Advanced Control Systems and Dynamics Assignment

By

KULKARNI , OM

Department of Engineering Design and Mathematics
UNIVERSITY OF THE WEST OF ENGLAND

A MSc coursework assignment submitted to the University of the West of England in accordance with the requirements of the degree of MASTER OF SCIENCE IN ROBOTICS in the Faculty of Engineering.

July 8, 2025

Abstract

This report details the design, implementation, and analysis of a Computed Torque Controller (CTC) for trajectory tracking of a rigid 2-link serial manipulator. The primary objective was to develop the manipulator's dynamic model and apply CTC, a feedback linearization technique, to achieve accurate tracking of a predefined joint-space trajectory. The control strategy involved deriving the nonlinear dynamic equations (Mass, Coriolis/Centrifugal, and Gravity terms) and formulating the CTC law. Proportional-Derivative (PD) gains for the outer-loop error dynamics were tuned to meet performance criteria, and the asymptotic stability of the closed-loop error system was demonstrated using Lyapunov analysis. System performance was evaluated through simulations, including tests assessing the controller's ability to reject time-varying external torque disturbances. Results indicate successful trajectory tracking, achieving a effective disturbance rejection, thereby demonstrating the viability and effectiveness of the CTC approach for controlling the nonlinear dynamics of the 2-link manipulator.

Contents

1 Introduction

1.1 Background

Figure 1.1: Industrial Robot Arm.

The inspiration behind robot manipulators was to mimic the human arm. They are programmable machines, generally designed to move objects or perform specific tasks [1]. They consist of a series of rigid bodies (links) connected by joints (like revolute/rotating or prismatic/sliding joints) that allow relative motion. An end-effector, specific to the task (e.g., a gripper, welder, or camera), is typically attached to the end of the manipulator chain. General purpose robot arms are currently being tested in academia but have not yet found there way into industry and manufacturing.

Controlling these manipulators accurately is challenging due to their complex dynamics, which are often highly nonlinear (effects depend non-proportionally on the state) and coupled (motion of one joint affects others). This necessitates sophisticated control strategies, like the Computed Torque Control, to achieve desired performance.

1.2 Problem Statement

This project addresses the control of a 2-link rigid serial manipulator operating in the vertical plane. Given the manipulator's constant physical parameters (link lengths, masses, and moments of inertia), the first objective is to derive its dynamic equations of motion using the Euler-Lagrange formulation. The dynamic equations can be expressed as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau - \tau_{\text{ext}}(t) \tag{1.1}$$

The terms in this equation are explained in the Section -2

The primary control objective is to design and implement a controller that forces the manipulator's joint angles $q(t)$ to accurately track a specified time-varying reference trajectory $q_{ref}(t) = [q_{ref1}(t), q_{ref2}(t)]^T$, despite the presence of external disturbances. The control architecture employs Computed Torque Control (CTC), formulated as:

$$\tau = M(q)[\ddot{q}_{ref} + K_d\dot{e} + K_p e] + C(q,\dot{q})\dot{q} + G(q) \tag{1.2}$$

1.3 Objectives and Report Outline

The primary objectives of this work are to:

- Derive the dynamic model of the specified 2-link manipulator using the Euler-Lagrange method.
- Design a Computed Torque Controller (CTC) to enable the manipulator to track a given joint-space trajectory.
- Tune the controller's Proportional-Derivative (PD) gains for suitable performance.
- Analyze the stability of the resulting closed-loop system via Lyapunov's method.
- Simulate the system to evaluate tracking performance and the controller's ability to reject external disturbances.

This report details these steps. Section 2 presents the system modeling and derived dynamic equations. Section 3 describes the CTC design process, including gain tuning. Section 4 covers the stability analysis. The results, including performance evaluation and disturbance rejection, are presented in Section 5. Finally, Section 6 concludes the report, summarizing findings.

2 System Modeling

2.1 Manipulator Description

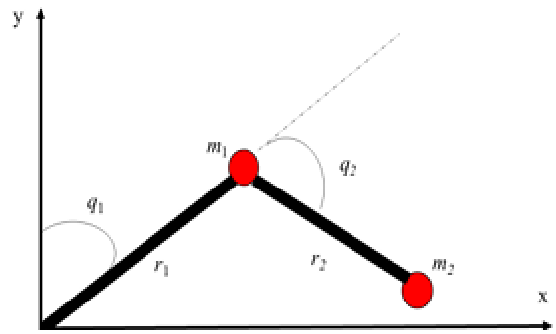


Figure 2.1: 2-Link Robot Arm Diagram.

The system under consideration is a two-link (2-DOF) serial robotic manipulator operating in the vertical x-y plane, as illustrated in Figure 2.1. The manipulator consists of two rigid links connected by revolute joints, forming an RR configuration.

- Link 1: Has length r1 and is connected to the base at the origin (0,0) via the first revolute joint. Its orientation is defined by the angle q1, measured counter-clockwise from the positive x-axis. A mass m1 is located at the distal end of this link (at the second joint).
- Link 2: Has length r2 and connects the end of Link 1 to the end-effector via the second revolute joint. Its orientation is defined by the angle q2, measured counter-clockwise relative to the orientation of Link 1. A mass m2 is located at the distal end of this link (the end-effector point).

The configuration of the manipulator is fully described by the joint angle vector $q = [q_1, q_2]^T$ and the joint torques τ_1 and τ_2, are applied at Joint 1 and Joint 2, respectively. The presence of gravity acting in the negative y-direction will be considered in the dynamic model. [2]

Table 2.1: Fixed Parameters for the Two-Link Manipulator

Parameter	**Value**	**Unit**
r_1	1.0	m
r_2	0.8	m
J_1	5.0	kg·m^2
J_2	5.0	kg·m^2
m_1	0.5	kg
m_2	1.5	kg
g	9.8	m/s^2

2.2 Parameters

The joint trajectory is a function of time(t) is given as $q_{\text{ref}} = [q_{\text{ref1}}, q_{\text{ref2}}]^T$ where:

$$q_{\text{ref1}}(t) = 1.25 - \frac{7}{5}e^{-t}$$
$$q_{\text{ref2}}(t) = 1.25 + e^{-t} - \frac{1}{4}e^{-4t}$$

and the external Torque Disturbances were a function of time given by:

$$T_{\text{ext1}}(t) = 1.25 - \frac{7}{5}e^{-t}$$
$$T_{\text{ext2}}(t) = 1.25 + e^{-t} - \frac{1}{4}e^{-4t}$$

2.3 Dynamic Equations of Motion

The standard form of the dynamic model was derived through the Euler-Lagrange equation. This section just lists the standard form, defining the $M(q)$, $C(q,\dot{q})$ and $G(q)$ terms. [3] The derivation of these terms is given in the *params.m* file in the appendix.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau - \tau_{\text{ext}}(t) \tag{2.1}$$

- $q(t) = [q_1(t), q_2(t)]^T$ represents the joint angles

- $M(q)$ is the inertia matrix given by $\mathbf{M}(\mathbf{q}) = \begin{bmatrix} \frac{12}{5}\cos(q_2) + \frac{324}{25} & \frac{6}{5}\cos(q_2) + \frac{149}{25} \\ \frac{6}{5}\cos(q_2) + \frac{149}{25} & \frac{149}{25} \end{bmatrix}$

- $C(q, \dot{q})$ contains Coriolis and centrifugal terms given by $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\frac{12}{5}\sin(q_2)\dot{q}_2 & -\frac{6}{5}\sin(q_2)\dot{q}_2 \\ \frac{6}{5}\sin(q_2)\dot{q}_1 & 0 \end{bmatrix}$

- $G(q)$ accounts for gravitational forces given by $\mathbf{G}(\mathbf{q}) = \begin{bmatrix} -\frac{98}{5}\sin(q_1) - \frac{294}{25}\sin(q_1 + q_2) \\ -\frac{294}{25}\cos(q_1 + q_2) \end{bmatrix}$

- τ is the control torque input

- $\tau_{\text{ext}}(t)$ represents known external disturbances

This equation will help model both the Computed Torque Control and also the manipulator forward dynamics in the upcoming section.

3 Computed Torque Control

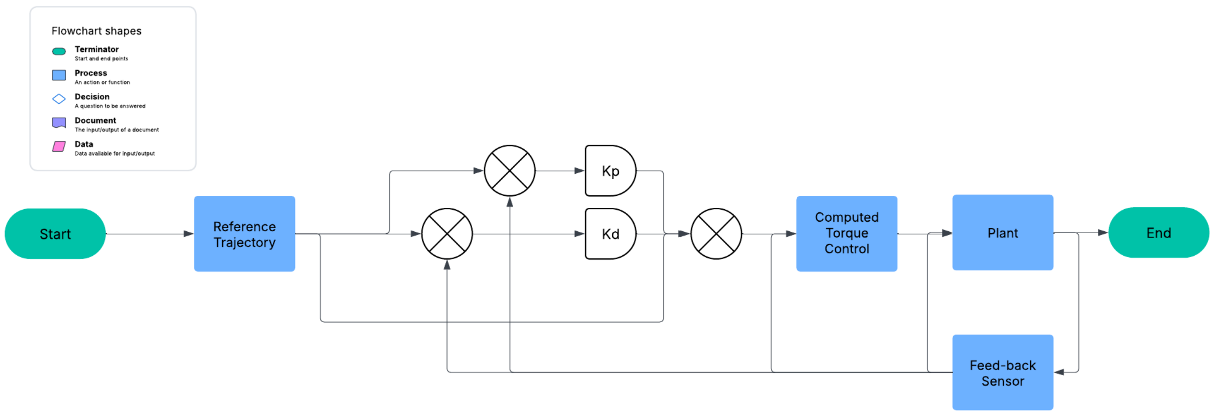


Figure 3.1: Diagram of the Computed-Torque-Control Sytem

3.1 Control Law

The Computed Torque Control (CTC) is a widely used technique in robotics for trajectory tracking. Its primary goal is to compensate for the highly nonlinear and coupled dynamics of robot manipulators, making the system behave like a simpler, linear system.[4]

The control law for Computed Torque Control is typically expressed as:

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

This equation is almost the same as the dynamic equation of motion except for two terms. $\ddot{q}_{des}$ is the desired joint acceleration vector, which is determined by a linear feedback control law acting on the tracking error. We will consider a PD-control for calculating this term.

Secondly, the term $\tau_{\text{ext}}(t)$ is missing here as this term encapsulates the noise within the system and is unknown to the controller.

The $\sim$ term over $M(q), C(q,\dot{q})$ and $G(q)$ suggests that they do not need to be exactly the same as the ones in the dynamic motion equation. The only constraint over them is that $\tilde{M}(q)^{-1} \cdot M(q) \approx 1$ and $M^{-1} \cdot (C(q,\dot{q}) + G(q)) \approx M^{-1} \cdot (\tilde{C}(q,\dot{q}) + \tilde{G}(q))$

3.2 Desired Acceleration and PD-Control

The desired acceleration $\ddot{q}_{des}$ is the key component that dictates how the robot tracks the desired trajectory. It is typically defined based on the desired trajectory and feedback from the current state (position and velocity) of the robot to correct for tracking errors. [5]

A common choice for $\ddot{q}_{des}$ is given by a linear feedback control law, such as a PD (Proportional-Derivative) controller applied to the tracking error:

$$\ddot{q}_{\text{des}} = \ddot{q}_d + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q})$$

By appropriately choosing the positive definite gain matrices K_p and, K_d the error dynamics can be made stable, ensuring that the tracking error e converges to zero over time. The term $\ddot{q}_d$ acts as a feedforward term, anticipating the required acceleration, while the feedback terms $K_p(q_d - q)$ and $K_d(\dot{q}_d - \dot{q})$ correct for deviations from the desired trajectory based on position and velocity errors.

3.3 PD Gain Selection

Empirical tuning, often called "trial and error" tuning, is a practical method for setting the gains of a Proportional-Derivative (PD) controller directly on the physical system. It relies on observing the system's response to changes in the gains and iteratively adjusting them until a desired performance is achieved. [6]

The general steps followed were:

1. Starting with $K_d = 0$ and tuning K_p to achieve a desired level of responsiveness, potentially observing the onset of oscillations.

2. Introducing and tuning K_d to damp oscillations, reduce overshoot, and improve settling time.

3. Iteratively fine-tuning both K_p and K_d to balance responsiveness, stability, and tracking accuracy based on observed performance.

The final tuned values were:

$$K_p = \begin{bmatrix} 251 & 0 \\ 0 & 997 \end{bmatrix} K_d = \begin{bmatrix} 51 & 0 \\ 0 & 673 \end{bmatrix}$$

3.4 Lyapunov Stability Analsis

Lyapunov stability analysis is a powerful tool for proving the stability of nonlinear systems without explicitly solving their differential equations. For Computed Torque Control (CTC), it can be used to demonstrate that the tracking error converges to zero, meaning the robot follows the desired trajectory.

The core idea is to find a Lyapunov candidate function, which is a scalar function of the system's state (in this case, the tracking error) that is positive definite and whose time derivative along the system's trajectories is negative semi-definite or negative definite.

3.4.1 Lyapunov Candidate Function

A common and effective Lyapunov candidate function for mechanical systems is a quadratic form related to the energy of the error system. [7] Consider the error state vector $x_e = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$. A suitable candidate function is:

$$V(x_e) = \frac{1}{2}\dot{e}^T \dot{e} + \frac{1}{2}e^T K_p e \tag{3.1}$$

This function is:

- Positive definite when $K_p \succ 0$ (positive definite matrix)
- Typically implemented with K_p as a diagonal matrix of positive gains
- Equal to zero *only* when $e = 0$ and $\dot{e} = 0$

Differentiate $V(x_e)$ with respect to time:

$$\dot{V}(x_e) = \frac{d}{dt}\left(\frac{1}{2}\dot{e}^T \dot{e} + \frac{1}{2}e^T K_p e\right) \tag{3.2}$$

$$= \dot{e}^T \ddot{e} + \frac{1}{2}\left(\dot{e}^T K_p e + e^T K_p \dot{e}\right) \tag{3.3}$$

$$= \dot{e}^T \ddot{e} + \dot{e}^T K_p e \tag{3.4}$$

$$= \dot{e}^T \left(\ddot{e} + K_p e\right) \tag{3.5}$$

Substitute the Error Dynamics into the Derivative using the closed-loop error dynamics:

$$\ddot{e} = -K_p e - K_d \dot{e} \tag{3.6}$$

Substitute into (3.4):

$$\begin{aligned}\dot{V}(x_e) &= \dot{e}^T \left(-K_p e - K_d \dot{e} + K_p e\right) \\ &= -\dot{e}^T K_d \dot{e} \leq 0\end{aligned}$$

Given $K_d \succ 0$ (positive definite, typically diagonal with positive entries), the quadratic form $\dot{e}^T K_d \dot{e}$ satisfies:

- $\dot{e}^T K_d \dot{e} > 0 \;\; \forall \dot{e} \neq 0$
- $\dot{e}^T K_d \dot{e} = 0 \;\; \text{iff} \;\; \dot{e} = 0$

This implies the time derivative of the Lyapunov function:

$$\dot{V}(x_e) = -\dot{e}^T K_d \dot{e} \leq 0 \tag{3.7}$$

is negative semi-definite.

Stability Results:

1. The closed-loop system is *stable in the sense of Lyapunov*
2. Tracking errors remain bounded: $\|e\| < \infty$, $\|\dot{e}\| < \infty$

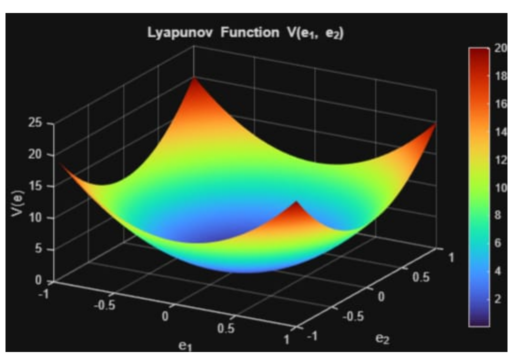
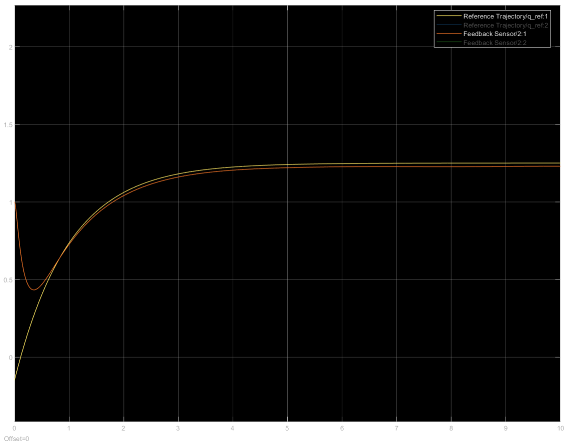
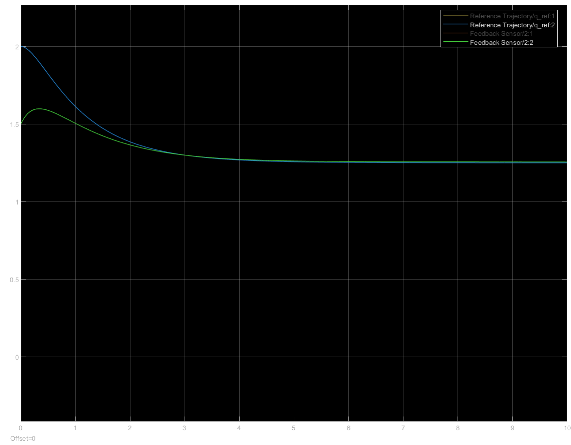


Figure 3.2: Lyapunov Candidate Function vs 1st and 2nd Component of error term

4 Results



(a) 1st Component of Positional Error term(e)



(b) 2nd Component of Positional Error term(e)

Figure 4.1: Positiona Error Term (e) vs Time

The observed tracking performance (Fig. 4.1) demonstrates:

Effective trajectory following:

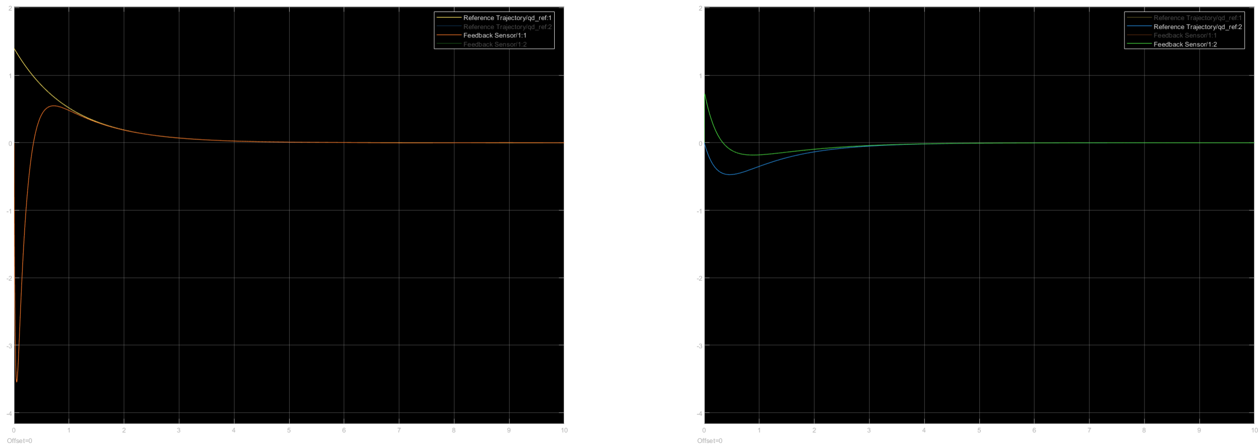
$$\lim_{t \to \infty} \|q_d(t) - q(t)\| \approx 0$$

This experimental validation confirms the CTC-PD controller successfully:

- Compensates for nonlinear dynamics through the computed torque framework
- Maintains stability via the PD feedback linearization
- Achieves asymptotic tracking per Lyapunov stability analysis

4.0.1 Velocity Tracking

Comparing velocity tracking graph to the position tracking graph, the velocity tracking shows more dynamic behavior and initial errors, which is typical. The position errors are the integral of the velocity errors, and the smoothness of the position tracking suggests that while there are velocity deviations, they are managed effectively by the controller to ensure accurate position tracking.



(a) 1st Component of Velocity Error term($\dot{e}$) (b) 2nd Component of Velocity Error term($\dot{e}$)

Figure 4.2: Velocity Error Term ($\dot{e}$) vs Time

4.0.2 Torque Tracking

This graph provides insight into the control effort required by the actuators. High peak torques suggest that the system demands significant power during rapid movements or when correcting large initial errors. The behavior of the torques is consistent with the observed position and velocity tracking, where initial errors are quickly reduced.

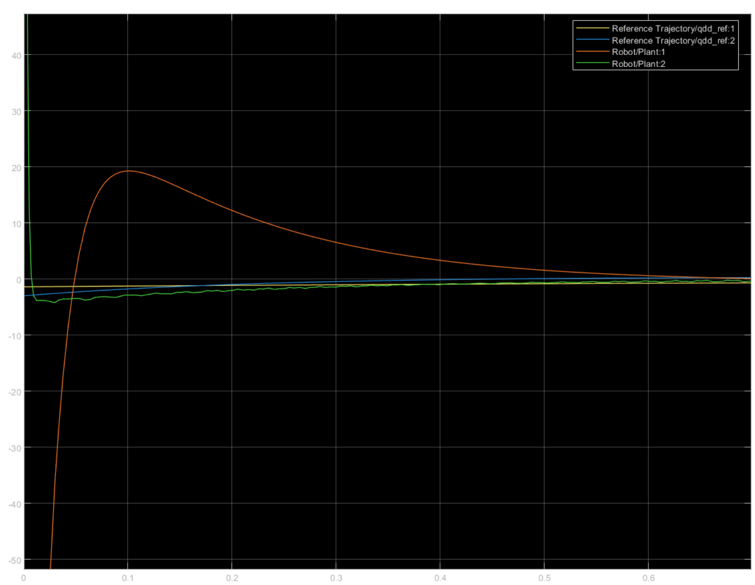


Figure 4.3: Comparison Graph between the commanded torque and the system output torque

5 Discussion and Conclusion

5.0.1 Discussion of Initial Errors

The position, velocity, and consequently the torque graphs all show large initial errors at time t=0. This is a direct result of the mismatch between the robot's actual initial state (position and velocity) and the initial state defined by the reference trajectory function.

The problem statement specifies the initial joint states as:

- $q_1(0) = 1.0\ rad$
- $\dot{q}_1(0) = 0\ rad/s$
- $q_2(0) = 1.5\ rad$
- $\dot{q}_2(0) = 0\ rad/s$

However, evaluating the provided reference trajectory functions at $t = 0$

$$q_{ref1}(0) = 1.25 - \frac{7}{5}e^0 = 1.25 - 1.4 = -0.15\ rad$$
$$q_{ref2}(0) = 1.25 + e^0 - \frac{1}{4}e^0 = 1.25 + 1 - 0.25 = 2.0\ rad$$

The initial position errors are therefore:

- $e_1(0) = q_{ref1}(0) - q_1(0) = -0.15 - 1.0 = -1.15\ rad$
- $e_2(0) = q_{ref2}(0) - q_2(0) = 2.0 - 1.5 = 0.5\ rad$

Similarly, the initial desired velocities from the reference trajectory need to be calculated by differentiating the reference trajectory functions and evaluating at t=0:

$$\dot{q}_{ref1}(t) = \frac{d}{dt}\left(1.25 - \frac{7}{5}e^{-t}\right) = \frac{7}{5}e^{-t}$$
$$\dot{q}_{ref1}(0) = \frac{7}{5}e^0 = 1.4\ rad/s$$
$$\dot{q}_{ref2}(t) = \frac{d}{dt}\left(1.25 + e^{-t} - \frac{1}{4}e^{-4t}\right) = -e^{-t} + e^{-4t}$$
$$\dot{q}_{ref2}(0) = -e^0 + e^0 = 0\ rad/s$$

The initial velocity errors are therefore:

- $\dot{e}_1(0) = \dot{q}_{ref1}(0) - \dot{q}_1(0) = 1.4 - 0 = 1.4\ rad/s$
- $\dot{e}_2(0) = \dot{q}_{ref2}(0) - \dot{q}_2(0) = 0 - 0 = 0\ rad/s$

These calculations confirm that there are significant initial position and velocity errors at the start of the simulation $(t = 0)$. The Computed Torque Control law, which includes feedback terms proportional to these errors $K_p e$ and $K_d \dot{e}$, generates large initial torque commands to quickly reduce these initial errors and bring the robot onto the desired trajectory. The large peak torques observed in the torque graph are because of the controller's effort to correct these substantial initial state differences.

5.1 Conclusion

The analysis of the joint position, velocity, and torque graphs demonstrates the effective performance of the Computed Torque Control strategy in tracking the desired trajectories for the two-joint robot. Despite significant initial position and velocity errors resulting from the mismatch between the robot's starting state and the reference trajectory's initial state, the controller successfully drives these errors towards zero. The large initial torque commands observed are a result of the control system actively working to correct these discrepancies and bring the robot to the desired path. The smooth convergence of position and velocity errors, coupled with the settling of torque commands, indicates that the empirically tuned PD gains within the CTC provide adequate feedback to stabilize the error and achieve accurate trajectory following.

References

[1] R. Academy, *Introduction to robot manipulator arms*, https://robotacademy.net.au/masterclass/robotic-arms-and-forward-kinematics/, 2023.

[2] N. U. M. Lab, *Modern robotics*, https://hades.mech.northwestern.edu/index.php/Modern_Robotics, 2023.

[3] R. Tedrake, *Underactuated robotics: Multibody dynamics*, https://underactuated.mit.edu/multibody.html, 2023.

[4] L. V. University of Nevada, *Computed torque control: Lecture notes for me729 - week 13*, https://www.daslhub.org/unlv/courses/me729-sp/week13/lecture/Note_09_Computed_Torque_Control.pdf, 2023.

[5] K. Lynch, *Modern robotics: Motion control with torque or force inputs*, https://www.youtube.com/watch?v=MV-xBPP3H2k, 2018.

[6] K. Lynch, *Empirical pid tuning*, YouTube Video, Available at: https://www.youtube.com/watch?v=uXnDwojRb1g.

[7] S. Boyd, *Lyapunov functions*, https://web.stanford.edu/class/ee363/lectures/lyap.pdf.