

# ETA-17 CNC20T

## Operator's manual

Lathe



# ETA·17

ETA-17 Sofia

Bulgaria

[http: ETA-17.com](http://ETA-17.com)

## 1. General

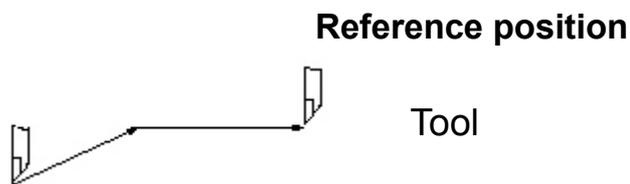
### 1.1. Manual operation.

#### •Manual reference position return (ZRN mode)

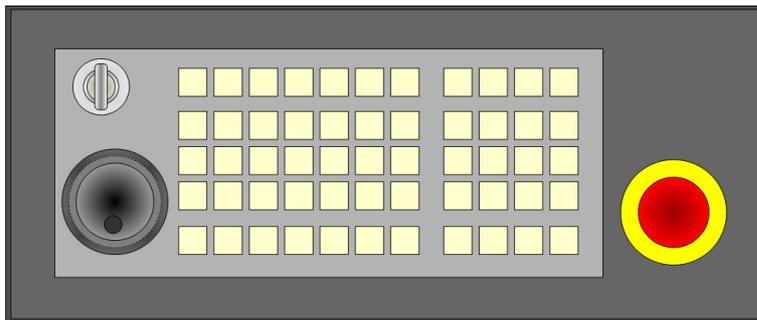
The CNC machine tool has a position called reference position. Here either the tool is replaced or the coordinate system origin is set. Ordinarily, after the power is turned on, the tool is moved to the reference position.

The manual reference position return is an operation of moving the tool to the reference position using pushbutton located on the operator's panel.

### Manual reference position



### Machine operator's panel

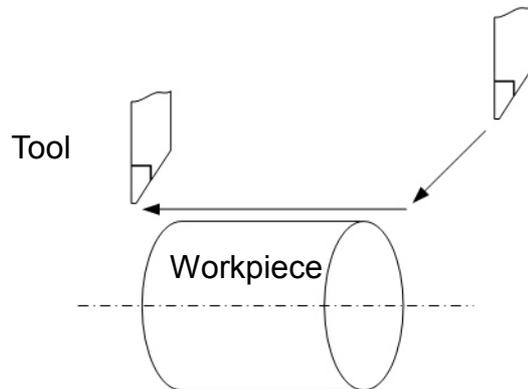


The tool can be moved to the reference position also with program command. This operation is called automatic reference position return.

## • 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



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

#### 1. Jog feed ( JOG mode ).

The tool moves continuously while the pushbutton remains pressed.

#### 2. Incremental feed ( STEP mode ).

The tool is moved to a predetermined distance each time the button is pressed.

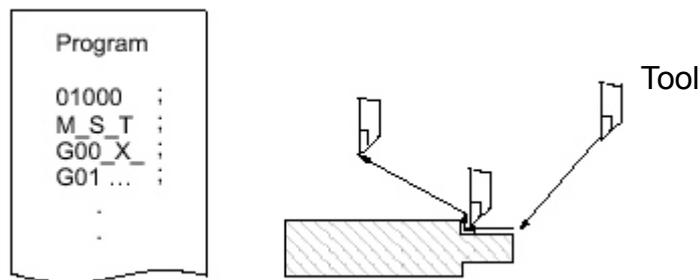
#### 3. Manual handle ( HNDL mode ).

The tool is moved to a distance corresponding to the degree of manual handle rotation.

## 1.2. Tool movement by programming – automatic operation

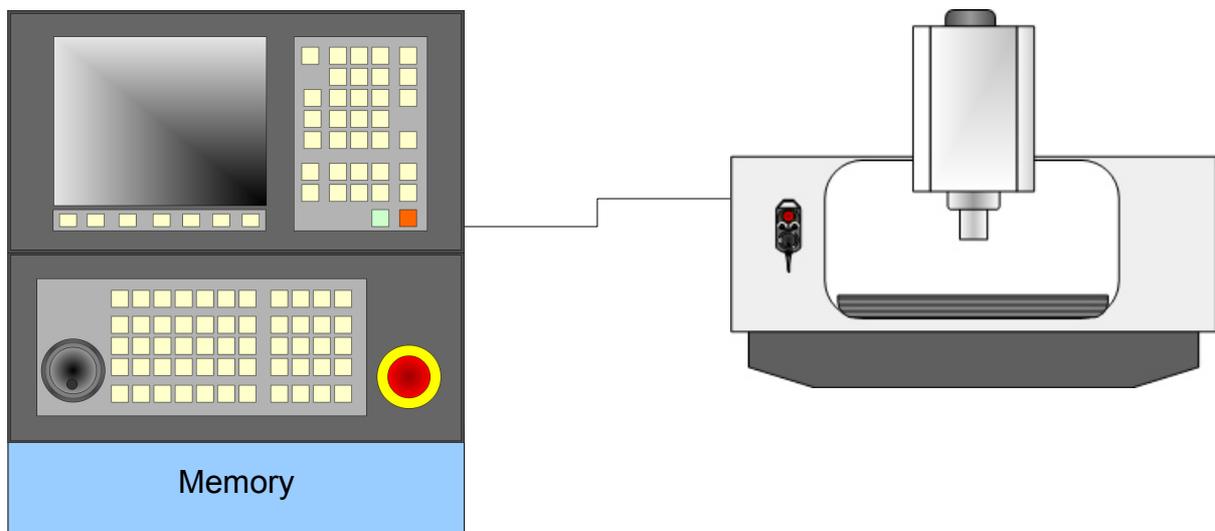
*Automatic operation* means operating the machine according to the created program. It includes the program in memory, DNC or MDI operation.

### Tool movement by program



### •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.



•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.

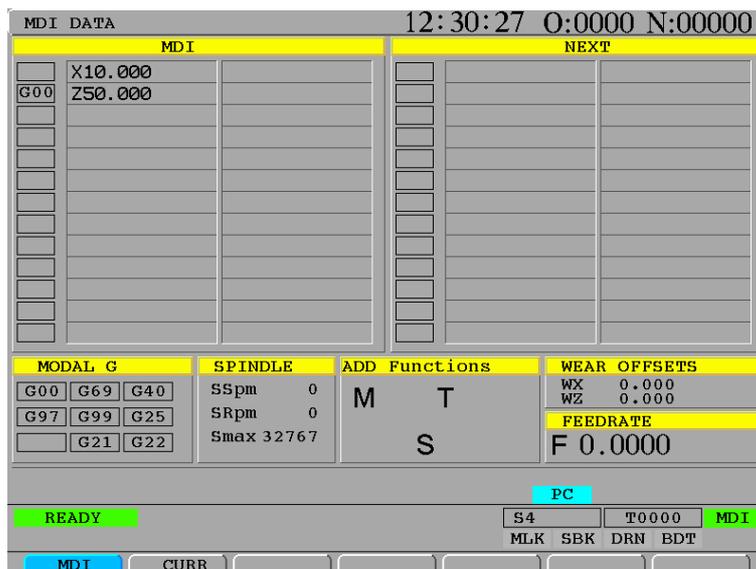
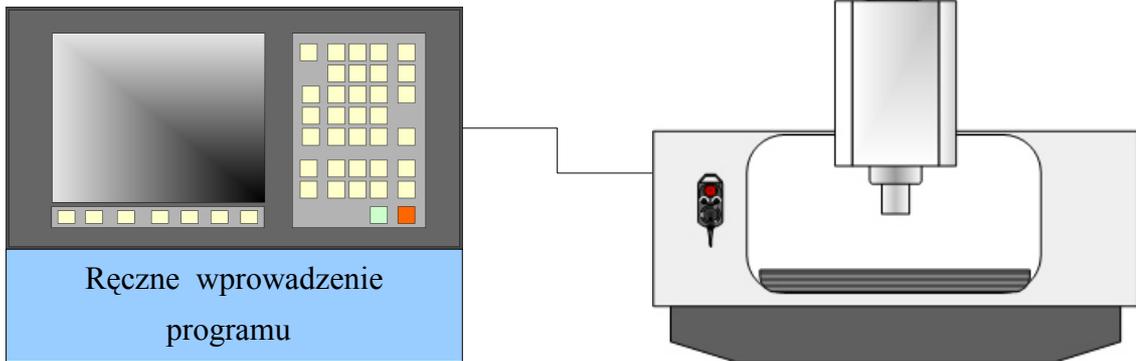
•MDI operation ( mode MDI ).

After a command group is entered from the MDI keyboard, the machine can run according to these commands. This mode is called MDI operation.

**MDI operation**

CNC MDI keyboard

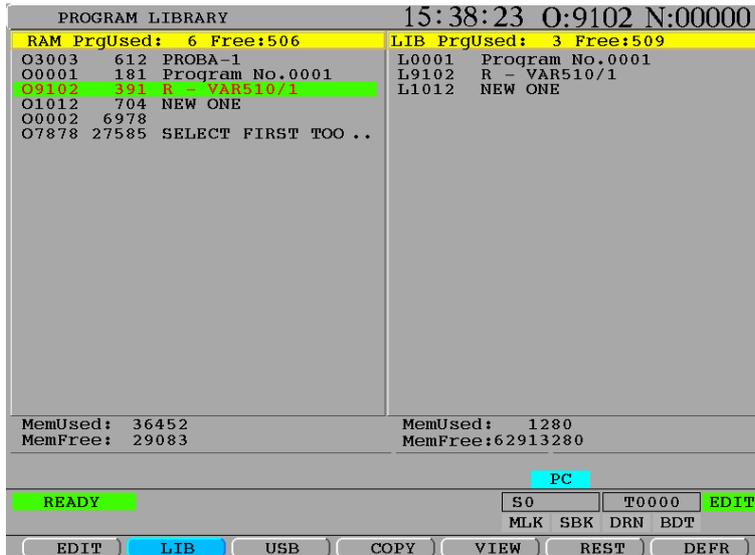
Machine



### 1.3. Automatic operation

#### •Program selection

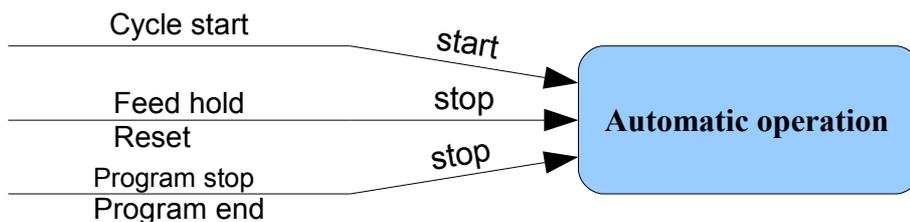
In EDIT mode select the program used for the corresponding workpiece. Ordinarily, one program is used for a workpiece. If two or more programs are stored in memory, select the program needed by searching the corresponding program number.



#### •Start and stop

Pressing the "cycle start" pushbutton causes automatic operation to 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 machine process is completed, automatic operation is stopped.

#### Start and stop for automatic operation



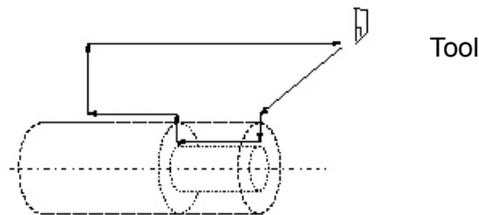
## 1.4. Testing a program

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. Checking 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.



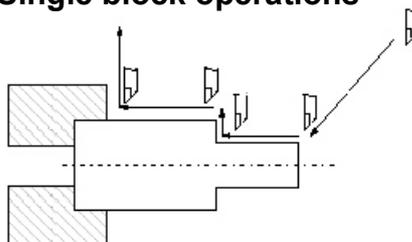
- **FEED OVERRIDE**

Check the program by changing the feedrate specified by the program.

- **SINGLE BLOCK execution**

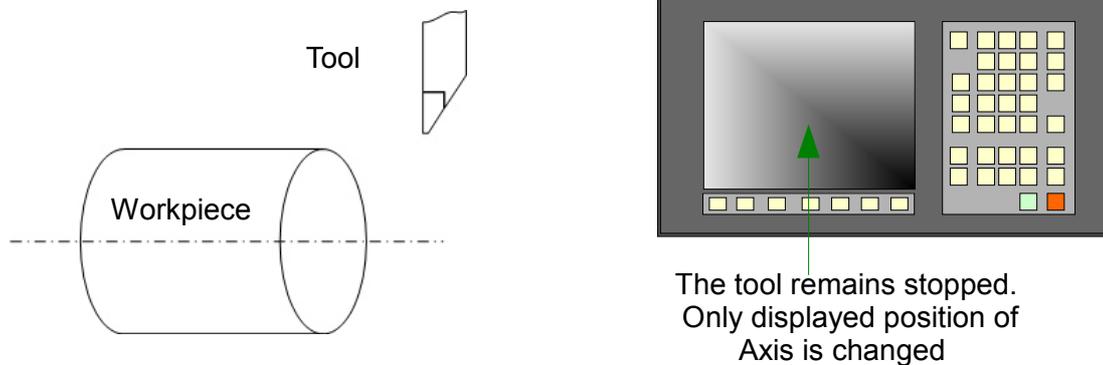
When the cycle start pushbutton is pressed, the tool executes only one operation and stops afterwards. By pressing the cycle start again, the tool executes one more operation and then stops. The whole program can be checked in this manner.

#### Single block operations



## 1.4.2. Position display change without running the machine

### • MACHINE LOCK.



***CAUTION:*** Once this function is used, the coordinate system is shifted. For this reason, before starting the machine 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 realization depends on the machine builder.

## 1.5. Editing a part of the program

Once the created program is loaded into memory, it can be corrected or modified using the TFT/MDI panel. This operation can be executed in "edit" mode.



- Offset value

## Displaying and setting Offset Value

GEOM OFFSETS					12:45:48 O:0000 N:00000	
No.	X	Z	R	T		
▶01	112.300	26.300	0.000	1		
02	45.000	32.600	0.000	6		
03	99.000	8.300	0.000	5		
04	0.000	0.000	0.000	3		
05	0.000	0.000	0.000	0		
06	0.000	0.000	0.000	0		
07	0.000	0.000	0.000	0		
08	0.000	0.000	0.000	0		
09	0.000	0.000	0.000	0		
10	0.000	0.000	0.000	0		
11	0.000	0.000	0.000	0		
12	0.000	0.000	0.000	0		

ABSOLUTE	DISTANCE	CURRENT	MODAL G
X 0.000	X 0.000		G00 G21 G97 G40 G25
Z 0.000	Z 0.000		G69 G22 G99

PC	
READY	S4 T0000 MDI
MLK SBK DRN BDT	
OFSW	OFSG WORK V100 V500

The tool has its corresponding dimensions (length, diameter). When a workpiece is machined, the tool movement 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 an example, the following data can be set:

- Offset values;
- Variables;
- Measurement units inch / mm;

The above data is called setting data (SETTINGS)

SETTINGS		17:37:56 O:7878 N:00000	
SETTINGS STATUS			
▶ PRM MODIFY	= 0	(0-OFF 1-ON )	
PRM RELOAD	= 0	(0-OFF 1-ON )	
INPUT UNIT	= 0	(0-MM 1-INCH)	

PC	
READY	S0 T0001 MDI
MLK SBK DRN BDT	
SET	PRM DGN LAD AXS

## • Setting and displaying parameters

The CNC functions have versatility in order to be used for different kinds of machines.

For example, CNC parameters can specify the following:

- Rapid traverse rate along each axis
- Input metric/inch system
- Cutting feedrate
- Backlash compensation and etc.

Data specifying these characteristics are called **parameters**

Note: shown parameters may be different on each machine.

PARAMETERS		13:14:02 O:0000 N:0000			
No.	VALUE				
0500	1	0520	0	0540	4500
0501	1	0521	0	0541	4500
0502	1	0522	0	0542	4500
0503	1	0523	0	0543	30
0504	1	0524	0	0544	0
0505	1	0525	0	0545	0
0506	1	0526	0	0546	0
0507	1	0527	15000	0547	0
0508	0	0528	0	0548	0
0509	0	0529	10	0549	0
0510	0	0530	10	0550	0
0511	0	0531	0	0551	0
0512	0	0532	0	0552	0
0513	0	0533	400	0553	0
0514	0	0534	200	0554	0
0515	0	0535	100	0555	4500
0516	1000	0536	0	0556	0
0517	3000	0537	0	0557	0
0518	0	0538	0	0558	0
0519	0	0539	0	0559	0

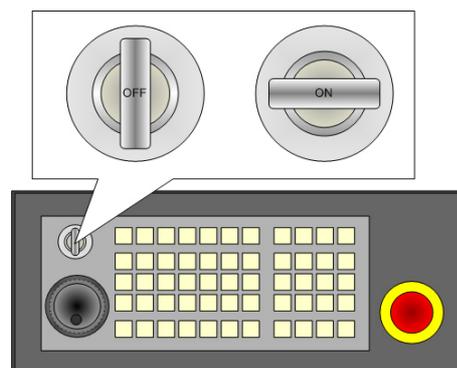
PC

READY S4 T0000 MDI  
MLK SBK DRN BDT

SET PRM DGN LAD AXS SIM

## • Data protection key ( PROTECT KEY ).

A key, called data protection key is available on the operator's panel. It is used to prevent parameters or part of programs from erroneous loading, modification or deletion.



## 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 13:52:35 O:0733 N:00019
CURRENT CNC PROGRAM
O0733 ;
N05 G50 X5. Z42. ;
N010 G50 Z160.0 X220. ;
N012 G73 U14.0 W14.0 R5 ;
N013 G73 P014 Q019 U2. W1. F300 ;
N014 G00 X80.0 W-40.0 ;
N015 G01 W-20.0 F150 S600 ;
N016 X120.0 W-10.0 ;
N017 W-20.0 S400 ;
N018 G02 X160.0 W-20.0 R20.0 ;
N019 G01 X180.0 W-10.0 S280 ;
N020 G70 P014 Q019 ;
M99 ;
%
```

PC

READY S4 T0000 EDIT

MLK SBK DRN BDT

EDIT LIB USB GHLP CMNT

```
PROGRAM LIBRARY 15:38:23 O:9102 N:00000
RAM PrgUsed: 6 Free:506 LIB PrgUsed: 3 Free:509
O3003 612 PROBA-1 L0001 Program No.0001
O0001 181 Program No.0001 L9102 R - VAR510/1
O9102 391 R - VAR510/1 L1012 NEW ONE
O1012 704 NEW ONE
O0002 6978
O7878 27585 SELECT FIRST TOO ..
```

MemUsed: 36452 MemUsed: 1280  
MemFree: 29083 MemFree:62913280

PC

READY S0 T0000 EDIT

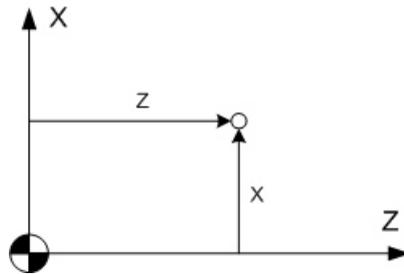
MLK SBK DRN BDT

EDIT LIB USB COPY VIEW REST DEFR

## 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

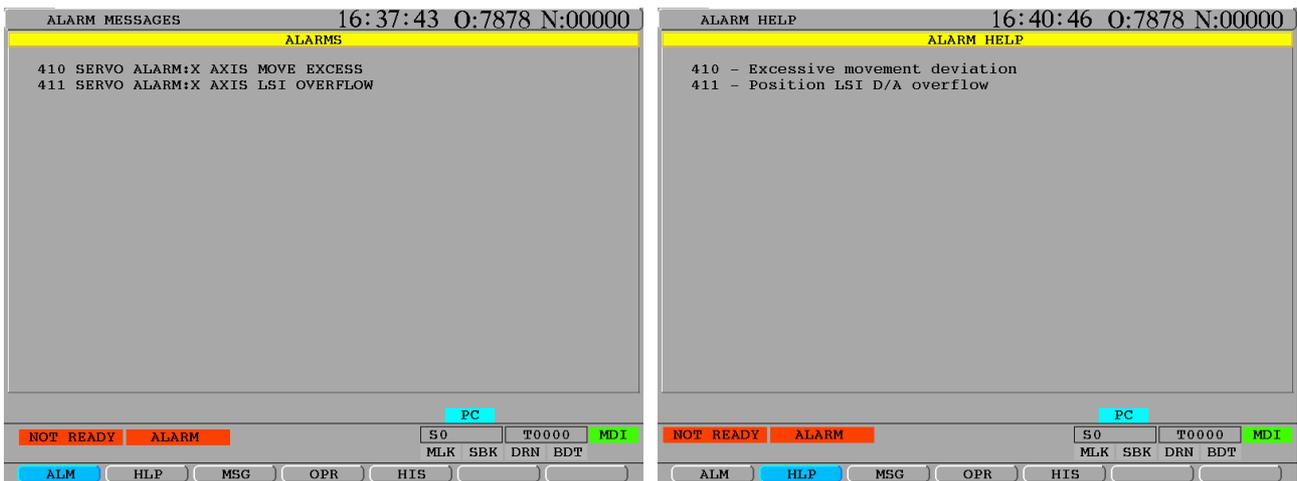


POSITIONS (ABSOLUTE)		12:24:33 O:0000 N:00000	
X	37.750 mm	RELATIVE	
		X	0.000
Z	10.000 mm	RAPID OVR:	
		Z	0.000
F 0.0000 mm/min		FEED OVR:	
S 0 rpm		100%	
G R 1		MACHINE	
ALARMS & MESSAGES		X	0.000
		Z	0.000
		SPNDL OVR:	
		100%	
		MPG OVR:	
		* 10	
		DISTANCE	
		X	0.000
		Z	0.000
		CYCLE TIME:	
		00:00:54	
		PARTS COUNT:	
		1	
PC			
READY		S4	T0000 AUTO
		MLK	SBK DRN BDT
ABS		REL	ALL

In addition, number of parts, cycle time and real time is also displayed.

## 1.7.3 Alarm display

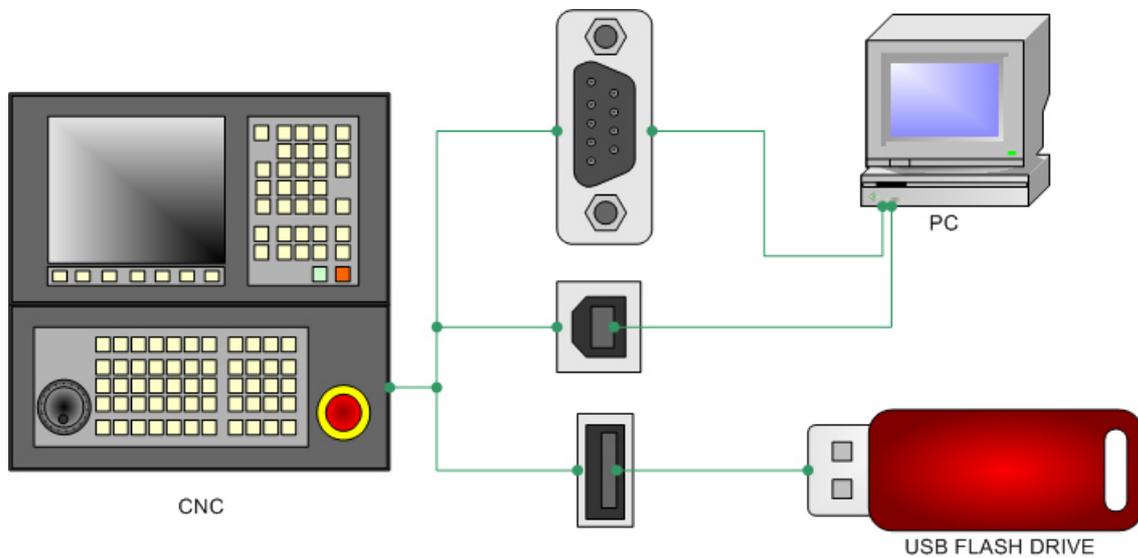
When a trouble occurs during operation, error code and the alarm message are displayed on the TFT screen. See the appendix for the list of error codes and their meanings. It is included a short description of each error code.



Using buttons PgUp and PgDn from MDI can switch between error code and it's description.

### 1.8. Data output.

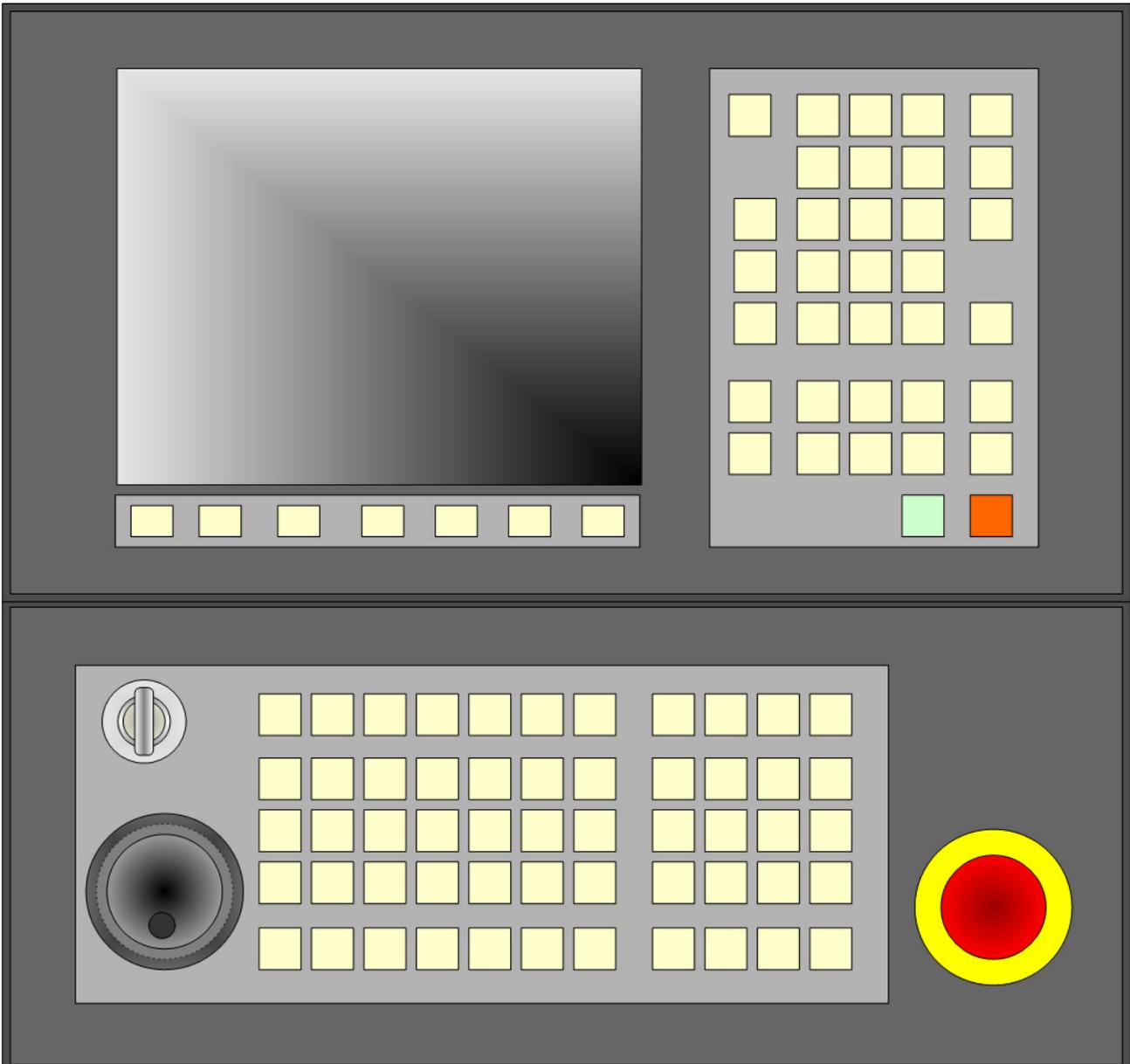
Programs, offset values, parameters, etc. Entered in CNC memory can be output to an other device via RS232C, USB or USB Storage Flash.



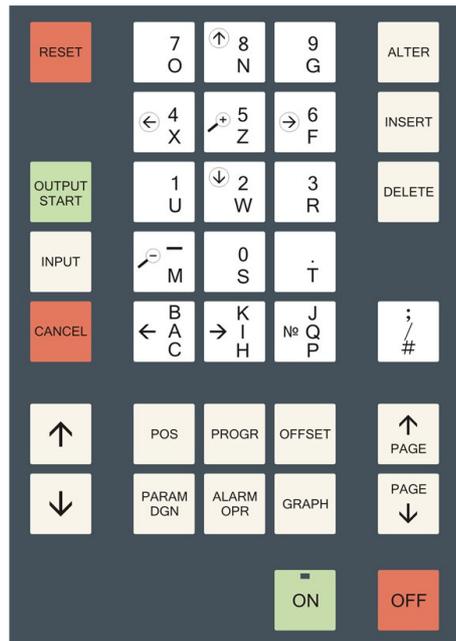
## 2. Peripheral devices

The peripheral devices available include TFT/MDI panel connected to CNC, machine operator's panel and external input/output devices such as PC or USB Flash Drive.

### 2.1. TFT / MDI panel



## MDI keyboard



### **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 application. Their functions are displayed at the bottom of TFT 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 TFT display. 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 ] – Alternation

[ 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:

Cursor UP – this key is used to move the cursor in forward and downward direction.

Cursor DOWN – moves cursor in a reverse and upward direction

## Page change keys

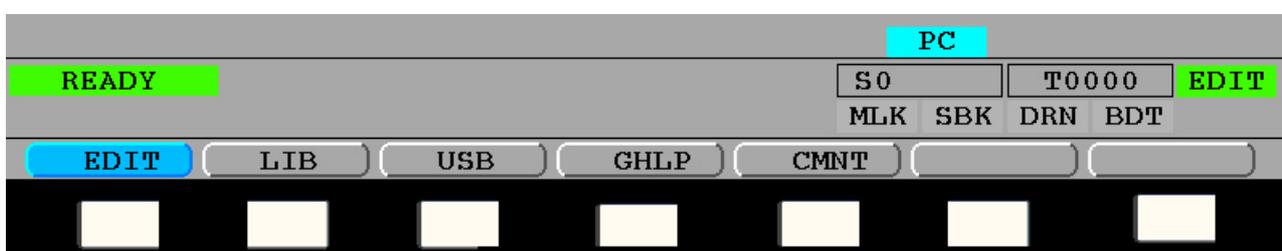
Page UP – this key is used to changeover the page on the screen in the forward direction.

Page DOWN – used to changeover the page on the screen in the reverse direction.

## 2.2. Functional and soft keys

### 2.2.1. General screen operations

1. Press a functional key on the 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 screen 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 ]

[ OFFSET ] - Press this key to display the offset screen.

[ DGNOS ]

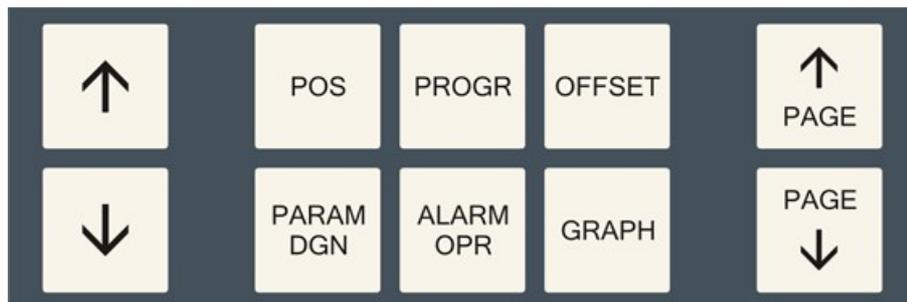
[ PARAM ] - Press this key to display the parameter/diagnostic screen.

[ OPR ]

[ ALARM ] - Press this key to display the alarm screen.

[ AUX ]

[ GRAPH ] - Press this key to display the graphic functions of the system.



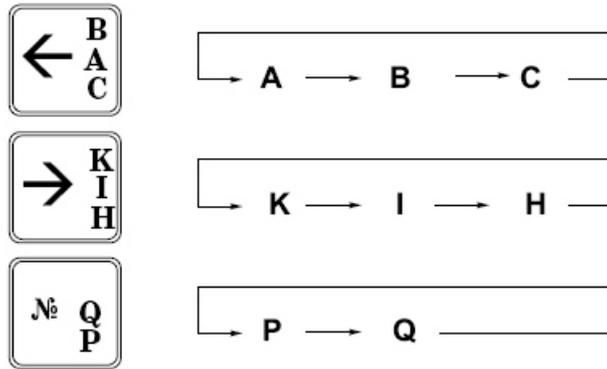
## 2.2.3. Key input and input buffer

### • 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 the TFT 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 but 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 [DELETE] key deletes only the last input symbol.

### 2.3. External I/O devices

CNC system 20 provide asynchronous serial interface RS-232C, USB serial interface (RS emulator), USB flash disk.

The following devices can be connected:

- **Personal computer with RS-232 or USB**
- **USB Flash Drive**

The following data types can be input/output to or from CNC

Programs

Offsets

Parameters

Variables

Diagnostics (work zone variables for PMC-X) ( DGN 300 ÷ DGN G99 )

The communication protocol is well known Z-modem protocol, which ensures error free data transfer. Transfer rate can be set by parameter (PRM No 120)

Values:

0 = 115200 bps (default)

4= 4800 bps

9 = 9600 bps

19 = 19200 bps

57 = 57600 bps

115 = 115200 bps (same as 0)

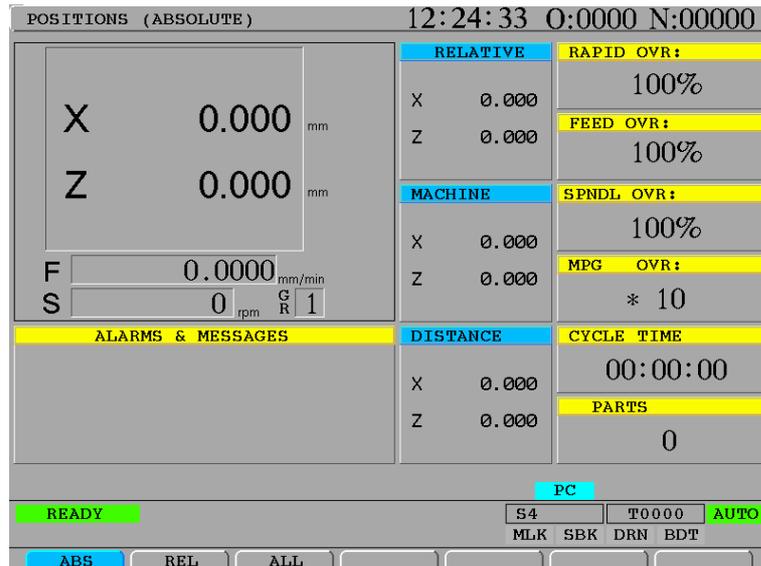
Protocol for PC must be Z-modem without crash recovery, flow control X-on/X-off.

## 2.4. Power ON/OFF

### 2.4.1 Turning the power ON.

#### Procedure of turning the power ON

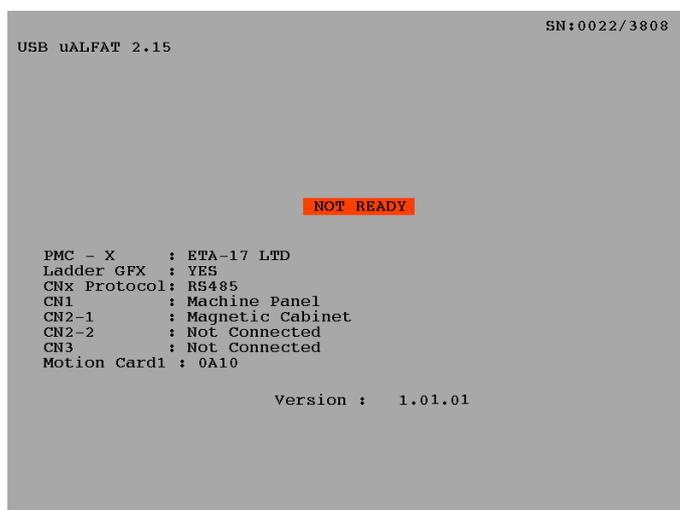
1. Check the appearance of the CNC machine (for an example, check whether the front and rear doors are closed)
2. Turn the power ON according to the manual issued by the machine builder.
3. After the CNC is on, check whether the position screen is displayed.



#### WARNING:

When pressing the <POWER ON> key, do not touch any other keys on the TFT/MDI panel, until the positional or alarm screen is displayed. Some of the keys are used for maintenance or have special operation purpose. When they are pressed at start up, unexpected operation may be caused.

### 2.4.2. Display of the software configuration



### **2.4.3 Power OFF**

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 devices are connected, disconnect them first.
4. Push the <POWER OFF> button.

#### NOTES:

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:

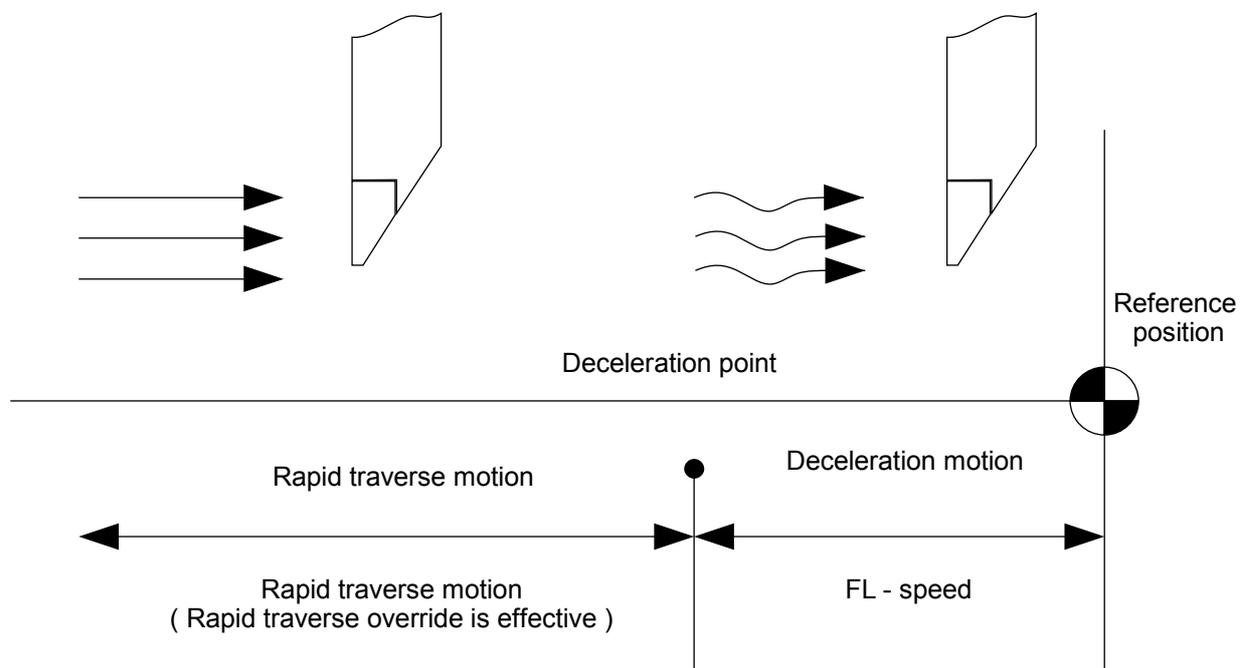
1. Manual reference position return.
2. JOG feed.
3. Incremental feed.
4. Manual handle feed.

#### 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 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 on machine panel.
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. Press the button and wait until tool returns to the reference position. To the deceleration point the tool is moved at rapid traverse 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.

## **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 is made. If **a**, **b**, and **c** 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  $X=a$ ,  $Z=b$ ,  $3th=c$  after reference position return.

## **RESTRICTIONS :**

### **• Moving the tool after reference 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 changed.

### **• 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 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:

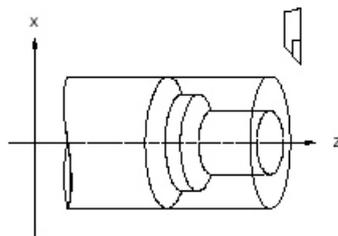
Rotary switch position		Rotary switch position	
Metric input [mm/min]	Inch input [inch/min]	Metric input [mm/min]	Inch input [inch/min]
0	0	50	2,0
2,0	0,08	79	3,0
3,2	0,12	126	3,0
5,0	0,2	200	8,0
7,9	0,3	320	12
12,6	0,5	500	20
20	0,8	790	30
32	1,2	1260	50

The current jog feed can be viewed on the position screen (POS) when such an data is not displayed on the machine operator's panel

Note:

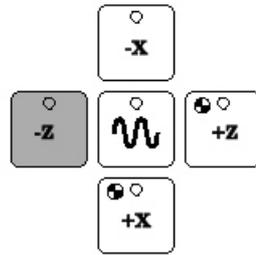
The feedrate accuracy is about 3%

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 jog feed button. Manual operation is allowed for one axis at a time.



While the button is pressed, the tool moves in a selected direction.

## PROCEDURE FOR JOG FEED



1. Press the JOG mode button – one of the mode selection buttons.
2. Press the feed axis and direction selection button to select the direction of moving the tool. While the button is pressed, the tool moves at a feedrate, specified in the table. The tool 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 is pressed. If a rapid traverse override is changed 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.

### RESTRICTIONS :

#### • **ACCELERATION / DECELERATION FOR RAPID TRAVERSE**

Feedrate, time constant and 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 traverse button does not activate rapid traverse but the jog feedrate. This function can be disabled by setting a parameter.

### **3.3. INCREMENTAL FEED ( STEP MODE ).**

In incremental (step) mode, pressing the feed axis and the direction selection button 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 present or not enabled.

#### **Procedure for incremental feed**

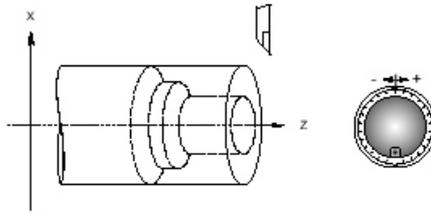
1. Press the step button – one of the mode selection buttons.
2. Select the distance for each tool movement.
3. Press the feed axis and the direction selection buttons. Each time this button is pressed, the tool moves one step. The feedrate is the same as the jog feedrate.
4. If the axis direction button 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 effective.

The above operations are just examples. For more details refer for the manual issued by the machine builder.

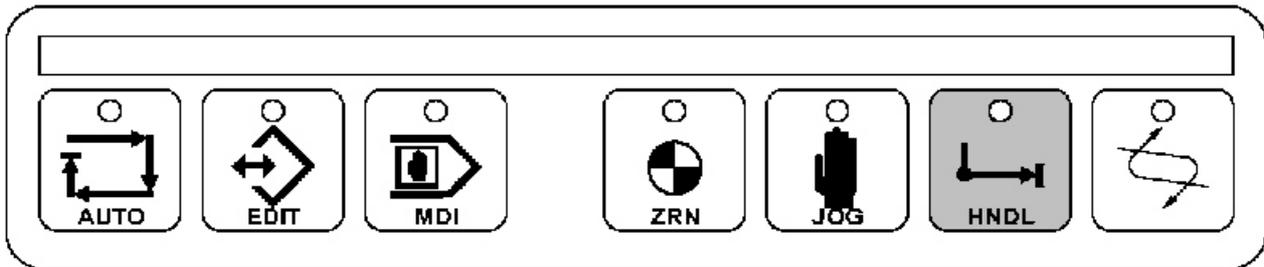
### **3.4. MANUAL HANDLE FEED ( HNDL MODE ).**

In the handle mode the tool can be moved by rotating 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. Also distance for one graduation can be 10 times or 100 times least input increment (or set by parameter).



## 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 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 of 100 graduations.

The above operations are just example. 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 MDI panel.

#### ***DNC operation.***

Function for operating a machine while reading a program from a 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 button 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 button 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 thread cutting cycle 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. 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 steps from ( 4 ) to ( 6 ).

### **Stopping and terminating memory operation**

Memory operation can be stopped using one of the following two methods:

- The stop command includes **M00** ( program stop ), **M01** ( optional stop ) and **M02** and **M30** ( program end )

#### Note:

Functions **M02** and **M30** returns the cursor at the beginning 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 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 machine builder.

### **Optional stop ( M01 ).**

Similar 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 or G32 are executed.

### **Reset**

The automatic operation can be stopped and the system can enter in reset state 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 ( function Block Delete ).**

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.

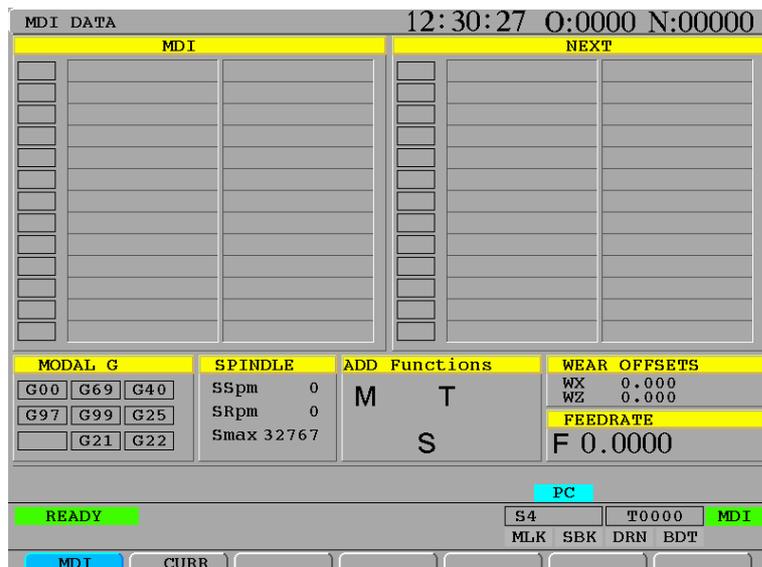
The following operations are given as an example. For more details refer to the manual, provided by the machine builder.

### **Procedure for MDI operation.**

Example:        **X10.0 Z200.5**

Only one command block can be entered from the **TFT /MDI** panel.

1. Press **MDI** button from mode select buttons.
2. Press the [ **PRGRM** ] button on TFT/MDI panel.
3. Press the soft key [ **MDI** ] to display a screen with MDI data at the top left.



4. Input „X10.0” by the address/ numeric keys.

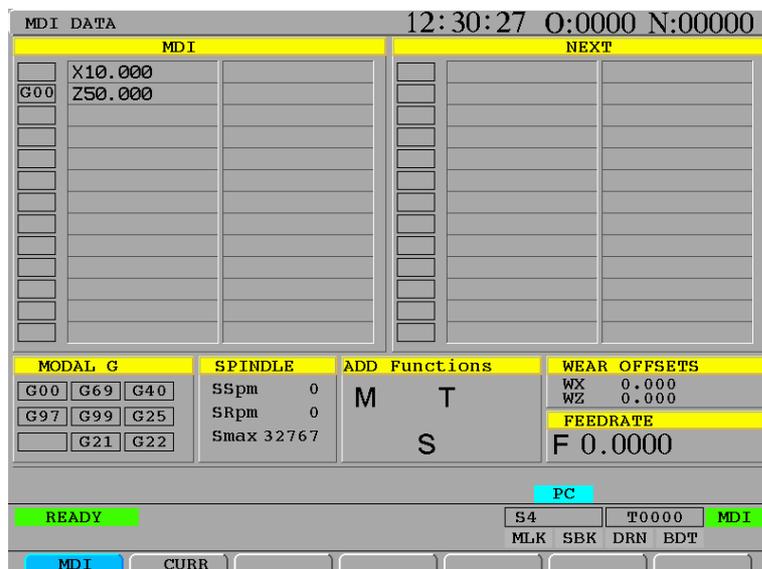
5. Press the [ INPUT ] key.

The data X and 10.0 is entered 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 „Z50.0” by the address/numeric keys.

7. Press the [ INPUT ] button.

The data Z and 50.0 is input and displayed.



8. Press the [ OUTPUT / START ] key or the "cycle start" button on the machine operator's panel (depending on the machine builder).

Pressing the [ RESET ] buffer clears the contents of the whole buffer.

## WARNING:

Modal G codes cannot be canceled. Refer 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 (;) must to be entered.
- A macro or subprogram call cannot be specified.
- In **MDI** operations, the screen **SETTINGS** determine whether the commands are absolute or incremental, no matter of G90 or G91 modal code.
- 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 bottom 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, registered in the CNC memory. However, repeat and brunch instructions cannot be specified.
- In **DNC** operation mode, to return to the main program from 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 can be used 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

- 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 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 implemented 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 commands 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.

### **5.2. FEED 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.

## RESTRICTIONS:

- **Override range**

The override can be specified ranges from **0%** to **150 %** by step of **10 %**.

### **5.3. RAPID OVERRIDE**

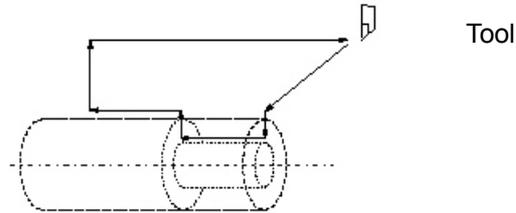
To the rapid traverse rate can be applied override by **F 0, 25%, 50% i 100%**.

Select one of the four overrides for the rapid traverse. For more information regarding the rapid traverse override refer to the manual provided by the machine builder. The following types of rapid traverse are available. Rapid traverse

1. Rapid traverse by G00.
2. Rapid traverse during a canned cycle.
3. Rapid traverse 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 in the spindle.



### 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

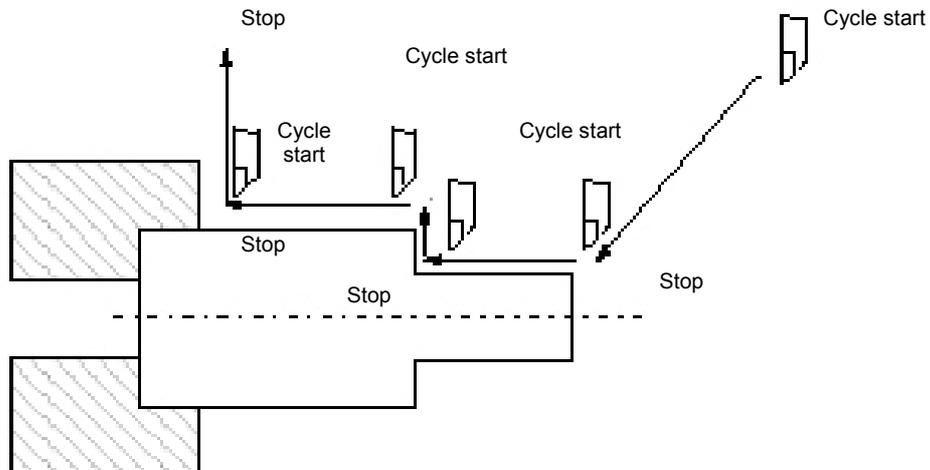
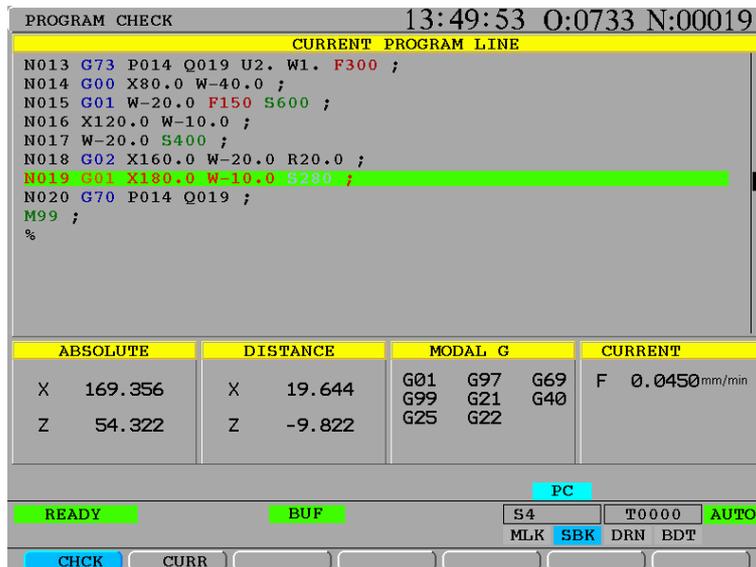
The dry run feedrate changes as shown in the table below according to the rapid traverse button and the corresponding parameter.

Rapid traverse button	Program command	
	Rapid traverse	Feed
ON	Rapid traverse	Jog maximum Feedrate
OFF	Jog feedrate or rapid traverse rate*	Jog feedrate

\* / Jog feedrate if the corresponding parameter P01.6 = 1, else – rapid traverse /

## 5.5. SINGLE 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 command is 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.

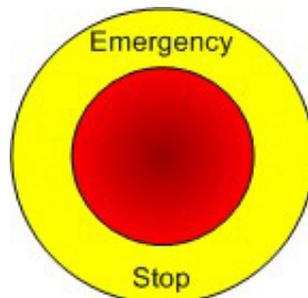
However, if the block contains an address other than O, N or P the single block stop is performed in a block containing M98P\_ M99, command.

## **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.

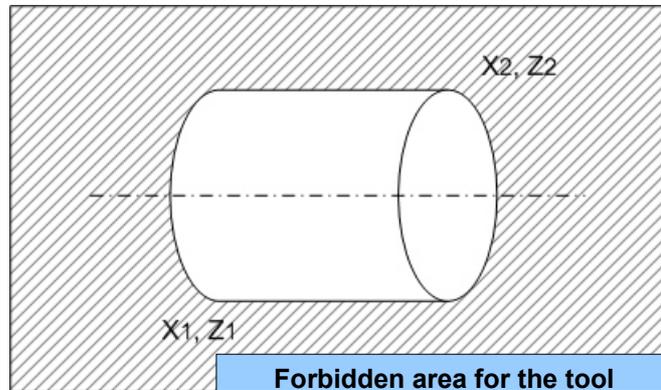


This button is self-locked when pressed. Although it varies depending on the machine builders, the button can usually be unlocked by twisting.

The Emergency Stop interrupts the current to the servomotors. 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 alarm is displayed, the tool can be moved in 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:

$$L \text{ mm} = F / 7500$$

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:

Numbers	Message	Contents
6n0	OVERTRAVEL + n	Exceeded the n-th axis (1-6) + direction
6n1	OVERTRAVEL - n	Exceeded the n-th axis (1-6) - direction

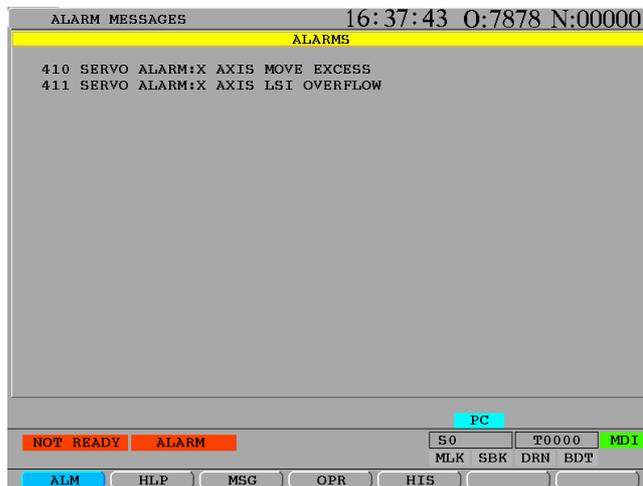
## 7. ALARMS AND SELF-DIAGNOSIS FUNCTIONS

When an alarm occurs, the corresponding alarm screen appears to indicate the cause of the alarm. The cause 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 self-diagnosis functions.

### 7.1. ALARM DISPLAY

- Alarm screen

When an alarm occurs, the following alarm screen appears:



- 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.

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 indicates 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:

- Nr 000 to 249: Program error
- Nr 250 to 254: I/O error
- Nr 300 to 399: Fatal error
- Nr 400 to 499: Servo alarms
- Nr 600 to 609: Overheat alarms
- Nr 610 to 699: Overtravel alarms

For more detail information regarding the alarms and their codes see the appendix.

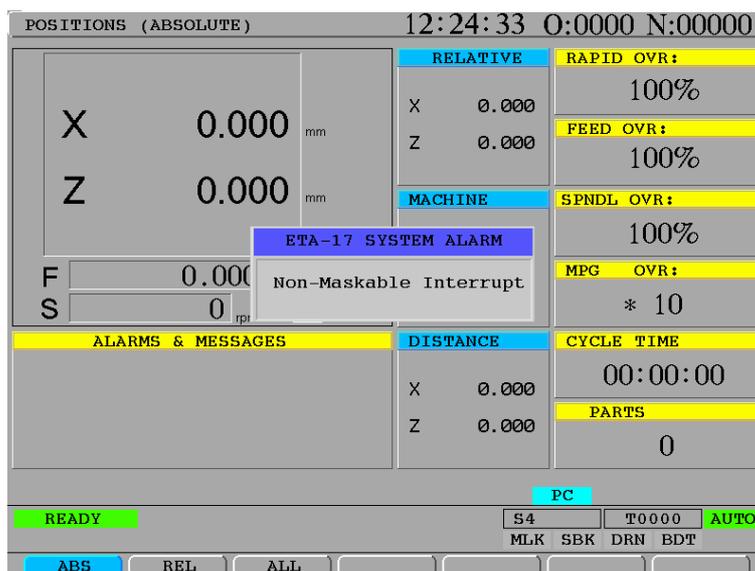
- **Error codes description in brief**

An error code screen is displayed by pressing the buttons [PgUp] or [PgDn]. It contains a brief description of the probable cause.

**WARNING:**

If a rimmed alarm without 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 – [PgUp], [PgDn].
  - (2) Press the [No] key.
  - Input the diagnostic number to be displayed.
  - Press the [INPUT] key.

DIAGNOSTIC		17:47:46 O:7878 N:00000	
SYSTEM DIAGNOSTIC			
700	SCT	ITL	OVZ
701	RST	INP	DWL
702	STP	REST	EMS
		RSTB	CSU
800	1000	810	9995
801	2000	811	10995
802	0	812	11995
803	0	813	621
804	0	814	621
805	0	815	0
820	1000	821	2000
822	3000	823	FFFF
824	FFFF	825	FFFF
830	0.505	831	1.166
832	1.349	833	0.000
834	0.000	835	0.000
900	0.000	902	0.000
901	0.000	903	0.000
		904	0.000
		905	0.000
		908	0.002
		909	0.000
<b>PC</b>			
<b>READY</b>		S0	T0101
		MLK	MDI
		SBK	DRN
		BDT	
SET	PRM	<b>DGN</b>	LAD
		AXS	

DGN No:	bit No:	#7	#6	#5	#4	#3	#2	#1	#0
700			CSCT	CITL	COVZ	CINP	CDWL	CMTN	CFIN

When the digit is „1” (light), 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: A in-position check is being executed.
- COVZ: Feed override is 0% ( feedrate 0 ).
- CITL: Interlock signal is on.
- CSCT: Speed arrival signal of spindle is waiting.

When automatic operation is refused, the cause is displayed.

DGN No:	bit No:	#7	#6	#5	#4	#3	#2	#1	#0
701				CRST					

CRST: One of the following: a signal from the [RESET] button on the MDI panel, emergency stop or reset from external unit.

Indicated automatic operation stop or feed hold status. Used for troubleshooting.

DGN No:	bit No:	#7	#6	#5	#4	#3	#2	#1	#0
702		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, TEACH IN HANDLE.
- Other alarm is generated.

REST: This signal is set when external reset, emergency stop or reset button is on.

EMS: This bit is set when emergency stop is set on.

RSTB: This bit is set when the reset button is on.

CSU: This is set when the emergency stop is turned on or when servo alarm has been generated.

DGN 0800..0806: - Current error in servo contour along each axis.

DGN 0820..0826: - 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 USB Flash drive.

#### Procede 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 protect key to "unlocked".
4. Press the [ **PRGRM** ] button to display the program screen [ **LIB** ] - "Program Library".
5. Press the [ **INPUT** ] button.

The program is stored with the number written in it.

#### UWAGA:

To abandon the input mode, press the [ **RESET** ] key.

#### • Multiple program 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:

<i>Numbers</i>	<i>Description</i>
70	The size of memory is insufficient to store the whole program
72	Too many programs in 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 USB flash drive.

#### Procedure for program output

1. Press the [EDIT] mode button on the machine operator's panel.
2. Press the [PRGRM] button to display the program screen.
3. Press the functional button [ LIB ] and use cursor buttons to select desired program.
4. Press the [OUTPUT/START] button to send program to PC or USB flash.
5. In case of PC – use receive function of DNC program or your terminal program for transfer.

If button [OUTPUT/START] button is pressed until the [ALTER] button is kept pressed, all programs from memory will be transferred. The programs can be copied from CNC RAM to CNC FLASH library. To do this just use functional button [COPY] when desired program is selected. The program will be copied from RAM to Flash library or from Flash library to RAM, depending on selected program – if chosen program is in RAM it will be copied to Flash, else – it will be copied from flash to RAM.

## 8.2. OFFSET DATA INPUT AND OUTPUT.

### 8.2.1. Offset data input.

The offset data is loaded into the memory of CNC using serial connection from PC or from USB Flash drive. The input format is the same as the offset value output.

#### Procedure for offset data input.

1. Make sure that 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 ( **USB Flash** ).

## Procedure for offset data output.

1. Make sure that 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 [ OUTPUT / START ] button.

The output format is as follows:

**G10 P00001 X\_\_Z\_\_Q\_**

- P** – number of correction.
- X** – offset X.
- Z** – offset Z.
- Q** – correction type.

## 8.3. Parameters input and output.

### 8.3.1. Parameter input.

Parameters are loaded into the memory of CNC using serial connection from a PC or from USB Flash drive. 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 value replaces the existing one.

1. Make sure that the device is ready to transmit.
2. Press the function button [ **DGNOS /PARAM** ] on the MDI to display the "Settings" screen.
3. Enter '1' in parameter change value (PRM MODIFY). Now parameters can be changed and alarm No 100 appears.
4. Press the soft key [ **PRM** ] to display the "PARAMETERS" screen.
5. Press the [ **INPUT** ] button.  
Parameters are transferred and stored in memory. After the operation is completed, the screen message "DATA IMPORT" disappears.
6. Use button [DGNOS/PARAM] to display the "SETTINGS" screen and enter '0' in (PRM MODIFY) switch.
7. Restart CNC.

### 8.3.2. Parameters output.

Parameters can be transferred from CNC to PC via serial port (RS232 or USB).

#### Procedure for parameters output.

1. Make sure that device is ready to receive data.
2. Press the [EDIT] mode button on the machine operator's panel.
3. Press the functional button [DGNOS/PARAM] to display the parameter screen and then press [OUTPUT/START] button.

Output format:

N \_ P \_ ; where N\_ is parameter number, P\_ - parameter value

Parameters can be stored in nonvolatile memory (EEPROM) in CNC and restore them if needed. To store parameters from RAM to EEPROM use [DGNOS/PARAM] button in MDI mode to display the settings screen and press [INSRT] button. To restore parameters set value of (PRM MODIFY) to '1' and restart CNC.

## 8.4. Custom macro variables input/output

### 8.4.1. Custom macro variables input

The values of custom macro variables ( #100 .....#131 i #500 ....#531 ) are loaded in the CNC memory using serial connection from a PC. 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 variables input procedure

1. Make sure that the device is ready to transmit.
2. Press the [ **EDIT** ] 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 i #500 ....#531 ) can be input or output. Variables from #100 to #131 do not save their values after power is off.*

## 8.4.2. Custom macro variable output

The values of the custom macro variables ( #100 .....#131 and #500 ....#531 ) can be transferred from the CNC memory to PC via serial interface.

### Custom macro common variables output procedure

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 [ OFFSET ] to display the "VARIABLES" screen.
4. Press the [ OUTPUT / START] button. Variables are transferred from CNC memory to a PC.

### Output format

The output format is as follows:

N \_ V \_

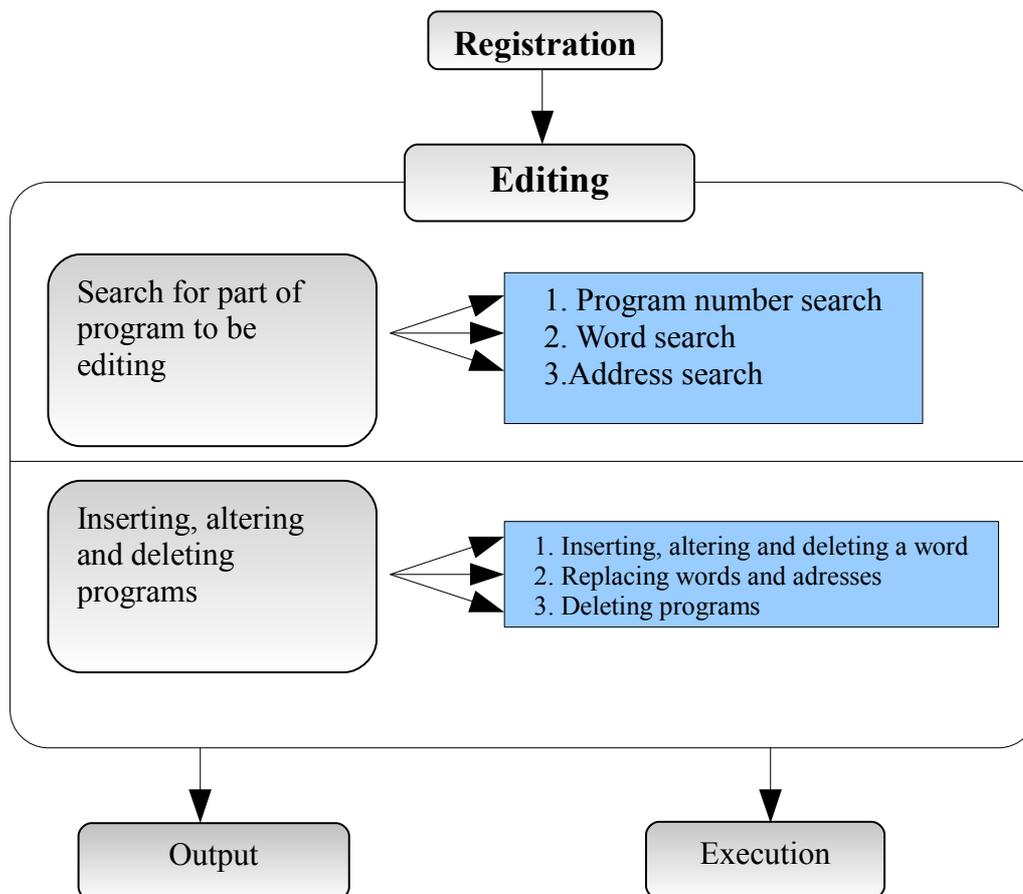
*where:*

N \_ : Variable number

V \_ : Variable 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 program.



## 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 a words

1. Select EDIT mode.
2. Press the functional button [ PRGRM ] to display the program screen.
3. Select a program to be edited.
4. Search the word to be modified by:
  - 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 is 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.
- [ **DELETE** ] 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 operation such as single block, stop or feed hold operation. If a 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 (press [RESET] button) upon completion of editing, before executed the program.

### 9.1.1. Word search.

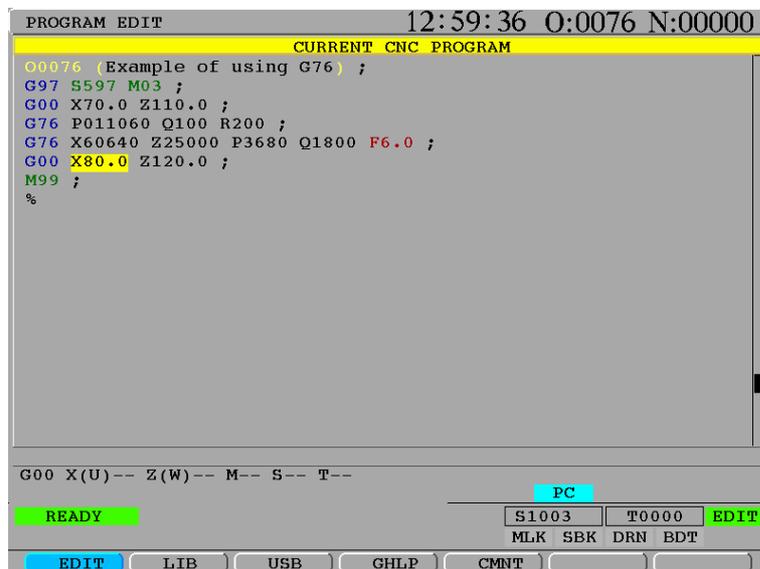
A word can be found by moving the cursor throughout all the words in the text (scanning), by word search or by address search.

#### 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.
3. Holding down the cursor keys [↓] or [↑] scans consecutively all the words in the program.
4. Pressing the page key [Pg ↓] displays the next page and searches for the first word of the page.
5. Pressing the page key [Pg ↑] displays the previous page and searches for the first word of the page.

#### Example:

When **X80** is scanned.



## Procedure for searching a word

### Example:

Searching for **M28**

1. Input address [M28].
2. Pressing the cursor key [↓] starts the search operation. Upon completion of the search operation, the cursor is positioned over M28 if this word is present in the program or alarm may occurs. Pressing the cursor key [↑] perform search operation in the reverse direction (to the beginning of a program).

### **Alarm:**

Alarm number 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. To do this, press the [RESET] button when editing the program. 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**

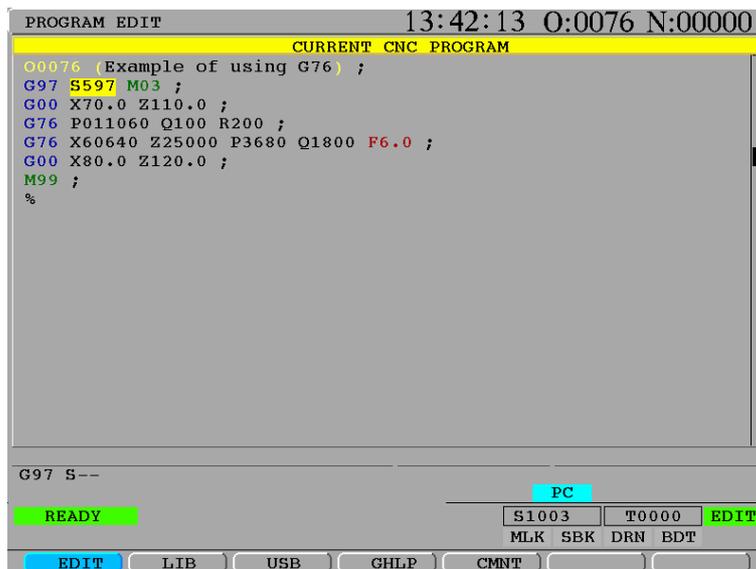
#### **Procedure for inserting a word**

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.

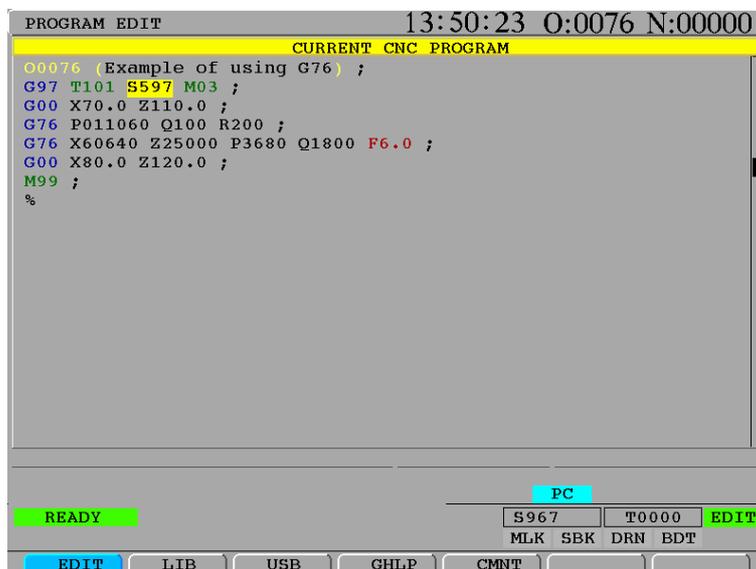
### Example:

Inserting **T101**

1. Input S597. Then search for S597.
2. Input [T] [1] [0] [1].
3. Press the [ **INSRT** ] button.



T101 is inserted.



#### 9.1.4. Altering a word

##### Procedure for altering a word

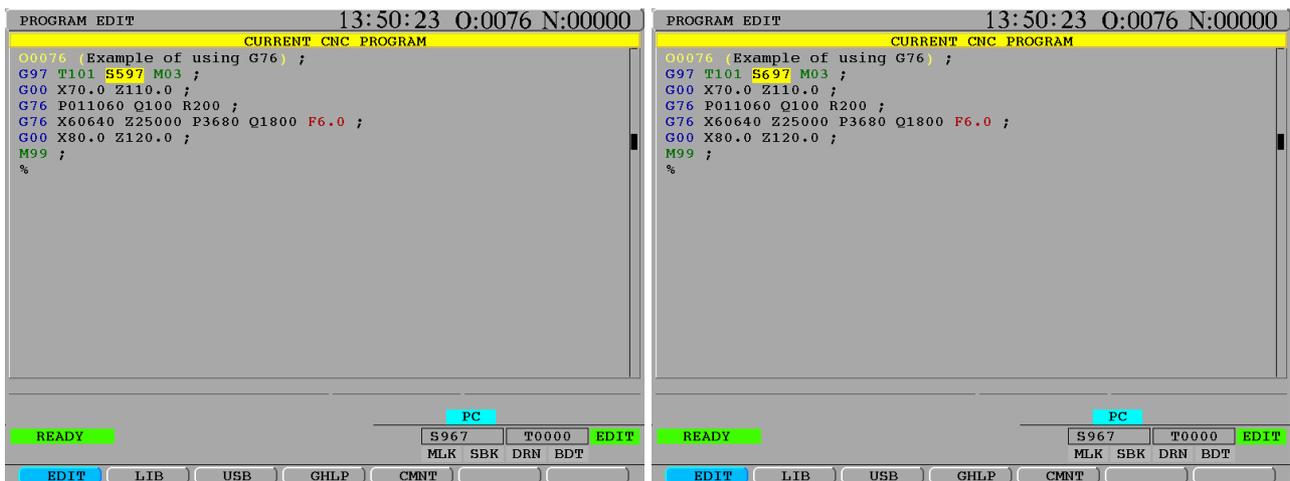
1. Find the word to be altered.
2. Input the address.
3. Input data.
4. Press button [ ALTER ].

Example:

Modifying S597 to S697.

1. Find S597.
2. Input [S] [6] [9] [7].
3. Press button [ ALTER ].

S597 has been changed to S697.



### 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 [ DELET ] 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 **S597**.

1. Find **S597**.
2. Press button [ DELET ] **twice**.
3. S597 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.

#### Method 1.

1. Select mode EDIT.
2. Press the address button [O] and write searched number.
3. Press the cursor button [↓].
4. Upon completion of search operation the cursor is positioned over selected program. If such a program has not been found, an alarm occurs.

#### Method 2.

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 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

#### 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. Select desired program to delete using cursor buttons.
4. Press the [ DELET ] button. Selected program will be 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”.
- Press and hold [ALTER] button and twice press [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 [ **PROGRAM** ] and the soft button [LIB] to display the PROGRAM LIBRARY screen.
3. Input the new program number (up to 4 digits).
4. Press button [ **INSRT** ].  
The program has been created.

#### **ALARM**

Alarm number 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 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 function. This chapter also describes procedures for selecting the desired information by the soft keys.

- **Screen transition chart.**

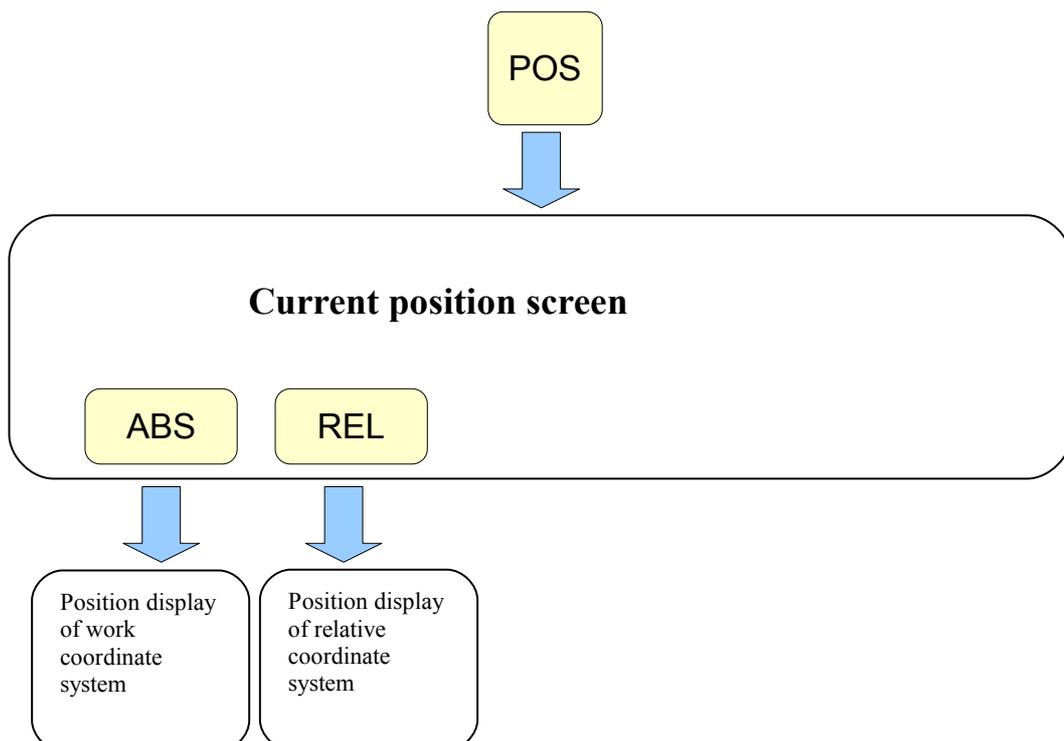
The screen transition for when each functional key on the MDI panel is pressed is shown below. The subsections referenced for each screen are also shown. For details on each screen and the setting procedures, see the appropriate subsections.

- **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.

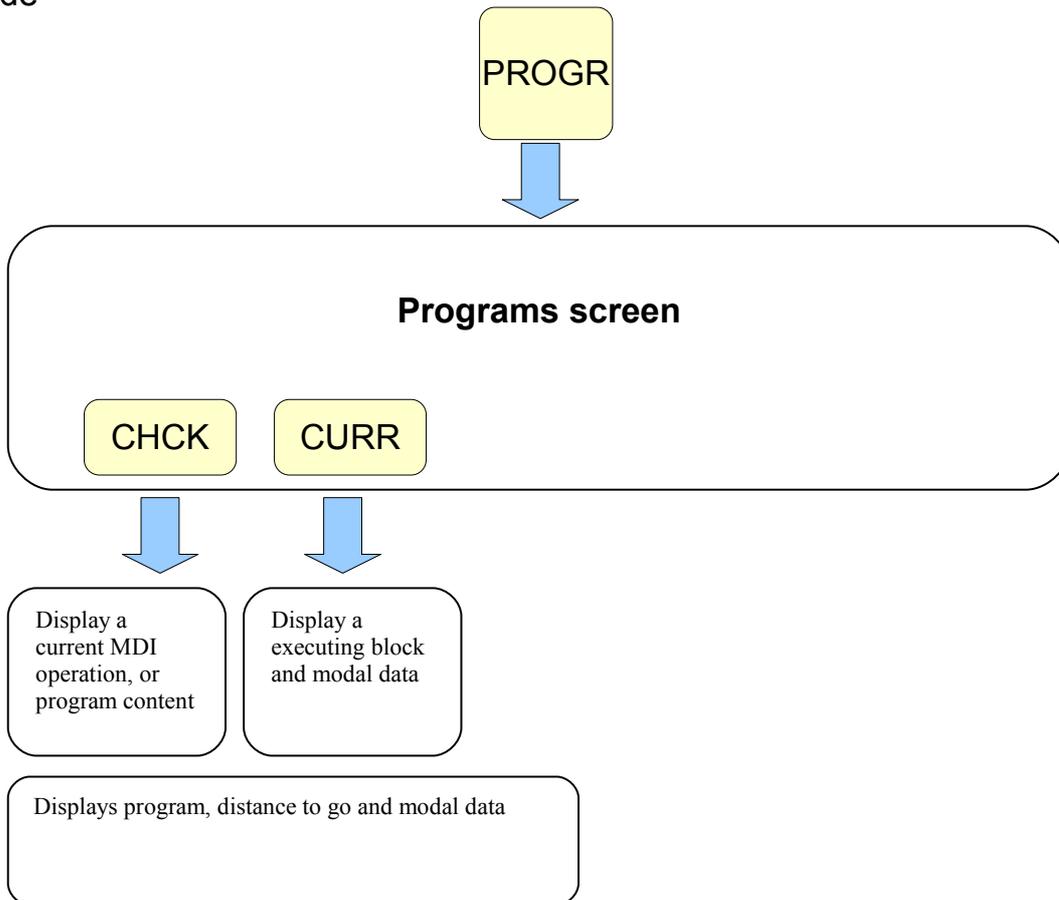
### Position display screen.

Screen transitions triggered by the function key [ **POS** ].

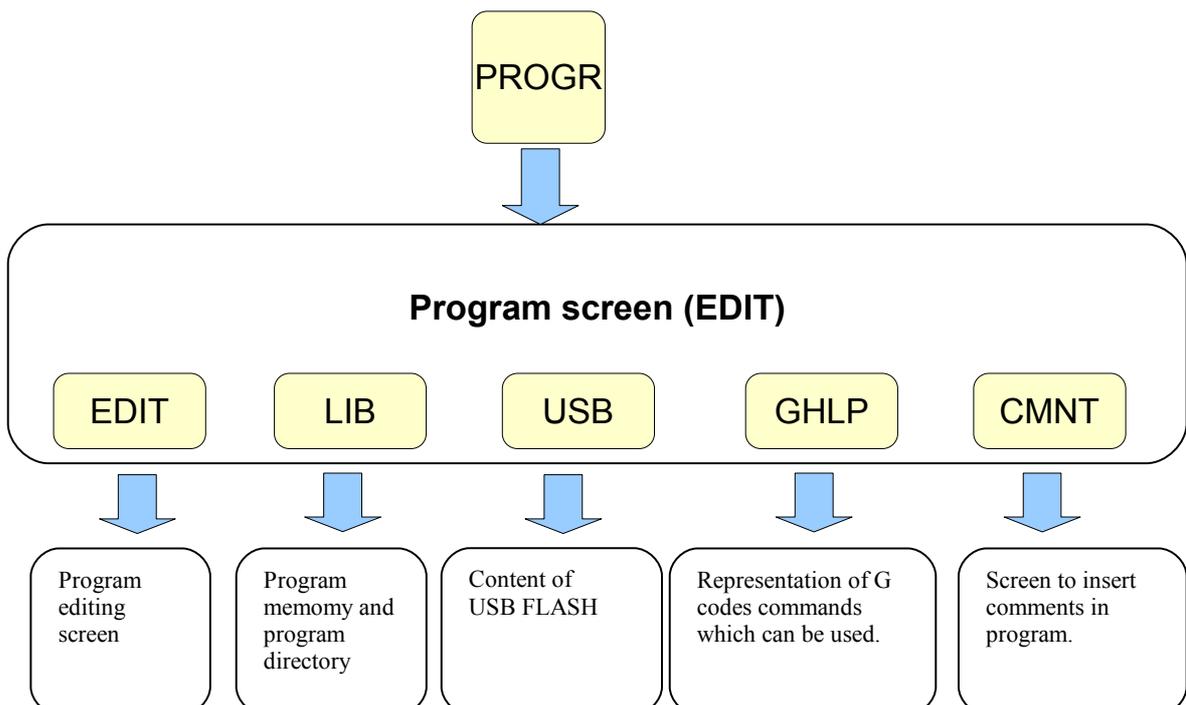


## Program screen

Screen transition triggered by the functional key [ PROGR ] in AUTO or MDI mode

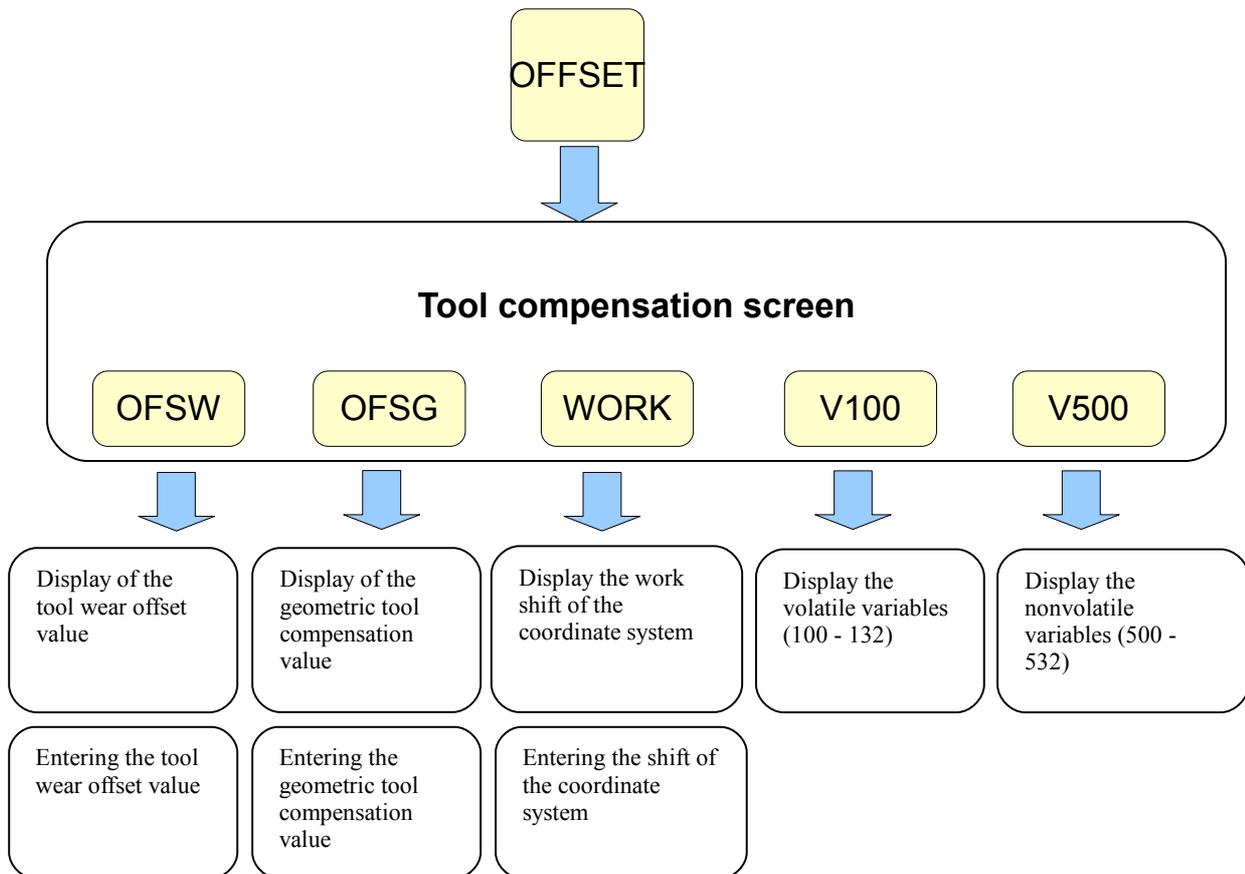


Program screen: screen transition triggered by the functional key [ PROGR ] in EDIT mode



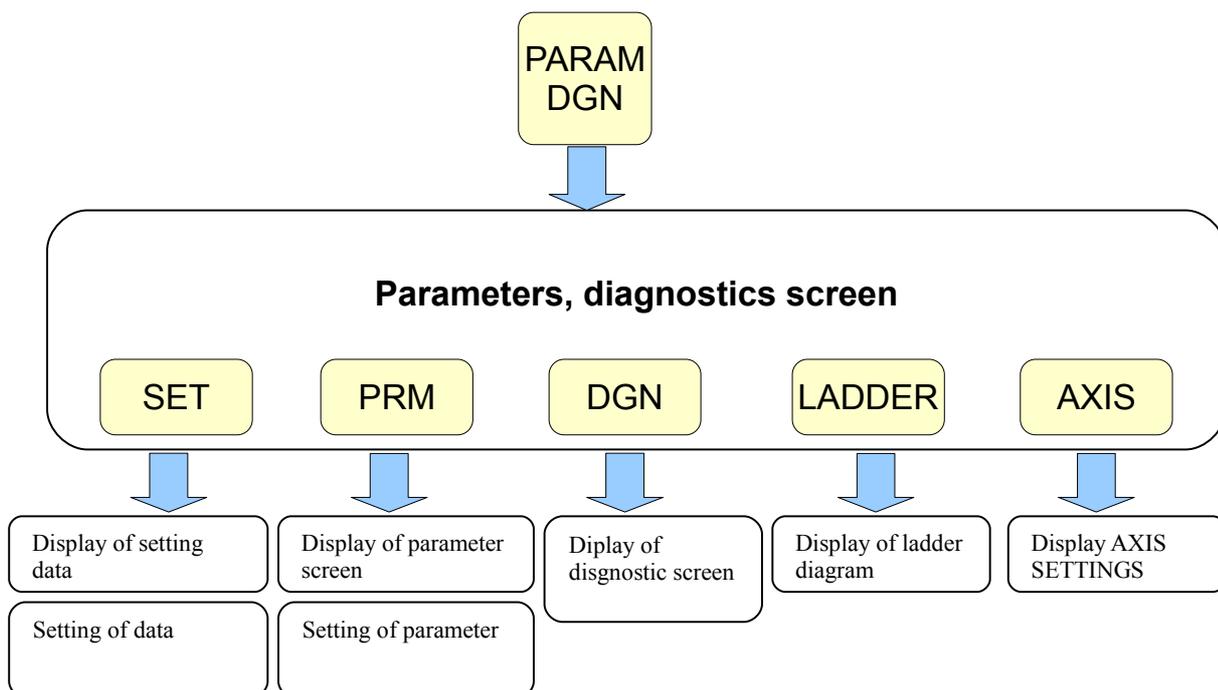
## Offset screen

Screen transition triggered by the functional key [ OFFSET ]



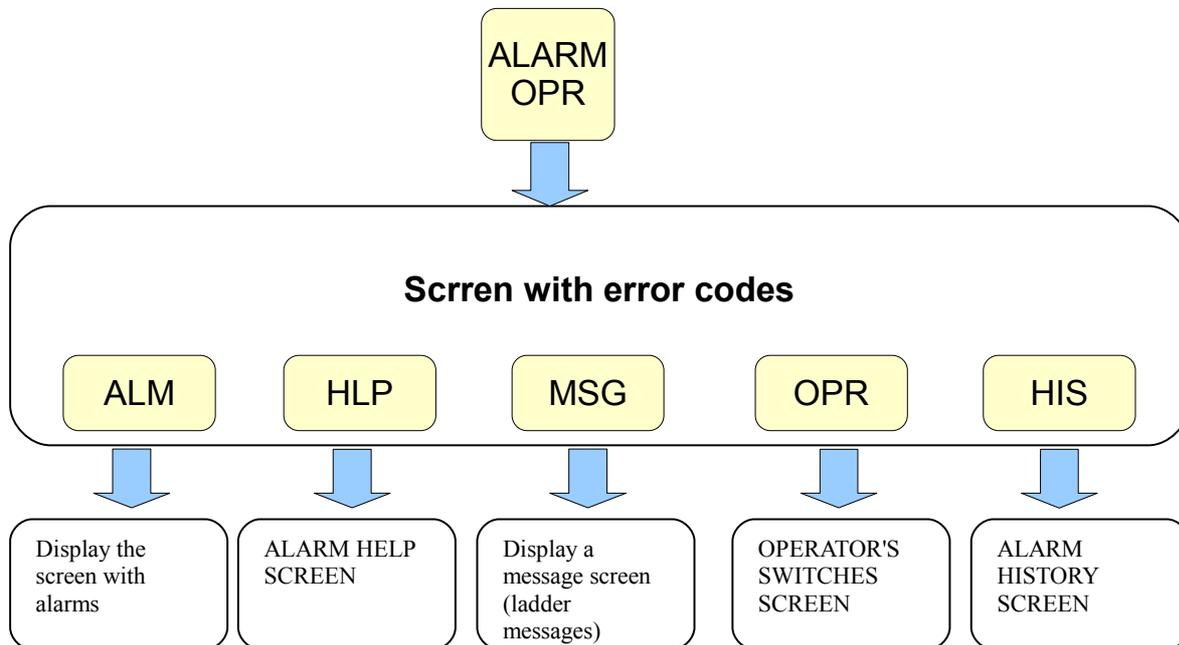
## Parameter/diagnostic screen

Screen transition triggered by the functional key [ PARAM / DGN ].



## Alarm screen.

Screen transition triggered by the functional key [ ALARM / OPR ].



## •Setting screen

The table below lists the data set on each screen

No	Setting screen	Content 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 macrom common variables ( #100 to #132 ) or ( #500 do #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 two 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.

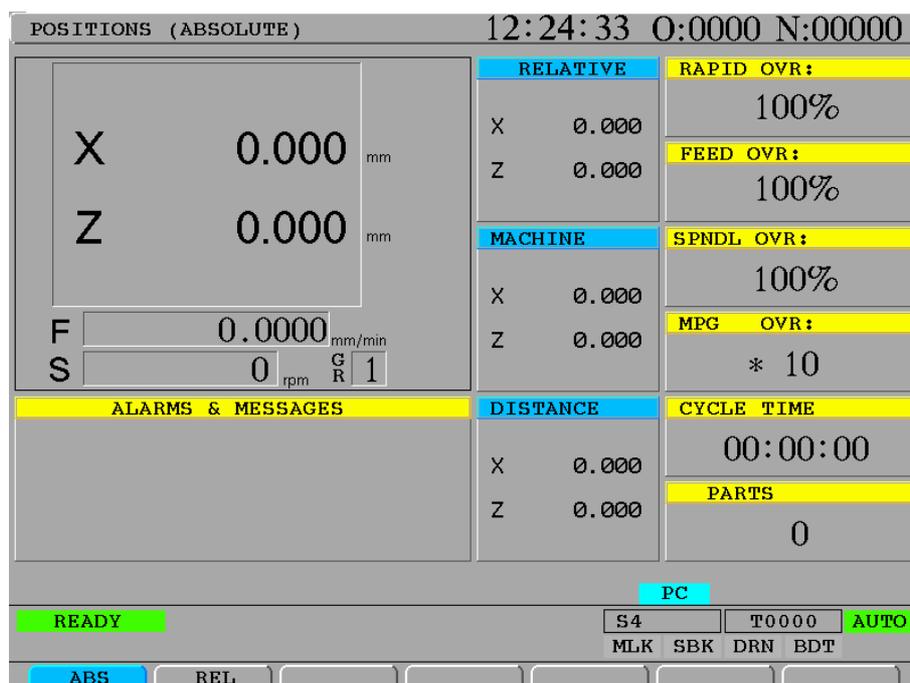
The above screens can also display the feedrate, run time, gear, parts, spindle revolution, messages etc.

### 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 shown.

#### Procedure for display the current position screen in the workpiece coordinate system.

1. Press the functional button [ POS ].
2. Press the software button [ ABS ].



### 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 ].



#### • 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 relative axis coordinates along a selected axis.

1. Input the address of the axis name ( U,W etc. ) in the relative coordinate screen.
2. Press the [ CAN ] button. The relative coordinates of the selected axis are reset.

Note:

*Pressing the address of the axis name once again cancels it.*

- **Coordinate display.**

The current position 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.
- Distance to go.

- **Distance to go**

The remaining distance is displayed in AUTO or MDI mode. The distance that tool should be moved in the current block is displayed.

- **Machine coordinate**

The least command increment is used as the unit for values displayed in the machine coordinate system. Coordinates are according machine zero positions.

### **11.1.3. Display of run time and parts count.**

The run time, cycle time and the number of machine parts are displayed on the current position display screen.

#### **Procedure for displaying run time and parts count on the position display screen**

1. Press the functional key [ **POS** ] to display the current position display screen.

The number of machined parts (PARTS), real time and cycle time (CYCLE TIME) are displayed

- **Part count.**

Indicates the number of machined parts. The number is incremented each time M02 or M30 or M80 commands are 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.

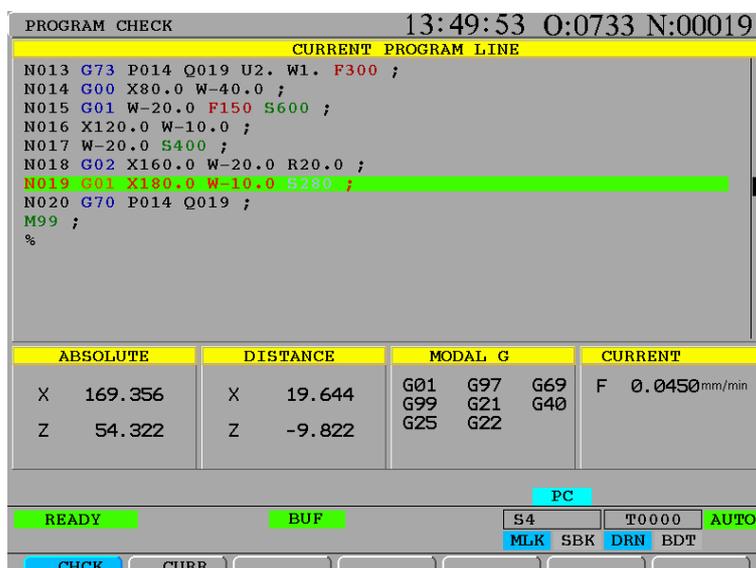
Program screen for MDI operation.

### 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 [ CHCK ]. The cursor is positioned over the current executed block.



#### • Additional information on this screen

The absolute position, distance to go, feedrate and modal G codes are shown also at the bottom of the program.





### 11.3. Screens displayed by functional key [ PRGRM ] ( in EDIT mode ).

This section describes the screen 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 list of the programs). w

#### 11.3.1. Displaying used memory and the list of programs.

Displays the number of registered programs, used memory and 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 ].



#### • Used memory.

##### Program numbers used.

**Program numbers used (6)** : The number of registered programs (including subprograms).

**Free (506)**: Number of programs that can be additionally registered.

##### Used memory area

**Used memory (36452)**: Used data memory (indicated in number of characters).

**Free (29083)**: Memory that can be additionally used for storing data (indicated in number of characters).

### 11.4.1. Setting and displaying the tool offset value.

The tool offset value and tool radius compensation value are specified by T code in the program. These values can be displayed and set on the screen.

#### Procedure for setting and displaying the cutter compensation value.

1. Press the functional button [ OFFSET ].
2. Press the soft button [ OFSW ].

The screen displays the tool offset data.

WEAR OFFSETS					12:41:58 O:0000 N:00000	
No.	X	Z	R	T		
01	1.000	2.000	0.000	0		
02	0.100	0.600	0.000	6		
▶03	3.000	0.900	0.000	0		
04	0.000	0.000	0.000	0		
05	0.000	0.000	0.000	0		
06	0.000	0.000	0.000	0		
07	0.000	0.000	0.000	0		
08	0.000	0.000	0.000	0		
09	0.000	0.000	0.000	0		
10	0.000	0.000	0.000	0		
11	0.000	0.000	0.000	0		
12	0.000	0.000	0.000	0		

ABSOLUTE	DISTANCE	CURRENT	MODAL G	
X 0.000	X 0.000		G00	G21
Z 0.000	Z 0.000		G97	G40
				G25
			G69	G22
			G99	

PC				
READY	S4	T0000	MDI	
	MLK	SBK	DRN	BDT
OFSW	OFSG	WORK	V100	V500

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.
- 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 [ **OFFSET** ].
2. Press the soft button [ **OFSG** ].

The tool geometry offset compensation values are displayed on the screen.

GEOM OFFSETS					12:45:48 O:0000 N:00000		
No.	X	Z	R	T			
▶ 01	112.300	26.300	0.000	1			
02	45.000	32.600	0.000	6			
03	99.000	8.300	0.000	5			
04	0.000	0.000	0.000	3			
05	0.000	0.000	0.000	0			
06	0.000	0.000	0.000	0			
07	0.000	0.000	0.000	0			
08	0.000	0.000	0.000	0			
09	0.000	0.000	0.000	0			
10	0.000	0.000	0.000	0			
11	0.000	0.000	0.000	0			
12	0.000	0.000	0.000	0			
ABSOLUTE		DISTANCE		CURRENT		MODAL G	
X	0.000	X	0.000		G00	G21	
Z	0.000	Z	0.000		G97	G40	
					G69	G25	
					G99	G22	
PC							
READY		S4		T0000		MDI	
		MLK		SBK		DRN	
OFSG		WORK		V100		V500	

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 using the cursor moving buttons.

- Press the [X] or [Z] button. Press the [ INPUT ] button.

4. Enter the geometry 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.

### 11.4.3. Work coordinate shift.

#### Procedure for displaying and setting the work coordinate shift value

1. Press the functional key [ OFFSET ].
2. Press the soft key [ WORK ].

On the screen is displayed work coordinate shift screen

WORK COORDINATE SHIFT		12:48:44 O:0000 N:00000	
SHIFT VALUE		MEASUREMENT	
X	10.000	X	0.000
Z	6.000	Z	0.000
ABSOLUTE	DISTANCE	CURRENT	MODAL G
X 10.000	X 0.000		G00 G21 G97 G40 G25 G69 G22 G99
Z 6.000	Z 0.000		
PC			
READY		S4	T0000 MDI
		MLK	SBK DRN BDT
OFSW	OFSG	WORK	V100 V500

3. Enter X (Z) and shift value.
4. Press the button [ INPUT ].

#### Note:

Could be entered relative values with U (W), too. This function is possible if in 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 on screens VARIABLES. 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 of custom macro common variables

1. Press the functional button [ **OFFSET** ].

2. Press the soft button [ **V500** ].

On the screen are displayed nonvolatile variables from #500 to #535

VARIABLES (NV)		18:24:07 O:7878 N:00001			
No.	VALUE	No.	VALUE		
▶ 500	792	512	9298	524	0
501	298	513	0	525	0
502	929	514	0	526	0
503	985	515	0	527	7929856
504	0	516	0	528	0
505	0	517	0	529	0
506	0	518	0	530	0
507	0	519	0	531	0
508	0	520	0	532	0
509	929	521	0	533	0
510	56	522	0	534	0
511	856	523	0	535	0

ABSOLUTE		DISTANCE		NEXT	MODAL G
X	-44.144	X	0.000		G01 G40
Y	12.796	Y	0.000		G17 G43
Z	-2.000	Z	0.000		G90 G80
					G94 G98
					G21 G67
					G54

USE			
READY	S0	T0101	EDIT
	MLK	SBK	DRN BDT
OFS	V100	V500	WORK

3. Move the cursor to the variable that will be changed.

4. Enter the new value and press [ **INPUT** ] button.

5. The screens with variables from 100 ÷ 131, and screen with variables from 500 ÷ 531, take turns by pressing the soft buttons [ **V100** ] (volatile) and [ **V500** ] (nonvolatile).

## 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, spindle and the other parts of the system.

This chapter describes how to set parameters by MDI panel. Parameters can also be set by an external input/output device or nonvolatile memory.

If the [ PARAM / DGN ] functional key is pressed, the following data can be displayed and set:

- Setting data
- Parameters
- Self – diagnostic data Dane od auto diagnostyki
- Controller ladder diagram

### 11.5.1. Displaying and setting parameters.

Parameters are set to determine the specifications and functions of the machine. The parameter's settings 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 parameters settings.

#### Procedure for displaying and setting parameters

1. When setting a parameter, first enable parameters modifying. 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		13:14:02 O:0000 N:00000	
No.	VALUE		
0500	1	0520	0
0501	1	0521	0
0502	1	0522	0
0503	1	0523	0
0504	1	0524	0
0505	1	0525	0
0506	1	0526	0
0507	1	0527	15000
0508	0	0528	0
0509	0	0529	10
0510	0	0530	10
0511	0	0531	0
0512	0	0532	0
0513	0	0533	400
0514	0	0534	200
0515	0	0535	100
0516	1000	0536	0
0517	3000	0537	0
0518	0	0538	0
0519	0	0539	0
		0540	4500
		0541	4500
		0542	4500
		0543	30
		0544	0
		0545	0
		0546	0
		0547	0
		0548	0
		0549	0
		0550	0
		0551	0
		0552	0
		0553	0
		0554	0
		0555	4500
		0556	0
		0557	0
		0558	0
		0559	0

PC

READY S4 T0000 MDI  
MLK SBK DRN BDT

SET PRM DGN LAD AXS SIM

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 page change keys.

5. Enter the desired value by 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 Emergency Stop button.

2. Press the functional button [ **PARAM / DGN**].

3. Press the soft button [ **SET** ] to display the SETTINGS screen.



4. Move the cursor to the parameter modify field by cursor move keys.

5. Press [1] and then [ INPUT ] to enable writing. At this time CNC enters in alarm state.

6. After modification of desired parameters move again cursor to the parameter modify field and press [0] and then [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.

Desired field is reached by the cursor move keys.

4. Wprowadzić nową wartość i przycisnąć przycisk [ INPUT ].

- Parameter write ( **PRM MODIFY** ).

Enables or disables parameter's write.

- 0 : disabled
- 1 : enabled

- Parameters reload ( **PRM RELOAD** ).

Reload parameters from EEPROM memory.

- 0 : disabled
- 1 : enabled

If this field is enabled, parameters will be automatically reloaded from EEPROM memory each time when CNC is restarted.

- Input unit.

- 0 : metric
- 1 : inch

## 11.6. Screens displayed by pressing the functional key [ ALARM ].

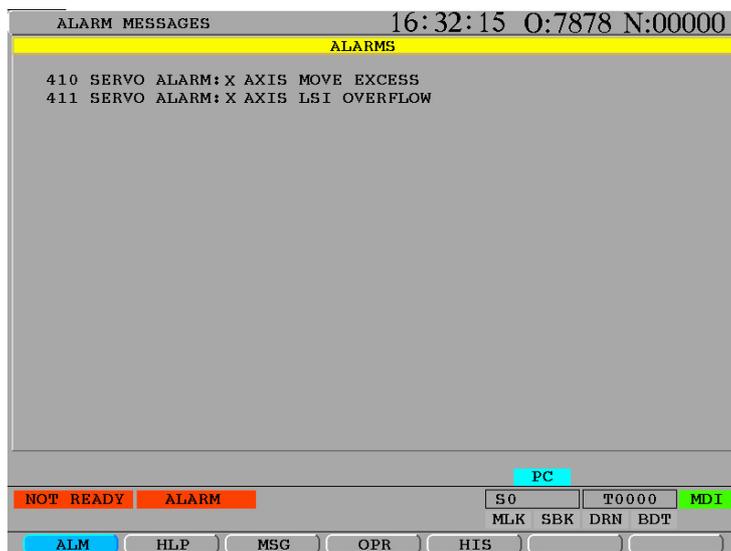
The system displays the alarm codes and the operator's messages by pressing the [ALARM ] button.

### 11.6.1. Displaying alarm messages.

#### Procedure for displaying the alarm messages

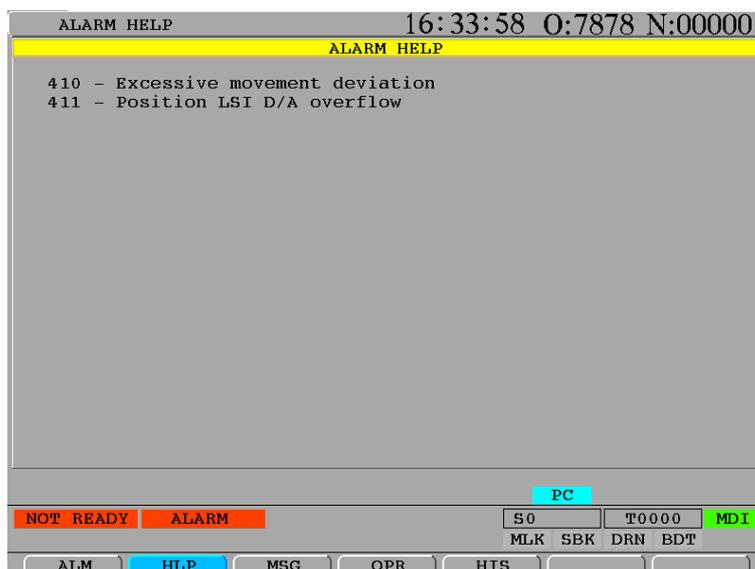
1. Press the functional button [ ALARM / OPR].
2. Press the soft button [ ALM ].

The alarm messages are displayed with their codes.



3. Press the button [ PgUp ] or soft button [HLP].

The alarm messages are displayed with a short description.



## 11.6.2. Displaying operator messages.

### Procedure for displaying operator messages

1. Press the functional button [ ALARM / OPR].
2. Press the soft button [ MSG ].

•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 the 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 and block number.

The program number and the sequence block number are displayed on the right corner of the screen as shown below.

```
PROGRAM CHECK 13:49:53 O:0733 N:00019
CURRENT PROGRAM LINE
N013 G73 P014 Q019 U2. W1. F300 ;
N014 G00 X80.0 W-40.0 ;
N015 G01 W-20.0 F150 S600 ;
N016 X120.0 W-10.0 ;
N017 W-20.0 S400 ;
N018 G02 X160.0 W-20.0 R20.0 ;
N019 G01 X180.0 W-10.0 S280 ;
N020 G70 P014 Q019 ;
```

The program number and the sequence block number that are displayed depend on the screen.

In the program screen in EDIT mode the number is that being edited and the sequence number prior the cursor are displayed.

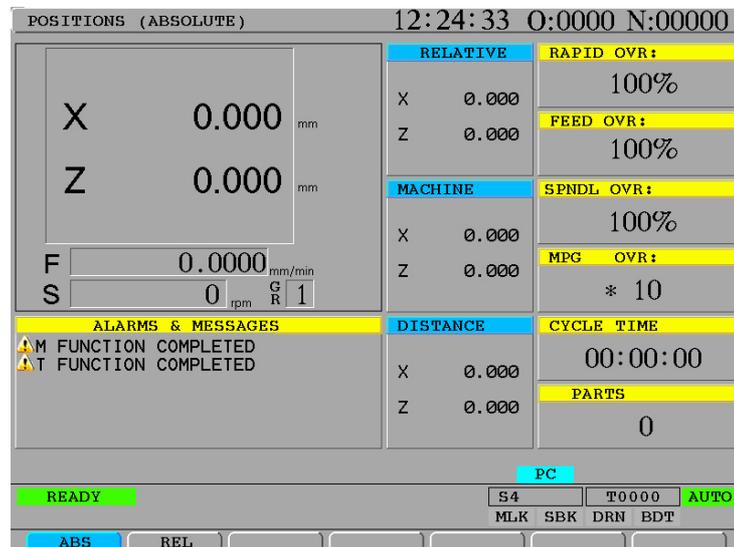
In other modes:

The program number and sequence number executed last are displayed.

## 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 codes 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)



NOT RAEDY: Indicates that the CNC is in the emergency stop state

PC : PC is selected for data transfer (USB serial port or RS232)

USB : USB flash drive is selected for data transfer

BUF : Indicates that the block to be executed next is being read

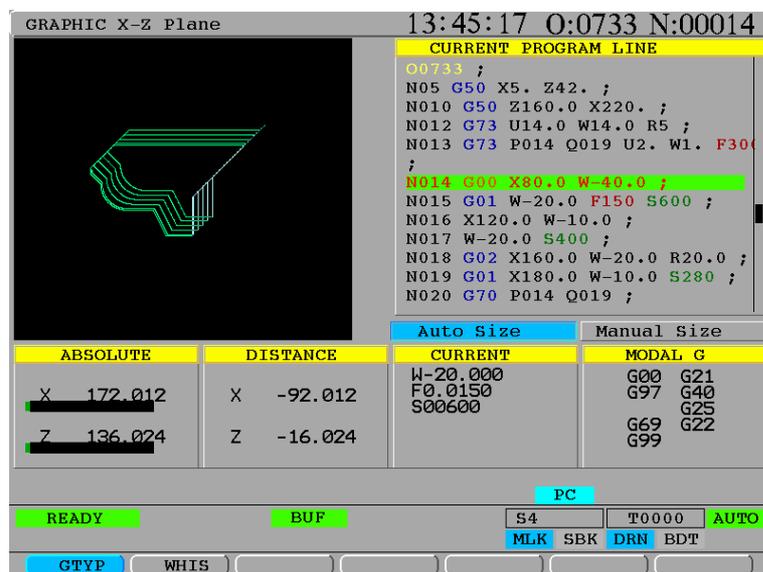
## 11.8. Screen group ( GRAPHICS ).

The CNC can show the tool path on this screen. There are two modes of graphic parameters – AUTO SIZE and MANUAL SIZE.

By pressing the [ALTER] button the showing mode is changed.

Note that in AUTO size mode the program is scanned and the minimum and maximum of coordinates for X and Z axis in the program are used to scale the graphics.

In manual mode operator can use the numeric keys to zoom in or out the graphics.



Options in screen „GRAPHIC TYPE”:

[ INSRT ] – in mode „Manuel Size” – sets the graphic parameters (zooms to selected rectangle).

[ INIT ] – zmazuje bieżącą grafikę, zachowując jej parametry graficzne.

[ ZOOM ] - wykonuje się automatyczne dostrojenie graficznych parametrów za pomocą zadanego przez operatora prostokątnego obszaru. Bieżąco wybrany prostokąt ma wymiary grafiki. Jego wymiary można zmniejszyć za pomocą przycisku [7], zwiększyć przyciskiem [3], przesunąć w lewo przyciskiem [6], w prawo przyciskiem [4], do góry [8], na dół [2], a jeżeli przyciśniemy przycisk [5] graficzny system współrzędnych będzie skalowany. Za pomocą przycisku [M] bieżący obszar graficzny może być zwiększony.





# ***PROGRAMMING***



# Contents

<b>1. INTRODUCTION</b> .....	7
<b>2. CONTROLLED AXES</b> .....	8
2.1 AXES CONTROLLED BY THE SYSTEM .....	8
2.2 LEAST INPUT INCREMENT .....	8
2.3 LEAST OUTPUT (COMMAND) INCREMENT .....	8
2.4 MAXIMUM STROKES .....	9
<b>3. PREPARATORY FUNCTIONS (G CODE)</b> .....	10
<b>4. INTERPOLATION FUNCTIONS</b> .....	13
4.1 POSITIONING (G00) .....	13
4.2 LINEAR INTERPOLATION (G01) .....	14
4.3 CIRCULAR INTERPOLATION (G02, G03) .....	15
<b>5. THREAD CUTTING (G32)</b> .....	19
<b>6. FEED FUNCTIONS</b> .....	22
6.1 RAPID TRAVERSE .....	22
6.2 CUTTING FEED RATE .....	22
6.2.1 Tangential speed constant control .....	22
6.2.2 Cutting feed rate clamp .....	23
6.2.3 Per minute feed (G98) .....	23
6.2.4 Per revolution feed (G99) .....	23
6.3 OVERRIDE .....	24
6.3.1 Feed rate override .....	24
6.3.2 Rapid traverse override .....	24
6.4 AUTOMATIC ACCELERATION/DECELERATION .....	25
6.4.1 Automatic acceleration/deceleration after interpolation .....	25
6.5 SPEED COMMAND AT CORNERS .....	26
6.6 DWELL (G04) .....	27
<b>7. REFERENCE POINT</b> .....	28
7.1 AUTOMATIC REFERENCE POINT RETURN (G28) .....	28
7.2 REFERENCE POINT RETURN CHECK (G27) .....	28
7.3 SECOND REFERENCE POINT RETURN (G30) .....	29

<b>8. COORDINATE SYSTEMS</b> .....	30
<b>8.1 COORDINATE SYSTEM SETTING (G50)</b> .....	30
8.1.1 Coordinate system setting .....	30
8.1.2 Coordinate system shift .....	32
8.1.3 Automatic coordinate system setting .....	32
8.1.4 Automatic coordinate system shift .....	33
8.1.5 Direct measured value input for work coordinate system shift .....	33
<b>9. COORDINATE VALUES</b> .....	34
9.1 ABSOLUTE AND INCREMENTAL PROGRAMMING .....	34
9.2 INCH/METRIC SYSTEM SETTING .....	35
9.3 DECIMAL POINT PROGRAMMING .....	36
9.4 DIAMETER AND RADIUS PROGRAMMING .....	37
<b>10. SPINDLE SPEED FUNCTIONS (S FUNCTIONS)</b> .....	38
10.1 SPINDLE SPEED COMMAND .....	38
10.2 CONSTANT SURFACE SPEED CONTROL (G96, G97) .....	38
10.2.1 Command .....	38
10.2.2 Spindle speed override .....	39
10.2.3 Clamping maximum spindle speed (G50) .....	40
10.2.4 Rapid traverse in constant surface speed control .....	40
10.3 SPINDLE SPEED DETECTION .....	41
<b>11. TOOL FUNCTIOS (T FUNCTIONS)</b> .....	43
11.1 TOOL SELECTION FUNCTION .....	43
11.2 HANDY TOOL LIFE MANAGEMENT .....	43
11.2.1 Display and setting of data required for tool life management .....	44
11.2.2 Compensation for programmed T code .....	44
<b>12. MISCELLANEOUS FUNCTIONS (M FUNCTIONS)</b> .....	46
12.1 MISCELLANEOUS FUNCTIONS .....	46
<b>13. PROGRAM CONFIGURATION</b> .....	48
13.1 BEGINNING OF THE PROGRAM .....	48
13.2 PROGRAMMED BLOCK .....	49
13.3 DISPOSITION OF THE PROGRAMS IN THE MEMORY .....	50
13.4 SUBPROGRAM .....	52
13.4.1 Subprogram calling .....	52
13.4.2 Subprogram return .....	53
13.5 COMMENT .....	54
13.6 OPTIONAL BLOCK SKIP .....	54
13.7 END OF PROGRAM .....	54

<b>14. FUNCTION TO SIMPLIFY PROGRAMMING</b> .....	55
14.1 CANNED CYCLES.....	55
14.1.1 Outer/internal diameter cutting cycle .....	55
14.1.2 Thread cutting cycle (G92).....	57
14.1.3 End face turning cycle (G94).....	57
14.1.4 Usage of canned cycle .....	60
14.2 MULTIPLE REPETITIVE CYCLE (G70 to G76) .....	61
14.2.1 Stock removal in turning (G71).....	61
14.2.2 Stock removal in facing (G72) .....	62
14.2.3 Pattern repeating (G73).....	64
14.2.4 Finishing cycle (G70) .....	66
14.2.5 End face peck drilling cycle (G74) .....	69
14.2.6 Outer/internal diameter drilling cycle (G75) .....	70
14.2.7 Multiple thread cutting cycle (G76) .....	71
14.2.8 Notes of multiple repetitive cycles (G70 to G76) .....	73
14.3 CHAMFERING AND CORNER R.....	74
14.4 MIRROR IMAGE FOR DOUBLE TURRETS (G68, G69) .....	76
14.5 DIRECT DRAWING DIMENSION PROGRAMMING .....	77
<b>15. COMPENSATION FUNCTIONS</b> .....	80
15.1 TOOL OFFSET .....	80
15.1.1 Basic Tool Offset .....	80
15.1.2 Tool geometry offset and tool wear offset .....	80
15.1.3 T code for tool offset.....	81
15.1.4 Tool selection .....	82
15.1.5 Offset number .....	82
15.1.6 Offset .....	82
15.1.6.1 Wear offset .....	82
15.1.6.2 Geometry offset .....	84
15.2 TOOL NOSE RADIUS COMPENSATION (G40 to G42) .....	85
15.2.1 Imaginary tool nose.....	86
15.2.2 Direction of imaginary tool nose .....	87
15.2.3 Offset number .....	88
15.2.4 Work position and move command .....	90
15.2.5 Notes on tool radius compensation .....	96
15.2.6 Detailed description of tool nose radius compensation .....	101
15.3 CHANGING OF TOOL OFFSET AMOUNT (G10).....	126
<b>16. MEASUREMENT</b> .....	127
16.1 SKIP FUNCTION (G31) .....	127
16.2 AUTOMATIC TOOL OFFSET (G36, G37) .....	128
16.3 DIRECT SETTING OF THE TOOL COMPENSATION VALUE .....	131
16.4. DIRECT SETTING OF THE COORDINATE SYSTEM SHIFT VALUE.....	132

<b>17. CUSTOM MACRO</b> .....	133
<b>17.1 VARIABLES</b> .....	133
<b>17.1.1</b> Expression of variable .....	133
<b>17.1.2</b> Reference of variables .....	133
<b>17.1.3</b> Display and setting of variable value .....	134
<b>17.2 KIND OF VARIABLE</b> .....	134
<b>17.2.1</b> Common variable #100 to #131 and #500 to #531 .....	134
<b>17.2.2</b> System variables .....	135
<b>17.3 MACRO INSTRUCTIONS (G65)</b> .....	137
<b>17.3.1</b> Variable arithmetic command .....	139
<b>17.3.2</b> Control command .....	142
<b>17.4 CAUTIONS ON CUSTOM MACRO</b> .....	144
<b>17.5 APLICATION OF CUSTOM MACRO</b> .....	144
<b>17.5.1</b> Shearing machine .....	144
<b>17.5.2</b> Interface signal .....	145

## 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; } G01 is effective in this range
X20. Z20. F20; }
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** respectively 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 fluctuation detect OFF
G26	G26		Spindle speed fluctuation 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.

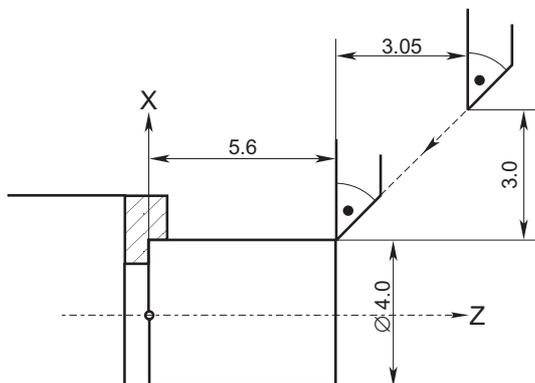
*Format:*     **G00 X(U)\_\_\_\_\_Z(W)\_\_\_\_\_;**

*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)

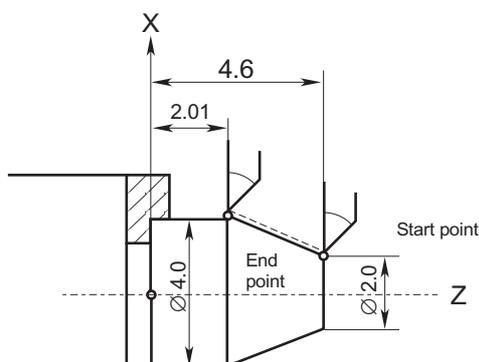
The linear interpolation can be performed by commanding **G01**.

*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 $\alpha$  W $\beta$  Ff;**

feedrate in **X** axis direction: **F $x$  =  $\alpha$ .f/L**

feedrate in **Z** axis direction: **F $z$  =  $\beta$ .f/L**

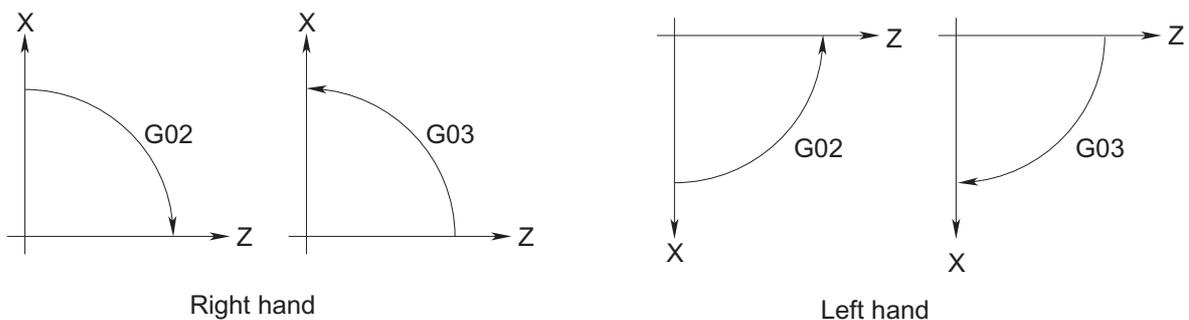
where:  $L = \sqrt{\alpha^2 + \beta^2}$

### 4.3 Circular Interpolation (G02, G03)

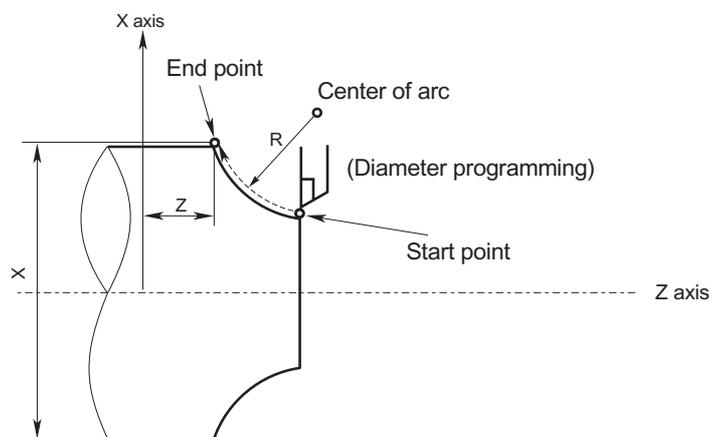
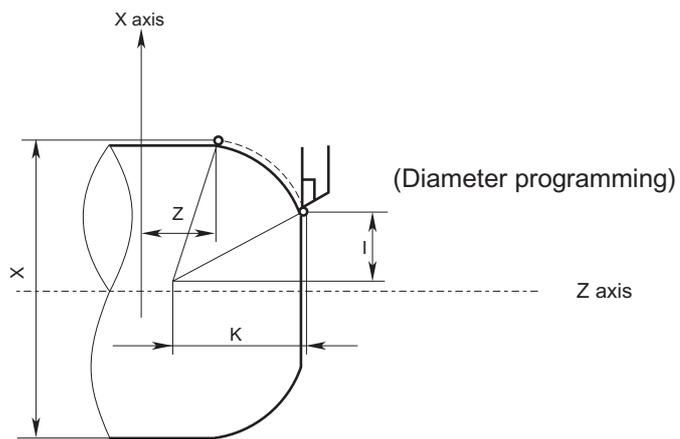
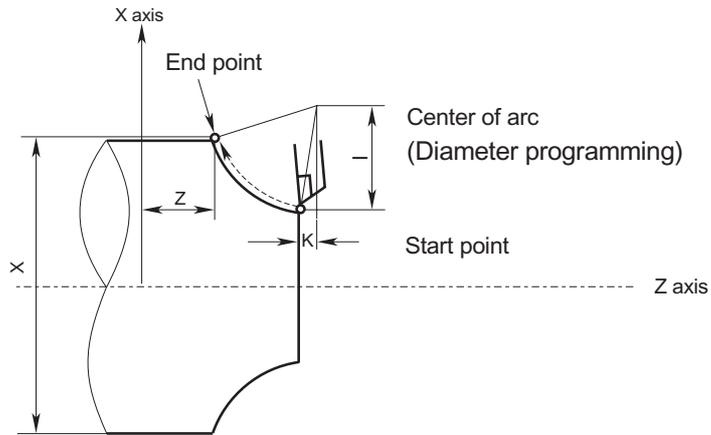
The following command will move the tool along a circular arc.

	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.



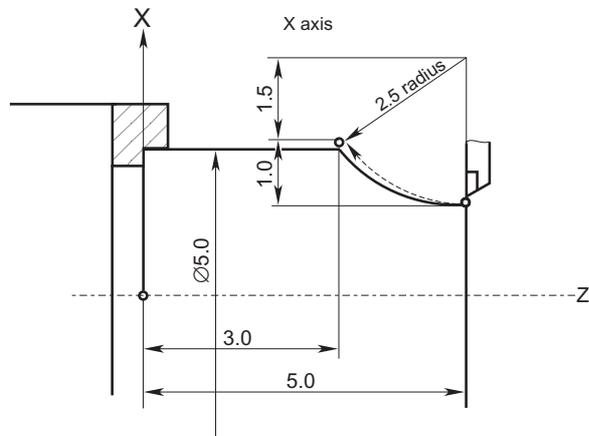
Examples:



(For absolute command)

(Diameter programming)

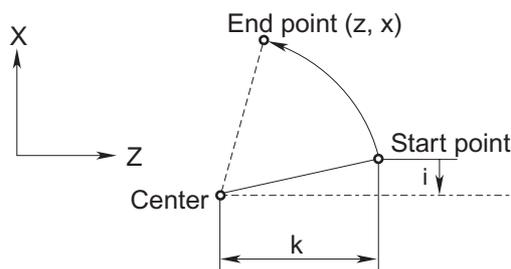
G02X5.0Z3.0I2.5F0.03;  
 or G02U2.0W-2.0I2.5F0.03  
 or G02X5.0Z3.0R2.5F0.03  
 or G02U2.0W-2.0R2.5F0.03



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**.



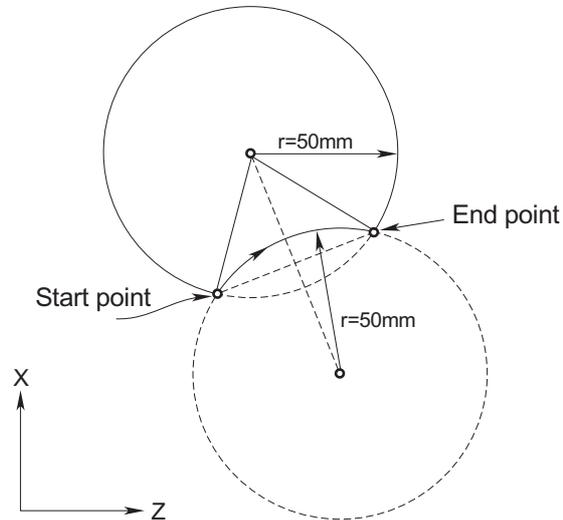
This radius is specified by address **R**. In this case an arc exceeding 180° can not be commanded.

*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 omitted. **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°).

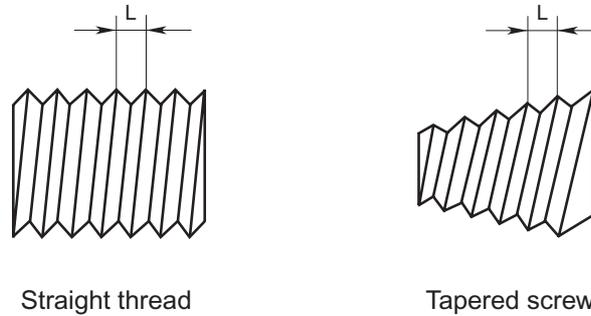
The error between the specified feedrate and the actual tool feedrate is  $\pm 2\%$  or less.

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**.



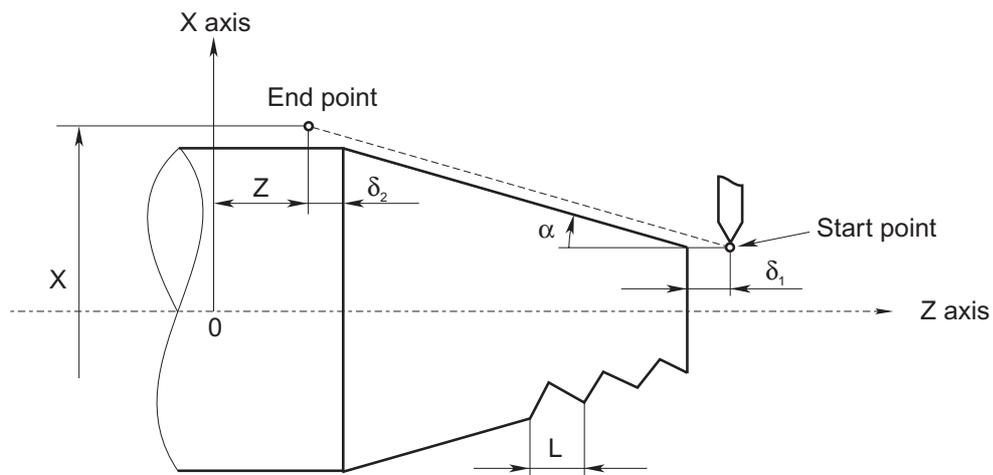
**Format: G32 X(U)\_\_\_\_\_ Z(W)\_\_\_\_\_ F\_\_\_\_\_;**

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.

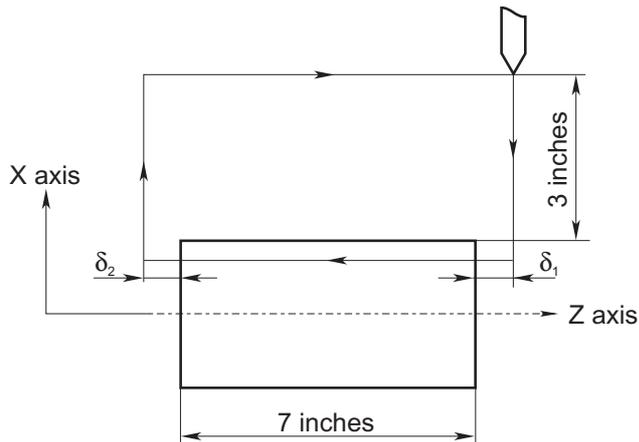
In case of taper thread cutting it is necessary to specify different values for **X** and **Z** axis.



The thread lead must be specified as a radius value.

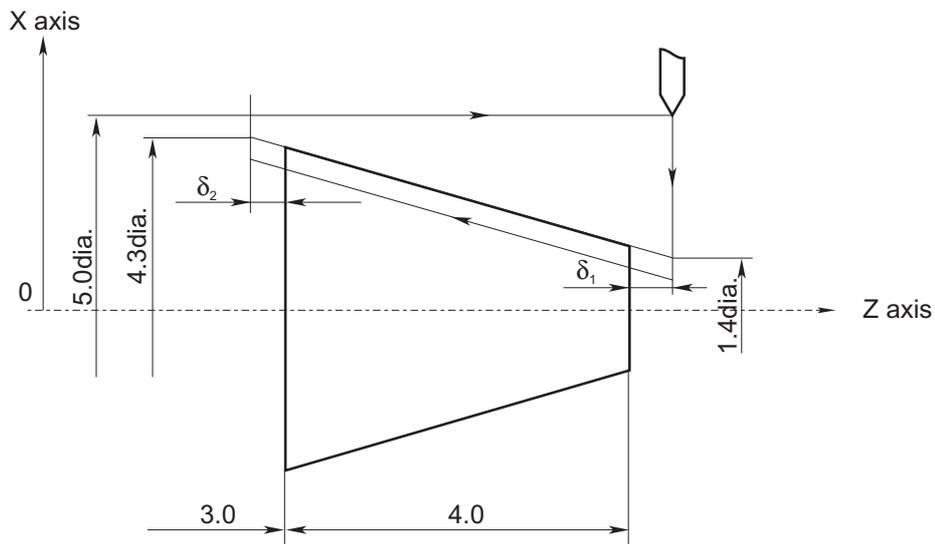
In general, the lag of the servosystem will produce somewhat incorrect leads at the starting and ending points of the thread cut. Therefore, when specify the thread length, it is necessary to specify longer one than the thread to be machined.

*Example: Straight thread cutting*



The following values are used in programming:  
 Thread lead : 0.4 inch  
 $\delta_1$  : 0.3 inch  
 $\delta_2$  : 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 failed 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%**

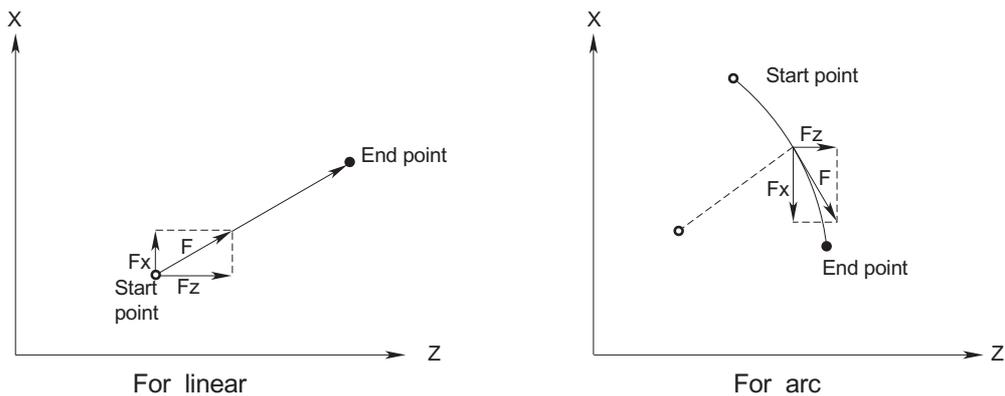
where: **F0** is a constant speed set by parameter No.0533.

### 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.



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

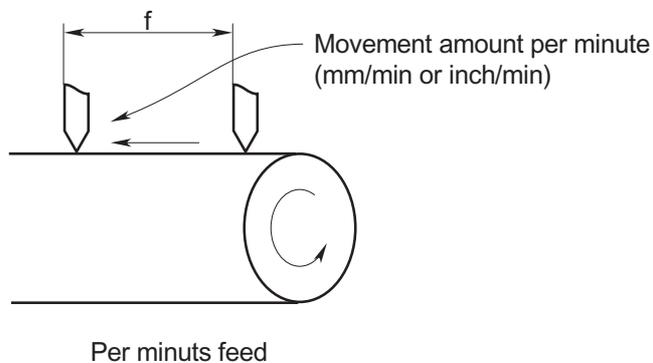
$$F = \sqrt{F_x^2 + F_z^2}$$

### 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.

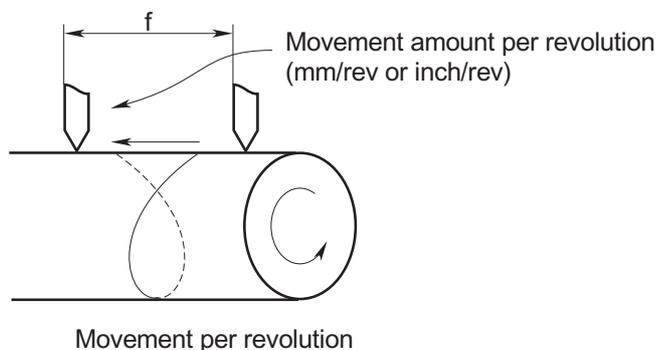
### 6.2.3 Per minute feed (G98)

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.



### 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 specified.



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

$$F_m = F_r \cdot R$$

where:

**F<sub>m</sub>** - per minute feedrate

**F<sub>r</sub>** - 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  $\pm 2\%$ .

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

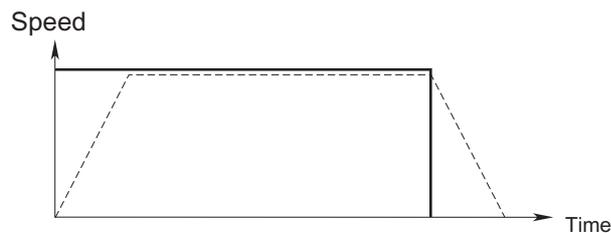
Automatic acceleration/deceleration is performed when starting and ending movement, resulting in smooth start and stop. Automatic acceleration/deceleration is performed also when feed rate changes.

#### ***Different kinds of acceleration/deceleration are used:***

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

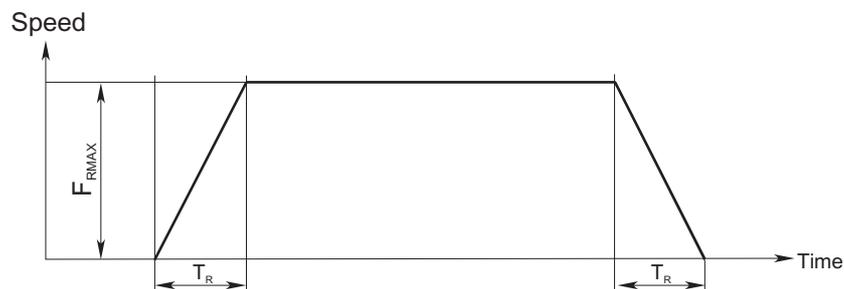
***Cutting feed:*** - Exponential acceleration/deceleration. It is set by parameter No. 0529.

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



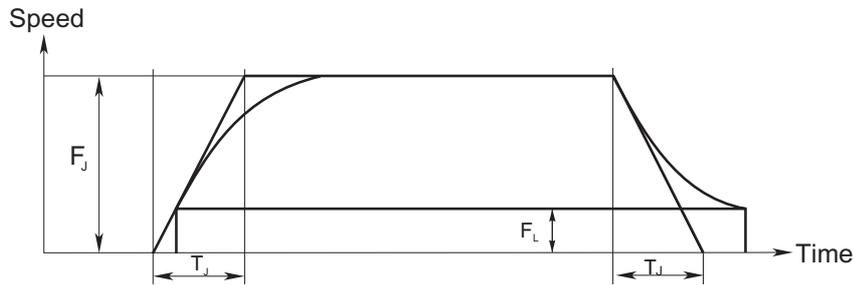
— Rate after interpolation  
- - - Rate after acceleration/  
deceleration control

#### **Rapid traverse**



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

### JOG feed

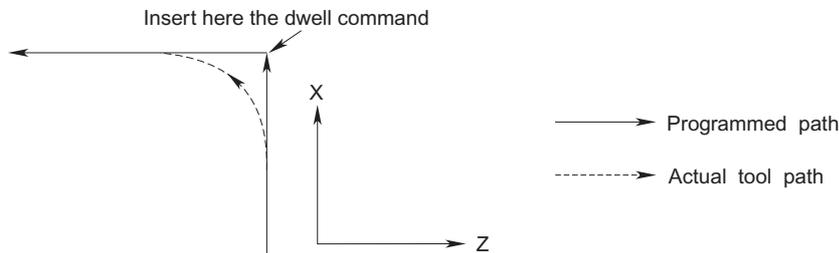


- $F_J$ : Jog feed rate
- $T_J$ : Jog feed time constant  
(Parameter No.529)
- $F_L$ : Low feed rate after deceleration  
(Parameter No.530)

## 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.

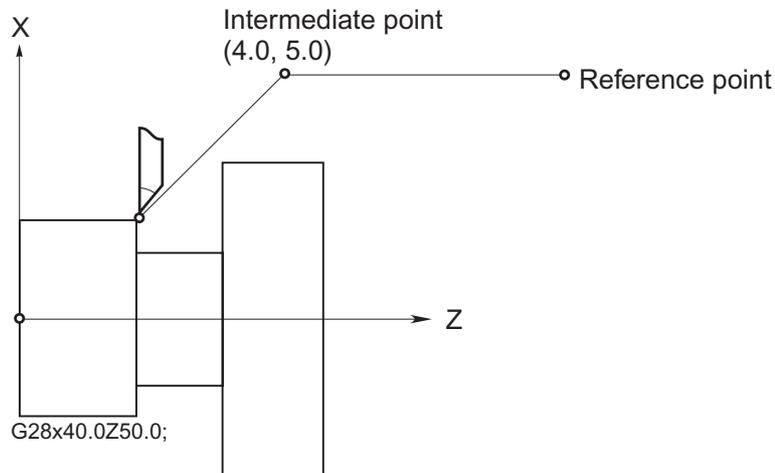
## 7. REFERENCE POINT

### 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 automatic 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)

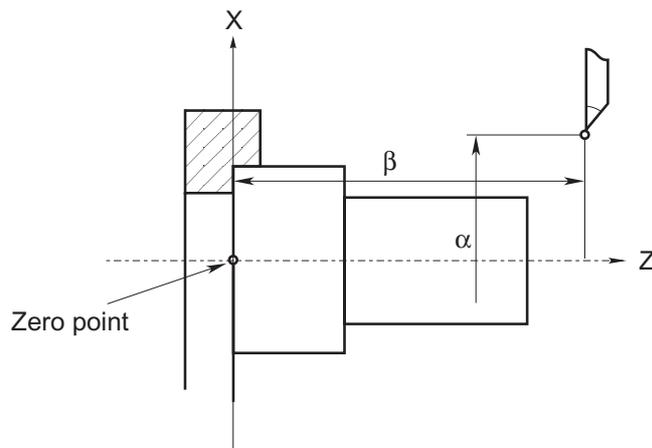
**Format:** G30 X(U)\_\_\_\_\_ Z(W)\_\_\_\_\_;

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  $X\alpha$   $Z\beta$  is commanded

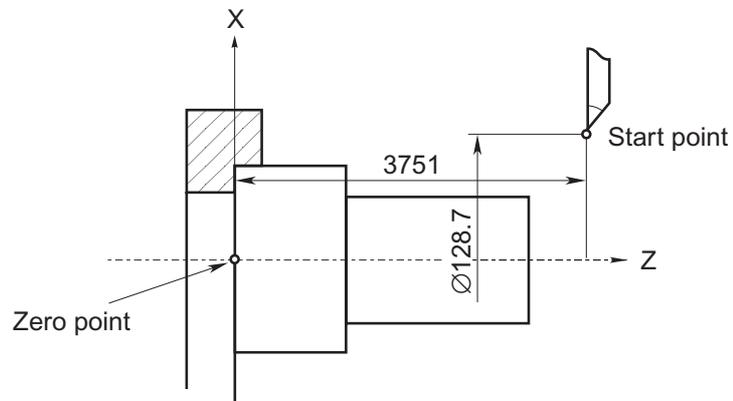
### 8.1 COORDINATE SYSTEM SETTING (G50)

#### 8.1.1 Coordinate system setting

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

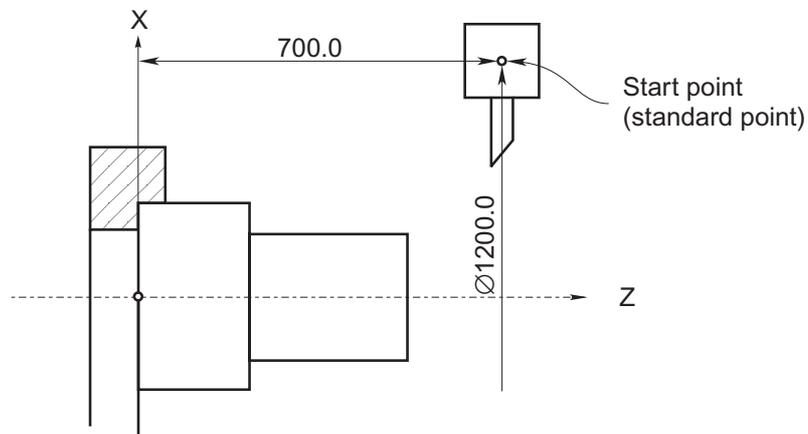
**G50 X**\_\_\_ **Z**\_\_\_;

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 referred 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 radius 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.



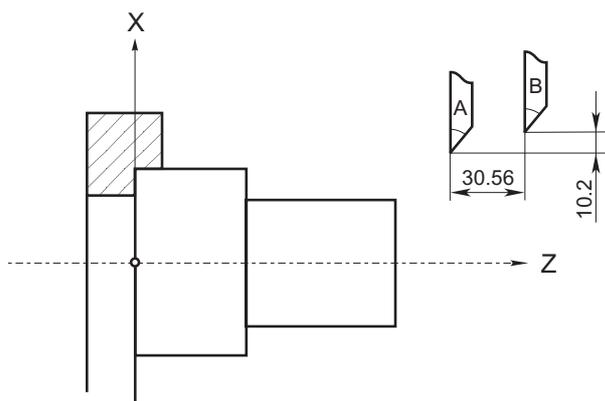
As shown in the illustration above, the reference point on the turret is aligned with the start point and the coordinate system is set at the head of the program. When an absolute command is assigned, the reference point will move to the position commanded. In order to move the tip of the cutting edge 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\_\_\_\_\_;**

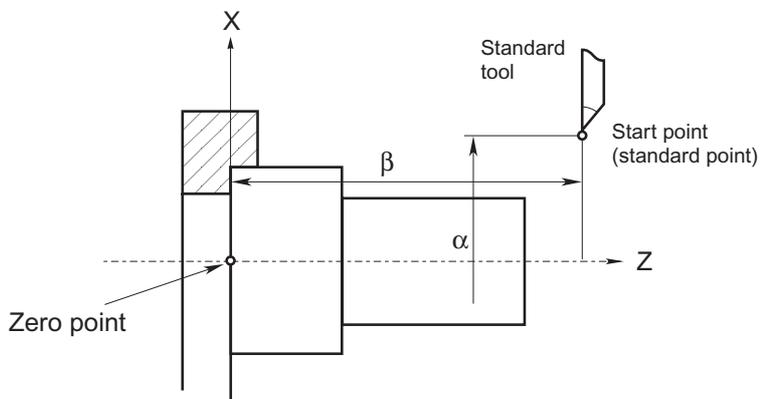
This command creates a new coordinate system which is shifted in comparison with the current one with translation given by **U** and **W**.



### 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\_\_\_\_\_;**



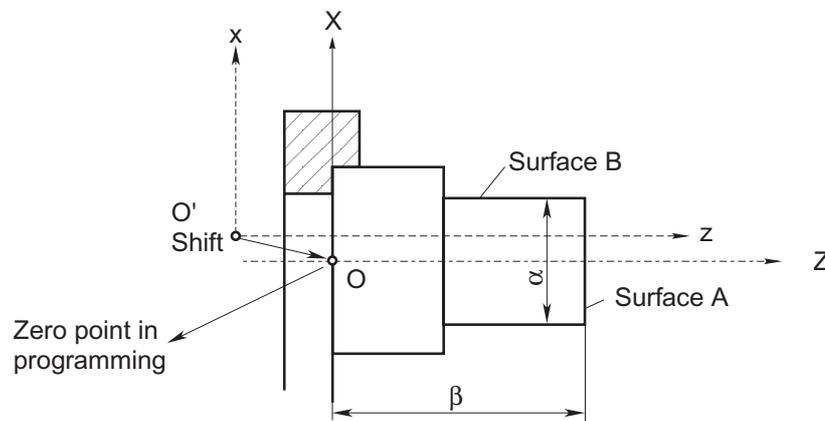
When the tool tip of the standard tool is set as the standard point

#### 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:



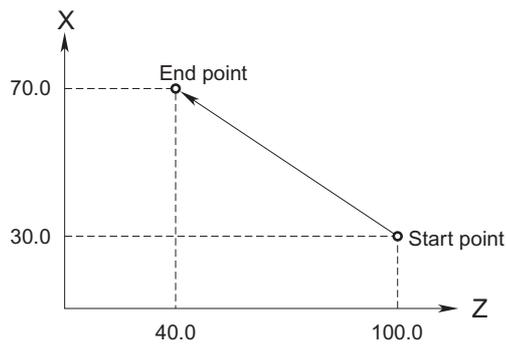
- (1) *Cut the workpiece along the surface A using a standard tool in manual operation.*
- (2) *Retract the tool only in the X direction without Z axis movement and stop the spindle.*
- (3) *Measure distance  $\beta$  from the zero point in programming to surface A.*
- (4) *Cut the workpiece along surface B by manual operation.*
- (5) *Retract the tool only in Z direction without X axis movement and stop the spindle.*
- (6) *Measure the diameter  $\alpha$  at surface B.*
- (7) *Set the measured values  $\alpha$  and  $\beta$  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

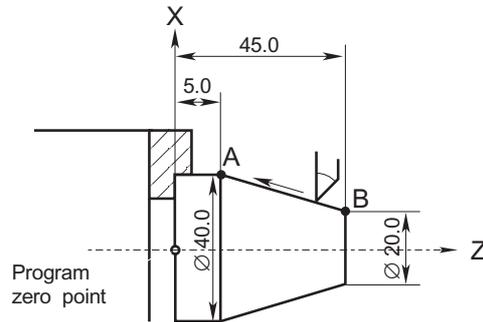
Absolute command	Incremental command	Notes
X	U	X axis move command
Z	W	Z axis move command

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 independent 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.

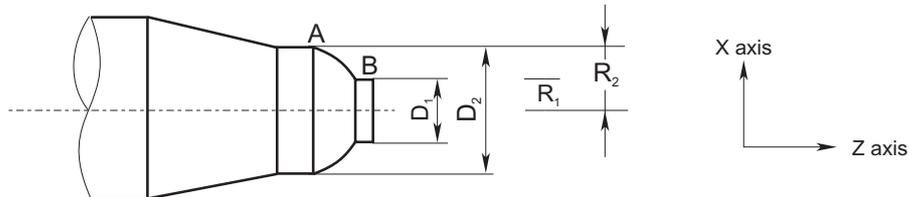
### 9.3 Decimal Point Programming

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 dimensions can be specified in two ways: Diameter and radius.



$D_1, D_2 \dots \dots$  Diameter programming  
 $R_1, R_2 \dots \dots$  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

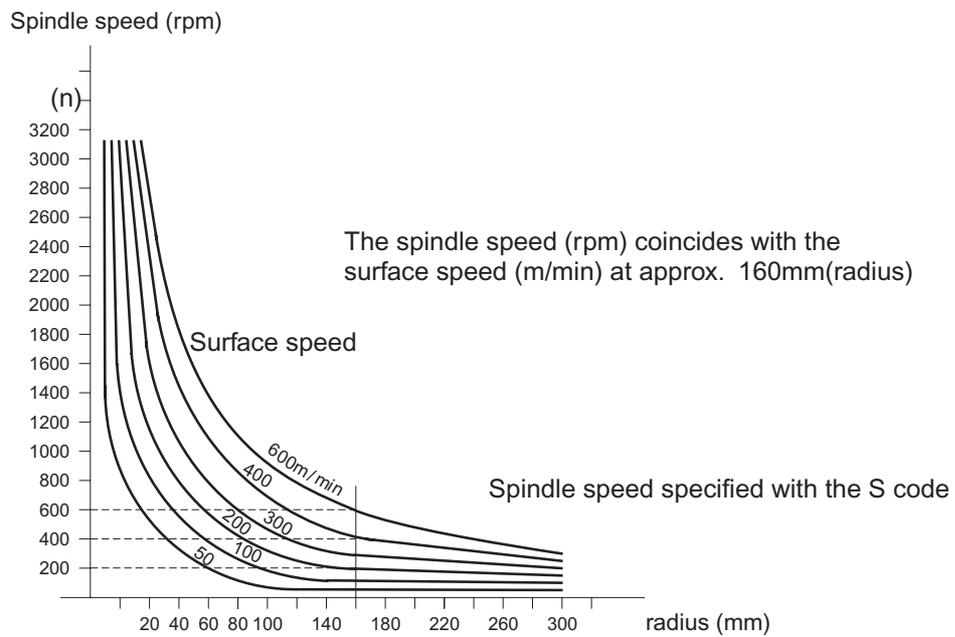
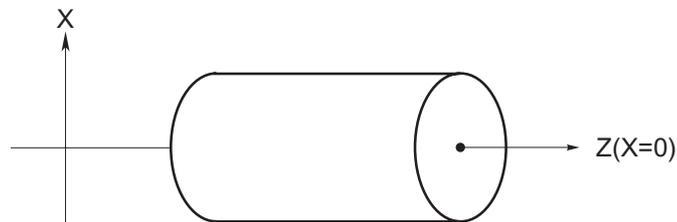
Constant Surface Speed Control is specified by the following command:

**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**.

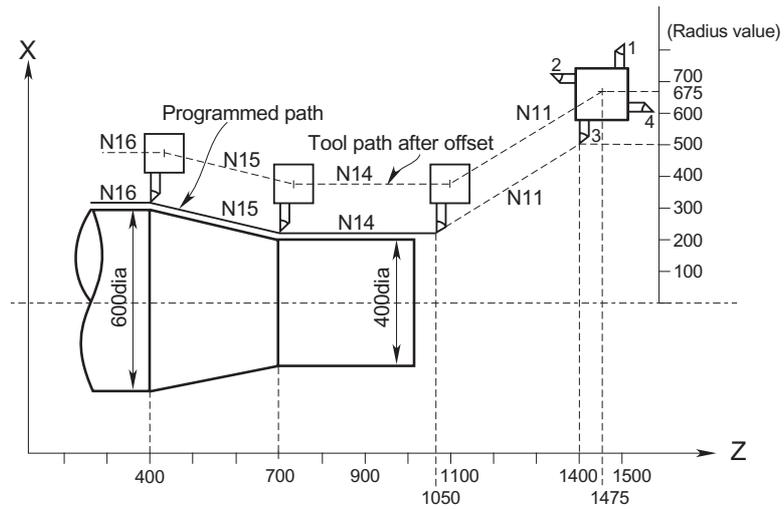
The constant surface speed is calculated even when a machine is operating in "MACHINE LOCK" status.

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\_\_\_\_\_;**

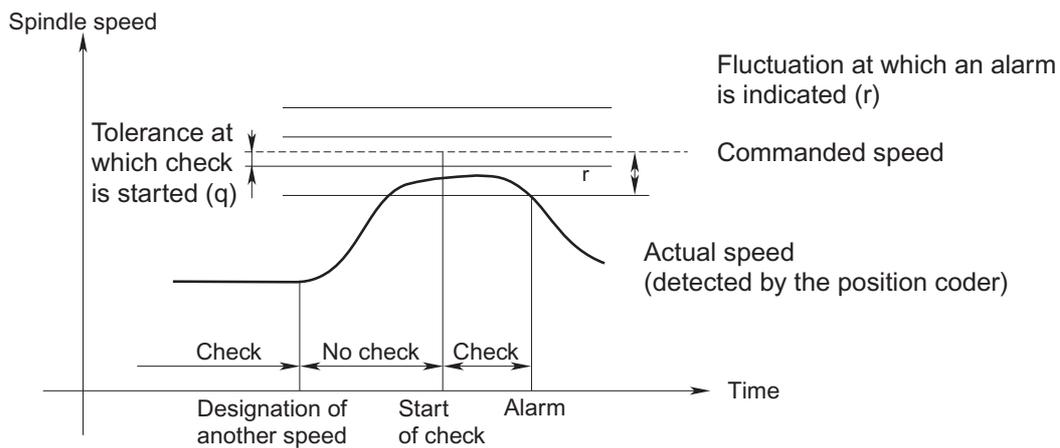
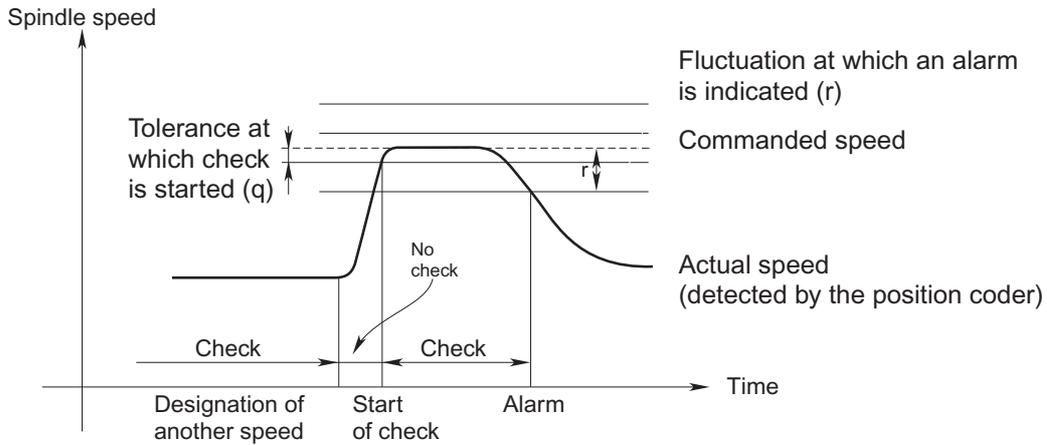
*where:* **P:** Time in ms for starting check when the commanded speed is not reached after a certain time

**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.



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 tool and 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 correction 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**" - number 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 COUNT (TOTAL)**" - total parts count

### 11.2.2 Compensation for programmed T code

The programmed T code **T00 XX** consists 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 **00 = oo**

Tool offset number **XX = xx**

**(2) After the first life**

Tool selection number **00=oo+(tool selection number compensation value)**

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

**(3) After the N life**

Tool selection number **00=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

<i><b>Program</b></i>	<i><b>After first life</b></i>	<i><b>After second life</b></i>
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.

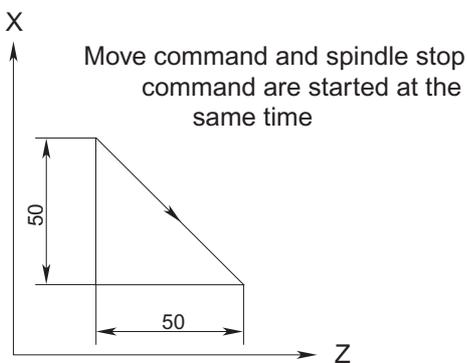
When a move command and **M** codes are specified in the same block, the commands are executed in one of the following two ways;

- (1) *Simultaneous execution of the move command and **M** function*
- (2) *Executing **M** function commands upon completion of the move command execution*

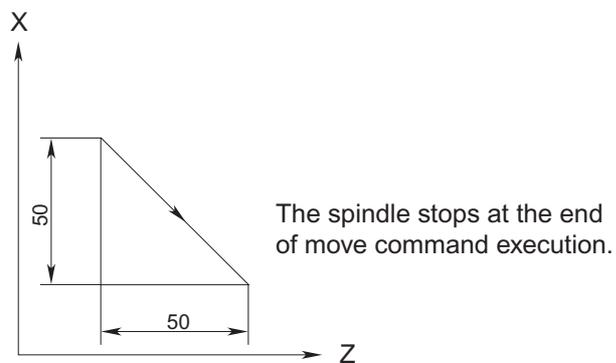
*Example:*

Diameter programming

**N1 G01 X -100.0 Z50.0 M05;**



Sequence i



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 sentences. 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*

Exemplary 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:

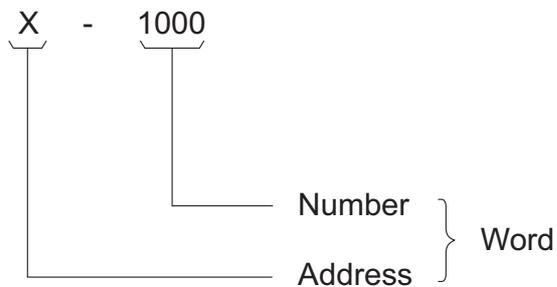
**LF** is **LINE FEED** in **ASCII** standard

**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
Sequence 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 per 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 repetitions	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:*

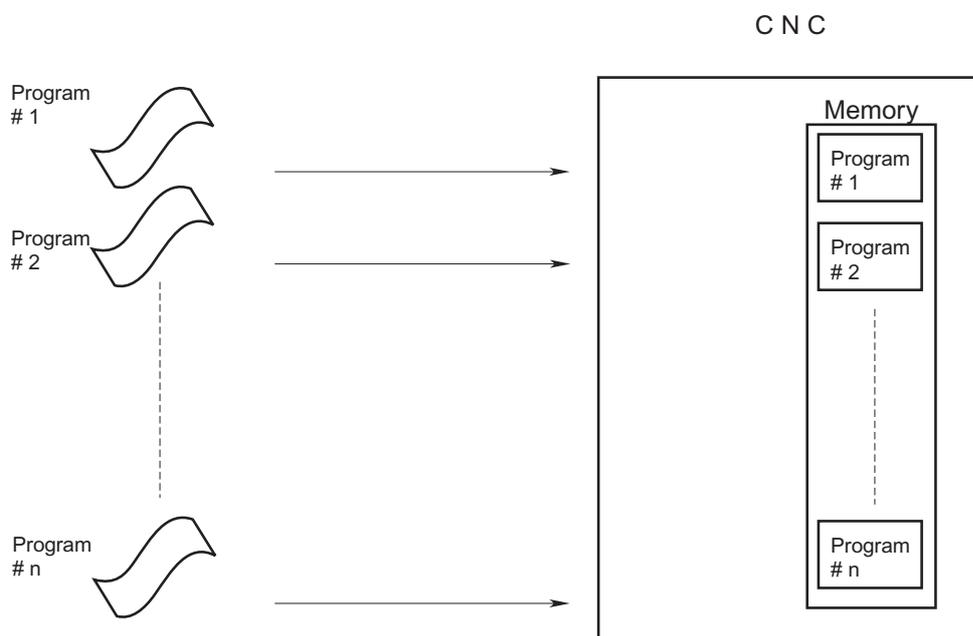
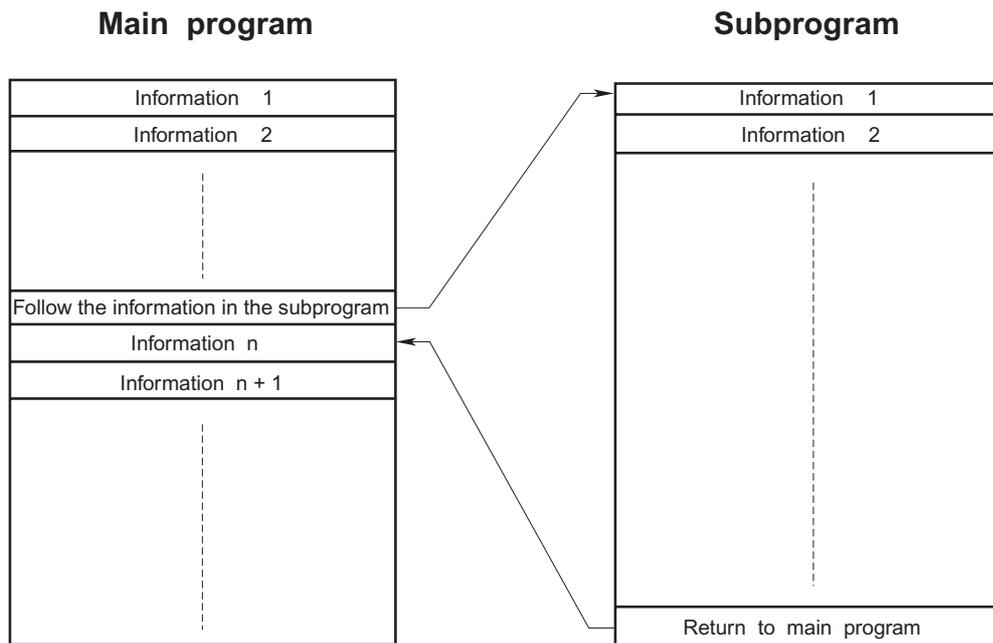
**N0010 G01 X4. Z0.2;**

### 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):

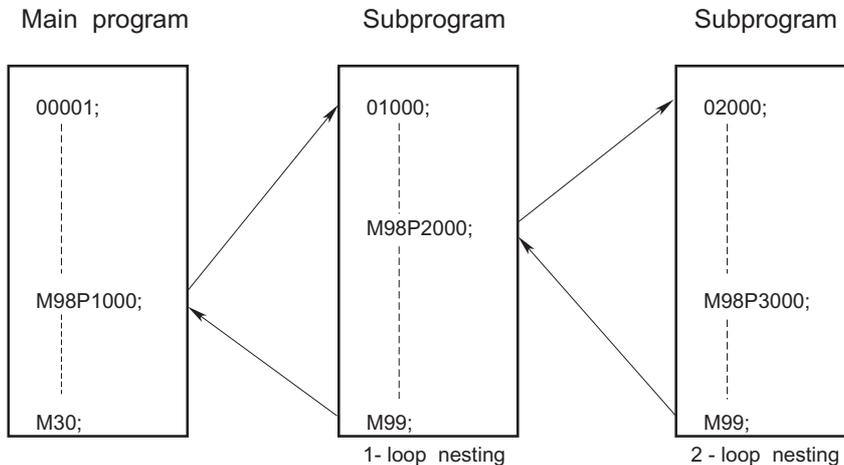
**Oxxxx**

*where:* **xxxx is 1 – 9999 number**



## 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:

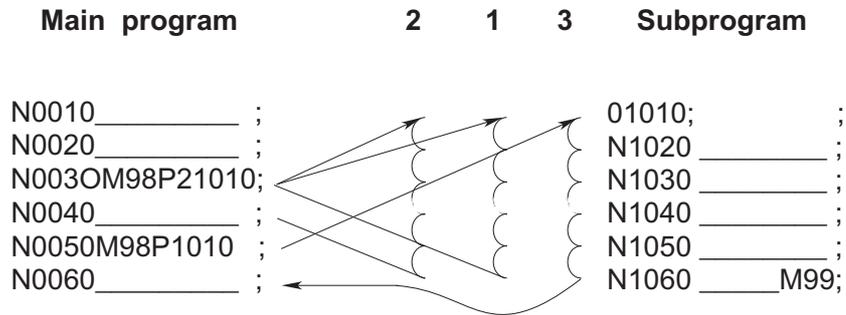
...

**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:

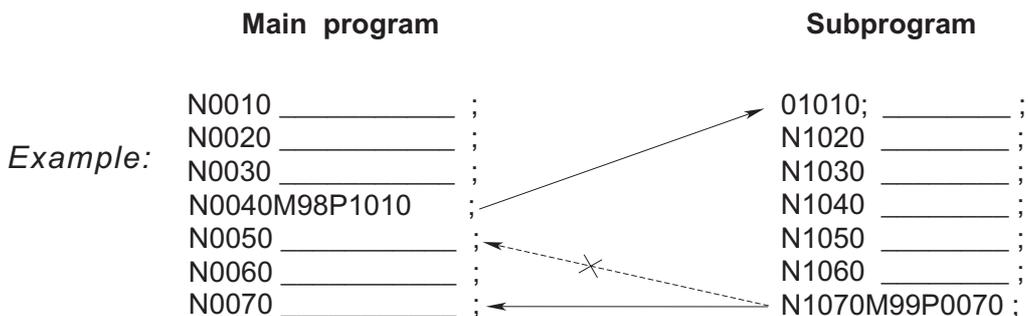
**M99 Pnnnn;**

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 comments 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 allowed.

### 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 sentence, the addresses before it are always valid, 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

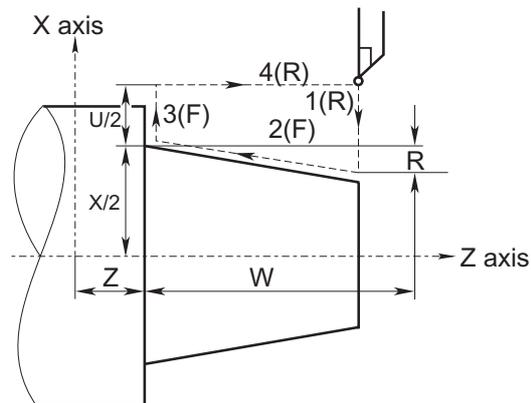
For repetitive machining peculiar to turning, such as metal removal in rough cutting, a series of paths usually specified in a range of several blocks can be specified in one block. For such operations can be used program cycles with suitable parameters.

### 14.1 Canned cycles

#### 14.1.1 Outer/internal diameter cutting cycle

This cycle is specified with the following command:

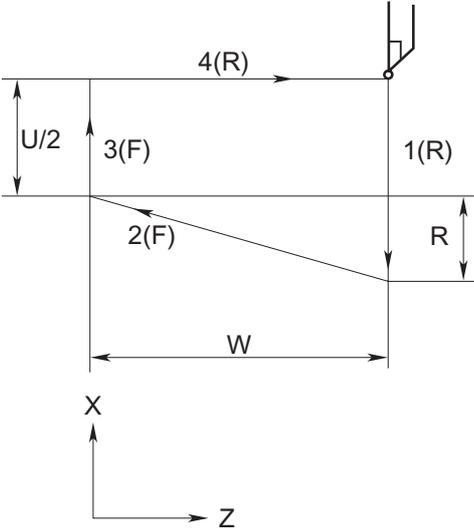
**G90 X(U)\_\_\_\_\_ Z(W)\_\_\_\_\_ R\_\_\_\_\_ F\_\_\_\_\_;**



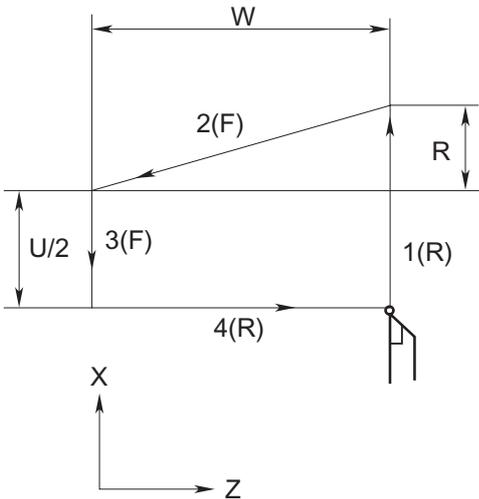
*where:*                    **R** - rapid traverse  
                              **F** - specified by **F** code

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

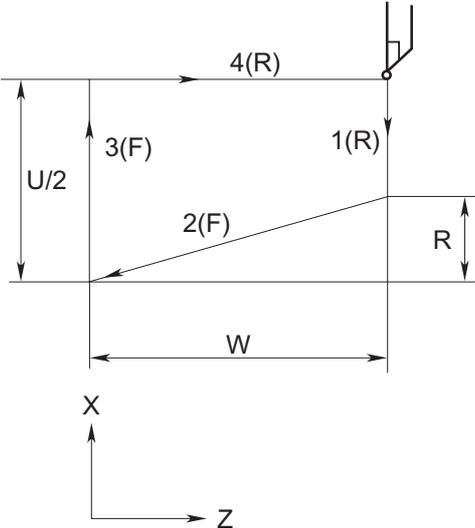
1)  $U < 0, W < 0, R < 0$



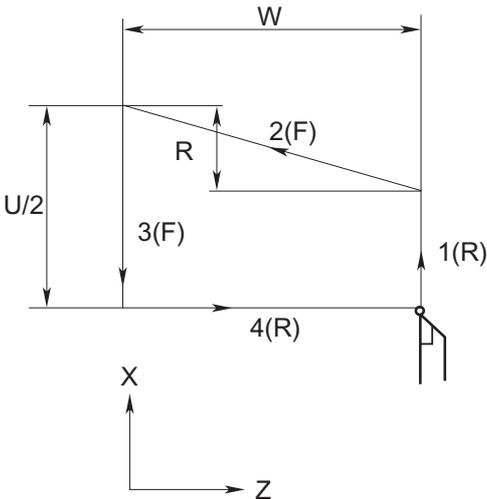
2)  $U > 0, W < 0, R > 0$



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



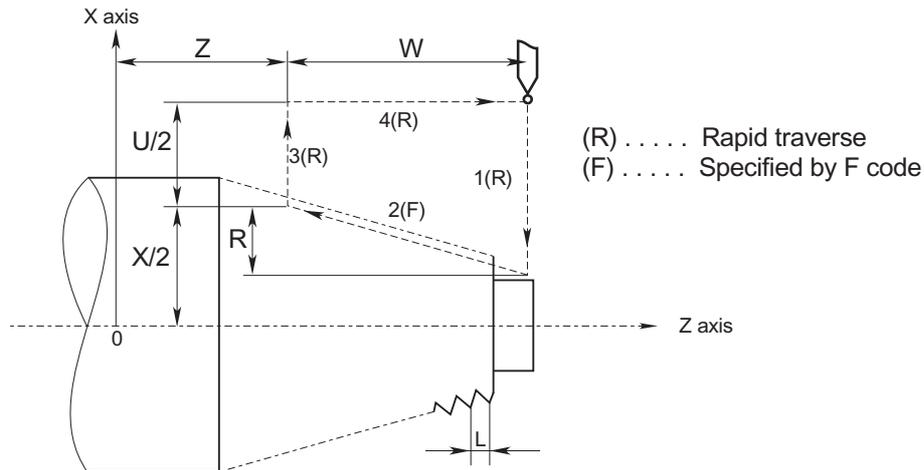
4)  $U > 0, W < 0, R < 0, \text{ at } |R| \leq \frac{|U|}{2}$



### 14.1.2 Thread cutting cycle (G92)

This cycle is specified by the following command:

**G92 X(U)\_\_\_ X(W)\_\_\_ R\_\_\_ F\_\_\_;**



where:        **R** - rapid traverse  
              **F** - specified by **F** code  
              **L** - thread cutting

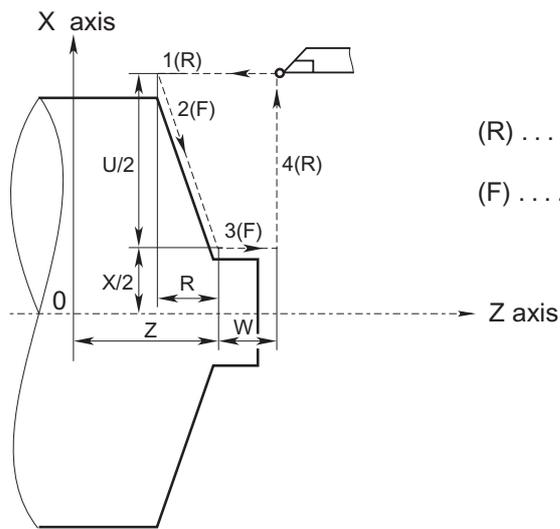
In incremental programming the cases are the same as in **G90**. The range of thread leads, limitation of spindle speed are the same as in **G32** (thread cutting). In the "SINGLE BLOCK" mode, the whole block is performed without break.

### 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\_\_\_;**

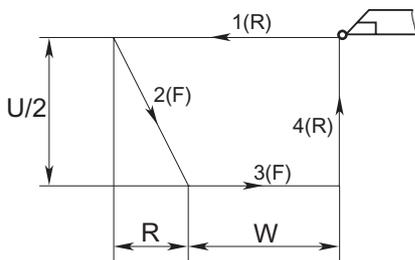
where:        **R** - rapid traverse  
              **F** - specified by **F** code



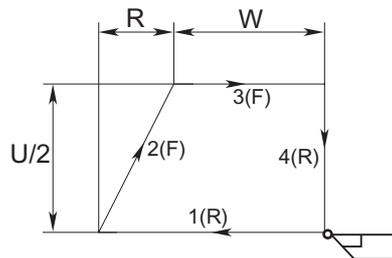
(R) . . . . . Rapid traverse  
 (F) . . . . . Specified by F code

**In incremental programming the following cases are considered:**

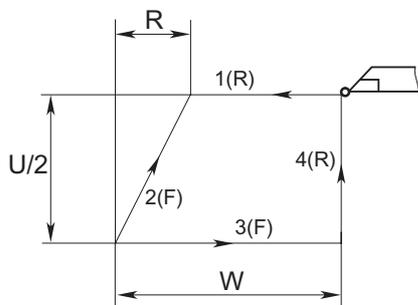
**1)  $U < 0, W < 0, R < 0$**



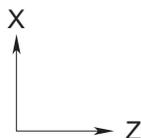
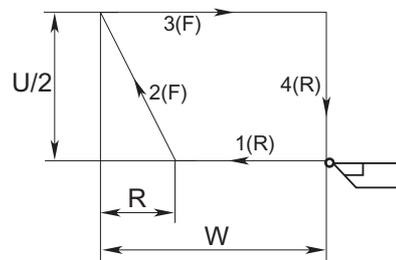
**2)  $U > 0, W < 0, R < 0$**



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

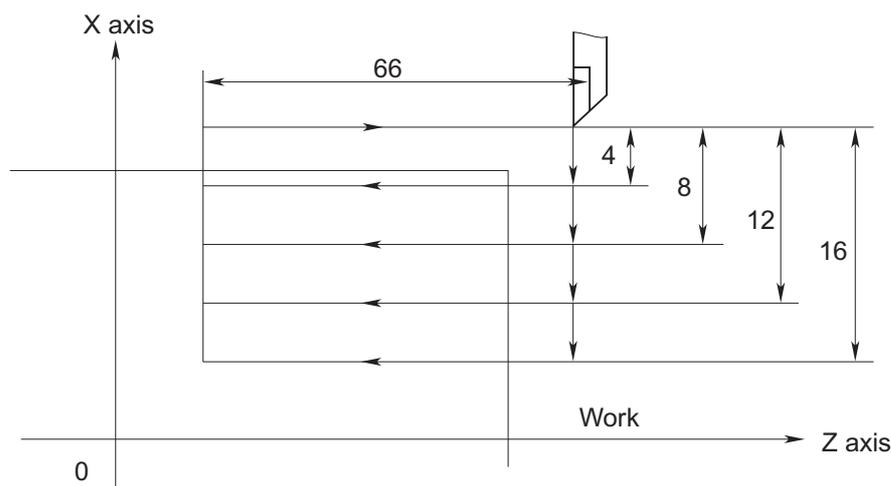


**4)  $U > 0, W < 0, R > 0$ , at  $|R| \leq \frac{U}{2}$**



***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;**

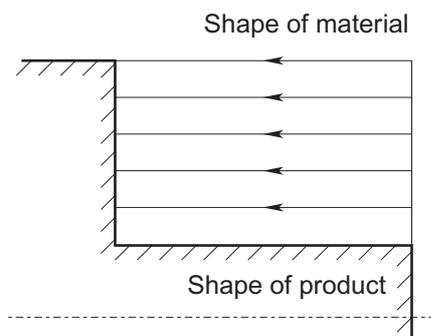
**N012 G90 X20500 Z1000;**

### 14.1.4 Usage of canned cycle

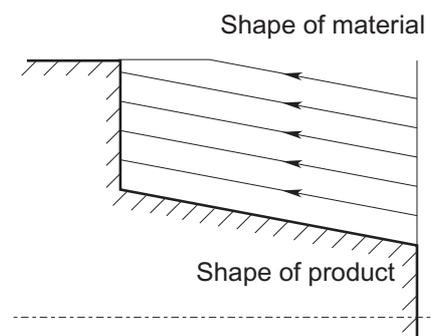
After .....

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

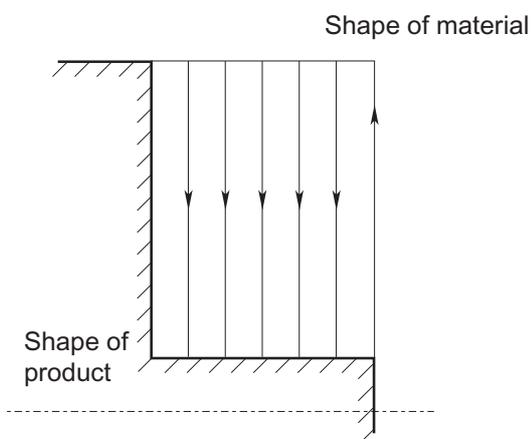
#### 1. Straight cutting cycle



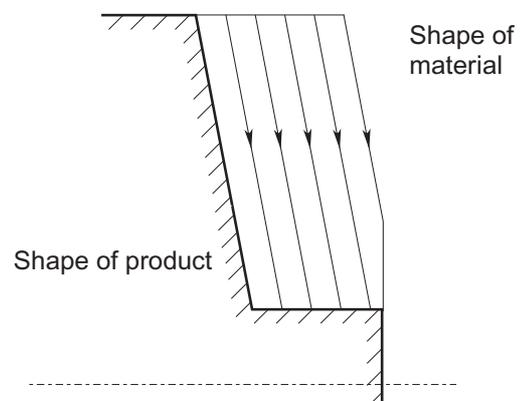
#### 2. Taper cutting cycle



#### 3. Face cutting cycle



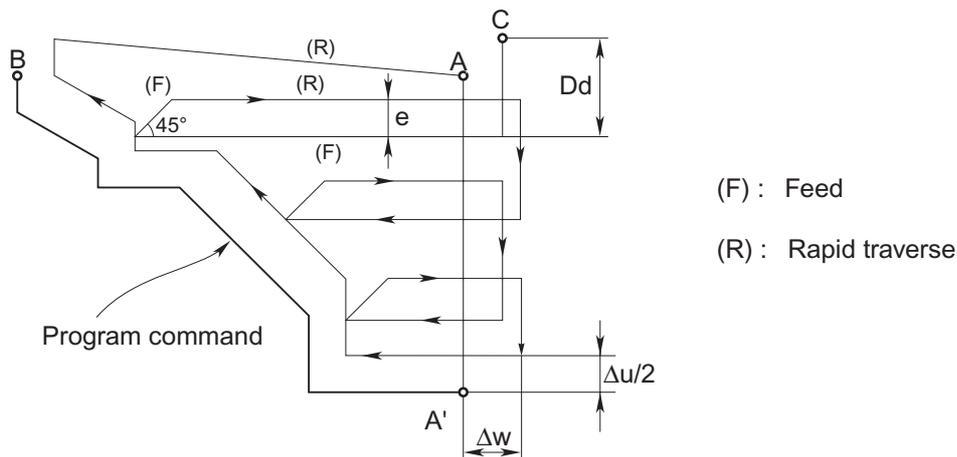
#### 4. Face taper cutting cycle



## 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  $\Delta d$  (depth of cut), with finishing allowance  $\Delta u/2$  and  $\Delta w$  left.



**G71 U( $\Delta d$ ) R( $e$ );**

**G71 P( $ns$ ) Q( $nf$ ) U( $\Delta u$ ) W( $\Delta w$ ) F( $t$ ) S( $s$ ) T( $t$ );**

*where:*

**$\Delta d$ :** depth of cut. Designate without sign. This designation is modal and is valid until another 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 another value is designated. Also this value can be specified by the parameter No.718 and the parameter is changed by the program command.

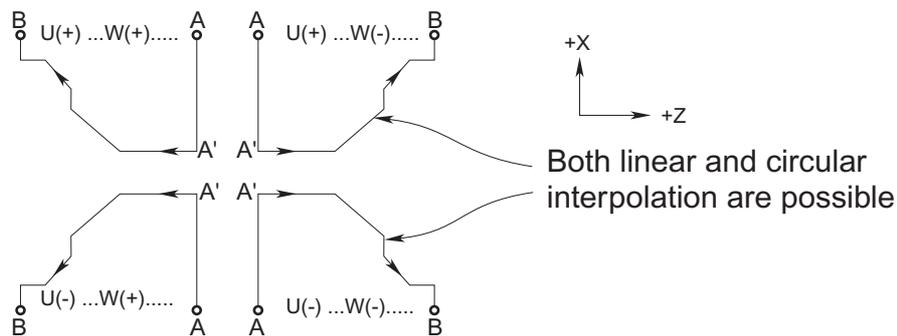
**$ns$ :** sequence number of the first block from the program of finishing shape

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

$\Delta u$ : distance and direction of finishing allowance in **X** direction

$\Delta w$ : distance and direction of finishing allowance 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  $\Delta u$  and  $\Delta w$  are as follows:



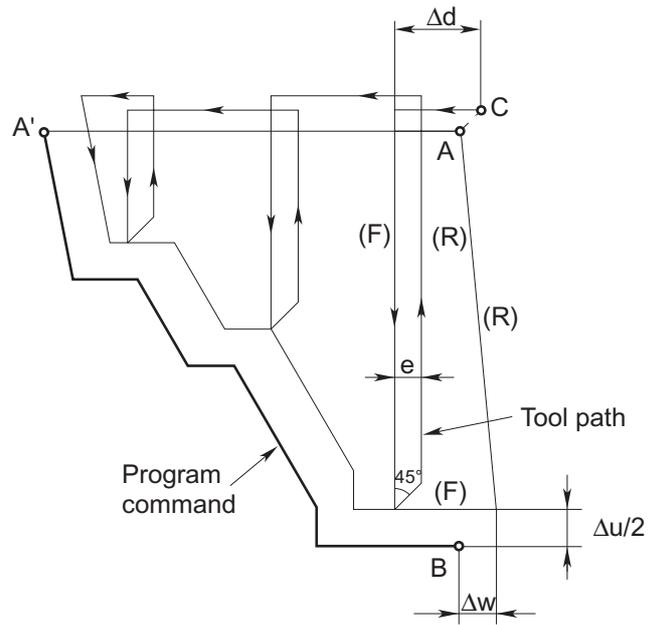
The tool path between **A** and **A'** is specified in the block with sequence number “**ns**” including **G00** or **G01**, and in this block a move command in the **Z** axis can not be specified.

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)

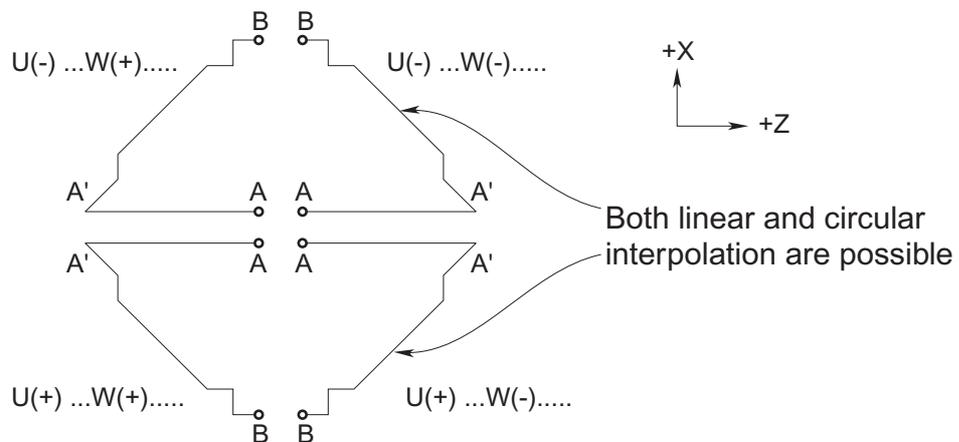
This cycle is the same as **G71** except that cutting is made by operations parallel to **X** axis.



Format: **G72 W( $\Delta d$ ) R( $e$ );**  
**G72 P( $ns$ ) Q( $nf$ ) U( $\Delta u$ ) W( $\Delta w$ ) F( $f$ ) S( $s$ ) T( $t$ );**

The meaning of  $\Delta d$ ,  $e$ ,  $ns$ ,  $nf$ ,  $\Delta u$ ,  $\Delta w$ ,  $f$ ,  $s$  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  $\Delta u$  and  $\Delta w$  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 **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.

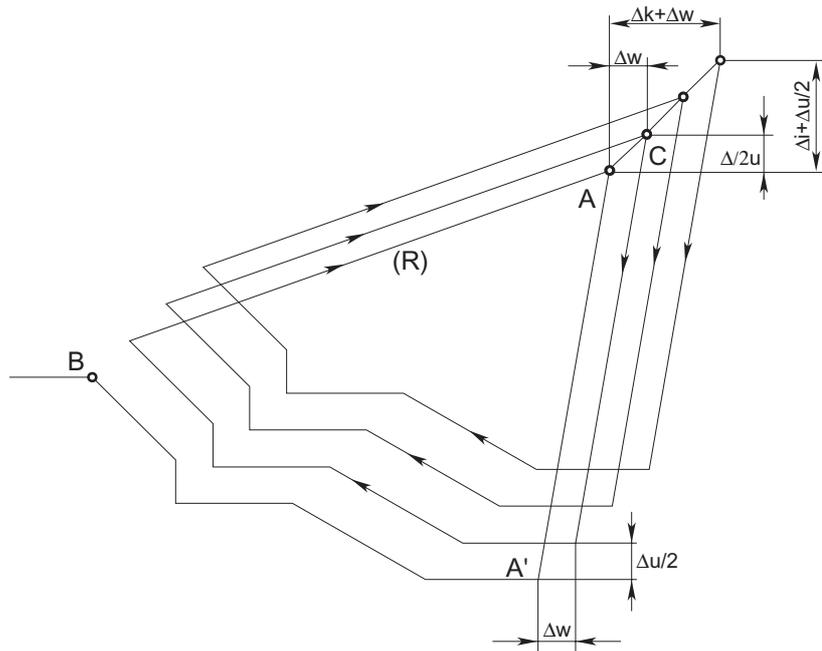
**F**, **S** and **T** functions in the blocks, which sequence number from "**ns**" to "**nf**", are ignored. Also in this area the **G96** and **G97** codes are not effected end calling of the subprograms is not valid.

Notes:

1. While both  $\Delta d$  and  $\Delta u$  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( $\Delta i$ ) W( $\Delta k$ ) R( $d$ );**

**G73 P( $ns$ ) Q( $nf$ ) U( $\Delta u$ ) W( $\Delta w$ ) F( $f$ ) S( $s$ ) T( $t$ );**

where:

- $\Delta i$ :** distance and direction of relief in **X** axis. This designation is modal and is not changed until another value is designated. Also this value can be specified by the parameter No.719, and the parameter is changed by the program command.
- $\Delta k$ :** 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 another 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

**$\Delta u$ :** distance and direction of finishing allowance in **X** axis

**$\Delta w$ :** distance and direction of finishing allowance 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 P(ns) Q(nf);**

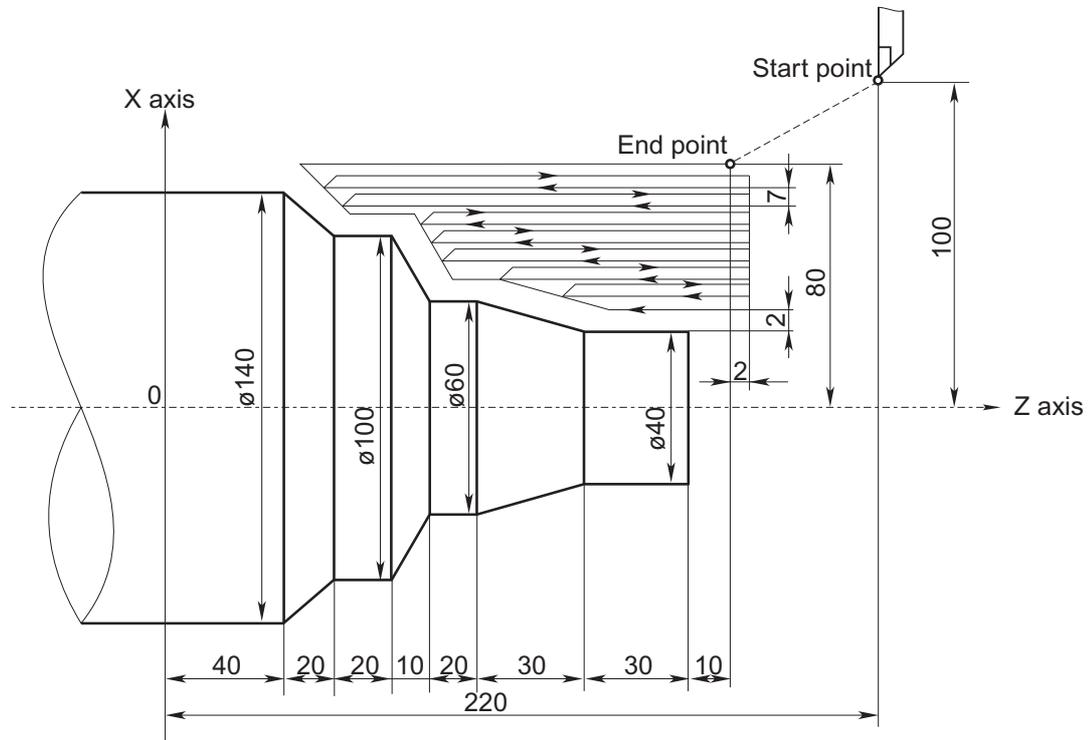
*where:*

**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, referred in **G70** through **G73**.

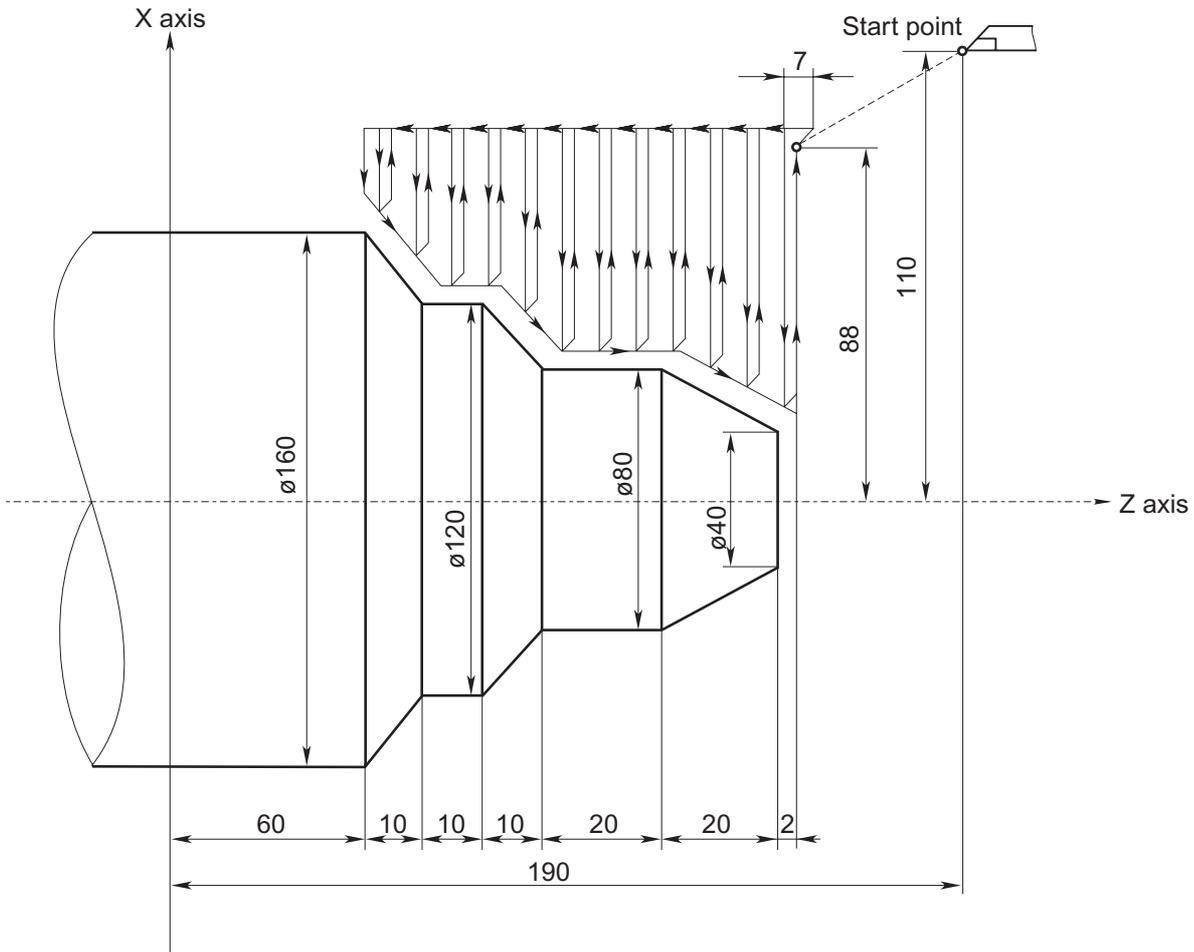
Example of programming by multiple repetitive cycle (**G70, G71**)



```

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**)

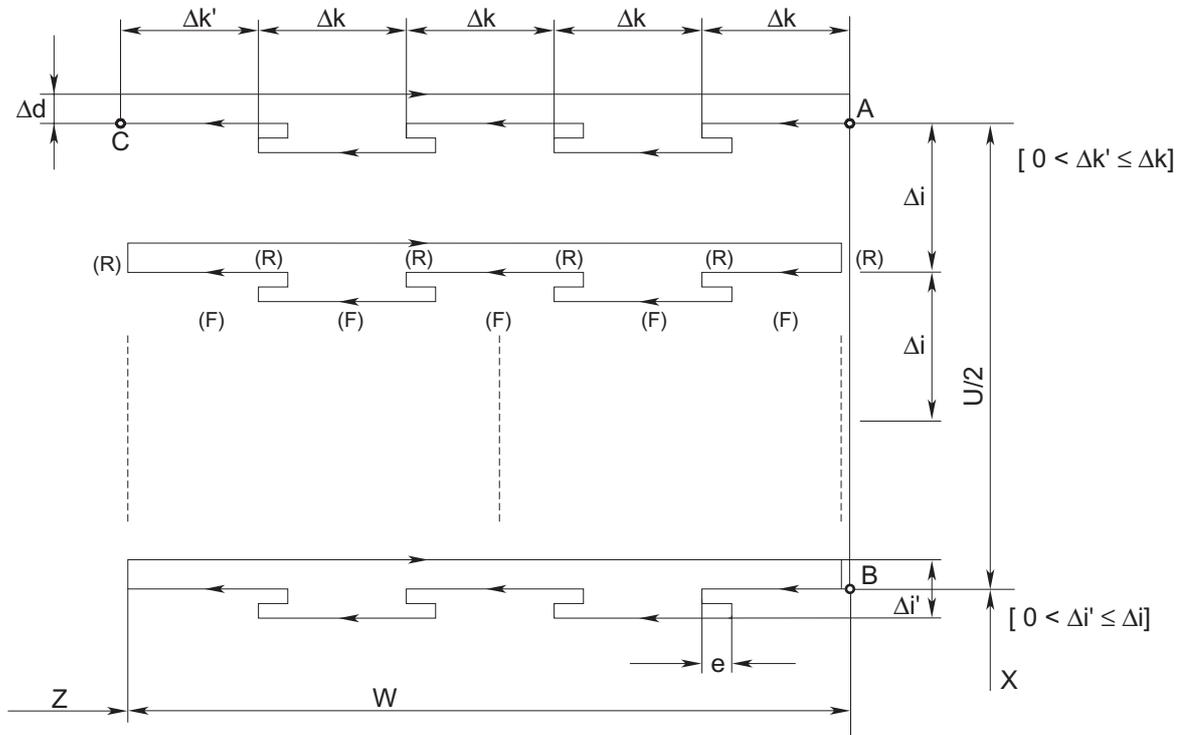


```

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**

**$\Delta i$ :** movement amount in **X** direction (without sign)

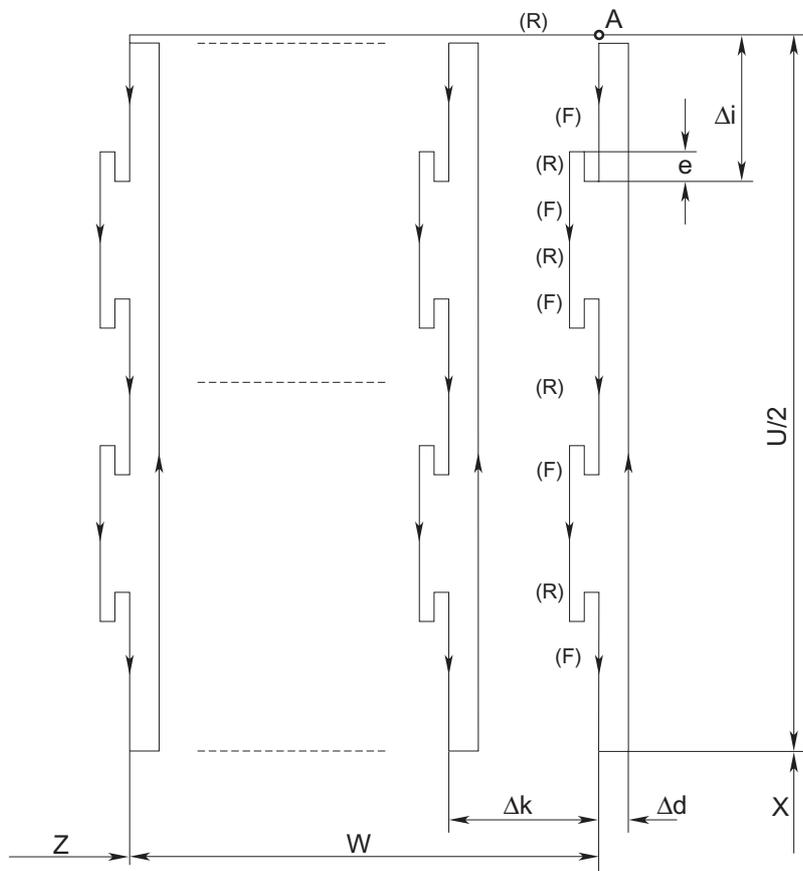
**$\Delta k$ :** depth of cut in **Z** direction (without sign)

**$\Delta d$ :** 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:*     **G76 P (m) (r) (a) Q( $\Delta$ d min) R(d);**  
                  **G76 X(U)\_\_\_ Z(W)\_\_\_ R(i) P(k) Q( $\Delta$ d) F(l);**

*where:*

**m:**     repeat count in finishing (1 to 99)

This value is modal and is valid until another 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.

**$\Delta$ dmin:** 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 allowance

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

**i:**     difference of thread radius

If  $i \neq 0$ , the taper thread can be made

**k:**     height of the tread

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

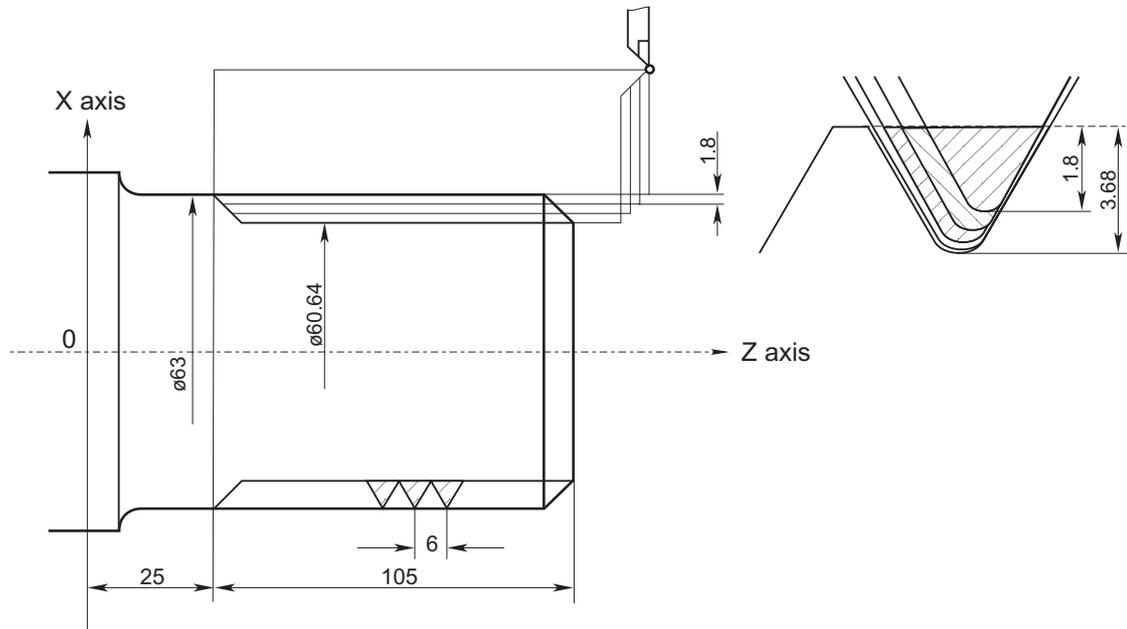
**$\Delta$ d:**   depth of cut in first cut

**l:**     lead of cut

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

**G76 P011060 Q100 R200 ;**

**G76 X60640 Z25000 P3680 Q1800 F6.0 ;**

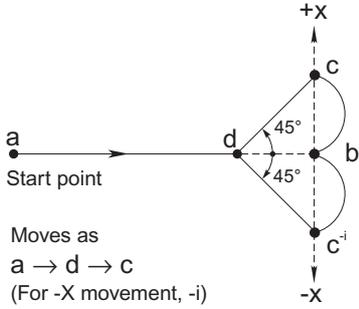
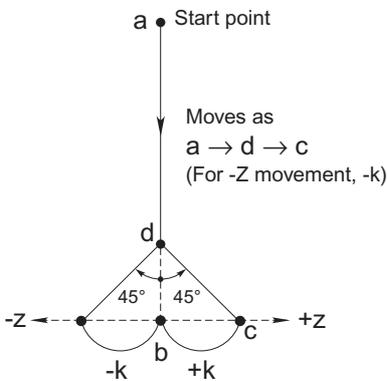
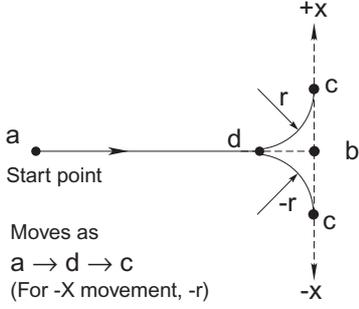
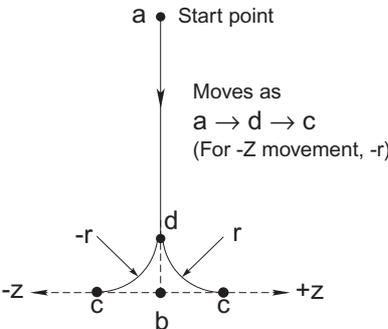


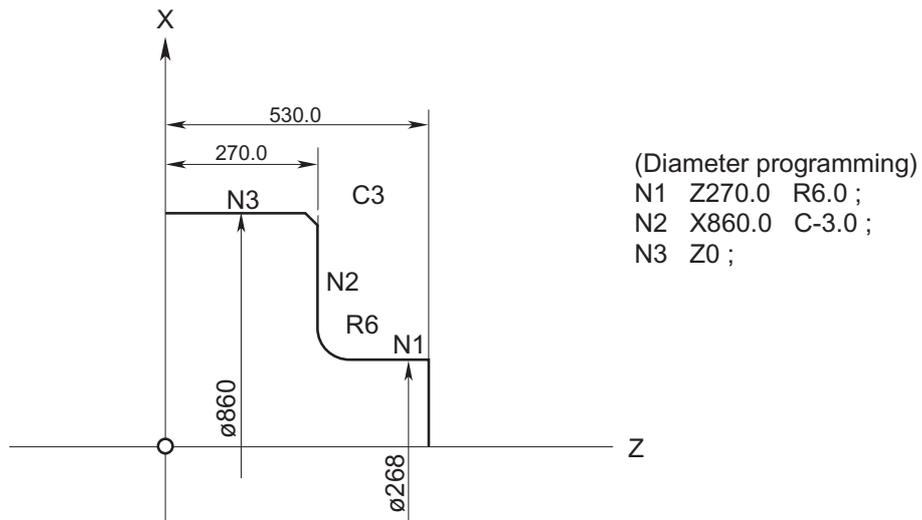
#### 14.2.8 Notes of multiple repetitive cycles (**G70** to **G76**)

- (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.

<p>Chamfering Z → X</p>	<p><math>G01Z(W) b/\hat{C} \pm i;</math> Specifies movement to point b with an absolute or incremental command in the figure on the right.</p>	
<p>Chamfering X → Z</p>	<p><math>G01X(U) b/\hat{C} \pm k;</math> Specifies movement to point b with an absolute or incremental command in the figure on the right.</p>	
<p>Corner R Z → X</p>	<p><math>G01ZbR\hat{\ } \pm r;</math> Specifies movement to point b with an absolute or incremental command in the figure on the right.</p>	
<p>Corner R X → Z</p>	<p><math>G01XbR\hat{\ } \pm r;</math> Specifies movement to point b with an absolute or incremental command in the figure on the right.</p>	



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:***

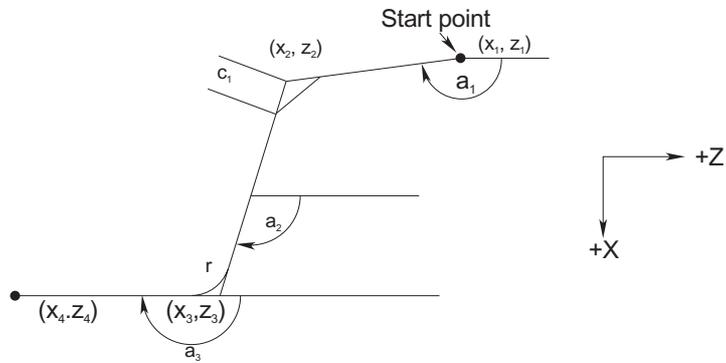
- (1) One **C** or **R** is commanded when **X** and **Z** are in a block, contents addresses **G01**(PS054).
- (2) The move amount of **X** or **Z** is less than chamfering **C** value and corner **R** value (PS55).
- (3) Next block to the block where chamfering and corner **R** were specified has not **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.



### 14.5 Direct Drawing Dimension Programming

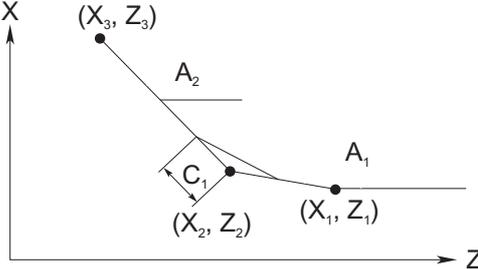
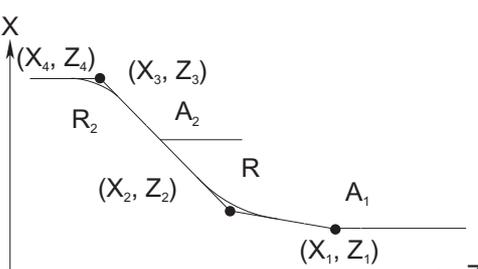
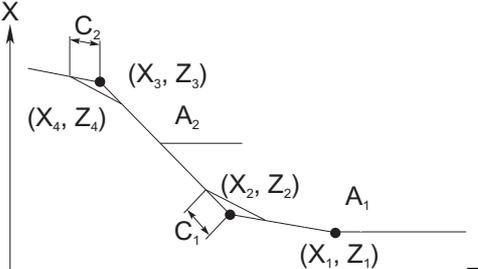
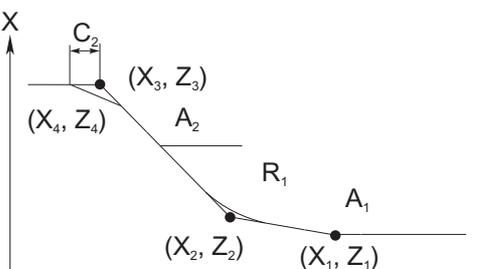
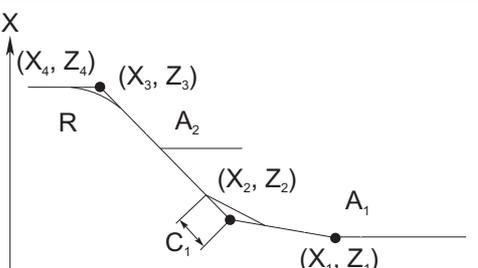
Angles of straight lines, chamfering value, corner rounding value and other dimensional values on machining drawings can be programmed by directly inputting these values.

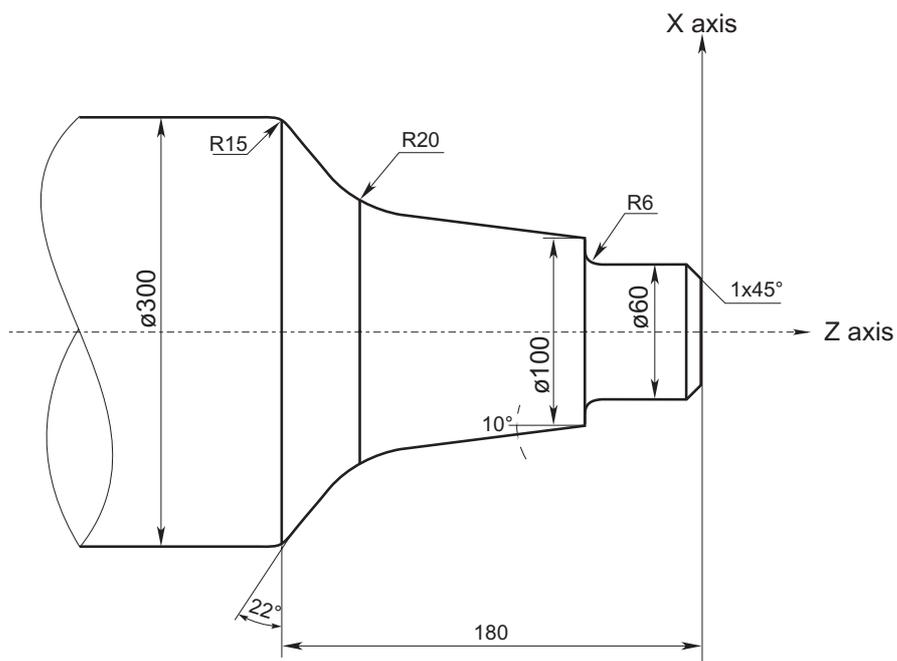


**$X(x_2) Z(z_2) C(c_1);$**       or       **$A(a_1) C(c_1);$**   
 **$X(x_3) Z(z_3) R(r_2)$**        **$X(x_3) Z(z_3) A(a_2) R(r_2);$**   
 **$X(x_4) Z(z_4);$**        **$X(x_4) Z(z_4);$**

Commands table

	Commands	Movement of tool
1	$X_2\_ Z_2\_ A\_;$	
2	$A_1\_;$ $X_3\_ Z_3\_ A_2\_;$	
3	$X_2\_ Z_2\_ R_1\_;$ $X_3\_ Z_3\_;$ or $A_1\_ R_1\_;$ $X_3\_ Z_3\_ A_2\_;$	

	Commands	Movement of tool
4	$X_2$ ___ $Z_2$ ___ $C_1$ ___; $X_3$ ___ $Z_3$ ___; or $A_1$ ___ $C_1$ ___; $X_3$ ___ $Z_3$ ___ $A_2$ ___;	 <p>The diagram shows a coordinate system with X and Z axes. A tool path starts at point <math>(X_3, Z_3)</math>, moves down and right to <math>(X_2, Z_2)</math>, then right to <math>(X_1, Z_1)</math>. Arcs <math>A_2</math> and <math>A_1</math> are shown. A center point <math>C_1</math> is indicated with a dashed line and a right-angle symbol.</p>
5	$X_2$ ___ $Z_2$ ___ $R_1$ ___; $X_3$ ___ $Z_3$ ___ $R_2$ ___; $X_4$ ___ $Z_4$ ___; or $A_1$ ___ $R_1$ ___; $X_3$ ___ $Z_3$ ___ $R_2$ ___; $X_4$ ___ $Z_4$ ___;	 <p>The diagram shows a coordinate system with X and Z axes. A tool path starts at point <math>(X_4, Z_4)</math>, moves down and right to <math>(X_3, Z_3)</math>, then down and right to <math>(X_2, Z_2)</math>, and finally right to <math>(X_1, Z_1)</math>. Arcs <math>R_2</math>, <math>R</math>, <math>A_2</math>, and <math>A_1</math> are shown.</p>
6	$X_2$ ___ $Z_2$ ___ $C_1$ ___; $X_3$ ___ $Z_3$ ___ $C_2$ ___; $X_4$ ___ $Z_4$ ___; or $A_1$ ___ $C_1$ ___; $X_3$ ___ $Z_3$ ___ $A_2$ ___ $C_2$ ___; $X_4$ ___ $Z_4$ ___;	 <p>The diagram shows a coordinate system with X and Z axes. A tool path starts at point <math>(X_4, Z_4)</math>, moves down and right to <math>(X_3, Z_3)</math>, then down and right to <math>(X_2, Z_2)</math>, and finally right to <math>(X_1, Z_1)</math>. Arcs <math>A_2</math> and <math>A_1</math> are shown. Center points <math>C_2</math> and <math>C_1</math> are indicated with dashed lines and right-angle symbols.</p>
7	$X_2$ ___ $Z_2$ ___ $R_1$ ___; $X_3$ ___ $Z_3$ ___ $C_2$ ___; $X_4$ ___ $Z_4$ ___; or $A_1$ ___ $R_1$ ___; $X_3$ ___ $Z_3$ ___ $A_2$ ___ $C_2$ ___; $X_4$ ___ $Z_4$ ___;	 <p>The diagram shows a coordinate system with X and Z axes. A tool path starts at point <math>(X_4, Z_4)</math>, moves down and right to <math>(X_3, Z_3)</math>, then down and right to <math>(X_2, Z_2)</math>, and finally right to <math>(X_1, Z_1)</math>. Arcs <math>R_1</math>, <math>A_2</math>, and <math>A_1</math> are shown. Center point <math>C_2</math> is indicated with a dashed line and a right-angle symbol.</p>
8	$X_2$ ___ $Z_2$ ___ $C_1$ ___; $X_3$ ___ $Z_3$ ___ $R_2$ ___; $X_4$ ___ $Z_4$ ___; or $A_1$ ___ $C_1$ ___; $X_3$ ___ $Z_3$ ___ $A_2$ ___ $R_2$ ___; $X_4$ ___ $Z_4$ ___;	 <p>The diagram shows a coordinate system with X and Z axes. A tool path starts at point <math>(X_4, Z_4)</math>, moves down and right to <math>(X_3, Z_3)</math>, then down and right to <math>(X_2, Z_2)</math>, and finally right to <math>(X_1, Z_1)</math>. Arcs <math>R</math>, <math>A_2</math>, and <math>A_1</math> are shown. Center point <math>C_1</math> is indicated with a dashed line and a right-angle symbol.</p>



***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  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  and  $270^\circ$  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;

```

.  
.  
.

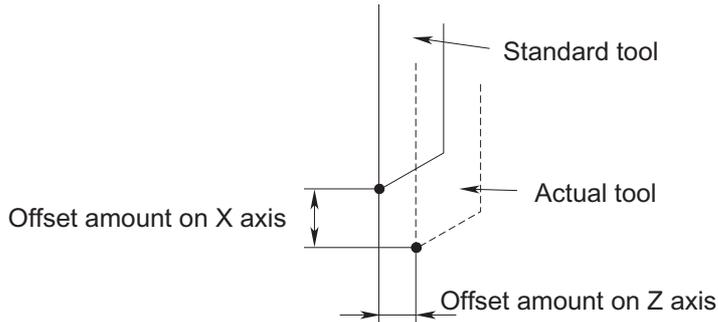
## 15.COMPENSATION FUNCTIONS

### 15.1 Tool Offset

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

#### 15.1.1 Basic Tool Offset

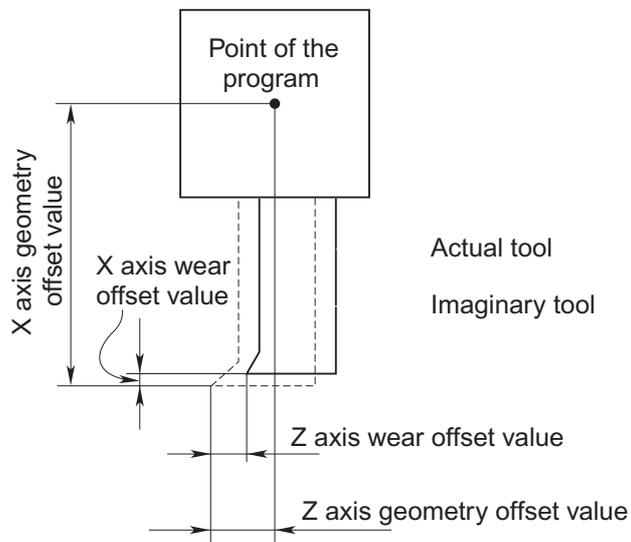
Tool offset is used to compensate for the difference when the tool actually used differs from the imagined tool used in programming (standard tool, usually)

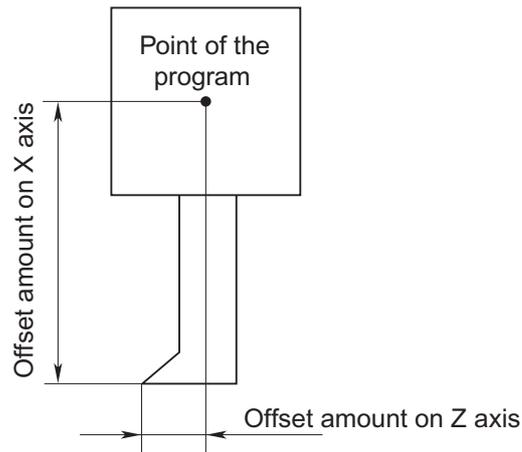


#### 15.1.2 Tool geometry offset and tool wear offset

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 the 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.

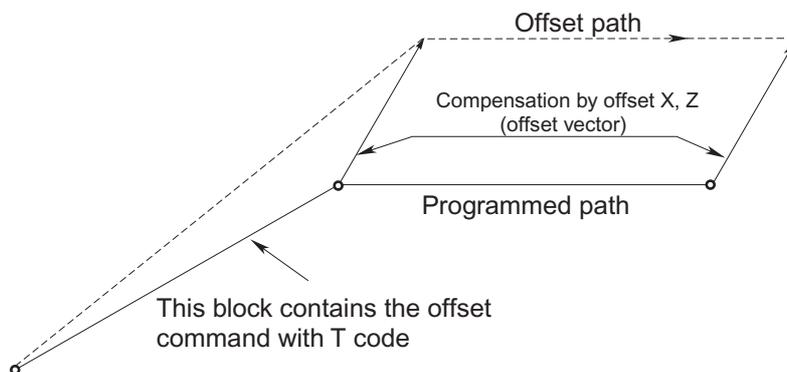
### 15.1.5 Offset number

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 transferring 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

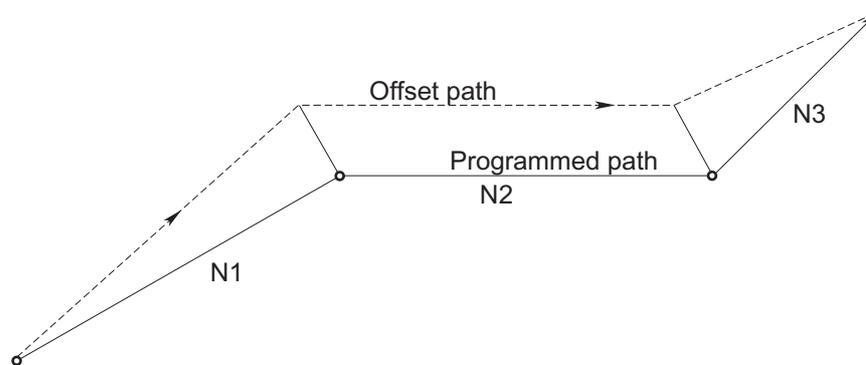
#### 15.1.6.1 Wear offset

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.

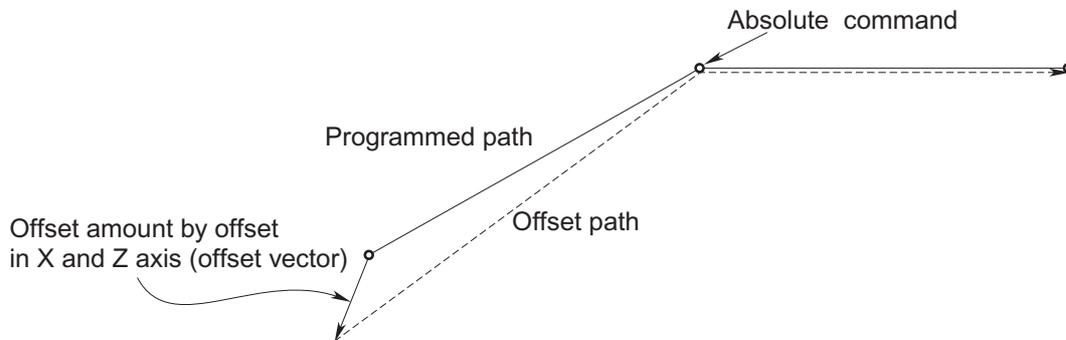
The tool will not move in the following block;

```
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.



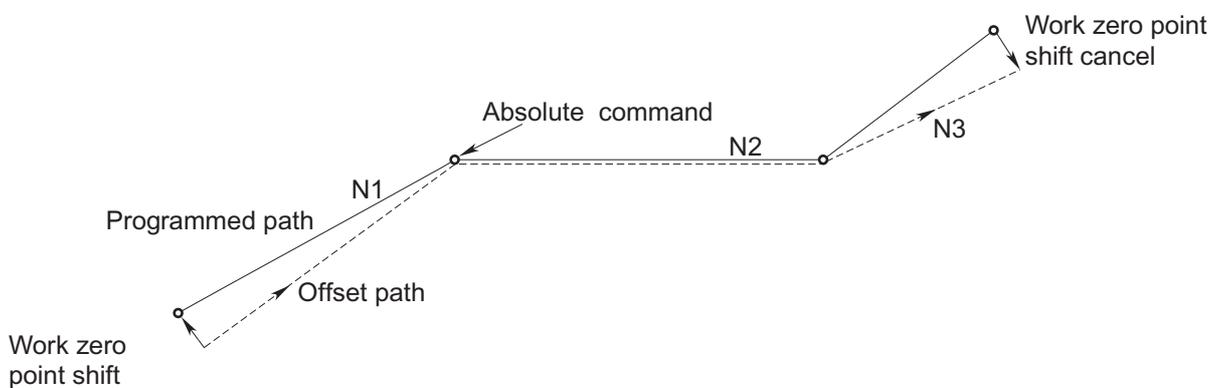
As well as wear offset, the geometry offset is determined by the parameter No.013 GMOFS whether to add or subtract the programmed end point of each block.

#### **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 GOFU=1.

*Example:*

```
N1 X50.0 Z100.0 T0202;  
N2 Z200.0;  
N3 X100.0 Z250.0 T020;
```

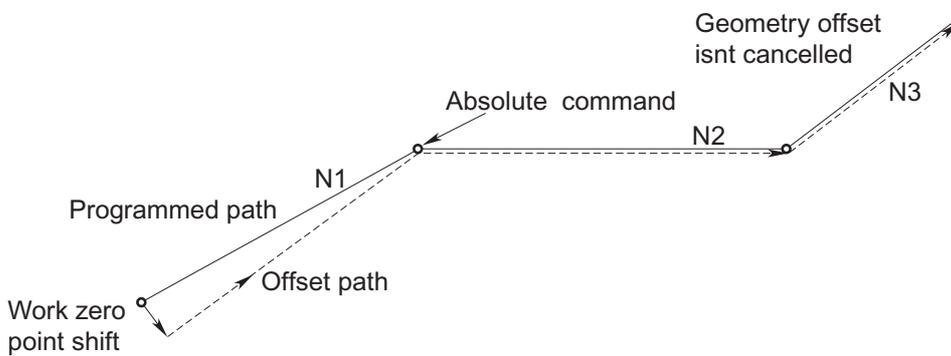


(Assume that there are offset amounts set at OFGX and OFGZ of the No. 02 geometry offset memory)

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

Example:

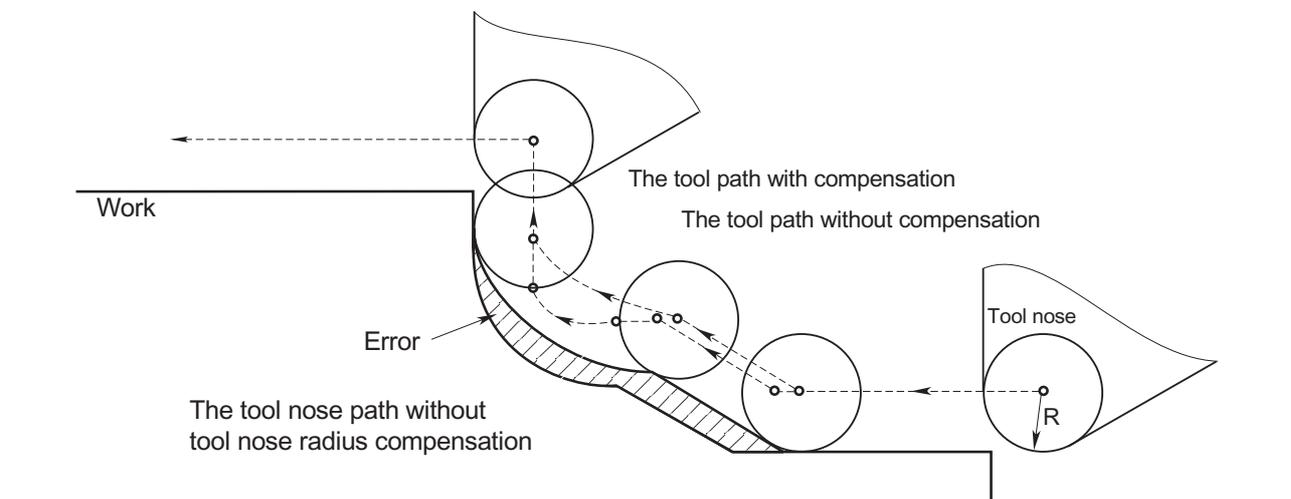
```
N1 X50.0 Z100.0 T0202;
N2 X200.0;
N3 X100.0 Z250.0 T0200;
```



(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.

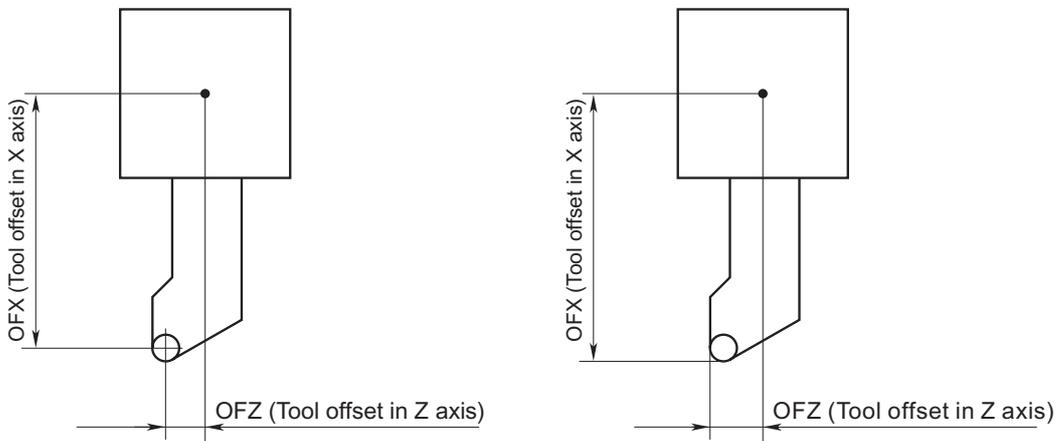


### 15.2.1 Imaginary tool nose

The tool nose at position **A** does not actually exist. 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.

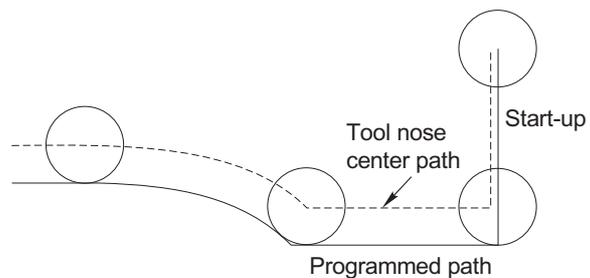
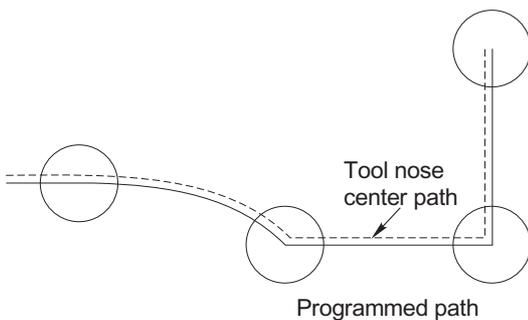
When the turret center is placed over the start point



#### (1) programming using the tool nose center

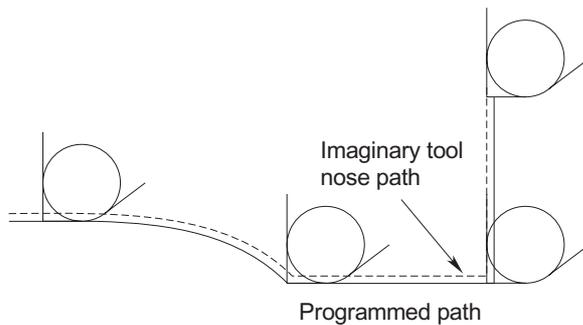
Unless tool nose radius compensation is performed, the tool nose center path is the same as the programmed path

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

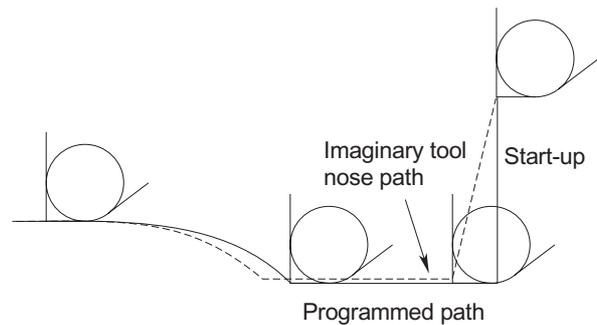


## (2) programming using imaginary tool nose

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

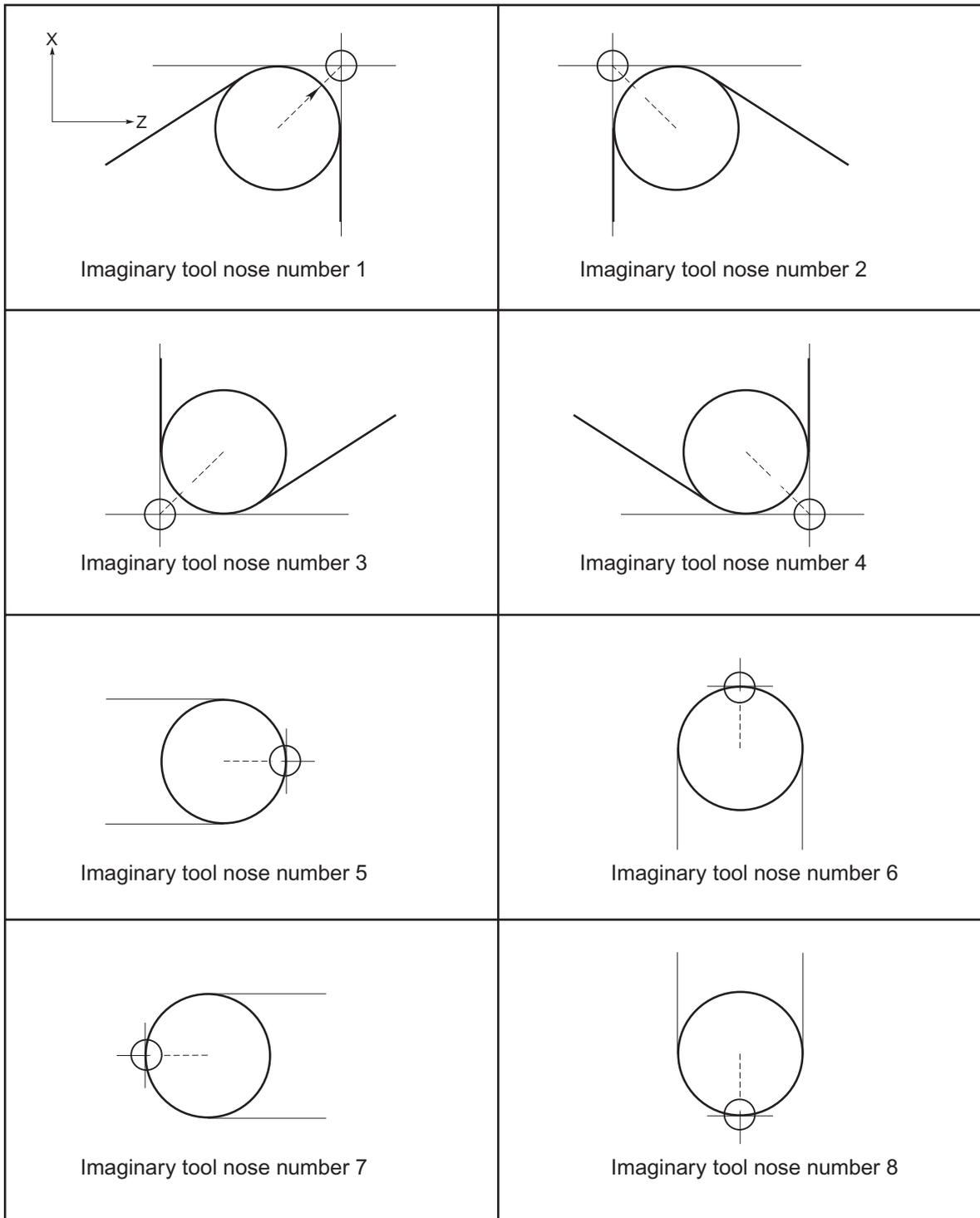
The direction of the imaginary tool nose viewed from the tool nose center is determined by the direction of the tool during cutting. This direction must be set in advance as well as the offset values.

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:

$$\text{OFR} = \text{OFGR} + \text{OFWR}$$

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.



### 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 (limaginary 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 (limaginary 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:

$$\text{OFR} = \text{OFGR} + \text{OFWR}$$

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.

*Example:*

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

The range of the offset values is:

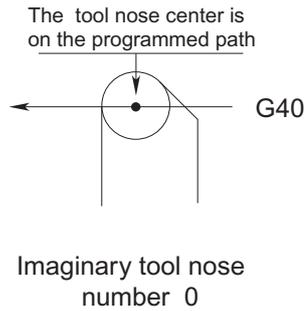
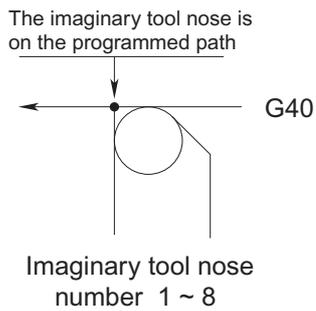
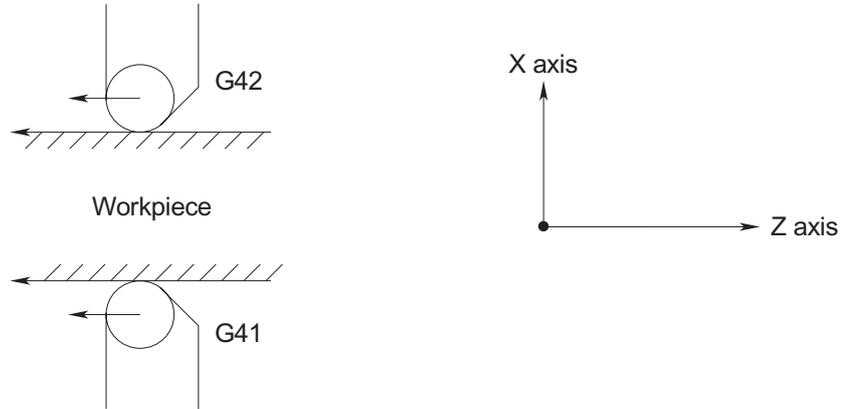
	mm input	inch system
Offset amount	0 - ±999.999mm	0 - ±99.9999inches

#### 15.2.4 Work position and move command

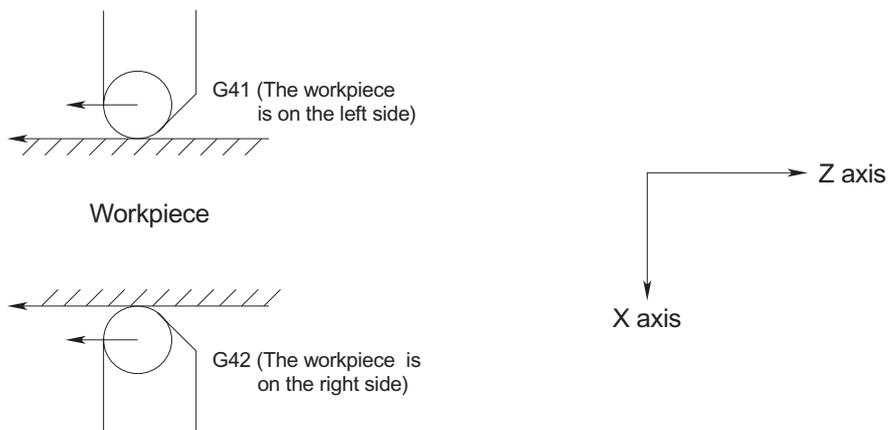
In tool nose radius compensation, the position of the workpiece in respect to the tool must be specified.

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



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

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

```

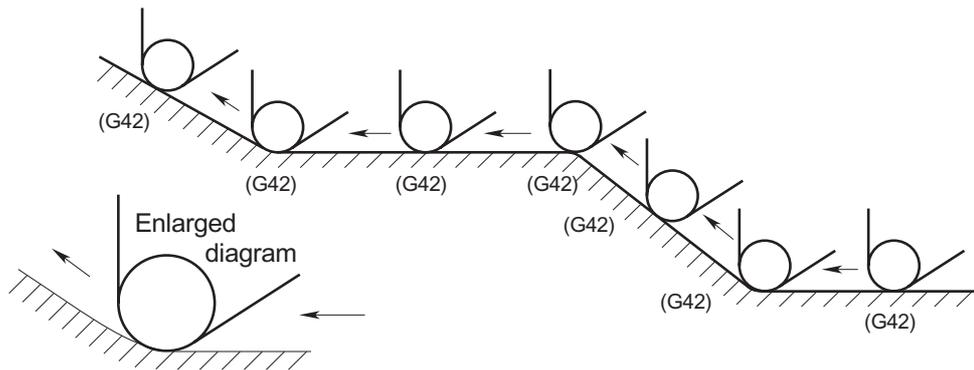
G41 X ..... Z ..... ;
    X ..... Z ..... ;
    X ..... Z ..... ;
G42 X ..... Z ..... ;
    X ..... Z ..... ;
    X ..... Z ..... ;
G40 X ..... Z ..... ;
    X ..... Z ..... ;
  
```

} G41 mode  
} G42 mode  
} G40 mode

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.

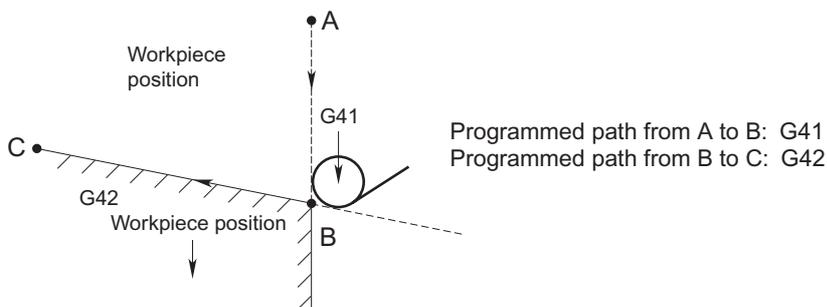
**(1)** *In the case of workpiece position does not change*

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:



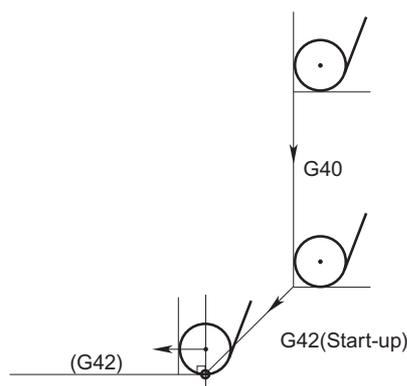
Although the workpiece does not exist on the right side of the programmed path in the above case, the existence of the workpiece is assumed in the movement from **A** to **B**. The workpiece position must not be changed in the block next to the start-up block. In the above example, if the block specifying 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 G41 or 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.

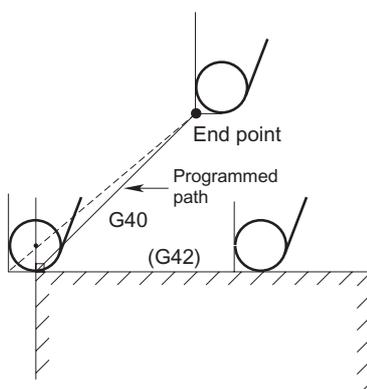


**(4) Offset cancel** (Start-up block)

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

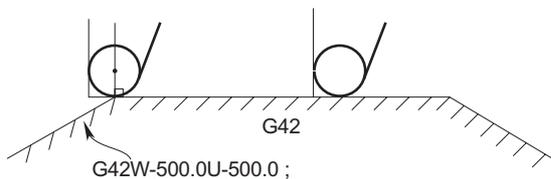
```
G41____;
____;
G40____;
```

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 preceding block at the end point of the preceding block.

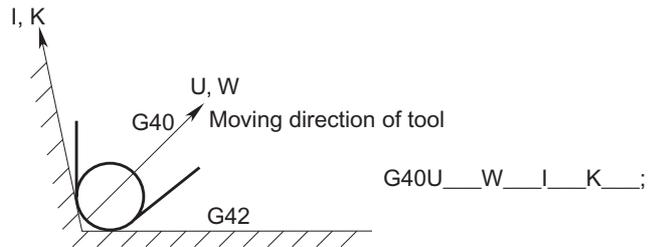


In the block that first specifies **G41/G42**, the above positioning of the tool nose center is not performed.

- (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

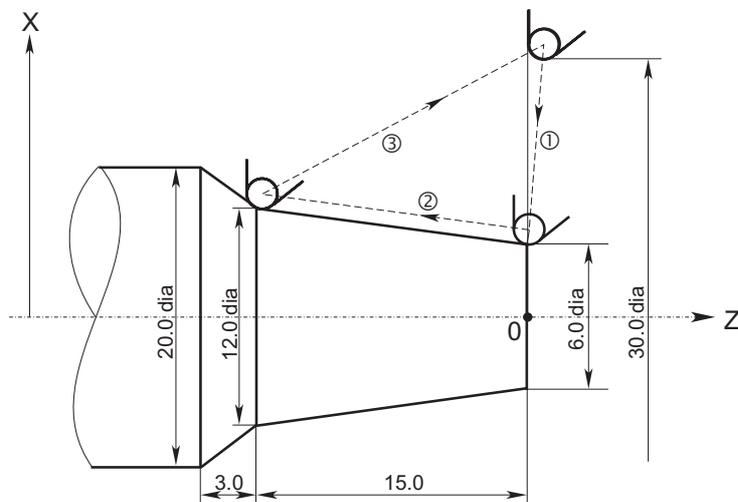
The workpiece position specified by addresses **I** and **K** is the same as that in the preceding block. If **I** and/or **K** is specified with **G40** in the cancel mode, the **I** and/or **K** is ignored.

**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;

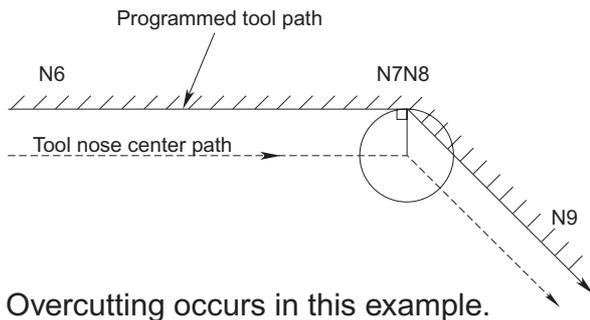
### 15.2.5 Notes on tool radius compensation

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

Blocks without a move commands are:

- |   |                                  |                              |
|---|----------------------------------|------------------------------|
| ① | <b>M05;</b>                      | <b>M code</b>                |
| ② | <b>S21;</b>                      | <b>S code</b>                |
| ③ | <b>G04 X1000;</b>                | <b>dwll</b>                  |
| ④ | <b>G01 U0;</b>                   | <b>feed distance of zero</b> |
| ⑤ | <b>G98;</b>                      | <b>G code only</b>           |
| ⑥ | <b>G10 P01 X100 Z200 R50 Q2;</b> | <b>offset change</b>         |

If two or more of the above blocks are specified consequentially, 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.



**(G42 mode)**

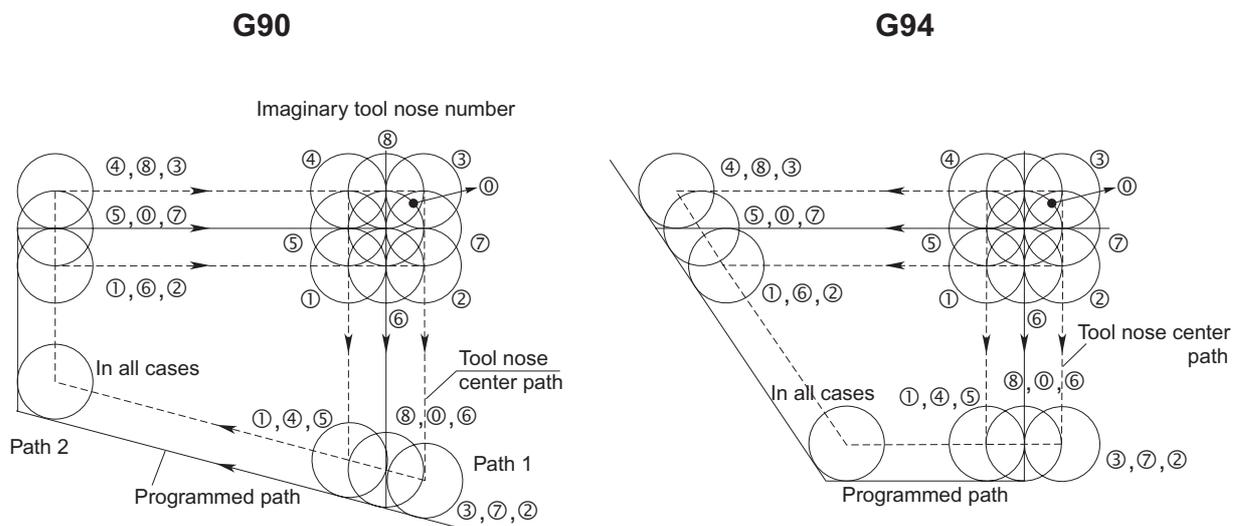
```
N6W1000.0;
N7S21;
N8M04;
N9U-1000.0W1000.0;
```

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

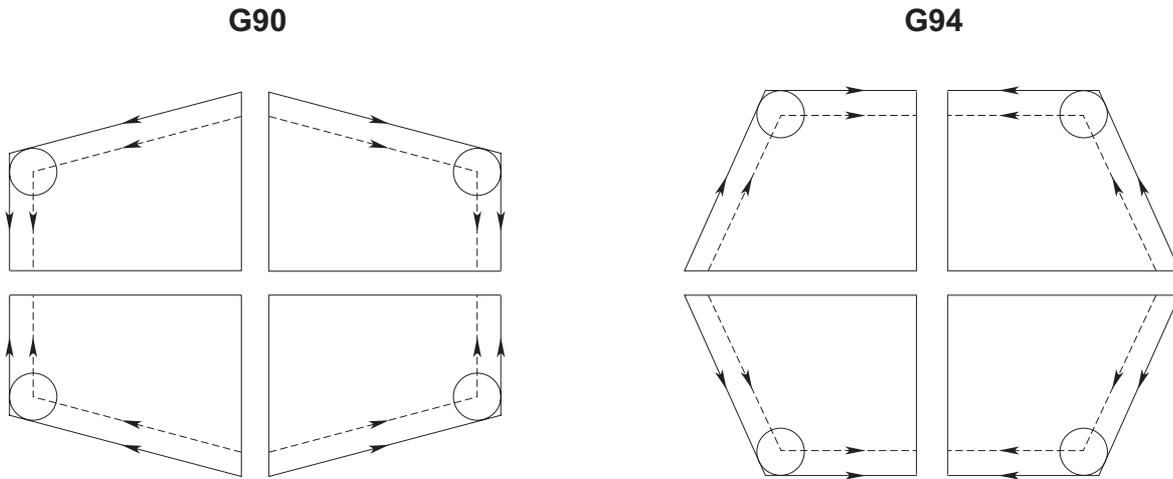
Tool nose radius compensation with **G90/G94** is as follows:

**- motion of the imaginary tool nose**

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



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



- **compensation with G71, G72 or G73**

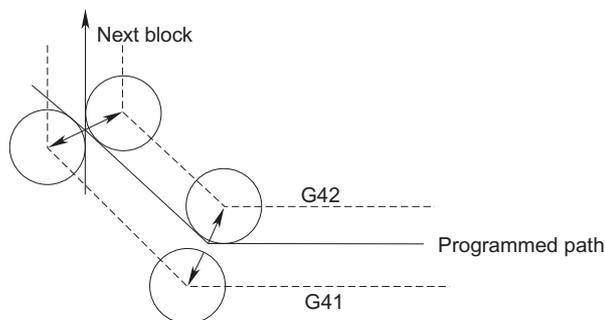
See 14.2.1.

- **when G74 or G76 or G78 is specified**

Tool nose radius compensation is not performed in this case.

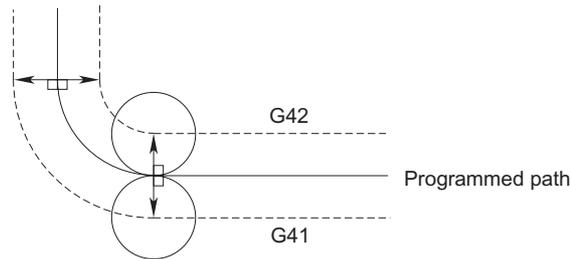
- **when chamfering is performed**

Movement after compensation is shown below.



**- when a corner arc is inserted**

Movement after compensation is as follows:

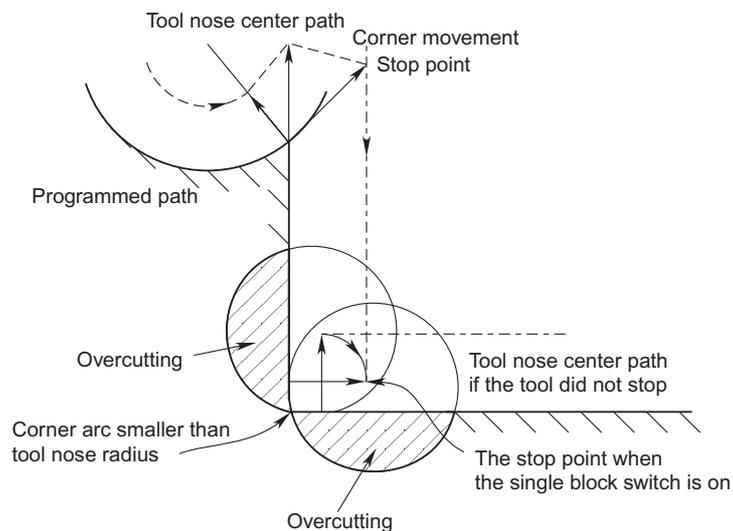


**- when the block is specified from the MDI**

Tool nose radius compensation is not performed in this case.

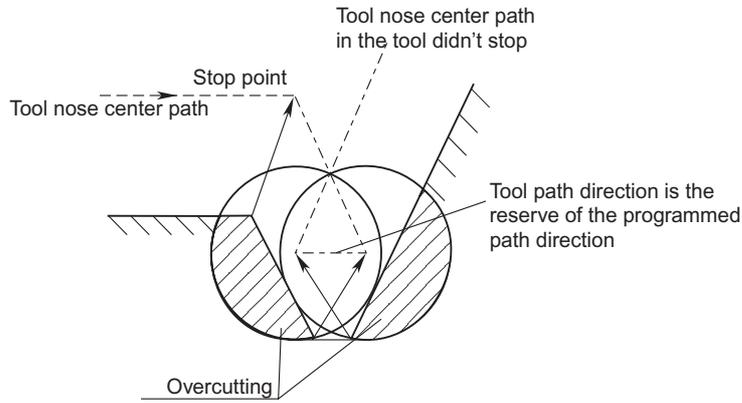
**- when machining at an inside corner smaller than the tool nose radius**

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.



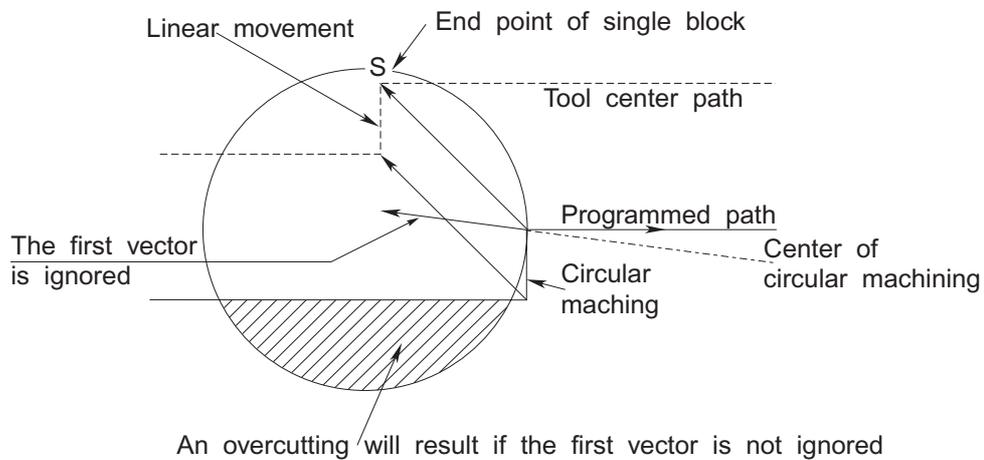
**- machining a groove smaller than the tool nose radius**

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

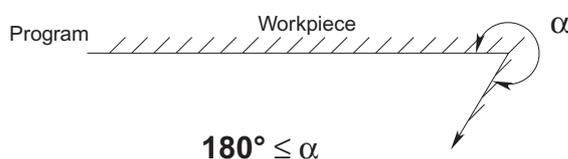
- **X** or **Z** moves are specified in the block and move distance is not zero

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

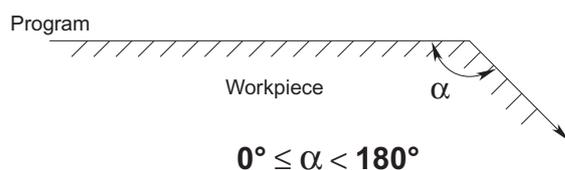
Two blocks are read during the start-up. The first block is executed, and the second block is entered into the tool nose radius compensation buffer. In the SINGLE BLOCK mode, two blocks are read and when the first one is executed, the machine stops.

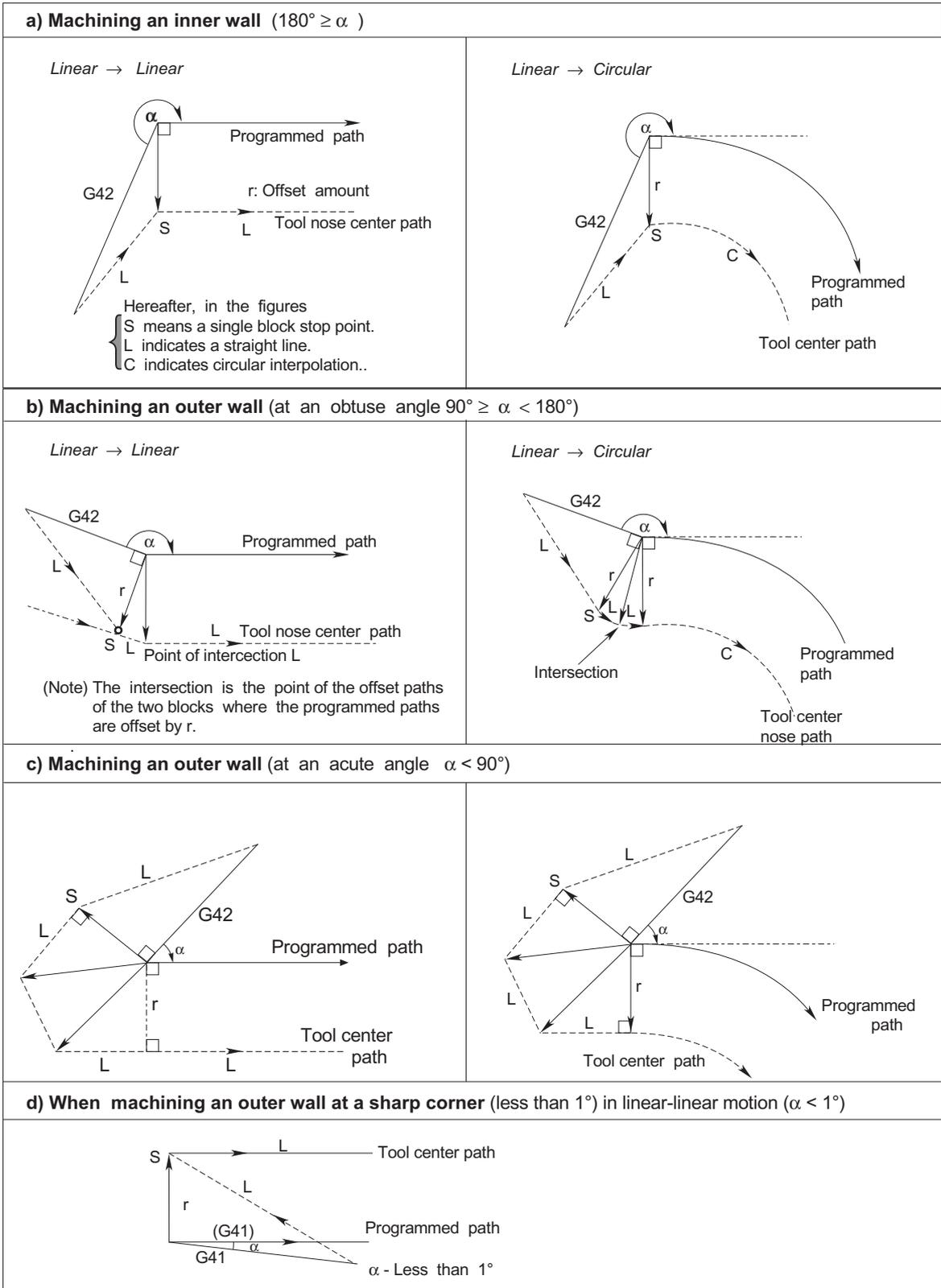
The meaning of “*inner-side*” and “*outer-side*” used later is as follows: An angle of intersection created by two blocks of move commands is referred to as “*inner-side*” when it is over 180° and “*outer-side*”, when it is from 0° to 180°.

#### (1) Inner-side



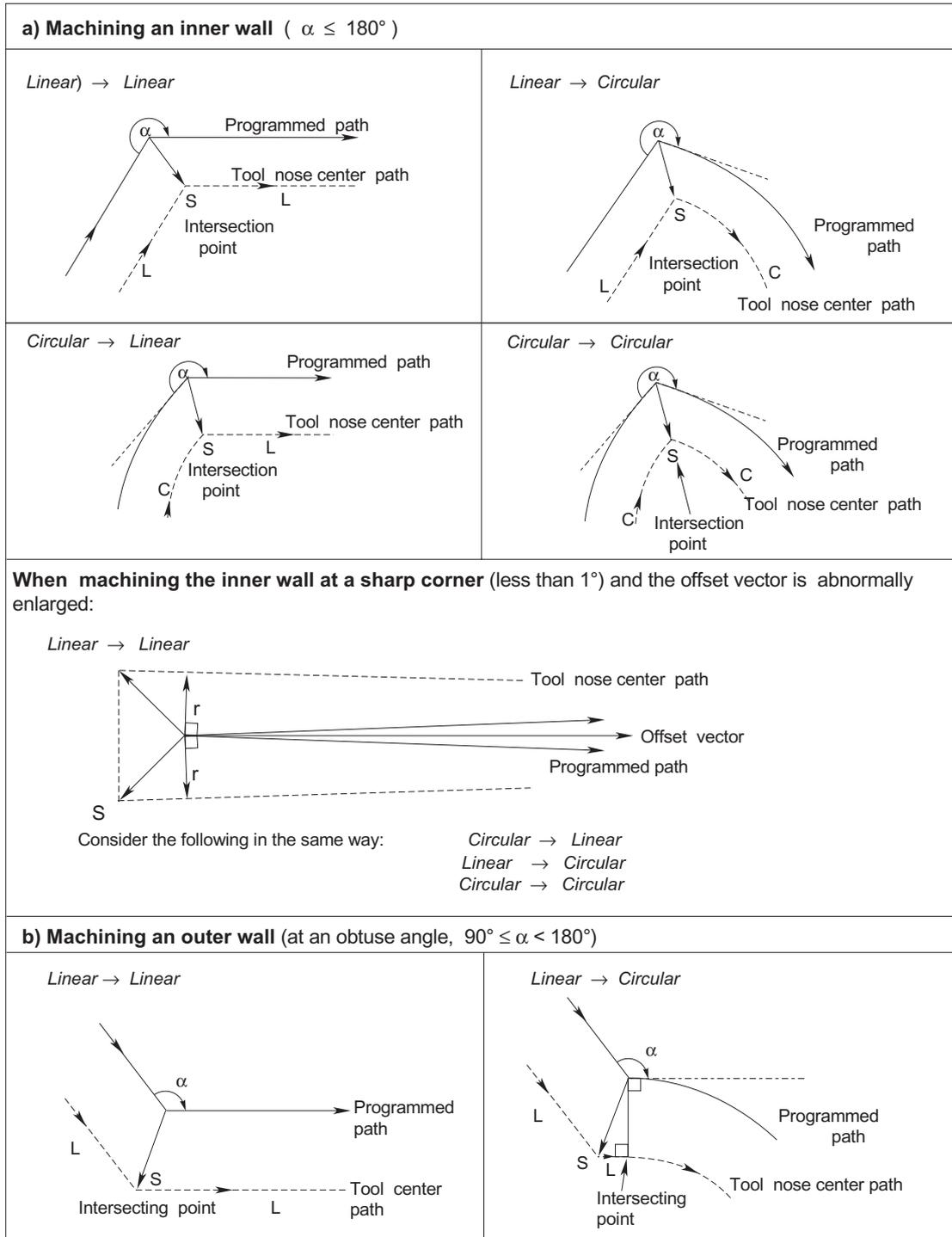
#### (2) outer-side

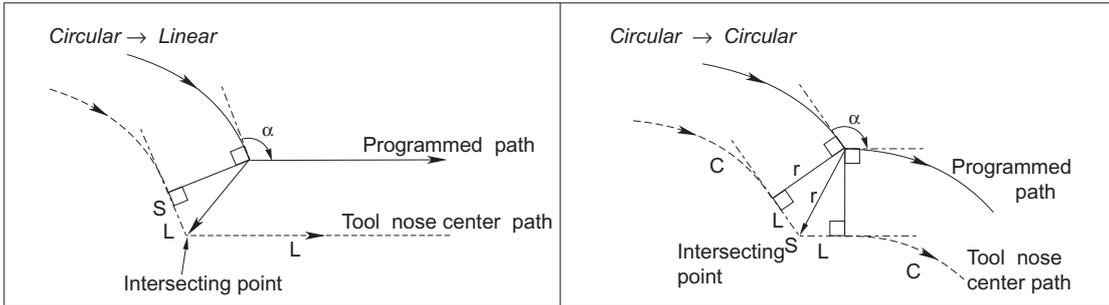




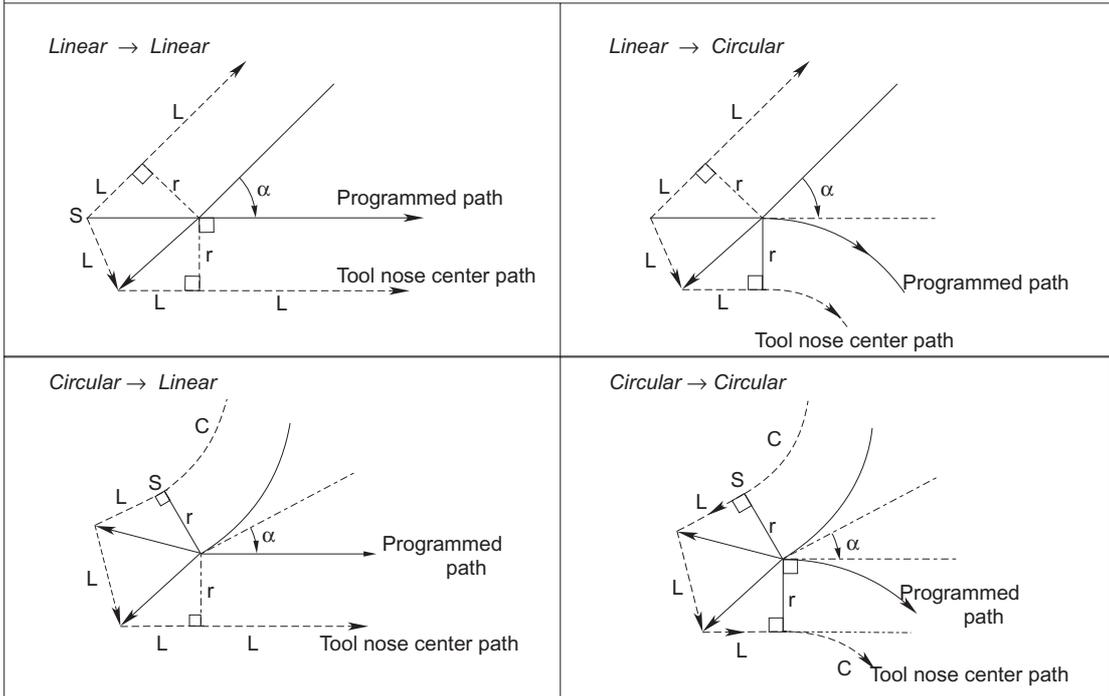
### (3) Offset mode

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). Otherwise, overcutting or insufficient cutting will result.



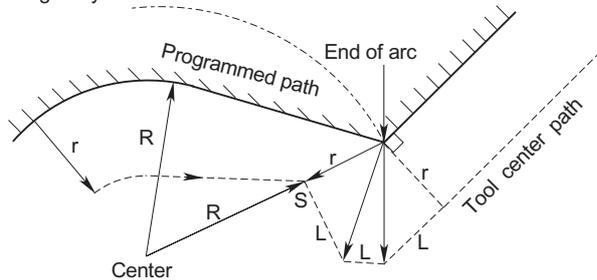


**c) Machining an outer wall (at an acute angle,  $\alpha < 90^\circ$ )**

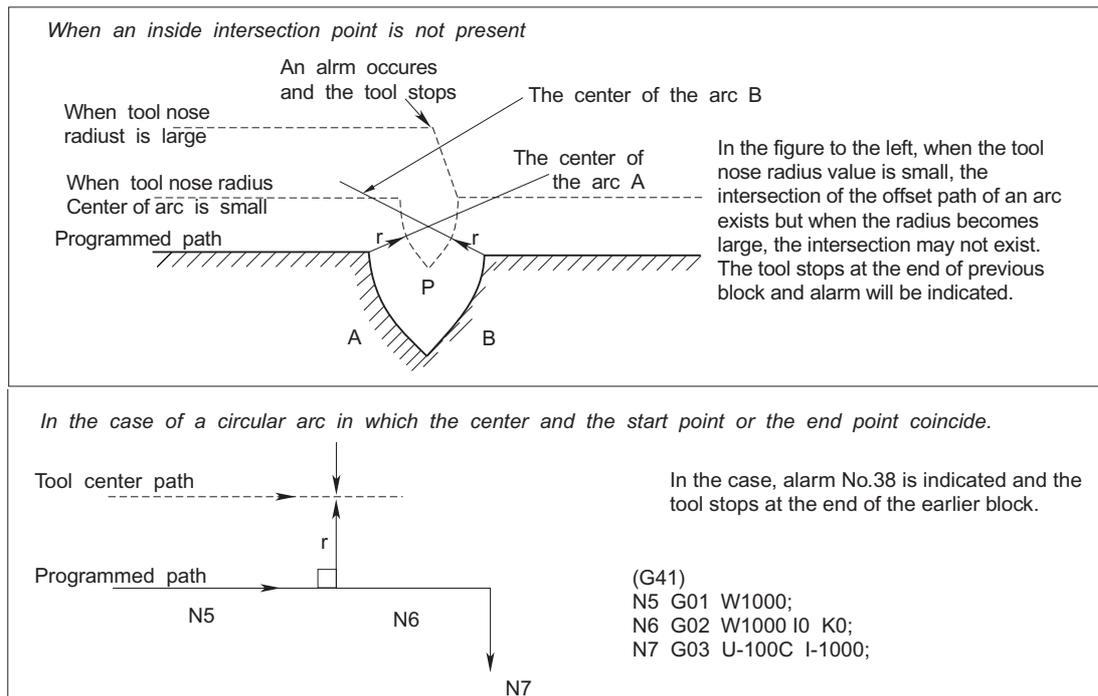


**d) Exceptional case**

When the end point is not on the arc  
Imaginary circle



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.

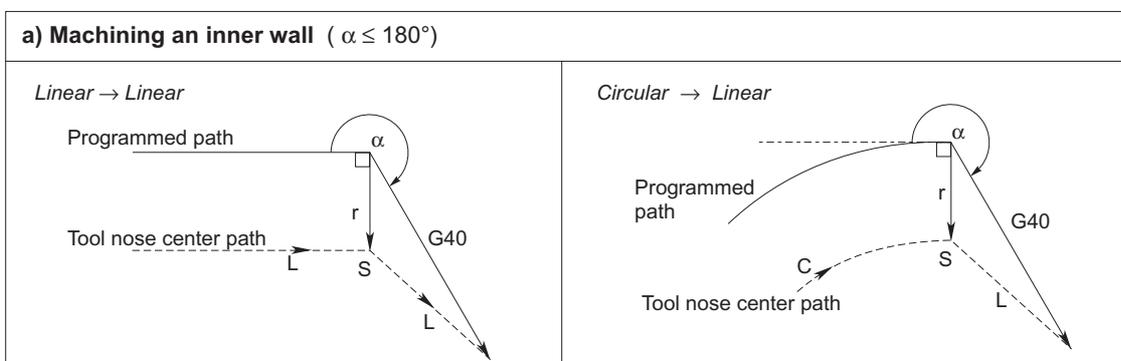


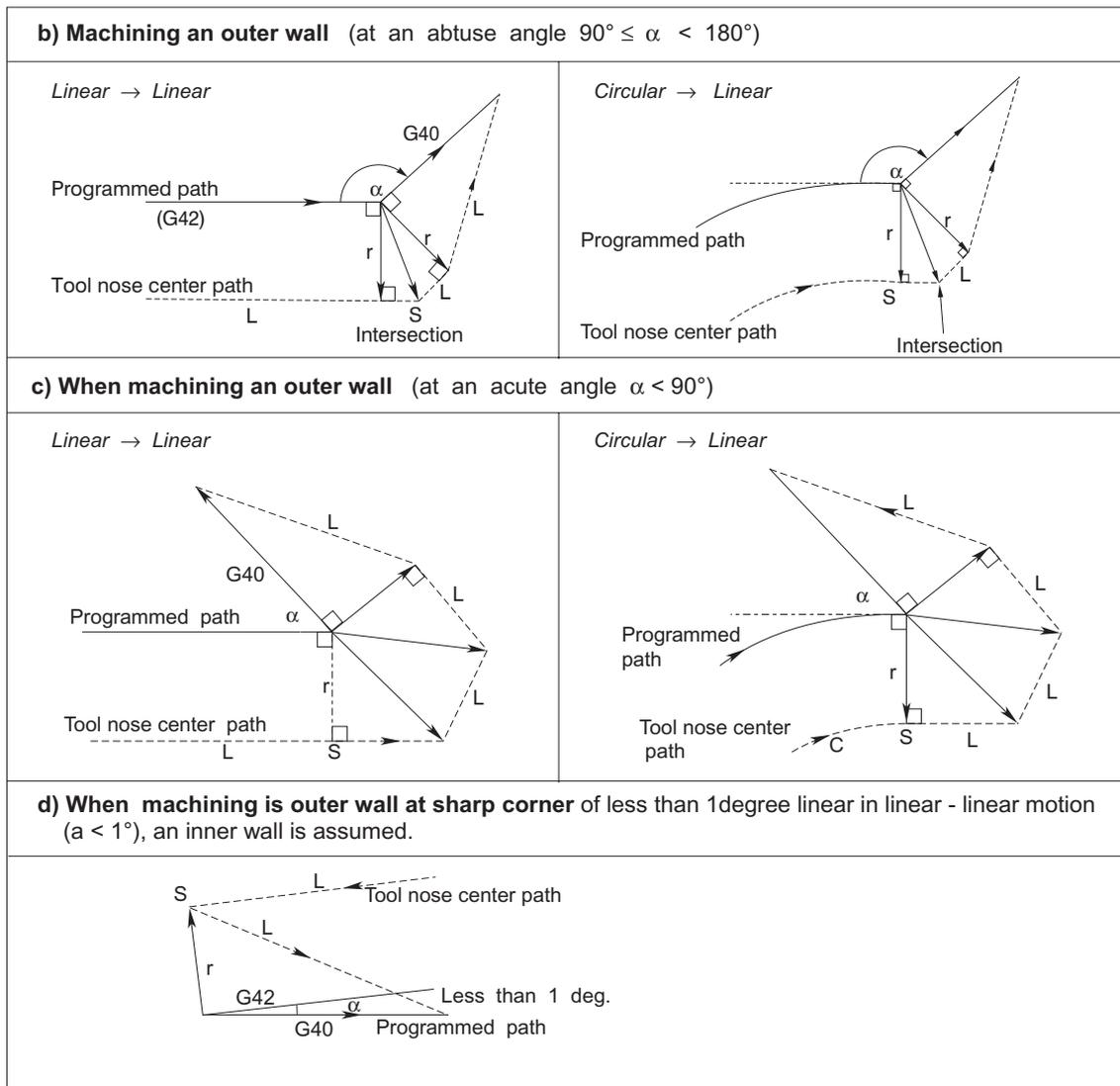
#### 4) Offset cancel

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.



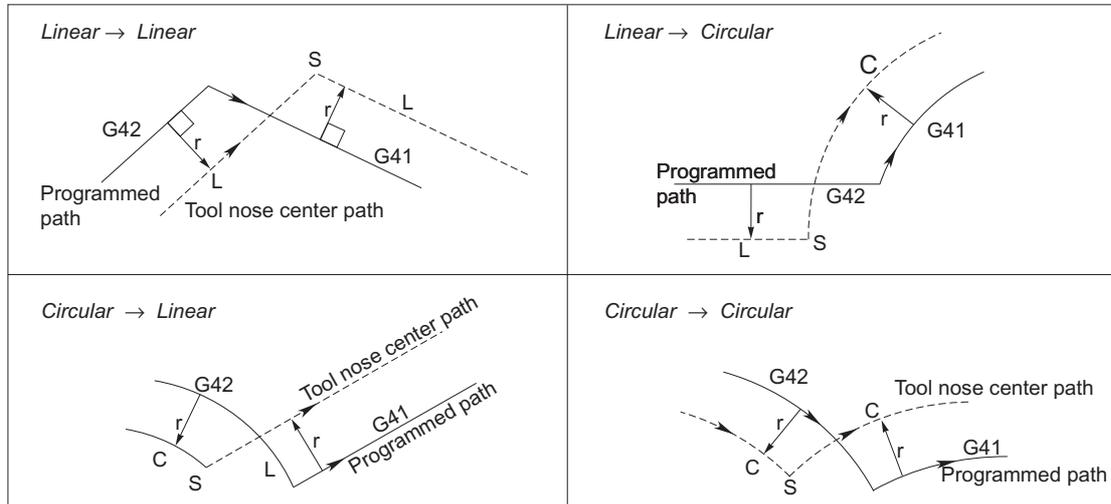


### 5) Change of offset direction in offset mode

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

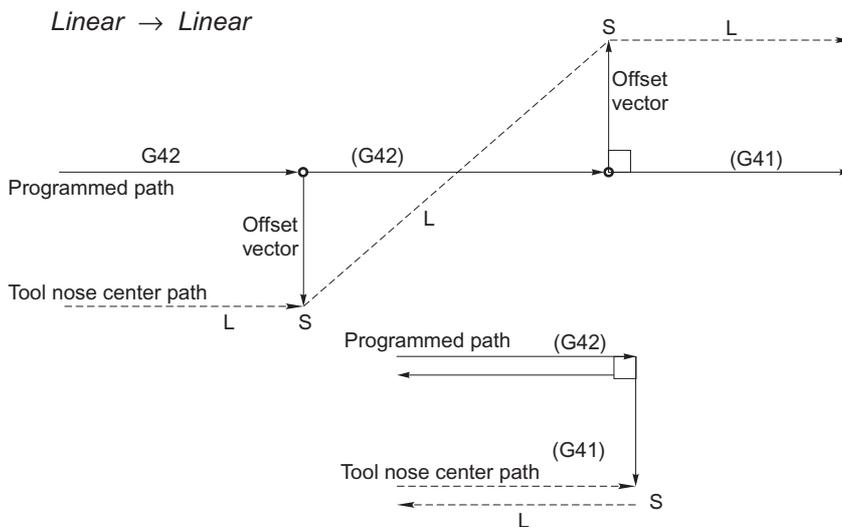
G code	Sign of offset value	
	+	-
G41	Left offset	Right offset
G42	Right offset	Left offset

The following drawings 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.

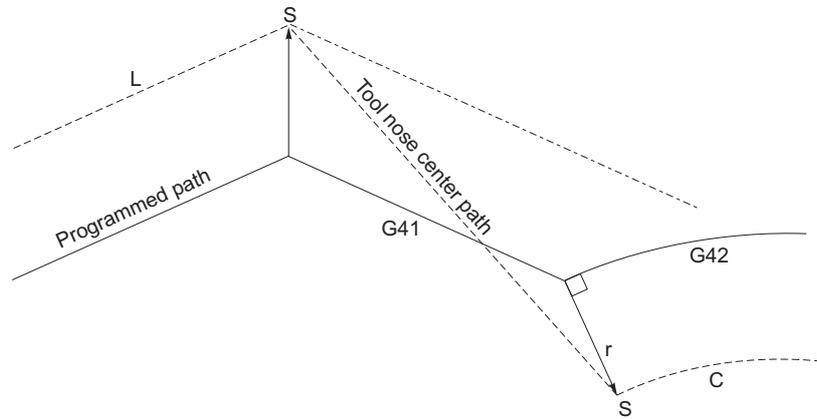


**When an intersection is not obtained if offset is normally performed**

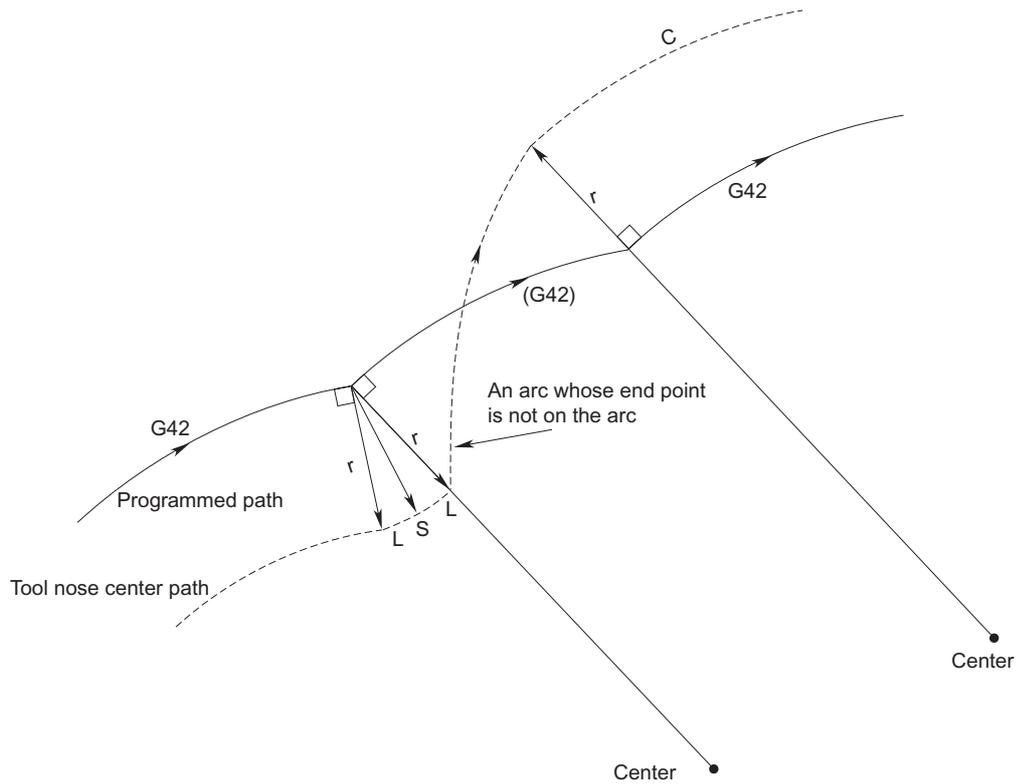
When changing the offset from block **A** to block **B** using **G41** and **G42**, if intersection with the offset path is not obtained, the vector normal to block **B** is created at the start point of block **B**.



Linear → Circular



Circular → Circular

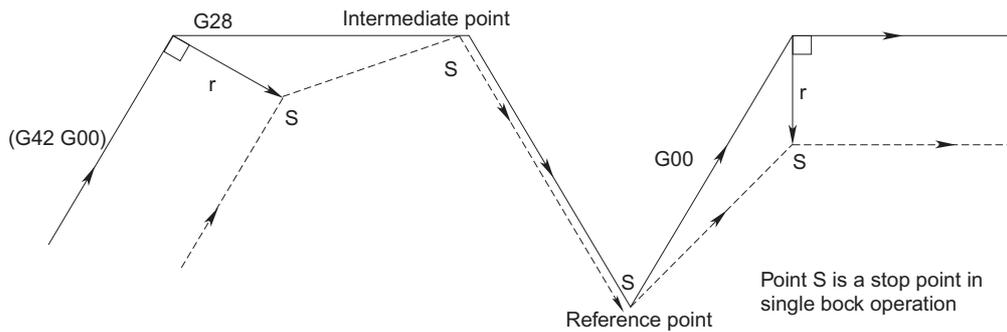


### 6) Temporary offset cancel

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

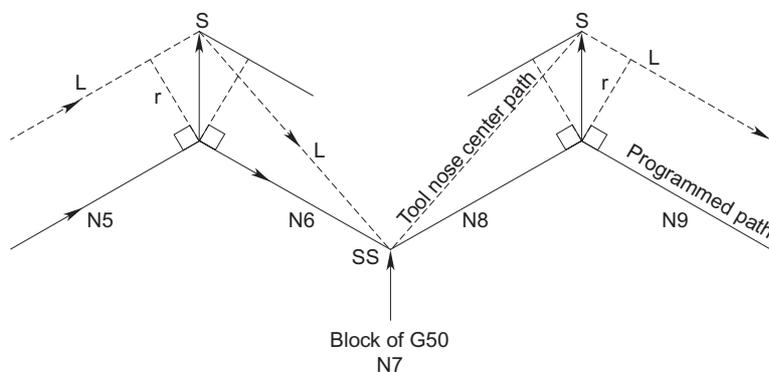
If **G28** is specified in offset mode, offset will be cancelled at the intermediate point, and the offset mode will be automatically restored after reaching the reference point.



### 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;**

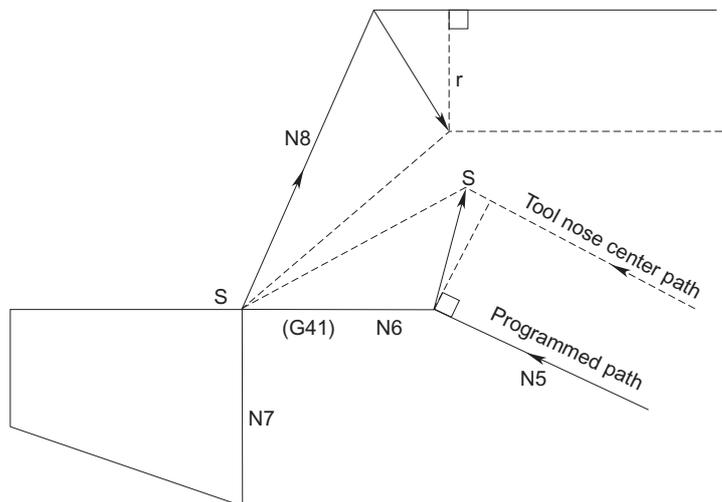
**N6 U-300.0 W600.0;**

**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)**

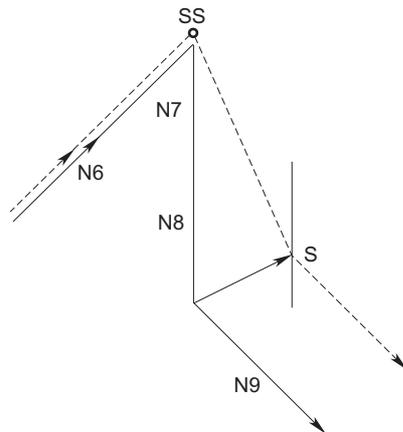
### 8) A block not specifying tool movement

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

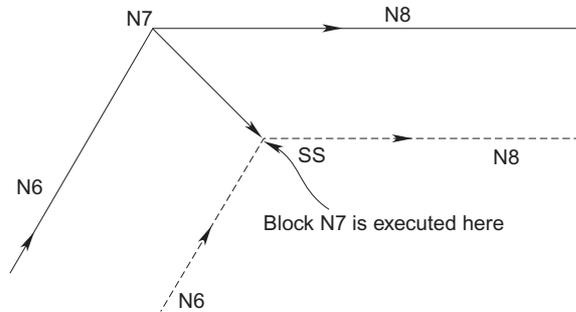
If a block not specifying tool movement is input at start-up, the offset vector is not produced.



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

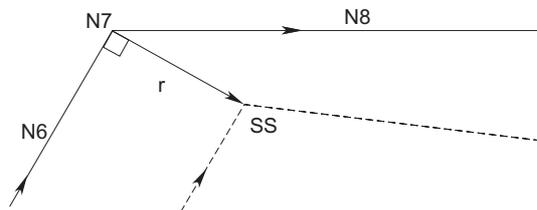
**b) when specified in offset mode**

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.



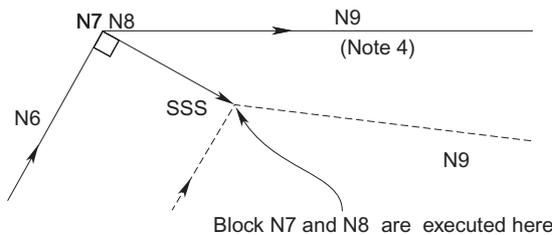
```
N6 G91U2000W1000;
N7 G04P1000;
N8 W1000;
```

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 length 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.

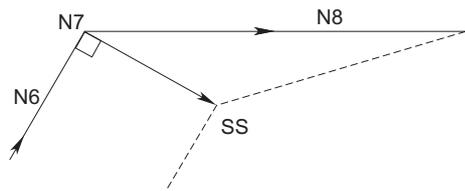


```
N6 G91 X100.0 Y200.0;
N7 S21;
N8 G04 X1.0;
N9 X100.0;
```

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

**c) when specified with offset cancel command**

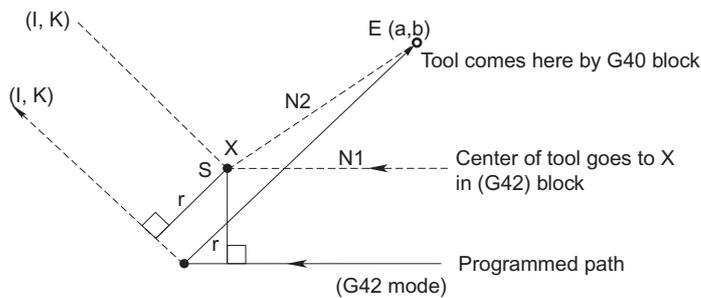
When a block not specifying tool movement is input with an offset cancel command, a vector whose length is equal to the offset value is produced in a direction normal to the tool motion specified in the preceding block. This vector is cancelled when the next command is executed.



```
N6 U100.0W100.0;
N7 G40;
N8 U0W10.0;
```

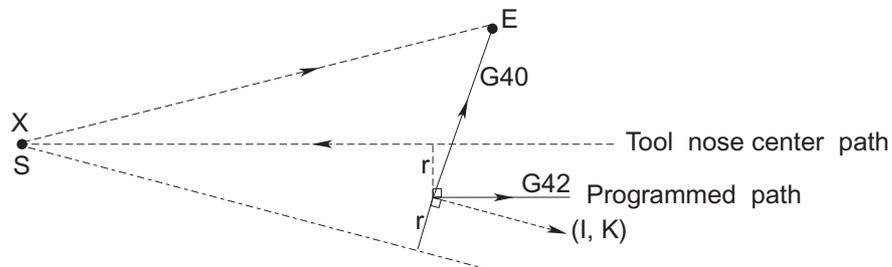
**9) When a block includes G40 and I\_\_\_ K\_\_\_;**

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)**.

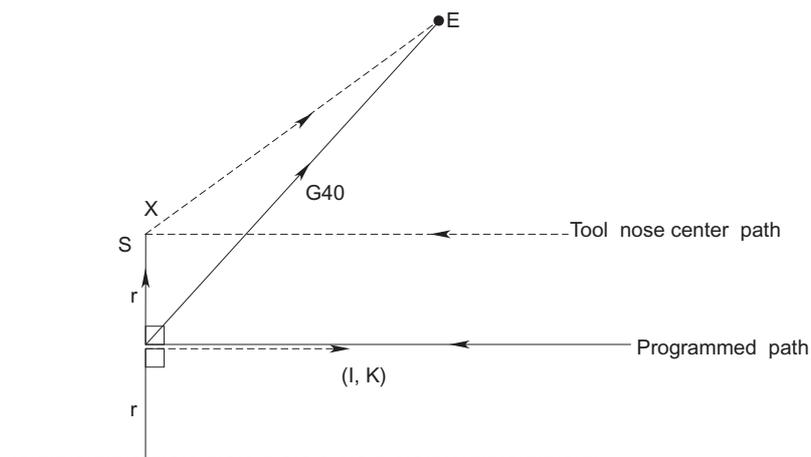


```
N1 (G42 mode)
N2 G40X a Z b I___ K___ ;
```

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.



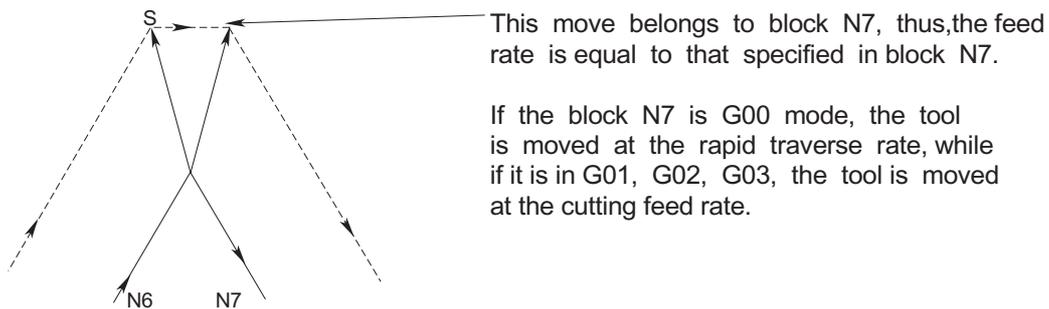
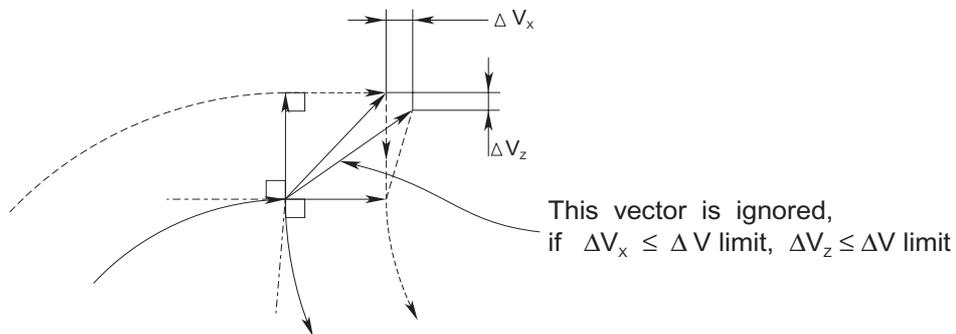
### 10) Corner movement

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

$$\text{If } V_x \leq DV_{\text{limit}} \text{ and } V_z \leq DV_{\text{limit}},$$

the latter vector is ignored. The value of  $\Delta V_{\text{limit}}$  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.



### 11) Interference check

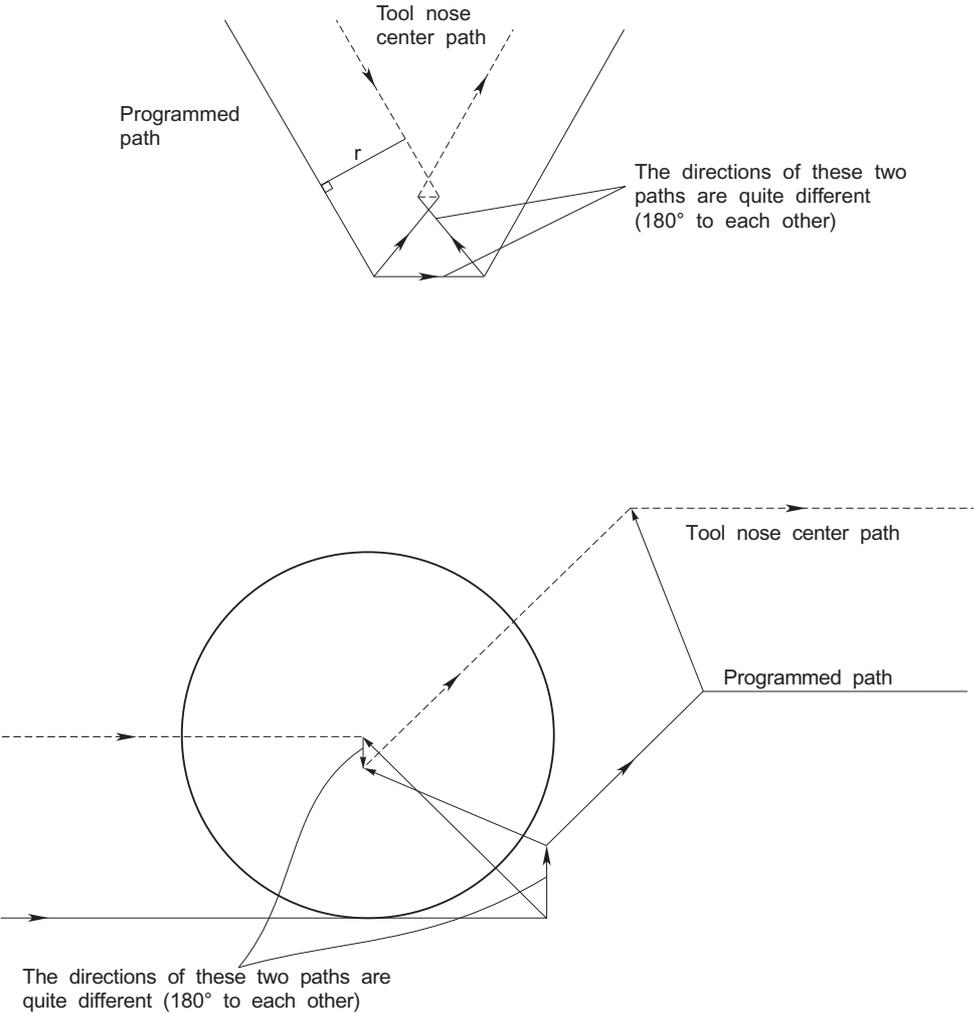
Tool overcutting is called “**interference** “. The interference check function checks for tool overcutting in advance. The interference check is performed even if overcutting does not occur.

#### a) Reference conditions for interference

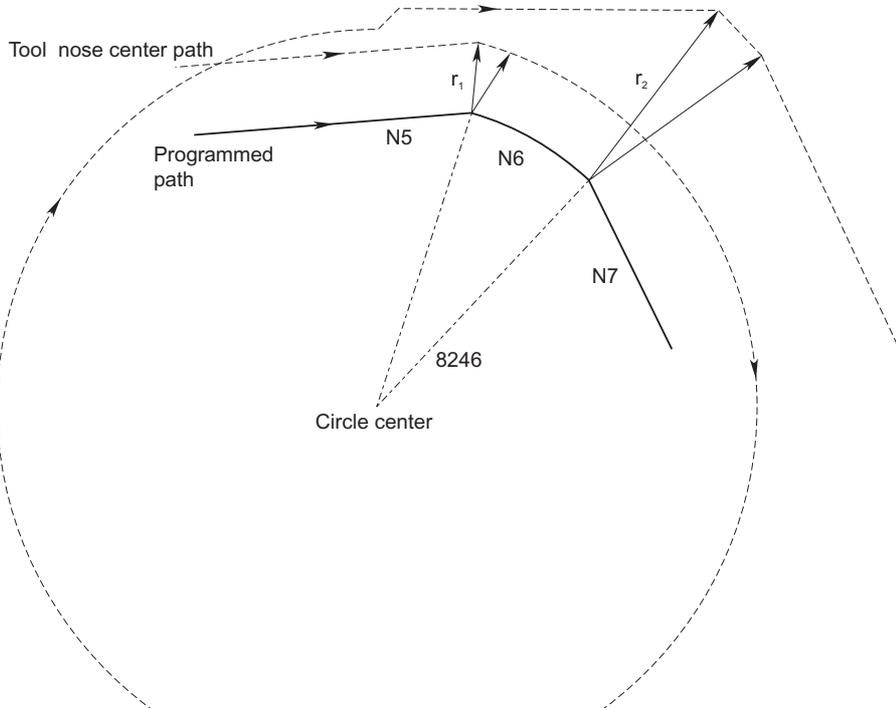
1) The direction of the tool nose center path in tool nose radius compensation is different from that of the programmed path.

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):*



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**

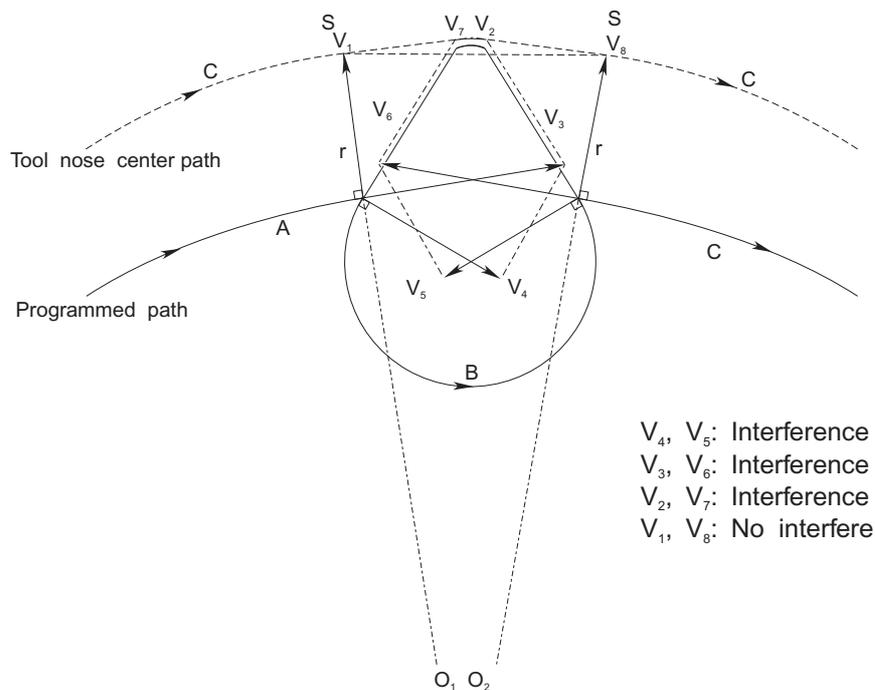
**1) Removal of the vector causing the interference**

When tool nose radius compensation is performed for blocks **A**, **B** and **C** -  $V_1, V_2, V_3$  and  $V_4$  and  $V_5, V_6, V_7$  and  $V_8$  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  $V_4$  and  $V_5$ . Interference  $V_4$  and  $V_5$  are ignored.  
 Interference check between vectors  $V_3$  and  $V_6$ . Interference  $V_3$  and  $V_6$  are ignored.  
 Interference check between vectors  $V_2$  and  $V_7$ . Interference  $V_2$  and  $V_7$  are ignored.  
 Interference check between vectors  $V_1$  and  $V_8$ . Interference can not be ignored.

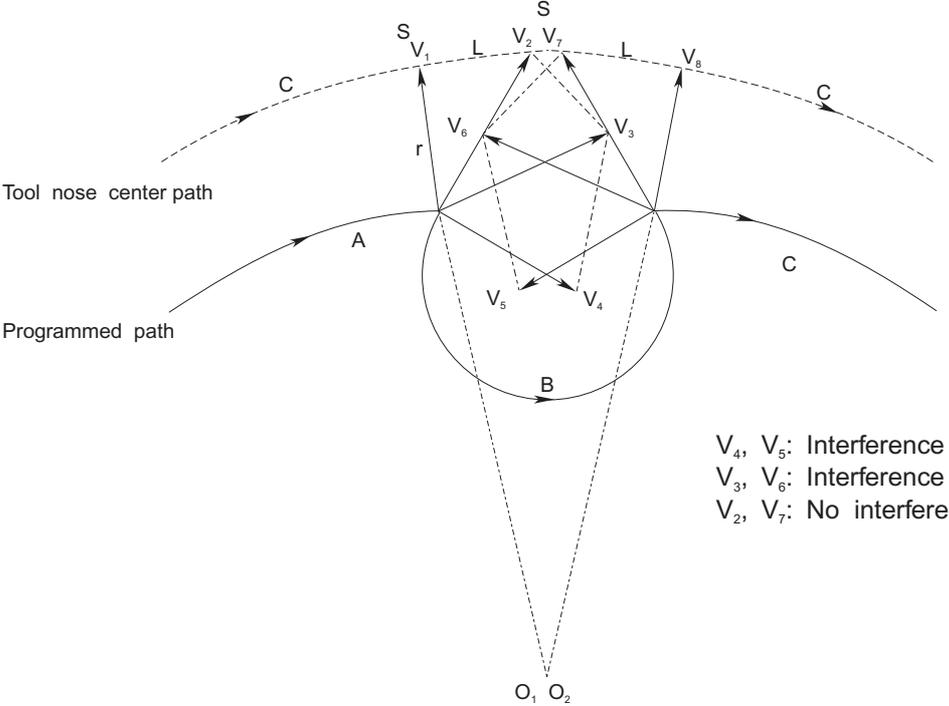
If while checking a vector with no interference is detected, subsequent vectors are not checked. If block **B** is a circular movement, a linear movement is produced if the vectors are interfered.

**Example 1)** The tool moves linearly from  $V_1$  to  $V_8$

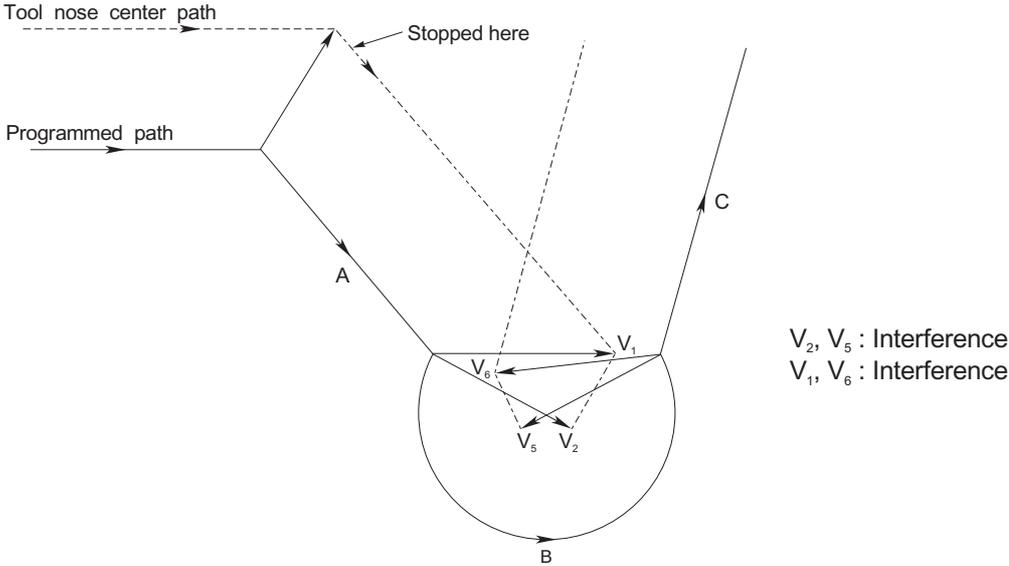


Example 2) The tool moves linearly as follows;

tool path:  $V_1 \rightarrow V_2 \rightarrow V_7 \rightarrow V_8$



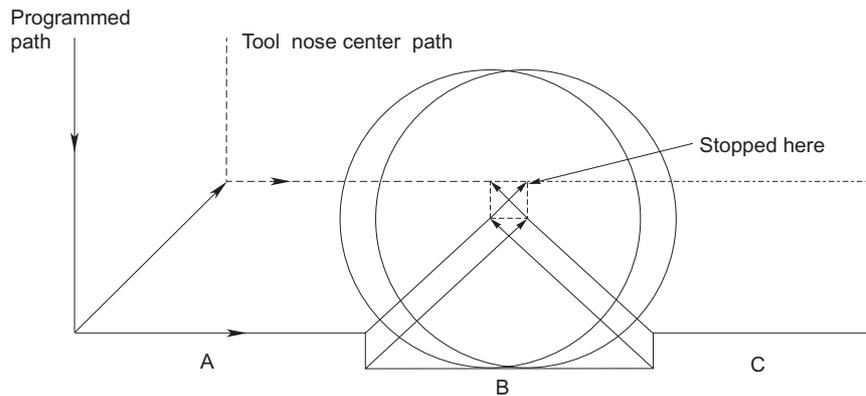
2) if the interference occurs after correction 1), the tool stops and alarm (PS41) is indicated



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

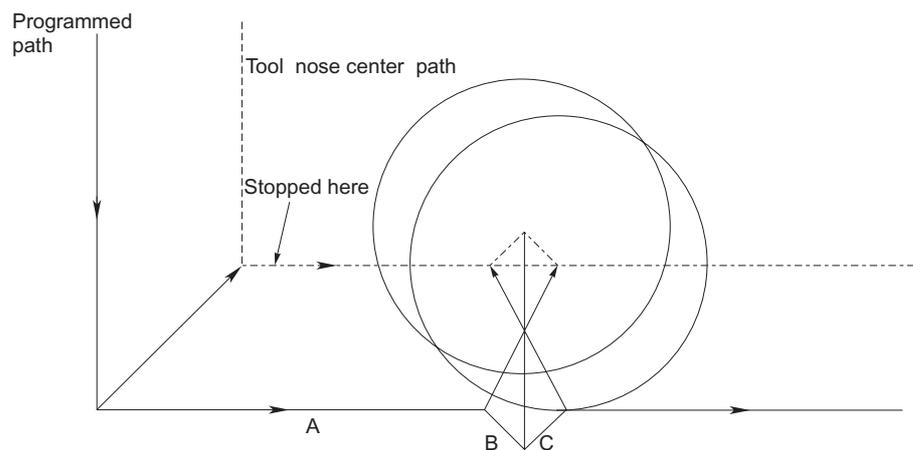
There are many examples, for instance the following:

**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.

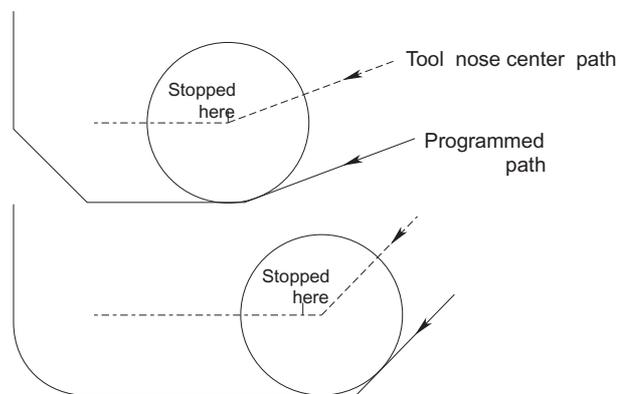
**2) A groove depth, smaller than the tool nose radius**



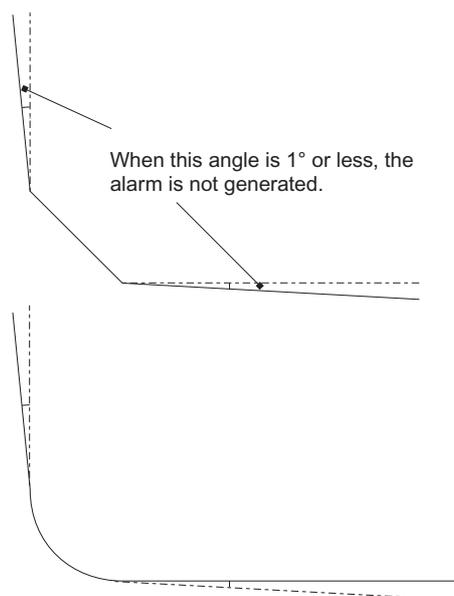
## 12) Correction in chamfering and corner arcs

**a)** In chamfering or corner arcs, tool nose radius compensation can be performed only 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.

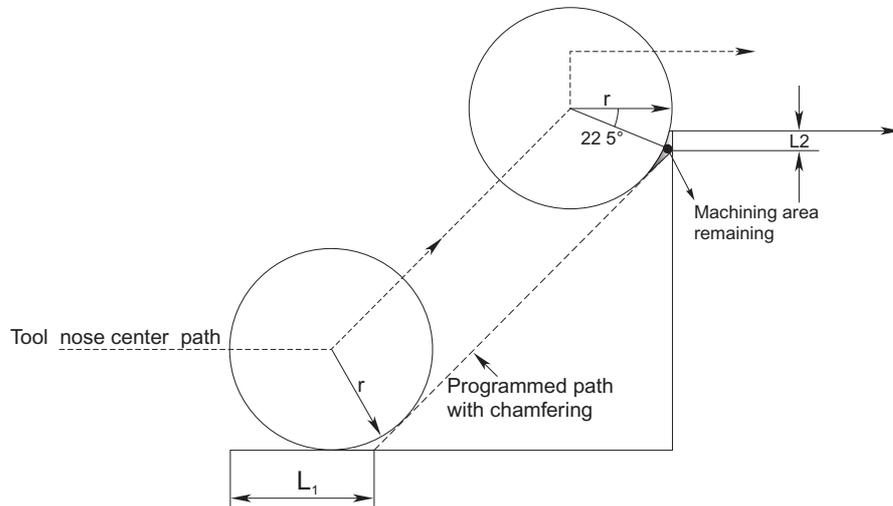


**c)** The valid inclination angle of the programmed path in the blocks before and after the corner is one degree or less so that the alarm does not occur.

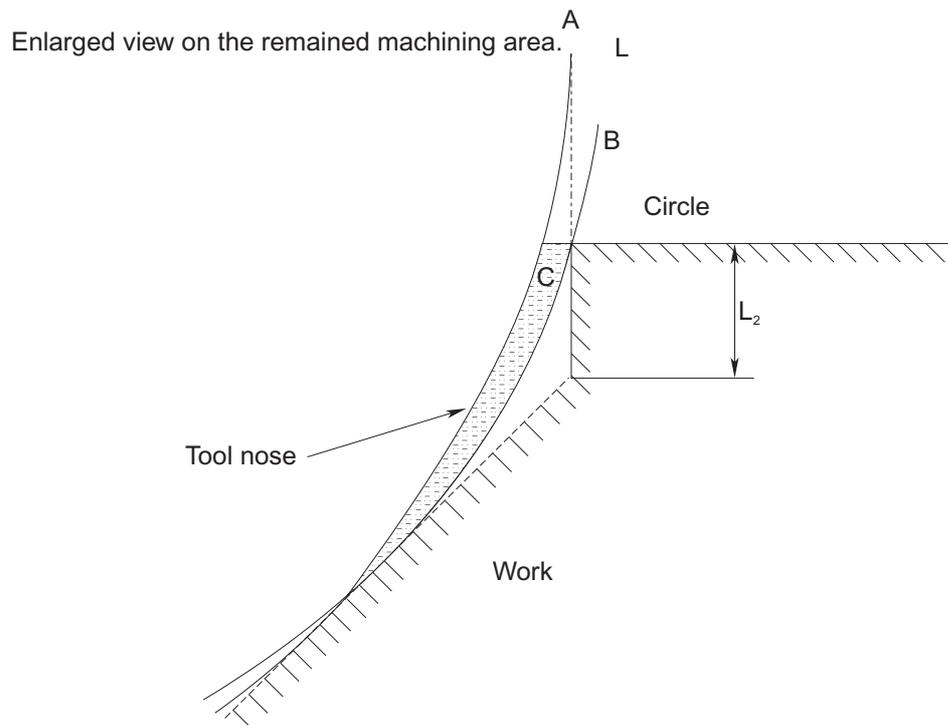


**d) When machining area remains**

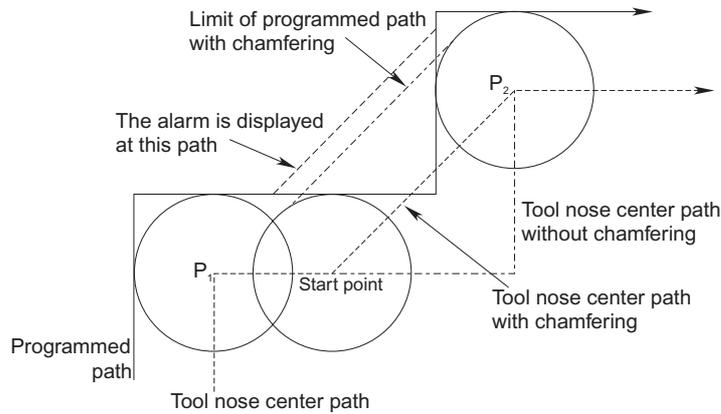
**1) The following example shows a machining area which can not be cut**



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 :**

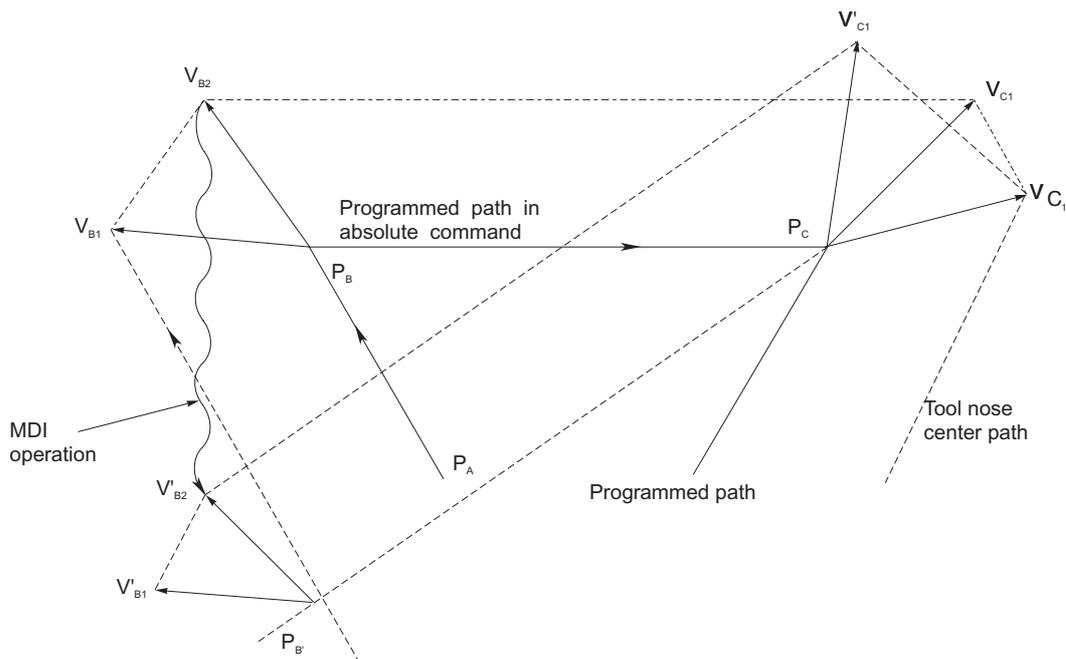


In outer chamfering with an offset, a limit is imposed on the programmed path. Path during chamfering coincides with the intersection points **P1** or **P2** without chamfering respectively, outer chamfering is limited. If the chamfering value is more than the limit value as specified, alarm **PS51**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

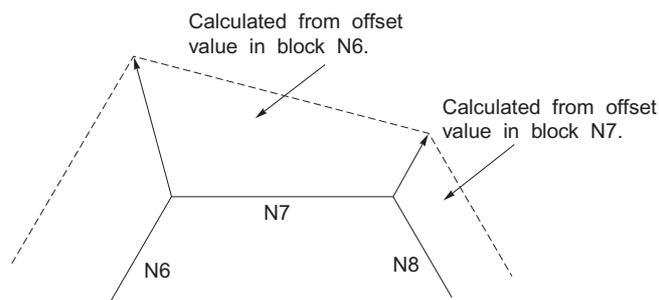


When points  $P_A$ ,  $P_B$  and  $P_C$  are programmed in absolute command, the tool is stopped by the SINGLE BLOCK function after executing the block from  $P_A$  to  $P_B$  and the tool is moved by MDI operation. Vectors  $V_{B1}$  and  $V_{B2}$  are translated to  $V'_{B1}$  and  $V'_{B2}$  and offset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

#### 14) General precautions for offset operations

##### a) changing the offset value

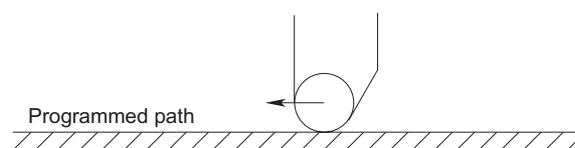
In general, the offset value is changed in cancel mode, or when changing tools. If the offset value is changed in offset mode, the vector at the end point of the block is calculated for the new offset value. The imaginary tool number and tool offset number are changed in the same way.



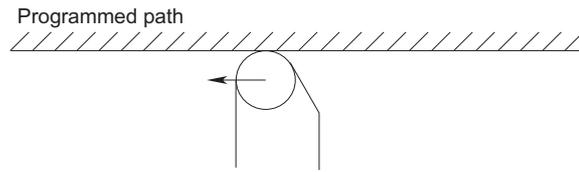
##### b) the polarity of the offset amount and tool nose center path

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 outer 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** - offset 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

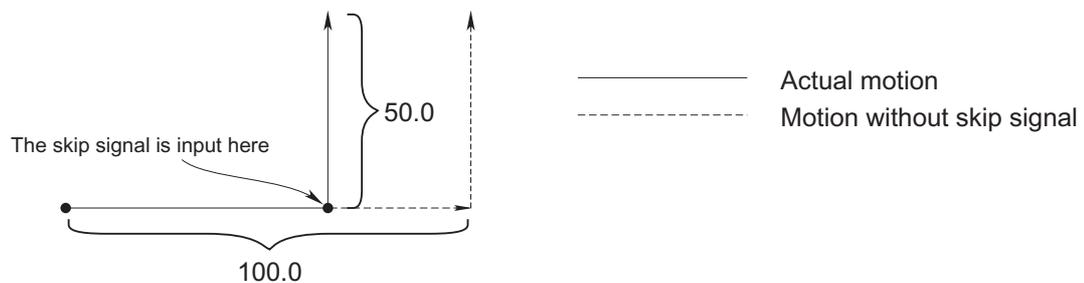
### 16.1 Skip function (G31)

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.

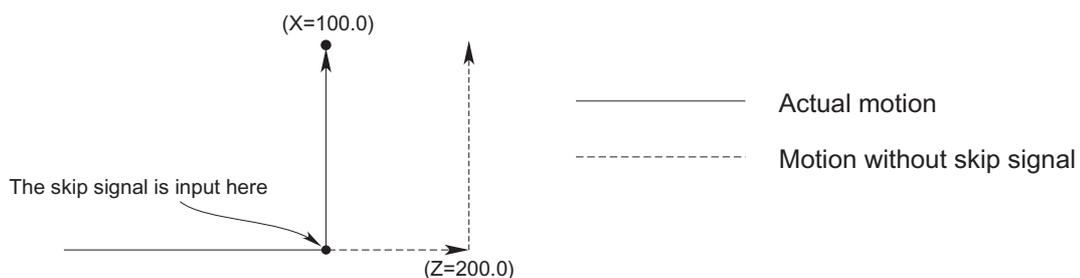
*Example:* **G98 G31 W100.0 F100;**  
**W50.0;**



#### 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.

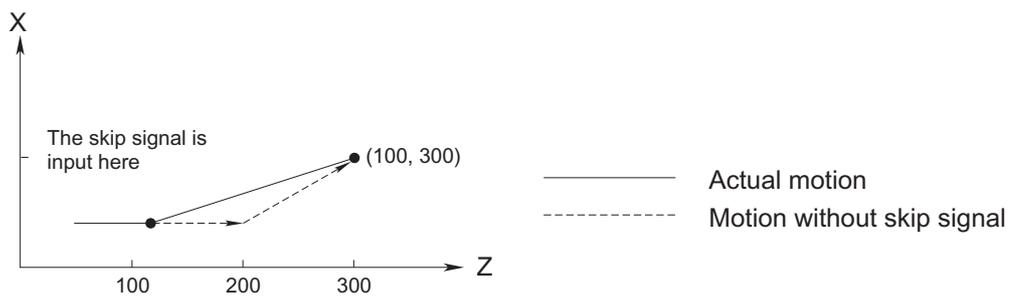
*Example:* **G31 Z200.0 F100.0;**  
**X100.0;**



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<sub>a</sub>;**

or

**G37 Zz<sub>a</sub>;**

In this case, the measured position should be **x<sub>a</sub>** or **z<sub>a</sub>** (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.

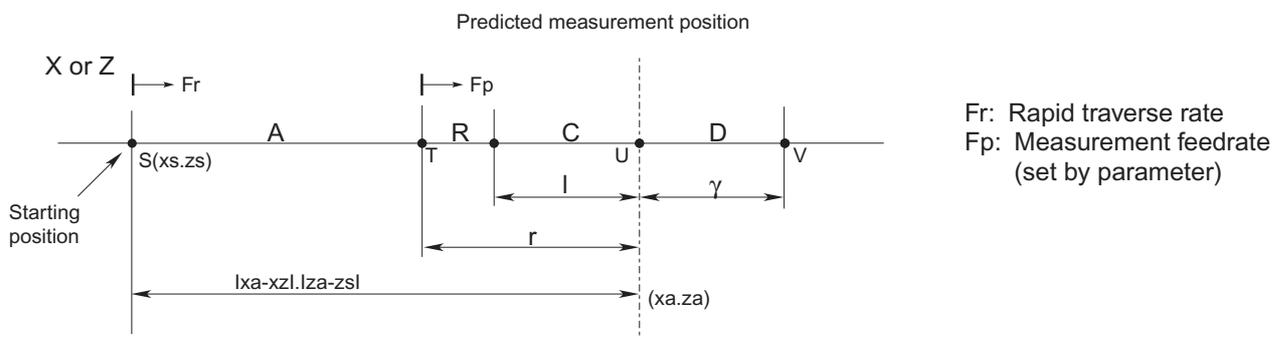
**c) Offset**

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 **x<sub>a</sub>** or **z<sub>a</sub>** specified in **G36 Xx<sub>a</sub>** or **G37 Zz<sub>a</sub>**.

Offset amount **X** = current offset amount **X** + (**α** - **x<sub>a</sub>**)

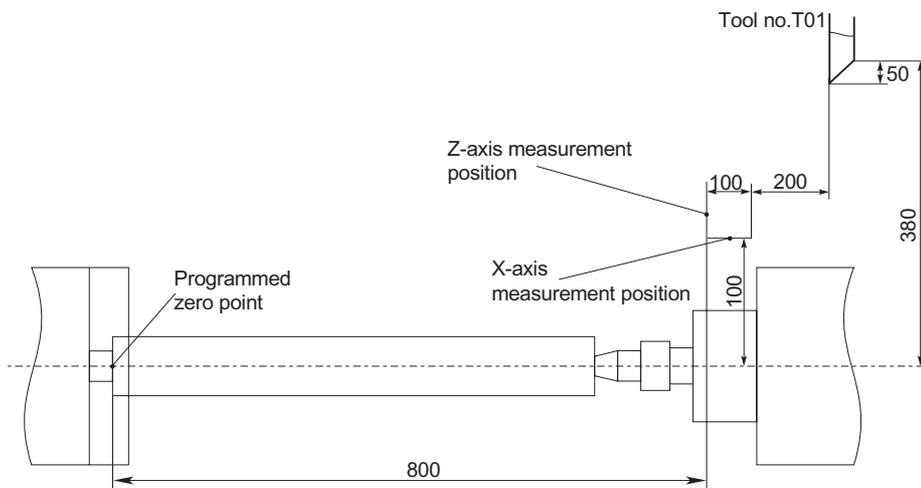
Offset amount **Z** = current offset amount **Z** + (**β** - **z<sub>a</sub>**)

**d) Feedrate and alarm**



The tool moves at the rapid traverse rate across area **A** from the starting position towards the measured position predicated by  $x_a$  or  $z_a$  in **G36** or **G37**. Then the tool stops at point **T**( $x_a - \gamma_x$  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:*



<b>Offset amount</b> (before measurement)	<b>Offset amount</b> (after measurement)
<b>X</b> 100000	98000
<b>Z</b> 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

<b>G36 X200000;</b>	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
<b>G00 X204000;</b>	Retracts a little along the <b>X</b> axis.
<b>G37 Z800000;</b>	Moves to the <b>Z</b> 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
<b>T0101;</b>	Further offset by the difference. The new amount becomes valid when the <b>T</b> code is specified again.

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

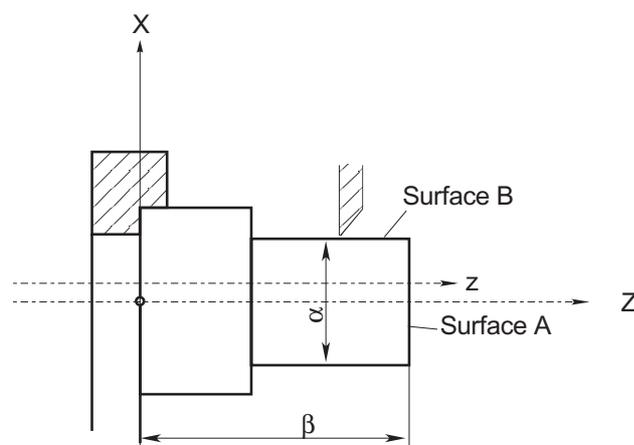
When a **T** code is specified in the same block as **G36** or **G37**, an alarm No.82 is generated.

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

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

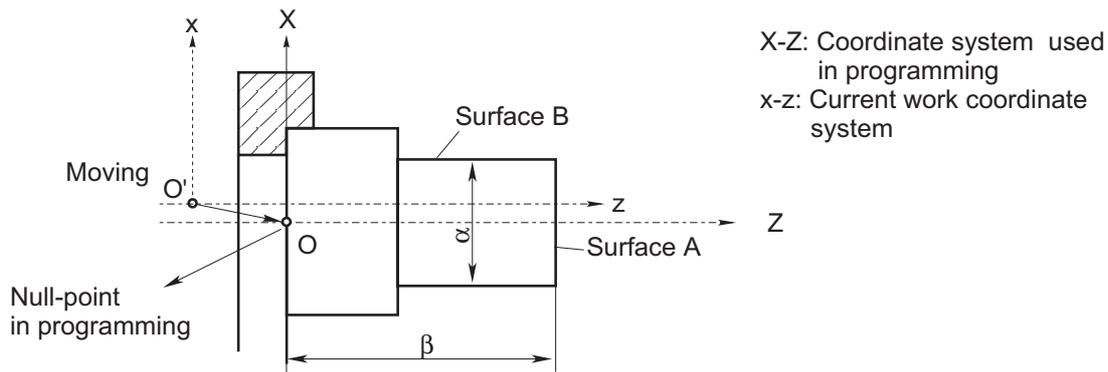
### 16.3 Direct setting of the tool compensation value

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  $\beta$  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  $\beta$  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  $\alpha$  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  $\alpha$  and press the [INSERT] key.

**16.4. Direct setting of the coordinate system shift value.**

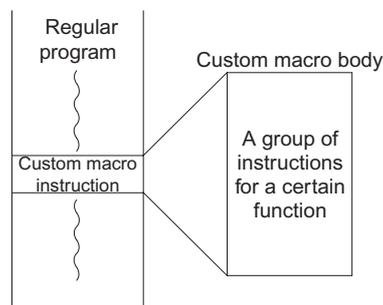


- a) In manual mode move the tool to surface **A**;
- b) Pull out the tool along the **X** axis without moving along the **Z** axis and stop the spindle;
- c) Measure the distance **B** between the standard null-point and the surface **A**;
- d) Select the screen "COORDINATE SYSTEM SHIFT"; Press the [**Z**] key, enter the measured distance  $\beta$  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  $\alpha$  between the standard null-point and the surface **B**;
- h) Select the screen "COORDINATE SYSTEM SHIFT"; Press the [**X**] key, enter the measured distance  $\alpha$  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:*

<b>F#103</b>	equivalent to F13 when #103=13
<b>Z- #110</b>	equivalent to Z-250 when #110=250
<b>G#130</b>	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

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

$$\#1032 = \sum_{i=0}^{15} \#(100+i) \times 2^i$$

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

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

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

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 offset 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	$\#i = \#j \times \#k$
G65	H05	Division	$\#i = \#j \div \#k$
G65	H11	Logical sum	$\#i = \#j . \text{OR} . \#k$
G65	H12	Logical multiplication	$\#i = \#j . \text{AND} . \#k$
G65	H13	Exclusive OR	$\#i = \#j . \text{XOR} . \#k$
G65	H21	Square root	$\#i = \sqrt{\#j}$
G65	H22	Absolute value	$\#i =   \#j  $
G65	H23	Remainder	$\#i = \#j - \text{trunc}(\#j / \#k) \times \#k$ Discard fractions less than 1
G65	H24	Conversion from BCD to binary	$\#i = \text{BIN}(\#j)$
G65	H25	Conversion from binary to BCD	$\#i = \text{BCD}(\#j)$
G65	H26	Combined multiplication/division	$\#i = (\#j \times \#k) \div (\#l)$
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	$\#i = \#j . \text{SIN} . (\#k)$
G65	H32	Cosine	$\#i = \#j . \text{COS} . (\#k)$
G65	H33	Tangent	$\#i = \#j . \text{TAN} . (\#k)$
G65	H34	Arctangent	$\#i = \text{ATAN} . (\#j/\#k)$
G65	H80	Unconditional divergence	GOTO n
G65	H81	Conditional divergence 1	IF $\#j = \#k$ , GOTO n
G65	H82	Conditional divergence 2	IF $\#j \neq \#k$ , 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 $\#j \geq \#k$ , GOTO n
G65	H86	Conditional divergence 6	IF $\#j \leq \#k$ , 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  $\#i = \#j - trunc (\#j / \#k) \times \#k$**

**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#i Q#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 #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  $\#i=\#j \times \text{COS}(\#k)$**

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

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

**(19) Tangent  $\#i=\#j \times \text{TAN}(\#k)$**

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

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

**(20) Arctangent  $\#i=\#j \times \text{ARCTAN}(\#k)$**

**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

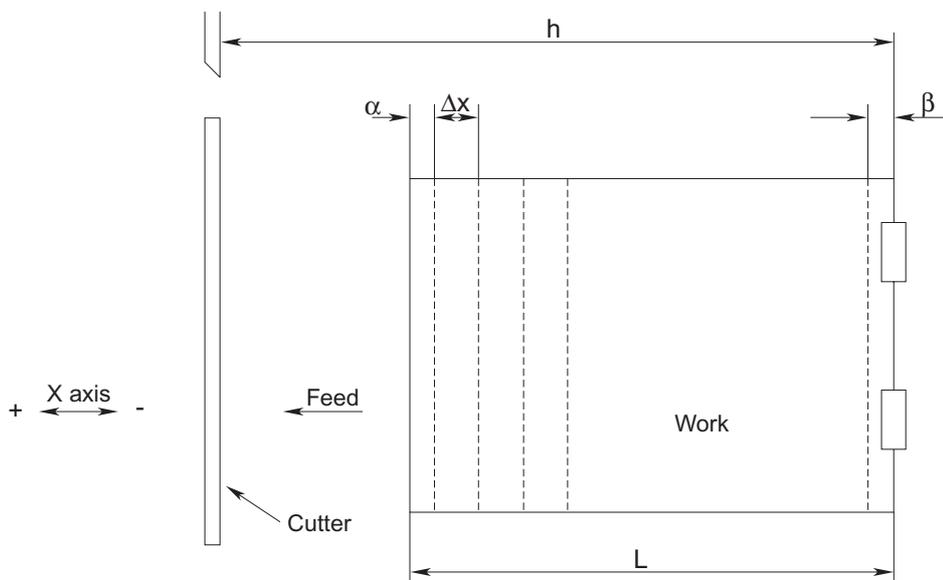
**17.4 Cautions on Custom Macro**

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 Application of Custom Macro**

**17.5.1 Shearing machine**



- #500: workpiece width ( $L$ )
- #501: first stock removal ( $\alpha$ )
- #502: cutting width ( $\Delta x$ )
- #503: workpiece gripping allowance ( $\beta$ )
- #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;
```

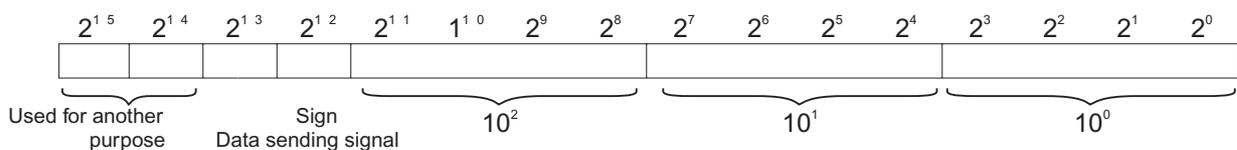
**Main program:**

```
O0009
G92 X0;
M98 P9110;
XO;
M02;
```

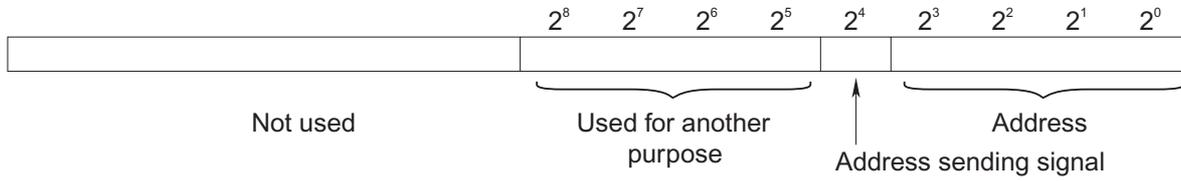
**17.5.2 Interface signal**

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;