Trajectory Generation
CS 685

Previously

- How to describe robot' s pose (R, T)
- Mobile robot kinematics, differential drive)
- Notion of degree of freedom (DOF)
- Rigid body motion has 6-DOF

- Robots motion – time-varying pose
- Path, vs trajectory

- Smooth trajectories

Paths and Trajectories

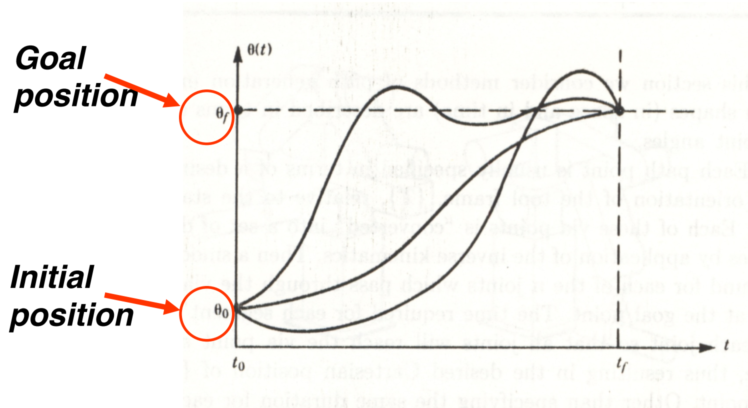
- In general – control problem – need to generate set of control commands to accomplish the task
- In an open loop setting there are two components

1. Geometric Path Generation
2. Trajectory tracking (trajectory – time indexed path)

1D trajectories

- Trajectory, scalar function of time
- We want smooth trajectories
- Temporal derivatives
- Continuous velocity and acceleration profiles

Several Possible Path Shapes for a Single Joint



Cubic Polynomials

4 constraints on $\theta(t)$

$\theta(0) = \theta_0, \quad \theta(t_f) = \theta_f,$: initial and final values

$\dot{\theta}(0) = 0, \quad \dot{\theta}(t_f) = 0.$: the function is continuous in velocity

These 4 constraints can be satisfied by a polynomial of at least third degree.

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

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

Cubic Polynomials

$$\theta_0 = a_0,$$
$$\theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3,$$
$$0 = a_1,$$
$$0 = a_1 + 2a_2 t_f + 3a_3 t_f^2.$$

: combining with constraints

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

The cubic polynomial that connects any initial joint angle position with any desired final position when the joint starts and Finishes at zero velocity.

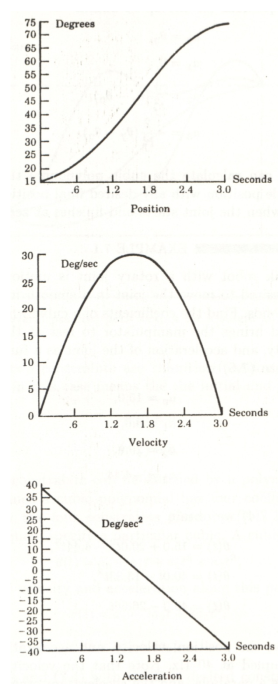
Example

A single-link robot with a rotary joint:

Move the joint in a smooth manner from θ=15 to θ=75 in 3 seconds

$$a_0 = 15.0$$
$$a_1 = 0.0$$
$$a_2 = 20.0$$
$$a_3 = -4.44$$

$$\theta(t) = 15.0 + 20.0t^2 - 4.44t^3$$
$$\dot{\theta}(t) = 40.0\,t - 13.33t^2$$
$$\ddot{\theta}(t) = 40.0 - 26.66\,t$$



1D trajectories

- Acceleration profile not smooth – higher order polynomial
- Continuous velocity and acceleration profiles

$$\theta(t) = at^5 + bt^4 + ct^3 + dt^2 + et + f$$

$$\dot{\theta}(t) = 5at^4 + 4bt^3 + 3ct^2 + 2dt + e$$

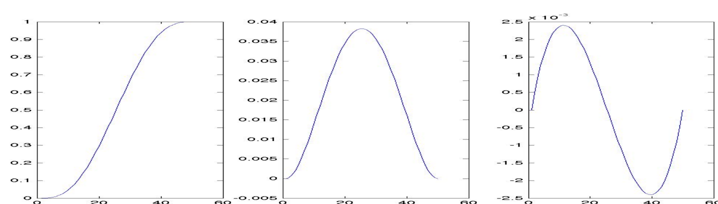
$$\ddot{\theta}(t) = 20at^3 + 12bt^2 + 5ct + 2d$$

- Given initial and final conditions for *t=0* and *t=T*

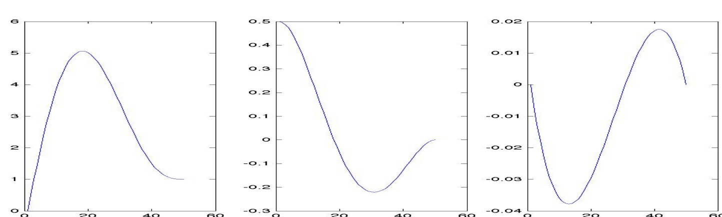
θ	$\dot{\theta}$	$\ddot{\theta}$
θ_0	$\dot{\theta}_0$	$\ddot{\theta}_0$
θ_T	$\dot{\theta}_T$	$\ddot{\theta}_T$

1-D trajectories

- Solve for coefficients, plot trajectories

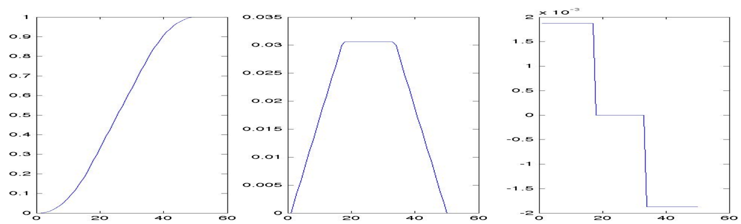


- Non-zero initial values – velocity overshoot at T

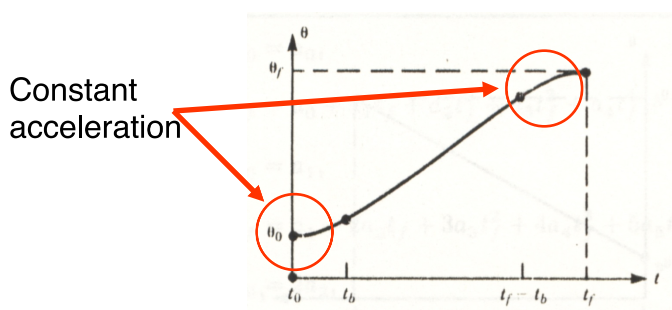


Problems with polynomials

- Overshoots velocity at final value T
- Velocity peaks in the middle, otherwise is far less then maximum
- We should like to have a flatter velocity profiles
- Solution: hybrid trajectories with polynomial segments for acceleration and de=acceleration
- Linear segments with parabolic blends (trapezoidal velocity profiles)

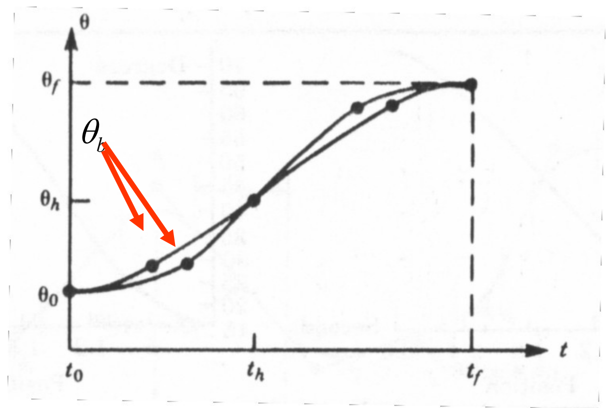


Linear Function With Parabolic Blends



The linear function and the two parabolic functions are splined together so that the entire path is continuous in position and velocity.

Linear Function With Parabolic Blends



The parabolic blends have the same duration, are symmetric about the halfway point in time, and the halfway point in position.

Linear Function With Parabolic Blends

$$\ddot{\theta} t_b = \frac{\theta_h - \theta_b}{t_h - t_b},$$

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta} t_b^2.$$

$$t = 2t_h$$

$$\ddot{\theta} t_b^2 - \ddot{\theta} t t_b + (\theta_f - \theta_0) = 0$$

The velocity at the end of the blend region must equal the velocity of the linear section.

Usually acceleration is chosen and then solve for t_b

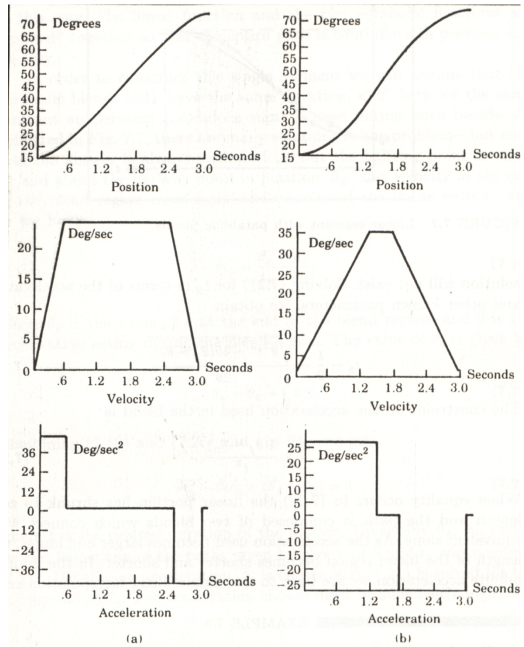
$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}$$

$$\ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t^2}.$$

: the constraints on acceleration

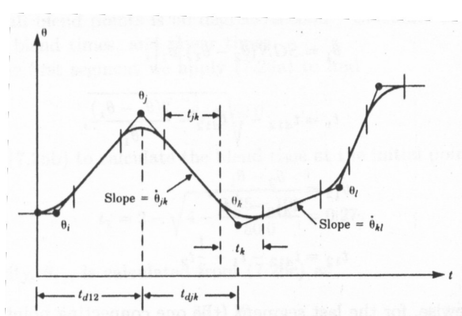
When equality occurs, the linear portion has shrunk to zero length.

Two possible choices of linear path with parabolic blends

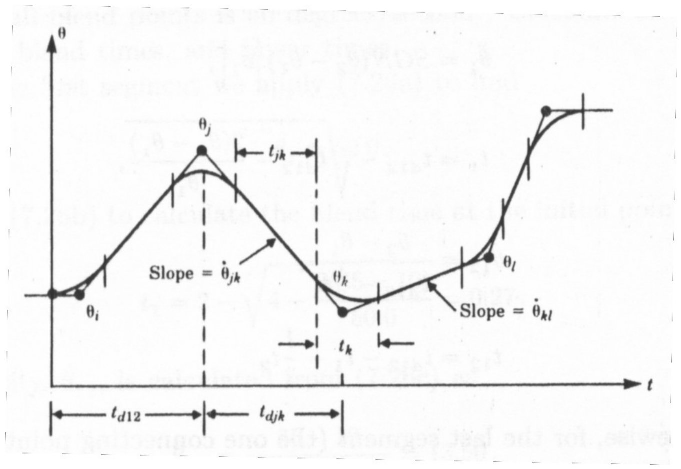


Multi-segment trajectories

- Often need to move through set of way points without stopping
- E.g. to avoid obstacles, or perform a task
- Over constrained problem, we need to surrender ability to reach every point



Linear Function With Parabolic Blends for a Path With Via Points



Linear function connects the via points and parabolic blend regions are around each via point

The Path Generator

- The results of computations constitute a plan for the trajectory. At execution time the path generator will use these numbers to compute θ, $\dot{\theta}$, $\ddot{\theta}$

Paths and Trajectories

- In general – control problem – need to generate set of control commands to accomplish the task
- In an open loop setting there are two components

1. Geometric Path Generation
2. Trajectory tracking (trajectory – time indexed path)

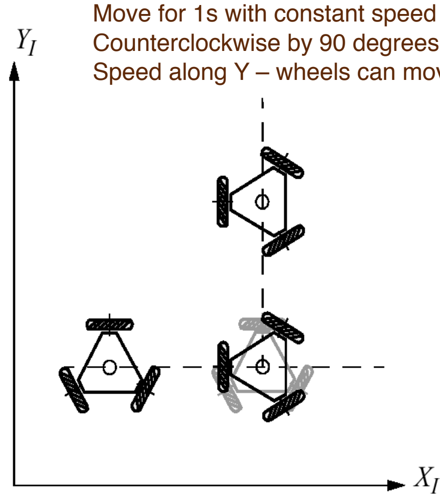
- Example omni-directional robot – can control all degress of freedom independently

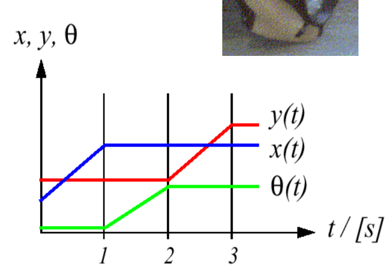
$$\delta_M = \delta_m + \delta_s = 3 + 0 = 3$$

Path / Trajectory Considerations: Omnidirectional Drive

3.4.3

Move for 1s with constant speed along X, change orientation Counterclockwise by 90 degrees in 1s, move for 1s with constant Speed along Y – wheels can move and roll – omni-directional drive

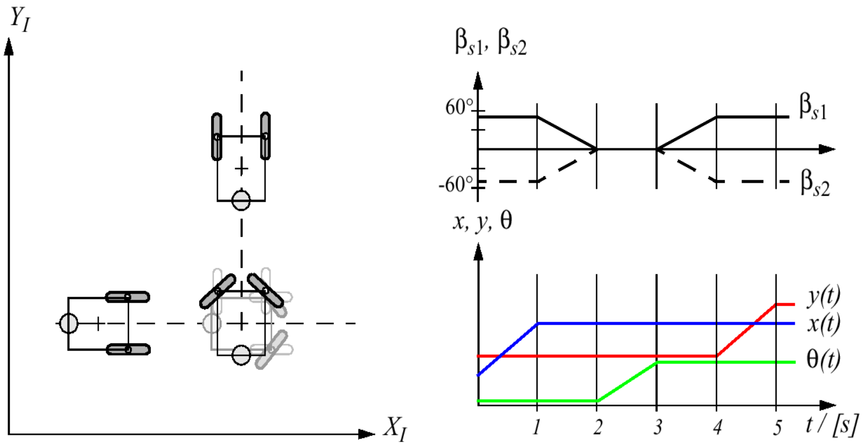




3.4.3

Path / Trajectory Considerations: Two-Steer

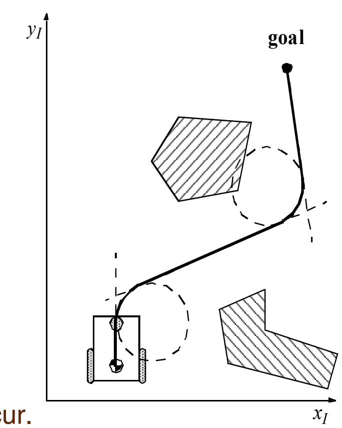
Move for 1s with constant speed along X, rotate steered wheels by -50/50 degrees; change orientation counterclockwise by 90 degrees in 1s, move for 1s with constant speed along Y



3.6.1

Motion Control: Open Loop Control

- Trajectory (path) divided in motion segments of clearly defined shape:
  - straight lines and segments of a circle.
- Control problem:
  - pre-compute a smooth trajectory based on line and circle segments
- Disadvantages:
  - It is not at all an easy task to pre-compute a feasible trajectory
  - limitations and constraints of the robots velocities and accelerations
  - does not adapt or correct the trajectory if dynamical changes of the environment occur.
  - The resulting trajectories are usually not smooth



Pose trajectories

- Examples so far: 1D trajectories (and velocity and acceleration profiles)
- Multi-segment 1D trajectories
- Multi-segment 2D trajectories comprised of lines and circles

- How to generate trajectory for rigid body so as to move from initial pose (R_0, T_0) to final pose (R_1, T_1)
- Interpolation
- Translation only case for $s = [0,1]$ generate intermediate translations as:

$$T = (1 - s)T_0 + sT_1$$

Interpolation of rotations

- Interpolation of rotations

$$R = (1 - s)R_0 + sR_1$$

- This won't work, rotation matrix properties are violated
- Spherical interpolation using quaternions
- Interpolation using exponential parametrization

$$\vec{\omega} = (1 - s)\vec{\omega}_0 + s\vec{\omega}_1$$

- Similarly for full Rigid Body Motion

Incremental Motion

- Small incremental rotations

$$R_1 = (\hat{\omega}\sigma_t + I)R_0$$

- Inertial Navigation Systems
- Estimate velocity, orientation, and position wrt to inertial frame (frame of reference with respect to which is motion described)
- IMU – inertial measurement unit - measures accelerations and angular velocities and integrate them over time (3 orthogonally mounted gyros measure the angular velocity of the body)