Society of Production Engineering

ICPES 2025

40th INTERNATIONAL CONFERENCE ON PRODUCTION ENGINEERING - SERBIA 2025

DOI: 10.46793/ICPES25.270M

University of Nis
Faculty of Mechanical Engineering

Nis, Serbia, 18 - 19th September 2025

DEVELOPMENT OF A 2-AXIS SCARA ROBOT MOTION CONTROLLER BASED ON MCU

Bogdan MOMCILOVIC[1*], Nikola SLAVKOVIC[2]
Orcid: 0009-0000-4017-5399; **Orcid:** 0000-0003-1147-284X;
[1]University of Belgrade, Faculty of Mechanical Engineering, Serbia
*Corresponding author: bmomcilovic@mas.bg.ac.rs

Abstract: *The robot systems are designed and developed in such a way that they can successfully realize the specified task. Robot control systems must enable control of robot movement and all other activities necessary for completing the programmed tasks. One of the most important parts of any robot is its motion, and precise control in real-time. This paper proposes the development of a robot motion controller based on the MIKROE MINI-M4 development board with the STM32F415RG microcontroller. Research related to the field of motion controller development presented in this paper includes the realization of interpolation tasks in joint coordinates (PTP control) and interpolation of linear path segments in world coordinates (CP control). To perform these tasks, a mathematical model of interpolation in joint and world coordinates of a two-axis robot with non-trivial kinematics is introduced. The interpolator model is developed so that the joints' or the end-effector's velocities have a trapezoidal profile. To achieve the required trapezoidal velocity profile, synchronized control sequences are generated as a series of pulses with appropriate frequencies for each axis. As a verification, before implementing the mathematical model into the motion controller, simulations of the mathematical model itself and the robot's motion in the joint and world coordinates are performed in the MATLAB environment. The motion controller is implemented on the aforementioned development board with a microcontroller (MCU) using the MIKROC programming environment. The series of pulses with appropriate frequencies is tested on an oscilloscope as the first verification method of the realized motion controller. Reference points of movement defined in joint or world coordinates are obtained by interpreting the program in the MATLAB environment and transferred via serial communication to the MCU with integrated motion control using the MIKROE EasyPIC V7 development board. Experimental verification of the developed motion controller was performed by laser engraving the programmed trajectory of the robot end-effector according to the desired PTP or CP motion control.*

Keywords: *industrial robot, motion controller, non-trivial kinematics, interpolator, MCU*

1. INTRODUCTION

The use of robots in packaging represents a significant technological advancement, evolving from rudimentary mechanized systems to sophisticated, intelligent automation. Over the years, robots have become integral to the packaging process, known for their precision, efficiency, and versatility [1]. In the assembly line of any industry, the SCARA robot is commonly seen. It is generally used for pick and place tasks, but it has a wide range of applications. Hiroshi Makino developed the first SCARA (Selective

Compliance Assembly Robot Arm) robotic arm in Japan in 1979 [2].

Robot systems are developed to ensure they can efficiently accomplish the desired tasks. The robot control systems have to facilitate the control of both movement and any other auxiliary activities essential for executing the programmed tasks. Among the key components of a robot, its ability to move with precision and the need for real-time control is of utmost importance. The robot controller plays a crucial role in ensuring accuracy and directly affects the speed of the robot's movement. A need common to many robot applications is the accurate following of a specific path by the end-effector of the robot, which translates to accurately positioning the individual links that constitute the robot. The individual links are driven by joint actuators, which are rotary electric motors. In other words, the motion of the joint produced by the actuators determines the position and orientation of the end-effector at any time [3, 4]. In recent years, studies have focused on improving robot controllers in both hardware and software. The software includes algorithms that perform kinematics calculations, trajectory planning, and communication, while the hardware platform refers to the hardware circuit [5].

With the advancement of microcontroller (MCU) technology, it is now feasible to implement motion control systems using low-cost, high-performance MCUs that can meet the speed and precision requirements of robot mechanisms. Microcontrollers are well-suited for controlling robotic arms because they are relatively inexpensive, easy to program, and can be used to implement complex control algorithms. Additionally, microcontrollers can be used to interface with a variety of sensors and actuators, which makes them ideal for controlling robotic arms in a variety of different environments [6].

A standard commercial industrial robot usually does not enable open access to its control system. It presents a kind of box where the user cannot change the robot program. An open control system is required for verification of custom control algorithms in the laboratory and for the students' education. It should allow easy modification of the control structure, the ability to add axes, the possibility of connection of additional components, and integrate it into the control structure and monitoring functionalities from a high-level robotic controller to low-level control of servo drives [7].

This paper proposes the development of an open-source robot motion controller based on the MIKROE MINI-M4 development board, featuring the STM32F415RG microcontroller, for, in this development phase, a 2-axis SCARA robot with non-trivial kinematics. The controller is designed to realize interpolation tasks in joint coordinates (PTP control) and interpolate linear path segments in world coordinates (CP control).

2. OUTLINE OF THE CONCEPT

The paper presents the developed MCU-based motion controller of the 2-axis SCARA robot. Fig. 1 shows the developed system, including software and hardware integration with the realized prototype of the industrial SCARA-type robot.

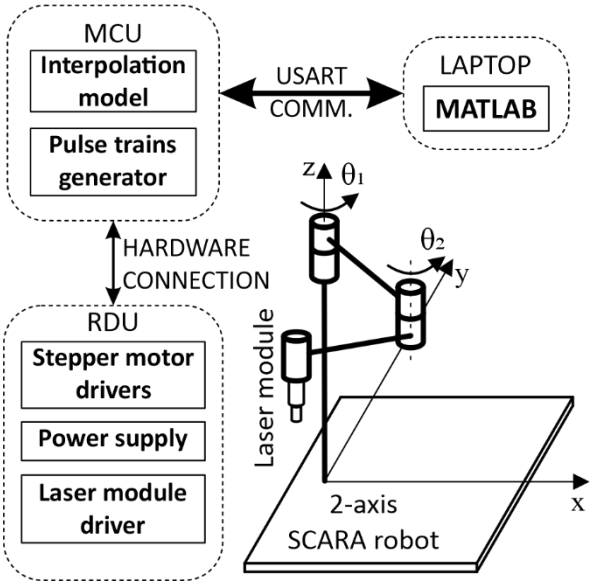

Figure 1. Outline of the concept

The communication with the robot is allowed through the achieved serial (USART) communication between MATLAB and the MCU. The reference points of movement

defined in joint or world coordinates are obtained by interpreting the program in the MATLAB environment. The interpreted data are transferred via serial communication to the MCU.

After obtaining the necessary data, the MCU, with an implemented interpolation model and motion control module, generates pulse trains to control the robot's stepper motors. To verify the developed system and the accuracy of the movements, a laser module is used as the end-effector of the robot to engrave the given trajectory. To move the robot, the neccesary hardware connections with robot drive unit (RDU) are performed.

3. INTERPOLATION MODEL

Path planning is one of the essential problems in robotics. The task of robot motion is defined by the path along which the robot, or end-effector, must move. The path can be represented by a geometric curve or a sequence of positions in joint or world coordinates. The problem of trajectory planning involves determining the desired reference joint or world coordinates as a function of time for the control system so that the robot follows the desired path.

3.1 PTP motion planning

In this paper, for the PTP motion planning, a linear trajectory with parabolic transitions is used, which enables synchronous PTP motion with a trapezoidal profile of joints' velocities, $\dot{\theta}_1$ and $\dot{\theta}_2$ [8]. For a given initial and final positions, q_0 and q_f, the motion duration t_f, as well as the acceleration time t_a, the segments of the polynomial trajectory can be generated as (1).

$$
\begin{aligned}
q(t) &= \begin{cases} q_0 + \frac{1}{2}\cdot\ddot{q}\cdot t^2, 0 \le t \le t_a \\ q_0 + \ddot{q}\cdot t_a \cdot \left(t - \frac{t_a}{2}\right), t_a < t \le t_{fa} \\ q_f - \frac{1}{2}\cdot\ddot{q}\cdot (t_f - t)^2, t_{fa} < t \le t_f \end{cases} \\
\dot{q}(t) &= \begin{cases} \ddot{q}\cdot t, 0 \le t \le t_a \\ \ddot{q}\cdot t_a, t_a < t \le t_{fa} \\ \ddot{q}\cdot (t_f - t), t_{fa} < t \le t_f \end{cases} \\
\ddot{q}(t) &= \begin{cases} \ddot{q}, 0 \le t \le t_a \\ 0, t_a < t \le t_{fa} \\ -\ddot{q}, t_{fa} < t \le t_f \end{cases} \\
t_{fa} &= t_f - t_a
\end{aligned}
\tag{1}
$$

For the 2-DOF planar manipulator, the function $q(t)$ is defined as $q(t) = [\theta_1(t), \theta_2(t)]^T$. To realize the control of the stepper motors, the previous equations need to be defined in the discrete domain with the interpolation period T.

The time functions of acceleration, velocitiy and position from equation (1) could be expressed as $\ddot{\theta}_i(k\cdot T)$, $\dot{\theta}_i(k\cdot T)$ and $\theta_i(k\cdot T)$, for $i = 1,2$. Since the interpolation period T and the motion duration $t = t_f$ are known, the interpolation period T can be excluded from the argument, so the trajectory segments in discrete domain are generated as (2):

$$
\begin{aligned}
\ddot{\theta}_i(k) &= \begin{cases} \ddot{\theta}_{i\max}, k = 0, ..., n_a - 1 \\ 0, k = n_a, ..., (n - n_a) - 1 \\ -\ddot{\theta}_{i\max}, k = (n - n_a), ..., n - 1 \end{cases} \\
\dot{\theta}_i(k+1) &= \dot{\theta}_i(k) + \ddot{\theta}_i(k)\cdot T \\
& k = 0, ..., n-1 \\
\theta_i(k+1) &= \theta_i(k) + \dot{\theta}_i(k)\cdot T + \frac{1}{2}\cdot\ddot{\theta}_i(k)\cdot T^2 \\
& k = 0, ..., n-1
\end{aligned}
\tag{2}
$$

where $n = div(t_f/T)$, $n_a = div(t_a/T)$, and $div()$ represents the integer division operator. Maximum joint velocities and accelerations are calculated using (3):

$$\dot{\theta}_{1m} = \frac{\Delta\theta_1}{(n-n_a)\cdot T} \quad \dot{\theta}_{2m} = \frac{\Delta\theta_2}{(n-n_a)\cdot T}$$
$$\ddot{\theta}_{1\max} = \frac{\dot{\theta}_{1m}}{n_a \cdot T} \qquad \ddot{\theta}_{2\max} = \frac{\dot{\theta}_{2m}}{n_a \cdot T} \tag{3}$$

where $\Delta\theta_1 = \theta_{1f} - \theta_{10}$, and $\Delta\theta_2 = \theta_{2f} - \theta_{20}$.

3.2 CP motion planning

Within this paper, the case of trajectory planning for rectilinear path segments using a parametric description [8, 9] was considered. The parametric description of the straight-line path segment connecting points P_i and P_f, Fig.2, can be expressed as (4):

$$\boldsymbol{p}(s) = \boldsymbol{p}_i + \frac{s}{|\boldsymbol{p}_f - \boldsymbol{p}_i|}\cdot(\boldsymbol{p}_f - \boldsymbol{p}_i)$$
$$\boldsymbol{p}(0) = \boldsymbol{p}_i = [x_0, y_0, 0]^T \tag{4}$$
$$\boldsymbol{p}(|\boldsymbol{p}_f - \boldsymbol{p}_i|) = \boldsymbol{p}_f = [x_f, y_f, 0]^T$$

where s represents the curvilinear coordinate, which determines the distance between the point P in relation to the initial position P_i.

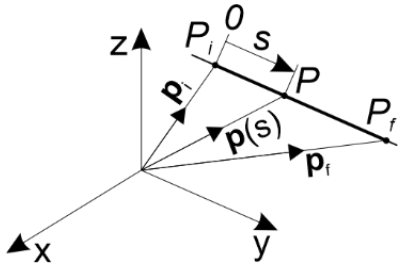

Figure 2. Parametric description of the straight-line

For a given initial and final positions, $\boldsymbol{p_i}$ and $\boldsymbol{p_f}$, the motion duration t_f, as well as the acceleration time t_a, the first and second derivatives of the curvilinear coordinate s are calculated as (5):

$$\dot{s} = \frac{\Delta s}{t_f - t_a}$$
$$\ddot{s} = \frac{\dot{s}}{t_a} \tag{5}$$
$$\Delta s = |\boldsymbol{p}_f - \boldsymbol{p}_i|$$

The rectilinear path segments are now calculated as (6):

$$\ddot{s}(t) = \begin{cases} \ddot{s}, 0 \le t \le t_a \\ 0, t_a < t \le t_{fa} \\ -\ddot{s}, t_{fa} < t \le t_f \end{cases}$$
$$\dot{s}(t) = \begin{cases} \ddot{s}\cdot t, 0 \le t \le t_a \\ \dot{s}, t_a < t \le t_{fa} \\ -\ddot{s}\cdot(t_f - t), t_{fa} < t \le t_f \end{cases} \tag{6}$$
$$s(t) = \begin{cases} \frac{1}{2}\cdot\ddot{s}\cdot t^2, 0 \le t \le t_a \\ \dot{s}\cdot\left(t - \frac{t_a}{2}\right), t_a < t \le t_{fa} \\ -\frac{1}{2}\cdot\ddot{s}\cdot(t_f - t)^2, t_{fa} < t \le t_f \end{cases}$$

For the known interpolation period T and the motion duration $t = t_f$, the trajectory in the discrete domain can be expressed as (7):

$$\ddot{s}(k) = \begin{cases} \ddot{s}, k = 0, ..., n_a - 1 \\ 0, k = n_a, ..., (n-n_a) - 1 \\ -\ddot{s}, k = (n-n_a), ..., n-1 \end{cases}$$
$$\dot{s}(k+1) = \dot{s}(k) + \ddot{s}(k)\cdot T$$
$$k = 0, ..., n-1 \tag{7}$$
$$s(k+1) = s(k) + \dot{s}(k)\cdot T + \frac{1}{2}\cdot\ddot{s}(k)\cdot T^2$$
$$k = 0, ..., n-1$$

Based on the previously calculated and equation (4), as well as the first and second derivatives of equation (4), the end-effector's acceleration $\ddot{\boldsymbol{p}}_{\boldsymbol{e}}$, velocity $\dot{\boldsymbol{p}}_{\boldsymbol{e}}$ and position $\boldsymbol{p}_{\boldsymbol{e}}$ in the world coordinates are determined as (8) for $k = 0,1, \dots, n-1$:

$$\ddot{\boldsymbol{p}}_{\boldsymbol{e}}(k)=\begin{bmatrix}\ddot{x}_e(k)\\ \ddot{y}_e(k)\\ 0\end{bmatrix}=\ddot{s}(k)\cdot\frac{(\boldsymbol{p}_f-\boldsymbol{p}_i)}{|\boldsymbol{p}_f-\boldsymbol{p}_i|}$$

$$\dot{\boldsymbol{p}}_{\boldsymbol{e}}(k)=\begin{bmatrix}\dot{x}_e(k)\\ \dot{y}_e(k)\\ 0\end{bmatrix}=\dot{s}(k)\cdot\frac{(\boldsymbol{p}_f-\boldsymbol{p}_i)}{|\boldsymbol{p}_f-\boldsymbol{p}_i|} \qquad (8)$$

$$\boldsymbol{p}_{\boldsymbol{e}}(k)=\begin{bmatrix}x_e(k)\\ y_e(k)\\ 0\end{bmatrix}=\boldsymbol{p}_i+s(k)\cdot\frac{(\boldsymbol{p}_f-\boldsymbol{p}_i)}{|\boldsymbol{p}_f-\boldsymbol{p}_i|}$$

For the control software realization, the 2-DOF manipulator's joint coordinates, $\theta_1(k)$ and $\theta_2(k)$, are determined based on the calculated position of the end-effector $\boldsymbol{p}_{\boldsymbol{e}}(k)$ using the inverse kinematics equations (9) for SCARA type robot:

$$\begin{gathered}c\theta_2=\frac{x_e^{\,2}(k)+y_e^{\,2}(k)-a_1^{\,2}-a_2^{\,2}}{2\cdot a_1\cdot a_2}\\ s\theta_2=-\sqrt{1-c^2\theta_2}\\ \theta_2(k)=A\tan 2\left(s\theta_2,c\theta_2\right)\\ \alpha=A\tan 2\left(y_e(k),x_e(k)\right)\\ \beta=A\tan 2\left(a_2\cdot s\theta_2,a_1+a_2\cdot c\theta_2\right)\\ \theta_1(k)=\alpha-\beta\end{gathered} \qquad (9)$$

where a_1 and a_2 represent the lengths of the robot's segments. The inversion of the Jacobian matrix J^{-1} could be used to determine the velocities and accelerations in joint coordinates.

4. MOTION CONTROL DEVELOPMENT

The development concept of an MCU-based motion controller presented in this paper consists of three parts: 1) software implementation of the interpolation model, 2) routine development for generating pulse trains, and 3) hardware connection of the MCU to a prototype of the industrial SCARA-type robot.

4.1 Software implementation

For programming the selected STM32F415RG microcontroller mikroC PRO for ARM compiler was used. The environment of the compiler enabled writing, compiling, and debuging the code.

The presented interpolation model for the joint and world coordinates trajectory planning is implemented in the form of routines that are executed depending on the desired motion of the robot (PTP or CP mode). In order to control the motion of a stepper motors driven robot, it is necessary to convert the calculated values of the joint coordinates into the appropriate number of pulses.

Given that the trajectory is divided into segments according to the interpolation period T and the number of segments is known, the joint coordinate's change is calculated relative to the previous segment, for each current segment. For the PTP mode, this change is calculated using (11), according to (2):

$$\Delta\theta_i(k)=\dot{\theta}_i(k-1)\cdot T+\frac{1}{2}\cdot\ddot{\theta}_i(k-1)\cdot T^2 \qquad (11)$$

while for the CP mode, this change is calculated using (12), according to (9):

$$\Delta\theta_i(k)=\theta_i(k)-\theta_i(k-1) \qquad (12)$$

The joint coordinates' changes (11) and (12) are converted into the corresponding number of pulses based on the angular resolution of each robot axis. The angular resolution of the robot axis φ_k is calculated based on the hardware characteristic of the stepper motor α_k, the set microstep on the stepper motor driver ms, and the joint's gear ratio i_p using (13):

$$\varphi_k=\frac{\alpha_k}{ms\cdot i_p} \qquad (13)$$

The corresponding number of steps $k_{k,i}$ is now obtained based on equation (14), noting that the angular resolutions of the individual axis $\varphi_{k,i}$ do not have to be equal.

$$k_{k,i}(k)=\frac{\Delta\theta_i(k)}{\phi_{k,i}} \qquad (14)$$

Based on the calculated number of steps $k_{k,i}$ and the known interpolation period T, the one pulse duration T_{imp} is calculated, while the

high-level duration T_h is known in advance, Fig.3.

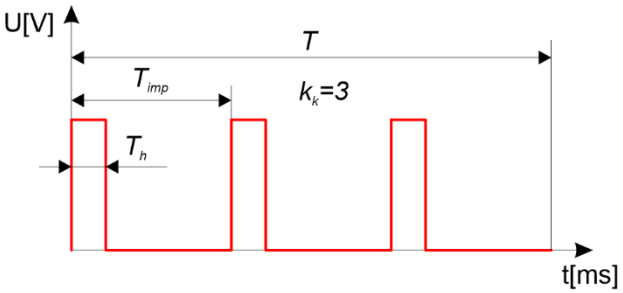

Figure 3. Pulse train characteristics

4.2 Pulse trains generation

To create a routine that generates pulse trains in real time on the selected MCU, a timer is used, which counts clock cycles to measure time intervals.

```
routine: timer_interrupt()
        On every microsecond
          DO:
              micros_count++;
routine: pulse_train()
        while (micros_count <= T)
          DO:
              updatePortAState();
              updatePortBState();
              switchPortsStates();
routine: updatePortA(B)State();
        if (micros_count <= Th)
          DO:
              PortAState = 1;
              (PortBState = 1;)
        elif (micros_count <= TimpA(B) -Th)
           DO:
              PortAState = 0;
              (PortBState = 0;)
routine: switchPortsStates();
        Update state of the physical pin
          DO:
              GPIO_A = PortAState;
              GPIO_B = PortBState;
```

Figure 4. The algorithm for pulse trains generation

The task of the developed routine is to perform a synchronous update of the corresponding ports' states on the MCU, defined as outputs. The algorithm for making such a routine is shown in Fig. 4.

The pulse trains generated in this way are tested on a 2-channel oscilloscope as the first verification method of the realized motion controller, Fig 5.

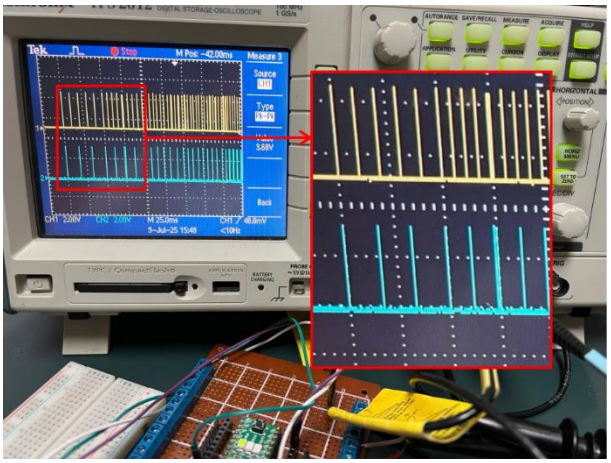

Figure 5. Oscilloscope verification

The oscilloscope display confirmed the synchronous generation of the control sequence in real time, based on the calculated number of pulses for each path segment, as well as the one pulse duration T_{imp}.

4.3 Hardware connection

To realize the motion controller of the 2-axis SCARA robot, the hardware connection of the MCU with the necessary electronic components is performed. The connection between all the components used to prove the presented concept is shown in Fig. 6.

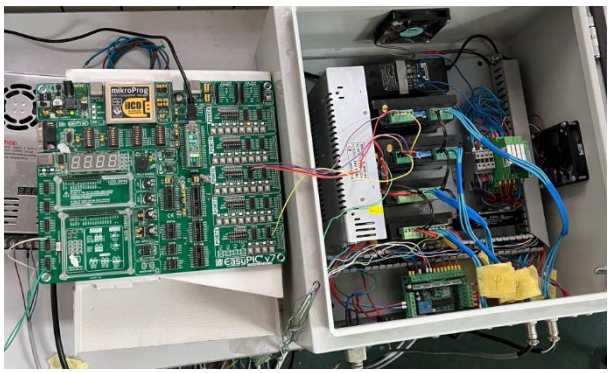

Figure 6. Hardware connection of the components

To achieve the communication and data exchange between MATLAB and MCU, the MIKROE MINI-M4 development board with the STM32F415RG microcontroller is placed on the MIKROE EasyPIC V7 development board, which provides the necessary peripherals. In this case, a laser module is used as an end-effector of the robot. The laser module provides engraving of

the robot's toolpath. The remaining components represent the standard elements for controlling the robot, which uses stepper motors for drives.

5. EXPERIMENTAL VERIFICATION

The developed MCU-based 2-axis SCARA robot motion controller is verified through several experiments. To verify the accuracy of the positioning and the robot's tooltip movement according to the given trajectory, the laser module is used as the robot's end-effector.

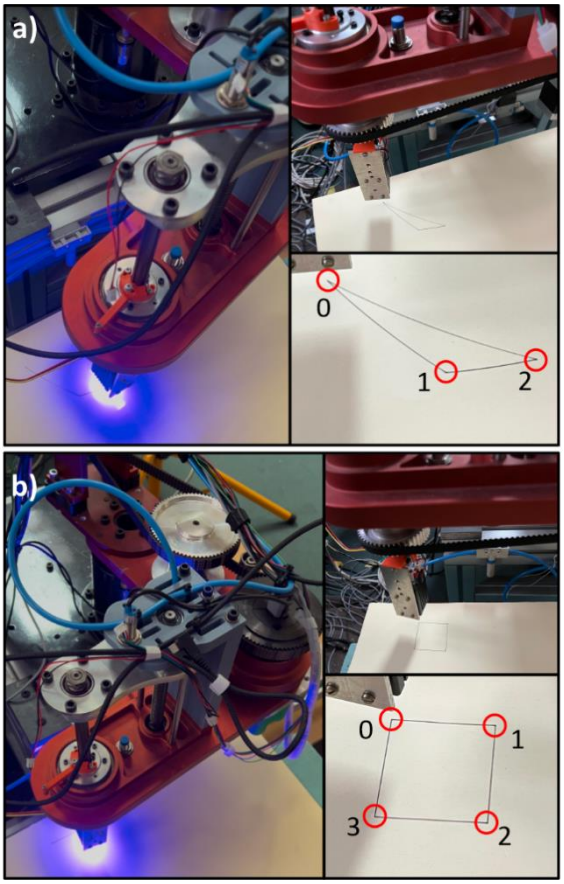

Figure 7. Experimental verification

The developed motion controller is tested with the desired trajectories in PTP or CP motion mode. In both cases, the trajectory consists of several positions defined in the joint (PTP mode) or world (CP mode) coordinates. The given acceleration time is $t_a = 0.5s$, time of the robot movement between two positions is $t_f = 5s$, the high-level duration is $T_h = 250us$, and interpolation period is $T = 50ms$.

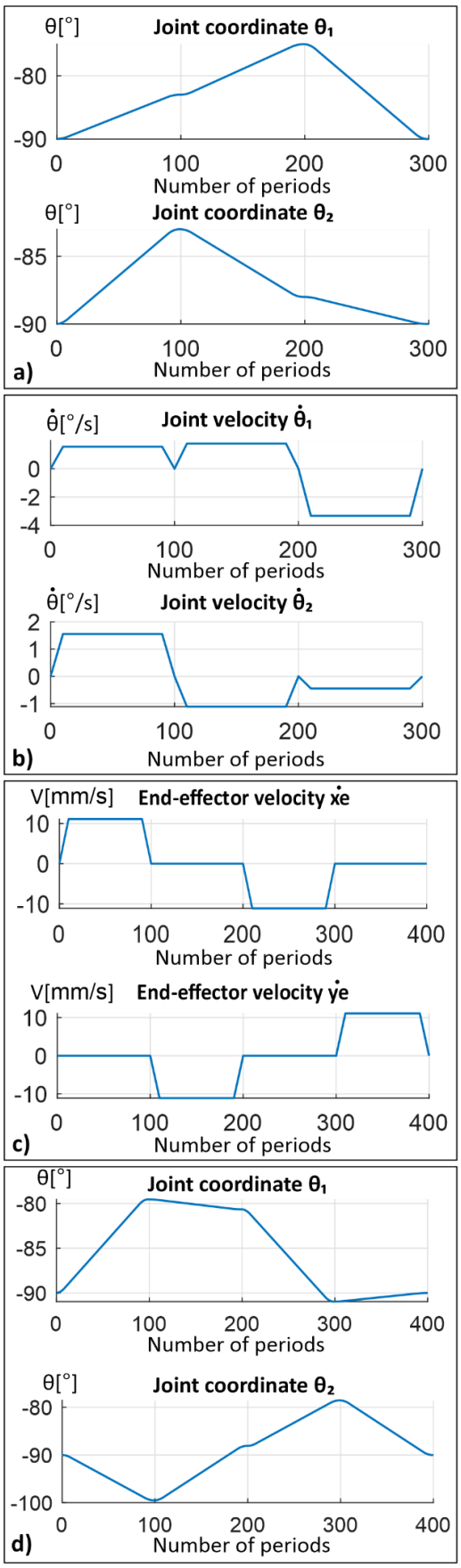

Figure 8. Interpolation results for PTP and CP mode

The engraved trajectory of the robot's toolpath as the result of the defined robot's

movement in PTP motion mode with defined positions 0-1-2-0, is shown in Fig. 7a. According to the defined interpolation model, the robot's toolpath between two positions (0-1, 1-2 or 2-0) represents a part of the arc obtained as a result of synchronous movement of the two axes. In this case, the robot positions are defined in the joint coordinates.

The engraved trajectory of the robot's toolpath, as the result of the defined robot's movement in CP motion mode with defined positions 0-1-2-3-0, is shown in Fig. 7b. In this case, the robot positions are defined in the world coordinates, and the robot's toolpath between two positions is a straight line.

Based on the interpolation model and the set parameters of the robot motion in PTP mode, Fig. 8a shows the joint coordinates dependence of the interpolation periods, and Fig. 8b shows the joint velocities dependence of the interpolation periods. Defined robot positions in the joint coordinates are: [0(-90°, -90°), 1(-83°, -83°), 2(-75°, -88°), 0(-90°,90°)].

For CP motion mode, Fig. 8c shows the end effector velocity dependence of the interpolation periods, and Fig. 8d shows the joint coordinates dependence of the interpolation periods for the given motion in CP mode. Defined robot positions in the world coordinates are: [0(-275mm, -275mm), 1(-225mm, -275mm), 2(-225mm, -325mm), 3(-275mm, -325mm), 0(-275mm, -275mm)].

6. CONCLUSION

This paper presents the development of the MCU-based motion controller for the 2-axis SCARA robot with non-trivial kinematics. To achieve this, the interpolation model is implemented on the MCU as routines supporting both point-to-point (PTP) and continuous path (CP) robot control. The presented concept is validated through several experiments, and the end-effector trajectories are obtained via laser engraving.

By measuring the physical distance between reference points on the obtained trajectories, it can be concluded that the robot moves precisely, according to the given coordinates. Furthermore, the measured time for the robot's movements corresponds to the programmed movement time.

Future research will focus on enhancing the interpolation model for circular path segments and developing methods for planning the orientation of the end-effector.

ACKNOWLEDGEMENT

This research was supported by the Ministry of Science, Technological Development and Innovations of the Serbian Government under the contract No. 451-03-137/2025-03/ 200105.

REFERENCES

[1] Riyan Vikas Patel, 2024, Opportunities and Challenges for Adoption of Robotics in Packaging Industry: A Review, IOSR Journal of Engineering, Vol. 14, and Issue 1, pp. 32-38.

[2] Tay, See Han, Wai Heng Choong, and Hou Pin Yoong. "A review of SCARA robot control system." 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET). IEEE, 2022.

[3] Mustafa, Mustafa M., and Ibrahim Hamarash. "Microcontroller-based motion control for DC motor driven robot link." 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM). IEEE, 2019.

[4] Gopal, Madan. Control systems: principles and design. McGraw-Hill Science, Engineering & Mathematics, 2008.

[5] Cong, V. D. (2021). Industrial Robot Arm Controller Based on Programmable System-on-Chip Device. FME Transactions, 49(4).

[6] Rahulraj Mahan, Aarti Patil, Jyoti Kedar, 2024, Movementable Robotic Arm Using MicroController, International Journal of Enhanced Research in Management & Computer Applications, Vol. 13, Issue 4, pp. 654-659.

[7] Robert UVEGES, Frantisek DUROVSKY, David LINDR, OPEN ROBOTIC CONTROLLERS, Acta

Electrotechnica et Informatica, Vol. 16, No. 3, 2016, pp. 8–13.

[8] D. Milutinović, Industrijski roboti, Univerzitet u Beogradu - Mašinski fakultet, 2024.

[9] L. Sciavicco, B. Siciliano, Modelling and control of robot manipulators, Springer Science and Business Media,2012.