

ADAPTIVE PD SLIDING MODE CONTROL FOR 4 DOF SCARA VARIANT

by

Manjeet Tummalapalli
Bachelor of Technology, K L University, Guntur (2016)

A project

presented to Ryerson University

in partial fulfilment of the

requirements for the degree of

Master of Engineering

in the program of

Aerospace Engineering

Toronto, Ontario, Canada, 2019

©Manjeet Tummalapalli, 2019

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A PROJECT

I hereby declare that I am the sole author of this project. This is a true copy of the project, including any required final revisions.

I authorize Ryerson University to lend this project to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my project may be made electronically available to the public.

ADAPTIVE PD SLIDING MODE CONTROL FOR 4 DOF SCARA VARIANT

Manjeet Tummalapalli

Master of Engineering, Aerospace Engineering, Ryerson University, Toronto (2019)

Abstract

This project proposes a new SCARA variant with 4 degree of freedom. The proposed variant is achieved by swapping joint 2 and joint 3 of the standard SCARA robots. An adaptive controller is defined based on the advantages and disadvantages of PD, and SMC controllers. The purpose of the project is to understand the dynamics of the variant and to track the performance for trajectories. Simulations for tracking performance are carried under linear and circular trajectories. The variant is studied over the three controllers; PD, PD-SMC and A-PD-SMC. The variant under the adaptive controller is most efficient in terms of tracking performance and the control inputs to the system. The system is simulated under high speed and with the influence of friction at the joints. The control gains are held constant for both the trajectories and hence the controller is able to perform good under changing trajectories. Due to the use of the adaptive law, the system is at the ease of implementation and since no priori knowledge if the system is needed, it is model free. Therefore, the proposed adaptive PD-SMC has proven to provide good, robust trajectory tracking.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to my supervisor, Dr. Puren Ouyang, without whose continued guidance, profound support and extraordinary patience the completion of this project would not have been possible.

Finally, I would like to extend my heartfelt thanks to my family and friends. Thank you for all the love and support. You have my deepest gratitude for your kindness and generosity.

Table of Contents

Abstract.....	iii
List of Tables.....	vii
List of Figures	viii
Chapter 1: Introduction	1
1.1 Background.....	1
1.2 Motivation and Objectives.....	6
1.3 Organization of Contents	8
Chapter 2: Literature Review	9
2.1 PD/PID Control.....	10
2.1.1. PD with Desired Gravity Compensation	11
2.1.2 Non-Linear PD/ PID Control	12
2.2 Other Control Methods.....	14
2.2.1 Adaptive Control.....	14
2.2.2 Learning Control.....	19
2.3 Sliding Mode Control.....	19
2.4 Trajectory Planning	22
2.5 Control of Scara Robot.....	23
2.6 Remarks.....	26
Chapter 3. Manipulator Design.....	27
3.1 Coordinate Frames.....	27
3.2 Forward Kinematics	28
3.3 Inverse Kinematics.....	30
3.4 Singularity Analysis	32
3.5 Dynamic Analysis.....	34
3.6 Remarks.....	39
Chapter 4: Adaptive PD Sliding Mode Control.....	41
4.1 Dynamic Model.....	41
4.2 PD Formulation	42
4.3 PD-SMC Formulation.....	43
4.4 Adaptive PD-SMC Formulation.....	45

4.5 Friction in Joints.....	46
4.6 Remarks.....	46
Chapter 5: Simulations and Results	48
5.1 Simulation Setup.....	48
5.2 Trajectory Planning for Simulations	49
5.3 Scara Robot Simulation Results for Linear Trajectories	50
5.3.1 Tracking Errors of Linear Trajectory I.....	52
5.3.2 Control Torques for Linear Trajectory I.....	54
5.3.3 Task Space End-Effector Tracking for Linear Trajectory I.....	56
5.3.4 Tracking Errors for Linear Trajectory II	57
5.3.5 Control Torques for Linear Trajectory II	59
5.3.6 Task Space End-Effector Tracking for Linear Trajectory II	60
5.4 Scara Robot Simulation Results for Circular Trajectories	62
5.4.1 Tracking Errors of Circular Trajectory I.....	64
5.4.2 Control Torques for Circular Trajectory I.....	66
5.4.3 Task Space End-Effector Tracking for Circular Trajectory I	67
5.4.4 Tracking Errors for Circular Trajectory II.....	68
5.4.5 Control Torques for Circular Trajectory II:.....	70
5.4.6 Task Space End-Effector Tracking for Circular Trajectory II:.....	72
5.5 Remarks.....	73
6.1 General Review.....	76
6.2 Main Contributions	77
6.3 Future Work	78
List Of Appendices	79
Singularity Analysis.....	79
Appendix A Finding The Transformation Matrices	79
Appendix B Singularity Analysis	79
Appendix C Dynamic Equations	80
Matlab Function for Control	82
References	90

List of Tables

Table 3.1 D-H Parameters of the SCARA Variant System	28
Table 5.1 SCARA Robot Specifications	49
Table 5.2 Linear Trajectories Specifications	50
Table 5.3 Control Parameter Specifications of Linear Motion	51
Table 5.4 Mean and Standard Deviation of Axial Tracking Errors for Linear Trajectory	54
Table 5.5 Maximum input torque for the Joints	54
Table 5.6 Mean and Standard Deviation for End-Effector in Task Space for Linear Trajectory <i>I</i>	57
Table 5.7 Mean and Standard Deviation of Axial Tracking Errors for Linear Trajectory <i>II</i>	59
Table 5.8 Maximum input torque for the Joints for Linear Trajectory <i>II</i>	59
Table 5.9 Mean and Standard Deviation for End-Effector in Task Space	62
Table 5.10 Circular Trajectory Specifications	62
Table 5.11 Mean and Standard Deviation of Axial Tracking Errors for Circular Trajectory <i>I</i>	65
Table 5.12 Maximum input torque for the Joints	66
Table 5.13 Mean and Standard Deviation for End-Effector Tracking Error in Task Space for Circular Trajectory <i>I</i>	68
Table 5.14 Mean and Standard Deviation of Axial Tracking Errors for Circular Trajectory <i>II</i>	70
Table 5.15 Maximum input torque for the Joints for Linear Trajectory <i>II</i>	71
Table 5.16 Mean and Standard Deviation for End-Effector in Task Space for Circular Trajectory <i>II</i> .	73

List of Figures

Fig. 1.1 Estimated worldwide annual shipments of industrial robots by regions	2
Fig. 1.2 Estimated annual supply of industrial robots at year-end worldwide by industries (2015-2017)	2
Fig. 1.3 Estimated annual worldwide supply of industrial robots 2009 – 2017 and 2018*- 2021*	3
Fig. 1.4. SCARA Configuration	4
Fig. 1.5 SCARA Workplace	5
Fig. 2.1 PD Control with desired gravity compensation	12
Fig. 2.2 Nonlinear closed-loop system	13
Fig. 2.3 Feedback and Adaptive Loop for Adaptive Controller	15
Fig. 2.4 Gain scheduling Controller	16
Fig. 2.5 MRAC Laws (a) Series High-Gain Scheme (b) Parallel Scheme	17
Fig. 2.6 Self-tuning Controller	18
Fig. 2.7 Chattering about the sliding surface	21
Fig. 2.8 Tracking and Contour Error	23
Fig. 3.1 The SCARA Variant, with directions of motion of joints	28
Fig. 3.2 The SCARA Variant, defining the coordinate frames and D-H Parameters	28
Fig. 3.3 Trigonometric method for finding the inverse kinematics	31
Fig. 3.4 Case 1: The lengths L_1 & L_3 are = 0, joints overlap.	33
Fig. 3.5 Case 2: $\theta_3 = 0, \pi$, the arm is at full extension or overlap with link 2.	34
Fig. 5.1 The selected two trajectories for Linear Motion	51
Fig. 5.2 Angular positions and velocities for joints and Position of the End-effector for Linear Trajectory <i>I</i>	52
Fig. 5.3 The angular positions, angular velocities of the Joints and the End-effector position for Trajectory <i>II</i>	52
Fig. 5.4 The Tracking Errors for Joint 1 and Joint 2 between the controllers	53
Fig. 5.5 Tracking Errors for Joint 3 and Joint 4 between the controllers	53
Fig. 5.6 Input control torques for Linear Trajectory <i>I</i> , for the joints.	55
Fig. 5.7 (a) Tracking Error of the End-Effector in Task Space for Linear Trajectory <i>I</i>	56
Fig. 5.7 (b) Tracking Error of the End-Effector in Task Space for Linear Trajectory <i>I</i> for PD-SMC & A-PD-SMC Controllers	57
Fig. 5.8 The Tracking Errors for Joint 1 and Joint 2 between the controllers for Trajectory <i>II</i>	58
Fig. 5.9 The Tracking Errors for Joint 3 and Joint 4 between the controllers for Trajectory <i>II</i>	58

Fig. 5.10 Input control torques for Linear Trajectory <i>II</i> , for the joints.	60
Fig. 5.11(a) End-Effector Task Space Tracking Error for Linear Trajectory <i>II</i>	61
Fig. 5.11 (b) Tracking Error of the End-Effector in Task Space for Linear Trajectory <i>I</i> for PD-SMC & A-PD-SMC Controllers.	61
Fig. 5.12 The selected two trajectories for Circular Motion	63
Fig. 5.13 Angular positions and velocities for joints and Position of the End-effector for Circular Trajectory <i>I</i>	63
Fig. 5.14 Angular positions and velocities for joints and Position of the End-effector for Circular Trajectory <i>II</i>	64
Fig. 5.15 The Tracking Errors for Joint 1 and Joint 2 between the controllers	64
Fig. 5.16 Tracking Errors for Joint 3 and Joint 4 between the controllers	65
Fig. 5.17 Input control torques for Circular Trajectory <i>I</i> , for the joints.	67
Fig. 5.18 (a) End-Effector Task Space Tracking Error for Circular Trajectory <i>I</i>	67
Fig. 5.18 (b) End-Effector Task Space Tracking Error for PD-SMC & A-PD-SMC Controllers	68
Fig. 5.19 The Tracking Errors for Joint 1 and Joint 2 between the controllers for Trajectory <i>II</i>	69
Fig. 5.20 The Tracking Errors for Joint 3 and Joint 4 between the controllers for Trajectory <i>II</i>	69
Fig. 5.21 Input control torques for Circular Trajectory <i>II</i> , for the joints	71
Fig. 5.22 (a) End-Effector Task Space Tracking Error for Circular Trajectory <i>II</i>	72
Fig. 5.22 (b) End-Effector Task Space Tracking Error for PD-SMC & A-PD-SMC Controllers	72
Fig. 5.23 Mean Joint 2 Tracking Error for PD-SMC & A-PD-SMC Controllers	73
Fig. 5.24 Mean End-Effector Tracking Error for PD-SMC and A-PD-SMC	74
Fig. 5.25 Average Torques for the joints over the various trajectories.	75

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Bringing together of two technologies namely: numerical control machines for precise manufacturing and tele-operated machines for the remote handling of hazardous materials lead to the development of early industrial robots' development in the early 1960s. The term robotics was first coined by Russian Isaac Asimov, popularly defining it as the science studying the intelligent connection between perception and action. Using the definition for robotics, a robotic system is said to be a complex system, functionally represented by multiple subsystems; controllers, actuators and sensors. A manipulating apparatus, namely a robot manipulator is a mechanical structure consisting of a sequence of rigid bodies linked together by means of articulations (joints). The first robotic manipulators were characterized by versatility, adaptability and execution repeatability (Siciliano B. S., 2009).The widespread of robot technology can be viewed as a result of the increase in the wider range of applications in manufacturing industry due to the reduction of manufacturing costs, increase in productivity and quality and the possibility of elimination of dangerous tasks for the human operator.

The trend towards automation continued to increase following the global financial crisis in 2009. In the year 2017, the robot sales worldwide increased by 30%, a new peak for fifth year in a row (International Federation of Robotics, 2018). The IFR Executive summary also

states that since 2010, the demand of industrial robots has accelerated considerably due to the ongoing trend towards automation and continued innovative technical improvements in industrial robots.

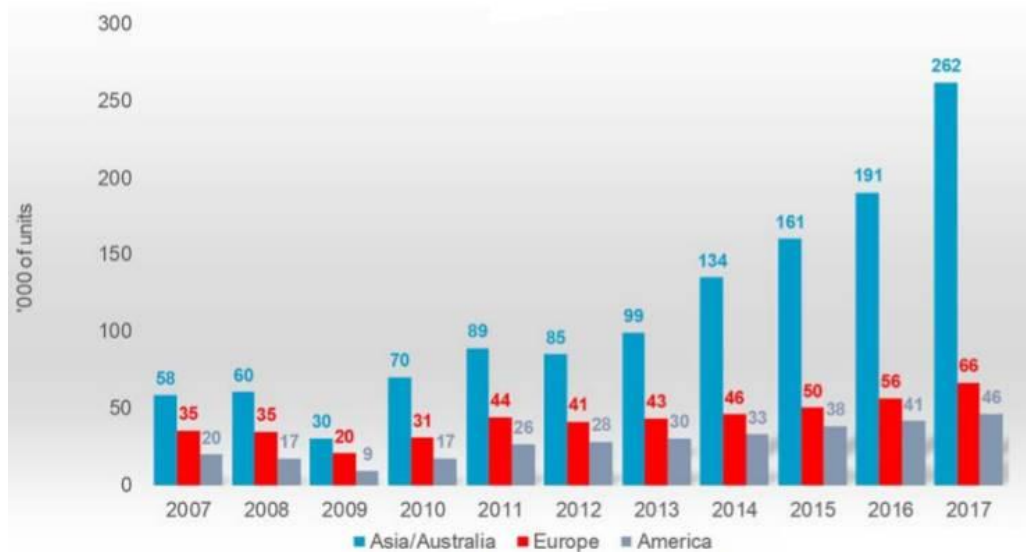


Fig. 1.1 Estimated worldwide annual shipments of industrial robots by regions (International Federation of Robotics, 2018)

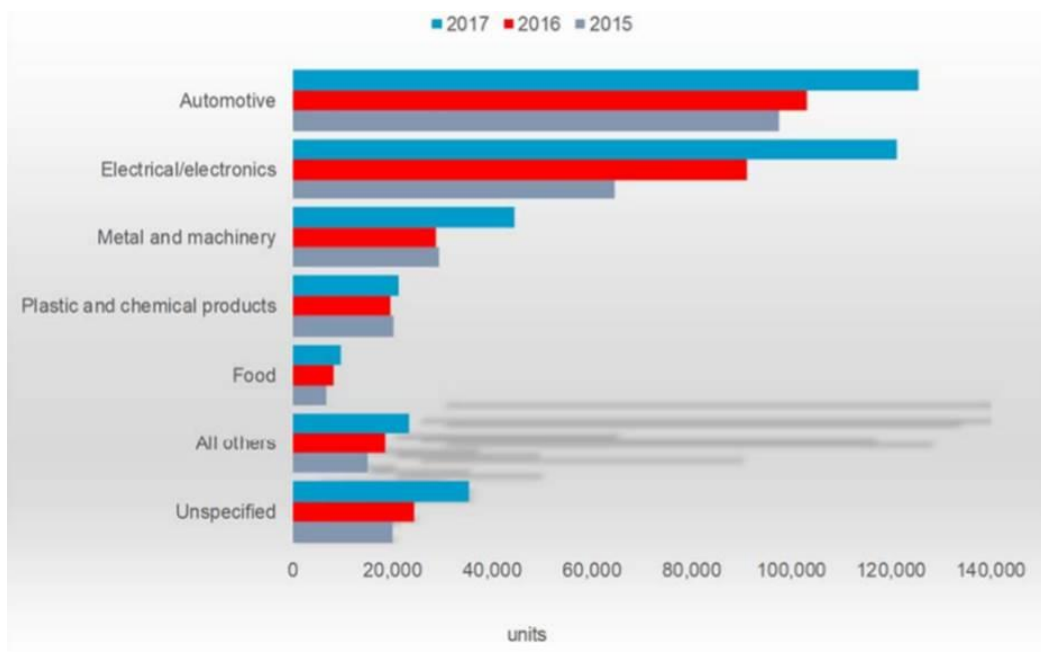


Fig. 1.2 Estimated annual supply of industrial robots at year-end worldwide by industries (2015-2017) (International Federation of Robotics, 2018)

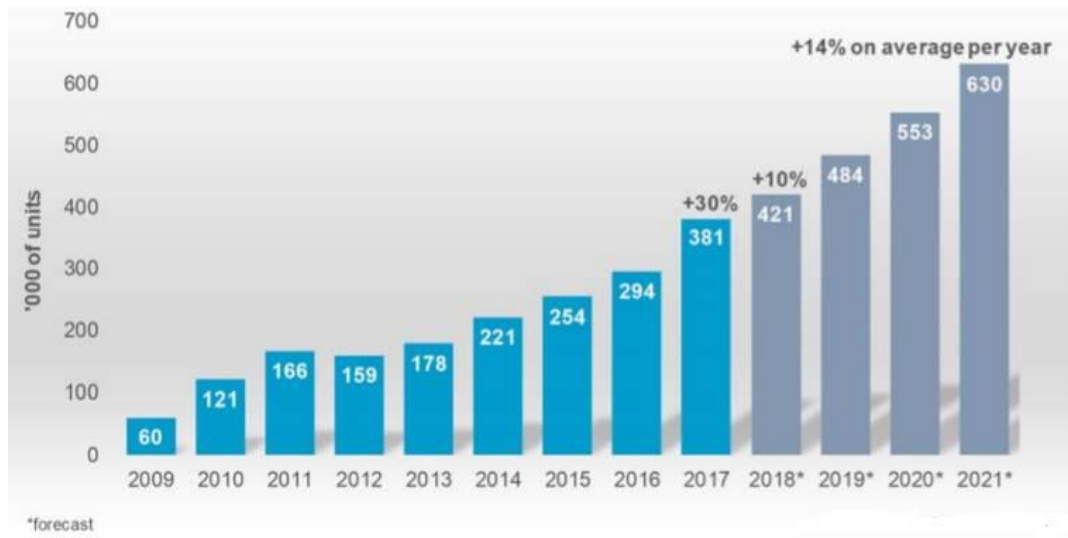


Fig. 1.3 Estimated annual worldwide supply of industrial robots 2009 – 2017 and 2018*- 2021*
(International Federation of Robotics, 2018)

Fig 1.1 depicts the graphical description of the worldwide estimate of annual shipments for industrial robots whereas Fig. 1.2 depicts the distribution of robots by industry. From both we can see that the peak is at year 2017 has seen 262,000 units shipped in Asia/Australia and that the automotive and electrical/electronics industries are supplied with the highest number of robots per year with about 120,000 units each for 2017 alone. Due to the wide range of applications towards automation, industrial robots show significant characteristics of versatility and accuracy. According to Figure 1.2, industrial robots remain an essential component to many automated systems.

Adding to the earlier definition of robot manipulators, the fundamental structure of a manipulator is a serial or open kinematics chain which has good dexterity and simple straight forward dynamics while on contrary the closed chain manipulators have a fundamental advantage of high structural stiffness. Further, the geometry of the robotic manipulators is decided by its industrial application.

SCARA geometry, where the acronym stands for Selective Compliance Assembly Robot Arm, is a special geometry for robotic manipulators that is designed using three revolute and one prismatic joint. SCARA was invented by Sankyo and Pentel under the guidance of Hiroshi Makino at the University of Yamanshi in the year 1981 (Shabana, 2001). These joints are arranged in such a way that all the axes of motion are parallel as shown in Fig. 1.4. The SCARA robot characterizes in increased mechanical properties offering high stiffness to vertical loads with relatively compliant laterally. This feature is convenient for a variety of assembly tasks. In SCARA manipulator, the arm is movable in the X and Y directions but rigid in the Z direction (Shabana, 2001). The workspace of a robotic is the 3D or planar surface or plane on which the end effector can reach to complete the task given to the robot. The workspace of a conventional SCARA robot is given in Fig. 1.5. The parameter d_3 is for the prismatic joint and describes the motion in the Z direction. θ_2 & θ_1 are the angles of rotations for the arms (links) describing the motion of the revolute joint. It should be noted that the fourth degree of freedom is from the end-effector and is also a revolute joint with the axis of rotation in the Z direction (θ_4) but isn't shown in the Fig.1.4.

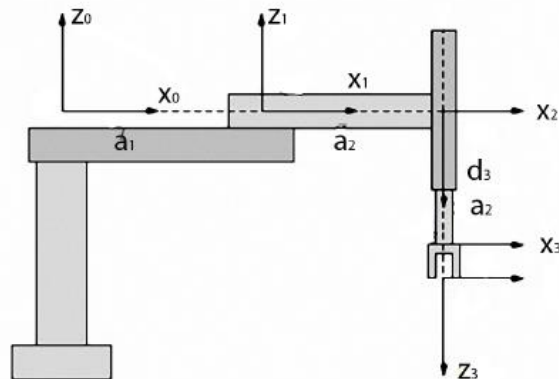


Fig. 1.4 SCARA Configuration (Zargoun, 2013-2014)

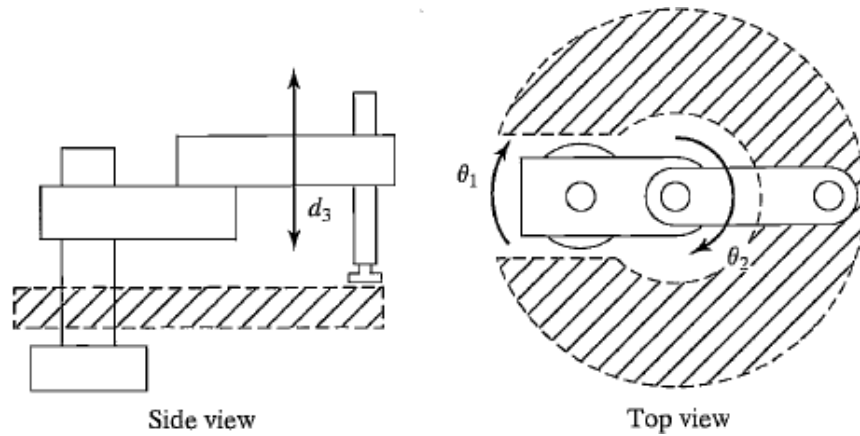


Fig. 1.5 SCARA workplace (Craig J. J., 2005)

Robot applications involves the completion of a generic task with a specific prescribed motion. Control laws are utilized for this purpose. Since, programmable robots play a significant role in the industry, the need for research and development of effective control laws is critical. The errors of the system, the inherent imprecisions and disturbances are compensated using the efficient design of control laws. The imprecisions can be caused due to unmodeled friction, vibrational dynamics, process related disturbances, hardware deficiencies and other inaccuracies of the system dynamics (Slotine, 1991). Due to the many uncertainties and parameter fluctuations, there is a need for achieving robust controllers. Almost most of the industrial applications require high accuracy trajectory tracking. It is important to consider each of the uncertainties and previously mentioned imprecisions into account when implementing a control law.

1.2 MOTIVATION AND OBJECTIVES

The present available standard SCARA was developed in the year 1981 and has been extensively used for industrial applications. What if the applications' requirements are modified, per say the physical limitation of the workspace restricts vertical movement in the axis three? Or maybe there is a modification to the applications of the robot, which can cause a change in the level of tracking performance needed, or the available control forces for tracking, and the power of the control activity, there may arise a question that a modification to the robot configuration may accompany to these changes. The motivation behind this research is the popularity and applications of a SCARA robot and the complexity involved in analysing its kinematics and dynamics. The main objective is to try to relocate the vertical prismatic joint to axis 2 and see how it effects in the dynamics and the performance. Work is done to first generalize the equations and state a generic sequence for achieving the dynamics of any manipulator, including the SCARA variant. Further to improve the tracking performance of the robot and hence to reduce the tracking errors. The outcome of the project is to propose a new adaptive hybrid control law that is a combination of PD and SMC controllers, using both the advantages of each to achieve a powerful and robust control law. The tracking is done on joint level of the robot, because by providing accurate joint control the tracking performance of the manipulator (end-effector) can be improved.

The proposed hybrid Adaptive PD-SMC law has the following goals:

1. Alternative to PID/PD control laws.

2. To achieve a control law that incorporates the advantages of nonlinear control, PD & SMC control schemes advantages.
3. To improve the overall performance of the manipulator by reducing individual joint errors.
4. Considering the fluctuations in parameters and friction, the control should be able to provide a robust tracking performance.
5. To provide a model-free control law that doesn't require priori knowledge of the dynamics of the system.

In this project the proposed adaptive hybrid PD-SMC control law will be tested for various trajectories. Simulations will be done using the available PD, PD-SMC and the proposed adaptive PD-SMC for linear and circular contours and the errors in tracking will be compared. Using linear and non-linear contour, the project aims at proving that given a set of control gains, the proposed control law is efficient in the sense that the tracking is better in performance when compared to PD and PD-SMC controllers. To achieve the said goals, the following objectives will have to be achieved:

1. The coordinate frames and DH parameters of the robot are defined.
2. Kinematic analysis, which gives the relationship between the tool tip pose with the individual joint pose and vice-versa.
3. Perform singularity analysis, to find the points that produce unbounded results in joint velocities and torques.
4. Model the dynamics of the 4 DOF SCARA variant.

5. Compare the dynamics of the variant to that of the standard SCARA robot.
6. Formulate the hybrid adaptive PD - Sliding Mode control law.
7. Compare the hybrid control law to the conventional PD and PD-SMC law at joint and end-effector level.

1.3 ORGANIZATION OF CONTENTS

The chapter 2 deals with the literature review of previous research on robotic manipulator control. The various results obtained in improving tracking performance will be summarized. The methodology behind hybridization of various control laws will also be briefly discussed.

Chapter 3 will contain the kinematics and singularity analysis of the robot. It will also cover the mathematical formulation of the manipulator dynamics. From the obtained dynamics results, the formulation of standard PD and combination of PD-SMC will also done. This chapter will also formulate the proposed hybrid adaptive control law.

Chapter 4 will contain the various selected control parameters for control laws. The chapter will later contain results from simulations for the 4 DOF SCARA robot using PD, PD-SMC and adaptive PD-SMC control laws for linear and circular contours.

Chapter 5 deals with the summarization of the results, and finally Chapter 6 mentions the conclusions and the scope for future work in enhancing the hybrid control laws.

Chapter 2: LITERATURE REVIEW

Robotic systems possess intense nonlinearities and are a subject to modelling parameter uncertainties and external disturbances. These cause the measurement of system to be tedious and designing a control law to be of more difficulty. Convergence of the system state to the requested state will require a high-level control law (Adelhedi, Jribi, Bouteraa, & Derbel, 2015). An idealized controller may be implemented to achieve good performance when the dynamics of the system is perfectly known. However, in most practical applications due to the complexity of the system, the physical parameters are difficult to determine. External disturbances such as joint friction and payloads can make the use of ideal controllers difficult. The prime reason why adaptive controllers were introduced was to compensate for the dynamic model uncertainties and external disturbances (Sage, 1999)

There exist some control laws that do not require previous knowledge of the system dynamics, but these control methods draw back in the sense that they do not account for the nonlinearities in the dynamics. For applications that require precise and accurate control systems, the said control laws will not be enough. The most common in use with respect to the present industrial robots is 'proportional, integral, derivative' (PID) control for each degree of freedom. More sophisticated nonlinear control laws have been developed such as computed torque and inverse dynamic control wherein linearization and decoupling for the equations of motion are done. Owing to the modelling uncertainties, nonlinear adaptive control techniques have been developed where the dynamics parameters are identified online (during the process) (Dombre & Khalil, 2002)

The control technique followed and the way it is implemented on the system has an influence on the performance and range of applications of a robotic manipulator. For example, if trajectory control is the prime objective of the control law, software and hardware implementations difficulties will arise, which are different from point-to-point control where reaching the final position is of importance. Manipulator mechanical design also has an influence on the kind of control scheme utilized. For example, the control problem of a cartesian manipulator is substantially different from that of an anthropomorphic manipulator. Finally, the driving system of the joints also has an influence on the control scheme implemented. For example, if a manipulator is actuated by electric motors, the presence of gears of high ratios in the system will linearize the system dynamics and thus decouple the joints, in turn reducing the nonlinear effects. But this comes with a price of occurrence of joint friction, elasticity and backlash that limits the system performance. On the other hand, consider a robot actuated by direct drives; the system will have eliminated the drawbacks from friction, elasticity and backlash but the nonlinearities and coupling between joints comes to play (Siciliano, 1999).

Therefore, based on the application, the level of accuracy and easy of computing, various control laws have been developed. Here, some popular control systems have been reviewed.

2.1 PD/PID CONTROL

PD/ PID control is designed based on the principle of linearization of the system about an operating point. Since the stability of the entire system depends of the local linearization, this control scheme is guaranteed to be locally stable. (Murray, Li, & Sastry , 1994). Because of the simple structure and easy implementation, PID control (including PI and PD control)

has picked up widespread popularity amongst engineering applications. Over 90% of all control schemes use some form of PID control (Åström, 2001). Consider PD control law in its simplest form

$$\tau = -K_v \dot{e} - K_p e \quad (2.1)$$

where K_v and K_p are positive definite matrices, with the tracking error is given by $e = \theta - \theta_d$ and τ represents the torque of the system. (Murray, Li, & Sastry , 1994). It can be observed that the control law doesn't contain a feedback term, therefore for complicated trajectories achieving exact tracking is next to impossible.

2.1.1. PD WITH DESIRED GRAVITY COMPENSATION

The general PID controller calculates the required actuation forces from the difference between the desired and actual joint position, which is nothing but the error. In other words, as the force is based in the positional error, it implies that in general the error is never equal to zero. If the dynamic model of the robot (under the influence of inertial, Coriolis centripetal and gravitational forces) is known, the forces required for necessary performance can be predicted. Further, these forces can be generated by the robot motors, regardless of the position error signal. (Bajd M.)

The choice of PID gains relies on relatively complex formulae in order to ensure global asymptotic stability (Qu, 1991) (Arimoto, 1984). The landmark work of (Takegaki, 1981) provided a robust position control that is easily tuned for global asymptotic stability. The PD control law with desired gravity compensation for setpoint control proposed by (Takegaki, 1981) can be written as:

$$\tau = K_p e - K_d \dot{q} + g(q_d) \quad (2.2)$$

where τ represents the torque in this formula and $g(q_d)$ is the vector of gravitational torques of the system at the desired joint angles. Graphically, this control law is represented as shown in Fig. 2.2.

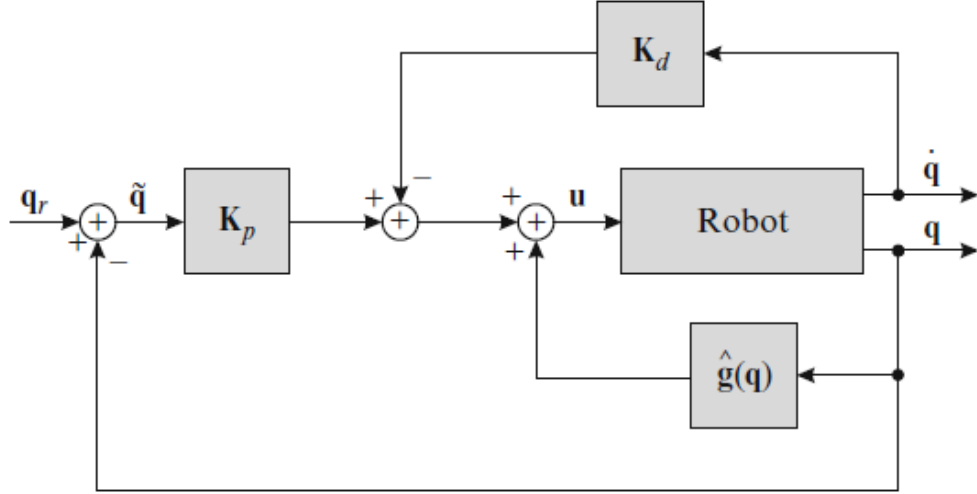


Fig. 2.1 PD Control with desired gravity compensation (Bajd, Mihelj, Lenarcic, Stanovnik, & Munih)

In PD controller with gravity compensation can drive the robot joints asymptotically to their desired values. This controller is also independent of the initial position and the velocity. Despite the simple structure of the controller, it still requires the system dynamics to calculate the gravitational torque vector. Therefore, the gravity component be a feedforward term that helps controller compensate for the dynamics.

2.1.2 NON-LINEAR PD/ PID CONTROL

The standard PD/PID control suffers in performance due to the linearity of the control law and the poor compensation for the inherent nonlinearity of the used practical dynamic systems and sensors. (Rugh, 1987) proposed a method for compensating this. The author

has designed a law by linearizing the plant about every closed-loop set point with the PID controller in shown in Fig. 2.2

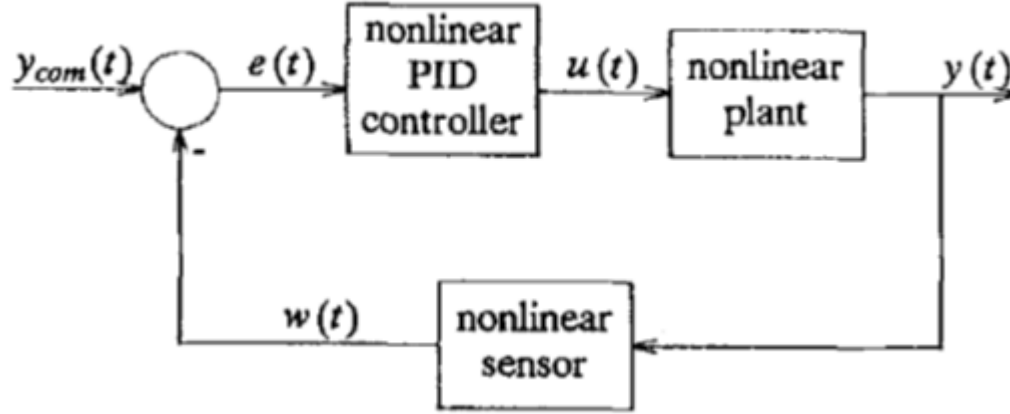


Figure 2.2 Nonlinear closed-loop system (Rugh, 1987)

The nonlinear PD (NPD) control is a control where the gains are the function of the tracking errors. Consider the general system:

$$\ddot{x} + c(\dot{x}, x) = u \quad (2.3)$$

where \dot{x} and x are system states and $c(\dot{x}, x)$ is a nonlinear, state-dependent term. According to Xu et al., (Xu, 1995) for force control of a setpoint task, we have the NPD control in the form:

$$u = Ke_f + B\dot{e}_f + f_d + \hat{c}(\dot{x}, x) \quad (2.4)$$

where K and B are gains, f_d is the desired force, $e_f = f_d - f$, and $\hat{c}(\dot{x}, x)$ is the estimate of $c(\dot{x}, x)$. The NPD's control action depends on whether the system is moving towards or away from the desired setpoint as follows (Xu, 1995)

- Moving away from setpoint: increase gains to stop the system
- Moving toward setpoint: decrease gains to minimize the residual energy at the goal

Several other previous studies were conducted involving NPD control, however, many were applied only to point-set control of linear systems (Armstrong, 2000) (Seraji). Ouyang et al. in (Ouyang, 2005) have extended NPD to a form of adaptive NPD learning control designed for repetitive tasks of robotic mechanisms.

The NPD control laws provide a method of compensation for the nonlinear system dynamics of many mechanical systems but may demand some involved stability requirements. Also, as with other forms of control with feedforward terms, knowledge of the system dynamics is required for this approach.

2.2 OTHER CONTROL METHODS

The previous section has introduced the fundamental control schemes used over most of the industrial applications. The more advanced methods of control are presented in this section, which are proven methods for increasing the efficiency and implementation of the control systems.

2.2.1 ADAPTIVE CONTROL

Based on the previously mentioned control schemes it is evident that there can always be better performance given more intimate knowledge of the plant dynamics. Most of the current control techniques in fact use methods that rely on good understanding of the plant under study in some form or the other (Sastry, 1989). However, in most industrial applications, the plant to be controlled is complex and its physical parameters are not fully

understood. In common language, "to adapt" means to change the behavior to compensate for new circumstances. Intuitively, an adaptive controller is thus a controller that can modify its behavior in response to the changes in the dynamics of the process and the character of the disturbances (Åström & Wittenmark, 1995). On the other hand, it should be noted that feedback control also attempts at reducing effects of external disturbances and plant uncertainties. Adaptive controller can be defined as a controller with adjustable parameters and a mechanism for adjusting the parameters, when needed. An adaptive controller can be visualized with two loops; one loop is the normal feedback with the process and the controller and the other is the parameter adjustment loop. It should also be noted that the parameter adjustment loop is often slower than the normal feedback loop.

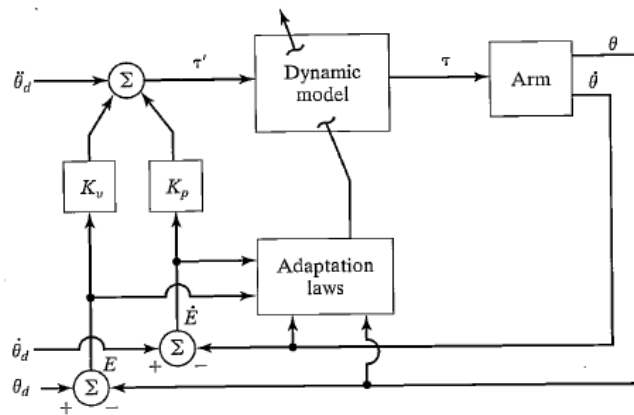


Fig. 2.3 Feedback and Adaptive Loop for Adaptive Controller (Craig, 2005)

The identification of the plant is periodic and is based on the previous estimates and new data. Therefore, the system is said to be recursive and the identification and control can be done simultaneously. It should also be noted that when we say an adaptive system identifies, it can also be used to determine whether the system is nonlinear, finite or infinite dimensional or has continuous or discrete event dynamics. (Sastry, 1989). If the type of the

system is already been identified using other analysis, then the adaptive control law will be limited to the parametric adaptive control.

There four types of adaptive systems, namely:

1. Gain scheduling
2. Model-reference adaptive control
3. Self-tuning regulators
4. Dual control

Gain scheduling can be seen as having two loops. Inner loop composing of the process and the controller and the outer loop adjusting the controller parameters based on operating conditions. Gain scheduling can be regarded as a mapping from process parameters to controller parameters as seen from Fig. 2.4 below. Although gain scheduling is one of the earliest and a one with an intuitive approach, it has a drawback of being an open-loop adaption scheme with no real intelligence. Further, depending on the system, the extent of design maybe enormous.

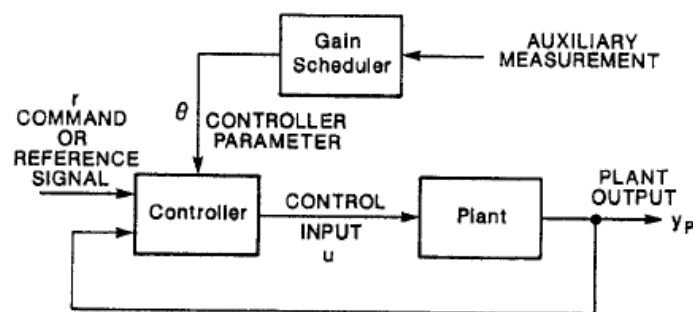


Fig. 2.4 Gain scheduling Controller (Sastry, 1989)

The model-reference adaptive system (MRAS) originally was proposed to solve a problem in which the performance specifications were given in terms of a reference model. This control scheme tells how the output should ideally respond to the command signal. A block diagram of the system is shown in Fig. 2.5, which shows the two methods of MRAC Laws, namely series and parallel, in terms of the placement of the controller in relation to the plant.

The controller can be seen to have two loops, the inner loop is an ordinary feedback for the process and controller. The adjustment of the controller parameters is done in the outer loop, in such a way that the error, which is the difference of the process output y and model output y_m is small. The major challenge of MRAS is to find the mechanism for the adjustment of the system parameters so that the system is stable. The adjustment of the parameter is done using the following mechanism

$$\frac{d\theta}{dt} = -\gamma e \frac{\delta e}{\delta \theta} \quad \text{Eqn. (2.5)}$$

In this equation, $e = y - y_m$ denotes the model error and θ is a control parameter.

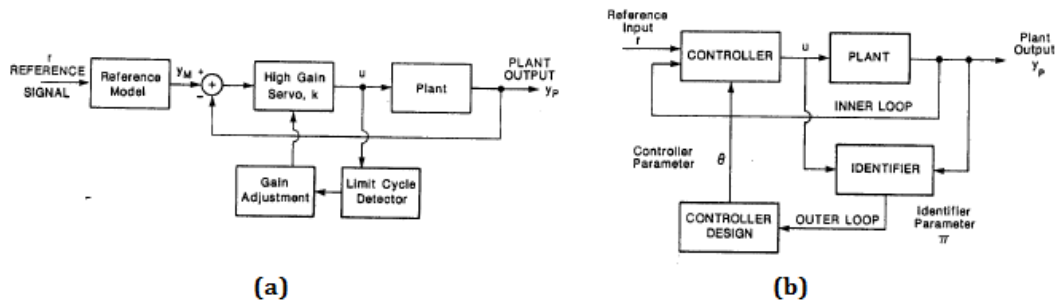


Fig. 2.5: MRAC Laws (a) Series High-Gain Scheme (b) Parallel Scheme (Sastry, 1989)

Self-tuning regulators (STR) are obtained when the estimates of the process parameters are updated, and the controller parameters are obtained from the solution of a design problem using the estimated parameters. The block diagram of STR is shown in the Fig. 2.6. The inner

loop consists of the process and an ordinary feedback controller. The parameters of the controller are adjusted in the outer loop, using recursive estimator and a design calculator.

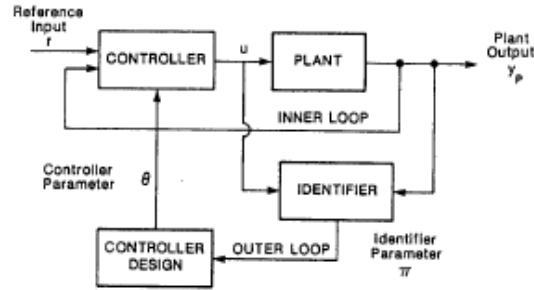


Fig. 2.6 Self-tuning Controller (Sastry, 1989)

The STR scheme is very flexible in the choice of the under-lying design and estimation methods. The controller parameters are updated indirectly via the design calculator in the self-tuner as shown in the Fig. 2.6. There is also a possibility of reparametrizing the process so that the model can be expressed in terms of the controller parameters. This simplifies the algorithm which calculates the design parameter estimates.

The adaptive control schemes discussed so far have heuristic approaches. There appears to be some limitations for example, parameter uncertainties are not considered in the design of the controller. There is a need to design adaptive controllers from some generic principles. An unknown constant can be modeled by the differential equation

$$\frac{d\theta}{dt} = 0 \quad (2.6)$$

With the initial distribution that reflects the parameter uncertainty. Parameter drift can be described by adding some random variable to the right-hand side of the Eqn. (2.6). It is to be noted that the resulting plant using Eqn. (2.6) has no distinction between these control parameters and other state variables. Thus, resulting in a controller which can handle rapid

parameter variations. The controller can be regarded as being composed of two parts: a nonlinear estimator and feedback controller. The estimator generates the conditional probability distribution of the state, called hyper-state. The feedback controller composing of a nonlinear function maps the hyper-state into the space of the control variables.

2.2.2 LEARNING CONTROL

This control scheme is like the adaptive control in the sense that it utilizes gathered past processes information to determine the future control parameters. The difference between both the control schemes is that the learning controller makes use of long-term history and saves the previous states with their appropriate responses, while the adaptive controller depends on very recent history and reacts only to the current state. (White, 1992)

The major advantage of the learning control method is that it is not necessary to have a parametric model of the system. The uncertainty in parameter estimation is compensated as the system "learns" as it propagates through the trials. The learning control has similar applications to adaptive control in the sense that it simplifies implementation of the controller and in turn improves the system performance by improving the reliability of the system.

2.3 SLIDING MODE CONTROL

Modelling inaccuracies can be due to the parameters of the system or of the unmodelled dynamics in the system. Adaptive controller can be a method of addressing these inaccuracies but sliding mode control is an alternative.

Modeling inaccuracies can be classified from a control point of view as:

- Structured (parametric) uncertainties - inaccuracies of the terms actually included in the model
- Unstructured uncertainties (unmodeled dynamics) - corresponds to the inaccuracies or the underestimations of the system order

Since modeling inaccuracies have strong adverse effects on the nonlinear control systems, there is a need to address this issue explicitly. A typical robust control consists a nominal part, similar to feedback linearization and additional terms to deal with the modelling uncertainties. Sliding control is one such robust control scheme which is based intuitively on the fact that it is easy to control a 1st order system (nonlinear or uncertain) than it is to control a general n th order system. Therefore, a simplification is introduced, which converts a n th order problem to an equivalent 1st order one. But there is a limitation; the almost performance from the transformed system comes with a cost of high control activity.

For this control scheme, a time varying sliding surface is represented using a scalar equation as (Slotine, 1991)

$$s = \left(\frac{d}{dt} + \lambda \right)^{n-1} x \quad (2.7)$$

It can be seen that this is a n th order dynamic system. Where λ is strictly positive constant. Using this definition, formulated in (Slotine, 1991), the system can be converted to a first order stabilized problem of keeping the sliding surface s at zero. It can also be seen that the equation of s contains $x^{(n-1)}$, therefore we only need to differentiate s once for the output u to appear. Furthermore, the bounds on the tracking error vector is directly related to s , which is a true measure of the tracking performance. If the dynamics of the system is exactly

known, then by solving the dynamics on the sliding mode $\dot{s} = 0$, an expression of the control force called equivalent control u_{eq} can be obtained. This control force can be interpreted as the continuous control law that would maintain perfect tracking $\dot{s} = 0$.

The advantages of sliding mode control are that its easy to implement and the robustness available for the uncertainties. (Huang, Kua, & Chang, 2008.) (Wu, Li, Zhang, & Zhu., 2014). The SMC operates on a 'sliding manifold' in the state space. Once the sliding surface has been established, the controller successful reduces and later even eliminates the parametric uncertainties and external disturbances. The SMC advantages are primarily twofold; first the accurate compensation of uncertainties for non linear systems and the second is the finite time convergence to the sliding surface (Fridman, 2012). However, the discontinuous nature of the SMC induces rise to high frequency oscillations that generate dangerous chattering or vibration (Levant, June, 2010.) (Alobaidi, March, 2013.). These vibrations can be addressed using continuous approximations of the discontinuous feedback law. An alternative to this is the use of approximations to the sign function (Kapoor & Ohri, 2013) (Abera A, January 2007). Hysteresis, saturations and sigmoids can also be used to achieve a continuous and smooth control signal.

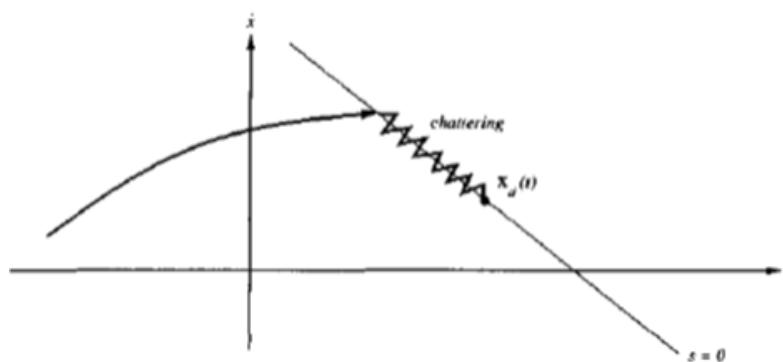


Fig. 2.7: Chattering about the sliding surface (Slotine, 1991)

2.4 TRAJECTORY PLANNING

The goal of trajectory planning is to generate the reference inputs to the motion control system which ensures that the manipulator executes the planned trajectories. The user specifies several parameters to describe the motion. Planning consists of generating a time sequence of the values attained by a polynomial function, interpolating the desired trajectory. (Siciliano, 1999). Techniques are required for trajectory generation, which is generic when the initial and final points are given (point to point motion).

Once the trajectory has been defined, a method is required to estimate how accurate the system is following the assigned trajectory using the controller. One such performance estimator for industrial manipulators is tracking error. It is defined as the result of the individual axial tracking performance of each actuator. It can also be stated as how well the controller for each axis can track the desired reference position or state variable.

Mathematically, the tracking error can be expressed as

$$E = r - p \quad (2.8)$$

Where r is the reference position or desired position and P is the actual position. Another performance estimator for robotic manipulators is contour error, which can be defined as the component that is orthogonal to the desired path or contour. Contour errors generally arise due to the difference in the loop parameters, disturbances in loads, and the complexity of the contour shape. Additional errors can arise due to mechanical backlash

and friction (Koren, 1991). Due to the limitation of scope, this project considers only tracking error as a parameter for the estimation of the system performance.

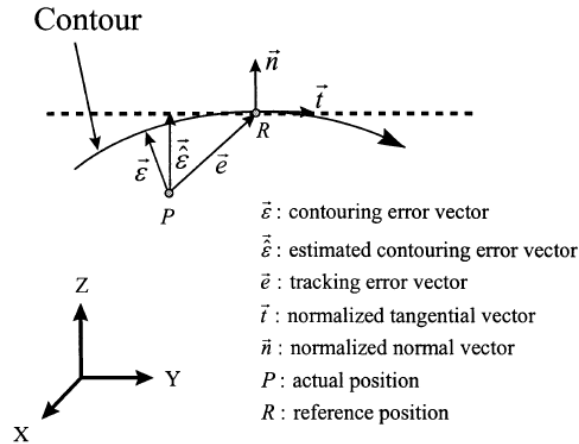


Fig. 2.8 Tracking and Contour Error (Yeh, 2003)

2.5 CONTROL OF SCARA ROBOT

Due to the many uncertainties and parameter fluctuations, there is a need for achieving robust controllers. Almost most of the industrial applications require high accuracy trajectory tracking. It is important to consider each of the uncertainties and previously mentioned imprecisions into account when implementing a control law.

One of the earliest research on the control for SCARA robot was done in (Dessaint, April 1992) and it has implemented using an adaptive controller for a SCARA robot powered by direct drive. The adaptive law consisted of the combination of PD regulator and a feedforward compensation of the full dynamics. These feedforward terms are adjusted by the adaption law so that the steady state errors are zero. The main advantages of this controller are that there isn't the need to measure the accelerations and the inverse of the mass matrix, which saves both time in computation and increases the ease of

implementation. The tracking performances using this control law is studied on a 2 DOF SCARA.

The PID approach to control a SCARA robot was satisfactory when the speed of the process was slow. For high speed applications, inertia, centrifugal and Coriolis and gravitational terms which are nonlinear play a prominent role. In (Lee, Lee, Son, Lee, & Han, February 1997) Jang M. Lee et.al. in 1997, designed a controller using a combination of PID and inverse dynamic algorithm. The tracking performance was studied over the process time of 1.4 seconds (faster process compared to 2 seconds process time used in (Dessaint, April 1992)) and it was shown that the computational power was good enough and was still room for implementation of learning algorithms. Another key attribute of this research is that gains of the PID are self tuned using an intelligent algorithm (fuzzy rules using desired system states).

In (Visioli, February 2002) the importance of trajectory tracking control and how difficult the task can be is discussed. The paper also mentions that though several innovative solutions are presented, they are not discussed from an industrial point of view (implementation problems, tuning of parameters, etc.). The third axis, which is for the vertical motion of the manipulator, is dynamically decoupled from others. Therefore, the author assumes the robot to be planar and considers only the first two links. Experiments are done to compare the tracking for decentralized (typical PID, Sliding Mode control) controllers with centralized (model-based, computed torque and neural network control methods). The author summarizes that for a large number of industrial applications, decentralized controllers' performance is sufficiently accurate.

Amir et.al. in (Moghadam , Gharib, & Moavenian) designed a robust controller for a SCARA robot using Quantitative feedback theory (QFT). The design of this controller includes the linearization of the robot dynamics and of suitable robust disturbance rejection bounds by reducing the sensitivity function. The tracking has results in good performance due to the fact there is a reduction in the cross-coupling effect between joints and the linear uncertainties in system modelling have been improved. But though the results obtained are of significant efficiency, the implementation of a QFT controller is comparatively difficult.

Owing to the increasing popularity and the complexity of motor drives for robotic manipulators and the nonlinearities in the dynamics, modelling process using mathematical representation or white box approach is complicated. Therefore, in (Zargoun, Modelling and Control of SCARA manipulator, 2013-2014), Ibrahmin et al. have modelled a control system for 4DOF SCARA using Computer-Aided Design (CAD). After the system has been designed using CAD and SolidWorks, the PID controller was later designed in MATLAB/Simulink environment. Simulations using this Simulink environment have shown that the tracking performance has less than 1% error, which for pick and place applications is widely acceptable. But the drawback is that the 4DOF model of the robot has been converted to an equivalent 2DOF system based on the decoupling between joints.

SMC (Sliding Mode Control), which will be detailly explained in the later chapters, is a highly robust and easy to implement control method. An adaptive SMC has been utilized by F. Adelhedi et al. in (Adelhedi, Jribi, Bouteraa, & Derbel, 2015) for the control of SCARA robot with 3 DOF. Owing to the quality of adaptive systems to deal with parameter uncertainties, the authors have assumed the mass of third link to be unknown. The controller was able to

deal with the physical and environmental perturbations and later represent the stabilized value of the mass at joint three. Therefore, through this research, the adaptive controller is proven to show satisfactory control stability and trajectory tracking considering parameter fluctuations.

(Sen, Bakırcioğlu, & Kalyoncu, Dec 07-09, 2017) has also made use of CAD model of SCARA robot and the position control was performed in MATLAB/Simulink. PID controller was applied for step inputs in reference trajectory, and the obtained results were satisfactory.

2.6 REMARKS

Various control algorithms were presented in this chapter, with the objective of improving tracking control of the robotic manipulators. The key advantages and limitations of each method were also discussed. It was shown that in order to achieve better tracking performance, there is a need to the priori knowledge of the system. But such a system will be difficult to implement. Therefore, advanced control schemes such as adaptive and sliding mode control were introduced.

The following chapter will formulate the control algorithm for the adaptive hybrid PD-sliding mode control law. The main purpose of this project is to improve the tracking performance of a 4 DOF SCARA variant over the conventional PD and the hybrid PD-SMC control laws while maintaining the ease of implementation.

The scope of this project is to utilize the available literature on adaptive control schemes and its implementation on SCARA robot and to design a SCARA variant in the hopes of improving its tracking performance by modifying the available control laws.

Chapter 3. MANIPULATOR DESIGN

The previous chapters have discussed the importance of the control schemes for industrial robots and the available control schemes. The previous work regarding the control of industrial SCARA were also discussed. This chapter deals with defining the 4 DOF SCARA variant, in terms of geometry, coordinate frames and robot specifications. Further the kinematic analysis will also be done to understand the relation between the various joints in the robot and the end-effector. The dynamics equations for the robot are derived which will be used for the establishing the control laws for the various controllers in the later chapters.

3.1 COORDINATE FRAMES:

To describe the geometry of the robot motion, the cartesian coordinate frame with the origin O_i and axes X_i, Y_i & Z_i for each link must be determined as shown in Fig. 3.1,

- The locations of the origin can be placed at any arbitrary locations within the geometry accordingly.
- A right-handed orthonormal coordinate frame is established, firstly Z_0, Z_1, Z_2, Z_3 & Z_4 axes are defined such that their directions represent the rotational axis.
- The X axes are defined in such a way that it lies along the common normal from Z_{i-1} axis and Z_i axis.
- Finally, Y axes are assigned to complete the right-handed triad.

The Denavit-Hartenberg parameters are the four parameters that are linked with a specific convention for attaching reference frames to the links of a spatial kinematic chain. Confirming the locations of origin and the coordinate frames, the Denavit-Hartenerg (DH) parameters have been formulated for the robot system.

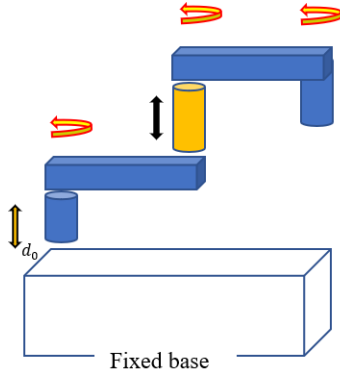


Fig. 3.1 The SCARA Variant, with directions of motion of joints

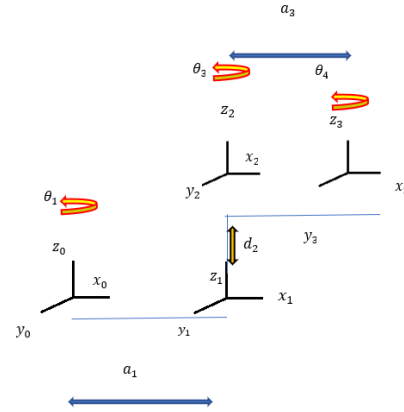


Fig. 3.2 The SCARA Variant, defining the coordinate frames and D-H Parameters

By observing the figure, it is evident that the following are the four parameters:

- a_i = distance from the intersection Z_{i-1} and X_i to origin along X_i
- θ_i = angle of rotation from X_{i-1} to X_i about Z_{i-1}
- α_i = angle of rotation from Z_{i-1} to Z_i about X_i
- d_i = distance from origin of intersection of Z_{i-1} and X_i along Z_{i-1}

By observing the geometry of the robot, the D-H Parameters are summarized as follows:

Reference Frame	Link length a_i	Twist angle α_i	Link offset d_i	Joint twist θ_i
1	L_1	0	0	θ_1
2	0	0	d_2	0
3	L_3	0	0	θ_3
3	0	0	0	θ_4

Table 3.1 D-H Parameters of the SCARA Variant System

3.2 FORWARD KINEMATICS:

Forward kinematics gives the relationship between the tool tip position and the orientation with the individual joint positions and orientation. The forward kinematics for a robot

consisting of a series of chains is found using the rigid transformation to characterize the relative movement allowed at each joint and separate rigid transformation to define the dimensions of each link.

The rigid transformation matrix is given as follows for consecutive reference frames, using the D-H parameters:

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & d_i \cos\alpha_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & d_i \sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Where $i-1$ is the reference coordinate and i is the source coordinate. A series of successive rotation and translation is required to obtain the transformation matrix of the end-effector with respect to the base reference frame.

$$T_0^4 = T_0^1 T_1^2 T_2^3 T_3^4 \quad (3.2)$$

Where, T_0^4 is the final transformation matrix of the RPRR SCARAR Variant. *MATLAB* has been utilized to get the expressions of the transformation matrices as following (*Appendix A*):

$$T_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_1 \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_1 \sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_3 \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & l_3 \sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3^4 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Therefore, using Eqn. (3.2) & (3.3), the symbolic final transformation matrix can be obtained as follows:

$$T_0^4 = \begin{bmatrix} \cos(\theta_1 + \theta_3 + \theta_4) & -\sin(\theta_1 + \theta_3 + \theta_4) & 0 & L_3 \cos(\theta_1 + \theta_3) + L_1 \cos\theta_1 \\ \sin(\theta_1 + \theta_3 + \theta_4) & \cos(\theta_1 + \theta_3 + \theta_4) & 0 & L_3 \sin(\theta_1 + \theta_3) + L_1 \sin\theta_1 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

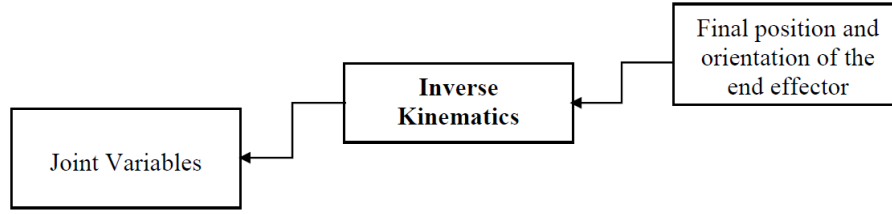
Now, in order to get the final positions of the end-effector in terms of terms of the joint variable, we consider the last column of T_0^4 matrix from Eqn. 3.4.

$$\begin{aligned} X &= L_3 \cos(\theta_1 + \theta_3) + L_1 \cos\theta_1 \\ Y &= L_3 \sin(\theta_1 + \theta_3) + L_1 \sin\theta_1 \\ Z &= d_2 + d_0 \end{aligned} \quad (3.5)$$

It is to be noted that in the Eqn. (3.5), for the expression of Z , an additional term d_0 was added. This is not present in the matrix expression of the transformation matrix but is added due to the height of the first R joint above the fixed surface. This can be observed in Fig. 3.1, therefore, d_0 is the height of the fixture on which the first joint rests.

3.3 INVERSE KINEMATICS

Inverse kinematics is the mathematical process of finding the variable parameters or the movements of the individual joints, for known end-effector details. In robotics inverse kinematics is obtained from the kinematics equations following the sequence:



Unlike the forward kinematics there is no predefined method to calculate the joint variables. The equations of inverse kinematics are unique and are obtained from basic trigonometric operations. Consider the trigonometric operations performed on the 3D space using the joint and end-effector variables shown in the Fig. 3.3.

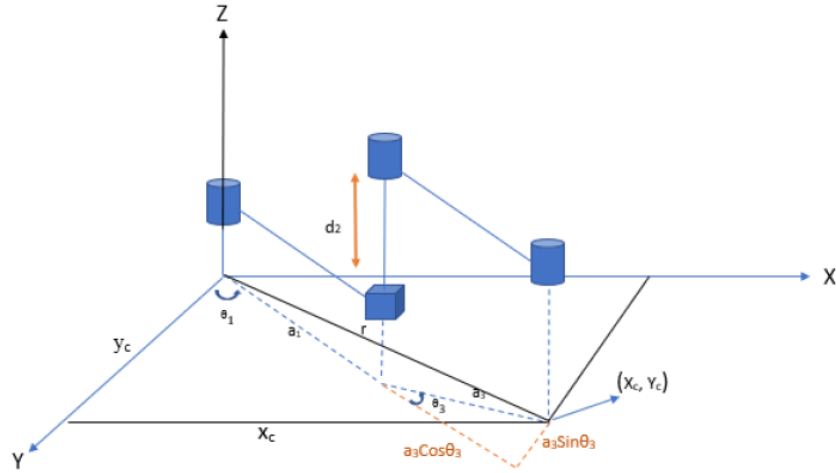


Fig. 3.3 Trigonometric method for finding the inverse kinematics

Using the representation in Fig. 3.3 and the end-effector coordinates, which are symbolically represented as (x_c, y_c, z_c) , trigonometric operations are performed to obtain expressions of individual joint variables in terms of the end-effector coordinates.

$$\cos(\theta_3) = \frac{x_c^2 + y_c^2 - L_1^2 - L_3^2}{2L_1L_3}$$

$$\sin\theta_3 = \pm \sqrt{(1 - \cos^2\theta_3)}$$

The following final expressions were obtained, which are the inverse kinematic equations:

$$\theta_3 = \arctan(\sin\theta_3, \cos\theta_3) \quad (3.6)$$

$$\theta_1 = \arctan(x_c, y_c) - \arctan(L_3 \sin\theta_3, L_1 + L_3 \cos\theta_3) \quad (3.7)$$

$$d_2 = z_c - d_0 \quad (3.8)$$

$$\theta_4 = 0 \quad (3.9)$$

It is to be noted that the fourth angle θ_4 is kept at a constant value $\theta_4 = 0$. This is done due to the limitations of the number of equations available to solve for it. The θ_4 is therefore, during the later simulations, run from 0 to 2π , for one complete revolution, for the total duration of the time. This method can be avoided by considering the pose of the robot, which the sum of the θ_1 θ_3 θ_4 & d_2 variables and applying controller for it. Hence making the system analysis full for the 4 DOF but is a tedious process and due to limitations in scope, this method hasn't been opted.

3.4 SINGULARITY ANALYSIS:

Singularity analysis is done to obtain the joint variable locations where the end-effector cannot move. These are the locations for joint variables where the end-effector doesn't respond as expected. It is crucial to locate these points as they correspond to unbounded joint velocities and torques for corresponding bounded joint velocities and torques. (Abdolmalaki, 2017).

Singularity condition can be solved by equating the determinate of the Jacobian Matrix to zero. This gives the joint variables and their locations where the singularity condition occurs.

The Jacobian matrix and corresponding equations will be found in *Appendix B*. The results obtained from solving the determinate of Jacobian to zero ($|J| = 0$) are as follows:

$$L_1 L_3 \sin \theta_3 = 0 \quad (3.10)$$

Solving Eqn. 3.10, the following is obtained.

$$L_1 = 0, L_3 = 0 \text{ \& } \theta_3 = 0, \pi \quad (3.11)$$

Studying the results obtained in Eqn. (3.11) graphically, in the 3-D plane, the singularity condition is given out in the following two cases:

- Case 1: $L_1 = 0, L_3 = 0$ corresponds to the condition where links 1 & 3 are removed.

Therefore, this eliminates the structures of links 1 & 3, as shown in Fig 3.4

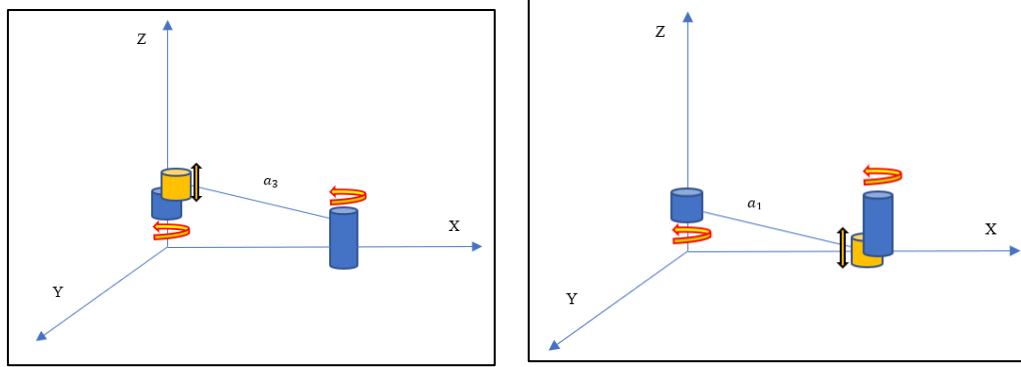


Fig. 3.4 Case 1: The lengths L_1 & L_3 are = 0, joints overlap.

- Case 2: $\theta_3 = 0, \pi$ where the arm is at full extension and overlaps to link 2 respectively as shown in Fig. 3.5

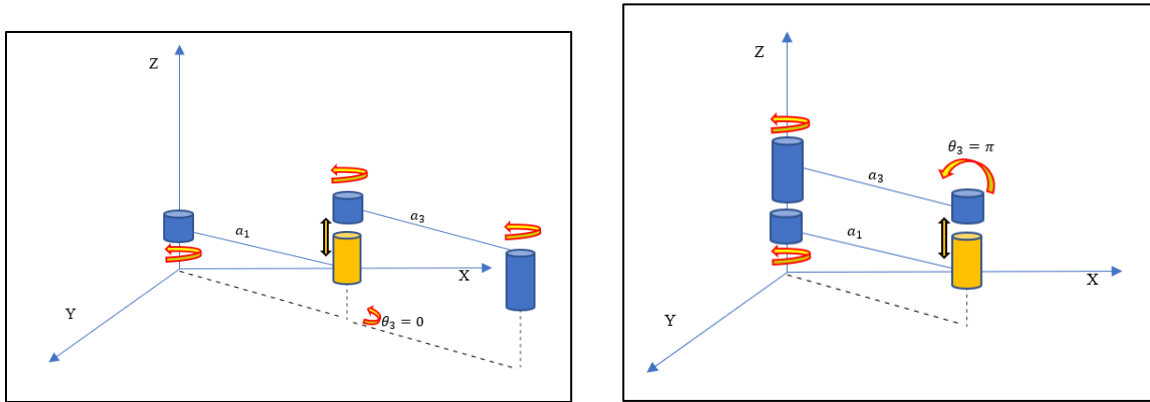


Fig. 3.5 Case 2: $\theta_3 = 0, \pi$, the arm is at full extension or overlap with link 2.

3.5 DYNAMIC ANALYSIS

Manipulator dynamics is essential for both the analysis of performance and for the design of controllers. The kinematic analysis is used to study the motion of the robot without the consideration of forces and torques. Whereas the dynamic equations explicitly describe the relationship between various robot forces and the motion (Craig, 2005). The equations of motion for the SCARA variant are discussed in this section, which is nothing but the way in which the motion of the manipulator arises from the torques applied by the actuators or from external forces applied to the manipulator.

In general, there are two methods to obtain the equations of motion; Newton-Euler formulation, and the Lagrangian Formulation. The Newton-Euler formulation is derived by the direct interpretation of Newton's second law of motion, which describes the dynamics in terms of forces and momentums. In Lagrangian framework, the dynamics are described in

terms of work and energy in the generalized coordinates. The dynamic equations for this project are derived using Lagrangian method.

The study makes use of 'Lagrange Formulation' which is the difference of the system's kinetic and potential energy as follows:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}} \right) - \left(\frac{\partial T}{\partial q_i} \right) = \tau_i \quad (3.12)$$

The Lagrange method when computed over the total system, derives the required control torque to be applied on each joint. Therefore, these Lagrangian equations of motion, which are nonlinear due to the coupling between various joint forces, provide a very powerful technique for analysing the state of the system. In general form the Lagrangian dynamic equations are written as:

$$M(\ddot{q}) + C(\dot{q}, q) + g(q) = \tau \quad (3.13)$$

where M is the mass inertia term, C is a vector of Coriolis and centrifugal terms and G is the gravitational constant. Since the degree of freedom for the present study is 4, expanding Eqn. (3.13) for four joints above Eqn. 2 becomes:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} \quad (3.14)$$

Now, the general Eqn. (3.13) & (3.14) must be expanded further to obtain the expression of τ_i in terms of M_{ij} , C_{ij} & g_i , where ij is from 1 to 4. A general method for deriving the mass inertia terms is given as follows:

Rewriting Eqn. (3.13) as:

$$\tau_i = \sum_{j=\max(i,k)}^4 M_{ik} \ddot{q}_k + \sum_{m=1}^4 C_{ik} \dot{q}_m + G_i \quad (3.15)$$

- Mass inertia term:

$$M_{ik} = \sum_{j=\max(i,k)}^4 \text{Trace}(U_{jk} I_j U_{ji}^T) \quad (3.16)$$

$$U_{ij} = \frac{\partial T_0^i}{\partial q_j} = \begin{cases} T_0^{j-1} Q_j T_{j-1}^i & \text{for } j \leq i \\ 0 & \text{for } j > i \end{cases} \quad (3.17)$$

Where T has been defined from the Eqn (3.3) & (3.4). The matrix Q is defined as:

$$Q \text{ matrix for P joint, } Q_P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and for R joint } Q_R = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.18)$$

The inertia matrix I is given by:

$$I_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (3.19)$$

- Centrifugal and Coriolis term:

$$C_{ij}(\theta, \dot{\theta}) = \sum_{k=1}^n \tau_{ijk} \dot{\theta}_k = \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i} \right) \dot{\theta}_k \quad (3.20)$$

Using Eqns. (3.15 – 3.20), the equations of motion are derived using *MATLAB, Appendix C*

and the results are obtained as follows:

- *Mass inertia matrix M =*

$$M = \begin{bmatrix} p_1 + p_2 \cos \theta_3 & 0 & p_3 + 0.5 p_2 \cos \theta_3 & p_5 \\ 0 & p_4 & 0 & 0 \\ p_3 + 0.5 p_2 \cos \theta_3 & 0 & p_3 & p_5 \\ p_5 & 0 & p_5 & p_5 \end{bmatrix} \quad (3.21)$$

Where p_1 to p_5 are given by:

$$\begin{aligned} p_1 &= \sum_1^4 I + m_1 r_1^2 + m_3 r_3^2 + L_1^2 (m_1 + m_2) + (L_1^2 + L_2^2) (m_4 + m_3) + 2(L_1 m_1 r_1 \\ &\quad + L_3 m_3 r_3) \\ p_2 &= 2[L_1 L_2 (m_3 + m_4) + L_1 m_3 r_3] \\ p_3 &= \sum_3^4 I + m_3 r_3^2 + L_3^2 (m_3 + m_4) + 2L_3 m_3 r_3 \\ p_4 &= m_2 + m_3 + m_4 \\ p_5 &= I_4 \end{aligned} \quad (3.22)$$

- *Centrifugal and Coriolis term:*

Solving the Eqn. (3.19) for the SCARA variant, the following result has been obtained for the matrix

C .

$$C = \begin{bmatrix} -\dot{\theta}_3 & 0 & -(\dot{\theta}_1 + \dot{\theta}_3) p_6 & 0 \\ 0 & 0 & 0 & 0 \\ \dot{\theta}_1 p_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.23)$$

Where,

$$p_6 = (L_3 m_3 + L_3 m_4 + m_3 r_3) L_1 \sin \theta_3 \quad (3.24)$$

- Gravity term G :

$$G = \begin{bmatrix} 0 \\ 9.81(m_2 + m_3 + m_4) \\ 0 \\ 0 \end{bmatrix} \quad (3.25)$$

Therefore, by bringing together Eqns. (3.21, 3.23 & 3.25), the general matrix form for the equations of motion is:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} p_1 + p_2 \cos \theta_3 & 0 & p_3 + 0.5 p_2 \cos \theta_3 & p_5 \\ 0 & p_4 & 0 & 0 \\ p_3 + 0.5 p_2 \cos \theta_3 & 0 & p_3 & p_5 \\ p_5 & 0 & p_5 & p_5 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \end{bmatrix} + \begin{bmatrix} -\dot{\theta}_3 p_6 & 0 & -(\dot{\theta}_1 + \dot{\theta}_3) p_6 & 0 \\ 0 & 0 & 0 & 0 \\ \dot{\theta}_1 p_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 9.81(m_2 + m_3 + m_4) \\ 0 \\ 0 \end{bmatrix} \quad (3.26)$$

Where the p_1 to p_6 are defined in Eqns. (3.22 & 3.24)

For comparison, the generic methodology described above and the *MATLAB, Appendix C* were used to derive the equations of motion for the standard SCARA robot. The following results were obtained:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} p_1 + p_2 \cos \theta_2 & p_3 + 0.5 p_2 \cos \theta_2 & 0 & p_5 \\ p_3 + 0.5 p_2 \cos \theta_2 & p_3 & 0 & p_5 \\ 0 & 0 & p_4 & 0 \\ p_5 & p_5 & 0 & p_5 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \end{bmatrix} + \begin{bmatrix} -\dot{\theta}_2 p_6 & -(\dot{\theta}_1 + \dot{\theta}_2) p_6 & 0 & 0 \\ \dot{\theta}_1 p_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 9.81(m_2 + m_3 + m_4) \\ 0 \\ 0 \end{bmatrix} \quad (3.27)$$

Where the p_1 to p_6 are defined as

$$\begin{aligned}
p_1 &= \sum_1^4 I + m_1 r_1^2 + m_3 r_3^2 + L_1^2(m_1 + m_2 + m_3 + m_4) + L_2^2(m_2 + m_3 + m_4) \\
&\quad + 2(L_1 m_1 r_1 + L_2 m_2 r_2) \\
p_2 &= 2[L_1 L_2(m_2 + m_3 + m_4) + L_1 m_3 r_3] \\
p_3 &= \sum_2^4 I + m_2 r_2^2 + L_2^2(m_2 + m_3 + m_4) + 2L_3 m_3 r_3 \\
p_4 &= m_3 + m_4 \\
p_5 &= I_4 \\
p_6 &= (L_2 m_2 + L_2 m_3 + L_2 m_4 + m_2 r_2) L_1 \sin \theta_2
\end{aligned} \tag{3.28}$$

It can be observed that the dynamics equations for both the configurations are similar with modifications in the changed axes 2 & 3. Moreover, it can also be seen that the Eqn. (3.28), the equations of p_1 to p_6 contain extra terms in mass and length of the links when compared to the similar equations of p_1 to p_6 from the SCARA variant in Eqn. (3.22 & 3.24).

3.6 REMARKS

For the modified SCARA robot, the robot parameters are defined. The kinematics analysis has been carried out to get the relation between the task space and joint space parameters. The generic methodology to define the dynamics of the system has laid out and been utilized to derive the equations of motion for the SCARA variant. To understand how the variant defers from the standard SCARA, the equations of motion has been derived using the generic equations described here for the standard SCARA as well. In comparison the equations of motion are almost similar for both the configurations, except in terms of the changed axes 2 & 3. Moreover, the presence of few extra terms in the equations for standard SCARA when compared to the variant, it can be said that the control is comparatively easier for the variant.

Using the equations of motion, stability could be analyzed using the Lyapunov theory, but due to the limitation of time and scope in this project, that part has been skipped. The control laws defined in the later sections will use these set of equations of motion.

CHAPTER 4: ADAPTIVE PD SLIDING MODE CONTROL

Once the dynamics of the system are available, the later part is the design of the control laws. For the presented dynamics in Chapter 3, the controllers are designed in this chapter. The control laws are presented for PD and PD-SMC schemes, to be used later in the simulations. Finally, the adaptive control scheme is formulated for a manipulator, the rationale and the methodology behind it are also discussed.

4.1 DYNAMIC MODEL

Control of a robotic manipulator can be achieved by the controlling the input torques to the system. Therefore, it is evident that first the dynamic equations of the system be defined to get the relationship between the torques/forces applied by the actuators and the resulting motions of the manipulator. The dynamic model for the 4DOF SCARA variant has been studied and discussed in detail in section 3.5, but we reproduce the generic dynamic Lagrangian equation of motion for a robotic manipulator, since we use it repeatedly in this chapter. The dynamics can be mathematically represented by the following nonlinear differential equation for an n -link rigid manipulator:

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q) + D(t, \dot{q}, q) = \tau(t) \quad (4.1)$$

where:

- $q = q(t)$, $\dot{q} = \dot{q}(t)$, $\ddot{q} = \ddot{q}(t)$ are functions of time and are $n \times 1$ vectors representing the joint position, velocity, and acceleration, respectively.
- $M(q)$ is an $n \times n$ inertia matrix.
- $C(\dot{q}, q)$ is an $n \times n$ matrix containing the centrifugal-Coriolis terms.

- $G(q)$ is an $n \times 1$ vector of gravitational torques or forces.
- $D(t, \dot{q}, q)$ is an $n \times 1$ vector representing friction, uncertainty, disturbance terms.
- $\tau(t)$ is an $n \times 1$ vector of the joint torques or forces.

4.2 PD FORMULATION

As discussed in earlier sections 2.1, PD control is based on local linearization and it guarantees local stability. The PD control is based only on the tracking errors of the trajectory and therefore has no feedback term. For this project, a general PD control is utilized with a gravity compensation vector to increase the robustness of the system. This vector is calculated by the system dynamics at desired joint angles. Hence, can be treated as feedforward term to assist the controller in compensating for the dynamics.

In order to formulate the PD control law, first we must define the tracking errors of the system as follows:

$$\begin{aligned} e(t) &= \theta_d(t) - \theta(t) \\ \dot{e}(t) &= \dot{\theta}_d(t) - \dot{\theta}(t) \\ \ddot{e}(t) &= \ddot{\theta}_d(t) - \ddot{\theta}(t) \end{aligned} \tag{4.2}$$

Where e , \dot{e} and \ddot{e} are the tracking errors in position, velocity and acceleration respectively.

θ_d , $\dot{\theta}_d$ and $\ddot{\theta}_d$ are the desired position, velocity and acceleration respectively.

By substituting Eqn. (4.2) in Eqn. (4.1), the dynamic model (required torque) for achieving the desired trajectory is given as:

$$\tau_d(t) = M(q)\ddot{q}_d + C(\dot{q}, q)\dot{q}_d + G(q) + D(t, \dot{q}, q) \tag{4.3}$$

Where, $\tau_d(t)$ is the desired torque input.

Therefore, from Eqn. (4.3) – Eqn. (4.1) and utilizing Eqn. (4.2), the dynamic model of the system is rewritten in terms of the tracking errors as follows:

$$\tau_d(t) - \tau(t) = M(q)\ddot{e} + C(\dot{q}, q)\dot{e} \quad (4.4)$$

It should be noted that in Eqn. (4.4), there is no feedback term. The following PD control law has been selected for the tracking control of the rigid robotic manipulators,

$$\tau_{PD}(t) = K_p e(t) + K_d \dot{e}(t) \quad (4.5)$$

Where, the K_p and K_d are positive definite matrices defining the proportional and derivative gains respectively. From Eqn. (4.5) it can be seen that the PD control is easy to implement and does not require priori knowledge of the system, whereas only requires the tracking errors of the system. Therefore, the PD control utilized is a model-free linear feedback control of the system.

4.3 PD-SMC FORMULATION

From the discussions in the earlier chapters, it can be noted that both PD control and SMC have their respective advantages and disadvantages. The advantages and weaknesses of a PD controller can be stated as follows. They are easy to implement as they are model free but on the other hand due to their linear nature have relatively low tracking performance. On the other hand, SMC has gives high tracking, but needs priori knowledge of the dynamics of the system. The exact dynamics, in some cases, may not be readily available or could be difficult to obtain. An attempt to combine the two control schemes was done in (Acob, 2011) and following control law was outlined for the hybrid PD-SMC scheme:

$$\begin{aligned}\tau_{PD-SMC}(t) &= K_p e(t) + K_d \dot{e}(t) + K_s \text{sgn}(s) \\ s &= \lambda e(t) + \dot{e}(t)\end{aligned}\tag{4.6}$$

Where s is the sliding surface of the system.

K_s is the positive definite matrix containing the SMC gains

λ is the matrix containing the positive definite vector of the slopes of the sliding surface.

and where the $\text{sgn}(s)$ is the signum function defined as follows

$$\text{sgn}(s) = \begin{cases} -1 & \text{if } s < 0 \\ 0 & \text{if } s = 0 \\ 1 & \text{if } s > 0 \end{cases}\tag{4.7}$$

This control scheme contains a nonlinear term, due to the switching term of SMC and the feedback from the PD control, Therefore the weakness of a control law has been compensated by the other. Therefore, this controller satisfies the complementary principle for the hybridization of two control systems. (Ouyang P. , 2005)

But the control scheme in Eqn. (4.6), in reality causes, the trajectory to oscillate along the sliding surface. This causes vibrations or chatter, which is undesirable. This chatter can be eliminated by using a thin boundary layer \emptyset , which ensures that outside the boundary layer, the defined control law is attractive. (Slotine, 1991). Therefore, in an attempt to reduce the chattering for the hybrid PD-SMC controller, the following modification is done to the control torque:

$$\tau_{PD-SMC}(t) = K_p e(t) + K_d \dot{e}(t) + K_s \min\left(\text{abs}\left(\frac{s}{\emptyset}\right), 1\right) \text{sgn}(s/\emptyset)\tag{4.8}$$

The introduction of the boundary layer \emptyset , to the controller gives the effect of a lowpass filter structure to the local dynamics of the sliding boundary layer. This filter like behaviour smoothens the control discontinuity inside the boundary layer and eliminates chatter.

4.4 ADAPTIVE PD-SMC FORMULATION

In order to achieve better tracking performances for the PD-SMC controller, an adaptive law has been used to estimate the dynamics of the system. An adaptive controller, as explained in section 2.2.1, has basically a parameter estimation loop, which is useful in situations where the dynamics of the system vary or are initially unknown. Many adaptive controllers have been developed for various applications (Xu, 2001) (Tomei, 1991) (Middelton, 1988). This section deals with the formulation of an adaptive PD-SMC, A-PD-SMC in short, which consists of a PD control part for bringing the trajectory to the normalized surface, SMC control part to force the trajectory close to the sliding surface and finally an adaptive control part to estimate the unknown dynamics of the system. This controller, A-PD-SMC is an extension of the previous PD-SMC law, with an addition of a parameter estimator.

For the 4DOF SCARA variant under study, the following A-PD-SMC law is proposed:

$$\tau_{A-PD-SMC} = \hat{\tau}_r + K_p e(t) + K_d \dot{e}(t) + K_s \min\left(\text{abs}\left(\frac{s}{\phi}\right), 1\right) \text{sgn}(s/\phi) \quad (4.9)$$

The A-PD-SMC is much similar to the PD-SMC law in Eqn. (4.8), expect an addition term $\hat{\tau}_r$, which is defined as the estimation of the reference or desired torque, given in Eqn. (4.3). The estimation $\hat{\tau}_r$, is defined as follows (Ouyang, Tang, Yue, & Jayasinghe, 2016):

$$\dot{\hat{\tau}}_r = \sigma s, \quad \hat{\tau}_r(0) = \hat{\tau}_{r0} \quad (4.10)$$

Where, σ is a defined as the adaptive gain (positive gain) and $\hat{\tau}_{r0}$ is the initial vector of the estimated torque vector. It should be noted that from Eqn. (4.9), the proposed A-PD-SMC law is feedback, nonlinear with no priori knowledge of the system, depending only on the tracking errors and utilized reference torques to bring the tracking to the sliding surface.

4.5 FRICTION IN JOINTS

The design and analysis of friction compensators depend heavily on the type of the friction model used and the suitability of the analysis technique employed. (H  louvy, Dupont, & Canudas de Wit, 1994). Brain Armstrong et.al. has listed a survey which describes models of machine frictions, followed with a detailed study of analysis techniques and with a survey of friction compensation methods. It has to be accepted that the modelling of friction in a system is a multifaceted phenomenon, incorporating Coulumb and Viscous friction, nonlinear friction at low velocities, temporal phenomena and the elasticity of the interface. From the reference (H  louvy, Dupont, & Canudas de Wit, 1994), the following friction matrix has been utilized for the SCARA variant under study:

$$f = - \left(gains.f_c + (gains.f_s - gains.f_c) e^{\left(\frac{\dot{\theta}}{gains.f_t} \right)^2} \right) sign\dot{\theta} + gains.b \times \dot{\theta} \quad (4.11)$$

Where, $gains.f_c$, $gains.f_s$, $gains.f_t$ & $gains.f_b$ are friction gain parameters. This section has not been detailed in further as a lot of analytical work has to be detailed in other to understand the complete friction modelling, therefore, the friction model has been borrowed.

4.6 REMARKS:

In this chapter the three control schemes to be utilized for the control of the SCARA Variant are defined. The control laws are borrowed from sources and slight modifications were done in order to reduce chatter. The PD-SMC is seen to have both the advantages of PD and SMC controllers and the resultant controller is nonlinear with a feedback loop for system

information. The defined A-PD-SMC law is an attempt to compensate for the system parameters variations and/or lack of information. The adaptive loop in these controller estimates the unknown parameters based on the reference system information given. The defined control laws are in the later chapter used for the simulations of tracking performance.

CHAPTER 5: SIMULATIONS and RESULTS

This chapter deals with a number of simulations for the trajectory tracking control of the 4DOF SCARA Variant. The performance will be tested under the three control laws; PD, PD-SMC and A-PD-SMC. Two trajectories, namely circular and linear, in 3-D space will be used. Further, a mechanism for establishing smooth motions for all the joints is also mentioned in this chapter. The SCARA Variant manipulator will be analyzed for position errors and control torques under the various control schemes. The study done in this chapter is critical as the efficiency of the control laws defined for tracking a 3-D, 4DOF robotic manipulator will be analysed.

5.1 SIMULATION SETUP

The simulations for the robotic manipulator are done over the three defined control laws and for two trajectories (linear and circular). Different control gain parameters are used for each, as the different trajectories call for different control parameters, though if such a system with no change in control parameters can be designed, it will make the implementation of the controller very easy.

The SCARA Variant robot used in this simulation is made of $n = 4$ rigid links of length L_i that are actuated about their respective joints, where the subscript i denotes the joint number. The total degree of freedom of the manipulator is therefore 4. The sequence of the joints is R-P-R-R, where R stands for a revolute joint and P stands for a prismatic joint respectively. The motion of the manipulator is in the 3-D plane, and the rigid links have their centre of

mass at a distance r_i from the preceding joint along its length. The angle of each link relative to the preceding one is denoted by q_i . Whereas, for the joint 2, the prismatic joint, the notation q_2 , is the displacement of the joint from the fixed base. The structural specifications of the SCARA Variant are shown in Table 5.1.

Link i	Mass m_i (Kg)	Length L_i (m)	Distance to Centre of Mass r_i (m)	Inertia I_i kgm^2
1	0.4	0.25	0.125	0.002
2	0.2	0.15	0.075	0.0002
3	0.2	0.25	0.125	0.002
4	0.1	0.15	0.075	0.002

Table 5.1 SCARA Robot Specifications

5.2 TRAJECTORY PLANNING FOR SIMULATIONS

The desired trajectories are input to the simulations in terms of a start and end point for each axis of the 3-D space. In order to ensure smooth trajectories for each joint, a fifth order polynomial is used to parametrically establish the position and velocity (Craig, 2005). The fifth order polynomial is given as:

$$s_c(t) = 6\left(\frac{t}{T}\right)^5 - 15\left(\frac{t}{T}\right)^4 + 10\left(\frac{t}{T}\right)^3 \quad (5.1)$$

Where t is the amount of time elapsed in the trajectory and T is the total duration of the trajectory. The sampling rate is selected as 2000 [Hz], which was enough to ensure smooth trajectories of both linear and circular contours. The position vectors of the end-effector are defined by using the fifth order polynomial in Eqn. (5.1), as follows:

- For Linear Trajectory, where the subscript 1 & 2 represent the initial coordinate and final coordinate of the line respectively:

$$\begin{aligned}
x &= x_1 + s_c(t)(x_2 - x_1) \\
y &= y_1 + s_c(t)(y_2 - y_1) \\
z &= z_1 + s_c(t)(z_2 - z_1)
\end{aligned} \tag{5.2}$$

- For circular trajectory, the position vectors are given by the subscript 0 represents the centre of the circle and R is the radius of the circle.

$$\begin{aligned}
x &= x_0 + R \cos(2\pi s_c(t)) \\
y &= x_0 + R \cos(2\pi s_c(t)) \\
z &= x_0 + R \cos(2\pi s_c(t))
\end{aligned} \tag{5.3}$$

5.3 SCARA ROBOT SIMULATION RESULTS FOR LINEAR TRAJECTORIES

Two linear trajectories are chosen to study the performance of the system under the different control laws. The Table 5.2 shows the different starting and ending points for the end-effector for the two trajectories in 3-D space.

Trajectory	X coordinate		Y coordinate		Z coordinate	
	x_1	x_2	y_1	y_2	z_1	z_2
Linear Trajectory I	-0.11	0.15	0.09	0.25	0.20	0.48
Linear Trajectory II	0.14	0.10	0.26	0.10	0.50	0.21

Table 5.2 Linear Trajectories Specifications

The graphical motion of the end-effector in 3-D for the two linear trajectories mentioned in Table 5.2 are shown below in Fig 5.1. The selection of the two linear starting and ending points were done arbitrarily.

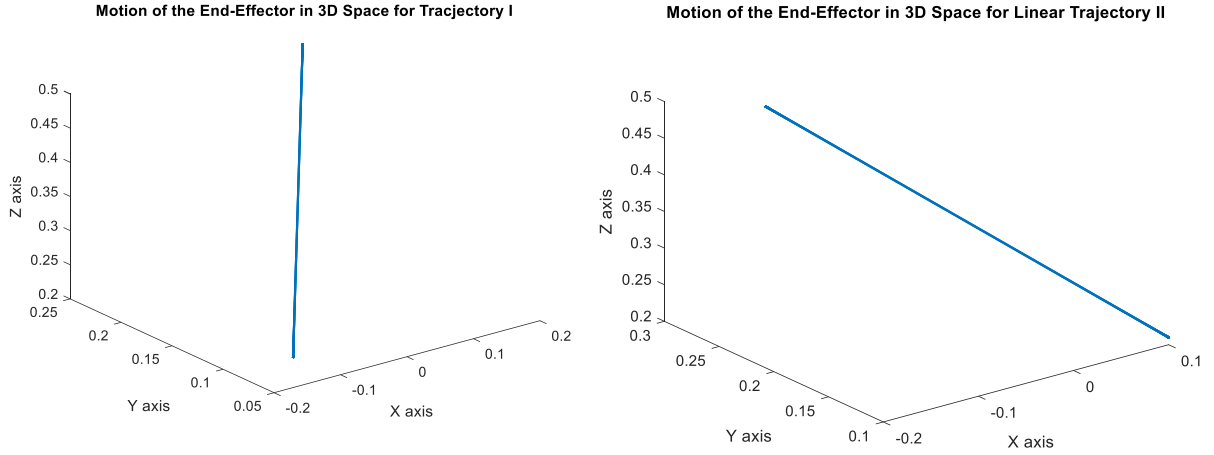


Fig. 5.1 The selected two trajectories for Linear Motion

The control parameters used for the robot for the linear trajectory simulations are listed in Table 5.3. An attempt was made to keep the control gains similar between both the linear trajectories, in order to increase the efficiency and the utility of the controllers.

Controller		
PD	PD-SMC	A-PD-SMC
$K_p = \text{diag}[1500, 2500, 1250, 750]$	$K_p = \text{diag}[1500, 2500, 1250, 750]$	$K_p = \text{diag}[1500, 2500, 1250, 750]$
$K_d = \text{diag}[1050, 1750, 875, 1225]$	$K_d = \text{diag}[1050, 1750, 875, 1225]$	$K_d = \text{diag}[1050, 1750, 875, 1225]$
	$K_s = \text{diag}[100, 75, 50, 50]$	$K_s = \text{diag}[100, 75, 50, 50]$
	$\lambda = \text{diag}[100, 100, 100]$	$\lambda = \text{diag}[100, 100, 100]$
	$\phi = 0.2$	$\phi = 0.2$
		$\sigma = 2000$

Table 5.3 Control Parameter Specifications of Linear Motion

The joints individual position and velocity based on the end-effector coordinates given are shown in Fig.5.2 & Fig.5.3 for linear trajectories *I* and *II* respectively. In these figure, the angular position and the velocity of the joints are calculated using the inverse kinematics

from the end-effector coordinates (x_c, y_c, z_c) , which is given by the set of Eqn. (5.2) and graphically in Fig. 5.2 & 5.3.

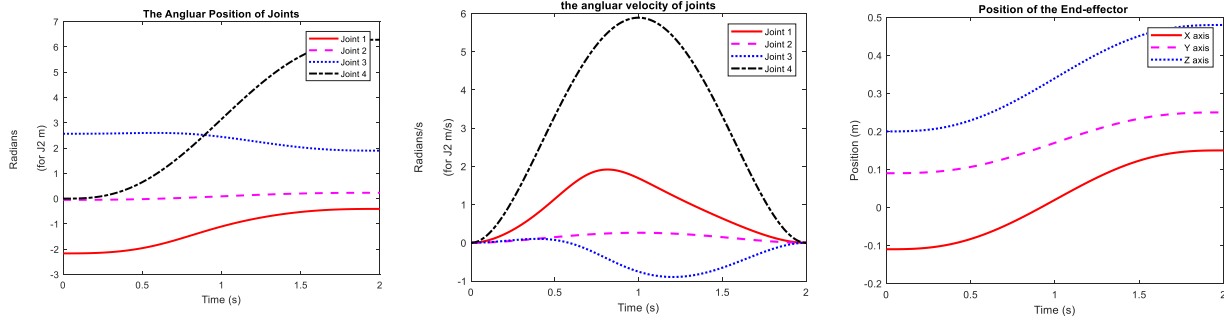


Fig. 5.2 Angular positions and velocities for joints and Position of the End-effector for Linear Trajectory I

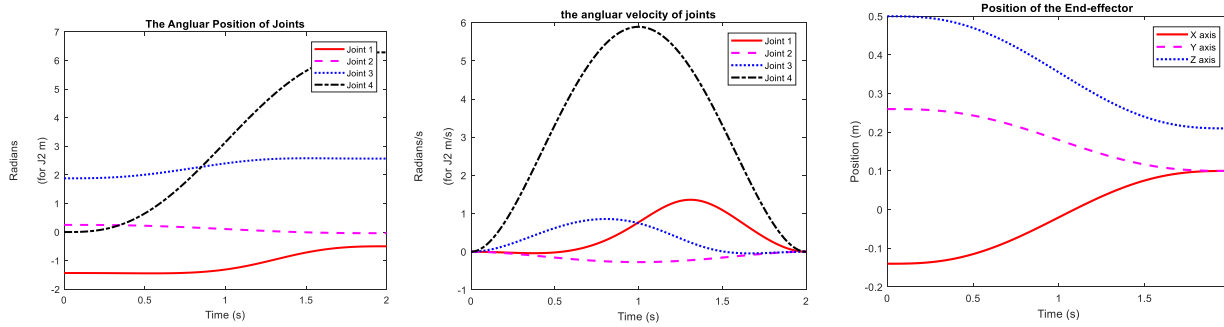


Fig. 5.3 The angular positions, angular velocities of the joints and the end-effector position for Trajectory II

5.3.1 TRACKING ERRORS OF LINEAR TRAJECTORY I:

Using the desired trajectory, the tracking errors of each axis is calculated and since, the tracking performance of the individual joints directly correspond to the performance of the end-effector. Simulations are carried out for the desired end-effector coordinates given by Linear Trajectory I. The simulations are carried using the control parameters listed in Table 5.3. The tracking errors for the four joints are compared over the three control methods: PD, PD-SMC and A-PD-SMC.

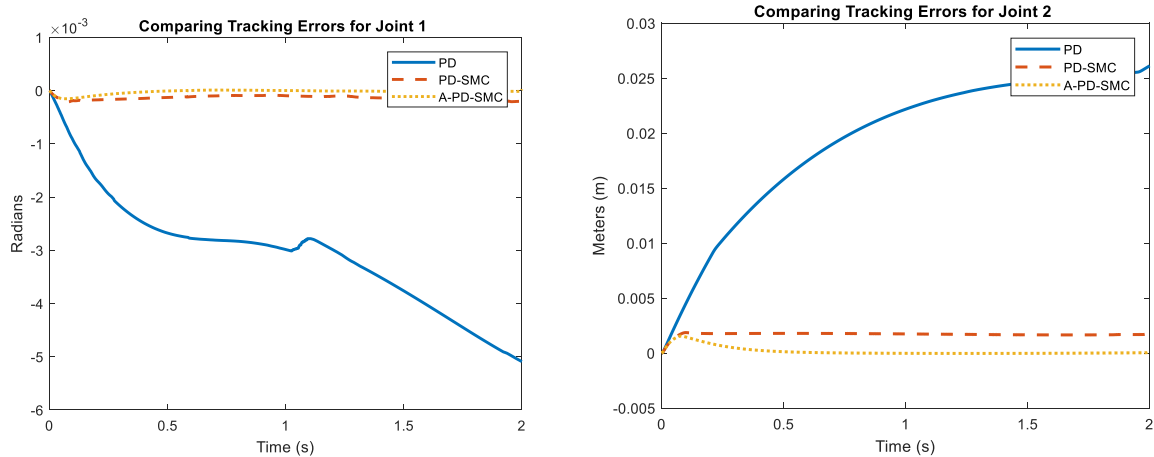


Fig. 5.4 The Tracking Errors for Joint 1 and Joint 2 between the controllers

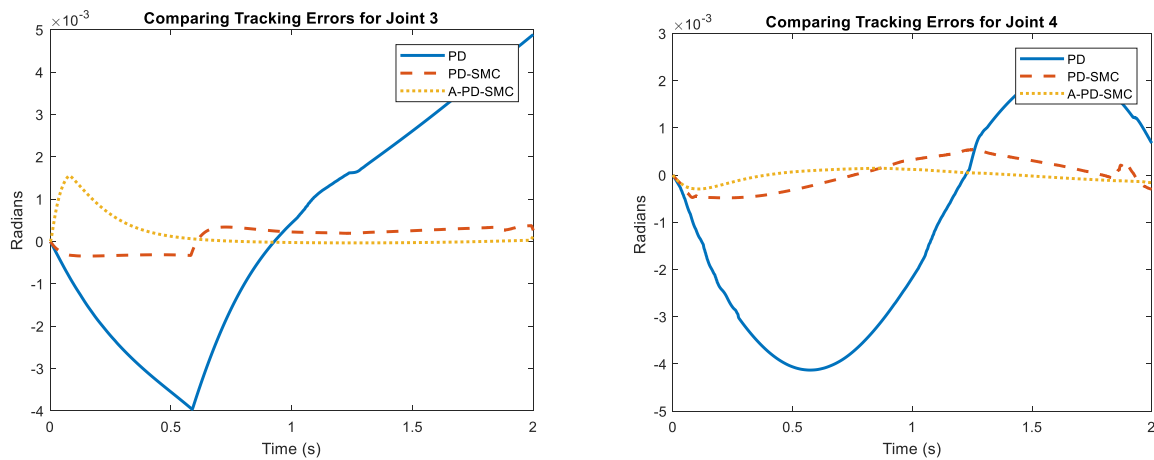


Fig. 5.5 Tracking Errors for Joint 3 and Joint 4 between the controllers

Fig. 5.4 and Fig. 5.5 depict the joint space tracking errors for each joint. It is evident from the graphs that the tracking performance is best for A-PD-SMC, since the tracking errors are the closest to zero. PD controller has the least performance as it can be seen from the above figures that the errors in joint tracking are exponentially changing. When comparing the performance for A-PD-SMC and PD-SMC, both controllers are able to track the trajectory with high accuracy, but the adaptive controller has a better edge. From the mean and standard deviation results in Table 5.4, it can be verified that both PD-SMC and A-PD-SMC controllers

achieve better tracking than the standard PD controller, since the mean of the errors of each axis are lesser to that of PD. The drawback from the set of values of mean and standard deviation is that the deviation of the errors for joint 1, is higher for adaptive than the one for hybrid. But since the mean is lesser for adaptive to hybrid by a scale of 10^{-1} , this drawback can be neglected. The average mean of the joint's errors for joints 1,3 and 4 in radians for PD, PD-SMC and A-PD-SMC are -1.206×10^{-3} , 6.68×10^{-5} , and -3.3×10^{-6} respectively.

Controller	Joint 1		Joint 2		Joint 3		Joint 4	
	Mean (rad)	S.D (rad)	Mean (m)	S.D. (m)	Mean (rad)	S.D. (rad)	Mean (rad)	S.D. (rad)
PD	-0.0030	0.0011	0.0194	0.0067	3.81×10^{-4}	0.0026	-0.0010	0.0023
PD-SMC	-1.36×10^{-4}	3.48×10^{-5}	0.0017	1.83×10^{-4}	9.16×10^{-5}	2.67×10^{-4}	2.45×10^{-4}	3.26×10^{-4}
A-PD-SMC	-2.23×10^{-5}	3.84×10^{-5}	1.67×10^{-4}	3.75×10^{-4}	2.3×10^{-5}	1.33×10^{-4}	-1.06×10^{-5}	1.2×10^{-4}

Table 5.4 Mean and Standard Deviation of Axial Tracking Errors for Linear Trajectory I

5.3.2 CONTROL TORQUES FOR LINEAR TRAJECTORY I:

The maximum input torque results listed in Table 5.5 show that for the maximum torque for PD controller for Joints 1 is zero, because all the values are in negative (with the minimum being -10.52). It can be seen that for all the joints, the maximum torques required are the highest for A-PD-SMC controller (except for Joint 2, which is less than PD-SMC by 2.04 %). This result is as expected because better tracking performance comes with the cost of high control activity. Over all the joints, the control torque is the highest for Joint 2, which is the prismatic joint. This is due to the fact that the resultant of the control of vertical motion (along Z axis) requires more energy.

Controller	τ_1	τ_2	τ_3	τ_4
PD	0	91.53	10.50	9.97
PD-SMC	10.27	93.47	13.39	7.66
A-PD-SMC	19.47	91.56	21.58	8.90

Table 5.5 Maximum input torque for the Joints

The variation of the input control torques required for each controller is depicted in Fig. 5.6. Though there is a presence of vibrations in the control torque requirements for joint 1, joint and joint 3, the A-PD-SMC controller is able to reduce the initial higher torque, quickly when compared to PD and PD-SMC controller. When looking at the control torques for joint 2, the initial torque shoots exponentially to approximately 70 kgm² and then the adaptive controller is efficient in the sense that it is able to reduce the input torque exponential within 0.5 seconds of the simulation. The initial high torque could be the reason of the initial inertial torque of the highly coupled system.

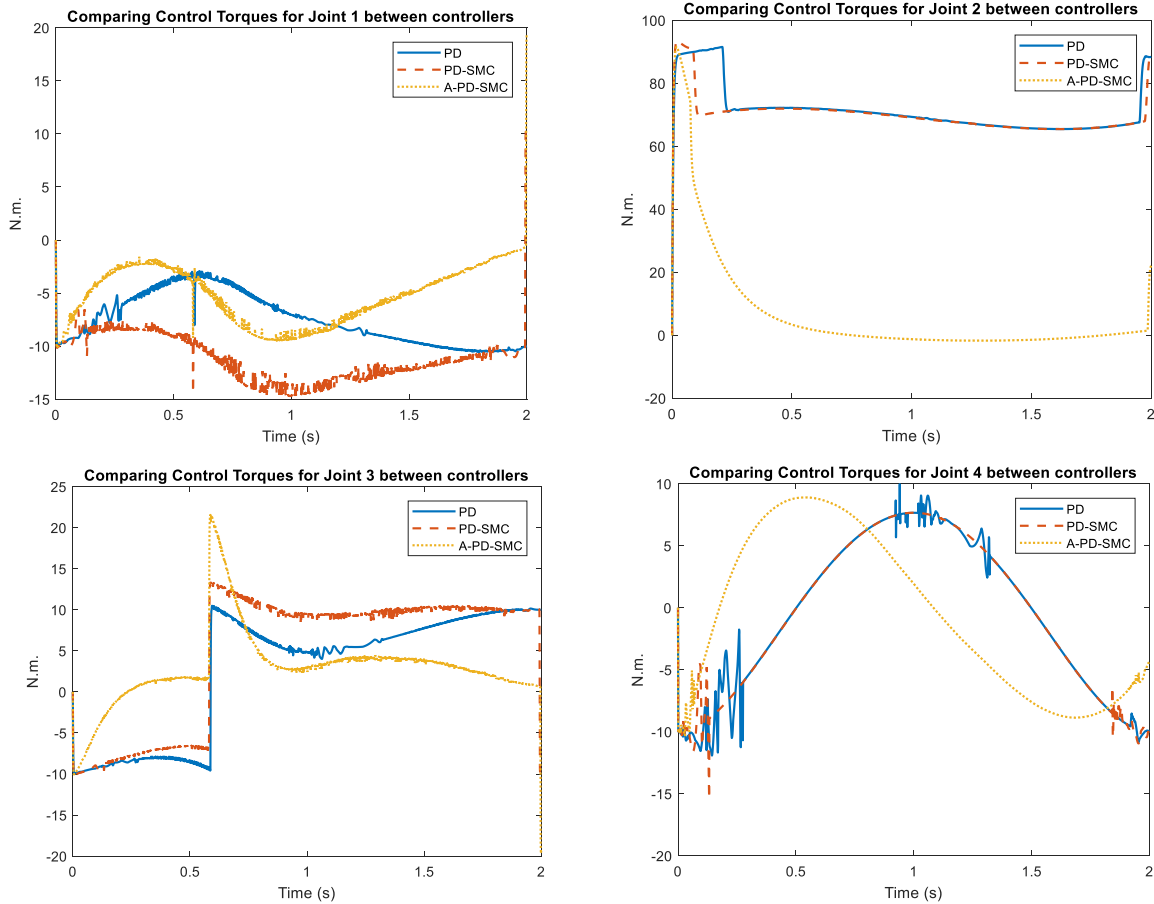


Fig. 5.6 Input control torques for Linear Trajectory I, for the joints.

5.3.3 TASK SPACE END-EFFECTOR TRACKING FOR LINEAR TRAJECTORY I:

The end-effector tracking error is also of interest in terms of performance evaluation. After the implementation of each of the control law, the end-effector coordinates are respectively calculated using the forward kinematics, from the actual individual joint space coordinates, From the results obtained from Fig. 5.7 (a), it is observed that when compared to PD controller both PD-SMC and A-PD-SMC have better end-effector tracking in all axes. The tracking for the Y axis for PD-SMC and A-PD-SMC controllers seem to be zero, which we know isn't possible in reality (perfect tracking is highly difficult). Therefore, in order to element the scaling effects in this graph, only the hybrid and adaptive controller's end-effector were plotted in Fig. 5.7 (b), and between the two, the adaptive controller provides a better end-effector tracking in the Y axis. The Z axis tracking error is most efficient for adaptive law, keeping the zero near about zero and with the least deviation, proving the efficiency of the controller in most vertical industrial applications.

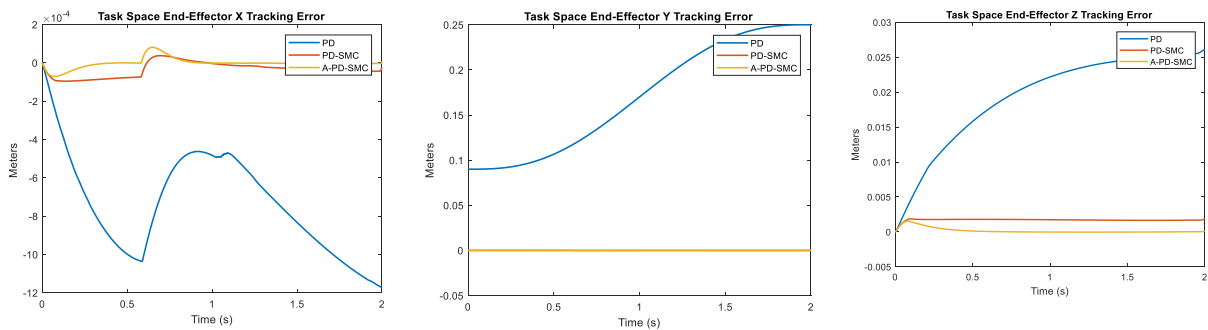


Fig. 5.7 (a) Tracking Error of the End-Effector in Task Space for Linear Trajectory I

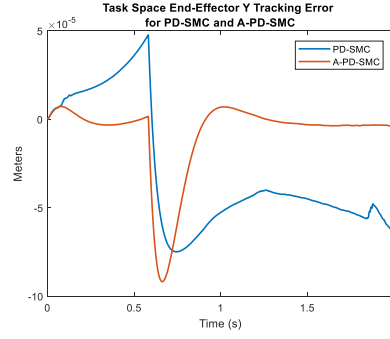


Fig. 5.7 (b) Tracking Error of the End-Effector in Y axis Task Space for Linear Trajectory I for PD-SMC & A-PD-SMC Controllers.

The below Table 5.6 gives the values of the mean and standard deviations for tracking errors in end-effector for the different controllers. The average mean of the task space error for A-PD-SMC is 1.91×10^{-5} , for PD-SMC is 5.45×10^{-4} & for PD controller it is 0.062888 over the three axes. In general, the errors and deviations are less for adaptive scheme when compared to standard PD and PD-SMC controllers, proving the adaptive scheme achieves the highest end-effector tracking performance.

Controller	x axis		y axis		z axis	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
PD	-7.34×10^{-4}	2.58×10^{-4}	0.1700	0.0603	0.0194	0.0067
PD-SMC	-3.44×10^{-5}	3.75×10^{-5}	-2.97×10^{-5}	3.54×10^{-5}	0.0017	1.83×10^{-4}
A-PD-SMC	-1.58×10^{-6}	2.62×10^{-5}	-8.10×10^{-6}	2.20×10^{-5}	1.67×10^{-4}	3.75×10^{-4}

Table 5.6 Mean and Standard Deviation for End-Effector in Task Space for Linear Trajectory I

5.3.4 TRACKING ERRORS FOR LINEAR TRAJECTORY II

The simulations are later carried for tracking performance of the described Linear Trajectory II in Table 5.2. The tracking errors for the four joints are compared over the derived three control methods: PD, PD-SMC and A-PD-SMC. The simulations are carried using the control parameters listed in Table 5.3. The results are found in Fig. 5.8 & 5.9.

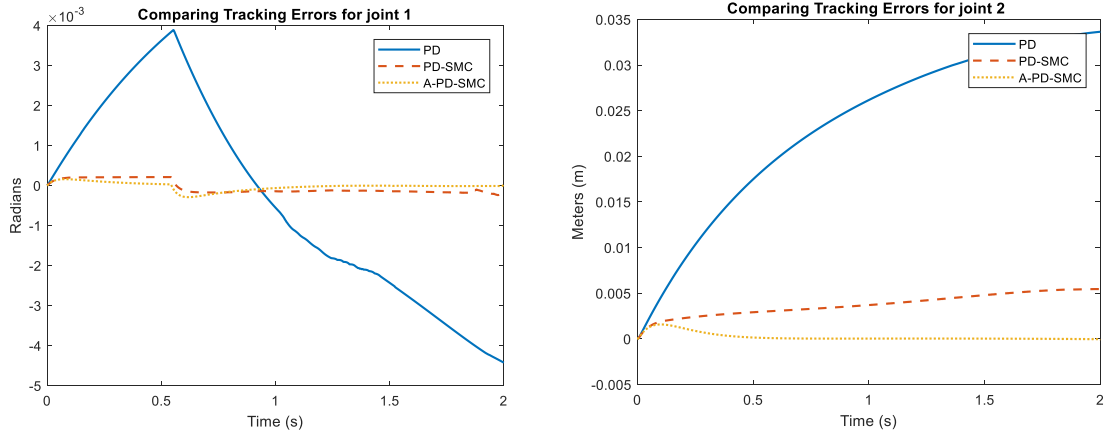


Fig. 5.8 The Tracking Errors for Joint 1 and Joint 2 between the controllers for Trajectory II

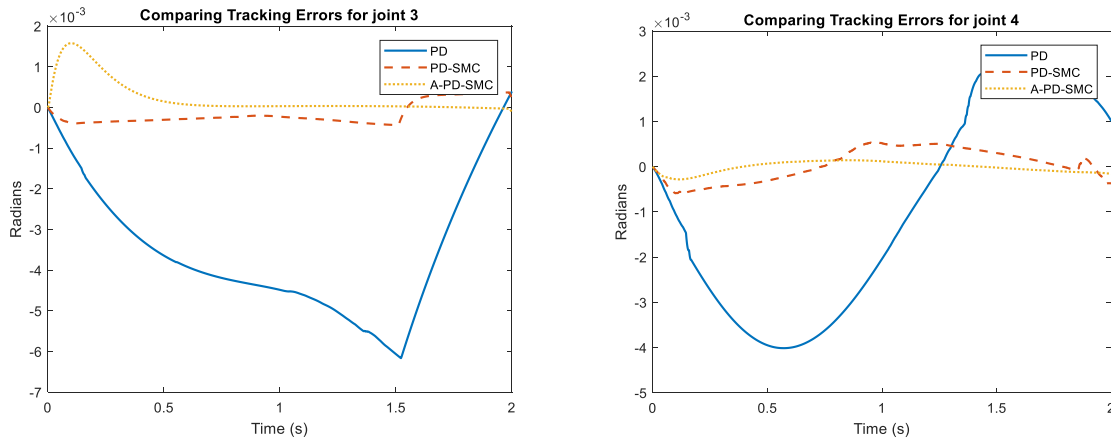


Fig. 5.9 The Tracking Errors for Joint 3 and Joint 4 between the controllers for Trajectory II

As seen from the joint tracking results for Linear Trajectory I, the performance is better for PD-SMC and A-PD-SMC when compared to the standard PD controller. From the mean and standard deviation results in Table 5.7, it can be verified that both PD-SMC and A-PD-SMC controllers achieve better tracking than the standard PD controller. When comparing both the hybrid and adaptive controller, the standard deviation for all the joints are lesser for the later. Therefore, this change in the trajectory changes improves the performance. The only joint where the adaptive controller fails to improve the tracking performance when compared to PD-SMC law is for joint 4, where the mean tracking error has changed from the positive 4.42×10^{-5} to negative -9.78×10^{-5} . The average mean of the joints errors for joints 1,

3 and 4 in radians for PD, PD-SMC and A-PD-SMC are -1.59×10^{-3} , -5.734×10^{-5} , and -3.27×10^{-5} respectively.

Controller	Joint 1		Joint 2		Joint 3		Joint 4	
	Mean (rad)	S.D (rad)	Mean (m)	S.D. (m)	Mean (rad)	S.D. (rad)	Mean (rad)	S.D. (rad)
PD	-3.822×10^{-4}	0.0025	0.0234	0.0092	-0.0035	0.0016	-8.86×10^{-4}	0.0024
PD-SMC	-5.32×10^{-5}	1.56×10^{-4}	0.0037	0.0012	-1.63×10^{-4}	2.60×10^{-4}	4.42×10^{-5}	3.56×10^{-4}
A-PD-SMC	-2.26×10^{-5}	1.01×10^{-4}	2.25×10^{-4}	4.20×10^{-4}	2.3×10^{-5}	1.43×10^{-4}	-9.78×10^{-5}	1.18×10^{-4}

Table 5.7 Mean and Standard Deviation of Axial Tracking Errors for Linear Trajectory II

5.3.5 CONTROL TORQUES FOR LINEAR TRAJECTORY II:

The results for the maximum required input control torque for tracking performance in Linear Trajectory II are tabulated in Table 5.8. The required control torques increased from PD, to PD-SMC and to A-PD-SMC, for joints 1 and 3. There is a decrease from hybrid to adaptive control torque for joint 2, which is like the trend observed for the linear trajectory I. This percentage of decrease is 2.10%. There is also a decrease from the standard PD to the hybrid control torque for Joint 4, which also follows the similar trend to Table 5.5 for Linear Trajectory I, where the percent in decrease is 15%. The control torques are the highest for joint 2, where the control action is in the vertical direction, opposing gravity, which is also like the trends in torque for Linear Trajectory I.

Controller	τ_1	τ_2	τ_3	τ_4
PD	11.344	91.50	9.96	9.11
PD-SMC	12.23	92.92	10.04	7.70
A-PD-SMC	19.87	90.96	20.70	8.92

Table 5.8 Maximum input torque for the Joints for Linear Trajectory II

The control torques required for each controller are depicted in Fig. 5.10. Similar to the control torque requirements for Linear Trajectory I, the initial torques are reduced quickly for the adaptive scheme when compared to the others. The presence of chatter is still evident

even with the change of the trajectory. This could be improved by changing the control parameters between the same trajectory shape, but such implementation is not always industrially feasible.

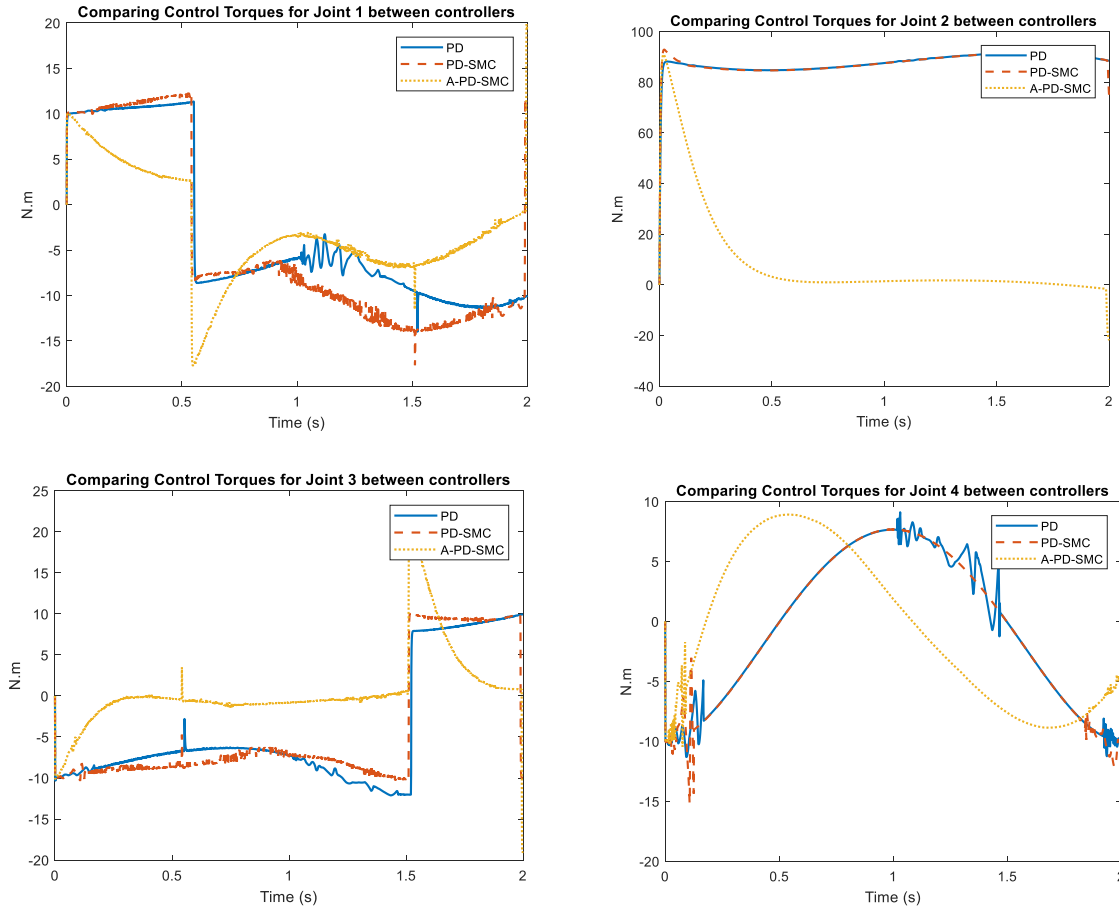


Fig. 5.10 Input control torques for Linear Trajectory II, for the joints.

5.3.6 TASK SPACE END-EFFECTOR TRACKING FOR LINEAR TRAJECTORY II:

The end-effector tracking error is also of interest in terms of performance evaluation. From the results obtained from Fig. 5.11, it is observed that the end-effector tracking for PD is overshooting, exponentially changing with time. The adaptive controller is the most efficient

and the Fig. 5.11 (b) shows the comparison between PD-SMC and A-PD-SMC tracking errors as the earlier figure, due to scaling factors, doesn't depict this relation. From Fig. 5.11 (b), between PD-SMC and A-PD-SMC, the later achieves better end-effector tracking.

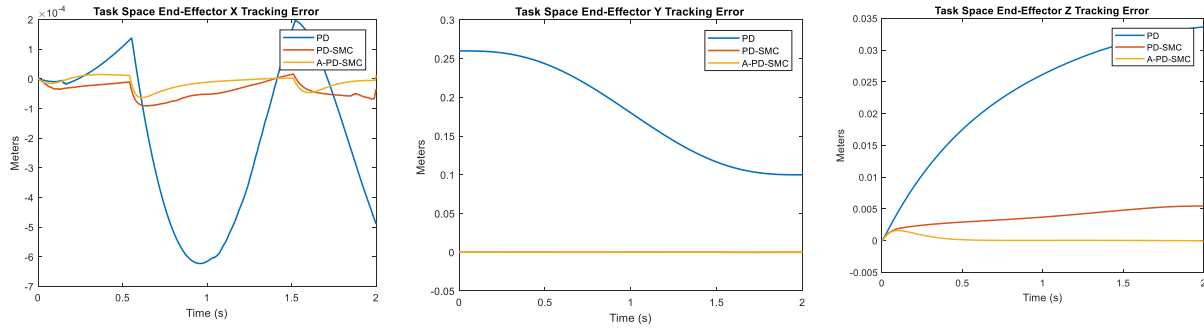


Fig. 5.11 (a) Tracking Error of the End-Effector in Task Space for Linear Trajectory II

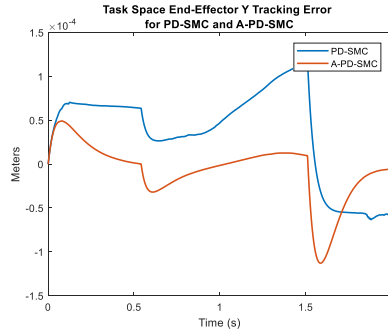


Fig. 5.11 (b) Tracking Error of the End-Effector in Y axis Task Space for Linear Trajectory II for PD-SMC & A-PD-SMC Controllers.

The Table 5.8 shows the means and standard deviations for the end-effector tracking. The average mean of the task space error for A-PD-SMC is 4.72×10^{-5} , for PD-SMC is 1.23×10^{-3} & for PD controller it is 0.06774 over the three axes. Similar results to the ones observed for linear trajectory I are observed where the adaptive scheme has proven to provide the highest tracking performance in end-effector task space.

Controller	x axis		y axis		z axis	
	Mean	S.D	Mean	S.D.	Mean	S.D.
PD	-1.76×10^{-4}	2.60×10^{-4}	0.1800	0.0603	0.0234	0.0092
PD-SMC	-3.91×10^{-5}	2.60×10^{-5}	3.60×10^{-5}	5.23×10^{-5}	0.0037	0.0012
A-PD-SMC	-1.25×10^{-5}	2.0×10^{-5}	-7.07×10^{-5}	3.30×10^{-5}	2.25×10^{-4}	4.20×10^{-4}

Table 5.9 Mean and Standard Deviation for End-Effector in Task Space

5.4 SCARA ROBOT SIMULATION RESULTS FOR CIRCULAR TRAJECTORIES

Since the linear trajectories are proven to show better tracking performance for both the joint and task space, there is a need to verify the efficiency of the adaptive controller for other trajectory shapes. For this purpose, two circular trajectories are chosen to study the performance of the system under the different control laws. The Table 5.10 shows the specifications of the circular trajectories; centres of the circular and the radius for each axis. The two trajectories in 3-D space are shown in Fig. 5.12.

Trajectory	X coordinate centre	Y coordinate centre	Z coordinate centre	Radius
Circular Trajectory I	0.12	0.12	0.12	0.2
Circular Trajectory II	0.2	0.2	0.2	0.1

Table 5.10 Circular Trajectory Specifications

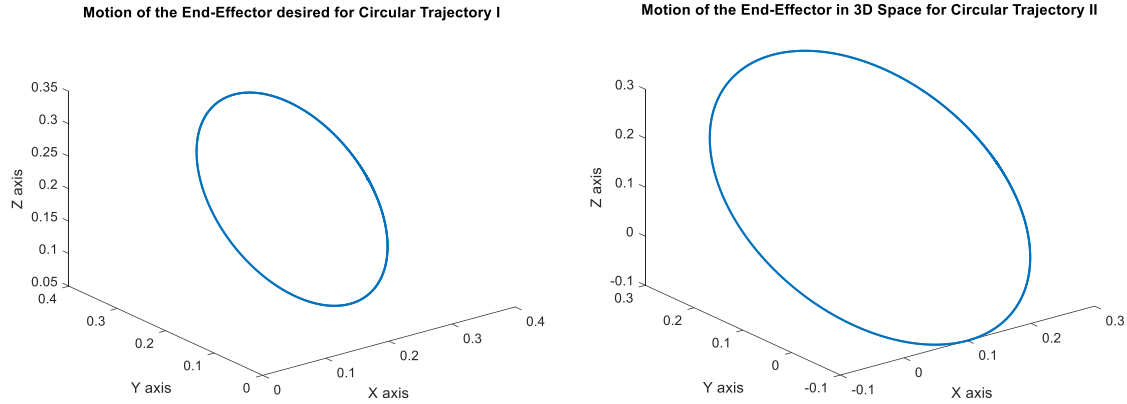


Fig. 5.12 The selected two trajectories for Circular Motion

The initial assumption was to use different control parameters between linear and circular trajectories, but after some tuning, the control parameters listed in Table 5.3 also prove to show acceptable performance for circular trajectories. This, therefore, increases the efficiency and the utility of the controllers, making it easy for industrial implementation.

For the two circular trajectories, the angular positions and angular velocities of the joints over the simulation are shown in Fig. 5.13 & 5.14 along with the position of the end-effector.

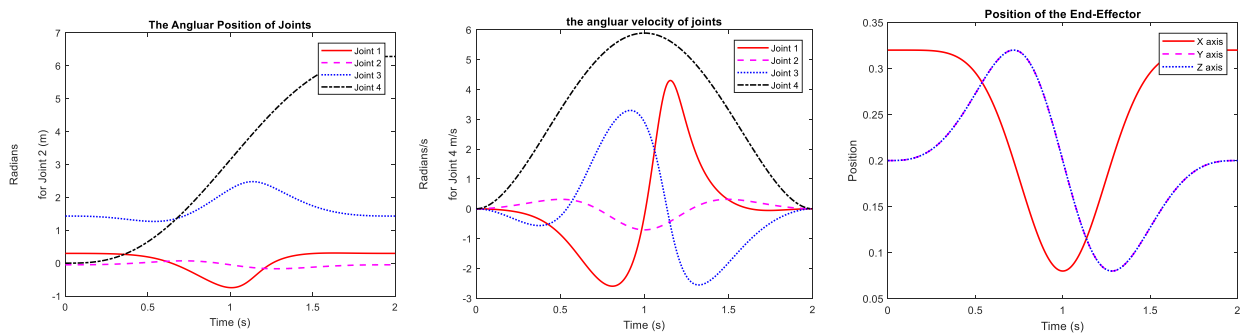


Fig. 5.13 Angular positions and velocities for joints and Position of the End-effector for Circular Trajectory I

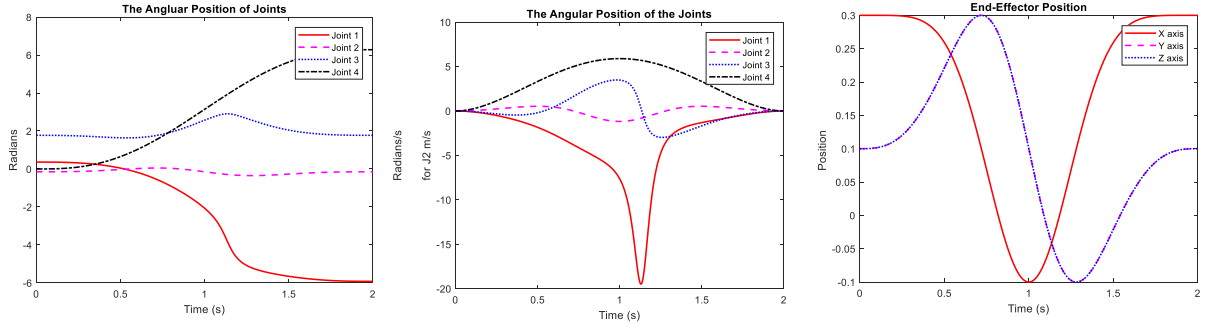


Fig. 5.14 Angular positions and velocities for joints and Position of the End-effector for Circular Trajectory II

5.4.1 TRACKING ERRORS OF CIRCULAR TRAJECTORY I:

The system is studied for the tracking performance for the end-effector coordinates in Trajectory I. The tracking errors for the four joints are compared over the derived three control methods: PD, PD-SMC and A-PD-SMC. The simulations are carried using the control parameters listed in Table 5.3.

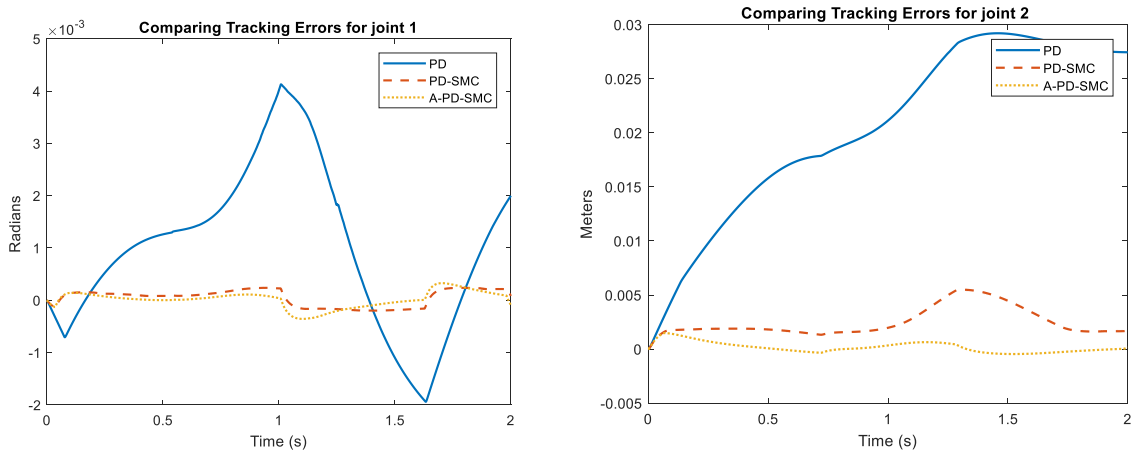


Fig. 5.15 The Tracking Errors for Joint 1 and Joint 2 between the controllers

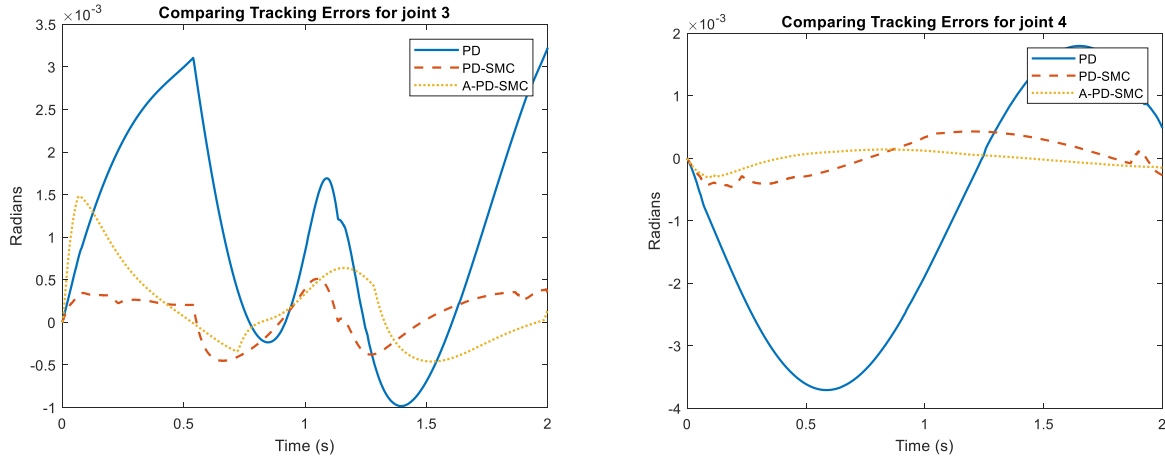


Fig. 5.16 Tracking Errors for Joint 3 and Joint 4 between the controllers

Fig. 5.15 and Fig. 5.16 depict the joint space tracking errors. It is evident from the graphs are the tracking performance is best for A-PD-SMC when compared to PD and SMC controllers. From the mean and standard deviation results in Table 5.11, it can be verified that both PD-SMC and A-PD-SMC controllers achieve better tracking than the standard PD controller. The adaptive controller outperformed the PD-SMC controller by a margin of 68% for Joint 1, 93% for joint 2, 72% for joint 3, and over 100% for joint 4. The average mean of the joints errors for joints 1, 3 and 4 in radians for PD, PD-SMC and A-PD-SMC are 3.426×10^{-3} , 5.515×10^{-5} , and 9.76×10^{-6} respectively. The standard deviation also shows the similar results with the adaptive controller having the values much closer to the mean values, when compared to the hybrid PD-SMC controller.

Controller	Joint 1		Joint 2		Joint 3		Joint 4	
	Mean (rad)	S.D (rad)	Mean (m)	S.D. (m)	Mean (rad)	S.D. (rad)	Mean (rad)	S.D. (rad)
PD	0.0010	0.0015	0.0205	0.0080	0.0010	0.0012	-9.72×10^{-4}	0.0020
PD-SMC	5.03×10^{-5}	1.58×10^{-4}	0.0025	0.0013	8.677×10^{-5}	2.70×10^{-4}	2.84×10^{-5}	2.91×10^{-4}
A-PD-SMC	1.61×10^{-5}	1.55×10^{-4}	1.63×10^{-4}	4.80×10^{-4}	2.38×10^{-5}	2.11×10^{-4}	-1.05×10^{-5}	1.19×10^{-4}

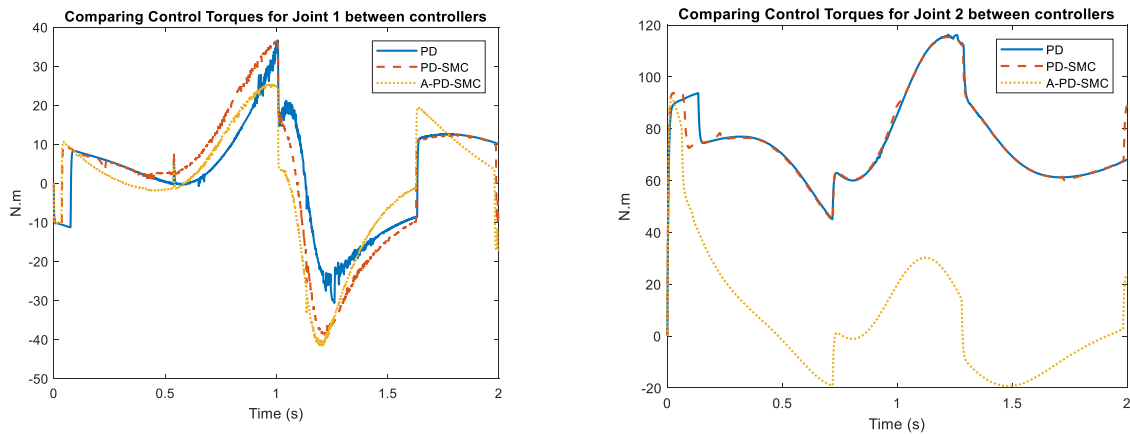
Table 5.11 Mean and Standard Deviation of Axial Tracking Errors for Circular Trajectory I

5.4.2 CONTROL TORQUES FOR CIRCULAR TRAJECTORY I:

The torque results listed in Table 5.12 show that the required control torque for the joints has reduced from PD to PD-SMC to A-PD-SMC. The required control torque has decreased from PD-SMC to A-PD-SMC by 30.2% for joint 1, 20.3% for joint 2, and by 3.1% for joint 4. There is a slight increase in this required torque by 2.9% joint 3. This, unlike the results obtained for linear trajectories, where between the hybrid and adaptive, a trend in decrease is observed for control torques. This increased tracking performance with a reduced control torque input is a characteristic observation. The reduced control torque could be due to poles placement of the system. The eigen values for the linearized system could provide insights as to which trajectories increase the performance of the system.

Controller	τ_1	τ_2	τ_3	τ_4
PD	36.63	116.331	14.53	10.56
PD-SMC	36.48	115.334	13.45	9.19
A-PD-SMC	25.46	91.92	13.84	8.90

Table 5.12 Maximum input torque for the Joints



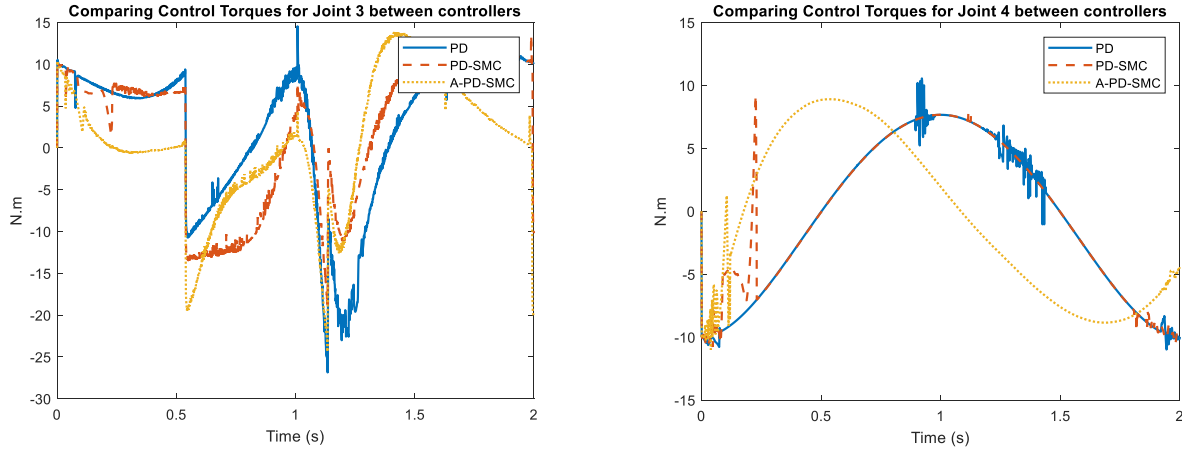


Fig. 5.17 Input control torques for Circular Trajectory I, for the joints.

Fig. 5.17 shows the plots for the control torque for the different joints. There is chatter for most region of the PD and PD-SMC controllers in Joints 1, 3, & 4. This chatter is reduced when using the adaptive controller, once again proving its efficiency. The required control torque for joint 2, follows the similar trends with the linear trajectories, where initially the torque is exponentially increased and then quickly reduced to the regions close to zero. There is a drastic decrease in the control torques for joint 3, for time $t=0.5$ s.

5.4.3 TASK SPACE END-EFFECTOR TRACKING FOR CIRCULAR TRAJECTORY I:

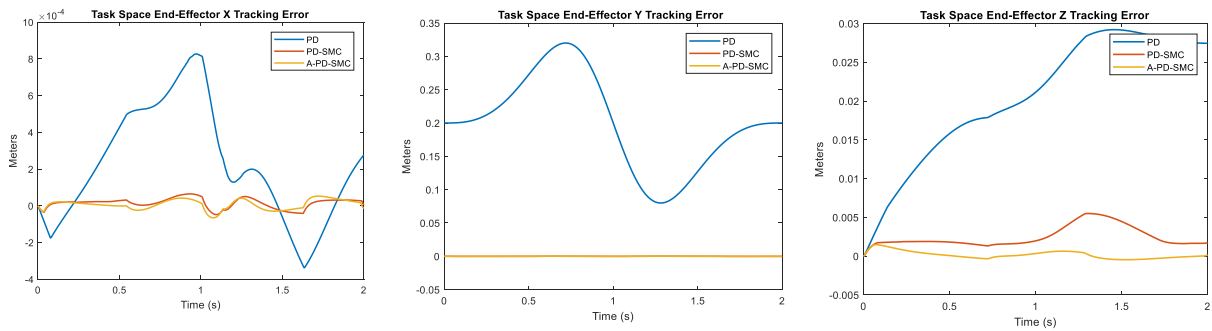


Fig. 5.18 (a) End-Effector Task Space Tracking for Circular Trajectory I

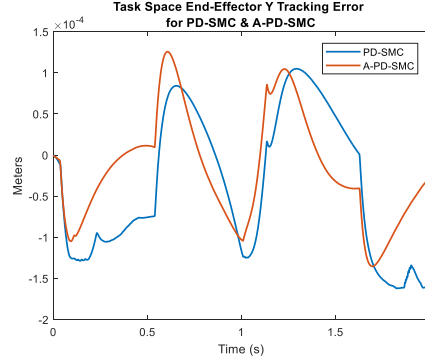


Fig. 5.18 (b) End-Effector Task Space Y Axis Tracking Error for PD-SMC & A-PD-SMC Controllers

The end-effector tracking error is also of interest in terms of performance evaluation. From the results obtained from Fig. 5.18, it is observed that when compared to PD controller both PD-SMC and A-PD-SMC have better end-effector tracking in all axes. The adaptive law can be seen to bring the Z axis tracking error to zero at the end of the simulation, since proving to the most efficient from vertical industrial applications. The average mean of the task space error for A-PD-SMC is 5.076×10^{-5} , for PD-SMC is 8.25×10^{-4} & for PD controller it is 0.07357 over the three axes.

Controller	x axis		y axis		z axis	
	Mean	S.D	Mean	S.D.	Mean	S.D.
PD	2.18×10^{-4}	3.09×10^{-4}	0.2000	0.0698	0.0205	0.0080
PD-SMC	1.47×10^{-5}	2.71×10^{-5}	-3.74×10^{-5}	8.76×10^{-5}	0.0025	0.0013
A-PD-SMC	6.80×10^{-6}	2.70×10^{-5}	-1.75×10^{-5}	6.67×10^{-5}	1.63×10^{-4}	4.80×10^{-4}

Table 5.13 Mean and Standard Deviation for End-Effector Tracking Error in Task Space for Circular Trajectory I

5.4.4 TRACKING ERRORS FOR CIRCULAR TRAJECTORY II

The simulations for tracking performance are carried for the described Circular Trajectory II. The tracking errors for the four joints are compared over the derived three control methods: PD, PD-SMC and A-PD-SMC. The simulations are carried using the control parameters listed in Table 5.3. The results are found in Fig. 5.19 and 5.20. The results from

the graphs of tracking errors are as expected, with the adaptive PD-SMC performing better than PD-SMC and the standard PD controller.

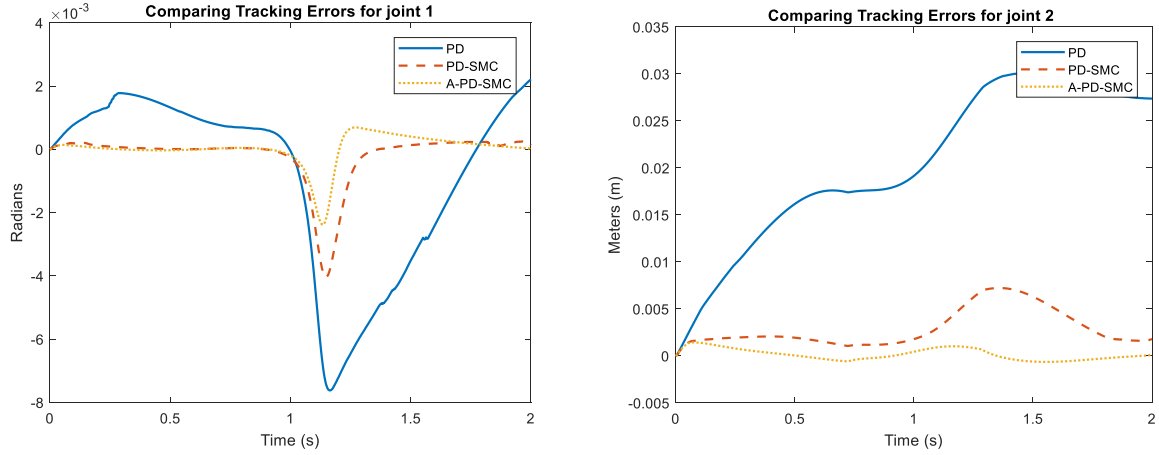


Fig. 5.19 The Tracking Errors for Joint 1 and Joint 2 between the controllers for Trajectory II

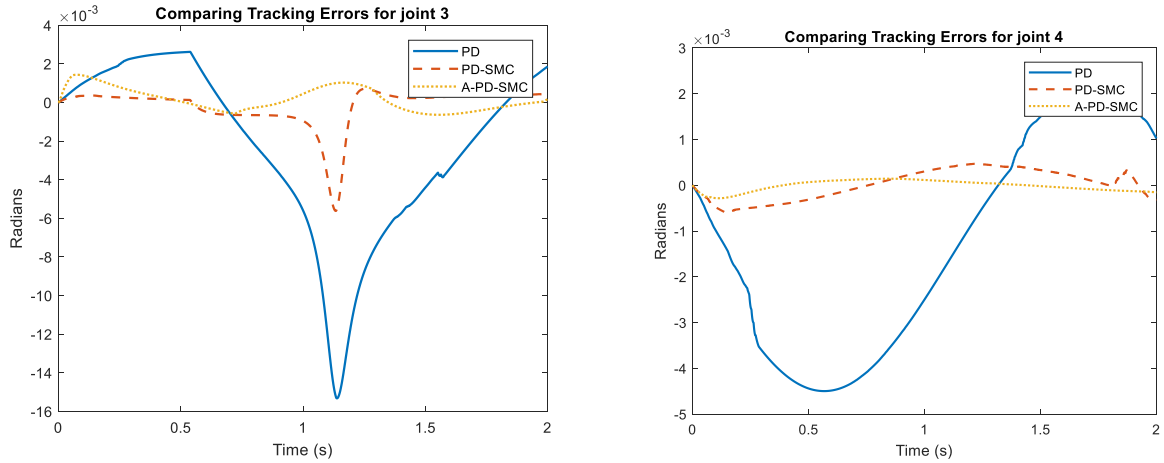


Fig. 5.20 The Tracking Errors for Joint 3 and Joint 4 between the controllers for Trajectory II

From the mean and standard deviation results in Table 5.14, it can be verified that both PD-SMC and A-PD-SMC controllers achieve better tracking than the standard PD controller. The mean and the standard deviations of the errors for the joints are slightly more for adaptive when compared to the hybrid controller. On the other hand, joint 1 for adaptive is able to achieve a mean error of zero, whereas the hybrid controller of joint 4 is able to achieve the

mean as zero. The average mean of the joint's errors for joints 1, 3 and 4 in radians for PD, PD-SMC and A-PD-SMC are -1.433×10^{-3} , -1.34×10^{-4} , and -1.566×10^{-3} respectively.

Controller	Joint 1		Joint 2		Joint 3		Joint 4	
	Mean (rad)	S.D (rad)	Mean (m)	S.D. (m)	Mean (rad)	S.D. (rad)	Mean (rad)	S.D. (rad)
PD	-0.0008	0.0028	0.0204	0.0082	-0.0023	0.0315	-0.0012	0.0059
PD-SMC	-0.0002	0.0008	0.0029	0.0020	-0.0002	0.0011	0.0000	0.0047
A-PD-SMC	0.0000	0.0277	0.0002	0.0055	-0.0045	0.0008	-0.0002	0.0001

Table 5.14 Mean and Standard Deviation of Axial Tracking Errors for Circular Trajectory II

5.4.5 CONTROL TORQUES FOR CIRCULAR TRAJECTORY II:

The torque results listed in Table 5.15 it can be seen that the pattern for the control torques is not similar to the one observed for Circular Trajectory I over all the joints. The highest control input is observed for joint 1 with the adaptive controller with a value of 182.6 N.m.. Joint 4 has the closest variations with the trajectory I with -36.13%, -14.65% and -0.05% for PD, PD-SMC & and A-PD-SMC respectively. Joint 3 has the highest variation with trajectory I, with 266.40%, 226.70% & 629.89% for PD, PD-SMC & A-PD-SMC respectively. Joint 1 control inputs have varied with trajectory I, with 138.19%, 53.86% & 617.55% for PD, PD-SMC & A-PD-SMC respectively. The Joint 2 control inputs for adaptive controllers has increased from trajectory I to trajectory II by 0.53%. Between PD & PD-SMC for Joint 1 the control torque has decreased by 35.6%, for Joint 2 by 0.11%, for Joint 3 by 17.4%, for Joint 4 increased by 3.3%. Between PD-SMC and A-PD-SMC for Joint 1 the control torque has increased by 239%, for Joint 3 by 129.8%, for Joint 4 by 10.9% and whereas for Joint 2 it has decreased by 30.69%.

The control torques required for each controller is depicted in Fig. 5.21. The presence of chatter is also significantly less for the adaptive control than the other controllers. When compared to the linear trajectory control, the initial control torques required for circular

trajectory, over all the joints are less. Moreover, apart from joint 4, all other joints require smoother control inputs when compared to the linear trajectories. The chatter present in the input torque for joint 4 using the adaptive controller is fairly in the region of time $t=0-0.5$ seconds, and is significantly smooth for the remaining time, which is unlike the other controllers.

Controller	τ_1	τ_2	τ_3	τ_4
PD	87.2520	133.5024	53.2377	7.7570
PD-SMC	56.1309	133.3524	43.9422	8.0163
A-PD-SMC	182.6885	92.4137	101.0177	8.8945

Table 5.15 Maximum input torque for the Joints for Circular Trajectory II

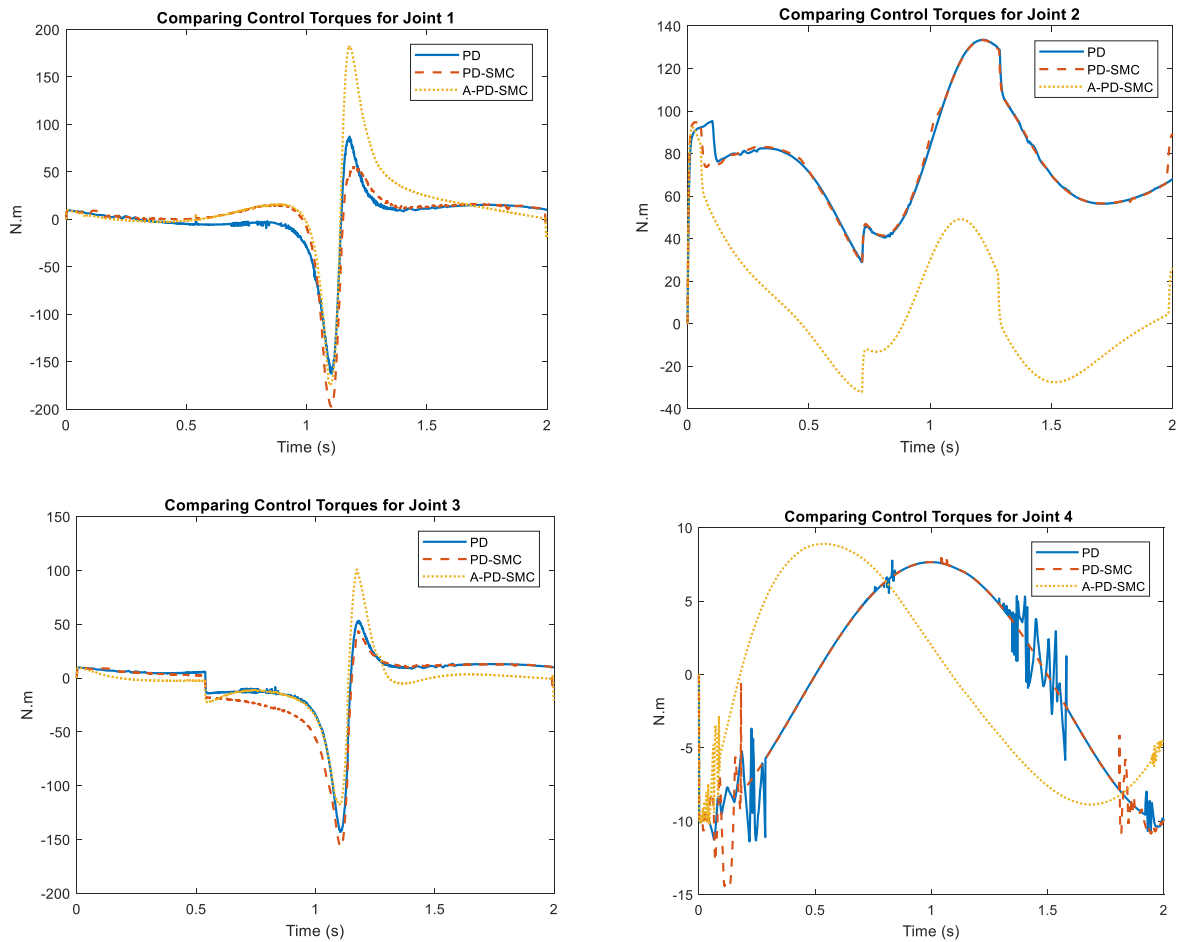


Fig. 5.21 Input control torques for Circular Trajectory II, for the joints.

5.4.6 TASK SPACE END-EFFECTOR TRACKING FOR CIRCULAR TRAJECTORY II:

From the results obtained from Fig. 5.21, it is observed that like the previous results, the end-effector tracking is the better for adaptive and the hybrid controllers when compared to the standard PD. The maximum and minimum values of the end-effector tracking error for PD-SMC are found to be 1.8525×10^{-4} and -0.0013 respectively. Similarly, for A-PD-SMC the maximum and minimum value are 7.2145×10^{-4} and -4.9238×10^{-4} respectively. Which are very less values for the tracking errors, therefore it can be said that the task space tracking in y axis is the most efficient for PD-SMC and A-PD-SMC. Between, A-PD-SMC and PD-SMC controller, but Y axis, from Fig. 5.21 (b), the adaptive controller has between performance.

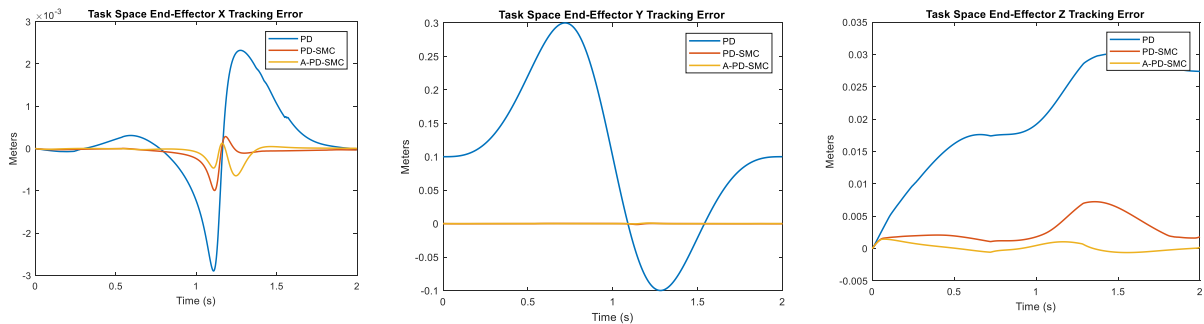


Fig. 5.22 (a) End-Effector Task Space Tracking Error for Circular Trajectory II

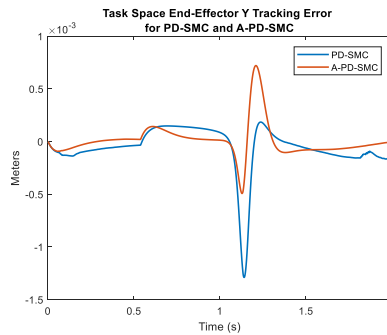


Fig. 5.22 (b) End-Effector Task Space Tracking Error for PD-SMC & A-PD-SMC Controllers

The Table. 5.16 can be finally used to prove that the system is most efficient in terms of end-effector tracking when the adaptive controller is used. A-PD-SMC over PD-SMC achieves

about 33.6%, over 100% and 94.5% better tracking end-effector tracking for x, y and z axis respectively. The average mean of the task space error for A-PD-SMC is 3.745×10^{-5} , for PD-SMC is 9.210×10^{-4} & for PD controller it is 0.04019 over the three axes.

Controller	x axis		y axis		z axis	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
PD	1.84×10^{-4}	9.91×10^{-4}	0.1000	0.1163	0.0204	0.0082
PD-SMC	-7.85×10^{-5}	1.69×10^{-4}	-5.84×10^{-5}	2.14×10^{-4}	0.0029	0.0020
A-PD-SMC	-5.21×10^{-5}	1.41×10^{-4}	7.45×10^{-6}	1.57×10^{-4}	1.57×10^{-4}	5.94×10^{-4}

Table 5.16 Mean and Standard Deviation for End-Effector in Task Space for Circular Trajectory II.

5.5 REMARKS

The simulation results show that irrespective of the trajectory shape, the system under the adaptive PD-SMC controller, achieves better tracking performance, both in joint and task space, over PD and PD-SMC controllers. It can be said that since the joint 2 is the vertical joint, working against the forces of gravity, and that is the joint which has been moved, it is in our interest to study how the joint behaves under adaptive controller. From Fig. 5.22 It can be seen that under the adaptive controller, the joint 2 tracking performance is better and is within approximately 0.7×10^{-3} m, which is good performance.

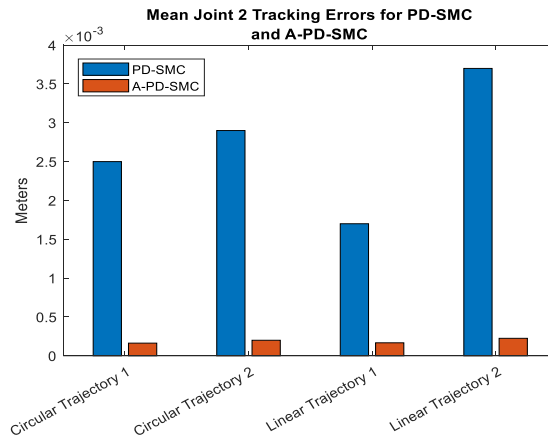


Fig. 5.23 Mean Joint 2 Tracking Error for PD-SMC & A-PD-SMC Controllers

Since, the joint space tracking performance directly effects the task space tracking performance, it can be seen from Fig. 5.23, that the mean task tracking is very less for A-PD-SMC when compared to PD-SMC for all the trajectories.

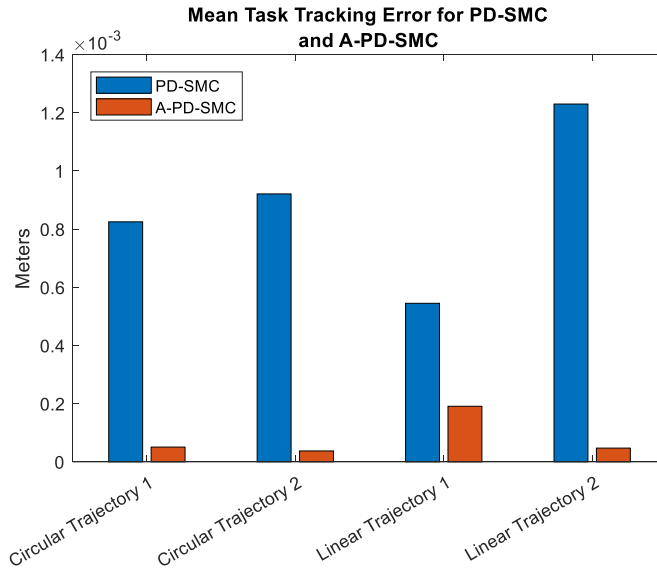


Fig. 5.24 Mean End-Effector Tracking Error for PD-SMC and A-PD-SMC

The control inputs are the highest of the joint 2, mostly over all trajectories, because the linear motion of this joint is in the vertical direction. The tracking error of the end effector in the Y axis is very negatable, therefore, the SCARA variant under the influence of the proposed adaptive control achieves almost perfect y axis tracking. Without any change of the control parameters, good tracking is observed for both linear and circular trajectories, making the system robust and easy to implement. By comparing the obtained curves for the control input requirements over all the trajectories, it has been observed that the chatter in the curves is comparatively less for adaptive controller and that the required control input curves are smoother for circular trajectories when compared to linear. When the average of the control input torques for all the joints are compared for each of the trajectories, in Fig. 5.24, it can be seen that the adaptive controllers require the highest input. Circular trajectory

I , is able to achieve higher tracking performance using the adaptive controller with comparatively lesser inputs than the corresponding PD & PD-SMC controllers. Therefore, if the coordinates of the circular trajectory II are used, then the required input are less to get higher tracking. When comparing circular and linear trajectories, the later requires comparatively lesser control inputs than the earlier, making linear trajectories tracking less power consuming.

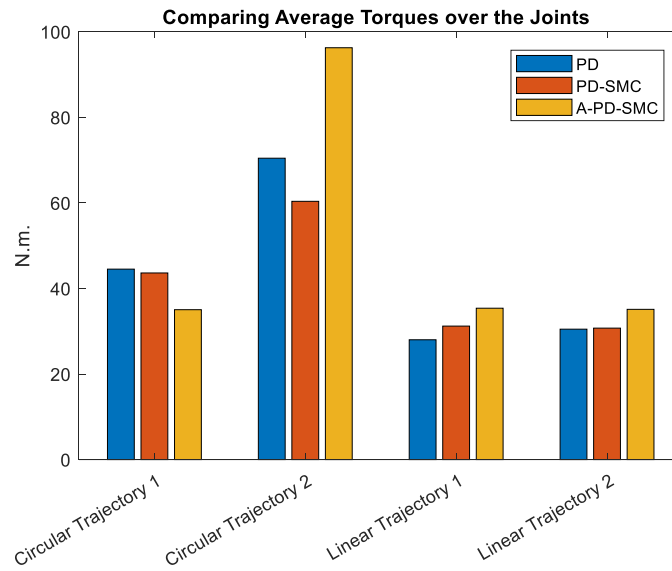


Fig. 5.25 Average Torques for the joints over the various trajectories.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 GENERAL REVIEW

In this project, a modification has been performed for the axis two of the standard SCARA robotic manipulator. The standard SCARA robotic manipulator has a revolute joint for Joint 2 and a prismatic joint for Joint 3. The axis 2 (joint 2) of the modified SCARA is now a prismatic joint, whereas the axis 3 (joint 3) is the revolute joint. The kinematics and dynamics were analysed for the SCARA variant were studied.

In this project, the existing control schemes for adaptive PD-SMC and hybrid PD-SMC were modified, incorporating the boundary layer restrictions, in order to increase the tracking performance for the SCARA variant. Simulations were performed using two trajectory shapes; linear and circular and two coordinates for each. Meaning a total for four trajectories were simulated to analyse the performance of the controller.

The proposed PD-SMC and A-PD-SMC were able to achieve much higher performance than the standard PD controller. Relative to PD-SMC, the A-PD-SMC achieved higher performance in both joint and task space. The adaptive controller, for all the simulations are provided higher end-effector tracking to the desired trajectories, when compared to PD and PD-SMC controllers. The Y and Z axis are able to achieve almost perfect tracking using A-PD-SMC. Between the two linear trajectories, the control torques have little variation, whereas the variation between the control torques for the two circular trajectories is more. Circular

trajectories on average require more control inputs to achieve better tracking than linear trajectories.

The tuning of the control parameters has led to smoother control inputs requirements for the joints, though some chatter is still evident. The selection of the control parameters has been such that for both the trajectories, the given satisfactory performance. This is an important characteristic as, under the same system, good tracking performance can be achieved for either linear or circular trajectory.

6.2 MAIN CONTRIBUTIONS

The following are the main contributions pertaining to the work done in this project:

- An alternative to the Standard SCARA Robot was proposed, with a vertical Z axis Prismatic joint in axis 2.
- A model-free adaptive PD-SMC controller was proposed with control parameters which can be tuned easily for acceptance results in terms of tracking and control inputs requirements.
- The control system is robust in the sense that it is fast and is easily implemented because there no need to change the set control parameters between linear and circular trajectories.

6.3 FUTURE WORK

The work done in this report shows that the adaptive PD-SMC control law can provide a robust, model-free and intuitive tracking performance over PD and PD-SMC controllers. The future scope of the project can be carried out to study the contour performance of the controllers. Work can also be done to demonstrate how the modified SCARA robot functions in correspondence to the standard SCARA robot using the defined control laws. Tracking and contour performance can also be explored for other trajectories shapes such as arbitrary and non-linear. Finally, experimental results can be obtained to validate the application of the proposed SCARA variant and the proposed controller.

List of Appendices

SINGULARITY ANALYSIS:

For revolute joints $J_i = \begin{pmatrix} Z_{i-1} \times (O_n - O_{i-1}) \\ Z_{i-1} \end{pmatrix}$

For prismatic joints $J_i = \begin{pmatrix} Z_{i-1} \\ 0 \end{pmatrix}$

Where Z_i is the corresponding T_0^i matrix's third column.

and O_i is the corresponding T_0^i matrix's fourth column.

The MATLAB code can be found in *Appendix B* for the symbolic evaluation of the Jacobian.

Appendix A Finding the Transformation Matrices

```
syms d2 t1 t3 t4 L1 L3
RPRR_T_01 = Transformation_matrix( 0, t1, L1, 0);
RPRR_T_12 = Transformation_matrix( d2, 0, 0, 0);
RPRR_T_23 = Transformation_matrix( 0, t3, L3, 0);
RPRR_T_34 = Transformation_matrix(0, t4, 0, 0);
RPRR_T_04 = RPRR_T_01*RPRR_T_12*RPRR_T_23*RPRR_T_34;
function T = Transformation_matrix(d,t,r,a)
T =[cos(t) -cos(a)*sin(t) sin(a)*sin(t) r*cos(t) ;
sin(t) cos(a)*cos(t) -sin(a)*cos(t) r*sin(t) ;
0 sin(a) cos(a) d ;
0 0 0 1 ];
End
```

Published by MATLAB® R2017B

Appendix B Singularity Analysis:

```
Z_0=[0 0 1]';
Z_1=RPRR_T_01(1:3,3);
Z_2=RPRR_T_02(1:3,3);
Z_3=RPRR_T_03(1:3,3);
%*****
*****
% O matrices are defined as per the definition
```



```

O_0=[0 0 0]';
O_1=RPRR_T_01(1:3,4);
O_2=RPRR_T_02(1:3,4);
O_3=RPRR_T_03(1:3,4);
O_4=RPRR_T_04(1:3,4);
%*****
*****
J1=[cross(Z_0,(O_4-O_0));Z_0]; % R joint
J2=[Z_1;zeros(3,1)]; % P joint
J3=[cross(Z_2,(O_4-O_2));Z_2]; % R joint
J4=[cross(Z_3,(O_4-O_3));Z_3]; % R joint
%*****
*****
Jacobian=[J1 J2 J3 J4];
Jacobian =simplify([Jacobian(1:3,:);Jacobian(6,:)]);
simplify(det(Jacobian));
Published by MATLAB® R2017B

```

Appendix C Dynamic Equations:

Defining U Matrices:

```

% Defining U Matrices
%*****
***** %***First column for R joint

U{1,1} = Q_REVOLUTE() * RPRR_T_01;
U{2,1} = Q_REVOLUTE() * RPRR_T_02;
U{3,1} = Q_REVOLUTE() * RPRR_T_03;
U{4,1} = Q_REVOLUTE() * RPRR_T_04;

%***Second Column for P joint
U{1,2} = zeros(4,4);
U{2,2} = RPRR_T_01 * Q_PRISMATIC() * RPRR_T_12;
U{3,2} = RPRR_T_01 * Q_PRISMATIC() * RPRR_T_12*RPRR_T_23;
U{4,2} = RPRR_T_01 * Q_PRISMATIC() *
RPRR_T_12*RPRR_T_23*RPRR_T_34;

%***Third Column for R joint
U{1,3} = zeros(4,4);
U{2,3} = zeros(4,4);
U{3,3} = RPRR_T_02 * Q_REVOLUTE() * RPRR_T_23;
U{4,3} = RPRR_T_02 * Q_REVOLUTE() * RPRR_T_23*RPRR_T_34;

%***Fourth Column for R joint
U{1,4} = zeros(4,4);
U{2,4} = zeros(4,4);

```

```

U{3,4} = zeros(4,4);
U{4,4} = RPRR_T_03 * Q_REVOLUTE() * RPRR_T_34;

%***** Q matrix for Revolute joints
%*****

function Q_RJ = Q_REVOLUTE()
Q_RJ = [ 0 -1 0 0
1 0 0 0
0 0 0 0
0 0 0 0];
end
%*****
%*****

%***** Q matrix for Prismatic joints
%*****

function Q_PJ = Q_PRISMATIC()
Q_PJ = [ 0 0 0 0
0 0 0 0
0 0 0 1
0 0 0 0];
end
%*****
%*****

```

For M Matrix

```

% Defining M matrix

for i = 1:4
for k = 1:4
j = max(i,k);
M = 0;
for j = 1:4
M = M + trace(U{j,k}*I{j}*U{j,i}.');
end
Mass_Matrix(i,k) = M;
end
end

```

For C Matrix

```

syms t1_dot d2_dot t3_dot t4_dot
q = [t1; d2; t3; t4];

```

```

q_dot = [t1_dot; d2_dot; t3_dot; t4_dot];

for i = 1:2
    for j = 1:4
        C = 0;
        for k = 1:4
            C = C + ((1/2)*((diff(M(i,j),q(k)) +
diff(M(i,k),q(j)) - diff(M(k,j),q(i))))*q_dot(k));
        end
        C_Matrix(i,j) = C;
    end
end
end

```

Published by MATLAB® R2017B

MATLAB FUNCTION FOR CONTROL

The MATLAB function used for the control analysis for the system are presented here. For similarity purposes, only the files for A-PD-SMC are depicted for once linear trajectory.

Declaring Global Variable

```
global robot gains tdom pdom N friction torque M C G
```

Declaring Robot Parameters

```

robot.mass    = [0.2 0.1 0.2 0.1]*20;
robot.l       = [0.25 0.15 0.25 0.15]; %length of the link
robot.r       = robot.l/2;             %centre of mass distance
robot.I       = [01 0.1 01 0.1]*0.002;
robot.tormax  = 8;

```

Declaring Sampling and Time definition

```

N          = 2000;    %sampling rate in Hz
tf         = 2;       %total time in seconds
tdom.dt    = tf/(N-1);
tspan      = 0:tdom.dt:tf;

```

Initializing Control Torques

```
torque.td1 = zeros(N,4);
torque.td2 = zeros(N,4);
torque.td3 = zeros(N,4);
torque.td4 = zeros(N,4);
```

Defining Control Gains

```
gains.kp    = [6,10,5,3]'*250; %proportional gain
gains.kd    = [6,10,5,7]'*175; %derivative gain
gains.ki    = [0 0 0 0]';      %integral gain
gains.K     = [20 15 10 10]'*5; %smc control gain.
gains.l     = [10 10 10 10]'*10; %lambda slope sliding surface
gains.phi   = 0.2; %boundary layer thickness
gains.sigma = 2000; %adaptive gain
```

Defining Friction Parameters

```
friction.b    = 1.5; %initial 1.5
friction.fc    = 3;%initial 3
friction.fs    = 5;%initial 5
friction.ft    = 100;%initial 100
friction.delta = 2;%initial 2
```

Defining Linear Trajectory Initial and Final Points

```
% tracking a 3D line
x1=-0.14; x2=0.10;
y1=0.26; y2=0.10; %trajectory II
z1=0.50; z2=0.21;
```

Defining the Smooth Trajectory using 5TH degree polynomial

```
% Define the end-effector motion using fifth order polynomial
r    = 10*(tspan/tf).^3-15*(tspan/tf).^4+6*(tspan/tf).^5;

% calculate the desired position vectors
x=x1+r*(x2-x1);
y=y1+r*(y2-y1);
z=z1+r*(z2-z1);
```

Finding the Angular Position and Velocity of the Joints

```
q=inverse_4dof_matrix([x' y' z'],robot.l,N);
q(:,4) = (2*pi*r);
```

```
% find the angular velocity by diff function
dq=diff(q)/tdom.dt; dq(N,:)=dq(N-1,:);

% save the desired position and velocity to a structure
tdom.Qd=q;
tdom.dQd=dq;
```

Getting the smooth motion of the End-Effector for the simulation

```
%motion of the end effector in 3D space
zz=forward_4dof(q,robot.l);
```

Ode function for A-PD-SMC Controller

```
if i == 3

    td0 =[ 0 0 0 0];

    p0      = [tdom.Qd(1,1) tdom.dQd(1,1) tdom.Qd(1,2)
tdom.dQd(1,2) tdom.Qd(1,3) tdom.dQd(1,3) tdom.Qd(1,4)
tdom.dQd(1,4) td0];

    opts     = odeset('RelTol',1e-5,'AbsTol', 1e-3*ones(12,1));

    [t2,x5] = ode15s(@system_time_4dof_APDSMC,tspan,p0,opts);

    err25    = [tdom.Qd(:,1) tdom.dQd(:,1) tdom.Qd(:,2)
tdom.dQd(:,2) tdom.Qd(:,3) tdom.dQd(:,3) tdom.Qd(:,4)
tdom.dQd(:,4)] - x5(:,1:8);
```

Control System for A-PD-SMC Controller

Inputs – time, angular position and velocity of joints, desired trajectory, gains, robot specifications, friction

Output – velocity and acceleration of the system

```
function xdot = system_time_4dof_APDSMC(t,y)

global robot gains friction tdom torque

%% Robotic Manipulator's Properties

m = robot.mass;
r = robot.r;
```

```

Iz = robot.Iz;
l  = robot.l;

%% Angle handling

q  = [y(1); y(3); y(5); y(7)];
dq = [y(2); y(4); y(6); y(8)];
td = [y(9); y(10); y(11); y(12)];
j   = int32(t/tdom.dt+1)

%% cosines & sines

c    = cos(q);
s    = sin(q);

%% Matrix M

M(1,1) =
(Iz(1)+Iz(2)+Iz(3)+Iz(4))+(m(1)*r(1)^2)+(m(3)*r(3)^2)+(l(1)^2*(m
(1)+m(2)))+(l(1)^2+l(3)^2)*(m(4)+m(3)))+(2*(l(1)*m(1)*r(1))+(l
(3)*m(3)*r(3)))+(2*c(3)*((l(1)*l(3))*(m(3)+m(4)))+(l(1)*m(3)*r
(3))));
M(1,2) = 0;
M(1,3) =
Iz(3)+Iz(4)+(m(3)*r(3)^2)+(l(3)^2*(m(3)+m(4)))+(2*l(3)*m(3)*r(3)
)+(1/2)*c(3)*((l(1)*l(3))*(m(3)+m(4)))+(l(1)*m(3)*r(3)));
M(1,4) = Iz(4);

M(2,1) = 0;
M(2,2) = m(2) + m(3) + m(4);
M(2,3) = 0;
M(2,4) = 0;

M(3,1) =
Iz(3)+Iz(4)+(m(3)*r(3)^2)+(l(3)^2*(m(3)+m(4)))+(2*l(3)*m(3)*r(
3)))+(1/2)*c(3)*((l(1)*l(3))*(m(3)+m(4)))+(l(1)*m(3)*r(3)));
M(3,2) = 0;
M(3,3) = Iz(3) +Iz(4)+(m(3)*r(3)^2)+(l(3)^2*(m(3)+m(4))) +
(2*l(3)*m(3)*r(3));
M(3,4) = Iz(4);

M(4,1) = Iz(4);
M(4,2) = 0;
M(4,3) = Iz(4);
M(4,4) = Iz(4);

%% Matrix C

```

```

C(1,1) = (-dq(3))*(l(3)*m(3)) + (l(3)*m(4)) + (
m(3)*r(3))*(l(1)*s(3));
C(1,2) = 0;
C(1,3) = -(dq(1) + dq(3))*(l(3)*m(3)) + (l(3)*m(4)) +
(m(3)*r(3))*(l(1)*s(3));
C(1,4) = 0;

C(2,1) = 0;
C(2,2) = 0;
C(2,3) = 0;
C(2,4) = 0;

C(3,1) = (-dq(1))*(l(3)*m(3)) + (l(3)*m(4)) +
(m(3)*r(3))*(l(1)*s(3));
C(3,2) = 0;
C(3,3) = 0;
C(3,4) = 0;

C(4,1) = 0;
C(4,2) = 0;
C(4,3) = 0;
C(4,4) = 0;

%% Matrix G

G(1,1) = 0;
G(2,1) = 9.8*(m(4)+m(3)+m(2));
G(3,1) = 0;
G(4,1) = 0;

%% Friction Matrix f

f = -(friction.fc+(friction.fs-friction.fc)*exp(-
(dq/friction.ft).^2)).*sign(dq)+friction.b*dq;
f=2*f;
%% PD Controller

qd = tdom.Qd(j,:)' ;
dqd = tdom.dQd(j,:)' ;
e = qd-q;
de = dqd-dq;

%Control Input
sf=de+gains.l.*e;

```

```

%APD-SMC control
u =
gains.kp.*e+gains.kd.*de+gains.K.*min(abs(sf/gains.phi),1).*sign
(sf/gains.phi) + td;

torque.td3(j,:) = u';
torque.td2(j,:)=f';
torque.td1(j,:)=f';
torque.td3(j,:)=f';

%% Find xdot
xx = M\ (u-f-C*dq-G);
xdot = [dq(1) xx(1) dq(2) xx(2) dq(3) xx(3) dq(4) xx(4)
gains.sigma*sf']';

end

```

Forward Kinematics

Inputs –joint space position

Output – end effector position

```

function pos=forward_4dof(q,l)

pos(:,2)=l(3)*cos(q(:,1)+q(:,3))+l(1)*cos(q(:,1));
pos(:,1)=l(3)*sin(q(:,1)+q(:,3))+l(1)*sin(q(:,1));
pos(:,3)=q(:,2)+0.25;

end

```

Inverse Kinematics

Inputs – end-effector position, robot link lengths, and sampling rate

Output – angular positions of the joints

```

function angle=inverse_4dof_matrix(pos,l,N)

%% Angular position calculations

for i=1:N

x = pos(i,1);
y = pos(i,2);
z = pos(i,3);
% dx = vel(i,1); dy=vel(i,2); dz = vel(i,3);

% 1. distance_2

```



```

angle(i,2) = z - 0.25;

% 2. theta_3
cos3 = ((x)^2 + (y)^2 - l(1)^2 - l(3)^2)/(2*l(1)*l(3));
% law of cosines (c^2 = a^2 + b^2 -2*a*b*cos3)
sin13 = (sqrt(1-(cos3)^2));
sin23 = -(sqrt(1-(cos3)^2)); % two solutions

angle(i,3) = atan2(sin13,cos3);
%angle(i,3) = atan2(sin23,cos3);
%theta_3_2 = atan2(sin23,cos3)*rad2deg;

% 3. theta_1
xx = atan2(x,y) ;
yy1 = atan2((l(3)*sin(angle(i,3))),l(1)+(l(3)*cos(angle(i,3))));
%yy2 = atan2((r1+(r3*cosd(theta_3_2))),r3*sind(theta_3_2));

angle(i,1) = (xx-yy1);
%theta_1_2 = -(xx-yy2)*rad2deg;

% 4. theta_4
angle(i,4) = 0;

end
end

```

Ode setup and control system for PD Controller

```

if i == 1

p0      = [tdom.Qd(1,1) tdom.dQd(1,1) tdom.Qd(1,2) tdom.dQd(1,2)
tdom.Qd(1,3) tdom.dQd(1,3) tdom.Qd(1,4) tdom.dQd(1,4)];

opts     = odeset('RelTol',1e-5,'AbsTol', 1e-3*ones(8,1));

[t2,x2] = ode15s(@system_time_4dof_PD,tspan,p0,opts);
err22    = [tdom.Qd(:,1) tdom.dQd(:,1) tdom.Qd(:,2) tdom.dQd(:,2)
tdom.Qd(:,3) tdom.dQd(:,3) tdom.Qd(:,4) tdom.dQd(:,4)] -
x2(:,1:8)];

u  = gains.kp.*e+gains.kd.*de;
xx  = M\ (u-f-C*dq-G);
xdot = [dq(1) xx(1) dq(2) xx(2) dq(3) xx(3) dq(4) xx(4)]';

```

Ode setup and control system for PD-SMC Controller

```

if i == 2

```

```

p0      = [tdom.Qd(1,1) tdom.dQd(1,1) tdom.Qd(1,2)
tdom.dQd(1,2) tdom.Qd(1,3) tdom.dQd(1,3) tdom.Qd(1,4)
tdom.dQd(1,4)];

opts    = odeset('RelTol',1e-5,'AbsTol', 1e-3*ones(8,1));

[t2,x3] = ode15s(@system_time_4dof_PDSC,tspan,p0,opts);

err23 = [tdom.Qd(:,1) tdom.dQd(:,1) tdom.Qd(:,2)
tdom.dQd(:,2) tdom.Qd(:,3) tdom.dQd(:,3) tdom.Qd(:,4)
tdom.dQd(:,4)] - x3(:,1:8);

sf=de+gains.l.*e;

u  =
gains.kp.*e+gains.kd.*de+gains.K.*min(abs(sf/gains.phi),1).*sign
(sf/gains.phi);

torque.td3PDSC(j,:) = u';
torque.td2(j,:)=f';
torque.td1(j,:)=f';
torque.td3(j,:)=f';

xx  = M\ (u-f-C*dq-G);
xdot = [dq(1) xx(1) dq(2) xx(2) dq(3) xx(3) dq(4) xx(4)]';

```

Published by MATLAB® R2017B

REFERENCES

- Abdolmalaki, R. Y. (2017). Geometric Jacobian Derivation and Kinematic Singularity Analysis for Smokie Robot Manipulator & the Barrett WAM. *5th National Conference on Robotics and Mechatronics (ICROM)*.
- Abera A, B. B. (January 2007). Digital Redesign of Sliding Mode Control Algorithms Using Saturation Function: State and Multirate Output Feedback Approach. *Proceedings of the 45th IEEE Conference on Decision and Control*. Manchester.
- Acob, J. M. (2011). *Hybrid PD Sliding Mode Control for Robotic Manipulators*. Toronto: Ryerson University.
- Adelhedi, F., Jribi, A., Bouteraa, Y., & Derbel, N. (2015). Adaptive Sliding Mode Control Design of a SCARA Robot Manipulator System under Parametric Variations. *Journal of Engineering Science and Technology, Review 8 (5)*, 117-123.
- Alobaidi, S. (March, 2013.). Evaluating Effects of Chattering on the Sliding Mode System. *IJCSI International Journal of Computer Science*, vol. 10.
- Arimoto, S. &. (1984). Stability and robustness of PID feedback control for robot manipulators of sensory capability. *Robotics Research: The First International Symposium*,, 783-799.
- Armstrong, B. &. (2000). Nonlinear PID Control with Partial State Knowledge: Damping without Derivatives. . *The International Journal of Robotics Research*, 19(8),, 715-731.
- Åström , K. J., & Wittenmark , D. (1995). *Adaptive Control*. Addison-Wesley.
- Åström, K. J. (2001). *The future of PID control. Control Engineering Practice*, 9(11) 1163-1175.)..

- Bajd M., M. J. (n.d.). *Robotics. Intelligent systems, Control and Automation: Science and Engineering*. 2010: Springer.
- Bajd, T., Mihelj, M., Lenarcic, J., Stanovnik, A., & Munih, M. (n.d.). *Robotics -2010*. Springer.
- Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson.
- Dessaint, L. A. (April 1992). An Adaptive Controller for a Direct-Drive SCARA Robot. *IEEE Transactions on Industrial Electronics*, Vol. 39, No. 2.
- Dombre, E., & Khalil, W. (2002). *Modeling, Identification & Control of Robots*. Kogan Page Science.
- Fridman, L. (2012). Sliding Modes after the First Decade of the 21st Century. *Sliding Mode Enforcement after 1990: Main Results and Some Open Problems*, 412, Springer Verlag: Berlin, , 3–57,.
- Hélouvry, B. A., Dupont, P., & Canudas de Wit, C. (1994). A Survey of Model, Analysis Tools and Compensation Methods for the Control of Machnies with Friction. *Automatica*, Vol. 30, No. 7,, 1083-1138.
- Huang, Y. J., Kua, T. C., & Chang, S. H. (2008.). Adaptive sliding-mode control for nonlinear systems with uncertain parameters. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions Vol. 38, no. 2,, 534-539*.
- International Federation of Robotics. (2018). *Executive Summary World Records 2018 Industrial Robots*.
- Kapoor, N., & Ohri, J. (2013). Fuzzy Sliding Mode Controller (FSMC) with Global Stabilization and Saturation Function for Tracking Control of a Robotic Manipulator. *Journal of Control and Systems Engineering Vol. 1, no. 2, 50*.

- Koren, Y. &. (1991). Variable-gain cross-coupling controller for contouring. *CIRP Annuals-Manufacturing Technology*, 40(1), 371-374.
- Lee, J. M., Lee, M. C., Son, K., Lee, M. H., & Han, S. H. (February 1997). Implementation of a Dynamic Control System for SCARA Robot. *IFAC Proceedings Volumes Volume 30, Issue 1*, 297-302.
- Levant, A. (June, 2010.). Chattering Analysis. *IEEE Trans. on Automatic Control*, Vol. 55 No. 6, 1380-1389.
- Middelton, R. &. (1988). Adaptive Computed Torque Control for Rigit Robotic Manipulators. *Systems and Control Letters* 10(1), 9-16.
- Moghadam , A. A., Gharib, M. R., & Moavenian, M. (n.d.). Modelling anf Control of a SCARA robot using quantitative feedback theory. *Journal of Systems and Control Engineering, Proc. IMechE*, Vol.223 Part I.
- Murray, R. M., Li, Z., & Sastry , S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.
- Ouyang, P. (. (2005). *Hybrid Intelligent Machine Systems: Design, Modeling and Control*. Saskatchewan: PhD Thesis, University of Saskatchewan, Department of Mechanical Engineering.
- Ouyang, P. (2005). *Hybrid Intelligent Machine Systems: Design, Modeling and Control*. PhD Thesis, University of Saskatchewan, Department of Mechanical Engineering. Saskatchewan: University of Saskatchewan.
- Ouyang, P., Tang, J., Yue, W. H., & Jayasinghe, S. (2016). Adaptive PD plus sliding mode control for robotic manipulator. *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, (pp. 930-934). Banff, AB.

- Qu, Z. &. (1991). Robust PID control of robots. . *International Journal of Robotics and Automation*, 6(4), 228-235.
- Rugh, W. (1987). Design of nonlinear PID controllers. *American Institute of Chemical Engineers Journal*, 33(10), , 1738-1742.
- Sage, H. G. (1999). Robust Control of Robot Manipulators: A Survey. *International Journal of Control*, 72(16), 1498-1522.
- Sastry, S. &. (1989). *Adaptive Control: Stability, Convergence, and Robustness*. . Prentice-Hall, Inc.
- Sen, M. A., Bakircioğlu, V., & Kalyoncu, M. (Dec 07-09, 2017). Modelling and PID Control of Scara Robot. *International Conference on Engineering Technologies (ICENTE'17)*.
- Seraji, H. (n.d.). Nonlinear and Adaptive Control of Force and Compliance in Manipulators. *The International Journal of Robotics Research*, 17(5), 467-484.
- Shabana, A. (2001). *Computational Dynamics*. John Wiley & Sons, Inc.
- Siciliano, B. S. (2009). *Robotics: Modelling, Planning and Control*. London: Springer.
- Siciliano, L. S. (1999). *Modelling and Control of Robot Manipulators*. Springer .
- Slotine, J.-J. E. (1991). *Applied Nonlinear Control*. New Jersey: Prentice-Hall, Inc.
- Takegaki, M. &. (1981). A new feedback method for dynamics control of manipulators. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control*, 103, 119-125.
- Tomei, P. (1991). Adaptive PD Controller for Robot Manipulator . *IEE Transaction on Robotic Automation* 7(4), 565-570.
- Visioli, A. (February 2002). On the Trajectory Tracking Control of Industrial SCARA Robot Manipulator. *IEEE Transactions on Industrial Electronics*, Vol. 49, No.1,.
- White, D. A. (1992). *Handbook of Intelligent Control*. . Van Nostrand Reinhold.

- Wu, X., Li, Y., Zhang, J., & Zhu., Q. (2014). Sliding mode control for neutral systems with uncertain parameters. *International Journal of Modelling, Identification and Control*, Vol. 21, Issue 1, 65-71.
- Xu, L. &. (2001). Adaptive Robust Precision Motion Control of Linear Motors with Negligible Electrical Dynamics : Theory and Experiments. *IEEE/ASME Transactions on Mechatronics* 6(4), 444-452.
- Xu, Y. H. (1995). A nonlinear PD controller for force and contact transient control. *Control Systems, IEEE*, 15(1), 15-21.
- Yeh, S. S. (2003). Analysis and Design of Integrated Control for Multi-Axis Motion Systems. *IEEE Transactions on Control Systems Technology*, 11(3), 375-382.
- Zargoun, B. I. (2013-2014). Modelling and Control of SCARA manipulator. *International Conference on Robot PRIDE 2013-2014-Medical and Rehabilitation Robotics and Instrumentation*, . ConfPRIDE 2013-2014.