Addis Ababa University

Addis Ababa University Institute of Technology

School of Electrical and Computer Engineering

SLIDING MODE CONTROL OF A TWO DERGEE OF FREEDOM ROBOT ARM USING PERMANENT MAGNET SYNCHRONOUS MOTOR

A thesis Submitted to the Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University

In partial Fulfillment of the Requirement for the Degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING (ELECTRICAL CONTROL ENGINEERING)**

By Getachew Dereje

Advisor: Dr. Mengesha Mamo

Addis Ababa

March, 2018

Addis Ababa University

Addis Ababa University Institute of Technology

School of Electrical and Computer Engineering

SLIDING MODE CONTROL OF A TWO DERGEE OF FREEDOM ROBOT ARM USING PERMANENT MAGNET SYNCHRONOUS MOTOR

By Getachew Dereje Ourge

APPROVED BY BOARD OF EXAMINERS

_____	_____
Chairman, Department of Graduate Committee	Signature
Dr. Mengesha Mamo	_____
Advisor	Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

ACKNOWLEDGMENTS

I am very much thankful to my advisor, Dr. Mengesha Mamo for his support and valuable guidance throughout the study.

I thanks to all of my colleagues who support me during my graduate study. And also I would like to thanks my family for their love and partnership during my study

DECLARATION

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or other universities, all sources of materials used for this thesis work have been fully acknowledged.

Name: Getachew Dereje Signature: _____

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa

Date of Submission: May, 2018

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Mengesha Mamo Signature: _____

Advisor's Name

ABSTRACT

This thesis study aims at the problem of modeling and control of a two DOF robot arm using Permanent magnet synchronous actuators. Permanent magnet synchronous actuator have been used in various industrial and domestic applications because of its advantages like simple structure, large torque, long use. Trajectory tracking is a very difficult topic in robot arm control. This is due the nonlinearities and input couplings present in the dynamics of the arm caused by dynamics changes rapidly as the manipulator moves within its working range. Moreover, for a robot with gear transmissions, the gears have nonlinearities such as hysteresis, backlash, friction, and nonlinear elasticity. In industrial applications such complex systems are controlled using the traditional PID controllers. However, a major drawback of the linear PID control is failed to track the desired trajectory of the robot arm when the nonlinear system dynamics is dominant. In this thesis sliding mode controller is modeled to overcome the shortcomings of PID controller. The SMC is designed to control the joint trajectory of the robot. SMC adopts a switching function, and these results in high-frequency oscillations (the so-called chattering) in the control signal. To reduce chattering, saturation function is used.

The simulation is carried out using Level-2 MAT-LAB s-function and Simulink. PID controller is designed for the comparison purpose with SMC without disturbance and parameter variation and the result shows Steady State error of SMC is 0.029% for joints one and 0.0098% for joint two and increased to 15% and 24% respectively overshoot remain the same, rise time is decreased by 0.16sec, and settling time is decreased by 0.4sec for both joints when PID controller is used. The SMC and PID with disturbance and parameter variation also tested and results show that the performances with disturbance and parameter variation are almost the same for SMC. The control input voltage is increased by 0.03v and 0.02v for joint one and two respectively for SMC. However the amplitude of the control effort of PID controller with disturbance and parameter variation is much larger when compared to the control effort without disturbance and it is increased by 18v and 6.5v. Generally from the result it is possible to conclude that SMC has better performance, more stable and robust than PID controller.

Keywords: -SMC, PID, Level-2 MAT-LAB s-function, Simulink, DOF

TABLE OF CONTANTES

LIST OF ABBREVIATION

BLDC	brushless Direct current
BSMC	baseline sliding mode controller
DC	direct current
DH	Denavit-Hartenberg
DOF	degree of freedom
PI	proportional integral
PID	proportional integral derivative
PMS	permanent magnet synchronous
PUMA	programmable universal machine for assembly
RMS	root mean square
SMD	sliding mode control
SMC	sliding mode control
SMC's	sliding mode controllers
VSC	variable structure control
VSS	variable structure system

LIST OF SYMBOL

B	Motor friction factor
c_i	Number of constraints imposed by joints
e	error (the difference between actual and desired output)
$\dot{e}$	Derivative of the error (derivative of the difference between actual and desired out
F	fixe reference frame
f_i	Degrees of freedom permitted by joints
f	Final control law
g	Gravity acceleration
g_r	Gear ratio
i_d, i_q	d and q axis currents
J	jacobian matrix
J	Motor Inertia
j	number of mechanism joints
j_i	Number of joints with i degrees of freedom
K_i	Kinetic energy
k_d	Derivative constant
k_i	Integral constant
k_p	Proportional constant
L	Lagrangian
L_q, L_d	q and d axis inductances
l_{ci}	Links center of mass
n	number of mechanism links
P	number of pole pairs
P_i	Potential energy
$\dot{q}$	Angular velocity
R	Motor resistance of the stator windings
S	sliding surface

LIST OF FIGURE

LIST OF TABLE

CHAPTER 1
INTRODUCTION

1.1. Back Ground

Most of the industry in our world uses robot to increase their product and profit. Because robot has so many an attractive features in an industrial environment. Among those advantages are decreased labor costs, increased productivity and precision, increased flexibility compared with specialized machines [1]. The first programmed robot was being able to execute a sequence of movement, such as moving to location A, closing gripper, moving to location B etc... [2].

Robotics is relatively young field of modern technology. The science of robotics has grown tremendously over the past twenty years, fueled by rapid advances in computer and sensor technology as well as theoretical advances in control and computer vision. A complete treatment of the discipline of robotics would require several volumes. Nevertheless, at the present time, the vast majority of robot applications deal with industrial robot arms operating in structured factory environments so that a first introduction to the subject of robotics must include a rigorous treatment of robot dynamic control [2].

There are a lot of control methodologies that can be used for the control of robot arms. The most common methodologies are: feedback linearization control Methodology, passivity-based control methodology, sliding mode control methodology, robust Lyapunov-based control methodology, adaptive control non-linear methodology and artificial intelligence-based methodology. [2]
Non-linear control methodologies are more general because they can be used in linear and no-linear systems. These controllers can solve different problems such as invariance to system uncertainties and resistance to the external disturbance.

Among those control methodology; sliding mode controller is the powerful nonlinear controller which has been focused by many researchers in recent years. This theory was first proposed in the early 1950 by Emelyanov and several co-workers and has been extensively developed since then with the invention of high speed control devices [3], [4]. The reason why to choose this controller is it have good control performance than conventional PID controller and solve two most important challenging topics in control which names, stability and robustness The reason why to choose this controller is solve two most important challenging topics in control which

names, stability and robustness [2], [5] [6]. But, pure sliding mode controller has the following disadvantages. First; chattering problem; this can cause the high frequency oscillation of the controller out put. To reduce the chattering different paper have been reported by many researchers and classified in to two most important methods, i.e. boundary layer saturation function method and estimated uncertainties method [7], [8]. Second, sensitivity; this controller is very sensitive to the noise when the input signals very close to zero. The other one is nonlinear equivalent dynamic formulation; this problem is very important to have a good performance and it is difficult to calculation because it is depending on nonlinear dynamic equation [9], [10], [11] In this thesis sliding mode control using reaching method is chosen and the dynamic of the two degree of freedom robot is derived including the dynamic of permanent magnet synchronous actuator. In most of the relevant work dynamic of the actuator dynamics is neglecting and assumed that the supply torque or force has been given to the robot joint. The chattering problem is solved by using saturation function in instead of using pure signum function.

Statement of the Problem

Robot manipulators are multi-body system having nonlinear, coupled dynamics and parameter fluctuations. The nonlinear and coupled dynamics represent example: - gravitational forces which dependent on position, carioles and centrifugal force. Reaction forces in joints due to the acceleration of other links, friction force etc... In industrial application such complex systems are controlled using the traditional proportion-integral-derivation (PID) controllers. Which are easy to implement require the tuning of only three parameters, and relatively acceptable tracking performance of the robot. But the major drawback of PID control is failed to accomplish the desired tracking performance of the robot arm when the nonlinear system dynamics is dominating the linear dynamic. For example in robotics systems use for assembling operations with heavy work pieces carried by manipulator's gripper, robot with high speed operating etc.

Different researchers have presented to avoid the shortcoming of PID control using nonlinear controller and different approaches have been proposed for designing the nonlinear control, (I) the SMC [12].(II) The fuzzy logic control [13].In this thesis to overcome the shortcoming of PID controller sliding mode controller is modeled by considering the actuator dynamics. Nonlinear equivalent dynamic which include the actuators dynamic formulation is to improve SMC performance response. Here without considering the actuator dynamics can cause imperfection in

the final control law. However SMC adopts switching function in its design, and these results in high-frequency oscillations called as chattering in the control signal which is undesirable since it can damage the actuator and system. To reduce the chattering problem of sliding mode controller saturation function has been used instead of pure sliding mode. The two degree of freedom robot is actuated by two permanent magnet synchronous motor. Because of its advantages like cheaper, cleaner and quitter than hydraulic and pneumatic actuator

1.2 Objective of the Thesis

1.2.1 General Objective

To develop a controller for two degree of freedom robot arm using permanent magnet synchronous actuator

1.2.2 Specific Objective

- To develop dynamics and kinematics of a two degree of freedom robot arm
- To simulate and demonstrate sliding mode controller of a two degree of freedom robot arm using Permanent magnet synchronous actuator

1.3 Methodology

The proposed thesis first concentrated at study of the literatures about controller design and theoretical and structural backgrounds about robot manipulator and controller. Then after we get a good understanding about robot manipulator and controller from the reviewed literature, we have compared the performance various controller and then we have selected our technique which can remove the shortcoming of the proportional integral derivative controller.

Since appropriate dynamic model equation has very important in designing the robot controller. To develop the model we have divided this thesis in to two main parts. First we developed the robot manipulator dynamic equation. This stars from the equation of position and orientation description, forward and inverse kinematics, dynamic analysis and forces, kinematic and potential energy by using Lagrange equation. The second is the model of actuator dynamics of robot with some assumption. These actuators are PMSM. And finally the overall nonlinear system model or manipulator dynamic and actuator dynamics is developed. As most of our literature work indicated us the control input is torque or forces but in our case instead of these we use the actuators voltage from the developed mathematical model. This is to improve one of the drawbacks of our technique.

After the dynamic models have been developed, based on the SMC theory the two degree of freedom robot manipulator controllers are designed. To handle the problem of chattering reaching law method are designed.

The performance of the proposed method is tested by simulation using MATLAB simulation (level-2matlab s-function), and it has better performance than PID controller.

1.4 Outline of the Thesis

This thesis is organized into six chapters. The first chapter narrates the overview of robotics and different control mechanisms, statement of the problem and objectives of the thesis.

In chapter two, different literatures related to two degree of freedom robot and their control systems are reviewed.

Mathematical modeling of robot dynamics, actuator robot dynamics and overall dynamics of the system are presented in chapter three. Sliding mode control design, sliding surface design and switching surface design algorithm are presented and discussed in chapter four.

Simulation results and discussion are presented in chapter five. Finally, chapter six presents conclusions and future scope**.**

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

A robot is a reprogrammable multifunction machine responsible to move parts, materials, tools or specialized devices through variable programmed motion for the performance of different tasks. To perform the variety of tasks it needs the necessary combination of actuator, manipulator and controller of the robot. The robot end-effecter manipulator needs an appropriate selection of actuator and controller (control mechanism) to track the desired input effectively [2] [14] [15] [16] [17] [18].

2.2 Related works on robot control system

Different modeling and designing of controller strategies were presented in various researches for trajectory tracking of robot manipulator; from these researches which focus on trajectory tracking of robot manipulator and other related controller of robot has been reviewed in the following section.

D.R.V.a.P.N.T.Jyoti Ohri [14] presents the comparison of PID and SMC for robotic Manipulator. The researcher assume that the actuators do not have dynamics of their own and arbitrary torques can be commanded at the joint of the robot. The result shows that; the performance of the SMC is better than PID under sine and cosine trajectory. In case of uncertainty PID controller is tracking error is increased but in case of SMC performance remains same so SMC is more robust than PID controller. The researcher solves the chattering phenomenon by saturation function in SMC.

S. M. R. B. a. S. R. . Farzin Piltan [15] this researcher is focuses on design of a chattering free mathematical Baseline sliding mode control for highly nonlinear dynamic robot manipulator, in presence of uncertainties and external disturbance. Baseline sliding mode is provided to reduce the drawback of the pure sliding mode controller namely chattering.

Baseline controller was designed to control the sliding surface slope of sliding mode controller. Since there was an output from the sliding surface slope model, this means that there would be two inputs in to the baseline controller. Similarly, the output of the controller resulted from the controller input of the sliding surface slope. Two controllers was cascaded together to control the

sliding surface slope. The first were the PID controller that corrected the error between the desired joint variable and the measured joint variable; while the second was only PI controller that corrects the sliding surface error and integral of error. This controller tracking performance was tested without disturbance in SMC and BSMC for step input as desired joint variable. Based on the result, SMC has chattering in the presence of switching mode function but BSMC has no chattering.

For comparing purpose the researcher used step trajectory response with 10% and 20% disturbance relative to input signal amplitude. As results BSMC performance were better than SMC and BSMC can eliminate the chattering in the presence of 10% disturbance. The two controllers have oscillation in the presence of 20% disturbance uncertainty but BSMC is more stable and more robust than SMC. Torque performance in the presence of 10% disturbance for first three links industrial robot gives significant chattering elimination for BSMC when compared to SMC. This elimination of chattering phenomenon is very significant in presence of 20% disturbance

S. a. B. Jolly Shah [16] this researcher is focuses on design of computed torque control methodology and it was done for first 10sec and second 120sec and show that the validity of the method. The Position errors are coming to be less than 1% for almost initial 10 sec but it's increasing as time going on. While velocity errors are more for initial 10secs but it's decreasing and getting almost zero as time increasing. This method shows that error between derived and predicted position of link 1 and link 2 is almost around zero for first 10sec but as time increase errors are increasing but still it is under control. So if it's required to control a 2 DOF robot for few seconds computed torque controller is suitable but for long time duration it is producing certain errors in performance [16].

M. M. Fateh [17] this method presents a novel approach to control electrically driven robot manipulators based on voltage control. The voltage-based control is preferred comparing to torque-based control. This approach is robust in the presence of manipulator uncertainties because it is free of the manipulator model. The control law is very simple, fast response, efficient, robust, and can be used for high-speed tracking purposes. The feedback linearization is applied to the electrical equations of the dc motors to cancel the electrical current terms which transfer all manipulator dynamics to the electrical circuit of motor.

In this method, each joint of the manipulator is driven by electrical actuator in the control system. The inserted torque on the joint to drive the manipulator is the load torque of motor, which is considered in a dynamic equation

As an industrial robot transfers objects of different masses, the dynamics of manipulator will be changed. Therefore, the manipulator may be subject to external disturbances when robot operates in the workspace.

The control approach has been proposed based on electrical view on the motor circuit of a manipulator driven by electrical actuator. The motor current has been considered for compensating all dynamic torques provided by the manipulator. The control approach is based on feedback linearization to omit the current terms and providing a linear decoupled system in the form of first integrator. The control law is free of the manipulator model, so the control approach is robust subject to manipulator uncertainties and external disturbances. The design is simple using the independent joint strategy and the control algorithm has a fast response from few calculations. Therefore, the control can be used for high-speed operation, as well. A smooth trajectory is proposed to reduce the dynamical reactions to manipulator when starting and stopping under the allowing values of motor inputs. Tracking a specific trajectory provides satisfactory responses from the case of point-to-point control, as well as in the case of rejecting disturbances, tracking, set point and high-speed applications. The control law shows a very small tracking error in comparing to PID control system. The method is robust, simple, accurate, with less computing as compared with inverse dynamic control system [17].

CHAPTER THREE
MATHEMATICAL MODELLING

3.1 Introduction

In the study of robotics, we are constantly concerned with the location of objects in three-dimensional space. These objects are the links between the manipulator, the parts and tools with which it deals, and other objects of the manipulator's environment. These objects are described by just two attributes: position and orientation. In order to describe the position and orientation of a body in space, we will always attach a coordinates system, or frame, rigidly to the object. We then proceed to describe the position and orientation of this frame with respect to some reference coordinate system. (See Figure 1)Any frame can serve as a reference system within which to express the position and orientation of a robot body. Robot is made up of several links connected serially, which are connected by joints that allow relative motion of neighboring bodies. These joints are usually instrument with position controllers. In the case of rotary or revolute joints, these displacements are called joint angles. Some manipulators contain sliding (or prismatic) joints, in which the relative displacements between links are a translation, sometimes called the joint offset. The number of degrees of freedom that a manipulator possesses is the number of independent position variables that would have to be specified in order to locate all parts of the mechanism. This is a general term used for any mechanism. For example, a four-bar linkage has only one degree of freedom (even though there are three moving members). In typical industrial robots case a manipulator is usually an open kinematic chain. Each of joint position is usually define with a single variable which implies the number of degree of freedom. At the free end of the chain of links that make up the manipulator is the end-effectors'. Depending on the intended application of the robot, the end-effectors' could be a gripper, a welding torch, an electromagnet, or another device. Generally we describe the position of the manipulator by giving a description of the tool frame, which is attached to the end-effectors', relative to the base frame attached to the nonmoving base of the manipulator. A very basic problem for the study of mechanical manipulation is called forward kinematics. This is the static geometrical problem of computing the position and orientation of the end-effectors' of the manipulator [19]. Generally robots are actuated electrically, hydraulically or pneumatically. Hydraulic actuators are beyond comparison in their speed of response and torque producing capability. Therefore hydraulic robots are used primarily for lifting

heavy loads. The drawbacks of hydraulic robots are that they tend to leak hydraulic fluid, require much more peripheral equipment, such as pumps, which also requires more maintenance, and they are noisy. Pneumatic robots are economical and simple but cannot be controlled exactly. As a result, pneumatic robots are limited in their range of applications and popularity [1]. Most of this industrial automated robot uses the electrical motor.

In this chapter a two DOF robot arm and two PMSM are used for each of the two revolute joints of the 2-DOF robot manipulator is modeled. SMC designed with saturation function to eliminate chattering problem.

The robot's degree of freedom (DOF) depends on the number of links and joints, their types and the kinematics chain of the robot. This introduces the concept of calculating degrees of freedom with an algebraic formula. Several ways exist to do this. One of the easiest, and earliest, is Grubler's formula, which was derived in 1883. For a given mechanism,

$$F = \lambda(n-1) - \sum_{i=1}^{j} c_i \tag{3.1}$$

Since

$$\lambda = c_i + f_i$$

We drive Grubler's Criterion that enables us to calculate degrees of freedom

$$F = \lambda(n-j-1) + \sum_{i=1}^{j} C_i \tag{3.2}$$

Link is the rigid part of the robot body (example arm of the robot).

Joint is a part of the robot body which allows controlled or free relative motion of two links (connection element).

End effectors' is the link between the manipulator which is used to hold the tools (gripper, spray gun, welding gun)

Base is the link between the manipulator which is usually connected to the ground and is directly connected to the world coordinate system.

Kinematic pair is a pair of links which relative motion is bounded by the joint connecting to them

Kinematics Chains mechanisms can be configured as kinematics chains. The chain is closed as the ground link begins and ends the chain; otherwise it is open.

Serial robots: -The manipulator of a serial robot is, in general, an open kinematics chain. The joints must be controlled individually.

3.1.1 Pose of a rigid body

Rigid-body motion in the three-dimensional Cartesian space comprises translation and rotational. Translation is defined by using the three Cartesian coordinates and the rotation needs three angular coordinates. Hence, the rigid body motion can be defined completely by six coordinating in the study of the kinematics of robot manipulator, one can constantly deals with the position and orientation of several bodies in space. The bodies of interest include the links of the manipulator, tools, and work space. To identify the position and orientation of a body, i.e. it's 'pose' or 'configuration', a fixed reference coordinate system is established, which is called the 'fixed frame'

Next the Cartesian coordinate system attached to the moving body is employed to describe its Pose.

3.1.2 Position description

The position of any point **p**, in the rigid body in motion with respect to the fixed reference frame can be described by the three-dimensional Cartesian vector. Let us the coordinate of point **p** or the components of vector $\mathbf{p_x}$, $\mathbf{p_y}$ and$\mathbf{p_z}$, in the fixed frame **F**, it is denoted as

$$[P]_F = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{3.3}$$

Where the subscript **F** stands for the reference frame where the vector **p** is represented. The subscript x, y and z represents the projection of the position vector **p**, onto the coordinate axes of the fixed reference frame, namely, along x, y and z respectively. Vector **p** can alternatively be expressed as

$$P = P_x x + P_y y + P_z z \tag{3.4}$$

Where x, y and z denoted the unit vectors along the axes x, y and z of the frame **F**, respectively [20].

$$[\mathrm{X}]\mathrm{F}=\begin{bmatrix}1\\0\\0\end{bmatrix}, [\mathrm{y}]\ \mathrm{F}=\begin{bmatrix}0\\1\\0\end{bmatrix} and\ [Z]F = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

3.1.3 Orientation description

To describe the orientation of a body; attach a coordinate system to the body and then give a description of this coordinate system relative to the reference system.

In Figure 2, coordinate system (B) has been attached to the body in a known way. A description of {B} relative to (A) now suffices to give the orientation of the body. Thus, positions of points are described with vectors and orientations of bodies are described with an attached coordinate system. One way to describe the body-attached coordinate system, (B),is to write the unit vectors of its three principal axes-2 in terms of the coordinate system {A}.We denote the unit vectors giving the principal directions of coordinate system (B} as $\hat{X}_B$, $\hat{Y}_B$ and $\hat{Z}_B$'When written in terms of coordinate system {A},they are called ${}^A_B\hat{X}$, ${}^A_B\hat{Y}$ and ${}^A_B\hat{Z}$. It will be convenient if we stack these three unit vectors together as the columns of a 3 x 3 matrix, in the order ${}^A_B\hat{X}$, ${}^A_B\hat{Y}$ and ${}^A_B\hat{Z}$. We will call this matrix a rotation matrix because this particular rotation matrix describes {B}relative to {A}, the choice of frame in which to describe the unit vectors is arbitrary as long as it is the same for each pair being dotted. The dot product of two unit vectors yields the cosine of the angle between them, so it is clear why the components of rotation matrices are often referred to as direction cosines. The situation of a position and an orientation pair arises so often in robotics that we define an entity called a frame, which is a set of four vectors giving position and orientation information. For example, in Figure 3.1 one vector locates the fingertip position and three more describe its orientation. Equivalently, the description of a frame can be thought of as a position vector and a rotation matrix. We give a position vectors which locates its origin relative to some other embedding frame [19].

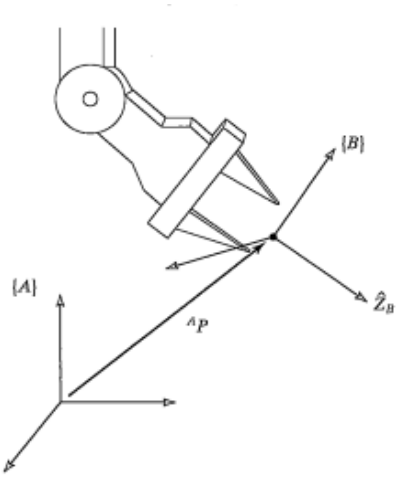

Figure 3.1: Locating an object in position and orientation [19]

3.2 Robot Kinematics

3.2.1 Forward and Inverse Kinematics

Kinematics studies the motion of bodies without consideration of the forces or moments that cause the motion. Robot kinematics refers the analytical study of the motion of a robot manipulator. Formulating the suitable kinematics models for a robot mechanism is very crucial for analyzing the behavior of industrial manipulators. There are mainly two different spaces used in kinematics modeling of manipulators namely, Cartesian space and Quaternion space. The transformation between two Cartesian coordinate systems can be decomposed into a rotation and a translation. There are many ways to represent rotation, including the following: Euler angles, Gibbs vector, Cayley-Klein parameters, Pauli spin matrices, axis and angle, orthonormal matrices, and Hamilton's quaternion. Of these representations, homogenous transformations based on 4x4 real matrices (orthonormal matrices) have been used most often in robotics. Denavit-Hartenberg (1955) showed that a general transformation between two joints requires four parameters. These parameters known as the Denavit-Hartenberg (DH) parameters have become the standard for describing robot kinematics. The robot kinematics can be divided into forward kinematics and inverse kinematics. Forward kinematics problem is straightforward and there is no complexity deriving the equations. Hence, there is always a forward kinematics solution of a manipulator. Inverse kinematics is a much more difficult problem than forward kinematics. The solution of the inverse kinematics problem is computationally expansive and generally takes a very long time in the real time control of manipulators. Hence, only for a very small class of kinematically simple manipulators (manipulators with Euler wrist) have complete analytical solutions. The relationship between forward and inverse kinematics is illustrated in Figure 3.2 [20].

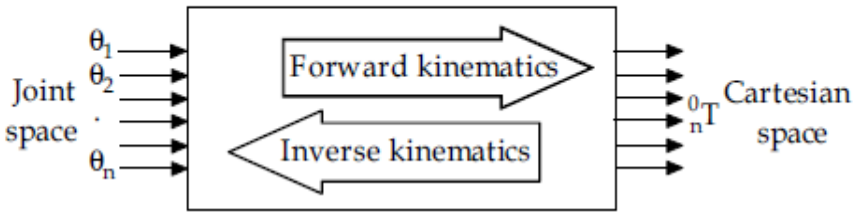

Figure 3.2: Schematic representation of forward and inverse kinematics [20]

Two main solution techniques for the inverse kinematics problem are analytical and numerical methods. In the first type, the joint variables are solved analytically according to given configuration data. In the second type of solution, the joint variables are obtained based on the

numerical techniques. In this chapter, the analytical solution of the manipulators is examined rather than numerical solution. There are two approaches in analytical method: geometric and algebraic solutions. Geometric approach is applied to the simple robot structures, such as 2-DOF planar manipulator or less DOF manipulator with parallel joint axes. For the manipulators with more links whose arms extend into 3 dimensions or more the geometry gets much more tedious. In this case, algebraic approach is more beneficial for the inverse kinematics solution.

There are some difficulties to solve the inverse kinematics problem when the kinematics equations are coupled, multiple solutions and singularities exist. Mathematical solutions for inverse kinematics problem may not always correspond to the physical solutions and method of its solution depends on the robot structure.

3.3 Homogenous Transformation Modeling

3.3.1 Forward Kinematics

A manipulator is composed of serial links which are affixed to each other revolute or prismatic joints from the base frame through the end-effectors'. Calculating the position and orientation of the end-effector in terms of the joint variables is called as forward kinematics. In order to have forward kinematics for a robot mechanism in a systematic manner suitable kinematics model should use. Denavit-Hartenberg method that uses four parameters is the most common method for describing the robot kinematics. These parameters a_{i-1},α_{i-1}, d_i and θ_i are the link length, link twist, link offset and joint angle, respectively. A coordinate frame is attached to each joint to determine DH parameters. Z_i Axis of the coordinate frame is pointing along the rotary or sliding direction of the joints [21].

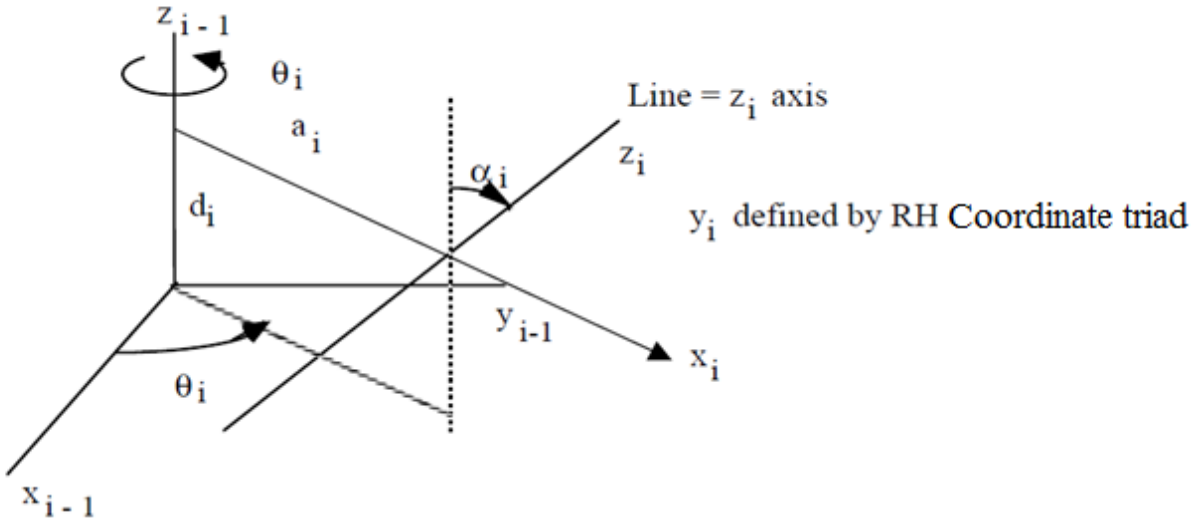

Figure 3.3: Coordinate frame assignment for a general manipulator [21]

As shown in figure 3.3, the distance from $Z_{i-1}$to Z_i measured along $X_{i-1}$is assigned as α_{i-1}, the angle between $Z_{i-1}$and Z_i measured along X_i is assigned as to α_{i-1}, the distance from $X_{i-1}$to X_i measured along Z_i is assigned as d_i and the angle between X_{i-1} to X_i measured about Z_i is assigned as θ_i. The general transformation matrix ${}_i^1T$ for a single link can be obtained as follows. For revolute joints θ_i are the joint variable with d_i,a_i and α_i constant. For prismatic joints the joint variable is d_i with θ_i , a_i, and α_i constant (a_i is typically zero) [21].

Given a revolute joint a point X_i located on the i^{th}link can be located in $i-1$ axis by the following transformation set which consist of four homogeneous transformations (2 rotations and 2 translations). The set that will accomplish this is

$${}_i^{i-1}T = H(d,\ Z_{i-1})\ H(\theta,\ Z_{i-1})\ H(a,\ X_{i-1})\ H(\alpha,\ X_i \text{ (i=1….n)} \quad (3.5)$$

Where

$$H(d, Z_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H(\theta, Z_{i-1}) = \begin{bmatrix} c_{\theta i} & -s_{\theta i} & 0 & 0 \\ s_{\theta i} & c_{\theta i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H(a, x_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H(\alpha, x_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha i} & -s_{\alpha i} & 0 \\ 0 & s_{\alpha i} & c_{\alpha i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Given the ${}_{i}^{i-1}T$ transformation matrices of one joint axes relative to the preceding axes, one can relate any point in the i^{th}link to the global reference frame by the following transformation set. Its coordinate's u_i in global axes are (n = DOF) and for n=2 and using the following DH parameters in table 3.1.

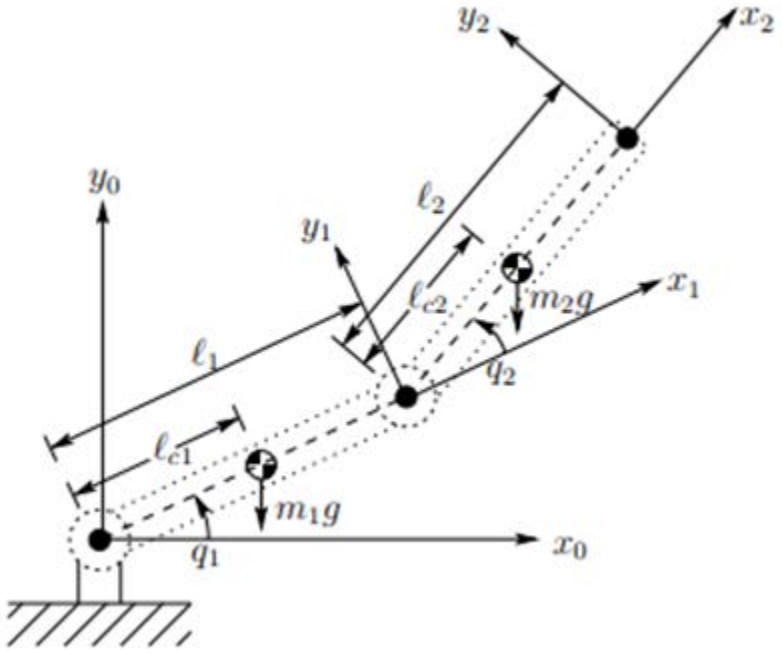

Figure 3.4: Frame assignments for the 2-DOF robot arm [22]

The DH parameters for the 2-DOF robotic arm shown in figure 3.4 can be determined and listed in table 3.1.

Table 3.1: DH parameters for the 2-DOF robotic arm [22]

Link*(i)*	q_i	d_i	α_i	a_i
1	q_1	0	0	l_1
2	q_2	0	0	l_2

The homogenous transformation matrices for the 2-DOF robotic arm shown in figure 3.4 are derived as follows:

$$ {}^{0}_{1}T = \begin{bmatrix} c_{q1} & -s_{q1} & 0 & lc_{q1} \\ s_{q1} & c_{q1} & 0 & l_1 s_{q1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6.a) $$

$$ {}^{1}_{2}T = \begin{bmatrix} c_{q2} & -s_{q2} & 0 & l_2 c_{q2} \\ s_{q2} & c_{q2} & 0 & l_2 s_{q2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6.b) $$

The T-matrices are thus given by

$$ {}^{2}_{0}T = {}^{0}_{1}T\,{}^{1}_{2}T = \begin{bmatrix} c_{q12} & -s_{q12} & 0 & a_1 c_{q1} + a_2 c_{q12} \\ s_{q12} & c_{q12} & 0 & a_1 s_{q1} + a_2 s_{q12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7) $$

The homogenous transformation matrix defined in (3.7) is the one that defines the forward kinematics of the 2-DOF robotic arm shown at figure 3.4. From this matrix, the position and orientation of end effectors is a non-linear function of joint variables P(x, y) = f (q). Having derived the forward kinematics or direct kinematics of the figure 3.4, it's now possible to obtain the end-effector position and orientation from the individual joint angles (q_1 *and* q_2)

$$ x_2 = a_1 cos q_1 + a_2 \cos(q_1 + q_2) \quad (3.8) $$

$$ y_2 = a_1 sin q_1 + a_2 sin(q_1 + q_2) \quad (3.9) $$

$$ Z_2 = 0 \quad (3.10) $$

And the end-effectors' orientation matrix is defined by the first three rows and three columns of the transformation matrix (3.7).

By differentiating the above two expressions (3.8 and 3.9)

$$ \dot{x}_2 = -a_1 \dot{q}_1 sin q_1 - a_2(\dot{q}_1 + \dot{q}_2)\sin(q_1 + q_2) \quad (3.11\text{-a}) $$

$$ \dot{y}_2 = a_1 \dot{q}_1 cos q_1 + a_2(\dot{q}_1 + \dot{q}_2)\cos(q_1 + q_2) \quad (3.11\text{-b}) $$

In a matrix form

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix}=\begin{bmatrix}-a_1 sinq_1 - a_2 sinq_{12} & -a_2 sinq_{12}\\ a_1 cosq_1 + a_2 cosq_{12} & a_2 cosq_{12}\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}$$

$$\dot{X} = J \bullet \dot{q} \tag{3.12}$$

Where, $\dot{X}$ is the velocity of end effector, J is Jacobean matrix and $\dot{q}$ represents joint ratesJacobean matrix represents the relationship between rates of change of pose with respect to joint rates. Rank deficiency of Jacobean represents singularity. This means that the joint angular velocities become infinite when the determinant of the Jacobian matrix component becomes zero. This is when angular position of the second elbow joint q_2 is either 0^oor180^o which leads to the loss of solution number of the inverse kinematics.

3.3.2 Inverse Kinematics

In the previous we section showed how to determine the end-effectors' position and orientation in terms of the joint variables. This section is concerned with the inverse problem of finding the joint variables in terms of the end-effector position and orientation. This is the problem of inverse kinematics, and it is, in general, more difficult than the forward kinematics problem.

We begin by formulating the general inverse kinematics problem. Following this, we describe the principle of kinematic decoupling and how it can be used to simplify the inverse kinematics of most modern manipulators. Using kinematic decoupling, we can consider the position and orientation problems independently.

3.4 The General Inverse Kinematics Problem

Here it is desired to find the angles q_1 and $q_2$corresponding to the given end effector's position and orientation for the planar motion the position and orientation of the end effector can be specified by the origin of the frame. i.e. (p_x ,p_y) and the orientation of the frame attached to the end effector with respect to the x axis. Hence they are specified as the input. The solutions are the obtained using two approaches. In this thesis we use the geometric solution approach.

Squaring both side of the above equation (3.8) and (3.9) of position and orientation gives

$$c_{q2} = \frac{x^2+y^2-a_1^2-a_2^2}{2a_1a_2} \tag{3.13}$$

$$\tan_{q1} = \frac{y(a_2c_{q2}+a_1)-xa_1s_{q2})}{x(a_1+a_2c_{q2})+ya_2s_{q2}} \tag{3.14}$$

Joint velocity and end-effectors velocity has constraint and expressed as

$$\dot{q} = J^{-1}\dot{X} \tag{3.15}$$

$$J^{-1}=\begin{bmatrix} c_{q12}/a_1s_{q2} & s_{q12}/a_2s_{q2} \\ -(a_2c_{q12}+a_1c_{q1})/a_1a_2s_{q2} & -(a_2s_{q12}+a_1s_{q1})/a_1a_2s_{q2} \end{bmatrix}$$

3.5 Dynamic analysis and forces

In the previous section we studied the kinematics position and differential motion of robot. Dynamics of robot related to the accelerations, loads, masses and inertias. In order to be able to accelerate a mass we need to exert a force on it. Similarly to cause an angular acceleration in a rotation body a torque must be exerted on it.

To accelerate a robot's links, it is necessary to have actuators capable of exerting large enough forces and torques on the links and joints to move them at a desired acceleration and velocity Otherwise the links may not be moving as fast as necessary and consequently the robot may not maintain its desired positional accuracy. To calculate how strong each actuator be, it is necessary to determine the dynamic relationships that govern the motion of the robot.

These relationships are the force mass acceleration and the torque inertia angular acceleration dynamic equations. Based on this equation and considering the external loads on the robot, the designer can calculate the largest loads to which the actuators may be subjected, thereby designing the actuators to be able to deliver the necessary forces and toques or volts [23].

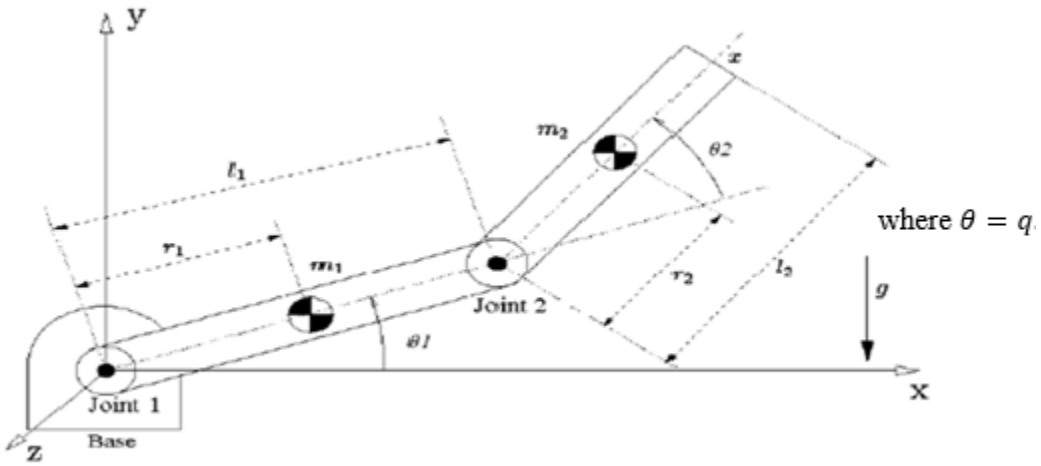

Figure 3.5: Robot manipulator with two degrees of freedom [24]

The practical parameters of a manipulator, meaning and value of the physical of parameters are as listed in Table 3.2 [22]

Table 3.2 List of Parameters of Robot Arm

Parameter	Notation	value
Length link 1	l_1	0.45m
Length link 2	l_2	0.45m
Mass link 1	m_1	23.902kg
Mass link 2	m_2	3.906kg
Link (1) center of mass	$r_1 = l_{c1}$	0.091m
Link (2) center of mass	$r_2 = l_{c2}$	0.048m
Inertia link 1	I_1	1.266kgm^2
Inertia link 2	I_2	0.093kgm^2
Gravity acceleration	g	9.81m/s^2

Using the Lagrangian method, we can derive the dynamic equation of motion for the two degree of freedom robot arm of the above Figure 3.5 by considering that the center of mass for each link is at the center of the link and the moments of inertia are I_1 and I_2 for link 1 and link 2 respectively [24].The dynamics of a simple manipulator is worked out to illustrate the Lagrange-Euler formulation to clarify the problems involved in dynamic modeling. For the manipulator links 1 and 2, joint variables are q_1 and q_2 link lengths are $a_1$and $a_2$and mass of links are m_1 and m_2 and $r_1$and r_2 are the distance between the joint and center of mass of the link 1 and 2 respectively which is shown in the Figure 3.5. The linear and angular velocities are q_1, q_2, $\dot{q}_1$, and $\dot{q}_2$ respectively. In this case, the datum (zero potential energy) is chosen at the axis of rotation "o'' The joint variable is

$$q = [q_1 \quad q_2]^T \tag{3.16}$$

And the generalized torque vector is

$$\tau = [\tau_1 \quad \tau_2]^T \tag{3.17}$$

With τ_1 and τ_2 torques supplied by the actuators

3.5.1 Kinematics and Potential Energy

For link 1 the kinematics and potential energies are

$K_1 = \frac{1}{2} m_1 I_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2$ (3.18)

$P_1 = -m_1 g I_{c1} sinq_1$ (3.19)

For link 2 the position and velocity of the center of mass are

$x_2 = a_1 cosq_1 + I_{c2}\cos(q_1 + q_2)$ (3.20)

$y_2 = a_1 sinq_1 + I_{c2} sin(q_1 + q_2)$ (3.21)

$\dot{x}_2 = -a_1 \dot{q}_1 sin\theta_1 - I_{c2}(\dot{q}_1 + \dot{q}_2)\sin(q_1 + q_2)$ (3.22)

$\dot{y}_2 = a_1 \dot{q}_1 cosq_1 + I_{c2}(\dot{q}_1 + \dot{q}_2)\cos(q_1 + q_2)$ (3.23)

So that the velocity squared is

$v^2 = \dot{x}^2 + \dot{y}^2$

$v^2 = a_1^2 \dot{q}_1^2 + I_{c2}^2(\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_1 + \dot{q}_2) + 2a_1 I_{c2}(\dot{q}_1^2 + \dot{q}_1 \dot{q}_2)(cosq_1 cosq_{12} + sinq_1 sinq_{12})$

$= a_1^2 \dot{q}_1^2 + I_{c2}^2(\dot{q}_1^2 + \dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + 2a_1 I_{c2} cosq_2(\dot{q}_1^2 + \dot{q}_1 \dot{q}_2)$ (3.24)

Then the kinetic energy for the second mass is

$$K_2 = \frac{1}{2} m_2 v^2 + \frac{1}{2} I_2(\dot{q}_1 + \dot{q}_2)^2$$

Substituting for v^2 from (3.24) given:

$K_2 = \frac{1}{2} m_2 a_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 I_{c2}^2(\dot{q}_1^2 + \dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + m_2 a_1 I_{c2} cosq_2(\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) + \frac{1}{2} I_2(\dot{q}_1 + \dot{q}_2)^2$

(3.25)

The potential energy for link 2 is

$P_2 = -m_2 g a_1 sinq_1 - m_2 g I_{c2}\sin(q_1 + q_2)$ (3.26)

3.5.2 Lagrangian Equation

L = K – P (3.27)

First, the kinetic energy of link one and two of the system are calculated as.

$K = K_1 + K_2$ (3.28-a)

And the kinematic energy is

$K = \frac{1}{2} m_2 a_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 I_{c2}^2(\dot{q}_1^2 + \dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + m_2 a_1 I_{c2} cosq_2(\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) + \frac{1}{2} I_2(\dot{q}_1 + \dot{q}_2)^2 +$

$\frac{1}{2} m_1 I_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2$ (3.28-b)

The potential energy of the system can be written as

$P = P_1 + P_2 = -m_1 g I_{c1} sinq_1 - m_2 g(a_1 \sin q_1 + I_{c2}\sin(q_1 + q_2))$ (3.30c)

The lagrangian's equation for the system is

$$L = \frac{1}{2}m_2a_1^2\dot{q}_1^2 + \frac{1}{2}m_2I_{c2}^2(\dot{q}_1^2 + \dot{q}_2^2 + 2\dot{q}_1\dot{q}_2) + m_2a_1I_{c2}cosq_2(\dot{q}_1^2 + \dot{q}_1\dot{q}_2) + \frac{1}{2}I_2(\dot{q}_1 + \dot{q}_2)^2 +$$

$$\frac{1}{2}m_1I_{c1}^2\dot{q}_1^2 + \frac{1}{2}I_1\dot{q}_1^2 + m_1gI_{c1}sinq_1 + m_2g(a_1\sin q_1 + I_{c2}\sin(q_1 + q_2)) \qquad (3.29)$$

The derivatives of the lagrangian from equation are

$$\frac{\partial L}{\partial \dot{q}_1} = m_2a_1^2\dot{q}_1 + m_2I_{c2}^2(\dot{q}_1 + \dot{q}_2) + 2m_2a_1I_{c2}cosq_2\dot{q}_1 + m_2a_1I_{c2}cosq_2\dot{q}_2 + I_2\dot{q}_1 + I_2(\dot{q}_1 +$$

$$\dot{q}_2) + I_1\dot{q}_1 + m_1I_{c1}^2\dot{q}_1$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) = [m_1I_{c1}^2 + m_2a_1^2 + m_2I_{c2}^2 + 2m_2a_1I_{c2}C_2 + I_1 + I_2]\ddot{q}_1 + [m_2I_{c2}^2 + m_2a_1I_{c2}cosq_2$$

$$+ I_2]\ddot{q}_2 - 2m_2a_1I_{c2}sinq_2\dot{q}_1\dot{q}_2 - m_2a_1I_{c2}sinq_2\dot{q}_2^2$$

$$\frac{\partial L}{\partial q_1} = -m_1gI_{c1}cosq_1 - m_2g(a_1\cos q_1 + I_{c2}\cos(q_1 + q_2)$$

From above derivation, the first equation of motion is given as:

$$\tau_1 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) - \frac{\partial L}{\partial q_1}$$

$$\tau_1 = [m_1I_{c1}^2 + m_2(I_{c2}^2 + a_1^2 + 2a_1I_{c2}C_2) + I_2 + I_1]\ddot{q}_1 + [m_2(I_{c2}^2 + m_2a_1I_{c2}C_2) + I_2]\ddot{q}_2 -$$

$$m_2a_1I_{c2}S_2(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) + m_1gI_{c1}cosq_1 + m_2g(a_1\cos q_1 + I_{c2}\cos q_1) \qquad (3.30)$$

Similarly, $\frac{\partial L}{\partial \dot{q}_2} = m_2I_{c2}^2(\dot{q}_1 + \dot{q}_2) + m_2a_1I_{c2}C_2\dot{q}_1 + I_2(\dot{q}_1 + \dot{q}_2)$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) = m_2I_{c2}^2(\ddot{q}_1 + \ddot{q}_2) + m_2a_1I_{c2}C_2\ddot{q}_1 - m_2a_1I_{c2}S_2\dot{q}_1\dot{q}_2 + I_2(\ddot{q}_1 + \ddot{q}_2)$$

$$\frac{\partial L}{\partial q_2} = m_2a_1I_{c2}S_2(\dot{q}_1^2 + \dot{q}_1\dot{q}_2) - m_2gI_{c2}C_{12}$$

$$\tau_2 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) - \frac{\partial L}{\partial q_2}$$

$$\tau_2 = [m_2(I_{c2}^2 + a_1I_{c2}C_2) + I_2]\ddot{q}_1 + (m_2I_{c2}^2 + I_2)\ddot{q}_2 + m_2a_1I_{c2}S_2\dot{q}_1^2 + m_2ga_2C_{12} \qquad (3.31)$$

Finally, according to Lagrange's equation, the arm dynamics are given by the two coupled nonlinear differential equations of (3.30) and (3.31) [24].

3.6 Manipulator Dynamics

Writing the arm dynamics in vector form yields

$$M(q)\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} + \begin{bmatrix}-m_2a_1I_{c2}S_2(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2)\\ m_2a_1I_{c2}\dot{q}_1^2sinq_2\end{bmatrix} + \begin{bmatrix}m_1gI_{c1}cosq_1 + m_2g(a_1\cos q_1 + I_{c2}\cos q_1)\\ m_2ga_2\cos(q_1 + q_2)\end{bmatrix}$$

$$= \begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}$$

Robotic manipulator with n rigid degrees of freedom is characterized by a set of n generalized coordinates $q^T = [q_1, q_2 \dots \dots q_n]$ called the joint space. The dynamic equation governing the nonlinear model of n-DOF robotic manipulator is given as [24]

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau \tag{3.32}$$

Where

$$M(q) = \begin{bmatrix} m_1 I_{c1}^2 + m_2(I_{c2}^2 + a_2^2) + 2m_2 a_1 I_{c2} C_2 + I_2 + I_1 & m_2 I_{c2}^2 + I_2 + m_2 a_1 I_{c2} C_2 \\ m_2 I_{c2}^2 + I_2 + m_2 a_1 I_{c2} cos\theta_2 & m_2 I_{c2}^2 + I_2 \end{bmatrix} \tag{3.33}$$

$C_i = cosq_i$ and $S_i = sinq_i$

$$\begin{bmatrix}\tau_1 \\ \tau_2\end{bmatrix} = \begin{bmatrix} [m_1 I_{c1}^2 + m_2(I_{c2}^2 + a_2^2) + 2m_2 a_1 I_{c2} C_2 + I_2 + I_1 & m_2 I_{c2}^2 + I_2 + m_2 a_1 I_{c2} C_2 \\ m_2 I_{c2}^2 + I_2 + m_2 a_1 I_{c2} C_2 & m_2 I_{c2}^2 + I_2 \end{bmatrix} \begin{bmatrix}\ddot{q}_1 \\ \ddot{q}_2\end{bmatrix}$$
$$+ \begin{bmatrix} 0 & -m_2 a_1 I_{c2} S_2 \\ m_2 a_1 I_{c2} S_2 & 0 \end{bmatrix} \begin{bmatrix}\dot{q}_1^2 \\ \dot{q}_2^2\end{bmatrix} + \begin{bmatrix} -m_2 a_1 I_{c2} S_2 & -m_2 a_1 I_{c2} S_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix}\dot{q}_1 \dot{q}_2 \\ \dot{q}_2 \dot{q}_1\end{bmatrix}$$
$$+ \begin{bmatrix} m_1 g I_{c1} C_1 + m_2 g(a_1 C_1 + I_{c2} C_{12}) \\ m_2 g I_{c2}\, C_{12} \end{bmatrix}$$

Property 1: the inertia matrix $M\ (q)$ is symmetric and positive definite, $M^T = M$

Property2: the matrix of $(\dot{M} - 2C)$ is skew-symmetric, i.e. for any vector of X, we have $X^T(\dot{M} - 2C)\ X=0$.

Using table 3.2 values the dynamic equation of the robot can be written as:

$$\begin{bmatrix}\tau_1 \\ \tau_2\end{bmatrix} = \begin{bmatrix} (2.357 + 0.168C_2 & 0.102 + 0.084C_2) \\ (0.102 + 0.084C_2) & 0.102 \end{bmatrix} \begin{bmatrix}\ddot{q}_1 \\ \ddot{q}_2\end{bmatrix} + \begin{bmatrix} 0 & -0.084S_2 \\ 0.084S_2 & 0 \end{bmatrix} \begin{bmatrix}\dot{q}_1^2 \\ \dot{q}_2^2\end{bmatrix} +$$
$$\begin{bmatrix} -0.084S_2 & -0.084S_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix}\dot{q}_1 \dot{q}_2 \\ \dot{q}_2 \dot{q}_1\end{bmatrix} + \begin{bmatrix} 38.58S_1 + 1.84S_{12} \\ 1.84S_{12} \end{bmatrix} \tag{3.34}$$

3.7 Considered Disturbances

As an industrial robot transfers objects of different masses, the dynamics of manipulator will be changed. Therefore, the manipulator model involves uncertainties. Moreover, the approximated friction is included in the dynamic of (3.35). Hence it is not repeatable to have a perfect model of friction. The manipulator is subject to external disturbances and approximate friction [25]. We can modify (3.32) to include friction $(F(\dot{q}))$, external dynamics and other left terms as a global uncertain term denoted by vector(τ_d).

Some of this effect can be modeled, identified, and compensated for the control algorithms, our goal is to design sliding mode controller to provide high reduction of un-modeled disturbances and robustness to model errors. Unknown disturbances and model errors can be divided into two:

- ✓ The sum of disturbances originating at motor.
- ✓ The sum of disturbances acting at the link.

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) + F(\dot{q}) + \tau_d = \tau \quad (3.35)$$

The external disturbances can be considered as external forces injected into the robotic system, and are supposed to have 25% of the desired trajectory with the following expression.

$$\tau_d = [0.25sint \quad 0.25sint]^T \quad (3.36)$$

Also, the friction term is considered as [25].

$$F(\dot{q}) = [20\dot{q}_1 + 0.8sng\dot{q}_1 \quad 4\dot{q}_2 + 0.16sgn(\dot{q}_2)]^T \quad (3.37)$$

Representing the above equation (3.34) with friction is expressed,

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}(2.357 + 0.168\,cos(q_1) & 0.102 + 0.084cos(q_1)\\ (0.102 + 0.084cos(q_1) & 0.102\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} +$$

$$\begin{bmatrix}-0.168sin(q_2)\dot{q}_2 + 20 + \frac{0.8sgn(\dot{q}_1)}{\dot{q}_1} & -0.084sin(q_2)\dot{q}_2\\ 0.084sin(\dot{q}_2)\dot{q}_1 + 4 + \frac{0.16sgn(\dot{q}_2)}{\dot{q}_2} & 0\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} +$$

$$\begin{bmatrix}\frac{38.58cos(q_1)+1.84cos(q_1+q_2)}{q_1}\\ \frac{1.84cos(q_1+q_2)}{q_1}\end{bmatrix}[q_1] \quad (3.38)$$

3.8 Actuators Dynamics of Robot

Permanent Magnet (PM) motors are by far the biggest user of permanent magnet materials capturing 60 percent of the PM market. Also the utilization of modern control strategies has shown a new perspective in the use of PM motors in high performance applications including factory automation, robotics, aerospace, etc. [24]. The Permanent Magnet Synchronous Motors (PMSM) is high-performance electromechanical motion devices which replace traditional dc servomotors and fractional horsepower induction machines because of their high performance capability. The necessity for high performance in PMSM systems increases as the demand for precision controls [26].The mathematical equations describing electrical and mechanical dynamics of a PMSM are as follows.

The stator of the PMSM and the wound rotor synchronous motor is similar. In addition there is no difference between the back electromotive force produced by permanent magnet and that produced by an excited coil. Hence the mathematical model of a PMSM is similar to that of the wound rotor synchronous motor. The following assumptions are made in derivation.

Assumptions

- Stator winding produces distribution magneto motive force. Space harmonics in the air gaps are neglected.
- Air gap reluctance has a constant component and sinusoidal varying component.
- Balanced three phase supply voltage is considered, although magnetic saturation is considered, eddy current and hysteresis effects are neglected.
- Damper windings are not considered.

The parameters of PMSM are as listed in table 3.3

Table 3.3: Parameters of PMSM [27]

Parameters	*Jonits*		
	Notation	1	2
Moment of Inertia	$J_i \; in \; kg.m^2$	0.0003616	0.0003616
Stator Winding Resistance	$R_{si} \; in \; \Omega$	0.62	0.62
q axis Inductance	$L_i \; in \; mH$	0.002075	0.002075
Motor Damping Constant	$B_i in \; N.m.s$	0.0000944	0.0000944
Flux Linkage established By Magnet	$\lambda_{mi} \; in \; v.s$	0.08627	0.08627
Pole Pairs	P	4	4
tourqe Constnt	k_i	1.026	1.026

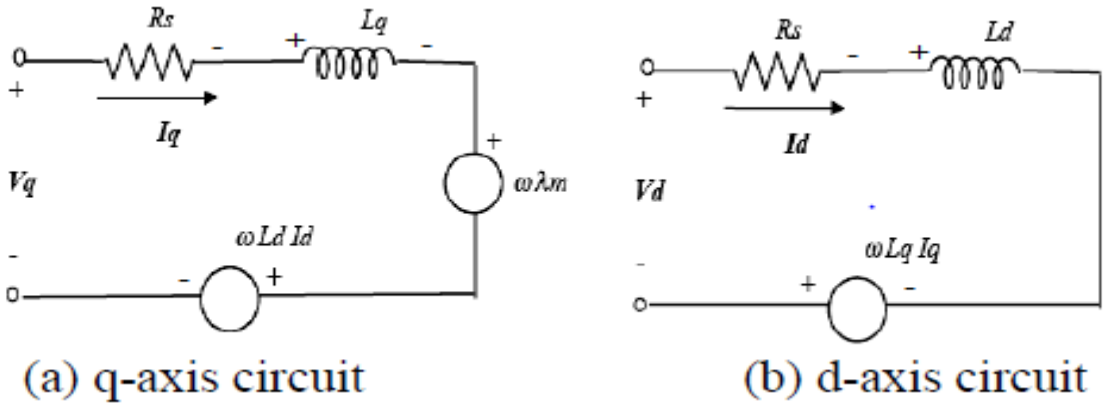

Figure 3.6: Equivalent Circuit of a PM Synchronous Motor [28]

From the above Figure 3.6 we can drive the equation of the PMSM as

$$V_d = R_s i_d + \frac{d}{dt}\lambda_d - \omega_I \lambda_q \quad (3.39)$$

$$V_q = R_s i_q + \frac{d}{dt}\lambda_q - \omega_I \lambda_d \quad (3.40)$$

Where

$$\lambda_q = L_q i_q \quad (3.41)$$

And

$$\lambda_d = L_d i_d + \lambda_{af} \quad (3.42)$$

λ_{fa} Is the magnet mutual flux linkage and the motor is run always at the synchronous speed so that the slip frequency speed is zero. $i.e\ \omega_I = 0$.

The input power P_i is represented as

$$P_i = V_a i_a + V_b i_b + V_c i_c \quad (3.43)$$

While in terms of d, q variables

$$\text{Power} = 3(V_d i_d + V_q i_q)/2 \quad (3.44)$$

Note that in practice, magnetic circuits are subjected to saturation as current is increases. Especially, when q_i is increased, the value of L_q is decreased and L_m and L_d are subjected to armature reaction. Since i_d is maintained to zero or negative value (demagnetizing) in most operating conditions, saturation of L_d is rarely occurs and V_d become amost zero and finally we have V_q [28].

The electric torque τ produced by motor

$$\tau = 3P\left[\lambda_m i_q + \left(L_d - L_q\right) i_d i_q\right]/2 \quad (3.43)$$

3.9 System model [29]

In the proposed approach, each joint of the manipulator is driven by a permanent magnet synchronous motor in the control system. The inserted torque on the joint to drive the manipulator is the load torque of motor, which is considered in a dynamic equation formed as

$$\tau = \tau_L + B_m\omega_r + J_m\frac{d}{dt}\omega_r \tag{3.44}$$

$$\frac{d\theta}{dt} = \omega_r \; angular \; velocity$$

And the above equation can be written in

$$\tau = \tau_L + B_m\omega_r + J_m\frac{d}{dt}\omega_r \Rightarrow \tau = \tau_L + B_m\dot{\theta} + J_m\ddot{\theta} \tag{3.45}$$

It is from equation (3.46) that the produced torque is composed of two distinct mechanisms. The first term correspond to "the mutual reaction torque" occurring between i_q and the permanent magnet. While the second term correspond to "the reluctance torque" due to the differences in d-axis and q-axis reluctance or (inductance) for constant flux operation when i_d equals zero, the electric torque τ

$$\tau = \frac{3P\lambda_m i_q}{2} = K_i i_q \tag{3.46}$$

Where K_i is the motor torque Constant. This torque equation is resembles that of the regular dc machine and hence provides ease of control [28].

τ_L And τ represent the generated motor torque and the load torque, respectively. With the purpose of increasing motion speed of the manipulators, motors are equipped with the high reduction gears. The reduction gear relates the motor position to the joint position as

$q = g_r\theta$ And $\tau_L = g_r\tau$

Where

g_r is the diagonal matrix of reduction ratio. In the following a practical constraint is considered. The dynamical equation of a two-link robot manipulator is in the standard form is rewritten as:

$$M(q)\ddot{q} + B(q)(q,\dot{q}) + C(q)(\dot{q}^2) + G(q) + F(\dot{q}) + \tau_d = \tau \tag{3.47}$$

It should be noted that, the applicable control input for driving robot arm is the voltage of the motors. So by using equations (3.42)-(3.48) and neglecting the inductance L_q, because of its tiny amount, the following equation is achieved [29]. Taking equation (3.46) and substituting for T_L by torque of the manipulator (τ) given as and using the reduction gear ratio $\tau_L = g_r\tau_e$ and

$q = g_r\theta$

$$\tau = g_r[M(q)\ddot{q} + B(q)(q,\dot{q}) + C(q)(\dot{q}^2) + G(q) + F(\dot{q}) + \tau_d] + g_r^{-1}[B_m\dot{q} + J_m\ddot{q}] \quad (3.48)$$

Substituting the electric torque equation (3.47) in to (3.49) and solving for input voltage from equation (3.39) gives the following equation.

$$RK_i^{-1}\{[J_m g_r^{-1} + g_r M]\ddot{q} + (B_m g_r^{-1} + g_r[B(q)(q,\dot{q}) + C(q)(\dot{q}^2)]) + g_r G + g_r F(\dot{q}) + g_r\tau_d\} = V \quad (3.50)$$

All dynamical terms are transferred to the electrical voltage of motors as stated by (3.49). Dynamics of error is free of manipulator dynamics so it introduces a robust control system. In this technical point of view, the proposed control law has an advantage to solve such uncertain dynamics which many control approaches are involved. Therefore, the proposed control law is advantageous comparing to the robust control laws which require either the manipulator dynamics or the manipulator parameters. Also, for tracking purpose it is more efficient than the PID control system which it is well known as a simple and useful method to control manipulators. Many industrial robots use a form of so called PID control law.

The standard form of the n-link robot arm can be expressed in the compact form as [29].

$$D\ddot{q} + H(q,\dot{q})\dot{q} + d = U \quad (3.51)$$

U is the control command and the other parameters are

$$D = RK_i^{-1}\{[J_m g_r^{-1} + g_r M] \quad (3.52)$$

$$H = RK_i^{-1}\{B_m g_r^{-1} + g_r[B(q)(q,\dot{q}) + C(q)(\dot{q}^2)]\}z + RK_i^{-1}\{(B_m g_r^{-1}) + g_r G \quad (3.53)$$

$$d = RK_i^{-1}\{g_r F(\dot{q}) + g_r\tau_d\} \quad (3.54)$$

Remark1: by noting that the parameters R, K_i, J_m and g_r are positive definite diagonal matrices, the matrix D is symmetric and positive definite

Remark2: From relations 3.52 and property 1, the matrix $\dot{D} - 2[B(q)(q,\dot{q}) + C(q)(\dot{q}^2)])$ is skew-symmetric too [25].

Using the values of all parameters given in table 3.2 and table 3.3 and gear ratio for motor of both joints are 60:1; the final model of the overall system equation of (3.51) can be written as

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} (0.0369 + 0.00168\cos(q_2)) & 0.00108 + 8.4e-4\sin(q_2) \\ (0.00108 + 8.4e-4os(q_2) & 0.0026 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -1.692e-3\sin(q_2)\dot{q}_2 + 0.2012 + \frac{0.001607\text{sgn}(\dot{q}_1)}{\dot{q}_1} & -8.4e-4\sin(q_2)\dot{q}_2 \\ 8.4e-4\sin(q_2)\dot{q}_2 + 0.040185 + \frac{0.008057\text{sgn}(\dot{q}_2)}{\dot{q}_2} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_2 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} \frac{0.3886\cos(q_1)+0.0185\cos(q_1+q_2)}{q_1} \\ \frac{0.0185\cos(q_1+\dot{q}_2)}{q_1} \end{bmatrix} \quad (3.55)$$

The nonlinear representation of the system can be expressed as the following

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = \begin{bmatrix} (0.0369 + 0.00168\cos(q_2)) & 0.00108 + 8.4e-4\sin(q_2) \\ (0.00108 + 8.4e-4os(q_2) & 0.0026 \end{bmatrix}^{-1} \left\{ \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} - \begin{bmatrix} -1.692e-3\sin(q_2)\dot{q}_2 + 0.2012 + \frac{0.001607\text{sgn}(\dot{q}_1)}{\dot{q}_1} & -8.4e-4\sin(q_2)\dot{q}_2 \\ 8.4e-4\sin(q_2)\dot{q}_2 + 0.040185 + \frac{0.008057\text{sgn}(\dot{q}_2)}{\dot{q}_2} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_2 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} \frac{0.3886\cos(q_1)+0.0185\cos(q_1+q_2)}{q_1} \\ \frac{0.0185\cos(q_1+\dot{q}_2)}{q_1} \end{bmatrix} \right\} \quad (3.56)$$

CHAPTER FOUR
SLIDING MODE CONTROL DESIGN

4.1 Introduction

Sliding mode control is a nonlinear control technique featuring remarkable properties of accuracy, robustness, and easy tuning and implementation. SMC systems are designed to drive the system states onto a particular surface in the state space, named sliding surface. Once the sliding surface is reached, sliding mode control keeps the states on the close neighborhood of the sliding surface. Hence the sliding mode control is a two part controller design. The first part involves the design of a sliding surface so that the sliding motion satisfies design specifications. The second is concerned with the selection of a control law that will make the switching surface attractive to the system state [30].

There are two main advantages of sliding mode control. First is that the dynamic behavior of the system may be tailored by the particular choice of the sliding function. Secondly, the closed loop response becomes totally insensitive to some particular uncertainties. This principle extends to model parameter uncertainties, disturbance and non-linearity that are bounded. From a practical point of view SMC allows for controlling nonlinear processes subject to external disturbances and heavy model uncertainties.

The motion of the system confined to the switching line or a surface is referred to as sliding. A sliding mode will exist in the vicinity of the switching surface the state velocity vectors are directed toward the surface. In such a case, the switching surface attracts trajectories when they are in its vicinity; and once a trajectory intersects the switching surface, it will stay on it thereafter. A surface s(x) = 0 is attractive if

1. Any trajectory starting on the surface remains there, and
2. Any trajectory starting outside the surface tends to it at least asymptotically.

Thus, for a sliding motion to occur we need [30]

$$\lim_{s\to 0^+} \dot{s} < 0 \; and \; \lim_{s\to 0^-} \dot{s} > 0 \text{ [30]} \tag{4.1}$$

They ensure that the motion of the state trajectory x on either side of the switching surface s(x) = 0 is toward the switching surface. The two conditions may be combined to give

$s\dot{s} < 0$ (4.2)

The neighborhood of the switching surface; however, that the closed-loop system is now modeled by differential equations with discontinuous right-hand sides. The modeling differential equations are discontinuous on the switching surface s(x) = 0. This is the reason we often refer to the switching surface as the discontinuity surface. A rigorous analysis of such systems cannot be carried out using methods of classical theory of differential equations. A solution to the differential equation $\dot{x}=f(x)$ with a continuous right-hand side is a function $x\ (t\ )$ that has a derivative and satisfies the differential equation everywhere on a given interval. In the case when the right-hand side of $\dot{x}=f(x)$is discontinuous [30].

Sliding mode control systems comprise a collection of different, usually quite simple, feedback control laws and a decision rule. Depending on the status of the system, a decision rule, often termed the switching function, determines which of the control laws is "on-line" at any one time. The transient dynamics of a variable structure control system consists of two modes: a "reaching mode" (or "non-sliding mode"), followed by a "sliding mode". Therefore the design of variable structure control variable structure control involves, first, the design of an appropriate n-dimensional switching function s(x) for a desired sliding mode dynamics, and second, the design of a control for the reaching mode such that a reaching condition is met. The desired sliding mode dynamics is usually a fast and stable error-free response void of overshoot (an asymptotic convergence to the final state will be achieved in sliding mode). For the reaching mode, the desired response usually is to reach the switching manifold, described by [31].

$$s(x) = C^T x = 0 \quad (4.3)$$

In finite time with small overshoot with respect to the switching manifold. For an n-input system, there are n switching functions and $2^n - 1$ sliding manifolds of different dimensions. The first m of them are designated as:

$$s_i = \{xs_i = c_i^T x = 0\}, i = 1, \dots, n, \quad (4.4)$$

Which may be called basic sliding manifolds since each of them is associated with a single switching function. The dynamical behavior of the system when confined to the surface is described as the ideal sliding motion. The advantages of obtaining such a motion are twofold: firstly there is a reduction in order and secondly the sliding motion is insensitive to parameter variations implicit in the input channels [32]. Proposed a new approach, based on a new method called the reaching law method, for the design of VSC of nonlinear systems. The method simultaneously takes care of ensuring the reaching condition, arranging the logic for the free-order switching, influencing the dynamic quality of the system during the reaching phase, and providing the means for controlling the chattering level. The procedure of using this method is straightforward and easy to carry out, even for nonlinear systems. The reaching law is a differential equation which specifies the dynamics of a switching function s(x). The differential equation of an asymptotically stable s(x) is itself a reaching condition. In addition, by the choice of the parameters in the differential equation, the dynamic quality of VSC system in the reaching mode can be controlled. A practical general form of the reaching law is:

$$\dot{s} = -Qsgn(s) - ks \tag{4.6}$$

Where

$$\text{Q=diag } [Q_1, \dots, Q_n], Q_i > 0$$

$$\text{Sgn(s)} = \text{diag}[sgn(s_1), \dots, sgn(s_n)]^T$$

$$\text{K=diag } [K_1, \dots, K_n], K_i > 0$$

$$\text{h(s)} = \text{diag}[h(s_1), \dots, h(s_n)]^T$$

$$s_i h_i(s_i) > 0, h_i(0) = 0$$

Three practical special cases of (3.5) are given below.

a) Constant rate reaching:

$\dot{s} = -Qsgn(s)$ [32] (4.7)

This law forces the switching variable s(x) to reach the switching manifolds S at a constant rate $|\dot{s}_i| = -Q_1$. The merit of this reaching law is its simplicity. But, if Q_1 is too small, the reaching time will be too long. On the other hand, a Q_1 too large will cause severe chattering.

b) Constant plus proportional rate reaching:

$\dot{s} = -Qsgn(s) - ks$ (4.8)

Clearly, by adding the proportional rate term -Ks, the state is forced to approach the switching manifolds faster when s is large. It can show that the reaching time for x to move from an initial state x_0 to the switching manifold s_i is finite, and is given by:

$T_i = \frac{1}{k_i} ln \frac{k_i|s_i|+q_i}{q_i}$ [32] (4.9)

c) Power rate reaching:

$\dot{s}_i = -k_i|s_i|^{\alpha} sgn(s_i)$, 0<α<1, i=1… n. [32] (4.10)

This reaching law increases the reaching speed when the state is far away from the switching manifold, but reduces the rate when the state is near the manifold. The result is a fast reaching and low chattering reaching mode.

$$s_i = s_{i0}\ to\ s_i = 0 \text{ yeildes}$$

$T_i = \frac{|s_{i0}|^{1-\alpha}}{(1-\alpha)k_i}, i = 1, \dots, n,$ (4.11)

The reaching time T_i is finite. Thus power rate reaching law gives a finite reaching time. In addition, because of the absence of the -Q sgn (s) term on the right-hand side of (4.11), this reaching law eliminates the chattering. Having selected the reaching law equation, the control law now be determined. Compute the time derivative of s(x) along the reaching mode trajectory, then from (4.5) and (4.6) [33].

$\dot{s}(x) = \frac{\partial s}{\partial x} A(x) + \frac{\partial s}{\partial x} B(x)u = -Qsgn(s) - Ks$ [32] (4.12)

Noting that the matrix $\frac{\partial s}{\partial x}B(x)$ is nonsingular, this equation is solved for the control law, giving:

$$u = \left[\frac{\partial s}{\partial x}B(x)\right]^{-1} \times \left[\frac{\partial s}{\partial x}A(x) + Qsgn(s) + Ks\right] \quad (4.13)$$

This control law appears independent of system perturbation and external disturbances, which is not realistic. In fact, the control u does depend on perturbation and disturbances, and it should include their parameters. A final but important note is that the control law (4.13), obtained via a reaching law, automatically leads to the free-order switching scheme. From the practical point of view, this scheme appears to be the most efficient. The principle of designing SMC law for arbitrary-order plants is to force the error and derivative of error of a variable to zero. The robot arm is to track a desired motion $q_d(t)$. Define an error vector:

$$\begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix} = \begin{bmatrix} q_{di} \\ \dot{q}_{di} \end{bmatrix} = \begin{bmatrix} q_{di} - x_i \\ \dot{q}_{di} - x_i \end{bmatrix} \quad (4.14)$$

And then define an n-dimensional vector switching function:

$$s(e) = Ce = [\lambda \quad I]\begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix} = \lambda e_i + \dot{e}_i \text{ [32]} \quad (4.15)$$

Where $\dot{e}$ is the tracking speed error and

$\boldsymbol{\lambda}$=diag $[\lambda_1, \dots, \lambda_n], \lambda_i > 0,$

Those determine the bandwidth of the system. Adopt the reaching law (4.8), and taking the time derivative of (4.15) gives:

$$\dot{s}(e_i) = \lambda\dot{e}_i + \ddot{e}_i = \lambda\dot{e}_i + \ddot{q}_{di} + D^{-1}(q) \times [H(q,\dot{q})(\lambda e_i + \dot{e}_i) + d_i - v_i] \quad (4.16)$$

And the final control law:

$$u = D(q)[Qsng(s) + ks + \lambda\dot{e}_i + \ddot{q}_{di}] + H(q,\dot{q})(\lambda e_i + \dot{e}_i) + d_i \quad (4.17)$$

Where u a new control input. Substituting (4.17) into (4.16)

$$|u| \leq U \tag{4.18}$$

In order to govern the system states $\dot{e}\ and\ e$ to reach the sliding surface s = 0 in a limited time and to remain there, the control law should be designed such that the following sliding condition is satisfied.

$$\frac{1}{2}\frac{d}{dt}[s^T D s] < -\eta(s^T s)^{\frac{1}{2}}, \eta > 0 \tag{4.19}$$

This aim is fulfilled in the following lemma.

Lemma 1: In the SMC design of a system with dynamical equation (3.50) and sliding surface (4.14), if the control input u is selected as (4.17), by considering U as (4.18) and $Q_{ii} = diag[Q_{11}, Q_{22}, \dots, Q_{nn}]$ with the following components:

$$Q_{ii} = [F + |k_v s| + T_d + \eta],\ i = 1,2, \dots, n \text{ [32]} \tag{4.20}$$

Then, the sliding condition (4.19) is satisfied by equation (4.18).

Proof: Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}[s^T D s] \tag{4.21}$$

Since D is positive definite, for $s \neq 0$ we have $V > 0$ and by taking derivative from relation (4.22) and regarding the symmetric property of D, it can be written

$$\dot{V} = \frac{1}{2}s^T \dot{D} s + s^T D \dot{s} \tag{4.22}$$

By substituting (4.16) into (4.22) and considering that $s^T(\dot{D} - 2H)s = 0$ we have:

$$\dot{V} = \frac{1}{2}s^T \dot{D} s - s^T H s + S^T(f + d - v) = S^T(u + d - v) \tag{4.23}$$

By replacing the relation (4.17) into (4.23), $\dot{V}$ can be rewritten as:

$$\dot{V} = S^T(u + d - U - k_v s - Q sgn(s)) = S^T(u + d - k_v s) - \sum_{i=i}^{n} Q_{ii}|s_i| \tag{4.24}$$

Since the following inequality (4.25) is valid and by regarding the relation (4.20), we have:

$$[U + |k_v s| + T_D + \eta] \geq |u + \tau_d - k_v s| \tag{4.25}$$

$$\dot{V} = -\sum_{i=1}^{n} \eta_i |s_i| \tag{4.26}$$

A typical choose in this thesis is saturation function instead of sign function. The use of sign function in the corrective control law leads to high oscillations in control effort which is undesired phenomenon and is called chattering.

$$sat\frac{s}{\varphi} = \begin{cases} 1 & s \geq 1 \\ \frac{s}{\varphi} & -k < s < \varphi \\ -1 & s \leq -1 \end{cases} \tag{4.13}$$

By this, there is a boundary layer φ around the sliding surface such that once the state trajectory reaches this layer, then it will be remaining there

4.2 PID controller design

The system equation from (3.51) is:

$$D\ddot{q} + H(q,\dot{q})\dot{q} + d = U$$

We can have from (4.9)

$$\ddot{q} = D^{-1}[-(H(q,\dot{q})\dot{q}] + \widehat{U}$$

With

$$\widehat{U} = D^{-1}U \Leftrightarrow U = D\widehat{U} \tag{4.14}$$

So we decouple the system to have the new (non-physical) input

$$\widehat{U} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4.15}$$

The physical control inputs to the system are:

$$\begin{bmatrix} u_{q1} \\ u_{q2} \end{bmatrix} = D \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4.16}$$

PID controller for the input would be

$$\widehat{U} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{matrix} K_{p1}(q_{d1} - q_{a1}) + K_{D1}(\dot{q}_{d1} - \dot{q}_{a1}) + K_{I1}\int(q_{d1} - q_{a1})dt \\ K_{p2}(q_{d2} - q_{a1}) \quad K_{D2}(\dot{q}_{d2} - \dot{q}_{d2}) + K_{I2}\int(q_{d2} - q_{a2})dt \end{matrix} \tag{4.17}$$

CHAPTER FIVE
SIMULATION RESULTS AND DISCUSSION

5.1 Introduction

In this chapter, the simulation is carried out using Level-2 MATLAB s-function and Simulink to verify the performance of the designed sliding mode controller and comparing the simulation results with the conventional PID controller. The Level-2 M-file S-function allow to create blocks that have all of the features and capabilities of Simulink built-in blocks, including multiple input and output ports, the ability to accept vector or matrix signals of any data type supported by Simulink, real or complex signals, signal frames, the ability to operate at multiple sample rates and simple. The tracking performance in SMC and PID without disturbance for sinusoidal trajectory was tested. Based on results, SMC has better performance than PID.

The robustness of the designed controller is tested based on the power of disturbance elimination and insensitive to parameter variation in the SMC and PID. The value of disturbance applied to the sine and cosine trajectory of joint variable is predefined as 25% of the input signal. The parameter variation as friction term is considered. Based on the result SMC is more robust than PID.

By trial & error the best values for the parameters was found to be

- ✓ For a SMC

  $\lambda_1 = 30, \lambda_2 = 8$ and $k_1 = 2\ and\ k_2 = 2$, Q_1=Q_2=200
- ✓ For a PID

$$K_{p1} = 2400, K_{p2} = 1350, K_{I1} = 200, K_{I2} = 8240, K_{D1} = 2850, K_{D2} = 44580$$

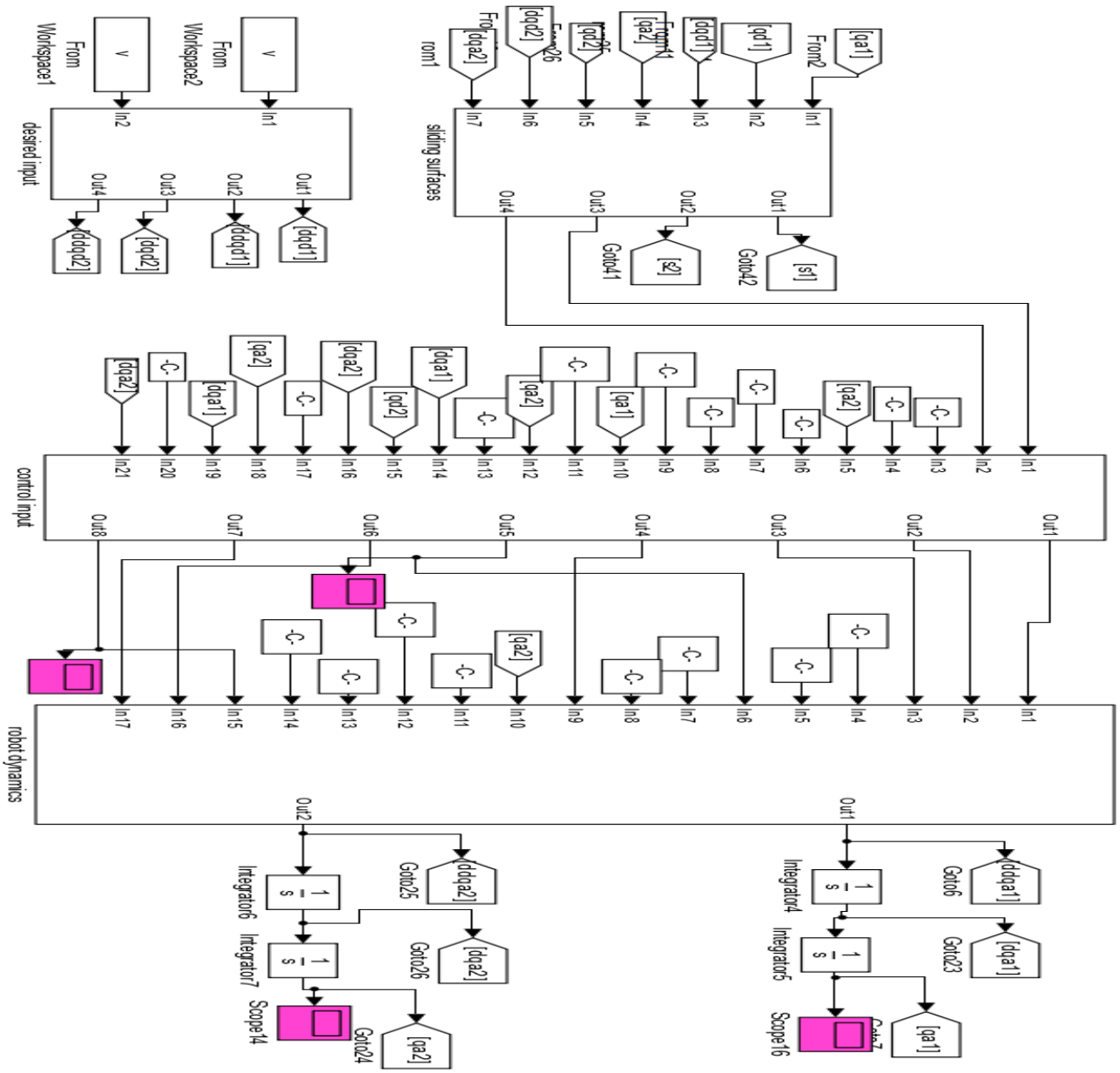

Figure 5.1: SMC Level-2 MATLAB s-function and Simulink model of the two DOF robot arm without disturbance and parameter variation

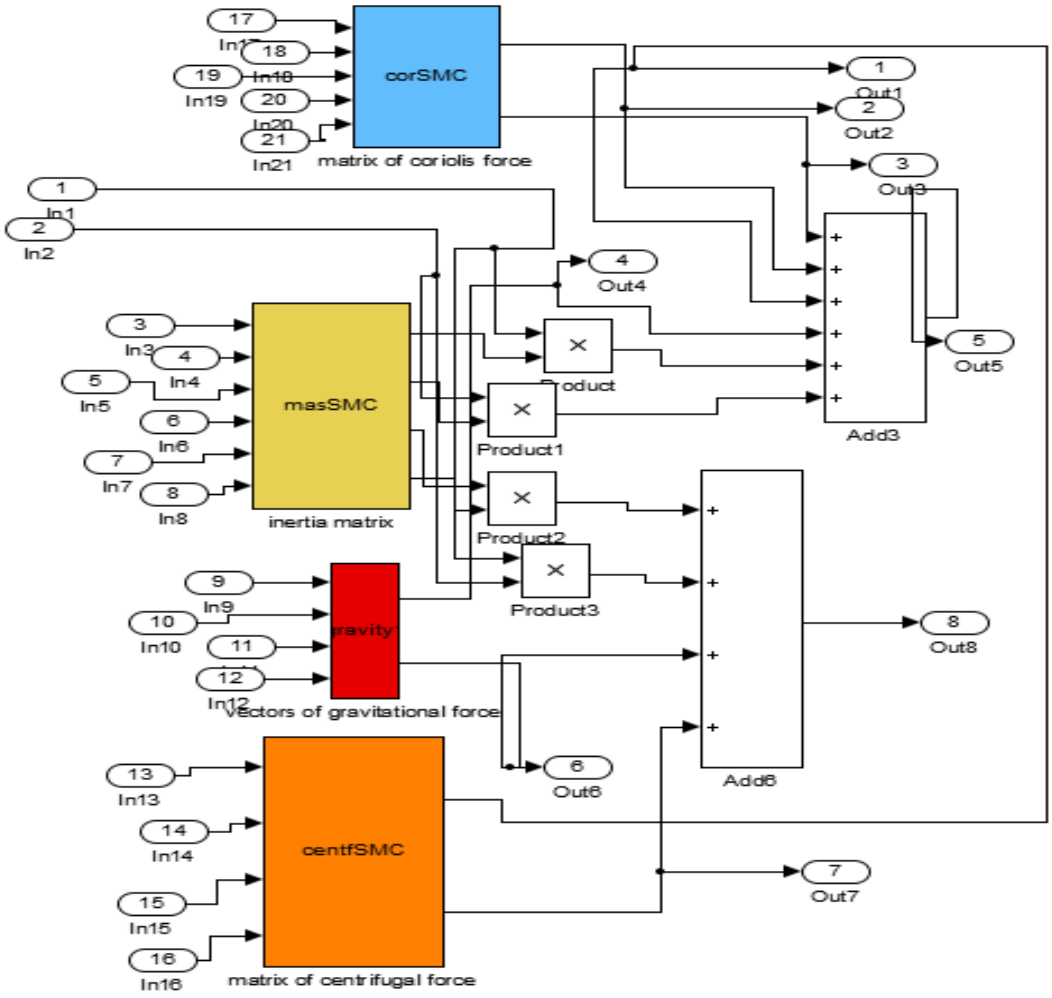

Figure 5.2: Control input subsystem components are inertia matrix, matrix of centrifugal and centripetal and vector of gravitational force

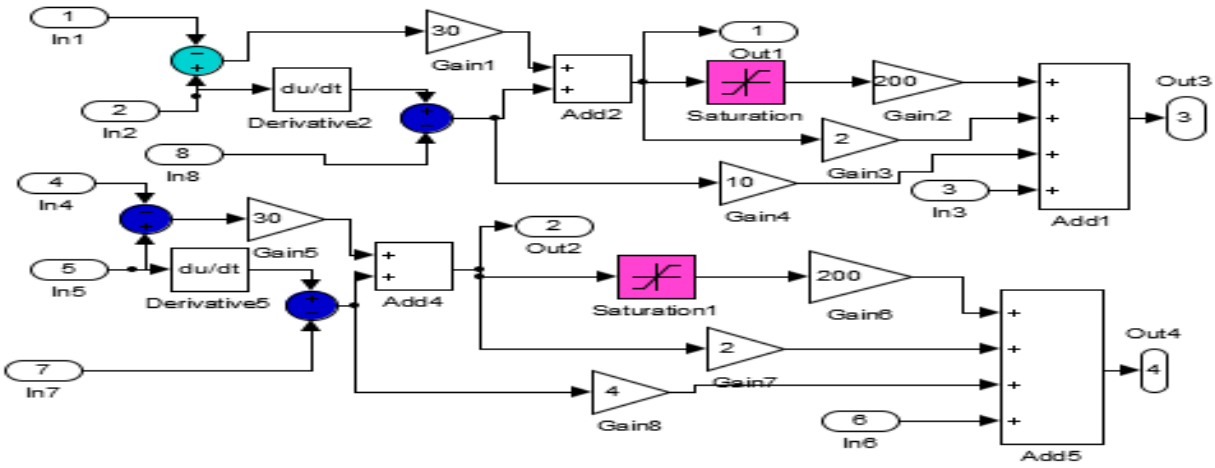

Figure 5.3: Simulation diagram of sliding surfaces

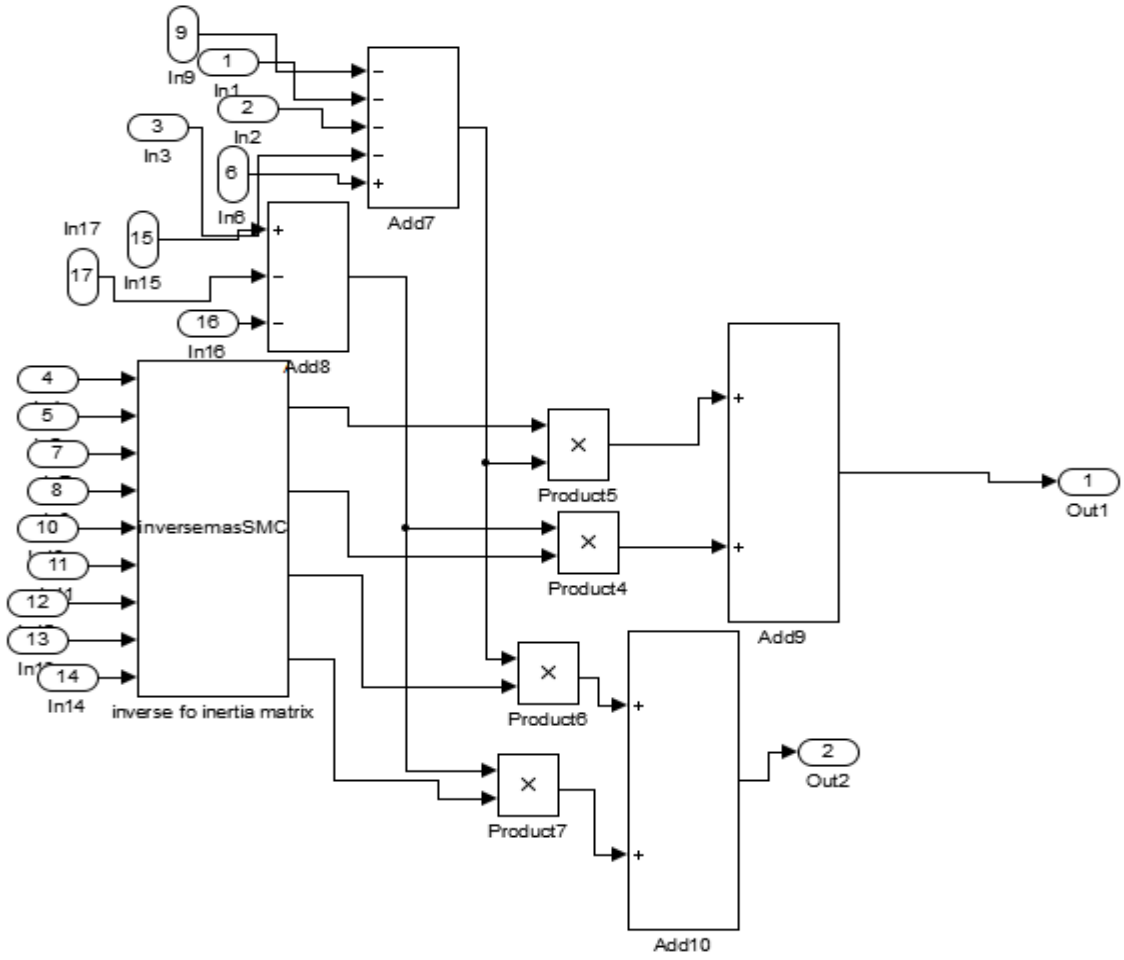

Figure 5.4: Simulation diagram of Robot dynamics subsystem components are inverse of inertia matrix and all control input components

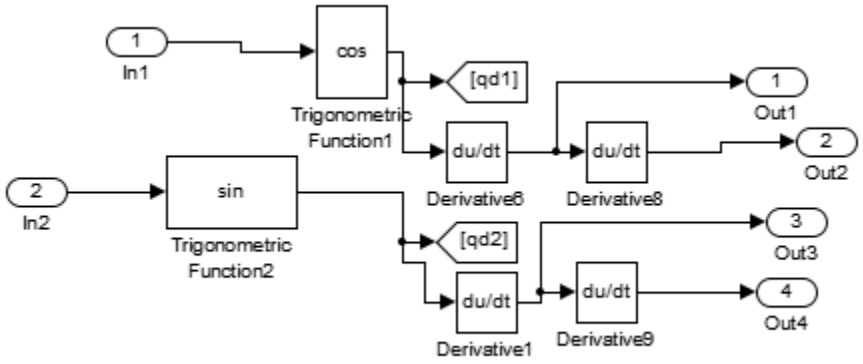

Figure 5.5: Simulation diagram of desired trajectory subsystem

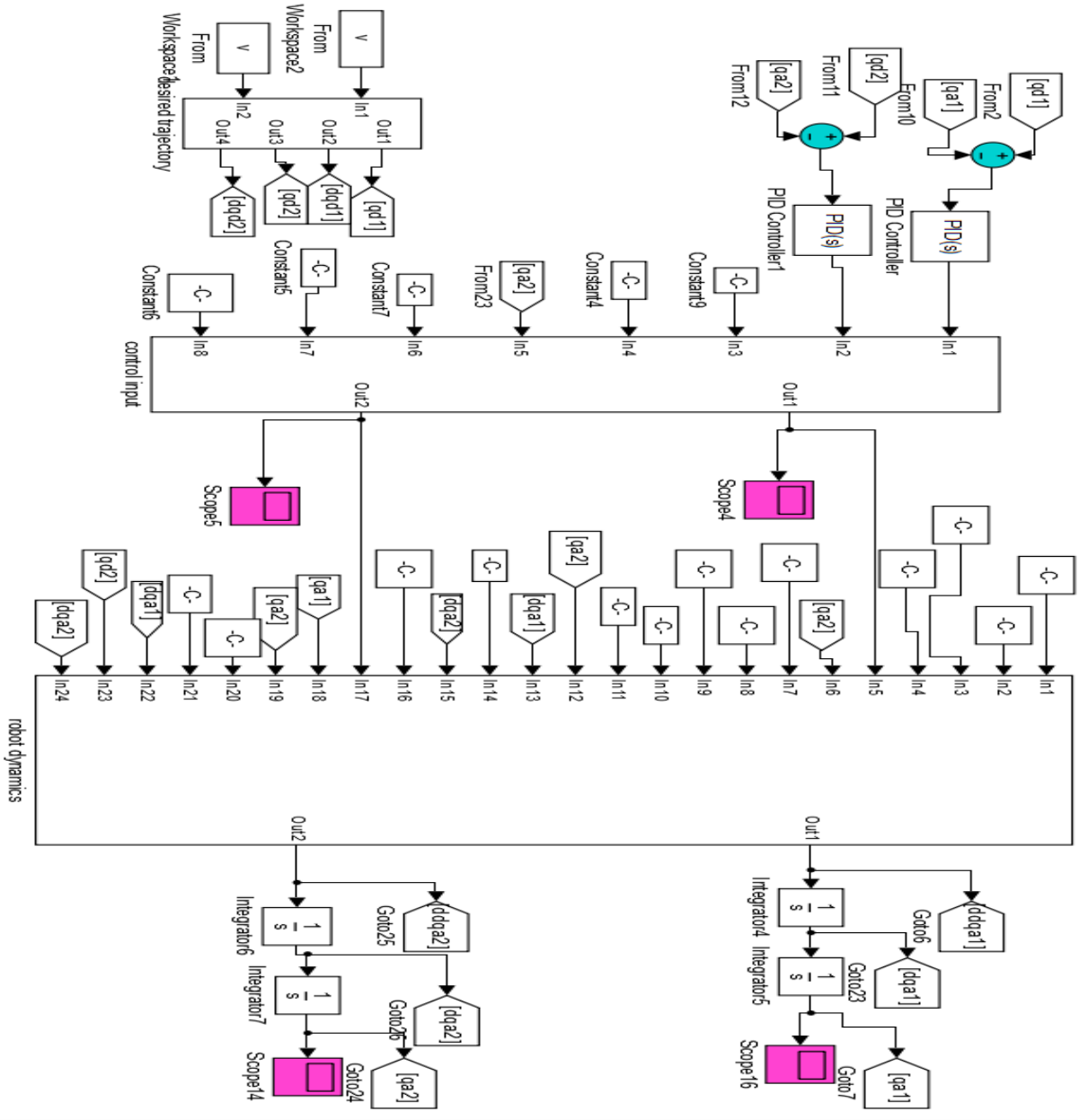

Figure 5.6: PID Level-2 MATLAB s-function and Simulink model of the two DOF robot arm without disturbance and parameter variation

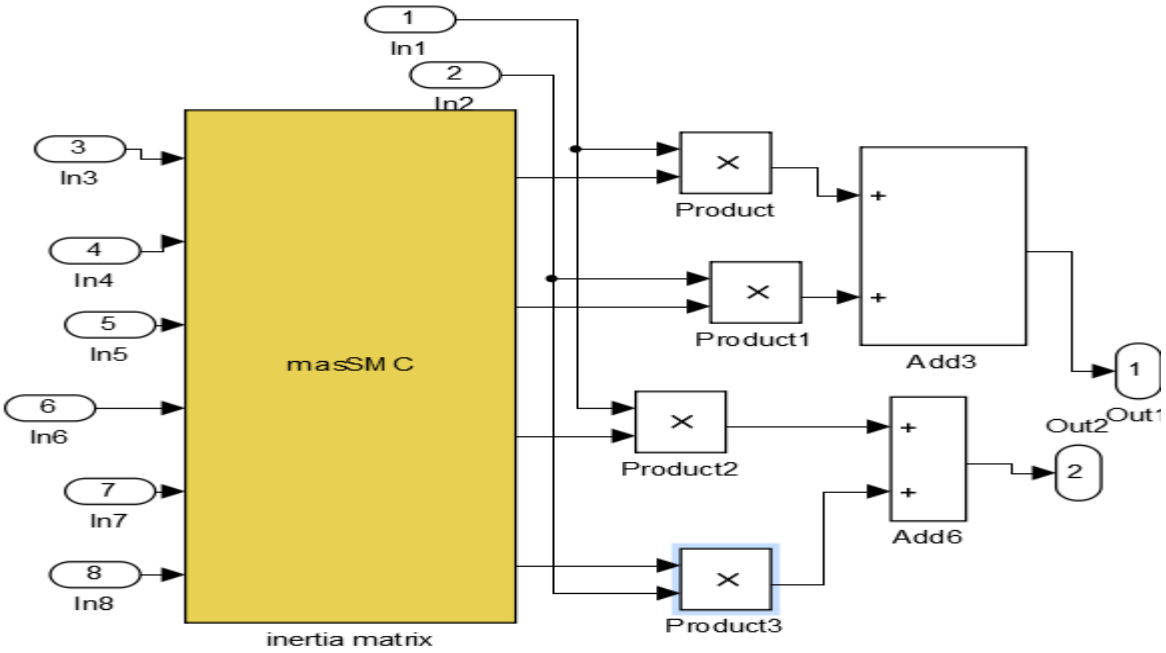

Figure 5.7: Simulation diagram of control input subsystem components are PID controller and inertia matrix

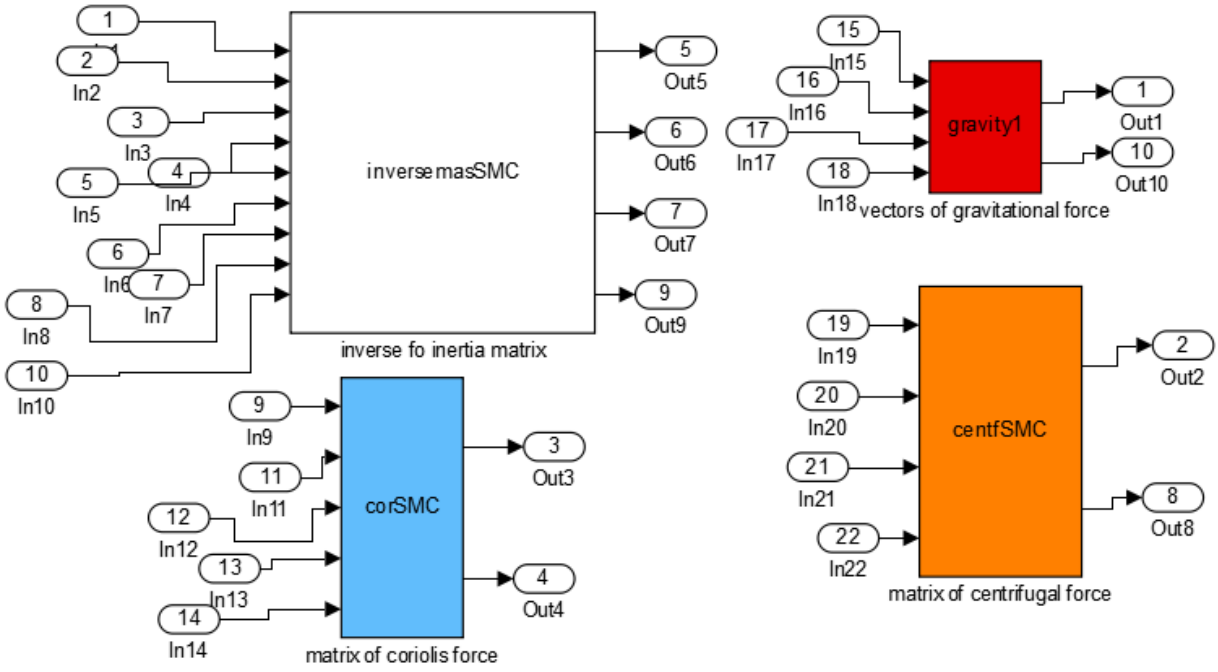

Figure 5.8: Simulation diagram of robot dynamics subsystem components in PID controller are inverse inertia matrix, vector of gravitational forces centripetal and centrifugal forces

5.2 Performance of sliding mode controller

The SMC response for the two degree of freedom robot is shown in figure 5.9 and 5.10 for angular position one and two respectively. Based on figure 5.9, 5.10, 5.11 and 5.12 the performance of SMC for angular position one and two are shown in table 5.4. Figure 5.11 and 5.12 shows the plot of angular position error of joint one and two which is 0.00029 rad/sec and 0.000098 rad/sec respectively and clearly show the steady state error Figure 5.13 and 5.14 below shows the plot of angular velocity of arm one and arm two respectively and the performance of each angular velocity is acceptable as shown in Figure 5.15 and figure 5.16 the angular velocity error of arm one and arm two is 0.0031 rad/sec.

Table 5.4: performance of SMC of angular position one and two

performance	SMC	
	For angular position of arm one	For angular position of arm two
Rising time (sec)	0.2	0.2
Settling time (sec)	0.45	0.65
Overshoot (%)	1.8	1.95
Steady state error (%)	0.029	0.0098

Figure 5.17 and 5.18 below shows the plot of sliding surface (s) and derivative of sliding surface respectively (ṡ). The sliding surface and derivative of sliding surface is smooth and converges to zero as expected. The reaching time to the sliding manifold is 2sec for both surfaces. Figure 5.19 and 5.20 shows control efforts of joint angle one and control effort of joint angle two for SMC respectively and the maximum voltage used in order to drive the robot arm one is 0.4v and 0.02v for arm two.

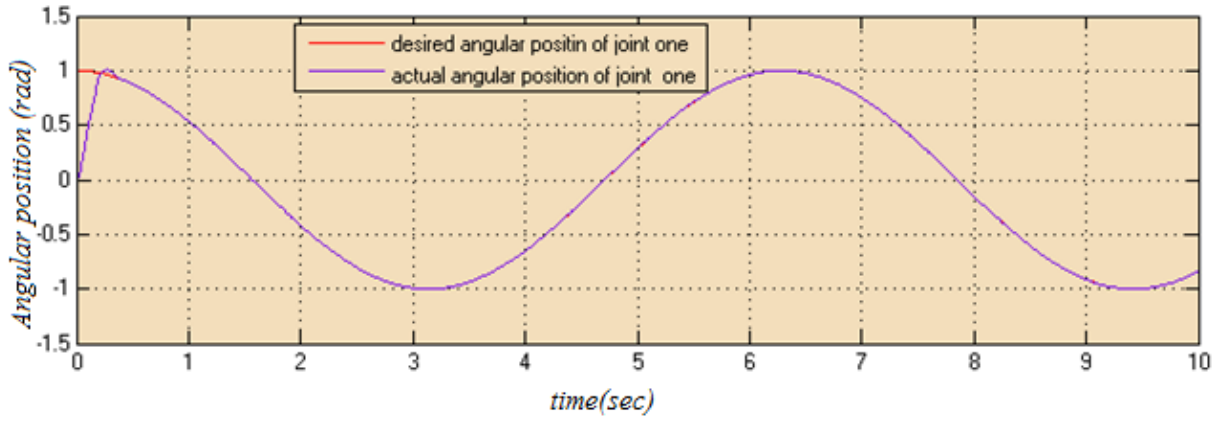

Figure 5.9: Angular position of joint one using SMC without disturbance and parameter variation

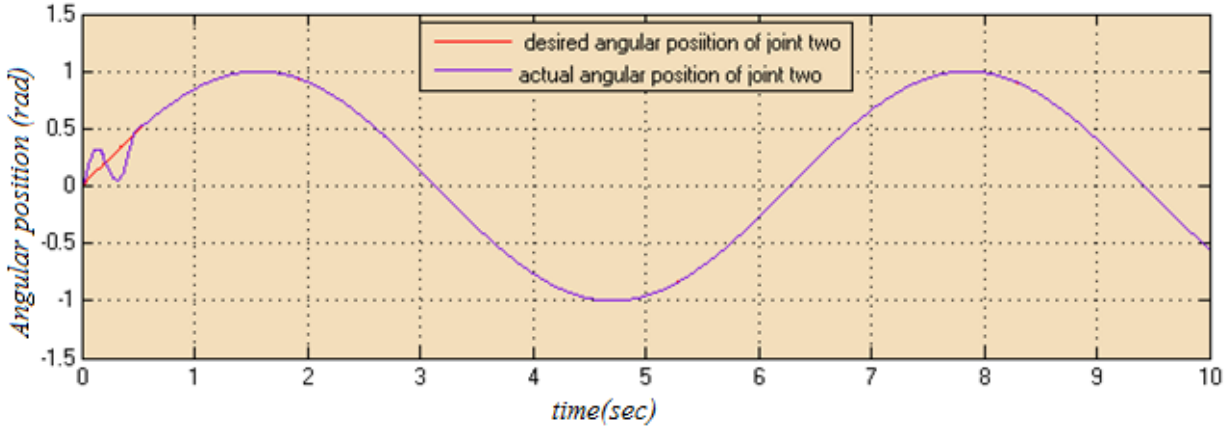

Figure 5.10: Angular position of joint two using SMC without disturbance and parameter variation

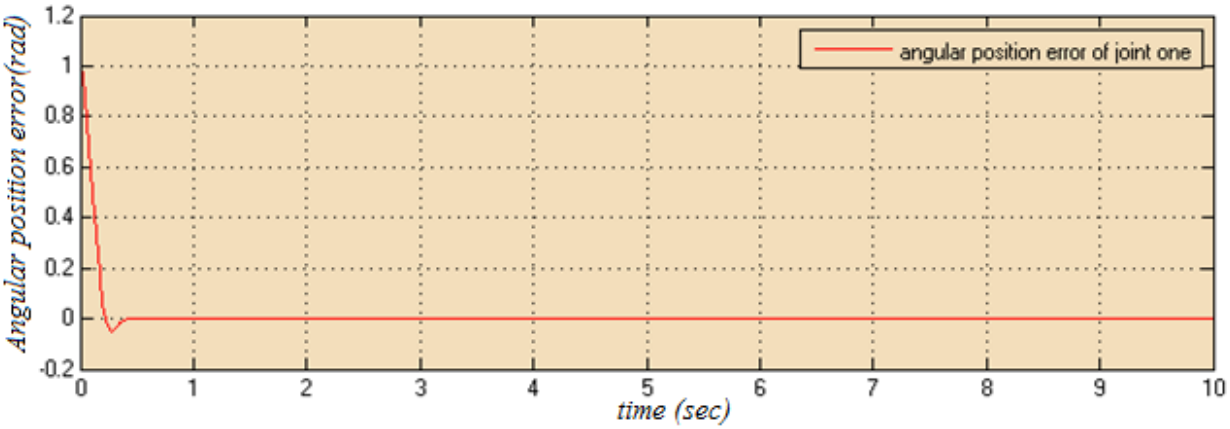

Figure 5.11: Angular position error of joint one using SMC without disturbance and parameter variation

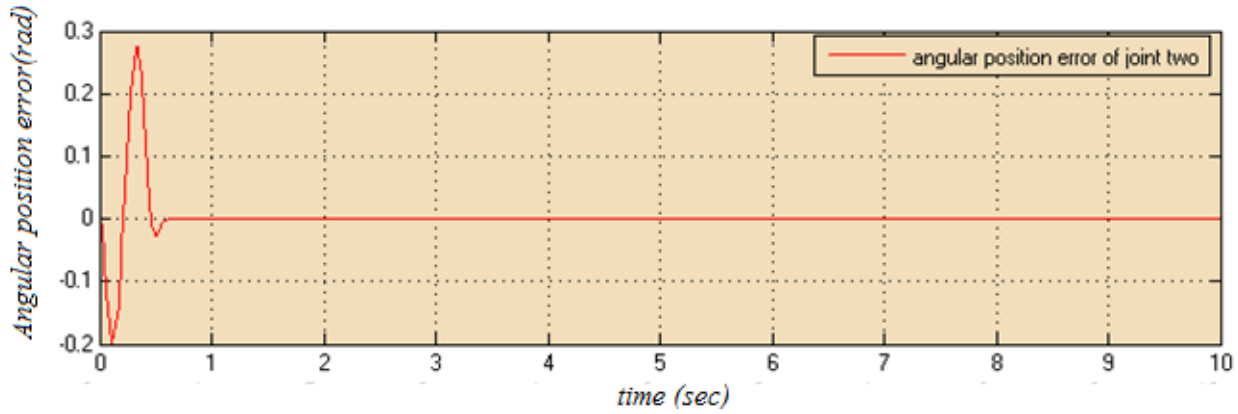

Figure 5.12: Angular position tracking error of joint two using SMC without disturbance and parameter variation

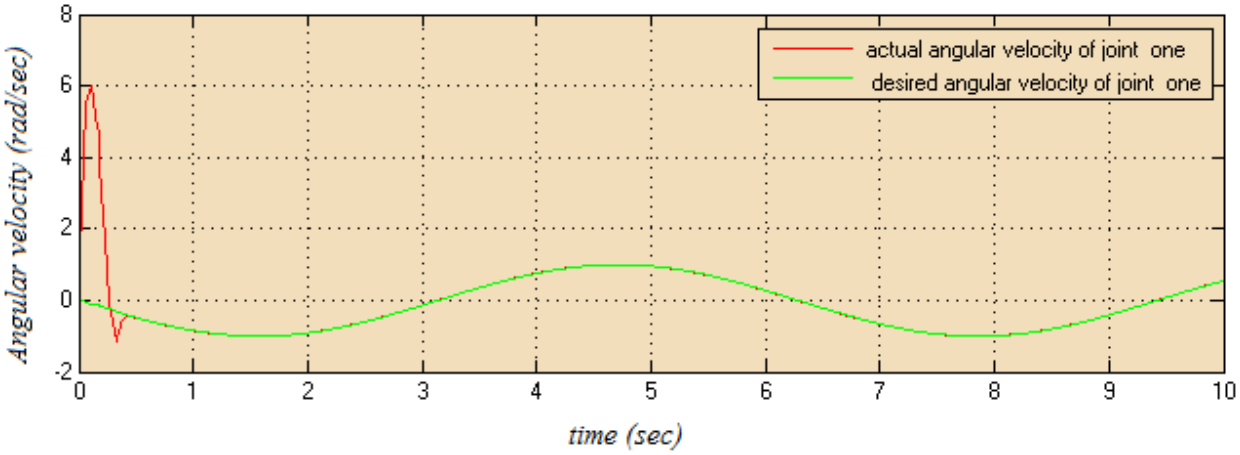

Figure 5.13: Angular velocity tracking of joint one using SMC without disturbance and parameter variation

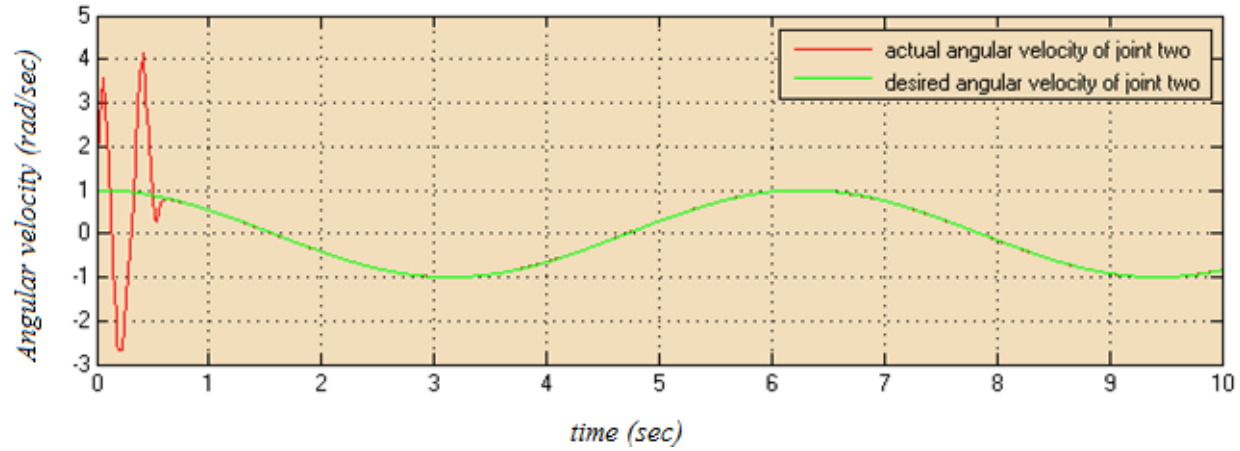

Figure 5.14: Angular velocity tracking of joint two using SMC without disturbance and parameter variation

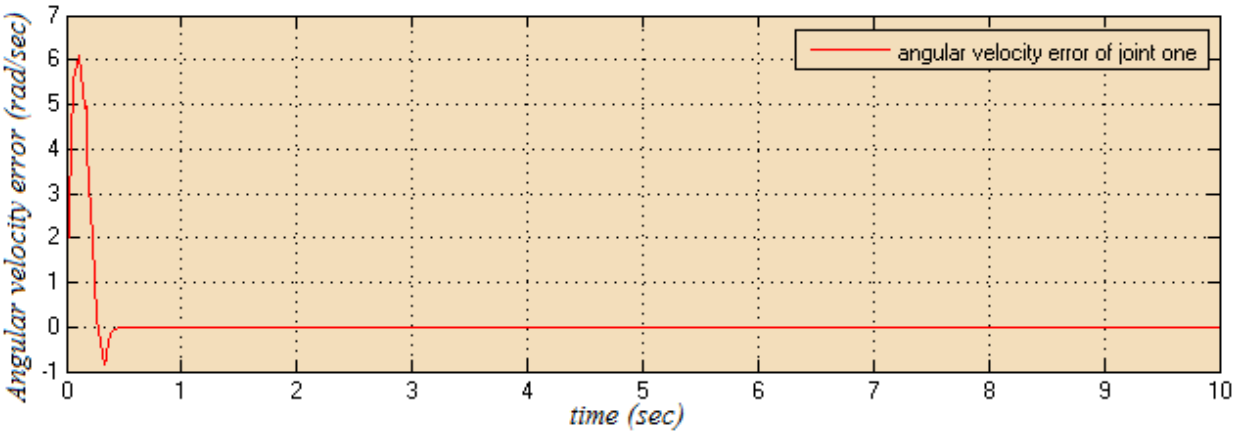

Figure 5.15: Angular velocity tracking error of joint one using SMC without disturbance and parameter variation

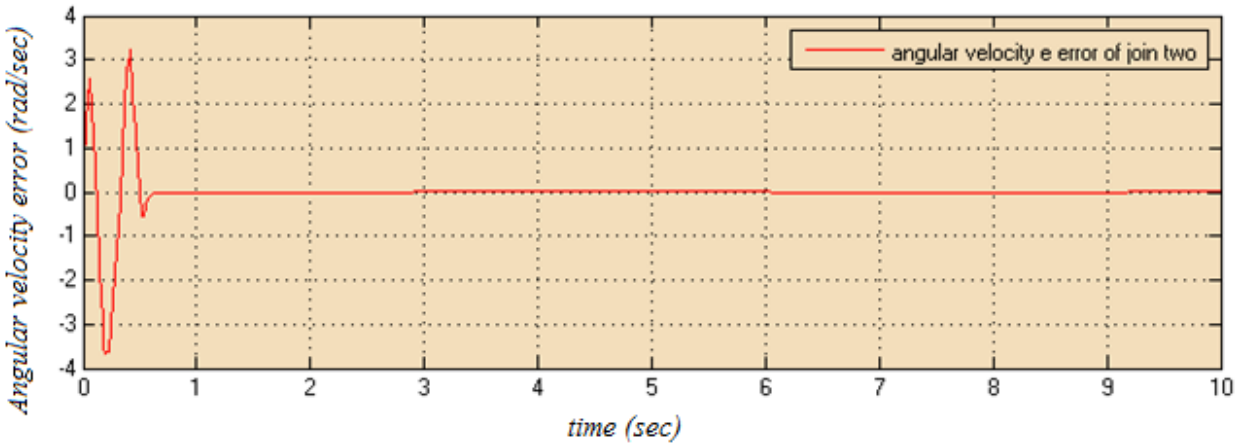

Figure 5.16: Angular velocity tracking error of joint two using SMC without disturbance and parameter variation

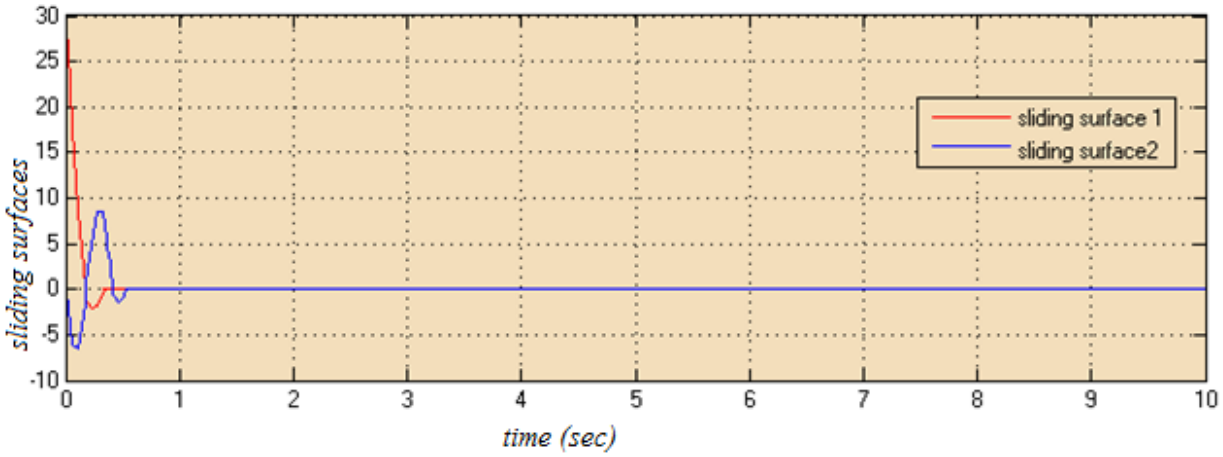

Figure 5.17: Sliding surfaces for joint one and theta two without disturbance and parameter variation

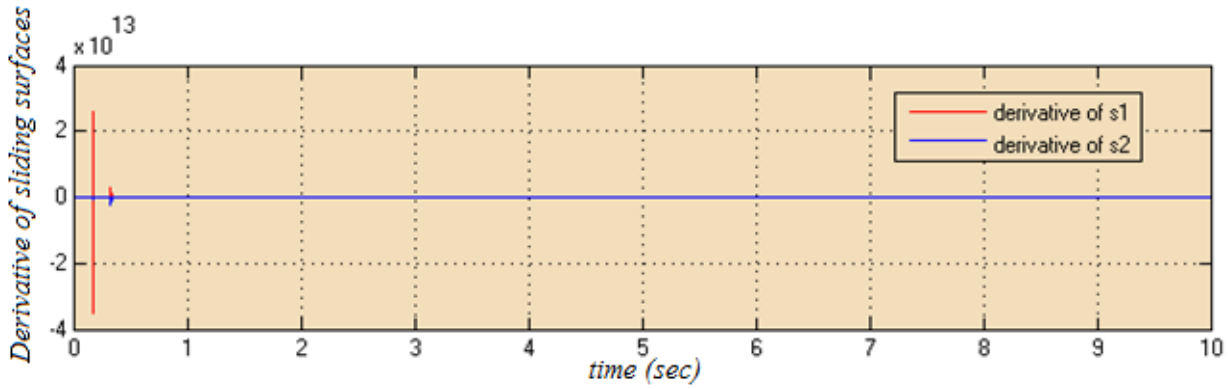

Figure 5.18: Derivative of sliding surfaces for joint one and two without disturbance and parameter variation

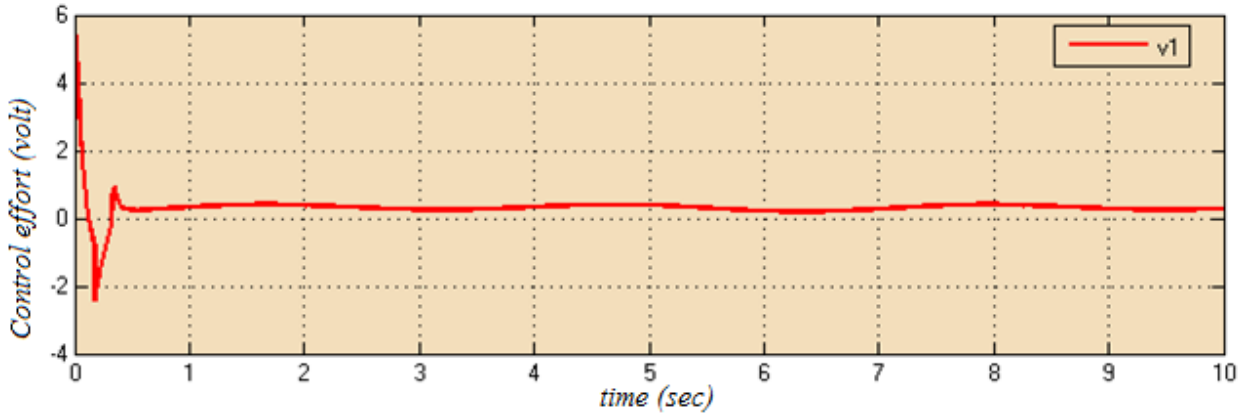

Figure 5.19: Control effort of joint one using SMC without disturbance and parameter variation

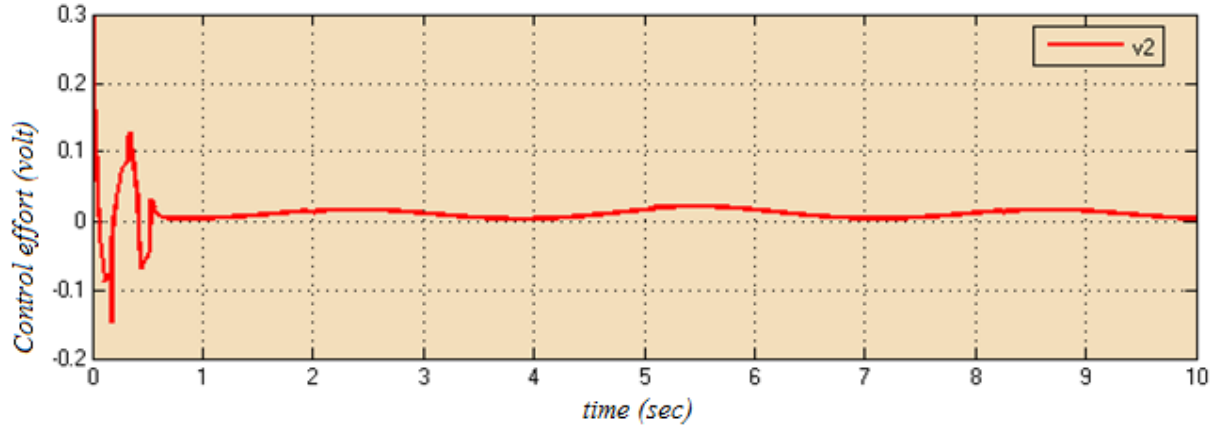

Figure 5.20: Control effort of joint two using SMC without disturbance and parameter variation

5.3 Performance Comparison of SMC and PID Control with disturbance and parameter variation

The Comparison of SMC and PID angular position tracking is discussed as follows. Figure 5.21 and 5.22 shows the simulation result of PID controller for joint one and two. Figure 5.23 and 5.24 are shows the plot angular position tracking error of joint one and two respectively. Depending on the result shown Steady State error of SMC is 0.029% for joint one and 0.0098% for joint two and increased to 15% and 24% respectively overshoot remain the same, rise time is decreased by 0.16sec, and settling time is decreased by 0.4sec for both joints when PID controller is used. Figure 5.25 and 5.26 below shows the plot of angular velocity tracking of joint one and two respectively. The performance of angular velocity of each joint is not as desired one which is poor speed performance when compared to SMC as shown from figure 5.27 and 5.28 of the angular velocity tracking error for joint one and two respectively. The angular velocity of PID is increased to 1 rad/sec for joint one and increased to 0.4rad/sec for joint two. Figure 5-29 and 5-30 shows control effort of joint one and joint two of PID respectively and it needs more high control effort in order to track the desired trajectory by PID relative to the SMC control effort. the maximum voltage use in order to track the desired trajectory by PID it needs 515v for joint one and 28v for joint two. Which is increased by 515.6v and 37.08v respectively. This is more than the maximum voltage supplied by the robot actuator. Based on simulation result of figure 5.21 to 5.30, both trajectory tracking controllers performance response is used for comparisons and SMC has good performance response than PID controller.

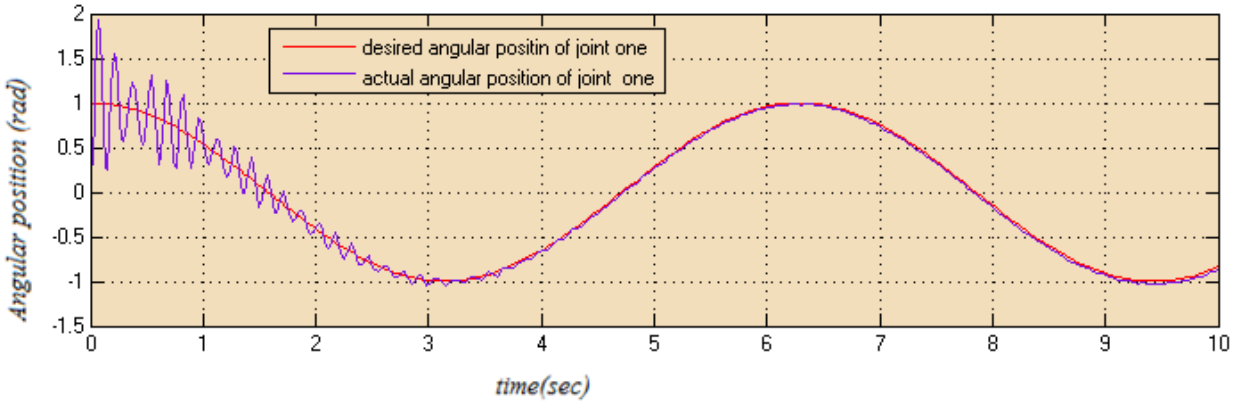

Figure 5.21: Angular position tracking of joint one using PID without disturbance and parameter variation

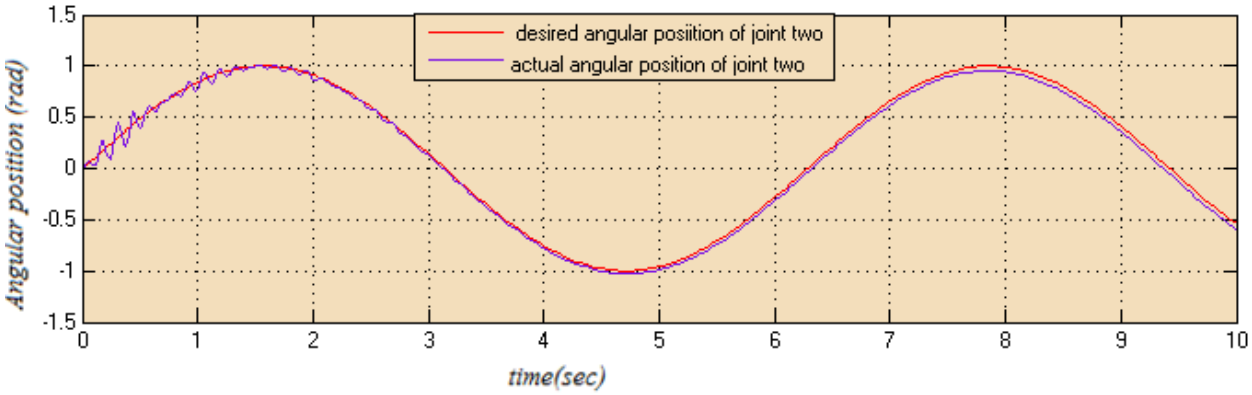

Figure 5.22: Angular position tracking of joint two using PID without disturbance and parameter variation

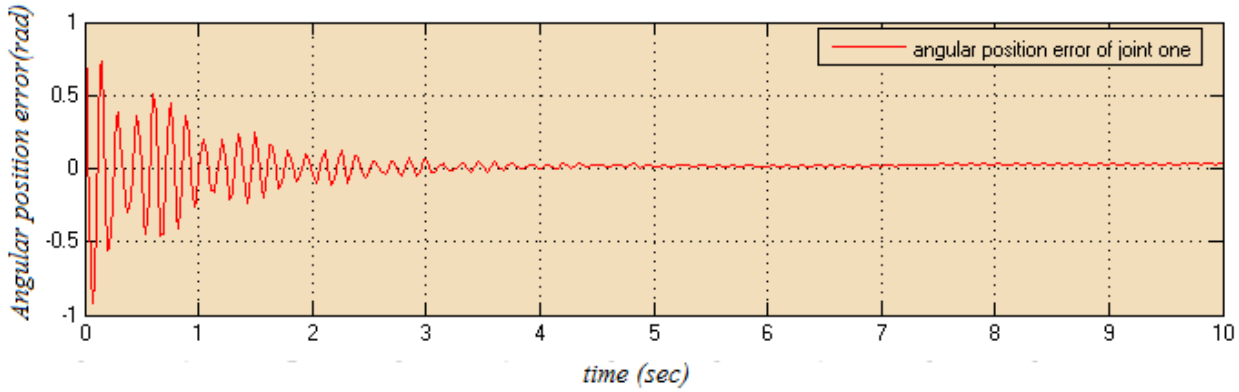

Figure 5.23: Angular position error of joint one using PID without disturbance and parameter variation

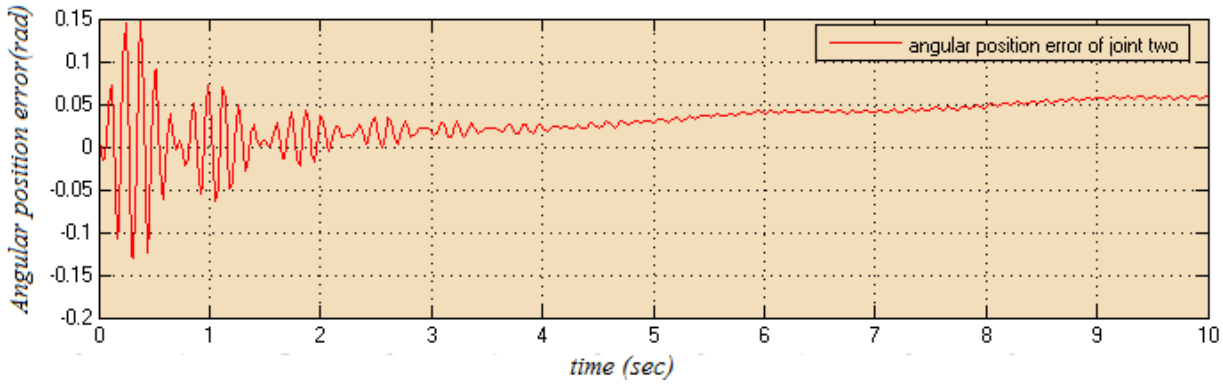

Figure 5.24: Angular position error of joint two using PID without disturbance and parameter variation

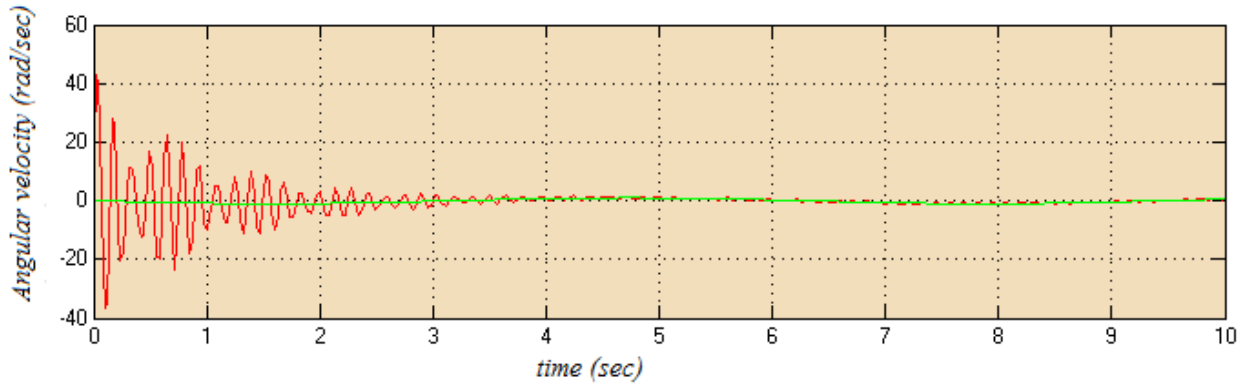

Figure 5.25: Angular velocity tracking of joint one using PID controller without disturbance and parameter variation

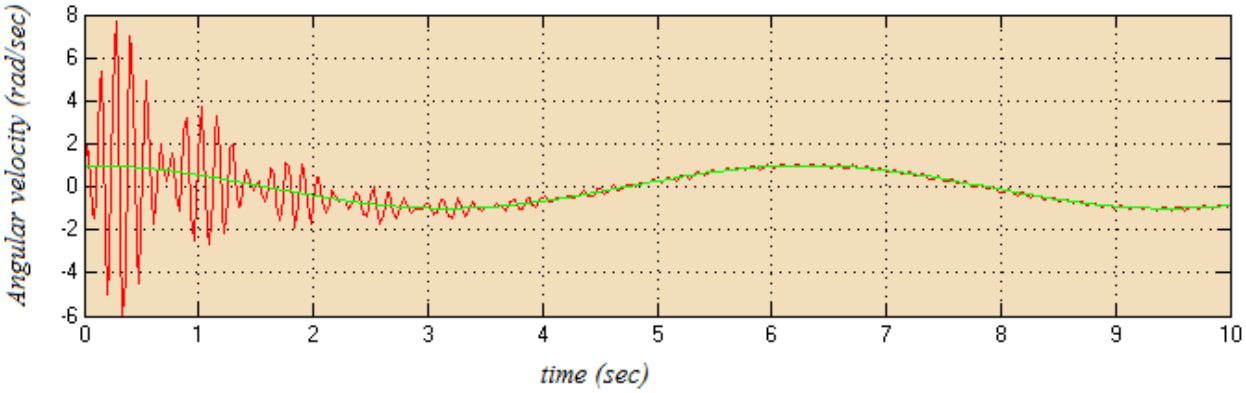

Figure 5.26: Angular velocity tracking of joint one using PID without disturbance and parameter variation

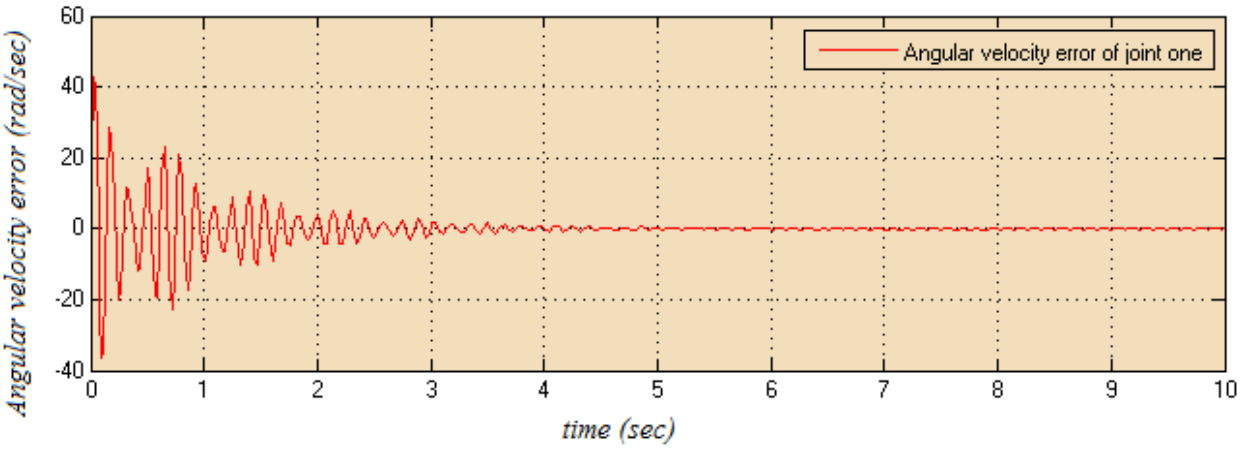

Figure 5.27: Angular velocity tracking error of joint one using PID without disturbance and parameter variation

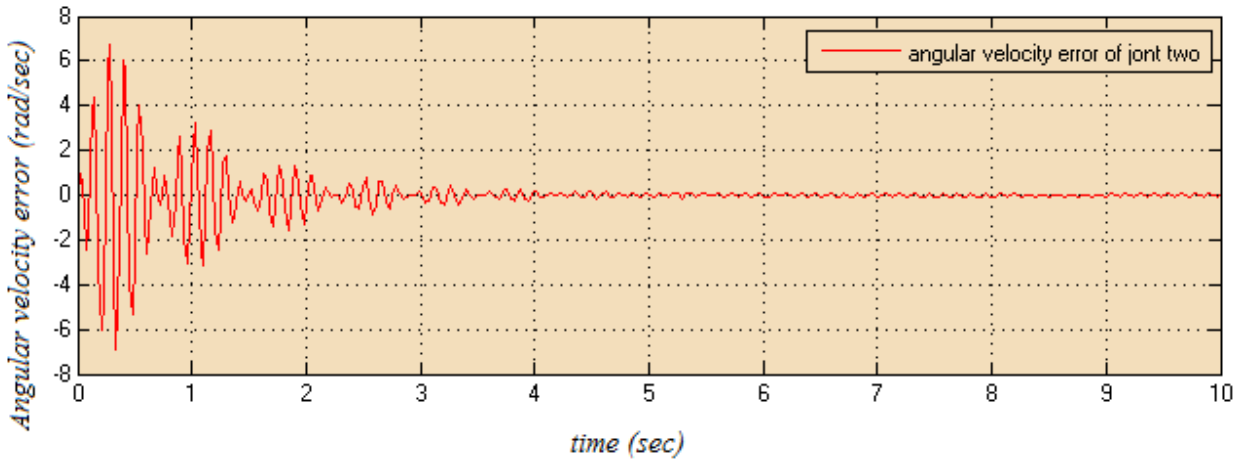

Figure 5.28: Angular velocity tracking error of joint two using PID without disturbance and parameter variation

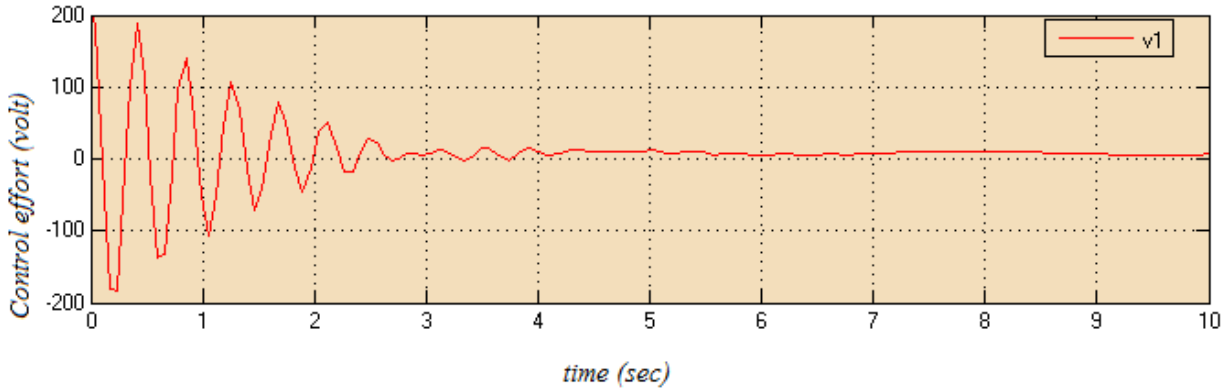

Figure 5.29: Control effort of joint one using PID without disturbance and parameter variation

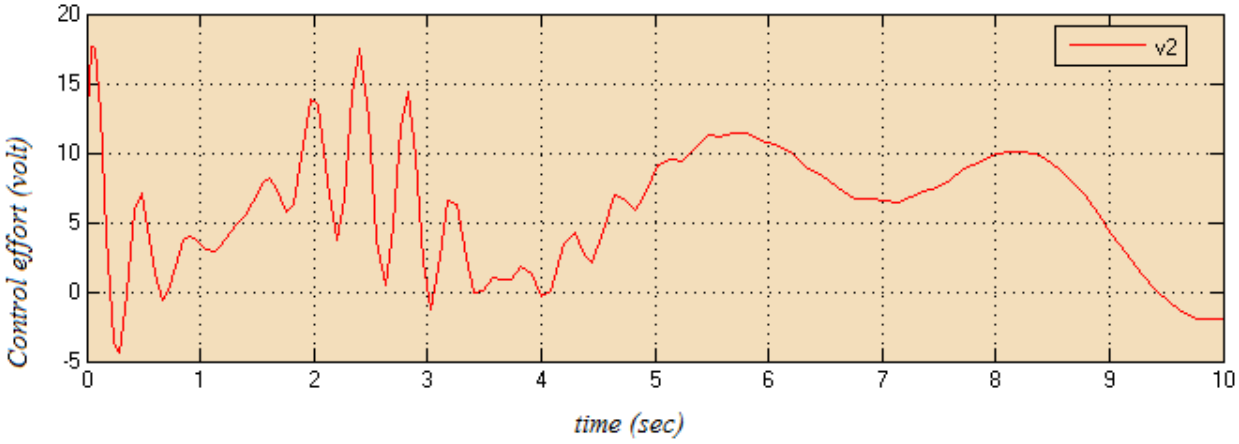

Figure 5.30: Control effort of joint two using SMC without disturbance and parameter variation

5.4 Disturbance rejection and sensitivity to parameter variation of SMC and PID Control

The trajectory tracking of 2 DOF for SMC and PID controller with disturbance and parameter variation is tested in this part. Here the predefine 25% of amplitude of desired input signal disturbance and the parameter Variation (frictional force) is considered and the performance of the SMC and PID are compared. According to Figure 5.31 and 5.38 of the SMC results shows the performances with disturbance and without disturbance are almost the same. However from Figure 5.39 to 5.46 of PID controller results with disturbance and parameter variation shows that the Steady State error is increased to 2% and 9% for joint one and two respectively. Figure 5.47 to 5.50 shows the control effort performance in SMC and PID controller. In the control effort, smaller amplitude means less energy. Figure 5.49 and 5.50 of SMC control effort is incerased by 0.03v for joint one andincreased by 0.02v for joint two. the amplitude of the control input of PID controller with disturbance and parameter variation is much larger when compared to the control effort without disturbance and it is increased by 18v and 6.5v. Generally from the result that the SMC is more robust and stable than PID controller.

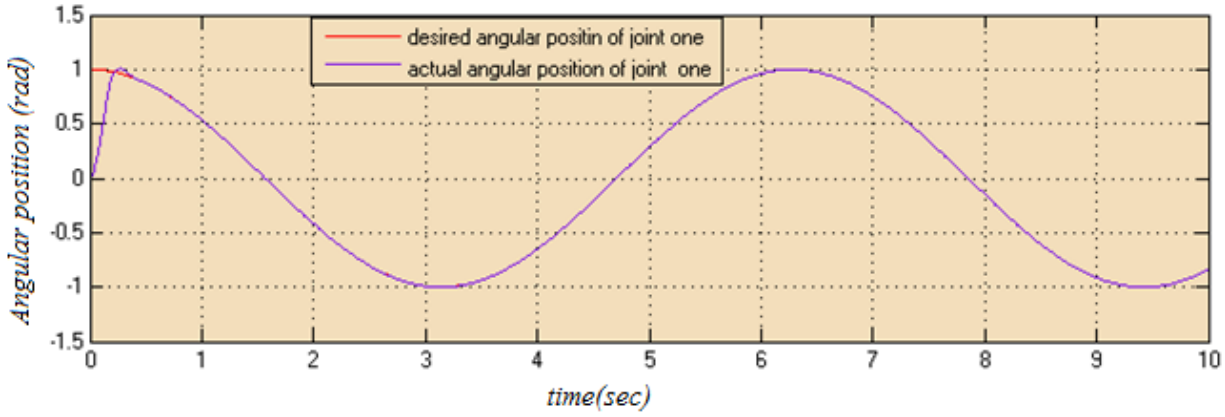

Figure 5.31: Angular position tracking of joint one using SMC with disturbance and parameter variation

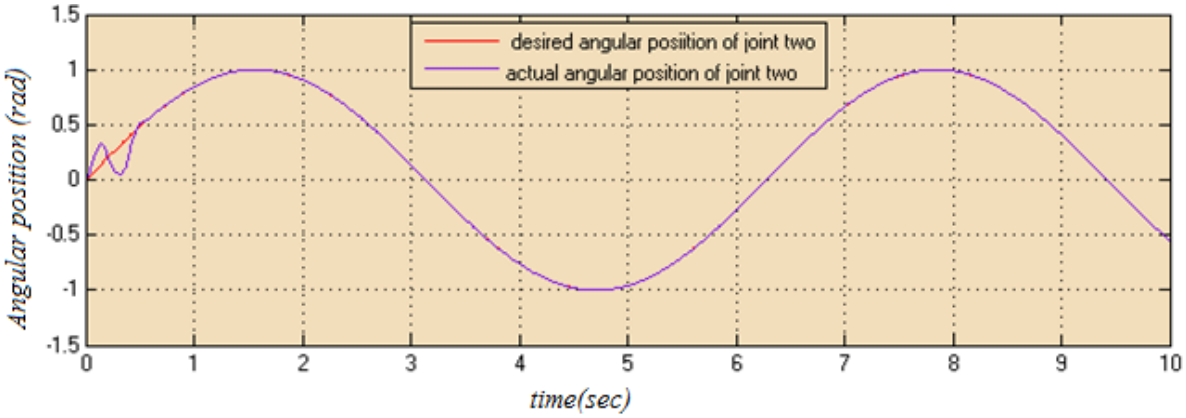

Figure 5.32: Angular position tracking of joint two using SMC with disturbance and parameter variation

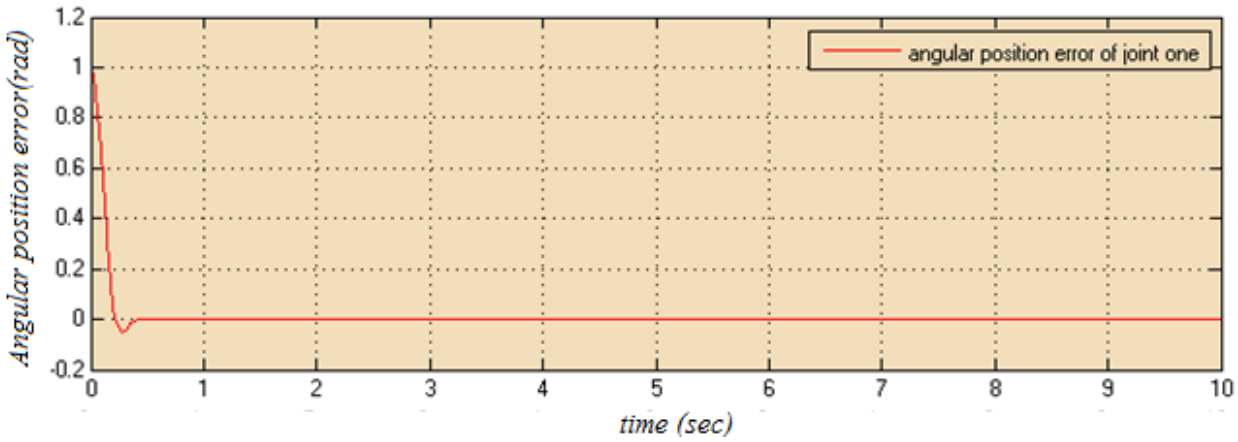

Figure 5.33: Angular position error of joint one using SMC with disturbance and parameter variation

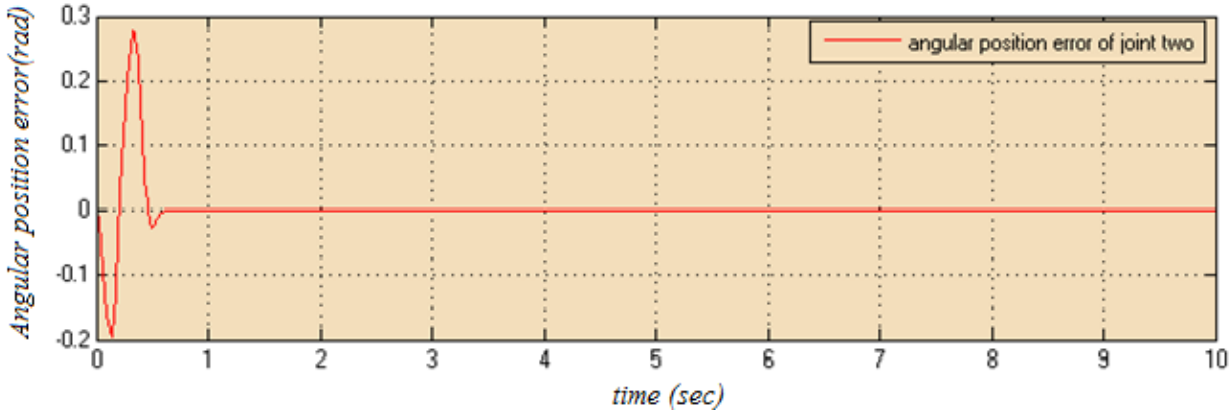

Figure 5.34: Angular position error of joint two using SMC with disturbance and parameter variation

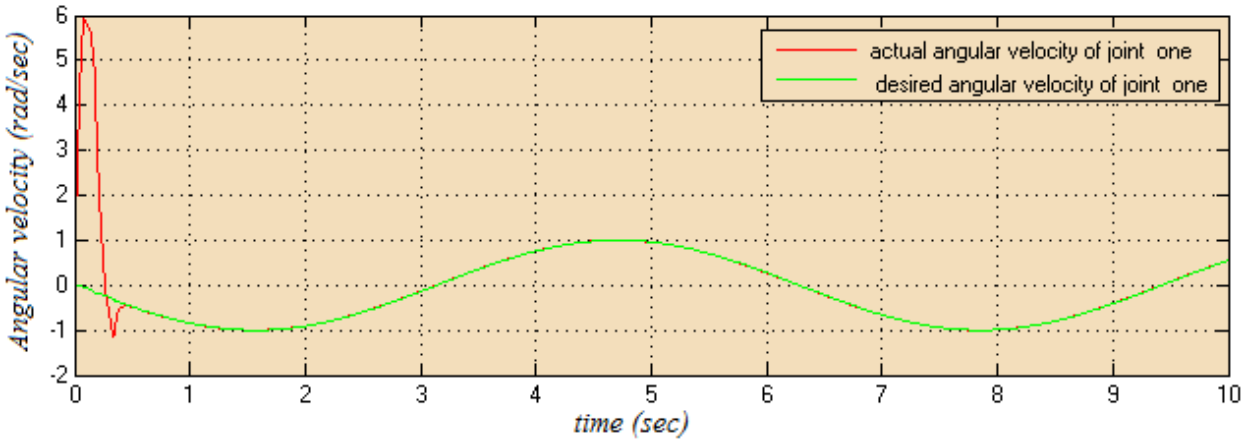

Figure 5.35: Angular velocity tracking of joint one using SMC with disturbance and parameter variation

5

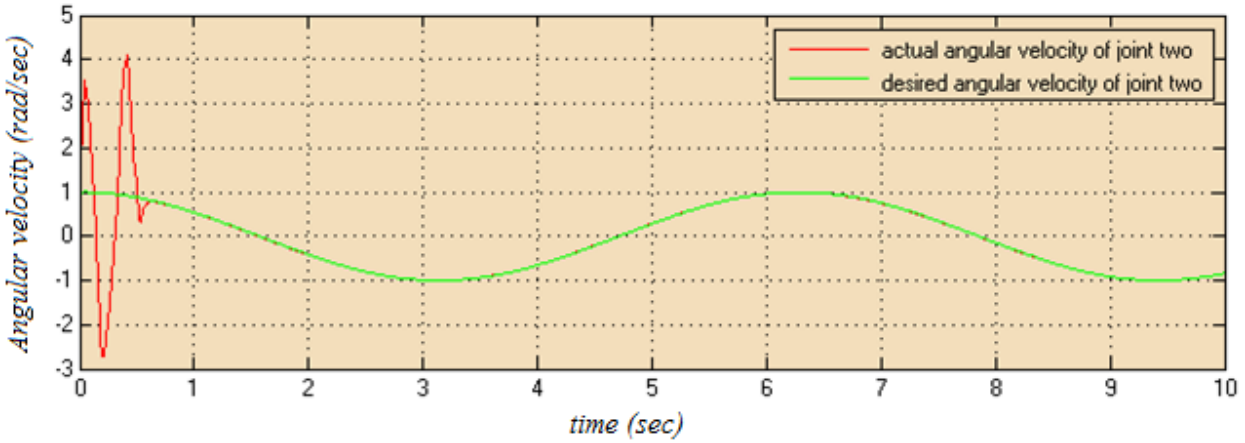

Figure 5.36: Angular velocity tracking of joint two using SMC with disturbance and parameter variation

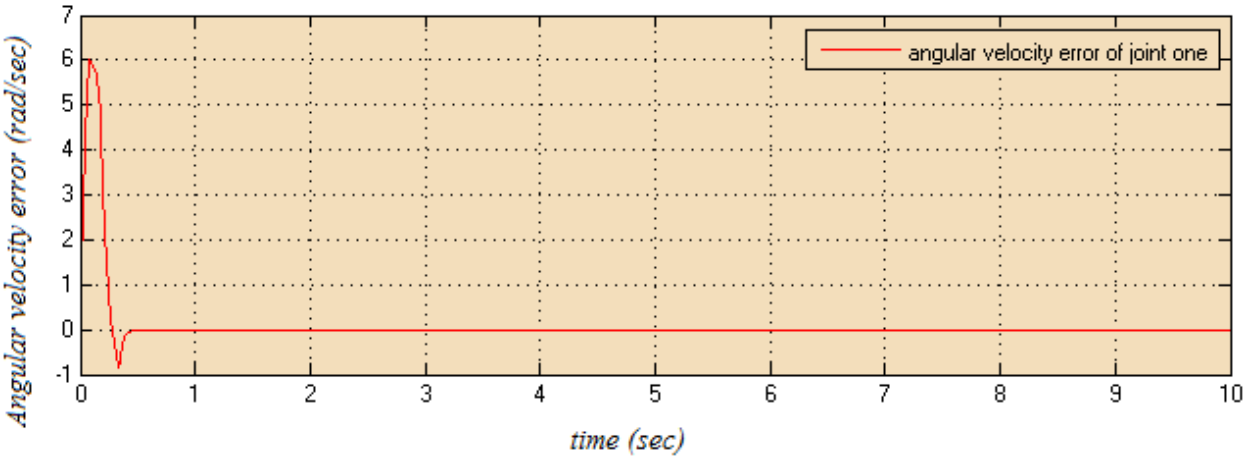

Figure 5.37: Angular velocity tracking error of joint one using SMC with disturbance and parameter variation

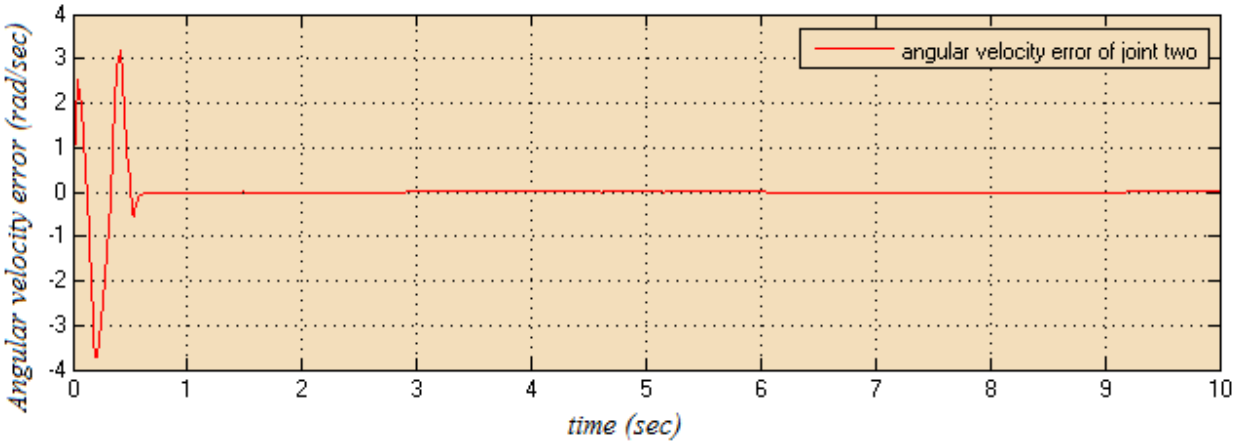

Figure 5.38: Angular velocity tracking error of joint two using SMC with disturbance and parameter variation

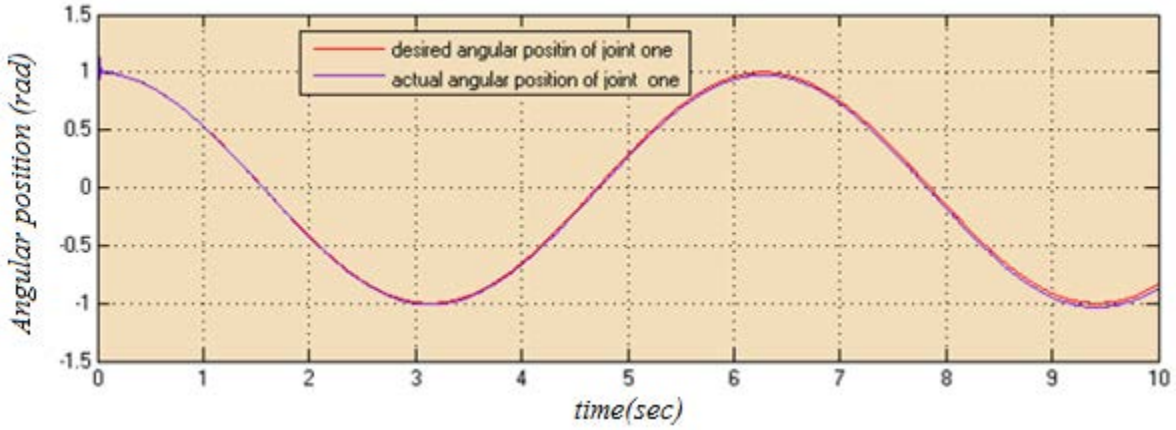

Figure 5.39: Angular position tracking of joint one using PID with disturbance and parameter variation

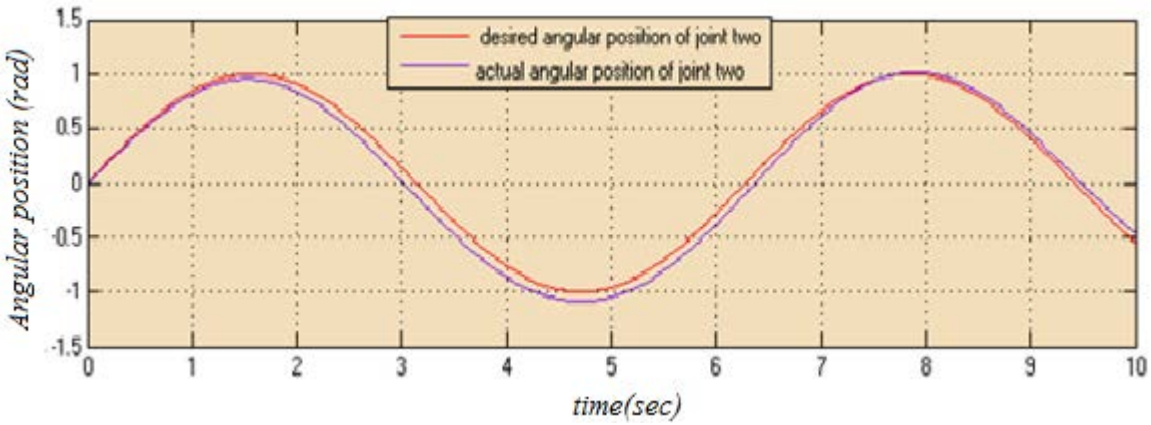

Figure 5.40: Angular position tracking of joint one using PID with disturbance and parameter variation

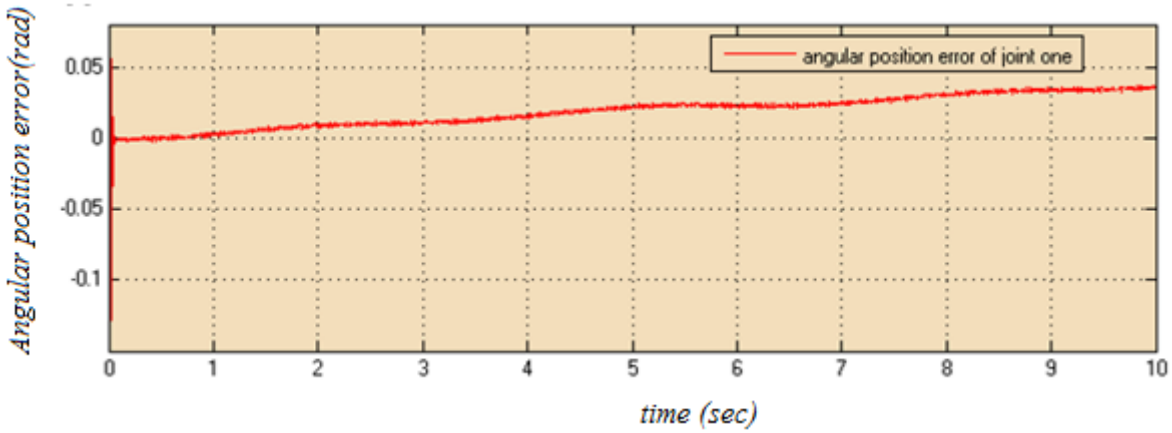

Figure 5.41: Angular position error of joint one using PID with disturbance and parameter variation

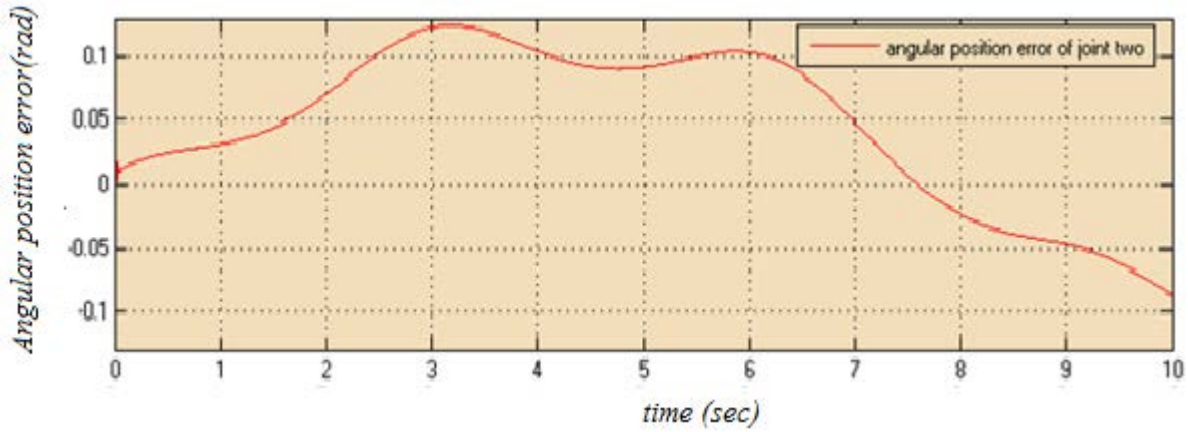

Figure 5.42: Angular position error of joint two using PID with disturbance and parameter variation

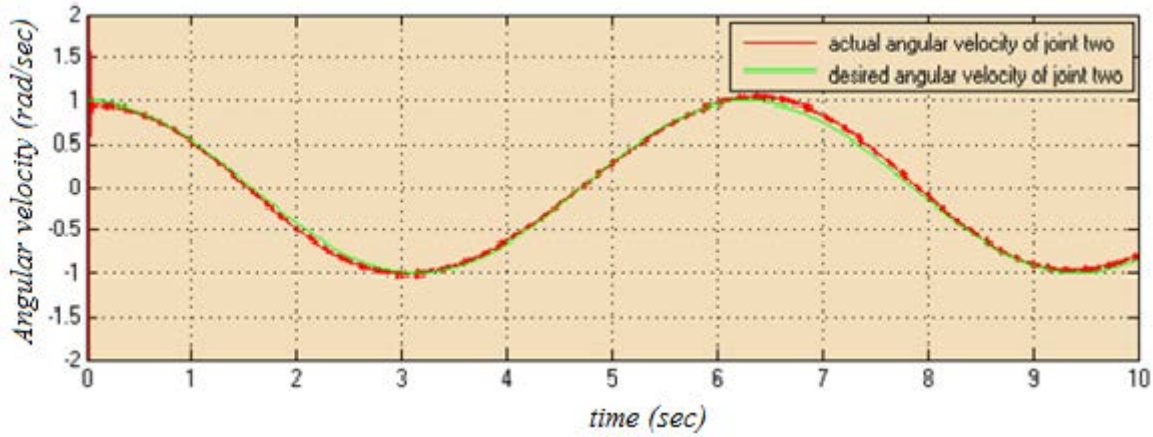

Figure 5.43: Angular velocity tracking of joint one using PID with disturbance and parameter variation

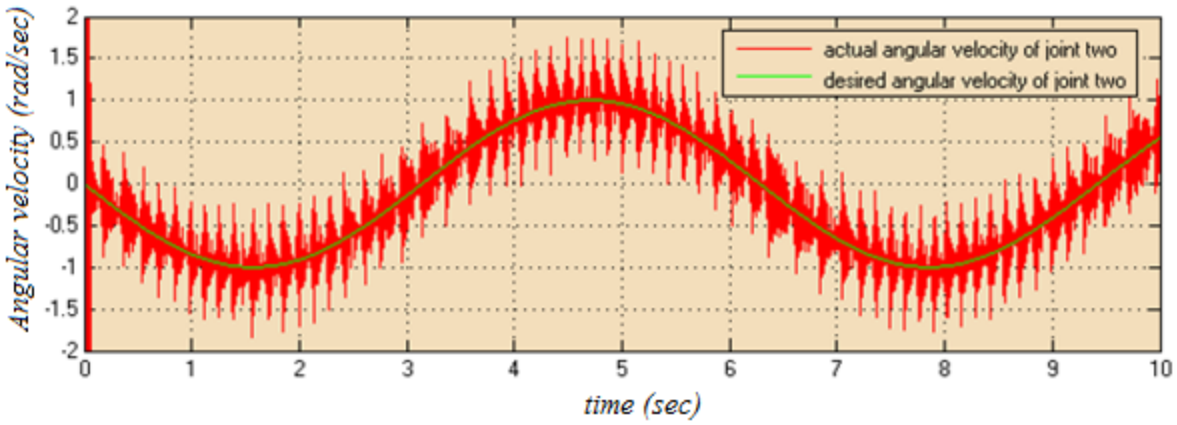

Figure 5.44: Angular velocity tracking error of joint two using PID without disturbance and parameter variation

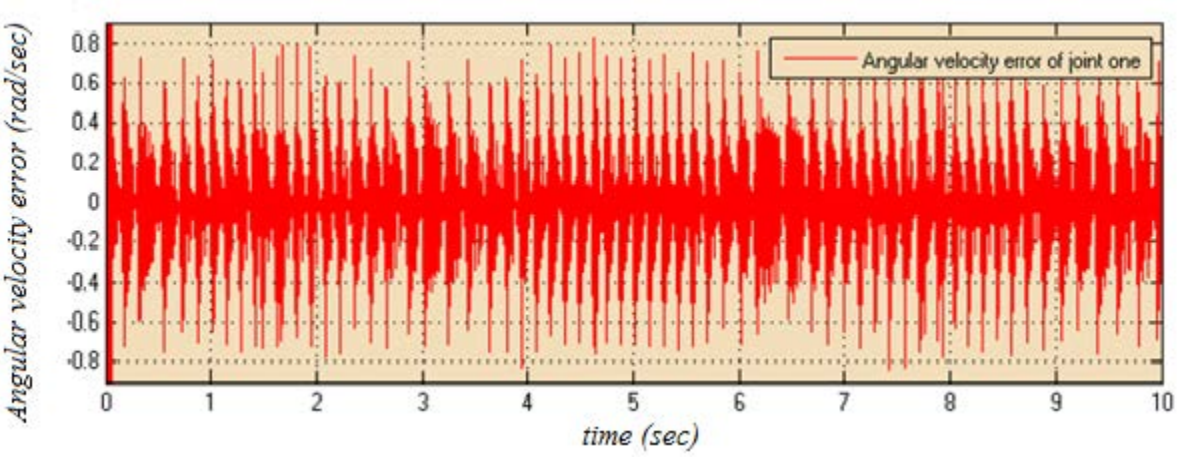

Figure 5.45: Angular velocity tracking error of joint one using PID with disturbance and parameter variation

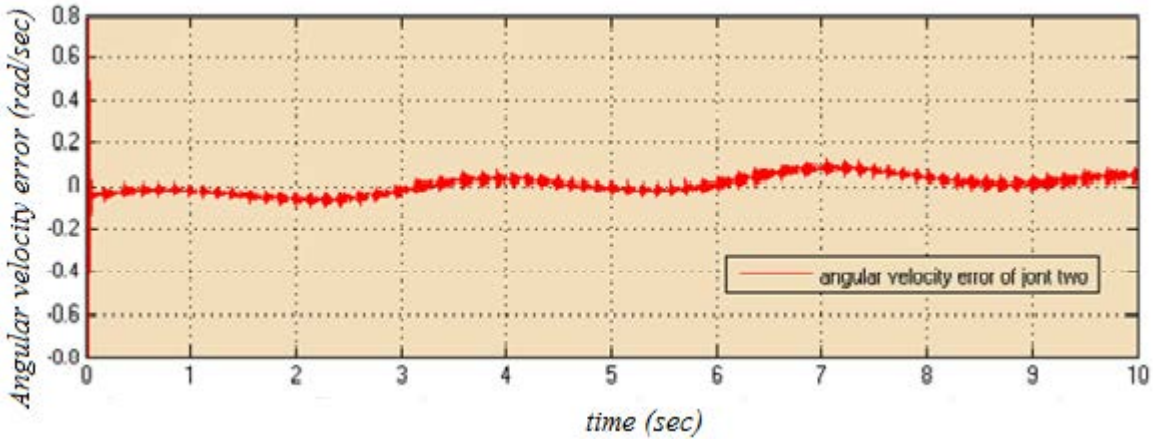

Figure 5.46: Angular velocity tracking error of joint two using PID with disturbance and parameter variation

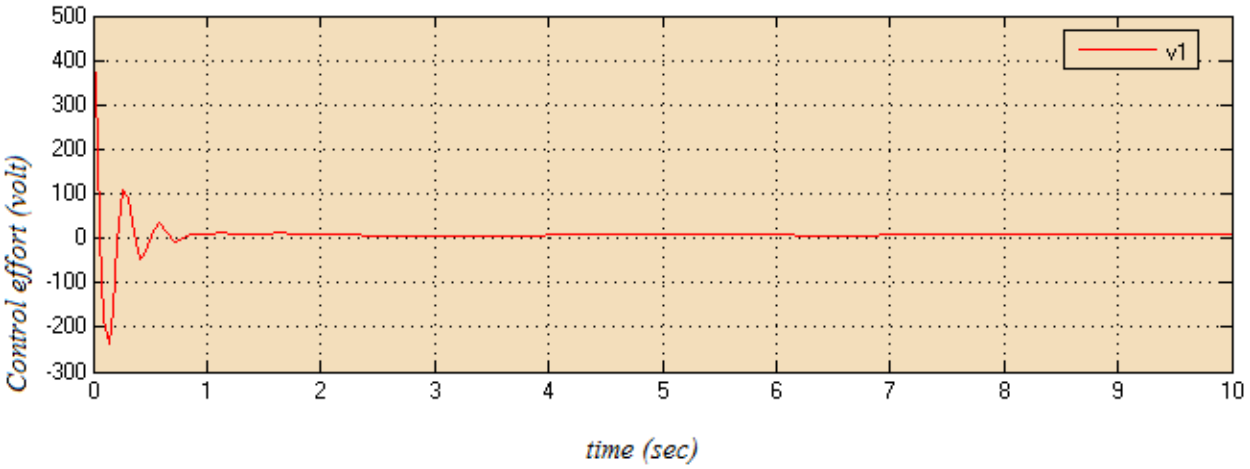

Figure 5.47: Control effort of joint one using PID with disturbance and parameter variation

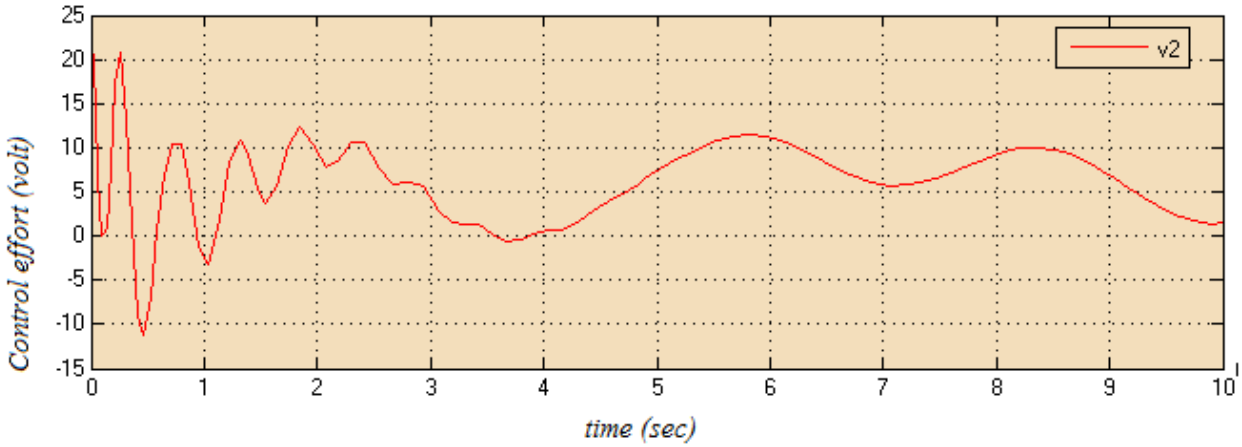

Figure 5.48: Control effort of joint two using PID with disturbance and parameter variation

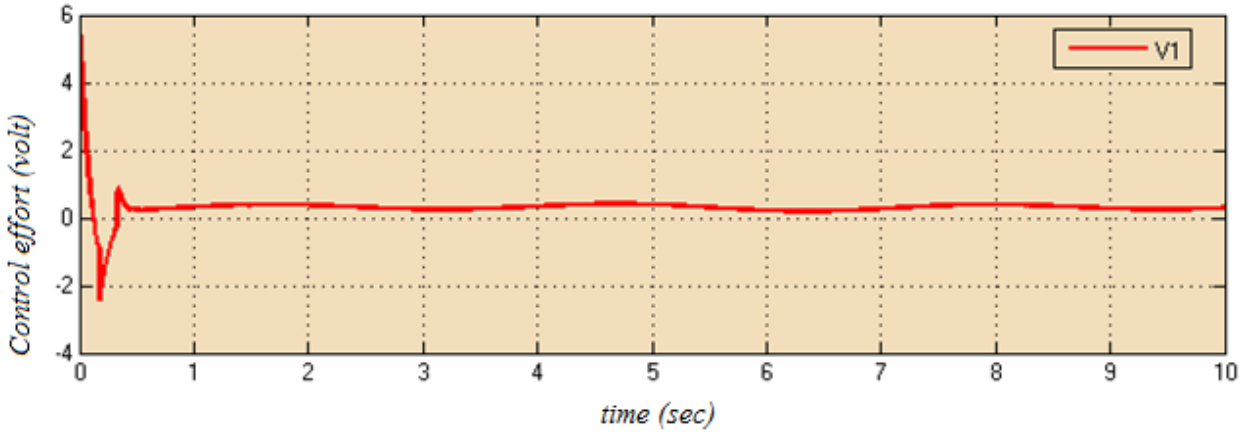

Figure 5.49: Control effort of joint one using SMC with disturbance and parameter variation

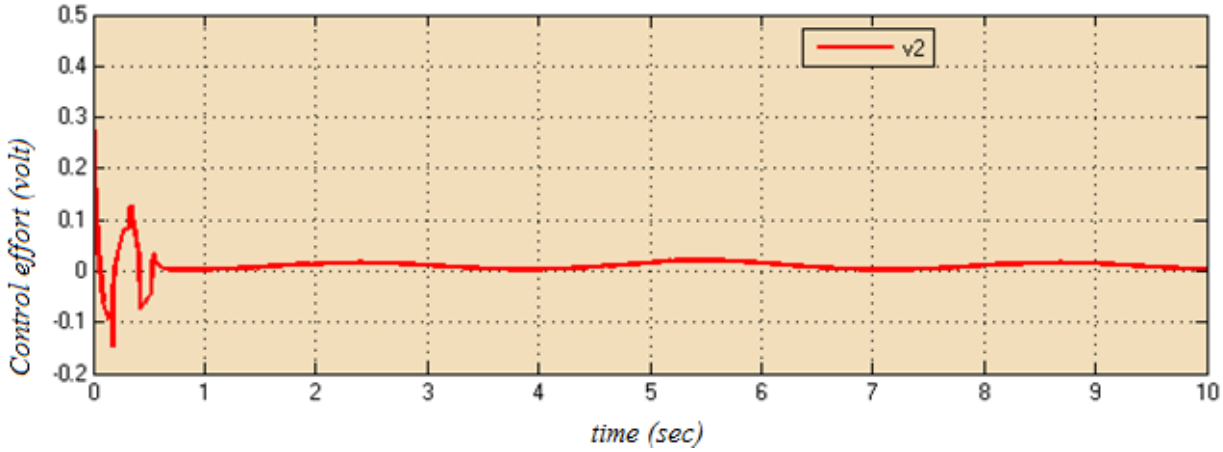

Figure 5.50: Control effort of joint two using SMC with disturbance and parameter variation

The designed SMC result shows good performance response when the system is under the disturbance of 25% of input amplitude. However when the system disturbance is 50% of the input amplitude the sliding surface shown in Figure 5.51 and the derivative of sliding surface shown in Figure 5.52 of joint two introduce chattering and this causes the chattering in the control effort of joint two as shown in figure 5.53 below

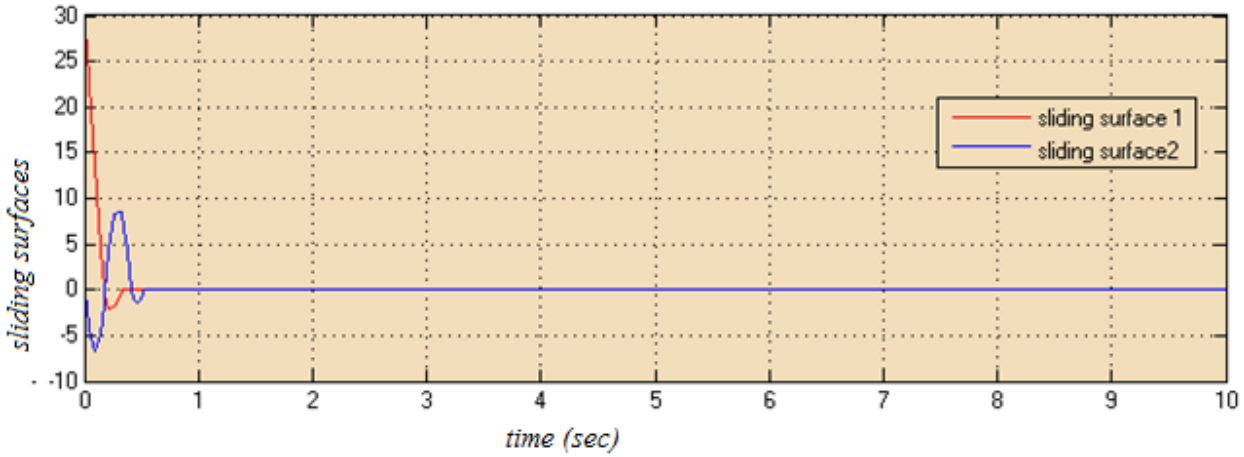

Figure 5.51: Sliding surfaces for joint one and theta two with disturbance and parameter variation

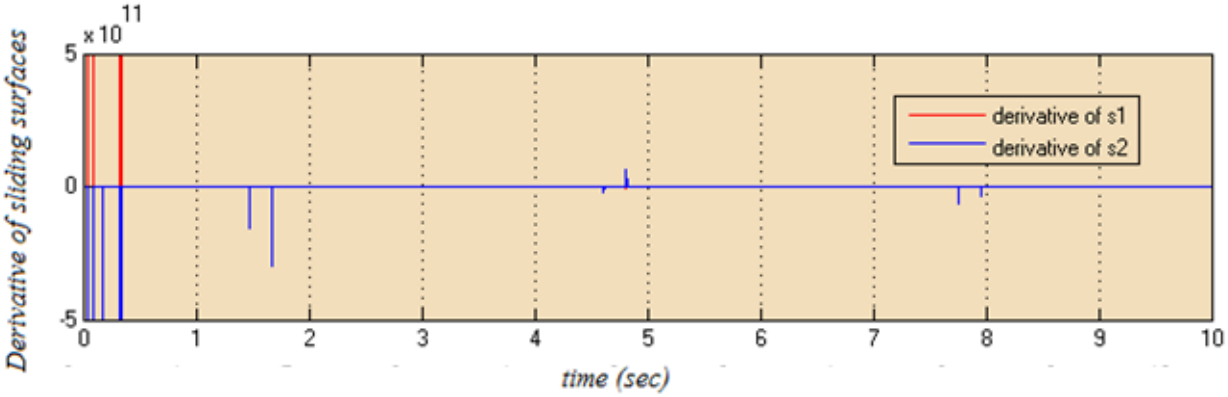

Figure 5.52: Derivative of sliding surfaces for joint one and two without disturbance and parameter variation

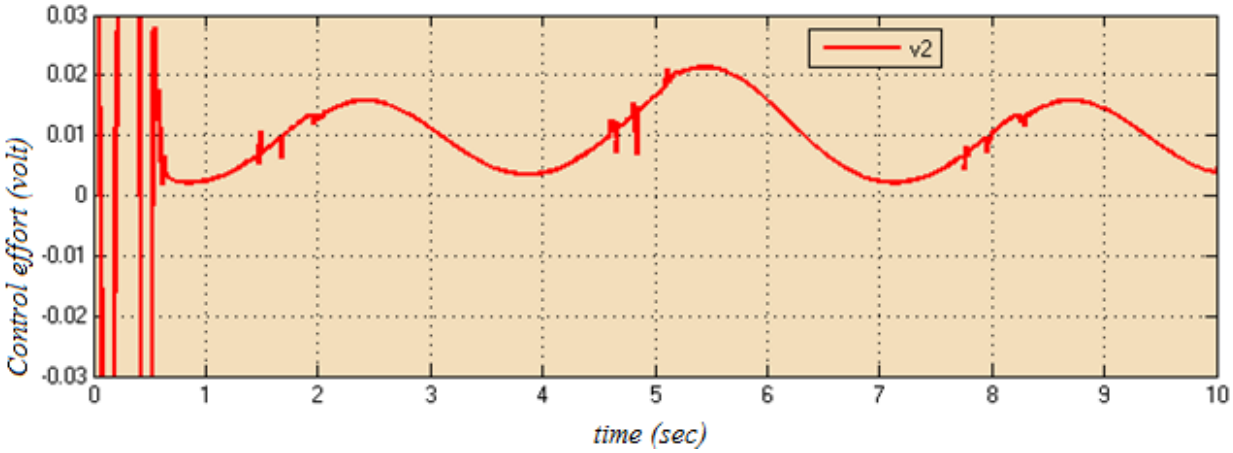

Figure 5.53: Control effort of joint two using SMC with disturbance and parameter variation

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, a SMC is proposed for a two degree of freedom robot arm manipulator system and simulation results are presented. Sliding mode control of two DOF robot arm manipulator system is developed. Secondly, the SMC introduce the chattering phenomena which is one of the draw-back of it. The chattering phenomenon is overcome by the use of a saturation function in place of a pure signum function. In this thesis SMC is designed successfully and PID controller designed for comparative study. Based on result and analysis conclusion has been made that the modern controller (SMC) capable of controlling the robotic manipulator than the conventional controller (PID). The simulation result shows that SMC controller has better performance, more robust when compared to PID controller for sinusoidal trajectory tracking.

6.2 Future works

During the study of this thesis we consider the disturbance to 25% of input amplitude and the designed controller successfully accomplished the mission of the study. We also try to increase the disturbance to 40% of the input amplitude and designed controller faces the problem of chattering with control effort of second joint. So in the future one can design other chattering elimination method which may overcame any amount of disturbance introduced in the robot.

REFERENCES

[1] S. H. a. M. V. Mark W. Spong, introduction robot dynamics and control, second edition ed., January 28, 2004, pp. pp.9-12.

[2] J. a. W. L. Slotine, Applied nonlinear control, Prentice-Hall Inc, 1991.

[3] K. Young, "Controller design for amanipulator using theory of variable structures," *IEEE Trans. On systems, Man and Cybernetics,* no. SMC-8, pp. pp. 210-218, 1978.

[4] J. Slotine, "Sliding controller design for nonlinear systems," *Int. J. Control,* vol. 40, pp. pp.421-434, 1984.

[5] I. Utkin, "Sliding mode in controloptimization," no. Springer-Verlag, 1992.

[6] H. a. B. S. Iordanov, "Experimental evaluation of the robustness ofdiscrete sliding," *IEEE Trans. On control System Technology,* vol. 5, pp. 254-260, 1997.

[7] J. a. S. S. Slotine, "Tracking control of nonlinear systems using sliding surfaces, with application to robot manipulators," *Int. J. Control,,* pp. 465-492, 1998.

[8] M. I. T. a. A. G. Emami, "Development of a systematic methodology of," *IEEE Trans. On Fuzzy Systems,,* vol. 3, pp. 346-61, 1998.

[9] F. H. H. a. K. M. Harashima, "Practical robust control of robot arm using variable," *IEEE Conference,* pp. 532-539, 1986.

[10] E. A. S. K. Taha, "Robust control of a constrained robot arm. InternationalConference," *IEEE,* pp. 90-91, 1993.

[11] M. Hamerlain, "Robust control with reduced knowledge of unmodeled dynamics using sliding mode," *EEE conference,* pp. 260-265, 1995.

[12] UtkinV.I, "Variable structuresystems with sliding modes," *IEEE Transaction onAutomatic Control,* Vols. Vol.AC-22, pp. 212-220, 1977.

[13] E. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," *Pt-oc. IEEE,* vol. No.2, pp. 212-220, 1974.

[14] D. R. V. a. P. N. T. Jyoti Ohri, "Comparison of Robustness of PID Control and SlidingMode Control of Robotic Manipulator," *Intelligent Systems & Communication (ISDMISC) Proceedings published by International Journal of Computer Applications® (IJCA),* pp. 5-9, 2011.

[15] S. M. R. B. a. S. R. . Farzin Piltan, "Design New Control Methodology of Industrial Robot Manipulator," *nternational Journal of Hybrid Information Technology,* vol. 5, pp. 41-49, October, 2012.

[16] S. a. B. Jolly Shah, "Dynamic analysis of two link robot manipulator for control design using computed torque control," *international journal of research in computer applications and robotics,* vol. 3, no. 1, pp. 52-59, January 2015.

[17] M. M. Fateh, "On the Voltage-Based Control of Robot Manipulators," *International Journal of Control, Automation, and Sy stems,* Vols. 6,no,5, pp. 702-712, October 2008.

[18] B. B. R. I. B. G. V. S. K. Rao, "modelling and control of 2-dof obotic manipulator using BLDC motor," *International Journal of Science, Engineering and Technology Research (IJSETR),* vol. 3, no. 10, pp. 2760-63, October 2014 .

[19] J. J. Craig, Introduction to Robotics, United States of America: Pearson Education International, 2005.

[20] S. M. G. a. P. R. K. Kawamura, Industrial Robotics Theory, Modelling and control, First published January 2007.

[21] M. M. A. Ashi, "Trajectory Tracking Control of A 2-DOF Robot Arm Using Neural Networks," *Scientific Research& Graduate Studies Affairs,* pp. 11-17, febrary 2014.

[22] R. F. &. K. R., "Experimental Evaluation of Model-based Controllers on a Direct-Drive Robot Arm," *Mechatronics,,* pp. 267-282, 2001.

[23] S. b. niku, "introduction to robotics analysis,control and application," *san luis obispo,* pp. 147-156, 2010.

[24] T. L. W. Y. C. Chern, "Integral Variable Structure Approach for Robot Manipulators," *IEE Proc. (D),,* pp. 160-165, 1992.

[25] M. M. Fateh, "International Journal of Control, Automation, and Systems," Vols. 6, no.5, pp. 702-712, October 2008.

[26] J. J. NAVANEETHAN S1, "Speed control of Permanent Magnet Synchronous Motor using Power Reaching Law based Sliding Mode Controller," *TRANSACTIONS on SYSTEMS and CONTROL, E-ISSN WSEAS ,* vol. Volume 10, no. 270-77, 2015.

[27] *matlab,* 2013.

[28] P. Muneeb Ahmad1, "Analysis of Permanent Magnet Synchronous Motor under Different Operating Condition Using Vector Controlled in MATLAB," *international journal of innovative research in electrical, electronics, instrumentation and control engineering ,* vol. 1, no. 4, july 2013 .

[29] S. E. Shafie, "Sliding Mode Control of Robot Manipulators via Intelligen Approaches," pp. 137-67.

[30] V. Utkin, "Variable Structure systems with Sliding Modes," *IEEE Transaction on Automatic Control,* pp. 212-222, 1977.

[31] F. moldoveanu, "sliding mode controller design for robot manipulators," vol. 7 (56) No. 2, pp. 98-103, 2014.

[32] W. H. J. Gao, " Variable Structure Control of Nonlinear Systems A New A New," *IEEE Trans. on Industrial Electronics,* vol. no.1 , pp. 45-55 , 1993.

APPENDIX A

Level-2 mat-lab s-function programming

Level-2 mat-lab s-function programming for inertia matrix for both controller

```
function masSMC(block)

% calls to the main body of the function.
setup(block);

%endfunction

%   Set up the S-function block's basic characteristics such as:

function setup(block)

  % Register the number of ports.
  block.NumInputPorts  = 6;
  block.NumOutputPorts = 4;

  % Set up the port properties to be inherited or dynamic.
  block.SetPreCompInpPortInfoToDynamic;
  block.SetPreCompOutPortInfoToDynamic;

  % Register the parameters.
  block.NumDialogPrms     = 0;

  % Register the sample times.
  %  [0 offset]            : Continuous sample time
  %  [positive_num offset] : Discrete sample time
  %
  %  [-1, 0]               : Inherited sample time
  %  [-2, 0]               : Variable sample time
  block.SampleTimes = [0 0];

   block.SimStateCompliance = 'DefaultSimState';
```

```
   %% relevant
   block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
   block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
  block.RegBlockMethod('Outputs', @Outputs);
  block.RegBlockMethod('Terminate', @Terminate);
  block.RegBlockMethod('Start', @Start);
  block.RegBlockMethod('Update', @Update);
  %%outputs
  %%call to generate out puts

 function DoPostPropSetup(block)
block.NumDworks = 1;

  block.Dwork(1).Name            = 'x1';
  block.Dwork(1).Dimensions      = 1;
  block.Dwork(1).DatatypeID      = 0;      % double
  block.Dwork(1).Complexity      = 'Real'; % real
  block.Dwork(1).UsedAsDiscState = true;

  function Start(block)
block.Dwork(1).Data = 0;
 function Outputs(block)
m1=block.InputPort(1).Data;
m2=block.InputPort(2).Data;
qd2=block.InputPort(3).Data;
m4=block.InputPort(4).Data;
m5=block.InputPort(5).Data;
m6=block.InputPort(6).Data;
block.OutputPort(1).Data =  m1+(m2*cos(qd2));
block.OutputPort(2).Data = m4+(m5*cos(qd2));
block.OutputPort(3).Data = m4+(m5*cos(qd2));
block.OutputPort(4).Data = m6;
%% set the sampling of the out put ports
function Update(block)

block.Dwork(1).Data = block.InputPort(1).Data;
block.Dwork(1).Data = block.InputPort(2).Data;
block.Dwork(1).Data = block.InputPort(3).Data;
block.Dwork(1).Data = block.InputPort(4).Data;
block.Dwork(1).Data = block.InputPort(5).Data;
block.Dwork(1).Data = block.InputPort(6).Data;
%% set the sampling of the out put ports
 function SetInpPortFrameData(block, idx, fd)

  block.InputPort(idx).SamplingMode = fd;
  block.OutputPort(1).SamplingMode  = fd;
  block.OutputPort(2).SamplingMode  = fd;
  block.OutputPort(3).SamplingMode  = fd;
  block.OutputPort(4).SamplingMode  = fd;
%endfunction


function Terminate(~)

%%end
```

Level-2 mat-lab s-function programming for corioles for both controller

```
function corSMC(block)

% calls to the main body of the function.
setup(block);

%endfunction

%   Set up the S-function block's basic characteristics such as:

function setup(block)

  % Register the number of ports.
  block.NumInputPorts  = 5;
  block.NumOutputPorts = 2;

  % Set up the port properties to be inherited or dynamic.
  block.SetPreCompInpPortInfoToDynamic;
  block.SetPreCompOutPortInfoToDynamic;

  % Register the parameters.
  block.NumDialogPrms     = 0;

  % Register the sample times.
  %  [0 offset]            : Continuous sample time
  %  [positive_num offset] : Discrete sample time
  %
  %  [-1, 0]               : Inherited sample time
  %  [-2, 0]               : Variable sample time
  block.SampleTimes = [0 0];

   block.SimStateCompliance = 'DefaultSimState';
   %% relevant
  block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
  block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
  block.RegBlockMethod('Outputs', @Outputs);
  block.RegBlockMethod('Terminate', @Terminate);
  block.RegBlockMethod('Update', @Update);
   block.RegBlockMethod('Start', @Start);
  %%outputs
  %%call to generate out puts
  function DoPostPropSetup(block)
block.NumDworks = 1;

  block.Dwork(1).Name            = 'x1';
  block.Dwork(1).Dimensions      = 1;
  block.Dwork(1).DatatypeID      = 0;      % double
  block.Dwork(1).Complexity      = 'Real'; % real
  block.Dwork(1).UsedAsDiscState = true;
```

```
  function Start(block)
block.Dwork(1).Data = 0;
 function Outputs(block)
 c1=block.InputPort(1).Data;
 qd2=block.InputPort(2).Data;
 dqd2=block.InputPort(3).Data;
 c2=block.InputPort(4).Data;
 dqd1=block.InputPort(5).Data;
block.OutputPort(1).Data = c1*sin(qd2)*(dqd2)*(dqd2);
block.OutputPort(2).Data = c2*sin(qd2)*(dqd1)*(dqd1);

%% set the sampling of the out put ports
function Update(block)
block.Dwork(1).Data = block.InputPort(1).Data;
block.Dwork(1).Data = block.InputPort(2).Data;
block.Dwork(1).Data = block.InputPort(3).Data;
block.Dwork(1).Data = block.InputPort(4).Data;
block.Dwork(1).Data = block.InputPort(5).Data;


function SetInpPortFrameData(block, idx, fd)

  block.InputPort(idx).SamplingMode = fd;
  for i=1:block.NumOutputPorts
  block.OutputPort(i).SamplingMode  = fd;
  end
%endfunction



function Terminate(~)


%endfunction
```

Level-2 mat-lab s-function programming for centrifugal of the plant for PID

```
function centfSMC(block)

% calls to the main body of the function.
setup(block);

%endfunction

%   Set up the S-function block's basic characteristics such as:

function setup(block)

  % Register the number of ports.
  block.NumInputPorts  = 4;
  block.NumOutputPorts = 2;

  % Set up the port properties to be inherited or dynamic.
  block.SetPreCompInpPortInfoToDynamic;
```

```
  block.SetPreCompOutPortInfoToDynamic;

  % Register the parameters.
  block.NumDialogPrms     = 0;

  % Register the sample times.
  %  [0 offset]            : Continuous sample time
  %  [positive_num offset] : Discrete sample time
  %
  %  [-1, 0]               : Inherited sample time
  %  [-2, 0]               : Variable sample time
  block.SampleTimes = [0 0];

   block.SimStateCompliance = 'DefaultSimState';
   %% relevant
  block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
  block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
  block.RegBlockMethod('Start', @Start);
  block.RegBlockMethod('Outputs', @Outputs);
  block.RegBlockMethod('Terminate', @Terminate);
   block.RegBlockMethod('Start', @Start);
  block.RegBlockMethod('Update', @Update);
   block.RegBlockMethod('Terminate', @Terminate);
  %%outputs
  %%call to generate out puts
  function DoPostPropSetup(block)
block.NumDworks = 1;

  block.Dwork(1).Name            = 'x1';
  block.Dwork(1).Dimensions      = 1;
  block.Dwork(1).DatatypeID      = 0;      % double
  block.Dwork(1).Complexity      = 'Real'; % real
  block.Dwork(1).UsedAsDiscState = true;

  function Start(block)
block.Dwork(1).Data = 0;

 function Outputs(block)
 b1=block.InputPort(1).Data;
 qd2=block.InputPort(2).Data;
 dqd1=block.InputPort(3).Data;
 dqd2=block.InputPort(4).Data;
block.OutputPort(1).Data = b1*sin(qd2)*(dqd1)*(dqd2);
block.OutputPort(2).Data =b1*sin(qd2)*(dqd1)*(dqd2);

%% set the sampling of the out put ports
function Update(block)

block.Dwork(1).Data = block.InputPort(1).Data;
block.Dwork(1).Data = block.InputPort(2).Data;
block.Dwork(1).Data = block.InputPort(3).Data;
block.Dwork(1).Data = block.InputPort(4).Data;

%% set the sampling of the out put ports
function SetInpPortFrameData(block, idx, fd)
```

```
  block.InputPort(idx).SamplingMode = fd;
  for i=1:block.NumOutputPorts
  block.OutputPort(i).SamplingMode  = fd;
  end
%endfunction



function Terminate(~)


%endfunction
```

Level-2 mat-lab s-function programming for gravity for both controller

```
function gravity1(block)

% calls to the main body of the function.
setup(block);

%endfunction

%   Set up the S-function block's basic characteristics such as:

function setup(block)

  % Register the number of ports.
  block.NumInputPorts  = 4;
  block.NumOutputPorts = 2;

  % Set up the port properties to be inherited or dynamic.
  block.SetPreCompInpPortInfoToDynamic;
  block.SetPreCompOutPortInfoToDynamic;

  % Register the parameters.
  block.NumDialogPrms     = 0;

  % Register the sample times.
  %  [0 offset]            : Continuous sample time
  %  [positive_num offset] : Discrete sample time
  %
  %  [-1, 0]               : Inherited sample time
  %  [-2, 0]               : Variable sample time
  block.SampleTimes = [0 0];

   block.SimStateCompliance = 'DefaultSimState';
   %% relevant
  block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
  block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
  block.RegBlockMethod('Outputs', @Outputs);
```

```
  block.RegBlockMethod('Terminate', @Terminate);
   block.RegBlockMethod('Start', @Start);
  block.RegBlockMethod('Update', @Update);
  %%outputs
  %%call to generate out puts
  function DoPostPropSetup(block)
block.NumDworks = 1;

  block.Dwork(1).Name            = 'x1';
  block.Dwork(1).Dimensions      = 1;
  block.Dwork(1).DatatypeID      = 0;      % double
  block.Dwork(1).Complexity      = 'Real'; % real
  block.Dwork(1).UsedAsDiscState = true;

  function Start(block)
block.Dwork(1).Data = 0;
 function Outputs(block)
g1=block.InputPort(1).Data;
qd2=block.InputPort(2).Data;
g2=block.InputPort(3).Data;
qd1=block.InputPort(4).Data;
block.OutputPort(1).Data =g1*cos(qd2)+g2*cos(qd1+qd2);
block.OutputPort(2).Data = g2*cos(qd1+qd2);

%% set the sampling of the out put ports
function Update(block)

block.Dwork(1).Data = block.InputPort(1).Data;
block.Dwork(1).Data = block.InputPort(2).Data;

%% set the sampling of the out put ports
function SetInpPortFrameData(block, idx, fd)

  block.InputPort(idx).SamplingMode = fd;
  for i=1:block.NumOutputPorts
  block.OutputPort(i).SamplingMode  = fd;
  end
%endfunction



function Terminate(~)


%endfunction
```

Level-2 mat-lab s-function programming for inverse of inertia matrix for both controller

```
function inversemasSMC(block)
```

```
% calls to the main body of the function.
setup(block);

%endfunction

%   Set up the S-function block's basic characteristics such as:

function setup(block)

  % Register the number of ports.
  block.NumInputPorts  = 9;
  block.NumOutputPorts = 4;

  % Set up the port properties to be inherited or dynamic.
  block.SetPreCompInpPortInfoToDynamic;
  block.SetPreCompOutPortInfoToDynamic;

  % Register the parameters.
  block.NumDialogPrms     = 0;

  % Register the sample times.
  %  [0 offset]            : Continuous sample time
  %  [positive_num offset] : Discrete sample time
  %
  %  [-1, 0]               : Inherited sample time
  %  [-2, 0]               : Variable sample time
  block.SampleTimes = [0 0];

   block.SimStateCompliance = 'DefaultSimState';
   %% relevant
   block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
   block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
  block.RegBlockMethod('Outputs', @Outputs);
  block.RegBlockMethod('Terminate', @Terminate);
  block.RegBlockMethod('Start', @Start);
  block.RegBlockMethod('Update', @Update);
  %%outputs
  %%call to generate out puts

 function DoPostPropSetup(block)
block.NumDworks = 1;

  block.Dwork(1).Name            = 'x1';
  block.Dwork(1).Dimensions      = 1;
  block.Dwork(1).DatatypeID      = 0;      % double
  block.Dwork(1).Complexity      = 'Real'; % real
  block.Dwork(1).UsedAsDiscState = true;

  function Start(block)
block.Dwork(1).Data = 0;
 function Outputs(block)
 m11=block.InputPort(1).Data;
 m12=block.InputPort(2).Data;
 m13=block.InputPort(3).Data;
 m21=block.InputPort(4).Data;
```

```
 qd2=block.InputPort(5).Data;
 m22=block.InputPort(6).Data;
 m23=block.InputPort(7).Data;
 m31=block.InputPort(8).Data;
 m32=block.InputPort(9).Data;

block.OutputPort(1).Data = block.Dwork(1).Data + m11/(m12+m13-m21*(qd2*qd2));
block.OutputPort(2).Data =block.Dwork(1).Data + (m22-m23*cos(qd2))/(m12+m13-
m21*(qd2*qd2));
block.OutputPort(3).Data =block.Dwork(1).Data +(m22-m23*cos(qd2))/(m12+m13-
m21*(qd2*qd2));
block.OutputPort(4).Data =block.Dwork(1).Data +(m31+m32*cos(qd2))/(m12+m13-
m21*(qd2*qd2));
%% set the sampling of the out put ports
function Update(block)

block.Dwork(1).Data = block.InputPort(1).Data;

%% set the sampling of the out put ports
 function SetInpPortFrameData(block, idx, fd)

  block.InputPort(idx).SamplingMode = fd;
  block.OutputPort(1).SamplingMode  = fd;
  block.OutputPort(2).SamplingMode  = fd;
  block.OutputPort(3).SamplingMode  = fd;
  block.OutputPort(4).SamplingMode  = fd;


%endfunction


function Terminate(~)

%%end
```