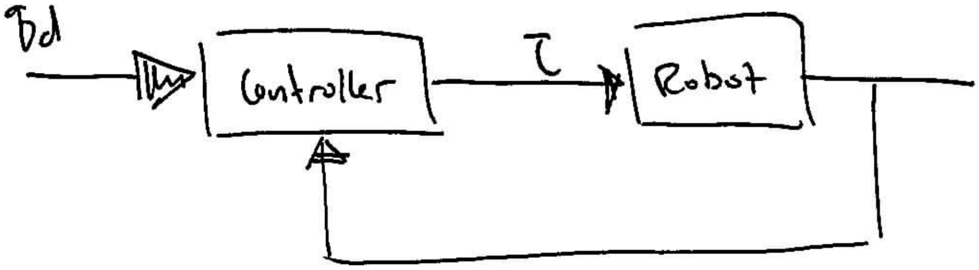
Lecture 14   11/15/2021   ME 221

Last time:



$$\tau = K_p e + K_I \int e\,dt + K_d \dot{e}$$

$$e = q_d - q$$

Today:

- few remarks on anonymous functions
- Trajectory generation
  - Joint space trajectories
  - Cartesian space

L & P CH. 9
JAZAR CH. 13

Anonymous functions:

- Why?

→ convenience!

- the "code" doesn't care if it's anonymous or whether a regular functions

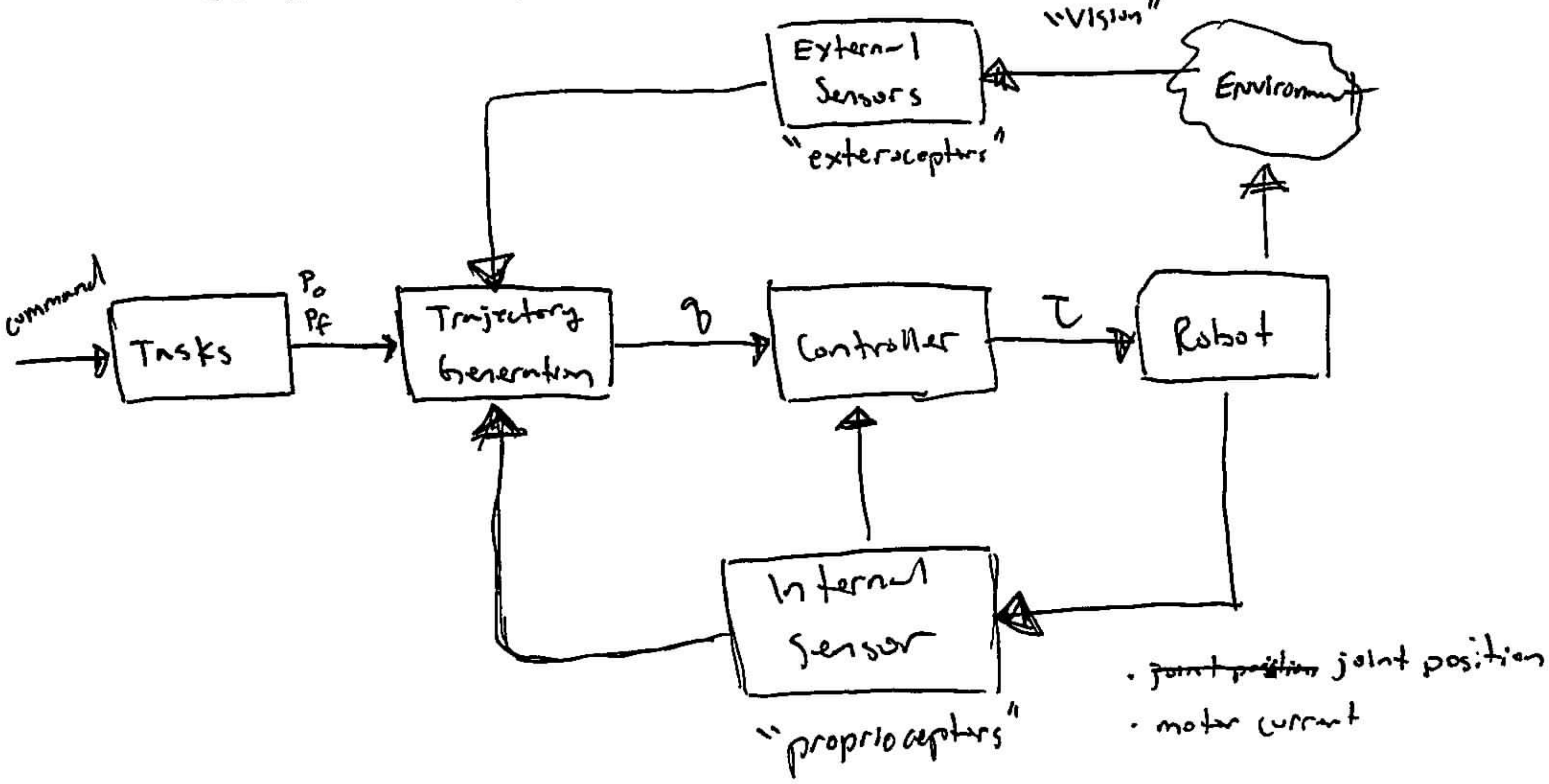
P_3 = some number

myfunc =@(P_1, P_2) $P_1 + P_2 \cdot P_3$

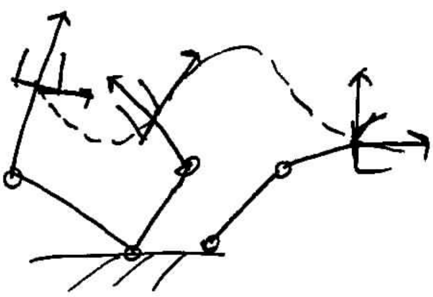
myfunc =@ (P_1, P_2, P_3) $P_1 + P_2 \cdot P_3$

Trajectory Generations

overview "conceptual" Robot Control



Objective of Trajectory Generation



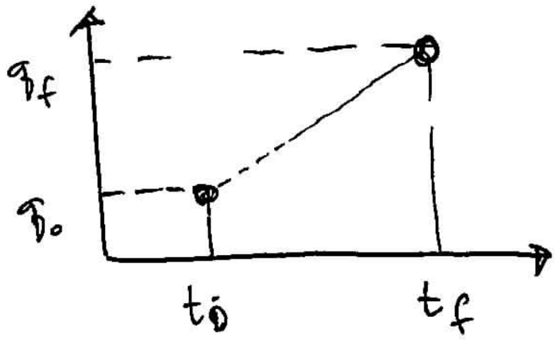
- Design the ~~movent a~~ movement of the robot
- can be @ joint level or @ the endeffector level (cartesian)

- mostcommon

  → polynomials to parameterize trajectory

Example

first order, (line)

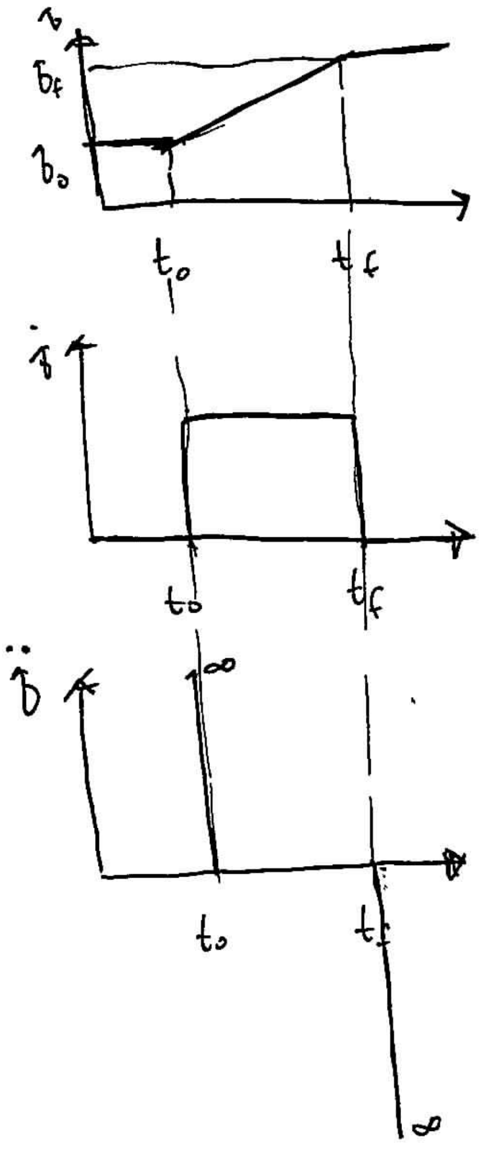


$$q(t) = a_0 + a_1 t$$

$$q_0 = a_0 + a_1 t_0$$
$$q_f = a_0 + a_1 t_f$$

$$\Rightarrow \underbrace{\begin{bmatrix} q_0 \\ q_f \end{bmatrix}}_{\text{Known}} = \underbrace{\begin{bmatrix} 1 & t_0 \\ 1 & t_f \end{bmatrix}}_{\text{Known}} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

$$b = Ax \Rightarrow \boxed{x = A^{-1} b}$$

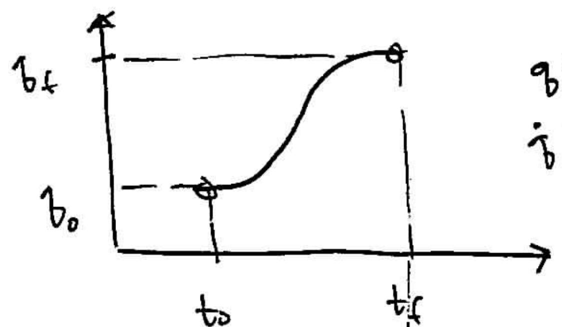


Linear Example

Example:

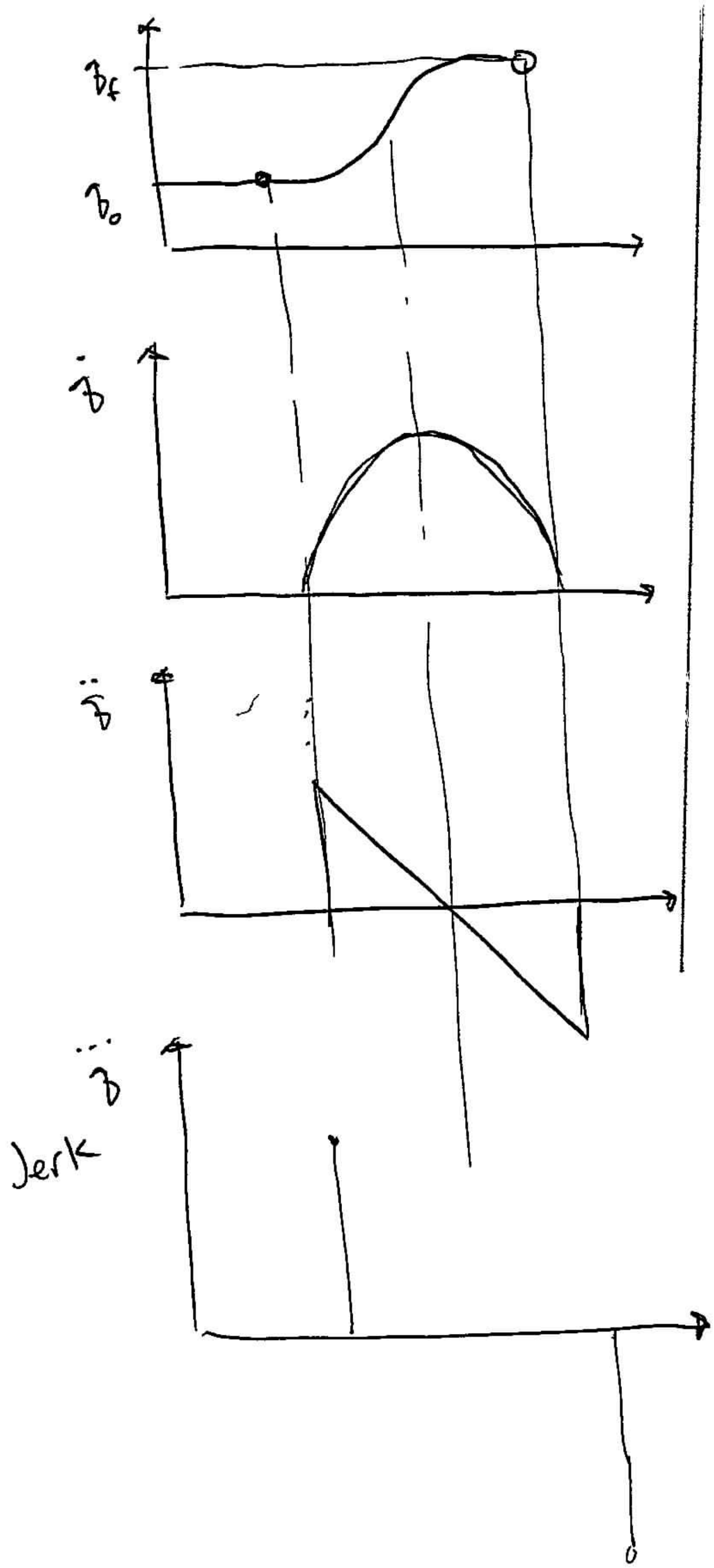
Cubic

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2$$



$$q(t_0) = q_0$$
$$\dot{q}(t_0) = \dot{q}_0$$
$$q(t_f) = q_f$$
$$\dot{q}(t_f) = \dot{q}_f$$

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


z_f
z_0
$\dot{z}$
$\ddot{z}$
$\dddot{z}$
Jerk

Example

Quintic polynomial : $q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$

$$q(t_0) = q_0 \qquad q(t_f) = q_f$$
$$\dot{q}(t_0) = \dot{q}_0 \qquad \dot{q}(t_f) = \dot{q}_f$$
$$\ddot{q}(t_0) = \ddot{q}_0 \qquad \ddot{q}(t_f) = \ddot{q}_f$$

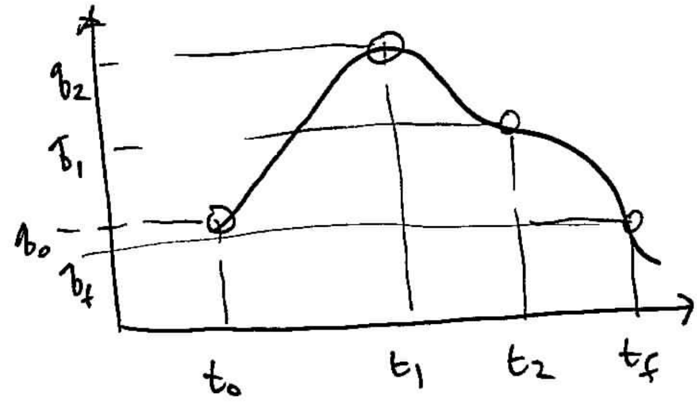
$$b = A x \rightarrow x = A^{-1} b$$

$$\begin{bmatrix} q_0 \\ \dot{q}_0 \\ \ddot{q}_0 \\ q_f \\ \dot{q}_f \\ \ddot{q}_f \end{bmatrix} = A \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}$$

- The number of required conditions determines the degree of the polynomial

- types conditions
  - positions along a desired trajectory (waypoints, vias)
  - velocity, acceleration, jerk at two point so that smoothness can be controlled

Example : way points



$q(t_0) = q_0 \quad \dot{q}(t_0) = \dot{q}_0 \quad \ddot{q}(t_0) = \ddot{q}_0$

$q(t_f) = q_f \quad \dot{q}(t_f) = \dot{q}_f \quad \ddot{q}(t_f) = \ddot{q}_f$

$q(t_1) = q_1$

$q(t_2) \quad q_2$

8 constraints $\longrightarrow$ 7^{th} polynomial.

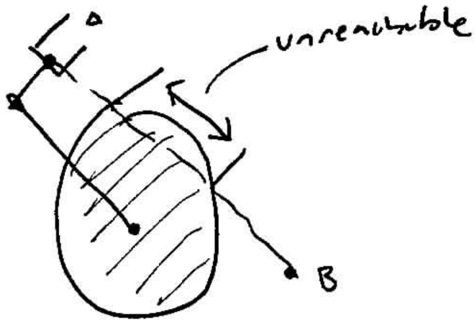
$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 + a_7 t^7$$

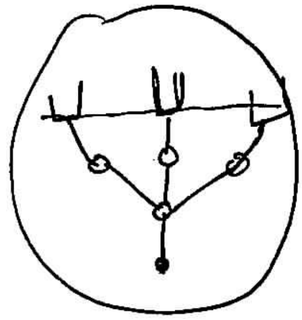
$\dot{q}(t) \cdots$

$\ddot{q}(t) \cdots$

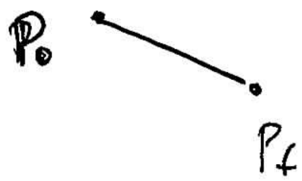
$$\begin{array}{r} \\ \dot{q}_0 = \\ \ddot{q}_0 = \\ \\ \dot{q}_f = \\ \ddot{q}_f = \\ \\ \\ \end{array} \begin{bmatrix} q_0 \\ 0 \\ 0 \\ q_f \\ 0 \\ 0 \\ q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix}$$

~~Path~~ Trajectory generation in Cartesian space

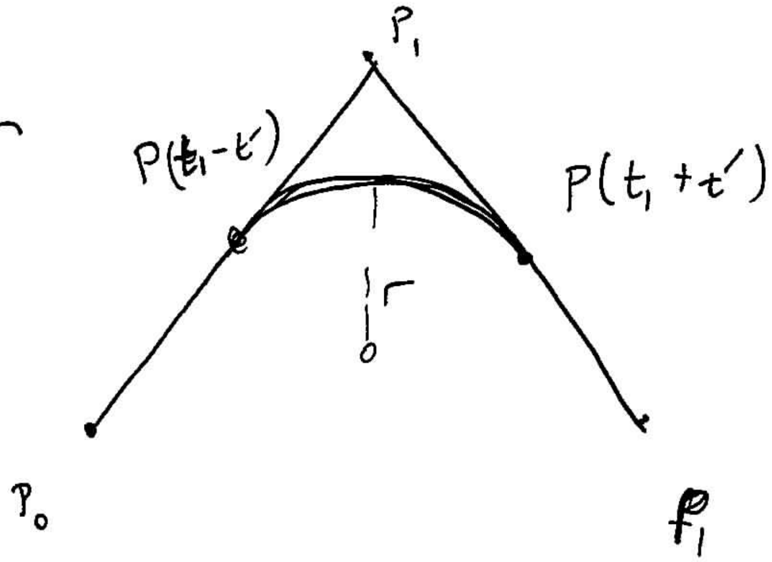




- typically of end-effector planning



- connect them



Jazar

- How to deal with positions (x, y, z) & orientation? (R)

time scaling position & orientation   Li Park Ch.13

- seperate $\underbrace{x, y, z}_{P}$, R

$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ $\quad \vec{P}(s) = \vec{P}_0 + s(\vec{P}_f - \vec{P}_0)$ $\quad s \in [0, 1]$

polynomials

$X(s) = X(0) + s \cdot (X_f - X_0)$

$s = a_0 + a_1 t + a_2 t^2 \dots$

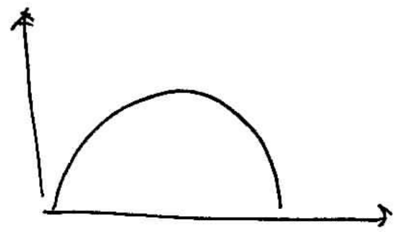
s

Known

$\dot{x} = \frac{dx}{ds} \dot{s}$

$\ddot{x} = \frac{dx}{ds} \ddot{s} + \frac{d^2 x}{ds^2} \dot{s}^2$

$\dot{s}$



ω

$R(s) = R_0 \underbrace{e^{\log(R_0^T R_f) s}}_{\Delta R}$

$R = e^{\omega}$

L & P Ch.13

Typical Procedure

↓ task

Points in Cartesian Space

↓ Interpolation

Cartesian Path → $P(s)$, $R(s)$

↓ Sample, IK

Sequence in Joint space

↓ Interpolate

Path in Joint space