Robot Control: Motion Control with Torque or Force Inputs

CS 6341 Robotics
Professor Yu Xiang
The University of Texas at Dallas

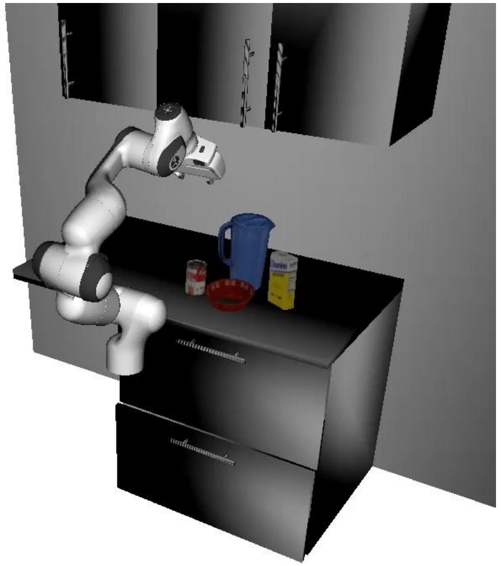
Motion Control

- Goal: follow a given robot trajectory
  - Trajectory of desired end-effector configuration $X_d(t)$
  - Trajectory of desired joint positions $\theta_d(t)$

Can include

$$\dot{\theta}_d(t)$$

$$\ddot{\theta}_d(t)$$

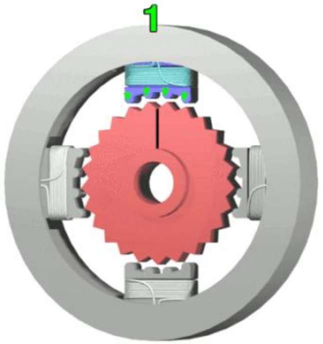


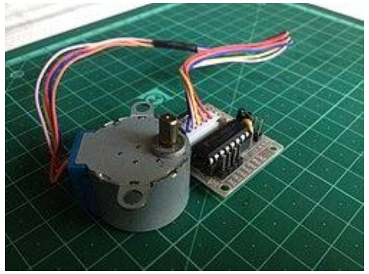
Motion Control

Last lecture

task specifications → Controller → Joint velocities

Stepper Motors





- Limited to applications with low or predictable force-torque requirements
- Do not make use of a dynamic model of the robot

Today

task specifications → Controller → Forces and torques

Controller generates joint torques and forces to track a desired trajectory

Motion Control of a single joint with Torque or Force Inputs

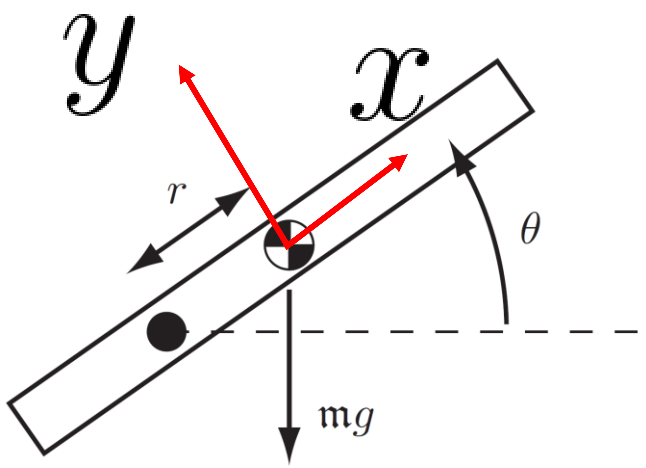
- Dynamics of a singe-joint robot

Rotational dynamics of a Single Rigid Body

Torque; inertia matrix

$$m_b = \mathcal{I}_b \dot{\omega}_b + [\omega_b]\mathcal{I}_b \omega_b$$

Angular velocity $\omega_b = (\omega_x, \omega_y, \omega_z)$

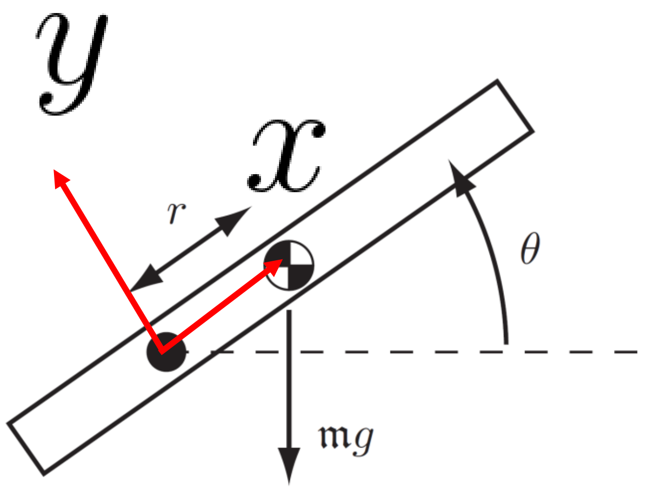


If the principal axes are aligned with the axes of {b}, $\mathcal{I}_b$ is a diagonal matrix

$$m_b = \begin{bmatrix} \mathcal{I}_{xx}\dot{\omega}_x + (\mathcal{I}_{zz} - \mathcal{I}_{yy})\omega_y\omega_z \\ \mathcal{I}_{yy}\dot{\omega}_y + (\mathcal{I}_{xx} - \mathcal{I}_{zz})\omega_x\omega_z \\ \mathcal{I}_{zz}\dot{\omega}_z + (\mathcal{I}_{yy} - \mathcal{I}_{xx})\omega_x\omega_y \end{bmatrix}$$

$$\omega_b = [0, 0, \dot{\theta}]^T \qquad m_{bz} = I_{zz}\ddot{\theta}$$

Motion Control with Torque or Force Inputs



Move the body frame to the joint

Steiner's theorem

$$\mathcal{I}_q = \mathcal{I}_b + \mathfrak{m}(q^{\mathrm{T}}qI - qq^{\mathrm{T}})$$

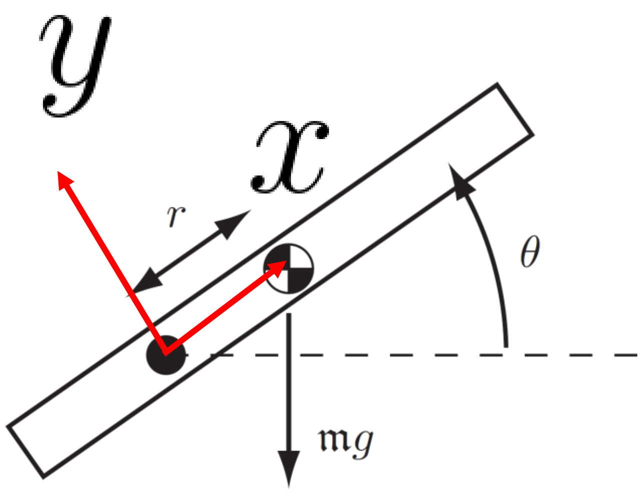
We still have torque $m_{bz} = I_{zz}\ddot{\theta}$

- Torque applied by the joint τ
- Gravity torque $m_a = r_a \times f_a$
  $\tau_g = -mgr\cos\theta$
- Friction torque (damping) $\tau_f = -b\dot{\theta}$

$$\tau - mgr\cos\theta - b\dot{\theta} = I_{zz}\ddot{\theta}$$

Motion Control with Torque or Force Inputs

- Motion Control of a single joint



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

Scalar inertia

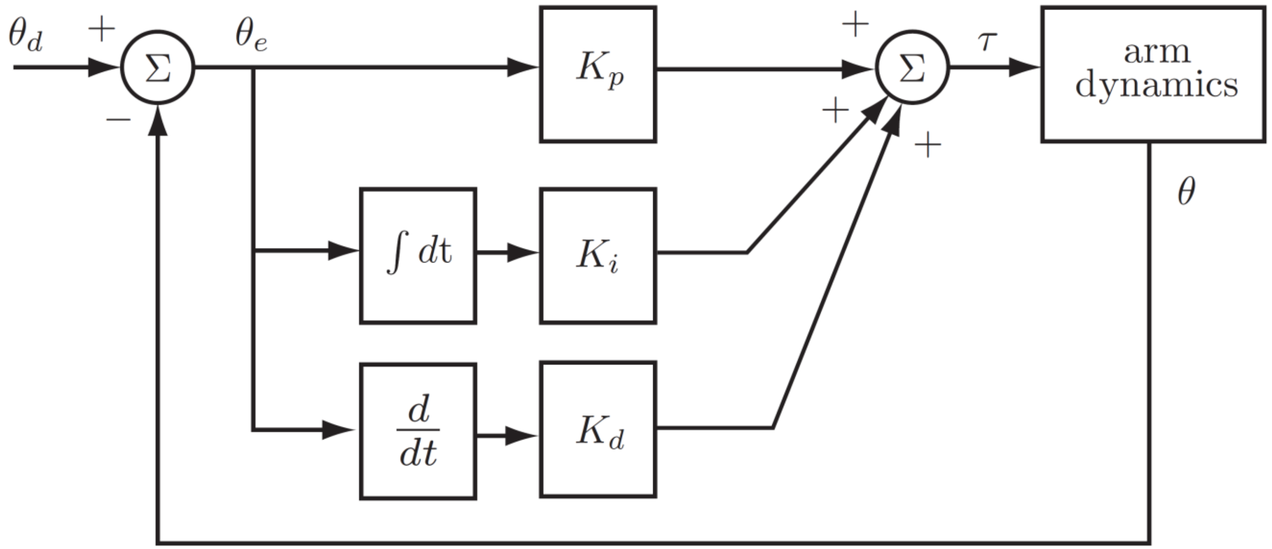
mass

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

Motion Control of a Single Joint

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

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



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 the torque

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

Setpoint control

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

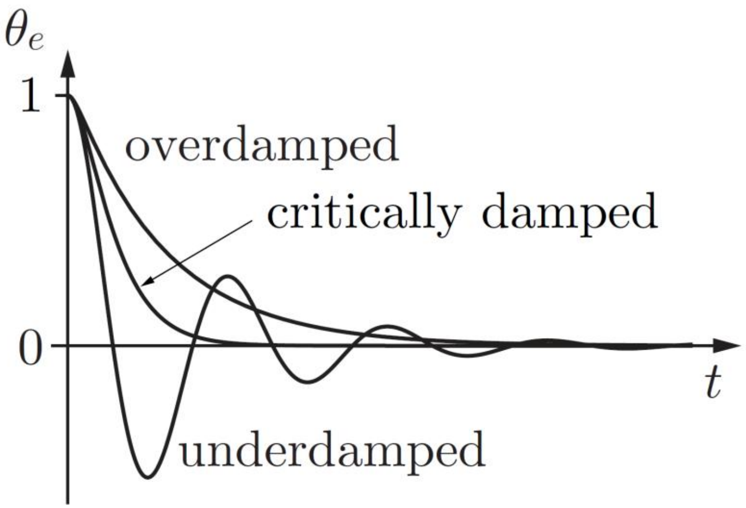
Error dynamics $M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = 0$

PD Control

- Standard second-order form

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

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



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

Critically damped: $\zeta = 1$

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 ,

$$\dot{\theta} = \ddot{\theta} = 0$$

$$K_p\theta_e = \mathfrak{m}gr\cos\theta$$

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

Non-zero steady-state error

PID Control

- Setpoint error dynamics

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

Disturbance torque

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

Taking derivatives

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

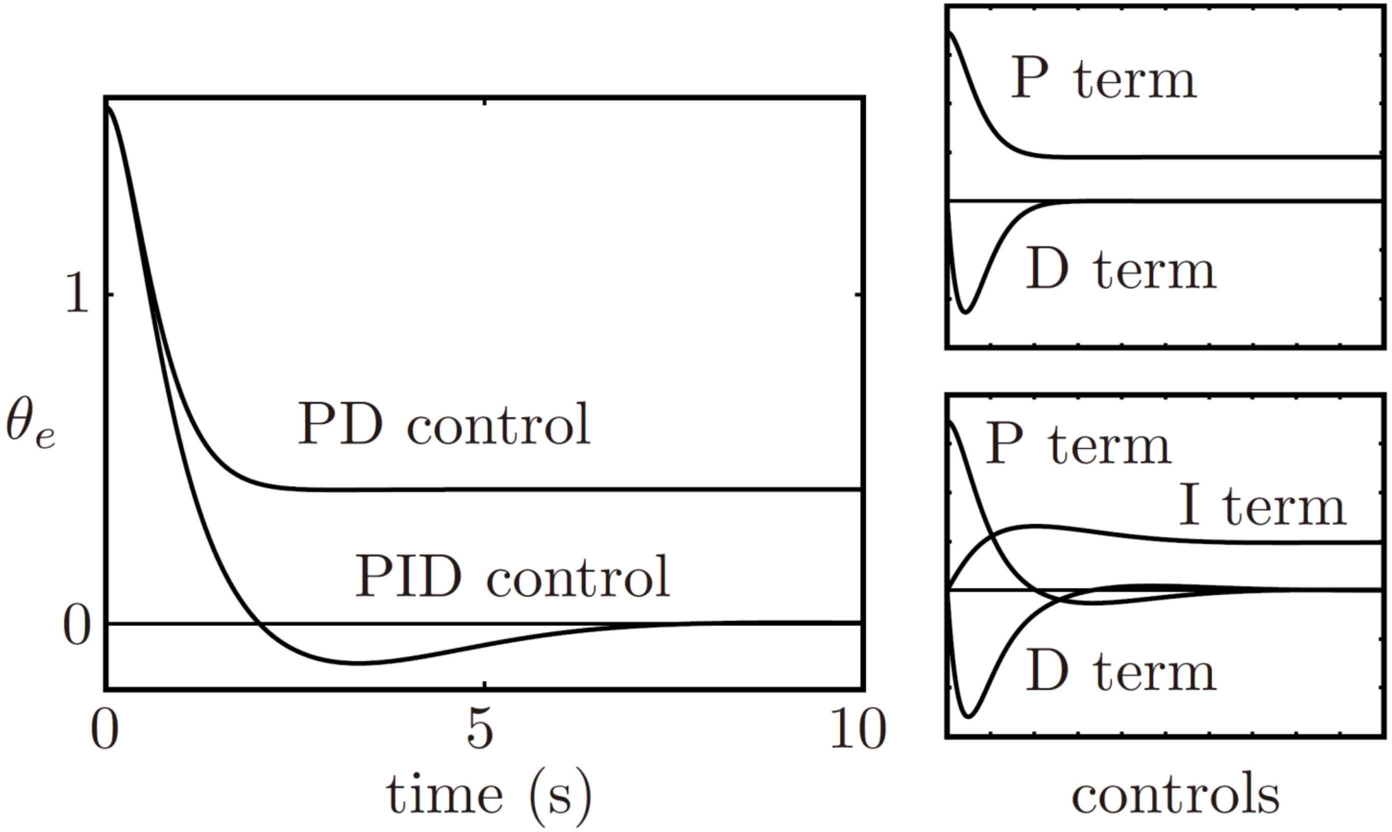
Third-Order Error Dynamics

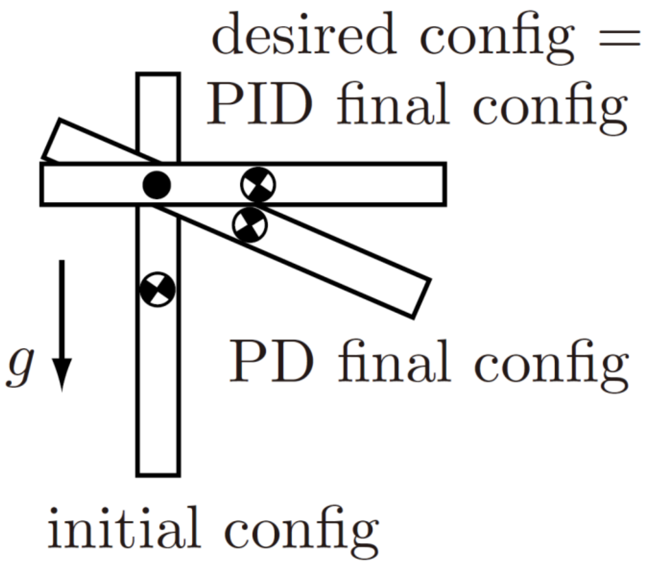
$$s^3 + \frac{b + K_d}{M}s^2 + \frac{K_p}{M}s + \frac{K_i}{M} = 0$$

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





$$\theta(0) = -\pi/2, \dot{\theta}(0) = 0$$

$$\theta_d = 0, \dot{\theta}_d = 0$$

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

Accumulate error in position

Feedforward Plus Feedback Linearization

- Goal: achieve the following error dynamics

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

A PID controller can achieve exponential decay of the trajectory error

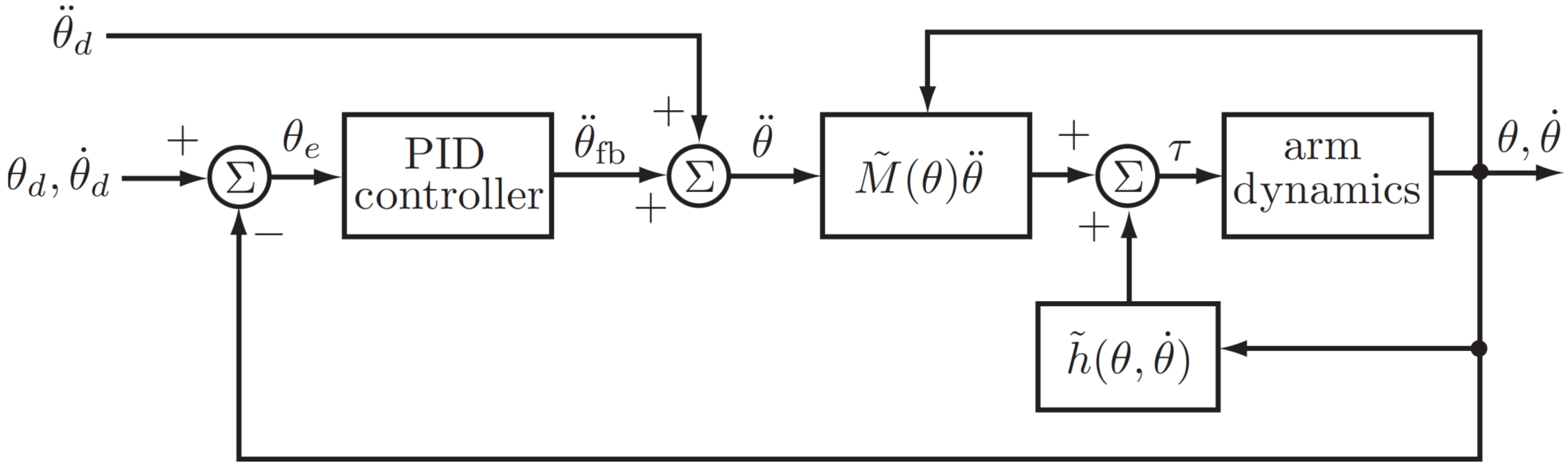
- We first choose $\ddot{\theta} = \ddot{\theta}_d - \ddot{\theta}_e$ $\quad \ddot{\theta} = \ddot{\theta}_d + K_d\dot{\theta}_e + K_p\theta_e + K_i \int \theta_e(\mathrm{t})d\mathrm{t}$

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

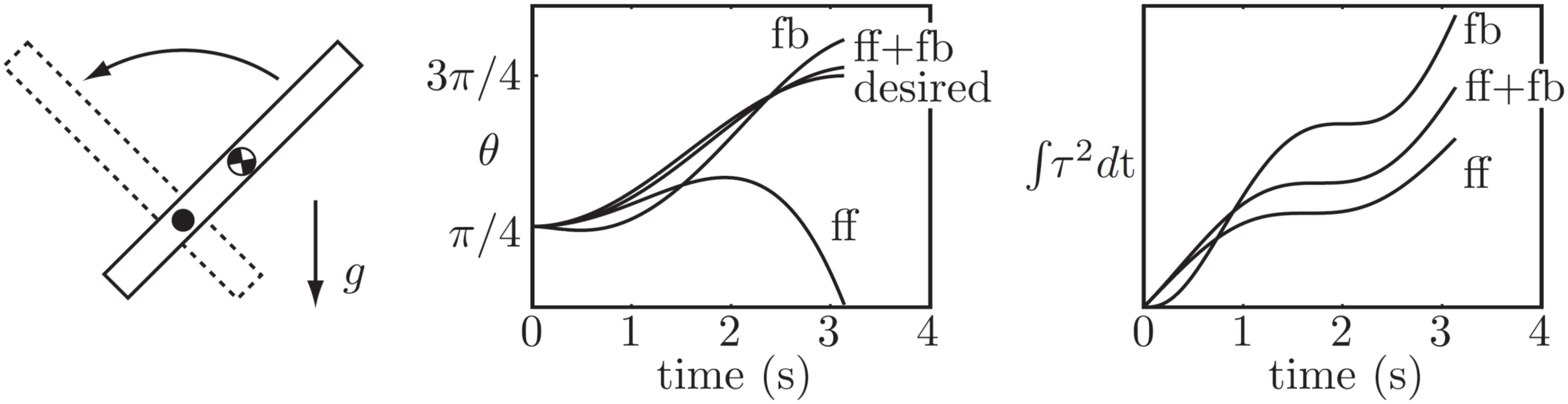
$$\tau = \tilde{M}(\theta)\left(\ddot{\theta}_d + K_p\theta_e + K_i \int \theta_e(\mathrm{t})d\mathrm{t} + K_d\dot{\theta}_e\right) + \tilde{h}(\theta, \dot{\theta})$$

Feedforward Plus Feedback Linearization

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



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 (approximately)

- 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

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

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

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

Summary

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

Further Reading

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