Modeling and Control of Four Degrees of Freedom Surgical Robot Manipulator Using MATLAB/SIMULINK

Farzin Piltan[1], Ali Taghizadegan[1] and Nasri B Sulaiman[1,2]

[1]Intelligent System and Robotic Lab, Iranian Institute of Advanced Science and Technology (IRAN SSP), Shiraz/Iran
[2]Department of Electrical Engineering, Faculty of Engineering, University Putra Malaysia, Malaysia
piltan_f@iranssp.org, www.iranssp.org/english

Abstract

Recent development of robot technology is revolutionizing the medical field. The concept of using robot assistance in medical surgery has been receiving more and more recognition throughout the world. Robot-assisted surgery has the advantage of reducing surgeons' hand tremor, decreasing post-operative complications, reducing patients' pains, and increasing operation dexterity inside the patients' body. Robotic assistants have been broadly used in many medical fields such as orthopedics, neurology, urology and cardiology, and robot assisted surgery is keeping expanding its influences in more general medical field.

This research study aims at utilizing advanced robotics manipulator technologies to help surgeons perform delicate procedures associated with surgery. The Four-axis Virtual Robot arm (FVR) is a MATLAB-based computer program, which can be used to simulate the functions of a real robotic manipulator in terms of design parameters, movement and control. It has been designed with adjustable kinematic parameters to mimic a 4-axis articulate robotic manipulator with revolute joints having 4 degrees of freedom. The FVR can be manipulated using direct kinematics to change the spatial orientation of virtual objects in three dimensions. Picking and placing of virtual objects can be done by using the virtual proximity sensors and virtual touch sensors incorporated in to the jaw design of the FVR. Furthermore, it can be trained to perform a sequence of movements repeatedly, to simulate the function of a real surgical robotic manipulator. All steps to modeling are discussed in this research. Proportional-Integral-Derivative control technique is used to control of FVR.

Keywords: robot technology, medical surgery, robot-assisted surgery, Four-axis Virtual Robot arm (FVR), MATLAB-based computer program, kinematic parameters

1. Introduction and Background

Robot-assisted surgery has become a burgeoning field in recent years. An interdisciplinary subject involves both robot technologies and medical intervention. Because of its potentials to improve precision, enhance dexterity, eliminate tremor, reduce complication rates, and enable novel procedures not previously achievable, robot-assisted surgery has drawn broad attentions from the robotics research community and the medical world over the past decade. One successful implementation of robot-assisted surgery is the DAVINCI surgical system of Intuitive Surgical. It utilizes a tele-operation control mode with a master controlled by the surgeon and a slave surgical assistant operating on the patient. Despite its capability of performing abdominal procedures, the DAVINCI system is too bulky, and lacks the precision and dexterity required for delicate applications that require higher accuracy. Therefore, researchers are actively developing

and implementing novel robotic systems to accommodate more demanding surgical procedures. Figure 1 shows DAVINCI Surgical robot System [1-3].

Figure 1. Intuitive Surgical DAVINCI Surgical System

Robot-assisted surgery presents many challenges, out of which nonlinear manipulation, high-precision dexterous operation, distal tool dexterity, insertion depth perception and contact force feedback are major concerns. Researchers have started to investigate some of these aforementioned concerns by developing robotic assistants, but a comprehensive robotic system that is capable of assisting general surgical procedures for eyes, heart and addressing existing surgical challenges is still missing. Besides, there are also many interesting robotics-related theoretical problems to be investigated under the light of surgery, *e.g.*, multi-arm manipulation, robot performance evaluation, high-precision robot design, force sensing implementation, *etc.,* Figure 2 shows the application of robotic manipulators in medical industries [4-6].

Figure 2. The Application of Robotic Manipulator in Medical Industries

Robot manipulators are set of links which connected by joints, they are multi input and multi output (MIMO), nonlinear, time variant, uncertain dynamic systems and are developed either to replace human work in many fields such as medical. Complexities of the tasks caused to design mechanical architectures and robot manipulator with nonlinear behavior. These factors are:

- Time-varying parameters based on tear and ware.
- Simplifying suppositions in system modelling caused to have un-modelled dynamic parameters.
- External disturbance and noise measurement, which it is caused to generate uncertainties.

Surgical robot manipulators are nonlinear uncertain systems with human-like behavior therefore, control of these systems are complicated. Robot manipulators are divided into two main groups, serial links robot manipulators and parallel links robot manipulators. FVR robot manipulator is a serial link robot manipulator, in this type of robot links and joints are serially connected between base and end-effector. Study of robot manipulators are classified into two main subjects:

- kinematics
- Dynamics

Study of kinematics is important to design controller in practical applications. Dynamic modeling of robot manipulator is used to illustrate the behavior of robot manipulator (*e.g.,* MIMO, nonlinear, uncertain parameters and …), design of nonlinear conventional controller such as conventional Computed Torque Controller and Sliding Mode Controller and for modeling. It also used to explain some dynamic parameters effect to system behavior. According to the literature [7-8] FVR surgical robot manipulator is a serial links, four degrees of freedom and highly nonlinear dynamic systems, which modeling and control of this system are the main challenges in this research.

To control of FVR surgical robot manipulator, three purposes are very important:

- **Stability**: Stability is due to the proper functioning of the system. A system called stable if for any bounded input signal the system's output will stay bounded. Therefore, limitation of output deviation is very important for any design.
- **Robust**: Robust method is caused to achieve robust and stable performance in the presence of uncertainty and external disturbance. A system is robust when it would also work well under different conditions and external disturbance.
- **Reliability:** to control of nonlinear and uncertain systems, reliability play important role and most of model-base controller are reliable.

As a result, modeling and design a linear Proportional-integral-Derivative controller are the main challenges in this work. A controller (control system) is a device which cans sense information from system (*e.g., robot manipulator*) to improve the performance of the system using actuation and computation. According to the control theory, systems' controls are divided into two main groups: conventional control theory and soft computing control theory. Conventional control theories are work based on manipulator dynamic model. This technique is highly sensitive to the knowledge of all parameters of nonlinear robot manipulator's dynamic equation. Conventional control theories are divided into two main groups:

- Linear control theory
- Nonlinear control theory.

Soft computing (intelligent) control theory is free of some challenges associated to conventional control theory. This technique is worked based on intelligent control theory. This theories are divided into the following groups:
Fuzzy logic theory,

- Neural network theory,
- Genetic algorithm
- Neuro-fuzzy theory

Linear control theories are used in linear and nonlinear systems. This type of theory is used in industries and some of medical system's because design of this type of controller is simple than nonlinear controller. Generally, this type of controller is divided into the following groups [9-13]:

- Proportional-Derivative (PD) Controller
- Proportional-Integral (PI) Controller
- Proportional-Integral-Derivative (PID) Controller

This paper is organized as follows: In second section, modelling of surgical robot manipulator Lagrange dynamic and kinematic formulations and steps to modelling are presented. Detail of design PID controller, modelling and implementation of PID controller for surgical robot manipulator are presented in section three. In Section 4, the simulation result is presented and finally in Section 5, the conclusion is presented.

2. Surgical Robot Manipulator

System or plant is a set of components which work together to follow a certain objective. Based on above definition, in this research surgical robot manipulator is system. A robot is a machine which can be programmed to do a range of tasks. They have five fundamental components; brain, body, actuator, sensors and power source supply. A brain controls the robot's actions to best response to desired and actual inputs. A robot body is physical chasses which can use to holds all parts together. Actuators permit the robot to move based on electrical part (*e.g.,* motors) and mechanical part (*e.g.,* hydraulic piston). Sensors give robot information about its internal and external part of robot environment and power source supply is used to supply all parts of robot. Robot is divided into three main groups: robot manipulator, mobile robot and hybrid robot.

Robot manipulator is a collection of links which connect to each other by joints. Each joint provides one or more Degrees Of Freedom (DOF). The fixed link in this system is called the base, while the last link whose motion is prescribed and used to interact with the environment is called the end-effector [1]. Robot manipulator is divided into two main groups, serial links robot manipulator and parallel links robot manipulators. Figures 3 and 4 shows serial links surgical robot manipulator. Before the introduction and difference between of parallel and serial robot manipulators, some definitions are needed:
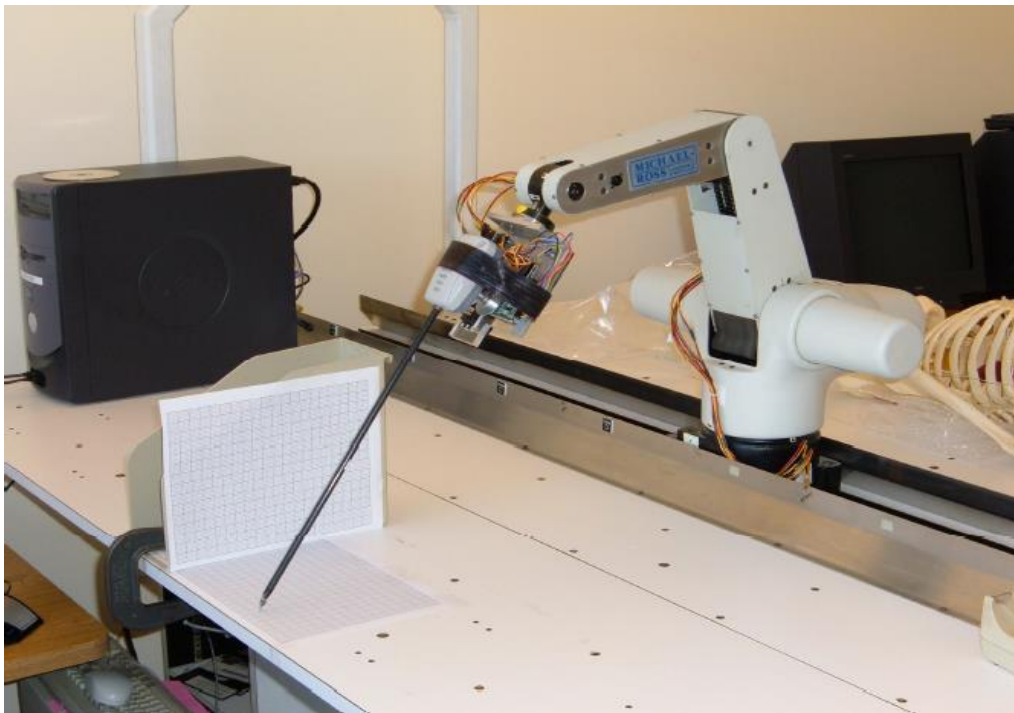

Figure 3. Type of Surgical Robotic Manipulator

- **Degree of Freedom (DOF):** in mechanics, the DOF of a mechanical system the direction of movements that define its configuration [1- 2]. This item is related to the number of joints and the type of each joint.
- **Link:** is the individual body that makes up a device, and could be either without actuator (static) or actuator (dynamic).
- **Joint:** corresponds to the connection between two links providing the required physical constraints on the relative motion between these two links. Most of joints is divided into three types; revolute joints that have one angular DOF, prismatic joints which included one translate DOF and spherical joints that consisting of three DOF.

In serial or open-chain links robot manipulator, links and joints are serially connected between base and final frame (end-effector). Parallel or closed-chain robot manipulators have at least two direction between base frame and end-effector. Serial link robot manipulators having good operating characteristics such as large workspace and high flexibility but they have disadvantages of low precision, low stiffness and low power operated at low speed to avoid excessive vibration and deflection. However the advantages of parallel robot manipulator are large strength-to-weight ratio due to higher structural rigidity, stiffness and payload they have disadvantage such as small workspace. Study of robot manipulators is classified into two main subjects: kinematics and dynamics.
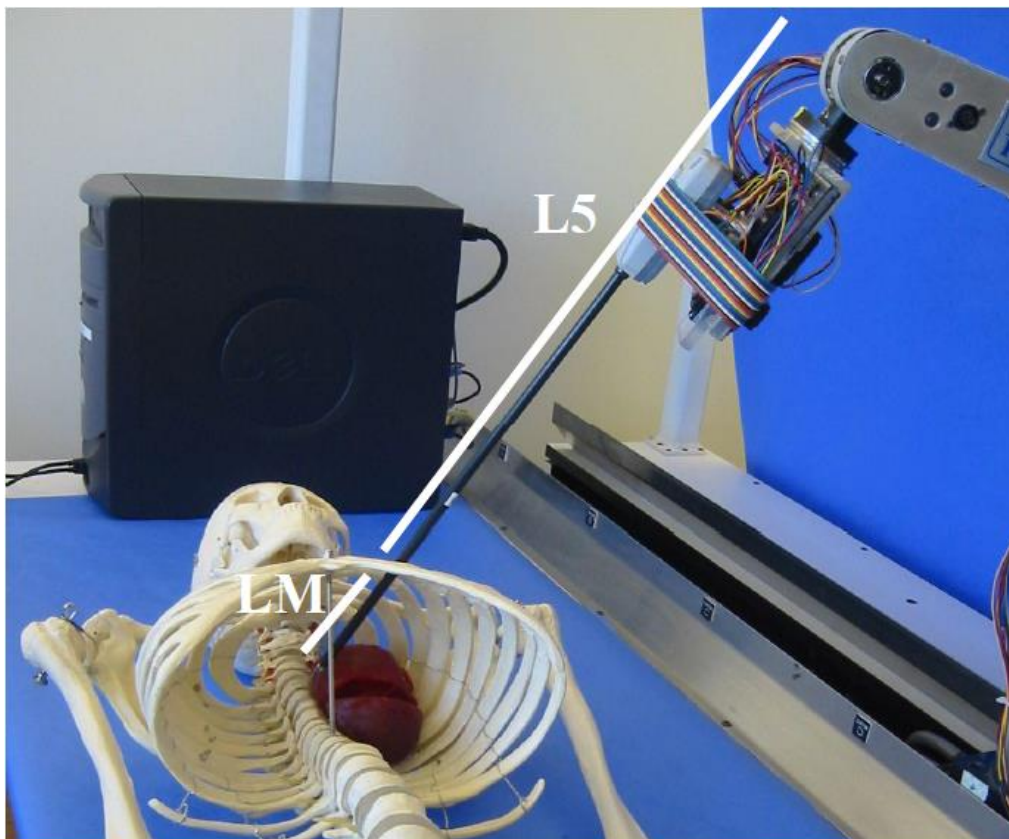

Figure 4. Endoscopic Tools Robotic Manipulator

System Kinematics: The study between rigid bodies and end-effector without any forces is called Robot manipulator Kinematics. Study of this part is very important to design controller and in practical applications. The study of motion without regard to the forces (manipulator kinematics) is divided into two main subjects: forward and inverse

kinematics. Forward kinematics is a transformation matrix to calculate the relationship between position and orientation (pose) of task (end-effector) frame and joint variables. This part is very important to calculate the position and/or orientation error to calculate the controller's qualify. Forward kinematics matrix is a 4×4 matrix which 9 cells are show the orientation of end-effector, 3 cells show the position of end-effector and 4 cells are fix scaling factors. Inverse kinematics is a type of transformation functions that can used to find possible joints variable (displacements and/or angles) when all position and orientation (pose) of task be clear [3]. Figure 5 shows the application of forward and inverse kinematics.
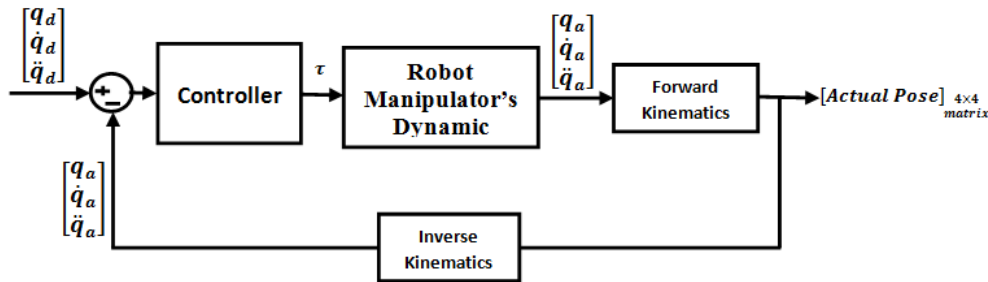

Figure 5. The Application of Forward and Inverse Kinematics

In this research to forward kinematics is used to system modeling. The main target in forward kinematics is calculating the following function:

$$\Psi(X, q) = 0 \tag{1}$$

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.

Calculate robot manipulator forward kinematics is divided into four steps as follows;

- Link descriptions
- Denavit-Hartenberg (D-H) convention table
- Frame attachment
- Forward kinematics

The first step to analyze forward kinematics is link descriptions. This item must to describe and analyze four link and joint parameters. The link description parameters are; link length (a_i), twist angle (α_i), link offset (d_i) and joint angle (θ_i). Where link twist, is the angle between Z_i and Z_{i+1} about an X_i, link length, is the distance between Z_i and Z_{i+1} along X_i and d_i, offset, is the distance between X_{i-1} and X_i along Z_i axis. In these four parameters three of them are fixed and one of parameters is variable. If system has rotational joint, joint angle (θ_i) is variable and if it has prismatic joint, link offset (d_i) is variable.

The second step to compute Forward Kinematics (F.K) of robot manipulator is finding the standard D-H parameters. The Denavit-Hartenberg (D-H) convention is a method of drawing robot manipulators free body diagrams. Denvit-Hartenberg (D-H) convention study is compulsory to calculate forward kinematics in robot manipulator. Table 1 shows the standard D-H parameters for N-DOF robot manipulator.

Table 1. The Denavit Hartenberg Parameter

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

The third step to compute Forward kinematics for robot manipulator is finding the frame attachment matrix. The rotation matrix from $\{F_i\}$ to $\{F_{i-1}\}$ is given by the following equation;

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

Where $U_{i(\theta_i)}$ is given by the following equation [3];

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

and $V_{i(\alpha_i)}$ is given by the following equation [3];

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

So (R_n^0) is given by [3]

$$R_n^0 = (U_1 V_1)(U_2 V_2) \dots \dots \dots (U_n V_n) \tag{5}$$

$$^{n-1}_nT = \begin{bmatrix} R_n^{n-1} & d_n^{n-1} \\ 0 & 1 \end{bmatrix} \tag{6}$$

The transformation ^0_nT (frame attachment) matrix is compute as the following formulation;

$$^{i-1}_1T = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i\,C\alpha_{i-1} & C\theta_i\,C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}d_i \\ S\theta_i\,S\alpha_{i-1} & C\theta_i\,S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The forth step is calculate the forward kinematics by the following formulation [3]

$$FK = ^0_nT = ^0_1T \cdot ^1_2T \cdot ^2_3T \dots \dots ^{n-1}_nT = \begin{bmatrix} R_n^0 & d_n^0 \\ 0 & 1 \end{bmatrix} \tag{8}$$

Based on above formulation the final formulation for 4-DOF surgical robot manipulator is;

$$
{}^{0}_{4}T = \begin{bmatrix} N_x & B_x & T_x & P_x \\ N_y & B_y & T_y & P_y \\ N_z & B_z & T_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}
$$

Table 2 shows the 4-DOF surgical robot manipulator Denavit-Hartenberg notations.

Table 2. 4-DOF Surgical Robot Manipulator D-H Notations

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

Based on frame attachment matrix the position and orientation (pose) matrix compute as bellows;

$$
Nx = cos(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) + sin(\theta_1) \times sin(\theta_1)) + sin(\theta_2 + \theta_3) \times cos(\theta_1)) + (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) - cos(\theta_4) \times sin(\theta_1)) \tag{10}
$$

$$
Ny = (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) + sin(\theta_2 + \theta_3) \times sin(\theta_1)) + (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + cos(\theta_4) \times cos(\theta_1)) \tag{11}
$$

$$
Nz = cos(\theta_4) \times sin(\theta_2 + \theta_3) - cos(\theta_2 + \theta_3)) + sin(\theta_4) \times sin(\theta_2 + \theta_3) \tag{12}
$$

$$
Bx = - (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) + sin(\theta_4) \times sin(\theta_1)) + sin(\theta_2 + \theta_3) \times cos(\theta_1)) + (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) - cos(\theta_4) \times sin(\theta_1)) \tag{13}
$$

$$
By = -(cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) + sin(\theta_2 + \theta_3) \times sin(\theta_1)) + (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + cos(\theta_4) \times cos(\theta_1)) \tag{14}
$$

$$
Bz = -cos(\theta_4) \times sin(\theta_2 + \theta_3) - cos(\theta_2 + \theta_3)) + sin(\theta_4) \times sin(\theta_2 + \theta_3) \tag{15}
$$

$$
Tx = (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times cos(\theta_1) + sin(\theta_4) \times sin(\theta_1)) - sin(\theta_2 + \theta_3) \times cos(\theta_1) \tag{16}
$$

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

$$Tz = cos(\theta_4) \times sin(\theta_2 + \theta_3) + cos(\theta_5) \times cos(\theta_2 + \theta_3) \quad (18)$$

$$Px = 0.4331 \times sin(\theta_2 + \theta_3) \times cos(\theta_1) + 0.0203 \times cos(\theta_2 + \theta_3) \times \quad (19)$$
$$cos(\theta_1) - 0.1491 \times sin(\theta_1) + 0.4318 \times cos(\theta_2 cos(\theta_1))$$

$$Py = 0.4331 \times sin(\theta_2 + \theta_3) \times sin(\theta_1) + 0.0203 \times cos(\theta_2 + \theta_3) \times \quad (20)$$
$$sin(\theta_1) + 0.1491 \times cos(\theta_1) + 0.4312 \times cos(\theta_2) \times sin(\theta_1)$$

$$Pz = -0.4331 \times cos(\theta_2 + \theta_3) + 0.0203 \times sin(\theta_2 + \theta_3) + 0.4318 \quad (21)$$
$$\times sin(\theta_2)$$

This part focuses on modeling robot kinematics. To modeling system kinematics using MATLAB/SIMULINK following steps are introduced:

- An Embedded MATLAB™ Function block lets you compose a MATLAB function within a Simulink model like the following example (Figure 6):

```
1    function [mean, stdev] = stats(vals)
2
3    % calculates a statistical mean and a standard
4    % deviation for the values in vals.
5
6 -  eml.extrinsic('plot');
7
8 -  len = length(vals);
9 -  mean = avg(vals,len);
10 - stdev = sqrt(sum(((vals-avg(vals,len)).^2))/len);
11 - plot(vals, '-+');
12
13   function mean = avg(array, size)
14 - mean = sum(array)/size;
```

Figure 6. Embedded MATLAB Function

- This capability is useful for coding algorithms that are better stated in the textual language of the MATLAB software than in the graphical language of the Simulink product. This block works with a subset of the MATLAB language called the Embedded MATLAB subset, which provides optimizations for generating efficient, production-quality C code for embedded applications. Figure 7 shows the Kinematics Embedded MATLAB function.

Figure 7. Kinematics Embedded MATLAB Function

System's Dynamic: A dynamic function is the study of motion with regard to the forces. Dynamic modeling of surgical robot manipulators is used to illustrate the behavior of robot manipulator (*e.g.,* nonlinear dynamic behavior), design of nonlinear conventional controller and for simulation. It is used to analyses the relationship between dynamic functions output (*e.g.,* joint motion, velocity, and accelerations) to input source of dynamic functions (*e.g.,* force/torque or current/voltage). Dynamic functions is also used to explain the some dynamic parameter's effect (*e.g.,* inertial matrix, Coriolios, Centrifugal, and some other parameters) to system's behavior [3].

The equation of multi degrees of freedom (DOF) surgical robot manipulator dynamics is considered by the following equation [7]:

$$[A(q)]\ddot{q} + [N(q, \dot{q})] = [\tau] \tag{22}$$

Where τ is actuator's torque and is $n \times 1$ vector, A (q) is positive define inertia and is $n \times n$ symmetric matrix based on the following formulation;

$$A(q) = \begin{bmatrix} A_{11} & A_{12} & \dots & \dots & \dots & A_{1n} \\ A_{21} & \dots & \dots & \dots & \dots & A_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ A_{n.1} & \dots & \dots & \dots & \dots & A_{n.n} \end{bmatrix} \tag{23}$$

$N(q, \dot{q})$ is the vector of nonlinearity term, and q is $n \times 1$ joints variables. If all joints are revolute, the joint variables are angle (θ) and if these joints are translated, the joint variables are translating position (d). The nonlinearity term of robot manipulator is derived as three main parts; Coriolis $b(q)$, Centrifugal $C(q)$, and Gravity $G(q)$. Consequently the robot manipulator dynamic equation can also be written as [8]:

$$[N(q, \dot{q})] = [V(q, \dot{q})] + [G(q)] \tag{24}$$

$$[V(q, \dot{q})] = [b(q)][\dot{q} \, \dot{q}] + [C(q)][\dot{q}]^2 \tag{25}$$

$$\tau = A(q)\ddot{q} + b(q)[\dot{q}\,\dot{q}] + C(q)[\dot{q}]^2 + G(q) \tag{26}$$

Where,

$b(q)$ is a Coriolis torque matrix and is $n \times \frac{n \times (n-1)}{2}$ matrix, $C(q)$ is Centrifugal torque matrix and is $n \times n$ matrix, Gravity is the force of gravity and is $n \times 1$ matrix, $[\dot{q}\,\dot{q}]$ 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: $[\dot{q}_1{}^2, \dot{q}_2{}^2, \dot{q}_3{}^2, \ldots.]^T$. According to the basic information from system's modelling, all functions are derived as the following form;

$$Outputs = function\ (inputs) \tag{27}$$

In the dynamic formulation of robot manipulator the inputs are torques matrix and the outputs are actual joint variables,

$$q = function\ (\tau) \tag{28}$$

$$\ddot{q} = A^{-1}(q).\{\tau - N(q,\dot{q})\} \tag{29}$$

$$q = \iint A^{-1}(q).\{\tau - N(q,\dot{q})\} \tag{30}$$

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

$$b(q) = \begin{bmatrix} b_{112} & b_{113} & \ldots & b_{11n} & b_{123} & \ldots & b_{12n} & \ldots & \ldots & b_{1.n-1.n} \\ b_{212} & \ldots & \ldots & b_{21n} & b_{223} & \ldots & \ldots & \ldots & \ldots & b_{2.n-1.n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ b_{n.1.2} & \ldots & \ldots & b_{n.1.n} & \ldots & \ldots & \ldots & \ldots & \ldots & b_{n.n-1.n} \end{bmatrix} \tag{31}$$

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

$$C(q) = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{bmatrix} \tag{32}$$

The Gravity vector (G) is a $n \times 1$ vector;

$$G(q) = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \tag{33}$$

The dynamic formulations for 4 Degrees of Freedom serial links surgical robot manipulator are computed by;

$$A(\ddot{\theta})\begin{bmatrix}\ddot{\theta}_1\\\ddot{\theta}_2\\\ddot{\theta}_3\\\ddot{\theta}_4\end{bmatrix} + B(\theta)\begin{bmatrix}\dot{\theta}_1\dot{\theta}_2\\\dot{\theta}_1\dot{\theta}_3\\\dot{\theta}_1\dot{\theta}_4\\\dot{\theta}_2\dot{\theta}_3\\\dot{\theta}_2\dot{\theta}_4\\\dot{\theta}_3\dot{\theta}_4\end{bmatrix} + C(\theta)\begin{bmatrix}\dot{\theta}_1^2\\\dot{\theta}_2^2\\\dot{\theta}_3^2\\\dot{\theta}_4^2\end{bmatrix} + G(\theta) = \begin{bmatrix}\tau_1\\\tau_2\\\tau_3\\\tau_4\end{bmatrix}$$

(34)

Where

$$A(q) = \begin{bmatrix}A_{11} & A_{12} & A_{13} & 0\\A_{21} & A_{22} & A_{23} & 0\\A_{31} & A_{32} & A_{33} & 0\\0 & 0 & 0 & A_{44}\end{bmatrix}$$

(35)

According to [8] the inertial matrix elements (A) are

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

(36)

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

(37)

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

(38)

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

(39)

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

(40)

$$A_{33} = I_{m3} + I_6 + 2I_{15}$$

(41)

$$A_{44} = I_{m4} + I_{14}$$

(42)

$$A_{21} = A_{12}, A_{31} = A_{13} \text{ and } A_{32} = A_{23}$$

(43)

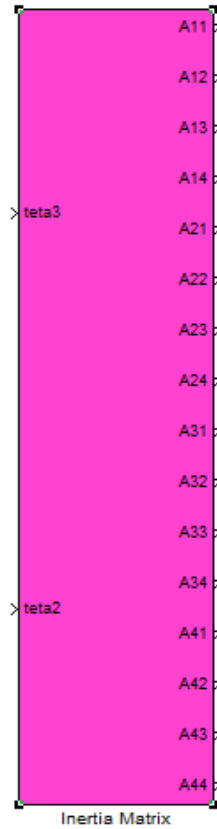Figure 8 shows the kinetics energy in MATLAB/SIMULINK.

Figure 8. Kinetics Energy in MATLAB/SIMULINK

The Corilios (b) matrix elements are;

$$b(q) = \begin{bmatrix} b_{112} & b_{113} & 0 & b_{123} \\ 0 & 0 & b_{214} & b_{223} \\ 0 & 0 & b_{314} & 0 \\ b_{412} & b_{413} & 0 & 0 \end{bmatrix} \tag{44}$$

Where,

$$b_{112} = 2\big[- I_3\sin(\theta_2)\cos(\theta_2) + I_5\cos(\theta_2 + \theta_2 + \theta_3) + I_7\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{12}\sin(\theta_2 + \theta_2 + \theta_3) - I_{15}2\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{16}\cos(\theta_2 + \theta_2 + \theta_3) + I_{21}\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{22}\big(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)\big)\big] + I_{10}\big(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)\big) + I_{11}\big(1 - 2\sin(\theta_2)\sin(\theta_2)\big) \tag{45}$$

$$b_{113} = 2\big[I_5\cos(\theta_2)\cos(\theta_2 + \theta_3) + I_7\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{12}\cos(\theta_2)\sin(\theta_2 + \theta_2) + I_{15}2\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{16}\cos(\theta_2)\cos(\theta_2 + \theta_3) + I_{21}\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{22}\big(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)\big)\big] + I_{10}\big(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)\big) \tag{46}$$

$$b_{123} = 2[-I_8\sin(\theta_2 + \theta_3) + I_{13}\cos(\theta_2 + \theta_3) + I_{18}\sin(\theta_2 + \theta_3)] \tag{47}$$

$$b_{214} = I_{14}\sin(\theta_2 + \theta_3) + I_{19}\sin(\theta_2 + \theta_3) + 2I_{20}\sin(\theta_2 + \theta_3)(1 - 0.5) \qquad (48)$$

$$b_{223} = 2[-I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3)] \qquad (49)$$

$$b_{314} = 2[I_{20}\sin(\theta_2 + \theta_3)(1 - 0.5)] + I_{14}\sin(\theta_2 + \theta_3) + I_{19}\sin(\theta_2 + \theta_3) \qquad (50)$$

$$b_{412} = b_{214} = -[I_{14}\sin(\theta_2 + \theta_3) + I_{19}\sin(\theta_2 + \theta_3) + 2I_{20}\sin(\theta_2 + \theta_3)(1 - 0.5)] \qquad (51)$$

$$b_{413} = -b_{314} = -2[I_{20}\sin(\theta_2 + \theta_3)(1 - 0.5)] + I_{14}\sin(\theta_2 + \theta_3) + I_{19}\sin(\theta_2 + \theta_3) \qquad (52)$$

Based on above discussion $[b(q)]$ is 4×6 matrix and $[\dot{q}\dot{q}]$ is 6×1, therefore $[b(q).\dot{q}\dot{q}]$ is 4×1.

$$[b(q).\dot{q}\dot{q}]_{4\times1} = \begin{bmatrix} b_{112}.\dot{q}_1\dot{q}_2 + b_{113}.\dot{q}_1\dot{q}_3 + 0 + b_{123}.\dot{q}_2\dot{q}_3 \\ 0 + b_{214}.\dot{q}_1\dot{q}_4 + b_{223}.\dot{q}_2\dot{q}_3 \\ b_{314}.\dot{q}_1\dot{q}_4 \\ b_{412}.\dot{q}_1\dot{q}_2 + b_{413}.\dot{q}_1\dot{q}_3 \end{bmatrix} \qquad (53)$$

Figure 9 and 10 show the Coriolis Embedded MATLAB function.
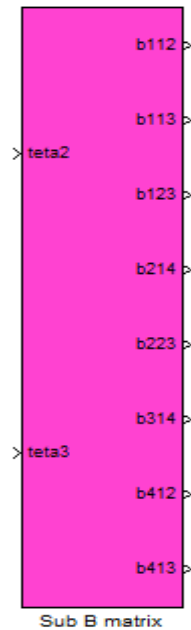
Figure 9. Coriolis Embedded MATLAB Function
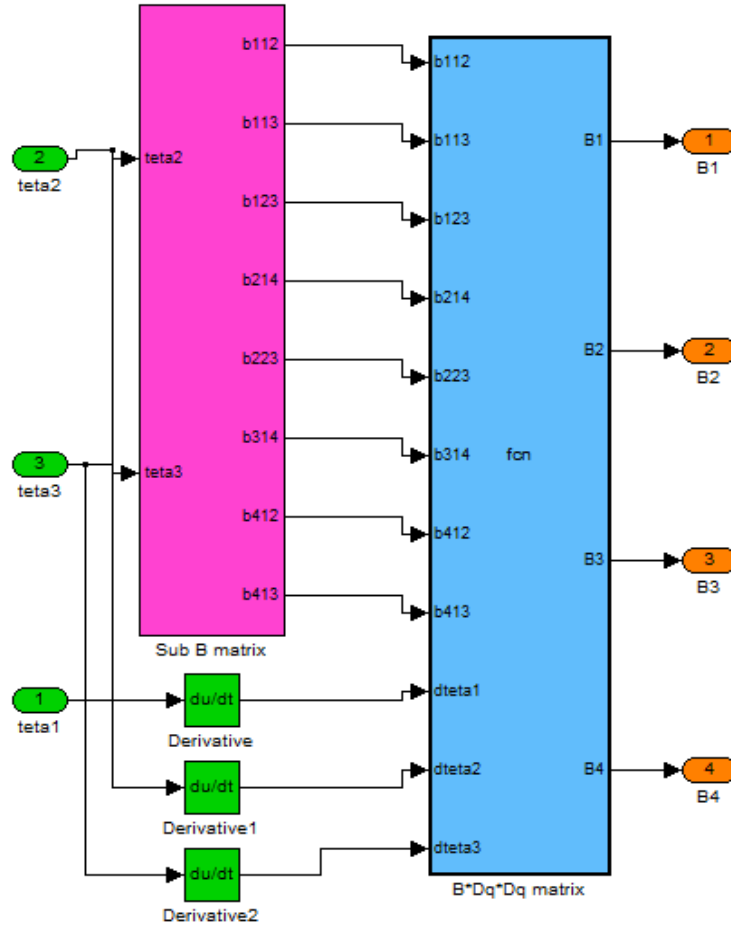
Figure 10. $b(q)[\dot{q}\ \dot{q}]$ **Embedded MATLAB Function**

According to [8] Centrifugal (C) matrix elements are;

$$C(q) = \begin{bmatrix} 0 & C_{12} & C_{13} & 0 \\ C_{21} & C_{22} & C_{23} & 0 \\ C_{31} & C_{32} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{54}$$

Where,

$$c_{12} = I_4\cos(\theta_2) - I_8\sin(\theta_2 + \theta_3) - I_9\sin(\theta_2) + I_{13}\cos(\theta_2 + \theta_3) + I_{18}\sin(\theta_2 + \theta_3) \tag{55}$$

$$c_{13} = 0.5b_{123} = -I_8\sin(\theta_2 + \theta_3) + I_{13}\cos(\theta_2 + \theta_3) + I_{18}\sin(\theta_2 + \theta_3) \tag{56}$$

$$c_{21} = -0.5b_{112} = I_3\sin(\theta_2)\cos(\theta_2) - I_5\cos(\theta_2 + \theta_2 + \theta_3) - I_7\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{12}\sin(\theta_2 + \theta_2 + \theta_3) + I_{15}2\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{16}\cos(\theta_2 + \theta_2 + \theta_3) - I_{21}\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{22}\big(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)\big) - 0.5I_{10}\big(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)\big) - 0.5I_{11}\big(1 - 2\sin(\theta_2)\sin(\theta_2)\big) \tag{57}$$

$$c_{22} = 0.5b_{223} = -I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3) \tag{58}$$

$$c_{23} = -0.5b_{113} = -I_5\cos(\theta_2)\cos(\theta_2 + \theta_3) - I_7\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{12}\cos(\theta_2)\sin(\theta_2 + \theta_2) - I_{15}2\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{16}\cos(\theta_2)\cos(\theta_2 + \theta_3) - I_{21}\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{22}(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)) - 0.5I_{10}(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)) \tag{59}$$

$$c_{31} = -c_{23} = I_{12}\sin(\theta_3) - I_5\cos(\theta_3) - I_{16}\cos(\theta_3) \tag{60}$$

$$c_{32} = \sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{15}2\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{16}\cos(\theta_2)\cos(\theta_2 + \theta_3) - I_{22}\cos(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) \tag{61}$$

$$[C(q).\dot{q}^2]_{6\times1} = \begin{bmatrix} c_{12}.\dot{q}_2^2 + c_{13}.\dot{q}_3^2 \\ c_{21}.\dot{q}_1^2 + c_{23}.\dot{q}_3^2 \\ c_{13}.\dot{q}_1^2 + c_{32}.\dot{q}_2^2 \\ 0 \end{bmatrix} \tag{62}$$

Figure 11 and 12 show the Centrifugal Embedded MATLAB function.
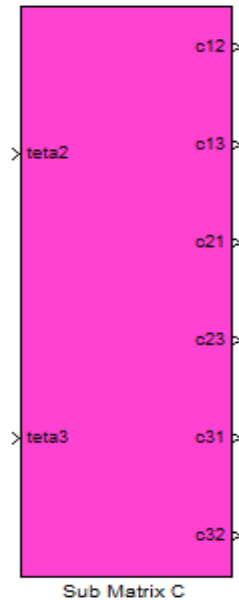
Figure 11. Centrifugul Embedded MATLAB Function
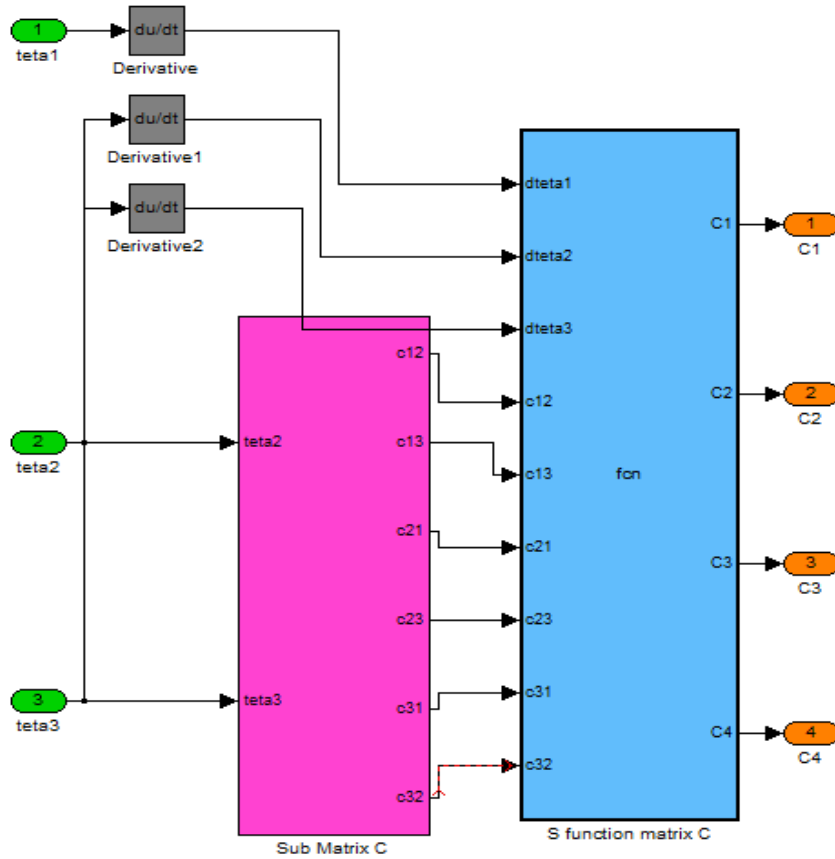
Figure 12. $\left[C(q).\dot{q}^2 \right]$ **Embedded MATLAB Function**

Gravity ($\boldsymbol{G}$) Matrix elements are [8];

$$[\boldsymbol{G}(\boldsymbol{q})]_{4 \times 1} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{G_2} \\ \boldsymbol{G_3} \\ \mathbf{0} \end{bmatrix} \tag{63}$$

Where,

$$\boldsymbol{G_2} = \boldsymbol{g_1}\cos(\boldsymbol{\theta_2}) + \boldsymbol{g_2}\sin(\boldsymbol{\theta_2} + \boldsymbol{\theta_3}) + \boldsymbol{g_3}\sin(\boldsymbol{\theta_2}) + \boldsymbol{g_4}\cos(\boldsymbol{\theta_2} + \boldsymbol{\theta_3}) + \boldsymbol{g_5}\sin(\boldsymbol{\theta_2} + \boldsymbol{\theta_3}) \tag{64}$$

$$\boldsymbol{G_3} = \boldsymbol{g_2}\sin(\boldsymbol{\theta_2} + \boldsymbol{\theta_3}) + \boldsymbol{g_4}\cos(\boldsymbol{\theta_2} + \boldsymbol{\theta_3}) + \boldsymbol{g_5}\sin(\boldsymbol{\theta_2} + \boldsymbol{\theta_3}) \tag{65}$$

Figure 13 shows the Gravity Embedded MATLAB function.
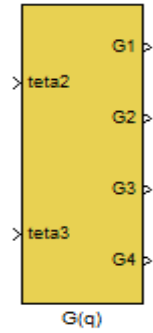
Figure 13. Gravity Embedded MATLAB function

If $[I]_{4 \times 1} = [B]_{4 \times 1} + [C]_{4 \times 1} + [G]_{4 \times 1}$

Figure 14 shows the $[I]_{4 \times 1}$ Embedded MATLAB function.
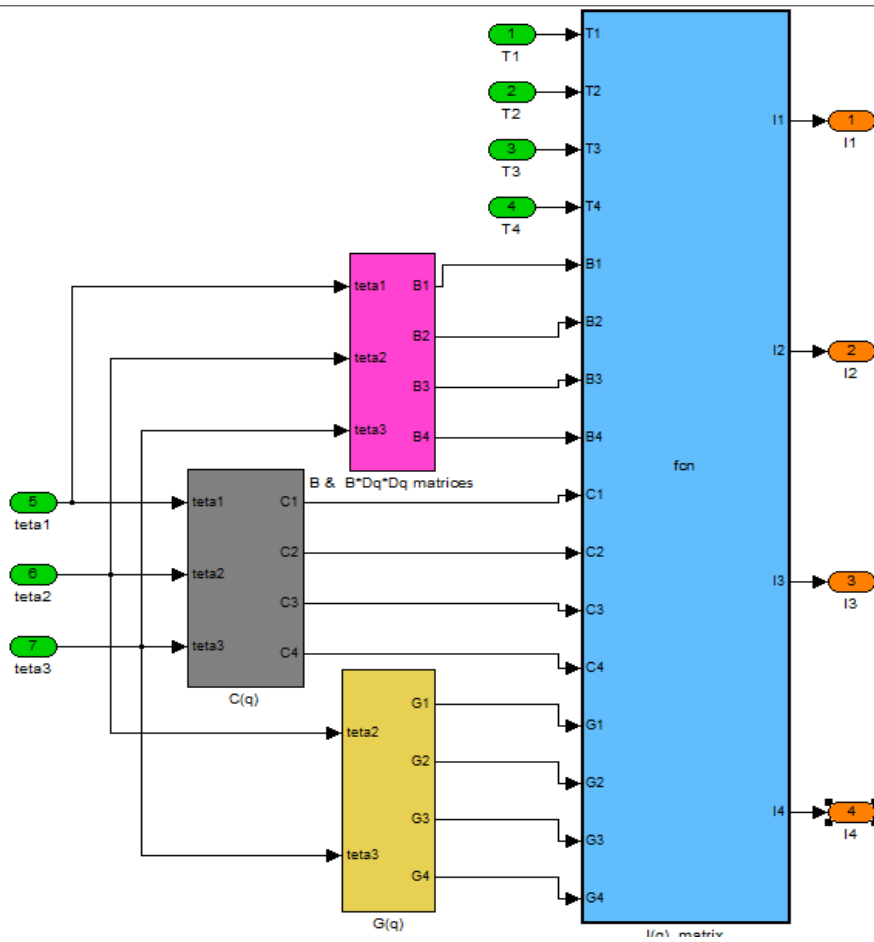
Figure 14. $[I]_{4 \times 1}$ Embedded MATLAB Function

Then $\ddot{q}$ is written as follows;

$$[\ddot{q}]_{4 \times 1} = [A^{-1}(q)]_{4 \times 4} \times \{[\tau]_{4 \times 1} - [I]_{4 \times 1}\} \tag{66}$$

K is presented as follows;

$$[K]_{4\times1} = \{[\tau]_{4\times1} - [I]_{4\times1}\} \tag{67}$$

$$[\ddot{q}]_{4\times1} = [A^{-1}(q)]_{4\times4} \times [K]_{4\times1} \tag{68}$$

$$[q]_{4\times1} = \iint [A^{-1}(q)]_{4\times4} \times [K]_{4\times1} \tag{69}$$

Basic information about inertial and gravitational constants is show in Tables 3 and 4

Table 3. Inertial Constant Reference ($Kg.m^2$)

$I_1 = 1.43 \pm 0.05$	$I_2 = 1.75 \pm 0.07$
$I_3 = 1.38 \pm 0.05$	$I_4 = 0.69 \pm 0.02$
$I_5 = 0.372 \pm 0.031$	$I_6 = 0.333 \pm 0.016$
$I_7 = 0.298 \pm 0.029$	$I_8 = -0.134 \pm 0.014$
$I_9 = 0.0238 \pm 0.012$	$I_{10} = -0.0213 \pm 0.0022$
$I_{11} = -0.0142 \pm 0.0070$	$I_{12} = -0.011 \pm 0.0011$
$I_{13} = -0.00379 \pm 0.0009$	$I_{14} = 0.00164 \pm 0.000070$
$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$I_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$I_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.000015$
$I_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

Table 4. Gravitational Constant ($N.m$)

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

Figure 15 shows the system's dynamic Embedded MATLAB function.
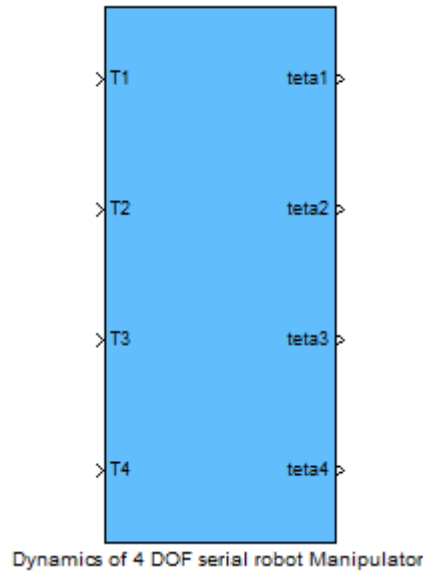
Figure 15. System's Dynamic Embedded MATLAB Function

3. Design and Modeling PID Control Technique for Surgical Robot

Linear control theory is used in linear and nonlinear systems. This type of theory is used in industries, because design of this type of controller is simple than nonlinear controller. However this type of controller used in many applications but it cannot guarantee performance in complex systems. Simple linear controllers are including:

Proportional (P) Control: It is used to responds immediately to difference of control input variables by immediately changing its influences variables, but this type of control is unable to eliminate the control input difference. Figure 16 shows the block diagram of proportional controller with application to serial links robot manipulator.
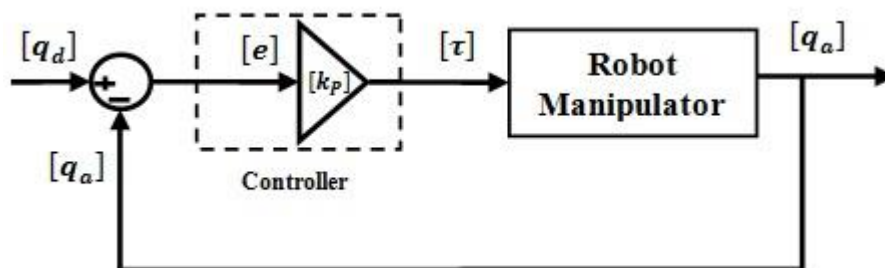
Figure 16. Block Diagram of Proportional Controller

Proportional plus Derivative (PD) control: This type of linear controller is widely used in control process where the results are sensitive to exceeded of set point. This controller, like Proportional controller, has permanent variation in presence of self-limitation control. In mathematically, the formulation of Proportional-Derivative part calculated as follows;

$$U_{PD} = K_p \times e + K_v(\frac{de}{dt}) = K_p \times e + K_v \dot{e} \tag{70}$$

The Derivative component in this type of methodology is used to cancel outs the change process variables change in presence of quick change in controllers input. Figure 17 shows the block diagram of Proportional-Derivative (PD) control of robot manipulator.
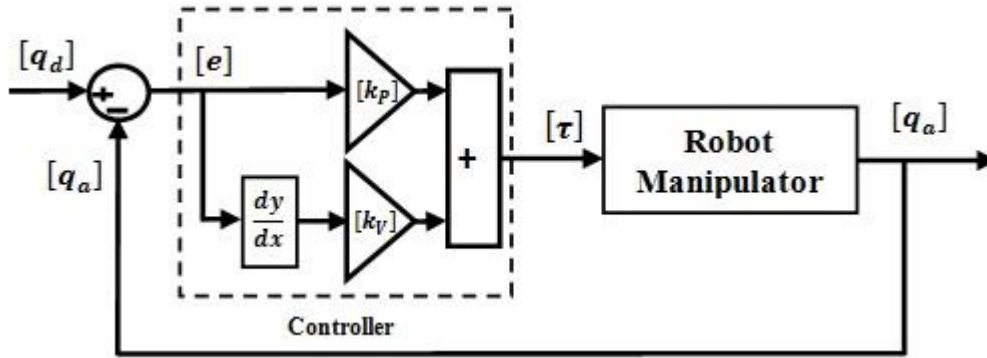
Figure 17. Block Diagram of PD Control of Robot Arm

Figure 18 shows the ramp response of PD controller.

Integral (I) control: this category, integrate the input signal deviation over a period of time. This part of controller is used to system stability after a long period of time. Figure 19 shows the block diagram of Integral (I) controller with application to robot manipulator. In contrast of Proportional type of controller, this type of controller used to eliminate the deviation.
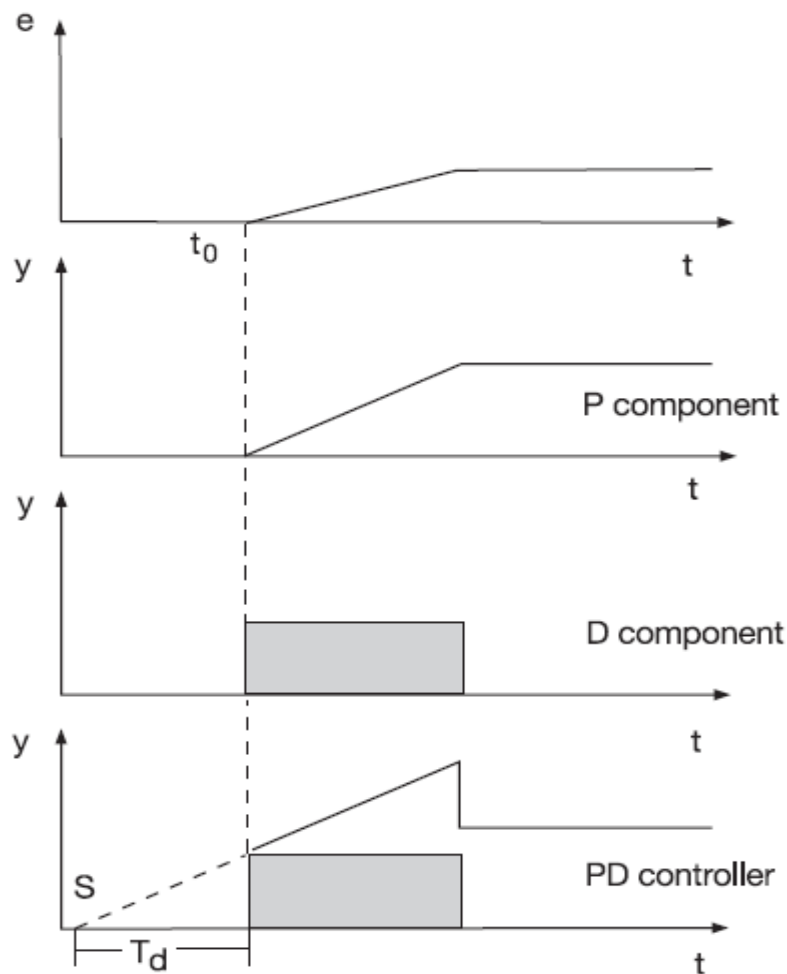

Figure 18. Ramp Response of a PD Controller

In mathematically, the formulation of integral part calculated as follows;

$$I = \frac{1}{T} \int e \cdot dt = \sum e \tag{71}$$
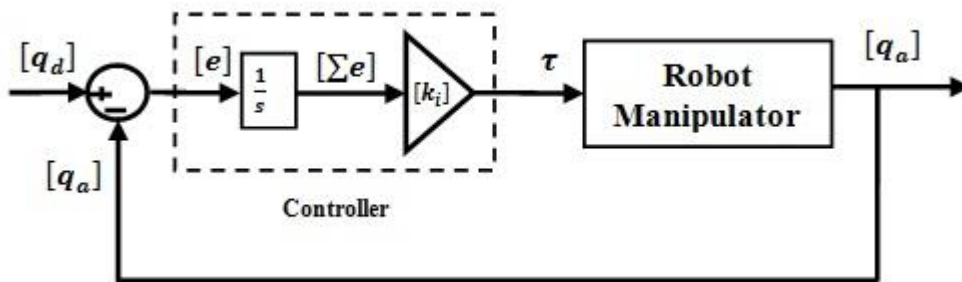
Figure 19. Block Diagram of Integral Control of Robot Manipulator

Figure 20 shows the step response of integral controller.
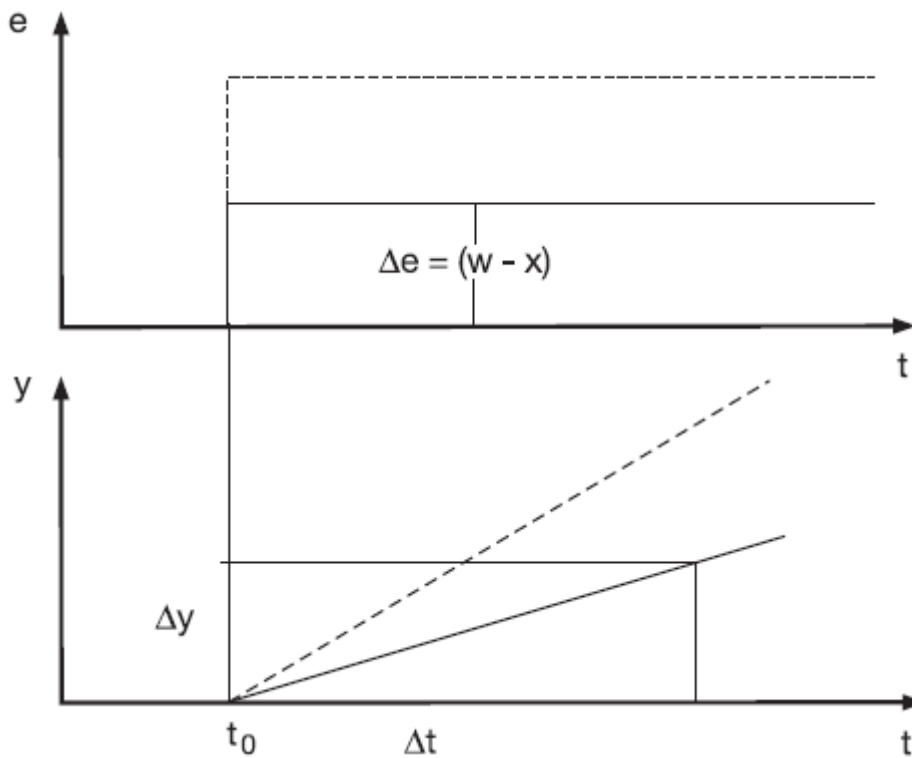
Figure 20. Step Response of an Integral (I) Controller

Proportional plus Integral (PI) control: According to integral type of controller, it takes relatively long time. The proportional type controller used to immediately response to the input variations. The proportional-integral (PI) controller has the advantages of both proportional and integral controller; it is rapid response to the input deviation as well as the exact control at the desired input. Figure 21 shows the block diagram of PI control of robot manipulator.

$$U_{PI} = K_p \times e + K_i \left(\frac{1}{T} \int e \cdot dt \right) = K_p \times e + K_i \sum e \tag{72}$$
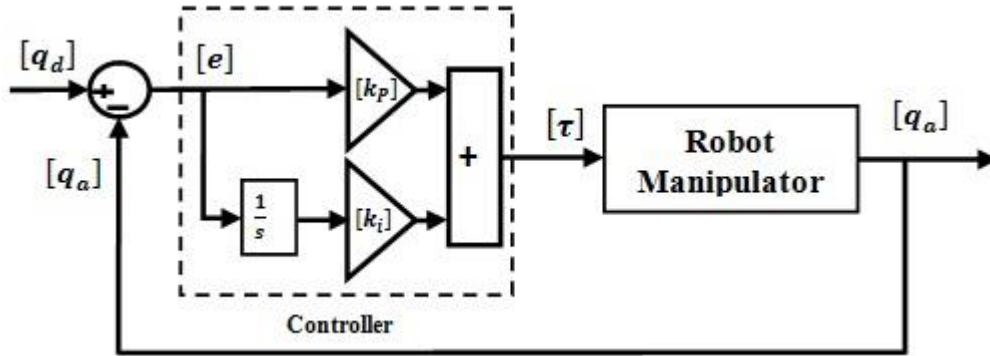
Figure 21. Block Diagram of PI Control of Robot Manipulator

Figure 22 shows the step response of PI controller.

Proportional-Integral-Derivative (PID) control: The combination of proportional (P) component, integral (I) component with a derivative (D) controller offered advantages in each case. This type of controller has rapid response to the input deviation, the exact control at the desired input as well as fast response to the disturbances. The PID controller takes the error between the desired joint variables and the actual joint variables to control the serial links robot manipulator. A proportional-derivative integral control system can easily be implemented. This method does not provide sufficient control for systems with time-varying parameters or highly nonlinear systems. Figure 23 shows the block diagram of PID control of robot manipulator. The formulation of PID controller calculated as follows;

$$U_{PID} = K_p \times e + K_i(\frac{1}{T}\int e.dt) + K_v(\frac{de}{dt}) = K_p \times e + K_i \sum e + K_v \dot{e} \qquad (73)$$
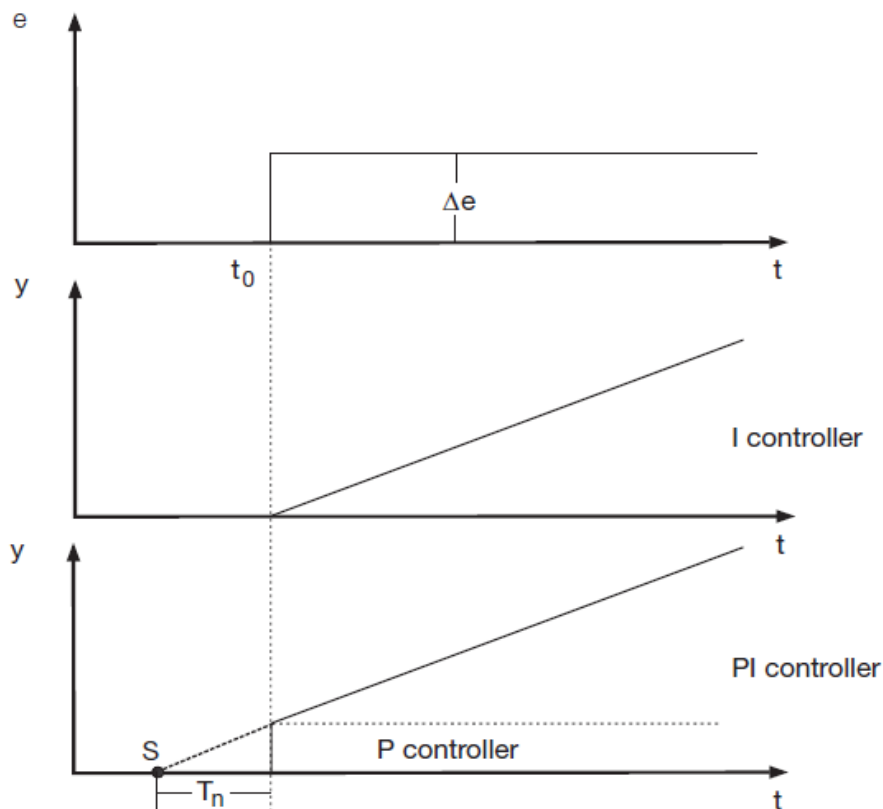

Figure 22. Step Response of a Proportional-Integral (PI) Controller
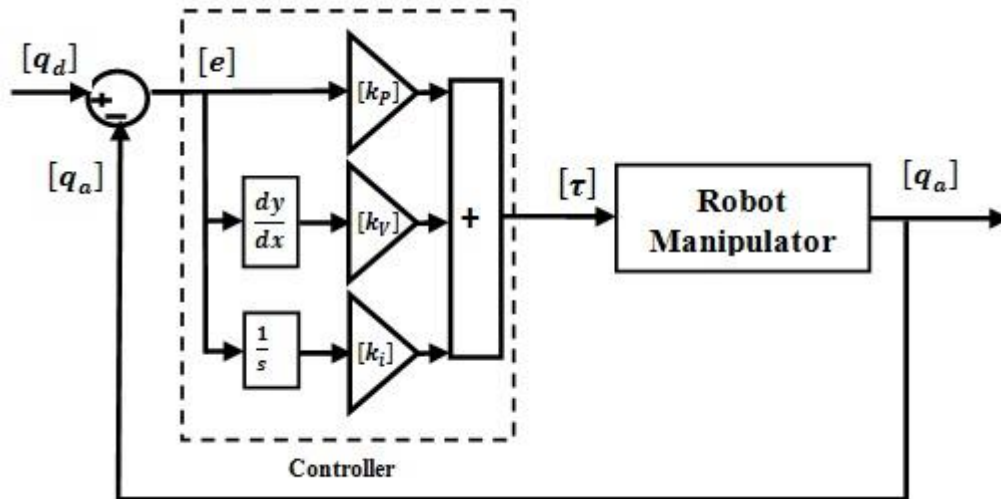
Figure 23. Block Diagram of PID Control of Robot Manipulator

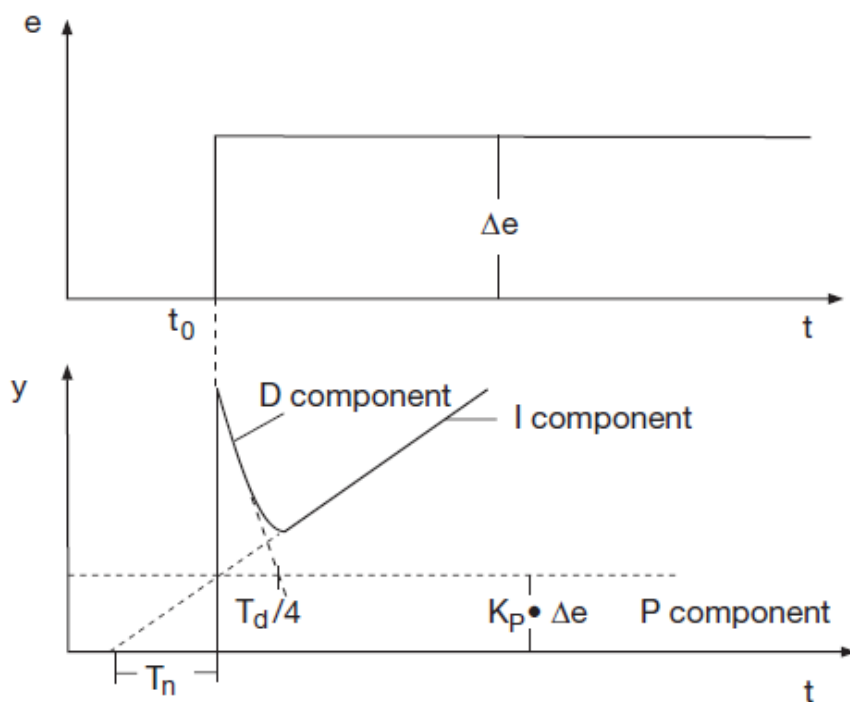Figure 24 shows the step response of PID controller.

Figure 24. Step Response of a Proportional-Integral-Derivative (PID) Controller

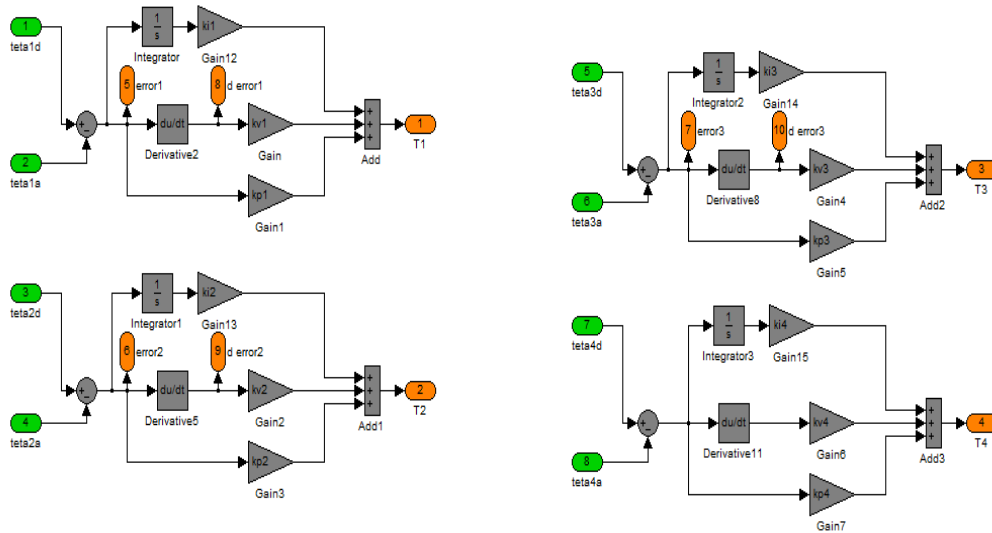Figure 25 and 26 show PID controller by Embedded MATLAB/SIMULINK.

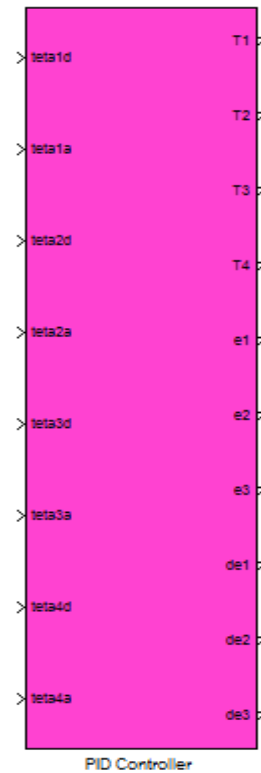**Figure 25. PID controller by Embedded MATLAB/SIMULINK**

Figure 26. Modeling PID Controller

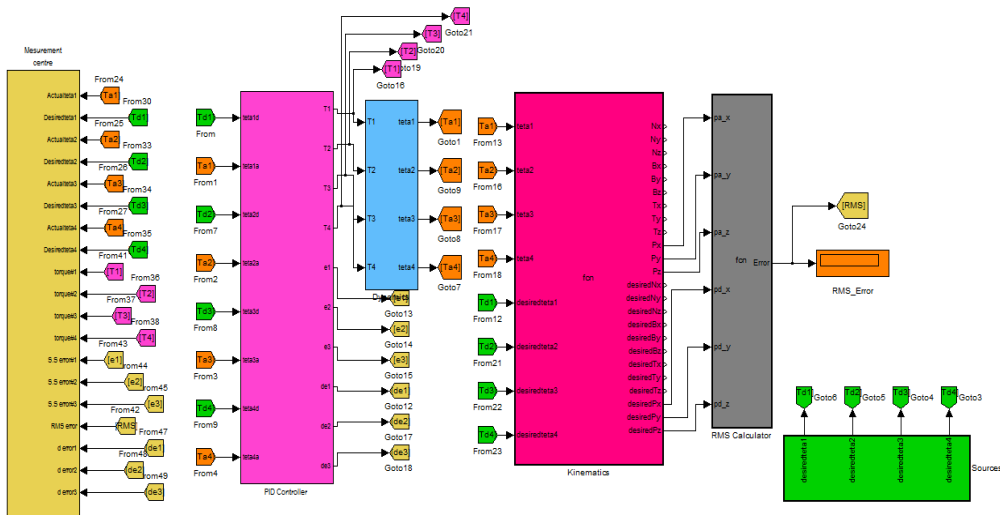Figure 27 shows modeling PID controller by MATLAB/SIMULINK.

Figure 27. Modeling PID Controller for Robot Manipulator by MATLAB/SIMULINK

4.   Results

Comparison of the Tracking Data and Information: the trajectory following of 4 DOF for PID controller is test in this section. According to Figure 28, PID controller has stability in certain condition. However PID controller doesn't have chattering or oscillation but this method has two challenges: robust and quality of performance. In rise time point of view, in some joints PID controller has about 3 second rise time.
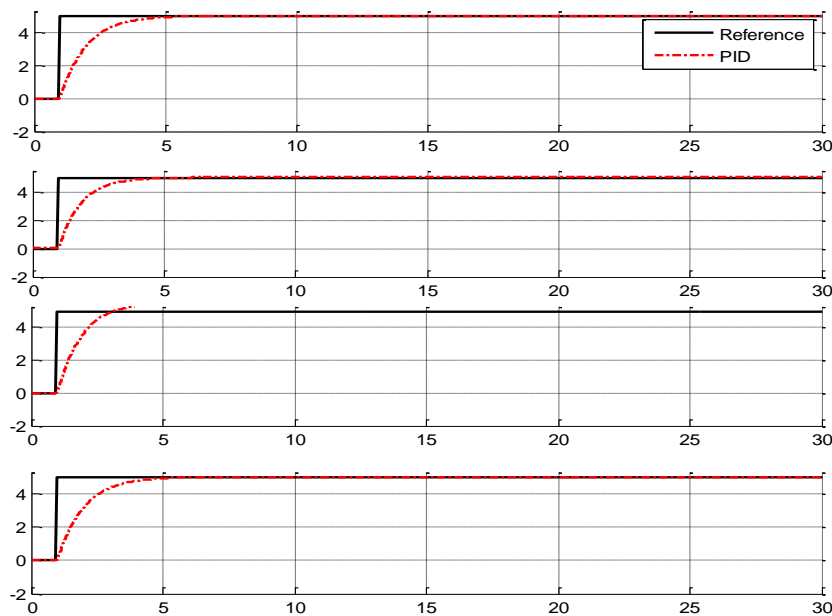

Figure 28. PID Control Trajectory Following

Comparison the Actuation Torque(τ_i): the control input, forces the robotic manipulators to track the desired trajectories. Figure 29 shows the torque performance in

PID controller. According to the following graph, PID controller has steady stable torque performance in second, third and forth joints but it has oscillation in the first joint. In the control forces, smaller amplitude means less energy. According to Figure 29, the amplitude of the control forces in first joint of PID controller is much larger than the other joints.
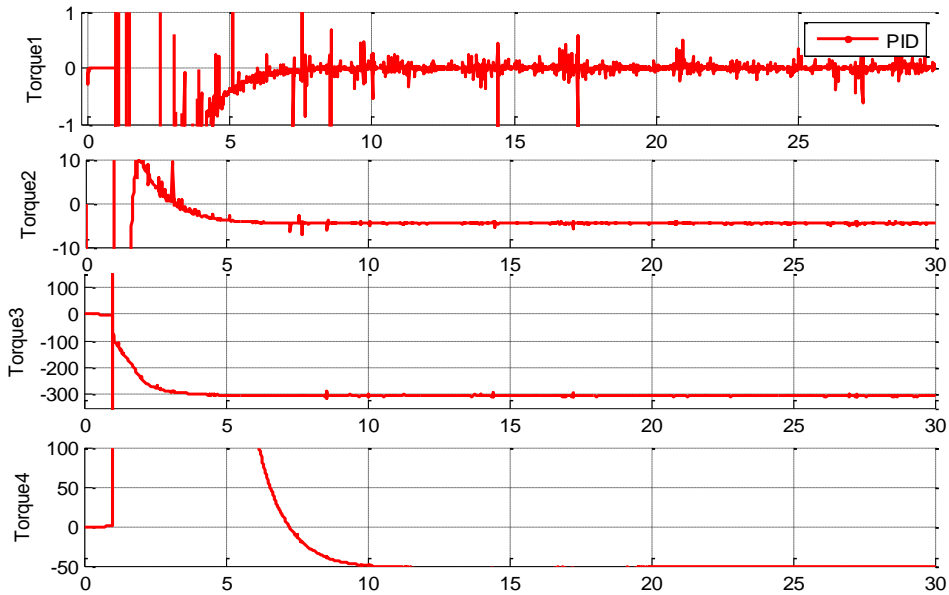

Figure 29. PID Control Torque Performance

Comparison the Disturbance Rejection: the power of disturbance rejection is very important to robust checking in any controllers. In this section trajectory accuracy, and torque performances are test under uncertainty condition. To test the disturbance rejection band limited white noise with 30% amplitude is applied to PID controller. In Figures 30 and 31, trajectory accuracy and torque performance are shown.
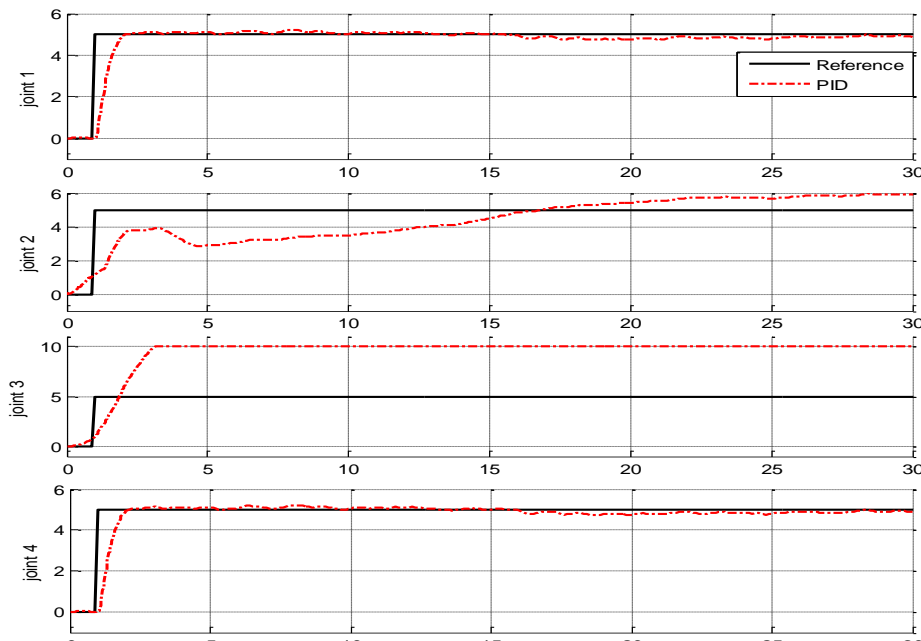

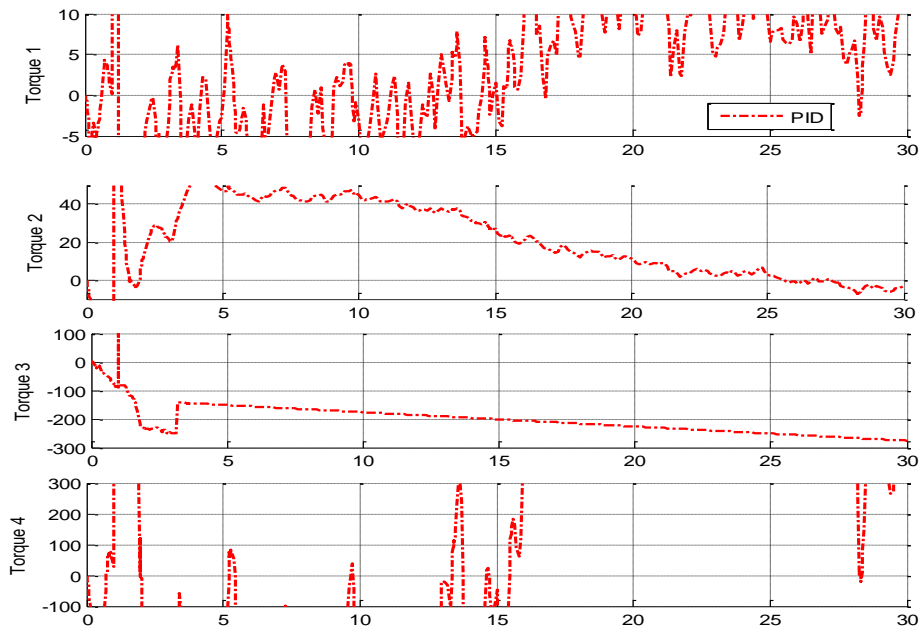**Figure 30. PID Control Trajectory Following in Presence of Uncertainty**

Figure 31. PID Control Torque Performance in Presence of Uncertainty

According to above graphs, PID controller has very many fluctuations in presence of external disturbance. After applied uncertainties the force amplitude in PID is increased which will lead to high-energy consumption.

Tracking Error: in this part, tracking steady state error for all joints is compared. Figure 32 shows the steady state error in presence of uncertainties. According to this Figure, however PID controller is used in many applications but it has steady error PID in presence of uncertainty especially for first, third and forth joints. In the following graph PID controller has irregular fluctuations.
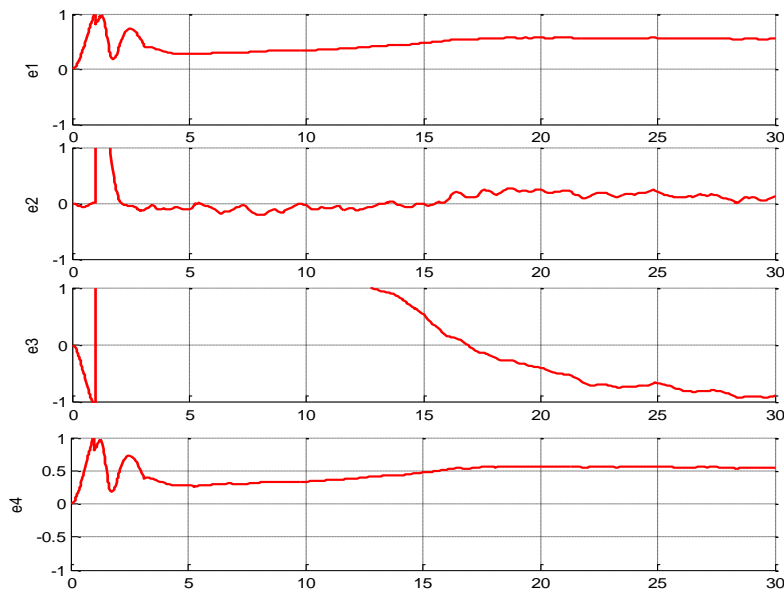

Figure 32. PID Control Error Performance in Presence of Uncertainty

Figure 33 shows root means square (RMS) error in presence of uncertainty for PID controller. Based on Figure 33, PID controller has very position deviations. According to below graph in presence of uncertainties PID controller has steady RMS error.
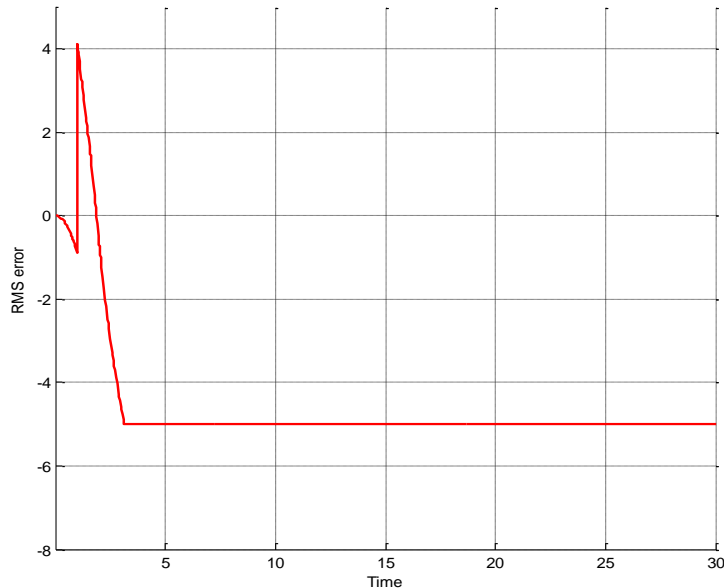

Figure 33. PID Control RMS-error Performance in Presence of Uncertainty

5.   Conclusion

Refer to this research; the Four-axis Virtual Robot arm (FVR) in MATLAB/SIMULINK is modeled based computer program, which can be used to simulate the functions of a real robotic manipulator in terms of design parameters, movement and control. In this research, after extract the dynamic and kinematics formulations, these formulations are model and implemented by MATLAB/SIMULINK for graduate/undergraduate student. To test this system's modeling which is a type of surgical robots, PID controller is candidate. This type of controller is design and after then model in MATLAB/SIMULINK. This controller is test under different situation. This research is a studying paper and used for graduate/undergraduate student as a perfect reference.

References

[1]   L. Sciavicco and B. Siciliano, "Modeling and Control of Robot Manipulators", 2nd ed. London, U.K., Springer-Verlag, **(2000)**, pp. 14-17, 19-23, 26, 45, 49, 59, 60, 67, 99, 100.
[2]   Z. Bingul, "Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization", InTech, **(2012)**, 14, 15, 17, 19, 20, 22, 49, 59, 60, 66, 67.
[3]   T. R. Kurfess, "Robotics and automation handbook", CRC, **(2005)**.
[4]   C. Wu, "Robot accuracy analysis based on kinematics", IEEE Journal of Robotics and Automation, vol. 2, no. 3, **(1986)**, pp. 171-179.
[5]   J. J. E. Slotine and W. Li, "Applied nonlinear control", Prentice hall Englewood Cliffs, NJ, vol. 461, **(1991)**.
[6]   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", IEEE international conference of intelligent control (ISIC), **(2008)**, pp. 189-194.
[7]   B. Siciliano and O. Khatib, Springer handbook of robotics: Springer-Verlag New York Inc, **(2008)**.

[8] B. Armstrong, O. Khatib and J. Burdick, "The explicit dynamic model and inertial parameters of the PUMA 560 arm", IEEE International Conference on Robotica and Automation, **(2002)**, pp. 510-518.

[9] F. Piltan, S. Emamzadeh, Z. Hivand, F. Shahriyari and M. Mirazaei, "PUMA-560 Robot Manipulator Position Sliding Mode Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses", International Journal of Robotics and Automation, vol. 3, no. 3, **(2012)**, pp. 106-150.

[10] F. Piltan, S. Rahmdel, S. Mehrara and R. Bayat, "Sliding Mode Methodology Vs. Computed Torque Methodology Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control Courses", International Journal of Engineering, vol. 6, no. 3, **(2012)**, pp. 142-177.

[11] F. Piltan, M. H. Yarmahmoudi, M. Shamsodini, E. Mazlomian and 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, vol. 3, no. 3, **(2012)**, pp. 167-191.

[12] H. Davarpanah, F. Piltan, S. Jowkar, M. Beheshti and S. Rahbar, "Intelligent Precision Improvement on Robot Assisted Minimally Invasive Direct Coronary Artery Bypass", International Journal of Bio-Science and Bio-Technology, vol. 7, no. 3, **(2015)**, pp. 261-274. http://dx.doi.org/10.14257/ijbsbt.2015.7.3.28, (Scopus, SJR=0.24, Q3).

[13] M. R. Avazpour, F. Piltan, H. Ghiasi, M. H. Mazloom and A. Sahamijoo, "Intelligent Trajectory Tracking Control of Robot-Assisted Surgery", International Journal of Hybrid Information Technology, vol. 8, no. 5, **(2015)**, pp. 73-90, http://dx.doi.org/10.14257/ijhit.2015.8.5.08.

Authors

Farzin Piltan, is an outstanding scientist in the field of Electronics and Control engineering with expertise in the areas of nonlinear systems, robotics, and microelectronic control. Mr. Piltan is an advanced degree holder in his field. Currently, Mr. Piltan is the Head of Mechatronics, Intelligent System, and Robotics Laboratory at the Iranian Institute of Advanced Science and Technology (IRAN SSP). Mr. Piltan led several high impact projects involving more than 150 researchers from countries around the world including Iran, Finland, Italy, Germany, South Korea, Australia, and the United States. Mr. Piltan has authored or co-authored more than 140 papers in academic journals, conference papers and book chapters. His papers have been cited at least 3900 times by independent and dependent researchers **from around the world including Iran, Algeria, Pakistan, India, China, Malaysia, Egypt, Columbia, Canada, United Kingdom, Turkey, Taiwan, Japan, South Korea, Italy, France, Thailand, Brazil and more**. Moreover, **Mr. Piltan has peer-reviewed at least 23 manuscripts for respected international journals in his field**. Mr. Piltan will also serve as a technical committee member of the upcoming EECSI 2015 Conference in Indonesia. Mr. Piltan has served as an editorial board member or journal reviewer of several international journals in his field as follows: International Journal Of Control And Automation (IJCA), Australia, ISSN: 2005-4297, International Journal of Intelligent System and Applications (IJISA), Hong Kong, ISSN:2074-9058, IAES International Journal Of Robotics And Automation, Malaysia, ISSN:2089-4856, International Journal of Reconfigurable and Embedded Systems, Malaysia, ISSN:2089-4864.

Mr. Piltan has acquired a formidable repertoire of knowledge and skills and established himself as one of the leading young scientists in his field. Specifically, he has accrued expertise in the design and implementation of intelligent controls in nonlinear systems. Mr. Piltan has employed his remarkable expertise in these areas to make outstanding contributions as detailed follows:Nonlinear control for

industrial robot manipulator (2010-IRAN SSP), Intelligent Tuning The Rate Of Fuel Ratio In Internal Combustion Engine (2011-IRANSSP), Design High Precision and Fast Dynamic Controller For Multi-Degrees Of Freedom Actuator (2013-IRANSSP), Research on Full Digital Control for Nonlinear Systems (2011-IRANSSP), Micro-Electronic Based Intelligent Nonlinear Controller (2015-IRANSSP), Active Robot Controller for Dental Automation (2015-IRANSSP), Design a Micro-Electronic Based Nonlinear Controller for First Order Delay System (2015-IRANSSP).

The above original accomplishments clearly demonstrate that Mr. Piltan has performed original research and that he has gained a distinguished reputation as an outstanding scientist in the field of electronics and control engineering. Mr. Piltan has a tremendous and unique set of skills, knowledge and background for his current and future work. He possesses a rare combination of academic knowledge and practical skills that are highly valuable for his work. **In 2011, he published 28 first author papers, which constitute about 30% of papers published by the Department of Electrical and Electronic Engineering at University Putra Malaysia.** Additionally, **his 28 papers represent about 6.25% and 4.13% of all control and system papers published in Malaysia and Iran, respectively, in 2011**.

Ali Taghizadegan, is currently studying as a student in the **second grade** of Shahid dastgheib's 1 high school and **Research Student** at Iranian Institute of Advanced Science and Technology, Research and Training Center, IRAN SSP. He is research student of team (6 researchers) to design Micro-electronic Based nonlinear controller for Four Degrees of Freedom Surgical Robot Manipulator since August 2015. His current research interests are nonlinear control, artificial control system, Microelectronic Device, and HDL design.

Nasri Sulaiman, is a Senior Lecturer in the Department Electrical and Electronic Engineering at the Universiti Purta Malaysia (UPM), which is one of the leading research universities in Malaysia. He is a supervisor and senior researcher at research and training center called, Iranian Institute of Advanced Science and technology (Iranssp) since 2012. He obtained his M.Sc. from the University of Southampton (UK), and Ph.D. in Microelectronics from the University of Edinburgh (UK). He has published more than 80 technical papers related to control and system engineering, including several co-authored papers with Mr. Piltan. He has been invited to present his research at numerous national and international conferences. He has supervised many graduate students at doctoral and masters level. He is an outstanding scientist in the field of Micro-Electronics.

Dr. Nasri Sulaiman advisor and supervisor of several high impact projects involving more than 150 researchers from countries around the world including Iran, Malaysia, Finland, Italy, Germany, South Korea, Australia, and the United States. Dr. Nasri Sulaiman has authored or co-authored more than 80 papers in academic journals,

conference papers and book chapters. His papers have been cited at least 3000 times by independent and dependent researchers **from around the world including Iran, Algeria, Pakistan, India, China, Malaysia, Egypt, Columbia, Canada, United Kingdom, Turkey, Taiwan, Japan, South Korea, Italy, France, Thailand, Brazil and more**.

Dr. Nasri Sulaiman has employed his remarkable expertise in these areas to make outstanding contributions as detailed below:

- Design of a reconfigurable Fast Fourier Transform (FFT) Processor using multi-objective Genetic Algorithms (2008-UPM)
- Power consumption investigation in reconfigurable Fast Fourier Transform (FFT) processor (2010-UPM)
- Crest factor reduction And digital predistortion Implementation in Orthogonal frequency Division multiplexing (ofdm) systems (2011-UPM)
- High Performance Hardware Implementation of a Multi-Objective Genetic Algorithm, (RUGS), Grant amount RM42,000.00, September (2012-UPM)
- Nonlinear control for industrial robot manipulator (2010-IRAN SSP)
- Intelligent Tuning The Rate Of Fuel Ratio In Internal Combustion Engine (2011-IRANSSP)
- Design High Precision and Fast Dynamic Controller For Multi-Degrees Of Freedom Actuator (2013-IRANSSP)
- Research on Full Digital Control for Nonlinear Systems (2011-IRANSSP)
- Micro-Electronic Based Intelligent Nonlinear Controller (2015-IRANSSP)
- Active Robot Controller for Dental Automation (2015-IRANSSP)
- Design a Micro-Electronic Based Nonlinear Controller for First Order Delay System (2015-IRANSSP)