PUMA-560 Robot Manipulator Position Sliding Mode Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses

Farzin Piltan *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering SanatkadeheSabze Pasargad. CO (S.S.P. Co), NO:16 ,PO.Code 71347-66773, Fourth floor Dena Apr , Seven Tir Ave , Shiraz , Iran

Sara Emamzadeh *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering SanatkadeheSabze Pasargad. CO (S.S.P. Co), NO:16 ,PO.Code 71347-66773, Fourth floor Dena Apr , Seven Tir Ave , Shiraz , Iran

Zahra Hivand *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering SanatkadeheSabze Pasargad. CO (S.S.P. Co), NO:16 ,PO.Code 71347-66773, Fourth floor Dena Apr , Seven Tir Ave , Shiraz , Iran

Forouzan Shahriyari *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering SanatkadeheSabze Pasargad. CO (S.S.P. Co), NO:16 ,PO.Code 71347-66773, Fourth floor Dena Apr , Seven Tir Ave , Shiraz , Iran

Mina Mirzaei *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering SanatkadeheSabze Pasargad. CO (S.S.P. Co), NO:16 ,PO.Code 71347-66773, Fourth floor Dena Apr , Seven Tir Ave , Shiraz , Iran

Abstract

This paper describes the MATLAB/SIMULINK realization of the PUMA 560 robot manipulator position control methodology. This paper focuses on two main areas, namely robot manipulator analysis and implementation, and design, analyzed and implement nonlinear sliding mode control (SMC) methods. These simulation models are developed as a part of a software laboratory to support and enhance graduate/undergraduate robotics courses, nonlinear control courses and MATLAB/SIMULINK courses at research and development company (SSP Co.) research center, Shiraz, Iran.

Keywords: MATLAB/SIMULINK, PUMA 560 Robot Manipulator, Position Control Method, Sliding Mode Control, Robotics, Nonlinear Control.

1. INTRODUCTION

Computer modeling, simulation and implementation tools have been widely used to support and develop nonlinear control, robotics, and MATLAB/SIMULINK courses. MATLAB with its toolboxes such as SIMULINK [1] is one of the most accepted software packages used by researchers to enhance teaching the transient and steady-state characteristics of control and robotic courses [3_7]. In an effort to modeling and implement robotics, nonlinear control and advanced MATLAB/SIMULINK courses at research and development SSP Co., authors have developed MATLAB/SIMULINK models for learn the basic information in field of nonlinear control and industrial robot manipulator [8, 9].

The international organization defines the robot as "an automatically controlled, reprogrammable, multipurpose manipulator with three or more axes." The institute of robotic in The United States Of America defines the robot as "a reprogrammable, multifunctional manipulator design to move material, parts, tools, or specialized devices through various programmed motions for the performance of variety of tasks"[1]. Robot manipulator is a collection of links that connect to each other by joints, these joints can be revolute and prismatic that revolute joint has rotary motion around an axis and prismatic joint has linear motion around an axis. Each joint provides one or more degrees of freedom (DOF). From the mechanical point of view, robot manipulator is divided into two main groups, which called; serial robot links and parallel robot links. In serial robot manipulator, links and joints is serially connected between base and final frame (end-effector). Parallel robot manipulators have many legs with some links and joints, where in these robot manipulators base frame has connected to the final frame. Most of industrial robots are serial links, which in n degrees of freedom serial link robot manipulator the axis of the first three joints has a known as major axis, these axes show the position of end-effector, the axis number four to six are the minor axes that use to calculate the orientation of end-effector and the axis number seven to n use to reach the avoid the difficult conditions (e.g., surgical robot and space robot manipulator). Kinematics is an important subject to find the relationship between rigid bodies (e.g., position and orientation) and end-effector in robot manipulator. The mentioned topic is very important to describe the three areas in robot manipulator: practical application such as trajectory planning, essential prerequisite for some dynamic description such as Newton's equation for motion of point mass, and control purposed therefore kinematics play important role to design accurate controller for robot manipulators. Robot manipulator kinematics is divided into two main groups: forward kinematics and inverse kinematics where forward kinematics is used to calculate the position and orientation of end-effector with given joint parameters (e.g., joint angles and joint displacement) and the activated position and orientation of end-effector calculate the joint variables in Inverse Kinematics[6]. Dynamic modeling of robot manipulators is used to describe the behavior of robot manipulator such as linear or nonlinear dynamic behavior, design of model based controller such as pure sliding mode controller and pure computed torque controller which design these controller are based on nonlinear dynamic equations, and for simulation. The dynamic modeling describes the relationship between joint motion, velocity, and accelerations to force/torque or current/voltage and also it can be used to describe the particular dynamic effects (e.g., inertia, coriolios, centrifugal, and the other parameters) to behavior of system[1]. The Unimation PUMA 560 serially links robot manipulator was used as a basis, because this robot manipulator is widely used in industry and academic. It has a nonlinear and uncertain dynamic parameters serial link 6 degrees of freedom (DOF) robot manipulator. A nonlinear robust controller design is major subject in this work [1-15].

Controller is a device which can sense information from linear or nonlinear system (e.g., robot manipulator) to improve the systems performance [3]. The main targets in designing control systems are stability, good disturbance rejection, and small tracking error[5]. Several industrial robot manipulators are controlled by linear methodologies (e.g., Proportional-Derivative (PD) controller, Proportional- Integral (PI) controller or Proportional- Integral-Derivative (PID) controller), but when robot manipulator works with various payloads and have uncertainty in dynamic models this technique has limitations. From the control point of view, uncertainty is divided into two main groups: uncertainty in unstructured inputs (e.g., noise, disturbance) and uncertainty in structure dynamics (e.g., payload, parameter variations). In some applications robot

manipulators are used in an unknown and unstructured environment, therefore strong mathematical tools used in new control methodologies to design nonlinear robust controller with an acceptable performance (e.g., minimum error, good trajectory, disturbance rejection. Sliding mode controller is a powerful nonlinear robust controller under condition of partly uncertain dynamic parameters of system [7]. This controller is used to control of highly nonlinear systems especially for robot manipulators. Chattering phenomenon and nonlinear equivalent dynamic formulation in uncertain dynamic parameter are two main drawbacks in pure sliding mode controller [20]. The main reason to opt for this controller is its acceptable control performance in wide range and solves two most important challenging topics in control which names, stability and robustness [7, 17-20]. Sliding mode controller is divided into two main sub controllers: discontinues controller($\boldsymbol{\tau_{dis}}$) and equivalent controller($\boldsymbol{\tau_{eq}}$). Discontinues controller causes an acceptable tracking performance at the expense of very fast switching. Conversely in this theory good trajectory following is based on fast switching, fast switching is caused to have system instability and chattering phenomenon. Fine tuning the sliding surface slope is based on nonlinear equivalent part [1, 6]. However, this controller is used in many applications but, pure sliding mode controller has two most important challenges: chattering phenomenon and nonlinear equivalent dynamic formulation in uncertain parameters[20]. Chattering phenomenon (Figure 1) can causes some problems such as saturation and heat the mechanical parts of robot manipulators or drivers. To reduce or eliminate the chattering, various papers have been reported by many researchers which classified into two most important methods: boundary layer saturation method and estimated uncertainties method [1].

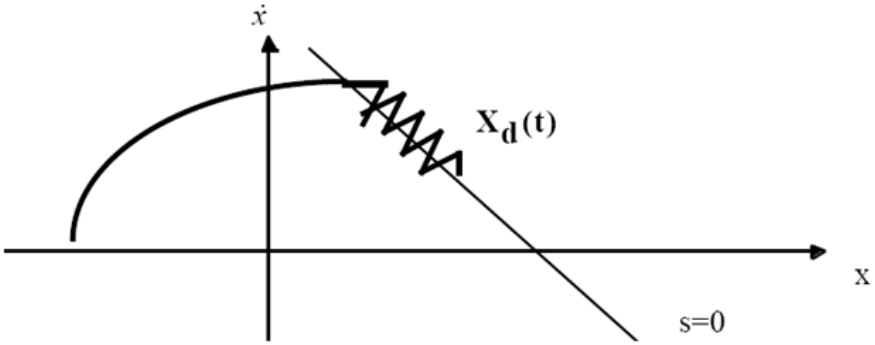


FIGURE 1: Chattering as a result of imperfect control switching [1].

In boundary layer saturation method, the basic idea is the discontinuous method replacement by saturation (linear) method with small neighborhood of the switching surface. This replacement caused to increase the error performance against with the considerable chattering reduction. Slotine and Sastry have introduced boundary layer method instead of discontinuous method to reduce the chattering[21]. Slotine has presented sliding mode with boundary layer to improve the industry application [22]. Palm has presented a fuzzy method to nonlinear approximation instead of linear approximation inside the boundary layer to improve the chattering and control the result performance[23]. Moreover, Weng and Yu improved the previous method by using a new method in fuzzy nonlinear approximation inside the boundary layer and adaptive method[24]. As mentioned [24]sliding mode fuzzy controller (SMFC) is fuzzy controller based on sliding mode technique to most exceptional stability and robustness. Sliding mode fuzzy controller has the two most important advantages: reduce the number of fuzzy rule base and increase robustness and stability. Conversely sliding mode fuzzy controller has the above advantages, define the sliding surface slope coefficient very carefully is the main disadvantage of this controller. Estimated uncertainty method used in term of uncertainty estimator to compensation of the system uncertainties. It has been used to solve the chattering phenomenon and also nonlinear equivalent dynamic. If estimator has an acceptable performance to compensate the uncertainties, the chattering is reduced. Research on estimated uncertainty to reduce the chattering is significantly growing as their applications such as industrial automation and robot manipulator. For instance, the applications of artificial intelligence, neural networks and fuzzy logic on

estimated uncertainty method have been reported in [25-28]. Wu et al. [30] have proposed a simple fuzzy estimator controller beside the discontinuous and equivalent control terms to reduce the chattering. Their design had three main parts i.e. equivalent, discontinuous and fuzzy estimator tuning part which has reduced the chattering very well. Elmali et al. [27]and Li and Xu [29] have addressed sliding mode control with perturbation estimation method (SMCPE) to reduce the classical sliding mode chattering. This method was tested for the tracking control of the first two links of a SCARA type HITACHI robot. In this technique, digital controller is used to increase the system's response quality. However this controller's response is very fast and robust but it has chattering phenomenon. Design a robust controller for robot manipulator is essential because robot manipulator has highly nonlinear dynamic parameters.

This paper is organized as follows:
In section 2, dynamic and kinematics formulation of robot manipulator and methodology of implemented of them are presented. Detail of classical SMC and MATLAB/SIMULINK implementation of this controller is presented in section 3. In section 4, the simulation result is presented and finally in section 5, the conclusion is presented.

2. PUMA 560 ROBOT MANIPULATOR FORMULATION: DYNAMIC FORMULATION OF ROBOTIC MANIPULATOR AND KINEMATICS FORMULATION OF ROBOTIC MANIPULATOR

Rigid-body kinematics: one of the main concern among robotic and control engineers is positioning the manipulator's End-effector to the most accurate place and transparent the effect of disturbance and errors which will affect on manipulator's final result. As a matter of fact, controlling manipulators are hard and expensive because they are multi-input, multi-output, time variant and non-linear, so it has been a topic for researchers to design the most sufficient controller to help the manipulator to achieve to the desired expectation under any circumstance. PUMA 560 is a good instance for manipulators, because it is widely used in both industry and academic, and the dynamic parameters for this robot arm have been identified and documented in literature. One of the main parts of a manipulator's controller is its kinematics which can be divided into two parts; forward kinematics and inverse kinematics. Implementation of inverse kinematic is hard and expensive. In this work we will aim on implementation of PUMA 560 robot manipulator kinematics. Study of robot manipulators is classified into two main groups: kinematics and dynamics. Calculate the relationship between rigid bodies and end-effector without any forces is called Robot manipulator Kinematics. Study of this part is pivotal to calculate accurate dynamic part, to design with an acceptable performance controller, and in real situations and practical applications. As expected the study of manipulator kinematics is divided into two main parts: forward and inverse kinematics. Forward kinematics has been used to find the position and orientation of task (end-effector) frame when angles and/or displacement of joints are known. Inverse kinematics has been used to find possible joints variable (displacements and angles) when all position and orientation of end-effector be active [1].

The main target in forward kinematics is calculating the following function:

$$\boldsymbol{\Psi}(\boldsymbol{X},\boldsymbol{q}) = \mathbf{0} \quad \textbf{(1)}$$

Where $\Psi(.) \in R^n$ is a nonlinear vector function, $X = [X_1, X_2, \ldots\ldots, X_l]^T$ is the vector of task space variables which generally endeffector has six task space variables, three position and three orientation, $q = [q_1, q_2, \ldots., q_n]^T$ is a vector of angles or displacement, and finally n is the number of actuated joints.

The Denavit-Hartenberg (D-H) convention is a method of drawing robot manipulators free body diagrams. Denvit-Hartenberg (D-H) convention study is necessary to calculate forward kinematics in serial robot manipulator. The first step to calculate the serial link robot manipulator forward kinematics is link description; the second step is finding the D-H convention after the frame attachment and finally finds the forward kinematics. Forward kinematics is a 4×4 matrix which 3×3 of them shows the rotation matrix, 3×1 of them is shown the position vector and last

four cells are scaling factor[1, 6]. Wu has proposed PUMA 560 robot arm kinematics based on accurate analysis [9].

The inverse kinematics problem is calculation of joint variables (i.e., displacement and angles), when position and orientation of end-effector to be known. In other words, the main target in inverse kinematics is to calculate $q = h^{-1}(X)$, where q is joint variable, q =[$q_1, q_2, \ldots\ldots, q_n$], and X are position and orientation of endeffector, X=[X,$Y, Z, \emptyset, \theta, \Psi$]. In general analysis the inverse kinematics of robot manipulator is difficult because, all nonlinear equations solutions are not unique (e.g., redundant robot, elbow-up/elbow-down rigid body), and inverse kinematics are different for different types of robots. In serial links robot manipulators, equations of inverse kinematics are classified into two main groups: numerical solutions and closed form solutions. Most of researcher works on closed form solutions of inverse kinematics with different methods, such as inverse transform, screw algebra, dual matrix, iterative, geometric approach and decoupling of position and orientation[1, 6]. Research on the Inverse Kinematics robot manipulator PUMA 560 series, like in some applications has been working. For instance, Zhang and Paul have worked on particular way of robot kinematics solution to reduce the computation[10]. Kieffer has proposed a simple iterative solution to computation of inverse kinematics[11]. Ahmad and Guez are solved the robot manipulator inverse kinematics by neural network hybrid method which this method is combining the advantages of neural network and iterative methods [12].

Singularity is a location in the robot manipulator's workspace which the robot manipulator loses one or more degrees of freedom in Cartesian space. Singularities are one of the most important challenges in inverse kinematics which Cheng et al., have proposed a method to solve this problem [13]. A systematic Forward Kinematics of robot manipulator solution is the main target of this part. The first step to compute Forward Kinematics (F.K) of robot manipulator is finding the standard D-H parameters. Figure 2 shows the schematic of the PUMA 560 robot manipulator.
The following steps show the systematic derivation of the standard D-H parameters.

1. Locate the robot arm
2. Label joints
3. Determine joint rotation or translation ($\theta \; or \; d$)
4. Setup base coordinate frames.
5. Setup joints coordinate frames.
6. Determineα_i, thatα_i, link twist, is the angle between Z_i and Z_{i+1} about an X_i.
7. Determine d_i and a_i , that a_i, link length, is the distance between Z_i and Z_{i+1} along X_i. d_i, offset, is the distance between X_{i-1} and X_i along Z_i axis.
8. Fill up the D-H parameters table. Table 1 shows the standard D-H parameters for n DOF robot manipulator.

The second step to compute Forward kinematics for robot manipulator is finding the rotation matrix (R_n^0). The rotation matrix from$\{F_i\}$ to $\{F_{i-1}\}$ is given by the following equation;

$$\boldsymbol{R_i^{i-1} = U_{i(\theta_i)} V_{i(\alpha_i)}} \quad \textbf{(2)}$$

Where $U_{i(\theta_i)}$ is given by the following equation [1];

$$\boldsymbol{U_{i(\theta_i)} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}} \quad \textbf{(3)}$$

and $V_{i(\alpha_i)}$ is given by the following equation [1];

$$\boldsymbol{V_{i(\theta_i)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_i) & -\sin(\theta_i) \\ 0 & \sin(\theta_i) & \cos(\theta_i) \end{bmatrix}} \quad \textbf{(4)}$$

So (R_n^0) is given by [1]

$$\boldsymbol{R_n^0 = (U_1V_1)(U_2V_2) \dots \dots \dots (U_nV_n)} \quad \textbf{(5)}$$

Link i	θ_i(rad)	α_i(rad)	a_i(m)	d_i(m)
1	θ_1	α_1	a_1	d_1
2	θ_2	α_2	a_2	d_2
3	θ_3	α_3	a_3	d_3
.....	.....	.....	.....	.....
.....	.....	.....	.....	.....
n	θ_n	n	a_5	d_n

TABLE 1: The Denavit Hartenberg parameter

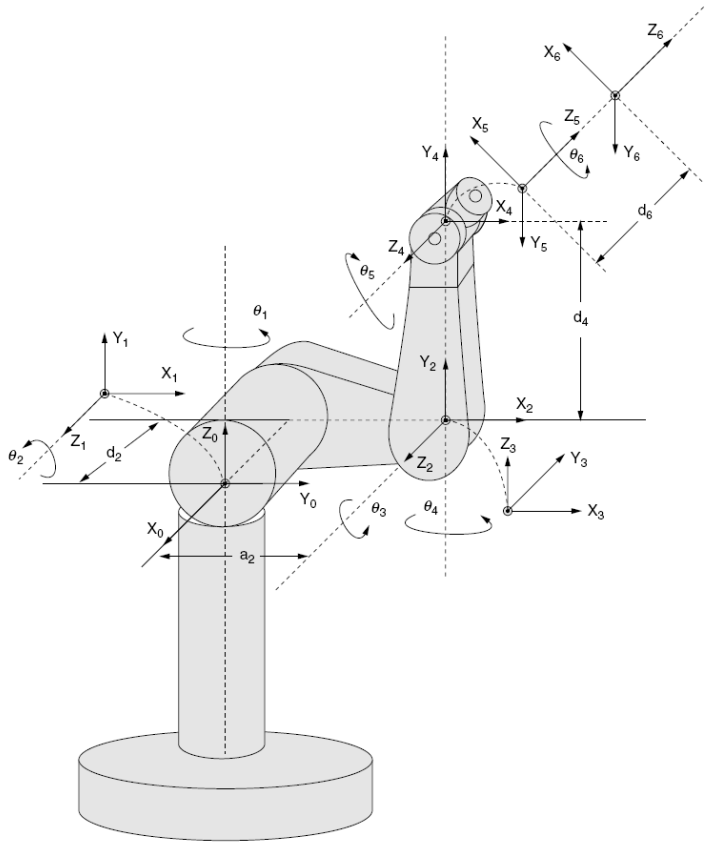


FIGURE 2: D-H notation for a six-degrees-of-freedom PUMA 560 robot manipulator[2]

The third step to compute the forward kinematics for robot manipulator is finding the displacement vector d_n^0, that it can be calculated by the following equation [1]

$$\boldsymbol{d_n^0 = (U_1 S_1) + (U_1 V_1)(U_2 S_2) + \cdots + (U_1 V_1)(U_2 V_2) \ldots . (U_{n-1} V_{n-1})(U_n S_n)} \quad \textbf{(6)}$$

The forth step to compute the forward kinematics for robot manipulator is calculate the transformation ${}_n^0T$ by the following formulation [1]

$$\boldsymbol{{}_n^0T = {}_1^0T.\,{}_2^1T.\,{}_3^2T \ldots \ldots {}_n^{n-1}T = \begin{bmatrix} R_n^0 & d_n^0 \\ 0 & 1 \end{bmatrix}} \quad \textbf{(7)}$$

Kinematics of PUMA 560 robot manipulator: In PUMA robot manipulator the final transformation matrix is given by

$$\boldsymbol{{}_6^0T = {}_1^0T.\,{}_2^1T.\,{}_3^2T \ldots \ldots {}_6^5T = \begin{bmatrix} R_6^0 & d_6^0 \\ 0 & 1 \end{bmatrix}} \quad \textbf{(8)}$$

That R_6^0 and d_6^0 is given by the following matrix

$$\boldsymbol{R_6^0 = \begin{bmatrix} N_x & B_x & T_x \\ N_y & B_y & T_y \\ N_z & B_z & T_z \end{bmatrix}; \quad d_6^0 = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}} \quad \textbf{(9)}$$

That ${}_6^0T$ can be determined by

$$\boldsymbol{{}_6^0T = \begin{bmatrix} N_x & B_x & T_x & P_x \\ N_y & B_y & T_y & P_y \\ N_z & B_z & T_z & P_z \end{bmatrix}} \quad \textbf{(10)}$$

Table 2 shows the PUMA 560 D-H parameters.

Link i	θ_i(rad)	α_i(rad)	a_i(m)	d_i(m)
1	θ_1	$-\pi/2$	0	0
2	θ_2	0	0.4318	0.14909
3	θ_3	$\pi/2$	0.0203	0
4	θ_4	$-\pi/2$	0	0.43307
5	θ_5	$\pi/2$	0	0
6	θ_6	0	0	0.05625

TABLE 2: PUMA 560 robot manipulator DH parameter [4].

As equation 8 the cells of above matrix for PUMA 560 robot manipulator is calculated by following equations:

$$\boldsymbol{Nx = cos(\theta\_6) \times (cos(\theta\_5) \times (cos(\theta\_4) \times cos(\theta\_2 + \theta\_3) \times cos(\theta\_1) + sin(\theta\_1) \times sin(\theta\_1)) + sin(\theta\_5) \times sin(\theta\_2 + \theta\_3) \times cos(\theta\_1)) + sin(\theta\_6) \times (sin(\theta\_4) \times cos(\theta\_2 + \theta\_3) \times cos(\theta\_1) - cos(\theta\_4) \times sin(\theta\_1))} \quad \textbf{(11)}$$

$$Ny = cos(\theta_6) \times (cos(\theta_5) \times (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) + sin(\theta_5) \times sin(\theta_2 + \theta_3) \times sin(\theta_1)) + sin(\theta_6) \times (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + cos(\theta_4) \times cos(\theta_1)) \quad (12)$$

$$Nz = cos(\theta_6) \times (cos(\theta_5) \times cos(\theta_4) \times sin(\theta_2 + \theta_3) - sin(\theta_5) \times cos(\theta_2 + \theta_3)) + sin(\theta_6) \times sin(\theta_4) \times sin(\theta_2 + \theta_3) \quad (13)$$

$$Bx = -sin(\theta_6) \times (cos(\theta_5) \times (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) + sin(\theta_4) \times sin(\theta_1)) + sin(\theta_5) \times sin(\theta_2 + \theta_3) \times cos(\theta_1)) + cos(\theta_6) \times (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) - cos(\theta_4) \times sin(\theta_1)) \quad (14)$$

$$By = -sin(\theta_6) \times (cos(\theta_5) \times (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) + sin(\theta_5) \times sin(\theta_2 + \theta_3) \times sin(\theta_1)) + cos(\theta_6) \times (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + cos(\theta_4) \times cos(\theta_1)) \quad (15)$$

$$Bz = -sin(\theta_6) \times (cos(\theta_5) \times cos(\theta_4) \times sin(\theta_2 + \theta_3) - sin(\theta_5) \times cos(\theta_2 + \theta_3)) + cos(\theta_6) \times sin(\theta_4) \times sin(\theta_2 + \theta_3) \quad (16)$$

$$Tx = sin(\theta_5) \times (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) + sin(\theta_4) \times sin(\theta_1)) - cos(\theta_5) \times sin(\theta_2 + \theta_3) \times cos(\theta_1) \quad (17)$$

$$Ty = sin(\theta_5) \times (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) - cos(\theta_5) \times sin(\theta_2 + \theta_3) \times sin(\theta_1) \quad (18)$$

$$Tz = sin(\theta_5) \times cos(\theta_4) \times sin(\theta_2 + \theta_3) + cos(\theta_5) \times cos(\theta_2 + \theta_3) \quad (19)$$

$$Px = 0.4331 \times sin(\theta_2 + \theta_3) \times cos(\theta_1) + 0.0203 \times cos(\theta_2 + \theta_3) \times cos(\theta_1) - 0.1491 \times sin(\theta_1) + 0.4318 \times cos(\theta_2 cos(\theta_1) \quad (20)$$

$$Py = 0.4331 \times sin(\theta_2 + \theta_3) \times sin(\theta_1) + 0.0203 \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + 0.1491 \times cos(\theta_1) + 0.4312 \times cos(\theta_2) \times sin(\theta_1) \quad (21)$$

$$Pz = -0.4331 \times cos(\theta_2 + \theta_3) + 0.0203 \times sin(\theta_2 + \theta_3) + 0.4318 \times sin(\theta_2) \quad (22)$$

PUMA-560 Kinematics Implementation Using MATLAB/SIMULINK

robot manipulator kinematics is essential part to calculate the relationship between rigid bodies and end-effector without any forces. Study of this part is fundamental to calculate accurate dynamic part, to design a controller with acceptable performance, and finally in real situations and particular applications.

In forward kinematics, variables of joints (revolute or prismatic) is given and position and orientation (pose) of rigid body is desired *(*Figure 3*).* In revolute joints the variables are θ_i which means it's joint angle with its neighbor joint. If the joint is prismatic, the variable is d_i which means link offset between joints. In forward kinematics the final result is a 4×4 matrix which 3 factors of it, is end-effector's position and 9 is it's orientation as shown in Figure 10.

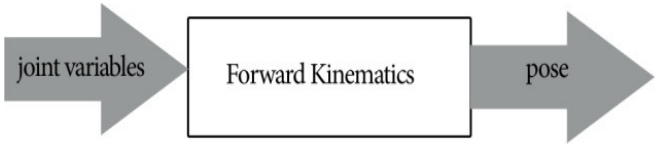


FIGURE3: Forward kinematics block diagram

Desired input is our goal. It means that we are expecting our End-effector to reach at that point.

Sometimes the result that manipulator is reaching at is not what we were expecting for. The main cause of this problem is the disturbances which effects on our system. Nonetheless to say, these disturbances are unwanted, affect on the result and they are the main reason for controller designing. Actual input means the point that end-effector has reached as a result. Actual input, if the disturbance does not affect on our system is the same as desired input and if it affect, is far from the desired input. Clearly saying, if the desired input and actual input become different, the meaning is that the end-effector has not reached to the expected point. At the very first place we must define our system .The system that we are working on is PUMA 560 which has 6 degrees of freedom (6 DOF) and it's joints are RRR. It means that all joints are revolute. As mentioned before, due to type of joints, desired inputs are varied. The joints of the system we are working on are RRR which means they are revolute. So, system's variables are θ_i.as shown in Figure 4, we have 12 inputs and 24 outputs. Inputs are both desired inputs and actual inputs. Our goal is testing if the actual result has reached to our desired. Also its outputs are two 4×4 matrixes. In every matrix, three of factors, show position and nine factors show orientation *(*Figure 4*).*

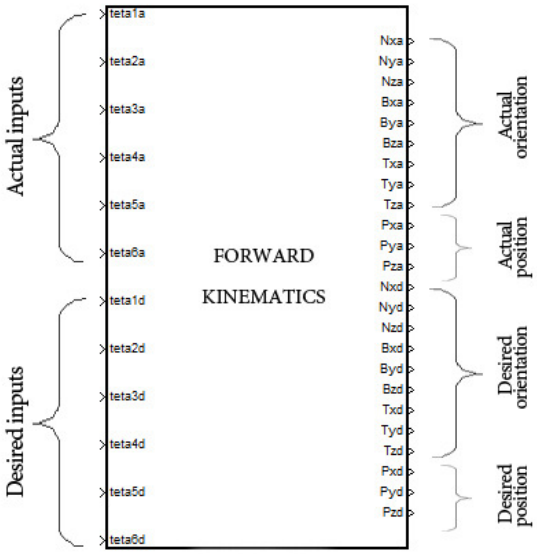


FIGURE 4: Forward kinematics block diagram: inputs and outputs

Note that we aim on controlling the position and we do not work on orientation. So in this system, we do not work on actual orientation and desired orientation. In the next step, we must implement the block diagram of our kinematics. Table 3 shows the input and outputs used in our kinematics block diagram. Also Table 4 and Table 5 show formulation for each variable used in kinematics.

Nxa,Nya,Nza,Bxa,Bya,Bza,Txa,Tya,Tza,Pxa,Pya,Pza	teta1a,teta2a,teta3a,teta4d,teta5d,teta6a,teta1d
Nxd,Nyd,Nzd,Bxd,Byd,Bzd,Txd,Tyd,Tzd,Pxd,Pyd,Pzd	teta2d,teta3d,teta4d,teta5d,teta6d

TABLE3: Inputs and outputs of kinematics

Nxa	cos(teta6a)*(cos(teta5d)*(cos(teta4d)*cos(teta2a+teta3a)*cos(teta1a)+sin(teta4d)*sin(teta1a))+ sin(teta5d)*sin(teta2a+teta3a)*cos(teta1a))+sin(teta6a)*(sin(teta4d)*cos(teta2a+teta3a)*cos(tet a1a)-cos(teta4d) * sin(teta1a))
Nya	cos(teta6a)*(cos(teta5d)*(cos(teta4d)*cos(teta2a+teta3a)*sin(teta1a)- sin(teta4d)*cos(teta1a))+sin(teta5d)*sin(teta2a+teta3a)*sin(teta1a))+sin(teta6a)*(sin(teta4d)*co s(teta2a+teta3a)*sin(teta1a)+cos(teta4d)* cos(teta1a))
Nza	cos(teta6a)*(cos(teta5d)*cos(teta4d)*sin(teta2a+teta3a)- sin(teta5d)* cos(teta2a+teta3a))+sin(teta6a)*sin(teta4d)*sin(teta2a+teta3a)
Bxa	-sin(teta6a)*(cos(teta5d)*(cos(teta4d)*cos(teta2a+teta3a)*cos(teta1a) +sin(teta4d)*sin(teta1a))+sin(teta5d)*sin(teta2a+teta3a)*cos(teta1a))+cos(teta6a)*(sin(teta4d)* cos(teta2a+teta3a)*cos(teta1a)-cos(teta4d)*sin(teta1a))
Bya	-sin(teta6a)*(cos(teta5d)*(cos(teta4d)*cos(teta2a+teta3a)*sin(teta1a) - sin(teta4d)*cos(teta1a))+sin(teta5d)*sin(teta2a+teta3a)*sin(teta1a)) +cos(teta6a)*(sin(teta4d)*cos(teta2a+teta3a)*sin(teta1a)+cos(teta4d)*cos(teta1a))
Bza	-sin(teta6a)*(cos(teta5d)*cos(teta4d)*sin(teta2a+teta3a)-sin(teta5d)* cos(teta2a+teta3a))+cos(teta6a)*sin(teta4d)*sin(teta2a+teta3a)
Txa	sin(teta5d)*(cos(teta4d)*cos(teta2a+teta3a)*cos(teta1a)+sin(teta4d)* sin(teta1a))- cos(teta5d)*sin(teta2a+teta3a)*cos(teta1a)
Tya	sin(teta5d)*(cos(teta4d)*cos(teta2a+teta3a)*sin(teta1a)-sin(teta4d) *cos(teta1a))- cos(teta5d)*sin(teta2a+teta3a)*sin(teta1a)
Tza	sin(teta5d)*cos(teta4d)*sin(teta2a+teta3a)+cos(teta5d)* cos(teta2a+teta3a)
Pxa	0.4331*sin(teta2a+teta3a)*cos(teta1a)+0.0203*cos(teta2a+teta3a)* cos(teta1a)- 0.1491*sin(teta1a)+0.4318*cos(teta2a)*cos(teta1a)
Pya	0.4331*sin(teta2a+teta3a)*sin(teta1a)+0.0203*cos(teta2a+teta3a)* sin(teta1a)+0.1491*cos(teta1a)+0.4312*cos(teta2a)*sin(teta1a)
Pza	-0.4331*cos(teta2a+teta3a)+0.0203*sin(teta2a+teta3a)+0.4318* sin(teta2a)

TABLE 4: Actual input formulas

Nxd	cos(teta6d)*(cos(teta5d)*(cos(teta4d)*cos(teta2d+teta3d)*cos(teta1d)+sin(teta4d)*sin(teta1d))+ sin(teta5d)*sin(teta2d+teta3d)*cos(teta1d))+sin(teta6d)*(sin(teta4d)*cos(teta2d+teta3d)*cos(teta1d)-cos(teta4d) * sin(teta1d))
Nyd	cos(teta6d)*(cos(teta5d)*(cos(teta4d)*cos(teta2d+teta3d)*sin(teta1d)- sin(teta4d)*cos(teta1d))+sin(teta5d)*sin(teta2d+teta3d)*sin(teta1d))+sin(teta6d)*(sin(teta4d)*cos(teta2d+teta3d)*sin(teta1d)+cos(teta4d)* cos(teta1d))
Nzd	cos(teta6d)*(cos(teta5d)*cos(teta4d)*sin(teta2d+teta3d)- sin(teta5d)* cos(teta2d+teta3d))+sin(teta6d)*sin(teta4d)*sin(teta2d+teta3d)
Bxd	-sin(teta6d)*(cos(teta5d)*(cos(teta4d)*cos(teta2d+teta3d)*cos(teta1d) +sin(teta4d)*sin(teta1d))+sin(teta5d)*sin(teta2d+teta3d)*cos(teta1d))+cos(teta6d)*(sin(teta4d)* cos(teta2d+teta3d)*cos(teta1d)-cos(teta4d)*sin(teta1d))
Byd	-sin(teta6d)*(cos(teta5d)*(cos(teta4d)*cos(teta2d+teta3d)*sin(teta1d) - sin(teta4d)*cos(teta1d))+sin(teta5d)*sin(teta2d+teta3d)*sin(teta1d)) +cos(teta6d)*(sin(teta4d)*cos(teta2d+teta3d)*sin(teta1d)+cos(teta4d)*cos(teta1d))
Bzd	-sin(teta6d)*(cos(teta5d)*cos(teta4d)*sin(teta2d+teta3d)-sin(teta5d)* cos(teta2d+teta3d))+cos(teta6d)*sin(teta4d)*sin(teta2d+teta3d)
Txd	sin(teta5d)*(cos(teta4d)*cos(teta2d+teta3d)*cos(teta1d)+sin(teta4d)* sin(teta1d))- cos(teta5d)*sin(teta2d+teta3d)*cos(teta1d)
Tyd	sin(teta5d)*(cos(teta4d)*cos(teta2d+teta3d)*sin(teta1d)-sin(teta4d) *cos(teta1d))- cos(teta5d)*sin(teta2d+teta3d)*sin(teta1d)
Tzd	sin(teta5d)*cos(teta4d)*sin(teta2d+teta3d)+cos(teta5d)* cos(teta2d+teta3d)
Pxd	0.4331*sin(teta2d+teta3d)*cos(teta1d)+0.0203*cos(teta2d+teta3d)* cos(teta1d)- 0.1491*sin(teta1d)+0.4318*cos(teta2d)*cos(teta1d)
Pyd	0.4331*sin(teta2d+teta3d)*sin(teta1d)+0.0203*cos(teta2d+teta3d)* sin(teta1d)+0.1491*cos(teta1d)+0.4312*cos(teta2d)*sin(teta1d)
Pzd	-0.4331*cos(teta2d+teta3d)+0.0203*sin(teta2d+teta3d)+0.4318* sin(teta2d)

TABLE 5: Desired input formulas

As mentioned before, we aim on position controlling. so we must connect desired position and actual position to RMS error block diagram to find out whether the end-effector has reached to expected point or not. Kinematics of our system is shown in Figure 5.

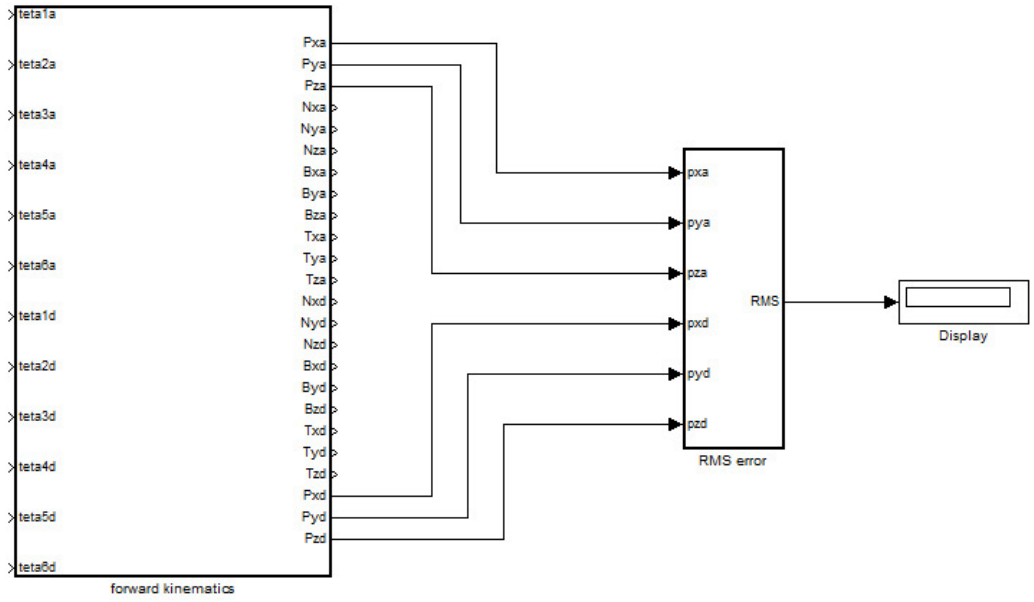


FIGURE 5: Kinematics of PUMA 560

Dynamic of Robot Manipulator

Dynamic equation is the study of motion with regard to forces. Dynamic modeling is vital for control, mechanical design, and simulation. It is used to describe dynamic parameters and also to describe the relationship between displacement, velocity and acceleration to force acting on robot manipulator. To calculate the dynamic parameters which introduced in the following lines, four algorithms are very important.

i. **Inverse dynamics**, in this algorithm, joint actuators are computed (e.g., force/torque or voltage/current) from endeffector position, velocity, and acceleration. It is used in feed forward control.

ii. **Forward dynamics** used to compute the joint acceleration from joint actuators. This algorithm is required for simulations.

iii. **The joint-space inertia matrix**, necessary for maps the joint acceleration to the joint actuators. It is used in analysis, feedback control and in some integral part of forward dynamics formulation.

iv. **The operational-space inertia matrix**, this algorithm maps the task accelerations to task actuator in Cartesian space. It is required for control of end-effector.

The field of dynamic robot manipulator has a wide literature that published in professional journals and established textbooks [1, 6, 14].

Several different methods are available to compute robot manipulator dynamic equations. These methods include the Newton-Euler (N-E) methodology, the Lagrange-Euler (L-E) method, and Kane's methodology [1].

The Newton-Euler methodology is based on Newton's second law and several different researchers are signifying to develop this method [1, 14]. This equation can be described the behavior of a robot manipulator link-by-link and joint-by-joint from base to endeffector, called forward recursion and transfer the essential information from end-effector to base frame, called backward recursive. The literature on Euler-Lagrange's is vast but a good starting point to learn about it is in[1]. Calculate the dynamic equation robot manipulator using E-L method is easier because this equation is derivation of nonlinear coupled and quadratic differential equations. The Kane's method was introduced in 1961 by Professor Thomas Kane[1, 6]. This method used to calculate the dynamic equation of motion without any differentiation between kinetic and potential energy functions. The equation of a multi degrees of freedom (DOF) robot manipulator is calculated by the following equation[6]:

$$\boldsymbol{M(q)\ddot{q} + N(q,\dot{q}) = \tau} \quad \textbf{(23)}$$

Where τ is $n \times 1$ vector of actuation torque, M (q) is $n \times n$ symmetric and positive define inertia matrix, $N(q,\dot{q})$ is the vector of nonlinearity term, and q is $n \times 1$ position vector. In equation 2.8 if vector of nonlinearity term derive as Centrifugal, Coriolis and Gravity terms, as a result robot manipulator dynamic equation can also be written as [80]:

$$\boldsymbol{N(q,\dot{q}) = V(q,\dot{q}) + G(q)} \quad \textbf{(24)}$$

$$\boldsymbol{V(q,\dot{q}) = B(q)[\dot{q}\ \dot{q}] + C(q)[\dot{q}]^2} \quad \textbf{(25)}$$

$$\boldsymbol{\tau = M(q)\ddot{q} + B(q)[\dot{q}\ \dot{q}] + C(q)[\dot{q}]^2 + G(q)} \quad \textbf{(26)}$$

Where,

$B(q)$ is matrix of coriolis torques, $C(q)$ is matrix of centrifugal torque, $[\dot{q}\ \dot{q}]$ is vector of joint velocity that it can give by: $[\dot{q}_1.\dot{q}_2, \dot{q}_1.\dot{q}_3, \ldots., \dot{q}_1.\dot{q}_n, \dot{q}_2.\dot{q}_3, \ldots..]^T$, and $[\dot{q}]^2$ is vector, that it can given by: $[\dot{q}_1{}^2, \dot{q}_2{}^2, \dot{q}_3{}^2, \ldots.]^T$.
In robot manipulator dynamic part the inputs are torques and the outputs are actual displacements, as a result in (2.11) it can be written as [1, 6, 80-81];

$$\boldsymbol{\ddot{q} = M^{-1}(q).\{\tau - N(q,\dot{q})\}} \quad \textbf{(27)}$$

To implementation (27) the first step is implement the kinetic energy matrix (M) parameters by used of Lagrange's formulation. The second step is implementing the Coriolis and Centrifugal matrix which they can calculate by partial derivatives of kinetic energy. The last step to implement the dynamic equation of robot manipulator is to find the gravity vector by performing the summation of Lagrange's formulation. The kinetic energy equation (M) is a $n \times n$ symmetric matrix that can be calculated by the following equation;

$$\boldsymbol{M(\theta) = m_1 J_{v1}^T J_{v1} + J_{\omega 1}^{TC1} I_1 J_{\omega 1} + m_2 J_{v2}^T J_{v2} + J_{\omega 2}^{TC2} I_2 J_{\omega 2} + m_3 J_{v3}^T J_{v3} + J_{\omega 3}^{TC3} I_3 J_{\omega 3} + m_4 J_{v4}^T J_{v4} + J_{\omega 4}^{TC4} I_4 J_{\omega 4} + m_5 J_{v5}^T J_{v5} + J_{\omega 5}^{TC5} I_5 J_{\omega 5} + m_6 J_{v6}^T J_{v6} + J_{\omega 6}^{TC6} I_6 J_{\omega 6}} \quad \textbf{(28)}$$

As mentioned above the kinetic energy matrix in n DOF is a $n \times n$ matrix that can be calculated by the following matrix [1, 6]

$$\boldsymbol{M(q)} = \begin{bmatrix} \boldsymbol{M_{11}} & \boldsymbol{M_{12}} & \ldots & \ldots. & \ldots.. & \boldsymbol{M_{1n}} \\ \boldsymbol{M_{21}} & \ldots & \ldots & \ldots. & \ldots.. & \boldsymbol{M_{2n}} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \boldsymbol{M_{n1}} & \ldots & \ldots & \ldots. & \ldots & \boldsymbol{M_{nn}} \end{bmatrix} \quad \textbf{(29)}$$

The Coriolis matrix (B) is a $n \times \frac{n(n-1)}{2}$ matrix which calculated as follows;

$$B(q) = \begin{bmatrix} b_{112} & b_{113} & \dots & b_{11n} & b_{123} & \dots & b_{12n} & \dots & \dots & b_{1.n-1.n} \\ b_{212} & \dots & \dots & b_{21n} & b_{223} & \dots & \dots & \dots & \dots & b_{2.n-1.n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_{n.1.2} & \dots & \dots & b_{n.1.n} & \dots & \dots & \dots & \dots & \dots & b_{n.n-1.n} \end{bmatrix} \quad (30)$$

and the Centrifugal matrix (C) is a $n \times n$ matrix;

$$C(q) = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{bmatrix} \quad (31)$$

And last the Gravity vector (G) is a $n \times 1$ vector;

$$G(q) = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \quad (32)$$

Dynamics of PUMA560 Robot Manipulator

To position control of robot manipulator, the second three axes are locked the dynamic equation of PUMA robot manipulator is given by [77-80];

$$M(\ddot{\theta}) \begin{bmatrix} \ddot{\theta}1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \mathrm{B}(\theta) \begin{bmatrix} \dot{\theta}_1\dot{\theta}_2 \\ \dot{\theta}_1\dot{\theta}_3 \\ \dot{\theta}_2\dot{\theta}_3 \end{bmatrix} + C(\theta) \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \end{bmatrix} + G(\theta) = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (33)$$

Where

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 & 0 \\ M_{21} & M_{22} & M_{23} & 0 & 0 & 0 \\ M_{31} & M_{32} & M_{33} & 0 & M_{35} & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix} \quad (34)$$

M is computed as

$$M_{11} = I_{m1} + I_1 + I_3 \times \cos(\theta_2)\cos(\theta_2) + I_7\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + I_{10}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{11}\sin(\theta_2)\cos(\theta_2) + I_{21}\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + 2 + [I_5\cos(\theta_2)\sin(\theta_2+\theta_3) + I_{12}\cos(\theta_2)\cos(\theta_2+\theta_3) + I_{15}\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + I_{16}\cos(\theta_2)\sin(\theta_2+\theta_3) + I_{22}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) \quad (35)$$

$$M_{12} = I_4\sin(\theta_2) + I_8\cos(\theta_2+\theta_3) + I_9 cos(\theta_2) + I_{13}\sin(\theta_2+\theta_3) - I_{18}\cos(\theta_2+\theta_3) \quad (36)$$

$$M_{13} = I_8\cos(\theta_2+\theta_3) + I_{13}\sin(\theta_2+\theta_3) - I_{18}\cos(\theta_2+\theta_3) \quad (37)$$

$$M_{22} = I_{m2} + I_2 + I_6 + 2[I_5\sin(\theta_3) + I_{12}\cos(\theta_2) + I_{15} + I_{16}\sin(\theta_3) \quad (38)$$

$$M_{23} = I_5\sin(\theta_3) + I_6 + I_{12}\cos(\theta_3) + I_{16}\sin(\theta_3) + 2I_{15} \quad (39)$$

$$M_{33} = I_{m3} + I_6 + 2I_{15} \quad (40)$$

$$M_{35} = I_{15} + I_{17} \quad (41)$$

$$M_{44} = I_{m4} + I_{14} \quad (42)$$

$$M_{55} = I_{m5} + I_{17} \quad (43)$$

$$M_{66} = I_{m6} + I_{23} \quad (44)$$

$$M_{21} = M_{12}, M_{31} = M_{13} \; and \; M_{32} = M_{23} \quad (45)$$

and Corilios (B) matrix is calculated as the following

$$B(q) = \begin{bmatrix} b_{112} & b_{113} & 0 & b_{115} & 0 & b_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{214} & 0 & 0 & b_{223} & 0 & b_{225} & 0 & 0 & b_{235} & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{314} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{412} & b_{412} & 0 & b_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{514} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (46)$$

Where,

$$b_{112} = 2\big[-I_3\sin(\theta_2)\cos(\theta_2) + I_5\cos(\theta_2+\theta_2+\theta_3) + I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{12}\sin(\theta_2+\theta_2+\theta_3) - I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{16}\cos(\theta_2+\theta_2+\theta_3) + I_{21}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{22}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big)\big] + I_{10}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) + I_{11}\big(1-2\sin(\theta_2)\sin(\theta_2)\big) \quad (47)$$

$$b_{113} = 2\big[I_5\cos(\theta_2)\cos(\theta_2+\theta_3) + I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{12}\cos(\theta_2)\sin(\theta_2+\theta_2) + I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{16}\cos(\theta_2)\cos(\theta_2+\theta_3) + I_{21}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{22}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big)\big] + I_{10}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) \quad (48)$$

$$b_{115} = 2[-\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{16}\cos(\theta_2)\cos(\theta_2+\theta_3) + I_{22}\cos(\theta_2+\theta_3)\cos(\theta_2+\theta_3)\,] \quad (49)$$

$$b_{123} = 2[-I_8\sin(\theta_2+\theta_3) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3)\,] \quad (50)$$

$$b_{214} = I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) + 2I_{20}\sin(\theta_2+\theta_3)(1-0.5) \quad (51)$$

$$b_{223} = 2[-I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3)\,] \quad (52)$$

$$b_{235} = 2[I_{16}\cos(\theta_3) + I_{22}\,] \quad (53)$$

$$b_{314} = 2[I_{20}\sin(\theta_2+\theta_3)(1-0.5)] + I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) \quad (54)$$

$$b_{412} = b_{214} = -[I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) + 2I_{20}\sin(\theta_2+\theta_3)(1-0.5)] \quad (55)$$

$$b_{413} = -b_{314} = -2[I_{20}\sin(\theta_2+\theta_3)(1-0.5)] + I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) \quad (56)$$

$$b_{415} = -I_{20}\sin(\theta_2+\theta_3) - I_{17}\sin(\theta_2+\theta_3) \quad (57)$$

$$b_{514} = -b_{415} = I_{20}\sin(\theta_2+\theta_3) + I_{17}\sin(\theta_2+\theta_3) \quad (58)$$

consequently coriolis matrix is shown as bellows;

$$B(q).\dot{q}.\dot{q} = \begin{bmatrix} b_{112}.\dot{q}_1\dot{q}_2 + b_{113}.\dot{q}_1\dot{q}_3 + 0 + b_{123}.\dot{q}_2\dot{q}_3 \\ 0 + b_{223}.\dot{q}_2\dot{q}_3 + 0 + 0 \\ 0 \\ b_{412}.\dot{q}_1\dot{q}_2 + b_{413}.\dot{q}_1\dot{q}_3 + 0 \\ 0 \\ 0 \end{bmatrix} \quad (59)$$

Moreover Centrifugal (C) matrix is demonstrated as

$$C(q) = \begin{bmatrix} 0 & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & 0 & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ C_{51} & C_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (60)$$

Where,

$$c_{12} = I_4\cos(\theta_2) - I_8\sin(\theta_2+\theta_3) - I_9\sin(\theta_2) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3) \quad (61)$$

$$c_{13} = 0.5b_{123} = -I_8\sin(\theta_2+\theta_3) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3) \quad (62)$$

$$c_{21} = -0.5b_{112} = I_3\sin(\theta_2)\cos(\theta_2) - I_5\cos(\theta_2+\theta_2+\theta_3) - I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{12}\sin(\theta_2+\theta_2+\theta_3) + I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2+\theta_2+\theta_3) - I_{21}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{22}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) - 0.5I_{10}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) - 0.5I_{11}\big(1-2\sin(\theta_2)\sin(\theta_2)\big) \quad (63)$$

$$c_{22} = 0.5b_{223} = -I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3) \quad (64)$$

$$c_{23} = -0.5b_{113} = -I_5\cos(\theta_2)\cos(\theta_2+\theta_3) - I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{12}\cos(\theta_2)\sin(\theta_2+\theta_2) - I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2)\cos(\theta_2+\theta_3) - I_{21}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{22}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) - 0.5I_{10}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) \quad (65)$$

$$c_{31} = -c_{23} = I_{12}\sin(\theta_3) - I_5\cos(\theta_3) - I_{16}\cos(\theta_3) \quad (66)$$

$$c_{32} = -0.5b_{115} = s\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2)\cos(\theta_2+\theta_3) - I_{22}\cos(\theta_2+\theta_3)\cos(\theta_2+\theta_3) \quad (67)$$

$$c_{52} = -0.5b_{225} = -I_{16}\cos(\theta_3) - I_{22} \quad (68)$$

In this research $q_4 = q_5 = q_6 = 0$, as a result

$$C(q).\dot{q}^2 = \begin{bmatrix} c_{112}.\dot{q}_2^2 + c_{13}.\dot{q}_3^2 \\ c_{21}.\dot{q}_1^2 + c_{23}.\dot{q}_3^2 \\ c_{13}.\dot{q}_1^2 + c_{32}.\dot{q}_2^2 \\ 0 \\ c_{51}.\dot{q}_1^2 + c_{52}.\dot{q}_2^2 \\ 0 \end{bmatrix} \quad (69)$$

Gravity (G) Matrix can be written as

$$G(q) = \begin{bmatrix} 0 \\ g_2 \\ g_3 \\ 0 \\ g_5 \\ 0 \end{bmatrix} \quad (70)$$

Where,

$$G_2 = g_1\cos(\theta_2) + g_2\sin(\theta_2 + \theta_3) + g_3\sin(\theta_2) + g_4\cos(\theta_2 + \theta_3) + g_5\sin(\theta_2 + \theta_3) \quad (71)$$

$$G_3 = g_2\sin(\theta_2 + \theta_3) + g_4\cos(\theta_2 + \theta_3) + g_5\sin(\theta_2 + \theta_3) \quad (72)$$

$$G_5 = g_5\sin(\theta_2 + \theta_3) \quad (73)$$

Suppose $\ddot{q}$ is written as follows

$$\ddot{q} = M^{-1}(q).\{\tau - [B(q)\dot{q}\dot{q} + C(q)\dot{q}^2 + g(q)]\} \quad (74)$$

and K is introduced as

$$K = \{\tau - [B(q)\dot{q}\dot{q} + C(q)\dot{q}^2 + g(q)]\} \quad (75)$$

$\ddot{q}$ can be written as

$$\ddot{q} = M^{-1}(q).K \quad (76)$$

Therefore K for PUMA robot manipulator is calculated by the following equations

$$K_1 = \tau_1 - [\, b_{112}\dot{q}_1\dot{q}_2 + b_{113}\dot{q}_1\dot{q}_3 + 0 + b_{123}\dot{q}_2\dot{q}_3] - [\, C_{12}\dot{q}_2{}^2 + C_{13}\dot{q}_3{}^2] - g_1 \quad (77)$$

$$K_2 = \tau_2 - [\, b_{223}\dot{q}_2\dot{q}_3] - [\, C_{21}\dot{q}_1{}^2 + C_{23}\dot{q}_3{}^2] - g_2 \quad (78)$$

$$K_3 = \tau_3 - \left[C_{31}\dot{q}_1{}^2 + C_{32}\dot{q}_2{}^2\right] - g_3 \quad (79)$$

$$K_4 = \tau_4 - [\, b_{412}\dot{q}_1\dot{q}_2 + b_{413}\dot{q}_1\dot{q}_3] - g_4 \quad (80)$$

$$K_5 = \tau_5 - [\, C_{51}\dot{q}_1{}^2 + C_{52}\dot{q}_2{}^2] - g_5 \quad (81)$$

$$K_6 = \tau_6 \quad (82)$$

An information about inertial constant and gravitational constant are shown in Tables 6 and 7 based on the studies carried out by Armstrong [80] and Corke and Armstrong [81].

$\mathrm{I}_1 = 1.43 \pm 0.05$	$I_2 = 1.75 \pm 0.07$
$I_3 = 1.38 \pm 0.05$	$I_4 = 0.69 \pm 0.02$
$I_5 = 0.372 \pm 0.031$	$\mathrm{I}_6 = 0.333 \pm 0.016$
$I_7 = 0.298 \pm 0.029$	$\mathrm{I}_8 = -0.134 \pm 0.014$
$I_9 = 0.0238 \pm 0.012$	$I_{10} = -0.0213 \pm 0.0022$
$I_{11} = -0.0142 \pm 0.0070$	$I_{12} = -0.011 \pm 0.0011$
$I_{13} = -0.00379 \pm 0.0009$	$I_{14} = 0.00164 \pm 0.000070$
$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$\mathrm{I}_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$\mathrm{I}_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.000015$
$\mathrm{I}_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

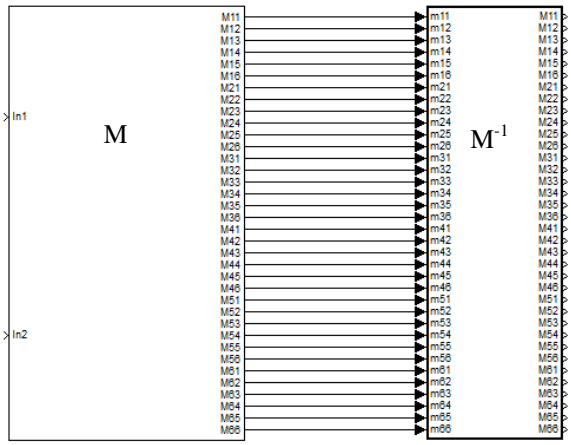
TABLE 6: Inertial constant reference (*Kg.m*^{2})

$g_1 = -37.2 \pm 0.5$	$g_2 = -8.44 \pm 0.20$
$g_3 = 1.02 \pm 0.50$	$g_4 = 0.249 \pm 0.025$
$g_5 = -0.0282 \pm 0.0056$	

TABLE 7: Gravitational constant (*N.m*)

Formulation and implementation of Matrix Entries: As mentioned before, every matrix entry has its own formula. Below you can find them:

Finding inverse matrix for kinetic energy: Kinetic energy has illustrated by **M**. The kinetic energy matrix is a **6 x 6** matrix [10]. In MATLAB, the command ***"inv(matrix)"*** will inverse a n x n matrix .what is more, the results must be taken into a separate matrix in order to be used in Dynamic equation. Both **M** and $\mathbf{M}^{-1}$ must be implemented in a separate *Matlab Embedded Function*. The outputs of **M** will be linked to inputs of $\mathbf{M}^{-1}$. The block diagram will be shown as Figure 6.



Coriolis Effect Matrix: The Coriolis Effect is a **15 x 6** matrix. The block diagram of $B(q).[\dot{q}\dot{q}]$ could be shown as Figure 7. We set $q_4 = q_5 = q_6 = 0$

:

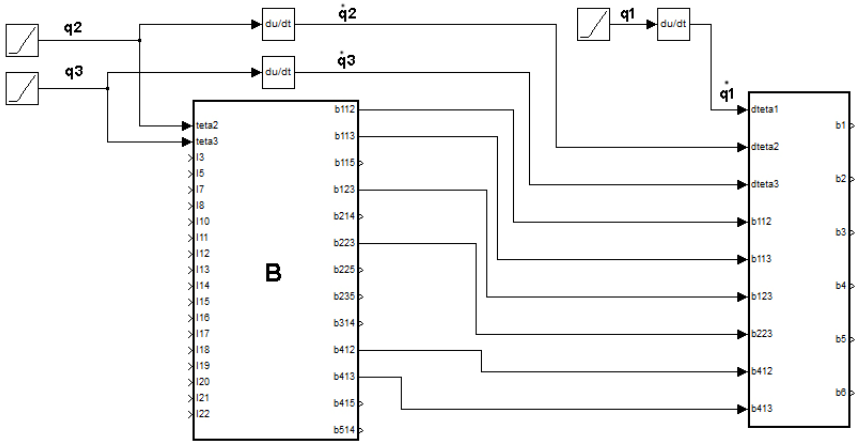


FIGURE7: Block diagram for coriolis effect

Centrifugal Force Matrix

Centrifugal force has illustrated as **C** and is a **6 x 6** matrix. In PUMA 560, the centrifugal force is a **6 x 6** matrix.after implementing centrifugal force in a block diagram, its time to implement $C(q)\dot{q}^2$. We set $q_4 = q_5 = q_6 = 0$ the block diagram for this part could be illustrated as Figure 8.

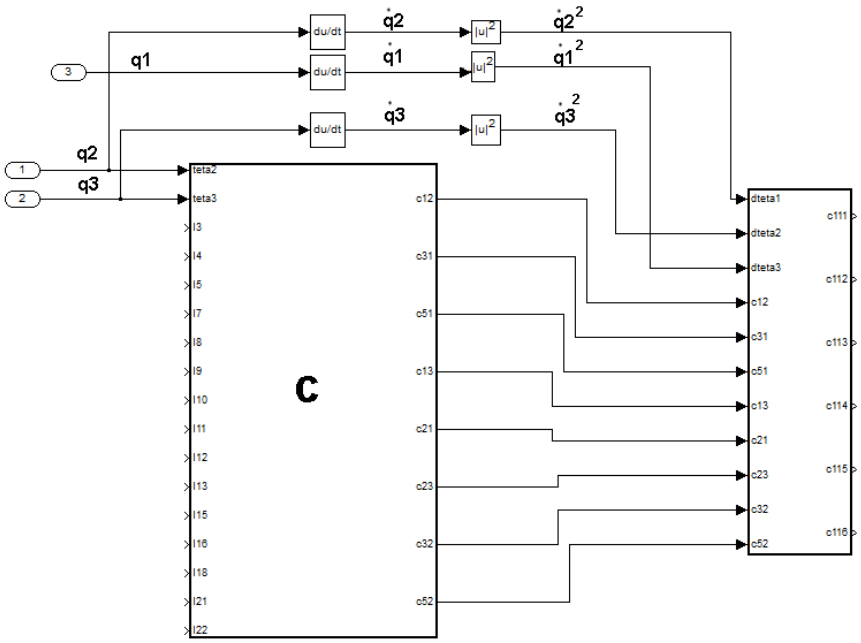


FIGURE8: Block diagram for Centrifugal force

Gravity Matrix

Gravity is shown as **g** and is a **6 x 1** matrix. In PUMA 560, the Gravity is a **6 x 1** matrix. The block diagram is presented as Figure 9.

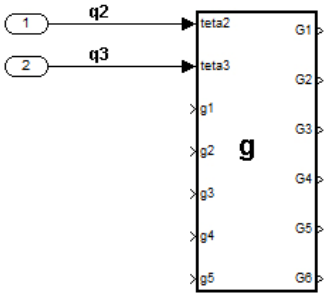


FIGURE9: Block diagram for Gravity

Implement Dynamic Formula in SIMULINK

$\boldsymbol{I}$ is summation between Coriolis Effect Matrix, Centrifugal force Matrix and Gravity matrix. $\boldsymbol{K}$ could be find in equation (77). Figure 10 is shown I and K implementation.

The block diagram I can be made as below:

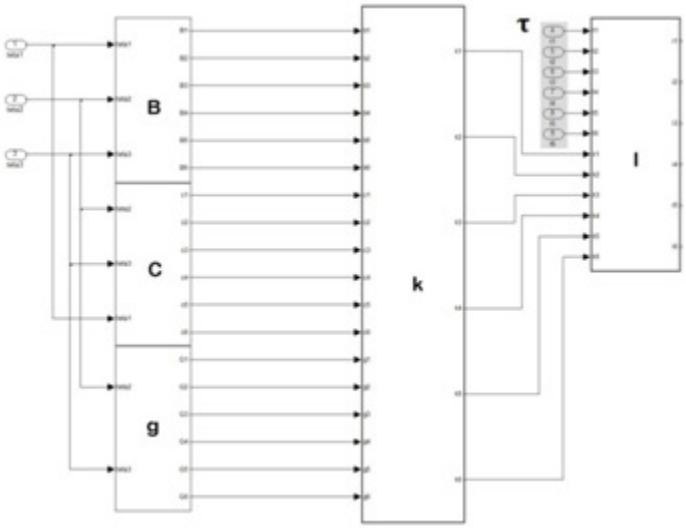


FIGURE 10: Block diagram for K and I

After masking I the block diagram will be making as Figure 11.

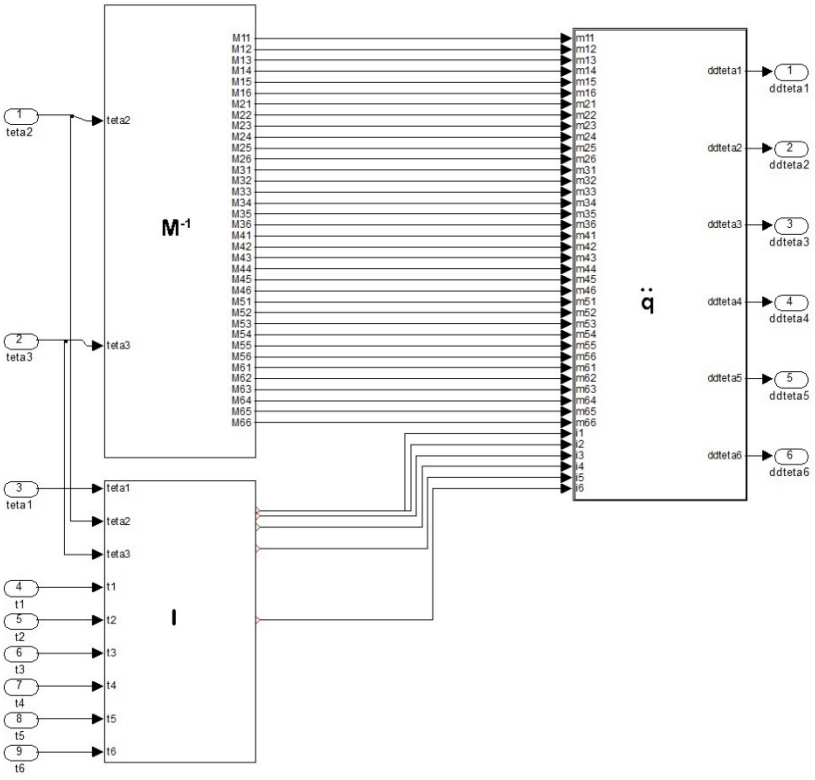


FIGURE 11: Block diagram for q̈

After implementing, block diagram should be masked. The block diagram shown in Figure 11, counts $\ddot{q}$.to count q, block diagram below must be implemented as Figure 12.

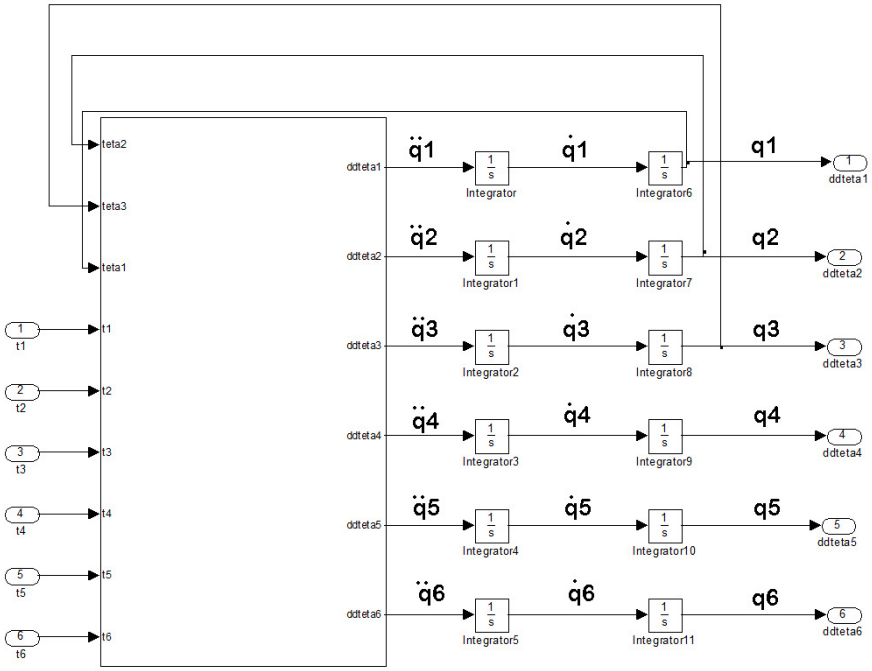


FIGURE 12: Block diagram for q

Now, everything should be masked and constants shown in Table1 and Table2 must be defined. At the end the final block diagram could be illustrated as Figure 13.

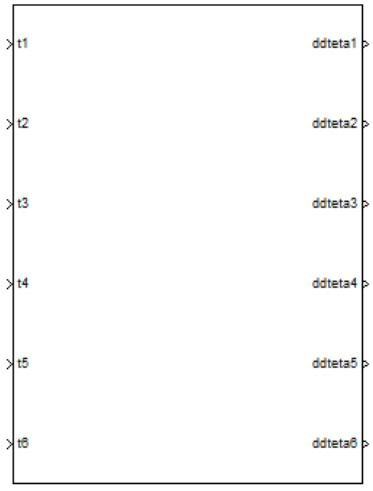


FIGURE 13: final block diagram for Dynamic model

3. CONTROL: SLIDING MODE CONTROLLER ANALYSIS, MODELLING AND IMPLEMENTATION ON PUMA 560 ROBOT MANIPULATOR

In this section formulations of sliding mode controller for robot manipulator is presented based on [1, 6]. Consider a nonlinear single input dynamic system is defined by [6]:

$$x^{(n)} = f(\vec{x}) + b(\vec{x})u \tag{83}$$

Where u is the vector of control input, $x^{(n)}$ is the n^{th} derivation of x, $x = [x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}]^T$ is the state vector, $f(x)$ is unknown or uncertainty, and $b(x)$ is of known *sign* function. The main goal to design this controller is train to the desired state; $x_d = [x_d, \dot{x}_d, \ddot{x}_d, \dots, {x_d}^{(n-1)}]^T$, and trucking error vector is defined by [6]:

$$\tilde{x} = x - x_d = [\tilde{x}, \dots, \tilde{x}^{(n-1)}]^T \tag{84}$$

A time-varying sliding surface $s(x,t)$ in the state space R^n is given by [6]:

$$s(x,t) = (\frac{d}{dt} + \lambda)^{n-1}\, \tilde{x} = 0 \tag{85}$$

where λ is the positive constant. To further penalize tracking error, integral part can be used in sliding surface part as follows [6]:

$$s(x,t) = (\frac{d}{dt} + \lambda)^{n-1} \left(\int_0^t \tilde{x}\, dt \right) = 0 \tag{86}$$

The main target in this methodology is kept the sliding surface slope $s(x,t)$ near to the zero. Therefore, one of the common strategies is to find input U outside of $s(x,t)$ [6].

$$\frac{1}{2}\frac{d}{dt}s^2(x,t) \le -\zeta |s(x,t)| \tag{87}$$

where **ζ** is positive constant.

$$\text{If } \mathbf{S(0)>0} \rightarrow \frac{\mathrm{d}}{\mathrm{dt}}\mathrm{S(t)} \le -\zeta \tag{88}$$

To eliminate the derivative term, it is used an integral term from t=0 to t=t_{reach}

$$\int_{t=0}^{t=t_{reach}} \frac{d}{dt} S(t) \le -\int_{t=0}^{t=t_{reach}} \eta \rightarrow S(t_{reach}) - S(0) \le -\zeta(t_{reach} - 0) \tag{89}$$

Where t_{reach} is the time that trajectories reach to the sliding surface so, suppose S($t_{reach} = 0$) defined as

$$0 - S(0) \le -\eta(t_{reach}) \rightarrow t_{reach} \le \frac{S(0)}{\zeta} \tag{90}$$

and

$$if\ S(0) < 0 \rightarrow 0 - S(0) \le -\eta(t_{reach}) \rightarrow S(0) \le -\zeta(t_{reach}) \rightarrow t_{reach} \le \frac{|S(0)|}{\eta} \tag{91}$$

Equation (91) guarantees time to reach the sliding surface is smaller than $\frac{|S(0)|}{\zeta}$ since the trajectories are outside of $S(t)$.

$$if\ S_{t_{reach}} = S(0) \rightarrow error(x - x_d) = 0 \tag{92}$$

suppose S is defined as

$$s(x,t) = (\frac{d}{dt} + \lambda) \quad \tilde{x} = (\dot{\mathrm{x}} - \dot{\mathrm{x}}_{\mathrm{d}}) + \lambda(\mathrm{x} - \mathrm{x}_{\mathrm{d}}) \tag{93}$$

The derivation of S, namely, $\dot{S}$ can be calculated as the following;

$$\dot{S} = (\ddot{\mathrm{x}} - \ddot{\mathrm{x}}_{\mathrm{d}}) + \lambda(\dot{\mathrm{x}} - \dot{\mathrm{x}}_{\mathrm{d}}) \tag{94}$$

suppose the second order system is defined as;

$$\ddot{x} = f + u \rightarrow \dot{S} = f + U - \ddot{x}_d + \lambda(\dot{x} - \dot{x}_d) \quad (95)$$

Where f is the dynamic uncertain, and also since $S = 0 \; and \; \dot{S} = 0$, to have the best approximation ,$\hat{U}$ is defined as

$$\hat{U} = -\hat{f} + \ddot{x}_d - \lambda(\dot{x} - \dot{x}_d) \quad (96)$$

A simple solution to get the sliding condition when the dynamic parameters have uncertainty is the switching control law:

$$U_{dis} = \hat{U} - K(\vec{x}, t) \cdot \mathbf{sgn}(s) \quad (97)$$

where the switching function $\mathbf{sgn}(S)$ is defined as [1, 6]

$$sgn(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \\ 0 & s = 0 \end{cases} \quad (98)$$

and the $K(\vec{x}, t)$ is the positive constant. Suppose by (90) the following equation can be written as,

$$\frac{1}{2}\frac{d}{dt}s^2(x,t) = \dot{S} \cdot S = \left[f - \hat{f} - K\mathbf{sgn}(s)\right] \cdot S = \left(f - \hat{f}\right) \cdot S - K|S| \quad (99)$$

and if the equation (94) instead of (93) the sliding surface can be calculated as

$$s(x,t) = (\frac{d}{dt} + \lambda)^2 \left(\int_0^t \tilde{x}\, dt\right) = (\dot{x} - \dot{x}_d) + 2\lambda(\dot{x} - \dot{x}_d) - \lambda^2(x - x_d) \quad (100)$$

in this method the approximation of U is computed as [6]

$$\hat{U} = -\hat{f} + \ddot{x}_d - 2\lambda(\dot{x} - \dot{x}_d) + \lambda^2(x - x_d) \quad (101)$$

Based on above discussion, the sliding mode control law for a multi degrees of freedom robot manipulator is written as [1, 6]:

$$\tau = \tau_{eq} + \tau_{dis} \quad (102)$$

Where, the model-based component τ_{eq} is the nominal dynamics of systems and τ_{eq} for first 3 DOF PUMA robot manipulator can be calculate as follows [1]:

$$\tau_{eq} = \left[M^{-1}(B + C + G) + \dot{S}\right]M \quad (103)$$

and τ_{dis} is computed as [1];

$$\tau_{dis} = K \cdot \mathbf{sgn}(S) \quad (104)$$

by replace the formulation (104) in (102) the control output can be written as;

$$\tau = \tau_{eq} + K.\mathbf{sgn}(S) \quad (105)$$

Figure 14 shows the position classical sliding mode control for PUMA 560 robot manipulator. By (105) and (103) the sliding mode control of PUMA 560 robot manipulator is calculated as;

$$\tau = \left[M^{-1}(B + C + G) + \dot{S}\right]M + K \cdot \mathbf{sgn}(S) \quad (106)$$

where $S = \lambda e + \dot{e}$ in PD-SMC and $S = \lambda e + \dot{e} + (\frac{\lambda}{2})^2 \sum e$ in PID-SMC.

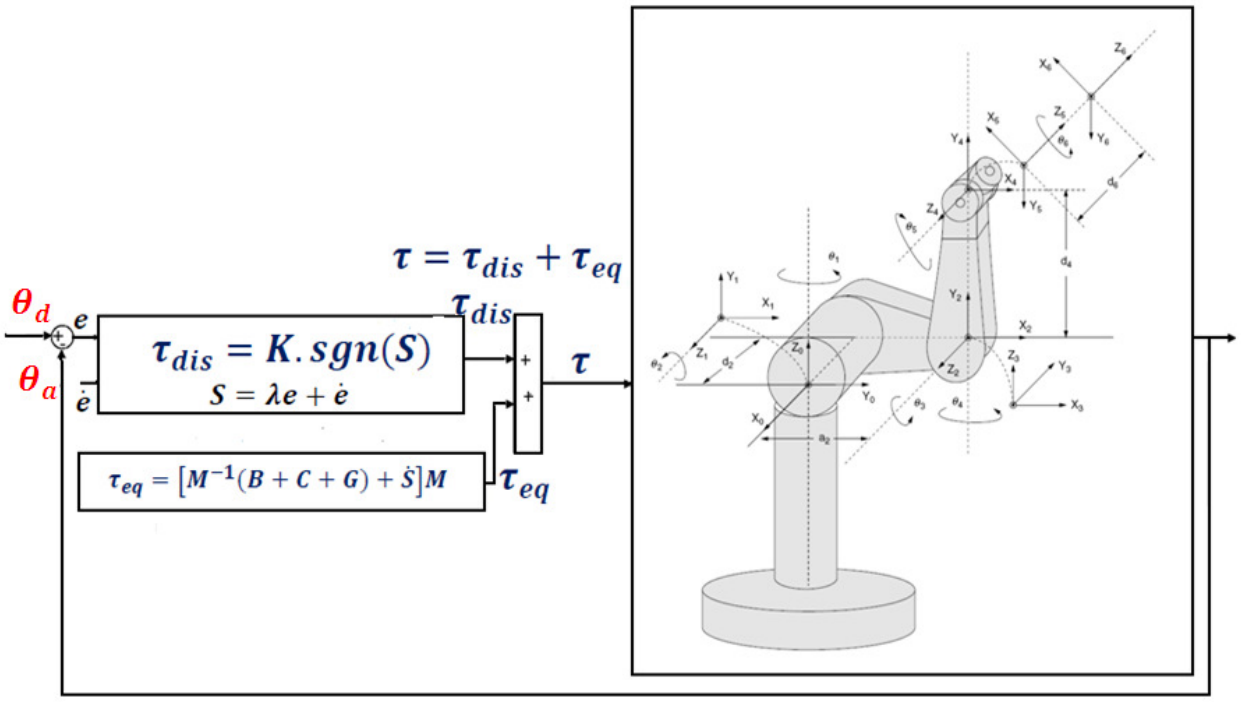


FIGURE 14: Block diagram of pure sliding mode controller with switching function

Implemented Sliding Mode Controller

The main object is implementation of controller block. According to T_{dis} equation which is T_{dis}=K * sign(s), this part will be created like figure 15. As it is obvious, the parameter e is the difference of actual and desired values and $\dot{e}$ is the change of error. Luanda (l1) and k are coefficients which are affected on discontinuous component and the saturation function accomplish the switching progress. A sample of discontinuous torque for one joint is like figure 15.

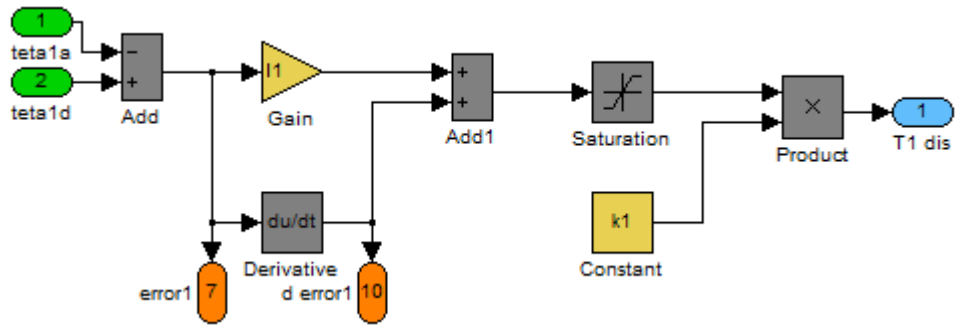


FIGURE 15: Discontinuous part of torque for one joint variable

As it is seen in figure 15 the error value and the change of error were chosen to exhibit in measurement center. In this block by changing gain and coefficient values, the best control system will be applied. In the second step according to torque formulation in SMC mode, the equivalent part should constructed. Based on equivalent formulation $\boldsymbol{\tau_{eq} = [M^{-1}(B + C + G) + \dot{S}]M}$ all constructed blocks just connect to each other as Figure 16. In this figure the N $(q, \dot{q})$ is the dynamic parameters block (i.e., A set of Coriolis, Centrifugal and

Gravity blocks) and the derivative of S is apparent. Just by multiplication and summation, the output which is equivalent torque will be obtained.

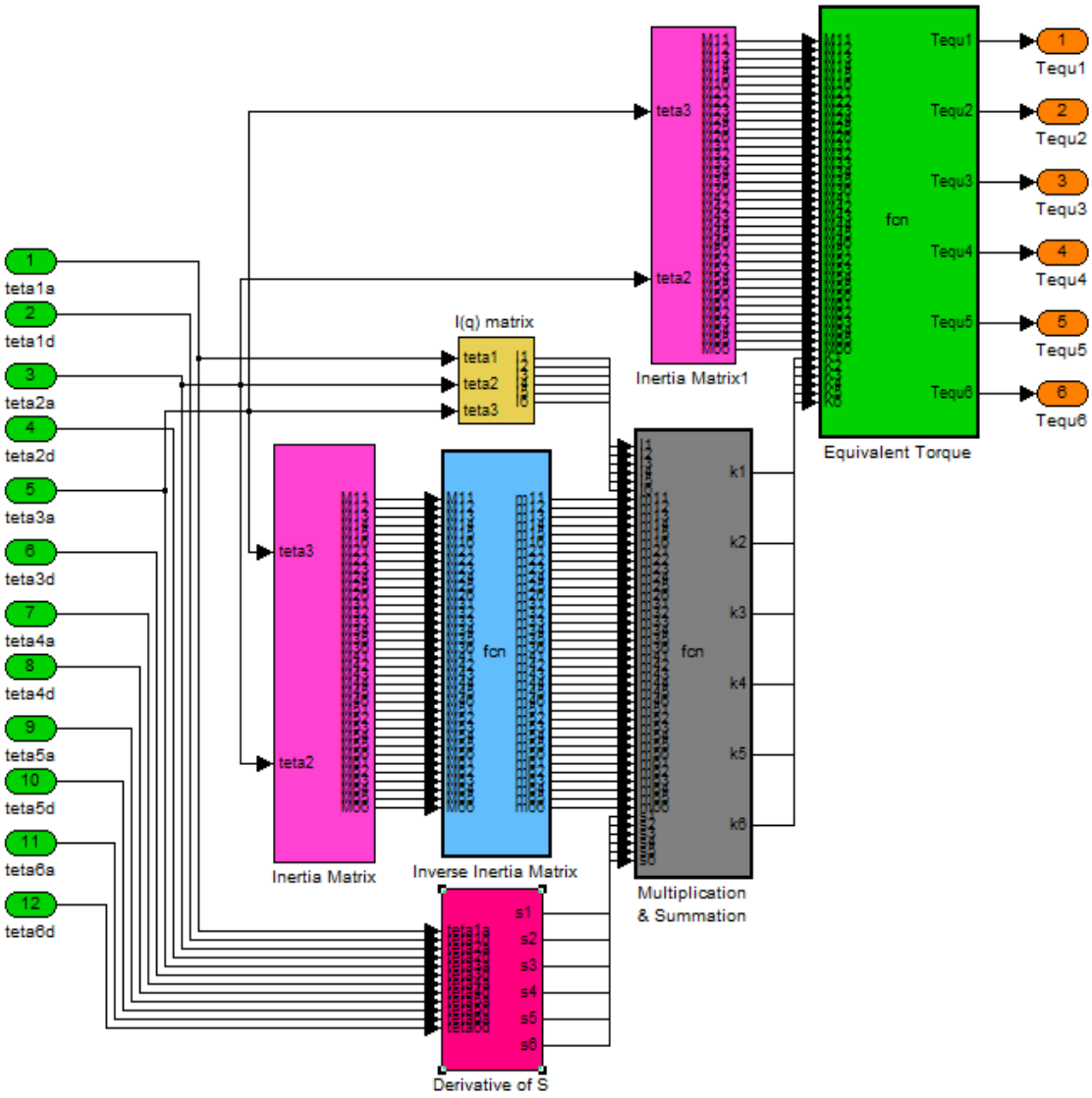


FIGURE 16: the equivalent part of torque with required blocks

The inputs are thetas and the final outputs are equivalent torque values. The relations between other blocks are just multiplication and summation as mentioned in torque equation. The next phase is calculation of the summation of equivalent part and discontinuous part which make the total torque value. This procedure is depicted in Figure 17.

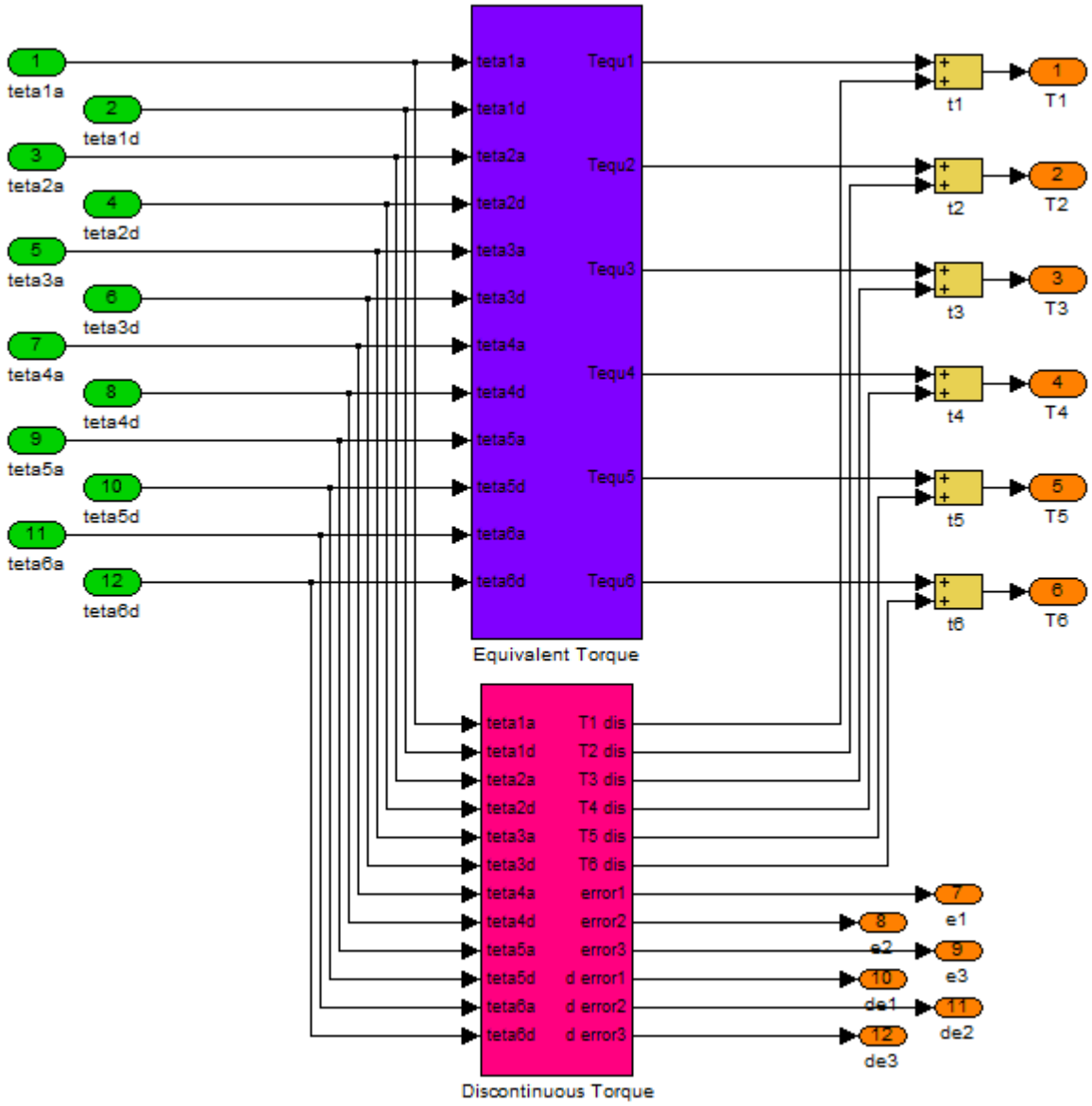


FIGURE 17: the total value of torque which is summation of equivalent & discontinuous blocks

In the next step transform our subsystems into a general system to form controller block and the outputs will be connected to the plant, in order to execute controlling process. Then, trigger the main inputs with power supply to check validity and performance. In Figure 18 Dynamics, Kinematics, Controller and the measurement center blocks are shown.

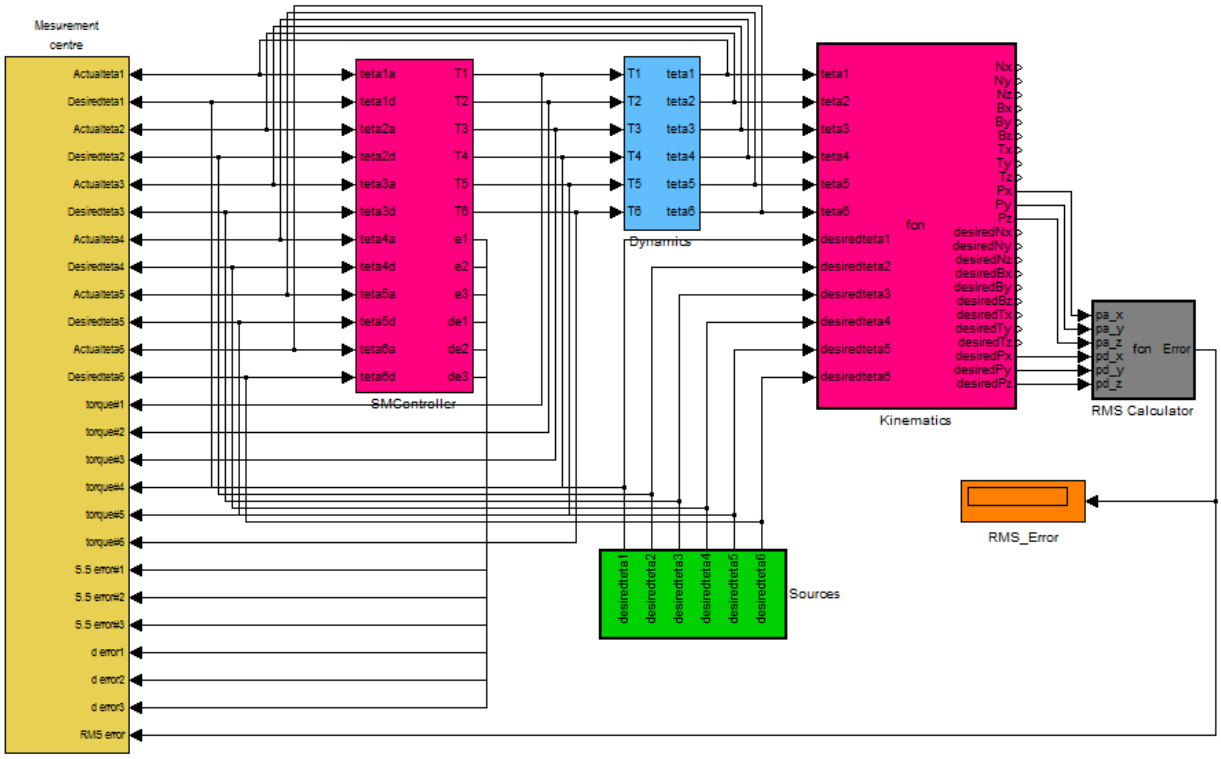


FIGURE 18: Measurement center, Controller, Dynamics and Kinematics Blocks

4. RESULTS

PD-sliding mode controller (PD-SMC) and PID-sliding mode controller (PID-SMC) were tested to Step and Ramp responses. In this simulation the first, second, and third joints are moved from home to final position without and with external disturbance. The simulation was implemented in MATLAB/SIMULINK environment. It is noted that, these systems are tested by band limited white noise with a predefined 40% of relative to the input signal amplitude which the sample time is equal to 0.1. This type of noise is used to external disturbance in continuous and hybrid systems.

Tracking Performances

Figures 19 and 20 show the tracking performance in PD-SMC and PID SMC without disturbance for Step and Ramp trajectories. The best possible coefficients in Step PID-SMC are; $K_p = K_v = K_i = 30,\ \emptyset_1 = \emptyset_2 = \emptyset_3 = 0.1, and\ \lambda_1 = 3, \lambda_2 = 6, \lambda_3 = 6$ and in Ramp PID-SMC are; $K_p = K_v = K_i = 5,\ \emptyset_1 = \emptyset_2 = \emptyset_3 = 0.1, and\ \lambda_1 = 15, \lambda_2 = 15, \lambda_3 = 10$ as well as similarly in Step PD-SMC are; $K_p = K_v = 10,\ \emptyset_1 = \emptyset_2 = \emptyset_3 = 0.1, and\ \lambda_1 = 1, \lambda_2 = 6, \lambda_3 = 8$; and at last in Ramp PD-SMC are; $K_p = K_v = 5,\ \emptyset_1 = \emptyset_2 = \emptyset_3 = 0.1, and\ \lambda_1 = 15, \lambda_2 = 15, \lambda_3 = 10$. From the simulation for first, second, and third links, different controller gains have the different result. Tuning parameters of PID-SMC and PD-SMC for two type trajectories in PUMA 560 robot manipulator are shown in Table 8 to 11.

	λ_1	k_1	ϕ_1	λ_2	k_2	ϕ_2	λ_3	k_3	ϕ_3	SS error$_1$	SS error$_2$	SS error$_3$	RMS error
data 1	3	30	0.1	6	30	0.1	6	30	0.1	0	0	-5.3e-15	0
data 2	30	30	0.1	60	30	0.1	60	30	0.1	-5.17	14.27	-1.142	0.05
data 3	3	300	0.1	6	300	0.1	6	300	0.1	2.28	0.97	0.076	0.08

TABLE 8: Tuning parameters of Step PID-SMC

	λ_1	k_1	ϕ_1	λ_2	k_2	ϕ_2	λ_3	k_3	ϕ_3	SS error$_1$	SS error$_2$	SS error$_3$	RMS error
data1	15	5	0.1	15	5	0.1	10	5	0.1	4.6e-12	-3.97e-12	-3.87e-12	0.0002441
data2	150	5	0.1	150	5	0.1	100	5	0.1	1005	1108	436.5	0.8
data3	15	50	0.1	15	50	0.1	10	50	0.1	-0.1877	-0.1	-0.03	0.0006579

TABLE 9: Tuning parameters of a Ramp PID-SMC

	k_1	λ_1	ϕ_1	k_2	λ_2	ϕ_2	k_3	λ_3	ϕ_3	SS error$_1$	SS error$_2$	SS error$_3$	RMS error
data1	10	1	0.1	10	6	0.1	10	8	0.1	1e-6	1e-6	1e-6	1.2e-6
data2	100	1	0.1	100	6	0.1	100	8	0.1	0.2	0.05	-0.02	-0.037
data3	10	10	0.1	10	60	0.1	10	80	0.1	0.22	-0.21	-0.19	0.09

TABLE 10: Tuning parameters of a Step PD-SMC

	k_1	λ_1	ϕ_1	k_2	λ_2	ϕ_2	k_3	λ_3	ϕ_3	SS error$_1$	SS error$_2$	SS error$_3$	RMS error
data1	5	15	0.1	5	15	0.1	5	10	0.1	-6e-12	-8.5e-11	-1.7e-11	8.3e-5
data2	50	15	0.1	50	15	0.1	50	10	0.1	0.09	0.06	0.02	0.00162
data3	5	150	0.1	5	150	0.1	5	100	0.1	377.7	377	272	0.732

TABLE 11: Tuning parameters of a Ramp PD-SMC

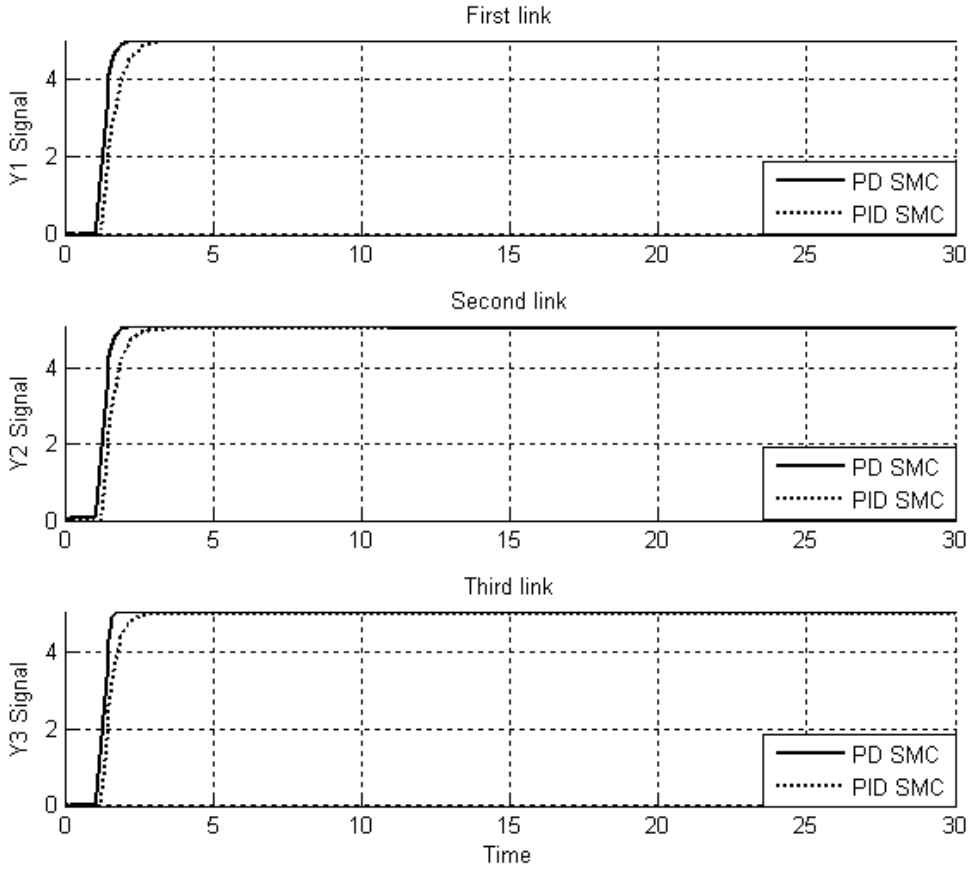


FIGURE 19: Step PD-SMC and PID-SMC for First, second and third link trajectory without any disturbance

By comparing step response, Figure 19, in PD and PID-SMC, conversely the PID's overshoot **(0%)** is lower than PD's **(1%)**, the PD's rise time **(0.483 Sec)** is dramatically lower than PID's **(0.9 Sec)**; in addition the Settling time in PD **(Settling time=0.65 Sec)** is fairly lower than PID **(Settling time=1.4 Sec)**.

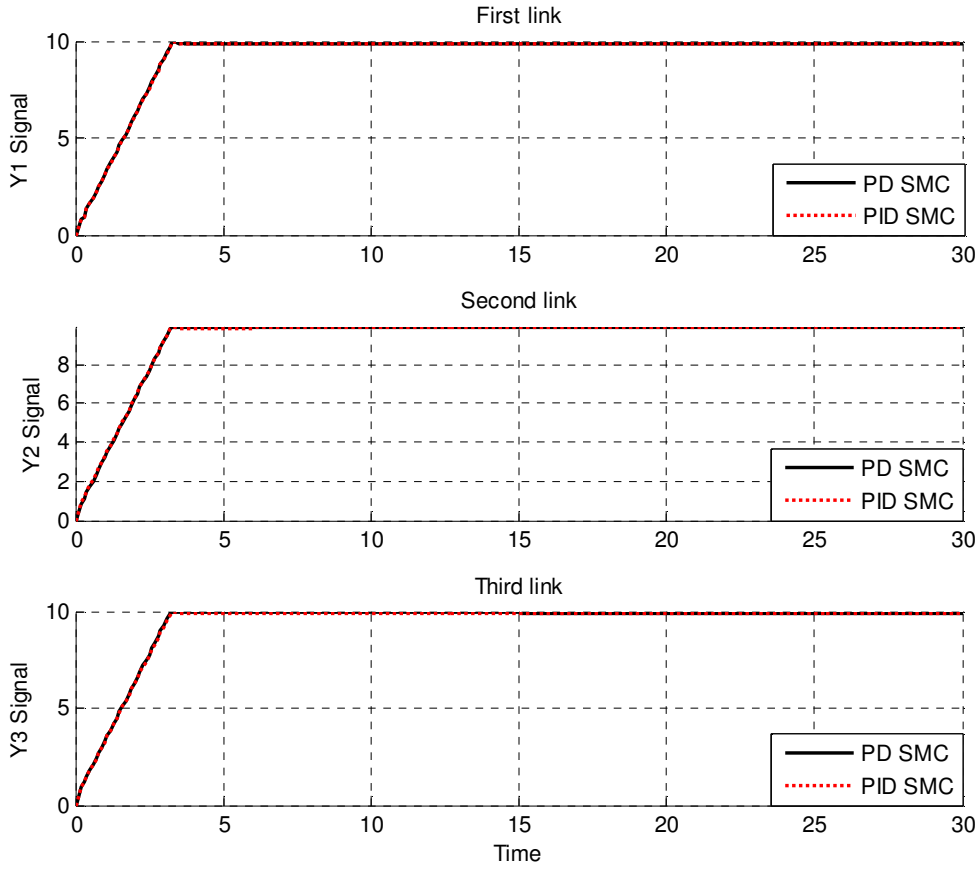


FIGURE 20: Ramp PD SMC and PID SMC for First, second and third link trajectory without any disturbance.

Figure 20 shows that, the trajectories response process that in the first 3.3 seconds rise to 10 then they are on a stable state up to the second 30.

Disturbance Rejection

Figures 21 and 22 are indicated the power disturbance removal in PD and PID-SMC. As mentioned before, SMC is one of the most important robust nonlinear controllers. Besides a band limited white noise with predefined of 40% the power of input signal is applied to the step and ramp PD and PID-SMC; it found slight oscillations in trajectory responses.

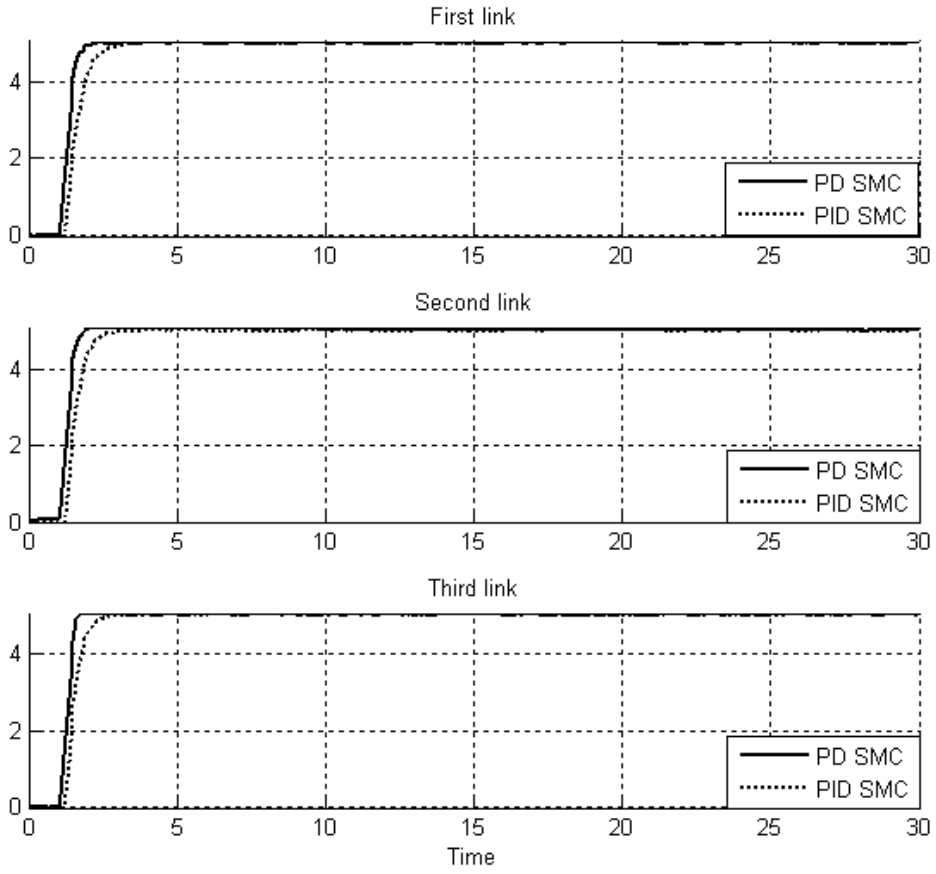


FIGURE 21: Step PD SMC and PID SMC for First, second and third link trajectory with external disturbance

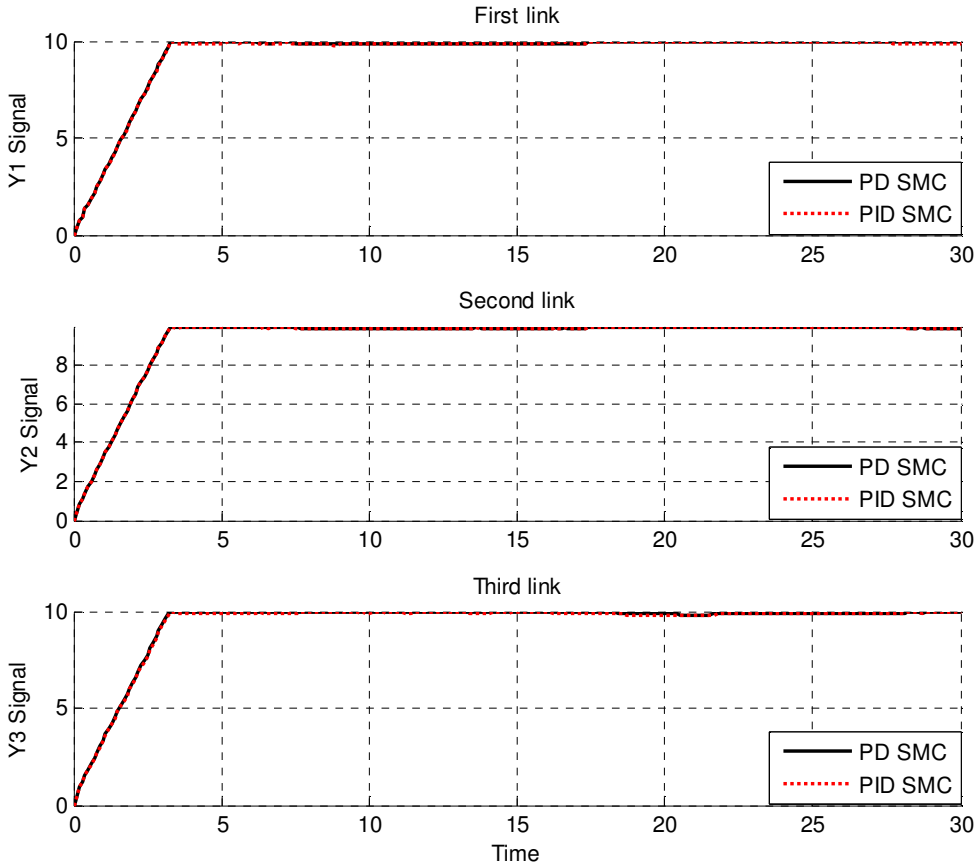


FIGURE 22: Ramp PD SMC and PID SMC for first, second and third link trajectory with external disturbance

Among above graphs (21 and 22), relating to step and ramp trajectories following with external disturbance, PID SMC and PD SMC have slightly fluctuations. By comparing overshoot, rise time, and settling time; PID's overshoot **(0.9%)** is lower than PD's **(1.1%)**, PD's rise time **(0.48 sec)** is considerably lower than PID's **(0.9 sec)** and finally the Settling time in PD **(Settling time=0.65 Sec)** is quite lower than PID **(Settling time=1.5 Sec)**.

Chattering phenomenon: As mentioned in previous chapter, chattering is one of the most important challenges in sliding mode controller which one of the major objectives in this research is reduce or remove the chattering in system's output. To reduce the chattering researcher is used $saturation$ function instead of $switching$ function. Figure 23 has shown the power of boundary layer (saturation) method to reduce the chattering in PD-SMC.

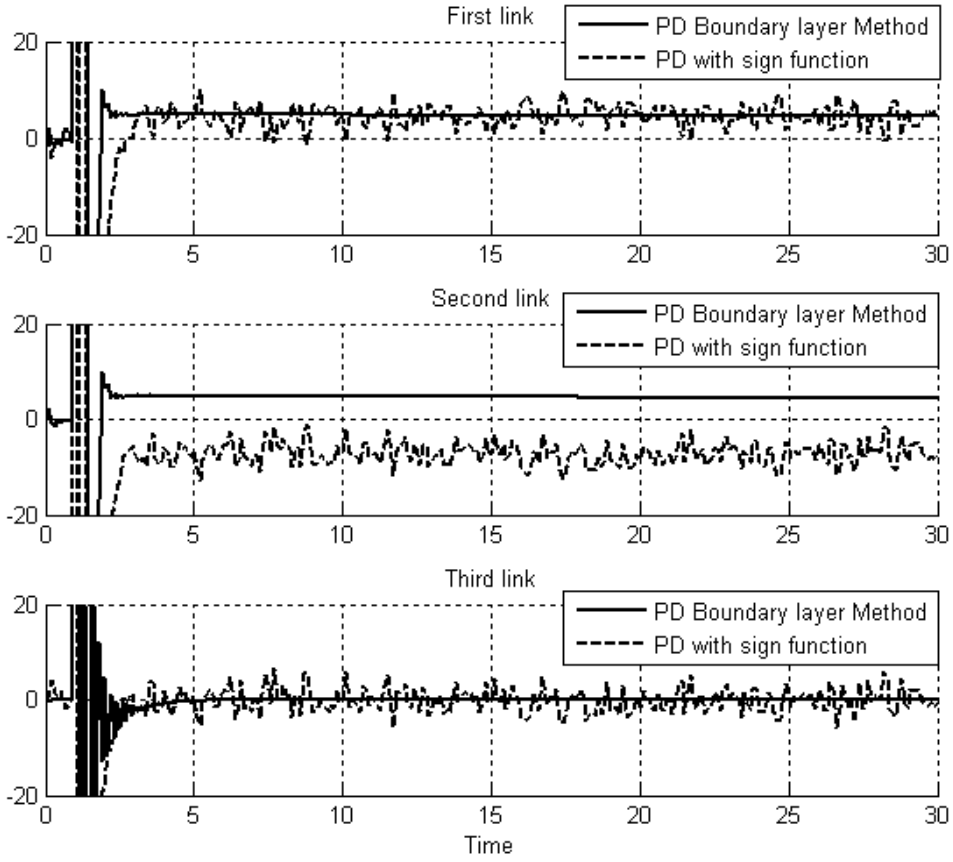


FIGURE 23: PD-SMC boundary layer methods Vs. PD-SMC with discontinuous (Sign) function

Figures 24 and 25 have indicated the power of chattering rejection in PD and PID-SMC, with and without disturbance. As mentioned before, chattering can caused to the hitting in driver and mechanical parts so reduce the chattering is more important. Furthermore band limited white noise with predefined of 40% the power of input signal is applied the step and ramp PD and PID-SMC, it seen that slight oscillations in third joint trajectory responses. Overall in this research with regard to the step response, PD-SMC has the steady chattering compared to the PID-SMC.

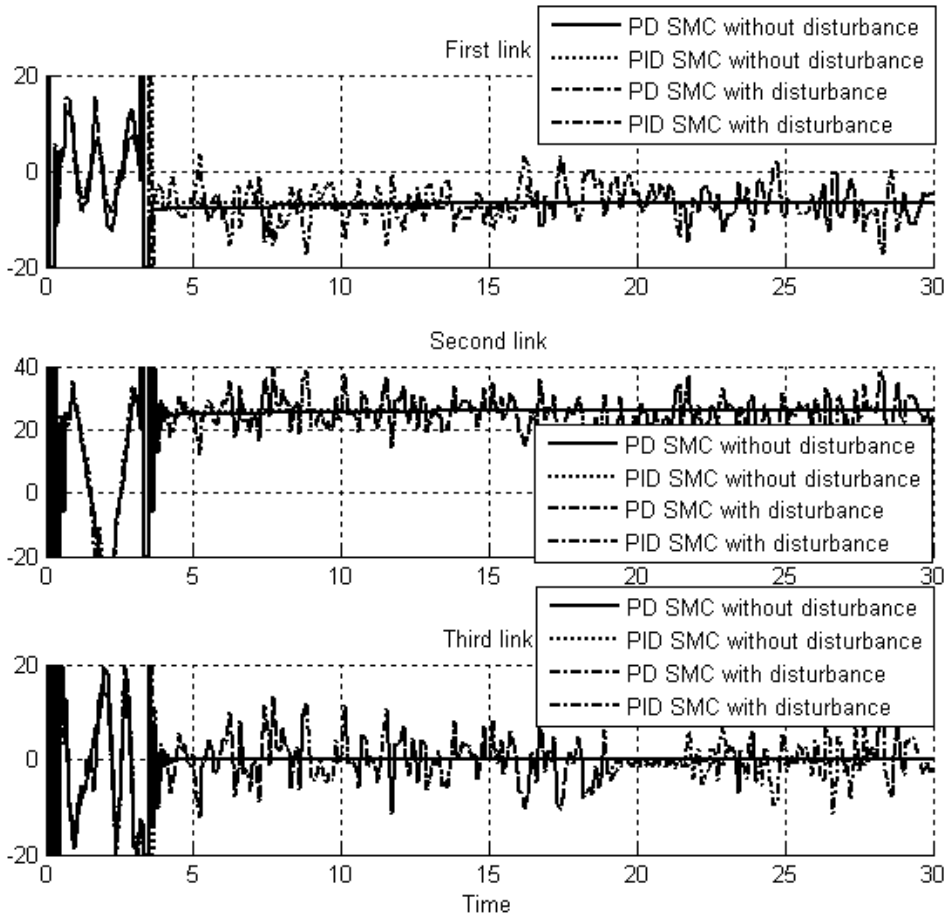


FIGURE 24: Step PID SMC and PD SMC for First, second and third link chattering without and with disturbance.

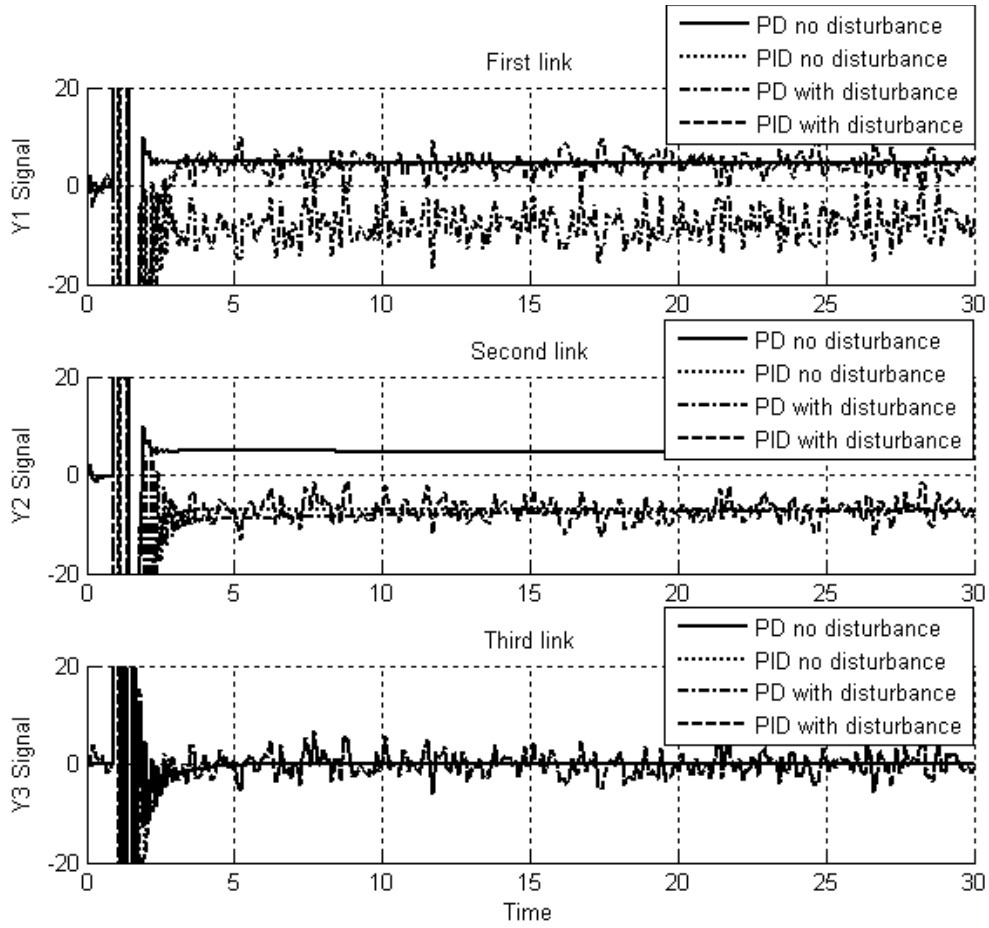


FIGURE 25: Ramp PID SMC and PD SMC for First, second and third link chattering without and with disturbance.

Errors in the model: Figures 26 and 27 have shown the error disturbance in PD and PID SMC. The controllers with no external disturbances have the same error response, but PID SMC has the better steady state error when the robot manipulator has an external disturbance. By comparing steady and RMS error in a system with no disturbance it found that the PID's errors **(Steady State error = 0 and RMS error=1e-8)** are approximately less than PD's **(Steady State error** $\cong 1e-6$ **and RMS error=**$1.2e-6$**)**.

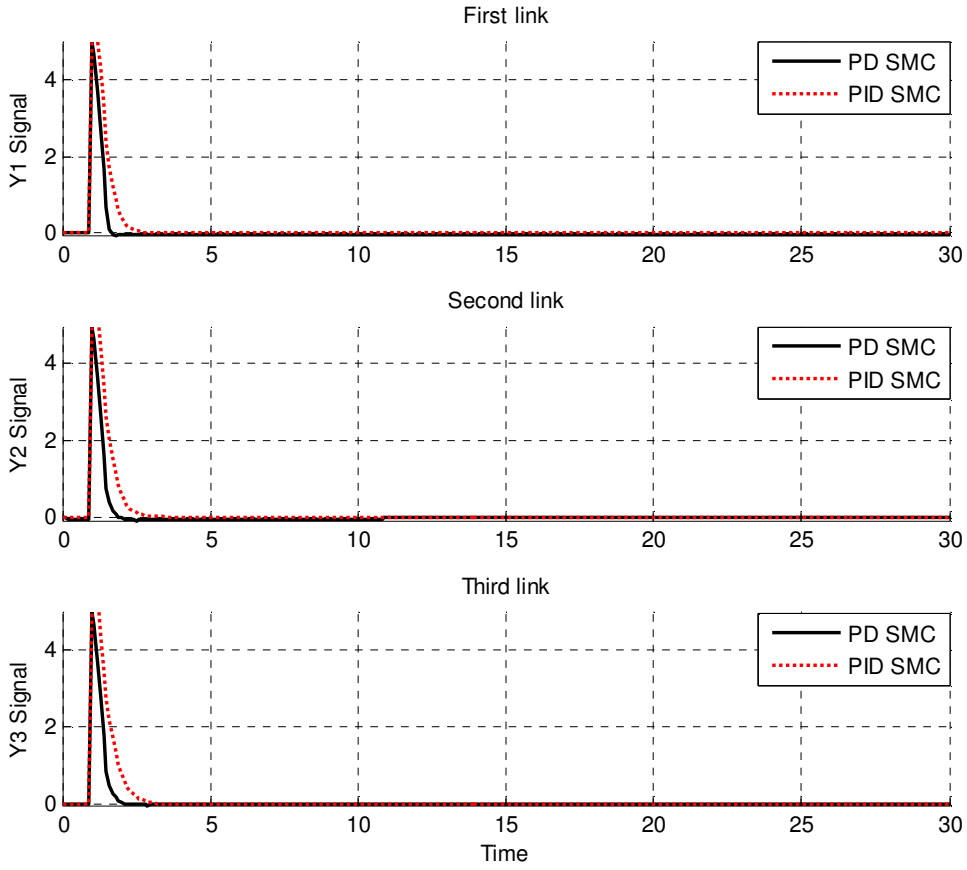


FIGURE 26: Step PID SMC and PD SMC for First, second and third link steady state error performance.

Above graphs (26 and 27) show that in first seconds; PID SMC and PD SMC are increasing very fast. By comparing the steady state error and RMS error it found that the PID's errors **(Steady State error = -0.0007 and RMS error=0.0008)** are fairly less than PD's **(Steady State error** $\cong \mathbf{0.0012}$ **and RMS error=**$\mathbf{0.0018}$**)**, When disturbance is applied to PD and PID SMC the errors are about 13% growth.

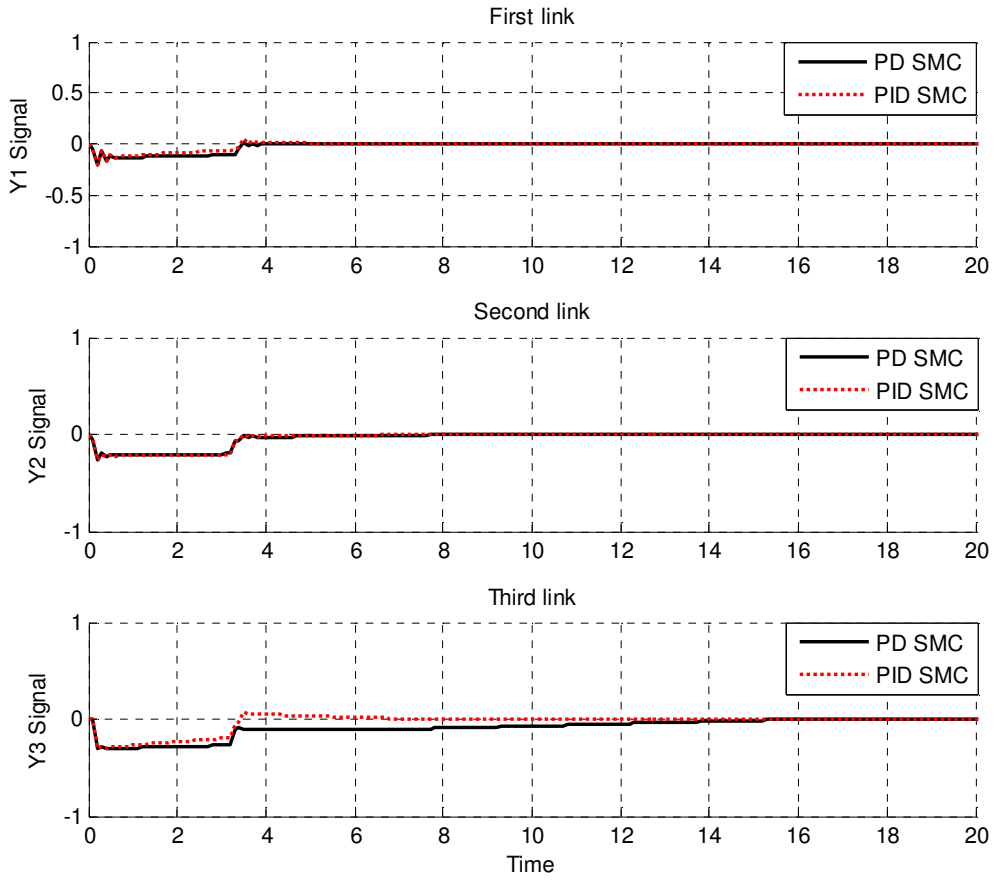


FIGURE 27: Ramp PID SMC and PD SMC for First, second and third link steady state error performance

5. CONCLUSION

In this research we introduced, basic concepts of robot manipulator (e.g., PUMA 560 robot manipulator) and nonlinear control methodology. PUMA 560 robot manipulator is a 6 DOF serial robot manipulator. One of the most active research areas in the field of robotics is robot manipulators control, because these systems are multi-input multi-output (MIMO), nonlinear, and uncertainty. At present, robot manipulators are used in unknown and unstructured situation and caused to provide complicated systems, consequently strong mathematical tools are used in new control methodologies to design nonlinear robust controller with satisfactory performance (e.g., minimum error, good trajectory, disturbance rejection). Sliding mode controller (SMC) is a significant nonlinear controller under condition of partly uncertain dynamic parameters of system. This controller is used to control of highly nonlinear systems especially for robot manipulators, because this controller is a robust and stable. Conversely, pure sliding mode controller is used in many applications; it has an important drawback namely; chattering phenomenon. The chattering phenomenon problem can be reduced by using linear saturation boundary layer function in sliding mode control law. Lyapunov stability is proved in pure sliding mode controller based on switching (sign) function.

REFERENCES

[1] T. R. Kurfess, Robotics and automation handbook: CRC, 2005.

[2] J. J. E. Slotine and W. Li, Applied nonlinear control vol. 461: Prentice hall Englewood Cliffs, NJ, 1991.

[3] K. Ogata, Modern control engineering: Prentice Hall, 2009.

[4] L. Cheng, Z. G. Hou, M. Tan, D. Liu and A. M. Zou, "Multi-agent based adaptive consensus control for multiple manipulators with kinematic uncertainties," 2008, pp. 189-194.

[5] J. J. D'Azzo, C. H. Houpis and S. N. Sheldon, Linear control system analysis and design with MATLAB: CRC, 2003.

[6] B. Siciliano and O. Khatib, Springer handbook of robotics: Springer-Verlag New York Inc, 2008.

[7] I. Boiko, L. Fridman, A. Pisano and E. Usai, "Analysis of chattering in systems with second-order sliding modes," IEEE Transactions on Automatic Control, No. 11, vol. 52,pp. 2085-2102, 2007.

[8] J. Wang, A. Rad and P. Chan, "Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching," Fuzzy Sets and Systems, No. 1, vol. 122,pp. 21-30, 2001.

[9] C. Wu, "Robot accuracy analysis based on kinematics," IEEE Journal of Robotics and Automation, No. 3, vol. 2, pp. 171-179, 1986.

[10] H. Zhang and R. P. Paul, "A parallel solution to robot inverse kinematics," IEEE conference proceeding, 2002, pp. 1140-1145.

[11] J. Kieffer, "A path following algorithm for manipulator inverse kinematics," IEEE conference proceeding, 2002, pp. 475-480.

[12] Z. Ahmad and A. Guez, "On the solution to the inverse kinematic problem(of robot)," IEEE conference proceeding, 1990, pp. 1692-1697.

[13] F. T. Cheng, T. L. Hour, Y. Y. Sun and T. H. Chen, "Study and resolution of singularities for a 6-DOF PUMA manipulator," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, No. 2, vol. 27, pp. 332-343, 2002.

[14] M. W. Spong and M. Vidyasagar, Robot dynamics and control: Wiley-India, 2009.

[15] A. Vivas and V. Mosquera, "Predictive functional control of a PUMA robot," Conference Proceedings, 2005.

[16] D. Nguyen-Tuong, M. Seeger and J. Peters, "Computed torque control with nonparametric regression models," IEEE conference proceeding, 2008, pp. 212-217.

[17] V. Utkin, "Variable structure systems with sliding modes," Automatic Control, IEEE Transactions on, No. 2, vol. 22, pp. 212-222, 2002.

[18] R. A. DeCarlo, S. H. Zak and G. P. Matthews, "Variable structure control of nonlinear multivariable systems: a tutorial," Proceedings of the IEEE, No. 3, vol. 76, pp. 212-232, 2002.

[19] K. D. Young, V. Utkin and U. Ozguner, "A control engineer's guide to sliding mode control," IEEE conference proceeding, 2002, pp. 1-14.

[20] O. Kaynak, "Guest editorial special section on computationally intelligent methodologies and sliding-mode control," IEEE Transactions on Industrial Electronics, No. 1, vol. 48, pp. 2-3, 2001.

[21] J. J. Slotine and S. Sastry, "Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators†," International Journal of Control, No. 2, vol. 38, pp. 465-492, 1983.

[22] J. J. E. Slotine, "Sliding controller design for non-linear systems," International Journal of Control, No. 2, vol. 40, pp. 421-434, 1984.

[23] R. Palm, "Sliding mode fuzzy control," IEEE conference proceeding,2002, pp. 519-526.

[24] C. C. Weng and W. S. Yu, "Adaptive fuzzy sliding mode control for linear time-varying uncertain systems," IEEE conference proceeding, 2008, pp. 1483-1490.

[25] M. Ertugrul and O. Kaynak, "Neuro sliding mode control of robotic manipulators," Mechatronics Journal, No. 1, vol. 10, pp. 239-263, 2000.

[26] P. Kachroo and M. Tomizuka, "Chattering reduction and error convergence in the sliding-mode control of a class of nonlinear systems," Automatic Control, IEEE Transactions on, No. 7, vol. 41, pp. 1063-1068, 2002.

[27] H. Elmali and N. Olgac, "Implementation of sliding mode control with perturbation estimation (SMCPE)," Control Systems Technology, IEEE Transactions on, No. 1, vol. 4, pp. 79-85, 2002.

[28] J. Moura and N. Olgac, "A comparative study on simulations vs. experiments of SMCPE," IEEE conference proceeding, 2002, pp. 996-1000.

[29] Y. Li and Q. Xu, "Adaptive Sliding Mode Control With Perturbation Estimation and PID Sliding Surface for Motion Tracking of a Piezo-Driven Micromanipulator," Control Systems Technology, IEEE Transactions on, No. 4, vol. 18, pp. 798-810, 2010.

[30] B. Wu, Y. Dong, S. Wu, D. Xu and K. Zhao, "An integral variable structure controller with fuzzy tuning design for electro-hydraulic driving Stewart platform," IEEE conference proceeding, 2006, pp. 5-945.

[31] Farzin Piltan , N. Sulaiman, Zahra Tajpaykar, Payman Ferdosali, Mehdi Rashidi, “Design Artificial Nonlinear Robust Controller Based on CTLC and FSMC with Tunable Gain,” International Journal of Robotic and Automation, 2 (3): 205-220, 2011.

[32] Farzin Piltan, A. R. Salehi and Nasri B Sulaiman.,” Design artificial robust control of second order system based on adaptive fuzzy gain scheduling,” world applied science journal (WASJ), 13 (5): 1085-1092, 2011.

[33] Farzin Piltan, N. Sulaiman, Atefeh Gavahian, Samira Soltani, Samaneh Roosta, “Design Mathematical Tunable Gain PID-Like Sliding Mode Fuzzy Controller with Minimum Rule Base,” International Journal of Robotic and Automation, 2 (3): 146-156, 2011.

[34] Farzin Piltan , A. Zare, Nasri B. Sulaiman, M. H. Marhaban and R. Ramli, , “A Model Free Robust Sliding Surface Slope Adjustment in Sliding Mode Control for Robot Manipulator,” World Applied Science Journal, 12 (12): 2330-2336, 2011.

[35] Farzin Piltan , A. H. Aryanfar, Nasri B. Sulaiman, M. H. Marhaban and R. Ramli “Design Adaptive Fuzzy Robust Controllers for Robot Manipulator,” World Applied Science Journal, 12 (12): 2317-2329, 2011.

[36] Farzin Piltan, N. Sulaiman , Arash Zargari, Mohammad Keshavarz, Ali Badri , “Design PID-Like Fuzzy Controller With Minimum Rule Base and Mathematical Proposed On-line

Tunable Gain: Applied to Robot Manipulator," International Journal of Artificial intelligence and expert system, 2 (4):184-195, 2011.

[37] Farzin Piltan, Nasri Sulaiman, M. H. Marhaban and R. Ramli, "Design On-Line Tunable Gain Artificial Nonlinear Controller," Journal of Advances In Computer Research, 2 (4): 75-83, 2011.

[38] Farzin Piltan, N. Sulaiman, Payman Ferdosali, Iraj Assadi Talooki, " Design Model Free Fuzzy Sliding Mode Control: Applied to Internal Combustion Engine," International Journal of Engineering, 5 (4):302-312, 2011.

[39] Farzin Piltan, N. Sulaiman, Samaneh Roosta, M.H. Marhaban, R. Ramli, "Design a New Sliding Mode Adaptive Hybrid Fuzzy Controller," Journal of Advanced Science & Engineering Research , 1 (1): 115-123, 2011.

[40] Farzin Piltan, Atefe Gavahian, N. Sulaiman, M.H. Marhaban, R. Ramli, "Novel Sliding Mode Controller for robot manipulator using FPGA," Journal of Advanced Science & Engineering Research, 1 (1): 1-22, 2011.

[41] Farzin Piltan, N. Sulaiman, A. Jalali & F. Danesh Narouei, "Design of Model Free Adaptive Fuzzy Computed Torque Controller: Applied to Nonlinear Second Order System," International Journal of Robotics and Automation, 2 (4):232-244, 2011.

[42] Farzin Piltan, N. Sulaiman, Iraj Asadi Talooki, Payman Ferdosali, "Control of IC Engine: Design a Novel MIMO Fuzzy Backstepping Adaptive Based Fuzzy Estimator Variable Structure Control ," International Journal of Robotics and Automation, 2 (5):360-380, 2011.

[43] Farzin Piltan, N. Sulaiman, Payman Ferdosali, Mehdi Rashidi, Zahra Tajpeikar, "Adaptive MIMO Fuzzy Compensate Fuzzy Sliding Mode Algorithm: Applied to Second Order Nonlinear System," International Journal of Engineering, 5 (5): 380-398, 2011.

[44] Farzin Piltan, N. Sulaiman, Hajar Nasiri, Sadeq Allahdadi, Mohammad A. Bairami, "Novel Robot Manipulator Adaptive Artificial Control: Design a Novel SISO Adaptive Fuzzy Sliding Algorithm Inverse Dynamic Like Method," International Journal of Engineering, 5 (5): 399-418, 2011.

[45] Farzin Piltan, N. Sulaiman, Sadeq Allahdadi, Mohammadali Dialame, Abbas Zare, "Position Control of Robot Manipulator: Design a Novel SISO Adaptive Sliding Mode Fuzzy PD Fuzzy Sliding Mode Control," International Journal of Artificial intelligence and Expert System, 2 (5):208-228, 2011.

[46] Farzin Piltan, SH. Tayebi HAGHIGHI, N. Sulaiman, Iman Nazari, Sobhan Siamak, "Artificial Control of PUMA Robot Manipulator: A-Review of Fuzzy Inference Engine And Application to Classical Controller ," International Journal of Robotics and Automation, 2 (5):401-425, 2011.

[47] Farzin Piltan, N. Sulaiman, Abbas Zare, Sadeq Allahdadi, Mohammadali Dialame, "Design Adaptive Fuzzy Inference Sliding Mode Algorithm: Applied to Robot Arm," International Journal of Robotics and Automation , 2 (5): 283-297, 2011.

[48] Farzin Piltan, Amin Jalali, N. Sulaiman, Atefeh Gavahian, Sobhan Siamak, "Novel Artificial Control of Nonlinear Uncertain System: Design a Novel Modified PSO SISO Lyapunov Based Fuzzy Sliding Mode Algorithm ," International Journal of Robotics and Automation, 2 (5): 298-316, 2011.

[49] Farzin Piltan, N. Sulaiman, Amin Jalali, Koorosh Aslansefat, "Evolutionary Design of Mathematical tunable FPGA Based MIMO Fuzzy Estimator Sliding Mode Based Lyapunov

Algorithm: Applied to Robot Manipulator," International Journal of Robotics and Automation, 2 (5):317-343, 2011.

[50] Farzin Piltan, N. Sulaiman, Samaneh Roosta, Atefeh Gavahian, Samira Soltani, "Evolutionary Design of Backstepping Artificial Sliding Mode Based Position Algorithm: Applied to Robot Manipulator," International Journal of Engineering, 5 (5):419-434, 2011.

[51] Farzin Piltan, N. Sulaiman, S.Soltani, M. H. Marhaban & R. Ramli, "An Adaptive sliding surface slope adjustment in PD Sliding Mode Fuzzy Control for Robot Manipulator," International Journal of Control and Automation , 4 (3): 65-76, 2011.

[52] Farzin Piltan, N. Sulaiman, Mehdi Rashidi, Zahra Tajpaikar, Payman Ferdosali, "Design and Implementation of Sliding Mode Algorithm: Applied to Robot Manipulator-A Review ," International Journal of Robotics and Automation, 2 (5):265-282, 2011.

[53] Farzin Piltan, N. Sulaiman, Amin Jalali, Sobhan Siamak, and Iman Nazari, "Control of Robot Manipulator: Design a Novel Tuning MIMO Fuzzy Backstepping Adaptive Based Fuzzy Estimator Variable Structure Control ," International Journal of Control and Automation, 4 (4):91-110, 2011.

[54] Farzin Piltan, N. Sulaiman, Atefeh Gavahian, Samaneh Roosta, Samira Soltani, "On line Tuning Premise and Consequence FIS: Design Fuzzy Adaptive Fuzzy Sliding Mode Controller Based on Lyaponuv Theory," International Journal of Robotics and Automation, 2 (5):381-400, 2011.

[55] Farzin Piltan, N. Sulaiman, Samaneh Roosta, Atefeh Gavahian, Samira Soltani, "Artificial Chattering Free on-line Fuzzy Sliding Mode Algorithm for Uncertain System: Applied in Robot Manipulator," International Journal of Engineering, 5 (5):360-379, 2011.

[56] Farzin Piltan, N. Sulaiman and I.AsadiTalooki, "Evolutionary Design on-line Sliding Fuzzy Gain Scheduling Sliding Mode Algorithm: Applied to Internal Combustion Engine," International Journal of Engineering Science and Technology, 3 (10):7301-7308, 2011.

[57] Farzin Piltan, Nasri B Sulaiman, Iraj Asadi Talooki and Payman Ferdosali.," Designing On-Line Tunable Gain Fuzzy Sliding Mode Controller Using Sliding Mode Fuzzy Algorithm: Applied to Internal Combustion Engine," world applied science journal (WASJ), 15 (3): 422-428, 2011.

[58] B. K. Yoo and W. C. Ham, "Adaptive control of robot manipulator using fuzzy compensator," Fuzzy Systems, IEEE Transactions on, No. 2, vol. 8, pp. 186-199, 2002.

[59] H. Medhaffar, N. Derbel and T. Damak, "A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator," International Journal of Modelling, Identification and Control, No. 1, vol. 1, pp. 23-29, 2006.

[60] Y. Guo and P. Y. Woo, "An adaptive fuzzy sliding mode controller for robotic manipulators," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, No. 2, vol. 33, pp. 149-159, 2003.

[61] C. M. Lin and C. F. Hsu, "Adaptive fuzzy sliding-mode control for induction servomotor systems," Energy Conversion, IEEE Transactions on, No. 2, vol. 19, pp. 362-368, 2004.

[62] N. Sulaiman, Z. A. Obaid, M. Marhaban and M. Hamidon , "Design and Implementation of FPGA-Based Systems-A Review," Australian Journal of Basic and Applied Sciences, No. 4, vol. 3, pp. 3575-3596, 2009.

[63] X. Shao and D. Sun, "Development of an FPGA-based motion control ASIC for robotic manipulators," IEEE Conference , 2006, pp. 8221-8225.

[64] Y. S. Kung, K. Tseng, C. Chen, H. Sze and A. Wang, "FPGA-implementation of inverse kinematics and servo controller for robot manipulator," Proc. IEEE Int. on Robotics and Biomimetics, pp. 1163–1168, 2006.

[65] X. Shao, D. Sun and J. K. Mills, "A new motion control hardware architecture with FPGA-based IC design for robotic manipulators," IEEE Conference, 2006, pp. 3520-3525.

[66] Y. S. Kung, C. S. Chen and G. S. Shu, "Design and Implementation of a Servo System for Robotic Manipulator," CACS, 2005.

[67] U. D. Meshram and R. Harkare, "FPGA Based Five Axis Robot Arm Controller," IEEE Conference, 2005, pp. 3520-3525.

[68] U. Meshram, P. Bande, P. Dwaramwar and R. Harkare, "Robot arm controller using FPGA," IEEE Conference, 2009, pp. 8-11.

[69] Y. S. Kung and G. S. Shu, "Development of a FPGA-based motion control IC for robot arm," IEEE Conference, 2006, pp. 1397-1402.

[70] Z. A. Obaid, N. Sulaiman and M. Hamidon, "Developed Method of FPGA-based Fuzzy Logic Controller Design with the Aid of Conventional PID Algorithm," Australian Journal of Basic and Applied Sciences, No. 3, vol. 3, pp. 2724-2740, 2009.

[71] S. T. Karris, Digital circuit analysis and design with Simulink modeling and introduction to CPLDs and FPGAs: Orchard Pubns, 2007.

[72] K. D. Rogers, "Acceleration and implementation of a DSP phase-based frequency estimation algorithm: MATLAB/SIMULINK to FPGA via XILINX system generator," Citeseer, 2004.

[73] F. J. Lin, D. H. Wang and P. K. Huang, "FPGA-based fuzzy sliding-mode control for a linear induction motor drive," IEEE journal of electrical power application, No. 5, Vol. 152, 2005, pp. 1137-1148.

[74] R. R. Ramos, D. Biel, E. Fossas and F. Guinjoan, "A fixed-frequency quasi-sliding control algorithm: application to power inverters design by means of FPGA implementation," Power Electronics, IEEE Transactions on, No. 1, vol. 18, pp. 344-355, 2003.

[75] Xiaosong. Lu, "An investigation of adaptive fuzzy sliding mode control for robot manipulator," Carleton university Ottawa,2007.

[76] S. Lentijo, S. Pytel, A. Monti, J. Hudgins, E. Santi and G. Simin, "FPGA based sliding mode control for high frequency power converters," IEEE Conference, 2004, pp. 3588-3592.

[77] B. S. R. Armstrong, "Dynamics for robot control: friction modeling and ensuring excitation during parameter identification," 1988.

[78] C. L. Clover, "Control system design for robots used in simulating dynamic force and moment interaction in virtual reality applications," 1996.

[79] K. R. Horspool, Cartesian-space Adaptive Control for Dual-arm Force Control Using Industrial Robots: University of New Mexico, 2003.

[80] B. Armstrong, O. Khatib and J. Burdick, "The explicit dynamic model and inertial parameters of the PUMA 560 arm," IEEE Conference, 2002, pp. 510-518.

[81] P. I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," IEEE Conference, 2002, pp. 1608-1613.

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

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

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

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

[86] Farzin Piltan, I. Nazari, S. Siamak, P. Ferdosali ,"Methodology of FPGA-based mathematical error-based tuning sliding mode controller" International Journal of Control and Automation, 5(1): 89-110, 2012.

[87] Farzin Piltan, M. A. Dialame, A. Zare, A. Badri ,"Design Novel Lookup table changed Auto Tuning FSMC: Applied to Robot Manipulator" International Journal of Engineering, 6(1): 25-40, 2012.

[88] Farzin Piltan, B. Boroomand, A. Jahed, H. Rezaie ,"Methodology of Mathematical Error-Based Tuning Sliding Mode Controller" International Journal of Engineering, 6(2): 96-112, 2012.

[89] Farzin Piltan, F. Aghayari, M. R. Rashidian, M. Shamsodini, "A New Estimate Sliding Mode Fuzzy Controller for Robotic Manipulator" International Journal of Robotics and Automation, 3(1): 45-58, 2012.

[90] Farzin Piltan, M. Keshavarz, A. Badri, A. Zargari , "Design novel nonlinear controller applied to robot manipulator: design new feedback linearization fuzzy controller with minimum rule base tuning method" International Journal of Robotics and Automation, 3(1): 1-18, 2012.

[91] Piltan, F., et al. "Design sliding mode controller for robot manipulator with artificial tunable gain". Canaidian Journal of pure and applied science, 5 (2), 1573-1579, 2011.

[92] Farzin Piltan, A. Hosainpour, E. Mazlomian, M.Shamsodini, M.H Yarmahmoudi. "Online Tuning Chattering Free Sliding Mode Fuzzy Control Design: Lyapunov Approach" International Journal of Robotics and Automation, 3(3): 2012.

[93] Farzin Piltan , M.H. Yarmahmoudi, M. Shamsodini, E.Mazlomian, A.Hosainpour. " PUMA-560 Robot Manipulator Position Computed Torque Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control and MATLAB Courses" International Journal of Robotics and Automation, 3(3): 2012.

[94] Farzin Piltan, R. Bayat, F. Aghayari, B. Boroomand. “Design Error-Based Linear Model-Free Evaluation Performance Computed Torque Controller” International Journal of Robotics and Automation, 3(3): 2012.

[95] Farzin Piltan, J. Meigolinedjad, S. Mehrara, S. Rahmdel. ” Evaluation Performance of 2nd Order Nonlinear System: Baseline Control Tunable Gain Sliding Mode Methodology” International Journal of Robotics and Automation, 3(3): 2012.

[96] Farzin Piltan, S. Rahmdel, S. Mehrara, R. Bayat.” Sliding Mode Methodology Vs. Computed Torque Methodology Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control Courses” International Journal of Engineering, 3(3): 2012.

[97] Farzin Piltan, M. Mirzaie, F. Shahriyari, Iman Nazari & S. Emamzadeh.” Design Baseline Computed Torque Controller” International Journal of Engineering, 3(3): 2012.