A REAL-TIME TRAJECTORY PLANNING OF REDUNDANT SERIAL MANIPULATOR MOVING ALONG CONSTRAINED PATH

Manav Kumar, Department of Mechanical Engineering, Galgotias University.
Shubham Kumar, Department of Mechanical Engineering, National Institute of Technology Patna.
Sharifuddin Mondal, Department of Mechanical Engineering, National Institute of Technology Patna.
Nav Rattan, Department of Mechanical Engineering, Galgotias University.

ABSTRACT:

This study presents the trajectory planning of a robotic arm with six degrees of freedom (DOF) in real-time with motion in a constraint path. A 6-DOF robotic arm with a base and six links is developed in SOLIDWORKS modelling software and used in MATLAB/Simulink environments. The joint motion limit is provided to support the motion at each joint, and the direction of motion is identified by giving the sine function. Both forward and inverse kinematics of the robotic arm have been performed. Two different trajectories in polynomial trajectory planning have been employed, such as cubic and quintic polynomials with the same waypoint. It is observed that the quintic polynomial trajectory is more effective than the cubic polynomial. It provides a smoother result with a minimum jerk and minimizes power consumption. The robotic arm with a quintic polynomial trajectory is a better option for trajectory planning for a smoother trajectory.

Keywords:Manipulator, Trajectory planning, Robotic arm, Polynomial trajectory.

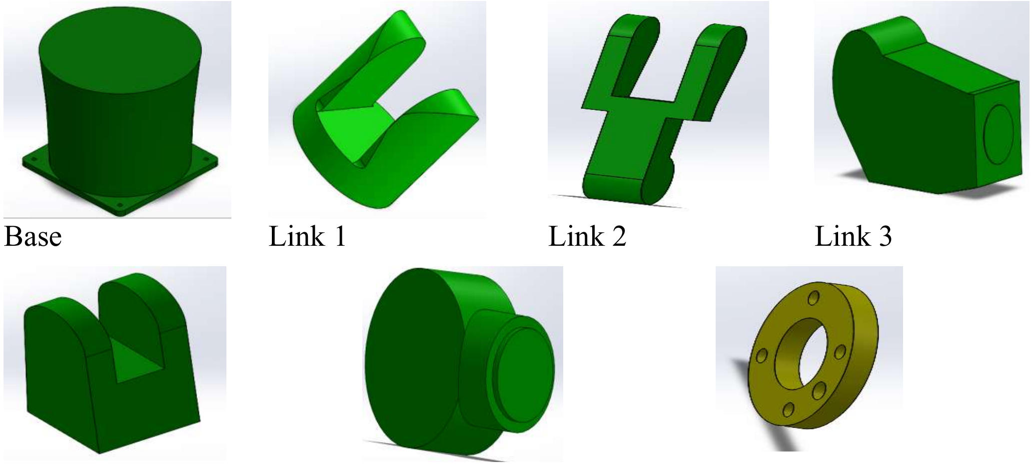
INTRODUCTION :

A particular kind of robot arm known as a robotic manipulator may be used for various activities in both industrial and non-industrial environments and includes several tasks such as handling materials, robotic surgery, painting, welding, assembly, automobiles, electronics, aerospace, healthcare, and others. The working of a robotic manipulator is like an arm mechanism that comprises a sequence of segments that are typically sliding or jointed and referred to as cross-slides. A robotic manipulator's workspace is a three-dimensional area where the robot arm can move around and carry out tasks. Its workspace greatly influences the capabilities of a robotic manipulator. The process of generating a sequence of joint angles that a manipulator should follow to change from its initial location to a desired final location while adhering to constraints such as maximum velocity, maximum acceleration, and avoiding obstacles in its path is known as "trajectory planning." Using interpolation algorithms to construct a smooth trajectory between the starting point and the destination position is a frequent approach that can be taken. The different research works related to trajectory planning for robotic applications are reviewed. Some important works are briefly illustrated here. Faroni et al. [1] investigated robot velocity, acceleration, and torque limits, modifying the velocity profile based on approximated look-ahead criteria. Bulut et al. [2] addressed path planning for hyper-redundant manipulators in cluttered, confined spaces, employing procedures such as finding tangent points and locating path points closest to the link's endpoint. Conkue et al. [3] defined kinematic redundancy when the joint space dimension exceeds the end-effector space, providing enhanced manipulator mobility. Hirose et al. [4] demonstrated how kinematic redundancy affords manipulators greater mobility within their working space. Wen et al. [5]emphasised locating collision-free routes for manipulators to navigate their workspaces without obstruction. Khatib et al. [6]categorised motion planning into low-level and high-level planning, focusing on collision avoidance and efficient collision-free planning. Chirikjian et al. [7] proposed maintaining a constant-length backbone curve for obstacle avoidance. Graham et al. [8] applied an algorithm to generate virtual torque at the manipulator's joints, bypassing complex inverse kinematics and operating in real-time. Konno et al. [9]utilised curvilinear theory to set the posture of hyper-redundant manipulators, employing the serpenoid curve method. Liang et al. [10] discussed using objective-switching functions to navigate workspace obstacles. Burhannuddin et al. [11] introduced collision avoidance path-planning techniques in Euclidean space. Concur et al. [12] suggested discretised paths as a solution to restrictions imposed by B-spline curves' analytical equations. Gammell et al. [13] computed collision-free routes between initial and end configurations as part of the planning process. Xidias et al. [14] proposed trajectory planning of manipulators in hyper-redundant using a time-optimal multi-population evolutionary algorithm and super-concave convex surfaces. Wei et al. [15] presented a time-optimal trajectory planning technique using two-population heredity and chaotic local search. Huashan et al. [16]emphasised considering the manipulator's intrinsic kinematic restrictions when planning trajectories. The present work investigates two different polynomials for trajectory planning of a serial redundant robotic arm in a constrained environment. The arm's trajectory, velocity, and acceleration constraints are considered. The relative trajectory performance of considered polynomials is evaluated.

The paper is arranged in the following order: Section 2 presents the design and modelling of a 6-DOF robotic arm. Section 3 contains the trajectory planning section, in which mathematical models of the considered trajectory polynomial functions are explained. Section 4 analyses and compares the results. Section 5 concludes the paper.

DESIGN AND MODELLING OF A 6-DOF ROBOTIC ARM :

If designated as having six degrees of freedom (DOF), the arm may move in six different directions, also known as axes of motion. The Robotic arm consists of other components, such as the Base, Link 1, Link 2, Link 3, Link 4, Link 5, and Link 6, as shown in Figure 1.



Base Link 1 Link 2 Link 3

Link 4 Link 5 Link 6

Figure 1: Different components of a robotic arm

A robotic arm's base is a part that attaches its body to the platform or structure on which it is mounted. A robotic arm's base is essential because it gives the arm stability and support as it completes its intended duties. Depending on how the robotic arm will be used, the base's design may be changed to accommodate different uses. The arm's workspace may be tailored to reach items in various locations and orientations by changing the position and orientation of the base. "link" refers to a stiff component of a robot arm that links two neighbouring joints. Each link in a chain typically has a specific length and can rotate along one or more axes, providing one or more degrees of freedom (DOF) for the chain as a whole. The number of joints and degrees of freedom required to carry out the desired duties can determine how many links are included in a robotic arm. To minimise the arm's total weight and reduce the energy required to move it, the individual links of a robotic arm are typically constructed out of lightweight and rigid materials. In addition, the links are frequently built to be modular, enabling them to be readily assembled and disassembled, enabling the arm's arrangement to be altered in various ways. A robotic arm's kinematic chain, which explains the relationship between the position and orientation of each link and joint in the arm, is comprised mainly of the arm's links.

Assembling links in a robotic arm is a crucial task that demands precise attention to detail and accurate measurement. This is one of the key steps in constructing a robotic arm. If this step is done improperly, the arm's performance might decline and become inflexible. Because of this, it's crucial to follow the manufacturer's directions and double-check that all the parts have been correctly assembled and aligned. A robotic arm assembly with 6-DOF is shown in Figure 2.

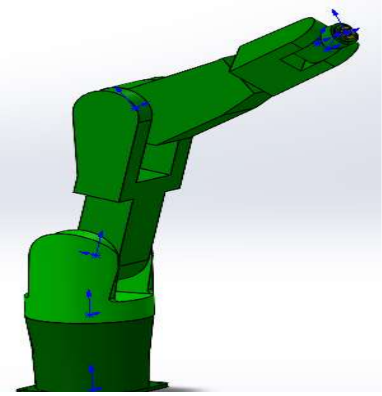


Figure 2: 6-DOF Robotic Arm Assembly

TRAJECTORY PLANNING:

Robot motion planning and control require careful consideration of a robot's trajectory. In addition to creating a smooth and feasible trajectory, a trajectory design for a manipulator necessitates ensuring that the trajectory does not violate any restrictions set on the manipulator's movements. Some of these restrictions are the maximum speed, the maximum acceleration, and the joint limitations. Different polynomial functions are used as trajectory generators. This study selects two different polynomial functions, cubic and quintic polynomials, for investigation.

CUBIC POLYNOMIAL :

The cubic polynomial is a polynomial of degree three, which can be written as:

$$q(t) = a_0 + a_1(t) + a_2(t^2) + a_3(t^3) \tag{1}$$

$$\dot{q}(t) = a_1 + 2a_2(t) + 3a_3(t^2) \tag{2}$$

$$\ddot{q}(t) = 2a_2 + 6a_3(t) \tag{3}$$

where $q(t)$= Trajectory displacement functionin t, $\dot{q}(t)$ = Trajectory velocity function in t, $\ddot{q}(t)$ = Trajectory acceleration function in t, t = time value, and a_0, a_1, a_2, a_3 = coefficient of polynomial. The initial and final trajectory and trajectory velocity are known and described as: $q(t_0)$ = initial trajectory point, $q(t_f)$= final trajectory point, $\dot{q}(t_0)$ = velocity at the initial point, and $\dot{q}(t_f)$= velocity at the final point. The above equations (1)-(3)can be given in the matrix form as $[A][X]=[B]$. The coefficients of cubic polynomials are obtained by solving the equation as $[X]=[A]^{-1}[B]$.

$$[A] = \begin{pmatrix} 1 & (t_0) & (t_0)^2 & (t_0)^3 \\ 0 & 1 & 2(t_0) & 3(t_0)^2 \\ 1 & (t_f) & (t_f)^2 & (t_f)^3 \\ 0 & 1 & 2(t_f) & 3(t_f)^2 \end{pmatrix}, [B] = \begin{pmatrix} q(t_0) \\ \dot{q}(t_0) \\ q(t_f) \\ \dot{q}(t_f) \end{pmatrix}, \text{ and } [X] = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

QUINTIC POLYNOMIAL:

The Quintic polynomial is a polynomial of degree five, which can be written as:

$$q(t) = a_0 + a_1(t) + a_2(t^2) + a_3(t^3) + a_4(t^4) + a_5(t^5) \tag{4}$$

$$\dot{q}(t) = a_1 + 2a_2(t) + 3a_3(t^2) + 4a_4(a^3) + 5a_5(t^4) \tag{5}$$

$$\ddot{q}(t) = 2a_2 + 6a_3(t) + 12a_4(t^2) + 20a_5(t^3) \tag{6}$$

where $a_0, a_1, a_2, a_3, a_4,$ and$a_5$are coefficients of polynomials and other terms that have the same meaning as explained in the previous polynomial.The coefficients of the quintic polynomial are obtained by solving the equation as $[X] = [A]^{-1}[B]$.

$$[A] = \begin{pmatrix} 1 & (t_0) & (t_0)^2 & (t_0)^3 & (t_0)^4 & (t_0)^5 \\ 0 & 1 & 2(t_0) & 3(t_0)^2 & 4(t_0)^3 & 5(t_0)^4 \\ 0 & 0 & 2 & 6(t_0) & 12(t_0)^2 & 20(t_0)^3 \\ 1 & (t_f) & (t_f)^2 & (t_f)^3 & (t_f)^4 & (t_f)^5 \\ 0 & 1 & 2(t_f) & 3(t_f)^2 & 4(t_f)^3 & 5(t_f)^4 \\ 0 & 0 & 2 & 6(t_f) & 12(t_f)^2 & 20(t_f)^3 \end{pmatrix}, [B] = \begin{pmatrix} q(t_0) \\ \dot{q}(t_0) \\ \ddot{q}(t_0) \\ q(t_f) \\ \dot{q}(t_f) \\ \ddot{q}(t_f) \end{pmatrix}, \text{ and } [X] = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix}.$$

ANALYSIS METHODOLOGY :

A 6-DOF robotic arm was constructed in SolidWorks modelling software with a base and six links. Each component was modelled separately and assembled in SolidWorks assembly, resulting in a robotic arm with six revolute joints. Subsequently, the model was exported to MATLAB/Simulink, where a Simulink file for the robotic arm was obtained upon extraction. Joint motion limits were then provided to facilitate motion at each joint. A sine function was applied to verify that each joint's input and output corresponded accordingly. Both forward and inverse kinematics analyses of the robotic arm were performed. A signal was generated in the signal builder and fed into the inverse kinematic block to convert the position into angles, which served as inputs to the robotic arm's joints, enabling the robot to follow the trajectory. Regarding angles, the robot's output was then converted back into positions using forward kinematics. The trajectory input in the builder, velocity, and output trajectory at the forward kinematics block were observed in the scope graph. Furthermore, various trajectories with the same waypoints were provided in the polynomial trajectory planning block, such as cubic polynomials and quintic polynomials.

RESULTS AND DISCUSSION :

In this study, trajectory planning on a 6-DOF serial redundant manipulator arm is performed. Cubic, quadratic, and quintic polynomials are provided as a trajectory for this. Position is given as an input in the form of trajectory position. Inverse kinematics converts this position value into an angle. Robot joints use it as an input, perform their operation, and provide output. This angle output is passed through forward kinematics and provides output in the form of position.

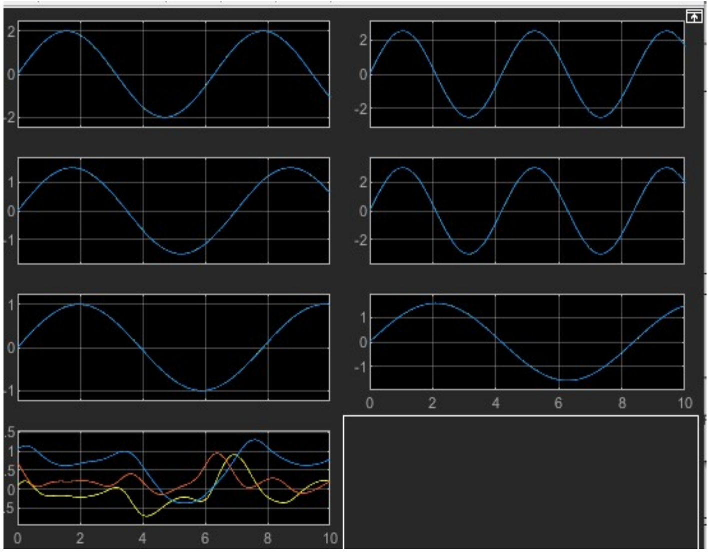


Figure 3: Scope graph of a sine function

As shown in Figure 3, the output of the scope curve for the sine function provides input at all joints. Signals are created for the X-axis, Y-axis, and Z-axis. The output of those signals is shown in Figure 4.

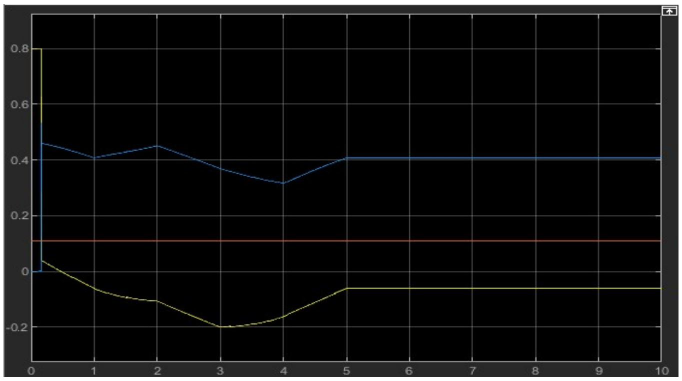


Figure 4: Signal Builder output curves for all three axes.

The main objective of that work is to check how accurate our robot's functioning is. The output obtained must match the input provided. Here, we see that the output matches the input, so we can say that our robot's functioning is accurate. Utilising cubic and quintic polynomials to plan the trajectory of a robotic arm's movements can help optimise the torque delivered by the arm. To achieve a smooth and effective motion, the appropriate torque must be calculated at each point along the arm's path. Altering the coefficients of these polynomials can get the optimal values for the required torque at each point along the arm's path.We calculate the value of the robotic arm's inertia using the mass and length of the link. Still, we use a reference inertia value from previous research and calculate the torque value for different trajectories. In that analysis, the torque value for the quintic polynomial is more optimal than for the other.

$$T = I_{eq} \times \ddot{q} \tag{4}$$

We also plot angular velocity and angular acceleration curves for cubic and quintic polynomials and analyse the variation of both for end effector motion.As shown in Figure 5, the trajectory curve and the velocity and acceleration curve for cubic polynomials vary from 0.5 to 1.5 seconds. In MATLAB programming, we provide the initial and final position. Time is broken into 100 parts using the Linspace command. With the help of the coefficient matrix, we get angular velocity and angular acceleration at all points in time.

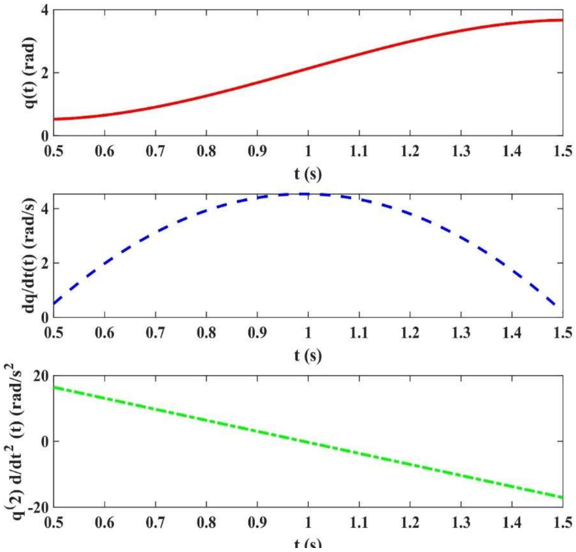


Figure 5: Cubic polynomial curves

Similarly, the maximum torque for a quintic polynomial is obtained and shown in Figure 6. After that, we analyse the minimum torque required for a quintic polynomial for the same Inertia value. So, the power consumption is less for quintic polynomials. In other words, optimum power is obtained for quintic polynomials.

To get the velocity and acceleration profiles, the polynomial equations must be differentiated to compute the torque profile, which is then used to optimise the torque. If the ideal torque required at each location is determined, the robotic arm can be controlled to be smooth and efficient. This prevents the robotic arm from demanding excessive torque, which could either damage the arm or waste energy.

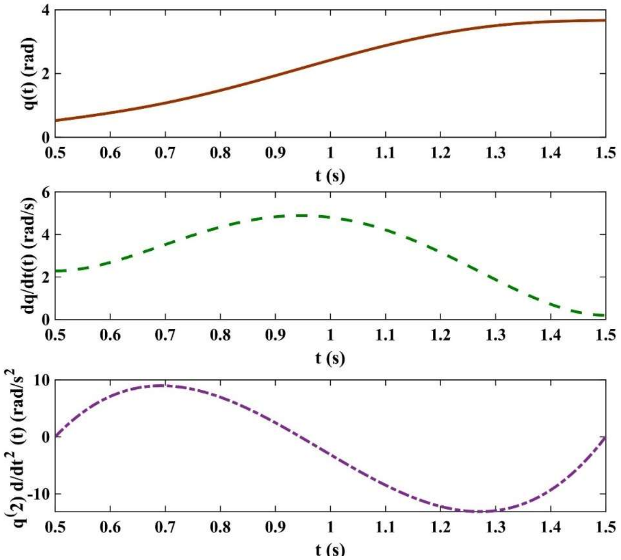


Figure 6: Quintic polynomial curve

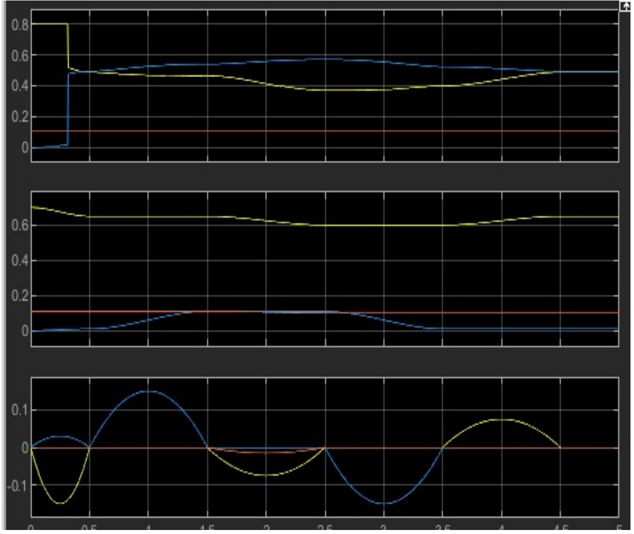


Figure 7: The response curve for cubic Polynomial

We provide cubic polynomial, quintic polynomial, and B-spline as inputs in polynomial trajectory; we select the same waypoint for all and analyse the results in a scope graph. We compare all three and conclude that the quintic polynomial is better than the other two because it has a lower jerk value, a smoother curve, and an optimised torque value. In Figure 7, there are three scope curves for cubic polynomial trajectory. The first is a curve for inverse and forward kinematics output. The second is an input position trajectory scope curve, and the third is an input velocity scope curve.

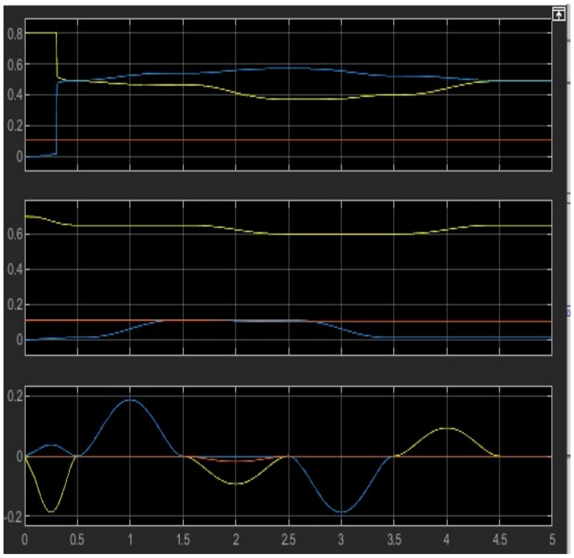


Figure 8: The response curve for quintic polynomial

In Figure 8, there are three scope curves for quintic polynomial trajectory. The first is a curve for inverse and forward kinematics output. The second one is an input position trajectory scope curve, and the third is an input velocity scope curve.

Figure 9 shows three scope curves for the B-spline polynomial trajectory. The first is the curve for the output of inverse and forward kinematics. The second is the input position trajectory scope curve, and the third is the input velocity scope curve. In the last section, we see that the velocity curve for a quintic polynomial is smoother than the other two. So, a quintic polynomial trajectory is preferable to another polynomial trajectory. It also has a minimum power consumption. We also got a velocity and acceleration profile for those polynomials and maximum torque with the help of equations. With those values, we analyze our result and optimize torque.

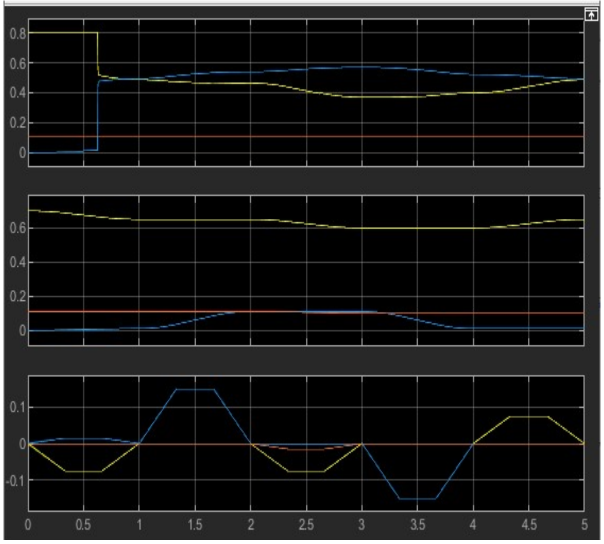


Figure 9: The response curve for B-spline

CONCLUSIONS :

This work presents the trajectory planning of a 6-DOF robotic redundant serial manipulator with motion in a constraint path. The model of a robotic manipulator is developed in SolidWorks, and trajectory-motion analysis is performed in MATLAB/Simulink. Two different polynomials for trajectory planning, namely cubic and quintic polynomials, are selected for the study, and their relative performance is compared. The analysis shows that the torque required is minimum for quintic polynomials and provides a smoother trajectory with a minimum jerk. Moreover, constraints are considered, particularly regarding acceleration constraints; the quintic polynomial trajectory performs better than the cubic polynomial trajectory. Hence, it is concluded that a robotic arm with a quintic polynomial is a better choice for trajectory planning to obtain a smoother trajectory. In the future, the present work will be extended by considering the base movement of robots.

REFERENCES :

[1] Faroni, M., Beschi, M., Visioli, A., Pedrocchi, N.: A real-time trajectory planning method for enhanced path-tracking performance of serial manipulators. Mech. Mach. Theory. 156, 104152 (2021).
[2] Bulut, Y., Conkur, E.S.: A real-time path-planning algorithm with extremely tight maneuvering capabilities for hyper-redundant manipulators. Eng. Sci. Technol. an Int. J. 24, 247–258 (2021).
[3] Conkur, E.S., Buckingham, R.: Clarifying the definition of redundancy as used in robotics. Robotica. 15, 583–586 (1997).
[4] Ma, S., Hirose, S., Yoshinada, H.: Development of a hyper-redundant multijoint manipulator for maintenance of nuclear reactors. Adv. Robot. 9, 281–300 (1994).
[5] Seereeram, S., Wen, J.T.: A global approach to path planning for redundant manipulators. IEEE Trans. Robot. Autom. 11, 152–160 (1995).
[6] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. Int. J. Rob. Res. 5, 90–98 (1986).
[7] Chirikjian, G.S., Burdick, J.W.: An obstacle avoidance algorithm for hyper-redundant manipulators. In: Proceedings., IEEE International Conference on Robotics and Automation. pp. 625–631 (1990).
[8] Graham, A., Buckingham, R.: Real-time collision avoidance of manipulators with multiple redundancy. Mechatronics. 3, 89–106 (1993).
[9] Ma, S., Konno, M.: An obstacle avoidance scheme for hyper-redundant manipulators-global motion planning in posture space. In: Proceedings of International Conference on Robotics and Automation. pp. 161–166 (1997).
[10] Liang, T.-C., Liu, J.-S.: An improved trajectory planner for redundant manipulators in constrained workspace. J. Robot. Syst. 16, 339–351 (1999).
[11] Burhanuddin, L.A., Islam, M.N., Yusof, S.M.: Evaluation of collision avoidance path planning algorithm. In: 2013 International Conference on Research and Innovation in Information Systems (ICRIIS). pp. 360–365 (2013).
[12] Conkur, E.S.: Path planning using potential fields for highly redundant manipulators. Rob. Auton. Syst. 52, 209–228 (2005).
[13] Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In: 2014 IEEE/RSJ international conference on intelligent robots and systems. pp. 2997–3004 (2014)
[14] Xidias, E.K.: Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces. Robot. Comput. Integr. Manuf. 50, 286–298 (2018)
[15] Deng, W., Qiwan, Z., Liu, P., Song, R.: Optimal time trajectory planning based on dual population genetic and chaotic optimization algorithm. Comput. Integr. Manuf. Syst. 24, 101–106 (2018)
[16] Liu, H., Lai, X., Wu, W.: Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. Robot. Comput. Integr. Manuf. 29, 309–317 (2013).