in the metric programming system, and 0.0001 inch in the inch programming system.

System variable	Position information	Read during movement	Tool nose radius compensation, tool offset
#5001 #5002	X-axis block end coordinate (ABSIO) Z-axis present coordinate	Possible	Not considered. Tool nose position (programmed command position)
#5041 #5042	X-axis present coordinate (ABSOT) Z-axis present coordinate	Impossible	Considered. Tool standard point (same as ABSOLUTE display of POS page)
#5061 #5062	X-axis skip signal position (ABSKP) Z-axis skip signal position	Possible	Considered. Tool standard point.
#5081 #5082	X-axis tool offset or wear offset amount Z-axis tool offset or wear offset amount	Impossible	
#5121 #5122	X-axis geometry offset amount Z-axis geometry offset amount	Possible	

Values can not be assigned to these variables **#5001** to **#5122.**

17.3 Macro instructions (G65)

General format: **G65 Hm P#i Q#j R#k;**

where:

m: Macro functions are indicated by 01 to 99

#i: Variable name where the calculation result is entered

#j: Variable name 1 calculated. May be constant

#k: Variable name 2 calculated. May be constant

Example:

m=02

P#100 Q#101 R#102.................................	#100=#101+#102
P#100 Q#101 R15....................................	#100=#101+15
P#100 Q-100 R#102..................................	#100= -100+#102
P#100 Q120 R50.......................................	#100=120+50
P#100 Q-#101 R#102................................	#100= -#101+#102

Decimal point can not be used for the variable values.

An angle is designated in degree in 1/1000 degree increments, e.g. 1°=1000.

List of Macro Instructions

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	Multiplication	#i = #j x #k
G65	H05	Division	#i = #j ÷ #k
G65	H11	Logical sum	#i = #j . OR . #k
G65	H12	Logical multiplication	#i = #j . AND . #k
G65	H13	Exclusive OR	#i = #j . XOR . #k
G65	H21	Square root	$\#i = \sqrt{\#j}$
G65	H22	Absolute value	#i = I #j I
G65	H23	Remainder	#i = #j - trunc (#j / #k)x #k trunc: Discard fractions less than 1
G65	H24	Conversion from BCD to binary	#i = BIN (#j)
G65	H25	Conversion from binary to BCD	#i = BCD (#j)
G65	H26	Combined multiplication/division	#i = (#i x #j)÷ (#k)
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 = ATAN . (#J/#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 occurrence

17.3.1 Variable arithmetic command

(1) *Definition and substitution of variable* ***#i=#j***

G65 H01 P#i Q#j;

Example: G65 H01 #101 Q1005; (#101=1005)
G65 H01 P#101 Q#110; (#101=#110)
G65 H01 P#101 Q-#112; (#101=-#112)

(2) *Addition* ***#i=#j+#k***

G65 H02 P#i Q#j R#k;

Example: G65 H02 P#101Q#102 R#103 (#101=#102+#103)

(3) Subtraction: **#i=#j - #k**

G65 H03 P#iQ#jQ#k;

Example: G65 H03 P#101 Q#102 R#103 (#101=#102 - #103)

(4) *Multiplication* ***#i=#j x #k***

G65 H04 P#i Q#j R#k

Example: G65 H04 P#101 Q#102 R#103 (#101=#102 x #103)

(5) *Division* ***#i=#j÷#k***

G65 H05 P#i Q#j R#k

Example: G65 H05 P#101 Q#102 R#103 (#101=#102÷#103)

(6) *Logical sum* ***#i=#j . OR . #k***

G65 H11 P#i Q#j R#k;

Example: G65 H11 P#101 Q#102 R#103 (#101=#102 . OR . #103)

(7) *Logical multiplication* ***#i=#j . AND . #k***

G65 H12 P#i Q#j R#k;

Example: G65 H12 P#101 Q#102 R#103 (#101=#102 . AND . #103)

(8) *Exclusive* ***OR #i=#j . XOR . #k***

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

Example: G65 H13 P#101 Q#102 R#103 (#101=#102 . XOR . #103)

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

G65 H21 P#i Q#j;

Example: G65 H21 P#101 Q#102 $(\#101=\sqrt{\#102})$

(10) *Absolute value* ***#i=|#j|***

G65 H22 P#101 Q#102;

Example: G65 H22 P#101 Q#102 (101=|#102|)

(11) *Remainder* ***#i=#j - trunc (#j/#k) x #k***

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

Example: G65 P#101 Q#102 R #103 (#101=#102 - trunc (#102 / #103)x#103

(12) *Conversion from **BCD** to binary* ***#i=BIN (#j)***

G65 H24 P#iQ#j;

Example: G65 H24 P#101 Q#102 (#101=BIN(#102);

(13) *Conversion from binary to **BCD*** ***#i=BCD (#j)***

G65 H25 P#i Q#j;

Example: G65 H25 P#101 Q#102 (#101=BCD (#102)

(14) *Combined* ***multiplication/division***

G65 H26 P#i Q#j R#k;

Example: G65 H26 P#101 Q#102 R#103 (#101=(#101x#102)÷#103

(15) *Combined square **root 1*** $\#i = \sqrt{\#j^2 + \#k^2}$

G65 H27 P#i Q#j R#k;

Example: G65 H27 P#101 Q#102 R#103 $\mathbf{\#101 = \sqrt{\#102^2 + \#103^2}}$

(16) *Combined square **root 2*** $\#i = \sqrt{\#j^2 - \#k^2}$

G65 H28 P#i Q#j R#k;

Example: G65 H28 P#101 Q#102 R#103 $\#101 = \sqrt{\#102^2 - \#103^2}$

(17) *Sine* ***#i=#j x SIN(#k)***

G65 H31 P#i Q#j R#k;

Example: G65 H31 P#101 Q#102 R#103 (#101= #102 x SIN(#103))

(18) *Cosine* ***#i=#j x COS (#k)***

G65 H32 P#i Q#j R#k;

Example: G65 H32 P#101 Q#102 R#103 (#101=#102 X COS(#103))

(19) *Tangent* ***#i=#j x TAN(#k)***

G65 H33 P#i Q#j R#k;

Example: G65 H33 P#101 Q#102 R#103 (#101=#102 x TAN(#103))

(20) *Arctangent* ***#i=#j x ARCTAN(#k)***

G65 H34 P#i Q#j R#K;

Example: G65 H34 P#101 Q#102 R#103 (#101=#102 x ARCTAN(#103))

When **Q** or **R** necessary for operation is not designated, the value is regarded as **“0”**.

17.3.2 Control command

(1) *Unconditional divergence*

G65 H80 Pn; **n -** sequence number

Example: G65 H80 P120; (deverge to N120)

(2) *Conditional divergence **1***

G65 H81 Pn Q#j R#k; **n-** sequence number

Example: G65 H81 P1000 Q#101 r#102;
#101=#102, GOTO n1000
#101= #102, GOTO next

(3) *Conditional divergence* ***2***

G65 H82 Pn Q#j R#k; **n -** sequence number

Example: G65 H83 P1000 Q#101 R#102;
#101= #102, GOTO N1000
#101= #102, GOTO next

(4) *Conditional divergence* ***3***

G65 H83 Pn Q#j R#k; **n-** sequence number

Example: G56 H83 P1000 Q#101 R#102
H101>#102, GOTO N1000
H101J#102, GOTO next

(5) *Conditional divergence* ***4***

G65 H84 Pn Q#j R#k; **n -** sequence number

Example: G65 H84 P1000 Q#101 R#102
H101<#102, GOTO N1000
H101i#102, GOTO next

(6) *Conditional divergence* ***5***

G65 H85 Pn Q#j R#k; **n -** sequence number

Example: G65H85 P1000 Q#101 R#102
#101i#102, GOTO N1000
#101<#102, GOTO next

(7) *Conditional divergence* ***6***

G65 H86 Pn Q#j R#k; **n -** sequence number

Example: G65 H86 P1000 Q#101 R#102
#101i#102, goto N1000
#101<#102, goto next

(8) ***P/S*** *alarm occurrence*

G65 H99 Pi;   i - alarm No.500

Example: G65 H99 P15

P/S alarm 515 occurrence

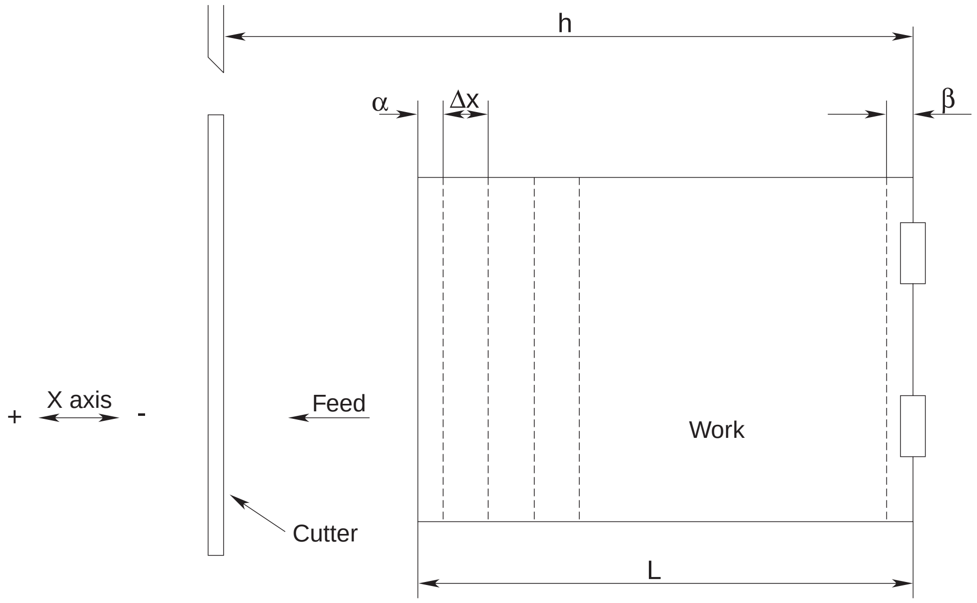
17.4 Cautions on Custom Macro

In **MDI** mode, the macro instructions can be commanded, but address data other than **G65** is not displayed. **H, P, Q** and **R** in the macro instructions must always be designated after **G65**.

In SINGLE BLOCK mode, operation does not stop in case of execution of a macro instruction. It is possible to effect SINGLE BLOCK STOP by setting parameter No. 011SBKM.

17.5 Aplication of Custom Macro

17.5.1 Shearing machine



#500: workpiece width (**L**)

#501: first stock removal (**α**)

#502: cutting width (**Δx**)

#503: workpiece gripping allowance (**β**)

#504: distance from reference point to tool (**h**)

Custom macro:

```
O9110;
G65 H03 P#100 Q#504 R#501;
N10 G65 H03 P#101 Q#504 R#100;
G00 X#100;
M20;     (cutting command)
G65 H03 P#100 Q#100 R#502;
G65 H85 P-10 Q#100 R#503;
M99;
```

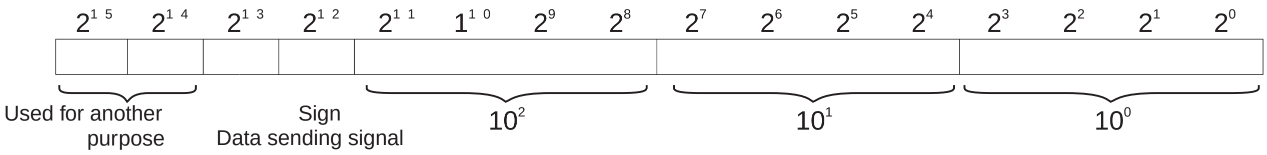
Main program:

```
O0009
G92 X0;
M98 P9110;
XO;
M02;
```

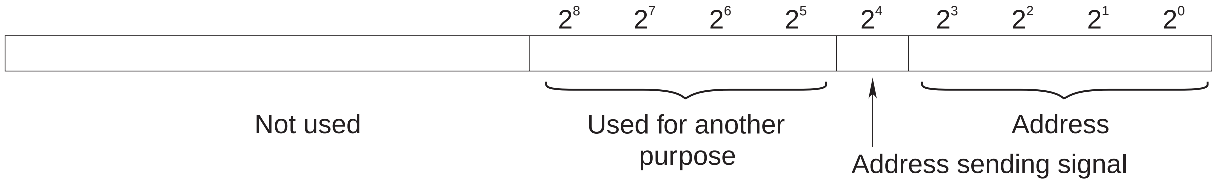
17.5.2 Interface signal

Read the three-digit signed BCD value by address switching to **#100**.

DI configuration



D0 configuration



Custom macro:

```
O9110
G65 H12 P#1132 Q#1132 R480;
G65 H11 P#1132 Q#1132 R23;
N10 G65 H81 P10 Q#1013 R0;
G65 H12 P#100 Q#1032 R4095;
G65 H24 P#100 Q#100;
G65 H81 P20 Q#1012 R0;
G65 H01 P#100 Q-#100;
N20 G65 H12 P#1132 Q#1132 R495;
M99;
```

OPERATION

Contents

1. GENERAL

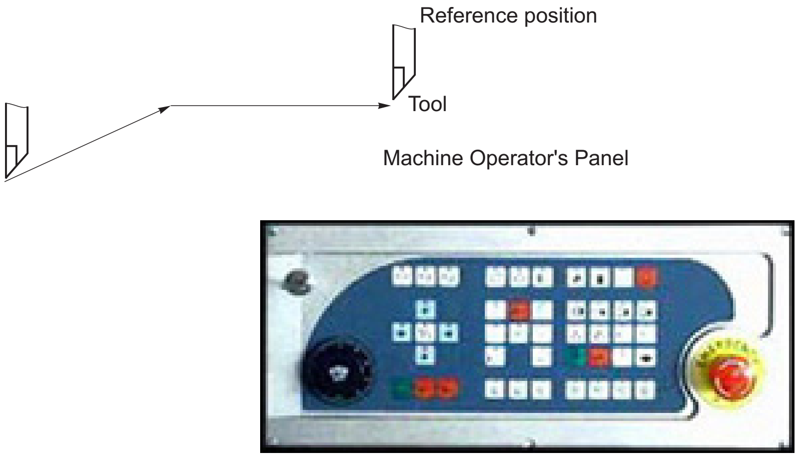
1.1 MANUAL OPERATION

◆ MANUAL REFERENCE POSITION RETURN (ZRN MODE)

The **CNC** machine tool has a position called reference position. Here either the tool is replaced or the coordinate system origin is set. Ordinarily, after the power is turned on, the tool is moved to the reference position.

The manual reference position return is an operation of moving the tool to the reference position using the pushbuttons located on the operator's panel.

Manual Reference Position

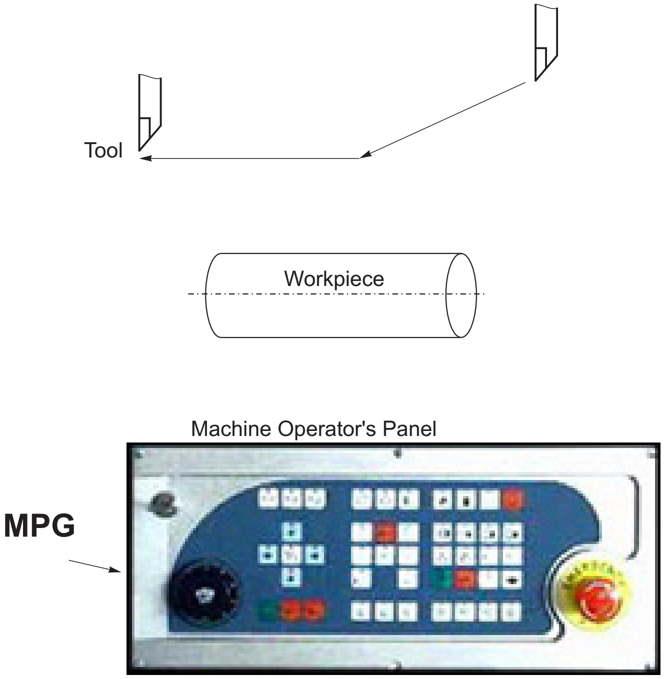


The tool can be moved to the reference position also with program commands. This operation is called automatic reference position return.

◆ MANUAL TOOL MOVEMENT

The tool can be moved along each axis using the pushbuttons located on the operator’s panel.

The Tool Movement by Manual Operation



<u>The tool can be moved in one of the following ways:</u>

(1) ***Jog feed***

The tool moves continuously while the pushbutton remains pressed.

(2) ***Incremental feed***

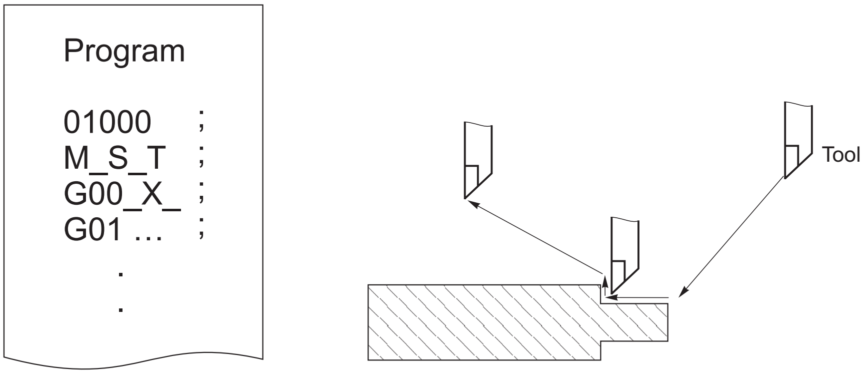
The tool is moved to a predetermined distance each time the button is pressed.

(3) ***Manual handle***

The tool is moved to a distance corresponding to the degree of manual handle rotation.

1.2 TOOL MOVEMENT BY PROGRAMMING - AUTOMATIC OPERATION

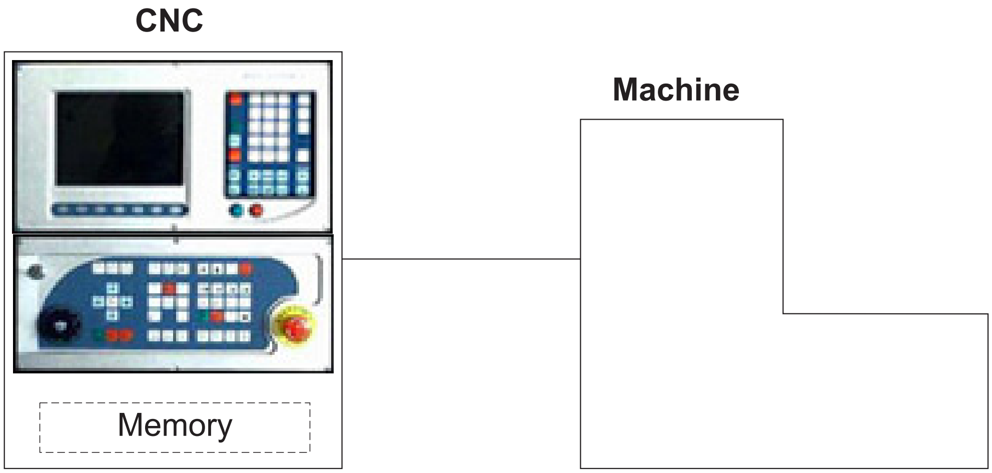
Automatic operation means operating the machine according to the created program. It includes the program in memory, **DNC** or **MDI** operation.



◆ OPERATION ACCORDING PROGRAM IN MEMORY (MODE AUTO)

Once the program is loaded in the memory of **CNC**, the machine can be run according to the instructions in this program. This operation is called memory operation.

Memory Operation



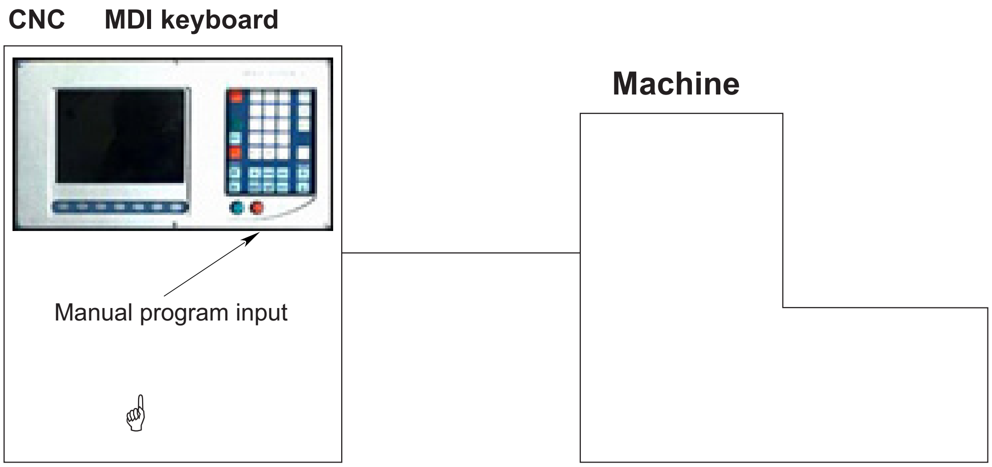
◆ DNC OPERATION (MODE AUTO/DNC)

In this mode of operation, the program is not hold in the **CNC** memory. It is read from the connected input/output device instead. This mode is useful when the program is too large to fit in the **CNC** memory.

◆ MDI OPERATION (MODE MDI)

After a command group is entered from the **MDI** keyboard, the machine can be run according to these commands. This mode is called **MDI** operation.

MDI Operation



1.3 AUTOMATIC OPERATION

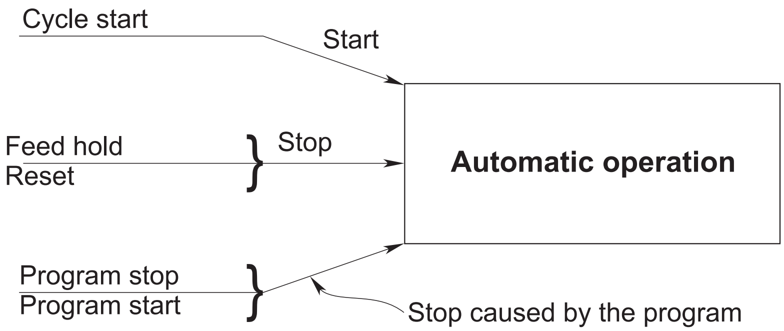
- **PROGRAM SELECTION**

In **EDIT** mode select the program used for the corresponding workpiece. Ordinarily, one program is used for a workpiece. If two or more programs are stored in memory, select the program needed by searching the corresponding program number.

- **START AND STOP**

Pressing the cycle start pushbutton causes automatic operation start. By pressing the stop button automatic operation stops. If the program stops or termination command is specified, operation is automatically stopped after reaching the command. When one machining process is completed, automatic operation is stopped.

Start and Stop for Automatic Operation



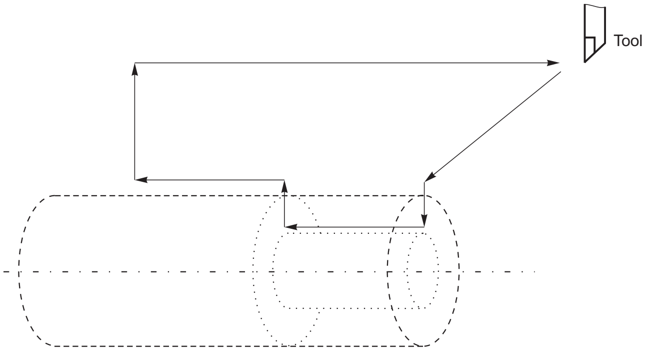
1.4 TESTING A PROGRAM

Before machining is started, an automatic running check can be executed. It checks whether the created program can operate the machine as desired. This check can be accomplished by running the machine without a workpiece or by viewing the coordinate changes without running the machine itself.

1.4.1 Check by running the machine

- **DRY RUN**

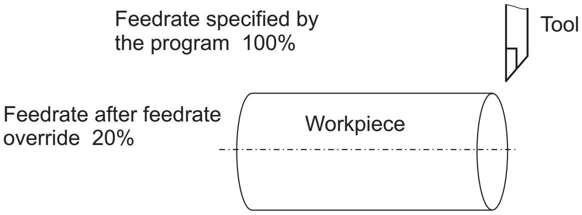
Remove the workpiece and check only the movement of the tool. Select the tool movement rate using the corresponding pushbutton on the operator’s panel.



- **FEED OVERRIDE**

Check the program by changing the feedrate specified by the program.

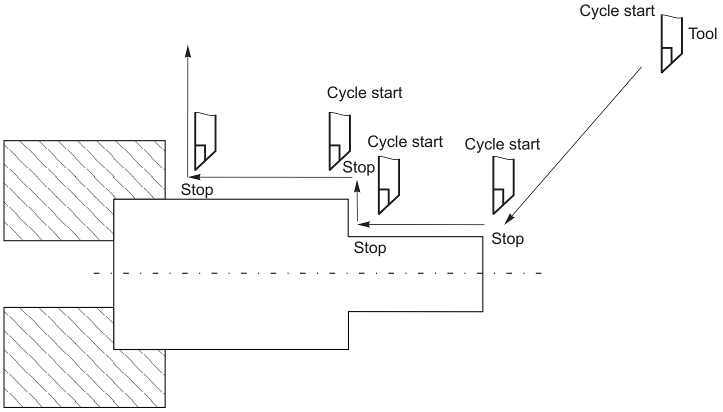
Feedrate Override


Feedrate specified by the program 100%
Tool
Feedrate after feedrate override 20%
Workpiece

◆ SINGLE BLOCK EXECUTION

When the cycle start pushbutton is pressed, the tool executes only one operation and stops afterwards. By pressing the cycle start again, the tool executes one more operation and then stops. The whole program can be checked in this manner.

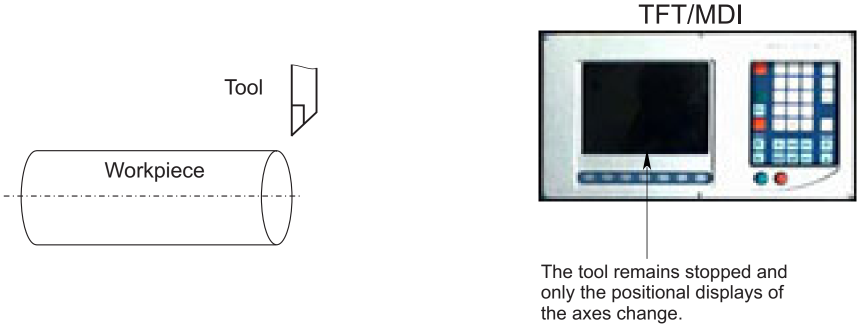
Single Block



1.4.2 Position display change without running the machine

◆ MACHINE LOCK

Machine Lock


TFT/MDI
Tool
Workpiece
The tool remains stopped and only the positional displays of the axes change.

CAUTION:

Once this function is used, the coordinate system is shifted. For this reason, before starting the machining itself, take measures for setting a correct coordinate system.

◆ AUXILIARY FUNCTION LOCK

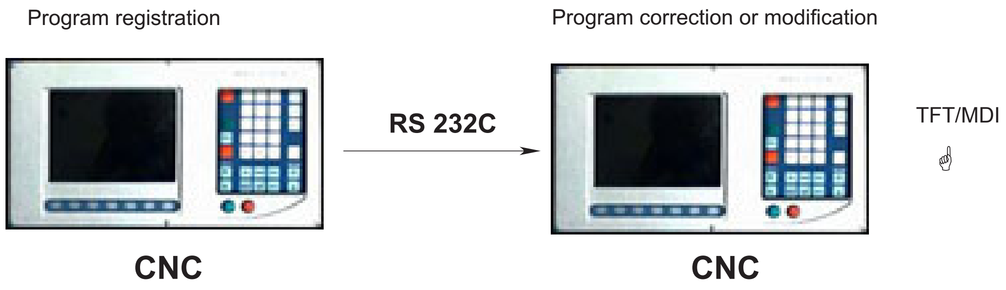
When automatic running is set in auxiliary function lock mode and machine lock, all auxiliary functions (spindle rotation, tool replacement, coolant on/off, etc.) are disabled. This function is available on the operator’s panel and the realization depends on the machine builder.

1.5 EDITING A PART OF THE PROGRAM

Once the created program is loaded into the memory, it can be corrected or modified using the TFT/MDI panel.

This operation can be executed using the “edit” function of the program.

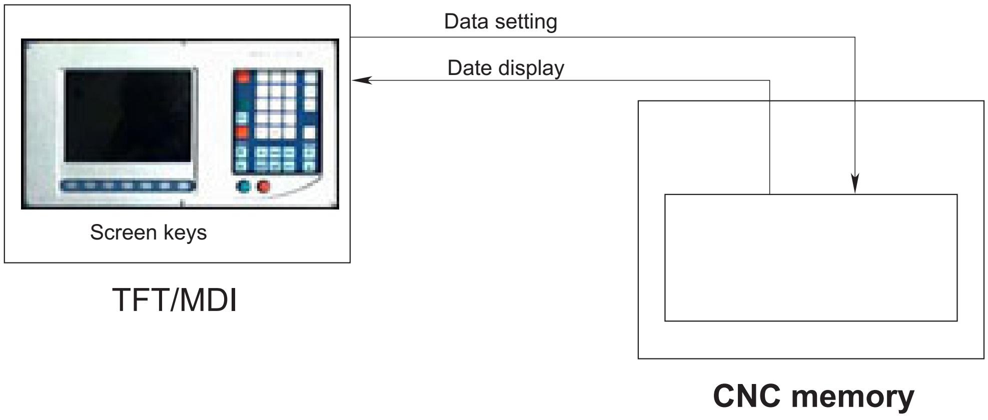
Part Program Editing



1.6 SETTING AND DISPLAYING DATA

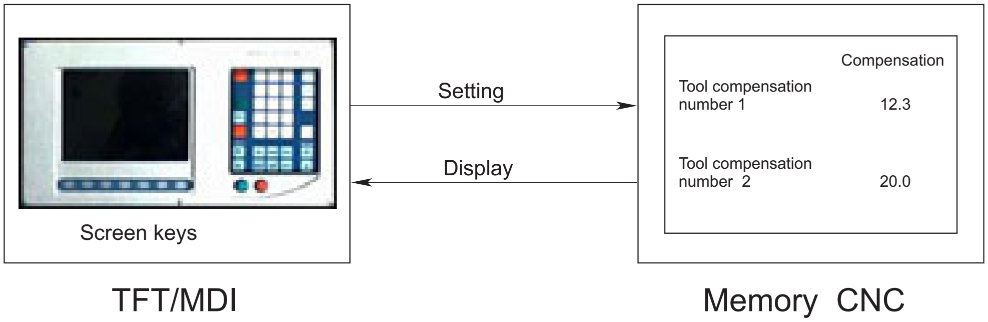
The operator can display or change the values stored in the CNC internal memory using the TFT/MDI panel.

Displaying and Setting Data



◆ OFFSET VALUE

Displaying and Setting Offset Value



	Compensation
Tool compensation number 1	12.3
Tool compensation number 2	20.0

The tool has its corresponding dimensions (length, diameter). When a workpiece is machined, the tool movement value depends on its dimensions.

If tool dimension data is set in **CNC** memory beforehand, the tool path is automatically corrected in such a way that the workpiece is machined according the data specified by the program. The tool dimension data is called offset value.

◆ SETTING AND DISPLAYING DATA BY THE OPERATOR

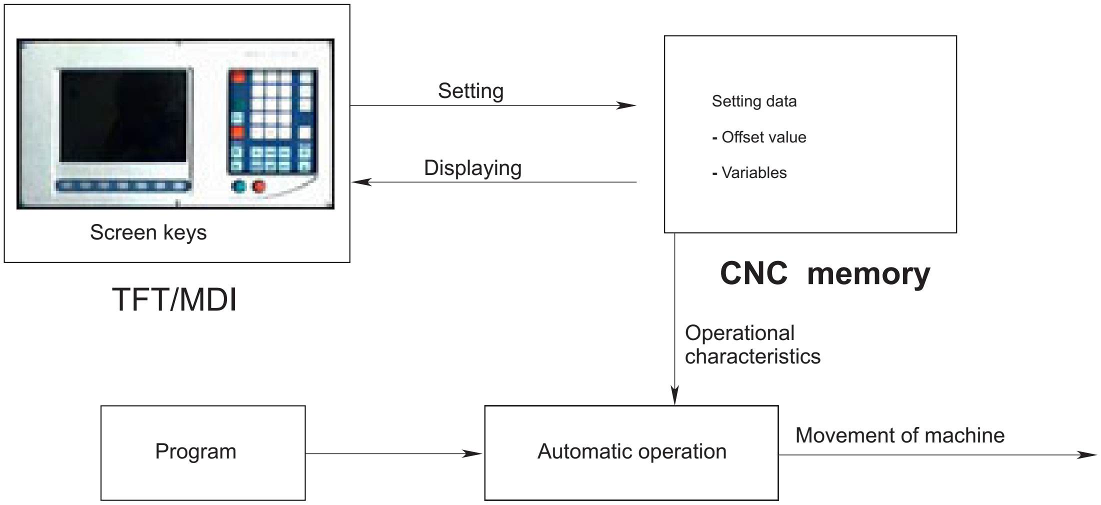
Apart from parameters, there is a data that is set by the operator during operation. This data change different machine characteristics. For example, the following data can be set:

- Offset values

- Variables

The above data is called ***setting data (SETTINGS)***.

Displaying and Setting Operator's Data



◆ SETTING AND DISPLAYING PARAMETERS

The **CNC** functions have versatility in order to be used for different kinds of machines.

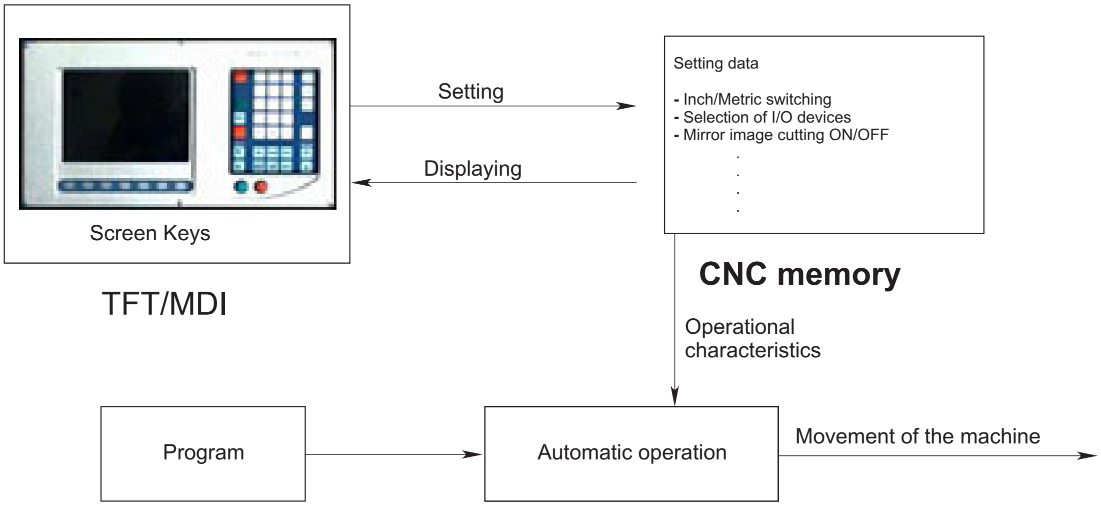
For example, **CNC** can specify the following:

- Rapid traverse rate along each axis
- Input metric/inch system
- Cutting feedrate
- Back lash compensation

Data specifying these characteristics are called ***parameters***.

Parameters differ depending on the machine.

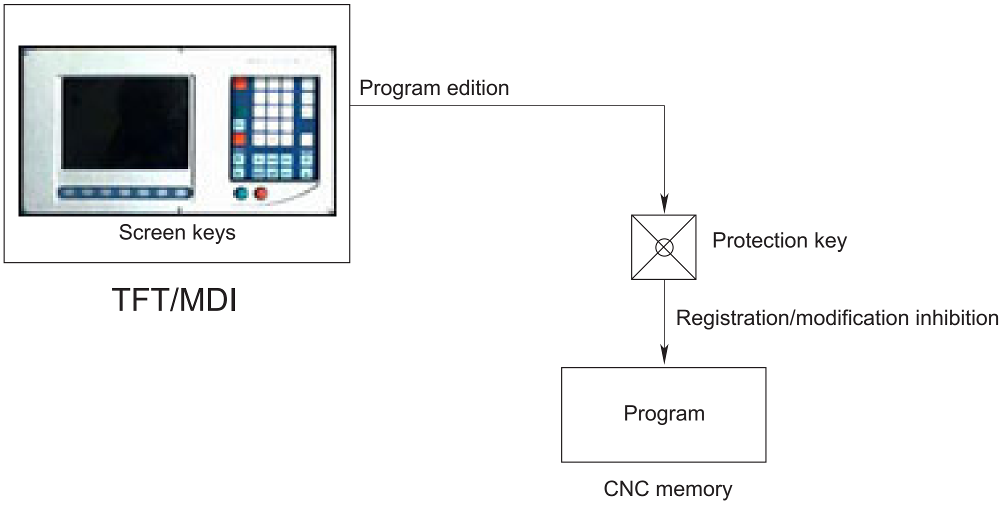
Displaying and Setting Operator's Setting Data



◆ DATA PROTECTION KEY (PROTECT KEY)

A key called data protection key is available on the operator’s panel. It is used to prevent part of the programs from erroneous loading, modification or deletion.

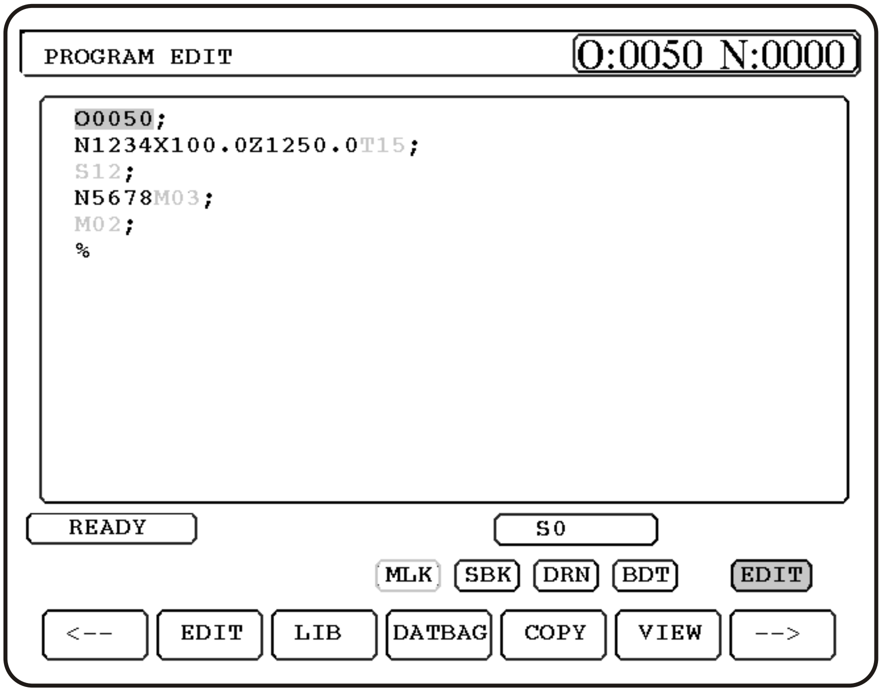
Data Protection Key



1.7 DISPLAY

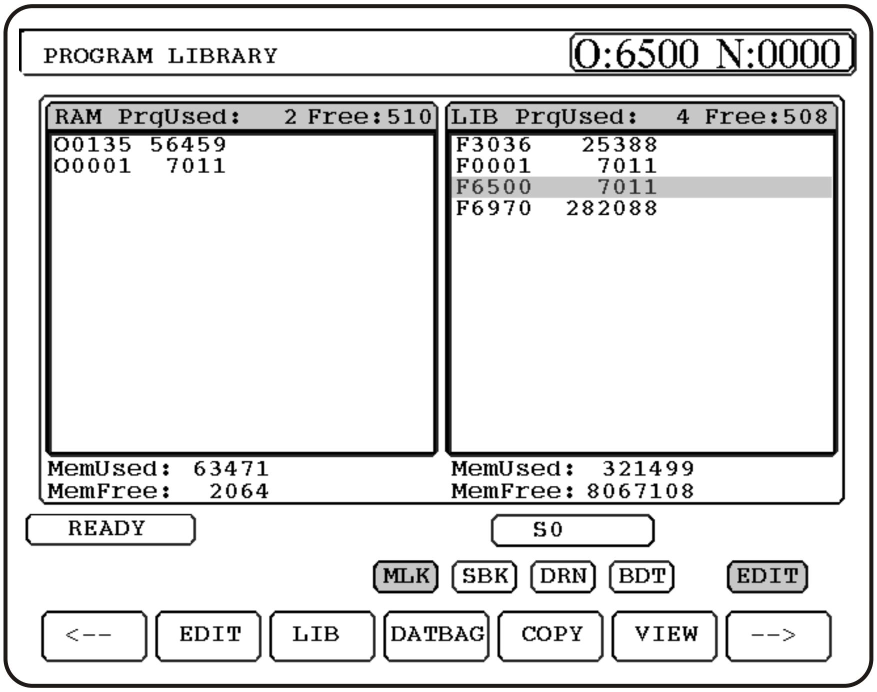
1.7.1 Program display

The contents of the current program can be displayed on the screen. In addition, the program list can be displayed.



PROGRAM EDIT O:0050 N:0000

```
O0050;
N1234X100.0Z1250.0T15;
S12;
N5678M03;
M02;
%
```



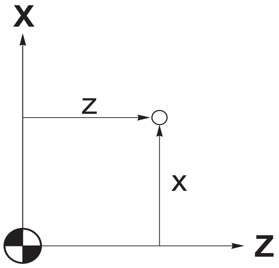
PROGRAM LIBRARY O:6500 N:0000

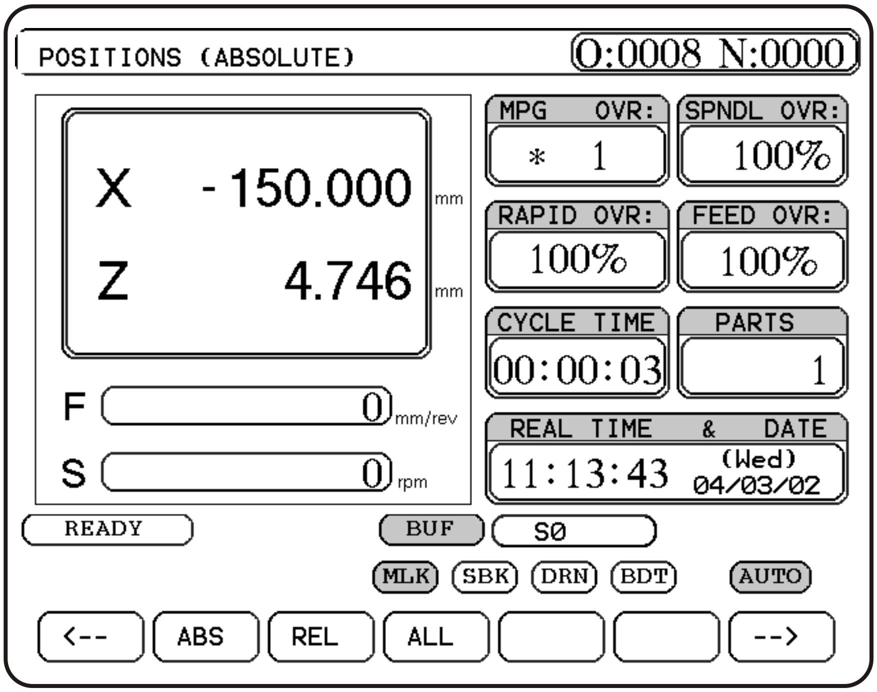
```
RAM PrqUsed:   2 Free:510    LIB PrqUsed:   4 Free:508
O0135 56459                  F3036    25388
O0001   7011                 F0001     7011
                             F6500     7011
                             F6970   282088

MemUsed:  63471              MemUsed:  321499
MemFree:   2064              MemFree: 8067108
```

1.7.2 Current position display

The current position of the tool is displayed with coordinate values. The distance from the current position to the target position can also be displayed.

Workpiece Coordinate System





POSITIONS (ABSOLUTE) O:0008 N:0000

X -150.000 mm
Z 4.746 mm

F 0 mm/rev
S 0 rpm

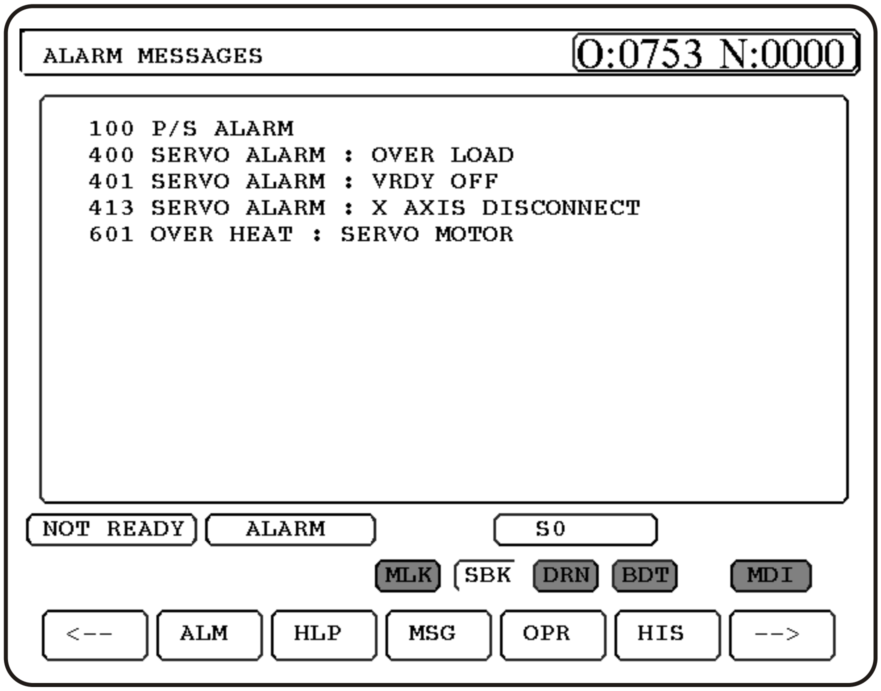
MPG OVR: * 1
SPNDL OVR: 100%
RAPID OVR: 100%
FEED OVR: 100%
CYCLE TIME 00:00:03
PARTS 1
REAL TIME & DATE 11:13:43 (Wed) 04/03/02

READY BUF S0

In addition, number of parts, cycle time and real time is displayed.

1.7.3 Alarm display

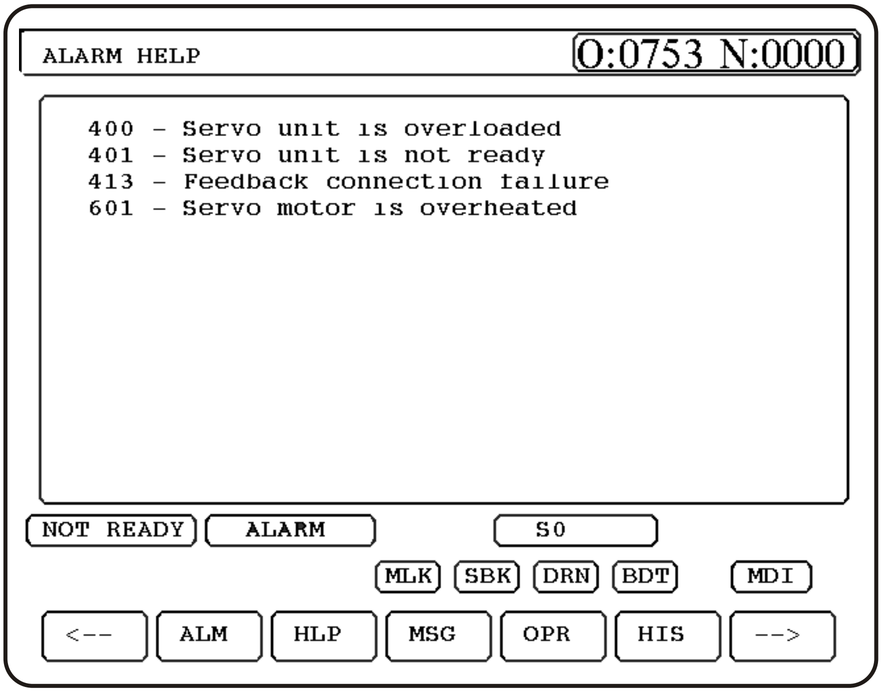
When a trouble occurs during operation, error code and the alarm message are displayed on the TFT screen. See the appendix for the list of error codes and their meanings. It is included a short description of each error code.



ALARM MESSAGES O:0753 N:0000

```
100 P/S ALARM
400 SERVO ALARM : OVER LOAD
401 SERVO ALARM : VRDY OFF
413 SERVO ALARM : X AXIS DISCONNECT
601 OVER HEAT : SERVO MOTOR
```

NOT READY ALARM S0



ALARM HELP O:0753 N:0000

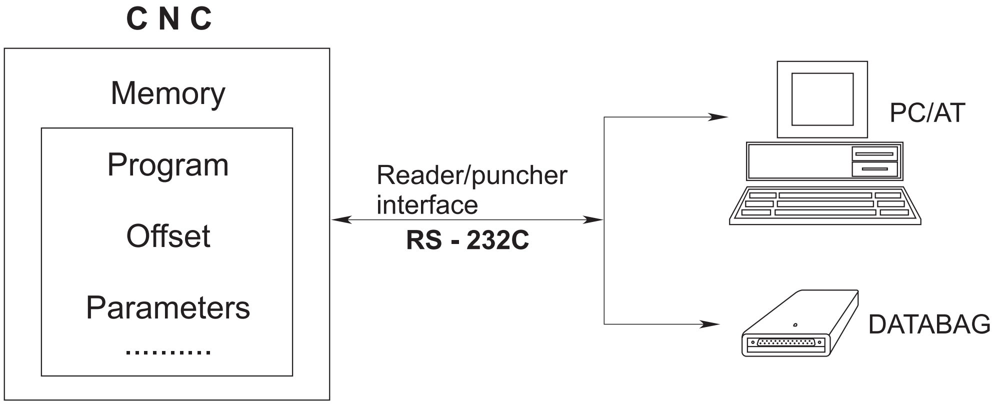
```
400 - Servo unit is overloaded
401 - Servo unit is not ready
413 - Feedback connection failure
601 - Servo motor is overheated
```

NOT READY ALARM S0

Press the buttons PgUp and PgDn to switch the above screens.

1.8 DATA OUTPUT

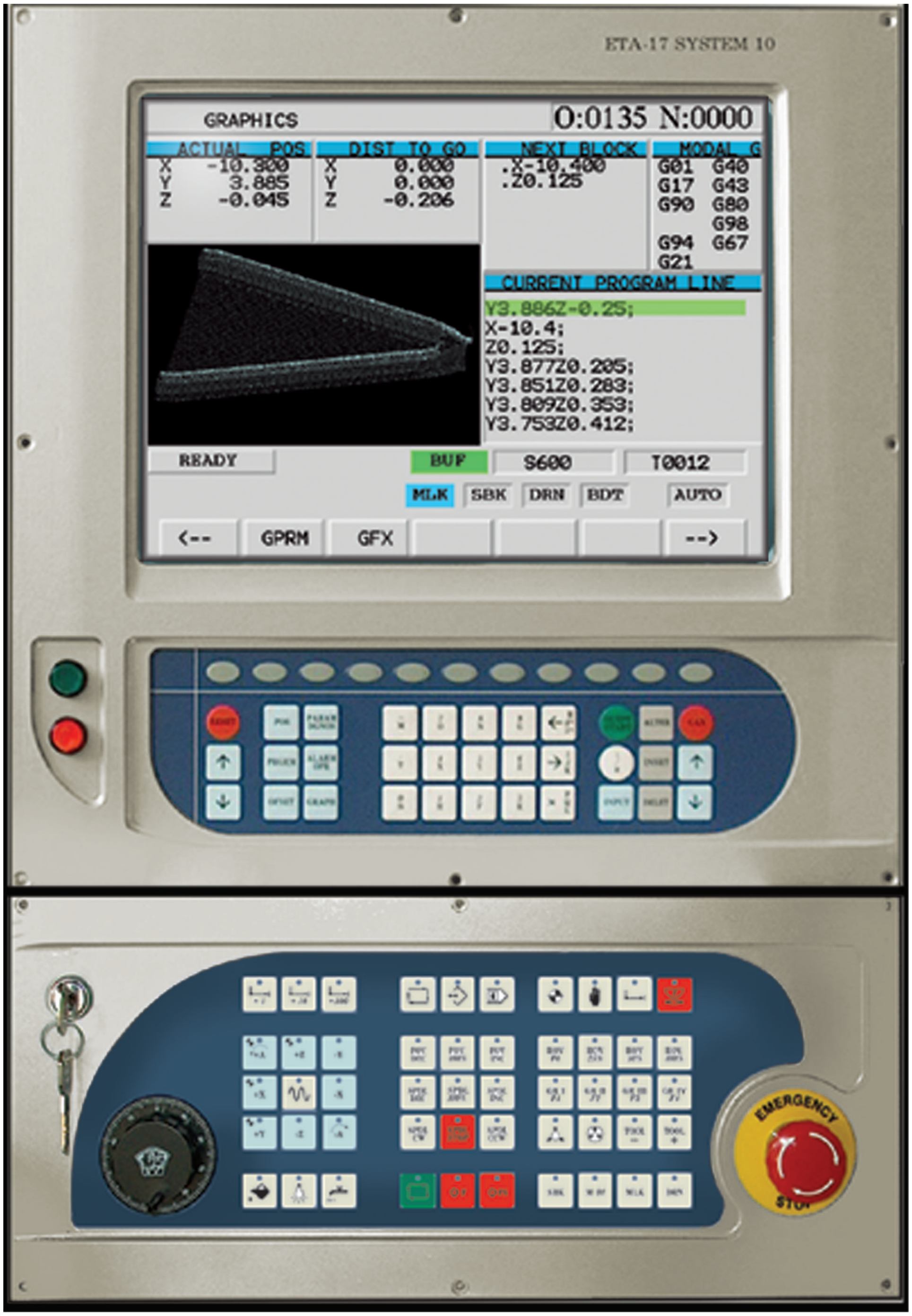
Programs, offset values, parameters, etc. input in **CNC** memory can be output to an outer device via **RS232C**, including personal computers and energy independent portable device for saving the data (**DataBag model 12M**).



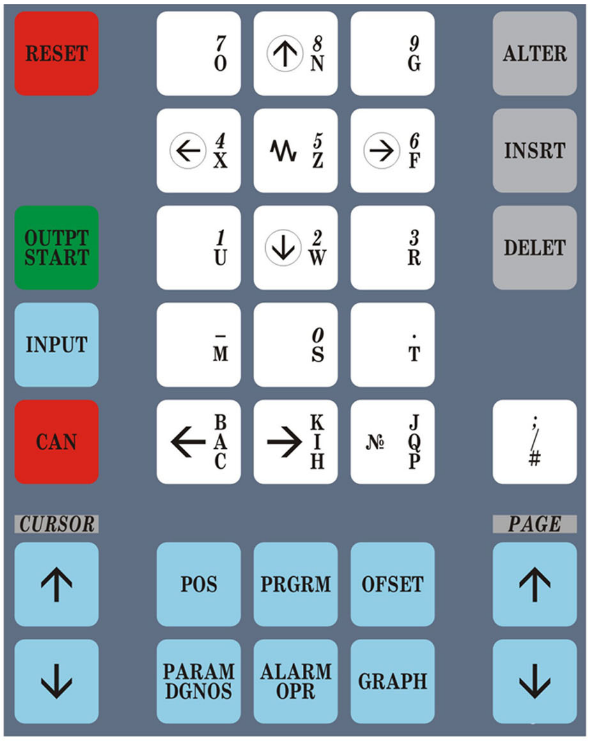
2. PERIPHERAL DEVICES

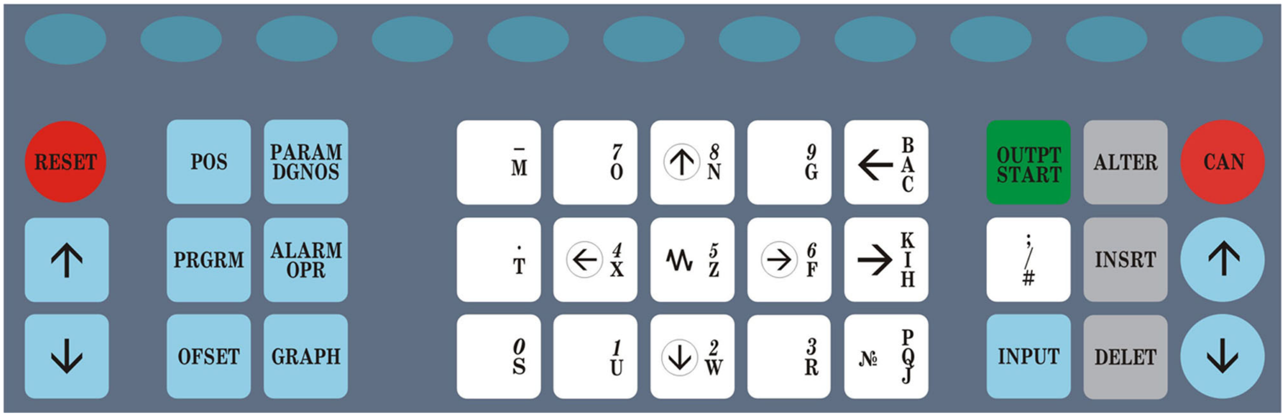
The peripheral devices available include TFT/MDI panel connected to **CNC**, machine operator’s panel and external input/output devices such as **PC** and energy independent data storage device (**DataBag**).

2.1 TFT/MDI PANEL



MDI keyboards





RESET key - [RESET]

Press this key to reset **CNC**, to cancel an alarm, etc.

***START key* - [OUTPUT/START]**

This key is used to start **MDI** operation or automatic mode, depending on the machine. Refer to the manual provided by the machine builder. This key is also used to output data to the input/output unit.

Soft keys

The soft keys have various functions depending on the applications. Their functions are displayed at the bottom of the **CRT** screen.

***Address and numeric keys* - [8N] [0S]**

These keys are used to input alphabetic, numeric and other characters.

***INPUT key* - [INPUT]**

When an address or numeric key is pressed, the data is input in a buffer, which is displayed on the **CRT** screen. To copy the data from the input key buffer to the corresponding register, press the **[INPUT]** key.

This key is also used to input data from an input/output unit.

***Cancel key* - [CAN]**

This key is used to delete the last character of the input key buffer.

***Program edit keys* - [ALTER], [INSRT], [DELET]**

These keys are used when editing a program:

[ALTER] **-** ***Alteration***
[INSRT] **-** ***Insertion***
[DELET] **-** ***Deletion***

***Functional keys* - [POS], [PRGRM] ...**

These keys are used to switch the different function screens. For more details on the function keys refer to the next chapter.

Cursor move keys

There are two different cursor move keys available:

- This key is used to move the cursor in a forward and downward direction.

- This key is used to move the cursor in a reverse and upward direction.

Page change keys

There are two page change keys available:

- This key is used to changeover the page on the **CRT** screen in the forward direction.

- This key is used to changeover the page on the **CRT** screen in the reverse direction.

2.2 FUNCTIONAL AND SOFT KEYS

2.2.1 General screen operations

1. Press a functional key on the **CRT/MDI** panel. A menu with soft keys appears depending on the selected function.

2. Press one of the soft keys. A screen corresponding to the menu appears. If the desired command is not in the current menu, press the key for menu continuation.

In some cases additional menus can be displayed.

3. The general screen display procedure is explained above. However, the actual display procedure varies from one screen to another. For more details, see the description of individual operations.

2.2.2 Functional keys

The functional keys are used to select the type of the screen and the display mode. The following functional keys are provided on the TFT/MDI panel:

[POS]	**-** Press this key to display the position screen.
[PRGRM]	**-** Press this key to display the program screen.
[MENU] **[OFSET]**	**-** Press this key to display the offset screen.
[DGNOS] **[PARAM]**	**-** Press this key to display the parameter/diagnosis screen.
[OPR] **[ALARM]**	**-** Press this key to display the alarm screen.
[AUX] **[GRAPH]**	**-** Press this key to display the graphic functions of the system.

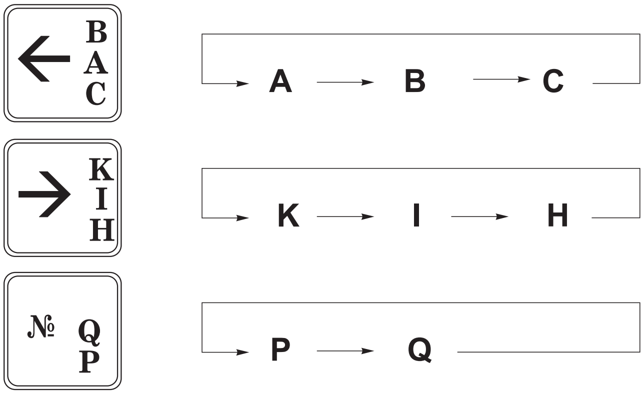
2.2.3 Key input and input buffer

- **FOR STANDARD KEY**

When an address or numerical key is pressed, the character corresponding to that key is input into the key input buffer. The contents of the key input buffer is displayed at the bottom of theTFT screen.

On the standard key panel, one and the same key is used to enter address or numeric value. That depends on the context.

Data of one word (address + numeric value) can be entered into the key input buffer at once. The following data input keys are used to input the addresses. Each time the key is pressed, the input address changes as shown below:



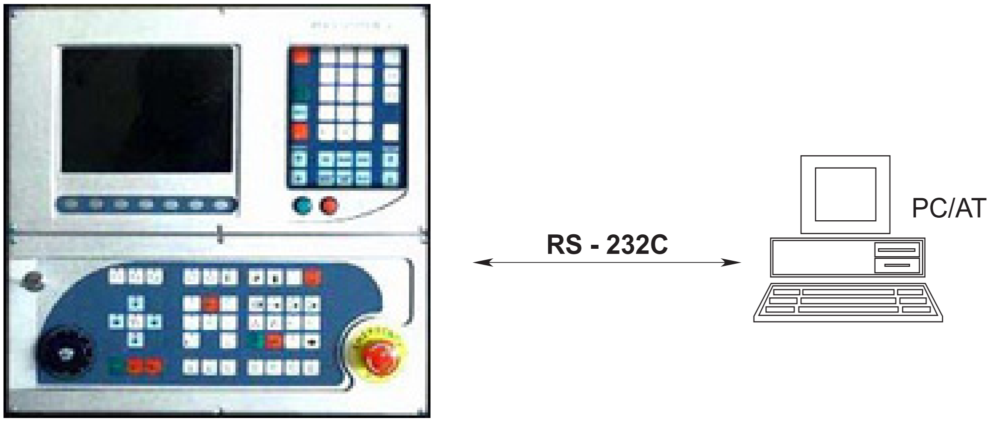
Pressing the **[CAN]** key deletes all the data stored in the key input buffer. When the buffer is not empty, each pressing of **[DELET]** key deletes only the last input symbol.

2.3 EXTERNAL I/O DEVICES

CNC systems 10 **T of ETA-17** provide asynchronous serial interface **RS-232C** for establishing a connection with external input/output devices. Special protocol for ensuring save data transfer is used.

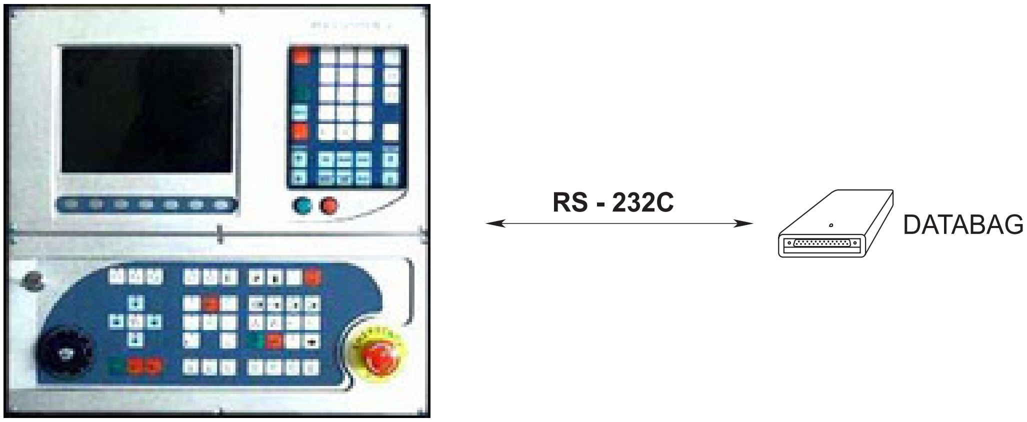
The following devices can be connected:

◆ PERSONAL COMPUTER IBM AT OR COMPATIBLE



The software package **NC Tools** provides the corresponding protocol.

◆ DATA STORAGE DEVICE “DATABAG”



DataBag can also be used for data transfer from one personal computer to another.

The following data types can be input/output to or from **CNC**:

- **PROGRAMS**
- **OFFSETS**
- **PARAMETERS**
- **VARIABLES**
- **A WORK ZONE FOR PMC-X (DGN 300 - DGN 699)**

The communication protocol ensures a safe and free of errors connection. The transfer rate is set automatically by the devices and can be reduced using a parameter.

2.4 POWER ON/OFF

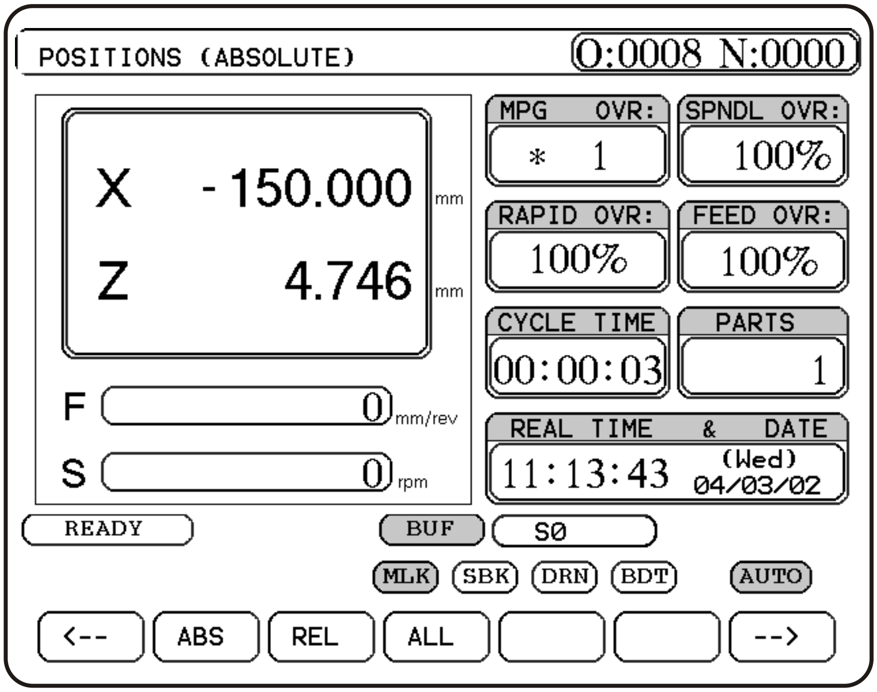
2.4.1 Turning the power on

PROCEDURE OF TURNING THE POWER ON

1. Check the appearance of the **CNC** machine. (For example, check whether the front and rear doors are closed.)

2. Turn the power on according to the manual issued by the machine builder.

3. After the power is turned on, check whether the position screen is displayed.



POSITIONS (ABSOLUTE) O:0008 N:0000

X -150.000 mm

Z 4.746 mm

F 0 mm/rev

S 0 rpm

MPG OVR: * 1

SPNDL OVR: 100%

RAPID OVR: 100%

FEED OVR: 100%

CYCLE TIME 00:00:03

PARTS 1

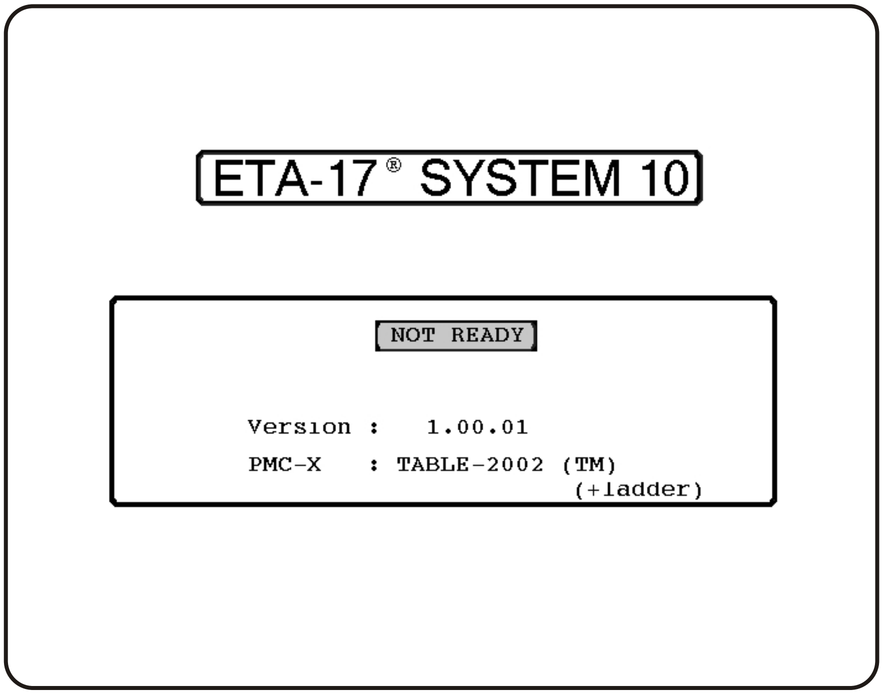
REAL TIME & DATE 11:13:43 (Wed) 04/03/02

4. Check whether the fan motor is rotating.

WARNING:

When pressing the **<POWER ON>** key, do not touch any other keys on the **CRT/MDI** panel, until the positional or the alarm screen is displayed. Some of the keys are used for maintenance or have special operation purpose. When they are pressed, unexpected operation may be caused.

2.4.2 Display of the software configuration



ETA-17® SYSTEM 10

NOT READY

```
Version :   1.00.01
PMC-X   : TABLE-2002 (TM)
                     (+ladder)
```

2.4.3 Power disconnection

1. Check whether the **LED** indicating the cycle start on the operator’s panel is off.

2. Check whether all movable parts of the **CNC** machine are stopped.

3. If an external input/output device is connected, disconnect it first.

4. Push the **<POWER OFF>** button.

Note:

For more details on turning the machine off refer to the manual provided by the machine builder.

3. MANUAL OPERATION

Manual operations are four kinds as follows:

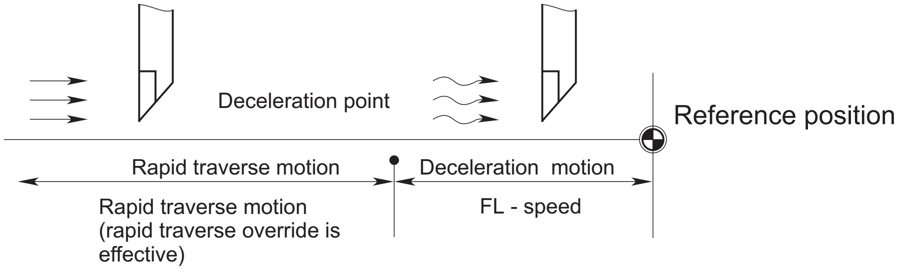
1. Manual reference position return.
2. Jog feed.
3. Incremental feed.
4. Manual handle feed.

3.1 MANUAL REFERENCE POSITION RETURN (ZRN MODE)

The tool is returned to the reference position as follows:

For each axis, the tool is moved in direction set by a parameter when the reference position return switch on the operator's panel is on. To the deceleration point the tool moves at rapid traverse rate and then moves to the reference position at the **FL** speed. The rapid traverse rate and the **FL** speed are set by parameters.

Four-step rapid traverse override can be set during the rapid traverse. When the tool returns to the reference position, the reference position return completion indicator lamp goes on.



PROCEDURE FOR MANUAL REFERENCE POSITION RETURN

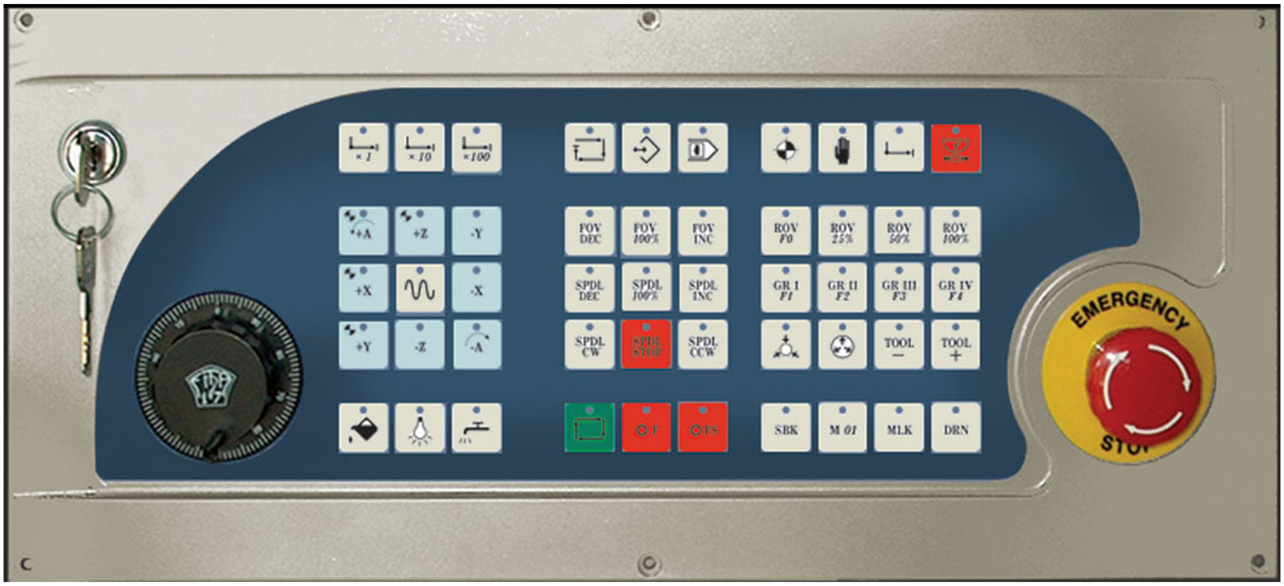
1. Press **ZRN** button to return to the reference position. That is one of the mode select buttons.

2. To decrease the feedrate, press the rapid traverse override switch. When the tool returns to the reference position an indicating lamp goes on, specifying the operation completion.

3. Choose the feed axis and the direction for reference position return. Continue pressing the button until the tool returns to the reference position. To the deceleration point the tool is moved at rapid traverse rate and then at **FL** speed to the reference position. It is set by a parameter.

4. Repeat the same operations for the other axes if necessary.

The above operations are only examples. For a full description of the procedure refer to the manual provided by the machine builder. The graphical interpretation of the buttons to the end is also an example. The actual realization depends on the machine builder.



◆ AUTOMATIC COORDINATE SYSTEM SETTING

If the corresponding parameter for automatic coordinate system setting is specified, the coordinate system is determined automatically when a reference position return is made. If **a**, **b**, **c** and **d** are specified in the corresponding parameters, the system specifies such a workpiece coordinate system that the tip of the basic tool or the reference position of the tool holder have coordinates **X=a**, **Y=b**, **Z=c** and **4th=d** after reference position return.

RESTRICTIONS:

◆ MOVING THE TOOL AFTER RETURN

Once the reference position return completion lamp goes on, i.e. the operation is completed, the tool cannot be moved until this mode is disabled.

◆ REFERENCE POSITION RETURN COMPLETION LAMP

The reference position return completion lamp can go off by either of the following operations:

- Moving from the reference position.

- Entering in emergency stop state.

◆ THE DISTANCE TO RETURN TO THE REFERENCE POSITION

For the distance to return the tool to the reference position, refer to the manual issued by the machine builder.

3.2 JOG FEED

In the jog mode, pressing a feed axis and direction selection switch on the machine operator’s panel, moves the tool continuously along the selected axis in the selected direction.

The jog feedrate is specified in the table below.

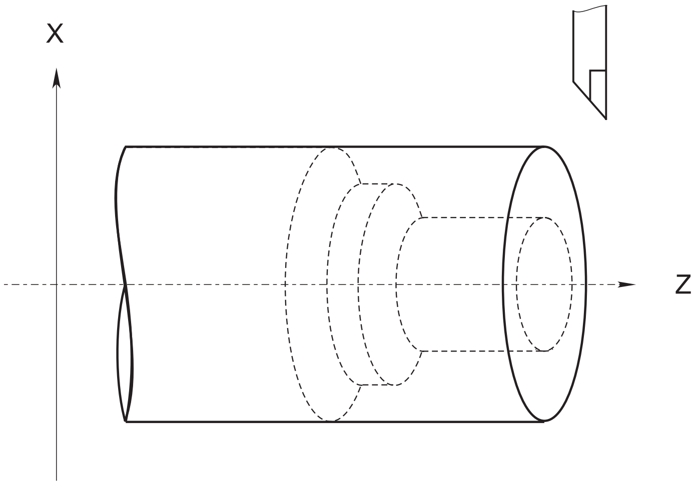
Rotary switch position		Rotary switch position	
Metric input (mm/min)	Inch input (inch/min)	Metric input (mm/min)	Inch input (inch/min)
0	0	50	2.0
2.0	0.08	79	3.0
3.2	0.12	126	5.0
5.0	0.2	200	8.0
7.9	0.3	320	12
12.6	0.5	500	20
20	0.8	790	30
32	1.2	1260	50

The current jog feed can be viewed on the positional screen (POS) when such an data is not displayed on the machine operator’s panel.

Note

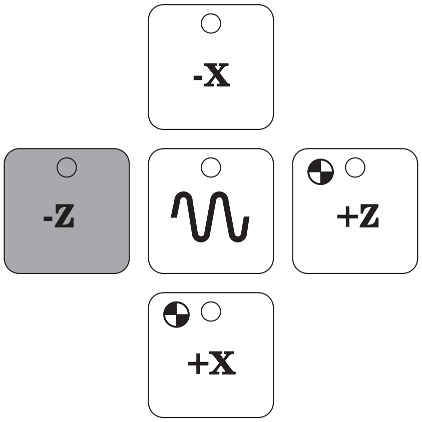
The feedrate error is about 3%.

The jog feedrate can be adjusted using the corresponding keys. Pressing the rapid traverse switch moves the tool at rapid traverse regardless of the position of the og feed button. Manual operation is allowed for one axis at a time.



While the switch is in on position, the tool moves in a direction specified by the switch.

PROCEDURE FOR JOG FEED



1. Press the jog button - one of the mode selection switches.

2. Press the feed axis and direction selection buttons to select the direction of moving the tool. While the button is pressed, the tool moves at a feedrate specified in the table. The too stops when button is released.

3. The jog feedrate can be set by the corresponding buttons.

4. Pressing the rapid traverse button in jog feed and selected direction of the tool, moves the tool in rapid traverse rate while the rapid traverse button is pressed. If a rapid traverse override is specified during rapid traverse, the latter is effective.

The above operations are just examples. For more details refer to the manual provided by the machine builder.

<u>RESTRICTIONS:</u>

◆ ACCELERATION/DECELERATION FOR RAPID TRAVERSE

Feedrate, time constant and the method of automatic acceleration/deceleration for manual rapid traverse are the same as G00 in a program command.

◆ CHANGE OF MODES

Jog feed is not enabled when pressing the keys for changing the feed axis and direction selection buttons. To enable jog feed, enter the jog mode first, then press the other keys.

◆ RAPID TRAVERSE PRIOR TO REFERENCE POSITION RETURN

If reference position return is not performed after power-on, pushing rapid traverse button does not activate rapid traverse but the jog feedrate. This function can be disabled by setting a parameter.

3.3 INCREMENTAL FEED

In incremental (step) mode, pressing the feed axis and the direction selection buttons on the machine operator’s panel moves the tool one step along the selected axis in the selected direction. The minimum distance the tool is moved is the least input increment. Each step can be 1, 10, 100 or 1000 times the least input increment.

This mode is effective when a manual pulse generator is not connected.

PROCEDURE FOR INCREMENTAL FEED

1. Press the step button - one of the mode selection buttons.

2. Select the distance for each tool movement.

3. Press the feed axis and the direction selection buttons. Each time this button is pressed, the tool moves one step. The feedrate is the same as the jog feedrate.

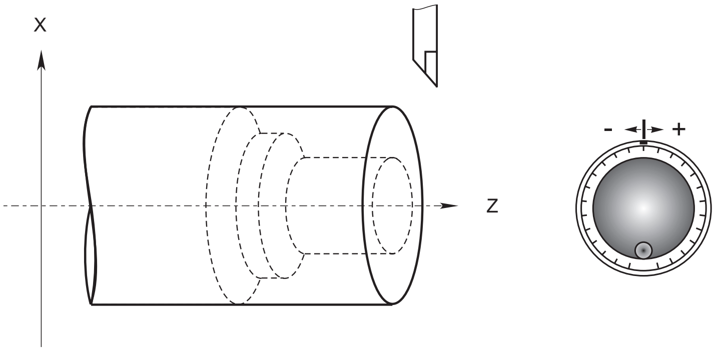
4. If the axis direction button is pressed after rapid traverse has been selected, movement is performed at a rapid traverse rate. If rapid traverse override is specified during incremental feed, the latter is affective.

The above operations are just examples. For more details refer to the manual issued by the machine builder.

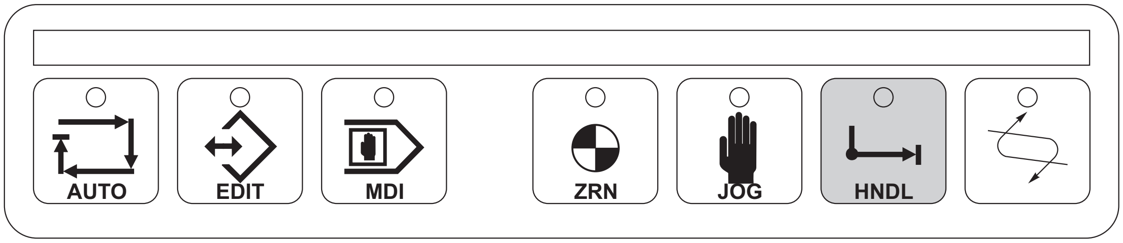
3.4 MANUAL HANDLE FEED

In the handle mode the tool can be moved by rotating the manual pulse generator on the machine operator’s panel. Select the axis the tool to be moved along with the handle feed axis selection buttons.

The minimum distance the tool is moved when using the manual pulse generator is equal to the least input increment. The minimum distance the tool is moved when the manual pulse generator is rotated by one graduation can 10 times the least input increment or a magnification specified by a parameter (usually 100 times).



PROCEDURE FOR MANUAL HANDLE FEED



1. Press the button selecting manual handle feed mode - one of the mode selection buttons.

2. Select the axis the tool has to be moved along by pressing the axis selection button.

3. Select the magnification for the distance of the tool movement when the handle is rotated by one graduation. The minimum distance the tool is moved when the manual pulse generator is rotated by one graduation is equal to the least input increment.

4. Move the tool along the selected axis by rotating the handle. Rotating the handle 360 degrees moves the tool at a distance equivalent to 100 graduations.

The above operations are just examples. For more details refer to the manual issued by the machine builder.

◆ AVAILABILITY OF MANUAL PULSE GENERATOR IN JOG MODE

In handle feed mode, the manual pulse generator is enabled or disabled by a parameter. When the corresponding parameter is set, both manual handle feed and incremental feed are enabled.

◆ AVAILABILITY OF MANUAL PULSE GENERATOR IN TEACH IN JOG MODE

In **TEACH IN JOG** mode, the manual pulse generator is enabled or disabled by a parameter.

WARNING:

Rotating the handle quickly with a large magnification such as x100 moves the tool too fast. This can destroy the machine.

Note:

The manual pulse generator should be rotated at a rate of five rotations per second or lower. If the rate is higher, the distance the tool moves may not match the graduation of the manual pulse generator.

4. AUTOMATIC OPERATION

Programmed operation of a **CNC** machine is **called automatic operation.**

This chapter explains the following types of automatic operation:

Memory operation
Operation by executing a program registered in **CNC** memory.

MDI operation
Operation by executing a block entered from the **MDI** panel.

DNC operation
Function for operating a machine while reading a program from an input/output unit.

4.1 MEMORY OPERATION

Programs are registered in the **CNC** memory in advance. When one of these programs is selected and the cycle start switch on the machine operator's panel is pressed, automatic operation starts and the cycle start lamp goes on.

When the feed hold switch on the machine operator's panel is pressed during automatic operation, the automatic operation is temporarily stopped. When the cycle start button is pressed again, the automatic operation is resumed.

When the reset switch on the TFT/MDI panel is pressed, the automatic operation is terminated and the reset state is set.

The following procedure is given as an example. For actual operation refer to the manual supplied by the machine builder.

PROCEDURE FOR MEMORY OPERATION

1. Press the **EDIT** mode selection button.

2. Select a program from the registered ones. To do this follow the steps below:

- **2.1.** Press **[PRGRM]** button and then the soft key **[LIB]**. Library screen with list of the programs will show up.
- **2.2.** Select the desired program.

3. Press the **AUTO** mode selection button.

4. Press the cycle start button on the machine operator's panel. The automatic operation starts and the cycle start lamp goes on. When the automatic operation terminates, the cycle start lamp goes off.

5. To stop or cancel memory operation, follow the steps below:

A. *Stopping the memory operation*

Press the feed hold button on the machine operator's panel. The feed hold lamp goes on and the cycle start lamp goes off. The machine executes the following operations:

(a) If the machine is moving, feed operation decelerates and stops.

(b) If dwell is performed, it is stopped.

(c) The current operation specified by **M, S** or **T** command is continued.

(d) If a tread cutting cycle is operated (**G76, G32, G92**) the machine stops after the execution of the block containing **G76, G32** or **G92**.

When the cycle start button on the machine operator's panel is pressed while the feed hold lamp is on, the machine operation is resumed.

B. ***Terminating memory operation***

Press **[RESET]** button on the TFT/MDI panel.

The automatic operation is terminated and the reset state is set. When reset is applied during movement, movement decelerates and stops.

EXPLANATIONS:

MEMORY OPERATION

After memory operation is started, the following commands are executed:

(1) A block is read from the specified program.

(2) The block command is decoded.

(3) The command execution is started.

(4) The command in the next block is read.

(5) Buffering is executed. This means that the command is decoded to allow immediate execution.

(6) Right after the preceding block is executed, execution of the next block can be started. The reason for this is command buffering.

(7) Hereafter, memory operation can be executed by repeating the steps from (4) to (6).

STOPPING AND TERMINATING MEMORY OPERATION

Memory operation can be stopped using one of the following two methods: specifying a stop command or pressing a key on the machine operator’s panel.

- The stop command includes **M00** (program stop), **M01** (optional stop) and **M02** and **M30** (program end).

Note: **M02** and **M30** put the cursor at the beggining of the program, if this is set by a parameter (**P019** - see Parameters' Description).

- There are two buttons that are used to stop memory operation: the feed hold button and **[RESET]** button.

PROGRAM STOP (M00)

Memory operation is stopped after a block containing **M00** is executed. When the program is stopped, all existing modal information remains unchanged as in single block operation. The operation can be resumed by pressing the cycle start button. Operation may vary depending on the machine builder. For more details refer to the manual issued by the machine builder.

OPTIONAL STOP (M01)

Similarly to **M00**, memory operation is stopped after a block containing **M01** has been executed. This code is effective only when the optional stop switch in the machine operator’s panel is set to **ON**. Operation may vary depending on the machine builder. For more details refer to the manual supplied by the machine builder.

PROGRAM END (M02, M30)

When **M02** or **M30** command is read (specified at the end of the main program), memory operation is terminated and the reset state is entered.

FEED HOLD

When the feed hold button on the operator’s panel is pressed during memory operation, the tool stops as an exception when **G92, G76** and **G32** are executed.

RESET

The automatic operation can be stopped and the system can enter in reset mode after pressing **[RESET]** key or after receiving an external reset signal. When reset operation is applied to the system during a tool movement, the tool stops.

OPTIONAL BLOCK SKIP (BLOCK DELETE FUNCTION)

When the optional block skip button on the machine operator’s panel is turned on, blocks starting with slash (*/*) are ignored.

4.2 MDI OPERATION

In the **MDI** mode, the program can be entered in the same format as normal programs and executed from the **MDI** panel.

MDI operations are used for simple tests.

The following operations are given as an example. For more details refer to the manual provided by the machine builder.

PROCEDURE FOR MDI OPERATION

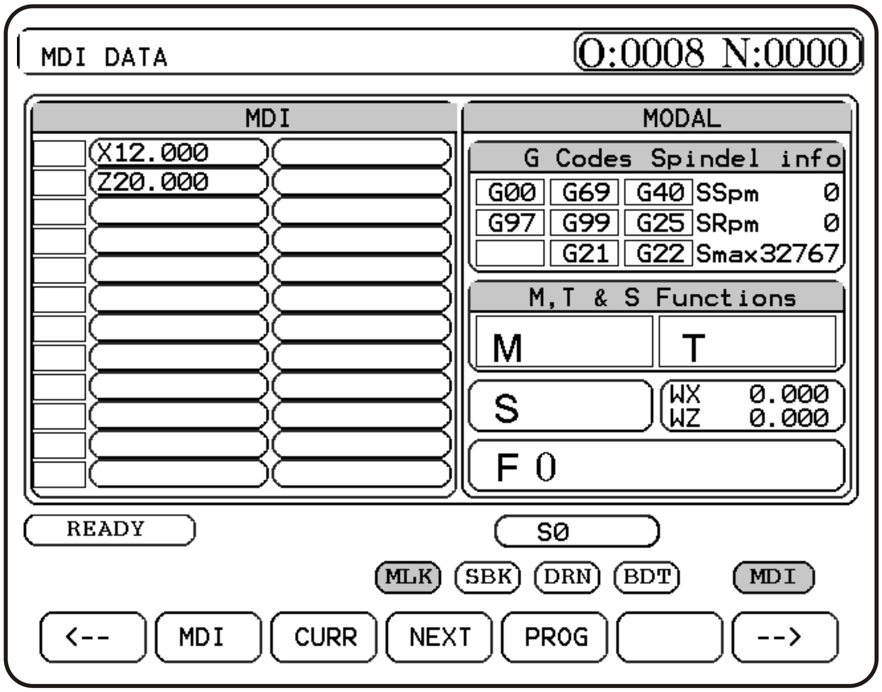
Example: **X10.5 Z200.5**

Only one command block can be entered from the TFT/MDI panel.

1. Press **MDI** key from the mode select buttons.

2. Press the **[PRGRM]** button.

3. Press the soft key **[MDI]** to display a screen with **MDI** at the top left.



MDI DATA O:0008 N:0000

MDI

X12.000
Z20.000

MODAL

G Codes Spindel info

G00	G69	G40	SSpm 0
G97	G99	G25	SRpm 0
	G21	G22	Smax32767

M,T & S Functions

M
T
S
WX 0.000
WZ 0.000
F 0

READY
S0

4. Input **“X 10.5”** by the address/numeric keys.

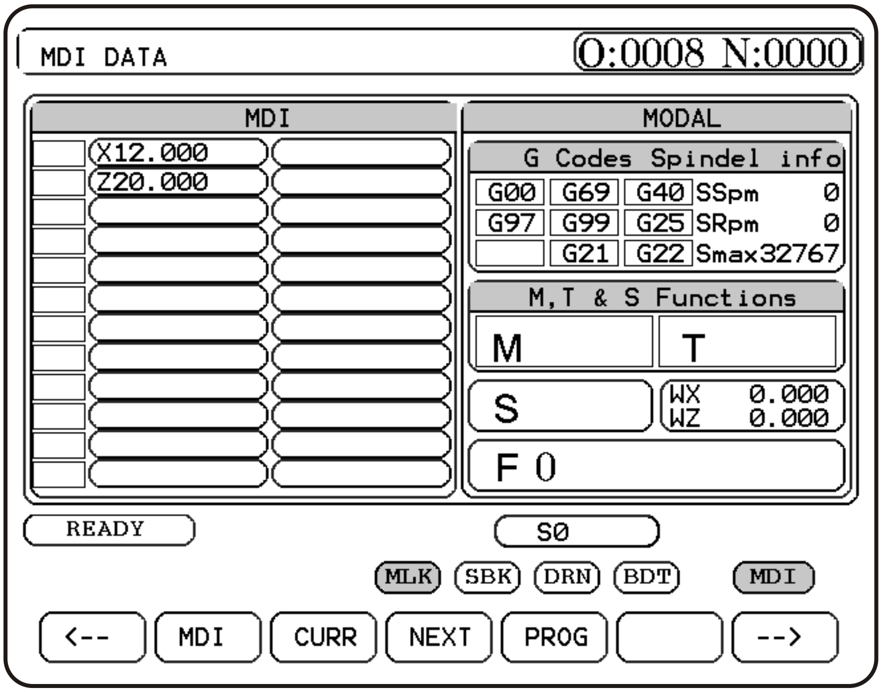
5. Press the **[INPUT]** key.

The data **X** and **10.5** is input and displayed. If you notice an error while entering data, before pressing the **[INPUT]** key, press the **[CAN]** key and reenter the correct data.

6. Input **“Z200.5”** by the address/numeric keys.

7. Press the **[INPUT]** button.

The data **Z** and **200.5** is input and displayed. If you have pressed wrong keys, do the operation again following the instructions described above.



MDI DATA O:0008 N:0000

X12.000
Z20.000

G00	G69	G40	SSpm 0
G97	G99	G25	SRpm 0
	G21	G22	Smax32767

M T
S
WX 0.000
WZ 0.000
F 0

8. Press the **[OUTPUT/START]** key or the cycle start button on the machine operator's panel (depending on the machine builder).

The **[RESET]** button clears the contents of the whole buffer.

WARNING:

Modal **G** codes cannot be canceled. Reenter the correct data again.

LIMITATIONS:

- A single **MDI** operation executes a single input block. Two or more blocks cannot be executed simultaneously.
- The end-of-block symbol (;) need not be entered.

- A macro or subprogram call cannot be specified.

- In **MDI** operations, the screen **SETTINGS** determine whether the commands are absolute or incremental.

- The input block is cleared when the **MDI** operation is completed or when **reset** is specified.

4.3 DNC OPERATION

In **DNC** operation, the machine is not operated by a program registered in the memory of the **CNC**. Instead, the program is read directly from a connected input/output unit. This mode is used when the program is too large to be loaded in the memory of the **CNC**.

PROCEDURE FOR DNC OPERATION

1. Prepare the input/output unit for transmitting.

2. Select **AUTO** mode and press the **[PRGM]** button to display some of the program screens.

3. Press the **[INPUT]** button. A message on the botton of the screen notifies the transfer operation.

4. Wait until the message **"DNC connection"** is displayed.

5. Press the cycle start button.

DNC operation starts. It can be stopped and resumed in the same way as the memory operation.

EXPLANATIONS:

- In **DNC** operation mode, the current program can call a subprogram registered in memory.

- In **DNC** operation mode, the current program can call a custom macro. However, repeat and brunch instructions cannot be specified.

- In **DNC** operation mode, the current program can call a custom macro registered in the **CNC** memory.

- In **DNC** operation mode, to return to the main program from the current subprogram or macro, a sequence number **M99P****** cannot be specified.

- In **DNC** operation mode the program cannot be displayed. Only the current and the following blocks can be displayed.

- In **DNC** operation mode, all the data from the input/output unit is buffered, so that an uninterrupted data stream is provided and the commands can be processed at the maximum speed. For that reason there have to be checked the screens with current/ next block but not the indication on the input/output unit itself, to know the currently executed point of the program.

5. TEST OPERATION

The following functions are used to check before actual machining is performed whether the machine operates as specified by the created program.

MACHINE LOCK AND AUXILIARY FUNCTION LOCK
FEEDRATE OVERRIDE
RAPID TRAVERSE OVERRIDE
DRY RUN
SINGLE BLOCK EXECUTION

5.1 MACHINE LOCK AND AUXILIARY FUNCTION LOCK

To display the change in the position without moving the tool, use **machine lock.**

PROCEDURE FOR MACHINE LOCK AND AUXILIARY FUNCTION LOCK

- **MACHINE LOCK**

Press the machine lock button on the machine operator's panel. The tool is not moved but the position along each axis on the display is changed as if the tool was moving.

WARNING:

The coordinate relation between the workpiece and the machine can change after executing the **machine lock** function during automatic operation. If such a situation occurs, reset the coordinate system for the workpiece by specifying a command for coordinate system setting or by executing a manual reference position return.

◆ AUXILIARY FUNCTION LOCK

Press the auxiliary function lock button on the machine operator’s panel. **M, S** and **T** codes are disabled and are not executed. For more information regarding the auxiliary function lock refer to the manual provided by the machine builder.

Note: **This mode is used by the machine builder.**

RESTRICTIONS:

◆ M, S AND T COMMANDS ONLY BY MACHINE LOCK

M, S and **T** commands are executed in the machine lock state.

◆ REFERENCE POSITION RETURN UNDER MACHINE LOCK

When **G27**, **G28** or **G30** command is specified and the machine is in machine lock state, the command is accepted but the tool is not moved to the reference position and the reference position return lamp does not go on.

◆ M CODES THAT ARE NOT LOCKED BY THE AUXILIARY FUNCTION LOCK

The commands **M00, M01, M02, M30, M98** and **M99** are executed even in the auxiliary function lock state.

5.2 FEEDRATE OVERRIDE

The programmed feedrate can be reduced or increased by a percentage using the corresponding keys. This function is used to check the program.

For example, when a feedrate of 100 mm/min is specified in the program, setting the override to 50% moves the tool at 50 mm/min.

Check the machining by changing the feedrate specified in the program.

PROCEDURE FOR FEEDRATE OVERRIDE

Set the feedrate override using the buttons on the machine operator's panel before or during automatic operation. For more details refer to the manual provided by the machine builder.

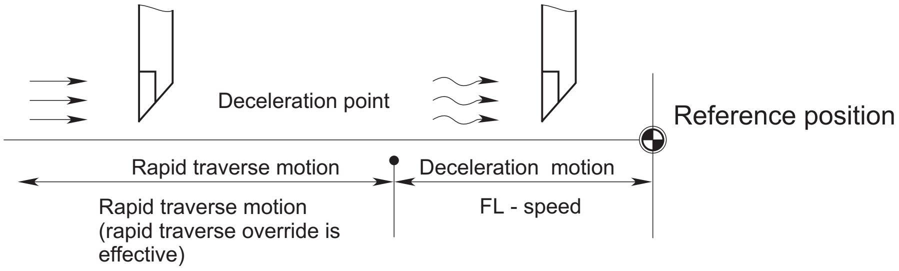
<u>RESTRICTIONS:</u>

- **OVERRIDE RANGE**

The override that can be specified ranges from 0 to 150% by a step of 10%.

5.3 RAPID TRAVERSE OVERRIDE

To the rapid traverse rate can be applied override by F0, 25%, 50% and 100%.



RAPID TRAVERSE OVERRIDE

Select one of the four overrides for the rapid traverse. For more information regarding rapid traverse override refer to the manual provided by the machine builder.

The following types of rapid traverse are available. Rapid traverse override can be applied to each of them.

1) Rapid traverse by G00

2) Rapid traverse during a canned cycle

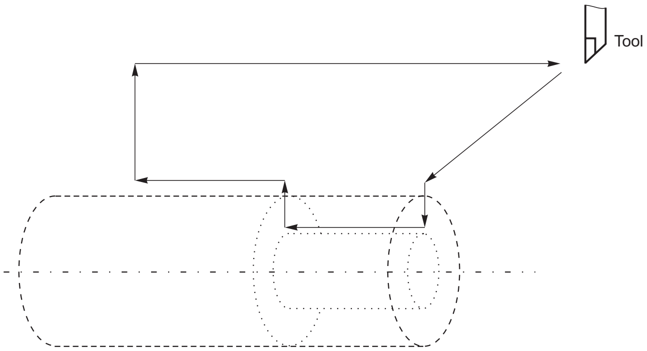
3) Rapid traverse in G27, G28 and G30

4) Manual rapid traverse

5) Rapid traverse in manual reference position return

5.4 DRY RUN

The tool is moved at a feedrate specified by the operator regardless of the feedrate specified in the program. This function is used for checking the movement of the tool when the workpiece is not placed on the table.



PROCEDURE FOR DRY RUN

Press the dry run button on the machine operator’s panel during automatic operation.

The tool moves at a feedrate specified by the operator. To change the feedrate use the rapid traverse button. For more information regarding dry run refer to the appropriate manual provided by the machine builder.

◆ DRY RUN FEEDRATE

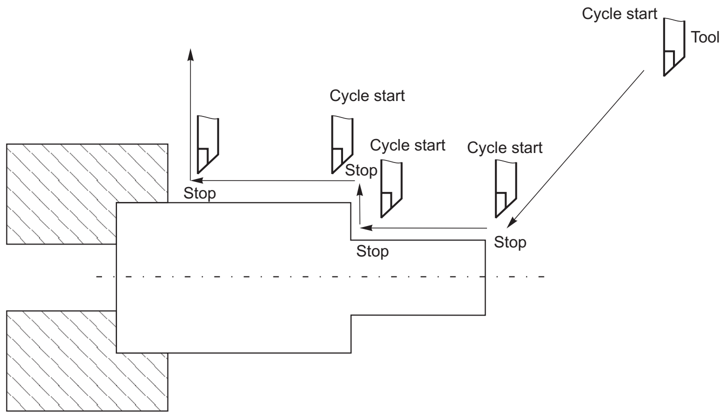
The dry run feedrate changes as shown in the table below according to the rapid traverse button and the corresponding parameter.

Rapid traverse button	Program command	
	Rapid traverse	Feed
ON	Rapid traverse rate	Jog maximum feedrate
OFF	Jog feedrate or rapid traverse rate *	Jog feedrate

*** -** Jog feedrate if the corresponding parameter is 1.
Rapid traverse rate if the same parameter is 0.

5.5 SINGE BLOCK

Pressing the single block button starts the single block mode. When the cycle start button is pressed in this mode, the tool stops after a single block in the program is executed. Check the program in the single block mode by executing it block by block.



PROCEDURE FOR SINGLE BLOCK

1. Press the single block button on the machine operator's panel. The execution of the program is stopped after the current block is executed.

2. Press the cycle start button to execute the next block. The tool stops after the block is executed.

For more information regarding single block execution refer to the manual provided by the machine builder.

◆ SINGLE BLOCK EXECUTION AND REFERENCE POSITION RETURN

If **G28** to **G30** commands are issued, the single block function is effective at the intermediate point.

◆ SUBPROGRAM CALL AND SINGLE BLOCK EXECUTION

Single block stop is not performed if the block contains **M98P_; M99;** or **G65**.

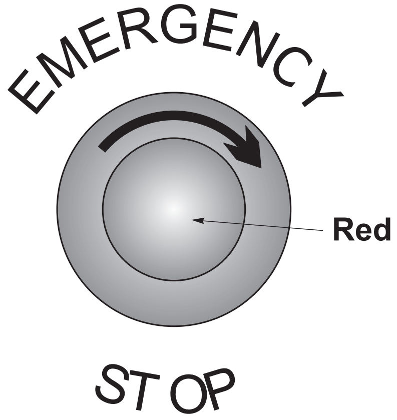
However, if the block contains an address other than **O, N** or **P**, the single block stop is performed in a block containing **M98P_** or **M99** command.

6. SAFETY FUNCTIONS

To stop the machine immediately for safety, press the Emergency Stop button. To prevent the tool from exceeding the stroke ends, special checks are available. This chapter describes emergency stop, overtravel check and stroke check.

6.1 EMERGENCY STOP

If you press the Emergency stop button on the machine operator's panel, the machine movement immediately stops.

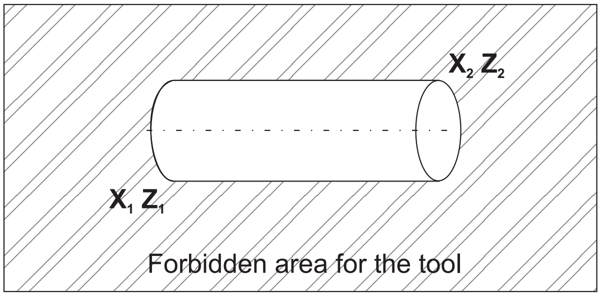


This button is locked when pressed. Although it varies depending on the machine builders, the button can usually be unlocked by twisting.

The Emergency Stop interrupts the current to the motor. Causes of troubles must be removed before the button is released.

6.2 STROKE CHECK

There can be specified an area in which the tool is allowed to move.



When the tool exceeds the stroke limit, an alarm is displayed and the tool is decelerated and stopped.

When the tool enters the forbidden area and the alarm is displayed, the tool can be moved in a direction reversed to that of coming.

◆ STROKE LIMIT

The boundaries are set by parameters.

Outside these boundaries is a forbidden area. The machine builder usually sets this area as a maximum stroke.

◆ OVERRUN AMOUNT OF STROKE LIMIT

If the maximum rapid traverse rate is F mm/min, the maximum overrun amount after the limit L in mm, is obtained from the following expression:

L mm = F/7500

The tool enters in the specified forbidden area by up to L mm.

◆ RELEASING THE ALARMS

If a stroke check alarm occurs, manually retract the tool from the forbidden area in a direction opposite to the displayed alarm direction. Press the **[RESET]** key to cancel the alarm.

ALARMS

Number	Message	Contents
6n0	**OVER TRAVEL: +n**	Exceeded the n-th axis (1-4) **+** direction
6n1	**OVER TRAVEL: -n**	Exceeded the n-th axis (1-4) **-** direction

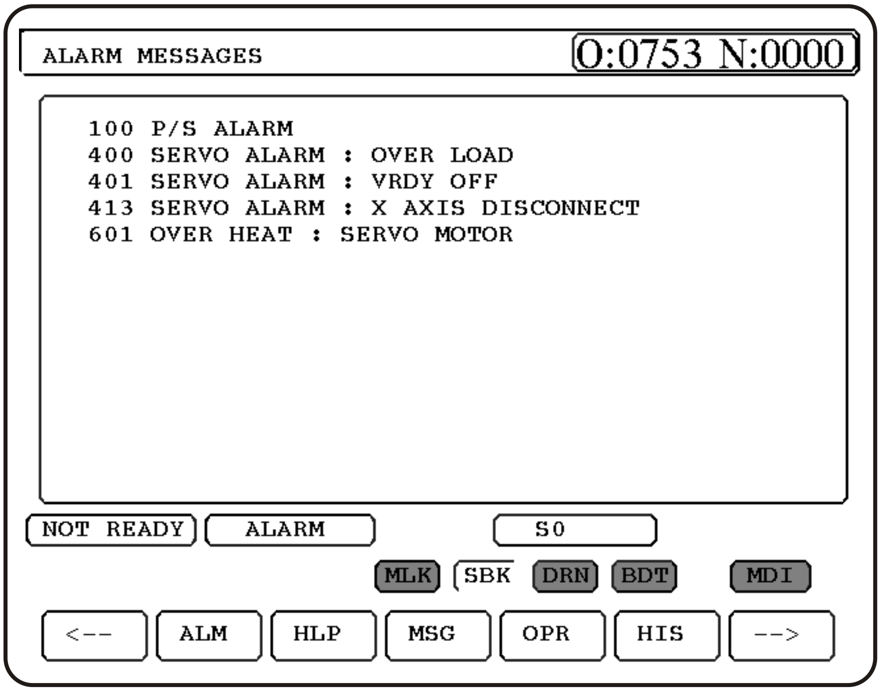
7. ALARM AND SELF-DIAGNOSIS FUNCTIONS

When an alarm occurs, the corresponding alarm screen appears to indicate the cause of the alarm. The causes of alarms are classified by error codes. The system may sometimes seem to be at a halt, although no alarm is displayed. In this case, the system may perform invisible to the user operations. The state of the system can be checked using the self-diagnosis functions.

7.1 ALARM DISPLAY

◆ **ALARM SCREEN**

When an alarm occurs, the following alarm screen appears.

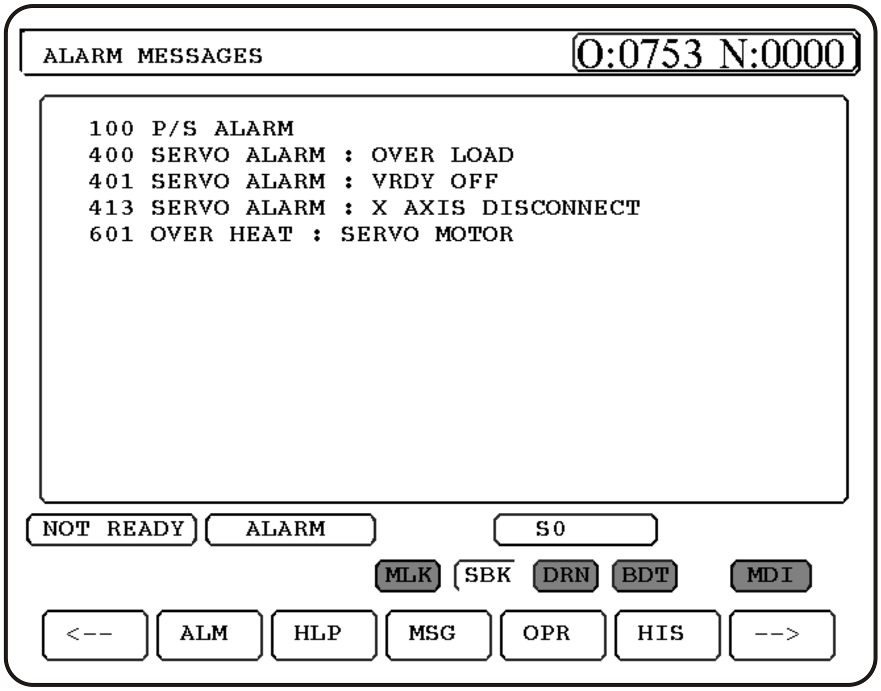


ALARM MESSAGES O:0753 N:0000

```
100 P/S ALARM
400 SERVO ALARM : OVER LOAD
401 SERVO ALARM : VRDY OFF
413 SERVO ALARM : X AXIS DISCONNECT
601 OVER HEAT : SERVO MOTOR
```

◆ ANOTHER METHOD FOR DISPLAYING ALARMS

In some cases the alarm screen may not be displayed. Instead, the message **ALARM** will blink at the bottom of the screen.



```
ALARM MESSAGES                    O:0753 N:0000

   100 P/S ALARM
   400 SERVO ALARM : OVER LOAD
   401 SERVO ALARM : VRDY OFF
   413 SERVO ALARM : X AXIS DISCONNECT
   601 OVER HEAT : SERVO MOTOR
```

In this case, to display the alarm screen do the following steps:

1. Press the **[OPR/ALARM]** button.

2. Press the soft key **[ALM]**.

◆ RESET OF THE ALARM

Error codes and messages indicate the cause of the alarm. To recover from the alarm, eliminate the cause and press the **[RESET]** key.

◆ ERROR CODES

The error codes are classified as follows:

No 000 to 250: Program errors

No 300 to 399: Fatal errors

No 400 to 499: Servo alarms

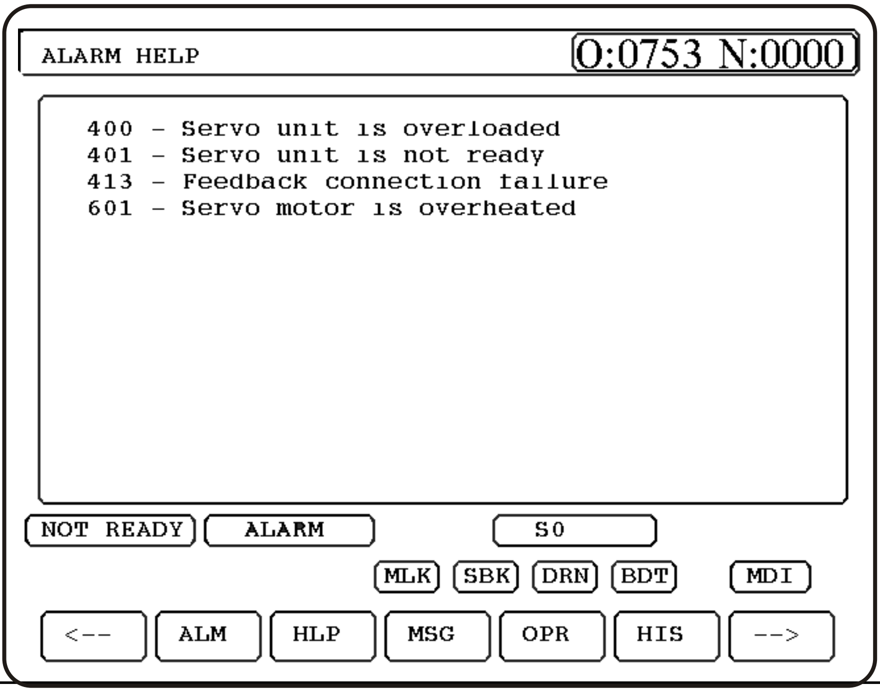
No 600 to 601: Overheat alarms

No 610 to 699: Overtravel alarms

For more information regarding the alarms and their codes see the appendix.

◆ ERROR CODES DESCRIPTION IN BRIEF

An error code screen is displayed by pressing the buttons **[PgUp]** or **[PgDn]**. It contain a brief description of the probable cause.



```
ALARM HELP                                  O:0753 N:0000

   400 - Servo unit is overloaded
   401 - Servo unit is not ready
   413 - Feedback connection failure
   601 - Servo motor is overheated
```

Note:

If a rimmed alarm without a number is displayed and in the upper left corner *"SYSTEM ALARM"* message is seen, it means that a system error has been detected and further machining is prohibited.

Contact the service technicians to find out the cause. Write down the error message beforehand.

7.2 CHECKING BY SELF - DIAGNOSTIC SCREEN

The system may sometimes seem to be at a halt, although no alarm message is displayed. In this case the system may be performing some invisible to the user operations. The state of the system can be checked by the self-diagnostic functions.

PROCEDURE FOR SELF-DIAGNOSIS

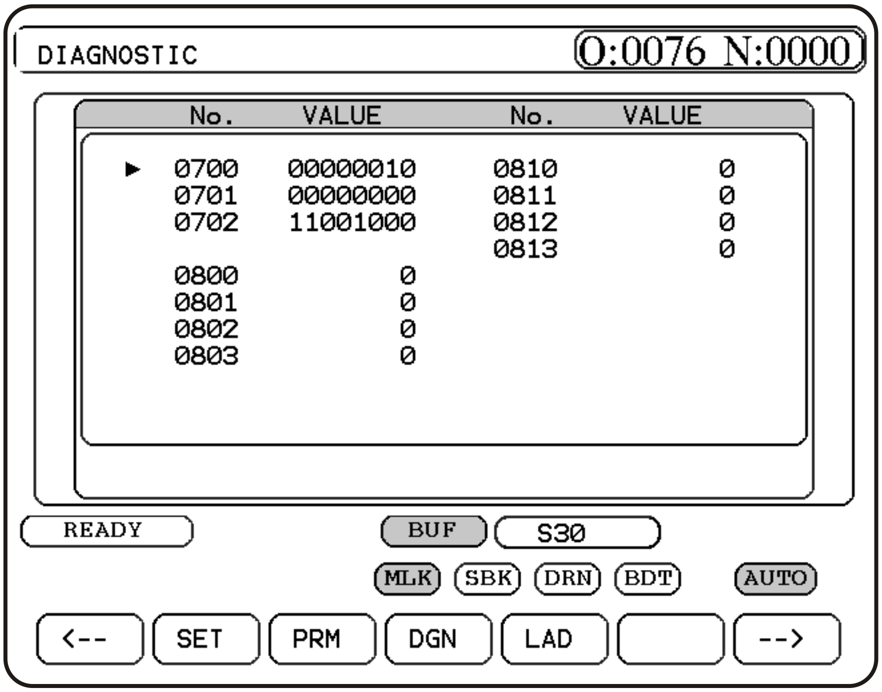
1. Press the function key **[DGNOS/PARAM]**.

2. Press the soft key **[DGN]**.

3. The diagnostic screen has more than one page. Select the screen doing the following operations:

(1) Change the page by the page change keys.

(2) **-** Press the **[No]** key.
- Input the diagnostic number to be displayed.
- Press the **[INPUT]** key.



```
DIAGNOSTIC                                   O:0076 N:0000

      No.      VALUE             No.      VALUE
  ▶   0700     00000010          0810         0
      0701     00000000          0811         0
      0702     11001000          0812         0
                                 0813         0
      0800            0
      0801            0
      0802            0
      0803            0
```

	#7	#6	#5	#4	#3	#2	#1	#0
0700		**CSCT**	**CITL**	**COVZ**	**CINP**	**CDWL**	**CMTN**	**CFIN**

When the digit is **“1”**, the corresponding status is effective.

CFIN: **M, S, T** or **B** function is being executed.

CMTN: A tool move command is being executed.

CDWL: Dwell is being executed.

CINP: An in-position check is being executed.

COVZ: Override is 0% (feedrate 0).

CITL: Interlock signal is turned on.

CSCT: Speed arrival signal of spindle is expected.

When automatic operation is refused, the cause is displayed.

	#7	#6	#5	#4	#3	#2	#1	#0
0701			**CRST**					

CRST: One of the following: a signal from the reset button on the **MDI** panel, emergency stop or reset from an external unit.

Indicates automatic operation stop or feed hold status. Used for troubleshooting.

	#7	#6	#5	#4	#3	#2	#1	#0
0712	**STP**	**REST**	**EMS**		**RSTB**			**CSU**

STP: Flag which stops the automatic operation. It is set in one of the following conditions:

- External reset signal has been received.

- Emergency stop signal has been received.

- Feed hold signal has been received.

- Reset button on the TFT/MDI panel is turned on.

- The mode is changed to the manual mode, such as **JOG, HANDLE/STEP, TEACH IN JOG** or **TEACH IN HANDLE.**

- Other alarm is generated.

REST: This is set when external reset, emergency stop or reset button is set to on.

EMS: This is set when the emergency stop is set to on.

RSTB: This is set when the reset button is on.

CSU: This is set when the emergency stop is turned on or when a servo alarm has been generated.

0800 . . . . . **0803** - Current error in servo contour along each axis.

0820 . . . . . **0823** - Machine position along each axis.

8. DATA INPUT/OUTPUT

8.1 PROGRAM INPUT/OUTPUT

8.1.1 Program input

This chapter describes how to load a program through the serial connection from a **PC** or from a portable device for storing data and programs (**DataBag**).

PROCEDURE FOR PROGRAM INPUT

1. Make sure that the device is ready to transmit.

2. Press the **[EDIT]** mode button on the machine operator’s panel.

3. Set the program save switch to “unlocked”.

4. Press the **[PRGRM]** button to display the program screen.

5. Press the **[INPUT]** button.
The programs are stored with the number written in it.

<u>Note:</u>

*To abandon the input mode press the **[RESET]** key.*

◆ **MULTIPLE PROGRAMS INPUT**

When the device stores more than one program, they are input sequentially to the end or until an alarm occurs.

◆ PROGRAM NUMBERS IN THE PERIPHERAL DEVICE

The number of the program in the device is assigned to the program. If the program is without **O** number, the first available in the system number is assigned.

◆ ERROR CODES

Number	Description
70	The size of memory is insufficient to store the whole program
72	Too many programs in the memory
73	An attempt was made to store a program with an existing program number
74	Invalid program number

8.1.2 Program output

This chapter describes how to store a program through the serial connection to a **PC** or to a portable device for storing data and programs (**DataBag**).

PROCEDURE FOR PROGRAM OUTPUT

1. Make sure that the device is ready to receive.

2. Press the **[EDIT]** mode button on the machine operator's panel.

3. Press the **[PRGRM]** button to display the program screen.

4. Specify the number of the desired program.

5. Press the **[OUTPT/START]** button.

The program with the specified number is output.

If the **[OUTPT/START]** button is pressed until the **[ALTER]** button is kept pressed, all the programs are output to the memory.

8.2 OFFSET DATA INPUT AND OUTPUT

8.2.1 Offset data input

The offset data is loaded into the memory of the **CNC** using serial connection from a **PC** or from a device for storing data (**DataBag**). The input format is the same as the offset value output.

PROCEDURE FOR OFFSET DATA INPUT

1. Make sure that the device is ready to transmit.

2. Press the **[EDIT]** mode button on the machine operator’s panel.

3. Press the **[MENU/OFFSET]** button and the soft key **[OFS]** to display the offset screen.

4. Press the **[INPUT]** button.

After the input operation is completed the offset data will be displayed on the screen.

8.2.2 Offset data output

The offset data is output from the memory of the **CNC** using serial connection to a **PC** or to a device for storing data (**DataBag**).

PROCEDURE FOR OFFSET DATA OUTPUT

1. Make sure that the device is ready to receive.

2. Press the **[EDIT]** mode button on the machine operator's panel.

3. Press the **[MENU/OFFSET]** button and the soft key **[OFS]** to display the offset screen.

4. Press the **[OUTP/START]** button.

OUTPUT FORMAT

The output format is as follows:

For tool compensation

N_O_

where:

N_: Offset number
O_: Tool compensation value

8.3 PARAMETERS INPUT AND OUTPUT

8.3.1 Parameters input

Parameters are loaded into the memory of the **CNC** using serial connection from a **PC** or from a device for storing data (**DataBag**). The input format is the same as the output format. When a parameter is loaded which has the same data number as a parameter already registered in the memory, the loaded parameter replaces the existing one.

1. Make sure that the device is ready to transmit.

2. Press the function button **[DGNOS/PARAM]** on the machine operator’s panel to display the “**SETTINGS”** screen.

3. Enter 1 in response to the prompt for parameters’ change **(PRM MODIFY)**. The parameters can be changed now.

4. Press the soft key **[PRM]** to display the **“PARAMETERS”** screen.

5. Press the **[INPUT]** button.

The parameters are stored in the memory. After the operation is completed, the screen message **“DATA IMPORT”** disappears.

6. Enter **0** in response to the prompt for parameters change.

7. Turn the power off and then on.

8.3.2 Parameters output

Parameters are output from the **CNC** using serial connection to a **PC** or to a data storage device (**DataBag**).

PARAMETERS OUTPUT

1. Make sure that the device is ready to receive.

2. Press the **[EDIT]** mode button on the machine operator's panel.

3. Press the functional button **[DGNOS/PARAM]** to display the parameter screen.

4. Press the **[OUTPT/START]** button.
All parameters are output in the format defined below.

OUTPUT FORMAT

The output format is as follows:

N_P_

where:

N_: Parameter number
P_: Parameter setting value

8.4 CUSTOM MACRO VARIABLES INPUT/OUTPUT

8.4.1 Custom macro variables input

The values of the custom macro variables **(#100 ... #131 and #500 ... #531)** are loaded in the **CNC** memory using serial connection either from a **PC** or from a data storage device (**DataBag**). The same format as from output is used. When a custom macro variable is loaded in the memory, the new value replaces the old one.

CUSTOM MACRO COMMON VARIABLES INPUT

1. Make sure that the device is ready to transmit.

2. Press the **[EDIT]** mode button on the machine operator’s panel.

3. Press the functional button **[OFSET]** to display the **“VARIABLES”** screen.

4. Press the **[INPUT]** button. Variables are loaded into the memory of the **CNC.**

Note:

*The common variables **(#100 .. #131 and #500 ... #531)** can be input and output. The common variables from **#100** to **#131**, however, do not save their values after the power is off.*

8.4.2 Custom macro variable output

The values of the custom macro variables **(#100 ... #131 and #500 ... #531)** are output from the **CNC** memory using serial connection either to a **PC** or to a data storage device **(DataBag)**.

CUSTOM MACRO COMMON VARIABLES OUTPUT

1. Make sure that the device is ready to receive.

2. Press the **[EDIT]** mode button on the machine operator’s panel.

3. Press the functional button **[OFSET]** to display the **“VARIABLES”** screen.

4. Press the **[OUTPT/START]** button. Variables are output from the memory of the **CNC**.

OUTPUT FORMAT

The output format is as follows:

N_V_

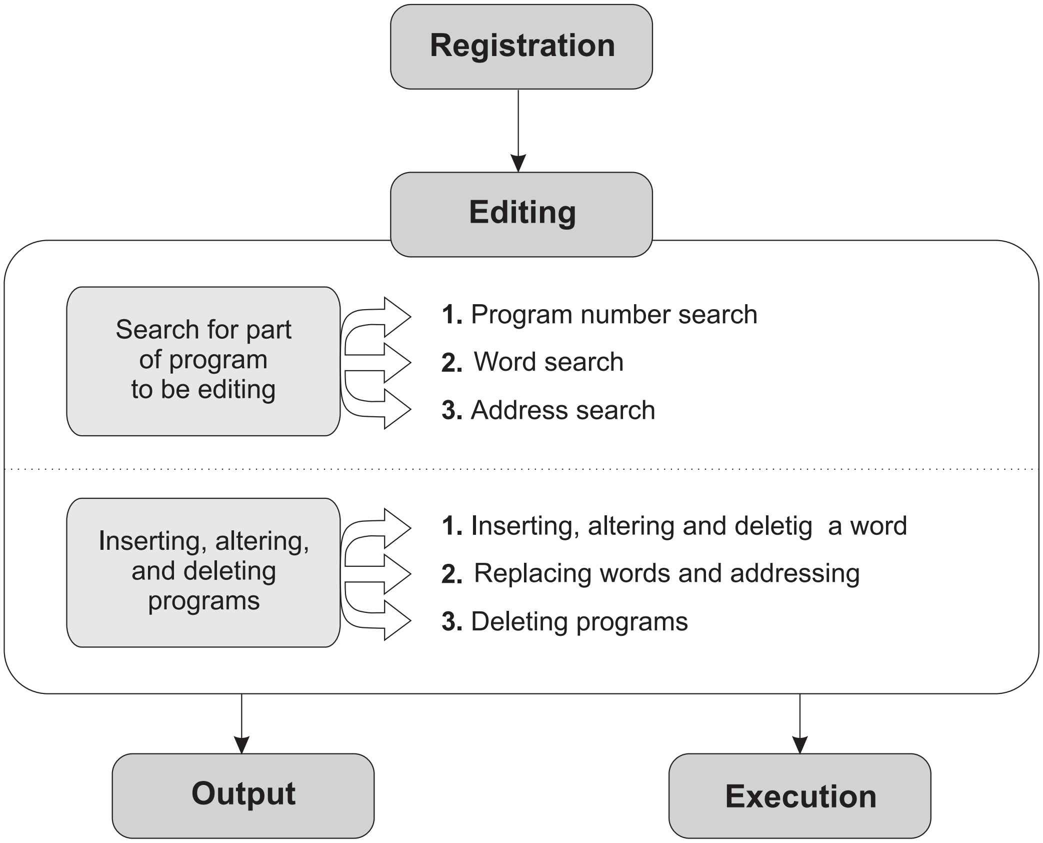
where:

N_: Variable number

V_: Variable setting value

9. EDITING PROGRAMS

This chapter describes how to edit programs registered in the **CNC** memory. Editing includes insertion, modification, deletion and replacement of words. Editing also includes deletion of the entire program. This chapter also describes program number search, sequence number search, word and address search - acts performed during editing a program.


Registration
Editing
Search for part of program to be editing
1. Program number search
2. Word search
3. Address search
Inserting, altering, and deleting programs
1. Inserting, altering and deletig a word
2. Replacing words and addressing
3. Deleting programs
Output
Execution

9.1 INSERTING, ALTERING AND DELETING A WORD

This section describes a procedure for inserting, modifying and deleting a word in a program registered in the **CNC** memory.

PROCEDURE FOR INSERTING, ALTERING AND DELETING WORDS

1. Select **EDIT** mode.

2. Press the function button **[PRGRM]** to display the program screen.

3. Select a program to be edited.
If the program to be edited is selected, perform operation **4**. If the program is not selected, search its program number.

4. Search the word to be modified.

- Scan method
- Word search method

5. Perform the operation altering, inserting or deleting the word.

◆ CONCEPT OF WORD AND EDITING UNIT

The word is an address followed by a number. The end of block symbol (**;**) is a word too. During editing, the word to be processed (editing unit) is highlighted. That is to be easily noticed the processed word.

◆ DATA INPUT

To insert or modify a word during editing, an address in entered by the corresponding data. The current input buffer is displayed at the bottom of the screen.

- after input is over, press the corresponding button to perform the desired edit/search function.
- **[DELET]** button deletes the last input symbol.
- **[CAN]** button cancels the input

WARNING:

The user cannot continue program execution after altering, inserting or deleting a word of the program during machine operation by operations such as single block stop or feed hold operation. If such a modification is made, the program may not be executed exactly according to the program displayed on the screen after machining is resumed. So, when the contents of the memory has to be modified, be sure to reset the system upon completion of editing, before executing the program.

9.1.1 Word search

A word can be found by moving the cursor throughout all the words in the text (scanning), by word search or by address search.

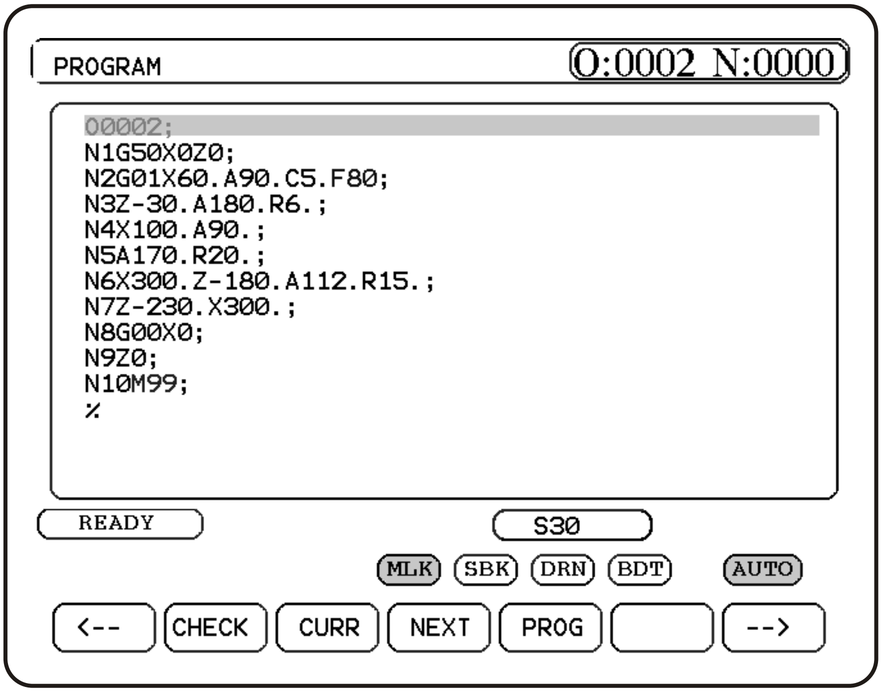
PROCEDURE FOR SCANNING A PROGRAM

1. Press the cursor key **[↓]**.
The cursor moves forward word by word on the screen. The cursor is positioned on the address of the selected word.

2. Press the cursor key **[↑]**.
The cursor moves backward word by word on the screen. The cursor is positioned on the address of the selected word.

Example:

When **Z04** is scanned.



```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

3. Holding down the cursor keys **[↓]** and **[↑]** scans consecutively all the words in the program.

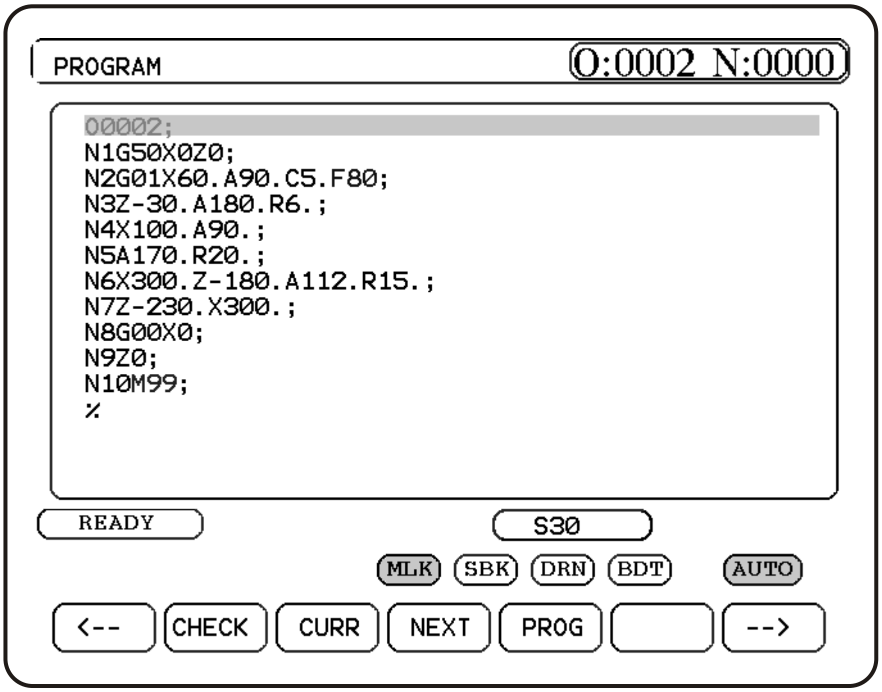
4. Pressing the page key [↓] displays the next page and searches for the first word of the page.

5. Pressing the page key [↑] displays the previous page and searches for the first word of the page.

PROCEDURE FOR SEARCHING A WORD

Example:

Searching for **S1000**



PROGRAM O:0002 N:0000

```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

READY

S30

1. Input address **[S]**.

2. Input **[1] [0] [0] [0]**.

**Notes:**

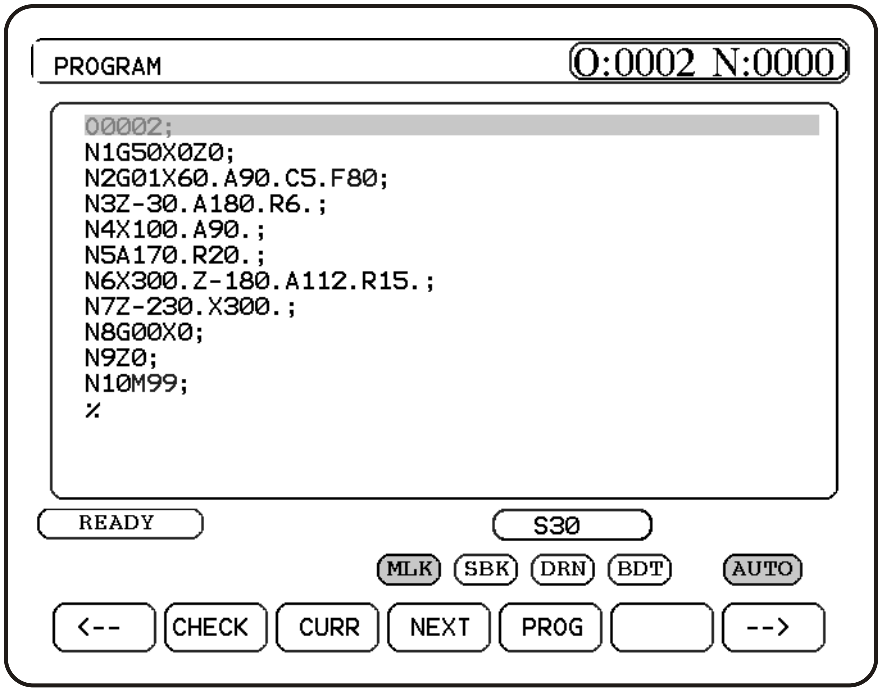
- ***S09** cannot be searched if only **S9** is input.*
- *To search for **S09**, input **S09**.*

3. Pressing the cursor key **[↓]** starts search operation.
Upon completion of the search operation, the cursor is positioned over **S1000**.
Pressing the cursor key **[↑]** rather than cursor key performs search operation in the reverse direction.

PROCEDURE FOR SEARCHING AN ADDRESS

Example:

Searching for **M28**



```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

1. Input address **[M]**.

2. Pressing the cursor key **[↓]** starts the search operation.

Upon completion of the search operation, the cursor is positioned over **M28**. Pressing the cursor key **[↑]** rather than cursor key performs search operation in the reverse direction.

Alarm number	Description
71	The word or address being searched was not found

9.1.2 Heading a program

The cursor can be jumped to the beginning of the program. This section describes two methods for that operation.

PROCEDURE FOR HEADING A PROGRAM

Press the **[RESET]** button when in **EDIT** mode. When the cursor returns to the beginning of the program, the contents of the program is displayed from the top of the screen.

9.1.3 Inserting a word

PROCEDURE FOR INSERTING A WORD

1. Find the word immediately before the word to be inserted.

2. Input the address to be inserted.

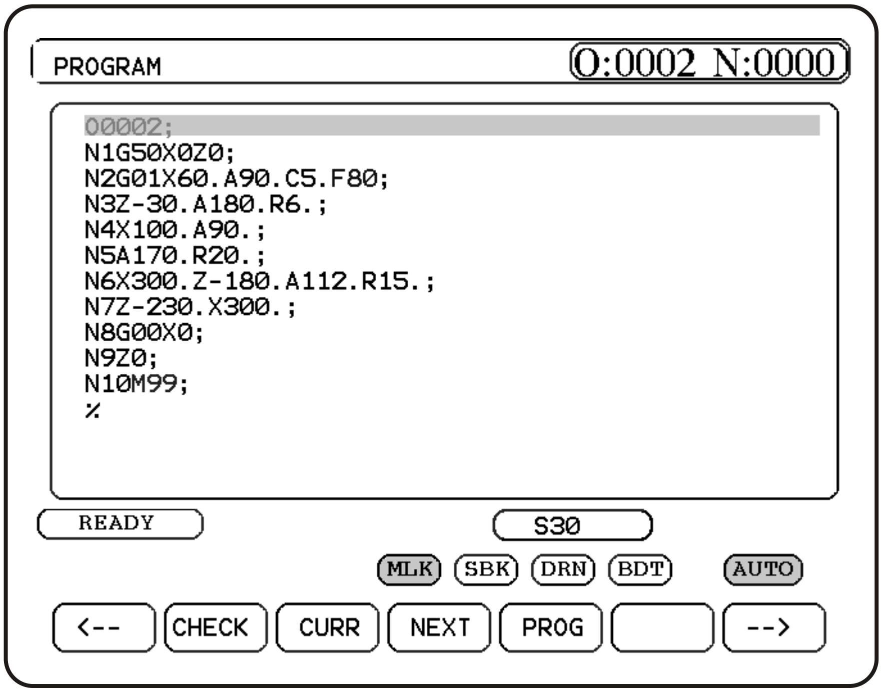
3. Input data.

4. Press the **[INSRT]** button.

Example:

Inserting **T0100**

1. Find **Z0**.

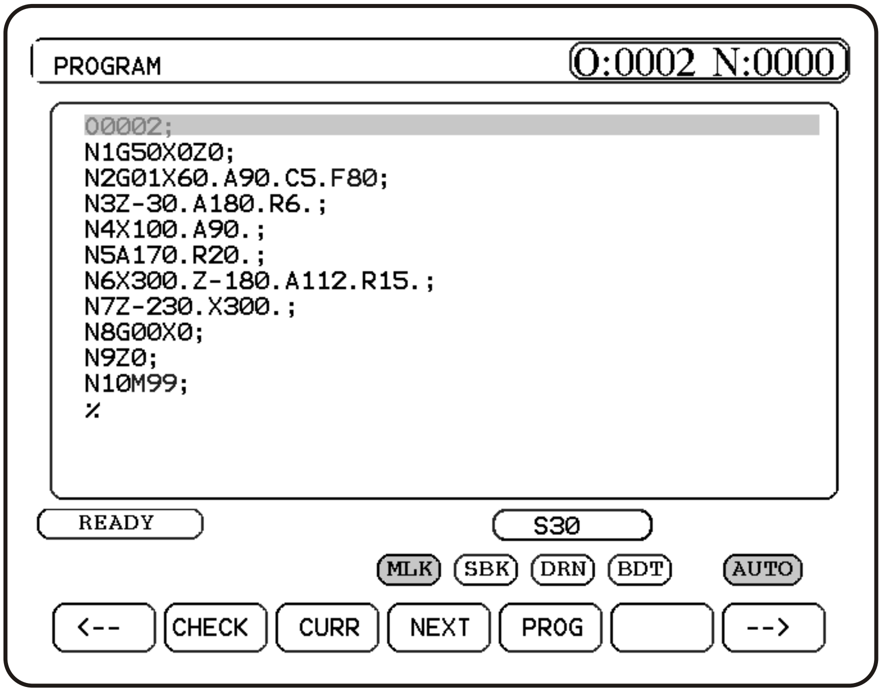


```
00002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

Z0 is found.

2. Input **[T] [0] [1] [0] [0]** .

3. Press the **[INSRT]** button.



```
00002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

T0100 is inserted.

9.1.4 Altering a word

PROCEDURE FOR ALTERING A WORD

1. Find the word to be altered.

2. Input the address.

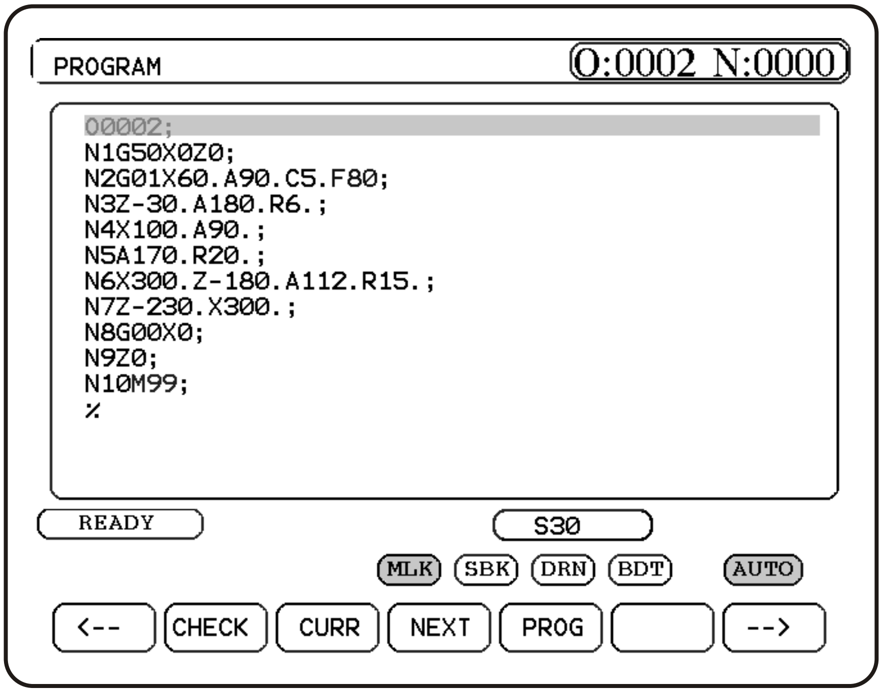
3. Input data.

4. Press the **[ALTER]** button.

Example:

Changing **T0100** to **M41**

1. Find **T0100**.

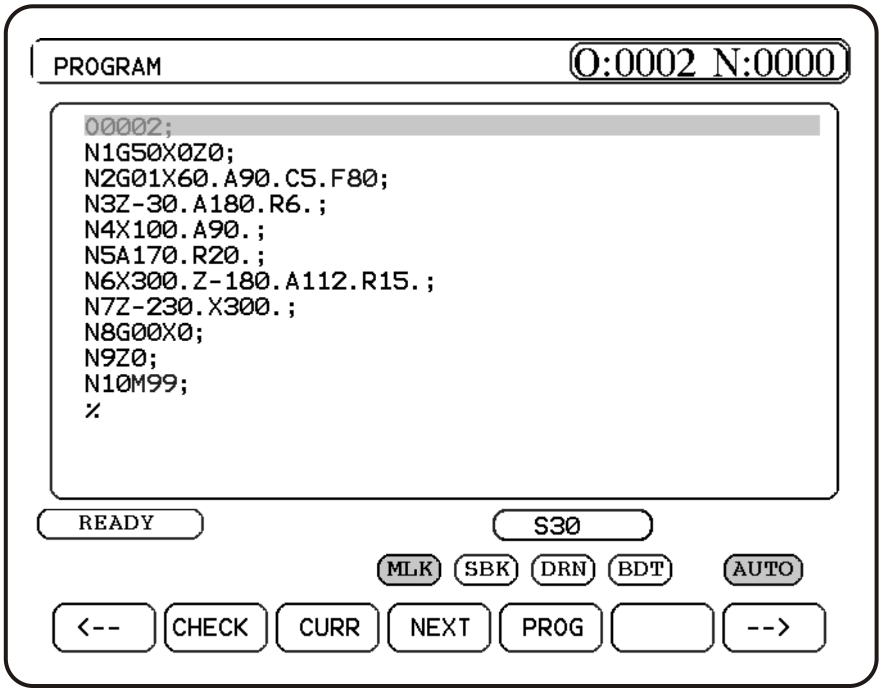


```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

T0100 has been found.

2. Input **[M] [4] [1]**.

3. Press the **[ALTER]** button.



```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

T0100 has been changed to **M41**.

9.1.5 Deleting a word

PROCEDURE FOR DELETING A WORD

1. Find the word to be deleted.

2. Press the **[DELET]** button twice.

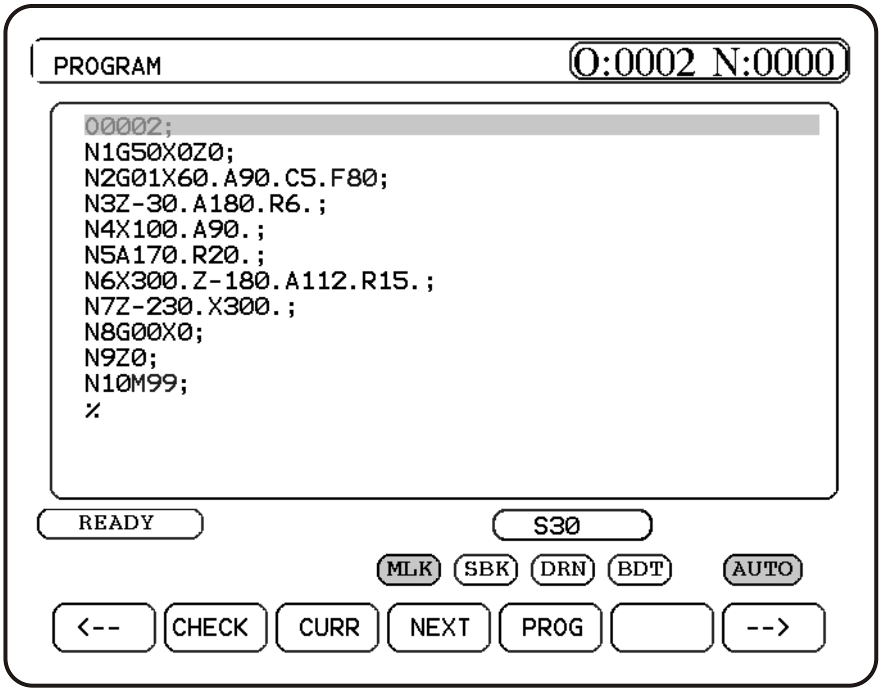
Note:

Pressing the button twice is for avoiding unexpected deleting. This function is active only when the input buffer is empty. Otherwise this button deletes the last symbol in the buffer.

Example:

Deleting **Z3.**

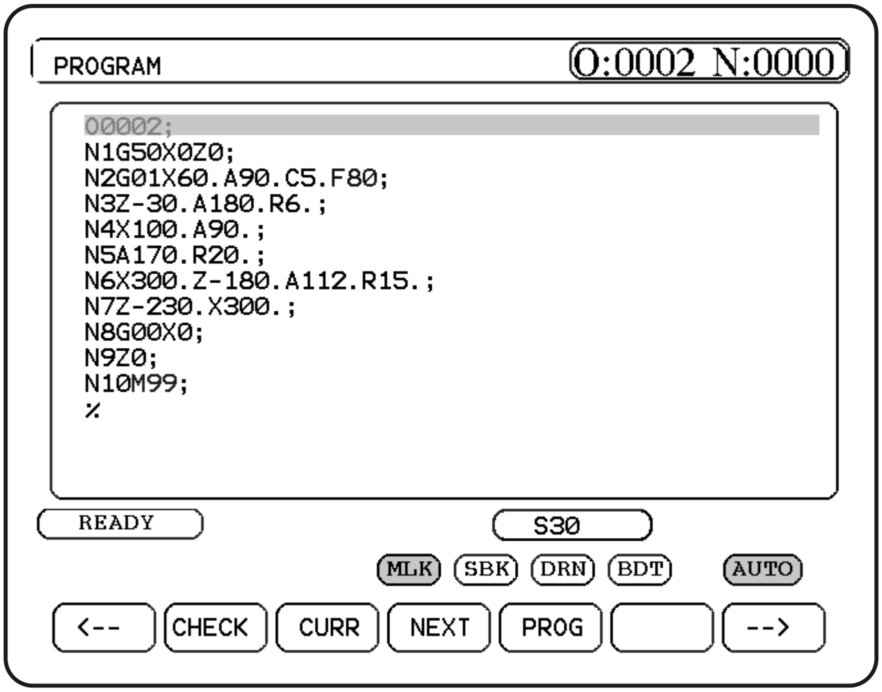
1. Find **Z3**.



```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

Z3 has been found.

2. Press the **[DELET]** button twice.



```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

Z3 is deleted.

9.2 PROGRAM NUMBER SEARCH

When the memory holds multiple programs, each of them can be found by its program number. There are two methods available for searching a program in the **CNC** memory.

PROCEDURE FOR PROGRAM NUMBER SEARCH

<u>METHOD 1</u>

1. Select **EDIT** mode.

2. Press the address button **[O]** and write search number.

3. Press the cursor move button **[↓]**.

4. Upon completion of search operation the cursor is positioned over the selected program. If such a program has not been found, an alarm is displayed.

<u>METHOD 2</u>

1. Select **EDIT** mode.

2. Press the functional button **[PRGRM]** and the soft button **[LIB]** to display the **PROGRAM LIBRARY** screen.

3. Use the cursor move buttons to view all the programs in the **CNC** memory.

ALARM

Alarm number	Description
71	The searched program has not been found

9.3 DELETING PROGRAMS

Programs registered in the **CNC** memory can be deleted either one by one or all at once.

9.3.1 Deleting one program

The system offers a function for deleting only one program in the **CNC** memory.

PROCEDURE FOR DELETING ONE PROGRAM

1. Select **EDIT** mode.

2. Press the functional button **[PRGRM]** and the soft button **[LIB]** to display the **PROGRAM LIBRARY** screen.

3. Insert "0" and the program's number.

4. Press the **[DELET]** button.
The program with the specified number is deleted.

9.3.2 Deleting all programs

The system offers a function for deleting all the programs in the **CNC** memory.

- Select **EDIT** mode.

- Select a display **[PROGRAM]**.

- Insert 0 - 9999 and press the button **"DELETE"**.

- All programs will be deleted.

10. CREATING PROGRAMS

Programs can be input from the MDI panel. This chapter describes the method for creating programs.

10.1 CREATING PROGRAMS USING THE MDI PANEL

Programs can be created in EDIT mode, using the EDIT functions described in chapter **9**.

PROCEDURE FOR CREATING PROGRAMS USING THE MDI PANEL

1. Select EDIT mode.

2. Press the functional button **[PRGRM]** and the soft button **[LIB]** to display the PROGRAM LIBRARY screen.

3. Input **"0"** and the new program number (up to 4 digits).

4. Press the **[INSRT]** button.
   The program has been created.

ALARM

Alarm number	Description
73	The program number is already used

11. SETTING AND DISPLAYING DATA

GENERAL

To operate a **CNC** machine tool, various data must be set from the TFT/MDI panel. The operator can monitor the state of the operation with the data displayed during operation.

This chapter describes how to display and set data for each functional. This chapter describes procedures for selecting the information desired by the soft keys.

◆ SCREEN TRANSITION CHART

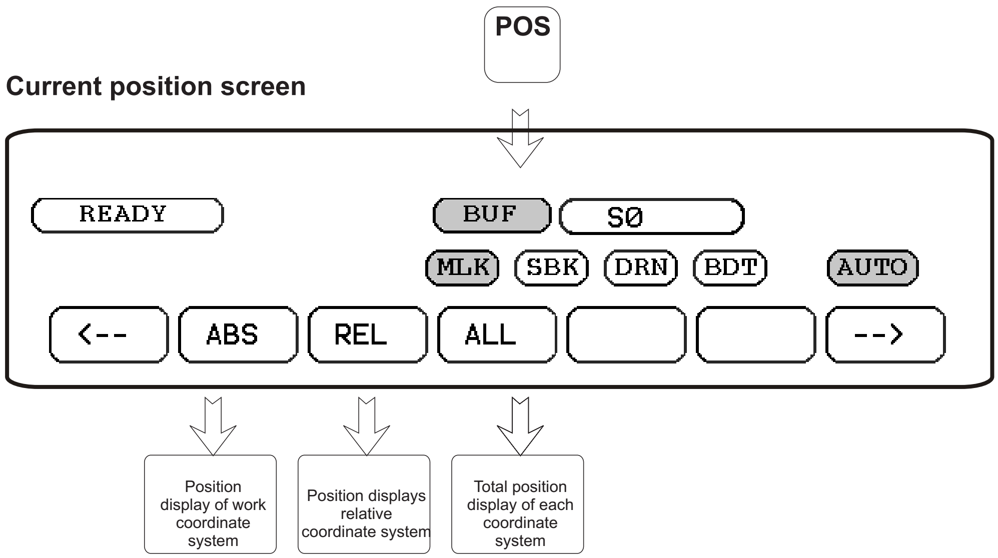
The screen transition for when each functional key on the **MDI** panel is pressed is shown below. The subsections referenced for each screen are also shown. For details on each screen and the setting procedures, see the appropriate subsections.

◆ DATA PROTECTION KEY

The machine may have a data protection key to protect programs. Refer to the manual issued by the machine tool builder for where the data protection key is located and how to use it.

POSITION DISPLAY SCREEN

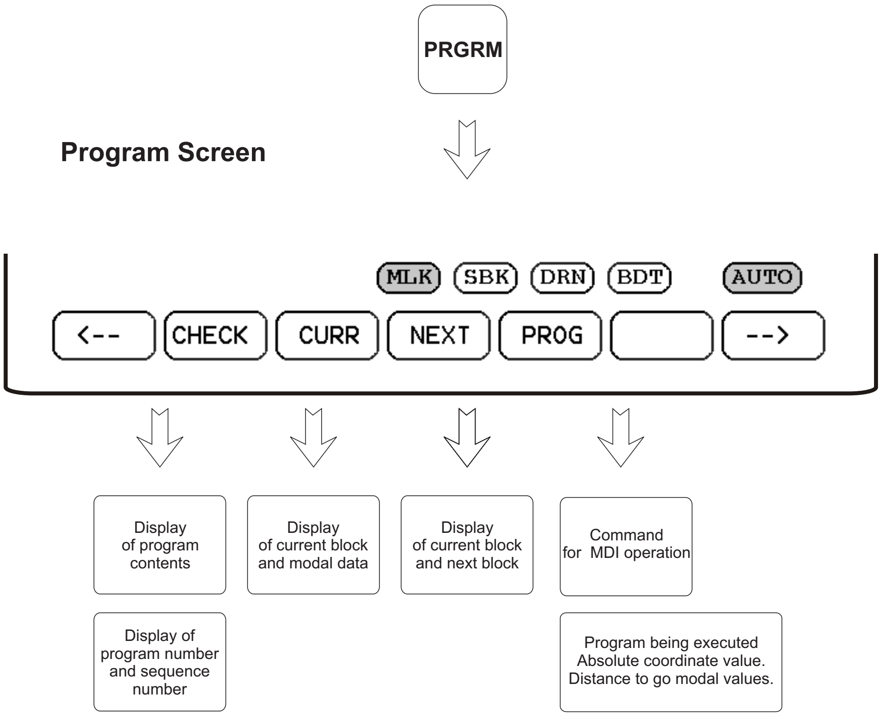
Screen transition triggered by the functional key **[POS]**



POS
Current position screen
READY
BUF
S0
MLK
SBK
DRN
BDT
AUTO
<--
ABS
REL
ALL
-->
Position display of work coordinate system
Position displays relative coordinate system
Total position display of each coordinate system

PROGRAM SCREEN

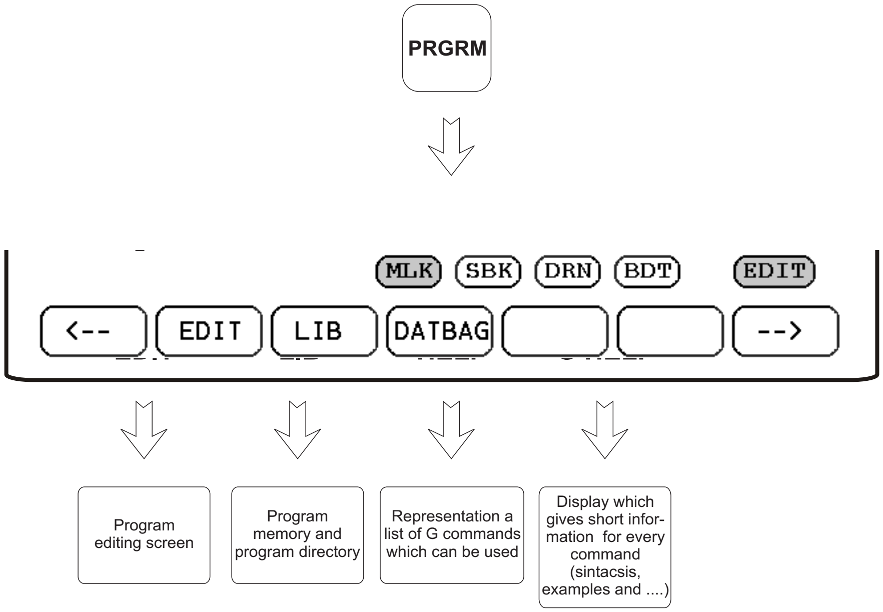
Screen transition triggered by the functional key **[PRGRM]** in **AUTO** or **MDI** mode



PRGRM
Program Screen
MLK
SBK
DRN
BDT
AUTO
<--
CHECK
CURR
NEXT
PROG
-->
Display of program contents
Display of current block and modal data
Display of current block and next block
Command for MDI operation
Display of program number and sequence number
Program being executed Absolute coordinate value. Distance to go modal values.

PROGRAM SCREEN

Screen transition triggered by the functional key **[PRGRM]** in **EDIT** mode



Program editing screen

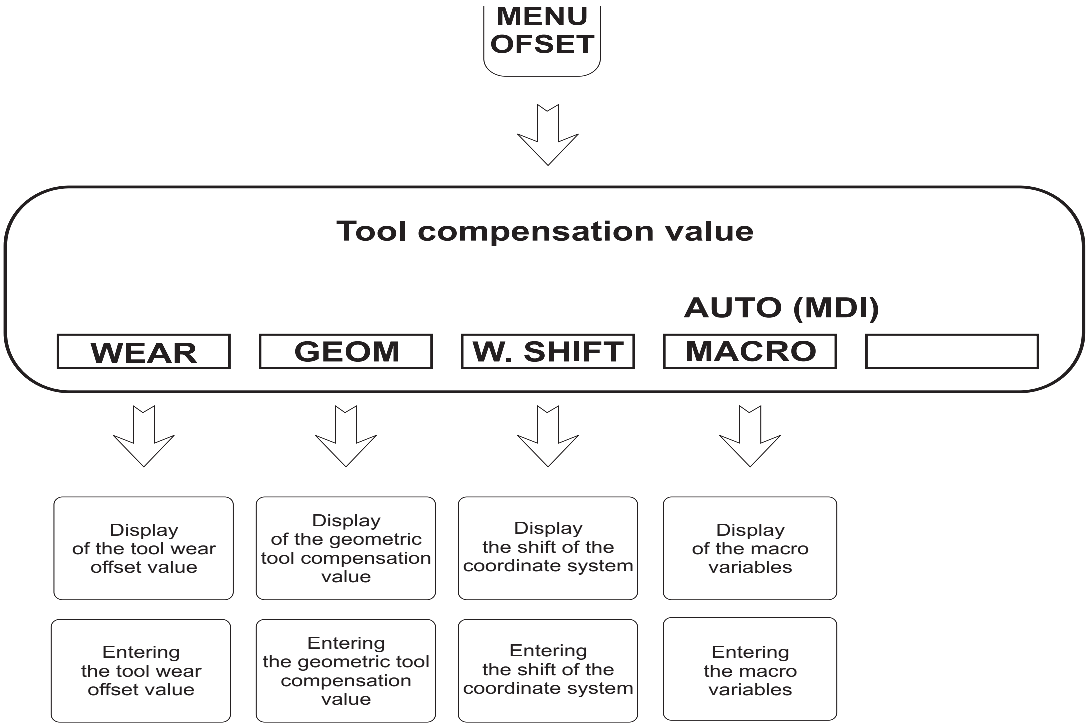
Program memory and program directory

Representation a list of G commands which can be used

Display which gives short information for every command (sintacsis, examples and ....)

OFFSET SCREEN

Screen transition triggered by the functional key **[MENU/OFSET]**



Display of the tool wear offset value

Display of the geometric tool compensation value

Display the shift of the coordinate system

Display of the macro variables

Entering the tool wear offset value

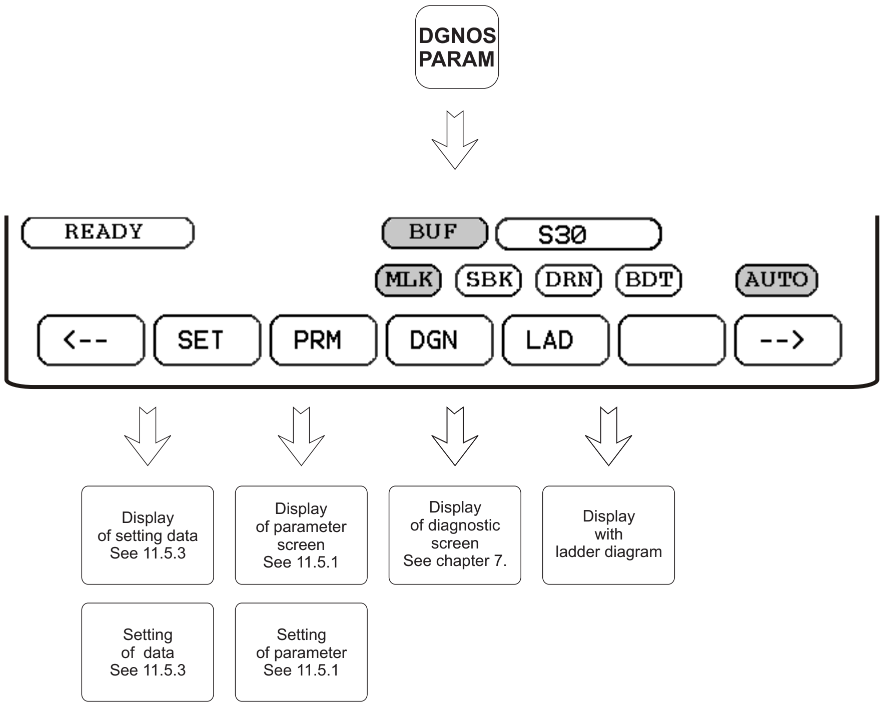
Entering the geometric tool compensation value

Entering the shift of the coordinate system

Entering the macro variables

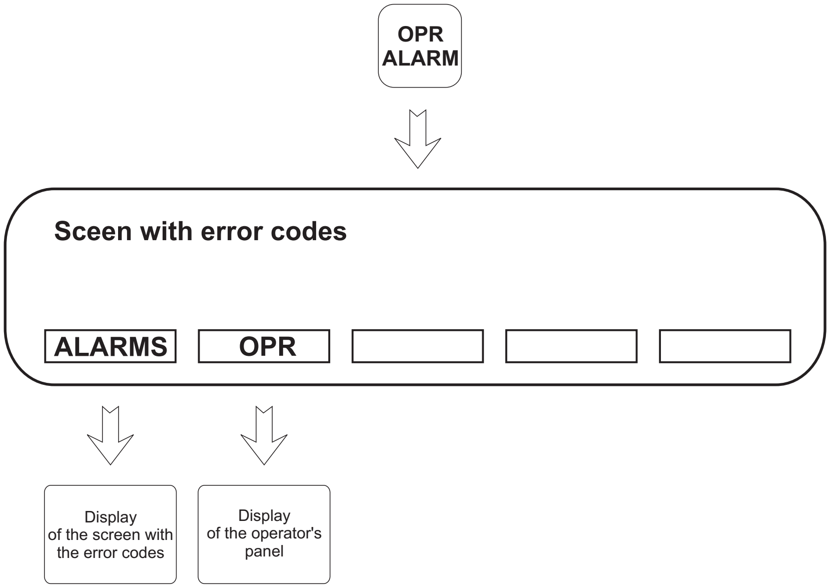
PARAMETER/DIAGNOSTIC SCREEN

Screen transition triggered by the functional key **[DGNOS/PARAM]**


DGNOS
PARAM
READY
BUF
S30
MLK
SBK
DRN
BDT
AUTO
<--
SET
PRM
DGN
LAD
-->
Display of setting data See 11.5.3
Display of parameter screen See 11.5.1
Display of diagnostic screen See chapter 7.
Display with ladder diagram
Setting of data See 11.5.3
Setting of parameter See 11.5.1

ALARM SCREEN

Screen transition triggered by the functional key **[OPR/ALARM]**


OPR
ALARM
Sceen with error codes
ALARMS
OPR
Display of the screen with the error codes
Display of the operator's panel

◆ SETTING SCREENS

The table below lists the data set on each screen.

Setting screens and data on them

№	Setting screen	Contents of setting	Remark
1	Tool offset value	Tool length offset value Cutter compensation value	11.4.1
2	Setting data	Setting data	11.5.3
3	Macro variables	Custom macro common variables (#100 до #132) или (#500 до#531)	11.4.3
4	Parameters	Parameters	11.5.1
		Pitch error compensation data	11.5.2
5	Software operator's panel	Additional operators switches	11.6.2
6	Work coordinate setting	Work origin offset value	11.4.2

11.1 SCREENS DISPLAYED BY FUNCTIONAL KEY [POS]

Press the functional key **[POS]** to display the current position of the tool. The following three screens are used to display the current position of the tool.

- **Position display screen for the work coordinate system**
- **Position display screen for the relative coordinate system**
- **Overall position display screen**

The above screens can also display the feedrate, run time and the number of parts.

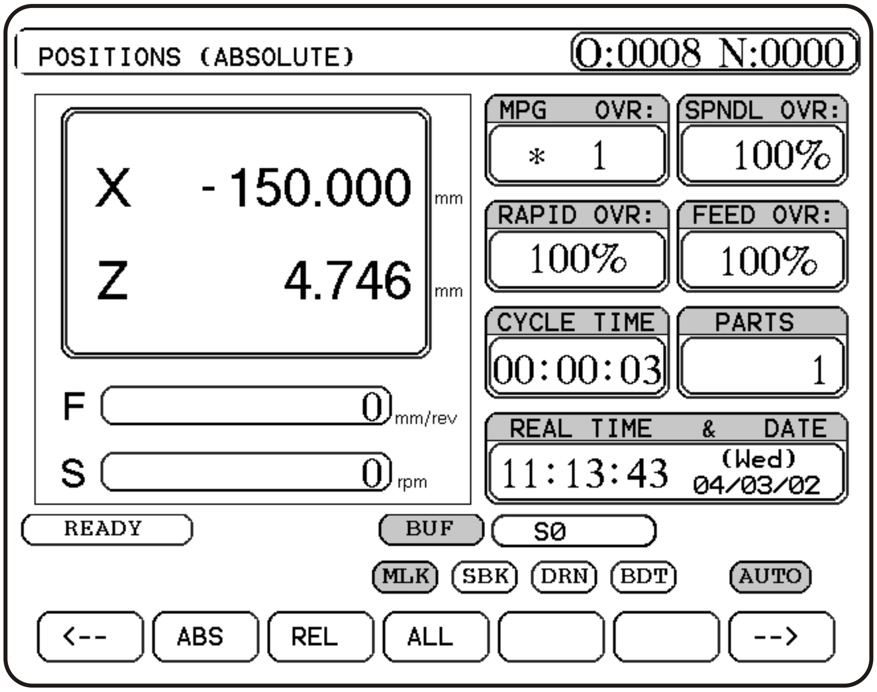
11.1.1 Position display in the work coordinate system

Displays the current position of the tool in the workpiece coordinate system. The current position changes as the tool moves. The title at the top of the screen indicates that absolute coordinates are used.

PROCEDURE FOR DISPLAY THE CURRENT POSITION SCREEN IN THE WORKPIECE COORDINATE SYSTEM

1. Press the functional button **[POS]**.

2. Press the soft button **[ABS]**.



POSITIONS (ABSOLUTE) O:0008 N:0000

X -150.000 mm

Z 4.746 mm

F 0 mm/rev

S 0 rpm

MPG OVR: * 1

SPNDL OVR: 100%

RAPID OVR: 100%

FEED OVR: 100%

CYCLE TIME 00:00:03

PARTS 1

REAL TIME & DATE 11:13:43 (Wed) 04/03/02

READY

BUF

S0

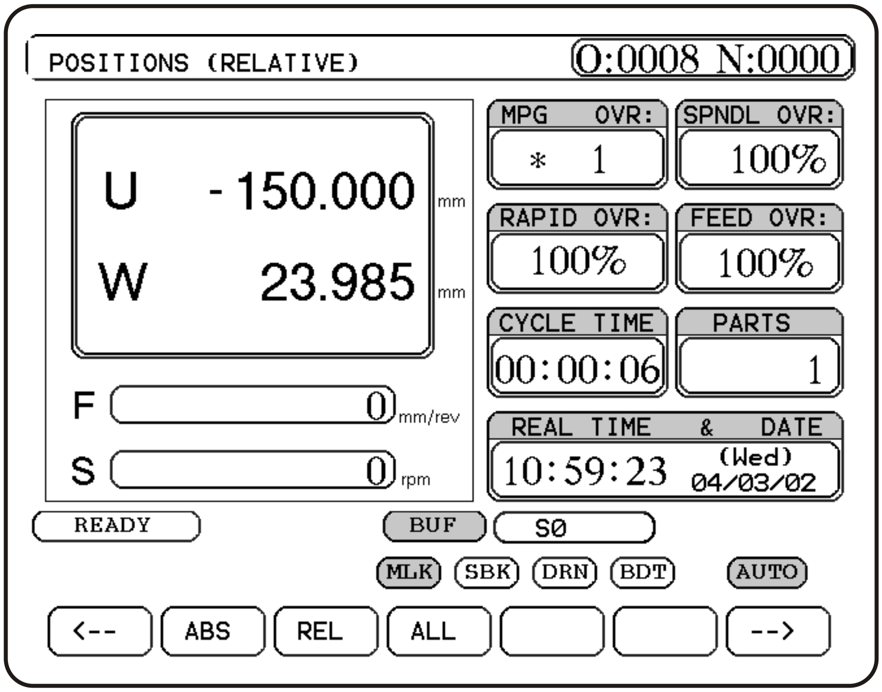
11.1.2 Position display in the relative coordinate system

Displays the current position of the tool in the relative coordinate system based on the coordinates set by the operator. The current position changes as the tool moves. The title at the top of the screen indicates that relative coordinates are used.

PROCEDURE FOR DISPLAY THE CURRENT POSITION SCREEN IN THE RELATIVE COORDINATE SYSTEM

1. Press the functional button **[POS]**.

2. Press the soft button **[REL]**.



POSITIONS (RELATIVE) O:0008 N:0000

U -150.000 mm

W 23.985 mm

F 0 mm/rev

S 0 rpm

MPG OVR: * 1

SPNDL OVR: 100%

RAPID OVR: 100%

FEED OVR: 100%

CYCLE TIME 00:00:06

PARTS 1

REAL TIME & DATE 10:59:23 (Wed) 04/03/02

READY

BUF S0

◆ SETTING RELATIVE COORDINATES

The current position of the tool in the relative coordinate system can be reset to **0** as follows:

PROCEDURE TO RESET THE AXIS COORDINATES ALONG A SELECTED AXIS

1. Input the address of the axis name (**Z, X**, etc.) in the relative coordinate screen.
The entered axis address starts to blink. Two or more axis names can be input simultaneously.

2. Press the **[CAN]** button. The relative coordinates of the axes selected above are reset.

Note:

Pressing the address of the axis name once again cancels it.

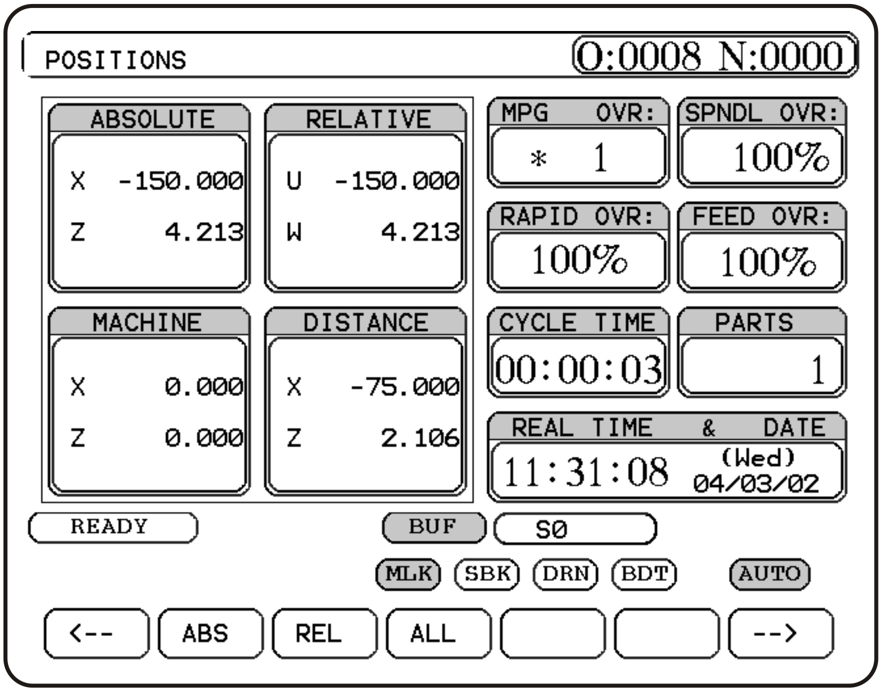
11.1.3 Overall position display

The screen displays the following positions: current position of the tool in the workpiece coordinate system, relative coordinate system and machine coordinate system and the remaining distance to the end of the operation.

PROCEDURE FOR DISPLAYING OVERALL POSITION DISPLAY SCREEN

1. Press the functional key **[POS]**.

2. Press the soft key **[ALL]**.



POSITIONS O:0008 N:0000

ABSOLUTE		RELATIVE	
X	-150.000	U	-150.000
Z	4.213	W	4.213

MACHINE		DISTANCE	
X	0.000	X	-75.000
Z	0.000	Z	2.106

MPG OVR: * 1

SPNDL OVR: 100%

RAPID OVR: 100%

FEED OVR: 100%

CYCLE TIME 00:00:03

PARTS 1

REAL TIME & DATE 11:31:08 (Wed) 04/03/02

READY

BUF

S0

◆ COORDINATE DISPLAY

The current positions of the tool in the following coordinate systems are displayed at the same time:

- Current position in the work coordinate system (absolute coordinates).

- Current position in the relative coordinate system (relative coordinates).

- Current position in the machine coordinate system (machine coordinates).

- Distance to go.

◆ DISTANCE TO GO

The remaining distance is displayed in **AUTO** or **MDI** mode. The distance the tool should be moved in the current block is displayed.

◆ MACHINE COORDINATE SYSTEM

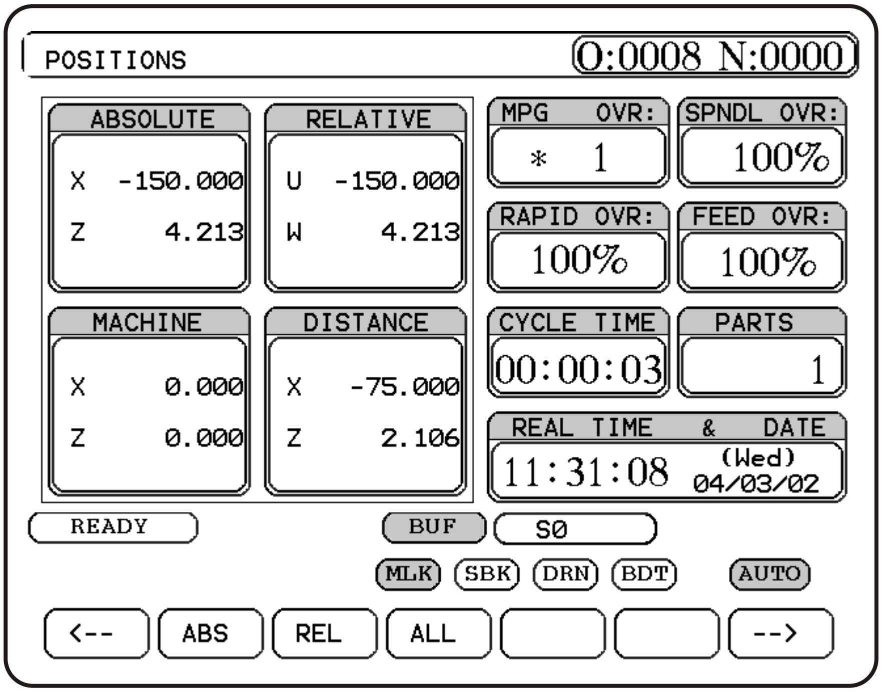
The least command increment is used as the unit for values displayed in the machine coordinate system.

11.1.4 Display of run time and parts count

The run time, cycle time and the number of machined parts are displayed on the current position display screen.

PROCEDURE FOR DISPLAYING RUN TIME AND PARTS COUNT ON THE CURRENT POSITION DISPLAY SCREEN

1. Press the functional key **[POS]** to display the current position display screen.



POSITIONS O:0008 N:0000

ABSOLUTE
X -150.000
Z 4.213

RELATIVE
U -150.000
W 4.213

MACHINE
X 0.000
Z 0.000

DISTANCE
X -75.000
Z 2.106

MPG OVR: * 1

SPNDL OVR: 100%

RAPID OVR: 100%

FEED OVR: 100%

CYCLE TIME 00:00:03

PARTS 1

REAL TIME & DATE 11:31:08 (Wed) 04/03/02

READY

BUF S0

The number of machined parts **(PARTS)**, real time **(REAL TIME)** and the cycle time **(CYCLE TIME)** are displayed under the current position.

◆ PART COUNT

Indicates the number of machined parts. The number is incremented each time **M02** or **M30** command is executed. Press the address key **[P]** and then **[CAN]** to reset the counter.

◆ REAL TIME

Displays the real time and date.

◆ CYCLE TIME

Indicates the run time of one automatic operation, excluding the stop and feed hold time. The value is automatically reset when a cycle start is performed.

11.2 SCREENS DISPLAYED BY FUNCTIONAL KEY [PRGRM] (IN AUTO OR MDI MODE)

This chapter describes the screens displayed by pressing the functional key **[PRGRM]** in **AUTO** or **MDI** mode. The first four of the following screens display the execution state of the current executed program in **AUTO** or **MDI** mode and the last screen displays the command values for **MDI** operation in **MDI** mode.

Program contents display screen

Current block display screen

Next block display screen

Program check screen

Program screen for MDI operation

11.2.1 Program contents display

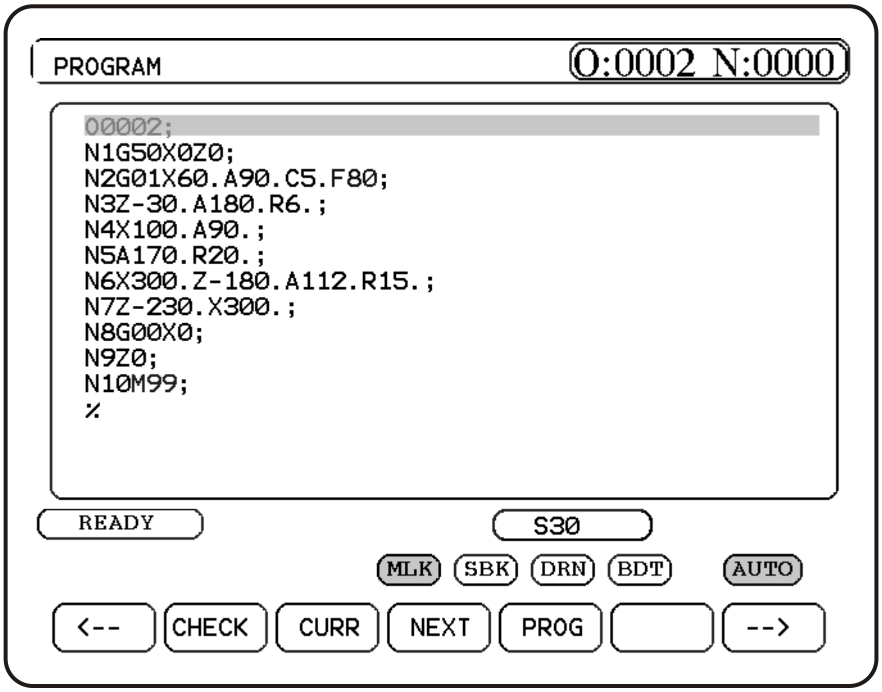
Displays the current executed program in AUTO mode.

PROCEDURE FOR DISPLAYING THE PROGRAM CONTENTS

1. Press the functional button **[PROG]** to display the program.

2. Press the soft button **[PROG]**.

The cursor is positioned over the current executed block.



PROGRAM O:0002 N:0000

```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

READY

S30

11.2.2 Current block display screen

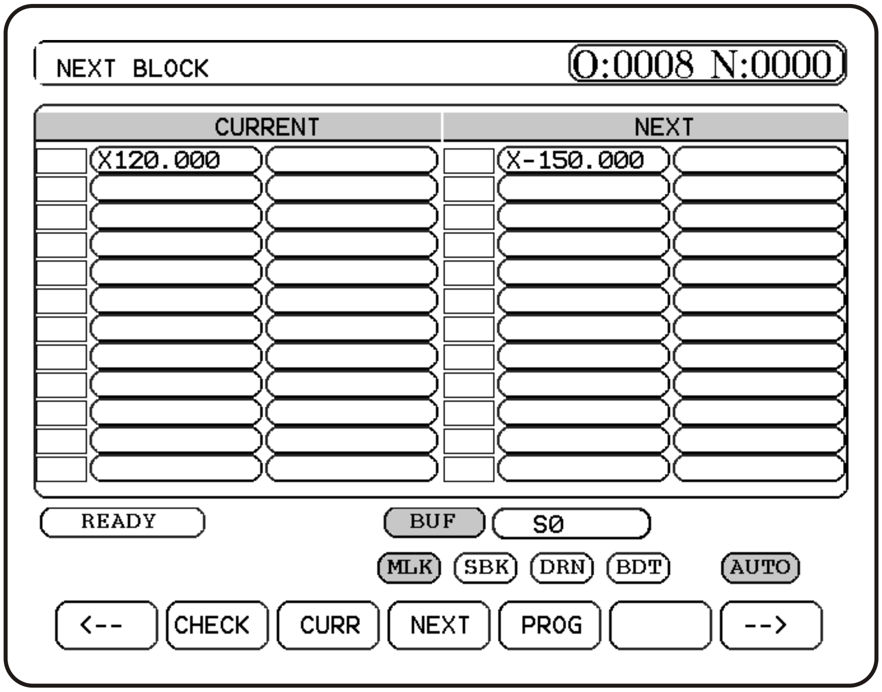
Displays the current executed block and the modal data in **AUTO** or **MDI** mode.

PROCEDURE FOR DISPLAYING THE CURRENT BLOCK DISPLAY SCREEN

1. Press the functional button **[PRGRM]** to display the program.

2. Press the soft key **[CURR]**.

The current executed block and modal data is displayed.



NEXT BLOCK O:0008 N:0000

CURRENT	NEXT
X120.000	X-150.000

11.2.3 Next block display screen

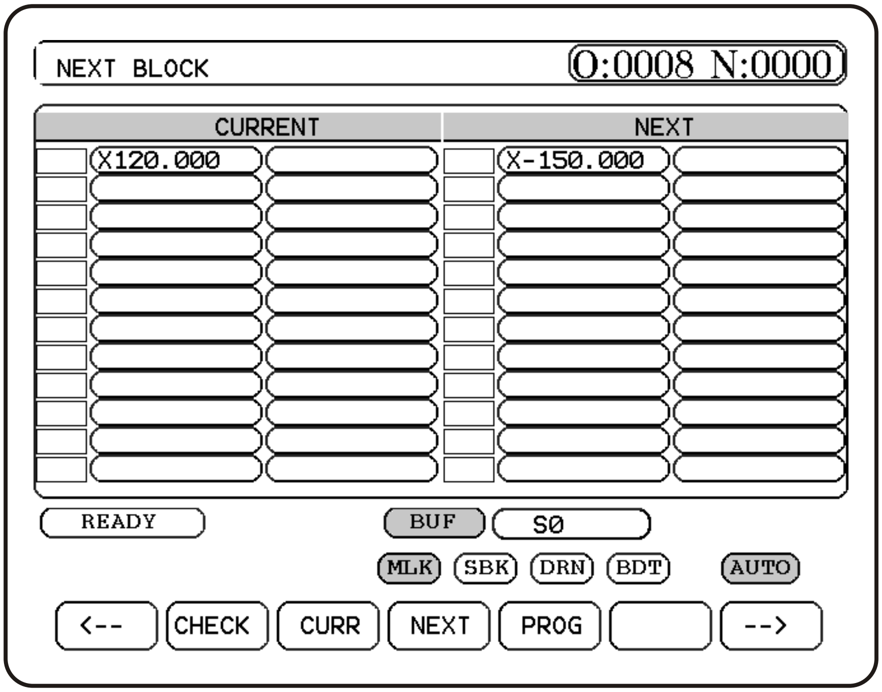
Displays the current executed block and the block to be executed next.

PROCEDURE FOR DISPLAYING THE NEXT BLOCK DISPLAY SCREEN

1. Press the functional button **[PRGRM]** to display the program.

2. Press the soft key **[NEXT]**.

The current executed block and the block to be executed next are displayed.



NEXT BLOCK O:0008 N:0000

CURRENT			NEXT		
	X120.000			X-150.000	

READY BUF S0

11.2.4 Program check screen

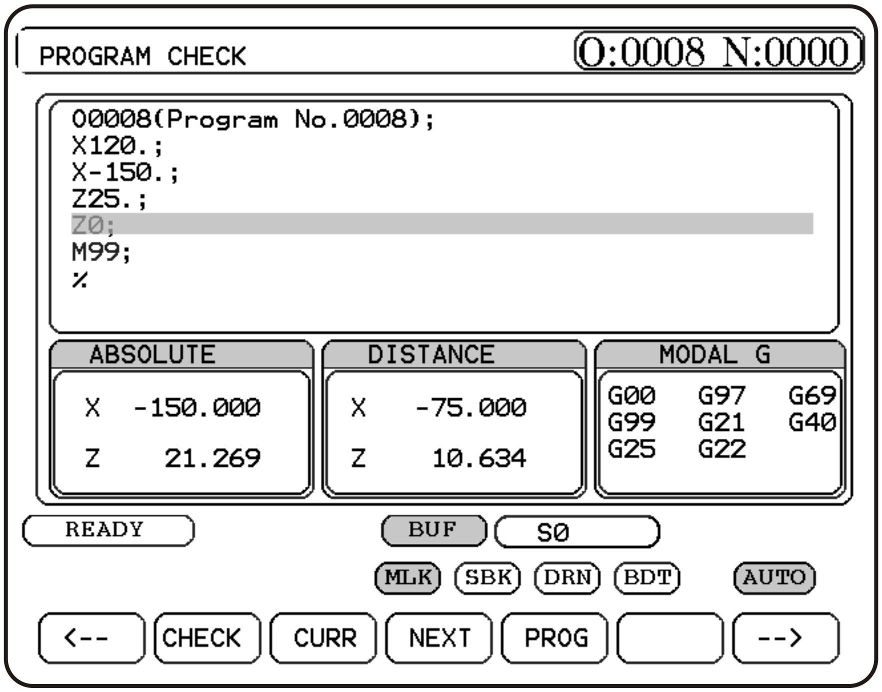
Displays the current executed program, current position of the tool and the modal data in **AUTO** mode.

PROCEDURE FOR DISPLAYING THE PROGRAM CHECK SCREEN

1. Press the functional button **[PRGRM]** to display the program.

2. Press the soft button **[CHECK]**.

The current executed program, current position of the tool and the modal data are displayed.



PROGRAM CHECK O:0008 N:0000

```
O0008(Program No.0008);
X120.;
X-150.;
Z25.;
Z0;
M99;
%
```

ABSOLUTE		DISTANCE		MODAL G		
X	-150.000	X	-75.000	G00	G97	G69
				G99	G21	G40
Z	21.269	Z	10.634	G25	G22	

◆ PROGRAM DISPLAY

For the current executed program, the block currently being executed is displayed first.

◆ **CURRENT POSITION DISPLAY**

The tool position in the workpiece coordinate system or in the relative coordinate system and the remaining distance to the end of the operation are displayed. The absolute and relative positions are switched by a parameter.

11.2.5 Program screen for MDI operation

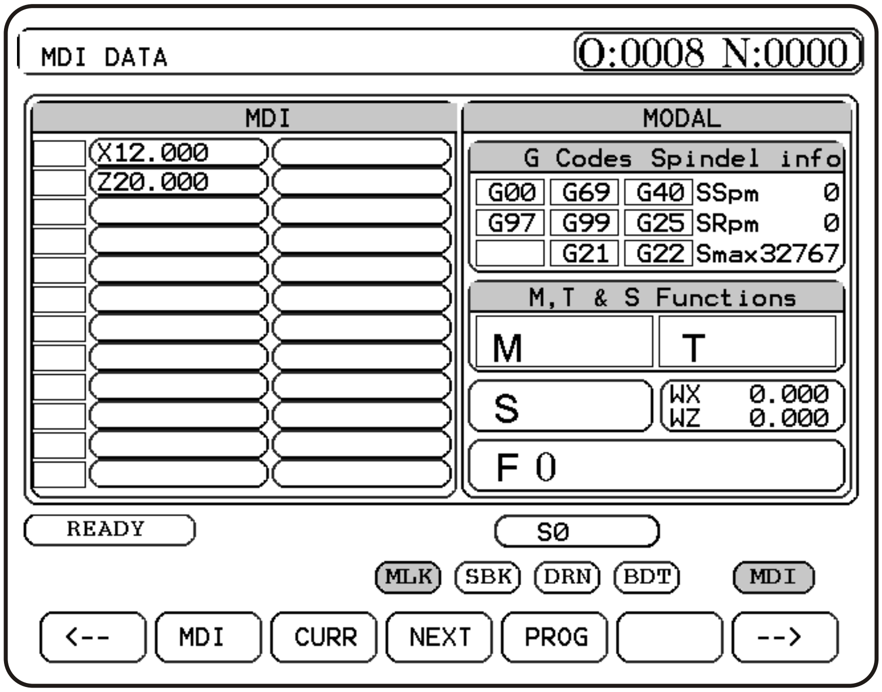
Displays the program input from the MDI panel and the modal data in MDI mode.

PROCEDURE FOR DISPLAYING THE PROGRAM SCREEN FOR MDI OPERATION

1. Press the functional button **[PRGRM]** to display the program.

2. Press the soft button **[MDI]**.

The program input from the **MDI** panel and the modal data are displayed.



MDI DATA O:0008 N:0000

MDI

X12.000
Z20.000

MODAL

G Codes Spindel info

G00	G69	G40	SSpm 0
G97	G99	G25	SRpm 0
	G21	G22	Smax32767

M,T & S Functions

M

T

S

WX 0.000
WZ 0.000

F 0

READY

S0

11.3 SCREENS DISPLAYED BY FUNCTIONAL KEY [PRGRM] (IN EDIT MODE)

This section describes the screens displayed by pressing the functional key **[PRGRM]** in **EDIT** mode. The functional button **[PRGRM]** in **EDIT** mode can display the program editing screen and the library screen (displays used memory and a list of the programs).

11.3.1 Displaying used memory and the list of programs

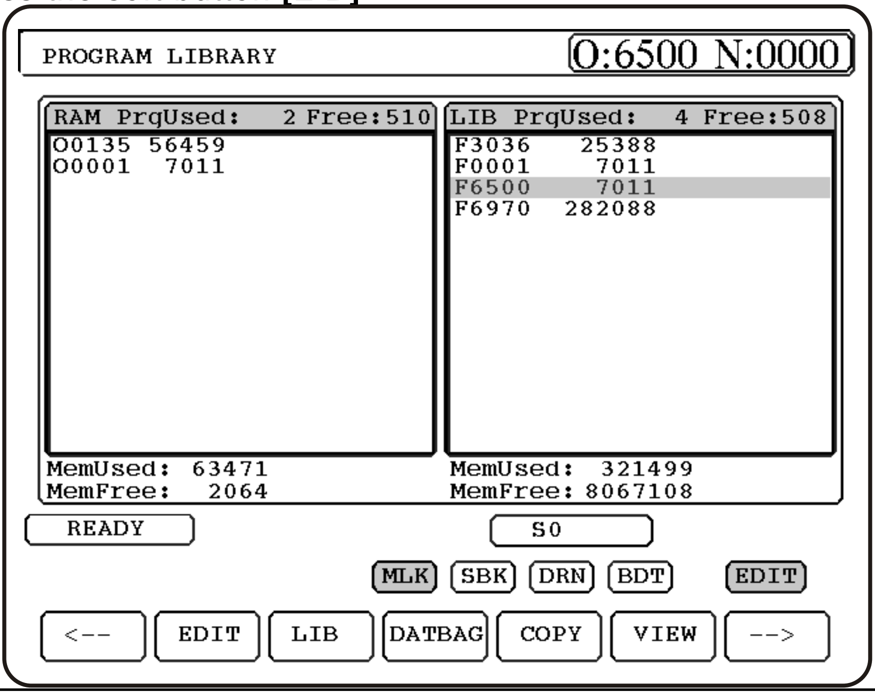
Displays the number of registered programs, used memory and a list of registered programs.

PROCEDURE FOR DISPLAYING USED MEMORY AND THE LIST OF PROGRAMS

1. Select **EDIT** mode.

2. Press the functional button **[PRGRM]**.

3. Press the soft button **[LIB]**.



```
PROGRAM LIBRARY                          O:6500 N:0000

RAM PrgUsed:   2 Free:510    LIB PrgUsed:   4 Free:508
O0135  56459                 F3036    25388
O0001   7011                 F0001     7011
                             F6500     7011
                             F6970   282088

MemUsed:  63471              MemUsed:  321499
MemFree:   2064              MemFree: 8067108
```

◆ USED MEMORY

Program numbers used

Program numbers used(16): The number of registered programs (including subprograms).

Free(496): Number of programs that can be additionally registered.

Used memory area

Used memory area(7777): Used data memory (indicated in number of characters).

Free(57758): Memory that can be additionally used for storing data (indicated in number of characters).

11.3.3 Displaying a short description of a concrete G cod.

PROCEDURE FOR DISPLAYING A CONCRETE G COD.

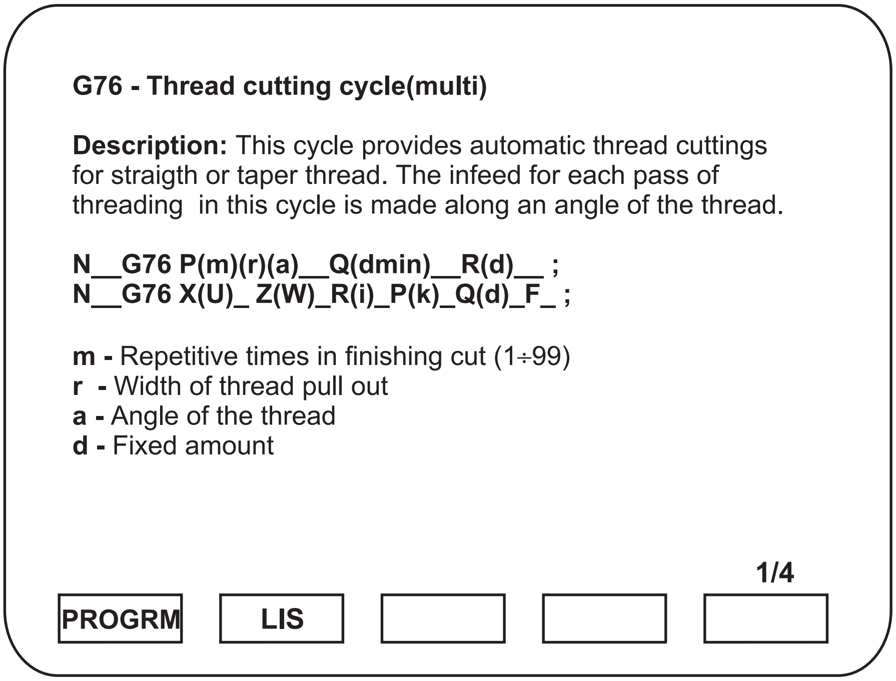
1. Select **EDIT** mode.

2. Press the functional button **[PRGRM]**.

3. Insert **G76**.

4. Press the soft button **[G HELP]**.

Displaying the below display:



G76 - Thread cutting cycle(multi)

Description: This cycle provides automatic thread cuttings for straigth or taper thread. The infeed for each pass of threading in this cycle is made along an angle of the thread.

N__G76 P(m)(r)(a)__Q(dmin)__R(d)__ ;
N__G76 X(U)_ Z(W)_R(i)_P(k)_Q(d)_F_ ;

m - Repetitive times in finishing cut (1÷99)
r - Width of thread pull out
a - Angle of the thread
d - Fixed amount

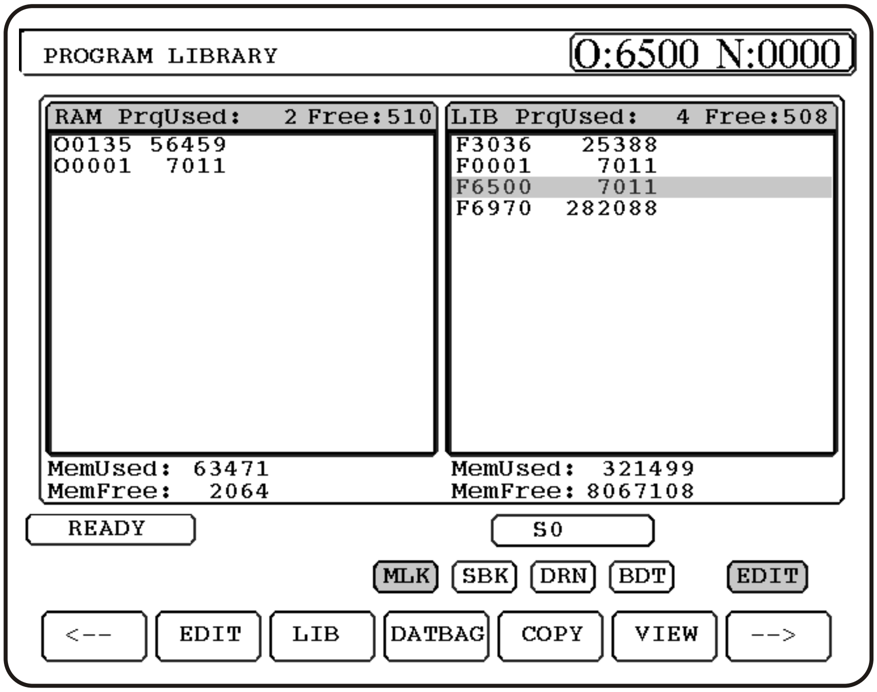
1/4

5. With the keys **[Pg Up]** and **[Pg Dn]** you can look at the all description for **G76**.

With this functional you can get information about all codes which can be used from a **CNC07T**. Here you can find a short describtion on every **G** cod, the sintacsis of the code, parameters' description, an example program and graphic visualization of the tool path.

◆ PROGRAM LIBRARY LIST

Registered programs and their numbers are displayed. The cursor is over the current program.



```
PROGRAM LIBRARY                                  O:6500 N:0000

RAM PrgUsed:   2 Free:510      LIB PrgUsed:   4 Free:508
O0135  56459                   F3036    25388
O0001   7011                   F0001     7011
                               F6500     7011
                               F6970   282088

MemUsed:  63471                MemUsed:  321499
MemFree:   2064                MemFree: 8067108
```

11.4 SCREENS DISPLAYED BY FUNCTIONAL KEY [MENU] [OFSET]

Press the functional button **[OFSET]** to display the tool compensation values. Other data is also displayed.

This chapter describes how to display and set the following data:

Tool offset value.

Tool geometric offset value.

Workpiece origin offset value.

Custom macro common variables.

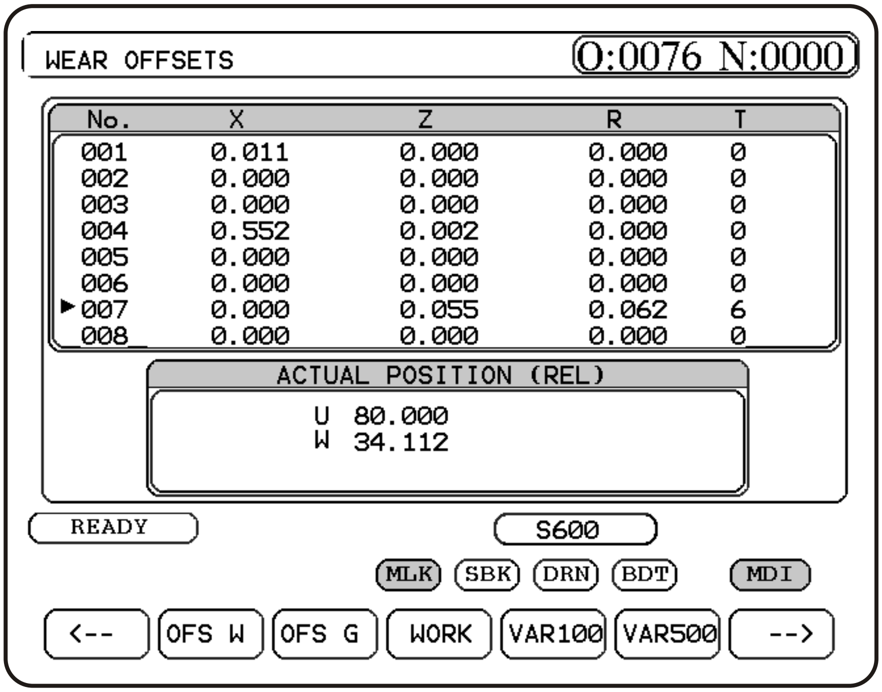
11.4.1 Setting and displaying the tool offset value

Tool length offset value and tool radius compensation value are specified by **T** code in the program. These codes can be displayed and set on the screen.

PROCEDURE FOR SETTING AND DISPLAYING THE CUTTER COMPENSATION VALUE

1. Press the functional button **[OFSET]**.

2. Press the soft button **[WEAR]**.
The screen displays the tool offset data.



WEAR OFFSETS O:0076 N:0000

No.	X	Z	R	T
001	0.011	0.000	0.000	0
002	0.000	0.000	0.000	0
003	0.000	0.000	0.000	0
004	0.552	0.002	0.000	0
005	0.000	0.000	0.000	0
006	0.000	0.000	0.000	0
▶007	0.000	0.055	0.062	6
008	0.000	0.000	0.000	0

ACTUAL POSITION (REL)

U 80.000
W 34.112

READY

S600

3. The desired compensation value can be selected by the following way:

- Move the cursor to the compensation value to be changed using the page and cursor keys.

- Press the **[X]** or **[Z]** button. Then press the **[INPUT]** button.

4. Enter the compensation value and press the **[INPUT]** button.

Will be able to use relative and absolute values for compensation. When **X** and **Z** are used absolute values will be inserted and by the use of **U** and **W** - relative.

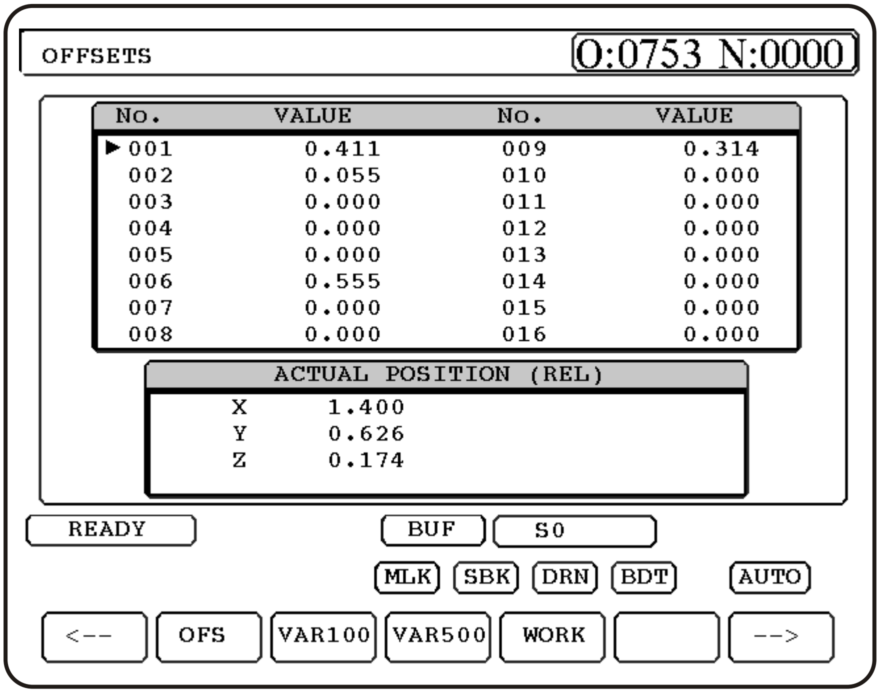
11.4.2 Displaying and setting the tool geometry offset value.

The tool geometric offset value and tool radius value are set by the geometry and wear offset number correction in the **T** code. These values can be set and displayed on the screen.

PROCEDURE FOR DISPLAYING AND SETTING THE TOOL GEOMETRY OFFSET COMPENSATION VALUE

1. Press the functional key **[OFSET]**.

2. Press the soft button **[WEAR]**.
The tool geometry offset compensation values are displayed on the screen.



OFFSETS O:0753 N:0000

No.	VALUE	No.	VALUE
▶ 001	0.411	009	0.314
002	0.055	010	0.000
003	0.000	011	0.000
004	0.000	012	0.000
005	0.000	013	0.000
006	0.555	014	0.000
007	0.000	015	0.000
008	0.000	016	0.000

ACTUAL POSITION (REL)

X	1.400
Y	0.626
Z	0.174

READY BUF S0

3.The desired geometry offset compensation value can be selected by the following way:

- Move the cursor to the geometry compensation value that will be changed with the use of the page change buttons and the cursor moving buttons.
- Press the **[X]** or **[Z]** button. Then press the **[INPUT]** button.

4. Enter the geometry compemsation value and press the **[INPUT]** button .

Will be able to use relative and absolute values for compensation. When **X** and **Z** are used absolute values will be inserted and by the use of **U** and **W** - relative.

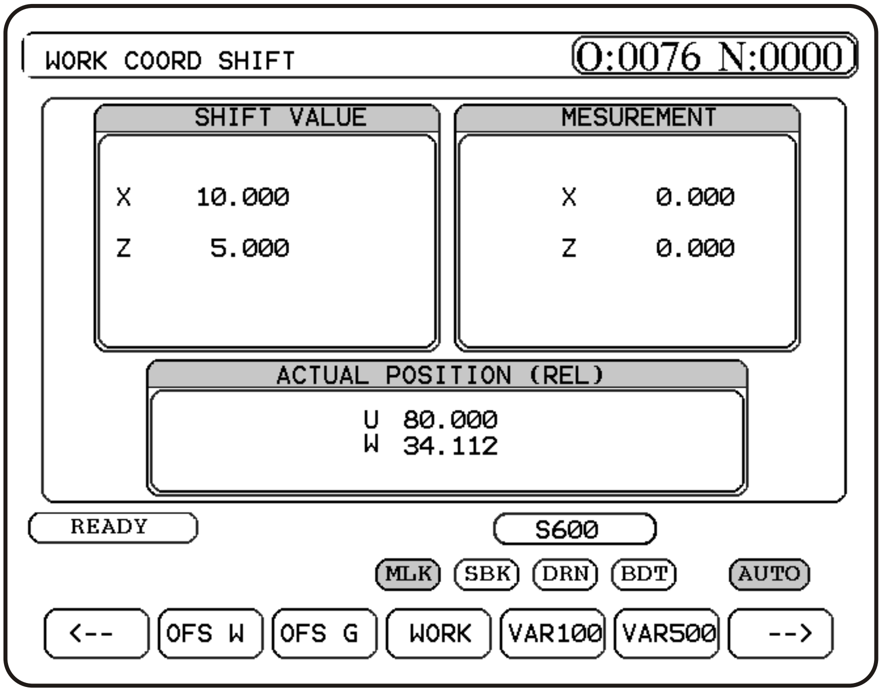
11.4.3 Work coordinate shift.

PROCEDURE FOR DISPLAYING AND SETTING THE WORK COORDINATE SHIFT VALUE

1. Press the functional key **[OFSET]**.

2. Press the soft key **[W.SHIFT]**.

On the screen is displayed:



```
WORK COORD SHIFT                      O:0076 N:0000

SHIFT VALUE                  MESUREMENT
X    10.000                  X    0.000
Z     5.000                  Z    0.000

ACTUAL POSITION (REL)
U  80.000
W  34.112

READY                        S600
```

3. Enter **X(Z)** and shift value.

4. Press the **[INPUT]** botton.

Note: Could be entered relative values with **U(W)**, too. This functional is possible if in one parameter No.10 bit WSFT is set.

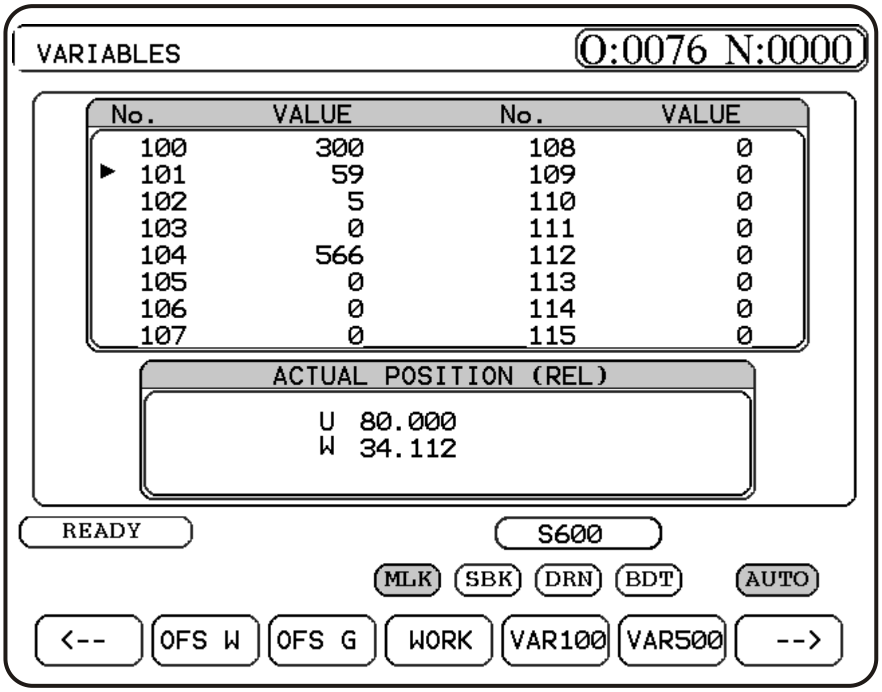
11.4.4 Displaying and setting custom macro common variables

The common variables (**#100** to **#131** and **#500** to **#531**) are displayed. When the absolute value of the common variable exceeds **99999999**, ******** is displayed. On this screen variable values can be changed. The variables could be set in relative coordinates, too.

PROCEDURE FOR DISPLAYING AND SETTING CUSTOM MACRO COMMON VARIABLES

1. Press the functional button **[OFSET]**.

2. Press the soft button **[MACRO]**. This screen is displayed:



VARIABLES   O:0076 N:0000

No.	VALUE	No.	VALUE
100	300	108	0
► 101	59	109	0
102	5	110	0
103	0	111	0
104	566	112	0
105	0	113	0
106	0	114	0
107	0	115	0

ACTUAL POSITION (REL)

U 80.000
W 34.112

READY   S600

3. Move the cursor to the variable that will be changed.

4. Enter the data and press the **[INPUT]** button.

5. The screens with variables from No.**100** to **131** and screens with variables from **500** to **531** take turns by pressing the soft button **[MACRO]**.

11.5 SCREENS DISPLAYED BY PRESSING THE FUNCTIONAL KEY [PARAM]

Parameters must be set to determine the specifications and the functions of the machine in order to fully utilize the characteristics of the servo motor and the other parts of the system.

This chapter describes how to set parameters by the **MDI** panel. Parameters can also be set by an external input/output device.

If the **[PARAM/DGNOS]** functional key is pressed, the following data can be displayed and set:

- **Setting data**

- **Parameters**

- **Self - diagnostic data**

- **Controller ladder diagram**

11.5.1 Displaying and setting parameters

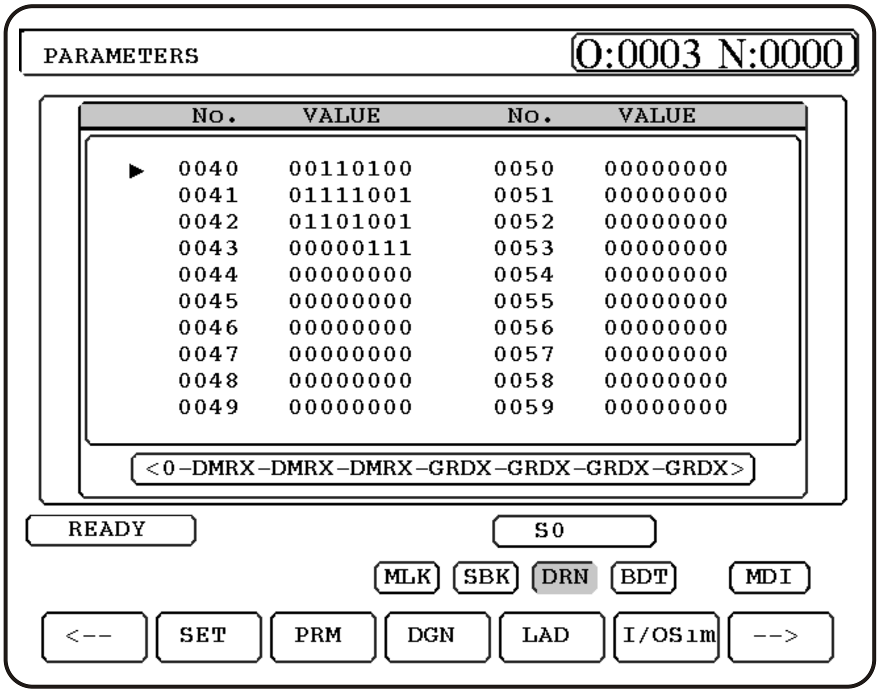
Parameters are set to determine the specifications and functions of the machine in order to fully utilize the characteristics of the servo motor. The parameter's setting depends on the machine. For more information refer to the parameter list prepared by the machine builder. Usually, the user doesn't need to change the parameter settings.

PROCEDURE FOR DISPLAYING AND SETTING PARAMETERS

1. When setting a parameter enable writing first. See the procedure for enabling/ disabling the parameter writing described below.

2. Press the functional button **[PARAM]** .

3. Press the soft button **[PRM]**. The parameter screen is displayed.



```
PARAMETERS                                  O:0003 N:0000

        No.      VALUE           No.      VALUE

   ▶   0040     00110100        0050     00000000
       0041     01111001        0051     00000000
       0042     01101001        0052     00000000
       0043     00000111        0053     00000000
       0044     00000000        0054     00000000
       0045     00000000        0055     00000000
       0046     00000000        0056     00000000
       0047     00000000        0057     00000000
       0048     00000000        0058     00000000
       0049     00000000        0059     00000000

      <0-DMRX-DMRX-DMRX-GRDX-GRDX-GRDX-GRDX>

READY                                S0
```

4. Move the cursor to the parameter number to be changed in one of the following ways:

- Press the **[No]** button and input the parameter number. Then press the **[INPUT]** button.
- Move the cursor to the parameter number using the cursor move keys and the page change keys.

5. Enter the desired value by the numeric keys and then press the **[INPUT]** button.

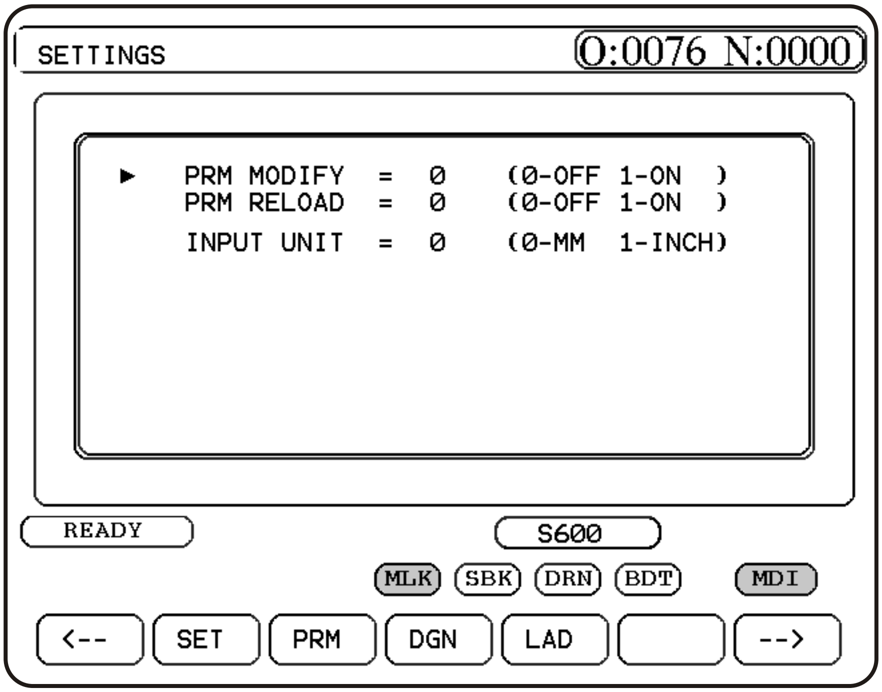
6. After the parameter is set, disable writing.

PROCEDURE FOR ENABLING/DISABLING PARAMETER WRITING

1. Select **MDI** mode or press the Emergency Stop button.

2. Press the functional button **[PARAM]**.

3. Press the soft button **[SET]** to display the **SETTING** screen.



```
SETTINGS                              O:0076 N:0000

  ►   PRM MODIFY  =  0    (0-OFF 1-ON  )
      PRM RELOAD  =  0    (0-OFF 1-ON  )
      INPUT UNIT  =  0    (0-MM  1-INCH)
```

4. Move the cursor to the parameter write enable field using the cursor move keys.

5. Press **[1]** and then the **[INPUT]** button to enable writing. At this time **CNC** enters in alarm state.

6. After parameters' setting move the cursor to the parameter write enable field and press **[0]**. Then press the **[INPUT]** button.

7. Press the **[RESET]** button to release the alarm condition. If the alarm **code 301** is displayed, turn the power off and then on again. Otherwise the alarm is not released.

◆ PARAMETERS THAT REQUIRE TURNING THE POWER OFF

Some parameters are not effective until the power is turned off and on again after they are set. Setting such parameters causes alarm **code 301**. In this case the power must be turned off and on again.

11.5.2 Displaying and entering setting data

On this screen the operator can enable/disable parameter writing and switch the different operational modes.

PROCEDURE FOR SETTING DATA INPUT

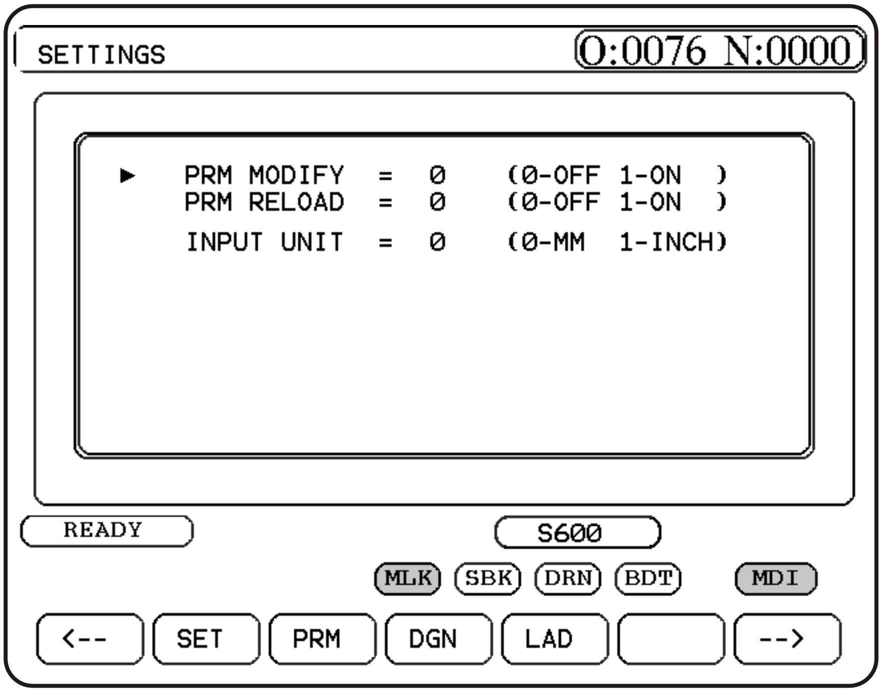
1. Select **MDI** mode.

2. Press the functional button **[PARAM]**.

3. Press the soft button **[SET]** to display the **SETTINGS** screen.

This screen consists of several fields.

The desired field is reached by the cursor move keys. An example of the setting data screen is shown below.



```
▶  PRM MODIFY  =  0    (0-OFF  1-ON   )
   PRM RELOAD  =  0    (0-OFF  1-ON   )
   INPUT UNIT  =  0    (0-MM   1-INCH)
```

- **PARAMETER WRITE (PRM MODIFY)**

Enables or disables parameters' write.

0: disabled

1: enabled

- **INPUT UNIT**

Setting an inch or metric input system.

0: metric

1: inch

- **INCREMENTING THE NUMBER OF THE RAWS**

0: disabled

1: enabled

Note: See parameter 550.

11.6 SCREENS DISPLAYED BY PRESSING THE FUNCTIONAL KEY [ALARM]

The system displays the alarm codes and the operator messages by pressing the **[ALARM]** button.

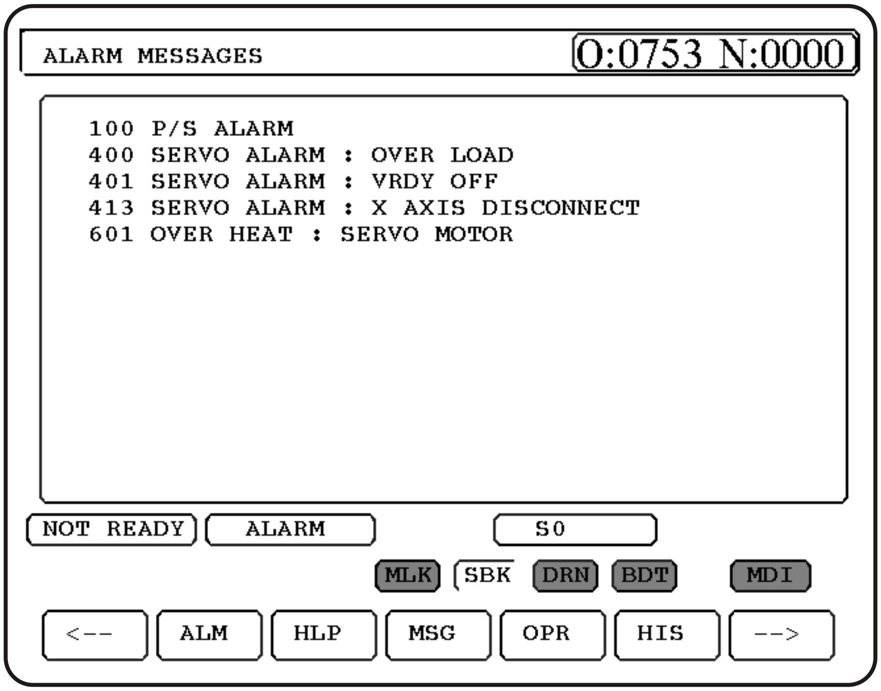
11.6.1 Displaying alarm messages

PROCEDURE FOR DISPLAYING THE ALARM MESSAGES

1. Press the functional button **[OPR/ALARM]**.

2. Press the soft button **[ALARMS].**

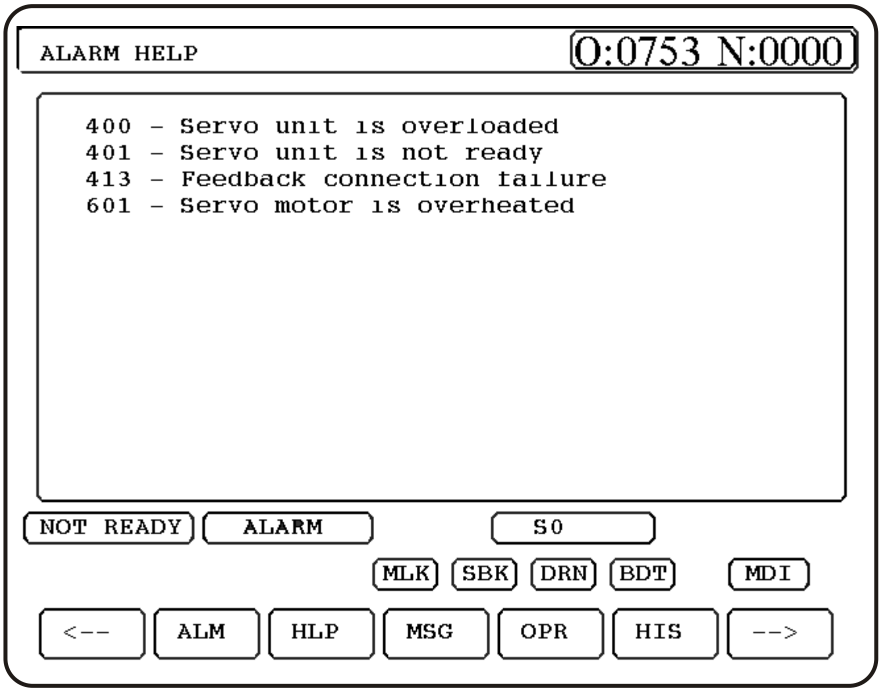
The alarm messages are displayed with their codes.



```
ALARM MESSAGES                         O:0753 N:0000

   100 P/S ALARM
   400 SERVO ALARM : OVER LOAD
   401 SERVO ALARM : VRDY OFF
   413 SERVO ALARM : X AXIS DISCONNECT
   601 OVER HEAT : SERVO MOTOR
```

3. Press the soft button **[PgUp]**.

The alarm messages are displayed with a short description.



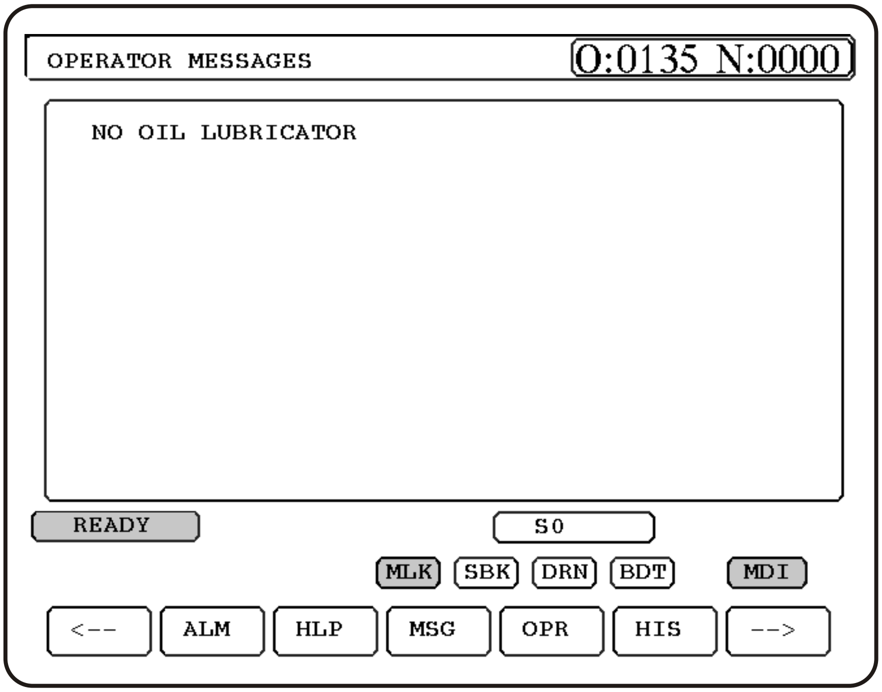
```
ALARM HELP                             O:0753 N:0000

   400 - Servo unit is overloaded
   401 - Servo unit is not ready
   413 - Feedback connection failure
   601 - Servo motor is overheated
```

11.6.2 Displaying operator messages

PROCEDURE FOR DISPLAYING OPERATOR MESSAGES

1. Press the functional button **[OPR/ALARM]**.

2. Press the soft button **[MSG]**.



OPERATOR MESSAGES O:0135 N:0000

NO OIL LUBRICATOR

READY

S0

◆ Specific messages to the operator are displayed on this screen. They are embedded in the controller program of the machine. For more details refer to he manual provided by the machine builder.

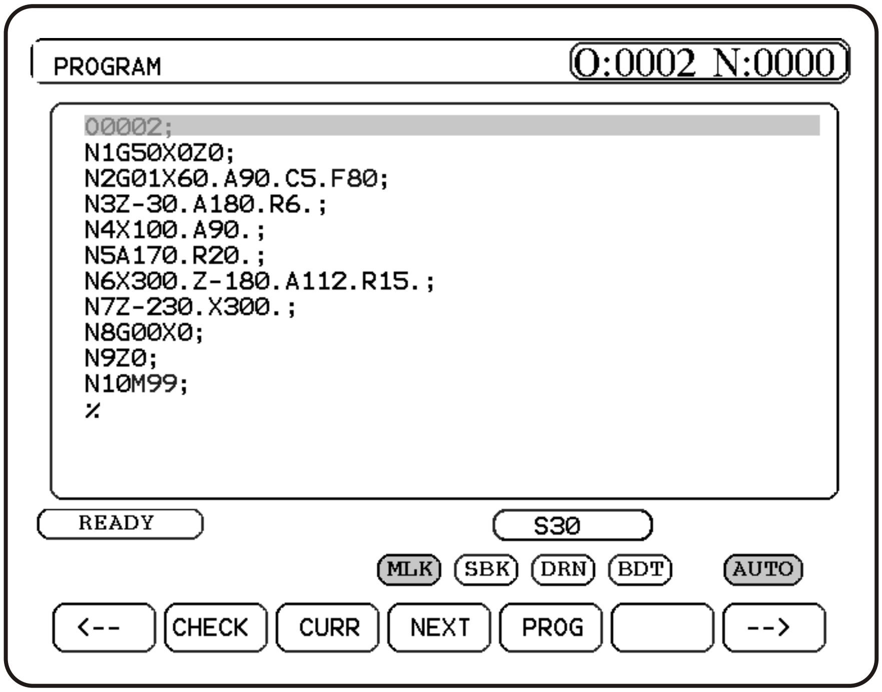
11.7 DISPLAYING THE PROGRAM NUMBER, SEQUENCE BLOCK NUMBER, STATUS AND THE WARNING MESSAGES

The program number, sequence block number and the current status of the **CNC** machine are always displayed on the screen except when the power is turned on.

This chapter describes the program number, sequence block number and status display.

11.7.1 Displaying the program number and sequence block number

The program number and the sequence block number are displayed on the top right corner of the screen as shown below.



```
O0002;
N1G50X0Z0;
N2G01X60.A90.C5.F80;
N3Z-30.A180.R6.;
N4X100.A90.;
N5A170.R20.;
N6X300.Z-180.A112.R15.;
N7Z-230.X300.;
N8G00X0;
N9Z0;
N10M99;
%
```

The program number and the sequence block number that are displayed depend on the screen and is shown below.

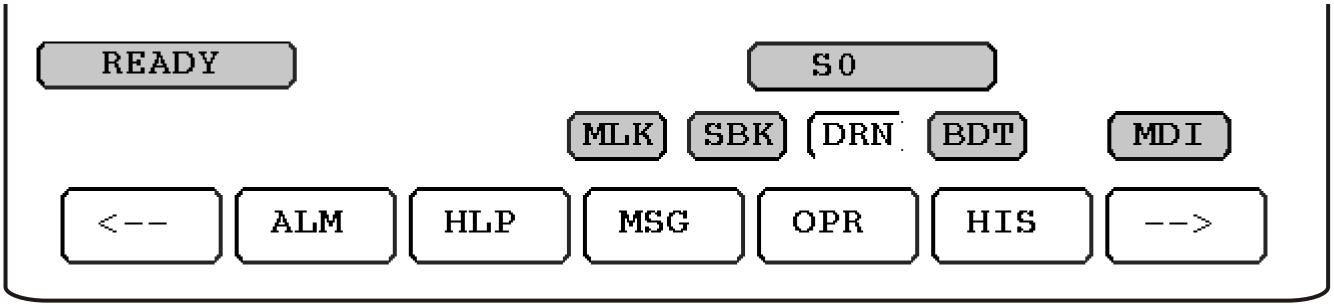
In the program screen in **EDIT** mode: the program number that is being edited and the sequence number prior to the cursor are displayed.

Other than above screens:

The program number and the sequence number executed last are also indicated.

11.7.2 Displaying the status and the warning messages

The system operator can monitor the machine state via the data for the current mode, automatic operation state, the alarm odes and the program editing state, which are all displayed on the screen.



◆ CURRENT MODE

MDI : Manual data input
AUTO : Automatic operation
EDIT : Memory editing
HNDL : Manual handle feed
JOG : manual feed
TJOG/THNDL : teaching mode
STEP : manual incremental feed
ZRN : Manual reference position return

◆ ALARM STATUS

ALARM : Indicates that an alarm is issued
BT : Indicates that the battery is low (inverse message)

◆ OTHER STATUS INFORMATION

DATA IMPORT : Indicates that data is being input
DATA EXPORT : Indicates that data is being export
BUF : Indicates that the block to be executed next is being read
NOT READY : Indicates that the system is in the emergency stop state

11.8 SCREEN GROUP GRAPHICS.

11.8.1 Screen "GRAPHIC PARAMETERS"

All parameters of the graphics refer to the established absolute coordinate system (relative and machine coordinates don't change the graphic). The units of the parameters of the graphics are 0.001mm when working in millimeters and 0.0001 when working in inches. The values of the all coordinates (absolute, relative and machine) can be seen on the screen **"POSITIONS"** and with pressing the functional key **"ALL"**.

Parameters of the graphic:

center X - coordinate of the **X** axis and the center of the graphic coordinate system

center Z - coordinate of the **Z** axis and the center of the graphic coordinate system

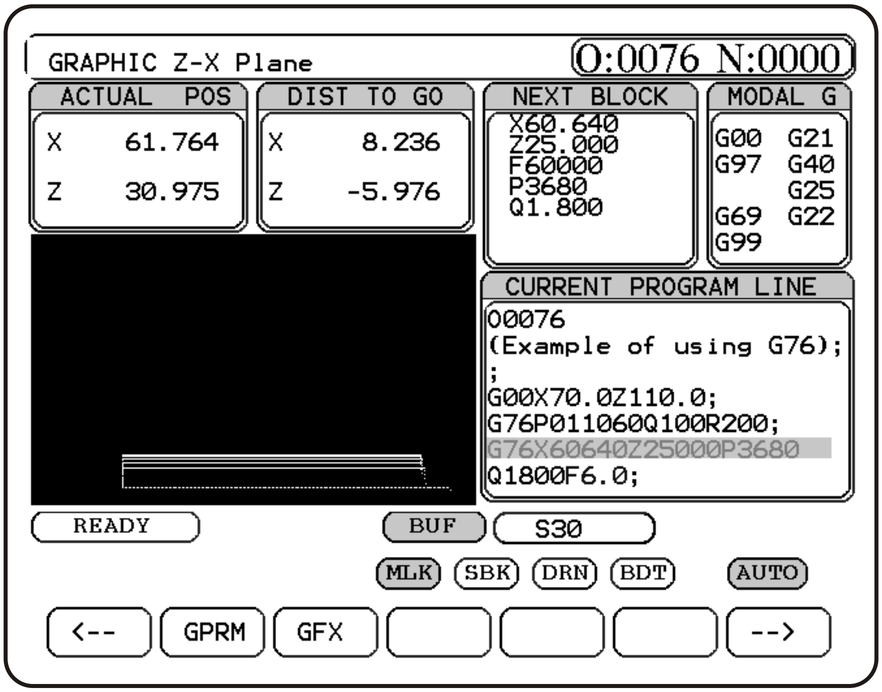
WIDTH - defines the width of the graphic that will be displayed on the screen

HEIGHT - defines the height of the graphic that will be displayed on the screen

The direction of the axes **X** and **Z** on the graphic are set by parameter No.123. The graphic meaning of the parameters can be understood with the following example:

Example:

CENTER **X** = -100000
CENTER **Z** = -100000
WIDTH = 800000
HEIGHT = 500000



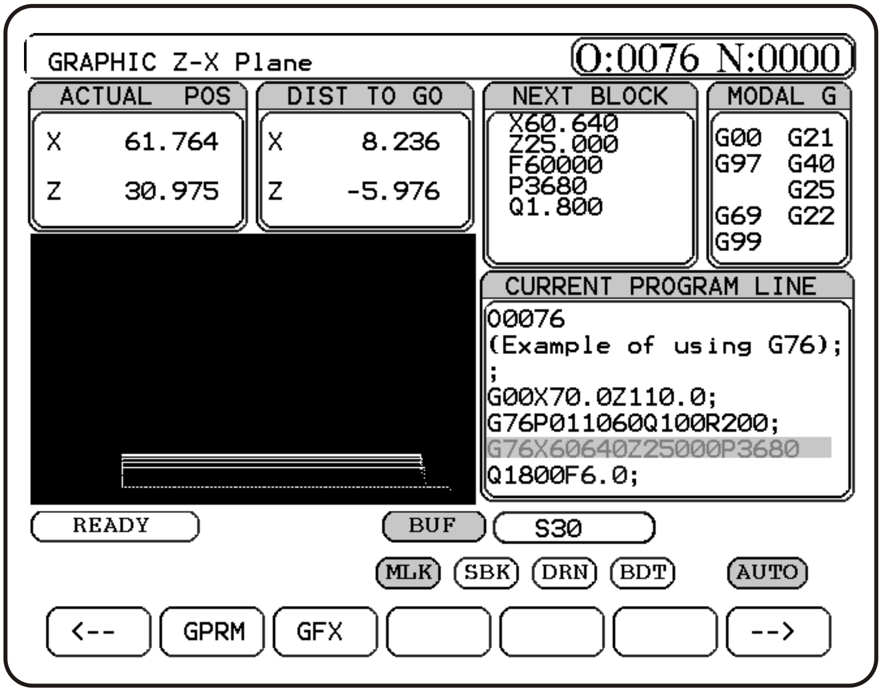
```
GRAPHIC Z-X Plane                    O:0076 N:0000
ACTUAL POS      DIST TO GO      NEXT BLOCK      MODAL G
X   61.764      X    8.236      X60.640         G00  G21
Z   30.975      Z   -5.976      Z25.000         G97  G40
                                F60000               G25
                                P3680           G69  G22
                                Q1.800          G99
CURRENT PROGRAM LINE
O0076
(Example of using G76);
;
G00X70.0Z110.0;
G76P011060Q100R200;
G76X60640Z25000P3680
Q1800F6.0;
READY          BUF   S30
```

To visualize the graphic of the executed program the values of the absolute coordinates should be changed in the interval:

X - from -100000 to 400000

Z - from -100000 to 700000

11.8.2 Screen "GRAPHICS".



```
GRAPHIC Z-X Plane                    O:0076 N:0000
ACTUAL POS      DIST TO GO      NEXT BLOCK      MODAL G
X   61.764      X    8.236      X60.640         G00  G21
Z   30.975      Z   -5.976      Z25.000         G97  G40
                                F60000               G25
                                P3680           G69  G22
                                Q1.800          G99
CURRENT PROGRAM LINE
O0076
(Example of using G76);
;
G00X70.0Z110.0;
G76P011060Q100R200;
G76X60640Z25000P3680
Q1800F6.0;
READY          BUF   S30
```

On this screen the tool traectory and the kind of the feed is shown which is used for moving the tool.

Operations:

[INIT] **-** Clears the current graphic, but the praphic parameters are kept.

[ZOOM] **-** Automatically sets the graphic parameters by a defined rectangular area by the operator. After the first pressing of the functional key **[ZOOM]**, a rectangle is shown on the screen. With the key **[ALTER]** the up-right and down-left point of the rectangle is selected alternatively. The current selected point is moved with the keys [←]**,** [→]**,** [↑] and [↓]**.** By the second push of **[ZOOM]** the graphic coordinate system is scaled. By pressing [CAN] the scale is removed.

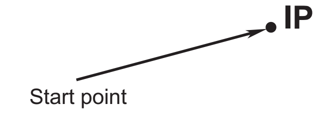
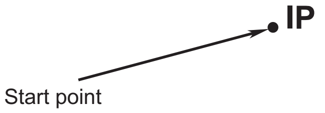
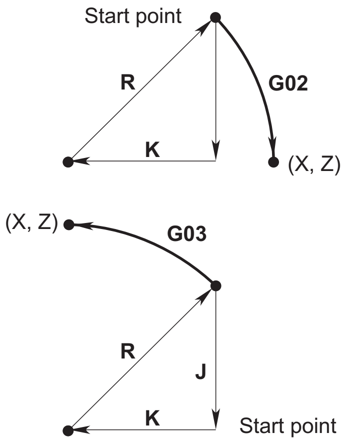
[EXPAND] **-** Increases the current graphic area.

APPENDIX

LIST OF FUNCTIONS

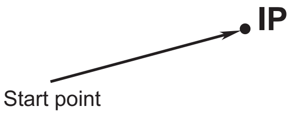
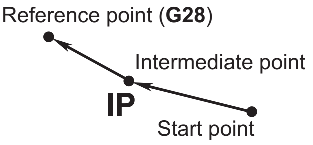
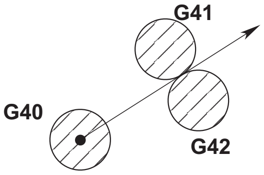
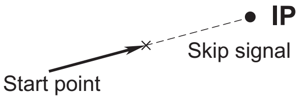
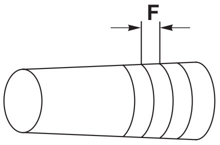
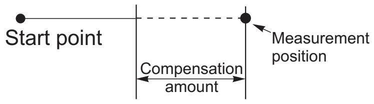
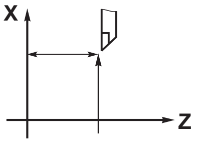
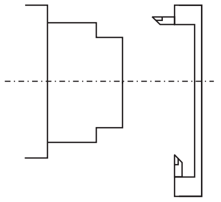
Some functions cannot be added as options depending on the model.

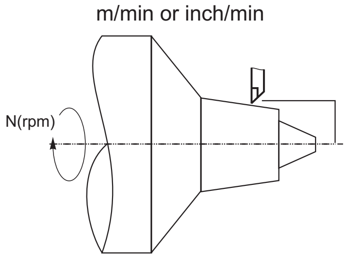
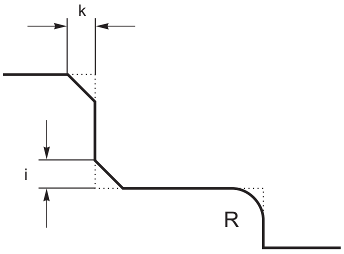
In the tables below, **IP_** presents a combination of arbitrary axis addresses using **X** and **Z** (such as **X_ Z_**).

Functions	Illustration	Format
Positioning (G00)		G00 **IP _ ;**
Linear interpolation (G01)		G00 **IP**_F_**;**
Circular interpolation **(G02, G03)**	 Start point R G02 K (X, Z) (X, Z) G03 R J K Start point	$\left\{\begin{matrix}\textbf{G02}\\\textbf{G03}\end{matrix}\right\}$ **X_ Z_** $\left\{\begin{matrix}\textbf{R\_}\\\textbf{I\_K\_}\end{matrix}\right\}$ **F_ ;**
Dwell (G04)		**G04** $\left\{\begin{matrix}\textbf{X\_}\\\textbf{P\_}\end{matrix}\right\}$ **;**

Change of offset value by program (G10)		Geometry offset amount
		G10 P_X_Z_ R_Q_;
		P=100 + Geometry offset amount
		
		Wear offset value
		G10 P_X_Z_ C_Q_;
		P=1Wear offset amount number

Inch/metric conversion (G20, G21)		Inch input: G20 Metric input: G21
Spindle speed fluctoation detection ON/OFF(G25, G26)		**G25:** **G26P_Q_R_;**

Functions	Illustration	Format	Note
Reference point return check (G27)		**G27 IP _ ;**	
Reference point return (G28) **2nd reference point return (G30)**	 Reference point (G28) Intermediate point IP Start point	**G28 IP _ ;** **G30 P IP _ ;**	
Cutter compensation (G40, G41, G42)		$\left\{\begin{matrix}\textbf{G40}\\\textbf{G41}\\\textbf{G42}\end{matrix}\right\}$ **IP_**	
Skip function (G31)		**G31 IP_ F_ ;**	
Thread cutting (G32)		Equal lead thread cutting **G32 IP_ F_ ;**	
Automatic tool compensation (G36, G37)		**G36X xa** **G37X za**	Depens on a secret parameter
Coordinate system setting. **Spindle speed setting (G50)**		**G50X_Z_ ;** (Coordinate system setting) **G50S_ ;** (Spindle speed setting)	
Mirror image for double turret		**G68 ;** Mirror image for double turret on. **G69 ;** Mirror image cancel.	Depens on a secret parameter
Custom macro (G65)	General format **G65HmP#iQ#jR#K** **m -** Specifies macro functions with 01 to 99 **#i -** Variable number of operation result #j - Variable number 1 used in operation (or constant) **#k -** Variable number 2nd used in operation (or constant) Meaning: #i = #j ⊕ #k	**G65 HmP#iQ#jR#к**	

Functions	Illustration	Format	Note
Per-minute feed **Per-revolution feed**	mm/min inch/min mm/rev inch/rev	**G98 ..... F _ ;** **G99 ..... F _ ;**	
Constant surface speed control ON/OFF (G96, G97)	 	**G96S_ ;** **G97; ....** cancel	
Chamfering, corner R	 	**G01 X(U)** $\left\{\begin{matrix} \mathbf{C \pm k} \\ \mathbf{R \pm k} \end{matrix}\right\}$ **G01 Z(W)** $\left\{\begin{matrix} \mathbf{C \pm k} \\ \mathbf{R \pm k} \end{matrix}\right\}$	Depens on a secret parameter
Canned cycle **(G71 to G76)** **(G90, G92, G94)**		**N_G70P_Q_ ;** **G71U_R_ ;** **G71P_Q_U_W_F_S_T_ ;** **G72W_R_ ;** **G72P_Q_U_W_F_S_T_ ;** **G73U_W_R_ ;** **G73P_Q_U_W_F_S_T_ ;** **G74R_ ;** **G74X(u)_Z(w)_P_Q_R_F_ ;** **G75R_ ;** **G75X(u)_Z(w)_P_Q_R_F_ ;** **G76P_Q_R_ ;** **G76X(u)_Z(w)_P_Q_R_F_ ;** $\left\{\begin{matrix} \mathbf{G90} \\ \mathbf{G92} \end{matrix}\right\}$ **X_Z_I_F_;** **G94X_Z_K_F_ ;**	

RANGE OF COMMAND VALUE

LINEAR AXIS

◆ IN CASE OF METRIC THREAD FOR FEED SCREW AND METRIC INPUT

	Increment system
Least input increment	0.001mm
Least command increment	0.001mm
Max. programable dimension	±9999.999mm
Max. rapid traverse	100000mm/min
Feedrate range	1 to100000mm/min
Least feed	0.001, 0.01, 0.1, 1mm/step
Tool compensation	0 to ±999.999mm
Dwell time	0 to ±9999.999sec

◆ IN CASE OF METRIC THREAD FOR FEED SCREW AND INCH INPUT

	Increment system
Least input increment	0.0001 inch
Least command increment	0.001mm
Max. programable dimension	±999.9999 inch
Max. rapid traverse	100000mm/min
Feedrate range	0.01 to 4000inch/min
Least feed	0.0001, 0.001, 0.01, 0.1inch/step
Tool compensation	0 to ±99.9999 inch
Dwell time	0 to ±9999.999sec

◆ IN CASE OF INCH THREAD FOR FEED SCREW AND INCH INPUT

	Increment system
Least input increment	0.0001 inch
Least command increment	0.0001inch
Max. programable dimension	±999.9999inch
Max. rapid traverse	4000inch/min
Feedrate range	0.01 to 4000inch/min
Least feed	0.0001, 0.001, 0.01, 0.1inch/step
Tool compensation	0 to ±99.9999inch
Dwell time	0 to ±9999.999sec

◆ IN CASE OF INCH THREAD FOR FEED SCREW AND METRIC INPUT

	Increment system
Least input increment	0.001mm
Least command increment	0.0001inch
Max. programable dimension	±9999.999mm
Max. rapid traverse	4000inch/min
Feedrate range	1 to 100000mm/min
Least feed	0.001, 0.01, 0.1, 1 mm/step
Tool compensation	0 to ±999.999mm
Dwell time	0 to ±9999.999sec

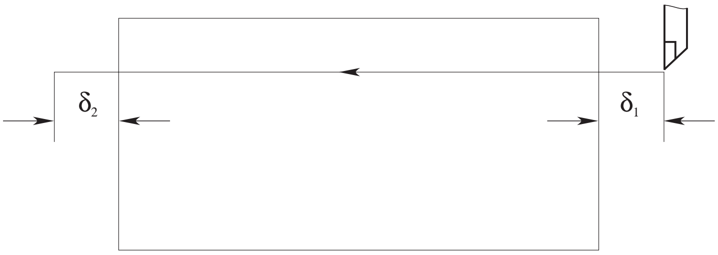
NOMOGRAPHS

APPENDIX 3.1

INCORRECT THREADED LENGTH

The leads of a thread are generally incorrect in $\boldsymbol{\delta_1}$ and $\boldsymbol{\delta_2}$ as shown in the figure below, due to automatic acceleration and deceleration. Thus distance allowance must be made to the extent of $\boldsymbol{\delta_1}$ and $\boldsymbol{\delta_2}$ in the program.

A. *Determinated of* $\boldsymbol{\delta_2}$



$\boldsymbol{\delta_2}$ is determined by cutting speed **V**(mm/sec) and time constant of the servo system as shown below:

$$\delta_2 = \mathbf{T_1 - V(mm)} \ldots\ldots$$

where: $\mathbf{T_1}$ is in sec , and **V** is in mm/sec

V is determined by thread lead - **L** and spindle speed **R** as shown below:

$$\mathbf{V = R(rpm) \times L(mm)/60}$$

Time constant $\mathbf{T_1}$ of the servo system usually is 0.033sec.

B. *Determinated of* $\boldsymbol{\delta_1}$

The value of $\boldsymbol{\delta_1}$ is determinated by cutting speed **V** in thread cutting, time constant **T_1** for the servo system and the thread accuracy "a", as shoun below.

The lead at the beginning of thread cutting is shorter than the specified lead **L**. Thread accuracy **"a",** as shown below:

$$\delta_1 = \{t - T_1 + T_1\exp(-\frac{t}{T_1})\} \quad \textbf{(a)}$$

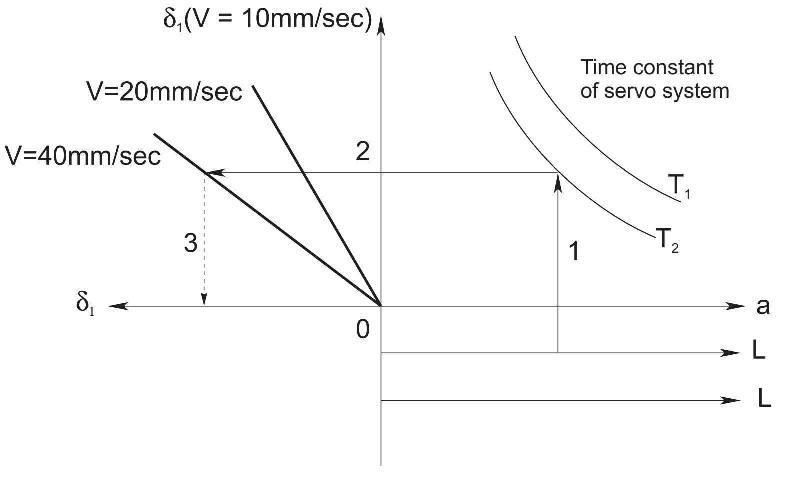
$$a = \exp(-\frac{t}{T_1})\} \ldots\ldots\ldots \quad \textbf{(b)}$$

$$a = \frac{\Delta L}{L} \quad \textbf{(c)}$$

When the value of **"a"** is determined the time lapse until the thread accuracy is attained can be calculated by formula (b).

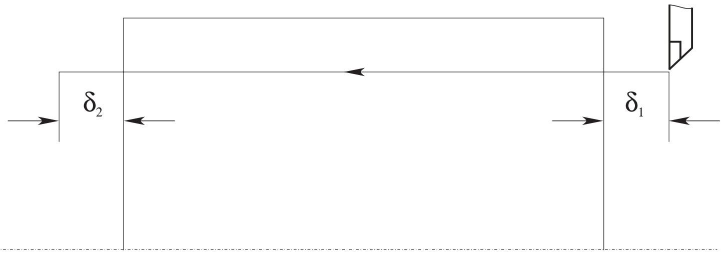
The time **"t"** is substituted in the same way as for **δ_2**.

Since the calculation of **δ_1** is rathercomplex. Instructions on how to read the appropriate **δ_1** value on the nomography is shown below:



First specify the class and the lead of the thread. The thread accuracy **"a"** will be obtained at (1) and depending on the time constant of cutting feed acceleration/ deceleration, the **δ_1** value when **V**=10mm/sec will be obtained at (2). Then **S_1** value at any speed can be obtained at (3) depending on the thread cutting speed.

Simple Calculation of Incorrect Thread Length



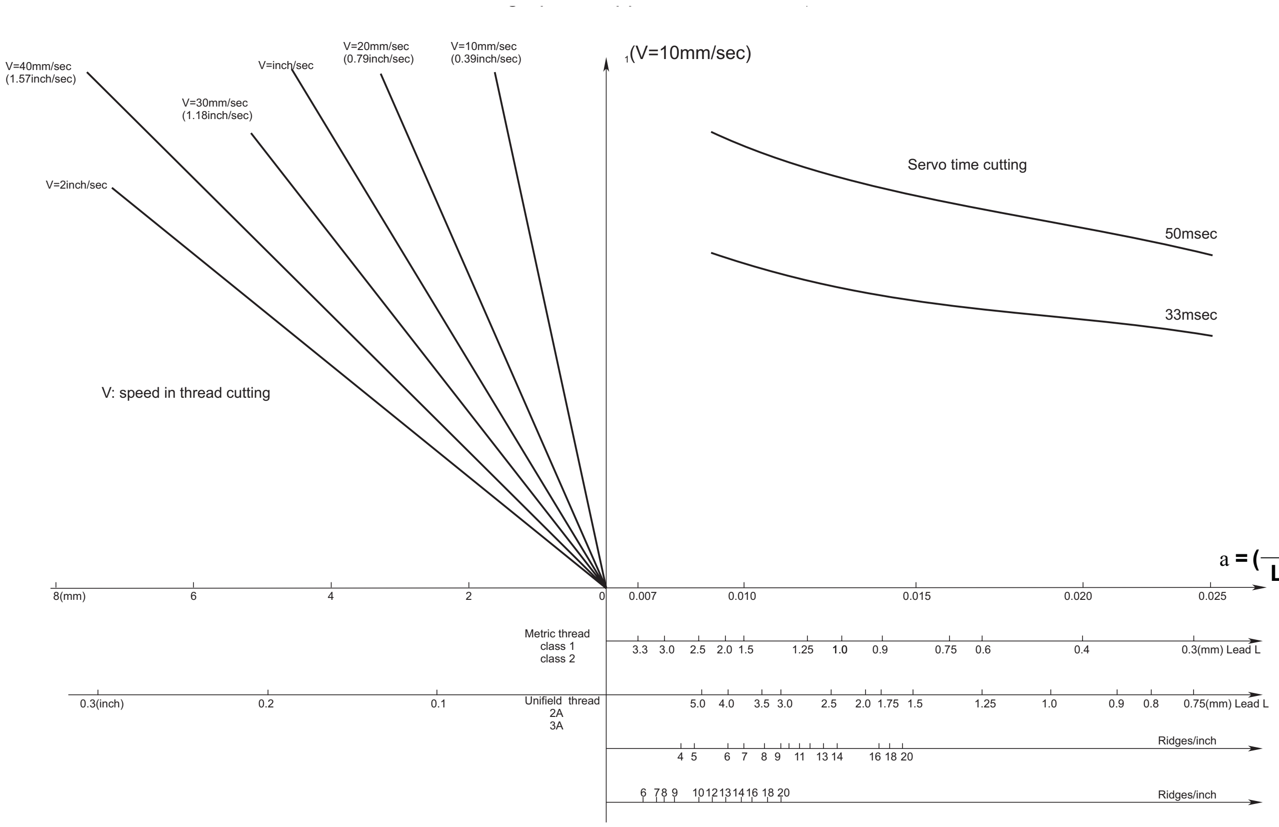
$$\delta_2 = \frac{L.R}{1800\,*}\ (mm)$$

$$\delta_1 = \frac{L.R}{1800\,*}\ (-1-1na)$$

$$= \delta_2\ (-1-1na)$$

"a" - indicates the error allowances
L - thread lead (mm)
R - spindle speed (rpm)
***** - when time constant **T** of the servo system is 0.033sec.
(-1-1na) depending on the following table:

a	-1-1na
0.005	4.298
0.01	3.605
0.015	3.200
0.02	2.912



Example:

R = 350 rpm

L = 1mm

a = 0.01

$$\delta_2 = \frac{\mathbf{3500x1}}{\mathbf{1800}} = \mathbf{0.194mm}$$

$$\delta_1 = \delta_2 \ \mathbf{X\ 3.605 = 0.701mm}$$

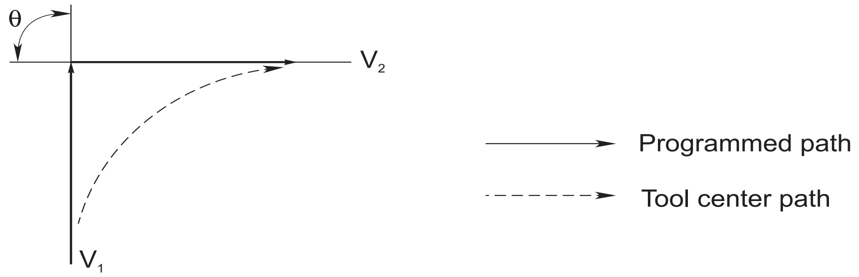
APPENDIX 3.2

TOOL PATH AT CORNER

When servo system delay (by exponential acceleration/deceleration at cutting or caused by the positioning system when a servo motor is used) is accompanied by cornering, a sligth deviation is produced between the tool path (toolcenter path) and the programmed path as shown of the next figure.

Time constant T_1 of the exponential acceleration/deceleration is fixed to 0.

Slight deviation between the tool path and the programmed path.



This tool path is determined by the following parameters:

- Feedrate (**V_1**, **V_2**)
- Corner angle (**θ**)
- Exponential acceleration/deceleration time constant (**T_1**) at cutting (**T_1 = 0**)
- Presence or absence of buffer register

The above parameters are used to theoretically analyze the tool path.

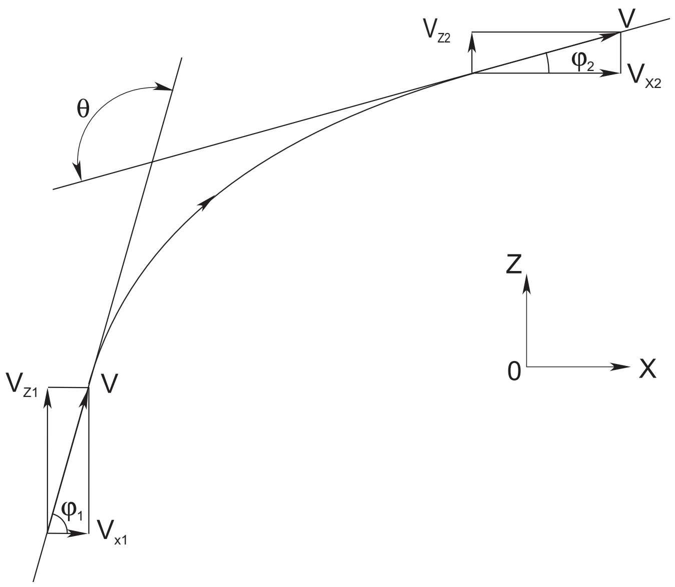
When actually programming, the above items must be considered and programming must be performed carefully so that the shape of the workpiece is within the desired precision.

In the other words, when the shape of the workpiece is not within the theoretical precision, the commands of the next block must not be read until the specified feedrate becomes zero. The dwell function is then used to stop the machine for the appropriate period.

◆ ANALYSIS

The tool path shown below on the figure is analysed on the following conditions:
Feedrate is constant at both blocks before and after cornering.
The controller has a buffer register.

◆ DESCRIPTION OF CONDITIONS AND SYMBOLS



$$V_{X1} = V\cos\phi_1$$

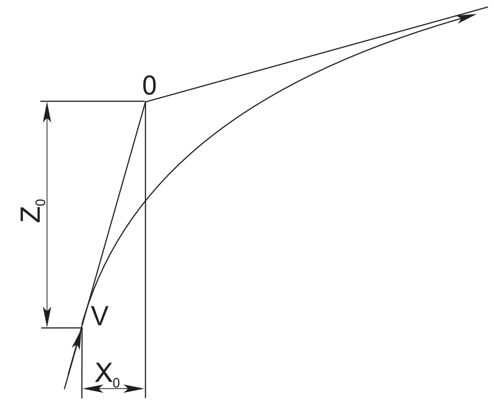
$$V_{Z1} = V\sin\phi_1$$

$$V_{X2} = V\cos\phi_2$$

$$V_{Z2} = V\sin\phi_2$$

V : Feed rate at the block before and after cornering
V_{X1} : **X**-axis component of feedrate of preceding block
V_{Z1} : **Z**-axis component of feedrate of preceding block
V_{X2} : **X**-axis component of feedrate of following block
V_{Z2} : **Z**-axis component of feedrate of following block
θ : Corner angle
ϕ_1 : Angle formed by the specified path direction of the preceding block and **X**-axis
ϕ_2 : Angle formed by the specified path direction of the following block and **X**-axis

◆ INITIAL VALUE CALCULATION



The initial value when cornering begins, that is, the **X** and **Z** coordinates at the end of command distribution by the controller, is determined by the feedrate and the positioning system time constant of the servo motor.

$$X_0 = V_{X1}(T_1 + T_2)$$

$$Z_0 = V_{Z1}(T_1 + T_2)$$

T_1: Exponential acceleration/deceleration time constant. **(T = 0**)

T_2: Time constant of positioning system. (Inverse of position loop gain.)

◆ ANALYSIS OF CORNER TOOL PATH

The equations below represent the feedrate for the corner section in **X** axis direction and **Z** axis direction..

$$V_X(t) = (V_{x2} - V_{x1})\,[1 - \frac{V_{X1}}{T_1 + T_2}\{T_1\exp(-\frac{t}{T_1}) - T_2\exp(-\frac{t}{T_2})\} + V_{X1}\,] =$$

$$= \frac{V_{X1} - V_{X2}}{T_1 - T_2}\{T_1\exp(-\frac{t}{T_1}) - T_2\exp(-\frac{t}{T_2})\} + V_{X2} \,...........$$

$$V_z(t) = \frac{V_{Z1} - V_{Z2}}{T_1 - T_2}\{T_1\exp(-\frac{t}{T_1}) - T_2\exp(-\frac{t}{T_2})\} + V_{z2} \,...........$$

Therefore, the coordinates of the tool path at time **t** are calculated from the following equations:

$$X(t) = {}_0^1 V_X(t)dt - X_0 =$$

$$= \frac{V_{X2} - V_{X1}}{T_1 - T_2}\{T_1^2\exp(-\frac{t}{T_1}) - T_2^2\exp(-\frac{t}{T_2})\} - V_{X2}(T_1 + T_2 - t) \,........$$

$$Z(t) = {}_0^1 V_z(t)dt - Z_0 =$$

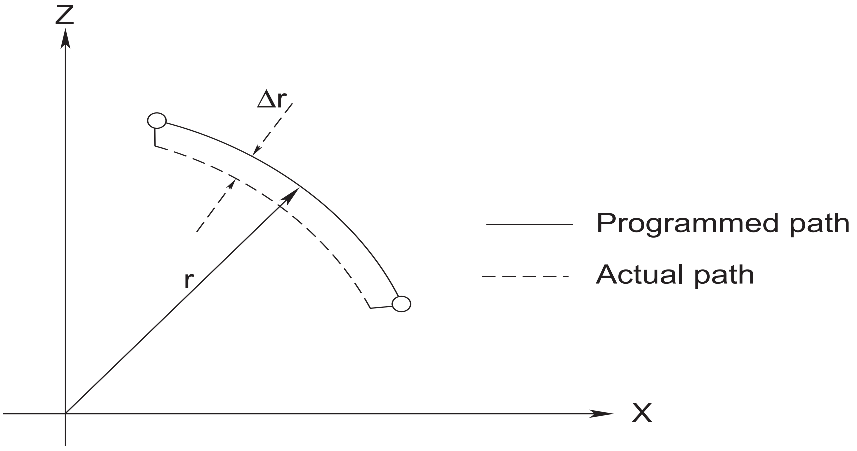
$$= \frac{V_{z2} - V_{z1}}{T_1 - T_2}\{T_1^2\exp(-\frac{t}{T_1}) - T_2^2\exp(-\frac{t}{T_2})\} - V_{z2}(T_1 + T_2 - t) \,.........$$

APPENDIX 3.3

RADIUS DIRECTION ERROR AT CIRCLE CUTTING

When a servo motor is used, the positioning system caused an error between input commands and output results. Since the tool advances along the specified segment, an error is not produced in linear interpolation. In circular interpolation, however, radial errors may be produced, specially for circular cutting at high speeds.

This error can be obtained as follows:



$$\Delta r = 1/2(T_1^2 + T_2^2)\frac{V^2}{r} \text{ ....}$$

Δr : Maximum radius error (mm)

V : Feedrate (mm/s)

r : Circle radius (mm)

T_1 : Exponential acceleration/deceleration time constant(sec) at cutting (T = 0)

T_2 : Time constant of positioning system (sec)(Inverse of position loop gain).

Since the machining radius **r**(mm) and allowable error **Δr**(mm) of the workpiece is given in actual machining, the allowable limit feedrate **V**(mm/sec) is determinated by the equation above.

Since the acceleration/deceleration time constant at cutting which is set by this equipment varies with the machine tool builder.

Table of System Errors for ETA-17 CNC 10,
model T, version 3.00

1. P/S alarms

Alarm	Contents
003	Data exceeding the maximum allowable number of digits was input.
005	The address was not followed by the appropriate data but was followed by another address or **EOB** code.
006	Sign (**-**) input error. Sign (**-**) was input after an address with which it cannot be used. Or two or more (**-**) signs were input.
007	Decimal point (.) input error (A decimal point was input after an address in which it can't be used. Or two decimal points were input.)
009	Unusable character was input in significant area.
010	An unusable **G** code was commanded.
011	Feedrate was not commanded to a cutting block or the feedrate was inadequate.
023	In circular interpolation by radius designation, a negative value was commanded for address R.
029	The offset value specified by **T** code is too large.
030	The offset number in **T** function specified for tool offset is too large.
031	In setting an offset amount by **G10**, the offset number following the address **P** was excessive or it was not specified.
032	In setting an offset amount by **G10**, the offset amount was excessive.
033	A point of intersection cannot be determined for tool nose radius compensation.
034	The start up or cancel was going to be performed in the **G02** or **G03** mode tool nose radius compensation.
035	Skip cutting **G31** was specified in tool nose radius compensation mode.
038	Overcutting will occur in tool nose radius compensation because the arc start point or the end point coincides with the arc center.
039	Chamfering or corner **R** was specified with a start-up, a cancel, or switching between **G41** and **G42** in tool nose radius compensation. The program may cause overcutting to occur in chamfering or corner **R**.
G40	Overcutting will occur in tool nose radius compensation in a canned cycle **G90** or **G94**.

Alarm	Contents
041	Overcutting will occur in tool radius compensation.
050	Chamfering and corner R are commanded in the thread cutting block.
051	The block next to the block for which chamfering and corner R are commanded is not **G01**.
052	Improper movement or the more distance of the block next to that for which chamfering and corner R are commanded.
054	A block in which the chamfering or the corner R was specified includes a taper command.
055	In the block for which chamfering and corner R are commanded, the move distance is commanded less than the corner R amount.
059	The program with the selected number cannot be searched, in external program number search.
060	Commanded sequence number was not found in the sequence number search.
061	Address **P** or **Q** is not specified in **G70, G71, G72** or **G73** command.
062	* The depth of cut in **G71** or **G72** is zero or negative value; * The repetitive count in **G73** is zero or negative value; * The negative value is specified to **Δi** or **Δk** in **G74** or **G75**; * A value other than zero is specified to address **U** or **W**, thouh **Δi** or **Δk** is zero in **G74** or **G75**; * A negative value is specified to **Δi** thougt the relief direction in **G74** or **G75** is determined; * Zero or negative value is specified to the height of thread or depth of cut of 1st time in **G76**; * The specified minimum depth of cut in **G76** is greater than the height of thread; * An anusable angle of tool tip is specified in **G76**.
063	The sequence number specified by address **P** in **G70, G71, G72** or **G73** command cannot be searched.
065	* **G00** or **G01** is not commanded at the block with the sequence number which is specified by address **P** in **G71, G72** or **G73** command; * Address **X(W)** or **Z(U)** was commanded in the block with a sequence number which is specified by address **P** in **G71** or **G72**, respectively.
066	An Unallowable **G** code was commanded between two blocks specified by address **P** and **Q** in **G71, G72** or **G73**.
067	**G70, G71, G72** or **G73** command with address **P** and **Q** was specified in **MDI** mode.
069	The final move command in the blocks specified by **P** and **Q** of **G70, G71, G72** and **G73** ended with chamfering or corner **R**.

Alarm	Contents
070	The memory area is insufficient.
071	The address to be searched was not found. Or the program with specified program number was not found in program number search.
072	The number of programs to be stored exceeded 125 (option).
074	The program number is other than 1 to 9999.
076	Address **P** was not commanded in the block which includes an **M9**8 command or **G65** command.
077	The subprogram was called in three or five folds.
078	A program number or a sequence number which was specified by address **P** in the block which includes an **M98, M99, G65** or **G66** was not found.
079	The contents of the program stored in memory did not agree with the tape during collation.
080	When the axis reached the area specified by parameter **N731**, **N732**, the PMC signals **XAE** and **ZAE** were not output from the machine tool. (Automatic tool compensation function.)
081	Automatic tool compensation was specified without a **T** code. (Automatic tool compensation function.)
082	**T** code and automatic tool compensation were specified in the same block. (Automatic tool compensation function.)
083	In automatic tool compensation, an invalid axis was specified or the command is incrementall. (Automatic tool compensation function.)
090	The reference point return cannot be performed normally because the reference point is too close to the reference point or the speed is too slow.
092	The commanded axis by **G27** (Reference point return check) did not return to the reference point.
100	**PWE** is set to **1**. Turn it to **0** and reset the system.
111	The calculation result of the macro instruction exceeds the allowable range ($\mathbf{-2^{32}} \div \mathbf{+2^{32}}$).
112	Division by zero was specified. (Including tan 90°)
114	An undefined **H** code is designated in the **G65** block.
115*	A value not defined as a variable number is designated.
116	The variable number designated with **P** is forbidden for assignment.
119	The argument of **SQRT** or **BCD** is negative.
125	An unusable address is used in **G65** block.
128	The sequence number specified in the branch command was not **0** to **9999**. Or, it cannot be searched.

Alarm	Contents
200	**CONNECT BROKEN:** Communication via **RS** was broken by user.
201	**REMOTE DON'T REPLY:** The remote device has terminated
202	**REMOTE CAN'T OPEN THE FILE:** The remote device can't find the specified file.
203	**REMOTE CAN'T WRITE:** The remote device can't write the received data.
204	**LOAD HAS NOT FINISHED:** The loading operation has not finished completely..
205	**DATA ERROR:** The data which the device received from remote hasn't correct format.
300	**PMC-X ALARM: PMC-X** not present.
301	**TURN THE POWER OFF:** Some parameter was modified which required turn the power off(reset the system)
302	**CLEAR MEMORY:** The power turned off while rewriting the memory by program edit operation. Reload all data.

2. Servo alarms

Servo alarm	Contents
400:(OVER LOAD)	Overload signal turns on **X, Z**
401:(VRDY OFF)	Velocity control **READY** signal **VRDY** turns off.
402:(OVER LOAD)	**4**th **-** axis overload signal turns off.
403:(VRDY OFF)	**4**th **-** axis velocity control **READY** signal **VRDY** turns off.
404:(VRDY ON)	Position control **READY** signal **PRDY** turns off, while velocity control **READY** signal **VRDY** does not turn off, or velocity control **READY** signal **VRDY** turns on, although **READY** signal **PRDY** is not turned on yet when turning on the power supply.
405:(WRONG ZRN)	A position control system error. There is a reference point return failure due to a problem in the CNC or the servo system. Start operations with the manual reference point return.
410:X AXIS EXCESS ERROR	For **X** axis, the position deviation amount while stopped is greater than the parameter setting value. Refer to parameter **P593**.
411:X AXIS EXCESS ERROR	For **X** axis, the position deviation amount while moving is greater than the parameter setting value. Refer to parameter **P504**.
412:X AXIS EXCESS DRIFT	Drift amount of **X** axis is exceed. (Exceed **500 VELO)**
413:X AXIS LSI OVER	The contents of the **X** - axis error register exceed ± **32767** or the velocity command value of **DA** converter is outside a range of **-8192** to **+8191**. This error is usually caused by various parameter setting errors.
415:X AXIS MOTION OVER	An attempt was made to specify a velocity exceeding **511875** detection unit/sec in **X** axis. This error is usually caused by **CMR**. setting error.
416:X AXIS DISCONNECT	There is an **X** axis pulse coder position detection system abnormally..
420:Z AXIS EXCESS ERROR	For **Z** axis, the position deviation amount while stopped is greater than the parameter setting value. Refer to parameter **P594**.
421:Z AXIS EXCESS ERROR	For **Z** axis, the position deviation amount while stopped is greater than the parameter setting value. Refer to parameter **P505**.
422:Z AXIS EXCESS DRIFT	Drift amount of **Z** axis is exceed. (Exceed **500 VELO)**
423:Z AXIS LSI OVER	The position deviation amount exceeds ± **32767** in **Z** axis or the velocity command value of **DA** converter is outside a range of **-8192** to **+8191.** This error is usually caused by various parameter setting errors.
425:Z AXIS MOTION OVER	An attempt was made to specify a velocity exceeding **511875** detection unit/sec in **Z** axis. This error is usually caused by **CMR** setting error.
426:Z AXIS DISCONNECT	There is an **Z** axis pulse coder position detection system abnormally..
430:3 AXIS EXCESS ERROR	For the **3**rd axis, the position deviation amount while stopped is greater than the parameter setting value. Refer to parameter **P595**.
431:3 AXIS EXCESS ERROR	For the **3**rd axis, the position deviation amount while stopped is greater than the parameter setting value. Refer to parameter **P506**.
432:3 AXIS EXCESS DRIFT	Drift amount of **3**rd axis is exceed. (Exceed **500 VELO)**
433:3 AXIS LSI OVER	The position deviation amount exceeds ± **32767** in the **3**rd axis or the velocity command value of **DA** converter is outside a range of **-8192** to **+8191.** This error is usually caused by various parameter setting errors.
435:3 AXIS MOTION OVER	An attempt was made to specify a velocity exceeding **511875** detection unit/sec in the **3**rd axis. This error is usually caused by **CMR** setting error.
436:3 AXIS DISCONNECT	An error of the position detection system of the **3**rd axis pulse coder

Servo alarm	Contents
440:4 AXIS EXCESS ERROR	For the **4**th axis, the position deviation amount while stopped is greater than the parameter setting value. Refer to parameter **P596**.
441:4 AXIS EXCESS ERROR	For the **4**th axis, the position deviation amount while stopped is greater than the parameter setting value. Refer to parameter **P507**.
442:4 AXIS EXCESS DRIFT	Drift amount of **4**th axis is exceed. (Exceed **500 VELO)**
443:4 AXIS LSI OVER	The position deviation amount exceeds ± **32767** in the **4**th axis or the velocity command value of **DA** converter is outside a range of **-8192** to **+8191.** This error is usually caused by various parameter setting errors.
445:4 AXIS MOTION OVER	An attempt was made to specify a velocity exceeding **511875** detection unit/sec in the **4**th axis. This error is usually caused by **CMR** setting error.
446:4 AXIS DISCONNECT	An error of the position detection system of the **4**th axis pulse coder

3. Errors from passing the limits.

Number	Content	Explanation
6n0	**+n**	Passing the defined limits of the (+) axis.
6n1	**-n**	Passing the defined limits of the (-) axis.

Note: *Parameters **P700 - P708** assin the work zone

** Software overtavel limits could be ignored if you press the buttons **CAN + P** while turn the power on.

4. Overheat alarm

Alarm	Content
700 OVER HEAT: CONTOL UNIT	
602 OVER HEAT: SERVO MOTOR	Overheat of servo motor **X**-axis.