Robot Control: Motion Control with Forces or Torques

CS 6301 Special Topics: Introduction to Robot Manipulation and Navigation

Professor Yu Xiang

The University of Texas at Dallas

Robot Control

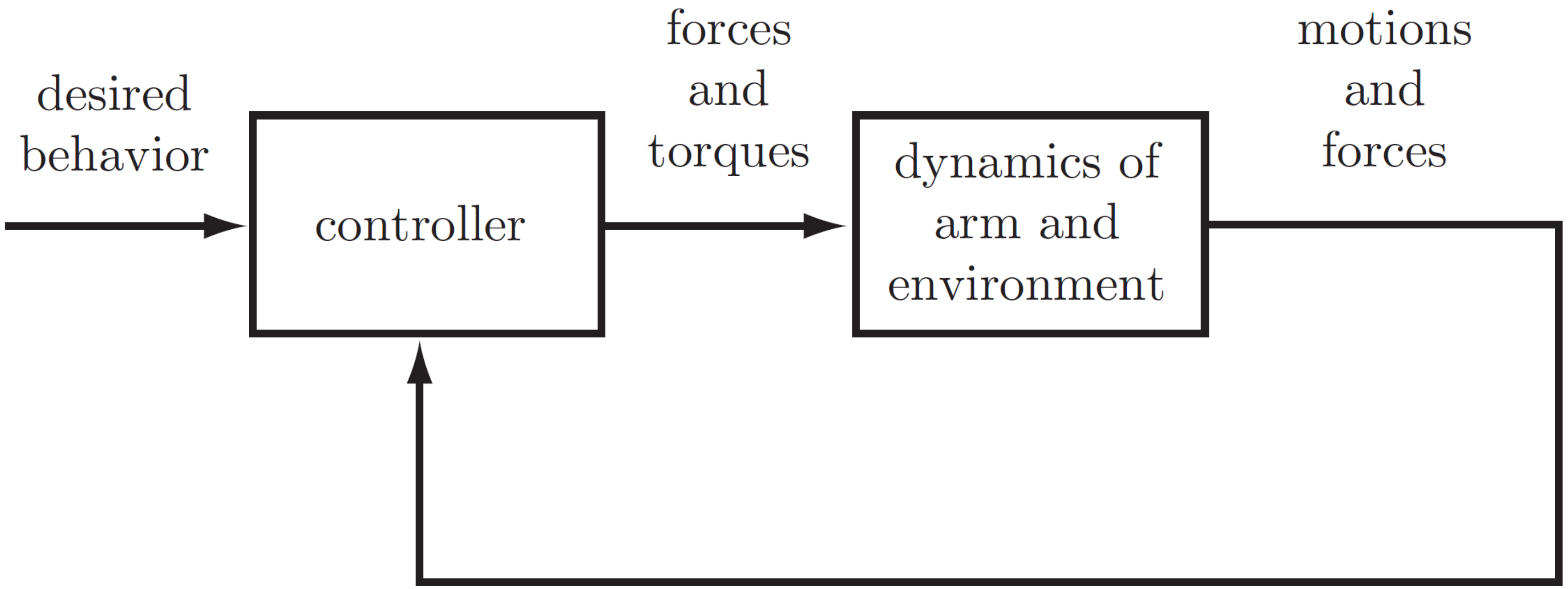
- Convert task specifications to force and torques at the actuators

- Types
  - Motion control
  - Force control
  - Hybrid motion-force control
  - Impedance control

- Feedback control
  - Use sensors for position, velocity and force
  - Compare with the desired behavior to compute the control signals

Control System Overview

- A simplified system



Motion Control with Velocity Inputs

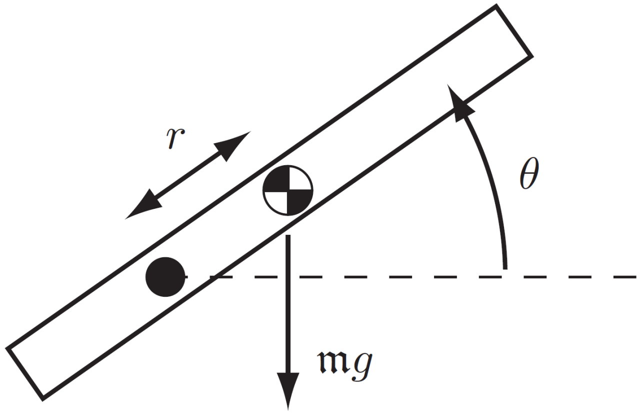
- Motion control with velocity inputs
  - Given a desired trajectory of a robot in joint space or in task space
  - Direct control of the joint velocities

- Limited to applications with low or predictable force-torque requirements

- Do not make use of a dynamic model of the robot

Motion Control with Torque or Force Inputs

- Controller generates joint torques and forces to track a desired trajectory
- Motion Control of a single joint



Dynamics $\tau = M\ddot{\theta} + \mathfrak{m}gr\cos\theta$

Scalar inertia

mass

Friction torque $\tau_{\text{fric}} = b\dot{\theta}$

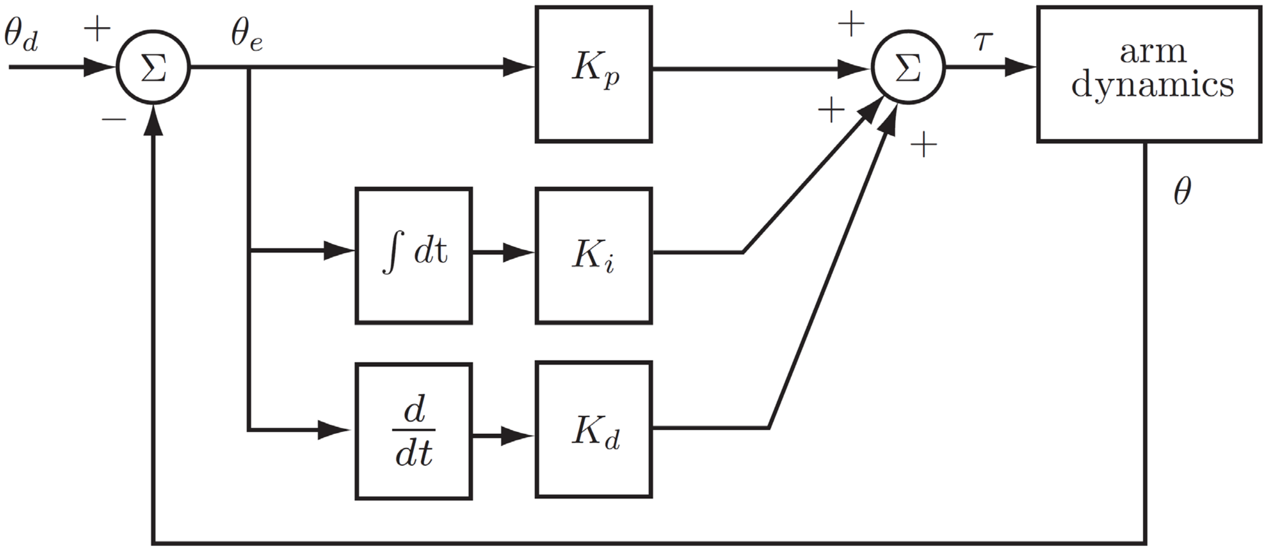
$$\tau = M\ddot{\theta} + \mathfrak{m}gr\cos\theta + b\dot{\theta}$$

$$\tau = M\ddot{\theta} + h(\theta, \dot{\theta})$$

Motion Control of a Single Joint

- Feedback control: PID control
  - Proportional-Integral-Derivative control

$$\tau = K_p\theta_e + K_i \int \theta_e(\mathrm{t})d\mathrm{t} + K_d\dot{\theta}_e \qquad \theta_e = \theta_d - \theta$$



PD Control

- Dynamics $\tau = M\ddot{\theta} + \mathfrak{m}gr\cos\theta + b\dot{\theta}$
- PD control law $K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta})$

Assume $g = 0$

$$M\ddot{\theta} + b\dot{\theta} = K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta})$$

Control objective: constant θ_d $\dot{\theta}_d = \ddot{\theta}_d = 0$

$$\theta_e = \theta_d - \theta \qquad \dot{\theta}_e = -\dot{\theta} \qquad \ddot{\theta}_e = -\ddot{\theta}$$

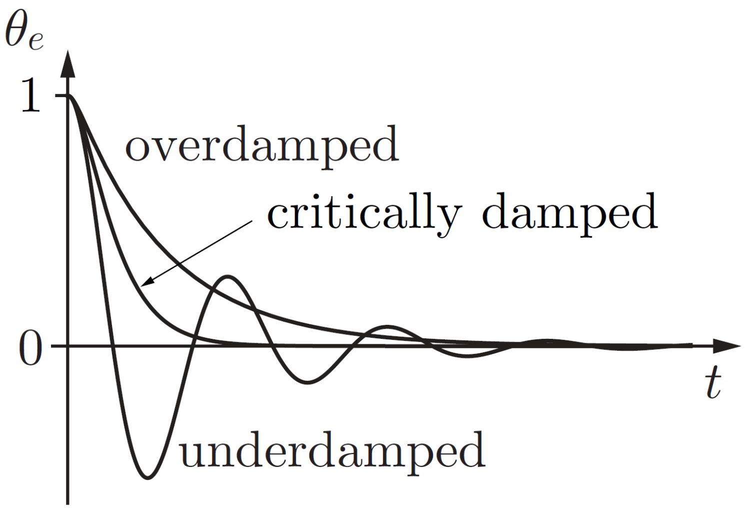
Error dynamics $M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = 0$

PD Control

- Standard second-order form

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = 0$$

$$\ddot{\theta}_e + \frac{b + K_d}{M}\dot{\theta}_e + \frac{K_p}{M}\theta_e = 0 \quad \rightarrow \quad \ddot{\theta}_e + 2\zeta\omega_n\dot{\theta}_e + \omega_n^2\theta_e = 0$$



$$\zeta = \frac{b + K_d}{2\sqrt{K_p M}} \qquad \omega_n = \sqrt{\frac{K_p}{M}}$$

PD Control

- When g > 0, the error dynamics

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = \mathfrak{m}gr\cos\theta$$

When the joint comes to rest at a configuration θ , $K_p\theta_e = \mathfrak{m}gr\cos\theta$

the final error θ_e is nonzero when $\theta_d \neq \pm\pi/2$

Non-zero steady-state error

PID Control

- Setpoint error dynamics

$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e + K_i \int \theta_e(\mathrm{t})dt = \tau_{\mathrm{dist}}$$

Disturbance torque

$\mathfrak{m}gr \cos\theta$

Taking derivatives

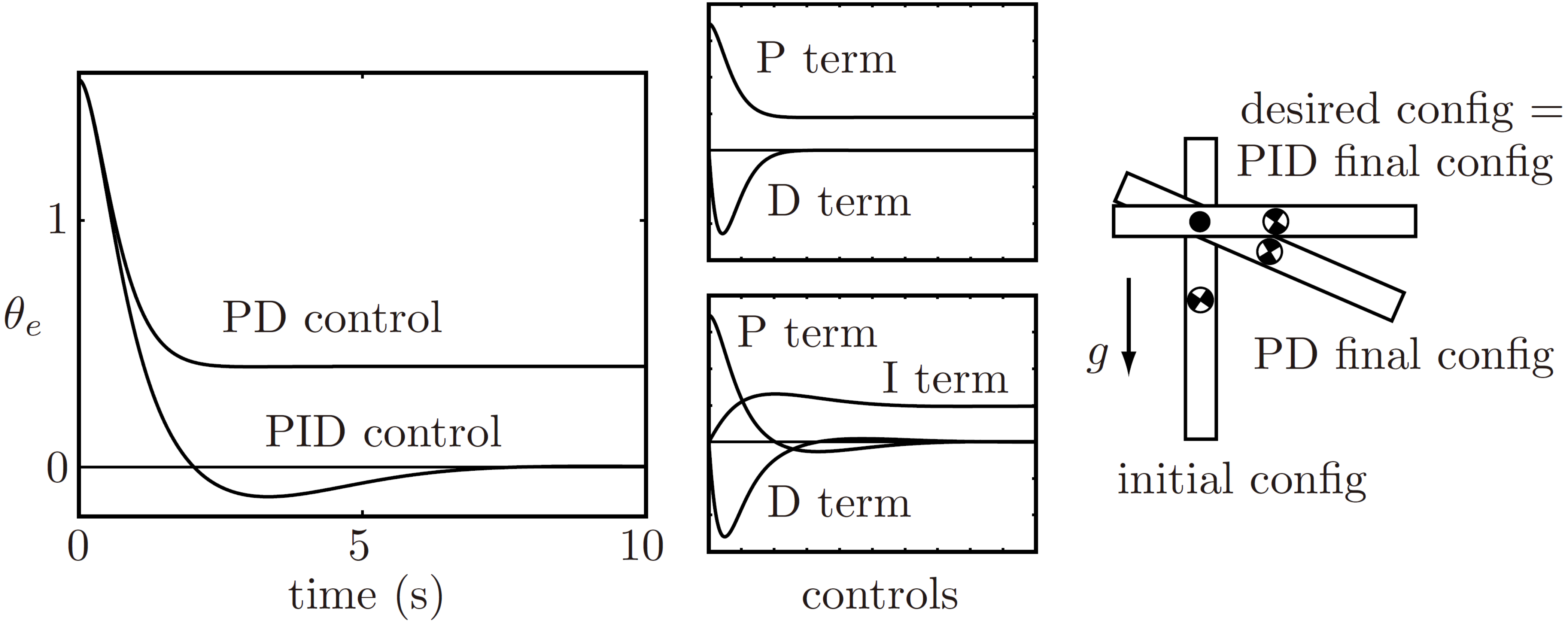
$$M\theta_e^{(3)} + (b + K_d)\ddot{\theta}_e + K_p\dot{\theta}_e + K_i\theta_e = \dot{\tau}_{\mathrm{dist}}$$

Third-Order Error Dynamics $$s^3 + \frac{b + K_d}{M}s^2 + \frac{K_p}{M}s + \frac{K_i}{M} = 0$$

If τ_{dist} Constant

If all roots have a negative real part, then the error dynamics is stable, and θ_e converges to zero

PID Control


θe
1
0
PD control
PID control
0
5
10
time (s)
P term
D term
P term
I term
D term
controls
desired config =
PID final config
g
PD final config
initial config

PID Control

```
time = 0                            // dt = servo cycle time
eint = 0                            // error integral
qprev = senseAngle                  // initial joint angle q
loop
  [qd,qdotd] = trajectory(time) // from trajectory generator

  q = senseAngle                    // sense actual joint angle
  qdot = (q - qprev)/dt             // simple velocity calculation
  qprev = q

  e = qd - q
  edot = qdotd - qdot
  eint = eint + e*dt

  tau = Kp*e + Kd*edot + Ki*eint
  commandTorque(tau)

  time = time + dt
end loop
```

Feedforward Control

- Uses the dynamics of the robot
- The controller's model of the dynamics

$$\tau = \tilde{M}(\theta)\ddot{\theta} + \tilde{h}(\theta, \dot{\theta})$$

$\tilde{M}(\theta) = M(\theta) \text{ and } \tilde{h}(\theta, \dot{\theta}) = h(\theta, \dot{\theta})$ if the model is perfect

- Given $\theta_d, \ \dot{\theta}_d, \text{ and } \ddot{\theta}_d$

Feedforward torque $\tau(t) = \tilde{M}(\theta_d(t))\ddot{\theta}_d(t) + \tilde{h}(\theta_d(t), \dot{\theta}_d(t))$

The dynamics model of the controller cannot be perfect in practice

Feedforward Plus Feedback Linearization

- Goal: achieve the following error dynamics

$$\ddot{\theta}_e + K_d\dot{\theta}_e + K_p\theta_e + K_i \int \theta_e(\mathrm{t})dt = c$$

A PID controller can achieve exponential decay of the trajectory error

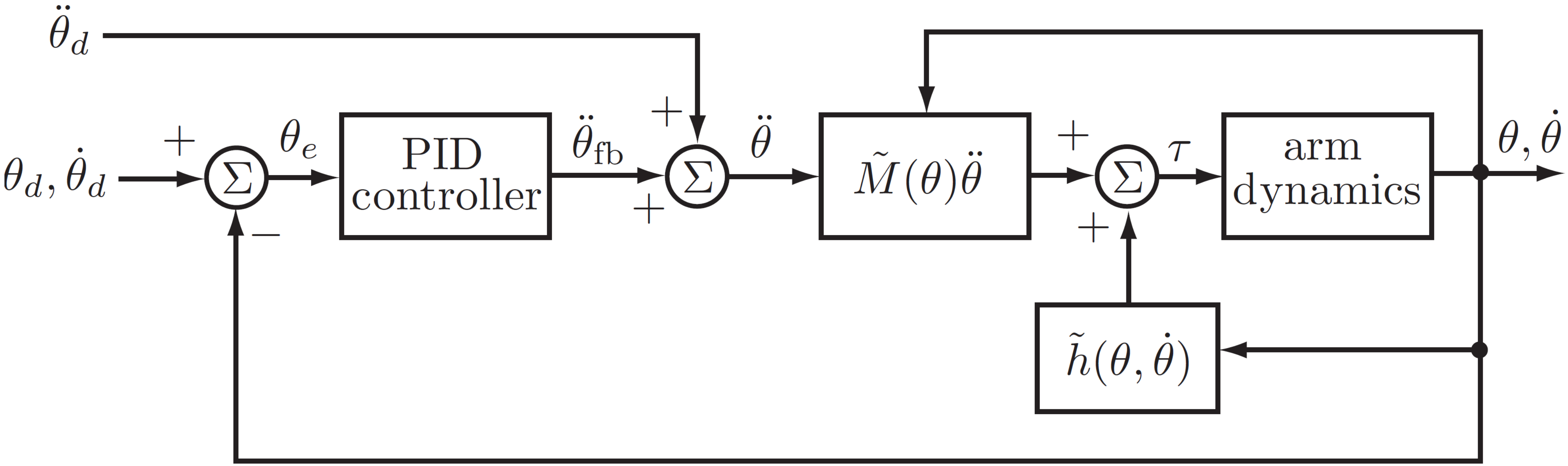
- We first choose $\ddot{\theta} = \ddot{\theta}_d - \ddot{\theta}_e$ $\qquad \ddot{\theta} = \ddot{\theta}_d + K_d\dot{\theta}_e + K_p\theta_e + K_i \int \theta_e(\mathrm{t})dt$

- Feedforward plus feedback linearizing controller (inverse dynamics controller, computed torque controller)

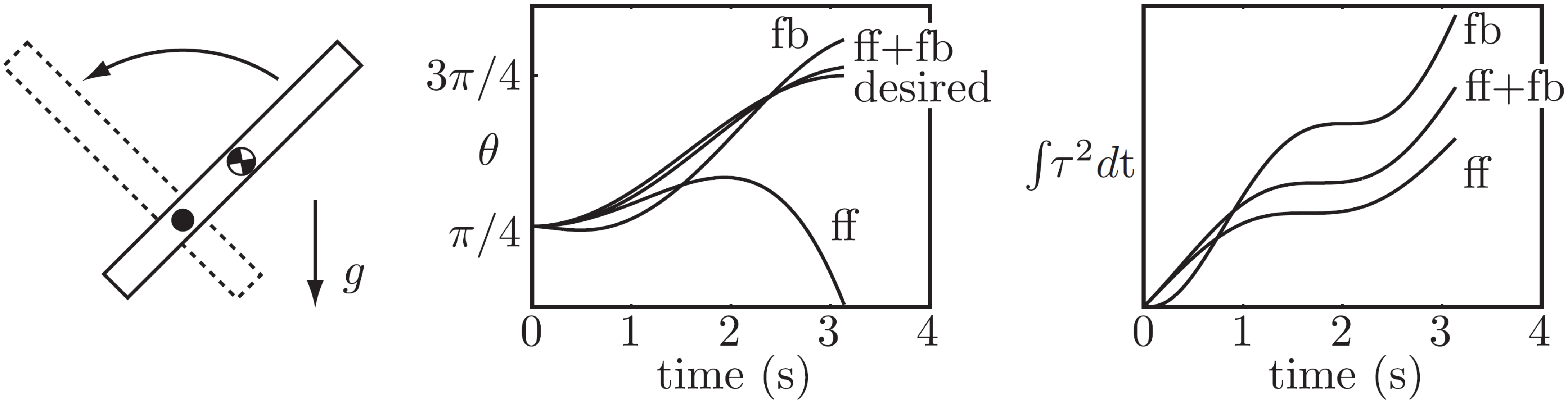
$$\tau = \tilde{M}(\theta)\left(\ddot{\theta}_d + K_p\theta_e + K_i \int \theta_e(\mathrm{t})dt + K_d\dot{\theta}_e\right) + \tilde{h}(\theta, \dot{\theta})$$

Feedforward Plus Feedback Linearization

$$\tau = \tilde{M}(\theta)\left(\ddot{\theta}_d + K_p\theta_e + K_i\int \theta_e(\mathrm{t})dt + K_d\dot{\theta}_e\right) + \tilde{h}(\theta,\dot{\theta})$$



Feedforward Plus Feedback Linearization



Motion Control of a Multi-joint Robot

- Dynamics

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})$$

$$n \times n$$

- Decentralized control
  - Each joint is controlled independently
  - When dynamics are decoupled

- Centralized control
  - Full state information for each of the n joints is available to calculate the controls for each joint

Centralized Multi-joint Control

- Computed torque controller

$$\tau = \tilde{M}(\theta)\left(\ddot{\theta}_d + K_p\theta_e + K_i\int \theta_e(\mathrm{t})d\mathrm{t} + K_d\dot{\theta}_e\right) + \tilde{h}(\theta,\dot{\theta})$$

K_p, K_i, K_d positive-definite matrices

We choose the gain matrices as $k_pI,\ k_iI,\ \text{and}\ k_dI$

- PID control and gravity compensation

When the model is not good

$$\tau = K_p\theta_e + K_i\int \theta_e(\mathrm{t})d\mathrm{t} + K_d\dot{\theta}_e + \tilde{g}(\theta)$$

Task-Space Motion Control

- Motion as a trajectory of the end-effector in the task space

$$(X(t), \mathcal{V}_b(t)) \qquad X \in SE(3) \qquad [\mathcal{V}_b] = X^{-1}\dot{X}$$ Twist

- Option 1: convert the trajectory to joint space
  - Forward kinematics $X = T(\theta) \qquad \mathcal{V}_b = J_b(\theta)\dot{\theta}$
  - Inverse kinematics

$$\begin{aligned}\theta(t) &= T^{-1}(X(t)),\\ \dot{\theta}(t) &= J_b^{\dagger}(\theta(t))\mathcal{V}_b(t),\\ \ddot{\theta}(t) &= J_b^{\dagger}(\theta(t))\left(\dot{\mathcal{V}}_b(t) - \dot{J}_b(\theta(t))\dot{\theta}(t)\right)\end{aligned}$$

may require significant computing power

Task-Space Motion Control

- Task-space dynamics

$$\mathcal{V} = J(\theta)\dot{\theta} \qquad \dot{\mathcal{V}} = \dot{J}(\theta)\dot{\theta} + J(\theta)\ddot{\theta}$$

$$\begin{aligned} \dot{\theta} &= J^{-1}\mathcal{V}, \\ \ddot{\theta} &= J^{-1}\dot{\mathcal{V}} - J^{-1}\dot{J}J^{-1}\mathcal{V} \end{aligned}$$

Dynamics $\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})$

$$\tau = M(\theta)\left(J^{-1}\dot{\mathcal{V}} - J^{-1}\dot{J}J^{-1}\mathcal{V}\right) + h(\theta, J^{-1}\mathcal{V})$$

$$\begin{aligned} J^{-\mathrm{T}}\tau \quad = \quad & J^{-\mathrm{T}}MJ^{-1}\dot{\mathcal{V}} - J^{-\mathrm{T}}MJ^{-1}\dot{J}J^{-1}\mathcal{V} \\ & + J^{-\mathrm{T}}h(\theta, J^{-1}\mathcal{V}). \end{aligned}$$

Section 8.6 in Modern Robotics

Task-Space Motion Control

- Task-space dynamics

$$\mathcal{F} = \Lambda(\theta)\dot{\mathcal{V}} + \eta(\theta, \mathcal{V})$$

$$\begin{aligned}
\Lambda(\theta) &= J^{-\mathrm{T}} M(\theta) J^{-1}, \\
\eta(\theta, \mathcal{V}) &= J^{-\mathrm{T}} h(\theta, J^{-1}\mathcal{V}) - \Lambda(\theta)\dot{J}J^{-1}\mathcal{V}.
\end{aligned}$$

Task-Space Motion Control

- Option 2: task-space dynamics

$$\mathcal{F}_b = \Lambda(\theta)\dot{\mathcal{V}}_b + \eta(\theta, \mathcal{V}_b)$$

- Joint forces and torques

$$\tau = J_b^{\mathrm{T}}(\theta)\mathcal{F}_b$$

- Computed torque controller

$$\tau =$$

$$J_b^{\mathrm{T}}(\theta)\left(\tilde{\Lambda}(\theta)\left(\frac{d}{dt}([\mathrm{Ad}_{X^{-1}X_d}]\mathcal{V}_d) + K_p X_e + K_i \int X_e(\mathrm{t})d\mathrm{t} + K_d \mathcal{V}_e\right) + \tilde{\eta}(\theta, \mathcal{V}_b)\right)$$

Summary

- Motion control with torque or force Inputs
    - PID control
    - Computed torque control

- Task-space motion control

Further Reading

- Chapter 11 in Kevin M. Lynch and Frank C. Park. Modern Robotics: Mechanics, Planning, and Control. 1st Edition, 2017.