

Animation of 2R Planar Manipulator to Determine the Time Behavior of Cubic Polynomial

S. Suraj

U.G Student,
Dept. of Robotics and Automation,
P.S.G College of Technology,
Coimbatore – 614004

Anbarasi M P

Assistant Professor (Sr. Gr.),
Dept. of Robotics and Automation,
P.S.G College of Technology,
Coimbatore – 614004

Niresh J

Assistant Professor (Sr. Gr.),
Dept. of Automobile Engineering,
P.S.G College of Technology,
Coimbatore – 614004

Abstract:- Robots are widely used in automation production system. With the development in technology, robotics is used widely in various fields. In recent times, many research studies have been done on autonomous robotic manipulators. A challenging task in research is how to quickly, accurately and smoothly grasp objects. The robot arm being the executive mechanism for the robot, plays a major role in completing an assigned task. The kinematics, control theory and computation are the important subjects involved in the field of robotic arm. This paper represents the idea to make the trajectory of the 2R planar smooth. In order to make the trajectory smooth there are constraints placed on position and velocity, if the initial and final value for joint angle and velocity is known, one can apply the cubic polynomial to make the path smooth. This work aims on determining the time behavior of cubic polynomial using MATLAB software.

I. INTRODUCTION

Since the industrial revolution, there has been a demand to reduce manufacturing cost and improve quality of products. The inflexibility and high cost of hard automation have led to the introduction of robotic manipulators for production tasks because they are controlled by computers or microprocessors which can be easily reprogrammed [1]. In many fields, high technological standards are difficult to attain without robots [2]. In today's world, robots and humans are working hand-in-hand for completion of their assigned task [3]. Robots are used to perform unsafe, hazardous, highly repetitive and unpleasant tasks. Manipulators are used extensively in industries because it can carry heavy payloads and does work more faster and smarter which in turn increases the productivity [6]. It has many different functions such as material handling, assembly, arc welding, resistance welding, machine tool load and unload functions, paintings, spraying, etc [4]. Therefore, It can be foreseen that with the improvement of intelligence in robot it is possible to improve the efficiency, save cost, avoid risk of personnel and ensure personal safety [7]. Now a days robots accompany people in day-to-day life and help them in variety of tasks. The range of robot utilization is very wide, from toys to very complex robots [5]. The ever-changing nature of human environment proposes great challenges to robot manipulation [9]. In recent years, the market of service robotics has been developing actively. The main part of the market of professional in value terms is covered by medical devices.

Agriculture and logistics are also areas which are actively developing [10]. One of the early stages of designing a robot arm is to deal with its kinematics [6]. It is always better and useful to perform simulations before investigations with real robots [5]. This paper presents the idea to animate the motion of the 2R manipulator subjected to constraints of cubic polynomial using MATLAB. A two degree of freedom robot with two revolute joint is used in this paper for understanding and this 2R robot is generally suited for small parts insertion and assembly, like electronic component insertion. Thus, for each link it is possible to obtain the trajectory planning which is the time behavior of the intermediate configurations of the links between the initial and final destination.

II. MANIPULATORS

In robotics, a manipulator is a device used to manipulate materials without any direct physical contact by the operator. Manipulators are made of an assembly of links and joints. Links are the rigid part and joints serves as the connection between two links which can be seen in Fig 1.



Fig 1. Links and joints

The device attached to the end of the manipulator that interacts with the environment to perform tasks is the end effector. The joints may be of various types like Revolute, Cylindrical, Prismatic, Spherical, Planar as shown in Fig 2. Revolute joint allows relative motion about an axis. Cylindrical joint allows relative rotation and translation about one axis. Prismatic joint allows relative translation

about an axis. Spherical joint allows three degrees of rotational freedom about its center of joint. Planar joint allows relative translation on a plane and relative rotation about an axis perpendicular to that plane.






Revolute	
Cylindrical	
Prismatic	
Spherical	
Planar	

Fig 2. Joints and its representation

Generally, a robotic arm is a mechanical arm which is programmable and has similar functions to that of a human arm. The arm may be a part of a more complex robot or the total mechanism itself. Manipulators are classified into Planar manipulators, Spherical manipulators, Spatial manipulators, Open loop manipulators, Parallel manipulators, Hybrid manipulators. Planar manipulator is a manipulator in which all the moving links move in planes parallel to one another as illustrated in Fig 3. In Spherical manipulator all the links perform spherical motions about a common stationary point as seen in Fig 4. A Spatial manipulator has at least one of its links to possess a spatial motion as shown in Fig 5. An Open loop manipulator has its links forming an open loop chain, Parallel manipulators is made up of a closed loop chain as demonstrated in Fig 6 & 7. A Hybrid manipulator is a combination of open and closed loop chain which is illustrated in Fig 8.

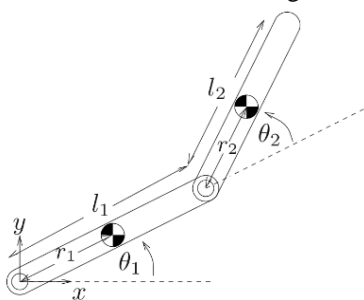


Fig 3. Planar manipulator

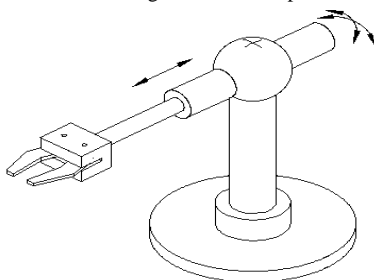


Fig 4. Spherical manipulator

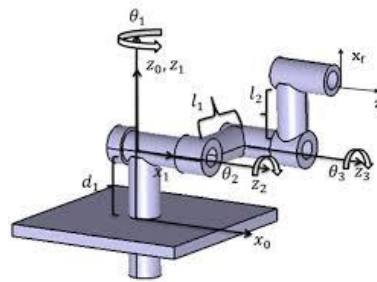


Fig 5. Spatial manipulator



Fig 6. Open loop manipulator



Fig 7. Parallel manipulator

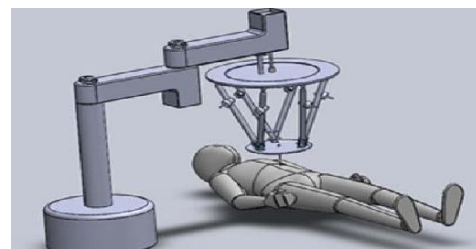


Fig 8. Hybrid manipulator

III. TRAJECTORY PLANNING

The main objective is to move the end effector from one point to another. The user needs to just specify where to move rather than telling what position, velocity and acceleration it should have at every instant instead the controller should calculate the position, velocity and acceleration with which it should move.

First, the total time taken for the arm to move from the initial joint angle to the final joint angle is divided into equal divisions. Now the control signals from the controller are released to the actuators by utilizing a n^{th} degree polynomial function where $(n \geq 3)$ at each instant of time which ensures the smooth motion of the link. This in turn reduces the jerks and vibrations during motion which leads to reduced wear and tear. Consider a rest-to-rest motion of a 2R manipulator, i.e., the initial and final velocities must be zero. In total there are 4 constraints, the initial and final position, initial and final velocity. To satisfy these four constraints we need a cubic polynomial.

Initially at $t = 0$, $\theta(t_i) = \theta_i$, $\dot{\theta}(t_i) = 0$

Finally at $t = t_f$, $\theta(t_f) = \theta_f$, $\dot{\theta}(t_f) = 0$

$\theta(t) = c_0 + c_1t + c_2t^2 + c_3t^3$ (1)

$\dot{\theta}(t) = c_1 + 2c_2t + 3c_3t^2$ (2)

Using the initial and final conditions, the constants are obtained.

Finally, considering each condition the joint variable equation is obtained as:

$\theta(t) = \theta_i + \frac{3(\theta_f - \theta_i)}{t_f^2} t^2 - \frac{2(\theta_f - \theta_i)}{t_f^3} t^3$ (3)

θ_f - Final value of θ

θ_i - Initial value of θ

If the initial, final position and the final time is known, the joint variable θ can be calculated at any instant. Now, this can be applied in MATLAB software for a 2R planar manipulator.

The following are the steps involved in the simulation. First, define n to be the number of sample points in the trajectory. Then define link lengths a_1 and a_2 , after which the final time is defined as 1. Now using linspace function the interval 0 to 1 is divided into 100 segments and is scaled with t_f (final time). Next, define the initial and final values of θ for both link 1 and link 2 in degrees and then convert it into radians since MATLAB by default uses radians. Define a loop inside which the value of θ_1 and θ_2 is calculated for every iteration using the cubic joint variable equation (Eqn. 3). The values of coordinates in every iteration for each link is calculated using forward kinematic equations and then it is plotted on a graph. In Forward Kinematics the required position and orientation are determined from a given set of joint angles [8]. The values of θ_1 and θ_2 is stored in an array. This array is used to plot the values of θ_1 and θ_2 with respect to time. Applying Forward Kinematics, (x_2, y_2) as shown in Fig. 9 are obtained, which gives the position and orientation for the 2R manipulator.

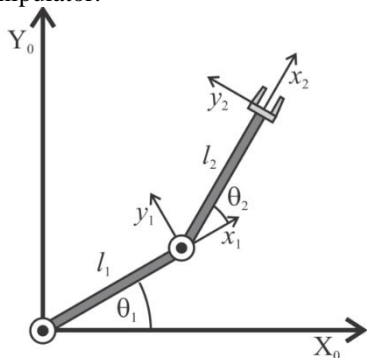


Fig 9. 2R Manipulator (Forward Kinematics)

We get,

$x_2 = l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1$ (4)

$y_2 = l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1$ (5)

These equations help to calculate the coordinates of the end effector. Then another window is created to plot θ_1 and θ_2 against time, then label x and y coordinates.

IV. SIMULATION RESULT

A simulation is less expensive than a real robot and real-world setups [5]. Simulations are easier to setup, faster, less expensive and more convenient [5]. The simulation is performed in MATLAB software and it is observed that the link 1 (red) and link 2 (green) starts to move gradually from the initial position. And finally, the 2R manipulator reaches its final destination.

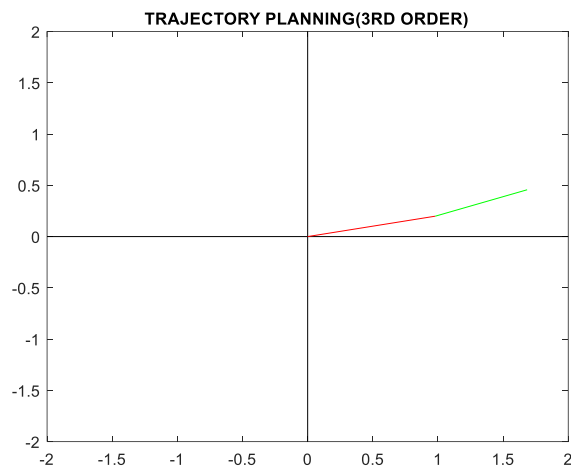


Fig 10. MATLAB Animation of 2R planar manipulator with cubic polynomial joint variables

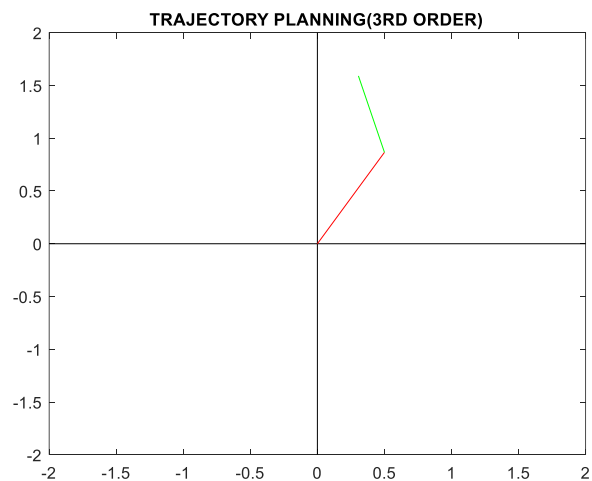


Fig 11. MATLAB Animation of 2R planar manipulator with cubic polynomial joint variables (Final position)

Both the links finally reach their positions smoothly. Here the joint space scheme was utilized, so it is solved directly by considering only the joint angles but if in case the cartesian scheme is used, θ_1, θ_2 is found out by applying inverse kinematics for initial and final coordinates and then the same procedure shall be applied.

In Inverse Kinematics joint angles of all joints in the arm are determined from the position and orientation [8]. In reality, the inverse kinematic equations are more important because the controller will calculate the joint values using the equations 11 or 12, 18 or 19 and makes the robot to reach the desired position with the specified orientation.

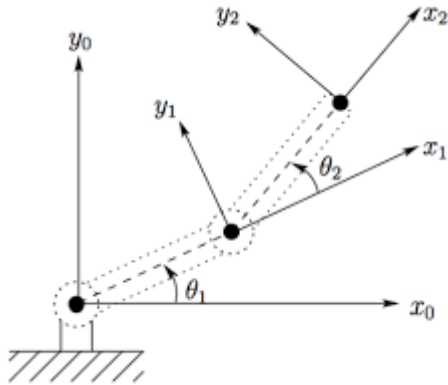


Fig 12. 2R Manipulator (Inverse Kinematics)

The end effector position is known from the forward kinematics. Now, with the use of these end effector coordinates, the expression for inverse kinematics is obtained.

The coordinates of the end effector are:

$$P_x = l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \dots\dots\dots (6)$$

$$P_y = l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \dots\dots\dots (7)$$

Squaring and adding the above two equations will lead to:

$$\cos \theta_2 = \frac{P_x^2 + P_y^2 - l_2^2 - l_1^2}{2l_1 l_2} \dots\dots\dots (8)$$

From $\cos \theta_2$ the value of $\sin \theta_2$ can be determined,

$$\sin \theta_2 = \sqrt{1 - (\cos \theta_2)^2} \dots\dots\dots (9)$$

$$\sin \theta_2 = -\sqrt{1 - (\cos \theta_2)^2} \dots\dots\dots (10)$$

Now find θ_2 ,

$$\theta_{2,I} = \text{atan2}(\sin \theta_2, \cos \theta_2) \dots\dots\dots (11)$$

$$\theta_{2,II} = \text{atan2}(-\sin \theta_2, \cos \theta_2) \dots\dots\dots (12)$$

The 2 values that is obtained for θ_2 corresponds to the elbow up and elbow down position.

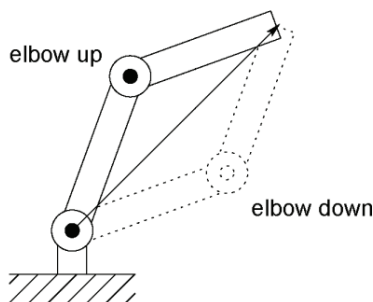


Fig 13. Elbow up and Elbow down orientation

To find θ_1 ,

$$P_x(l_2 \sin \theta_2) = (l_2 \sin \theta_2)(l_2 \cos(\theta_2) + l_1) \cos \theta_1 - (l_2 \sin \theta_2)^2 \sin \theta_1 \dots\dots\dots (13)$$

$$P_y(l_2 \cos(\theta_2) + l_1) = (l_2 \sin \theta_2)(l_2 \cos(\theta_2) + l_1) \cos \theta_1 + (l_2 \cos(\theta_2) + l_1)^2 \sin \theta_1 \dots\dots\dots (14)$$

Solving these equation's, we obtain $\sin \theta_1$,

$$\sin \theta_1 = \frac{P_y(l_2 \cos(\theta_2) + l_1) - P_x(l_2 \sin \theta_2)}{(l_2 \cos(\theta_2) + l_1)^2 + (l_2 \sin \theta_2)^2} \dots\dots\dots$$

$$P_x(l_2 \cos(\theta_2) + l_1) = (l_2 \cos(\theta_2) + l_1)^2 \cos \theta_1 - (l_2 \cos(\theta_2) + l_1)(l_2 \sin \theta_2) \sin \theta_1 \dots\dots\dots (15)$$

$$P_y(l_2 \sin \theta_2) = (l_2 \sin \theta_2)^2 \cos \theta_1 + (l_2 \cos(\theta_2) + l_1)(l_2 \sin \theta_2) \sin \theta_1 \dots\dots\dots (16)$$

Solving these equation's, we obtain $\cos \theta_1$,

$$\cos \theta_1 = \frac{P_x(l_2 \cos(\theta_2) + l_1) + P_y(l_2 \sin \theta_2)}{(l_2 \cos(\theta_2) + l_1)^2 + (l_2 \sin \theta_2)^2} \dots\dots\dots (17)$$

The two values of θ_1 are obtained for both elbow down and elbow up position by substituting the corresponding θ_2 values.

Now find θ_1 ,

$$\theta_{1,I} = \text{atan2}(\sin \theta_{1,I}, \cos \theta_{1,I}) \dots\dots\dots (18)$$

$$\theta_{1,II} = \text{atan2}(-\sin \theta_{1,II}, \cos \theta_{1,II}) \dots\dots\dots (19)$$

Using these equations, find the joint variables for the given coordinates.

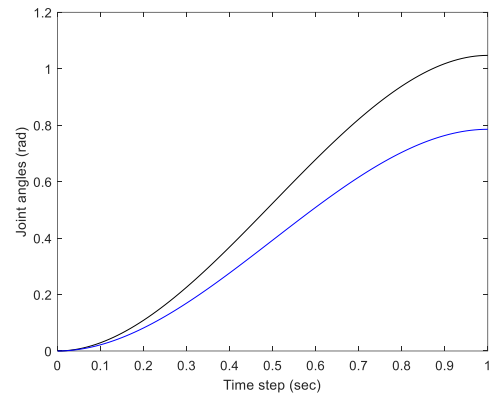


Fig 14. Plot in MATLAB between Joint angles and time

The graph output between θ_1 and θ_2 against time is plotted. It can be inferred from the graph that the joint angle for both the links slowly starts increasing then gradually speeds up and finally slows down smoothly.

V. CONCLUSION

During runtime the path generator calculates the position, velocity and acceleration and feeds into the control system of the robot. For each instant of time, the joint angle is calculated by the controller utilizing the cubic polynomial. The nature of cubic polynomial helps in achieving a smooth motion for the end effector to move from one position to another. In factories around the world, robots perform a lot of manipulation on a daily basis. They can lift massive objects, move with high speed, and repeat difficult tasks with unerring precision. Yet outside of the controlled environment, even the most advanced robot would be unable to get a glass of water. Robot manipulation in human environments is a young research area, which is expected to grow rapidly in the coming years as more researchers seek to create robots that actively help us in our daily lives.

VI. REFERENCES

- [1] L-W. Tsai., "Robot Analysis: The Mechanics of Serial and Parallel Manipulators", John Wiley & Sons, Inc. NY, 1999
- [2] Reza N. Jazar, "Theory of Applied Robotics: Kinematics, Dynamics and Control", Springer, Second Edition, 2010

- [3] Sharath Surati¹, Shaunak Hedao¹, Tushar Rotti¹, Vaibhav Ahuja¹, Nishigandha Patel², Pick and Place Robotic Arm: A Review Paper, IJRET, Vol. 08, 2022
- [4] Thu Zar, Theingi, Su Yin Win, "Experimental Results on Circular and Linear Movements of Two-link Robot Arm", IJSRST, Vol. 4, 2018
- [5] Mahmoud Gouasmi, Mohammed Ouali, Brahim Fernini¹ and M'hamed Meghatria., "Kinematic Modelling and Simulation of a 2-R Robot Using SolidWorks and Verification by MATLAB/Simulink", SAGE journals, 2012
- [6] V G Pratheep¹, M Chinnathambi², E B Priyanka³, P Ponnuragan⁴, Pridhar Thiagarajan⁵, "Design and Analysis of six DOF Robotic Manipulator", IOP Conference Series, 2021
- [7] Jie Cai¹, Jinlian Deng², Wei Zhang¹, Weisheng Zhao³, "Modeling Method of Autonomous Robot Manipulator Based on D-H Algorithm", Hindawi, 2021
- [8] Jignesh Vinod Patel, "Robotic Manipulator (Robotic Arm)", IJERT, Vol. 10, 2021
- [9] Jinda Cui and Jeff Trinkle, "Toward next-generation learned robot manipulation", Science Robotics, 2021
- [10] Ksenia I. Goryanina*, Aleksandr D. Lukyanov and Oleg I. Katin, "Review of robotic manipulators and identification of the main problems", MATEC Web of Conferences, 2018