CLUSTER INNOVATION CENTRE, DELHI UNIVERSITY

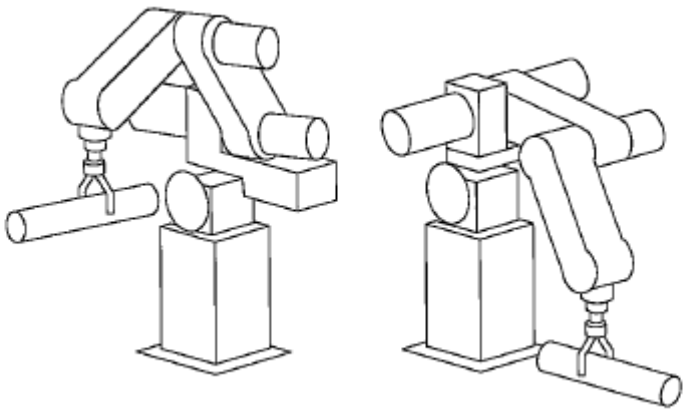
Ordinary Differential Equations

TRAJECTORY PLANNING AND ITS APPLICATIONS

10/4/2017
VAIBHAV JAIN
B.Tech - ITMI
Cluster Innovation Centre, Delhi University

Introduction

Trajectory refers to a time history of position, velocity, and acceleration for each degree of freedom. Problem of Trajectory Planning includes the human-interface problem of how we wish to specify a trajectory or path through space. In order to make the description of manipulator motion easy for a human user of a robot system, the user should not be required to write down complicated functions of space and time to specify the task. Rather, we must allow the capability of specifying trajectories with simple descriptions of the desired motion, and let the system figure out the details. For example, the user might want to be able to specify nothing more than the desired goal position and orientation of the end-effector and leave it to the system to decide on the exact shape of the path to get there, the duration, the velocity profile, and other details.



In executing a trajectory, a manipulator moves from its initial position to a desired goal position in a smooth manner

Joint Space Scheme

Trajectory can be planned in different ways. One of them is Joint Space Scheme. In Joint Space scheme, path is generated in which the path shapes (in space and in time) are described in terms of functions of joint angles.
Each path point is usually specified in terms of a desired position and orientation of the tool frame, {T}, relative to the station frame, {S}. Each of these via points is "converted" into a set of desired joint angles by application of the inverse kinematics. Then a smooth function is found for each of the n joints that pass through the via points and end at the goal point. The time required for each segment is the same for each joint so that all joints wifi reach the via point at the same time, thus resulting in the desired Cartesian position of {T} at each via point. Other than specifying the same duration for each joint, the determination of the desired joint angle function for a particular joint does not depend on the functions for the other joints. Hence, joint-space schemes achieve the desired position and orientation at the via points. In between via points, the shape of the path, although rather simple injoint space, is complex if described in Cartesian space. Joint-space schemes are usually the easiest to compute, and, because we make no continuous correspondence between joint space and Cartesian space, there is essentially no problem with singularities of the mechanism.

Cubic Polynomials

In making a single smooth motion, at least four constraints on Θ(t) are evident. Two constraints on the function's value come from the selection of initial and final values:

$$\theta(0) = \theta_0,$$
$$\theta(t_f) = \theta_f.$$

An additional two constraints are that the function be continuous in velocity, which in this case means that the initial and final velocity is zero:

$$\dot{\theta}(0) = 0,$$
$$\dot{\theta}(t_f) = 0.$$

These four constraints can be satisfied by a polynomial of at least third degree.
(A cubic polynomial has four coefficients, so it can be made to satisfy the four constraints given by (7.1) and (7.2).) These constraints uniquely specify a particular cubic. A cubic has the form

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3,$$

So the joint velocity and acceleration along this path are clearly

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2,$$
$$\ddot{\theta}(t) = 2a_2 + 6a_3 t.$$

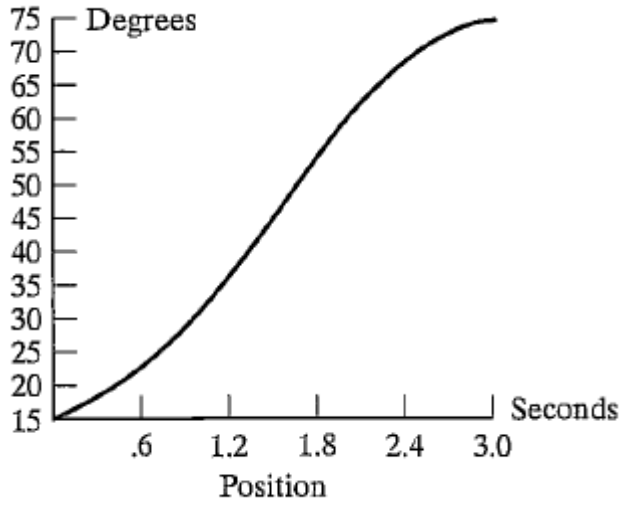
Combining these with the four desired constraints yields four equations in four unknowns

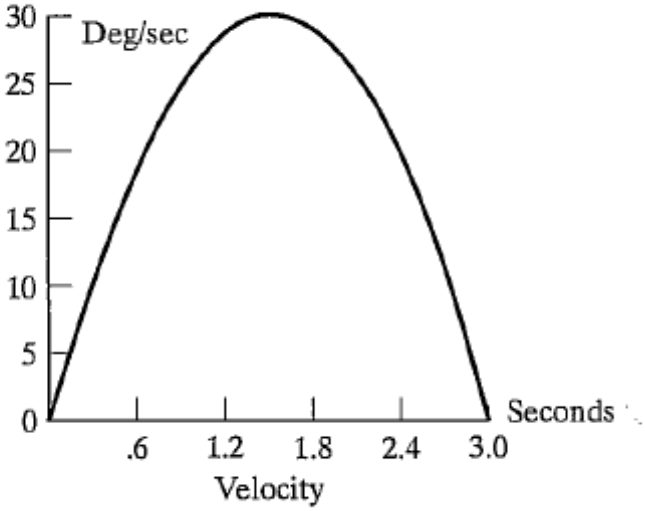
$$\begin{aligned}\theta_0 &= a_0,\\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3,\\ 0 &= a_1,\\ 0 &= a_1 + 2a_2 t_f + 3a_3 t_f^2.\end{aligned}$$

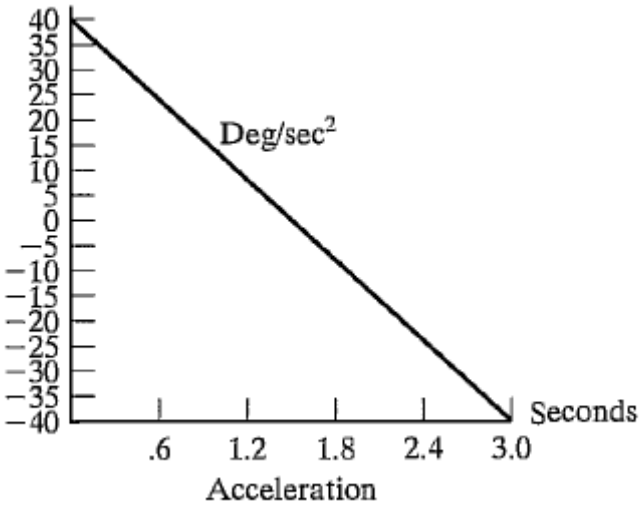
Solving these equations for the 'a', we obtain

$$\begin{aligned}a_0 &= \theta_0,\\ a_1 &= 0,\\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0),\\ a_3 &= -\frac{2}{t_f^3}(\theta_f - \theta_0).\end{aligned}$$

Using these, we can calculate the cubic polynomial that connects any initial joint angle position with any desired final position. This solution is for the case when the joint starts and finishes at zero velocity.
Following are results of these equations after solving for constants:


75
70
65
60
55
50
45
40
35
30
25
20
15
Degrees
Seconds
.6
1.2
1.8
2.4
3.0
Position


30
25
20
15
10
5
0
Deg/sec
Seconds
.6
1.2
1.8
2.4
3.0
Velocity


40
35
30
25
20
15
10
5
0
−5
−10
−15
−20
−25
−30
−35
−40
Deg/sec²
Seconds
.6
1.2
1.8
2.4
3.0
Acceleration

Higher-order polynomials

Higher-order polynomials are sometimes used for path segments. For example, if we wish to be able to specify the position, velocity, and acceleration at the beginning and end of a path segment, a quintic polynomial is required, namely:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

where the constraints are given as

$$\begin{aligned}
\theta_0 &= a_0, \\
\theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\
\dot{\theta}_0 &= a_1, \\
\dot{\theta}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\
\ddot{\theta}_0 &= 2a_2, \\
\ddot{\theta}_f &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3.
\end{aligned}$$

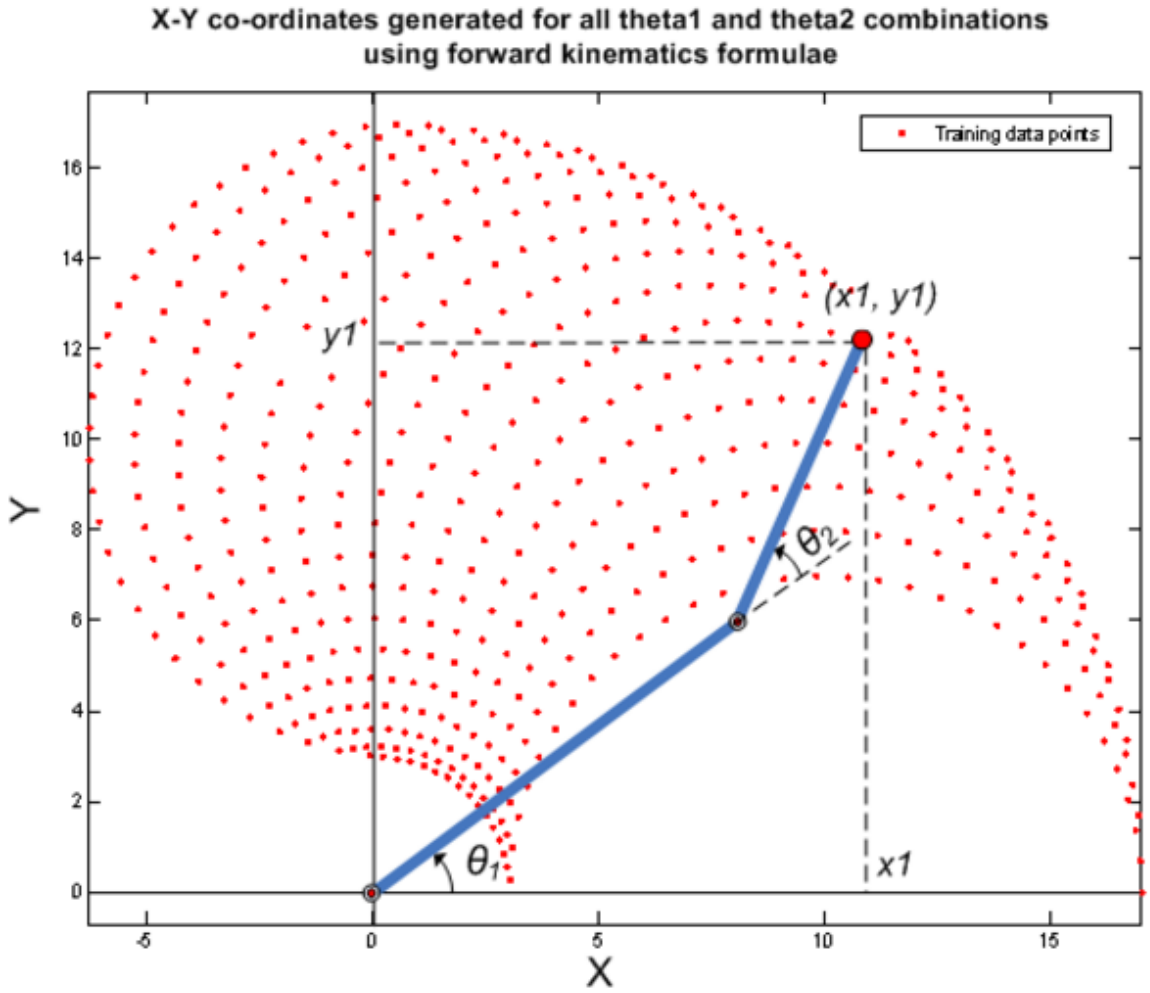
Working Space of an Arm

Working space is the set of points in real space where the arm can reach after following a specific trajectory from any other point in this space. Calculation of work space is critical as it allows us to plan a trajectory for several operations which needs to be done with high precision.

Following is the MATLAB code which generates the graph shown in figure following it. For this code we assume that the first joint has limited freedom to rotate and it can rotate between 0 and 90 degrees. Similarly, the second joint also has limited freedom to rotate and can rotate between 0 and 180 degrees.

```
l1 = 10; % length of first arm
l2 = 7; % length of second arm

theta1 = 0:0.1:pi/2; % all possible theta1 values
theta2 = 0:0.1:pi; % all possible theta2 values

[THETA1,THETA2] = meshgrid(theta1,theta2); % generate a grid of
theta1 and theta2 values

X = l1 * cos(THETA1) + l2 * cos(THETA1 + THETA2); % compute x
coordinates
Y = l1 * sin(THETA1) + l2 * sin(THETA1 + THETA2); % compute y
coordinates

data1 = [X(:) Y(:) THETA1(:)]; % create x-y-theta1 dataset
data2 = [X(:) Y(:) THETA2(:)]; % create x-y-theta2 dataset
```

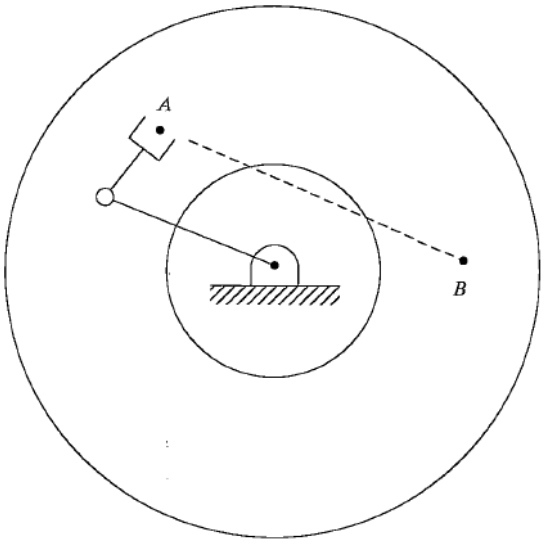
The following plot shows all the X-Y data points generated by the above code by cycling through different combinations of theta1 and theta2 and deducing x and y coordinates for each.



GEOMETRIC PROBLEMS

Problems of type 1: intermediate points unreachable

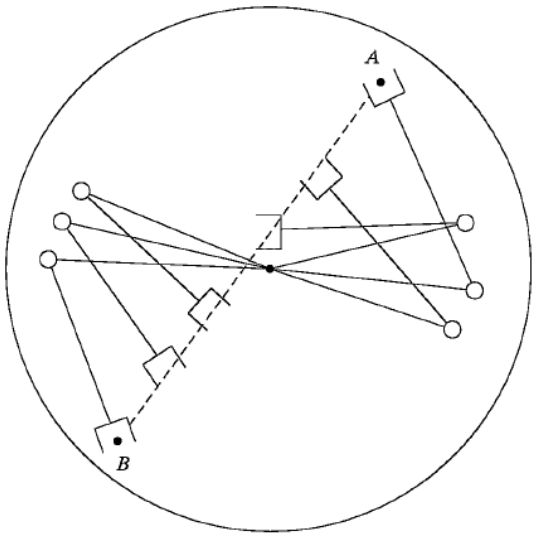
Although the initial location of the manipulator and the final goal point are both within the manipulator workspace, it is quite possible that not all points lying on a straight line connecting these two points are in the workspace.



Cartesian-path problem of type 1.

Problems of type 2: high joint rates near singularity

There are locations in the manipulator's workspace where it is impossible to choose finite joint rates that yield the desired velocity of the end-effector in Cartesian space.



Cartesian-path problem of type 2.