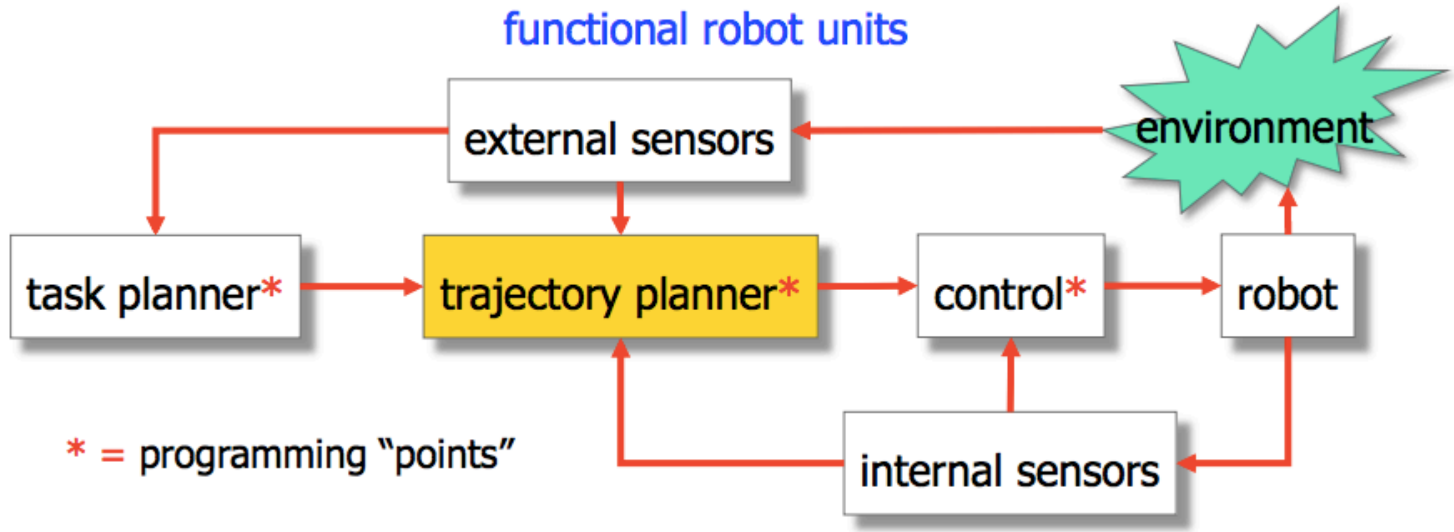
CMPUT 312
Intro Robotics & Mechatronics:

Trajectory Planning

Control

Slides from Prof. A. De Luca - lecture notes (Robotics 1)

functional robot units



1. define Cartesian pose points (position+orientation) using the teach-box
2. program an (average) velocity between these points, as a 0-100% of a maximum system value (different for Cartesian- and joint-space motion)
3. linear interpolation in the joint space between points sampled from the built trajectory

From Task to Trajectory

TRAJECTORY — of motion $p_d(t)$ / of interaction $F_d(t)$

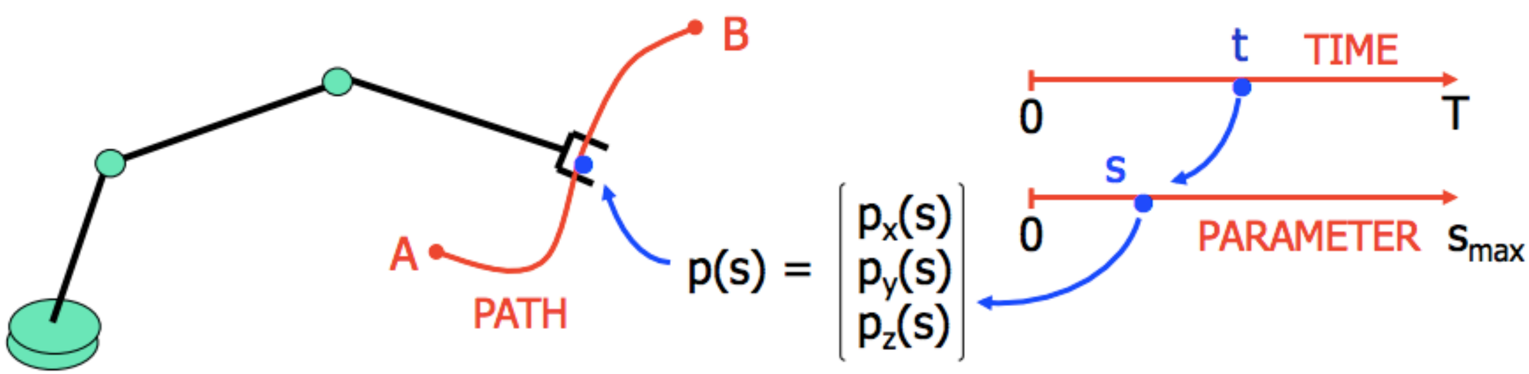
||

GEOMETRIC PATH — parameterized by s: $p=p(s)$ (e.g., s is the arc length)

+

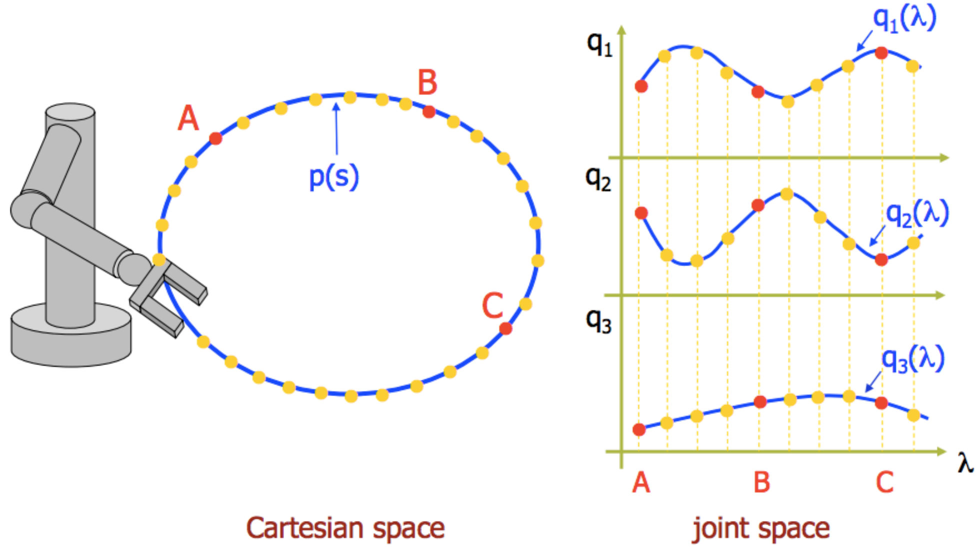
TIMING LAW — describes the time evolution of $s=s(t)$

$\}\ p(s(t))$



example: TASK planner provides A, B
TRAJECTORY planner generates p(t)

Example



TASK planning

①
- sequence of pose points ("knots") in Cartesian space

  interpolation in Cartesian space

- Cartesian geometric path (position + orientation): $p = p(s)$

  path sampling and kinematic inversion

②
- sequence of "knots" in joint space

  interpolation in joint space

- geometric path in joint space: $q = q(\lambda)$

analytic inversion

Cartesian vs. Joint 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

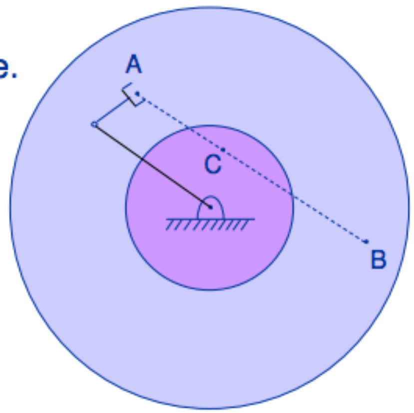
  (for orientation : equivalent axes, Euler angles,…
- More expensive at run time

  (after the path is calculated need joint angles
  in a lot of points)
- Discontinuity problems

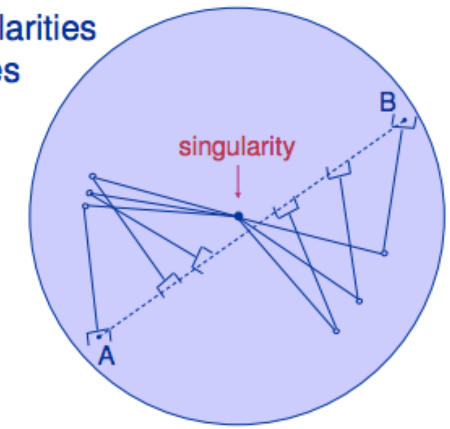
Difficulties in Cartesian Space

Initial and Goal
Points are reachable.

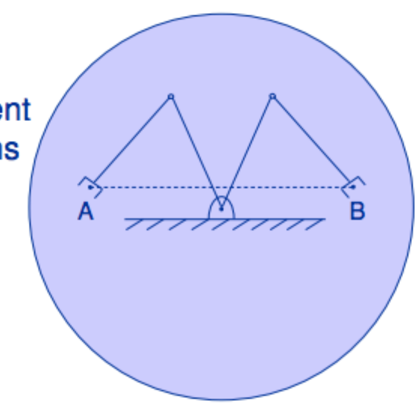
Intermediate points
(C) unreachable.



Approaching singularities
some joint velocities
go to ∞
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.)



Trajectory Planning in any space:

Assume one <u>generic</u> variable <u>u</u>

(can be x, y, z, orientation - α, β, γ)

joint variables direction cosines

<u>Candidate curves</u> :

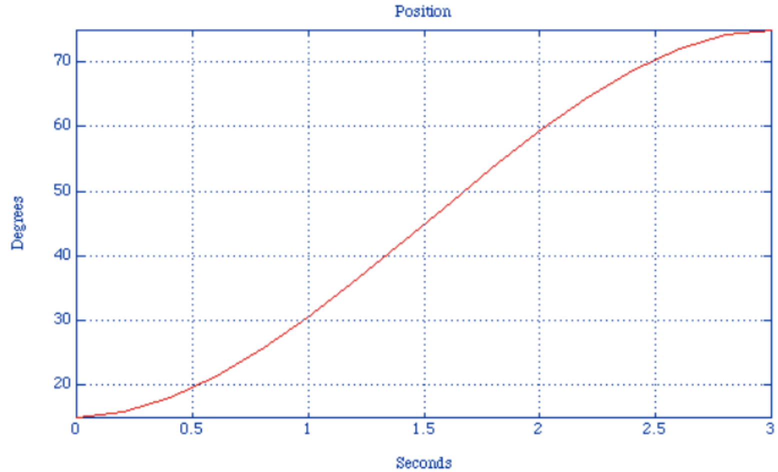
straight line (discontinuous velocity at path points)

straight line with blends

cubic polynomials (splines)

higher order polynomials (quintic,…) or other curves

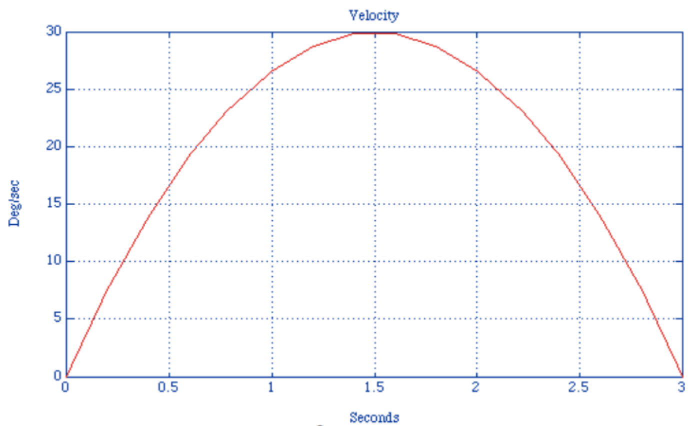
Cubic Polynomial



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

Initial Conditions:

$$u(0)=u_0 \; ; \; \mathrm{u}(t_f)=u_f$$

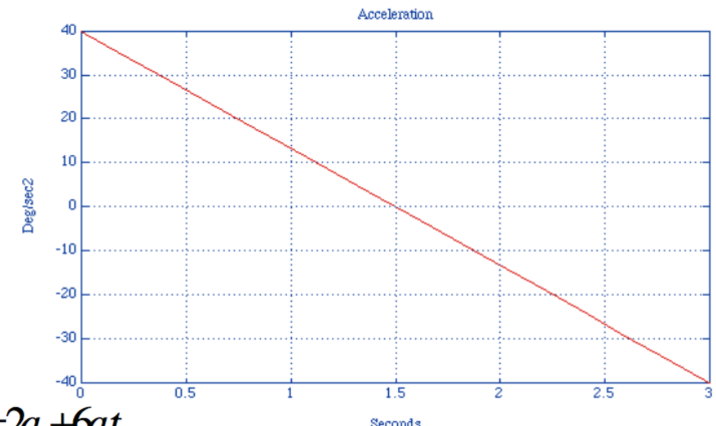


$$\dot{u}(t)=a_1+2a_2t+3a_3t^2$$

Initial Conditions:

$$\dot{u}(0)=0 \, ; \; \dot{u}(t_f)=0$$

Starts and ends at rest



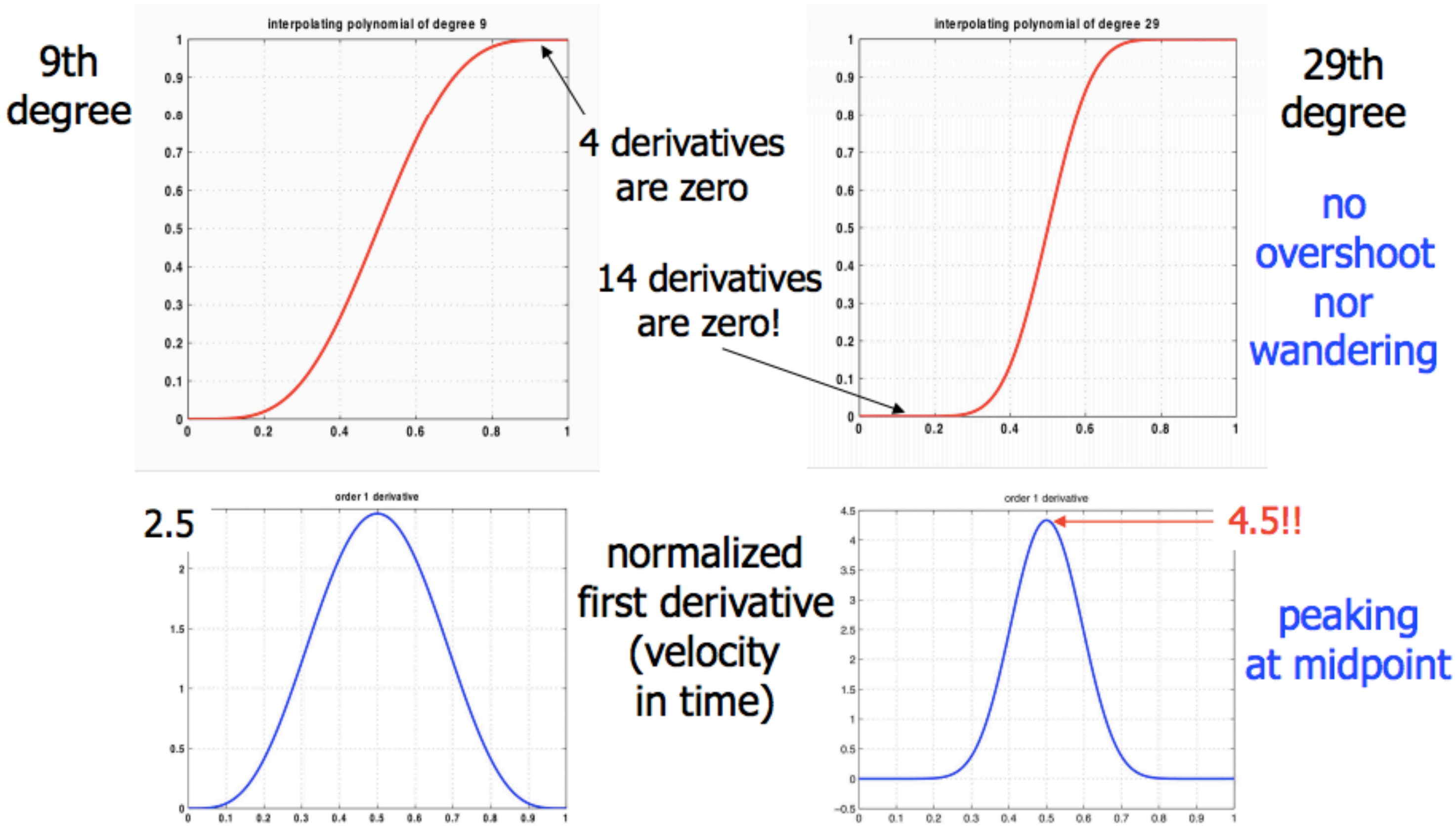
$$\ddot{u}(t)=2a_2+6a_3t$$

$$\dddot{u}(t)=6a_3$$ (constant)

Solution :

$$u(t)=u_0+\frac{3}{t_f^2}(u_f-u_0)t^2+\left(-\frac{2}{t_f^3}\right)(u_f-u_0)t^3$$

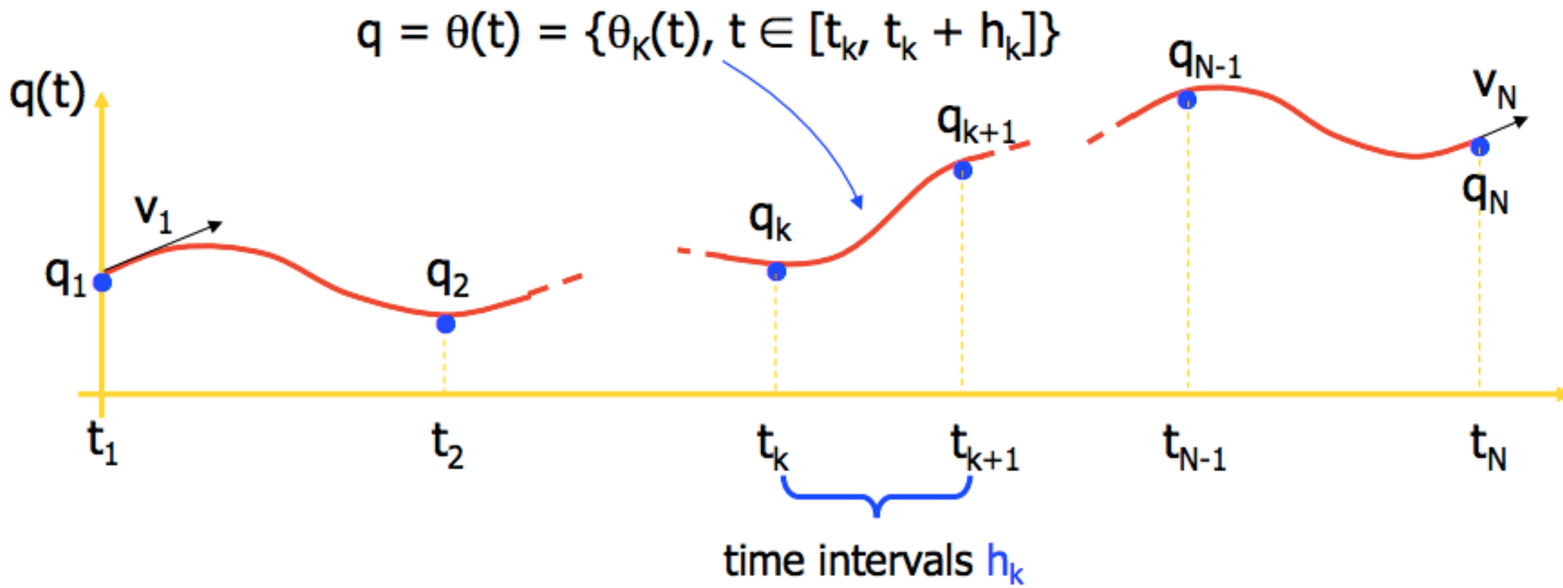
- 3rd order >> still non-zero accelerations
- use higher order polynomials (**5th** , 7th , ...)



- higher order polynomials >> computationally more expensive
- hard to handle **N** way-points.
- oscillations arise out of the interpolation points (**wandering**)

practical solution >> spline

Problem: interpolate N knots, with continuity up to the second derivative ?



- solution

  spline: N-1 cubic polynomials, concatenated so as to pass through N knots and being continuous up to the second derivative at the N-2 internal knots

$$\theta_K(\tau) = a_{k0} + a_{k1}\,\tau + a_{k2}\,\tau^2 + a_{k3}\,\tau^3 \qquad \tau \in [0, h_k],\ \tau = t - t_k \quad (k = 1, \ldots, N-1)$$

continuity conditions for velocity and acceleration

$$\dot{\theta}_K(h_k) = \dot{\theta}_{K+1}(0) \qquad \ddot{\theta}_K(h_k) = \ddot{\theta}_{K+1}(0) \qquad k = 1, \ldots, N-2$$

Algorithmic Implementation

1. if all velocities v_k at internal knots were known, then each cubic in the spline would be uniquely determined by

$$\theta_K(0) = q_K = a_{K0}$$
$$\dot{\theta}_K(0) = v_K = a_{K1}$$

$$\begin{pmatrix} h_K^2 & h_K^3 \\ 2h_K & 3h_K^2 \end{pmatrix} \begin{pmatrix} a_{K2} \\ a_{K3} \end{pmatrix} = \begin{pmatrix} q_{K+1} - q_K - v_K h_K \\ v_{K+1} - v_K \end{pmatrix} \quad (1)$$

2. impose the continuity for accelerations (N-2 conditions)

$$\ddot{\theta}_K(h_k) = 2\, a_{K2} + 6\, a_{K3}\, h_K = \ddot{\theta}_{K+1}(0) = 2\, a_{K+1,2}$$

3. expressing the coefficients a_{k2}, a_{k3}, $a_{k+1,2}$ in terms of the still unknown knot velocities (see step 1.) yields a linear system of equations that is always (easily) solvable

$$\begin{pmatrix} & & \\ & A(h) & \\ & & \end{pmatrix} \begin{pmatrix} v_2 \\ v_3 \\ \vdots \\ v_{N-1} \end{pmatrix} = \begin{pmatrix} b(h,q,v_1,v_N) \\ \\ \\ \end{pmatrix}$$

tri-diagonal matrix always invertible

unknown

known vector

to be substituted then back in (1)

$$
A(h)\begin{pmatrix} v_2 \\ v_3 \\ \vdots \\ v_{N-1} \end{pmatrix} = \begin{pmatrix} b(h,q,v_1,v_N) \end{pmatrix}
$$

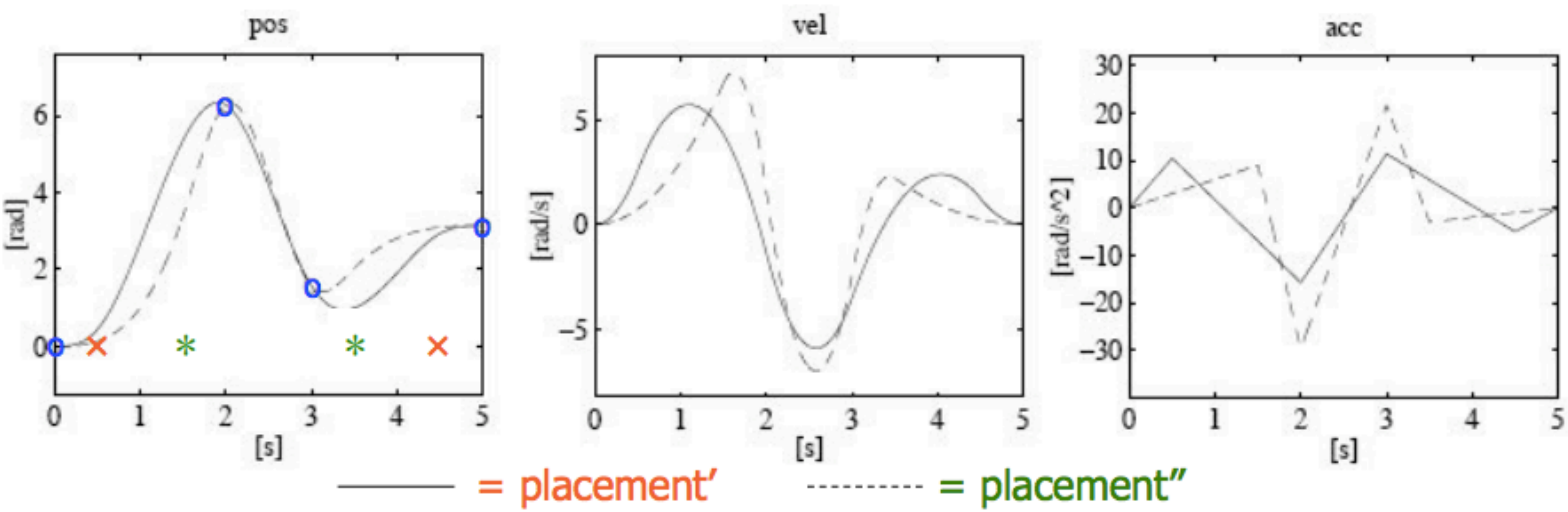
$$
A(h) = \begin{pmatrix}
2(h_1+h_2) & h_1 & & & \\
h_3 & 2(h_2+h_3) & h_2 & & \\
 & & \ddots & & \\
 & & h_{N-2} & 2(h_{N-3}+h_{N-2}) & h_{N-3} \\
 & & & h_{N-1} & 2(h_{N-2}+h_{N-1})
\end{pmatrix}
\qquad
b = \begin{pmatrix}
\frac{3}{h_1 h_2}[h_1^2(q_3 - q_2) + h_2^2(q_2 - q_1)] - h_2 v_1 \\
\frac{3}{h_2 h_3}[h_2^2(q_4 - q_3) + h_3^2(q_3 - q_2)] \\
\vdots \\
\frac{3}{h_{N-3} h_{N-2}}[h_{N-3}^2(q_{N-1} - q_{N-2}) + h_{N-2}^2(q_{N-2} - q_{N-3})] \\
\frac{3}{h_{N-2} h_{N-1}}[h_{N-2}^2(q_N - q_{N-1}) + h_{N-1}^2(q_{N-1} - q_{N-2})] - h_{N-2} v_N
\end{pmatrix}
$$

diagonally dominant matrix (for $h_k > 0$)

- a spline is uniquely determined from the set of data $q_1,\ldots,q_N$, $h_1, \ldots, h_{N-1}$, v_1, v_N
- in time, the total motion occurs in $T = \Sigma_k\, h_k = t_N - t_1$
- the time intervals h_k can be chosen so as to minimize T (linear objective function) under (nonlinear) bounds on velocity and acceleration in [0,T]
- in time, the spline construction can be suitably modified when the acceleration is also assigned at the initial and final knots

Example

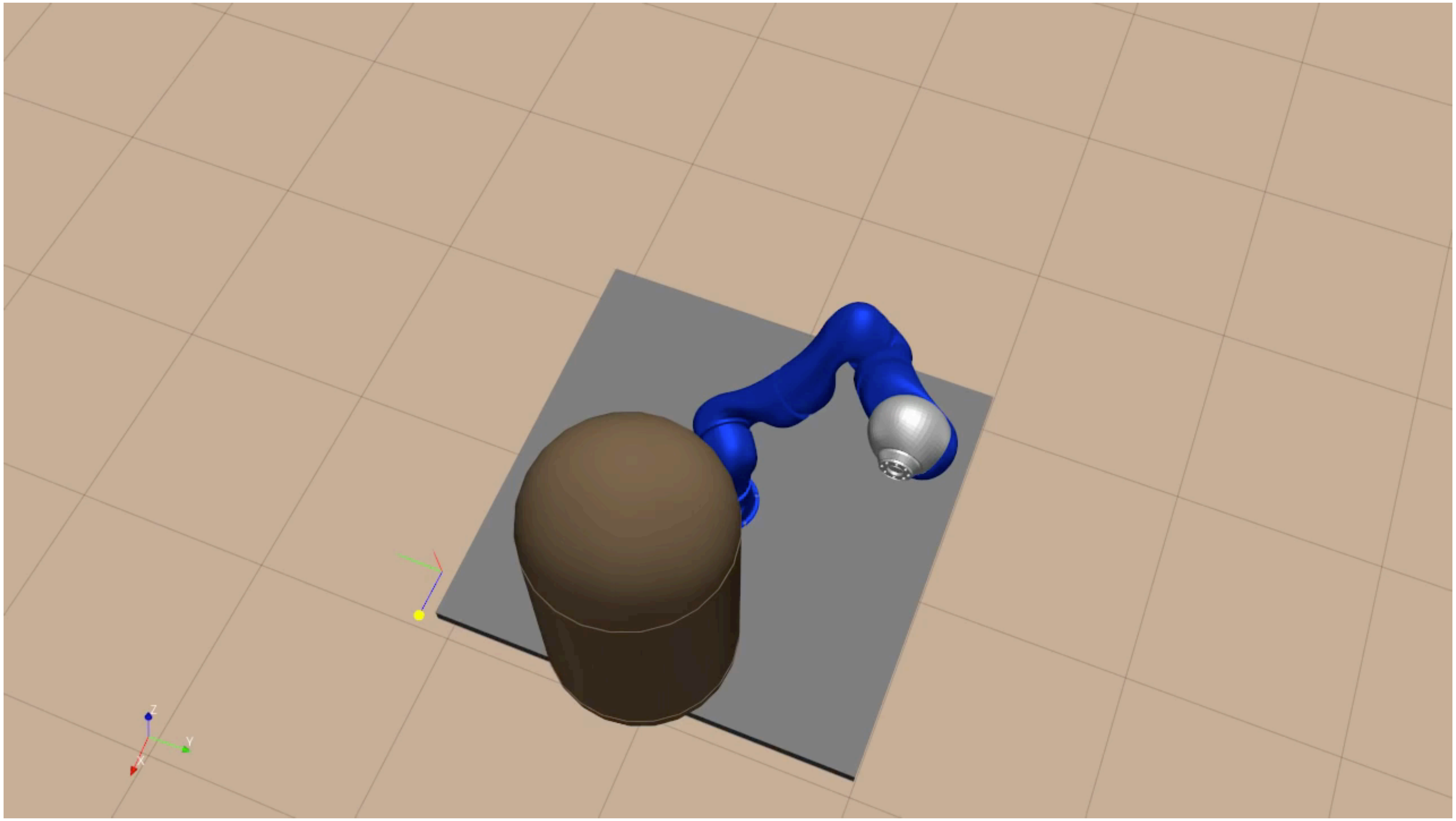
- $N = 4$ knots (3 cubic polynomials)
  - joint values $q_1 = 0$, $q_2 = 2\pi$, $q_3 = \pi/2$, $q_4 = \pi$
  - at $t_1 = 0$, $t_2 = 2$, $t_3 = 3$, $t_4 = 5$ (thus, $h_1 = 2$, $h_2 = 1$, $h_3 = 2$)
  - boundary velocities $v_1 = v_4 = 0$
- 2 added knots to impose accelerations at both ends (5 cubic polynomials)
  - boundary accelerations $\alpha_1 = \alpha_4 = 0$
  - two placements: at $t_1' = 0.5$ and $t_4' = 4.5$ (×), or $t_1'' = 1.5$ and $t_4'' = 3.5$ (*)



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

Obstacle Avoidance



Learning from demonstration