



YILDIZ TECHNICAL UNIVERSITY

Modeling and Control of a 3-DOF RRR Serial Manipulator

Ege Günay

Student ID: 24568020

June 2025

Abstract

This report presents the modeling, simulation, and control of a 3-DOF (Revolute-Revolute-Revolute, RRR) serial manipulator. Forward and inverse kinematics, dynamic equations, and classical as well as advanced control approaches are developed. The performances of PD, Computed Torque, Gravity, Impedance, and Variable Structure Controllers are evaluated via MATLAB/Simulink simulations. Comparative analysis demonstrates the strengths and limitations of each controller in both joint and task space.

1 Introduction (Literature Review)

Robotic manipulators are fundamental in modern automation and manufacturing, require precise modeling and advanced control for complex motion tasks. Classical control methods such as PID/PD have been widely used due to their simplicity, but may lack robustness in the presence of nonlinearities and disturbances [1]. Computed torque controllers (CTC), based on dynamic models, can linearize and decouple robot dynamics, thus enabling higher performance, especially for trajectory tracking [2]. Gravity compensation further improves steady-state accuracy, particularly for vertical or heavy-load tasks. More recent approaches, such as impedance control and variable structure (sliding mode) control, offer enhanced robustness and safe interaction capabilities, crucial for collaborative robotics [3,4].

2 Robotic System Used in This Research

2.1 Kinematic and Dynamic Equations

2.1.1 Forward Kinematics of the RRR Manipulator

The forward kinematics of a planar RRR (3 revolute joint) manipulator is derived using the Denavit–Hartenberg (DH) convention, a widely adopted systematic approach in robotic modeling [1,2]. Each joint rotates in the plane, and all links are assumed to lie in the XY plane. Forward kinematics provides the position and orientation of the end effector for any given set of joint angles.

According to the DH convention, the parameters for each link are summarized in Table 1:

Table 1: Denavit–Hartenberg Parameters for the RRR Manipulator

i	θ_i	d_i	a_i	α_i
1	θ_1	0	a_1	0
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0

Based on these parameters, homogeneous transformation matrices between consecutive joint frames can be systematically constructed. The general transformation from

frame $(i - 1)$ to frame i is:

$$T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i & 0 & a_i \sin \theta_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $i = 1, 2, 3$ for this manipulator.

For the RRR manipulator, each individual transformation matrix is given explicitly as follows:

$$T_1^0 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & a_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

These matrices describe the relative transformations between consecutive coordinate frames attached to the joints. By sequentially multiplying these matrices, the overall transformation from the base to the end-effector can be obtained, capturing both position and orientation in the plane:

$$T_3^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 = \begin{bmatrix} \cos(\theta_{123}) & -\sin(\theta_{123}) & 0 & x \\ \sin(\theta_{123}) & \cos(\theta_{123}) & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{where } \theta_{123} = \theta_1 + \theta_2 + \theta_3.$$

The translational part of this matrix yields the end-effector position:

$$\begin{aligned} x &= a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \phi &= \theta_1 + \theta_2 + \theta_3 \end{aligned}$$

All the mathematical relationships and transformation matrices above provide a systematic method to compute the precise position and orientation of the end-effector for any joint configuration of the planar RRR manipulator. These results form the basis for trajectory planning, simulation, and advanced control algorithm development for the robotic system.

2.1.2 Inverse Kinematics of the RRR Manipulator

Given the desired end-effector position (x_e, y_e) and orientation ϕ , the joint angles of a planar RRR manipulator can be analytically determined using a geometric approach. First, the effect of the third link is subtracted to locate the so-called wrist center (i.e., the end of the second link), as follows:

$$\begin{aligned}x_w &= x_e - a_3 \cos(\phi) \\y_w &= y_e - a_3 \sin(\phi)\end{aligned}$$

This step effectively reduces the problem to a two-link manipulator, where (x_w, y_w) gives the position of the wrist relative to the base.

The angle θ_2 is then calculated using the law of cosines applied to the triangle formed by the first two links and the wrist position:

$$\theta_2 = \arccos\left(\frac{x_w^2 + y_w^2 - a_1^2 - a_2^2}{2a_1a_2}\right)$$

To find θ_1 , two auxiliary angles are introduced. The angle β represents the orientation from the base to the wrist, while ψ accounts for the offset created by the second link:

$$\begin{aligned}\beta &= \text{atan2}(y_w, x_w) \\ \psi &= \text{atan2}(a_2 \sin \theta_2, a_1 + a_2 \cos \theta_2) \\ \theta_1 &= \beta - \psi\end{aligned}$$

Finally, the third joint angle is calculated to achieve the desired end-effector orientation:

$$\theta_3 = \phi - (\theta_1 + \theta_2)$$

With these computations, the unique set of joint angles θ_1 , θ_2 , and θ_3 that achieve the specified end-effector pose can be analytically determined, provided the target lies within the manipulator's reachable workspace. This systematic approach is robust and widely used in both simulation and control of serial robotic arms.

2.1.3 Dynamic Modeling of the RRR Manipulator

The dynamic model of the 3-DOF planar RRR manipulator is derived to capture the motion and forces involved in manipulator movement, forming the foundation for accurate simulation and control. The modeling process begins with the calculation of kinetic and potential energies for all links.

Each link contributes both translational and rotational kinetic energy, given by:

$$T = \sum_{k=1}^3 \left(\frac{1}{2} m_k \mathbf{v}_k^T \mathbf{v}_k + \frac{1}{2} I_k \dot{\theta}_k^2 \right)$$

where \mathbf{v}_k is the linear velocity of the center of mass of link k , and I_k is its moment of inertia about the z -axis.

The potential energy due to gravity is expressed as:

$$\begin{aligned}V &= m_1 g \cdot \frac{a_1}{2} \sin \theta_1 \\ &\quad + m_2 g \left(a_1 \sin \theta_1 + \frac{a_2}{2} \sin(\theta_1 + \theta_2) \right) \\ &\quad + m_3 g \left(a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) + \frac{a_3}{2} \sin(\theta_1 + \theta_2 + \theta_3) \right)\end{aligned}$$

This term accumulates the gravitational potential of each link's center of mass, taking into account the serial structure.

For planar motion, only the z -component of each link's inertia tensor is relevant:

$$D_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I_k \end{bmatrix}, \quad k = 1, 2, 3$$

The velocities and angular velocities of each link's center of mass are computed via Jacobian matrices. The total kinetic energy for each link is:

$$T_k = \frac{1}{2} m_k \dot{q}^T J_{v_k}^T J_{v_k} \dot{q} + \frac{1}{2} \dot{q}^T J_{\omega_k}^T D_k J_{\omega_k} \dot{q}$$

where J_{v_k} is the linear velocity Jacobian and J_{ω_k} is the angular velocity Jacobian for link k .

The Jacobian matrices for each link are:

$$\text{Link 1: } J_{v_1} = \begin{bmatrix} -\frac{a_1}{2} \sin \theta_1 & 0 & 0 \\ \frac{a_1}{2} \cos \theta_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad J_{\omega_1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\text{Link 2: } J_{v_2} = \begin{bmatrix} -a_1 \sin \theta_1 - \frac{a_2}{2} \sin(\theta_{12}) & -\frac{a_2}{2} \sin(\theta_{12}) & 0 \\ a_1 \cos \theta_1 + \frac{a_2}{2} \cos(\theta_{12}) & \frac{a_2}{2} \cos(\theta_{12}) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad J_{\omega_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\text{Link 3: } J_{v_3} = \begin{bmatrix} -a_1 \sin \theta_1 - a_2 \sin(\theta_{12}) - \frac{a_3}{2} \sin(\theta_{123}) & -a_2 \sin(\theta_{12}) - \frac{a_3}{2} \sin(\theta_{123}) & -\frac{a_3}{2} \sin(\theta_{123}) \\ a_1 \cos \theta_1 + a_2 \cos(\theta_{12}) + \frac{a_3}{2} \cos(\theta_{123}) & a_2 \cos(\theta_{12}) + \frac{a_3}{2} \cos(\theta_{123}) & \frac{a_3}{2} \cos(\theta_{123}) \\ 0 & 0 & 0 \end{bmatrix},$$

$$J_{\omega_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

The overall joint-space inertia matrix is obtained as:

$$M(q) = \sum_{k=1}^3 (m_k J_{v_k}^T J_{v_k} + J_{\omega_k}^T D_k J_{\omega_k})$$

The inertia matrix $M(q)$ accounts for the configuration-dependent mass and moment of inertia of the system. For the planar 3-DOF manipulator, it can be written as:

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

where

$$M_{11}(q) = I_1 + I_2 + I_3 + m_1 r_1^2 + m_2 [l_1^2 + r_2^2 + 2l_1 r_2 \cos(q_2)] \\ + m_3 [l_1^2 + l_2^2 + r_3^2 + 2l_1 l_2 \cos(q_2) + 2l_1 r_3 \cos(q_2 + q_3) + 2l_2 r_3 \cos(q_3)]$$

$$M_{12}(q) = I_2 + I_3 + m_2 [r_2^2 + l_1 r_2 \cos(q_2)] \\ + m_3 [l_2^2 + r_3^2 + l_1 l_2 \cos(q_2) + l_1 r_3 \cos(q_2 + q_3) + 2l_2 r_3 \cos(q_3)]$$

$$M_{13}(q) = I_3 + m_3 [r_3^2 + l_1 r_3 \cos(q_2 + q_3) + l_2 r_3 \cos(q_3)]$$

$$M_{21}(q) = M_{12}(q)$$

$$M_{22}(q) = I_2 + I_3 + m_2 r_2^2 + m_3 [l_2^2 + r_3^2 + 2l_2 r_3 \cos(q_3)]$$

$$M_{23}(q) = I_3 + m_3 [r_3^2 + l_2 r_3 \cos(q_3)]$$

$$M_{31}(q) = M_{13}(q)$$

$$M_{32}(q) = M_{23}(q)$$

$$M_{33}(q) = I_3 + m_3 r_3^2$$

Explicit expressions can be systematically derived using the manipulator's kinetic energy and Jacobians.

This matrix captures the combined effect of all masses and inertias, and its elements depend on the robot's current configuration.

The center of mass position for each link, necessary for energy and gravity calculations, is:

$$\mathbf{r}_1 = \frac{a_1}{2} \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \\ 0 \end{bmatrix}, \quad \mathbf{r}_2 = a_1 \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \\ 0 \end{bmatrix} + \frac{a_2}{2} \begin{bmatrix} \cos(\theta_{12}) \\ \sin(\theta_{12}) \\ 0 \end{bmatrix} \\ \mathbf{r}_3 = a_1 \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \\ 0 \end{bmatrix} + a_2 \begin{bmatrix} \cos(\theta_{12}) \\ \sin(\theta_{12}) \\ 0 \end{bmatrix} + \frac{a_3}{2} \begin{bmatrix} \cos(\theta_{123}) \\ \sin(\theta_{123}) \\ 0 \end{bmatrix}$$

The Lagrangian, defined as $L(q, \dot{q}) = T(q, \dot{q}) - V(q)$, summarizes the energy structure of the manipulator and enables the systematic derivation of the equations of motion. Applying the Euler-Lagrange equations yields:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = \tau_i, \quad \text{for } i = 1, 2, 3$$

which leads to the standard robot manipulator dynamic equation:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

where $M(q)$ is the inertia matrix, $C(q, \dot{q})$ is the Coriolis and centrifugal matrix, $G(q)$ is the gravity torque vector, and τ is the input torque vector.

The Coriolis and centrifugal effects, captured by $C(q, \dot{q})$, arise from the velocity-dependent terms in the dynamic model. These can be systematically computed using the Christoffel symbols:

$$C_{ijk} = \frac{1}{2} \left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{jk}}{\partial \theta_i} \right)$$

$$C_{ij} = \sum_{k=1}^3 C_{ijk} \dot{\theta}_k$$

After differentiation, the explicit form of $C(q, \dot{q})\dot{q}$ is:

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} -m_2 a_1 a_2 \sin \theta_2 \dot{\theta}_2^2 - m_2 a_1 a_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 a_1 a_2 \sin \theta_2 \dot{\theta}_1^2 \\ 0 \end{bmatrix}$$

This vector expresses the dynamic coupling between the joints due to rotation.

Finally, the gravity vector $G(q)$ quantifies the torques required at each joint to counteract gravitational effects:

$$G_i(q) = \frac{\partial V(q)}{\partial \theta_i}, \quad i = 1, 2, 3$$

Using the potential energy expression above, the gravity vector becomes:

$$G(q) = \begin{bmatrix} (m_1 \frac{a_1}{2} + m_2 a_1 + m_2 \frac{a_2}{2} \cos \theta_2 + m_3 a_1 + m_3 a_2 \cos \theta_2 + m_3 \frac{a_3}{2} \cos(\theta_2 + \theta_3)) g \cos \theta_1 \\ (m_2 \frac{a_2}{2} + m_3 a_2 + m_3 \frac{a_3}{2} \cos \theta_3) g \cos(\theta_1 + \theta_2) \\ m_3 \frac{a_3}{2} g \cos(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}$$

This comprehensive formulation ensures all dynamic interactions—including inertia, Coriolis, centrifugal, and gravitational effects—are accurately represented for simulation and control purposes.

2.2 Dimensions and Specifications of the System

The robotic system employed in this study is a planar 3-degree-of-freedom (3-DOF) serial manipulator consisting of three revolute joints, commonly referred to as an RRR manipulator. All joints rotate in the XY-plane, and each link is assumed to be rigid, homogeneous, and mass-distributed along its length. The base of the manipulator is fixed to the origin of the workspace.

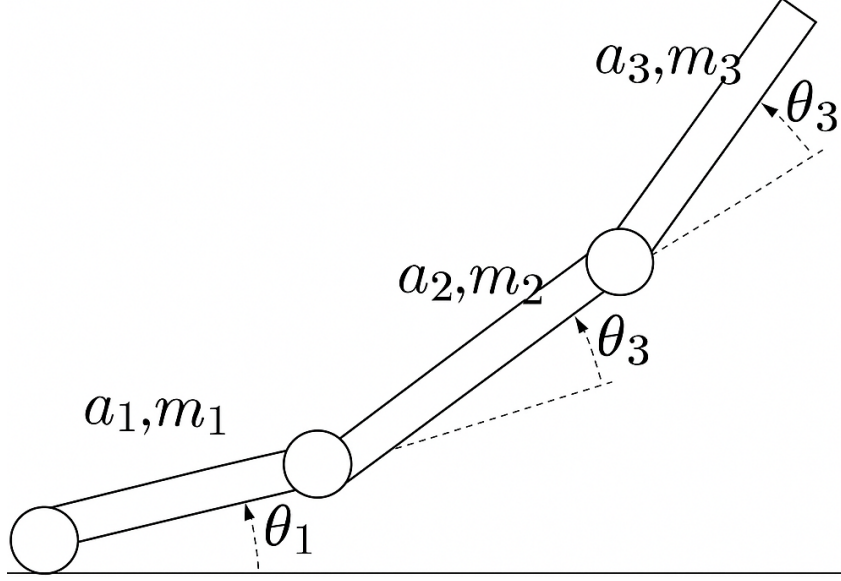


Figure 1: Schematic of the 3-DOF planar RRR manipulator with link lengths a_1 , a_2 , a_3 and joint angles θ_1 , θ_2 , θ_3 .

The physical and inertial parameters of the manipulator are summarized in Table 2.

Table 2: Physical and Inertial Parameters of the RRR Manipulator

Link	Length a_k [m]	Mass m_k [kg]
1	1.0	2.0
2	1.0	1.5
3	0.8	1.0

Additional specifications:

- **Workspace:** The manipulator covers a circular area with a maximum reach of $a_1 + a_2 + a_3 = 2.8$ m.
- **Gravity:** Uniform gravitational field $g = 9.81$ m/s², acting in the negative y -direction.

These specifications serve as the foundation for the kinematic and dynamic modeling, as well as for the simulation and controller design phases presented in subsequent sections.

3 Simulation Environment and Response of the Robotic System

This section presents the validation of the developed kinematic and dynamic models for the 3-DOF RRR manipulator, as well as a description of the simulation environment used throughout the project.

3.1 Validation of Kinematic Models

To verify the correctness of the forward kinematics, a known joint configuration was applied and the resulting end-effector position was computed analytically:

- $\theta_1 = 30^\circ, \theta_2 = 45^\circ, \theta_3 = -15^\circ$
- The resulting end-effector pose was: $x = 1.375$ m, $y = 1.899$ m, $\phi = 60^\circ$

Inverse kinematics was validated through a round-trip numerical test. Using the computed end-effector pose as the target:

- Given: $x_e = 1.375$ m, $y_e = 1.899$ m, $\phi = 60^\circ$
- The inverse kinematics algorithm returned: $\theta_1 = 30.01^\circ, \theta_2 = 44.98^\circ, \theta_3 = -14.99^\circ$

The close match between original and reconstructed joint angles confirms the numerical consistency and accuracy of both the forward and inverse kinematic formulations.

3.2 Validation of Dynamic Models

To ensure correctness of the dynamic model derived via the Euler–Lagrange formulation, three key components were numerically validated using the configuration:

$$q = [14^\circ, 14^\circ, -9^\circ], \quad \dot{q} = [0, 0, 0]$$

- The joint-space inertia matrix was evaluated as:

$$M(q) = \begin{bmatrix} 1.5935 & 0.8729 & 0.2063 \\ 0.8729 & 0.3878 & 0.1118 \\ 0.2063 & 0.1118 & 0.0130 \end{bmatrix}$$

- The gravity vector was computed as:

$$G(q) = \begin{bmatrix} 10.6343 \\ 3.4954 \\ 0.4638 \end{bmatrix}$$

- The Coriolis and centrifugal vector vanished under zero velocity:

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

These evaluations confirm the internal consistency and numerical validity of the dynamic equations derived in symbolic form.

3.3 Simulation Setup

All simulations were performed in **MATLAB R2024b** using the built-in `ode45` solver for numerical integration. Unless otherwise specified, a fixed simulation step size of $\Delta t = 0.01$ seconds was adopted. The state vectors included joint angles and angular velocities. Trajectories were interpolated linearly with extrapolation enabled for continuity. Simulation data was stored for post-processing and figure generation, using standardized axis formatting throughout.

The following tables summarize the common simulation parameters and test conditions used in all experiments.

Joint Space Set-Point Regulation:

Table 3: Simulation parameters for joint space set-point regulation experiments.

Parameter	Value
MATLAB Version	R2024b
Numerical Solver	ode45
Simulation duration (T)	5 s
Time step (Δt)	0.01 s
Link lengths (a_1, a_2, a_3)	1.0 m, 1.0 m, 0.8 m
Link masses (m_1, m_2, m_3)	2.0 kg, 1.5 kg, 1.0 kg
Gravity (g)	9.81 m/s ²
Initial joint positions ($\mathbf{q}(0)$)	$[0.1, 0.1, -0.1]^T$ rad
Initial joint velocities ($\dot{\mathbf{q}}(0)$)	$[0, 0, 0]^T$ rad/s
Desired joint configuration (\mathbf{q}_d)	$[\pi/4, \pi/6, -\pi/6]^T$ rad

Task Space Trajectory Tracking:

All task-space trajectory tracking simulations utilized the same environment described above. However, the experiments focused on following a predefined reference trajectory in Cartesian space rather than a set-point.

Table 4: Simulation parameters for end-effector (task space) trajectory tracking.

Parameter	Value
MATLAB Version	R2024b
Numerical Solver	ode45
Simulation duration (T)	20 s
Time step (Δt)	0.01 s
Link lengths (a_1, a_2, a_3)	1.0 m, 1.0 m, 0.8 m
Link masses (m_1, m_2, m_3)	2.0 kg, 1.5 kg, 1.0 kg
Gravity (g)	9.81 m/s ²
Initial joint positions ($\mathbf{q}(0)$)	Via inverse kinematics from (x_d, y_d, ϕ_d)
Initial joint velocities ($\dot{\mathbf{q}}(0)$)	$[0, 0, 0]^T$ rad/s
Reference trajectory center (x_0, y_0)	(1.5, 0) m
Reference trajectory radius r	0.5 m
Reference angular velocity ω	0.5 rad/s
Reference orientation ϕ_d	0 (fixed)
Evaluation metric	RMS end-effector position error

4 System Responses Comparison to Different Controllers

This section presents a detailed comparison of the dynamic responses of the RRR manipulator under various control strategies. For each controller, joint angle trajectories and torque profiles are examined during a set-point regulation task. The objective is to evaluate the controllers' ability to drive the manipulator from an initial configuration to a desired target configuration, and to quantify their tracking performance using RMS error metrics. Simulation results and analyses provide insight into the effectiveness, robustness, and limitations of each approach in regulating joint motion under identical initial conditions.

4.1 Joint Space Tracking Performance

Joint space tracking performance is assessed by commanding the RRR manipulator to move from an initial configuration to a specified joint-space set-point. For each controller, the resulting joint angle trajectories and actuator torque profiles are analyzed. Tracking accuracy is quantitatively evaluated using the root mean square (RMS) error between actual and desired joint positions, under identical simulation parameters for all cases.

4.1.1 PD Controller Design and Simulation

In this section, a classical Proportional-Derivative (PD) controller is implemented for joint space control of the RRR manipulator. The control law is applied independently to each joint as

$$\tau_i = K_{p,i}(q_{d,i} - q_i) + K_{d,i}(\dot{q}_{d,i} - \dot{q}_i)$$

or, in vector form,

$$\boldsymbol{\tau} = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}})$$

where \mathbf{K}_p and \mathbf{K}_d are diagonal gain matrices. This controller assumes decoupled joint dynamics and does not explicitly compensate for nonlinear coupling or gravity effects.

To provide a systematic and robust selection of PD gains, the closed-loop dynamics of each joint are approximated as a second-order system by neglecting coupling and nonlinearities. Unlike approaches based on a single or nominal configuration, the diagonal elements of the inertia matrix $M_{ii}(q)$ were evaluated over the full joint workspace, and their maximum values were used for gain calculations. This ensures robust performance and stability for all admissible robot poses.

Mathematically, for each joint i , the closed-loop dynamics can be approximated as:

$$M_{ii}(q)\ddot{q}_i + K_{d,i}\dot{e}_i + K_{p,i}e_i = 0$$

which can be matched to the standard second-order system:

$$\ddot{e}_i + 2\zeta\omega_n\dot{e}_i + \omega_n^2e_i = 0$$

where ζ is the desired damping ratio and ω_n is the desired natural frequency. Equating coefficients, the gains are selected as:

$$K_{d,i} = 2\zeta\omega_n M_{ii,\max}, \quad K_{p,i} = \omega_n^2 M_{ii,\max}$$

where $M_{ii,\max}$ is the maximum value of the i th diagonal element of the inertia matrix across the full joint workspace.

The maximum inertia values were obtained by systematically evaluating $M_{ii}(q)$ across the feasible ranges of all joint variables. The final gain values were numerically computed in MATLAB using these maxima, with the following controller parameters: $\zeta = 1$ (critical damping) and $\omega_n = 5$ rad/s.

The resulting values are:

- $M_{11,\max} = 9.7058$
- $M_{22,\max} = 1.4669$
- $M_{33,\max} = 0.3800$

Therefore, the analytically calculated PD gains are:

• **Joint 1:**

$$K_{d,1} = 2 \times 1 \times 5 \times 9.7058 = 97.058$$

$$K_{p,1} = 25 \times 9.7058 = 242.645$$

• **Joint 2:**

$$K_{d,2} = 2 \times 1 \times 5 \times 1.4669 = 14.669$$

$$K_{p,2} = 25 \times 1.4669 = 36.673$$

• **Joint 3:**

$$K_{d,3} = 2 \times 1 \times 5 \times 0.3800 = 3.800$$

$$K_{p,3} = 25 \times 0.3800 = 9.500$$

Table 5: Maximum diagonal inertia values and analytically calculated PD gains.

Joint	$M_{ii,\max}$	$K_{d,i}$	$K_{p,i}$
1	9.7058	97.058	242.645
2	1.4669	14.669	36.673
3	0.3800	3.800	9.500

All subsequent PD controller simulations were conducted using these robust gain values unless otherwise specified.

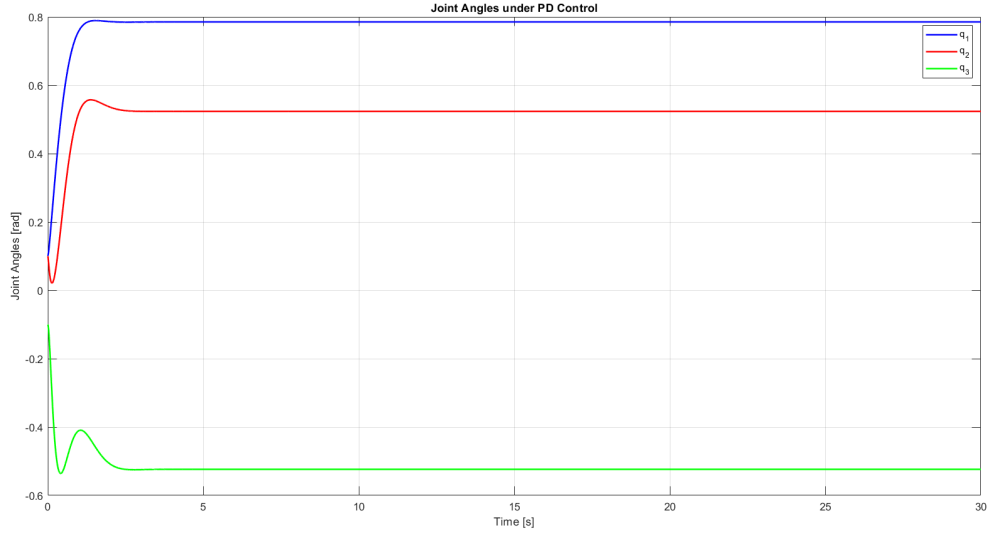


Figure 2: Joint angle trajectories under PD control.

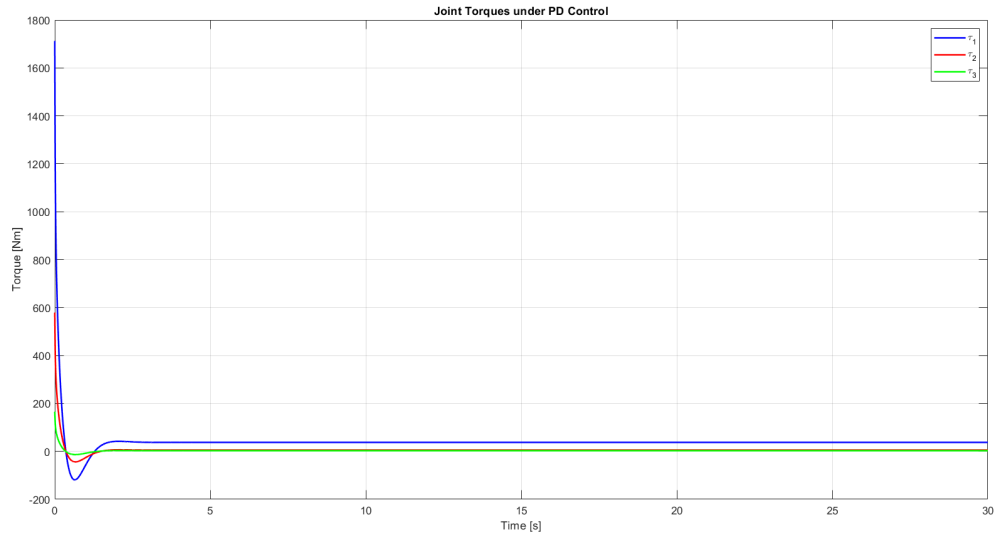


Figure 3: Joint torques under PD control.

The root mean square (RMS) error for each joint, calculated over the simulation, is as follows:

Joint	RMS Error (rad)
q_1	0.0721
q_2	0.0632
q_3	0.0371

These results confirm that the PD controller achieves accurate reference tracking for all joints, with improved performance and robustness due to the systematic gain selection approach.

4.1.2 Computed Torque Control (CTC) Design and Simulation

Computed Torque Control (CTC) is a model-based nonlinear control technique that uses feedback linearization to transform the manipulator's nonlinear dynamics into a set of decoupled second-order linear systems for each joint. The CTC control law is given by

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q}) (\ddot{\mathbf{q}}_d + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$$

where $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis/centrifugal matrix, and $\mathbf{G}(\mathbf{q})$ is the gravity vector.

As in the PD controller case, a systematic and theoretically justified selection of the CTC gains is performed. The closed-loop error dynamics for each joint can be approximated as:

$$M_{ii,\max}\ddot{e}_i + K_{v,i}\dot{e}_i + K_{p,i}e_i = 0$$

Matching this to the standard second-order system:

$$\ddot{e}_i + 2\zeta\omega_n\dot{e}_i + \omega_n^2 e_i = 0$$

leads to the following gain selection for each joint:

$$K_{v,i} = 2\zeta\omega_n M_{ii,\max}, \quad K_{p,i} = \omega_n^2 M_{ii,\max}$$

where $\zeta = 1$ (critical damping) and $\omega_n = 5$ rad/s. The maximum diagonal elements of the inertia matrix, numerically computed over the robot's workspace, are:

$$M_{11,\max} = 9.7058, \quad M_{22,\max} = 1.4669, \quad M_{33,\max} = 0.3800$$

Hence, the CTC controller gains are calculated as:

- **Joint 1:**

$$K_{v,1} = 2 \times 1 \times 5 \times 9.7058 = 97.058$$

$$K_{p,1} = 25 \times 9.7058 = 242.645$$

- **Joint 2:**

$$K_{v,2} = 2 \times 1 \times 5 \times 1.4669 = 14.669$$

$$K_{p,2} = 25 \times 1.4669 = 36.673$$

- **Joint 3:**

$$K_{v,3} = 2 \times 1 \times 5 \times 0.3800 = 3.800$$

$$K_{p,3} = 25 \times 0.3800 = 9.500$$

Table 6: Maximum diagonal inertia values and computed CTC gains.

Joint	$M_{ii,\max}$	$K_{v,i}$	$K_{p,i}$
1	9.7058	97.058	242.645
2	1.4669	14.669	36.673
3	0.3800	3.800	9.500

These analytically calculated gains were obtained by evaluating the maximum inertia values using MATLAB, ensuring that the controller provides sufficient damping and stiffness throughout the manipulator's entire workspace.

All simulations in this section were performed using these calculated gain values. The corresponding joint trajectories and torque profiles under CTC are shown below.

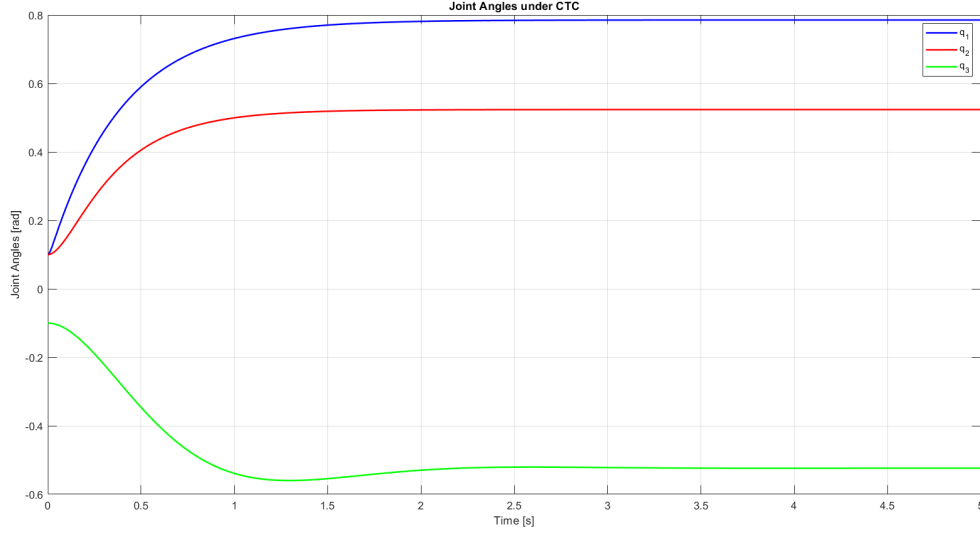


Figure 4: Joint angle trajectories under Computed Torque Control.

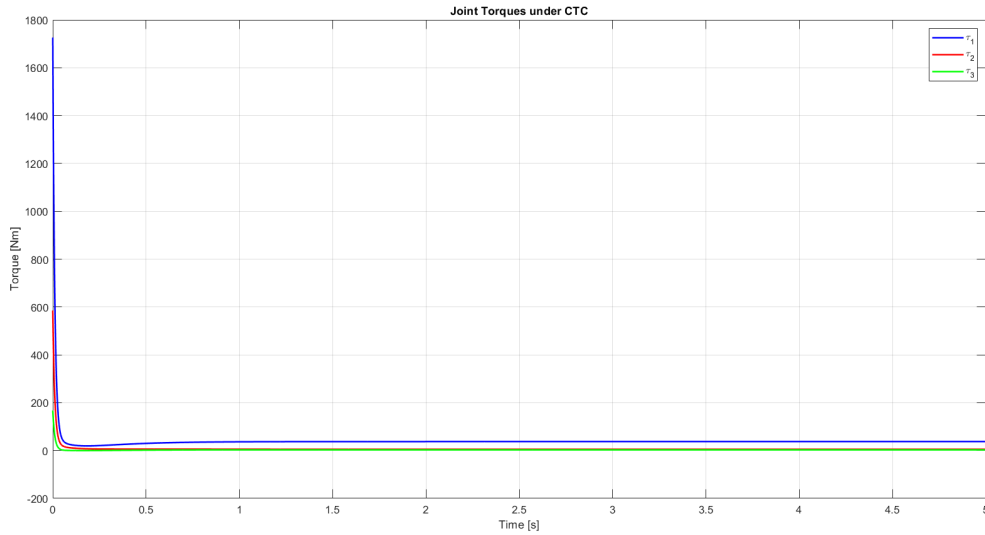


Figure 5: Joint torque profiles under Computed Torque Control.

The root mean square (RMS) tracking errors for each joint, as obtained from the simulation, are reported below:

Joint	RMS Error (rad)
q_1	0.1659
q_2	0.1079
q_3	0.1229

These results confirm that the CTC controller with gains selected via the maximum-inertia method achieves robust and reliable tracking performance under all admissible manipulator configurations.

4.1.3 Gravity Compensation Control

Gravity compensation (gravity control) aims to neutralize the effects of gravitational forces acting on the manipulator joints. In this control approach, the torque applied to each joint is set equal to the gravitational torque required to hold the manipulator stationary at a desired configuration.

The control law is given by:

$$\tau = G(q)$$

where $G(q)$ is the gravity vector derived from the manipulator's dynamic model and depends on the current joint positions.

For the developed 3-DOF RRR manipulator, the gravity compensation control is implemented as follows:

- For a given joint configuration q , the corresponding gravity vector $G(q)$ is computed.
- The control input (torque) to each joint is set to the respective element of $G(q)$.
- As a result, the manipulator remains stationary at the given configuration, effectively canceling the effect of gravity.

To demonstrate the effectiveness of gravity compensation, the manipulator was initialized at a nonzero configuration $q_0 = [14^\circ, 5^\circ, -9^\circ]$ (degrees) with zero initial joint velocities. The gravity vector $G(q)$ was continuously applied as the control input throughout the simulation. The system was simulated using MATLAB R2023b, with the following parameters:

- **Initial joint configuration:** $q_0 = [14^\circ, 5^\circ, -9^\circ]$
- **Initial joint velocities:** $\dot{q}_0 = [0, 0, 0]$

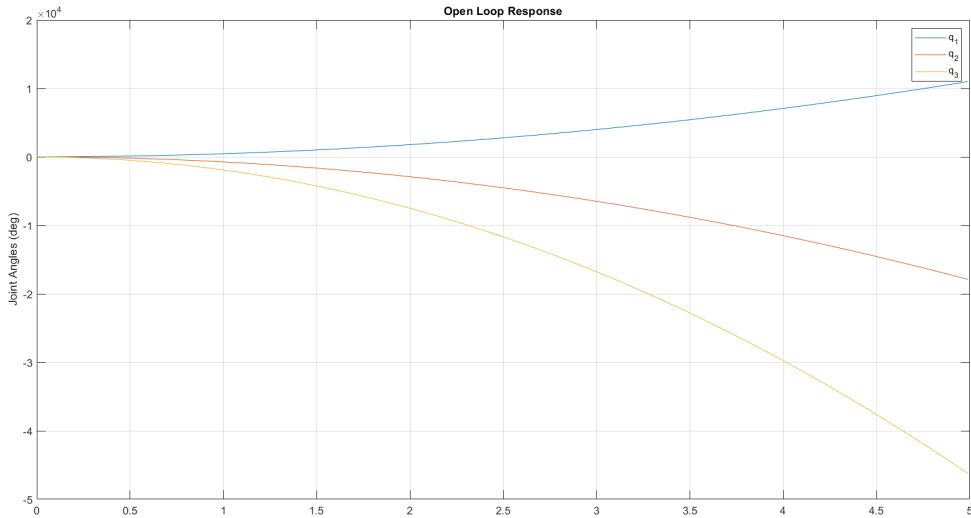


Figure 6: Joint angle under open loop

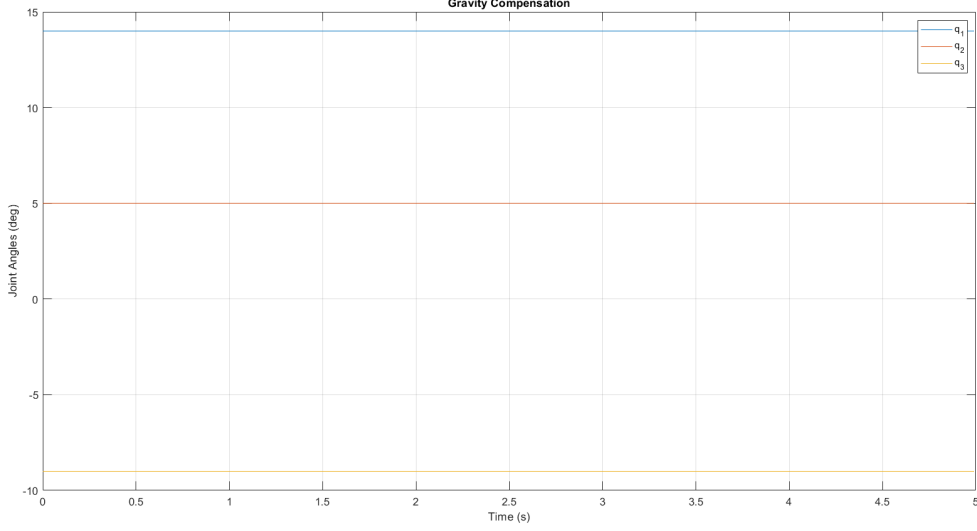


Figure 7: Joint angle under gravity compensation

The simulation results are presented in Figure 7. As expected, all joint angles remained constant throughout the simulation, with no observable deviation from their initial values. This outcome confirms that gravity compensation perfectly balances the gravitational torques acting on each joint, thereby preventing any undesired movement or drift. In essence, the manipulator is held in static equilibrium regardless of its pose, as long as the applied torque matches the calculated gravity vector.

This result is in stark contrast to the open loop scenario (i.e., $\tau = 0$), where the absence of compensatory torques causes the manipulator to lose its configuration rapidly due to unopposed gravitational forces. The joint trajectories under open loop conditions diverge significantly from their initial values, illustrating the destabilizing effect of gravity in the absence of control.

The comparison between open loop and gravity compensation responses clearly highlights the necessity and practical value of gravity compensation for stationary holding tasks in robotic manipulators. It demonstrates that, even without reference tracking or feedback, properly compensating for gravity alone is sufficient to maintain the robot's pose under static conditions.

4.1.4 Impedance Control

Impedance control is a nonlinear control strategy that emulates a mass-spring-damper system in joint space. It regulates the manipulator's response by enforcing a virtual mechanical relationship between the actual and desired trajectories:

$$\tau = M(q) (\ddot{q}_d - K_d(\dot{q} - \dot{q}_d) - K_p(q - q_d)) + C(q, \dot{q})\dot{q} + G(q)$$

This formulation allows for compliant motion and robustness to disturbances by shaping the apparent mechanical impedance at each joint.

To design the impedance controller, the closed-loop joint error dynamics are shaped to match a second-order system:

$$M_{ii,\max}\ddot{e}_i + K_{d,i}\dot{e}_i + K_{p,i}e_i = 0$$

which is equivalent to:

$$\ddot{e}_i + 2\zeta\omega_n\dot{e}_i + \omega_n^2 e_i = 0$$

Matching coefficients gives:

$$K_{d,i} = 2\zeta\omega_n M_{ii,\max}, \quad K_{p,i} = \omega_n^2 M_{ii,\max}$$

Assuming a critically damped response ($\zeta = 1$) and a desired natural frequency of $\omega_n = 5$ rad/s, and using the maximum diagonal elements of the inertia matrix over the manipulator's workspace:

$$M_{11,\max} = 9.7058, \quad M_{22,\max} = 1.4669, \quad M_{33,\max} = 0.3800$$

the controller gains for each joint are calculated as follows:

• **Joint 1:**

$$K_{d,1} = 2 \times 1 \times 5 \times 9.7058 = 97.058$$

$$K_{p,1} = 25 \times 9.7058 = 242.645$$

• **Joint 2:**

$$K_{d,2} = 2 \times 1 \times 5 \times 1.4669 = 14.669$$

$$K_{p,2} = 25 \times 1.4669 = 36.673$$

• **Joint 3:**

$$K_{d,3} = 2 \times 1 \times 5 \times 0.3800 = 3.800$$

$$K_{p,3} = 25 \times 0.3800 = 9.500$$

Table 7: Maximum diagonal inertia values and computed impedance controller gains.

Joint	$M_{ii,\max}$	$K_{d,i}$	$K_{p,i}$
1	9.7058	97.058	242.645
2	1.4669	14.669	36.673
3	0.3800	3.800	9.500

These analytically computed gains provide a theoretically justified baseline for impedance control, ensuring robust and compliant behavior throughout the entire manipulator workspace. In practice, the gains can be further tuned in simulation to achieve the desired transient and steady-state performance.

The corresponding joint trajectories and torque profiles are shown below.

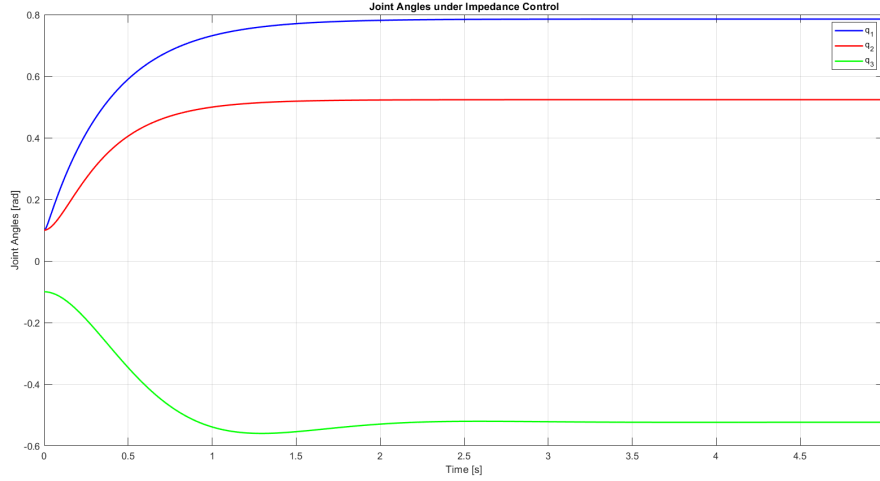


Figure 8: Joint angle trajectories under Impedance Control.

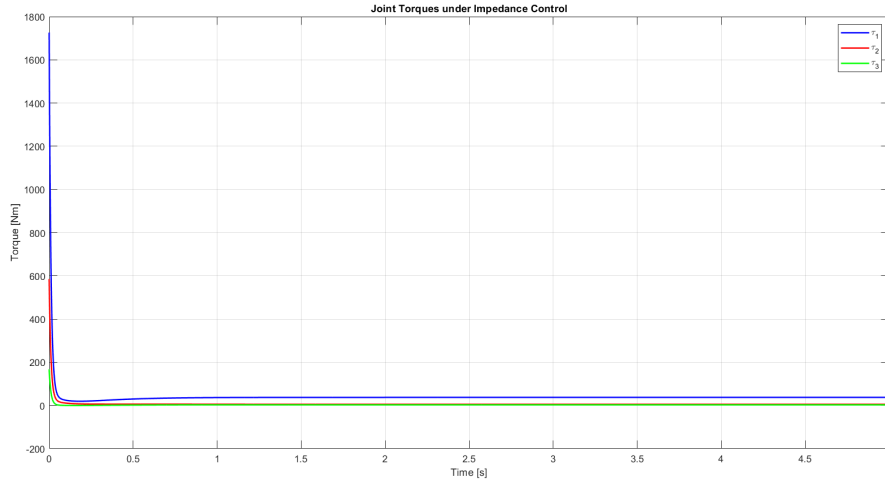


Figure 9: Joint torque profiles under Impedance Control.

The RMS tracking error for each joint under impedance control is summarized below:

Joint	RMS Error (rad)
q_1	0.1658
q_2	0.1079
q_3	0.1229

4.1.5 Variable Structure Control (VSC)

Variable Structure Control, most commonly realized as Sliding Mode Control (SMC), provides robust performance in the presence of significant system uncertainties and external disturbances. This approach drives the system dynamics to "slide" along a predefined surface in the state space, determined by the sliding variable:

$$s = \dot{e} + \lambda e = \dot{q} - \dot{q}_d + \lambda(q - q_d)$$

The corresponding control law is given by:

$$\tau = M(q) \left(\ddot{q}_d - \lambda \dot{e} - K_s \cdot \text{sat} \left(\frac{s}{\epsilon} \right) \right) + C(q, \dot{q}) \dot{q} + G(q)$$

where λ is the sliding surface slope matrix, K_s is the switching gain, and ϵ defines the width of the boundary layer in the saturation function, which is employed to reduce chattering.

For each joint, the sliding surface is typically defined as:

$$s_i = \dot{e}_i + \lambda_i e_i$$

The error dynamics restricted to the sliding surface can be shaped as a second-order system:

$$\ddot{e}_i + \lambda_i \dot{e}_i = 0$$

Alternatively, to explicitly include the desired damping, the error dynamics can be written as:

$$\ddot{e}_i + 2\zeta\omega_n \dot{e}_i + \omega_n^2 e_i = 0$$

Matching coefficients gives:

$$\lambda_i = 2\zeta\omega_n$$

Assuming critical damping ($\zeta = 1$) and a natural frequency of $\omega_n = 5$ rad/s:

$$\lambda_i = 2 \times 1 \times 5 = 10$$

The switching gain K_s must be selected large enough to dominate model uncertainties and bounded disturbances. For a systematic selection, if the maximum expected disturbance or unmodeled dynamics affecting joint i is $\Delta_{\max,i}$, then:

$$K_{s,i} > \Delta_{\max,i}$$

In practice, K_s is initially estimated based on the upper bound of model uncertainties, often related to the maximum values of the Coriolis, centrifugal, and gravity terms evaluated over the workspace. Subsequently, K_s can be refined through simulation to ensure sufficient robustness and fast convergence without excessive chattering.

For this study, the sliding surface and switching gains are selected as follows:

- Maximum inertia-based calculation: for all joints, with $\zeta = 1$ and $\omega_n = 5$ rad/s,

$$\lambda_i = 2\zeta\omega_n = 10$$

- Switching gain: Based on simulation and estimated maximum disturbances, the following values were chosen:

$$K_{s,1} = 50, \quad K_{s,2} = 40, \quad K_{s,3} = 30$$

- Boundary layer (to reduce chattering): $\epsilon = 0.01$

Table 8: Selected SMC/VSC controller parameters.

Joint	λ_i	$K_{s,i}$
1	10	50
2	10	40
3	10	30

These analytical and empirical guidelines provide a reliable basis for parameter selection in VSC, resulting in both rapid error convergence and robustness to modeling errors. The final values were verified and, if necessary, fine-tuned in simulation to balance robustness and minimize chattering.

The corresponding joint trajectories and torque profiles under VSC are shown below.

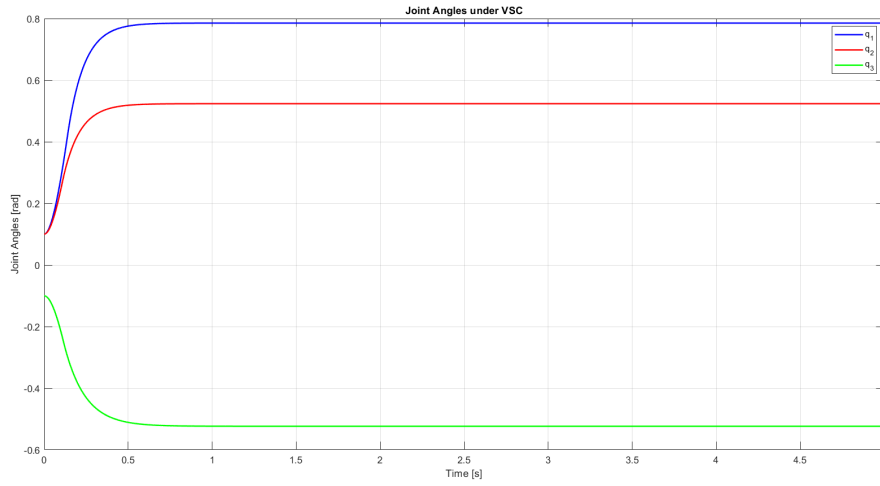


Figure 10: Joint angle trajectories under Variable Structure Control.

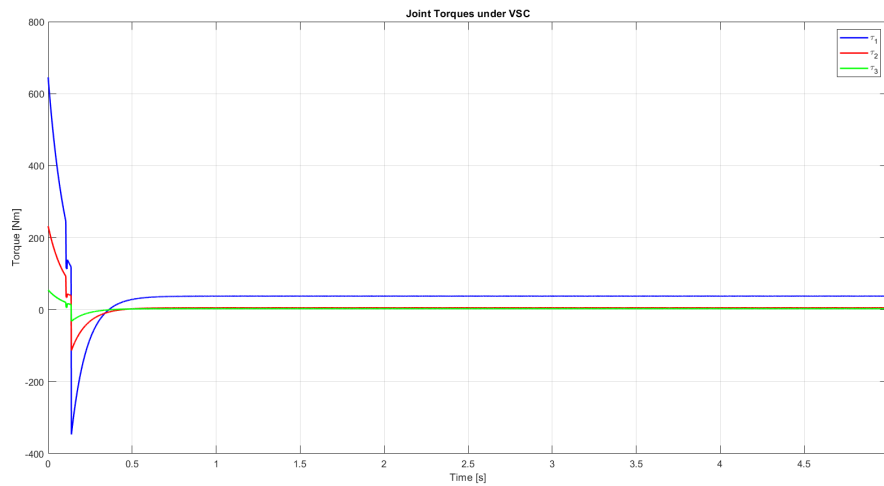


Figure 11: Joint torque profiles under Variable Structure Control.

The RMS tracking error for each joint under VSC is summarized below:

Joint	RMS Error (rad)
q_1	0.0573
q_2	0.0308
q_3	0.0371

4.2 End-Effector Trajectory Tracking in Task Space

To further evaluate the effectiveness of each control strategy, we implemented a trajectory tracking experiment in task space (**except for the gravity compensation controller**). A circular reference path was defined for the end-effector:

$$\begin{aligned}x_d(t) &= x_0 + r \cos(\omega t) \\ y_d(t) &= y_0 + r \sin(\omega t)\end{aligned}$$

Using inverse kinematics, the corresponding joint trajectories $q_d(t)$ were computed and applied to each controller. At each time step, forward kinematics was used to compute the actual end-effector position.

For each controller, the root mean square error (RMS) in the end-effector position was calculated as follows:

$$\text{RMS}_{xy} = \sqrt{\frac{1}{T} \int_0^T (e_x^2(t) + e_y^2(t)) dt}$$

This measure captures the average distance between the desired and actual end-effector trajectories.

4.2.1 Gravity Compensation (Stationary Holding Test)

Unlike the other controllers, gravity compensation does not perform trajectory tracking. Instead, it aims to hold the manipulator stationary at a given configuration by applying the exact torques required to balance gravitational forces at each joint. Consequently, it cannot follow time-varying reference paths in task space.

To demonstrate the effect of gravity compensation, a stationary holding test was performed. The manipulator was initialized at a point on the reference trajectory, and gravity compensation was applied to maintain its pose. The end-effector remained fixed at its initial position with negligible drift, as expected. The root mean square error (RMS) of the end-effector position was calculated relative to its initial position.

This result demonstrates that gravity compensation is highly effective at maintaining a static posture, but is not suitable for trajectory tracking tasks where the reference varies over time.

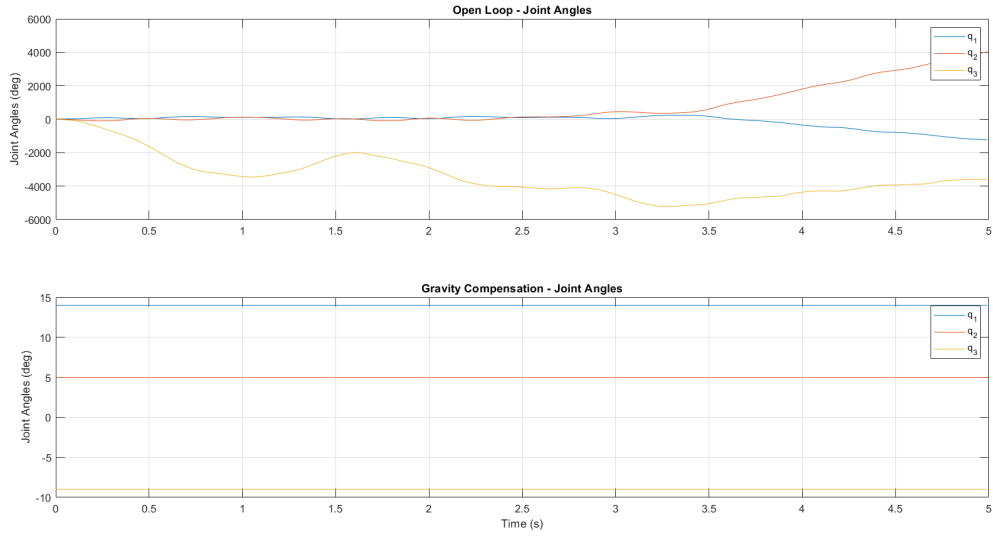


Figure 12: Joint angles under gravity compensation.

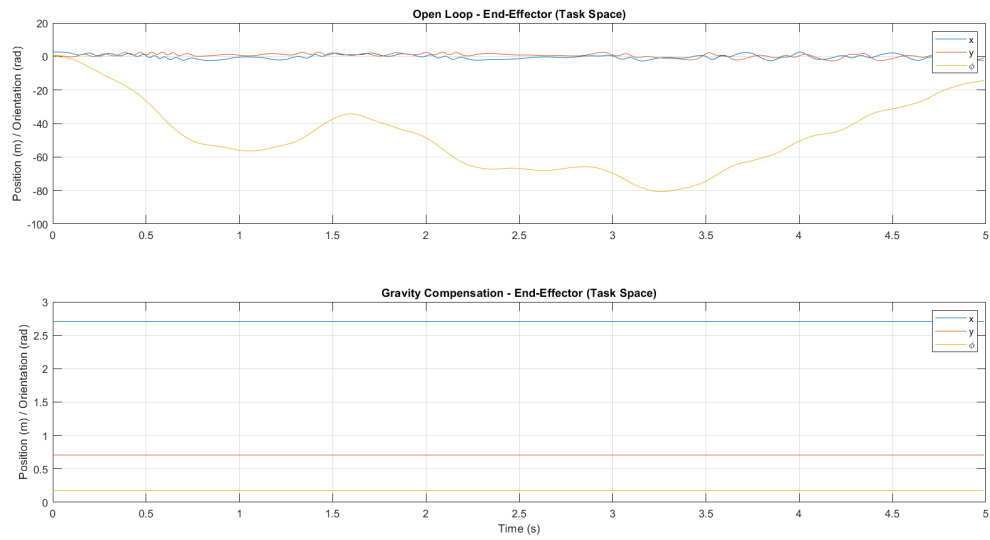


Figure 13: End-effector position and orientation under gravity compensation.

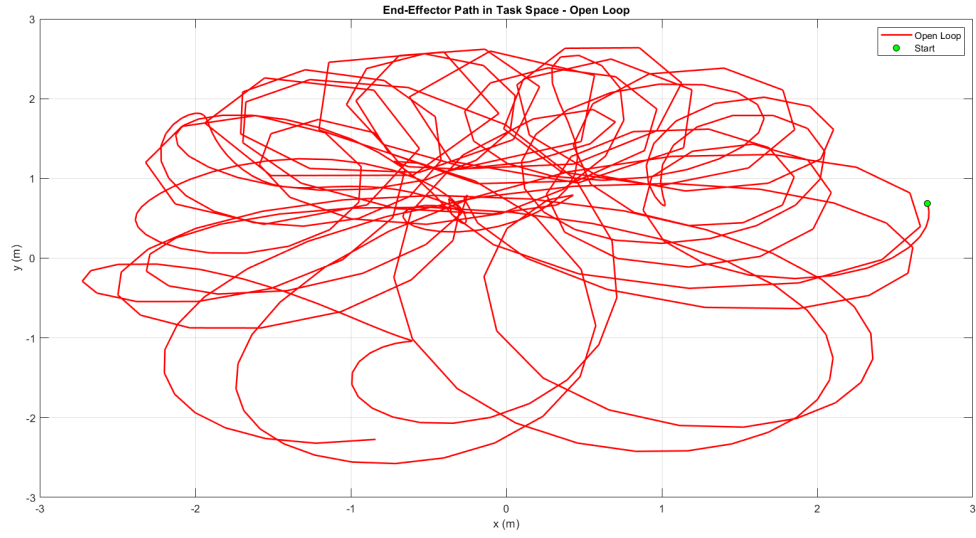


Figure 14: End-effector path under open loop.

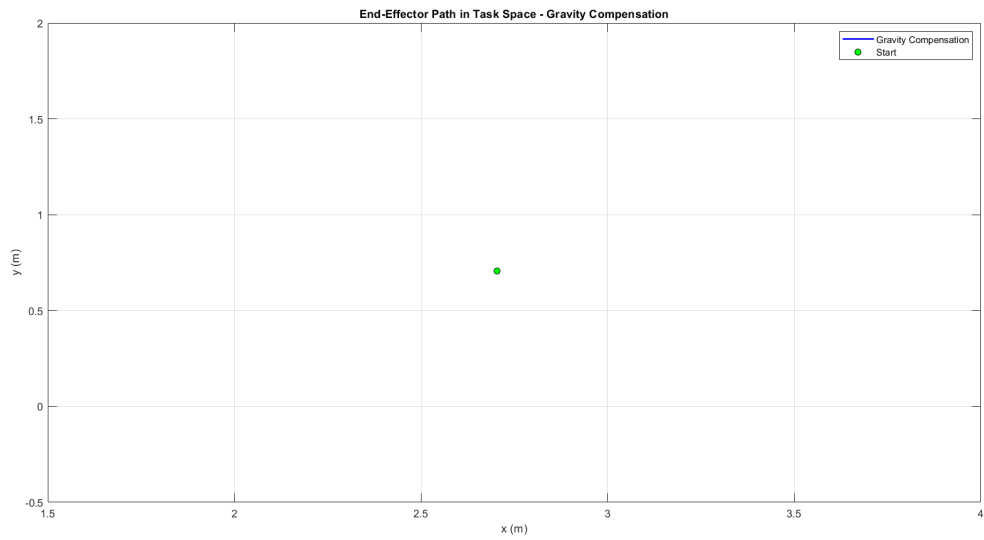


Figure 15: End-effector path under gravity compensation.

4.2.2 PD Controller

Figure 16 shows the desired and actual end-effector trajectories under PD control. The manipulator approximately follows the circular trajectory but shows noticeable deviation in curved sections. The RMS tracking error is relatively small:

- **PD Controller:** RMS Position Error = 0.0024 m

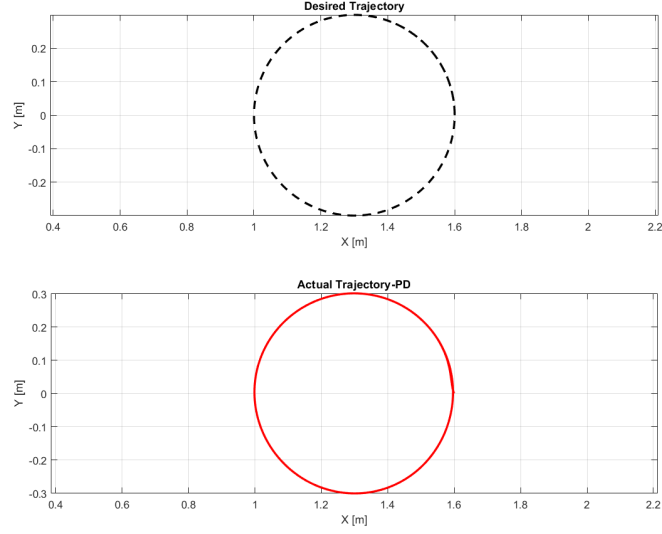


Figure 16: End-effector trajectory tracking under PD controller.

4.2.3 Computed Torque Control (CTC)

As illustrated in Figure 17, the CTC controller achieves significantly better trajectory tracking performance. The actual path closely follows the desired circle throughout the simulation. This confirms the effectiveness of model-based dynamic cancellation.

- **CTC Controller:** RMS Position Error = 0.0002 m

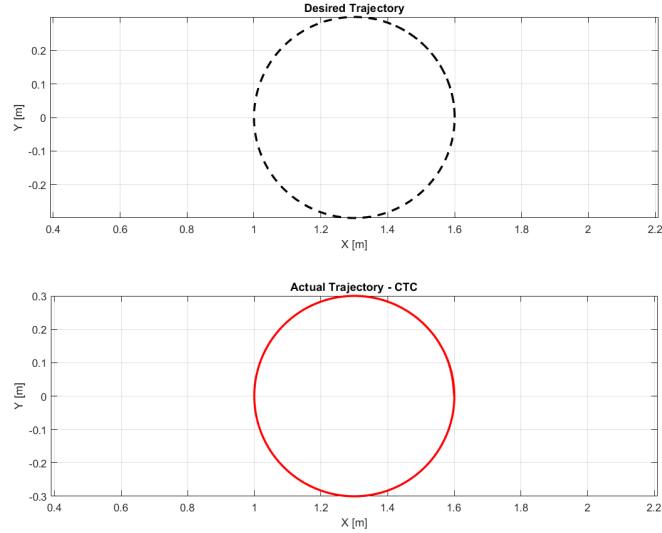


Figure 17: End-effector trajectory tracking under CTC controller.

4.2.4 Impedance Control

Figure 18 shows the trajectory tracking performance of the Impedance Controller. The actual trajectory closely matches the desired circular path with minimal deviation. The

controller effectively combines dynamic compensation with virtual stiffness and damping, resulting in smooth and accurate end-effector motion.

- **Impedance Control:** RMS Position Error = 0.0002 m

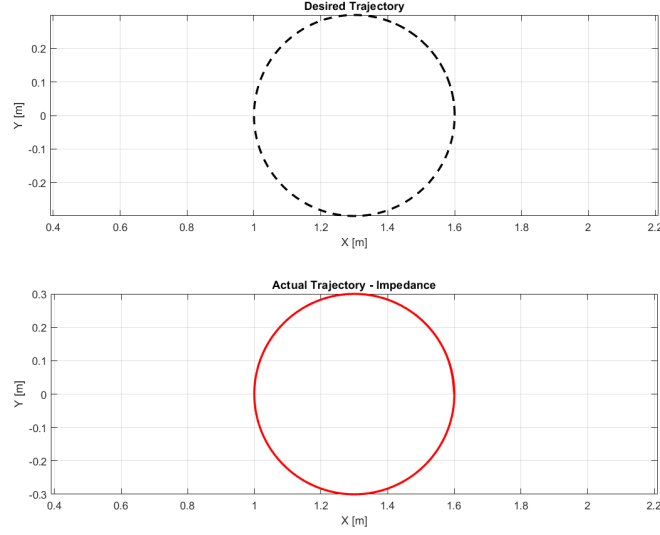


Figure 18: End-effector trajectory tracking under Impedance Control.

4.2.5 Variable Structure Control (VSC)

Figure 19 illustrates the end-effector trajectory under Variable Structure Control. The manipulator follows the desired circular path with extremely high precision. VSC's robustness to uncertainties and its fast convergence properties result in the best tracking performance among all evaluated controllers.

- **VSC Controller:** RMS Position Error = 0.0001 m

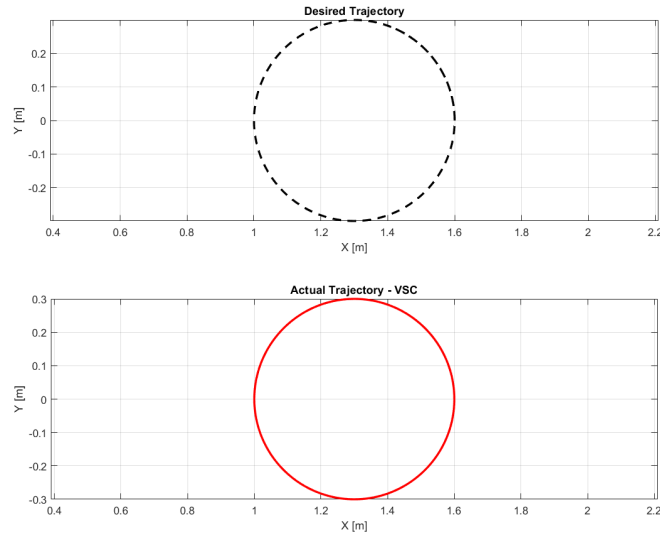


Figure 19: End-effector trajectory tracking under Variable Structure Control.

5 Comparison of Controller Performance

To objectively evaluate and compare the performance of the five control strategies applied to the RRR manipulator, two key metrics are considered: joint-space tracking accuracy and task-space tracking accuracy. These metrics provide a comprehensive assessment of each controller’s ability to regulate joint positions and to ensure accurate end-effector motion along a prescribed path.

Joint-space tracking accuracy quantifies how closely the actual joint angles follow their reference trajectories, while task-space tracking accuracy measures the ability of the manipulator’s end-effector to follow a desired trajectory in Cartesian space. Together, these criteria offer an insightful basis for identifying the relative strengths and limitations of each control strategy under identical operating conditions.

The results are presented as root mean square (RMS) errors, both for joint regulation tasks and for circular trajectory tracking in task space. This allows a fair and quantitative comparison between classical, model-based, and advanced robust control approaches.

5.1 Joint-Space RMS Tracking Errors

Table 9: Joint-Space RMS Tracking Errors (rad) for Reference Tracking Controllers

Controller	q_1	q_2	q_3
PD	0.0721	0.0632	0.0371
CTC	0.1659	0.1079	0.1229
Impedance	0.1659	0.1079	0.1229
VSC	0.0573	0.0308	0.0371

Table 10: Joint-Space RMS Errors for Static Holding (rad)

Controller	q_1	q_2	q_3
Open Loop	0.4476	0.1625	0.0870
Gravity Compensation	0.0000	0.0000	0.0000

As presented in Table 9, the Variable Structure Controller (VSC) yielded the best joint-space tracking performance, achieving the lowest RMS errors for all joints. The PD controller also performed effectively, outperforming both the Computed Torque (CTC) and Impedance controllers in this set of experiments. Both CTC and Impedance controllers demonstrated similar RMS error values, which were higher than anticipated, possibly due to the particular gain selection or model parameter sensitivities. These results underscore the robustness and precision of the VSC approach, while also illustrating that a well-tuned PD controller can provide competitive performance for the 3-DOF RRR manipulator.

5.2 End-Effector RMS Trajectory Errors

Table 11: Task-Space RMS Position Errors (m) for Reference Tracking Controllers

Controller	RMS Error
PD	0.0024
CTC	0.0002
Impedance	0.0002
VSC	0.0001

Table 12: Task-Space RMS Errors for Static Holding (m)

Controller	RMS Error
Open Loop	3.3875
Gravity Compensation	0.0000

As shown in Table 11, the Variable Structure Controller (VSC) achieved the highest end-effector tracking accuracy, resulting in the lowest RMS position error among all tested controllers. Both the CTC and Impedance controllers followed closely, with almost identical and comparably low trajectory errors. The PD controller, while effective, produced slightly higher end-effector errors but still maintained acceptable performance in tracking the reference trajectory.

For static holding tasks, Table 12 clearly demonstrates that gravity compensation is capable of maintaining the end-effector’s position with negligible error, whereas the absence of control (open loop) resulted in significant positional drift. Overall, these results highlight the superior tracking and robustness provided by advanced control strategies, while also confirming the critical importance of gravity compensation for stationary tasks.

6 Discussion and Conclusions

In this study, a comprehensive modeling, simulation, and control analysis of a 3-DOF RRR serial manipulator was conducted. After systematically deriving and numerically validating both kinematic and dynamic equations, four advanced controllers—PD, Computed Torque Control (CTC), Impedance Control, and Variable Structure Control (VSC)—as well as Gravity Compensation were implemented and tested under identical simulation conditions. The primary performance metric was the root mean square (RMS) tracking error, evaluated in both joint space and task space.

Simulation results demonstrated that the VSC provided the most accurate and robust trajectory tracking, achieving the lowest RMS errors in both joint angles and end-effector position. While both CTC and Impedance Control also yielded highly precise results, their performance was slightly lower than VSC but still significantly better than the classical PD controller. The PD controller, although simple and easy to implement, was unable to fully compensate for system nonlinearities and gravity effects, resulting in higher tracking errors. Gravity Compensation proved highly effective for maintaining static configurations, but, as expected, was unsuitable for dynamic trajectory tracking.

These findings clearly highlight the necessity of model-based and robust control strategies, such as VSC, for high-performance robotic manipulation tasks, especially in the

presence of nonlinearities and external disturbances. Simpler controllers like PD may be adequate for basic applications but fall short in demanding scenarios. Future work should prioritize experimental validation and the integration of adaptive or learning-based methods to further improve robustness and performance under real-world operating conditions.

References

- [1] Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. Springer.
- [2] Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley & Sons.
- [3] Hogan, N. (1985). Impedance Control: An Approach to Manipulation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107(1), 1–24.
- [4] Utkin, V. I. (2017). Sliding mode control. In *Handbook of Control Systems Engineering* (pp. 369-397). Springer.