Advanced Topics for Robotics
智能机器人前沿探究

Lecture 3: Robot Control

Jianyu Chen (陈建宇)

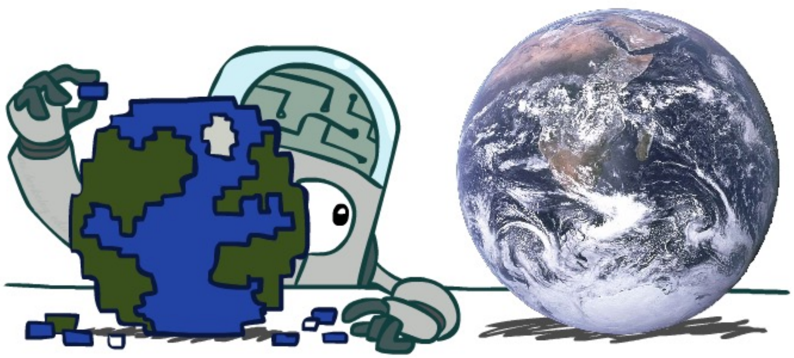
Institute for Interdisciplinary Information Sciences

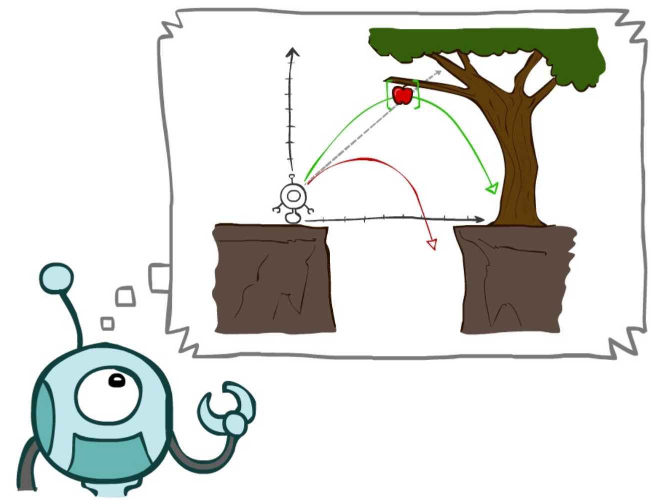
Tsinghua University

Contents

Dynamics and control

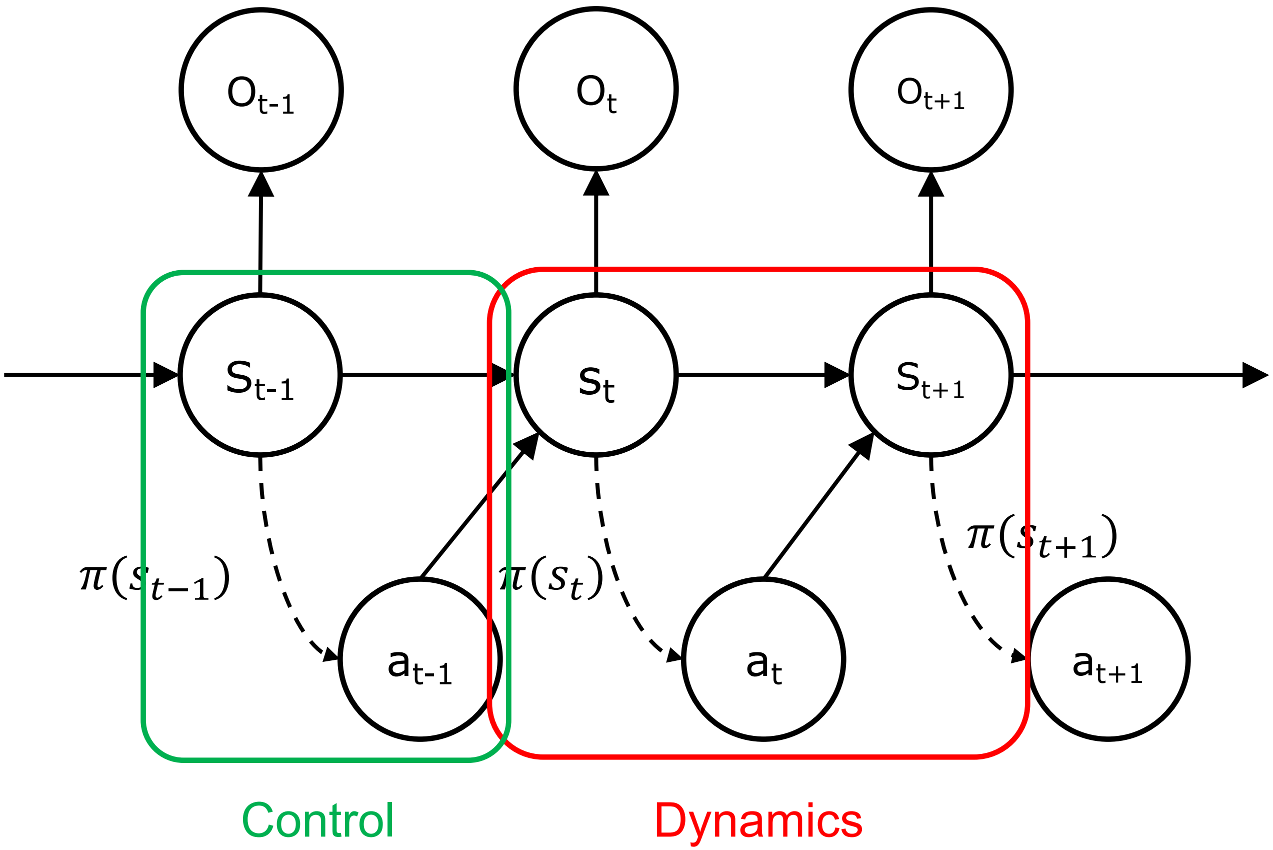
- Dynamics talks about **understanding** the world
- Control talks about **changing** the world





Dynamics and control

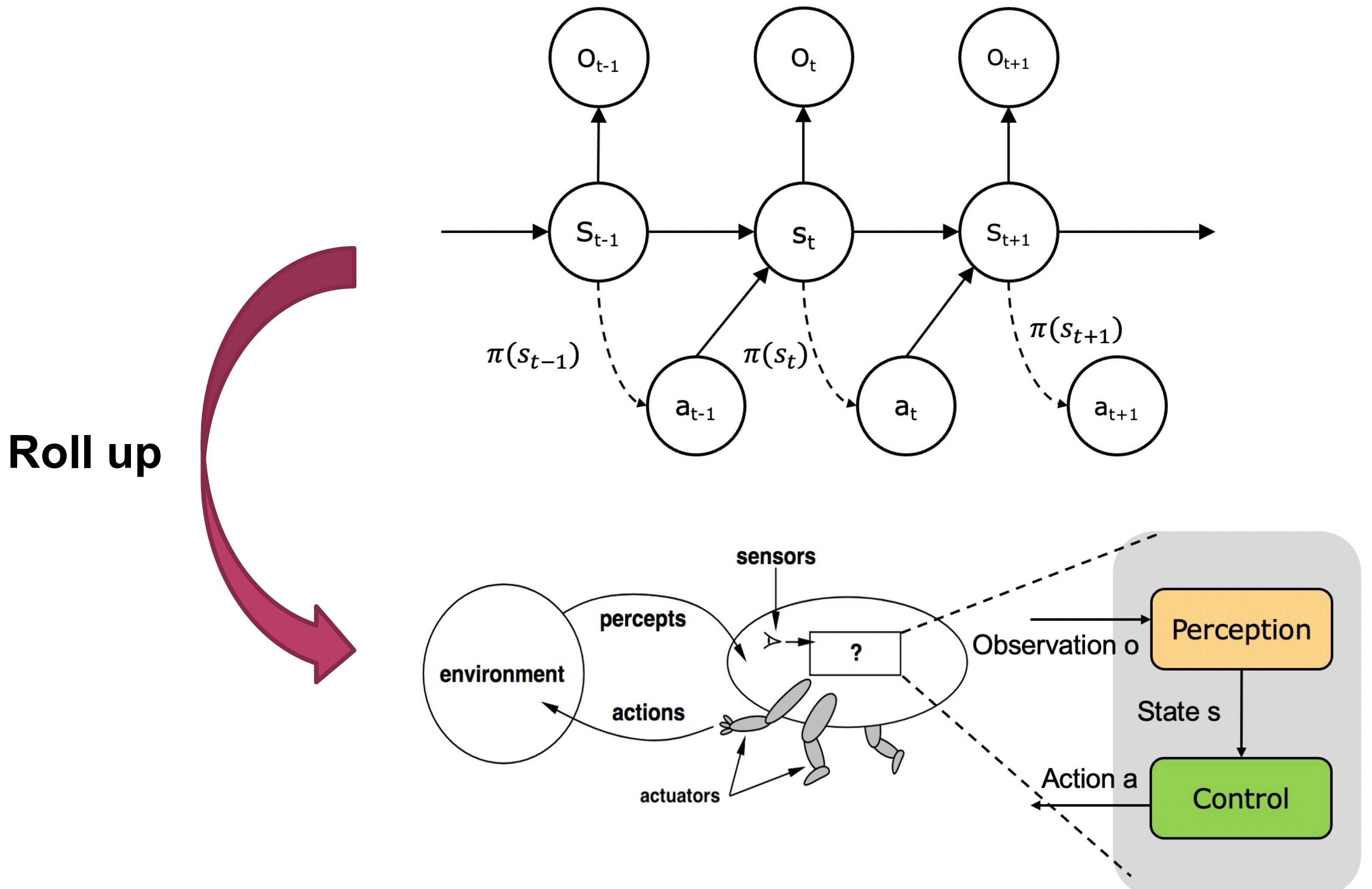
- Control is equal with the **policy**



s: state
a: action
o: observation
π : policy

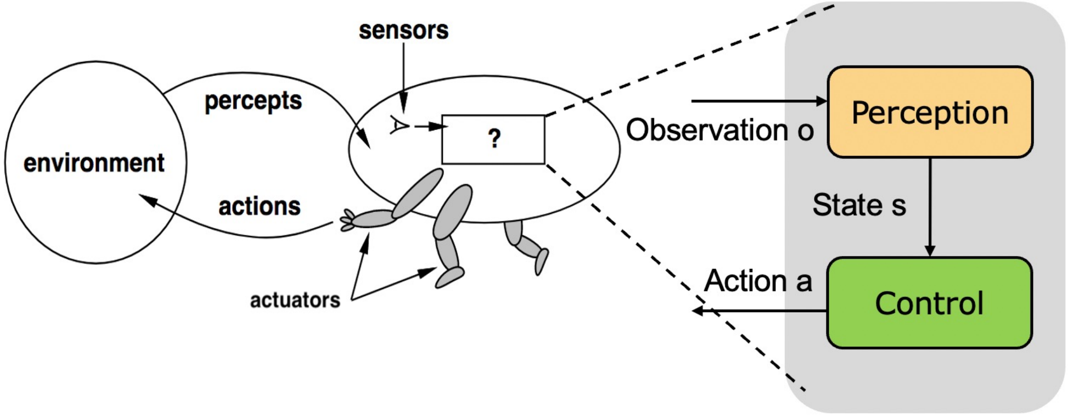
Feedback control

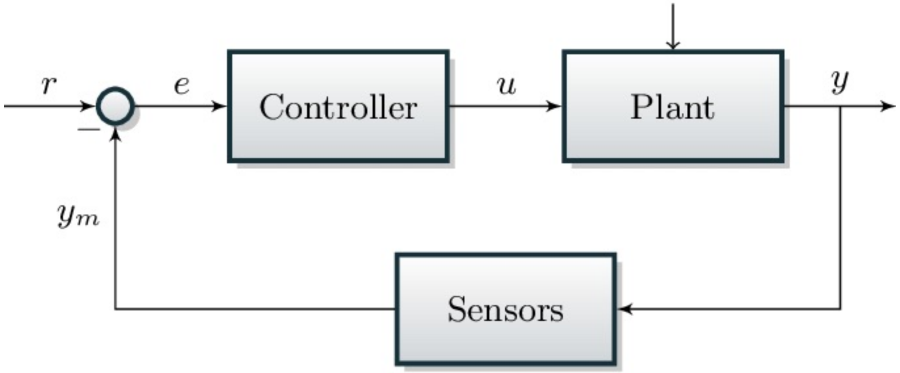
- Policy as a **feedback controller**



Feedback control

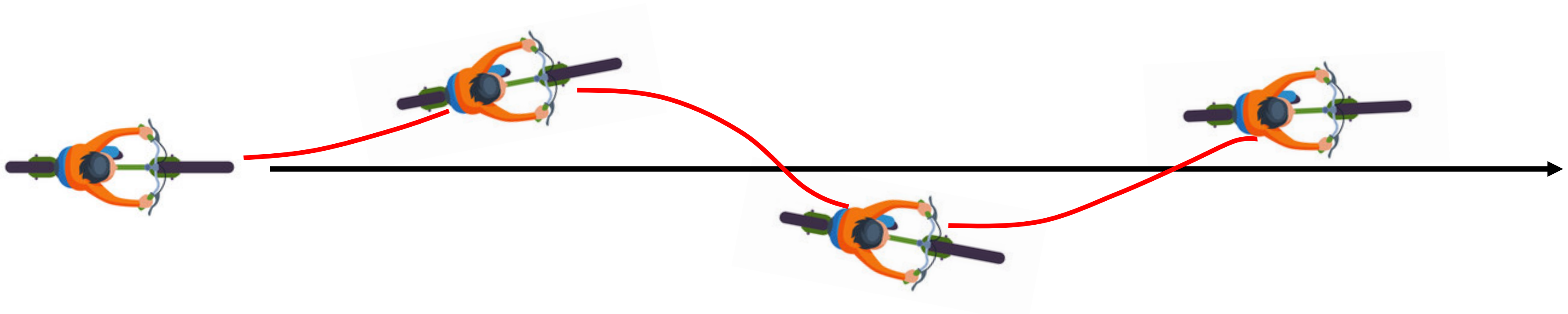
- Feedback control diagram



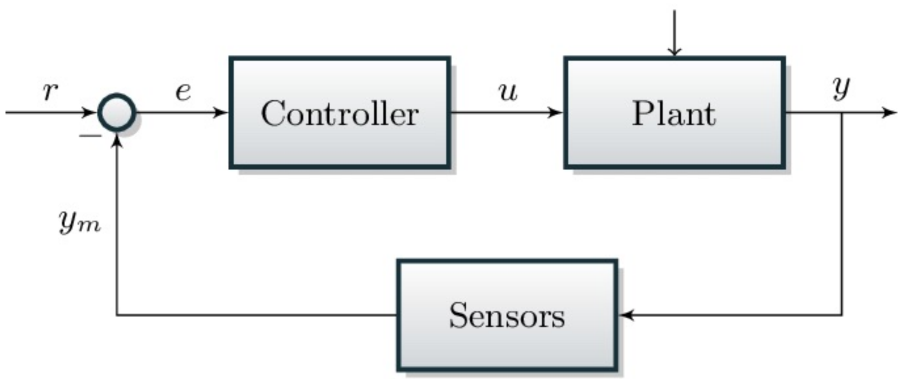


Feedback control

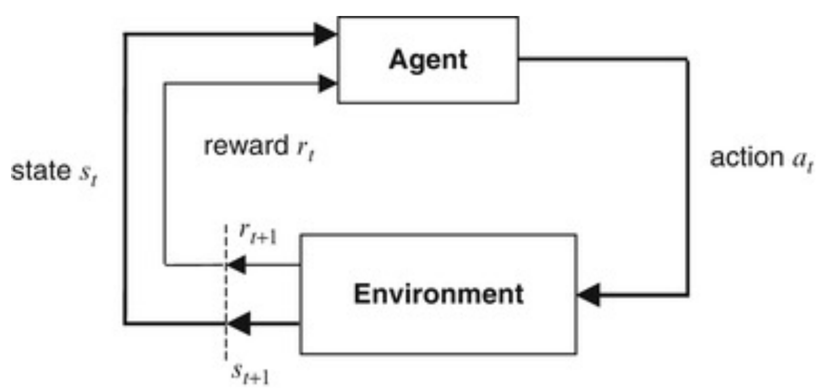
- Think about what's going on when you are riding a bicycle and try to track a line carefully.



- You are using **feedback control**!



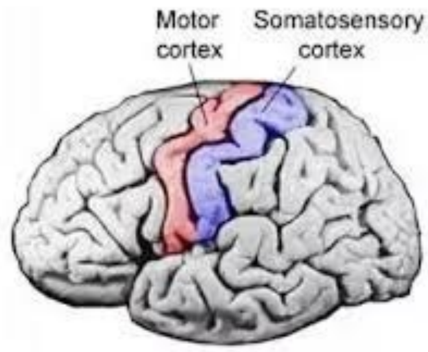
Feedback control diagram

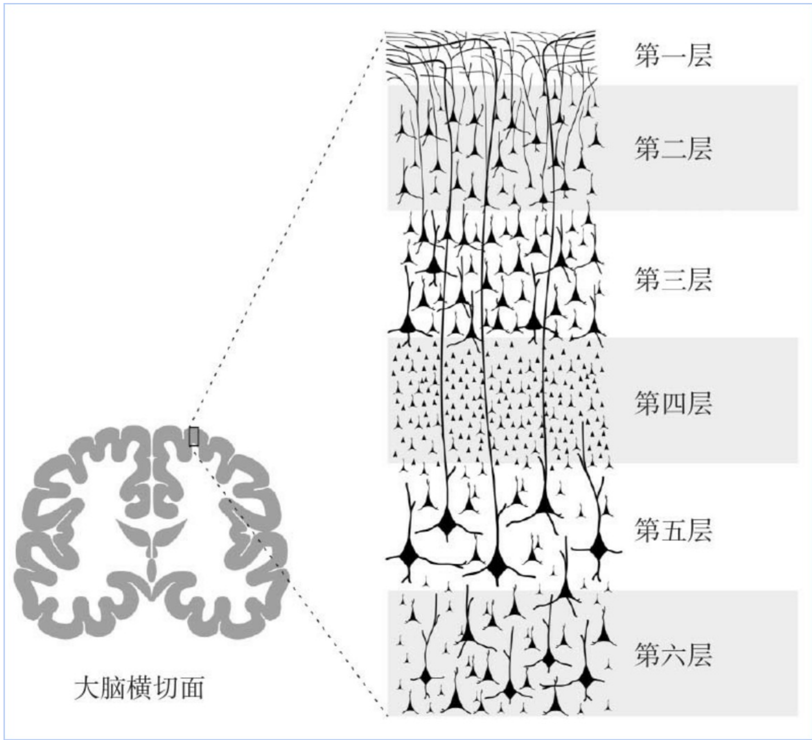


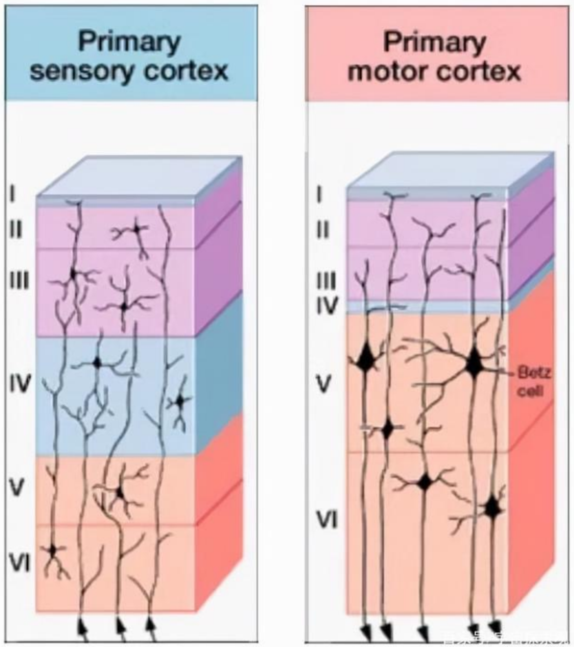
Reinforcement learning diagram

Controller in human brain

- The sensory motor cortex
- “End-to-end” control style

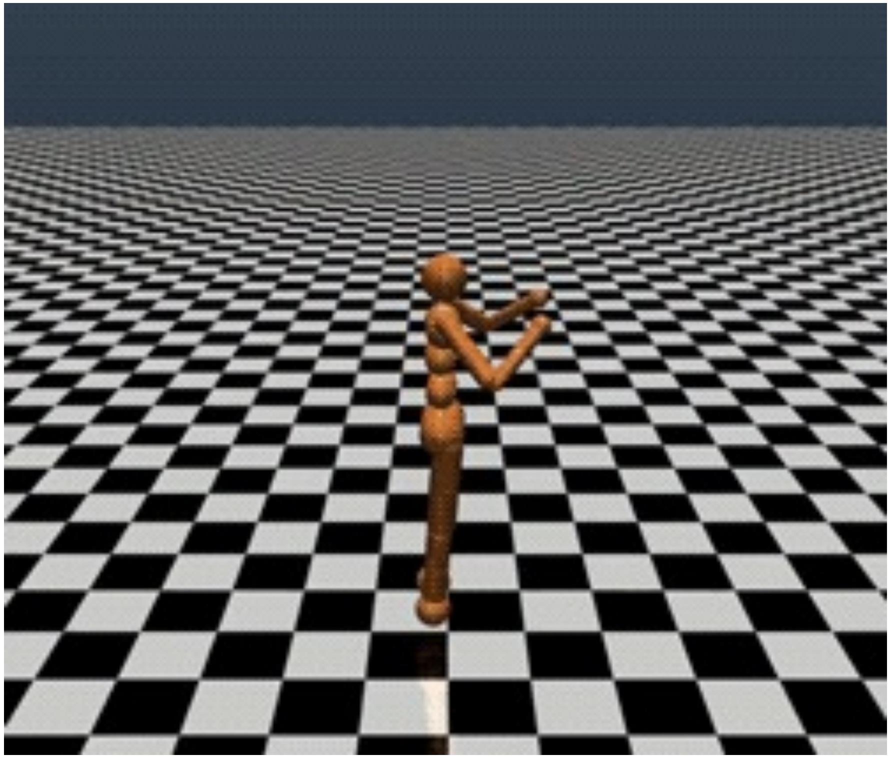






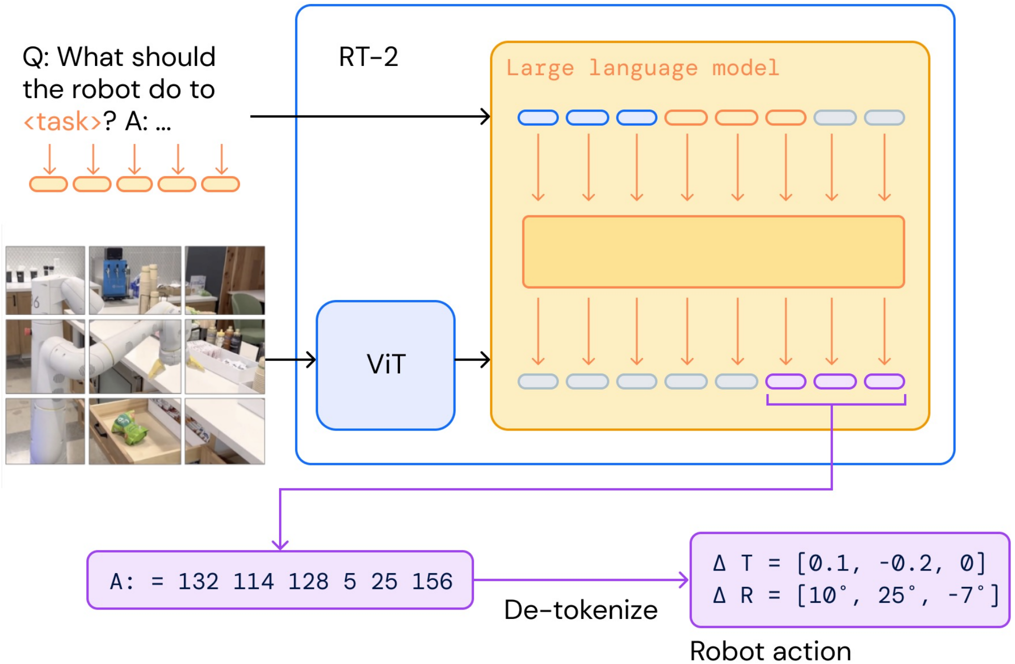
Pure end-to-end control for robot?

- Too big complexity to learn
- Too much computational burden



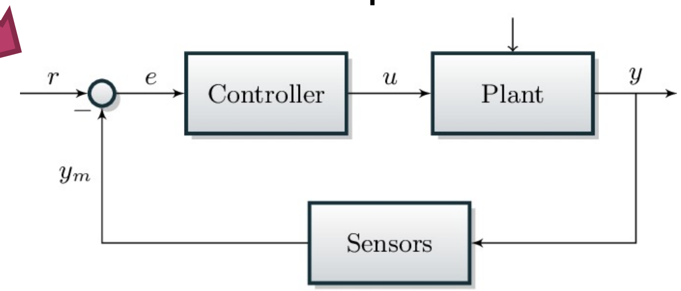
The “End-to-end” control in robotics

Google robot transformer



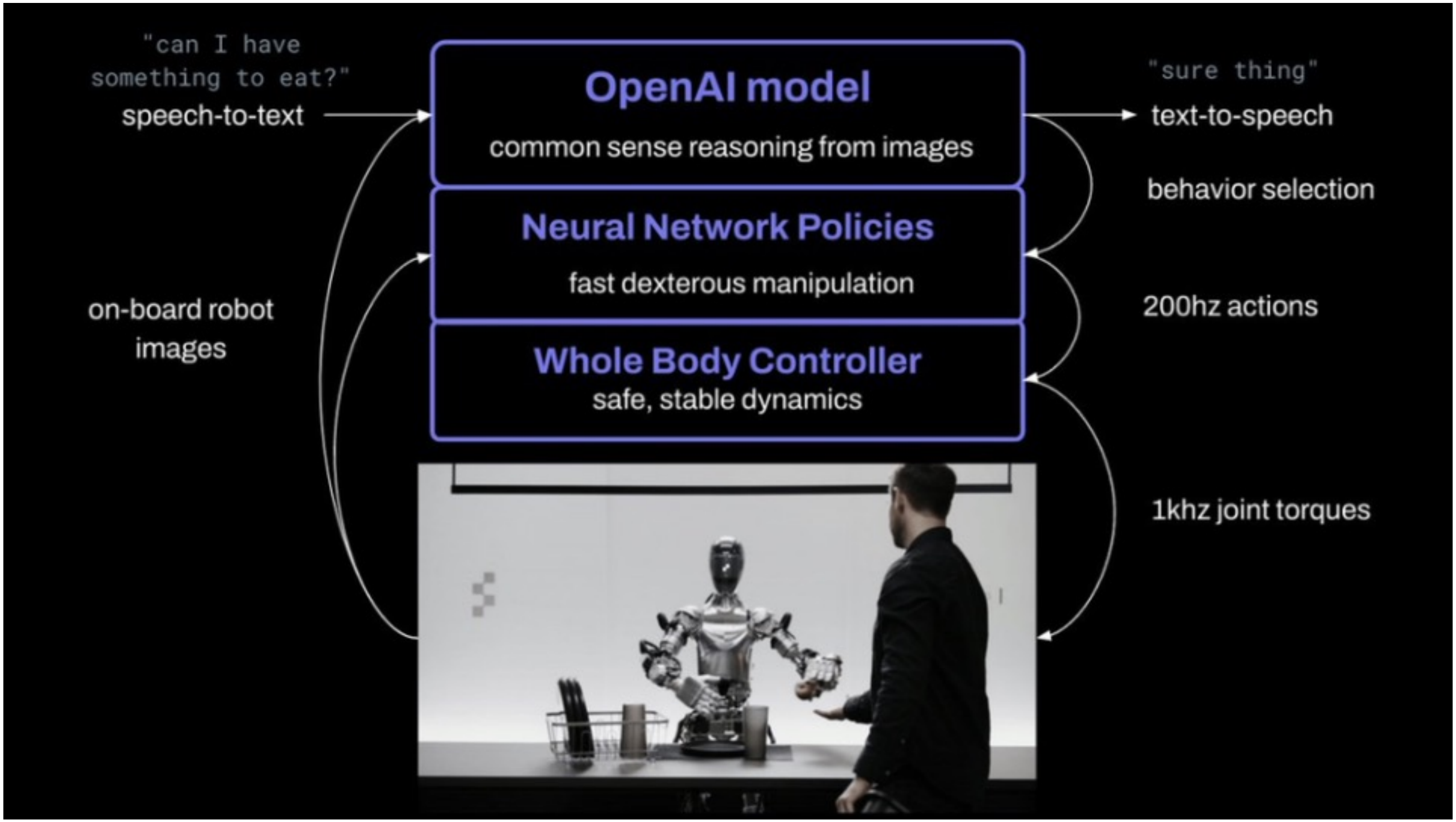
End-effector motion

Low-level controller to torque



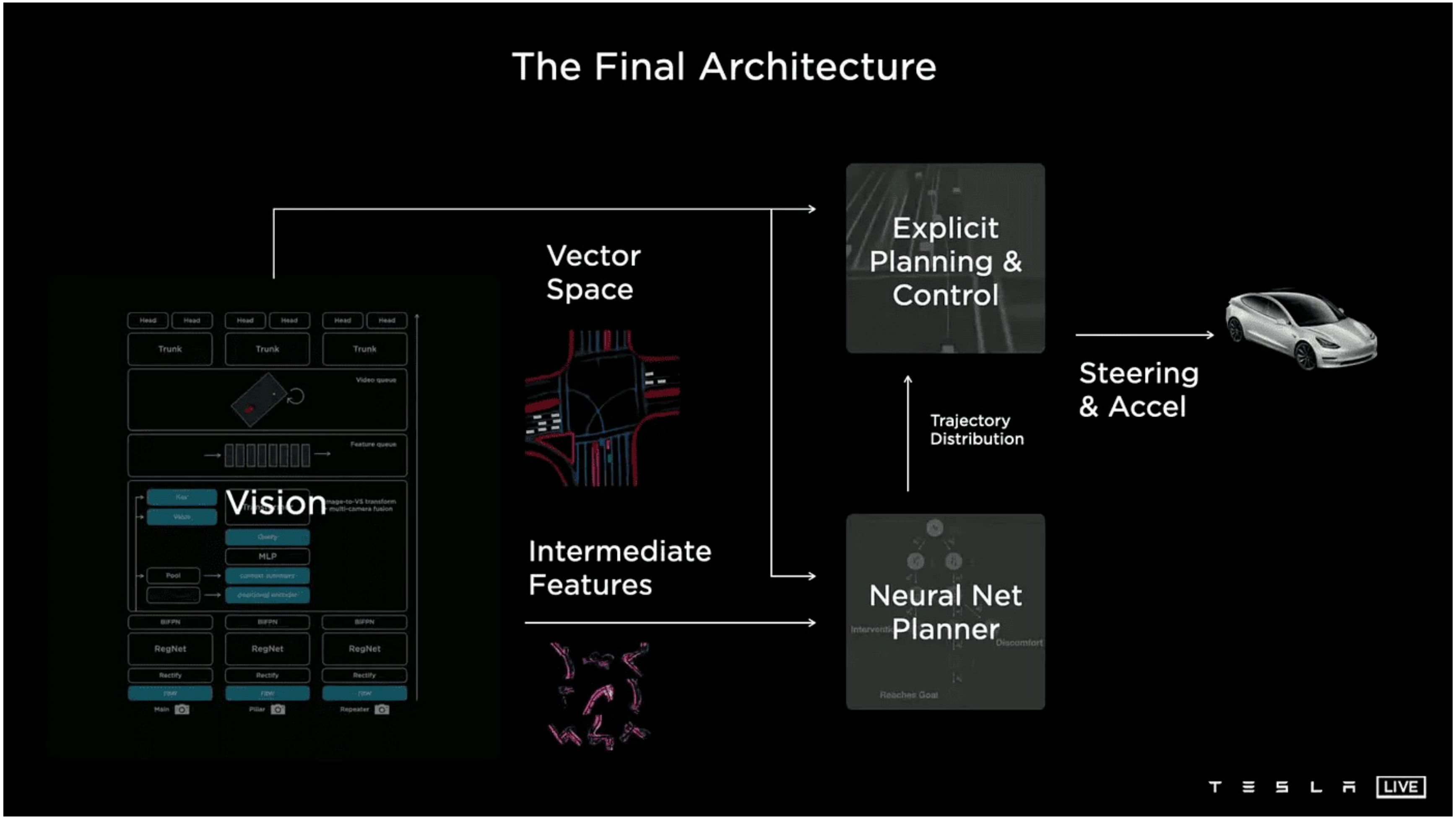
The “End-to-end” control in robotics

- Figure AI



The "End-to-end" control in robotics

- Tesla autopilot

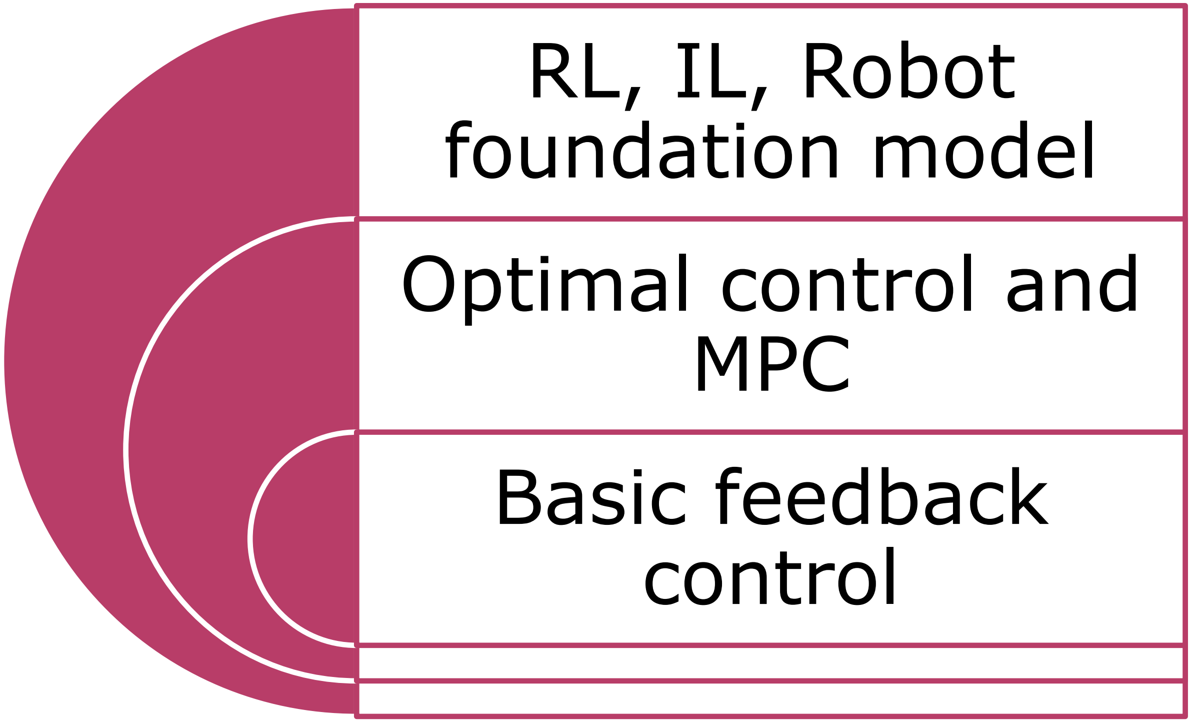


How about pure control?

- Mostly just simple, repetitive motions, but very precise

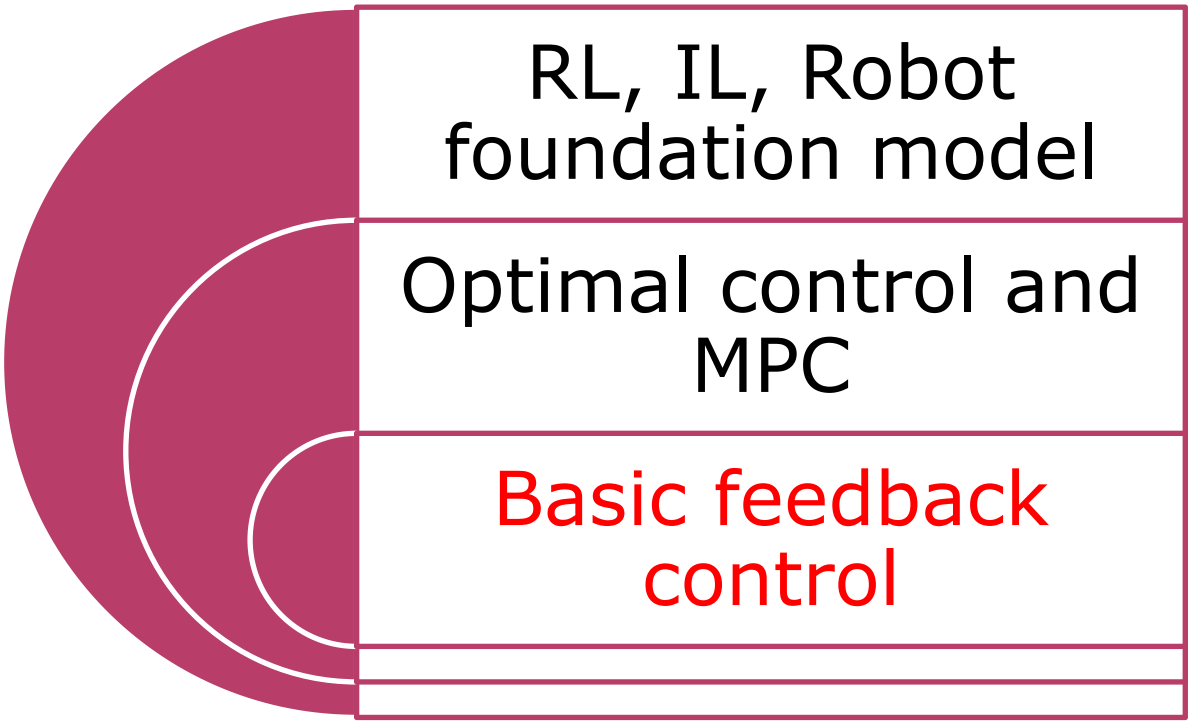
The scope of different robot controllers

- The feedback controller extends to optimization, learning, and LLM.



The scope of different robot controllers

- The feedback controller extends to optimization, learning, and LLM.



Contents

Dynamic equations for robots

- The dynamic equations for robots has a generic form

$$u = ml^2\ddot{\theta} + b\dot{\theta} + mgl\,\sin\theta$$

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

Force

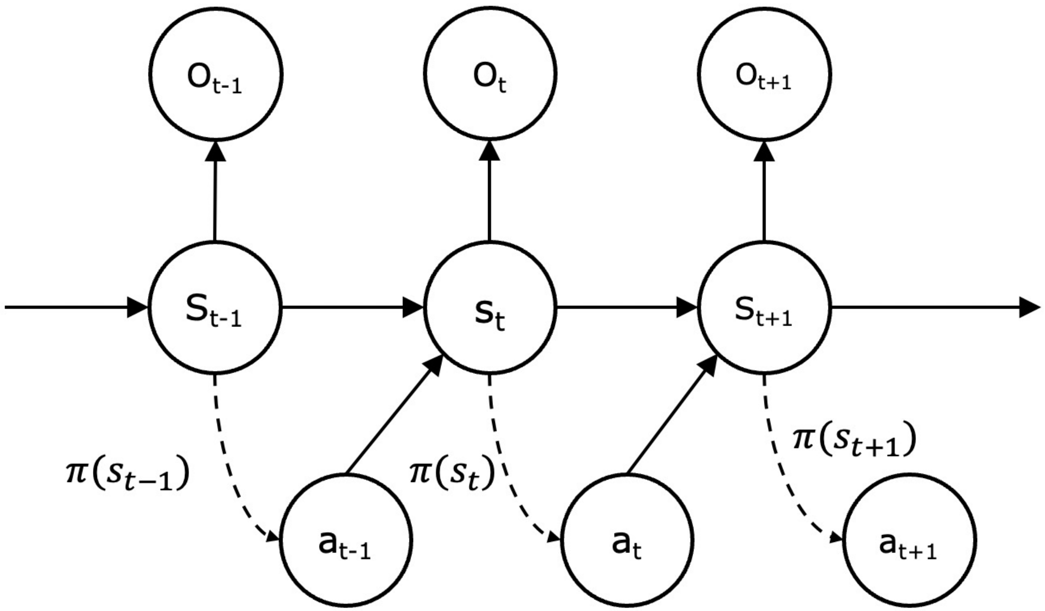
Coriolis & Centripetal

Gravity

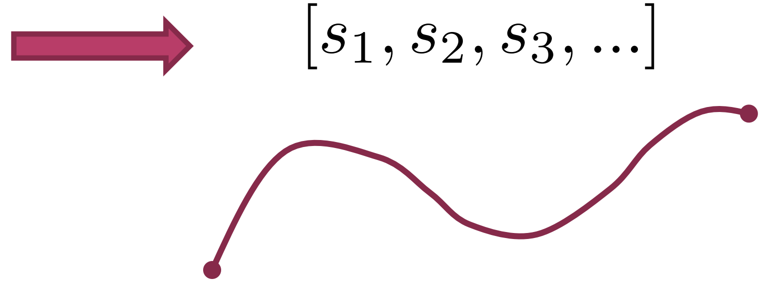
Mass matrix, symmetric and positive definite

System trajectory

- Dynamics decide the robots' **system future trajectories**



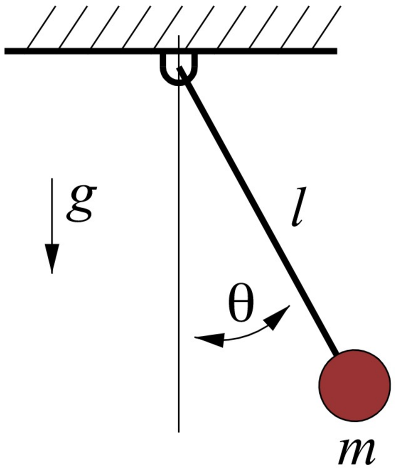
$$[s_1, s_2, s_3, \ldots]$$



System trajectory

- Let's look at an example of pendulum
- Now consider for overdamped pendulum

$$b\dot{\theta} \gg ml^2\ddot{\theta}$$



- Therefore

$$ml^2\ddot{\theta} + b\dot{\theta} \approx b\dot{\theta} = u_0 - mgl\sin\theta$$

- Now for this first-order system we can define the state to be $x = \theta$, and thus

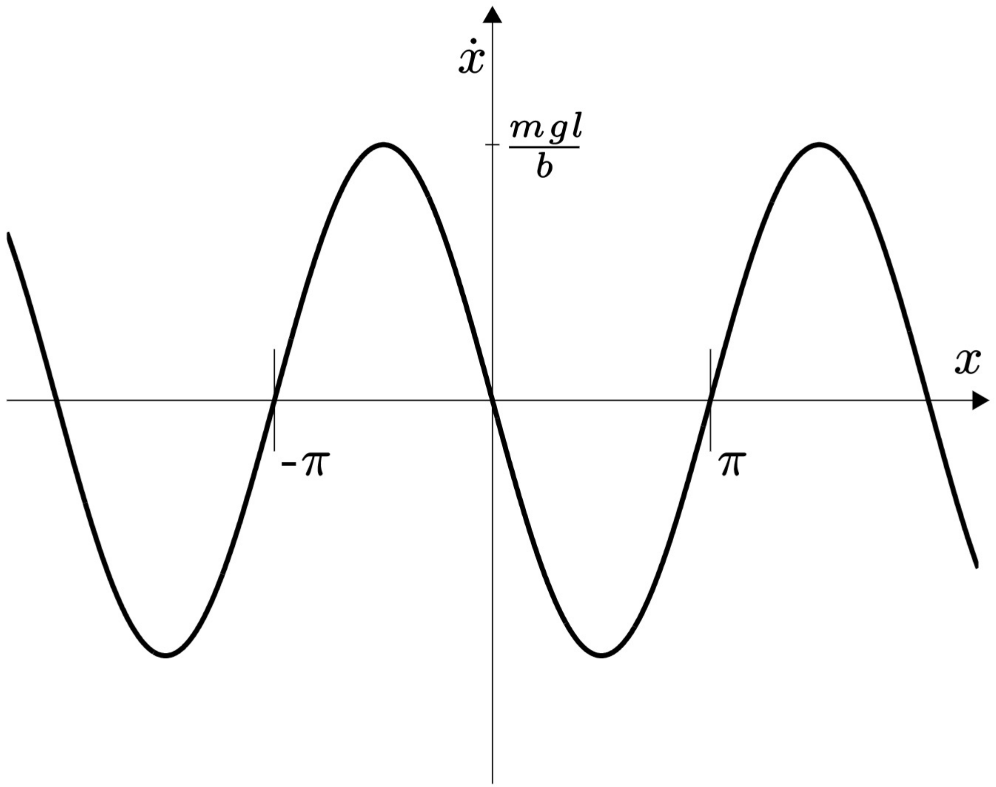
$$b\dot{x} = u_0 - mgl\sin x$$

System trajectory

- Let $u_0 = 0$, then

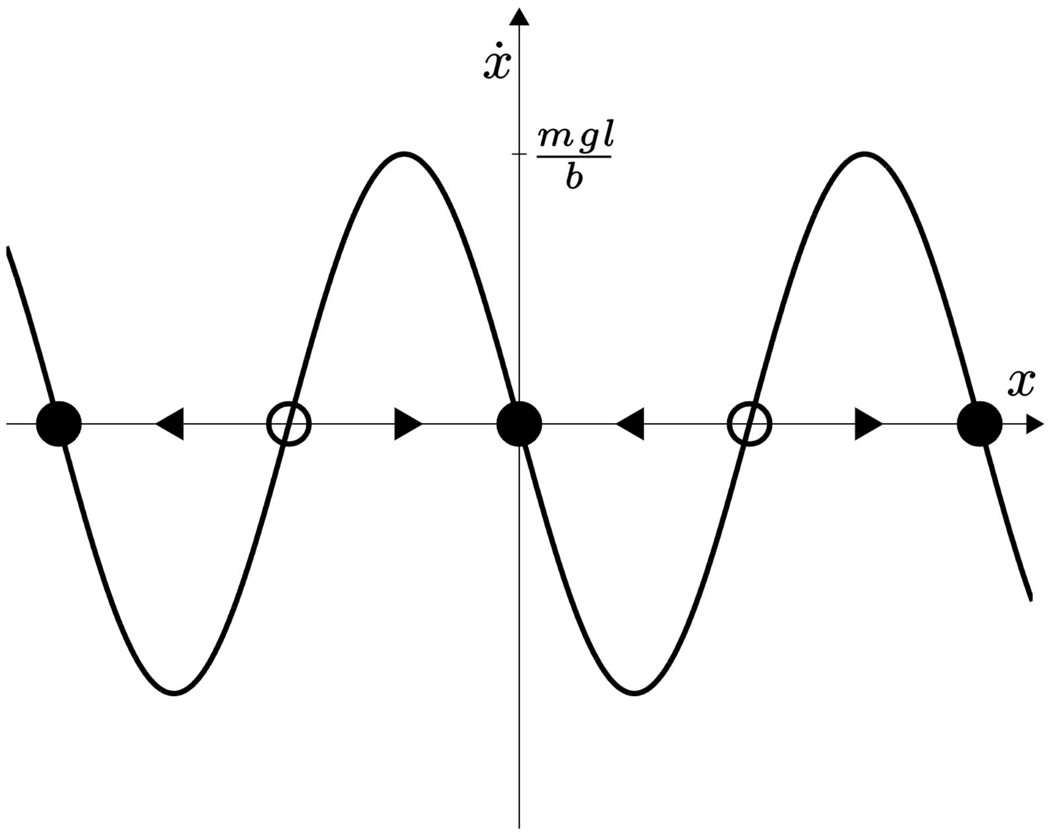
$$b\dot{x} = -mgl \sin x$$

- The system trajectory can be plotted as



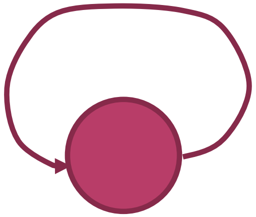
System trajectory

- The system has fixed points at $x^* = \{\ldots, -\pi, 0, \pi, 2\pi, \ldots\}$
- Some of them are stable, some are unstable
- There is a region of attraction around the stable points

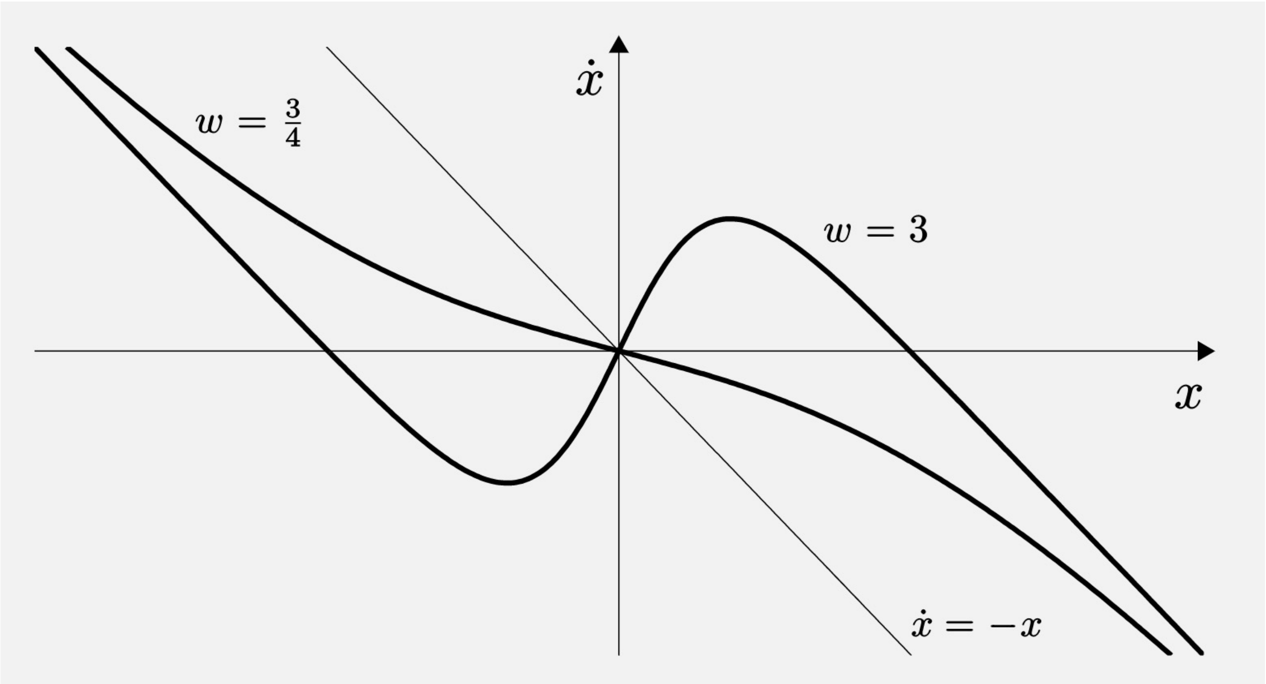


RNN

- Simple RNN with single layer and tanh activation



$$\dot{x} + x = \tanh(wx)$$



Phase portrait

- Now consider the undamped pendulum with zero torque

$$u = ml^2\ddot{\theta} + b\dot{\theta} + mgl\sin\theta$$

  where b=0 and u=0

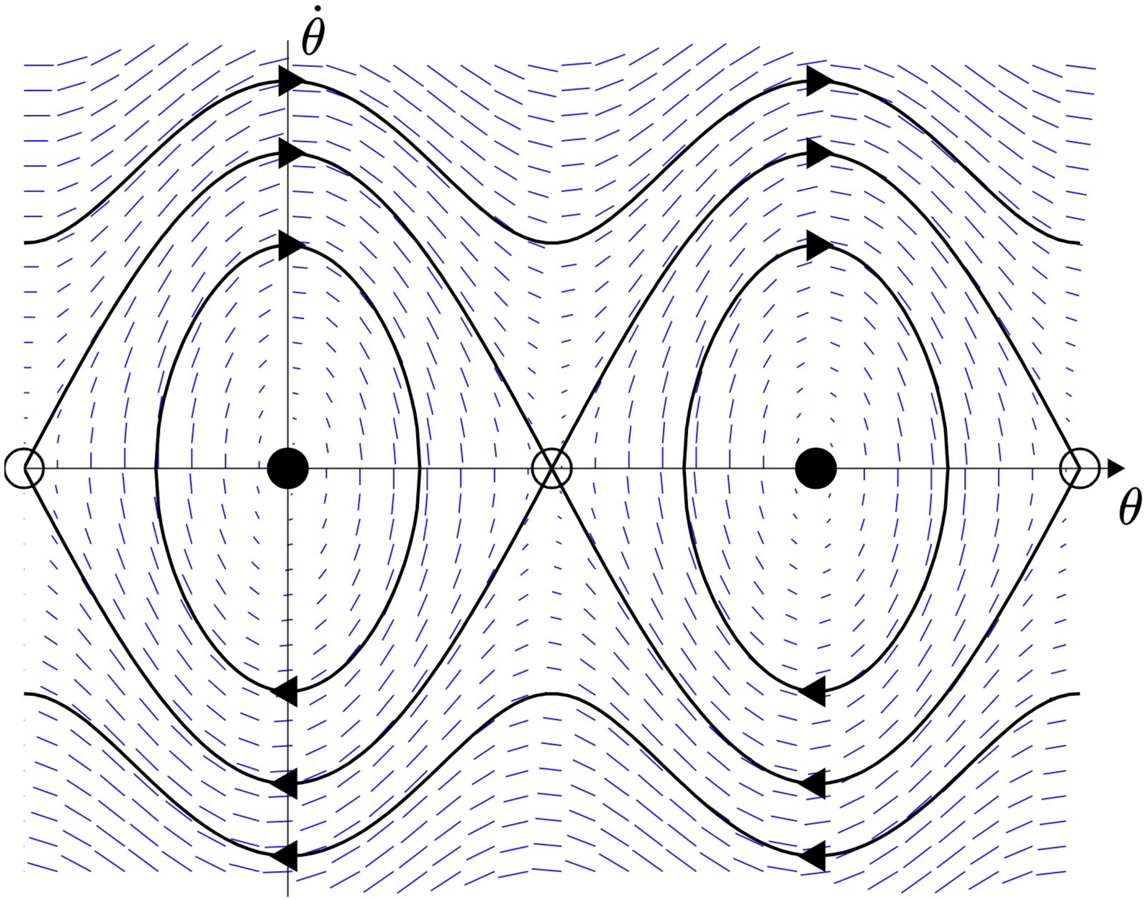
- This is a second-order system, which can be written as a first-order system with coupled state $x = [\theta, \dot{\theta}]^T$:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = f(x_1, x_2, u)$$

- In this case, the graphical depiction of the system is not a curve, but a vector field where the vectors $[\dot{x}_1, \dot{x}_2]^T$ are plotted over the domain of the state (x_1, x_2)

Phase portrait

- This vector field is called the phase portrait of the system

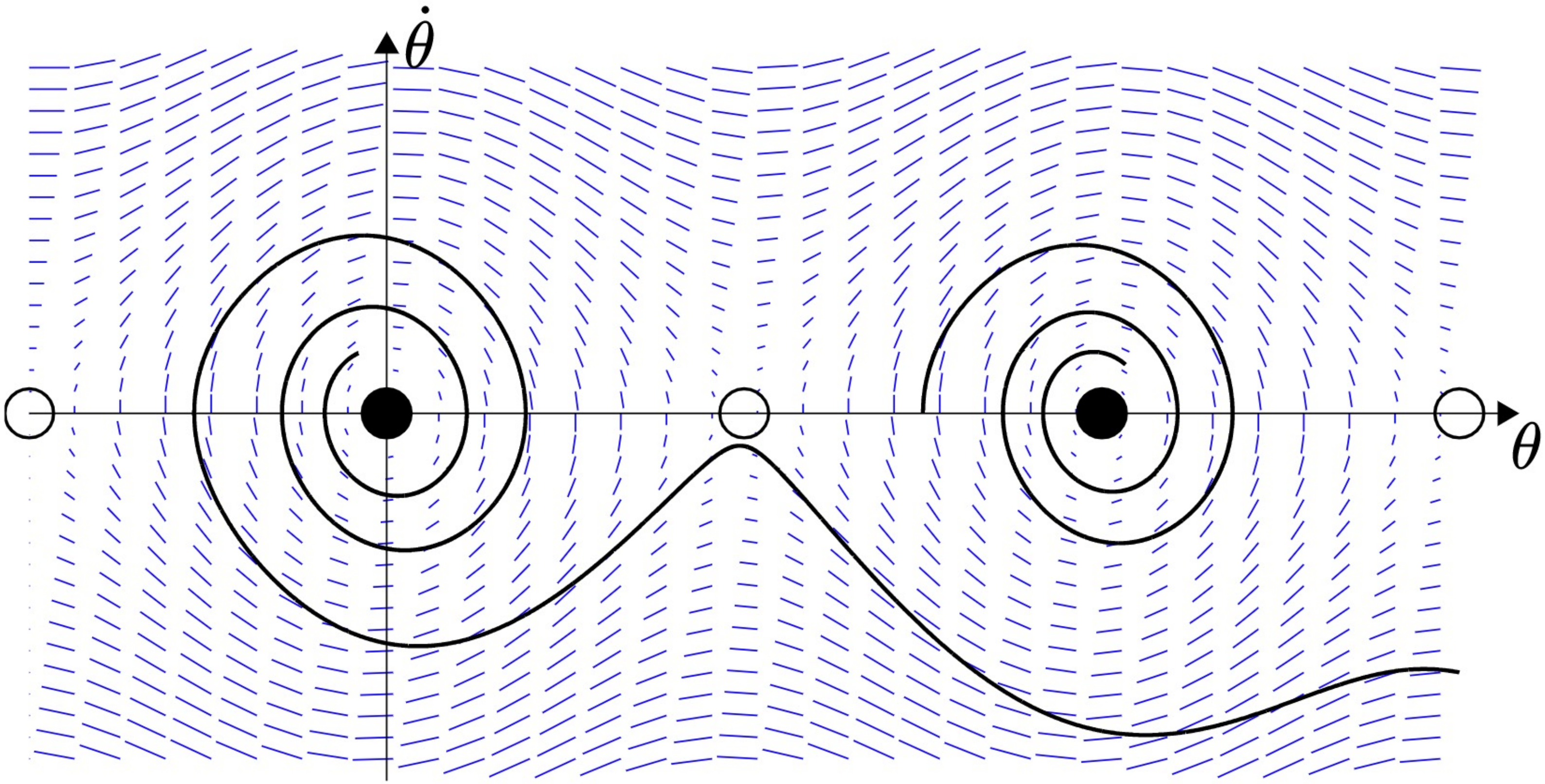


Which points are stable?

$$ml^2\ddot{\theta} = -mgl\sin\theta$$

Phase portrait

- Now let's add the damping

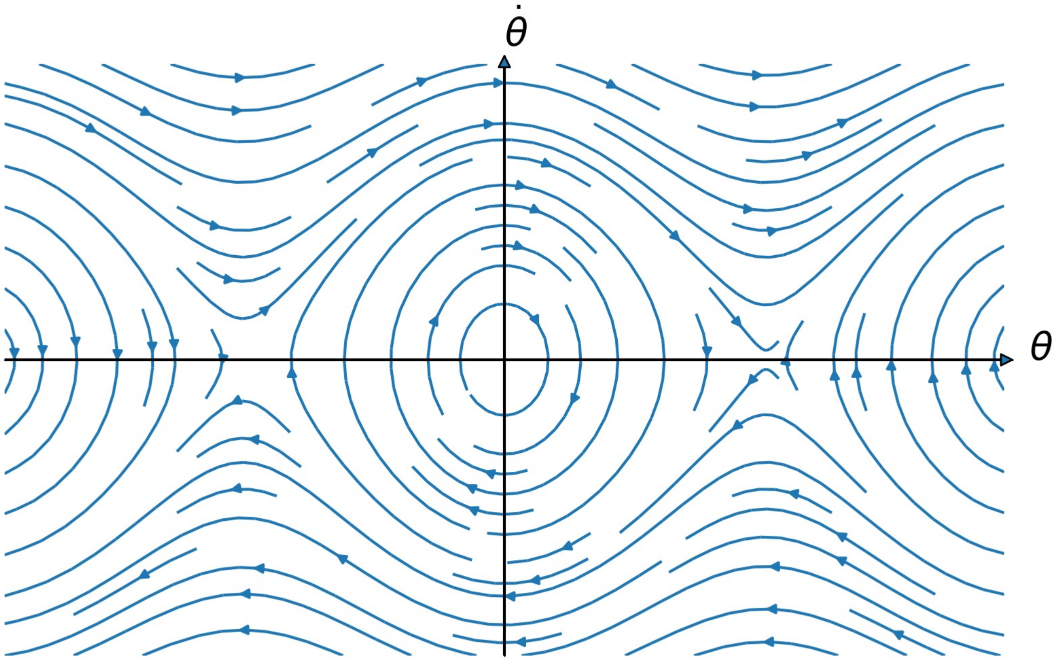


$$ml^2\ddot{\theta} = -mgl\sin\theta - b\dot{\theta}$$

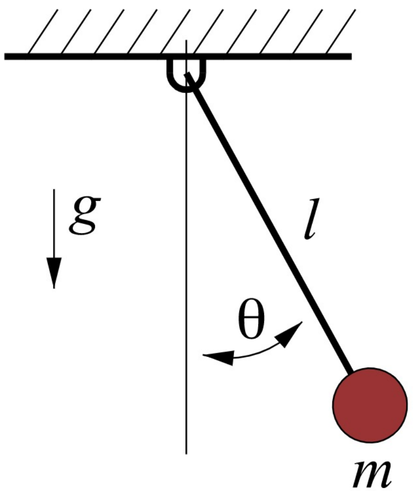
Which points are stable?

Phase Portrait of Uncontrolled Systems

- Dynamics (phase portrait) of the undamped uncontrolled pendulum



$$ml^2\ddot{\theta} + mgl\sin\theta = 0$$



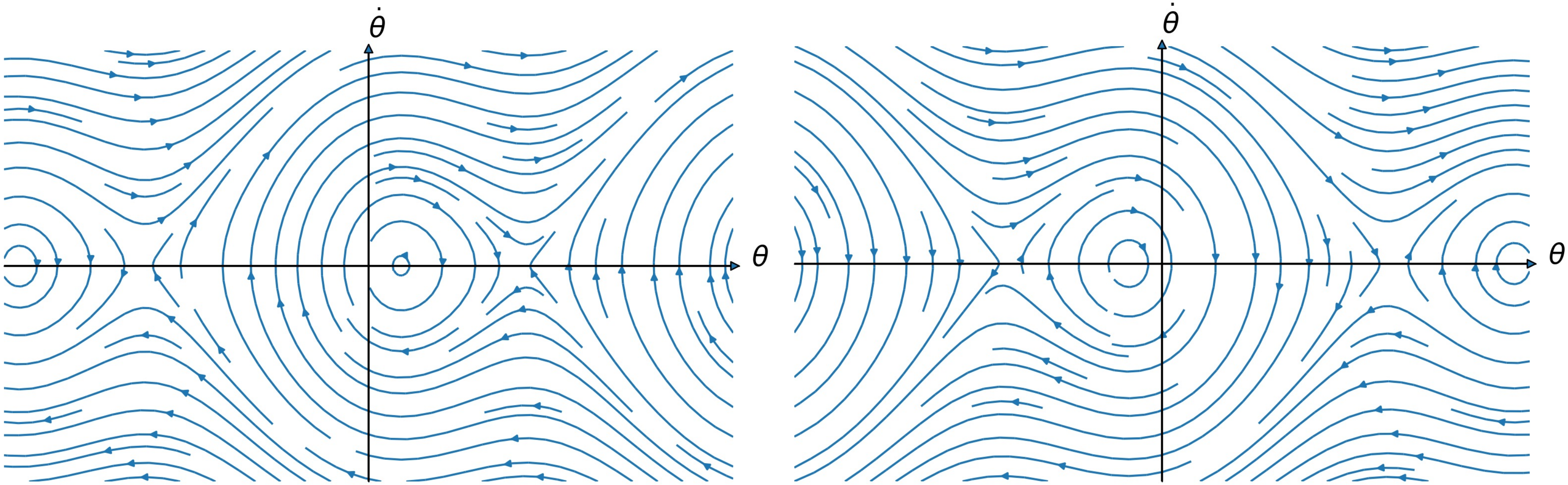
$$ml^2\ddot{\theta} + mgl\sin\theta = u$$

What's the physical meaning of the orbits?

- How about applying controls (torques)?

Phase Portrait of Controlled Systems

- The dynamics is changed when applying controls



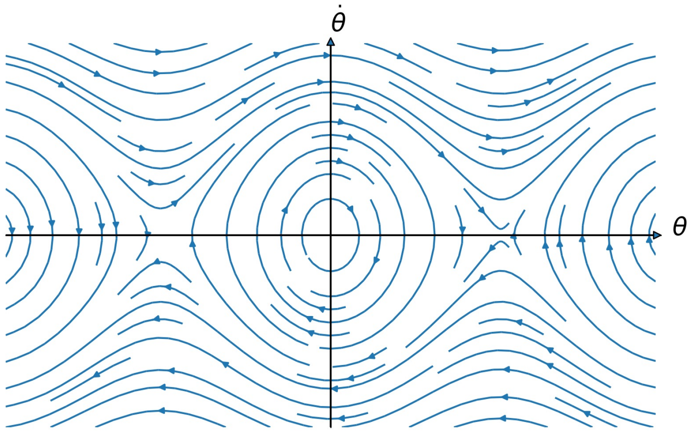
$u_0 = 5$ $\qquad$ $u_0 = -5$

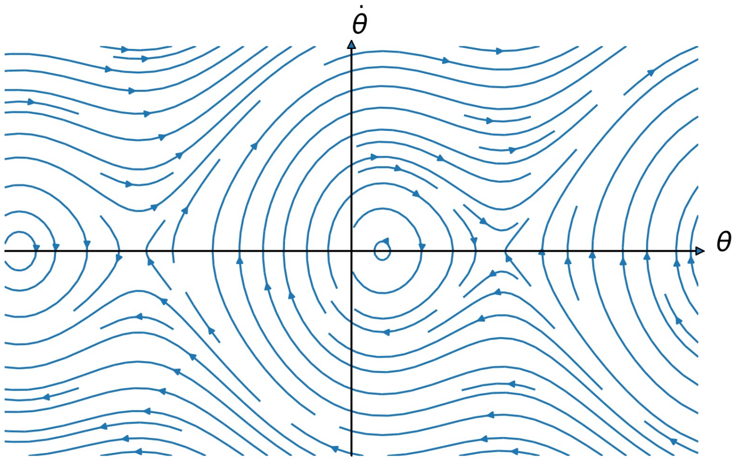
$$ml^2\ddot{\theta} + mgl\sin\theta = u_0$$

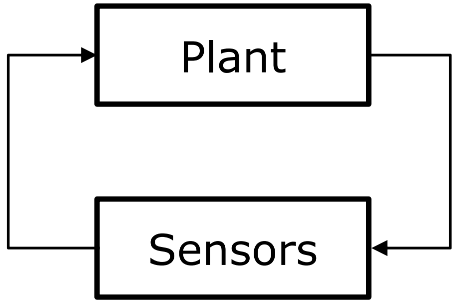
- Can we change the dynamics arbitrarily?
  - The x-component of vector is always $\dot{\theta}$
  - The trajectories are subject to physical constraints

Controller design

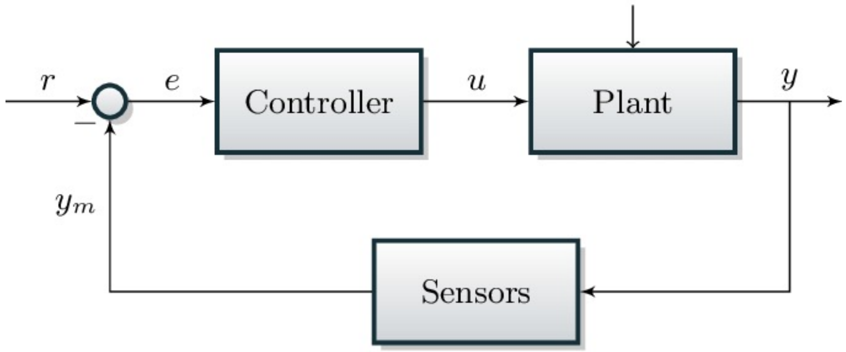
- Controller design is basically to change the dynamics towards the desired behavior







Uncontrolled system



Controlled system

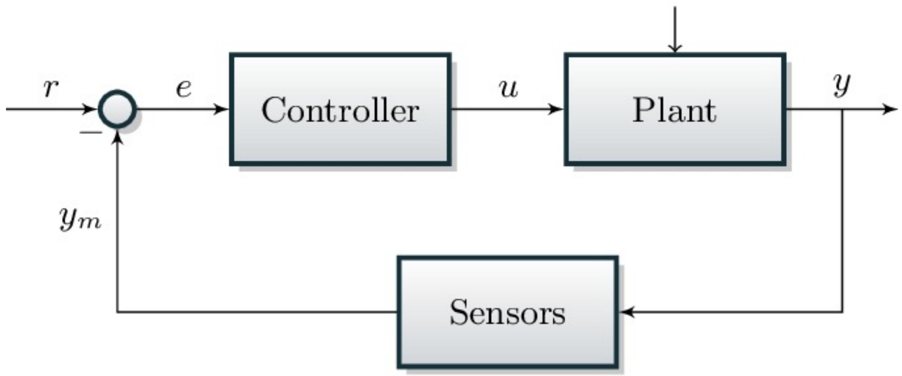
Error dynamics

- For motion control, we have

reference: $\theta_d(t)$

actual: $\theta(t)$

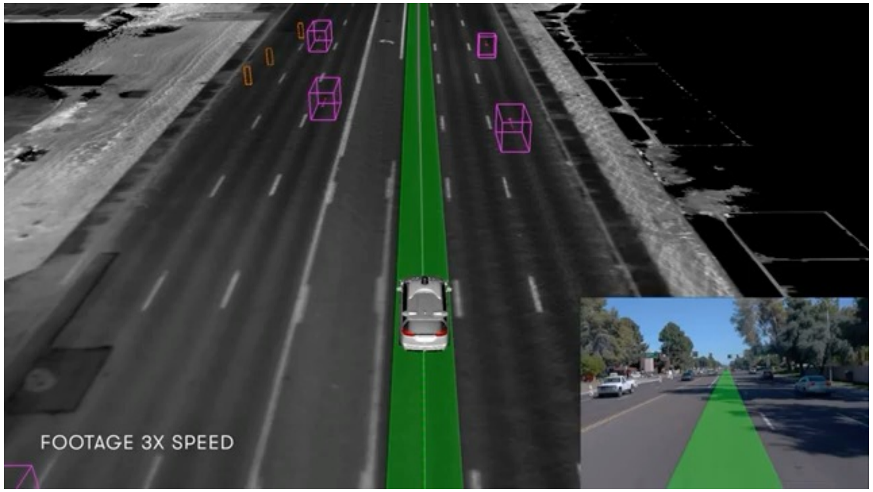
error: $\theta_e(t) = \theta_d(t) - \theta(t)$



- Basic desired behavior: we want the error becomes **zero**
- **Error dynamics**: how the tracking error $\theta_e(t)$ evolves over time

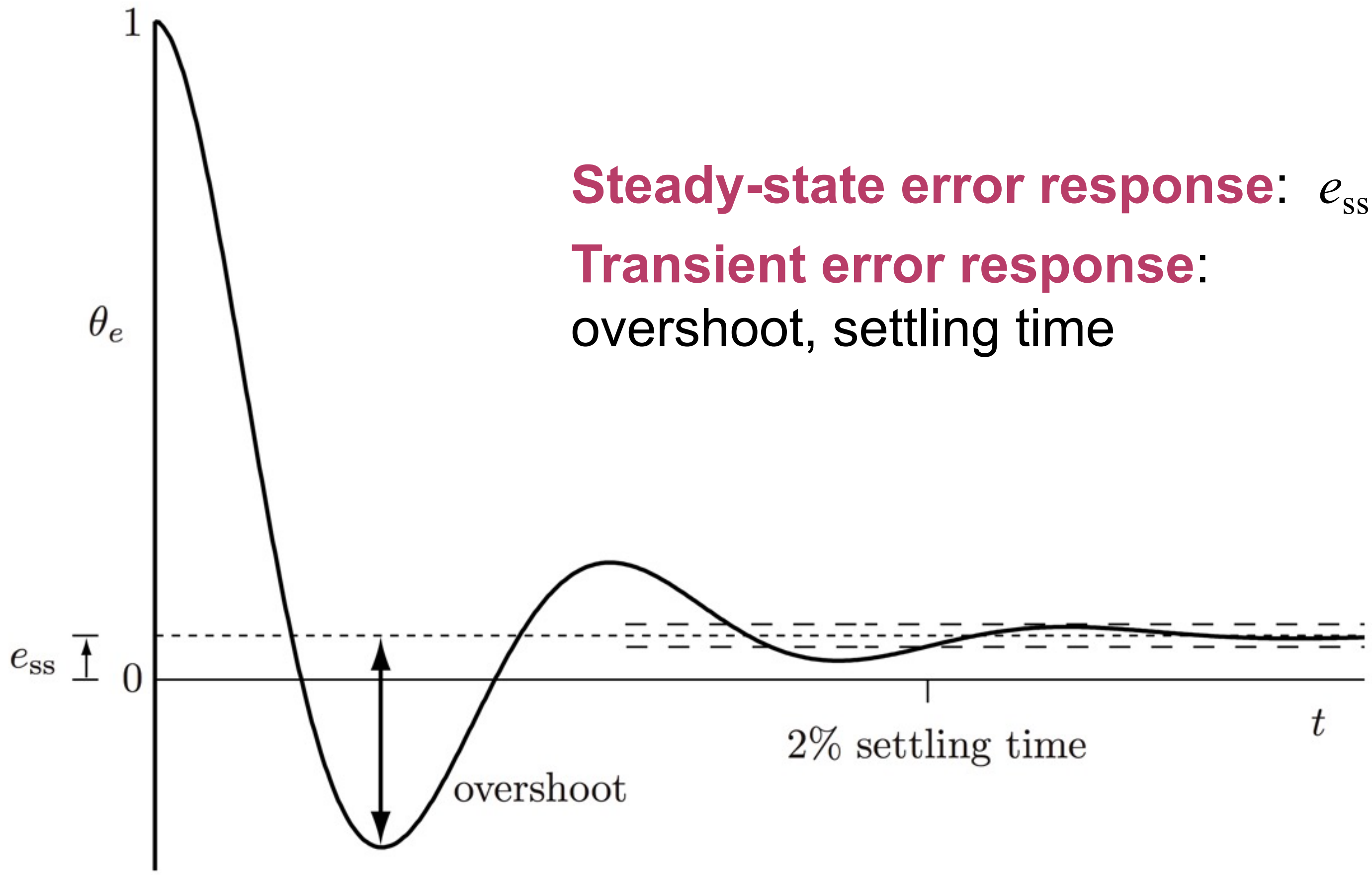
Error dynamics

- A lot of control problems aim to get zero tracking error



Error dynamics

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

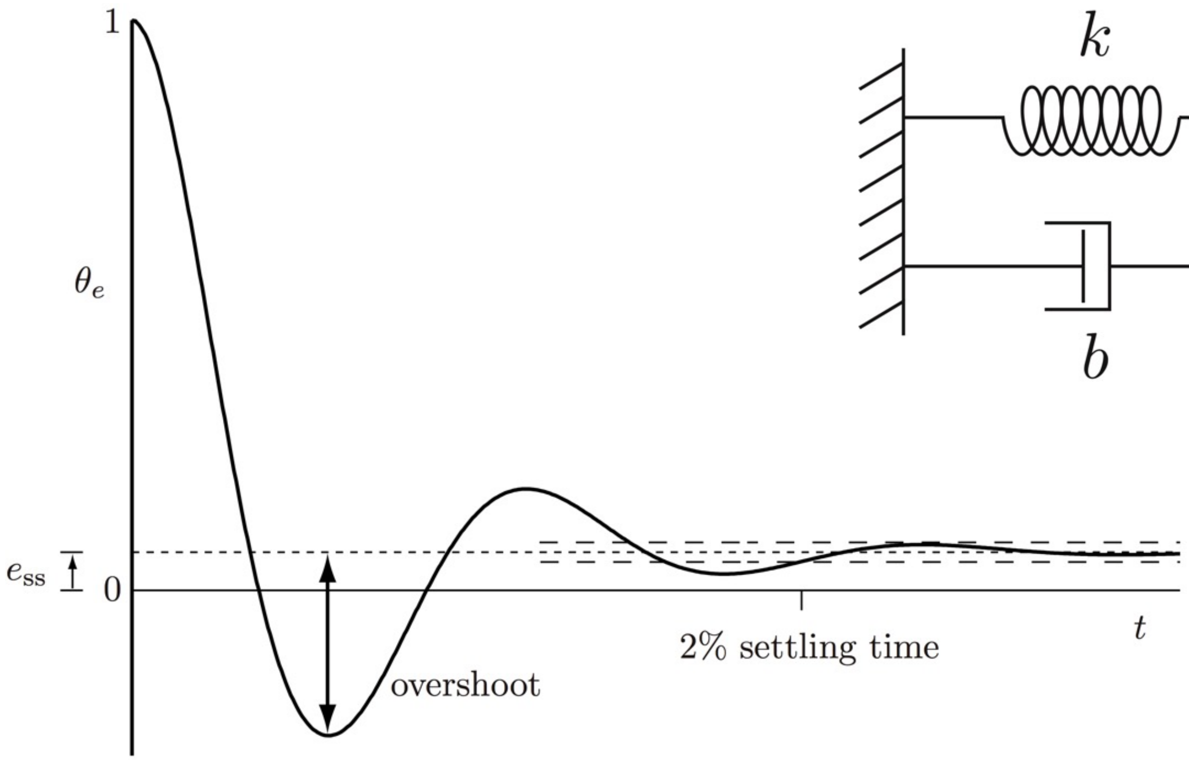


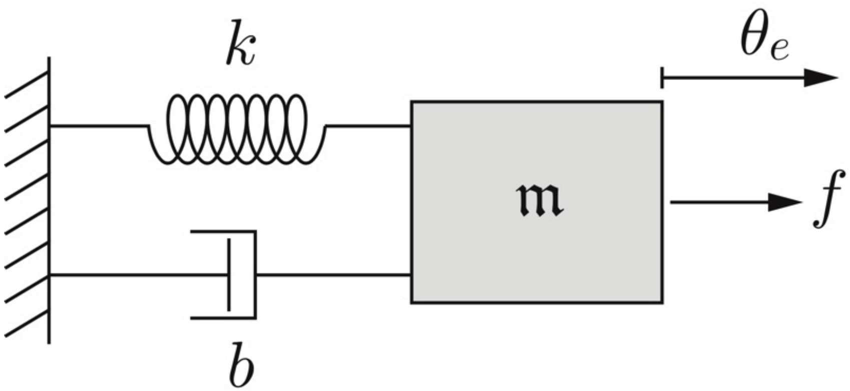
Steady-state error response: e_{ss}

Transient error response:
overshoot, settling time

Mass-spring-damper

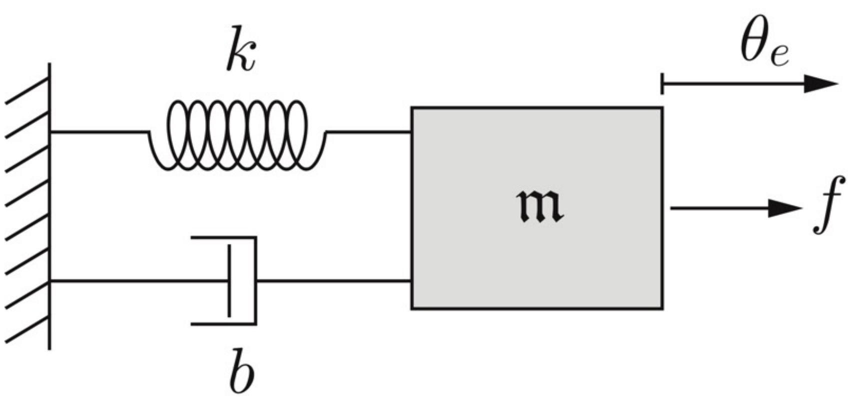
- This figure resembles the position response of a **mass-spring-damper**





Mass-spring-damper

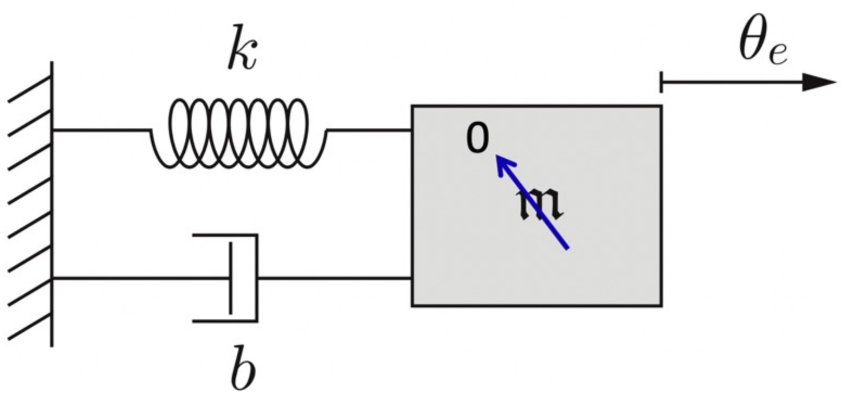
- Designing a controller is like to choose $\boldsymbol{k}$ and $\boldsymbol{b}$
  - **Spring constant** $\boldsymbol{k}$: larger value leads to faster convergence
  - **Damping constant** $\boldsymbol{b}$: larger value to reduce oscillation
- The motion can be described by a **second order differential equation**



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

First-order error dynamics

- Reduce the mass-spring-damper to a massless system results in **first-order error dynamics**



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

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

standard first-order form

time constant

$$\mathfrak{t} = b/k$$

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

First-order error dynamics

- First-order error dynamics:

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

where $\mathfrak{t}$ is called the **time constant**. The solution is:

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

- At time $\mathfrak{t}$, the error will decay to approximately 37% of its initial value

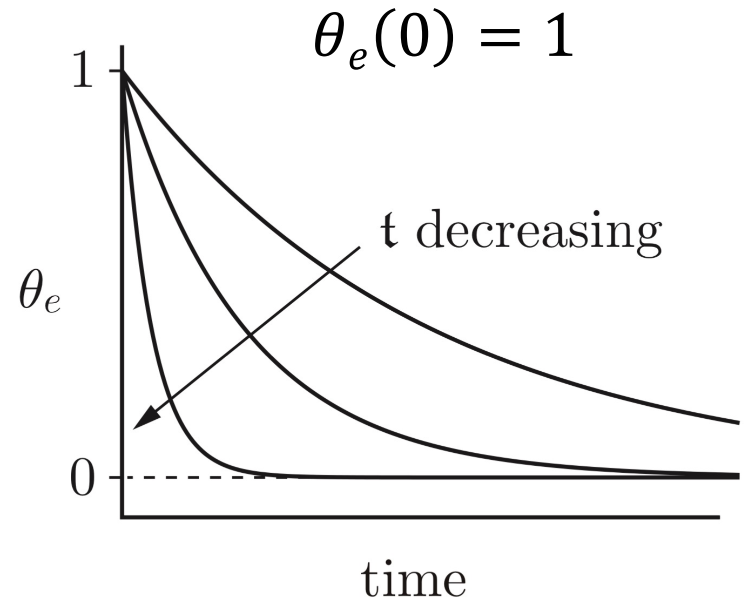
First-order error dynamics

- The **steady state error** is zero
- There is **no overshoot**
- The 2% **settling time** is defined by:

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

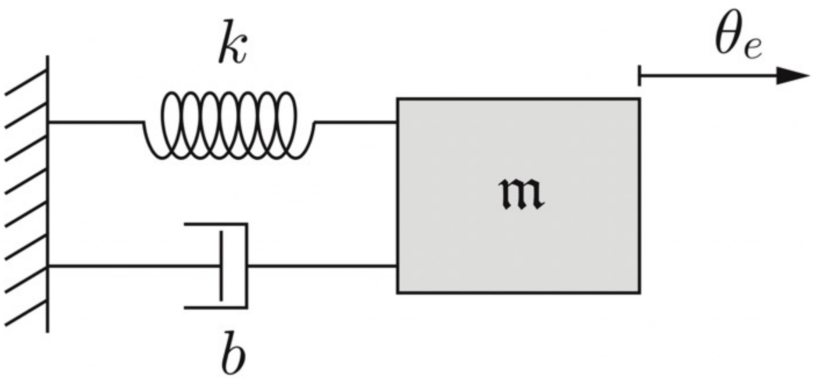
  - The solution is:

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



Second-order error dynamics

- Mass-spring-damper: **second-order error dynamics**



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

natural frequency   damping ratio

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

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

standard second-order form

Second-order error dynamics

- Now let's look at the **second-order error dynamics** in the standard form:

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

- The corresponding **characteristic polynomial** is:

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

- Its solutions are

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

$\zeta > 1:$ Overdamped

$\zeta = 1:$ Critically damped

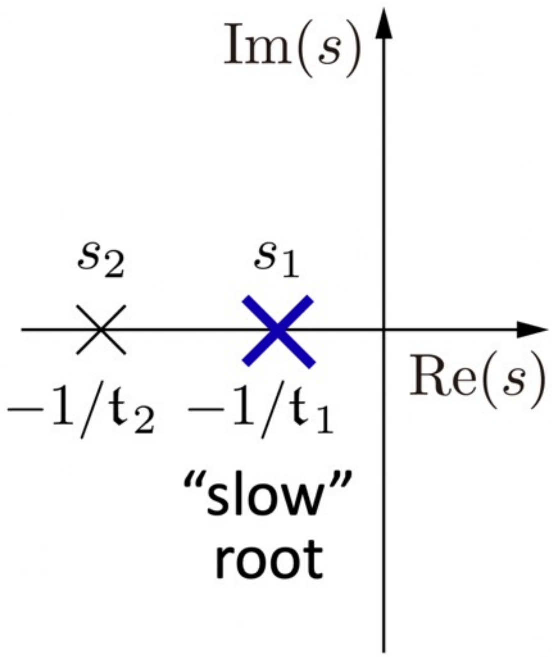
$\zeta < 1:$ Underdamped

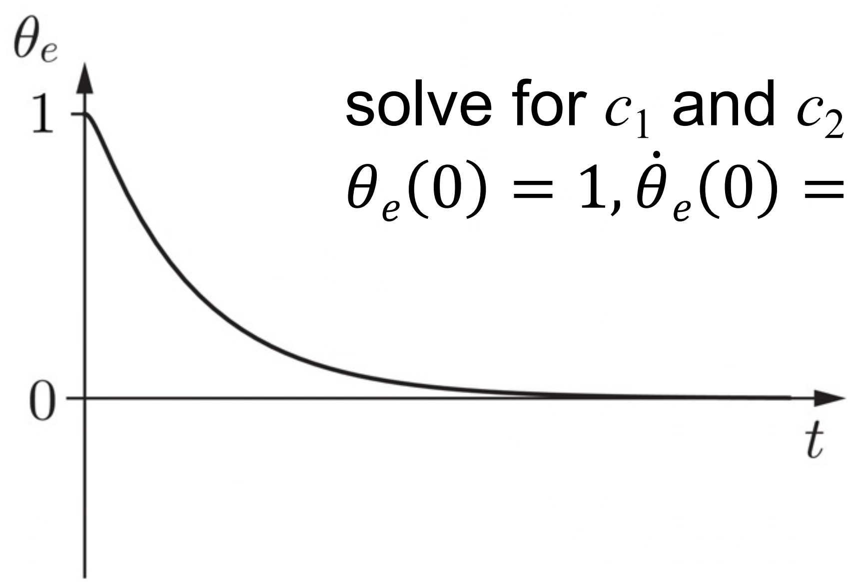
Second-order error dynamics

$\zeta > 1$: Overdamped

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

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



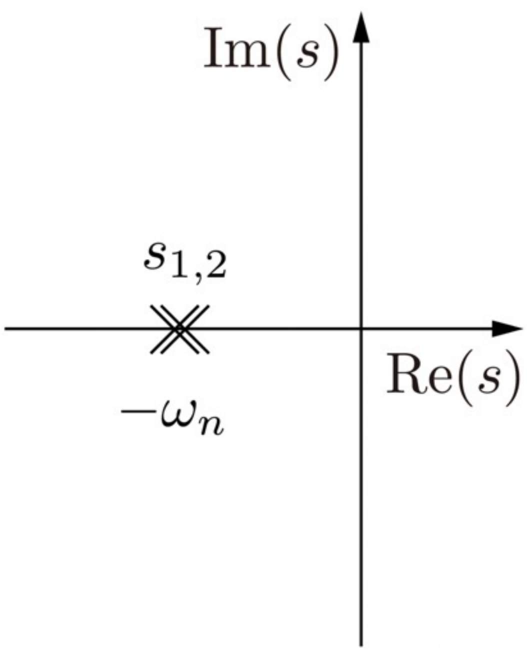


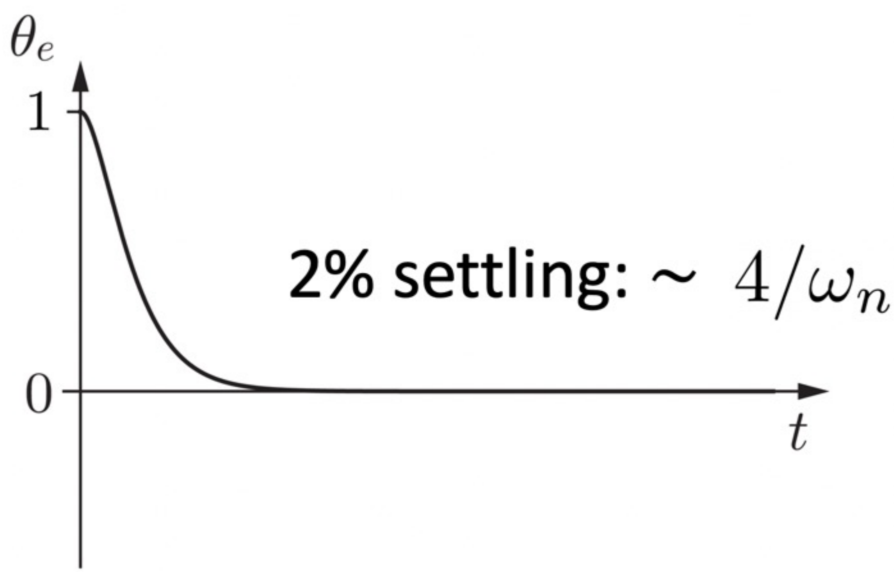
solve for c_1 and c_2 using $\theta_e(0) = 1, \dot{\theta}_e(0) = 0$

Second-order error dynamics

$\zeta = 1$: Critically damped

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





Second-order error dynamics

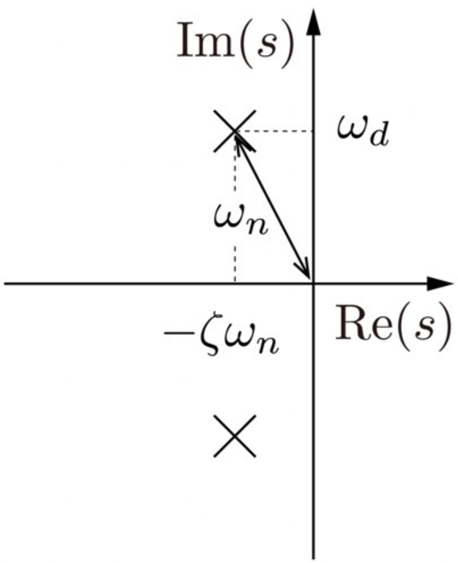
$\zeta < 1$: Underdamped

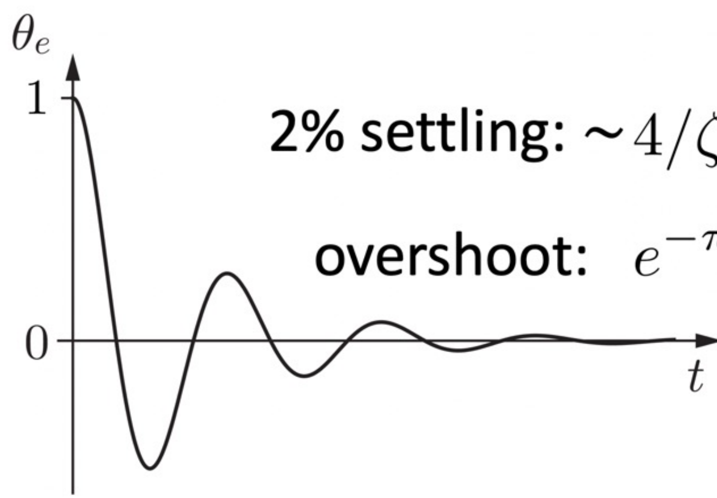
$$\theta_e(t) = (c_1 \cos \omega_d t + c_2 \sin \omega_d t)\, e^{-\zeta \omega_n t}$$

damped natural frequency

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

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



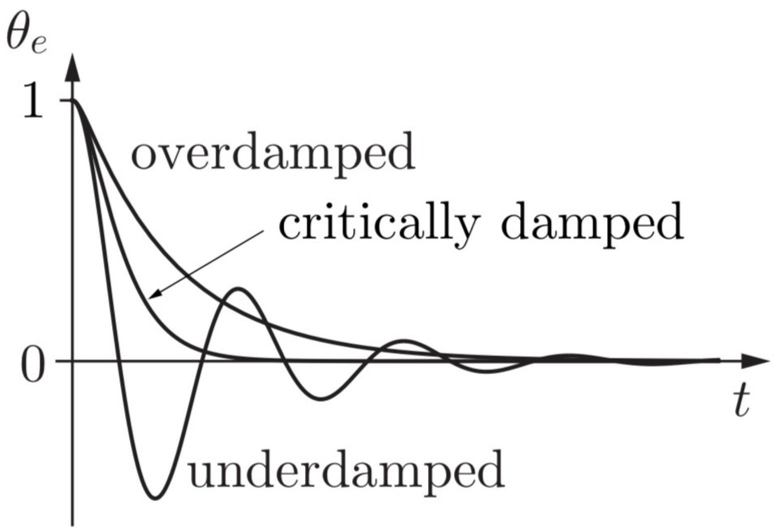


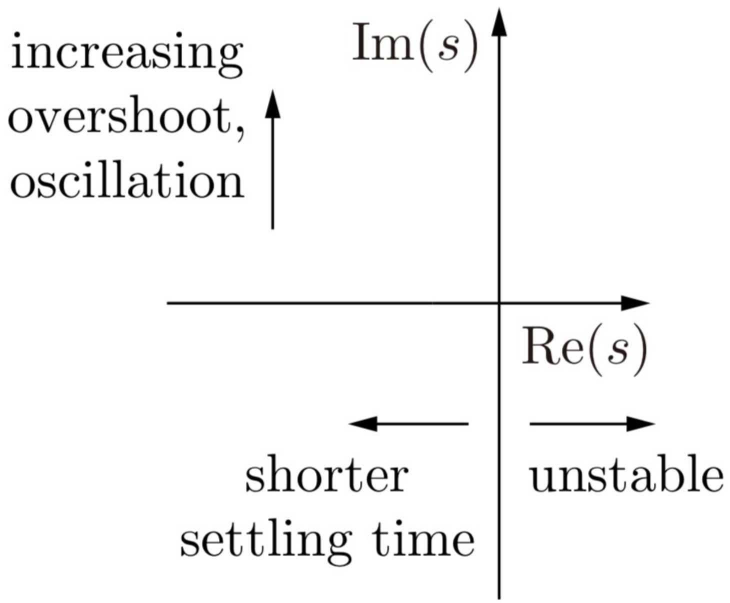
2% settling: $\sim 4/\zeta\omega_n$

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

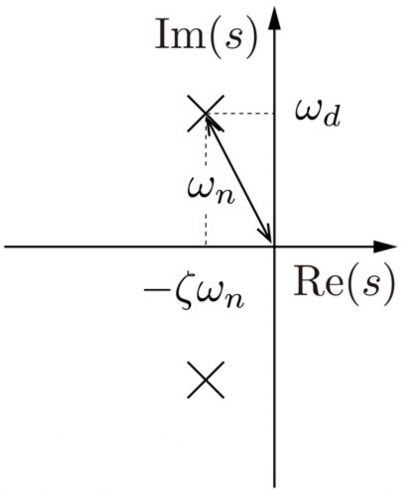
Second-order error dynamics

- As long as $\omega_n^2 > 0$ and $\zeta\omega_n > 0$, the system is stable (has zero steady state error)





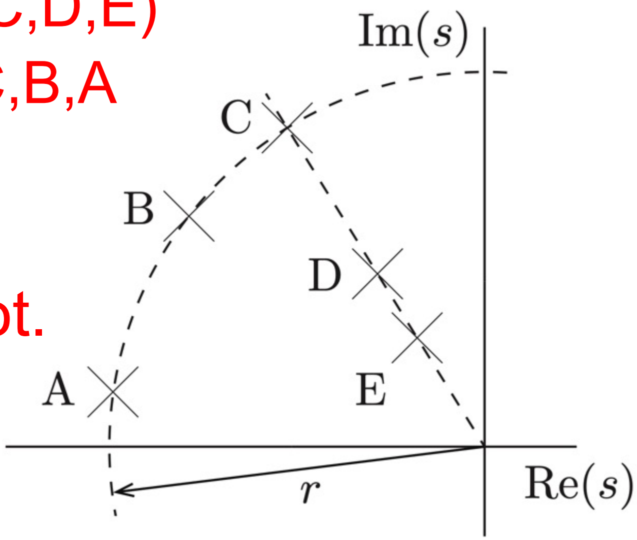
Exercise



Shown are one of the roots of five different second-order systems, A, B, C, D, and E. List them in the following orders:

(A,B,C),D,E

1. Natural frequency, highest to lowest.
2. Damped natural frequency, highest to lowest. C,B,D,E,A
3. Damping ratio, highest to lowest. A,B,(C,D,E)
4. Settling time, longest to shortest. E,D,C,B,A



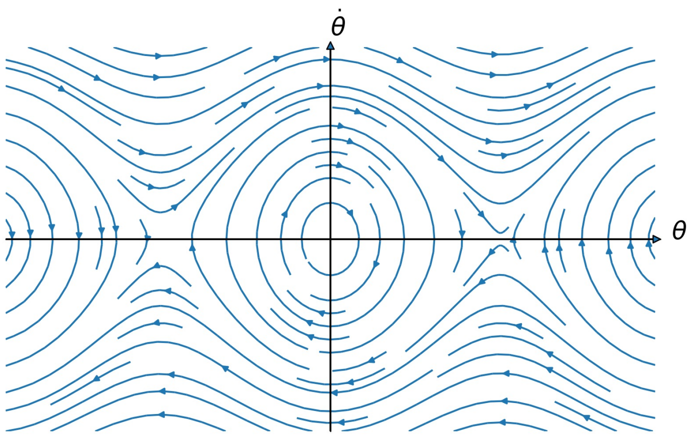
Which has the "best" response?

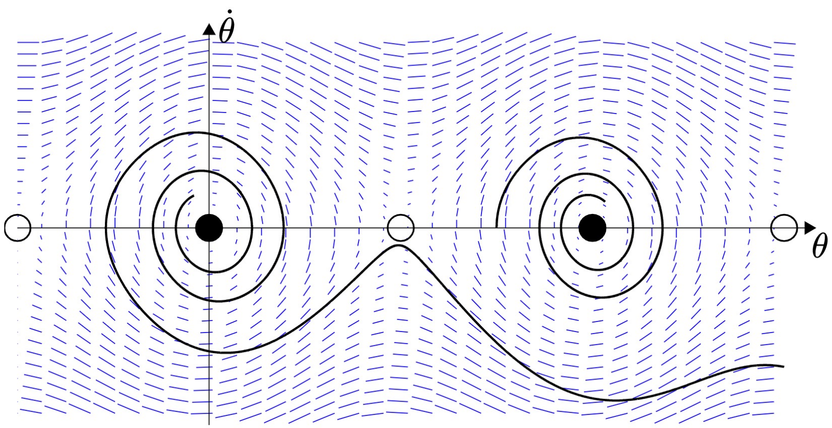
A. Largest damping ratio, lowest overshoot. Fastest settling time.

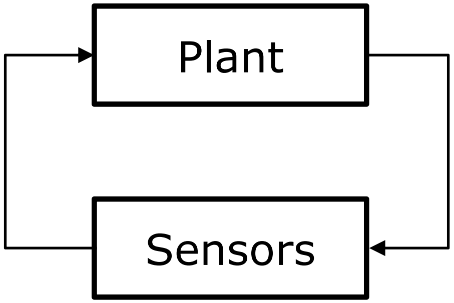
Contents

Feedback controller design

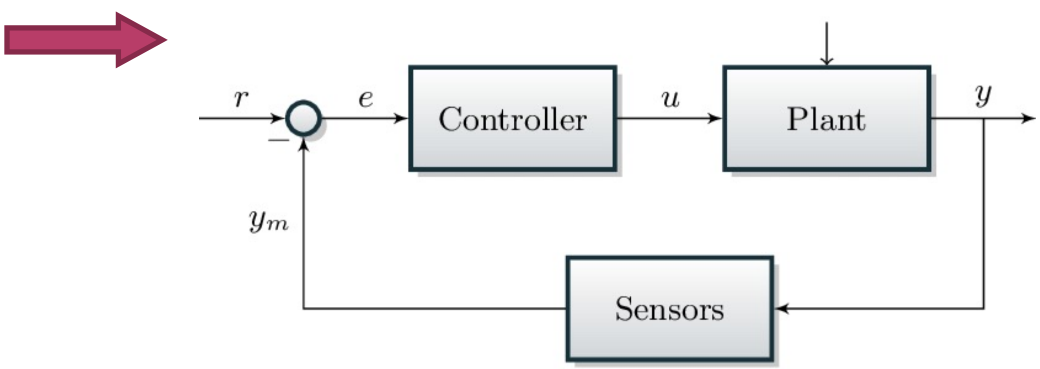
- How can we design a feedback controller to achieve desired system behavior?







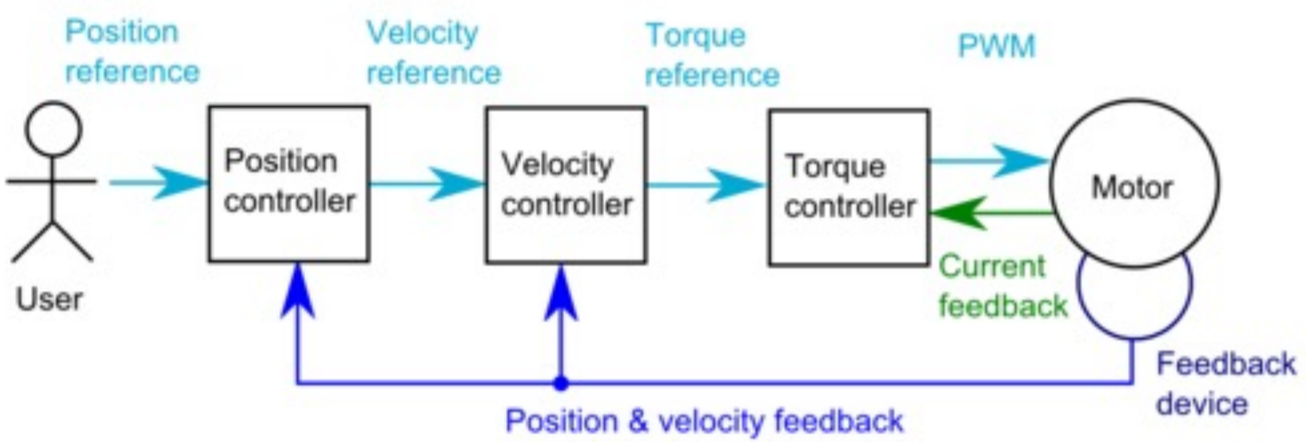
Uncontrolled system



Controlled system

Action space

- **Different choices of action space**
  - Torque
  - Velocity
  - Position
  - Mixed



Position control with velocity inputs

- Let's first look at the case when we can use **velocity** $\dot{\theta}(t)$ **as inputs** to track $\theta_d(t)$.
- Feedback control
  - **P control**: Proportional to the error

$$\dot{\theta}(t) = K_p(\theta_d(t) - \theta(t)) = K_p\theta_e(t)$$

  - **Setpoint control**: $\theta_d(t)$ is constant

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

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

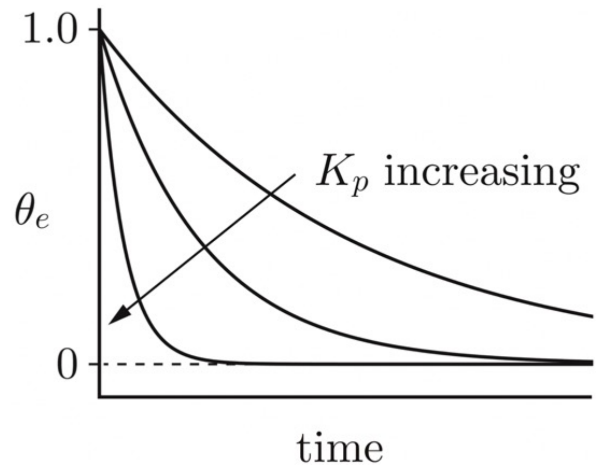
Error dynamics of P control

- The error dynamics is a **first-order linear system**:

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

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

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



- The time constant is $\tau = 1/K_p$
- The 2% settling time is four times the time constant $4/K_p$
- The above are under the assumption that $\theta_d(t)$ is constant. What if $\theta_d(t)$ is changing with $\dot{\theta}_d(t) = c$?

Error dynamics of P control

- Now error dynamics becomes:

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

- The solution is:

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

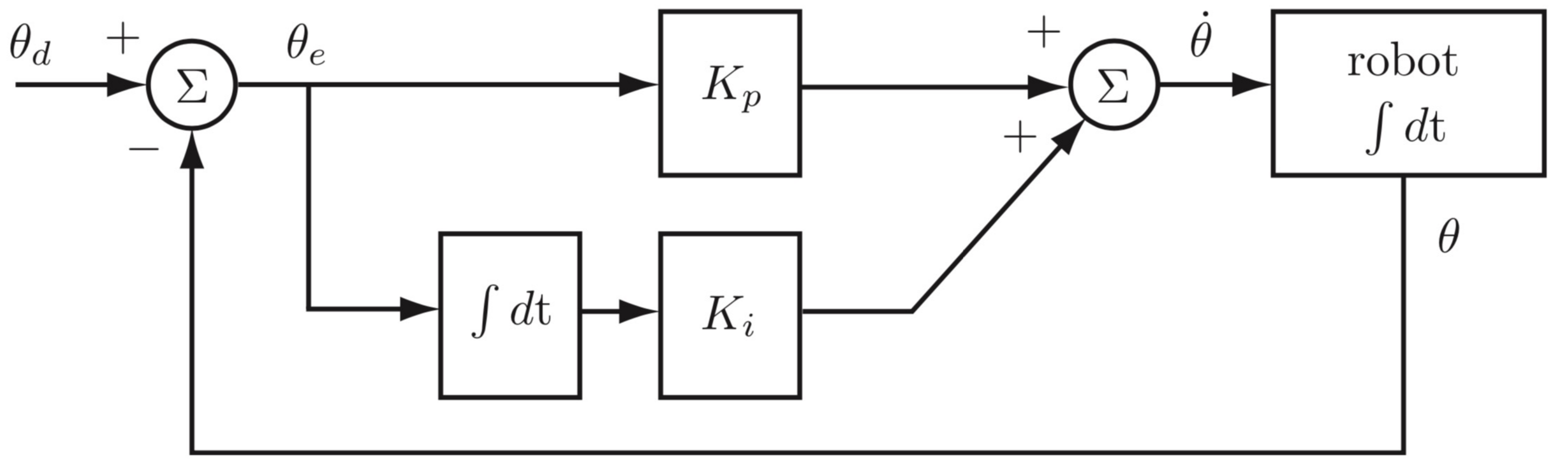
- This converges to a **nonzero** steady-state error c/K_p
- How to alleviate the nonzero error?
  - Choose large control gain K_p
  - Joint has velocity limits
  - May cause instability

PI control

- Another way is to use the **proportional-integral (PI)** controller, which adds a **time-integral** term of error:

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

- The controller diagram is:



Error dynamics of PI control

- With PI controller, the error dynamics when $\dot{\theta}_d(t) = c$ is:

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

- Taking derivatives we have

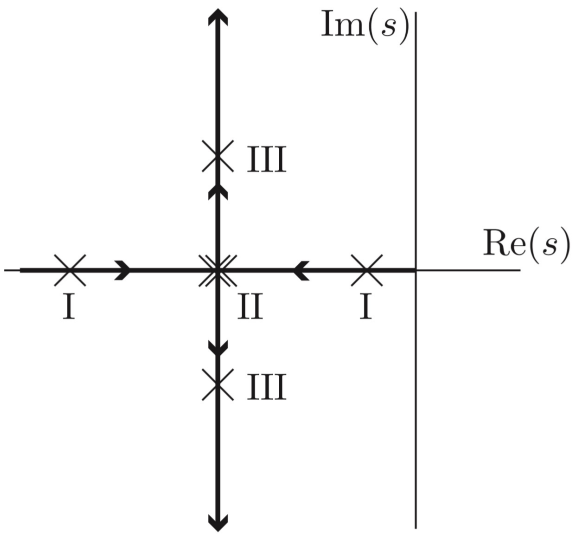
$$\ddot{\theta}_e(t) + K_p\dot{\theta}_e(t) + K_i\theta_e(t) = 0$$

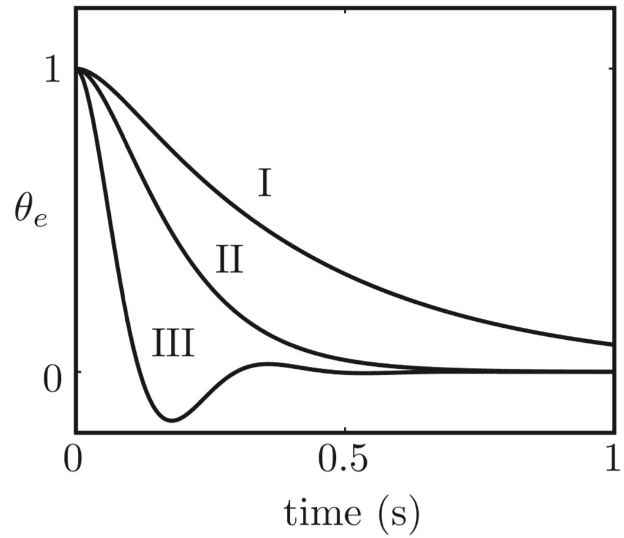
- Write this in the standard form we have the natural frequency $\omega_n = \sqrt{K_i}$ and damping ratio $\zeta = K_p/(2\sqrt{K_i})$
- When $K_i > 0$ and $K_p > 0$, the error dynamics is stable
- The roots of the characteristic equation are:

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

Error dynamics of PI control

- If we keep $K_p = 20$ and increase K_i from zero, we can draw a plot of roots, called **root locus**.

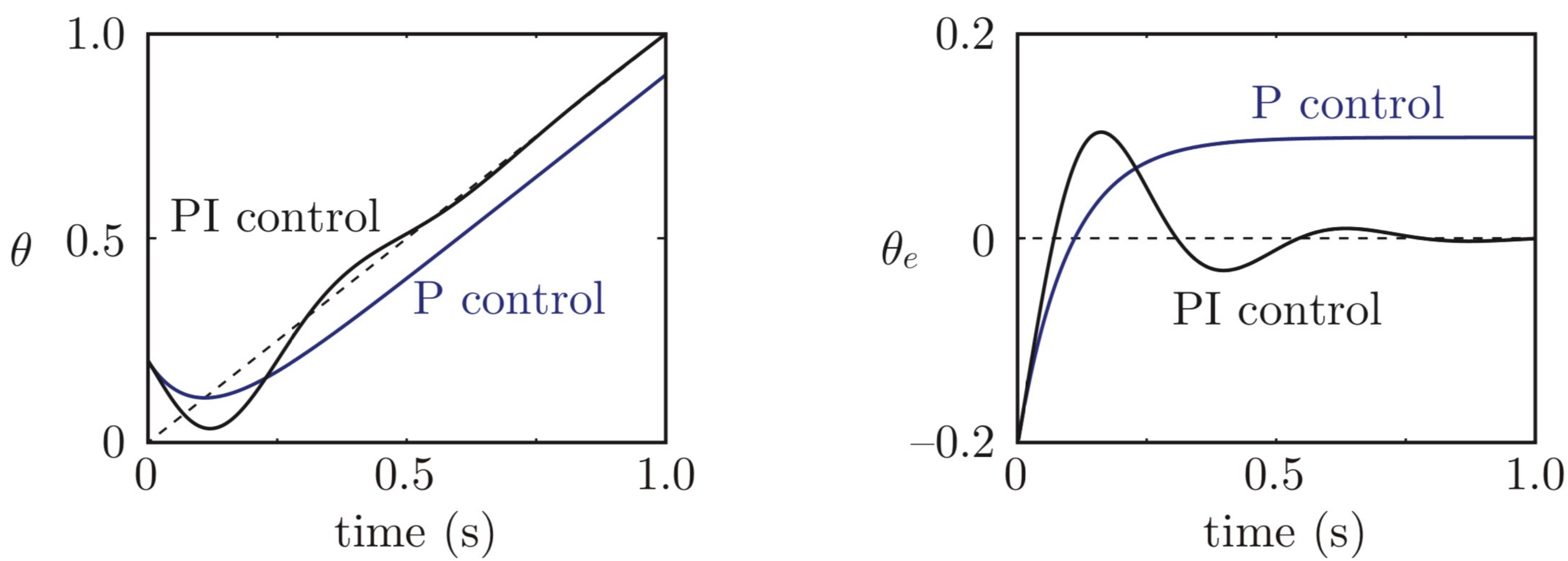




- We can choose the **critical damping point** $K_i = K_p^2/4$
  - When overdamped, the convergence speed is dominated by the slower one.
  - When underdamped, there are oscillations.

PI control vs P control

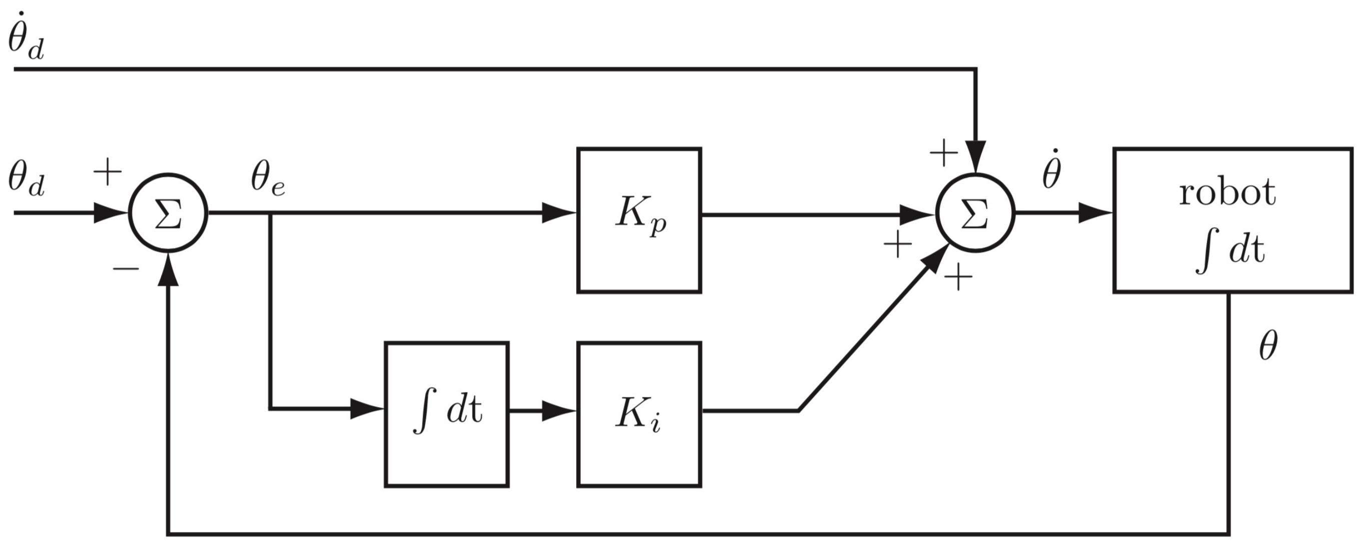
- When $\dot{\theta}_d(t) = c$
  - P control has a constant steady state error
  - PI control has **zero** steady state error



Mixed position and velocity control

- Drawback of pure feedback position control:
  - Requires time to accumulate error
- Combine **feedforward velocity** and **feedback position** control

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



Exercise

- You are designing a P controller to track a joint reference trajectory that is moving at a constant rate of 3 radians/s. What is the smallest gain Kp that ensures a steady-state position error of no more than 0.1 radians?
  - Steady state error $\frac{c}{K_p} \leq 0.1$ $\quad K_p \geq c/0.1 = 3/0.1 = 30$
- To eliminate steady-state error, you decide to use a PI controller. What gains Kp and Ki should you choose to achieve critically damping and a settling time of 0.1 s?
  - Critically damping: $\zeta = K_p/(2\sqrt{K_i}) = 1$
  - Settling time: $\frac{4}{\omega_n} = 0.1$ $\quad \omega_n = 40$

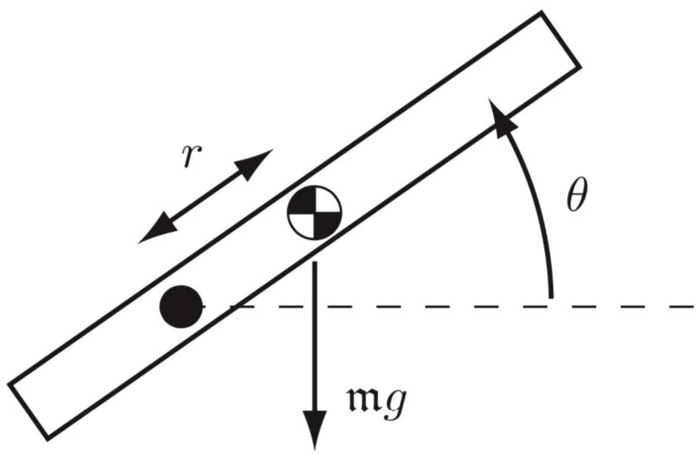
$$K_i = \omega_n^2 = 1600 \qquad K_p = 2\sqrt{K_i} = 80$$

Position control with torque inputs

- Now we consider using the **torque as inputs**, which is more flexible. This requires us to use dynamics.
- Let's start with the single joint control problem, its dynamics is:

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

where M is the inertia, b is the friction coefficient.



PD control

- Assume the link moves in horizontal (g=0), and use another type of controller, the **proportional-derivative (PD)** control:

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

- When $\dot{\theta}_d = \ddot{\theta}_d = 0$ we have:

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

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

- When $b + K_d$, K_p positive, the error dynamics is stable with zero steady state error

Drawbacks of PD control

- Now consider the link moves in a vertical plane (g>0). With PD controller the error dynamics is

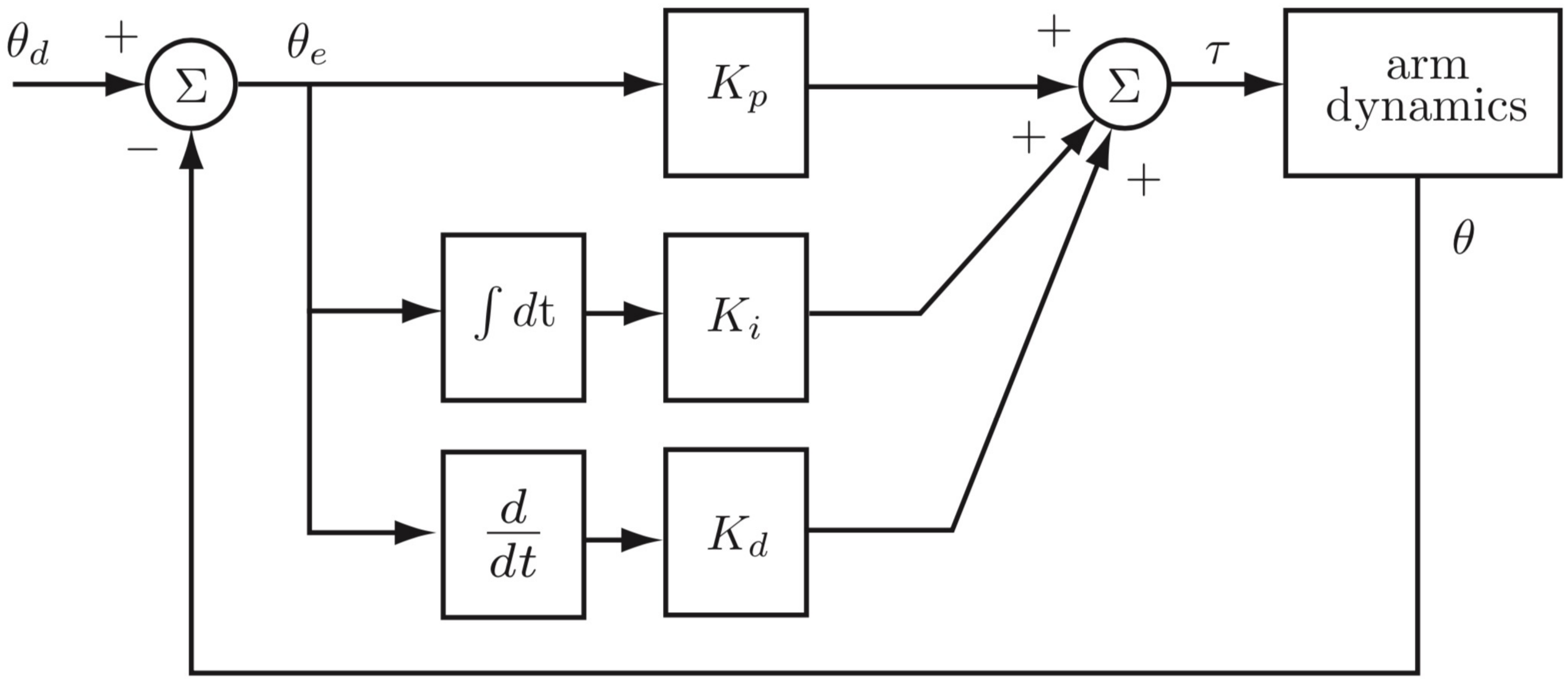
$$M\ddot{\theta}_e + (b + K_d)\dot{\theta}_e + K_p\theta_e = \mathfrak{m}gr\cos\theta$$

- The steady state satisfies $K_p\theta_e = \mathfrak{m}gr\cos\theta$
- When $\theta \neq \pm\pi/2$
  - There will be nonzero steady state error

PID control

- Let's add the integral term and obtain the **proportional-integral-derivative (PID) controller**:

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



Error dynamics of PID control

- The error dynamics of PID control is:

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

where τ_{dist} is the torque subject to the gravity term.

- Taking derivatives we get:

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

- Assume τ_{dist} changes slowly, so:

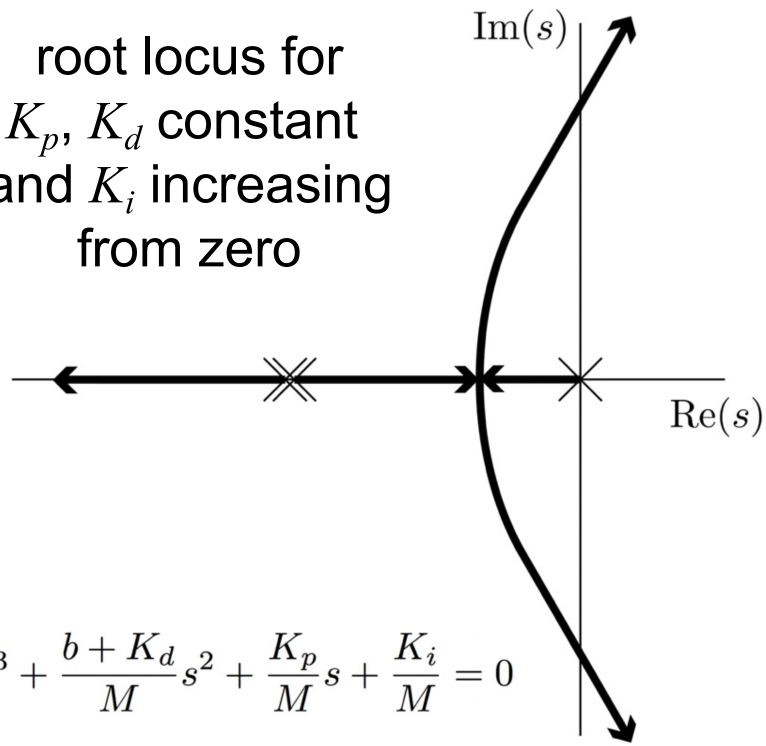
$$s^3 + \frac{b + K_d}{M}s^2 + \frac{K_p}{M}s + \frac{K_i}{M} = 0.$$

Stability of PID control

- To make the dynamics stable, the parameters should result in negative real components of all the roots:

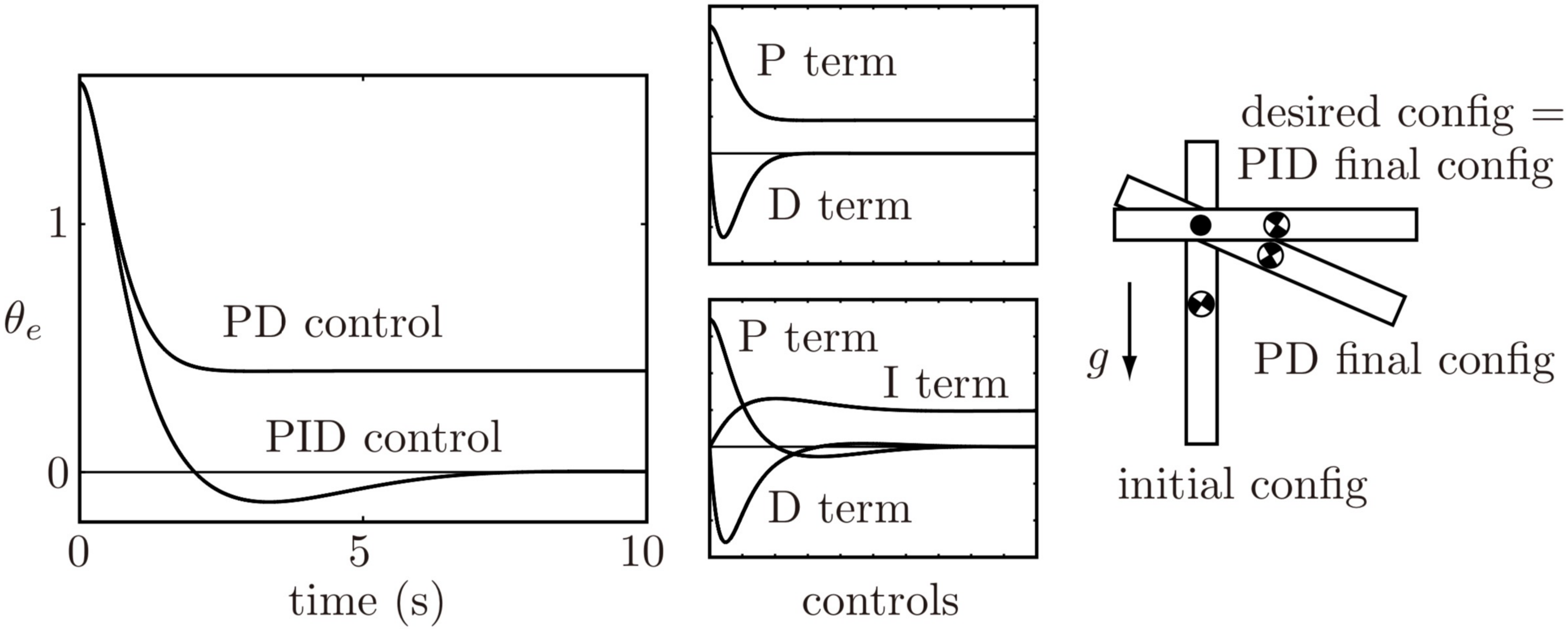
$$\begin{aligned} K_d &> -b \\ K_p &> 0 \end{aligned} \qquad \frac{(b+K_d)K_p}{M} > K_i > 0.$$

root locus for
K_p, K_d constant
and K_i increasing
from zero



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

PID vs PD control



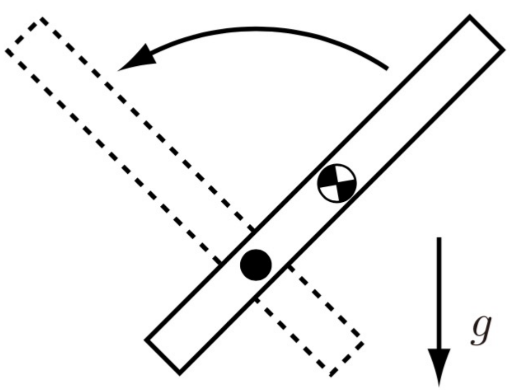
Mixed position and torque control

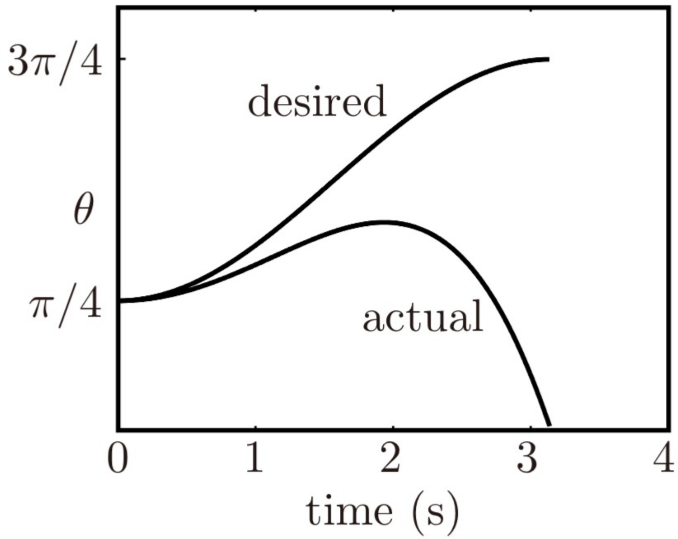
- **Feedforward torque control**:

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

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

- When will pure feedforward torque control work?
  - If the model is exact
  - No initial state errors





Mixed position and torque control

- **Feedback position control**: Deal with error dynamics when model inaccurate
- **Feedforward torque control**: use model information to improve performance
- Let's combine the PID position control with the robot dynamics model $\{\tilde{M}, \tilde{h}\}$ to achieve the error dynamics:

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

- Since $\ddot{\theta}_e = \ddot{\theta}_d - \ddot{\theta}$, we have:

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

Mixed position and torque control

- Substituting this acceleration into the robot dynamics model

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

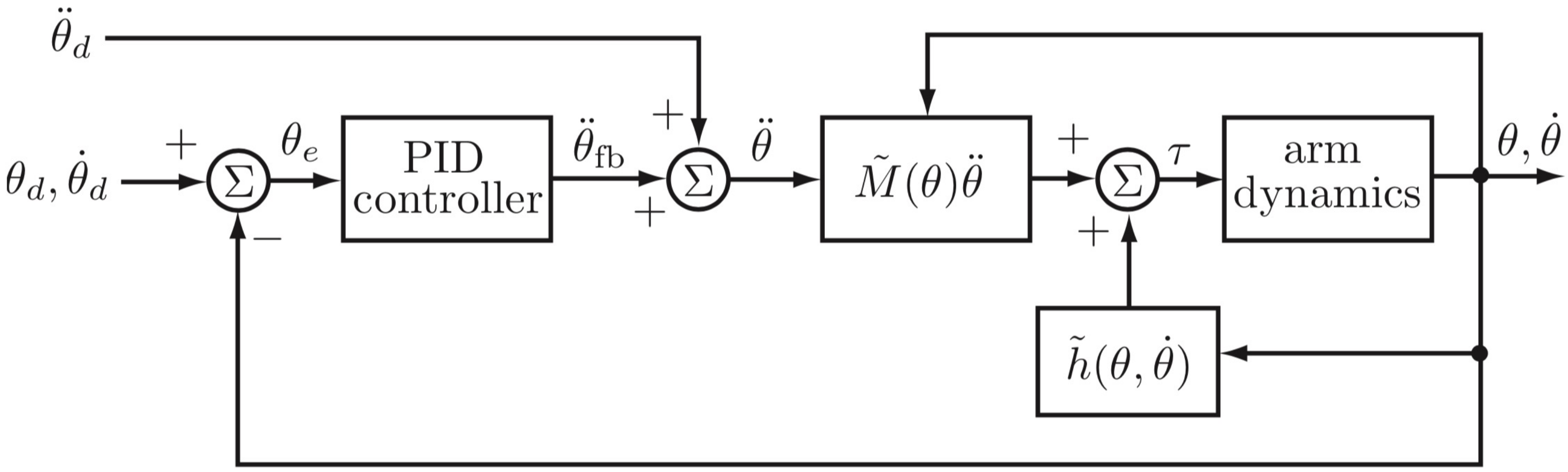
- We have:

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

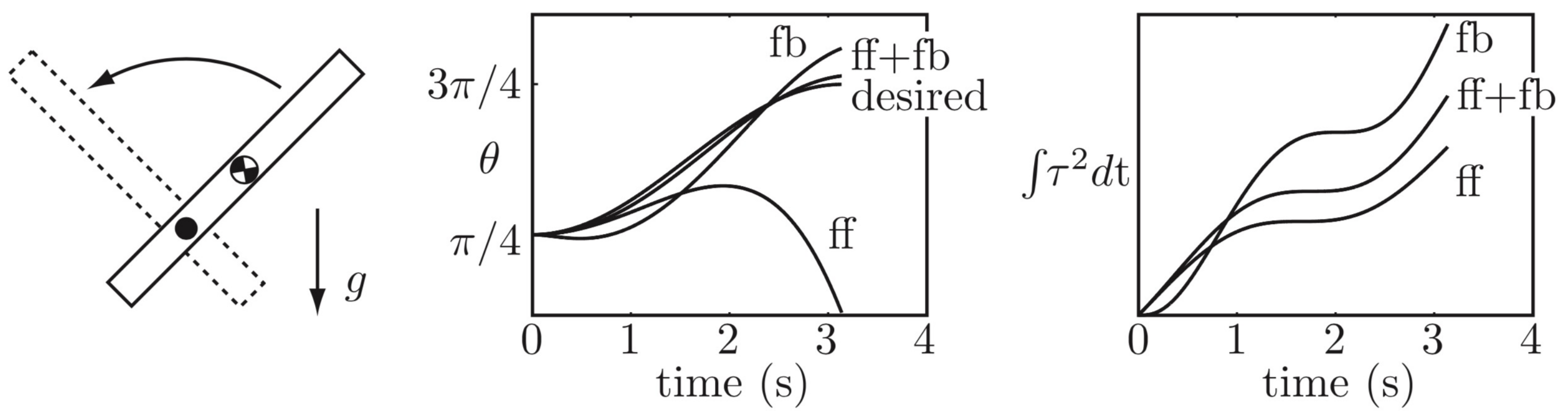
- This is also called the **computed torque controller**.

Mixed position and torque control

- Diagram of the computed torque controller



- A comparison



Control a multi-joint robot

- The above is considering the single link case, to extend it to a multi-joint robot, we just need to use the more general vector form of robot dynamics:

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

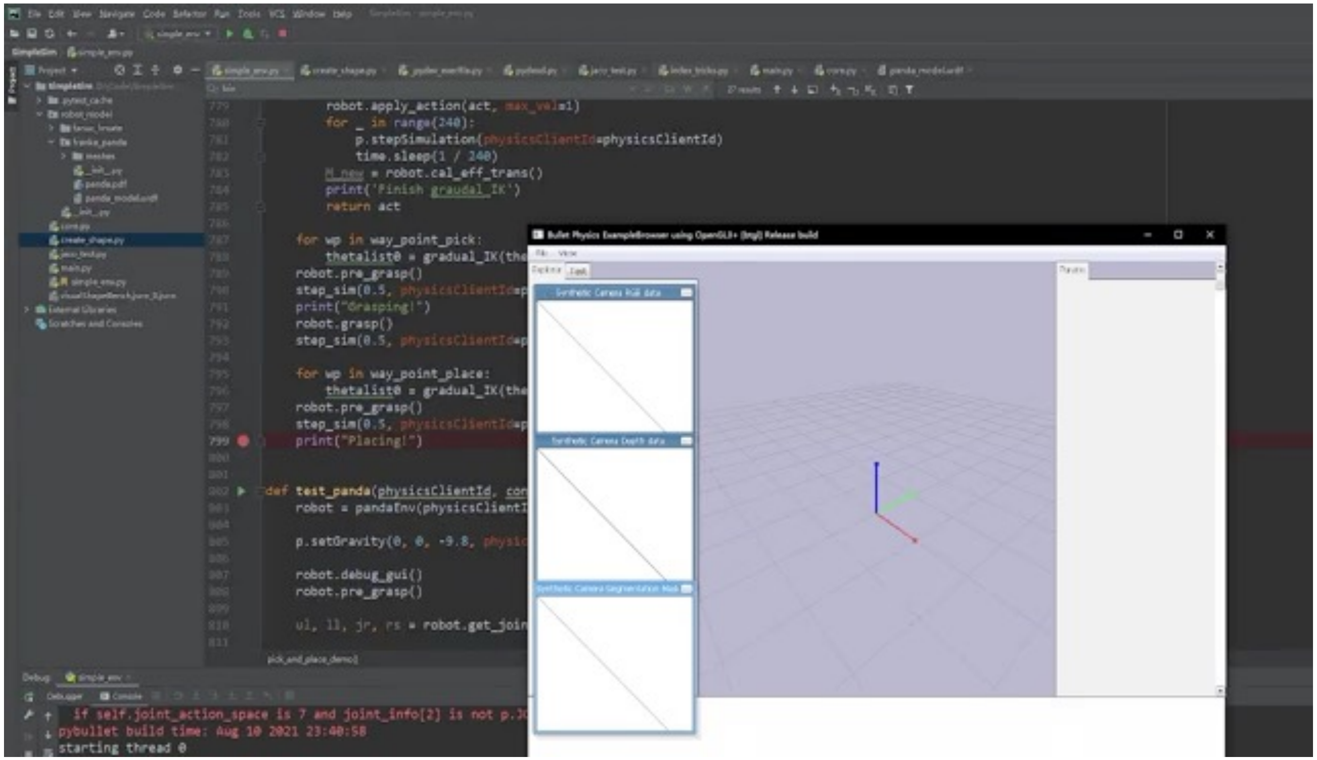
- We can use the same form of the computed torque control for the single link case:

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

where the controller gains are matrices $k_p I$, $k_i I$ and $k_d I$

Motion control examples

- Mixed position and torque control (PID+feedforward)



- Suitable for **model-based approach** where
  - Robot dynamics model and planned acceleration trajectory is available

Motion control examples

- Position control with torque inputs (PD)

- Suitable for general **learning-based methods**
  - Do not need to consider dynamics model and acceleration target

Force control

- Instead of to track a **desired motion**, the goal of **force control** is to apply a **desired wrench** to the environment
- Equation of motion with end-effector wrench:

$$\cancel{M(\theta)\ddot{\theta}} + \cancel{c(\theta, \dot{\theta})} + g(\theta) + \cancel{b(\dot{\theta})} + J^{\mathrm{T}}(\theta)\mathcal{F}_{\mathrm{tip}} = \tau$$

- Typically in force control the robot is moving slowly:

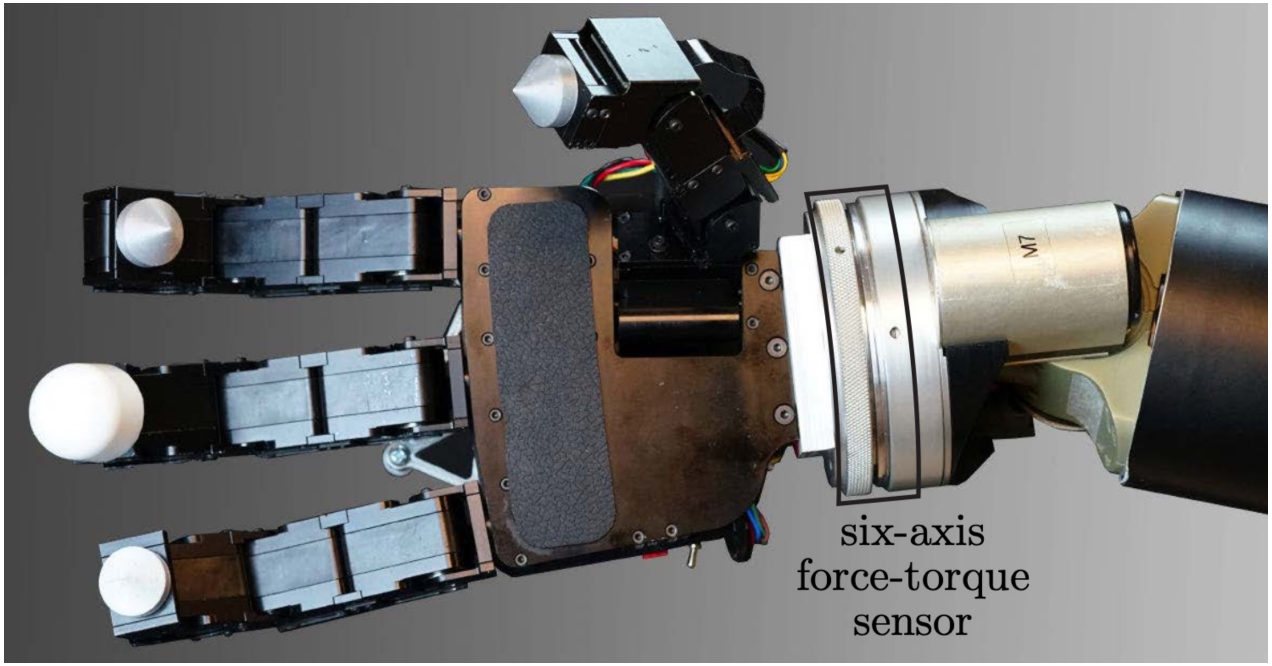
$$g(\theta) + J^{\mathrm{T}}(\theta)\mathcal{F}_{\mathrm{tip}} = \tau$$

- Force control **without** end-effector measurement feedback:

$$\tau = \tilde{g}(\theta) + J^{\mathrm{T}}(\theta)\mathcal{F}_d$$

Force-torque sensor

- To improve force control, we can equip the robot with a **force-torque sensor** at the end-effector



Force control with force feedback

- With the force measurement, we can add **force feedback**

$$\tau = \tilde{g}(\theta) + J^{\mathrm{T}}(\theta)\left(\mathcal{F}_d + K_{fp}\mathcal{F}_e + K_{fi}\int \mathcal{F}_e(\mathrm{t})d\mathrm{t}\right)$$

$$\mathcal{F}_e = \mathcal{F}_d - \mathcal{F}_{\mathrm{tip}}$$

- The force feedback helps to eliminate steady state error with there's modeling error and disturbances
- Derivative control is generally not used, since the force sensor is usually very noisy

Thank you!

清華大學 交叉信息研究院
Institute for Interdisciplinary Information Sciences, Tsinghua University

ISR Lab
Intelligent Systems and Robotics Lab