Computed Torque control of 3R articulated robot

* Modeling, Simulation, and Performance Analysis

Priyadharshan R
Department of Robotics and Automation
PSG College of Technology
Coimbatore, Tamil Nadu

Sivanesan A
Department of Robotics and Automation
PSG College of Technology
Coimbatore, Tamil Nadu

Dr. M.P Anbarasi
Department of Robotics and Automation
faculty of Engineering Technology
PSG College of Technology
Coimbatore, Tamil Nadu

Abstract*— *Robot modeling plays an important role in testing the functionalities of the robots in real-time applications. Imitation of a robot in the computer using a software application called MATLAB/SIMULINK that gives a precise understanding of the attributes of that manipulator such as the Kinematics and Dynamics of a robot. This will also helpful in controlling the motion of the manipulator by changing the control parameters of a manipulator. This paper is dedicated to the implementation of a robot model, where Computed Torque Control (CTC) is integrated to oversee the robot's motion by modulating the input parameter (torque). By the end, we have a clear idea about robot motion and the findings have valuable insights into the field of Robotics and Automation.

Keywords—3R articulated spatial robot, Kinematics and Dynamics of a robot, MATLAB/SIMULINK, Computed torque control, Non Linear Control.

I. INTRODUCTION

A robot helps humans to enhance efficiency, to facilitate and complete specific work within or before the allocated time. Different types of robots are used for different industrial applications. Any task cannot be completed in one go. It requires a sequence of actions or repetitive approach to reach the desired output for the designed robots should be practiced to do that task. The trial and error method is impractical for physical robots operating in real-world environments. Therefore, a better approach is chosen for creating an exact prototype of the robot in a software environment. To be specific, a 3R articulated robot (3 revolute joint robot) is generated within the MATLAB software using toolboxes called Simulink and Simscape [1]. Various control techniques are implemented to manage the behavior of robot and optimize its performance [4]. One of the control methods is regulating the manipulator via computed torque control [2]. It is a control technique used to control the motion of robot manipulators and a method that calculates the control inputs (torques) required to carry out desired joint movements . This involves creating a Simscape system that represents the mechanical components of the manipulator and deriving the dynamic equations of motion for the manipulator and thus the controller calculates the desired joint torques [3]. Implementation of the controller in Simulink using blocks involves feedback control, where the desired joint positions are compared with the actual joint positions, and control inputs are computed accordingly.

This paper is organized as follows:

In section II, Robot configuration is explained. In section III, Modeling of Robot in MATLAB/SIMULINK is discussed. Detail of classical CTC and MATLAB/SIMULINK implementation of this controller is presented in section IV. In section V, the simulation result is presented and finally in section VI, the conclusion is presented.

II. ROBOT CONFIGURATION

A. Parts of Robot

In the framework of robot arms or manipulators, the parts are joints, links, end effector and actuators. These are collectively used to build a manipulator. The definitions of the preceding segments are as follows.

Joints: These are rotating elements in the robot arm's structure that enables the robot to articulate and move. There are many types of joint which can be present in a manipulator; here we are using revolute joints to build a articulated robot.

Links: These are rigid elements that interconnect at the joints (basically that connect the joints). They determine the overall configuration of the manipulator. They are used to secure the manipulator's stability.

End effector: The end effector functions as the specialized tool or attachment which is at the of the robot arm. The common uses of the end effector are gripping, welding, lifting and pick and place objects.

Actuators: These are located at the joints to provide motive force for articulation and movement of robot's link and end effectors. The commonly used actuators are electric motors, hydraulic or pneumatic cylinders.

Serial Manipulator: In this arrangement, the robot's joints are sequentially connected, with each joint linked to the one preceding it. This configuration is frequently found in industrial robotic systems and provides precise control over the motion of each individual joint.

Parallel Manipulator: In the parallel manipulator configuration, multiple arms are linked in parallel to a shared base and end-effector. This design is frequently chosen for applications demanding increased rigidity, exceptional accuracy, and enhanced stability.

Articulated Robot: An articulated robot is a robotic system that has multiple revolute joints that allows rotational movement of the robot arm (the joint allows 360 degree rotation). They are employed in industries for manufacturing processes, industrial automation.

B. 3R Articulated spatial robot

A 3R articulated spatial robot is a type of robot characterized by its specific configuration of joints, enabling it to move and operate within a three-dimensional space. The "3R" in "3R Articulated Spatial Robot" stands for 3 joints and "R" typically stands for revolute.The term "articulated" in this context refers to the type of robot configuration, it has multiple segments or links connected by the joints. These joints enable the robot arm move in various directions.The term "spatial" describes that the robot can operate and manipulate objects in a 3D environment. In contrast to planar robots that can only operate in a two-dimensional plane, a spatial robot can manipulate things freely in all three axes—X, Y, and Z (3 dimensions).A 3R articulated spatial robot is used for tasks that require achieving flexible and precise movements in 3D space. Its ability to rotate its joints and move in various directions makes it suitable for applications such as grasping, manufacturing, pick-and-place operations, and tasks that demand in various orientations.

III. Robot Modeling

MATLAB stands for "Matrix laboratory" and it is an integrated development environment (IDE) where you do data analysis, modeling, numerical computation, simulation, data visualization, scripting and programming. It is a versatile tool for various modeling tasks in the field of science, research and engineering such as Mathematical modeling, Simulink modeling, Statistical modeling, Machine learning modeling, Image and signal processing, Control system modeling and so on.

MATLAB has an extension named Simulink that is specifically designed for analyzing, modeling, simulating dynamic systems in engineering and scientific fields. Simulink is especially used for modeling dynamic systems and analysis of the characteristics and behaviors of that system when different inputs are handled in that environment. Employ Simulink to create a dynamic system especially a 3R spatial manipulator. The design process utilizes various toolboxes which are available in the Simulink library browser to optimize the design process. The library browser includes a toolbox called 'Simscape' in which the 3R manipulator is constructed.

which will serve as the testbed for implementing the computer torque algorithm. The specific configuration of this robot is achieved by appropriately connecting joints using the Simscape Multibody toolbox. The choice of joints within Simscape Multibody is crucial in determining the desired configuration of the robot. The Simulink block diagram presented in Figure 1 serves as a visual representation of the 3R robot's configuration, which is an integral part of the simulation setup. In Figure 2, a detailed 3D model of the 3R robot is presented, offering a visual insight into its physical structure and design.

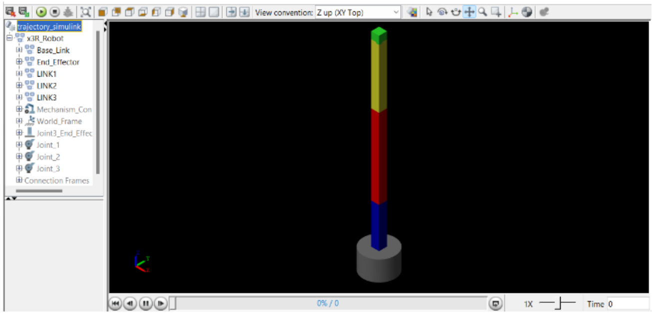

Figure 2 Detailed view of 3R Robot Model

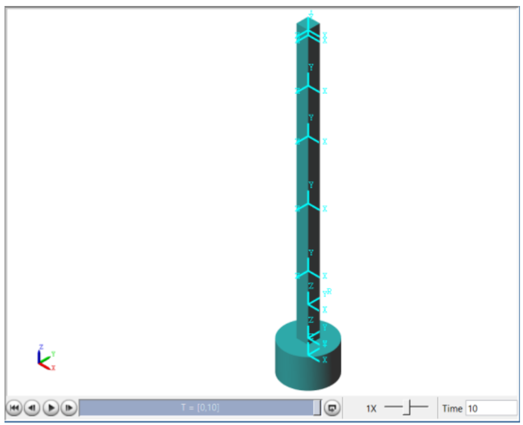

Figure 3 Detailed View of Robot Frames

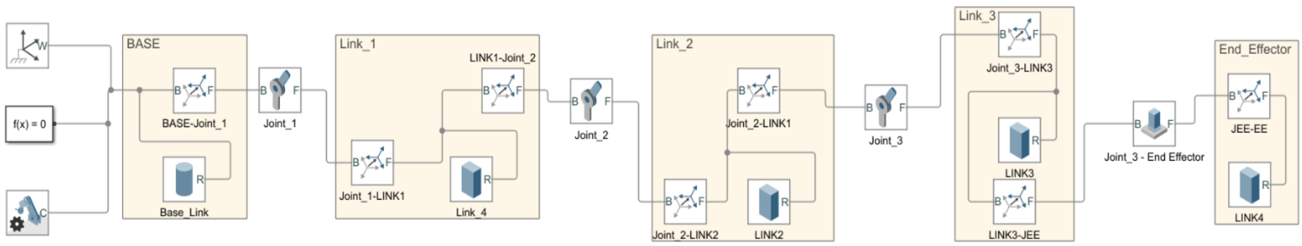

Figure 1 Simulink Block Diagram for Robot Configuration

A. Design process

To implement the computer torque control algorithm, a physical system is required for testing. In this context, a system has been designed using the Mechanical Explorer. In our case, the system under consideration is a 3R manipulator,

In Figure 3, detailed views of each frame in the 3D robot model are provided, enabling a thorough examination of the robot's individual components and connections. These views contribute to a deeper understanding of the robot's structural intricacies and facilitate the analysis of its behavior.

In Figure 4, the detailed block diagram illustrates the key components of joint control in the robotic system. Each joint seamlessly integrates torque input, position sensing, and velocity sensing. This configuration is crucial for the effective application of Computed Torque Control (CTC), ensuring precise and responsive control over the robot's motion.

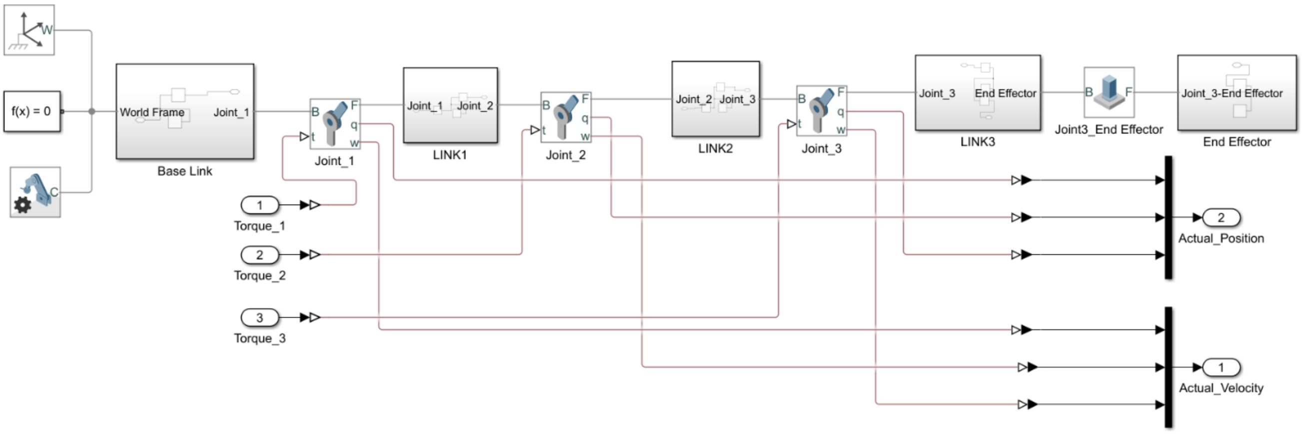

Figure 4 Joint Control Block diagram

In addition to that integration of torque input, position sensing, and velocity sensing as depicted in Figure 4, simulation utilizes the versatile *rigidBodyTree* framework available in Simulink. The *rigidBodyTree* is a fundamental component in robotic modeling that allows the representation of complex robotic systems with articulated structures. Its utility extends beyond our specific control implementation and finds applications in various robotics toolboxes, making it a valuable resource for modeling and simulating a wide range of robotic systems. This framework facilitates the creation of detailed and dynamic robot models, enabling comprehensive analyses and simulations in fields such as kinematics, dynamics, and control.

IV. Computed Torque control

Computed Torque Control (CTC), a form of inverse dynamic control, stands as a transformative paradigm in the domain of robotic motion control, specifically excelling in addressing the intricate dynamics governing 3R robotic manipulators. CTC, often employed in scenarios where reference inputs vary over time, transcends the conventional set point control problem and delves into the more generalized realm of servo control. Within this framework, it assumes that planned trajectories (r(t)) exhibit smoothness with at least two derivatives, ensuring that the control problem is well-defined. Introduced by Bejczy in 1974, the computed-torque method offers a groundbreaking solution to the robot servo problem. This approach stands out by making direct and comprehensive use of the dynamic model of the manipulator. Its primary objective extends beyond reducing the influence of gravity alone rather, it seeks to counteract the combined effects of gravity, Coriolis and centrifugal forces, friction and even the manipulator's inertia tensor.

In practical applications of computed-torque control for 3R robotic systems, estimations of crucial robotic arm parameters become paramount. Instead of relying on exact values, these implementations hinge on estimates of parameters such as the manipulator's inertia tensor (D), the Coriolis and centrifugal force vector (H), and the gravity loading vector (G). This reliance on estimated values, denoted as $\{D, H, G\}$, serves as a pragmatic approach to tackle the complexities inherent in real-world scenarios. The estimates are refined and adjusted in real-time to ensure that CTC operates effectively in dynamically changing environments.This detailed combination of CTC's comprehensive approach to dynamic modeling and its adaptability through parameter estimation underscores its potential for enhancing the precision and adaptability of 3R robotic manipulators in practical applications, marking a significant advancement in the field of robotic motion control. It often employs Proportional-Derivative (PD) controllers, specifically tuned with proportional (k_P) and derivative (k_D) gain matrices. These controllers are pivotal components in achieving the desired precision and performance in motion control.

The incorporation of the PD controllers in the CTC framework further augments its capabilities. The proportional gain (k_P) matrix plays a crucial role in determining the control effort proportional to the error between the desired trajectory and the actual robot positions. It essentially sets the natural frequency of the system, influencing the speed at which the robot responds to deviations from the desired trajectory. On the other hand, the derivative gain (k_P) matrix provides damping to the system, minimizing overshoot and oscillations. The values of k_P and k_D are precisely fine-tuned to suit the specific characteristics of the robotic manipulator and the requirements of the task.

In practical applications of CTC for 3R robots, this combination of control-law partitioning, dynamic modeling, and PD controllers results in precise trajectory tracking and robust disturbance rejection. The computed torque, obtained through dynamic modeling and control-law partitioning, is essentially the control effort required to compensate for nonlinearities and external disturbances. The PD controllers ensure that this control effort is applied effectively, allowing the robot to follow the desired trajectory with minimal error and settling quickly to the desired positions.

This integrated approach not only enhances precision but also significantly reduces the impact of modeling errors and disturbances in real-world scenarios. By amalgamating the strengths of dynamic modeling, control-law partitioning, and PD control, CTC for 3R robotic manipulators stands as a cutting-edge solution that holds immense potential for advancing the field of robotic motion control, particularly in situations where precision and adaptability are paramount.

Feedback Linearization and Feedforward-Feedback Loop

Within the framework of Computed Torque Control (CTC) for 3R robotic manipulators, an intriguing application emerges in the realm of feedback linearization [4]. This technique assumes significance in contexts where the accurate tracking of reference inputs takes precedence. It operates through a two-pronged approach: the feedforward loop, aimed at eradicating the inherent nonlinearities in the system dynamics, and the feedback loop, tasked with tracking the reference input.

A. Derivation of Inner Feedforward Loop

The essence of CTC revolves around a deep understanding of the robot's dynamics. Consider the dynamic equation:

$$Q = D(q)\ddot{q} + H(q,\dot{q}) + G(q) \tag{1}$$

Where:

q is the vector of joint variables.

$Q(q,\dot{q},t)$ represents the torques applied at the robot's joints.

The tracking error (e) represents the difference between the desired joint positions (q_d) and the actual joint positions (q) of the robot at any given time. It quantifies how closely the robot's behavior aligns with the desired trajectory in joint space.

e is the error vector defined as $e = q - q_d$ (2)

To control the robot to follow a desired path in joint space, a computed torque control law is introduced:

$$Q = D(q)(\ddot{q}_d - k_D\dot{e} - k_P e) + H(q,\dot{q}) + G(q) \tag{3}$$

Where:

k_P and k_D are constant gain diagonal matrices.

The stability of this control law depends on the eigenvalues of the matrix

$$[\mathrm{A}] = \begin{bmatrix} 0 & I \\ -k_P & -k_D \end{bmatrix} \tag{4}$$

Stability Analysis

To analyze the stability of the control system, we start by calculating the control torques

$$Q_c = D(q_d)\ddot{q_d} + H(q_d,\dot{q_d}) + G(q_d) \tag{5}$$

required to track the desired trajectory:

Next, apply the computed torque control law (Equation 5) to the robot and obtain the error dynamics equation:

$$\ddot{e} + k_D\dot{e} + k_P e = 0 \tag{6}$$

This is a linear differential equation for the error variable e between the actual and desired joint positions. In matrix form, it can be written as:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -k_P & -k_D \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = [A] \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{7}$$

The stability of the control system is determined by the eigenvalues of matrix [A]. For the control system to be asymptotically stable, all eigenvalues of [A] must have negative real parts.

B. Gain Matrices

The gain matrices $k_P \; and \; k_D$ are diagonal matrices, and their values can be adjusted to control the response speed of each joint independently. A simple choice for these matrices is to set ξ_i=0 for all joints, resulting in each joint behaving as a critically damped linear second-order system with a natural frequency ω_i

$$k_D = \begin{bmatrix} 2\xi_1\omega_1 & 0 & 0 & 0 \\ 0 & 2\xi_2\omega_2 & 0 & 0 \\ 0 & 0 & \dots. & 0 \\ 0 & 0 & 0 & 2\xi_n\omega_n \end{bmatrix} \tag{8}$$ and

$$k_P = \begin{bmatrix} \omega_1^2 & 0 & 0 & 0 \\ 0 & \omega_2^2 & 0 & 0 \\ 0 & 0 & \dots. & 0 \\ 0 & 0 & 0 & \omega_n^2 \end{bmatrix} \tag{9}$$

Since $k_P \; and \; k_D$ are diagonal, we have the flexibility to adjust these gain matrices independently for each joint. A straightforward choice for these matrices is to set ξ_i=0 for i=1,2,…,n. This configuration ensures that each joint responds as a critically damped linear second-order system with its own natural frequency ω_i

$$Q = D(q)(\ddot{q}_d) + H(q,\dot{q}) + G(q) + D(q)(-k_D\dot{e} - k_P e) \tag{10}$$

Where :

$$Q_{ff} = D(q)(\ddot{q}_d) + H(q,\dot{q}) + G(q) \tag{11}$$

$$Q_{fb} = D(q)(-k_D\dot{e} - k_P e) \tag{12}$$

Feedforward Component (Q_{ff}): This term represents the required torques based on the open-loop control law. When there is no tracking error, the control input Q_{ff} ensures that the robot accurately follows the desired path q_d.

Feedback Component (Q_{fb}): The feedback component introduces correction torques to mitigate errors in the robot's path. This feedback loop is essential for reducing discrepancies between the actual and desired joint positions.

Computed torque control is often referred to as feedback linearization, a valuable technique for designing nonlinear control systems for robots. This approach aims to eliminate all nonlinearities, transforming the control problem into a linear second-order equation governing the error signal:

$$\ddot{e} + k_D\dot{e} + k_P e = 0 \tag{13}$$

In this equation, the gain matrices k_P and k_D control the response speed and damping of each joint independently. This enables precise motion control and trajectory tracking, compensating for the inherent nonlinearities and disturbances in the robot system.

By integrating this feedback linearization technique and the PD controller into CTC for 3R robotic manipulators, we establish a potent framework capable of eliminating nonlinear terms and achieving precise trajectory tracking. This integrated approach enhances the precision and adaptability of robotic motion control systems, making them exceptionally well-suited for real-world applications where dynamic and precise performance is a prerequisite.

C. Implemented Computed Torque Controller

In the initial phase of implementation, construction of the dynamics and kinematics blocks, essentially forming the plant of the system, and integrated them with the necessary power supply components within the robot's workspace. The primary objective of this endeavour is as to introduce a controller block capable of effectively regulating the robot's behavior. The chosen control strategy adheres to the following approach the Proportional-Derivative (PD) equation, which forms the basis for the controller design. The PD controller block, as illustrated in Figure 5, represents the linearized portion of the control system, often referred to as the PID controller. In this context, "e" represents the error, defined as the disparity between the actual and desired values, while "$\dot{e}$" denotes the rate of change of this error.

The PD controller incorporates two essential gain parameters, "k_P" and "k_D." "k_P" stands for the proportional gain, which determines the strength of the proportional response to the error. Meanwhile, "k_D" represents the derivative gain, influencing the control action based on the rate of change of the error. Additionally, the "d" term signifies the double derivative of the joint variable, further contributing to the controller's overall behavior.

To illustrate, here is a sample PD controller *Block diagram*

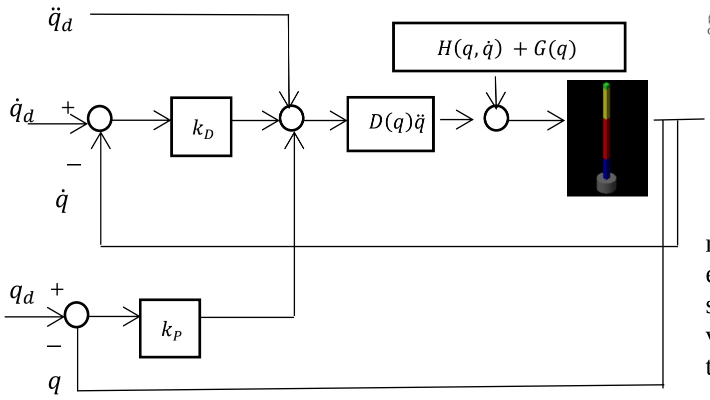

Figure 5 PD Controller Block Diagram

The integration of the PD controller into the overall control system allows us to effectively regulate the robot's joint movements and improve its response to desired trajectories. This approach is a fundamental component of the Computed Torque Control (CTC) methodology, enables accurate and responsive robotic control. In the control framework, Utilizing a trapezoidal velocity profile, the power of this approach becomes evident to closely design the desired trajectories for robot. By precisely defining waypoints, generation of smooth and continuous sequences of joint positions (q), velocities (q_d), and accelerations(q_{dd}). These trajectories serve as fundamental reference signals and integrated into control architecture, which includes the Computed Torque Controller (CTC) and a PD controller. This approach not only minimizes abrupt changes in motion (jerk) but also optimizes control, facilitating the robot's precision in tracking predefined paths. This adaptive methodology proves invaluable across a spectrum of applications demanding high-fidelity and efficient robotic movement.

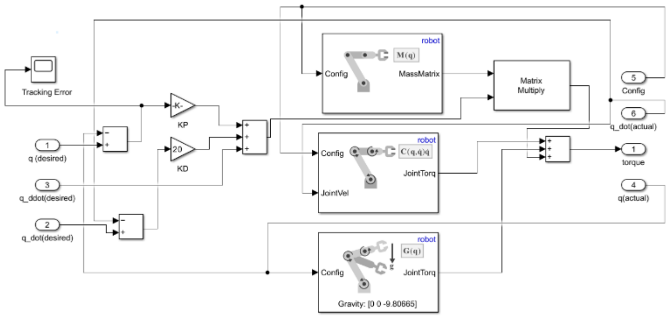

Figure 6 CTC Block Diagram in Simulink

Within the Simulink-based framework, the rigidBodyTree plays a pivotal role in the automatic computation of key dynamic components. The mass matrix block efficiently derives the system's mass distribution by considering the inherent properties of each link and joint, resulting in a comprehensive representation of the robot's inertia. The gravity block computes the gravitational forces exerted on the robot's links based on the system's configuration and the gravitational constant, enabling precise gravity compensation. Meanwhile, the Coriolis effects block calculates the Coriolis and centrifugal terms, accounting for the robot's motion and velocity.

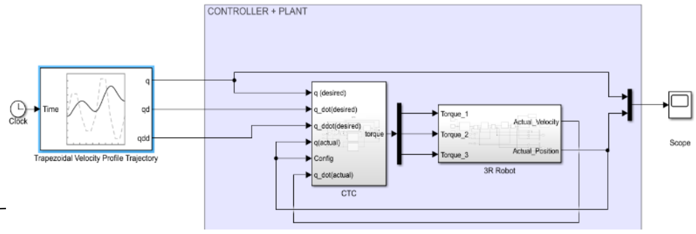

Figure 7 Robot Trajectory Planning and Control Architecture

These Simulink blocks, seamlessly integrated with the rigidBodyTree, facilitate dynamic modeling and control, ensuring the accuracy and effectiveness of robotic simulations. Notably, Figure 6 provides an encompassing view of control architecture, showcasing the integration of these dynamic components within the Simulink environment.

Figure 7 shows stands as a visual representation of the interplay within control system. This block diagram offers a comprehensive view of how trajectory planning, control elements, and real-time feedback combine within our robotic system. It discloses the seamless interaction between the precisely designed trajectories and the robot's swift execution. Figure 7 provides a complete perspective of the trajectory input controller and its amalgamation with the robotic system, underlining the effectiveness of approach in guiding the robot along predefined paths.

V. RESULTS

This section reports the outcomes of applying the Computed Torque Controller (CTC) to a robotic manipulator. In this section, To optimize controller performance, The specific gains: a proportional gain (k_P) of **250** and a derivative gain (k_D) of **20** were chosen. These gain values were selected to strike a balance between trajectory tracking precision and disturbance rejection capability. To ensure precise trajectory tracking, the waypoints are strategically defined for the robot's motion, specifically **[0.4, 1, 0.9; -0.2, 1.20, 0.8; 0.6, 1.4, 0.7]**, which facilitated the generation of smooth and continuous joint trajectories.

In the Simulink, simulation of the 3R articulator robot with computed torque control, a detailed assessment is conducted to evaluate the performance of the control algorithm. The comparison between actual joint positions and their desired positions revealed the significance of the root mean square (RMS) error for each joint as a critical metric.

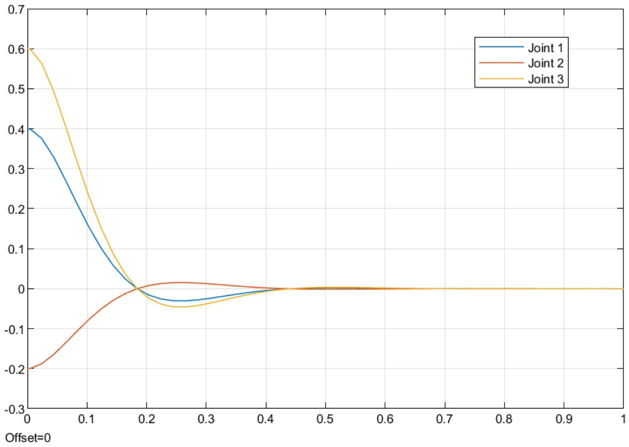

Figure 8 Tracking error of 3R Robot

Figure 8 illustrates the error analysis for our control system. It demonstrates how well the CTC minimizes the discrepancies between desired and actual joint positions. The error remains consistently low throughout the trajectory, highlighting the effectiveness of our control approach in achieving accurate position tracking.

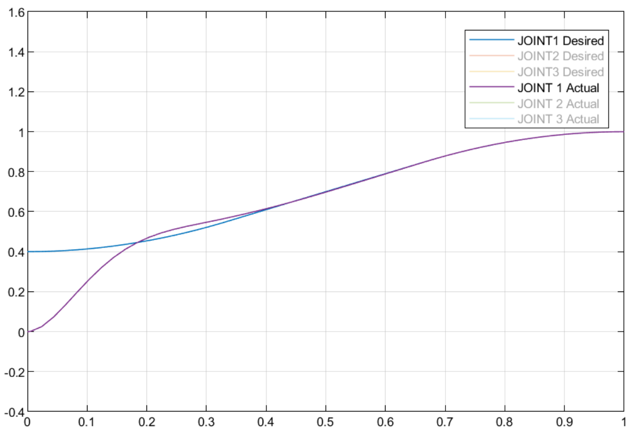

Figure 9: Joint 1 - Desired vs. Actual positions

. Figure 9 illustrates the joint 1 desired vs. actual trajectory, while Table 1 provides detailed values for this trajectory. The RMS error between the desired and actual trajectory for Joint 1 is calculated to be **0.163702** units, highlighting the precision achieved in tracking this trajectory.

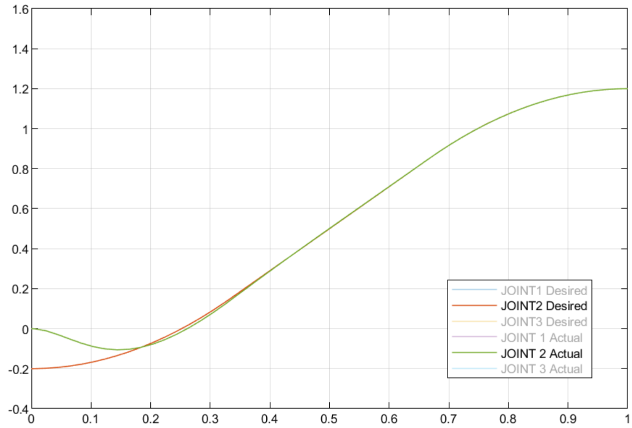

Figure 10: Joint 2 – Desired vs Actual positions

Figure 10 extends the analysis to the second joint of the robotic manipulator. It presents the alignment between the desired and actual positions, indicating the capability of our CTC in maintaining accuracy across multiple joints. The RMS error for Joint 2 is calculated to be **0.081856** units, reaffirming the CTC's capability to ensure accurate tracking.

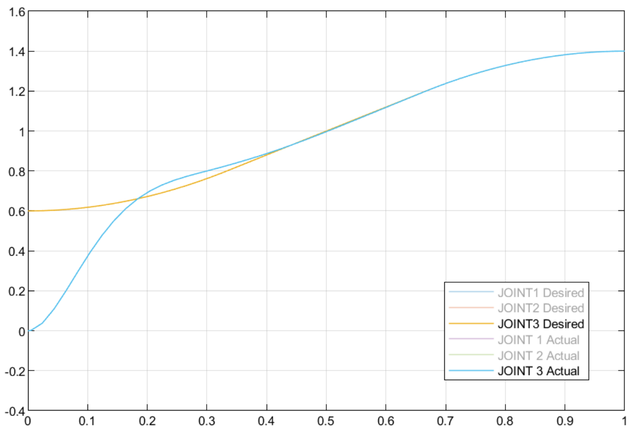

Figure 11: Joint 2 - Desired vs. Actual

Figure 11 zooms in on the third joint, highlighting its performance in tracking the desired trajectory. The minimal deviation between the desired and actual positions showcases the robustness of our control framework. Table 3 provides numerical data. The RMS error for Joint 3 is calculated to be **0.245559** units, emphasizing the controller's efficiency in maintaining trajectory fidelity.

TABLE I. TIME VS. JOINT 1 - DESIRED VS. ACTUAL

S.NO	Time in s	Joint 1 (Desired) Position in m	Joint 1 (Actual) Position in m
1.	0	0.4000	0.0000
2.	0.2	0.4000	0.0013
3.	0.4	0.4280	0.3703
4.	0.6	0.5089	0.5373
5.	0.8	0.6316	0.6332
6.	1.0	0.7576	0.7558
7.	1.2	0.8817	0.8817
8.	1.4	0.9672	0.9672
9.	1.6	0.9997	0.9997
10.	1.8	1.0000	1.0000

Time-series data for joint 1 positions (desired vs. actual).

TABLE II. TIME VS. JOINT 2 - DESIRED VS. ACTUAL

S.No	Time t in s	Joint 2 (Desired) Position in m	Joint 2 (Actual) Position in m
1.	0	-0.2000	0.0000
2.	0.2	-0.1999	-0.0005
3.	0.4	-0.1347	-0.1058
4.	0.6	0.0541	0.0399
5.	0.8	0.3405	0.3400
6.	1.0	0.6345	0.6354
7.	1.2	0.9241	0.9239
8.	1.4	1.1234	1.1233
9.	1.6	1.1992	1.1992
10.	1.8	1.2000	1.2000

Time-series data for joint 2 positions (desired vs. actual)

TABLE III. TIME VS. JOINT 3 - DESIRED VS. ACTUAL

S.No	Time in s	Joint 3 (Desired) Position in m	Joint 3 (Actual) Position in m
1.	0	0.6000	0.0000
2.	0.2	0.6000	0.0019
3.	0.4	0.6373	0.5508
4.	0.6	0.7452	0.7877
5.	0.8	0.9088	0.9112
6.	1.0	1.0768	1.0741
7.	1.2	1.2423	1.2423
8.	1.4	1.3562	1.3563
9.	1.6	1.3995	1.3995
10.	1.8	1.4000	1.4000

Time-series data for joint 3 positions (desired vs. actual).

VI. CONCLUSION

In this study, the fundamental aspects of a 3R robot manipulator and its associated control methodologies, particularly the computed torque controller (CTC), have been thoroughly examined. The 3R robot's inherent nonlinearity within its dynamic parameters poses a significant challenge, which CTC, based on feedback linearization, adeptly addresses by providing precise control for both deterministic and partially uncertain systems through the application of a nonlinear feedback control law. The fine-tuning of specific k_P and k_D values for each joint has been demonstrated to directly impact the robot's responsiveness, speed, and stability, as evidenced by the comparison between actual and desired joint positions, with the root mean square (RMS) error serving as a critical performance metric. It's crucial to emphasize that CTC's efficacy is contingent on the availability of accurate dynamic and physical parameters, highlighting the need for continued research in parameter estimation techniques to further enhance the control system's precision and robustness, ultimately contributing valuable insights to the field of Robotics and Automation.

REFERENCES

[1] Farzin Piltan , Mohammad Hossein Yarmahmoudi, Mohammad Shamsodini, Ebrahim Mazlomian & Ali Hosainpour, "PUMA-560 Robot Manipulator Position Computed Torque Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control and MATLAB Courses", International Journal of Robotics and Automation, (IJRA), Volume (3) : Issue (3) : 2012

[2] FUNDAMENTALS OF ROBOTICS: Analysis and Control by Robert J. Schilling.

[3] S K Saha Professor Department of Mechanical Engineering Indian Institute of Technology (IIT) Delhi New Delhi, "Introduction to Robotics"

[4] Reza N. Jazar Theory of Applied Robotics Kinematics, Dynamics, and Control

[5] "INTRODUCTION TO ROBOTICS ANALYSIS, CONTROL, APPLICATIONS" Saeed Benjamin Niku, Ph.D., P.E. Professor Mechanical Engineering Department California Polytechnic State University San Luis Obispo.

[6] "Optimal Control Theory An Introduction", Donald E. Kirk Professor Emeritus of Electrical Engineering San José State University San José, California.

[7] Kinematic Modeling and Simulation of a SCARA Robot by Using Solid Dynamics and Verification by MATLAB/Simulink.

[8]] Farzin Piltan, N. Sulaiman, M. H. Marhaban, Adel Nowzary, Mostafa Tohidian," "Design of FPGA based sliding mode controller for robot manipulator," International Journal of Robotic and Automation, 2 (3): 183-204, 2011.

[9] I. Eksin, M. Guzelkaya and S. Tokat, "Sliding surface slope adjustment in fuzzy sliding mode controller," Mediterranean Conference, 2002, pp. 160-168.

[10] Samira Soltani & Farzin Piltan, "Design Artificial Nonlinear Controller Based on Computed Torque like Controller with Tunable Gain". World Applied Science Journal,14 (9): 1306- 1312, 2011.

[11] Farzin Piltan, H. Rezaie, B. Boroomand, Arman Jahed," Design robust back stepping online tuning feedback linearization control applied to IC engine," International Journal of Advance Science and Technology, 42: 183-204, 2012.