INDUSTRIAL ROBOTICS

Chapter 7:

“Trajectory Generation”

Lecturer:
Engr. Khairuddin bin Osman

Faculty:
Faculty of Electronic & Computer Engineering

BENE 4113
Objective

After this chapter, the students are expected to learn the following:

1. Plan point to point trajectories in joint space and task space

2. Plan trajectories with via points

3. Plan trajectories with velocity and acceleration constraints
To move a manipulator (end effector) to a Goal point, at least 2 points need to be considered:

- Avoiding obstacle in the path between initial position and final position
- Path is specified by user which usually includes intermediate points (via points or knots) between initial and final position. In some instance, user may also specify velocity

A Path is a physical or geometric representation in space.
A Trajectory is a path with time profile (velocity, acceleration). Time history of position, velocity and acceleration of each joint.
To implement the path, the task is divided into: Motion Planning $\rightarrow$ Trajectory Generation $\rightarrow$ Trajectory Tracking

Motion planning: user specifies paths with the 2 considerations

Trajectory generation: controller computes position, velocity and acceleration to interpolates (approximates) the user specified path by polynomial functions and then generate timed-based control set points.

Spong, Mark W., *Motion Control of Robotic Manipulator*, University of Illinois at Urbana-Champaign
Basic Problem

Move the manipulator arm from some initial position \( \{T_A\} \) to some desired final position \( \{T_C\} \).

(May be going through some via point \( \{T_B\} \))

Path points: Initial, final and via points

Trajectory: Time history of position, velocity and acceleration for each DOF

Constraints: Spatial, time, smoothness

\( \{T_A\} \) (Tool frame)

\( \{S\} \) (Stationary frame)
Trajectory Planning

Joint space
- Easy to go through via points
  (Solve inverse kinematics at all path points and plan)
- No problems with singularities
- Less calculations
- Can not follow straight line

Cartesian space
- We can track a shape
  (for orientation: equivalent axes, Euler angles, …)
- More expensive at run time
  (after the path is calculated need joint angles in a lot of points)
- Discontinuity problems
Initial and Goal Points are reachable.

Intermediate points (C) unreachable.
Approaching singularities
some joint velocities
go to $\infty$
causing deviation
from the path
Start point (A) and goal point (B) are reachable in different joint space solutions (The middle points are reachable from below.)
Actual planning in any space:
Assume one generic variable $U$
(can be x, y, z, orientation - $\alpha$, $\beta$, $\gamma$)
joint variables, direction cosines

Candidate curves:
straight line (discontinuous velocity at path points)

straight line with blends

cubic polynomials (splines)

higher order polynomials (quintic,...) or other curves
Cubic Polynomial

- Polynomial Trajectory
  - subject to constraints
  - boundary conditions

- Initial & Final Positions
  \[
  \begin{align*}
  \theta(0) &= \theta_0 \\
  \theta(t_f) &= \theta_f \\
  \dot{\theta}(0) &= 0 \\
  \dot{\theta}(t_f) &= 0
  \end{align*}
  \]

  4 constraints

- Initial & Final Velocities

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

Four Equations (constraints) in Four Unknowns \((a_0, \ldots, a_3)\)

\[
\begin{align*}
  a_0 &= \theta_0 \\
  a_2 &= \frac{3}{t_f^2} (\theta_f - \theta_0) \\
  a_1 &= 0 \\
  a_3 &= \frac{-2}{t_f^3} (\theta_f - \theta_0)
\end{align*}
\]
Single Cubic Polynomial

Position, velocity and acceleration profiles for a single cubic segment that starts and ends at rest:
Cubic Polynomial with Via Points

Choose velocities at via points

- Initial & final positions
- """" velocities
- """" acceleration

6 constants

Quintic Polynomial

\[ \dot{\theta}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \]

6 Equations in 6 unknowns:

\[ a_0 = \dot{\theta}_0 \]
\[ a_1 = \ddot{\theta}_0 \]
\[ a_2 = \frac{\dddot{\theta}_0}{2} \]
\[ a_3 = \frac{20 \theta_f - 20 \theta_0 - (\delta \dot{\theta}_f + 12 \dot{\theta}_0) t_f - (3 \ddot{\theta}_0 - \ddot{\theta}_f) t_f^2}{2 t_f^3} \]

\[ a_4 = \frac{30 \theta_0 - 30 \theta_f + (14 \dot{\theta}_f + 16 \dot{\theta}_0) t_f + (3 \ddot{\theta}_0 - 2 \ddot{\theta}_f) t_f^2}{2 t_f^4} \]

\[ a_5 = \frac{12 \theta_f - 12 \theta_0 - (6 \dot{\theta}_f + 6 \dot{\theta}_0) t_f - (\ddot{\theta}_0 - \ddot{\theta}_f) t_f^2}{2 t_f^5} \]
Linear Segment with Parabolic Blend

- Constant deceleration
- Constant velocity
- Same magnitude of acceleration
- Same Δt’s at start & end
- Same magnitude of slope

Velocity profile
There are many possible solutions, given
\[ \theta(0) = \theta_0 \quad \dot{\theta}(0) = 0 \]
\[ \theta(t_p) = \theta_f \quad \ddot{\theta}(t_p) = 0 \]

\[ \theta_b = \theta_0 + \frac{1}{2} \ddot{\theta} t_b^2 \quad \text{— (1)} \]

\[ \ddot{\theta}_b = \ddot{\theta}_0 + \ddot{\theta} t_b = \frac{\theta_h - \theta_b}{t_h - t_b} \quad \text{— (2)} \]

where \[ \theta_h = \frac{1}{2} \left( \frac{\theta_f + \theta_0}{2} \right) \]

Given \( \theta_0, \theta_f \) and \( t_f \) (move time)

Can choose \( \ddot{\theta} \) and \( t_b \) to satisfy (1) & (2)

Combining (1) & (2) and since \( t_f = 2t_h \), we get

\[ \ddot{\theta} t_b^2 - \dot{\theta} t_f t_b + (\theta_f - \theta_0) = 0 \]

\[ t_b = \frac{t_f}{2} - \frac{\sqrt{\ddot{\theta}^2 t_f^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}} \]

For \( t_b \) to exist,

\[ \ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t_f^2} \quad \text{(acceleration must be sufficiently high)} \]

When equality holds:

\[ t_b = \frac{t_f}{2} = t_h \quad \text{(no constant velocity)} \]
Segment with Parabolic Blend with Vias
Linear Segment with Parabolic Blend with Vias

Given:
- positions \( u_i, u_j, u_k, u_l, u_m \)
- desired time durations \( t_{dij}, t_{djk}, t_{dkl}, t_{dlm} \)
- the magnitudes of the accelerations: \( |\ddot{u}_i|, |\ddot{u}_j|, |\ddot{u}_k|, |\ddot{u}_l| \)

Compute:
- blends times \( t_i, t_j, t_k, t_l, t_m \)
- straight segment times \( t_{ij}, t_{jk}, t_{kl}, t_{lm} \)
- slopes (velocities) \( \dot{u}_{ij}, \dot{u}_{jk}, \dot{u}_{kl}, \dot{u}_{lm} \)
- signed accelerations

Formulas (7.24), (7.26) and (7.28)

System usually calculates or uses default values for accelerations
The system can also calculate desired time durations based on default velocities.
Linear Segment with Parabolic Blend with Vias

First segment

\[ \ddot{u}_1 = \text{sign}(u_2 - u_1) |\ddot{u}_1| \]

\[ t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(u_2 - u_1)}{\ddot{u}_1}} \]

\[ \dot{u}_{12} = \frac{u_2 - u_1}{t_{d12} - \frac{1}{2} t_1} \]

\[ t_{12} = t_{d12} - t_1 - \frac{1}{2} t_2 \]

Inside segments

\[ \dot{u}_{jk} = \frac{u_k - u_j}{t_{djk}} \]

\[ \ddot{u}_k = \text{sign}(\dot{u}_{kl} - \dot{u}_{jk}) |\ddot{u}_k| \]

\[ t_k = \frac{\dot{u}_{kl} - \dot{u}_{jk}}{\ddot{u}_k} \]

\[ t_{jk} = t_{djk} - \frac{1}{2} t_j - \frac{1}{2} t_k \]

Last segment

\[ \ddot{u}_n = \text{sign}(u_{n-1} - u_n) |\ddot{u}_n| \]

\[ t_n = t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 - \frac{2(u_n - u_{n-1})}{\ddot{u}_n}} \]

\[ \dot{u}_{(n-1)n} = \frac{u_n - u_{n-1}}{t_{d(n-1)n} - \frac{1}{2} t_n} \]

\[ t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{1}{2} t_{n-1} \]
Trajectory Planning with Obstacles

Path planning for the whole manipulator
- Local vs. Global Motion Planning
  - Gross motion planning for relatively uncluttered environments
  - Fine motion planning for the end-effector frame
- Configuration space (C-space) approach

Planning for a point robot
- graph representation of the free space, quadtree
- Artificial Potential Field method

Multiple robots, moving robots and/or obstacles
THANK YOU