TASK TIME OPTIMIZATION OF TWO DOF ROBOT ARM MANIPULATOR USING GENETIC ALGORITHM

CHANDRA SHEKHAR, S.C.JAYSWAL
1,2Mechanical Engineering Department MMMUT, GORAKHPUR UP INDIA
E-mail: 1chandrashekarskd24@gmail.com, 2scjay_gkp@yahoo.co.in

Abstract - Task time optimization of 2 DOF planar robotic arm manipulator proposed by genetic algorithm is presented in this paper. The point to point trajectory by cubic polynomial from initial to intermediate point as well as from intermediate to final point is used. The algorithm prepared which gives optimum (minimum) task time while the torque of the manipulator not exceeding a maximum predefined value. To avoid the singular configuration, direct kinematics has been used.

Keywords - Genetic algorithm, optimization, singular configuration

I. INTRODUCTION

In the present era, a large number of robots are used in industries for simple pick and place operation and other operations which greatly improving the production efficiency. The repetitive tasks performed by the robots and in that situation the minimization of energy consumption and time will become significantly important.[1]

In the figure.1 shown, the robot manipulator arm pick up an object at point A (initial point) and place at goal position B (goal position or final position). The motion of manipulator arm generates a path or trajectory between the initial position and final position through the intermediate points. This generation can be best if the output parameters of generation with the given input constraints are optimum. To minimize the task time of manipulator is the primary need for the trajectory generation while the torque not exceeding a maximum predefined value. Many authors used genetic algorithm (GA) for trajectory path planning of robot manipulator arm. S.G.Yue et al. [2] proposed the algorithm to minimize vibration of flexible redundant robot manipulator arm based on genetic algorithm. Pires and machado [3] focus on the problem of path planning based on genetic algorithm with the objective function of minimum path length and without any collision with the obstacle in the workspace. Pires et al. [4] optimize robot structure, optimizing the required manipulator trajectory using genetic algorithm. P.Garg and M. Kumar [5] use genetic algorithm for arm to identify the optimal trajectory based on minimum joint torque and use 4th degree polynomial.

The objective of this paper is to optimize the trajectory of 2 DOF planar manipulator arm with minimum task time and without exceeding the maximum predefined torque value also considering the total joint travelling distance of the manipulator and total Cartesian trajectory length in the objective function.

II. ROBOT KINEMATICS

For controlling the motion of a manipulator it is necessary to develop techniques for representing the position of at points in time. Kinematics is the science of motion that treats the subject without regard to the forces that cause it. The science of kinematics studies the position, the velocity, the acceleration, and all higher order derivatives of the position variables. Hence, the study of the kinematics of manipulators refers to the geometrical and time-based properties of the motion.
2.1 Forward kinematics
The joint space representation the joint angles \((\theta_1, \theta_2)\) of robot manipulator arm can be converted into world space representation \((X_2, Y_2)\), this is called forward kinematics. For 2 DOF planar manipulator arm, it is done by the following equations:

\[
X_2 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \quad (1)
\]

\[
Y_2 = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \quad (2)
\]

Where, \(L_1, L_2\) are link lengths.

2.2 Inverse kinematics
The world space representation \((X, Y)\) of robot manipulator arm can be converted into joint space representation the joint angles \(\theta_1, \theta_2\) by the following equations for 2 DOF planar manipulator arm

\[
cos\theta_2 = \frac{x_2^2 + y_2^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (3)
\]

\[
tan\theta_1 = \frac{y_2 (l_2 \cos\theta_2 + l_1) - x_2 l_2 \sin\theta_2}{x_2 (l_2 \cos\theta_2 + l_1) + y_2 l_2 \sin\theta_2} \quad (4)
\]

2.3 Work space of 2 DOF manipulator arm

Figure 3 shows the workspace of 2 DOF planar manipulator arm. The workspace is of disk shape. 2 DOF planar manipulator arm has two links \(L_1, L_2\) and two joint angles \(\theta_1, \theta_2\), if angles have no constraints and free to rotate \(360^\circ\) then the workspace will be a disk (shaded portion) as shown in figure.3 with outer radius \(L_1 + L_2\) and inner radius \(L_1 - L_2\). Further constraints on joint angles reduce the workspace.

III. TRAJECTORY PLANNING

Let the trajectory path between the initial position and goal position is connected via intermediate points by several segments with continuous acceleration at the connection points. The intermediate points can be given as particular points through which trajectory should pass through. If one can specify the position, velocity and acceleration at beginning and end of these segments, cubic polynomial can be used. Generally Polynomial curves are very flexible and useful where a model is developed empirically. They fit a wide range of curvatures. Polynomial curves have certain advantages:

- Easily computed and
- Infinitely differentiable.

Let there is one intermediate point \(\theta_m\) between the initial and goal position.

A cubic polynomial curve is represented mathematically, between initial and intermediate point in time \(t_i\) as

\[
\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (5)
\]

Where \(\theta\) is the function of time \(t\) and \(a_0, a_1, a_2, a_3\) are the arbitrary constants.

As there is four arbitrary constant in cubic polynomial require four boundary conditions to generate the trajectory.

Suppose, equation (5) shows the position function then one time differentiation will give velocity as

\[
v(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (6)
\]

Again, differentiation will give acceleration, as

\[
a(t) = 2a_2 + 6a_3 t \quad (7)
\]

For a robot manipulator motion takes \(t_1\) time from initial point \(A\) to the goal point \(B\) the four conditions may be such that:

I. At \(t=0\), \(\theta(t) = \theta_0\) (initial position)

II. At \(t=t_1\), \(\theta(t) = \theta_m\) (intermediate position)

III. At \(t=0\), \(v(t)=0\)

IV. At \(t=t_1\), \(v(t)=0\)

On applying these four boundary conditions in Equation (5), (6), we get the arbitrary constants as

\[
a_0 = \theta_0
\]

\[
a_1 = 0
\]

\[
a_2 = \frac{3}{t_f^2} (\theta_m - \theta_0)
\]

\[
a_3 = \frac{2}{t_f^2} (\theta_m - \theta_0)
\]

And a cubic polynomial curve is represented mathematically, between intermediate point and goal position in time \(t_2\) as
\[ \theta(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \quad (8) \]

Where \( \theta \) is the function of time \( t \) and \( b_0, b_1, b_2, b_3 \) are the arbitrary constants.

As there is four arbitrary constant in cubic polynomial require four boundary conditions to generate the trajectory.

Suppose, equation (8) shows the position function then one time differentiation will give velocity as

\[ v(t) = b_1 + 2b_2 t + 3b_3 t^2 \quad (9) \]

Again, differentiation will give acceleration, as
\[ a(t) = 2b_2 + 6b_3 t \quad (10) \]

For a robot manipulator motion takes \( t_f \) time from initial point A(say) to the goal point B(say) the four conditions may be such that:

1. \( \text{At } t=0, \theta(t) = \theta_m \) (intermediate position)
2. \( \text{At } t=t_2, \theta(t_2) = \theta_f \) (final position)
3. \( \text{At } t=0, v(t)=0 \)
4. \( \text{At } t=t_2, v(t)=0 \)

On applying these four boundary conditions in Equation (8), and (9), we get the arbitrary constants as
\[ b_0 = \theta_m \]
\[ b_1 = 0 \]
\[ b_2 = \frac{3}{t_f^2} (\theta_f - \theta_m) \]
\[ b_3 = \frac{2}{t_f^3} (\theta_f - \theta_m) \]

### 3.1 Mathematical modeling of the objective function

Four output parameters are used to achieve the trajectory robotic manipulator workspace. All output parameters are translated into penalty functions to be minimized. Each output parameter is computed individually and is added in the fitness function calculation. The fitness function \( f_f \) for calculating the required trajectories is as follows:

\[ f_f = \gamma_1 f_x + \gamma_2 f_y + \gamma_3 f_q + \gamma_4 t_f \quad (11) \]

The optimization goal consists of finding a set of design parameters that minimize the \( f_f \) according to the preferences given by the weight factors \( \gamma_1, \gamma_2, \gamma_3, \) and \( \gamma_4, \) where each different set of weighing factors must result in a different solution.

Mathematical modeling for torque equation of 2DOF planar manipulator arm requires kinematic and dynamic analysis of the manipulator. Figure 4 shows two DOF planar manipulator with link lengths L1, L2 and M1, M2 are the masses of links which are supposed to be concentrated at link ends.

![Figure 4 Kinematic and dynamic parameter representation](image)

Then by geometry
\[ X_1 = L_1 \cos \theta_1 \]
\[ Y_1 = L_1 \sin \theta_1 \]
\[ X_2 = L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2) \]
\[ Y_2 = L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2) \]

The Lagrangian dynamic formulation provides the way of driving the equation of motion from a scalar function called the Lagrangian function \( L \), which is defined as the difference between the kinetic energy and potential energy of a mechanical system. It is energy balance approach to dynamics.

Assumptions made for Lagrangian expression as:-
1. Each manipulator arm is of uniform quality
2. Each link is driven by a motor

Lagrangian expression is given as
\[ L(\theta, \dot{\theta}) = k(\theta, \dot{\theta}) - u(\theta) \quad (16) \]

Where, \( L(\theta, \dot{\theta}) \) = Lagrangian function as a function of position and velocity
\[ k(\theta, \dot{\theta}) = \text{kinetic energy of the link} \]
\[ u(\theta) = \text{Potential energy of the link} \]
\[ k = \frac{1}{2} M_1 \dot{X}_1^2 + \frac{1}{2} M_4 \dot{Y}_1^2 + \frac{1}{2} M_2 \dot{X}_2^2 + \frac{1}{2} M_2 \dot{Y}_2^2 \quad (17) \]

On putting the values of velocities in x and y direction, we can be obtained by differentiate the position equation with respect to time, shown by 2 DOF manipulator kinematics equations.

Since \( v_{c_1} \) and \( \omega_{c_1} \) are functions of \( (\theta, \dot{\theta}) \), so that the kinetic energy of a manipulator can be described by a scalar formula as a function of joint position and velocity as
\[ k(\theta, \dot{\theta}) = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} \quad (18) \]
Where, \[
M(\theta) = 2 \times 2 \text{ manipulator mass matrix}
\]
This equation is called quadratic form equation.

Potential energy of 2 DOF manipulator arm
\[
\begin{align*}
u = M_1gL_1\sin\theta_1 + M_2g(L_1\sin\theta_1 + L_2\sin(\theta_1 + \\theta_2))
\end{align*}
\]
(19)
As \(P_{ij}\) are the functions of \(\theta\), the potential energy of the manipulator can be described by a scalar function joint position \(u(\theta)\).

The equation of motion for the manipulator is than given by
\[
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_i}}\right) - \frac{\partial L}{\partial \theta_i} = \tau
\]
(20)
Where, \(\tau\) is the \(2 \times 1\) vector of actuator torque

Putting the value of Lagrangian function \(L\) from equation (x)
We get
\[
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_i}}\right) - \frac{\partial L}{\partial \theta_i} = \tau_i
\]
(21)
The above equation can be written the form as shown below
\[
M(\theta)\ddot{\theta} + V(\dot{\theta}, \theta) + g(\theta) = \tau
\]
(22)
Where \(M(\theta)\) is mass matrix of \(2 \times 2\)
\(V(\dot{\theta}, \theta)\) is vector of centrifugal and coriolis terms
\(\frac{\partial L}{\partial \theta_i} = \text{vector of gravity terms of } 2 \times 1\)
\(\tau = \text{vector of torque}= [\tau_1, \tau_2]^T\)

Total torque \(J_t\) can be calculated for total trajectory time \(t_f\) as
\[
\begin{align*}
\text{Total torque } \tau = \int_0^{t_f} [\tau_1(t) + \tau_2(t)] \, dt \\
\end{align*}
\]
(23)
\[
f_\tau = \begin{cases} 
0 & \tau < \tau_{\text{max}} \\
\tau - \tau_{\text{max}} & \text{otherwise}
\end{cases}
\]
(24)
The index \(f_q\) represents the total joint travelling distance of the manipulator as
\[
f_q = \sum_{i=1}^{a} \sum_{j=2}^{b} |\theta_{ij} - \theta_{ij-1}|
\]
(25)
Where \(a\) is a number of robot links and \(b\) is the number of joint positions from initial to goal position. The index \(f_c\) represents total Cartesian trajectory length as
\[
f_c = \sum_{i=2}^{b} d(p_i, p_{i-1})
\]
(26)
Where \(p_i\) is arm Cartesian position of the robot and \(d(.,.)\) is a function that gives the distance between the two arguments.
The index \(t_f\) represents the total consumed time for robot motion, as
\[
t_f = t_1 + t_2
\]
(27)
Where, \(t_1\) and \(t_2\) are the execution time from initial to intermediate configuration, and from intermediate to target configuration, respectively.

IV. GENETIC ALGORITHM MOTION PLANNING

The GA planning scheme renders an optimized trajectory having minimum space, minimum time, while not overtaking a maximum defined torque, without colliding with any obstacle in the workspace. The motion planning adopts direct kinematics to avoid singular configuration problems. The trajectory parameters are encoded directly, using real codification, as strings to be used by GA.

For 2R, redundant robot there are six parameters should be optimized as shown in the following chromosome:
\[
[\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, t_1, t_2]
\]
Where \(\theta_1, \theta_2\) and \(\dot{\theta}_1, \dot{\theta}_2\) are intermediate joint angle and velocity for \(i\)th joint respectively, \(t_1\) istime taken from initial to intermediate via point, and \(t_2\) time taken from intermediate to final point.

4.1 Genetic Algorithm (GA)

Genetic algorithm works on the principle of natural selection and natural genetics to constitute search and optimization. It is a computerized search.

![Diagram of genetic algorithm](image-url)
Figure 5 shows the flow chart for the process of genetic algorithm. Generation of initial population takes place randomly and then three main elements of GA optimization called selection, crossover and mutation takes place for new population according to the crossover probability and mutation probability. With the newly generated population Objective function calculated for that and check for minimization. The loop repeated for the Maximum Number of generations given as input. And in the end give the optimized results.[7]

V. SIMULATION

The simulation of 2 R manipulator is performed for the input values given in table 1.

Table 1: Input parameters for robot trajectory

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Robot Trajectory Input parameters</th>
<th>Parameters value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Link length</td>
<td>1.1<del>1.9m, 1.2</del>0.8m</td>
</tr>
<tr>
<td>2.</td>
<td>Link mass</td>
<td>m1=1kg, m2=0.8kg</td>
</tr>
<tr>
<td>3.</td>
<td>Starting position</td>
<td>(x,y)=(-1.5,0)</td>
</tr>
<tr>
<td>4.</td>
<td>Final position</td>
<td>(x,y)=(0,1.6)</td>
</tr>
<tr>
<td>5.</td>
<td>Maximum allowable torque</td>
<td>15 Nm</td>
</tr>
<tr>
<td>6.</td>
<td>Minimum allowable torque</td>
<td>35 Nm</td>
</tr>
</tbody>
</table>

The joints velocities and accelerations of the final and initial positions are assumed to be zeros. Moreover all robot joints are free rotate 360°. Table 2 shows the input parameters for GA optimization.

Table 2: Input parameters for GA optimization

<table>
<thead>
<tr>
<th>S.No.</th>
<th>GA Optimization Parameters</th>
<th>Parameters value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Crossover Probability</td>
<td>0.8 per chromosome</td>
</tr>
<tr>
<td>2.</td>
<td>Mutation Probability</td>
<td>0.05 per locus</td>
</tr>
<tr>
<td>3.</td>
<td>Population</td>
<td>200</td>
</tr>
<tr>
<td>4.</td>
<td>String Size</td>
<td>9</td>
</tr>
<tr>
<td>5.</td>
<td>Maximum generation</td>
<td>80</td>
</tr>
</tbody>
</table>

The weight factors set of fitness faction is \([\gamma_1, \gamma_2, \gamma_3, \gamma_4]=[1, 1.8, 2, 1]\).
Figure 6 shows the trajectory path between the initial position and goal position through intermediate point. The cubic polynomial is used in both the trajectories i.e. from initial to intermediate and intermediate to final position. Figure 7 shows the variation of objective function with the generations. As the 80 generations are taken for GA after 50 generations objective function, become constant. Figure 8 shows variation of joint angles with time further figure 9 and figure 10 shows variation of joint angle velocities and variation of joint angle acceleration with time respectively. Figure 11 shows the variation of joint torque with time, figure 12 shows the consumed time for point to point trajectory and at last figure 13 shows the variation of total trajectory length with the generation of genetic algorithm. The total elapsed time for trajectory generation is $t_g = 6.996541$ seconds.

**CONCLUSION**

Point-to-point trajectory path planning of 2 DOF planar robot arm was presented in detail. The robot kinematics of 2 DOF planar manipulator (include forward kinematics and inverse kinematics), workspace, trajectory planning with cubic polynomial curves, dynamic modeling for torque and objective function formulation with the consideration of total joint travelling distance and total Cartesian trajectory length study was shown. Finally the optimization of objective function using genetic algorithm was done for minimum task time while not exceeding the maximum torque. The optimization with genetic algorithm and trajectory generation with cubic polynomial is easy to debug, less time consuming as compared to other trajectories.

**REFERENCES**


