International Journal of Hybrid Information Technology
Vol. 9, No. 5 (2016), pp. 431-474
http://dx.doi.org/10.14257/ijhit.2016.9.5.37

Design High Frequency Surgical Robot Controller: Design FPGA-Based Controller for Surgical Robot Manipulator Simscape Modeling

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

[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
Email: piltan_f@iranssp.org, www.iranssp.org/english

Abstract

Recent developments of robotics allocated many of industrial and medical activities. So that most of industries turned to use surgical robots in their production line or in their surgery. Being precise, spent less time-consuming, present uniform quality with less cost and reducing waste and energy are *some advantages of using robots in industry.*

This paper has two important objectives: a) study on modeling and controlling of 4 degrees of freedom (DOF) based on Simscape software and b) design FPGA-based controller for this type of surgical robot manipulator. Simscape provides an environment for modeling and simulating physical systems. Simscape modeling can be designed to control and test system-level performance. Conventional PID controller is a stable linear type model-free controller that reduces the delay time in highly nonlinear system. In this research, linear controller need real time mobility operation, and one of the most important devices which can be used to solve this challenge is Field Programmable Gate Array (FPGA). FPGA can be used to design a controller in a single chip Integrated Circuit (IC). To design PID type FPGA-based controller two types algorithm are needed: derivative algorithm and integral algorithm. In HDL based derivative algorithm the minimum input arrival time before clock is 16.466 ns and the maximum frequency is 60.73 MHz, but in the best design action, the maximum frequency to design this single chip algorithm should be 63.629 MHz. In HDL integral algorithm the minimum input arrival time before clock is 15.599 ns and the maximum frequency is 64.1 MHz, but in the best design action, the maximum frequency to design this single chip algorithm should be 178.190 MHz.

Keywords*: surgical robots, four degrees of freedom, PID Controller, FPGA, Digital control, single chip digital controller*

1. Introduction

Simscape is one of the parts in MATLAB/SIMULINK programs, which is used for modeling and implementation systems in physical. Using Simulink can analyze the behavior of a system without having to build it. Although it is one of one of the Simulink's toolbox, it works in different way from it. Simscape can be used for model and control of systems and test system-level performance .we can make model of physical ingredients such as electronic motors, hydraulic valves .This models are exactly the same as physical models [1-8].

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

ISSN: 1738-9968 IJHIT

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

A Field Programmable Gate Array (FPGAs) is a small Field Programmable Device (FPD) that supports thousands of gates. FPGAs are divided in two categories: SRAM-based FPGA, and Antifuse-based FPGA which SRAM-based FPGA used many semiconductor and consists of an array of logic element, routing paths, FPGA I/O pins, on

chip memory, and the other resource and Antifuse-Based FPGAs used two metal layers to sandwich the layere of non-conductive silicon. A typical SRAM- Based FPGAs and Antifuse-Based FPGAs have shown in Figures 3 and 4 [9-15].

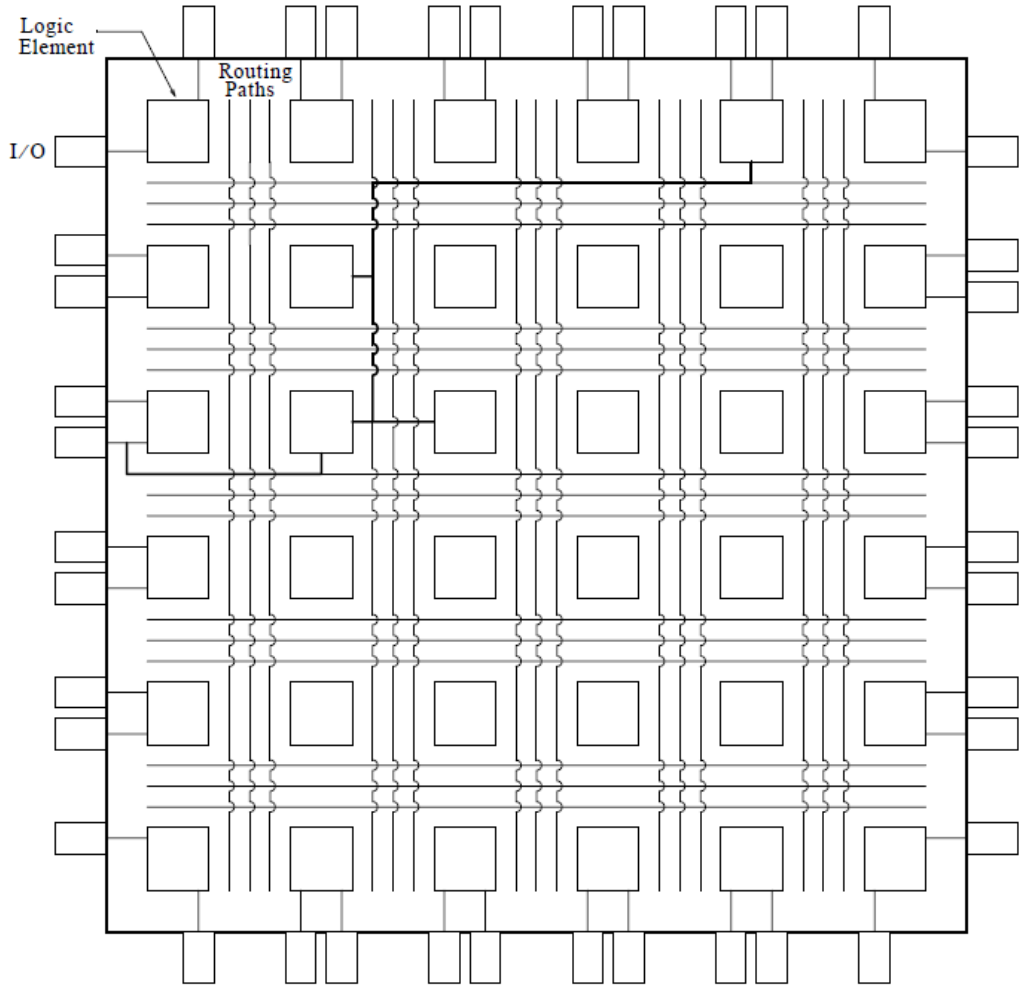

Figure 2. SRAM-based FPGA

Several semiconductor vendors provides a wide range of FPGAs such as Xilinx, Altera, Atmel and Lattice that each one has own unique architecture. In most of FPGAs, logic elements consist of one or more RAM-based n input lookup tables, and one or more Flip-Flops. FPGAs can be used in wide range area such as, Fast Fourier Transforms (FFT), Discrete Cosine Transforms (DCT), Convolution, and Finite Impulse Response (FIR) filters. A FPGA chip can be programmed by Hardware Description Language (HDL) and HDL contains two type of language, Very High Description Language (VHDL) and Verilog which VHDL is one of the powerful programming language that can be used to describe the hardware design. VHDL was developed by the Institute of Electrical and Electronics Engineers (IEEE) in 1987 and Verilog was developed by Gateway Design Automation in 1984. This language became an IEEE standard in 1995 and was updated in 2001 [10-15].

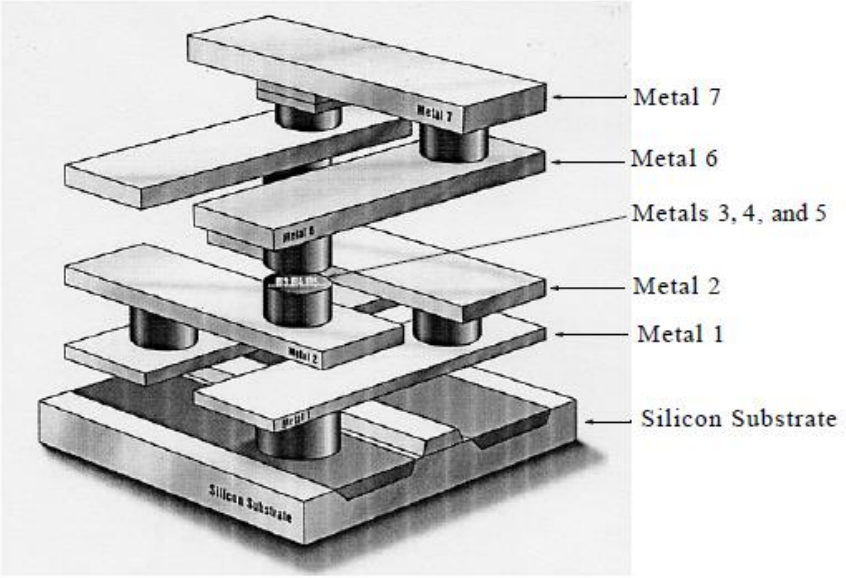

Figure 4. Antifuse-Based FPGAs

In order to provide information about implementing linear controller using HDL on Xilinx FPGAs, this part present introducing of the Xilinx architecture. The Xilinx FPGAs has 6 major blocks namely; Configuration Logic Blocks (CLB'S), Block RAM's(B RAMS); multipliers; Digital Clock Managers (DCM'S); standard, and high speed I/O (IOB's), Figure 5, that can be connected to each other by fully buffered SRAM programmable switching matrix. The switching matrix is programmed and controlled by data of the configuration on loaded in to SRAM. The CLB take up over 75% of area resourse, so all of the other blocks related to the CLB array size. The BRAM's and multipliers are in a narrow space between the CLB's. The DCM's are blocks at the bottom and the top of each part of BRAM's and multipliers. The IOB's are parallel to serial embedded transceiver that can be used for high speed interfaces between multiple FPGA's

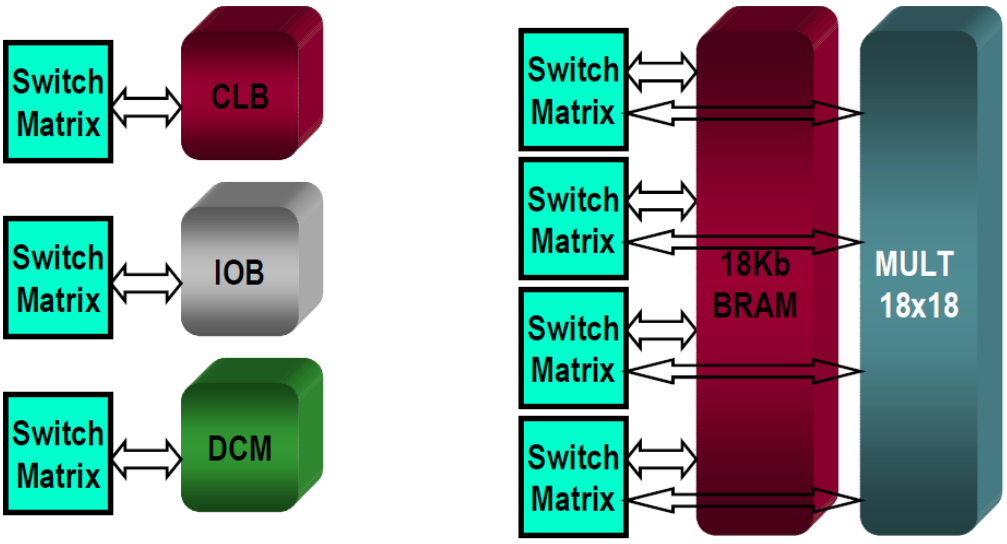

Figure 5. Block Diagram of Xilinx FPGA Architecture

A controller (control system) is a device, which cans sense information from system to improvement the dynamic behavior of first order delay system based on actuation and

computation. From scientific perspectives, control theory is divided into two parts; linear control theory and nonlinear control theory [11-14]. Linear control theory is divided into following groups:

- Proportional-Derivative (PD) control algorithm
- Proportional-Integral (PI) control algorithm
- Proportional-Integral-Derivative (PID) control algorithm

Nonlinear control theory is also has two main divisions;

- Conventional control theory
- Soft computing control theory

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. Proportional algorithm 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 6 shows the block diagram of proportional controller [11-15].

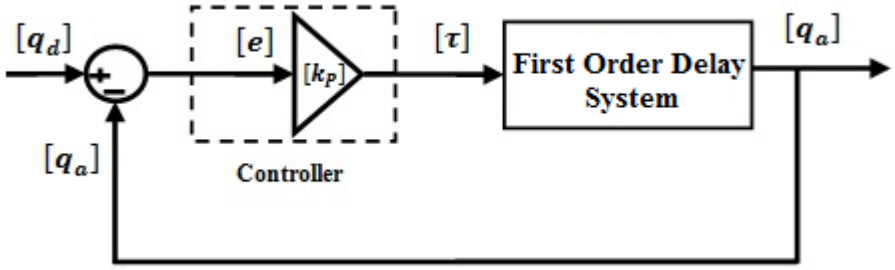

Figure 6. Block Diagram of Proportional Algorithm

Derivative category: derivatives the input signal deviation over a period of time. This part of controller is used to system speed (rate of input signal) in a short time. Figure 7 shows the block diagram of derivative (D) controller with application to FOD system. In mathematically, the formulation of derivative part calculated as follows;

$$\boldsymbol{D = \frac{d}{dt}(e) = \dot{e}} \tag{1}$$

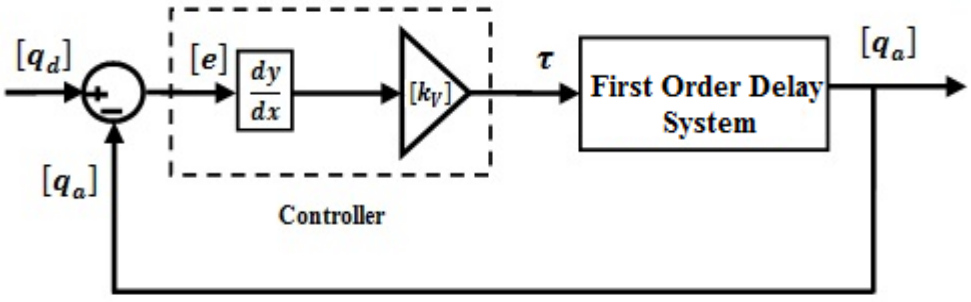

Figure 7. Block Diagram of Derivative Algorithm

Integral Algorithm: 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 8 shows the block diagram of Integral (I) controller with application to first order delay system. In contrast of Proportional type of controller, this type of controller used to

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

$$\boldsymbol{I} = \frac{\mathbf{1}}{\boldsymbol{T}} \int \boldsymbol{e}.\boldsymbol{dt} = \sum \boldsymbol{e} \quad (2)$$

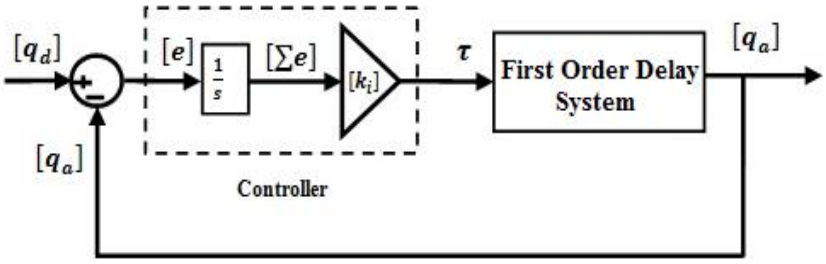

Figure 8. Block Diagram of Integral Control of FOD system

Figure 9 shows the step response of integral controller.

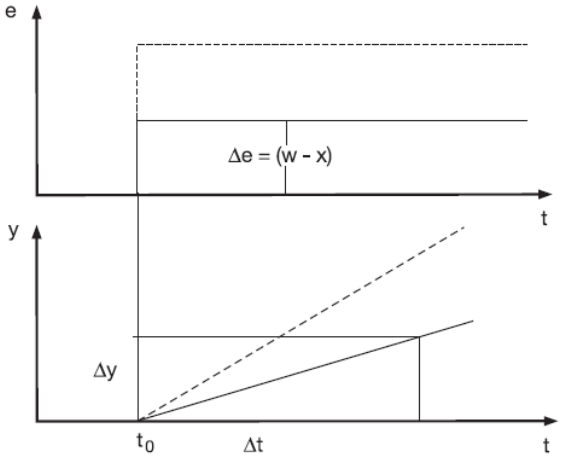

Figure 9. Step Response of an Integral (I) Controller

In this paper, FPGA based PID control algorithm is investigated. To design FPGA-based PID controller, SPARTAN 3E-XA3S1600E is used. The information of this device is introduced as the following Table (Table 1).

Table 1. Summary of XA Spartan-3E FPGA Attributes

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits(1)	Block RAM bits(1)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XA3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XA3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XA3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	190	77
XA3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XA3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

Notes:

1. By convention, one Kb is equivalent to 1,024 bits.

This paper is organized as follows; second part focuses on the system modeling dynamic formulation and Design PD, PI controller. Third part is focused on the

methodology. Simulation result and discussion is illustrated in forth part. The last part focuses on the conclusion and compare between this method and the other ones.

2. Theory

Physical Modeling of Surgical Robot Manipulator Using Simscape:

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. Simscape's library has different parts. Our model is mechanical model so we choose our block diagrams from Sims Mechanics library. It includes different libraries.
Some blocks diagrams that we need for implementation:

- **Body**

Body includes our links in system. The Body block represents a rigid body with properties you customize. It includes:

1. The Body's Mass and Moment Of Inertia Tensor
2. The Coordinates for the Body's Center Of Gravity (CG)
3. One or more Body coordinate systems (CSs)

Figure 10 shows the body block diagram.

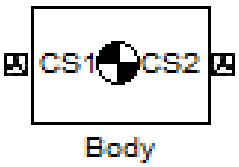

Figure 10. Body Block Diagram

- **Ground**

A Ground block represents an immobile ground point at rest in the absolute inertial World reference frame.by connecting a joint to it, it make it moveless. It can also connect to the Machine environment. It cannot be connected to sensor or actuator because the ground point cannot be moved. Figure 11 shows the ground block diagram.

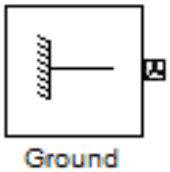

Figure 11. Ground Block Diagram

- **Machine environment**

Machine environment provides conditions to work in different mechanical environments. In fact, it's kind of converter between simscape and Simulink. One of the important part in machine environment is gravity vector, Figure 12 shows the machine environment block diagram.

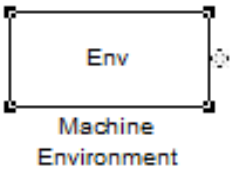

Figure 12. Machine Environment Block Diagram

- **Revolute**

In joints library, the revolute block is sing of rotational degrees of freedom (DOF) about a defined axis between the two bodies. Figure 13 shows the revolute joint block diagram.

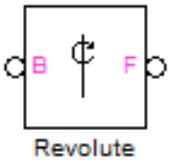

Figure 13. Revolute Joint Block Diagram

- **Joint sensor**

This block measures the acceleration, velocity, reaction force, torque and position of a joint. Figure 14 shows the joint sensor block diagram.

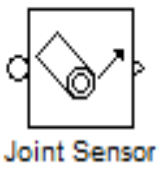

Figure 14. Joint Sensor Block Diagram

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 15 shows the application of forward and inverse kinematics.

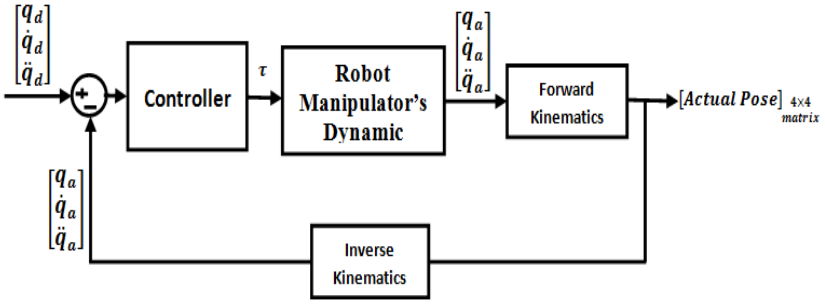

Figure 15. The Application of Forward and Inverse Kinematics

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

$$\boldsymbol{\Psi(X, q) = 0} \tag{3}$$

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 2 shows the standard D-H parameters for N-DOF robot manipulator.

Table 2. 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;

$$\boldsymbol{R_i^{i-1} = U_{i(\theta_i)} V_{i(\alpha_i)}} \tag{4}$$

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

$$\boldsymbol{U_{i(\theta_i)} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) & 0 \\ sin(\theta_i) & cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}} \tag{5}$$

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

$$\boldsymbol{V_{i(\theta_i)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha_i) & -sin(\alpha_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) \end{bmatrix}} \tag{6}$$

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

$$\boldsymbol{R_n^0 = (U_1 V_1)(U_2 V_2) \dots\dots\dots (U_n V_n)} \tag{7}$$

$$\boldsymbol{{}^{n-1}_{n}T = \begin{bmatrix} R_n^{n-1} & d_n^{n-1} \\ 0 & 1 \end{bmatrix}} \tag{8}$$

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

$${}^{i-1}_{1}T = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i\, C\alpha_{i-1} & C\theta_i\, C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i\, S\alpha_{i-1} & C\theta_i\, S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

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

$$\boldsymbol{FK = {}^0_nT = {}^0_1T.{}^1_2T.{}^2_3T \dots\dots {}^{n-1}_{n}T = \begin{bmatrix} R_n^0 & d_n^0 \\ 0 & 1 \end{bmatrix}} \tag{10}$$

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

$$\boldsymbol{{}^0_4T = \begin{bmatrix} N_x & B_x & T_x & P_x \\ N_y & B_y & T_y & P_y \\ N_z & B_z & T_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}} \tag{11}$$

Table 3 shows the 4-DOF surgical robot manipulator Denavit-Hartenberg notations.

Table 3. 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;

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

$$Ny = (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) + sin(\theta_2 + \theta_3) \times sin(\theta_1)) + (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + cos(\theta_4) \times cos(\theta_1)) \quad (13)$$

$$Nz = cos(\theta_4) \times sin(\theta_2 + \theta_3) - cos(\theta_2 + \theta_3)) + sin(\theta_4) \times sin(\theta_2 + \theta_3) \quad (14)$$

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

$$By = -(cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) + sin(\theta_2 + \theta_3) \times sin(\theta_1)) + (sin(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + cos(\theta_4) \times cos(\theta_1)) \quad (16)$$

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

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

$$Ty = (cos(\theta_4) \times cos(\theta_2 + \theta_3) \times sin(\theta_1) - sin(\theta_4) \times cos(\theta_1)) - sin(\theta_2 + \theta_3) \times sin(\theta_1) \quad (19)$$

$$Tz = cos(\theta_4) \times sin(\theta_2 + \theta_3) + cos(\theta_5) \times cos(\theta_2 + \theta_3) \quad (20)$$

$$\boldsymbol{Px = 0.4331 \times sin(\theta_2 + \theta_3) \times cos(\theta_1) + 0.0203 \times cos(\theta_2 + \theta_3) \times cos(\theta_1) - 0.1491 \times sin(\theta_1) + 0.4318 \times cos(\theta_2 cos(\theta_1)} \quad (21)$$

$$\boldsymbol{Py = 0.4331 \times sin(\theta_2 + \theta_3) \times sin(\theta_1) + 0.0203 \times cos(\theta_2 + \theta_3) \times sin(\theta_1) + 0.1491 \times cos(\theta_1) + 0.4312 \times cos(\theta_2) \times sin(\theta_1)} \quad (22)$$

$$\boldsymbol{Pz = -0.4331 \times cos(\theta_2 + \theta_3) + 0.0203 \times sin(\theta_2 + \theta_3) + 0.4318 \times sin(\theta_2)} \quad (23)$$

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]:

$$[\boldsymbol{A(q)}]\ddot{\boldsymbol{q}} + [\boldsymbol{N(q, \dot{q})}] = [\boldsymbol{\tau}] \quad (24)$$

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;

$$\boldsymbol{A(q)} = \begin{bmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & \dots & \dots. & \dots.. & \boldsymbol{A_{1n}} \\ \boldsymbol{A_{21}} & \dots & \dots & \dots. & \dots.. & \boldsymbol{A_{2n}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \boldsymbol{A_{n.1}} & \dots & \dots & \dots. & \dots & \boldsymbol{A_{n.n}} \end{bmatrix} \quad (25)$$

$N(q, \dot{q})$ 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 $\boldsymbol{b(q)}$, Centrifugal $\boldsymbol{C(q)}$, and Gravity $\boldsymbol{G(q)}$. Consequently the robot manipulator dynamic equation can also be written as [8]:

$$[\boldsymbol{N(q, \dot{q})}] = [\boldsymbol{V(q, \dot{q})}] + [\boldsymbol{G(q)}] \quad (26)$$

$$[\boldsymbol{V(q, \dot{q})}] = [\boldsymbol{b(q)}][\dot{\boldsymbol{q}}\ \dot{\boldsymbol{q}}] + [\boldsymbol{C(q)}][\dot{\boldsymbol{q}}]^{\boldsymbol{2}} \quad (27)$$

$$\boldsymbol{\tau = A(q)\ddot{q} + b(q)[\dot{q}\ \dot{q}] + C(q)[\dot{q}]^2 + G(q)} \quad (28)$$

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: $[\dot{q}_1.\dot{q}_2, \dot{q}_1.\dot{q}_3, \dots., \dot{q}_1.\dot{q}_n, \dot{q}_2.\dot{q}_3, \dots..]^T$, 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;

$$\boldsymbol{Outputs} = \boldsymbol{function}\ (\boldsymbol{inputs}) \tag{29}$$

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

$$\boldsymbol{q} = \boldsymbol{function}\ (\boldsymbol{\tau}) \tag{30}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{A^{-1}(q)}.\{\boldsymbol{\tau} - \boldsymbol{N(q,\dot{q})}\} \tag{31}$$

$$\boldsymbol{q} = \iint \boldsymbol{A^{-1}(q)}.\{\boldsymbol{\tau} - \boldsymbol{N(q,\dot{q})}\} \tag{32}$$

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

$$\boldsymbol{b(q)} = \begin{bmatrix} \boldsymbol{b_{112}} & \boldsymbol{b_{113}} & \ldots & \boldsymbol{b_{11n}} & \boldsymbol{b_{123}} & \ldots & \boldsymbol{b_{12n}} & \ldots & \ldots & \boldsymbol{b_{1.n-1.n}} \\ \boldsymbol{b_{212}} & \ldots & \ldots & \boldsymbol{b_{21n}} & \boldsymbol{b_{223}} & \ldots & \ldots & \ldots & \ldots & \boldsymbol{b_{2.n-1.n}} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \boldsymbol{b_{n.1.2}} & \ldots & \ldots & \boldsymbol{b_{n.1.n}} & \ldots & \ldots & \ldots & \ldots & \ldots & \boldsymbol{b_{n.n-1.n}} \end{bmatrix} \tag{33}$$

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

$$\boldsymbol{C(q)} = \begin{bmatrix} \boldsymbol{C_{11}} & \cdots & \boldsymbol{C_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{C_{n1}} & \cdots & \boldsymbol{C_{nn}} \end{bmatrix} \tag{34}$$

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

$$\boldsymbol{G(q)} = \begin{bmatrix} \boldsymbol{g_1} \\ \boldsymbol{g_2} \\ \vdots \\ \boldsymbol{g_n} \end{bmatrix} \tag{35}$$

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

$$\boldsymbol{A(\ddot{\theta})}\begin{bmatrix} \ddot{\boldsymbol{\theta}}_1 \\ \ddot{\boldsymbol{\theta}}_2 \\ \ddot{\boldsymbol{\theta}}_3 \\ \ddot{\boldsymbol{\theta}}_4 \end{bmatrix} + \mathbf{B}(\boldsymbol{\theta})\begin{bmatrix} \dot{\boldsymbol{\theta}}_1\dot{\boldsymbol{\theta}}_2 \\ \dot{\boldsymbol{\theta}}_1\dot{\boldsymbol{\theta}}_3 \\ \dot{\boldsymbol{\theta}}_1\dot{\boldsymbol{\theta}}_4 \\ \dot{\boldsymbol{\theta}}_2\dot{\boldsymbol{\theta}}_3 \\ \dot{\boldsymbol{\theta}}_2\dot{\boldsymbol{\theta}}_4 \\ \dot{\boldsymbol{\theta}}_3\dot{\boldsymbol{\theta}}_4 \end{bmatrix} + \boldsymbol{C(\theta)}\begin{bmatrix} \dot{\boldsymbol{\theta}}_1^2 \\ \dot{\boldsymbol{\theta}}_2^2 \\ \dot{\boldsymbol{\theta}}_3^2 \\ \dot{\boldsymbol{\theta}}_4^2 \end{bmatrix} + \boldsymbol{G(\theta)} = \begin{bmatrix} \boldsymbol{\tau_1} \\ \boldsymbol{\tau_2} \\ \boldsymbol{\tau_3} \\ \boldsymbol{\tau_4} \end{bmatrix} \tag{36}$$

Where

$$\boldsymbol{A(q)} = \begin{bmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & \boldsymbol{A_{13}} & \mathbf{0} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} & \mathbf{0} \\ \boldsymbol{A_{31}} & \boldsymbol{A_{32}} & \boldsymbol{A_{33}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{A_{44}} \end{bmatrix} \tag{37}$$

According to **[8]** the inertial matrix elements ($\boldsymbol{A}$) are

$$\boldsymbol{A_{11} = I_{m1} + I_1 + I_3 \times \cos(\theta_2)\cos(\theta_2) + I_7\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + I_{10}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{11}\sin(\theta_2)\cos(\theta_2) + I_{21}\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + 2 + [I_5\cos(\theta_2)\sin(\theta_2+\theta_3) + I_{12}\cos(\theta_2)\cos(\theta_2+\theta_3) + I_{15}\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + I_{16}\cos(\theta_2)\sin(\theta_2+\theta_3) + I_{22}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)}$$ (38)

$$\boldsymbol{A_{12} = I_4\sin(\theta_2) + I_8\cos(\theta_2+\theta_3) + I_9 cos(\theta_2) + I_{13}\sin(\theta_2+\theta_3) - I_{18}\cos(\theta_2+\theta_3)}$$ (39)

$$\boldsymbol{A_{13} = I_8\cos(\theta_2+\theta_3) + I_{13}\sin(\theta_2+\theta_3) - I_{18}\cos(\theta_2+\theta_3)}$$ (40)

$$\boldsymbol{A_{22} = I_{m2} + I_2 + I_6 + 2[I_5\sin(\theta_3) + I_{12}\cos(\theta_2) + I_{15} + I_{16}\sin(\theta_3)}$$ (41)

$$\boldsymbol{A_{23} = I_5\sin(\theta_3) + I_6 + I_{12}\cos(\theta_3) + I_{16}\sin(\theta_3) + 2I_{15}}$$ (42)

$$\boldsymbol{A_{33} = I_{m3} + I_6 + 2I_{15}}$$ (43)

$$\boldsymbol{A_{44} = I_{m4} + I_{14}}$$ (44)

$$\boldsymbol{A_{21} = A_{12}\,, A_{31} = A_{13}\ and\ A_{32} = A_{23}}$$ (45)

The Corilios ($\boldsymbol{b}$) matrix elements are;

$$\boldsymbol{b(q)} = \begin{bmatrix} \boldsymbol{b_{112}} & \boldsymbol{b_{113}} & \boldsymbol{0} & \boldsymbol{b_{123}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{b_{214}} & \boldsymbol{b_{223}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{b_{314}} & \boldsymbol{0} \\ \boldsymbol{b_{412}} & \boldsymbol{b_{413}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$ (46)

Where,

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

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

$$\boldsymbol{b_{123} = 2[-I_8\sin(\theta_2+\theta_3) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3)\,]}$$ (49)

$$\boldsymbol{b_{214} = I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) + 2I_{20}\sin(\theta_2+\theta_3)(1-0.5)}$$ (50)

$$\boldsymbol{b_{223} = 2[-I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3)\,]}$$ (51)

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

$$b_{412} = b_{214} = -[I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) + 2I_{20}\sin(\theta_2+\theta_3)(1-0.5)] \quad (53)$$

$$b_{413} = -b_{314} = -2[I_{20}\sin(\theta_2+\theta_3)(1-0.5)] + I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) \quad (54)$$

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

$$[\boldsymbol{b}(\boldsymbol{q}).\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}]_{4\times1} = \begin{bmatrix} b_{112}.\dot{q_1}\dot{q_2} + b_{113}.\dot{q_1}\dot{q_3} + 0 + b_{123}.\dot{q_2}\dot{q_3} \\ 0 + b_{214}.\dot{q_1}\dot{q_4} + b_{223}.\dot{q_2}\dot{q_3} \\ b_{314}.\dot{q_1}\dot{q_4} \\ b_{412}.\dot{q_1}\dot{q_2} + b_{413}.\dot{q_1}\dot{q_3} \end{bmatrix} \quad (55)$$

According to [8] Centrifugal $(\boldsymbol{C})$ matrix elements are;

$$\boldsymbol{C}(\boldsymbol{q}) = \begin{bmatrix} 0 & C_{12} & C_{13} & 0 \\ C_{21} & C_{22} & C_{23} & 0 \\ C_{31} & C_{32} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (56)$$

Where,

$$c_{12} = I_4\cos(\theta_2) - I_8\sin(\theta_2+\theta_3) - I_9\sin(\theta_2) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3) \quad (57)$$

$$c_{13} = 0.5b_{123} = -I_8\sin(\theta_2+\theta_3) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3) \quad (58)$$

$$c_{21} = -0.5b_{112} = I_3\sin(\theta_2)\cos(\theta_2) - I_5\cos(\theta_2+\theta_2+\theta_3) - I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{12}\sin(\theta_2+\theta_2+\theta_3) + I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2+\theta_2+\theta_3) - I_{21}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{22}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) - 0.5I_{10}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) - 0.5I_{11}\big(1-2\sin(\theta_2)\sin(\theta_2)\big) \quad (59)$$

$$c_{22} = 0.5b_{223} = -I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3) \quad (60)$$

$$c_{23} = -0.5b_{113} = -I_5\cos(\theta_2)\cos(\theta_2+\theta_3) - I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{12}\cos(\theta_2)\sin(\theta_2+\theta_2) - I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2)\cos(\theta_2+\theta_3) - I_{21}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{22}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) - 0.5I_{10}\big(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\big) \quad (61)$$

$$c_{31} = -c_{23} = I_{12}\sin(\theta_3) - I_5\cos(\theta_3) - I_{16}\cos(\theta_3) \quad (62)$$

$$c_{32} = \sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2)\cos(\theta_2+\theta_3) - I_{22}\cos(\theta_2+\theta_3)\cos(\theta_2+\theta_3) \quad (63)$$

$$\left[\boldsymbol{C}(\boldsymbol{q}).\dot{\boldsymbol{q}}^{\boldsymbol{2}}\right]_{\boldsymbol{6\times1}}=\begin{bmatrix}\boldsymbol{c_{12}}.\boldsymbol{q_2^2}+\boldsymbol{c_{13}}.\boldsymbol{q_3^2}\\ \boldsymbol{c_{21}}.\boldsymbol{q_1^2}+\boldsymbol{c_{23}}.\boldsymbol{q_3^2}\\ \boldsymbol{c_{13}}.\boldsymbol{q_1^2}+\boldsymbol{c_{32}}.\boldsymbol{q_2^2}\\ \boldsymbol{0}\end{bmatrix} \tag{64}$$

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

$$[\boldsymbol{G}(\boldsymbol{q})]_{\boldsymbol{4\times1}}=\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{G_2}\\ \boldsymbol{G_3}\\ \boldsymbol{0}\end{bmatrix} \tag{65}$$

Where,

$$\boldsymbol{G_2}=\boldsymbol{g_1}\boldsymbol{\cos}(\boldsymbol{\theta_2})+\boldsymbol{g_2}\,\boldsymbol{\sin}(\boldsymbol{\theta_2}+\boldsymbol{\theta_3})+\boldsymbol{g_3}\boldsymbol{\sin}(\boldsymbol{\theta_2})+\boldsymbol{g_4}\boldsymbol{\cos}(\boldsymbol{\theta_2}+\boldsymbol{\theta_3})+\boldsymbol{g_5}\boldsymbol{\sin}(\boldsymbol{\theta_2}+\boldsymbol{\theta_3}) \tag{66}$$

$$\boldsymbol{G_3}=\boldsymbol{g_2}\,\boldsymbol{\sin}(\boldsymbol{\theta_2}+\boldsymbol{\theta_3})+\boldsymbol{g_4}\boldsymbol{\cos}(\boldsymbol{\theta_2}+\boldsymbol{\theta_3})+\boldsymbol{g_5}\boldsymbol{\sin}(\boldsymbol{\theta_2}+\boldsymbol{\theta_3}) \tag{67}$$

If $[\boldsymbol{I}]_{\boldsymbol{4\times1}}=[\boldsymbol{B}]_{\boldsymbol{4\times1}}+[\boldsymbol{C}]_{\boldsymbol{4\times1}}+[\boldsymbol{G}]_{\boldsymbol{4\times1}}$

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

$$[\ddot{\boldsymbol{q}}]_{\boldsymbol{4\times1}}=\left[\boldsymbol{A^{-1}}(\boldsymbol{q})\right]_{\boldsymbol{4\times4}}\times\{[\boldsymbol{\tau}]_{\boldsymbol{4\times1}}-[\boldsymbol{I}]_{\boldsymbol{4\times1}}\} \tag{68}$$

K is presented as follows;

$$[\boldsymbol{K}]_{\boldsymbol{4\times1}}=\{[\boldsymbol{\tau}]_{\boldsymbol{4\times1}}-[\boldsymbol{I}]_{\boldsymbol{4\times1}}\} \tag{69}$$

$$[\ddot{\boldsymbol{q}}]_{\boldsymbol{4\times1}}=\left[\boldsymbol{A^{-1}}(\boldsymbol{q})\right]_{\boldsymbol{4\times4}}\times[\boldsymbol{K}]_{\boldsymbol{4\times1}} \tag{70}$$

$$[\boldsymbol{q}]_{\boldsymbol{4\times1}}=\iint\left[\boldsymbol{A^{-1}}(\boldsymbol{q})\right]_{\boldsymbol{4\times4}}\times[\boldsymbol{K}]_{\boldsymbol{4\times1}} \tag{71}$$

Basic information about inertial and gravitational constants is show in tables 4 and 5

Table 4. Inertial Constant Reference ($Kg.m^2$)

$\mathrm{I}_1=1.43\pm0.05$	$I_2=1.75\pm0.07$
$I_3=1.38\pm0.05$	$I_4=0.69\pm0.02$
$I_5=0.372\pm0.031$	$\mathrm{I}_6=0.333\pm0.016$
$I_7=0.298\pm0.029$	$\mathrm{I}_8=-0.134\pm0.014$
$I_9=0.0238\pm0.012$	$I_{10}=-0.0213\pm0.0022$
$I_{11}=-0.0142\pm0.0070$	$I_{12}=-0.011\pm0.0011$
$I_{13}=-0.00379\pm0.0009$	$I_{14}=0.00164\pm0.000070$

$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$I_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$I_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.000015$
$I_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

Table 5. Gravitational Constant (*N.m*)

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

3. Steps to System Modeling By Simscape

To model our system, the position and orientation of our base and links are needed.
Ground point position: Figure16 shows the position of ground.

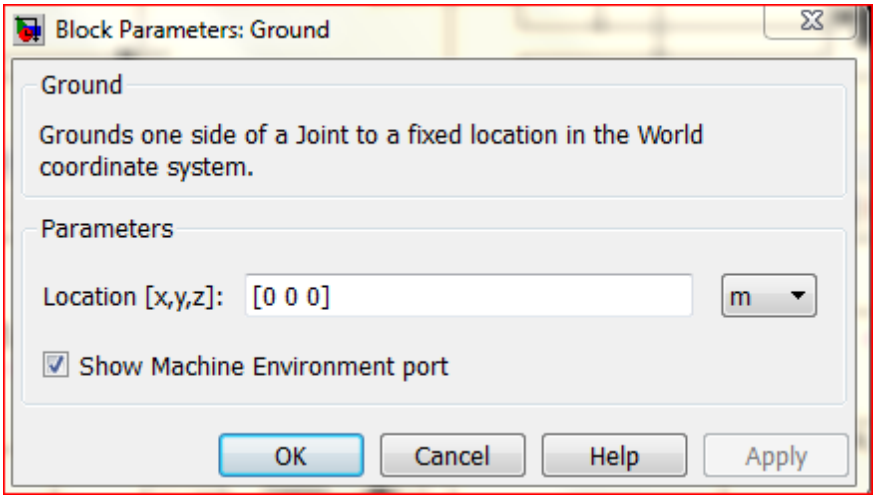

Figure 16. Ground Position

Arm position: Figure 17 shows the position of arm.

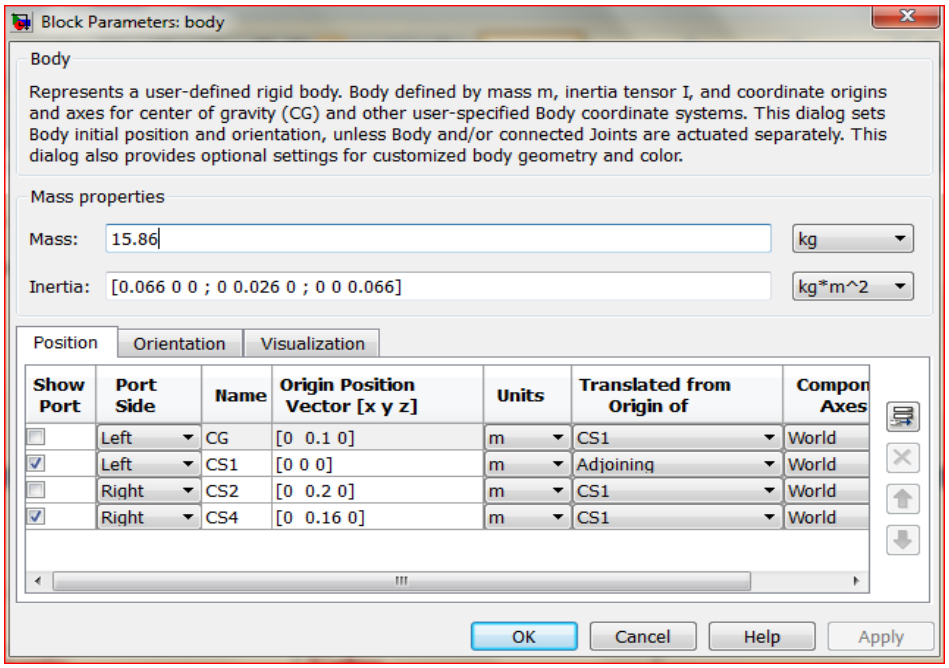

Figure 17. Arm Position

Arm orientation: Figure 18 shows the orientation of arm.

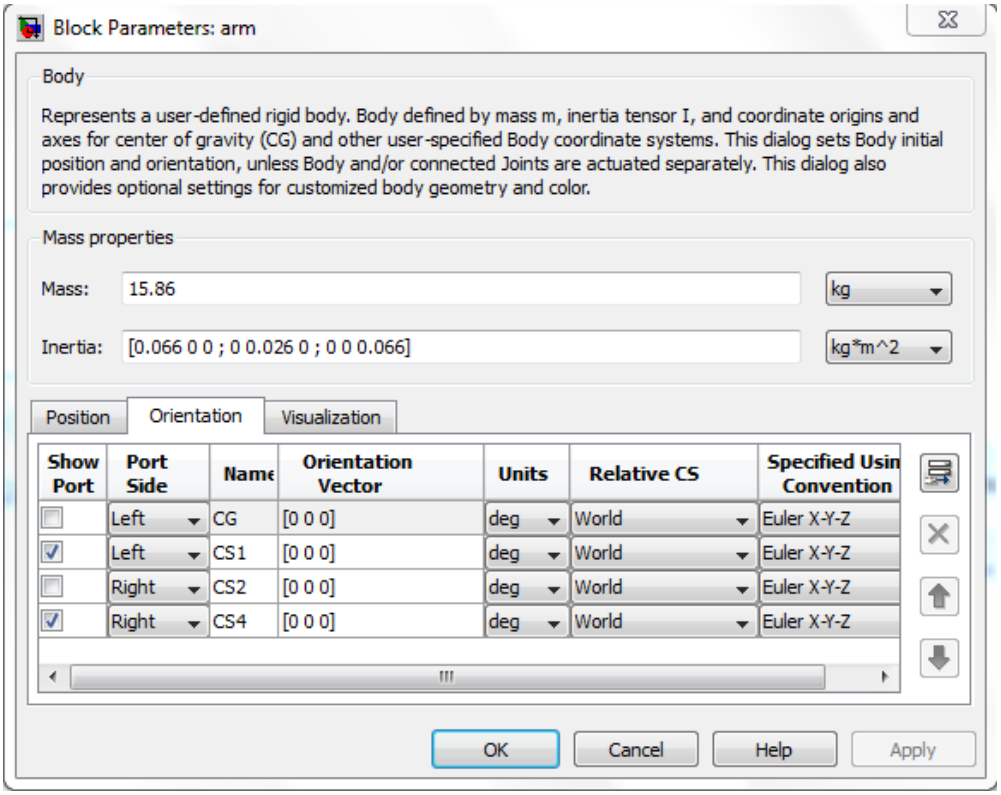

Figure18. Arm Orientation

Forearm position: Figure 19 shows the position of forearm.

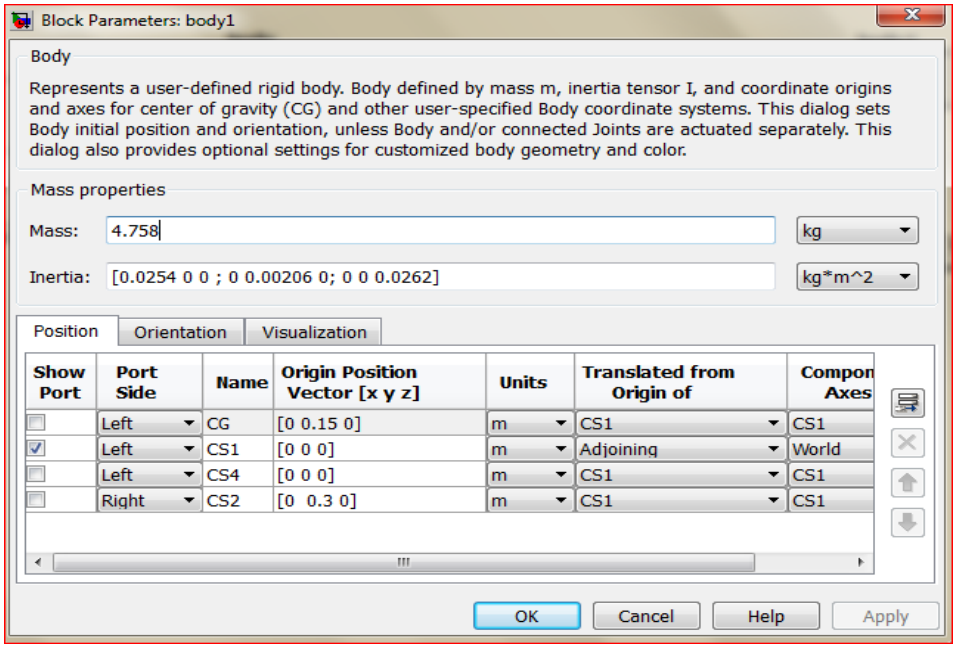

Figure19. Forearm Position

Forearm orientation: Figure 20 shows the orientation of the forearm.

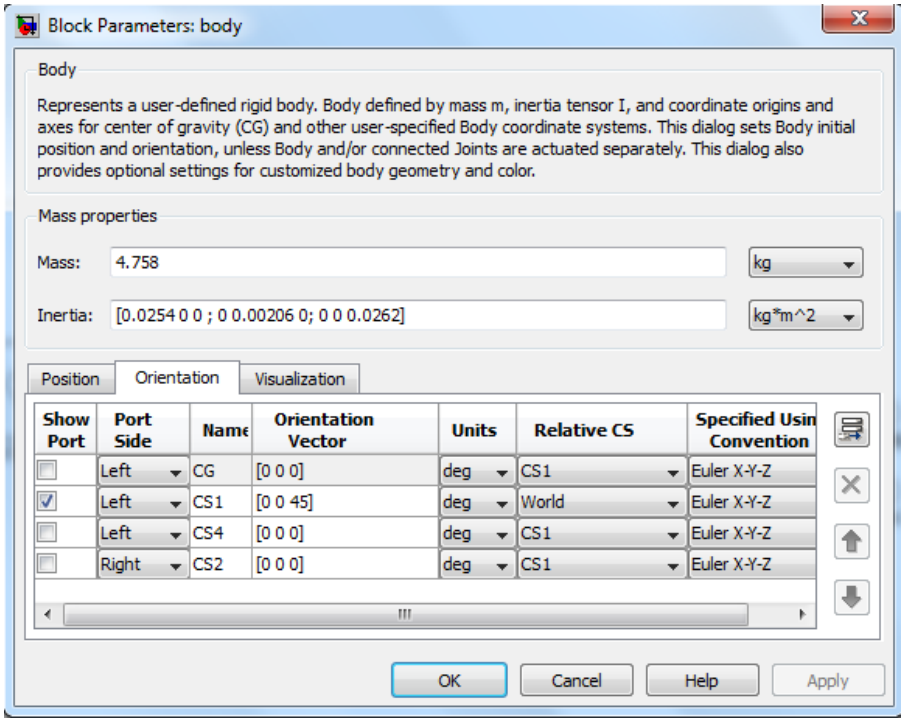

Figure 20. Forearm Orientation

Wrist position: Figure 21 shows the position of wrist**.**

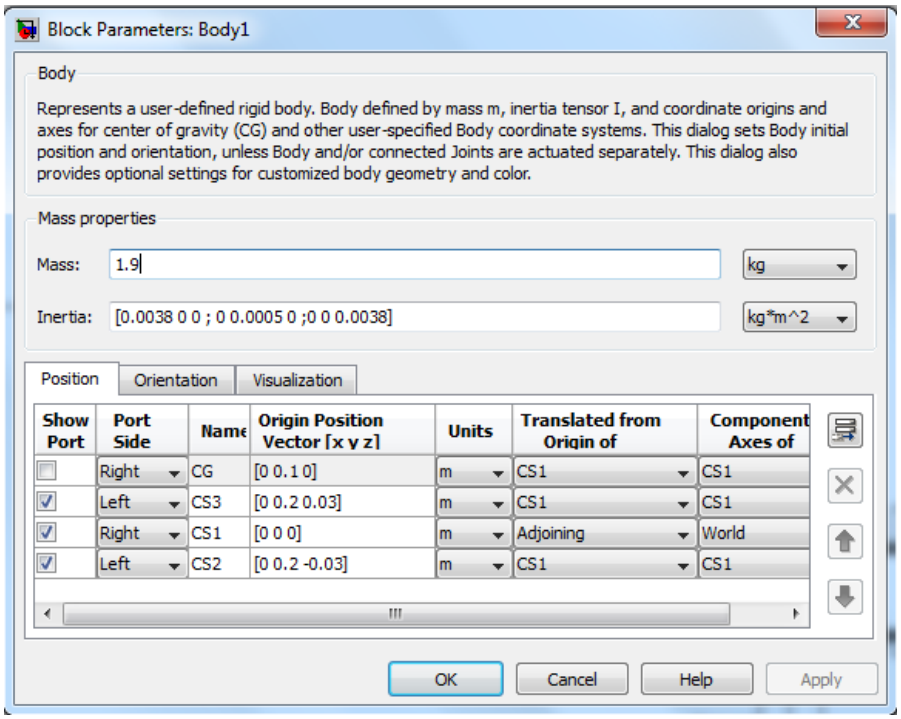

Figure 21. Wrist Position

Wrist orientation: Figure 22 shows the orientation of wrist.

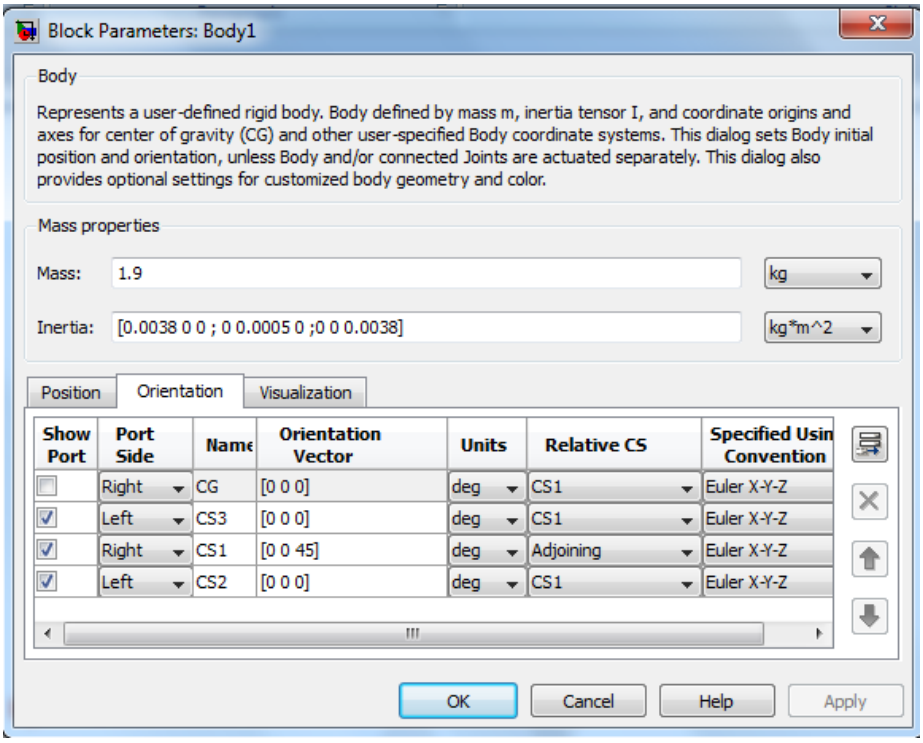

Figure 22. Wrist Orientation

Finger1: finger 23, 24 properties should be as follow:

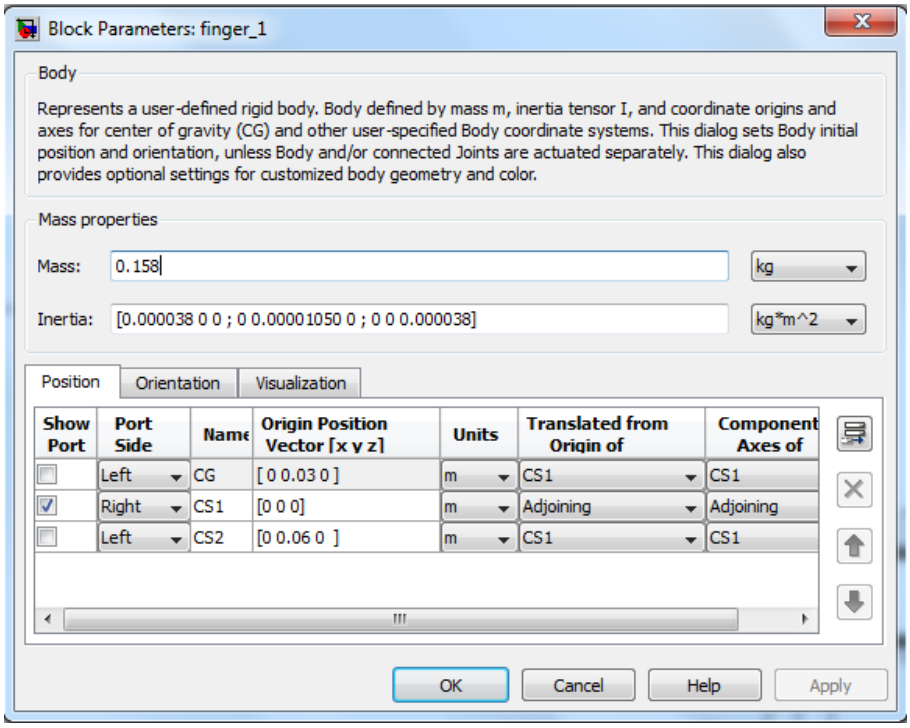

Figure 23. Finger1 Position

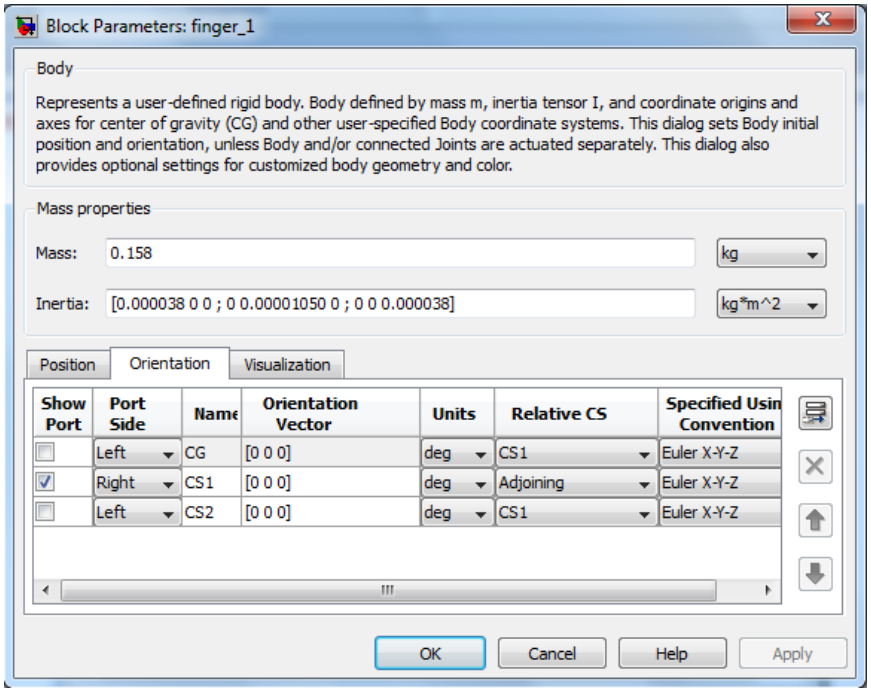

Figure 24. Finger1 Orientation

Finger2: Figure 25 shows the position of finger2.

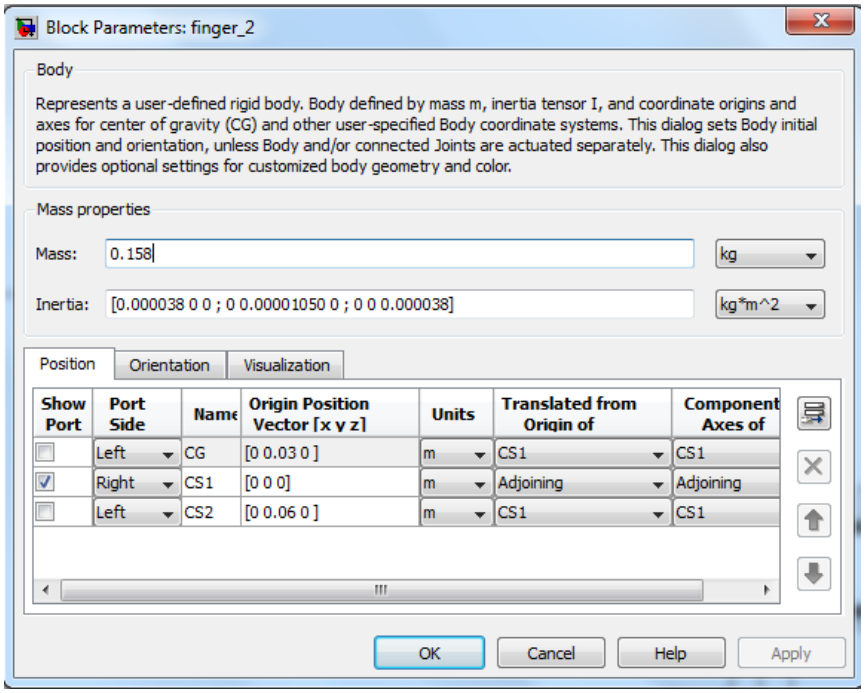

Figure 25. Finger2 Position

Figure 26 shows the orientation of finger2.

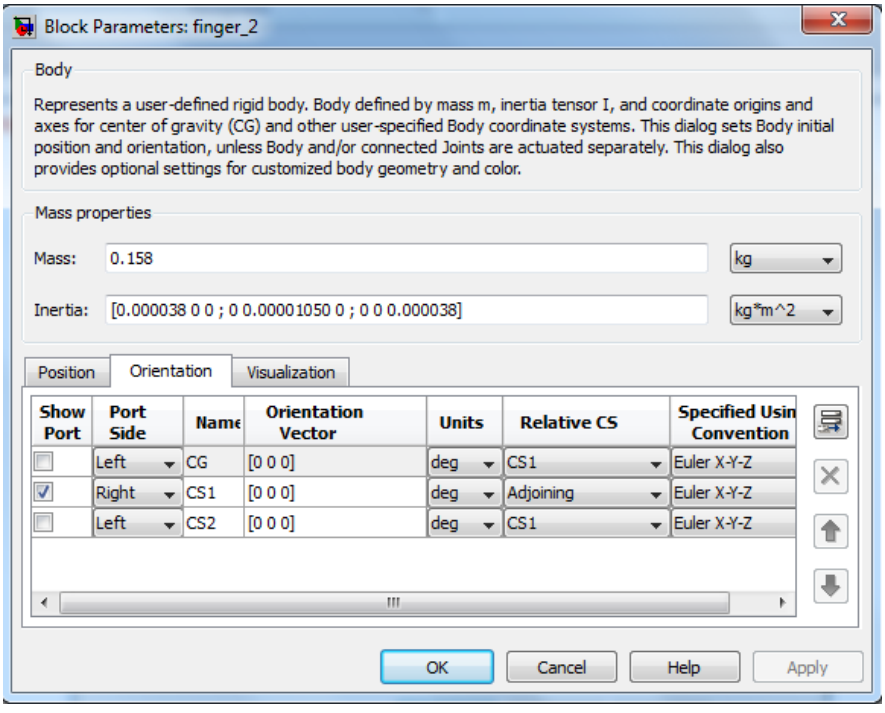

Figure 26. Finger2 Orientation

For each link, one revolute is chosen. So we have 5 joints.
Joint 1(revolute1): Figure 27 shows the information of Joint 1(revolute1).

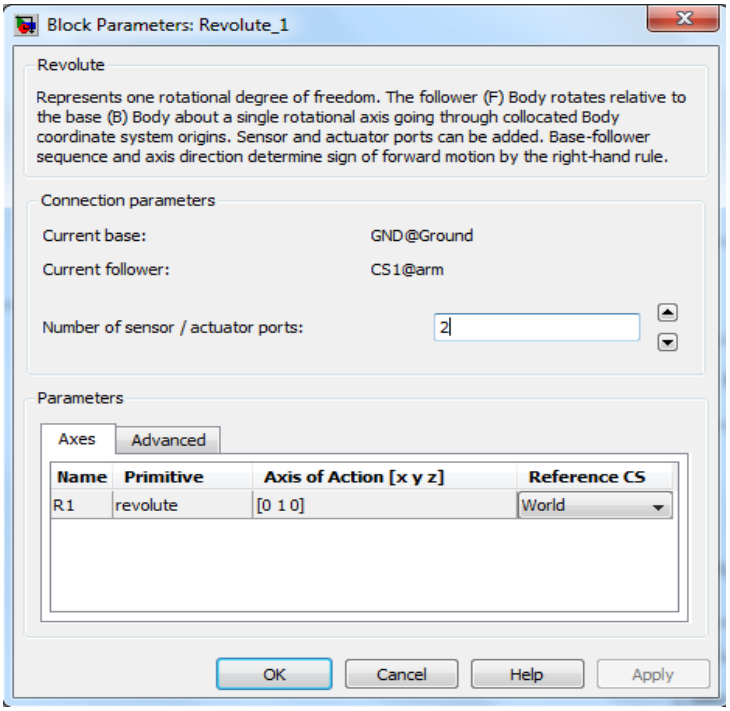

Figure 27. Joint 1(Revolute1)

Joint 2(revolute2): Figure 28 shows the information of Joint 2(revolute2).

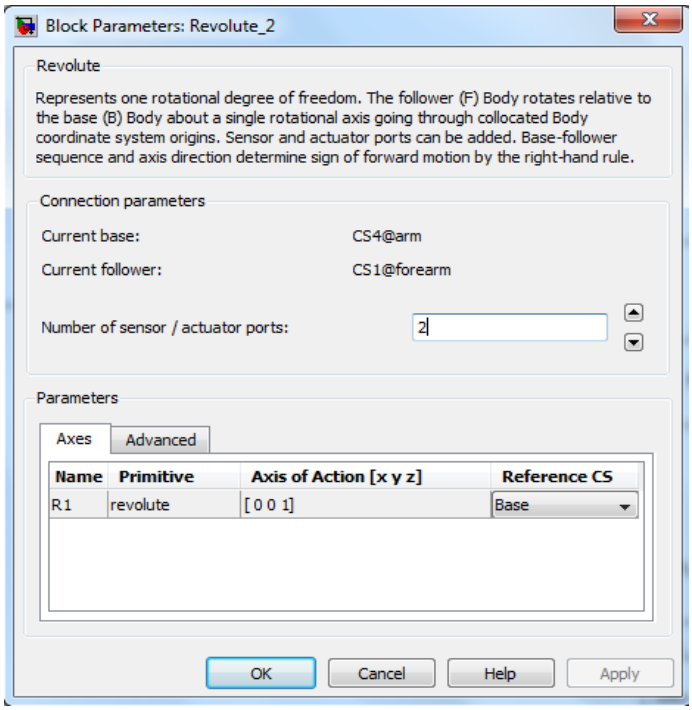

Figure 28. Joint 2(Revolute2)

Joint 3(revolute3): Figure 29 shows the information of Joint3 (revolute3).

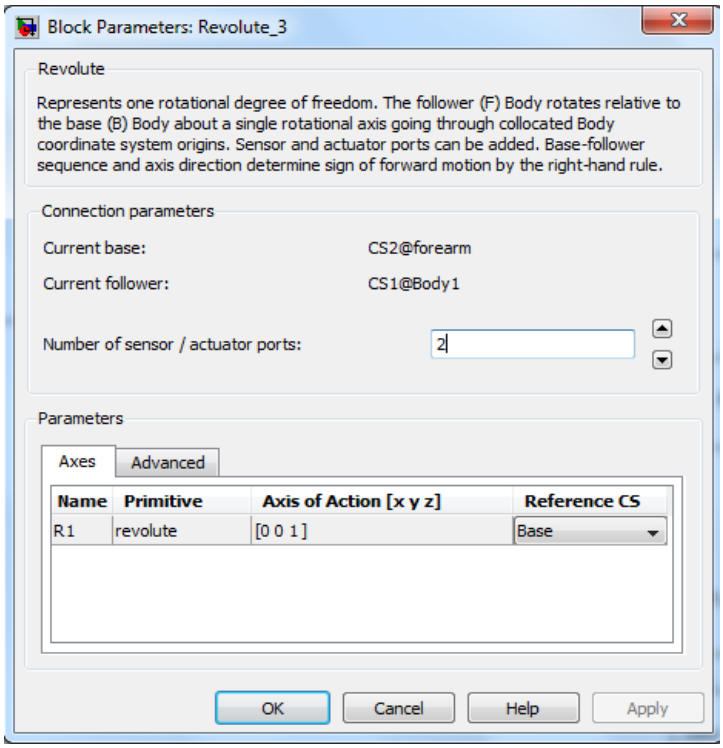

Figure 29. Joint3 (Revolute3)

Joint 4_1 (revolute4_1): Figure 30 shows the information of Joint4_1 (revolute4_1).

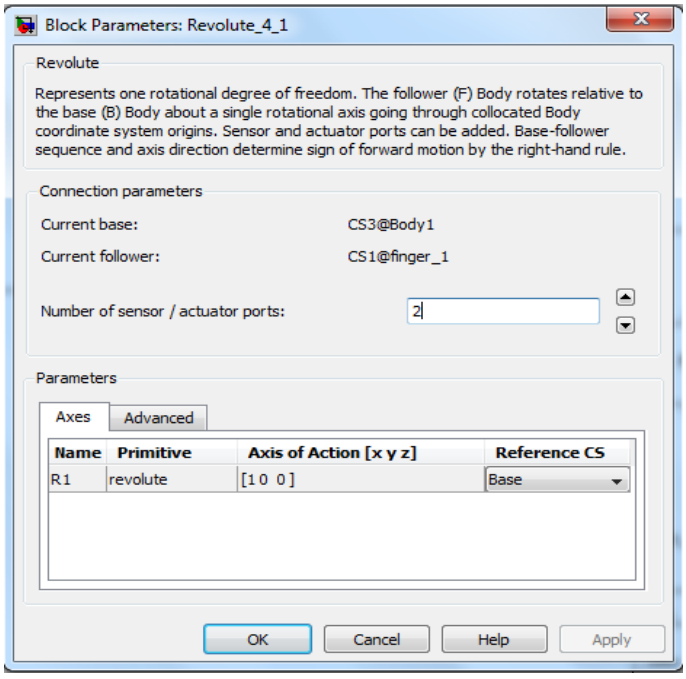

Figure 30. Joint4_1 (revolute4_1)

Joint 4_2 (revolute4_2): Figure 31 shows the information of Joint4_2 (revolute4_2).

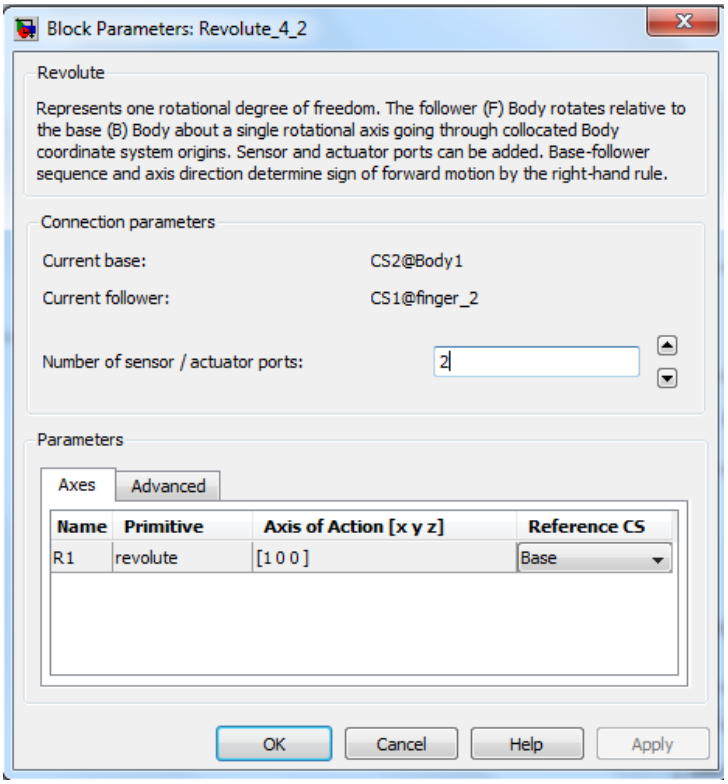

Figure 31. Joint4_2 (Revolute4_2)

Joint sensor: We have 5 joint sensors in this system that their information is the same. Figure 32 shows the joint sensor information.

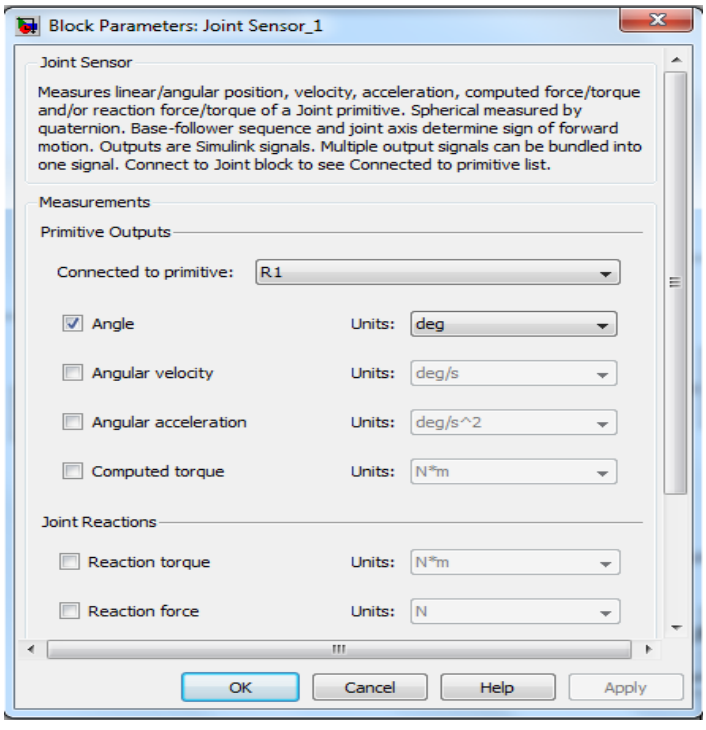

Figure 32. Joint Sensors

Joint actuator in this system, 5 joint actuators are implemented like as sensors. Figure33 shows the information.

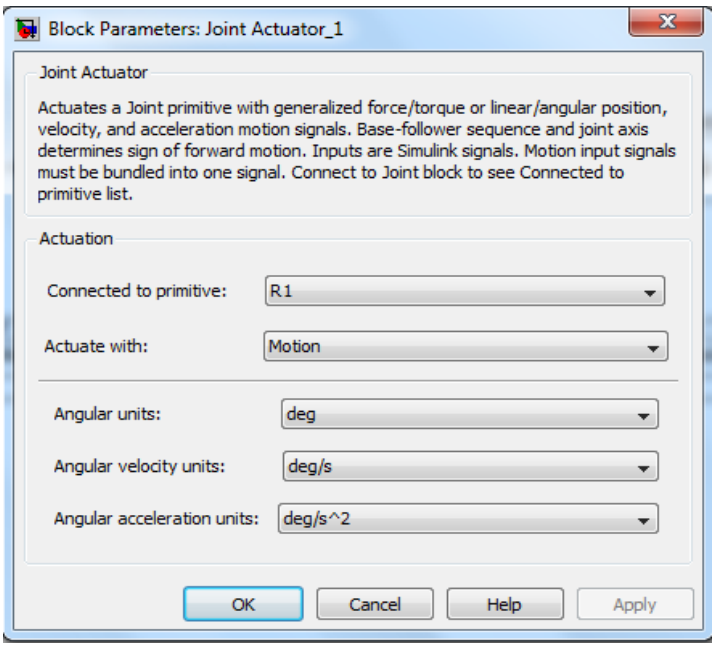

Figure33. Joint Actuators

Machine environment: machine environment should be as follow:

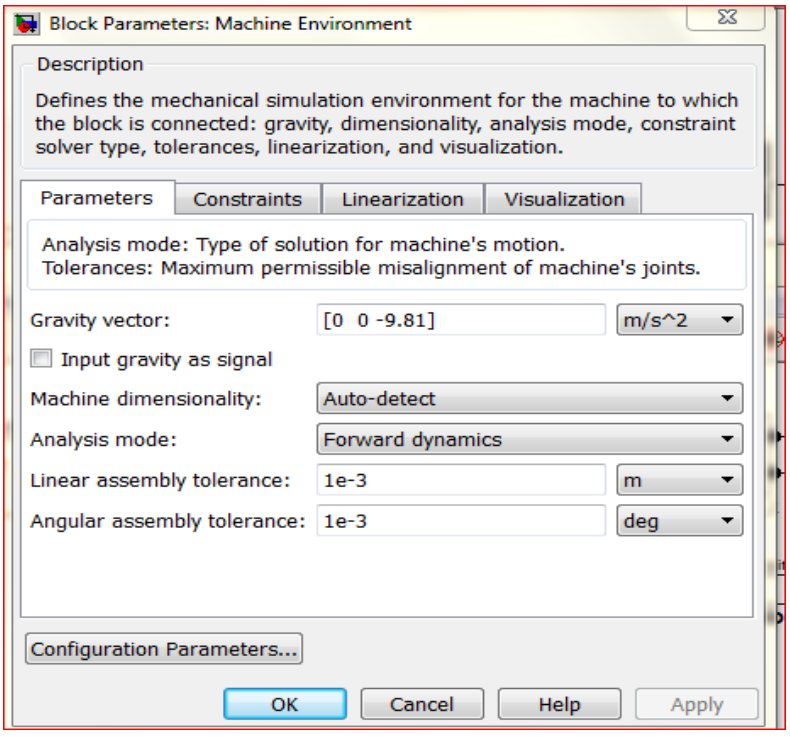

Figure 34. Machine Environment

Figure 35 shows the implementation of system.

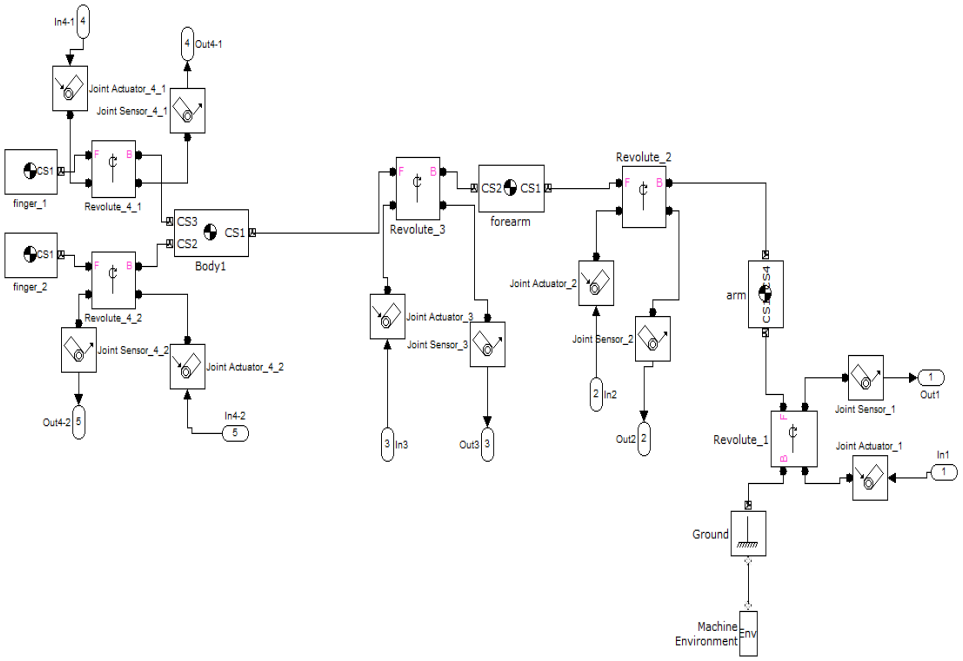

Figure 35. System's Model

After modeling and subsysteming the system, slider gains should be connected as follow:

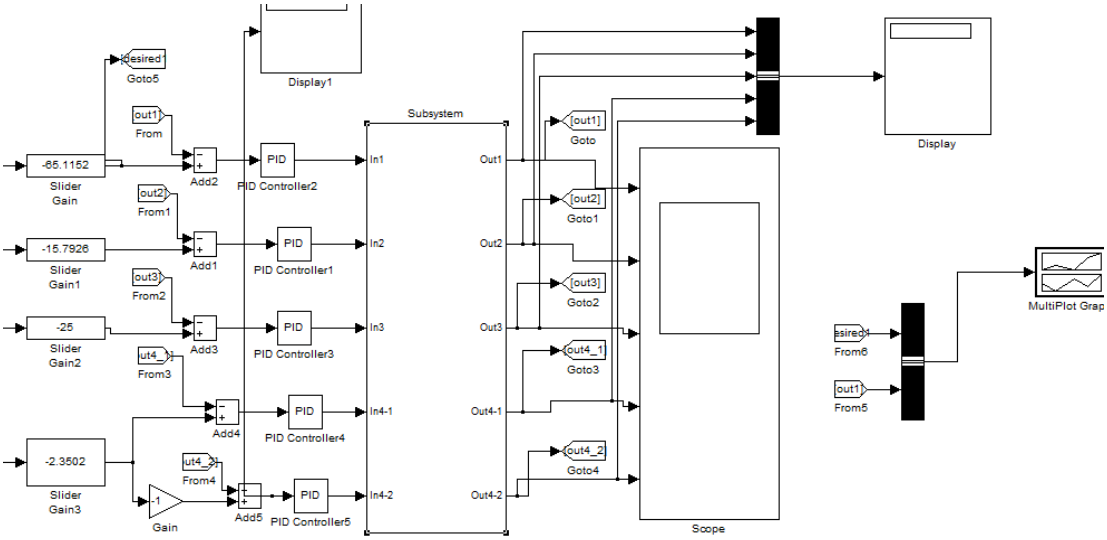

Figure36. System's Model

Figure 37. shows the value of slider gains.

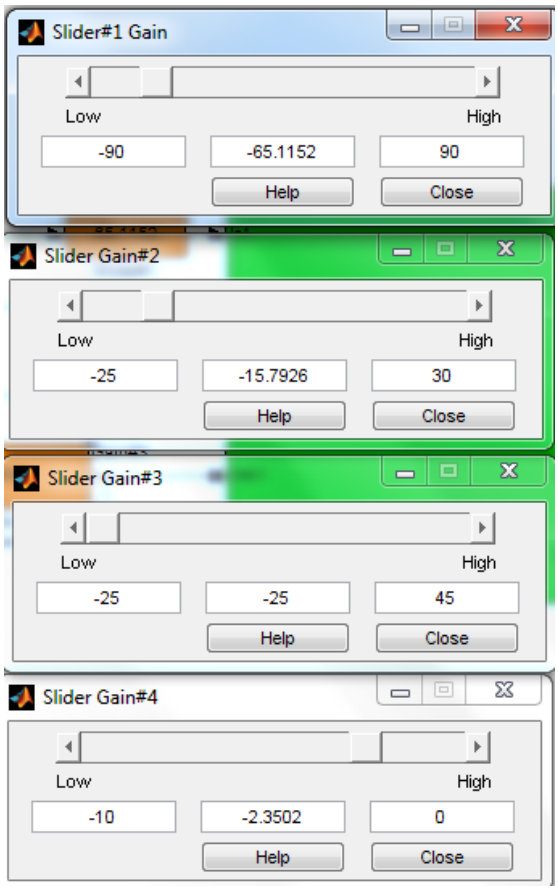

Figure 37. Slider Gains Value

System's physical model will be as follow:

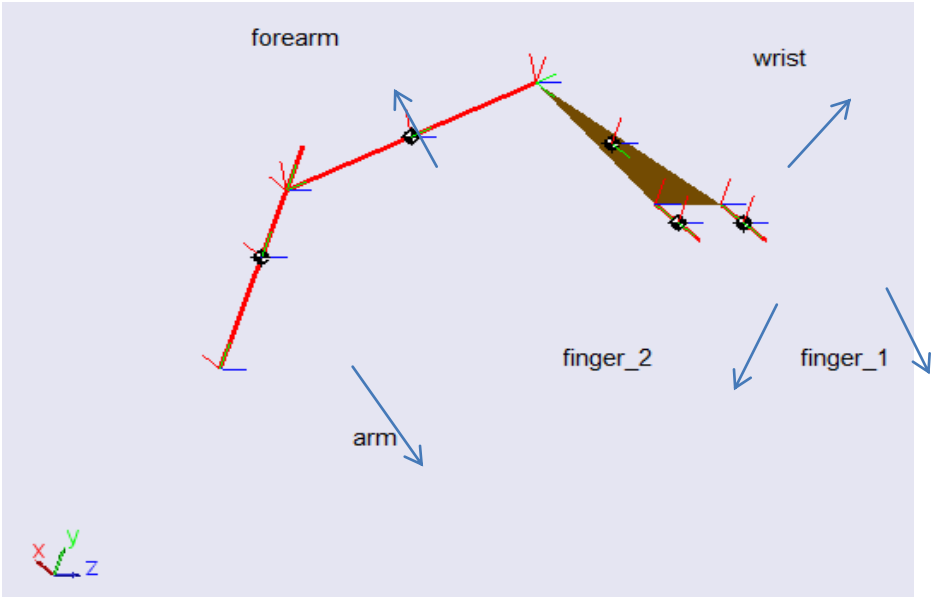

Figure 38. System's Physical Model

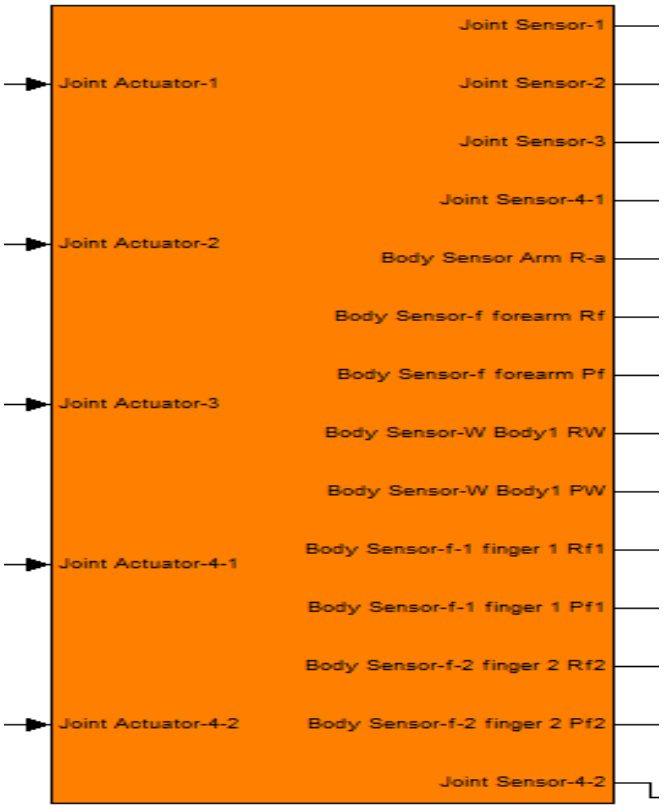

Figure 39. 4 DOF Robot Manipulator

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;

$$\boldsymbol{U_{PD} = K_p \times e + K_v(\frac{de}{dt}) = K_p \times e + K_v \dot{e}} \quad \textbf{(72)}$$

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 40 shows the block diagram of Proportional-Derivative (PD) controller.

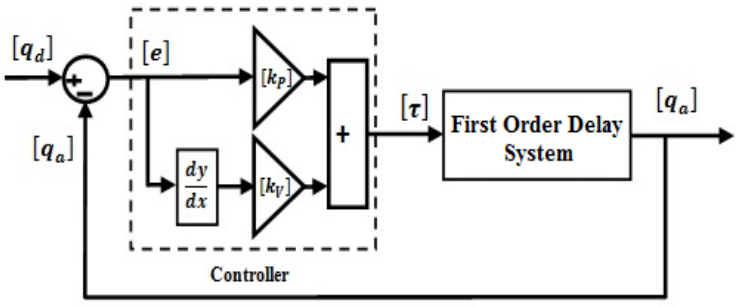

Figure 40. Block Diagram Of PD Controller

Figure 41 shows the ramp response of PD controller.

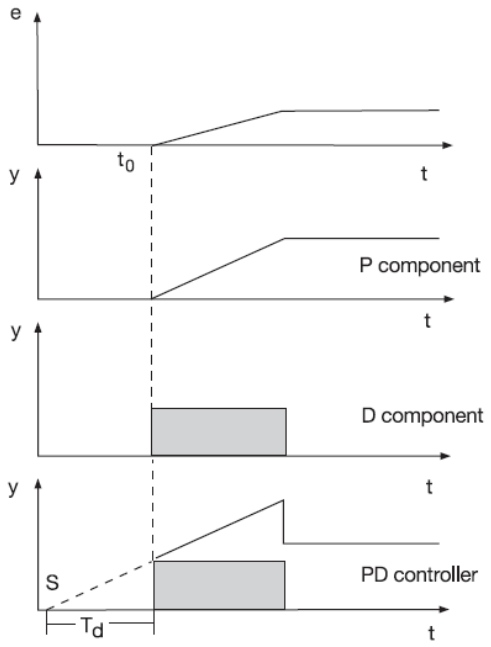

Figure 41. Ramp Response of a PD Controller [13]

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 42 shows the block diagram of PI control of FOD system.

$$\boldsymbol{U_{PI} = K_p \times e + K_i(\frac{1}{T}\int e.dt) = K_p \times e + K_i \sum e} \quad \textbf{(73)}$$

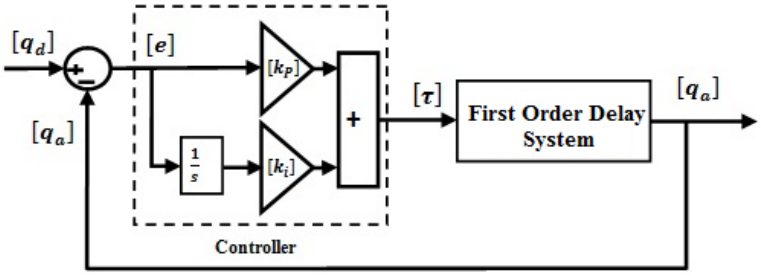

Figure 42. Block Diagram of PI Controller for First Order Delay System

Figure 43 shows the step response of PI controller.

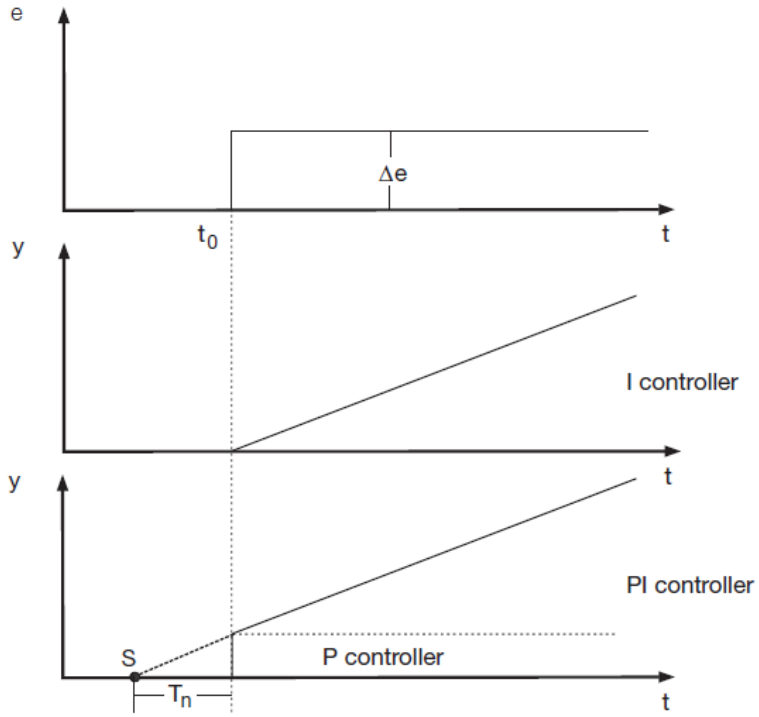

Figure 43. Step Response of a Proportional-Integral (PI) Controller [13]

3. Methodology

Design PID Control Algorithm: 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 variables and the actual variables to control the FOD systems. 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 44 shows the block diagram of PID control of FOD system. The formulation of PID controller calculated as follows;

$$\boldsymbol{U_{PID} = K_p \times e + K_i(\frac{1}{T}\int e.dt) + K_v(\frac{de}{dt}) = K_p \times e + K_i \sum e + K_v \dot{e}} \quad \textbf{(74)}$$

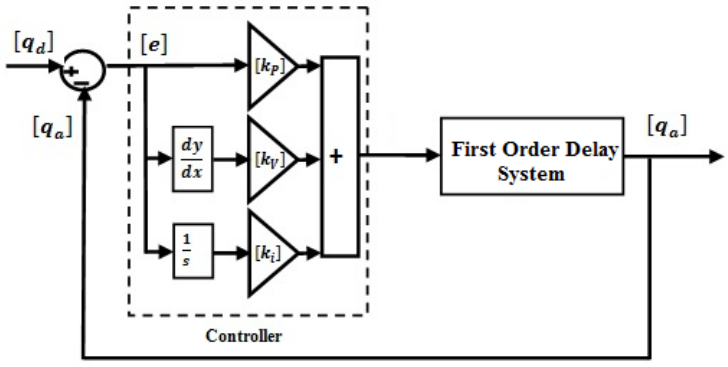

Figure 44. Block Diagram of PID Control of FOD System

Figure 45 shows the step response of PID controller.

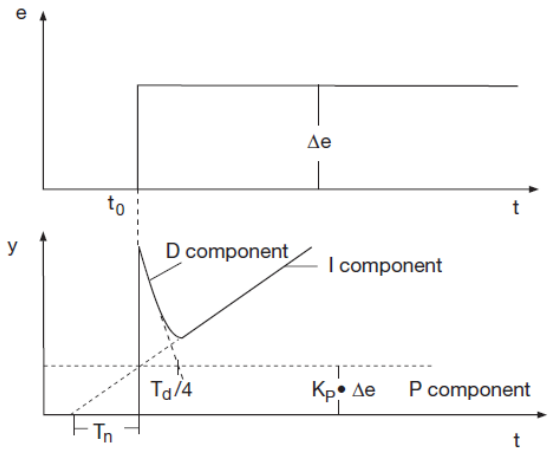

Figure 45. Step Response of a Proportional-Integral-Derivative (PID) Controller Design FPGA-Based Derivative Algorithm

The following formulation shows the derivative algorithm:

$$\boldsymbol{d(e) = \frac{Din(t) - Din(t-1)}{\Delta t} = \big(Din(k+1) - Din(k)\big) \times sample\ time} \quad \textbf{(75)}$$

$$\boldsymbol{Din = q_d - q_a} \quad \textbf{(76)}$$

However $\boldsymbol{q_d}$ **and** $\boldsymbol{q_a}$ are 30 bits but Din is 40 bits. In derivative algorithm, delay time is the main challenge. In this research the value of sample time is"01010".

To design $\boldsymbol{Din(k+1)}$, design a register has the main role. The vast majority of modern commercial systems are built with registers using positive edge-triggered D flip-flops. A group of cascaded flip flops used to store related bits of information is known as a register. Figure 46 shows D flip-flop.

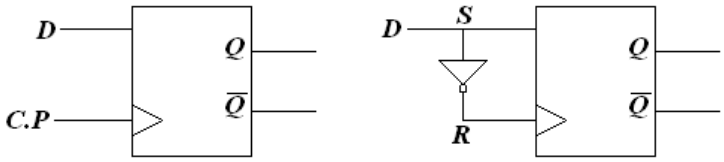

Figure 46. D Flip Flop

Figure 47 shows the RTL schematic of FPGA-based register.

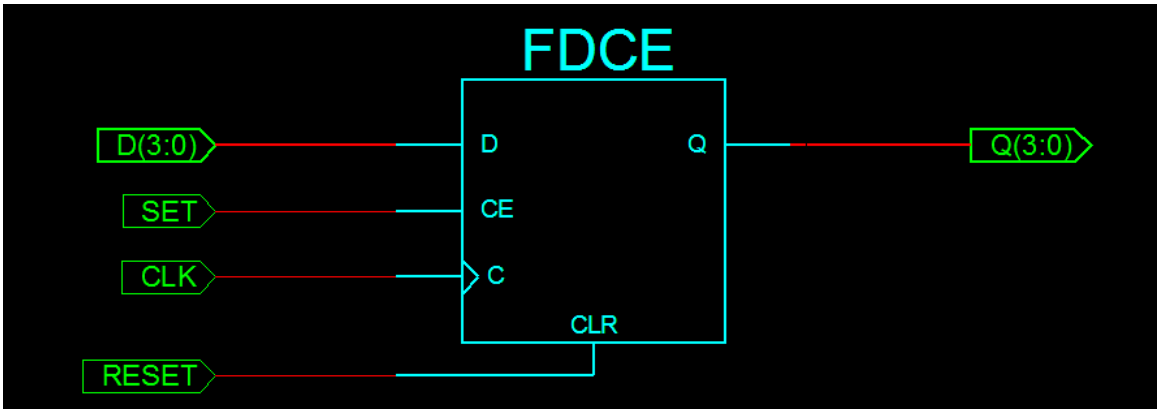

Figure 47. Outline Register Algorithm in HDL using Spartan 3E

Figure 48 shows the outline of FPGA-based derivative algorithm with 40 bits input and 40 bits output. Regarding to this algorithm, the derivation of input signal is calculated in output.

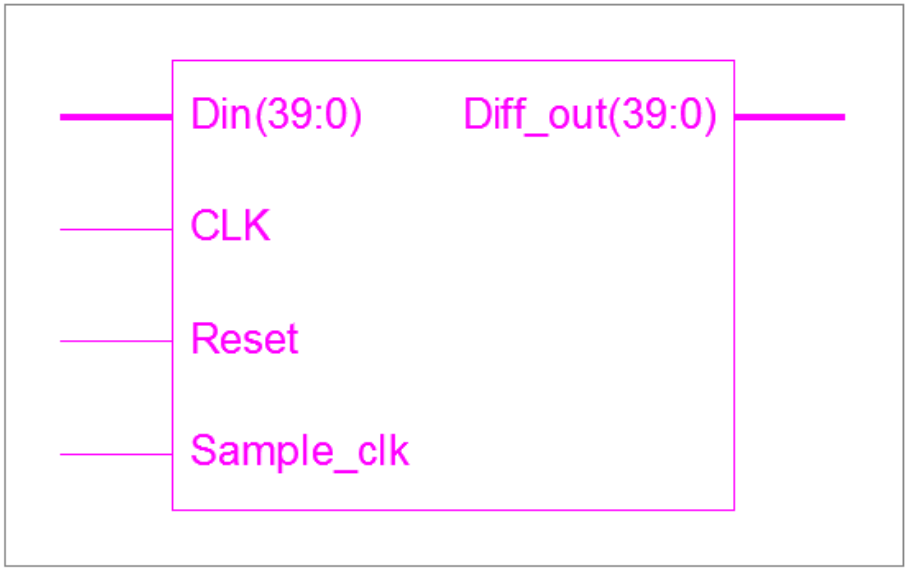

Figure 48. Outline FPGA-based Derivative Algorithm

Figure 49 shows the interior view FPGA-based derivative algorithm using VHDL program. Regarding to this algorithm, CLK is used to synchronize three types register which used in this algorithm and SAMPLE-CLK is used to system synchronization.

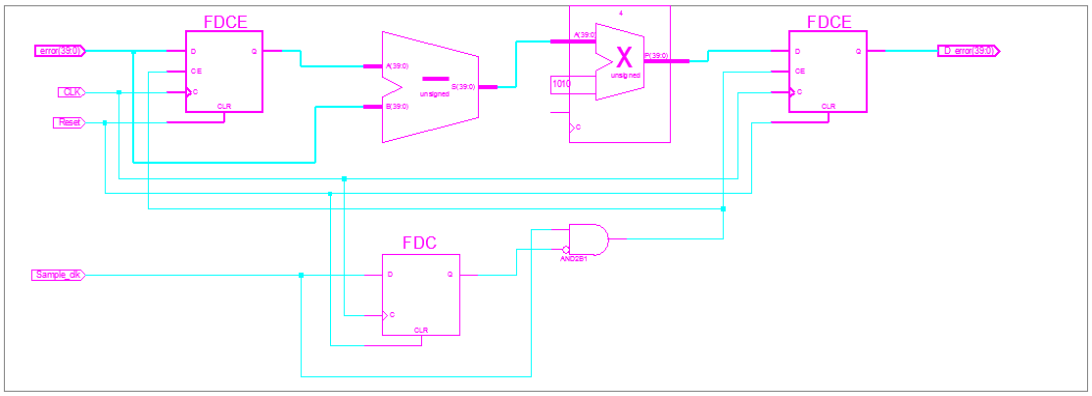

Figure 49. Interior View: FPGA-Based Derivative Algorithm

The VHDL code which applied to automotive Spartan 3E-XA3S1600E shows in Figure 50.

```
entity Derivative_control is
   Port(error : in  STD_LOGIC_VECTOR (39 downto 0);
        D_error : out  STD_LOGIC_VECTOR (39 downto 0);
        Reset : in  STD_LOGIC;
        CLK : in  STD_LOGIC;
        Sample_clk : in  STD_LOGIC);
end Derivative_control;

architecture Behavioral of Derivative_control is
----------------------------------------------------------------------------------------------------
constant sample_rate : std_logic_vector(4 downto 0) := "01010";
signal last_error : std_logic_vector(39 downto 0);
signal data_sample_error : std_logic_vector(39 downto 0);
signal diff_data : std_logic_vector(44 downto 0);
signal last_sample_clk : std_logic;
signal sample_clk_edge : std_logic;
----------------------------------------------------------------------------------------------------
begin
----------------------------------------------------------------------------------------------------
sample_clk_edge <= (not last_sample_clk) and Sample_clk;
process(CLK, Reset)
begin
   if(Reset = '1')then
      last_sample_clk <= '0';
   elsif(rising_edge(CLK))then
      last_sample_clk <= Sample_clk;
   end if;
end process;
----------------------------------------------------------------------------------------------------
process(CLK)
begin
   if(Reset = '1')then
      last_error <= (others => '0');
      D_error <= (others => '0');
   elsif(rising_edge(CLK) and sample_clk_edge = '1')then
```

Figure 50. VHDL Code: FPGA-Based Derivative Algorithm

Design FPGA-Based Integral Algorithm: The following formulation shows the derivative algorithm:

$$\boldsymbol{I(out)_{(K)} = I(out)_{(k-1)} + (Ki \times e(k) \times sample\ time)} \quad \textbf{(77)}$$

Figure 51 shows the outline of FPGA-based integral algorithm with 40 bits input and 40 bits output.

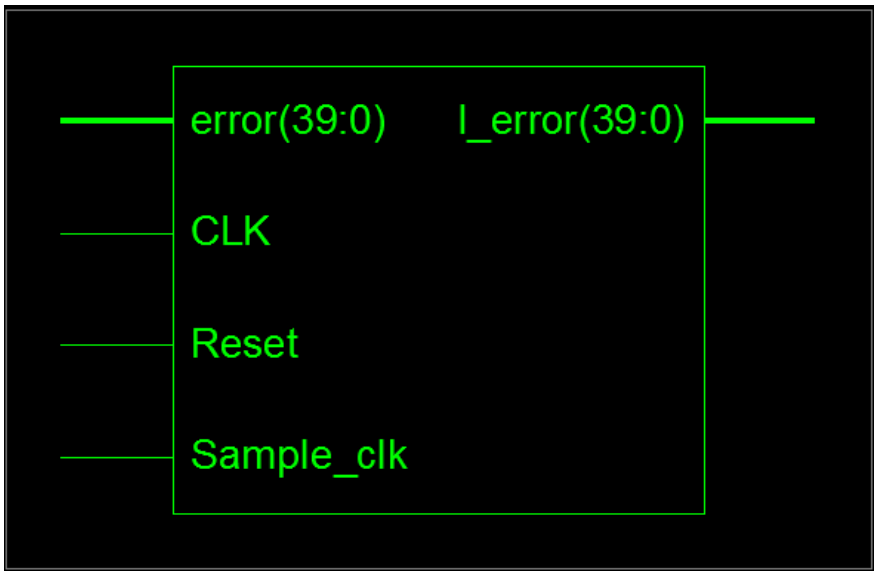

Figure 51. Outline FPGA-based Integral Algorithm

Figure 52 shows the interior view FPGA-based derivative algorithm using VHDL program. Regarding to this algorithm, CLK is used to synchronize three types register which used in this algorithm and SAMPLE-CLK is used to system synchronization.

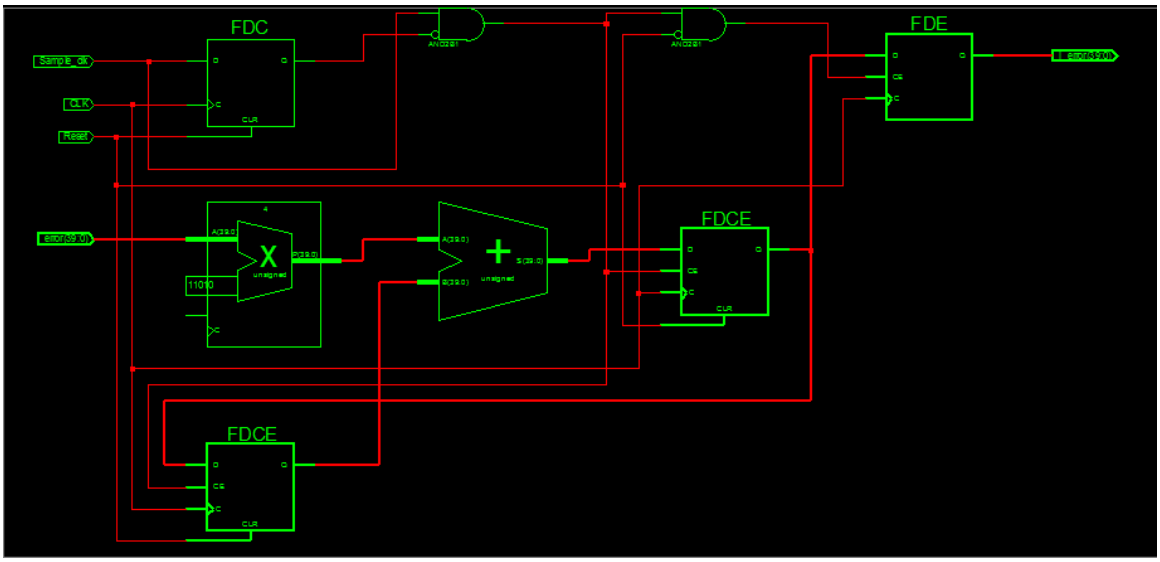

Figure 52. Interior View: FPGA-Based Integral Algorithm

The integral VHDL code which applied to automotive Spartan 3E-XA3S1600E shows in Figure 53.

```
entity I_control is
   Generic (Ki : std_logic_vector(4 downto 0) := B"11010");
   Port(error : in  STD_LOGIC_VECTOR (39 downto 0);
        I_error : out  STD_LOGIC_VECTOR (39 downto 0);
        Reset : in  STD_LOGIC;
        CLK : in  STD_LOGIC;
        Sample_clk : in  STD_LOGIC);
end I_control;

architecture Behavioral of I_control is
------------------------------------------------------------------
--constant sample_rate : std_logic_vector(4 downto 0) := "01010";
signal last_I : std_logic_vector(39 downto 0);
signal data_sample_sub : std_logic_vector(39 downto 0);
signal I_data : std_logic_vector(44 downto 0);
signal I_error_sig : std_logic_vector(39 downto 0);
signal last_sample_clk : std_logic;
signal sample_clk_edge : std_logic;
------------------------------------------------------------------
begin
------------------------------------------------------------------
sample_clk_edge <= (not last_sample_clk) and Sample_clk;
process(CLK, Reset)
begin
   if(Reset = '1')then
      last_sample_clk <= '0';
   elsif(rising_edge(CLK))then
      last_sample_clk <= Sample_clk;
   end if;
```

Figure 53. VHDL Code: FPGA-based Integral Algorithm

The Z formulation of PID controller is design as follows:

$$\boldsymbol{U[k] = U[k-1] + K_1 \times e[k] + K_2 \times e[k-1] + K_3 \times e[k-2]} \quad \textbf{(78)}$$

In this algorithm, we have five inputs (actual input, desired input, CLK, reset and SAMPLE-CLK) and an output (PID-Control). Actual and desired inputs are 30 bits and PID control output is 35 bits. In this design, CLK is used to activate the PID sub-parts (registers), reseat is used to re-start of registers and SAMPLE-CLK is used to synchronization proportional, derivate and integral parts. Figure 54 shows the outline of FPGA-based PID control algorithm.

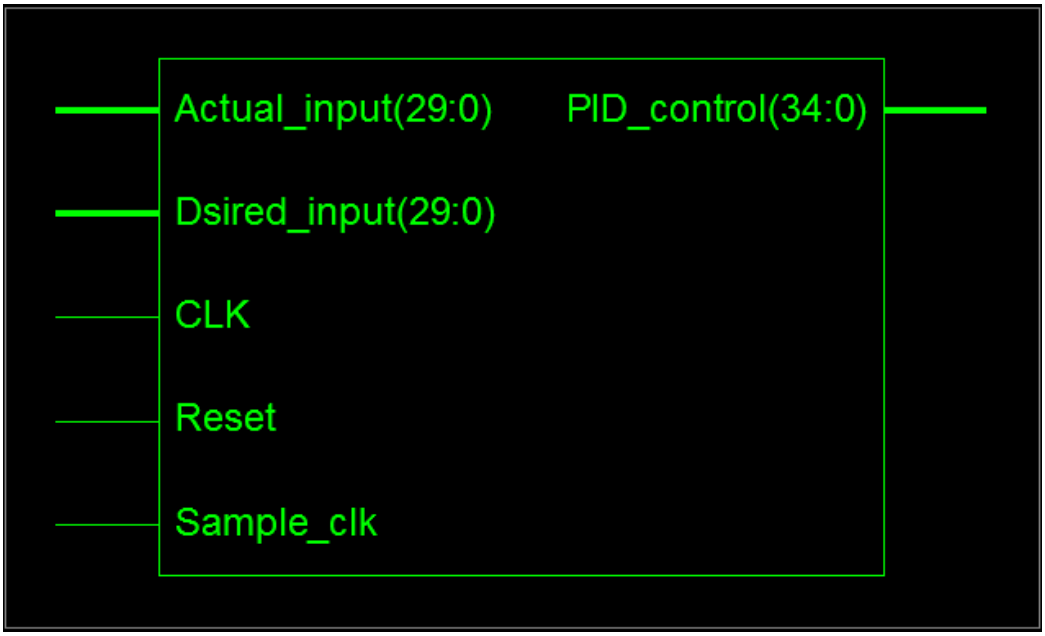

Figure 54. Outline: FPGA-based PID Algorithm

The interior view of PD algorithm shows in Figure 55.

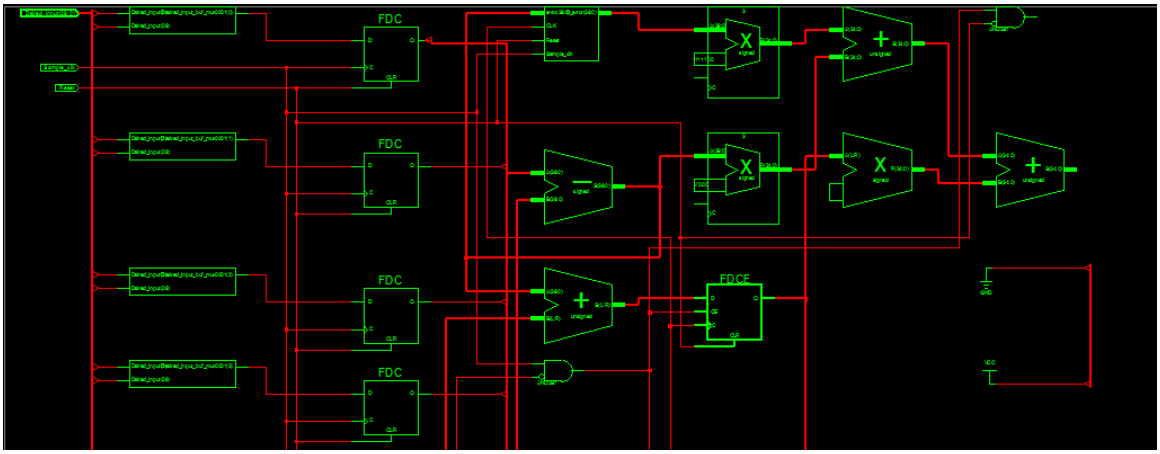

Figure 55. Interior View: FPGA-based PID Algorithm

Regarding to Figure 56, the PID algorithm have some buffers for desired and actual inputs. Figure 57 shows the interiors of buffer.

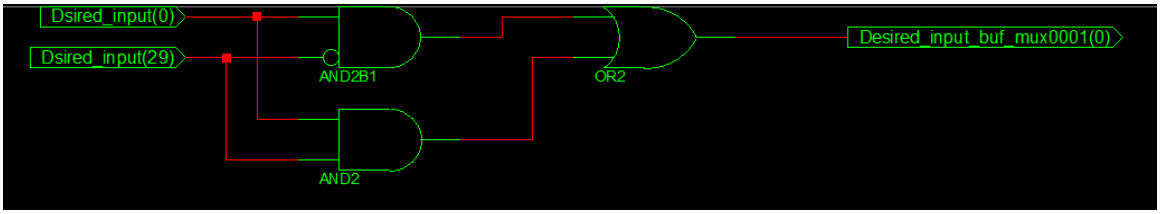

Figure 56. Interior View: FPGA-Based Buffer

The VHDL program of FPGA-based PID controller shows in Figure 57.

```
entity PID_Controler is
      Generic(Kp : std_logic_vector(7 downto 0) := B"11111010";
              Kv : std_logic_vector(7 downto 0) := B"00111100";
              Ki : std_logic_vector(7 downto 0) := B"00111100");
      Port(Actual_input : in  STD_LOGIC_VECTOR (29 downto 0);
        Dsired_input : in  STD_LOGIC_VECTOR (29 downto 0);
        PID_control : out  STD_LOGIC_VECTOR (34 downto 0);
        Reset : in  STD_LOGIC;
        CLK : in  STD_LOGIC;
        Sample_clk : in  STD_LOGIC);
end PID_Controler;
--------------------------------------------------------------------------------
architecture Behavioral of PID_Controler is
--------------------------------------------------------------------------------
------Signals definitions
--------------------------------------------------------------------------------
signal Actual_input_buf : std_logic_vector(39 downto 0);
signal Desired_input_buf : std_logic_vector(39 downto 0);
signal Error_gain : std_logic_vector(47 downto 0);
signal Error_diff : std_logic_vector(39 downto 0);
signal Error_integral : std_logic_vector(39 downto 0);
signal Error : std_logic_vector(39 downto 0);
signal Error_diff_gain : std_logic_vector(47 downto 0);
signal Error_integral_gain : std_logic_vector(47 downto 0);
signal PID_control_buf : std_logic_vector(39 downto 0);
```

Figure 57. VHDL Code: FPGA-based PID Algorithm

The device utilization summary shows in Figure 58.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	219	14752	1%
Number of Slice Flip Flops	153	29504	0%
Number of 4 input LUTs	264	29504	0%
Number of bonded IOBs	98	304	32%
Number of MULT18X18SIOs	12	36	33%
Number of GCLKs	2	24	8%

Figure 58. Device Utilization Summaries

4. Test and Result

In this research linear controlling had implemented in simscape using MATLAB/SIMULINK and XILINX using FPGA. All joints are moved from home to final position without external disturbance. Controller's coefficient calculate by test and error.

PID controller

By checking and review the result of PD and PI controller and combining them, it can be concluded that PID controller is like PI controller with an extra parameter which has a little effect on controlling of system. So we can say that there is not much different between PID and PI controller method but PID controller is more accurate. Figure 59 shows the trajectory following of PID controller.

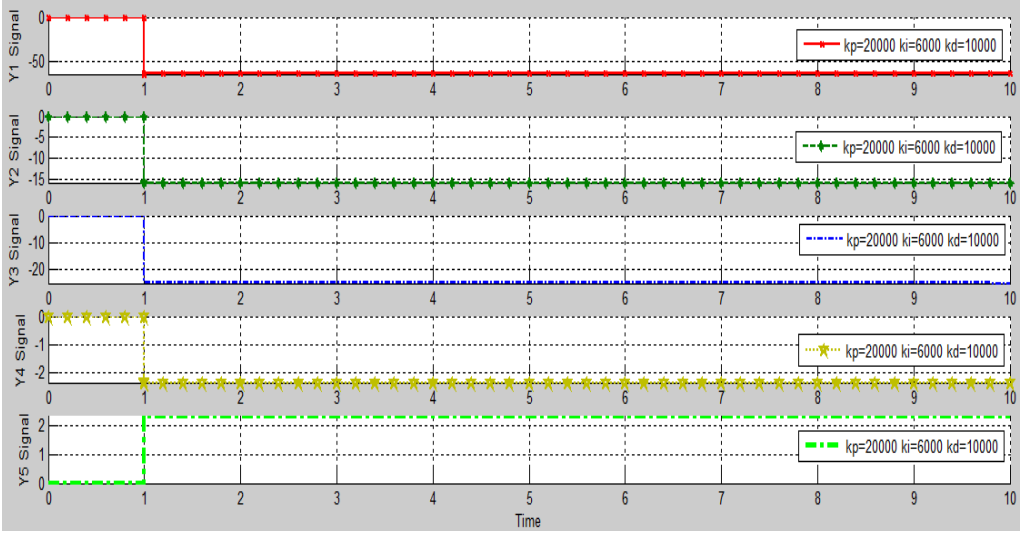

Figure 59: Trajectory Following Control by PID Controller

PID Algorithm:
Figure 60 shows the HDL synthesize report for PID algorithm.

```
HDL Synthesis Report

Macro Statistics
# Multipliers                                  : 4
 40x5-bit multiplier                          : 1
 40x8-bit multiplier                          : 3
# Adders/Subtractors                           : 5
 40-bit adder                                 : 3
 40-bit subtractor                            : 2
# Registers                                    : 9
 1-bit register                               : 2
 40-bit register                              : 7
```

Figure 60. HDL Synthesis Report: PID Algorithm

Figure 61 shows advanced HDL synthesis report in PID algorithm.

```
Advanced HDL Synthesis Report

Macro Statistics
# Multipliers                                  : 4
 40x5-bit multiplier                          : 1
 40x8-bit multiplier                          : 2
 40x8-bit registered multiplier               : 1
# Adders/Subtractors                           : 5
 35-bit adder                                 : 2
 40-bit adder                                 : 1
 40-bit subtractor                            : 2
# Registers                                    : 224
 Flip-Flops                                   : 224
```

Figure 61. Advanced HDL Synthesis Report: PID Algorithm

Figure 62 shows the device utilization summary in PID algorithm.

```
Device utilization summary:
---------------------------

Selected Device : xa3s1600efgg400-4

 Number of Slices:                      219  out of  14752     1%
 Number of Slice Flip Flops:            153  out of  29504     0%
 Number of 4 input LUTs:                264  out of  29504     0%
 Number of IOs:                          98
 Number of bonded IOBs:                  98  out of    304    32%
    IOB Flip Flops:                      60
 Number of MULT18X18SIOs:                12  out of     36    33%
 Number of GCLKs:                         2  out of     24     8%
```

Figure 62. Device Utilization Summary: PID Algorithm

Figure 63 shows the timing summary in PID algorithm.

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 15.716ns (Maximum Frequency: 63.629MHz)
   Minimum input arrival time before clock: 4.683ns
   Maximum output required time after clock: 22.296ns
   Maximum combinational path delay: No path found
```

Figure 63. Timing Summary: PID Algorithm

Regarding to Figure 64, the Maximum frequency in this design is **44.85 MHz**. however the rate of clock is 63.629 MHz but in PID algorithm the maximum frequency is 44.85MHz.

5. Conclusion

Refer to this paper, linear controler had designed in Simscape using MATLAB/SIMULINL and Xilinx based on FPGA. To improve the 4 DOF surgical joint's result, PID control algorithm is introduced. When PID controller is applied to system, caused to change the time response. The FPGA-based PID controller design based on derivative and integration algorithms. The maximum frequency of input clock pulse in this design is 63.629 MHz but the maximum output PID algorithm is about 44.85 MHz. In comparison with PD controller, PID controller is more stable but the maximum output frequency in PD (51.89 MHz) is better than PID (44.85 MHz).

Acknowledgements

This work was supported by the Iranian Institute of Advance Science and Technology Program of Iran under grant no. **2015-Persian Gulf-2.**

Project Title: Research on Intelligent FPGA-Based Algorithm for Four Degrees of Freedom Joints

Iranian center of Advance Science and Technology (IRAN SSP) is one of the independent research centers specializing in research and training across of Control and Automation, Electrical and Electronic Engineering, and Mechatronics & Robotics in Iran. At IRAN SSP research center, we are united and energized by one mission to discover and develop innovative engineering methodology that solve the most important challenges in field of advance science and technology. The IRAN SSP Center is instead to fill a long standing void in applied engineering by linking the training a development function one side and policy research on the other. This center divided into two main units:

- Education unit
- Research and Development unit

Please follow IRANSSP research and training group in: http://iranssp.org/english/

References

[1] R. Faraji, A. Rouholamini, H.R. Naji, R. Fadaeinedjad and M.R. Chavoshian, "FPGA-Based Real Time Incremental Conductance Maximum Power Point Tracking Controller for Photovoltaic Systems", Power Electronics, IET, vol. 7, no. 5, **(2014)**, pp. 1294-1304.

[2] S. Ghosh, R.K. Barai, S. Bhattarcharya, P. Bhattacharyya, S. Rudra, A. Dutta and R. Pyne, "An FPGA Based Implementation of a Flexible Digital PID Controller for a Motion Control System", In Computer Communication and Informatics (ICCCI), 2013 International Conference on, IEEE, **(2013)**, pp. 1-6.

[3] K. Lochan and B.K. Roy, "Control of Two-link 2-DOF Robot Manipulator Using Fuzzy Logic Techniques: A Review", In Proceedings of Fourth International Conference on Soft Computing for Problem Solving, Springer India, **(2015)**, pp. 499-511.

[4] J. Li, L. Liu, Y. Wang and W. Liang, "Adaptive Hybrid Impedance Control of Robot Manipulators With Robustness Against Environment's Uncertainties", In Mechatronics and Automation (ICMA), 2015 IEEE International Conference on, IEEE, **(2015)**, pp. 1846-1851.

[5] Y.Q. Bao and Y. Li, "FPGA-Based Design of Grid Friendly Appliance Controller", Smart Grid, IEEE Transactions on, vol. 5, no. 2, **(2014)**, pp. 924-931.

[6] B. Siciliano and O. Khatib, "Springer Handbook of Robotics. Springer Science & Business Media, Eds., **(2008)**.

[7] F. Piltan, A. Taghizadegan and N. Sulaiman, "Modelling and Control of Four Degrees of Freedom Surgical Robot Manipulator Using MATLAB/SIMULINK", International Journal of Hybrid Information Technology, http://dx.doi.org/10.14257/ijhit.2015.8.11.05, vol. 8, no.11, **(2015)**, pp. 47-78.

[8] K. D. Young, V. Utkin and U. Ozguner, "A Control Engineer's Guide to Sliding Mode Control", IEEE International Workshop on Variable Structure Systems, **(2002)**, pp. 1-14.

[9] O. Kaynak, "Guest Editorial Special Section on Computationally Intelligent Methodologies and Sliding-Mode Control", IEEE Transactions on Industrial Electronics, vol. 48, no. 1, **(2001)**, pp. 2-3.

[10] I. Boiko, L. Fridman, A. Pisano and E. Usai, "Analysis of Chattering in Systems with Second-Order Sliding Modes", IEEE Transactions on Automatic Control, vol. 52, no. 11, **(2007)**, pp. 2085-2102.

[11] V. Utkin, "Variable Structure Systems with Sliding Modes", IEEE Transactions on Automatic Control, vol. 22, no. 2, **(2002)**, pp. 212-222.

[12] P. I. Corke and B. Armstrong-Helouvry, "A search for Consensus Among Model Parameters Reported for the PUMA 560 Robot", IEEE International Conference on Robotica and Automation, **(1994)**, pp. 1608-1613.

[13] F. Piltan, N. Sulaiman, M. H. Marhaban, A. Nowzary and M. Tohidian, "Design of FPGA-based Sliding Mode Controller for Robot Manipulator", International Journal of Robotic and Automation, vol. 2, no.3, **(2011)**, pp. 183-204.

[14] F. Piltan, N. Sulaiman, A. Jalali and K. 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, vol. 2, no.5, **(2011)**, pp. 317-343.

[15] F. Piltan, I. Nazari, S. Siamak and P. Ferdosali, "Methodology of FPGA-Based Mathematical error-Based Tuning Sliding Mode Controller", International Journal of Control and Automation, vol.5, no.1, **(2012)**, pp. 89-118.

Authors

Ali Taghizadegan, He 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.

Farzin Piltan, He 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.

Nasri Sulaiman, He 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)

