Creation of cross-platform toolkit and methods of programming trusted CNC systems

Liliya Martinova ¹ and Nikolay Fokin*¹

¹ Moscow State University of Technology STANKIN, Department of Computer Control Systems, 1 Vadkovsky, Moscow, 127055, Russia

Abstract. The developed cross-platform toolkit is presented and the methodology for its use for programming trusted CNC systems is described. The use of canned cycles in a trusted CNC system allows to quickly and flexibly change production processes, in particular when changing the range of products or if it is necessary to replace equipment with a changing the CNC system. Canned cycles built on cross-platform toolkit can be adapted and customized to meet new requirements and changes in the production process, which is critical to quickly respond to changing market conditions and ensure timely order fulfillment. The cross-platform toolkit presented in the article implement logical relationships between the variables that determine the geometry of the processed elements and the processing strategy embedded in canned cycles and logical control commands, which is aimed at ensuring the trust of the CNC system. The architectural model of the toolkit provides for the formation of universal templates for developing macroprograms in a high-level language and multi-level interaction of program modules with a gallery of macroprograms. The rules for structuring and arrangement of macroprograms in the CNC system are described.

1 Introduction

The CNC system, as one of the most important components of trusted production systems, must ensure the safety and reliability of the CNC machine tool[1-4], minimize the risk of errors and accidents, thereby increasing production efficiency [5, 6]. The development of canned cycles using the proposed toolkit and their subsequent use in accordance with prescribed rules in a trusted CNC system makes it possible to increase the efficiency of technological preparation and the safety of machine operation, to ensure more stable and predictable production results. This promotes uniformity and consistency in production, and also simplifies the training of new CNC machine operators, since the canned cycles contain optimal parameters and settings for each type of processing [7-8], which ultimately leads to increased trust of the control system.

The creation of canned cycles in a trusted CNC system must be a flexible and scalable process so that they can be customized and adapted to specific requirements and production conditions [9-12]. An existing alternative to the proposed approach - macroprogramming in

* Corresponding author: nikolfo@gmail.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).
high-level languages of the corresponding CNC systems - allows you to create custom cycles using the required set of program variables \([13, 14]\). Its main advantage is the ability to optimize the control code of the cycle for a specific CNC system; however, such a disadvantage as the complexity of programming, which requires highly qualified performers \([15]\), negatively affects the trust of the CNC system.

The developed cross-platform toolkit \([15, 16]\) implements logical relationships between the variables that determine the geometry of the processed elements of part and the processing strategy embedded in canned cycles and logical control commands. The architectural model of this toolkit provides for the formation of universal templates for developing macroprograms in a high-level language and multi-level interaction of program modules with a gallery of macroprograms \([17-19]\). The rules for structuring and arrangement of macroprograms in the CNC system are regulated.

## 2 Architecture of cross-platform toolkit for creating and debugging part programs

Formally, the concept of specialized toolkit is based on a 3-level architecture (Fig. 1), which includes: 1) user interface level, 2) independent software level, containing software that is independent of the specifics of a particular CNC system and the graphics features of the operating system, and 3) a level depending on the specifics of the CNC system \([18]\).

![Fig. 1. Architecture of toolkit for creating and debugging part programs.](image)

The user interface level contains graphical interfaces implemented on the basis of the Windows and Android operating systems that interact with software modules through a specialized library, a module for managing files and control program libraries, a control program template module, and a directory of post-processing modules. The level dependent on the CNC system is represented by a gallery of macro programs.

The module of party program template is used to build a part program in the form of a strict sequence of typical technological operations of machinings, as well as to edit the source data of of any of the. Post-processing modules generate part programs code for a specific CNC
system. The postprocessor carries out sequential calls of cycles from the gallery of macroprograms for each of the specific CNC systems (here implemented for the CNC systems: Fanuc, Siemens, AxiOMA Control).

Each macroprogram in the CNC system performs a specific block of functions and uses strictly defined global variables of the CNC systems that do not directly affect the operation of the main and auxiliary components of the technological equipment.

3 Methodology for programming trusted CNC systems using cross-platform toolkit

The programming technique is determined by the interaction of macroprograms and is based on the transfer of blocks of global variables to the final processing loop, where predefined global variables interact with local loop variables through mathematical, trigonometric and logical calculations. All calculations of the trajectories of movement of the machine's executive bodies are carried out within cycles using variables in the CNC system.

![Diagram of a programming technique]

**Fig. 2.** A programming technique based on the sequential interaction of macroprograms and positioning and processing cycles.

Specialized tools generate a sequence of calling macroprograms in a certain order for correct reading and sequential writing of global parameters in the CNC system. Fig. 2 presents a
programming technique based on the sequential interaction of macroprograms and positioning and processing cycles.

4 Interaction of setup macroprograms

Programming using the developed tools involves the interaction of macroprograms and preparatory subroutines with processing cycles by transferring the values of global variables to the final processing cycle, where they interact with local cycle variables through mathematical, trigonometric and logical calculations.

Processing cycles have common variables, which makes it possible to combine these variables into a specific set of auxiliaries, but mandatory macroprograms: a setup macroprogram, a setup macroprogram of a typical technological operation of machining, and a macroprogram containing information about M-codes.

The WORKSET setup macroprogram contains parameters that determine the position of the workpiece zero-point, safety planes, tool change point and maximum spindle speed.

The setup macroprogram of a typical technological operations of machining TOOLSET is responsible for the following actions:

- selection of coordinate system, zero point of processing, retraction of the tool to the change point;
- calling a tool by position number or corrector number;
- bringing the tool to the safety plane;
- turning on the spindle with a pre-selected frequency in a given direction;
- activation of working feed for subsequent cycles;
- activation of electrical automation devices.

The setup macroprogram of a typical technological operation of machining can be used in conjunction with a single processing cycle or with a sequence of cycles in the case of processing with one tool. Changing the variables in it is done by calling it again before each processing cycle with a different tool number, corrector or cutting mode from the previous one, followed by overwriting the local variables used in the cycle.

If machining routines require the use of one tool with a specific position and offset, then TOOLSET can be specified once. After completing the selected cycle, the tool will move to the safety plane rather than to the change point, which will optimize strokes and reduce machining time. This also applies to turning cycles - you can program the processing of several similar processing elements with different geometric configurations with one tool.

The positioning cycles block (POSITIONS) is designed to calculate the coordinates of each point of a given array. Positioning cycles (POSITIONS) are an intermediate link between the TOOLSET cycle and machining cycles (Cycles). They can be set either once or repeatedly for one processing cycle (Drilling, Milling) or for a group of cycles. The calculation of each coordinate occurs in the adjacent POSITIONSCalc cycle, called from each processing cycle. In POSITIONSCalc, depending on the selected machining plane, a calculated machining cycle is called, generally called MachiningCycleCalc. The rule of 3 nestings is observed, when up to 3 subroutines can be called sequentially from one control program.

A macroprogram containing information about M-codes is used to control auxiliary components of electrical automation: supply and stop of cutting fluid or air blowing, activation and stopping of the chip conveyor, opening and closing of the parts catcher, activation and deactivation of the spindle brake, switching the spindle to interpolated axis mode and back to spindle mode, rotating the driven tool on the turning and milling machine
clockwise, counterclockwise and stopping rotation. The file with M-codes, as well as the control program files for them, are saved automatically in the “General Documents” folder in OS Windows and the “Documents” folder in OS Android.

Positioning cycles are designed to calculate the coordinates of each point in a given array. Positioning cycles can be specified either once or multiple times for one processing cycle or for a group of cycles. Each coordinate is calculated in a separate macroprogram called from each processing cycle. In the calculation macroprogram, depending on the selected processing plane, the calculation processing cycle is called. The rule of 3 nestings is observed, when up to 3 subroutines can be called sequentially from one control program. The calculated coordinates along the axes are written into global variables <81> and <82> for the classically used planes XY (G17), XZ (G18), YZ (G19). Each array position is represented by 2 coordinates: the abscissa axis and the ordinate axis. Values global variables that specify the location of the center of the processed element <81> along the x-axis and <82> along the ordinate axis, as well as the initial angle from the x-axis <88> are recalculated in accordance with the array used and returned to the processing cycle for the correct construction of the processing path. Each coordinate of the cycle is carried out using a counter for the number of positions.

Plane rotation cycle. Cycles for milling standard elements (pockets, grooves, spigots, etc.) can work not only in predefined planes, but also in an arbitrary plane in space, which is important for processing inclined surfaces on vertical milling machines that have a specialized rigidly fixed rotary head on the spindle [4]. In this case, the part is based as in conventional 3-axis processing, and the tool axis corresponds to the axis of the normal vector to the plane being processed, and processing of ordinary flat elements occurs with the simultaneous use of 3 axes: X, Y, Z [2]. The rotation of the plane is carried out as follows: the base plane in the machine coordinate system is sequentially rotated around each of the axes, while the points of the plane have coordinates in the coordinate system of the plane being rotated. When turning, the coordinates of the points are recalculated from the plane coordinate system to the machine coordinate system (Fig. 3).

Fig. 3. Position of the Xi, Yi, Zi point in the machine coordinate system after rotating the plane around the abscissa-ordinate-applicate axes.

To calculate the plane transformation, preliminary calculations of the position of the point before rotation around the axes have already been performed when developing cycles on one of the predefined planes (XY, ZX, YZ) and are represented by the variables Xi', Yi', Zi'. The calculation assumes rotation around each axis in a certain sequence: 1 - abscissa, 2 - ordinate,
3 - applicate. For the calculation, the inverse rotation matrix in three-dimensional space is used, obtained by sequentially multiplying the inverse rotation matrices around the Oz, Oy, Ox axes.

5 Rules for recording macroprograms into memory

The rules for writing macroprograms into the memory of the CNC systems considered in the work are presented in Table 1. The editor's operating scenarios for correct editing of the control program in the CNC system are preliminarily specified: the variables of the WORKSET subroutine are written to the global variables of the machine before a new call to the WORKSET subroutine. The MCODES and TOOLSET routines work in a similar way. If processing is expected in the counter-spindle, then a repeated call of the WORKSET and MCODES subroutines is necessary to reconfigure the coordinate system, tool change point, safety planes and control of electrical automation devices.

Table 1. Rules for recording macroprograms into the memory of CNC systems

<table>
<thead>
<tr>
<th>Rule</th>
<th>Fanuc</th>
<th>Siemens</th>
<th>АксиОМА Контрол</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage area</td>
<td>Macro area O6000-O6300</td>
<td>Dir/Cycles/User cycles</td>
<td>NC/USERCYCLE (need to create a folder)</td>
</tr>
</tbody>
</table>
| Calling a macroprogram      | Modeless call command G65 P6000-P6300 A… B… Z… | Direct call from the control program Cycle (…, …,) | Direct call from the control program Cycle (…, …,) At the beginning of each program the operand #include "USERCYCLES/USERCYCLES.c"
| Global variables            | R1-R81                         | #101-#181                       | $dPerm [1]-$dPerm [81]              |

If machining routines require the use of one tool with a specific position and offset, then TOOLSET can be specified once. After completing the selected cycle, the tool will move to the safety plane rather than to the change point, which will optimize strokes and reduce machining time. This also applies to turning cycles - you can program the processing of several similar processing elements with different geometric configurations with one tool.

6 Conclusion

The work of the presented cross-platform toolkit for the preparation and editing of part programs is based on the methods and logic of the interaction of macroprograms, preparatory routines with canned cycles by transferring the values of global variables to the final processing cycle, where they interact with the local variables of the canned cycle through mathematical, trigonometric and logical calculations. The method of preparation of part programs using specialized tools is based on the rules of structuring and arrangement of macroprograms in the CNC system. The architectural model of the toolkit provides for the formation of universal templates for the development of macroprograms, which is aimed at ensuring the trustworthiness of the CNC system.
7 Acknowledgment

This work was supported by the Ministry of Science and Higher Education of the Russian Federation under project FSFS-2023-0004

References


19. G. M. Martinov, L.I. Martinova and N.N. Fokin, MATEC Web of Conferences 346(4), 03098 (2021) DOI: 10.1051/matecconf/202134603098