The 8th TSME International Conference on Mechanical Engineering
12-15 December 2017
Bangkok, Thailand

DRC0025

Optimal trajectory design for robotic arm

Mohamad Khairi Ishak [1,*], Muhammad Izzat Roslan and Elmi Abu Bakar[2,]

[1]School of Electrical and Electronic Engineering, Universiti Sains Malaysia,
[2]School of Aerospace Engineering, Universiti Sains Malaysia,
Engineering Campus, Nibong Tebal, 14300, Penang, Malaysia.

* Corresponding Author: khairiishak@usm.my

Abstract. The robotic kinematics is necessary for describing an end-effector's position, orientation and motion of all the joints, while trajectory planning is important for planning how robotic move based on velocity, time, and kinematics. This paper presents a design for the kinematics and trajectory planning procedures of the 3 Degree of Freedom (DOF) robotic arm. By extracting the parameters from the robotic arm, the kinematics model is built. Simulation is performed to verify that the kinematics model matches the real robot using MATLAB. The positive results from the simulation have shown that the robotic arm and end effector can reach a commanded position of the object within its workspace and desired trajectory planning path. Then, the robotic arm has been implemented in real-time environment for the pick and place operation.

1. Introduction

The robotic kinematics is important for describing an end-effector's position, orientation and motion of all the joints. The robotic kinematics studies the motion of a robot mechanism without considering of forces and torque that cause it [1]. It allows calculating the position and orientation of robot manipulator's end-effector related to the base of the manipulator as a function of the joint variables. Kinematics focuses on position, velocity, acceleration, and an accurate kinematics model must be established to investigate the motion of a robot manipulator.

Denavit-Haternberg (DH) convection was often employed in kinematics modeling of the robotics [2]. A coordinate frame was attached at each joint and the DH parameter for each link. There were four parameters to be obtained that was θi, di, ai, and αi. The link length ai, and link twists αi, which describe the relative location of the two attached joint axes in space. Joints are also described by two parameters, namely the link offset di, which is the distance from one link to the next along the axis of the joint, and the joint angle θi, which is the rotation of one link regarding the next about the joint axis [3]. Then, these parameters were applied to construct a DH parameters table. A transformation matrix between different frames can be obtained from the table. The forward kinematics equation was used to find the coordinates of the x, y, and z positions of the end effector. The forward kinematics transformation matrices can be obtained after DH parameter values were found.

Trajectory planning in the other hand relates to the way a robot is moved from one location to another in a controlled manner. It requires the use of both kinematics and dynamics of robots. Trajectory planning is concerned about when each of the paths must be attained, thus specifying timing. Trajectory planning is normally carried out in the joint space of the robot. Planning a trajectory in the joint space rather than in the Cartesian space has a major advantage, namely that the control

system acts on the manipulator's joints rather than on the end effector [4]. It would be easier to adjust the trajectory according to the design requirements if working in the joint space. Trajectory planning in the joint space would also allow avoiding the problems arising with kinematic singularities and manipulator redundancy. The main disadvantage of planning the trajectory in the joint space is that the motion actually performed by the robot end effector is not easily foreseeable, due to the non-linearity introduced when transforming the trajectories of the joints into the trajectories of the end effector through direct kinematics [5].

Simulation of robot systems is becoming very popular. The need for accurate and computationally efficient manipulator dynamics has been emphasized in recent years. Simulation can be used for layout evaluation, feasibility studies, presentations with animation and off-line programming. Building up new robot models and setting up experiments only takes a few hours. A simulated robotics setup is less expensive than real robots and it often runs faster than real robots with all the parameters are displayed on the screen [6]. A simulation is important for robot programmers because it allowing them to test and predict the behavior of a robot, and verify and optimize the path planning of the process. This will save time and money. Being able to simulate opens a wide range of options, helping to solve many problems. One can investigate, design, visualize and test an object before making it a reality [7]. The final design can be verified before one embarks on the costly and time-consuming process of building a prototype. Simulation of robots could be achieved using either of the following models, the geometrical model, the kinematic model and the dynamic model. The modeling and simulation of robot systems by using software will ease the process of designing, constructing and inspecting the robots in the real world. SolidWorks and MATLAB/Simulink software are often used in the simulation because it allows the robotic to be designed, visualize, check the theory [8]. Overall, a 3DOF robotic arm is developed to “pick and place” operation using MATLAB tools.

2. Robot and Kinematics Modelling

2.1. Robotic arm description

The robotic arm has 3 directions of motion (DOF) plus a grip movement (3+1) with 4 servo motors. Each RC servomotor represents the joints that enable movement of the robotic arm. The frame is made from the lightweight and durable aluminum to reduce overall robot arm weight and further lessen the load of the servo motor. Figure 1 shows the location of each servo motors on the robotic arm.

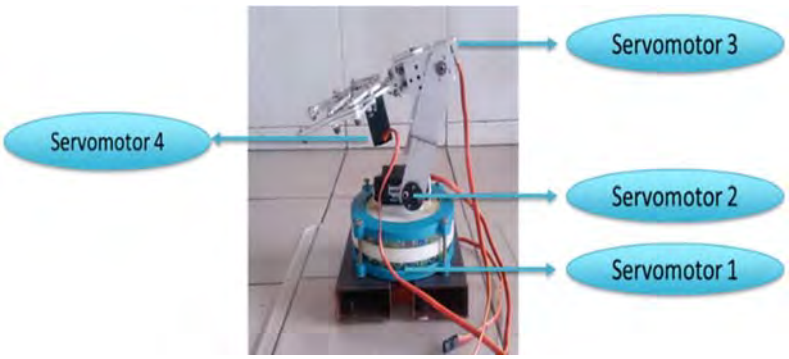

Figure 1: Location of each RC servo motor on the robotic arm

Robotic arm has three rotational joints and a moving grip. Joint 1 its axis of motion is z0. This joint provides a rotational θ1 angular motion around z0 axis in x0y0 plane. Joint 2 its axis is perpendicular to Joint 1 axis. It provides a rotational θ2 angular motion around z1 axis in the x1y1 plane. Z2 axes of

Joint 3 are parallel to Joint 2 z-axis and provide θ3 angular motions in x2y2 planes. A graphical view of all the joints is displayed in Figure 2.

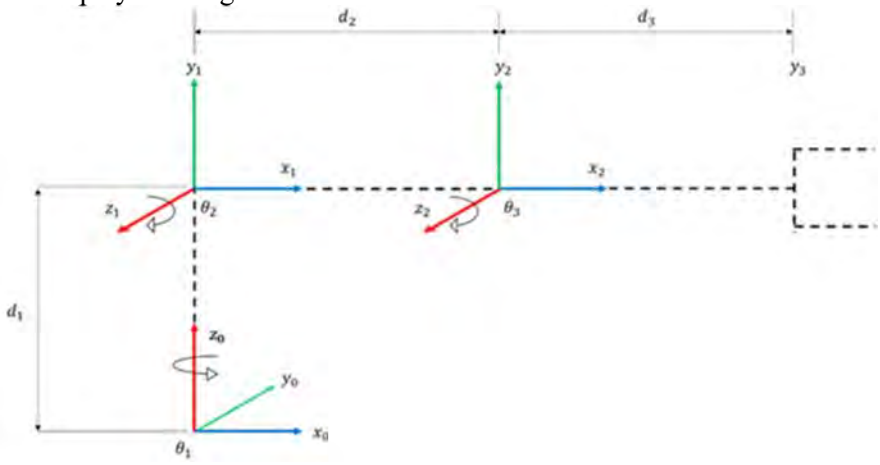

Figure 2: Line drawing of the reference frames for the 3DOF robotic arm

2.2. Forward kinematics

The Denavit-Hartenberg (D-H) analyze is used for the direct kinematics calculation. D-H parameters for the robotic arm are defined for the assigned frames in Table 1.

Table 1: DH parameter

i	α_i (degree)	a_i (cm)	d_i (cm)	θ_i (degree)
1	90	0	2	θ_1
2	0	0	11.5	θ_2
3	0	0	10	θ_3

The transformation between each two successive joints can be written by simply substituting the parameter from the D-H parameters table into the A matrix. For example, A1 between frames 0 and 1 (with sin 90 = 1 and cos 90 = 0 for α = 90° and designating C1 as cos θ1) as well as A2 through A3 for the other joint is:

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

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Calculating the position and orientation of the end effector with given joint angle is called Forward Kinematic analysis. Forward Kinematics equations are generated from the transformation matrixes shown in (1, 2 and 3) and the forward kinematics solution of the arm is the product of these four matrices as ${}^{0}T_3$ (with respect to base) shown in the equation (4).

$T_3^0 = A_1A_2A_3$

$$\begin{bmatrix} C_1C_2C_3 - C_1S_2S_3 & -C_1C_2S_3 - C_1C_3S_2 & S_1 & d_2S_1 + d_3S_1 \\ C_2C_3S_1 - S_1S_2S_3 & -C_2S_1S_3 - C_3S_1S_2 & -C_1 & -d_2C_1 - d_3C_1 \\ C_2S_3 + C_3S_2 & C_2C_3 - S_2S_3 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The first three columns in the matrices represent the orientation of the end effectors, whereas the last column represents the position of the end effector. The orientation and position of the end effector can be calculated in terms of joint angles. Figure 3 show the relationship between joints variables with position and orientation of the end effector.

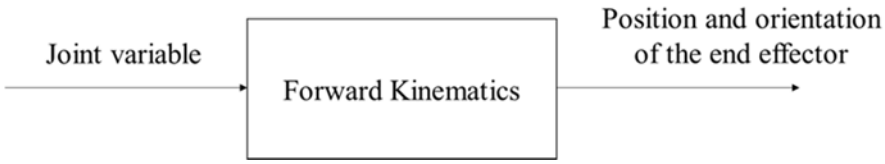

Figure 3: Forward kinematics relationships between joint variable and position and orientation of the end effector

2.3. Trajectory planning

Trajectory planning relates to the way a robot is moved from one location to another in a controlled manner. Trajectory planning approximates the desired path by a class of polynomial functions and generates a sequence of time-based control set points for the control of manipulator. In this project, the desired joints have to move to a new value of θ_f at the time t_f. The way to do this is to use a polynomial to plan a trajectory, such that the initial and final boundary match what already know, namely, that θ_i and θ_f are known, and that the velocities at the beginning and the end of the motion are zero. These four pieces of information will allow solving for the four unknowns or a third-order polynomial in the form of:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad \text{For position} \quad (5)$$

$$\theta(t) = a_i + 2a_2t + 3a_3t^2 \quad \text{For velocity} \quad (6)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3t \quad \text{For acceleration} \quad (7)$$

Where the initial and final conditions are:

$$\theta(t_i) = \theta_i \quad (8)$$

$$\theta(t_f) = \theta_f \quad (9)$$

$$\dot{\theta}(t_i) = 0 \quad (10)$$

$$\dot{\theta}(t_f) = 0 \quad (11)$$

The trajectory planning for the pick and place operation in this project is shown in Figure 4. The robotic arm will move from the initial position to the position 1 in 3 seconds and will continue until it returns to its initial position. The round trip time for this operation is 12 seconds.

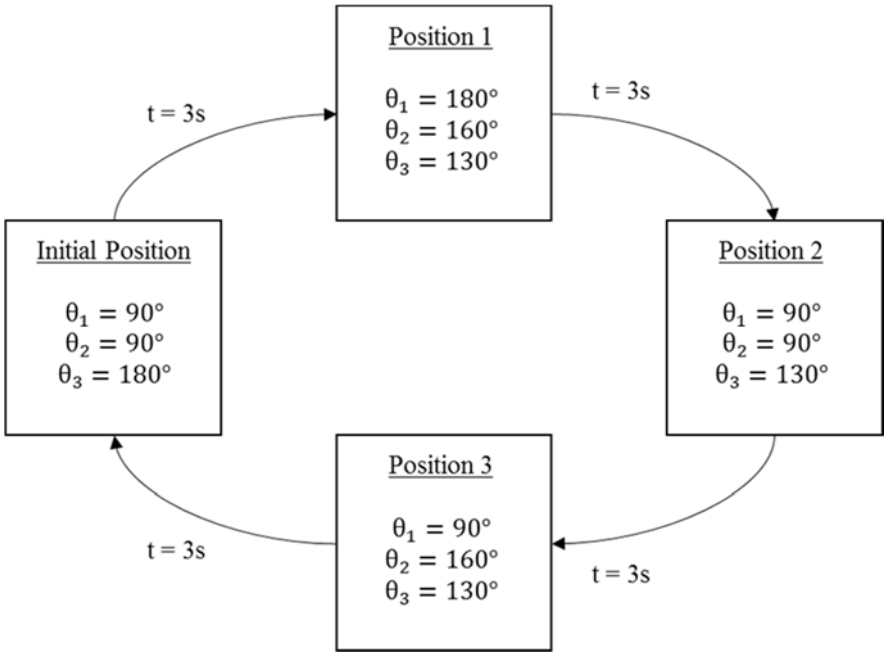

Figure 4: Pick and place operation

3. Design and simulation

Simulation is an imitation of the operation of a robot design in real-world. The motion of the robot is simulated to perform task and system parameters are calculated such as joint variable, angular displacement, angular velocities, angular accelerations, homogenous transformation matrix and other. Besides, simulation of robotic arm motion is performed to determine whether the robot design can perform the given task perfectly. Peter Corke Robotic Toolbox is used in MATLAB to simulate the robotic arm. GUI and trajectory planning programs are created in MATLAB simulation.

3.1. Robot plotting in MATLAB

The robotic arm that being simulated is a 3-DOF robot with the end effector of the gripper which has the ability to pick and place objects. By plotting the known Denavit-Hartenberg Parameters, the variables can be changed and observe without truly interacting with the real model. Graphical User Interface (GUI) is created in MATLAB simulation where the individual joint variable can be adjusted and the end position of the gripper is displayed in x-y-z coordinates.

3.2. Trajectory planning in MATLAB

Trajectory planning of the designed robotic arm in MATLAB simulation required the use of forward and inverse kinematics, First, D-H parameters of the designed robotic arm are set. After that, the position of the end effector is translated to the desired x-y-z coordinates. Then, the angular displacement of each joint variable θ1, θ2 and θ3 are obtained from the inverse kinematics of transformation matrix of translation motions. To complete the planned task in MATLAB, there are total 18 desired x-y-z coordinates, 18 transformation matrix and 18 set of joint variables θ1, θ2 and θ3. For zero initial boundary velocity and acceleration, a number of steps, N equal to 5 is used in trajectory generation equation. The trajectory generation is obtained by substituting the starting joint variable, final joint variable, and N for each sequence of motion of the robotic arm. Although angular displacement, angular velocity, and angular acceleration of joint are obtained, only angular displacement is concerned in the simulation of robotic arm motion. Finally, the virtual robotic arm that

performing given task is shown by using robot plotting method which already mentions above. Looping such as while and for loop is needed to ensure the robot plotting run continuously and thus virtual robotic arm motion is created.

4. Results and discussion

4.1 Simulation results of the Robotic Arm in MATLAB

There were two simulation results in MATLAB to be displayed in this sections which are GUI simulation results and trajectory planning results. Figure 4.1 shows the designed robotic arm plot in the workspace. It displays a graphical representation of robotic arm based on D-H parameter. The created GUI is shown in Figure 5. The slider bars allow the user to slide the respective joint variable to the desired joint angles and the user also can insert their desired joint angles into the text box to control the motion of the robotic arm as shown in Figure 5. The results are shown in x-y-z and raw-pitch-yaw coordinates. The results proved the capability of the robotic arm to be controlled in a real environment with given workspace.

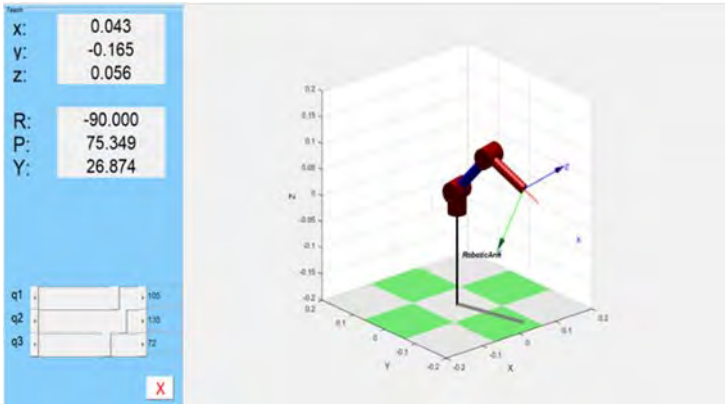

Figure 5: Robotic arm simulation in MATLAB

MATLAB is used to calculate the third-order polynomial trajectory planning based on the position desired during pick and place operation. The joints angles, velocities, acceleration are shown in Figures 6, 7 and 8 for entire motion.

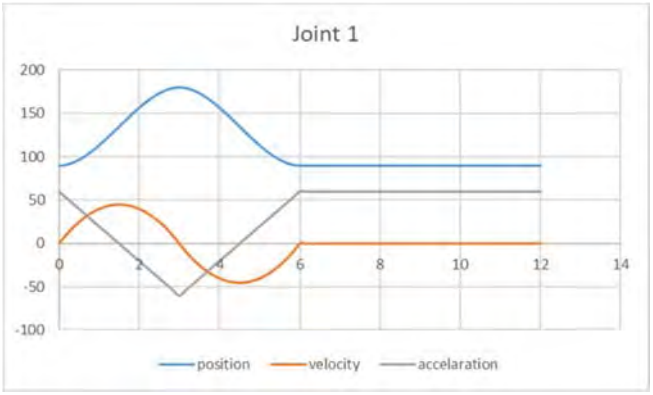

Figure 6: joint 1 position, velocity, and acceleration

Figure 7: Joint 2 positions, velocity, and acceleration

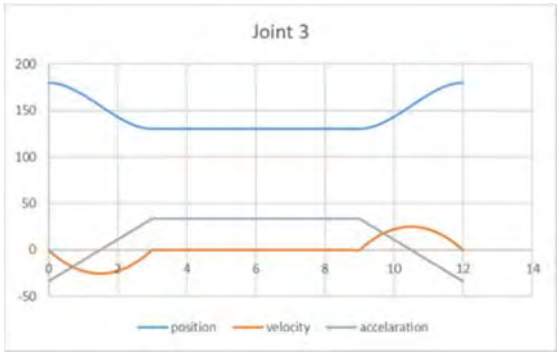

Figure 8: Joint 3 positions, velocity, and acceleration

Figure 9 shows a trajectory planning of robotic arm in MATLAB. It proved that the robotic arm end effector can reach to the location of the object within its workspace and desired trajectory planning path.

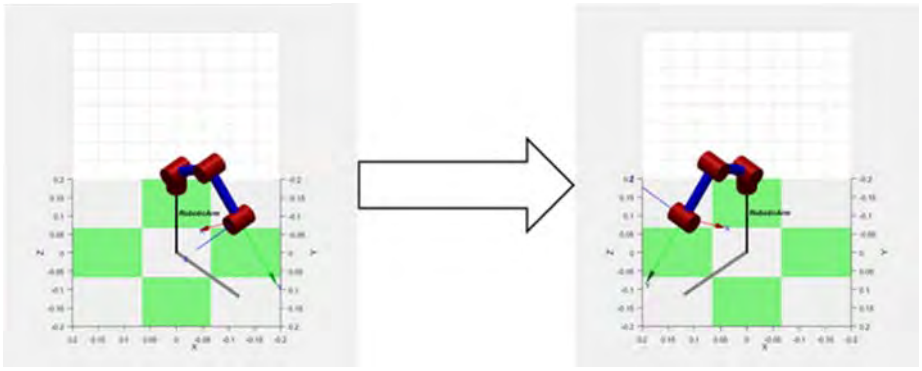

Figure 9: Trajectory planning of robotic arm in MATLAB

4.2 Pick and place operation

Once the modeled has been established, the robotic arm has been implemented in practical test. The robotic arm is able to do pick and place operation in a real-time environment as shown in Figure 10.

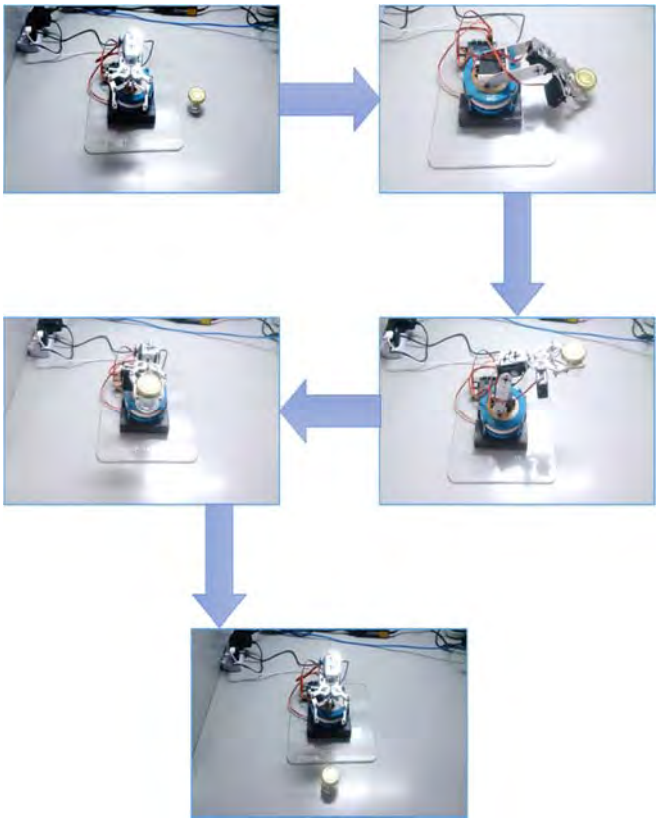

Figure 10: Pick and place operation

5 Conclusion

In this paper, a complete kinematics analysis of 3DOF robotic arm has been investigated and modelled. MATLAB is used to verify the theory and the simulation of the robotic arm motion. Graphical User Interface (GUI) has been developed to test and simulate the motion of the robotic arm. The robotic arm has been implemented in real-time environment according to the established model.

References

[1] P. Corke, *Robotics , Vision* and Control, 2011 (Springer).

[2] I. These, M. Rv-, and a J. I. Robot, "Modeling and simulation of robot arm 1.," vol. 3, no. 4, pp. 220–229, 2015.

[3] S. B. Niku, *Introduction To Robotics: Analysis,Control,Applications, 2nd Edition*. Wiley India Pvt. Limited.

[4] A. Gasparetto and V. Zanotto, "A new method for smooth trajectory planning of robot manipulators," *Mech. Mach. Theory*, vol. 42, no. 4, pp. 455–471, 2007.

[5] G. Sahar and J. M. Hollerbach, "Planning of Minimum- Time Trajectories for Robot Arms," *Int. J. Rob. Res.*, vol. 5, no. 3, pp. 90–100, 1986.

[6] O. Michel, "'Webots TM : Professional Mobile Robot Simulation,'" *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, pp. 39–42, 2004.

[7] L. Žlajpah, "Simulation in robotics," *Math. Comput. Simul.*, vol. 79, no. 4, pp. 879–897, 2008.

[8] M. Gouasmi, M. Ouali, B. Fernini, and M. Meghatria, "Kinematic modelling and simulation of a 2-R robot using solidworks and verification by matlab/simulink," *Int. J. Adv. Robot. Syst.*, vol. 9, pp. 1–13, 2012.