Lecture 16 11/22/2021 ME221

Last time:

→ focus was on trajectory generation & matlab implementation.

Today:

- Topics on the Final Exam
- Survey of control techniques.

Final Exam Topics:

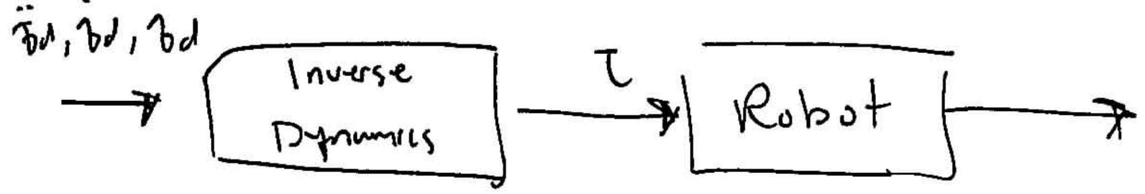
- Jacobian generating vectors
- singularities
- manipulability
- Dynamics (Lagrangian Mechanics)
- Trajectory generation using polynomial
- Controls
  - PID
  - ~~Feedfwd~~ Feedforward
  - computed Torque
  - end-effector control

Feed forward Control.

Q: If you know <u>exactly</u> the robot dynamics, how would you use that model for control?

$$\tau = \tilde{M}(q_d)\ddot{q}_d + \tilde{C}(q_d, \dot{q}_d)\dot{q}_d + \tilde{g}(q_d)$$

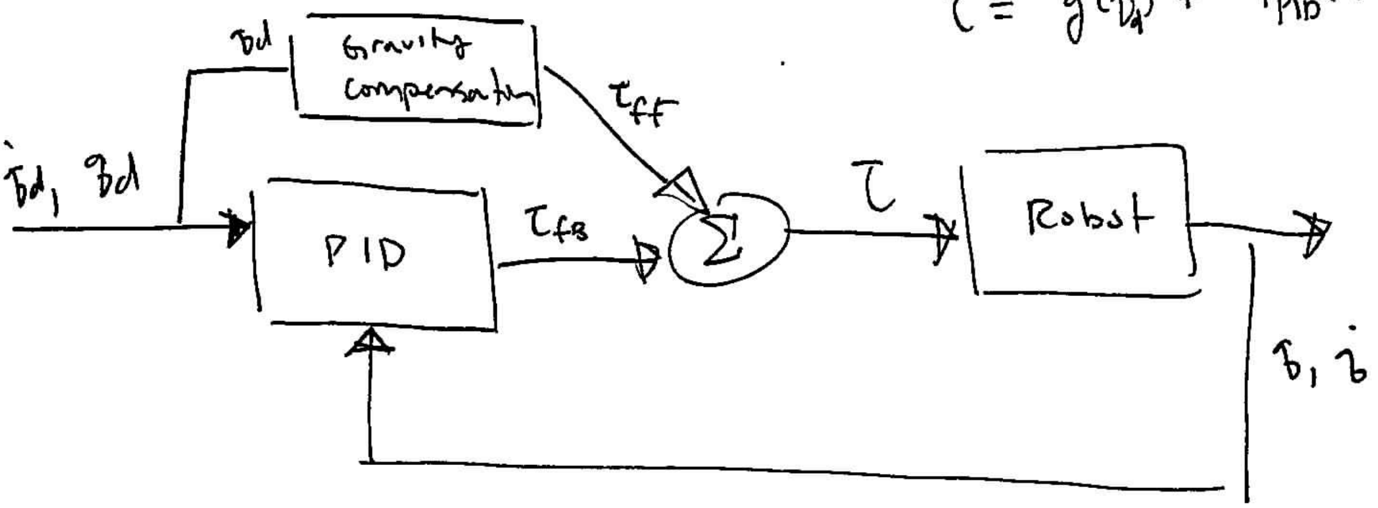
"Inverse Dynamics"



- <u>Impossible</u> to have a perfect model
- computationally expensive

We can "feed forward" parts of the dynamics

$$\tau = \tilde{g}(q_d) + f_{PID}(q, \dot{q})$$



$$\tau = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + K_I\int q_d - q \, dt + \tilde{g}(q_d)$$

If you have a good model

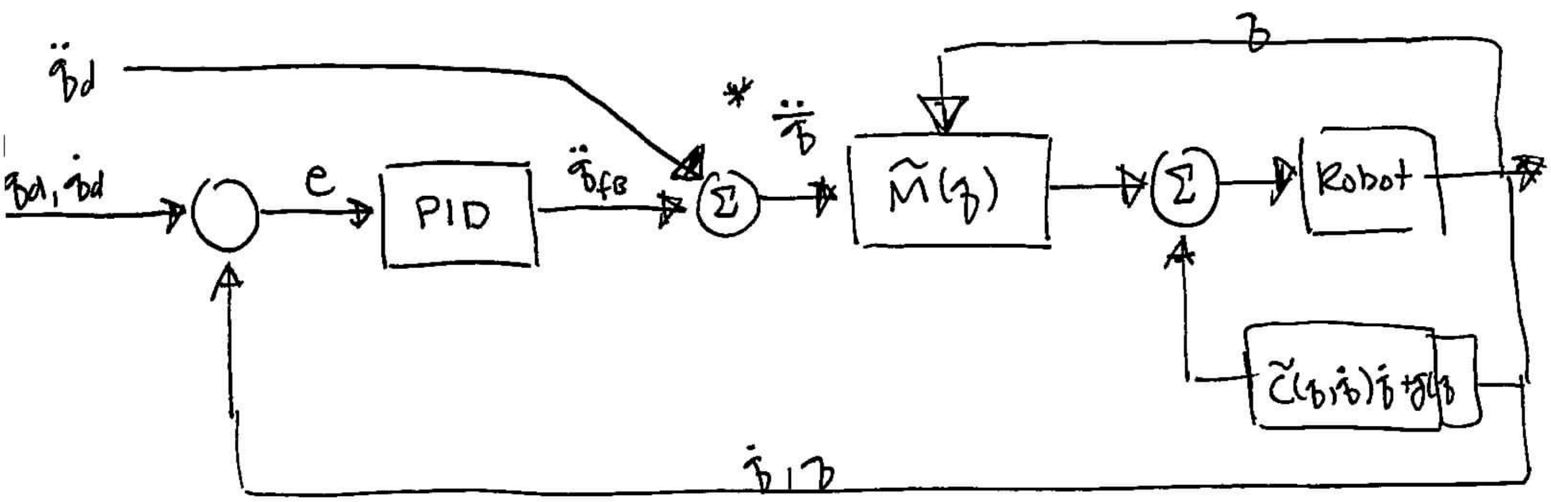
We can use our model to cancel the effect of robot dynamics

$$\tilde{M}(q)\ddot{q} + \tilde{C}(q,\dot{q})\dot{q} + \tilde{g}(q) = \tau$$

$$\tau = \tilde{M}(q)\left[\ddot{q}_d + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + K_I\int q_d - q \, dt\right]^*$$

$$+ \tilde{C}(q,\dot{q})\dot{q} + g(q)$$

"Compute torque Controller" or "Inverse Dynamics Controller"



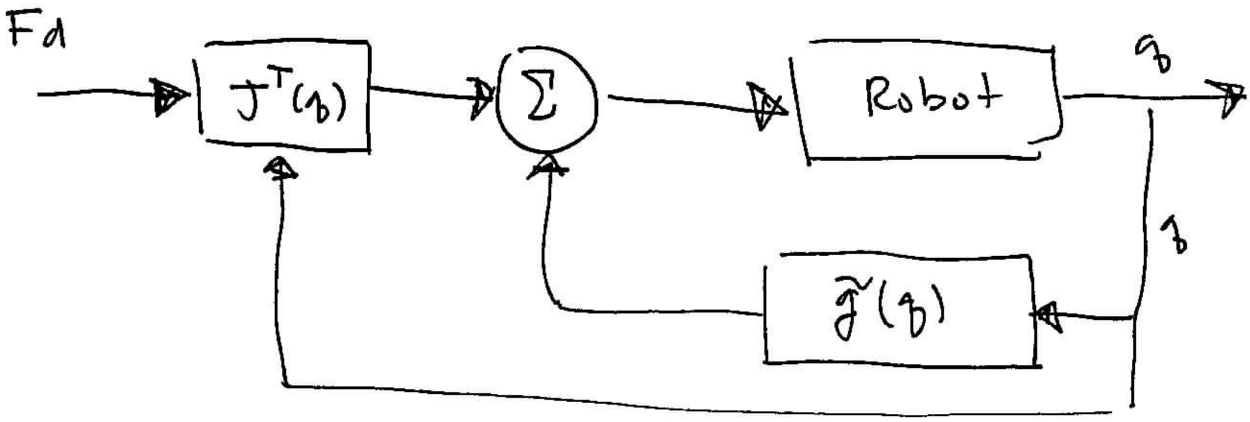
$M(q)\ddot{q}$ - - -

Force Control

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau$$

F_d : desired Tip force (end-effector)

$$\tau = \tilde{g}(q) + J^T(q) F_d$$



We can also add PI loop

* we don't derivative because in practice $\dot{F}_d$ → noisy!

$$\tau = \tilde{g}(q) + J^T(q)\left[F_d + K_p(F_d - F_{ext}) + K_I \int F_d - F_{ext}\, dt\right]$$

need force sensor!

Task space Control: (PD-control)

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau$$

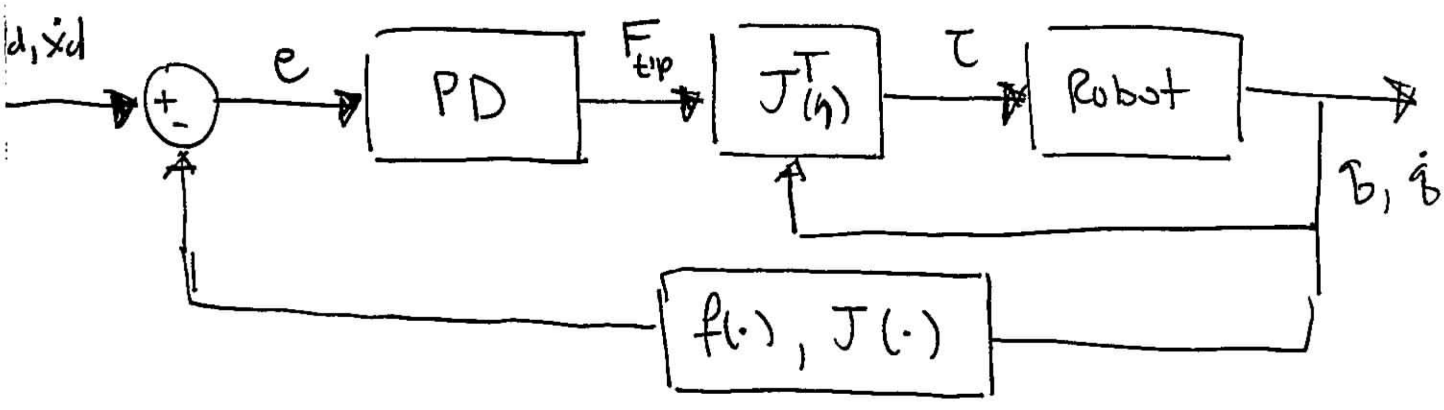
$$e = x_d - x \qquad\qquad e = x_d - \overbrace{f(q)}^{\text{forward kinematics}}$$

$$\dot{e} = \dot{x}_d - \dot{x} \quad \Longrightarrow \quad \dot{e} = \dot{x}_d - J(q)\dot{q}$$

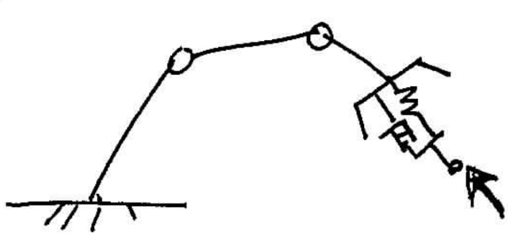
$$F_{tip} = K_p(x_d - f(q)) + K_d(\dot{x}_d - J(q)\dot{q})$$

$$\tau = J^T(q) F_{tip}$$

$$\tau = J^T(q)\left[K_p(x_d - f(q)) + K_d(\dot{x}_d - J(q)\dot{q})\right]$$



Impedance controller: we want tip to behave as a spring damper



We can convert the Robot Dynamics into task space dynamics:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau$$

$$\begin{cases} \dot{x} = J(q)\dot{q} \\ \end{cases} \qquad \begin{cases} \tau = J^T F \\ F = J^{-T}\tau \end{cases}$$

$$\Downarrow$$

$$\ddot{x} = J(q)\ddot{q} + \dot{J}\dot{q}$$

$$= J\ddot{q} + \dot{J}\dot{q} \Rightarrow \ddot{q} = J^{-1}\ddot{x} - J^{-1}\dot{J}\dot{q}$$

$$\begin{cases} \ddot{q} = J^{-1}\ddot{x} - J^{-1}\dot{J}J^{-1}\dot{x} \\ \dot{q} = J^{-1}\dot{x} \end{cases}$$

$$M\left(J^{-1}\ddot{x} - J^{-1}\dot{J}J^{-1}\dot{x}\right) + C\left(J^{-1}\dot{x}\right) + g = \tau$$

$$J^{-T}M\left(J^{-1}\ddot{x} - J^{-1}\dot{J}J^{-1}\dot{x}\right) + J^{-T}C\left(J^{-1}\dot{x}\right) + J^{-T}g = J^{-T}\tau$$

$$J^{-T}MJ^{-1}\ddot{x} - J^{-T}MJ^{-1}\dot{J}J^{-1}\dot{x} + J^{-T}CJ^{-1}\dot{x} + J^{-T}g = J^{-T}\tau$$

$$\Rightarrow \left[J^{-T}MJ^{-1}\right]\ddot{x} + \left[J^{-T}CJ^{-1} - J^{-T}MJ^{-1}\dot{J}J^{-1}\right]\dot{x} + J^{-T}g = F$$

$$\boxed{\Lambda(q)\ddot{x} + \Gamma(q,\dot{q})\dot{x} + \eta(q) = F}$$

Computed Torque @ the End-effector:

$$\tau = J^T \Big[\tilde{\Lambda}(q) \Big(\ddot{x}_d + K_p (x_d - x) + K_d (\dot{x}_d - \dot{x}) + K_I \int (x_d - x)\, dt \Big) + \tilde{\Gamma}(q, \dot{q})\, \dot{x} + \tilde{\eta}(q) \Big]$$

- Springer handbook of Robotics ch. 8
- Jazar Ch. 15
- Lynch & Park Ch. 11
