JURNAL INFOTEL
VOL. 17, NO. 1, FEBRUARY 2025, PP. 191–209. DOI:10.20895/INFOTEL.V17I1.1257

RESEARCH ARTICLE

An Application of Inverse Kinematics and LQR Control in a 3-DOF Robot Arm

Adri Firmansya Sofyan[1], Erwin Susanto[2,*], Rafsanjani Nurul Irsyad[3], Alfitho Satya Prabaswara[4], and Irham Mulkan Rodiana[5]

[1,2,3,4,5]School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia

*Corresponding email: erwinelektro@telkomuniversity.ac.id

Received: October 15, 2024; Revised: December 31, 2024; Accepted: February 05, 2025.

Abstract: Linear Quadratic Regulator (LQR) is one of the optimal control methods on state space-based systems. The LQR control method is an option to be applied to the 3 DOF robot arm because multi-link systems such as robot arms are basically non-linear with quite complex modeling. Conventional methods in controls often involve trade-offs. These trade-offs are required to obtain optimal stability among the robot arm parameters. System modeling is formulated using the Lagrangian dynamics and Euler-Lagrange method to obtain a nonlinear model of the system and then linearized it using Taylor series expansion. LQR control allows improvements in system performance, including settling time and overshoot by tuning the weight parameters Q and R. The values of the Q and R matrices can be adjusted to obtain a good system response for a particular trajectory. Tuning the Q and R parameters can also improve the stability of the system by reducing overshot but causing the rise time of the system to increase. The results show the waypoint average errors in the end effector on the x, y, z axes are 14.9%, 16,26% and 3.67% respectively. Based on the results, this can be done by tuning the value of Q matrix and the value of R matrix. Then, the tuned parameters can be used to move on a predetermined trajectory.

Keywords: inverse kinematics, LQR, optimal control, 3DOF robot arm

1 Introduction

Common applications of robot arms in industry are widely used for digital manufacturing processes, pick and place, welding, subtractive and additive manufacturing, this shows an example of the versatility of robots and their use for industry [1]. Manipulator robot has at least one link, where flexibility and reachability can be increased based on the needs and application [2]. The 3 DOF arm robot type is the arm robot with the minimum DOF that is commonly used because it can move across 3 dimensions.

As the DOF increases, the stability of a robot arm system will decrease, which can reduce performance and make the system unstable. Control methods can be used to improve the stability of the system [3]. There are several choices of control methods commonly used in robot arm manipulators with their respective advantages and disadvantages. For example, PID controllers [4, 5] are easy to model but have a complicated tuning process, fuzzy logic controllers [6] does not require mathematical modeling but takes a long time to calculate the control action, and model predictive controllers [7] has good performance in simulation, but is difficult to scale to higher DOF.

LQR control method is one of the optimal control methods on state space based system [8]. The LQR controller takes into account each dynamic state space as well as control inputs to make an ideal decision [9]. LQR focuses on the mathematical optimization of an objective cost function for a dynamic system that has multiple inputs and multiple outputs (MIMO) [3], and is expected to be more suitable and reliable for use in multi-link systems such as the 3 DOF robot arm. The performance of the LQR method is highly dependent on the values of the Q and R matrices, which can take a long time to set correctly and can be difficult without knowledge of the system [10].

This paper describes the modeling and implementation of the LQR control methods on a 3 DOF robotic arm System modeling starts with finding the angular position of each joint. It's calculated using inverse kinematics with the end-effector position as input [11]. System modeling uses Lagrangian dynamics to obtain the relationship between force and movement in the robot arm [12]. The non-linear system is then linearized using Taylor series expansion by ignoring the high order term to obtain a linear model of the system [13]. The modeling in this paper combines three methods of [11–13] to obtain a state space equation which is a mathematical representation of the physical model of the 3 DOF robot arm that shows the relation between inputs, outputs and state variables of the system. The system performance is then tested by observing the system's step response based on the values of Q and R matrices. The results of observation are then used to adjust the Q and R matrices to maintain the stability of the system in performing pick and place movements.

2 Research Method

The research methods in this paper include the design of the LQR system and control method used to regulate the performance of the robot arm trajectory using the principles of inverse kinematics and inverse dynamics.

2.1 System Design

The robot arm that will be used consists of 3 joints to move in a spherical workspace with joint 1 at the base for horizontal movement and joints 2 and 3 at links 2 and 3 for vertical movement.

Each link of the 3 DOF robot arm mass, length and center of gravity can be seen in Table 1 as follows.

LQR control will provide a gain matrix based on the given Q and R parameters. The gain matrix will be calculated using MATLAB. The gain matrix obtained is then sent into

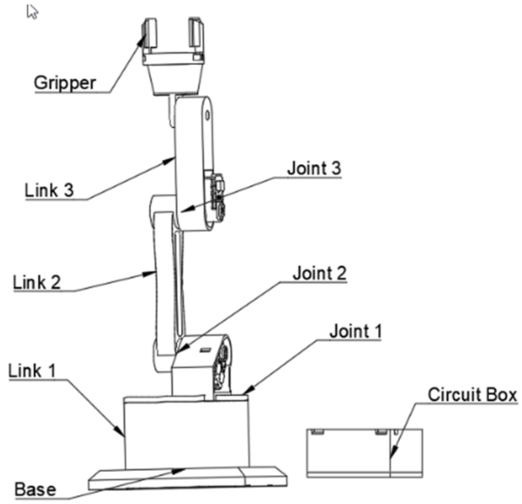


Figure 1: 3 DOF robot arm design.

Table 1: Parameters of the 3 DOF robot arm

Parameter	**Value**
l_1	$0,133m$
l_2	$0,150m$
l_3	$0,200m$
m_1	$0,1495kg$
m_2	$0,0520kg$
m_3	$0,2820kg$
pl_2	$0,0750m$
ql_3	$0,1066m$

the Arduino IDE along with the reference position to be processed on the microcontroller to move the plant. Feedback from the plant is in the form of the current angular position obtained from the position sensor on the joint actuator. The following is a block diagram to show the input and output of this system.

2.2 LQR Control

LQR control is designed using quadratic index performance using state matrix x and input control u. The quadratic linear term shows the quadratic cost function and linear system dynamics which aims to determine the gain k that minimizes the cost function [14]. The equation of the cost function is as follows.

$$J = \int_0^\infty \left[x^T Q x + u^T R u\right] dt \tag{1}$$

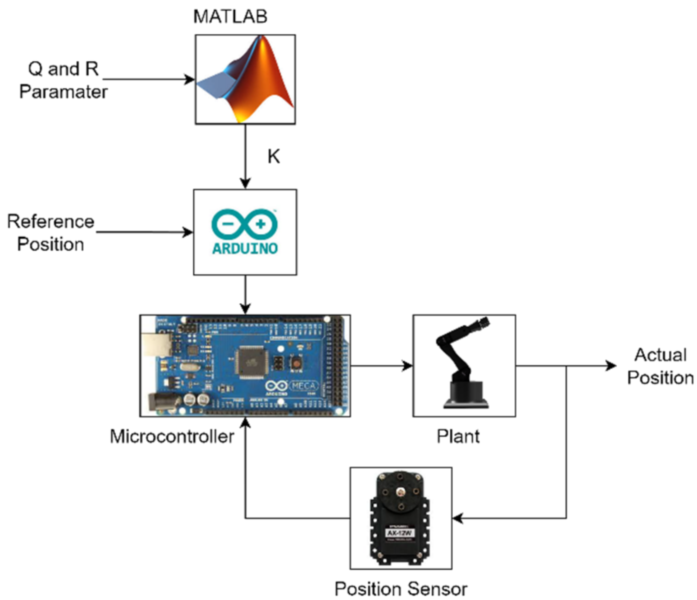


Figure 2: Block diagram of the system.

The LQR cost function is formulated as J, while Q and R are weighing matrices. Q and R are each semi positive definite matrices [9]. Based on optimal control theory, the LQR equation is obtained as follows.

$$u = -R^{-1}B^{T}Px \tag{2}$$

Where u and x are the input vector and state vector of the system respectively. B is the input matrix, dan P is the solution of the following Algebraic Riccati Equation (ARE),

$$PA + A^{T}P + Q - BPR^{-1}B^{T}P = 0 \tag{3}$$

Equation 2 can be simplified to

$$u = -Kx \tag{4}$$

where

$$K = R^{-1}B^{T}P \tag{5}$$

is the gain matrix for the optimal control that will be used on the plant. The optimal cost function can be obtained by changing the values of Q and R parameters. In theory, the effects of parameters Q and R on the robot arm system can be seen in the following table.

The robot arm system is modeled mathematically in the form of state space equations

$$\dot{x} = Ax + Bu \tag{6}$$

where A and B are state matrix and input matrix respectively. The A and B matrices obtained from the state space equation are substituted into the Algebraic Riccati Equation

Table 2: Effect of Q and R parameters on the robot arm

Action	Effect on robot arm
Increasing Q	The system will prioritize so that the set point position is reached immediately and tends not to be concerned with the amount of torque used.
Increasing R	The system will be more conservative about the amount of torque used and tends not to be concerned with the time it takes to reach the desired set point.

(ARE) in Equation 3. The gain value K can be obtained from Equation 5 by solving ARE to obtain the value of P. The relationship between all these variables can be seen in the closed-loop diagram in Figure 3 below.

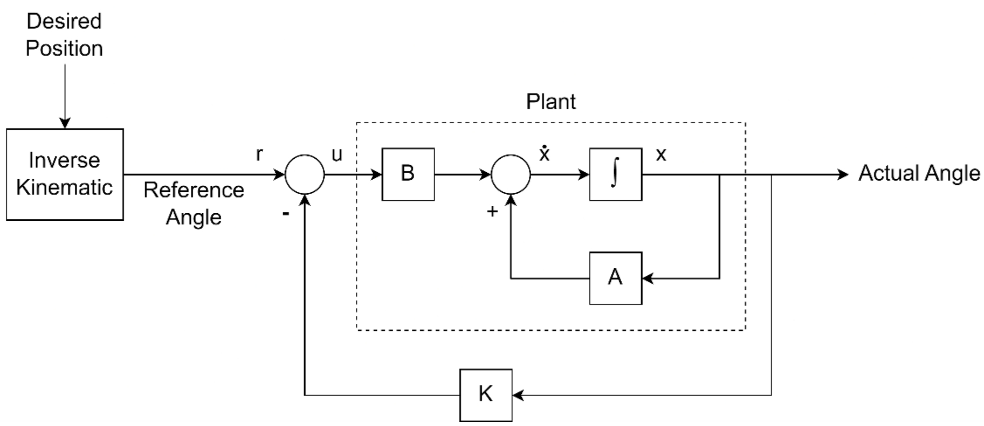


Figure 3: Closed-loop diagram.

The actual position of the end-effector will be affected by several main factors, namely rounding numbers during the calculation process, recoil due to the movement of the robot arm and the effect of gravity, especially when there is a load on the gripper so that there will be a difference or error between the reference and actual positions. In this robot arm system, the desired error is ≤ 1 cm in each of the X, Y and Z coordinates.

2.3 Inverse Kinematics

Inverse kinematics is a method to determine the angle value at each joint based on the position and orientation of the end-effector [15]. Inverse kinematics derivation can be done using Pythagorean law and trigonometric rules by looking at the top and side views of the robot arm [16].

From Figure 4, which shows the top view of the robot, we can obtain the angle at joint 1, which is as follows.

$$\theta_1 = \tan^{-1}(\frac{y}{x}) \tag{7}$$

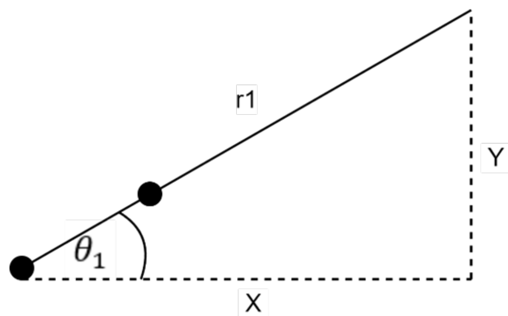


Figure 4: 3 DOF robot arm top view.

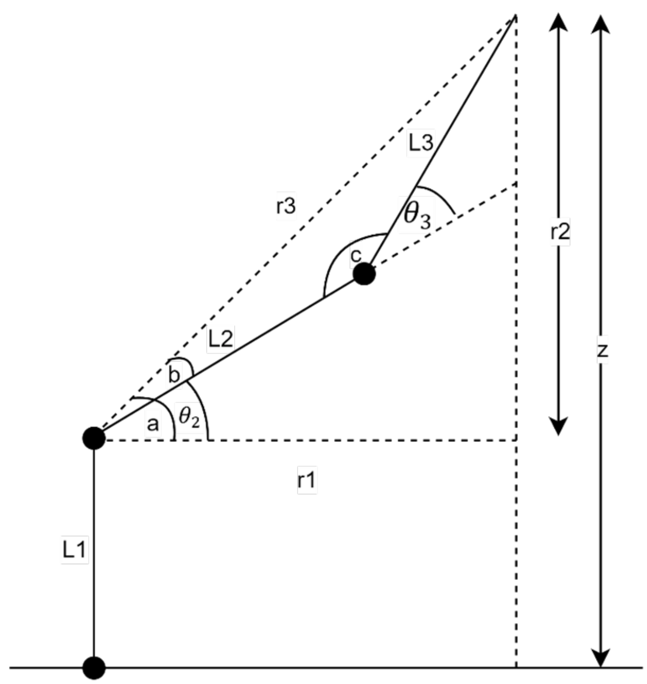


Figure 5: 3 DOF robot arm side view.

Based on Figure 4 and Figure 5 above, the values of r_1, r_2, and r_3 are:

$$r_1 = \sqrt{x^2 + y^2}$$

$$r_2 = z - L_1$$

$$r_3 = \sqrt{{r_1}^2 + {r_2}^2}$$

$$r_3 = \sqrt{x^2 + y^2 + (z - L_1)^2}$$

The value of θ_2 can be calculated by adding the angle values of a and b.

$$\theta_2 = a - b$$

$$a = \tan^{-1}\left(\frac{r_2}{r_1}\right)$$

$$a = \tan^{-1}\left(\frac{z - L_1}{\sqrt{x^2 + y^2}}\right)$$

$${L_3}^2 = {L_2}^2 + {r_3}^2 - 2L_2 r_3 \cos b$$

$$b = \cos^{-1}\left(\frac{{L_2}^2 + x^2 + y^2 + (z - L_1)^2 - {L_3}^2}{2L_2\sqrt{x^2 + y^2 + (z - L_1)^2}}\right) \tag{8}$$

$$\theta_2 = \tan^{-1}\left(\frac{z - L_{13}}{\sqrt{x^2 + y^2}}\right) - \cos^{-1}\left(\frac{{L_2}^2 + x^2 + y^2 + (z - L_1)^2 - {L_3}^2}{2L_2\sqrt{x^2 + y^2 + (z - L_1)^2}}\right) \tag{9}$$

The value of θ_3 is obtained using the law of cosine as follows.

$${r_3}^2 = {L_2}^2 + {L_3}^2 - 2L_2 L_3 \cos(c)$$

$$\cos(c) = \frac{{L_2}^2 + {L_3}^2 - x^2 - y^2 - (z - L_1)^2}{2L_2 L_3}$$

$$\cos(\pi - \theta_3) = \frac{{L_2}^2 + {L_3}^2 - x^2 - y^2 - (z - L_1)^2}{2L_2 L_3}$$

$$\theta_3 = \pi - \cos^{-1} = \left(\frac{{L_2}^2 + {L_3}^2 - x^2 - y^2 - (z - L_1)^2}{2L_2 L_3}\right) \tag{10}$$

Each angle in Equation 7-10 was calculated in radian.

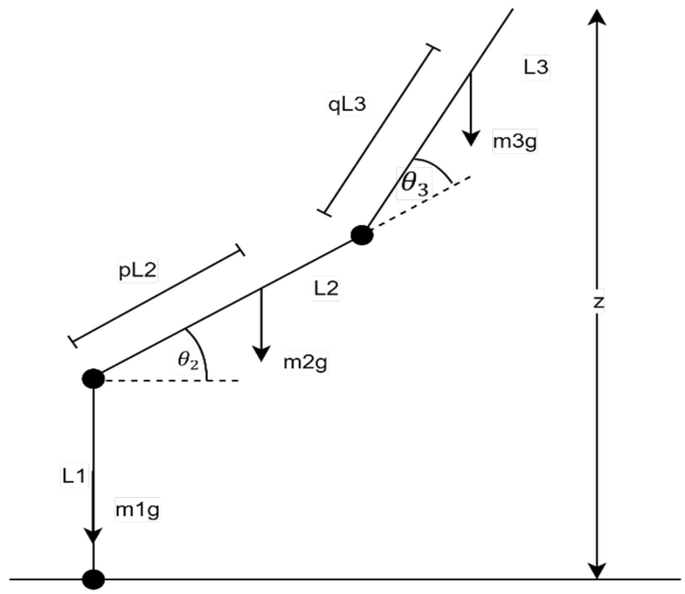


Figure 6: Free body diagram of the 3 DOF robot arm.

2.4 Langrangian Mechanic

Lagrangian mechanics is very useful in analyzing the movement of discrete particles within a certain number of degrees of freedom, so it is suitable for analyzing multi-link systems [12]. The Lagrange equation is derived based on the kinetic energy and potential energy of the system and expressed with respect to the general coordinates [17,18]. The Lagrange equation in this system is derived based on the total kinetic and potential energy at the center of gravity of each link. From Figure 6 above, the Lagrange equation obtained is as follows.

$$\mathcal{L}(\theta,\dot{\theta}) = K(\theta,\dot{\theta}) - P(\theta) \tag{11}$$

$$\begin{aligned}\mathcal{L}(\theta,\dot{\theta}) = &\left[\frac{1}{2}m_3q^2{l_3}^2\cos^2(\theta_2+\theta_3) + m_3ql_2l_3\cos(\theta_2+\theta_3)\cos(\theta_2) + \frac{{l_2}^2}{2}(m_3+m_2p^2)\cos^2(\theta_2)\right]\\ &\dot{\theta}_1^2 + \left[\frac{1}{2}m_3q^2{l_3}^2 + m_3ql_2l_3\cos(\theta_3) + \frac{{l_2}^2}{2}(m_3+m_2p^2\right]\\ &\dot{\theta}_2^2 + \left[\frac{1}{2}m_3q^2{l_3}^2\right]\dot{\theta}_2^2 + \left[m_3q^2{l_3}^2 + m_3ql_2l_3\cos(\theta_3)\right]\\ &\dot{\theta}_2\dot{\theta}_3 - \left[\frac{m_1}{2} + m_2 +,_3\right]gl_1 + \left[m_2p+m_3\right]gl_2\sin(\theta_2) - \left[m_3q\right]gl_3\sin(\theta_2+\theta_3)\end{aligned} \tag{12}$$

2.5 Inverse Dynamics

Inverse dynamics is a method used to determine the forse and torque required to produce a desired motion in dynamic system, such as robot arm. The dynamic model of a robot manipulator is formulated by considering forces, motions, and kinetic energy. This often used for stability analysis and control design of manipulators. The dynamic equation of the manipulator can be written in simplified form as [19,20]:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \tag{13}$$

In the context of robotics, this equation is commonly known as inverse dynamics, where $M(\theta)$ is the generalized mass inertial matrix, $C(\theta, \dot{\theta})$ is the generalized bias force including Coriolis and centrifugal forces matrix and $G(\theta)$ is the gravitational force matrix [21]. The value of each matrix in the inverse dynamics can be found using the following Euler-Lagrange equation [22,23].

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}(\theta, \dot{\theta})}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}(\theta, \dot{\theta})}{\partial \dot{\theta}} = \tau \tag{14}$$

Equation 14 above can be expanded as follows.

$$\begin{aligned} \frac{\partial \mathcal{L}(\theta, \dot{\theta})}{\partial \dot{\theta}} &= M(\theta)\dot{\theta} \\ \frac{d}{dt}\left(\frac{\partial \mathcal{L}(\theta, \dot{\theta})}{\partial \dot{\theta}}\right) &= M(\theta)\ddot{\theta} + \dot{M}(\theta)\dot{\theta} \end{aligned} \tag{15}$$

The kinetic energy in the Lagrange Equation 11 can be expanded into the following equation.

$$K(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta}$$

So, the Lagrange Equation 11 becomes,

$$\mathcal{L}(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} - P(\theta)$$

and,

$$\frac{\partial \mathcal{L}(\theta, \dot{\theta})}{\partial \theta} = \frac{1}{2}\frac{\partial}{\partial \theta}\dot{\theta}^T M(\theta)\dot{\theta} - \frac{\partial P(\theta)}{\partial \theta} \tag{16}$$

Substitute Equation 15 and 16 into Equation 14 and we get the following equation.

$$M(\theta)\ddot{\theta} + \underbrace{\dot{M}(\theta)\dot{\theta} - \frac{1}{2}\frac{\partial}{\partial \theta}\dot{\theta}^T M(\theta)\dot{\theta}}_{C(\theta, \dot{\theta})} + \underbrace{\frac{\partial P(\theta)}{\partial \theta}}_{G(\theta)} = \tau \tag{17}$$

By grouping each term in Equation 17 based on the form in Equation 13, the value of each matrix in the equation of motion can be determined. The following are the values of each matrix in the equation of motion.

Mass inertia matrix:

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

$$M_{11} = m_3 q^2 {l_3}^2 \cos^2(\theta_2 + \theta_3) + 2m_3 q l_2 l_3 \cos(\theta_2 + \theta_3)\cos(\theta_2) + {l_2}^2(m_3 + m_2 p^2)\cos^2(\theta_2)$$

$$M_{12} = M_{13} = M_{21} = M_{31} = 0$$

$$M_{22} = m_3 q^2 {l_3}^2 + 2m_3 q l_2 l_3 \cos(\theta_3) + {l_2}^2(m_3 + m_2 p^2)$$

$$M_{23} = M_{32} = m_3 q^2 {l_3}^2 + m_3 q l_2 l_3 \cos(\theta_3)$$

$$M_{33} = m_3 q^2 {l_3}^2$$

Coriolis/ centrifugal matrix:

$$C(\theta, \dot{\theta}) = \begin{bmatrix} C_{11} \\ C_{21} \\ C_{31} \end{bmatrix}$$

$$\begin{aligned} c_{11} =& [-2m_3 q^2 {l_3}^2 \cos(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3) - 2m_3 q l_2 l_3 \sin(\theta_2 + \theta_3) \\ &- 2{l_2}^2(m_3 + m_2 p^2)\cos(\theta_2)\sin(\theta_2)]\dot{\theta}_1\dot{\theta}_2 \\ &+ [-2m_3 q^2 {l_3}^2 \cos(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3) \\ &- 2m_3 q l_2 l_3 \sin(\theta_2 + \theta_3)\cos(\theta_2)]\dot{\theta}_1\dot{\theta}_3 \end{aligned}$$

$$\begin{aligned} c_{21} =& [-m_3 q l_2 l_3 \sin(\theta_3)]\,\dot{\theta}_3^2 - [2m_3 q l_2 l_3 \sin(\theta_3)]\,\dot{\theta}_2\dot{\theta}_3 \\ &+ [m_3 q^2 l_3^2 \cos(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3) \\ &+ m_3 q l_2 l_3 \sin(2\theta_2 + \theta_3) + {l_2}^2(m_3 + m_2 p^2)\cos(\theta_2)\sin(\theta_2)]\dot{\theta}_1^2 \end{aligned}$$

$$\begin{aligned} c_{31} =& \left[m_3 q^2 {l_3}^2 \cos(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3) + m_3 q l_2 l_3 \sin(\theta_2 + \theta_3)\cos(\theta_2)\right]\dot{\theta}_1^2 \\ &+ [m_3 q l_2 l_3 \sin(\theta_3)]\,\dot{\theta}_2^2 \end{aligned}$$

Gravity matrix:

$$G(\theta) = \begin{bmatrix} G_{11} \\ G_{21} \\ G_{31} \end{bmatrix}$$

$$G_{11} = 0$$

$$G_{21} = (m_3 + m_2 p) g l_2 \cos(\theta_2) + m_3 q g l_3 \cos(\theta_2 + \theta_3)$$

$$G_{31} = m_3 q g l_3 \cos(\theta_2 + \theta_3)$$

2.6 System Linearization

The derivation of the Inverse Dynamics can provide an overview of the system dynamics, but there are still some non-linear functions that must be linearized first to get the system state space equation. The state space equation for the 3 DOF robot arm is $\dot{x} = Ax + Bu$. where,

$$x = \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix} \quad \dot{x} = \begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\\ \ddot{\theta}_1\\ \ddot{\theta}_2\\ \ddot{\theta}_3\end{bmatrix} \quad u = \begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\end{bmatrix} \tag{18}$$

The system can be linearized around the reference point (x^*, u^*) using the Taylor series expansion.

$$\dot{x} = f(x,u) \approx f(x^*,u^*) + \left[\frac{\partial f}{\partial x}\right]_{x=x^*,u=u^*}(x - x^*) + \left[\frac{\partial f}{\partial u}\right]_{x=x^*,u=u^*}(u - u^*)$$

We assume the specific case where linearization at the reference point (x^*, u^*) causes the value of $f(x^*, u^*)$ to be zero, so what remains is the standard form of the linear state space equation [22].

$$\begin{aligned}\dot{x} &= \begin{bmatrix}\dot{\theta}\\ M^{-1}(\theta)[u - C(\theta,\dot{\theta})\dot{\theta} - G(\theta)]\end{bmatrix},\\ &\approx A(x - x^*) + B(u - u^*)\end{aligned}$$

Where A and B are constant matrices. Evaluation of the Taylor series expansion yields the following equation [23].

$$A = \begin{bmatrix}0 & I\\ -M^{-1}(\theta)\frac{\partial G(\theta)}{\partial \theta} & -M^{-1}(\theta)C(\theta,\dot{\theta})\end{bmatrix}_{x=x^*,u=u^*}$$

$$B = \begin{bmatrix}0\\ M^{-1}(\theta)\end{bmatrix}_{x=x^*,u=u^*}$$

Based on the A and B matrices obtained, the state space equation of the 3 DOF legan robot system is as follows.

$$\dot{x} = Ax + Bu$$

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\\ \ddot{\theta}_1\\ \ddot{\theta}_2\\ \ddot{\theta}_3\end{bmatrix} = \begin{bmatrix} 0_{3x3} & I_{3x3} \\ -M^{-1}(\theta)\frac{\partial G(\theta)}{\partial \theta} & -M^{-1}(\theta)C(\theta,\dot{\theta}) \end{bmatrix}_{x=x^*,u=u^*} \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix} + \begin{bmatrix} 0_{3x3} \\ M^{-1}(\theta)\end{bmatrix}_{x=x^*,u=u^*} \tag{19}$$

Linearization is performed around the reference points

$$x^T = \begin{bmatrix}0 & 0 & \frac{\pi}{2} & 0 & 0 & 0\end{bmatrix}$$

and

$$u^T = \begin{bmatrix}0 & 0 & 0\end{bmatrix}$$

. By using the parameters in Table 1, the values of A and B matrices are:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -98.4528 & -98.4528 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 150.6591 & 0 & 0 \\ 0 & 150.6591 & -150.6591 \\ 0 & -150.6591 & 508.5536 \end{bmatrix}$$

3 Results

Testing on the system includes two parts. The first is to test the system response to the LQR control that has been designed, and the second test is on the inverse kinematics sub-system.

3.1 LQR Control Testing

This test is carried out to ensure that changes in parameters Q and R in the LQR control model have fulfilled the properties in Table 2. In this test, the system will be given a step signal input by moving each joint from an angular position of 0 to an angular position of 90. In each test, an increase and decrease in each Q and R value will be carried out and then record the torque value at each joint and observe changes to the given Q and R values. in this test, the step response at each joint will also be observed. The initial values of Q and R matrices used are:

$$Q = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12 \end{bmatrix} \quad R = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 770 & 0 \\ 0 & 0 & 770 \end{bmatrix}$$

1. Test for Q matrix effect
   Increase the Q matrix to:

$$\begin{bmatrix} 25 & 0 & 0 & 0 & 0 \\ 0 & 250 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 250 \\ 0 & 0 & 0 & 0 & 15 \end{bmatrix}$$

   Decrease the Q matrix to:

$$\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 150 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 150 \\ 0 & 0 & 0 & 0 & 9 \end{bmatrix}$$

2. Test for R matrix effect
   Increase the R matrix to:

$$\begin{bmatrix} 130 & 0 & 0 \\ 0 & 800 & 0 \\ 0 & 0 & 800 \end{bmatrix}$$

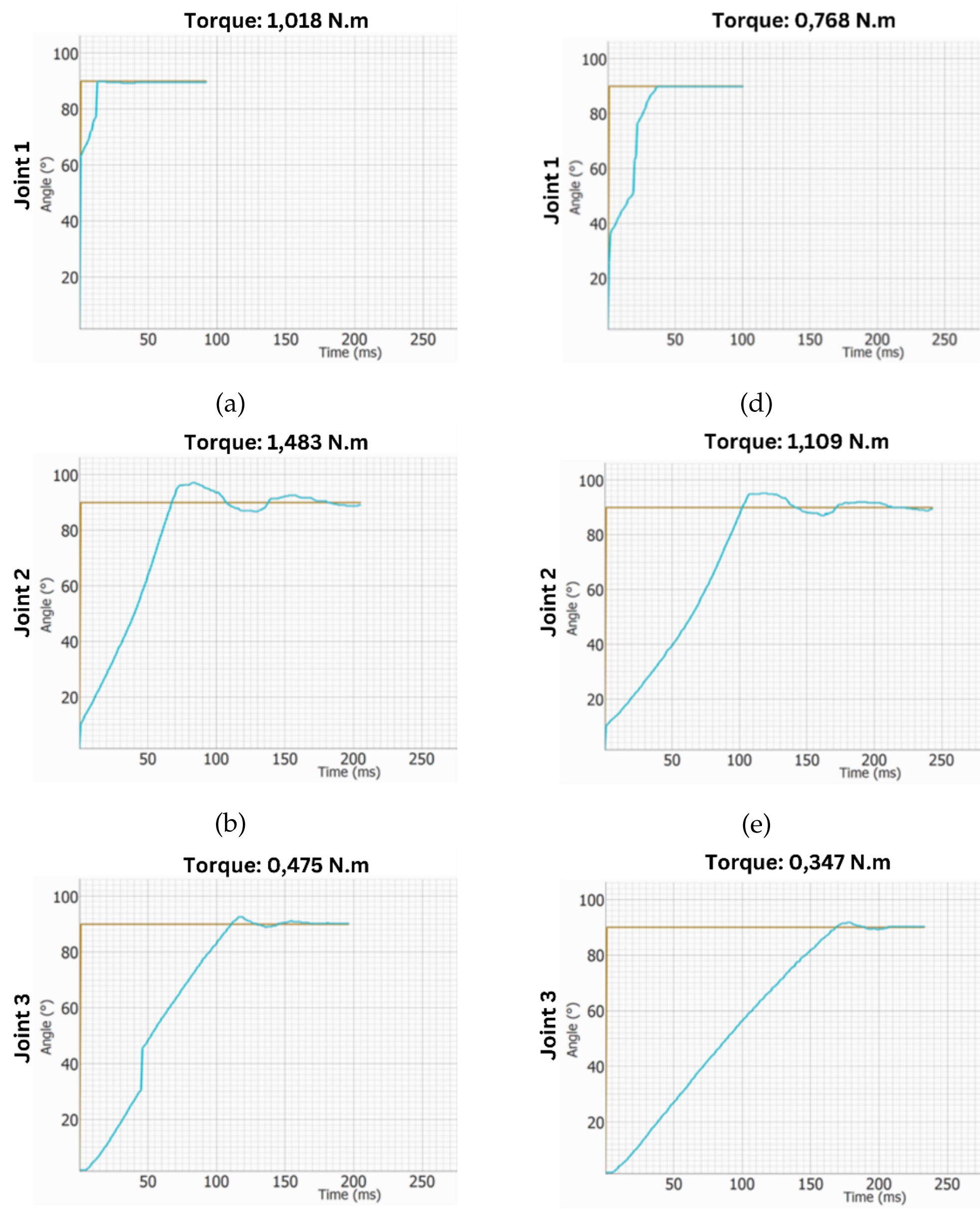
   Decrease the R matrix to:

$$\begin{bmatrix} 70 & 0 & 0 \\ 0 & 600 & 0 \\ 0 & 0 & 600 \end{bmatrix}$$

LQR control allows improvements in system performance, including settling time and overshoot by tuning the weight parameters Q and R. The results show that by selecting appropriate Q and R, fast settling time and low overshoot can be achieved as shown in Figure 7 (a-c) for Q matrix set up and **??** (d-f) for R matrix set up.

3.2 Inverse Kinematics Test

The inverse kinematics test aims to observe the error between the actual position of the end-effector and the reference position. Testing was carried out on a trajectory for pick and place movements consisting of 7 waypoints with gain values that have been tuned for that waypoint trajectory. The following is the waypoint trajectory, and the gain matrix k used.



(a) (d) (b) (e) (c) (f)

Figure 7: Step response of each joint based on value in Q matrix when increased (1-3) or decreased (4-6).

Waypoints 2 and 5 are positions for picking up and place the load respectively, while waypoint 7 is the stand-by position of the end-effector. The tuned gain value used in the testing is as follows.

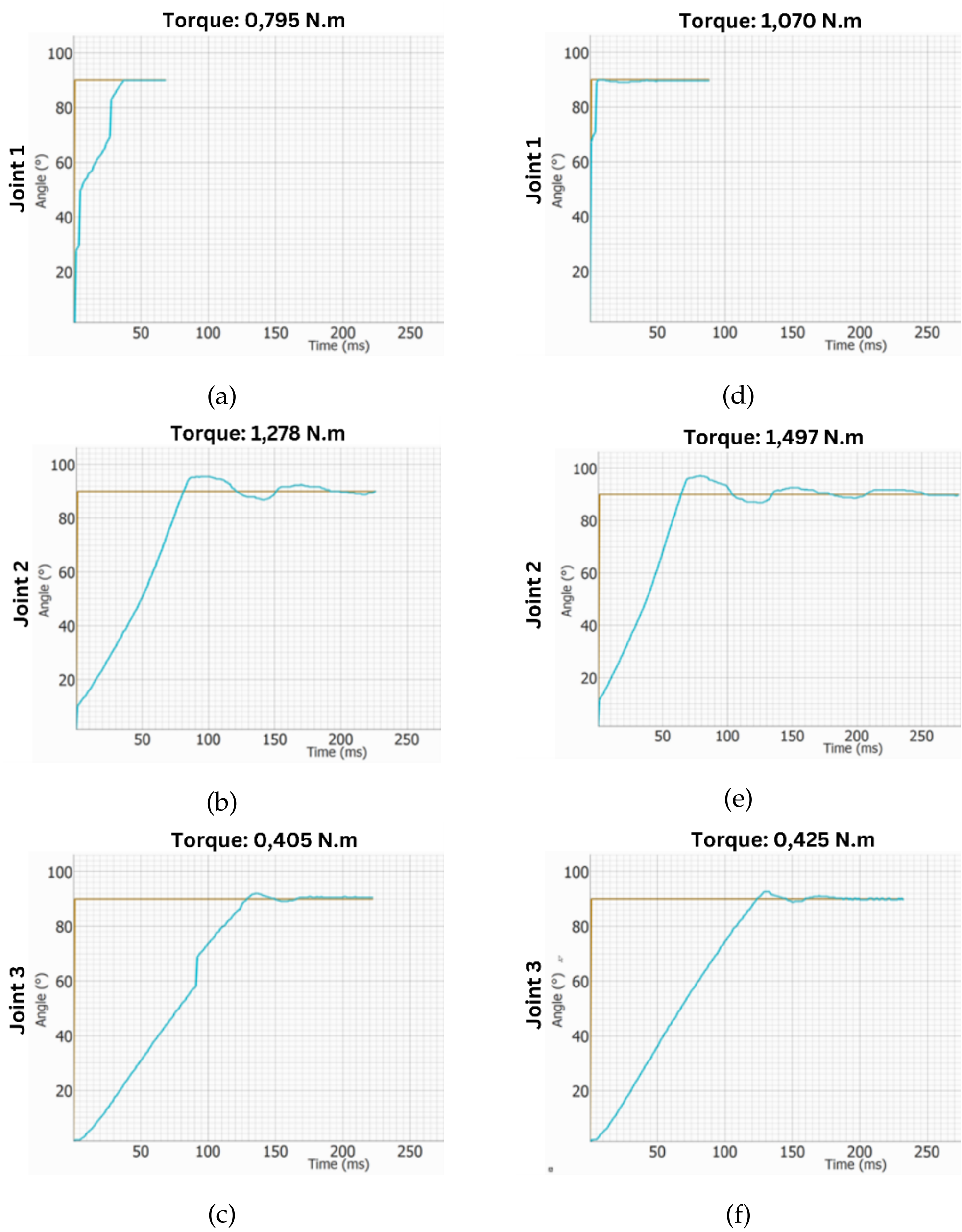


Figure 8: Step response of each joint based on value in R matrix when increased (1-3) or decreased (4-6).

$$k = \begin{bmatrix} 0.3162 & 0 & 0 & 0.5419 & 0 & 0 \\ 0 & 0.6722 & -0.0854 & 0 & 0.8265 & 0.0823 \\ 0 & -0.3281 & 0.0636 & 0 & -0.2479 & 0.2460 \end{bmatrix}$$

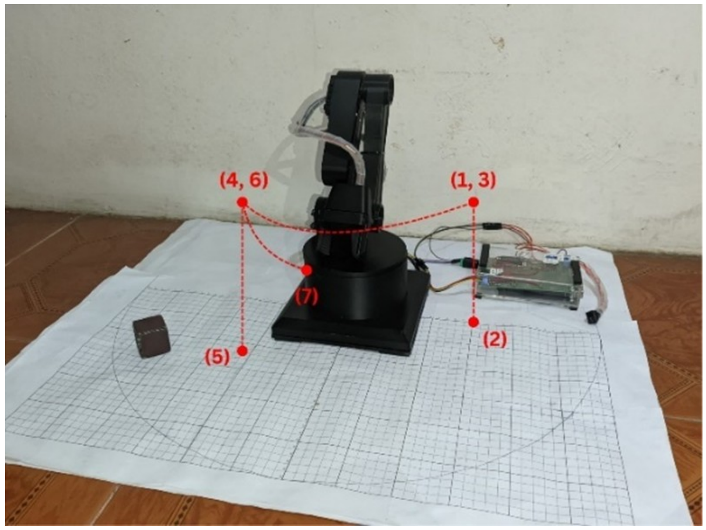


Figure 9: Waypoint position in workspace.

Table 3: Waypoint trajectory

		Waypoint						
		1	2	3	4	5	6	7
Reference position (cm)	X	0	0	0	12	12	12.1	20
	Y	15	15	15	-11	-11	-11.5	0
	Z	20	5	20	20	4.9	20	20

Table 4: Position and error of the actual waypoint position

		Waypoint						
		1	2	3	4	5	6	7
Actual position (cm)	X	0	-0.1	0.2	0.2	12.2	12.1	19.8
	Y	15.1	14.8	15.2	-11.3	-11.4	-11.5	-0.3
	Z	19.5	5.2	19.4	19.3	4.9	18.8	19.2
Error (cm)	X	0	0.1	0.2	0.2	0.1	0.1	0.2
	Y	0.1	0.2	0.2	0.3	0.4	0.5	0.3
	Z	0.5	0.2	0.6	0.7	0.1	1.2	0.8

Based on the results in Table 4, it can be calculated that the waypoint average errors in the end effector on the x, y, z axes are 14.9%, 16,26% and 3.67% respectively. However, the large errors values occurred at small positions so that it is not so problematic qualitatively. Video of the testing can be seen in this link: Demo_ArmRobot Trajectory.

4 Discussion

In Figure 7, an increase in the value of the Q matrix will increase the torque used at each joint. This is because an increase in the value of the Q matrix means that the system prioritizes so that the specified state is immediately achieved, in this case the reference angular position, so it requires a lot of effort, namely the torque at each joint, as seen from the system response in Figure 7 (a-c) where the system rise time is faster than when the Q matrix value is lowered where the system behavior shows the opposite as seen in Figure 7 (d-f). Faster rise time also result a higher overshoot in this case, also known as aggressive control. While in **??**, an increase in the value of matrix R will decrease the torque used at each joint. This is because an increase in the value of the R matrix means that the system will be more conservative about the amount of effort used, so the system takes longer to reach the desired state, as seen from the system response in **??** (a-c) where the system rise time becomes longer. On the other hand, a decrease in the value of the R matrix causes the system to show the opposite behavior as seen in **??** (d-f). In this case, slower rise time resulting a smaller overshoot in the step response. This behavior is known as conservative control. The effect of q and r parameters obtained from testing the LQR control system has shown results that are in line with the effect of q and r parameters that have been described in Table 2. We can use this result to improve stability, where the system has a low overshot by increasing the value of R matrix or decreasing the value of Q matrix.

The data from Table 4 shows that the errors in the X and Y coordinates all have an error ≤ 1 cm. However, the error in the Z coordinate is relatively larger than X and Y and there are error values that exceed 1 cm. The main factors that cause this are the gravitational force on each link and the recoil that occurs when the robot arm moves, especially when the joint needs a large torque value to overcome gravity when lifting load. This can be seen from the relatively larger error value of the Z coordinate when lifting load in the 3rd and 6th waypoint. Each joint actuator can be connected to a different power source to overcome the torque issue.

The two results above show that the LQR control method is reliable in multi-link systems like arm robotics system. When compared to other control methods such as PID, fuzzy logic and MPC [5–7], LQR offers consistent modeling at higher DOFs, simpler tuning process and relatively fast system response. This method offers good performance if the system modeling and parameter tuning are also good. The drawback of this method is that modeling for nonlinear systems such as robot arms requires a linearization process that can be rather tricky and the lack of structured methods in the tuning process which is generally done by trial-and-error methods.

5 Conclusion

The test results show that the 3 DOF robot arm can operate using the LQR control method to set the optimal torque for each joint to move in a certain trajectory. Tuning the Q and R parameters in LQR control uses the trial-and-error method to find the optimal combination for a particular trajectory. Tuning the Q and R parameters is done to regulate system stability by reducing overshoot in the system response. Still, on the other hand, it causes the rise time value to be greater. The horizontal position of the end-effector in X and Y coordinates has a relatively small error compared to the vertical position Z coordinate, due to gravitational force causing a heavier load when moving vertically. The results show that the end effector's waypoint average errors on the x, y, z axes are 14.9%, 16,26%, and 3.67% respectively.

Acknowledgments

The authors would like to thank DRTPM DIKTI for financial support through research grant number 106/E5/PG.02.00.PL/2024; 043/SP2H/RT-MONO/LL4/2024; 047/LIT07/PPM-LIT/2024, which enabled the completion of the study and preparation of this manuscript.

References

[1] W. S. Barbosa, M. M. Gioia, V. G. Natividade, R. F. F. Wanderley, M. R. Chaves, F. C. Gouvea, and F. M. Gonçalves, "Industry 4.0: examples of the use of the robotic arm for digital manufacturing processes," *Int. J. Interact. Des. Manuf. (IJIDeM)*, vol. 14, pp. 1569–1575, Dec. 2020.

[2] M. Ben-Ari and F. Mondada, "Kinematics of a robotic manipulator," in *Elements of Robotics*, pp. 267–291, Cham: Springer International Publishing, 2018.

[3] P. Saraf and R. N. Ponnalagu, "Modeling and simulation of a point to point spherical articulated manipulator using optimal control," in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, IEEE, Feb. 2021.

[4] M. Akhtaruzzaman, A. A. Shafie, M. R. Khan, and M. M. Rahman, "Robot assisted knee joint RoM exercise: A PID parallel compensator architecture through impedance estimation," *Cognitive Robotics*, vol. 4, pp. 42–61, 2024.

[5] P. Chotikunnan and R. Chotikunnan, "Dual design pid controller for robotic manipulator application," *Journal of Robotics and Control (JRC)*, vol. 4, no. 1, pp. 23–34, 2023.

[6] A. M. Abdul-Sadah, K. M. H. Raheem, and M. M. S. Altufaili, "A fuzzy logic controller for a three links robotic manipulator," in *3RD INTERNATIONAL SCIENTIFIC CONFERENCE OF ALKAFEEL UNIVERSITY (ISCKU 2021)*, AIP Publishing, 2022.

[7] Y. Chen, X. Luo, B. Han, Q. Luo, and L. Qiao, "Model predictive control with integral compensation for motion control of robot manipulator in joint and task spaces," *IEEE Access*, vol. 8, pp. 107063–107075, 2020.

[8] A. Ortegan-Vidal, F. Salazarn-Vasquez, and A. Rojasn-Moreno, "A comparison between optimal LQR control and LQR predictive control of a planar robot of 2DOF," in *2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, IEEE, Sept. 2020.

[9] J. Dong, R. Liu, B. Lu, X. Guo, and H. Liu, "LQR-based balance control of two-wheeled legged robot," in *2022 41st Chinese Control Conference (CCC)*, IEEE, July 2022.

[10] Fahmizal, H. A. Nugroho, A. I. Cahyadi, and I. Ardiyanto, "Tuning LQR parameters using neuro evolution of augmenting topologies (NEAT) on a double pendulum cart," in *2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, IEEE, Aug. 2022.

[11] M. Saraf, A. Agarwal, A. Chaudhary, and A. Ganthale, "Kinematic modelling and motion mapping of robotic arms," *J. Phys. Conf. Ser.*, vol. 1969, p. 012002, July 2021.

[12] S. Mustary, M. A. Kashem, M. A. Chowdhury, and M. M. Rana, "Mathematical model and evaluation of dynamic stability of industrial robot manipulator: Universal robot," *Systems and Soft Computing*, vol. 6, p. 200071, Dec. 2024.

[13] Z. Su, H. Yao, J. Peng, Z. Liao, Z. Wang, H. Yu, H. Dai, and T. C. Lueth, "LQR-based control strategy for improving human–robot companionship and natural obstacle avoidance," *Biomimetic Intelligence and Robotics*, vol. 4, p. 100185, Dec. 2024.

[14] K. Ishihara, T. D. Itoh, and J. Morimoto, "Full-body optimal control toward versatile and agile behaviors in a humanoid robot," *IEEE Robot. Autom. Lett.*, vol. 5, pp. 119–126, Jan. 2020.

[15] H. D. Salman, M. N. Hamzah, and S. H. Bakhy, "Kinematics analysis and implementation of three degrees of freedom robotic arm by using matlab," *The Iraqi Journal for Mechanical and Materials Engineering*, vol. 21, no. 2, pp. 118–129, 2021.

[16] A. R. Al Tahtawi, M. Agni, and T. D. Hendrawati, "Small-scale robot arm design with pick and place mission based on inverse kinematics," *jrc*, vol. 2, no. 6, 2021.

[17] S. Park, "Design, implementation, and control of a ball-balancing robot," *IEEE Access*, vol. 12, pp. 127380–127389, 2024.

[18] F. Renda, A. Mathew, and D. F. Talegon, "Dynamics and control of soft robots with implicit strain parametrization," *IEEE Robot. Autom. Lett.*, vol. 9, pp. 2782–2789, Mar. 2024.

[19] E. Susanto, "Pemodelan dan kendali sistem dinamik menggunakan multibody," 2023.

[20] J. Batista, D. Souza, L. Dos Reis, A. Barbosa, and R. Araújo, "Dynamic model and inverse kinematic identification of a 3-dof manipulator using rlspso," *Sensors*, vol. 20, no. 2, p. 416, 2020.

[21] H. Ren and P. Ben-Tzvi, "Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks," *Rob. Auton. Syst.*, vol. 124, p. 103386, Feb. 2020.

[22] X. Jing and C. Li, "Dynamic modeling and solution of 6-DOF parallel mechanism," *IEEE Access*, vol. 10, pp. 33695–33703, 2022.

[23] G. E. Dongming, S. Guanghui, Z. Yuanjie, and S. Jixin, "Impedance control of multi-arm space robot for the capture of non-cooperative targets," *J. Syst. Eng. Electron.*, vol. 31, pp. 1051–1061, Oct. 2020.