

**Description of the Functional Characteristics of the Software**  
**«CNC Milling Machine Simulator»**

<https://sunspire.site/>

## **1. General Description of the Software Product**

The CNC milling machine simulator is a multimedia application designed to provide basic introduction to novice mechanical engineering specialists in the principles of programming parts milling operations using standard (ISO) G-code.

The main task of the application is syntactic analysis (parsing) of control program code in order to construct a graphical model of cutting tool trajectories in three-dimensional space.

Main functions of the application:

- editing the code of control programs;
- operations with control program files;
- setting the geometric parameters of the cutting tool;
- continuous/step-by-step execution of control program blocks;
- three-dimensional visualization of tool movements in the working space of the machine;
- simplified visualization of the machined surface of the part;
- calculation of processing modes;
- quick reference guide to using G-code.

The main limitations of the application: low accuracy of modeling the cutting surface, the inability to use polygonal geometry as a workpiece, a simplified model of machine tooling elements.

Type of target computing device and supported platform: IBM-compatible personal computer running Microsoft Windows, Apple Macintosh personal computer running MacOS, mobile devices based on Android and iOS operating systems.

The graphics component of the software uses the OpenGL 2.0 component base. The graphical user interface of the program is implemented in Russian, English and Chinese.

Multi-platform support allows you to use the software product on various computing devices, including interactive whiteboards, smartphones, tablets and desktop computers, which, in turn, increases the flexibility and mobility of the educational process, corresponding to the modern level of informatization of education.

Minimum system requirements:

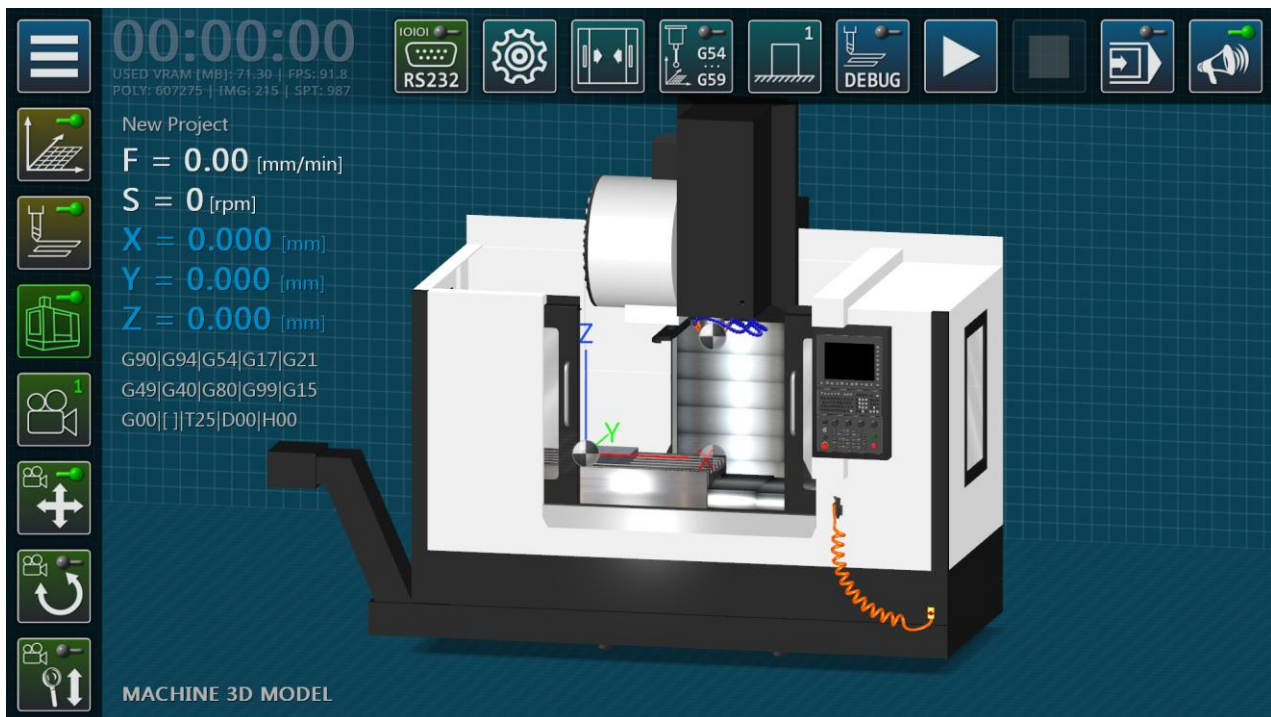
- central processor: Intel/AMD, at least 2 GHz;
- RAM: at least 1 GB;
- video memory: at least 512 MB;
- screen resolution: no less than 1024x768x32;
- OpenGL 2.0 (as well as Vulkan on mobile platforms);
- DirectX version 9.0.c (for Microsoft Windows OS);
- standard keyboard and computer mouse with scroll wheel (touch screen for mobile platforms);
- means of sound reproduction (speakers or headphones).

The software has a special version that supports hardware interaction with the NEWKer CNC Series 16iM physical CNC controller. This version of the program is supplied as part of a hardware and software training complex developed by Yantai ChangHao Intelligent Technology Co., Ltd (China, Yantai). The interaction of the software and hardware components of the training complex is carried out in accordance with the ModBus application layer network protocol. The CNC controller is directly connected to a personal computer via the RS-485 interface.

## 2. Graphical User Interface Description

The main screen of the application is an interactive three-dimensional scene depicting a geometric model of a three-axis milling machining center placed in a conditional surrounding space. The model is monitored using a virtual camera, the angle of which can be changed during operation.









Along with three-dimensional graphic elements, two-dimensional control elements are displayed on the main screen - buttons, switches and basic text information about the simulated processes (Fig. 1).



**Figure 1** – Simulator Main Screen









On the left side of the main screen there is a vertical panel of the main functional buttons (Table 1).

**Table 1 – Main Function Buttons of the Main Screen**




Button Image	Function Name	Function Description
	Main Menu	Displaying the main menu
	Coordinate System	Show/hide the coordinate system and grid of the machine workspace
	Tool Paths	Show/hide the graphical model of cutting tool paths
	3D Model of the Machine	Show/hide the complex graphical model of milling machine elements
	Current Camera	Switching virtual camera mode
	Camera Offset Mode	Virtual camera planar offset mode (for touch screen devices)
	Camera Rotation Mode	Virtual camera rotation mode (for touch screen devices)
	Camera Zoom Mode	Virtual camera zoom in/out mode (for touch screen devices)

At the top of the main screen there is a horizontal panel of buttons for controlling the simulation process and settings (Table 2).

**Table 2** – Buttons for Controlling the Simulation Process and Settings

Button Image	Function Name	Function Description
	Simulator Settings	Display the application settings panel for selecting localization and changing the simulation speed
	Automatic Doors	Changing the state of automatic doors (opening/closing)
	Workpiece Binding Mode	Mode for determining the position of workpiece working coordinate systems using a measuring sensor
	Tool Binding Mode	Mode for determining tool offset and diameter using a measuring sensor
	Selection of Technological Equipment	Choosing the method of fixing the workpiece on the coordinate table
	Debug Mode	Quickly debug the control program and build a complete model of tool paths
	Program Start	Launching the machine control program in continuous block execution mode
	Program Pause	Pausing the machine control program

**Table 2 – Continuation**

Button Image	Function Name	Function Description
	Program Stop	Stopping the execution of the machine control program
	Step-by-Step Mode	Turning on/off the step-by-step execution mode of the machine control program
	Sound Control	Turning on/off the sound of the simulation process

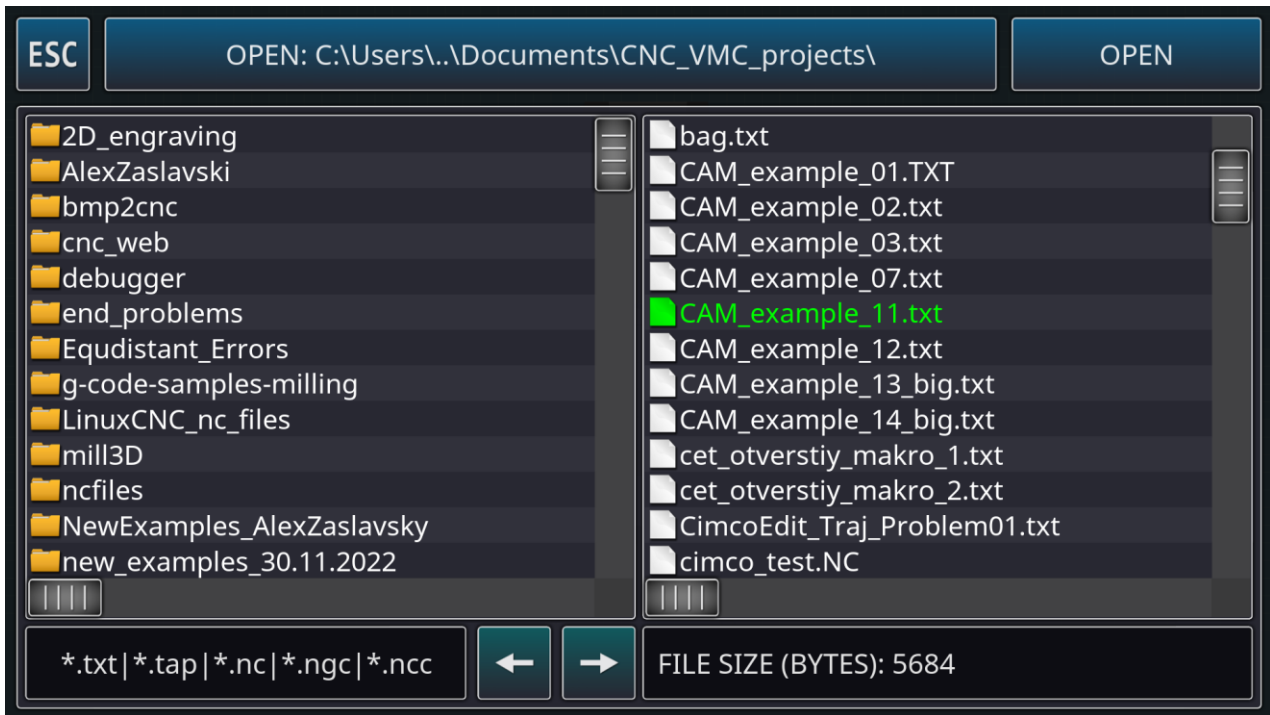
When you click on the «Main Menu» button, a radial menu for selecting auxiliary application dialog screens is displayed on the screen (Fig. 2).



**Figure 2 – Simulator Main Menu**

**Button 1 – «Create Project»** – is intended to reset the current simulation parameters. This function is used if, after working with an open file, it is necessary to start a new control program project. When you click on the «Create Project» button, a modal dialog box appears on the screen to confirm that the current simulation parameters have been reset.

**Button 2 – «Open Project»** – is intended for selecting and opening a machine control program file. When you click on the «Open Project» button, the file opening screen is displayed (Fig. 3), in the main area of which a list of directories and a list of control program files written to the projects' working directory are displayed.



**Figure 3 – Open File Screen**

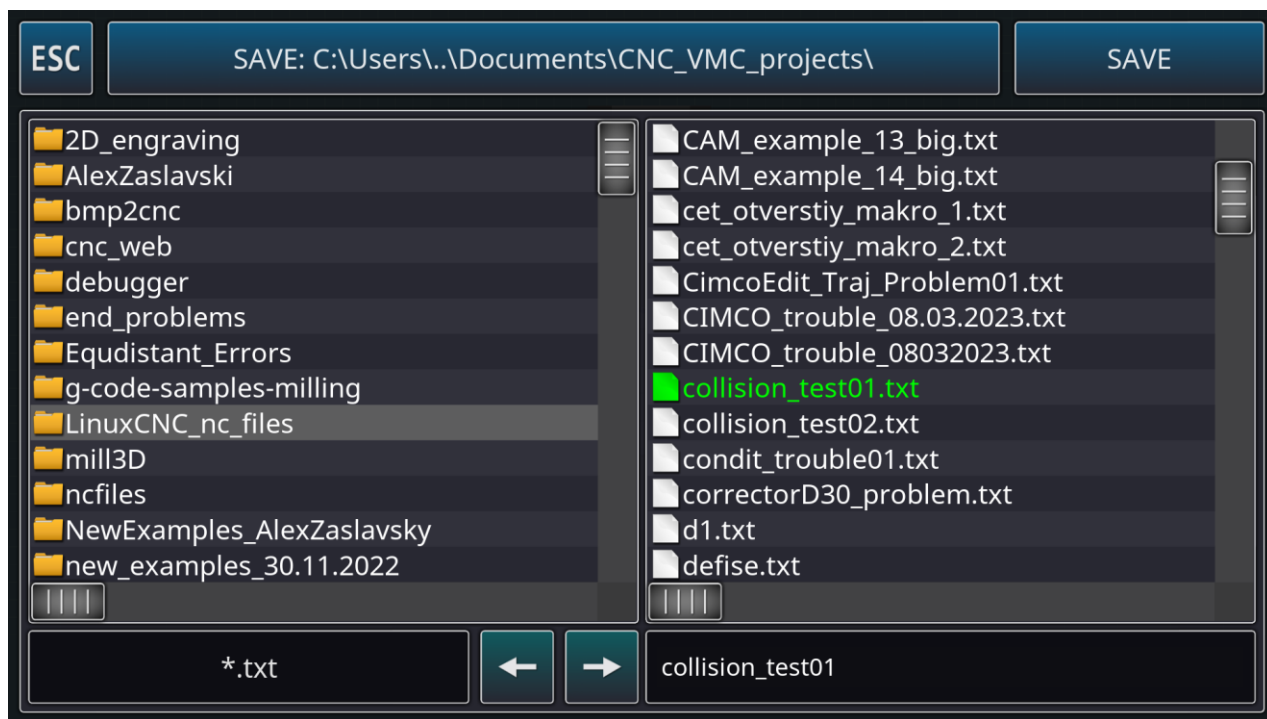
At the bottom of the file opening screen there is a file type filter that allows you to display files of a given format in the list. The simulator supports 5 formats of text files of control programs: \*.txt, \*.tap, \*.nc, \*.ngc, \*.ncc. The size of the selected file in bytes is displayed at the bottom right of the screen. The selected file in the list



is highlighted in green. When a file is selected, the «Open» button at the top right of the screen becomes active.

In the mobile version of the application, an additional «Send» button is displayed at the top of the file opening screen, allowing you to send the selected file using an external application (for example, an email client).

**Button 3 – «Save Project»** – is intended for selecting the local path and name of the machine control program file to be saved. When you click on the “Save Project” button, the file saving screen is displayed (Fig. 4), the principle of interaction with which is similar to the file opening screen. The name of the saved file is entered into a special text field located in the lower right part of the screen, without specifying the extension. The file extension is selected using the switch at the bottom left of the screen.



**Figure 4 – Save File Screen**

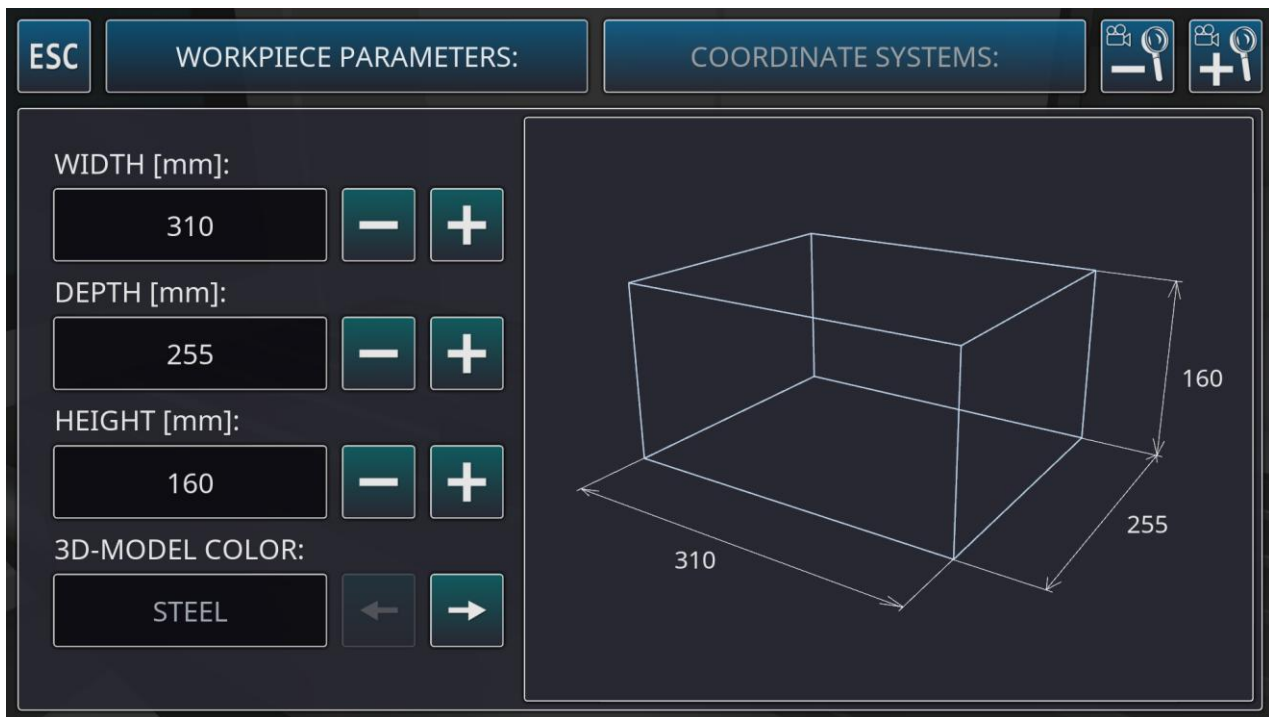
Control program files and nested directories of standard control program examples are located in the «C:\Users\...\Documents\CNC\_VMC\_projects» in

Microsoft Windows. In the Android system, the default path of the control program files is «/storage/emulated/0/CNC\_VMC\_projects» or «/storage/emulated/0/Android/data/com.virtlab.cncvmcsim/files/CNC\_VMC\_projects» (for later versions of Android OS). On iOS and macOS systems, control program files are stored in the application's working directory.

**Button 4 – «License Information»** – is designed to display a screen that displays basic licensing information about the licensor and licensee of the software product.

**Button 5 – «Exit the Simulator»** – is designed to display the simulator shutdown dialog screen.

**Button 6 – «Workpiece Parameters»** – is designed to display the screen for setting the parameters of the workpiece being processed (Fig. 5).

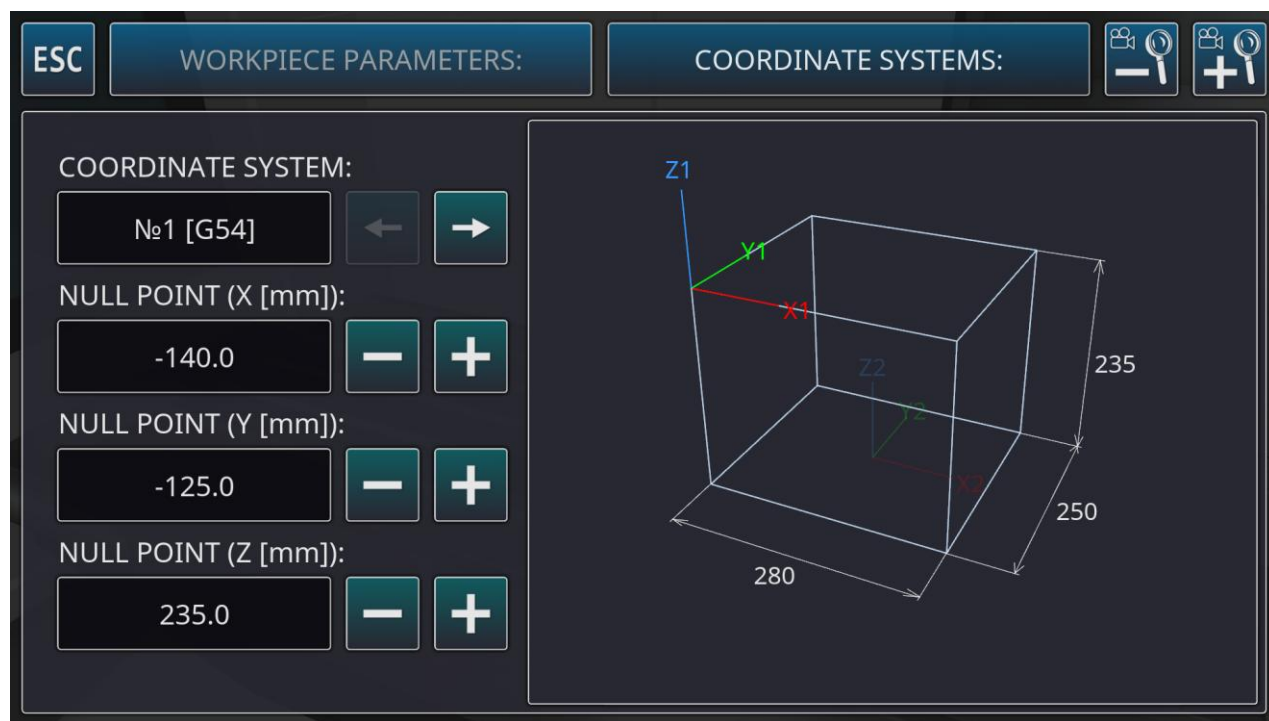


**Figure 5 – Workpiece Setting Screen**

On the left side of the workpiece parameters screen there is a block of numeric and string fields with buttons to increase/decrease values. Using these

parameters, you can set the geometric dimensions of the workpiece being processed and select the color of the simulated surface (based on the name of the material). On the right side of the screen, a frame geometric model of a rectangular workpiece and its main dimensions are displayed. The geometric model of the workpiece can be rotated using the mouse or touching the touch screen. Also in the upper right part of the screen there are buttons for scaling the model.

In the upper central part of the screen there are header buttons designed to switch the workpiece setup mode. By default, when opening the workpiece parameters dialog screen, the dimensions of the workpiece being processed are adjusted. The «Coordinate Systems» button provides the ability to configure the relative position of the workpiece coordinate systems (Fig. 6).



**Figure 6** – Coordinate System Setup Screen

Six workpiece coordinate systems are available for pre-setting. Further switching between working coordinate systems during the simulation process is carried out programmatically by using special control codes G54-G59. In this mode,

the coordinates presented in the left panel are not machine coordinates, but represent axis offsets relative to the center of the workpiece. A table of actual machine coordinates or zero offsets of the part is displayed on the screen when selecting a mechanical vice as a method of securing the part to the coordinate table, as well as in the mode of binding the workpiece using a sensor.

Information about the workpiece parameters is saved at the beginning of the control program file in the form of comments.

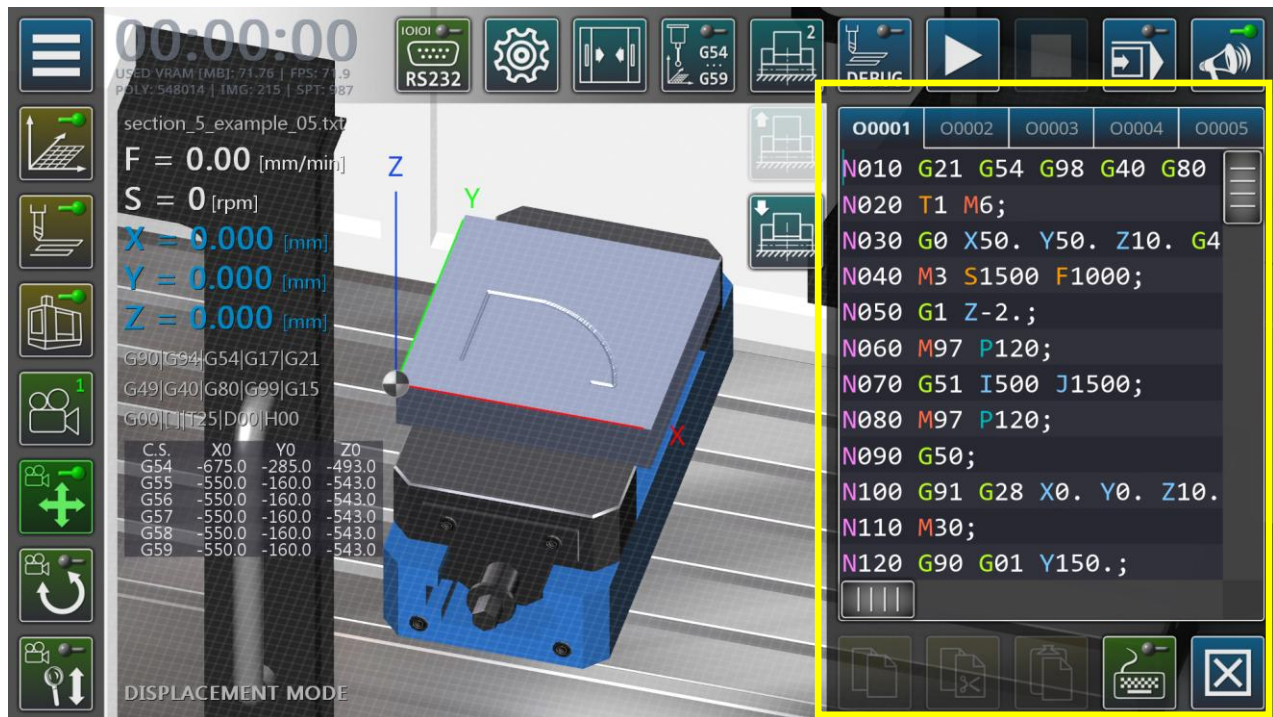
**Button 7 – «Tools Parameters»** – is designed to display the screen for setting the cutting tool parameters (Fig. 7).

**Figure 7 – Cutters Setting Screen**

On the left side of the tool parameters screen there is a block of numeric and string fields with buttons for increasing/decreasing values, as well as switching positions. Using these parameters, you can specify the type of cutting tool used in the simulation, the design, as well as the geometric dimensions of the working part of the cutting tool.

On the right side of the screen a simplified drawing of the selected tool and its technological equipment is displayed with the main dimensions displayed. The drawing is updated in real time as tool parameters change. At the top of the screen there are title buttons similar to the workpiece parameters screen. The «Table of Correctors» button displays on the screen a summary table of the geometric dimensions of all used tools with the corresponding numbers of correctors used by the G41/G42/G43/G44 software functions.

**Button 8 – «Control Program Editor»** – is designed to display a text editor for control program code on the main application screen (Fig. 8).



**Figure 8 – Control Program Editor in Standard (Reduced) Display Mode**

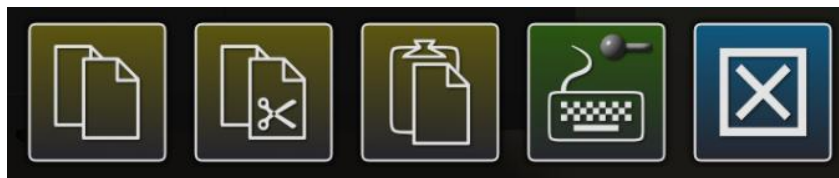
The main part of the text editor is represented by a list of control program lines available for editing from the physical keyboard. Editing the text of the machine control program is carried out only in stopped simulation mode. When the simulation process is running, the text editing mode is not available to the control



program. In this case, the text of the control program is scrolled automatically, and the currently executing program block is highlighted in green.

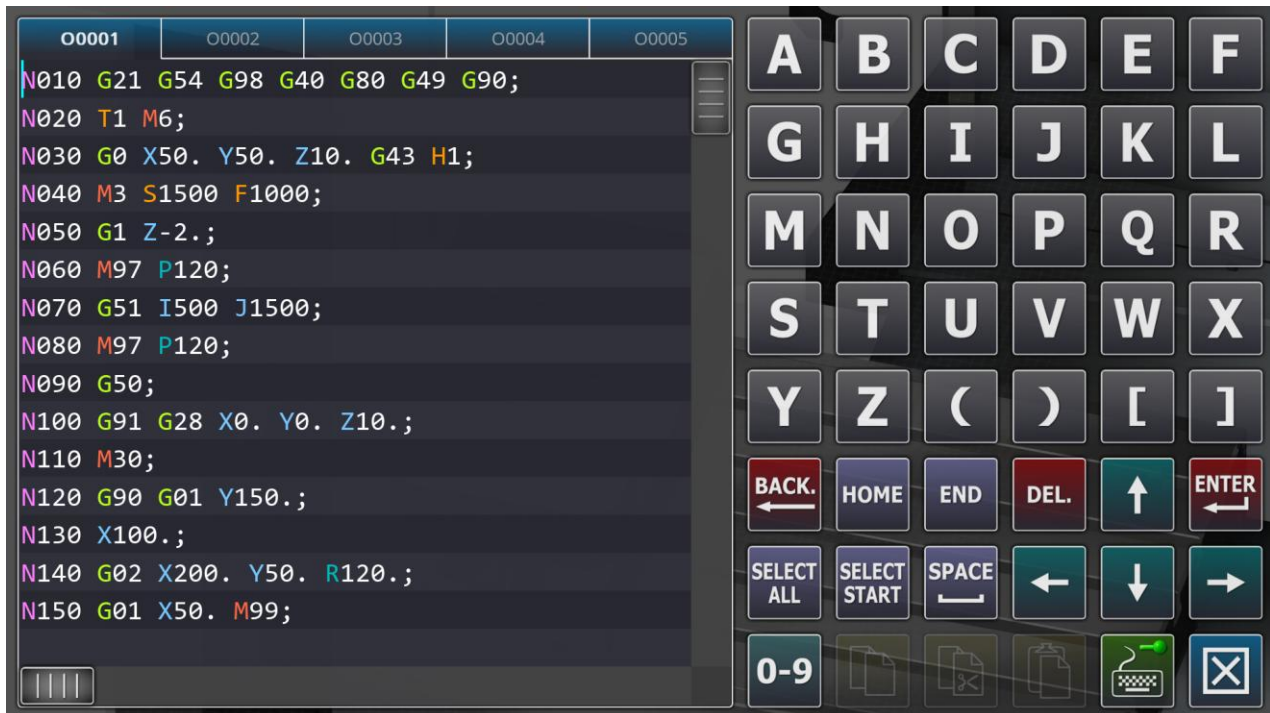
At the top of the text editor there are tabs designed to switch control programs of the current project (5 in total). Using additional tabs, you can include external subroutines in the control program, the numbering of which corresponds to the tab numbers.

At the bottom of the text editor there are the main functional buttons: «Copy», «Cut», «Paste», «Virtual Keyboard» and «Close Editor» (Fig. 9).



**Figure 9** – Basic Functional Buttons of a Text Editor

The «Virtual Keyboard» toggle button allows you to switch the editor to full-screen mode (Fig. 10).



**Figure 10** – Control Program Editor in Full-Screen Display Mode

In full-screen mode, a text editor appears on the left side of the screen, and a virtual alphanumeric keyboard appears on the right side of the screen, allowing you to edit text on touchscreen devices that do not have a physical keyboard (such as tablets or smartphones).

In editing mode, the text of the control program code is color-marked by groups of letter codes and addresses. Letter codes and addresses of one group (for example, technological parameters T/H/D/S/F or coordinates X/Y/Z/I/J/K) are indicated in one color. In the control program execution mode, color marking of codes and addresses will not be used.

**Button 9 – «Processing Mode Calculator»** – is designed to display the milling modes calculator screen (Fig. 11).

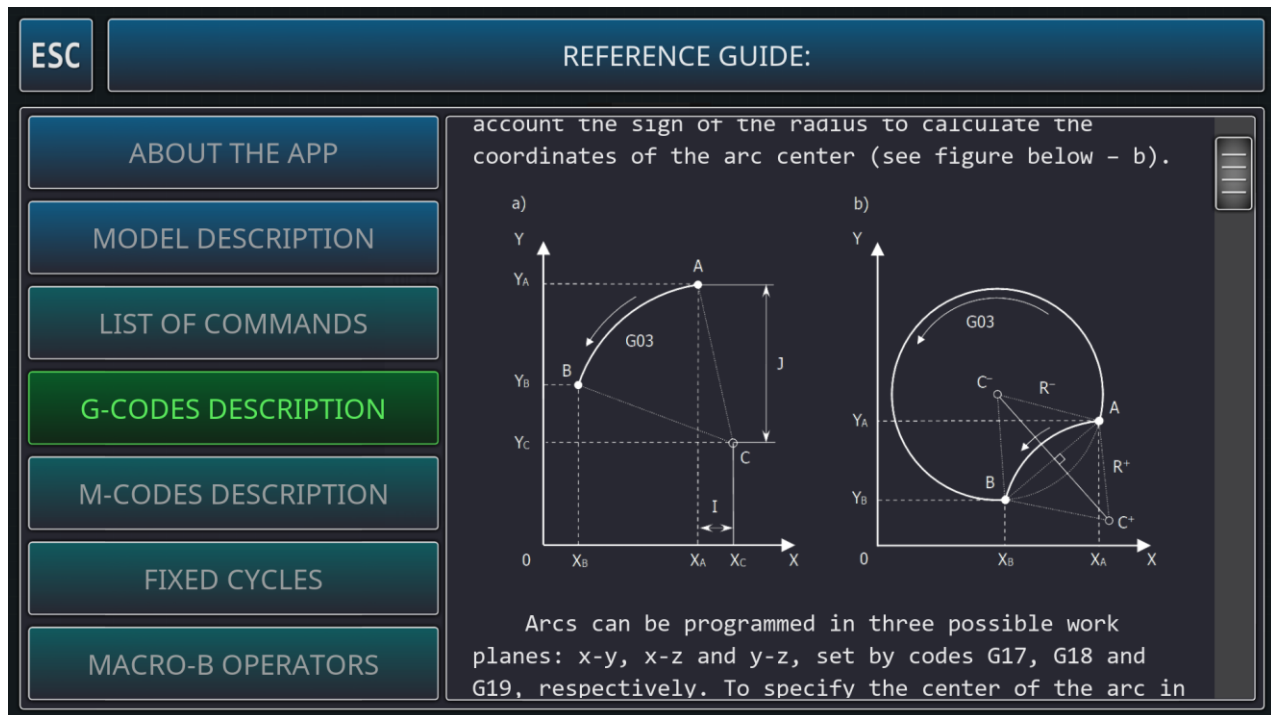
SPEED AND FEED CALCULATOR:	
CUTTING SPEED [mm/min]: 10	WORKPIECE MATERIAL: PLASTIC
CUTTER DIAMETER [mm]: 0.5	CUTTER DIAMETER INTERVAL [mm]: 0.5
NUMBER OF CUTTER TEETH: 1	RECOM. CUTTING SPEED [mm/min]: 300-400
FEED PER TOOTH [mm]: 0.005	RECOM. FEED PER TOOTH [mm]: 0.02
SPINDLE SPEED [rpm]: 6366	FEED RATE [mm/min]: 31.8

**Figure 11** – Processing Mode Calculator Screen

Using numeric and string fields located in the main part of the screen, input parameters of the milling process are set. The fields whose values are highlighted in blue are auxiliary and display reference values of processing parameters. The output

values of the processing mode are the spindle rotation speed and the cutting tool feed rate. These values are calculated automatically and displayed at the bottom of the screen.

**Button 10 – «Reference Guide»** – is designed to display the application's built-in help screen (Fig. 12). The Reference Guide provides basic information about the application and the CNC tools used. The reference information is structured into 7 sections: «About the Application», «Model Description», «List of Commands», «G-codes Description», «M-codes Description», «Fixed Cycles», «MacroB Operators».



**Figure 12 – Built-in Help Screen**

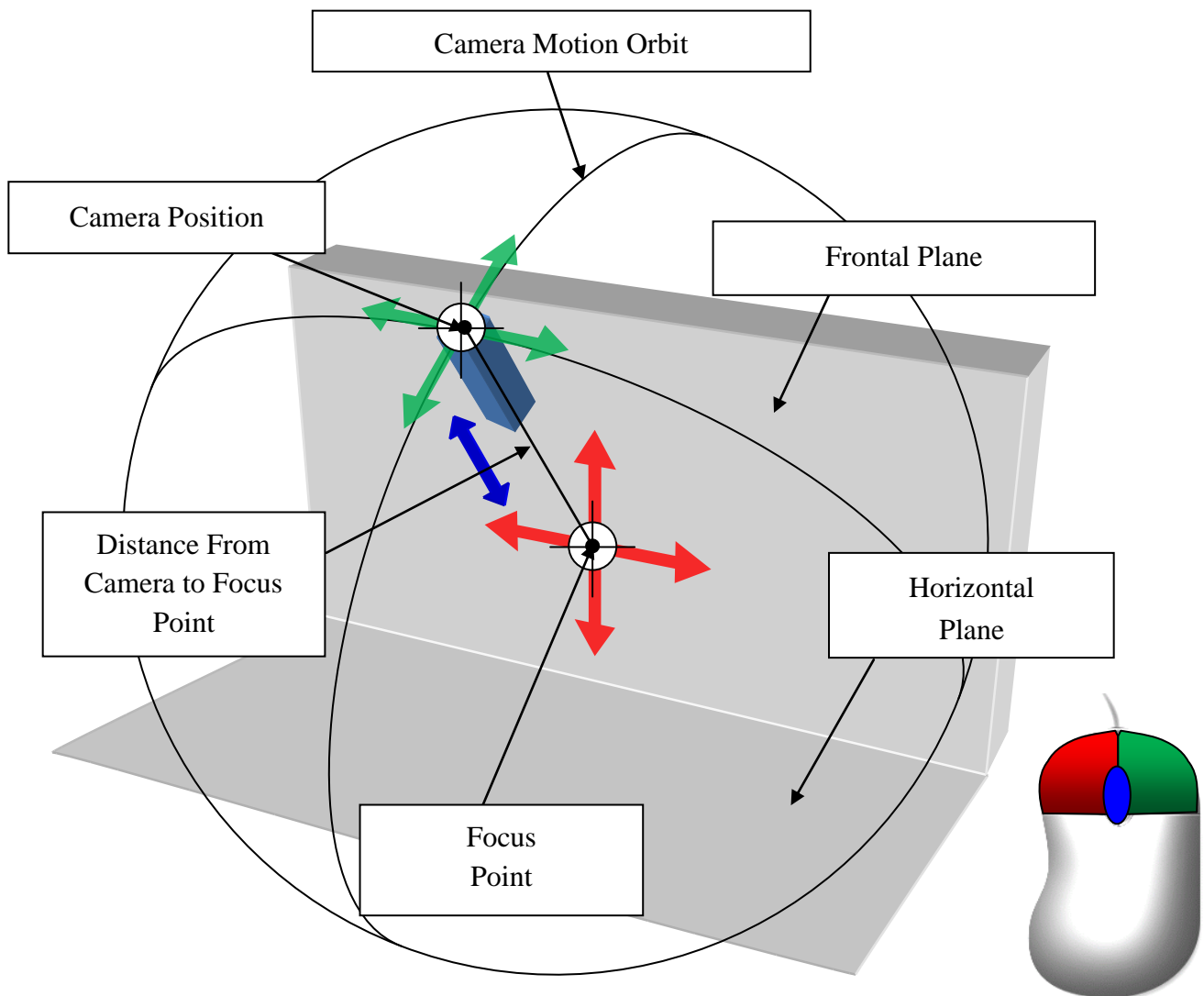
### 3. The Principle of Controlling a Virtual Camera Using a Mouse

Basic manipulations with the camera in mode №1 can be performed using a computer mouse. In this case, pressing and holding the left mouse button with concomitant movement of the mouse leads to movement of the camera focus point in the frontal plane of space. Pressing and holding the right mouse button while



moving the mouse causes the camera to rotate relative to the focus point. The rotation angles (azimuth and elevation) of the camera are limited by the dimensions of the model space. Changing the camera distance is done by rotating the scroll wheel in forward and reverse directions (Fig. 13).

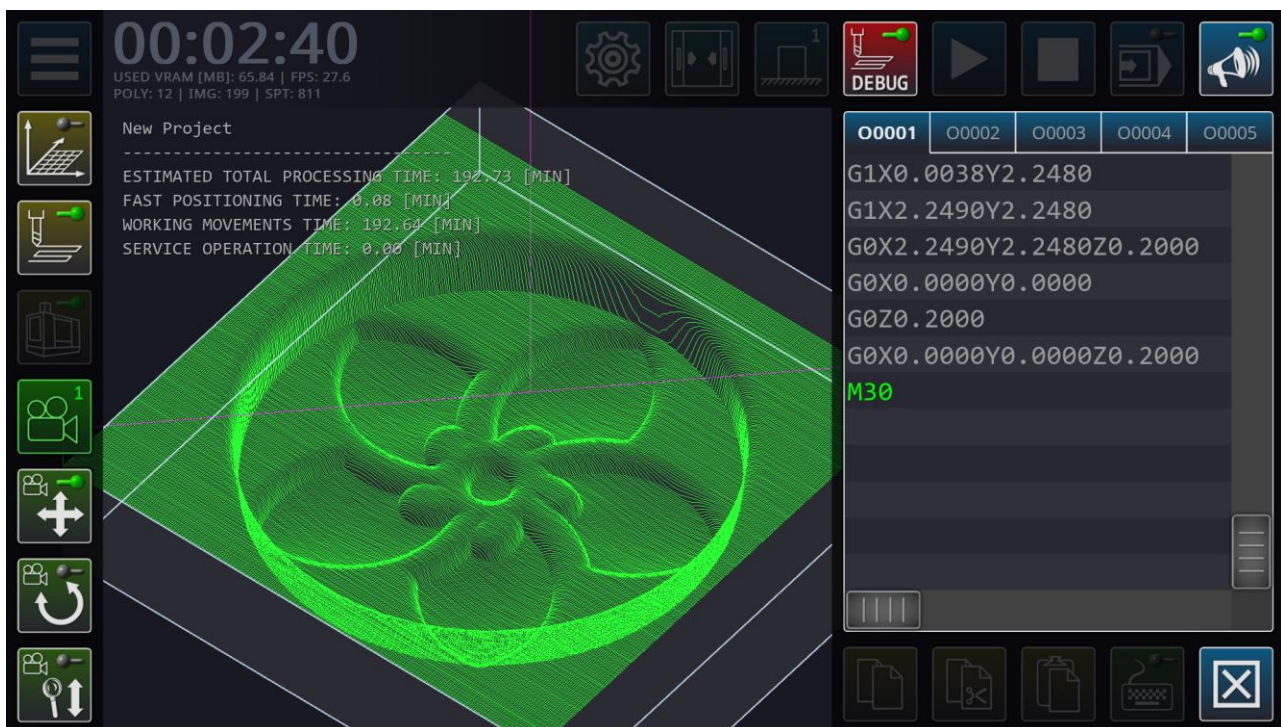
On devices with a touch screen, the camera is controlled using gestures (Touch Screen).



**Figure 13** – Scheme for Controlling a Virtual Camera Using a Computer Mouse

#### 4. Cutting Tool Trajectories Calculation Mode or Debug Mode

In debugging mode, syntactic analysis (parsing) of the control program is carried out quickly – without actually moving the cutting tool and switching operating modes of the machine. In this mode, the three-dimensional model of the machine, cutting tool and equipment is not displayed on the screen. In this case, the dimensions of the workpiece, the active coordinate system, designations of zero points and tool path lines are displayed on the screen (Fig. 14).

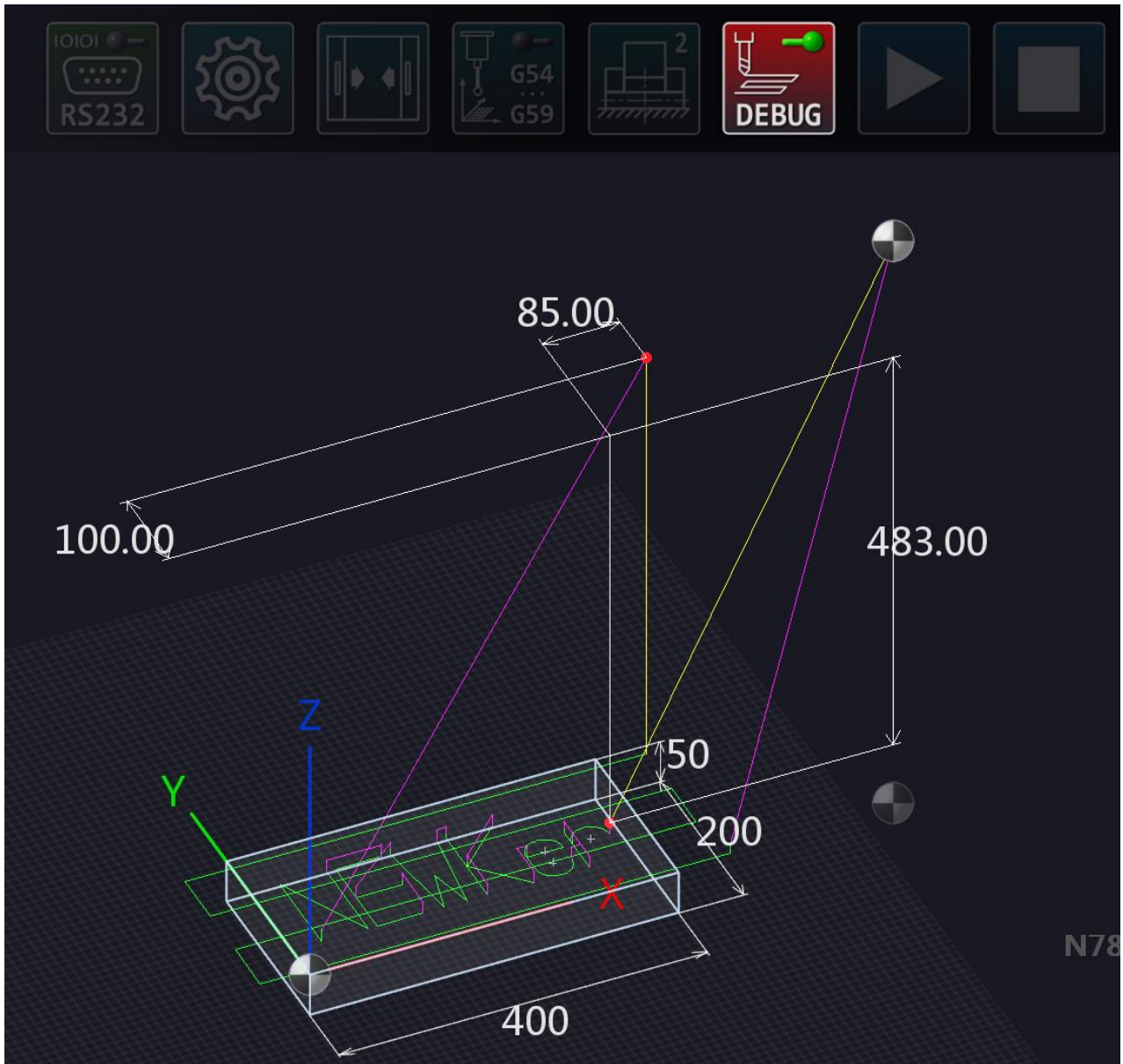


**Figure 14** – View of the Simulator Main Screen in the Debug Mode

The left side of the main screen displays the estimated time spent on performing main and auxiliary technological operations in minutes and seconds.

Once the trajectory model is built, individual sections of the trajectory can be measured using the measurement function. This function is only available if the device has a mouse. Touchscreen devices do not measure trajectories. The measurement is carried out along three coordinate axes. To measure the selected

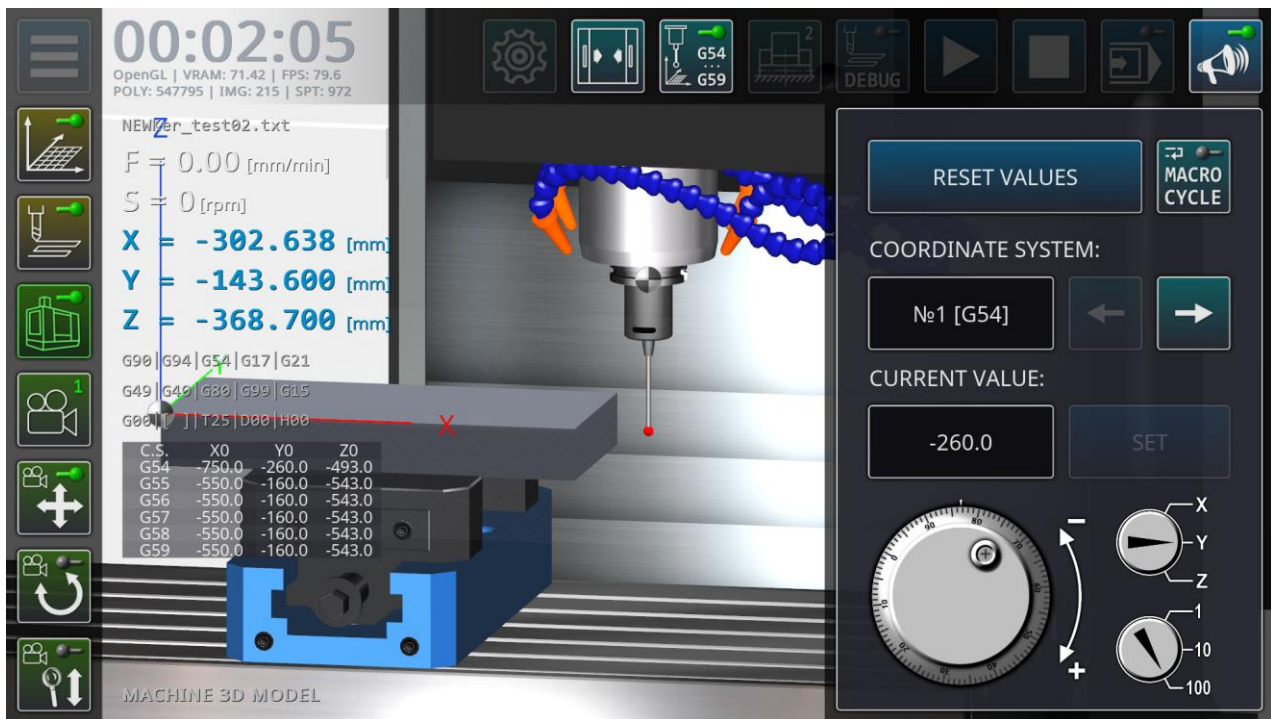
area, with the left mouse button you need to select two control points lying on the trajectory (Fig. 15). Measurements can be carried out at two arbitrary points belonging to different sections of the trajectory. In addition, during the measurement process it is possible to identify characteristic corner points of the workpiece.



**Figure 15** – Measuring Trajectory Sections in Debug Mode

## 5. Mode for Determining the Position of Workpiece Working Coordinate Systems Using a Measuring Sensor

In the workpiece binding mode, the tool installed in the machine spindle is replaced by a 3D probe (Touch Probe). The control panel for the workpiece binding process is displayed on the right side of the screen. A table of part correctors is displayed in the lower left part of the screen (Fig. 16).



**Figure 16** – View of the Simulator Main Screen in the Workpiece Binding Mode

The workpiece offset table contains the machine coordinates of the position of all working coordinate systems G54-G59. Using the control panel buttons on the right side of the screen, you can: reset the values of all zero offsets, select the current edited coordinate system, set the spatial position of the probe using handwheels that simulate the remote control of a manual pulse generator (MPG), and also start an automatic binding cycle (the «Macro Cycle» button).

The automatic cycle of workpiece binding is not edited by the user, and snaps to the lower left corner of the workpiece (when viewed from above).



By analogy with a real MPG console, on the control panel on the right side of the screen you can select the current axis of the machine, the manual feed speed, and also feed the measuring tool by rotating the corresponding flywheel.

If the measuring tool does not touch the workpiece surface, the main elements of the control panel are in the active state (Fig. 17.a). At the moment the measuring tool touches one of the sides of the workpiece, the control panel goes into the mode of recording the value in the correction table (Fig. 17.b.). In this mode, only the «Set» button remains active. In this case, the remaining controls go into inactive mode.



**Figure 17** – View of the Control Panel for the Workpiece Binding Process in the Probe Positioning Mode (a) and in the Measured Value Recording Mode (b)

## 6. Mode for Determining Tool Offset and Diameter Using a Measuring Sensor

In tool binding mode, the tool binding sensor is displayed on the machine's coordinate table. The position of the sensor corresponds to the machine zero along the X and Y axes. The tool binding process is controlled in the same way as the workpiece binding mode. The right side of the screen displays a control panel for the tool binding process. A table of tool correctors is displayed in the lower left part of the screen (Fig. 18).



**Figure 18** – View of the Simulator Main Screen in the Tool Binding Mode

The tool offset table contains the radius and length values of all used tools. Using the control panel buttons on the right side of the screen, you can: reset the values of all tool offsets, select the current tool being measured, set the spatial position of the tool using the handwheels, and also start an automatic binding cycle (the «Macro Cycle» button).

The automatic tool binding cycle is not editable by the user and performs standard tool binding by length and radius. Spindle speed is switched on automatically for large diameter tools.

By analogy with the workpiece binding mode, if the tool does not touch the measuring sensor, the main elements of the control panel are in the active state (Fig. 19.a). At the moment the tool touches the upper (in the case of binding by length) or side (in the case of binding by radius) surface of the sensor, the control panel goes into the mode of recording the value in the correction table (Fig. 19.b.). In this mode, only the «Set» button remains active. In this case, the remaining controls go into inactive mode.



**Figure 19** – View of the Control Panel for the Tool Binding Process in Positioning Mode (a) and in Measured Value Recording Mode (b)

## 7. Brief Instructions for Working with the Milling Machine Simulator in the Mode of Connecting a Physical CNC Controller

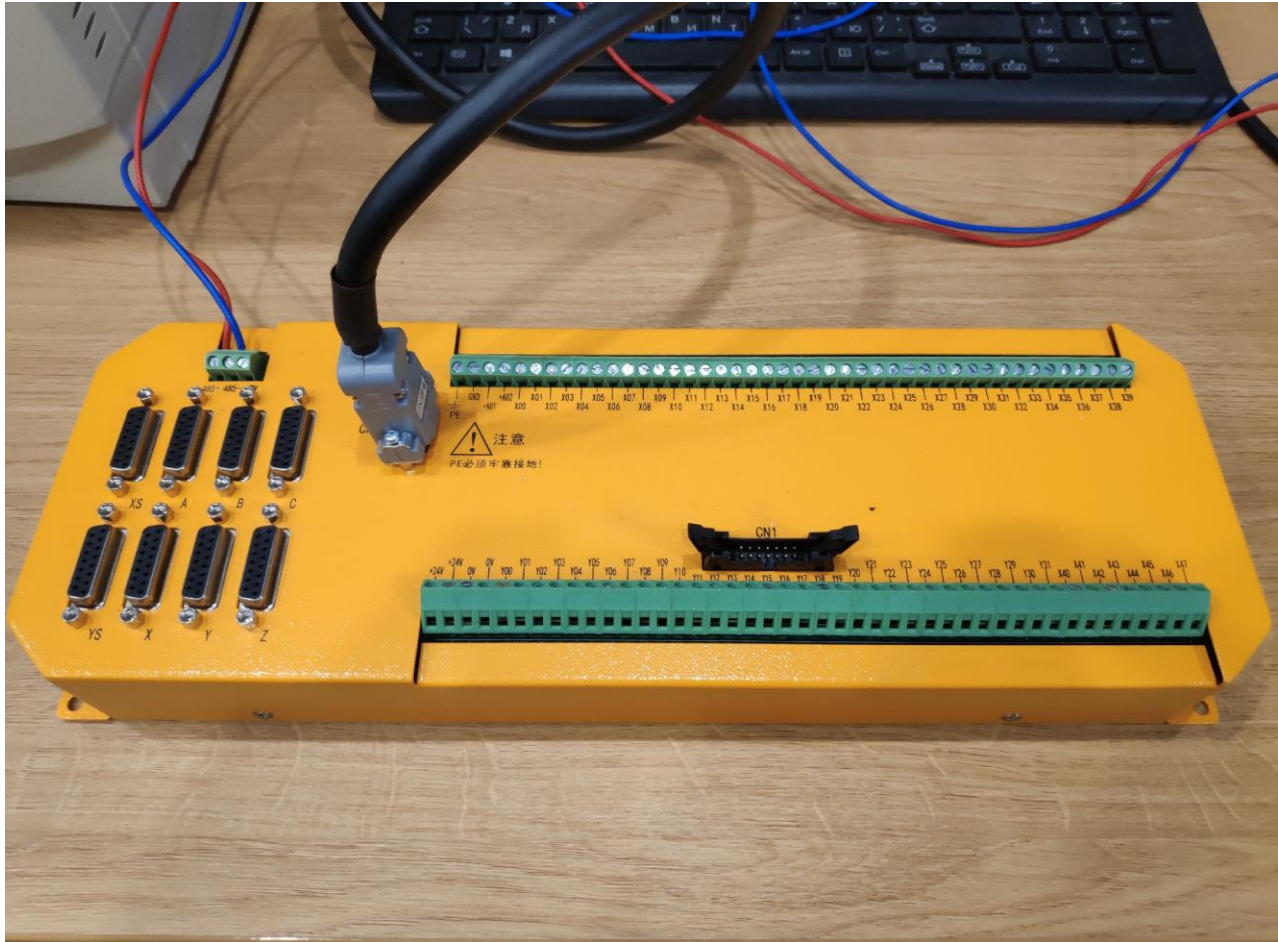
This section applies only to a special version of the simulator that supports hardware interaction with the NEWKer CNC Series 16iM physical CNC controller.

The software and hardware training complex includes the following components: an ergonomic safe case, a NEWKer CNC Series 16iM CNC controller (Fig. 20), a control panel with a handwheel (MPG), a relay or interface board (Fig. 21), a «485 to USB» converter, set of cables, personal computer with simulation software.



**Figure 20** – Appearance of the NEWKer CNC Series 16iM CNC controller



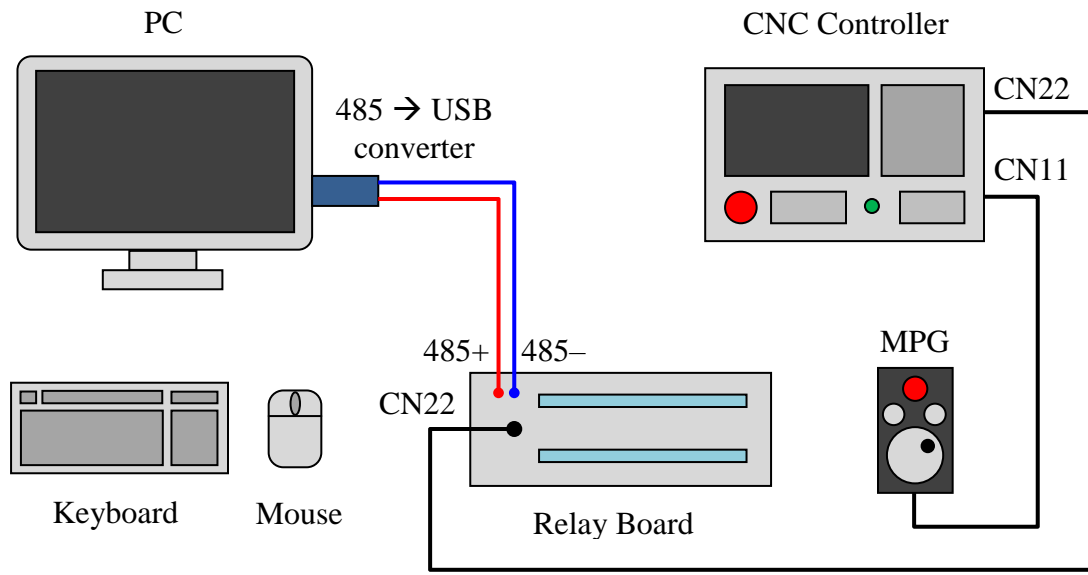


**Figure 21** – Appearance of the CNC Controller Interface Board

The CNC controller is connected to a personal computer in accordance with the ModBus protocol via the RS-485 bus. This uses the maximum serial port bandwidth of 115200 bps.

The main block of the CNC controller is connected to the relay board using the «CN22» connector. A remote control with a handwheel is connected to the main block of the CNC controller using the «CN11» connector.

The «485+» and «485-» outputs of the relay board are connected to the «D+/A+» and «D-/A-» connectors of the «485 to USB» converter, respectively. The «485 to USB» converter connects directly to the USB port of a personal computer. Figure 22 shows a simplified connection diagram for all devices.



**Figure 22** – Simplified Diagram of the Software and Hardware Training Stand

When connecting a CNC controller to a computer, on the side of the simulator program, it is necessary to configure the connection parameters. In the “media” subdirectory of the simulator program there is a configuration file “comm.ini”, which includes the following parameters:

7	COM port number
115200	bit/sec throughput
n	parity
8	data bits
1	stop bits
127.0.0.1	IP address for internal data transfer on the PC side via UDP protocol
7000	port number for internal data transfer on the PC side via UDP protocol
10000	time for automatic termination of the communication process after the simulator is completed

The COM port number should always match the actual port number in the Windows system settings. Port bandwidth, parity, data bits and stop bits must be the same on both the PC side and the CNC controller side.

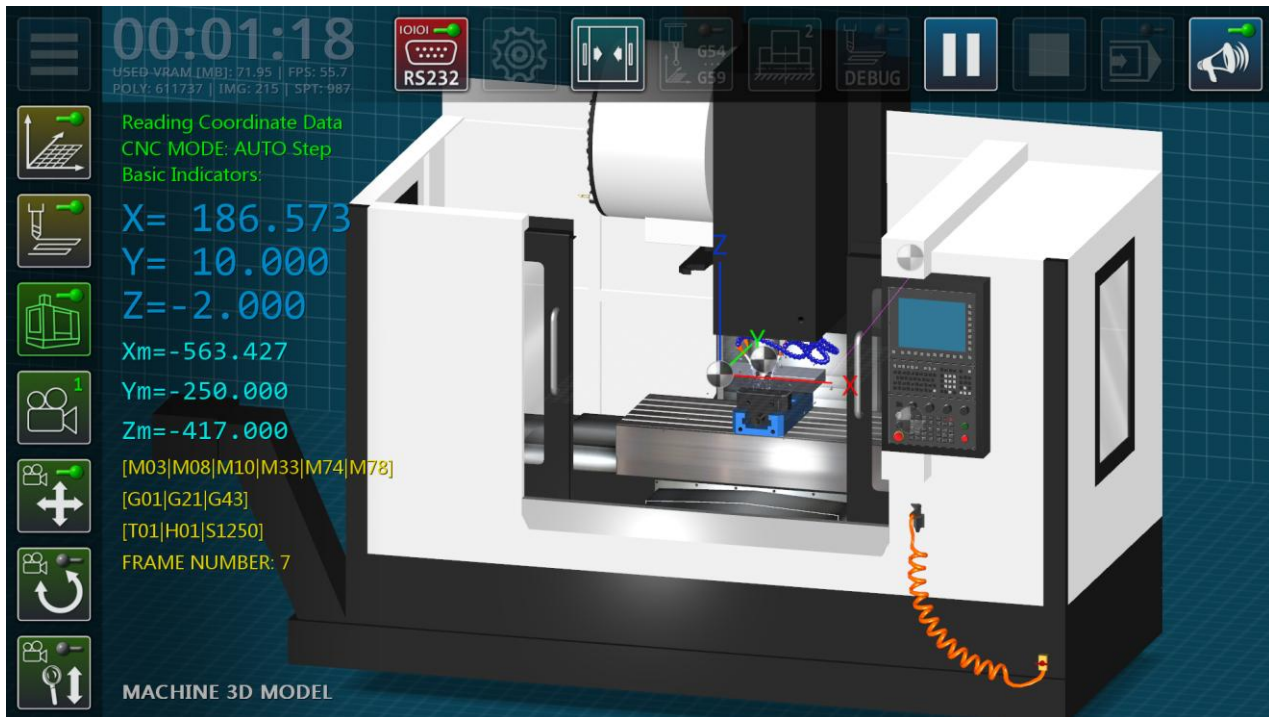
To enable the hardware control mode of a virtual machine in the simulator program, you need to left-click on the button with the image of a COM port on the top control panel (Fig. 23). This button is available only in a special version of the program. The cutting tool, workpiece parameters, and method of securing the workpiece on the coordinate table must first be configured in the simulator. The CNC controller must be launched before the simulator program.



**Figure 23** – Location of the Button to Turn On/Off the Hardware Simulation Control Mode

After pressing the hardware control enable button, the simulator goes into communication mode with the CNC controller. In this mode, access to the main menu, the function of constructing a trajectory model (debugging mode), as well as other functions of the simulator are blocked. The main screen displays communication parameters with the CNC controller, current tool coordinates, as well as active technology codes of the controller (Fig. 24).

Data exchange with the CNC controller occurs by continuously alternating sending requests to read coordinate data and system parameters of the controller.



**Figure 24** – View of the Simulator Main Screen in Hardware Control Mode

Since the speed and volume of data transfer is limited by the communication protocol used by the CNC controller, the simulator follows a strict request queue. To achieve maximum synchronization when reading coordinate data, other controller parameters are not read during this period.

In the hardware control mode of a virtual machine, execution of the control program on the CNC controller side is possible only in automatic step-by-step mode. In this case, the simulator program forcibly turns on the step-by-step mode of executing the control program on the CNC controller and gives a command to move to the next block of the control program upon completion of the current block.

If the CNC controller is set to Hand Continuous/Incremental or Hand MPG modes, the «START»/«PAUSE» button on the top panel of the simulator is not available. In this mode, the simulator executes all manual control commands from the CNC controller – turning on/off spindle speed, supplying/turning off cooling, manual positioning of the tool.

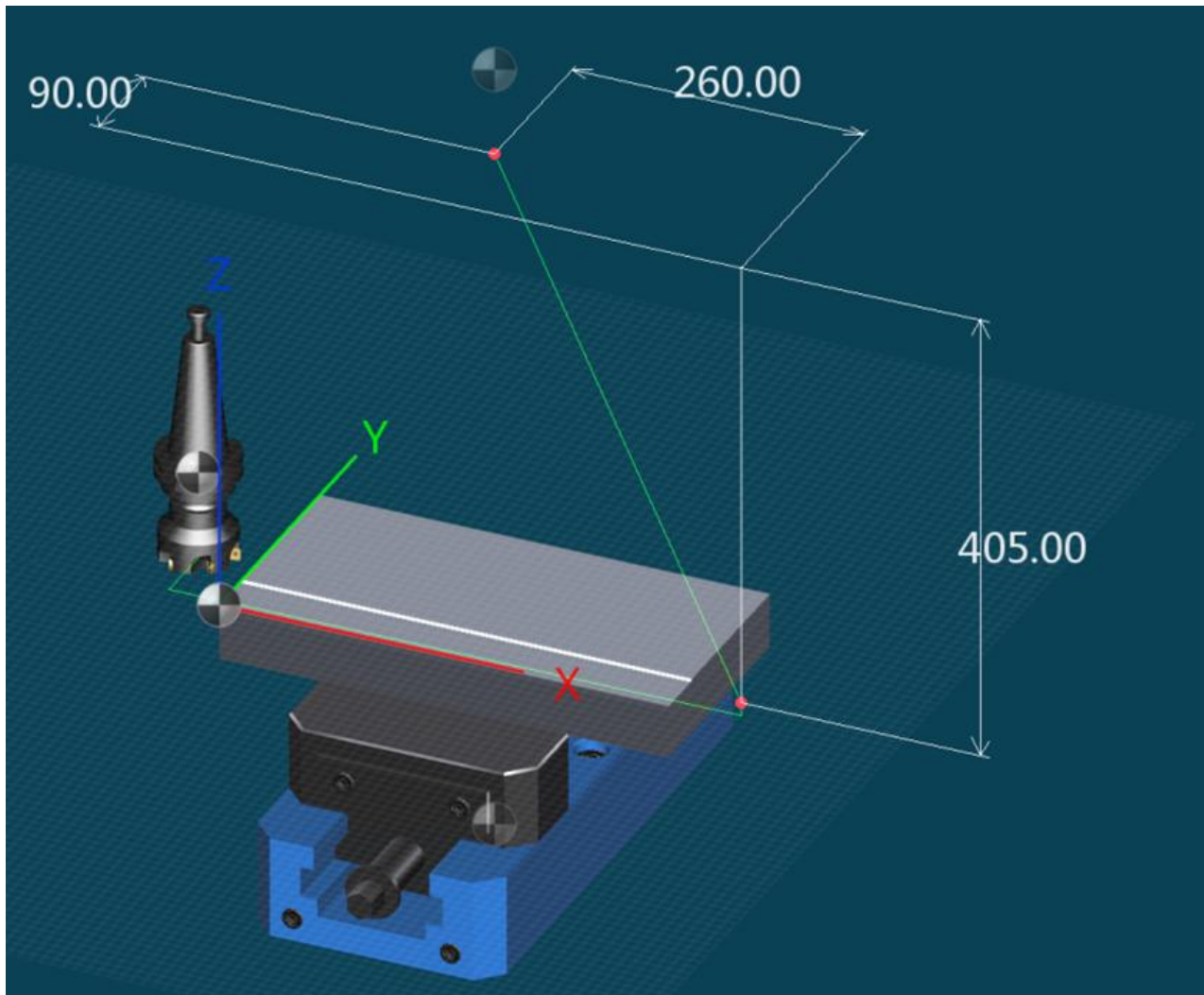
If the CNC controller is set to automatic control mode, the «START»/«PAUSE» button of the simulator program is available for pressing when reading system information from the CNC controller. In this case, the execution of the control program on the CNC controller can be suspended by pressing the «PAUSE» soft button in the simulator, and continued by similarly pressing the «START» button (Fig. 25).



**Figure 25** – Image of the «START»/«PAUSE» Button in the Simulator Program in Automatic Control Mode on the CNC Controller

In the software pause mode of the control program execution (on the simulator side), data exchange between the personal computer and the CNC controller is temporarily suspended. In this mode, linear sections of the trajectory in simulators can be measured using the standard measurement function (if the device has a mouse). The execution of the control program on the CNC controller side can also be paused and continued using the physical «PAUSE» and «START» buttons on the front panel.





**Figure 26** – Measuring the Geometric Characteristics of a Tool Path Section

During the execution of the control program, the simulator constructs the tool trajectory. To measure individual straight sections of the trajectory, the execution of the control program must be paused using the «START»/«PAUSE» soft button, and then with two subsequent clicks of the left mouse button, select the starting and ending points of the selected section of the trajectory. In this case, the linear dimensions of the trajectory section along three coordinate axes will be displayed on the main screen of the simulator (Fig. 26).

Also, measurements can be carried out at two arbitrary points belonging to different sections of the trajectory. In addition, during the measurement process it is possible to identify characteristic corner points of the workpiece.

In hardware control mode, you can activate the dimensional binding mode of the workpiece and tool using measuring sensors.

The communication session between the simulator program and the CNC controller is terminated by clicking the hardware mode on/off button again.

## **8. Quick Reference for Milling Operations Programming in the Simulator**

This section duplicates the built-in help system of the simulator program. The following describes the basic principles of milling programming using standard (ISO) G code. The rules for processing and executing commands in the simulation model are as close as possible to real control systems based on Haas, Fanuc, Fagor, etc. in the ISO code specification, but may differ from real systems.

The built-in command interpreter supports the following control codes and letter addresses:

- G00 – Rapid movement of the tool;
- G01 – Linear interpolation with working feed;
- G02 – Circular interpolation clockwise;
- G03 – Circular interpolation counterclockwise;
- G04 – Program execution delay;
- G15 – Cancel polar coordinates mode;
- G16 – Enable polar coordinates mode;
- G17 – Selecting the x-y working plane;
- G18 – Selecting the z-x working plane;
- G19 – Selecting the y-z working plane;
- G20 – Inch system of measurements;
- G21 – Metric system of measurements;

G28 – Return to the reference point;  
G29 – Return from the reference point;  
G40 – Cancel tool radius compensation;  
G41 – Left tool radius compensation;  
G42 – Right tool radius compensation;  
G43 – Tool length compensation in the positive direction;  
G44 – Tool length compensation in the negative direction;  
G49 – Cancel tool length compensation;  
G50 – Reset all scaling factors to 1.0;  
G51 – Enable coordinate scaling mode;  
G52 – Setting the local coordinate system;  
G53 – Transition to the machine coordinate system;  
G54 – Selecting the working coordinate system №1;  
G55 – Selecting the working coordinate system №2;  
G56 – Selecting the working coordinate system №3;  
G57 – Selecting the working coordinate system №4;  
G58 – Selecting the working coordinate system №5;  
G59 – Selecting the working coordinate system №6;  
G65 – Calling a macro program;  
G68 – Enabling coordinate rotation mode;  
G69 – Disabling coordinate rotation mode;  
G73 – High-speed drilling fixed cycle;  
G74 – Fixed cycle of cutting the left thread with a tap;  
G76 – Fine boring fixed cycle;  
G80 – Cancel fixed cycle;  
G81 – Fixed drilling cycle;  
G82 – Fixed cycle drilling with a delay;



G83 – Fixed cycle of periodic drilling;  
G84 – Fixed cycle of cutting the right thread with a tap;  
G85 – Boring fixed cycle;  
G86 – Boring fixed cycle;  
G87 – Reverse boring fixed cycle;  
G88 – Boring fixed cycle;  
G89 – Boring fixed cycle;  
G90 – Absolute positioning mode;  
G91 – Relative positioning mode;  
G92 – Workpiece coordinate system transformation;  
G94 – Feed per minute mode;  
G95 – Feed per revolution mode;  
G98 – The mode of returning to the starting point in fixed cycles;  
G99 – The mode of returning to the R-point level in fixed cycles;  
M00 – Pausing a control program execution;  
M01 – Pausing a control program execution;  
M02 – End of control program;  
M03 – Enable spindle rotation clockwise;  
M04 – Enable spindle rotation counterclockwise;  
M05 – Spindle rotation stop;  
M06 – Tool change operation;  
M07 – Turn on coolant supply;  
M08 – Turn on coolant supply;  
M09 – Turn off coolant supply;  
M30 – End of control program;  
M97 – Calling an internal subroutine;  
M98 – Calling an external subroutine;

M99 – End of subroutine;

X – x-axis coordinate;

Y – y-axis coordinate;

Z – z-axis coordinate;

F – Linear speed of tool movement;

S – Spindle speed;

T – Tool number in tool change command;

D – Tool radius offset number;

H – Tool length offset number;

C – Chamfer value;

R – Arc radius, tool retraction distance in fixed cycles, angle value in coordinate rotation mode, or rounding radius;

Q – Depth of passage in the fixed cycle of periodic drilling;

L – The number of subroutine calls or the number of repetitions of the fixed cycle;

P – Subroutine identifier, delay time or scale factor in scaling mode;

I – Offset of the center of the arc along the x-axis or scale factor in scaling mode;

J – Offset of the center of the arc along the y-axis or scale factor in scaling mode;

K – Offset of the center of the arc along the z-axis or scale factor in scaling mode.

### ***8.1. Linear Interpolation Commands G00 and G01***

Standard format for G00 and G01 commands blocks:

**G00 X\_ Y\_ Z\_;**

**G01 X\_ Y\_ Z\_ (F\_);**

The G00 command is for rapid movement of the tool in a straight line. The G00 code is modal, i.e. is valid until another command from the same group is called, such as G01, G02 or G03. This command is active in the initial state of the CNC system. Rapid positioning is used for approaching the tool to the workpiece before machining, and also for retracting the tool from the workpiece after machining.

The G01 command is for linear movement of the tool at the cutting feedrate specified by address F in the same block as G01 or in previous blocks of the NC program. Code G01 is modal. Linear interpolation movement is used for straight line cutting.

### ***8.2. Circular Interpolation Commands G02 and G03***

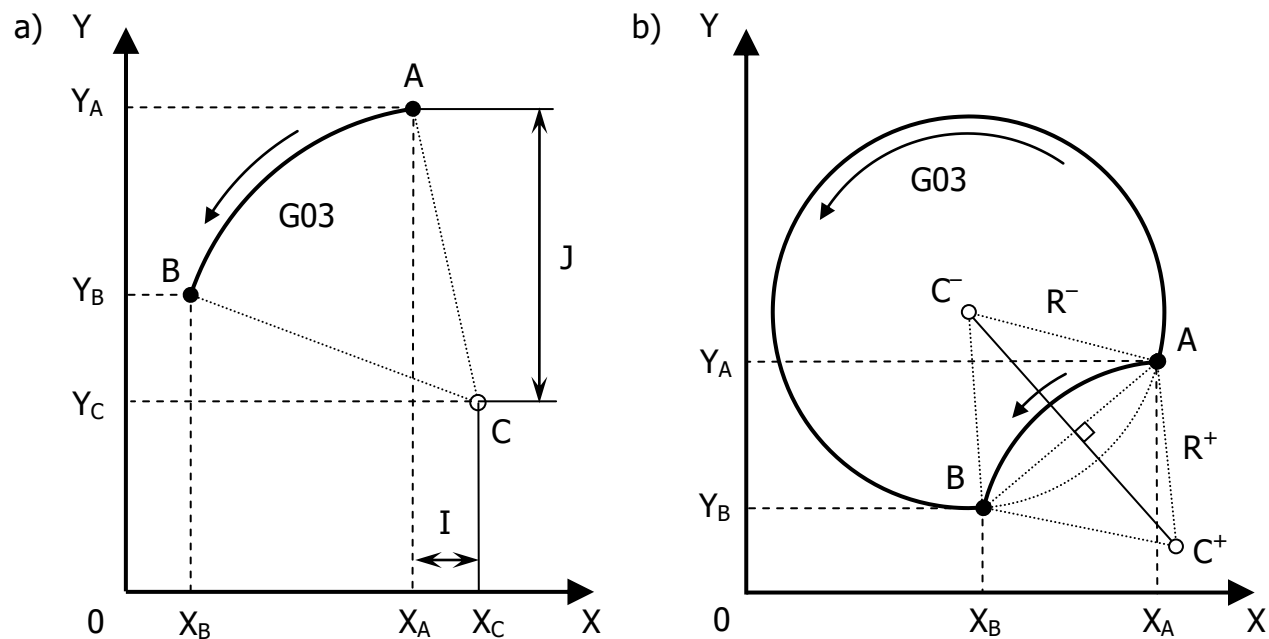
Standard format for G02 and G03 commands block:

**G02 (G03) X\_ Y\_ Z\_ I\_ J\_ K\_ (R\_) (F\_);**

Circular interpolation is used for turning curved surfaces, the shape of which is described by an arc of a circle of a certain radius. There are two arc programming methods.

The first method is to set the coordinates of the arc center using the addresses I, J, K and the end point X, Y, Z, and the radius of the arc is calculated automatically. The second method involves specifying the radius of the arc R and the coordinates of the end point X, Y, Z, while the coordinates of the arc center are calculated automatically.

Clockwise circular interpolation is specified with the G02 command, and counterclockwise circular interpolation is specified with the G03 command, respectively. When specifying the center of the arc with addresses I, J, K, offsets are specified relative to the starting point of the arc along the x, y, z axes, respectively (Fig. 27.a). If the arc is given by radius R, the control system takes into account the sign of the radius to calculate the coordinates of the arc center (Fig. 27.b).



**Figure 27 – Arc Programming Principle**

Arcs can be programmed in three possible work planes: x-y, x-z and y-z, set by codes G17, G18 and G19, respectively. To specify the center of the arc in the x-y plane, addresses I and J are used, in the x-z plane – I and K, in the y-z plane – J and K. If an end point coordinate that is not related to the current work plane is specified in the block with the G02 (G03) the control system performs helical interpolation along an axis perpendicular to the working plane.

When defining an arc, the control system checks the distances from the center of the arc to the start and end points. If the distances have a discrepancy that exceeds the allowable value, the control system will issue an error message.

### ***8.3. Chamfering and Rounding Programming***

Chamfers and roundings can be added between linear and circular motion blocks. To specify a chamfer, an additional address C is used, separated from the main addresses of the block by a comma:

**G01 (G02/G03) X\_ Y\_ Z\_ (I\_ J\_ K\_ (R\_)), C\_;**

After address C, a numerical value is entered that specifies the chamfer. The chamfer can be constructed in the places of conjugations: «segment–segment», «segment–arc», «arc–segment» and «arc–arc».

To set the rounding, an additional address R is used, separated from the main addresses of the block by a comma:

**G01 (G02/G03) X\_ Y\_ Z\_ (I\_ J\_ K\_ (R\_)), R\_;**

After the address R, a numerical value is entered that specifies the radius of the rounding. Rounding can be built at the mates: «segment–segment», «segment–arc», «arc–segment» and «arc–arc».

Mandatory conditions for programming chamfers and rounding:

- the chamfer/rounding is programmed only in the block that specifies the movement at the working feed;
- the following block must also specify the movement at the working feed;
- empty blocks or block-comments following the block defining chamfer/rounding are not allowed;
- the following block must not include G-codes that are not related to linear/circular interpolation (codes G01, G02, G03, G90, G91, G20, G21, G04 and M-codes other than M06 are allowed);
- if the end point of the chamfer/rounding is outside the path, an error message is issued.

Chamfers and roundings can be programmed in scaling (G51), coordinate rotation (G68) and tool radius compensation (G41/G42) modes. Chamfers and roundings can only be programmed in Cartesian coordinates (G15) in any of the three working planes (G17/G18/G19).

#### ***8.4. Program Execution Delay Command G04***

Standard format for a G04 command block:

**G04 P\_ [X\_] [U\_];**

The G04 command is designed to execute a delay for the time specified by address P. Also, the control system allows the use of addresses X and U to specify the delay time. If the time is set with a decimal point, the control system defines the set value as seconds, for example, P2.5 – delay the execution of the control program by 2.5 seconds. If the time is set without a decimal point, the control system defines the set value as milliseconds, for example P4000 – delay the execution of the control program by 4000 milliseconds or 4 seconds. The G04 command can be programmed in the same block with other commands, and the delay will be executed when the block is completed.

### ***8.5. Working Plane Selection Commands G17, G18, G19***

Standard format for G17, G18 and G19 commands block:

**G17 (G18) (G19) \_;**

The G17, G18 and G19 commands are for selecting the current work plane – x-y, x-z and y-z, respectively. In this simulation, changing the work plane affects only the parameters for constructing arcs of circles using codes G02 and G03. Offset commands G41, G42, G43, G44, and canned cycles G73-G84 are only executed in the x-y work plane.

### ***8.6. Polar Coordinate Mode Control Commands G15, G16***

Standard format for G15 and G16 commands block:

**G17 (G18) (G19) G90 (G91) G16;**

...

**G15 \_;**

The modal G16 command turns on the polar coordinate mode, which is overridden by the G15 command. In the polar coordinates mode, any movements are programmed using a radius vector that has a given length and angle of rotation in the selected work plane. The positive direction of the angle is the counterclockwise direction of the positive direction of the 1st axis of the selected plane, and the

negative direction is the clockwise direction. Radius and angle can be specified with an absolute value command (G90) or an incremental value command (G91).

When the radius (distance between the zero point of the polar coordinate system and the specified position) is specified by the absolute value method (G90), the zero point of the workpiece coordinate system is set as the origin of the polar coordinate system.

When the radius (distance between the current position and the specified position) is specified by the incremental value method (G91), the current position of the tool is set as the origin of the polar coordinate system.

Numeric values for radius and angle are programmed with X, Y and Z addresses depending on the current work plane:

G17 (XY plane): The X-axis specifies the radius and the Y-axis specifies the angle;

G18 (ZX plane): The Z-axis specifies the radius and the X-axis specifies the angle;

G19 (YZ plane): The Y-axis specifies the radius and the Z-axis specifies the angle.

### ***8.7. System of Measures Selecting Commands G20, G21***

Standard format for G20 and G21 commands block:

**G20 (G21) \_;**

The G20 command sets the programming mode to inches. The G21 command sets the programming mode to millimeters. Codes G20 and G21 are modal. The default programming mode (G20 or G21) is determined by the application settings.

### ***8.8. Command to Return to the Reference Point G28***

Standard format for a G28 command block:

**G28 X\_ Y\_ Z\_;**

The G28 command is intended for accelerated movement of the executive bodies to the zero point of the machine with the possibility of setting an intermediate point. Note: In this simulation, the machine zero point corresponds to the reference point as well as the tool change position (on the z-axis).

The number of coordinates specified in one block with the G28 code determines the number of axes along which zero return is performed. If the G28 command is given without specifying the X, Y, Z coordinates, the control system returns the coordinate table and spindle to zero positions simultaneously on all axes without moving to an intermediate point.

The X, Y, Z coordinates specified in one block with the G28 code determine the position of the intermediate point (in the active working coordinate system), through which the control system returns the machine's executive bodies to zero positions.

Below are examples of using the G28 command.

```
G91 G28 Z100 X0;
```

In this block of the program, the spindle is raised by 100 units along the z axis relative to (G91) the starting point and then simultaneously brought to the machine zero of the spindle (along the z axis) and the coordinate table (along the x axis).

```
(G91)
```

```
N1 G28 Z0;
```

```
N2 G28 X0 Y0;
```

This example illustrates the use of the G28 command in two blocks in succession. In block N1, the spindle is returned to the zero position (along the z axis). In block N2, the coordinate table is returned to the zero position (simultaneously along the x and y axes). The intermediate points in this example correspond to the origins of the movements because the relative positioning mode



(G91) is set to zero increments in the z-axis (in block N1) and the x, y axes (in block N2).

G90 G28 X150 Y0 Z200;

In this block of the program, the accelerated linear movement of the tool to the point [x=150; y=0; z=200] of the active work coordinate system specified by absolute coordinates (G90) and then simultaneously bring the spindle (along the z axis) and the coordinate table (along the x and y axes) to machine zero.

It is recommended to use the G28 command in relative positioning mode (G91) to avoid the tool colliding with the workpiece.

In case the G28 command is called in automatic tool radius compensation mode (G41/G42), the offset vector is temporarily reset to zero. After executing the G28 command, automatic tool radius compensation resumes.

In case the G28 command is called in the automatic tool length compensation mode (G43/G44), the offset vector is temporarily reset to zero in the area of return to the reference point. If the tool length compensation cancel command (G49) is specified in the same block as the G28 command, the offset vector is reset to zero at the return point to the intermediate point. After executing the G28 command, automatic tool length compensation resumes unless the G49 offset cancel code is programmed.

It is not allowed to use the G28 command in the scaling (G51) and rotation (G68) modes of coordinates, as well as in the same block with the G53 command.

### ***8.9. Command to Return from the Reference Point G29***

Standard format for a G29 command block:

**G29 X\_ Y\_ Z\_;**

The G29 command is designed to rapidly move the tool to the programmed position through an intermediate point previously defined using the G28 code.

The number of coordinates specified in one block with the G29 code determines the number of axes along which movement is performed. If a G29 command is given without specifying the X, Y, Z coordinates, the end point of the move is the intermediate point previously defined using the G28 code.

In case the G29 command is called in the automatic tool radius compensation mode (G41/G42), the offset vector is temporarily reset to zero. After executing the G29 command, automatic tool radius compensation resumes.

In case the G29 command is called in the automatic tool length compensation mode (G43/G44), the offset vector is restored at the end point of the move.

It is not allowed to use the G29 command in the scaling (G51) and rotation (G68) modes of coordinates, as well as in the same block with the G53 command.

#### ***8.10. Tool Radius Compensation Mode Control Commands G40, G41, G42***

Standard format for G40, G41 and G42 commands blocks:

**G41 (G42) \_;**

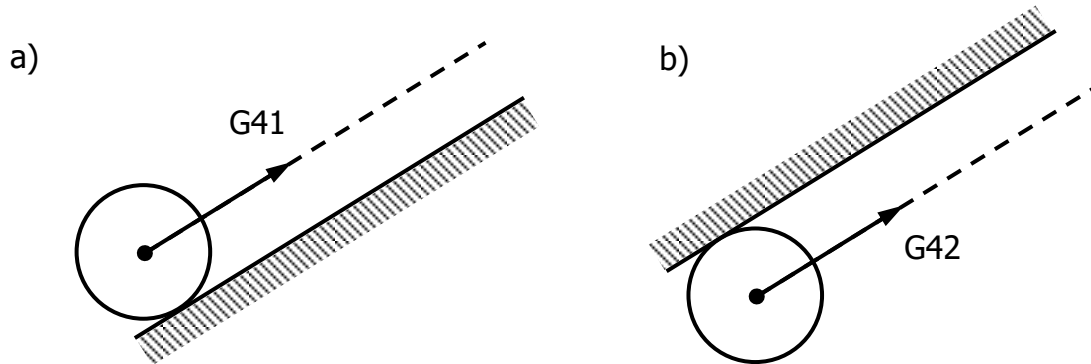
...

**G40 \_;**

The control system has a built-in algorithm for calculating equidistance when processing contours in the automatic tool radius compensation mode. The algorithm for calculating the equidistance is based on the general principles of calculations of the Fanuc control system. Depending on the version of the simulator program, the principle of constructing an equidistance when going around sharp corners may vary. Next, the principle of calculating the equidistance of the Fanuc system (TYPE A) will be described.

The automatic tool radius compensation mode is enabled using modal codes G41 and G42. If in the direction of travel it is necessary to move the tool to the left of the programmed path by the value of its radius, the G41 command is used (Fig. 28.a). To move the tool to the right of the path, use the G42 command (Fig. 28.b).

The control system automatically calculates the path that the tool should follow based on the programmed workpiece contour and the tool radius value found in the tool offset table.



**Figure 28** – Principle of Tool Movement in Automatic Radius Compensation Mode

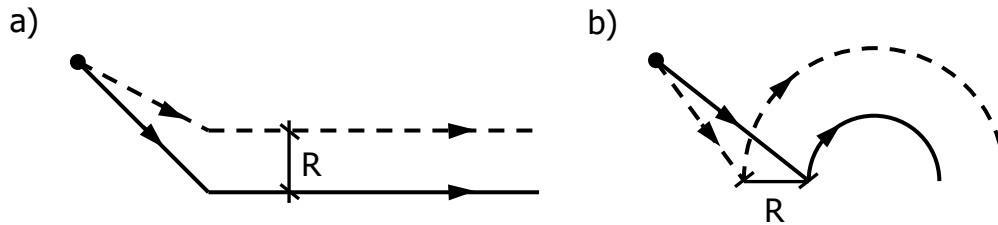
Automatic radius correction can technically be performed in any working plane, however, changing the working plane with codes G17-G19 in automatic correction mode is prohibited. For this machine configuration, it is advisable to perform automatic correction for the tool radius only in the working plane x-y (G17).

The G41 and G42 commands are canceled by the G40 modal code. Also, the tool radius compensation vector is temporarily reset to zero when executing G28, G29, G53, M06, G73-G84 commands.

Before executing a G41 (G42) command (or in the same block with this command), a tool offset number must be specified using address D, which must match the tool number in use, specified by address T. When the simulation starts, no tool offset number is specified (D0), therefore, the control system calculates the compensation vector with a zero value of the tool radius, and the equidistant line coincides with the line of the programmed trajectory.

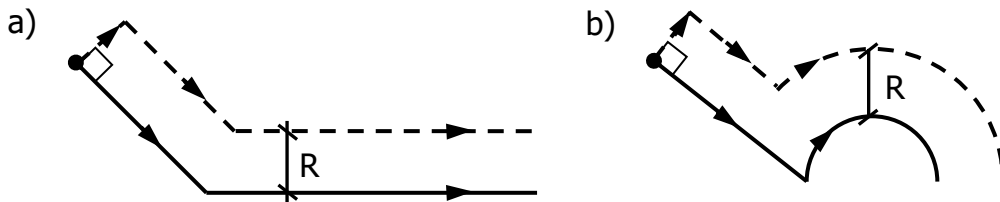
Activating the tool radius compensation mode (G41 or G42) is only possible when G00 or G01 commands (linear interpolation) are active. If the compensation mode is turned on in a section of circular interpolation (G02 or G03), the control

system will issue an error message. Thus, if a G41 (G42) command is programmed in one block with a straight-line move, the offset is entered smoothly in a straight line, the beginning of which is located on the programmed path, and the end is on the offset of the next move (Fig. 29 a, b).



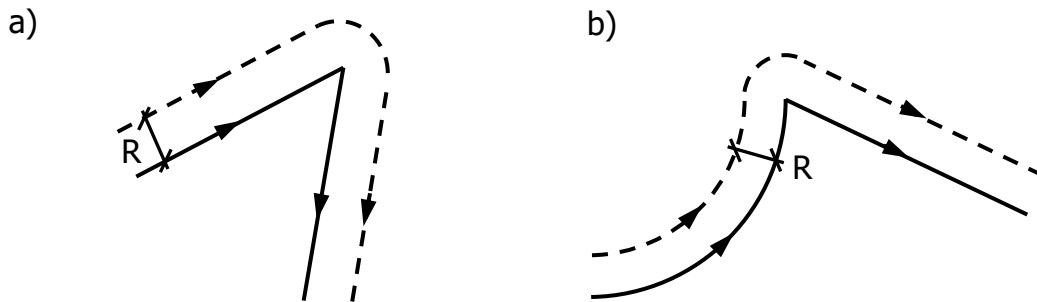
**Figure 29** – Enabling Automatic Tool Radius Compensation Mode in One Block with Linear Movement

If the command G41 (G42) is programmed in a separate block, the offset is entered at right angles to the path (Fig.30 a, b).



**Figure 30** – Enabling Automatic Tool Radius Compensation Mode by Programming in a Separate Block

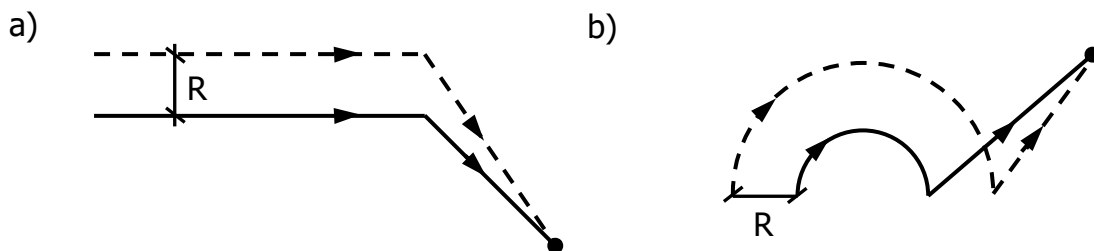
In the process of calculating the equidistant, the control system checks the following blocks of the control program containing displacements in order to determine the intersection points of the current section with the next section of the calculated path in the working plane. The control system allows vertical movements (along the z-axis) in the tool radius compensation mode. Bypassing corners from the outside is carried out by constructing an additional circular movement (Fig. 31 a, b).



**Figure 31** – The Principle of Bypassing an Sharp Angle in the Automatic Tool Radius Compensation Mode

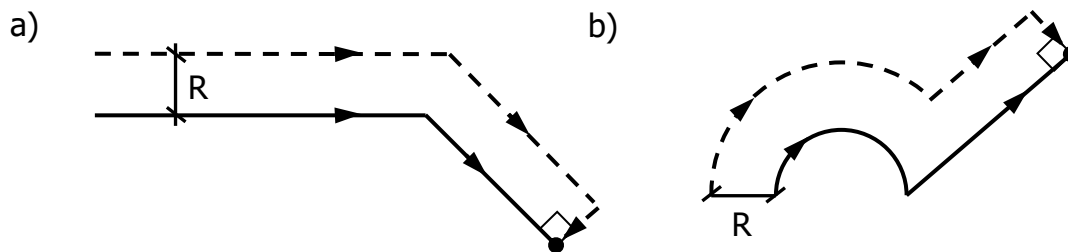
When bypassing the arc from the inside, if the radius of the programmed arc is less than the radius of the tool, the control system will issue an error message.

The automatic tool radius compensation cancel command G40 can be similarly programmed in a block with linear movement (G00 or G01) or in a separate block. If the offset mode is turned off in a circular interpolation section (G02 or G03), the control system will issue an error message. Thus, if a G40 command is programmed in a single block with a straight-line move, the exit from the equidistant is smoothly in a straight line, the beginning of which is located at the equidistant of the previous move, and the end of which is on the path of the current move (Fig. 32 a, b).



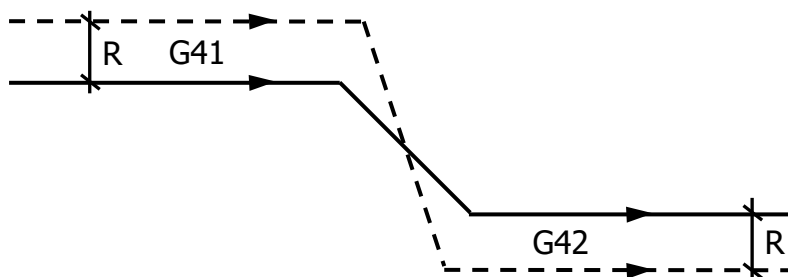
**Figure 32** – Turning Off the Automatic Tool Radius Compensation Mode in One Block with Linear Movement

If the G40 command is programmed in a separate block, the equidistant exit is at right angles to the path (Fig. 33 a, b).



**Figure 33** – Turning Off the Automatic Tool Radius Compensation Mode by Programming in a Separate Block

In automatic tool radius compensation mode, the direction of the compensation vector can be changed by calling the opposite command.



**Figure 34** – Changing the Direction of the Correction Vector on a Linear Section of the Trajectory

For example, if after turning on the offset mode on the left G41 (in subsequent blocks of the program) you specify the G42 command, the end point of the equidistance of this section of the path will be calculated from the opposite side of the path (Fig. 34). Changing the direction of the correction vector is possible only on straight sections of the trajectory.

Below is an example of using the G42 and G40 commands.

(G54 G90)

N01 G00 X50 Y50 Z-10;

N02 G91 G42 G01 X100 Y100;

N03 X100;

N04 G03 X50 Y50 R50;

N05 G01 Y100;

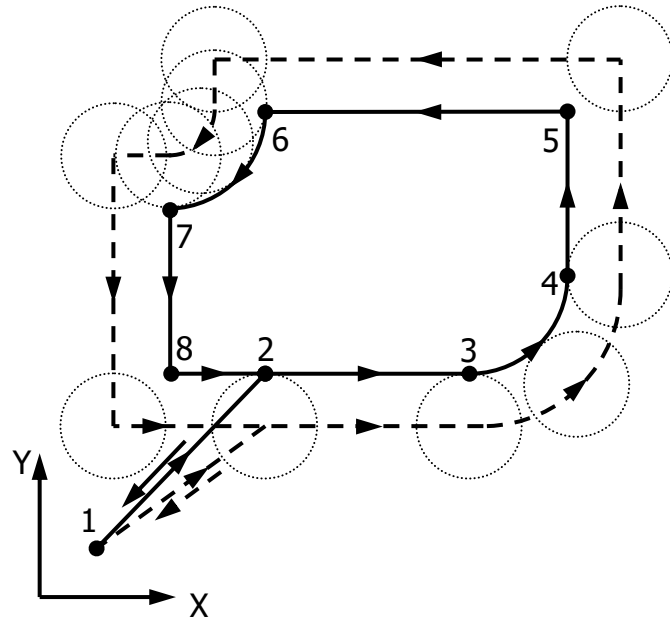
N06 X-150;

N07 G02 X-50 Y-50 R50;

N08 G01 Y-100;

N09 X50;

N10 G40 X-100 Y-100;



**Figure 35** – Example of Machining a Closed Contour in Automatic Tool Radius Compensation Mode

The result of executing the program is shown in Figure 35. When performing movements in the automatic tool radius compensation mode, the display shows the actual coordinates of the tool moving equidistant. The equidistant line is also displayed in conjunction with the main toolpath line.

### **8.11. Tool Length Compensation Mode Control Commands G43, G44, G49**

Standard format for G43, G44 and G49 commands blocks:

**G43 (G44) Z\_ H\_;**

...

**G49 \_;**

During the execution of the control program, the control system positions the base point of the tool along the programmed path. In this case, the peak of the tool is located below its base point by the value of the full overhang, which is added from the height of the visible part of the tool and the height of the chuck. To make the tool nose move along the programmed path rather than the base point, use the G43

and G44 tool length compensation enable commands. In this simulation, tool length compensation is performed only along the z-axis.

The offset direction is selected with the G43 or G44 commands. Selecting the offset value (tool length) from memory is done by entering the H code in one block with the G43 (G44) command. If the offset number is not specified in the same block as the offset enable command, the control system uses the previous offset value. When the simulation starts, the tool offset number is not specified (H0), therefore, the control system calculates the offset vector with zero tool length.

The G43 and G44 commands only indicate that longitudinal compensation should be applied. The control system starts applying the offset when a movement along the z-axis is programmed. If the offset is changed in the circular interpolation section (provided that the movement along the z-axis is specified), the control system will issue an error message. If, after turning on the length compensation mode, only the x and y axes are moved, the z axis corrective movement is not performed in that move.

When a G43 command is executed, the tool length offset value (stored in offset memory) specified by the H code is added to the z-coordinate of the end point of movement. When a G44 command is executed, the same value is subtracted from the z-coordinate of the move end point. The resulting coordinates indicate the end position of the tool after compensation, regardless of whether the positioning mode is set to absolute (G90) or relative (G91).

Below is an example of using the G43 command.

(T02)

N1 S1000 M03;

N2 G43 H02 Z0.1;

N3 M08;



In block N1, the spindle starts to rotate clockwise at a frequency of 1000 rpm. In block N2, automatic tool length compensation is enabled with offset H02 corresponding to the currently used tool T02. In the same block, the tool rapidly moves along the z axis to the 0.1 level. In block N3, the coolant supply is turned on.

To cancel the automatic length compensation mode, program the G49 command or select the zero offset H0. If the offset value is changed by changing the offset number in address H, the old offset value is replaced by the new one.

The length compensation vector is temporarily reset to zero when using the G28 and G53 commands. The length compensation mode is canceled when an M06 tool change command is called without automatic resumption.

### ***8.12. Scaling Mode Control Commands G50, G51***

Standard format for G50 and G51 commands blocks:

**G51 X\_ Y\_ Z\_ P\_ (I\_ J\_ K\_);**

...

**G50 \_;**

The modal G51 command turns on the coordinate scaling mode, which is canceled by the G50 command. The control system uses the scaling center to determine the scaling parameters, which can be specified in the same block together with the G51 command using X, Y, Z addresses. If the scaling center is not specified in the G51 command block, the control system uses the last tool position in the active work coordinate system as the scaling center. In addition to the scaling center coordinates, scaling factors are specified in the G51 command block. The control system allows the use of one scaling factor for three axes (P) or the use of independent factors for each axis (I, J, K).

With the G51 command, the control system multiplies by a scaling factor all X, Y, Z end points for rapid, linear and circular feeds. G51 also scales the I, J, K and

R parameters for G02 and G03. The control system shifts all these coordinates relative to the scaling center.

If a general scaling parameter P with an integer value is specified in the G51 command block, the control system takes into account the scaling factor divided by 1000 (for example, parameter P2000 corresponds to a scaling factor of 2.0, i.e. all movements along the x, y, z will be doubled). If the scaling parameter P is set with a fractional numeric value (using a decimal point), the control system takes into account the scaling factor without change (for example, parameter P1.5 corresponds to a scaling factor of 1.5, i.e. all movements along the x, y, z axes will be increased by 1.5 times). The values of the independent scaling parameters I, J, K for the x, y, z axes, respectively, are processed in a similar way.

Below are examples of using the G51 command.

(G54 G90)

N1 G00 X50 Y50 Z20;

N2 G01 Z-5;

N3 M97 P100;

N4 G00 Z20;

N5 X35 Y35;

N6 G01 Z-5;

N7 G51 X65 Y65 P2.0;

N8 M97 P100;

N9 G50;

N10 M30;

N100 G01 X80;

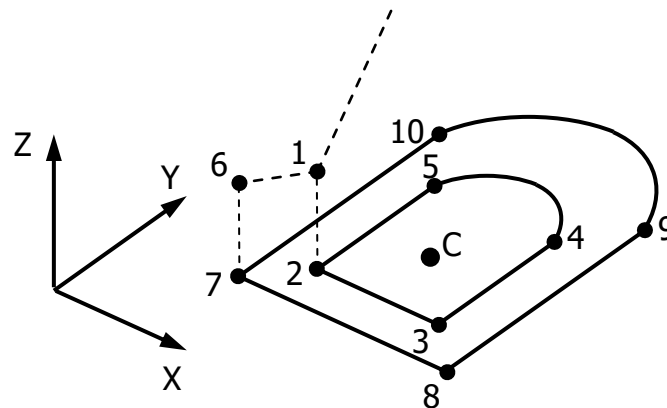
N101 G01 Y80;

N102 G03 X50 I-15;

N103 G01 Y50;

N104 M99;

In block N1, the accelerated movement of the tool to point №1 [ $x=50$ ;  $y=50$ ;  $z=20$ ] of the active work coordinate system (G54). In block N2, the tool is lowered at linear feed to point №2 [ $x=50$ ;  $y=50$ ;  $z=-5$ ]. In block N3, an internal subroutine is called for processing a contour of four movements. The internal subroutine starts from block N100. In block N100, the tool linearly moves to point №3 [ $x=80$ ;  $y=50$ ;  $z=-5$ ]. In block N101, the tool linearly moves to point №4 [ $x=80$ ;  $y=80$ ;  $z=-5$ ]. In block N102, a counterclockwise circular movement is carried out to point №5 [ $x=50$ ;  $y=80$ ;  $z=-5$ ]. In block N103, the tool linearly moves to point №2 [ $x=50$ ;  $y=50$ ;  $z=-5$ ], from which contour processing began. In block N104, the internal subroutine is terminated (M99), and control returns to block N4, in which the tool rapidly rises to point №1 [ $x=50$ ;  $y=50$ ;  $z=20$ ]. In block N5, the tool moves rapidly to point №6 [ $x=35$ ;  $y=35$ ;  $z=20$ ], and then in block N6 on a linear feed it falls to point №7 [ $x=35$ ;  $y=35$ ;  $z=-5$ ]. This places the tool at the starting point for contour processing. In block N7, the scaling mode (G51) is switched on, specifying the scaling center at point C [ $x=65$ ;  $y=65$ ;  $z=-5$ ] and a scaling factor of 2.0 for all axes. All subsequent movements are automatically recalculated by the control system relative to the scaling center. In block N8, the internal contour subroutine is called again. The tool sequentially moves to points №8, №9, №10 and №7.



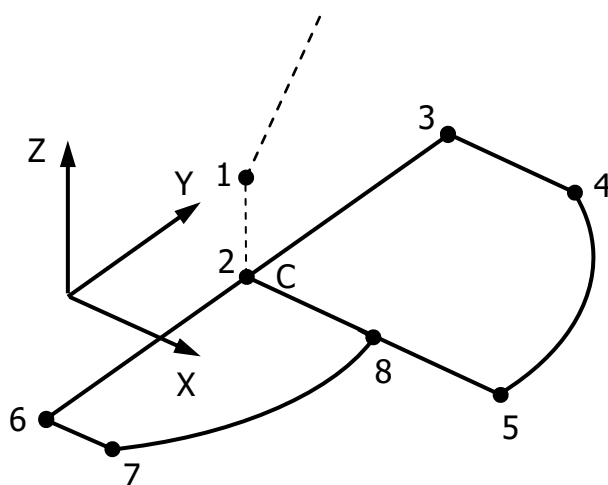
**Figure 36** – Example of Scaling a Closed Contour with One Coefficient Along the Coordinate Axes

At the end of the subroutine in block N9, the scaling mode is turned off with the G50 command. Block N10 ends the execution of the main program with the corresponding instruction M30. The sequence of movements from the above example is shown in Figure 36.

```
(G54 G90)
N1 G00 X50 Y50 Z10;
N2 G01 Z-5;
N3 M97 P101;
N4 G51 I500 J-1000;
N5 M97 P101;
N6 G50;
N7 M30;
N101 G41 G01 Y150;
N102 X100;
N103 G02 X200 Y50 R100;
N104 G40 G01 X50;
N105 M99;
```

In block N1, the accelerated movement of the tool to point №1 [x=50; y=50; z=10] of the active work coordinate system (G54). In block N2, the tool is lowered at linear feed to point №2 [x=50; y=50; z=-5]. In block N3, an internal subroutine is called for processing a contour of four movements. The internal subroutine starts from block N101. In block N101, the tool linearly moves to point №3 [x=50; y=150; z=-5], this turns on automatic tool radius compensation on the left (G41). In block N102, the tool linearly moves to point №4 [x=100; y=150; z=-5] in auto correction mode. In block N103, a circular movement is carried out clockwise to point №5 [x=200; y=50; z=-5]. In block N104, the tool linearly moves to point №2 [x=50; y=50; z=-5] from which contour processing started, automatic tool radius

compensation is disabled in this area. In block N105, the internal subroutine is terminated (M99) and control returns to block N4, in which the scaling mode (G51) is turned on. In block N4, the coordinates of the scaling center are not specified, therefore, the control system takes the current position of the tool (point №2) as the scaling center. In the same block, the scaling factors I and J for the x and y axes, respectively, are specified, and the scaling factor for the y axis has a negative sign, therefore, the contour obtained as a result of scaling will be mirrored relative to the scaling center along the y axis.



**Figure 37** – Example of Scaling a Closed Contour with Different Coefficients  
Along the Coordinate Axes

Parameters I and J are set to integer values, so the control system takes into account thousandths of these values (actual scaling factors for the x and y axes are 0.5 and  $-1.0$ , respectively). In block N5, the internal contour subroutine is called again. The tool sequentially moves to points №6, №7, №8 and №2. At the end of the subroutine in block N6, the scaling mode is turned off with the G50 command. Block N7 ends the execution of the main program with the corresponding command M30. The sequence of movements from the above example is shown in Figure 37.

It should be noted that in the above example, as a result of mirroring the contour along the y axis relative to the scaling center, the direction of the

programmed arc is automatically reversed (the arc from point №7 to point №8 is described counterclockwise), and the direction of the tool radius compensation vector is also changed. In the case of mirroring along two axes, these changes are not carried out by the control system at the same time.

The G51 command and its accompanying parameters are always programmed in a separate block. It is not allowed to enable scaling in the automatic tool radius compensation mode (G41/G42), while automatic tool radius compensation can be performed in the scaling mode. The correction vector itself is not scaled.

The modal G50 command cancels the coordinate scaling mode if it was previously activated with the G51 command. The G50 command can be specified in a separate program block or in a block together with other commands.

### ***8.13. Local Coordinate System Setting Command G52***

Standard format for a G52 command block:

**G52 X\_ Y\_ Z\_;**

The G52 command creates a local coordinate system within the active work coordinate system by offsetting the zero of the active work coordinate system by the values specified by the X, Y, Z addresses. The specified offset is canceled by calling the G52 command again with X, Y, Z values set to zero. Thus, the local coordinate system is aligned with the work coordinate system. Also, the local coordinate system is canceled by calling the work coordinate system selection command (G54-G59) again.

When setting the local coordinate system, the sequentially programmed move commands in the absolute positioning mode (G90) are the coordinate values in the local coordinate system. Each work coordinate system (G54-G59) has its own local coordinate system.

Below is an example of using the G52 command.

```
(G54)  
N1 G00 X50 Y50 Z50;  
N2 G52 X100 Y100;  
N3 G01 X0 Y0;  
N4 G02 X100 R50;  
N5 G52 X0 Y0;
```

In block N1, the tool is rapidly moved to the point [x=50; y=50; z=50] of the active work coordinate system (G54). In block N2, the local coordinate system is set: the origin of the working coordinate system is shifted by 100 units in the positive direction of the x-axis and by 100 units in the positive direction of the y-axis, while the position of the tool in the working coordinate system is determined by the point [x=-50; y=-50; z=50]. In block N3, the tool moves linearly to the point [x=0; y=0; z=50] of the local coordinate system. In block N4, a circular movement is carried out clockwise to the point [x=100; y=0; z=50] of the local coordinate system. In block N5, the local coordinate system is aligned with the work coordinate system (G54) along the x and y axes.

After calling the G52 command, when performing movements, the display shows the coordinates of the tool in the local coordinate system.

The G52 command and its accompanying parameters are always programmed in a separate block. It is not allowed to use the G52 command in the modes of scaling (G51) and rotation (G68) of coordinates.

#### ***8.14. Transition to Machine Coordinate System Command G53***

Standard format for a G53 command block:

**G53 \_;**

When a G53 command is executed, the control system temporarily cancels work coordinate offsets and uses the machine coordinate system. The G53 command

is non-modal, i.e. valid only in the block in which it is specified. Together with the G53 code, linear or circular movements can be programmed.

The G53 command only works in absolute positioning mode (G90). If a G53 command is given in relative positioning mode (G91), the control will issue an error message.

Below are examples of using the G53 command.

(G54 G90)

N1 G00 X100 Y50 Z200;

N2 G53 G01 X0 Y0;

N3 G01 X200;

In block N1, the tool is rapidly moved to the point [x=100; y=50; z=200] of the active work coordinate system (G54). In this example, the origin of the working coordinate system is determined by the point [x=-125; y=-125; z=0] relative to the machine zero corresponding to the intersection of the diagonals on the surface of the coordinate table and the spindle axis (note: relevant only for this simulation). In block N2, the tool at linear feed moves to machine zero simultaneously along the x and y axes, while the position of the tool in the work coordinate system is determined by the point [x=125; y=125; z=200]. In block N3, the tool is linearly moved to the point [x=200; y=125; z=200].

(G54 G90)

N1 G00 X125 Y125 Z200;

N2 G53 X100;

N3 G53 Y-100;

N4 G53 X0;

N5 G53 Y0;

N6 X0 Y0 Z50;



In block N1, the tool is rapidly moved to the point  $[x=125; y=125; z=200]$  of the active work coordinate system (G54). The offset of the origin of the working coordinate system relative to machine zero corresponds to the previous example. In block N2, the tool moves rapidly along the x-axis to the mark  $x=100$  relative to machine zero, while the position of the tool in the working coordinate system is determined by the point  $[x=225; y=125; z=200]$ . In block N3, the tool moves rapidly along the y axis to the mark  $y=-100$  relative to machine zero, while the position of the tool in the working coordinate system is determined by the point  $[x=225; y=25; z=200]$ . In block N4, the tool rapidly returns to machine zero along the x axis, which corresponds to the point  $[x=125; y=25; z=200]$  in the working coordinate system. In block N5, the tool rapidly returns to machine zero along the y-axis, which corresponds to the point  $[x=125; y=125; z=200]$  in the working coordinate system. Block N6 carries out the accelerated movement of the tool to the point  $[x=0; y=0; z=50]$  of the working coordinate system.

When making movements in the machine coordinate system using the G53 code, the display shows the machine coordinates of the tool.

In case the G53 command is called in automatic tool radius compensation mode (G41/G42), the offset vector is temporarily reset to zero. After executing the G53 command, automatic tool radius compensation resumes.

In case the G53 command is called in the automatic tool length compensation mode (G43/G44), the offset vector is temporarily reset to zero. After executing the G53 command, automatic tool length compensation resumes unless the G49 offset cancel code is programmed.

It is not allowed to use the G53 command in the scaling (G51) and rotation (G68) modes of coordinates, as well as in the same block with the G28 or G92 commands.

### ***8.15. Commands for Selecting the Working Coordinate System G54-G59***

Standard format for G54-G59 commands block:

**G54 (G55, G56 ...) \_;**

This simulation provides the ability to use six predefined work coordinate systems. The selection and setting (as active) of the working coordinate system is carried out using modal codes: G54, G55, G56, G57, G58, G59.

The zero offsets of each work coordinate system relative to the machine zero of the coordinate table are pre-configured in the corresponding section of the workpiece parameters screen. During the execution of the part program, these offsets can be changed using the G92 and G52 commands.

Below is an example of using the G54, G55 and G56 commands.

(G54)

N1 G90 G55 G00 X0 Y0 Z20;

N2 G02 G91 Y100 R-60;

N3 G90 G56 G00 X150 Y250 Z0;

N4 G54;

N5 G00 X0 Y0 Z0;

In block N1, the transition from the active work coordinate system (G54) to the work coordinate system G55 is performed by rapid movement in the absolute positioning mode (G90). In the G55 coordinate system, the end position of the tool is determined by the point [x=0; y=0; z=20]. In block N2, in the relative positioning mode (G91), a clockwise circular motion to the point [x=0; y=100; z=20] of the active work coordinate system (G55). In block N3, the transition from the active work coordinate system (G55) to the G56 work coordinate system is carried out by rapid movement in the absolute positioning mode (G90). In the G56 coordinate system, the end position of the tool is determined by the point [x=150; y=250; z=0]. In block N4, the working coordinate system G54 is selected as the active one. In

block N5, the tool rapidly moves to the zero of the active work coordinate system (G54).

When moving, the display shows the coordinates of the tool in the active work coordinate system.

The G54-G59 commands can be specified in a separate program block or in blocks together with other commands.

It is not allowed to use commands G54-G59 in the modes of scaling (G51) and rotation (G68) of coordinates.

### ***8.16. Macro Program Calling Command G65***

Standard format for G65 command block:

**G65 P\_ (L\_) (A\_ B\_ C\_ D\_ E\_ F\_ H\_ I\_ J\_ K\_ M\_ Q\_ R\_ S\_ T\_ U\_ V\_ W\_ X\_ Y\_ Z\_);**

The G65 command is used to call a macro program, the code of which is located in the main control program or in a separate program (with its own block numbering). If the macro program is located in the main program code, it must have a header of the form OXXXX specified in a separate block. Also, the macro code can be specified in an additional tab of the text editor. In the same block as the G65 command, the P parameter must be specified. The P integer parameter specifies a four-digit macro program number.

The optional integer parameter L specifies the number of consecutive calls to the macro, for example, L3 means that the macro will be called 3 times in a row. Up to 999 consecutive macro calls are allowed. If parameter L is not specified in a block with a G65 command, the macro is called once.

Any macro called must end with an M99 exit code. After the completion of the macro program, control is transferred to the block following the block with the macro call command (G65).

Macro programs use the space of local variables #1-#33, which can be assigned values by entering literal arguments after the G65 command. Any changes to the values of local variables in the macro program code affect only the macro program itself and are not returned to the main program.

The main method of passing arguments to the macro program involves the use of various letter addresses A-Z (except G, L, N, O, P). In this case, each letter address corresponds to a specific number of the local variable:

A – #1	I – #4	T – #20
B – #2	J – #5	U – #21
C – #3	K – #6	V – #22
D – #7	M – #13	W – #23
E – #8	Q – #17	X – #24
F – #9	R – #18	Y – #25
H – #11	S – #19	Z – #26

Example: G65 P1001 L12 A35.5 B-14.2 Q8;

In this example, the macro program O1001 is called 12 times, and when the macro program is called, the values of local variables are transferred to it: #1=35.5; #2=-14.2; #17=8.

An alternative method of passing arguments to a macro is to use the letter addresses A, B, C, I, J, K. In this case, each of the addresses I, J, K can be specified up to 10 times in one G65 block. Correspondence of letter addresses to numbers of local variables is as follows:

A – #1	I4 – #13	I8 – #25
B – #2	J4 – #14	J8 – #26
C – #3	K4 – #15	K8 – #27
I1 – #4	I5 – #16	I9 – #28
J1 – #5	J5 – #17	J9 – #29
K1 – #6	K5 – #18	K9 – #30
I2 – #7	I6 – #19	I10 – #31
J2 – #8	J6 – #20	J10 – #32
K2 – #9	K6 – #21	K10 – #33
I3 – #10	I7 – #22	
J3 – #11	J7 – #23	
K3 – #12	K7 – #24	

Example: G65 P1002 L3 I68.4 J-13 K4 I-18.5 J-9 K50.2 I19.2 J-1;

In this example, the macro program O1002 is called 3 times, and when the macro program is called, the values of local variables are transferred to it: #4=68.4; #5=-13; #6=4; #7=-18.5; #8=-9; #9=50.2; #10=19.2; #11=-1; #12=88.8.

Macro programs are processed in the same way as external subroutines. You can use up to 20 nested macro programs with their own code structure - conditional and unconditional jumps, loops, etc.

### ***8.17. Rotation Mode Control Commands G68, G69***

Standard format for G68 and G69 commands block:

**G68 X\_ Y\_ Z\_ R\_;**

...

**G69 \_;**

The modal G68 command turns on the coordinate rotation mode, which is canceled by the G69 command.

To determine the rotation parameters, the control system uses the center of rotation, which can be specified in the same block together with the G68 command using X, Y, Z addresses. If the center of rotation is not specified in the G68 command block, the control uses the last tool position in the active work coordinate system as the center of rotation.

In addition to the coordinates of the center of rotation, the rotation angle is specified in the G68 command block using the address R. If the rotation angle is specified as an integer value, the control system takes into account the value divided by 1000 (for example, parameter R45000 corresponds to a coordinate rotation angle of 45 degrees). If the angle of rotation is given as a fractional numeric value (using a decimal point), the control system takes into account the value of the angle without change (for example, parameter R180.0 corresponds to a coordinate rotation angle of 180 degrees).

A positive angle value rotates the coordinates counterclockwise about the center of rotation. A negative angle value rotates the coordinates clockwise about the center of rotation.

When coordinate rotation (G68) is used in relative positioning mode (G91), the angle of rotation changes from the previous value.

Below is an example of using the G68 and G69 commands.

(G54 G90)

N1 G01 X200 Y200 Z-5;

N2 M97 P4 L3;

N3 M30;

N4 G91 G68 R120.0;

N5 G01 X100;

N6 Y100;

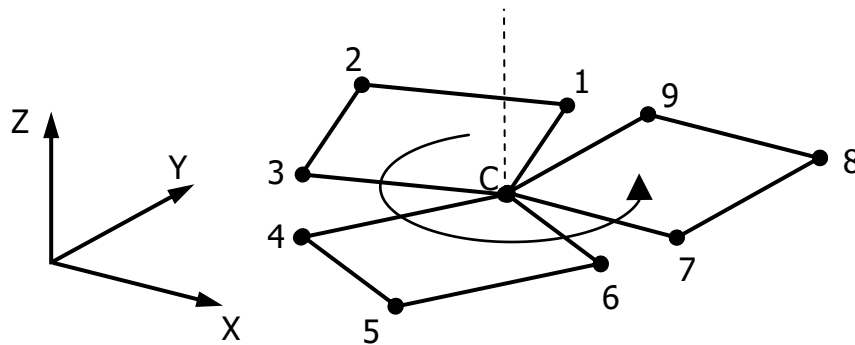
N7 X-100;

N8 Y-100;

N9 G69 G90 M99;

In block N1, the working movement of the tool to point C [x=200; y=200; z=-5] of the active work coordinate system (G54). In block N2, an internal subroutine is called for processing a contour of four movements. The internal subroutine starts from block N4. The N4 block turns on the relative positioning mode (G91) and the coordinate rotation mode (G68) with a rotation angle (R) of 120 degrees counterclockwise. Point C is taken as the center of rotation. In block N5, the tool moves linearly 100 units in the positive direction of the x-axis to point №1. In block N6, the tool linearly moves 100 units in the positive direction of the y-axis to point №2. In block N7, the tool linearly moves 100 units in the negative x-axis direction to point №3. In the N8 block, the tool moves linearly 100 units in the negative y-axis direction to point C. In the N9 block, the coordinate rotation mode is turned off

(G69), the absolute positioning mode is turned on (G90), and the internal subroutine is ended (M99). The internal subroutine is called three times in succession (L3). Each time the subroutine is called, the angle of rotation increases by 120 degrees. After executing three circuits, control returns to block N3, which ends the execution of the main program with the corresponding command M30. The sequence of movements from the above example is shown in Figure 38.



**Figure 38** – Example of Coordinate System Rotation

The G68 command and its accompanying parameters are always programmed in a separate block. Rotation cannot be enabled in automatic tool radius compensation mode (G41/G42), but automatic tool radius compensation can be performed in rotation mode.

The rotation mode can be activated in scaling mode. The scaling mode cannot be activated in rotation mode.

The modal G69 command cancels the coordinate rotation mode if it was previously activated with the G68 command. The G69 command can be specified in a separate program block or in a block together with other commands.

### **8.18. Positioning Mode Switching Commands G90, G91**

Standard format for G90 and G91 commands block:

**G90 (G91) \_;**



The G90 command turns on the absolute positioning mode, while the final coordinates of all subsequent rapid and work movements, determined by the X, Y, Z addresses, are set relative to the zero of the active work coordinate system.

The G91 command turns on the relative positioning mode, with the final coordinates of any subsequent movement, specified by the X, Y, Z addresses, being relative to the previous tool position.

Codes G90 and G91 are modal. By default, when starting the control program, the absolute positioning mode is enabled.

### ***8.19. Workpiece Coordinate System Transformation Command G92***

Standard format for a G92 command block:

**G92 X\_ Y\_ Z\_;**

The G92 command is designed to programmatically move the active working coordinate system to an arbitrary point. This moves the origin of the coordinate system so that the current position of the tool matches the values specified in the same block as the G92 command.

The G92 command itself does not cause axial movements. A G92 offset can be canceled by programming another G92 offset to change the current work offset back to the original value.

The following are examples of using the G92 command.

(G54 G90)

N1 G00 X150 Y50 Z200;

N2 G92 X0 Y0 Z0;

N3 G01 X100;

N4 G92 X250 Y50 Z200;

In block N1, the tool rapidly moves to the point [x=150; y=50; z=200] of the active work coordinate system (G54). In block N2, the working coordinate system is shifted in such a way that the current position of the tool is now determined by the

coordinates [x=0; y=0; z=0]. In block N3, the tool moves to the point [x=100; y=0; z=0]. In block N4, the working coordinate system returns to its original position.

(G55 G90)

N1 G00 X0 Y0 Z100 G92 Y-100;

N2 G01 X70 Y50 G92 Y150;

N3 G00 X0 Y0 Z100;

In block N1, the tool rapidly moves to the point [x=0; y=0; z=100] of the active work coordinate system (G55). In the same block, the offset of the working coordinate system is specified in such a way that the current position of the tool is now determined by the coordinates [x=0; y=-100; z=100]. In block N2, the tool moves to the point [x=70; y=50; z=100]. In the same block, the offset of the working coordinate system to the initial position is specified in such a way that the current position of the tool is now determined by the coordinates [x=70; y=150; z=100]. In block N3, the tool rapidly returns to the point [x=0; y=0; z=100].

In the example above, the G92 command was specified in the same block as the move, which is valid if the G92 command is entered after the move parameters (on the right side of the block). Otherwise, the control system will issue an error message.

After the simulation process is stopped, the programmed G92 offsets are reset to zero.

The G92 command cancels all G52 offsets in effect.

It is not allowed to use the G92 command in the modes of scaling (G51) and rotation (G68) of coordinates, as well as in the same block with the G53 command.

### ***8.20. Feed Mode Switching Commands G94, G95***

Standard format for G94 and G95 commands block:

**G94 (G95) \_;**

The G94 modal command turns on the minute feed mode. In this case, the value of the cutting feed specified by address F is determined in the units of [mm/min] or [inch/min] depending on the selected measurement system (G21 or G20). In G94 mode, the speed of work movements is determined only by the set F value, regardless of the spindle speed.

The G95 modal command enables the feed per revolution of the spindle. In this case, the value of the cutting feed specified by address F is determined in the units of [mm/rev] or [inch/rev] depending on the selected measurement system (G21 or G20). In G95 mode, the speed of work movements is determined by the set value F multiplied by the spindle speed S.

#### ***8.21. Tool Return Mode Control Commands G98, G99***

Standard format for G98 and G99 commands block:

**G98 (G99) \_;**

The G98 modal command enables the tool to return to the starting point defined before the start of the fixed cycle. The G99 mode command enables the tool return mode to the top plane defined by the fixed cycle R-parameter.

Using codes G98 and G99 in fixed cycles allows to bypass obstacles when drilling a series of holes in complex parts.

#### ***8.22. Fixed Cycle Cancel Command G80***

Fixed cycles are used for drilling and tapping operations, while reducing the amount of NC program. In this simulation, the main canned cycles of the Fanuc system are implemented.

Standard format for a G80 command block:

**G80 \_;**

The modal G80 command cancels any active fixed cycle. The G80 code can be programmed either in a separate block or in the same block with other

commands. By default, when the control program is started, the G80 command is active.

### ***8.23. Fixed Drilling Cycle G81***

Standard format for a G81 command block:

**G81 X\_ Y\_ Z\_ R\_ (L\_ [K\_]) (F\_);**

Cycle G81 drills a full depth hole in one work pass. The X and Y addresses specify the coordinates of the hole in the x-y plane. The Z address defines the coordinate of the bottom of the hole. The R parameter determines the coordinate of the drill retraction plane. Parameter L specifies the number of repetitions of the drilling cycle. The control system allows the use of address K instead of address L. The starting point of the cycle is the last tool position before the start of the cycle.

When a drilling cycle is executed in relative positioning mode (G91), the R parameter determines the offset from the cycle origin in the z-axis, and the hole bottom coordinate Z determines the offset relative to the retraction plane R.

Sequence of operations of cycle G81:

- 1) Rapid positioning of the tool to the starting point;
- 2) Rapid positioning of the tool in the retraction plane along the z-axis;
- 3) Linear movement of the tool to the bottom of the hole at cutting feed F;
- 4) Rapid retraction of the tool from the hole.

In the last operation, the tool retracts either to the home position (if G98 is active) or to the retraction plane (if G99 is active)

Repeating the drilling cycle multiple times is only useful in the relative positioning mode (G91), otherwise the drilling will be repeated with the same x and y coordinates of the hole center.

### ***8.24. Fixed Cycle Drilling with a Delay G82***

Standard format for a G82 command block:

**G82 X\_ Y\_ Z\_ R\_ P\_ (L\_ [K\_]) (F\_);**

The G82 cycle drills a hole to its full depth in one work pass, with a time delay at the bottom of the hole. The basic parameters of cycle G82 correspond to the parameters of cycle G81. The additional parameter P determines the delay time at the bottom of the hole. If the time is given with a decimal point, the control system defines the set value as seconds, for example, P2.5 – delay at the bottom of the hole for 2.5 seconds. If the time is set without a decimal point, the control defines the set value as milliseconds, for example P4000 – delay at the bottom of the hole for 4000 milliseconds or 4 seconds.

Sequence of operations of cycle G82:

- 1) Rapid positioning of the tool to the starting point;
- 2) Rapid positioning of the tool in the retraction plane along the z-axis;
- 3) Linear movement of the tool to the bottom of the hole at the working feed F;
- 4) Time exposure at the bottom of the hole;
- 5) Rapid retraction of the tool from the hole.

### ***8.25. Fixed Cycles of Periodic Drilling G83 and G73***

Standard format for G83 and G73 commands block:

**G83 (G73) X\_ Y\_ Z\_ R\_ Q\_ (P\_) (L\_ [K\_]) (F\_);**

The G83 cycle drills a hole, periodically retracting the tool completely out of the hole. The basic parameters of cycle G83 correspond to the parameters of cycle G82. The additional parameter Q defines the depth of one working pass. After each working pass, the tool is retracted either to the starting position or to the retraction plane, after which drilling continues until the final hole depth is reached. In the G83 cycle, a delay at the bottom of the hole is also possible. There is no delay in intermediate work passes.

The G73 cycle is similar to the G83 cycle, except that the tool is not completely retracted from the hole, but is raised by a certain amount. In this

simulation, the tool return value is not adjustable, and corresponds to the final depth of the previous work pass.

### ***8.26. Fixed Cycles of Cutting the Thread with a Tap G84 and G74***

Standard format for G84 and G74 commands block:

**G84 (G74) X\_ Y\_ Z\_ R\_ (Q\_) (P\_) (L\_ [K\_]) (F\_);**

Cycle G84 performs tapping of an internal right-hand thread. The thread parameters are determined by the value of the working feed F and the spindle speed S. The control system synchronizes the translational and rotational movements of the spindle. During the cycle, the spindle is switched on and off automatically. The retraction of the tap from the hole is always carried out at the working feed with the reverse rotation of the spindle.

Sequence of operations of cycle G84:

- 1) Rapid positioning of the tool to the starting point;
- 2) Rapid positioning of the tool in the retraction plane along the z-axis;
- 3) Working movement of the tool to the bottom of the hole (or to the first working depth) with the inclusion of clockwise spindle rotation;
- 4) Stopping the rotation of the spindle (time delay at the bottom of the hole if necessary);
- 5) Turning on the reverse rotation of the spindle and retraction the tool from the hole.

The G74 cycle is similar to the G84 cycle, except that the spindle is rotated in the opposite direction, which allows the tapping of an internal left-hand thread.

### ***8.27. Boring Fixed Cycles G76, G85, G86, G87, G88 and G89***

These cycles are used for boring holes with a specialized tool, the cutting edges of which, as a rule, have an eccentricity relative to the longitudinal axis.

Standard format for G76 command block (fine boring cycle):

**G76 X\_ Y\_ Z\_ R\_ Q\_ (P\_) (L\_ [K\_]) (F\_);**

Sequence of operations of cycle G76:

- 1) Accelerated positioning of the tool to the starting point;
- 2) Accelerated positioning of the tool in the retraction plane along the z-axis;
- 3) Working movement of the tool to the bottom of the hole;
- 4) Time exposure at the bottom of the hole;
- 5) Turning off the rotation of the spindle with precise angular positioning of the tool;
- 6) Retraction of the tool from the wall of the hole by the value Q at the working feed;
- 7) Accelerated withdrawal of the tool from the hole and reverse horizontal displacement by Q value;
- 8) Resuming the work of the spindle.

Standard format for G85 command block (boring cycle):

**G85 X\_ Y\_ Z\_ R\_ (L\_ [K\_]) (F\_);**

This cycle performs a single pass boring of a hole without dwelling at the bottom of the hole. Tool offset in the horizontal plane is not performed. Spindle rotation does not stop during the cycle. After reaching the bottom of the hole, the tool rises to the retraction plane at the working feed, and enters the initial plane already accelerated.

Standard format for G86 command block (boring cycle):

**G86 X\_ Y\_ Z\_ R\_ (L\_ [K\_]) (F\_);**

This cycle is performed in the same way as the G85 cycle, except that at the bottom of the hole, the spindle rotation is turned off and the tool is pulled out of the hole at an accelerated feed. After the cycle has been completed, the spindle rotation will resume.

Standard format for G87 command block (reverse boring cycle):

**G87 X\_ Y\_ Z\_ R\_ Q\_ (P\_) (L\_ [K\_]) (F\_);**

In this cycle, the hole is drilled from bottom to top, so the coordinate of the bottom of the hole Z is always higher than the R plane.

Sequence of operations of cycle G87:

- 1) Accelerated positioning of the tool to the starting point;
- 2) Shutdown of spindle rotation with precise angular positioning of the tool;
- 3) Accelerated horizontal displacement of the tool by the value Q;
- 4) Accelerated positioning of the tool in the retraction plane along the z axis;
- 5) Horizontal displacement of the tool by the amount Q at the working feed;
- 6) Start spindle rotation clockwise;
- 7) Linear movement of the tool to the bottom of the hole at the working feed F (movement from bottom to top);
- 8) Time delay at the bottom of the hole with a rotating spindle;
- 9) Turning off the rotation of the spindle with precise angular positioning of the tool;
- 10) Accelerated horizontal displacement of the tool by Q value;
- 11) Withdrawal of the tool from the hole at the working feed;
- 12) Resuming the work of the spindle.

Standard format for G88 command block (boring cycle):

**G88 X\_ Y\_ Z\_ R\_ (P\_) (L\_ [K\_]) (F\_);**

In a real Fanuc control system, the G88 cycle involves manual withdrawal of the tool from the hole. In this simulation, manual positioning of the tool is not performed, and the G88 cycle is performed in the same way as the G86 cycle, with the possibility of programming the time delay at the bottom of the hole.

Standard format for G89 command block (boring cycle):

**G89 X\_ Y\_ Z\_ R\_ (P\_) (L\_ [K\_]) (F\_);**

The G89 cycle is similar to the G85 cycle, except that this cycle uses a time delay at the bottom of the hole.



### ***8.28. Control Program Execution Pause Commands M00, M01***

Standard format for M00 and M01 commands block:

**\_ M01 (M00);**

In this simulation, codes M00 and M01 are executed in the same way, and serve to suspend the execution of the control program until it is manually resumed. The M00/M01 code can be programmed both in a separate block and in one block with other commands, while the control program will be suspended when the block is completed.

In this simulation, the M00/M01 code can be used to check the correctness of toolpaths and processing modes in order to debug the control program.

### ***8.29. Control Program Completion Commands M02, M30***

Standard format for M02 and M30 commands block:

**\_ M02 (M30);**

In this simulation, codes M02 and M30 are executed in the same way, and serve to complete the process of executing the control program. The M02/M30 code can be programmed in a separate block or in the same block with other commands, and the part program will be terminated after the block is executed.

When an M02/M30 code is found, parsing of the control program is completed. This automatically disables spindle rotation and coolant supply. Internal subroutines called by an M97 code must be programmed after an M02/M30 code.

### ***8.30. Spindle Rotation Control Commands M03, M04, M05***

Standard format for M03, M04 and M05 commands block:

**M03 (M04) S\_;**

The M03 and M04 commands are for turning the spindle clockwise and counterclockwise, respectively. When turning on the spindle rotation, the rotation speed [rpm] must be specified by address S in the same block with the M03/M04 command or in the previous blocks.

Spindle rotation is disabled with an M05 command.

### **8.31. Tool Switch Command M06**

Standard format for a M06 command block:

**T\_ D\_ M06;**

When command M06 is executed, the control system performs the following cycle of operations:

- 1) turning off the rotation of the spindle;
- 2) shutdown of coolant supply;
- 3) accelerated movement of the spindle to the tool change position (along the z axis);
- 4) correction of the angular position of the spindle;
- 5) rotation along the shortest distance of the tool magazine to the target position corresponding to the given tool number by address T;
- 6) tool change using a manipulator device.

After the tool change cycle is completed, the interpolation mode that was specified before the M06 command was executed with modal codes G00, G01, G02, G03 resumes. Spindle rotation and coolant are not automatically resumed, which requires them to be re-enabled with the appropriate codes.

If at the time of execution of the M06 command the number of the target tool is not specified by address T, or the number of the current tool already installed in the spindle is specified, the M06 command is ignored by the control system.

In case the M06 command is called in the automatic tool radius compensation mode (G41/G42), the compensation mode is canceled without resuming. If necessary, after each tool change, the automatic tool radius compensation mode should be re-enabled using the G41 or G42 commands, specifying the current tool offset number using address D.

In case an M06 command is called in the automatic tool length offset mode (G43/G44), the offset mode is canceled without resuming. If necessary, after each tool change, the automatic tool length compensation mode should be re-enabled using the G43 or G44 commands, specifying the current tool offset number using address H.

When trying to execute the M06 command in the scaling (G51) and rotation (G68) modes, the control system will display an error message.

### ***8.32. Coolant Control Commands M07, M08, M09***

Standard format for M07, M08 and M09 commands blocks:

**\_ M07 (M08);**

...

**\_ M09;**

In this simulation, codes M07 and M08 are executed in the same way, and serve to turn on the supply of coolant to the cutting zone. Coolant supply is disabled by code M09. Codes M07/M08 and M09 can be programmed both in a separate block and in the same block with other commands, while the coolant supply/shutdown operation will be performed upon completion of the block execution.

### ***8.33. Subroutine Control Commands M97, M98, M99***

Standard format for M97, M98 and M99 commands blocks::

**M97 P\_ (L\_);**

...

**M98 P\_ (L\_);**

...

**M99;**

The M97 command is used to call an internal subroutine whose code is located in the main control program after the M02/M30 end command. In the same

block with the M97 command, the parameter P is necessarily set. The integer parameter P determines the number of the subprogram start block, for example, P100 means transfer of control to the subprogram start block N100, while the subprogram start block must contain the address N.

The optional integer parameter L specifies the number of sequential calls to the subroutine, for example, L3 means that the internal subroutine will be called 3 times in a row. Up to 999 sequential subroutine calls are allowed. If parameter L is not specified in a block with an M97 command, the subroutine is called once.

The M98 command is used to call an external subroutine. Similar to macro programs (G65), an external subroutine can be located in the main part program or in a separate program (with its own block numbering). If the subroutine is located in the code of the main program, it must have a heading of the form OXXXX, given in a separate block. Also, the subroutine code can be specified in an additional tab of the text editor. In the same block with the M98 command, the parameter P must be specified. The complex parameter P determines the number of the subroutine, and may also contain the number of calls to the subroutine. For example, parameter P31002 means that external subroutine O1002 will be called 3 times in a row. In this simulation, if an external subroutine is located in an additional tab of the text editor, its number must match the number of the text editor's tab. If the P parameter does not contain the number of subprogram calls, the L parameter can be used in the same way as the M97 command format.

Any subroutine called must end with an M99 exit code. After the completion of the subroutine, control is transferred to the block following the block with the subroutine call command (M97 or M98).

In this simulation, multilevel subroutine calls are allowed – multiple subroutine calls from within a subroutine. In this case, the order of calling and terminating nested subroutines must be observed.

### **8.34. Parametric Programming**

The simulation model allows interpreting the main features of the Macro B parametric programming language: working with variables, unconditional and conditional jumps, as well as loops with a precondition.

#### **8.34.1. Working with Variables**

The presented control system works with shared and local variables of the form #\_. The variable name (number) is given as an integer. The number of the variable is preceded by the pound sign «#». A variable can be assigned a value using the «=» assignment operator.

The control system interprets arithmetic and logical expressions, as well as mathematical functions using nested brackets «[...]».

Supported arithmetic operators:

- 1) equality: #1=#2;
- 2) addition: #1=#2+#3;
- 3) subtraction: #1=#2-#3;
- 4) multiplication: #1=#2\*#3;
- 5) division: #1=#2/#3.

In the division operation, the use of a zero divisor value is not allowed, otherwise the control system will issue an error message.

Supported math functions:

- 1) arcsine: #1=ASIN[#2];
- 2) sine: #1=SIN[#2];
- 3) arccosine: #1=ACOS[#2];
- 4) cosine: #1=COS[#2];
- 5) arctangent: #1=ATAN[#2];
- 6) tangent: #1=TAN[#2];
- 7) square root: #1=SQRT[#2];

- 8) absolute value: #1=ABS[#2];
- 9) rounding: #1=ROUND[#2];
- 10) exponent: #1=EXP[#2];
- 11) natural logarithm: #1=LN[#2];
- 12) rounding down: #1=FIX[#2];
- 13) rounding up: #1=FUP[#2].

The trigonometric functions SIN, COS, and TAN have arguments with degrees of angle.

The control system checks the correctness of the arguments of the ASIN, ACOS, TAN, SQRT, and LN functions. The numerical value of the argument of the ASIN and ACOS functions must correspond to the range  $[-1; 1]$ . The value 90 is not allowed as an argument to the TAN function. Negative values are not allowed as an argument to the SQRT function. Values greater than zero are allowed as an argument to the LN function.

The ROUND function in assignment statements rounds the values of expressions and variables to an integer. If the ROUND function is used after alphabetic addresses, rounding is performed to 3 decimal places for metric units and 4 decimal places for inch units.

Supported logical operators:

- 1) conjunction AND: #1=#2 AND #3 (alternative designation: &);
- 2) disjunction OR: #1=#2 OR #3 (alternative designation: |).

Supported comparison operators (used in conditional expressions):

- 1) equal: EQ (alternative designation: ==);
- 2) not equal: NE (alternative designation: <>);
- 3) greater than: GT (alternative designation: >);
- 4) greater than or equal: GE (alternative designation: >=);
- 5) less than: LT (alternative designation: <);

6) less than or equal: LE (alternative designation: <=).

Variables can be given as values after letter addresses, such as X#1. Also, expressions in brackets can be used as values after literal addresses, for example, Y[#2+SIN[#3]\*#4]. Integer values of G/M codes can also be given by variables, such as G#4.

When compiling mathematical or logical expressions, it is necessary to observe the order of placement of brackets. If the number of opening and closing brackets does not match in one expression, the control system will issue an error message.

When evaluating expressions, the order of mathematical operations is performed in accordance with the rules of mathematics – parsing of the algebraic expressions is performed using the «Reverse Polish Notation» method.

#### *8.34.2. Unconditional and Conditional Transitions*

The GOTO unconditional transition operator is designed to transfer control to a specific block of the program specified after the word GOTO, for example, when executing the GOTO20 operator, a transition will be made to block N20. The block to which an unconditional transition is made must necessarily contain address N.

Below is an example of using an unconditional transition.

N10 G90 G00 X0 Y0 Z0;

N12 GOTO16;

N14 M30;

N16 G91 G01 X100;

In block N10, the absolute positioning mode (G90) is turned on and the tool is rapidly moved to the origin of the active work coordinate system. In block N12, an unconditional transition to block N16 is made, in which the relative positioning mode (G91) is turned on and a working linear movement of 100 units in the positive direction of the x-axis is performed.

A conditional transition provides branching of the control program according to the criterion for executing a conditional expression. The conditional transition operator IF is specified at the beginning of the block, after which the conditional expression in brackets must be indicated. The conditional expression must contain at least one comparison operator. If the conditional expression is true (the value of the conditional expression is true), the control executes the right side of the block (everything that follows the conditional expression). If the conditional expression is not executed (the value of the conditional expression is false), control is transferred to the block following the block with the conditional expression.

Below are examples of using conditional transition.

```
N15 IF [#1 GE 360] THEN G00 Z0;
```

In block N15, the condition is checked: «if the value of variable #1 is greater than or equal to 360». If the conditional expression is true, accelerated positioning to the mark z=0 is carried out.

The THEN statement after the conditional expression is optional.

```
N10 #3=20;
```

```
N12 IF [#1 LE #2 AND #2 GT 45] GOTO#3;
```

In block N10, variable #3 is assigned the value 20. In block N12, the condition is checked: «if the value of variable #1 is less than or equal to the value of variable #2 AND the value of variable #2 is greater than 45». If the conditional expression is true, the transition to block N20 is performed.

In the example shown, the transition parameter after the GOTO statement is set to a variable, but the block number cannot be set to a variable, i.e. the block being jumped to must contain address N with an explicit numeric value (N20).

By analogy with the StankoMachComplex control system (Tver), the simulator allows the use of multi-line conditions of the form:



<b>IF [...]</b> {... ...} <b>ELIF [...]</b> {... ...} <b>ELSE</b> {... ...}	<b>IF [...]</b> {...} <b>ELIF [...]</b> {...} <b>ELSE</b> {...}	<b>IF [...] {...}</b> <b>ELIF [...] {...}</b> <b>ELSE {...}</b>
---	--	---

In this notation format, the number of ELIF subconditions is unlimited. Curly braces are required. Between curly braces, any expression or sequence of G-codes can be specified as blocks.

The simulator allows you to perform unconditional jumps to the body of multiline conditions. In this case, the execution of the code is carried out by analogy with the BASIC programming language.

#### 8.34.3. *Loops with a Precondition*

A loop with a precondition is programmed using the syntax construct WHILE...DO...END. The WHILE statement is specified at the beginning of the block, after which the conditional expression in brackets must be specified. The conditional expression must contain at least one comparison operator. The conditional expression is followed by the loop start operator DO<sub>n</sub>, where n is an integer index of the nested loop from 1 to 3. This block is followed by blocks executed in the loop. The loop ends with the END<sub>n</sub> operator, where n is the index corresponding to the index of the beginning of the loop.

If the conditional expression is true (the value of the conditional expression is true), control is transferred to the block following the start-of-loop (DO) block. If the conditional expression is not executed (the value of the conditional expression is false), control is transferred to the block following the end of the loop (END) block.

The DO...END loop is executed as long as the conditional expression is true.

Below is an example of using a loop with a precondition.

```
N01 #1=300;  
N02 #2=200;  
N03 WHILE [#1 GT #2] DO1;  
N04 #1=#1-10  
N05 G91 X100;  
N06 Y-100;  
N07 X-100;  
N08 Y100;  
N09 END1;  
N10 M30;
```

In block N1, variable #1 is assigned the value 300. In block N2, variable #2 is assigned the value 200. In block N3, the loop with the precondition begins. The WHILE statement is followed by a conditional expression: «if the value of variable #1 is greater than the value of variable #2», if it is true, control is transferred to block N4. In block N4, the value of variable #1 is decreased by 10 (this happens at each iteration of the loop). In blocks N5, N6, N7, and N8, the tool moves along a square contour in relative positioning mode in the x-y plane. Block N9 ends the loop with the END statement. The DO...END loop is executed 10 times in a row. At the last iteration of the loop, the value of variable #1 is equal to 200, therefore, the loop condition is not fulfilled, and control is transferred to block N10, which ends the program execution with code M30.

The DO...END loop can be programmed without the WHILE precondition. You can use up to three nested loops, and you must follow the indexing order of the DO and END statements.

## CONTENT

1. General Description of the Software Product	1
2. Graphical User Interface Description	3
3. The Principle of Controlling a Virtual Camera Using a Mouse	15
4. Cutting Tool Trajectories Calculation Mode or Debug Mode	17
5. Mode for Determining the Position of Workpiece Working Coordinate Systems Using a Measuring Sensor	19
6. Mode for Determining Tool Offset and Diameter Using a Measuring Sensor	21
7. Brief Instructions for Working with the Milling Machine Simulator in the Mode of Connecting a Physical CNC Controller	23
8. Quick Reference for Milling Operations Programming in the Simulator	30
8.1. Linear Interpolation Commands G00 and G01	33
8.2. Circular Interpolation Commands G02 and G03	34
8.3. Chamfering and Rounding Programming	35
8.4. Program Execution Delay Command G04	36
8.5. Working Plane Selection Commands G17, G18, G19	37
8.6. Polar Coordinate Mode Control Commands G15, G16	37
8.7. System of Measures Selecting Commands G20, G21	38
8.8. Command to Return to the Reference Point G28	38
8.9. Command to Return from the Reference Point G29	40
8.10. Tool Radius Compensation Mode Control Commands G40, G41, G42	41
8.11. Tool Length Compensation Mode Control Commands G43, G44, G49	46
8.12. Scaling Mode Control Commands G50, G51	48
8.13. Local Coordinate System Setting Command G52	53
8.14. Transition to Machine Coordinate System Command G53	54

8.15. Commands for Selecting the Working Coordinate System G54-G59	57
8.16. Macro Program Calling Command G65	58
8.17. Rotation Mode Control Commands G68, G69	60
8.18. Positioning Mode Switching Commands G90, G91	62
8.19. Workpiece Coordinate System Transformation Command G92	63
8.20. Feed Mode Switching Commands G94, G95	64
8.21. Tool Return Mode Control Commands G98, G99	65
8.22. Fixed Cycle Cancel Command G80	65
8.23. Fixed Drilling Cycle G81	66
8.24. Fixed Cycle Drilling with a Delay G82	66
8.25. Fixed Cycles of Periodic Drilling G83 and G73	67
8.26. Fixed Cycles of Cutting the Thread with a Tap G84 and G74	68
8.27. Boring Fixed Cycles G76, G85, G86, G87, G88 and G89	68
8.28. Control Program Execution Pause Commands M00, M01	71
8.29. Control Program Completion Commands M02, M30	71
8.30. Spindle Rotation Control Commands M03, M04, M05	71
8.31. Tool Switch Command M06	72
8.32. Coolant Control Commands M07, M08, M09	73
8.33. Subroutine Control Commands M97, M98, M99	73
8.34. Parametric Programming	75
8.34.1. Working with Variables	75
8.34.2. Unconditional and Conditional Transitions	77
8.34.3. Loops with a Precondition	79