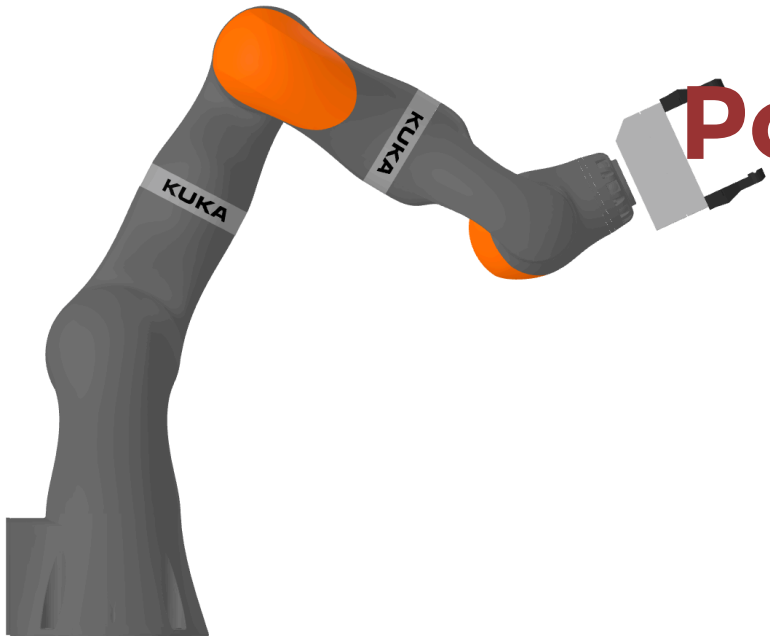


Position and Force Control

MIT 6.4210/2

Robotic Manipulation

Fall 2026, Lecture 14

Massachusetts Institute of Technology

Position and Force Control

feedback · force · stiffness

MIT 6.4210/2 Robotic Manipulation

Fall 2026, Lecture 14

So far, everything was kinematic

differential IK $\rightarrow v$

trajectory optimization $\rightarrow q_d(t)$

grasp planning $\rightarrow X^G$

who turns these into motor torques?

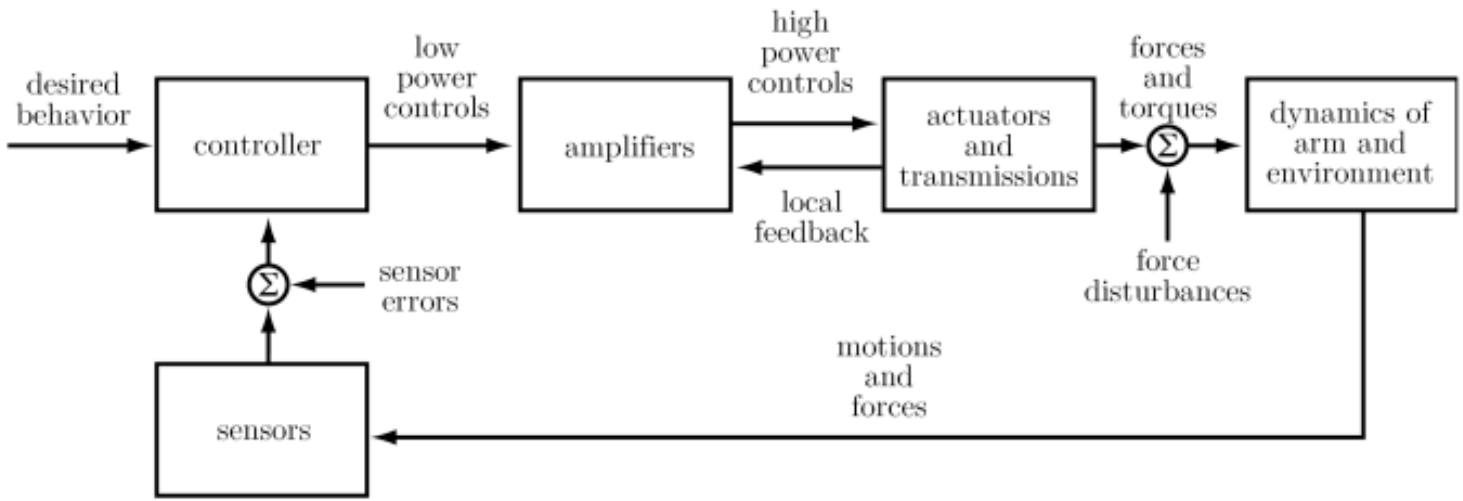
and what about tasks that are about force?

Control objectives

motion	positions, velocities
force	contact forces
hybrid	force in some directions, motion in others
impedance	how the robot responds to force

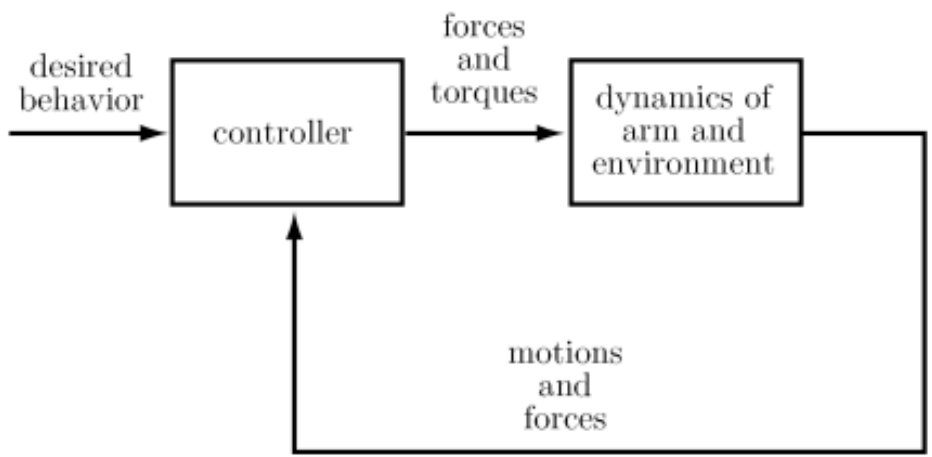
Drake: `PidController`, `JointStiffnessController`, `InverseDynamicsController`

Control system block diagram:



Modern Robotics, Lynch and Park, Cambridge University Press

Simplified block diagram:



Also assuming continuous-time (not discrete-time) control.

Modern Robotics, Lynch and Park, Cambridge University Press

Part I

position control

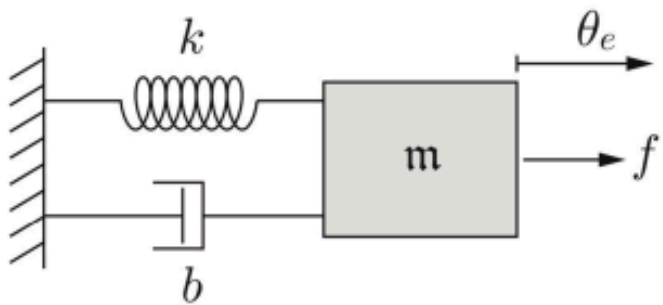
System dynamics, feedback controllers, and error response are often modeled by **linear ordinary differential equations**.

The simplest linear ODE exhibiting overshoot is second order, e.g.,

$$\mathrm{m}\ddot{\theta}_e + b\dot{\theta}_e + k\theta_e = f$$

or, if $f = 0$,

$$\ddot{\theta}_e + \frac{b}{\mathrm{m}}\dot{\theta}_e + \frac{k}{\mathrm{m}}\theta_e = 0$$



k and b depend on the control law

Modern Robotics, Lynch and Park, Cambridge University Press

For motion control,

reference: $\theta_d(t)$

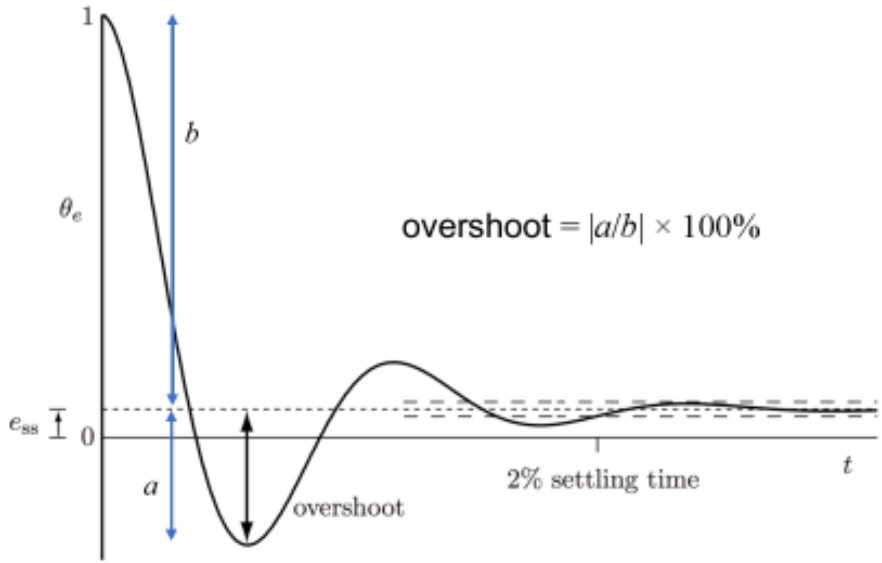
actual: $\theta(t)$

error: $\theta_e(t) = \theta_d(t) - \theta(t)$

Unit step error response:
$\theta_e(t)$ starting from $\theta_e(0) = 1$

Steady-state error response: e_{ss}

Transient error response:
overshoot, settling time



Modern Robotics, Lynch and Park, Cambridge University Press

The characteristic equation

a linear ODE with constant coefficients, for the error θ_e:

$$a_p\,\theta_e^{(p)} + a_{p-1}\,\theta_e^{(p-1)} + \cdots + a_1\dot{\theta}_e + a_0\theta_e = 0$$

substitute $\theta_e(t) = e^{st}$ — each derivative brings down an s:

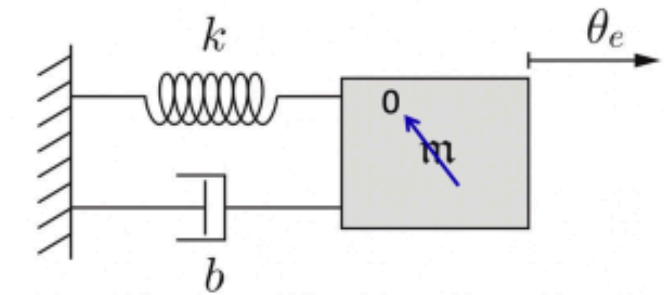
$$\left(a_p s^p + a_{p-1}s^{p-1} + \cdots + a_1 s + a_0\right) e^{st} = 0$$

$e^{st} \neq 0$, so $\quad a_p s^p + a_{p-1}s^{p-1} + \cdots + a_1 s + a_0 = 0$

p roots $s_i \Rightarrow \theta_e(t) = \sum_i c_i\, e^{s_i t}$ (distinct roots) · stable iff every $\mathrm{Re}(s_i) < 0$

all $a_i > 0$: necessary — and sufficient for first and second order

First-order error dynamics



$$m\ddot{\theta}_e + b\dot{\theta}_e + k\theta_e = 0$$

$$\dot{\theta}_e(t) + \frac{k}{b}\theta_e(t) = 0$$

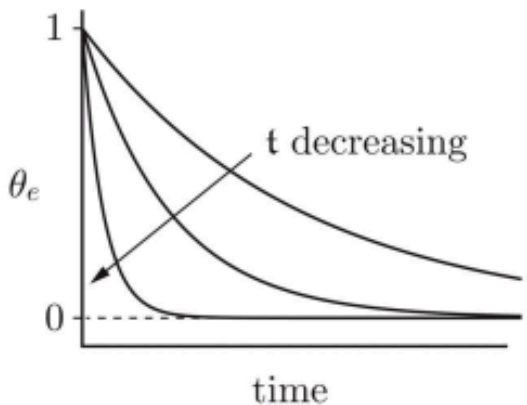
standard first-order form

time constant

$$t = b/k$$

$$\dot{\theta}_e(t) + \frac{1}{t}\theta_e(t) = 0$$

First-order error dynamics



$$\theta_e(0) = 1$$

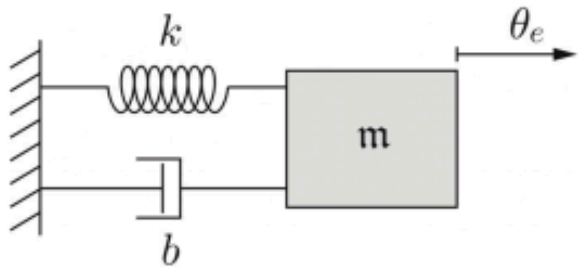
$$\theta_e(t) = e^{-t/\mathrm{t}}\theta_e(0)$$

$$\frac{\theta_e(t)}{\theta_e(0)} = 0.02 = e^{-t/\mathrm{t}}$$

$$\ln 0.02 = -t/\mathrm{t} \quad \rightarrow \quad t = 3.91\mathrm{t}$$

Modern Robotics, Lynch and Park, Cambridge University Press

Second-order error dynamics



$$\ddot{\theta}_e(t) + \frac{b}{\mathrm{m}}\dot{\theta}_e(t) + \frac{k}{\mathrm{m}}\theta_e(t) = 0$$

natural frequency **damping ratio**

$$\omega_n = \sqrt{k/\mathrm{m}} \qquad \zeta = b/(2\sqrt{k\mathrm{m}})$$

$$\ddot{\theta}_e(t) + 2\zeta\omega_n\dot{\theta}_e(t) + \omega_n^2\theta_e(t) = 0$$

standard second-order form

Second-order error dynamics

$$\ddot{\theta}_e(t) + 2\zeta\omega_n\dot{\theta}_e(t) + \omega_n^2\theta_e(t) = 0$$

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

$\zeta > 1$: **Overdamped**

$\zeta = 1$: **Critically damped**

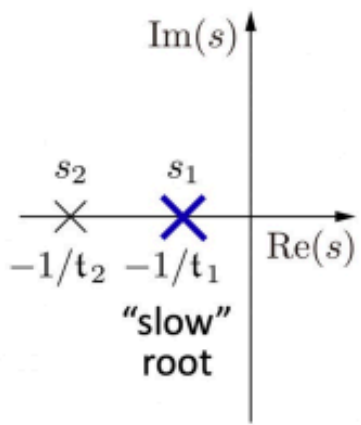
$\zeta < 1$: **Underdamped**

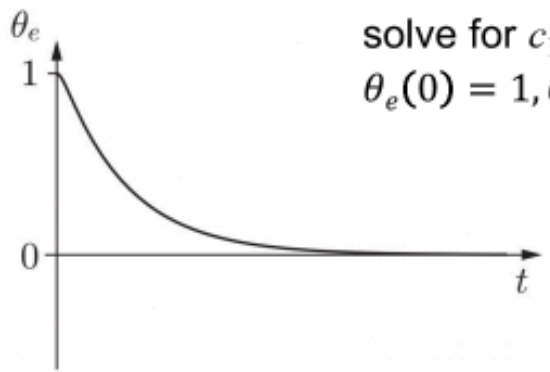
Modern Robotics, Lynch and Park, Cambridge University Press

$\zeta > 1$: Overdamped

$$\theta_e(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$$

$$s_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}$$
$$s_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$



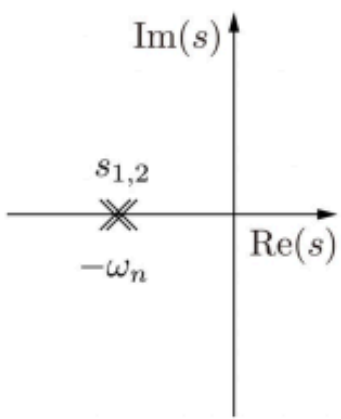


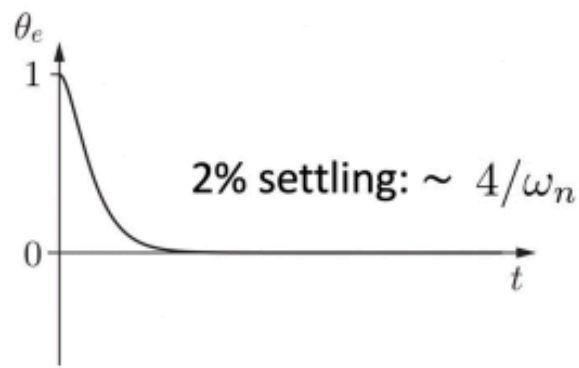
solve for c_1 and c_2 using
$\theta_e(0) = 1, \dot{\theta}_e(0) = 0$

Modern Robotics, Lynch and Park, Cambridge University Press

$\zeta = 1$: Critically damped

$$\theta_e(t) = (c_1 + c_2 t)e^{-\omega_n t} \qquad s_{1,2} = -\omega_n$$





2% settling: ~ $4/\omega_n$

Modern Robotics, Lynch and Park, Cambridge University Press

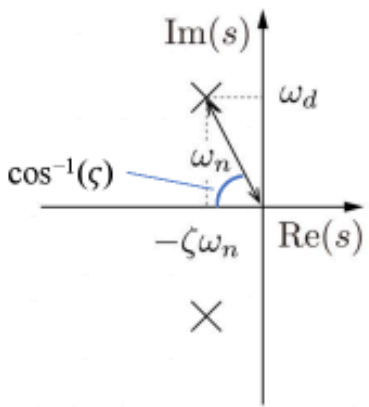
$\zeta < 1$: Underdamped

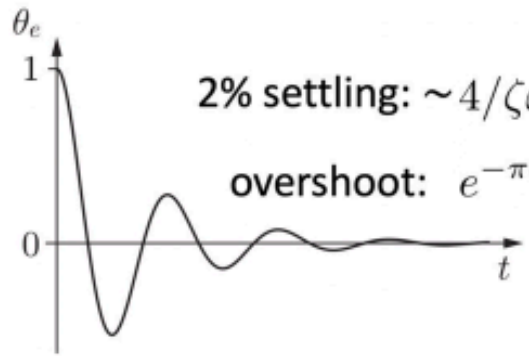
$$\theta_e(t) = (c_1 \cos \omega_d t + c_2 \sin \omega_d t)\, e^{-\zeta \omega_n t}$$

damped natural frequency

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

$$s_{1,2} = -\zeta \omega_n \pm j \omega_d$$

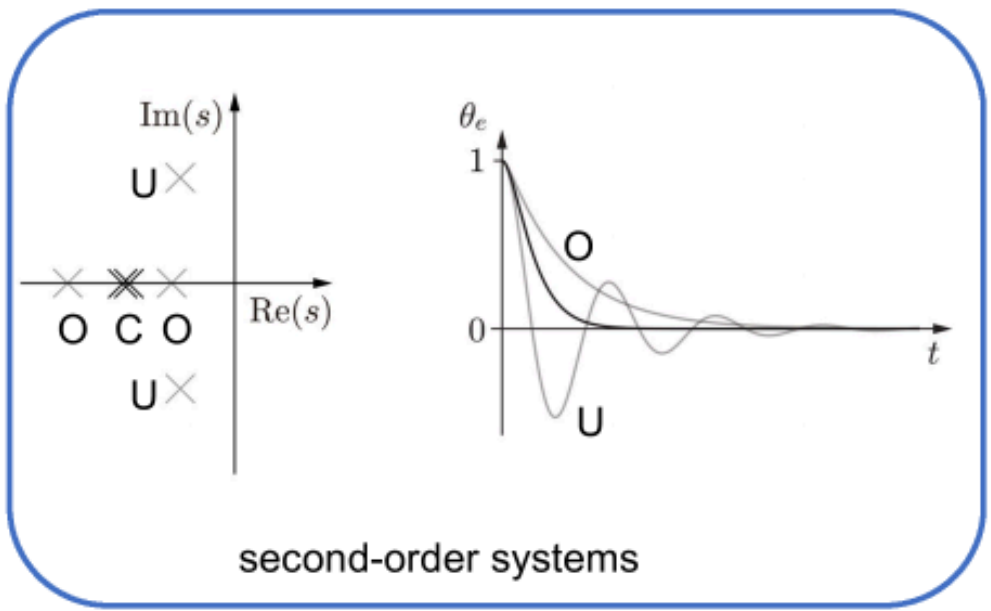


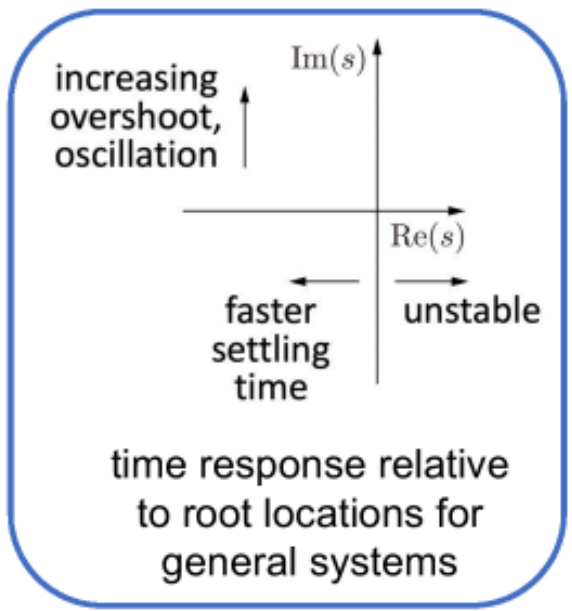


2% settling: $\sim 4/\zeta\omega_n$

overshoot: $e^{-\pi\zeta/\sqrt{1-\zeta^2}} \times 100\%$

Modern Robotics, Lynch and Park, Cambridge University Press





Modern Robotics, Lynch and Park, Cambridge University Press

For a single joint with the joint velocity as the control:

- **Open-loop (feedforward) control**: $\dot{\theta}(t) = \dot{\theta}_d(t)$
- **Closed-loop (feedback) control**: $\dot{\theta}(t) = f(\theta_d(t), \theta(t))$
- **FF + Proportional-Integral (PI) FB control**:

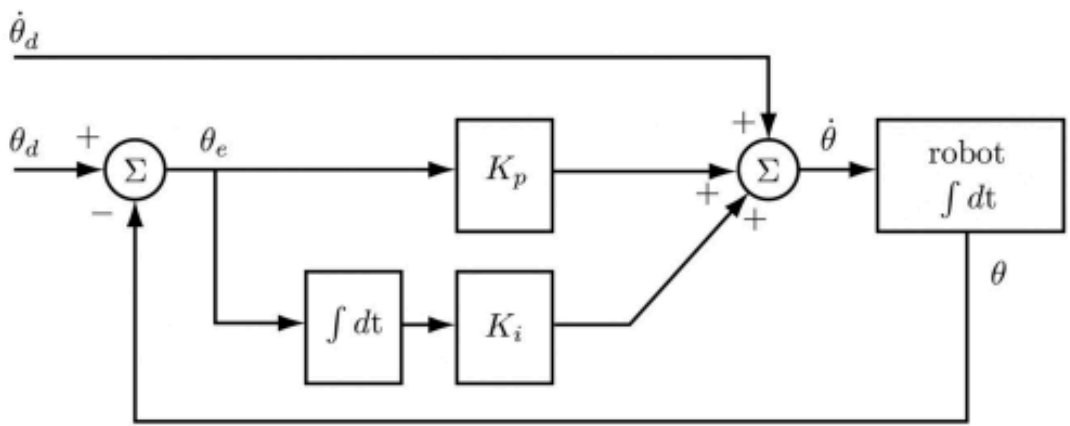
$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\mathrm{t})\, d\mathrm{t}\ ,\ K_p, K_i \geq 0$$

What is the point of FF control in this control law?

- *reduces to FF control if* $K_p, K_i = 0$
- *if no FF term:* **P control** *when* $K_i = 0$, **I control** *when* $K_p = 0$

Modern Robotics, Lynch and Park, Cambridge University Press

Block diagram



$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\mathrm{t})\, d\mathrm{t}$$

Modern Robotics, Lynch and Park, Cambridge University Press

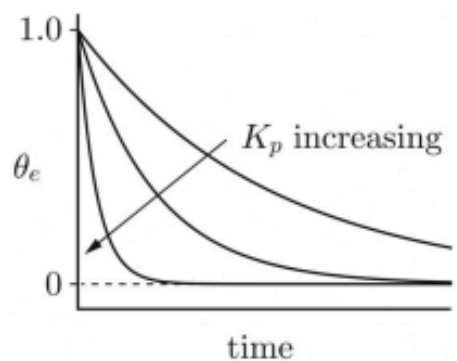
Setpoint control, $\theta_d(t) = c$, with a P controller

$$\dot{\theta}_e(t) = \cancelto{0}{\dot{\theta}_d(t)} - \dot{\theta}(t)$$

$$\dot{\theta}_e(t) = -K_p\theta_e(t) \quad \rightarrow \quad \dot{\theta}_e(t) + K_p\theta_e(t) = 0$$

$$\mathrm{t} = 1/K_p$$

$$\theta_e(t) = e^{-t/\mathrm{t}}\theta_e(0)$$



Modern Robotics, Lynch and Park, Cambridge University Press

Constant velocity control, $\theta_d(t) = ct + a$

P control

$$\dot{\theta}_e(t) = \dot{\theta}_d(t) - \dot{\theta}(t) = c - K_p\theta_e(t)$$

$$\dot{\theta}_e(t) + K_p\theta_e(t) = c$$

$$\theta_e(t) = \frac{c}{K_p} + \left(\theta_e(0) - \frac{c}{K_p}\right) e^{-K_p t}$$

PI control

$$\dot{\theta}(t) = K_p\theta_e(t) + K_i \int_0^t \theta_e(\mathrm{t})\, d\mathrm{t}$$

$$\dot{\theta}_e(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\mathrm{t})\, d\mathrm{t} = c$$

$$\ddot{\theta}_e(t) + K_p\dot{\theta}_e(t) + K_i\theta_e(t) = 0$$

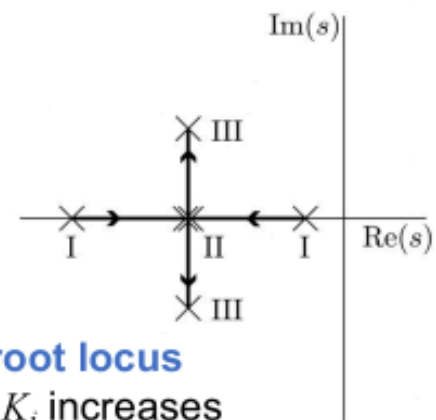
$$\zeta = K_p/(2\sqrt{K_i})$$

$$\omega_n = \sqrt{K_i}$$

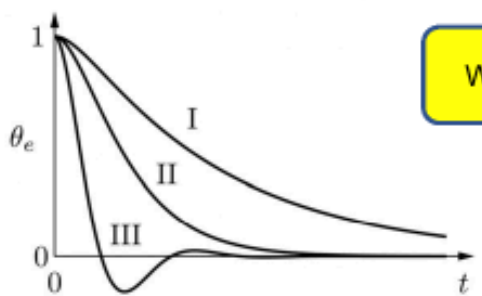
Modern Robotics, Lynch and Park, Cambridge University Press

characteristic equation of PI error dynamics: $s^2 + K_p s + K_i = 0$

$$s_{1,2} = -\frac{K_p}{2} \pm \sqrt{\frac{K_p^2}{4} - K_i}$$

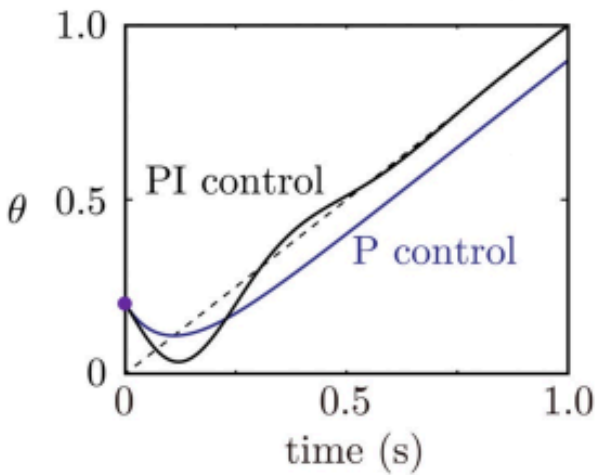


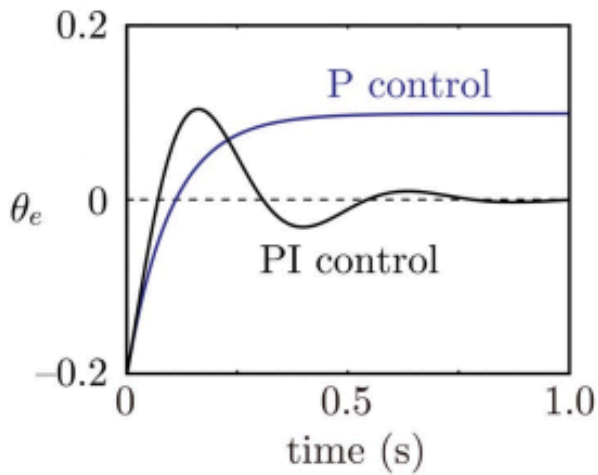
root locus
as K_i increases



What if $K_i = 0$?

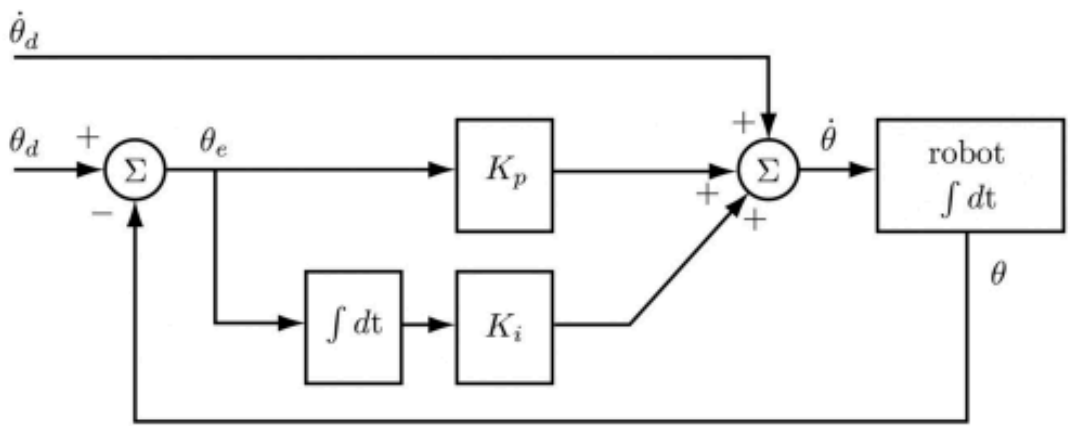
Modern Robotics, Lynch and Park, Cambridge University Press





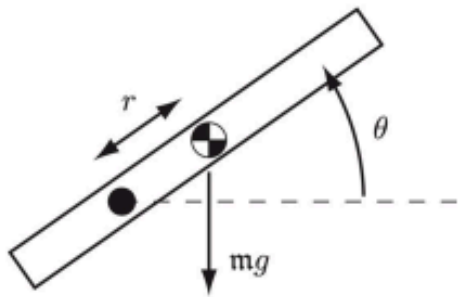
Modern Robotics, Lynch and Park, Cambridge University Press

Multi-joint control: θ, θ_d, and θ_e are vectors and $K_p = k_p I,\ K_i = k_i I$



$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\mathrm{t})\, d\mathrm{t}$$

Modern Robotics, Lynch and Park, Cambridge University Press



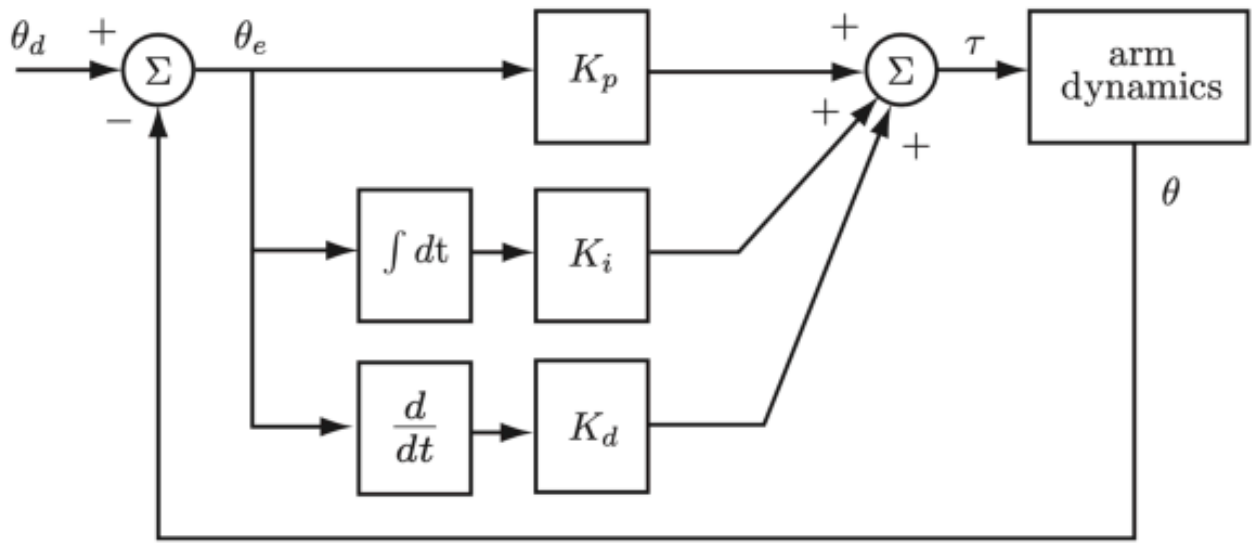
$$\tau = M\ddot{\theta} + \mathfrak{m}gr\cos\theta + b\dot{\theta}$$

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

Proportional-Integral-Derivative (PID) control

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

Modern Robotics, Lynch and Park, Cambridge University Press



Modern Robotics, Lynch and Park, Cambridge University Press

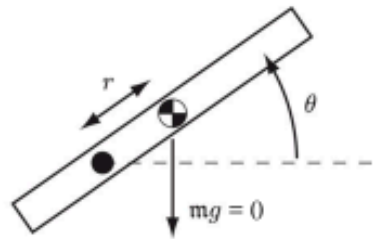
Setpoint PD control, $g = 0$

$$\tau = K_p\theta_e + \cancel{K_i}^{0} \int \theta_e(\mathrm{t})d\mathrm{t} + K_d\dot{\theta}_e$$

$$\tau = M\ddot{\theta} + \cancel{\mathrm{m}g}^{0} r\cos\theta + b\dot{\theta}$$

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

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



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

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

For what gains are the error dynamics stable?

Modern Robotics, Lynch and Park, Cambridge University Press

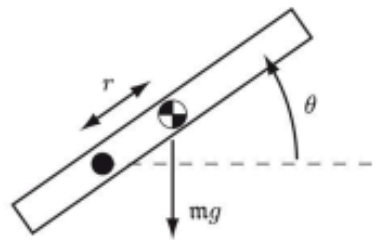
Setpoint PD control, $g \neq 0$

$$\tau = K_p\theta_e + \cancel{K_i}^{0} \int \theta_e(\mathrm{t})d\mathrm{t} + K_d\dot{\theta}_e$$

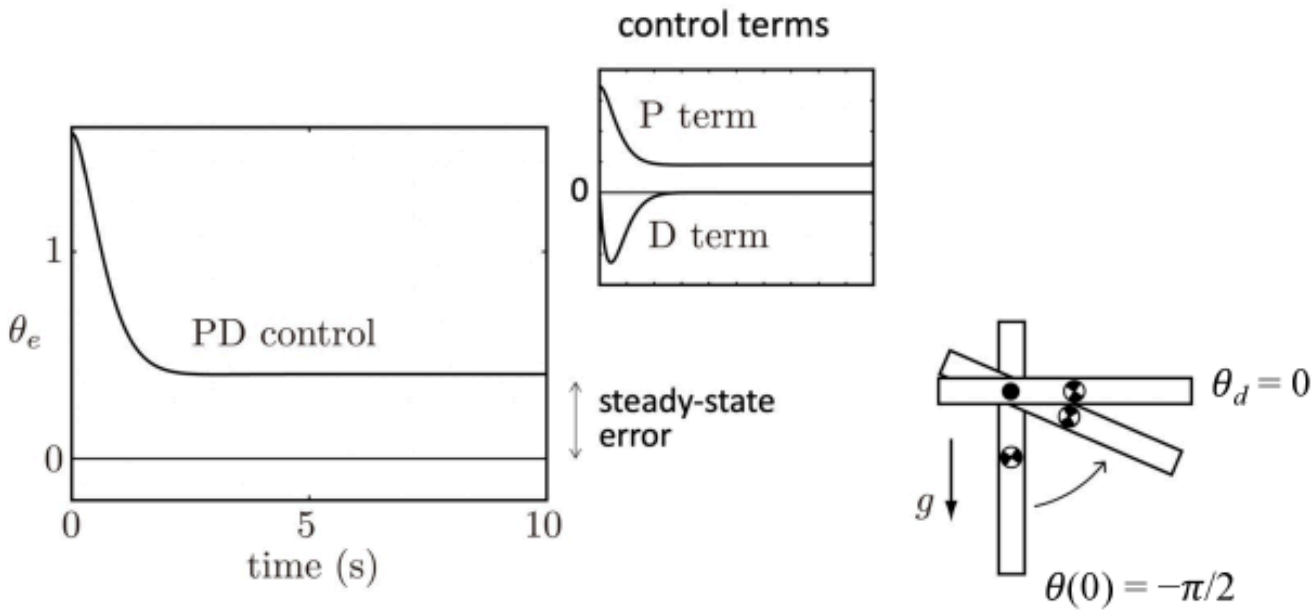
$$\tau = M\ddot{\theta} + \mathfrak{m}gr\cos\theta + b\dot{\theta}$$

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

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



Nonhomogeneous. What is the steady-state error?



Modern Robotics, Lynch and Park, Cambridge University Press

Setpoint PID control, $g \neq 0$

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

$$\tau = M\ddot{\theta} + \underbrace{mgr\cos\theta}_{\tau_{\text{dist}}} + b\dot{\theta}$$

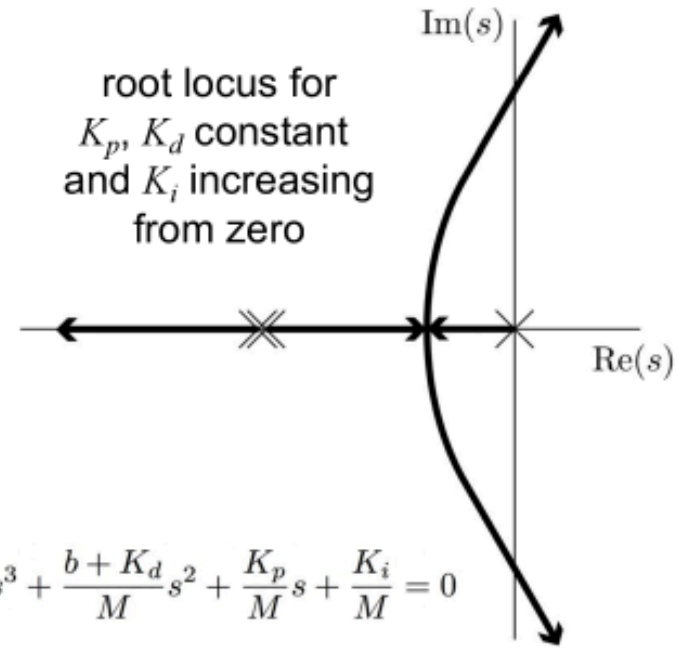
$$\dot{\theta}_d = \ddot{\theta}_d = 0$$

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

$$M\theta_e^{(3)} + (b + K_d)\ddot{\theta}_e + K_p\dot{\theta}_e + K_i\theta_e = 0$$

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

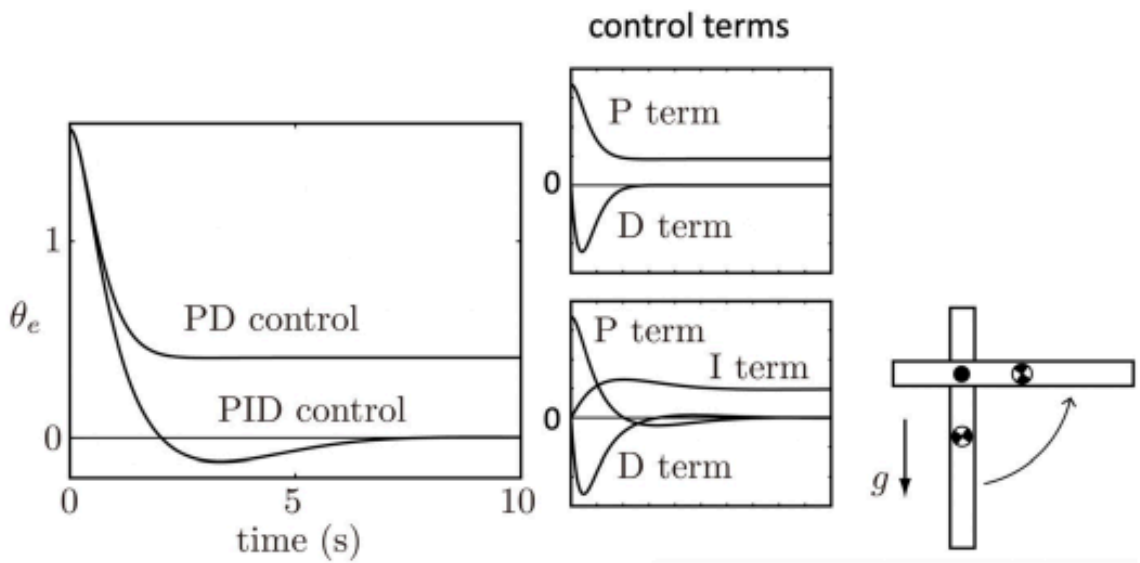
Modern Robotics, Lynch and Park, Cambridge University Press



$$K_d > -b$$
$$K_p > 0$$
$$\frac{(b+K_d)K_p}{M} > K_i > 0$$

K_i improves steady-state response but can worsen the transient response.

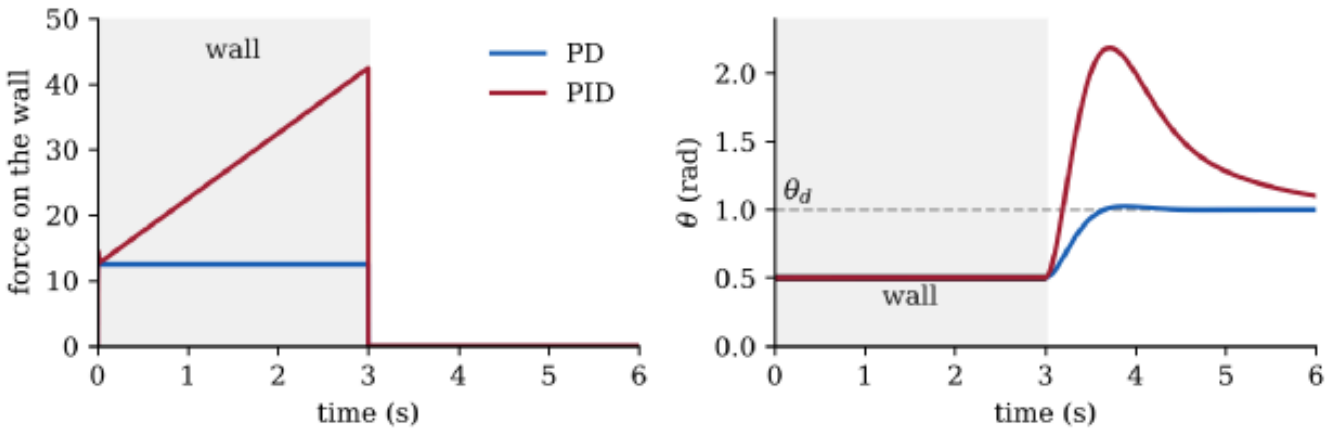
Modern Robotics, Lynch and Park, Cambridge University Press



Modern Robotics, Lynch and Park, Cambridge University Press

Integral windup

the integral removes steady-state error … by integrating *any* error



against a wall at 0.5 rad, $\theta_d = 1$ rad · wall removed at 3 s

PD: a constant push · PID: harder every second, then 1.2 rad of overshoot

in manipulation we usually set $K_i = 0$

$M = 1, K_p = 25, K_d = 7, K_i = 20$

Tell the controller what you know

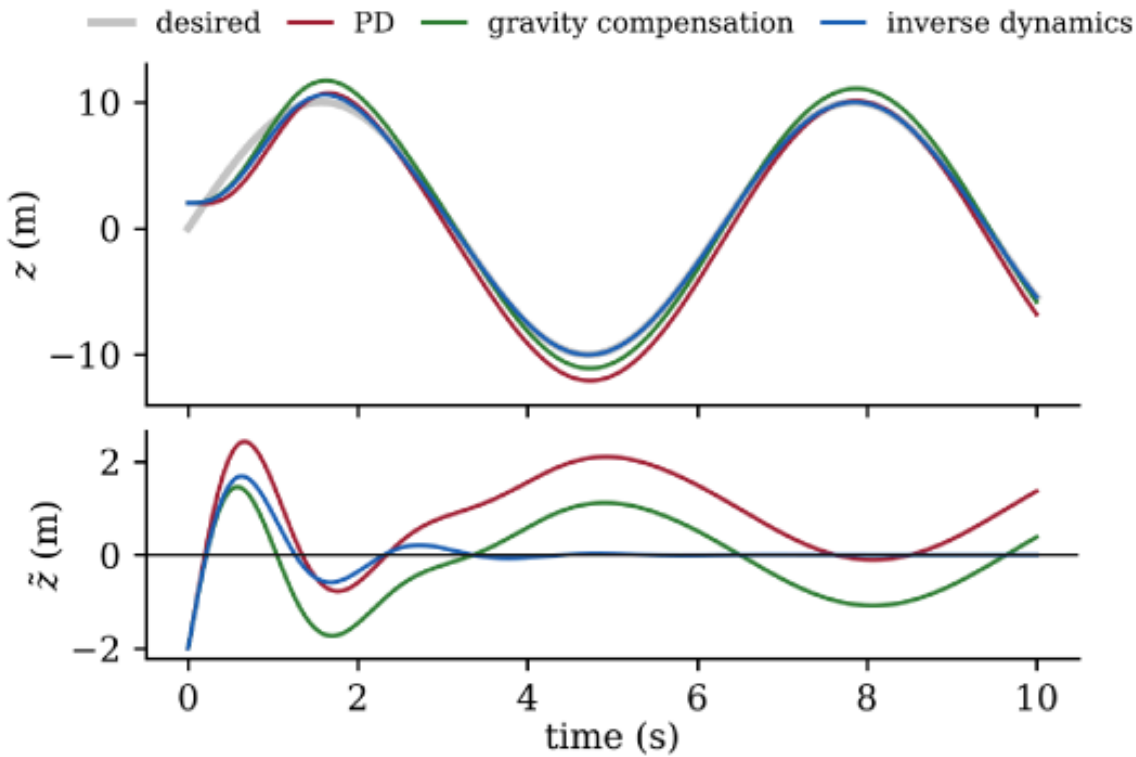
the joint: $\tau = M\ddot{\theta} + h(\theta, \dot{\theta}), \quad h = mgr\cos\theta + b\dot{\theta}$

PD	$\tau = K_p\theta_e + K_d\dot{\theta}_e$	sags, lags
gravity compensation	$\tau = mgr\cos\theta + K_p\theta_e + K_d\dot{\theta}_e$	no sag
inverse dynamics	$\tau = M[\ddot{\theta}_d + K_p\theta_e + K_d\dot{\theta}_e] + h(\theta, \dot{\theta})$	no lag

inverse dynamics leaves $\ddot{\theta}_e + K_d\dot{\theta}_e + K_p\theta_e = 0$ — whatever $\theta_d(t)$ does

gravity compensation needs no error to hold the arm up: lower gains

Tracking a sinusoid



point finger, $z_d = 10 \sin t$

Feedback corrects what you did not predict.

Feedforward supplies what you did.

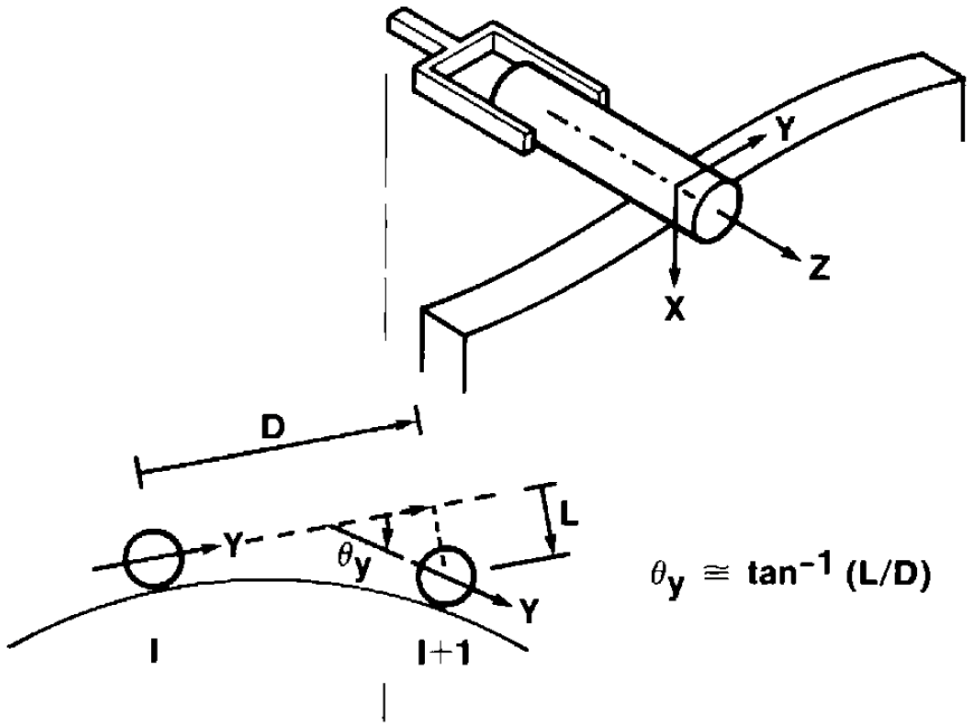
Part II

force control


CHEEZ-IT
original
JELL-O
40% less
Fruits

https://www.youtube.com/embed/XqRczI4Fepk?mute=1&rel=0&enablejsapi=1

EDGE FOLLOWING



D. E. Whitney, “Historical Perspective and State of the Art in Robot Force Control,” *The International Journal of Robotics Research*, vol. 6, no. 1, pp. 3–14, Mar. 1987.

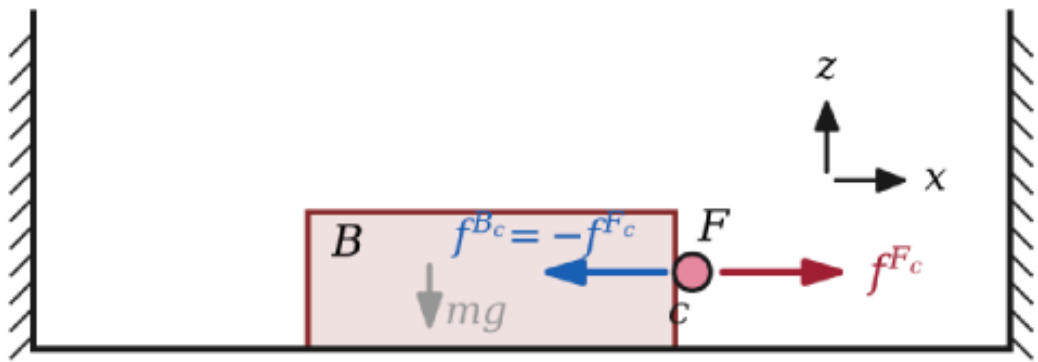


Inoue 1974 (at MIT AI Lab)

https://www.youtube.com/embed/PngdQGEUi7w?mute=1&rel=0&enablejsapi=1

The simplest robot: a point finger

$$m\ddot{q} = \underbrace{\tau_g}_{\text{gravity}} + \underbrace{u}_{\text{what we command}} + \underbrace{f^{F_c}}_{\text{contact}}, \qquad q = \begin{bmatrix} x \\ z \end{bmatrix}, \quad \tau_g = \begin{bmatrix} 0 \\ -mg \end{bmatrix}$$



simulate finger + box + bin (5 dof) · control the finger alone (2 dof)

Direct force control

assume slow: $\ddot{q} \approx 0$, so $f^{F_c} = -\tau_g - u$

$$u = -\tau_g - f^{F_c}_{desired}$$

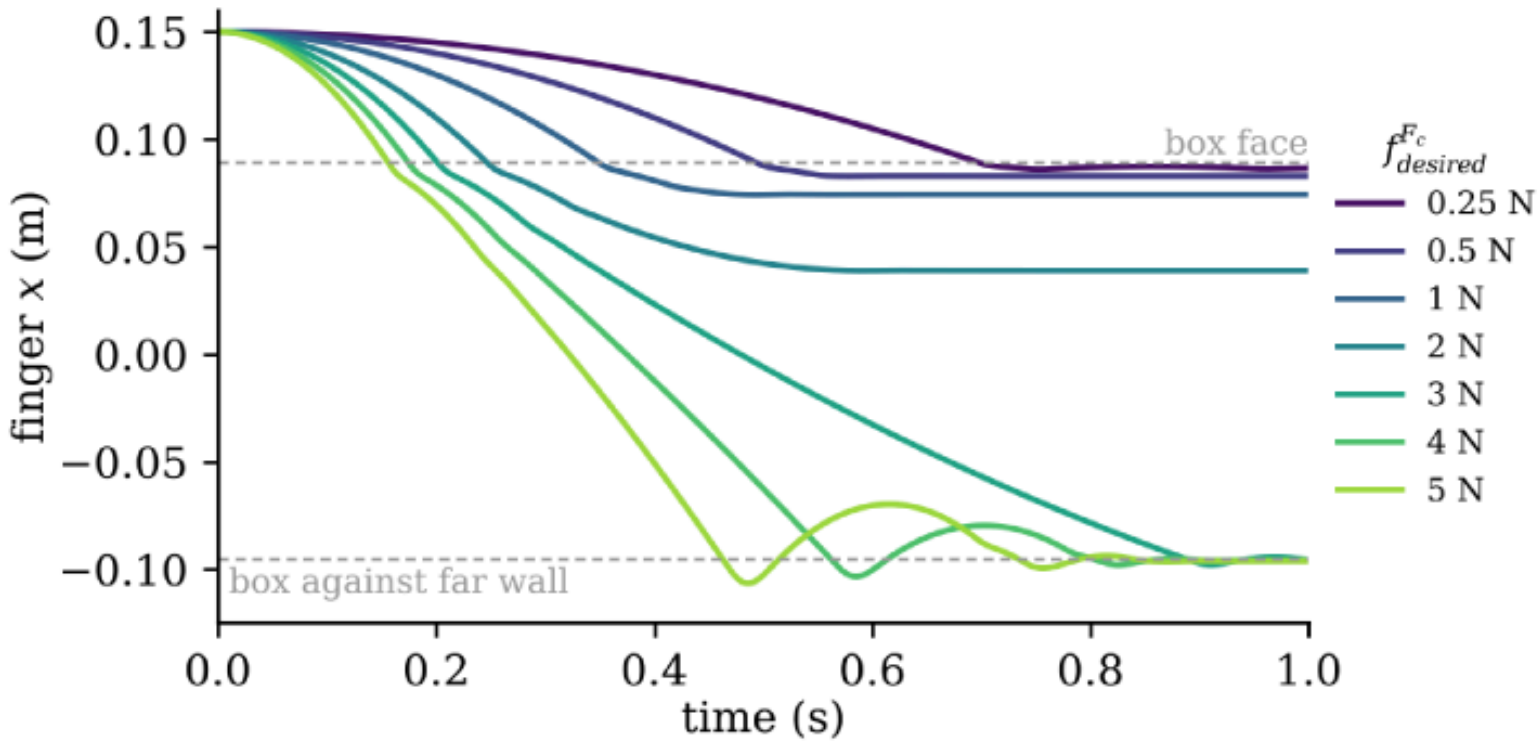
in contact: $f^{F_c} = f^{F_c}_{desired}$

needs the mass of the robot — not the box, not its geometry

out of contact: $m\ddot{q} = -f^{F_c}_{desired}$

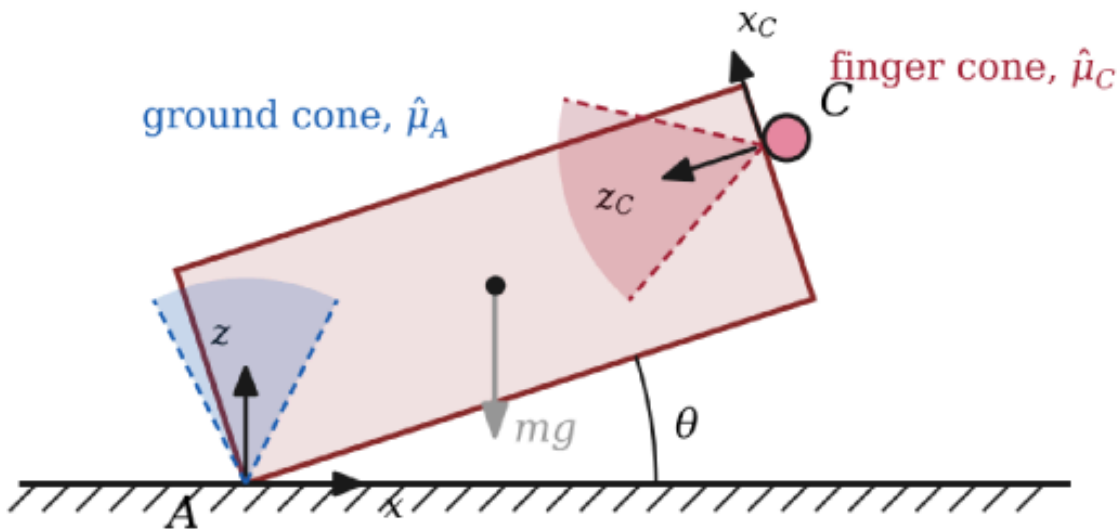
not getting the force you asked for moves you into contact

A constant force command



small: box stays · medium: moves and stops · large: pushed to the wall

Flip the box with forces



$$f_{C_z}^{B_C} \geq 0, \quad |f_{C_x}^{B_C}| \leq \hat{\mu}_C f_{C_z}^{B_C} \qquad f_{A_z}^{B_A} \geq 0, \quad |f_{A_x}^{B_A}| \leq \hat{\mu}_A f_{A_z}^{B_A}$$

...as a small QP

$$\min_{f_{\text{finger}},\, f_{\text{ground}}} \quad \left\| \tau^{B_A}_{\text{finger}} - \text{PID}(\theta_d, \hat{\theta}) \right\|^2$$

$$\text{s.t.} \quad \text{both friction cones}$$

$$f_{\text{ground}} + \hat{f}_{\text{gravity}} + f_{\text{finger}} = 0$$

high level: choose forces from the box angle low level: apply them

needs $\hat{\theta}$, conservative $\hat{\mu}$'s — forgiving about mass and contact location

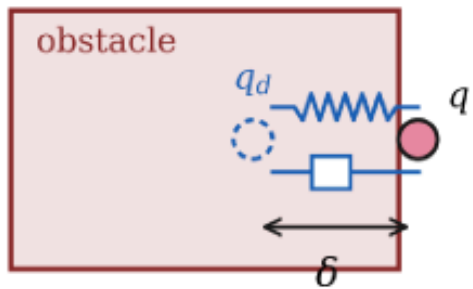

CHEEZ-IT

Indirect force control: program a spring

the finger $m\ddot{q} = \tau_g + u + f^{F_c}$

the spring we want $m\ddot{q} + b\dot{q} + k(q - q_d) = f^{F_c}$

match them $u = -\tau_g + k(q_d - q) - b\dot{q}$



$\|f\| = k\,\delta$

gravity compensation + PD, read differently: *stiffness control*

don't program the force — program how the robot responds to forces

Salisbury 1980; Hogan 1985

Why a spring

- **no integral term** — force does not grow while blocked
- **low gains** — soft, and safe around people
- **passivity** — a passive robot touching a passive world is stable

https://www.youtube.com/embed/1Or0kym2uic?mute=1&start=5&enablejsapi=1

https://www.youtube.com/embed/dnUwqngH0bM?start=35&mute=1&enablejsapi=1

https://www.youtube.com/embed/P-d-yTcxdzw?enablejsapi=1

Impedance $= (m, b, k)$

to feel like mass m_d:

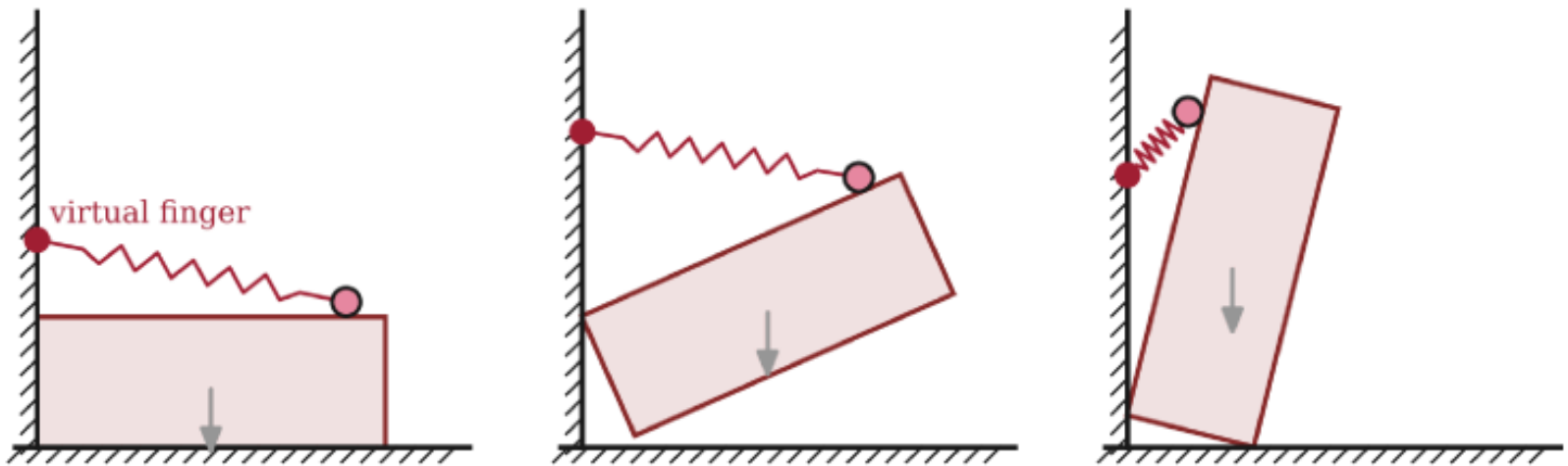
$$u = \frac{m}{m_d}(f_{ext} - b\dot{q} - k(q - q_d)) - f_{ext}$$

needs f_{ext} — a force sensor — unless $m_d = m$

easier to make the robot feel heavier, much harder to make it feel lighter

the iiwa reshapes only its rotor inertia

Flip the box with a spring



1. push into the corner

2. move it up the wall

3. mechanics does the rest

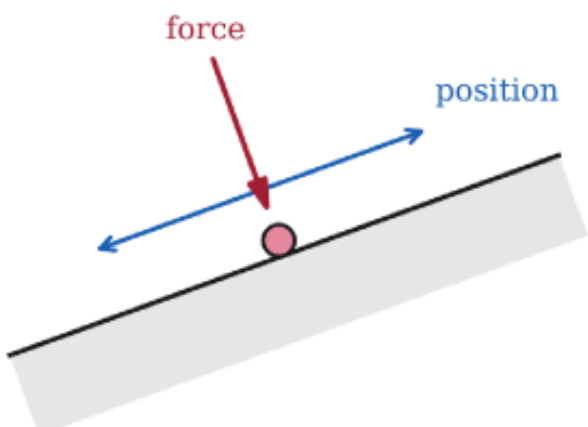
no slip detector, no replanning — just move the virtual finger

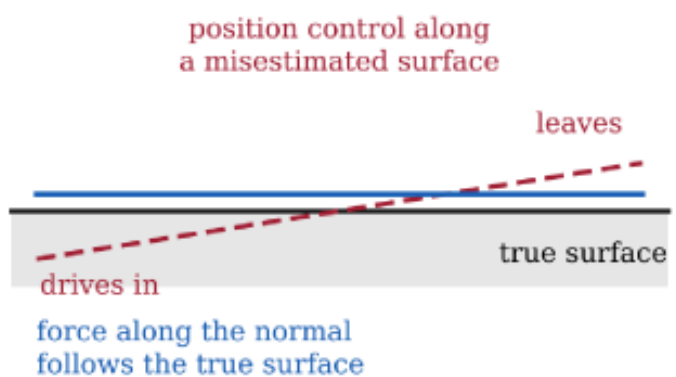
after manipulation.mit.edu, Ch. 8



Hybrid position/force control

$$u = -\tau_g + {}^{W}R^{C} \begin{bmatrix} k_p(x_d - x) + k_d(\dot{x}_d - \dot{x}) \\ -f^{F}_{desired,C_z} \end{bmatrix}$$





force along the normal, position along the surface — and robust to a wrong normal

Blending force and position control

$$u = -\tau_g + K_p(p^{F_d} - p^F) + K_d(v^{F_d} - v^F) - f^F_{\text{feedforward}}$$

$K_p = 0$ in a direction: force control there

$f_{\text{feedforward}} = 0$: stiffness control

close to the interface the iiwa exposes — next time