

Method for Configuring Virtual Robot as an Integral Part of the Control System

Nikola Slavković, Saša Živanović, Zoran Dimić, and Nikola Vorkapić

Abstract—The development of integrated computing environments provides opportunities for the development of virtual production. Virtual simulation is crucial when the robot performs tasks that include some manufacturing processes. Virtual robots are used for program verification before sending it to the real robot and enable collision checking between robot segments themselves and the robot and its environment. Virtual models of industrial robots could be configured in different environments and ways. This paper presents the method for configuring virtual robots as an integral part of the control system. The virtual robot's configuration is realized under the LinuxCNC software environment and relies on OpenGL and several interface classes written in Python programming language. Developing a robot kinematic model to implement a virtual robot integrated with an open-architecture control system is necessary. Models of robot segments were imported in ASCII STL format and connected according to the robot kinematic model, and then the virtual robot was integrated within the LinuxCNC control system. The method for configuring a virtual robot as well as its kinematic model is presented in the example of the BiSCARA robot. Verifying the robot control system, virtual model, and kinematic model has been performed through several examples of drawing contours on the configured virtual robot.

Index Terms—Virtual robots; Control system; Robot simulations; Kinematic modeling;

I. INTRODUCTION

Sudden changes in production programs, product short life cycle, as well as further progress in the development of integrated computing environments for the development of new products provide opportunities to achieve the paradigm of virtual production [1]. The use of industrial robots in manufacturing and assembly lines is continuously increasing due to the need for high efficiencies, high accuracy, high production rates, and repeatability [2]. The use of robots for pick and place, welding, subtractive and additive manufacturing has been spreading in the last years with the concept of industry 4.0 [3]. In addition to performing these tasks, the application of industrial robots has been extended to laser engraving, laser and plasma cutting, robot milling, etc.

Associate prof. Dr. Nikola Slavković is with the Faculty of Mechanical Engineering, University of Belgrade, 16 Kraljice Marije, 11120 Belgrade, Serbia (e-mail: nslavkovic@mas.bg.ac.rs).

Full prof. Dr. Saša Živanović is with the Faculty of Mechanical Engineering, University of Belgrade, 16 Kraljice Marije, 11120 Belgrade, Serbia (e-mail: szivanovic@mas.bg.ac.rs).

Research Associate Dr. Zoran Dimić is with the LOLA Institute, 70A Kneza Višeslava, 11030 Belgrade, Serbia (e-mail: zoran.dimic@li.rs).

Nikola Vorkapić, MSc (ME) is with the Faculty of Mechanical Engineering, University of Belgrade, 16 Kraljice Marije, 11000 Belgrade, Serbia (e-mail: nvorkapic@mas.bg.ac.rs).

Designing and testing a robotic cell is one of the essential modern manufacturing engineering tasks, especially when the robot does not just handle materials but also performs manufacturing processes [4]. To save time and verify the robot cell for performing these tasks before its production, virtual environments, i.e., virtual robots have a key role.

The notion of a virtual industrial robot is broad and includes complete models of robot structure, kinematic subsystem, tasks process, the environment, etc. All these models are integrated into a single software system, enabling some part of the virtual production.

Virtual industrial robots, that are included in the production engineering environment, could be configured in the different environments such as CAD/CAM systems, MatLab/Simulink environment, specialized CAM software for robot programming (RobotStudio, Robotmaster, ...), LinuxCNC as an integral part of robot control systems, virtual reality system, etc.

This paper presents the configured virtual robots in different software environments using a five-bar, i.e., dual SCARA or BiSCARA robot. Robots based on the five-bar mechanism are widespread in academic institutions because they are suitable for experimental work. This base mechanism is almost always improved with the additional translatory and rotary axis to develop 3- or 4-axis robots. The developed virtual robot can simulate robot tasks such as laser engraving, 3-axis milling, 3D printing, etc. The first such robot, Fig. 1, is described in a US patent in 1934 [5, 6]. In 1978, Prof. Hiroshi Makino invented the well-known SCARA robot [7], while in 1985, Donald C. Fyler came up with the idea of using a five-bar mechanism as a robot [5, 8].

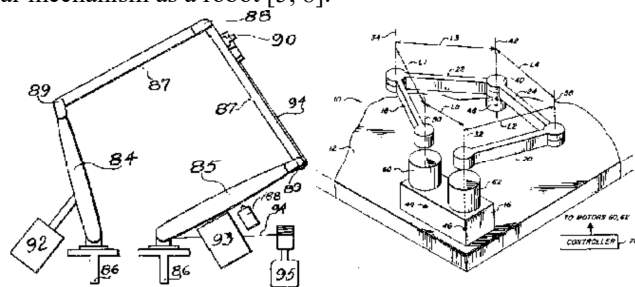


Fig. 1. Concept of five-bar, i.e., dual SCARA or BiSCARA, robot [1]

Mitsubishi Electric was the first company that commercialized the robot, MELFA RP-1A, based on this mechanism [5]. An example of a developed industrial prototype is the DexTAR robot (Dextrous Twin-Arm Robot) [9].

II. VIRTUAL INDUSTRIAL ROBOTS

A virtual industrial robot is a digital description of a robot, usually with simplified geometry, and is used for computer simulations of robot tasks.

To configure a virtual robot in a CAD environment [10], first, all elements of the robot structure are generated. These elements are then connected to the kinematic structure of a real robot. To be able to model a robot used in the simulation, one member of the robot mechanism must be fixed, and the moving components of the robot should be connected with appropriate kinematic connections (pin and/or slider). After this, in the next step, it is necessary to connect the coordinate systems of the workpiece and end-effector (EE) with the robot's coordinate system. In CAD/CAM environment, PTC Creo, the robot's coordinate system is defined as MACH_ZERO, while TOOL_POINT presents the end-effector's coordinate system. Coordinate systems with the same names have both an EE and a workpiece. The virtual EE is placed on the virtual robot by matching these coordinate systems. After the successful connection of the virtual workpiece and EE, a virtual industrial robot simulation according to a given program can be run. The model of the BiSCARA robot configured in the PTC Creo environment is shown in Fig. 2. The simulation of movement of the virtual robot tooltip is shown in the example of drawing a circle and a rectangle inscribed in a determined workspace of the robot.

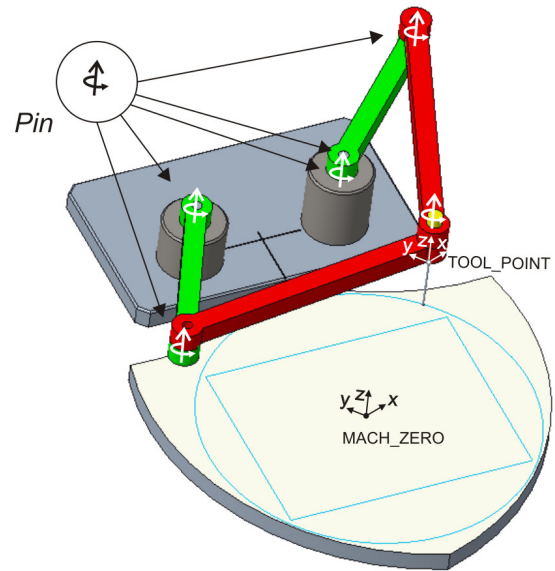


Fig. 2. Configuring of virtual robot in CAD/CAM environment

As described in the previous work [11], configuring the virtual robot in the MatLab/Simulink environment considers the development of the program for the joint space trajectory generation in the MatLab environment and configuring of the virtual robot in the Simulink environment. The structure of configured virtual robot in Simulink is presented in Fig. 3a.

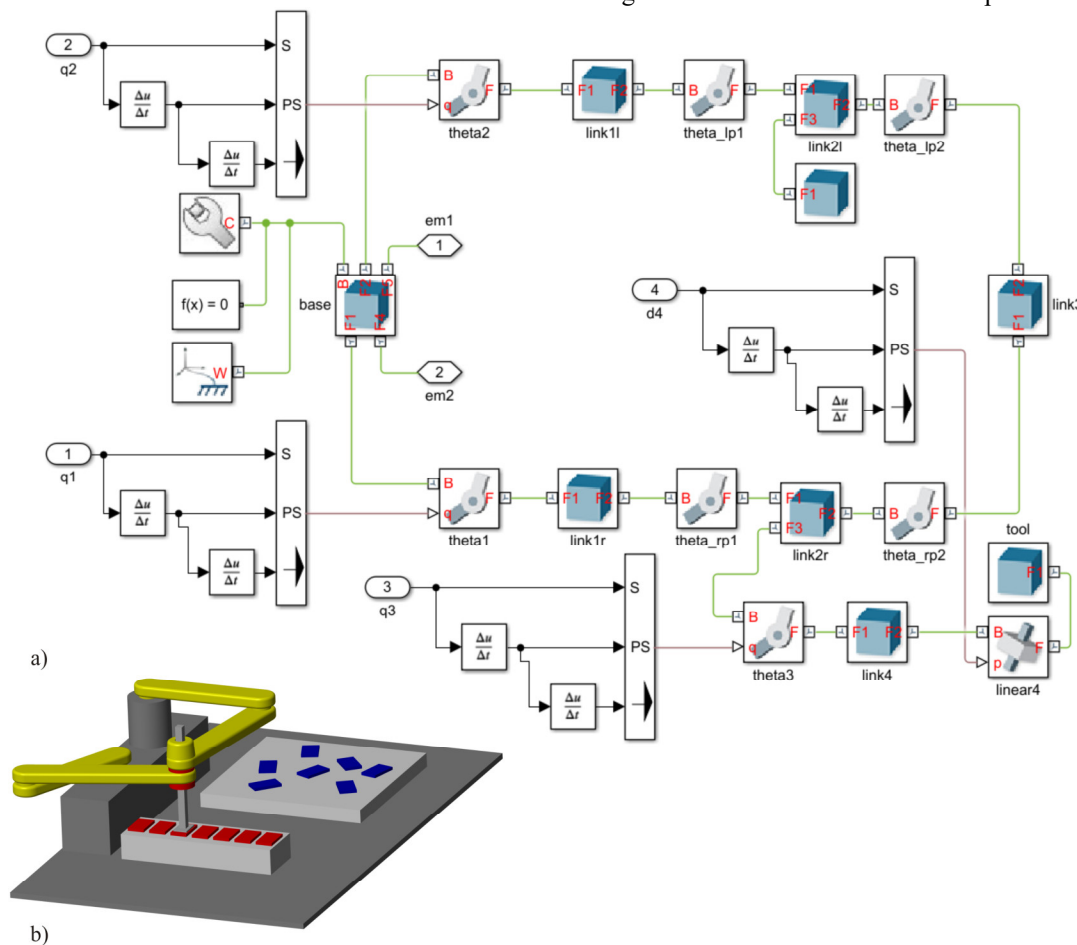


Fig. 3. Configuring of virtual robot in MatLab/Simulink environment

It consists of a base and three kinematic chains. The base is represented with the Brick Solid element connected with World Frame, Solver Configuration, and Mechanism Configuration elements. Any kinematic chain consists of Brick Solid elements and appropriate Revolute Joint and/or Prismatic Joint elements. The active joints in these kinematics chains have to be actuated by joint time series generated in MatLab consisting of joint angle, velocity, and acceleration [11]. The model of the BiSCARA robot configured in MatLab/Simulink environment is shown in Fig. 3b.

Virtual robots, as well as virtual machines, have a significant role when developing a control system where it is necessary to integrate solutions of inverse and direct kinematics. In this way, it is possible to develop a control system even before the realized physical prototype, which is of great importance because it allows virtual testing during development, known as virtual commissioning [12, 13, 14].

This paper considers the method for configuring virtual robots in the LinuxCNC control system. Virtual robots in the control system are significant during the robot exploitation and the configuring of the control and before the realization of the robot. Program verification using virtual robots is significant because it enables:

- testing the kinematic model during the development of the control system,
- visual detection of collisions between the moving parts of the robot and between the tool, workpieces, and fixtures in the workspace,
- checking if the robot can execute the specified toolpath within the limited workspace, joints ranges, and speeds,
- training and education for robot programming, etc.

III. KINEMATIC MODEL OF ROBOT

To realize the virtual robot as an integral part of the control system, it is necessary to perform a kinematic analysis of the considered robot [11, 15]. A Parallel BiSCARA robot can be viewed as a planar manipulator with two degrees of freedom. The kinematic analysis embraced solving direct and inverse kinematic problems, determination of Jacobian matrix, and workspace analysis [11]. The kinematic model of the BiSCARA robot is shown in Fig. 4.

The robot consists of a base, a platform, and two kinematic chains with struts lengths l_1 and l_2 . All mechanism elements are connected by a joint with one rotary degree of freedom. The frame $\{B\}$ represents the base frame, while the platform is represented by point P because the struts of length l_2 are connected at the point P.

As it can be seen from Fig. 4, the world coordinate vector is defined as

$${}^B \mathbf{p}_P = [x_P \quad y_P]^T \quad (1)$$

while joint coordinate vector is represented as

$$\boldsymbol{\theta} = [\theta_1 \quad \theta_2]^T \quad (2)$$

Besides another unit vector, the unit vector ${}^B \mathbf{a}_i$ is defined as

$${}^B \mathbf{a}_i = \begin{bmatrix} \cos(\theta_i) \cos(\gamma_i) \\ \sin(\theta_i) \end{bmatrix} \quad (3)$$

where $i=1, 2$ represents the number of the kinematic chain and γ_i represents the angle that defines the arrangement of kinematic chains and is introduced in order to generalize the solution of the inverse kinematic problem and parallel determination of joint coordinates during the solving an inverse kinematic problem.

Observing one kinematic chain, the following vector equations can be derived

$$\begin{aligned} {}^B \mathbf{p}_P &= {}^B \mathbf{b}_i + k_i \cdot {}^B \mathbf{w}_i \\ k_i \cdot {}^B \mathbf{w}_i &= l_1 \cdot {}^B \mathbf{a}_i + l_2 \cdot {}^B \mathbf{z}_i \end{aligned} \quad (4)$$

from which inverse and direct kinematic problems can be solved in analytic form.

The authors' previous work represents the complete solution of inverse and direct kinematic problems, as the determination of Jacobian matrix and workspace analysis [11]. Here are presented only the solutions of the inverse kinematic problem that is crucial for developing a virtual robot integrated with a control system. The joint coordinates can be determined using the following equations

$$\theta_i = A \tan 2(t_{i1/2}) \quad (5)$$

where

$$t_{i1/2} = \frac{y_P \pm \sqrt{A_i^2 + y_P^2 - B_i^2}}{A_i + B_i} \quad (6)$$

and

$$\begin{aligned} A_i &= (x_P - b_{ix}) \cos(\gamma_i) \\ B_i &= \frac{l_1^2 + (x_P - b_{ix})^2 + y_P^2 - l_2^2}{2l_1} \end{aligned} \quad (7)$$

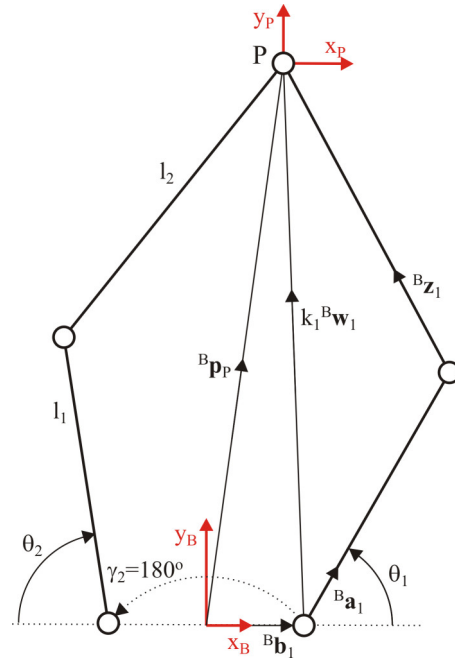


Fig. 4. Kinematic model of BiSCARA robot

IV. METHOD FOR CONFIGURING VIRTUAL ROBOTS IN CONTROL SYSTEM

The basic concept of a configuring virtual robot in the control system is shown in Fig. 5. The virtual robot's configuration is realized under the LinuxCNC software

environment and relies on OpenGL and several interface classes written in Python programming language. Python is an interpreted programming language suitable for scripting tasks, such as developing and configuring virtual environments [12].

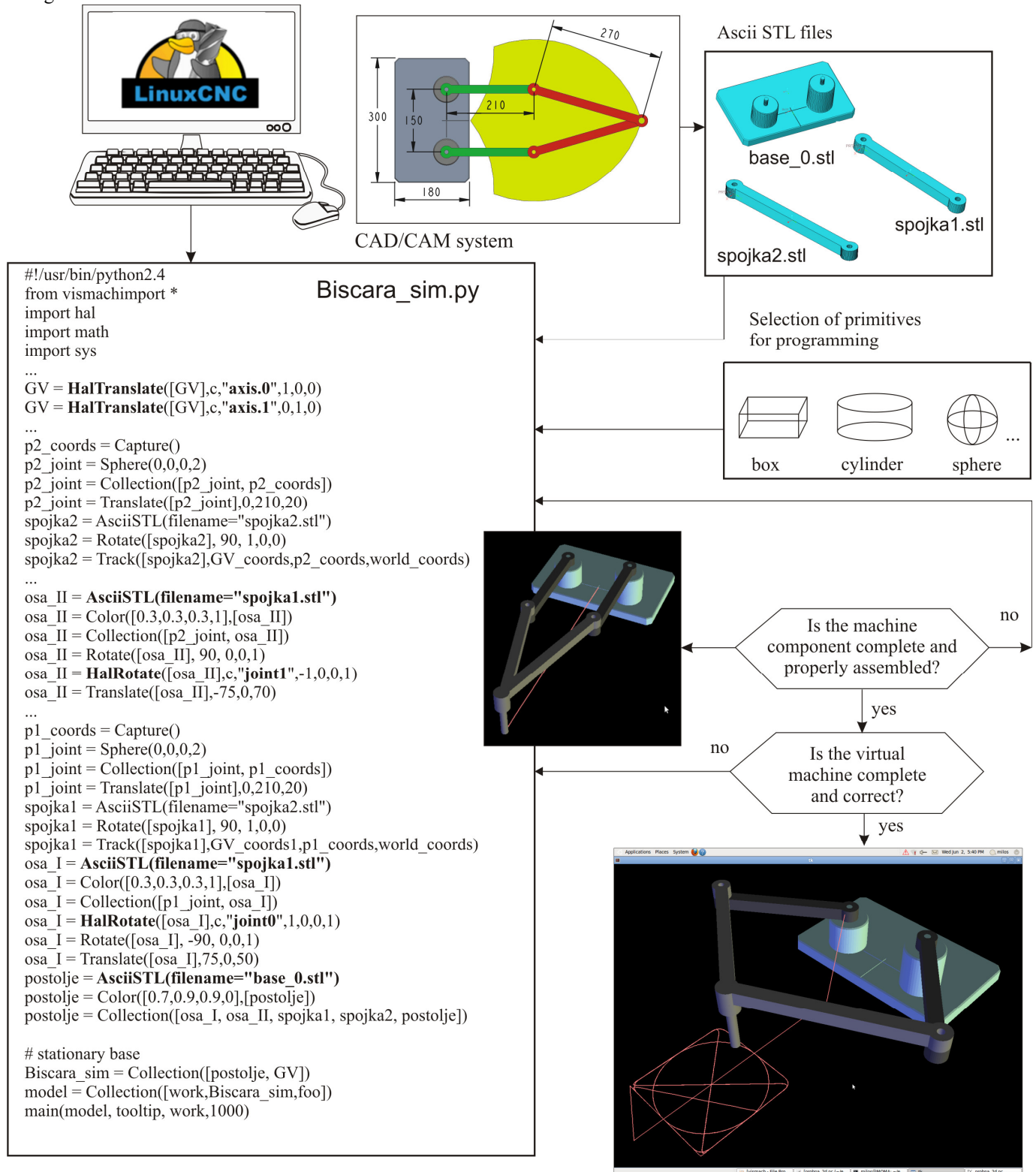


Fig. 5. The basic concept of a configuring virtual machine in a control system realized under the LinuxCNC software environment

Models of robot components were imported in ASCII STL format and connected according to the robot kinematic model. The ASCII STL files of the robot components were obtained from the CAD/CAM system.

After the virtual robot is configured, it is integrated within the LinuxCNC control system. During the program execution for a robot in G-code, the virtual robot components are moving in real-time, according to identical control signals as for the real robot.

The modeling procedure of the virtual robot starts by defining the geometric primitives as parts of the robot assembly in CAD/CAM system. Defining only the essential and functional parts of the robot 3D model may simplify the development of the virtual robot. Basic geometric primitives, e.g., boxes, cylinders, spheres, etc., can be used to form robot parts. The positions of the geometric primitives, as well as the robot parts, are programmed according to the adopted reference frame. The robot parts are connected to form the robot base, and the moving parts of the robot, in this case, are connected via rotational joints.

Step by step programming, testing, and error correction is the only way of configuring an error-free virtual robot. Besides the manual approach, using basic geometric primitives, a CAD system can be used for preparing 3D models of robot components, which are exported as separate files ASCII OBJ or ASCII STL code. In this paper, the virtual BiSCARA robot was configured using PTC Creo. The robot components were converted into an ASCII STL format and then loaded into the Axis GUI through appropriate Python calls, Fig. 5.

Afterward, the imported components are oriented and placed in the virtual environment, resulting in a fully functional virtual robot, as shown in Fig. 5. The virtual robot is placed in a separate window and allows drawing the toolpath within the limitations of the virtual robot elements' movements.

V. SIMULATION AND VERIFICATION

The developed virtual two-axis BiSCARA robot, integrated with the LinuxCNC control system, is suitable for drawing tool paths and laser engraving [15].

The programming system for the considered robot is entirely conventional and the same as for all machine tools which can perform milling or laser engraving. Programs for testing configured virtual robots are generated in the CAD/CAM system using trajectory milling or a special CAM system that can convert drawing in DXF format into G-code.

During the program's execution written in G-code, the virtual robot components are moving in real-time, according to the tool path, thanks to the kinematics built into the control system.

Figure 6 shows the first of the experiments on a virtual robot integrated with the control system, which represents the drawing of the programmed tool path consisting of regular geometric figures of squares with diagonals and inscribed circles.

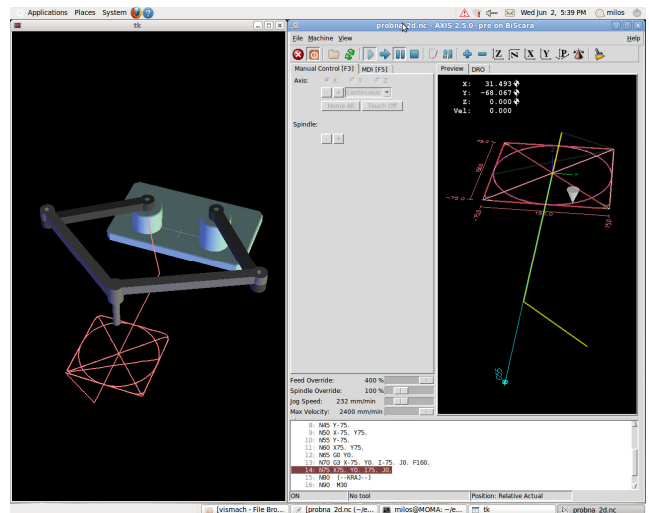


Fig. 6. Simulation of programmed tool path on virtual BiSCARA robot in LinuxCNC control system

Figure 7 shows one of the experiments on a virtual robot integrated with the control system, representing the drawing of the programmed tool path in the shape of the lion drawing. This experiment was used to test systems for programming complex 2D contours, which were programmed, based on DXF drawings and their conversion into G-code.

Some of the details during the virtual trial simulation of the BiSCARA robot are as follows: (i) the appearance of the programmed contour confirms the realized kinematic model, (ii) realized control system is correct, (iii) the used programming system gives correct G-code, (iv) positioning within the robot workspace was correct.

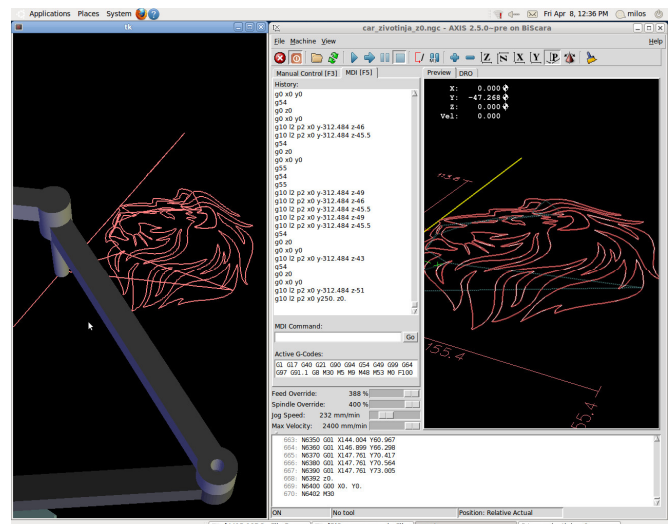


Fig. 7. Simulation of programmed tool path based on lion drawing on virtual BiSCARA robot

VI. CONCLUSION

Testing and verifying the robotic cell before its realization through virtual environments is one of the modern manufacturing engineering tasks. Configuring a virtual environment that includes virtual robots is of crucial importance to verify the programmed trajectory, collision detection, etc.

The paper presents the configuration of a virtual robot in the CAD/CAM and MatLab/Simulink environments on the example of the BiSCARA robot. The virtual CAD and complete kinematic models are used to implement robots in the LinuxCNC software according to a presented method for configuring virtual robots as an integral part of the control system. Configuring virtual robots relies on OpenGL and several interface classes written in Python programming language. First, robot segments were imported in ASCII STL format and connected according to the robot kinematic model, and then the virtual robot was integrated.

Further research will encompass the realization of a virtual prototype of a 4-axis robot by adding on the presented base mechanism one translatory and one rotary axis. Also, the development of a laboratory prototype of a 4-axis robot will be covered in further research.

ACKNOWLEDGMENT

This research was supported by the Science Fund of the Republic of Serbia, Grant No. 6523109, AI-MISSION 4.0 as well as by the Ministry of Education, Science and Technological Development of the Serbian Government under the contract No. 451-03-68/2022-14/200105.

REFERENCES

- [1] N. Vorkapić, S. Živanović, Z. Dimić, B. Kokotović, N. Slavković, "Virtual horizontal machining center LOLA HBG 80 for program verification and monitoring", *FME Transactions*, vol. 49, no. 3, pp. 696-703, 2021.
- [2] J.O. Oyekan, W. Hutabarat, A. Tiwari, R. Grech, M.H. Aung, M.P. Mariani, L. Lopez, T. Ricaud, S. Singh, C. Dupuis, "The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans", *Robotics and Computer-Integrated Manufacturing*, vol. 55, pp. 41-54, 2019.
- [3] W.S. Barbosa, M.M. Gioia, V.G. Natividade, R.F. Wanderley, M.R. Chaves, F.C. Gouvea, F.M. Gonçalves, "Industry 4.0: examples of the use of the robotic arm for digital manufacturing processes", *International Journal on Interactive Design and Manufacturing*, vol. 14, no. 4, pp. 1569-1575, 2020.
- [4] GC Vosniakos, P. Katsaros, I. Papagiannoulis, E. Meristoudi, "Development of robotic welding stations for pressure vessels: interactive digital manufacturing approaches", *International Journal on Interactive Design and Manufacturing*, pp. 1-16, 2022.
- [5] DexTAR, User's Manual, Version 1.0, by Mecademic Inc., 2014-2015.
- [6] W.L.G. Pollard Jr., Spray Painting Machine, US Patent 2,213,108, filed October 29, 1934, issued August 27, 1940.
- [7] H. Makino, A. Kato, Y. Yamazaki, "Research and commercialization of SCARA robot", *International Journal of Automation Technology*, vol. 1, no. 1, pp. 61-62, 2007.
- [8] D.C. Fyler, Control Arm Assembly, US Patent 4,712,971, filed February 13, 1985, issued December 15, 1987.
- [9] A. Joubair, M. Slamani, I.A. Bonev, "Kinematic calibration of a five-bar planar parallel robot using all working modes", *Robotics and Computer-Integrated Manufacturing*, vol. 29, pp. 15-25, 2013.
- [10] N. Slavkovic, S. Zivanovic, N. Vorkapic, "Konfigurisanje virtuelnog prototipa BiSCARA robota", *Časopis Tehnika-Mašinstvo*, Časopis saveza inženjera i tehničara Srbije, vol. 70, no. 3, str 311-317, 2021.
- [11] N. Slavkovic, S. Zivanovic, N. Vorkapic, Z. Dimic, "Development of the Programming and Simulation System of 4-axis Robot with Hybrid Kinematic", *FME Transactions*, (in print).
- [12] C.G. Lee, S.C. Park, (2014) "Survey on the virtual commissioning of manufacturing systems", *Journal of Computational Design and Engineering*, vol. 1, pp.13-222, 2014.
- [13] S. Zivanovic, S. Tabakovic, M. Zeljkovic, Z. Dimic, "Modelling and analysis of machine tool with parallel-serial kinematics based on O-X glide mechanism", *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 43, no. 456, 2021.
- [14] A. Rakic, S. Zivanovic, Z. Dimic, M. Knezevic, "Digital twin control of multi-axis wood CNC machining center based on LinuxCNC", *BioResources*, vol. 16, no. 1, pp.1115-1130, 2021.
- [15] N. Slavković, N. Vorkapić, S. Živanović, Z. Dimić, B. Kokotović, "Virtual BiSCARA robot integrated with open-architecture control system," Proc. 14th International Scientific Conference MMA 2021 – Flexible Technologies, Novi Sad, Serbia, pp. 63-66, 2021.