



Configuring of 3-axis vertical CNC Machine for Rapid Prototyping with two Translatory and one Rotary Axes

Nikola Vorkapić,^a Saša Živanović,^{a*} Nikola Slavković^a, Zoran Dimić,^b Branko Kokotović^a

^aUniversity of Belgrade, Faculty of Mechanical Engineering, Belgrade, Serbia

^bLOLA Institute, Belgrade

ABSTRACT

This paper describes the configuration of a 3-axis vertical CNC machine tool for rapid prototyping with one rotary and two translational axes. The machine works in a polar-cylindrical coordinate system. The structure of the machine is C'OXZ. The virtual machine model is configured in the PTC Creo software environment. After configuring the virtual machine, the simulation of the CLF-based was performed in the mentioned software environments, and then the verification according to the G-code program in the Vericut software environment was performed. Programming and control of the configured prototype machine are realized in the LinuxCNC software environment, which is based on the PC platform. Also, in this paper, digital twin of machine realized in a python software environment is shown. The presented results show the proper functioning of the whole system.

Key words: *CAD/CAM, CNC machine tool, virtual machine, LinuxCNC, digital twin;*

1. INTRODUCTION

Modern machine tools are very complex mechatronic systems. The capability and efficiency of a machine tool are mainly determined by its kinematics, computer numerical control system and the machining process [1]. Configuring a new machine tool is a process of functional completion on the basis of the conception of machine tools, using standard and special (calculated and made) components, kinematics and the open architecture system. For configuring 3-axis vertical CNC machines for rapid prototyping with two translatory and one rotary axes, the general methodology for configuring new machine tools is used [2].

The 3-axis vertical CNC machine with two translatory and one rotary axes, proposed in this paper, is a machine for rapid prototyping, which can be reconfigurable for various processes such as: drawing, laser engraving, 3D milling and 3D printing. In this phase of the research, the first variant of the machine that can draw contours on the

desktop was realized, which creates the basis for its further reconfiguration and application as a multifunctional machine tool. This machine has a specific kinematics, because the spatial Cartesian coordinates are represented in the polar cylindrical coordinate system in which the machine actually works.

The rapid prototyping is used in a variety of industries to describe a process for rapidly creating a part representation before final realization or commercialization [3]. Rapid prototyping (RP) is a technology for quick fabrication of physical models, functional prototypes and small series of parts directly from CAD models [4]. In its last phase, the considered machine will be reconfigured for material addition processes.

There are already many manufacturers of machines and equipment for rapid prototyping, and also there are a lot of different technologies for rapid prototyping [5]. There are four fundamental rapid prototyping processes [4, 6]: (i) subtractive, (ii) additive, (iii) combined (subtractive

* Corresponding author's.e-mail: szivanovic@mas.bg.ac.rs

and additive) and (iv) formative processes. The 3-axis vertical CNC milling machine with two translatory and one rotary axes, will be able to achieve, from the aspect of rapid prototyping, subtractive and additive processes, as well as drawing and laser engraving.

In order to improve the technical capabilities (accuracy and repeatability) of any RP machines benchmark geometry is essential. This means that the methodology for testing and design of benchmark artifact for evaluating the geometrical performance is very important [7]. To this end, a number of experiments were performed to draw different contours in order to verify the accuracy of the programmed contours as well as the developed control system.

The paper is organized as follows. Following this introductory section, the description of the mini polar CNC machine tools will be presented. The next section will describe the methodology of configuring the new machine. Section 3 describes programming and control system. Configured open architecture control system based on LinuxCNC, includes virtual machine in control system, i.e. digital twin. In the next section, the realization of the prototype and the test operation of the machine is shown. The concluding remarks are summarized in the last section.

2. MINI POLAR CNC MACHINE TOOLS

Mini CNC machine tools, which are realized with two translatory and one vertical rotary axis, are known on the market as Polar coaster machines or Polar Drawing machine. Among them, there are commercial solutions that originated from self-construction, as well as solutions that were made in self-construction. In terms of the number of actuated axes, they can be divided into two categories: 2-axis drawing machines and 3-axis drawing machines or 3D printing.

Figure 1 shows the models of polar coaster machines made in self-construction or which can be purchased as ready-made solutions on the market. The components of the supporting structure of the machine are made of plastic on a 3D printer. The connections between these components are made by a screw connection or with the help of threaded rods (Fig. 1e). Figures 1a, b, and e show the finished solutions of machines with two translational and one vertical rotary axis, which according to the number of controlled axes are realized as 2 ½ axes. These machines are realized as drawing machines and use a pencil as a tool. Moving along the X-axis allows taking

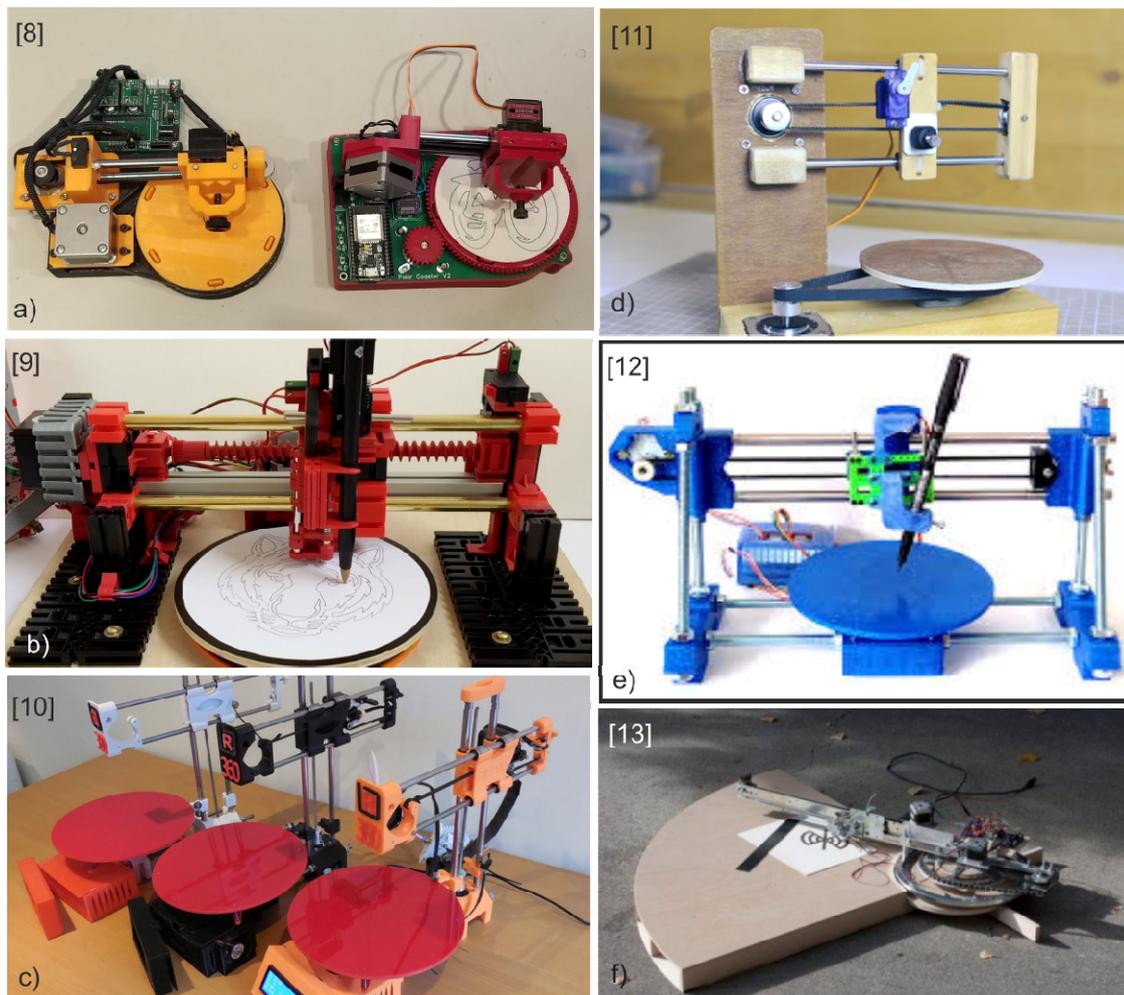


Fig. 1 Realized mini CNC machines with two translatory and one vertical rotary axis.

the appropriate stroke, while the rotating axis achieves the taking of an angle in the polar coordinate system. The X and C axes use NEMA 14 stepper motors, while a two-position servo motor is used for up and down movement of the pen. The translatory X axis moves with the help of a helical spindle (Figure 1b) or with the help of a toothed belt drive (Fig. 1a, c, d, e, f). The rotating axis moves with the help of a toothed belt drive or via a pair of gears (Fig. 1a). In the case of translatory axes, the guiding of the moving elements is realized through round roller guides, while in the case of the rotating axis, the guiding is performed through radial or axial roller bearings.

Figure 1c shows models of a machine with two translational and one vertical rotary axis which is realized as a 3-axis machine and can be used for contour drawing or 3D printing. Lifting or lowering along the Z-axis is achieved with the help of a spindle.

The machine in Figure 1f moves and is programmed in polar coordinates, with no rotating table, but the entire supporting structure of the X and Z axes rotates around a fixed point, better known as a pivot. The shown machine uses only a quarter of a circle as a workspace, i.e. it works in only one quadrant. The machine models shown in Figure 1 are controlled by Arduino control system.

The realized machine, Polar_CNC, shown in this paper is marked as C'OXZ according to the kinematic structure. The machine is 3-axis due to the possibility of reconfiguring the tool to perform various tasks. The control system is open architecture and is realized by LinuxCNC. For the realization of the considered machine

in this paper, the methodology for configuring new machine tools presented in the next chapter was applied.

3. THE METHODOLOGY OF THE CONFIGURATION OF A NEW MACHINE TOOL

Functional model for configuring the new machine tool is a structural presentation of function, activity or process modeling, which can be viewed at several different levels. The result of applying IDEF0 methodology to configure the new machine tool is a model which consists of a set of hierarchical diagrams with text descriptions [2]. One application of this methodology is shown in this paper. Figure 2 shows the basic flow of activities for configuring new 3-axis vertical CNC machine with two translatory and one rotary axes. Top level child diagram A0 describes a basic flow of activities and it is illustrated in Fig. 2. The basic flow of activities is: A1. Design of machine and configuring of the virtual prototype; A2. Configuring of control system; A3. Analysis and simulation on virtual prototype and postprocessing and A4. Realization of the machine tool and test work.

Concept of new machine tool is input for activity A1, where prepared project of machine tool and his digital representation/virtual prototype. Simultaneously with activity A1, activity A2 takes place that include, configuring of control system and also validation of control system on configured virtual prototype as a digital twin.

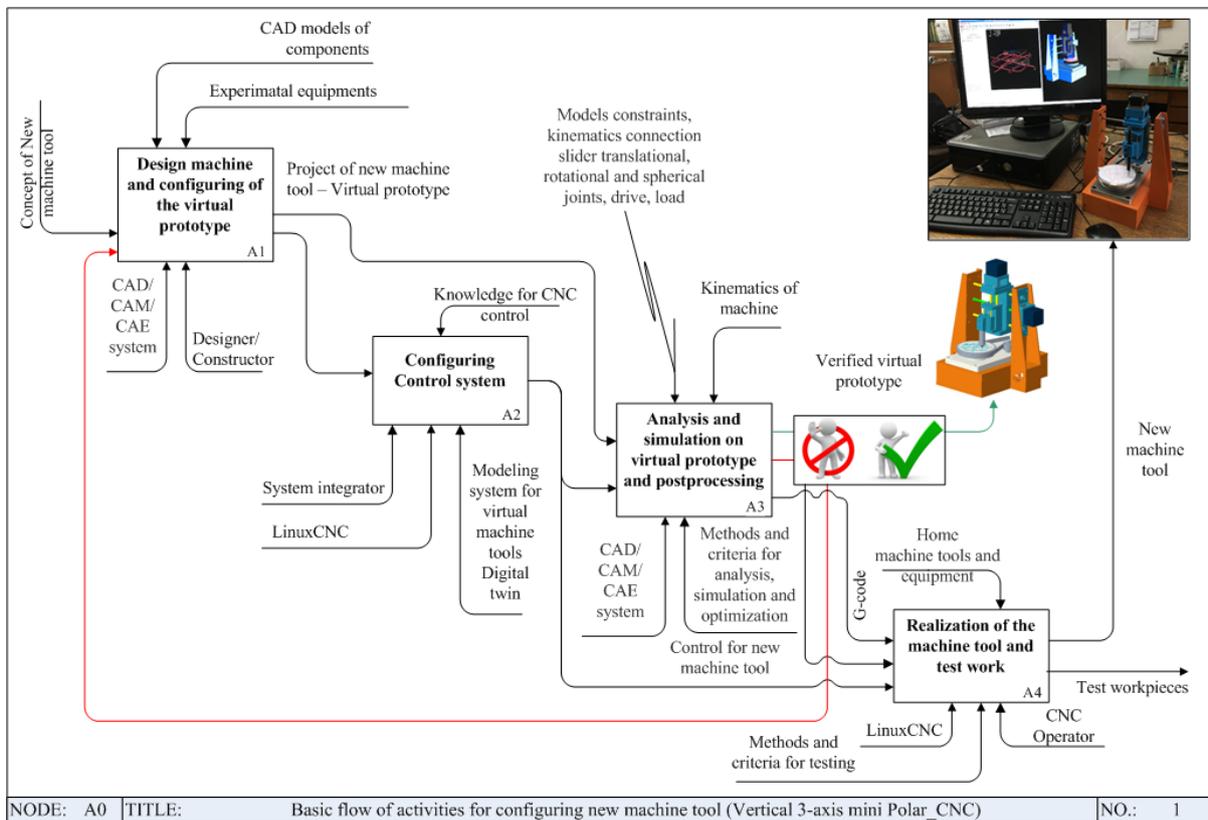


Fig. 2 The basic flow of activities for configuring new machine tool

In Activity A3, all possible analyzes and simulations are performed on a virtual prototype in order to obtain a verified virtual prototype ready for implementation. In this activity (A3), an appropriate postprocessor for the machine is being developed, which enables the translation of the G code from Cartesian to the polar coordinate system. After this, in activity A4 the considered machine is realized and tested. The transition from digital to the real world takes place in activity A4, where the realization and integration of the hardware and software part of machine tool and testing is performed. Some of the elements of the listed activities will be shown in more detail below.

3. PROGRAMMING AND CONTROL SYSTEM

The available CAD/CAM environment is used as a programming system. For the purposes of this work, CAD/CAM systems PTC Creo [14] were used, while for verification of programs in G code the Vericut [15] environment is used. An open-architecture system on a PC platform and a LinuxCNC are used for control.

3.1 Programming system

Programming using CAD/CAM systems is common. Since the machine is used to draw programmed contours, the CAD model can be a 2D contour or a selected contour on a 3D model. Contouring strategies without tool radius correction are used for programming. Program verification is possible by simulating of the tool path, as well as by simulating the operation on a virtual machine running according to a given program (CLF and/or G code). Figure 3 shows the methodology for programming of considered 3-axis machine. After generating the 2D contour, within the software package PTC Creo, programming was performed using a strategy for processing along a given contour without correcting the radius of the tool. By generating a CL file, it is possible to simulate the tool path and simulate the operation of a virtual machine according to a given program.

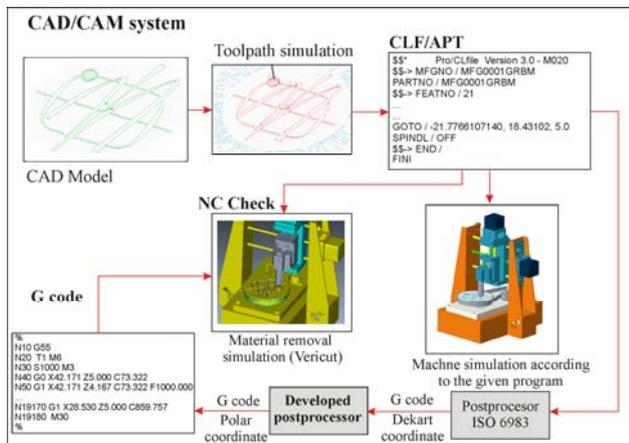


Fig. 3 Basic structure of programming system

After verification of the tool path according to the given program, post-processing is performed for any control unit that contains a post-processor for 3-axis processing in Cartesian coordinates. The generated G code according to the ISO 6983 standard needs to be additionally transform into the appropriate polar coordinates. This additional postprocessor has been developed, according to the methodology shown in Chapter 3.1.2. After that, it is again possible to perform program verification within the Vericut software environment, which verifies the developed postprocessor that converts Cartesian coordinates into polar coordinates.

3.1.1 Virtual machine in programming system PTC Creo

To check the kinematics of the displayed machine, its kinematic simulation is performed within the software package PTC Creo. The configured virtual machine model is shown in Fig. 4.

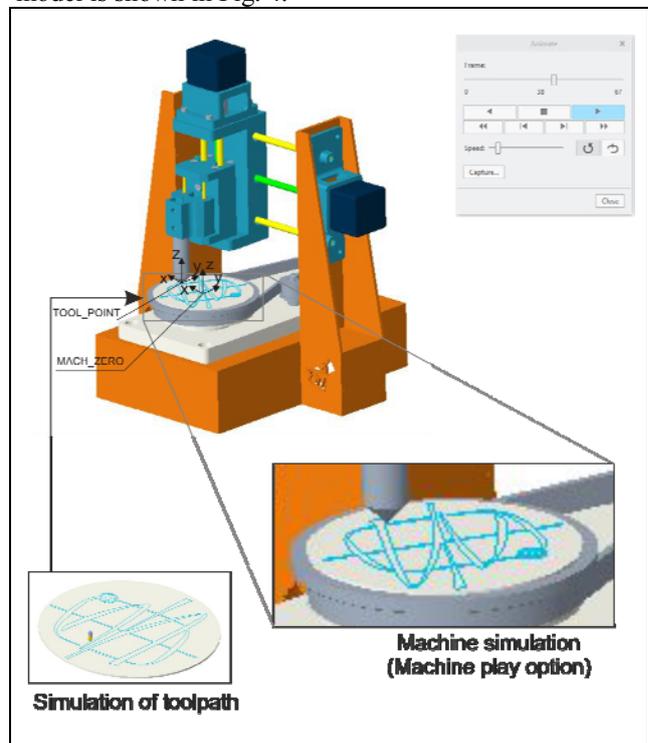


Fig. 4 Machine simulation of the toolpath

For the purposes of simulating the operation of a virtual machine according to a given program, it is previously necessary to define the kinematic connections between the moving elements on the machine. Since the model of the machine with two translatory and one rotary axis is shown, the slider (for translational axes) and pin (for rotary axis) kinematic connections will be used. Virtual machine operation cannot be started until the tool and workpiece coordinate systems on the machine match the coordinate systems of the virtual workpiece model and the virtual tool model. These two coordinate systems are MACH_ZERO and TOOL_POINT, which is described in detail in the paper [16].

3.1.2 Postprocessing

As it is said, the programming of Polar_CNC machine tools starts with generation of G-code, according to ISO 6983 standard in Cartesian coordinates. In order to run generated program on Polar_CNC it is necessary to postprocess (transform) such program to polar coordinates. This subsection describes developed postprocessor for transforming the program in Cartesian to polar (C2P) coordinates, Fig. 5. Developed postprocessor is a program generated in Python programming language.

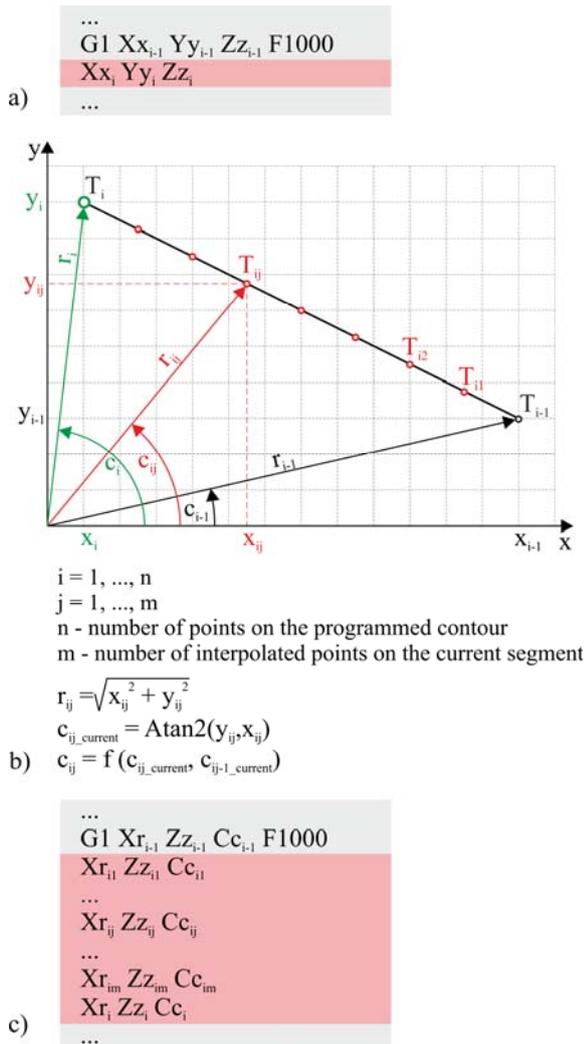


Fig. 5 Cartesian to polar coordinates transformation

Program reads one by one G-code line (in Cartesian coordinates) and checks if this line describes moving command. If this is not case the line is writing in new G-code program (in polar coordinates). If i -th line, Fig. 5a describes moving of tool the C2P transformation will be performed.

First, with Cartesian coordinates loading from this line and coordinates extracted from $i-1$ line the i -th linear or circular segment is completed describing by points T_{i-1} and T_i . In order to describe i -th segment in polar coordinates it has to be divided according to linear or

circular interpolation in Cartesian space with points T_{ij} , Fig. 5b). Now, each Cartesian points T_{ij} (x_{ij} , y_{ij} , z_{ij}) are transformed in appropriate points in polar coordinates described as (r_{ij} , z_{ij} , c_{ij}) according to equation present on Fig. 5b). It has to be mentioned that value of angle C of point T_{ij} , that will be written in appropriate sentence, depends on the value of current C angle, calculated for point T_{ij} , and the value of C angle that belong to previous interpolated point. Function for calculation of appropriate values of C angle is developed in order to avoid axis rewinding.

All of these points defined in polar coordinates are used to generate the several line of new G-code program (in polar coordinates), Fig. 5c). After last point is transform to sentence of new program, the $i+1$ line of G-code (in Cartesian coordinates) is load and the described procedure is repeated.

3.1.3 Virtual machine in Vericut

Verification of programs obtained by post-processing from the previous chapter 3.1.2 was realized in the Vericut software environment [15]. For complete verification, the possibility of configuring a virtual machine tool in the Vericut environment was used, which enables simulation of machine operation based on G code, with material removal simulation. For the example of the considered Polar_CNC machine, with structure C'OXZ, its virtual machine is configured.

The kinematic structure of the considered Polar_CNC machine tool C'OXZ describes the arrangement and type of connections between the base and the moving components of the machine. It is necessary to define the manner and order of binding of base components, tool branches and preparation branches, i.e. BASE, TOOL and STOCK. The fixed part of the machine is the base, to which the components of the tool branch (X and Z) and the components of the workpiece branch (C') should be added, Fig. 6.

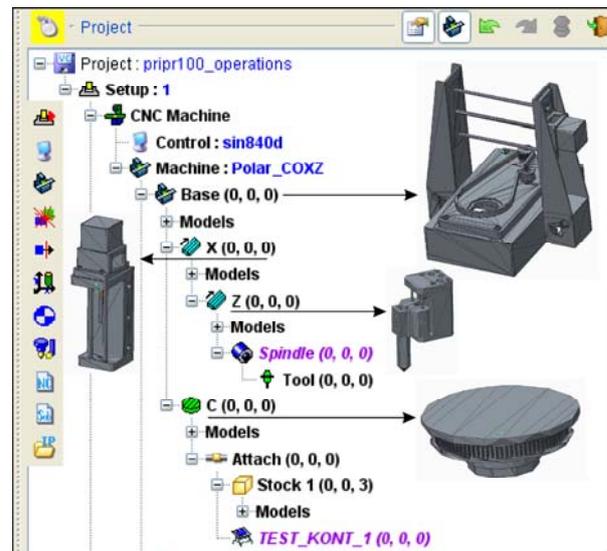


Fig. 6 The hierarchical tree structure of the C'OXZ machine in Vericut

It is characteristic of this machine that there is no movement along the Y axis. The configuring of the virtual machine in Vericut, starts from the base as the first fixed component, to which the rotary axis C' is added, as part of the workpiece branch. This axis represents the rotating workbench on which the attach workpiece (Stock) is placed and based. On the other hand, the fixed base is connected to the remaining two axes X and Z. The horizontal translational axis X (X Linear) was first added to the base, on which the vertical translational axis Z (Z Linear) was added. The main spindle (Spindle) and the tool (Tool) are connected to the Z axis, which completes the kinematic structure of the machine. After configuring the virtual machine, it should be saved and added to the database of existing virtual machine tools in Vericut.

For the realization of the processing simulation project it is necessary to define [16,17]: (i) workpiece (Stock), (ii) working object, coordinate system (Program zero point), (iii) zero-point position adjustment on the virtual machine), (iv) tools used in process and (v) NC programs. An example of one performed process simulation is shown in Fig. 7.

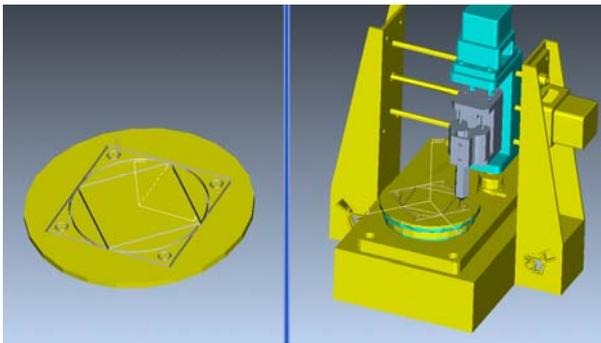


Fig. 7 Program verification on a virtual machine in Vericut

3.2 Control system

The open architecture system LinuxCNC [18, 19] was chosen as the control system, which is a real-time software for machine tools and robots, whose code can be freely used, modified, and distributed (GNU-General Public License). LinuxCNC enables programming of machines using programs in G code according to the RS274 standard, i.e. ISO 6983. In configuring of control system based on LinuxCNC (EMC2), the experiences from previous works were applied [16, 20].

The internal software structure of LinuxCNC is described in detail in [19, 20] and includes four basic program modules [20]: motion controller (EMCMOT), controller of discrete I / O (input / output) signals (EMCIO), process controller that coordinates them (EMCTASK) and a collection of textual or graphical user interfaces (GUIs), Fig.8.

The Axis GUI was chosen as the graphical user interface, which has a very intuitive environment for work and facilitates the work of the operator. In addition, the advantage of the Axis GUI environment is the possibility of integration with a virtual machine, which is shown in the next subsection.

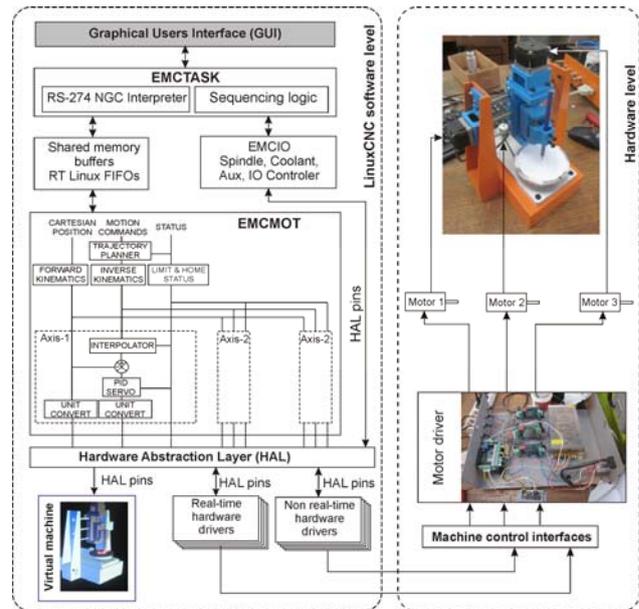


Fig. 8 Basic structure of control system (LinuxCNC)

3.2.1 Virtual machine in control system - Digital twin

For the purposes of implementation and testing of control system before implementation on the actual machine, a virtual machine is configured as a digital twin, which is integrated with the LinuxCNC system.

Virtual machine can integrate into an open architecture control system thus allowing machining simulation in OpenGL 3D environment. Such virtual machine enables program verification in G-code that includes the following: (i) visual detection of collisions; (ii) check if the machine can execute the specified tool path within the limited motions ranges and speeds.

To configure the virtual machine, the ability to load complete machine subassemblies was used, similar to configuring the machine in the Vericut software. The components need to be prepared in Ascii STL or Ascii OBJ format, which Python can load directly into the reference coordinate system, after which the component have to be oriented and appropriate placed within the virtual machine.

Afterwards, imported components are properly oriented and placed in the virtual environment, resulting in fully functional virtual machine shown in Fig.9. The virtual machine is placed in a separate window and allows toolpath drawing during movements of the virtual machine elements. During the execution of the machining program written in G code, virtual machine elements are moving in real-time, fully synchronized with moving elements of the real machine. Figure 9 shows the Digital twin of the machine integrated with the Axis GUI, on the example of drawing the contour of the coat of arms of the Republic of Serbia.

The digital twin is used during configuring control and also used for program verification and validation of the developed controller. During the control system development and configuring, the digital twin was made

at first as an exact copy of the real machine in kinematics meaning, before realization of real machine.

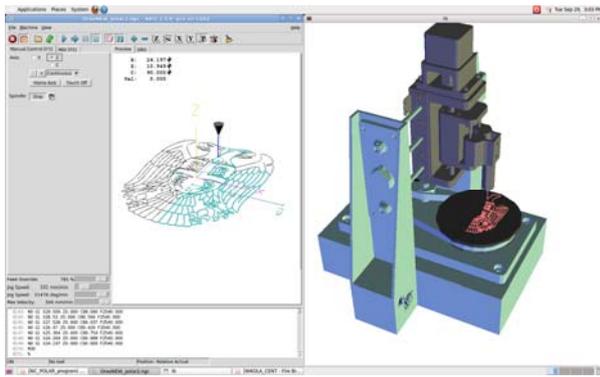


Fig. 9 Axis GUI of control system LinuxCNC and integrated virtual machine as digital twin

4. THE REALIZATION OF THE PROTOTYPE AND THE TEST OPERATION OF THE MACHINE

In Figure 10b a realized prototype of the Polar_CNC machine is shown. The machine consists of a massive supporting structure (Fig. 10a), which is made of steel sheet 3 mm thick. The sheets are joined by a welding process. In Figure 10c the X-axis module is shown. It consists of brackets made by 3D printing, made of PLA material. The brackets are connected by cylindrical roller guides and a helical spindle for X-axis drive. The X-axis module, together with the Z-axis assembly, is shown in Figure 10d. This module is also made by a combination of 3D printed parts and cylindrical roller guides together with a helical spindle and a stepper motor. The parts of the C 'axis are completely made of aluminum, which can be seen in Figure 10e. To increase the engine torque of the C 'axis, a belt drive with a gear ratio of 3.75 was constructed.

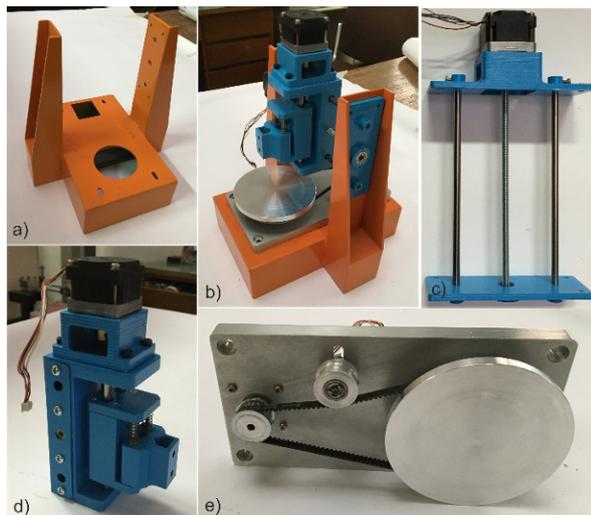


Fig. 10 Realization of a mini Polar_CNC 3-axis machine with two translational and one vertical rotary axis (a) base structure; (b) Polar_CNC assembly; (c) X-axis propulsion system; (d) X-axis slider and Z-axis slider drive assembly; (e) C-axis drive assembly with rotary table

The motors used to drive the axes of the machine are NEMA 14 stepper motors. The realized prototype of the machine is a 3-axis machine tool that can, in addition to drawing contours, also perform the processes of laser engraving, 3D printing and milling of soft foam materials (Styrofoam). After assembling the machine hardware and configuring the machine control system, a test of the machine is followed. The original considered type of machine was realized as a 2D printer, where the third Z axis is positional. During the test operation of the machine, the drawing of several different test contours was examined, which together with the environment of the machine are shown in Figure 11.

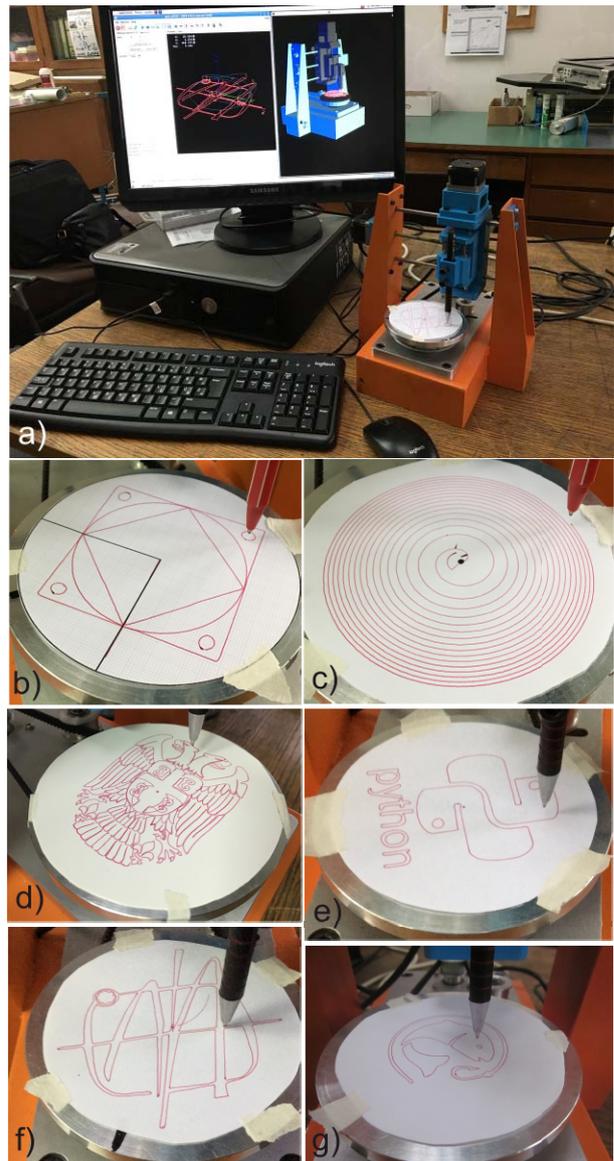


Fig. 11 Machine work environment and results of several trials

In Figure 11a the working environment of the machine is shown, while the drawn test parts are shown in Figures 11 b, c, d, e, f, g. The part shown in Figure 11b represents a contour resembling a test piece for testing NAS 979 machine tools. The drawn contour was used as a demonstration example to check the working accuracy of

machines. The errors of 0.5 mm that were obtained during the measurement, after drawing, indicate the gaps that are located in the tool itself on the machine. For this reason, the obtained drawing cannot be used to check the working accuracy of the machine. The second drawing is shown in Figure 11c represents the Fermat spiral or parabolic spiral which represents is a plane curve named after Pierre de Fermat. Polar coordinate representation for Fermat's spiral is given by Eq. (1):

$$r = \pm a\sqrt{\theta}, \quad \theta > 0 \quad (1)$$

This spiral fits into the boundaries of the working space of the machine, with a suitable choice of parameters. The contour of the Fermat spiral was generated using the CAD/CAM system PTC Creo, which was also used to obtain an adequate program in G code in the Cartesian coordinate system. This code was post-processed into polar coordinates and then plotted on a Polar_CNC machine. The parametric description of the curve is given below:

```
Fermat spiral for THETA>0
a=0.5
THETA=t*25*360
r=SQRT((a^2)*THETA)
z=0
```

(2)

The previously described contours represent regular geometric shapes. Figures 11 d, e, f, g, show more complex contours, which are also programmed in CAD/CAM system PTC Creo using complex planar vector contours drawn on a developed Polar_CNC machine as part of a test run.

5. CONCLUSIONS

The machine presented in this paper enables the development of a system for control and programming of machines with two translational and one rotary axis. Within this work, a postprocessor for transformation from Cartesian to polar coordinates was developed, which was verified by the work of the presented machine.

Test runs allow us to conclude the following: (i) no visible differences in the appearance of the drawn and programmed contours confirm that the realized concept is good, (ii) positioning accuracy within the boundaries of the machine workspace was good, (iii) developed postprocessor from Cartesian to polar coordinate G code give correct results during machine work, (iv) the simultaneous work of virtual and real machine was successful.

Further activities include adaptation of the mini CNC machine tool, so that it can carry a small laser, designated for laser engraving, and main spindle for milling, as well as construction and development of other concepts regarding mini CNC machine tools.

ACKNOWLEDGEMENTS

The presented research was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia by contract no. 451-03-68/2020-14/200105.

REFERENCES

- [1] Altintas, Y., Brecher, M., Weck, M. and Witt, S. (2005). Virtual machine tool. *CIRP Annals – Manufacturing Technology*, 54(2), 115-138, DOI:
- [2] Zivanovic, S., Glavonjic, M., Dimic, Z. (2009). Methodology for Configuring Desktop 3-axis Parallel Kinematic Machine. *FME Transactions*, 37, 107-115
- [3] Gibson, I., Rosen, D. and Stucker, B. (2010). *Additive Manufacturing Technologies, 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. 2nd Edition, Springer New York Heidelberg Dordrecht London.
- [4] Zivanovic S., Popovic, M., Vorkapic, N., Pjevic, M., Slavkovic N. (2020). An Overview of Rapid Prototyping Technologies using Subtractive, Additive and Formative Processes. *FME Transactions*, 48(1), 246-253, DOI:10.5937/fmet2001246Z
- [5] Zivanovic, S., Tabakovic, S., Randjelovic S. (2019). Rapid Prototyping of Art Sculptural Shapes According to the Sample. *Advanced Technologies and Materials*, 44(1), 27-32, DOI: 10.24867/ATM-2019-1-02
- [6] Chua, C.K., Leong, K.F. and Lim, C.S. (2003). *Rapid Prototyping: Principles and Applications*. 2nd Edition, World Scientific Publishing Co. Pte. Ltd.
- [7] Vorkapic, N., Pjevic, M., Popovic, M., Slavkovic, N., Zivanovic, S. (2020). An additive manufacturing benchmark artifact and deviation measurement method. *Journal of Mechanical Science and Technology*, 34(7), 3015-3026, DOI:10.1007/s12206-020-06 –y
- [8] Buildlog.Net Blog, Polar Coaster Version 2, from <https://www.buildlog.net/blog/2018/12/polar-coaster-version-2/>, accessed on 2020-10-20.
- [9] Polar Drawing Machine-Arduino Project Hub, from <https://create.arduino.cc/projecthub/ArduinoFT/polar-drawing-machine-f7a05c/>, accessed on 2020-10-20.
- [10] Microfabricator, from <http://microfabricator.com/articles/view/id/528781e89aad9d535f000001/r-360-simple-open-source-polar-3d-printer>, accessed on 2020-10-20.
- [11] How to make polar CNC drawing machine, from https://www.youtube.com/watch?v=dI_XWb1AK2U&ab_channel=MrInnovative, accessed on 2020-10-20.
- [12] Polar Pen Drawing Machine Plotter Arduino DIY Kit, from <http://www.3dmaxprinter.com/shop/do-it-yourself/polar-pen-drawing-machine-plotter-arduino-diy-kit/>, accessed on 2020-10-20.
- [13] Center pivot pen plotter, from <https://www.ponoko.com/blog/digital->

- manufacturing/center-pivot-pen-plotter/, accessed on 2020-10-20.
- [14] PTC Creo, from, <https://www.ptc.com>, accessed on 2020-10-10.
- [15] CGTech, Vericut, from, <https://www.cgtech.com/>, accessed on 2020-10-10.
- [16] Živanović, S., Vorkapić N., Dimić, Z. (2019). Konfigurisanje sistema za programiranje i upravljanje 3-osne mini CNC mašine alatke na Raspberry Pi platformi, *TEHNIKA: Časopis saveza inženjera i tehničara Srbije*, 6, 823-831, DOI: 10.5937/tehnika1906823Z
- [17] Zeljković, M., Tabaković, S., Živković, A., Živanović, S., Mladenović, C., Knežev, M. (2018). Osnove CAD/CAE/CAM Tehnologija, ISBN 978-86-6022-120-1, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 2018.
- [18] Linux CNC, Enhanced Machine Control - EMC2, from <http://www.linuxcnc.org/>, accessed on 2020-10-20.
- [19] Staroveški, T., Brezak, D., Udiljak, T. (2013). LINUXCNC – the Enhanced Machine Controller: application and an overview, *Tehnički vjesnik/Technical Gazette*, 20(6), 1103-1110.
- [20] Živanović, S., Dimić, Z. (2019). Virtuelna petosna mašina alatka integrisana sa sistemom programiranja i upravljanja, *TEHNIKA: Časopis saveza inženjera i tehničara Srbije* 3, 397-404.