Acta Mechanica Slovaca **27 (3): 12 - 17,** September **2023**
https://doi.org/10.21496/ams.2023.029

Acta Mechanica Slovaca
ISSN 1335-2393
www.actamechanica.sk

Design and Control of a 3D Printed Robotic Arm

Tamás Bárány [1], László Rónai [1*]

[1] Robert Bosch Department of Mechatronics, Faculty of Mechanical Engineering and Informatics, University of Miskolc, 3515 Miskolc, Hungary

Abstract: The paper is dealing with a 3D printed robotic arm, which is capable to use in the education. The RRR type, 4 degrees of freedom robot is designed in the Autodesk Fusion 360 software, the body of each joint is manufactured by the use of Fused Deposition Modelling technology. The system uses micro servo units as actuators, the control unit is an ATmega2560 microcontroller. Inverse kinematics problem of the arm is solved by geometric approach, joint angles to be determined serve the basis of the control. The workspace of the robot can be analysed via a special purpose Excel spreadsheet. The control program is developed under VS Code software. The system has an OLED screen, which can give information about the reachability of the coordinates, and the state of the robotic arm. The points to be reached are contained by the uploaded control program. The system is capable to use as demonstrating purposes, especially for engineering students.

Keywords: Robotic arm; Microcontroller programming; Inverse kinematics

1. Introduction

One important aspect of the Industry 4.0 [1] philosophy is the emphasis on robotics within automation. The expansion of robots in the recent decade covers a very large area, in addition to industry, there is also an increasing demand and interest in the development of robotic arms for other purposes. Nowadays, self-designed robots are becoming more and more popular [2]-[4].

As a result of the increasingly favourable purchase cost of 3D printers, the range of home users has grown rapidly in recent years, which has pushed forward the design of robotic arms suitable for demonstration purposes.

Robotic arms are used not only at industrial companies, but also to serve disabled or elderly people, e.g., feeding them [5] and for demonstration and educational purposes. Robotic arms can be used for 3D printing, a good example of which is given in [6]. In [7] a robotic arm called HydraX is presented, which has low production costs and 6 degrees of freedom (DOF). The unit is manufactured by 3D printing and programmed in G-code. The control of the robot is based on the Arduino Mega development platform, the robot can be used for additive and subtractive manufacturing processes. In [8] Ananias et al. introduces a low-cost, 3D printed 6 DOF robotic arm. The presented collaborative robot serves educational purposes; it has a graphical user interface application developed in MATLAB software. The repeatability of the system is under 0.4 mm.

This article deals with the design and control of a robotic arm with 4 DOF, built from rotational constraints. The main goal of the development is to demonstrate the possibilities offered by 3D printing, and to create a complex mechatronic system, which can be capable to use in the field of education.

The paper is organized as follows: Section 2 presents the design of the RRR type [9] robotic arm, which is performed in the Autodesk Fusion 360 software, paying attention to printability aspects. Section 3 deals with the analysis of the workspace of the robot, therefore a special purpose Excel file is developed, which checks whether

*** Corresponding author:** László Rónai, ***E-mail address:*** *laszlo.ronai@uni-miskolc.hu*

the robot can reach the given point based on the entered target positions. In order for the unit to be suitable for positioning based on the entered target coordinates, it is necessary to solve the inverse kinematics task [9], [10] of the robot, which is based on geometric approach.

The control strategy is described in Section 4. The program is developed onto an Arduino Mega platform, expanded with an OLED display that gives feedback to the user. The system contains SG90 micro-servos at the joints. The program code is created in VS Code development environment. The last Section presents the concluding remarks and plans for the future.

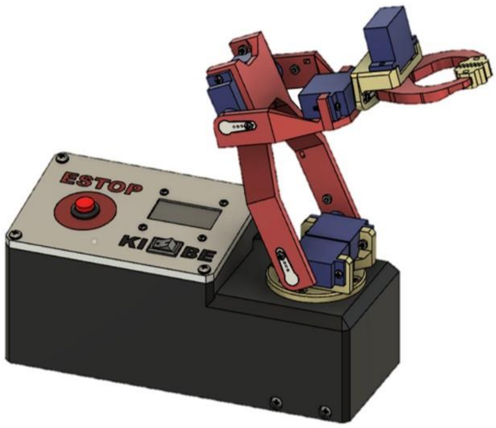

Figure 1: 3D model of the robot.

2. Design of the robot

The parts of the robot are made with 3D printing, therefore some manufacturability aspects had to be considered during its design. The first main consideration is that the model has a relatively large flat surface that can be laid on the table of the printer. Because if this surface is too small compared to the dimensions of the model, it can happen that it pops up during printing, separates from the table due to the effect of temperature changes.

The second aspect is the consideration of the geometric shapes having overhang. During Fused Deposition Modelling (FDM) printing process, the layers need to be supported. The third aspect is the consideration of the orientation of the layer lines, in order to ensure the printability and the proper mechanical properties.

The parametric model of the robot shown in Figure 1 is developed in Autodesk Fusion 360 software. The system has 4 DOF ensured by 7 micro servo units. Two-two actuators are placed in joint 1 and joint 2 to handle the loads that occur.

The control panel contains an emergency stop (ESTOP), an ON/OFF switch and an OLED screen. The activation of the ESTOP button halts the movement of the robotic arm. The ON/OFF switch ensures the starting condition of the moving cycle. The user specific messages of the state of the robot can be read from the built-in OLED screen.

The end-effector of the robot can be seen in Figure 2. It has two rotatable toothed parts at the end points, which provide the proper grasping process of different shaped workpieces.

Inside the base of the robot there is a metal ballast, and the development platform responsible for the control is also placed here. In addition to the robot, it is also necessary to design a circuit board (shield), which will play an important role in the orderly connection of the cables. The circuit board model is illustrated in Figure 3.

Figure 2: Design of the end-effector.

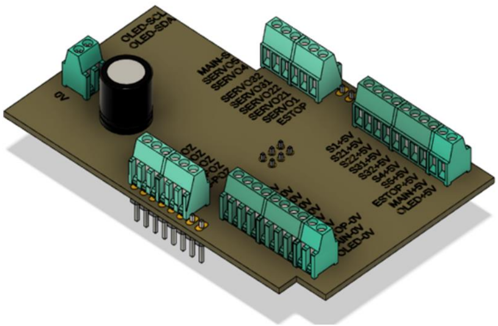

Figure 3: The designed shield of the development platform.

3. Inverse kinematics and workspace analysis

The following subsections deal with the solution of the inverse kinematics problem and the workspace analysis of the robotic arm.

3.1. Inverse kinematics of the robot

The basis of the control is the calculation of the joint angles in awareness of the target positions. For the joint angles to be produced, the inverse kinematics problem [9] – [11] must be solved, which can be discussed in several ways, these are: analytical, numerical methods, and the geometric approach. Due to the complexity of the construction, it is advisable to choose the geometric approach. The kinematic scheme of the system is shown in Figure 4. The length of each robotic segment is $a_0 = 69$ mm, $a_1 = 80$ mm, $a_2 = 125$ mm.

The coordinates (X, Y, Z) of the Tool Center Point (TCP) are given by the user, and the system calculates the angles of the joints. The angle α and β can be determined with the use of the law of cosines:

$$\alpha = \arccos\left(\frac{a_1^2 + r_2^2 - a_2^2}{2a_1 r_2}\right) \tag{1}$$

where $r_2 = \sqrt{X^2 + Y^2 + (Z - a_0)^2}$ is the distance between joint 1 and joint 3,

$$\beta = \arccos\left(\frac{a_1^2 + a_2^2 - r_2^2}{2a_1 a_2}\right) \tag{2}$$

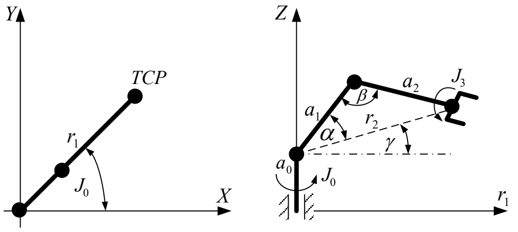

Figure 4: Definition of the angles of the joints.

Angle γ is calculated as follows:

$$\gamma = \arctan\left(\frac{Z - a_0}{r_1}\right) \tag{3}$$

where $r_1 = \sqrt{X^2 + Y^2}$. The angle of the joints can be written as:

$$J_0 = \arctan 2(Y, X) \tag{4}$$

$$J_1 = \alpha + \gamma \tag{5}$$

$$J_2 = \beta \tag{6}$$

where arctan2 is the two-argument arctangent function. The joint angle J_3 serves the orientation of the end-effector, which limit is ±90°.

3.2. Workspace analysis

A special purpose Excel file (see Figure 5) is prepared for the visual representation of the workspace, in which, after entering the X, Y and Z coordinates, the table determines whether the point specified by the user is reachable using the formulas defined for inverse kinematics in subsection 3.1.

In the course of the calculation, the table determines the angle of each joint. Then checks whether this angle is within the permitted angle range, and if the coordinate is available, it calculates the individual joint positions, which are displayed on a graph. In Figure 5, the projection view of the robotic arm is shown, thus the horizontal axis represents the ***r1*** projection distance, and the vertical axis is coordinate Z.

The detail of the worksheet has been filled with the available coordinate points accessible by robot in the range 0 ÷ 204 mm in the projection direction and 0 ÷ 275 mm in the direction Z.

The availability of each point is checked with the previously introduced formulae, if the point is available, it is drawn on the diagram.

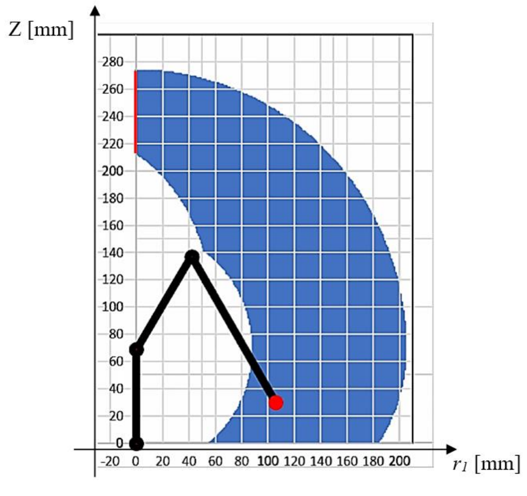

Figure 5: Workspace analysis of the arm.

In order to determine that a robot position is singular the Jacobian matrix is used. The rank of the matrix will depend on the configuration of the robot [10]. Configurations where the rank of the Jacobian matrix is less than the maximum are called singular positions. At a singular configuration, the inverse kinematics has no solution or infinite solution.

If axes of joint 0 and joint 3 are collinear the robotic arm will be in a singular position. It is

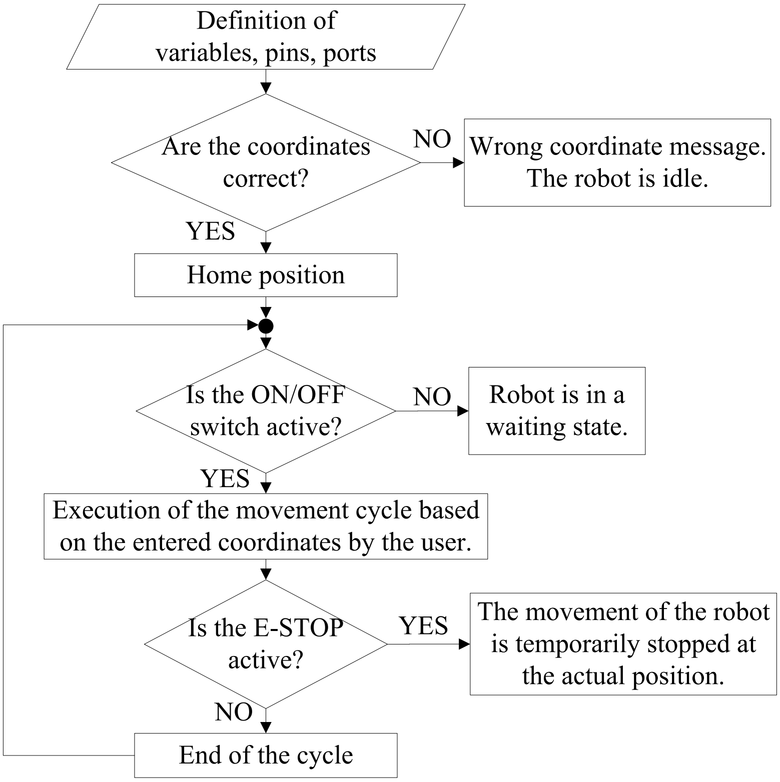

Figure 6: Flow chart of the control program.

possible when the robotic arm reaches its maximum projection distance in a vertical configuration, or the end-effector of the robot is in the position of $r1$ = 0 besides a given Z. The singular positions are denoted with a straight red line shown in Figure 5.

4. Description of the control strategy

The control program of the robot is developed in the Visual Studio Code. After the system has started, the definition of the variables, specification its initial values, and configuring the pins are performed. The correction of the user specified coordinates is checked, as can be seen in Figure 6.

The home position is taken joint by joint, followed by a check of the state of the ON/OFF switch. If the switch is ON, the movement instructions are executed cyclically, otherwise the robot will be in standby mode. Using the ESTOP, the robot will stop in the current position. At the end of the movement cycle, the robot will continue to operate depending on the state of the ON/OFF switch. The process can be monitored on the OLED display built into the top of the robot base.

The main parts of the program are contained in functions, e.g., the ***check()*** function for checking the correctness of coordinates. The joint angles generated after solving the inverse kinematics problem are stored by the robot controller in a two-dimensional array, which the program can use later when moving the individual motors.

The principle of programming the end-effector includes the determination of the opening-closing distance, which is adapted to the dimension of the workpiece. This requires the definition of a ***wp_dim*** constant (see Figure 7), which stores the size of the workpiece to be gripped. The variable ***open_tol*** specifies how much clearance there should be on the gripping sides of the object when the mechanism opening. The variable ***press*** is the extra distance with which the gripper squeezes more from both sides than the actual size of the object.

To determine the angle of the arms of the end-effector, a reference angle is needed, i.e., ***srv5Def*** constant, when ϕ=0°. This is the angle at which the end points of the arms are the same distance as the axle distance, i.e., then ***angles[i][4] = 0***. To calculate

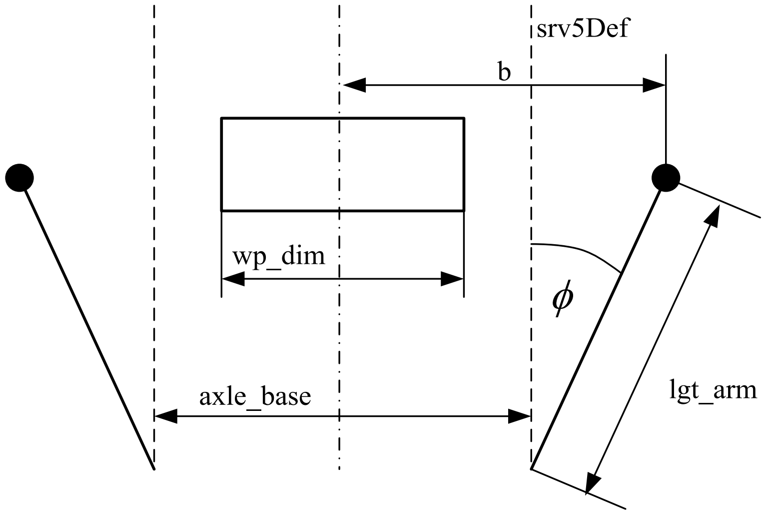

Figure 7: Explanation of the operation of the end-effector.

Table 1: Program detail for operating the end-effector.

```
if (cor[i][4]) b=(wp_dim/2)-press; //end-effector is closed
else b=(wp_dim/2)+open_tol; //end-effector is opened
if (b>=axle_base/2) angles[i][4]=srv5Def+asin(abs(b-axle_base/2)/lgt_arm);
else angles[i][4]=srv5Def-asin(abs(b-axle_base/2)/lgt_arm);
```

the angle ϕ, it is first necessary to determine the value of b, which is a function of whether the gripper is in an open or closed position is located. As a second step, this distance must be compared with half of ***axle_base*** to find out if the servo motor needs to pick up a workpiece larger or smaller than the base angle (see Table 1).

5. Results and discussion

The individual elements are 3D printed and assembled, which can be seen in Figure 8. The first tests showed that the operation of the robot needs to be further refined.

The home position is recorded very dynamically and quickly by swinging the robotic arm. The problem was caused by not knowing the current position of the joints, since the function ***servo.read()*** in the ***Servo.h*** library used to move the motors can only return the last entered angle value. To obtain the current angle value of each servo motor, the motors had to be disassembled and signal wires was soldered to their potentiometers. By connecting the soldered wires to the analogue inputs of the microcontroller and then scaling via the function ***map()***, the current angle values can be obtained.

In addition, another problem had to be solved, which is the following: In the course of testing of the system, it was found that the servos do not take the angular position that the code gives them with the function ***servo.write()***. Therefore, it was necessary to scale the angles to be entered. Measurements were required, in which the axes of each motor are positioned every 45 degrees starting from zero. The conclusion that can be drawn from the measurement is that the actual angular rotation of the motors is approximately 160° instead of 180°.

A floor plan is made to realistically display the workspace of the robotic arm (see Figure 8). This shows the proper position of the base of the system, the semi-circular workspace. The floor plan serves as an aid for the precise placement of the workpieces manipulated by the robotic arm, and for testing and checking accuracy and repeatability. During tests, the repeating accuracy within the range of 2 mm.

The load capacity of the robotic arm is determined for its maximum projection distance in a horizontal configuration. In the course of the determination of the loads, robotic parts made of the same material and density were grouped together in the Fusion 360 software. Thus, the centre of masses and their distances can be calculated. Based on the determined forces from the robotic parts, and the torque of the two servo motors placed between the a0 and a1 robotic segments, the maximum liftable weight can be determined with a torque equation. After calculations, the result is approximately 140 g.

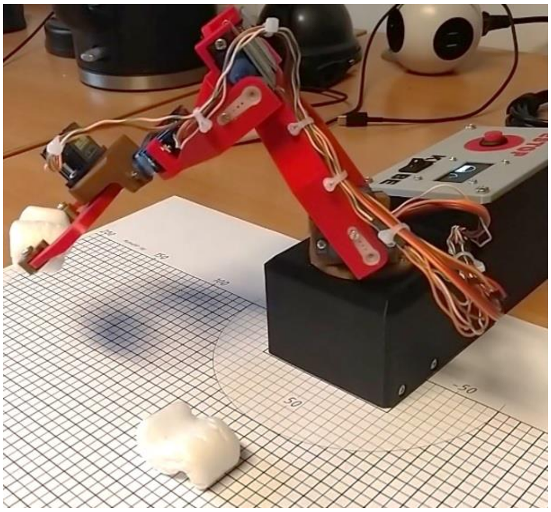

Figure 8: The constructed robotic arm.

6. Conclusions

The paper dealt with the design and control of a robotic arm. The main parts of the structure were designed in the Autodesk Fusion 360 software and printed with FDM technique. The inverse kinematics problem of the system was solved by the use of geometric approach. A special purpose Excel worksheet was made in order to check the correctness of the coordinates to be uploaded to the robot.

The development of the program code was performed in VS Code environment. Tests were performed to check the correctness of the developed system. Some modifications, which primarily affected servomotors were done.

The coordinates describing the movement of the robotic arm are currently contained in the code uploaded to the microcontroller. Future plans include giving the coordinates using SD card, from which the system would be able to read the new coordinates. In addition, the system will be perfectly applicable for demonstration purposes, which will serve to expand the mechatronics knowledge of engineering students.

References

[1] Vuksanović, D., Ugarak, J., Korčok, D. (2016). Industry 4.0: the Future Concepts and New Visions of Factory of the Future Development. Sinteza 2016 - International Scientific Conference on ICT and E-Business Related Research, Belgrade, Singidunum University, Serbia, pp. 293–298, doi: 10.15308/Sinteza-2016-293-298

[2] Marcinko, P., Virgala, I., Miková, L., Prada, E., Kelemenová, T., Kelemen, M., Varga, M. (2019). Chimney Cleaning and Inspection Robot. Acta Mechanica Slovaca, vol 23, no. 3, pp. 6–9, doi: 10.21496/ams.2019.016

[3] Gmiterko, A., Kelemen, M., Virgala, I., Miková, L. (2014). Educational Model of Four Legged Robot. Acta Mechanica Slovaca, vol 18, no. 3, pp. 68–73, doi: 10.21496/ams.2014.032

[4] Virgala, I., Miková, L., Kelemen, M. (2018). Robotic snakes. Acta Mechanica Slovaca, vol 22, no. 3, pp. 38–43, doi: 10.21496/ams.2018.025

[5] Kruthika, K., Kiran Kumar, B. M., Lakshminarayanan, S. (2016). Design and Development of a Robotic Arm. International Conference on Circuits, Controls, Communications and Computing (I4C), doi: 10.1109/CIMCA.2016.8053274

[6] Raghavan, A., Neethu P.P, Aji J. (2017). 3D Printing Robotic Arm. International Journal of Engineering and Management Research, vol. 7, Issue 3, pp. 631–634.

[7] Krimpenis, A. A., Papapaschos, V., Bontarenko, E. (2020). HydraX, a 3D printed robotic arm for Hybrid Manufacturing. Part I: Custom Design, Manufacturing and Assembly. Procedia Manufacturing, vol. 51, pp. 103–108, doi: 10.1016/j.promfg.2020.10.016

[8] Ananias, E., Gaspar, P. D. (2022). A Low-Cost Collaborative Robot for Science and Education Purposes to Foster the Industry 4.0 Implementation. applied system innovation, vol. 5(4), doi: 10.3390/asi5040072

[9] Kurfess, T. R. (2005). Robotics and Automation Handbook, CRC Press LLC

[10] Spong, M. W., Hutchinson, S., Vidyasagar, M. (2006). Robot Modeling and Control. Wiley

[11] Rónai, L., Szabó, T. (2016). Kinematical investigation and regulation of a 4DOF model robot. Acta Mechanica Slovaca, vol 20, no. 3, pp. 50–56, doi: 10.21496/ams.2016.023