

Design and kinematic analysis of spherical robot arm and pneumatic driven gripper

Ziya ÖZÇELİK ^{1*}, Muhammet Mevlüt KARACA ²

¹Department of Mechatronic Engineering, University of Selcuk, Türkiye

²Department of Mechanical Engineering, University of Karabük, Türkiye

Corresponding Author Email: zozcelik@selcuk.edu.tr

(Received: 01 September 2024, Accepted: 18 September 2024)

ATIF/REFERENCE: Özçelik, Z. & Karaca, M. M. (2024). Design and kinematic analysis of spherical robot arm and pneumatic driven gripper. *International Journal of Advanced Natural Sciences and Engineering Researches*, 8(8), 159-180.

Abstract – In this study, a global robotic arm was investigated. A pneumatic-controlled end-effector (gripper) design was developed for the gripping component of the robotic arm. The assembly of the completed robotic arm was simulated using a CAD program, addressing assembly conflicts and dimensional errors. Mechanical and electrical assemblies were carried out, and the connections to the microcontroller were established, rendering the robotic arm operational. To determine the coordinate values of the end effector and for the forward kinematic calculations of the robotic arm, the Denavit-Hartenberg method was employed. Inverse kinematic calculations were performed to compute the joint angle values necessary for the end effector to reach target coordinate values. The torque for each joint of the robotic arm was obtained using the Lagrange-Euler method, along with the overall mass matrix of the robotic arm, the inertia tensor of the links, the gravitational acceleration, and the Coriolis and centripetal force vectors. The dynamic equations derived from the Lagrange-Euler method and the trajectory planning equations were defined in the Matlab-Simulink environment. Graphs were generated in the Simulink section. The solid model designed in SolidWorks was imported into Matlab-Sim Mechanics for visual simulation. The kinematic equations and planned trajectory equations were integrated into the program. The equations within the program facilitated the position control of the robotic arm. The robotic arm underwent tests for repeatability and accuracy following the establishment of position control. Subsequently, the microcontroller connections were made to verify the precision of the end effector's coordinate values.

Keywords – Spherical Robot, Gripper Design, Robot Kinematics, Robot Dynamics, Trajectory Planning, Industrial Robotic Arm.

I. INTRODUCTION

The indispensable component of Industry 4.0, where the Internet of Things facilitates communication and smart manufacturing occurs, is industrial robots. Countries seeking to increase their share in the global economy by enhancing the use of industrial robots are preparing themselves for the industry 4.0 revolution. According to the accepted definition by the Robotics Institute of America, a robot is a reprogrammable multifunctional manipulator designed to transport items, parts, tools, or special devices through programmed movements to perform various tasks (Siciliano, Sciavicco, Villani, and Oriolo, 2009). The most comprehensive definition of an industrial robotic arm, according to the ISO 8373

standard, describes it as an automatic or controlled, reprogrammable, multi-purpose manipulator that can be fixed or mobile and programmable on three or more axes for use in industrial automation applications (“ISO 8373:2012(en), Robots and robotic devices—Vocabulary,” n.d.). During the manufacturing phase of a product, it is necessary to transport it from one location to another to proceed through stages such as production, assembly, packaging, and storage. During the transport phase, the robot moves the object along a predetermined path without subjecting it to any physical changes, with primary applications including palletizing (arranging items in a structured manner on a pallet), feeding parts to machining stations, sorting parts, and packaging (Siciliano et al., 2009). In addition to these applications, the transport between different production lines is facilitated by Automated Guided Vehicles (AGVs), which move along predefined paths, as illustrated in Figure 3.6 (Micieta et al., 2015). The workspace of a robotic manipulator is defined as the set of points that can be reached by the end effector (Cao, Lu, Li, and Zang, 2011). Any modification made to an object (i.e., a physical model change) or changes in the environment alter the motion of the robot within its workspace. The degree of freedom, which measures the mobility of the robotic arm, refers to the number of independent joint variables necessary to determine the position of all the arm’s links in space. This number is equal to the minimum number of joints that need to be actuated to control the system (Briot and Khalil, 2015). To perform a desired task, the robotic arm must have a sufficient degree of freedom. Typically, three degrees of freedom are used to place an object in a desired position in space. If there are more degrees of freedom than task variables, the manipulator is considered unnecessary from a kinematic perspective (Siciliano et al., 2009). Two-fingered or two-claw grippers are considered the most basic robotic end effectors due to their ease of use, simplicity in production, cost-effectiveness, and suitability for various industrial applications. Robots equipped with such end effectors are employed in tasks such as assembly, part pick-and-place operations, and simple manipulations (Samadikhoshkho, Zareinia, and Janabi-Sharifi, 2019). In pneumatic end effectors, the opening and closing actions are achieved using compressed air. Pneumatic end effectors are popular due to their compact size and lightweight, making them easy to incorporate into narrow spaces in the manufacturing industry (“Grippers for Robots,” n.d.). The advantage of the Lagrange-Euler approach lies in its ability to express the final closed-form equations simply in physical terms using terms such as manipulator inertia, gravity, friction, and Coriolis and centripetal forces (Schilling, 2003). In 1955, Denavit and Hartenberg defined matrices showing the transformation and rotational relationships between adjacent joints by placing a coordinate system at each joint (Fu, Gonzalez, and Lee, 1987). A PID control was applied to the manipulator of a Puma-type robot arm with three degrees of freedom. Initially, forward kinematic calculations were performed using the Denavit-Hartenberg method. The inverse kinematic equations were derived from the resulting equations in matrix algebraic form. The dynamic equations were found using the Lagrange-Euler method. A cycloid, harmonic, and polynomial path function were provided as inputs to the system. The analysis revealed that the harmonic path function was more advantageous (Duran and Ankaralı, 2010). First, parameters related to the forward kinematic equations were obtained using the Denavit-Hartenberg method. Subsequently, nonlinear dynamic equations were developed for the robotic arm. Three different applications of PID control were conducted based on a linearization model at the equilibrium point using Matlab/Simulink, and finally, the robot’s feedback linearization was applied. A comparison was made among these three controllers (Mustafa and Al-Saif, 2014).

In this study, the design and fabrication of a two-axis robotic arm capable of tracing a global trajectory have been carried out. Kinematic analyses were conducted using Matlab Simulink. The completed design and assembly of the robotic arm were integrated into Matlab Simulink using the Sim

Mechanics Link from SolidWorks. In the Matlab Sim Mechanics environment, block diagrams for the joints and links of the robotic arm were created. Forward kinematic calculations were performed using the Denavit-Hartenberg method, leading to the derivation of inverse kinematic equations in matrix algebraic form. The dynamic equations were established using the Lagrange-Euler method. While the motion messages and graphical representation of the robotic arm were processed through Matlab, encoder data reading and PID control of the servo motors were conducted via Arduino. For axis movements, the servo motors were driven with PWM signals under PID control using an Arduino Uno microcontroller.

II. MATERIALS AND METHODS

In this study, the aim was to design a pick-and-place robotic arm. In this context, a two-axis robotic arm capable of tracing a global trajectory was created using SolidWorks. Assembly conflicts were resolved during the process.

For motor selection, the Panasonic Minas A5 (MSME204G1G) servo motor was utilized. Considering the inertia generated by this motor, St52 steel was chosen for the body, and the design was developed accordingly. St52 steels, known for their high welding performance, are suitable for the overall structure of the robotic arm. Additionally, the amount of load the robotic arm is expected to lift was calculated while taking into account parameters such as precision, usability, and controllability in the industry. The technical specifications of the AC servo motor and gearbox used in the robotic arm are presented in Tables 1 and 2.

Table 1. Features of AC Servo motor used

Axis	Motor Type	Engine Power (kW)	Velocity (dev/dk)	Motor Torque (N.m)
1 ve 2	Panasonic Minas A5 (MSME204G1G)	2	3000	6,37

Table 2. Gearbox features used

Reducer Type	Reducer Ratio
Planetary Gear Reducer (Apex Dynamics PEII120)	1/25

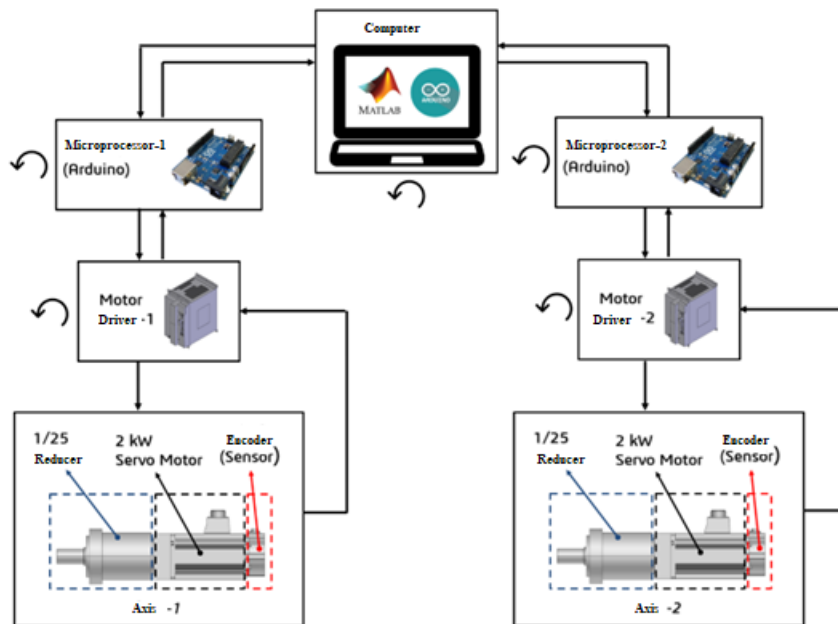


Figure 1 Control chart

An Arduino Uno microprocessor was used to control the motor (Figure 2)

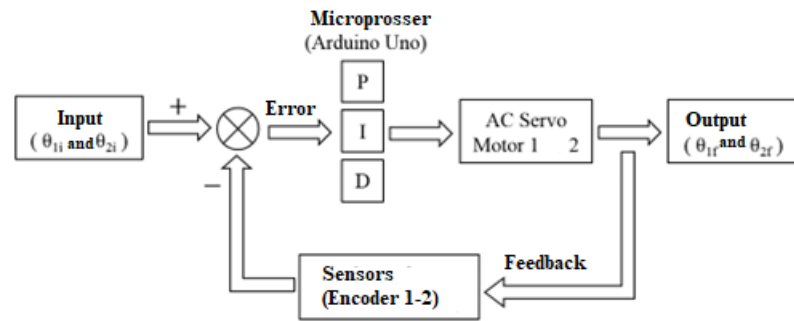


Figure 2 Robot arm feedback block diagram

The location information sent from the computer via the Matlab interface is transmitted to the motor driver with Arduino. After the motor drive positions of the motor, feedback is provided from the encoder to the Arduino. After the encoder information is processed in the Arduino, feedback is given to the Matlab interface on the computer. Incoming information is converted into graphs by storing the data in Matlab.

The dimensions and general appearance of the solid model of the robotic arm used in this study are shown in Figure 3

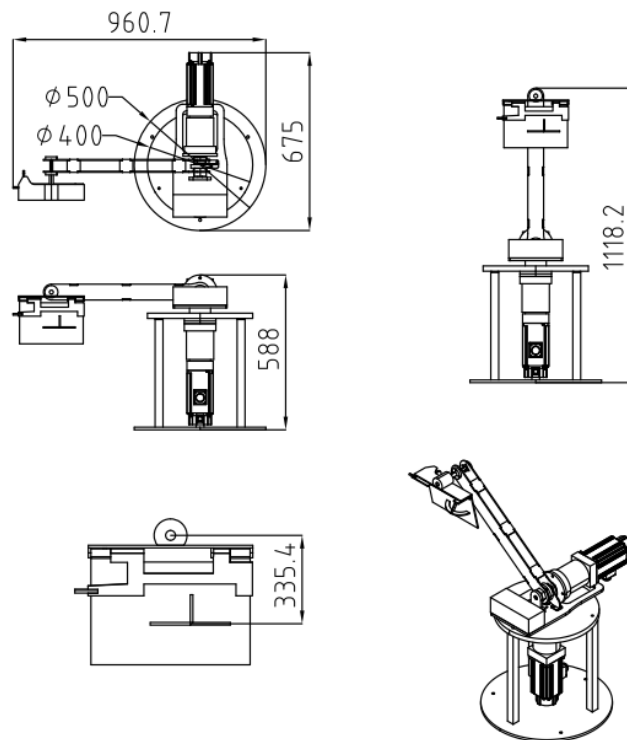


Figure 3 Robot arm

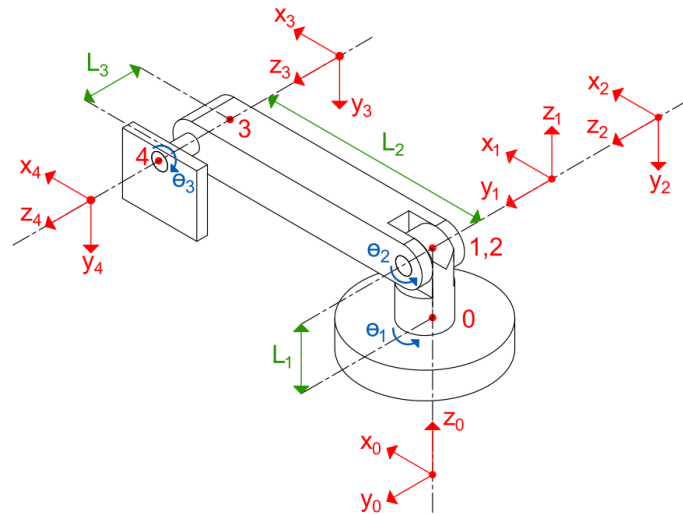


Figure 4 Designed robot arm axis assignments

After the axis assignments of the designed robot arm are made, the D-H parameters are given in Table 3.

Table 3 Table of D-H parameters

Axis No	D-H Variables			
\dot{i}	α_{i-1}	a_{i-1}	d_i	θ_i
1	0°	0	L_1	θ_1
2	-90°	0	0	θ_2
3	0°	L_2	0	0°
4	0°	0	L_3	$\theta_3 = -\theta_2$

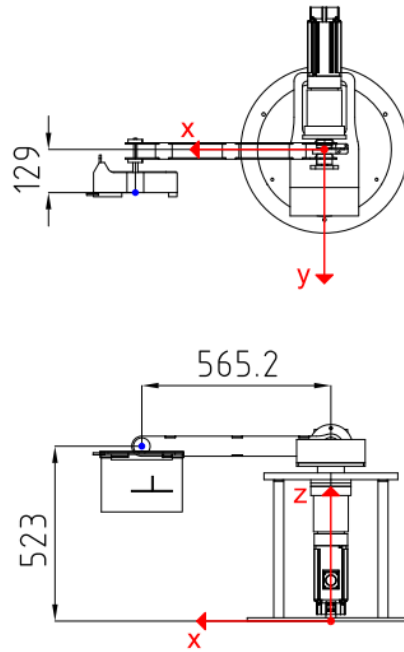
Joint transformation matrices are created.

$${}^0_4T = \begin{bmatrix} \cos\theta_1 & 0 & -\sin(\theta_1) & -\sin(\theta_1)L_3 + \cos(\theta_1)\cos(\theta_2)L_2 \\ \sin\theta_1 & 0 & \cos\theta_1 & \cos(\theta_1)L_3 + \sin(\theta_1)\cos(\theta_2)L_2 \\ 0 & -1 & 0 & -\sin(\theta_2)L_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The position vectors in the 0_4T transformation matrix is constructed as shown in Equation 1 with their elements in the third column of the matrix in Equation 2.

$$\begin{aligned} P_x &= -\sin(\theta_1)L_3 + \cos(\theta_1)\cos(\theta_2)L_2 \\ P_y &= \cos(\theta_1)L_3 + \sin(\theta_1)\cos(\theta_2)L_2 \\ P_z &= -\sin(\theta_2)L_2 + L_1 \end{aligned} \quad (2)$$

Position control: The length parameters on the robot arm are $L_1=523$ mm, $L_2=565.2$ mm, $L_3=129$ mm and $L_4=181.5$ mm, respectively. By giving 0° to all angles, the zero position of the robot is obtained. Figure 5 shows the dimensions of the tip retainer element when 0° is given to the first and second axis in the solid modeling program. These dimensions are calculated with reference to the base point of the robot arm. The forward kinematics are correct because the positions P_x , P_y and P_z of the end function of the robot arm coincide with the specifications shown in Figure 5 and the values calculated in Equation 3.


 Figure 5 Robot arm starting position ($\theta_1 = \theta_2 = 0$)

$$\begin{aligned}
 \theta_1 &= \theta_2 = 0 \\
 P_x &= -\sin(0) L_3 + \cos(0) \cos(0) L_2 = L_2 \\
 P_y &= \cos(0) L_3 + \sin(0) \cos(0) L_2 = L_3 \\
 P_z &= -\sin(0) L_2 + L_1 = L_1 \\
 (P_x, P_y, P_z) &= (565.2, 129, 523)
 \end{aligned} \tag{3}$$

To determine the positive and negative direction of rotation of the robot arm, if -30 degrees are given to the first axis and -60 degrees to the second axis, for example, the position values in Equation 4 are obtained.

$$\begin{aligned}
 \theta_1 &= 30^\circ, \theta_2 = 90^\circ \\
 P_x &= -\sin(-\pi/6) L_3 + \cos(-\pi/6) \cos(-\pi/3) L_2 = 309.2 \\
 P_y &= \cos(-\pi/6) L_3 + \sin(-\pi/6) \cos(-\pi/3) L_2 = -29.6 \\
 P_z &= -\sin\left(-\frac{\pi}{3}\right) L_2 + L_1 = 1012.4 \\
 (P_x, P_y, P_z) &= (309.2, -29.6, 1012.4)
 \end{aligned} \tag{4}$$

As a result of the values derived from the advanced kinematic equations, the position of the robot arm is visually shown in Figure 6. In the advanced kinematics of the robot arm, the axis set thrown at the base point is taken as reference. The directions of rotation are shown in Figure 7 For the first and second axis, it is seen that the counterclockwise is positive.

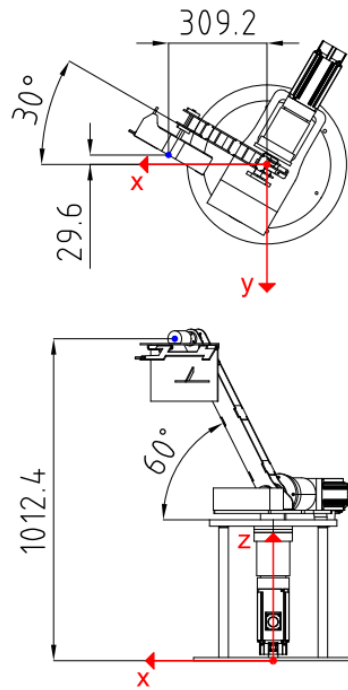


Figure 6 Position on the gripper given -30° to the first axis and -60° to the second axis

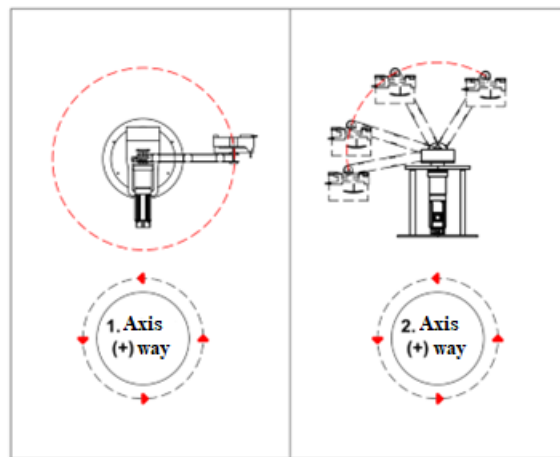


Figure 7 Axes positive directions of rotation

Inverse kinematic analytical solution: The advanced kinematics of the robot arm with six degrees of freedom is shown in Equation 5. Equation 6 shows the matrix containing the position and direction information of its matrix 0_6T .

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T \quad (5)$$

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

The main trigonometric formulas used in inverse kinematics are:

$$\cos \theta = a \text{ ise } \theta = \text{Atan2}(\sqrt{1 - a^2}, a) \tag{7}$$

$$\sin \theta = a \text{ ise } \theta = \text{arctan2}(a, \pm\sqrt{1 - a^2}) \tag{8}$$

$$\cos \theta = a \text{ ve } \sin \theta = b \text{ ise } \theta = \text{arctan2}b, a \tag{9}$$

$$a \sin \theta + b \cos \theta = 0 \text{ ise}$$

$$\theta = \text{arctan2}(-b, a) \text{ veya } \text{arctan2}(b, -a) \tag{10}$$

$$a \sin \theta + b \cos \theta = c \text{ ise}$$

$$\theta = \text{arctan2}(a, b) + \text{arctan}(\pm\sqrt{a^2 + b^2 - c^2}, c) \tag{11}$$

The inverse of the first joint is multiplied by the matrix representing the forward direction kinematics.

$${}^0_1T^{-1} \cdot {}^0_4T = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & -L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_1 r_{11} + \sin\theta_1 r_{21} & \cos\theta_1 r_{12} + \sin\theta_1 r_{22} & \cdot & \cdot \\ -\sin\theta_1 r_{11} + \cos\theta_1 r_{21} & -\sin\theta_1 r_{12} + \cos\theta_1 r_{22} & \cdot & \cdot \\ r_{31} & r_{32} & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot \\ \cdot & \cdot & \cos\theta_1 r_{13} + \sin\theta_1 r_{23} & \cos\theta_1 p_x + \sin\theta_1 p_y \\ \cdot & \cdot & -\sin\theta_1 r_{13} + \cos\theta_1 r_{23} & -\sin\theta_1 p_x + \cos\theta_1 p_y \\ \cdot & \cdot & r_{33} & p_z - L_1 \\ \cdot & \cdot & 0 & 1 \end{bmatrix} \tag{12}$$

The product of the transformation matrices is shown in equation 13

$${}^1_2T \cdot {}^2_3T \cdot {}^3_4T = \begin{bmatrix} \cos\theta_2^2 + \sin\theta_2^2 & 0 & 0 & \cos\theta_2 L_2 \\ 0 & 0 & 1 & L_3 \\ 0 & -\sin\theta_2^2 - \cos\theta_2^2 & 0 & -\sin\theta_2 L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

The equalization of the elements in the third column of matrices in these equations is shown in Equations 14-15-16. Using these synchronizations, angle values are arrived at from the given position values Px, Py, and Pz.

$$\cos\theta_1 p_x + \sin\theta_1 p_y = \cos\theta_2 L_2 \tag{14}$$

$$-\sin\theta_1 p_x + \cos\theta_1 p_y = L_3 \tag{15}$$

$$p_z - L_1 = -\sin\theta_2 L_2 \tag{16}$$

The process of squaring both sides of the equation in Equation 14 is shown in Equation 17.

$$(\cos\theta_1 p_x + \sin\theta_1 p_y)^2 + (-\sin\theta_1 p_x + \cos\theta_1 p_y)^2 = (\cos\theta_2 L_2)^2 + (L_3)^2 \tag{17}$$

If the expression in Equation 19 is simplified and written instead of L2 and L3, Equation 18 is obtained.

$$(p_x)^2 + (p_y)^2 = (3.1945104 * 10^5) \cos\theta_2^2 + 16641 \tag{18}$$

The equation in Equation 8 is solved by applying the trigonometric expression in Equation 10. Thus, given the position values Px and Py, the amount of rotation required for the second axis (θ_2) is determined.

$$a = \cos \theta_2 = \sqrt{\frac{(p_x)^2 + (p_y)^2 - 16641}{3.1945104 * 10^5}}$$

$$\theta_2 = \text{Atan2}(\sqrt{1 - a^2}, a) \tag{19}$$

The amount of angular rotation of the first axis can be obtained from Equation 15. In the solution of this equation, the trigonometric expression in Equation 7 is applied. Equation 20 illustrates the solution steps.

$$-\sin\theta_1 p_x + \cos\theta_1 p_y = L_3 = 129 = c$$

$$a = P_x, b = P_y$$

$$\theta_1 = \text{arctan2}(a, b) + \text{arctan}(\pm\sqrt{a^2 + b^2 - c^2}, c) \tag{20}$$

After obtaining the inverse kinematic equations (Equation 19 and Equation 20) of the robot arm, three Px, Py and Pz positions were given and angle values were found to test its accuracy as shown in Table 4. There are two possibilities in the angle values, and the correct probability is shown in bold format.

Table Hata! Belgede belirtilen stilde metne rastlanmadı. Table of D-H parameters

Reverse Kinematics							
	Input			Output			
	P_x	P_y	P_z	θ_1	θ_1	θ_2	θ_2
Trial 1	309,24	-29,58	1012,48	-29,99	-160,00	60,00	-60,00
Trial 2	88,11	410,61	922,66	-84,22	59,99	45,00	-45,00
Trial 3	254,90	437,33	805,60	-105,47	44,99	29,99	-29,99

The system response was observed with some values obtained using the Ziegler-Nichols method. Approximate $k_p=0.0473$, $k_i=1E-4$ and $k_d=1E-4$ values were found for the 1st axis. Approximate $k_p=0.0253$, $k_i=1E-4$ and $k_d=1E-5$ values were found for the 2nd axis. The robot arm, which was designed and assembled, was also integrated from Solidworks to Matlab Simulink using Sim Mechanics Link. Block diagrams of the joints and limbs of the robot arm were created in the Matlab Sim Mechanics environment. The created block diagram is shown in Figure 8.

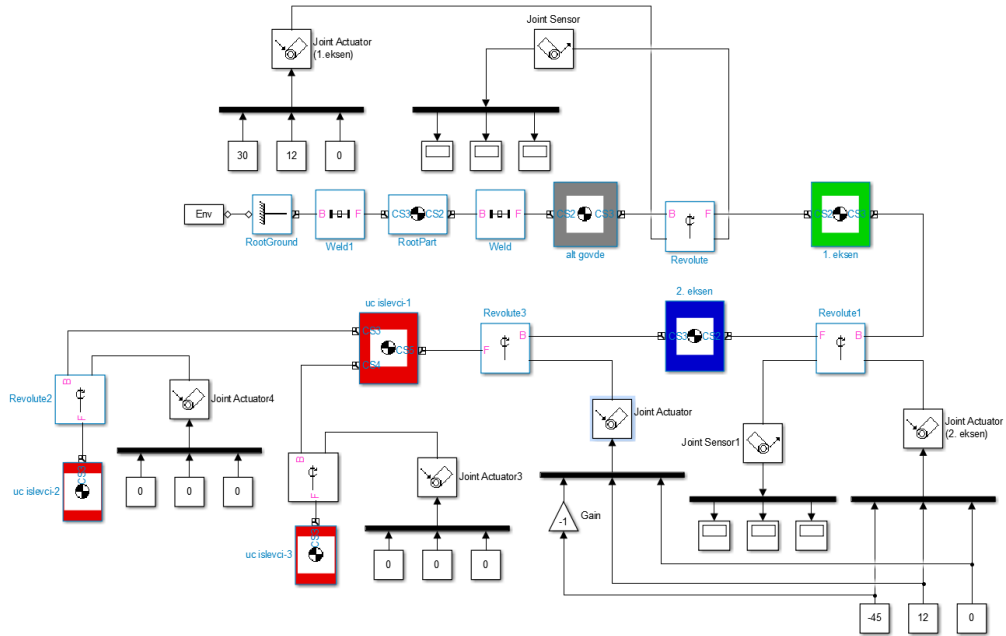


Figure 8 Matlab Simulink block diagram of the robot arm

In the block diagram, a visual simulation of the robot arm was created by driving the joints. The image of the robot arm after running the simulation is shown in Figure 9. The figure is created by giving 30° for the first axis and -45° for the second axis. This simulation was made ready for different studies to be carried out later.

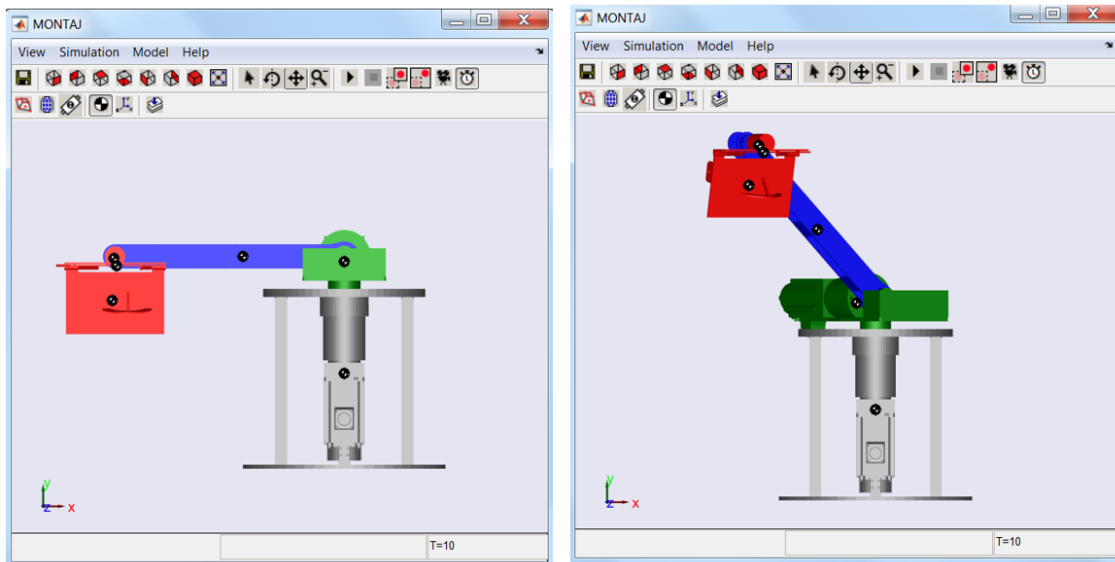


Figure 9: Robot arm simulation

Creation of the dynamic model of the robot arm by the Lagrangian Euler method: The expression due to the movement of the robot arm is shown in Equation 21. The unfolded version of this expression is shown in Equation 22 (Bingül and Küçük, 2017b).

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \tau \quad (21)$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}} \right) - \frac{\partial K}{\partial q} + \frac{\partial P}{\partial q} = \tau \quad (22)$$

If substituted in the equation expressing kinetic and potential energies and friction losses are neglected, the general expression in Equation 23 is obtained (Bingül and Küçük, 2017b).

$$\sum_{j=1}^n D_{ij}(q)\ddot{q}_j + \sum_{k=1}^n \sum_{j=1}^n c_{kj}^i(q)\dot{q}_k + y_i(q) + b_i(q) = \tau_i \quad (23)$$

The first term in Equation 23 represents the acceleration due to the torque and intrinsic forces exerted on the joints during the robot's motion. The second term shown in Equation 24 represents the Coriolis and Centrifugal force vector of the robot arm (Bingül and Küçük, 2017b).

The Coriolis Effect is the deflection that occurs in an object moving on a rotating platform. On a platform with clockwise rotation, the deflection is to the left of the object's movement; In another with a counterclockwise rotation, the deflection is to the right. The Coriolis effect on the robot arm is when one limb rotates over the other rotating limb, creating a deviation or error in the target (Thomas, 2016).

Centrifugal Force is a force that pulls an object out of the center of the curve as it moves in a curved path.

$$c_{kj}^i(q) = \frac{\partial}{\partial q_k} D_{ij}(q) - \frac{1}{2} \frac{\partial}{\partial q_k} D_{kj}(q) \quad 1 \leq i, j, k \leq n \quad (24)$$

The third term in Equation 4.82 symbolizes gravitational acceleration and its unfolded form is shown in Equation 25 (Bingül and Küçük, 2017b).

$$y_i(q) = - \sum_{k=1}^3 \sum_{j=1}^n g_k m_j A_{ki}^j(q) \quad (25)$$

As a result, when friction forces are neglected, the dynamical equation of the robot arm is expressed in Equation 26 (Bingül and Küçük, 2017b).

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau_i \quad (26)$$

Simulation model of the designed robot arm: The mass, center of mass and moment of inertia information of each limb of the designed robot arm were obtained from the mass properties section of the calculate tab in the Solidworks program. The data obtained are shown in Figures 10-11 and 12.

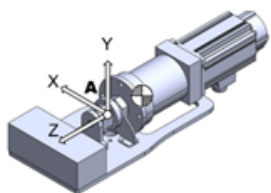
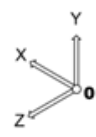
Mass (kg)	
44,59	
Center of mass according to coordinate system 1 (mm)	
X = -0.10994 Y = -101.92403 Z = -12.72789	
Moment of inertia with respect to coordinate system A (g*mm)	
lxx = 2225208720.39875 lxy = 13812.74074 lsx = -323017.25576	
lyx = 13812.74074 lyy = 319164026.05079 lyz = -358428123.44543	
lzx = -323017.25576 lzy = -358428123.44543 lzz = 2050572752.42063	

Figure 10 Mass, center of mass, and moment of inertia of the first axis in the robot arm

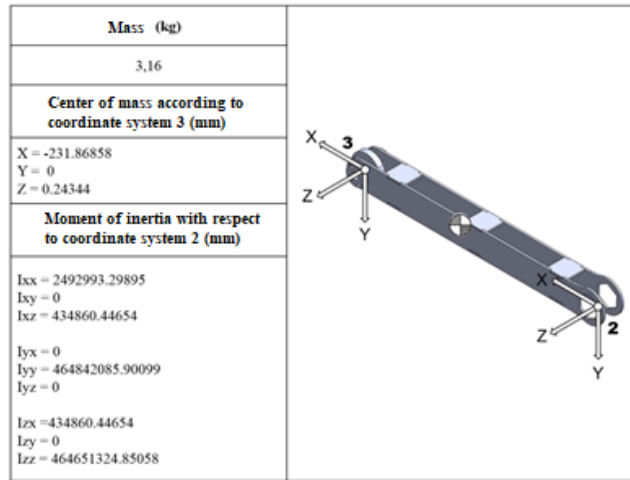


Figure 11 Mass, center of mass, and moment of inertia of the second axis on the robot arm

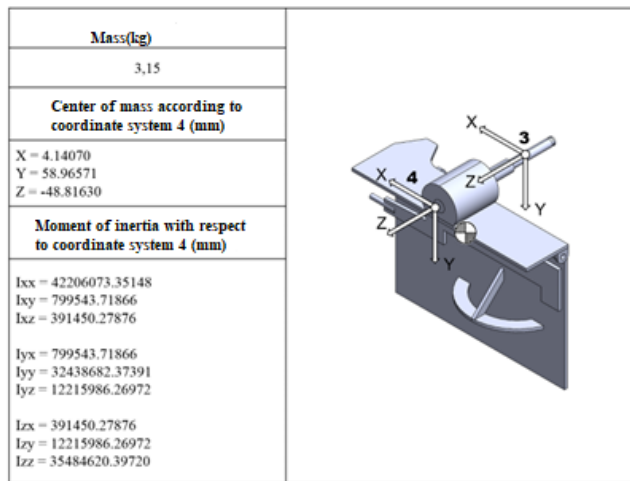


Figure 12 Mass, center of mass, and moment of inertia of the end functioner on the robot arm

The data obtained from the mass properties of the robot arm were used in the Lagrange Euler equation to calculate the general mass matrix of the robot arm, Coriolis and Centrifugal Forces and the torque values acting on the additions. The dynamic simulation model of the robot arm was created with block diagrams as shown in Figure 13 in the Simulink section of the Matlab program.

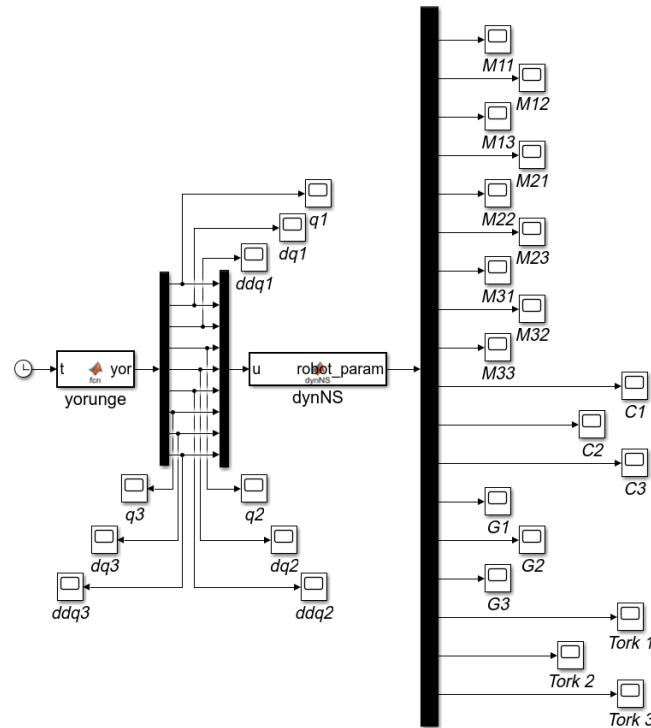


Figure 13 Robot arm Matlab -Simulink simulation model

After the block diagram was created and the codes were entered into the Matlab program, the simulation was run for 10 seconds. As a result of the simulation: Position, velocity and acceleration graphs of the joints are shown in Figures 14-15 and 16, Coriolis and Centrifugal Forces in Figure 17, Gravitational forces are shown in Figure 18, Torque forces in Figure 19. The general mass matrix elements are given in Figure 20.

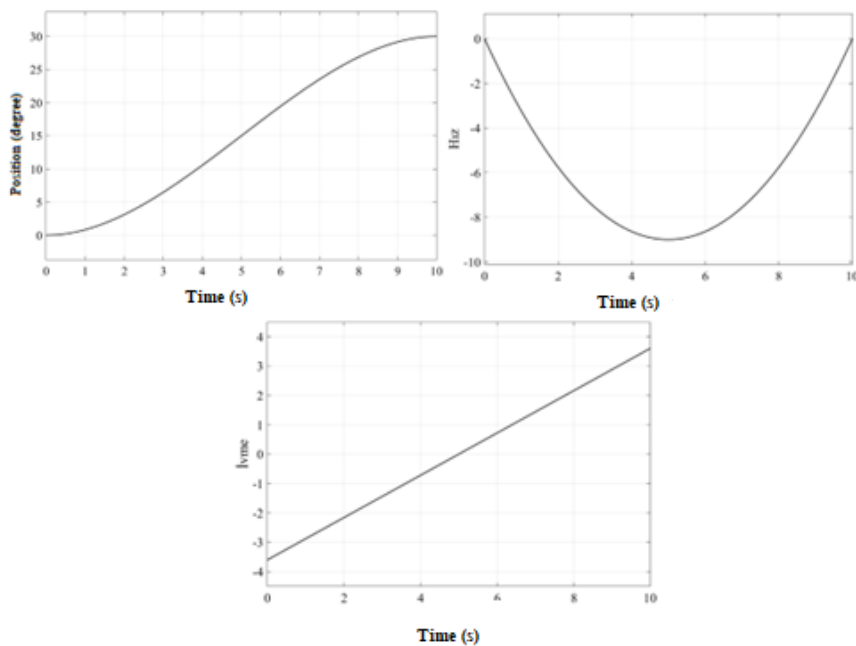


Figure 14. Position, velocity, and acceleration time graphs of the first addition

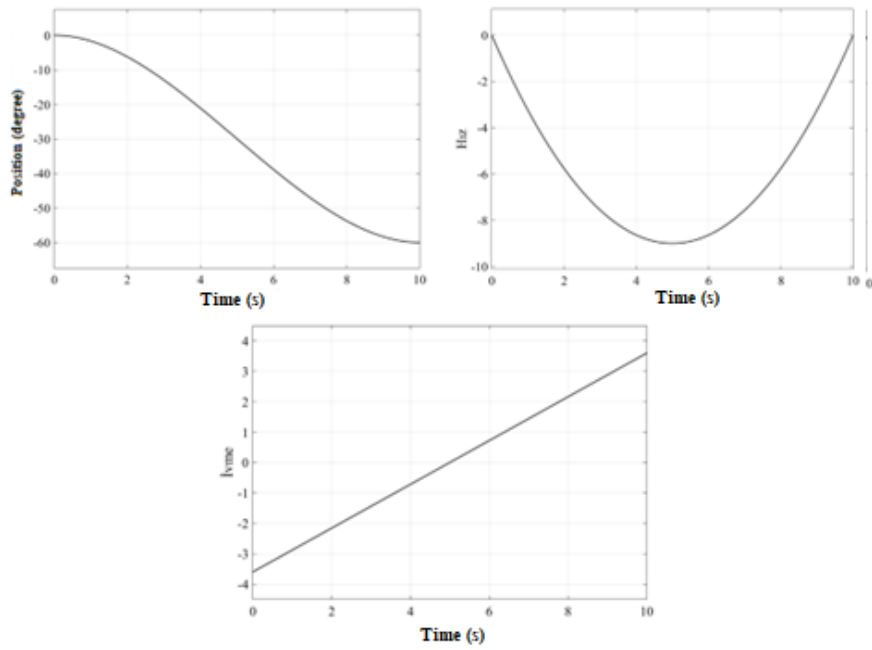


Figure 15. Timegraphs of position, velocity, and acceleration of the second addition

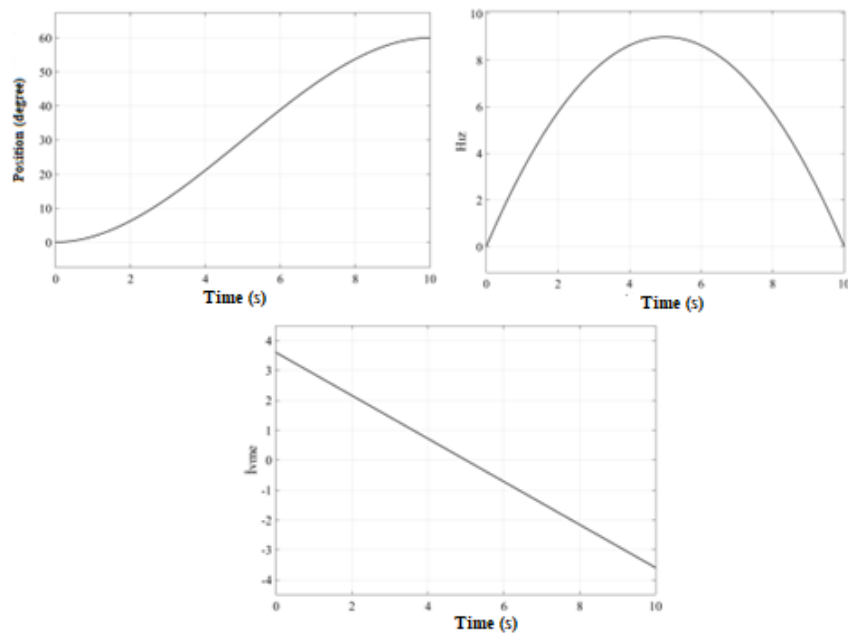


Figure 16. Position, velocity, and acceleration time graphs of the third addition

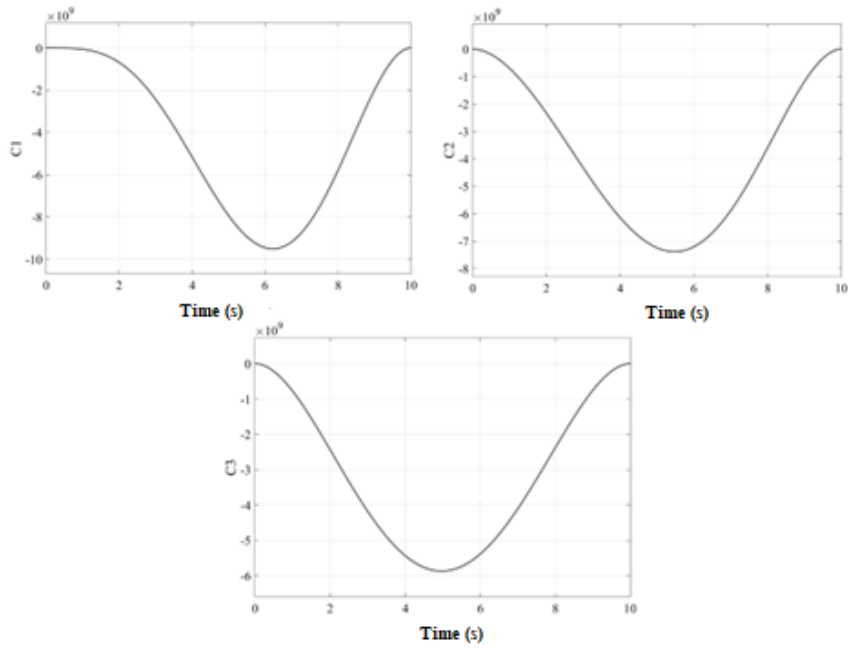


Figure 17 Coriolis and Centrifugal Forces

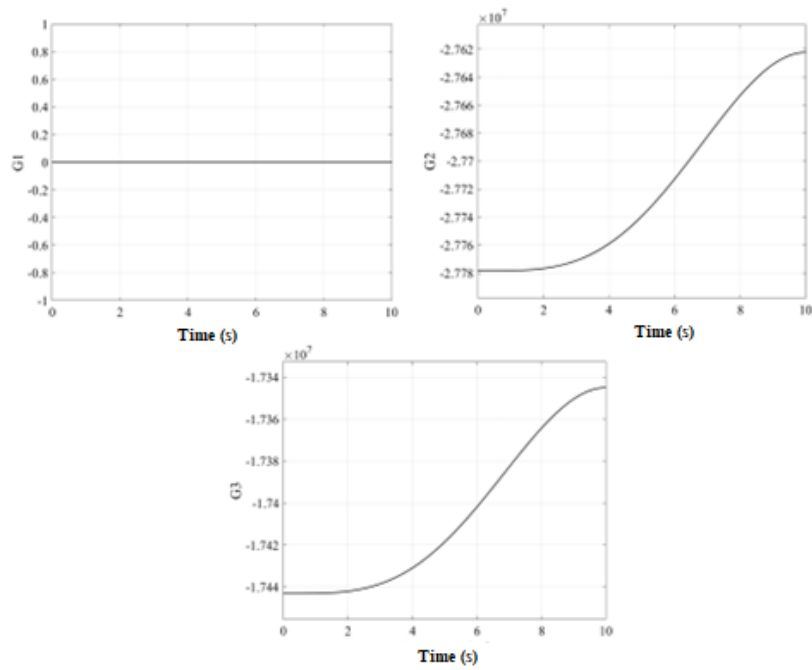


Figure 18. Gravity vectors

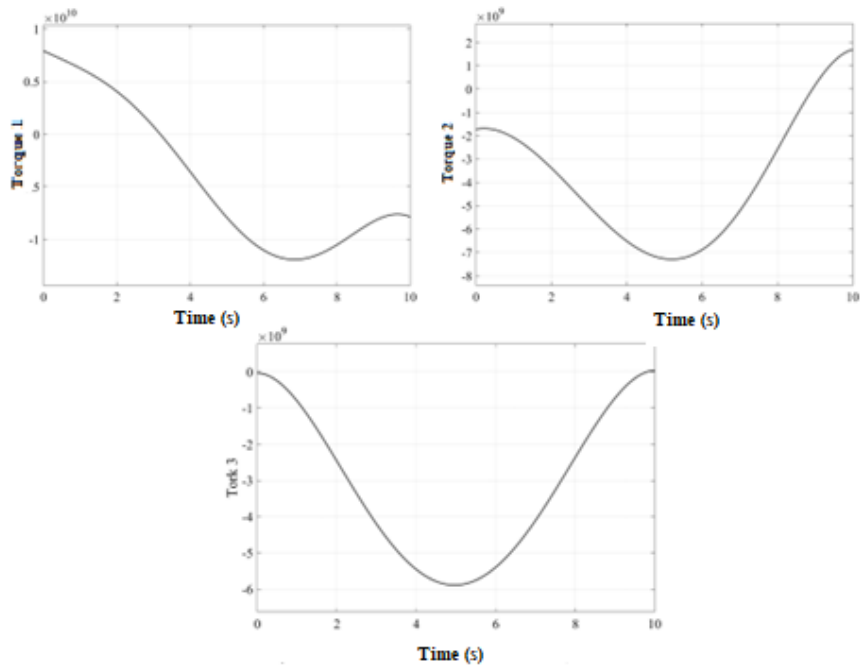


Figure 18 Torque values

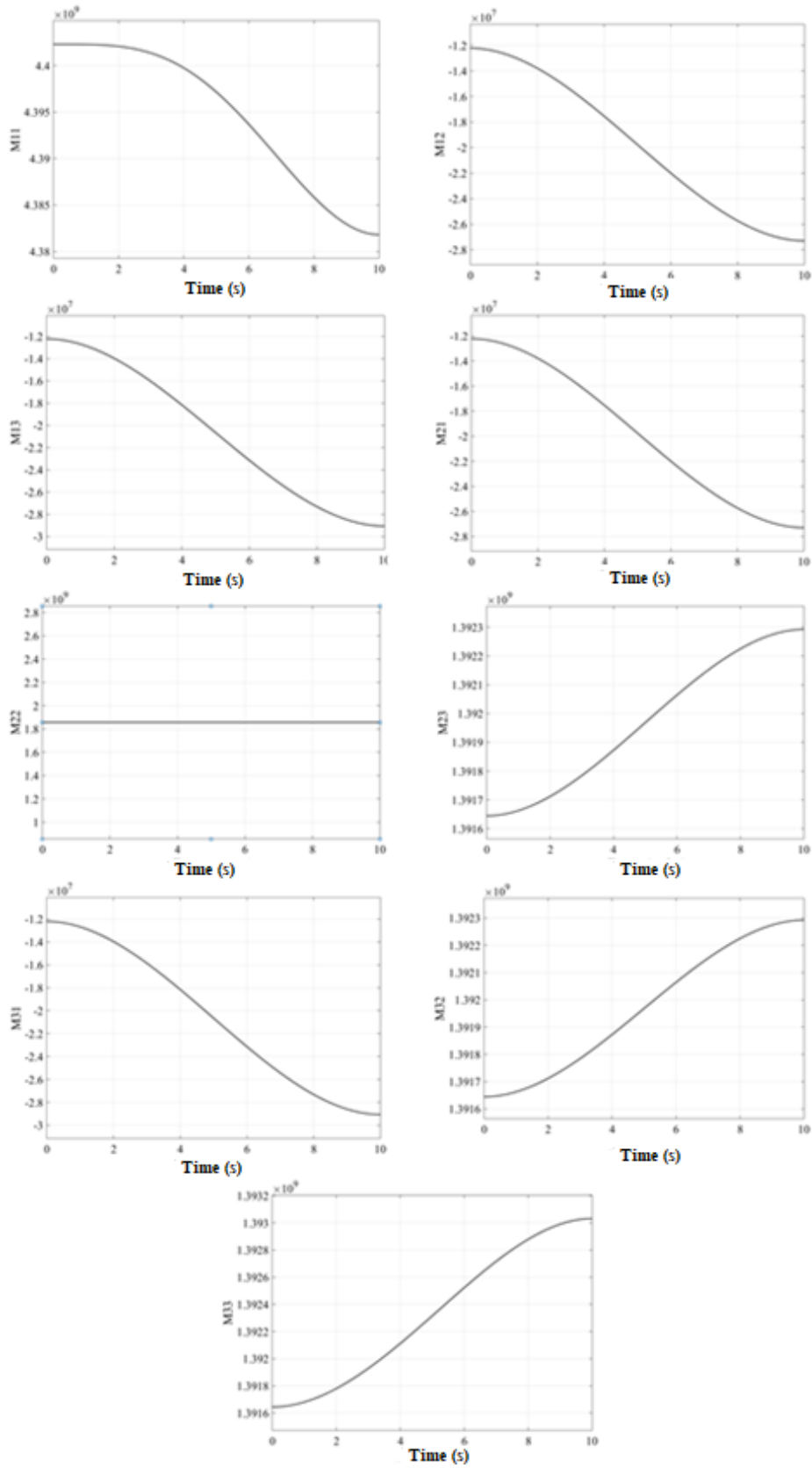


Figure 20. General mass matrix elements

Microprocessor (Arduino Uno) and Matlab Serial Communication: The robot arm was driven by establishing serial communication between Matlab and the Arduino program. While the motion messages and graphics creation of the robot arm were done via Matlab, the encoder data reading and PID control of the servo motors were done via Arduino. The PID control of the motors was created by using the Arduino library. The graphs obtained as a result of driving the robot arm are shown in chapter 21. While creating the graph, the sampling period was determined as 100 ms experimentally based on the processor speed and the accuracy (stability) of the graph. An example of a gesture message is:

When 11 is entered in the message Which axes will move, it will move in two axes.

When 110100015 is entered in the 1st axis motion message, that is, the first axis will make a 15° angle in the positive direction and the graph will be created with a sampling period of 100 ms.

When -110100035 is entered into the 2nd axis motion message, the first axis will make a 35° angle in the negative direction and the graph will be created with a sampling period of 100 ms.

The 14-bit encoder integrated into the rear of the motor driver is set to be 30000 pulses at 1 revolution. When the desired angle values are entered into the system on the robot arm, the axes move at the same time and the axis that reaches the desired angle stops. To test the robot arm, absolute error, relative error rates, and standard deviations were calculated by giving sample angle values.

When the first axis of the robot arm is given -30° and the second axis is given -60°, the graphs of position, velocity, acceleration and position time graphs of the end functionor are given in Figure 21.

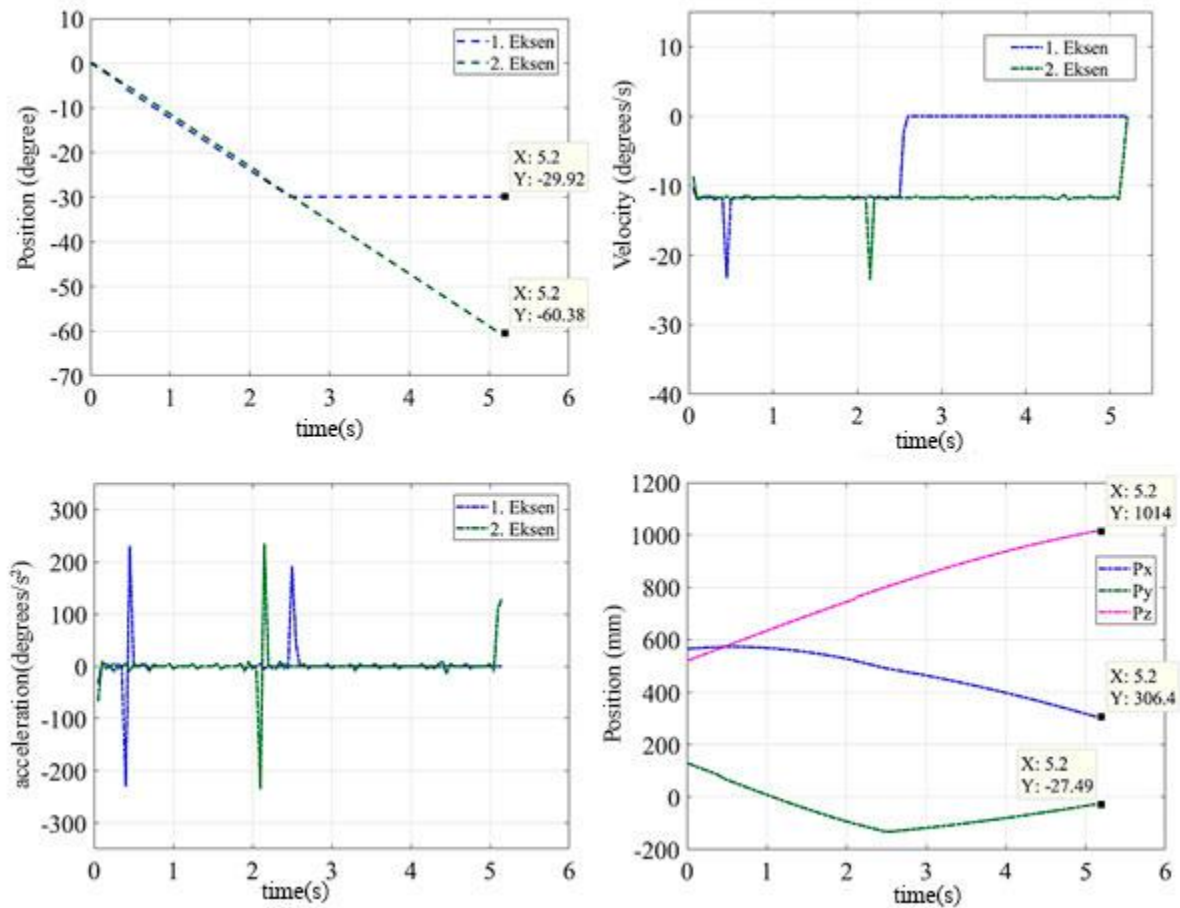


Figure 21. Position, velocity, acceleration and position time graphs of the end functioner in the -30° and -60° operating range

When the position time graph was examined, the absolute error values were calculated as (0.08) and (0.38) on the first and second axis, respectively, and the relative error values were calculated as (-0.267%) and (-0.629%), respectively. When the speed time graph is examined, it is seen that the robot arm peaks at some points and moves at an average speed of 11.76 degrees/second on both axes. When the

acceleration time graph was examined, it was seen that although the robot arm was moving at a constant speed, it peaked at some points. Intermittent peaks in the charts can be caused by the following reasons:

- Bedding problem
- Stability problem in design
- Interference caused by the circuit
- Tooth gap in the gearbox used

When the position time graph of the end functioner is examined, the absolute error values in the x, y and z axes are (2.838), (2.092) and (1.523), respectively, while the relative error values are (0.926%), (-7.91%) and (0.15%).

When the first axis of the robot arm is given 37° and the second axis is given -43° , the graphs formed are given in Figure 22, respectively, the position, velocity, acceleration and position time graphs of the end functioner.

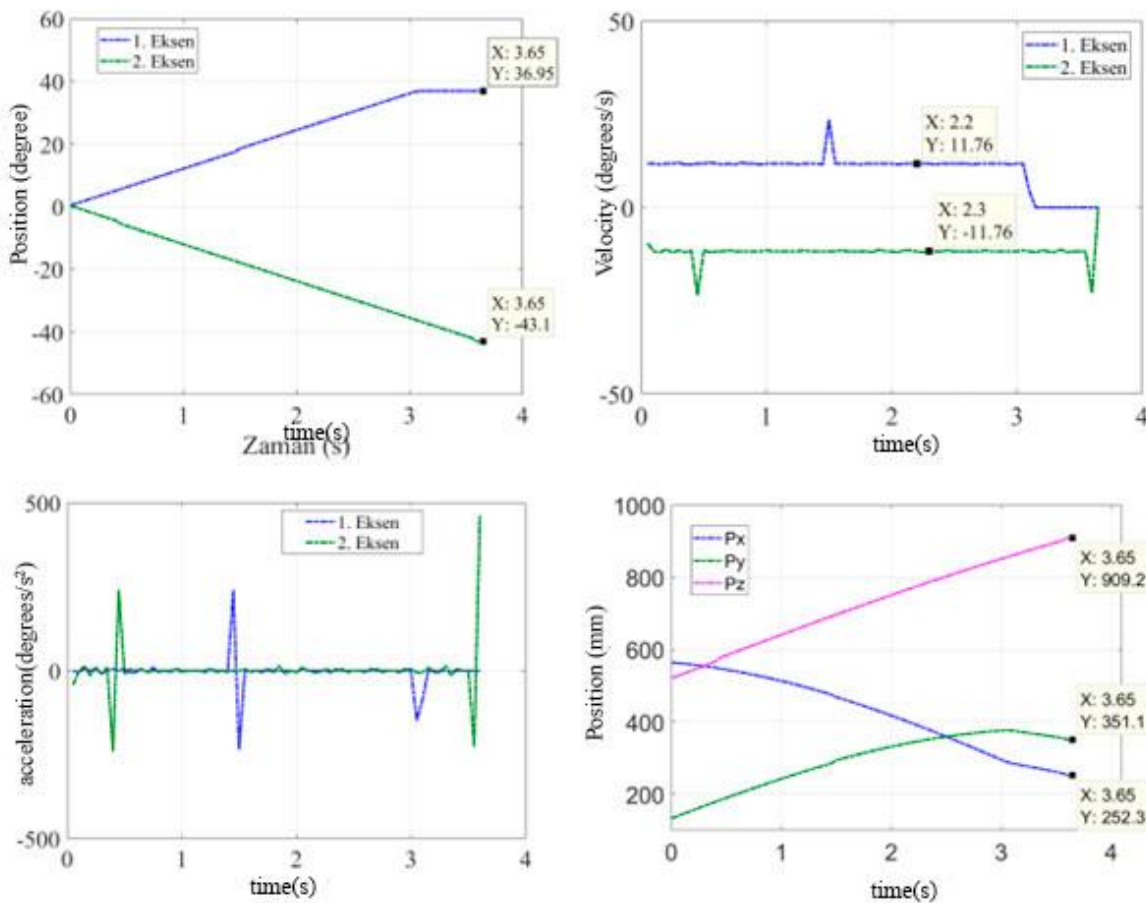


Figure 22 Data from the encoder given 37° to the first axis and -43° to the second axis

When the position time graph was examined, the absolute error values were calculated as (0.05) and (0.1) on the first and second axis, respectively, and the relative error values were calculated as (0.135%) and (-0.232%), respectively. When the speed time graph is examined, the robot arm moves at an average speed of 11.76 degrees/second in both axes. When the acceleration time graph was examined, it was seen that although the robot arm was moving at a constant speed, it peaked at some points. The intermittent peaks that occur in the graphs are also based on the reasons in the previous graph. When the position time graph of the end functioner is examined, the absolute error values in the x, y and z axes are (0.19), (0.69) and (0.735), respectively, while the relative error values are (0.075%), (0.197%) and (0.081%).

The position time graph that occurs when the robot arm is given values of 45° to the first axis and -30° to the second axis, and this process is set to the number of repetitions of 10, is given in Figure 23.

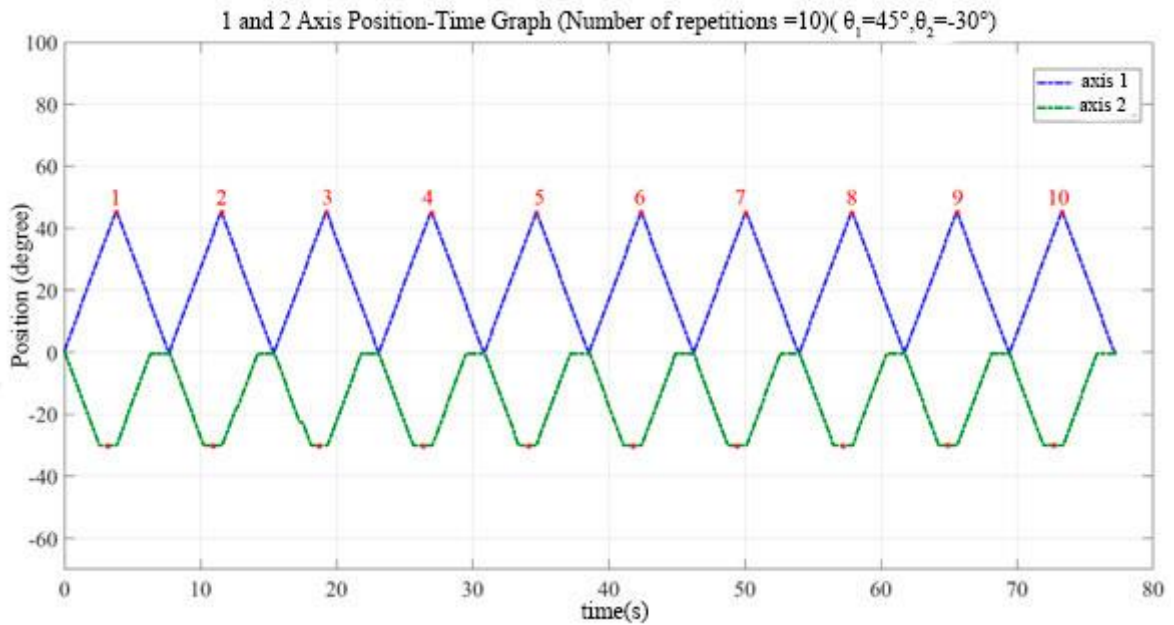


Figure 23. First and second axis position-time graph (Number of Repetitions = 10) ($\theta_1 = 45^\circ, \theta_2 = -30^\circ$)

The position time graph that occurs when the robot arm is given values of 45° to the first axis and -45° to the second axis and this process is set to the number of repetitions of 10 is given in Figure 24.

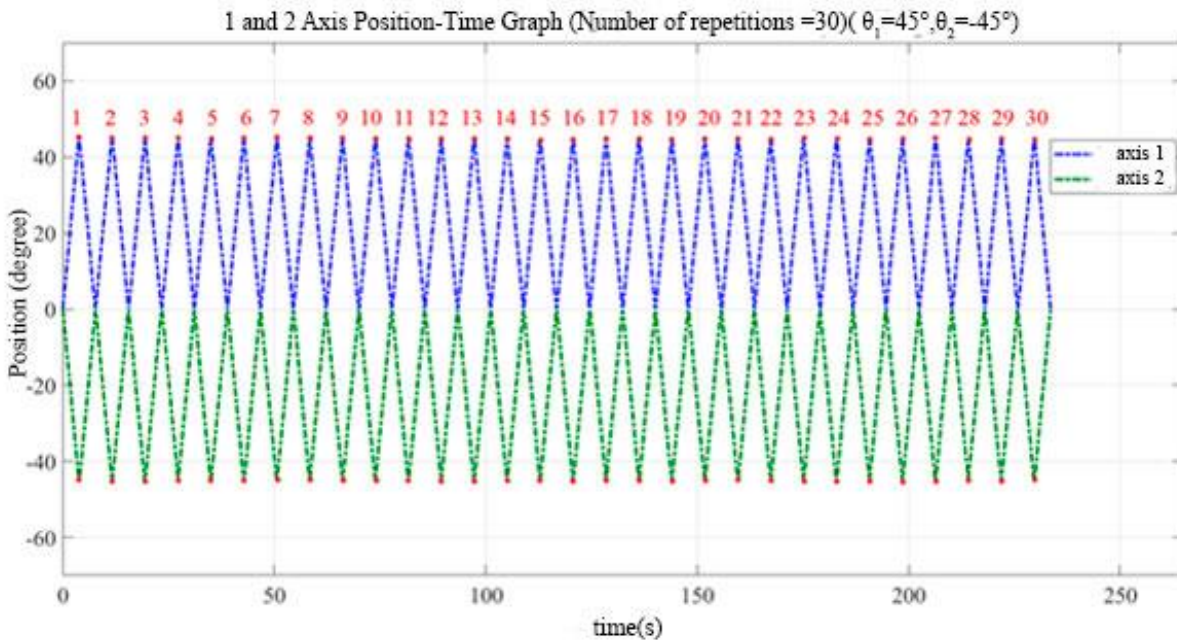


Figure 24 First and second axis position-time graph (Number of Repetitions = 30) ($\theta_1 = 45^\circ, \theta_2 = -45^\circ$)

III. CONCLUSIONS

In this study, the design and production of a two-axis, spherical trajectory tracking robotic arm was carried out. The coordinates at which the robotic arm will move were obtained and controlled by forward and inverse kinematic calculations. These calculations were verified by sample angle values. The resulting kinematic equations were embedded in the Matlab interface. Sim Mechanics was used in Matlab Simulink. The dynamical equations found by the Lagrange-Euler method and the equations created by trajectory planning were defined in the Matlab-Simulink program. Block diagrams of the joints and limbs

of the robot arm were created in the Matlab Sim Mechanics environment. After the block diagram was created and the codes were entered into the Matlab program, the simulation was run for 10 seconds. As a result of the simulation, Position, velocity and acceleration graphs of the joints, Coriolis and Centrifugal Forces, Gravitational forces, Torque forces, General mass matrix elements were obtained graphically.

During the trial phase of the designed robotic arm, it is important to note that the deviation and error values of the sample angle values, and the repetitive movements, process or operation do not require extremely high precision. However, based on the deviation values in the graphs, some manufacturing defects were observed in the experimental setup. These errors and deviations have led to the conviction that this robot arm is suitable for rough work. In the next study, these will be addressed with design corrections and manufacturing corrections. In addition, by selecting the appropriate PID values and adjusting the engine characteristics, deviations in the trajectory can be optimized. Using the Ziegler-Nichols method, approximate k_p , k_i and k_d values were found by using the most appropriate PID values.

The robotic arm and end effector are designed considering the different materials as well. With robotic arm, it can be used for different purposes by using different end effectors. In this study, the BAP research project was supported. In the next study, trajectory optimization will be studied, and image processing and artificial intelligence will be used.

IV. REFERENCES

- [1]. Ben-Ari, M., & Mondada, F. (2017). Elements of robotics. Springer Nature.
- [2]. Bejczy, A. K., & Paul, R. P. (1981, December). Simplified robot arm dynamics for control. In 1981 20th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes (pp. 261-262). IEEE.
- [3]. Bernier, C. (t.y.). How Pneumatic End Effectors Work? Robotiq. <https://blog.robotiq.com/bid/65604/How-Pneumatic-End-Effectors-Work>
- [4]. Bingül, Z. ve Küçük, S. (2017a). Robot Kinematığı (3.). Umuttepe Yayınları.
- [5]. Bingül, Z. ve Küçük, S. (2017b). Robot Dinamiğı ve Kontrolü (2.). Umuttepe Yayınları.
- [6]. Briot, S. ve Khalil, W. (2015). Dynamics Modeling of Parallel Robots. (S. Briot ve W. Khalil, Ed.) Mechanisms and Machine Science. Cham: Springer International Publishing. doi:10.1007/978-3-319-19788-3_8
- [7]. Cao, Y., Lu, K., Li, X., & Zang, Y. (2011). Accurate numerical methods for computing 2d and 3d robot workspace. International Journal of Advanced Robotic Systems, 8(6), 76.
- [8]. Chen, W. H., Chen, C. P., Tsai, J. S., Yang, J., & Lin, P. C. (2013). Design and implementation of a ball-driven omnidirectional spherical robot. Mechanism and Machine Theory, 68, 35-48.
- [9]. Diouf, A., Belzile, B., Saad, M., & St-Onge, D. (2024). Spherical rolling robots—Design, modeling, and control: A systematic literature review. Robotics and Autonomous Systems, 104657.
- [10]. Duran, M. A. ve Ankaralı, A. (2010). Üç Serbestlik Dereceli Puma Tipi Bir Manipülâtörün PID Kontrolü. Selçuk-Teknik Dergisi, 9(1), 79-98.
- [11]. Eren, O. ve Kaftanoğlu, B. (2001). Altı Serbestlik Dereceli Bir Endüstriyel Robotun Tasarımı İmalatı ve Çalıřtırılması. Makina Tasarım ve İmalat Dergisi, 4(2), 103-111.
- [12]. Fu, K. S., Gonzalez, R. C. ve Lee, C. S. G. (1987). Robotics: Control, Sensing, Vision, and Intelligence. McGraw-Hill Book Company.
- [13]. Grippers for Robots. (t.y.).RobotWorx.
- [14]. Hartenberg, R. S. ve Denavit, J. (1955). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, 215-221.
- [15]. Industrial Robots. (t.y.).Distrelec. 17 Şubat 2020, <https://www.distrelec.de/en/industrial-robots/cms/knowhow-industrial-robots>
- [16]. ISO 8373:2012(en), Robots and robotic devices—Vocabulary. (t.y.).
- [17]. Kumar, V. (2010). 50 Years of Robotics [From the Guest Editors]. IEEE Robotics Automation Magazine, 17(3), 8-8. doi:10.1109/MRA.2010.938493
- [18]. Mustafa, A. M. ve Al-Saif, A. (2014). Modeling, Simulation and Control of 2-R Robot. Global Journals Inc. (USA), 14(1), 49-54.
- [19]. Öztürk, M. (2014). Antropomorfik Robotların Dinamiğı ve Adaptif Kontrol Uygulamaları: Matlab/simulink Modelleme.
- [20]. Samadikhoshkho, Z., Zareinia, K. ve Janabi-Sharifi, F. (2019). A Brief Review on Robotic Grippers Classifications. 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), 1-4. doi:10.1109/CCECE.2019.8861780
- [21]. Schilling, R. J. (2003). Fundamentals of Robotics: Analysis and Control (5.). Prentice Hall of India.
- [22]. Spong, M. W., Hutchinson, S. ve Vidyasagar, M. (2004). Robot Dynamics and Control (2.).
- [23]. Siciliano, B., Sciavicco, L., Villani, L. ve Oriolo, G. (2009). Robotics: Modelling, planning and control. London: Springer.

- [24]. Tonbul, T. ve Sarıtaş, M. (2013). Beş Eksenli Bir Edubot Robot Kolunda Ters Kinematik Hesaplamalar ve Yörünge Planlaması. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 18(1), 145-167.
- [25]. Yavuz, E., Alıcı, M. ve Uyar, E. (2015). 3 Serbestlik Dereceli (3R) Bir Robot Manipülatörünün Kontrolü ve Görüntü İşlemeye Dayalı Nesne Taşınması Control of 3 DOF (3R) Robot Manipulator and Moving Objects Based on Image Processing (ss. 654-659).