Journal of Physics:
Conference Series

PURPOSE-LED PUBLISHING™

PAPER • **OPEN ACCESS**

Trajectory planning of a 5 DOF feeding serial manipulator using 6th order polynomial method

To cite this article: P A Parikh *et al* 2021 *J. Phys.: Conf. Ser.* **1921** 012088

View the article online for updates and enhancements.

You may also like

- Machine Learning Benchmarks for the Classification of Equivalent Circuit Models from Electrochemical Impedance Spectra
Joachim Schaeffer, Paul Gasper, Esteban Garcia-Tamayo et al.
- Performance of tensor decomposition-based modal identification under nonstationary vibration
P Friesen and A Sadhu
- Probing-error compensation using 5 degree of freedom force/moment sensor for coordinate measuring machine
Minho Lee and Nahm-Gyoo Cho



This content was downloaded from IP address 103.233.171.39 on 06/01/2025 at 12:14

Trajectory planning of a 5 DOF feeding serial manipulator using 6th order polynomial method

P A Parikh[1], R R Trivedi[2] and K D Joshi[3]

[1]*Department of Design, Nirma University, Ahmedabad, Gujatat, India*
[2]*School of Mechanical Engineering, Institute of Technology, Nirma University, Gujarat, India*
[3]*School of Engineering and Applied Sciences, Ahmedabad University, Gujarat, India*

E-mail: *[1]priyam.parikh@nirmauni.ac.in, [2]reena.trivedi@nirmauni.ac.in*

Abstract: In the present era the importance of household and assistive robots has been increasing. Usually assistive robots are serial manipulators having degree of freedom more than three. In this paper a case of vertically downward 5 DOF assistive robots is taken, which helps physically challenged people in feeding. This paper delivers the methodology to build the smooth trajectory for multi-degree of freedom robot using higher degree polynomials method. An optimal trajectory decreases jerk to the end-effector (As end-effector carries food) protects battery life, protects joints from the mechanical vibrations and reduces time to reach at the user. This paper also discusses the forward and inverse kinematic analysis of a T-R-R-R-R type robot, as it is highly recommended before trajectory planning. The paper also compares the trajectories generated by inverse kinematics (Cartesian Scheme) and Joint space scheme. In a nut shell this methodology will be helpful in designing the trajectories of assistive T-R-R-R-R type robots for physically challenged people like Parkinson patients, patients with vertigo or vestibular disorder, neurological disorder and many more.

Keywords: *Assistive robots, trajectory planning, higher order polynomials, physically challenged people, Forward Kinematics and Inverse kinematic analysis, vertically downward robot.*

1. INTRODUCTION

Nowadays human activities are replaced by robots, ranging from simple robot to complex robot [1]. Usually these serial manipulators are useful in feeding, painting, loading/ unloading, welding, assembling and dissembling. Few years back serial manipulators were only used in industries but nowadays, they are being used in house hold applications as well [2]. In the proposed research work, a case of 5 DOF feeding serial manipulator is shown. This robot is vertically downward and the type of the robot is T-R-R-R-R. It is used to feed semi liquid food to the patients suffering from various neurological diseases. House hold robots are smaller in size, due to which it's stability and accuracy matters a lot. Therefore trajectory planning becomes extremely important (As it is associated with feeding applications). In the current paper, trajectory planning of a T-R-R-R-R type 5 DOF robot is shown using 6th order polynomial method. Trajectory planning can be done via lower order polynomials, cycloid functions, exponential functions and parabolic methods. Lower order polynomials gives discontinuities in the joint rates, due to which motors might get damaged or burn. In addition with that, lower order polynomials gives instantaneous velocity and infinite acceleration, which gives immense amount vibrations in the middle of trajectory [4]. This phenomenon will lead robot towards wasting of food and might harm the user as well. To overcome

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

this problem, the trajectory planning is done using higher order polynomials (6^{th} order polynomial). However higher order polynomials also give shoots of acceleration in the middle of the trajectory, but gives zero velocity and acceleration at the boundary points [5]. Fourth and Fifth order polynomials give smooth trajectory planning but, does not give any information about jerk. Hence the trajectory is planned using 6^{th} order polynomial [6]. The first phase of the paper gives the primary information of the robot like 3D CAD model, hardware setup, simulation block diagram, problem description and methodology. In the second phase, Forward and Inverse kinematics is shown. In the third phase of the paper trajectory planning is discussed using inverse kinematics, cubic polynomial as well as 5^{th} and 6^{th} order polynomial. However the existing paper only talks about trajectory planning; motion planning and singularity checking is not covered. In the final phase of the paper all the trajectories are compared and concluded along with user testing and user experience. The hardware of the system is shown in figure 1 and 3D model along with its kinematic chain diagram is shown in figure 2. The basic details of the robot are shown in table 1. In the proposed system, multiple via points are not taken as it completes the task from home position and the delivery point.

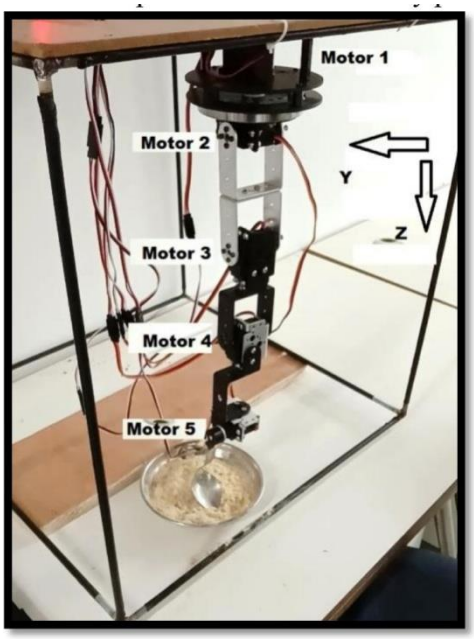


Figure 1. Hardware setup of the system

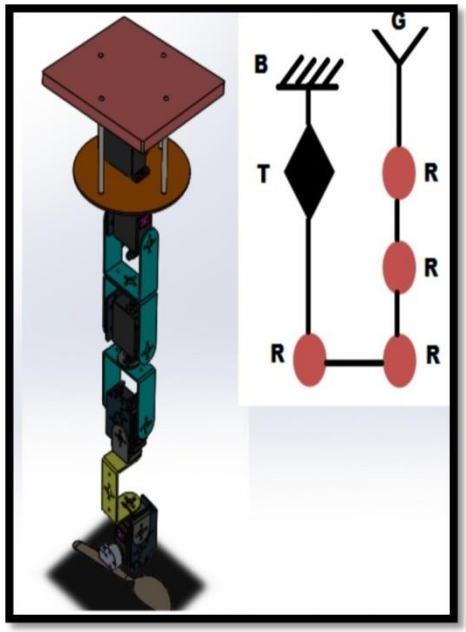


Figure 2. 3D CAD model with Kinematic Chain

Gallant, A. et al. 2018 worked on "Extending the capabilities of robotic manipulators using trajectory optimization"[7]. Gasparetto at al. 2010 did "Optimal trajectory planning for industrial robots" [8]. Kucuk S. at el. 2017 did "Optimal trajectory generation algorithm for serial and parallel manipulators" [9]. Kumar P. at el. 2017 did Workspace Optimization of 3PRR Parallel manipulator for drilling operation using Genetic Algorithm [10]. L. Shaoming at el. 2017 worked on "Research on trajectory tracking control of multiple degree of freedom manipulators," [11]. Li at el. 2018 published a paper on "An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using Quantic B-splines" [12]. Liu et al. 2018 did Trajectory Planning with Minimum Synthesis Error for Industrial Robots Using Screw Theory [13]. Macfarlane 2003 performed Jerk-bounded manipulator trajectory planning: design for real-time applications [14]. Rossi, C. et al. 2013 worked on Robot trajectory planning by assigning positions and tangential velocities. [15]. Parikh P.A. et al. 2021 did vision trajectory planning and planned trajectory using LSPB, cycloid and semi cycloid trajectory [16]. Valente, A. et. al. 2017 published a paper called Smooth trajectory generation for industrial robots performing high precision assembly processes [17]. Wang, H. 2019 wrote a paper on Smooth point-to- point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve [18]. Wang, X. et al. 2018 did Singularity analysis and treatment for

a 7R 6-DOF painting robot with non-spherical wrist [19]. Xiaojie Zhao 2016 did Trajectory planning for 6-DOF robotic arm based on Quantic polynomial [20]. Yishen Guan at el. 2005 worked on robotic trajectory planning using polynomial interpolations [21]. Zhang, J. at el. 2018 worked on a 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm [22]. Zhang S at el. 2020 published a paper on Trajectory Planning Based on Non-Convex Global Optimization for Serial Manipulators [23]. Priyam A. Parikh et al. 2020 published a paper on trajectory planning using 7^{th} and 9th order polynomials [24].

Table 1. Details of the Robot

Sr. No	Details of Robot used in this paper	
	Particular	**Details Remarks**
1	Degrees of Freedom	5
2	Type of Robot	T-R-R-R-R
3	Servo Motors X 5	Metal Gears, 15kg/cm torque (Stall torque)
4	Working Speed	0.13 sec/ 60 degree at 7.2 volts (No load)
5	Working Voltage	4.8 volt-7.2 volts
6	Controller Used	ARDUINO MEGA
7	Battery	LI-PO 7.2 volt
8	Sensor used	Gyroscope for feedback

2. PROBLEM DESCRIPTION AND METHODOLOGY

The trajectory planning is done in joint-space scheme (not in Cartesian scheme) to avoid workspace issues and simplicity in the microcontroller programming. The objective of the system is to deliver food to the patient without wasting it and without harming the patient. Lower order polynomials does not satisfy zero boundary conditions also gives massive amount of jerks to the links and to the servo motors. Higher order polynomials satisfies zero boundary conditions but not include third derivative of the angular displacement, therefore trajectory is planned using 6th order polynomials. However trajectory can be planned using 7th and 9th order polynomials also, but they give a lot of variations in the joint angles, which might lead robot toward wrong end point. In a nut shell, there are mainly three technical objectives. 1) Zero velocity and accelerations at the boundaries 2) Inclusion of the Third derivative of the angular displacement and 3) Elimination of instantaneous angular velocity and providing finite angular acceleration. Forward kinematics is applied to the get the end-point orientation, joint angles and link length. Inverse kinematics is also performed to get refined joint angles and removing internal singularities. However workspace and singularities are not discussed in this paper. For 3D modelling, SOLIDWORKS software is used and for simulation, Peter Croke Robotic tool box along with Simscape (MATLAB) is used. For forward kinematics and trajectory planning simulation, Roboanalyzer software is used. The robot is vertically downward and lies in the fourth quadrant. Here XZ plane is more important for visualization of the DH matrix.

3. FORWARD KINEMATICS

As discussed earlier the robot manipulator mainly contains joints, links and servo actuators. The links connected with joints makes open kinematic chain. The maximum position in a volume, where spoon can reach, called a workspace [26]. There are mainly two kinds of analysis in kinematics; forward kinematics and inverse kinematics. Forward kinematics is for determining the position and an orientation of the robot end effector with respect to the reference coordinate system. In this case joint

parameters and arm parameters are already defined [27]. Before getting into the inverse kinematics, forward kinematics is to be performed and understood using DH matrix frame assignment shown in figure 4. DH frame assignment for the proposed serial manipulator is shown in figure 5. Fig.4 shows the DH matrix frame assignment, where the methodology is explained to find end point of the end- effector using known θ (Joint angle) α (twisting angle) a (Link length) and d (joint distance).

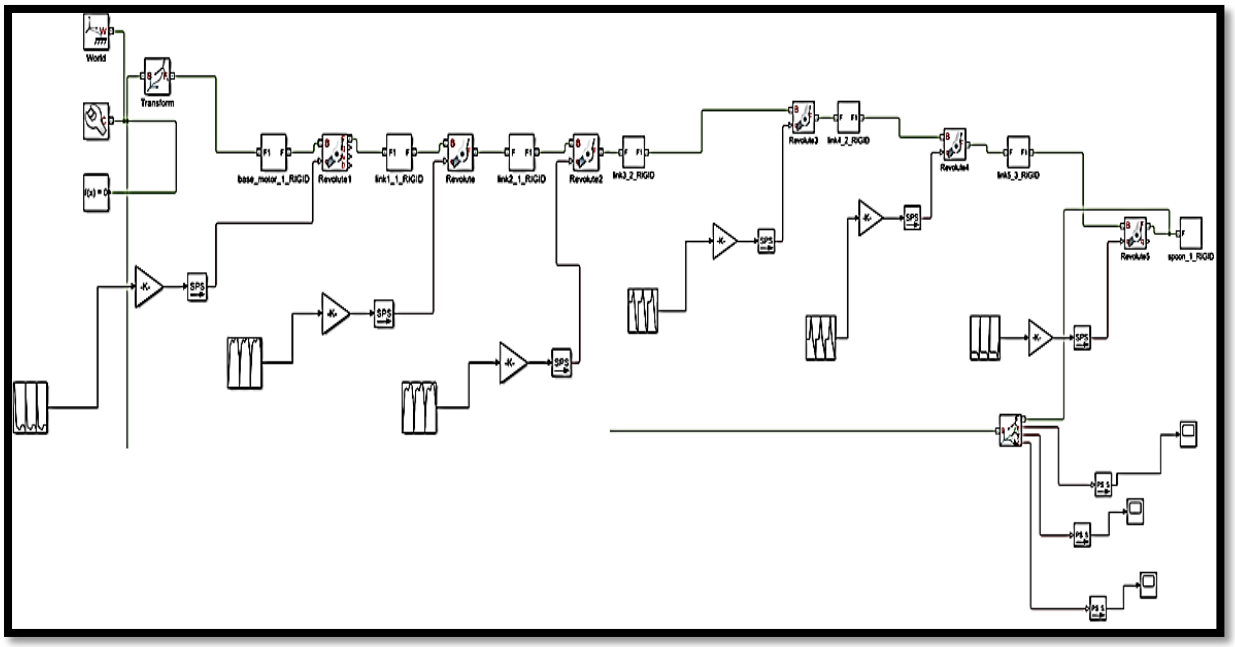


Figure 3. Kinematic Simulation

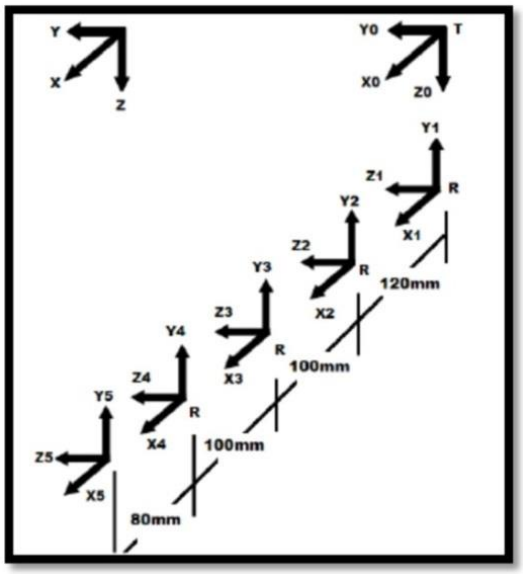


Figure 4. DH Matrix frame assignments

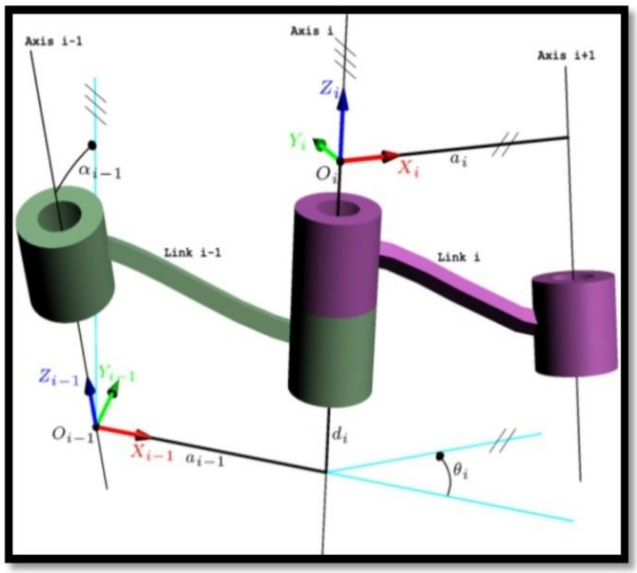


Figure 5. DH Matrix for the feeding robot

In equation 1, F is the function of angular displacement; by putting the value of angular displacement, values, of desired position and rotation can be found. θ_1, θ_2, θ_3, θ_4, and θ_5 are the input variables and x, y, z and R are the desired position and rotation respectively. The robot arm parameters are shown in Table 2, which are generated using DH matrix frame assignment shown in fig. 5. The forward kinematics assignment helps in finding the transformation of end-effector with respect to origin, which is shown in (2). Matrix shown in (3) is a transformation matrix, which is a

combination of rotation matrix (3X3) and orientation (3X1) matrix. Equations 4 to 8 are derived by DH matrix frame assignment along with screw principle of rotation and translation. Equation 9 is a reference matrix to find all transformation matrices of the consecutive links.

$$F(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = [x, y, z, R] \quad (1)$$

Table 2. Robot Joint Parameters: Forward Kinematics

θ (Joint angle) Initial in Radian	θ (Joint angle) Final in Radian	d (joint distance) mm	α (twisting angle) Radian	a (Link length) mm
θ1=π	θ1=π	50	–π/2	0
θ2=π/2	θ1=2π/3	0	0	140 (L1)
θ3=0	θ1=2π/9	0	0	120 (L2)
θ4=0	θ1=π/9	0	0	100(L3)
θ5= –π/2	θ1= –π/18	0	0	80(L4)

$${}^0_5T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^{03}_4T\,{}^4_5T \quad (2)$$

$${}^0_5T = \begin{bmatrix} v_{11} & v_{12} & v_{13} & P_x \\ v_{21} & v_{22} & v_{23} & P_y \\ v_{31} & v_{32} & v_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^0_1T = \text{Rot}(\hat{Z}, \theta_1)\text{Rot}(\hat{X}, -90) \quad (4)$$

$${}^1_2T = \text{Rot}(\hat{Z}, \theta_2)\text{Trans}(\hat{X}, L_1) \quad (5)$$

$${}^2_3T = \text{Rot}(\hat{Z}, \theta_3)\text{Trans}(\hat{X}, L_2) \quad (6)$$

$${}^3_4T = \text{Rot}(\hat{Z}, \theta_4)\text{Trans}(\hat{X}, L_3) \quad (7)$$

$${}^4_5T = \text{Rot}(\hat{Z}, \theta_5)\ \text{Trans}(\hat{X}, L_4) \quad (8)$$

Based on (9), all the transformation matrices are calculated (shown in (10) to (14)). Referring Table 2, feeding the initial joint angles in (9), we get the initial transformation matrix of the robot, which is shown in (15). In the similar way, feeding final angles in (9), we get the final position the robot in the form of transformation matrix, shown in (16). Note that (Px,Py,Pz) is the position vector, whereas v11 to v33 shows the rotational vector.

$$ {}^{i-1}_{i}T = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9) $$

$$ {}^{0}_{5}T = \begin{bmatrix} -1 & 0 & 0 & -130 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & -320 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10) $$

Similarly putting all the final values of each joint in (column 2) (9), we get the final transformation matrix, shown in (11). It should be noted that (P_x, P_y, P_z) is the positional vector, whereas v11 to v33 is the rotational vector.

$$ {}^{0}_{5}T = \begin{bmatrix} 0.9848 & 0.1736 & 0 & 332.8 \\ 0 & 0 & -1 & 0 \\ -0.1736 & 0.9848 & 0 & -102 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11) $$

4. INVERSE KINEMATICS

Inverse kinematics of a robot manipulator deals with the calculation of each joint variable, given the position and orientation of end effector. The computation of the inverse kinematics of robot manipulator is quite difficult if compared to the forward kinematics because of the nonlinearities and multiple solutions involved [9]. After getting end points (Px, Py and Pz) using forward kinematics, inverse kinematics helps in finding $\theta 1$ to $\theta 5$ respectively. According to inverse kinematics F is the function of x,y,z and R.

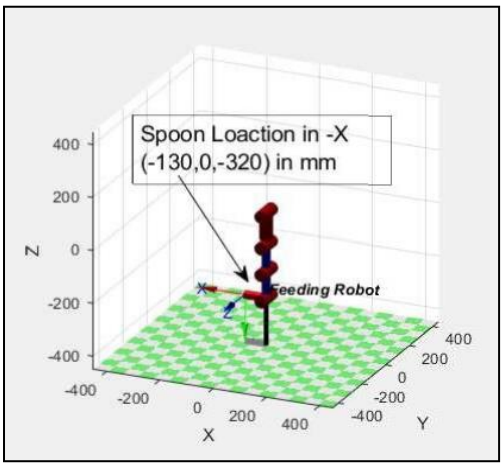


Figure 6. Initial and Final Position of the Robot in XYZ plane before performing DH frame Assignments

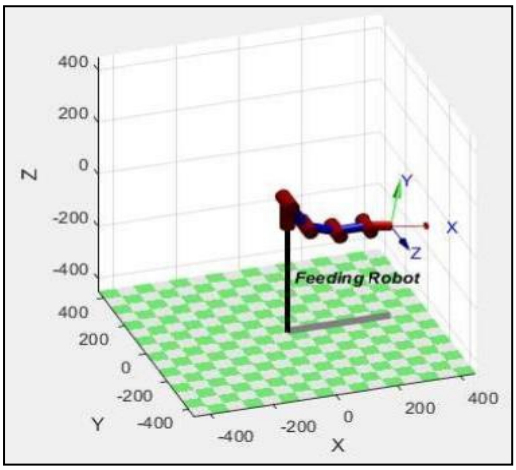


Figure 7. Initial and Final Position of the Robot in XZ plane after performing forward kinematics

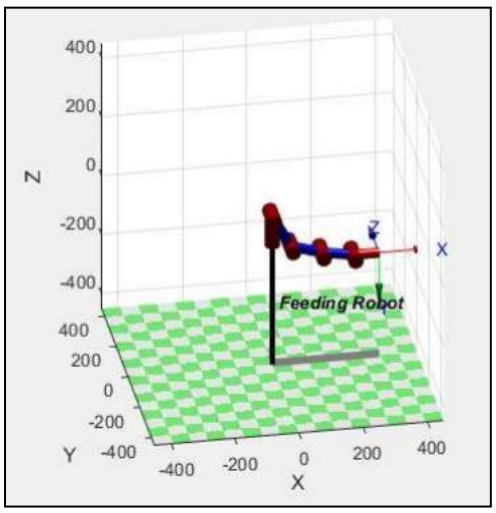


Figure 8. Initial and Final Position of the Robot in XZ plane after performing inverse kinematics

All the joint angles are found based on the known transformation matrix shown in (3). All the unknown joint angles are found using (16) to (29). These joint angles are shown in Table 3 (Both Initial and Final angles). Figure 6 and 7 shows the initial and final position of the robot as per forward kinematics, which is simulated in MATALAB. After applying inverse kinematic technique, the final
position of the robot is shown in fig.8. It should be noted that, $C_{2345} = \cos(\theta_2 + \theta_3 + \theta_4 + \theta_5)$ and $S_{234} = \sin(\theta_2 + \theta_3 + \theta_4)$.

$$F(x, y, z, R) = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5] \tag{12}$$

$$\left[{}^{0}_{1}T\right]^{-1}{}^{0}_{5}T = {}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T \tag{13}$$

$$P_X = C_1(L_1C_2 + L_2C_{23} + L_3C_{234} + L_4C_{2345}) \tag{14}$$

$$P_y = S_1(L_1C_2 + L_2C_{23} + L_3C_{234} + L_4C_{2345}) \tag{15}$$

$$P_z = 50 - (L_1S_2 - L_2S_{23} - L_3S_{234} - L_4S_{2345}) \tag{16}$$

Table 3. Robot Joint Parameters: Inverse Kinematics

Joints	θ_i (Joint angle) Initial in Radian	θ_f (Joint angle) Final in Radian	θ_i (Joint angle) Initial in Degree	θ_f (Joint angle) Final in Degree
Joint 1	$-12\pi/360$	$\pi/360$	–12.5	0
Joint 2	$120\pi/180$	$76\pi/180$	120	76
Joint 3	$\pi/360$	$-69\pi/180$	0	–69
Joint 4	$\pi/360$	$-6\pi/180$	0	–6
Joint 5	$-101\pi/180$	$\pi/360$	–101	0

5. TRAJECTORY PLANNING

As discussed earlier, robot trajectory planning is to ensure the smooth variation in the robotic joints. Trajectory planning also gives time history of position, velocity and acceleration at the intermediate point as well as final and starting point [25]. Following methods are the conventional methods to design a trajectory. It can be done using Cartesian scheme as well as joint space scheme. Here joint space scheme is used, since Cartesian is more complex and already covered in inverse kinematics [26].

- Cubic Polynomial
- Fifth order polynomial
- Liner trajectory function
- Higher order polynomials (Greater than 3rd Order)

If there is cubic polynomial method, there will be three coefficients, whereas in 5th and 6th order polynomials there will be five and six coefficients respectively. To deploy these methods all the initial conditions (i.e. time, angle, angular velocity and angular acceleration) must be known [27]

In the present case all the initial and final values of all the joints are known. Apart from that angular velocity at the beginning and at the end is kept zero [25]. θ is the angle in degree whereas t_i and t_f are the initial and final time respectively. θ_i and θ_f are the initial and final angles of the joints. All the angles are in degree and θ_i, $\dot{\theta}_i$ $\ddot{\theta}_i$, $\dddot{\theta}_i$ and $\frac{d\,\ddot{\theta}_i}{dt}$ are the initial angular displacement, angular velocity, angular acceleration and jerk respectively, whereas θ_f, $\dot{\theta}_f$, $\ddot{\theta}_f$, $\dddot{\theta}_f$ and $\frac{d\,\ddot{\theta}_f}{dt}$ are the final angular displacement, angular velocity, angular acceleration and jerk respectively [26]

5.1 Trajectory Planning using cubic Polynomial

The third order polynomial equation along with its first and second derivative is shown in equation (17), (18) and (19) respectively, where a, b, c and d are the coefficients (related to angular displacement, angular velocity, angular acceleration and jerk respectively). Initial conditions are mentioned in Table 4 for all the joints.

$$\theta(t) = a + bt + ct^2 + dt^3 \quad (17)$$

$$\dot{\theta}(t) = b + 2ct + 3dt^2 \quad (18)$$

$$\ddot{\theta}(t) = 2c + 6dt^1 \quad (19)$$

Using equation (30) to equation (32) and considering the initial conditions we get,

$a = \theta_i$

$b = 0$

$c = 3\frac{(\theta_f - \theta_i)}{t_f^2}$

$d = 2\frac{(\theta_f - \theta_i)}{t_f^3}$

$$\theta(t) = \theta_i + 3\frac{(\theta_f - \theta_i)}{t_f^4}t^4 + 2\frac{(\theta_i - \theta_f)}{t_f^5}t^5 \quad (20)$$

Putting values of all coefficients in equation (17) and solving it we get equation (20) as a generalized form of cubic polynomial. Equation (21) shows the angular displacement for all the joints, after putting all the values of coefficients given in Table 4.

$$\theta(t) = \begin{cases} -21 + 1.75t^2 - 0.2t^3, & For\ joint\ 1 \\ 119 - 4.83t^2 + 0.537t^3, & For\ joint\ 2 \\ -3.583t^2 + 0.398t^3, & For\ joint\ 3 \\ -0.583t^2 + 0.0648t^3, & For\ joint\ 4 \\ -87 + 8.083t^2 - 0.0898t^3, & For\ joint\ 5 \end{cases} \quad (21)$$

5.2 Trajectory Planning using cubic 5th Order polynomial

Just like cubic polynomial method, trajectory planning can be done using 5th order polynomial also. The fifth order polynomial equation along with its first and second derivative is shown in equation (22), (23) and (24) respectively. Initial conditions are mentioned in Table 5 for all the joints.

$$\theta(t) = a + bt + ct^2 + dt^3 + et^4 + ft^5 \quad (22)$$

$$\dot{\theta}(t) = b + 2ct + 3dt^2 + 4et^3 + 5ft^4 \quad (23)$$

$$\ddot{\theta}(t) = 2c + 6dt^1 + 12et^2 + 20ft^3 \quad (24)$$

$$\dddot{\theta}(t) = 6d + 24et + 60ft^2 \quad (25)$$

Using equation (22) to (25) and considering the initial conditions we get,

$$a = \theta_i$$
$$b = \dot{\theta}_i$$
$$c = \frac{\ddot{\theta}_i}{2}$$
$$d = -10\frac{(\theta_f - \theta_i)}{t_f^3}$$
$$e = 15\frac{(\theta_f - \theta_i)}{t_f^4}$$
$$f = \frac{(\theta_f - \theta_i)}{}$$

Putting values of all coefficients in equation (22) and solving it we get equation (26) as a generalized form of 5th order polynomial. Equation (27) shows the angular displacement for all the joints, after putting all the values of coefficients given in Table 5.

$$\theta(t) = \theta_i + \dot{\theta_i}t + \frac{\ddot{\theta_i}}{2}t^2 - 10\frac{(\theta_f - \theta_i)}{t_f^3}t^3 - 15\frac{(\theta_i - \theta_f)}{t_f^4}t^4 - 6\frac{(\theta_f - \theta_i)}{t_f^5}t^5 \quad (26)$$

$$\theta(t) = \begin{cases} -21 + 0.6122t^3 - 0.13119t^4 + 0.0074t^5, & For\ joint\ 1 \\ 119 - 1.69t^3 - 0.3622t^4 - 0.020t^5, & For\ joint\ 2 \\ -1.253t^3 + 0268t^4 - 0.016t^5, & For\ joint\ 3 \\ -0.2040t^3 + 0.04372t^4 - 0.0024t^5, & For\ joint\ 4 \\ -87 + 2.8275t^3 - 0.6056t^4 + 0.0342t^5, & For\ joint\ 5 \end{cases}$$

5.3 Trajectory planning using 6th order Polynomial

Just like cubic and 5th order polynomial methods, we have angular displacement in the form of 6th order polynomial, shown in equation (28). Putting values of all coefficients in as per equation (32) to (35), and solving it as AX=B, we get equation (41) as a generalized form of 6th order polynomial. Equation 42 shows the angular displacement for all the joints, after putting all the values of coefficients given in Table 6.

$$\theta(t) = a + bt + ct^2 + dt^3 + et^4 + ft^5 + gt^6 \quad (28)$$

$$\theta(\dot{t}) = b + 2ct + 3dt^2 + 4et^3 + 5ft^4 + 6gt^5 \quad (29)$$

$$\theta(\ddot{t}) = 2c + 6dt^1 + 12et^2 + 20ft^3 + 30gt^4 \quad (30)$$

$$\theta(t) = \begin{cases} \theta_i = a, & t = 0\ sec \\ \theta_f = a + bt_f + ct_f^2 + dt_f^3 + et_f^4 + ft_f^5 + gt_f^6 & t = t_f\ sec \end{cases} \tag{32}$$

$$\dot{\theta}(t) = \begin{cases} \dot{\theta}_i = b, & t = 0\ sec \\ \dot{\theta}_f = b + 2ct_f + 3dt_f^2 + 4et_f^3 + 5ft_f^4 + 6gt_f^5 & t = t_f\ sec \end{cases} \tag{33}$$

$$\ddot{\theta}(t) = \begin{cases} \ddot{\theta}_i = 2c, & t = 0\ sec \\ \ddot{\theta}_f = 2c + 6dt_f + 12et_f^2 + 20ft_f^3 + 30gt_f^4 & t = t_f\ sec \end{cases} \tag{34}$$

$$\dddot{\theta}(t) = \begin{cases} \dddot{\theta}_i = 6d, & t = 0\ sec \\ \dddot{\theta}_f = 6d + 24et_f + 60ft_f^2 + 120gt_f^3, & t = t_f\ sec \end{cases} \tag{35}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 & t_f^6 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 & 6t_f^5 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 & 30t_f^4 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{bmatrix} = \begin{bmatrix} \theta_i \\ \theta_f \\ \dot{\theta}_i \\ \dot{\theta}_f \\ \ddot{\theta}_i \\ \ddot{\theta}_f \\ \dddot{\theta}_i \end{bmatrix} \tag{36}$$

$$a = \theta_i \quad b = \dot{\theta}_i \quad c = \ddot{\theta}_i/2 \quad d = \dddot{\theta}_i/6 \tag{37}$$

$$e = 15\frac{(\theta_f - \theta_i)}{t_f^4} - \frac{5}{t_f^3}(3\dot{\theta}_f + 4\dot{\theta}_i) + 5\frac{(0.5\ddot{\theta}_f - \ddot{\theta}_i)}{t_f^2} - \frac{\dddot{\theta}_i}{6t_f} \tag{38}$$

$$f = 24\frac{(\theta_i - \theta_f)}{t_f^5} + \frac{1}{t_f^4}(39\dot{\theta}_f + 45\dot{\theta}_i) + \frac{(10\ddot{\theta}_i - 7\ddot{\theta}_f)}{t_f^3} + \frac{\dddot{\theta}_i + 0.5\dddot{\theta}_f}{2t_f^2} \tag{39}$$

$$g = 10\frac{(\theta_f - \theta_i)}{t_f^6} - \frac{1}{t_f^5}(34\dot{\theta}_f + 36\dot{\theta}_i) + \frac{(13\ddot{\theta}_f - 15\ddot{\theta}_i)}{2t_f^4} - \frac{2\ddot{\theta}_i}{3t_f^3} - \frac{\ddot{\theta}_f}{2t_f^3} \tag{40}$$

$$\theta(t) = \theta_i + 15\frac{(\theta_f - \theta_i)}{t_f^4}t^4 + 24\frac{(\theta_i - \theta_f)}{t_f^5}t^5 + 10\frac{(\theta_f - \theta_i)}{t_f^6}t^6 \tag{41}$$

$$\theta(t) = \begin{cases} -21 + 0.176t^4 - 0.043t^5 + 0.0023t^6, & For\ joint\ 1 \\ 119 - 0.4873t^4 - 0.12t^5 - 0.0077t^6, & For\ joint\ 2 \\ -0.3612t^4 + 0.0889t^5 - 0.0056t^6, & For\ joint\ 3 \\ -0.0588t^4 + 0.0140t^5 - 0.00085t^6, & For\ joint\ 4 \\ -87 + 0.0814t^4 - 2.005t^5 + 0.01237t^6, & For\ joint\ 5 \end{cases} \quad (42)$$

Table 4. Coefficients and Initial Conditions for Cubic Polynomial (All Joints)

Coefficients and Initial Conditions	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
a	-21	119	0	0	-87
b	0	0	0	0	0
c	1.75	-4.83	-3.583	-0.583	8.083
d	-0.2	0.537	0.398	0.0648	-0.898
e	0	0	0	0	0
f	0	0	0	0	0
g	0	0	0	0	0
θ_i	-21	119	0	0	-87
θ_f	0	61	-43	-7	-10
θ_i	0	0	0	0	0
θ_f	0	0	0	0	0
θ_i	0	0	0	0	0
θ_f	0	0	0	0	0

Table 5. Coefficients and Initial Conditions For 5^{th} order Polynomial (All Joints)

Coefficients and Initial Conditions	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
a	-21	119	0	0	-87
b	0	0	0	0	0
c	0	0	0	0	0
d	0.6122	-1.69	-1.253	-0.2040	2.8275
e	-0.1312	0.3622	0.2685	0.0437	-0.6056
f	0.0074	-0.0206	-0.016	-0.0024	0.3423
g	0	0	0	0	0
θ_i	-21	119	0	0	-87
θ_f	0	61	-43	-7	-10
θ_i	0	0	0	0	0
θ_f	0	0	0	0	0
θ_i	0	0	0	0	0
θ_f	0	0	0	0	0

Table 6. Coefficients and Initial Conditions For 6^{th} order Polynomial (All Joints)

Coefficients and Initial Conditions	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
a	-21	119	0	0	-87
b	0	0	0	0	0
c	0	0	0	0	0
d	0	0	0	0	0
e	0.176	-0.4873	-0.3612	-0.0588	0.8146
f	-0.043	0.125	0.0889	0.0140	-0.2005
g	0.0027	-0.0077	-0.0056	-0.0008	0.0123
θ_i	-21	119	0	0	-87
θ_f	0	61	-43	-7	-10
θ_i	0	0	0	0	0
θ_f	0	0	0	0	0
θ_i	0	0	0	0	0
θ_f	0	0	0	0	0

6. RESULTS

The trajectories, developed by joints space scheme and Cartesian scheme are compared and shown in fig. 9 to 12 for joint 1 to joint 5 respectively. All the unknown joint angles are found using inverse kinematics, and trajectories are plotted for the same initial and end points. The overall trajectory comparison is shown in Table 7.

For the link 1, initial point is -21 degree and end point is at 0 degree (Fig.9). In this case inverse kinematics works faster, but suddenly jumps at 1.4 sec from -21 to 0 degree, which creates a massive amount of angular acceleration and jerk at this point.6th and 5th order polynomial works smoother than inverse kinematics. However there is small amount of absolute error in this case. Similarly for joint 2, inverse kinematics adds unwanted movement between 2nd and 4th second. However cubic and 6^{th} order polynomial works better in this case compared to 5^{th} order method. Similarly for joint 3, cubic polynomial works better than 5^{th} and 6^{th} order polynomial, as they contain some finite amount of absolute error. In the same case inverse kinematics gives a sudden de-acceleration to the third link, which could disturb the other links also. Observing the trajectory of the 4^{th} link, it could be noticed that, for a small amount of movement 4^{th} link accelerates and de-accelerates (In the case of inverse kinematics). However this motion can be eliminated using some modified algorithms which lead the system towards the loss of 1 degree of freedom. If we compare the overall the trajectories it can be noticed that, in terms of accuracy 6^{th} order polynomial works better than cubic and 5^{th} order polynomial methods. As far as the stability is concerned, cubic polynomial gets stabilized very quickly.

Table 7: Overall trajectory comparison

Time t (Sec)	θ Cubic Polynomial	θ 5th Order Polynomial	θ 6th Order Polynomial	θ Inverse kinematics in Degrees
1	-80	-83	-86	-88
2	-62	-72	-80	-100
3	-42	-52	-60	-43
4	-14	-25	-35	-16
5	2	-6	-11	-8
6	10	6	-11	-10
Error in %	5.55	4.44	0.27	NA

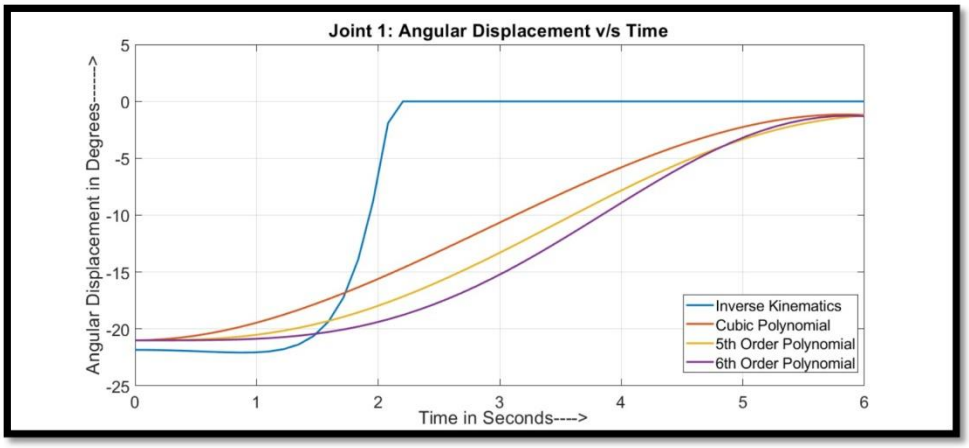


Figure 9. Trajectory Comparison of Joint 1

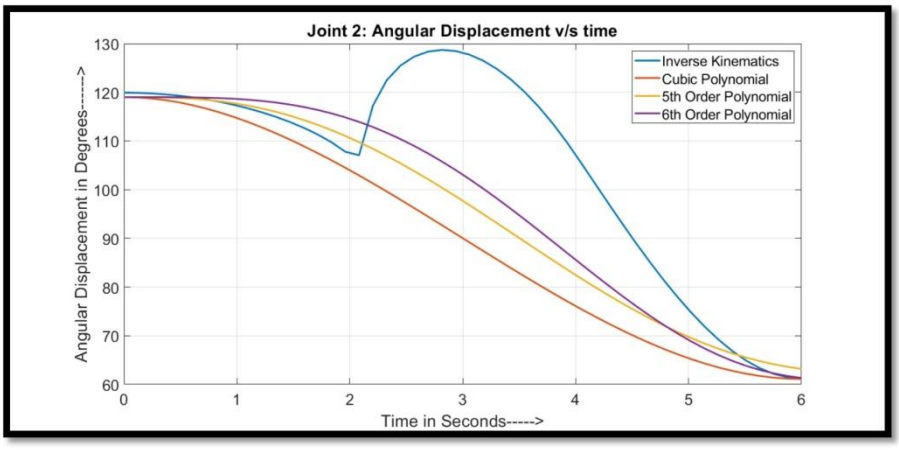


Figure 10. Trajectory Comparison of Joint 2

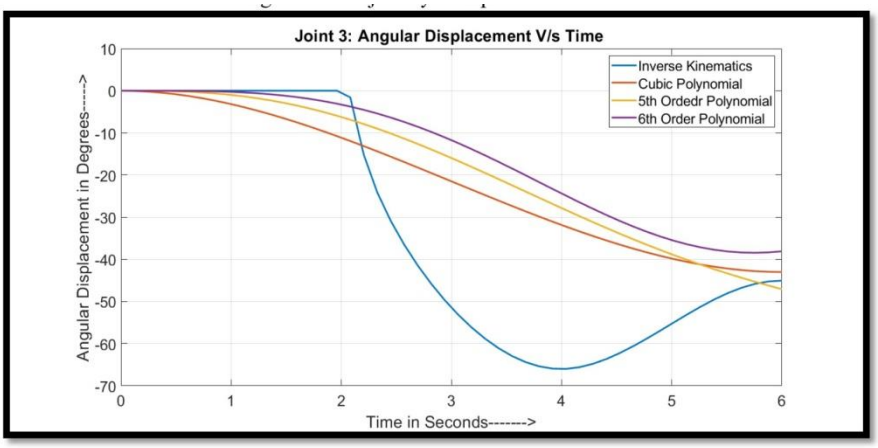


Figure 11. Trajectory Comparison of Joint 3

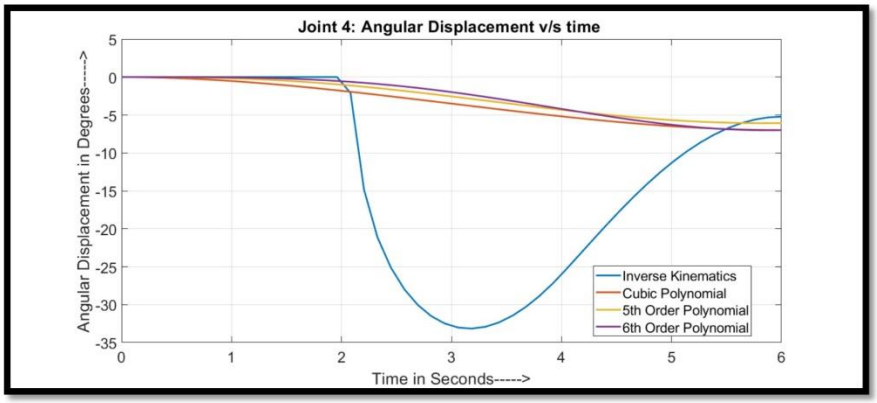


Figure 12. Trajectory Comparison of Joint 4

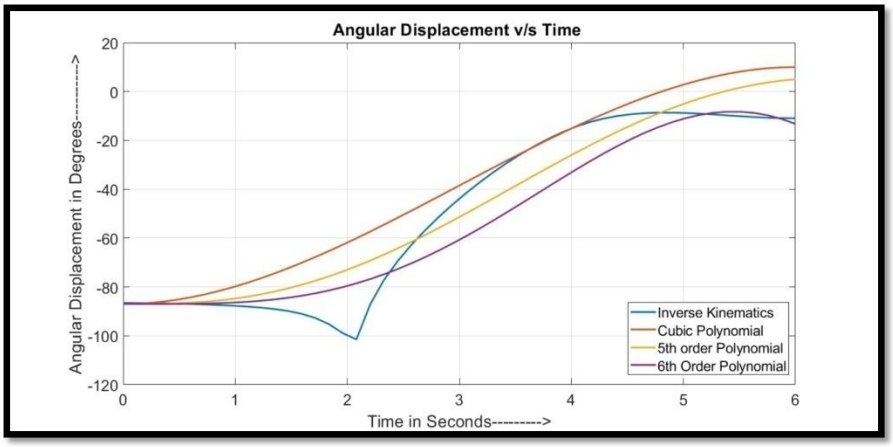


Figure 13. Trajectory Comparison of Joint 5

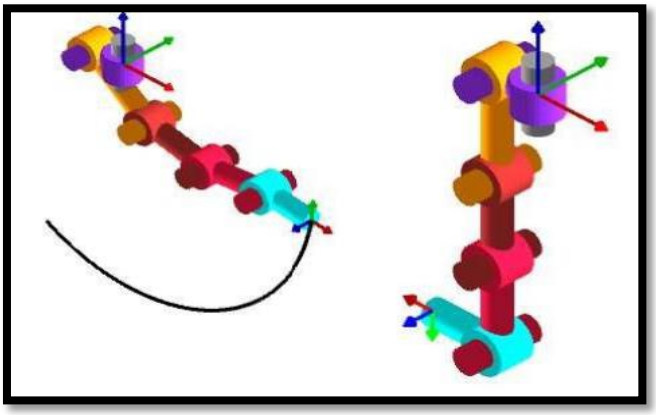


Figure 14. Final Position of the Robot in XYZ Plane after implementing 6th order polynomial method (Left side)

7. CONCLUSION

It should be noticed that compared to Cartesian scheme, joint space scheme gives smoother trajectory. Cubic polynomial works faster than 5th and 6th order polynomial methods, but accuracy is lesser. 5^{th} order trajectory gets settled quickly but leaves some amount of steady state error. 6th order trajectory creates a little amount of overshoot, but accuracy is extremely good. 6^{th} order polynomial satisfy zero angular acceleration and angular velocity conditions at the boundaries. 6^{th} order method protects joint actuators from achieving instantaneous velocity and infinite acceleration. Compared to 5^{th} order and cubic polynomials, 6^{th} order polynomial gives lesser jerks and vibration to the joint links, as a result food can be safely deliver without being wasted. In future, trajectory planning can be done using S-curve method and linear segment parabolic blend for multiple via points. Also the number of degrees of freedom can be taken six to avoid robot from singularities. The optimization algorithm can also be used to develop inverse kinematics algorithm, which can have a single solution instead of multiple solution.

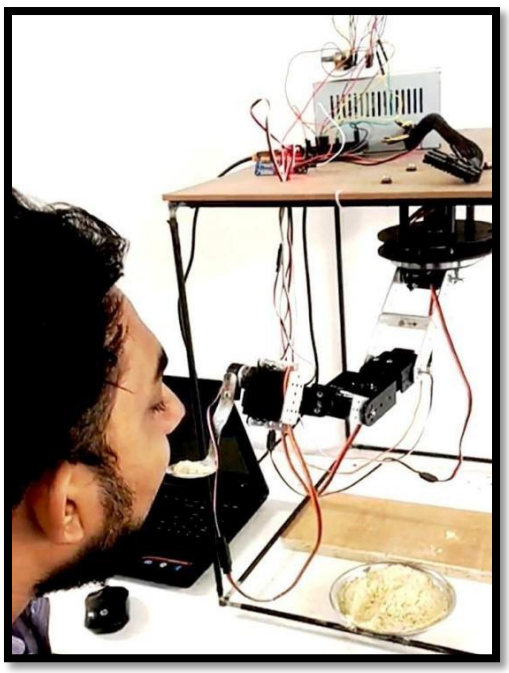


Figure 15. Final Position of the Robot in XZ plane after applying 6th order polynomial method (real time)

REFERENCES

[1] Takahashi, Y., & Hayase, M. (1996). Design of H∞ controller for precise positioning and comparison with PID control. Transactions of the Institute of Measurement and Control, 18(4), 175–182. https://doi.org/10.1177/014233129601800402
[2] Song, W.-K., & Kim, J. (2012). Novel Assistive Robot for Self-Feeding. Robotic Systems - Applications, Control and Programming. doi:10.5772/25756
[3] Dag, M., Svanelöv, E., & Gustafsson, C. (2015). Experiences of using Bestic, an eating aid for people with intellectual disabilities. *Journal of Intellectual Disabilities, 21(1), 8798*. doi:10.1177/1744629515623167
[4] Chiaverini, S., Siciliano, B., & Egeland, O. (1994). Review of the damped least- squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology, 2(2), 123–134*. doi:10.1109/87.294335
[5] Kolyubin, S., Shiriaev, A., & Jubien, A. (2018). Consistent kinematics and dynamics calibration of lightweight redundant industrial manipulators. T*he International Journal of Advanced Manufacturing Technology*. doi:10.1007/s00170-018-2868-y
[6] N. Zhang, Y. Zhang, J. Cheng and C. Ma, "Inverse kinematics solution for six-DOF serial robots based on BP neural network," *2017 Chinese Automation Congress (CAC), Jinan, 2017, pp. 1154-1157*, doi: 10.1109/CAC.2017.8242940.
[7] Gallant, A., & Gosselin, C. (2018). Extending the capabilities of robotic manipulators using trajectory optimization. *Mechanism and Machine Theory, 121, 502–514. doi:10.1016/j.mechmachtheory.2017.09*.
[8] Gasparetto, A., & Zanotto, V. (2010). Optimal trajectory planning for industrial robots. *Advances in Engineering Software, 41(4), 548–556*. doi:10.1016/j.advengsoft.2009.11.001
[9] Kucuk, S. (2017). Optimal trajectory generation algorithm for serial and parallel manipulators. *Robotics and Computer-Integrated Manufacturing, 48, 219–232*. doi:10.1016/j.rcim.2017.04.006
[10] Kumar, P., & Sudheer, A. P. (2017). Workspace Optimization of 3PRR Parallel manipulator for drilling operation using Genetic Algorithm. *Proceedings of the Advances in Robotics on - AIR '17*. doi:10.1145/3132446.3134881
[11] L. Shaoming and L. Ruipeng, "Research on trajectory tracking control of multiple degree of freedom manipulator," *2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Hefei, 2017, pp. 218-222*, doi: 10.1109/YAC.2017.7967408.
[12] Li, Y., Huang, T., & Chetwynd, D. G. (2018). An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using quintic B-splines. *Mechanism and Machine Theory, 126, 479–490*. doi:10.1016/j.mechmachtheory.2018.04.026
[13] Liu, Z., Xu, J., Cheng, Q. et al. Trajectory Planning with Minimum Synthesis Error for Industrial Robots Using Screw Theory. *Int. J. Precis. Eng. Manuf. 19, 183–193 (2018)*. https://doi.org/10.1007/s12541-018-0021-3
[14] Macfarlane, S., & Croft, E. A. (2003). Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Transactions on Robotics and Automation, 19(1), 42–52*.
[15] Rossi, C., & Savino, S. (2013). Robot trajectory planning by assigning positions and tangential velocities. *Robotics and Computer-Integrated Manufacturing, 29(1), 139–156*. doi:10.1016/j.rcim.2012.04.003
[16] Parikh P.A., Joshi K.D., Trivedi R. (2021) Vision-Based Trajectory Planning for a Five Degree of Freedom Assistive Feeding Robotic Arm Using Linear Segments with Parabolic Blend and Cycloid Functions. In: *Billingsley J., Brett P. (eds) Mechatronics and Machine Vision in Practice 4. Springer, Cham*. https://doi.org/10.1007/978-3-030-43703-9_16
[17] Valente, A., Baraldo, S., & Carpanzano, E. (2017). Smooth trajectory generation for industrial robots performing high precision assembly processes. *CIRP Annals, 66(1), 17–20*. doi:10.1016/j.cirp.2017.04.105
[18] Wang, H., Wang, H., Huang, J., Zhao, B., & Quan, L. (2019). Smooth point-to-point

trajectory planning for industrial robots with kinematical constraints based on high- order polynomial curve. *Mechanism and Machine Theory, 139, 284–293*. doi:10.1016/j.mechmachtheory.2019.05.002

[19] Wang, X., Zhang, D., Zhao, C., Zhang, H., & Yan, H. (2018). Singularity analysis and treatment for a 7R 6-DOF painting robot with non-spherical wrist. *Mechanism and Machine Theory, 126, 92–107*. doi:10.1016/j.mechmachtheory.2018.03.0

[20] Xiaojie Zhao, Maoli Wang, Ning Liu, Yongwei Tang. Trajectory planning for 6-DOF robotic arm based on Quintic polynomial. *Proceedings of the 2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017)*. https://doi.org/10.2991/caai-17.2017.23

[21] Yishen Guan, Yokoi, K., Stasse, O., &Kheddar, A. (2005). On robotic trajectory planning using polynomial interpolations. *2005 IEEE International Conference on Robotics and Biomimetics - ROBIO*. doi:10.1109/robio.2005.246411

[22] Zhang, J., Meng, Q., Feng, X., & Shen, H. (2018). A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm. *Robotics and Biomimetics, 5(1)*. doi:10.1186/s40638-018-0085-7

[23] Zhang, S., Dai, S., Zanchettin, A. M., & Villa, R. (2020). Trajectory Planning Based on Non-Convex Global Optimization for Serial Manipulators. *Applied Mathematical Modelling*. doi:10.1016/j.apm.2020.03.004

[24] Priyam A. Parikh, Reena Trivedi, and Jatin Dave, "Trajectory Planning for the Five Degree of Freedom Feeding Robot Using Septic and Nonic Functions," *International Journal of Mechanical Engineering and Robotics Research, Vol. 9, No. 7, pp. 1043-1050, July 2020.* DOI: 10.18178/ijmerr.9.7.1043-1050

[25] *Craig,J.J.(2009).Introduction To Robotics:Mechanics And Control, 3/E. India: Pearson Education.*

[26] *Saha, S. K. (2014). Introduction to Robotics. India: McGraw Hill Education (India) Private Limited.*

[27] *Park, F. C., Lynch, K. M. (2017). Modern Robotics: Mechanics, Planning, and Control. Singapore: Cambridge University Press.*