Oral Presentation

The 7th TSME International Conference on Mechanical Engineering
13-16 December 2016

DRC0008

Kinematics Analysis and Control Simulation of a 6-DOF Robot Arm for a Service Robot Application

Tossaporn Kongsujarit, Jeerapa Sookgaew and Pruittikorn Smithmaitrie*

Department of Mechanical Engineering, Faculty of Engineering, Prince of Songkla University,
Hat Yai, Songkhla, 90112
* Corresponding author: E-mail spruitti@me.psu.ac.th, Telephone Number 0-7428-7214

Abstract

This paper describes a solving method of inverse kinematics equations and simulation of a 6-DOF robot arm for a service robot. The robot arm is driven by cables which are connected to motors, and the motors are outside the arm structure. Joints of the robot arm consist of a 3-DOF shoulder, a 1-DOF elbow and a 2-DOF wrist. The link coordinates on each joint are based on the Denavit-Hartenberg principle. The algebraic solution technique and orthogonal rotation matrixes are applied for solving the kinematics equations. Gazebo software package is used for the robot arm simulation. In Gazebo, PID controllers are implemented to control the robot arm. Gravity, inertia and mass of the robot arm model are simulated according to the actual fabrication material and geometry. The desired target position of the robot hand is chosen, and then the robot arm trajectory is determined by the inverse kinematics equations. Simulation shows that the inverse kinematics are valid and show that the model can move to the desired position. The controlled robot arm system has low steady-state error. The settling time of travel to a position (x = 0.417 m, y = 0.151 m, z = 0.219 m) is 7.2 second. In the simulation, the PID gains of each joint are initially tuned by Good Gain method and later fine adjusted during implementation. The analysis and simulation model help improving the design of the robot arm and the PID gains can be used as a guide for estimation of the system response and selection of actuators.

Keywords: Inverse kinematics, Service robot, Digital servo motors, Simulation

1. Introduction

Service robots with basic tasks, such as navigation, object manipulation, object detection and human interaction are widely used nowadays for serving food, caring for older people, guiding in the museum, etc. This paper focuses on just the robot arm manipulation, controlling of the arm configuration and grasping an object.

In recent years, the focus of robot arms research and development are mostly for industrial use in which are not suitable for service robots due to its heavy weight and large size. This research focuses on service robots with appropriate arm size. The efficiency improvement of the robot arms movement depends mainly on the controller in which the set of kinematic equations are addressed.

This paper presents a method to determine forward and inverse kinematic equations of a 6-DOF robot arm based on Denavit-Hartenberg principle [1][2], the algebraic solving technique [3], the property of orthogonal rotation matrixes [4] and the robot simulation. The robot controller is initially developed by simulation using Gazebo [5] and ROS [6] software packages with Ubuntu 12.04 operating software. In Gazebo, PID controller [7][8] is applied on the controller system for improve the response of the system, settling time and the steady-state error of the system with Good Gain method [9] and later fine adjusted during implementation. The studied topics are arranged as follows: Design of the 6- DOF robot arm model, kinematics of a 6-DOF robot arm, simulation and conclusion.

2. Design of the 6-DOF Robot Arm

A 6-DOF robot arm is designed for grasping various objects. The robot arm is driven by cables and actuators located outside the arm structure, reducing the mass and inertia of the arm, increasing the payload capacity.

The robot arm consists of a 3-DOF shoulder joint (θ_1, θ_2, θ_3), a 1-DOF elbow joint (θ_4) and a 2-DOF wrist joint (θ_5, θ_6). The robot arm links are connected as a serial structure. Using the Denavit-Hartenberg principle, the joint reference frames are assigned as shown in Fig 1.

Fig 1. The robot arm joint reference frames.

3. Kinematics of the 6-DOF Robot Arm

According to the rotating reference frames, the robot arm parameters are presented in Table. 1 where α_{i-1} is the rotation angle from z_{i-1} axis to z_i axis about x_{i-1} axis, a_{i-1} is the distance between z_{i-1} axis and z_i axis along x_{i-1} axis, d_i is the distance between x_{i-1} axis to x_i axis along z_i axis, and θ_i is the rotation angle from x_{i-1} axis to x_i axis about z_i axis.

Table. 1 The robot parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i	Joint range
1	$0°$	0	0	θ_1	$-20° - 20°$
2	$90°$	0	0	θ_2	$-20° - 20°$
3	$-90°$	0	l_1	θ_3	$0° - 90°$
4	$90°$	0	0	θ_4	$-45° - 45°$
5	$90°$	0	l_2	θ_5	$-45° - 45°$
6	$-90°$	0	0	θ_6	$-45° - 45°$

3.1 Forward Kinematics

The forward kinematics are an analysis to determine position and orientation of the end effector's robot arm where all joints angles are known. The transformation matrix of the frame i relative to the frame i-1, ${}^{i-1}_{i}T$, can be written as

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

where $c\theta_i$ is defined as $cos\theta_i$ and $s\theta_i$ is defined as $sin\theta_i$.The transformation matrixes between each joint reference frames are as follows,

$$ {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

$$ {}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

$$ {}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ -s\theta_3 & -c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) $$

$$ {}^{3}_{4}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) $$

$$ {}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & -l_2 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6) $$

$$ {}^{5}_{6}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7) $$

Consequently, the transformation matrix solution describes the end effector frame $\{6\}$ relative to the fixed frame $\{0\}$. ${}^{0}_{6}T$ is

$$ {}^{0}_{6}T = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{5}T \cdot {}^{5}_{6}T \quad (8) $$

3.2 Inverse Kinematics

The inverse kinematics are an analysis to determine all joint angles for a given effector position and orientation. The given end effector configuration, ${}^{0}_{6}T$, can be written as

$$ {}^{0}_{6}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9) $$

where $\vec{n}$ is the normal vector, $\vec{o}$ is the orientation vector,$\vec{a}$ is the approach vector and $\vec{p}$ is the effector's position vector as shown in Fig 1.

By solving the algebraic equations [10][11] and the property of orthogonal rotation matrixes, the joint variables for a given effector configuration are presented as follows ;

$$ \theta_4 = Atan2(c\theta_4, \sqrt{1 - {c\theta_4}^2}) \quad (10) $$

where

$$ c\theta_4 = \frac{{l_1}^2 + {l_2}^2 - {p_x}^2 - {p_y}^2 - {p_z}^2}{2l_1 l_2}; $$

$$ \theta_6 = Atan2(b, a) + Atan2(\sqrt{a^2 + b^2 - c^2}, c) \quad (11) $$

where $a = 2l_2(o_x p_x + o_y p_y + o_z p_z)$,
$b = 2l_2(n_x p_x + n_y p_y + n_z p_z)$,
$c = {l_1}^2 - {l_2}^2 - {p_x}^2 - {p_y}^2 - {p_z}^2$;

$$\theta_2 = Atan2(s\theta_2, c\theta_2) \quad (12)$$

where

$$s\theta_2 = \frac{1}{l_1}\sqrt{\begin{matrix}\left(p_y - l_2(-o_y c\theta_6 - n_y s\theta_6)\right)^2 \\ +\left(p_x - l_2(-o_x c\theta_6 - n_x s\theta_6)\right)^2\end{matrix}},$$

$$c\theta_2 = \frac{p_z + l_2(o_z c\theta_6 + n_z s\theta_6)}{l_1};$$

$$\theta_1 = Atan2(c\theta_1, \sqrt{1 - {c\theta_1}^2}) \quad (13)$$

where

$c\theta_1 = [-p_x + l_2(-o_x c\theta_6 - n_x s\theta_6)]/l_1 s\theta_2$;

$$\theta_3 = Atan2(s\theta_3, c\theta_3) \quad (14)$$

where

$$s\theta_3 = \frac{\{c\theta_6(o_x s\theta_1 - o_y c\theta_1) + s\theta_6(n_x s\theta_1 - n_y c\theta_1)\}}{s\theta_4},$$

$$c\theta_3 = \{-c\theta_6\ [c\theta_2\ (o_x\ c\theta_1 + o_y\ s\theta_1) + o_z\ s\theta_2\ ] - s\theta_6\ [c\theta_2\ (n_x\ c\theta_1 + n_y\ s\theta_1) + n_z\ s\theta_2]\}/(s\theta_4);$$

$$\theta_5 = Atan2(s\theta_5, c\theta_5) \quad (15)$$

where

$$s\theta_5 = [a_x s\theta_1 c\theta_2 + a_y c\theta_1 s\theta_2 - a_z c\theta_2]/s\theta_4,$$

$$c\theta_5 = \{c\theta_6[-s\theta_2(n_x c\theta_1 + n_y s\theta_1) + n_z c\theta_2] - s\theta_6[-s\theta_2(o_x c\theta_1 + o_y s\theta_1) + o_z c\theta_2]\}/s\theta_4.$$

4. Simulation

Gazebo is used in conjunction with the ROS software to simulate the controlled response of the robot arm. The operating system of the computer is Ubuntu 12.04. Gazebo is capable of simulating gravity, inertia and control algorithm. A PID control system is implemented to control the robot arm motion. The function of the PID controller of Gazebo can be written as follows;

$$u = k_p e + k_i \int e\ dt + k_d \frac{de}{dt}$$

where k_p is the proportional gain, k_i is the Integral gain, k_d is the derivative gain, e is the error of system and u is the control signal.

In the controlled system, the forward and inverse kinematics equations are programmed into ROS software in C++ language. In validation of the equations, all joint angles are input into the forward kinematic algorithm for determination of the end effector configuration. After that, the end effector configuration is fed to the inverse kinematics algorithm to calculate all joint angles. Then, the calculated joint angles are used as inputs to Gazebo software for simulation. Procedure of the robot arm validation and configuration are illustrated in Fig 2.

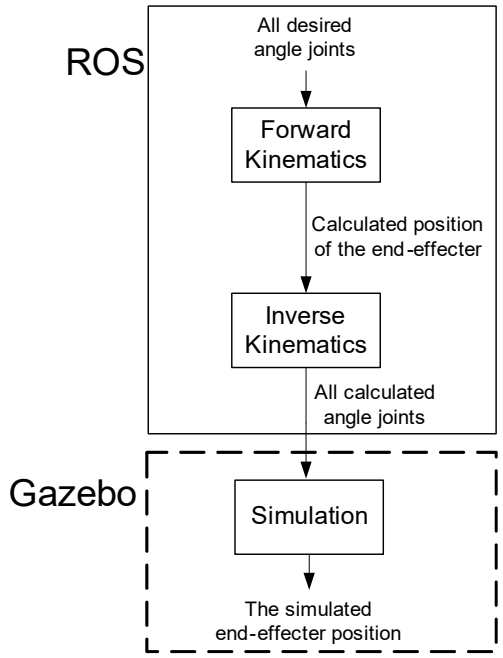


Fig 2. Validation of the inverse kinematics equation

4.1 Tuning PID gains

The Good Gain method is applied in the controller for improve its response; they are overshoot, settling time and steady-state error. The advantage of the Good Gain method is that it is a simple method, easy to understand, can be used on the real system or simulated system and does not require to get into the oscillated state during the tuning. Typical steps for tuning PID gains by Good Gain method are

i. Adjust k_p until the response of the controller gets good stability e.g. smooth response, low overshoot and low steady-state error. This gain value is called k_{pgg}

ii. Set k_i equal to

$$k_i = 1.5T_{ou}$$

where T_{ou} is the time between the first overshoot and first under shoot of the response of the controller

iii. Set k_p equal to

$$k_p = 0.8k_{pgg}$$

iv. Set k_d equal to

$$k_d = k_i/4$$

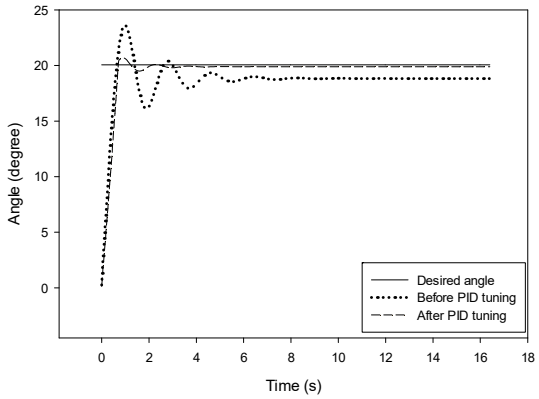
Then k_d will be manually adjusted during implementation to decrease overshoot and settling time.

Before tuning PID gains, the default PID gains of each joints of Gazebo are $k_p = 100, k_i = 0.1, k_d = 1$ and the responses of each joint are shown in Fig 3.

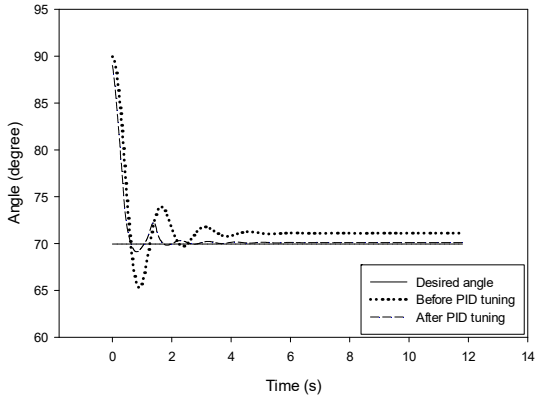
After tuning PID gains by Good Gain method and adjusting k_d during implementation, the simulation responses of each joint are shown in Fig 3. The corresponding gains are presented in Table 2.

Table. 2 The tuned PID gains of each joints by Good Gain method and adjusted k_d

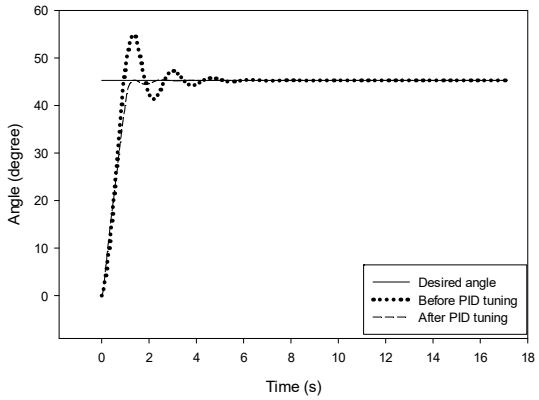
Joint	k_{pgg}	T_{ou}(s)	k_p	k_i	k_d
1	1000	1.2	800	1.8	10
2	1000	0.88	800	1.32	100
3	100	0.98	80	1.47	40
4	1000	0.9	800	1.35	10
5	80	1.18	64	1.77	30
6	100	0.84	80	1.26	30



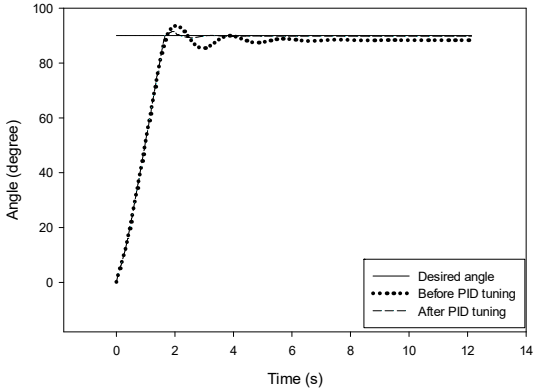
(a) Joint 1 response



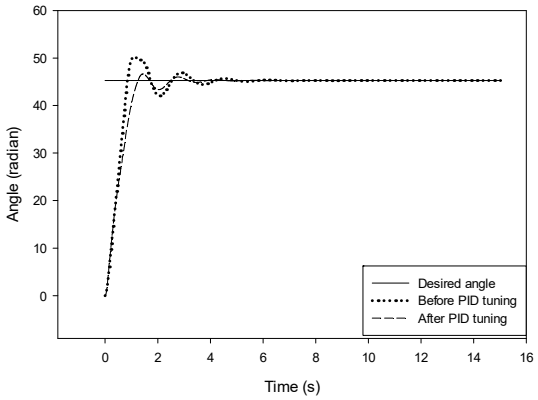
(b) Joint 2 response



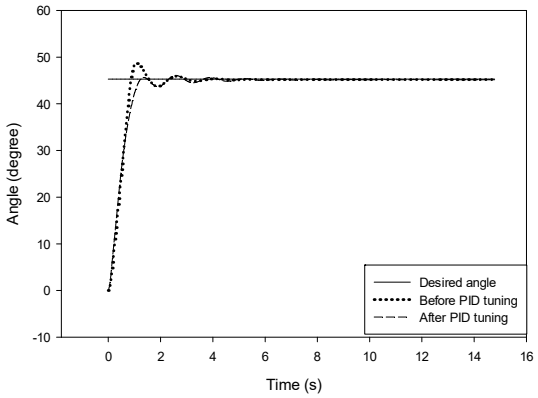
(c) Joint 3 response



(d) Joint 4 response



(e) Joint 5 response



(f) Joint6 response

Fig 3. The response of the controlled robot arm on each joint.

The response of the system; overshoot, settling time and steady-state error are shown in Table. 3. The system responses are compared between the default PID gains and the tuned PID gains based on Good Gain method and adjusted k_d.

Table. 3 The system characteristics of the joint responses.

Joint	Before PID tuning			After PID tuning		
	OS (%)	t_s (s)	e_{ss} (°)	OS (%)	t_s (s)	e_{ss} (°)
1	17.14	11.7	1.22	2.86	5.08	0.114
2	6.14	7.26	1.15	1.17	4.341	0.15
3	20.88	7.94	0	0.15	4.66	0
4	3.95	11.2	1.66	1.27	5.12	0.172
5	11.01	7.2	0	2.78	6.781	0
6	7.21	7.09	0.114	0.64	8.337	0.02

The results are in Table. 3 where OS is the overshoot (%), t_s is the settling time (s) and e_{ss} is the steady-state error (degree).

The result shown that steady-state error can be improved by Good Gain method tuning but overshoot and settling time are higher than of the default PID gain because k_p and k_i increase then k_d is adjusted to decrease overshoot and settling time.

4.2 Control of the Robot Arm Positioning

The initial joint angles of the robot arm in simulation are given $\theta_1 = 0°$, $\theta_2 = 90°$, $\theta_3 = 0°$, $\theta_4 = 90°$, $\theta_5 = 0° \; and \; \theta_6 = 0°$. The corresponding end effector initial position is

$$ {}^{0}_{6}T = \begin{bmatrix} -1 & 0 & 0 & 279.13 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 363.26 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

The initial joint angles of the robot arm configuration are shown in Fig. 4. When grasp objects on the table, This position of the robot arm will not hit the table because the robot arm is above the table.

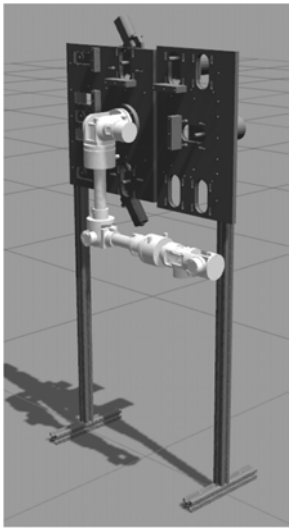


Fig. 4 The initial position and orientation of the robot arm.

One of a desired configuration of the robot arm is set as following joint angles: $\theta_1 = 20°$, $\theta_2 = 70°$, $\theta_3 = 0°$, $\theta_4 = 80°$, $\theta_5 = 20° and \theta_6 = 0°$.

Accordingly, the analytical end effector desired position is

$$ {}^{0}_{6}T = \begin{bmatrix} -0.647742 & -0.469846 & 0.599729 & 417.154 \\ -0.599729 & -0.17101 & -0.781716 & 151.832 \\ 0.429846 & -0.866025 & -0.17101 & 219.124 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

In the simulation, the final robot arm configuration is shown in Fig. 5. The end effector at the desired robot configuration is

$$ {}^{0}_{6}T = \begin{bmatrix} -0.647742 & -0.469846 & 0.599729 & 417.154 \\ -0.599729 & -0.17101 & -0.781716 & 151.832 \\ 0.429846 & -0.866026 & -0.17101 & 219.124 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

By comparing the analytical and simulation results of the effector transformation matrix, it shows that they are very close. The minor error is caused by substituting π value of 3.14159. Thus, the proposed forward and inverse kinematics solutions are validated.

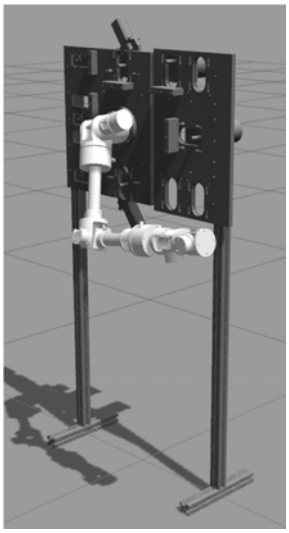


Fig. 5 The position and orientation of the robot arm at a desired configuration.

The end effector's trajectory in Cartesian space is shown in Fig. 6. The settling time of travel to the desired configuration is 7.2 second. The curve trajectory is caused by Point-to-Point (PTP) control [12]. The PTP control is a slew motion that all joint start simultaneously at the default joint speed. The end effector's trajectory is the result from motion of all links in the working space. In simulation, the default joints speed 0.5 rad/s. For PTP control, the robot arm moves from the start point to the desired point regardless what the path is.

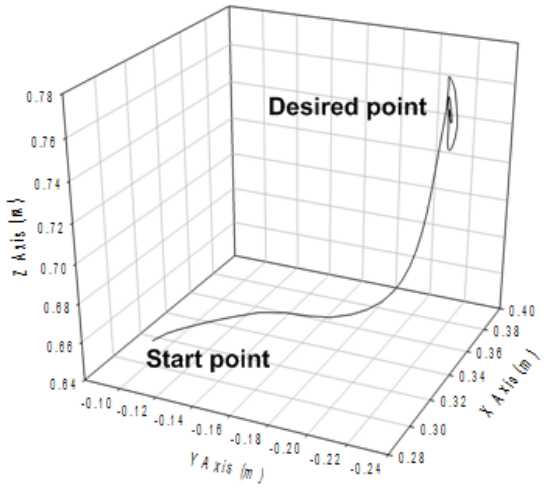


Fig. 6 The end effector trajectory.

5. Conclusion

Kinematics and control algorithm are critically important for controlling of the robot arm. Forward kinematics and inverse kinematics of the a 6-DOF robot arm for a service robot analyzed based on the D-H representation, algebraic equations and properties of the orthogonality are presented in the paper and are validated by the simulation of the end effector's robot. The PID gains are tuned by Good Gain method and later fine adjusted during implementation. When tuning PID gains, the controlled responses of each joints had better performance. The controlled robot arm system has low steady-state error, low overshoot and low settling time. The settling time of travel to a position (x = 0.417 m, y = 0.151 m, z = 0.219 m) is 7.2 second.

6. Acknowledgement

This research is supported by the smart mechatronics research unit, Prince of Songkla University.

7. References

[1] Saeed B. Niku, (2010). Introduction to Robotics, 2nd edition, John Wiley & Sons, New York.
[2] P. I. Corke, "A Simple and Systematic Approach to Assigning Denavit–Hartenberg Parameters," in *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 590-594, June 2007.
[3] John J. Craig. (2005). Introduction to Robotics Mechanics and Control, 3rd edition, Pearson Education International.
[4] Serdar Kucuk and Zafer Bingul (2006). Robot Kinematics: Forward and Inverse Kinematics, Industrial Robotics: Theory, Modelling and Control, Sam Cubero (Ed.).
[5] Open Source Robotics Foundation, *Gazebo*, URL: http://gazebosim.org/, access on 15/01/2016.
[6] Open Source Robotics Foundation, *Documentation*, URL: http://wiki.ros.org/, access on 10/01/2016.
[7] Fin Haugen (2004). PID Control, Fagbokforlaget.
[8] Antonio Visioli (2006). Practical PID Control, 1st edition, Springer.
[9] Fin Haugen (2010). The Good Gain method for PI(D) controller tuning, TechTeach.
[10] Adelhard Beni Rehiara (2011). Kinematics of Adept Three Robot Arm, Robot Arms, Prof. Satoru Goto (Ed.), InTech.
[11] G. S. Huang, C. K. Tung, H. C. Lin and S. H. Hsiao, Inverse kinematics analysis trajectory planning for a robot arm, *Control Conference (ASCC), 2011 8th Asian*, Kaohsiung, Taiwan.
[12] S. R. Deb, (2009). Robotics Technology and Flexible Automation, Tata McGraw-Hill Education Private Limited.