

Performance estimation of computed torque control for surgical robot application

V.Verma¹, A. Gupta², M. K. Gupta¹ and P. Chauhan³

¹ Department of Electrical and Electronics Engineering, University of Petroleum and Energy Studies, Dehradun, India-248007

Phone: +91-8006496996

² Department of Mechanical Engineering, University of Petroleum and Energy Studies, Dehradun, India-248007

³ Department of Analytics, University of Petroleum and Energy Studies, Dehradun, India-248007

ABSTRACT – In the current paradigm, the development in robotic technology has a huge impact to revolutionize the medical domain. Surgical robots have greater advantages over surgeon such as reduced operating time, reduced tremor, less blood loss, and high dexterity. To perform different operations during surgery a base robot is required with the task-specific end effector. In this paper, the selective compliant assembly robot arm (SCARA) has been considered as the base robot and the complete mathematical modeling of the robot is illustrated. The equation of Kinematics is derived from the D-H notation. SCARA dynamic model is derived from Euler Lagrange. In order to achieve trajectory tracking the Computed Torque Control technique (CTC) applied to the SCARA manipulator. The performance of the CTC technique for trajectory tracking of each joint of the SCARA robot has evaluated in contrast with tuned PD and PID controller. The simulation results were discussed and verified using MATLAB simulation software.

ARTICLE HISTORY

Revised: 22nd Nov 2019

Accepted: 08th May 2020

KEYWORDS

*Mathematical modeling;
Computed torque control;
Newton Euler;
SCARA robot;
Trajectory tracking*

INTRODUCTION

In recent decades, robotic researches have become great attention and has N number of applications in every field of research such as industrial, medical, agriculture and so on and so forth[1]. In industrial applications, robots are used for packaging, segregating, assembling, etc. In Agriculture used for picking, collecting and segregating fruits, vegetables, flowers, etc. In medical research used as assistive robots, surgical robots, etc.[2] For every application, the requirement of robot configuration differs. Therefore, Robotic manipulators were developed with various types of joints such as rectangular, cylindrical, spherical, revolute and horizontal joints to perform different tasks. The task comprises some trajectory or path and the robot is expected to maneuver or follow that required trajectory [3–5]. Different control schemes are used to follow that required trajectory because there exists the nonlinearity, uncertainty, external disturbances, strong coupling and time varied of the robotic system. There are Linear and nonlinear control schemes such as PD (Proportional-Derivative) Control, PID (Proportional Integration Differentiation) Control, CTC (Computed Torque Control), Adaptive control Fuzzy Control and so on [6–10]. This paper focuses on CTC, which is nonlinear feedback control.

The most commonly used robot in industrial and medical applications is SCARA (Selective Compliance Assembly Robot Arm) although the first robot was invented by Japan more than half-decade still it is an indispensable element in the automation industries. Speed, reliability, small workspace and cost-effectiveness make this robot widely used all over the world. The Selective Compliance Assembly Robot Arm (SCARA robot) can be seen in Figure 1 manufactured by Rexroth is considered an object of interest, which is a four-axis horizontal joined articulated arm configuration. The first two joints of the robot are revolute to establish the horizontal position of the robot. The third joint is the prismatic joint which defines the vertical position of the end tool. Finally, the last joint will provide the tool orientation. Therefore it has RRRP Configuration with the cylindrical workspace [11].

Building up a new robot model, analyzing and verification of the model should be taken care of before constructing up in the real world. The simulation is more popular because of the low cost of the computer, which is helpful in analyzing feasibility studies, the presentation with animation, layout evaluation, and offline programming. Therefore, the simulation tool like MATLAB, ADAMS, Vrep, and ROS and so on can be used to simulate robot to analysis robot motion [12]. During the surgical simulation for preoperative planning of maneuvering the robot to operate the patient, the joint control of robot for trajectory tracking is required. In order to achieve more accurate, precise trajectory tracking control, the classical PD and PID controller are compared with PD CTC control to calibrate the performances of joint control for the proposed system. The purpose of this paper is to illustrate, analyze and simulate the kinematic, dynamic analysis and control of the robot using the MATLAB simulation software [13]. The four cases for the PD-CTC controller are taken into consideration with different gain values the best from which is compared with the tuned PD and PID control in order to achieve the desired joint trajectory. The CTC control technique used to produce tracking control with minimum error.

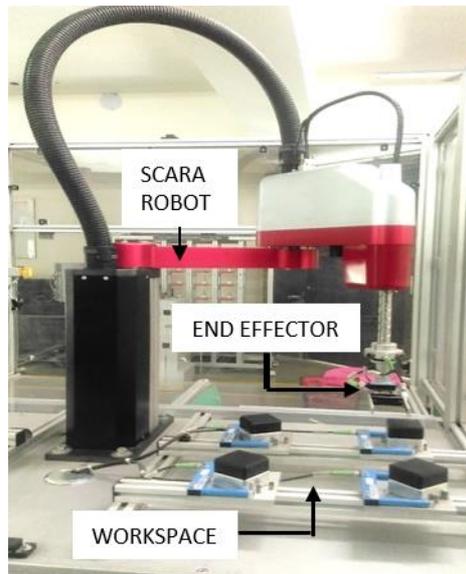


Figure 1. Selective Compliance assembly robot arm (SCARA robot)

KINEMATIC MODELING OF SCARA

The SCARA Manipulators are widely used in various assembly applications like Cutting, Selecting, Segregating, Pick and Place, etc. It has a horizontal jointed articulated arm configuration manipulator called Selective Compliance Assembly Robot Arm (SCARA). The end tool of the robot can be modified to perform various surgical tasks such as a holding camera and smart tools like endostitch, endosew, etc [14]. The frame assignment of the SCARA manipulator can be seen in Figure 2.

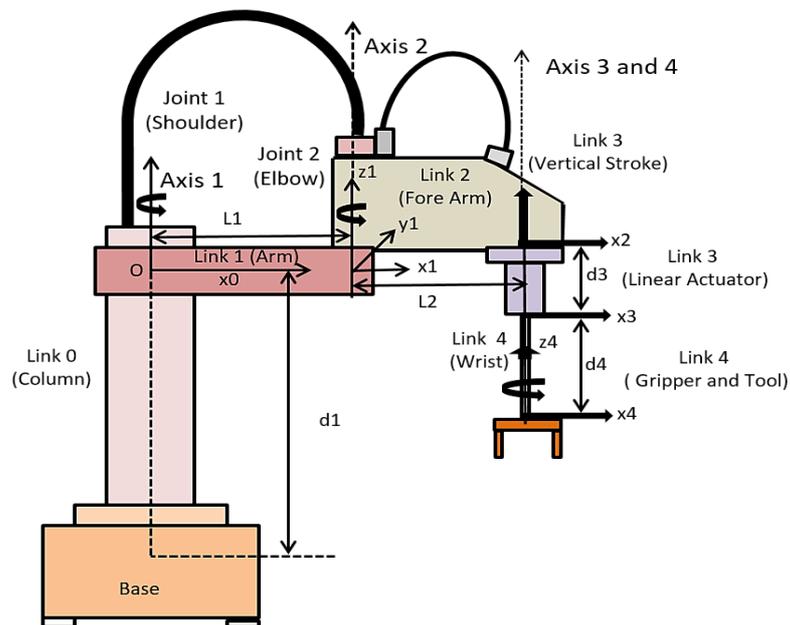


Figure 2. Mechanical Structure of 4-DOF SCARA (RRPR) manipulator arm

To identify the dynamics and kinematics of the manipulator the mass, link length and moment of inertia parameter is required. Therefore, Table 1 illustrates the parameters taken into consideration during the modeling of the manipulator in simulation.

Table 1. SCARA robot parameters

Parameters	Value
Mass of Link 1 (m_1)	3.1 kg
Mass of Link 2 (m_2)	0.85 kg
Mass of Link 3 (m_3)	0.56 kg
Mass of Link 4 (m_4)	0.24 kg
Link1 Length (l_1)	0.5 meter
Link1 Length (l_2)	0.2 meter
Link 1 Offset (d_1)	1.9 meter
Link 3 Offset (d_3)	0.24 meter
Link 4 Offset (d_4)	1.3 meter
Moment of Inertia for link 1 (I_1)	0.5728 kg.m ²
Moment of Inertia for link 2 (I_2)	0.2130 kg.m ²
Moment of Inertia for link 3 (I_3)	1.64 kg.m ²
Moment of Inertia for link 4 (I_4)	0.567 kg.m ²
Acceleration Due To Gravity (g)	9.8 m/sec ²

The Denavit-Hartenberg (D-H) parameter of the SCARA robot are defined in Table 2.

Table 2. D-H Parameter of the SCARA Manipulator

Axis	θ_i	d_i	a_i	α_i
1	θ_1	d_1	l_1	Π
2	θ_2	0	l_2	0
3	0	d_3	0	0
4	θ_4	d_4	0	0

The A_i represents the generalized Transformation Matrices of link i can be seen from Eq. (1) to Eq. (3).

$$A_i = Rot_{z,\alpha} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\theta_i} \tag{1}$$

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The joint parameters as given in the D-H table and the transformation matrix defined as A-matrix for each joint has defined as follows from Eq. (4) to Eq. (7).

$$A_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_1 \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_1 \sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$A_2 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 & l_2\cos\theta_2 \\ \sin\theta_2 & -\cos\theta_2 & 0 & l_2\sin\theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$A_4 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The Forward Kinematic Equation of the manipulator can be written as stated in Eq.(8).

$$T_0^4 = A_1 \dots A_4 = \begin{bmatrix} \cos(\theta_1 - \theta_2 - \theta_4) & \sin(\theta_1 - \theta_2 - \theta_4) & 0 & l_1\cos(\theta_1) + l_2\cos(\theta_1 - \theta_2) \\ \sin(\theta_1 - \theta_2 - \theta_4) & -\cos(\theta_1 - \theta_2 - \theta_4) & 0 & l_1\sin(\theta_1) + l_2\sin(\theta_1 - \theta_2) \\ 0 & 0 & -1 & d_1 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

On comparing the above Eq. (8) with Eq. (9):

$$T_0^4 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where, the arbitrary parameters used in Eq. (9) were illustrated in below from Eq. (10) to Eq. (17).

$$r_{11} = \cos(\theta_1 - \theta_2 - \theta_4) \tag{10}$$

$$r_{12} = r_{21} = \sin(\theta_1 - \theta_2 - \theta_4) \tag{11}$$

$$r_{22} = -\cos(\theta_1 - \theta_2 - \theta_4) \tag{12}$$

$$r_{13} = r_{31} = r_{32} = r_{23} = 0 \tag{13}$$

$$r_{33} = -1 \tag{14}$$

$$d_x = l_1\cos(\theta_1) + l_2\cos(\theta_1 - \theta_2) \tag{15}$$

$$d_y = l_1\sin(\theta_1) + l_2\sin(\theta_1 - \theta_2) \tag{16}$$

$$d_z = d_1 - d_3 - d_4 \tag{17}$$

where, d_x, d_y, d_z represents the end-effector position and can be called as d matrix with a 3x1 matrix dimension and similarly R as a rotation matrix with a 3x3 dimension.

The Inverse Kinematics given by the set of solution of the equations as follows:

$$\begin{bmatrix} \cos(\theta_1 - \theta_2 - \theta_4) & \sin(\theta_1 - \theta_2 - \theta_4) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 - \theta_2) \\ \sin(\theta_1 - \theta_2 - \theta_4) & -\cos(\theta_1 - \theta_2 - \theta_4) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 - \theta_2) \\ 0 & 0 & -1 & d_1 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \quad (18)$$

Since four DOF of SCARA matrix will not have a definite solution unless R is in the form as stated in Eq. (19).

$$\begin{bmatrix} \cos\beta & \sin\beta & 0 \\ \sin\beta & -\cos\beta & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (19)$$

In this case, $\theta_1 - \theta_2 - \theta_4$ can be determined by Eq.(20) and respective Eq.(21) defined θ_2 .

$$\theta_1 - \theta_2 - \theta_4 = \beta = A \tan(r_{12}, r_{11}) \quad (20)$$

$$\theta_2 = A \tan(\pm\sqrt{1-r^2}, r) \quad (21)$$

where, the arbitrary constant r^2 is stated n Eq. (22) and Eq. (23) illustrate the computation of θ_1 .

$$r^2 = \frac{d_x^2 + d_y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (22)$$

$$\theta_1 = A \tan(d_x, d_y) - A \tan(l_1 + l_2 C_2, l_2 S_2) \quad (23)$$

The tool roll angle θ_4 from Eq. (20) can be defined as stated in Eq. (24) and Eq. (25).

$$\theta_4 = \theta_1 + \theta_2 - \beta \quad (24)$$

$$\theta_4 = \theta_1 + \theta_2 - A \tan(r_{12}, r_{11}) \quad (25)$$

The prismatic joint variable d_3 is associated with a sliding tool up and down with a tool roll axis. The vertical component of tool motion is uncoupled from horizontal components. Finally, the prismatic joint d_3 has given as stated in Eq. (26).

$$d_3 = d_z + d_4 \quad (26)$$

SCARA ROBOT MANIPULATOR DYNAMIC FORMULATION

In this section, the dynamic model of the SCARA manipulator is discussed. The Newton Euler and Euler Lagrange are the two common methods used for finding the dynamic equation. The dynamic equation of the SCARA robot manipulator is derived from the Newton Euler Method. The generalized dynamic expression for N degree of freedom robot manipulator can be expressed as below in Eq. (27).

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) = \tau \quad (27)$$

where generalized force vector (n x 1 dimension) is expressed as τ , M is Inertia Matrix with the dimension of n x n. $M(q)$ is a positive symmetric matrix, N is Centrifugal and Coriolis Forces (n x 1 dimension), G is a Gravitational Force Vector, \dot{q} is Joint Angular Velocity Vector and q is Joint Position Vector.

The Inertia Matrix of the SCARA 4 DOF is as stated in Eq. (28).

$$M(q) = \begin{bmatrix} p_1 + p_2 \cos(\theta_2) & p_3 + 0.5 p_2 \cos(\theta_2) & 0 & -p_5 \\ p_3 + 0.5 p_2 \cos(\theta_2) & p_3 & 0 & -p_5 \\ 0 & 0 & p_4 & 0 \\ -p_5 & -p_5 & 0 & -p_5 \end{bmatrix} \tag{28}$$

and Coriolis ($N(q)$) Matrix is calculated as the following Eq. (29).

$$N(q) = \begin{bmatrix} -p_2 \sin(\theta_2) \dot{\theta}_2 & -0.5 p_2 \sin(\theta_2) \dot{\theta}_2 & 0 & 0 \\ 0.5 p_2 \sin(\theta_2) \dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{29}$$

Gravity Matrix (G) can be written as Eq. (30).

$$G(q) = \begin{bmatrix} 0 \\ 0 \\ -p_4 g \\ 0 \end{bmatrix} \tag{30}$$

The SCARA parameters are as follows mentioned in Eq. (31) to Eq. (37).

$$p_1 = l_1 + l_2 + l_3 + I_4 + m_1 l_1^2 + K1 + K2 \tag{31}$$

$$p_2 = 2(2l_1 l_2 m_2 + 4l_1 l_3 (m_3 + m_4)) \tag{32}$$

$$p_3 = I_2 + I_3 + I_4 + m_2 l_2^2 + 4l_2^2 (m_3 + m_4) \tag{33}$$

$$p_4 = m_3 + m_4 \tag{34}$$

$$p_5 = I_4 \tag{35}$$

$$K1 = m_2 (l_2^2 + 4l_1^2) \tag{36}$$

$$K2 = (m_3 + m_4) (4l_1^2 + 4l_2^2) \tag{37}$$

where I_i is the moment of Inertia around the centroid, m_i is the mass, l_i is the length of link i . The SCARA manipulator Jacobian, With respect to the robot base frame, is as mentioned in Eq. (38).

$$J(q) = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \cos(\theta_1 + \theta_2) & 0 & 0 \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{38}$$

CONTROL: COMPUTED TORQUE CONTROLLER ANALYSIS

The most widely used nonlinear and powerful controller used for almost all schemes for robot control in robot manipulators is Computed Torque Control (CTC) [13]. It has a special application on the different nonlinear systems based on feedback linearization by the use of nonlinear feedback law it would be able to compute required torque in the arm [15-18]. It performs significantly well when all dynamic and physical parameter of the system is known [19-23]. The CTC like control appears in Robust, Adaptive and Learning Control [19,23,24]. Figure 3 provides the block diagram for PD-CTC Controller, which illustrates that it's a feedback control system. The notation representing in the block diagram

\ddot{q}_d, \dot{q}_d and q_d are desired acceleration, desired velocity and desired position respectively were provided as the input to the system and acquires actual position (q_a) and velocity(\dot{q}_a) as an output.

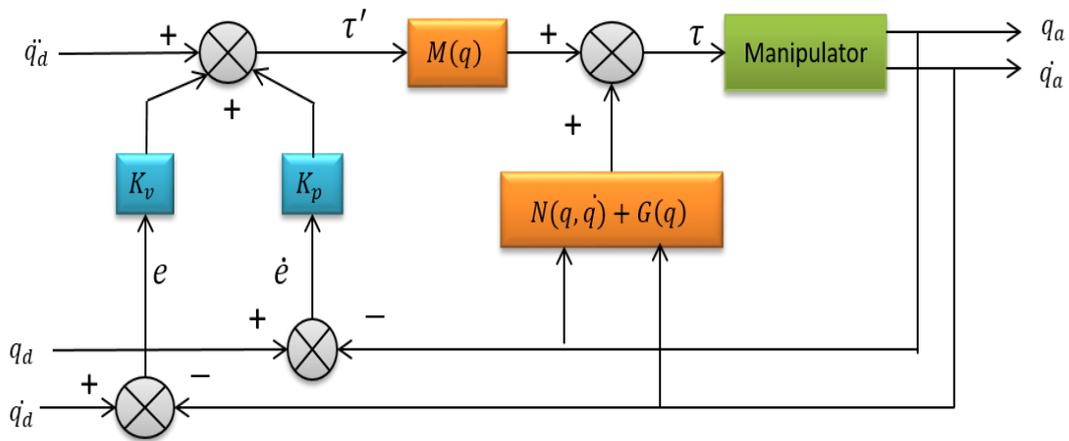


Figure 3. Block Diagram of PD–Computed Torque Control (PD-CTC)

Originally this algorithm is called as feedback Linearization Controller. It has assumed that the desired trajectory of the manipulator is $q_d(t)$ and $q_a(t)$ is the actual trajectory of the manipulator. The Tracking error can be defined as $e(t)$ in Eq. (39), i.e. displacement error:

$$e(t) = q_d(t) - q_a(t) \tag{3939}$$

The derivative of displacement error $e(t)$, we would be able to achieve velocity error denoted as $\dot{e}(t)$ stated in Eq. (40).

$$\dot{e}(t) = \dot{q}_d(t) - \dot{q}_a(t) \tag{40}$$

Similarly, on taking derivative of velocity error from Eq. (40) we would be able to achieve acceleration error represented as $\ddot{e}(t)$ stated in Eq. (41).

$$\ddot{e}(t) = \ddot{q}_d(t) - \ddot{q}_a(t) \tag{41}$$

Rewriting Eq. (27) with reference to actual angle denoted as q_a . The resultant is represented in Eq. (42).

$$M(q)\ddot{q}_a + N(q, \dot{q})\dot{q}_a + G(q) = \tau \tag{42}$$

If an alternative linear state space equation in the form $\dot{x} = Ax + BU$ can be, defined as Eq. (43).

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U \tag{40}$$

The Brunousky canonical form state that $U = -M^{-1}(q).N(q, \dot{q}) + M^{-1}(q).\tau$ with the help of Eq. (42) and Eq. (43) the Brunousky canonical form can be seen in Eq. (44) written in terms of state $x = [e^T \ \dot{e}^T]^T$. The U is represented as Eq. (45).

$$\frac{d}{dx} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U \tag{41}$$

With,

$$U = \ddot{q}_a + M^{-1}(q).\{N(q, \dot{q}) - \tau\} \tag{42}$$

Taking the inverse of the equation (45), the computed torque for the required arm as stated in Eq. (46).

$$\tau = M(q)(\ddot{q}_d - U) + N(q, \dot{q}) \tag{43}$$

On selecting the proportional–plus-derivative (PD) feedback for control input $U(t)$ results in PD computed torque control that guarantees the tracking of the desired trajectory as mentioned in Eq. (47).

$$\tau = M(q)(\ddot{q}_d + K_d \dot{e} + K_p e) + N(q, \dot{q}) \tag{44}$$

Equation (49) shows the resulting linear error dynamics from Eq. (47) and Eq. (48).

$$\tau' = \ddot{q} \quad (45)$$

$$\ddot{q} = \ddot{q}_d + K_d \dot{e} + K_p e \quad (46)$$

Or with $\ddot{e} = \ddot{q}_d - \ddot{q}_a$

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (47)$$

where, K_d and K_p are the Velocity and Position gain. For the critical damping performance of each joint, the Eq. (51) states the relationship between K_d and K_p .

$$K_d = 2\sqrt{K_p} \quad (48)$$

The computed torque control is used to linearize the error dynamics using nonlinear feedback, which provides better tracking performance in comparison with linear controllers. The computational cost is more as compared to linear controllers and inaccuracies in the dynamic model and other parameters limit the performance of the manipulator.

RESULTS AND DISCUSSIONS

The computed torque controller (CTC) control implemented for step responses. The simulation is implemented using the MATLAB/ Simulink Software. In this paper, the simulation results of the SCARA robot with 4 DOF is discussed moving the robot from its home to the final position taking consideration of different K_p and K_d values.

Performance Estimation

As stated in refer [15,16]; the performance estimation of the controller is estimated by its trajectory tracking. In PD-CTC controller the proportional and derivative gain directly affects the performance of the controller. The Performance of the PD-CTC controller is computed by tuning gains using trial and error. The CTC cases were considered with the different gain coefficient of the controller and their respective responses with respect to steady-state error and Root mean square error that can be seen in Table 1.

In PD-CTC controller two-gain coefficient have considered i.e. Proportional Gain (K_p) and Derivative gain (K_d). The proportional gain is directly proportional to the error. Therefore, if there is an increase in error we should increase the K_p value in the same proportion. To make a robot move in the desired trajectory the large value of K_p should be kept which will make the robot follow the desired trajectory or tries to reduce the error. The lower value of K_p makes the system sluggish, the reaction is slow to lead the heading change and it can happen it never reaches the desired values. Higher K_p made the system to respond rapidly and smoothly to reach the desired values. Too higher K_p will provide high control command even for smaller errors which leads to the system overshoot. Increasing K_p even after too higher value makes the system oscillating. As per the dynamics of the system, it may cause unnecessary vibrations to the system. The Derivative gain is the rate of change of error, which implies an increase in K_d will make the system to respond faster. Right value K_d reduced overshoot caused by increased K_p and system smooth and faster.

Considering the above tuning conditions of K_p and K_d the below mention cases are considered for PD-CTC Tuning. In Table 3, the case of PD-CTC with their respective Root mean square error and Steady-state error at each joint is calibrated. The trajectory tracking responses of each case can be seen in Figure 4.

Case 1: $K_d=60, K_p=360$

As the value of K_p is higher and respectively the value of K_d is less than K_p but relatively higher than in other cases. The selected values of K_p and K_d made the system to respond faster and follow the smoother trajectory to reach the desired angle without any overshoot.

Case 2: $K_d=2, K_p=10$

The value of K_p and K_d are lower with respect to case 1 but the value of K_p is higher than K_d similar to case 1. The lower value of K_p and K_d made system sluggish is response with respect to case 1. This makes the system under damped and causes overshoot in system response.

Case 3: $K_d=10, K_p=10$

In case 3, the value of K_p and K_d is equal. The large value of K_d will provide excessive force to the system to attain the desired trajectory. The equivalent value of K_p make system relatively sluggish without overshoot.

Case 4: $K_d=10, K_p=2$

In case 4, the value of K_p is much lower than the K_d . The lower value of K_p made the system extremely sluggish in response, which in turn system does not reach its desired trajectory in the system simulation time frame of 50 seconds.

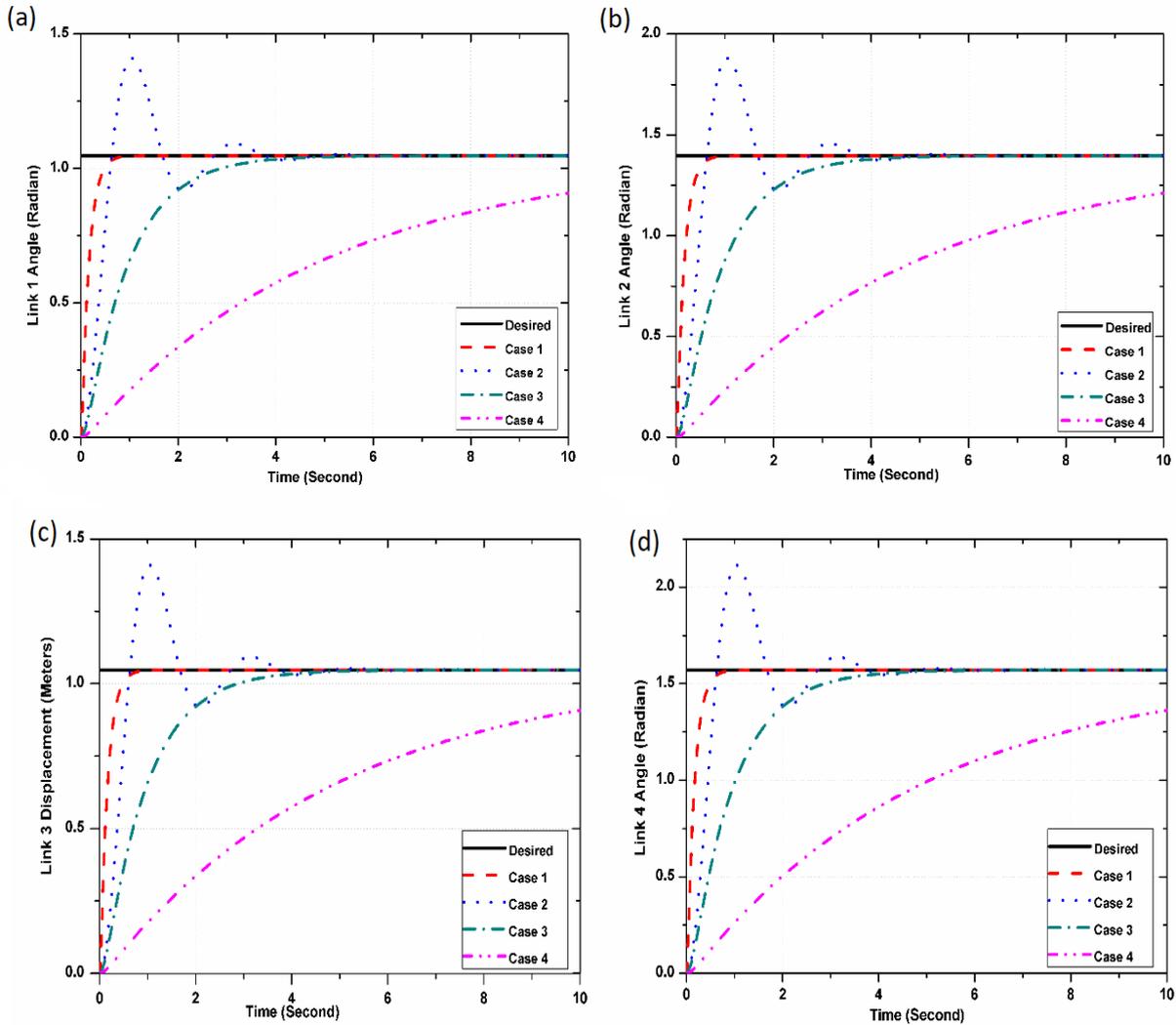


Figure 4. Computed Torque Control (CTC) at different cases and desired input step trajectory for (a) link 1 joint angle, (b) link 2 joint angle, (c) link 3 joint displacement and (d) link 4 joint angle

Table 3. Comparison table of Computed Torque Control (CTC) of different cases followed with their respective Steady State Error (SS Error) and Root Mean Square Error (RMS Error)

Case	$K_p 1$	$K_d 1$	$K_p 2$	$K_d 2$	$K_p 3$	$K_d 3$	$K_p 4$	$K_d 4$	SS Error1	SS Error2	SS Error3	SS Error4	RMS Error
Case 1	360	60	360	60	360	60	360	60	1.49×10^{-5}	1.986×10^{-5}	1.49×10^{-5}	2.234×10^{-5}	0.1003
Case 2	10	2	10	2	10	2	10	2	8.326×10^{-6}	1.110×10^{-5}	8.326×10^{-6}	1.249×10^{-5}	0.1959
Case 3	10	10	10	10	10	10	10	10	1.529×10^{-5}	2.039×10^{-5}	1.529×10^{-5}	2.294×10^{-5}	0.2456
Case 4	2	10	2	10	2	10	2	10	1.388×10^{-1}	1.851×10^{-1}	1.851×10^{-1}	1.388×10^{-1}	0.1704

From the above analysis illustrated in Table 3 based on SS and RMS Error the case 1 is considered as the best tuned PD-CTC controller with K_p and K_d valued as 360 and 60 respectively. The best case of PD-CTC controller case 1 is compared with the tuned PD and PID controller [8,17,18]. The trajectory tracking of the PD-CTC, PD and PID controller for step trajectory can be seen in Figure 5. According to Figure 5, the CTC, PD AND PID controller the CTC controller tracks the step input trajectory more precisely with the least RMS error of 0.1003.

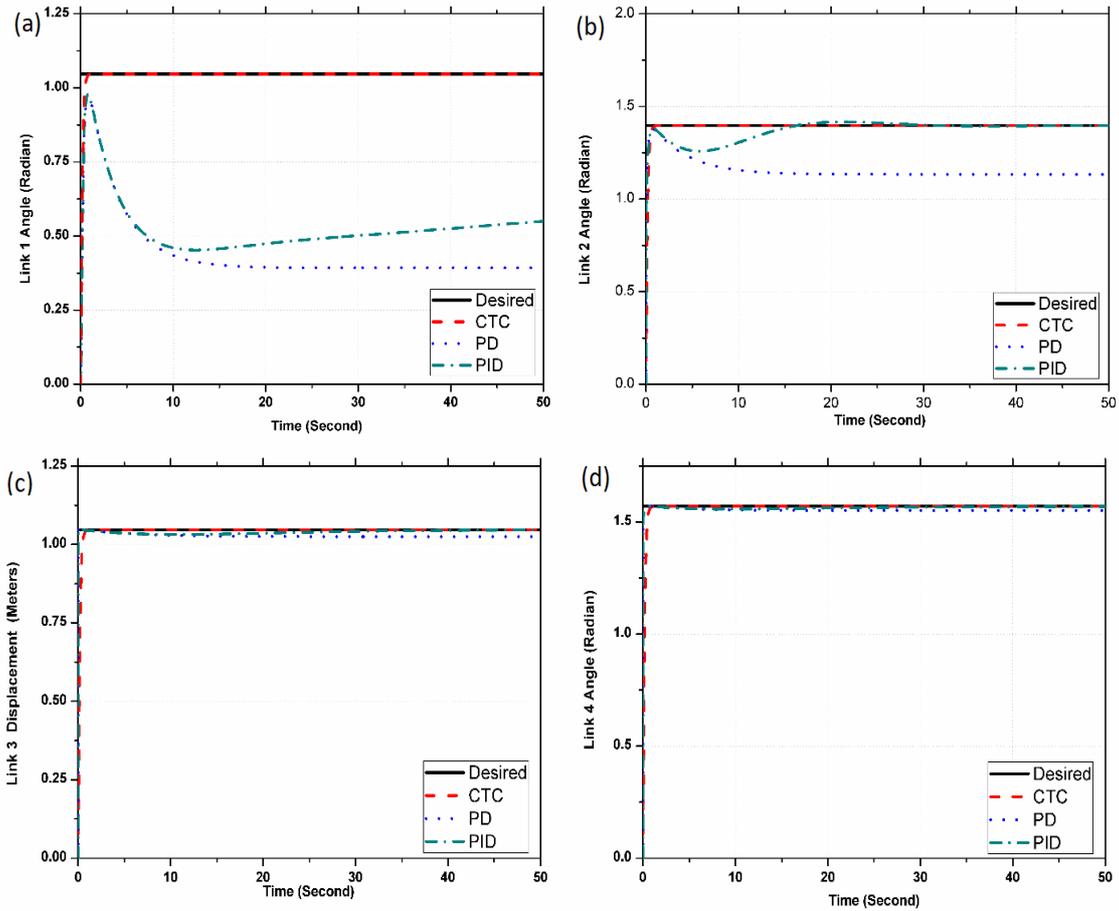


Figure 5. Computed Torque Control (CTC), Proportional-Derivative Control (PD), Proportional Integrated Derivative Control (PID) and desired input step trajectory for (a) link 1 joint angle, (b) link 2 joint angle, (c) link 3 joint displacement and (d) link 4 joint angle

For PD, PID and CTC the K_p and K_d values of each joint of the robot are mentioned in Table 4. The gain values of the controller for each joint are taken as constant in PD and PD- CTC whereas in PID control implementation the gain values of each joint of the manipulator are well-tuned.

Table 4. Different control techniques: CTC, PD and PID with their respective gains

Control Technique	Kp1	Ki1	Kd1	Kp2	Ki2	Kd2	Kp3	Ki3	Kd3	Kp4	Ki4	Kd4
CTC	10	0	2	10	0	2	10	0	2	10	0	2
PD	360	0	60	360	0	60	360	0	60	360	0	60
PID	300	0.00561	4	200	0.2206118	4	36	0.058339	6	30	0.05833	4

Table 5. Root Mean Square error of system and steady state error at each joint of the system at different control techniques

Control Technique	SS Error1	SS Error2	SS Error3	SS Error4	RMS error
CTC	1.49×10^{-5}	1.986×10^{-5}	1.49×10^{-5}	2.234×10^{-5}	0.1003
PD	0.6549	0.2639	0.02222	0	0.3534
PID	4.971×10^{-1}	2.291×10^{-4}	5.504×10^{-4}	7.090×10^{-1}	0.2486

On comparison of well-tuned PD and PID Controller with the best case of CTC control technique, their respective Steady-state error and Root mean square error can be seen in Table 5. The steady-state error for each joint of the robot manipulator as mentioned in Table 5 illustrates that CTC controller performs well in order to minimize the joint trajectory tracking. The steady-state error of each joint in CTC control is reduced to e^{-05} with Root Mean Square Error of 0.1003.

The RMS error of CTC is 71.61% less than PD controller and 59.65% less than PID controller. The performance of CTC controller for trajectory tracking is best with respect to PD and PID control taking steady-state and root mean square error as an estimation function.

CONCLUSIONS

In this paper, the kinematic and dynamic mathematical modeling of four - DOF SCARA robot is illustrated. The implementation of mathematical modeling performed in the simulation platforms MATLAB/SIMULINK. The kinematic modeling is computed by the analytical method using D-H notations. The dynamic modeling of the robot is computed using Newton Euler Method. In order to test the system modeling, which is a surgical robot we need high performance, precise, robust and speedy control, therefore, we choose three feedback loop controllers i.e., CTC, PD and PID. The CTC control technique is illustrated and implemented in MATLAB/Simulink with different Proportional Gain (K_p) and Derivative Gain (K_d) using the trial and error method. The best-suited CTC with its gain values is further compared with best-tuned gains of PD and PID controller in order to have a performance comparison of the different implemented controllers. This research proves that CTC with the optimized value of K_p and K_d is successfully proven to be having the least amount of RMS error 0.1003.

ACKNOWLEDGMENT

The paper presented the results of a search project performed in India. University of Petroleum and Energy Studies, Dehradun funded and supported this project under SEED research grant. We would like to thank and acknowledge their financial and logistics aid.

REFERENCES

- [1] J. J. Craig, P. Prentice, and P. P. Hall, "Introduction to Robotics Mechanics and Control Third Edition," 2005.
- [2] S. Guo, Q. Chen, N. Xiao, and Y. Wang, "A Fuzzy PID Control Algorithm for The Interventional Surgical Robot with Guide Wire Feedback Force," *IEEE International Conference on Mechatronics and Automation*, pp. 426–430, 2016, doi: 10.1109/ICMA.2016.7558601.
- [3] G. Q. Ma, Z. L. Yu, G. H. Cao, Y. Bin Zheng, and L. Liu, "The Kinematic Analysis and Trajectory Planning Study of High-Speed SCARA Robot Handling Operation," *Appl. Mech. Mater.*, vol. 687–691, pp. 294–299, 2014, doi:10.4028/www.scientific.net/amm.687-691.294.
- [4] A. Visioli and G. Legnani, "On the trajectory tracking control of industrial SCARA robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 224–232, 2002, doi: 10.1109/41.982266.
- [5] A. Gupta, A. K. Mondal, and M. K. Gupta, "Kinematic, Dynamic Analysis and Control of 3 DOF Upper-limb Robotic Exoskeleton," *J. Eur. des Systèmes Autom.*, vol. 52, no. 3, pp. 297–304, Aug. 2019, doi:10.18280/jesa.520311.
- [6] S. Guo, Q. Chen, N. Xiao, and Y. Wang, "A Fuzzy PID Control Algorithm for The Interventional Surgical Robot with Guide Wire Feedback Force," in *International Conference on Mechatronics and Automation*, pp. 426–430, 2016.
- [7] N. N. Son, C. Van Kien, and H. P. H. Anh, "A novel adaptive feed-forward-PID controller of a SCARA parallel robot using pneumatic artificial muscle actuator based on neural network and modified differential evolution algorithm," *Rob. Auton. Syst.*, vol. 96, pp. 65–80, 2017, doi: 10.1016/j.robot.2017.06.012.
- [8] J. Weng, "Design and Implementation of the SCARA Robot Arm," no. Icarob, pp. 71–74, 2017.
- [9] V. Verma, P. Chauhan, and M. K. Gupta, "Disturbance Observer-assisted Trajectory Tracking Control for Surgical Robot Manipulator," *J. Eur. des Syst. Autom.*, vol. 52, no. 4, pp. 355–362, 2019, doi: 10.18280/jesa.520404.
- [10] A. Gupta, V. Verma, A. Kumar, P. Sharma, M. K. Gupta, and C. S. Meera, "Stabilization of underactuated mechanical system using LQR technique," in *Advances in Intelligent Systems and Computing*, vol. 479, pp. 601–608, 2017, doi: 10.1007/978-981-10-1708-7_68.
- [11] S. Yamacli and H. Canbolat, "Simulation of a SCARA robot with PD and learning controllers," *Simul. Model. Pract. Theory*, vol. 16, no. 9, pp. 1477–1487, 2008, doi: 10.1016/j.simpat.2008.08.00.
- [12] A. Munawar, "Implementation of a Surgical Robot Dynamical Simulation and Motion Planning Framework," 2015.
- [13] A. Ogulmus, A. Cakan, and M. Tinkir, "Modeling And Position Control Of Scara Type 3D Printer," *Int. J. Sci. Technol. Res.*, vol. 5, no. 12, pp. 140–143, 2016.
- [14] V. Verma et al., "IoT and Robotics in Healthcare," in *Medical Big Data and Internet of Medical Things*, Boca Raton : Taylor & Francis, CRC Press, pp. 245–269, 2018, doi: 10.1201/9781351030380-10.
- [15] F. Piltan, A. Taghizadegan, and N. B. Sulaiman, "Modeling and Control of Four Degrees of Freedom Surgical Robot Manipulator Using MATLAB/SIMULINK," *Int. J. Hybrid Inf. Technol.*, vol. 8, no. 11, pp. 47–78, 2015, doi: 10.14257/ijhit.2015.8.11.05.

- [16] N. A. Anang A. Abdullah, Z. Jamaludin, T.H. Chiew, M. Maharof, S.N. Salim, Z. Retas, and S.C. Junoh., "Tracking performance of NPID controller for cutting force disturbance of ball screw drive," *J. Mech. Eng. Sci. ISSN*, vol. 11, no. 4, pp. 3227–3239, 2017, doi: 10.15282/jmes.11.4.2017.25.0291.
- [17] L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta," *ISA Trans.*, pp. 1–17, 2018, doi: 10.1016/j.isatra.2018.04.010.
- [18] M. A. Shamseldin, M. Sallam, A. M. Bassiuny, and A. M. Abdel Ghany, "Real-time implementation of an enhanced nonlinear PID controller based on harmony search for one-stage servomechanism system", *J. Mech. Eng. Sci.*, vol. 12, no. 4, pp. 4161-4179, Dec. 2018, doi: 10.15282/jmes.12.4.2018.13.0359.
- [19] F. Inel and S. Babesse, "Adaptive sliding mode control of a novel cable driven robot model", *J. Mech. Eng. Sci.*, vol. 13, no. 2, pp. 5150 - 5162, Jun. 2019, doi: 10.15282/jmes.13.2.2019.26.0423.
- [20] M. Polishchuk, M. Suyazov, and M. Opashnyansky, "Study on numerical analysis of dynamic parameters of mobile walking robot", *J. Mech. Eng. Sci.*, vol. 14, no. 1, pp. 6380 - 6392, Mar. 2020, doi: 10.15282/jmes.14.1.2020.14.0499.
- [21] C. A. Akhadkar, A. B. Deoghare, and A. M. Vaidya, "Simulation and experimental response of four-bar mechanism with tolerance stack", *J. Mech. Eng. Sci.*, vol. 13, no. 1, pp. 4512-4535, Mar. 2019, doi: 10.15282/jmes.13.1.2019.13.0383.
- [22] A. Gupta, , A. Singh, V. Verma, A.K. Mondal, and M.K Gupta, "Developments and clinical evaluations of robotic exoskeleton technology for human upper-limb rehabilitation". *Advanced Robotics*, pp.1-18, Apr. 2020, doi: 10.1080/01691864.2020.1749926.
- [23] T. H. Chiew, Z.Jamaludin, A.Y. Hashim, L. Abdullah, N.A. Rafan, and M. Maharof, "Second order sliding mode control for direct drive positioning system", *J. Mech. Eng. Sci.*, 11(4), pp.3206-3216, Dec 2017, doi: 10.15282/jmes.11.4.2017.23.0289.
- [24] V. Verma, R. Kumar, and V. Kaundal, "Implementation of ladder logic for control of pipeline inspection robot using PLC." *Advances in Intelligent Systems and Computing*, vol. 479, pp. 965-971, 2017, doi: 10.1007/978-981-10-1708-7_113.