T.C.

MUĞLA SITKI KOÇMAN UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

DISTURBANCE OBSERVER IMPLEMENTATION OF SCARA ROBOT WITH MATLAB SIMULATION

MASTER OF SCIENCE THESIS

FARIS FUQAHA

MAY 2021

MUĞLA

FARIS FUQAHA

ELECTRICAL & ELECTRONICS ENGINEERING
THESIS MUĞLA 2021

MSc

T.C.
MUĞLA SITKI KOÇMAN UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

DISTURBANCE OBSERVER IMPLEMENTATION OF SCARA ROBOT WITH MATLAB SIMULATION

MASTER OF SCIENCE THESIS

FARIS FUQAHA

MAY 2021
MUĞLA

MUGLA SITKI KOÇMAN UNIVERSITY

Graduate School of Natural and Applied Sciences

THESIS ACCEPTANCE APPROVAL

The thesis, submitted by **FARIS FUQAHA** entitled **"DISTURBANCE OBSERVER IMPLEMENTATION OF SCARA ROBOT WITH MATLAB SIMULATION,"** has been accepted 28/May/2021 in fulfillment of Master's Degree in the Department of Electrical and Electronics Engineering by the thesis committee listed below.

THESIS DEFENCE COMMITTEE

Assoc. Prof. Dr. Selami BEYHAN (Head of Committee) Signature:

İzmir Demokrasi University, İzmir

Electrical And Electronics Engineering

Assoc. Prof. Dr. Mahmut TENRUH (Supervisor) Signature:

Electrical And Electronics Engineering
Muğla Sıtkı Koçman Üniversitesi, Muğla

Dr. Kutlu KARAYAHŞİ (Member) Signature:

Electrical And Electronics Engineering,

Muğla Sıtkı Koçman Üniversitesi, Muğla

DEPARTMENT APPROVAL

Prof. Dr. Bahadır Süleyman YILDIRIM (Head of Department) Signature:

Electrical And Electronics Engineering ,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Assoc. Prof. Dr. Mahmut TENRUH (Supervisor) Signature:

Electrical And Electronics Engineering ,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Date of Defense: 28 /May /2021

I declare; all results, documents, information, and files that I gained and presented as part of this thesis were obtained by me within this thesis's study. Also, I declare; all original information and results belong to someone else that was not achieved within the study of this thesis were referenced by ethical rules of academy and science.

Faris Fuqaha

28/5/2021

Signature:

To My Family

ACKNOWLEDGEMENTS

At the very outset, all my prayers and thankfulness are to Allah the almighty for facilitating this work and granting me the opportunity to be surrounded by great and helpful people at Muğla Sıtkı Koçman University.

I want to express my everlasting thanks to my supervisor Assoc. Prof. Dr. Mahmut Tenruh, for his valuable encouragement, guidance, and monitoring. This work would not have reached the point of fruition, so I ask Allah to reward them on my behalf.

I would never have attempted to obtain a master's degree if it were not for the continuous encouragement that I received from my father, the man to whom I will be grateful for the rest of my life.

The warm heart, my mother, deserves all the credit here; she has been a source of inspiration to me for years. I would never forget her continuous prayer for the sake of my success.

Finally, I would like to thank my colleagues and friends for their good friendship during my master's period.

ÖZET

KONUM SAPMA GÖZLEMCİSİ İLE BİR SCARA ROBOTUN MATLAB SİMÜLASYON UYGULAMASI

Faris FUQAHA

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mahmut Tenruh

Mayıs 2021, 74 sayfa

Bu tezde SCARA robot manipülatörleri için doğrusal olmayan konum sapma gözlemcisi ve Uyarlı Kalman Filtresi tasarım çalışmaları yapılmakta, elde edilen sonuçların karşılaştırmalı incelemesi gerçekleştirilmektedir. Konum sapma gözlemcisi ve Uyarlı Kalman Filtresi kullanımı dış etkilere maruz olan kontrol sistemlerinin kararlılığının ve etkinliğinin artırılması için önemli yöntemlerdendir. Mekatronik sistemlerin izlenmesi için bu yöntemler yaygın olarak kullanılmaktadır.

Bu çalışma bir SCARA PRPR tipi robot manipülatörünün tasarımı ve yazılım uygulamasını kapsamaktadır. İlk olarak tasarımı yapılan SCARA robot için mekanik hesaplamalar gerçekleştirilmiştir. Bu hesaplamaların ardından, D-H ve analitik yöntemler kullanılarak ileri ve ters kinematik denklemler elde edilmiştir. MATLAB simülasyonu ile robot üzerindeki kuvvetleri tanımlayan dinamik analizi gerçekleştirilmiştir. Robotu etkileyen iki tür kuvvet mevcuttur. Birincisi, robot modellemesindeki belirsizliklerden kaynaklanan dahili kuvvetler, robotun eklemlerinde oluşan sürtünmeler gibi. İkincisi, uç efektörün işlevini etkileyen ve dış faktörlerden kaynaklanabilen harici bir kuvvettir. Harici bir kuvvet hedef koordinatlara ulaşmada hatalara sebep olur. Bu tezde bir konum sapma gözlemcisi algoritması kullanılarak bu sorunun çözülmesi hedeflenmekte, konum sapma gözlemcisi ve Uyarlı Kalman Filtresi sonuçları karşılaştırılmaktadır.

Anahtar Kelimeler: SCARA Robot, SCARA Robot Kinematiği, Ters Kinematik, Dinamik, Konum sapma gözlemcisi, Uyarlı Kalman Filtresi.

ABSTRACT

DISTURBANCE OBSERVER IMPLEMENTATION OF SCARA ROBOT WITH MATLAB SIMULATION

Faris FUQAHA

Master of Science (M.Sc.)

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Mahmut Tenruh

May 2021, 74 pages

In this thesis, the design of Nonlinear Disturbance Observers and Adaptive Kalman Filter for robotic manipulators of SCARA is studied and comparisons of the results have been realized. The use of Disturbance Observers and Adaptive Kalman Filter is an important way to increase the stability and efficiency of the control systems subject to disturbances. Rigorous observers for the monitoring of mechatronic devices have been commonly used.

This study includes the design and software implementation of a SCARA PRPR-type robot manipulator. First, the mechanical calculations are realized for the designed SCARA robot. Then, forward and inverse kinematic equations are derived by using D-H and analytical methods. The MATLAB simulation presents the dynamic analysis that describes the force effects on the robot. Two types of forces impact the robot. First, the internal force is created due to uncertainties in robot modeling, for example, friction in the joints of the robot. Second, the external force may occur because of the external factors that influence the end effector role. The external force causes errors in reaching the target coordinates. This thesis aims at solving this problem using an algorithm for observing disturbances, and the results with Disturbance Observer and Adaptive Kalman Filter are compared.

Keywords: SCARA robot, Kinematics of SCARA robot, Inverse Kinematics, Dynamics, Disturbance Observer, Adaptive Kalman Filter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SCARA	Selective Compliance Assembly Robot Arm
RRPR	Revolute - Revolute - Prismatic – Revolute
DH	Denavit-Hartenberg
DOF	Degrees Of Freedom
FK	Forward Kinematic
IK	Inverse Kinematic
T	Homogenous Transformation Matrix
α_i	Link Twist of Two Joints
a_i	Link Length
d_i	Link Offset
θ_i	Joint Angle
c	Cosin
s	Sin
β	Orientation angle
J	Jacobian Matrix
F	Forces
K	Kinetic energy
P	Potential energy
L	Lagrange equation
DOB	Disturbance Observer
M	Mass matrix
C	Coriolis matrix
G	Gravity matrix
τ	Tourqe
P	Prismatic matrix
R	Rotation matrix
c_{12}	$\cos\,(\theta_1 + \theta_2$
s_{12}	$\sin\,(\theta_1 + \theta_2)$

$c_{12\text{-}4}$	$\cos(\theta_1 + \theta_2 - \theta_4)$
$s_{12\text{-}4}$	$\sin(\theta_1 + \theta_2 - \theta_4)$
I	Moment of inertias
$\dot{\theta}$	angular velocity
$\ddot{\theta}$	angular acceleration
P_x, P_y	horizontal robot coordinates

1. INTRODUCTION

1.1. Introduction

Recently, robotics has witnessed an exponential jump in the designing and manufacturing aspects and their daily life usage. Nowadays, it is considered an attractive topic for scientists and researchers. In general, robotics is a multidisciplinary field where many scientists from different backgrounds work together; cybernetics, mechanics, controls, computers, electronics, etc.(Technische Universtität München, 2018).

In the 1940s, Robot as a mechanical artifact was conceived as an automaton in the human appearance, but devoid of sensation, when Russian Isaac Asimov, the well-known science fiction writer, the robot as a "positronic" brain programmed by a person, which meets some rules of ethic behavior ("Advanced Textbooks in Control and Signal Processing," 2018). Asimov then introduced the robotics as a science dedicated to the study of robots that was based on three fundamental laws (Critchlow, 1985):

1. A robot cannot damage a person by inaction to protect a human being.

2. A robot shall adhere to human orders unless such instructions are contrary to the first law.

3. A robot must preserve its life unless it struggles with the first or second rule.

These laws set behavior rules as conditions for the construction of a robot, which has since acquired the connotation of an industrial product designed by engineers or technical specialists to help humans.

Various Japanese manufacturers introduced the SCARA (Selective Compliance Assembly Robot Arm) robots in 1979 (Nof - Handbook of industrial robotics (1999).pdf, n.d.). SCARA Robotics manipulators are subject to many disturbances, such as joint frictions and other unknown payloads that affect the accuracy of positioning. Solving this problem through disturbance observers is an efficient way to eliminate these distractions and increase tracking efficiency.

In the late 1980s, the new control of mechatronic systems was introduced, and it was called a "disturbance observer" (Ha & Back, 2018; Kouhei, Shibata, & Murakami, 1996). Unlike the state observers used to recreate the system's unknown states, the disturbance observers are used to calculate unknown system inputs. It is used to estimate and remove unknown interference in control systems and to enhance the stability and efficiency of these systems.

1.2. Objectives

The main objectives of the thesis :

1. Studying the kinematic model obtained for the robot:
    - Research kinematics theory to analyze the four DOF arm of SCARA robot.
    - The Denavit-Hartenberg (D-H) model is used to derive the equations of forward kinematics for the SCARA Robot.
    - Calculation of the angles of the SCARA Robot by using inverse kinematics.
    - Derivation of the jacobian matrices (angular and linear velocity).
    - Dynamic analysis of the robot.
2. Applying inverse dynamics control.
3. Applying a Path and Trajectory planning algorithm.
4. Studying the forces that affect the joints and applying the disturbance observer. The robot is affected by two types of forces:
    - Internal force:

    It is formed due to the uncertainty in modeling the robot; for example, friction occurred at the joints of the robot.

    - External force:

    This occurs as an effect of external forces on the robot.

5. Implementation with Mat-Lab program.

1.3. Problem statement

When the robot is affected by external forces, it will lose its original position, and thus an error will occur in the Robot work. To eliminate the effect of the disturbance, it should be estimated using an online observer, and then the estimated value should be compensated in the controller. Then the robot will overcome this external force by applying an opposite force that is controlled by the disturbance observer.

1.4. Thesis Contributions

The contribution of this thesis is the design of the SCARA robot arm. The SCARA robot has four degrees of freedom. In this thesis, the robot is designed as "RRPR (Rotational Rotational Prismatic Rotational) type". Then, the disturbance observer (DOB) is applied with the SCARA robot arm, and the results are compared with the Adaptive Kalman Filter (AKF) to estimate and overcome unknown disturbance effects in the control system to improve the stability and performance. A previous study (Prajumkhaiy & Mitsantisuk, 2016) used the disturbance observer to estimate and compensate for the internal forces (friction forces) in the system with a PID (Proportional–Integral–Derivative) controller. This thesis proposes a different method, which uses the disturbance observer to estimate and suppress external forces with PD (Proportional–Derivative) inverse dynamic controller. The model is implemented in Matlab programming to observe the stability and performance improvements of the SCARA robot under the effects of the external forces.

1.5. Organization of the thesis

This thesis is organized as follows: Chapter 2 is the Background: This chapter describes the review of Robotics in general. Then, illustrates the types of robotics and the selection of SCARA robot arm to apply in this study. It also describes the dynamics, PD inverse controller, and disturbance observer. Chapter 3 gives the structure and mathematical analysis of the SCARA Robot. This chapter explains the design of the SCARA robot used in this thesis by mechanical calculations. First, the robot forward and inverse kinematic equations with D-H parameters are introduced. Second, the differential kinematics and statics are presented. Third, the dynamic equations and inherent dynamic properties of serial robotic manipulators are introduced. Finally, after obtaining all the matrices and equations from this analysis, the model with a disturbance observer is implemented with the MatLab program. Chapter 4 provides the results and discussion of this study. Conclusions and recommendations are given in Chapter 5.

2. LITERATURE REVIEW

2.1. The review of the robotic systems

The structures of robotic systems :

1. Mechanical structure: All links and joints.
2. Actuators: The mechanisms that provide the forces necessary to move the mechanical structure.
3. Controller system: provides the actuators with the input necessary to obtain the desired position, force, speed, etc.

2.1.1. Mechanical Structure

Link is called any connection point between two parts of a robot arm. These links are numbered from the first link (link 0) to the last link (link n), and each link has a coordination system. The following rules are used for the design of coordination for each link (Sukmaningrum, Puji Suciance, et al., 2008):

- Zn is the approach direction of the tool
- Yn is the slide direction of the gripper
- Xn is the normal direction of the other axis

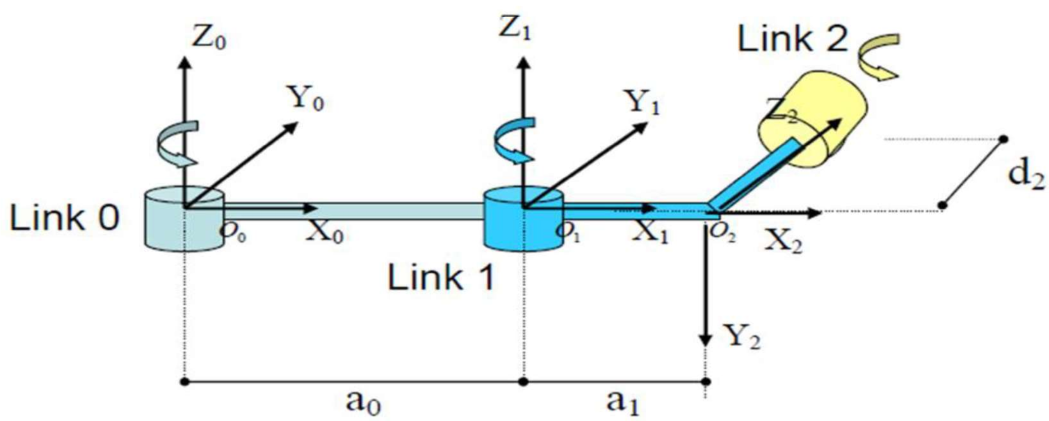


Figure 2.1. Establishing links (Sukmaningrum, Puji Suciance, et al., 2008)

Figure 2.1 and figure 2.2 shows links and the coordinates for each joint of the robot. The robot consists of links and joints. Each joint can connect two links, and a link is connected by two joints. Each joint and link has a number from the base frame to the end effector (base frame: the first device of the robot, end effector: the end device of the robot arm).

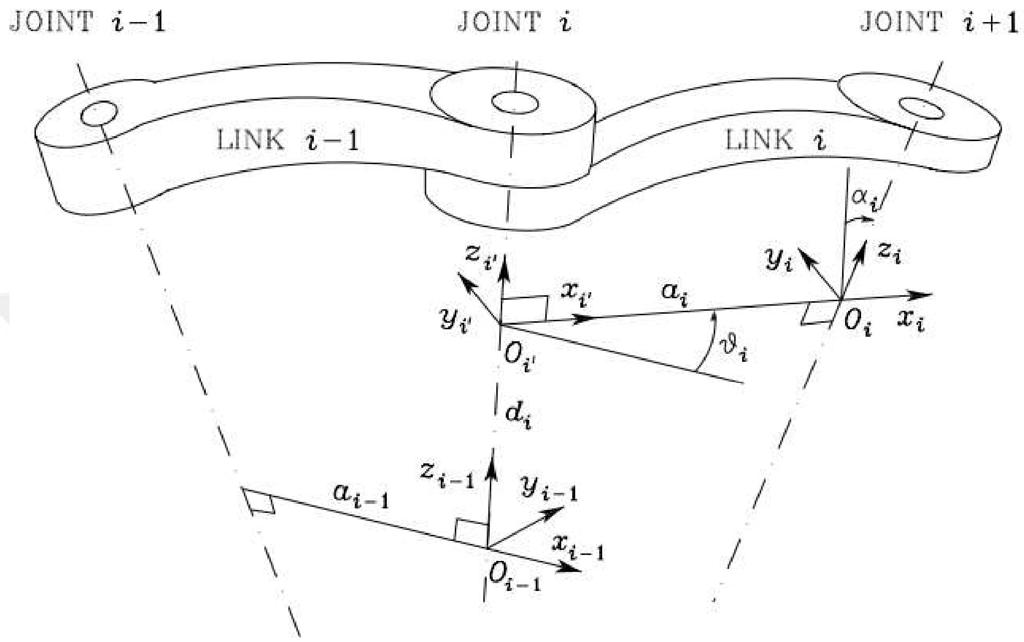


Figure 2.2. Denavit–Hartenberg kinematic parameters ("Advanced Textbooks in Control and Signal Processing," 2018)

2.1.2. Actuator

The selection of actuators determines how successful a robot is in executing a particular task. Mechanical or electrical actuators can have varying strengths and limitations, as shown in Table 1. The basic actuators used to regulate moments include (Sukmaningrum, Puji Suciance, et al., 2008):

- Air motors
- Hydraulic motors
- Breaker motors
- Stepper motors
- Servo motors

Electric motors are commonly used in robotics, whether they are stepper motors or servo motors. Servo motors are best suited for closed-loop systems, and stepper motors are better suited for open-loop systems.

Table 2.1: Actuator Comparison

Actuator Type	**Strengths**	Weaknesses
Air Motor	Low cost Easily maintained Simple to operate	Audible compressor noise Inefficient system Difficult to regulate Speed
Hydraulic Motor	High loads possible Simple to operate	Slow system Inefficient system High maintenance requirements
Brake	Low cost Effective for light loads Easy to Perform speed matching	Uncontrolled acceleration Components prone to wear Non-repeatable system
Stepper Motor	Simple control Constant load Accurate position	Cannot vary load Can lose steps Resonance problem
Servo Motor	High performance Small motor size Can operate at high speed	Higher cost system Performace limited by control Speed limited by electronics

2.1.3. Controller system

In general, the systems could be one of two types: open-loop or closed-loop.

Open-Loop (Sukmaningrum, Puji Suciance et al., 2008):

The controller sends a signal to the driver based on the set point. The driver moves the actuator to the appropriate point with the right amount. No action is taken unless the setpoint changes. Figure 2.3 shows the open-loop system diagram.

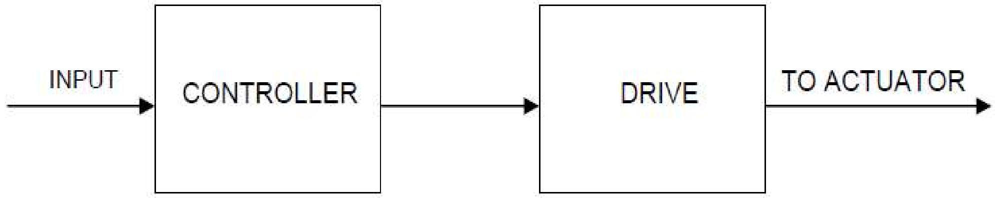


Figure 2.3. Open loop system diagram (Sukmaningrum, Puji Suciance et al., 2008)

Open-loop systems are most effective when an actuator should move constantly until commanded to stop or when the actuation is exceptionally accurate.

Closed-loop:

As shown in Figure 2.4, a closed-loop system operates by sending a command signal to the motor, then returns a feedback signal to the controller showing the motor status. The controller compares the control and feedback signals and thus determines the output.

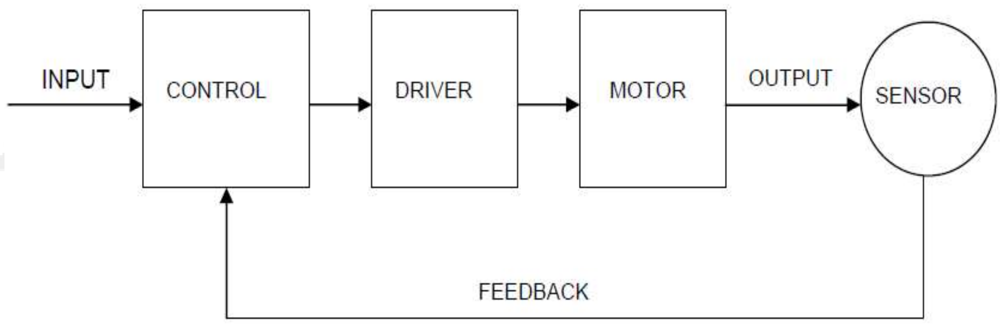


Figure 2.4. Closed loop system (Sukmaningrum, Puji Suciance, et al., 2008)

When control is required for a variety of complex motion, the closed-loop systems are effective. Closed-loop allows for precise control over speed and position through the use of feedback devices such as tachometers and encoders. Since additional components are required, a closed-loop system is more complex and can cost more initially; but these systems enhance the degrees of control**.**

2.2. Robotics for industrial application

Industrial robotics studies, the design of robots, control, and application in the industrial field, and its products have now reached the level of mature technology. A robot in industrial applications works in an environment whose physical and engineering properties are often known ("Advanced Textbooks in Control and Signal Processing," 2018). In the 1960s, the initial industrial robots were designed according to two technologies: teleoperators and numerical control machines. The first robot manipulators were identified with their precursors by:

- Flexibility; making it suitable to use at the tip of various end-effectors.
- Adaptability; due to the use of sensors, it can be adapted to previously unknown situations.
- Accuracy in positioning because it uses techniques with feedback control.
- Repeatability; can be programmed to uses in different operations.

Industrial robots can be analyzed in five categories:

- Cartesian Robots.

The cartesian robot is the most popular one used in industrial applications, because it is easy to use and program. The operators can choose this robot for the flexibility to meet the needs of the user in industrial application.

- Cylindrical Robots.

The cylindrical robots are very simple. They are made of two moving elements: rotary and linear actuators.

- 6-axis Robots.

These types of robots consist 6-Axis. These robots operate like a human arm and can pick up materials and move them from one plane to another, and it is used for large assembly jobs such as putting seats into a car on an assembly line.

- Delta Robots.

These types of robots are the fastest, precise and most expensive. They are the best for fast pick-and-place or product transfer applications, and they have a unique, dome-shaped work envelope in which they can achieve very high speeds.

- Scara Robots.

This type of robot will be discussed in the next section.

2.3. Selective Compliance Assembly Robot Arm (SCARA)

The first SCARA robot version by Hiroshi M. had 4 DOF with RRPR joints. Due to its ability to adaptability and versatility, the SCARA Robot contributed significantly to the flexibility and effectiveness of Japanese assembly systems in the 1980s and reduced the overall production costs (Villela, 2013). This robot has been used in various industries such as electronics, automotive, and pharmaceuticals in various sizes. The most popular applications include collection, positioning, assembly, and packaging.

The SCARA robot was first used in industrial applications for assembling the PCB of audio amplifiers. The first SCARA Robot arm is shown in Figure 2.5.

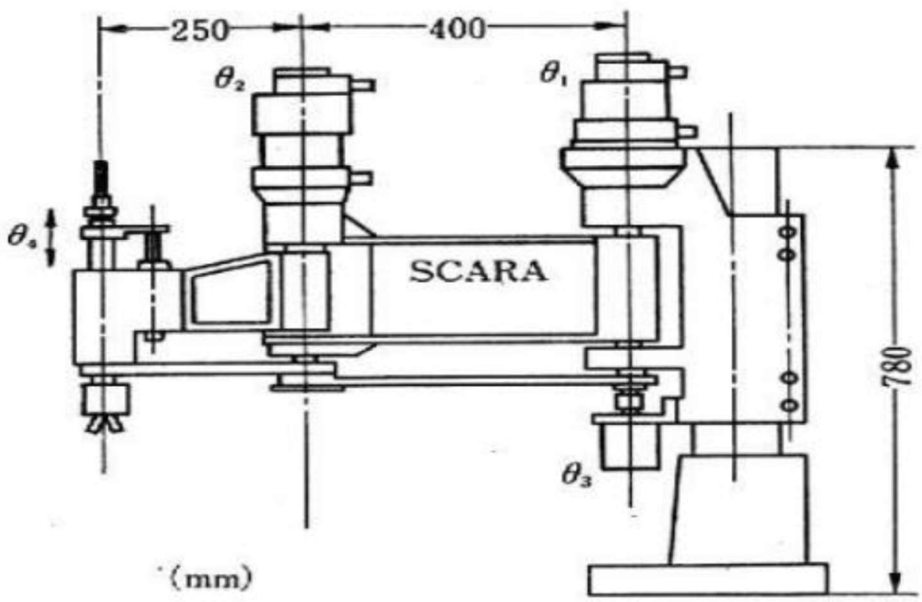


Figure 2.5. The first SCARA robot arm (Manasra, Salaymeh, & Natsheh, 2018)

Despite the great development and diversity of robotics, the SCARA robot is still very common in industries. Many factors contribute to the success of this robot:

- Precision
- High speed
- Small dimensions
- Smooth motion
- The structure is efficient and simple
- Simple for use and installation

- Small backlash

2.4. Dynamics

The dynamics of a robot describes the forces affecting the robot. While the equations of kinematics describe the movement of a robot without relation to the forces, the dynamic equations describe the relationship between motion and force. The equations of kinematics are used for designing robots, designing control algorithms, and simulation. The forces on a robot can be studied by using Euler-Lagrange equations.

In the equation of Euler-Lagrange, we start by deriving the equations for a one-degree system from Newton's Second Law. Then, the Euler-Lagrange equation in the general case is derived. To calculate the equations of Euler-Lagrange in the specific case, first, we should calculate the Lagrangian system for each joint. That is the difference between kinetic energy (KE) and potential energy (PE). After that, a partial differential equation is derived

Euler-Lagrange equation Can be written as follows:

$$\frac{d}{dt}\frac{\partial L}{\partial_{\dot{q}_i}} - \frac{\partial L}{\partial_{q_i}} = T_i \qquad i = 1, \dots \dots n \qquad (2.1)$$

Where L = K−P

K: total kinetic energy.

P: total potential energy, L: Lagrange equation.

2.5. PD inverse dynamic controller

One of the fundamental problems of the dynamics of mechanical systems is determining the moments and forces (internal or external) acting on the system, having given specified properties of the system motion. Historically, this so-called inverse problem was first considered in analytical mechanics in the works of

Newton, Lagrange, Poincare, and others. One of the first inverse problems was the Newtonian problem of determining a force under which the motion of planets is governed by the three of Kepler's laws (Jankowski, 2004). A result due to Bertrand (Féjoz & Kaczmarek, 2004), shows that in the motion of a given dynamic system if the acting forces depend solely on the Coordinates of their points of application, the unknown forces can be found provided a motion integral satisfying certain conditions is known. The Suslov problem consists of finding a potential that determines forces, causing a holonomic mechanical system with specified integrals. The opportunity of modeling many practical problems occurring in different fields of engineering~~.~~, with years, solving the inverse problem has become much more popular. The new problems concern not only the determination of generalized forces, but mechanical system parameters and constraints imposed on the system as well ("Methods of Solution to Inverse Problems," 2007), for which the motion of the system exhibits some desired properties

With progress in the development of inverse dynamics methods, it has appeared that to preserve the specified motion characteristics. It is necessary to control the considered mechanical system to guarantee the stability of motion and insensitivity to external forces disturbances. Moreover, the progress in the theory of control of mechanical systems caused an interest in solving the inverse dynamics problem. It was motivated by the fact that based on the inverse problem, it is possible to construct effective procedures for the synthesis of closed-loop controllers for motion control of material systems of various physical nature and Structure. Many of the widely used methods of the control system design are directly or indirectly related to the inverse dynamics problem. In some of these methods, a reference system is used, which defines the desired motion of the system, which has to be performed by the designed controller. The problems of synthesizing closed-loop control systems with state feedback can be formulated and solved as the inverse dynamics problem. And It is necessary to notice that the control laws can be formulated infinite form for both linear and nonlinear systems. Using the modern control theory, the construction of inverses of nonlinear systems can be done using the algorithm of Hirschorn (Hirschorn, 1979) and its developments. Also, a similar goal can be accomplished by applying the input-output decoupling control based on the geometric control theory.

2.6. Disturbance observer

The disturbance observer can be used to reduce the cost and number of sensors required in a control system (Mohammadi, 2011; Radke & Gao, 2006). There are many applications that can be used for disturbance observer such as direct drive motor control (Yu & Su, 2013), DC-servomotor control, current induction-motor control, and active vehicle steering. Disturbance observer has been used successfully since their introduction.

The concept of disturbance observer has inspired many applications in the fields of robotics and other mechatronics. Robot manipulators are affected by unknown forces (internal or external), and the disturbance observers are used to estimate these forces. Figure 2.6 illustrates a typical disturbance observer used in a robotic system. The DOB uses the robot's known forces and output variables to identify the unknown forces acting on the robot.

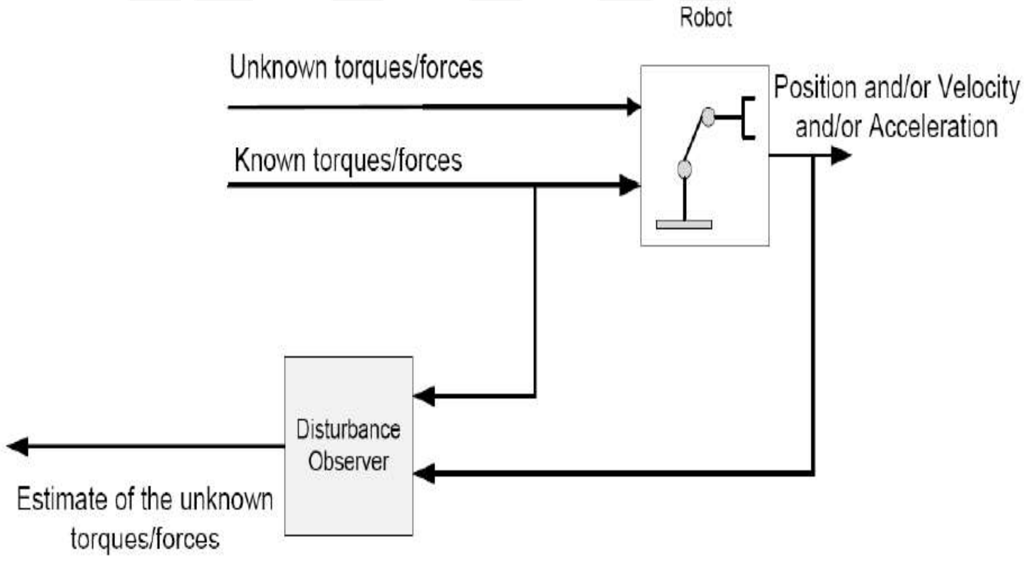


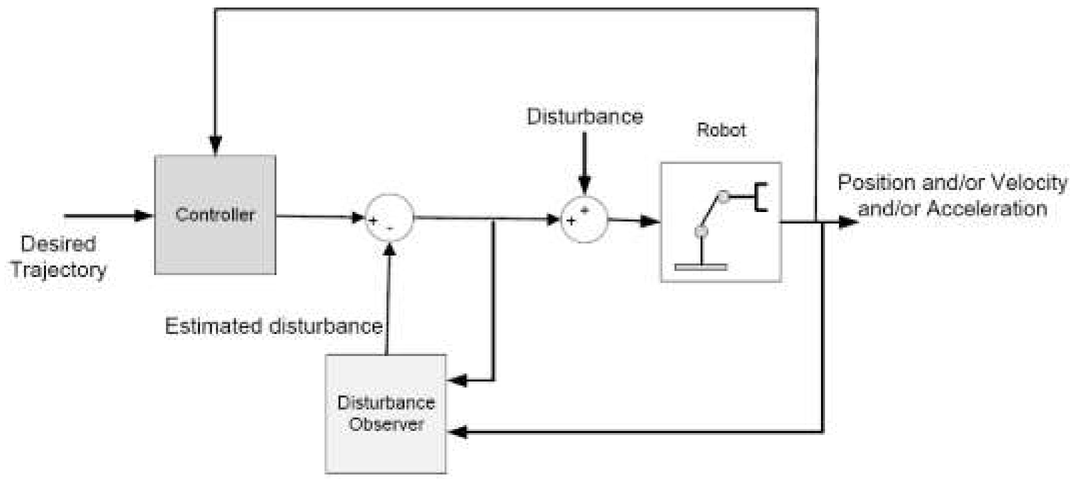
Figure 2.6. Block diagram of disturbance observer (Mohammadi, 2011)

Applications of disturbance observers in robotic systems:

There are a large number of applications in DOB in robotic systems. It can be classified into two categories, the first one is force sensing and, second one is disturbance rejection (Mohammadi, 2011):

1- Force sensing: The end-effector of the robot comes into contact with a compatible surface in many robotic applications. A sensor is needed to measure these contact forces to ensure good system performance. Using disturbance observers will eliminate the need for costly force sensors in these applications. Therefore, disturbance observators were successfully used to monitor sensorless force (K. S. Eom, Suh, Chung, & Oh, 1998; Katsura, Matsumoto, & Ohnishi, 2007; Shimada, Ohishi, Kumagai, & Miyazaki, 2010). It was argued that noise-corrupted measurements and compliant mechanical structure of force sensors could destabilize the interaction of robots with the environment, and thus justify using disturbance observers rather than force sensors.

2- Disturbance rejection: The concept behind disturbance observer-based control of robotic manipulators is to group all the internal and external disturbances affecting the manipulator into one disturbance term. The Robotic manipulators are subject to various types of disturbances, such as unknown payloads and joint frictions that negatively affect the robot, such as repeatability and accuracy of positioning. And using the disturbance observers is an efficient track of crushing these disturbances and enhancing tracking and increasing efficiency.



2.7. A disturbance observer used for disturbance rejection in a robotic control system (Mohammadi, 2011)

Figure 2.7 illustrates a feedback structure, including a DO for controlling one position of the robotic manipulator. Assume that the observer estimates the disturbance exactly. In this situation, the closed-loop system's disturbance will be canceled, and it seems we are dealing with a robot without disturbances, and we can easily design a controller. Due to the feedforward nature of the compensation, disturbance observers can work easily, well, and smoothly with no significant feedback gains (Arbor, 1997). For example, the disturbance observer can be used in independent joint control where load variations, joints, and dynamic uncertainties are collectively regarded as the lumped disturbance term (Chen, Yang, Guo, & Li, 2016; Kwang Sik Eom, Suh, & Chung, 2001; Homayounzade & Khademhosseini, 2019; Zhongyi, Fuchun, & Jing, 2008)

Effective elimination of these disturbances decouples the joints and enables simple controllers to be individually configured for each degree of freedom. Another usage by disturbance observers is to boost manipulator tracking and increase efficiency by calculating and compensating friction (Bona & Indri, 2005; Sawut, Umeda, Park, Hanamoto, & Tsuji, 2001). A significant feature of friction compensation systems based on disturbance observers is that they are not based on a specific friction model.

In general, industrial robots use fault detection systems to assess if there was a defect in the system, such as a collision. In several robotic applications, perturbation observers have been used to detect faults (Caccavale, Cilibrizzi, Pierri, & Villani, 2009; Schneider & Frank, 1996) (Chan, 1995).

Literature review

There are many literature studies to solve the Robotics problems in industrial applications, such as external disturbance on the robot. Kalman Filter-Based Disturbance Observer and its Applications to Sensorless Force Control suggests a design methodology to improve the disturbance observer design. The paper's major contribution is the design of a disturbance observer combination with a Kalman filter and a multisensor system (Mitsantisuk, Ohishi, Urushihara, & Katsura, 2011). Advanced Disturbance Observer Design for Mechanical Positioning Systems suggested a method to compensate for the disturbance that is called a robust internal-loop compensator (RIC) (Critchlow, 1985). Control of Redundant Manipulators Considering Order of Disturbance Observer had suggested disturbance observers of various orders in the joint and workspace to improve the performance of system hybrid position/force control of redundant manipulators. It has also been shown that a correct selection of the disturbance observer coefficient can increase robust stability without influencing basic performance (Komada, 2000).

A study about the design, construction, and control of a SCARA manipulator with 6 degrees of freedom with RRPRRP types introduces a SCARA robot which has six degrees of freadom, and desings an interface for the connection between the PC controller and the robot (Urrea, Cortés, & Pascal, 2016). Figure 2.8 shows the system's general diagram.

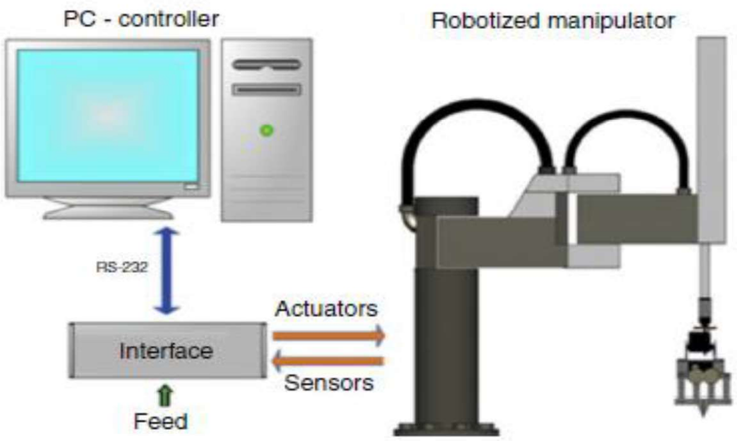


Figure 2.8. General diagram of the system (Urrea et al., 2016)

Figure 2.8 illustrates the general diagram of the system. The robot must communicate with the PC controller and an electrical interface is developed to perform the tasks (Urrea et al., 2016).

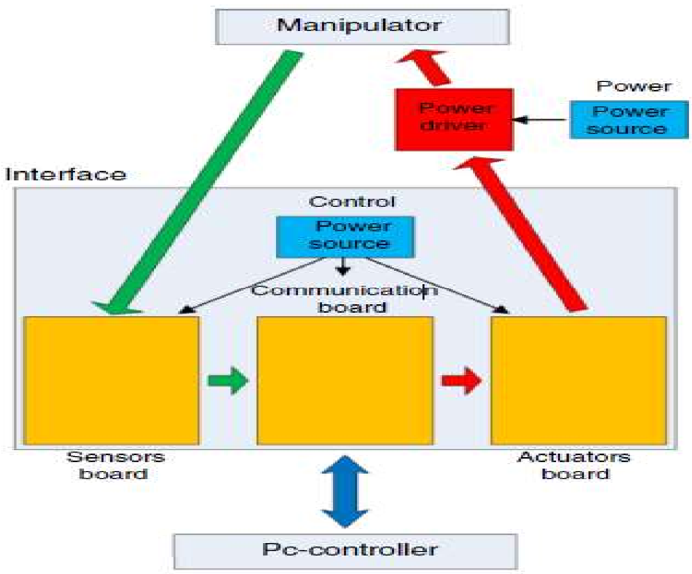


Figure 2.9. General electronic interface diagram (Urrea et al., 2016)

Figure 2.9. shows the general electronic interface diagram. The interface is responsible for the connection between the pc controller and the robot, converting the

PC controller numerical data into actuator signals, and converting the signals from the robot sensors into numbers that will be sent to the PC controller (Urrea et al., 2016).

In a study disturbance observer is used to control position nd force, and the SCARA robot is used to draw the parabola on the board with a pen at the end effector. The results show successful SCARA robot position and force controls without using a force sensor (Surapong & Mitsantisuk, 2016). Figure 2.10 shows the Structure and Joint of the SCARA robot used.

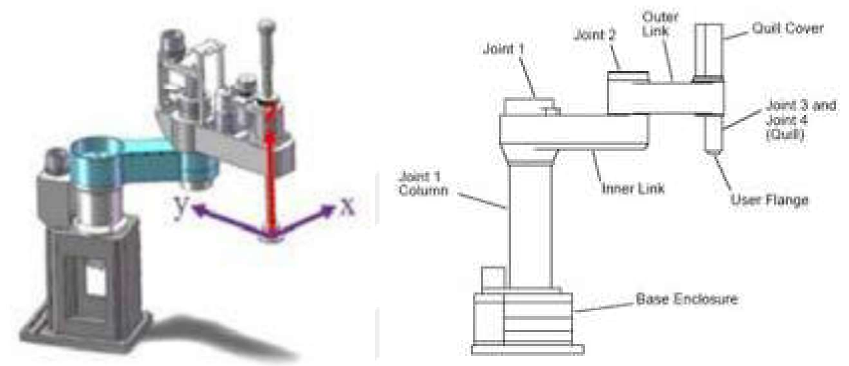


Figure 2.10. The structure and joints of the SCARA robot (Surapong & Mitsantisuk, 2016)

The SCARA robot structure shown in Fig. 2.10 consists of 4 interconnected joints. Joints 1,2 and 3 moves in the x-y plane, and joint four moves in the z-axis.

The SCARA robot is used to drive in the required direction. The disturbance force starts on the 5th second as the robot moves.

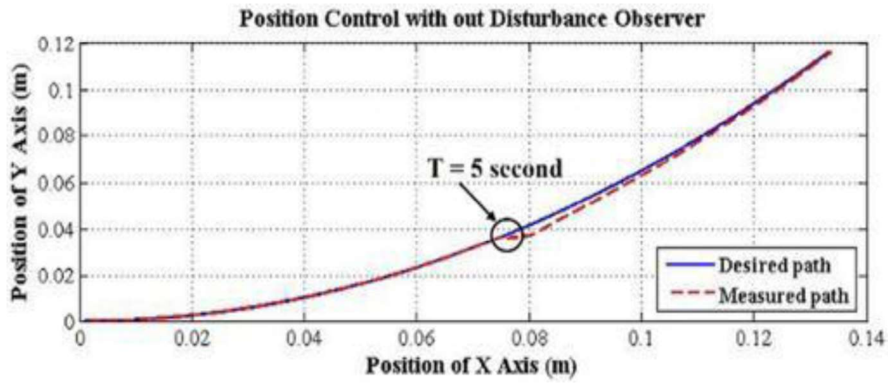


Figure 2.11. Position control without DOB (Surapong & Mitsantisuk, 2016)

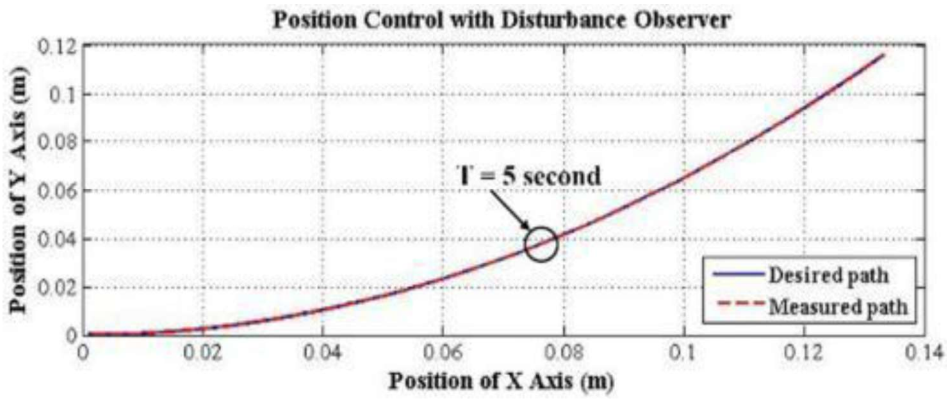


Figure 2.12. Position control with DOB (Surapong & Mitsantisuk, 2016)

Figure 2.12 shows Position control with DOB. The robot location has a fault in movement when the disturbance is triggered. It will step in the desired direction afterward. In Figure 2.11, the robot's movement is shown without DOB, where the force of disturbance is not compansated for.

3. STRUCTURE AND MATHEMATICAL ANALYSIS

Every company manufactures SCARA robots with various features, but the basic structure is almost identical. It is like a human arm with a shoulder, wrist, and elbow. The arrangement of the two links and four axes makes four degrees of freedom. In X-Y-Z space, two parallel rotary connections and a linear vertical joint allow independence. The end effector's rotational motion gives the fourth degree of freedom on the vertical axis. A strong base is used to support the structure. Figure 3.1. shows the structure of SCARA robot arm used in these theses.

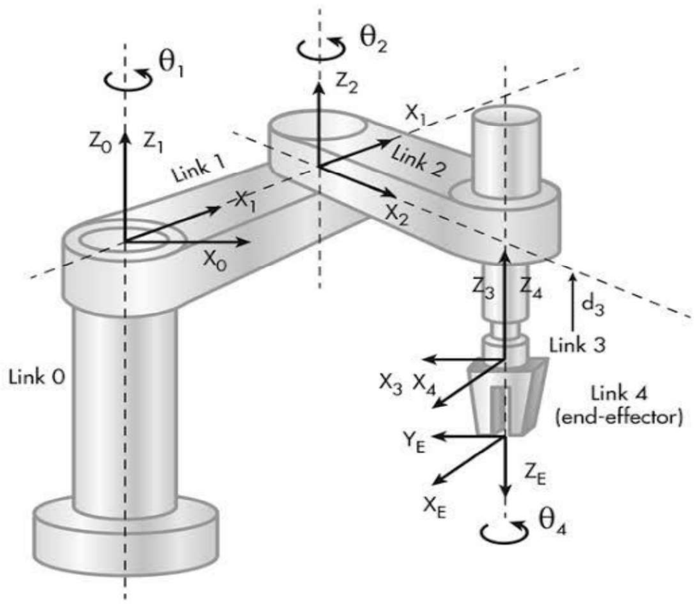


Figure 3.1. SCARA robot arm.

This chapter analyzes the kinematics such as forward, inverse, Jacobean, dynamics equation by using Lagrange formulation, PD Controller by using PD inverse dynamic control, Trajectory by using quantity polynomial trajectory. The disturbance observer equations are derived, and in the last section, all equations and matrices are implemented in MatLab programming.

3.1. Kinematics equations

The kinematics is the description of motion regardless of force on the robot. It explains the position, speed, and acceleration.

Robot manipulator's kinematics solutions are divided into two parts. The first type is the forward kinematics (Poveda-Bautista, Diego-Mas, & Leon-Medina, 2018), which specifies the end effector of the robot's manipulator in case all joints are known. The second type is inverse kinematics, which is related to measuring the variable joint angles. In this case, the desired position and orientation of the end-effector are known. The transformation from joint space to cartesian space is described by forward kinematics, while the conversion from cartesian space to joint space is described by inverse kinematics.

Figure 3.2 illustrates the joints and links for the SCARA robot arm from the base frame to the end-effector with DH parameters.

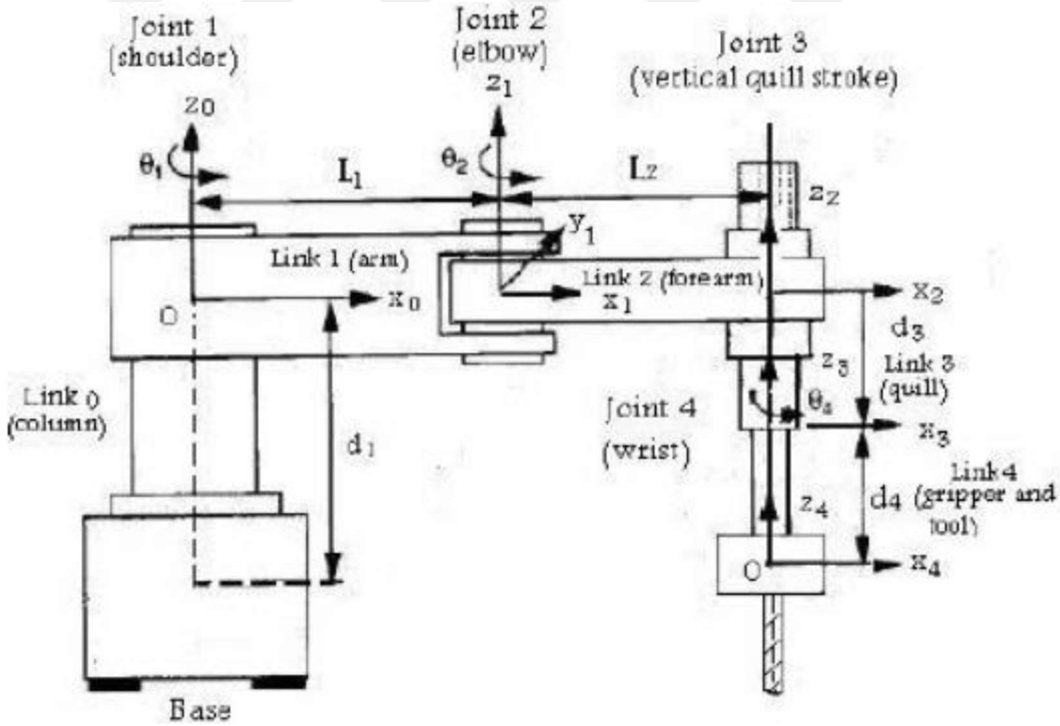


Figure 3.2. D-H Parameters for four- joint SCARA robot (Manasra et al., 2018)

3.2. Forward Kinematics

The Forward Kinematics aims to determine the position and orientation of the end effector according to the related base frame variables, angles, and distance.

We should find DH parameters to calculate the forward kinematics equation for the location and orientation of the end-effector. We have four kinematic parameters: link, angle, link twist, and joint distance. The first number represents the angle θ_i, the second number represents the length a_i, the third number represents the angle α_i, and the fourth number represents the distance d_i.

In detail:

- a_i is the distance from z_i to z_{i+1}, measured along z_i.
- α_i is the angle between z_i and z_{i+1}, measured about x_i.
- d_i is the distance from x_i to x_{i+1} measured along z_i.
- θ_i is the angle between x_i and x_{i+1} measured along z_i.

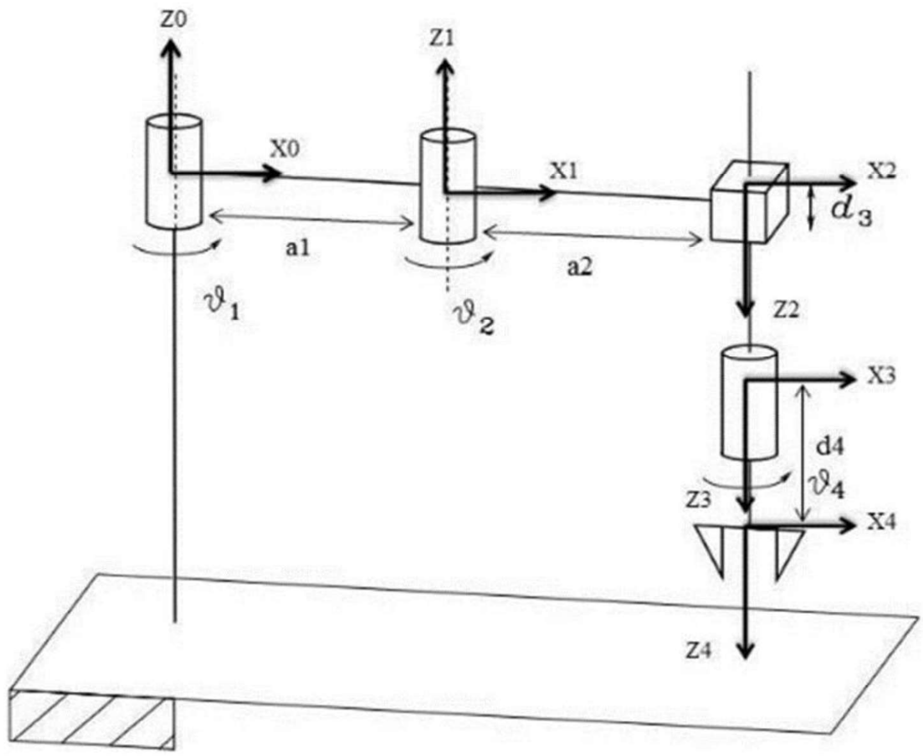


Figure 3.3. SCARA robot joint frames.

Table 3.2. illustrates D-H parameters of SCARA robot.

Table 3.1: DH parameters for the SCARA robot.

	θ_i	d_i	a_i	α_i
1	θ_1	0	a_1	0
2	θ_2	0	a_2	π
3	0	d_3	0	0
4	θ_4	d_4	0	0

In order to calculate D-H parameters of the end effector, the homogeneous transformation matrixes are calculated:

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 \tag{3.1}$$

T_1^0 : The homogeneous transformation matrix of the first joint.

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

T_2^1 : The homogeneous transformation matrix of the second joint

$$T_2^1 = \begin{bmatrix} c_2 & s_2 & 0 & a_2c_2 \\ s_2 & -c_2 & 0 & a_2s_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

T_3^2 : The homogeneous transformation matrix of the third joint

$$T_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

T_4^3 : The homogeneous transformation matrix of the fourth joint

$$T_4^3 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

T_4^0 : The homogeneous transformation matrix from fourth to first joint

$$T_4^0 = \begin{bmatrix} c_{12-4} & s_{12-4} & 0 & a_2c_{12} + a_1c_1 \\ s_{12-4} & -c_{12-4} & 0 & a_2s_{12} + a_1s_1 \\ 0 & 0 & -1 & -d_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Where:

$c_1 = cos\theta_1$,

$s_1 = sin\theta_1$,

$c_2 = cos\theta_2$,

$s_2 = sin\theta_2$,

$c_{12} = \cos(\theta_1 + \theta_2)$,

$s_{12} = \sin(\theta_1 + \theta_2)$,

$c_{12-4} = \cos(\theta_1 + \theta_2 - \theta_4)$

$s_{12-4} = \sin(\theta_1 + \theta_2 - \theta_4)$.

Hence:

x= $a_2 \cos(\theta_1 + \theta_2) + a_1 cos\theta_1$.

y= $a_2 \sin(\theta_1 + \theta_2) + a_1 sin\theta_1$.

z= $-d_3 + d_4$.

3.3. Inverse Kinematics

The Inverse Kinematics (IK) defines the joint angles for the desired position and orientation of the Cartesian space in order to calculate the angles of the robot by using a complete transformation matrix. IK is more a difficult and complicated problem than forward Kinematics. Each robot manipulator requires a specific process, taking into account the restrictions and structure of the device.

The joint variables in this SCARA robot are θ_1, θ_2, d_3 and θ_4. The vectors for prismatic and rotation matrixes give the desired end-effector position and orientation in the c cartesian space R_4^0, respectively.

For Prismatic joint:

$$\text{P}=\begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix} \tag{3.7}$$

For Rotation joint:

$$R_4^0 = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{3.8}$$

The specified desired homogenous transformation is given by:

$$T_4^0 = \begin{bmatrix} c_{12-4} & s_{12-4} & 0 & a_2c_{12} + a_1c_1 \\ s_{12-4} & -c_{12-4} & 0 & a_2s_{12} + a_1s_1 \\ 0 & 0 & -1 & -d_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

The inverse kinematics problem is given by:

$$\begin{bmatrix} \mathrm{c}_{12-4} & \mathrm{s}_{12-4} & 0 & \mathrm{a}_2\mathrm{c}_{12} + \mathrm{a}_1\mathrm{c}_1 \\ \mathrm{s}_{12-4} & -\mathrm{c}_{12-4} & 0 & \mathrm{a}_2\mathrm{s}_{12} + \mathrm{a}_1\mathrm{s}_1 \\ 0 & 0 & -1 & -\mathrm{d}_3 + \mathrm{d}_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathrm{R}_{11} & \mathrm{R}_{12} & \mathrm{R}_{13} & \mathrm{P}_\mathrm{x} \\ \mathrm{R}_{21} & \mathrm{R}_{22} & \mathrm{R}_{23} & \mathrm{P}_\mathrm{y} \\ \mathrm{R}_{31} & \mathrm{R}_{32} & \mathrm{R}_{33} & \mathrm{P}_\mathrm{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

The solution to d_3 from joint three is given by:

$$d_3 = d_4 - P_z \tag{3.11}$$

The solutions θ_2 from joint two is given by:

$$c_2 = \frac{P_x^2 + P_y^2 - (a_1^2 + a_2^2)}{2a_1a_2} \tag{3.12}$$

$$s_2 = \sqrt{1 - c_2^2} \tag{3.13}$$

$$\theta_2 = Atan2(s_2, c_2) \tag{3.14}$$

The solutions to θ_1 from joint one is given by:

$$s_1 = \frac{(a_1 + a_2c_2)P_y - (a_2s_2)P_x}{{P_x}^2 + {P_y}^2} \tag{3.15}$$

$$\theta_1 = \sin^{-1} s_1 = \sin^{-1} \frac{(a_1+a_2c_2)P_y-(a_2s_2)P_x}{{P_x}^2+{P_y}^2} \qquad (3.16)$$

The solutions to θ_4 from joint four is given by:

$$\beta = \theta_1 + \theta_2 - \theta_4 \qquad (3.17)$$

$$\theta_4 = \theta_1 + \theta_2 - \beta \qquad (3.18)$$

β is the orientation angle of the end-effector around z_4 with respect to the base frame.

3.4. Differential kinematics and statics

The Jacobian method is one of the most important solution methods in the study to control the motion of the SCARA robot arm. For the execution of trajectory and smooth planning within the dynamic equations, the Jacobian method is used. In order to check the target at the specified velocity, each joint velocity must be detected at the specified joint positions. This is done by the Jacobian method, which relates the joint velocity to the linear and angular velocity of the end effector. The Jacobian method explains the relationship between the linear, angular velocity, and joint velocity of the end effector.

The number of DOF of the manipulator represents the number of columns of the Jacobian matrix. There are always three rows for linear velocity in the directions x, y, z, and three rows for angular velocity. Therefore the Jacobian method is a four by four square matrix for a four DOF manipulator. From the following equations, we can calculate the Jacobian matrix :

Angular velocity :

$$j_\omega = [P_1Z_0 \quad P_2Z_1 \quad P_3Z_2 \quad P_4Z_3] \qquad (3.19)$$

P = 0; Prismatic.

P = 1; Revaluate

Linear velocity:

$$j_{v_i} = Z_{i-1} \text{ , prismatic} \qquad (3.20)$$

$$j_{v_i} = Z_{i-1} \times (O_n - O_{i-1}) \text{ Revolution} \qquad (3.21)$$

Where: J_ω and j_v the angular and linear velocity of the SCARA robot arm, respectively, and the matrix of Jacobian is 6x4.

The matrix of Jacobian is given by:

$$J_{(q)} = [J_1 \quad J_2 \quad J_3 \quad J_4] \qquad (3.22)$$

$$J_{(q)} = \begin{bmatrix} z_0 \times (O_4 - O_0) & z_1 \times (O_4 - O_1) & Z_2 & z_3 \times (O_4 - O_3) \\ z_0 & z_1 & 0 & z_3 \end{bmatrix} \qquad (3.23)$$

From the homogenous transformation computation:

T_1^0 : The homogeneous transformation matrix from the first joint with respect to the base frame

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.24)$$

T_2^0 : The homogeneous transformation matrix from the second joint with respect to the base frame

$$T_2^0 = \begin{bmatrix} c_{12} & s_{12} & 0 & a_2 c_{12} + a_1 c_1 \\ s_{12} & -c_{12} & 0 & a_2 s_{12} + a_1 s_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.25)$$

T_3^0 : The homogeneous transformation matrix from the third joint with respect to the base frame

$$T_3^0 = \begin{bmatrix} c_{12} & s_{12} & 0 & a_2 c_{12} + a_1 c_1 \\ s_{12} & -c_{12} & 0 & a_2 s_{12} + a_1 s_2 \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.26)$$

T_4^0 : The homogeneous transformation matrix from the first joint with respect to the base frame

$$T_4^0 = \begin{bmatrix} c_{12-4} & s_{12-4} & 0 & a_2c_{12} + a_1c_1 \\ s_{12-4} & -c_{12-4} & 0 & a_2s_{12} + a_1s_1 \\ 0 & 0 & -1 & -d_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

O_0: The reference frame of the original

$$O_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.28)$$

O_1: The fourth column of the matrix T_1^0

$$O_1 = \begin{bmatrix} a_1c_1 \\ a_1s_1 \\ 0 \end{bmatrix} \quad (3.29)$$

O_2: The fourth column of the matrix T_2^0

$$O_2 = \begin{bmatrix} a_2c_{12} + a_1c_1 \\ a_2s_{12} + a_1s_1 \\ 0 \end{bmatrix} \quad (3.30)$$

O_3: The fourth column of the matrix T_3^0

$$O_3 = \begin{bmatrix} a_2c_{12} + a_1c_1 \\ a_2s_{12} + a_1s_1 \\ -d_3 \end{bmatrix} \quad (3.31)$$

O_4: The fourth column of the matrix T_4^0

$$O_4 = \begin{bmatrix} a_2c_{12} + a_1c_1 \\ a_2s_{12} + a_1s_1 \\ d_4 - d_3 \end{bmatrix} \quad (3.32)$$

Moreover, we can find:

Z_0: Defined in the base reference frame

$$Z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.33)$$

Z_1: The first three elements of the third column of the matrix T_1^0

$$Z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.34)$$

Z_2: The first three elements of the third column of the matrix T_2^0

$$Z_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \qquad (3.35)$$

Z_3: The first three elements of the third column of the matrix T_3^0

$$Z_3 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \qquad (3.36)$$

We can write the angular velocity:

$$P_1 = 1,\ P_2 = 1,\ P_3 = 0,\ P_4 = 1$$

$$J_\omega = [P_1 Z_0 \quad P_2 Z_1 \quad P_3 Z_2 \quad P_4 Z_3] \qquad (3.37)$$

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

j_{v1}: The first linear velocity of Jacobian

$$j_{v1} = Z_0 \times (O_4 - O_0) \qquad (3.38)$$

$$= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} a_2 c_{12} + a_1 c_1 \\ a_2 s_{12} + a_1 s_1 \\ -d_3 + d_4 \end{bmatrix} = \begin{bmatrix} -(a_2 s_{12} + a_1 s_1) \\ a_2 c_{12} + a_1 c_1 \\ 0 \end{bmatrix}$$

j_{v2}: The second linear velocity of Jacobian

$$j_{v2} = Z_1 \times (O_4 - O_1) \qquad (3.39)$$

$$= \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} a_2 c_{12} \\ a_2 s_{12} \\ -d_3 + d_4 \end{bmatrix} = \begin{bmatrix} a_2 s_{12} \\ -a_2 c_{12} \\ 0 \end{bmatrix}$$

j_{v3}: The third linear velocity of Jacobian

$$j_{v3} = Z_2 \qquad (3.40)$$

$$= \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

j_{v4}: The fourth linear velocity of Jacobian

$$j_{v4} = Z_3 \times (O_4 - O_3) \qquad (3.41)$$

$$= \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ d_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

J_v: The linear velocity of Jacobian

$$J_v = \begin{bmatrix} a_2 s_{12} + a_1 s_1 & -a_2 s_{12} & 0 & 0 \\ a_2 c_{12} + a_1 c_1 & a_2 c_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.42)$$

Hence,

$J_{(q)}$: The Jacobian matrix

$$J_{(q)} = \begin{bmatrix} J_V \\ J_\omega \end{bmatrix} \qquad (3.43)$$

$$\mathbf{J}\ (q) = \begin{bmatrix} -(a_2 s_{12} + a_1 s_1) & a_2 s_{12} & 0 & 0 \\ a_2 c_{12} + a_1 c_1 & -a_2 c_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix}$$

3.5. Dynamics model

The derivation of the dynamics model plays a major role in the simulation of motion, design of control algorithms, and manipulator structure analysis. The dynamic analysis can be used for the mechanical design of the robot arm. The measurement forces and torques are necessary to perform model movement and provide valuable information to design the actuators, transmissions, and joints. Lagrange formulation can be used with a simple systematic derivation from this complex system. In this thesis, the Lagrange method is used to derive the dynamics equations of the SCARA robot arm.

The manipulator controller is based on the dynamic model of the robot. The dynamics model describes the relationship between the torques from the joints actuators and the structure motion. To derive the motion equations, there are some methods, such as the formulation of Newton-Euler and Lagrange. The general equation of motion is shown in Eq(3.44); where M(θ) is a mass matrix, C(θ,$\dot{\theta}$) is Coriolis matrix, G(θ) is gravity vector, and torque matrix $\tau(\theta)$.

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau(\theta) \qquad (3.44)$$

3.5.1. Lagrange Formulation

The dynamic model describes the relationship between the structure of motion and the torques from the joints actuators. The Lagrange Formulation of the mechanical system, in general, can be defined as follows:

$$L\ (\theta,\dot{\theta})\ =K\ (\theta,\dot{\theta})\ –P\ (\theta) \qquad (3.45)$$

Where K is the total kinetic energy, and P is the total potential energy of the robot system. The total kinetic energy of the manipulators can be found by calculating the kinetic energy for each motor.

Lagrange equation of motion is given by :

$$\frac{d}{dt}\frac{\partial L}{\partial_{\dot{q}_i}} - \frac{\partial L}{\partial_{q_i}} = T_i \qquad i = 1,2,3,4. \qquad (3.46)$$

Where T_i is the torque for the link i.

The total kinetic energy for link K_i in general is as follows:

$$K_i = \frac{1}{2}m_{li}\dot{p}_{li} + \frac{1}{2}J_{li}\omega_i \qquad (3.47)$$

Where

m_{li}: The mass of joint *i*

$\dot{p}_{li}$: The velocity of joint *i*

J_{li}: The Moment of Inertia

$I = \frac{1}{3}miL^2$

ω_i: The angular velocity of joint *i*

For Link i the potential energy is given by:

$$U_{li} = -m_{li}gP_{li} \qquad (3.48)$$

Where:

m_{li}: The mass of joint *i*

g : The gravitation of acceleration $= 9.8\frac{m}{s^2}$

P_{li}: The displacement of joint i

For the SCARA robot, the torques exerted on the robot joints are:

T_1: The torque from the first joint:

$$T_1 = (m_{l1}l_1^2 + m_{l2}(l_2^2 + a_1^2 + 2l_2a_1\cos\theta_2) + m_{l3+l4}(a_2^2 + a_1^2 + 2a_2a_1\cos\theta_2) + J_{l1}+J_{l2} + J_{l4})\,\ddot{\theta}_1 + (m_{l2}(l_2^2 + l_2a_1\cos\theta_2) + m_{l3+l4}(a_2^2 + a_2a_1\cos\theta_2) ++J_{l2} + J_{l4})\,\ddot{\theta}_2 - J_{l4}\ddot{\theta}_4 - (m_{l2}l_2a_1\sin\theta_2 + m_{l3+l4}a_1a_2\sin\theta_2)\,\dot{\theta}_1\dot{\theta}_2 - (m_{l2}l_2a_1\sin\theta_2 + m_{l3+l4}a_1a_2\sin\theta_2)\,(\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2 \qquad (3.49)$$

T_2: The torque from the second joint:

$$T_2 = (m_{l2}(l_2^2 + l_2a_1\cos\theta_2) + m_{l3+l4}(a_2^2 + a_2a_1\cos\theta_2) ++J_{l2} + J_{l4})\,\ddot{\theta}_1 + (m_{l2}l_2^2 + m_{l3+l4}a_2^2 ++J_{l2} + J_{l4})\,\ddot{\theta}_2 - J_{l4}\ddot{\theta}_4 + (m_{l2}l_2a_1\sin\theta_2 + m_{l3+l4}a_1a_2\sin\theta_2)\,\dot{\theta}_1^2 \qquad (3.50)$$

T_3: The torque from the third joint:

$$T_3 = (m_{l3} + m_{l4})\ddot{d}_3 - (m_{l3} + m_{l4})g \qquad (3.51)$$

T_4: The torque from the fourth joint:

$$T_4 = -J_{l4}\ddot{\theta}_1 - J_{l4}\ddot{\theta}_2 + J_{l4}\ddot{\theta}_4 \qquad (3.52)$$

After applying the Lagrange formulation and rearranging the terms, the following can be given:

M (θ): Mass matrix

$$\mathrm{M}\,(\theta) = \begin{bmatrix} M_{11} & M_{12} & 0 & M_{14} \\ M_{21} & M_{22} & 0 & M_{24} \\ 0 & 0 & M_{33} & 0 \\ M_{41} & M_{42} & 0 & M_{44} \end{bmatrix} \qquad (3.53)$$

Where:

$$M_{11} = m_{l1}l_1^2 + m_{l2}(l_2^2 + a_1^2 + 2l_2a_1\cos\theta_2) + m_{l3+l4}(a_2^2 + a_1^2 + 2a_2a_1\cos\theta_2) + J_{l1}+J_{l2} + J_{l4}$$

$$M_{12} = M_{21} = m_{l2}(l_2^2 + l_2a_1\cos\theta_2) + m_{l3+l4}(a_2^2 + a_2a_1\cos\theta_2) ++J_{l2} + J_{l4}$$

$$M_{22} = m_{l2}l_2^2 + m_{l3+l4}a_2^2 ++J_{l2} + J_{l4}$$

$$M_{14} = M_{41} = -J_{l4}$$

$$M_{24} = M_{42} = -J_{l4}$$

$$M_{33} = m_{l3} + m_{l4}$$

$M_{44} = J_{l4}$

C(θ,$\dot{\theta}$): Coriolis matrix

$$C(\theta,\dot{\theta}) = \begin{bmatrix} B\dot{\theta}_2 & B(\dot{\theta}_1 + \dot{\theta}_2) & 0 & 0 \\ -B\dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.54)$$

$$B = -m_{l2} l_2 a_1 \sin\theta_2 + m_{l3+l4} a_1 a_2 \sin\theta_2$$

G(θ): Gravity vector

$$G(\theta) = \begin{bmatrix} 0 \\ 0 \\ -(m_{l3} + m_{l4})g \\ 0 \end{bmatrix} \quad (3.55)$$

$$g = 9.8 \frac{m}{s^2}$$

Here, all the frictions are neglected.

3.6. PD inverse dynamics controller

3.6.1. Introduction to PD controller

The Proportional Derivative control is used to increase the response signal in the control system. The integral gives the control error with a fast response. The proportional is useful to increase the response signal for stable systems.

Integral or Proportional term:

The integral term is calculated by a value of output proportional to the current value of error. To adjust the proportional response, the error value is multiplied with a constant K_p, K_p called the proportional gain constant (Swain, Behera, Devendra Singh Mondavi, 2012).

The integral term is given by:

$$U(t) = K_p e(t) \quad (3.56)$$

When the proportional gain is high, this leads to a change in the output for a given change in the error. When the proportional gain is too high it may cause the system to become unstable (Datar, 2015). When the proportional gain is small, this results in a small output response, and then the input error is large. Therefore, the controller becomes less sensitive and less responsive. When the proportional gain is too low, the control response is too small. (Wasnik & Ghutke, 2018).

Derivative term:

The derivative term of process error can be found by calculating the slope of the error over time and multiplying this change with the derivative gain. The derivative gain K_d is the amount of contribution to the overall control of the system.

The derivative control can be described as follows:

$$u(t) = K_p \mathrm{e}(t) + K_d \frac{d}{dt} e(t) \qquad (3.57)$$

The derivative control is used to slow the rate of change in the output of the controller. It is also used to increase the stability in the controller, slow the controller's response, and decrease the overshoot's magnitude by the integral component. The derivative term is highly sensitive to the controller and noise in terms of error. The system can become unstable if the noise and derivative gain are large enough. Therefore, it is more common to use a differentiator with restricted bandwidth. This circuit is generally known as a phase-lead compensator (Swain et al., 2012).

3.6.2. PD inverse dynamics control

As its name indicates, the inverse dynamics control is directly linked to the solution of the inverse dynamics problem considered in analytical mechanics. Having the specified motion and the desired properties of the resulting systems, the control inputs that ensure the realization of these control objectives are to found. By appropriately inverting the plant's dynamic model to be controlled, a control law can then be constructed, which cancels the nonlinear part of the dynamics, decouples the interactions between the regulated variables, and specifies the time characteristics to the drop of the task errors. The solution of inverse dynamics is a completely satisfying approach to control design for both linear and nonlinear systems because it is natural and correct (Jankowski, 2004).

The suitable linearization of the system's dynamics can be done by splitting the control algorithm into two parts. The first one is the nonlinear compensation and decoupling, and the other one is a linear stabilizing control, as shown in Fig.3.6.

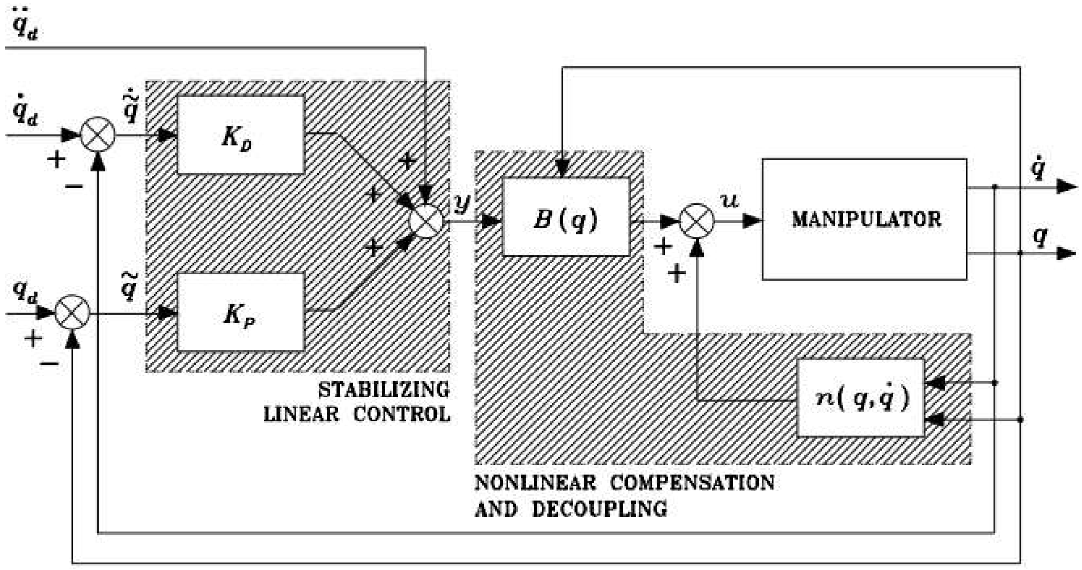


Figure.3.4. Inverse dynamics control ("Advanced Textbooks in Control and Signal Processing," 2018)

Figure 3.4. illustrates that there are two feedback loops: the first one is the internal loop based on the dynamic manipulator. The second one is an external loop-based on error tracking. The first feedback, the "internal loop," is used to achieve a linear and decoupling relationship between input and output. The second feedback, the "external loop," is used for the stability of the system. The architecture in the external loop controller is simple because it works with a linear and constant system time. For the implementation of PD inverse dynamic control, the B(θ), the Mass matrix, C($\theta, \dot{\theta}$), the Coriolis matrix and G(θ), the Gravity matrix, and the damping $n(\theta, \dot{\theta})$ must be calculated ("Advanced Textbooks in Control and Signal Processing," 2018). The calculation of PD inverse dynamic equations is as follows:

Let the vector control *u* to be a function of the robot states, which can define as

$$u = B(\theta)y + n(\theta, \dot{\theta}) \qquad (3.58)$$

Where

$$n(\theta, \dot{\theta}) = C(\theta, \dot{\theta}) + G(\theta) \qquad (3.59)$$

Equations (3.58) and (3.59) represent the nonlinear control law, and linear and decoupled systems with respect to the input *y*, respectively. Now *y* can be calculated as follows:

$$y = \theta_d + K_D\dot{\theta} + K_P\theta \qquad (3.60)$$

Where K_D *and* K_P are the proportional derivative controller gains. The controller gains should be a positive value for the robot system to be asymptotically stable. K_D *and* K_P are calculated by using manual tuning as and then used in diagonal matrices. The manual tuning is realized by increasing the K_D *and* K_P values until the system start to oscillate. We need the system to reach the fastest response without oscillation. When K_D *and* K_P values are high, the system will be more stable.

The control law can be written as follows,

$$\tau = M(\theta)(\ddot{\theta} + K_D\dot{e} + K_P e) + C\big(\theta, \dot{\theta}\big)\dot{\theta} + G(\theta) \qquad (3.61)$$

Where

M: Mass matrix.

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} & 0 & M_{14} \\ M_{21} & M_{22} & 0 & M_{24} \\ 0 & 0 & M_{33} & 0 \\ M_{41} & M_{42} & 0 & M_{44} \end{bmatrix} \qquad (3.62)$$

C: Coriolis matrix. :

$$C(\theta, \dot{\theta}) = \begin{bmatrix} B\dot{\theta}_2 & B(\dot{\theta}_1 + \dot{\theta}_2) & 0 & 0 \\ -B\dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.63)$$

G: Gravity matrix:

$$G(\theta) = \begin{bmatrix} 0 \\ 0 \\ -(m_{l3} + m_{l4})g \\ 0 \end{bmatrix} \qquad (3.64)$$

K_P : Propotinal controller gain:

$$K_P = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \qquad (3.65)$$

K_D: Derivative controller gains:

$$K_D = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \qquad (3.66)$$

e :Position tracking error

$$e = \theta_{ref} - \theta_{des} \qquad (3.67)$$

$\dot{e}$: Angular speed tracking error:

$$\dot{e} = \omega_{ref} - \omega_{des} \qquad (3.68)$$

3.7. Trajectory planning

This section's main objective is to find a path that connects between the initial and final configuration with other conditions specified in the endpoints as velocity and acceleration. Therefore we are going to deal with the problem of θ(t), the scalar joint angle variable.

Let θ(t) at the time t_0

$$\theta(t_0) = \theta_0 \qquad (3.69)$$

$$\dot{\theta}(t_0) = v_0 \qquad (3.70)$$

Let θ(t) at the time t_f

$$\theta(t_f) = \theta_f \qquad (3.71)$$

$$\dot{\theta}(t_f) = v_f \qquad (3.72)$$

We have two additional equations

$$\ddot{\theta}(t_0) = a_0 \qquad (3.73)$$

$$\ddot{\theta}(t_f) = a_f \qquad (3.74)$$

A class of polynomial functions approximates the desired path. The polynomial functions produce a time-based sequence "control setpoints" to control the manipulator from the initial point to the desired position. Figure 3.7 Illustrates the trajectory planning block diagram.

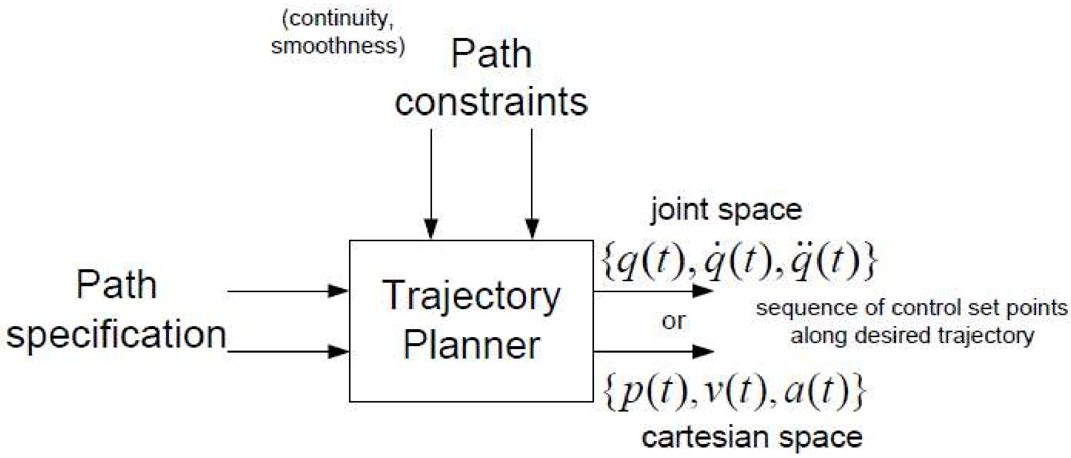


Figure 3.5. Trajectory planning block diagram

3.7.1. Polynomial Trajectories

In order to construct a trajectory between two points and determine the beginning and end speeds of the trajectory, there is one way to create a smooth curve, a function polynomial of time. There are three types of polynomial trajectories: cubic, quantic, and basis spline polynomials. If we have six conditions to be met, we need a polynomial function with independent fife coefficients to fulfill these conditions. Therefore, in this thesis, the Quantic Polynomial Trajectories type is used.

3.7.2. Quantic Polynomial Trajectories

We have six parameters in this case (each one for the initial and final configurations, initial and final velocities, initial and final accelerations). Therefore we need to have a polynomial of the fifth-order.

$$\theta(\mathrm{t}) = \mathrm{a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0} \qquad \text{For Distance} \qquad (3.75)$$

$$\dot{\theta}(\mathrm{t}) = \mathrm{5a_5 t^4 + 4a_4 t^3 + 3a_3 t^2 + 2a_2 t + a_1} \qquad \text{For Velocity} \qquad (3.76)$$

$$\ddot{\theta}(\mathrm{t}) = \mathrm{20a_5 t^3 + 12a_4 t^2 + 6a_3 t + 2a_2} \qquad \text{For Acceleration} \qquad (3.77)$$

We have six conditions:

$$e(t_i) = 0$$

$e(t_f) = \theta_f$

$\dot{e}(t_i) = 0$ (3.78)

$\dot{e}(t_f) = 0$

$\ddot{e}(t_i) = 0$

$\ddot{e}(t_f) = 0$

After substituting the values in the equations as follows:

$$e_i = \theta(t_i) = a_5 t_i^5 + a_4 t_i^4 + a_3 t_i^3 + a_2 t_i^2 + a_1 t_i + a_0 \quad (3.79)$$

$$e_f = \theta(t_f) = a_5 t_f^5 + a_4 t_f^4 + a_3 t_f^3 + a_2 t_f^2 + a_1 t_f + a_0 \quad (3.80)$$

$$\dot{e_\iota} = \dot{\theta}(t_i) = 5a_5 t_i^4 + 4a_4 t_i^3 + 3a_3 t_i^2 + 2a_2 t_i + a_1 \quad (3.81)$$

$$\dot{e_f} = \dot{\theta}(t_f) = 5a_5 t_f^4 + 4a_4 t_f^3 + 3a_3 t_f^2 + 2a_2 t_f + a_1 \quad (3.82)$$

$$\ddot{e_\iota} = \ddot{\theta}(t_i) = 20a_5 t_i^3 + 12a_4 t_i^2 + 6a_3 t_i + 2a_2 \quad (3.83)$$

$$\ddot{e_f} = \ddot{\theta}(t_f) = 20a_5 t_f^3 + 12a_4 t_f^2 + 6a_3 t_f + 2a_2 \quad (3.84)$$

$$\begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 1 & t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} e_i \\ e_f \\ \dot{e_\iota} \\ \dot{e_f} \\ \ddot{e_\iota} \\ \ddot{e_f} \end{bmatrix} \quad (3.85)$$

Then from e_i=0 , $\dot{e_\iota}$=0 , $\ddot{e_\iota}$=0 and $t_\iota = 0$,

$$a_0=0 ,\ a_1=0 ,\ a_2=0. \quad (3.86)$$

After finding the quantic trajectory with $t_i = 0$ and $t_f = 1$ with zero initial and final velocities, zero initial and final accelerations, the arm of the robot will move to the end position. We can see the curve of the quantic trajectory, as shown in Figures 3.8., 3.9, 3.10. These figures show the relationship between angle, speed, and acceleration with time.

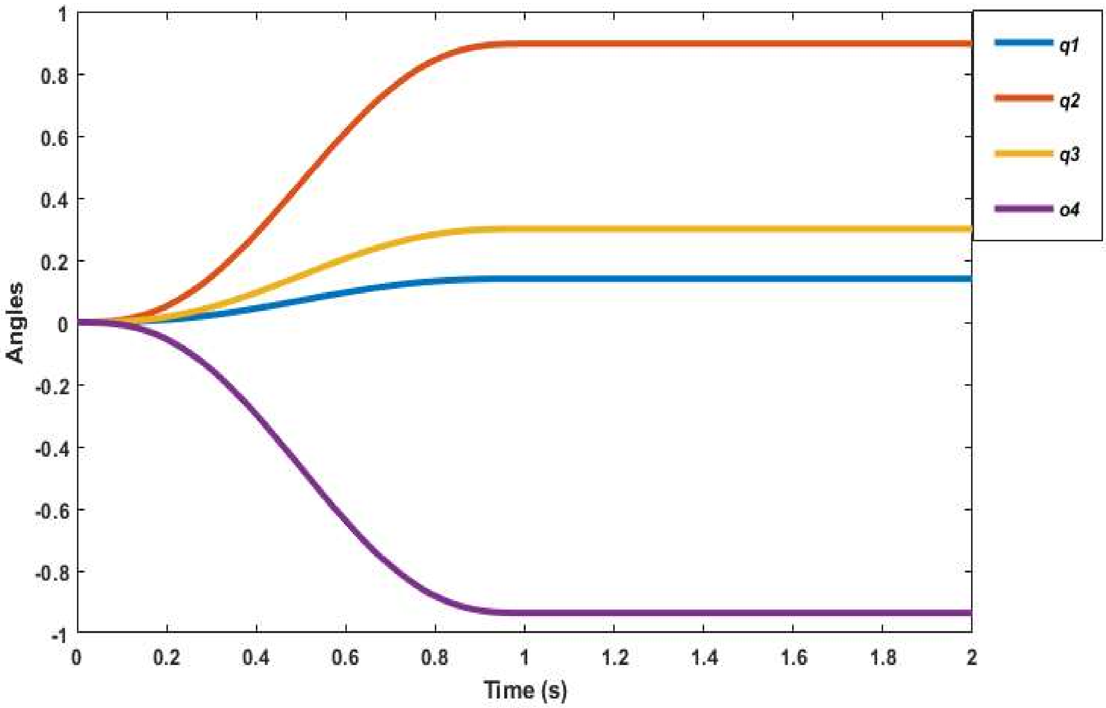


Figure 3.6. Angles with time quantic polynomial trajectory

Figure 3.6 shows the relationship between the angles and time equation polynomial trajectory. The initial time is zero, and the settling time is one for all joints. The first angle q_1 of the robot moves from zero to 0.1 during one second, and the second angle q_2 moves from zero to 0.9 during one second. The third angle q_3 moves from zero to 0.3 during one second, and the fourth angle q_4 moves from zero to -0.9 during one second. this figure shows that the system operates with no overshoot, zero steady states, smooth transition, and all the joints move in synchronization.

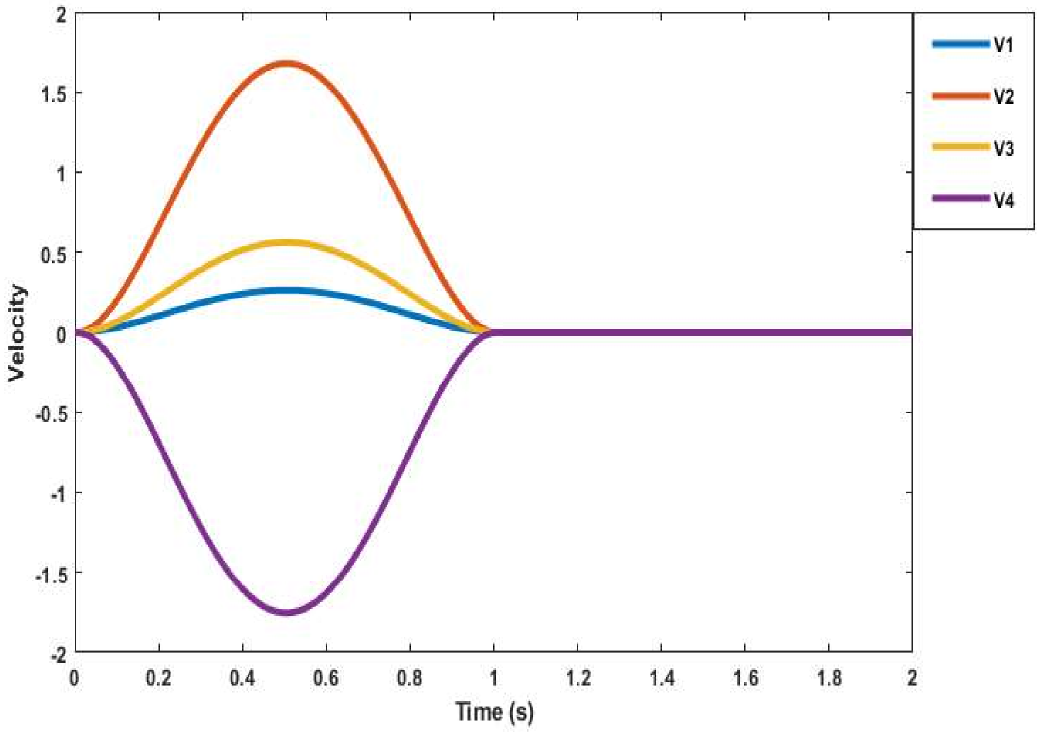


Figure 3.7. Velocity with time quantic polynomial trajectory

Figure 3.7 shows the relationship between the velocity and time equations polynomial trajectory, and it describes the initial velocity is zero, and the final velocity is zero during 1 second for all joints. The velocity of the joint one v_1 moves from zero initial to zero final during one second, the maximum overshoot value is 0.2, and the peak time is 0.5 seconds. The velocity of the joint two v_2 moves from zero initial to zero final velocity during one second, the maximum overshoot value is 1.6, and the peak time is 0.5 second. The velocity of the joint three v_3 moves from zero initial to zero final velocity during one second, the maximum overshoot value is 0.53, and the peak time is 0.5 second. The velocity of the joint four v_4 moves from zero initial to zero final during one second, the maximum overshoot value is -1.75, and the peak time is 0.5 seconds. The system needs one second to reach the steady-states.

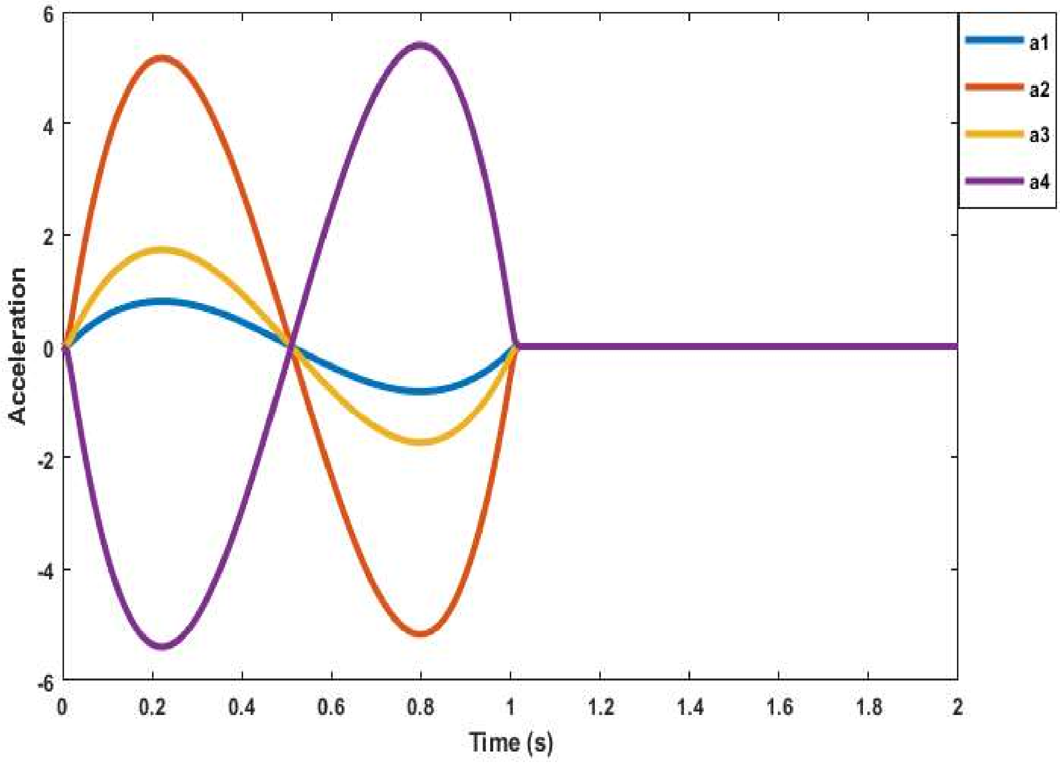


Figure 3.8. Acceleration with time quantic polynomial trajectory

Figure 3.8 shows the relationship between the acceleration and time equation polynomial trajectory. The initial acceleration is zero, and the final acceleration is zero. a_1 is the acceleration of the joint one, the maximum overshoot in the first joint is 0.9 during 0.25 seconds and the minimum overshoot is -0.9 during 0.75 seconds. a_2 is the acceleration of joint two, the maximum overshoot is 5 during 0.25 seconds, and the minimum overshoot is -5 during 0.75 seconds. a_3 is the acceleration of the joint three, the maximum overshoot is 1.9 during 0.25 seconds, and the minimum overshoot is -1.9 during 0.75 seconds. a_4 is the acceleration of the joint four, the minimum overshoot is -5.7 during 0.25 seconds, and the maximum overshoot is 5.7 during 0.75 seconds, the system needs one second to reach the steady-state.

3.8. Disturbance estimation

3.8.1. Disturbance estimation using DOB

The disturbance observer is used to estimate and overcome unknown disturbances affecting the control systems to improve stability and performance. In this section, the design of the disturbance observer is explained

The SCARA robot is affected by two types of disturbances:

1. Internal disturbance:

The Internal disturbance is formed due to the uncertainty in robot modeling; for example, frictions at the joints of the robot.

2. External disturbance:

Any external force affects the end effector of robot.

This thesis explains the design of the disturbance observer to suppress the external disturbance effect on the robot. To eliminate the effect of the disturbance, it should be estimated using an on-line observer, and then should be compensated in the controller.

The torque vector is described by the following equation:

$$\tau = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + \tau_d \qquad (3.87)$$

Where τ_d is the disturbance torque.

Then, the disturbance can be computed depending on (3.87) as:

$$\tau_d = A\,(\tau - (\,M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta)\,)) \qquad (3.88)$$

A is the observer gain, and it is used to return the disturbance to zero.

Finally, the applied torque equation can be described as the difference between the real torque in (3.87) and disturbance torque (3.88).

$$\tau_a = u - \tau_d \qquad (3.89)$$

Where

τ_a: Applied torque

τ_d: Disturbance torque

u: Real torque

In (3.89), the observer depends on the measured states of the robot to calculate the disturbance. These measurements can be done using an encoder and accelerometer sensor.

To apply this equation, the following points should be taken into consideration:

1. All states are measurable or observable, which means that the angular and linear velocity, acceleration, and position can be measured or observed.

2. Computation time of the observation process is less than the disturbance variation time.

3.8.2. Disturbance estimation using AKF

AKF is an adaptive observer that can be used to estimate states in linear systems and it supports auto-update for error variance and covariance matrices. The AKF is based on two steps: the first one is the prediction step, and the second one is the correction step. The prediction step uses control commands to estimate where the system will be at the next point. The correction step uses sensor observation to correct potential mistakes. The AKF makes two important assumptions: the first one is the gaussian, and the second one is all models are linear. The main advantage of this algorithm is that it does not require the noise covariance matrices, instead, it needs just initial values of them, and they will be updated depending on the behavior of error between the estimated states and the measured states.

Adaptive Kalman Filter was implemented based on the systematic estimating and updating steps listed as follows,

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (3.90)$$

$$P_k^- = AP_{k-1}A^T + Q_{k-1} \quad (3.91)$$

$$e_k = z_k - H\hat{x}_k^- \tag{3.92}$$

$$\alpha_1 = \frac{N_R - 1}{N_R} \tag{3.93}$$

$$\bar{e}_k = \alpha_1 \bar{e}_{k-1} + \frac{1}{N_R} e_k \tag{3.94}$$

$$\Delta R_k = \frac{1}{N_R - 1}(e_k - \bar{e}_k)(e_k - \bar{e}_k)^T - \frac{1}{N_R}(HP^- H^T)_k \tag{3.95}$$

$$R_k = |diag(\alpha_1 R_{k-1} + \Delta R_k)| \tag{3.96}$$

$$K_k = P_k^- H^T (HP_k^- H^T + R_k)^{-1} \tag{3.97}$$

$$\hat{x}_k = \hat{x}_k^- + K_k e_k \tag{3.98}$$

$$P_k = (I - K_k H)P_k^- \tag{3.99}$$

$$\omega_k = \hat{x}_k - \hat{x}_k^- \tag{3.100}$$

$$\alpha_2 = \frac{N_Q - 1}{N_Q} \tag{3.101}$$

$$\bar{\omega}_k = \alpha_2 \bar{\omega}_{k-1} + \frac{1}{N_Q} \widehat{\omega}_k \tag{3.102}$$

$$\Delta Q_k = \frac{1}{N_Q}(P_k - AP_{k-1}A^T) + \frac{1}{N_Q - 1}(\widehat{\omega}_k - \bar{\omega}_k)(\widehat{\omega}_k - \bar{\omega}_k)^T \tag{3.103}$$

$$Q_k = |diag(\alpha_2 Q_{k-1} + \Delta Q_k)| \tag{3.104}$$

where P_k is the estimation error covariance matrix, Q_k is the process noise covariance matrix, K_k is the Kalman gain, α_1, α_2, N_R and N_Q are positive constants, $\Delta R_k, \Delta Q_k$ are the noise covariance errors, $\widehat{\omega}$ is the state error, I is the identity matrix and the symbol $(.^-)$ denotes the value before the estimation.

The AKF is a recursive-based estimation algorithm that requires parameter initialization as listed in Table 3.2. The initial value of the estimated disturbance vector, initially estimated states, initial estimation errors, and initial inputs were set to zeros.

Table 3.2: AKF parameters initialization

Parameter	Value	Parameter	Value
$\xi_d(0)$	$\mathbf{0}_{4\times4}$	$Q(0)$	$0.001I_{12}$
$\hat{x}(0)$	$\mathbf{0}_{8\times1}$	$P(0)$	$0.001I_{12}$
$u(0)$	$\mathbf{0}_{4\times1}$	$R(0)$	I_{12}
N_R	10^6	N_Q	10^6
$\bar{e}(0)$	$\mathbf{0}_{4\times1}$	$\bar{\omega}(0)$	$\mathbf{0}_{12\times1}$

3.9. Implementation of the modeling

The block diagram of the controller is shown in Figure 3.11. The disturbance is observed depending on the controller torque and the system's states. This torque is compensated directly in the disturbance observer.

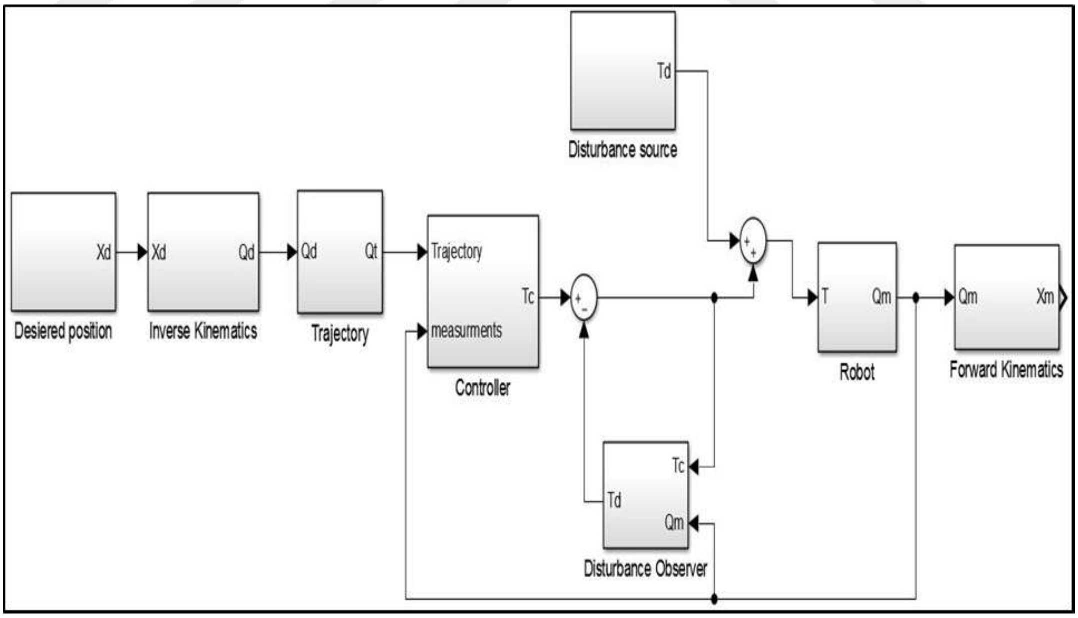


Figure 3.9. Block diagram of disturbance observation and rejection algorithm model

This study was implemented by the MatLab program after finding all the Kinematics and calculating all matrices. FK is used to find the position and orientation of the tool point, the point at the end of the robot arm. IK is essentially the completely opposite of FK. IK is about finding what the particular angles should be set for all the robot joints in order for the tool points at the end of the Robot arm to be in cartesian position and cartesian orientation. That means, what angles should be set in the robot in order to achieve that task. The goal of the Jacobian method is to know the velocity of the end effector. There are two types of robot velocities, the first one is the linear velocity for prismatic joints, and the second one is the angular velocity for revolute joints.

The trajectory planning is the robot movement's planning according to the desired trajectory, and the robot will move to the desired position from the present position of the end effector. It is desired that the robot be able to move smoothly without vibration, without exceeding the actuator's limits, and without crashing any objects in the workspace.

The PD inverse dynamics are useful in the control of robot to find the forces and torques on the robot. There are two methods to find the dynamics equations. The first one is the Lagrangian formulation, and the second one is the Newton-Euler formulation. This study uses the Lagrangian method due to its conceptual simplicity. The value of controller gains $K_d \; and \; K_p$ were calculated by using manual tuning.

When an external force affects the robot, the disturbance observer will overcome this force by compensating it with an opposite force to return it to zero. The disturbance observer calculates this disturbance and compensates for the external force to improve the stability and performance of the control system.

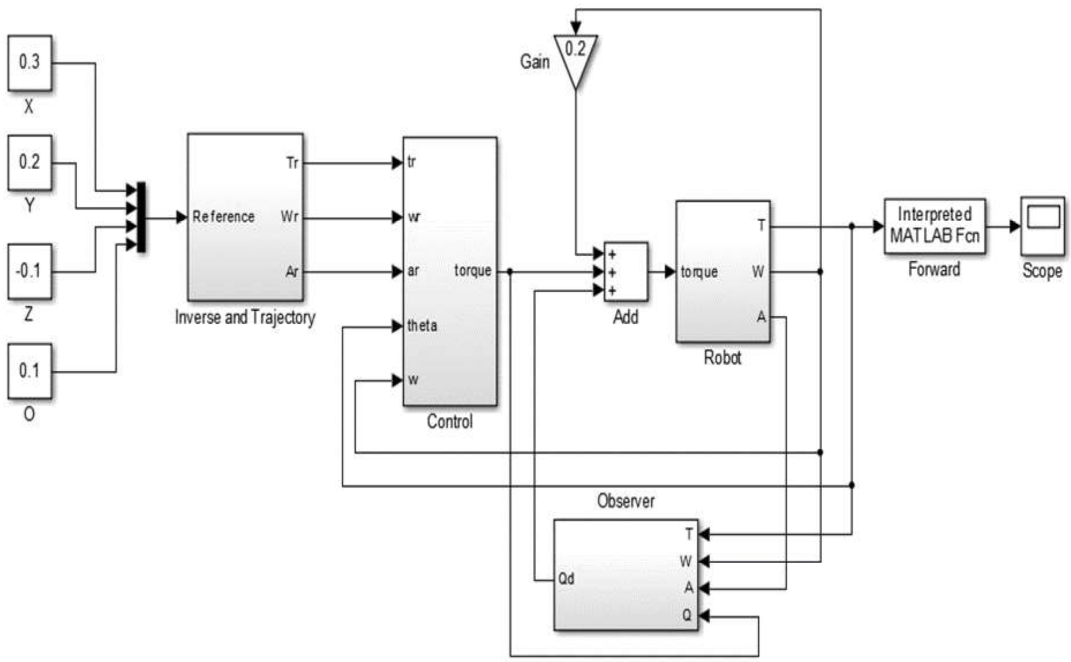


Figure 3.10. Block diagram of the system

Figure 3.10 illustrates a block diagram of the system; in the beginning, the robot is in the reference position and orientation (0.3 is the value of reference position in the x-axis, 0.2 is the value of reference position in the y-axis, -0.1 is the value of reference position in the z-axis, and 0.1 is the angle of orientation of the end effector). Then to convert from cartesian space (the robot moves from point to point by value x, y, z and rotation with angle) to joint space (the robot position is defined with the angles) and find unknown angles of the robot, inverse kinematics are used. The trajectory takes the values of angles from inverse kinematics to find the path for the planning of the robot (the trajectory function of time). The trajectory function gives the displacement, velocity, and acceleration. The control unit of robot takes displacement, velocity, and acceleration values of reference from the trajectory. The angles and velocity values are taken from the robot unit. All these parameters are used to calculate the torque by PD inverse dynamic in the robot. The disturbance observer takes the value of torque from the controller, and feedback from the robot (angles, velocity, and acceleration), to estimate and find the value of disturbance torque Q_d, and compensate it in the robot. The effect of the external force on the robot will changes angles, velocity, and acceleration.

The robot unit takes the value of torques (after compensating the value of disturbance torque) and calculates the acceleration velocity and angles by using a dynamic equation. To find the positions and the orientation of angles (the position in the x-axis, y-axis, z-axis, and angle oreantaion of end effector), forward kinematics are used.

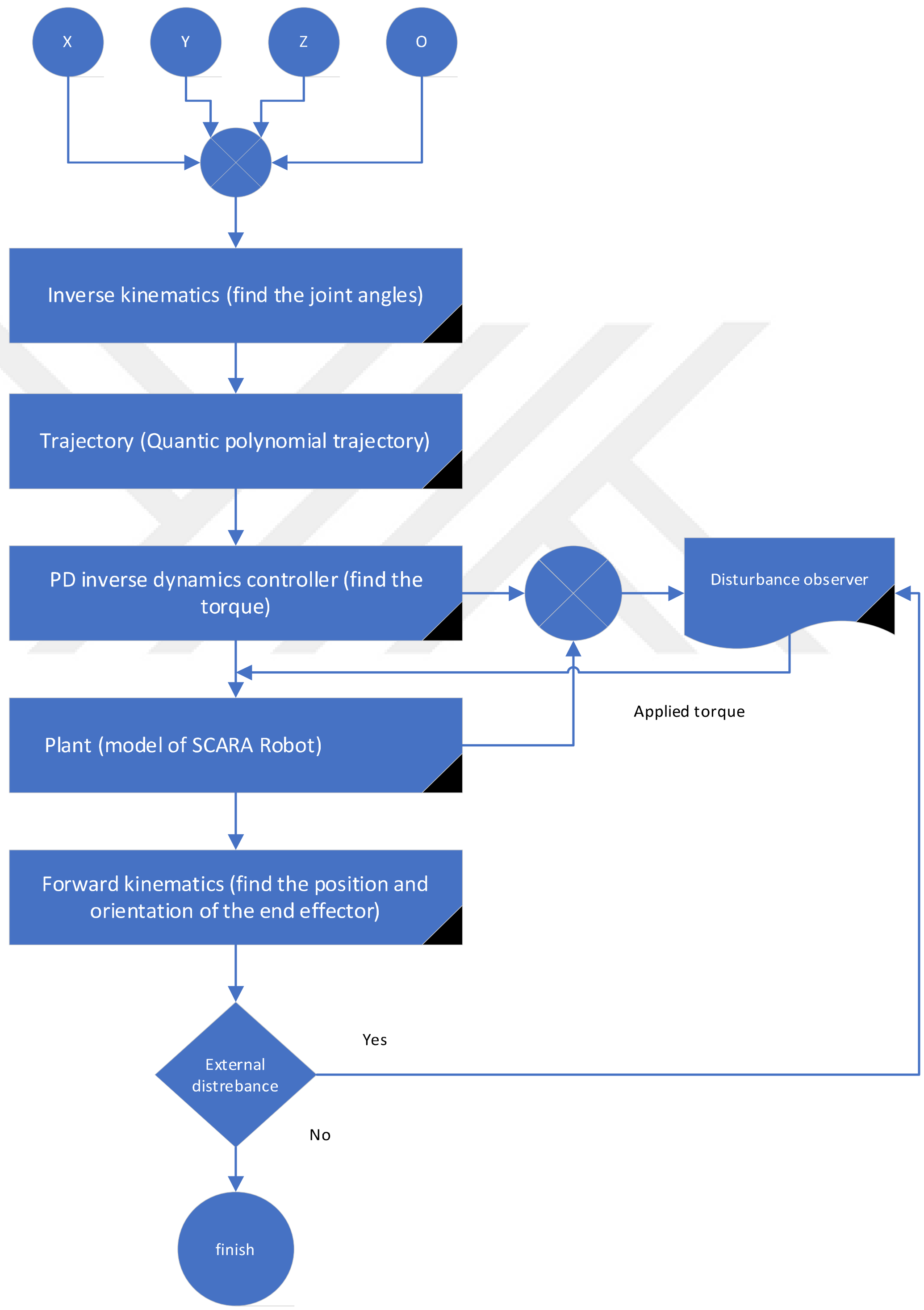


Figure 3.11. The system Flowchart

Figure 3.11 shows the system flowchart. The desired positions are represented by X, Y, Z, and O (angle of orientation). Then inverse kinematics is used to find the angles of robot. The trajectory is used to find the path of robot. After that, the PD inverse dynamics controller is used to find the torques. The plant of the SCARA robot determines the state of the robot. The forward kinematics are used to find the position orientation of the end effector. If the robot is affected by the external force, the disturbance observer takes feedback from the controller and the plant of the robot. If there is no external disturbance, the robot will move without problems.

4. RESULTS AND DISCUSSION

This thesis has applied external disturbance to the actuators. To study the effect of the external disturbance on the system's response, the proposed method was simulated and implemented using MATLAB software. The results contain two cases of study. In each case, the SCARA Robot should move from the initial position (0.4, 0, 0.2,0) to the desired position $(0.3\ ,0.2\ ,-0.1\ ,0.1\ )$ in one second.

In the first case, the value of disturbance is 0.5w, and in the second case, the value of disturbance is 0.1sin(t).

4.1 The results obtained from the first case, (the value of disturbance = 0.5 w) .

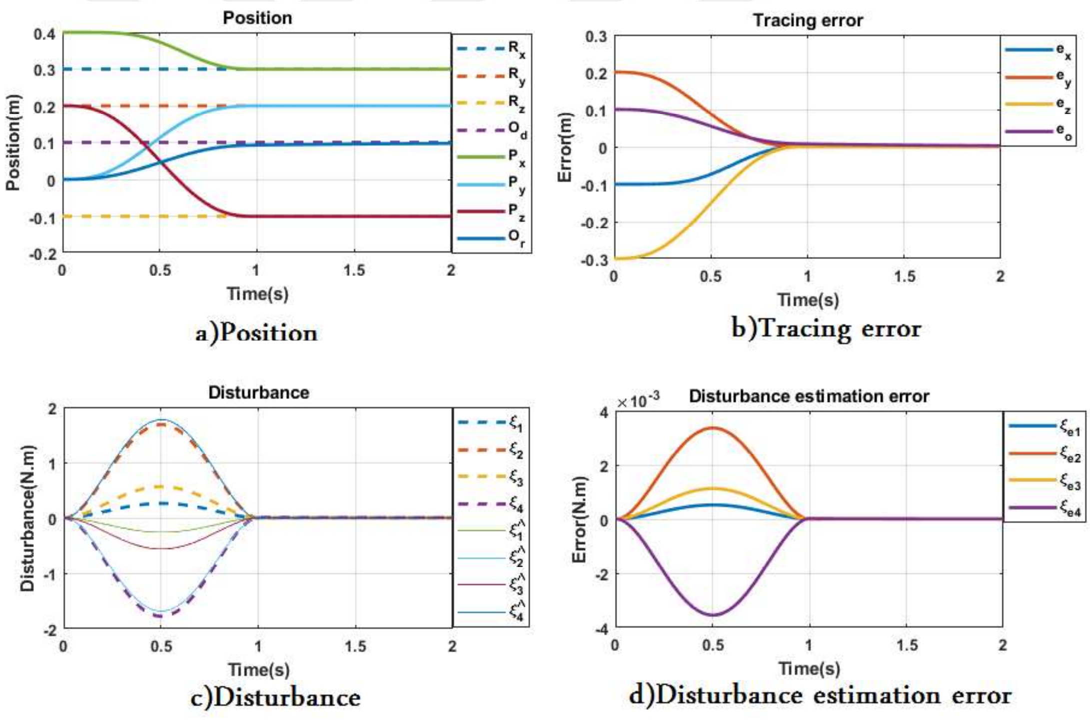


Figure 4.1(a,b,c,d). Robot response with DOB

Figure 4.1 illustrates the response of robot with DOB. The robot moves from the reference position (0.4, 0, 0.2 ,0) to the desired position (0.3,0.2,-0.1,0.1) during one second. The position on the X-axis changes from 0.4 to 0.3 during one second, and on the Y-axis changes from zero to 0.2 during one second.

The system on the Z-axis changes from 0.2 to -0.1 during one second, and the angle of orientation changes from zero to 0.1 during one second. That is, the system needs one second to reach to the steady-state, without overshoot, in smooth transition, and all the joints move in synchronization. Disturbance observers are used to estimate and suppress unknown disturbances affecting control systems and enhance the stability and performance. In this way, the efficiency will be increased, and time is saved in industrial applications by using the disturbance observer. Figure 4.1 also shows the tracking position error between the reference position and the desired position. In the beginning, the value of error will be very high, the error in X-axis is -0.1, and then it becomes zero during one second. The error in Y-axis drops from 0.2 to zero during one second. The error in Z-axis changed from -0.3 to zero during one second, and the error on the angle of orientation drops from 0.1 to zero during one second. And the figure illustrates the observed torque when external disturbance is applied to the actuators. In the beginning, the value of disturbance is zero. That is, there is no external force affect on the robot. After applying the disturbance on actuators, the value of disturbance changes. The value of disturbance on the X axis increases from zero to 0.17 during 0.5 seconds, and then decreased from 0.17 to 0 during one second. The value of disturbance on the Y axis increased from zero to 0.81 in 0.5 seconds, and then decreased from 0.81 to 0 during one second. The value of disturbance on the Z axis increased from zero to 0.24 during 0.5 seconds, and then decreased from 0.24 to 0 during one second. The value of disturbance on the orientation is decreased from zero to -0.81 during 0.5 seconds, and then increased from -0.81 to 0 during one second. If the robot is affected by these torques, it will lose the target, and thus the robot will operate in the wrong direction. And the last part in Figure 4.1 shows the disturbance estmation error after the robot is affected by external disturbance.

4.2 The results obtained from the second case, (the value of disturbance = 0.1sin(t)).

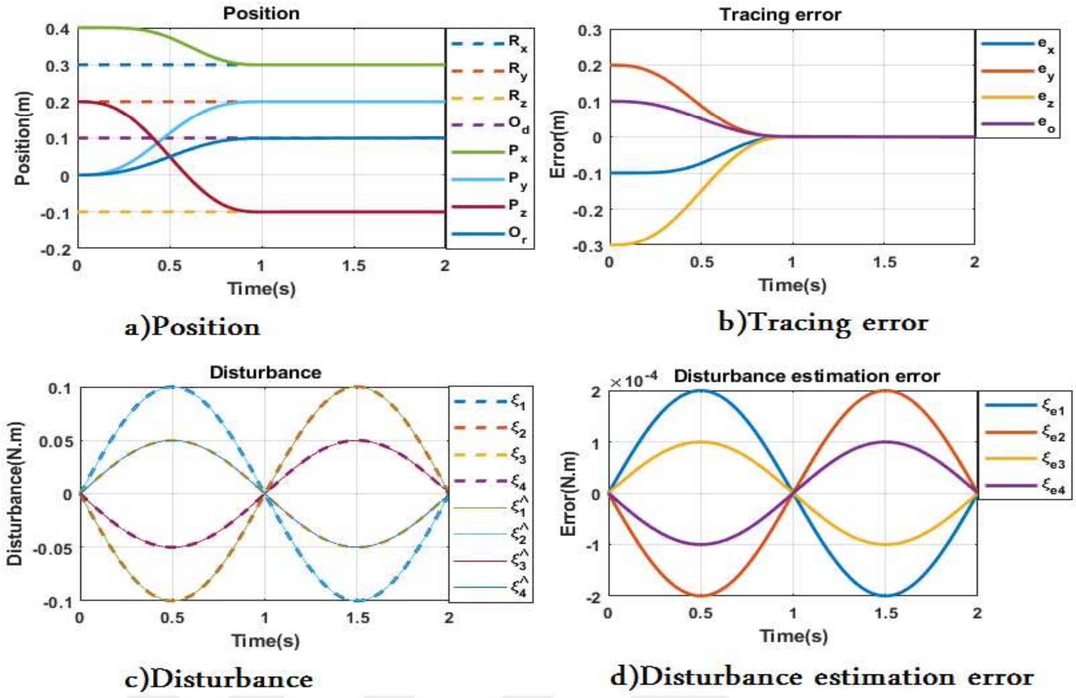


Figure 4.2(a,b,c,d). Robot response with DOB for case two

Figure 4.2 shows the response of robot with DOB for the case two. We can see the same results as in case one. The robot moves from the initial position (0.4, 0, 0.2,0) to the desired position (0.3,0.2,-0.1,0.1) in one second. That is, the system needs one second to reach the steady-state without overshoot, in smooth transition, and all the joints moving in synchronization. Even when the value of external force is changed, the disturebance observer will overcome this forces~~,~~ and the robot will move from the reference position (0.4, 0, 0.2,0) to the desired position (0.3,0.2,-0.1,0.1) without errors. The second part in Figure 4.2 shows the tracking position error between the reference position and the desired position. In the second case, the tracking error is the same as the first case because the position of the robot is not related to the torque distuerbance. The error on the X axis changes from -0.1 to zero during one second, and the error on the Y axis changes from 0.2 to zero during one second. The error on the Z axis changes from -0.3 to zero during one second, and the error in the angle of orientation changes from 0.1 to zero during one second. The third part in Figure 4.1 illustrates the torque disturbance from the external forces applied to the actuators. In the beginning, the value of disturbance is zero, after applying the external forces, this value starts to change. The value of disturbance on the X axis increases from zero to 1 during 0.5 seconds, and then decreased from 1 to -1 during 1.5 second. The value of disturbance on the Y axis decreased from zero to -1 during 0.5 seconds, and then incresed from -1 to 1 during 1.5 second. The value of disturbance on the Z axis increased from zero to 0.05 during 0.5 seconds, and then decreased from 0.05 to -0.05 during 1.5 second. The value of disturbance in the orientation changed from zero to -0.05 during 0.5 seconds, and then changed from -0.05 to 0.05 during 1.5 second.

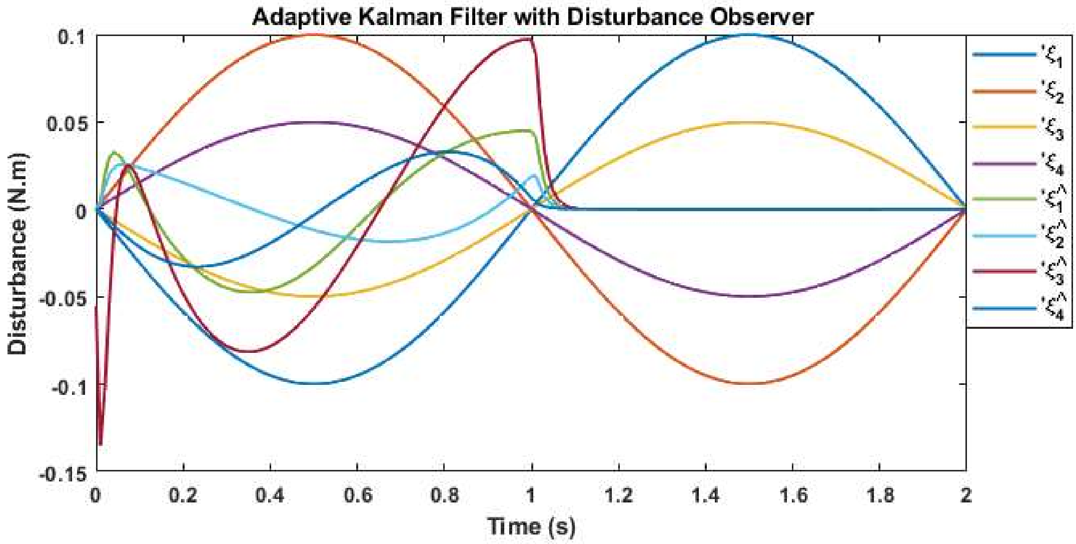


Figure 4.3. Robot response for AKF with DOB

Figure 4.3 illustrates the robot response for adaptive Kalman filter with non-linear disturbance observer. In the beginning, when we use the disturbance observer with the nonlinear system, the value of disturbance is zero in the first, second, third, and fourth joints, but when we use the adaptive Kalman filter with the nonlinear system, the disturbance is zero in the first, second, and fourth joints, but on the third joint starts from -0.06. That means, when we use the adaptive Kalman filter with the nonlinear system, we will have a high oscilation in the beginning, but the disturbance observer is more efficient with the nonlinear system. The adaptive Kalman filter with the linear system is more efficient than other observers but it has a problem with the nonlinear system and to solve this problem we use the nonlinear disturbance observer.

5. CONCLUSIONS AND RECOMMENDATIONS

In this study, the SCARA robot arm was modeled. The program life cycle applied has been verified and completed. The SCARA robot requires a comprehensive mathematical model, including complete SCARA robot arm kinematic studies. Forward (FK) and inverse (IK) equations were obtained by Denavit-Hartenberg. Jacobian and Trajectory Planning solutions were generated and implemented with the developed software. MATLAB software has been used to run simulation experiments. The SCARA robot structure was constructed to enable the researchers to examine both forward and inverse Kinematics robot parameters. In fact, the process of planning, creating, and controlling the actual robots require a large study.

When an external force was applied to the non linear system, the robot lost its target, and it operated in the wrong direction. The disturbance observer was activated to estimate external force and overcome this force, and the robot returned to the reference position.

In this thesis, the design of Nonlinear Disturbance Observers and Adaptive Kalman Filter for robotic manipulators of SCARA has been studied, and comparisons of the results have been realized. The results are graphically illustrated, and the movement of both joints and the end effector can be observed.

Since MATLAB requires high prossesing power to run, it takes a considerable amount of time. Therefore the software should be used with a high specifications personal computer.

The future study will concentrate on various issues, such as the creation of various types of controllers. It can be applied to the developed platform, then selecting the best control strategy for this type of manipulator. This is to improve the results and minimize the error between the real arm and the simulated one.

Many possible technologies may be applied to other types of robots, including path-planning, dynamics modeling, force control, and Disturbance observer.

REFERENCES

Advanced Textbooks in Control and Signal Processing. (2018). In Journal of Chemical Information and Modeling (Vol. 53).

Bona, B., & Indri, M. (2005). Friction compensation in robotics: An overview. Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC ’05, 2005, 4360–4367.

Caccavale, F., Cilibrizzi, P., Pierri, F., & Villani, L. (2009). Actuators fault diagnosis for robot manipulators with uncertain model. Control Engineering Practice, 17(1), 146–157.

Chan, S. P. (1995). A Disturbance Observer for Robot Manipulators with Application to Electronic Components Assembly. IEEE Transactions on Industrial Electronics, 42(5), 487–493.

Chen, W. H., Yang, J., Guo, L., & Li, S. (2016). Disturbance-Observer-Based Control and Related Methods - An Overview. IEEE Transactions on Industrial Electronics, 63(2), 1083–1095.

Critchlow, A. K. (1985). Introduction to robotics. 1–5.

Datar, R. (2015). Tuning of PID Controller for Ball and Beam System. International Journal of Engineering Research & Technology, 3(24), 1–3.

Eom, K. S., Suh, I. H., Chung, W. K., & Oh, S. R. (1998). Disturbance observer based force control of robot manipulator without force sensor. Proceedings - IEEE International Conference on Robotics and Automation, 4(May), 3012–3017.

Eom, Kwang Sik, Suh, I. H., & Chung, W. K. (2001). Disturbance observer based path tracking control of robot manipulator considering torque saturation. Mechatronics, 11(3), 325–343.

Féjoz, J., & Kaczmarek, L. (2004). Sur le théorème de Bertrand (d’après Michael Herman). Ergodic Theory and Dynamical Systems, 24(5), 1583–1589.

Ha, W., & Back, J. (2018). A Disturbance Observer-based Robust Tracking Controller for Uncertain Robot Manipulators. International Journal of Control, Automation and Systems, 16(2), 417–425.

Hirschorn, R. M. (1979). Invertibility of Multivariable Nonlinear Control Systems. IEEE Transactions on Automatic Control, 24(6), 855–865.

Homayounzade, M., & Khademhosseini, A. (2019). Disturbance Observer-based Trajectory Following Control of Robot Manipulators. International Journal of Control, Automation and Systems, 17(1), 203–211.

Jankowski, K. P. (2004). Inverse Dynamics Control in Robotics Applications.

Katsura, S., Matsumoto, Y., & Ohnishi, K. (2007). Modeling of force sensing and validation of disturbance observer for force control. IEEE Transactions on Industrial Electronics, 54(1), 530–538.

Komada, S. (2000). Control of redundant manipulators considering order of disturbance observer. IEEE Transactions on Industrial Electronics, 47(2), 413–420.

Kouhei, O., Shibata, M., & Murakami, T. (1996). Motion control for advanced mechatronics. IEEE/ASME Transactions on Mechatronics, 1(1), 56–67.

Manasra, A., Salaymeh, A., & Natsheh, Y. (2018). Design , control and implementation of SCARA robot for sorting missions with machine vision . By. Project Report, (May).

Methods of Solution to Inverse Problems. (2007). Fracture Mechanics, 293–322.

Mitsantisuk, C., Ohishi, K., Urushihara, S., & Katsura, S. (2011). Kalman filter-based disturbance observer and its applications to sensorless force control. Advanced Robotics, 25(3), 335–353.

Mohammadi, A. (2011). University of Alberta Disturbance Observer Design for Robotic and Telerobotic Systems Master of Science.

Poveda-Bautista, R., Diego-Mas, J. A., & Leon-Medina, D. (2018). Measuring the Project Management Complexity: The Case of Information Technology Projects. Complexity, 2018.

Prajumkhaiy, N., & Mitsantisuk, C. (2016). Sensorless Force Estimation of SCARA Robot System with Friction Compensation. Procedia Computer Science, 86, 120–123.

Radke, A., & Gao, Z. (2006). A survey of state and disturbance observers for practitioners. Proceedings of the American Control Conference, 2006(2), 5183–5188.

Sawut, U., Umeda, N., Park, K. B., Hanamoto, T., & Tsuji, T. (2001). Frictionless control of robot arm with sliding mode observer. Proceedings of the IEEE International Vehicle Electronics Conference 2001, IVEC 2001, 61–66.

Schneider, H., & Frank, P. M. (1996). Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation. IEEE Transactions on Control Systems Technology, 4(3), 274–282.

Shimada, N., Ohishi, K., Kumagai, S., & Miyazaki, T. (2010). Smooth touch and force control to unknown environment without force sensor for industrial robot. International Workshop on Advanced Motion Control, AMC, 36–41.

Sukmaningrum, Puji Suciance, F., Performaancial, Insurance, I., Pendahuluan, I., Belakang, L., Mohd Hussin, M. Y., … Mariana, F. B. (2008). No Title. (1) 49 د 73–69.

Surapong, N., & Mitsantisuk, C. (2016). Position and Force Control of the SCARA Robot Based on Disturbance Observer. Procedia Computer Science, 86(March), 116–119.

SWAIN, S., BEHERA, DEVENDRA SINGH MANDAVI, A. A., & Department. (2012). COMPARISION OF PERFORMANCE ANALYSIS OF DIFFERENT CONTROL STRUCTURES COMPARISION OF PERFORMANCE ANALYSIS Under the guidance of Prof . T . K . Dan. (10607022).

Technische Universtität München, L.-M.-U. M. (2018). . In e-conversion - Proposal for a Cluster of Excellence.

Urrea, C., Cortés, J., & Pascal, J. (2016). Design, construction and control of a SCARA manipulator with 6 degrees of freedom. Journal of Applied Research and Technology, 14(6), 396–404.

Villela, lucia maria aversa. (2013). 済無No Title No Title. Journal of Chemical Information and Modeling, 53(9), 1689–1699.

Wasnik, D. U., & Ghutke, P. (2018). Performance Evaluation between PLC and PID Controllers Used For Speed Control Of Induction Motor. 6(2), 661–665.

Yu, Z., & Su, J. (2013). Robust disturbance observer based control for multi-variable systems. In IFAC Proceedings Volumes (IFAC-PapersOnline) (Vol. 13).

Zhongyi, C., Fuchun, S., & Jing, C. (2008). Disturbance Observer-Based Robust Control of Free-Floating Space Manipulators. IEEE Systems Journal, 2(1), 114–119.