

Motion Control and PID

EECS 367

Intro. to Autonomous Robotics

ROB 320

Robot Operating Systems

Winter 2022

autorob.org







MAKE GIFS AT GIF SOUP.COM

Michael Jackson - "Smooth Criminal" - https://youtu.be/h_D3VFfhvs4

Anti-gravity lean

Anti-gravity lean



by DJ_O



Ignika Smooth criminal







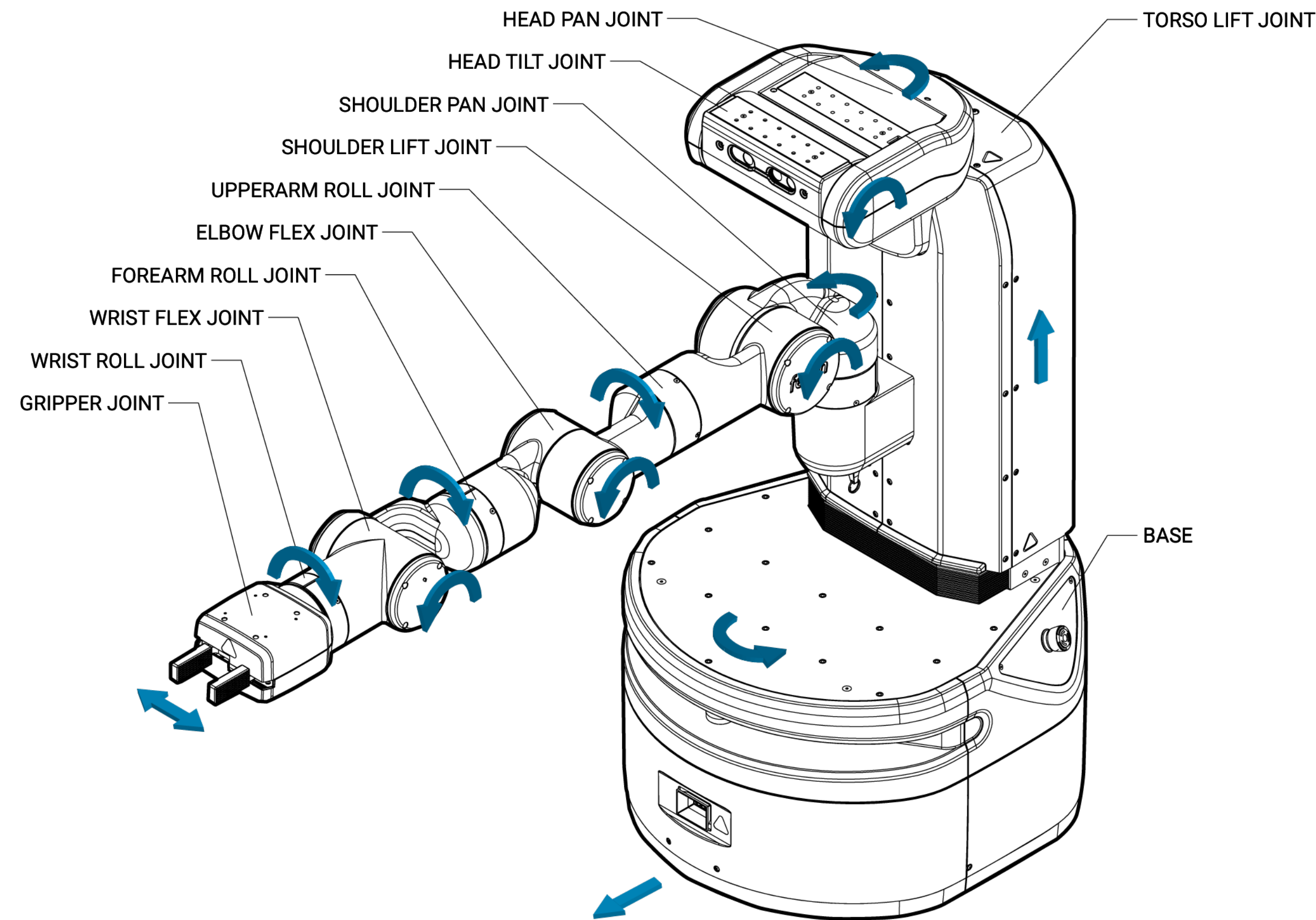
Biped Robotics Lab @ Michigan - <https://youtu.be/0gauVSUJzd0>

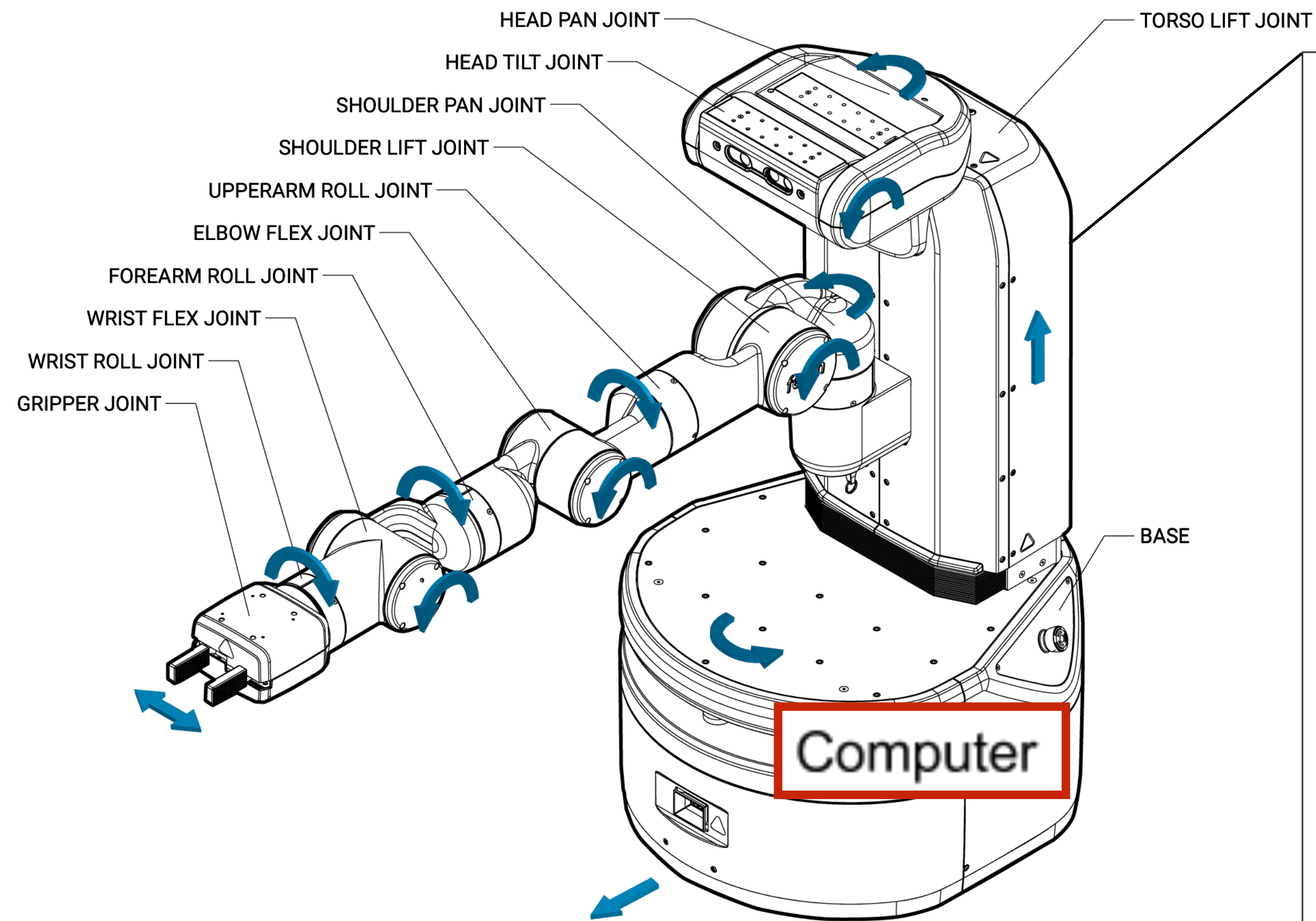
Control must be fast and often



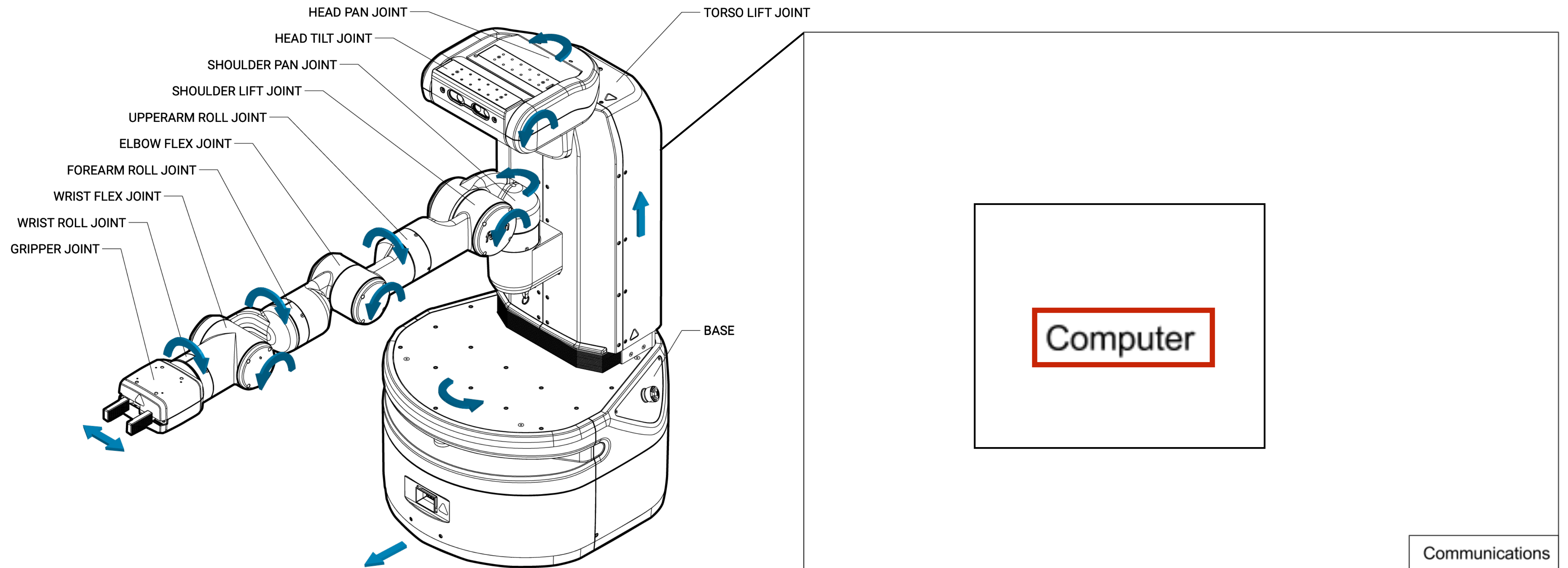
Communication inside a robot is
like its central nervous system

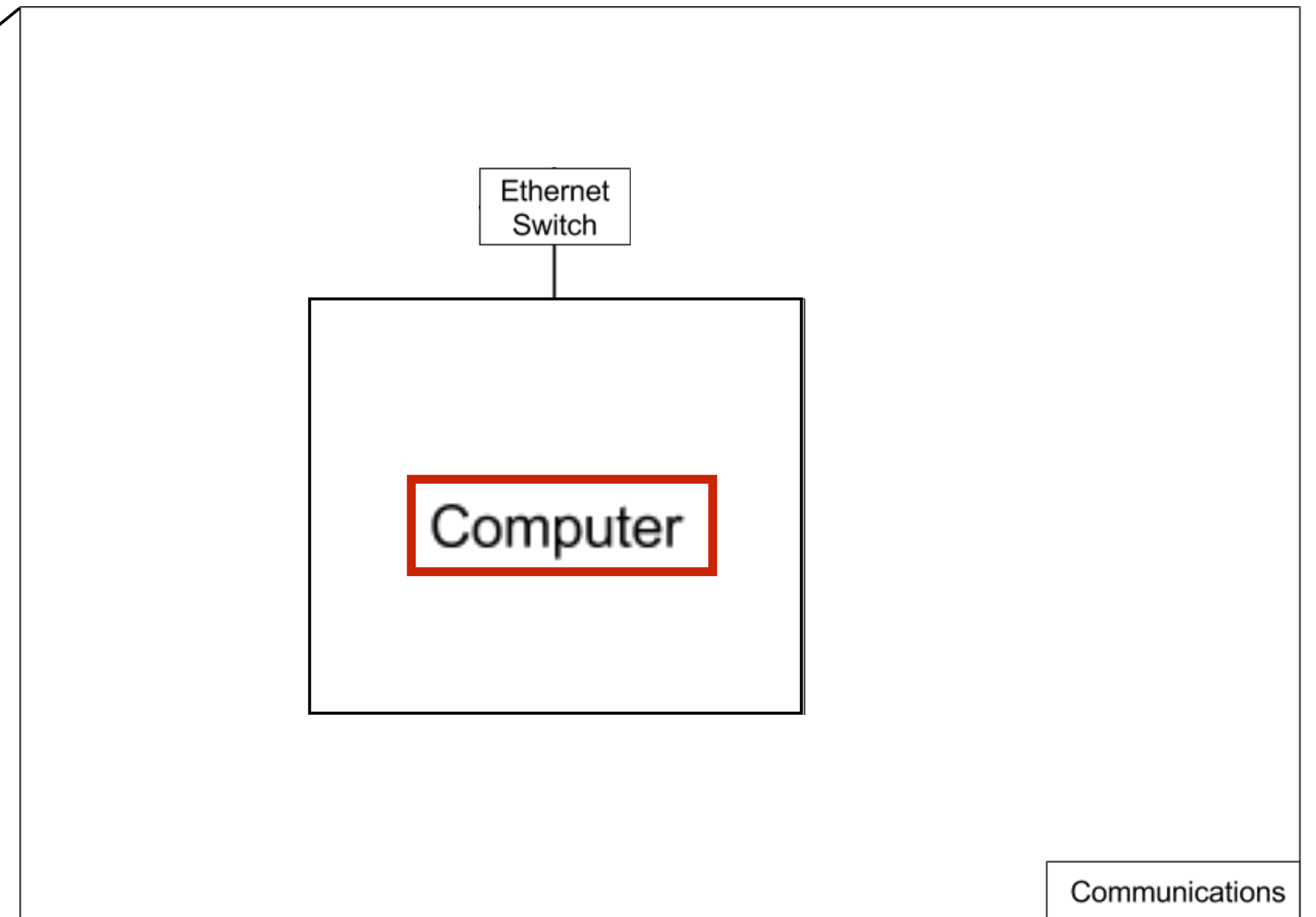
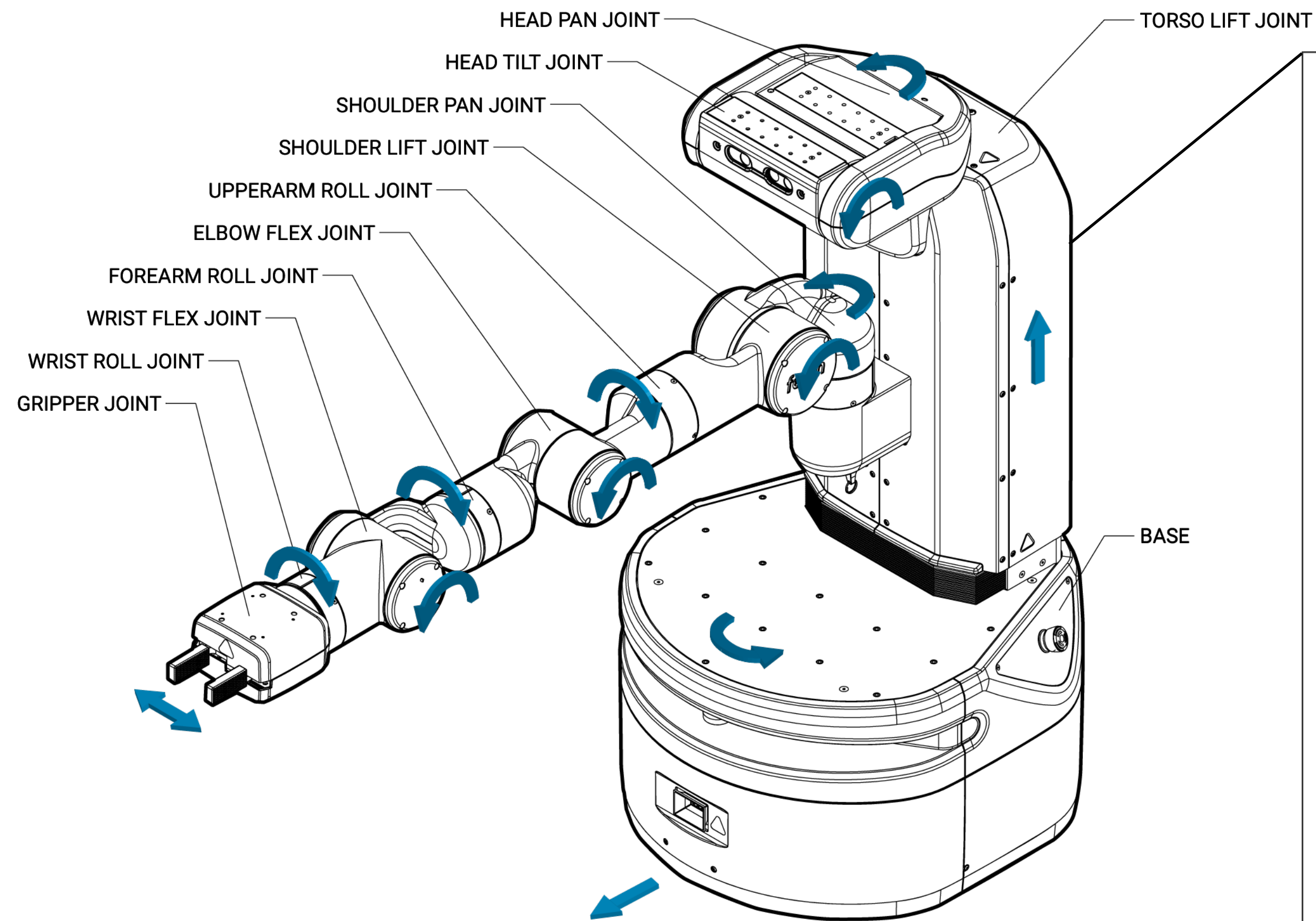
Communication inside a robot is like its central nervous system

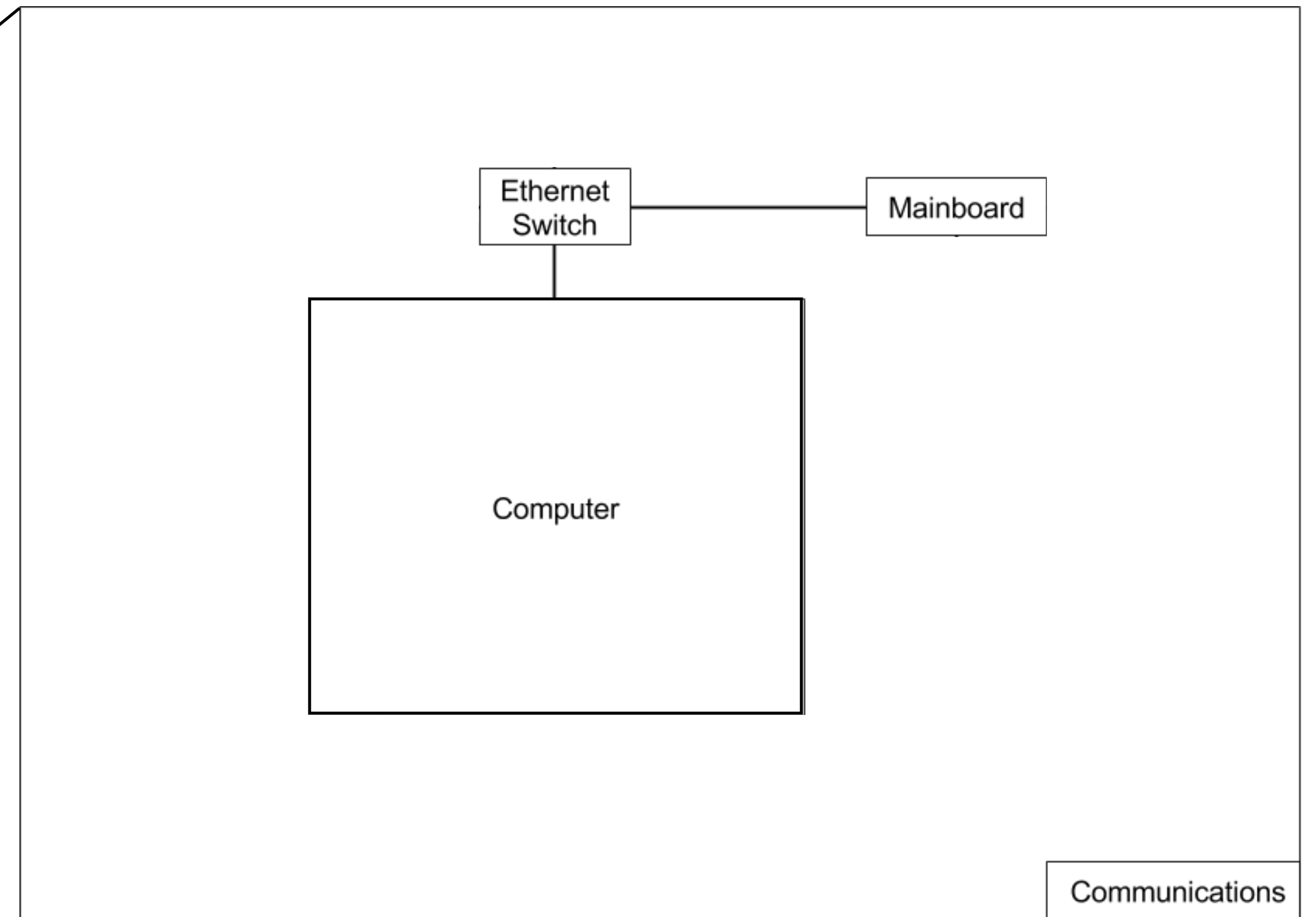
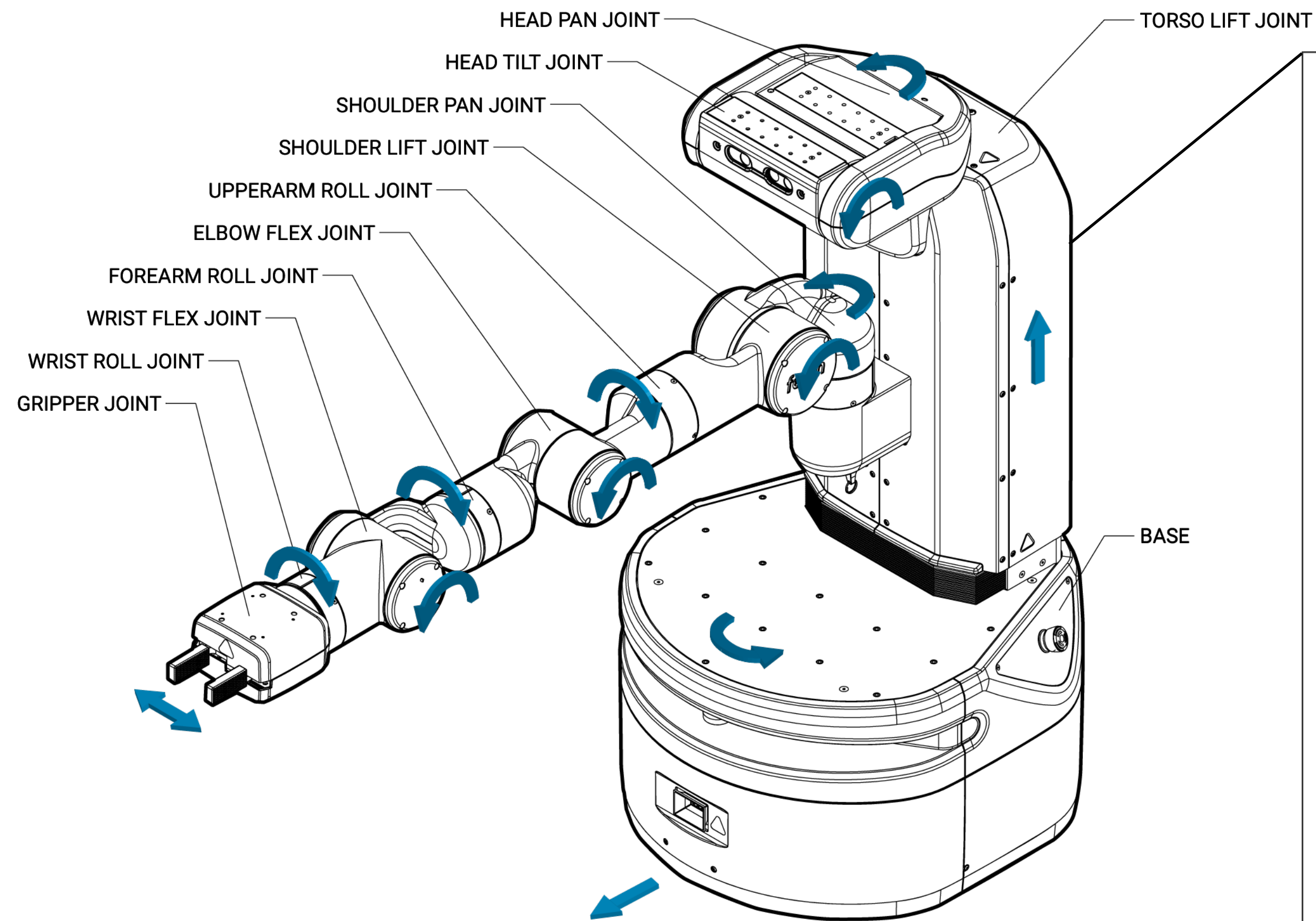


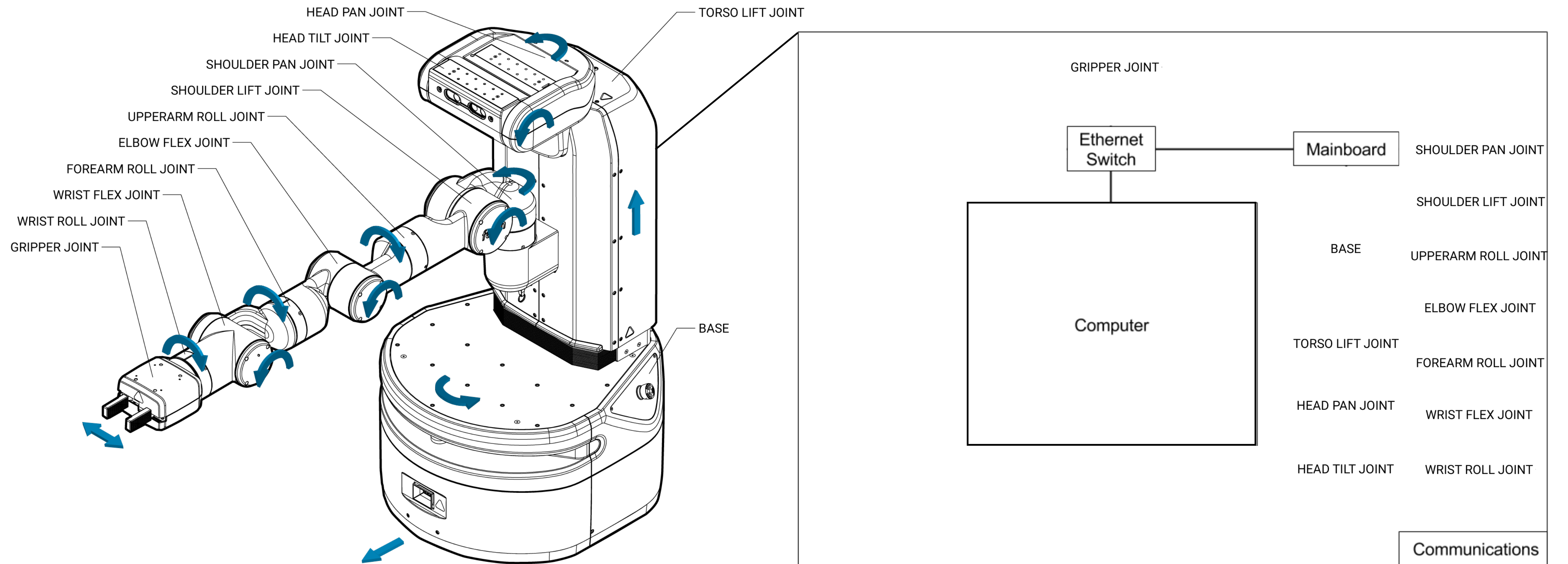


Communications

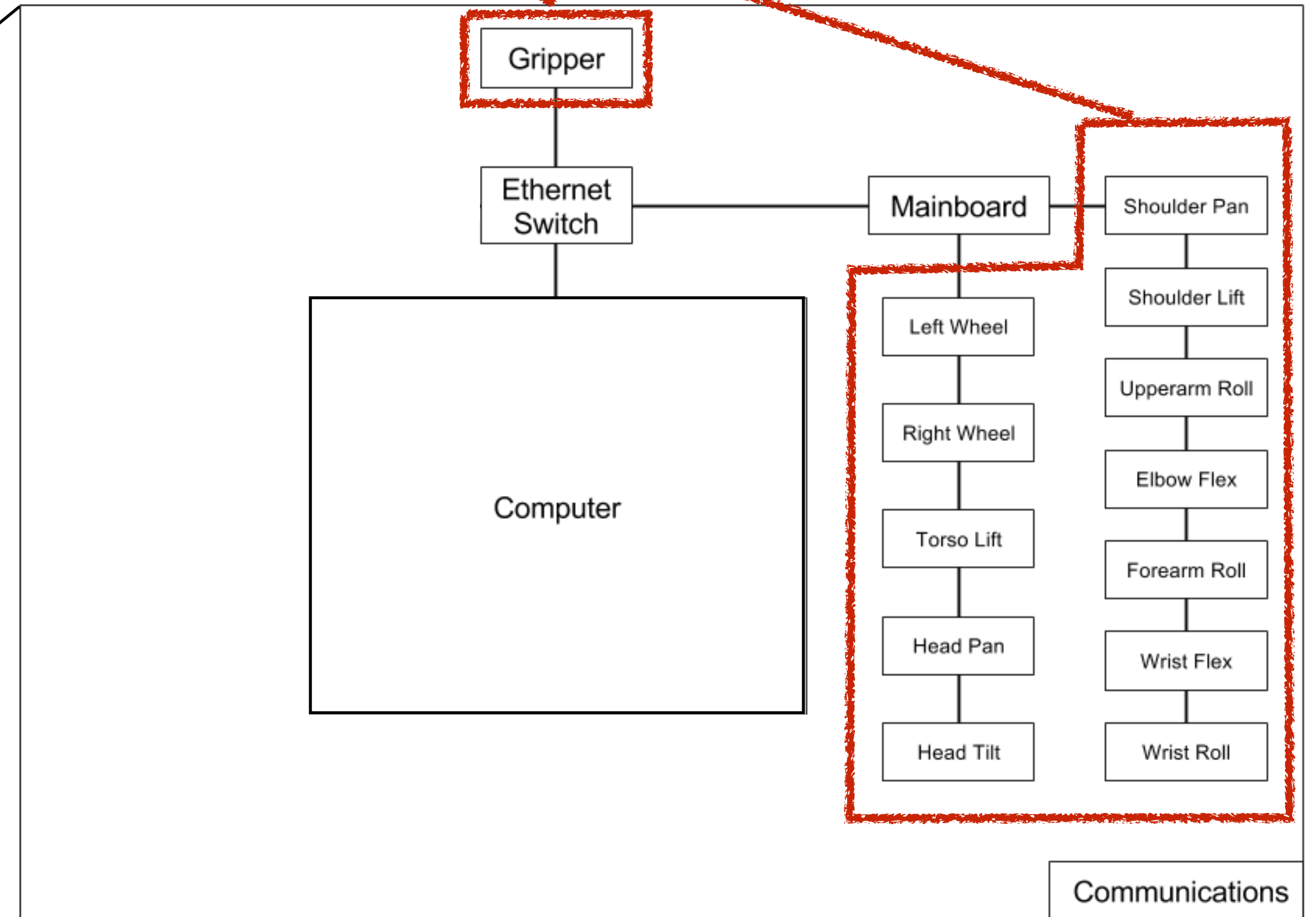
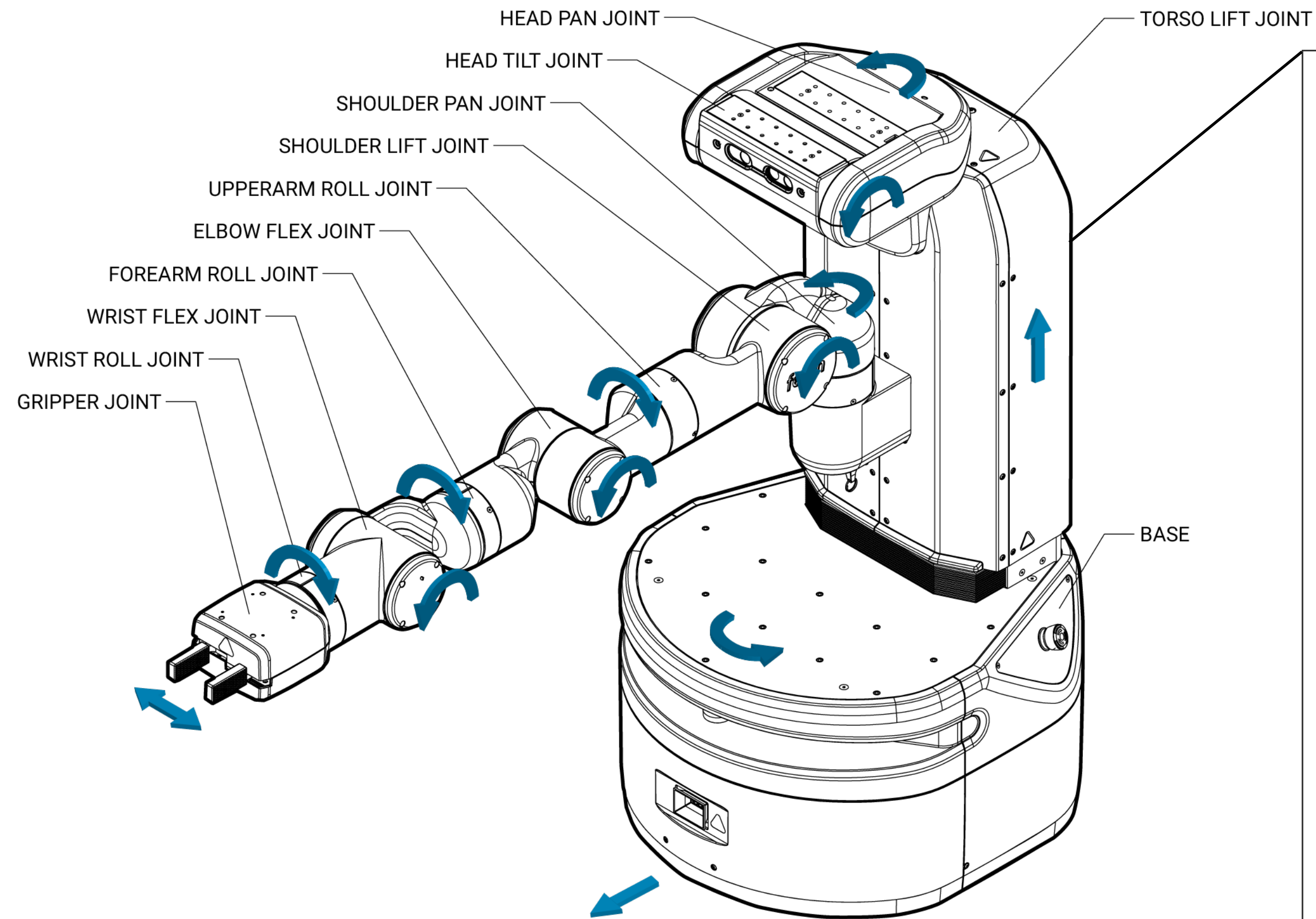




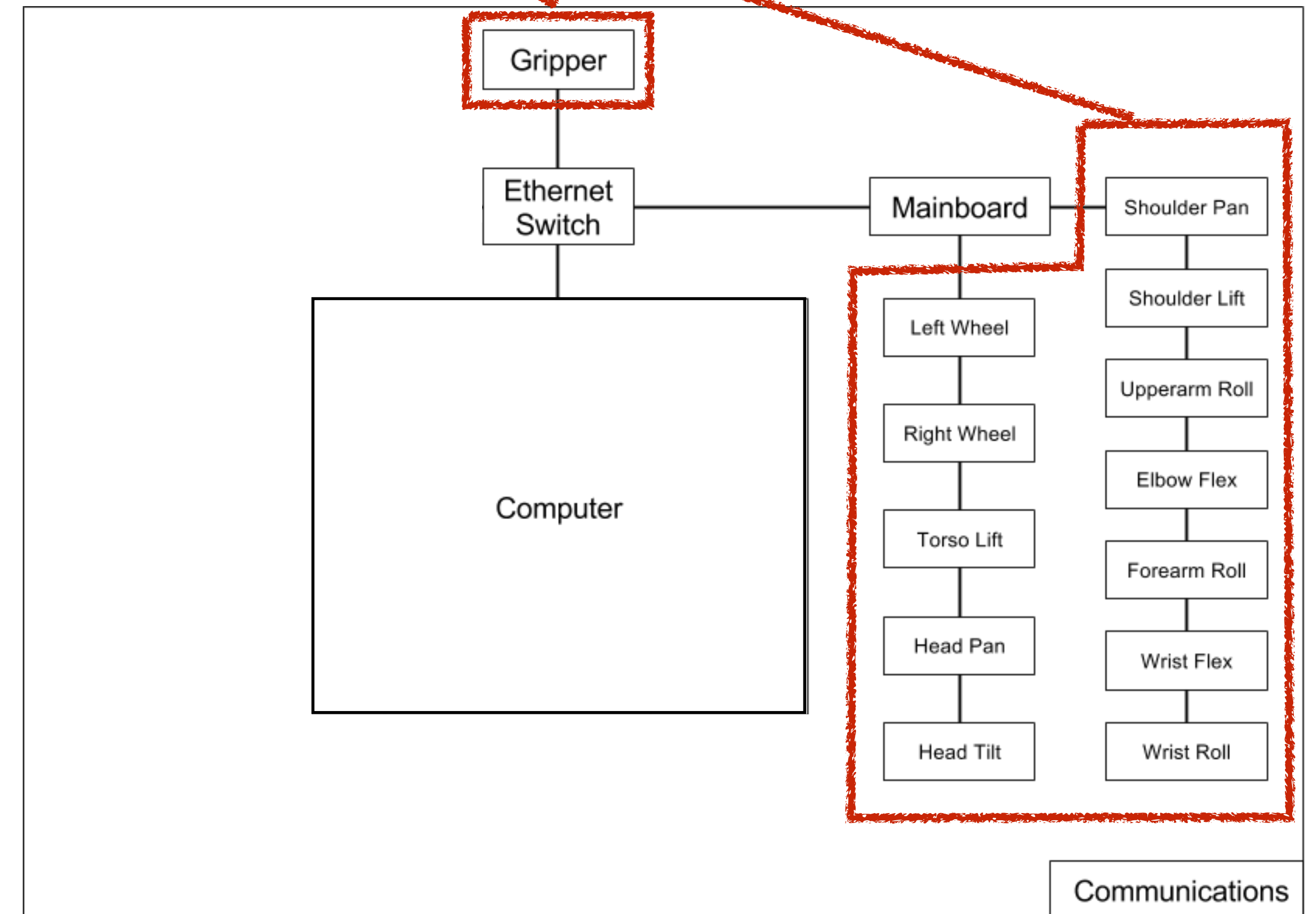
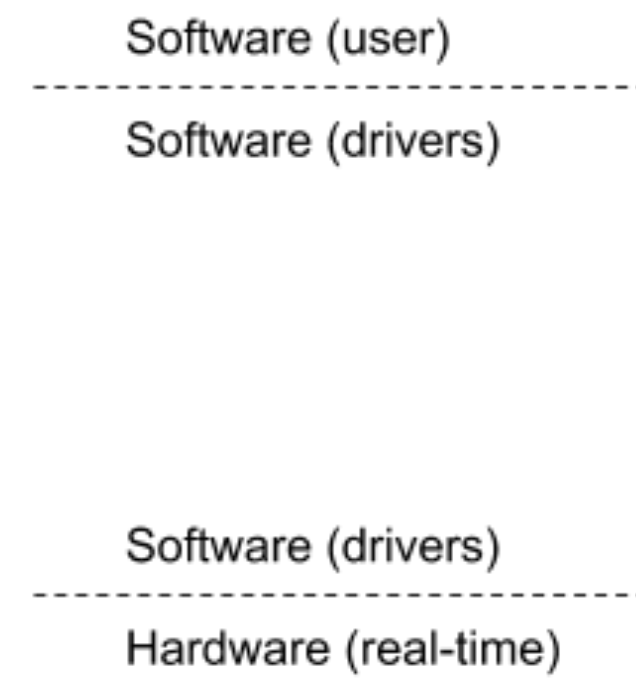
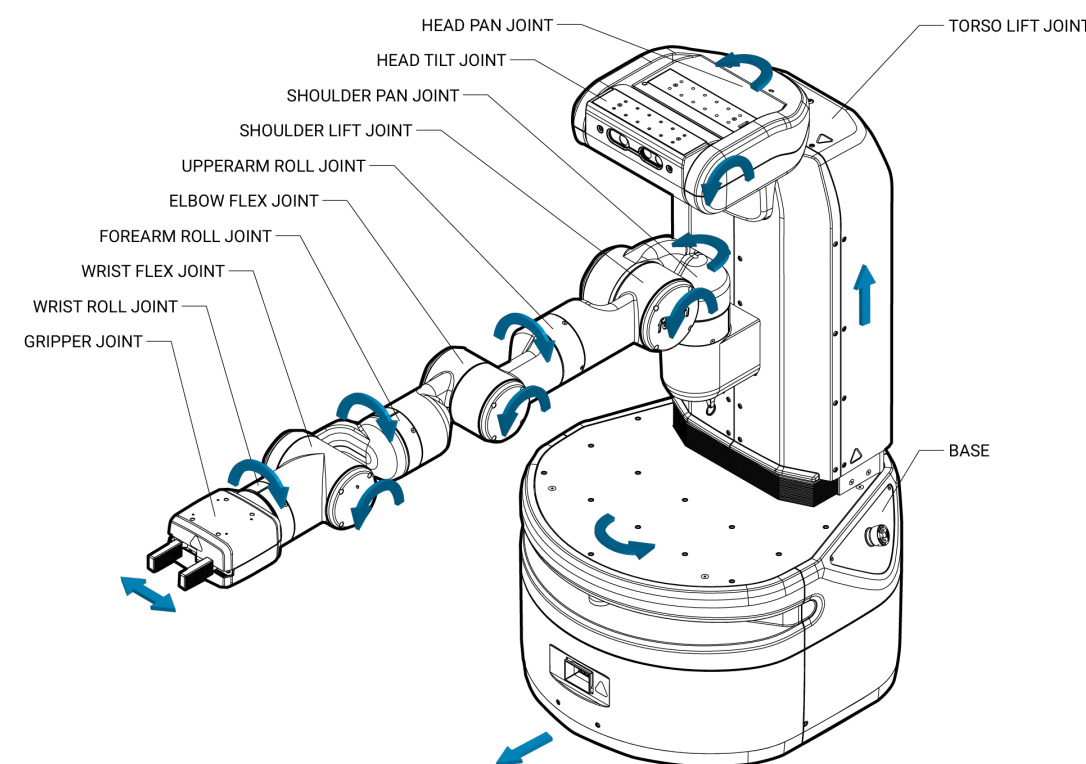
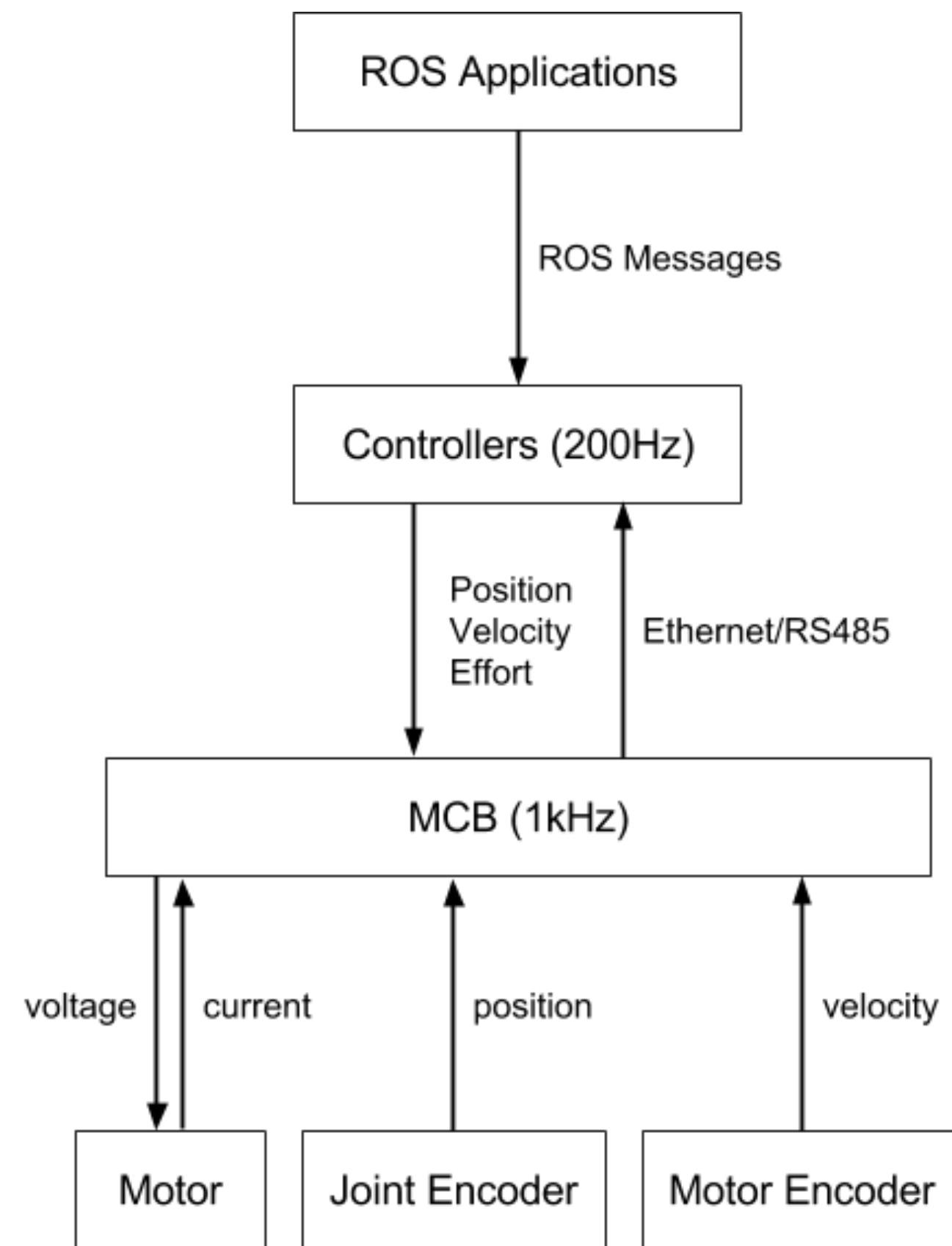


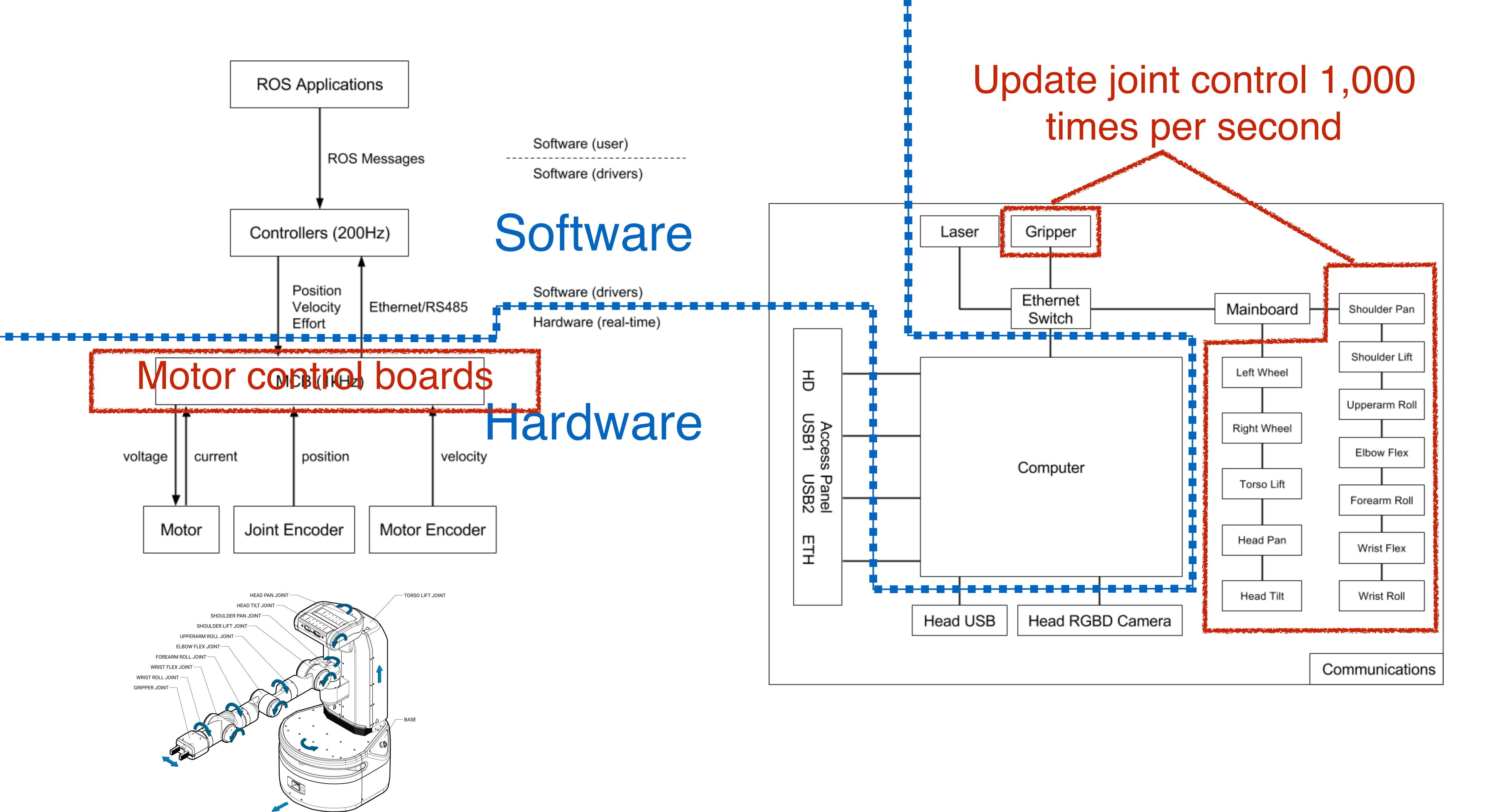


Update joint control 1,000 times per second

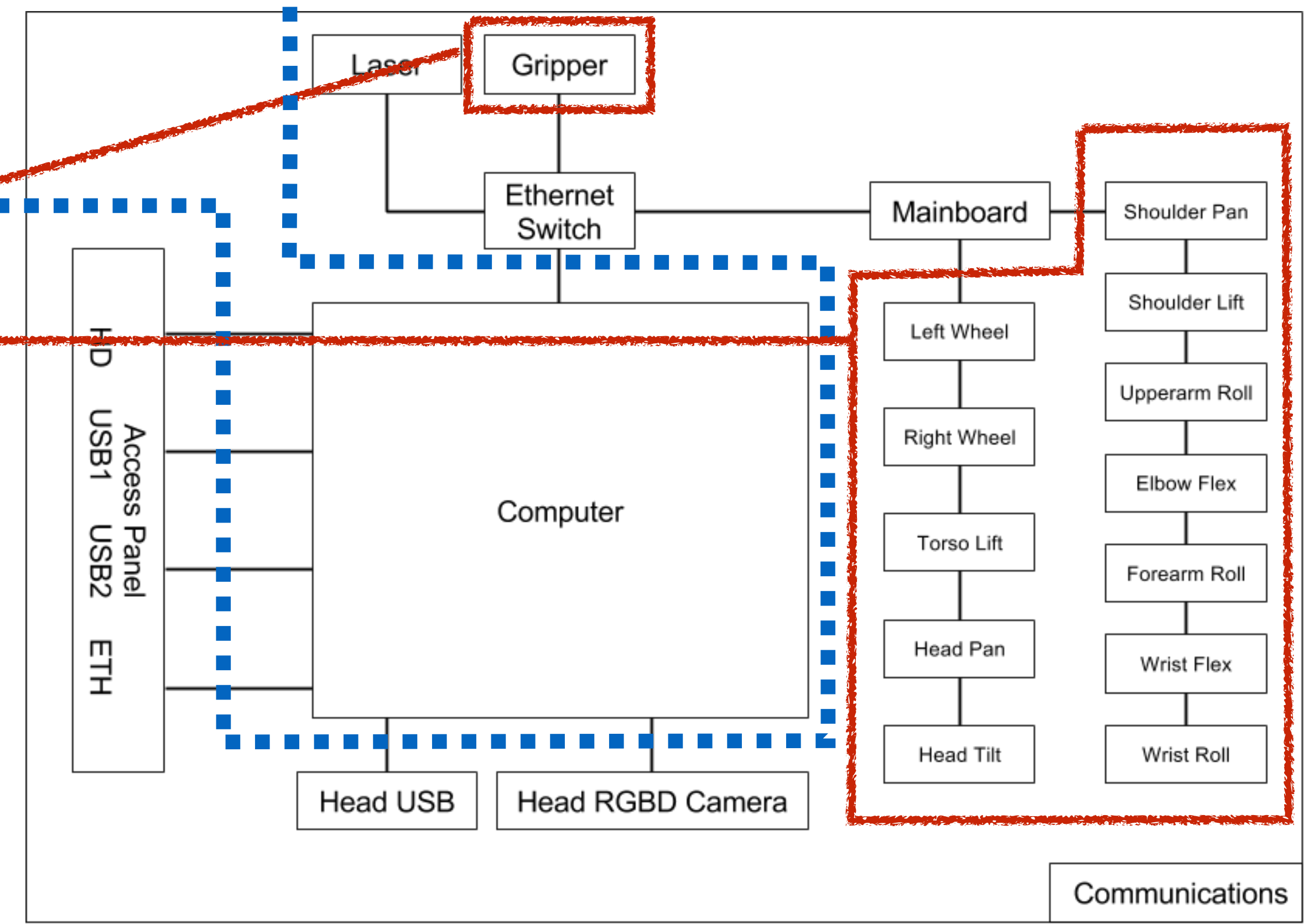
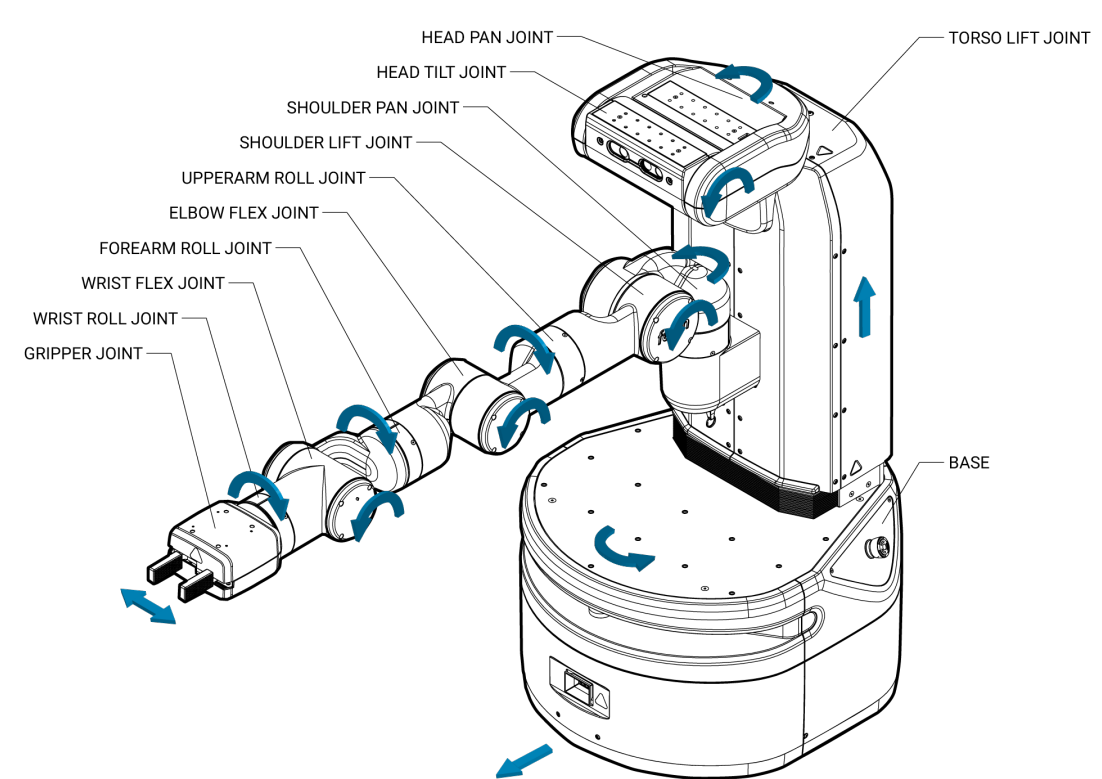
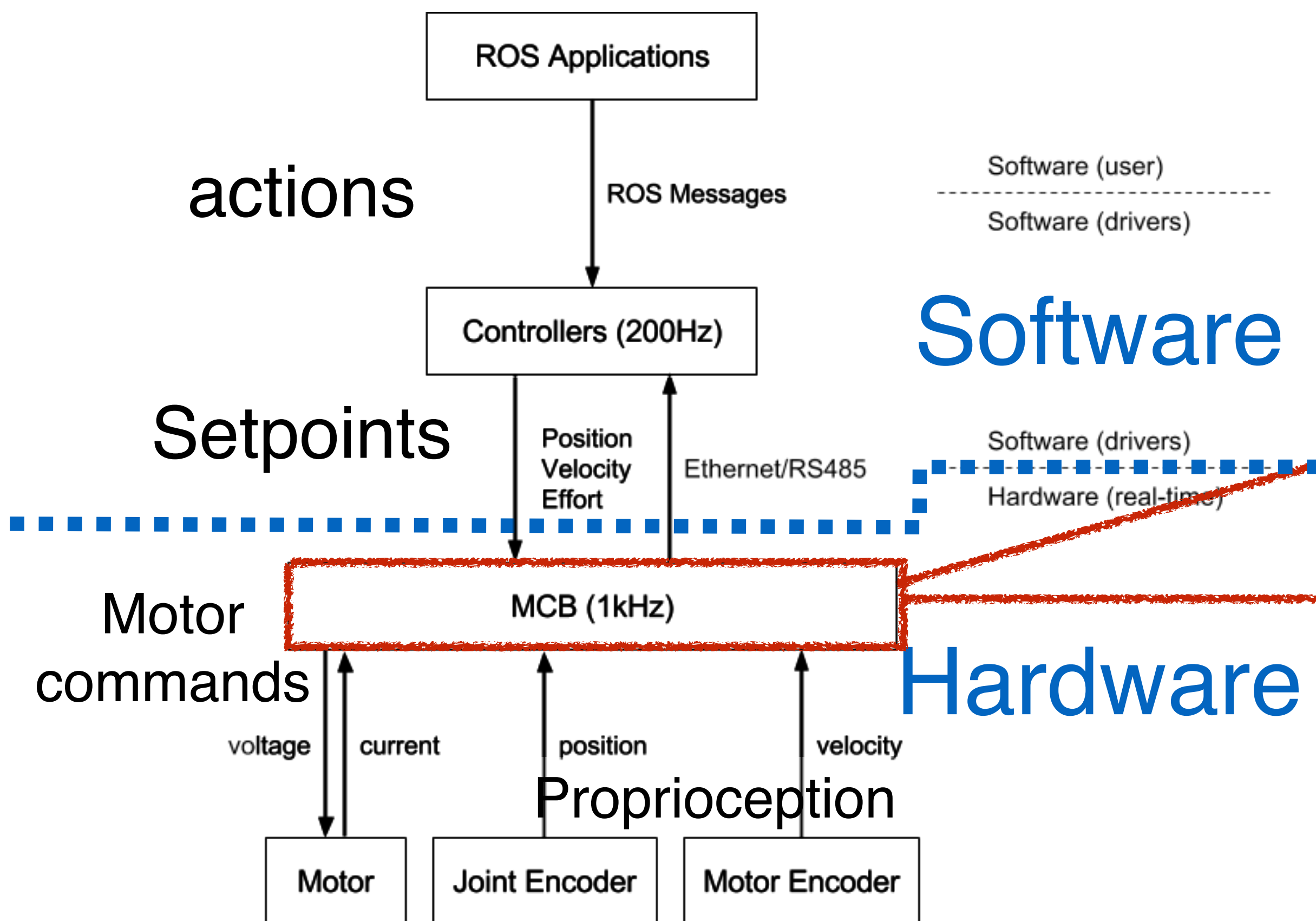


Update joint control 1,000 times per second



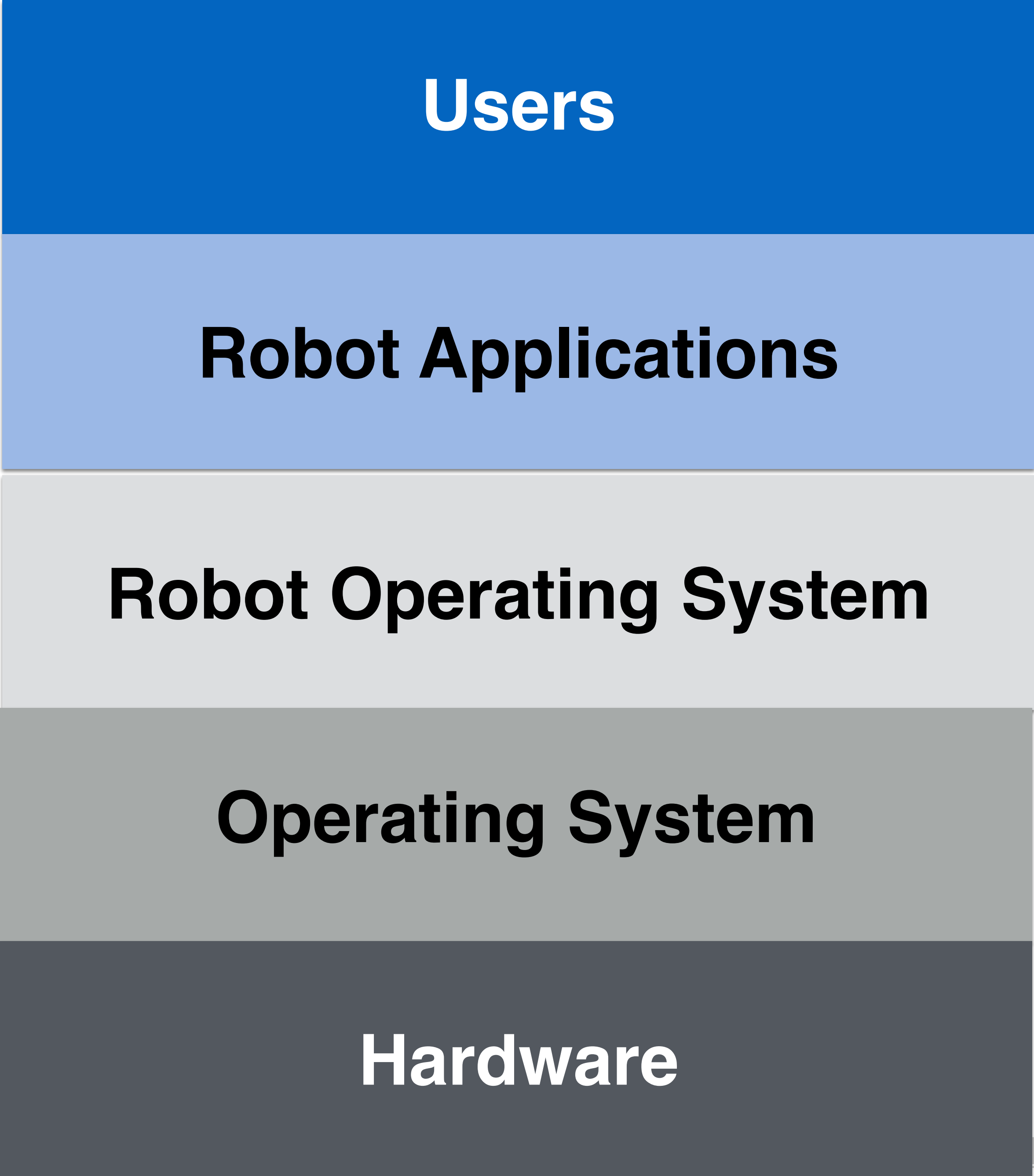
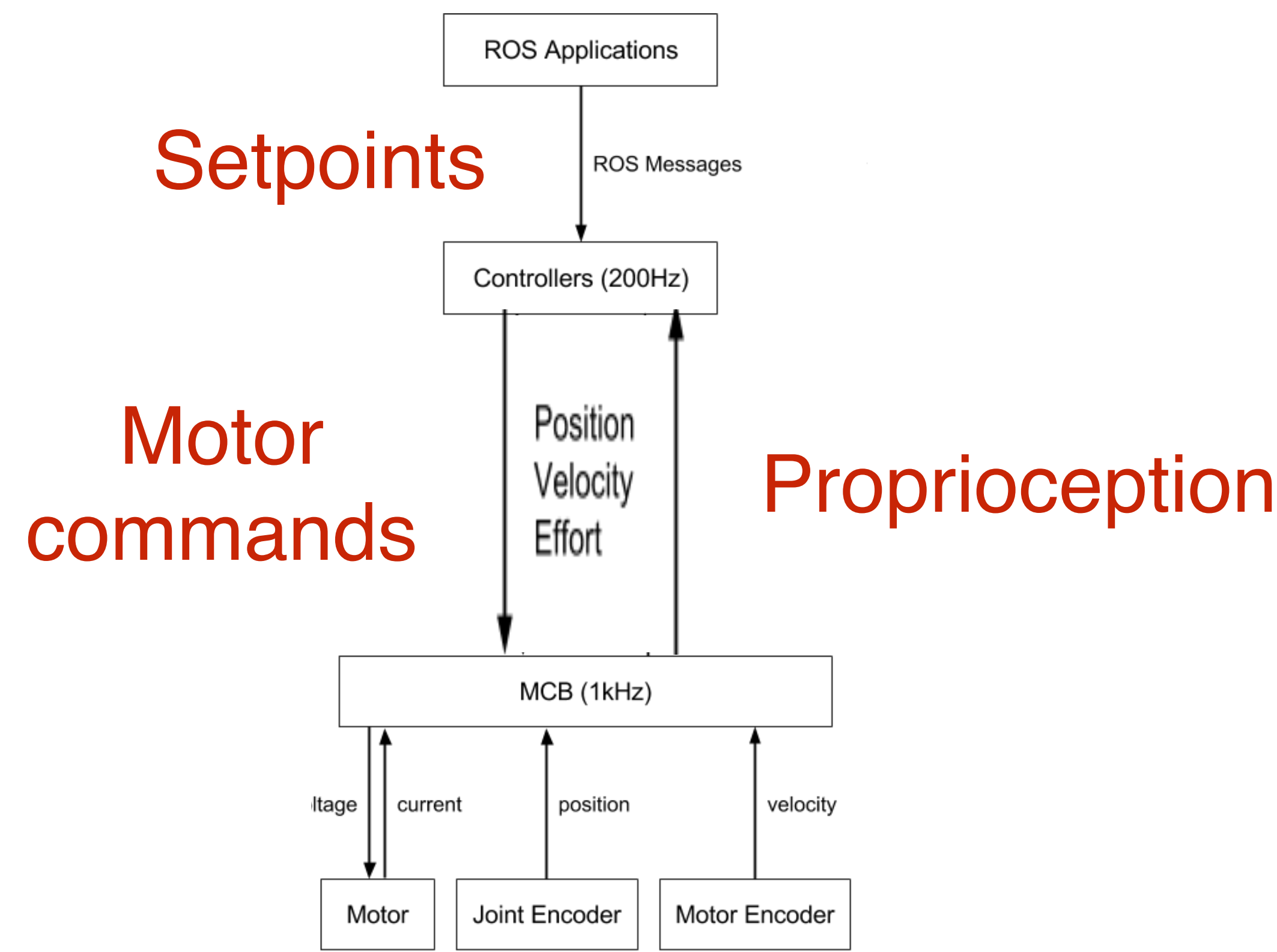


Motor control boards
Update joint control 1,000 times per second



Motion control:

process to achieve and maintain a desired state (or setpoint) for a joint through applying motor commands based on current proprioceived state



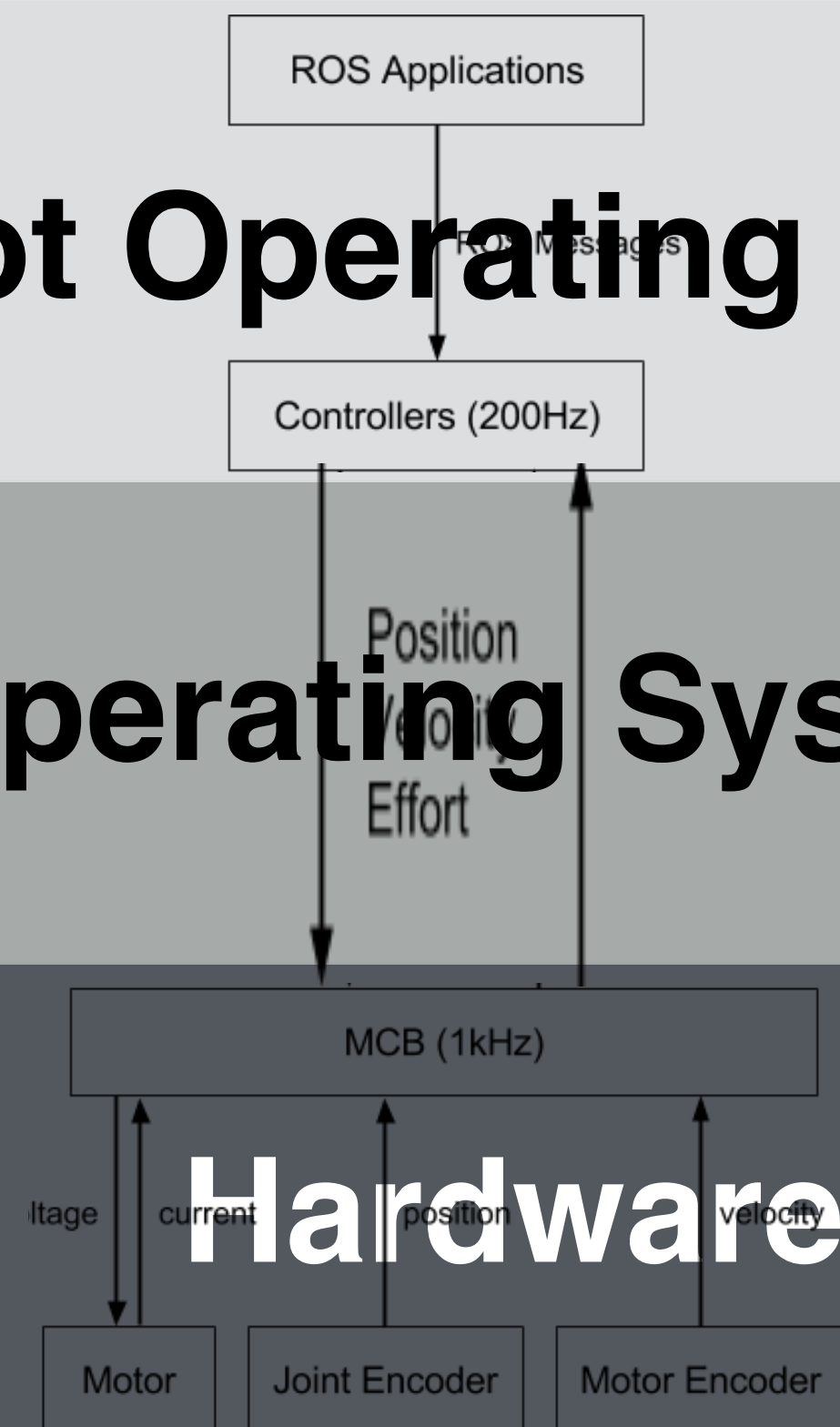
Users

Robot Applications

Robot Operating System

Operating System

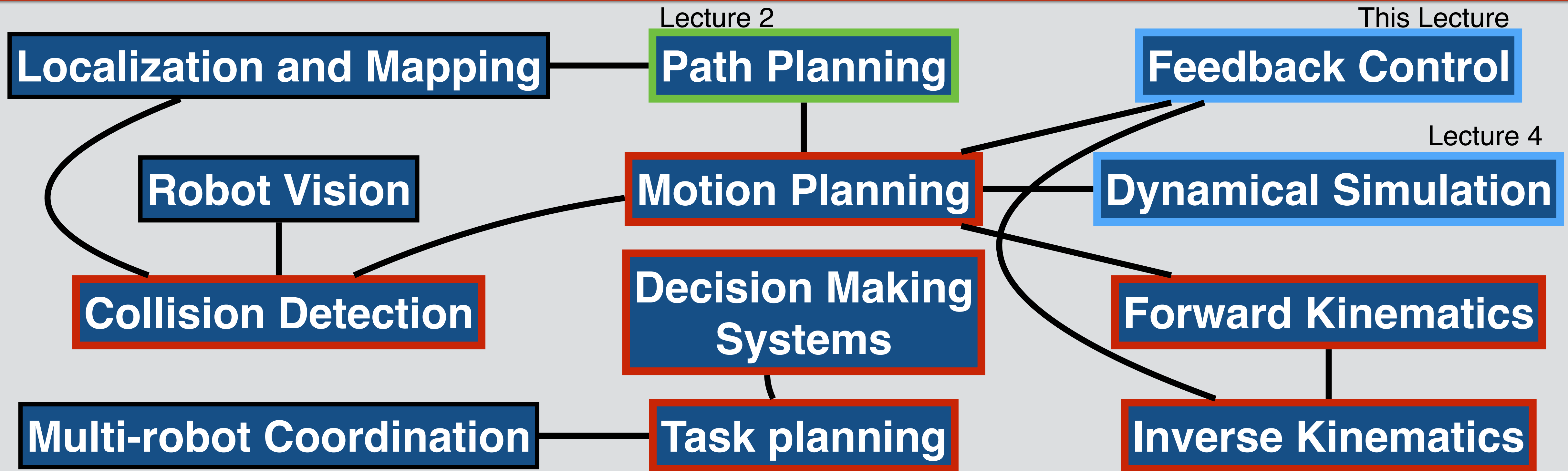
Hardware



Motion control

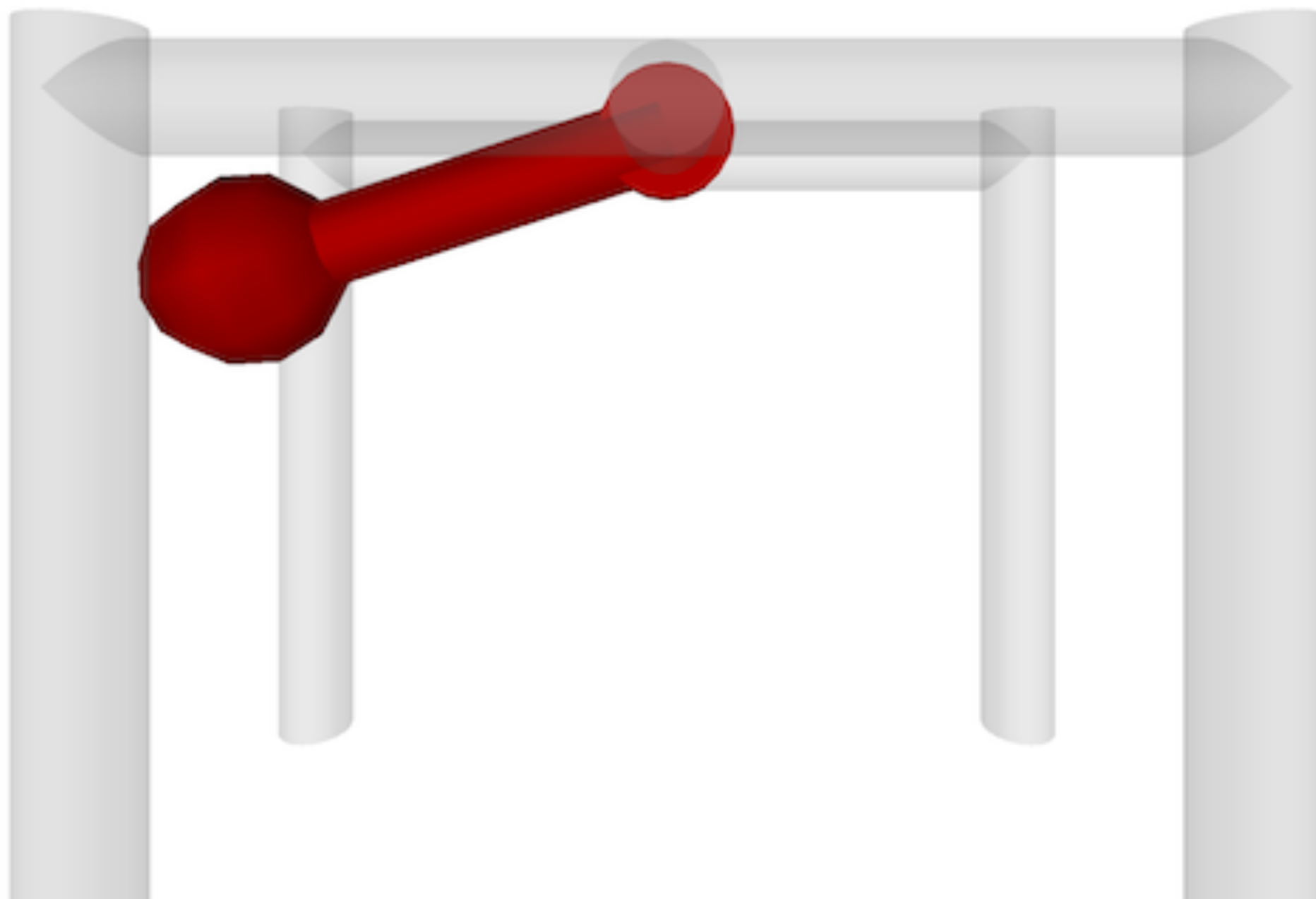
Robot Operating System

Covered at breadth in AutoRob



Robot Middleware Architecture (via Interprocess Communication)

Project 2: Pendularm



Lecture 4

Dynamical Simulation

Equations of motion: Simple pendulum
Integrators: Euler, Verlet, Velocity Verlet, RK4

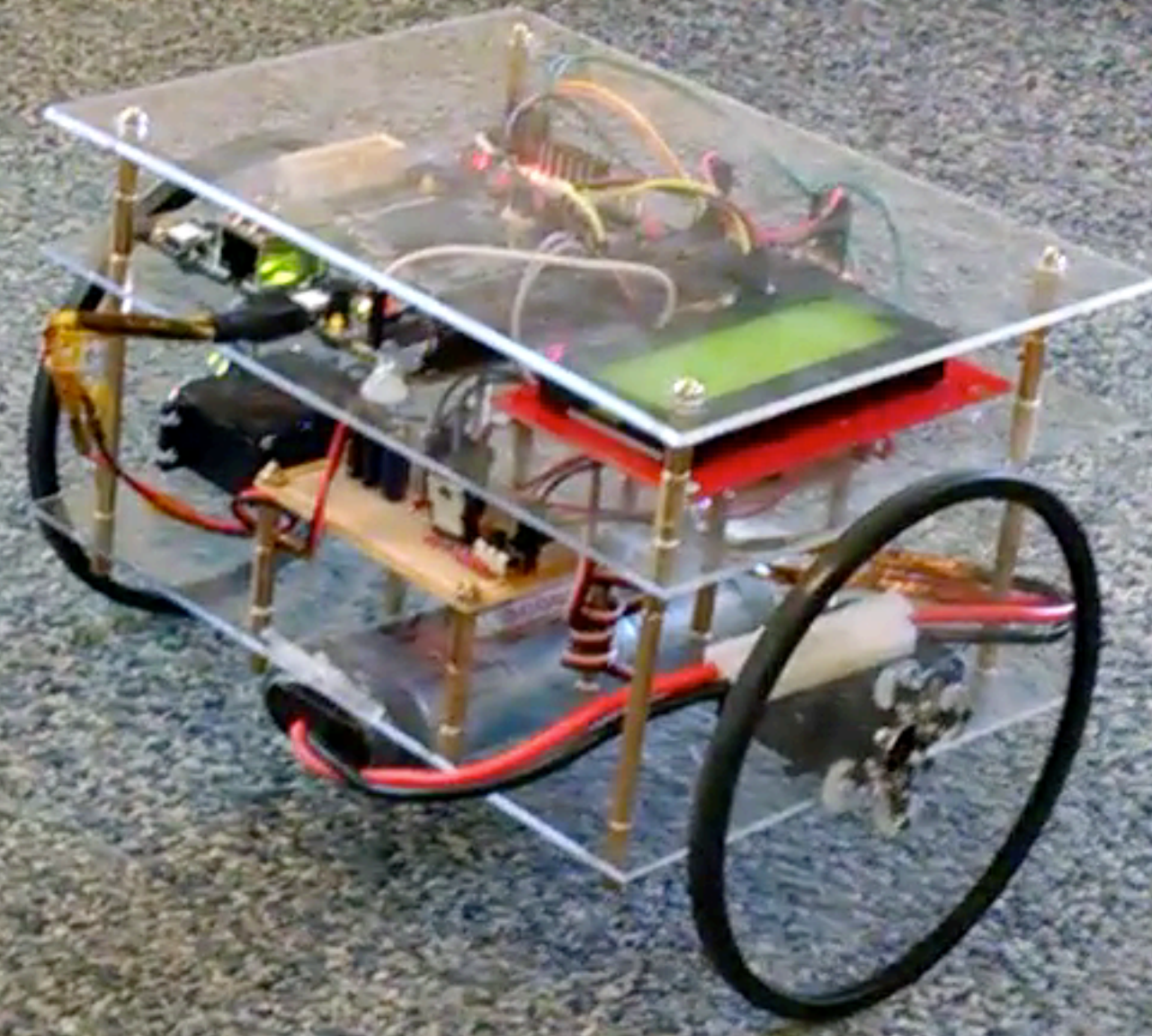
This Lecture

Feedback Control

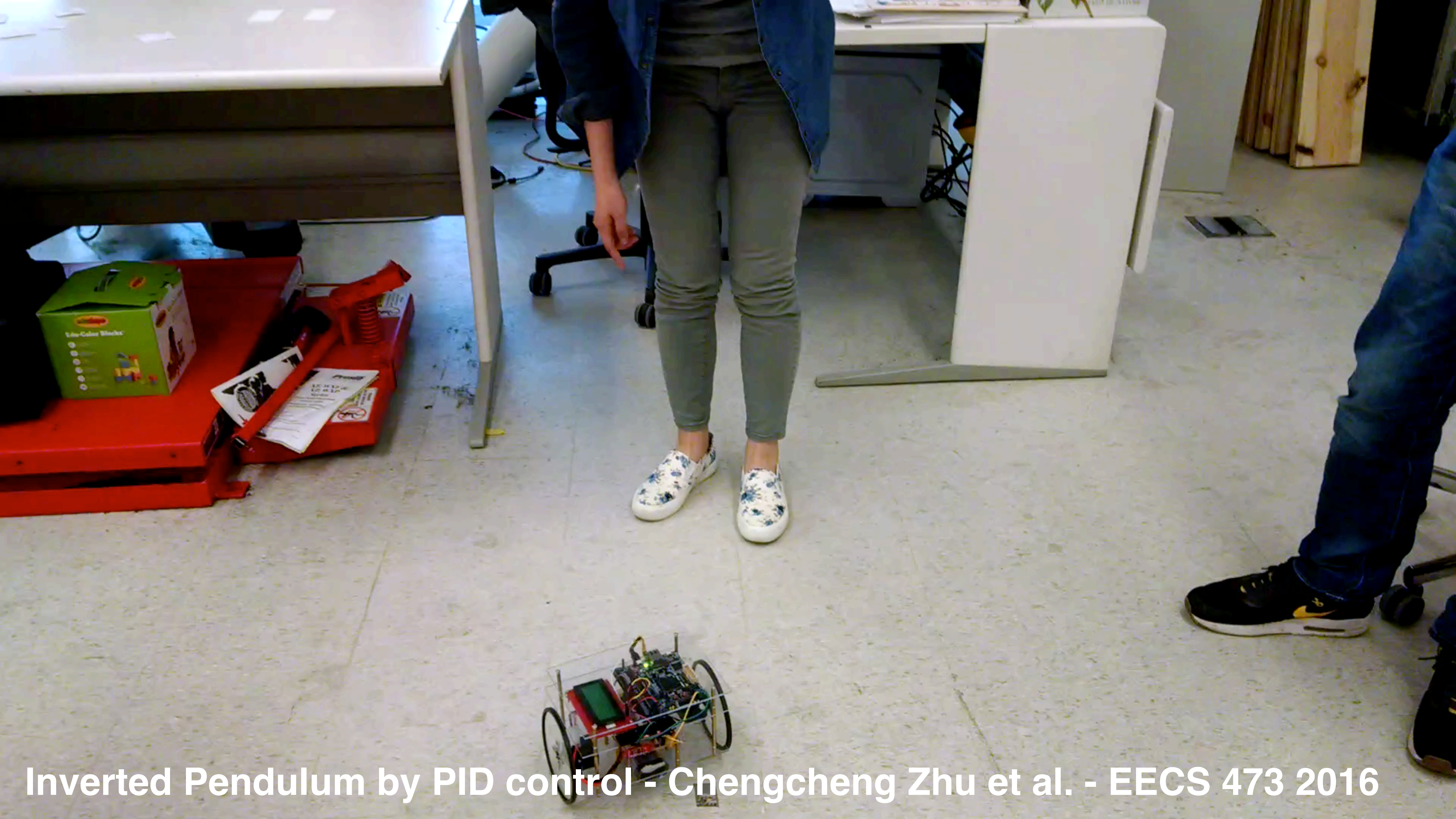
PID Control of 1 DoF robot

We will put it all together later in this lecture

What can you do with
pendularm?



Inverted Pendulum by PID control - Chengcheng Zhu et al. - EECS 473 2016



Inverted Pendulum by PID control - Chengcheng Zhu et al. - EECS 473 2016



Monster Inverted Pendulum - K. French et al. - ROB 550 - Fall 2016

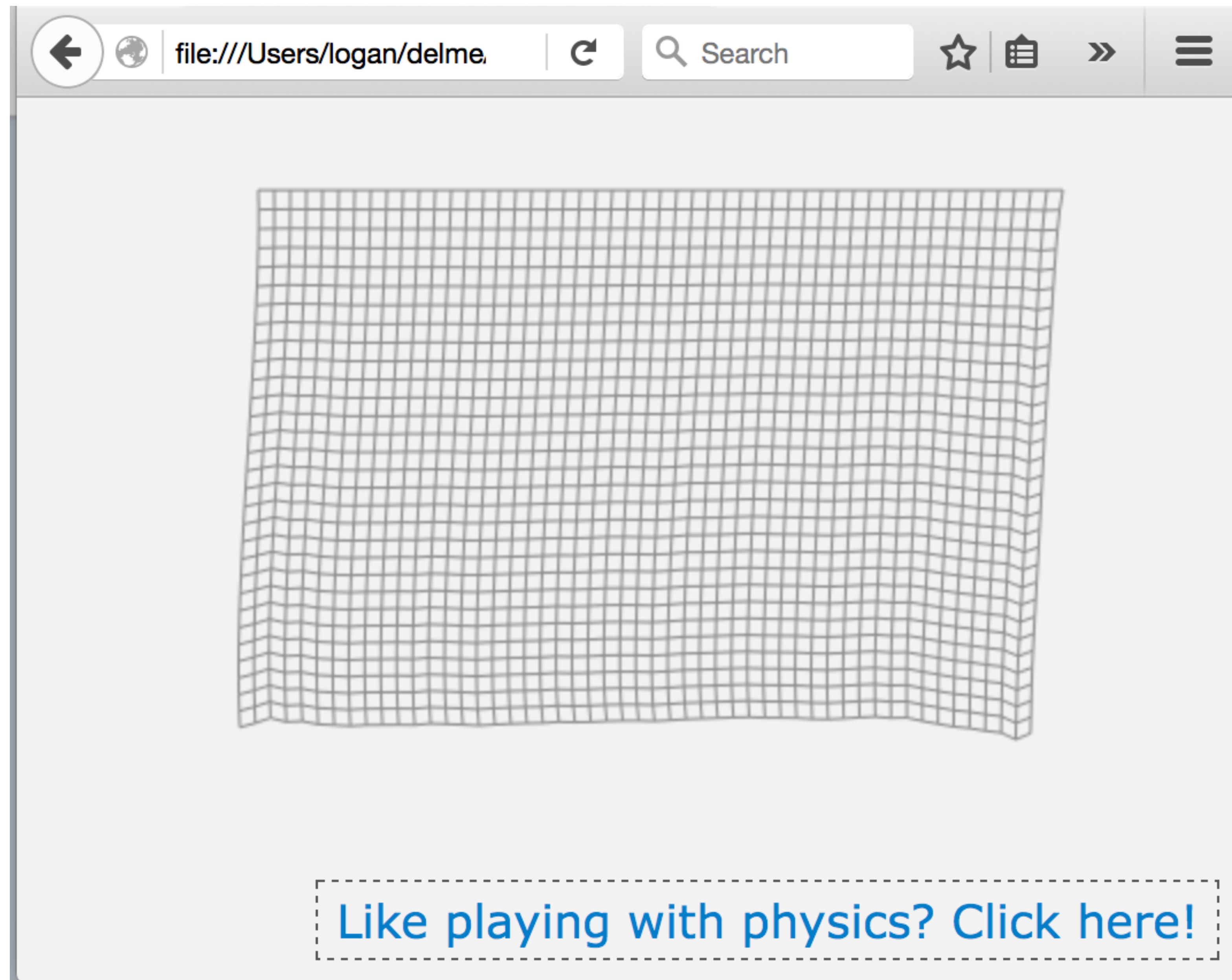
Building up to more...



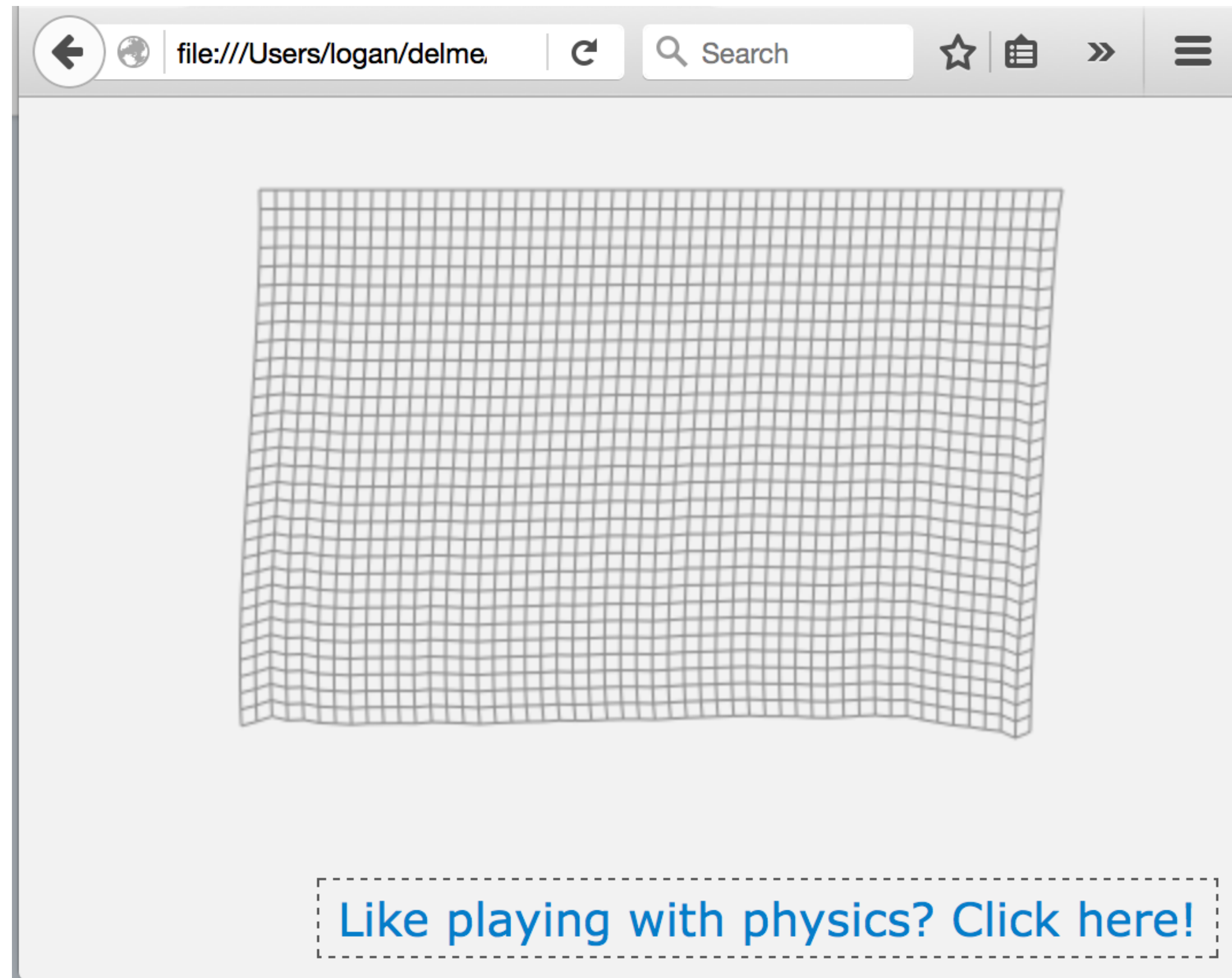
x16

RoBob Ross - Rabideau, Ekins, Rao, Jones - EECS 467 - Winter 2017

Let's start with something fun

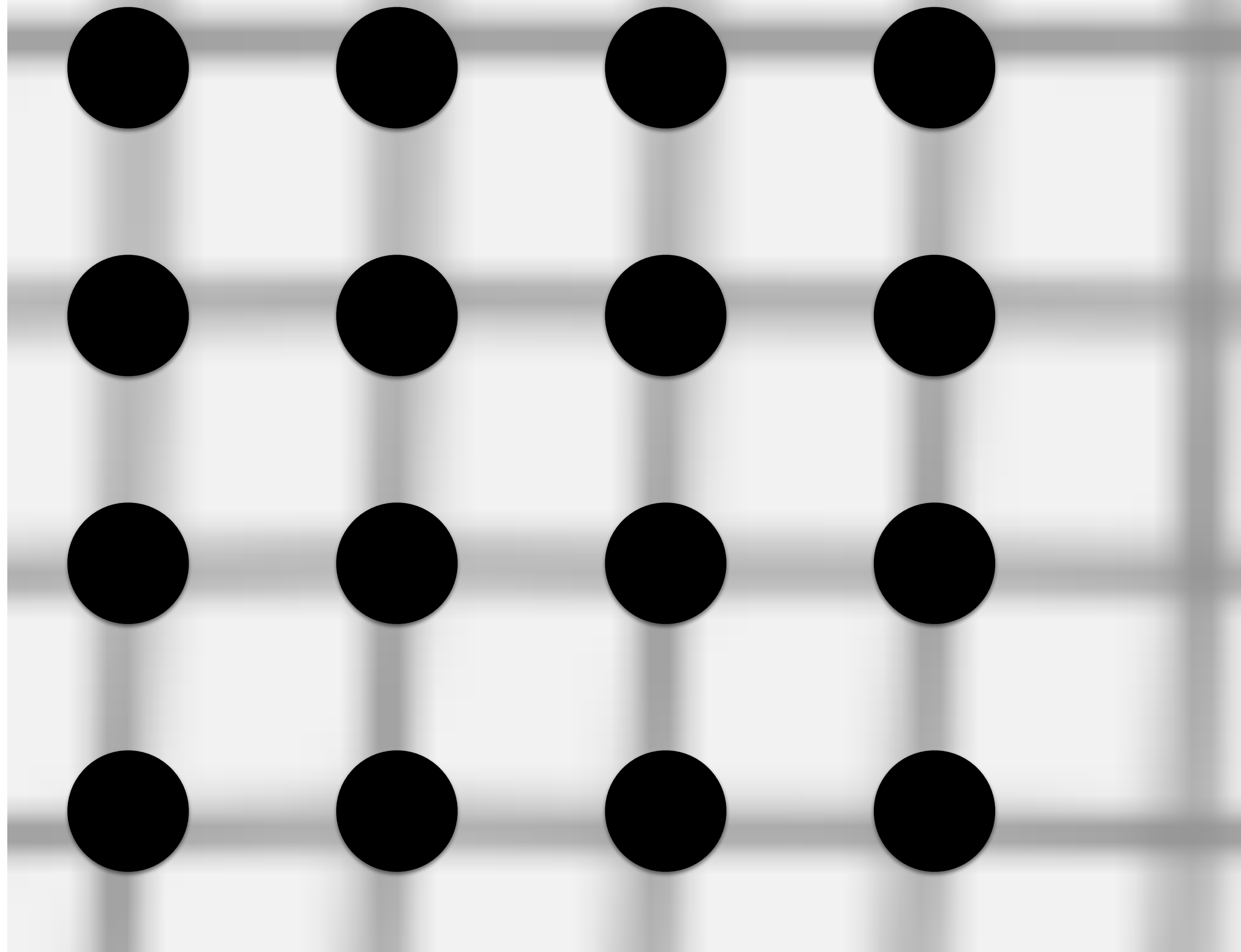


Verlet Simulated Mass-Spring Cloth

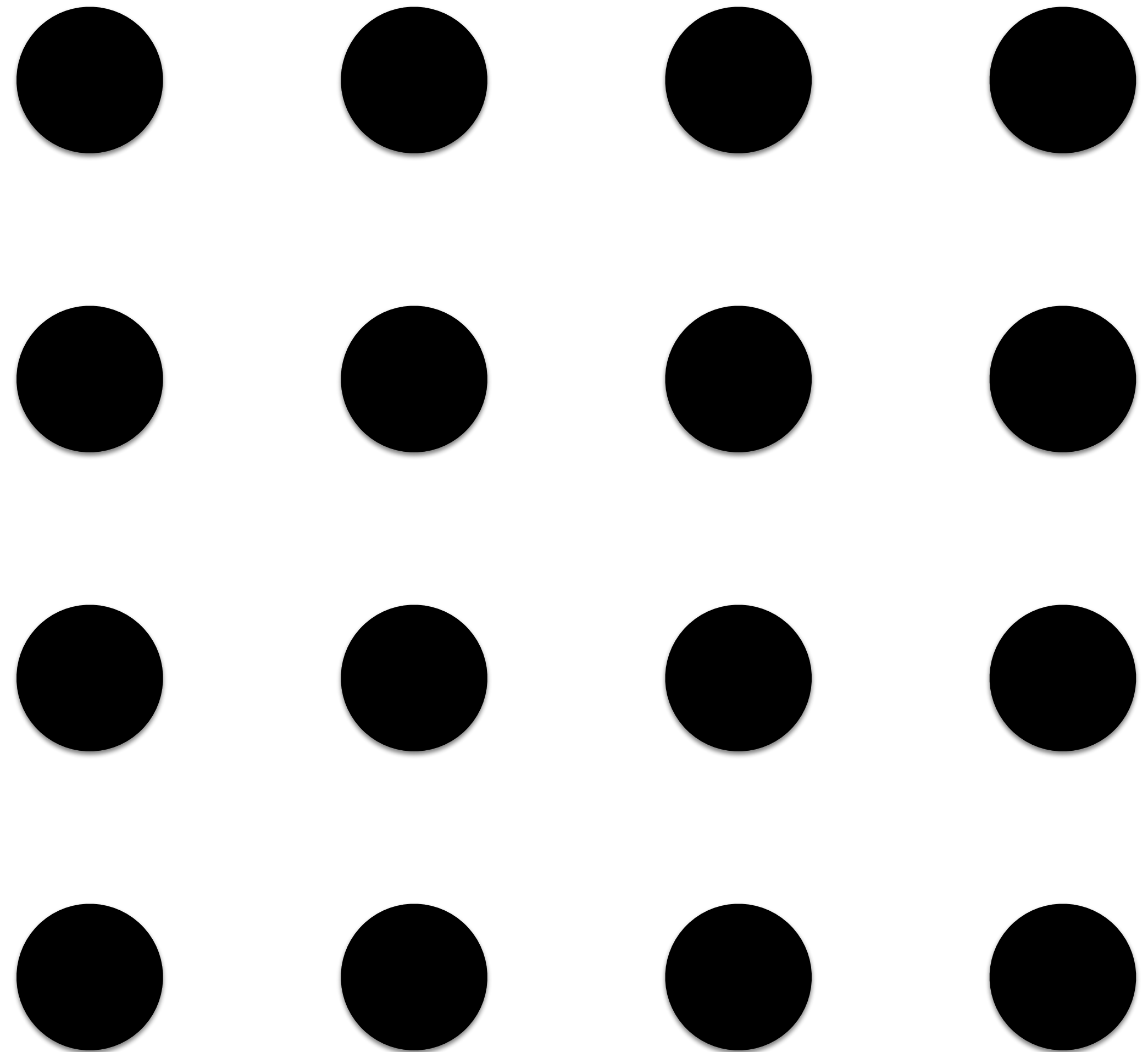


- Cloth represented as a network of point-mass nodes
- Neighboring nodes connected by high stiffness springs

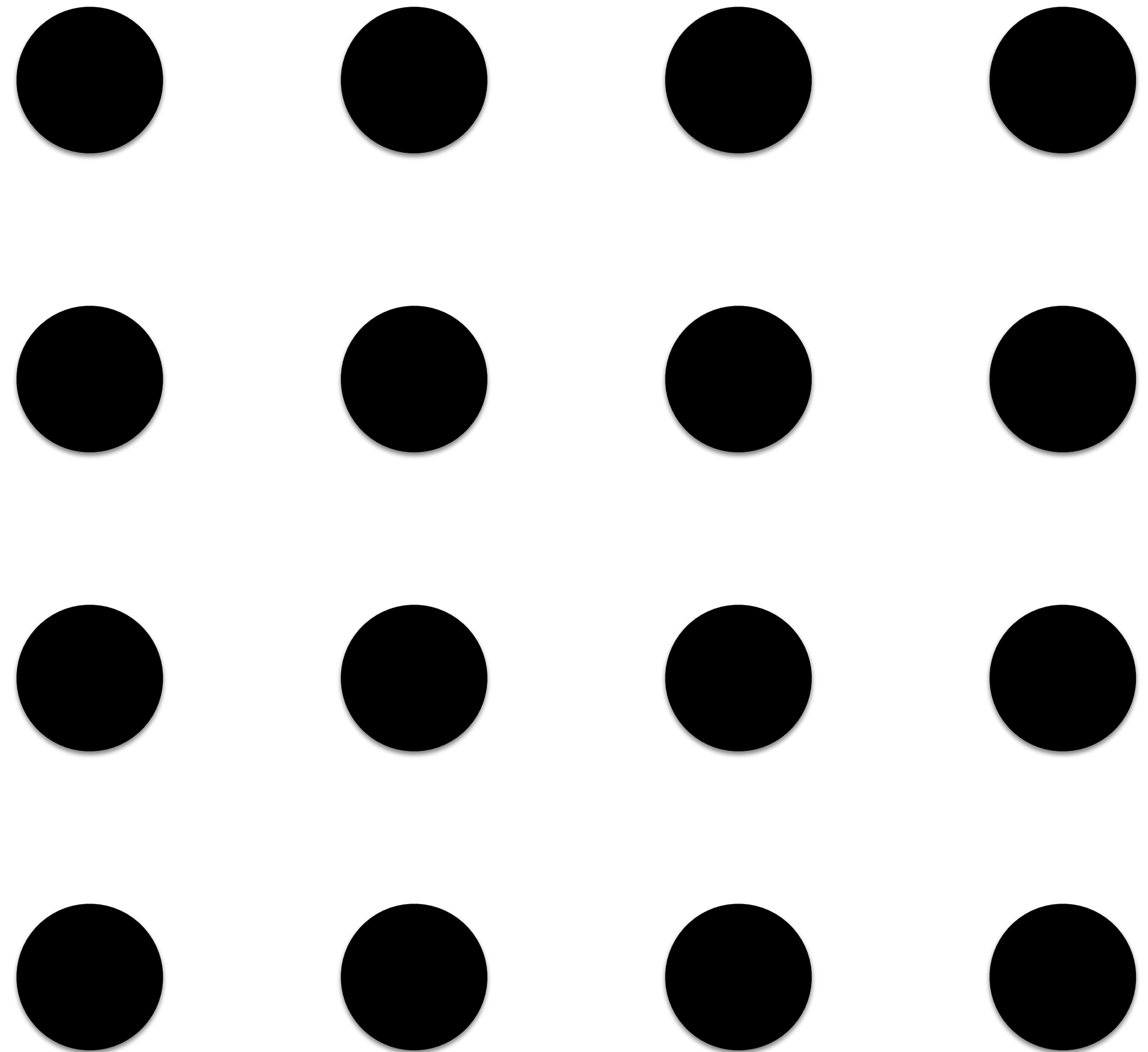
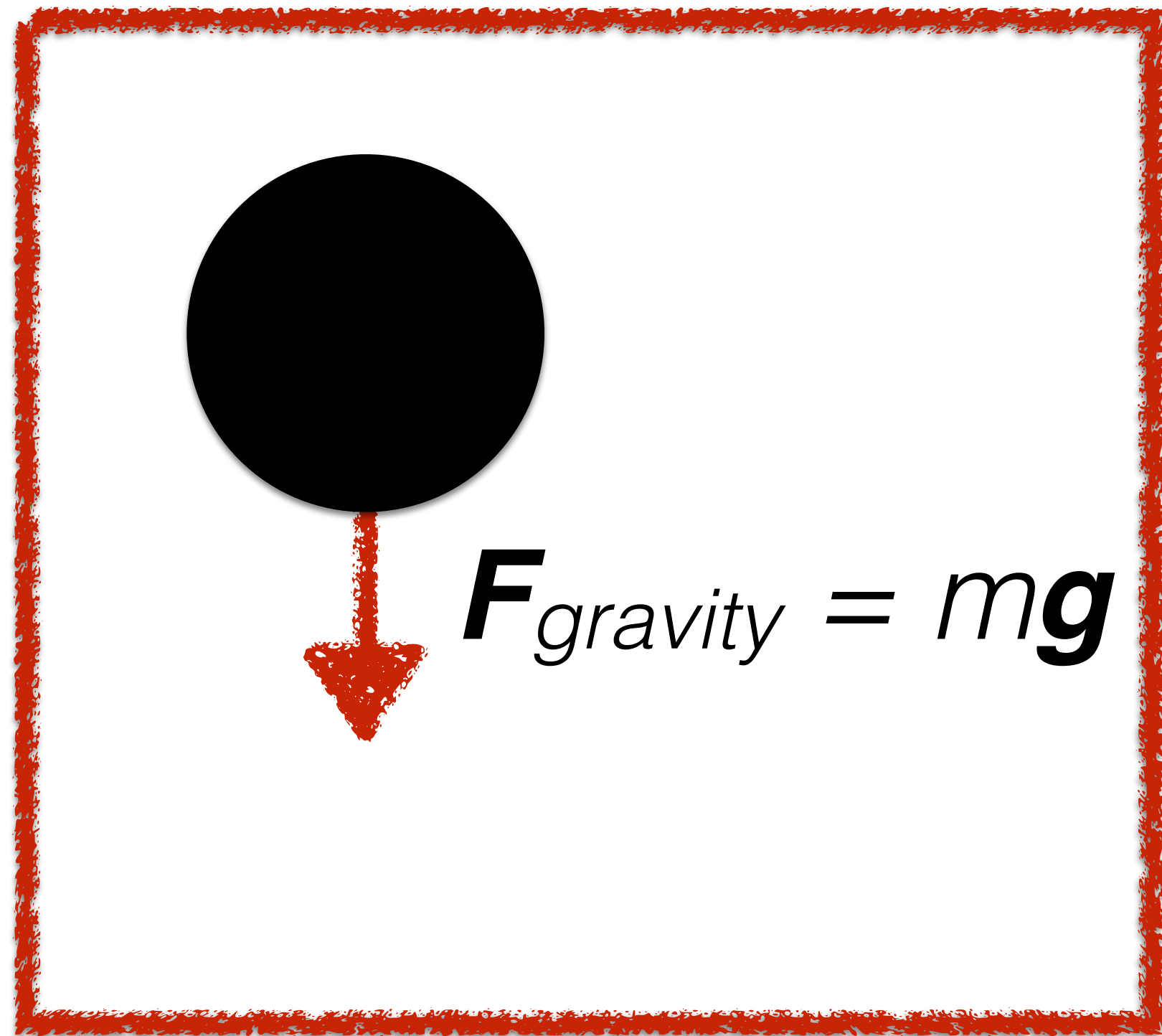
Start with point-mass particles



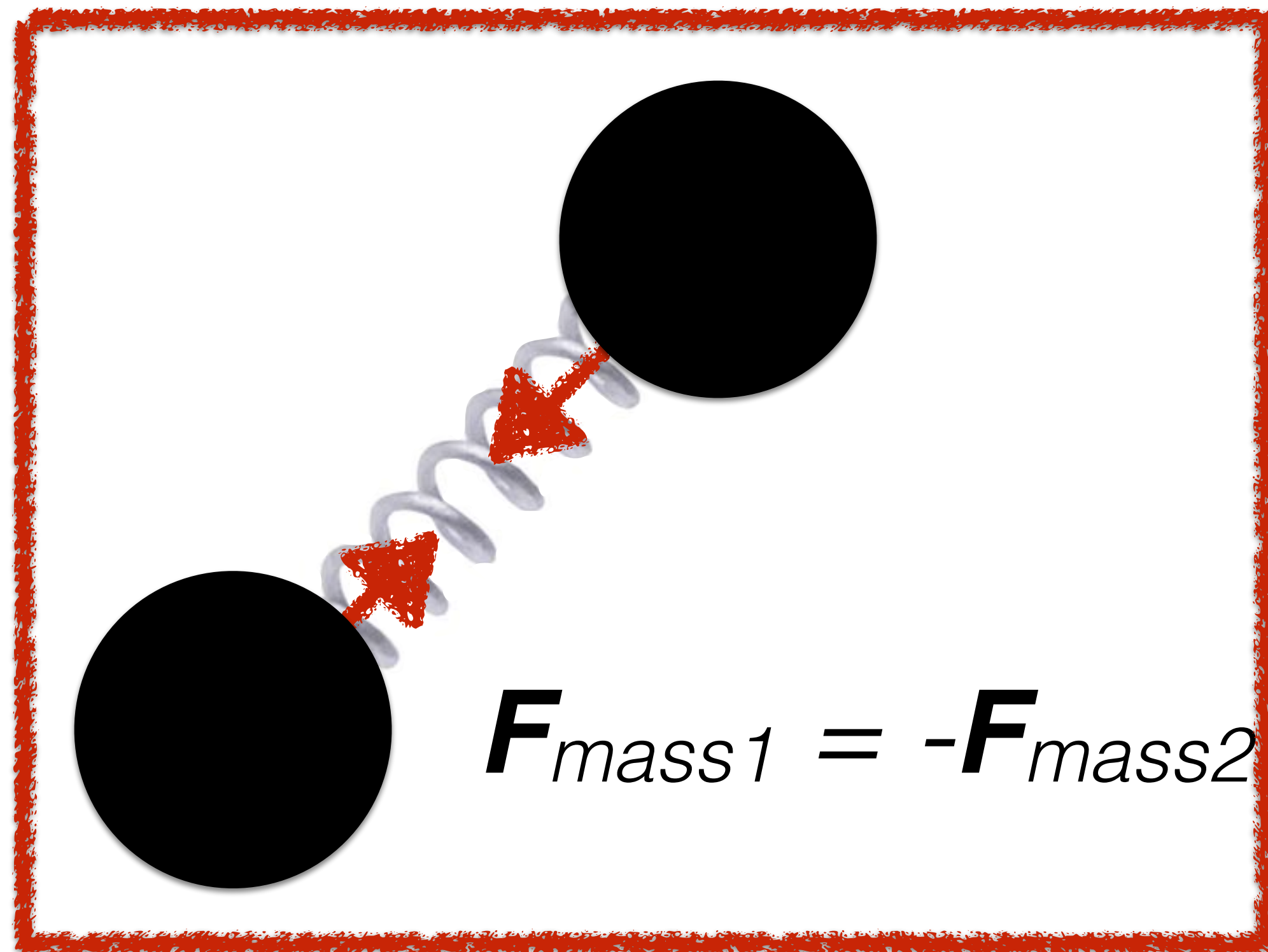
Start with point-mass particles



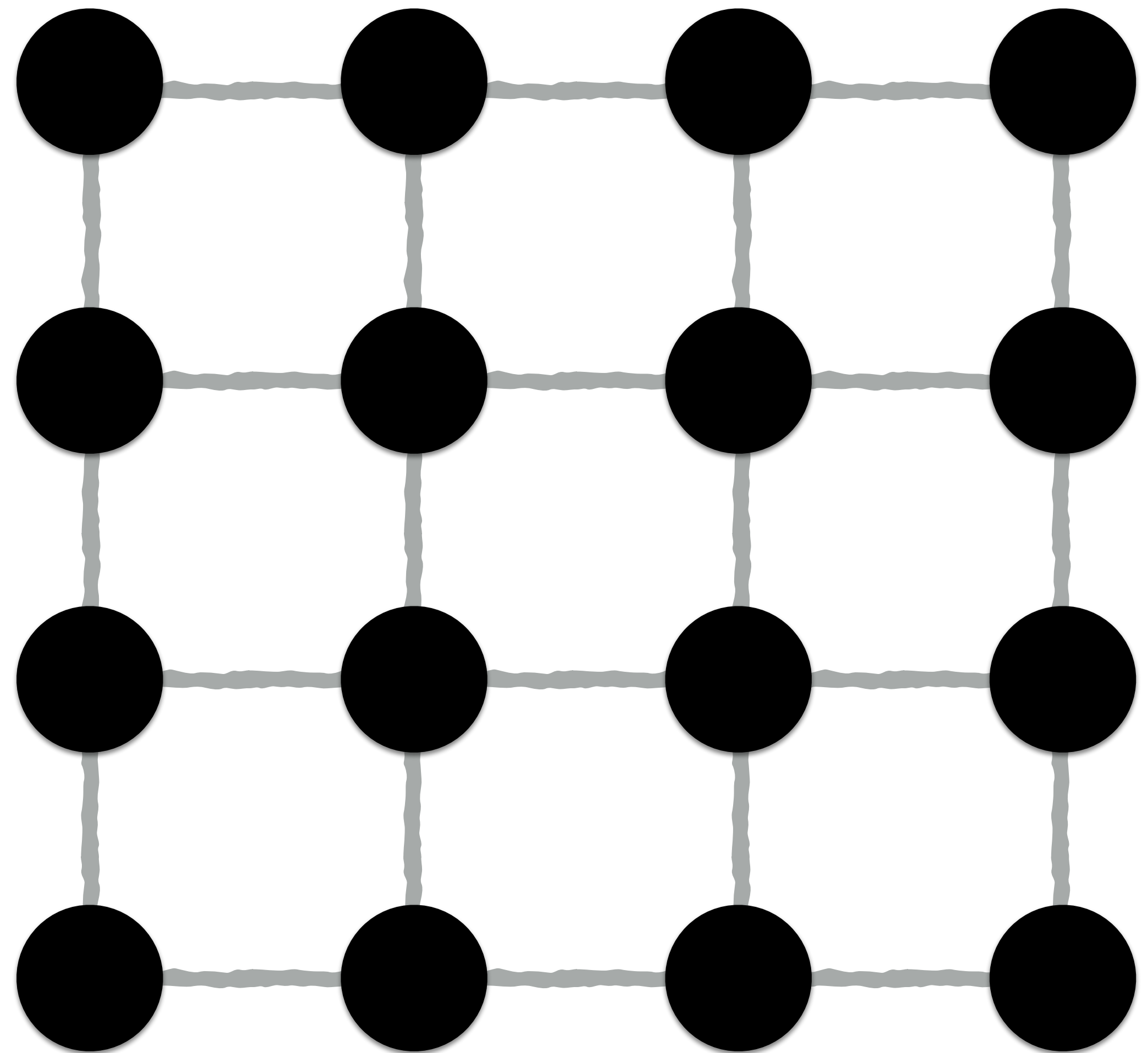
Apply gravity



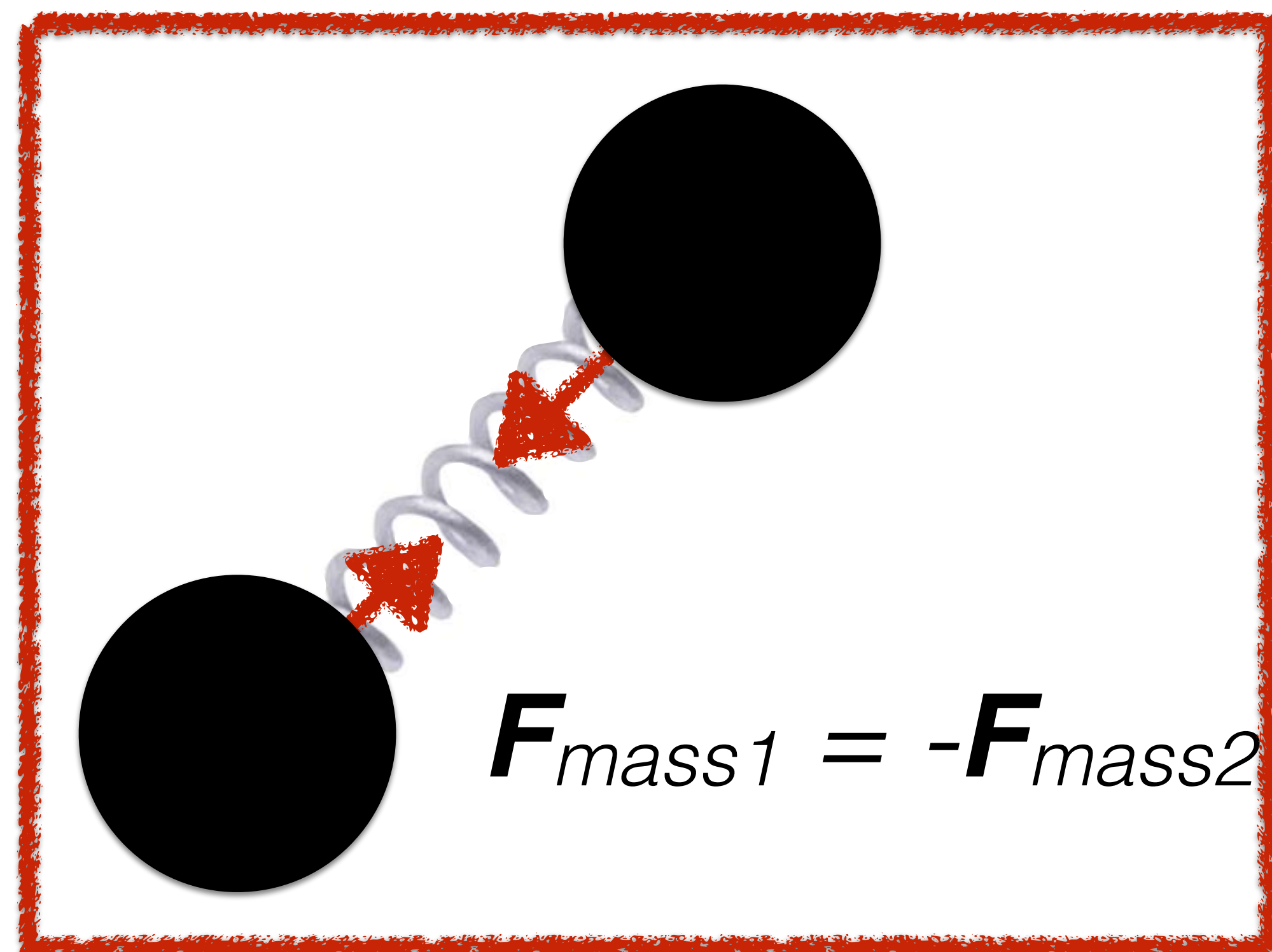
Enforce spring force constraints



spring with stiffness 1
and rest length r



Enforce spring force constraints



spring with stiffness 1
and rest length r

enforced through constraint relaxation
(multiple iterations of Gauss-Seidel)

$$d_1 = x_2^{(t)} - x_1^{(t)}$$

$$d_2 = \|d_1\|$$

Distance between particles

$$d_3 = \frac{d_2 - r}{d_2}$$

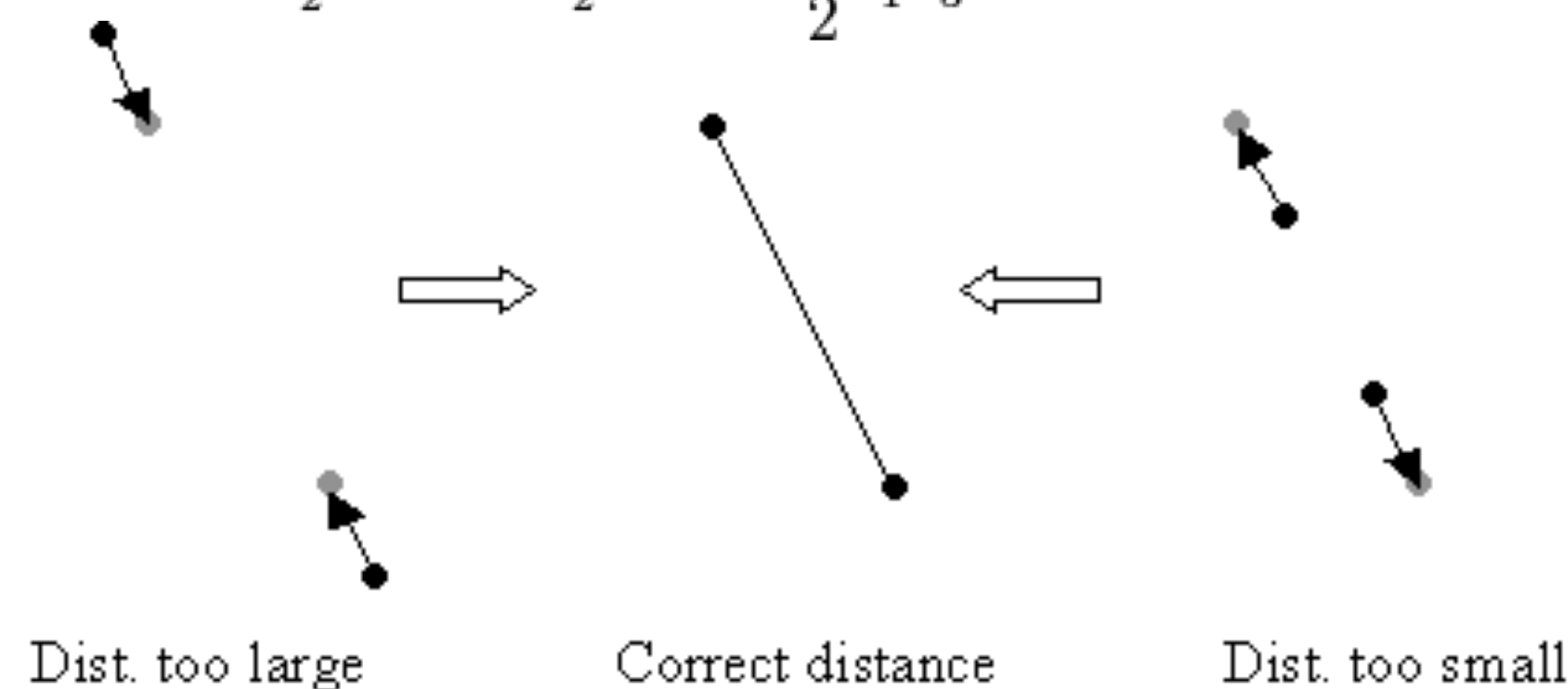
Distance from rest

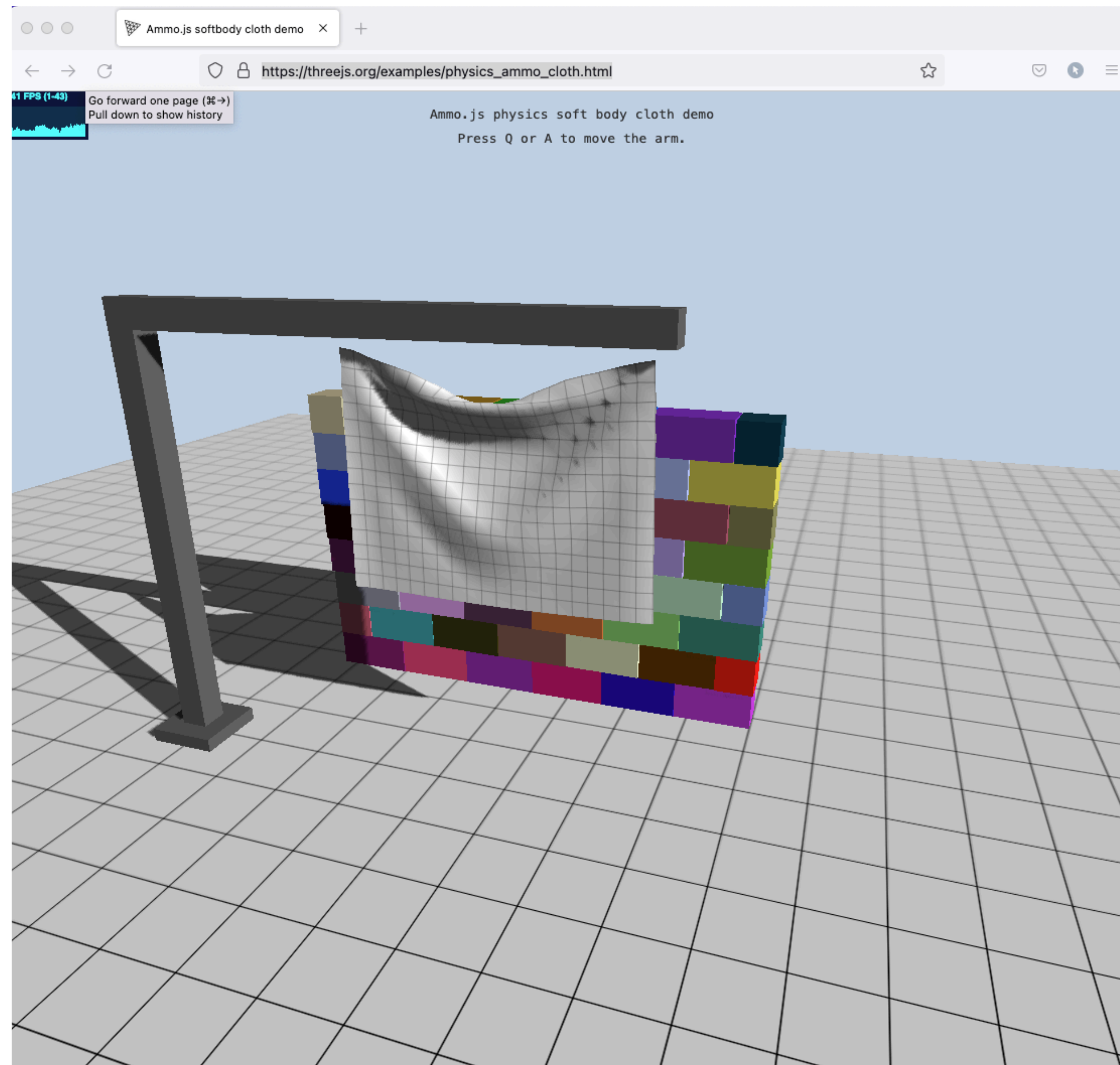
$$x_1^{(t+\Delta t)} = \tilde{x}_1^{(t+\Delta t)} + \frac{1}{2}d_1d_3$$

Correction towards

$$x_2^{(t+\Delta t)} = \tilde{x}_2^{(t+\Delta t)} - \frac{1}{2}d_1d_3$$

rest length



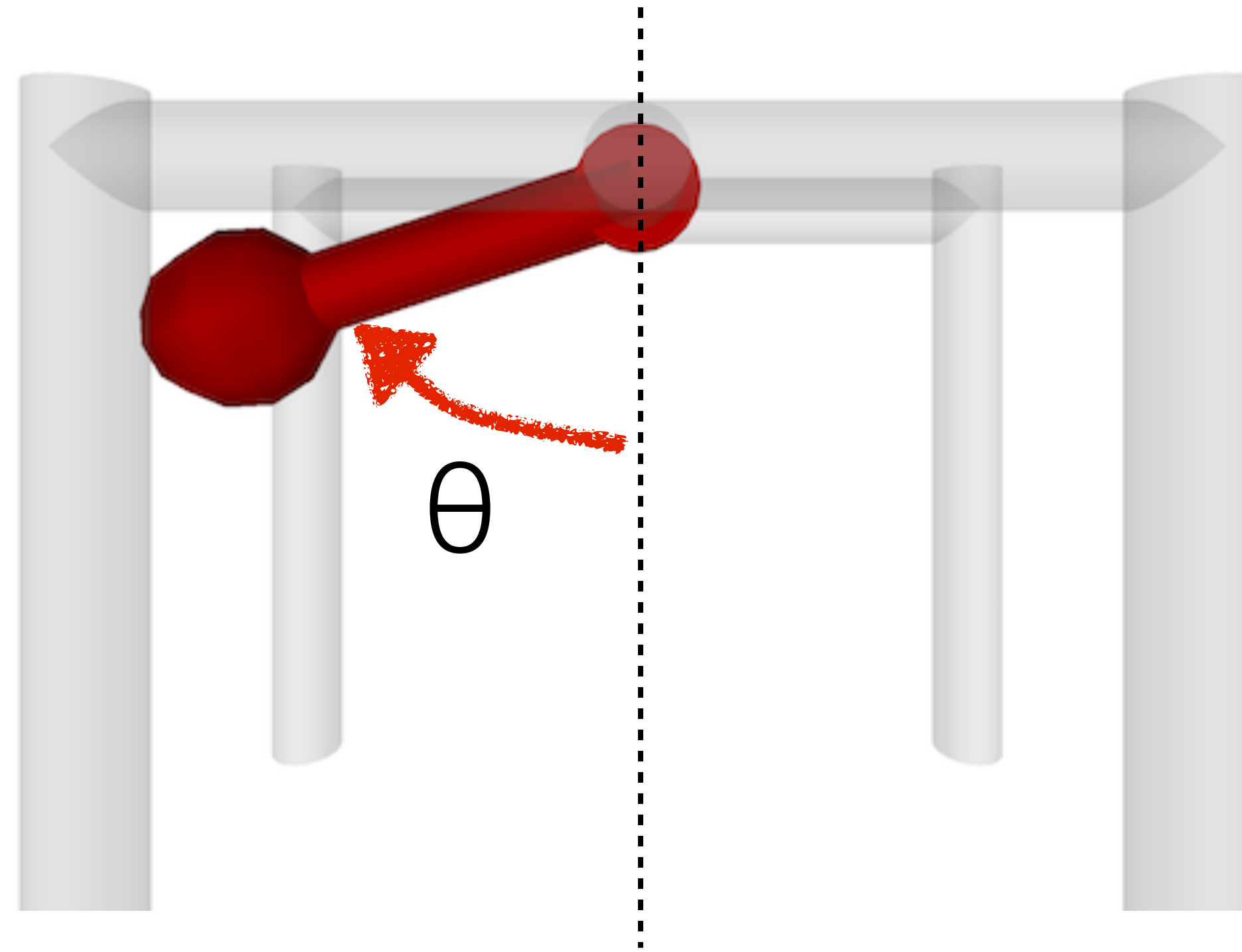


https://threejs.org/examples/physics_ammo_cloth.html

Pendulum Revisited

Pendulum Revisited

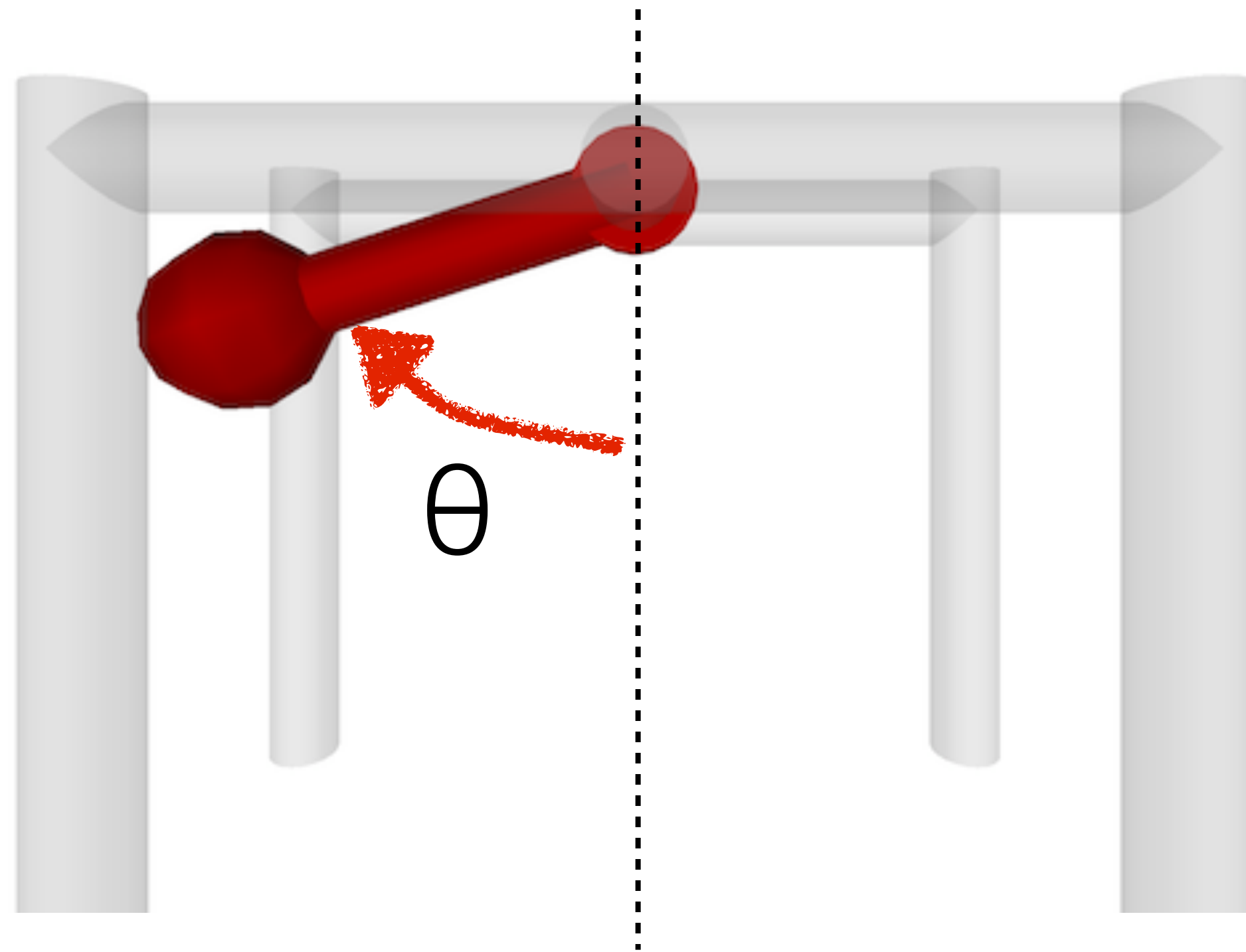
Generalized coordinates



Simulate the motion of DOFs
(pendulum angle) directly

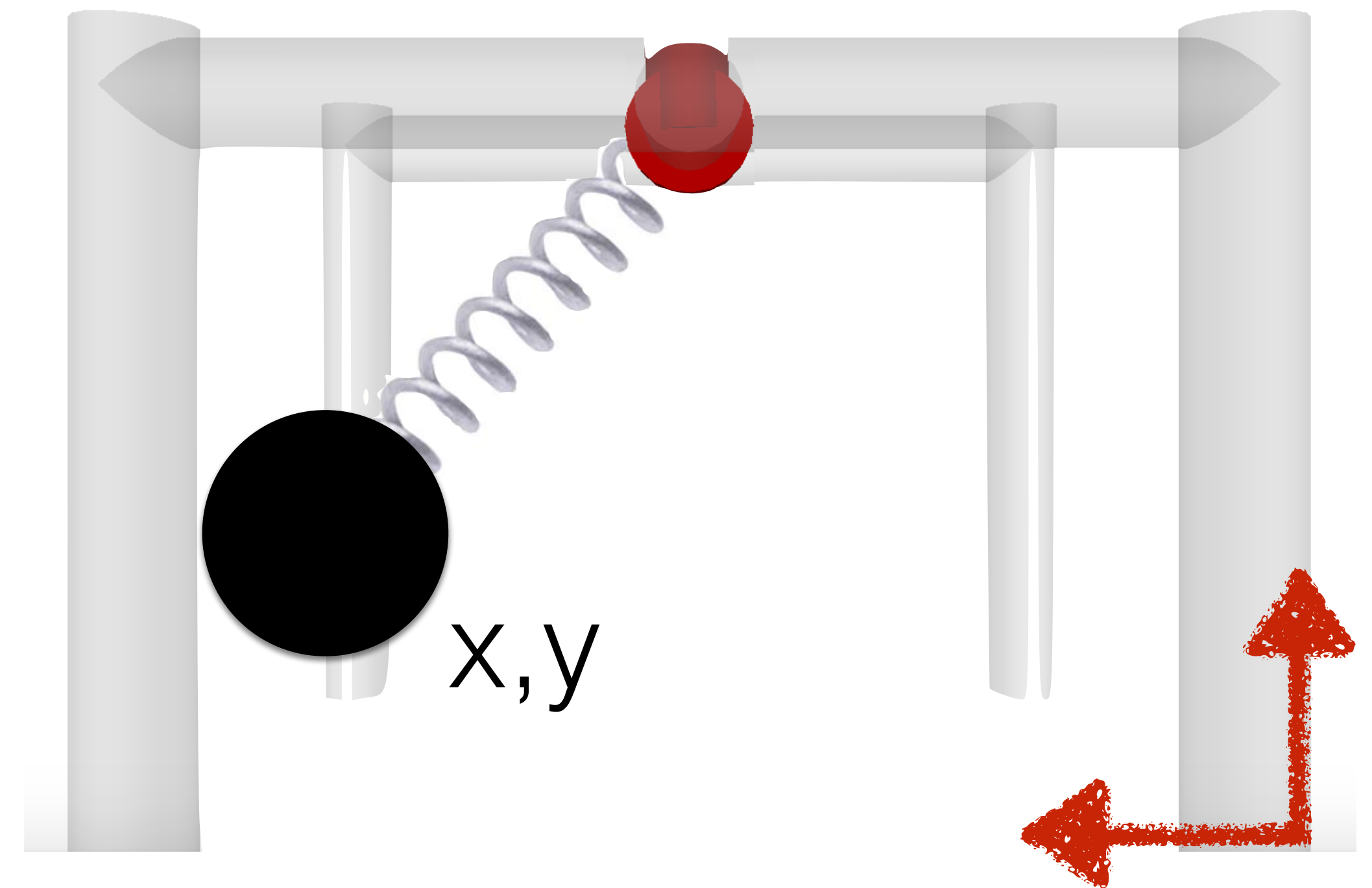
Why use generalized coordinates?

Generalized coordinates



Simulate the motion of DOFs
(pendulum angle) directly

Maximal coordinates



Simulate the motion of individual
bodies (pendulum bob location)
corrected by constraints (spring)

Why use generalized coordinates?

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

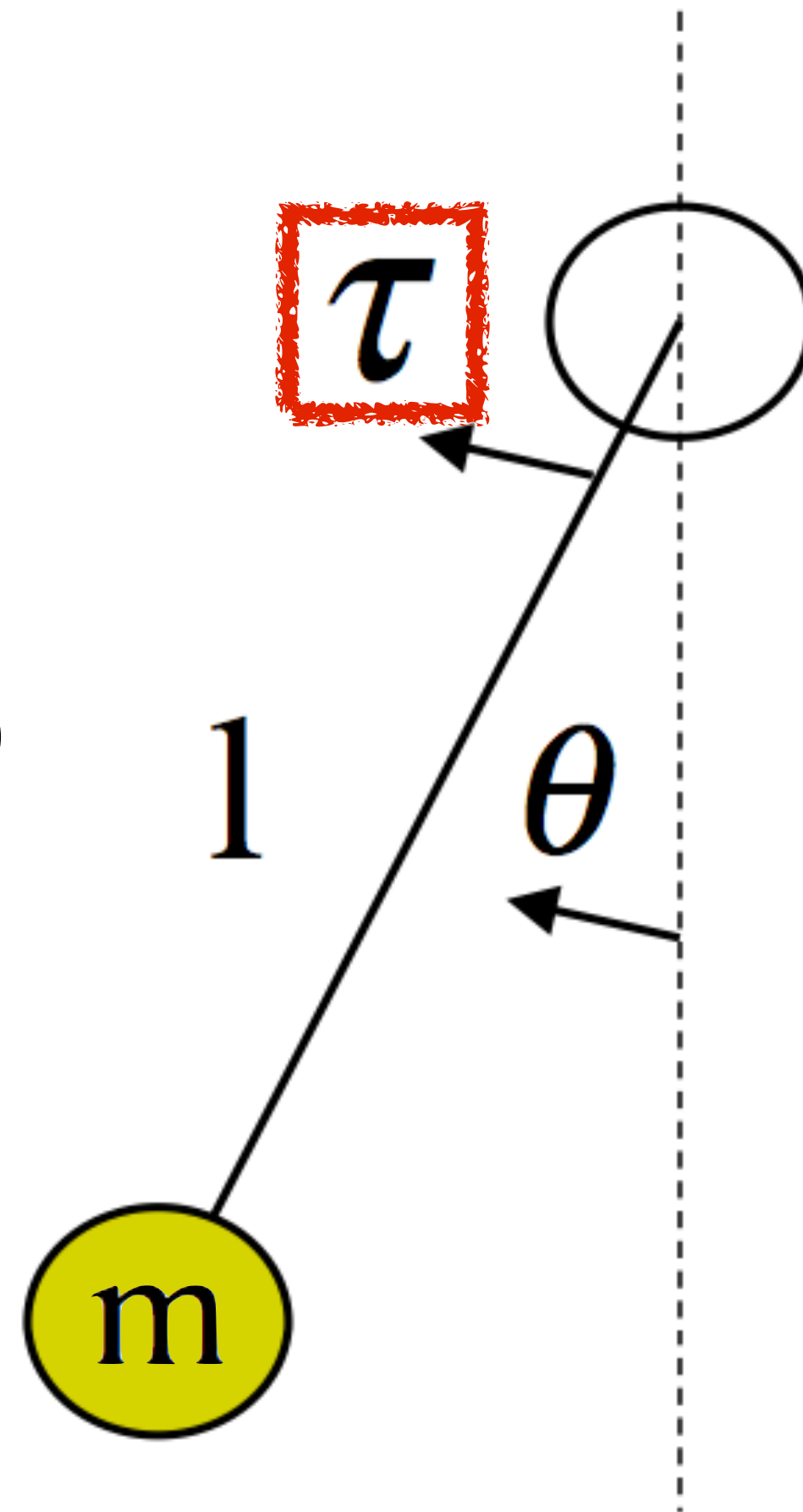
with Parallel Axis Theorem ($I=mr^2$)

$$\ddot{\theta} = -\frac{g}{l} \sin(\theta) + \frac{\tau}{ml^2}$$

Numerical integration over time

$$\theta_{t+dt} = \theta_t + \dot{\theta}_t dt$$

$$\dot{\theta}_{t+dt} = \dot{\theta}_t + \ddot{\theta}_t dt$$



Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Motor force expressed via DOFs

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

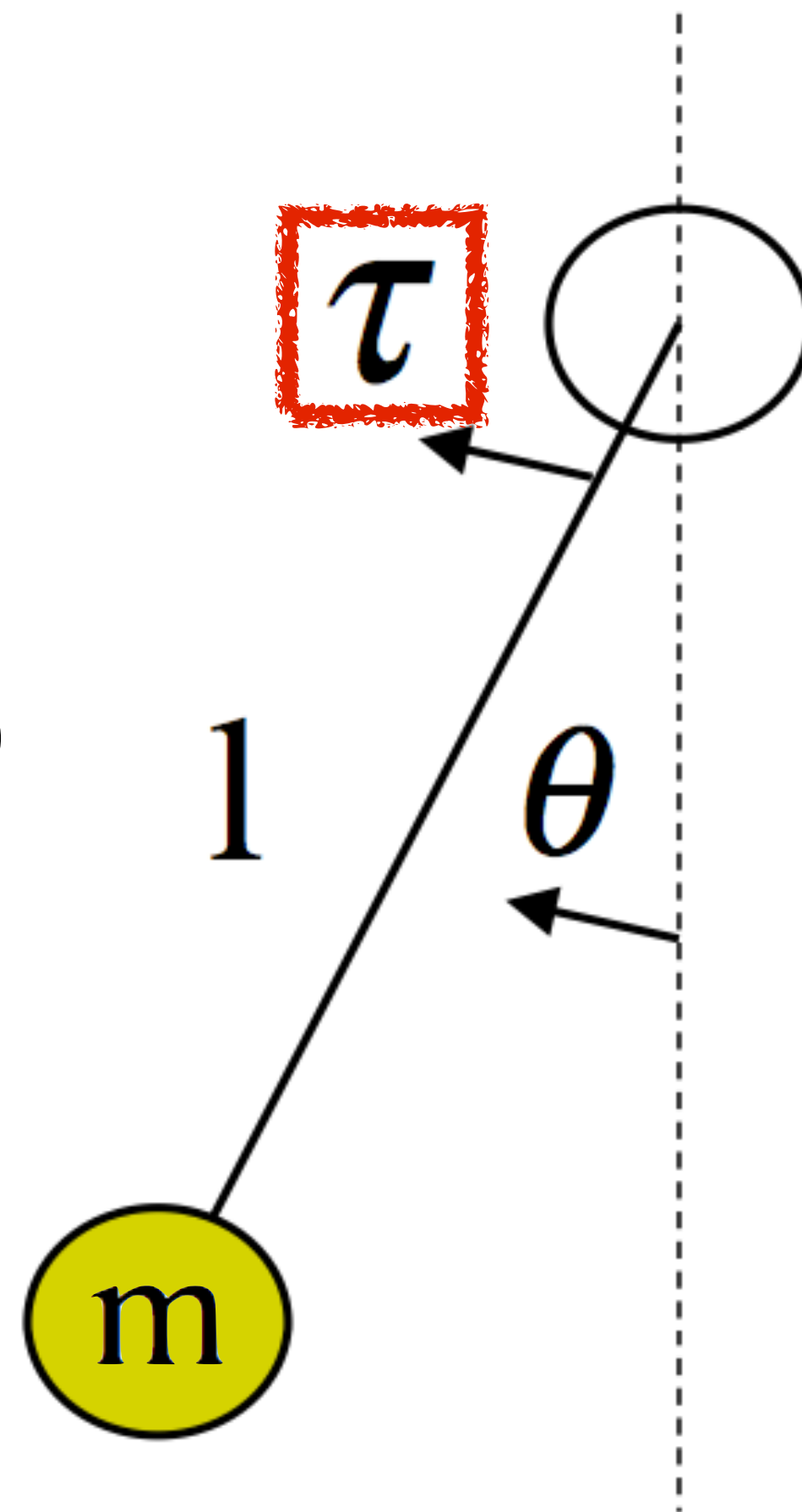
with Parallel Axis Theorem ($I=mr^2$)

$$\ddot{\theta} = -\frac{g}{l} \sin(\theta) + \frac{\tau}{ml^2}$$

Numerical integration over time

$$\theta_{t+dt} = \theta_t + \dot{\theta}_t dt$$

$$\dot{\theta}_{t+dt} = \dot{\theta}_t + \ddot{\theta}_t dt$$

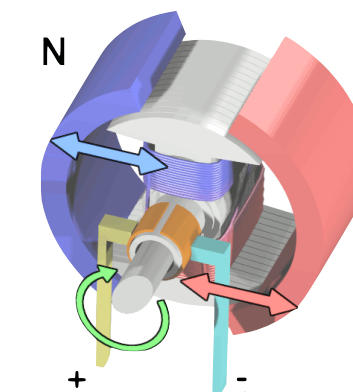
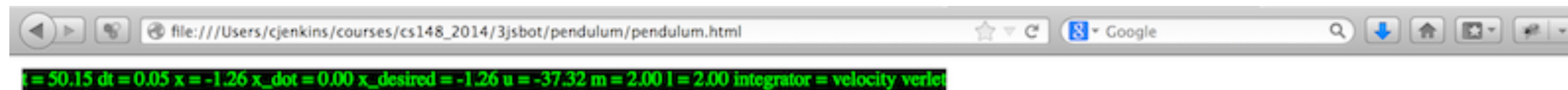


Motor produces torque
(angular force)

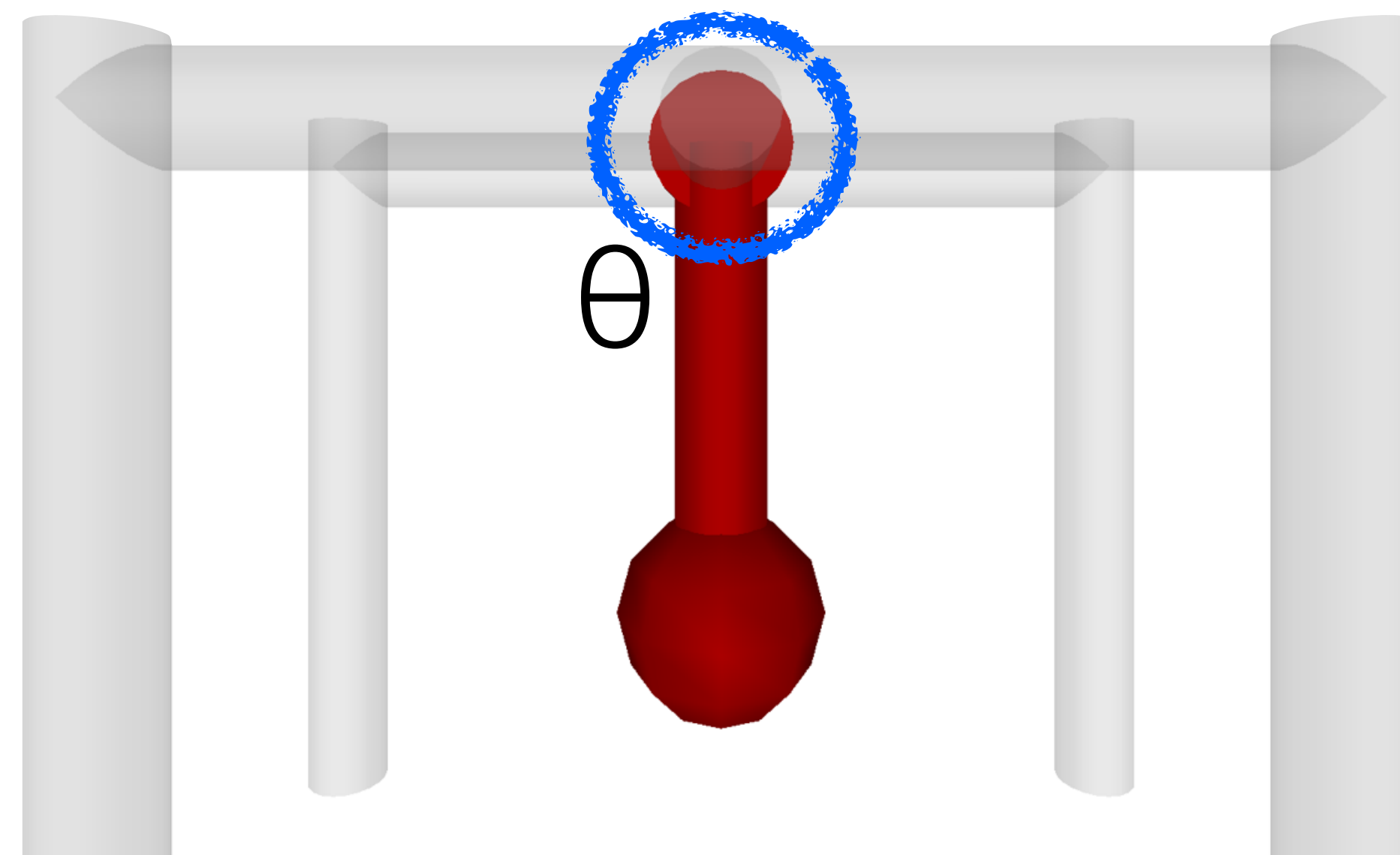
Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

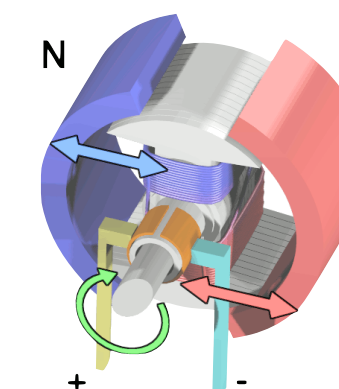
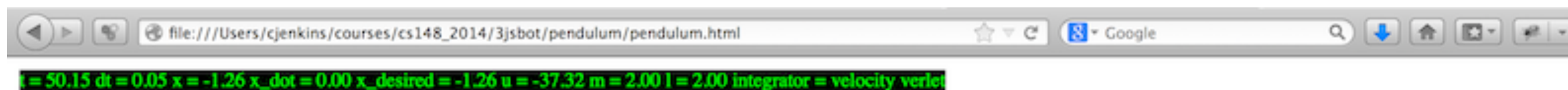
Motor force expressed via DOFs



assume 1DOF motor at pendulum axis



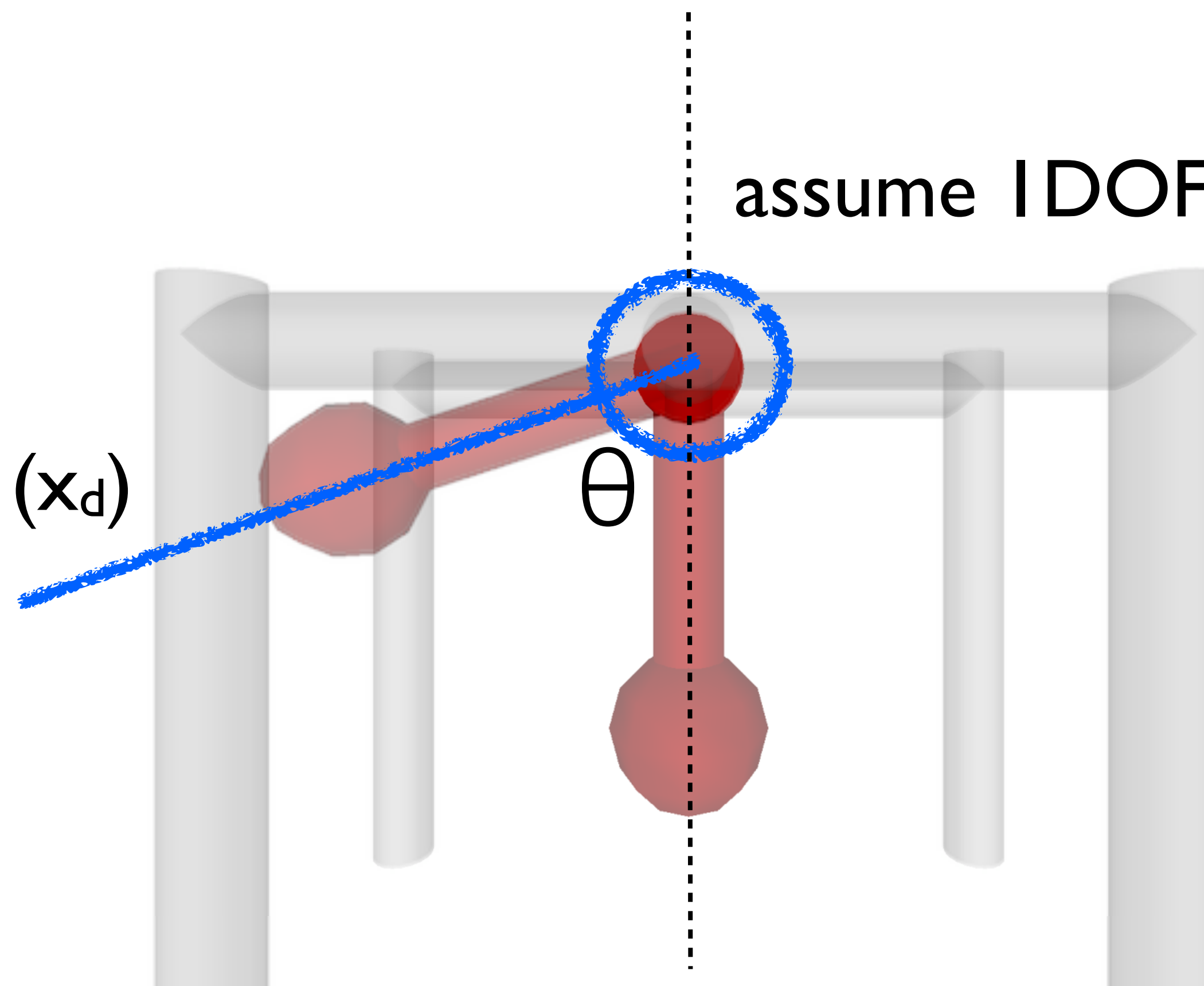
Can we control the pendulum?



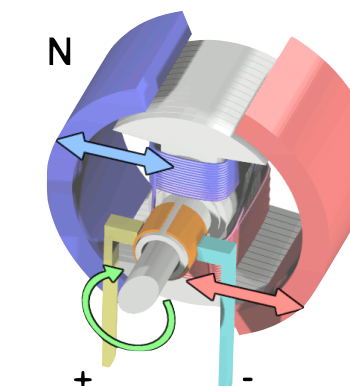
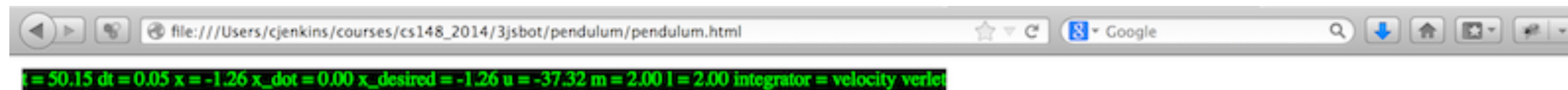
assume 1DOF motor at pendulum axis

Setpoint

maintain desired angle (x_d)



Can we control the pendulum?



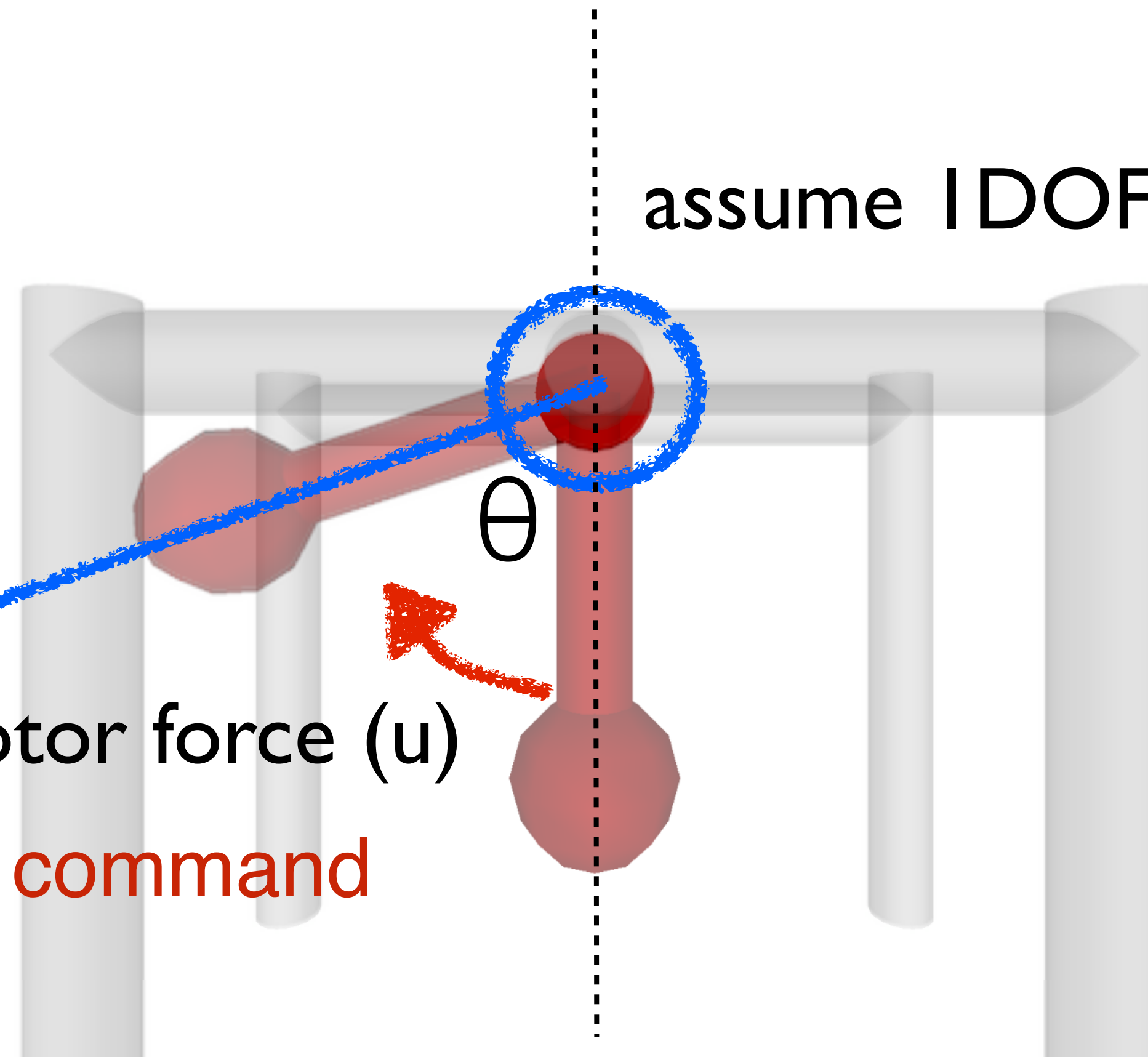
assume 1DOF motor at pendulum axis

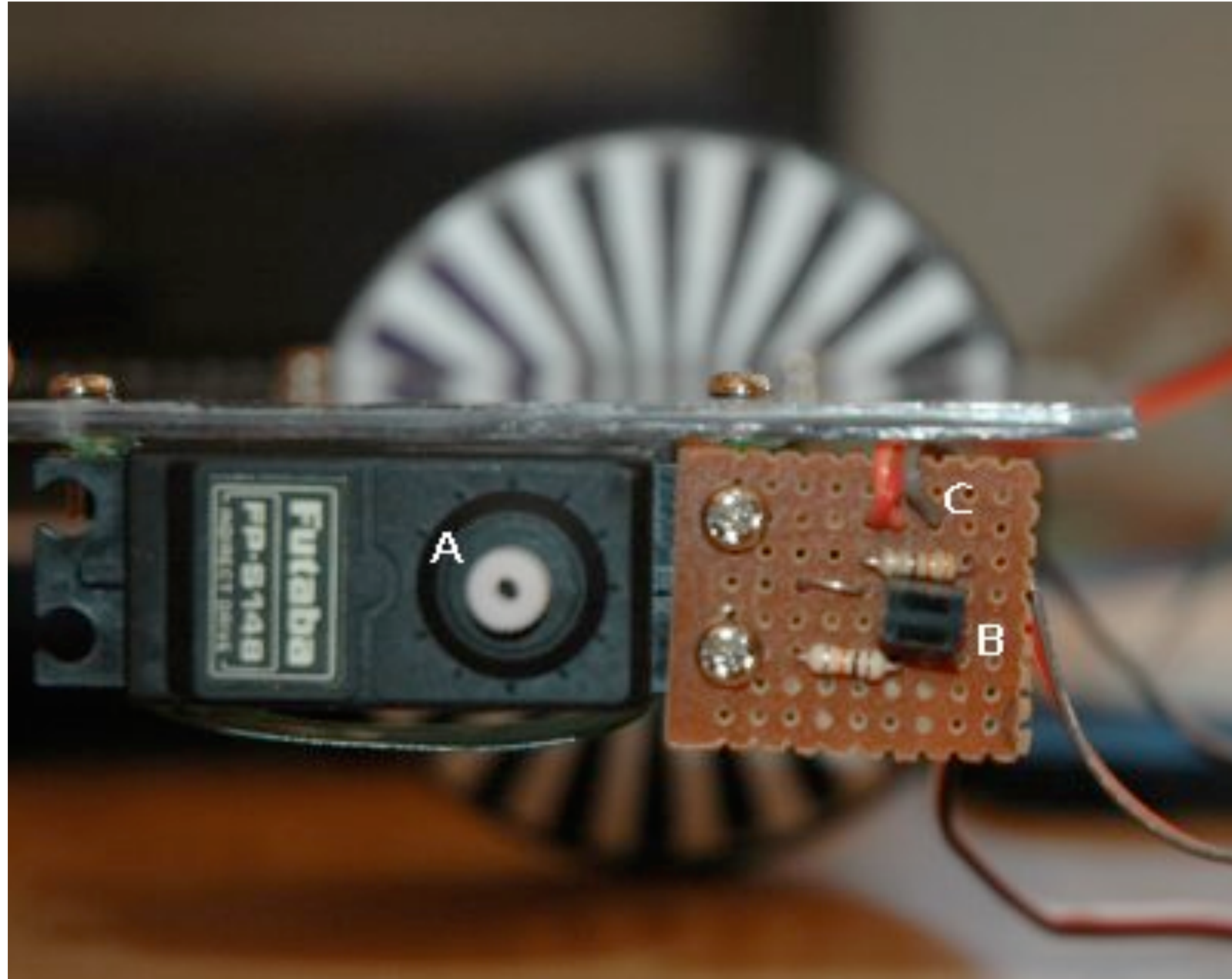
Setpoint

maintain desired angle (x_d)

apply motor force (u)

Motor command

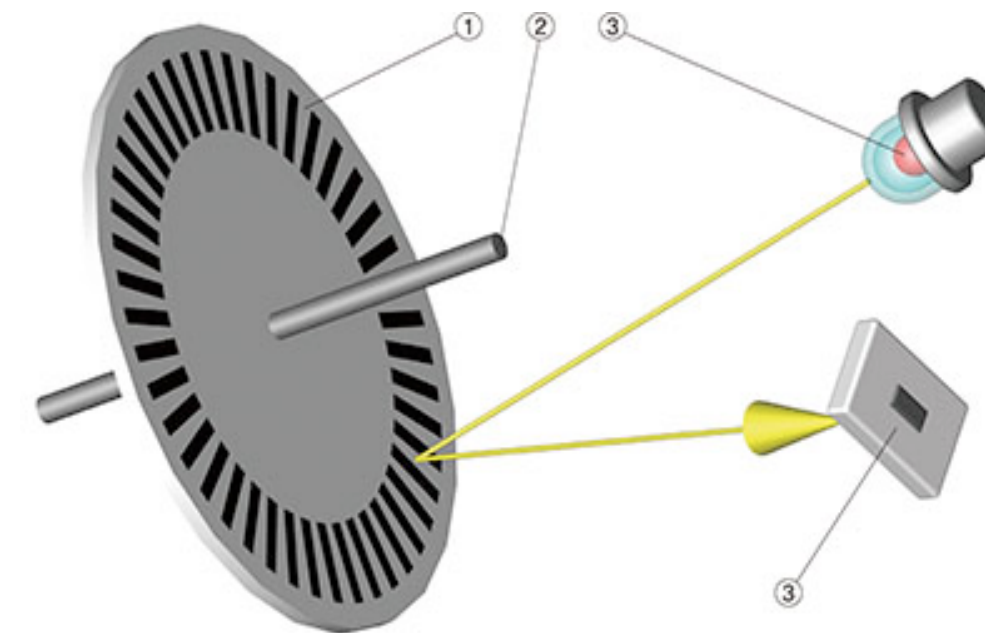
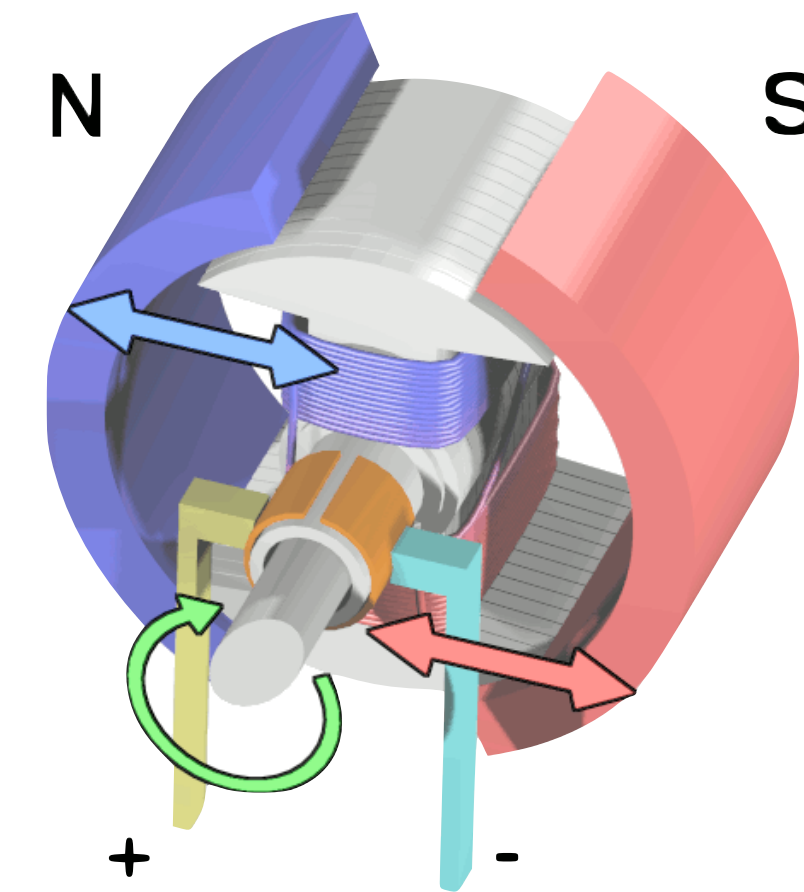




Servos have:

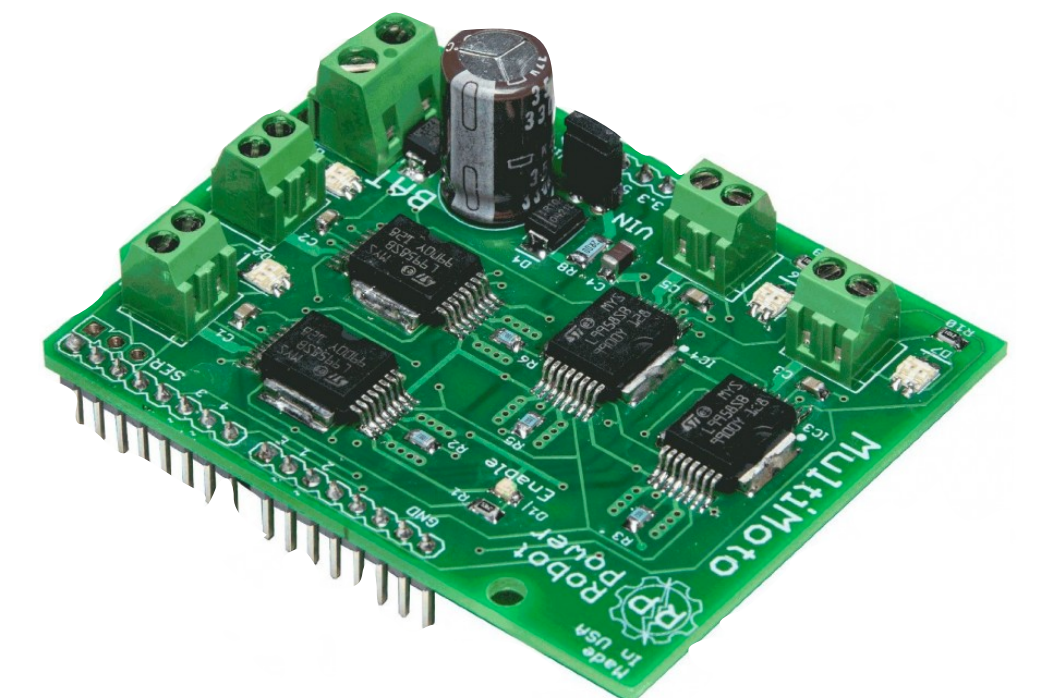
- actuators to produce motion
- proprioception to sense pose
- controller to regulate current to motor

Actuator
(electric)



Proprioception
(optical encoder)

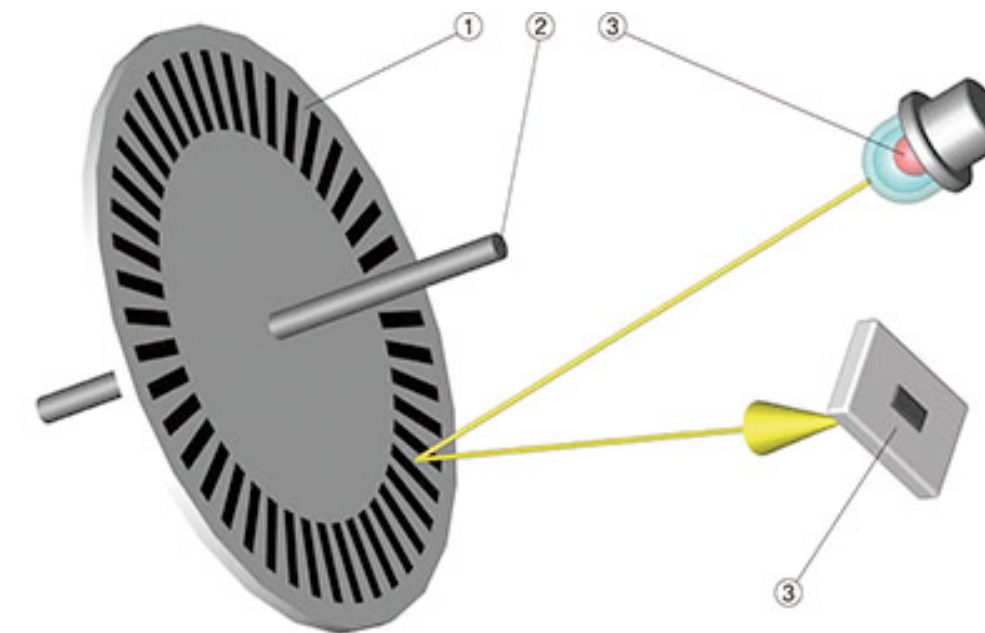
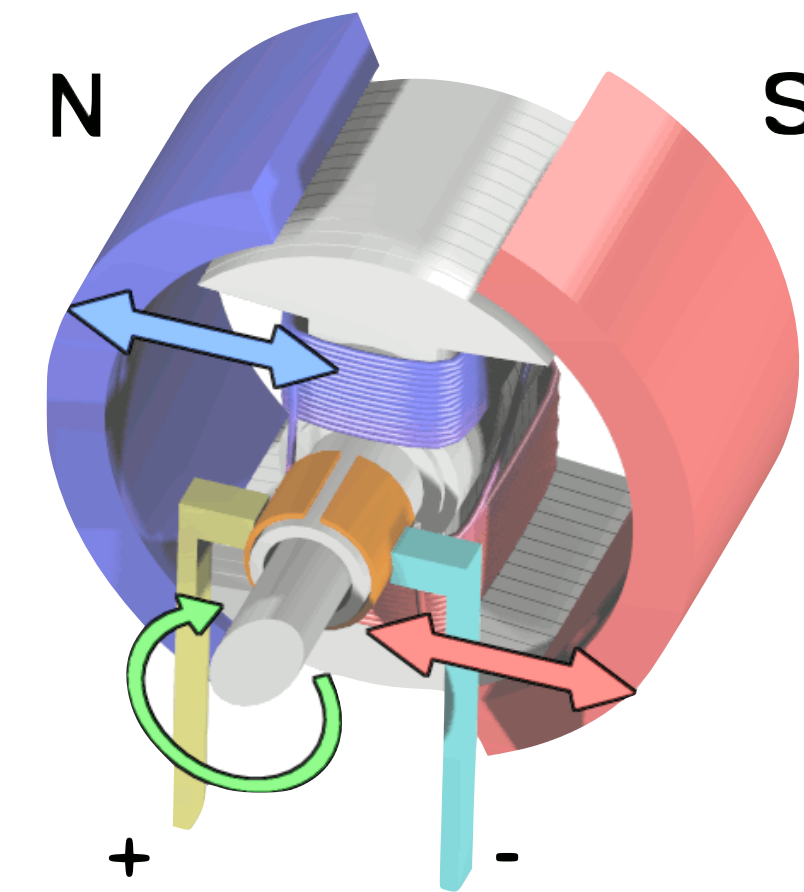
Motor Controller
(4 channel H-Bridge)





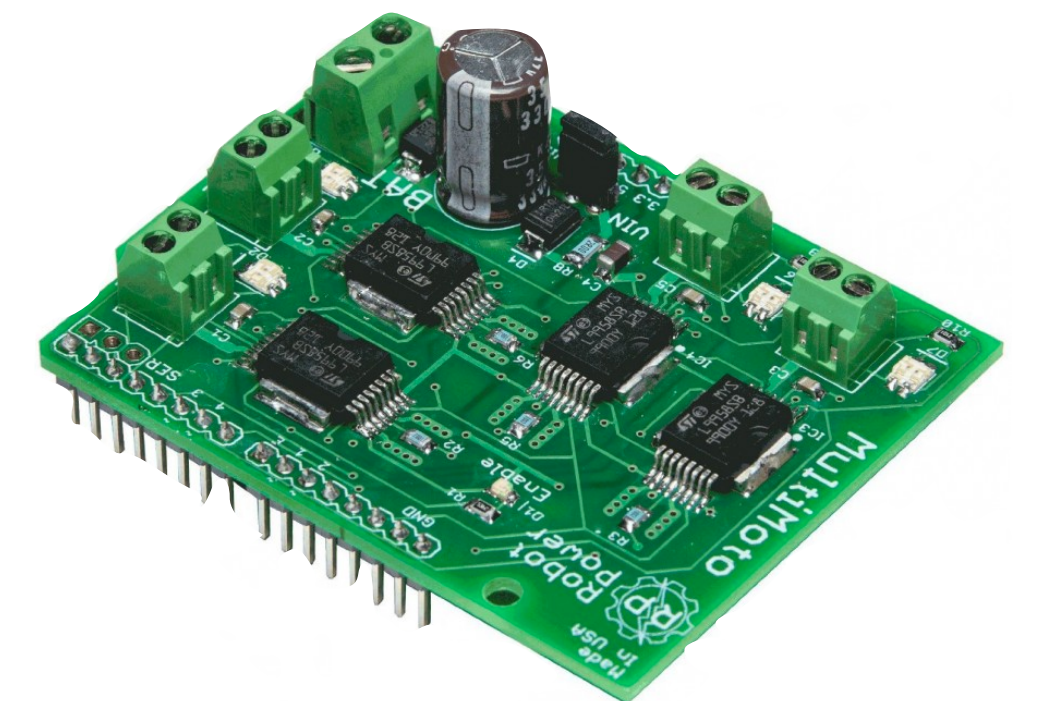
What is inside a servo?

Actuator (electric)



Proprioception (optical encoder)

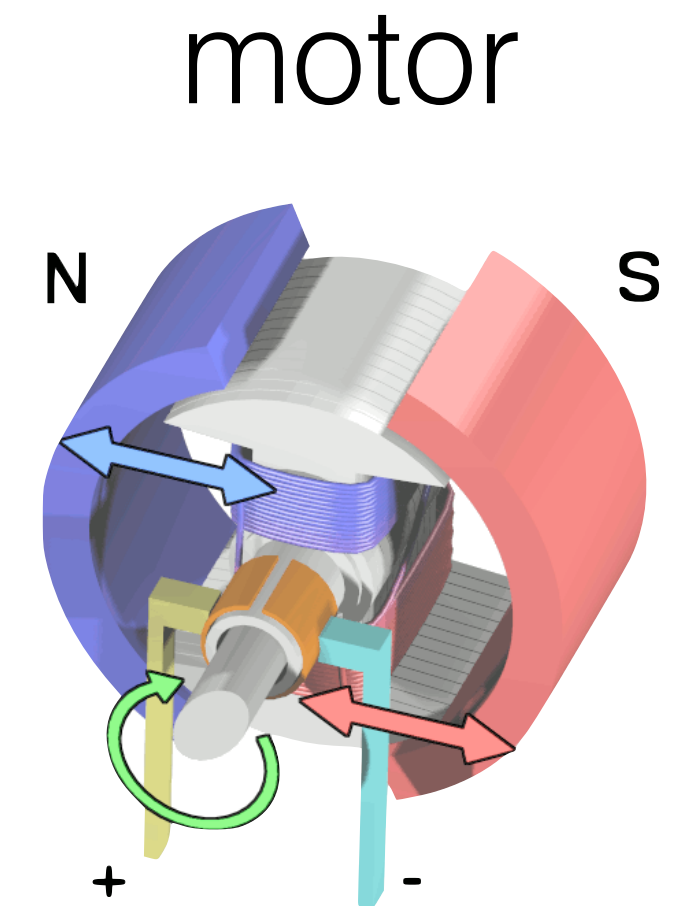
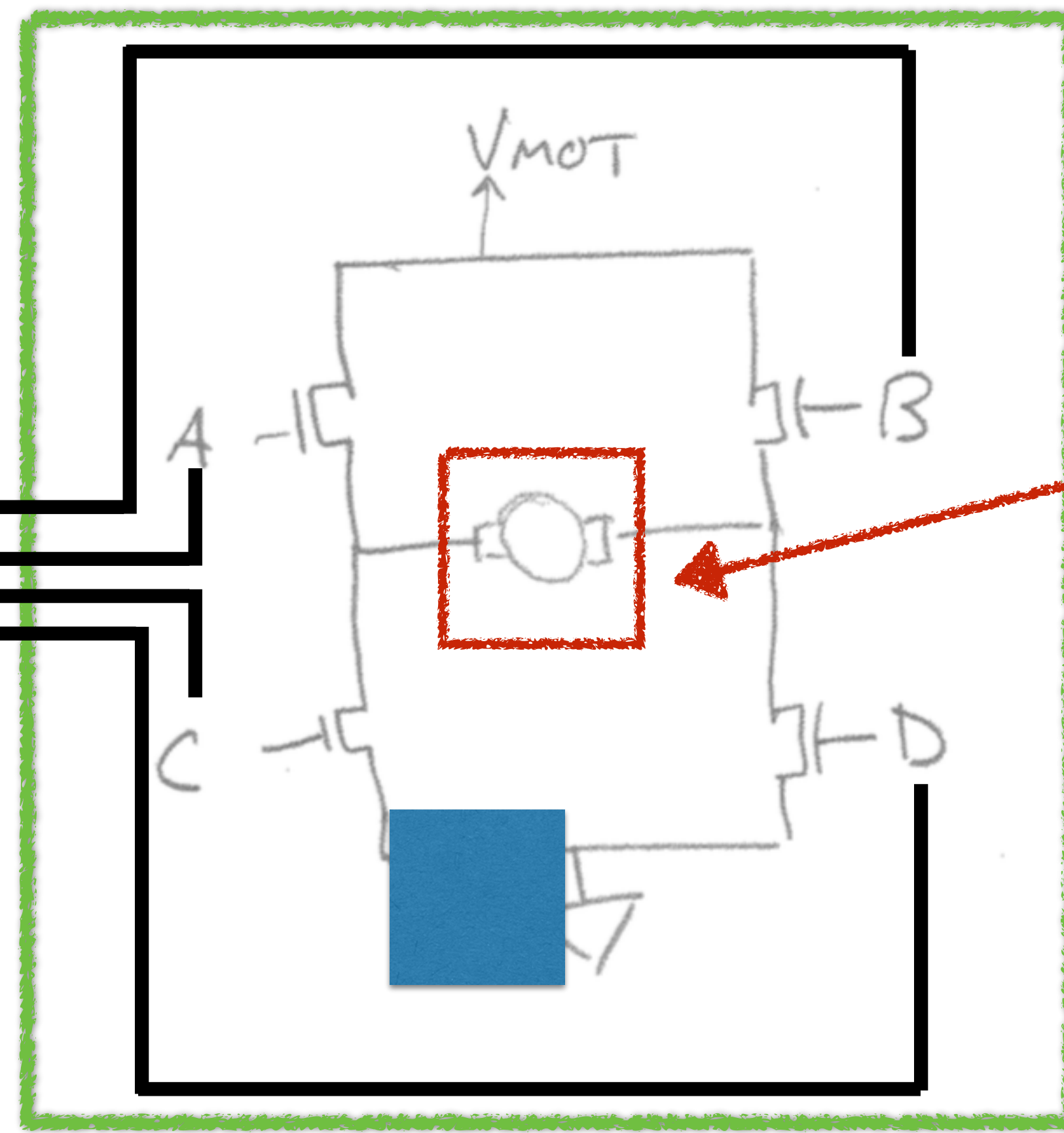
Motor Controller (4 channel H-Bridge)



What is inside a servo?

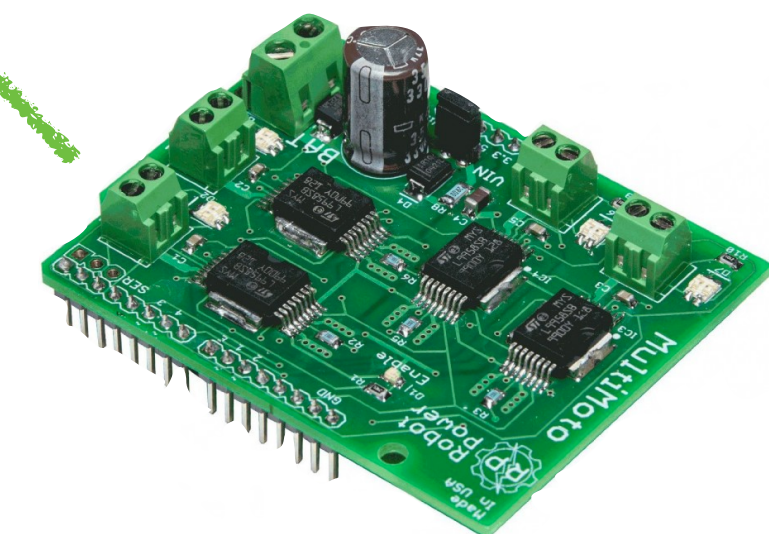
**Control
Signal**

PWM signal sent
to H-bridge
controller.



Current actuates
motor motion
about axis.

H-bridge sends
proportional
current to motor.

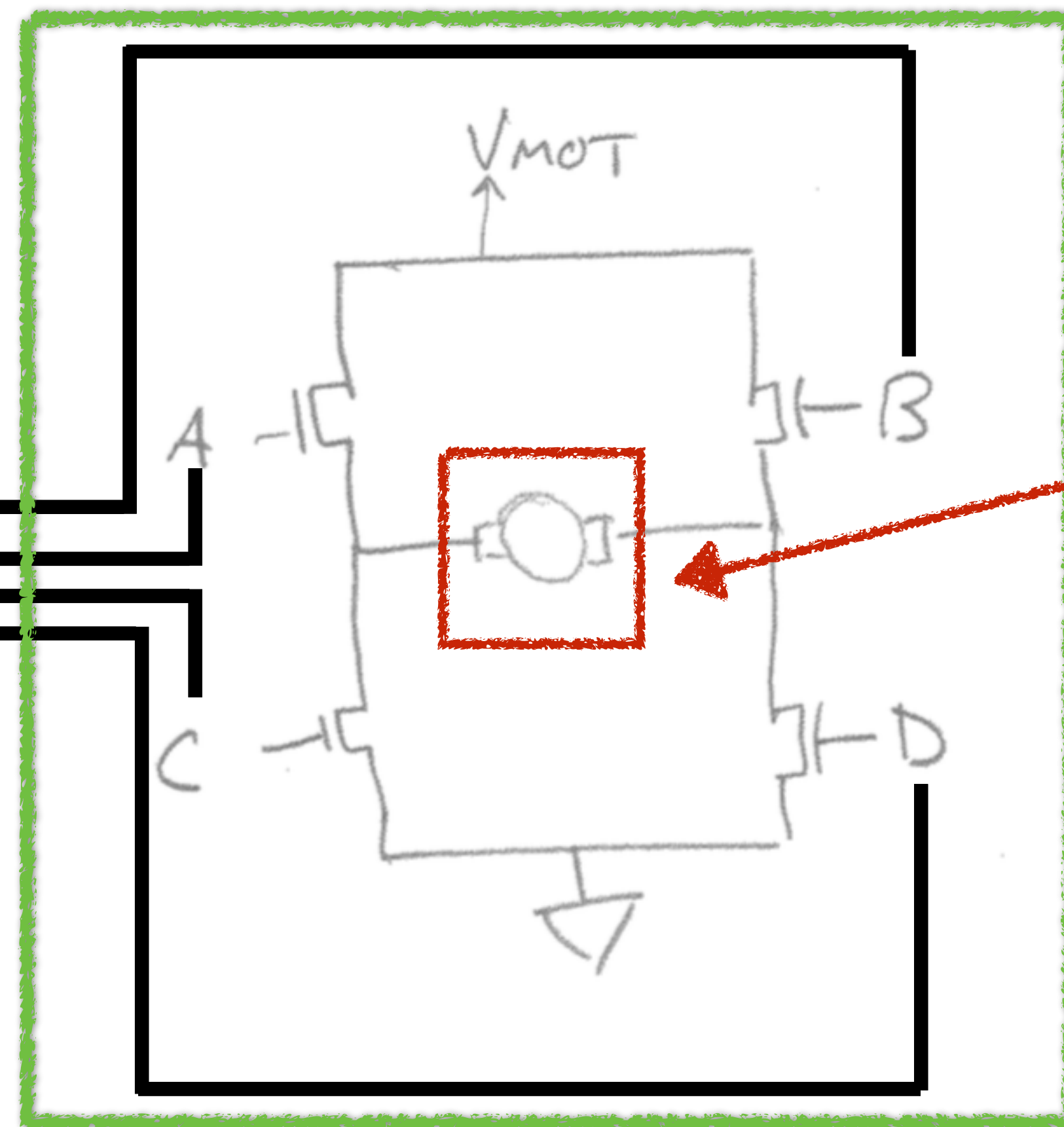


motor controller

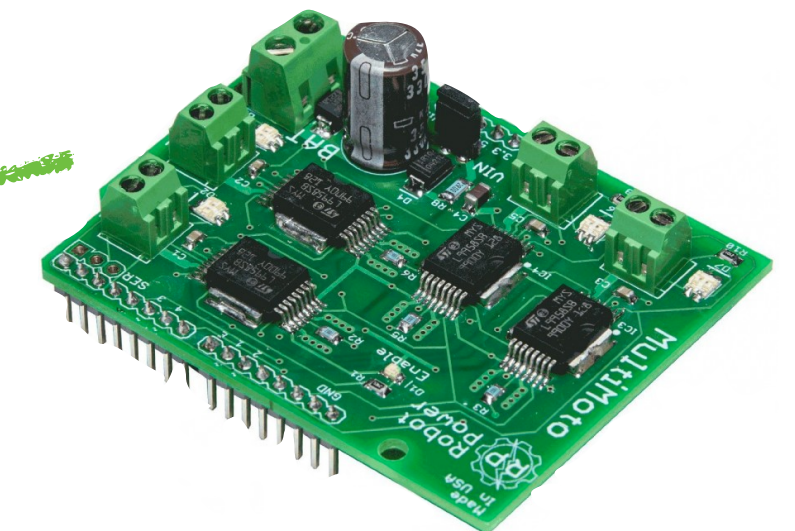
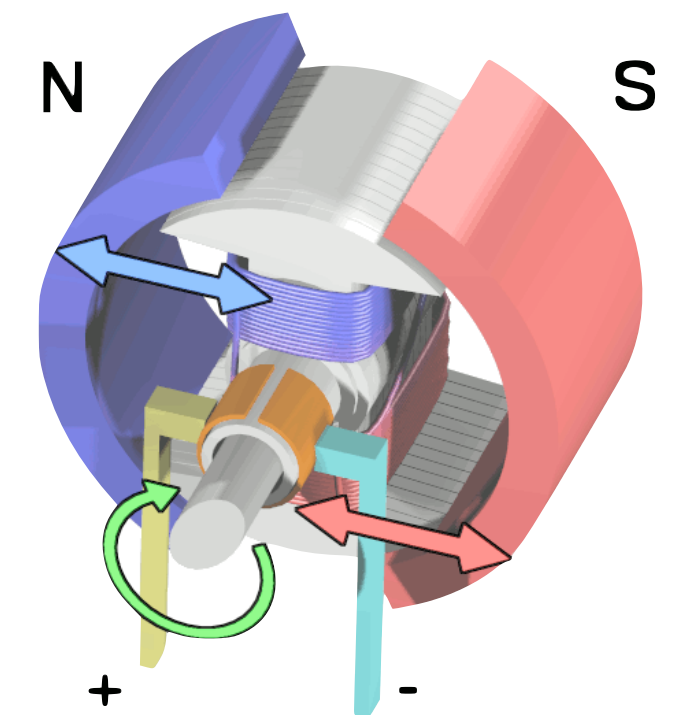
PWM
(Pulse Width Modulation)

**Control
Signal**

PWM signal sent
to H-bridge
controller.

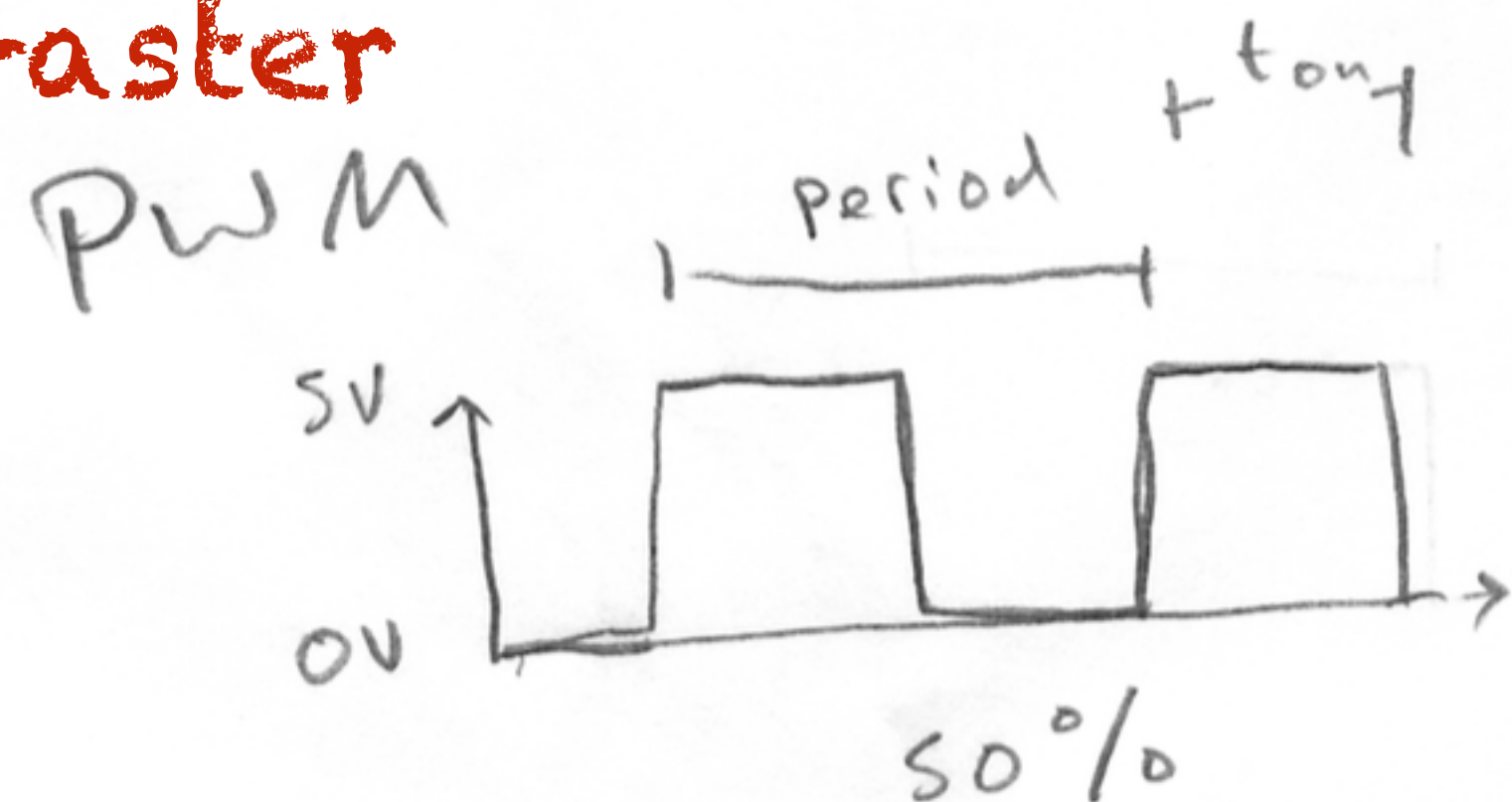


motor



motor controller

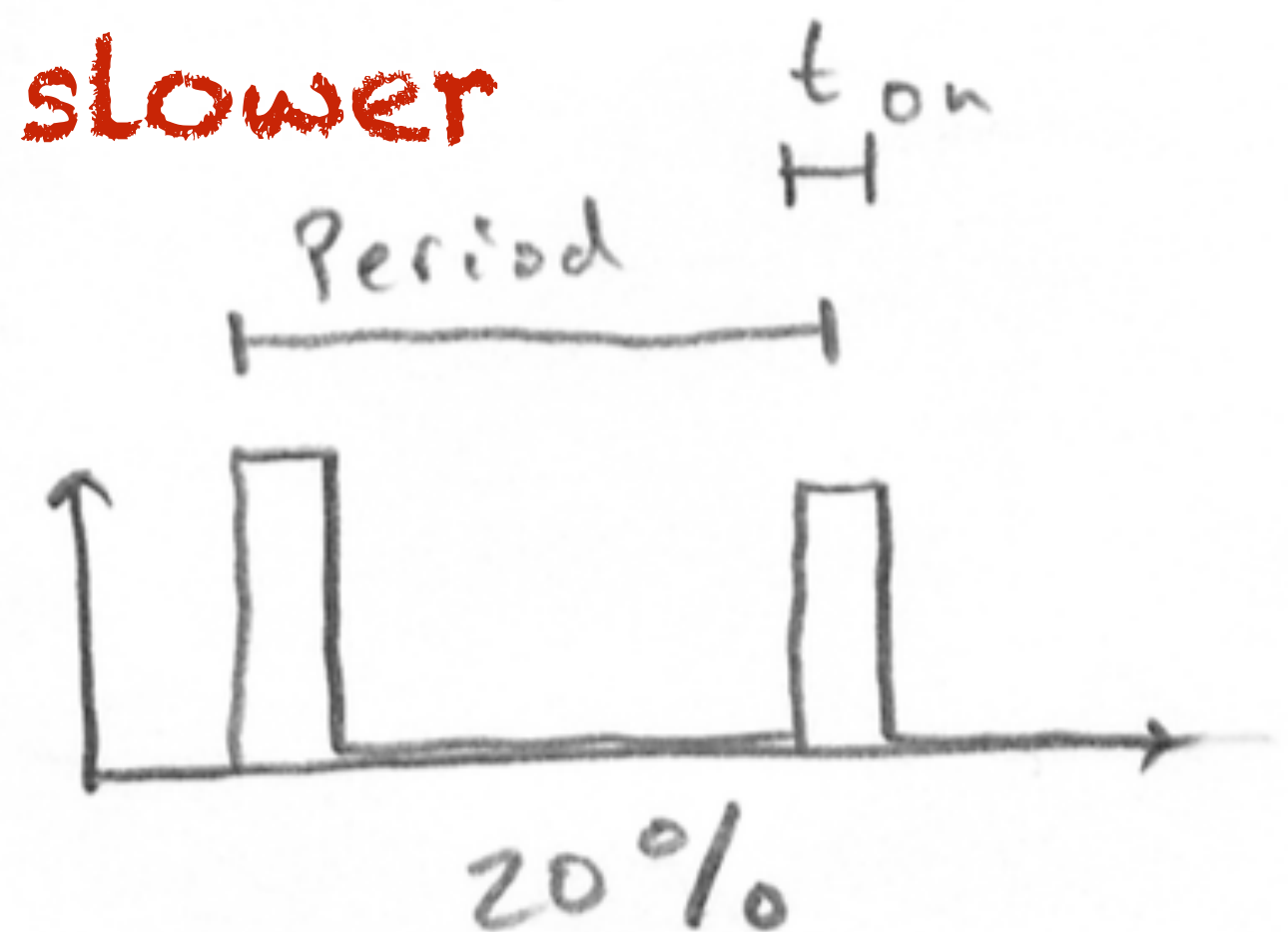
faster



PWM has a periodic duty cycle on each line.

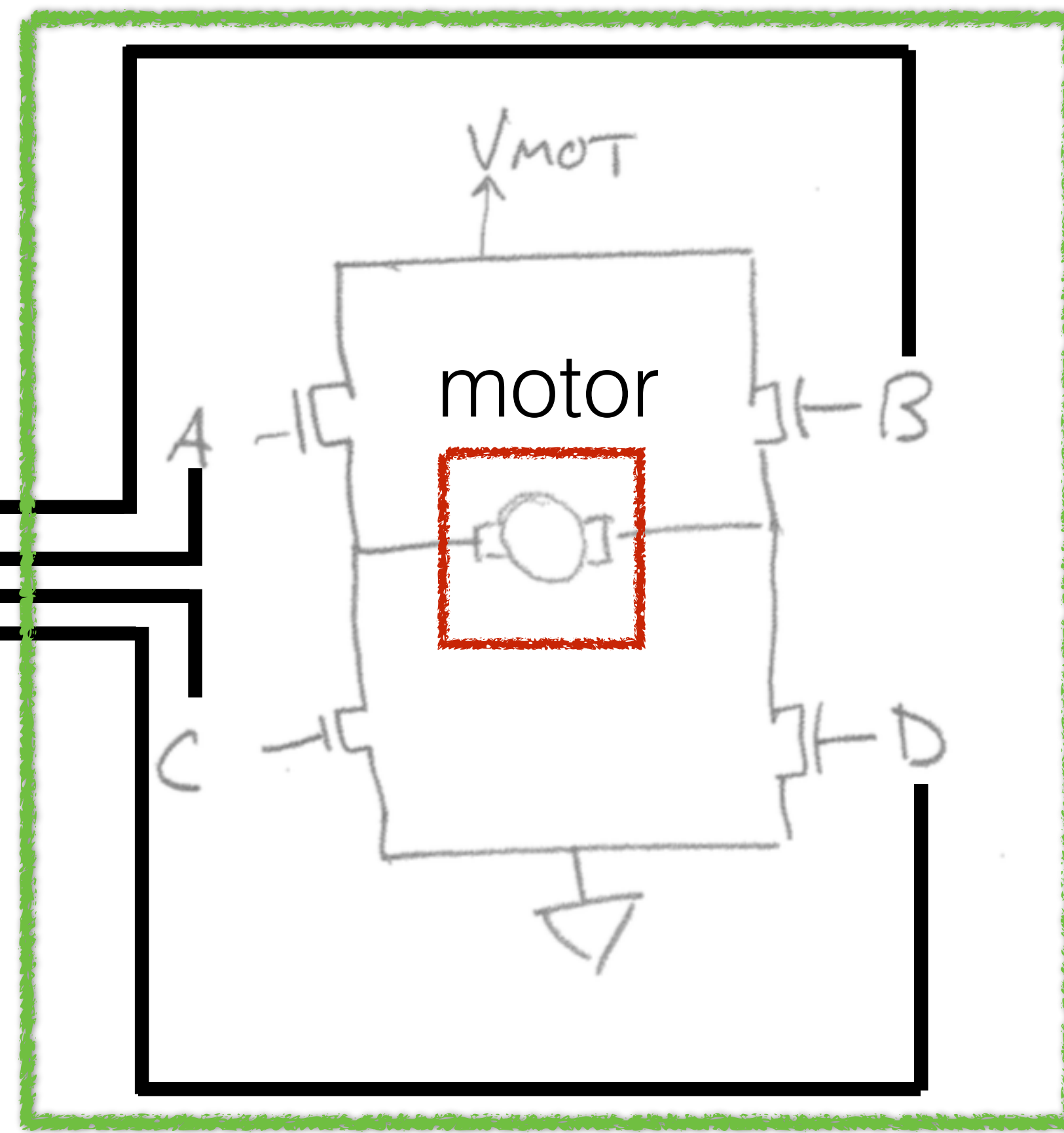
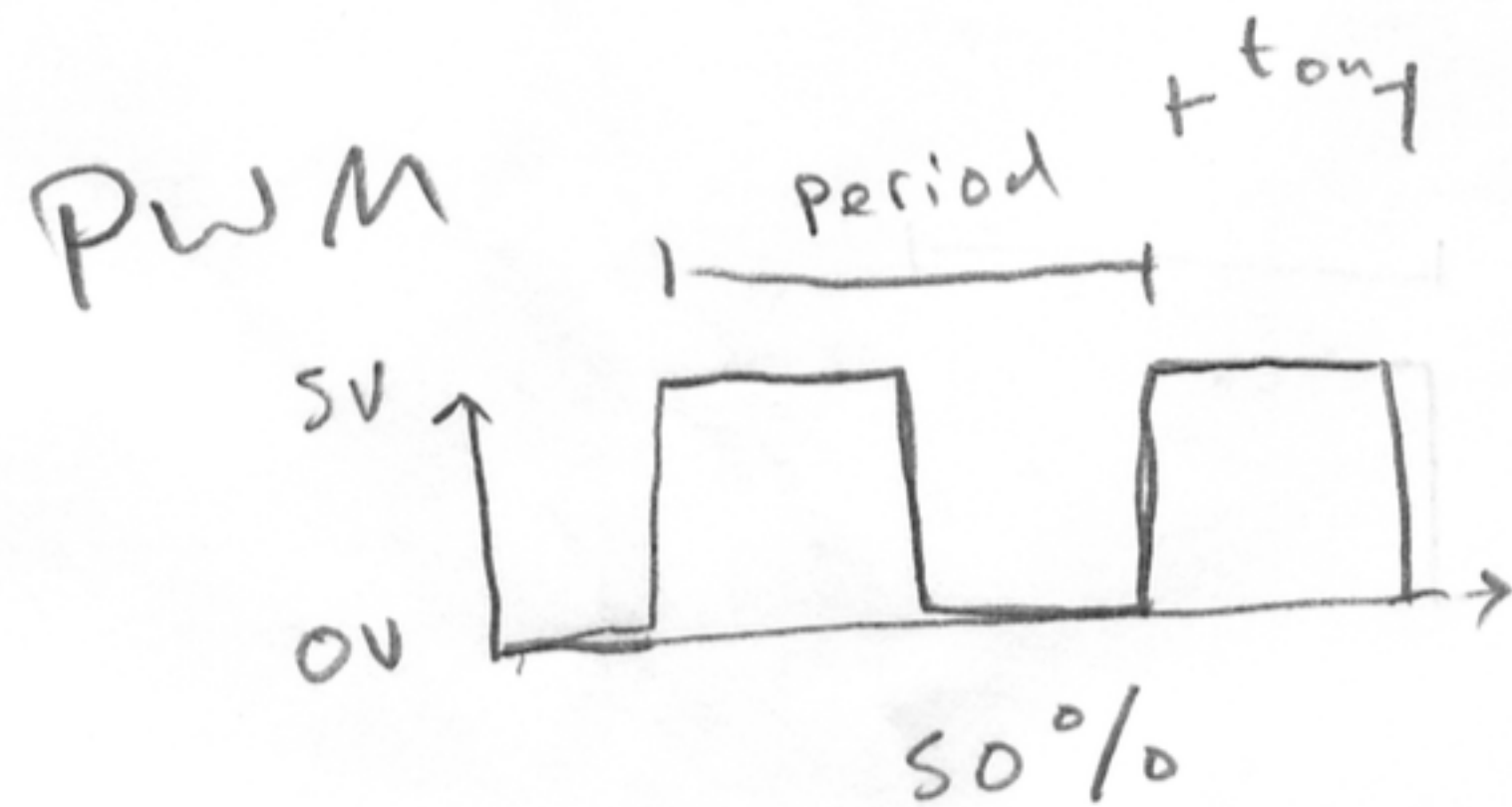
Percentage of time in high voltage ("on" state) determines speed of motor

slower



motor controller

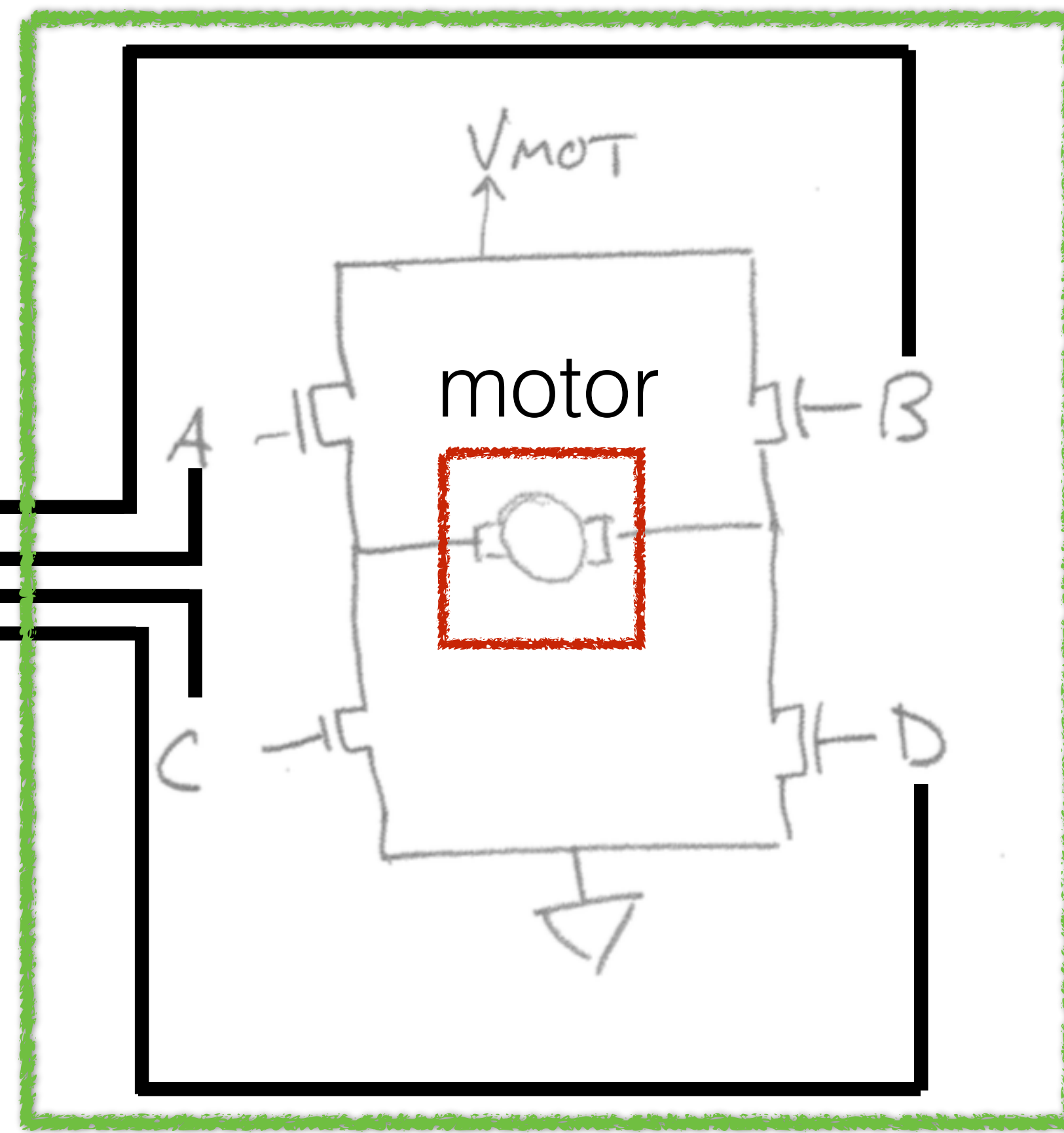
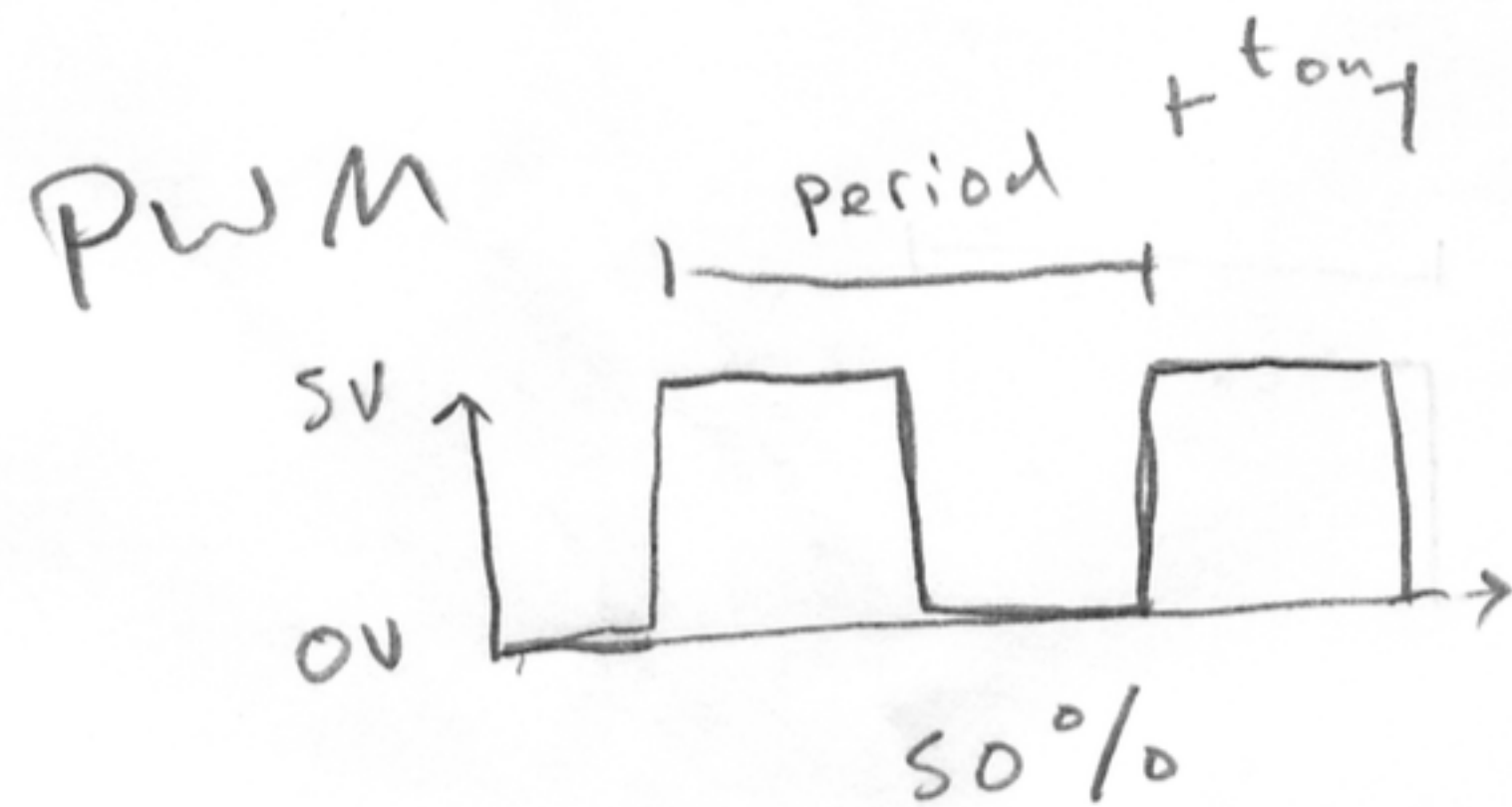
Control Signal



H-bridge has four
switches that are
"on" or "off"

motor controller

Control Signal



spin forward:

A and D on



spin backward:

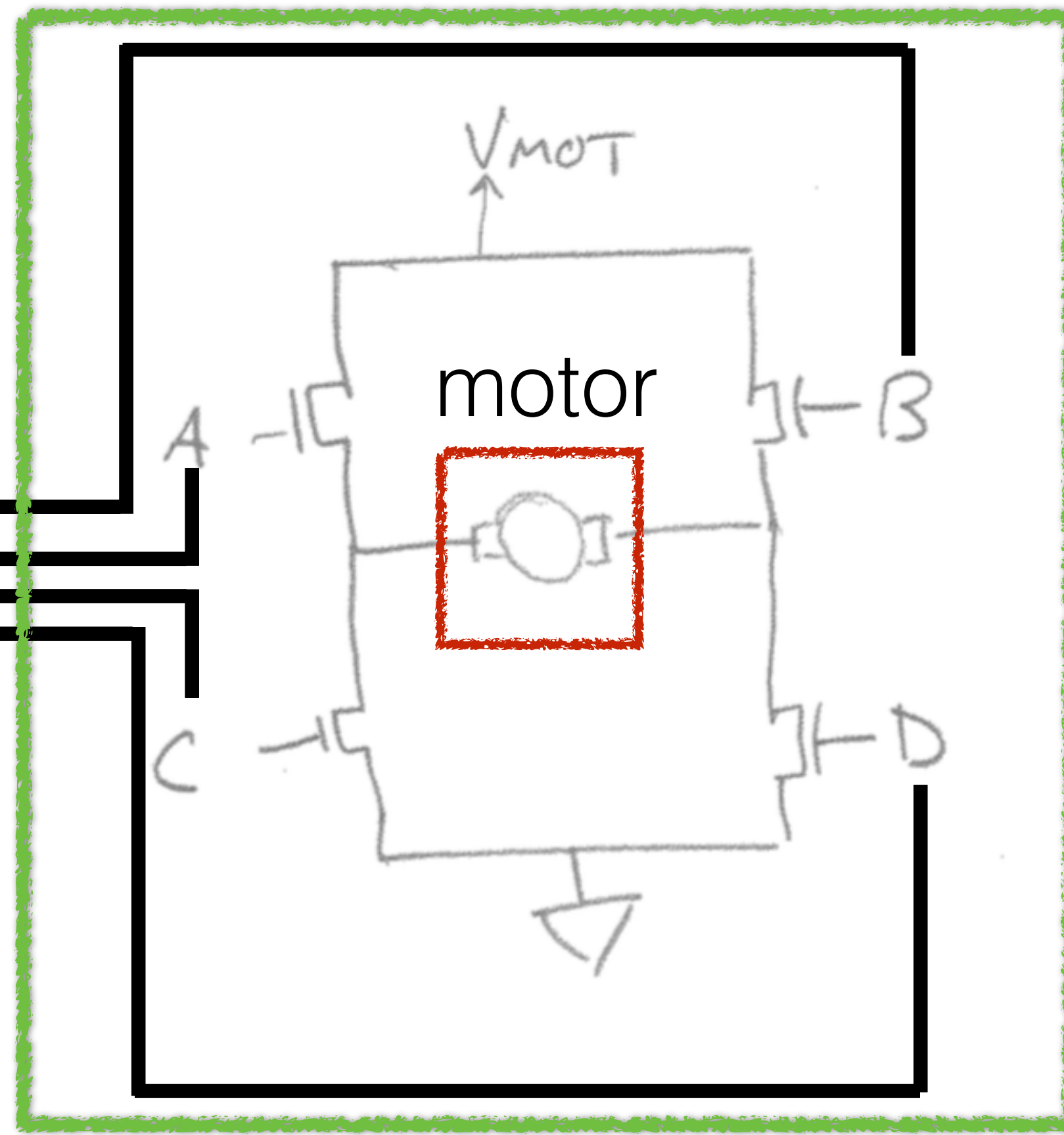
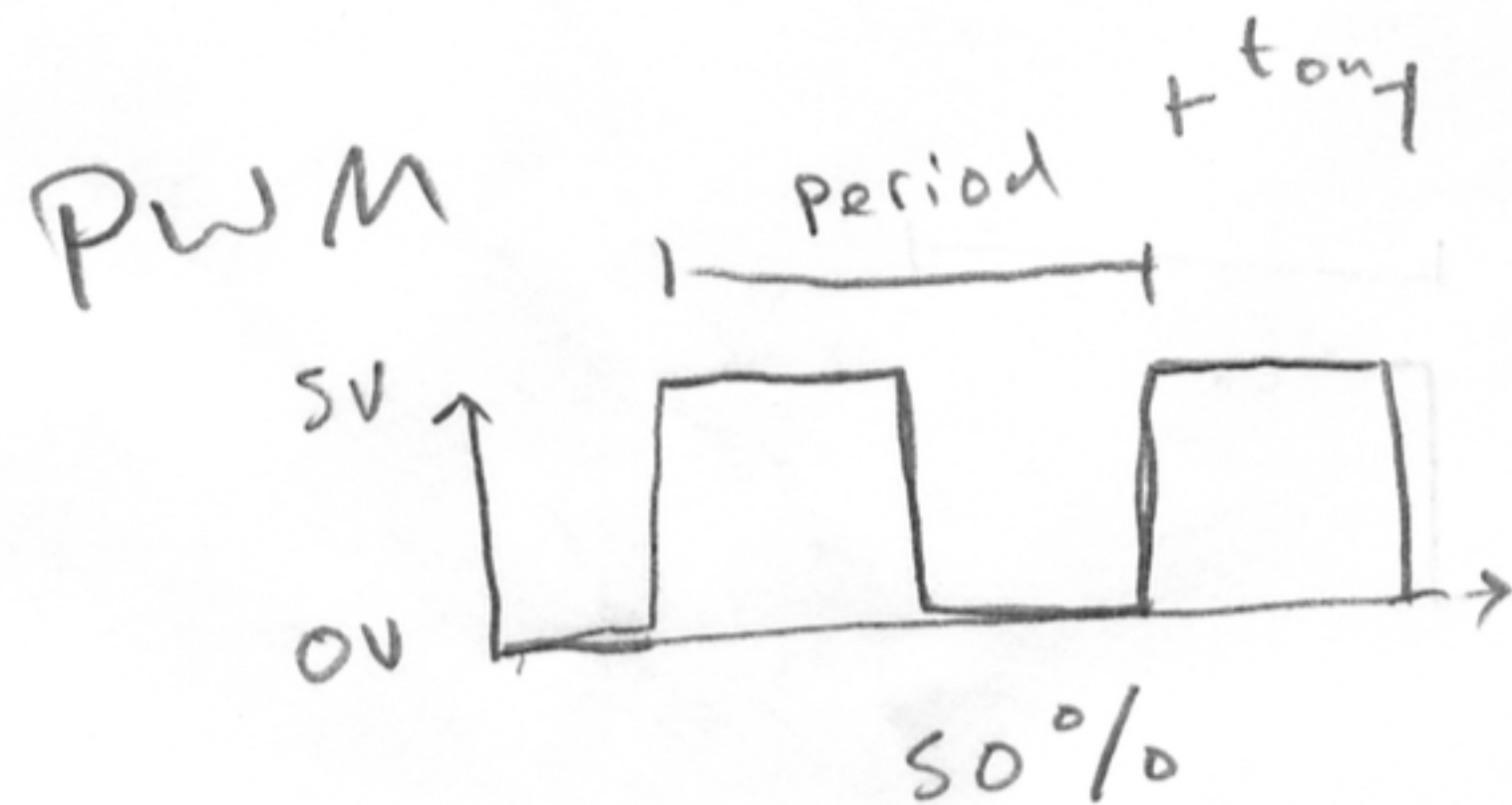
B and C on



H-bridge has four switches that are "on" or "off"

motor controller

Control Signal



spin forward:
A and D

spin backward:
B and C

what if none are on?

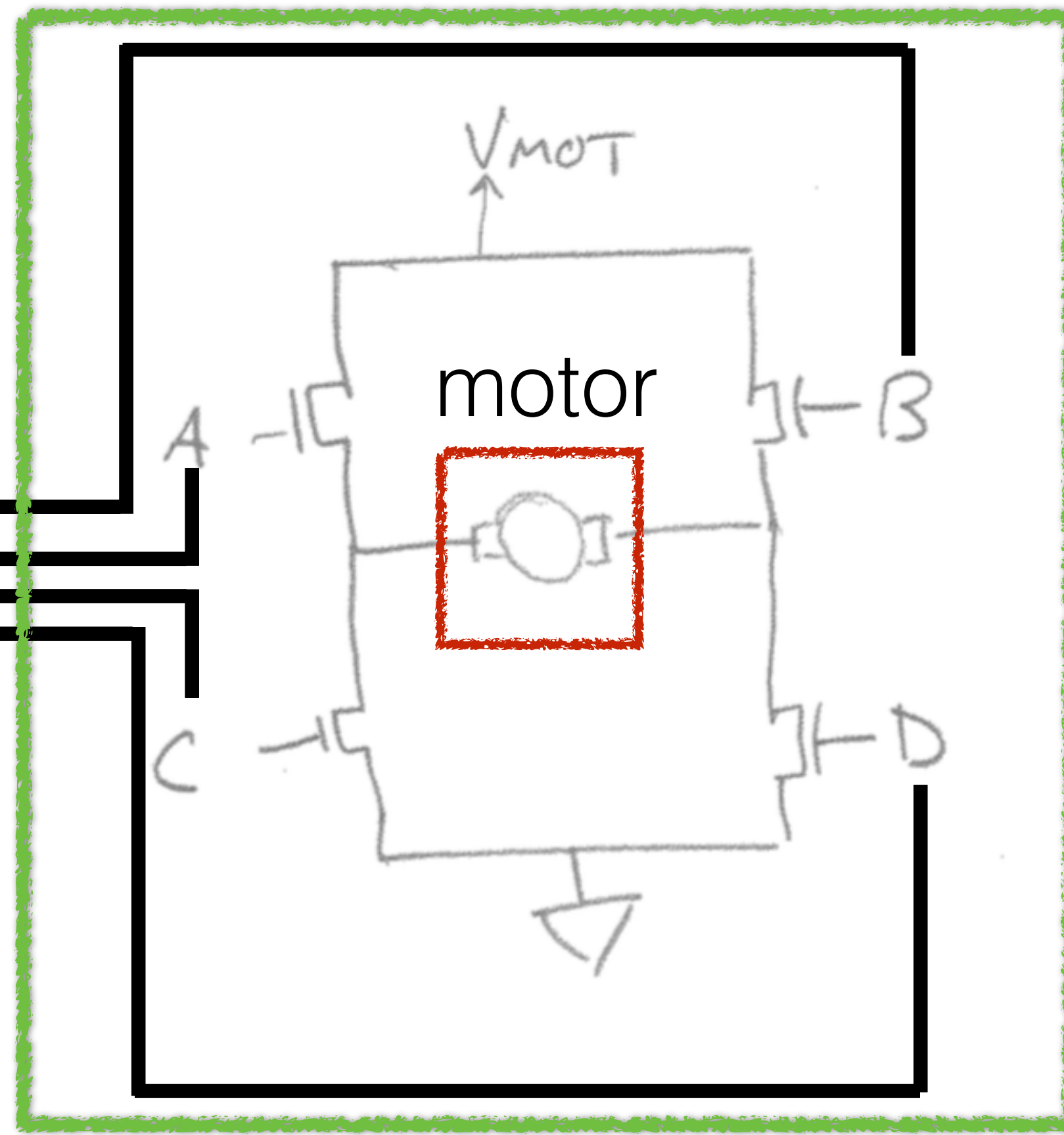
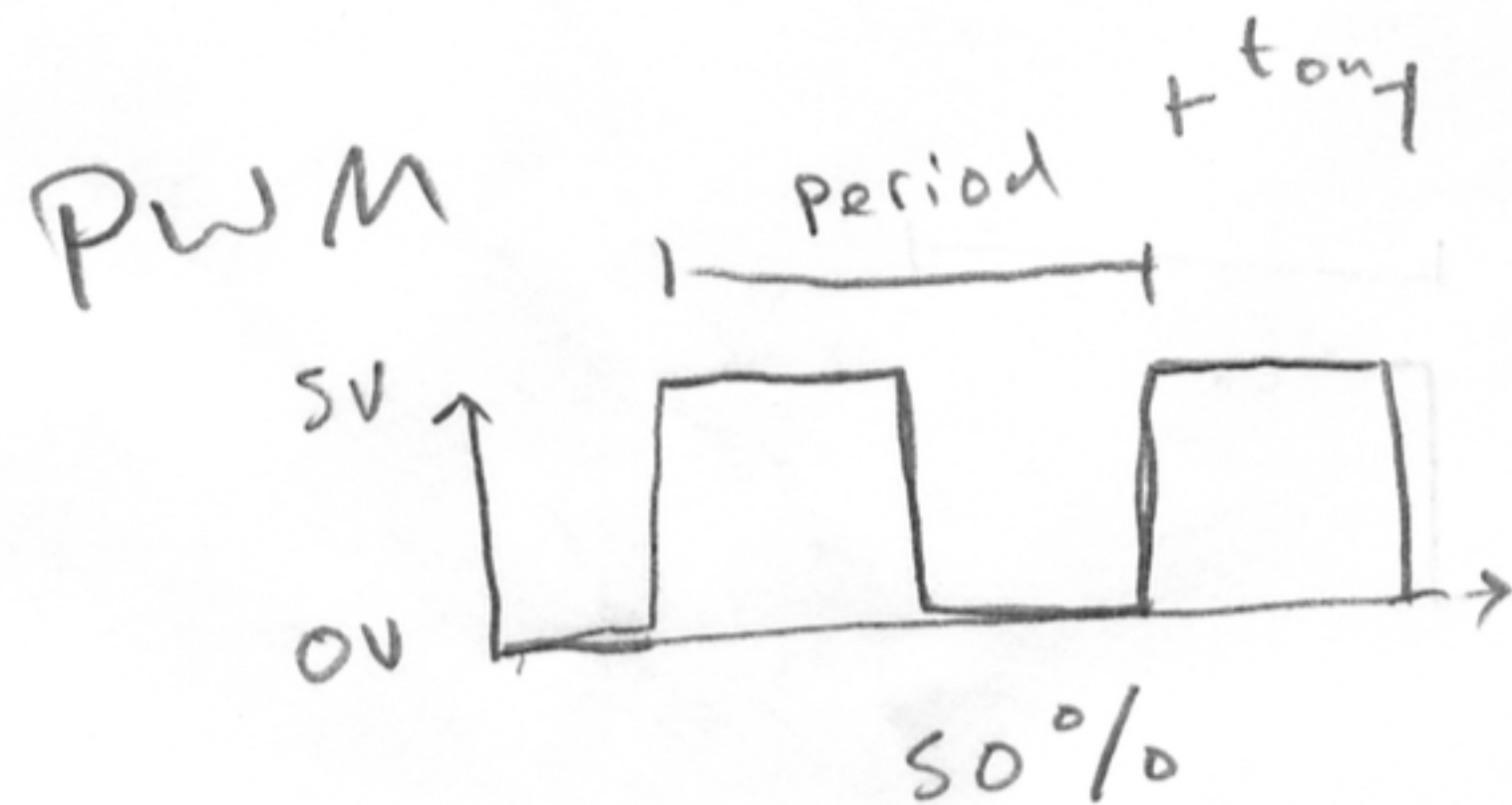
what if C and D?

what if A and B?

what if A and C?

motor controller

Control Signal



spin forward:

A and D

spin backward:

B and C

what if none are on?

open circuit

no current $\rightarrow$ no torque

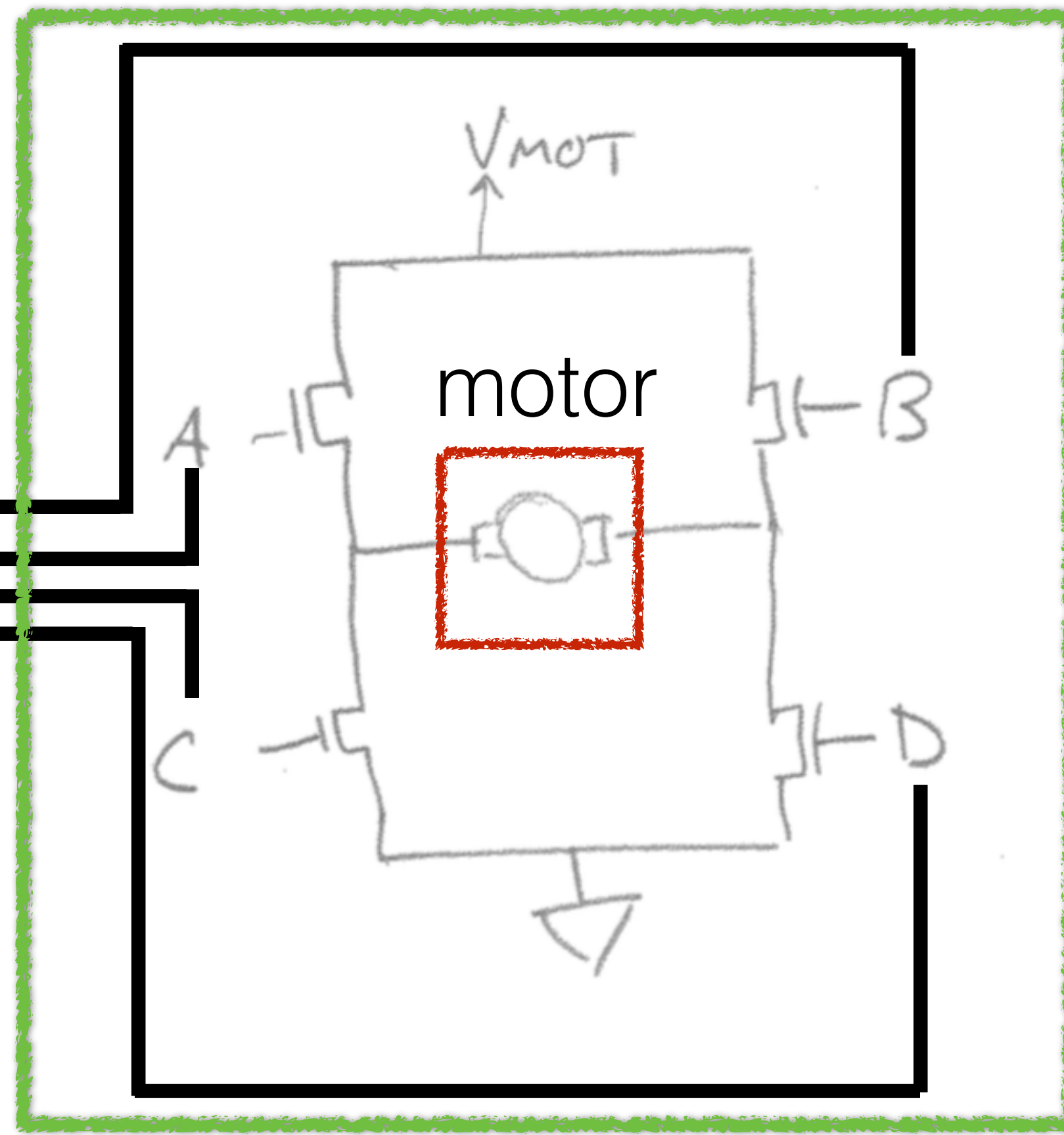
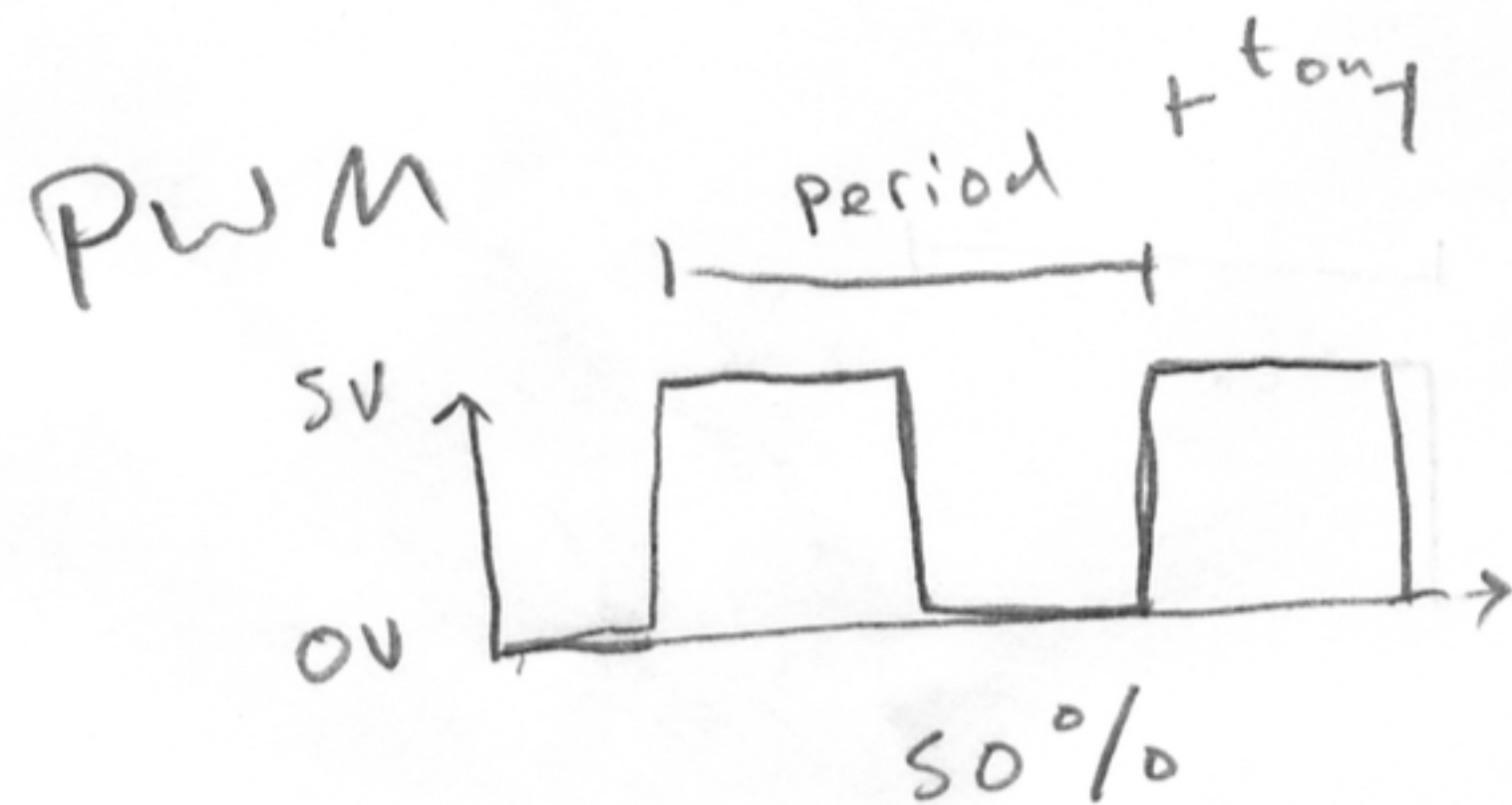
what if C and D?

what if A and B?

what if A and C?

motor controller

Control Signal



spin forward:

A and D

spin backward:

B and C

what if none are on?

open circuit

no current -> no torque

what if C and D?

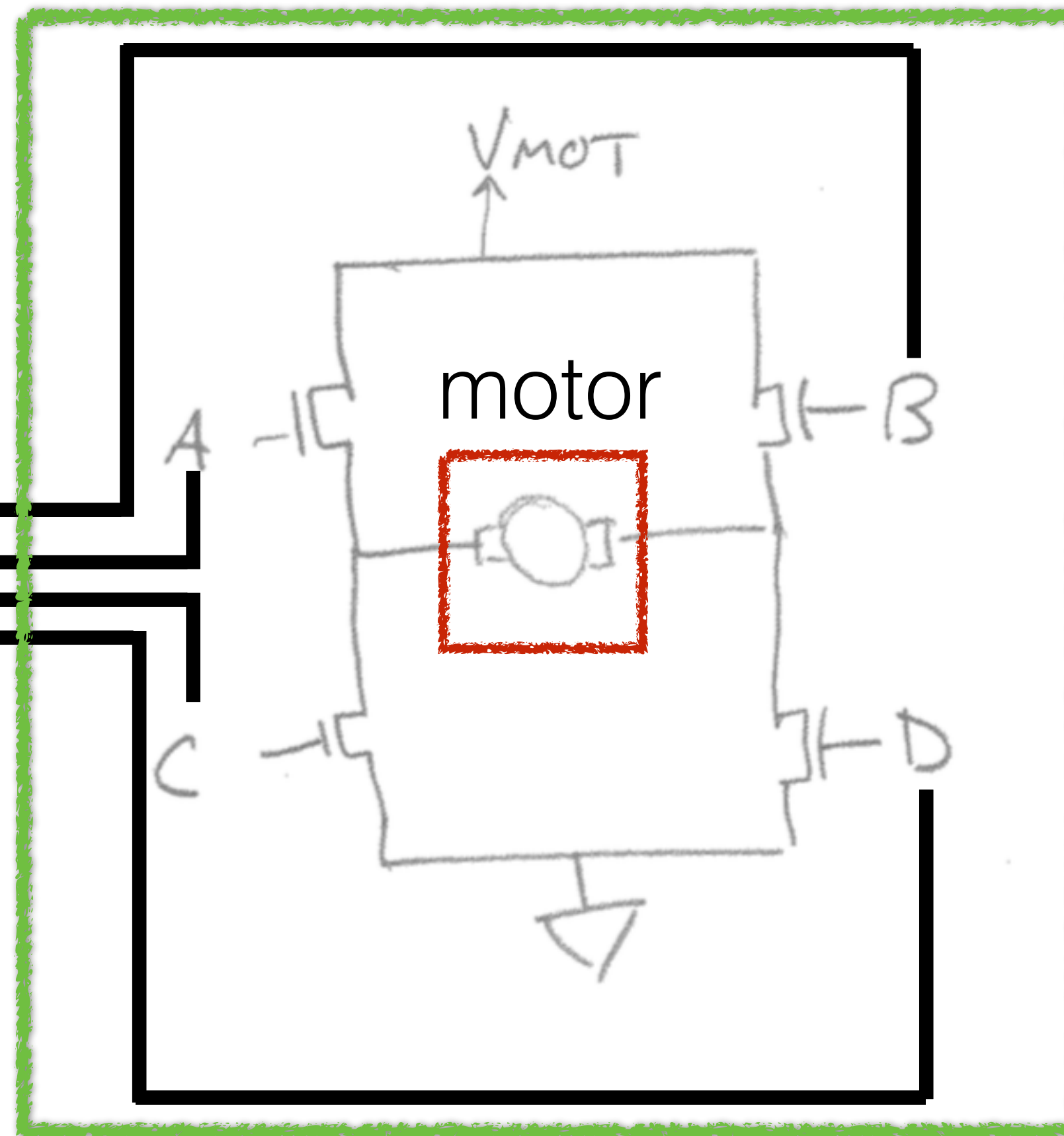
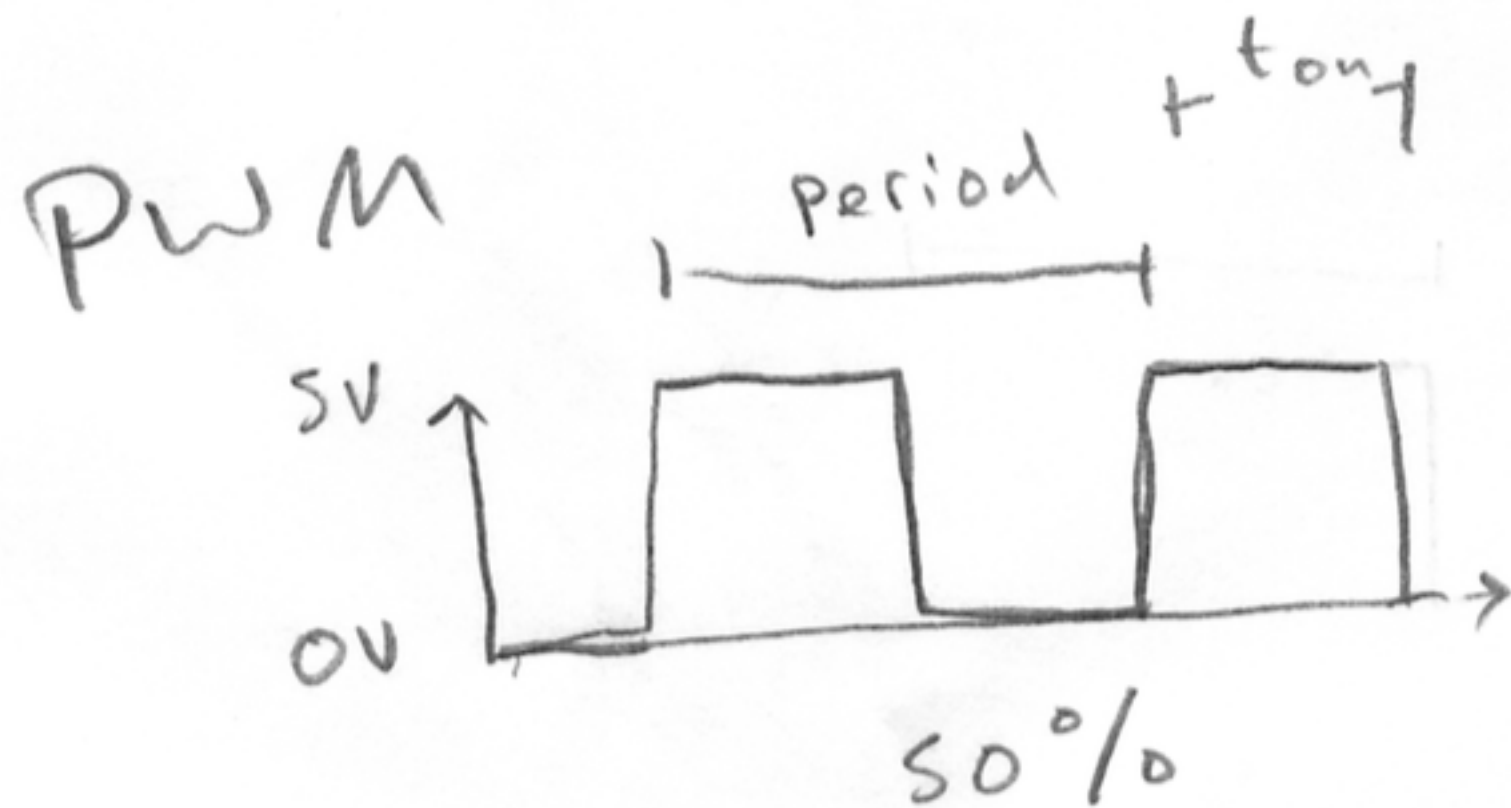
negative torque -> brake

what if A and B?

what if A and C?

motor controller

Control Signal



spin forward:

A and D

spin backward:

B and C

what if none are on?

open circuit

no current -> no torque

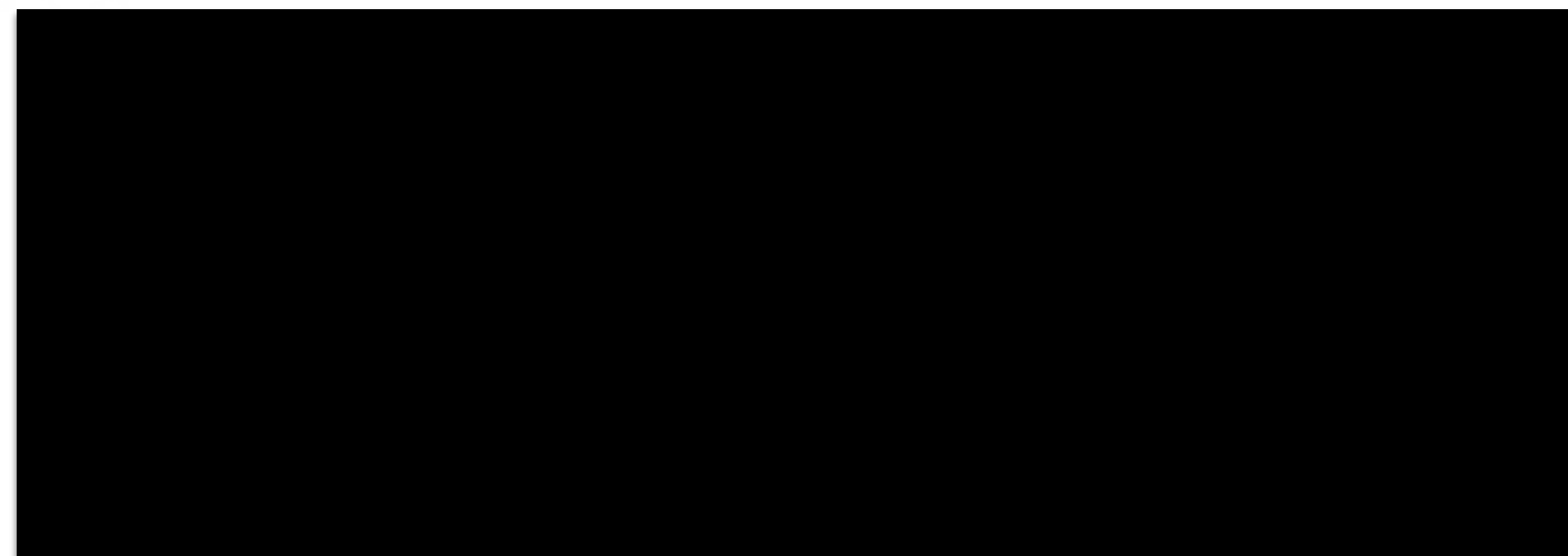
what if C and D?

negative torque -> brake

what if A and B?

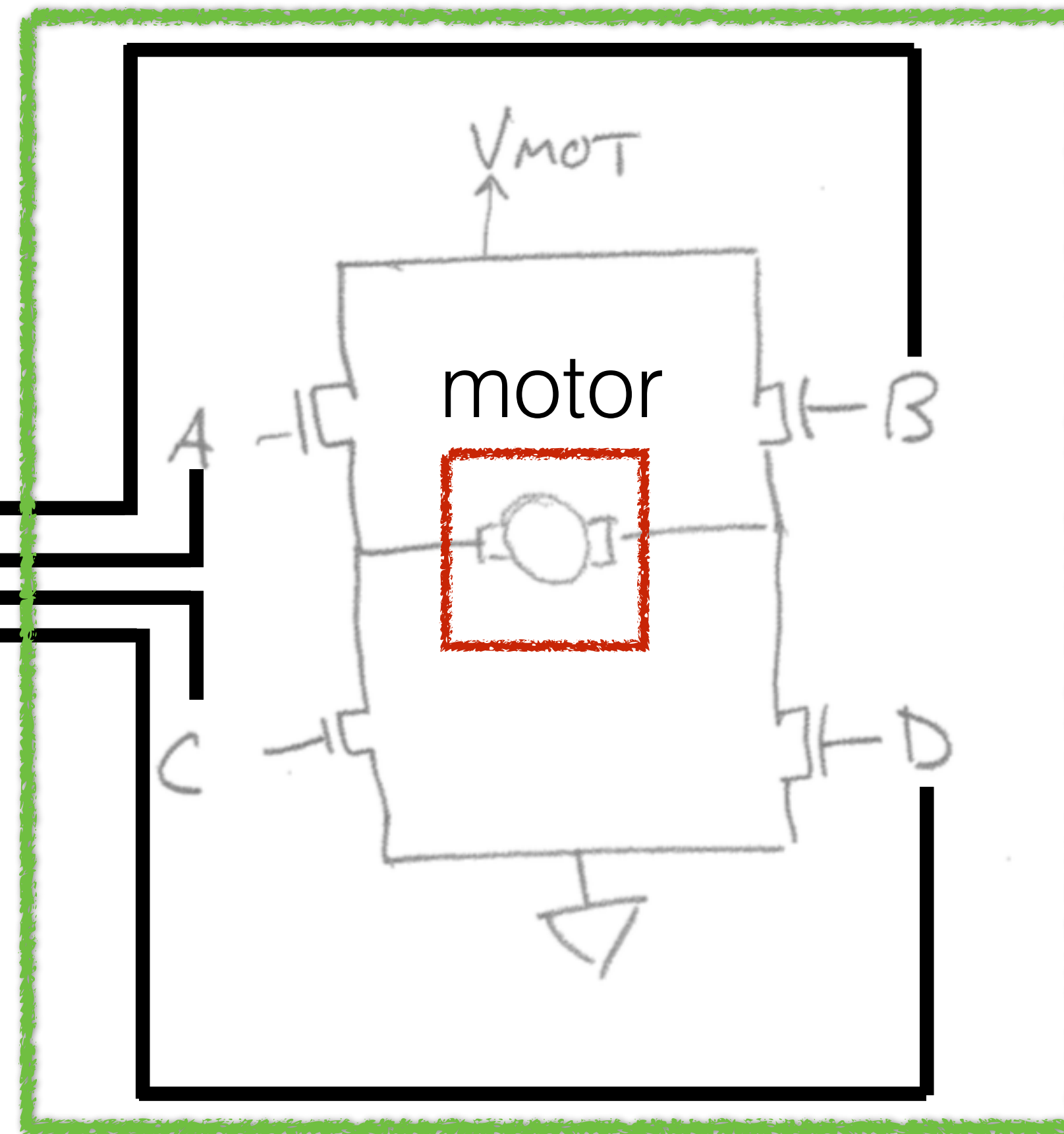
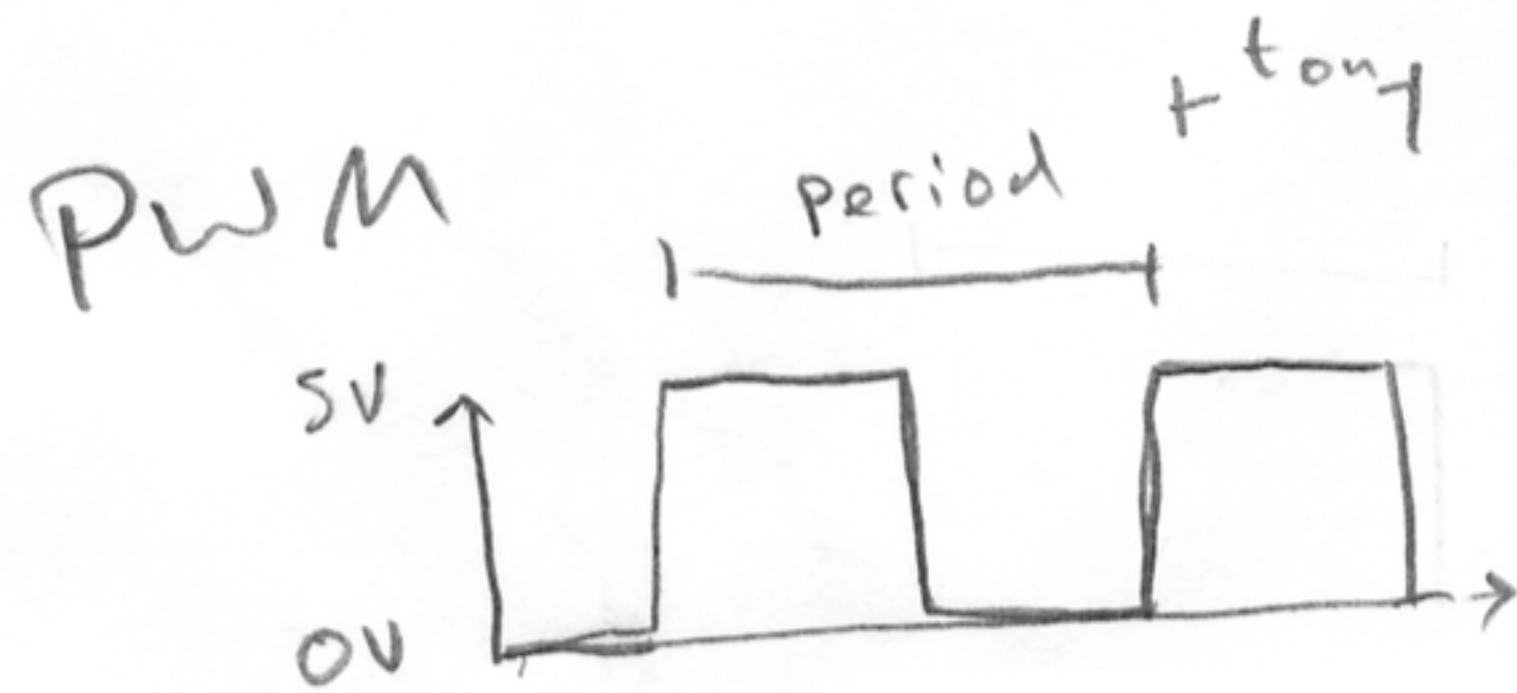
negative torque -> brake

what if A and C?



motor controller

Control Signal



spin forward:

A and D

spin backward:

B and C

what if none are on?

open circuit

no current -> no torque

what if C and D?

negative torque -> brake

what if A and B?

negative torque -> brake

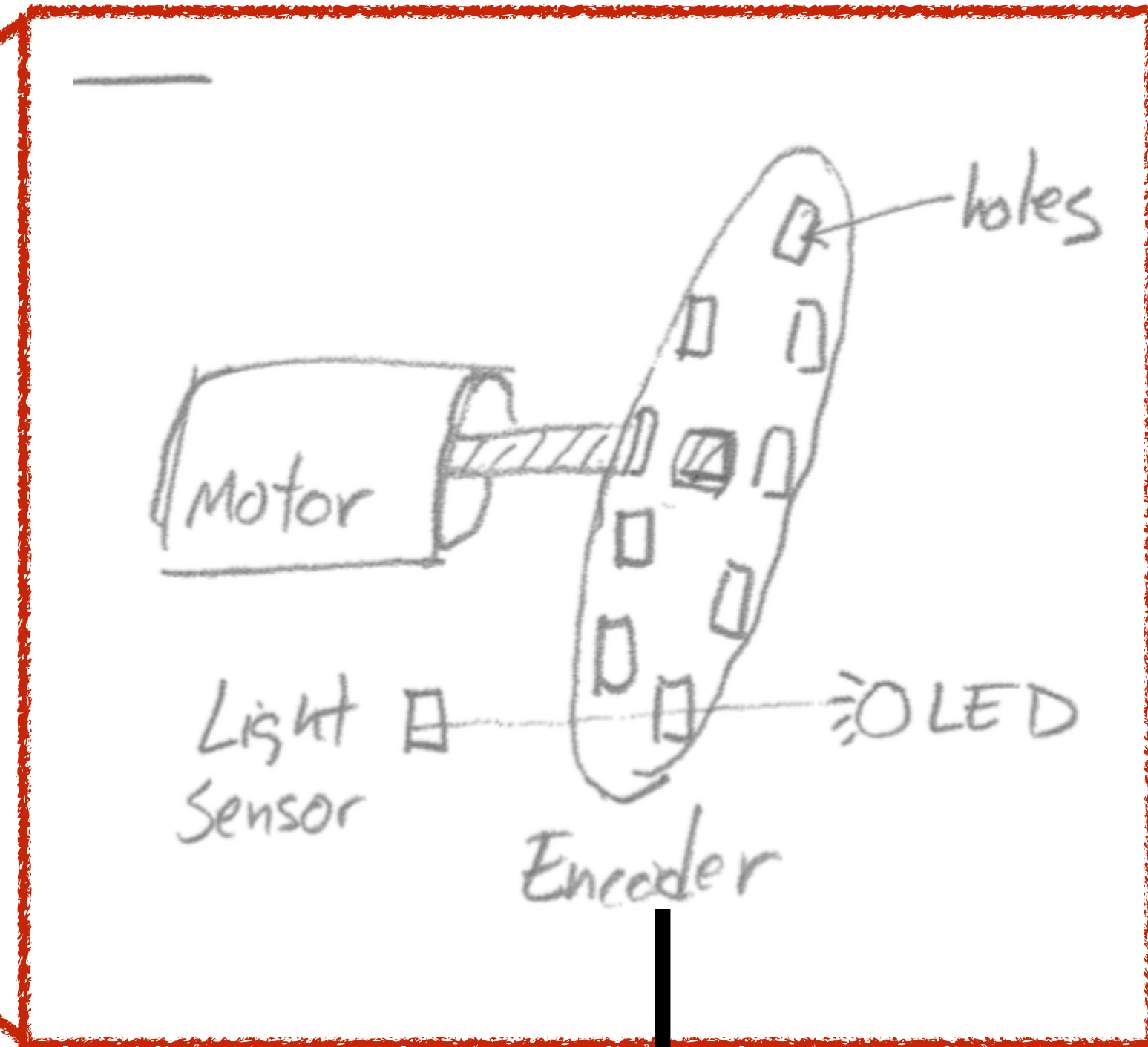
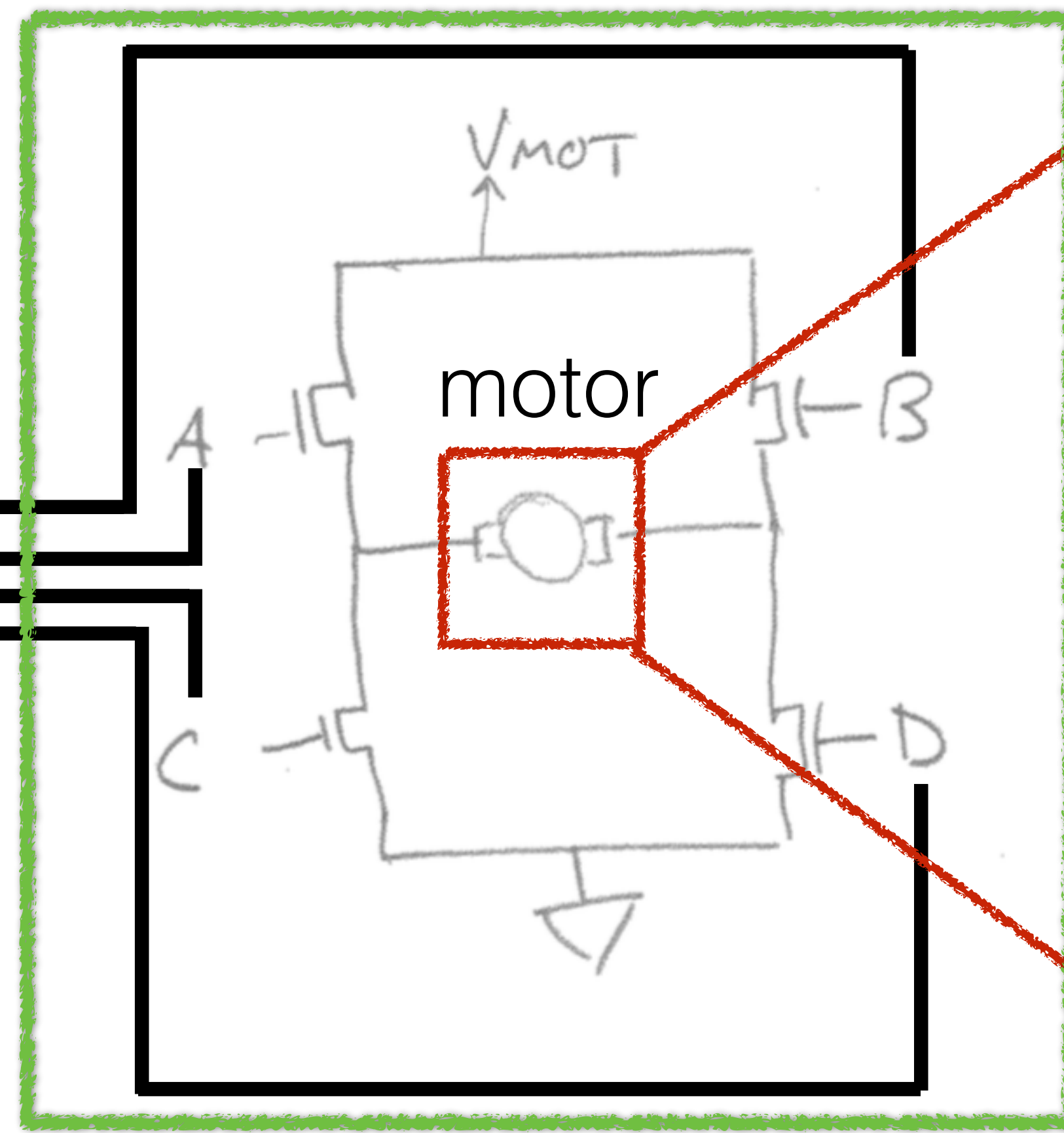
what if A and C?

proof

No more robot,
(our hardware prevents this)

motor controller

Servo Control

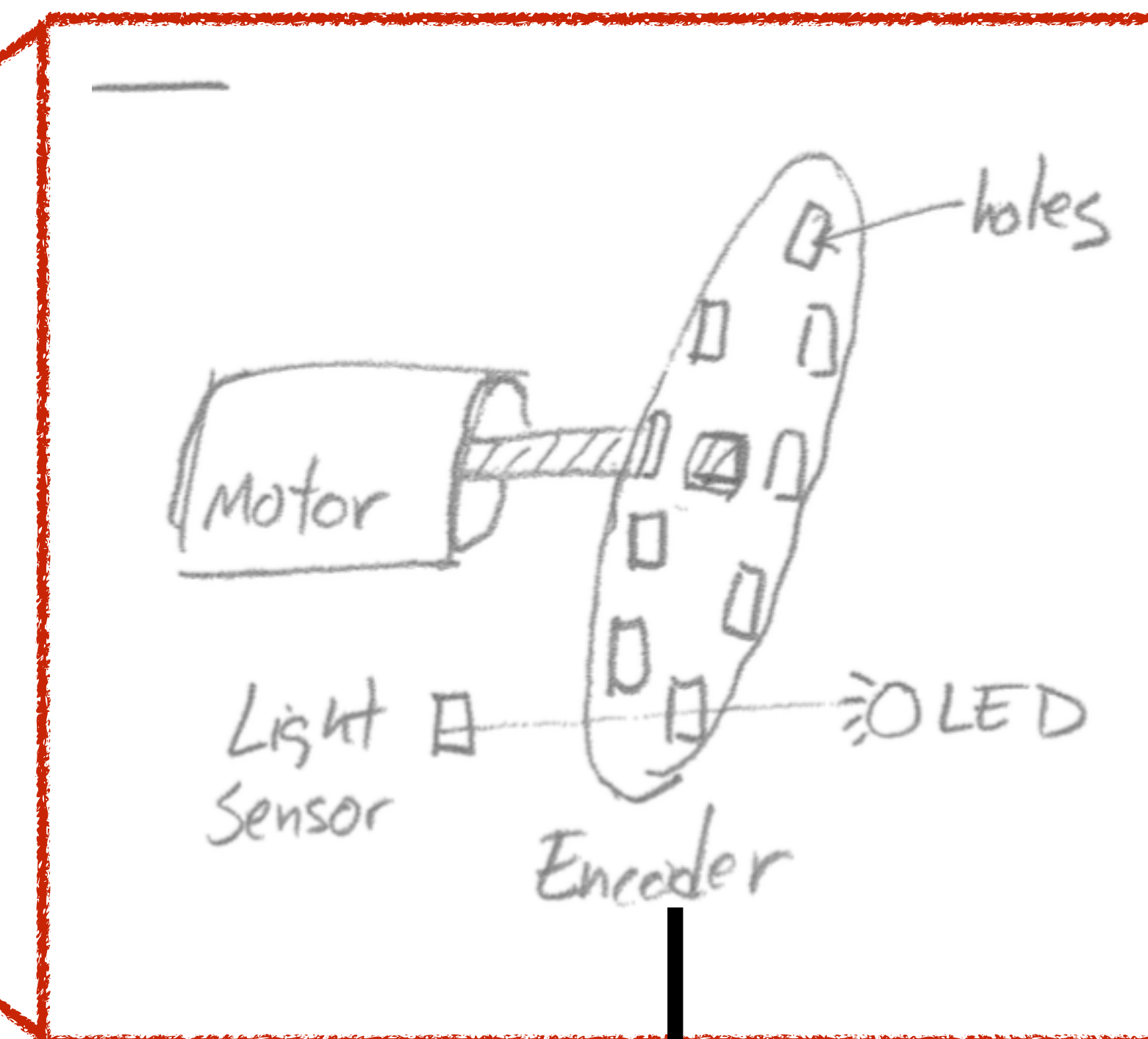
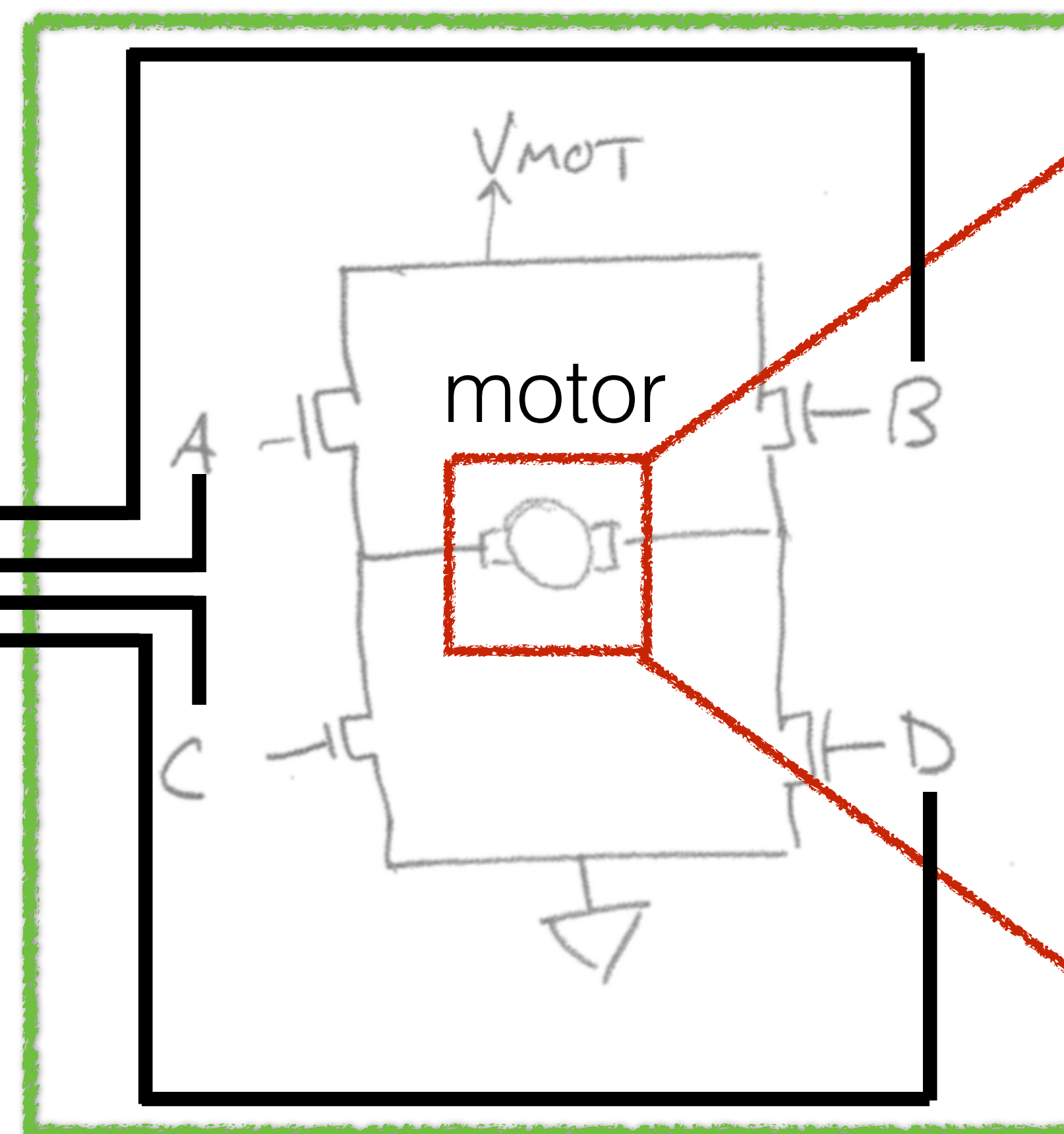


servo controller
generates control signal

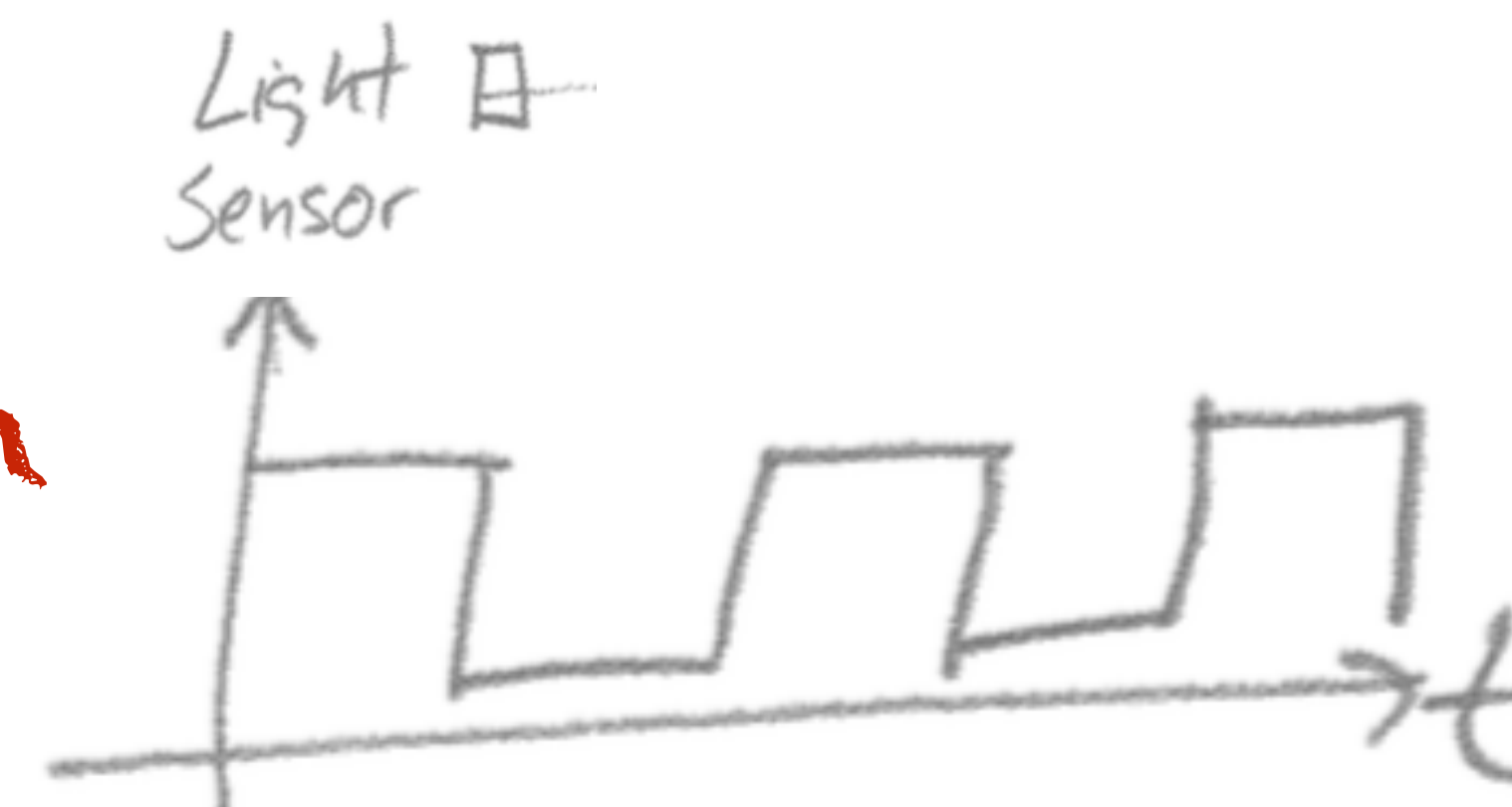
encoder sends joint
state to back to
servo controller

Servo Control

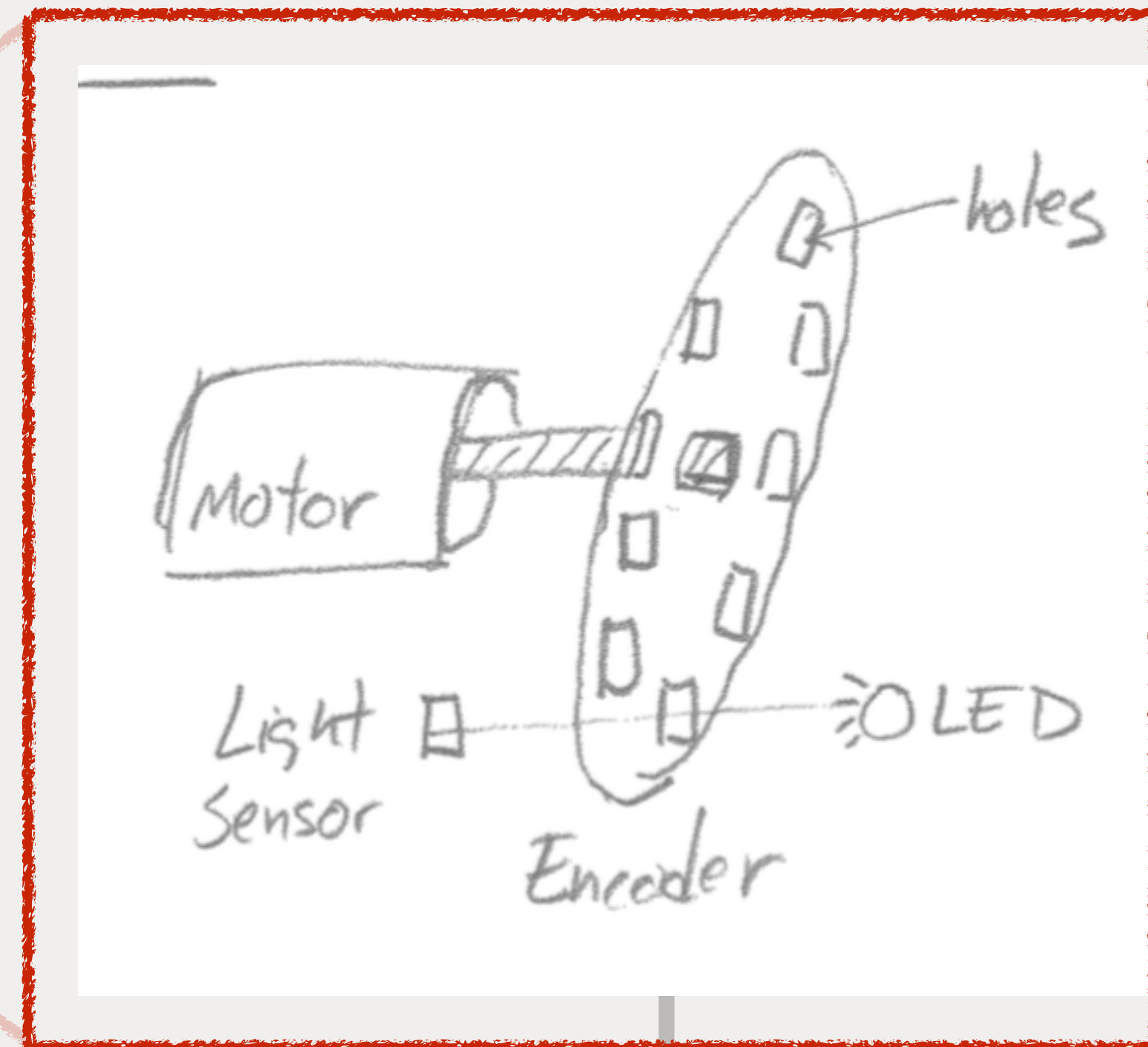
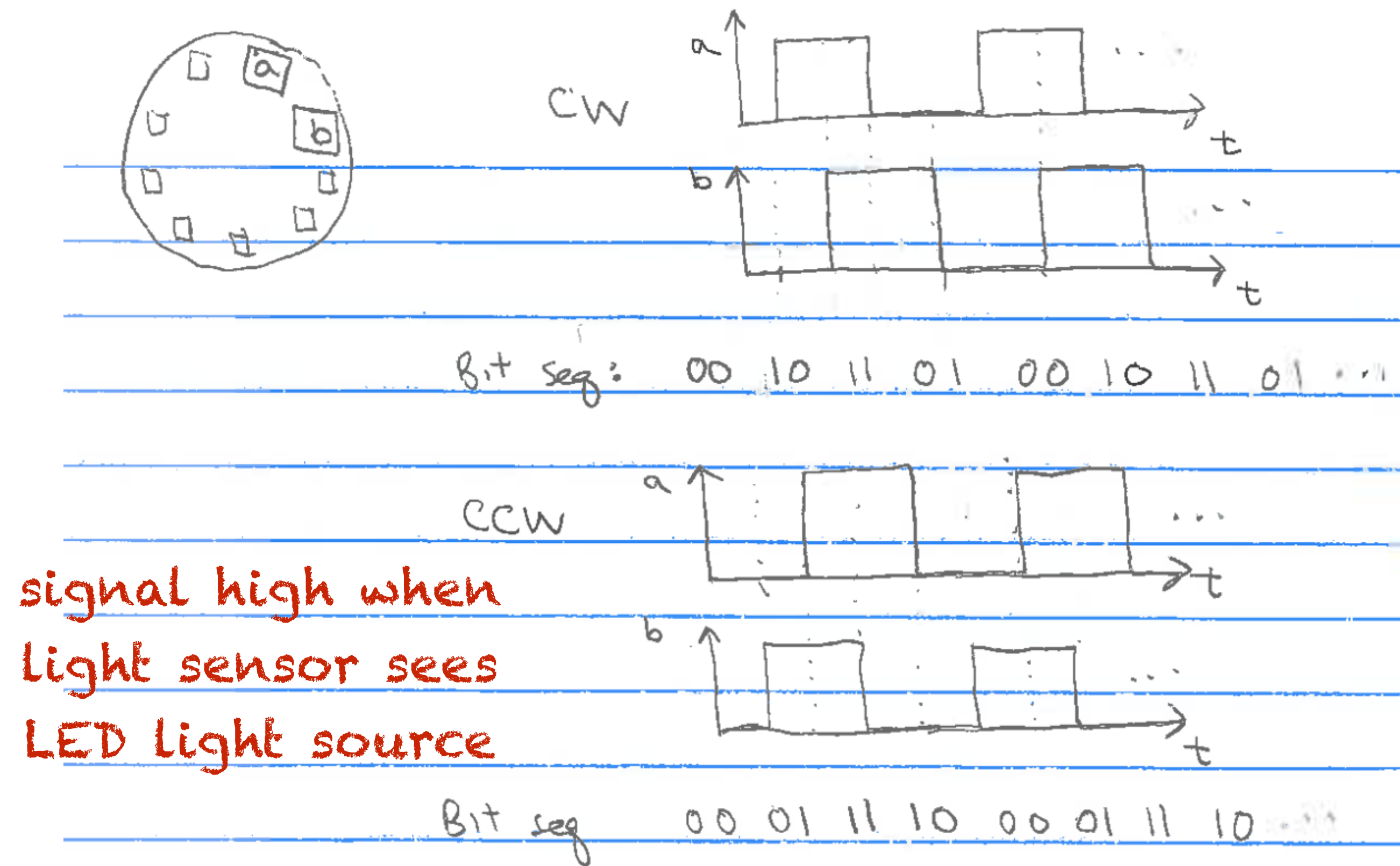
motor controller



as motor turns,
encoder counts transitions between
sensed and occluded light



Quadrature Encoder

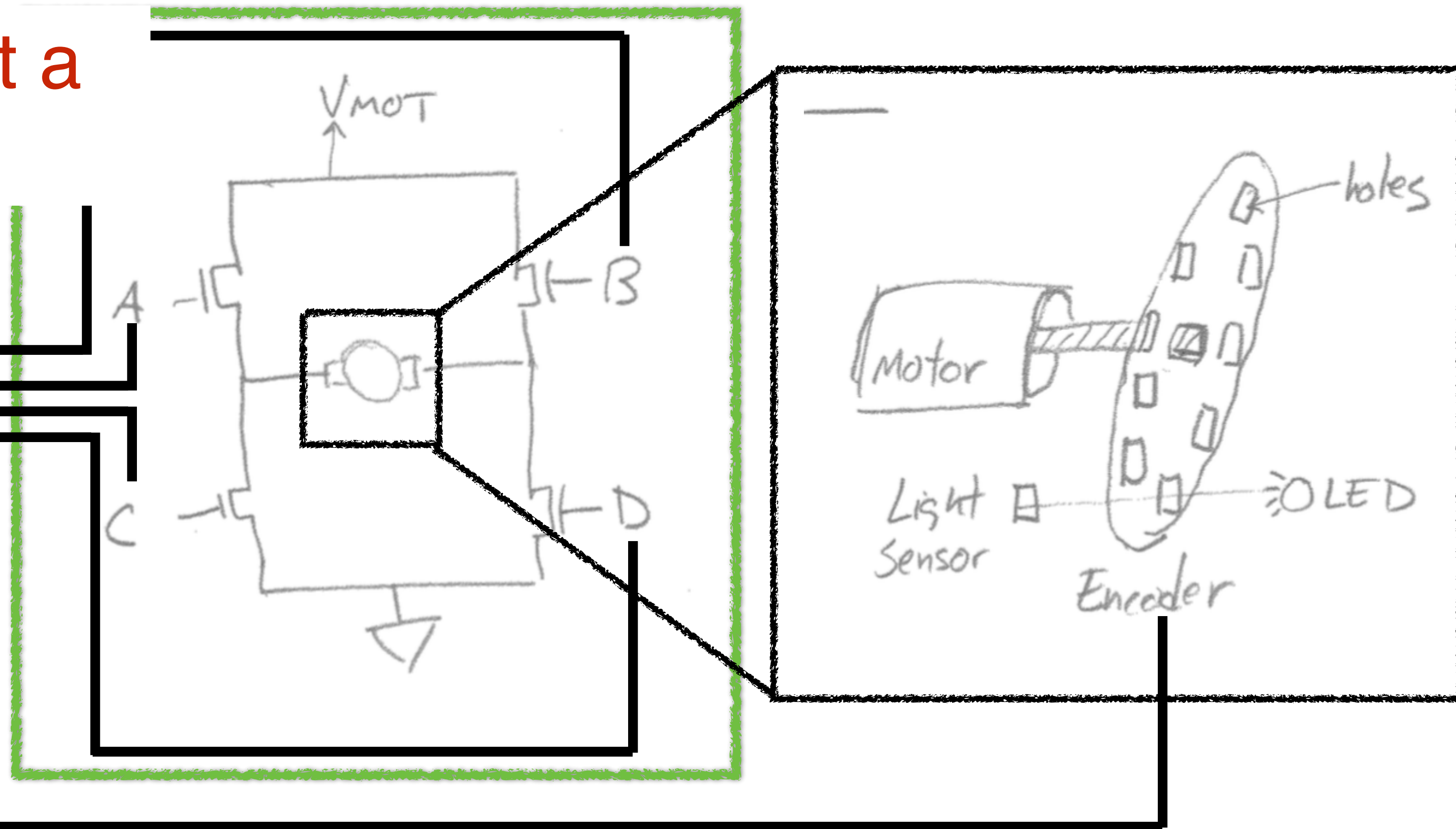


Encoders measure how much a motor has spun.

Quadrature encoders are commonly used. Why?

How do we implement a servo controller?

Servo Control



How do we implement a servo controller?

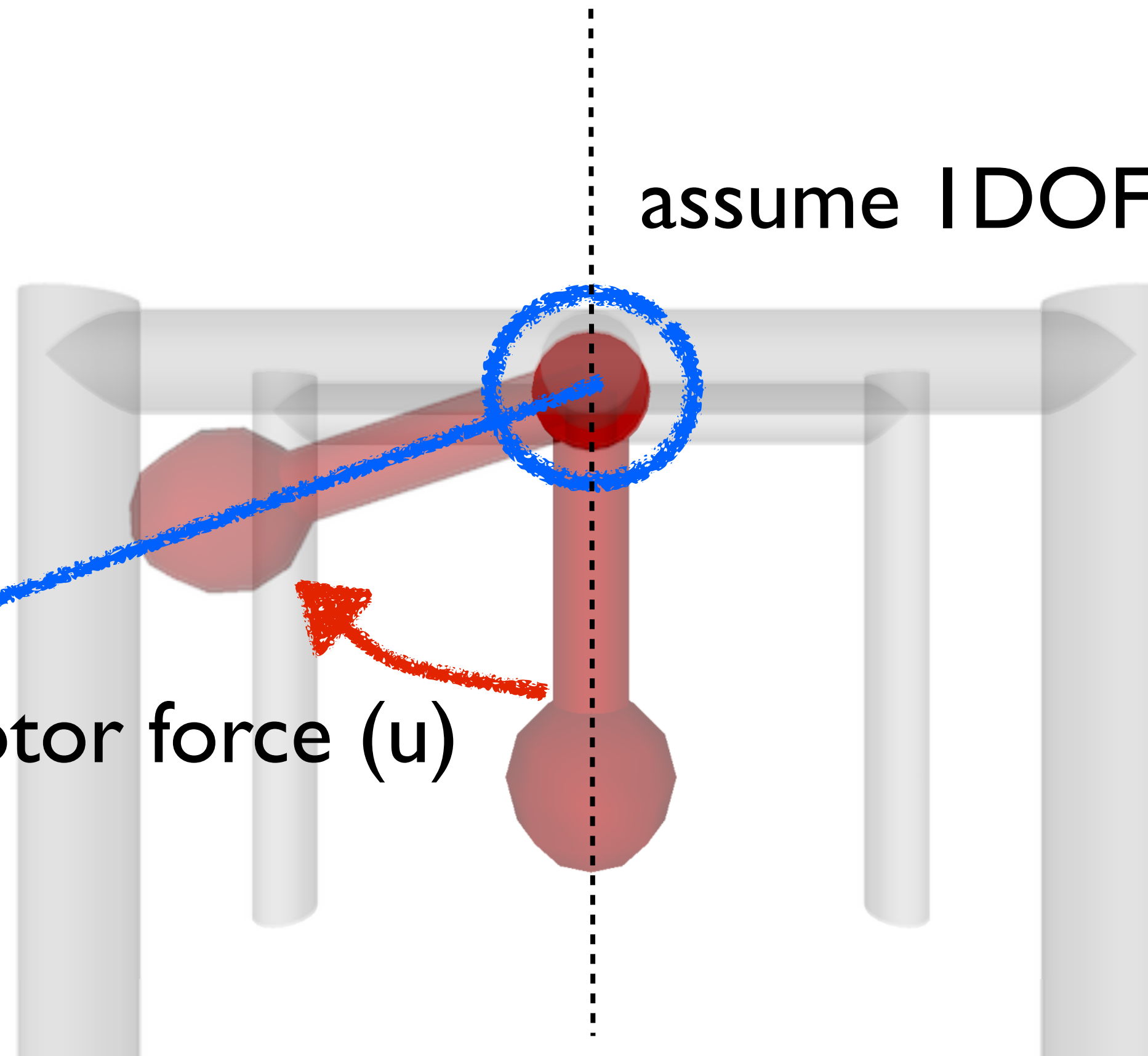
Servo Control

Users/cjenkins/courses/cs148_2014/3jsbot/pendulum/pendulum.html
26 $\dot{x} = 0.00$ $x_{\text{desired}} = -1.26$ $u = -37.32$ $m = 2.00$ $l = 2.00$ integrator = velocity verlet

assume 1DOF motor at pendulum axis

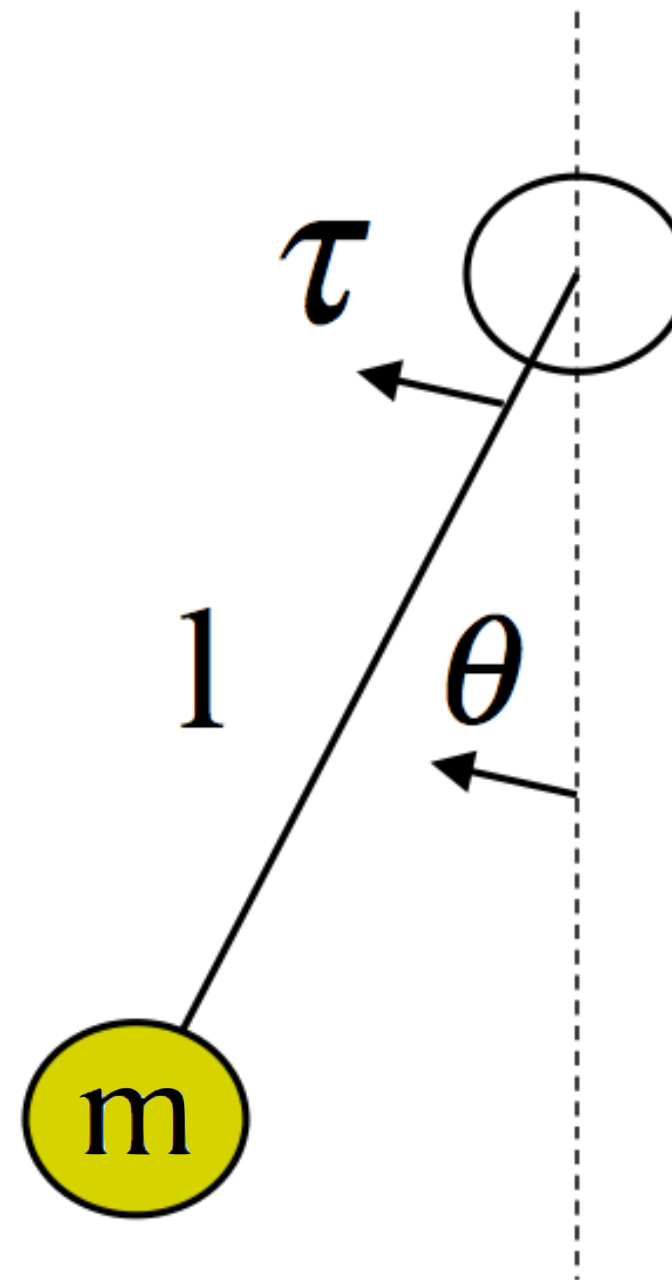
maintain desired angle (x_d)

apply motor force (u)

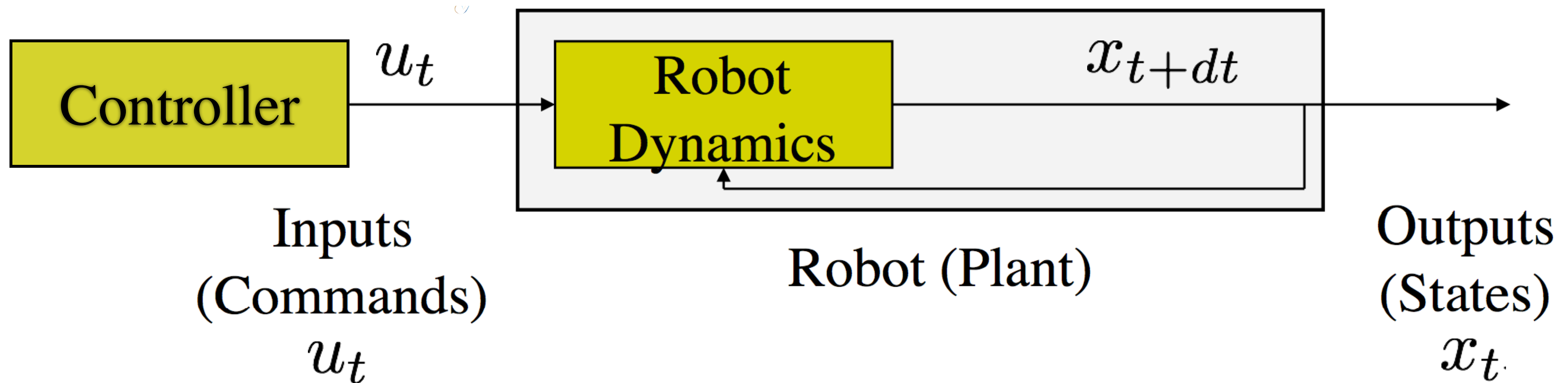
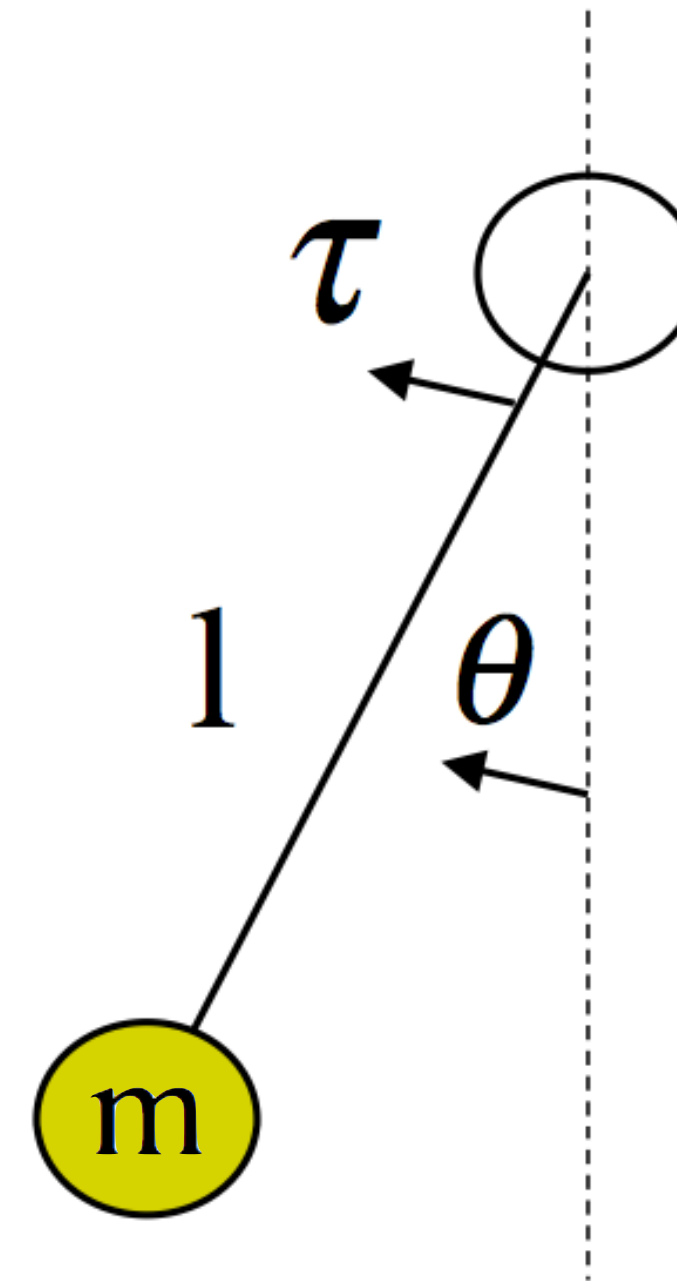


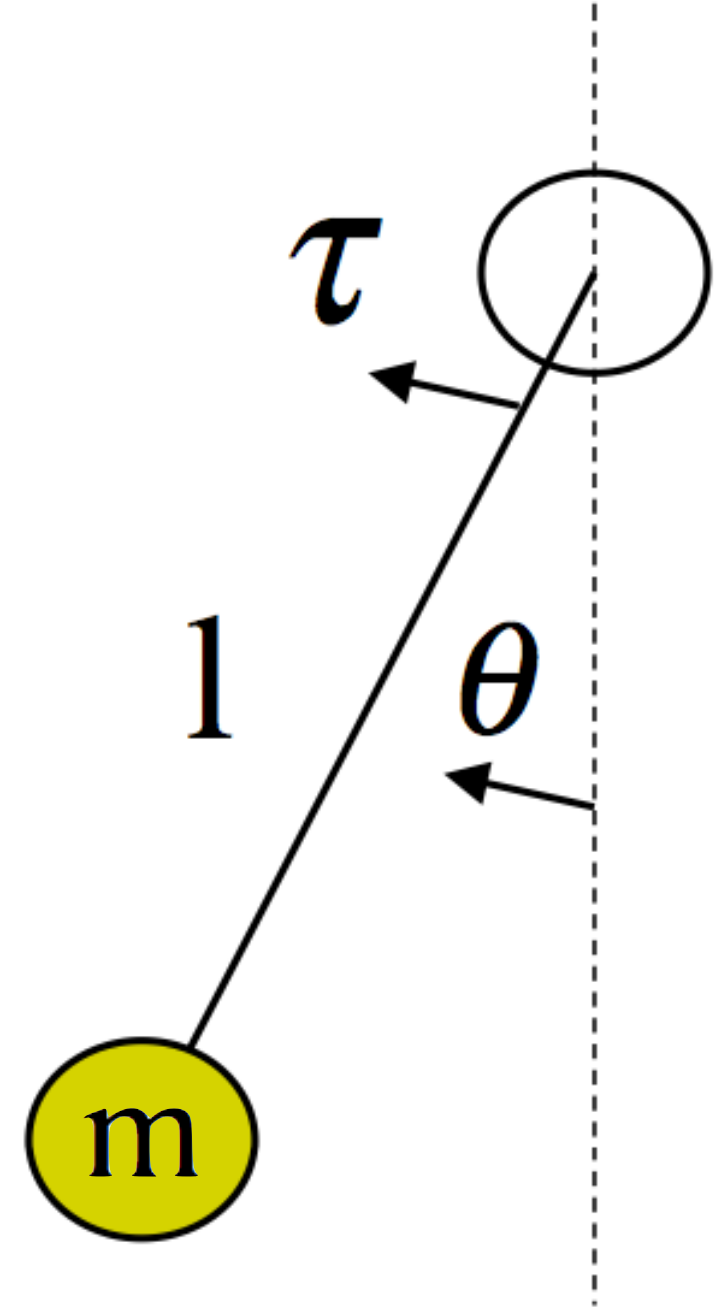
How do we implement a
servo controller?

Servo Control



Servo Control

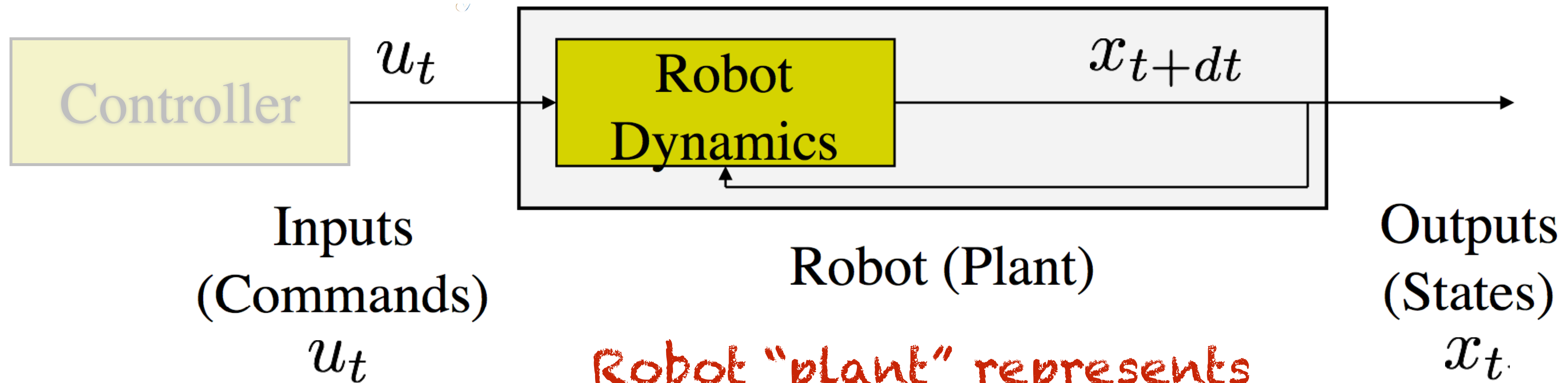




$$\ddot{\theta} = -\frac{g}{l}\sin(\theta) + \frac{\tau}{ml^2}$$

Motor command used within equations of motion

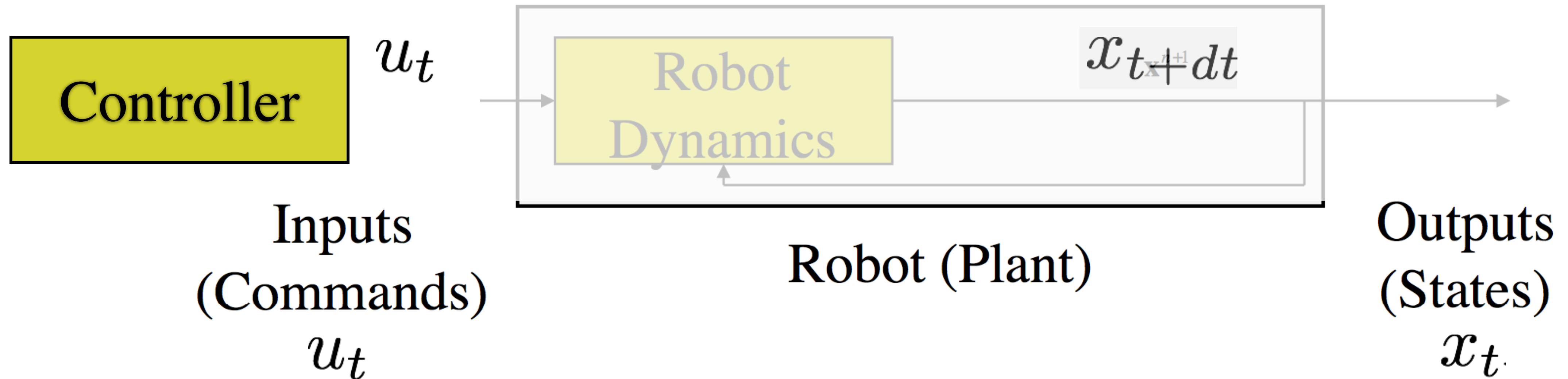
$$\dot{x} = f(t, x, u)$$



Robot "plant" represents system dynamics as function

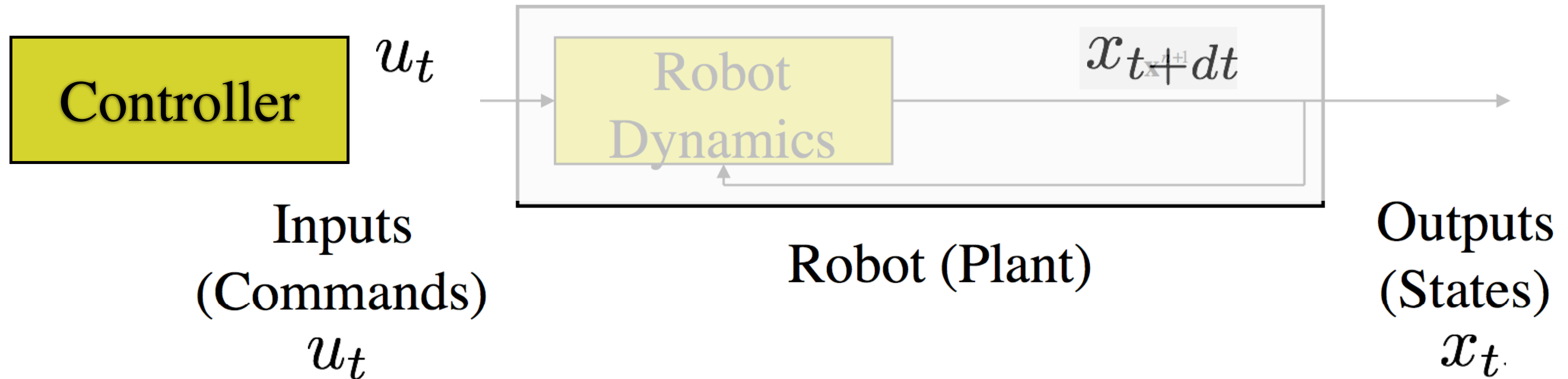
Servo controller generates
motor commands based
on desired state

$$u = \pi(t, x, x_d) \quad \dot{x} = f(t, x, u)$$



Servo controller generates
motor commands based
on desired state

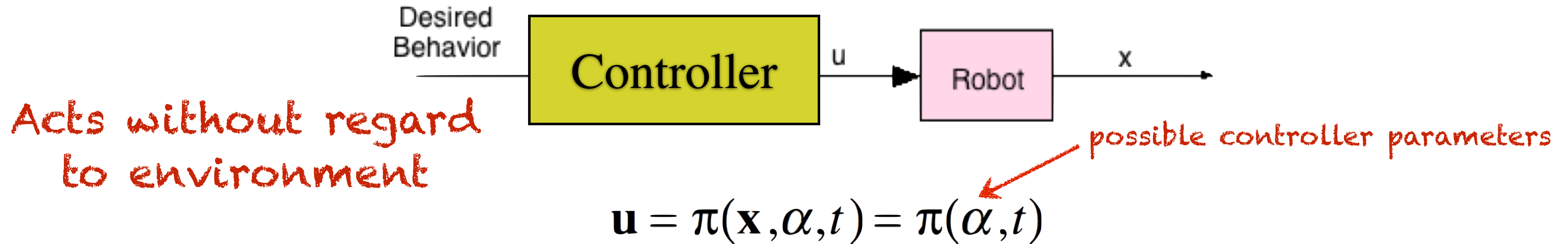
$$u = \pi(t, x, x_d) \quad \dot{x} = f(t, x, u)$$



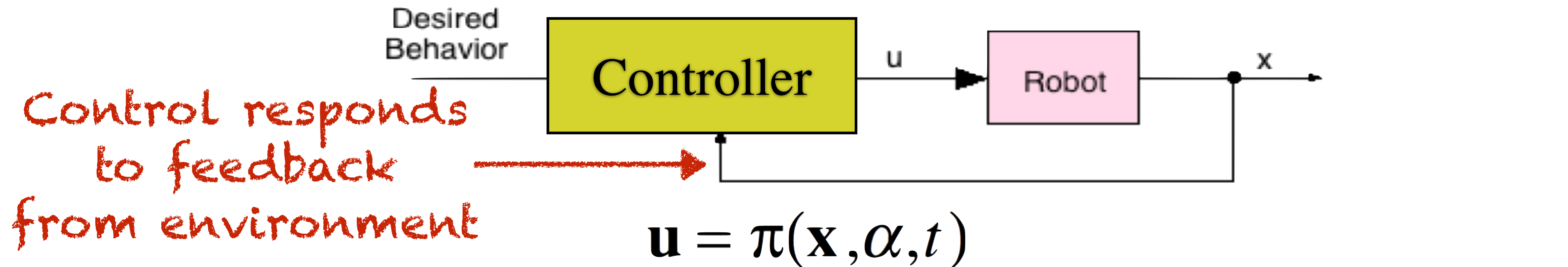
π is a "control policy"

Control Policies

Open Loop Control

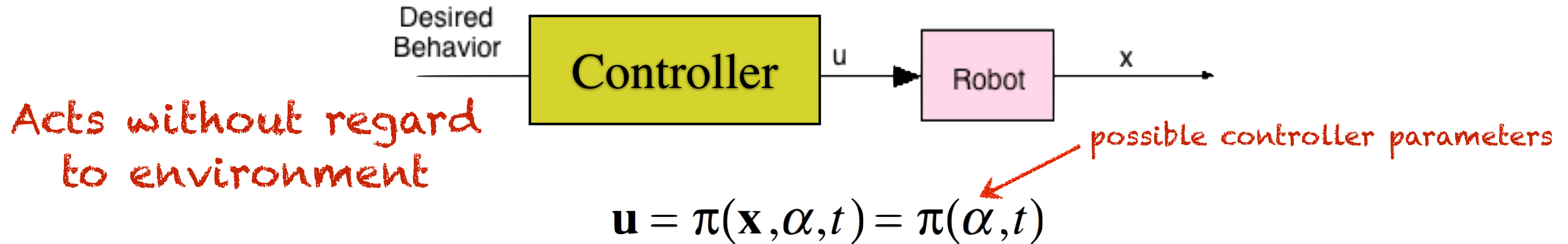


Closed Loop Control

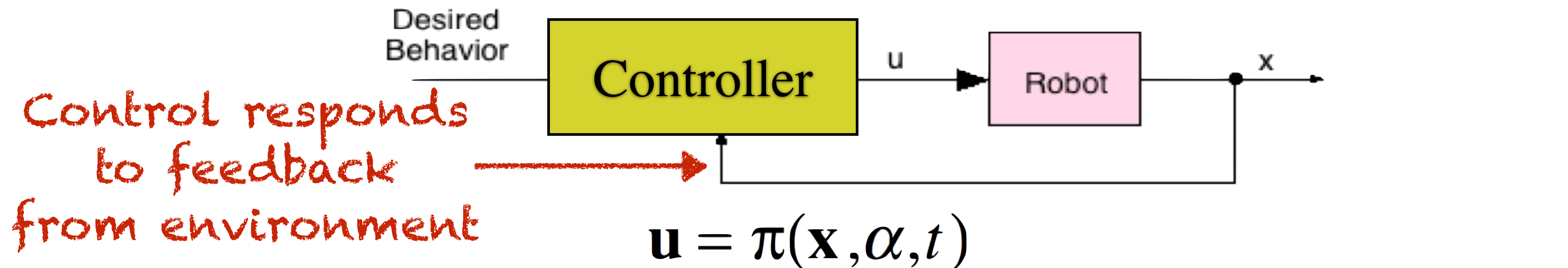


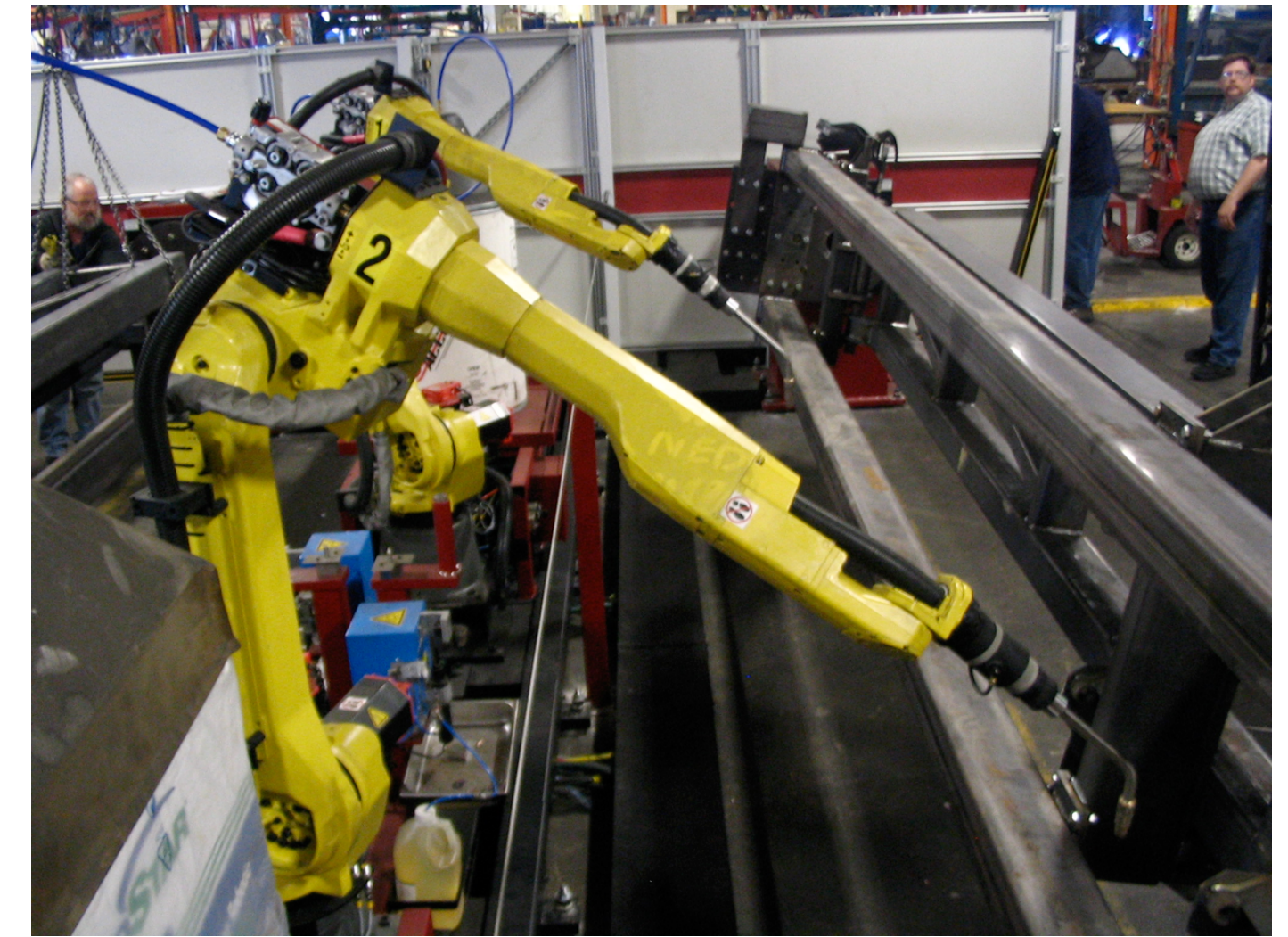
When to use open loop vs closed loop?

Open Loop Control

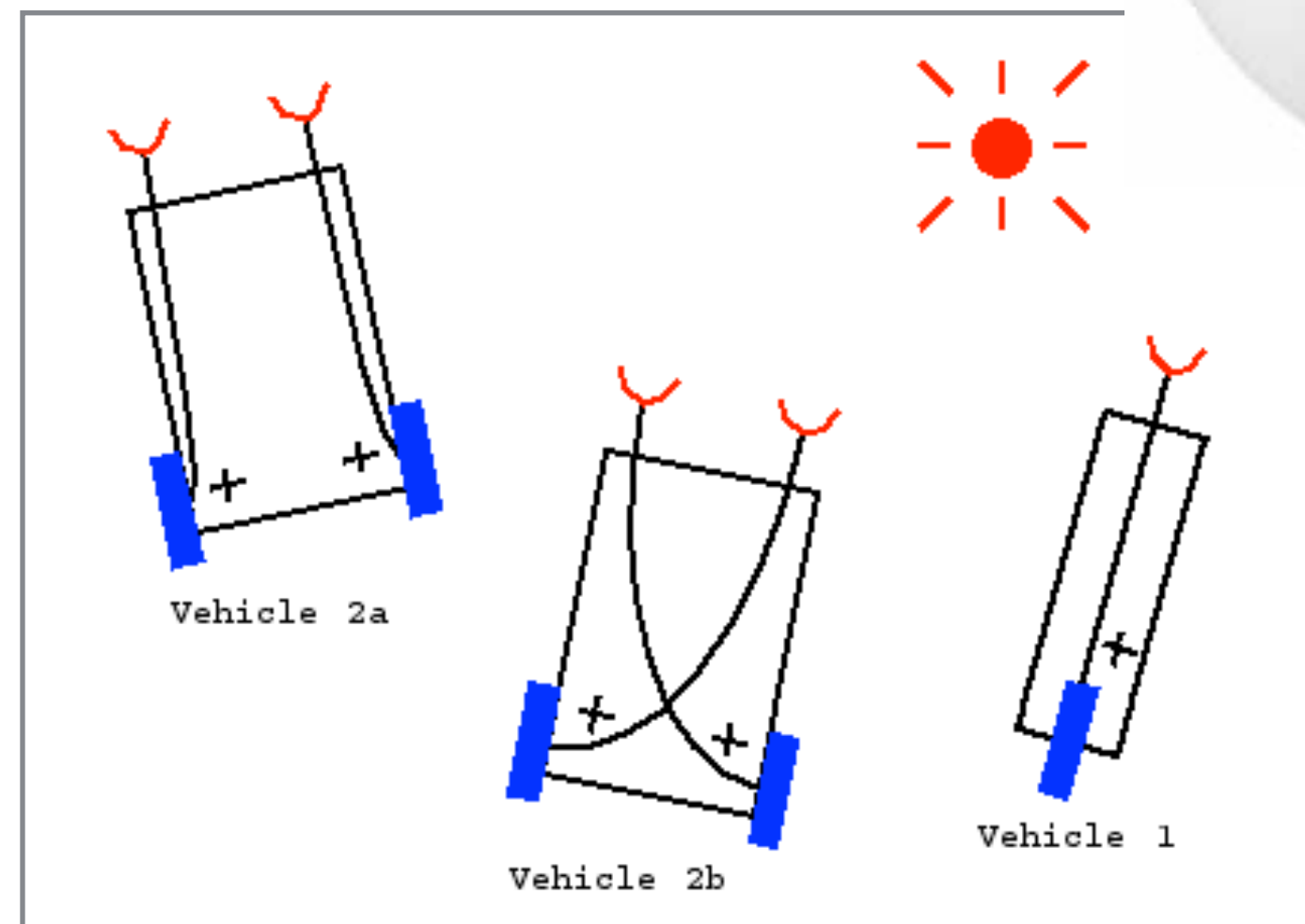


Closed Loop Control





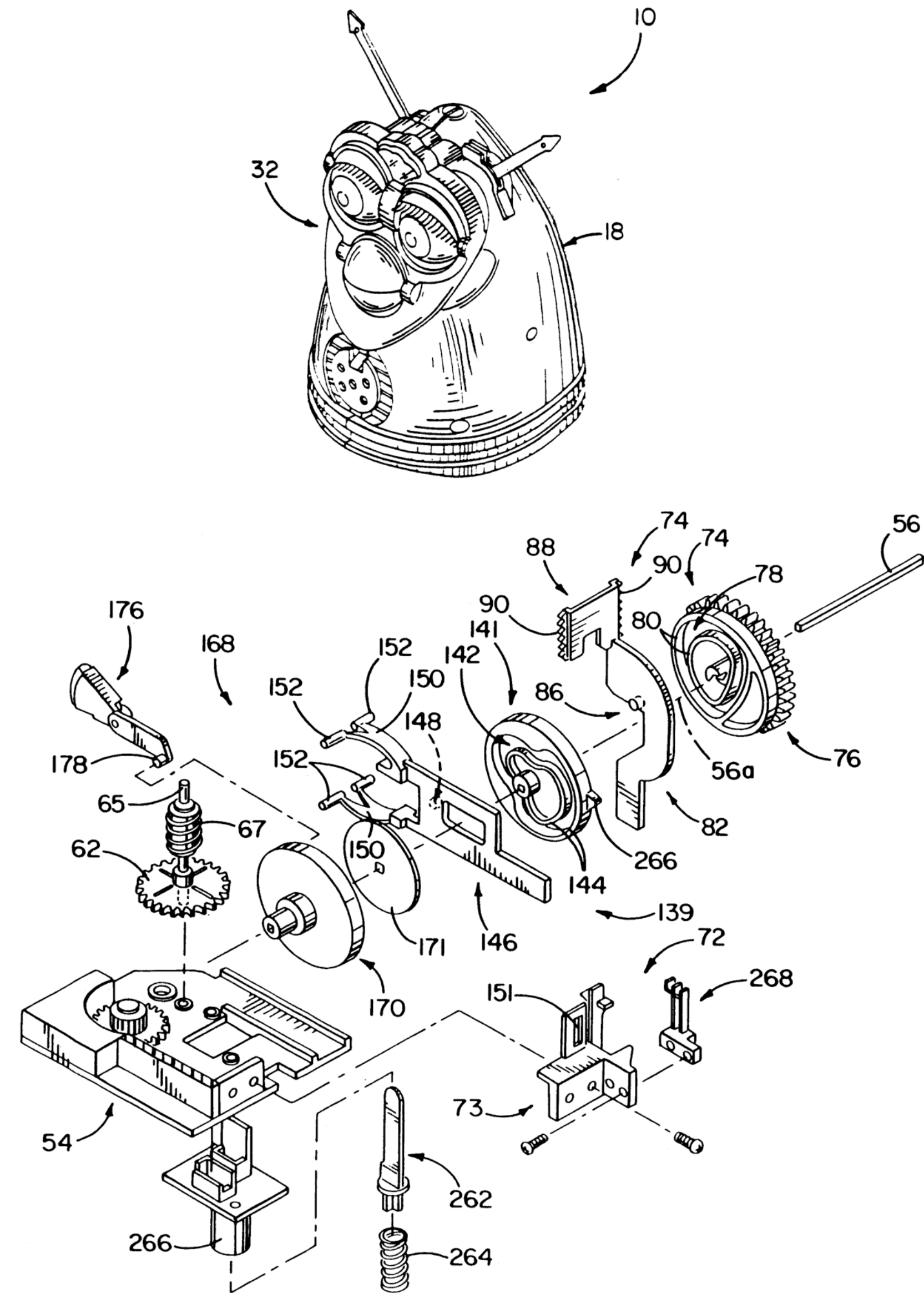
Which uses
open loop?





www.youtube.com/user/silvolf

Open loop: Furby





Disney

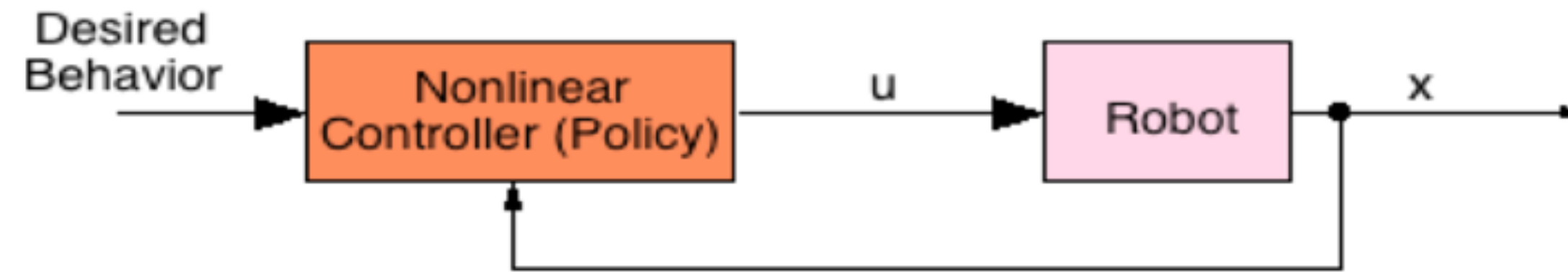


Michigan Robotics 367/320 - autorob.org

Types of Feedback Control

Types of Feedback Control

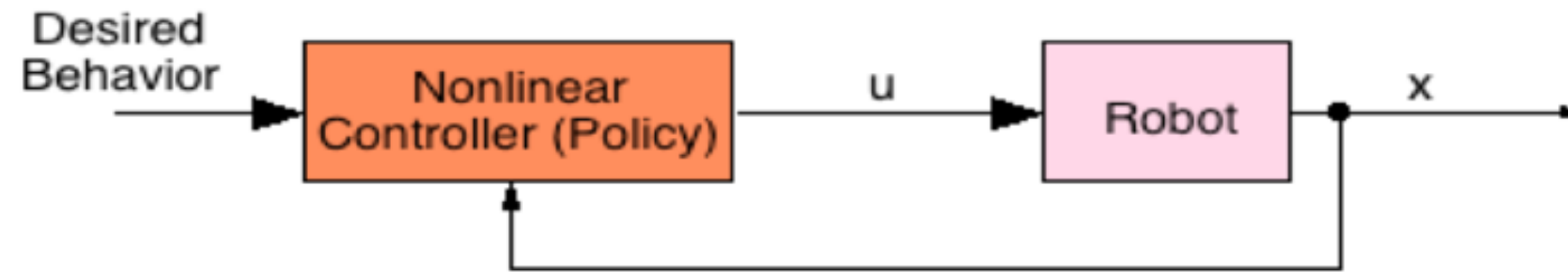
Feedback Control



$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t)$$

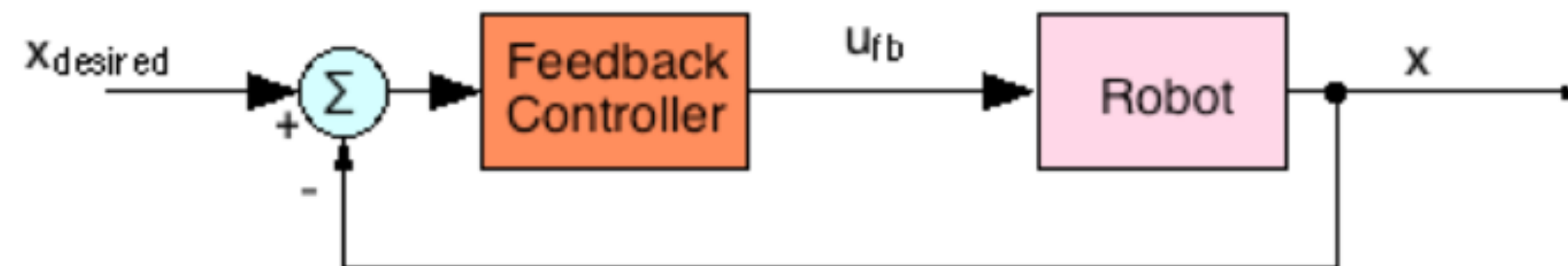
Types of Feedback Control

Feedback Control



$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t)$$

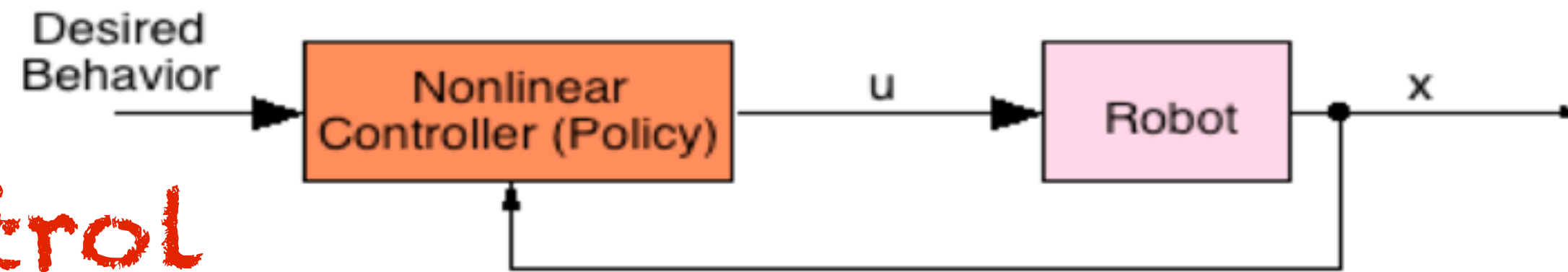
Negative Feedback Control



$$\mathbf{u} = \pi(\mathbf{x} - \mathbf{x}_{des}, \alpha, t)$$

Types of Feedback Control

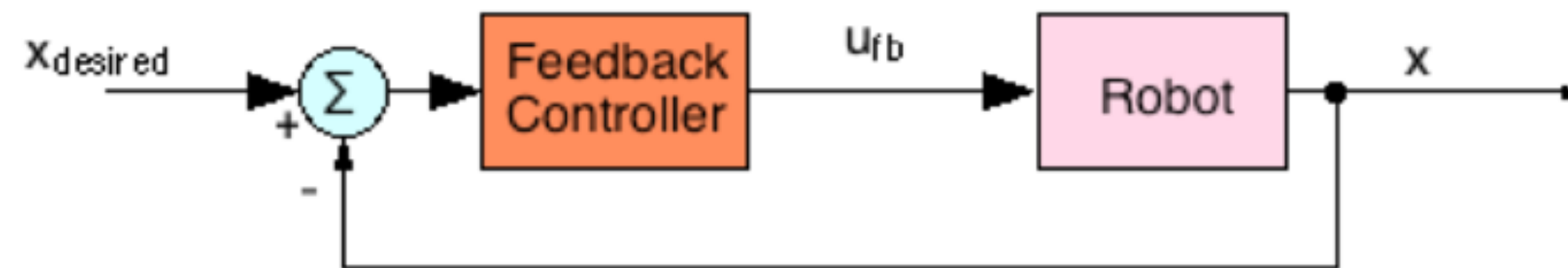
Feedback Control



$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t)$$

PID Control

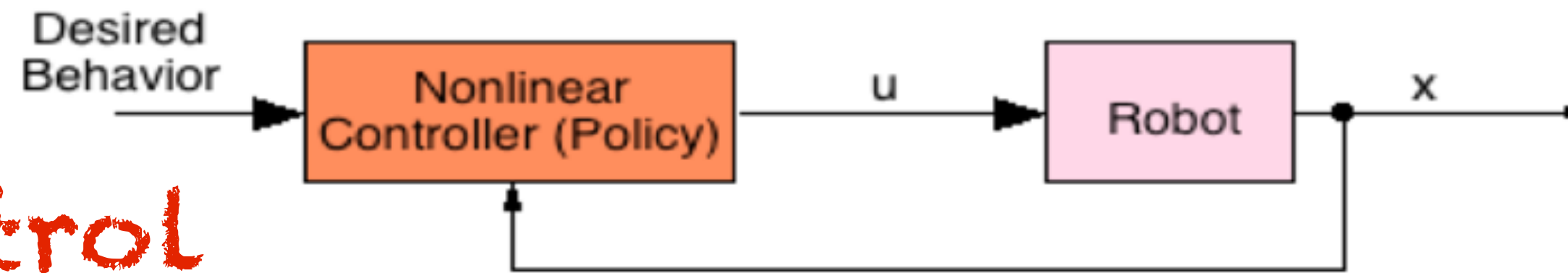
Negative Feedback Control



$$\mathbf{u} = \pi(\mathbf{x} - \mathbf{x}_{des}, \alpha, t)$$

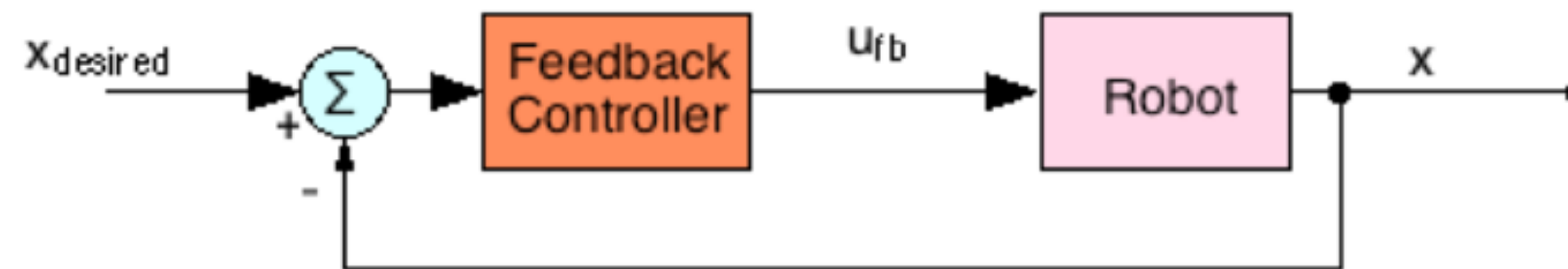
Types of Feedback Control

Feedback Control



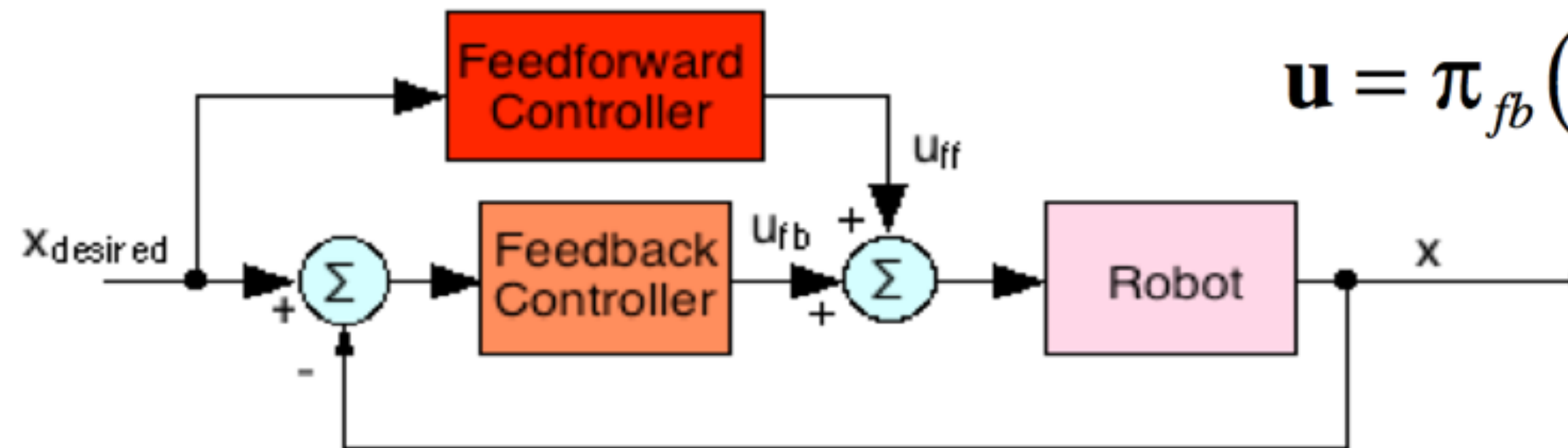
$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t)$$

Negative Feedback Control



$$\mathbf{u} = \pi(\mathbf{x} - \mathbf{x}_{des}, \alpha, t)$$

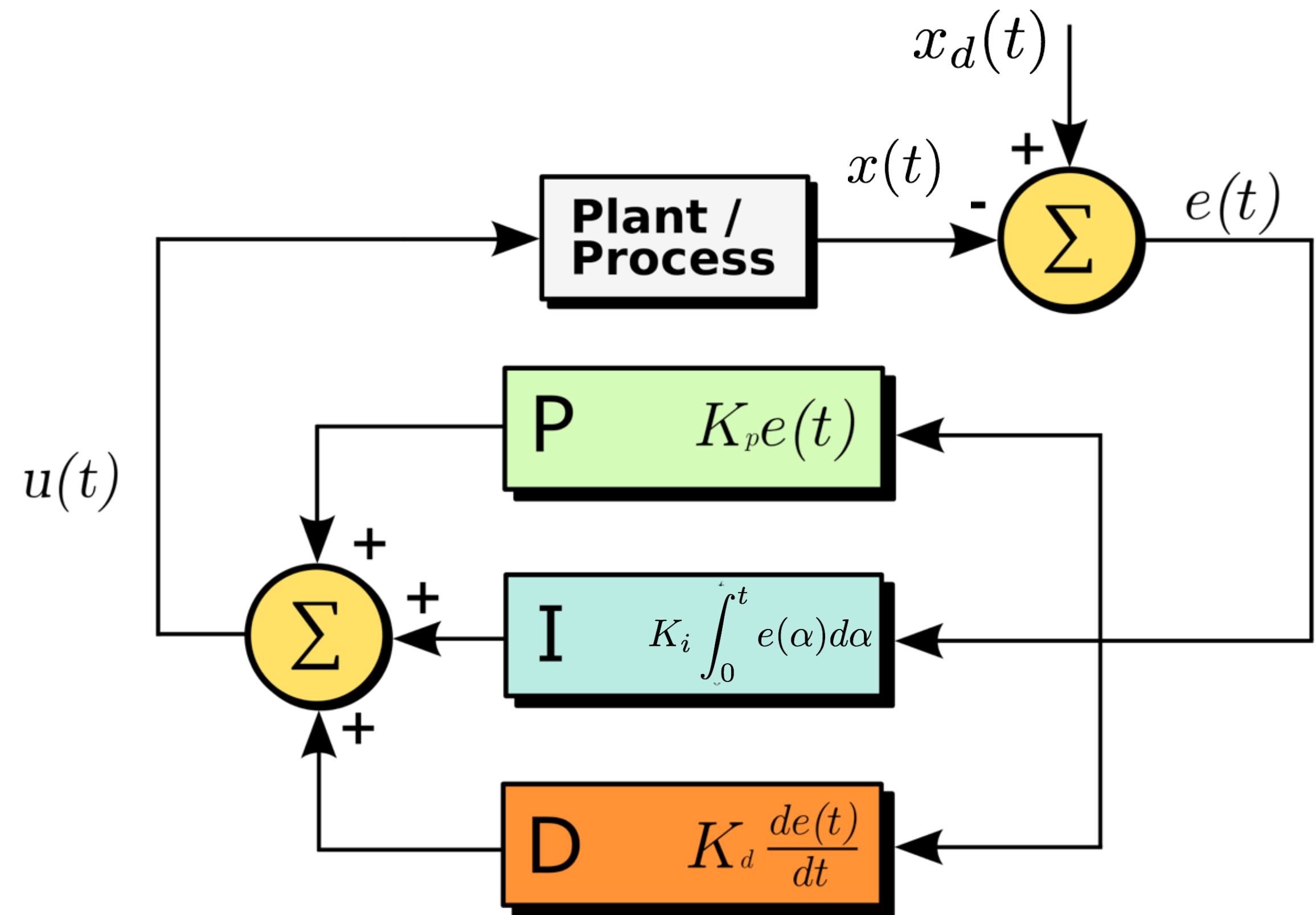
Neg. Feedback & Feedforward Control



$$\mathbf{u} = \pi_{fb}(\mathbf{x} - \mathbf{x}_{des}, \alpha, t) + \pi_{ff}(\mathbf{x}_{des}, \alpha, t)$$

PID Control

- Proportional-Integral-Derivative Control
- Sum of different responses to error
- Based on the mass spring and damper system
- Feedback correction based on the current error, past error, and predicted future error



PID Control

for pendulum (x is θ)

Error signal:

$$e(t) = x_{desired}(t) - x(t)$$

Control signal:

$$u(t) = K_p e(t) + K_i \int_0^t e(\alpha) d\alpha + K_d \frac{d}{dt} e(t)$$

P $K_p e(t)$

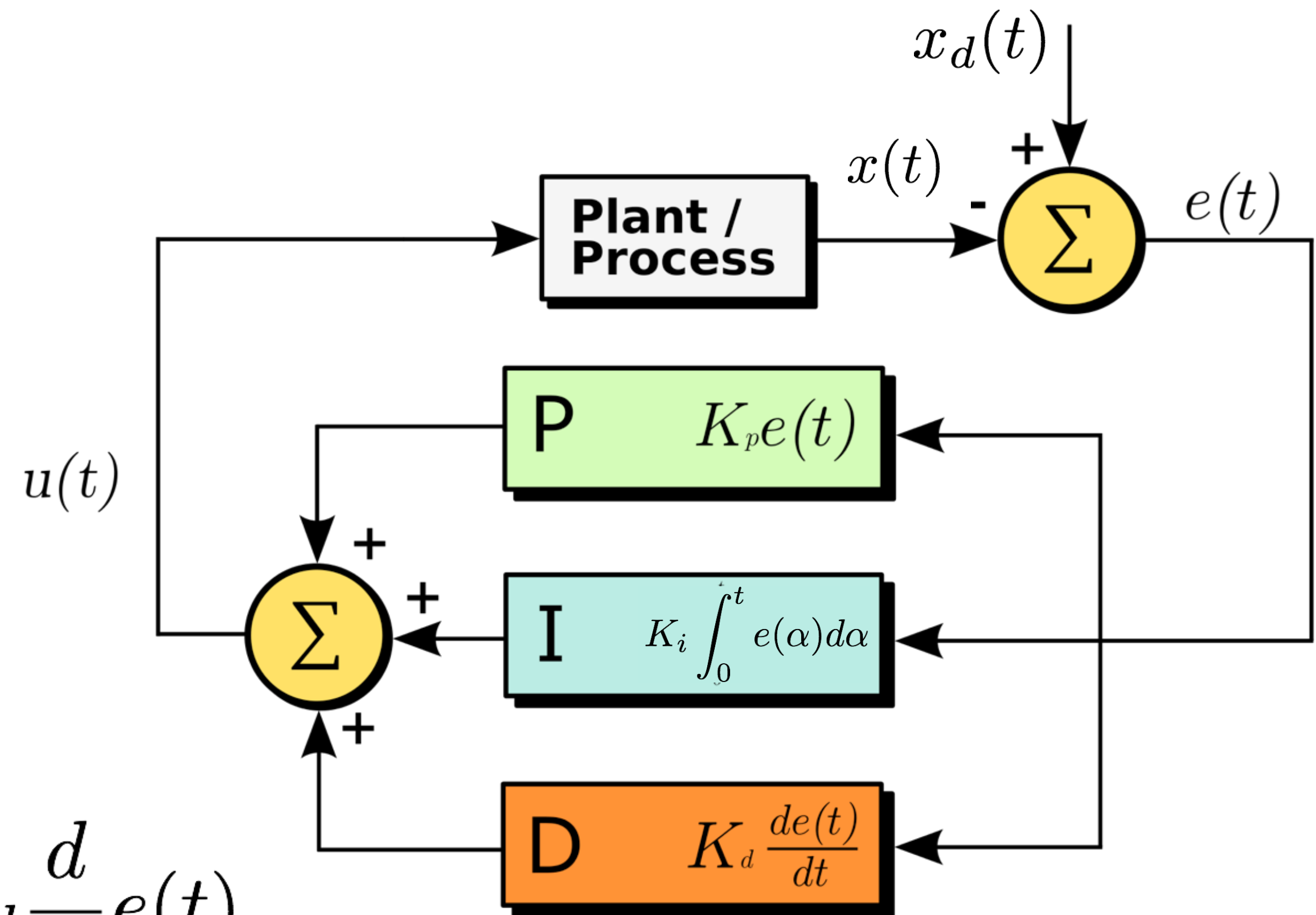
Current

I $K_i \int_0^t e(\alpha) d\alpha$

Past

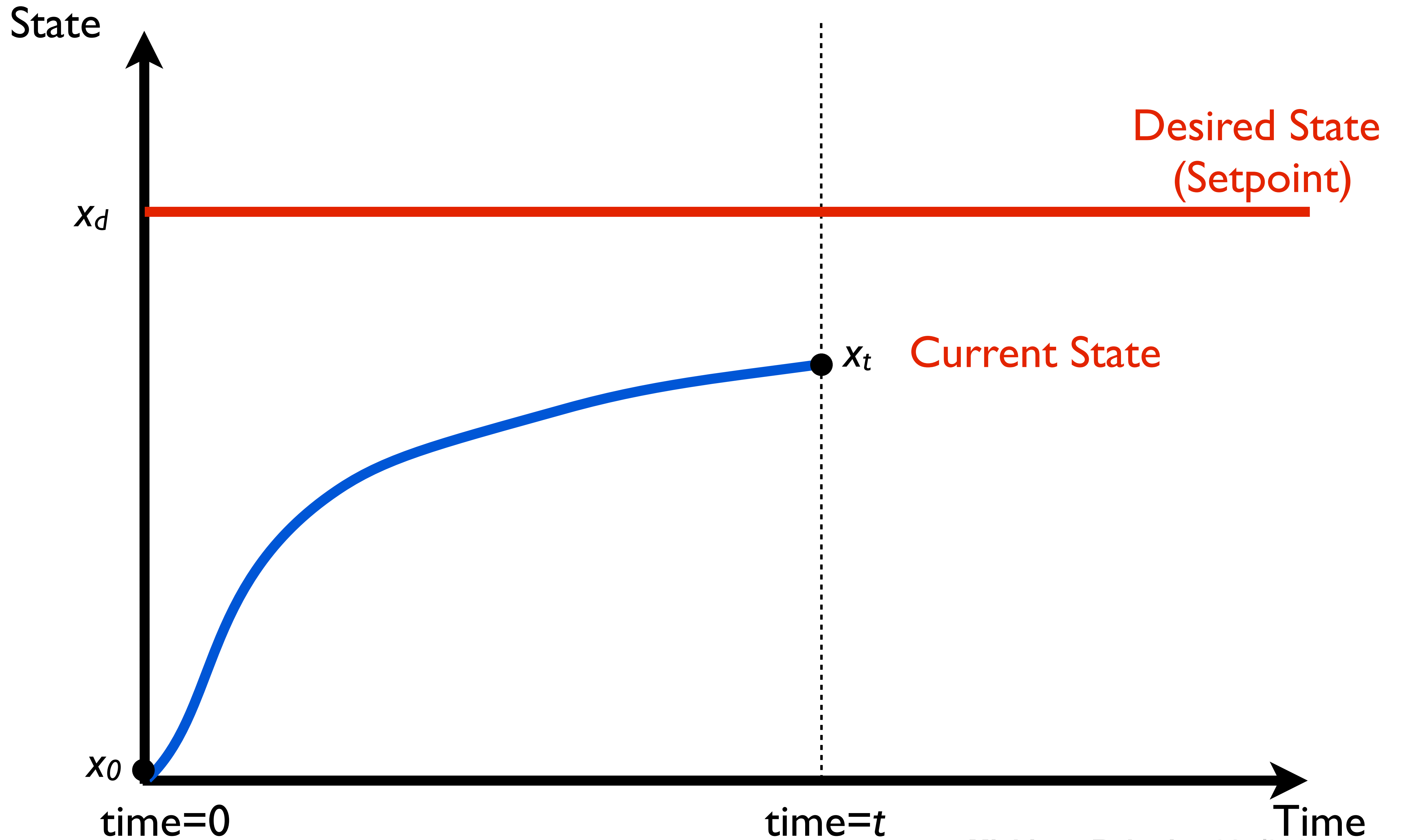
D $K_d \frac{de(t)}{dt}$

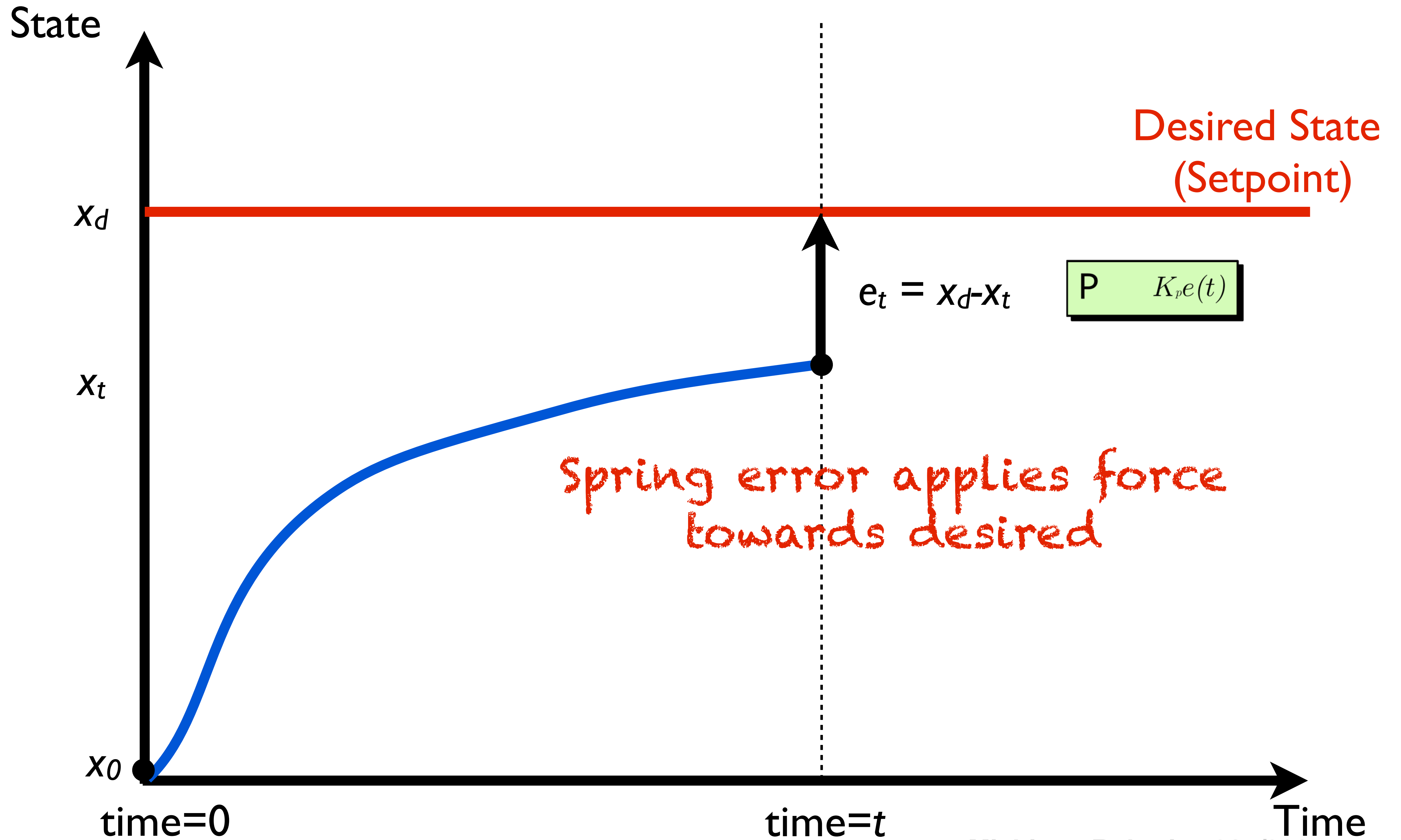
Future

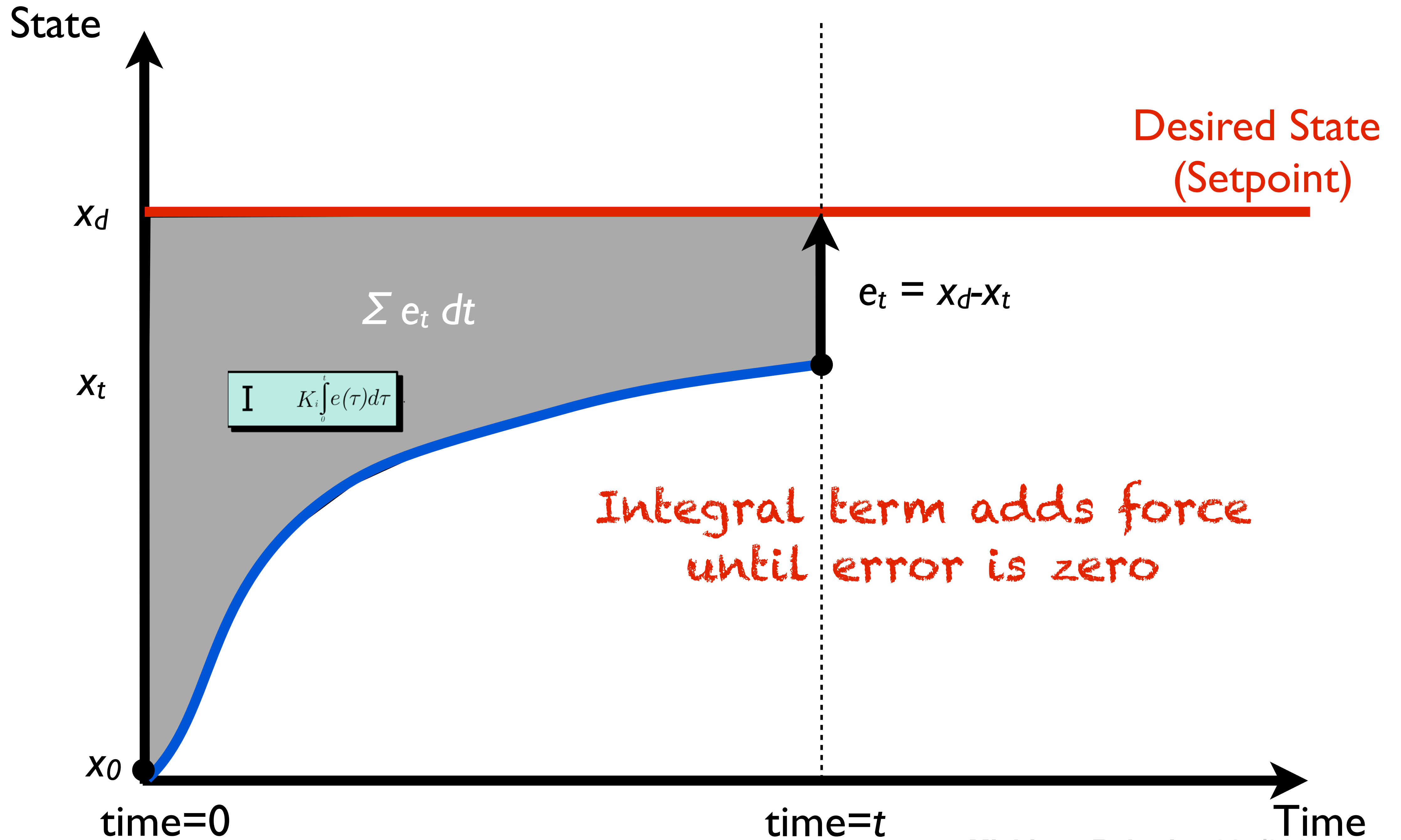


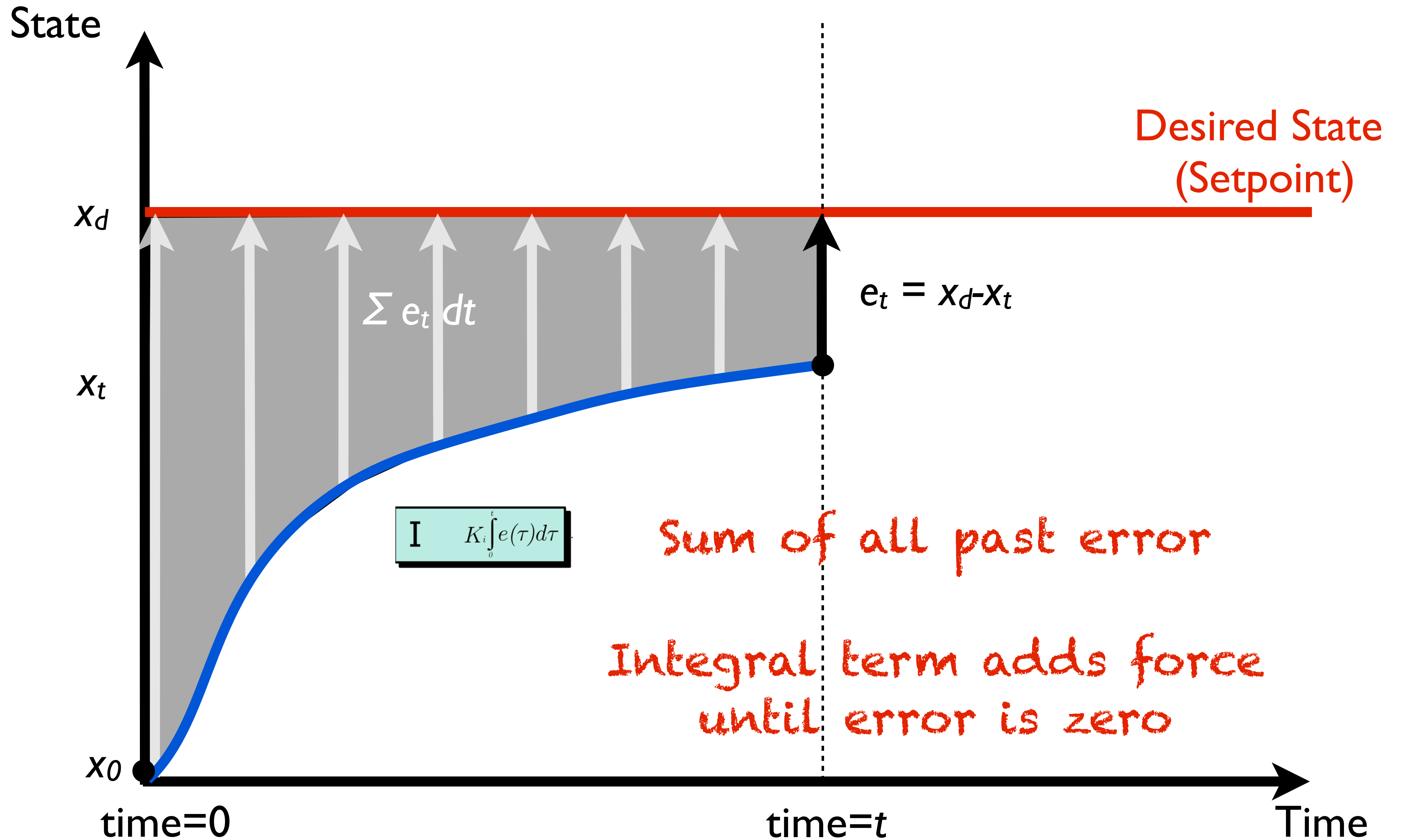
Consider PID wrt. state over time

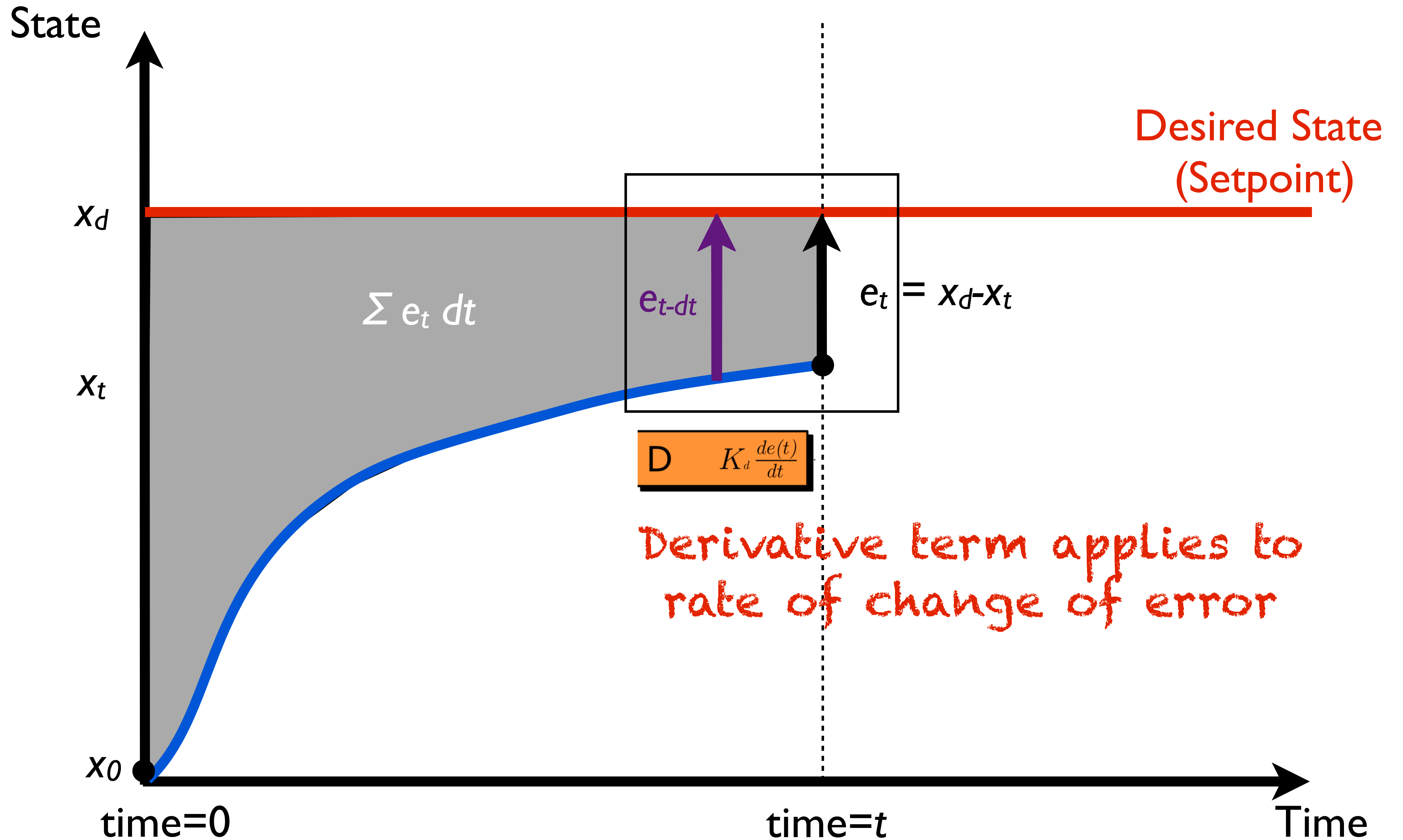


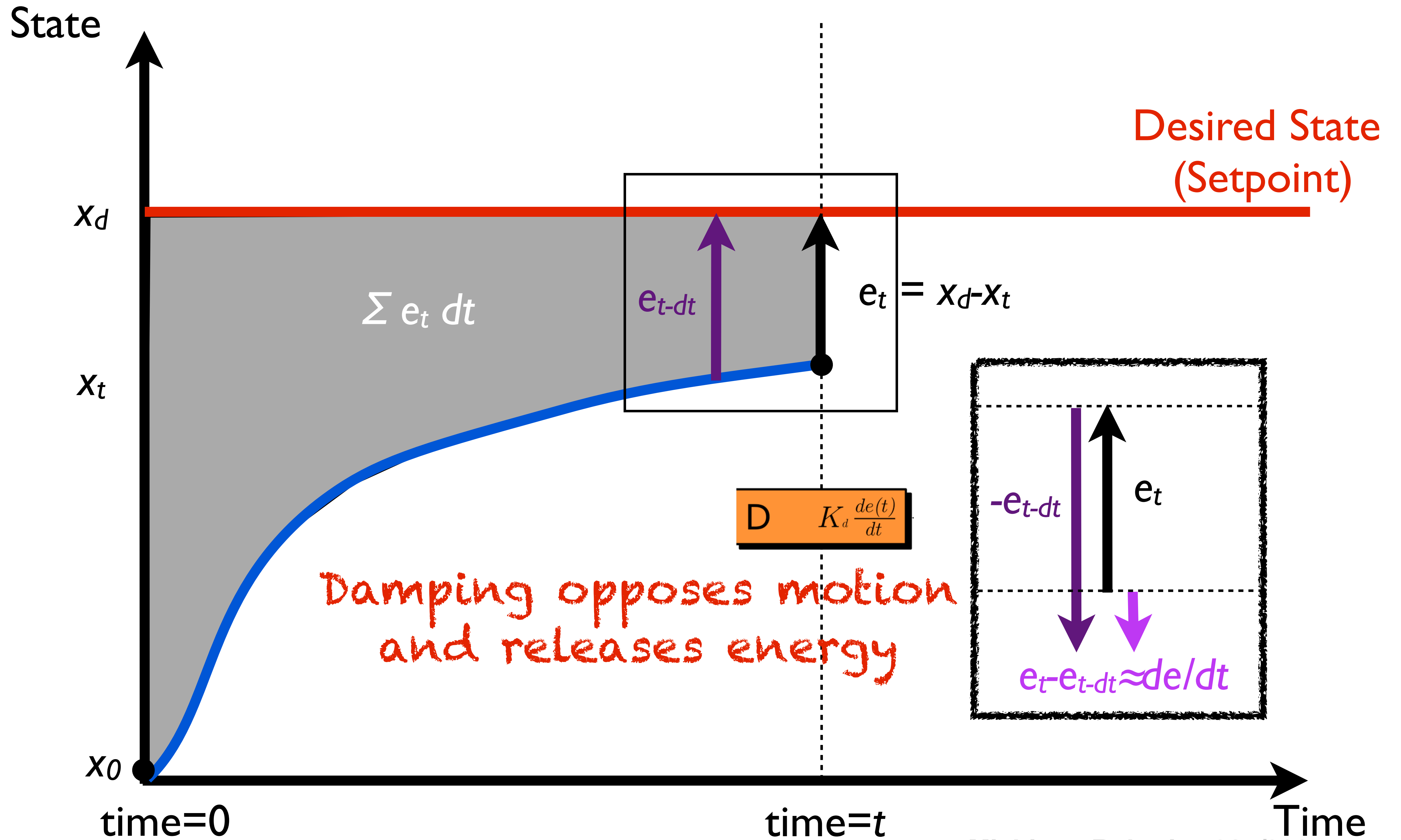


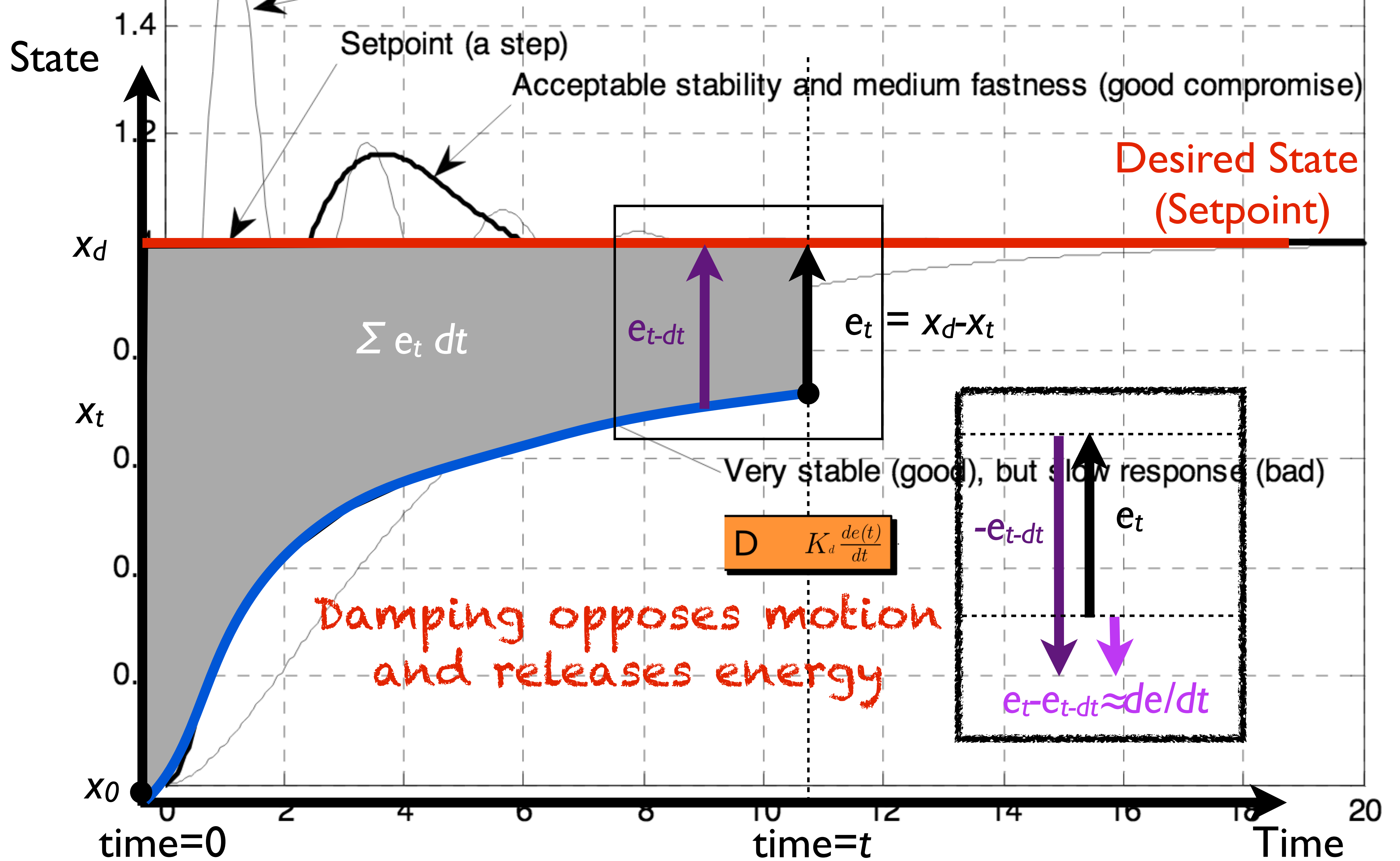




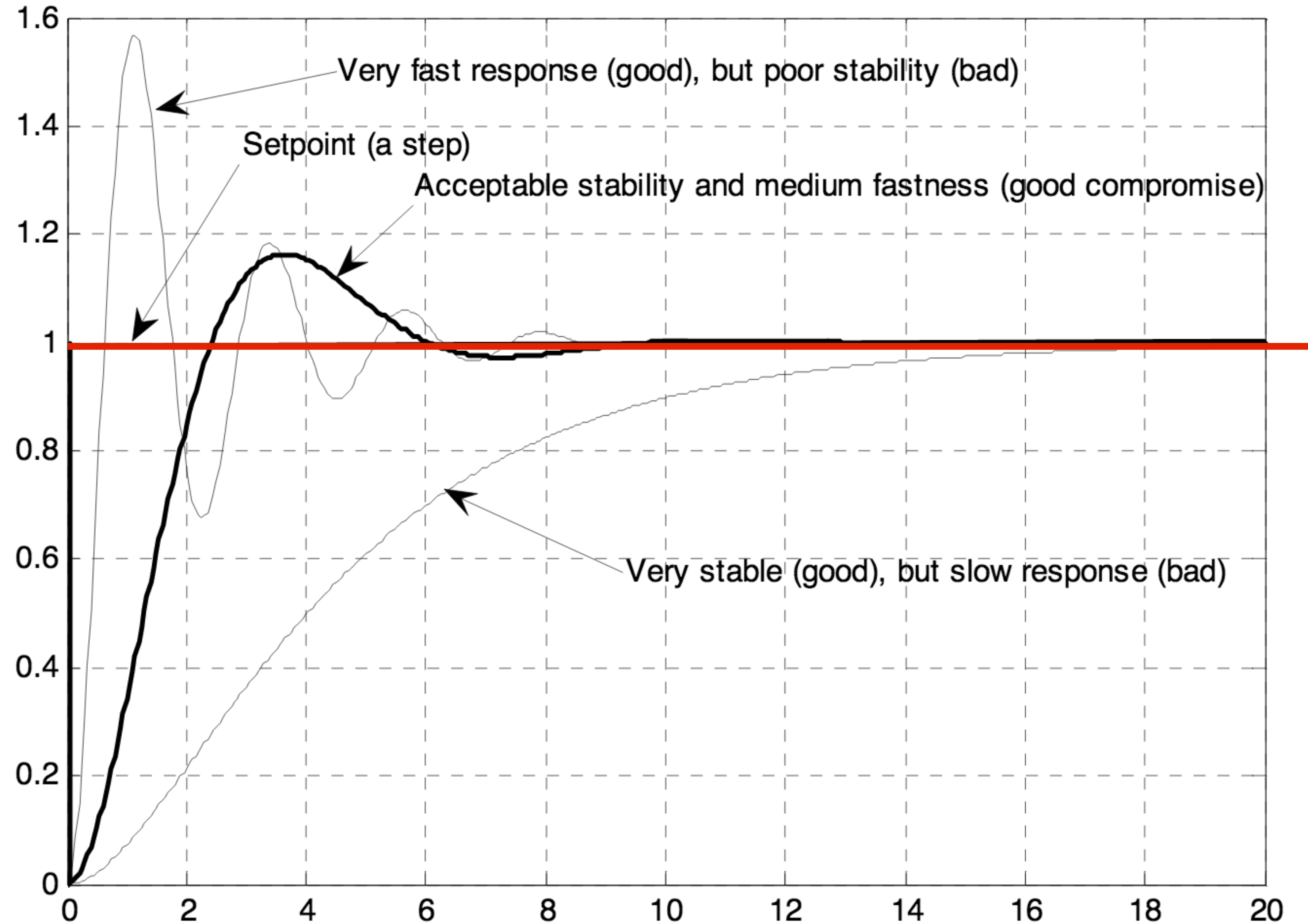








PID Converge



PID as a spring and damper model

PID Control

Error signal:

$$e(t) = x_{desired}(t) - x(t)$$

Control signal:

$$u(t) = K_p e(t) + K_i \int_0^t e(\alpha) d\alpha + K_d \frac{d}{dt} e(t)$$

$$\text{P} \quad K_p e(t)$$

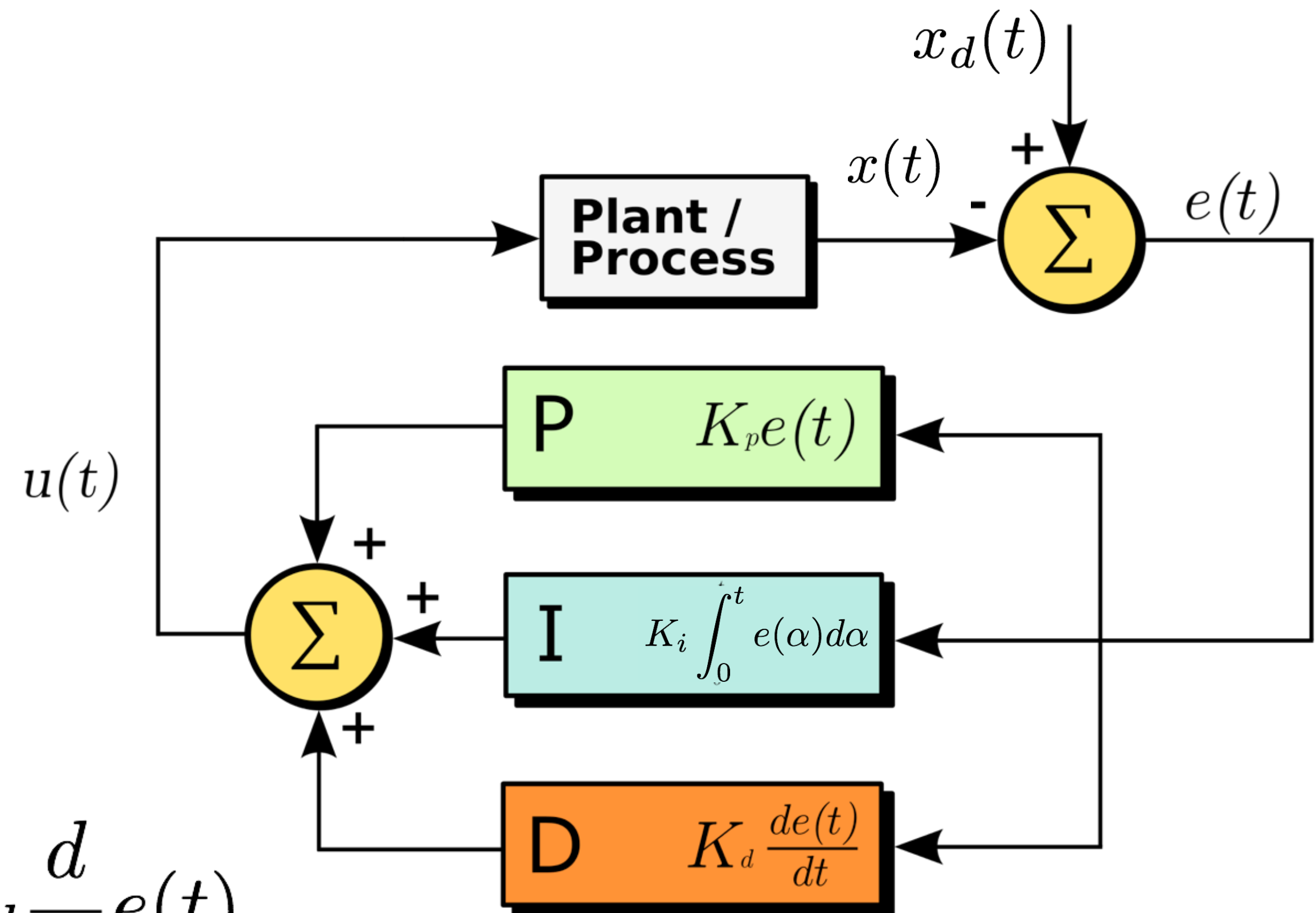
Current

$$\text{I} \quad K_i \int_0^t e(\alpha) d\alpha$$

Past

$$\text{D} \quad K_d \frac{de(t)}{dt}$$

Future



Hooke's Law

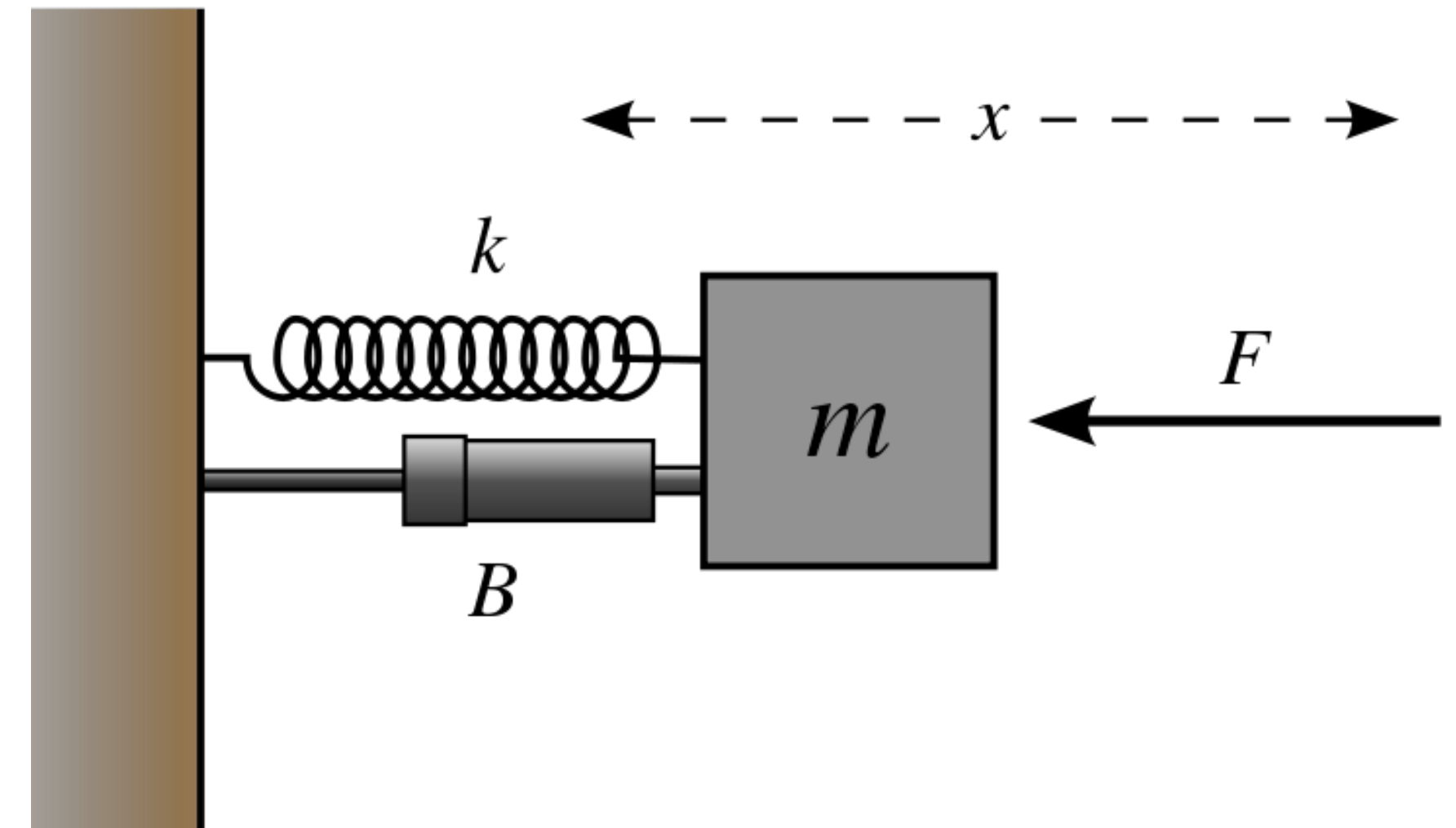
P $K_p e(t)$

- Describes motion of mass spring damper system as

$$F = -kx$$



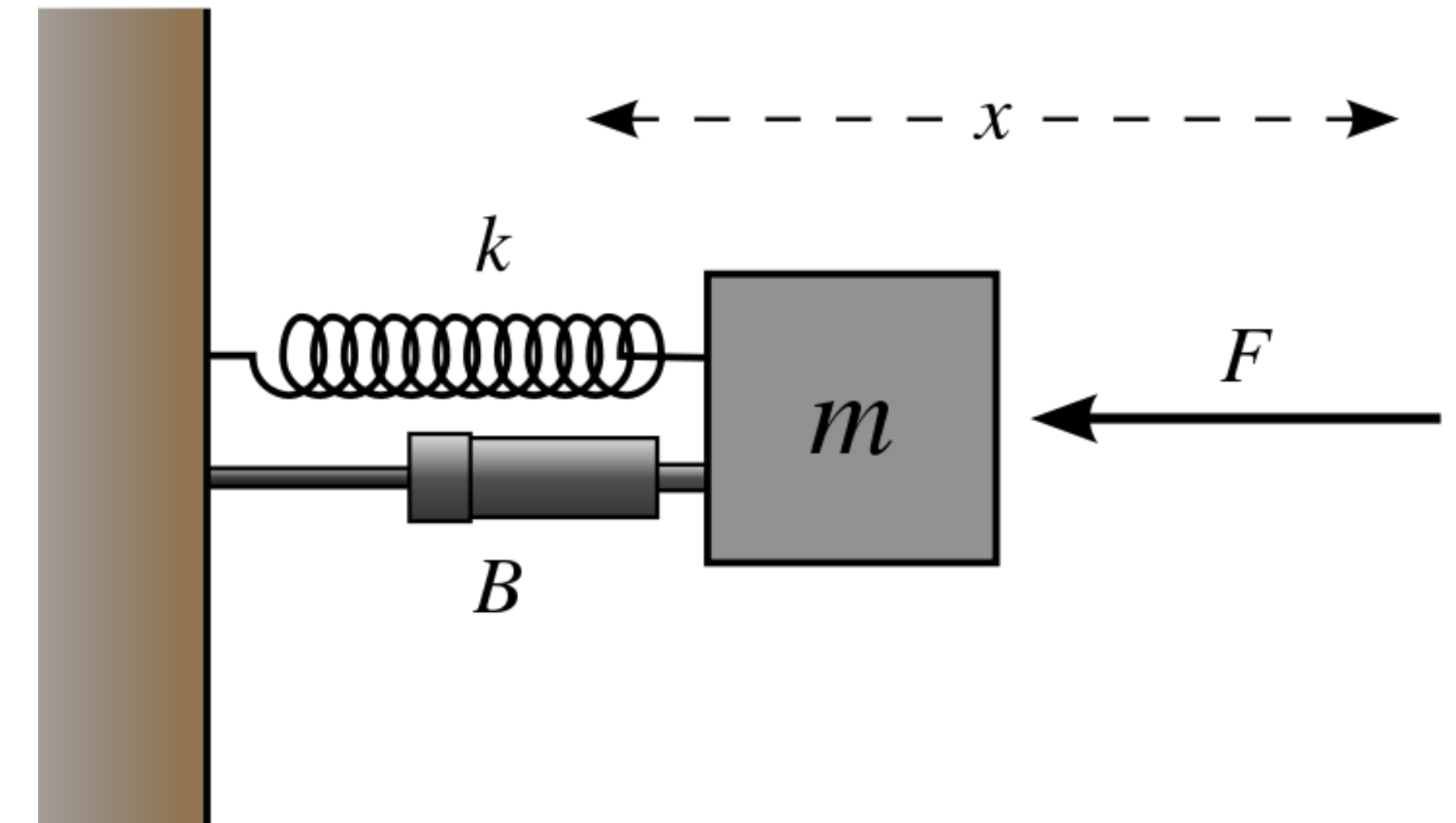
Robert Hooke
(1635-1703)



Hooke's Law

P $K_p e(t)$

- Describes motion of mass spring damper system as



$$F = -kx$$

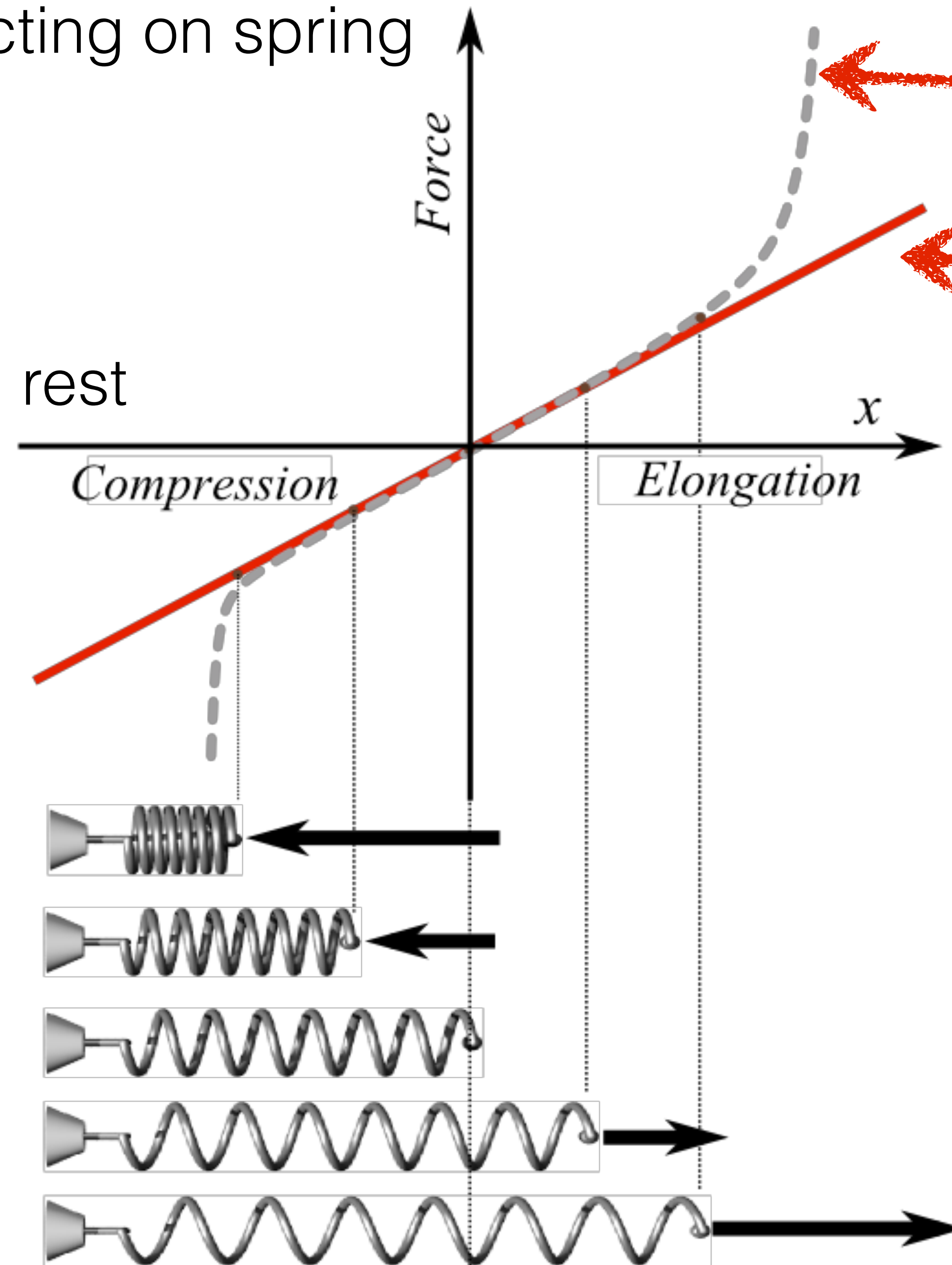
force moving
spring towards rest

spring
stiffness

distance from
rest displacement

Vertical: Force acting on spring

Horizontal: Position from rest

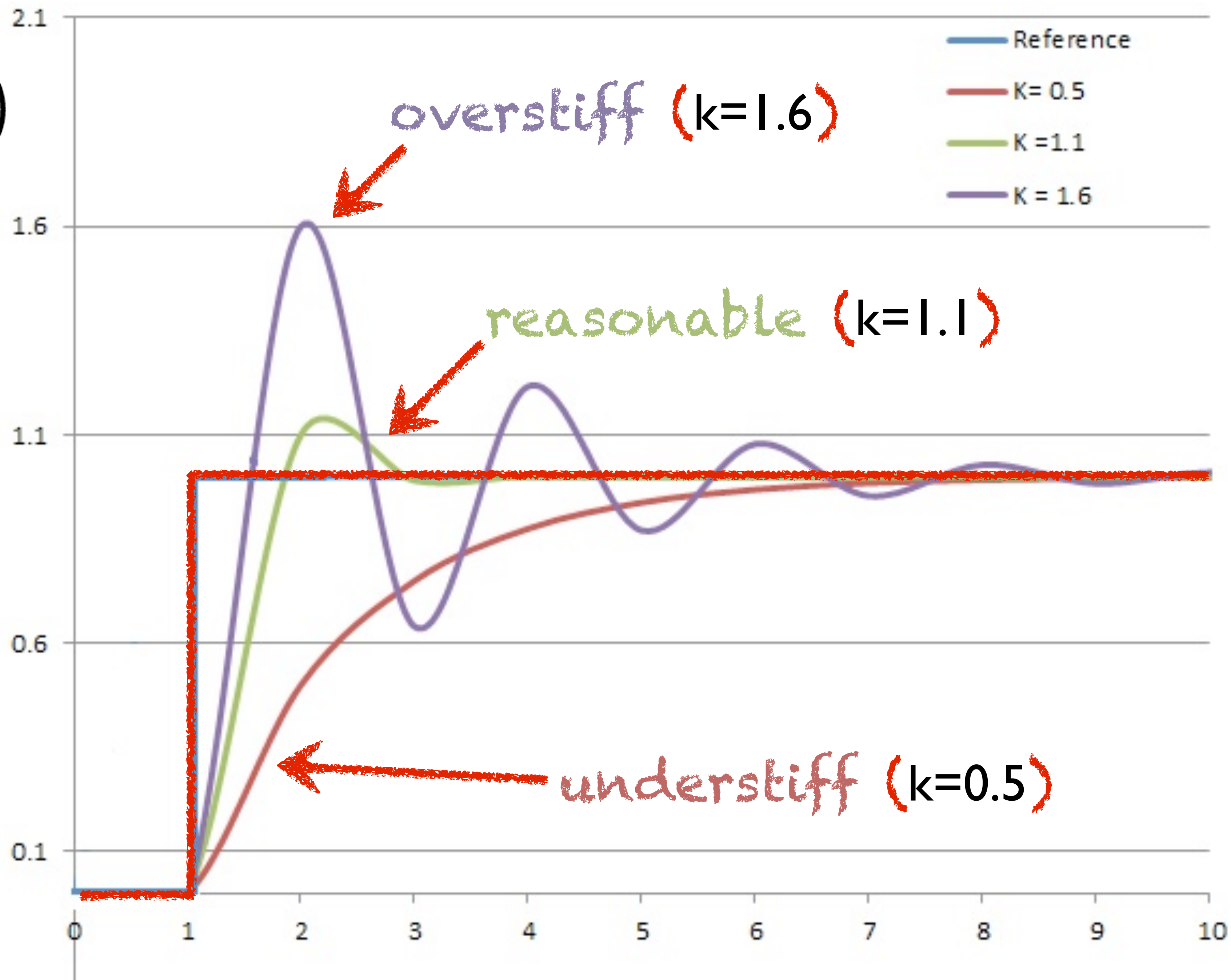


actual force relation

Linear approximation ($-kx$)

$$K_p e(t)$$

Position



Time

PID Control

Error signal:

$$e(t) = x_{desired}(t) - x(t)$$

Control signal:

$$u(t) = K_p e(t) + K_i \int_0^t e(\alpha) d\alpha + K_d \frac{d}{dt} e(t)$$

$$\text{P} \quad K_p e(t)$$

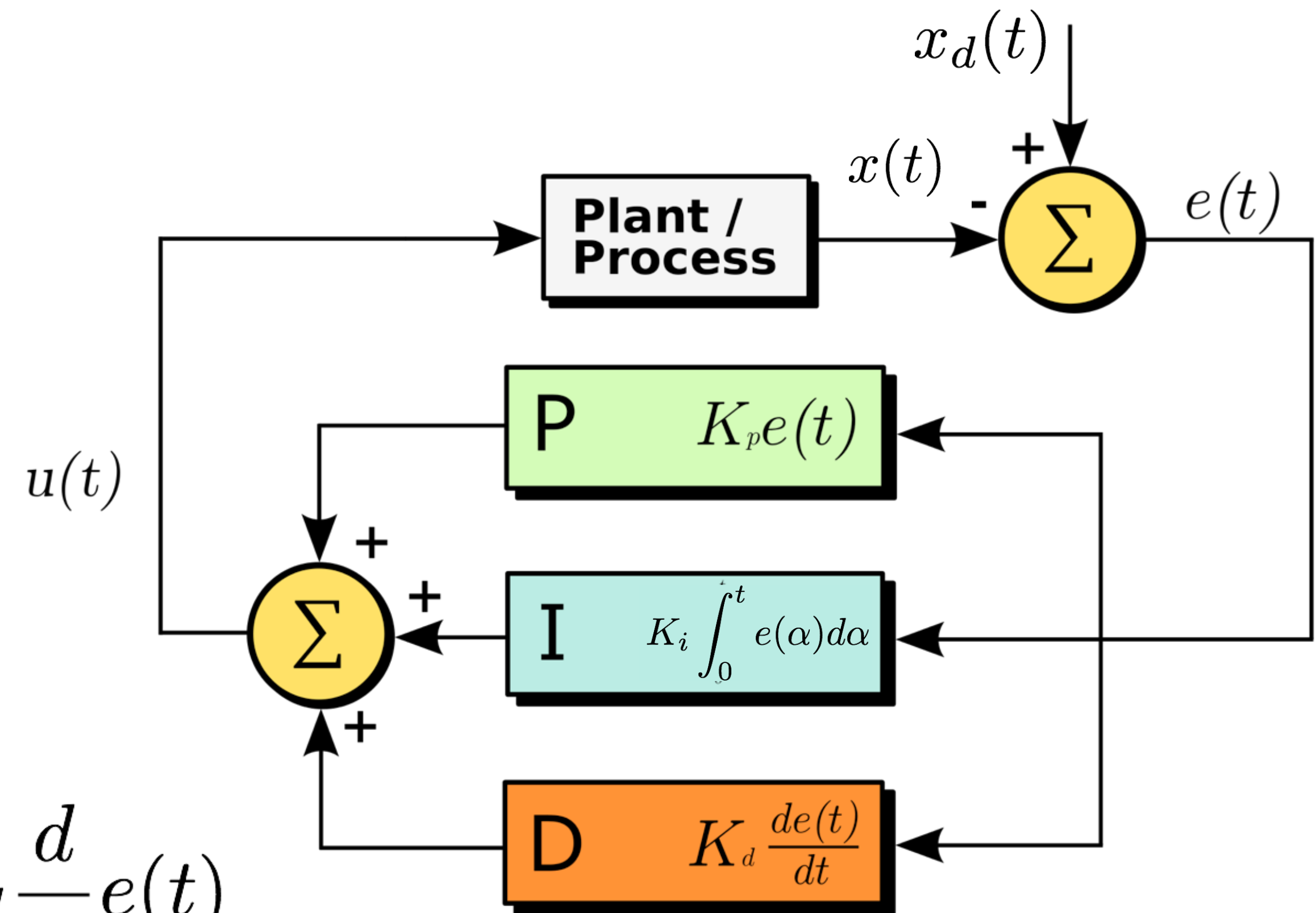
Current

$$\text{I} \quad K_i \int_0^t e(\alpha) d\alpha$$

Past

$$\text{D} \quad K_d \frac{de(t)}{dt}$$

Future

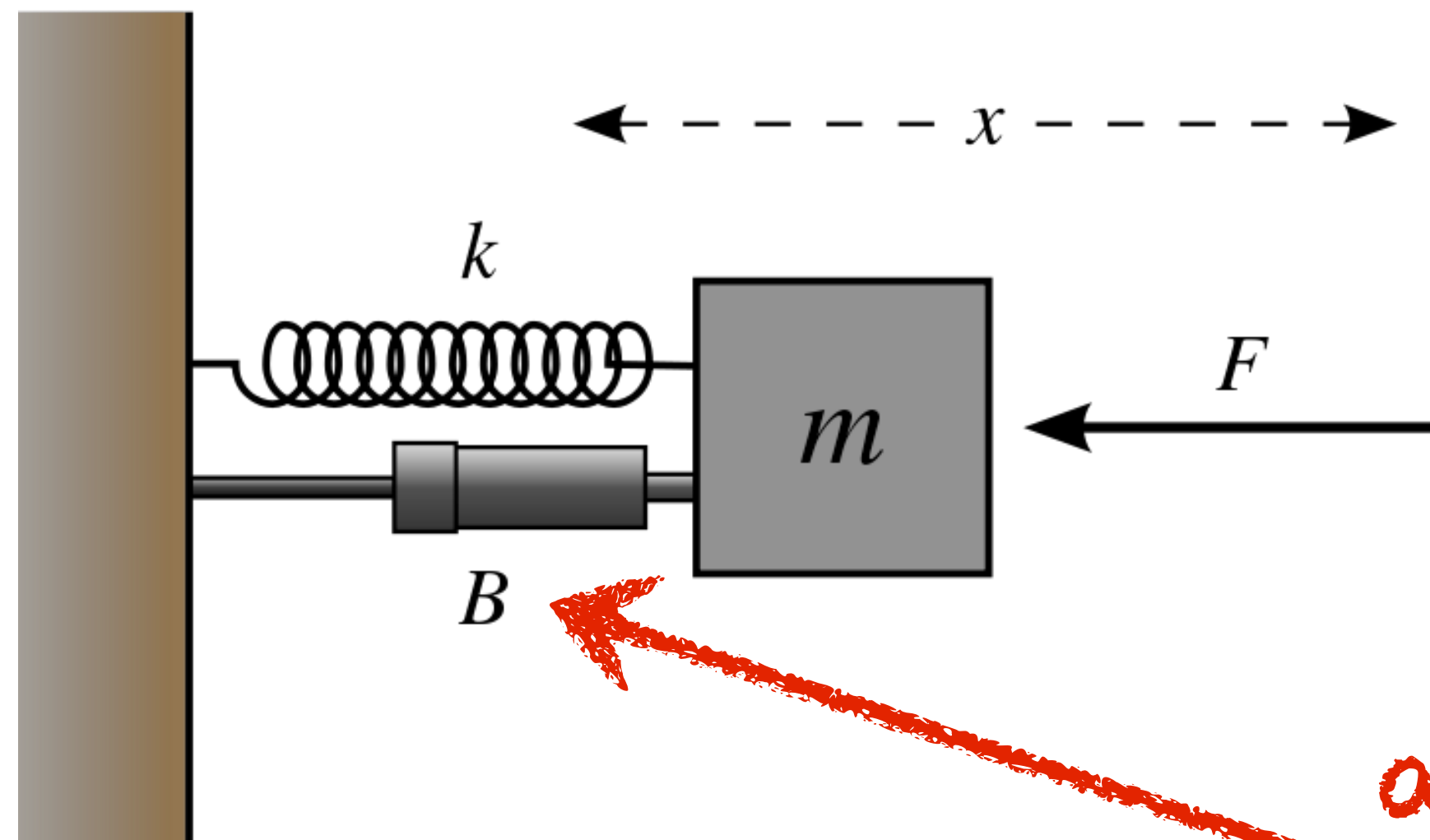


Spring and Damper

P $K_p e(t)$

D $K_d \frac{de(t)}{dt}$

$$F = -kx + -b\dot{x}$$

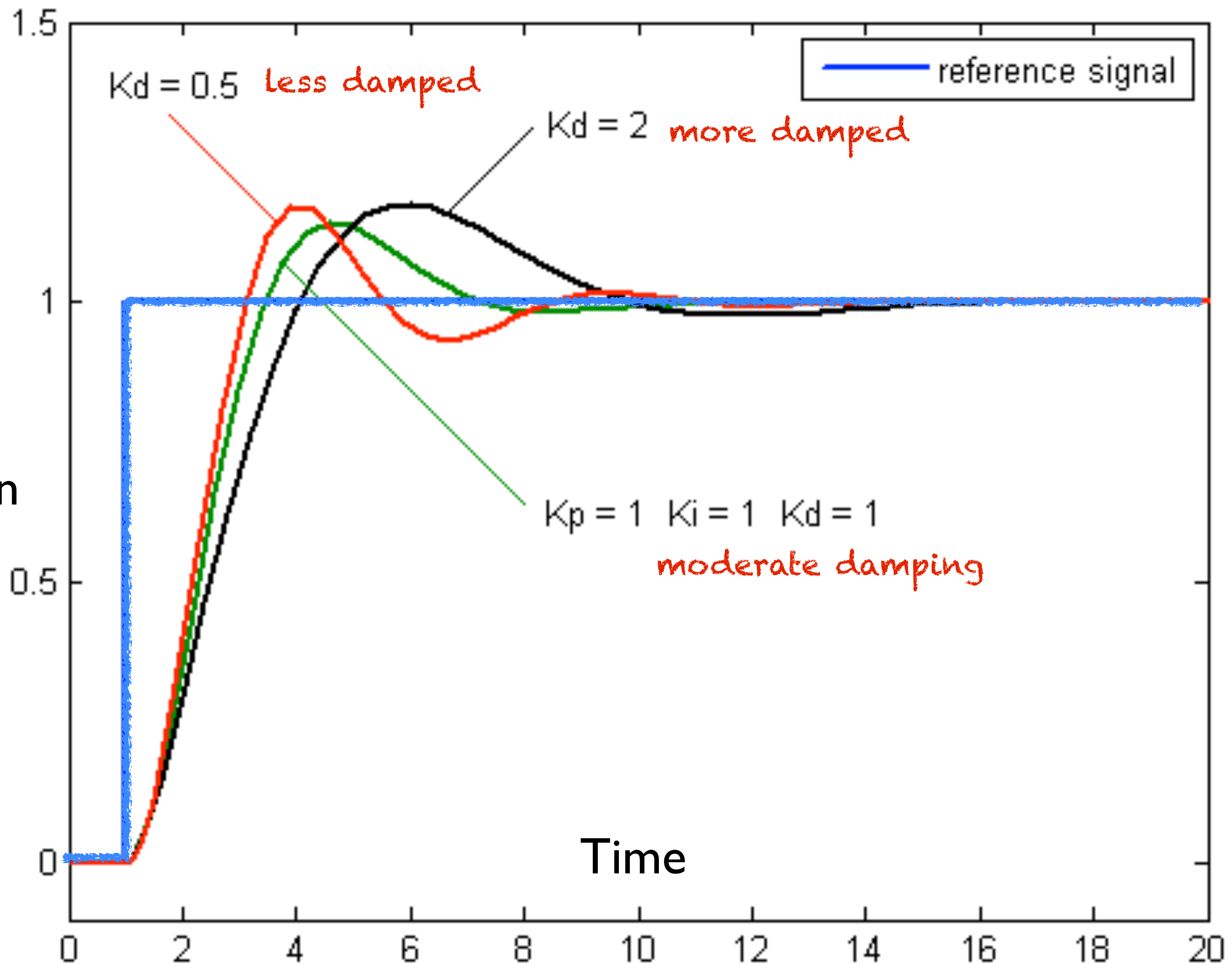


assuming constant set point,
velocity is derivative of error

add damper to
release energy

$$K_d \frac{d}{dt} e(t)$$

Position



PID Control

Error signal:

$$e(t) = x_{desired}(t) - x(t)$$

Control signal:

$$u(t) = K_p e(t) + K_i \int_0^t e(\alpha) d\alpha + K_d \frac{d}{dt} e(t)$$

$$\text{P} \quad K_p e(t)$$

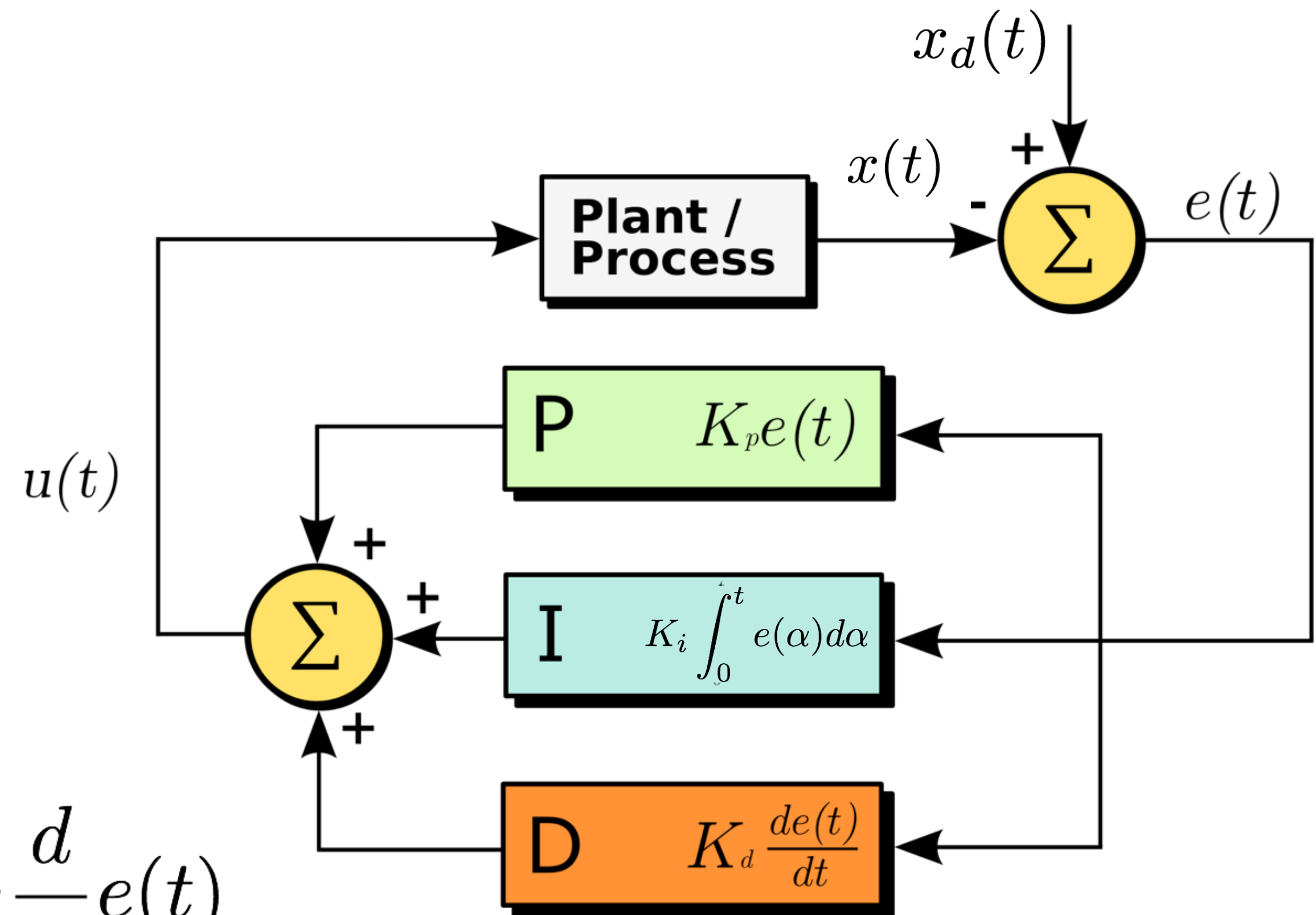
Current

$$\text{I} \quad K_i \int_0^t e(\alpha) d\alpha$$

Past

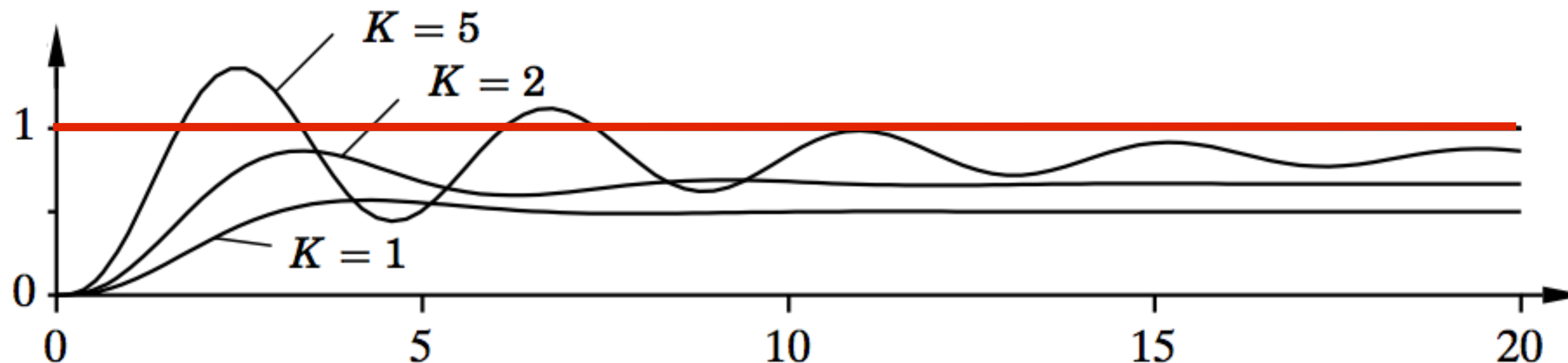
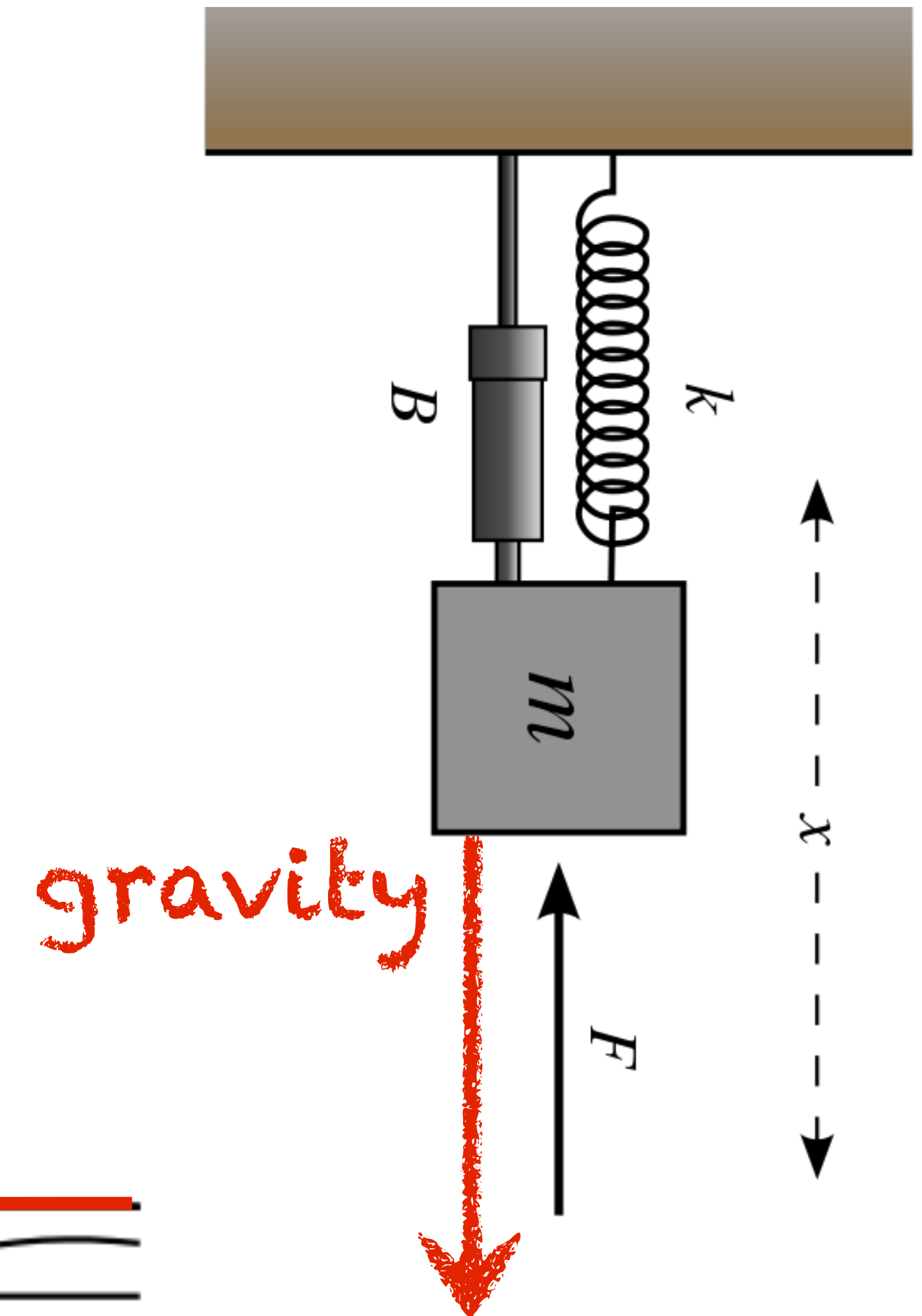
$$\text{D} \quad K_d \frac{de(t)}{dt}$$

Future



Steady state error

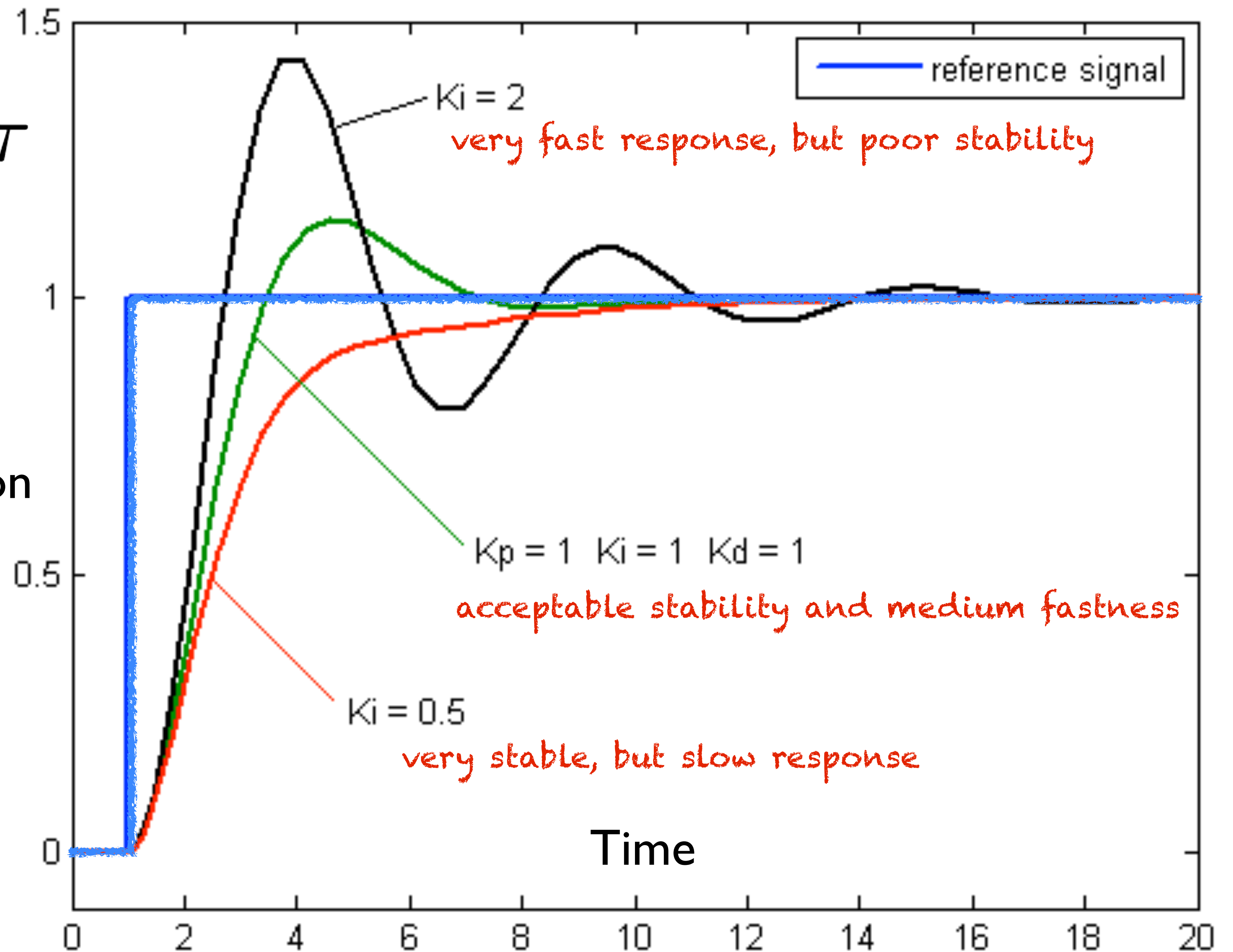
- Steady state error occurs when the system rests at equilibrium before reaching desired state
- Cause could be an significant external force, weak motor, low proportional gain, etc.
- PID integral term compensates by accumulating and acting against error toward convergence



$$K_i \int_0^t e(\tau) d\tau$$

$$\text{I} \quad K_i \int_0^t e(\tau) d\tau$$

Position



Time

Gain tuning

- Implementing PID algorithm will not necessarily produce a good controller
- Selection of the gains will greatly affect the performance of the controller
- PID gain tuning is more of an art than a science. Choose carefully.

$$u(t) = K_p e(t) + K_i \int_0^t e(\alpha) d\alpha + K_d \frac{d}{dt} e(t)$$

$$\text{P} \quad K_p e(t)$$

$$\text{I} \quad K_i \int_0^t e(\alpha) d\alpha$$

$$\text{D} \quad K_d \frac{de(t)}{dt}$$

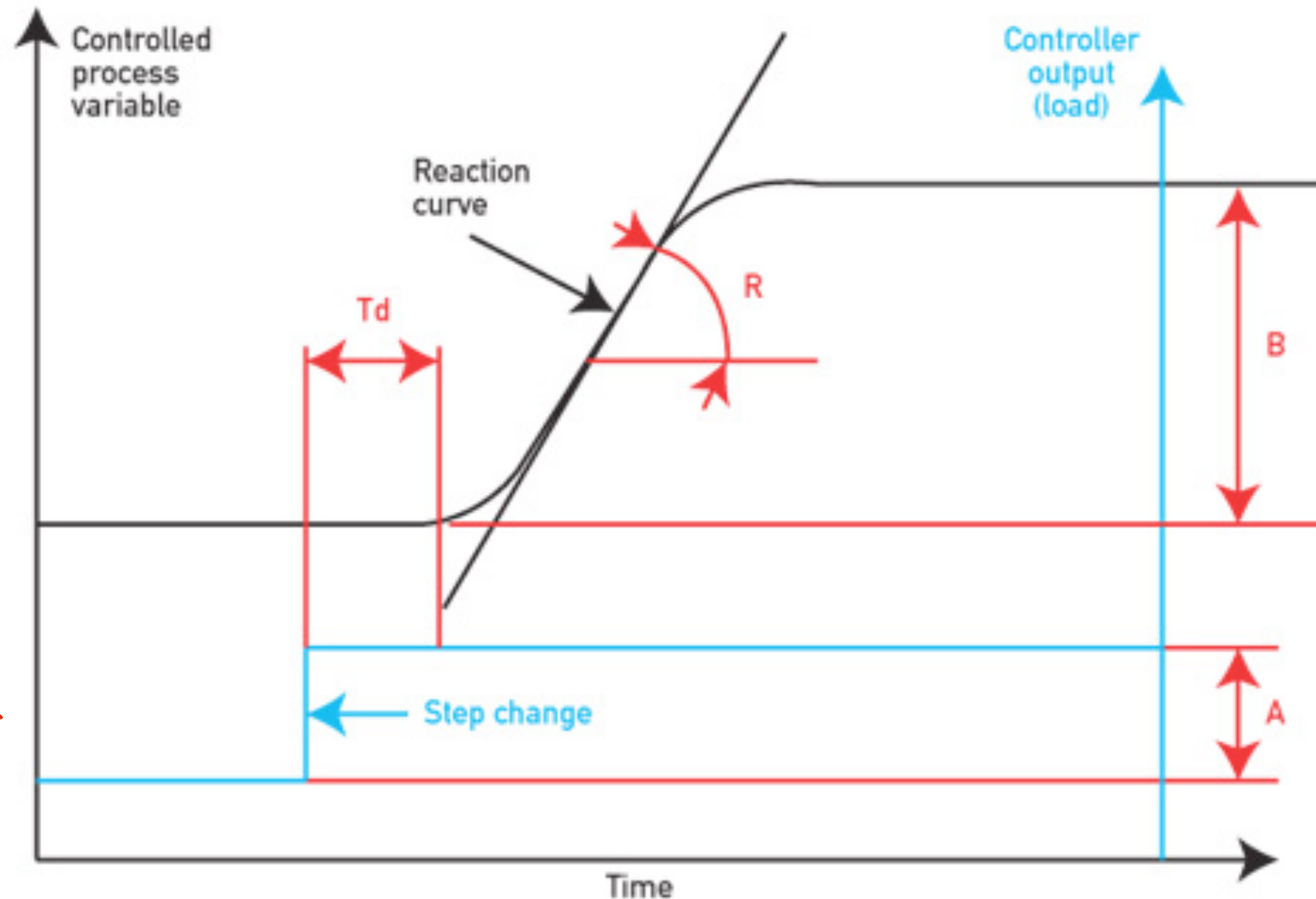
PID controller

From Wikipedia, the free encyclopedia

Effects of *increasing* a parameter independently^[14]

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability ^[11]
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

Ziegler-Nichols PID tuning



Explore on
your own

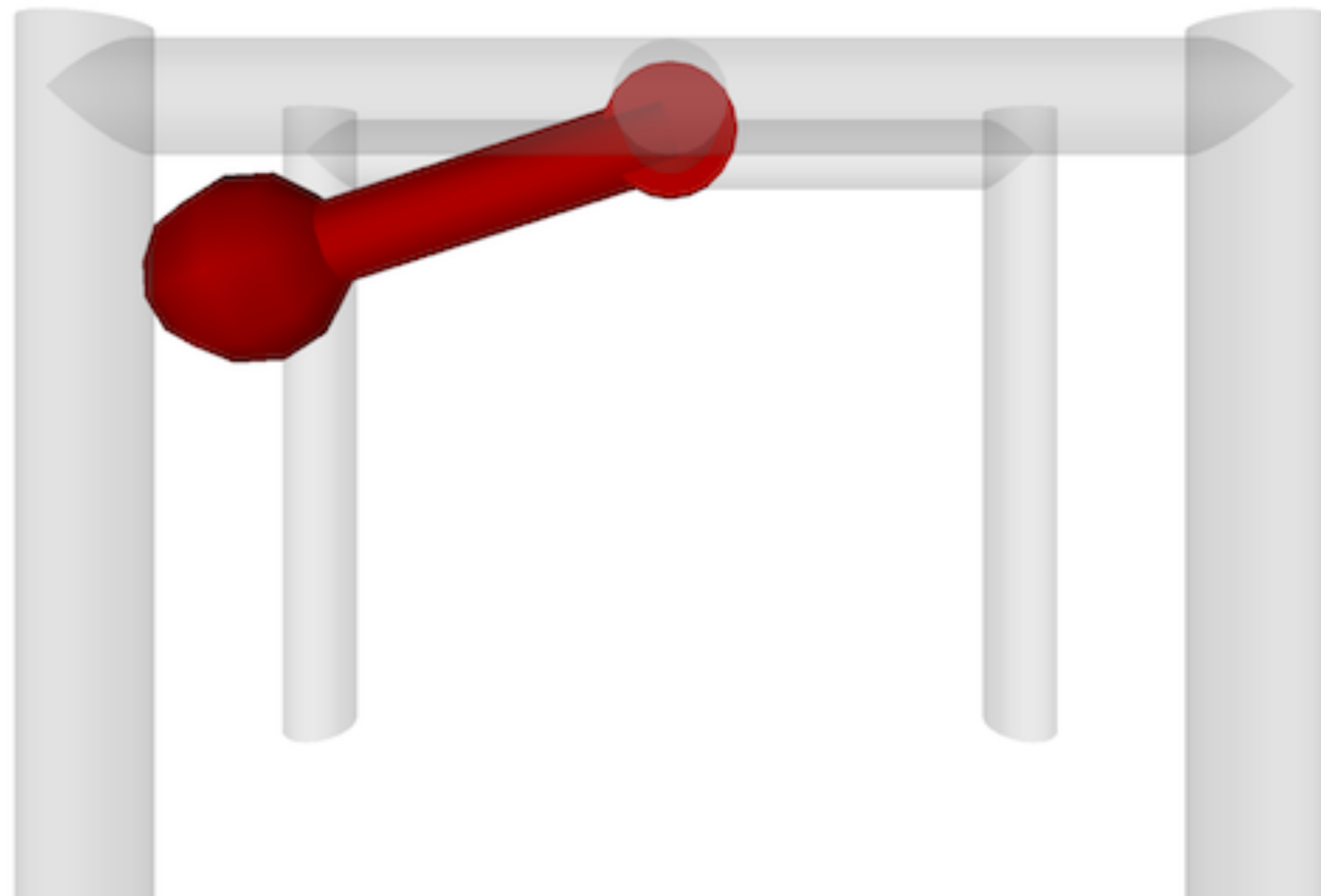
My approach to PID tuning

(take it or leave it)

- Start all gains at zero : $K_i = K_d = K_p = 0$
- Increase spring gain K_p until system roughly meets desired state
 - overshooting and oscillation about the desired state can be expected
- Increase damping gain K_d until the system is consistently stable
 - damping stabilizes motion, but system will have steady state error
- Increase integral gain K_i until the system consistently reaches desired
- Refine gains as needed to improve performance; Test from different states

Putting it all together (Pendularm)

Putting it all together (Pendularm)



Project 2: Pendularm

This Lecture

Feedback Control

PID Control of 1 DoF robot

Lecture 4

Dynamical Simulation

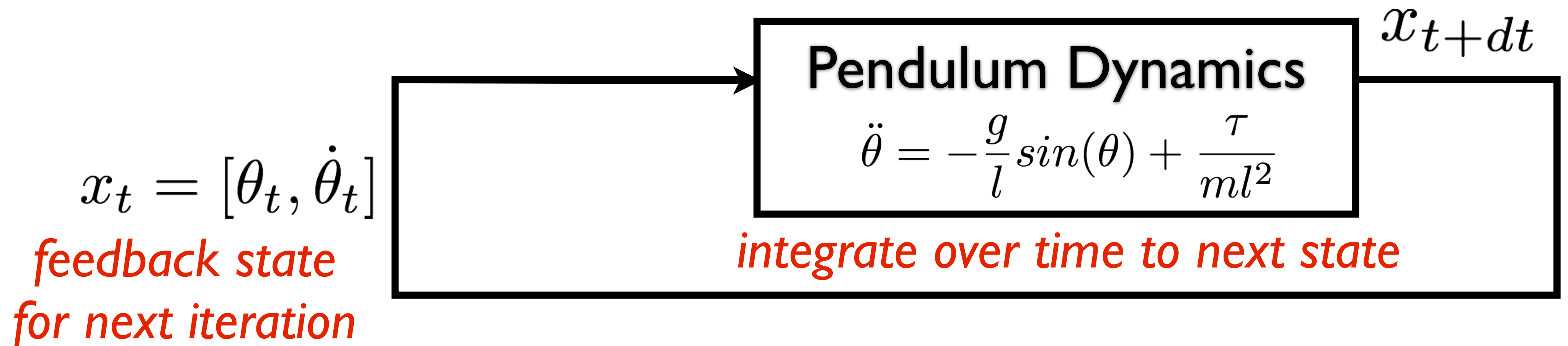
Equations of motion: Simple pendulum
Integrators: Euler, Verlet, Velocity Verlet, RK4

Putting it all together (Pendularm)

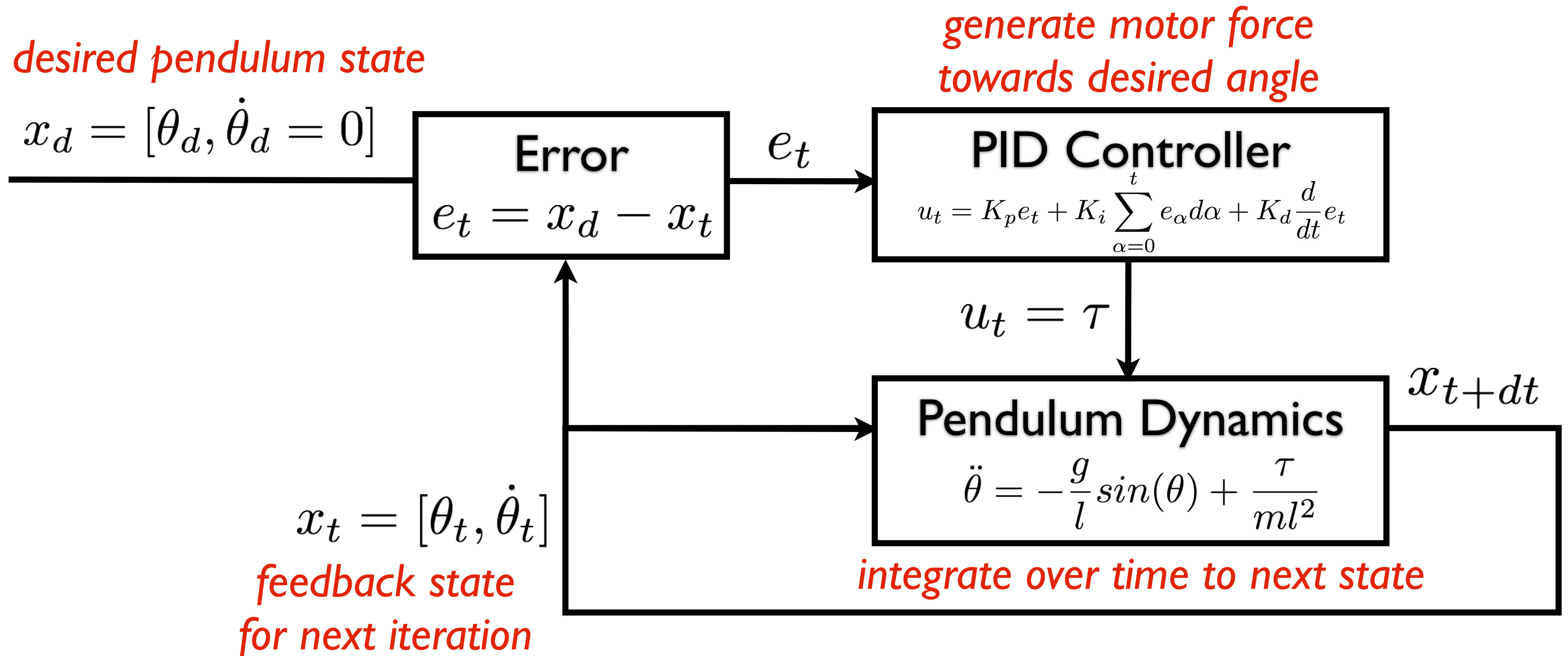
This Lecture

Feedback Control

PID Control of 1 DoF robot



Putting it all together (Pendularm)



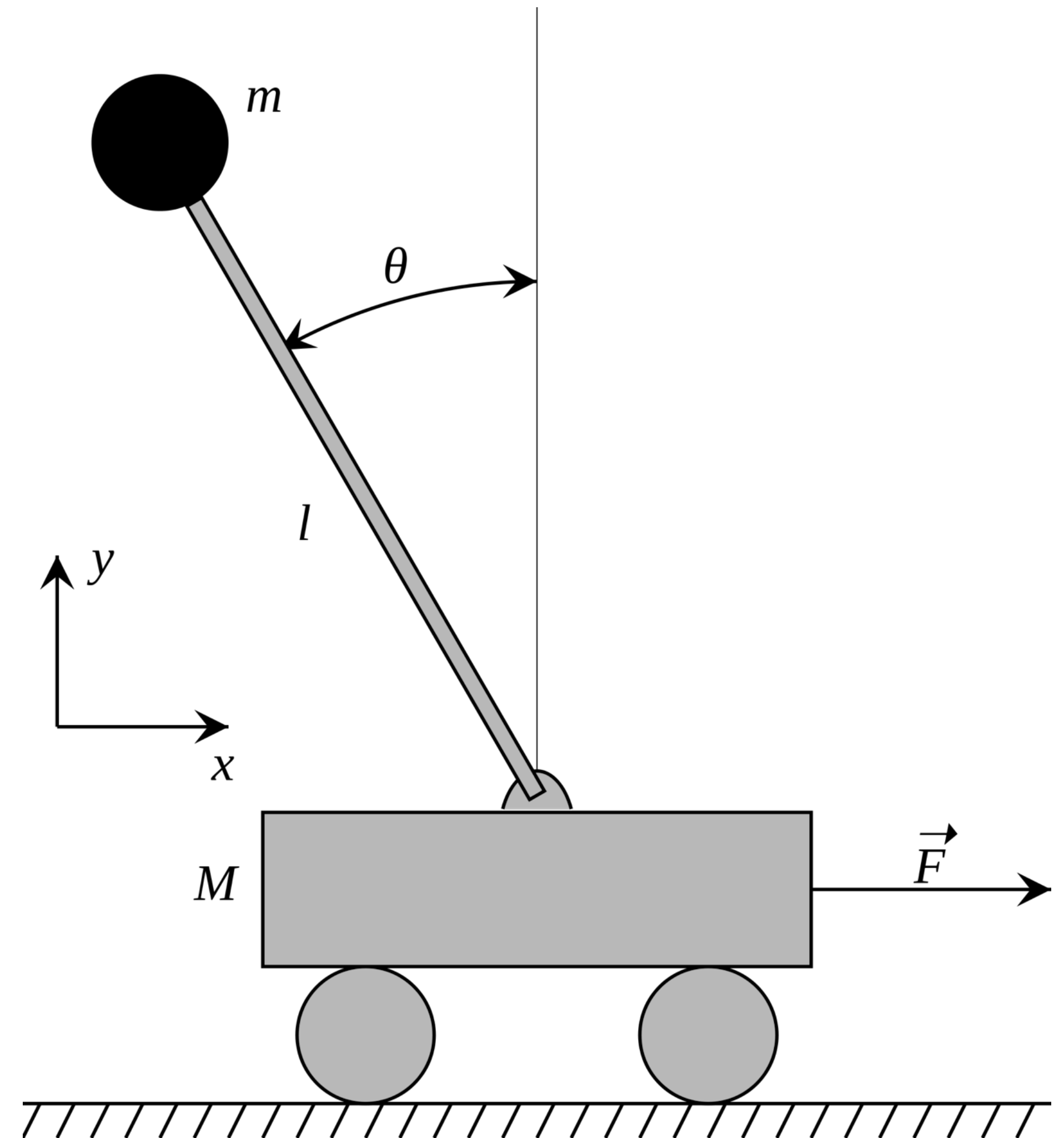
Warning:
Changing system $\rightarrow$ changing gains

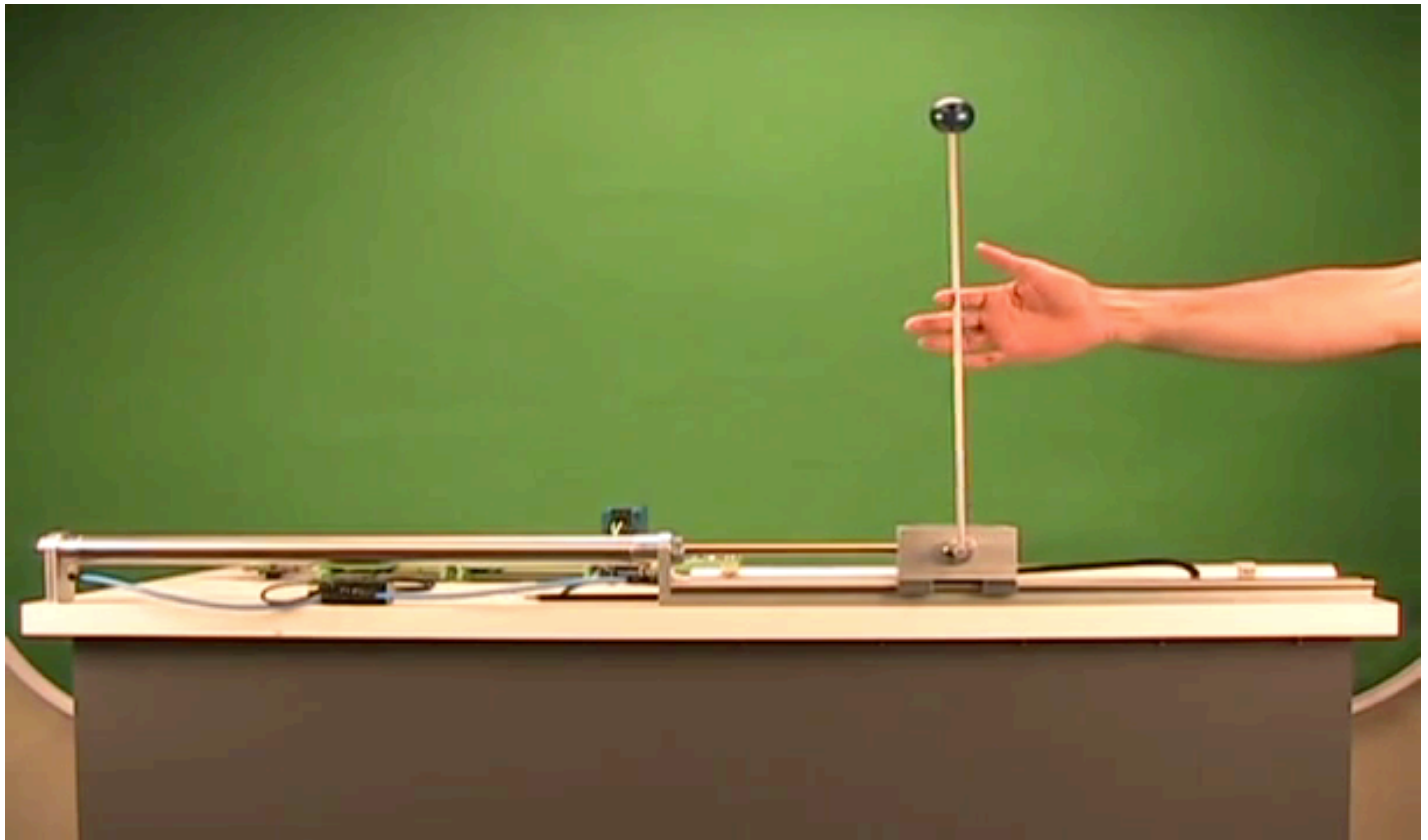
Other types of systems?

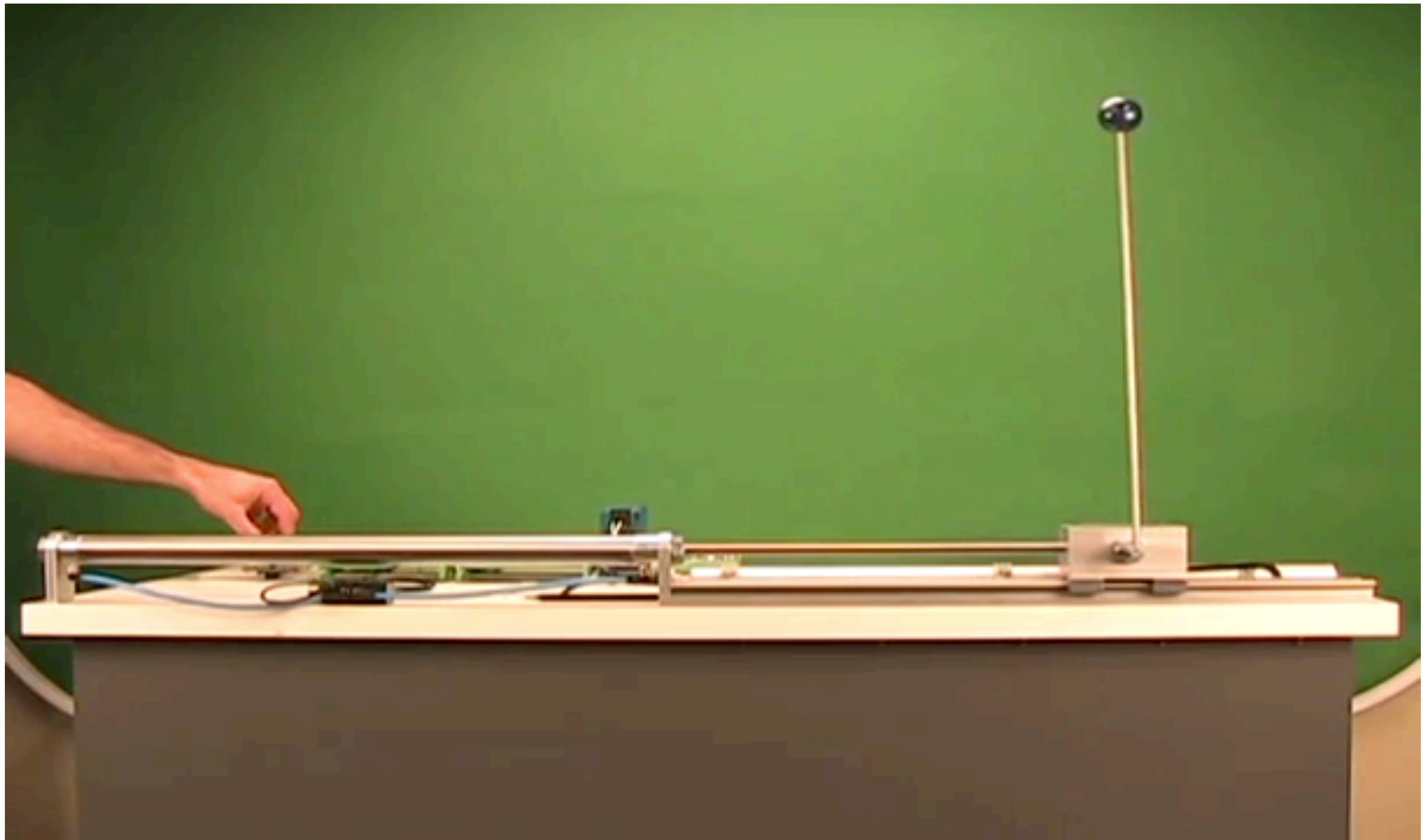
Cart Pole Balancer

$$(M + m) \ddot{x} - m\ell\ddot{\theta} \cos\theta + m\ell\dot{\theta}^2 \sin\theta = F$$

$$\ell\ddot{\theta} - g \sin\theta = \ddot{x} \cos\theta$$

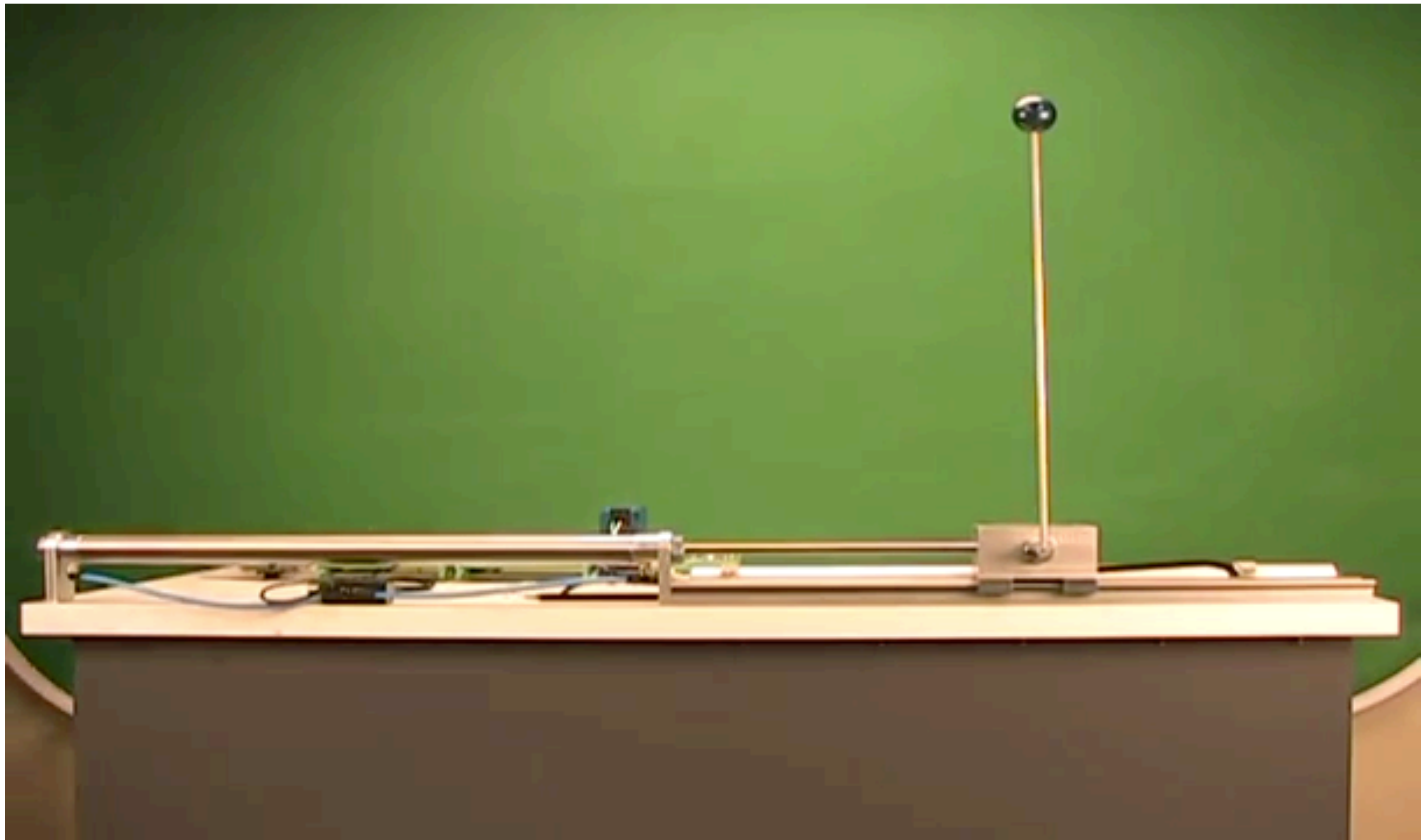






<https://www.youtube.com/watch?v=a4c7AwHFkT8>

Enfield Technologies
Michigan Robotics 567-520-4400 autorob.org



<https://www.youtube.com/watch?v=a4c7AwHFkT8>

Enfield Technologies
Michigan Robotics 567520 autorob.org

Differential Drive Inverted Pendulum





<https://www.youtube.com/watch?v=hpCwdu0e3i0>

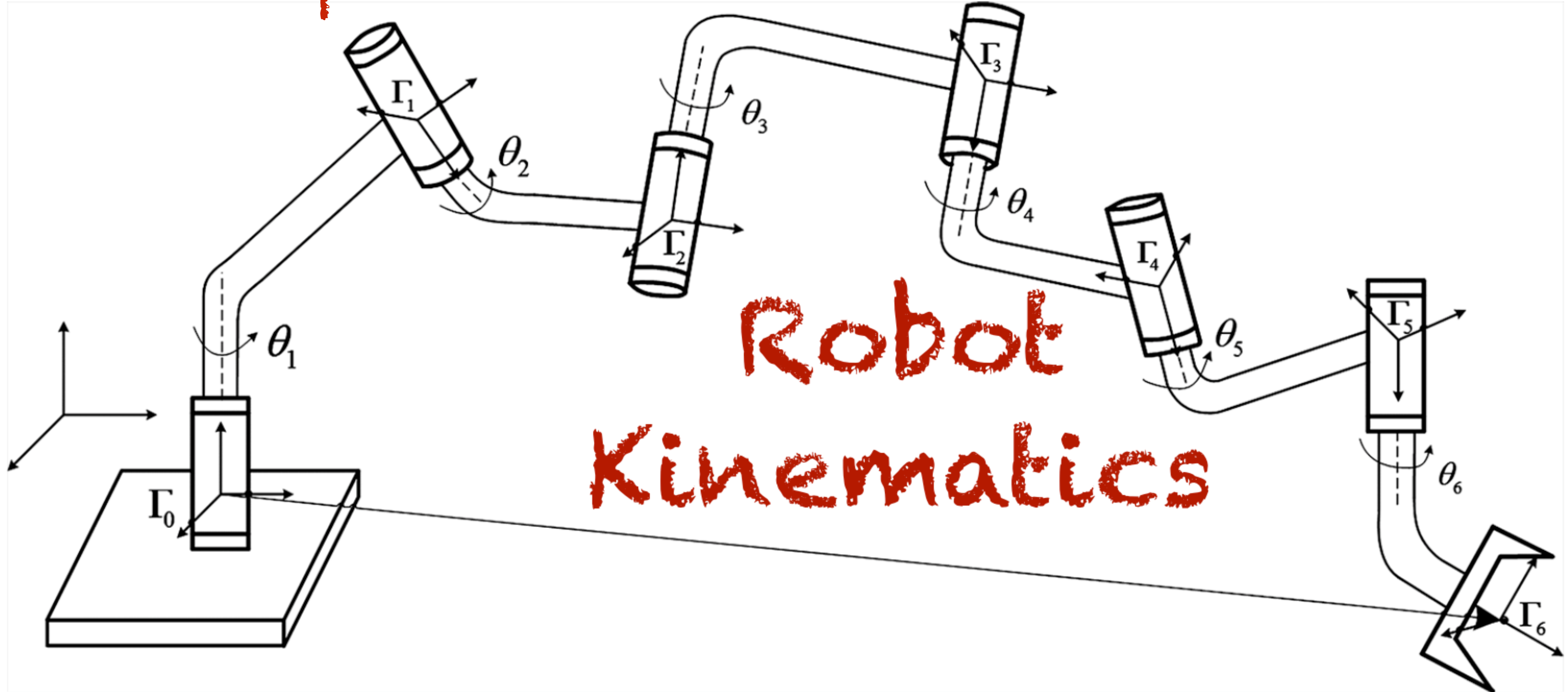
techbricks.nl
Michigan Robotics 367/320 - autorob.org

Do not forget to review:

Coordinate Systems
and
Linear Algebra



Next topic:



invisible segway

