

## HUMAN ARM SIMULATION BASED ON MATLAB WITH VIRTUAL ENVIRONMENT<sup>1</sup>

Prof. Dr. Mohammed Z. Al-Faiz, MIEEE<sup>2</sup>

Prof. Dr. Abduladhem A. Ali<sup>3</sup>

Assist. Lec. Abbas H.Miry<sup>4</sup>

### Abstract

This work presents a novel simulation methodology applied to a human arm. It is aimed to allow the robotic system to perform complex movement operations of human arm. The human arm is represented by using virtual reality (VR). The work includes mathematical modeling of the direct kinematics, inverse kinematic and the dynamics of the human arm. The model permits direct forward dynamics simulation, which accurately predicts hand position, also presents a solution to the inverse problem of determining set of joints angle to achieve a given position or motion. The method is implemented in the 3D space and uses the Simulink/ MATLAB Ver.2009a approach. This methodology can be used with different robots to test the behavior and control laws.

**Keywords:** *Human Arm, Inverse Kinematic, Levenbrge marquite. Virtual Reality.*

محاكاة ذراع الانسان باستعمال ال *MATLAB* والبيئة الافتراضية

### الخلاصة

هذا العمل يقدم محاكاة لذراع إنساني للسماح للنظام الآلي لأداء عمليات الحركة المعقدة التي تقوم بهذراع الإنسان. الذراع الإنساني مثلت باستعمال الحقيقة المظهرية، يتضمن العمل التمثيل الرياضي للنظام الحركي المباشر و العكسي والديناميكي. يستعرض البحث موقع ذراع الانسان بدقة ويقدم أيضاً حلاً إلى المشكلة المعكوسة لإيجاد زوايا المفاصل لذراع الإنسان. تُطبّق الطريقة في فضاء ثلاثي الأبعاد وتُستعملُ *Simulink / Matlab Ver 2009 a*. هذا المنهج يُمكن أن يُستعمل مع الرجال الآليين المختلفين لاختبار السلوك.

<sup>1</sup> This paper was presented in the Engineering Conference of Control, Computers and Mechatronics Jan. 30-31/2011, University of Technology.

<sup>2</sup> Computer Engineering Dept. Nahrain University.

<sup>3</sup> Computer Engineering Dept. Basrah University.

<sup>4</sup> Electrical Engineering Dept. AL-Mustansiriyah University.

## **1. Introduction**

Robotics is one of the main disciplines in industry used in the development of new technologies. The synergy of robotics with different applications like submarine task, car assembly operation, vision systems and artificial intelligence allow innovation and reduces the manufacturing costs. For this purpose, it is important that the robot programmers are able to visualize and test the behavior of the robots in different circumstances and with different parameters [1].

Motion is one of the most natural and important activities that a human being can practice, even before birth. Thus, one of the major difficulties by those who are fitted with prosthetic devices is the great mental effort needed during the first training stages. This makes it very difficult to precisely detect the spatial position as well as the forces done by the prosthesis in replacement of the muscle. This problem is still more complicated when working with upper limb prosthesis, whose function is to simulate its biological equivalents (shoulder, arm, forearm, elbow, and wrist). All these mechanisms present a very efficient system with a lot of information (degrees of freedom, speed, angles etc) to be simulated and absorbed by brain commands to help professionals dedicated to special physical needs patient rehabilitation [2].

Some other works only make a 3D virtual that deal with the use of Matlab for systems simulations, M.Z. Al-Faiz [3] presents architecture for posture learning of an anthropomorphic robotic arm using matlab/simulink with virtual reality. The approach was aimed to allow the robotic system to perform complex movement operations of human arm; in this paper only forward kinematic is used in the simulation. B.Lee et al [4] presents the simulation of humanoid walking pattern using 3D simulator with Virtual Reality Toolbox. By using the Virtual Reality Toolbox

incorporated with MATLAB, The simulator was composed of three modules, namely, waking pattern code, kinematics code and display code. V.Sanchez et al [1] simulated the methodology of the 5-DOF including mathematical modeling of the direct, inverse and differential kinematics as well as the dynamics of the manipulator. This method was applied to test the robot CATALYST 5 by using a project in Simulink and Matlab. The method implements the path following in the 3D space and uses the Matlab-Simulink approach.

The main objective of the present work is to show a complete simulation of human arm, where, the combination of Matlab/Simulink and virtual reality is proposed. The method allows to manipulate the human arm system and to visualize the robot's behavior from different perspectives.

## **2. Human Arm Modeling**

The modeling of a manipulator robot with n degree of freedom (DOF.) can be divided in three steps: Direct Kinematics, Inverse Kinematics, and Dynamics.

### **2.1. Kinematics of Human Arm Based Levenberg-Marquardt Algorithm**

Kinematics is the study of motion without regard to the forces that create it. The representation of the robot's end-effector position and orientation through the geometries of robots (joint and link parameters) are called forward Kinematics. The forward kinematics is a set of equations that calculates the position and orientation of the end-effectors in terms of given joint angles. This set of equations is generated by using the D-H parameters obtained from the frame assignment [5]. The inverse kinematics problem (IKP) for a robotic manipulator involves obtaining the required manipulator joint values for a given desired end-point position and orientation. It is usually complex due to the lack of a unique solution

and closed-form direct expression for the inverse kinematics mapping [6].

**2.2 Structure and Kinematics of Human Arm**

The development of a high-DOF kinematic discusses human arm model that can be used to predict realistic human arm postures. One may deal with human arm by 7-DOF and assume the origin at shoulder joint. The first joint is the shoulder joint s with 3 DOFs.

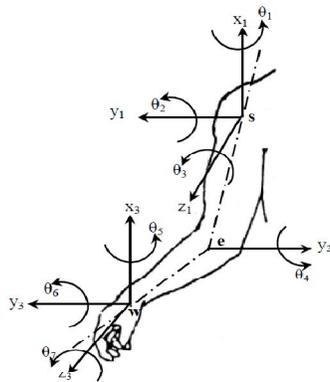


Figure.1. Kinematic chain of human arm

The elbow joint e has only one DOF. The wrist joint w is of the same type as the shoulder joint s and also has 3 DOFs. Note that the arrow at the end of the chain indicates the end effectors orientation and is not another link. It can be focused on a kinematic chain that is formed after a human arm. This means that the kinematic chain has 3 joints with spherical joints as shoulder and wrist joint and a hinge joint as the elbow joint. The spherical joints have 3 DOFS while the hinge joint has only one DOF, giving a total of 7 DOFs for this kinematic chain (see Fig.1.) The homogeneous transformation matrices for the frame transitions are set up with D-H parameters.

Table (1) D-H Parameters of human arm

Frame (joint)	qi (rad)	di (cm)	ai (cm)	αi (rad)
1	q1	0	0	π/2
2	q2	0	0	π/2
3	q3	0	20	-π/2
4	q4	0	25	π/2
5	q5	0	0	-π/2
6	q6	0	0	π/2
7	q7	0	0	-π/2

It has following transformation matrices

$$A1 = \begin{bmatrix} c1 & -s1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A2 = \begin{bmatrix} c2 & -s2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s2 & c2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A3 = \begin{bmatrix} c3 & -s3 & 0 & a3 \\ 0 & 0 & 1 & 0 \\ s3 & -c3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A4 = \begin{bmatrix} c4 & -s4 & 0 & a4 \\ 0 & 0 & -1 & 0 \\ s4 & c4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A5 = \begin{bmatrix} c5 & -s5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s5 & -c5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A6 = \begin{bmatrix} c6 & -s6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s6 & c6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_7 & -c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where 'c' is cosine of theta and 's' is sin of theta, the forward kinematic is represented by T

$$T = A_1 * A_2 * A_3 * A_4 * A_5 * A_6 * A_7 = \begin{bmatrix} nx & ox & ax & dx \\ ny & oy & ay & dy \\ nz & oz & az & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

### **2.3 Dynamic of Human Arm Based Genetic Algorithm**

Decentralized control has been widely accepted by the robotics industry due to ease of implementation and tolerance to failure. Conventional controllers for industrial robots are based on independent joint control schemes in which each joint is controlled separately by a simple position servo loop with predefined constant gains [7]. In this paper, the proposed scheme is based on the principle of treating each joint as a subsystem in which the joint dynamics are modeled by a third order system plus a disturbance torque. By a simple disturbance cancellation procedure, the disturbance in the joint dynamic equation is directly rejected so that simple fixed linear joint controllers can be readily designed. Fig. 1 shows the structure of Human arm which has 7 Degree Of Freedom(DOF), this mean seven motor are used to represent movement of human arm with independent joint control. The equation describing the dynamic behavior of the DC motor is given by the following equations:

$$G(s) = \frac{K_b}{[JLa s^3 + (Ra J + BLa) s^2 + (K_b K_T + Ra B) s]} \quad (2)$$

As PID controller has simple structure, easy to understand, the tuning technique

provides adequate performance in the vast majority of applications, the PID controller is widely used in most industrial processes. Despite continual advances in control theory it cannot effectively control complicated or fast system such as motor controlled systems, however, because the response of a plant depends on three parameters (P, I, and D) and gain must be manually tuned by trial and error. Most PID tuning rules use conventional methods. This requires considerable technical experience to apply tuning formulas to determine PID controller parameters.

GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection. Recently, GA has been recognized as an effective and efficient technique to solve optimization problems. As a mathematical means for optimization, GAs can naturally be applied to the optimal-tuning of PID controllers. With reference to a step input signal, the entire system will generate an output step response. The role of the PID controller is to drive this output response within the user's specifications. Obviously, the parameter settings of the PID controller should be finely tuned so as to meet as high requirements as possible. In this paper, genetic algorithm is operated with the following: number of population = 20, probability of crossover is 0.8, probability of mutation is 0.01.

The most crucial step in applying GA is to choose the objective functions that are used to evaluate the fitness of each chromosome. Some works use performance indices as the objective functions. The objective functions are Integral of Time multiplied by Squared Error (ITSE), Integral of Time multiplied by Absolute Error (ITAE), Integral of Absolute Magnitude of the Error (IAE), and Integral of the Squared Error (ISE). The integral term become summation in the discrete time.

The PID controller is used to minimize the error signals, or we can define more rigorously, in the term of error criteria: to

minimize the value of performance indices mentioned above. And because the smaller the value of performance indices of the corresponding chromosomes is the fitter the chromosomes will be, and vice versa, we define the fitness of the chromosomes as

$$fitness\ value = \frac{1}{Objective\ function} \quad (3)$$

In this paper a multi objective is introduced to get better time response by replacing fitness value in Eq (4) by

$$= \frac{1}{w*t_s + w*ov + Objective\ function} \quad (4)$$

Where

Objective function is taken from Eq (3).

$w \gg 1$  since the value of performance index is very large with respect to the settling time and over shot.

### **3. Inverse Kinematics Solution Using Optimization Method**

The problem of inverse kinematics is not linear, as rotations are involved. This means that analytical solutions are only available in limited situations. In all other cases, alternative methods will have to be employed. The most-used alternative is numerical solutions. Analytical solutions are the best option to use when available, as they are the fastest and most reliable inverse kinematics solvers. The problem of analytically solving inverse kinematics is that it does not scale to more complex bone sets and is therefore only an option for simple situations, like robot arms with few joints, or a single human leg. The numerical solution that can be utilized to solve the inverse kinematics problem is through the use of existing optimization algorithms. For optimization of a reference joint angle configuration regarding the similarity measure, one can use the Levenberg-Marquardt algorithm. The algorithm provides a standard technique for solving nonlinear least squares problems by iteratively converging to a minimum of function expressed as sum of squares. Combining the

Gauss-Newton and the steepest descent method, the algorithm unites the advantages of both methods. Hence, using the LM method, a more robust convergence behavior is achieved at points far from a local minimum, while a faster convergence is gained close at a minimum. Due to its numerical stability, the LM method has also become a popular tool for solving inverse kinematics problems as demonstrated in [8]. The LM algorithm is an iterative technique that locates a local minimum of a multivariate function that is expressed as the sum of squares of several non-linear, real-valued functions. It has become a standard technique for nonlinear least-squares problems, widely adopted in various disciplines for dealing with data-fitting applications. LM can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from a local minimum, the algorithm behaves like a steepest descent method: slow, but guaranteed to converge.

$$E(x) = \frac{1}{2} \sum e^T(x)e(x) \quad (5)$$

The Gauss-Newton method can be written as

$$x^{k+1} = x^k - [J(x^k).J(x^k)]^{-1}.J(x^k).e(x^k) \quad (6)$$

LM method can be expressed with the scaling factor

$$x^{k+1} = x^k - [J(x^k).J(x^k)\lambda.I]^{-1}.J(x^k).e(x^k) \quad (7)$$

Where  $I$  is the identity matrix[9].

It required minimizing the error between the target transformation matrix and calculated transformation matrix. The problem can be formularizing as optimization problem as following:

$$\text{Minimize } e(q) = \begin{bmatrix} |P_x^d - dx| \\ |P_y^d - dy| \\ |P_z^d - dz| \end{bmatrix} \quad (8)$$

Subject to

$$q_i^l \leq q_i \leq q_i^u \quad i=1 \dots 7$$

$$q^l = \left[ -\frac{\pi}{2} \quad -\frac{11\pi}{2} \quad -\frac{\pi}{2} \quad \frac{\pi}{12} \quad -\frac{\pi}{3} \quad -\frac{\pi}{9} \quad -\pi \right]$$

$$, q^u = \left[ \frac{\pi}{2} \quad \frac{2\pi}{3} \quad \frac{\pi}{2} \quad \frac{5\pi}{6} \quad \frac{\pi}{3} \quad \frac{\pi}{9} \quad 0 \right]$$

Where  $p^d$  Target position in the space. These constrain become

$$q_i^l - q_i \leq 0 \quad , q_i - q_i^u \leq 0 \quad i=1 \dots 7$$

By convert this constrained problem to unconstrained problem using penalty function[10], the problem become

$$e(q) = \begin{bmatrix} |P_x^d - dx| \\ |P_y^d - dy| \\ |P_z^d - dz| \\ G_{1i} \\ G_{2i} \end{bmatrix} \quad (9)$$

$$G_{1i} = \max (0, (q_i^l - q_i)),$$

$$, G_{2i} = \max (0, (q_i - q_i^u)) \quad i=1 \dots 7,$$

Eq (9) becomes the unconstrained objective function to be minimized.

#### 4. Virtual Reality

With the advent of high-resolution graphics, high-speed computing, and user interaction devices, virtual reality (VR) has emerged as a major new technology in recent years. An important new concept introduced by many VR systems is immersion, which refers to the feeling of complete immersion in a three-dimensional computer-generated environment by means of user-centered perspective achieved through tracking the user. This is a huge step forward compared to classical modeling and CAD/CAM packages, VR technology is currently used in a broad

range of applications, the best known being flight simulators, walkthroughs, video games, and medicine (virtual surgery). From a manufacturing standpoint, some of the attractive applications include training, collaborative product and process design, facility monitoring, and management. Moreover, recent advances in broadband networks are also opening up new applications for virtual environments in these areas [11].

#### 5. Proposed Model and VR Simulation for a Human Arm

The simulator was built using MATLAB with Virtual Reality Toolbox. MATLAB provides powerful engineering tool including frequently used mathematical functions. It is easy to implement control algorithm including visualization of data used in the algorithm. In addition, by using Virtual Reality Toolbox, it is convenient to treat 3D objects defined with VRML (Virtual Reality Modeling Language). Thus, it is possible to build a simulator within a relatively short period. Virtual Reality (VR) is a system which allows one or more users to move and react in a computer generated environment. The basic VR systems allow the user to visual information using computer screens. The simulation contains two parts ,first, building model for human arm in VRML,second, constitution the simulink a model in MATLAB then calling and running the model of human arm using virtual reality toolbox.

##### 5.1. The VRML Model for Human Arm

To realize the VRML model for Human arm save the file as All.wrl file, which is the file format for Virtual Reality software. The VRML model of the Human arm was designes in V-Realm Builder 2.0 . Fig.2 presented the VRML model of the Human arm. In V-Realm Builder 2.0 program the viewpoints were defined, the background properties, the properties for each element, its position and orientation, etc. In Fig.3 is

presented the V-Realm Builder 2.0 captured screen with all the views of the VRML model of the Human arm.

### **5.2 The Simulink Model for Human Arm**

The VRML model represents a Human arm. It is manipulated by a simple Simulink model as it can be seen in Fig.4. As it can be seen from Fig.4 the Human arm is represented by several blocks: the first one named Target position which represents the desired target ,the second block calculate the inverse kinematics of desired target to produce desired angle.The third block is dynamic which contain PID controller and fourth part is simulation part using virtual reality.

### **6. Simulation Results**

The implemented simulation of human arm which built in VR technique, the solution of inverse kinematic equations and control of joints are achieved by MATLAB Ver.R2009a. Fig.4 represents the connection between MATLAB\Simulink, which solve the kinematic equations, and VR model of human arm. The moving commands for this model are calculated in MATLAB and then they are implemented in VR. Different cases of PID control are studied and illustrated in Fig. 5,6,7,8 and 9 . From Fig.7,8,9 it can be seen the ability of performed GA based PID tuning with our fitness function ,that remove the overshoot and reduce the settling time with lower error, Finally the trajectory of human arm using the proposed fitness function has better motion . As inputs sources for the block that represent the target positioning kinematics part ,joint angle of human arm are calculated using optimization method (Levenberg-Marquardt algorithm).In dynamic part two methods of controlling are used, first using proptional gain equal to one, second using PID control tuning by genetic algorithm.

### **7. Conclusion**

This paper presents the mechanical analysis of human arm joints (shoulder, elbow and wrist).The modeling of the human arm includes direct and inverse kinematics as well as dynamics. This method was applied to test the human arm control by using a project in Simulink/Matlab with virtual reality. A proposed method was represented in the 3D space. The simulation shows the good performance of the proposed methodology. From the obtained results, it can be concluded that the Virtual Reality is useful to test the viability of designs before the implementation phase on a virtual reality prototype, the ability of performing of GA based PID tuning for human arm .The system is of high flexibility due to the easy programming language. So it is really useful for lab experiment and hardware design for human arm. It gives a clear direction to the design of human arm control system and the entire optimum design.

#### **Appendix: Dynamic of DC Motor**

The equations describing the dynamic behavior of the DC motor are given by the following equations;

$$v = Ri + L \frac{di}{dt} + e_b \quad (10)$$

$$T_m = K_T * i_a(t) \quad (11)$$

$$T_m = J \frac{d^2 \theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} \quad (12)$$

$$e_b = e_b(t) = K_b \frac{d\theta(t)}{dt} \quad (13)$$

After simplification and taking the ratio of  $\theta(s)/ v(s)$  we will get the transfer function as below,

$$G(s) = \frac{\theta(s)}{v(s)} = \frac{K_b}{[J La S^3 + (Ra J + B La) S^2 + (K_b K_T + Ra B) S]} \quad (14)$$

When an industrial robot transfers objects of different masses, the dynamics of DC motor will be changed, also, for disturbance rejection it is more efficient PID control system which it is well known as a simple and useful method to control manipulators.

Many industrial robots use a form of so called PID control law [12] as

$$\ddot{q}_d + k_p e + k_d \dot{e} + k_i \int e dt = u \quad (15)$$

Where  $u$  is the control law vector,  $e = q_d - q$  is the position error vector,  $q_d$  is the desired joint angle vector,  $q$  is the joint angle vector,  $k_p$ ,  $k_d$  and  $k_i$  are the coefficient of PID controller.

### References

- [1] Sanchez .V, Gutierrez .R, Valdovinos .G and Ortega .P, 2010 “5-DOF manipulator simulation based on MATLAB-Simulink methodology “, 20th International Conference on Electronics, Communications and Computer (CONIELECOMP, pp: 295 - 300 .
- [2] Grecu .V, Grecu .L, Demian .M and Demian .G, 2009 ,”A Virtual System for Simulation of Human Upper Limb “,Proceedings of the World Congress on Engineering London, U.K. , Vol. 2, pp: 1474-1478.
- [4] Lee .B, Kim .Y, Yoo .J, Park .I and Kim .J, 2008,”Walking Pattern Simulation based on Virtual Reality Toolbox”.,
- [3] Al-Faiz ,M.Z., 2009,” A Posture Human Robotic Arm by Inverse Kinematic With Virtual Environment”, MASAUM Journal of Basic and Applied Sciences,Vol.1, No. 4,
- [5] Xu ,D., and Acosta .A , 2005,” An Analysis of the Inverse Kinematics for a 5-DOF Manipulator” ,International Journal of Automation and Computing Vol. 2 , pp :114-124 .
- [6] Hua .C and shan .W. ,2006 , “Wavelet network solution for the inverse kinematics problem in robotic manipulator”,Chen Zhejiang Univ Science , Vol. 7 ,No. 4 PP: 525-529.
- [7] Hsia .T. C, Lasky T. A, and Guo .Y, 1988 “ Robust Independent Robot joint control: design and Simulation”, 1988 IEEE International Conference on Robotics and Automation , ,vol.3 pp:1329-1334.
- [8] Wampler. C. W, 1986”Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods”, IEEE Transactions on Systems, Man and Cybernetics, no 1 pp 93–101.
- [9] Endisch .C and Kennel .R, 2009” Levenberg-marquardt-based BS algorithm using adaptive pruning interval for system identification with dynamic neural networks”, IEEE International Conference on Systems, Man, and Cybernetics , San Antonio, TX, USA, pp 3402 - 3408.
- [10] Rao .S. S , 1996 ”Engineering Optimization Theory and Practice”,Third Edition John Wiley & Sons, Inc.
- [11] Tachi, S., 1998,” Virtual Reality in Robotics” In: Proceedings of the 8th International Symposium on Measurement and Control in Robotics ISMCR '98. Prague, pp. 29-34.
- [12] Craig .J. J,2009,” Introduction to Robotics, Mechanics and Control”, Addison-Wesley.

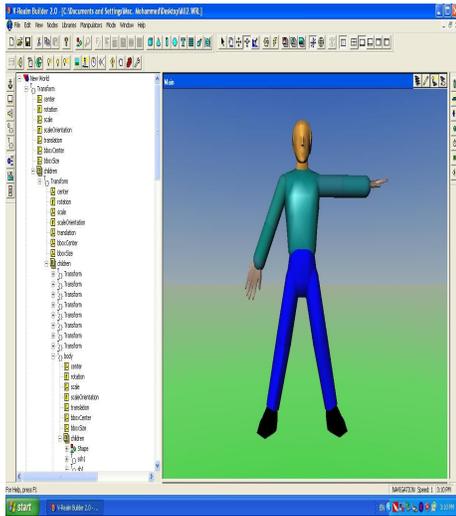


Figure. 2. The VRML model of the Human arm

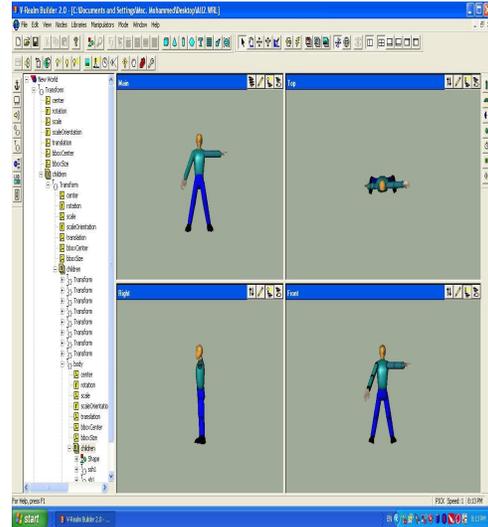


Figure . 3. The views of the VRML model of the Human arm

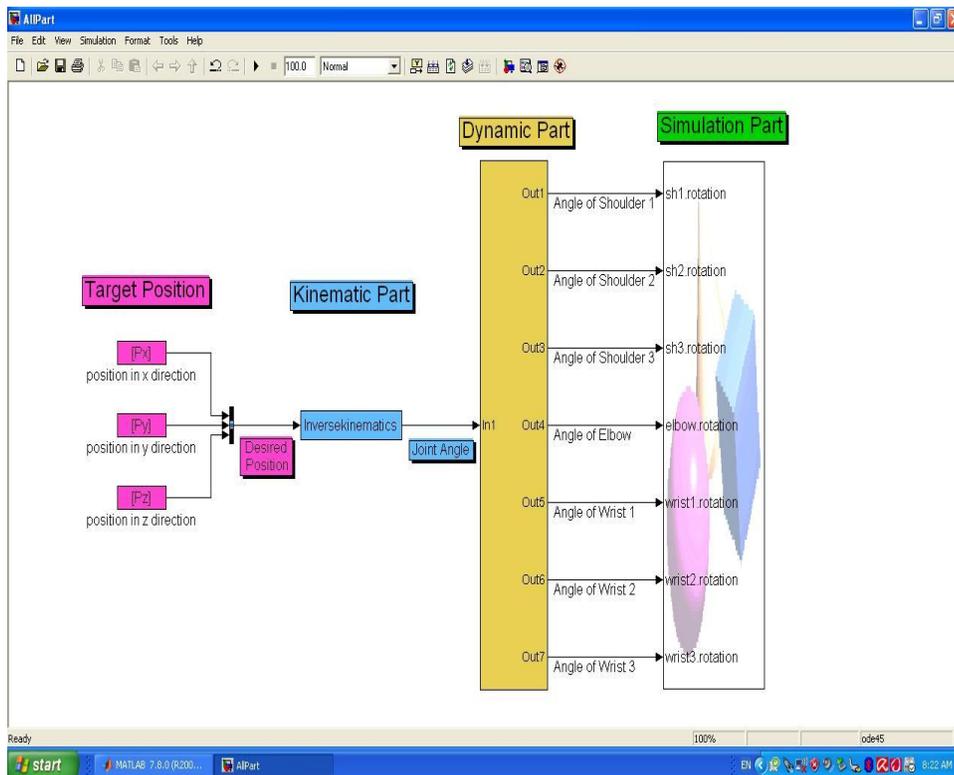


Figure . 4. The Simulink model for A Human Arm

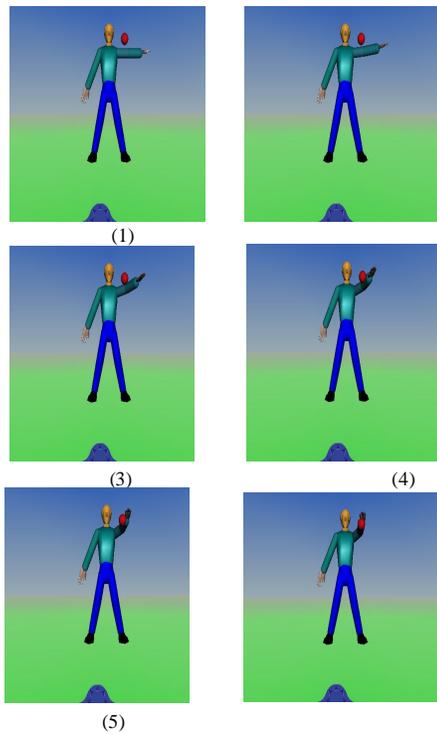


Figure. 5. Human arm movements with proposed PID

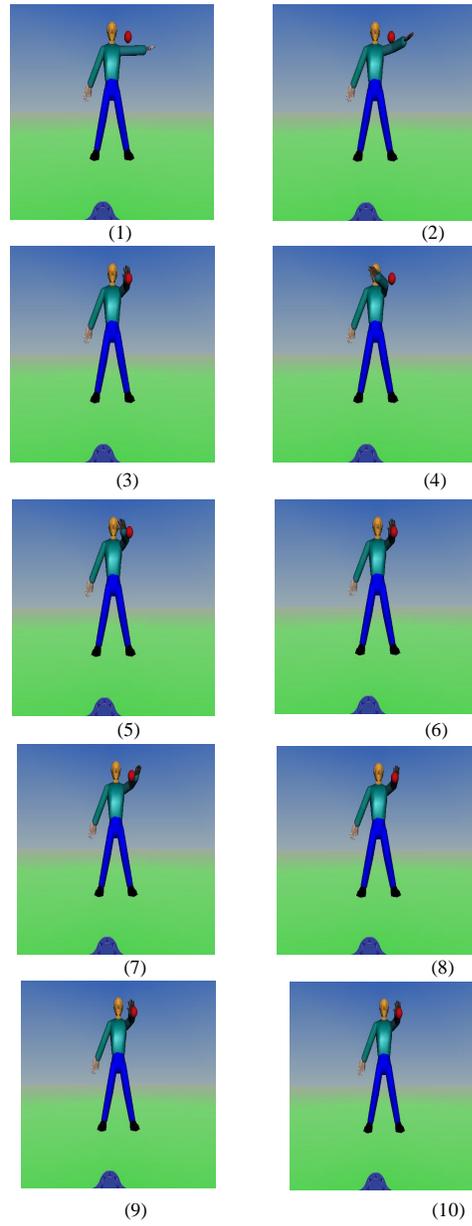


Figure. 6. Human arm movements with unity feedback

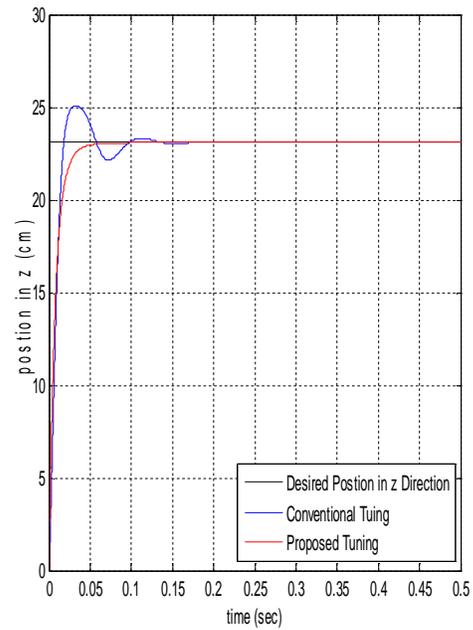
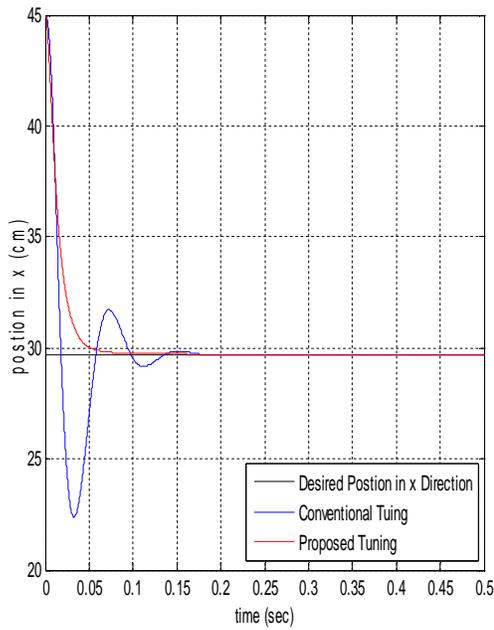


Figure.9 Position Responses in Z Direction

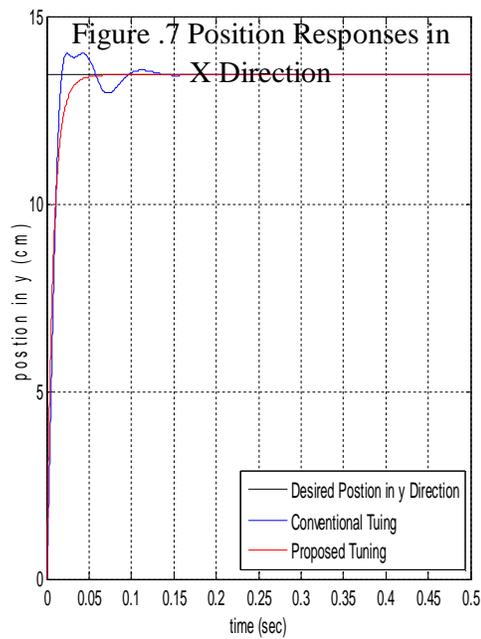


Figure .8. Position Responses in Y Direction