Manipulator trajectory planning

Václav Hlaváč

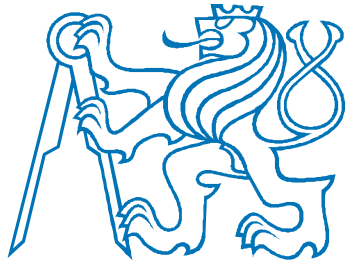
Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Cybernetics

Czech Republic

http://cmp.felk.cvut.cz/~hlavac



Courtesy to several authors of presentations on the web.

Lecture outline

We consider a robotic manipulator (open kinematic chain) for simplicity. Generalization of methods for mobile robots comes in future lectures.

- Problem formulation - trajectory generation.
- Path vs. trajectory.
- Path planning vs. trajectory generation.
- Joint space vs. operational (Cartesian) space.
- Via points and point to point planning / trajectory generation.
- Trajectory approximation / interpolation functions.

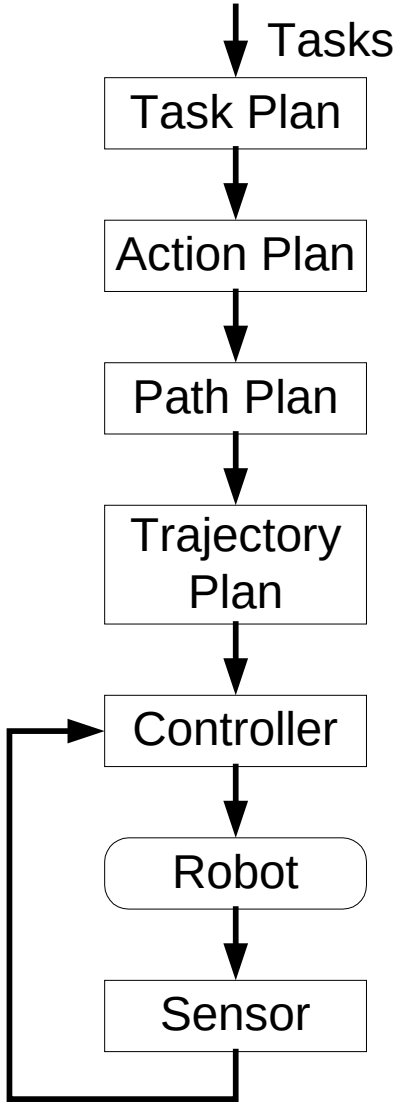
Terminology: path vs. trajectory

Both terms are often confused and used as synonyms informally.

- **Path**: an ordered locus of points in the space (either joint or operational), which the manipulator should follow.
  Path is a pure geometric description of motion.
- **Trajectory**: a path on which timing law is specified, e.g., velocities and accelerations in its each point.

Robot Motion Planning

- Path planning (global):
  - Geometric path.
  - Issues: obstacle avoidance, shortest path.
- Trajectory generating (local):
  - “Interpolate” or “approximate” the desired path by a class of polynomial functions and
  - generate a sequence of time-based “control set points” for the control of manipulator from the initial configuration to its destination.



Trajectory generation, problem formulation

- The aim of the trajectory generation:
  to generate inputs to the motion control system which ensures that the planned trajectory is executed.
- The user or the upper-level planner describes the desired trajectory by some parameters, usually:
  - Initial and final point (point-to-point control).
  - Finite sequence of points along the path (motion through sequence of points).
- Trajectory planning/generation can be performed either in the joint space or operational space.

Minimal requirements

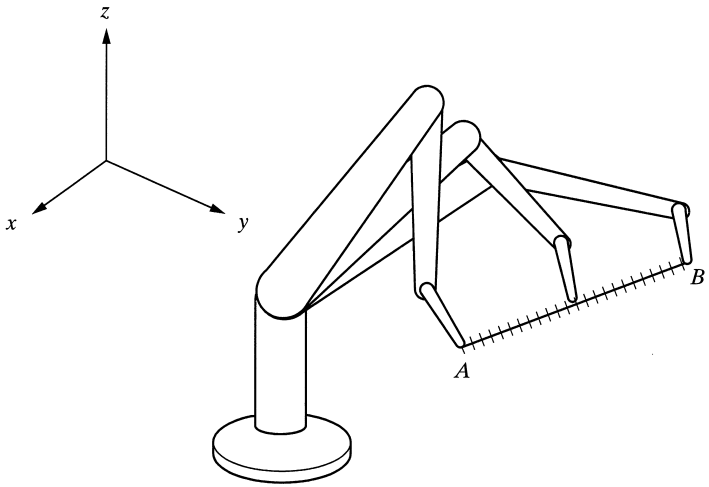
- Capability to move robot arm and its end effector from the initial posture to the final posture.
- Motion laws have to be considered in order not to:
  - violate saturation limits of joint drives.
  - excite the modeled resonant modes of the mechanical structure.
- Device planning algorithm (in the sense of the motion generation) which generates smooth trajectories.

Question: Why should trajectories be smooth?

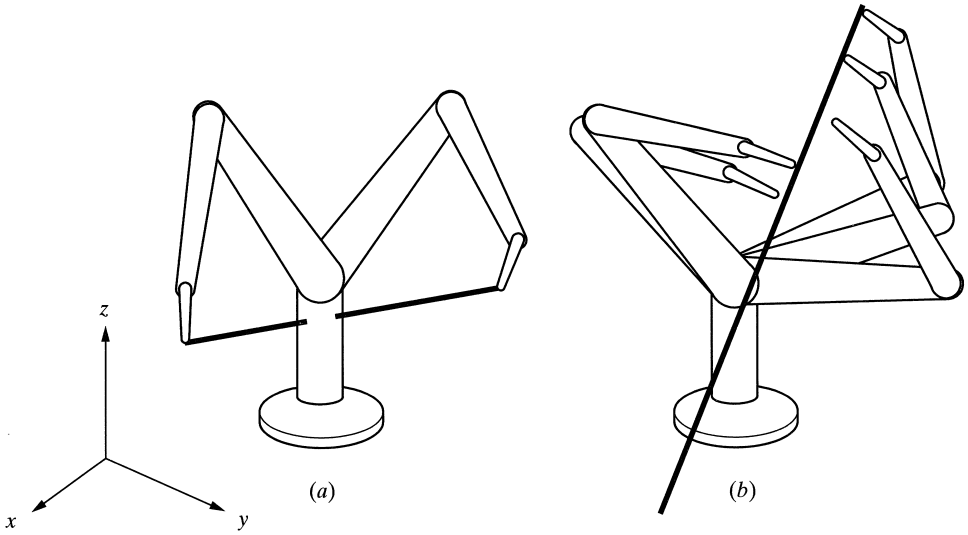
Joint space vs. operational space

- Joint-space description:
  - The description of the motion to be made by the robot by its joint values.
  - The motion between the two points is unpredictable.
- Operational space description:
  - In many cases operational space = Cartesian space.
  - The motion between the two points is known at all times and controllable.
  - It is easy to visualize the trajectory, but it is difficult to ensure that singularity does not occur.

Example
Joint vs. operational space



Sequential motions of a robot to follow a straight line.



Cartesian-space trajectory

(a) The trajectory specified in Cartesian coordinates may force the robot to run into itself, and

(b) the trajectory may requires a sudden change in the joint angles.

Trajectory in the operational space

- Calculate path from the initial point to the final point.
- Assign a total time T_{path} to traverse the path.
- Discretize the points in time and space.
- Blend a continuous time function between these points
- Solve inverse kinematics at each step.

- Advantages
  - Collision free path can be obtained.
- Disadvantages
  - Computationally expensive due to inverse kinematics.
  - It is unknown how to set the total time T_{path} .

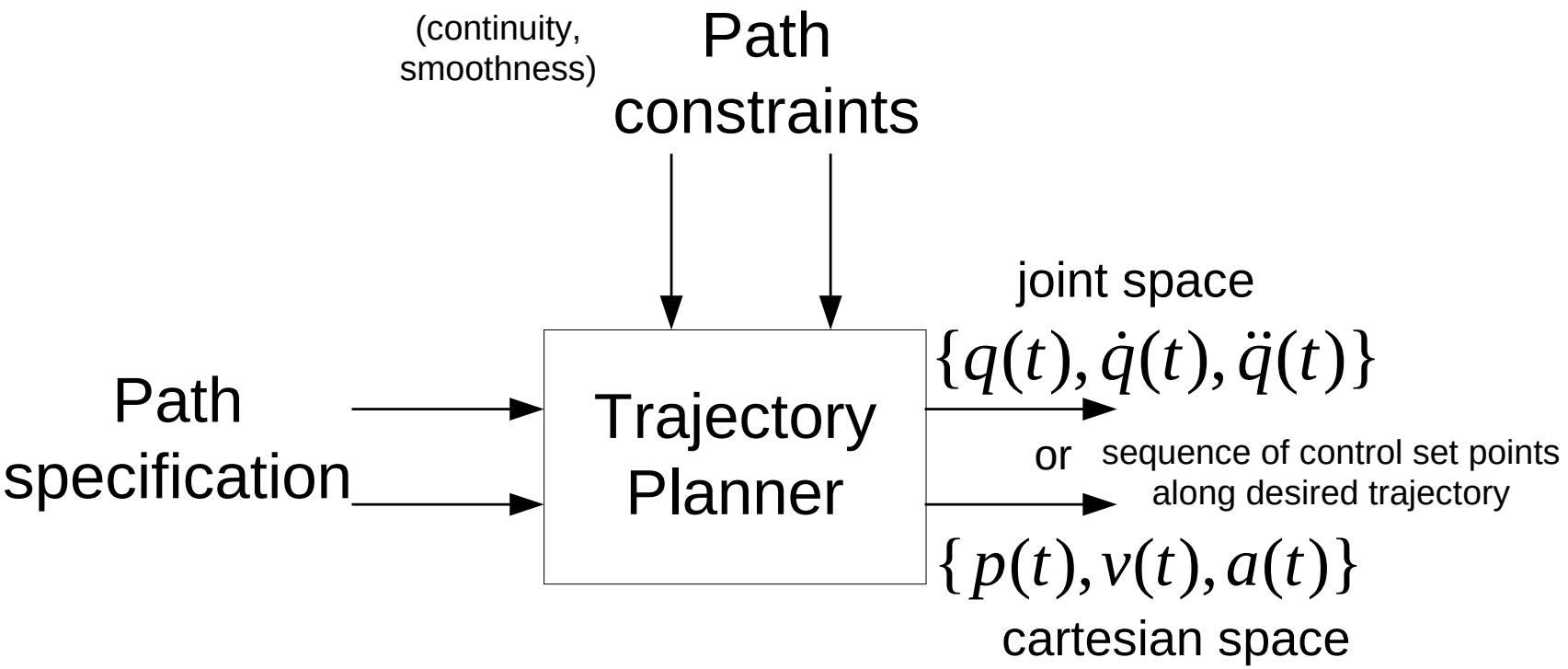
Trajectory in the joint space

- Calculate inverse kinematics solution from initial point to the final point.
- Assign total time T_{path} using maximal velocities in joints.
- Discretize the individual joint trajectories in time.
- Blend a continuous function between these point.

- Advantages
  - Inverse kinematics is computed only once.
  - Can easily take into account joint angle, velocity constraints.

- Disadvantages
  - Cannot deal with operational space obstacles.

Trajectory Planning

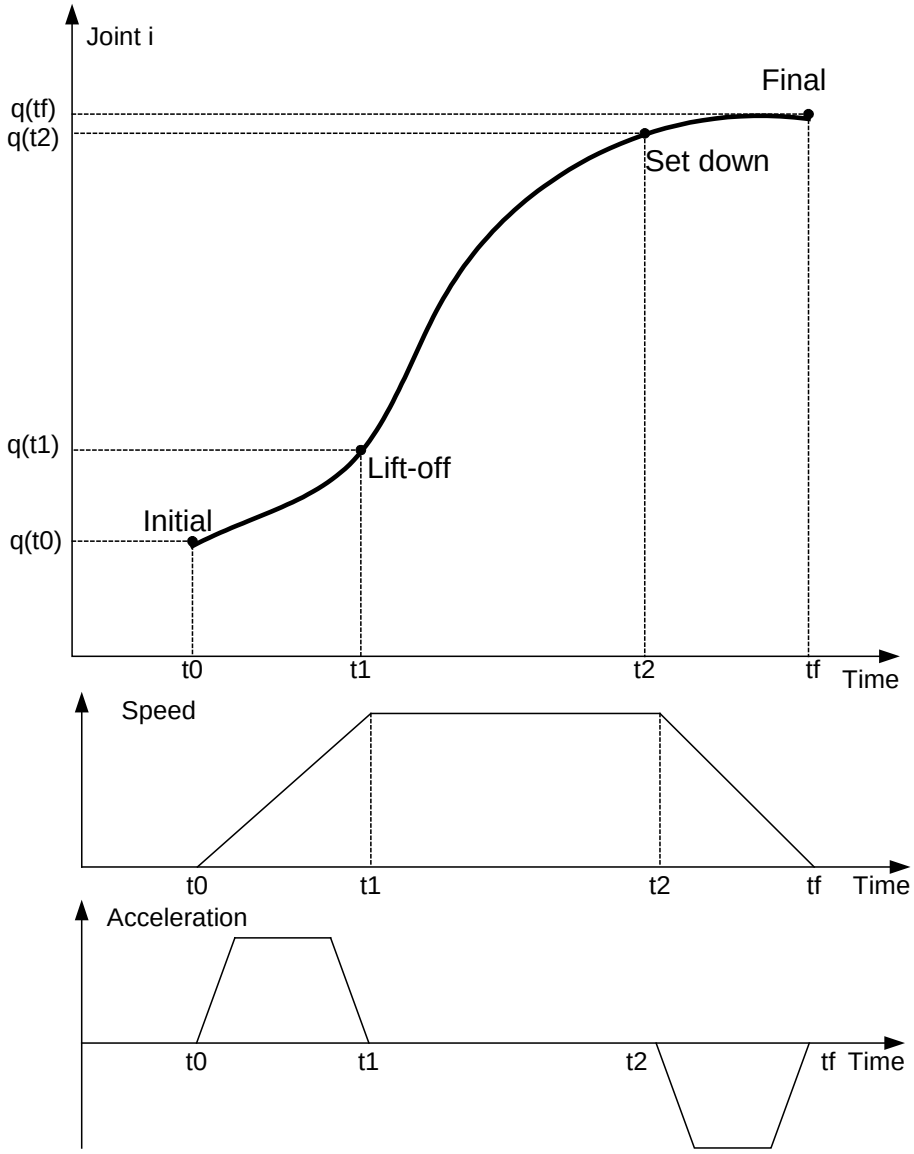


The best planning approach

- Combination of via points (global plan) and point to point (locally between two points).
- Via points provides a course approximation of the path.

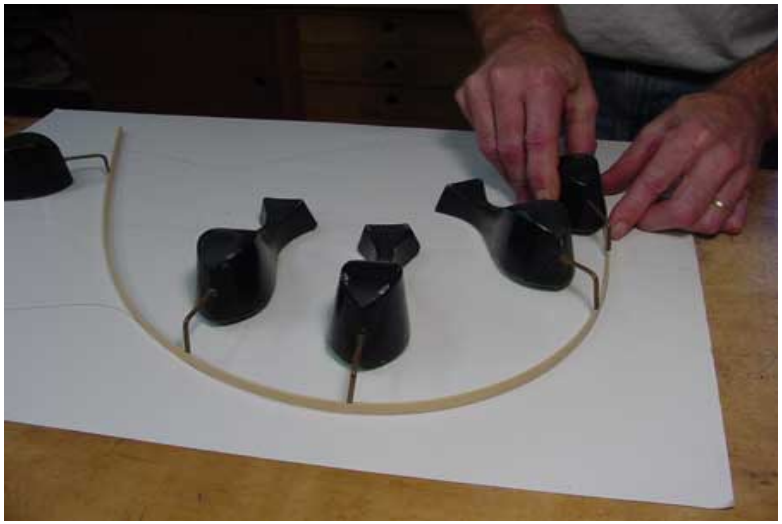
Trajectory planning

- Path Profile
- Velocity Profile
- Acceleration Profile



Curve interpolation, motivation

- Draftsman use ‘ducks’ and strips of wood (splines) to draw curves.
- Wood splines have second-order continuity.
- And pass through the control points.



Function used for interpolation

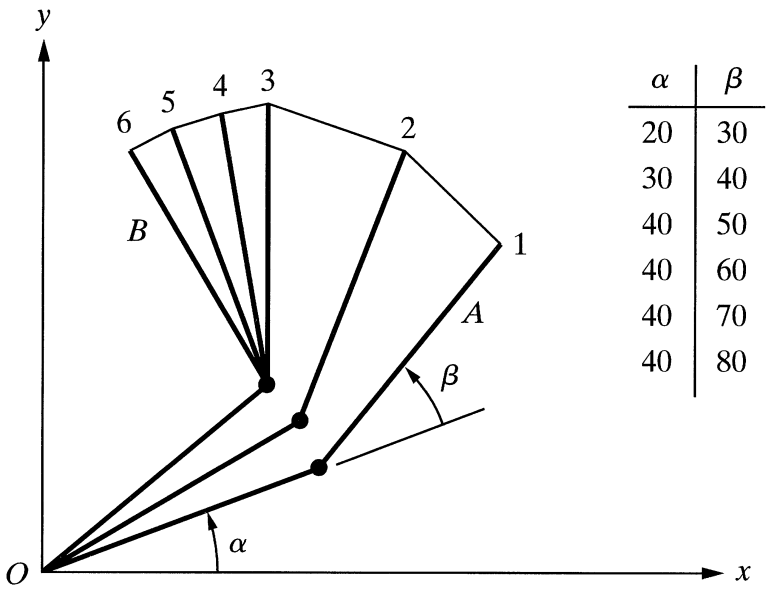
For instance:

- Polynomials of different orders.
- Linear functions with parabolic blends.
- Splines.

Trajectory generation, basics

- Different approaches will be demonstrated on a simple example
- Let us consider a simple 2 degree of freedom robot.
- We desire to move the robot from Point A to Point B.
- Let’s assume that both joints of the robot can move at the maximum rate of 10 degree/sec.

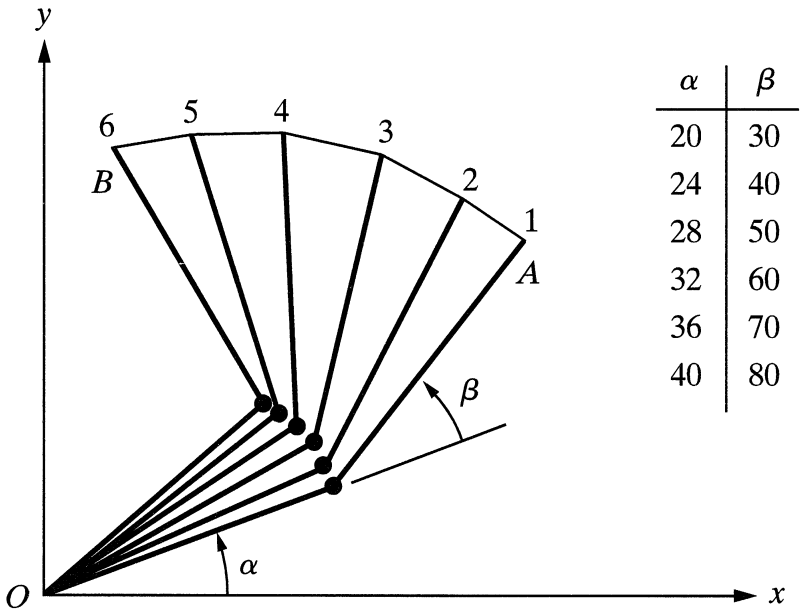
Non-normalized movement



α	β
20	30
30	40
40	50
40	60
40	70
40	80

- Move the robot from A to B, to run both joints at their maximum angular velocities.
- After 2 [sec], the lower link will have finished its motion, while the upper link continues for another 3 [sec].
- The path is irregular and the distances traveled by the robot's end are not uniform.

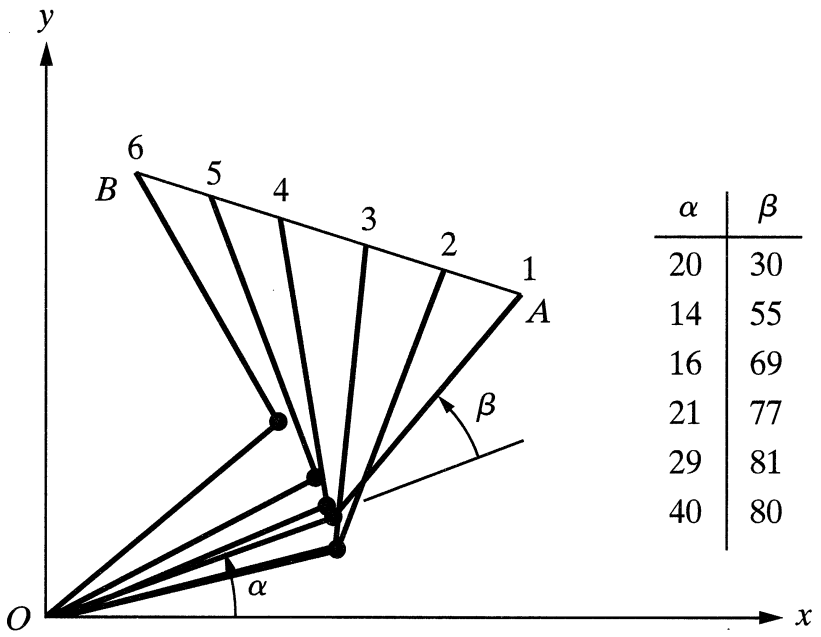
Normalized movement



α	β
20	30
24	40
28	50
32	60
36	70
40	80

- Let's assume that the motions of both joints are normalized by a common factor such that the joint with smaller motion will move proportionally slower and the both joints will start and stop their motion simultaneously.
- Both joints move at different speeds, but move continuously together.
- The resulting trajectory will be different.

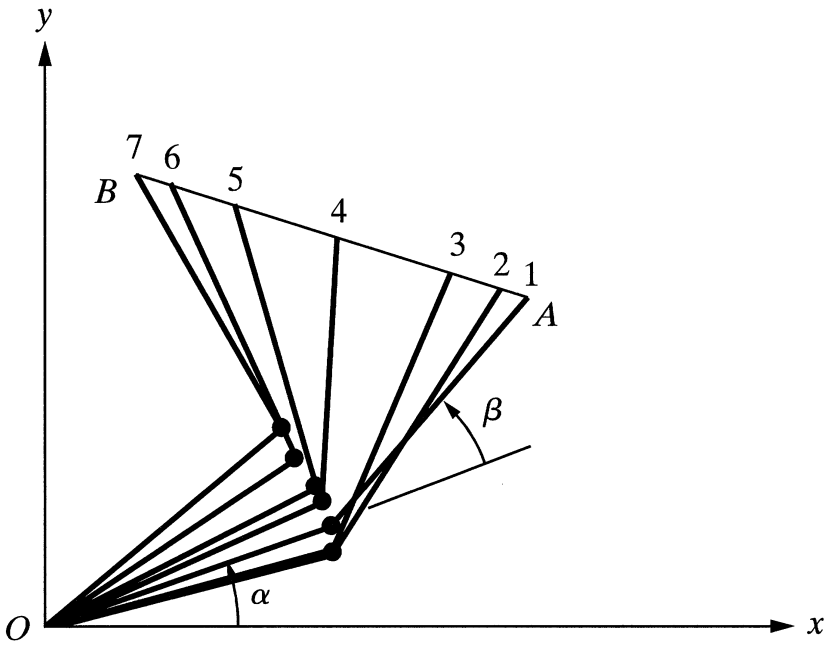
Straight line movement



α	β
20	30
14	55
16	69
21	77
29	81
40	80

- Let us assume that the robot hand follows a straight line between points A and B.
- The simplest way is to draw a line (interpolate) between A, B.
- Divide the line into five segments and solve for necessary angles α and β at each point.
- The joint angles do not change uniformly.

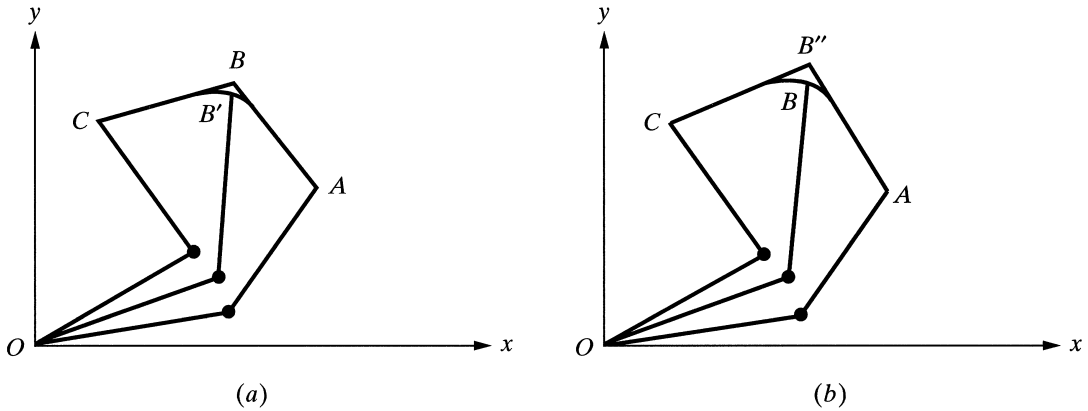
Straight line movement, version 2



- Again interpolation between A, B by a straight line.
- The aim is to accelerate at the beginning and decelerate at the end.
- Divide the segments differently.
  - The arm move at smaller segments as we speed up at the beginning.
  - Go at a constant cruising rate.
  - Decelerate with smaller segments as approaching point B.

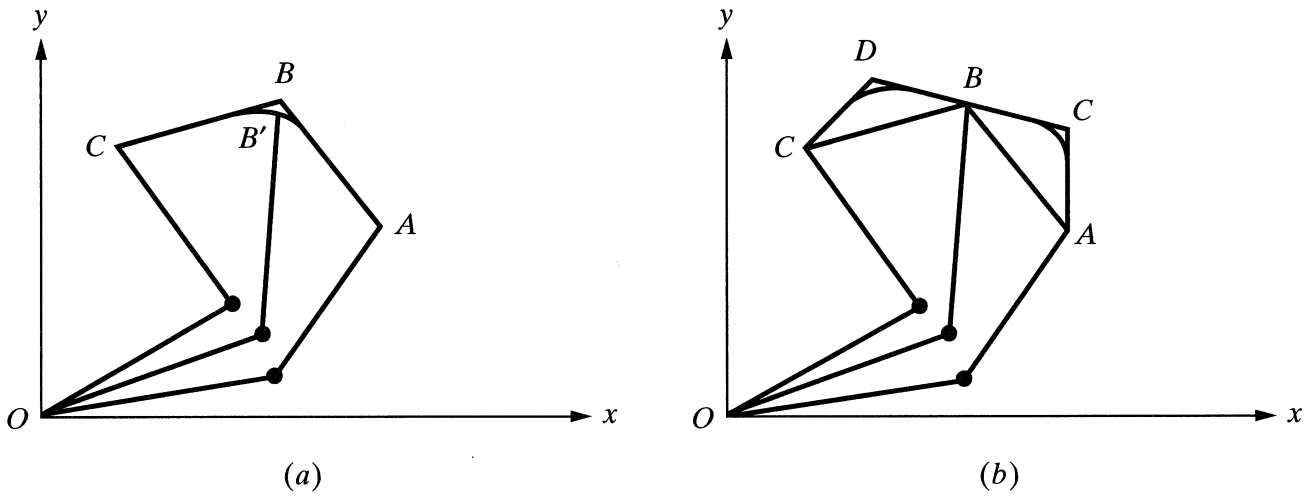
Continuous transition between points

- Stop-and-go motion through the via-point list creates jerky motions with unnecessary stops.
- Solution: take multiple neighboring trajectory into account and enforce constraints on the same tangent and acceleration on the trajectory point.
- How? Blend the two portions of the motion at point B.



Continuous transition between points, option 2

- Alternative scheme ensuring that the trajectory passes through control points.
- Two via points D and E are picked such that point B will fall on the straight-line section of the segment ensuring that the robot will pass through point B.



Interpolation appetizer

- The initial location and orientation of the robot is known.
- Using the inverse kinematic equations, we find the final joint angles for the desired position and orientation.
- Interpolation polynomial $\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$

$$\theta(t_i) = \theta_i$$
$$\theta(t_f) = \theta_f$$
$$\dot{\theta}(t_i) = 0$$
$$\dot{\theta}(t_f) = 0$$

Initial condition

$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2$$

The first derivative of the interpolating polynomial

$$\theta(t_i) = c_0 = \theta_i$$
$$\theta(t_f) = c_0 + c_1 t_f + c_2 t_f^2 + c_3 t_f^3$$
$$\dot{\theta}(t_i) = c_1 = 0$$
$$\dot{\theta}(t_f) = c_1 + 2c_2 t_f + 3c_3 t_f^2 = 0$$

Substituting the initial and final conditions

Building a 'Path Polynomial' Motion Set

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

$$\dot{q}(t) = \frac{dq}{dt} = a_1 + 2a_2 t + 3a_3 t^2 + \cdots$$

$$\ddot{q}(t) = \frac{d^2 q}{dt^2} = 2a_2 + 6a_3 t + \cdots$$

These are the 'trajectory' equations for a joint (Position, Velocity and Acceleration)

Solving the 'Path Polynomial' implies finding a_i's for specific paths

- We would have "boundary" conditions for position and velocity at both ends of the path.
- We would have the desired total time of travel.
- Using these conditions we can solve for a_0, a_1, a_2 and a_3 to build a 3rd order path polynomial for the required motion.

Solving the 'Path Polynomial' is a matter of finding a_i's for specific paths

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3$$

$$\dot{q}_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2$$

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3$$

$$\dot{q}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2$$

'Poly's' holding at starting time and position

'Poly's' holding at ending time and position

Solving the 'Path Polynomial' is a matter of finding a_i's for specific paths

- Writing these as Matrix Forms:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \bullet \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ q_f \\ \dot{q}_f \end{bmatrix}$$

Solving the 'Path Polynomial' is a matter of finding a_i's for specific paths

- If we set $t_0 = 0$ (starting time is when we start counting motion!) then:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \bullet \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ q_f \\ \dot{q}_f \end{bmatrix}$$

By examination, $a_0 = q_0$ & $a_1 = q_0$(dot)

Solving the 'Path Polynomial' is a matter of finding a_i's for specific paths

- Completing the solution consists of forming relationships for: a_2 & a_3
- Done by substituting a_0 & a_1 values and solving the last two equation simultaneously:

Be Careful and note the order of the positions and velocities!

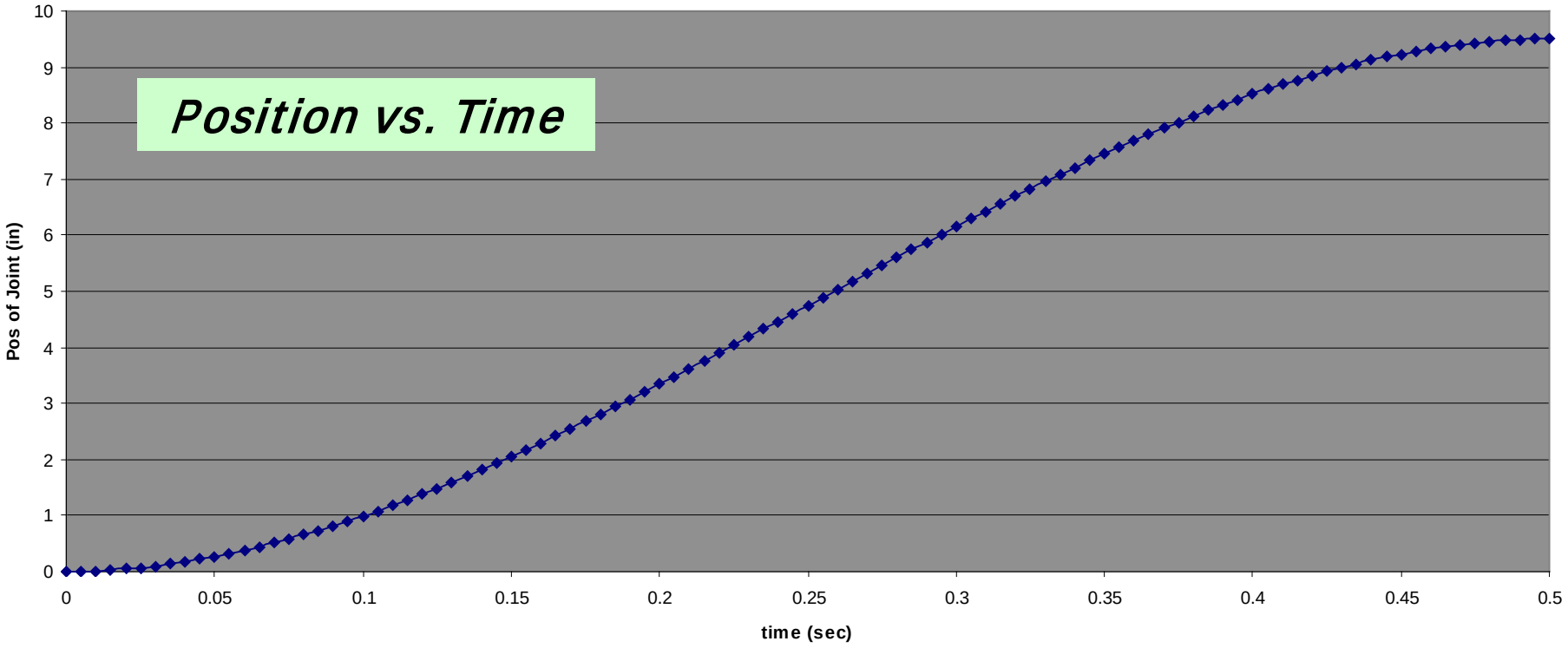
$$a_2 = \left(3\left(q_f - q_0\right) - t_f\left(2\dot{q}_0 + \dot{q}_f\right)\right) \Big/ t_f^2$$

$$a_3 = \left(2\left(q_0 - q_f\right) + t_f\left(\dot{q}_f + \dot{q}_0\right)\right) \Big/ t_f^3$$

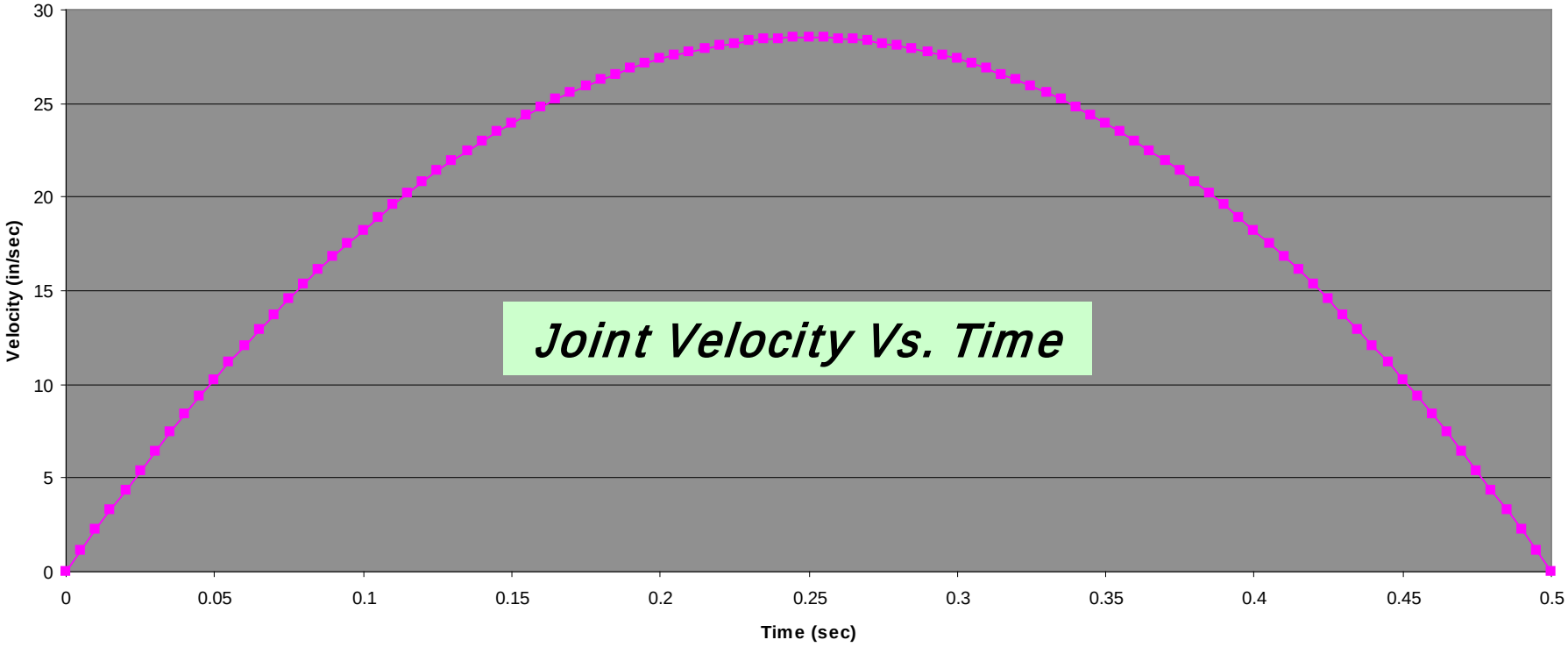
Applying it to the Coffee ‘Bot

- Start: X = 0; v = 0 @ time = 0
- End: X = 9.5”; v = 0 @ time = .5 sec
- a0 = 0 ; a1 = 0
- a2 = (3 * 9.5)/(0.5^2) = 114
- a3 = (2 *(- 9.5))/(0.5^3) = -152

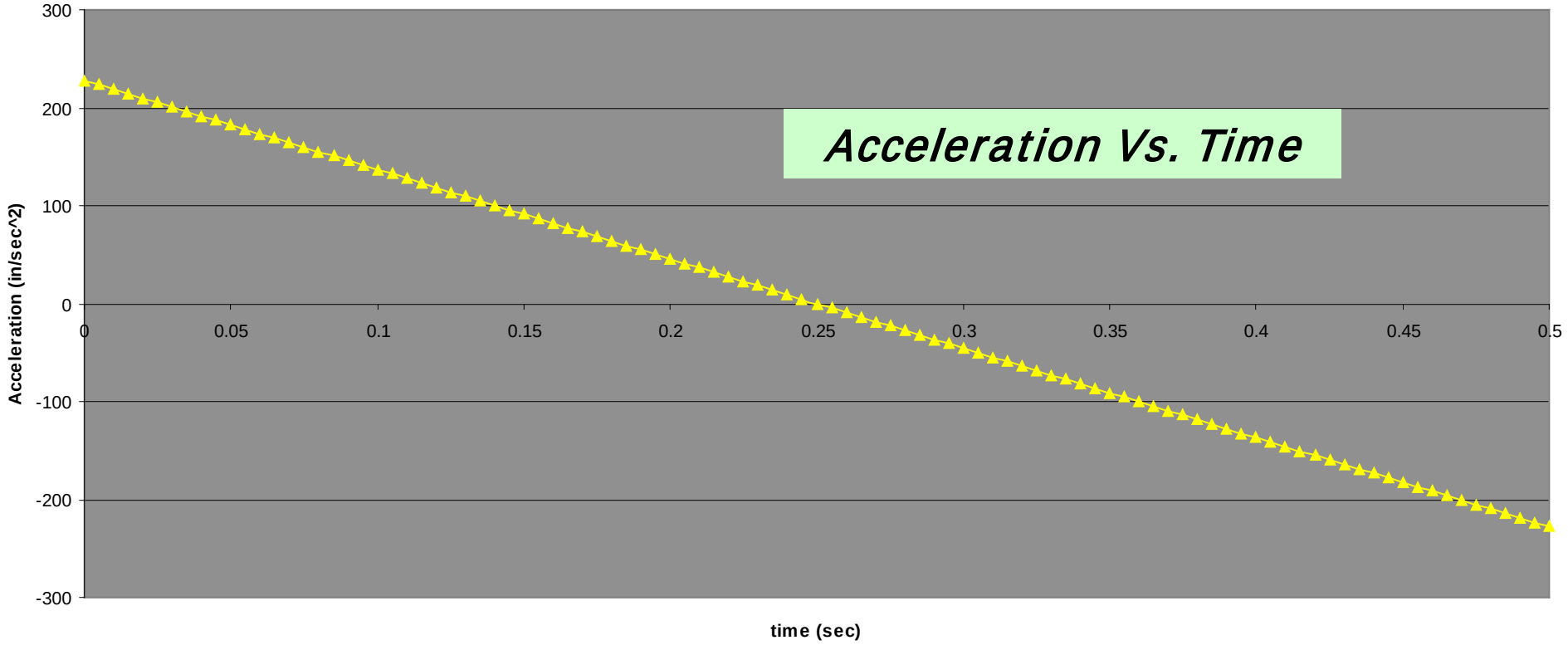
Applying it to the Coffee ‘Bot: Position



Applying it to the Coffee 'Bot: Velocity



Applying it to the Coffee 'Bot: Acceleration



The boundary conditions

1) Initial position
2) Initial velocity
3) Initial acceleration
4) Lift-off position
5) Continuity in position at t_1
6) Continuity in velocity at t_1
7) Continuity in acceleration at t_1
8) Set-down position
9) Continuity in position at t_2
10) Continuity in velocity at t_2
11) Continuity in acceleration at t_2
12) Final position
13) Final velocity
14) Final acceleration

Requirements

- Initial Position
  - Position (given)
  - Velocity (given, normally zero)
  - Acceleration (given, normally zero)
- Final Position
  - Position (given)
  - Velocity (given, normally zero)
  - Acceleration (given, normally zero)

Requirements

Intermediate positions

- set-down position (given)
- set-down position (continuous with previous trajectory segment)
- Velocity (continuous with previous trajectory segment)
- Acceleration (continuous with previous trajectory segment)

Requirements

Intermediate positions

- Lift-off position (given)
- Lift-off position (continuous with previous trajectory segment)
- Velocity (continuous with previous trajectory segment)
- Acceleration (continuous with previous trajectory segment)

Trajectory Planning

- n-th order polynomial, must satisfy 14 conditions,
- 13-th order polynomial

$$a_{13}t^{13} + \cdots + a_2 t^2 + a_1 t + a_0 = 0$$

- 4-3-4 trajectory

$$h_1(t) = a_{14}t^4 + a_{13}t^3 + a_{12}t^2 + a_{12}t + a_{10} \qquad t_0 \rightarrow t_1, \text{ 5 unknowns}$$

$$h_2(t) = a_{23}t^3 + a_{22}t^2 + a_{21}t + a_{20} \qquad t_1 \rightarrow t_2, \text{ 4 unknowns}$$

$$h_n(t) = a_{n4}t^4 + a_{n3}t^3 + a_{n2}t^2 + a_{n2}t + a_{n0} \qquad t_2 \rightarrow t_f, \text{ 5 unknowns}$$

- 3-5-3 trajectory

References

- B. Siciliano et al. Robotics (Modeling, planning and control), Springer, Berlin, 2009, chapter 4: Trajectory planning, pages 161-189.