IJCCCE Vol.15, No.2, 2015

Modeling and Control of 5250 Lab-Volt 5 DoF Robot Manipulator †

DR. Wathik Isaa Mahdi AL-Tameemi [1], Mr. Wael Mohammed Hasan Hadi [2]

[1,2]Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

e-mail: w.altameemi@uotechnology.edu.iq, alburaq_10@yahoo.com

Received: 14/12/2014

Accepted: 19/5/2015

Abstract – This paper presents the modeling and control simulation for Lab-Volt 5250 five degree of freedom robot manipulator based on the standard Denavit-Hartenberg approach. The dynamic model of the robot derived using Euler- Lagrange equation which is the energy balance equation. This dynamic model has a very high nonlinearity that is represented by using MATLAB, m-file and simulation to run the dynamic model in open and close loop. In this research, the close loop simulation is done by using two types of control theory that applied to control each joint of the robot manipulator independently, the first one is PD controller and the second one is an intelligent controller which is PD-like fuzzy controller used to control the joint position.

.

Keywords – Denavit- Hartenberg, Dynamics, PD, Fuzzy.

† *This paper has been presented in ECCCM-2 Conference* and accredited for publication according to IJCCCE rules.

1. Introduction

The implementation of dynamic control systems for manipulators has been hampered because the models are difficult to derive and computationally expensive, and because the needed parameters of the manipulator are generally unavailable [1]. This paper presents the dynamics and control of robot manipulators. Whereas the kinematic equations describe the motion of the robot without consideration of the forces and torques producing the motion, while the dynamic equations explicitly describe the relationship between force and motion. The equations of motion are important to consider in the design of robots, in simulation and animation of robot motion, and in the design of control algorithms. The use of the Euler-Lagrange equation is considered to drive the dynamics of robot manipulator which is derived from the principle of virtual work, so in order to drive the Euler-Lagrange equations one has to form the Lagrangian of the system, which is the difference between the kinetic energy and the potential energy. The Euler-Lagrange equations have several very important properties that can be exploited to design and analyze feedback control algorithms. Among these are explicit bounds on the inertia matrix, linearity in the inertia parameters, and the so-called skew symmetry and passivity properties. The Euler-Lagrange also can be derived based on Hamilton's principle of least action [2]. The dynamical system can be controlled in open loop or close loop. In open loop, the input is computed without observing the output that is to be controlled. In the complex systems the open loop controller isn't possible, because the controller will never know if the output reaches to the desired target. So the use of feedback controller will achieve the desired target.

2. Dynamic Model

At first the forward kinematics and velocity kinematics must be driven to use them in driving the dynamical model of the robot manipulator, see appendix.

The Euler-Lagrange equation used to derive the dynamical model of the Lab-Volt 5250 robot manipulator, which is the difference between the kinetic energy and the potential energy.

$$\boldsymbol{L = K - P} \tag{1}$$

Where K is the kinetic energy and P is the potential energy

$$K = \frac{1}{2} m \dot{y}^2 \tag{2}$$

$$P = mgy \tag{3}$$

$$\frac{\partial L}{\partial \dot{y}} = \frac{\partial K}{\partial \dot{y}} \text{and } \frac{\partial L}{\partial y} = -\frac{\partial P}{\partial y} \text{then}$$

$$\frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = f \tag{4}$$

The function L, which is the difference between the kinetic and P potential energy, is called the Lagrangian of the system, m mass, g acceleration of gravity and y height, (4) is called the Euler-Lagrange Equation. The kinetic energy of n-link robot manipulator composed of two components linear and angular kinetic energy, so the need for derivation of manipulator Jacobian is important because the Jacobian consists of both linear and angular velocities which are component of the kinetic energy where the Jacobian matrix expressed as follows

$$v_i = J_{v_i}(q)\dot{q}, \qquad w_i = J_{w_i}(q)\dot{q} \tag{5}$$

Where v_i and w_i are linear and angular velocities, q is the joint variable which is revolute or prismatic, J_{vi} and J_{wi} are linear and angular the Jacobian matrix. Suppose the mass of link i is m_i, and that the inertia matrix of link i equal I_i then the kinetic energy of the manipulator equal

$$K = \frac{1}{2}\dot{q}^T \sum_{i=1}^{n}[m_i J_{v_i}(q)^T J_{v_i}(q) + J_{w_i}(q)^T R_i(q) I_i R_i(q)^T J_{w_i}(q)]\dot{q} \quad (6)$$

In other words, the kinetic energy of the manipulator is of the form

$$K = \frac{1}{2}\dot{q}^T D(q)\dot{q} \quad (7)$$

Where D(q) is a symmetric positive definite matrix that is in general configuration dependent and is called the inertia matrix.

The potential energy in the case of rigid dynamics, the only source of potential energy is gravity. The potential energy of the i^{th} link can be computed by assuming that the mass of the entire object is concentrated at its center of mass and is given by

$$P_i = g^T r_{c_i} m_i \quad (8)$$

Where g is the vector giving the direction of gravity in the inertial frame and the vector r_{ci} gives the coordinates of the center of mass of link i. The total potential energy of the n-link robot is therefore

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} g^T r_{c_i} m_i \quad (9)$$

After simplification of the kinetic energy and potential energy and substitute them in the Euler-Lagrange equation the result is

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \quad (10)$$

Where D(q) is the inertia matrix, $C(q,\dot{q})$ is centrifugal and coriolis forces as shown in (11), g(q) is the gravity force and τ is the torque applied to joints and for the complete dynamics model of the robot see Appendix (A).

$$C_{kj} = \sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i = \sum_{i=1}^{n}\left[\frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_j} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\right] \quad (11)$$

3. Parameter Estimation

This section describes how the dynamic parameters are estimated quite roughly to ensure that the parameters are closed enough to the real unknown parameters. The center of mass of the Lab-Volt robot arm's joints can be estimated from assuming the links have uniform mass density so it is easy to calculate the center of mass by studying the manipulator thoroughly and allocating the center of mass for each link. But this method is not accurate, so the method which is used in this research for finding the center of mass quite roughly is the method of centroid. This provides accurate estimation for the center of mass of each link. Based on the centroid method the center of mass for each link will expressed as c_i for each link while link 4 is merged with link 5. Estimating the inertia parameters are definitely the most difficult task. The irregular shapes of the links makes it highly complicated to come up with realistic parameters without performing some kind of identification. As a fair simplification the links are modeled as cylindrical links with uniform mass density, where the estimating of the inertia parameters is definitely the most difficult task. As a fair simplification, the links are modeled as cylindrical links with uniform mass density; where the center of mass of each link is the geometric center of the cylinder [3]. Figure (1) shows an example of how this simplification can be applied on link 3 and the other links are same.

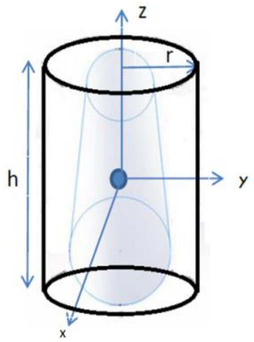

Figure 1. Link 3 modeled as cylinder

The inertia tensor of such a cylinder is [4]

$$I = \begin{bmatrix} \frac{1}{12}mh^2 + \frac{1}{4}mr^2 & 0 & 0 \\ 0 & \frac{1}{12}mh^2 + \frac{1}{4}mr^2 & 0 \\ 0 & 0 & \frac{1}{2}mr^2 \end{bmatrix} \quad (12)$$

Where m is the mass, r is the radius and h is the height of the cylinder. The cross products are identically zero such that the inertia tensor becomes a diagonal matrix in its principal axis form. The mass of each link can be determined from the approximated shape for each link as a cylinder but this is not accurate way of determined the masses of these links, so the calculation of the masses can be done by determining the geometry of each link and the material that these links were made from known, so the density of the material is known, also the other components of the links are known then, the calculation of the mass for each link was done. In our case the base of the robot was made from the steel, so the density of the steel is equal to 7830 kg/m^3 according to [5], and the other links was made from the aluminum, so the density of the aluminum is 2690 kg/m^3 according to [5], so the calculation of the mass for these links can be done. The following table of the parameters to determine the inertia tensor for the cylinder

Table 1. Cylinder parameters

link	Mass (kg)	Radius (m)	Height (m)
1	12.26	0.1613	0.28
2	1.515	0.045	0.38
3	0.94	0.0375	0.23
4	-	-	-
5	0.5024	0.105	0.15

4. Simulation of Open Loop Model

In Simulink a so-called interpreted MATLAB function was used to implement the dynamic model of the robot manipulator. This is a block with multiple input and output ports where inputs are the applied torque for each joint while outputs are the actual joint angles derivatives. Ode45 is used for solving the differential equations of the dynamical model.

In open loop there is no feedback from the system output. In other words, no information about the joint variables and its derivatives is available when computing the input torque. Due to the excitation of gravity on the links being dependent on the joint variables, it is quite intuitive that controlling the system in open loop is impossible. The behavior of the system can be studied by driving the system with the desired torque that is the constant torque derived when substituting in the dynamic equations for the desired joint variables and derivatives [3].

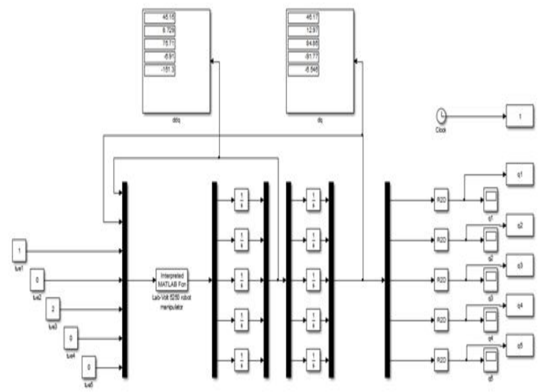

Figure 2. Open loop Simulink model

The applied torque for each joint that represented as a vector is $\tau = [1\ 1.5\ 2\ 2.5\ 3]^T$, so the behavior of the system in open loop model as shown in Figures 3 – 7.

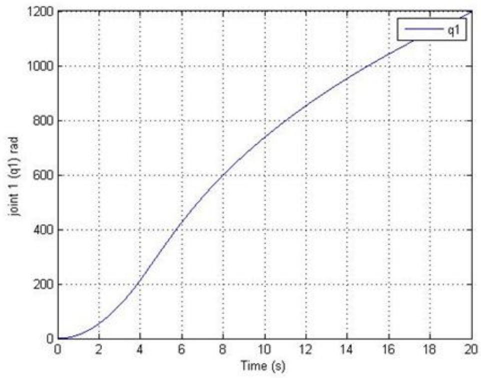

Figure 3. Joint 1 response in open loop

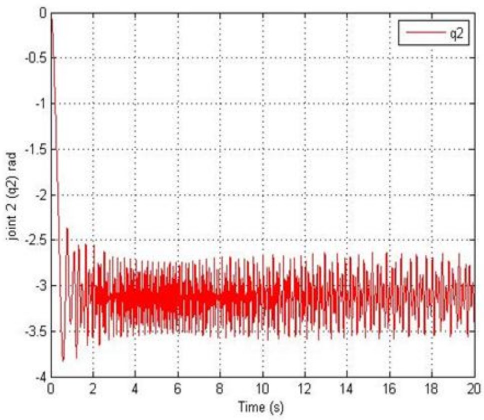

Figure 4. Joint 2 response in open loop

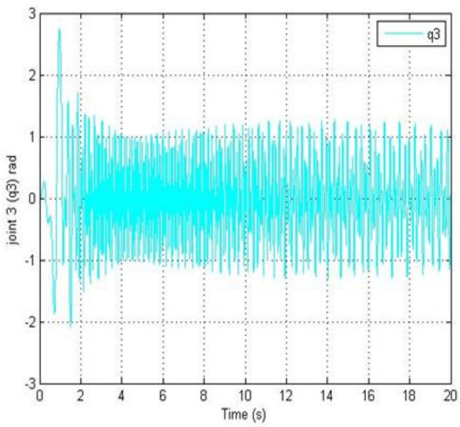

Figure 5. Joint 3 response in open loop

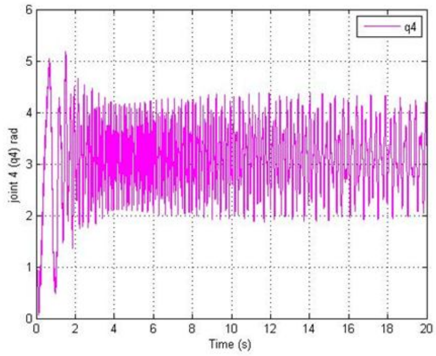

Figure 6. Joint 4 response in open loop

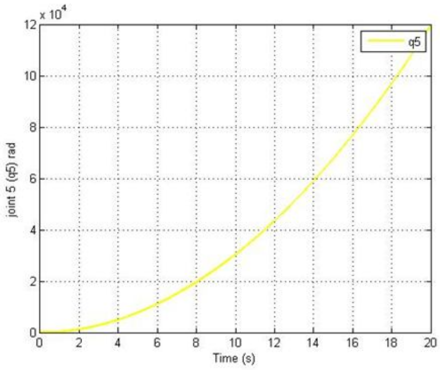

Figure 7. Joint 5 response in open loop

Form figures above the simulation when apply torques to the joints, all joints didn't give the desired position and the response for each joints is unstable. The conclusion corresponds to the simulations, the behavior of the system is unstable, and just a slightest disturbance in the system leads to a completely uncontrollable motion because the gravity on the links is dependent on the joint variables, and the input is computed without observing the output. The system requires feedback controllers to be stabilized.

5. Controller Design and Simulation

A system can be controlled in open loop or closed loop. With an open-loop controller, the input is computed without observing the output that it is controlling. Complex systems will not be possible to control in open loop, because the controller will never know if the output has achieved the desired goal. However, by adding feedback controllers, it might be possible to stabilize the system in closed loop [3].

One of the most important challenges in the field of robotics is robot manipulators control with acceptable performance, because these systems are multi-input multi-output (MIMO), nonlinear and uncertainty [6].

In this research the controller designed to control each joint of the robot manipulator independently.

5.1. PD Controller

The main objective of the controller is to develop link position tracking controllers for the robot manipulator model given by (10). To quantify the performance of the control objective, the link position tracking error e(t) is defined as:

$$e = q_d - q \tag{13}$$

Where $q_d(t)$ denotes the desired link position trajectory.

It is a remarkable fact that the simple PD scheme for set-point control can be shown to work in the general case of a system model in the form of (10). This can be proved by using Lyapunov stability analysis [3]. The proof is based on independent joint control, which means that each joint is controlled as a single-input/single-output (SISO) system. Adding PD controllers in the model, the input torque u can be written in vector form as

$$u = -K_p(q_{ref} - q) - K_d\dot{q} \tag{14}$$

K_p and K_d are positive definite diagonal matrices of proportional and derivative gains, see Figure 8.

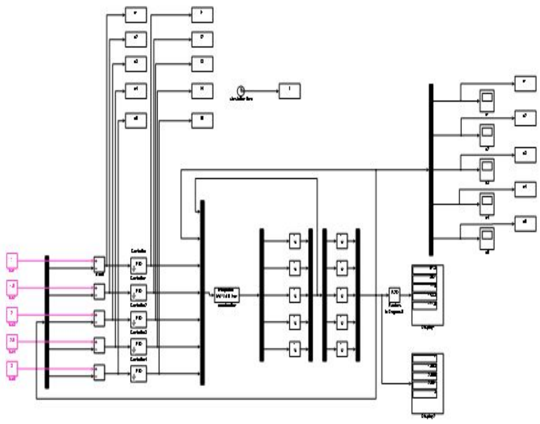

Figure 8. Close loop PD controller

In this section the results of the PD controller recorded when the desired joint angles are set to [1 1.5 2 2.5 3] respectively as shown in Figures 8-13.

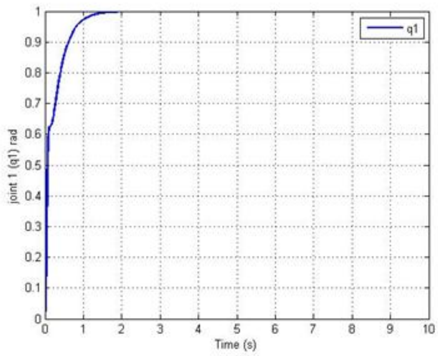

Figure 9. Joint 1 response in close loop model PD controller

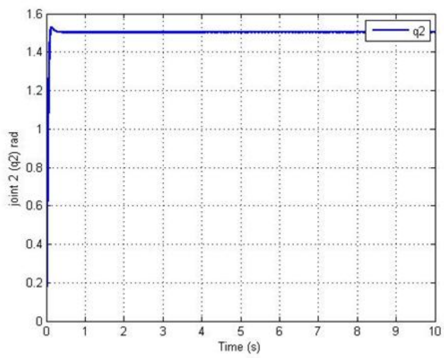

Figure 10. Joint 2 response in close loop model PD controller

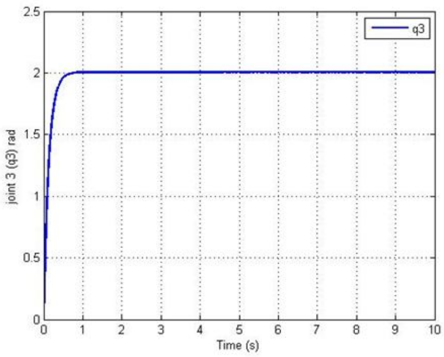

Figure 11. Joint 3 response in close loop model PD controller

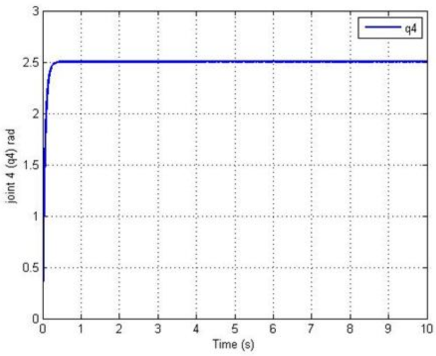

Figure 12. Joint 4 response in close loop model PD controller

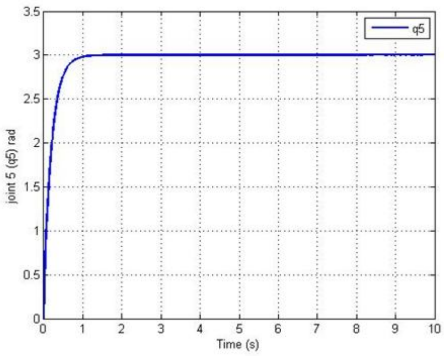

Figure 13. Joint 5 response in close loop model PD controller

5.2. Fuzzy controller

Fuzzy systems have been used in a wide variety of applications in engineering, science, business, medicine, psychology, and other fields. The fuzzy controller has four main components:

The "rule-base" holds the knowledge, in the form of a set of rules, of how best to control the system, the inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be, the fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base, and the defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant [7], see Figure 14.

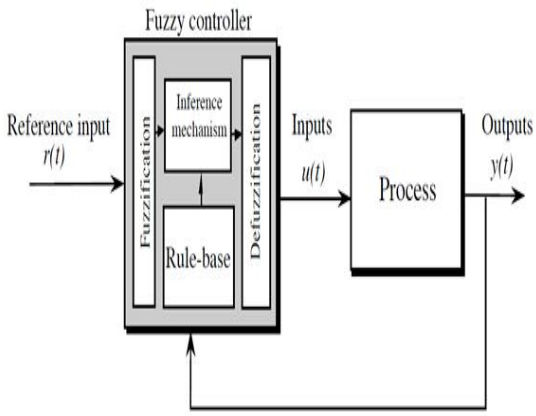

Figure 14. Fuzzy controller architecture. [7]

Fuzzy controller used to control the robot joint's angle independently by implementing the fuzzy controller as shown in Figure 15.

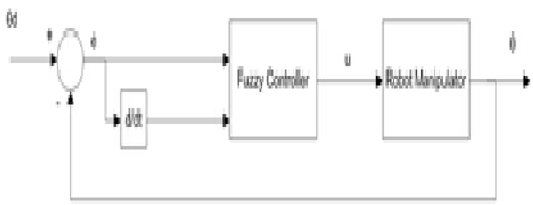

Figure 15. . Fuzzy controller for robot manipulator

The controller was designed using MATLAB fuzzy toolbox, so the membership functions were selected as a triangular shape for inputs and output as shown in Figures (16, 17, 18) the e universe of discourse chosen to be [-pi pi], the universe of discourse for de [-2.5 2.5]. These are chosen by some experiments and the universe of discourse for the output is chosen to be [-20 20] because this is the limit of the motors.

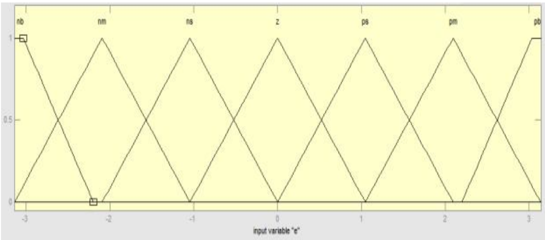

Figure 16. Membership function for e

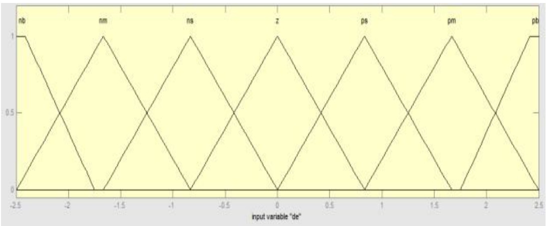

Figure 17. Membership function for de

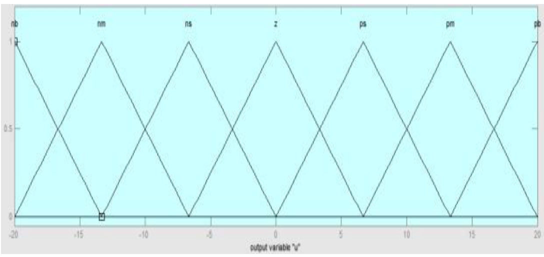

Figure 18. Membership function for u

Where

- Nb: negative big
- Nm: negative medium
- Ns: negative small
- Z: zero
- Ps: positive small
- Pm: positive medium
- Pb: positive big

These are the linguistic variables that describe each of the time varying fuzzy controller inputs and outputs are used to define the rule base of the fuzzy controller.

Table 2. Rule base table

e \ de	Nb	Nm	Ns	Z	Ps	Pm	Pb
Nb	Nb	Nb	Nb	Nm	Nm	Ns	Z
Nm	Nb	Nb	Nm	Nm	Ns	Z	Ps
Ns	Nb	Nm	Nm	Ns	Z	Ps	Pm
Z	Nm	Ns	Z	Z	Z	Ps	Pm
Ps	Nm	Ns	Z	Ps	Pm	Pm	Pb
Pm	Ns	Z	Ps	Pm	Pm	Pb	Pb
Pb	Z	Ps	Pm	Pm	Pb	Pb	Pb

The rules for the fuzzy controller are:

1- If e is Nb and de is Nb then u is Nb
2- If e is Nb and de is Nm then u is Nb
3- If e is Nb and de is Ns then u is Nb
4- If e is Nb and de is Z then u is Nm

And so on, as seen in Table 2.

The fuzzy controller Simulink is shown in Figure 19.

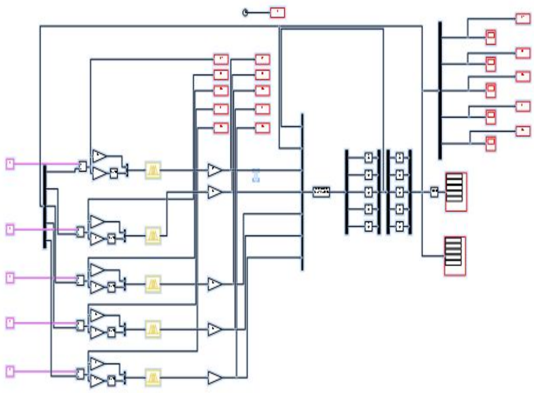

Figure 19. Close loop fuzzy controller

In this section the results of the fuzzy controller recorded when the desired joint angles are set to [1 1.5 2 2.5 3] respectively as shown Figures 20 - 24.

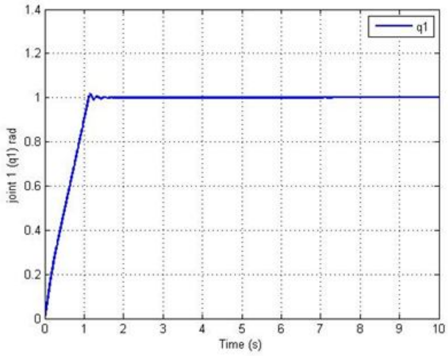

Figure 20. Joint 1 response in close loop model fuzzy controller

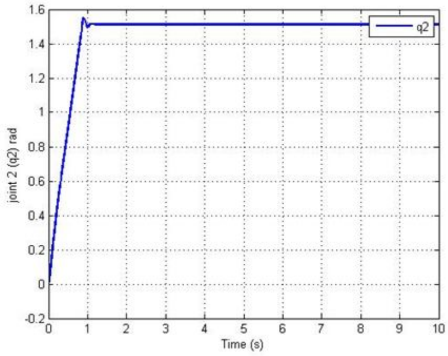

Figure 21. Joint 2 response in close loop model fuzzy controller

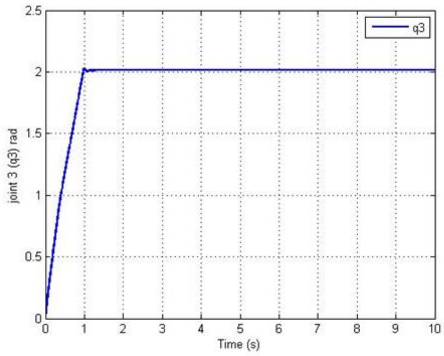

Figure 22. Joint 3 response in close loop model fuzzy controller

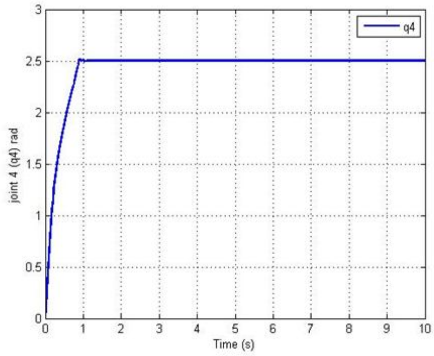

Figure 23. Joint 4 response in close loop model fuzzy controller

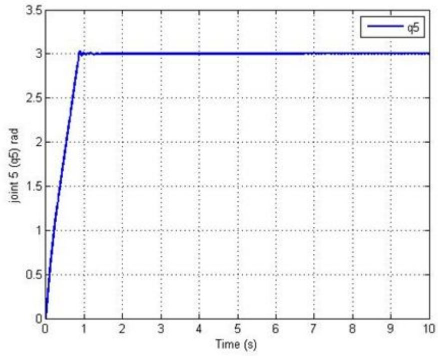

Figure 24. Joint 5 response in close loop model fuzzy controller

6. Discussion and Conclusion

As seen from the simulation result for the PD controller the signals reach the desired position (steady state time) with maximum time 1.15 second and the maximum percent over shoot is 2.044% for joint two, while in fuzzy controller the signals reaches the desired position (steady state time) with a maximum time 1.15 second and the maximum percent over shoot is 3.088% for joint two as shown in Tables 3 and 4.

Table 3. Performance specification for PD controller

Joint No.	Delay time (s)	Rise time (s)	Peak time (s)	Steady state time (ts.s) (s)	Maximum percent overshoot
Joint 1	0.0712	2	2	2.4	0
Joint 2	0.0398	0.1	0.133	0.2	2.044
Joint 3	0.093	0.75	2.006	0.6	0.314
Joint 4	0.0453	0.35	0.568	0.35	0.399
Joint 5	0.1309	1.4	2	0.8	0

Table 4. Performance specification for fuzzy controller

Joint No.	Delay time (s)	Rise time (s)	Peak time (s)	Steady state time (ts.s) (s)	Maximum percent overshoot
Joint 1	0.515	1.119	1.15	1.116	1.652
Joint 2	0.404	0.859	0.9	0.95	3.088
Joint 3	0.399	0.971	0.995	1	1.434
Joint 4	0.2266	0.891	0.908	0.9	0.754
Joint 5	0.371	0.869	0.888	0.88	1.117

Tables 3 and 4 show the characteristics of applying PD and fuzzy controllers respectively there is a little difference between them, so it doesn't matter which controller can be chosen from this side but when looking on the figures below that shows the applied torque for each joint in both controller, see that the fuzzy controller approaches the maximum motor delivered torque 20 N/m while PD controller exceed this range so the simulation can be done by putting a limiter on the control input signal to be insured that the control signal will be in the motor range but this effect on the performance of the controller when reaching the desired target so fuzzy controller will be the appropriate solution for this problem, see Figures 25 - 34.

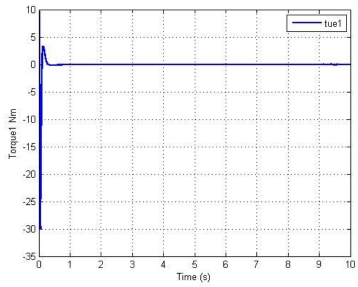

Figure 25. Joint 1 control signal for PD controller

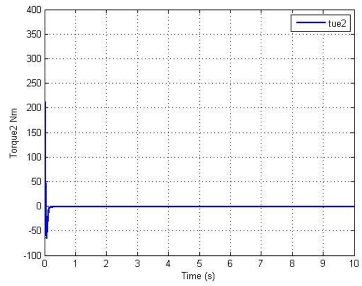

Figure 26. Joint 2 control signal for PD controller

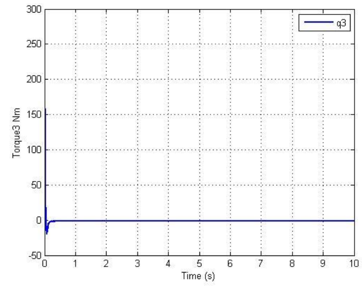

Figure 27. Joint 3 control signal for PD controller

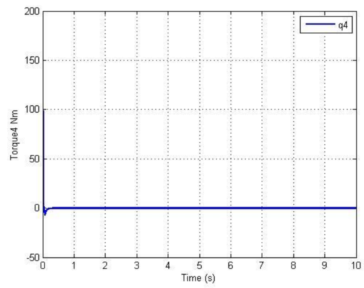

Figure 28. Joint 4 control signal for PD controller

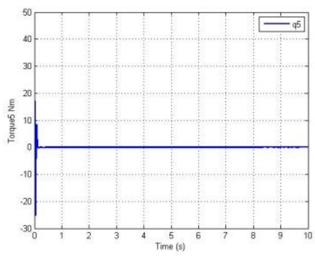

Figure 29. Joint 5 control signal for PD controller

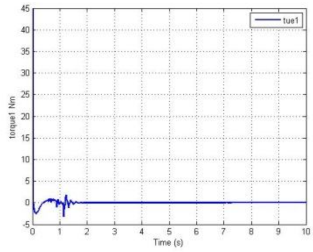

Figure 30. Joint 1 control signal for fuzzy controller

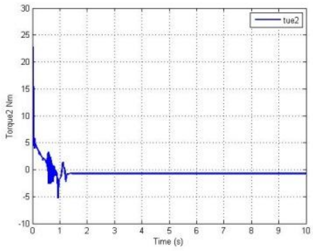

Figure 31. Joint 2 control signal for fuzzy controller

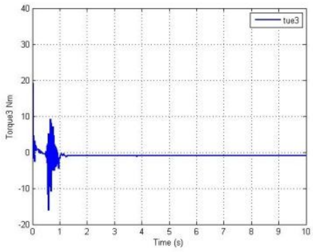

Figure 32. Joint 3 control signal for fuzzy controller

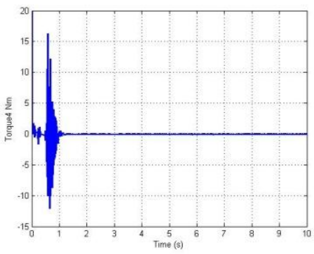

Figure 33. Joint 4 control signal for fuzzy controller

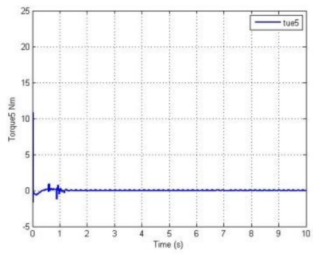

Figure 34. Joint 5 control signal for fuzzy controller

Appendix (A)

Forward Kinematics

The forward kinematics is a set of equations that calculates the position and orientation of the end- effecter in terms of given joint angles. This set of equations is generated by using the Denavit-Harbenterg (D-H) parameters obtained from the frame assignation. The parameters for the Lab-Volt 5250 arm are listed in Table A-1 which derived from Figure A-1, where θ_i represents rotation about the Z-axis, α_i rotation about the X-axis, d_i translation along the Z-axis, and a_i translation along the X-axis.

Table A-1. The D-H parameters of the Lab-Volt 5250 arm

Frame	a_i (mm)	α_i (degree)	d_i (mm)	θ_i
1	0	90	380	θ_1
2	380	0	0	θ_2
3	230	0	0	θ_3
4	0	90	0	θ_4
5	0	0	150	θ_5

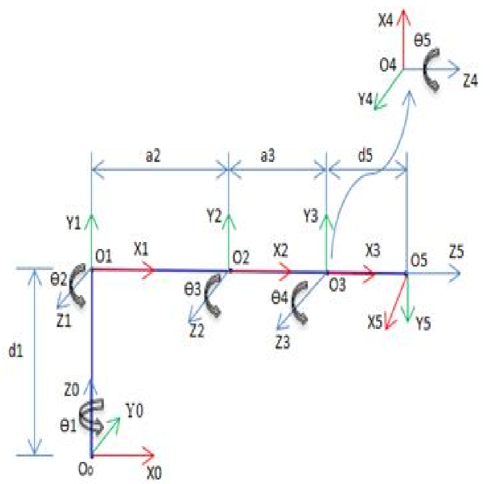

Figure 35 Coordinate Frames of 5250 Lab-Volt Robotic Arm

The forward kinematics describes the transformation from one frame to another, starting at the base and ending at the end-effecter. The transformation matrix Ai between two neighboring frames o_{i-1} and o_i is expressed as:

$$A_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A1)$$

$$A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A2)$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A3)$$

$$A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A4)$$

$$A_4 = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A5)$$

$$A_5 = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A6)$$

The translational kinetic energy KEt is 5×5 matrix, and its terms as shown below:

$$T_{11} = (m_3(a_{c3}{}^2c_{23}{}^2 + a_2{}^2c_2{}^2 + 2a_2a_{c3}c_{23}c_2))/2 + (m_4(a_2{}^2c_2{}^2 + a_{c3}{}^2c_{23}{}^2 + 2a_2a_3c_{23}c_2))/2 + (a_{c2}{}^2m_2c_2{}^2/2 + (a_2{}^2m_5(a_3{}^2c_{23}{}^2 + c_2{}^2 + d_{c5}^2s_{234}{}^2 + 2a_3d_{c5}s_{234}c_{23} + 2a_2d_{c5}s_{234}c_2 + 2a_2a_3c_{23}c_2))/2$$

$T_{12} = 0$

$T_{13} = 0$

$T_{14} = 0$

$T_{15} = 0$

$T_{21} = 0$

$$T_{22} = (m_2a_{c2}{}^2/2 + (m_5(a_2{}^2 + 2c_3a_2a_3 + 2s_{23}a_2d_{c5} + a_3{}^2 + 2s_{34}a_3d_{c5} + d_{c5}^2))/2 + (m_4(a_2{}^2 + 2c_3a_2a_3 + a_3{}^2))/2 + (m_3(a_2{}^2 + 2c_3a_2a_{c3} + a_3{}^2))/2$$

$$T_{23} = (m_5(a_3{}^2 + 2s_4a_3d_{c5} + a_2c_3a_3 + d_{c5}^2 + a_2s_{34}d_{c5}))/2 + (m_4(a_3{}^2 + a_2c_3a_3))/2 + (m_3(a_{c3}{}^2 + a_2c_3a_{c3}))/2$$

$T_{24} = (d_{c5}m_5(d_{c5} + a_2s_{34} + a_3s_4))/2$

$T_{25} = 0$

$T_{31} = 0$

$$T_{32} = (m_5(a_3{}^2 + 2s_4a_3d_{c5} + a_2c_3a_3 + d_{c5}^2 + a_2s_{34}d_{c5}))/2 + (m_4(a_3{}^2 + a_2c_3a_3))/2 + (m_3(a_{c3}{}^2 + a_2c_3a_{c3}))/2$$

$$T_{33} = (a_3{}^2m_4)/2 + (a_{c3}{}^2m_3)/2 + (m_5(a_3{}^2 + 2s_4a_3d_{c5} + d_{c5}^2))/2$$

$T_{34} = (d_{c5}m_5(d_{c5} + 2a_3s_4))/2$

$T_{35} = 0$

$T_{41} = 0$

$T_{42} = (d_{c5}m_5(d_{c5} + a_2s_{34} + a3s_4))/2$

$T_{43} = (d_{c5}m_5(d_{c5} + a_3s_4))/2$

$T_{44} = (d_{c5}^2m_5)/2$

$T_{45} = 0$, $T_{51} = 0$, $T_{52} = 0$, $T_{53} = 0$

$T_{54} = 0$, $T_{55} = 0$

The rotational kinetic energy KEr is 5×5 matrix, and its terms as shown below:

$$R_{11} = \frac{I_1}{2} + \frac{I_2}{2} + \frac{I_3}{2} + \frac{I_4}{2} + \frac{I_5}{2}$$

$R_{12} = R_{21} = 0$

$R_{13} = R_{31} = 0$

$R_{14} = R_{41} = 0$

$$R_{15} = R_{51} = \frac{-I_5c_{234}}{2}$$

$$R_{22} = \frac{I_2}{2} + \frac{I_3}{2} + \frac{I_4}{2} + \frac{I_5}{2}$$

$$R_{23} = R_{32} = \frac{I_3}{2} + \frac{I_4}{2} + \frac{I_5}{2}$$

$$R_{24} = R_{42} = \frac{I_4}{2} + \frac{I_5}{2}$$

$R_{25} = R_{52} = 0$

$$R_{33} = \frac{I_3}{2} + \frac{I_4}{2} + \frac{I_5}{2}$$

$$R_{34} = R_{43} = \frac{I_4}{2} + \frac{I_5}{2}$$

$R_{35} = R_{53} = 0$

$$R_{44} = \frac{I_4}{2} + \frac{I_5}{2}$$

$R_{45} = R_{54} = 0$

$$R_{55} = \frac{I_5}{2}$$

References

[1] Brian Armstrong, Oussama Khatib and Joel Burdick, “The Explicit Dynamic Model and Inertial Parameters of The PUMA 560 Arm,”IEEE International Conference on Robotics and Automation,Vol. 3, pp 510-518, 1986

[2] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, “RobotModeling and Control”, 1st Edition, John Wiley & Sons.2005.

[3] H. Høifødt, “Dynamic Modeling and Simulation of RobotManipulators”, M.Sc. thesis, Norwegian University of Scienceand Technology, 2011.

[4] Joseph E. Shigley, “Mechanical Engineering,” Eighth Edition, McGraw−Hill, 2006.

[5] J. L. Meriam, L. G. Kraige, “Engineering Mechanics, volume 1Statics”, 5th edition, John Wiley & Sons, 2002.

[6] Farzin Piltan et al, “Artificial Control of PUMA Robot Manipulator:A-Review of Fuzzy Inference Engine and Application toClassical Controller,” International Journal of Robotic andAutomation (IJRA), Vol.2, Issue 5, 2011.

[7] Kevin M. Passino, Stephen Yurkovich, “Fuzzy Control” Addison-Wesley Longman, 1998