

Design and Position Control of Arm Manipulator; Experimentally and in MATLAB SimMechanics

Amr Nasr A. E. Gaber
Instructor

Department of Mechatronics Engineering
Alexandria Institute of
Engineering and Technology (AIET),
Alexandria –Egypt

Sohair F. Rezeka
Professor

Department of Mechanical Engineering
Arab Academy for Science Technology &
Maritime Transport
Alexandria –Egypt

H. A. El Gamal
Professor

Department of Mechanical Engineering
Alexandria University
Alexandria –Egypt

Abstract:- This work presents the design and position control of a 6-DOF robot arm. Autodesk inventor was used for the design and dynamic simulation of the arm. An interface between the Matlab SimMechanics and Autodesk Inventor is developed to animate the motion of the robot arm. The interface is able to communicate with the robotic controller to send and retrieve information to/from the controller. An experimental test rig for the arm is built. The main components of the electrical subsystem are servo motors and stepper motor that drive the links, processing and control unit, Arduino board, and two laser pins for sensing the gripper position. The motion of servo motor depends on PWM (pulse width modulation) pin which is produced by a micro- controller. A PI control that depends on the inverse kinematics of the arm is applied to track a desired trajectory.

Keywords: 6-DOF robot arm - Autodesk inventor - Matlab SimMechanics - PI control - Arduino board - inverse kinematics.

I. INTRODUCTION

Rigid robots are dynamic systems that have multiple applications in industry, including welding, painting, and assembly of electronic parts. Additionally, robot manipulators can be used either in jobs where human life may be at risk or in processes that require a very high production rate. These numerous applications lead to the development of analytical tools to ensure better performance of the electromechanical systems.

The problem of the robot's positioning is a well-researched one in robotics, that is, the way in which its coordinates change. This is a relevant problem because robots are often used to move objects from one point to another. There are many research works in this field; many research contributions have been reported in the field of Mechanical modeling [1-6], trajectory planning [7-21], and robotic control schemes such as proportional-integration-derivative (PID) control [22], feed-forward compensation control (DFF)

[23], adaptive control [24-26], computed torque control (CTC) [27], and intelligent control [28-31].

The main goal of this paper is to design and to control the position of a 6-DOF robot arm. The objectives can be summarized as; to use the Matlab SimMechanics link toolbox to model and simulate the arm, and to develop an interface between the Matlab SimMechanics and Autodesk Inventor to animate the motion of the robot arm. The interface should be able to communicate with the robotic controller to send and retrieve information to/from the controller.

II. MODELING AND MECHANICAL DESIGN

The considered robotic manipulator is presented in Fig. 1. It is based on a five joint system which is popular in industry and a prismatic joint. Therefore, the experiments using the arm can be directly applied in the real business needs.

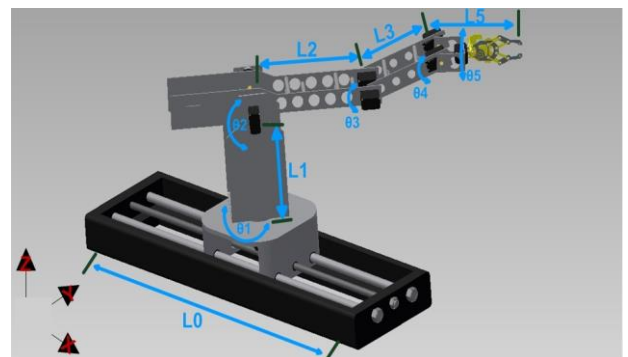


Fig.1: A six degree of freedom industrial manipulator

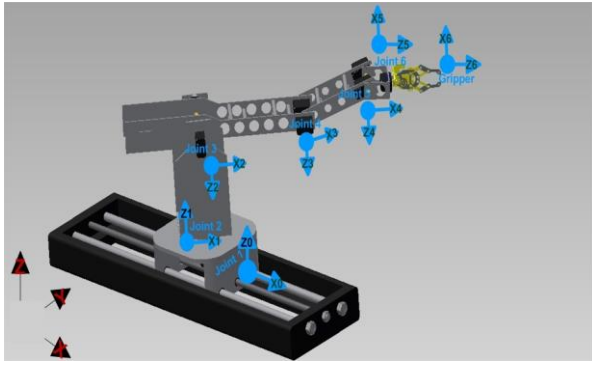


Fig.2: Frame assignment for 6-DOF industrial manipulator

Forward and Inverse Arm Kinematics:

A systematic technique for establishing the homogeneous matrix for each two adjacent links of manipulator was proposed by Denavit-Hatemberg (DH) in 1955. The DH convention is mainly implemented in robot manipulators, which consists of an open kinematic chain, in which each joint contains one-degree of freedom, and the joint is either revolute or prismatic. Links coordinate systems are established for each of the joints as shown in Fig. 2. Therefore, the joint-link parameters for the manipulator based on the frame assignment without link 1 (prismatic joint) are identified and tabulated in Table 1.

Table 1 D-H Parameters of the articulated arms.

Link	θ	d	a	α
Link 2	θ_1	L_1	0	90°
Link 3	θ_2	0	L_2	0
Link 4	θ_3	0	L_3	0
Link 5	$\theta_4 + 90^\circ$	0	0	90°
Link 6	θ_5	L_5	0	0

The overall transformation matrix for the manipulator from link 2 to link 6 is,

$${}^0T_5 = {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6$$

$$\begin{bmatrix} C_1 S_{234} C_5 + S_1 S_5 & -C_1 S_{234} S_5 + S_1 C_5 & C_1 C_{234} & C_1 (L_2 C_2 + L_3 C_{23} + L_5 C_{234}) \\ S_1 S_{234} C_5 - C_1 S_5 & -S_1 S_{234} S_5 - C_1 C_5 & S_1 C_{234} & S_1 (L_2 C_2 + L_3 C_{23} + L_5 C_{234}) \\ -C_{234} C_5 & C_{234} S_5 & -S_{234} & L_1 - L_2 S_2 - L_3 S_{23} - L_5 S_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The inverse kinematics analysis aims at finding the joint variables for a specific position and orientation of the end effector.

$$\theta_1 = \text{Atan2}(a_y, a_x) \quad (2)$$

$$\theta_5 = \text{Atan2} \left[\frac{\bar{T}}{T} (S_1 n_x - C_1 n_y, S_1 o_x - C_1 o_y) \right] \quad (3)$$

$$\theta_2 = \text{Atan2} \left(\frac{\bar{T}}{T} [\tan \theta_5 (C_1 n_x + S_1 n_y) + C_1 o_x + S_1 o_y, n_z \tan \theta_5 + o_z] \right) \quad (4)$$

$$\theta_3 = \text{Atan2} \left(\frac{\bar{T}}{T} \begin{bmatrix} -C_1 S_2 p_x - S_1 S_2 p_y - C_2 (p_z + L_1) + L_5 (C_1 S_2 a_x + S_1 S_2 a_y + C_2 a_z) \\ C_1 C_2 p_x + S_1 C_2 p_y - S_2 (p_z - L_1) - L_2 - L_5 (C_1 C_2 a_x + S_1 C_2 a_y - S_2 a_z) \end{bmatrix} \right) \quad (5)$$

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (6)$$

To decouple the orientation kinematics, three joints (Joint 1, 5, 6) were dispensed. The robot turns into 3-DOF sufficient to define the position of the hand with respect to three base frame axes x, y, z. Then the forward kinematics of 3-DOF will be,

$$X = \cos \theta_1 (L_2 \cos \theta_2 + L_3 \cos(\theta_2 + \theta_3)) \quad (7)$$

$$Y = \sin \theta_1 (L_2 \cos \theta_2 + L_3 \cos(\theta_2 + \theta_3)) \quad (8)$$

$$Z = L_1 + L_2 \sin \theta_1 + L_3 \sin(\theta_2 + \theta_3) \quad (9)$$

The inverse kinematics will be,

$$\theta_1 = \text{Atan2}(p_y, p_x) \quad (10)$$

$$\theta_2 = -\log \frac{[A + [B]^2]}{C} * i \quad (11)$$

$$A = -\left(\frac{p_y}{S_1}\right)^2 - (p_z - L_1)^2 - L_2^2 + L_3^2 \quad (12)$$

$$B = \left[-2L_2(p_z - L_1)^2 - \left(\frac{p_y L_1}{S_1}\right)^2 + \left(\frac{p_y}{S_1}\right)^2 + (p_z - L_1)^2 + L_2^2 - L_3^2 \right] \quad (13)$$

$$C = 2L_2(p_z - L_1) * i - \frac{2p_y L_2}{S_1} \quad (14)$$

$$\theta_3 = A \sin\left[\left(\frac{p_z - L_1}{L_2}\right) - \left(\frac{L_2 S_1}{L_2}\right) - S_2\right] \quad (15)$$

Mechanical Arm Model:

Features used to develop a system model of the manipulator were derived from the Autodesk. The dimensions of the links and the limits of joints motion are presented in Table 2 and 6061 aluminum sheets are considered.

Link	Mass (kg)	Length (mm)
L1	1.5	300
L2	0.5	290
L3	0.1	260
End-effector L5	0.05	110

Table 2 Robot parameters

Stress analysis:

The safety of the design and structure, with respect to stress and deflections respectively is checked using Autodesk Inventor. Figures 3 - 6 discuss the stress and strain of each link with external load 10N on the end-effector.

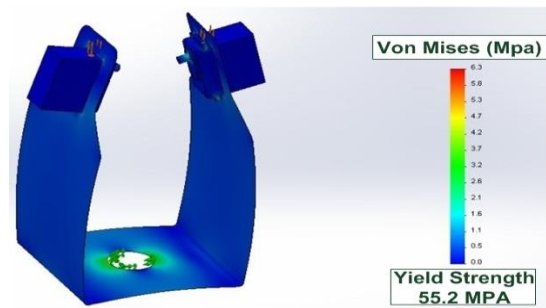


Fig.3: Stress affecting the 6mm base

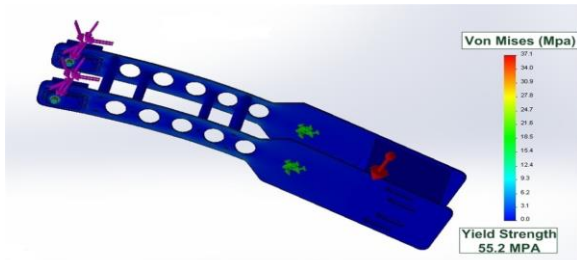


Fig.4: Stress distribution in link2

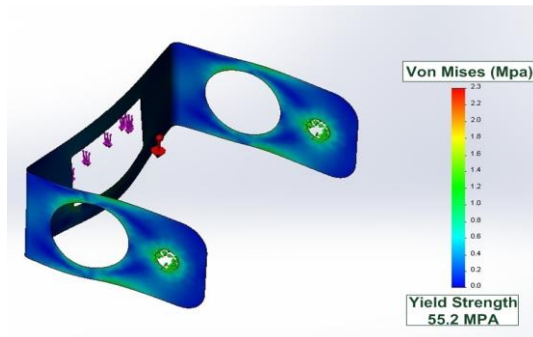


Fig.5: Stress distribution in link 3

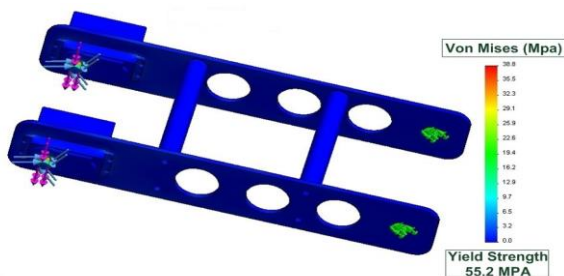


Fig.6: Stress distribution in link 4

III. JOINT SPACE TRAJECTORY PLANNING

Torque of each motor has been calculated based on dynamic of the robot and trajectory of each joint by using Autodesk inventor.

Joint Space Trajectory Planning: Defining both the initial location and orientation of the robot, then the inverse kinematic equations are used to find the final joint angles.

Consider one of the joints, which at the beginning of the motion segment t_i is at θ_i and its final position θ_f is at time t_f . One way to plan the trajectory is to consider a polynomial such that the initial and final joint displacement conditions are satisfied and that the velocities at the beginning and the end of the motion segment are zero (or known values). These six pieces of information allow solving for the six unknowns of the fifth degree polynomial in the form (16):

$$\theta = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (16)$$

Assuming that for joint 3:

Initial position = -50 deg, Final position = 170 deg, Initial velocity = Final velocity = 0 m/s, Initial acceleration = Final acceleration = 0 m/s², Time = 22 s

Then the polynomial parameters for joint 3 are obtained as:

$a_0 = -50$ deg, $a_1 = 0$ deg/s, $a_2 = 0$ deg/s², $a_3 = 0.189$ deg/s³,

$a_4 = -0.012$ deg/s⁴, $a_5 = 0.000214$ deg/s⁵

The trajectory of joint 3 is illustrated in Fig.7.

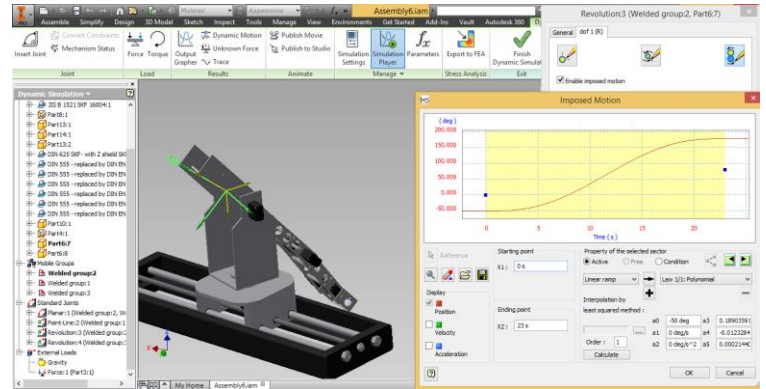


Fig.7: Input motion of Joint 3

The instantaneous velocity and acceleration during the considered time interval are shown in the Fig. 8-9.

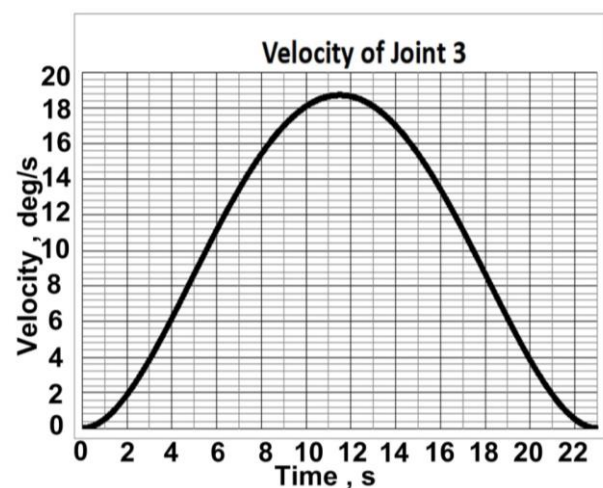


Fig.8: Velocity of Joint 3

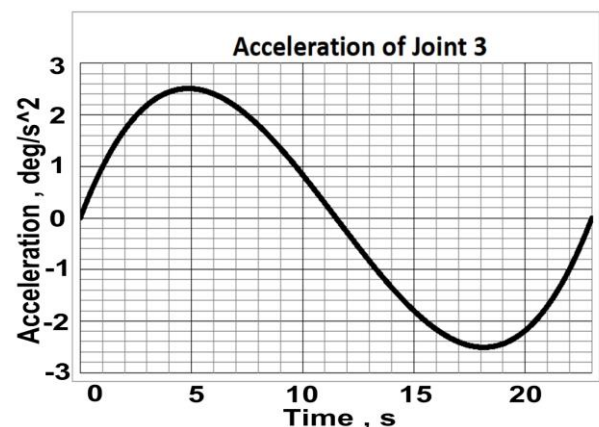


Fig.9: Acceleration of Joint 3

The motor torque for joint 3 based on previous trajectory to move this joint from θ_i (Initial position) to θ_f (Final position) with external force 10N on the end effector is presented in Figure 10.

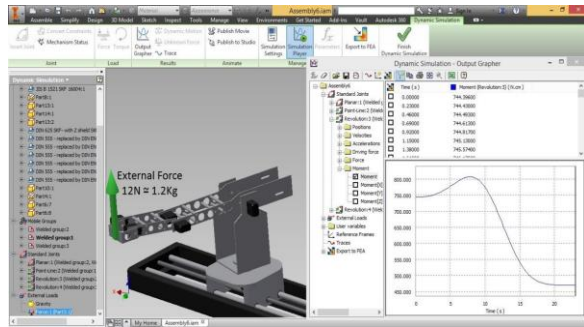


Fig.10: Torque behavior of Joint 3

Table 3 Torque required for each joint

Joint	Torque (N.m)
2	0.9
3	7.8
4	3.9
5	0.8

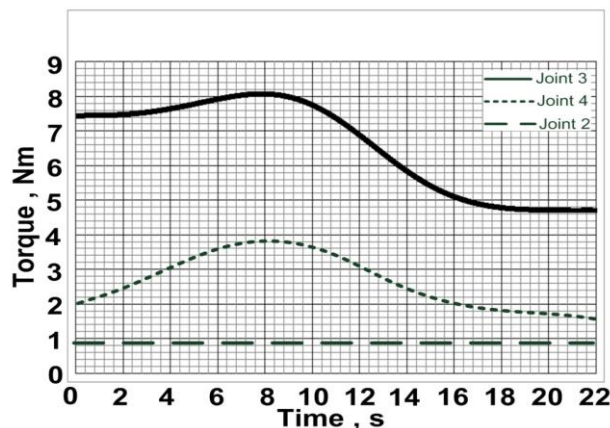


Fig.11: Comparison between torques required for each joint

Electrical subsystem (Servo Motors):

According to the dynamic analysis reported in the previous section, torque at each joint was estimated. The specifications are listed in Table 4

Table 4 Specifications of Servo Motors

Number of motors in each joint	Joint2 (2x4.0 Nm)	Joint3 (2x2.0 Nm)	Joint1 (1x1.5 Nm) 4(2x1.5 Nm) 5(1x1.5 Nm)
Voltage	6 -7.2V	6 -7.2V	6 -7.2V
Speed	0.16 sec/60degree (6v), 0.14 sec/60degree (7.2v)	0.16 sec/60degree (6v), 0.14 sec/60degree (7.2v)	0.16 sec/60degree (6v), 0.14 sec/60degree (7.2v)
Torque	3.8 Nm (6v), 4.0Nm (7.2v).	1.8 Nm (6v), 2.0Nm (7.2v), Stall torque	1.4 Nm (6v), 1.5 Nm (7.2v), Stall torque
Dimensions	59.2*29.2*54 mm	59.5 x29.2x51 mm	40.8x20.1x38 mm
Weight	1.65 N	1.64 N	0.56 N

The servo's position is controlled by providing pulse width modulation (PWM). The width of the pulses is measured in milliseconds. The rate at which the pulses are sent is not particularly important; it is the width of the pulse that matters to the control circuit. Typical pulse rates are between 400Hz and 50Hz.

The Servo.h library was used to provide the required PWM signal to the servo. Whenever sending a different angle value to the servo object, the code in the library takes care of sending the correct PWM signal to the servo. In this case the frequency is 50 HZ corresponding to a period of 20 ms, the pulse is varied from 1 ms to 2 ms. the pulse width represents the amount of rotation wanted from the servo. (The duty cycle is from 2% to 4% for a servo). The six pins of PWM in Arduino board (3, 5, 6, 9, 10, and 11) are used for all servos movement as shown in table 5. A stepper motor is used to actuate the prismatic joint.

Table 5 PWM pins of Arduino board and the usage.

Pin number	Usage
Pin 3	is used for the base motor
Pin 5	is used for the link 1
Pin 6	is used for the link 2
Pin 9	is used for the link 3
Pin 10	is used for the rotation of gripper
Pin 11	is used for the gripper open and close
Pin 7,8,12,13	is used for the Stepper motor

Position Sensing: Two perpendicular laser light tools are attached to the top at gripper to find out the point that gripper stands on. Two charts in X-Y plane and in X- Z plane are included to read the position of the gripper.

IV. CONTROL SYSTEM

SimMechanics Link:

Normal Simulink blocks represent mathematical operations, or operate on signals. Sim-mechanics is a Matlab® toolbox that enable importing Autodesk Inventor model from an Inventor environment to the Simulink environment as blocks of links connected by joints as outlined in Fig.12. Moreover, Sim-mechanics provides users with a tool to specify bodies and their mass properties, their possible motions, kinematic constraints, coordinate systems and the means of initiating and measuring motions. This makes it easier to do the analytical modeling without going through a complex modeling process. It saves the time and effort to derive the equations of motion. SimMechanics models, however, can be interfaced seamlessly with ordinary Simulink block diagrams.



Fig.12: SimMechanics Link Interface

The steps to build the dynamic model are:

1-Firstly, a CAD model needs to be built. Autodesk Inventor is used as a suitable platform for building such a model, as shown in Fig.1.

2-All body properties such as mass, moment of inertia or position of the central center of gravity were exported from a CAD/CAM environment – Autodesk Inventor. Initial simulation of the robot kinematics is conducted in Autodesk Inventor

3-The manipulator CAD model is then to be imported to Matlab Simulink as an XML file by the aid of Sim-mechanics toolbox.

4-The dynamic modelling in the Matlab/Simulink environment is completed using the SimMechanics library.

Fig. 13 presents a fragment of the block diagram of the dynamic model. It consists of six blocks represent six links. The linkage between each consecutive link-block is the joint block which is either prismatic or revolute. Since the case study considered in this work is focused only on the position control of the end effector, the prismatic joint and revolute joints 4 and 5 are fixed. The revolute joints (1, 2, and 3) are controlled and an actuator is added to each one of them. Revolute1 ($\Theta f1$), Revolute2 ($\Theta f2$), and Revolute3 ($\Theta f3$) are resulted from the controller and are fed to the joint actuators. The actual angular displacements of the revolute joints (1, 2, and 3) are read by adding sensor to each joint. These angles are used in the forward kinematics to determine the end effector position. The robotic arm can be visualized in the Matlab environment as in Fig. 14.

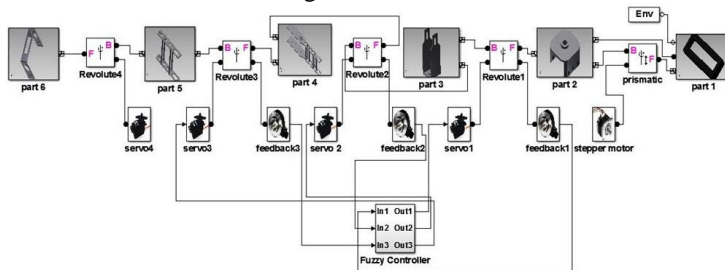


Fig.13: Fragment of the block diagram of the Robot dynamic model

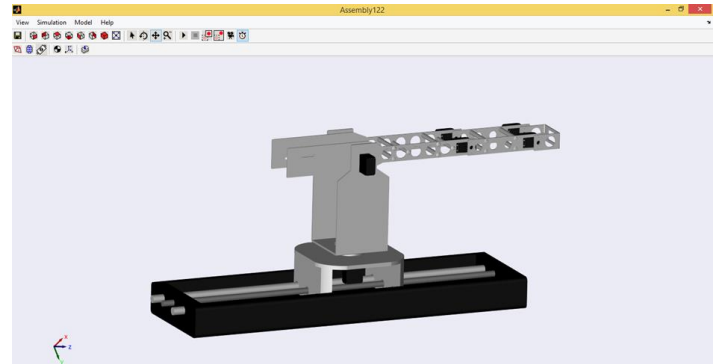


Fig.14: Robot visualization in the Matlab environment

PI Control:

Due to the high nonlinearities in the inverse of the robotic arm, it was difficult to synthesize a PID controller in order to track a desired trajectory. Therefore, using the inverse kinematics, the desired angular displacements are obtained and compared to the actual joint angular displacements. The errors (e) in joint displacements are fed to the PID controller as illustrated in Fig. 15. Three controllers (one for each joint) are used. The PID control output is Θ_i where

$$\Theta_s = k_p e + k_d \frac{de}{dt} + k_i \int e dt \quad (11)$$

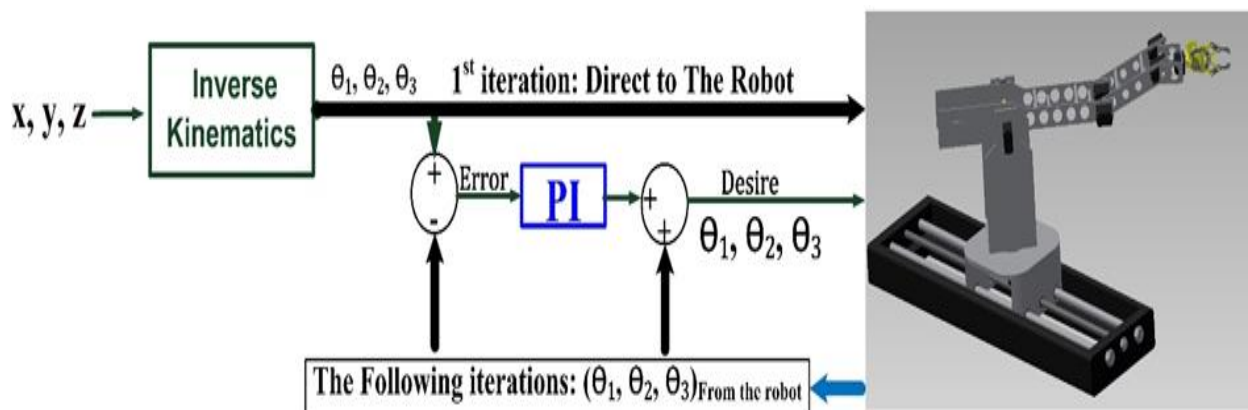


Fig.15: Using PI to Control the robot

The tuned PI parameters, that minimized the overshoot, steady-state error and the settling time, are: The proportional gain $k_p = 0.3$, The integral gain $k_i = 0.001$ and The derivative

gain $k_d = 0.0$. The PI- control system Simulink model is shown in Fig. 16.

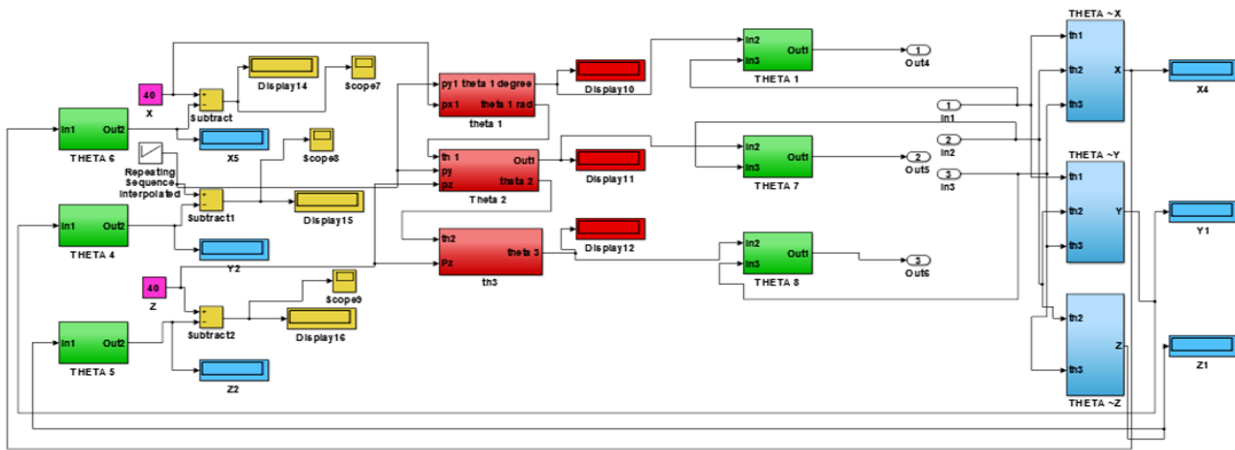


Fig. 16 PI-control system model at Matlab Simulink

V. RESULTS AND DISCUSSION

Point-to-Point Motion: It is assumed that the first point coordinates are 400, 400, 400 mm where the corresponding joint angles θ_1 , θ_2 and θ_3 are 45° , 43.7° and -59.47° , respectively. The performance of the PI control system is investigated for different specified second point coordinates. The error (e) in reaching the second point is considered as the norm of the error in the three directions.

$$e = \sqrt{e_x^2 + e_y^2 + e_z^2} \quad (12)$$

Where e_x , e_y , and e_z are the errors in x, y, and z directions, respectively.

Comparisons between the numerical and experimental results, upon implementing the PI controller, are given in figures 17- 19. It can be noticed from these figures that both results have the same trend but the errors associated with the experiments are higher than the numerical ones. The errors in the experimental results are related to:

- 1- Inaccuracy in position measurements using a laser light which must be perpendicular to the calibrator measurement of x, y and z axes.
- 2- Change in the lengths of the links due to bending deformation.
- 3- The accuracy of the servo motors used is 1 degree; therefore any angle less than 1° cannot be sensed by the servo encoder.

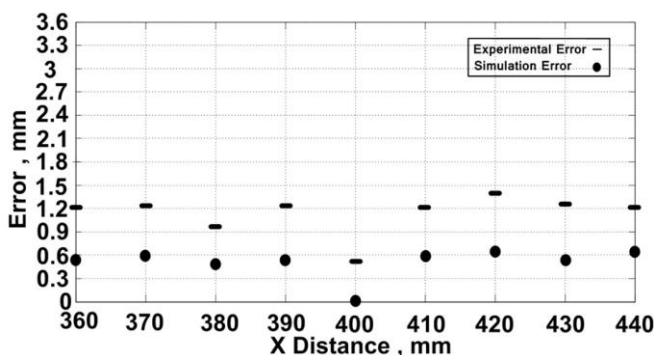


Fig.17: Comparison between numerical and experimental error with PI control as x-coordinate varies

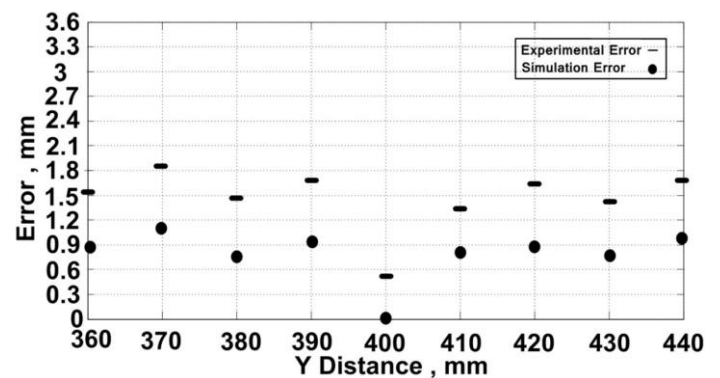


Fig.18: Comparison between numerical and experimental error with PI control as y-coordinate varies

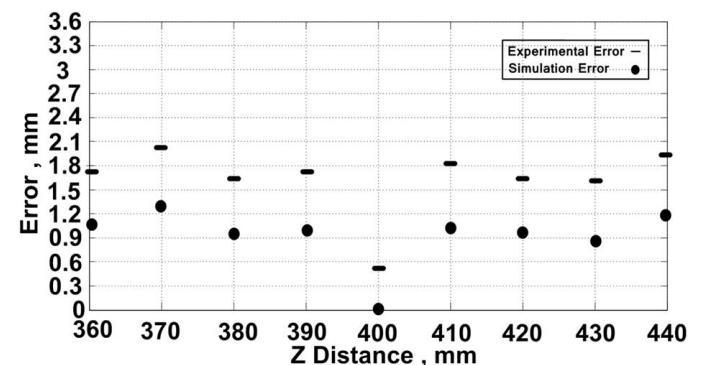


Fig.19: Comparison between numerical and experimental error with PI control as z-coordinate varies

The error response of the PI control system when $x = 440$ mm, $y = 440$ mm and $z = 440$ mm is illustrated in figure 20. For PI controller the error in X-direction is 1.1 mm, Y-direction is 1.1 mm, Z-direction is 1.2 mm.

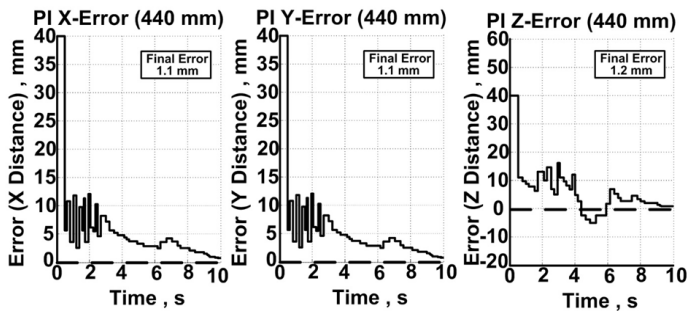


Fig.20: Error response of PI controller for motion in X, Y, Z-direction

CONCLUSIONS

The design and position control of a 6-DOF robot arm is presented. Autodesk inventor is used for the design and dynamic simulation of the arm. An interface between the Matlab SimMechanic and Autodesk Inventor is developed to animate the motion of the robot arm. The interface is able to communicate with the robotic controller to send and retrieve information to/from the controller. An experimental test rig for the arm is built. A PI controller is synthesized based on the inverse kinematics. It is tuned to eliminate the steady-state error between the desired and the actual displacement of the robot end effector, and minimize both the overshoot and settling time. The numerical simulation indicated that the controlled system response is satisfactory and the norm of the error is less than 1.2 mm. The experiments result in an error of 0.6 mm higher than that of the numerical one. Upon 100% variation of the values of weight of base link, no change is traced in the dynamic responses of hand displacement. The uncertainty in length of the third link on the performance of PI greatly affects the response. This is because the control action reflects only the inverse kinematics results and it is not sensitive to the position error. The uncertainty in external load on the performance of PI deteriorates as the load is more than 12.5 N (1.25 kg).

REFERENCES

- [1] Chung, J. and A. Velinsky, "Robust Control of a Mobile Manipulator – Dynamic Modeling Approach," Proc. American Control Conference, San Diego, California, USA, 2435-2439, 1999.
- [2] Mohri, A., Furuno, S. and Yamamoto, M., "Trajectory Planning of Mobile Manipulator with End- Effector Specified Path," Intelligent Robots and Systems, Maui, USA, 2264-2269, 2001.
- [3] Cheng, M. and Tsai, C., "Dynamic Modeling and Tracking Control of a Nonholonomic Wheeled Mobile Manipulator with Two Robotic Arms," IEEE Conference on Decision and Control, Maui, Hawaii, USA, 2932-2937, 2003.
- [4] Papadopoulos, E. and Poulakakis, J., "Planning and Model-Based Control for Mobile Manipulators," Intelligent Robots and Systems, IROS 2000, Japan, 1810-1815, 2000.
- [5] Tang, p. c., Krovi, V. N. and Agrawal, S.K., "Differential Flatness-Based Planning and Control of a Wheeled Mobile Manipulator – Theory and Experiment," IEEE/ASME, 16(4), 768-773, 2010.
- [6] Staicu, S., "Kinematics of the Caster Wheel of a Planar Mobile Robot," U.P.B. Sci. Bull., Series D, 72(4), 1-12, 2010.
- [7] Gurtler C., Vajta L., and Nagy I., "Trajectory Planning for Mobile Robots Based on Dynamical Models," Intelligent Engineering Systems. INES '97, IEEE, Budapest, 171-174, 1997.
- [8] Mechli, H., "Trajectory Planning for Mobile Robots in a Dynamic Environment," Vehicle Navigation and Information Systems Conference, Ottawa, Canada, 551-554, 1993.
- [9] Papadopoulos, E. and Poulakakis, J., "Planning and Model-Based Control for Mobile Manipulators," Intelligent Robots and Systems, IROS 2000, Japan, 1810-1815, 2000.
- [10] Khaled T., Atef A. and Bassuny M., "Dynamic analysis algorithm for a micro-robot for surgical applications," International Journal of Mechanics and Materials in Design, 7(1), 17-28, 2011.
- [11] Nagatani, K., Hirayama, T., Gofuku, A. and Tanaka, Y., "Motion Planning for Mobile Manipulator with Keeping Manipulability," Intl. Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, 2002.
- [12] Broquere, X., Sidobre, D. and Herrera, I., "Soft Motion Trajectory Planner for Service Manipulator Robot," IEEE Trans. On Robotics.
- [13] Brezak, M. and Petrovic, I., "Time-Optimal Trajectory Planning Along Predefined Path for Mobile Robots with Velocity and Acceleration Constraints," 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Budapest, Hungary, 942-947, 2011.

Robustness of PI Controller:

The uncertainties in the nominal values of the system parameters that are related to the robot construction are investigated. Upon 100% variation of the values of weight of base link, no change is traced in the dynamic responses of hand displacement and control demand.

Effect of change in the third link length (L3):

The original length of the third link is 370 mm. The effects of the uncertainties in length of the third link on the performance of PI control system are investigated. The end effector final position error is plotted in Figure 20 as L3 is changed from 200 mm to 500 mm. It can be noticed that for the performance of the PI controller is greatly affected by the uncertainties in L3. This is because the control action reflects only the inverse kinematics results and it is not sensitive to the position error.

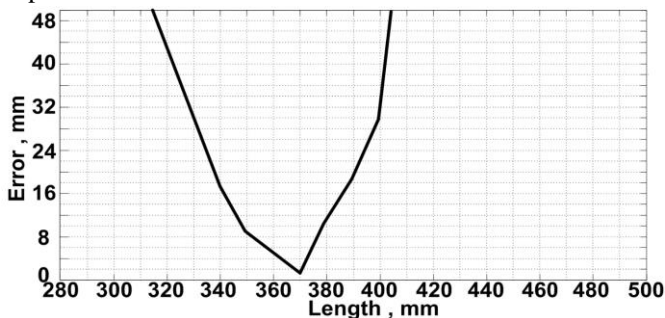


Fig. 21 Effect of change in third link length on PI performance

Effect of change in the load:

External uncertainties are investigated as the weight of the load is increased to 1.6 kg. The errors in the position of the gripper resulted according to the load increase upon using PI control are presented in figure 21. The performance of the PI-control deteriorates as the load is more than 12.5 N (1.25 kg).

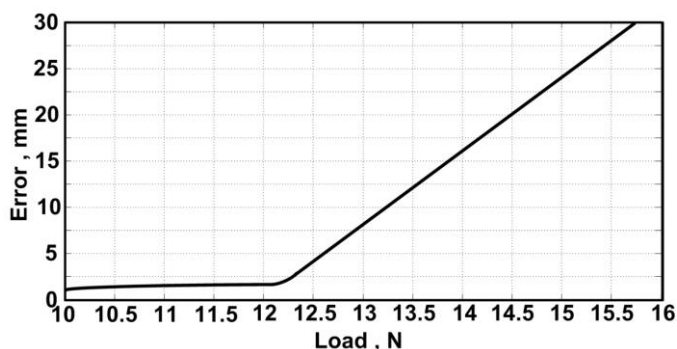


Fig. 22 Effect of change in load on PI performance

- [14] Gasparetto, A. and Zanutto, V., "A new method for smooth trajectory planning of robot manipulators," *Mechanism and Machine Theory*, 42, 455-471.
- [15] Ata, A.A., Rezeki, S.F., Shenawy, A. and Diab, M., "Design and Development of 5-DOF Color Sorting Manipulator for Industrial Applications," *International Journal of Mechanical Science and Engineering*, 7 (12), 734-741, 2013.
- [16] Receanu, D., "Modeling and Simulation of the Nonlinear Computed Torque Control in Simulink/MATLAB for an Industrial Robot," *Tech Science Press*, 10(2), 95-106, 2013.
- [17] Abbaszadeha, E. and Zarifb, M.H., "Control of Robot Manipulator with Uncertain Dynamics by Adaptive Fuzzy Controller," *Journal of Intelligent Automation Systems*, 2014.
- [18] Shahcheraghi, A., Piltan, F., Mokhtar, M., Avatefipour, O. and Khalilian, A., "Design a Novel SISO Off-line Tuning of Modified PID Fuzzy Sliding Mode Controller," *Information Technology and Computer Science*, 2, 72-83, 2014.
- [19] Mirzadeh, M., Haghighi, M., Khezri, S., Mahmoodi, J. and Karbasi, H., "Design Adaptive Fuzzy Inference Controller for Robot Arm," *Information Technology and Computer Science*, 9, 66-73, 2014.
- [20] Jahed, A., Piltan, F., Rezaie, H. and Boroomand, B., "Design Computed Torque Controller with Parallel Fuzzy Inference System Compensator to Control of Robot Manipulator," *Information Engineering and Electronic Business*, 3, 66-77, 2013.
- [21] Salehi, A., Piltan, F., Mousavi, M., Khajeh, A. and Rashidian, M.R., "Intelligent Robust Feed-forward Fuzzy Feedback Linearization Estimation of PID Control with Application to Continuum Robot," *Information Engineering and Electronic Business*, 1, 1-16, 2013.
- [22] Mailah, M., Pitowarno, E. and Jamaluddin, H., "Robust Motion Control for Mobile Manipulator Using Resolved Acceleration and Proportional-Integral Active Force Control," *International Journal of Advanced Robotic Systems*, 2(2), 125-134, 2005.
- [23] Abou Elyazed, M., Mabrouk, M., Abo Elnor, M. and Mahgoub, H., "Trajectory Planning of Five DOF Manipulator: Dynamic Feed Forward Controller over Computed Torque Controller," *International Journal of Engineering Research & Technology (IJERT)*, 4(9), 401-406, 2015.
- [24] Mbade, J. B., Ele, P., Mveh-Abia, C., Toure, Y., Graefe, V. and Ma, C., "Intelligent Mobile Manipulator Navigation Using Adaptive Neuro-Fuzzy Systems, IEEE International Conference on Robotics and Automation," New Orleans, USA, 447-474, 2005.
- [25] Hsu, Y., Chen, L. and Malki, H.A., "Fuzzy Neural Adaptive Controller Design: with Application to Multiple-Link Robot Control," *IEEE International Conference on Neural Networks*, 3, 1705-1709, 1997.
- [26] Boukattaya, M., Damak, T., and Jallouli, M., "Robust Adaptive Control for Mobile Manipulators, *International Journal of Automation and Computing*, 8(1), 8-13, 2011.
- [27] Chi-Wu, B. and Ke-fei, X., "Robust Control of Mobile Manipulator Service Robot Using Torque Compensation," *Proc. International Conference on Information Technology and Computer Science*, Norway, 69-72, 2009.
- [28] Hirukawa, T., Komada, S. and Hirai, J., "Image Feature-Based Navigation of Nonholonomic Mobile Robots with Active Camera," *SICE Annual Conference*, Takamatsu, Japan, 2502-2506, 2007.
- [29] Duan, P., Zhang, M. and Li, H., "The Mobile Manipulator Navigation Based on Fuzzy Control and Neural network," *Automation and Logistics IEEE International Conference*, Qingdao, 2358-2363, 2008.
- [30] Ming, A., Xie, Z., Yoshida, T., Yamashiro, M., Tang, C. and Shimojo, M., Home Service by a Mobile Manipulator System, *Information and Automation*, ICIA International Conference, China, 464-469, 2008.
- [31] Tsay, T.I.J., Lai, Y.F. and Hsiao, Y.L., "Material Handling of a Mobile Manipulator Using an Eye-in-Hand Vision System, *Intelligent Robots and Systems IROS*," IEEE/RSJ International Conference, Taipei, Taiwan, 4743-4748, 2010.