Reduction of positional error in a path of a 2 DOF serial planar manipulator

R N Tatte1*, H T Thorat2 and P V Deosant3

1,3PhD Research Scholar, Visvesvaraya National Institute of Technology, Nagpur, MH
2 Professor, Visvesvaraya National Institute of Technology, Nagpur, MH
*E-Mail: rupeshtatte096@gmail.com

Abstract: Tracing a path is one of the major applications of a robotic arm. In this paper, a 2 DOF serial planar manipulator which traces the desired path is discussed. Quite often the path is defined by accuracy points lying on it. To trace these points, joints of the links of the manipulator are required to be given as coordinated inputs \( \theta_1 \) and \( \theta_2 \) (input angles of two links). This path is traced with a desired speed indicated by the time at which the accuracy points are reached.

In open-loop control, this is achieved by giving time, voltage/current input to driving motors. This path is determined based on the torque, acceleration, velocity of the linkages and individual joints. Hence the relationship between time and joint angles is necessary. When joint angles \( \theta_1 \) and \( \theta_2 \) are interpolated between two accuracy points, precise coordinated motion is not possible. Some deviation from the desired path will occur. This is the positional error for tracing the path. Here, analysis has been done to find the effect of degree of interpolation on the positional error.

Keywords: joint angles; path; positional error; spline interpolation.

1. Introduction

In industry, to perform high precision work robotic arm is used. One of the major applications of a robotic manipulator is to accurately follow a desired path from initial to last position. To get the desired position, joint motors are to be controlled by controlling the current. How much current is required with time is dependent on torque and this torque is known from the relationship of joint angles\[1\]. To accomplish this, robotic manipulators are commanded with program sequences that are executed in digital computers. Within these computers, there is an operating software that provides the information on positions and orientations of the end effectors by computing them in the joint coordinated system as a function of time. Generally, whole path is generated by using single polynomial and it is used in most of the computers commanded industrial robots which are not exact enough such that difference between the actual and the desired position can be significant. Differences as high as 10 mm may be possible for the robots with roughly 0.1 mm repeatability\[2\]. If the function used of lower degree polynomial, the difference goes high. To decline this positional error, higher degree polynomial as a function of joint variable is recommended. However, the complex path cannot be executed by the single polynomial even if higher degree polynomial is used\[3\]. At the same time computer needs to handle more complex calculations and requires more storage data.

To overcome this complexity, whole path is being composed piece by piece known as a spline. By interpolating each spline, a continuous and smooth curve is made out of it of the required degree\[4\]. For interpolating each spline, joints of the links are required to be given as a coordinated input \( \theta_1 \) and \( \theta_2 \). Positions of joint angles have been identified using inverse kinematics of 2 DOF serial planar manipulator\[5\]. After accuracy points to obtained continuous motion, the relationship of joint angles with respect to time is required to be given. To start with the position of the accuracy points and the time...
interval between them is known, the relationship between time and joint angles are determined independently by interpolation. Such a method does not guarantee coordinated motion and hence the possibility of positional error that is a deviation from the desired path exists. An attempt has been made to use different degrees of interpolation to determine the effect of positional error.

2. Path generation using spline

Whenever end-effector moves from one point to another point, the motion is defined either as from point to point or as path generation. This path is generated with a specific objective which can be like avoiding the obstacles, optimizing to minimize energy/power requirement/execution time/jerk and so on. These are the criteria by which paths are identified [6].

Path can be defined by an equation or by accuracy points lying on the desired path. These equations are possible for a simpler path but they become complex for the intricate path. If there is a standard path, it is sufficient to give only one equation, however for a complex path it may not be sufficient to define it with only one equation. For this, it is required to use combinations of equation of required degree [7],[8],[9].

Generating coordinated inputs based on equations becomes complex under such situations. Using accuracy points is a simpler method. Using these accuracy points, coordinated inputs are obtained using inverse kinematics of 2 DOF serial planar manipulator. These positions are used for interpolation using ‘spline’ and continuous path is obtained[4]. Depending on the degree of spline used, path accuracy varies, however it is accurate at the defined point. Inaccuracies because of the use of spline to defined path are termed here as path inaccuracies and in terms of error, it is called as positional error ($e_{pd}$).

3. Generating path in time domain

Controlling the speed of an end-effector, while tracing a path, is also an important issue. Generating a specific path involves only the relationship between $\theta_1$ and $\theta_2$ and precise calculations of $\theta_1$ and $\theta_2$ is feasible for all the points defined on the path. However, if the path is defined with time, $\theta_1$ and $\theta_2$ are also required to be defined with reference to time and hence now the relationship between $\theta_1$ and $\theta_2$ is also required to be defined with reference to time. If the time verses $\theta_1$ and time verses $\theta_2$ curves are to be defined with reference to the accuracy points by some kind of interpolation, this is likely to cause deviation from the planned path. This deviation from the planned path is called as positional error ($e_{nc}$)[5]. Here it is called positional error due to non-coordination. Defining path by accuracy points and hence identifying time versus joint angle relationship based on the accuracy points can work as a generalized method for controlling joint motor motion.

![Figure 1: 2 DOF serial planar manipulator and its parameter.](image)
Fig. 1 shows the kinematic arrangement of 2 DOF serial planar manipulator where θ₁ is the angle between link 1 with respect to horizontal axis, θ₂ is the angle between link 2 with respect to link 1. Similarly, \((l_1, l_2)\) and \((\omega_1, \omega_{12})\) are the respective length and angular velocity of link 1 and link 2. ‘P’ is the position of end-effector with \(x\) and \(y\) coordinate.

Equations used for direct and inverse kinematics are as under [10].

\[
x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \tag{1}
\]
\[
y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{2}
\]
\[
\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \pm \sin^{-1}\left(\frac{l_2}{\sqrt{x^2 + y^2}} \sin\left(\cos^{-1}\left(\frac{l_1^2 + l_2^2 - x^2 - y^2}{2l_1l_2}\right)\right)\right) \tag{3}
\]
and
\[
\theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \tag{4}
\]

To get the equations of joint angle \(\theta_1\) and \(\theta_2\) with respect to time through interpolation, it is required to specify the positions of \(\theta_1\) and \(\theta_2\) with respect to time which has been obtained from the known accuracy points \((x, y)\) lying on the desired path. From these equations, it is possible to get \(\theta_1\) and \(\theta_2\) at different positions at a different time. Since these joint angles are a function of time \(t\), it is easy to find out the successive derivatives of the equation. The relationship of a joint coordinated system with respect to time has been obtained from the given polynomial equation.

\[
\theta(t) = a_n t^n + a_{n-1} t^{n-1} + \cdots + a_t + a_0 \tag{5}
\]
\[
\omega(t) = n a_n t^{n-1} + (n-1) a_{n-1} t^{n-2} + \cdots + a_1 \tag{6}
\]

Eqs. (5) and (6) is the ‘\(n\)’ degree polynomial equation for joint angle and joint angular velocity. To use these equations, it is required to calculate all \((n+1)\) constants in the polynomial. These constants have been calculated from the initial and final boundary conditions of each segment. The number of initial and final boundary conditions required are depended on the number of segments that are generated by splitting the path and also on the degree of the polynomial.

4. Estimation of positional error

In this, two different paths have been analysed which is a straight-line path and parabolic path for estimating positional error. To eliminate the effect of \(e_{pd}\), single equation curves are used. Hence here forth, positional error \(e\) and positional error due to non-coordination \(e_{nc}\) are same (\(e = e_{nc}\)). To define the time verses \(\theta\) relationship, interpolation between the accuracy points has been done using three different degrees of interpolation linear, quadratic and cubic. Based on these three interpolations the angles are identified for ten segments. Using direct kinematics, the position of end-effector is determined and this is called as the actual path. The end conditions used for the three cases are as under.

| \(\theta_1\) (degree) | \(\theta_2\) (degree) | \(\omega_1\) (deg./sec) | \(\omega_2\) (deg./sec) |
|----------------------|----------------------|------------------------|------------------------|
| -36.86               | 90                   | 0                      | 0                      |
| $\theta_1$ (degree) | $\theta_2$ (degree) | $\omega_1$ (deg./sec) | $\omega_2$ (deg./sec) |
|---------------------|---------------------|-----------------------|-----------------------|
| -36.86              | 90                  | 0                     | 0                     |
| -34.59              | 100.86              | 2.27                  | 10.86                 |
| -30.77              | 109.95              | 3.82                  | 9.08                  |
| -25.51              | 117.61              | 5.25                  | 7.66                  |
| -18.89              | 124.05              | 6.61                  | 6.44                  |
| -11.01              | 129.40              | 7.88                  | 5.34                  |
| -1.97               | 133.72              | 9.03                  | 4.32                  |
| 8.03                | 137.07              | 10.01                 | 3.34                  |
| 18.79               | 139.46              | 10.76                 | 2.38                  |
| 30.04               | 140.89              | 11.24                 | 1.43                  |
| 41.49               | 141.37              | 0                     | 0                     |

Table 2: End conditions used for interpolation in parabolic path.

For linear interpolation, initial and final joint angles are used in end conditions. For quadratic interpolation, initial and final joint angles, as well as initial angular velocities, are used in end conditions. Whereas in cubic interpolation, all the four boundary conditions are used. Straight-line path used in this analysis starts with the coordinates $x=0.5$ m, $y=0$ m and ends with $x=0$ m, $y=0.5$ m, whereas parabolic path starts with $x=0.5$ m, $y=0$ m and ends with $x=0$ m, $y=0.25$ m. Length of the links are taken as $l_1 = 0.4$ m and $l_2 = 0.3$ m.

The relationship between $\theta_1$ and $\theta_2$ for a segment is determined by linear interpolation. This has given 10 values of $\theta_1$ and $\theta_2$ and from these value 10 actual positions of end-effector are determined. The maximum positional error between the actual path and desired path for each segment is determined by taking perpendicular distance between the paths. A graph is plotted showing actual path and desired path. Further, the positional error in each segment is expressed as percentage positional error by taking segment length as reference length. This figure is also plotted segment-wise. Same procedure is repeated for all the three interpolation that is linear, quadratic and cubic interpolation. Same procedure is adopted for straight-line and parabolic path.

Figure 2 gives the actual path and desired path superimposed over each other for straight-line path and figure 3 gives it for parabolic path. Figure 4 and 5 give enlarge view for one segment each. All these figures represent that the positional error is very small. Figures 6 and 7 show percentage positional error for each segment for straight-line and parabolic path respectively. For straight-line path, linear interpolation gives positional error around 2.5 % uniformly all over the segment. However, in quadratic interpolation, there is a slight increase in positional error. However, cubic interpolation gives substantially low positional error around 1 %. In case of parabolic path, the error continuously decreases from 2 % to 1.4 % from first segment to the last segment for the linear interpolation. Quadratic interpolation has a similar trend but positional error is marginally low. However, the cubic interpolation has given the positional error around 0.6 %.
Figure 2: Actual path vs. Desired path for straight-line path

Figure 3: Actual path vs. Desired path for parabolic path.

Figure 4: Enlarge view of one segment for straight-line path.

Figure 5: Enlarge view of one segment for parabolic path.
5. Method to reduce positional error in first segment

It is observed that, cubic interpolation gives nearly three times higher positional error in first segment compared to succeeding segments, whereas positional error in linear and quadratic interpolation is at the same level as that of other segments. This is caused due to undefined slope in the initial boundary condition. This definition does not ensure that path starts in the right direction. While defining the path if slope of the curve at the beginning is not defined or considered as 0 degree/sec. such errors are expected for the initial condition.

To take care of these errors a simple method is adopted. One additional segment is created at the beginning of the path having very short length (one tenth of the first segment). The slope of the first two points of the segment from zero has been considered as an initial boundary condition and $\theta_1$ and $\theta_2$ are determined by the equations used for interpolation.

Figure 8 and 9 shows the effect of this additional procedure in the first segment. Though there is no difference in linear interpolation, quadratic interpolation has given very large reduction in positional error about 90 % in the first segment only and other segments there is no effect. In case of cubic interpolation, there is a drop in positional error about 30 % in the first segment but still, it is higher than the remaining segment.

6. Conclusion

It is observed that the interpolation of $\theta_1$ and $\theta_2$ between accuracy points gives positional error in a path, this is true for a straight-line path (first-degree curve) and also for parabolic path (second-degree curve). In both the profile, linear interpolation has given a large positional error in all the segments of around 2.2 %. There is no significant change if the interpolation is of second degree. However, if cubic interpolation is used in this case it reduced to nearly 0.8 %. However, an exception is observed in first segment where the positional error in case of cubic interpolation has not shown any reduction in positional error. This large positional error in first segment can further be reduced by taking one additional accuracy point at one tenth of the first segment from zero. Using this method, the first segment positional error using cubic interpolation came down to the level comparable to succeeding segments. However, it is further observed that quadratic interpolation in the first segment has still further reduced positional error.

Overall, it is observed that cubic interpolation for determining the intermediate position of $\theta_1$ and $\theta_2$ is advantageous compare to linear and quadratic interpolation. This investigation needs to be further extended to optimize the interpolation and jerk at accuracy point. This can further be extended, if the velocity profile along the path is provided.
References

[1] Luh J Y S and Lin C S 1980 Optimum path planning for mechanical manipulators Am. Soc. Mech. Eng., 104, 142-151.
[2] Chen J and Chao L M 1987 Positioning error analysis for robot manipulators with all rotary joints IEEE J. Robot. Autom., 3(6), 539–545.
[3] Akima H 1970 A new method of interpolation and smooth curve fitting based on local procedures J. ACM, vol. 17(4), 589–602.
[4] Keith L 1985 On-Line Polynomial Trajectories for Robot Manipulators The international journal of robotic research, 4(2), 38-48.
[5] Rout B K and Mittal R K 2008 Parametric design optimization of 2-dof r-r planar manipulator-a design of experiment approach Robot. Comput. Integr. Manuf., 24(2), 239–248.
[6] Savsani P et al 2014 Comparative study of different metaheuristics for the trajectory planning of a robotic arm IEEE Systems Journal, 1–12.
[7] Gasparetto A et al 2015 Path planning and trajectory planning algorithms: a general overview Motion and Operation Planning of Robotic Systems, 1, 3-27.
[8] Ichida K et al 1977 Curve fitting by a one-pass method with a piecewise cubic polynomial ACM Trans. Math. Softw., 3(2), 164–174.
[9] Ellis T R and McLain D H 1977 Algorithm 514: a new method of cubic curve fitting using local data ACM Trans. Math. Softw., 3(2), 175–179.
[10] Mittal R K and Nagrath I J 2003 Robotics and control Tata McGraw Hill Publication, New Delhi.