Lecture 16
Motion Control

Slide borrowed from Michigan Robotics autorob.org

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

Slide borrowed from Michigan Robotics autorob.org

https://youtu.be/7osiCOjdjBQ

Slide borrowed from Michigan Robotics autorob.org

https://youtu.be/7osiCOjdjBQ

Slide borrowed from Michigan Robotics autorob.org

https://youtu.be/KiGGTmuBujs

Slide borrowed from Michigan Robotics autorob.org

https://youtu.be/lXkYz0AfcAk

Slide borrowed from Michigan Robotics autorob.org

Biped Robotics Lab @ Michigan - https://youtu.be/-35xJfFMDkE

Slide borrowed from Michigan Robotics autorob.org

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

Slide borrowed from Michigan Robotics autorob.org

Control must be fast and often

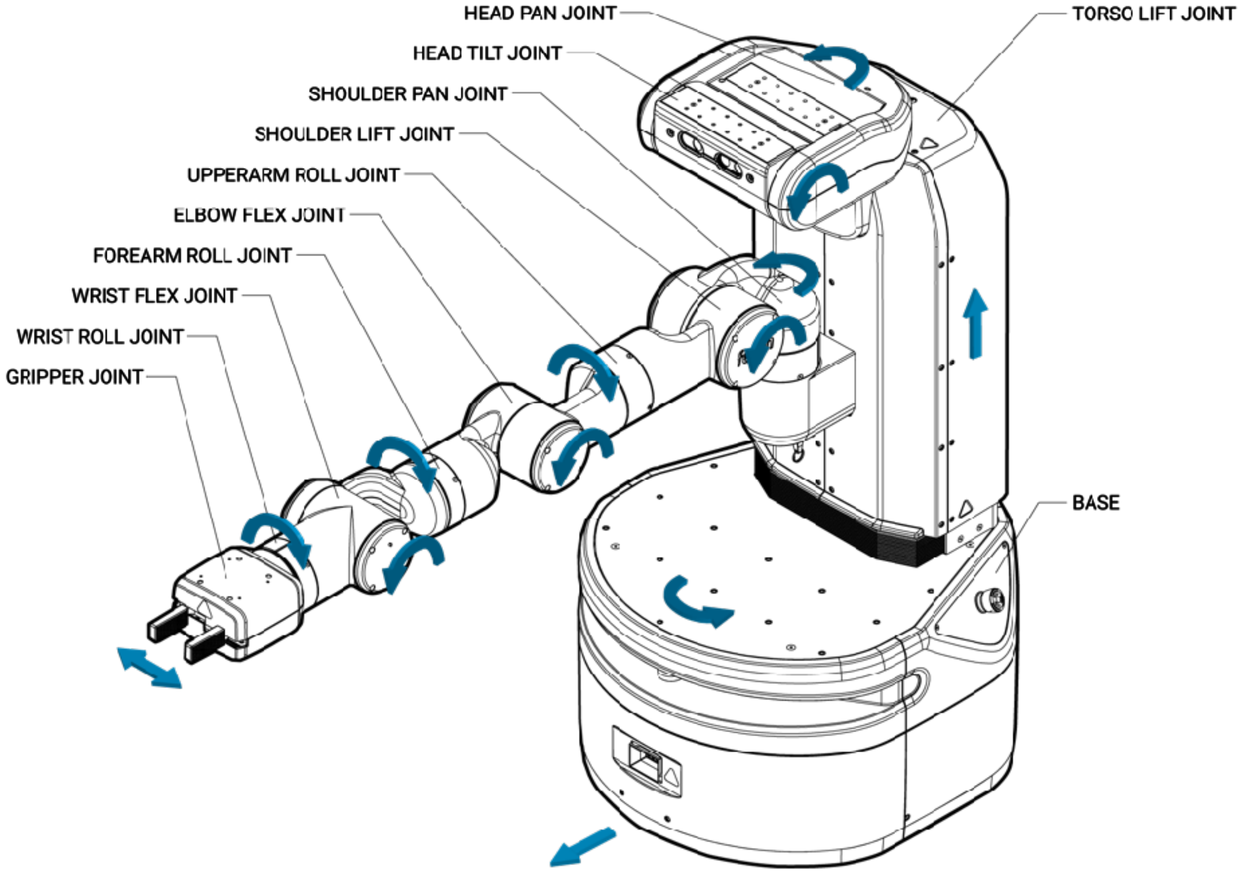
Slide borrowed from Michigan Robotics autorob.org

Slide borrowed from Michigan Robotics autorob.org

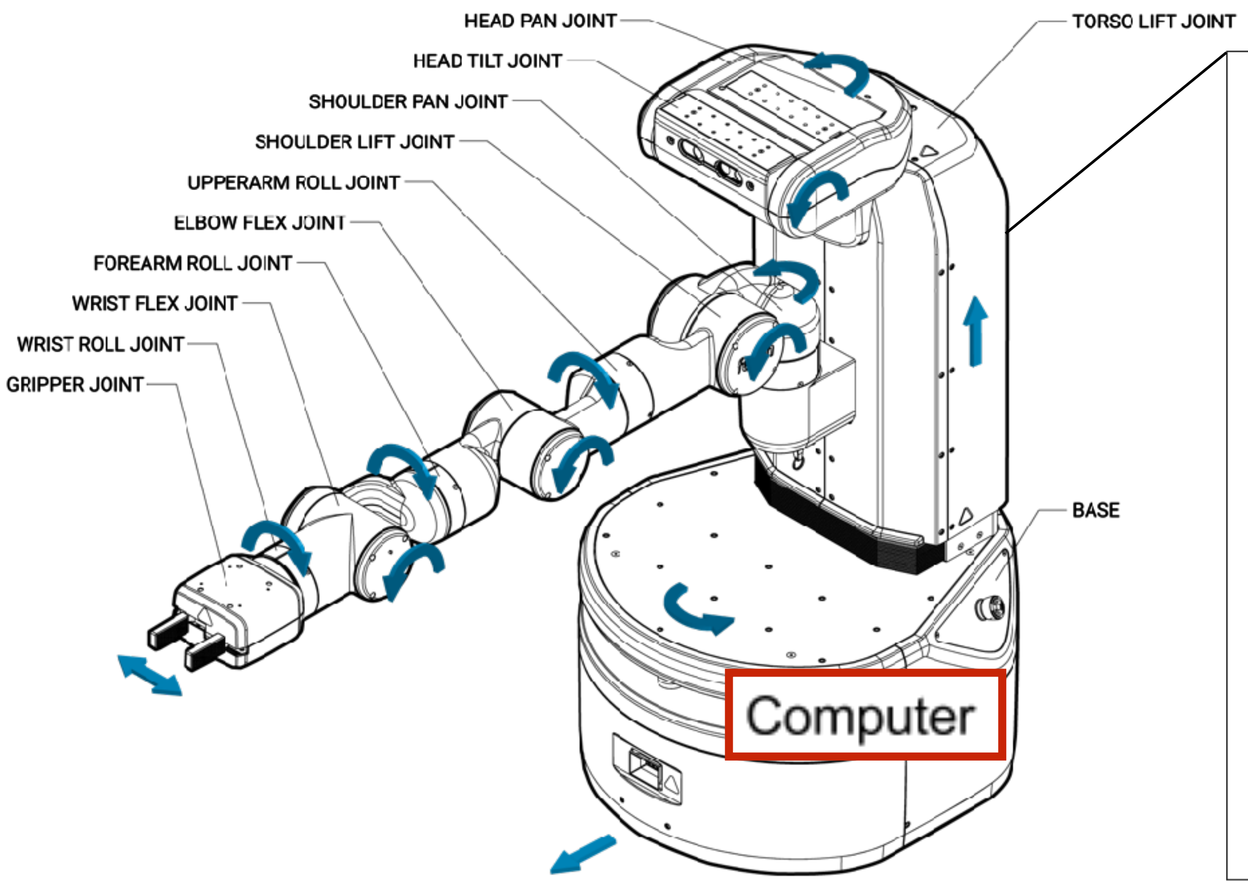
Communication inside a robot is like its central nervous system

Slide borrowed from Michigan Robotics autorob.org

Communication inside a robot is like its central nervous system

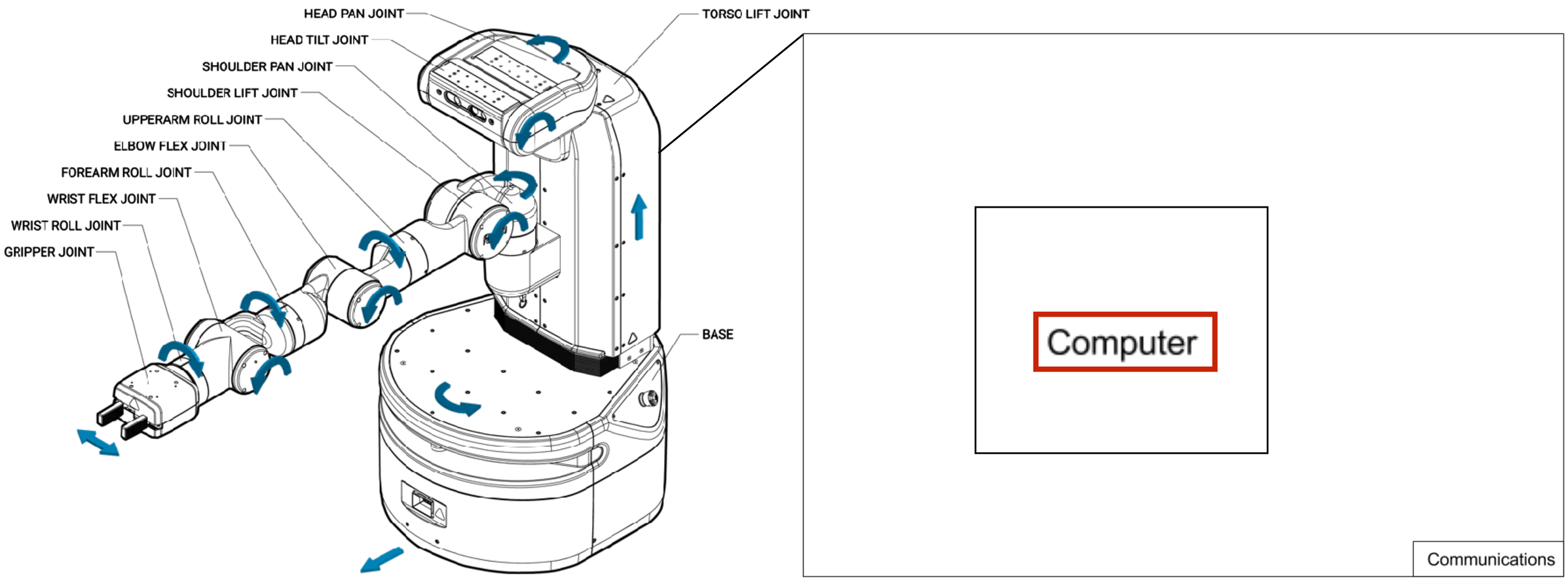


Slide borrowed from Michigan Robotics autorob.org

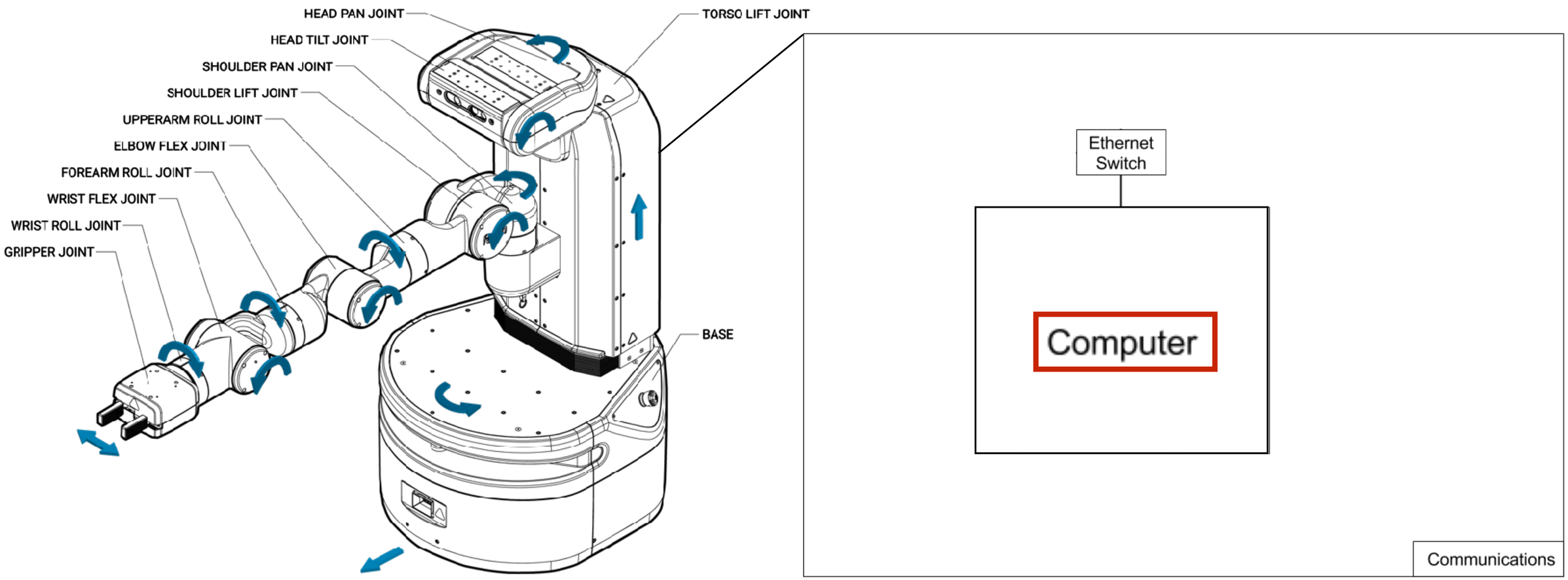


Communications

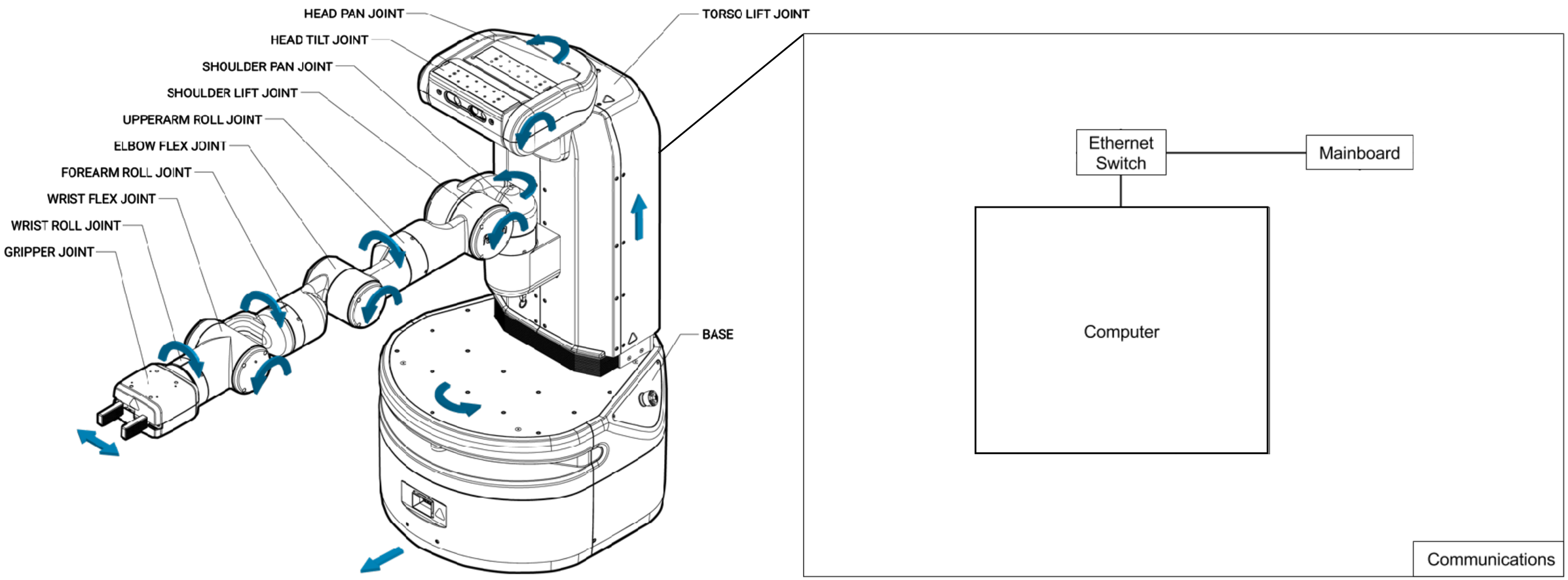
Slide borrowed from Michigan Robotics autorob.org



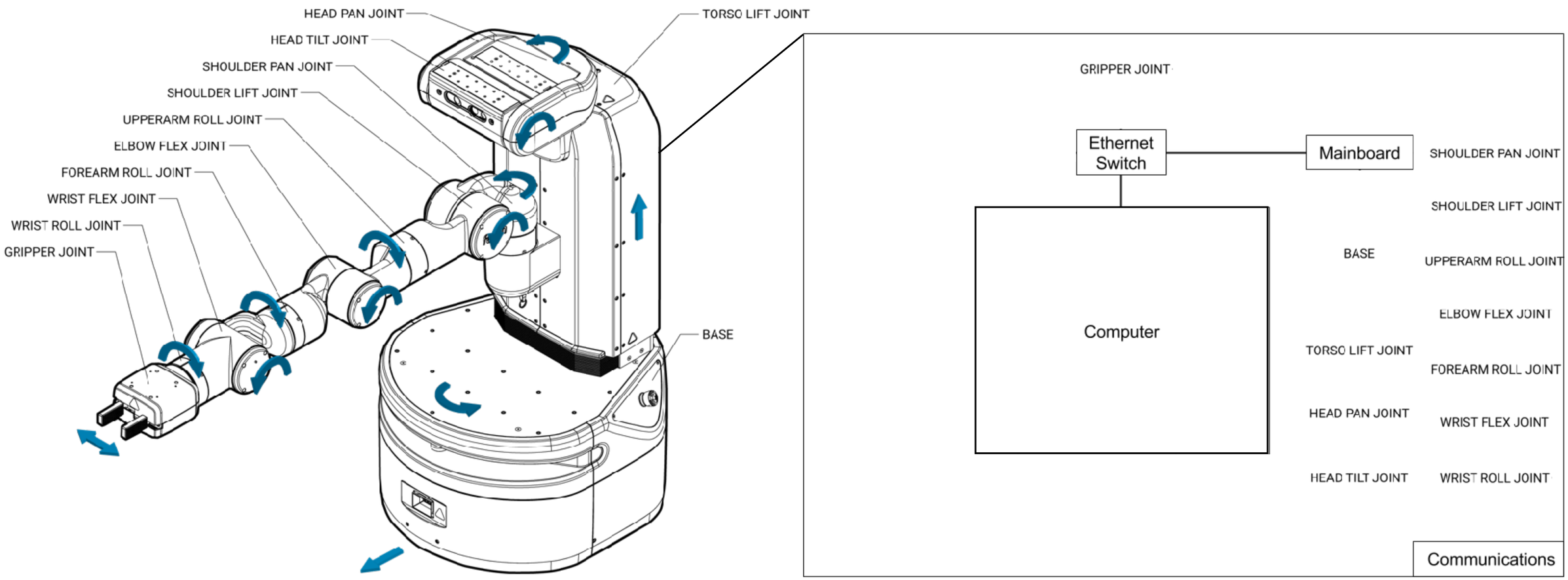
Slide borrowed from Michigan Robotics autorob.org



Slide borrowed from Michigan Robotics autorob.org

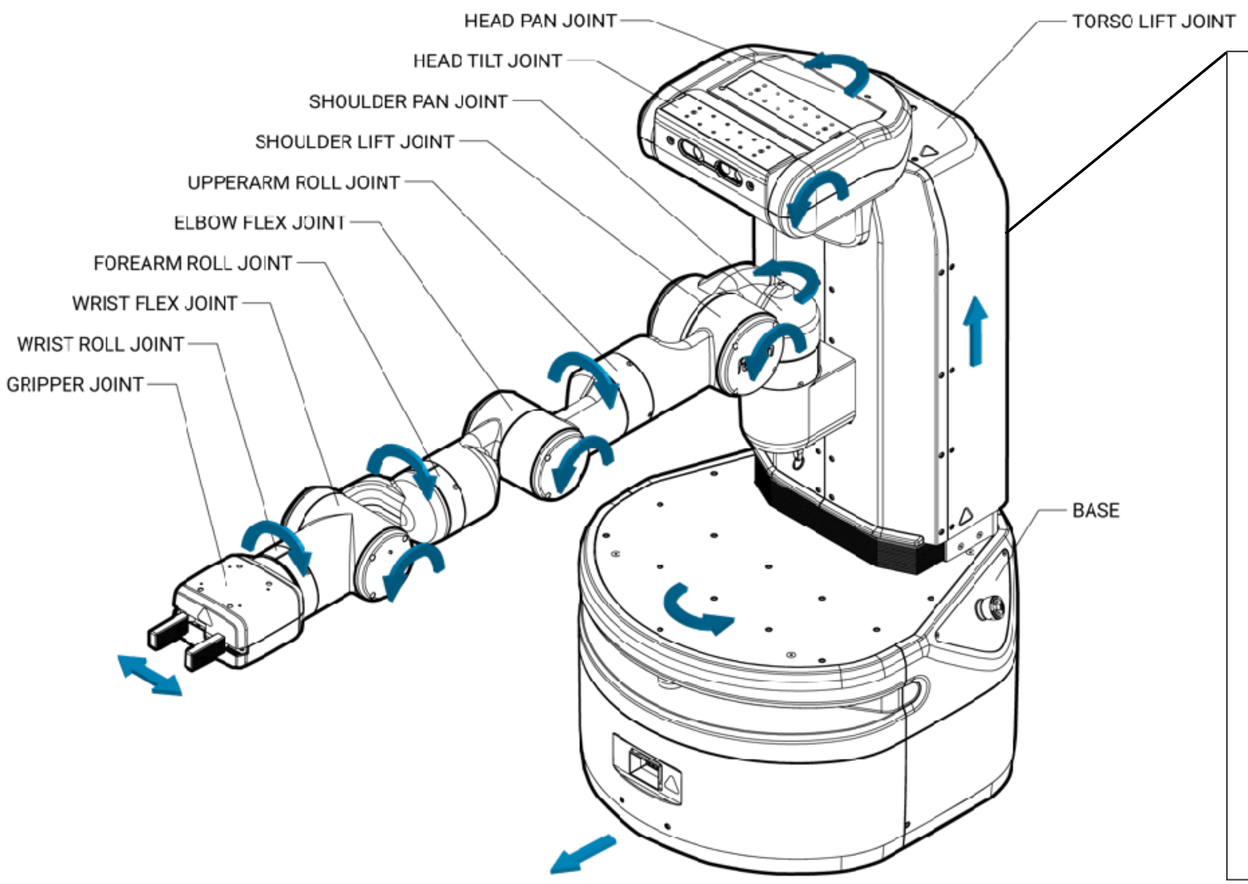


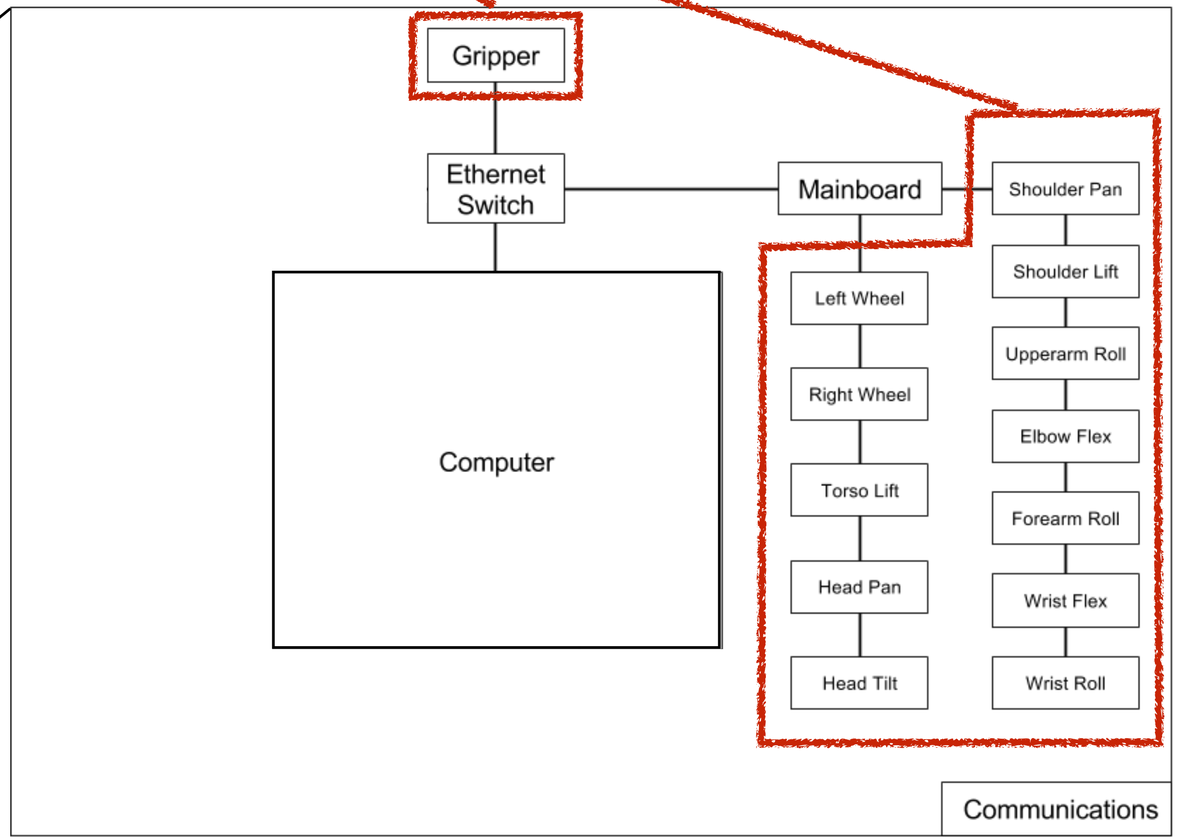
Slide borrowed from Michigan Robotics autorob.org



Slide borrowed from Michigan Robotics autorob.org

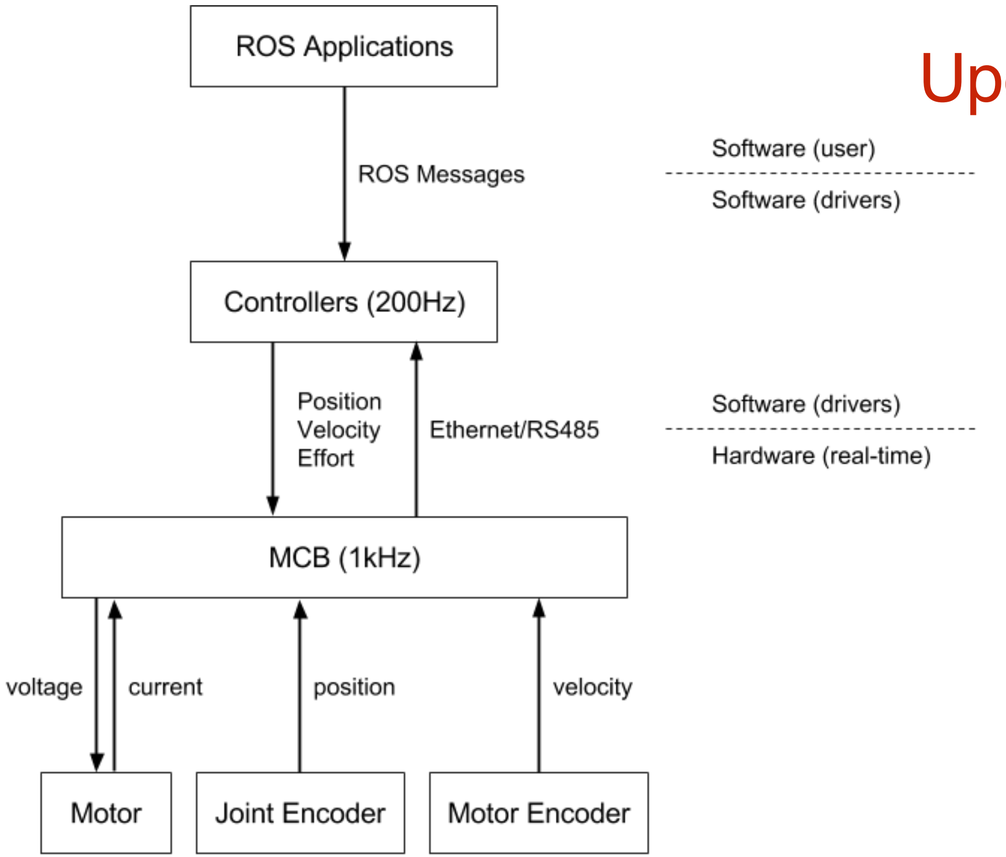
Update joint control 1,000 times per second

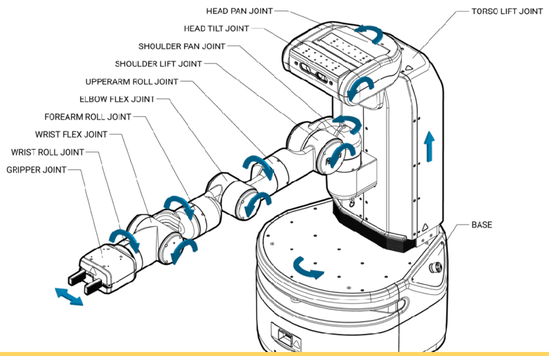


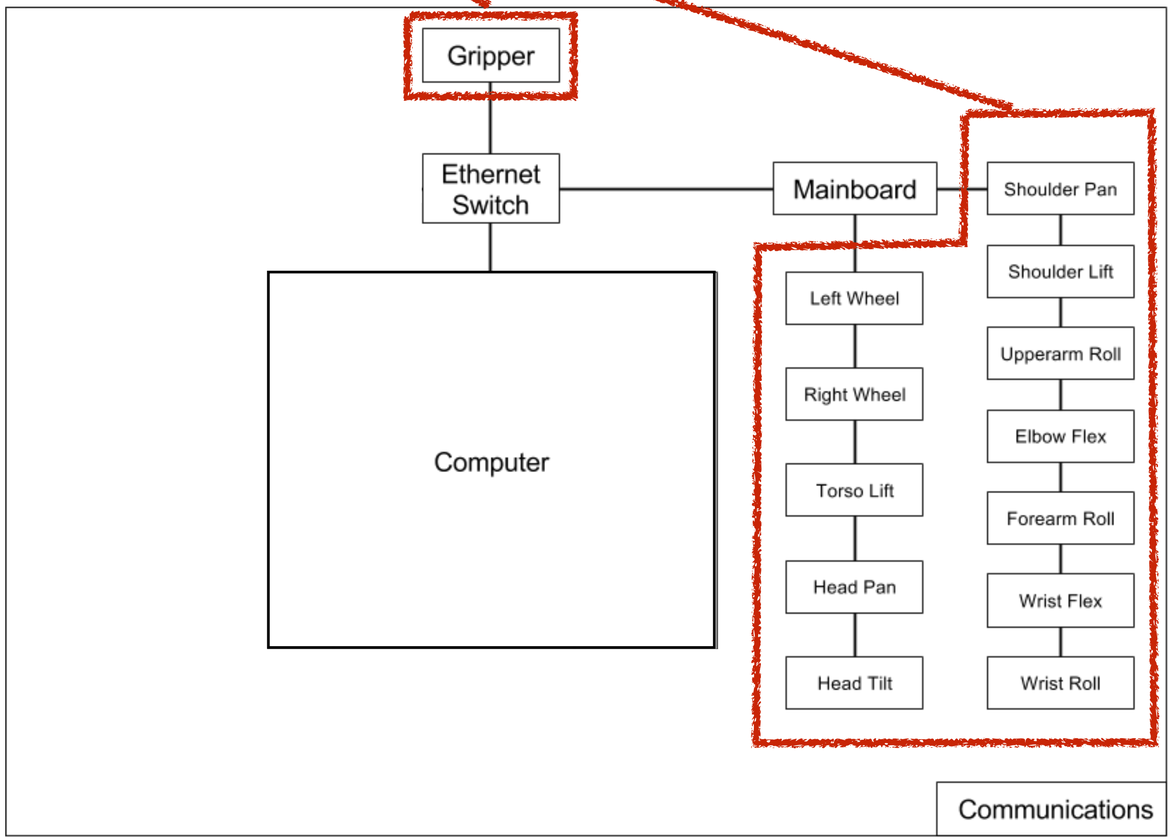


Slide borrowed from Michigan Robotics autorob.org

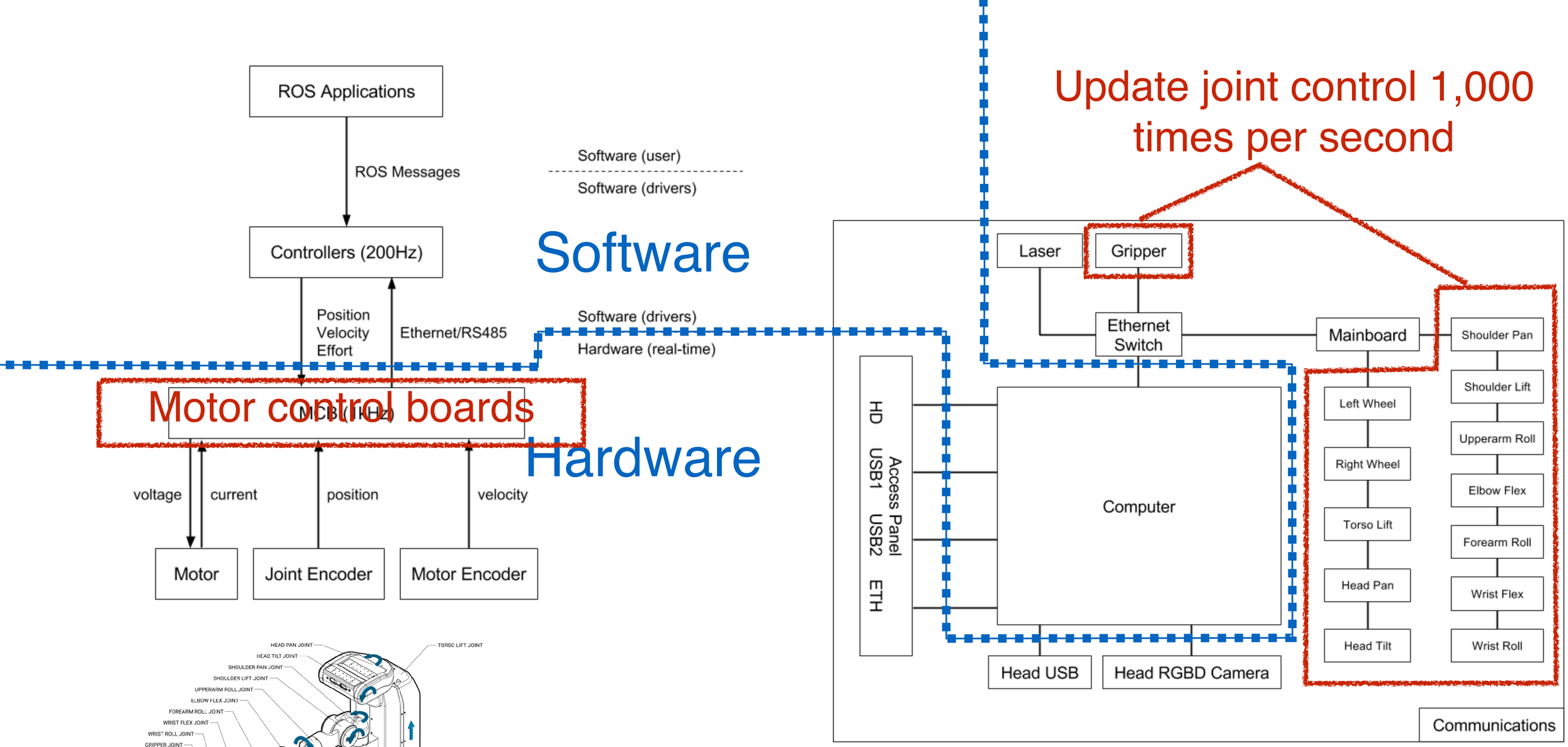
Update joint control 1,000 times per second

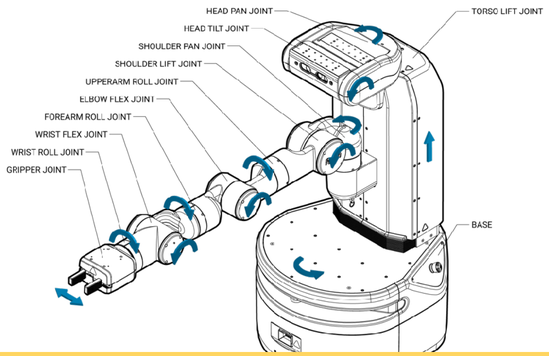






Slide borrowed from Michigan Robotics autorob.org





Slide borrowed from Michigan Robotics autorob.org

Motor control boards

Update joint control 1,000 times per second

actions

ROS Applications

ROS Messages

Controllers (200Hz)

Software (user)

Software (drivers)

Software

Setpoints

Position
Velocity
Effort

Ethernet/RS485

Software (drivers)

Hardware (real-time)

Motor commands

MCB (1kHz)

Hardware

voltage

current

position

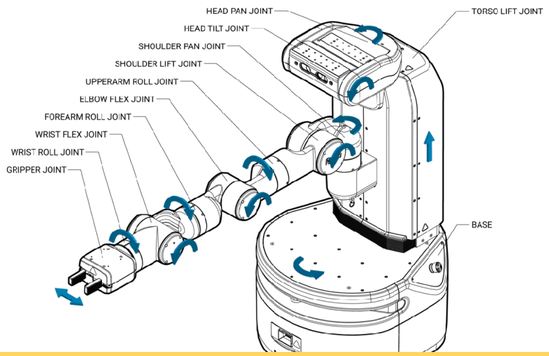
velocity

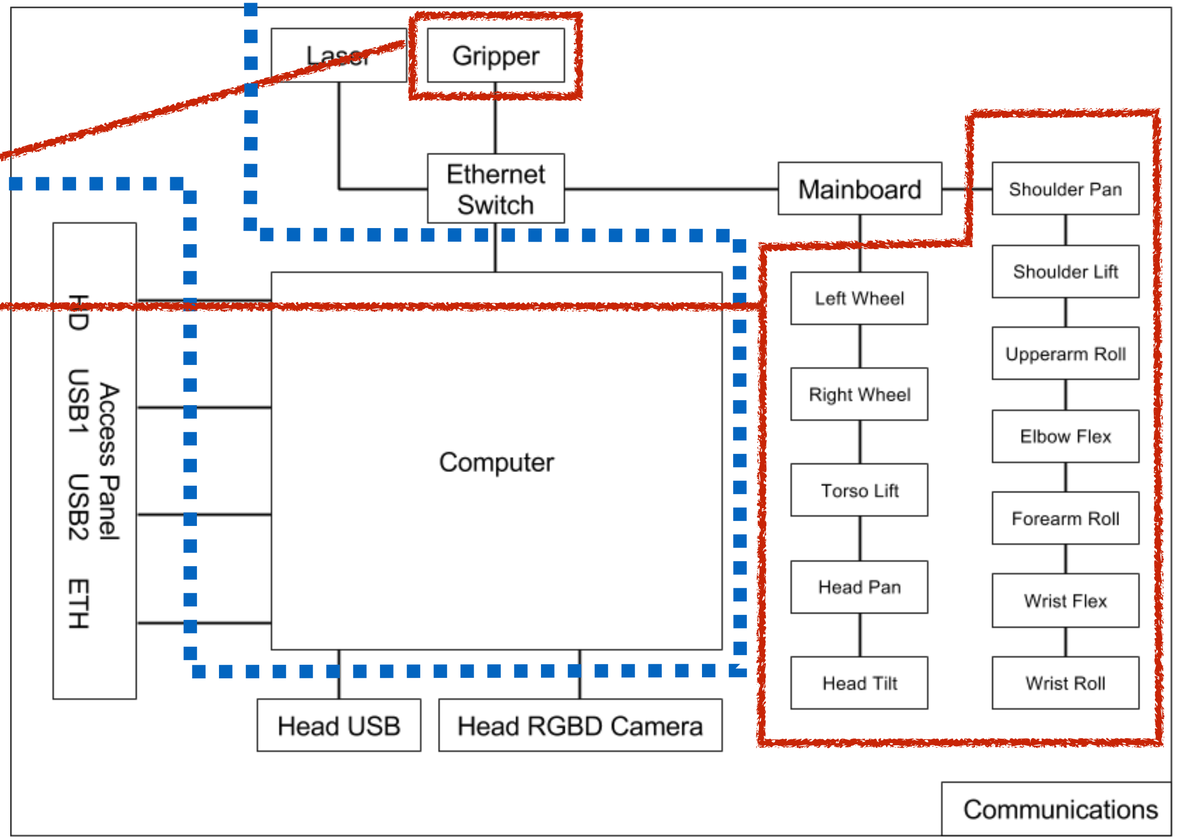
Proprioception

Motor

Joint Encoder

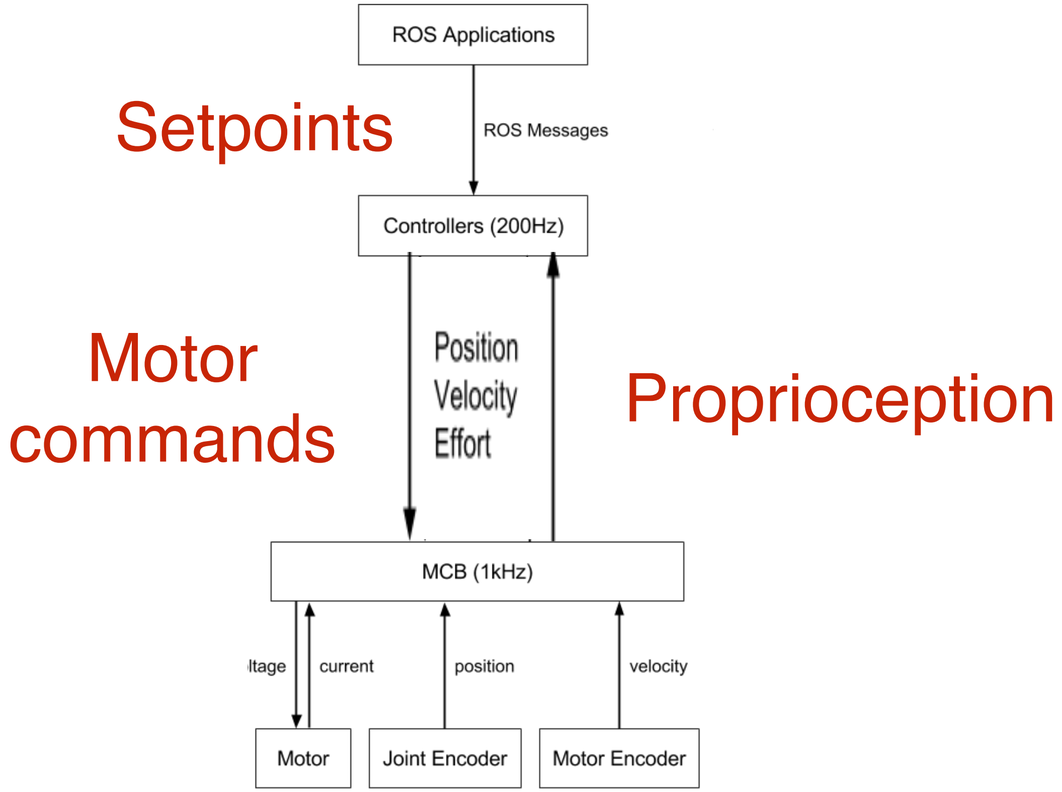
Motor Encoder

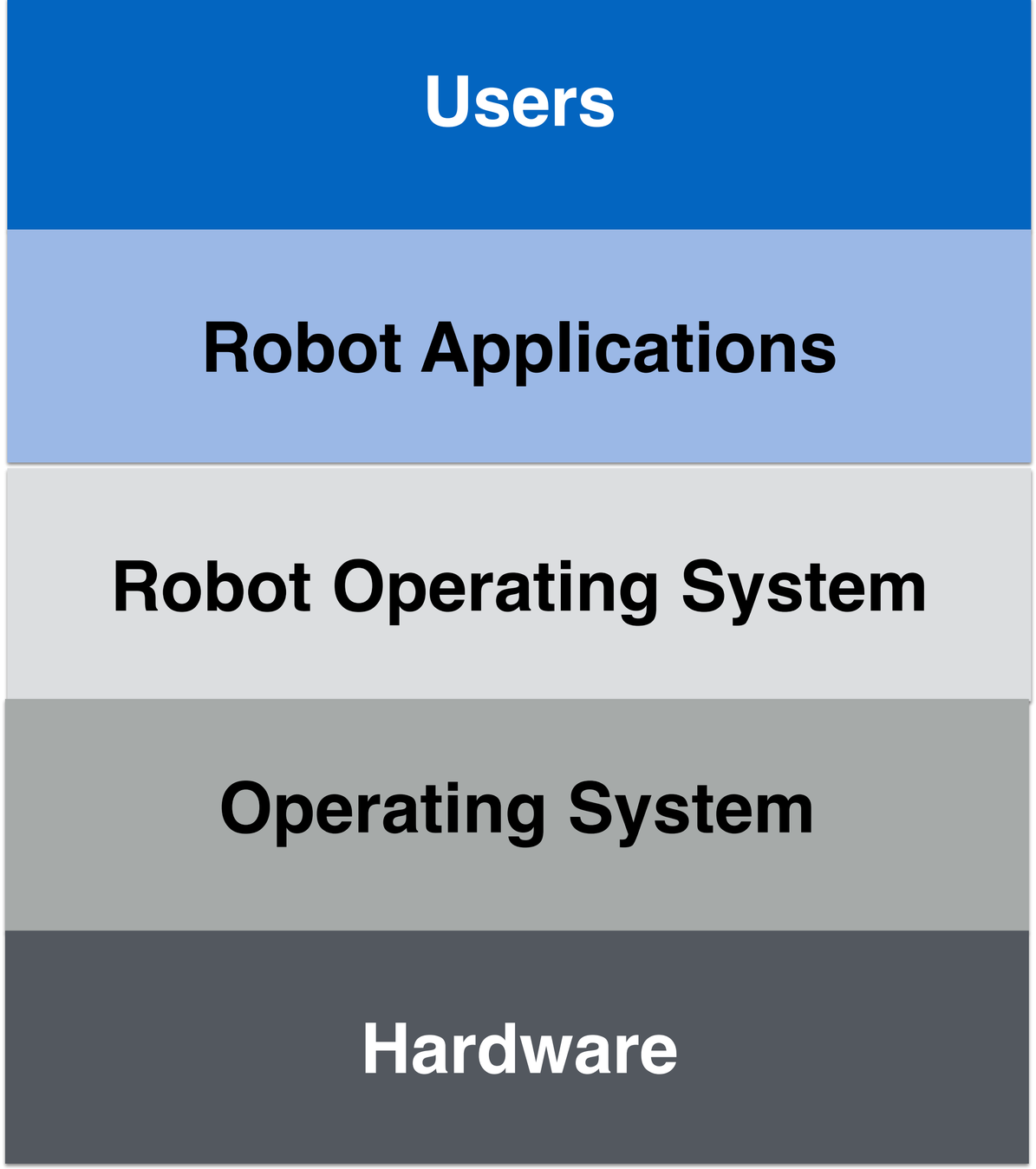




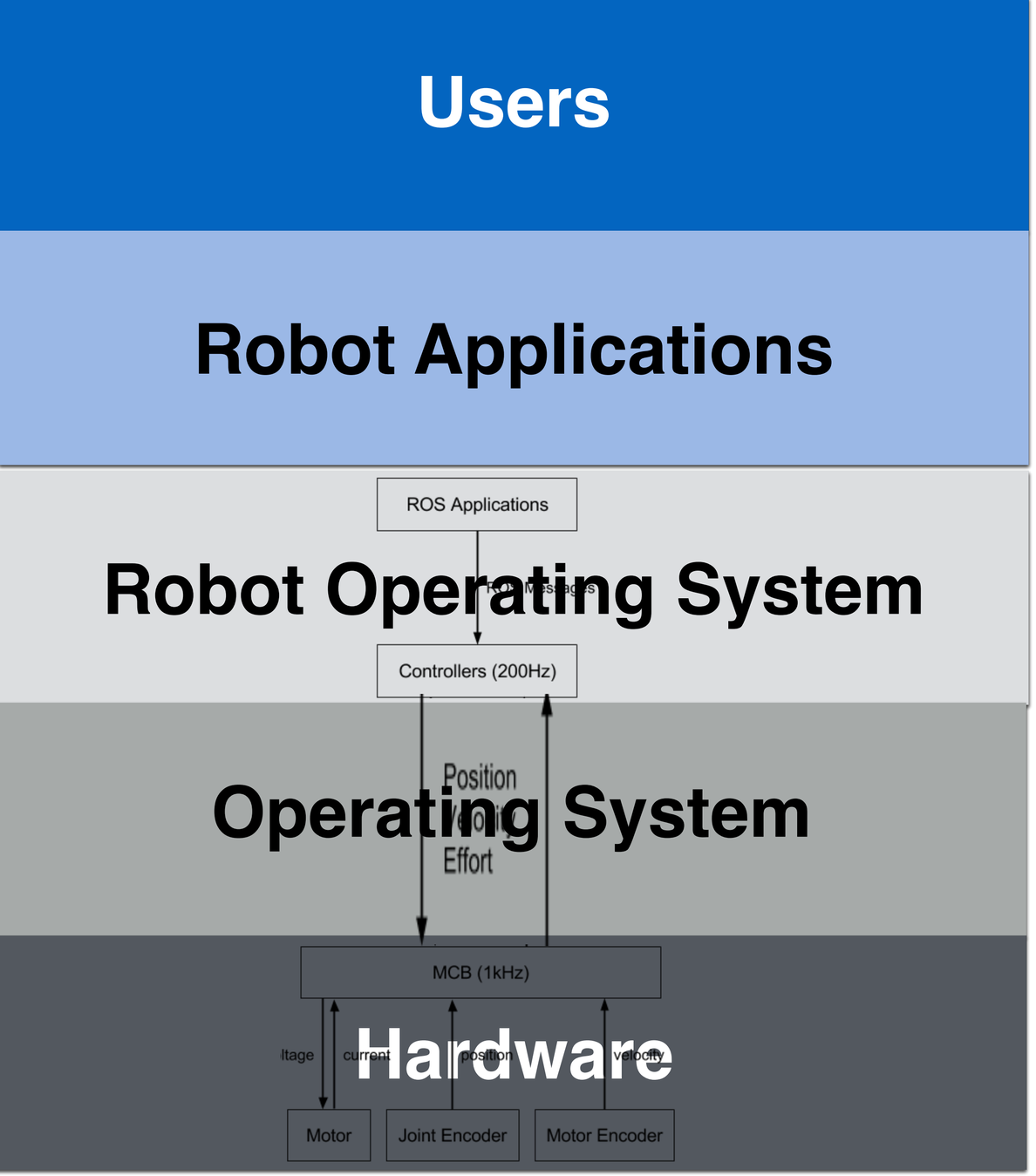
Slide borrowed from Michigan Robotics autorob.org

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





Slide borrowed from Michigan Robotics autorob.org



Motion control

Slide borrowed from Michigan Robotics autorob.org

What can you do with pendularm?

Slide borrowed from Michigan Robotics autorob.org

Inverted Pendulum by PID control - Chengcheng Zhu et al. @UMich

Slide borrowed from Michigan Robotics autorob.org

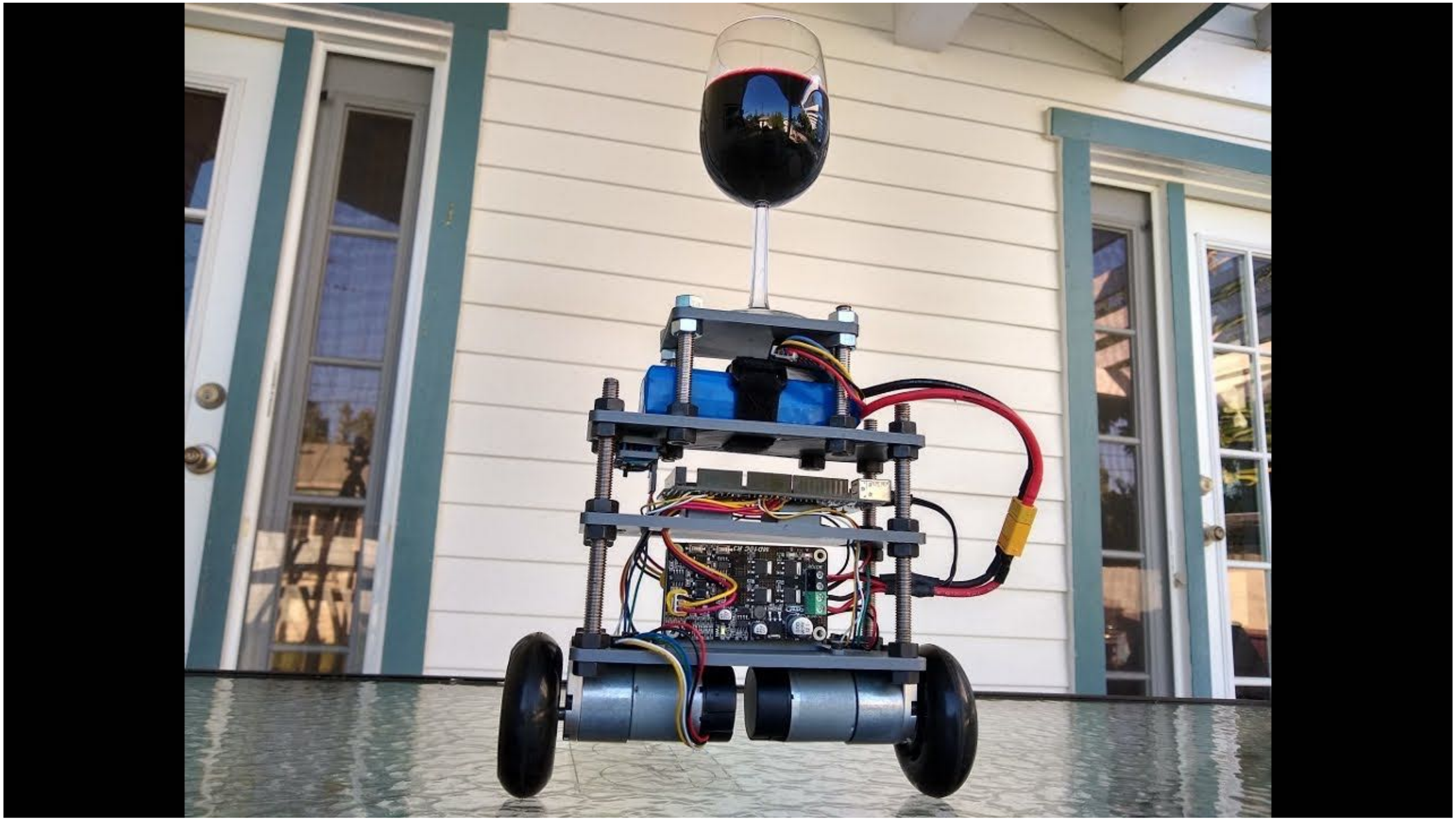
Inverted Pendulum by PID control - Chengcheng Zhu et al. @UMich

Slide borrowed from Michigan Robotics autorob.org

Slide borrowed from Michigan Robotics autorob.org

Building up to more…

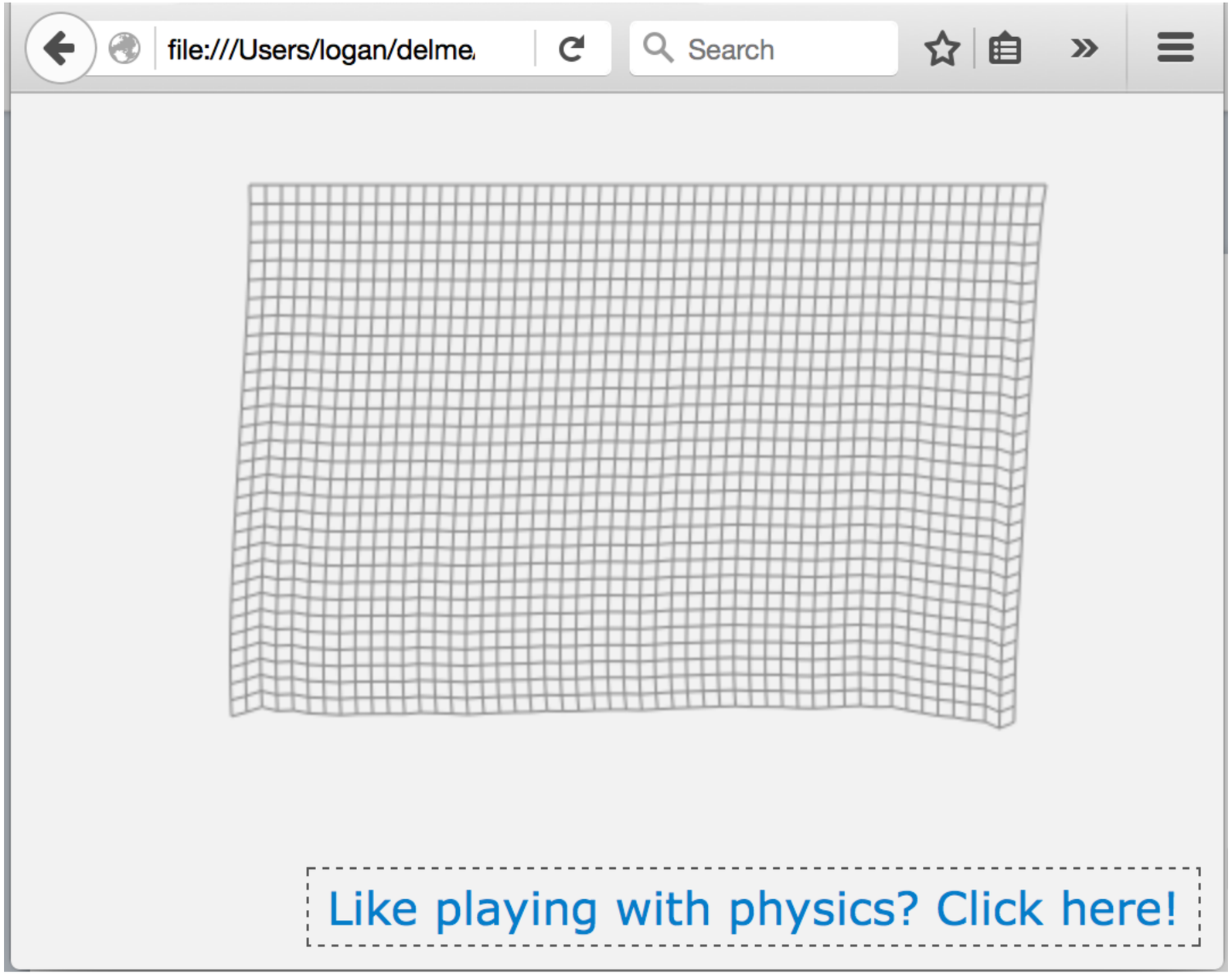
Slide borrowed from Michigan Robotics autorob.org



Shay Sackett - https://www.youtube.com/watch?v=t7skvD6v2TI

Let’s start with something fun

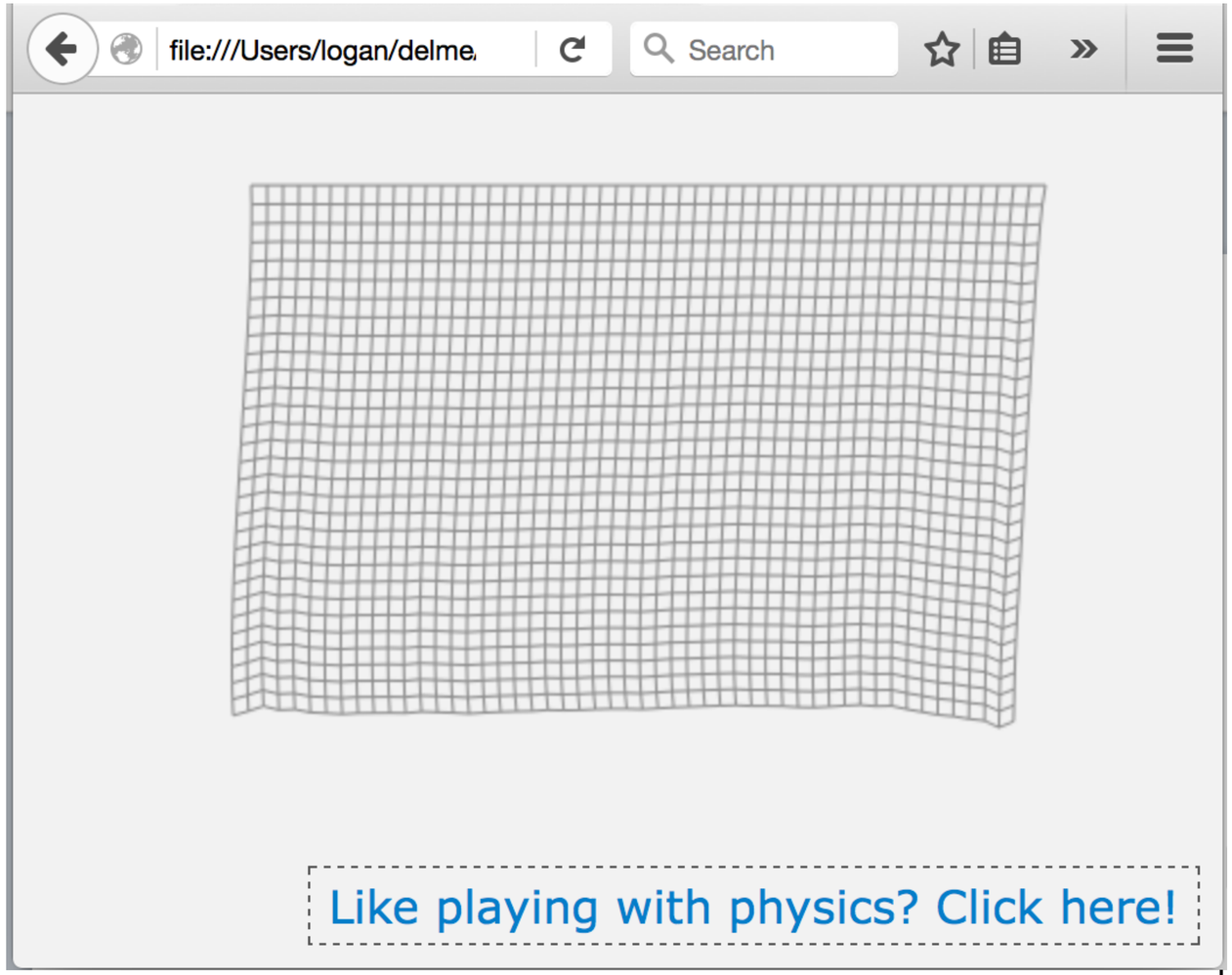
Slide borrowed from Michigan Robotics autorob.org



http://codepen.io/dissimulate/pen/KrAwx

Slide borrowed from Michigan Robotics autorob.org

Verlet Simulated Mass-Spring Cloth



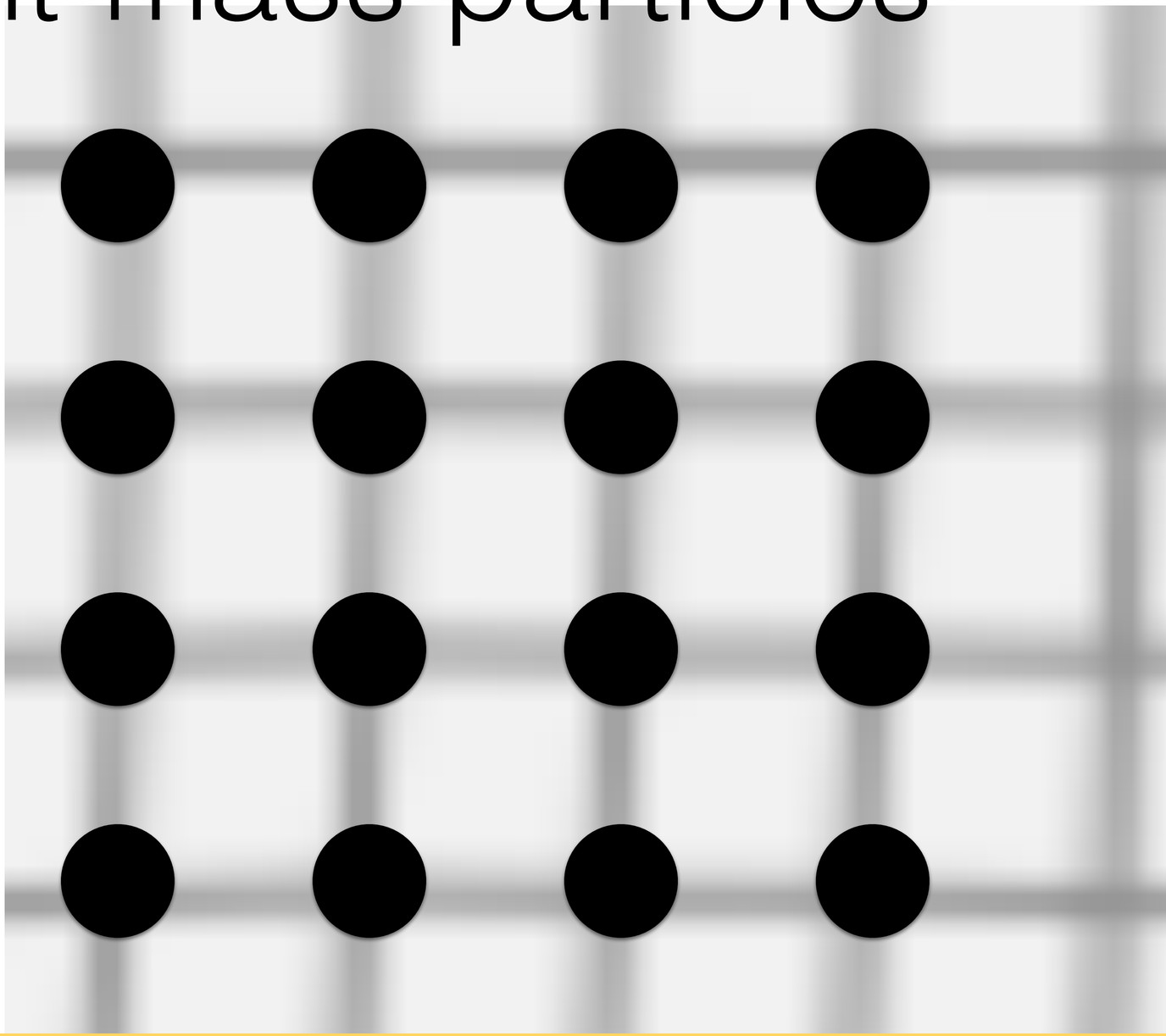
http://codepen.io/dissimulate/pen/KrAwx

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

https://github.com/Dissimulate/Tearable-Cloth

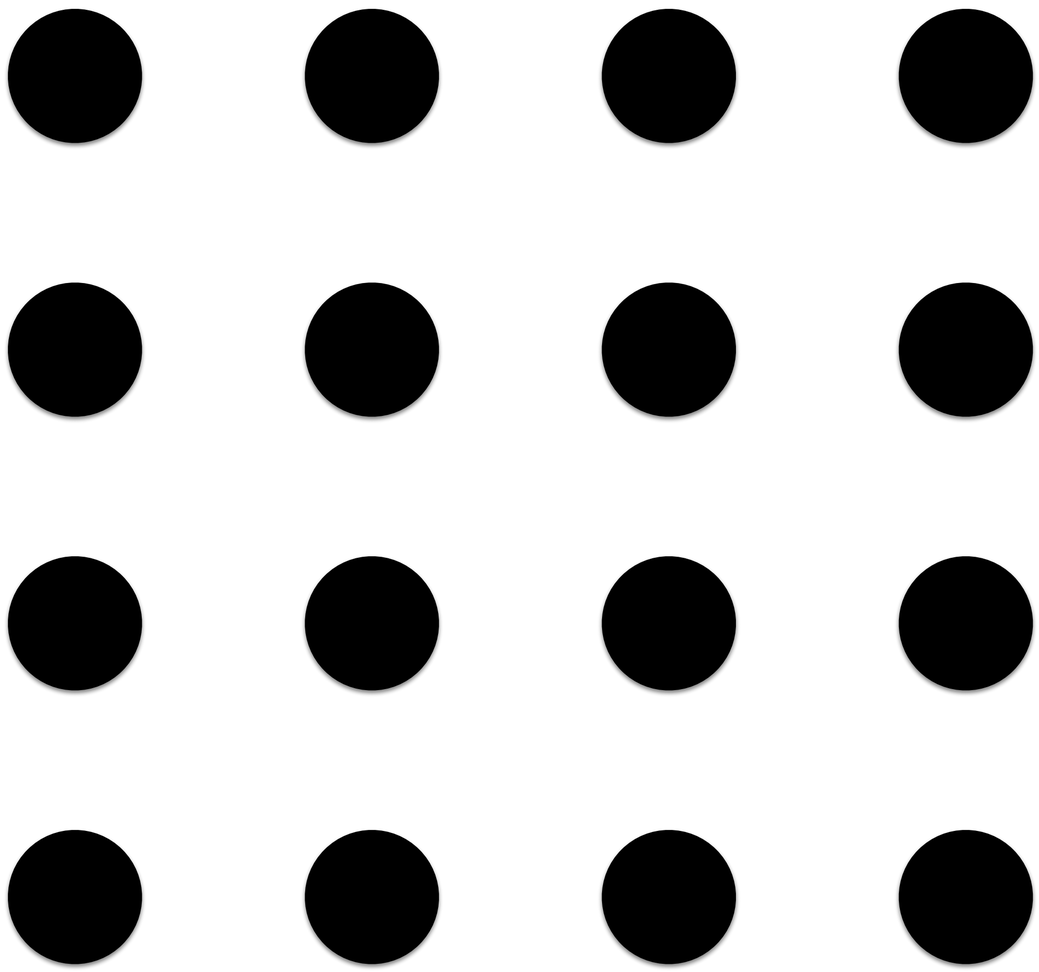
Slide borrowed from Michigan Robotics autorob.org

Start with point-mass particles



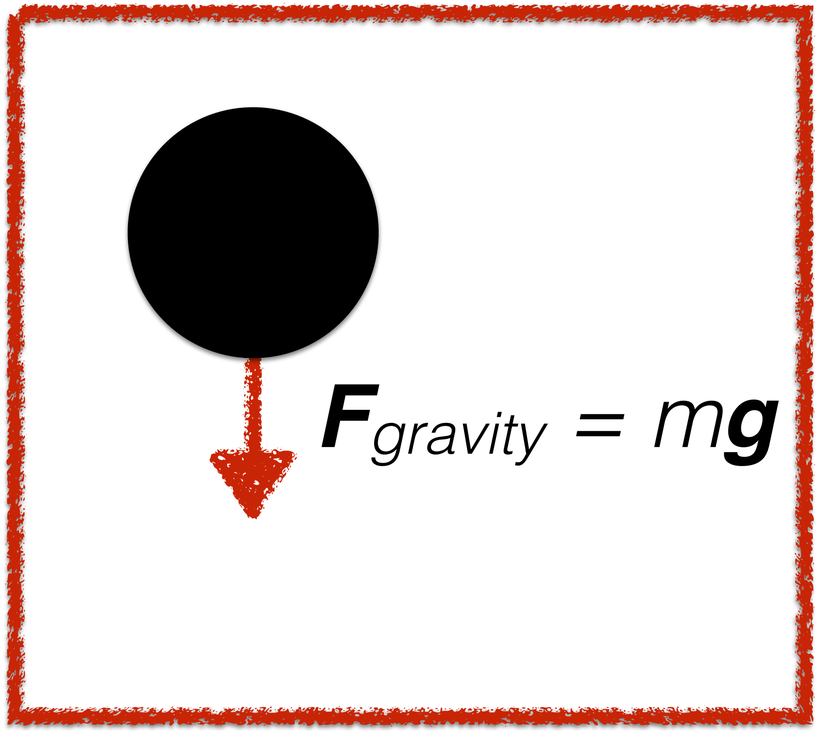
Slide borrowed from Michigan Robotics autorob.org

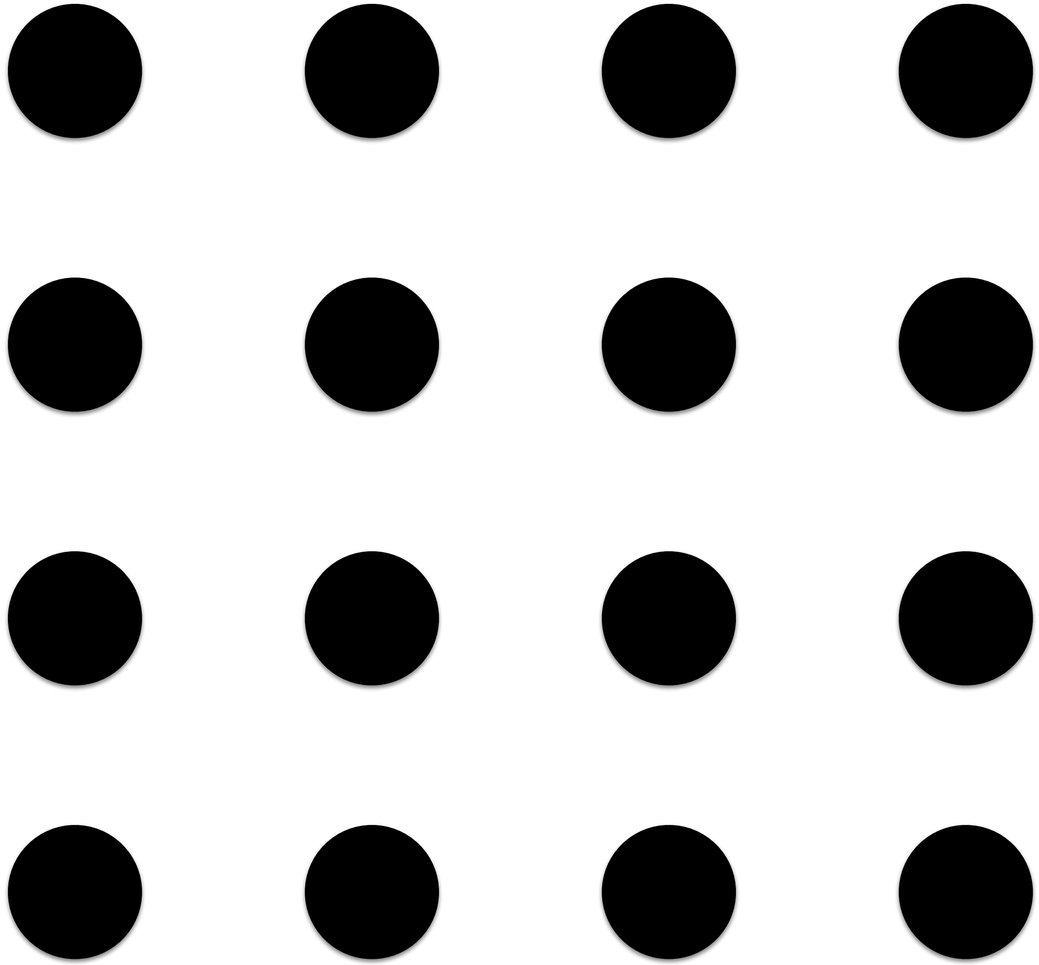
Start with point-mass particles



Slide borrowed from Michigan Robotics autorob.org

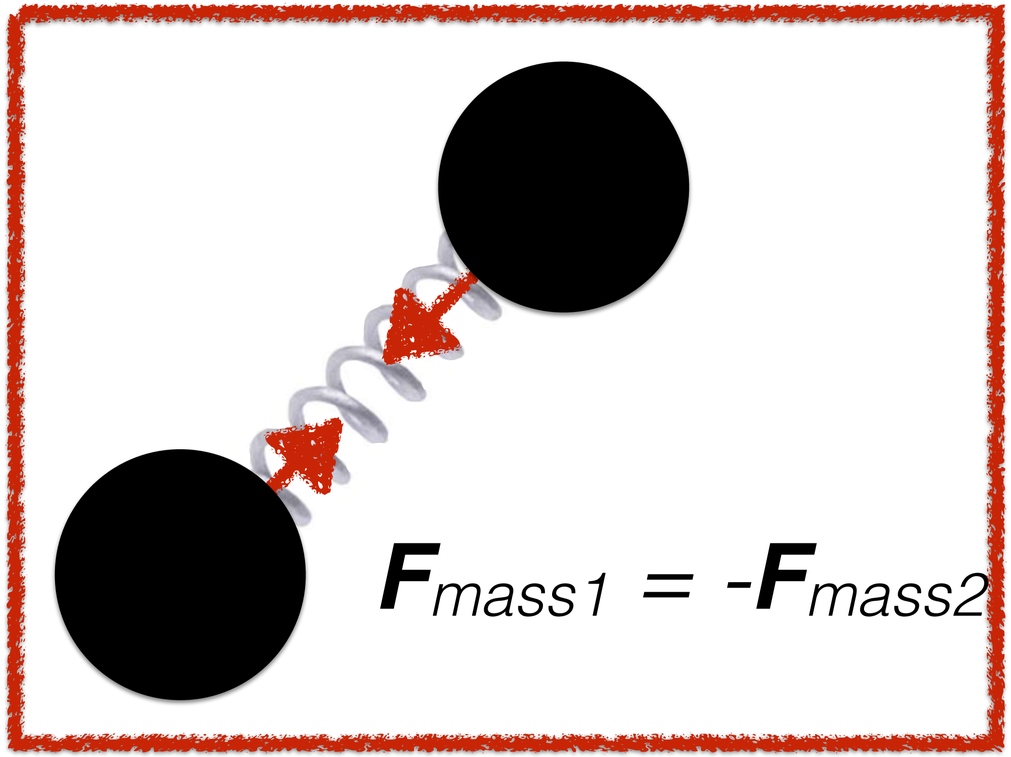
Apply gravity



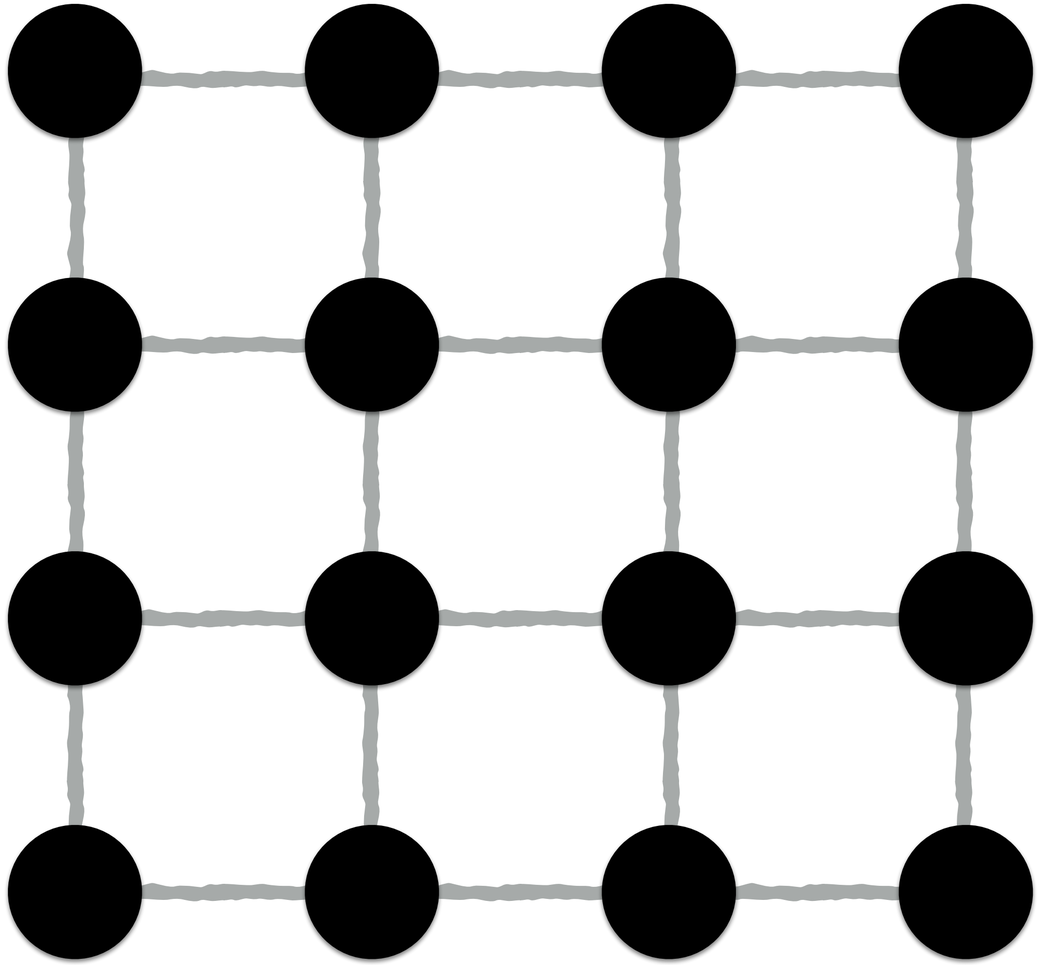


Slide borrowed from Michigan Robotics autorob.org

Enforce spring force constraints

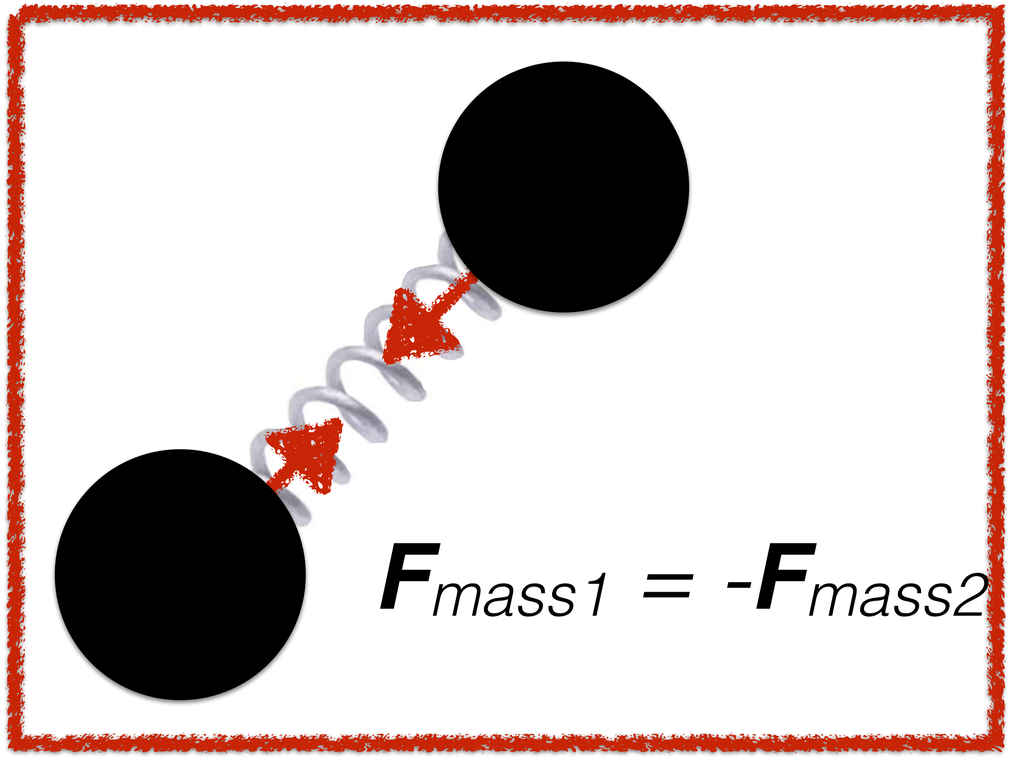


spring with stiffness 1
and rest length *r*



Slide borrowed from Michigan Robotics autorob.org

Enforce spring force constraints



spring with stiffness 1
and rest length *r*

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

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

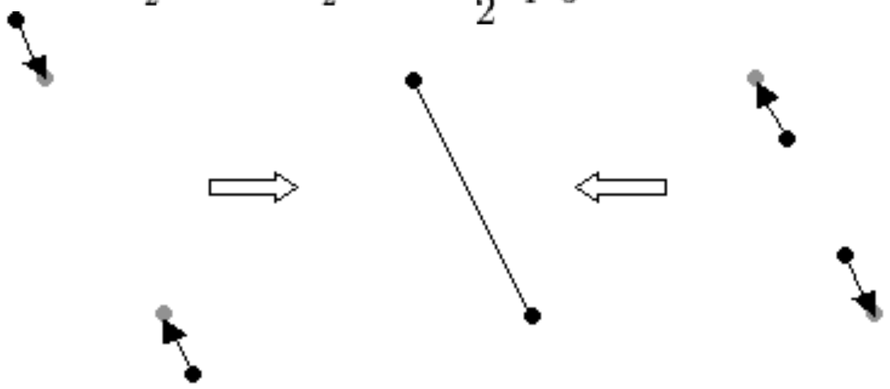
$$d_2 = \|d_1\|$$ Distance between particles

$$d_3 = \frac{d_2 - r}{d_2}$$ Distance from rest

$$x_1^{(t+\Delta t)} = \tilde{x}_1^{(t+\Delta t)} + \frac{1}{2} d_1 d_3$$

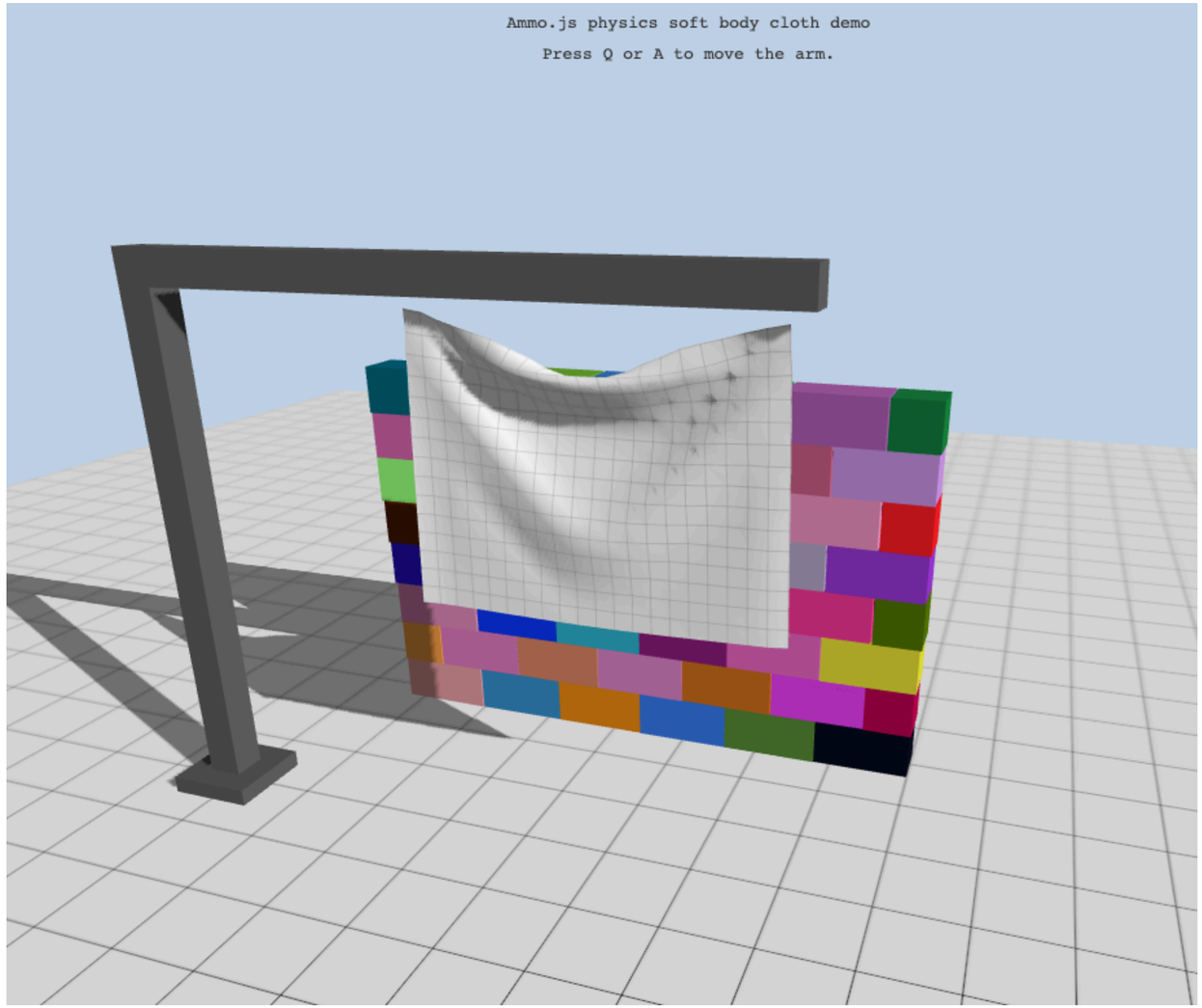
$$x_2^{(t+\Delta t)} = \tilde{x}_2^{(t+\Delta t)} - \frac{1}{2} d_1 d_3$$

Correction towards rest length



Dist. too large | Correct distance | Dist. too small

Slide borrowed from Michigan Robotics autorob.org



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

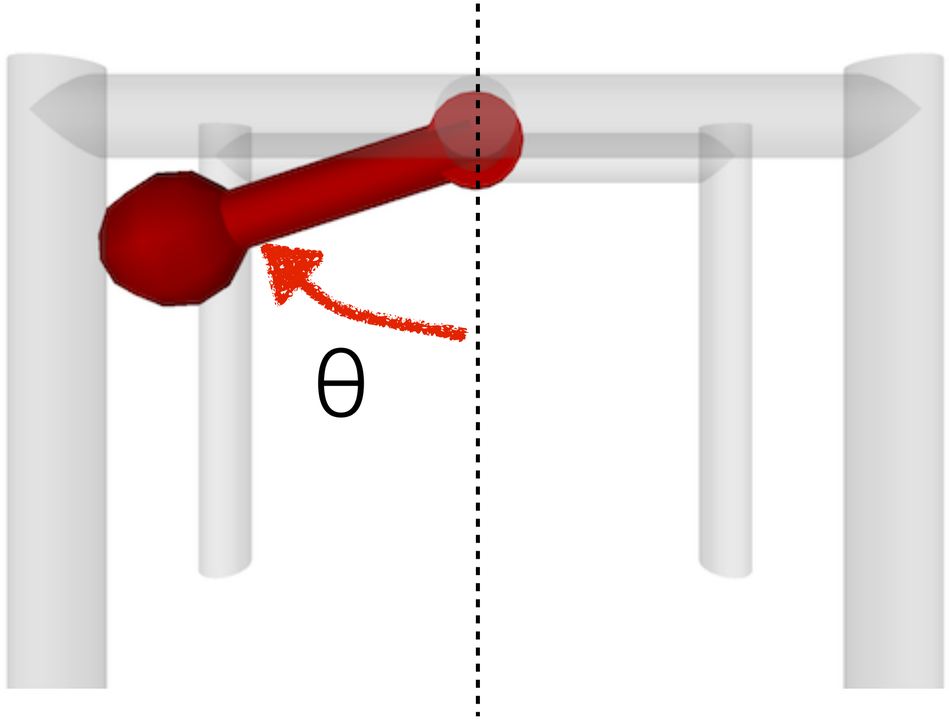
Slide borrowed from Michigan Robotics autorob.org

Pendulum

Slide borrowed from Michigan Robotics autorob.org

Pendulum

Generalized coordinates

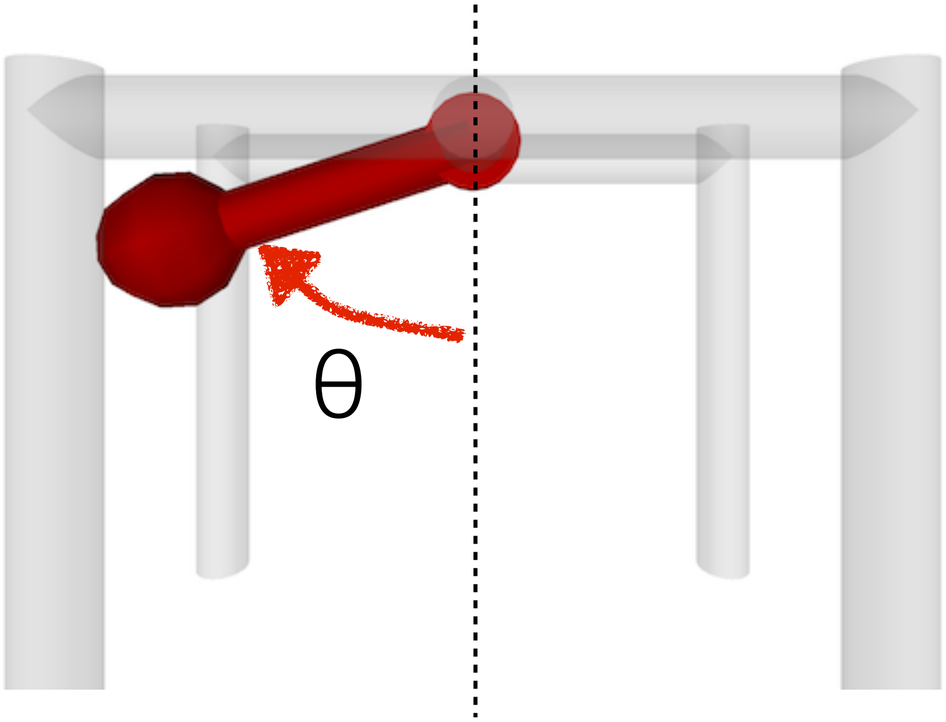


Simulate the motion of DOFs (pendulum angle) directly

Slide borrowed from Michigan Robotics autorob.org

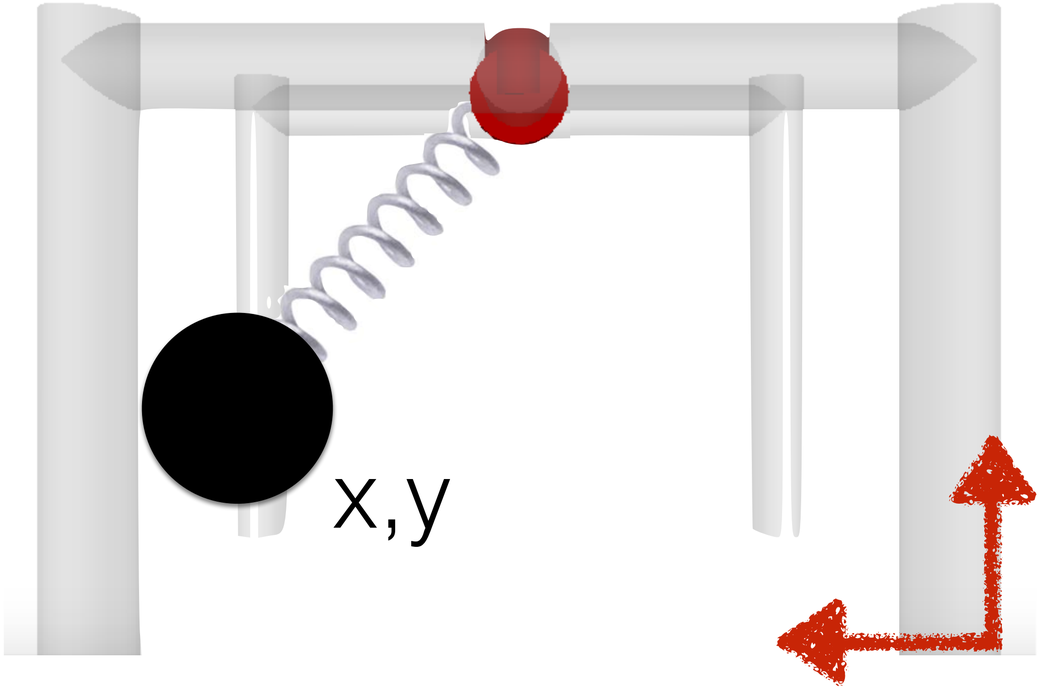
Pendulum

Generalized coordinates



Simulate the motion of DOFs (pendulum angle) directly

Maximal coordinates

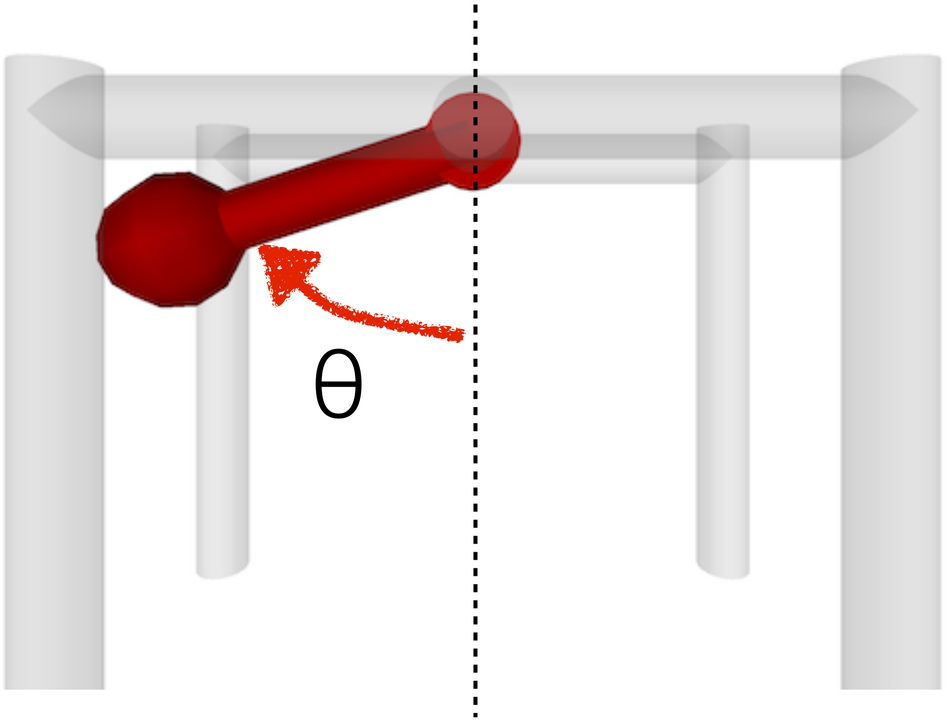


Simulate the motion of individual bodies (pendulum bob location) corrected by constraints (spring)

Slide borrowed from Michigan Robotics autorob.org

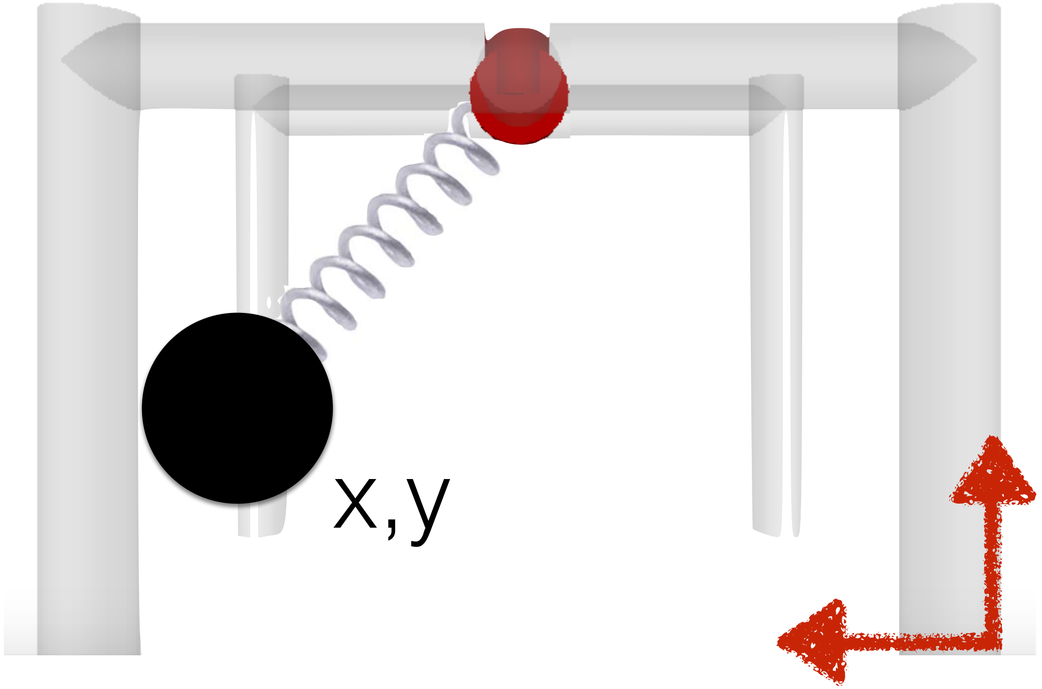
Why use generalized coordinates?

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)

Slide borrowed from Michigan Robotics autorob.org

Why use generalized coordinates?

Equation of motion
(with rotational inertia *I*)

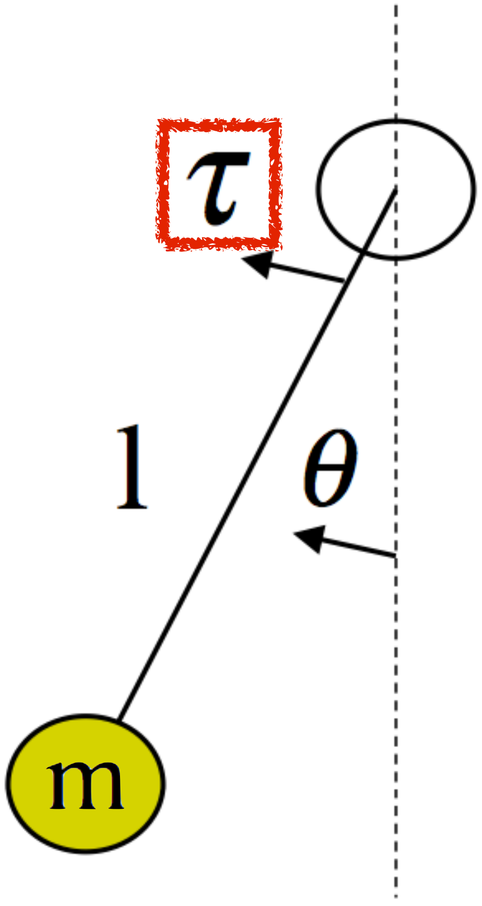
$$I\ddot{\theta} = -mgl\sin(\theta) + \tau$$

with Parallel Axis Theorem (I=mr^2)

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

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

Slide borrowed from Michigan Robotics autorob.org

Motor force expressed via DOFs

Equation of motion
(with rotational inertia *I*)

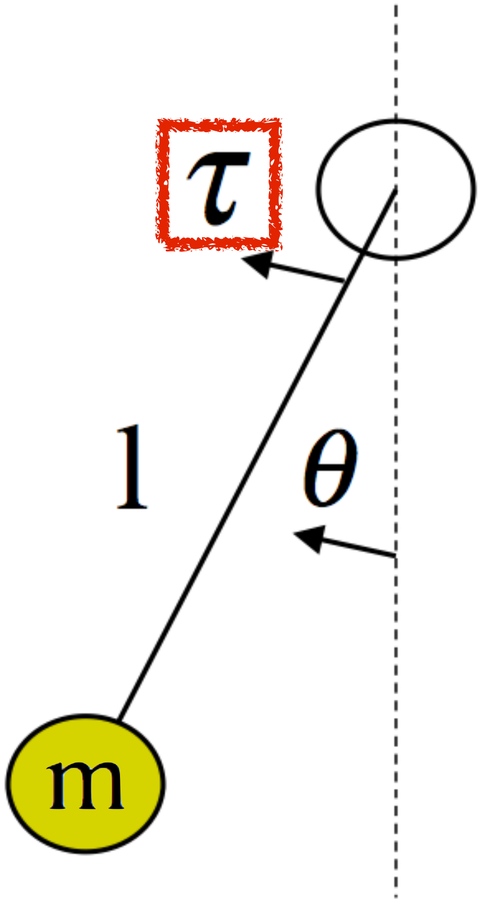
$$I\ddot{\theta} = -mgl\sin(\theta) + \tau$$

with Parallel Axis Theorem (I=mr^2)

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

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)

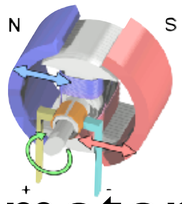
Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

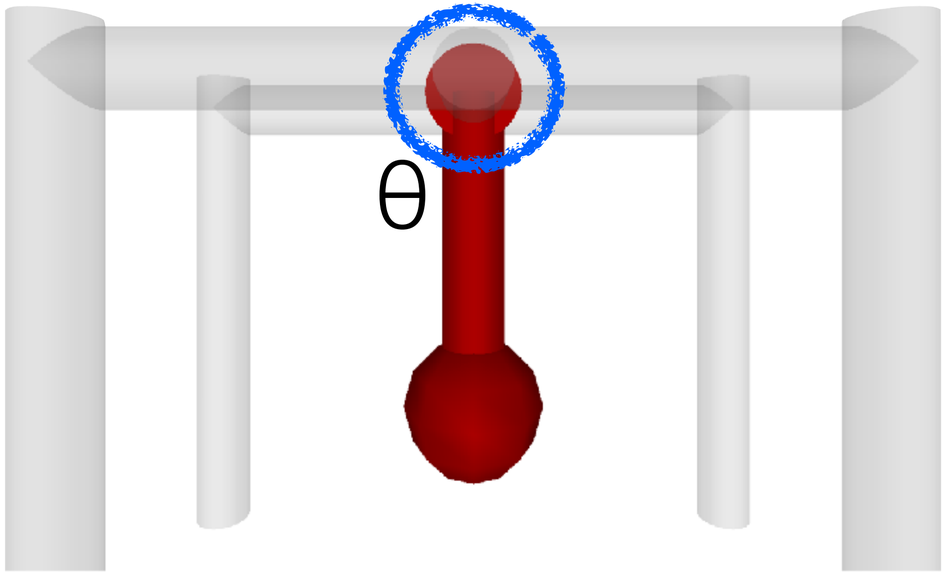
Slide borrowed from Michigan Robotics autorob.org

Motor force expressed via DOFs

t = 50.15 dt = 0.05 x = -1.26 x_dot = 0.00 x_desired = -1.26 u = -37.32 m = 2.00 l = 2.00 integrator = velocity verlet



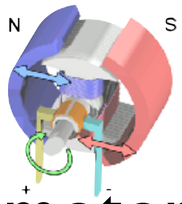
assume 1DOF motor at pendulum axis



Slide borrowed from Michigan Robotics autorob.org

Can we control the pendulum?

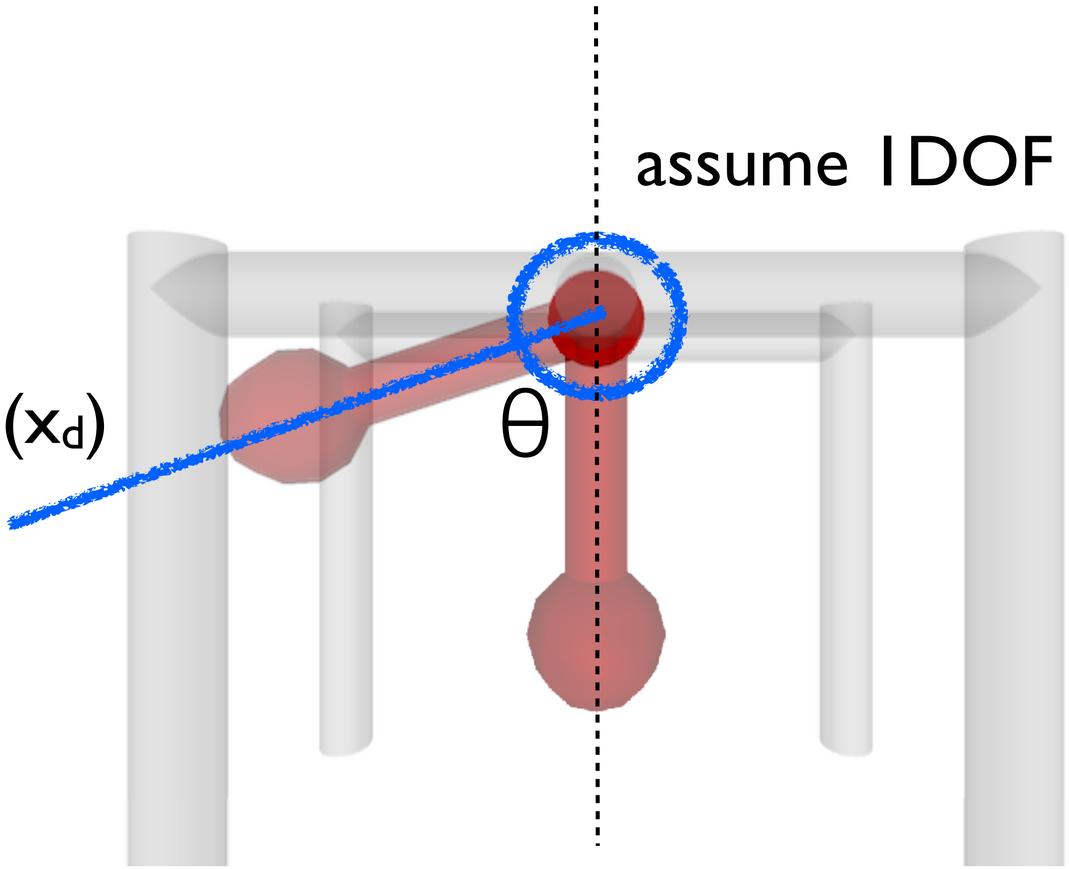
t = 50.15 dt = 0.05 x = -1.26 x_dot = 0.00 x_desired = -1.26 u = -37.32 m = 2.00 l = 2.00 integrator = velocity verlet



assume 1DOF motor at pendulum axis

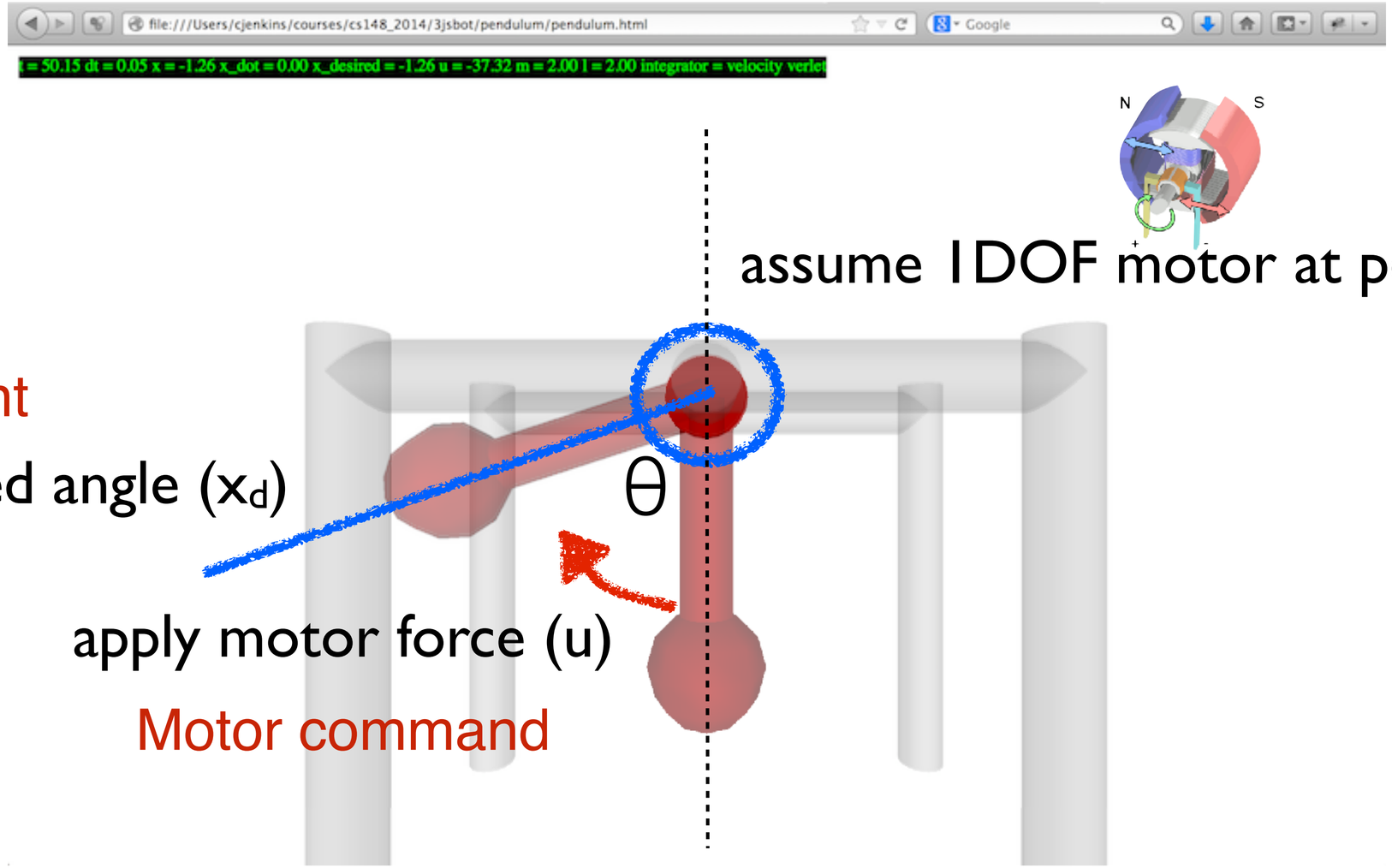
Setpoint

maintain desired angle (x_d)



Slide borrowed from Michigan Robotics autorob.org

Can we control the pendulum?



assume 1DOF motor at pendulum axis

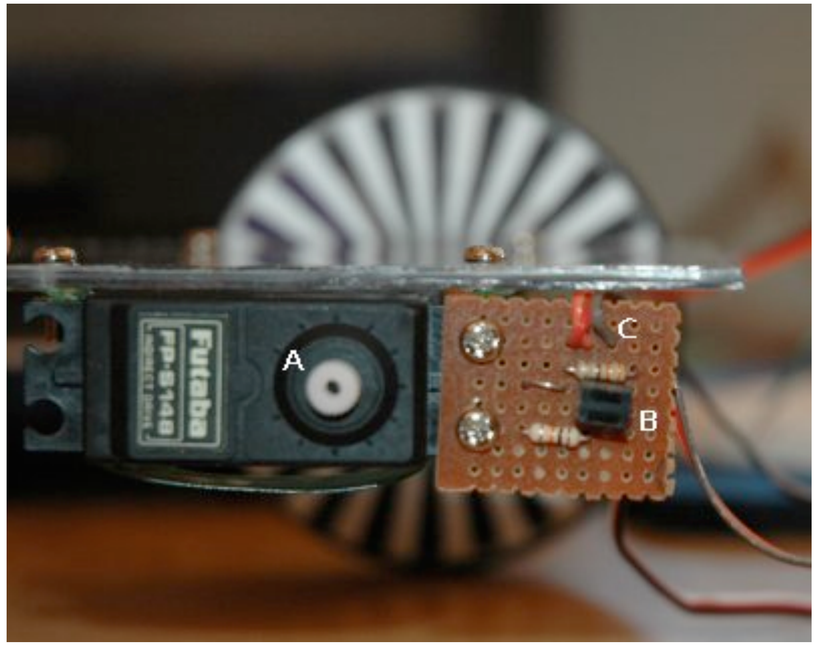
Setpoint

maintain desired angle (x_d)

apply motor force (u)

Motor command

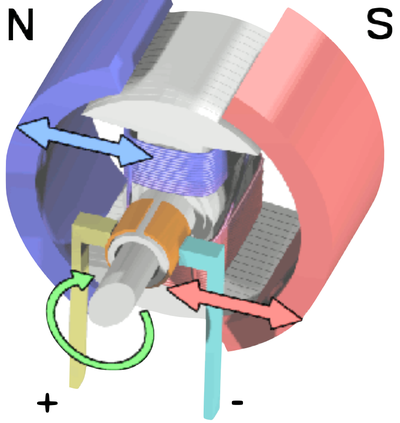
Slide borrowed from Michigan Robotics autorob.org

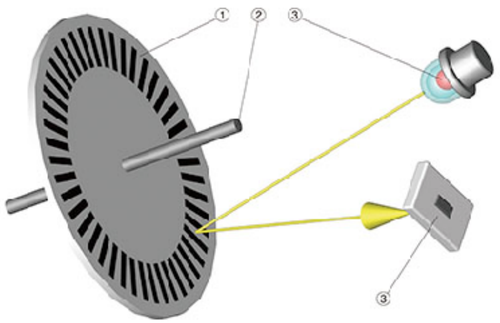


Servos have:

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

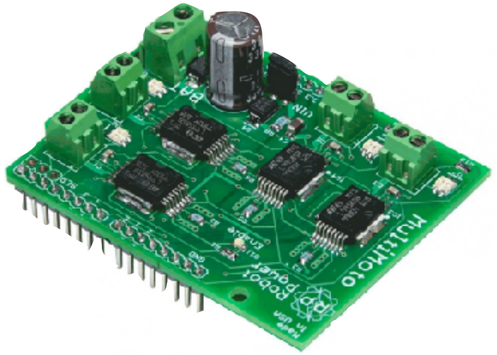
Actuator (electric)





Proprioception (optical encoder)

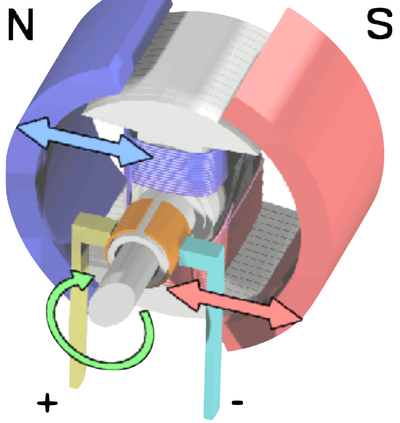
Motor Controller (4 channel H-Bridge)

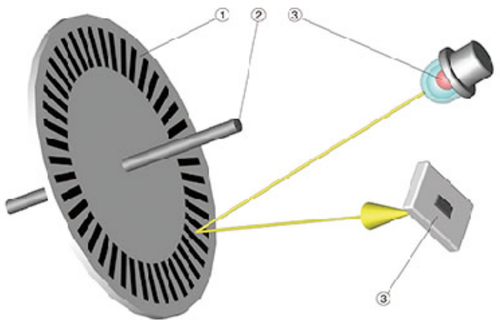


Slide borrowed from Michigan Robotics autorob.org

What is inside a servo?

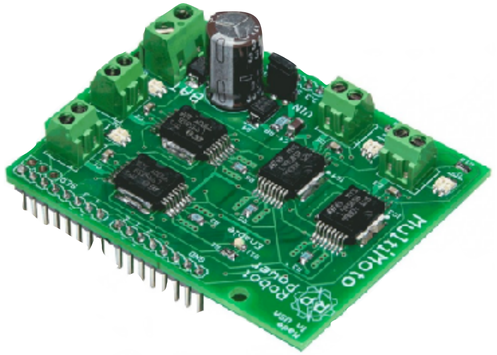
Actuator
(electric)





Proprioception
(optical encoder)

Motor Controller
(4 channel H-Bridge)

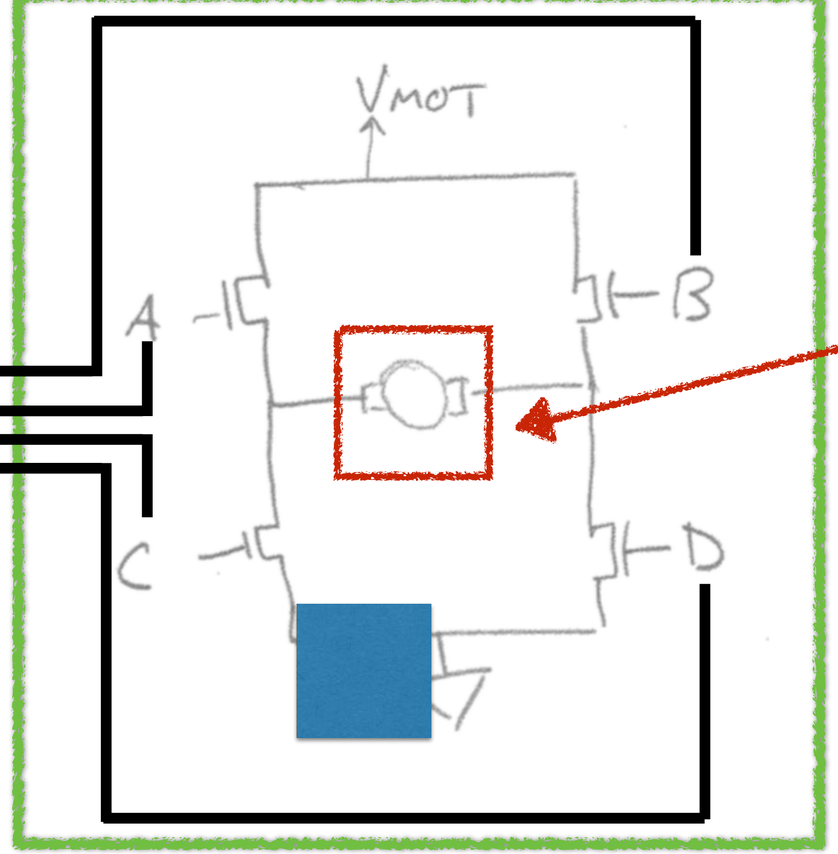


Slide borrowed from Michigan Robotics autorob.org

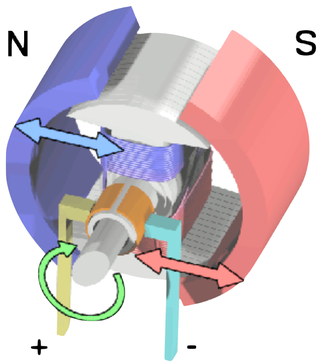
What is inside a servo?

Control Signal

PWM signal sent to H-bridge controller.

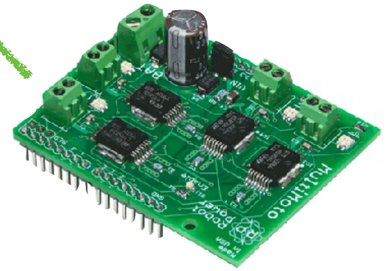


motor



Current actuates motor motion about axis.

H-bridge sends proportional current to motor.



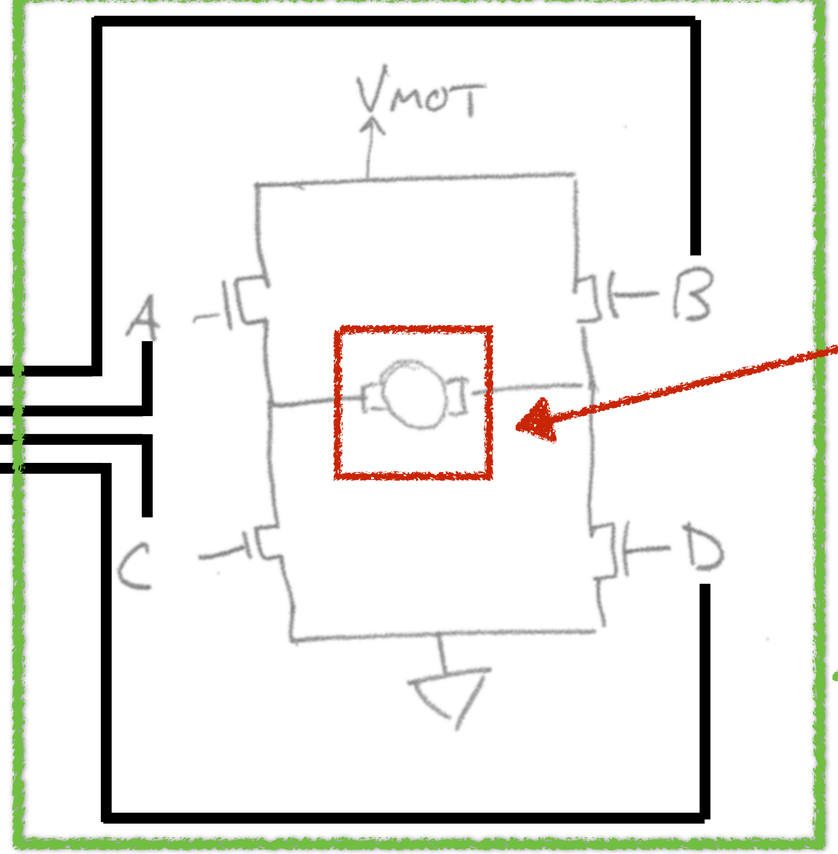
motor controller

Slide borrowed from Michigan Robotics autorob.org

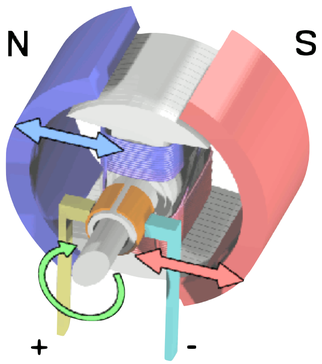
PWM
(Pulse Width Modulation)

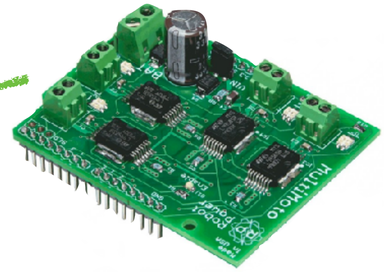
Control Signal

PWM signal sent to H-bridge controller.

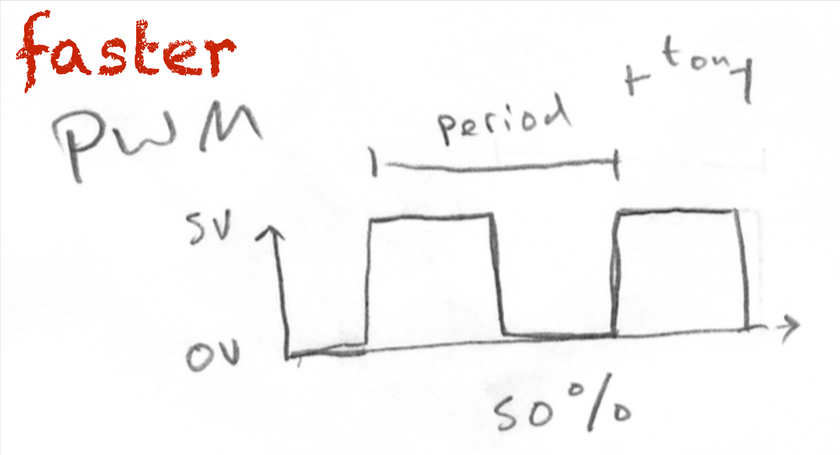


motor



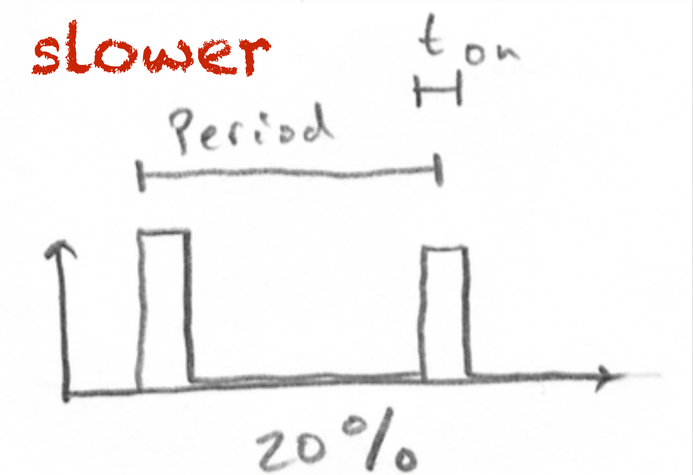


motor controller

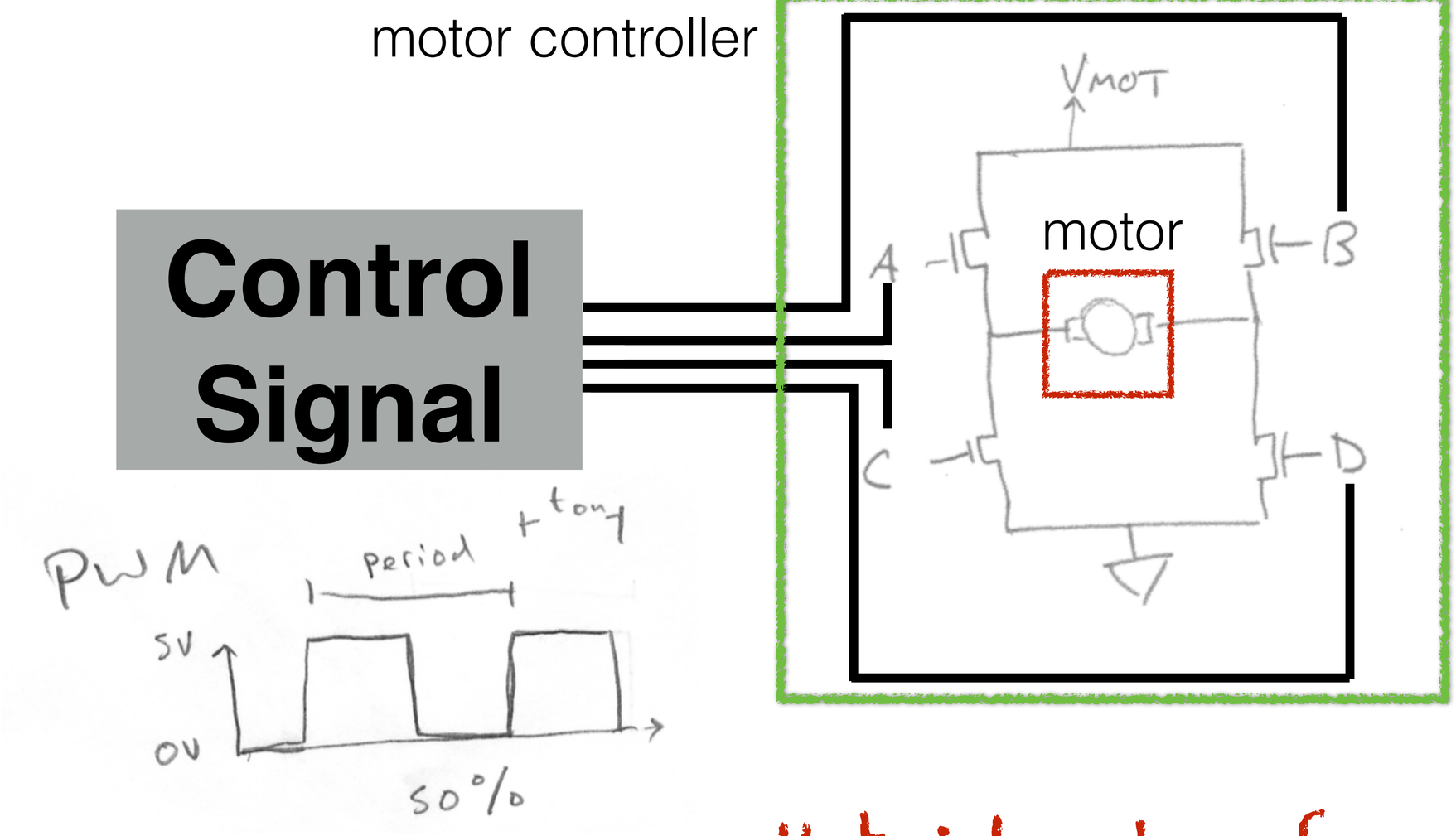


PWM has a periodic duty cycle on each line.

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



Slide borrowed from Michigan Robotics autorob.org

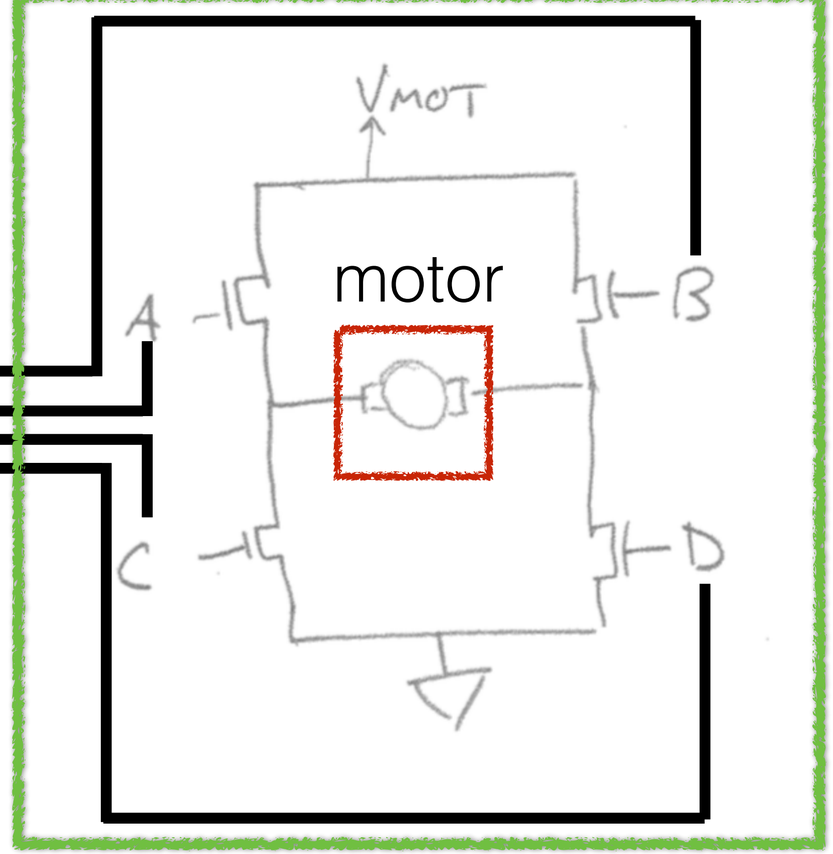


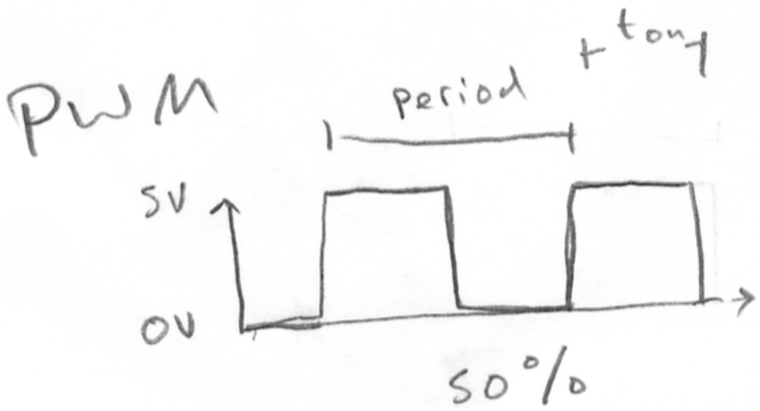
H-bridge has four switches that are "on" or "off"

Slide borrowed from Michigan Robotics autorob.org

motor controller

Control Signal





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

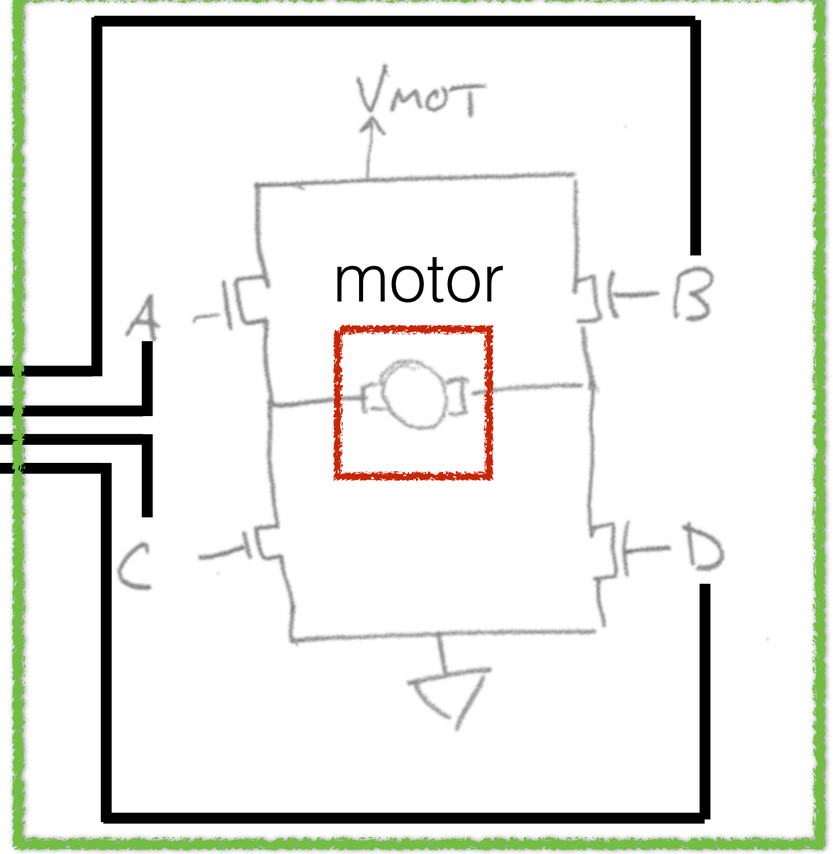
spin forward:
A and D on

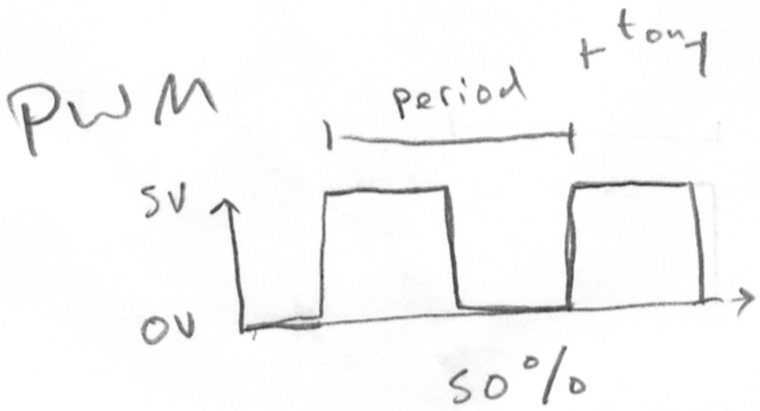
spin backward:
B and C on

Slide borrowed from Michigan Robotics autorob.org

motor controller

Control Signal





spin forward:
A and D

spin backward:
B and C

what if none are on?

[redacted]

what if C and D?

[redacted]

what if A and B?

[redacted]

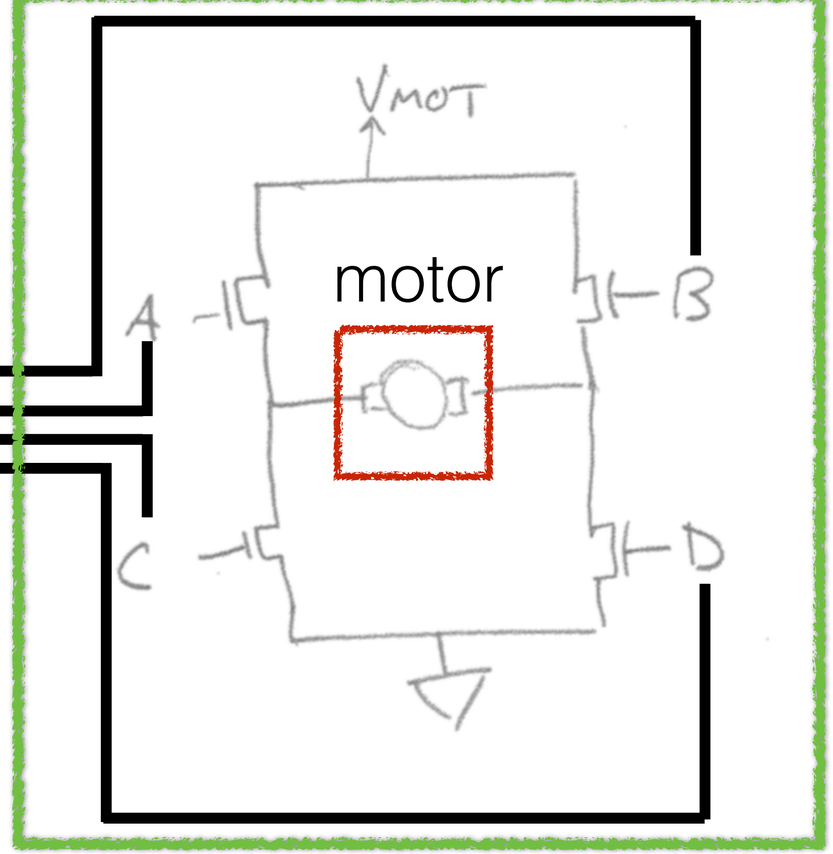
what if A and C?

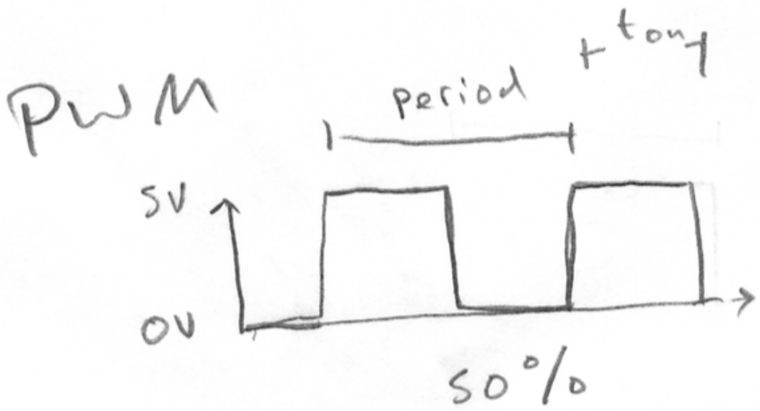
[redacted]

Slide borrowed from Michigan Robotics autorob.org

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?

[redacted]

what if A and B?

[redacted]

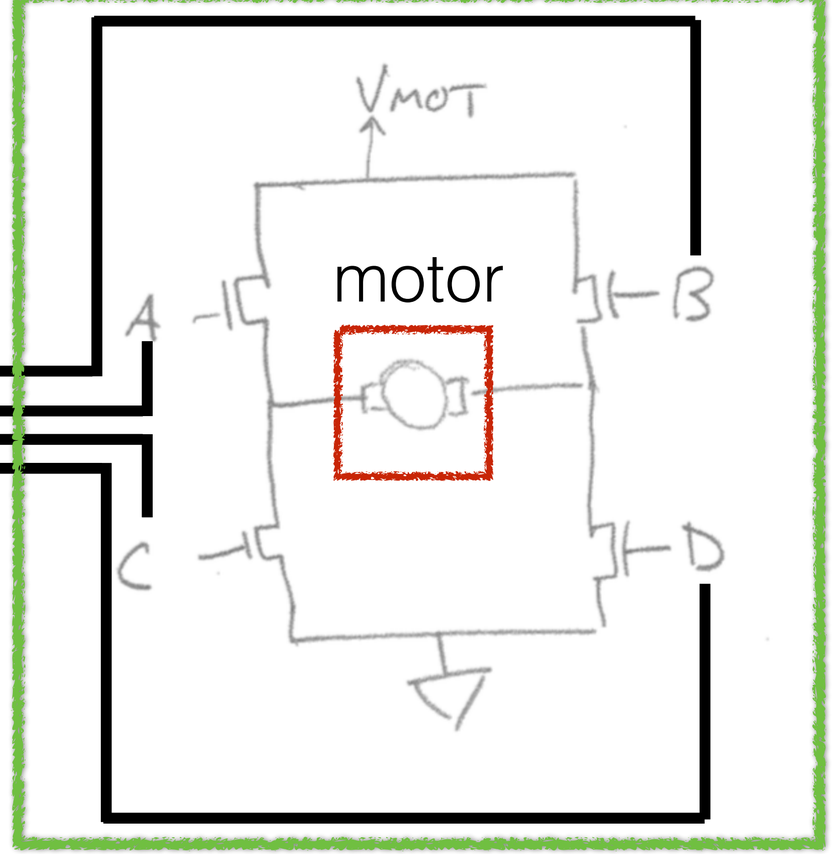
what if A and C?

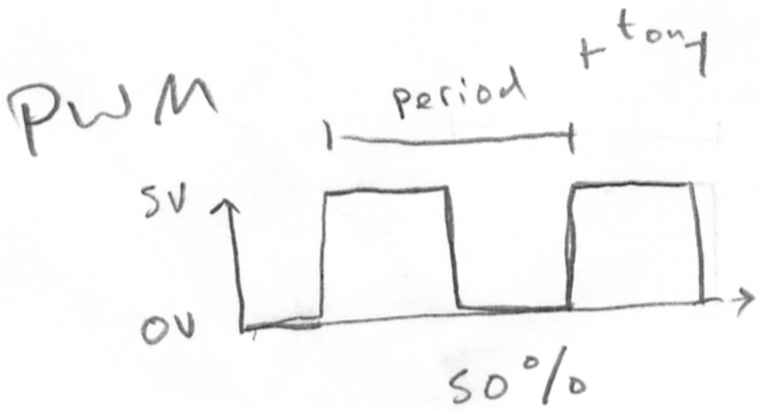
[redacted]

Slide borrowed from Michigan Robotics autorob.org

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?

[redacted]

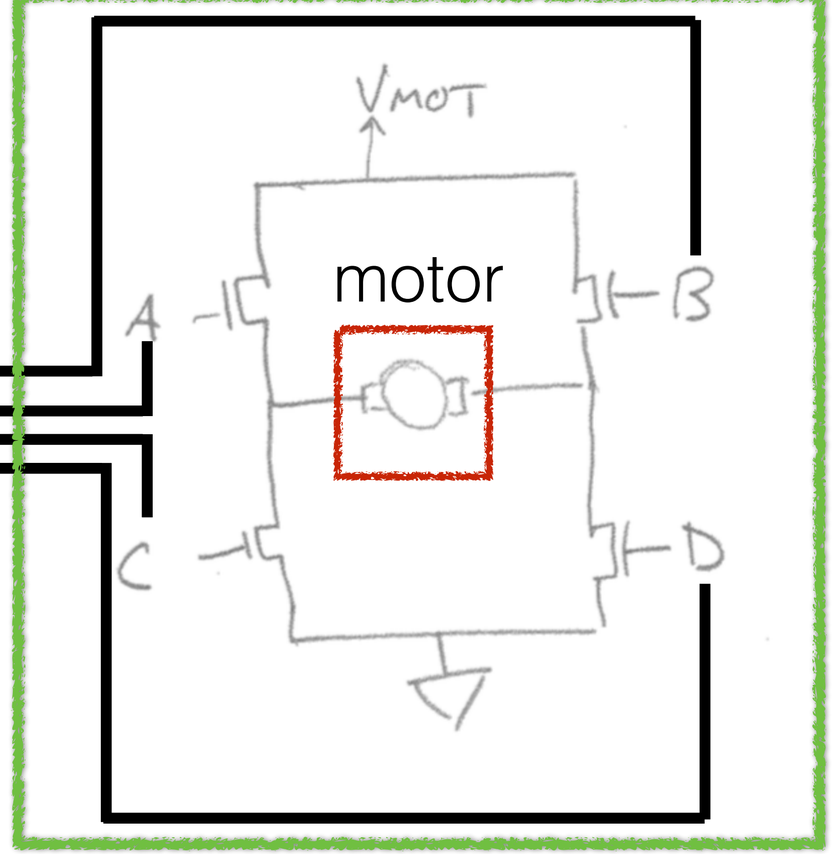
what if A and C?

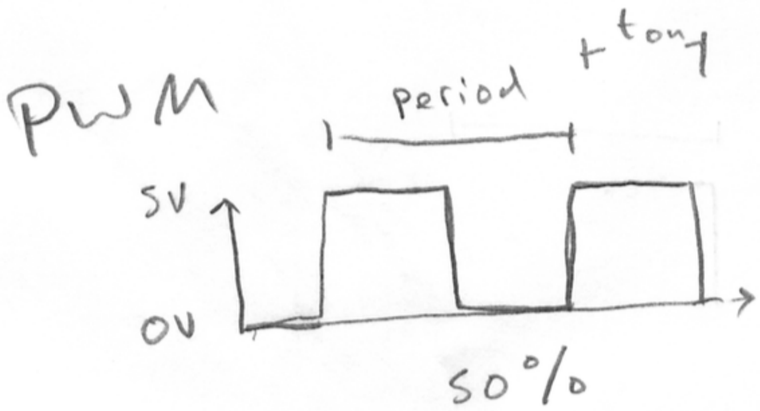
[redacted]

Slide borrowed from Michigan Robotics autorob.org

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

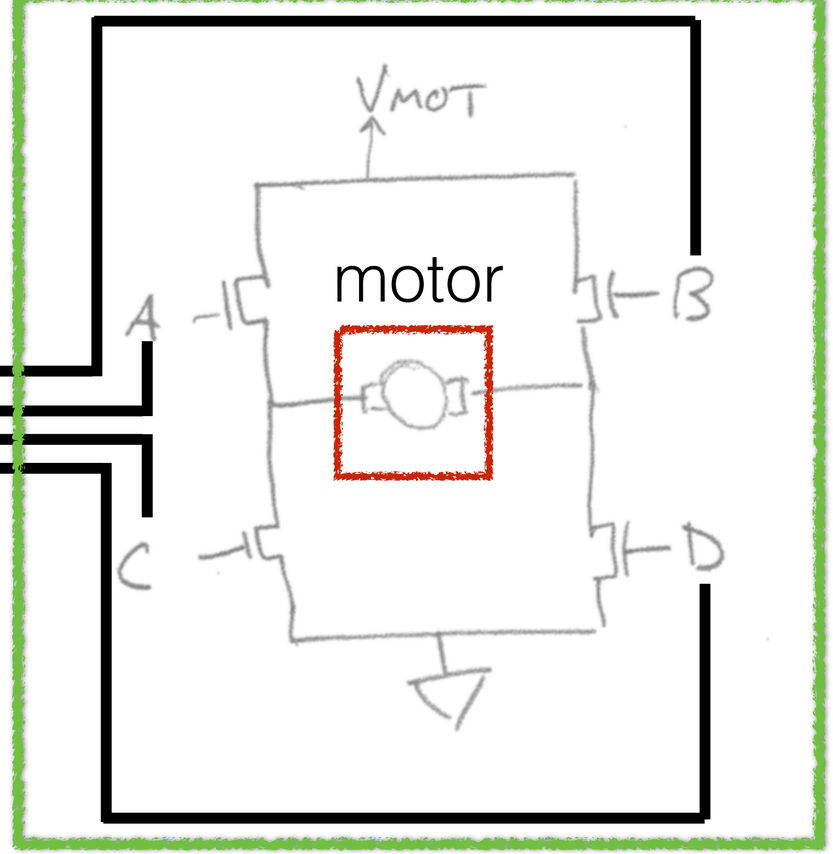
what if A and C?

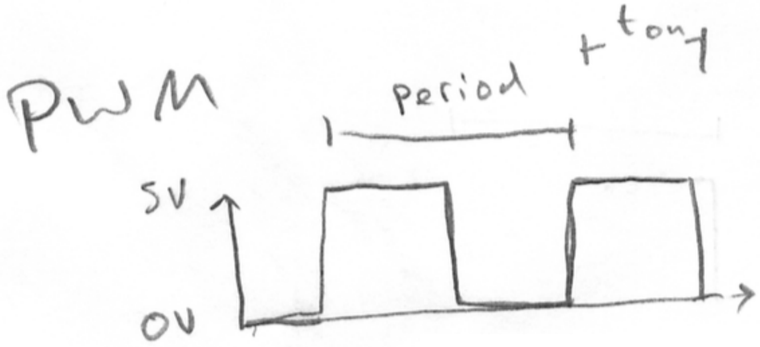
[redacted]

Slide borrowed from Michigan Robotics autorob.org

motor controller

Control Signal





No more robot, (our hardware prevents this)

what if A and C?

spin forward:
A and D

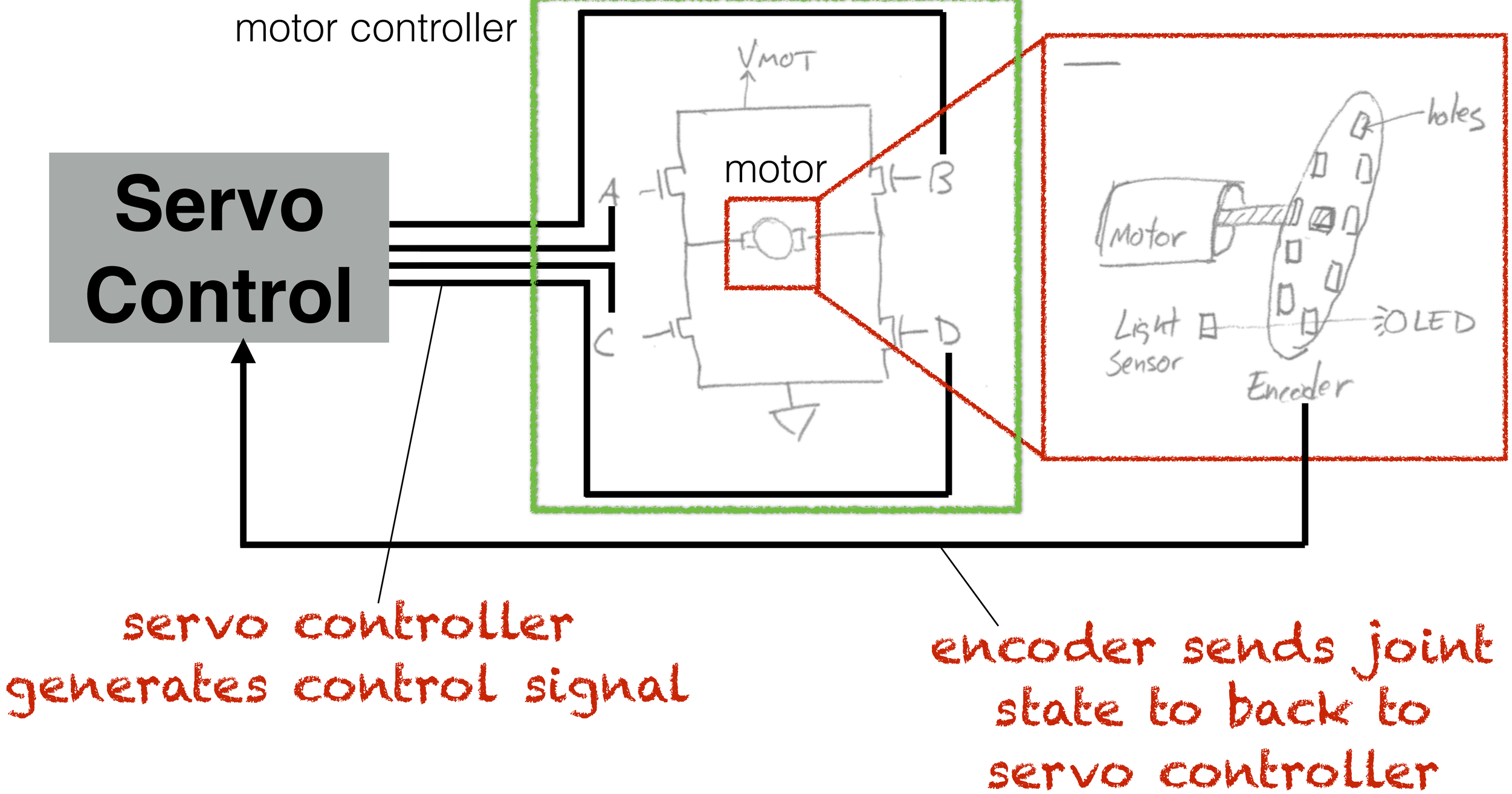
spin backward:
B and C

what if none are on?
open circuit
no current -> no torque

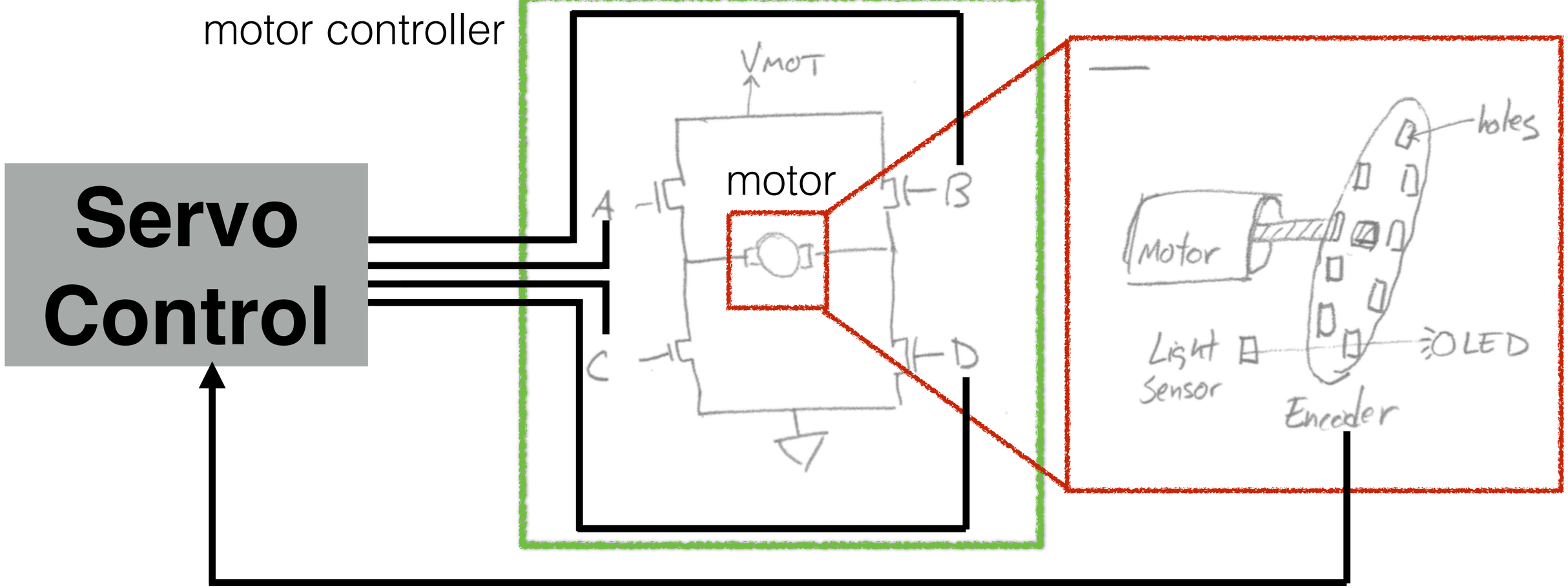
what if C and D?
negative torque -> brake

what if A and B?
negative torque -> brake

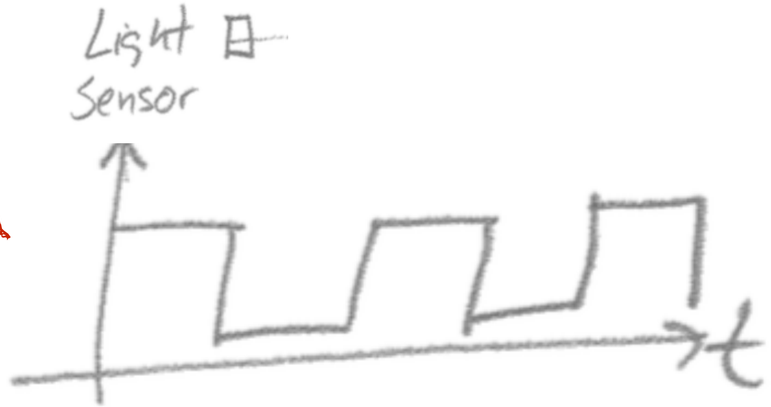
Slide borrowed from Michigan Robotics autorob.org



Slide borrowed from Michigan Robotics autorob.org

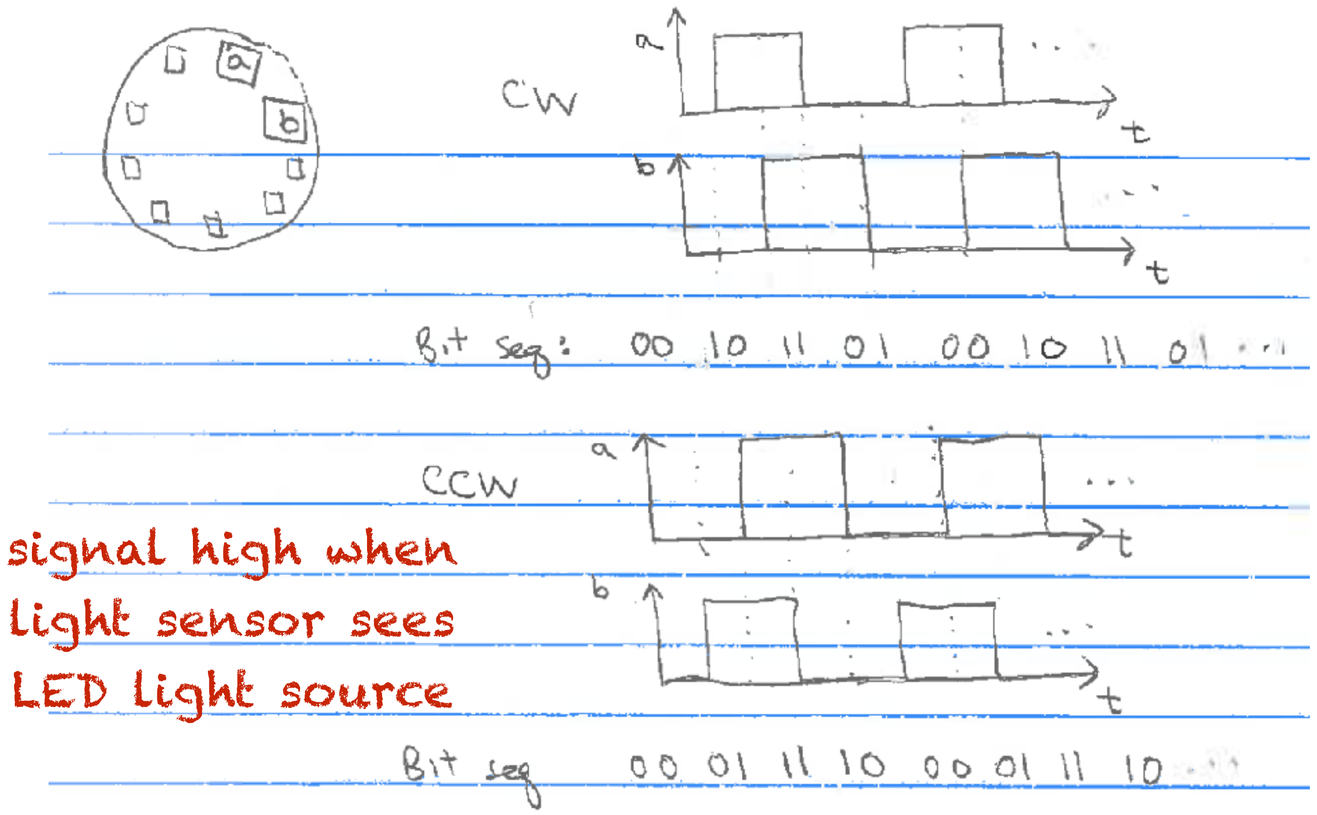


as motor turns,
encoder counts transitions between
sensed and occluded light

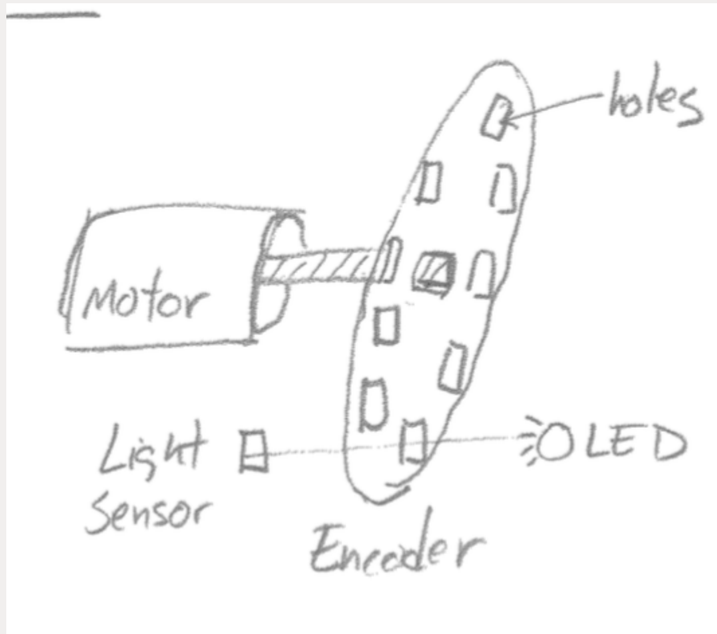


Slide borrowed from Michigan Robotics autorob.org

Quadrature Encoder



signal high when
light sensor sees
LED light source

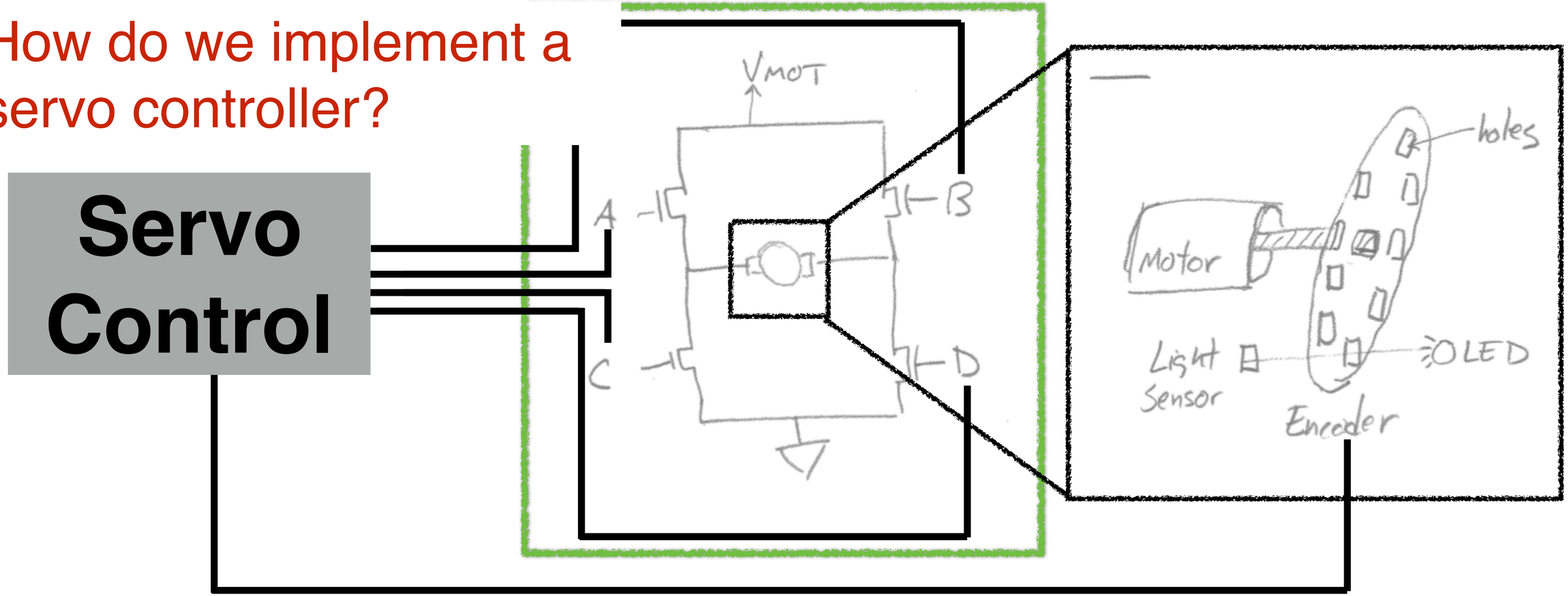


Encoders measure how much a motor has spun.

Quadrature encoders are commonly used. Why?

Slide borrowed from Michigan Robotics autorob.org

How do we implement a servo controller?

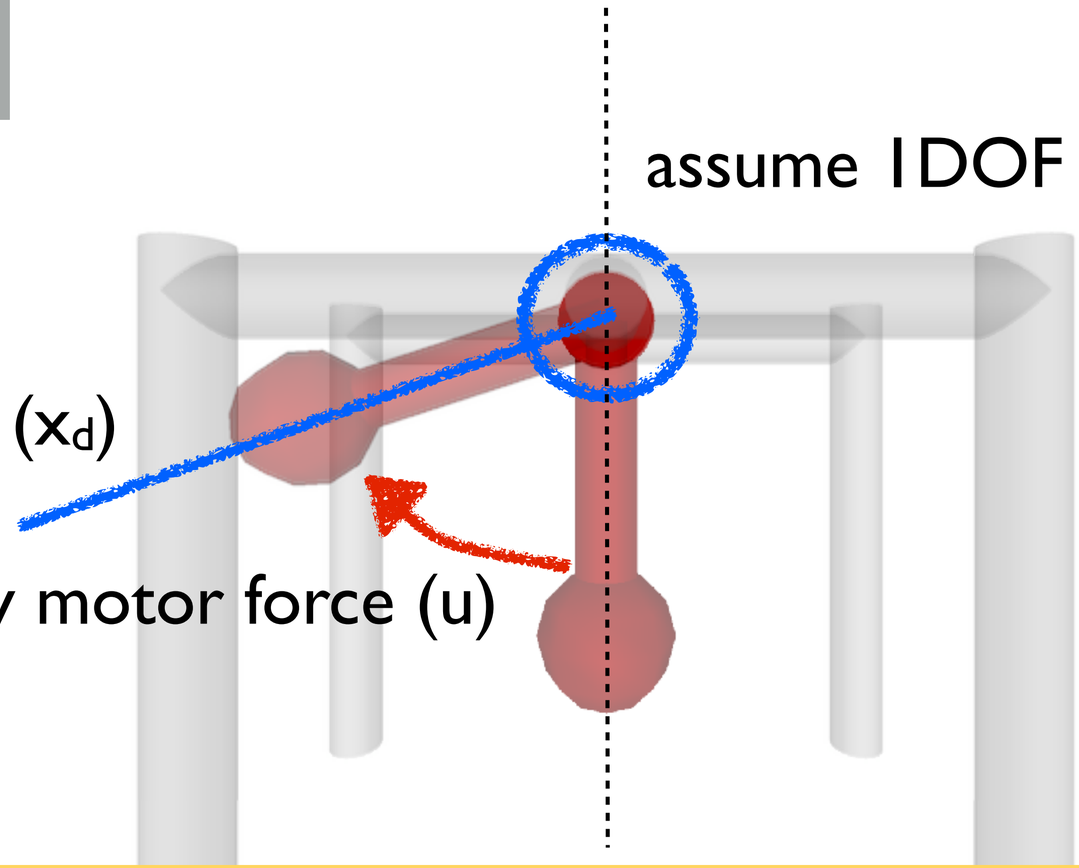


Slide borrowed from Michigan Robotics autorob.org

How do we implement a servo controller?

Servo Control

assume 1DOF motor at pendulum axis



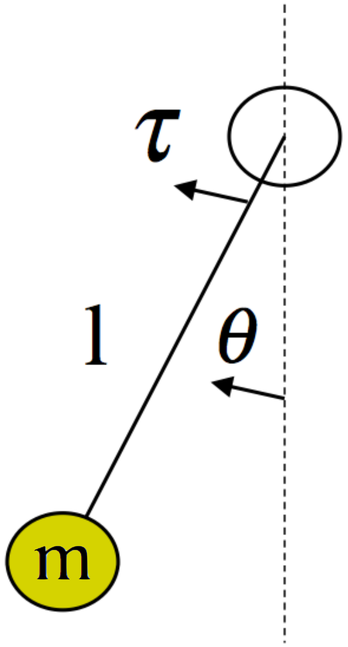
maintain desired angle (x_d)

apply motor force (u)

Slide borrowed from Michigan Robotics autorob.org

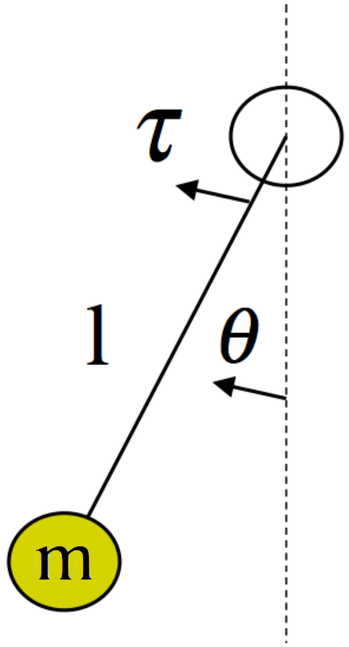
How do we implement a servo controller?

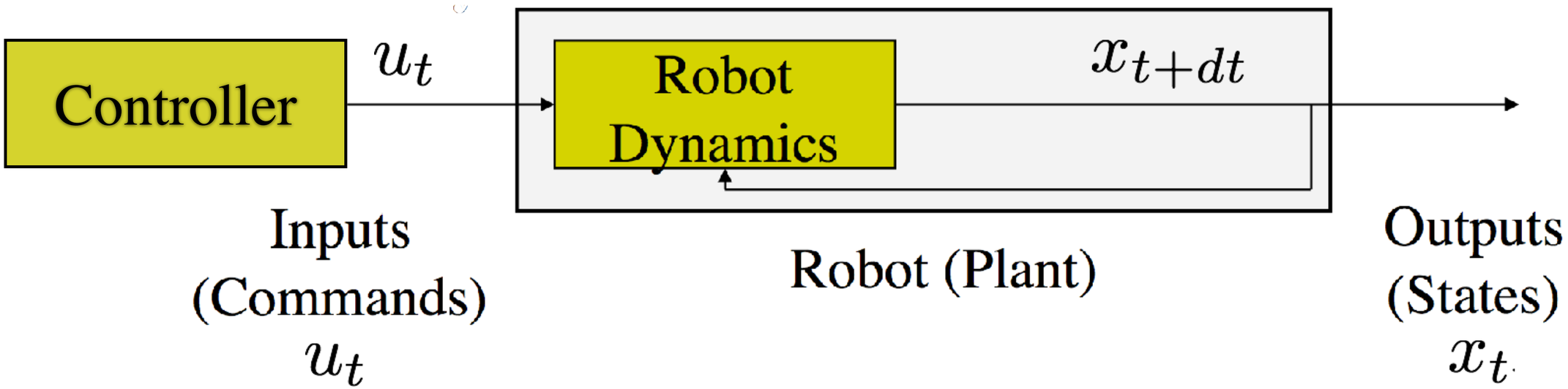
Servo Control



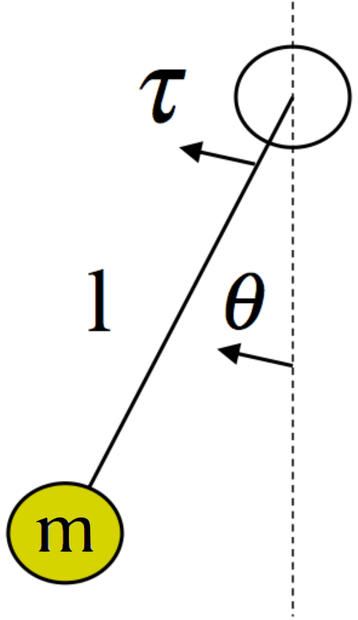
Slide borrowed from Michigan Robotics autorob.org

Servo Control





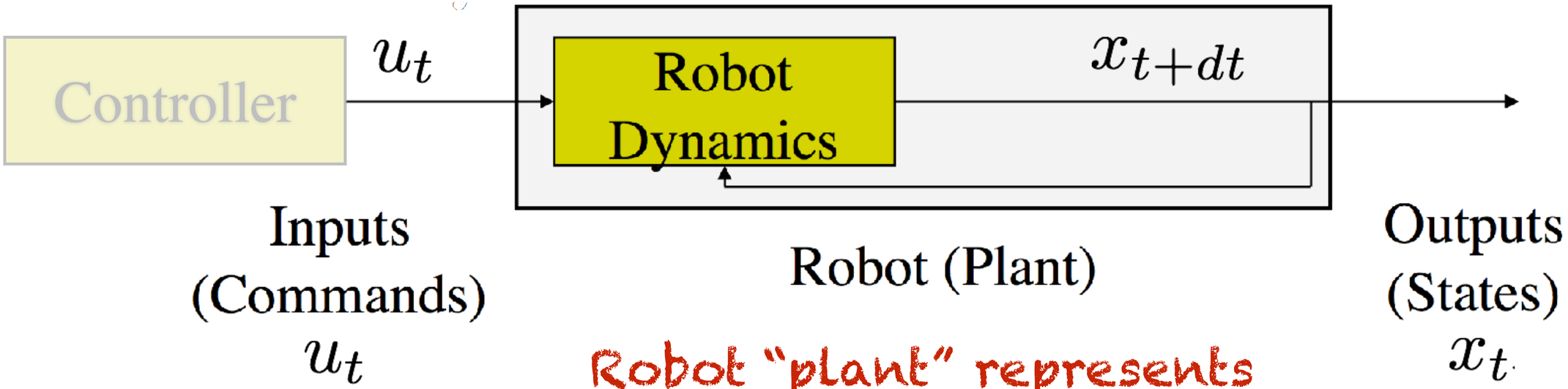
Slide borrowed from Michigan Robotics autorob.org



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

Motor command used within equations of motion

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

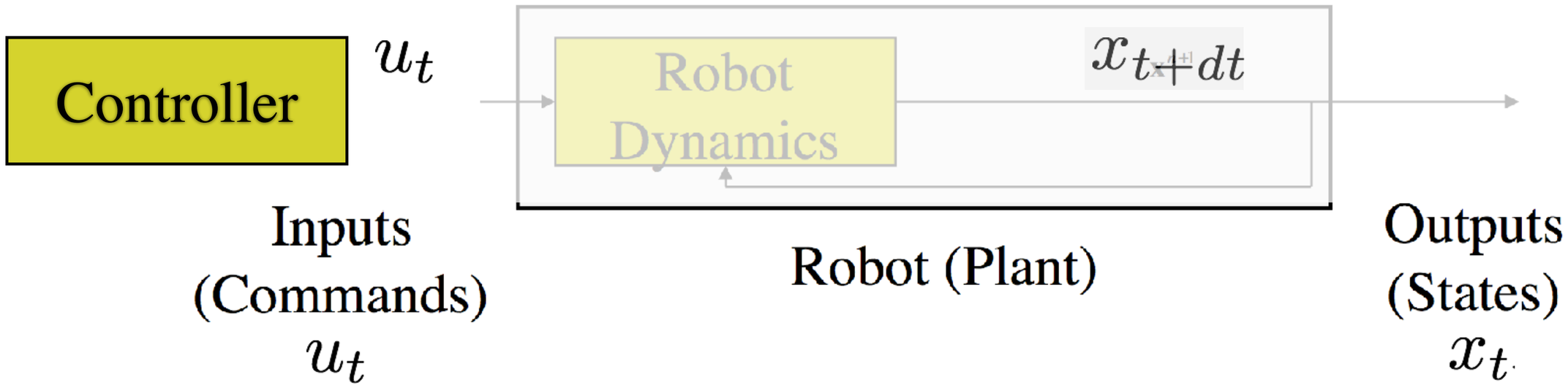


Robot "plant" represents system dynamics as function

Slide borrowed from Michigan Robotics autorob.org

Servo controller generates motor commands based on desired state

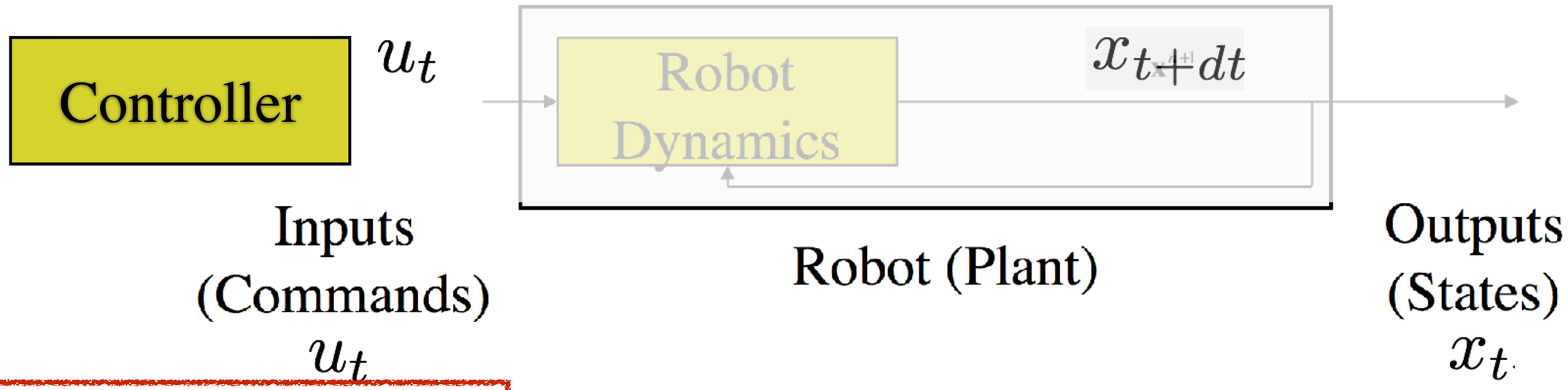
$$u = \pi(t, x, x_d) \qquad \dot{x} = f(t, x, u)$$



Slide borrowed from Michigan Robotics autorob.org

Servo controller generates motor commands based on desired state

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

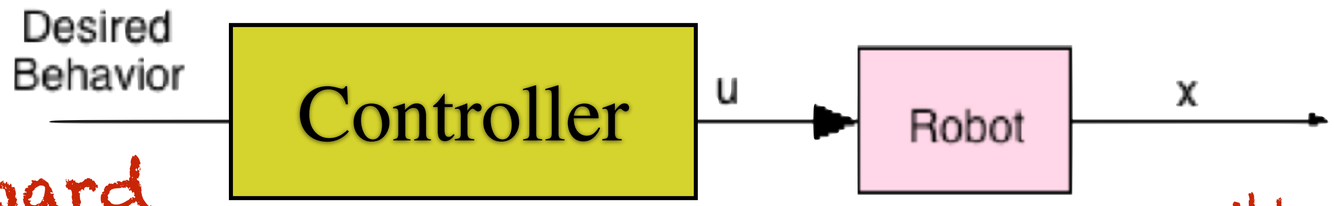


π is a "control policy"

Slide borrowed from Michigan Robotics autorob.org

Control Policies

Open Loop Control



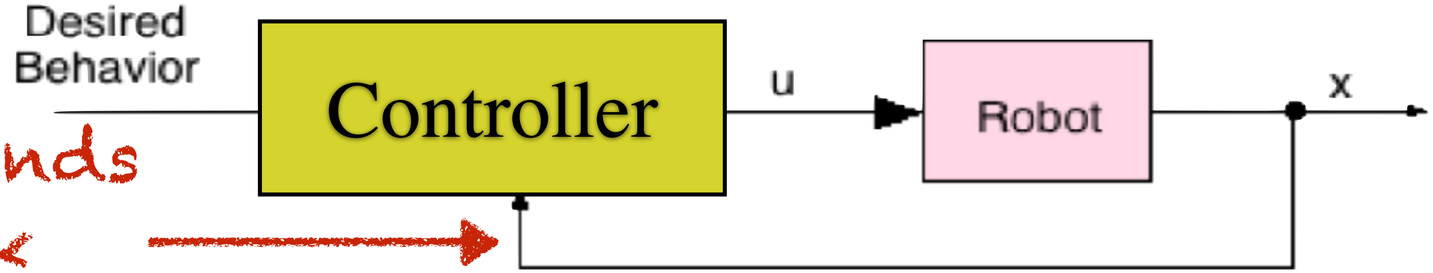
Acts without regard to environment

possible controller parameters

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

possible time dependence

Closed Loop Control

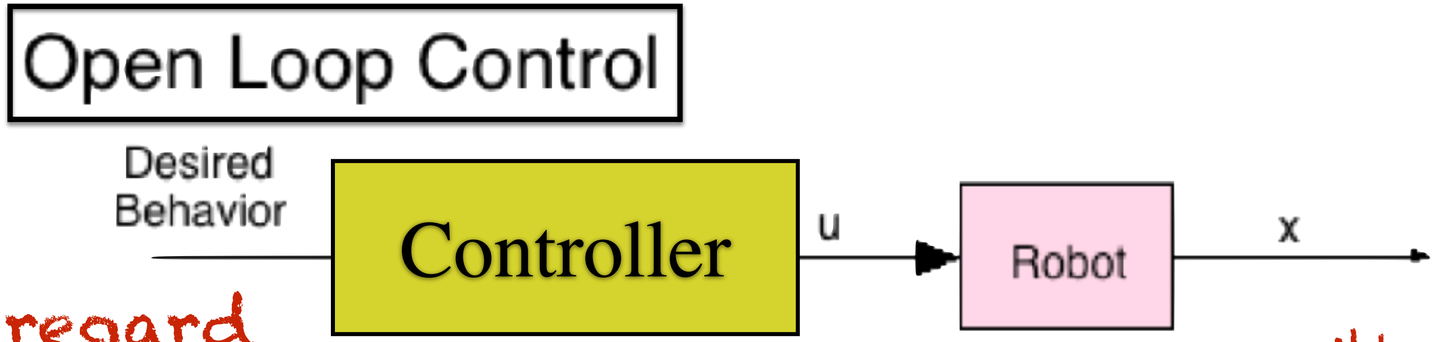


Control responds to feedback from environment

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

Slide borrowed from Michigan Robotics autorob.org

When to use open loop vs closed loop?

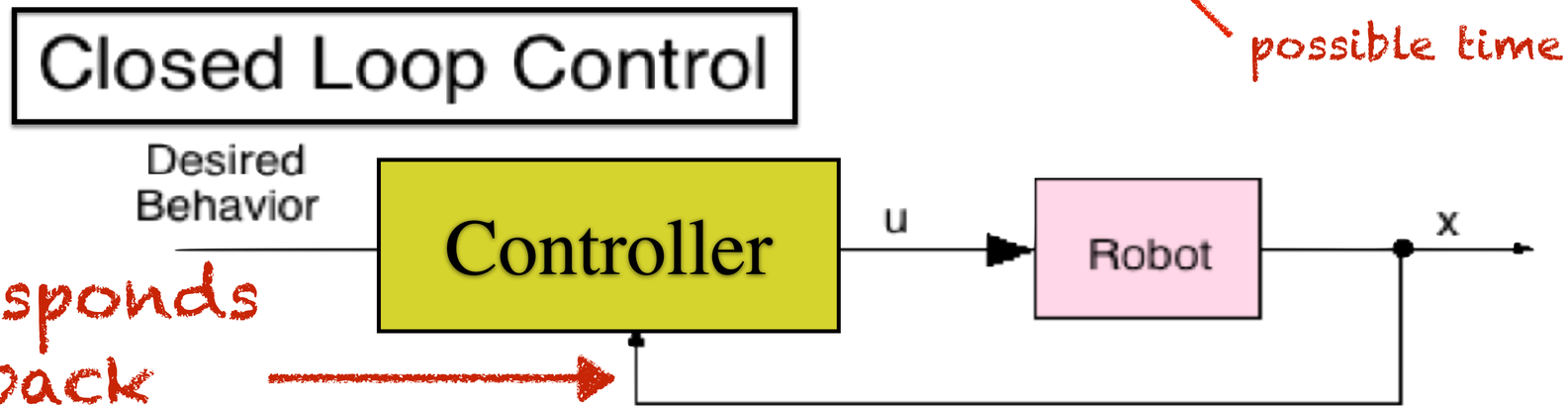


Acts without regard to environment

possible controller parameters

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

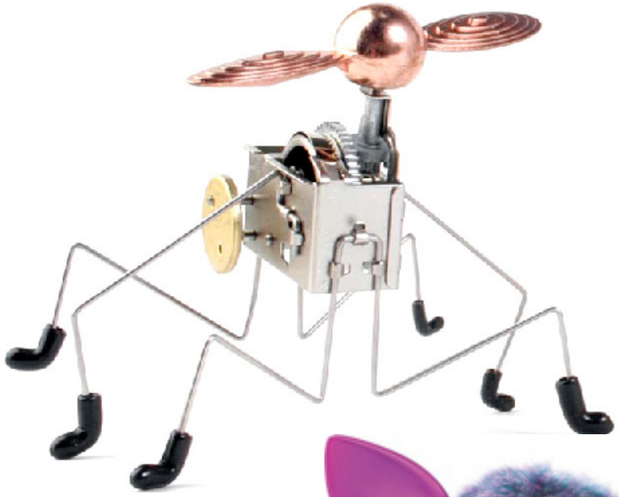
possible time dependence

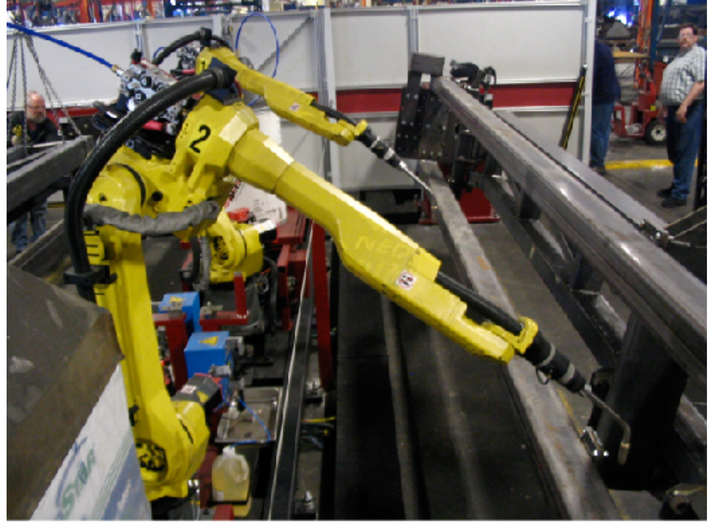


Control responds to feedback from environment

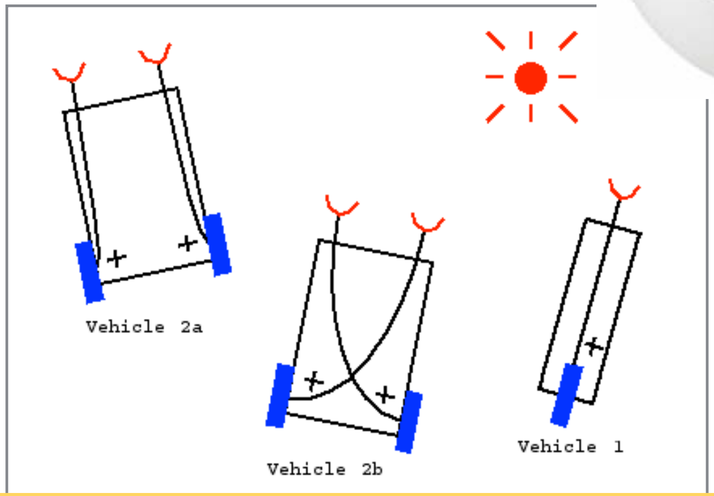
$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t)$$

Slide borrowed from Michigan Robotics autorob.org





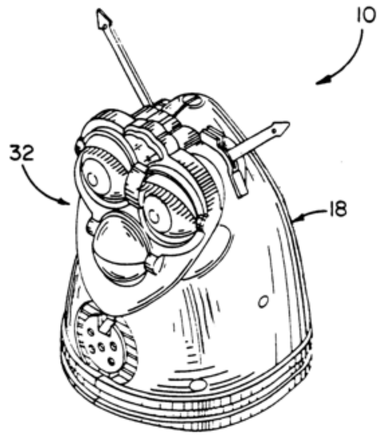
Which uses open loop?

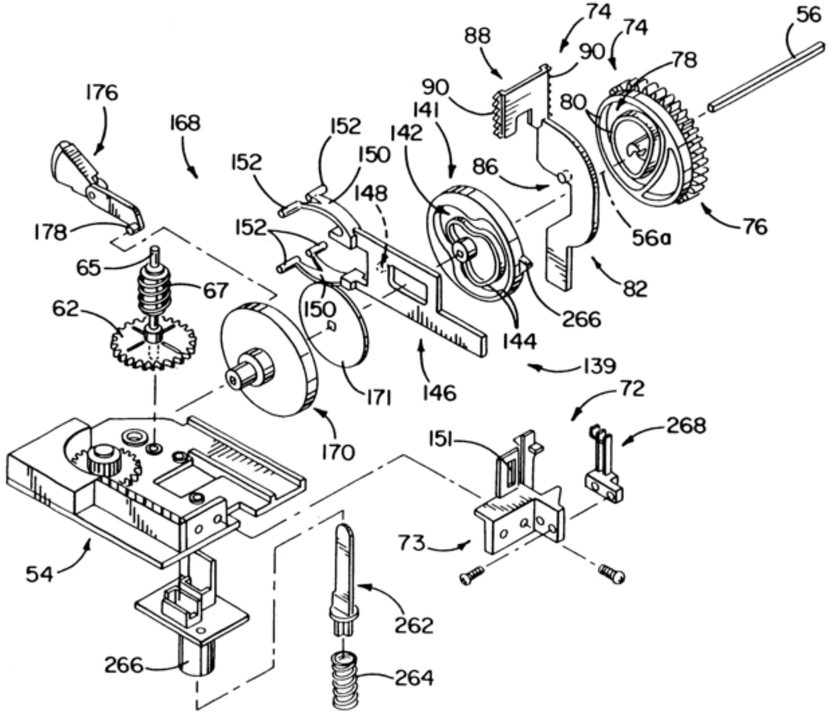


Slide borrowed from Michigan Robotics autorob.org

Slide borrowed from Michigan Robotics autorob.org

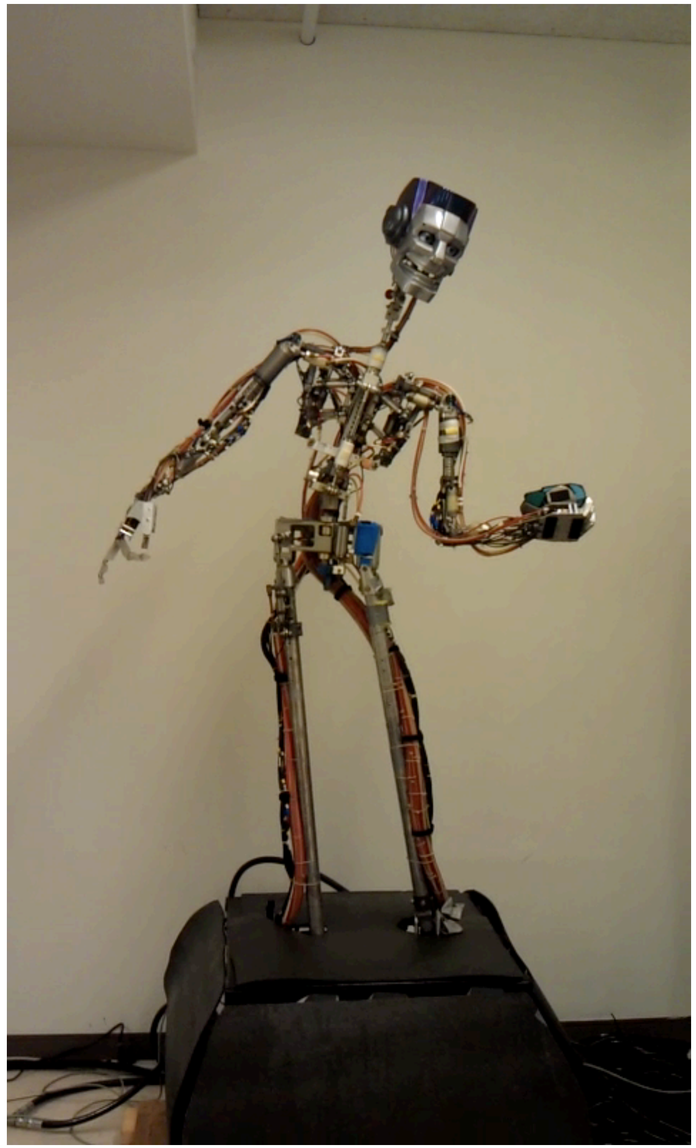
Open loop: Furby





Hasbro

Slide borrowed from Michigan Robotics autorob.org



Disney

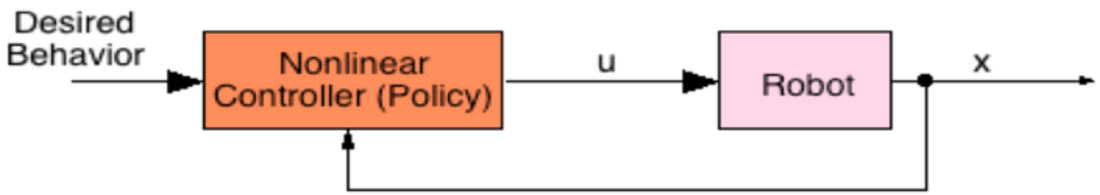
Slide borrowed from Michigan Robotics autorob.org

Types of Feedback Control

Slide borrowed from Michigan Robotics autorob.org

Types of Feedback Control

Feedback Control

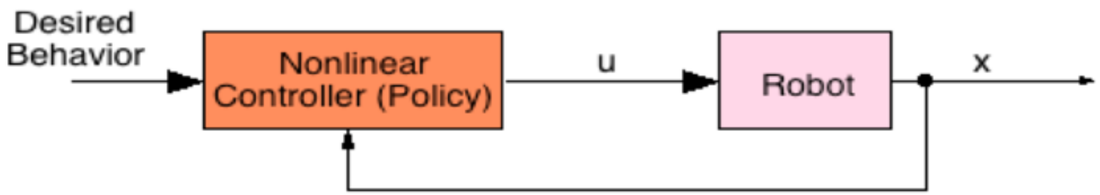


$$\mathbf{u} = \pi(\mathbf{x}, \alpha, t)$$

Slide borrowed from Michigan Robotics autorob.org

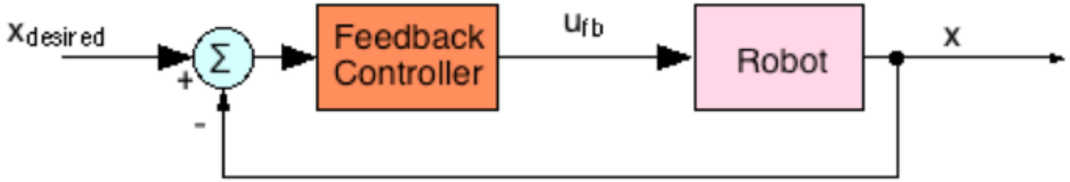
Types of Feedback Control

Feedback Control



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

Negative Feedback Control

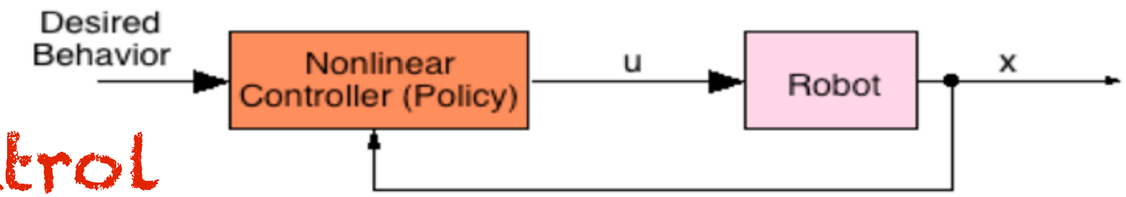


$$\mathbf{u} = \pi(\mathbf{x} - \mathbf{x}_{des}, \alpha, t)$$

Slide borrowed from Michigan Robotics autorob.org

Types of Feedback Control

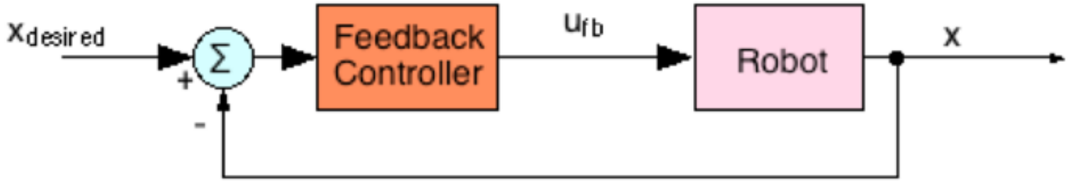
Feedback Control



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

PID Control

Negative Feedback Control

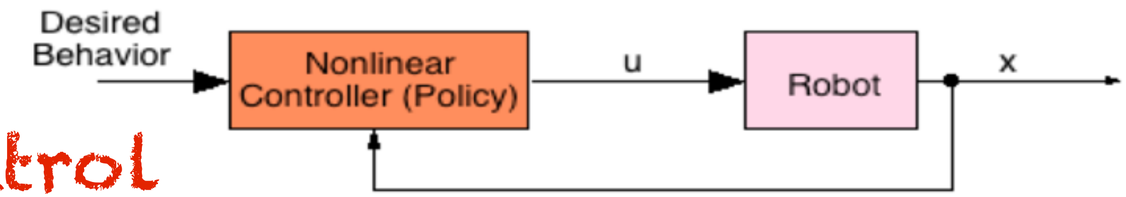


$$\mathbf{u} = \pi(\mathbf{x} - \mathbf{x}_{des}, \alpha, t)$$

Slide borrowed from Michigan Robotics autorob.org

Types of Feedback Control

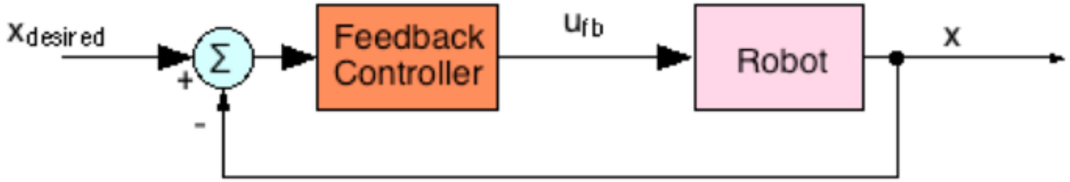
Feedback Control



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

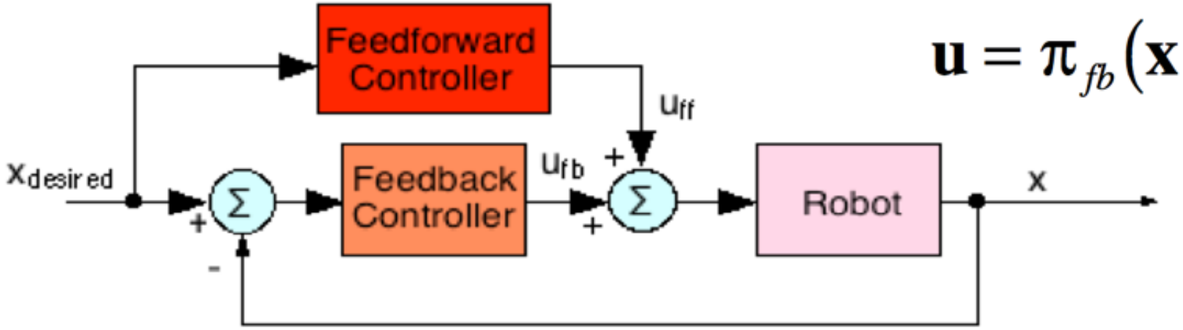
PID Control

Negative Feedback Control



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

Neg. Feedback & Feedforward Control



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

Slide borrowed from Michigan Robotics autorob.org

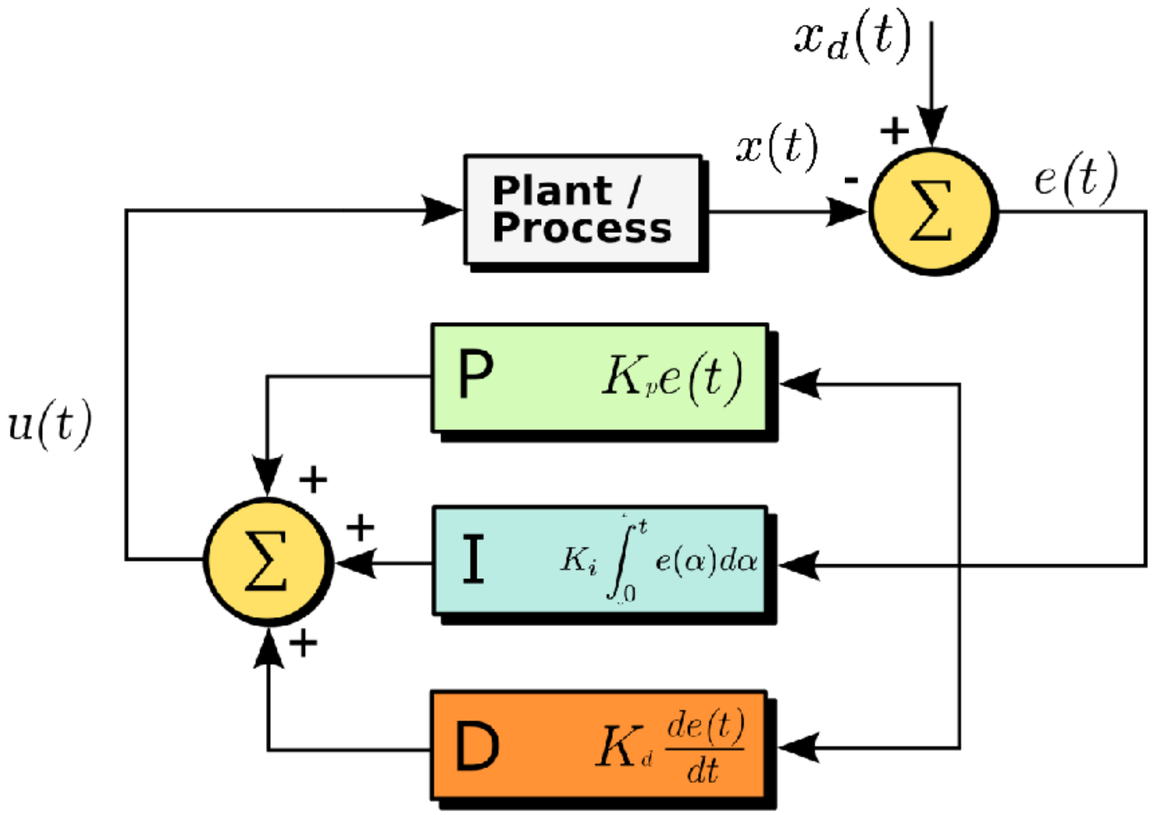
Remember PID?

PID Control

Slide borrowed from Michigan Robotics autorob.org

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



Slide borrowed from Michigan Robotics autorob.org

PID Control

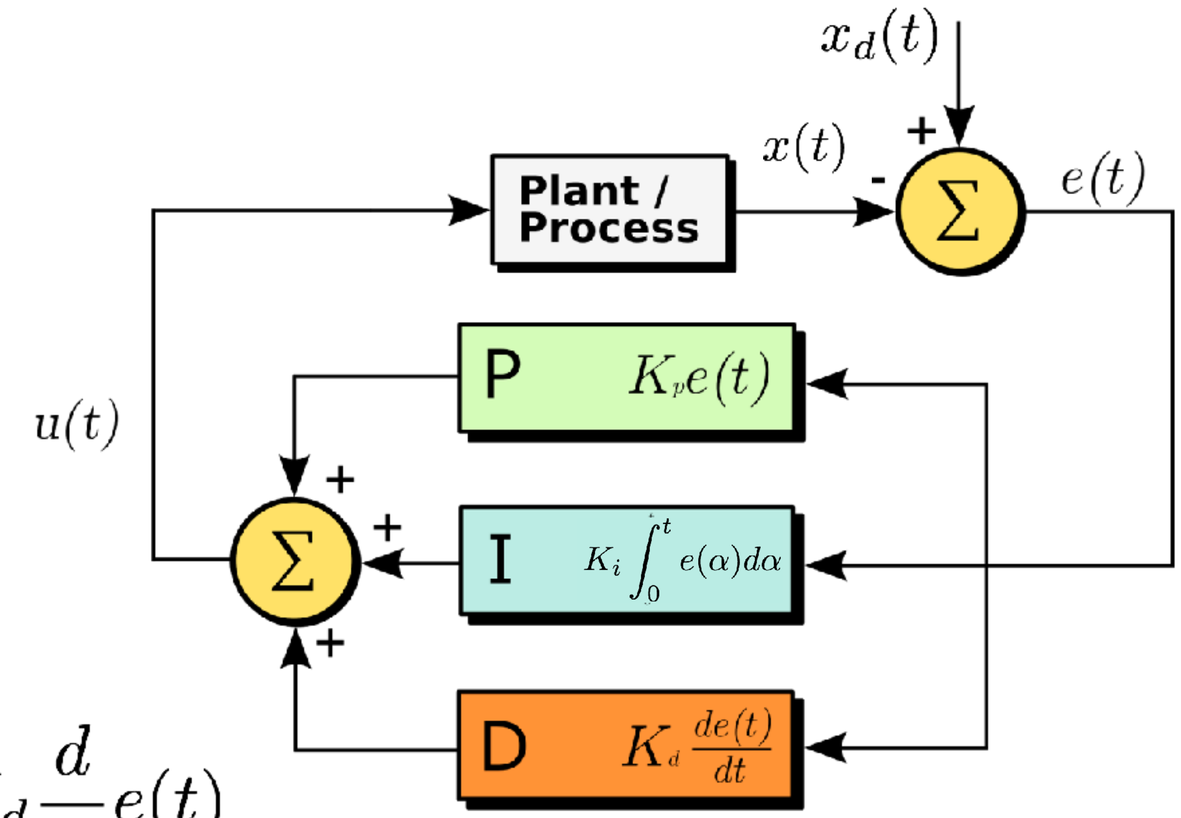
Error signal:

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

Control signal:

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

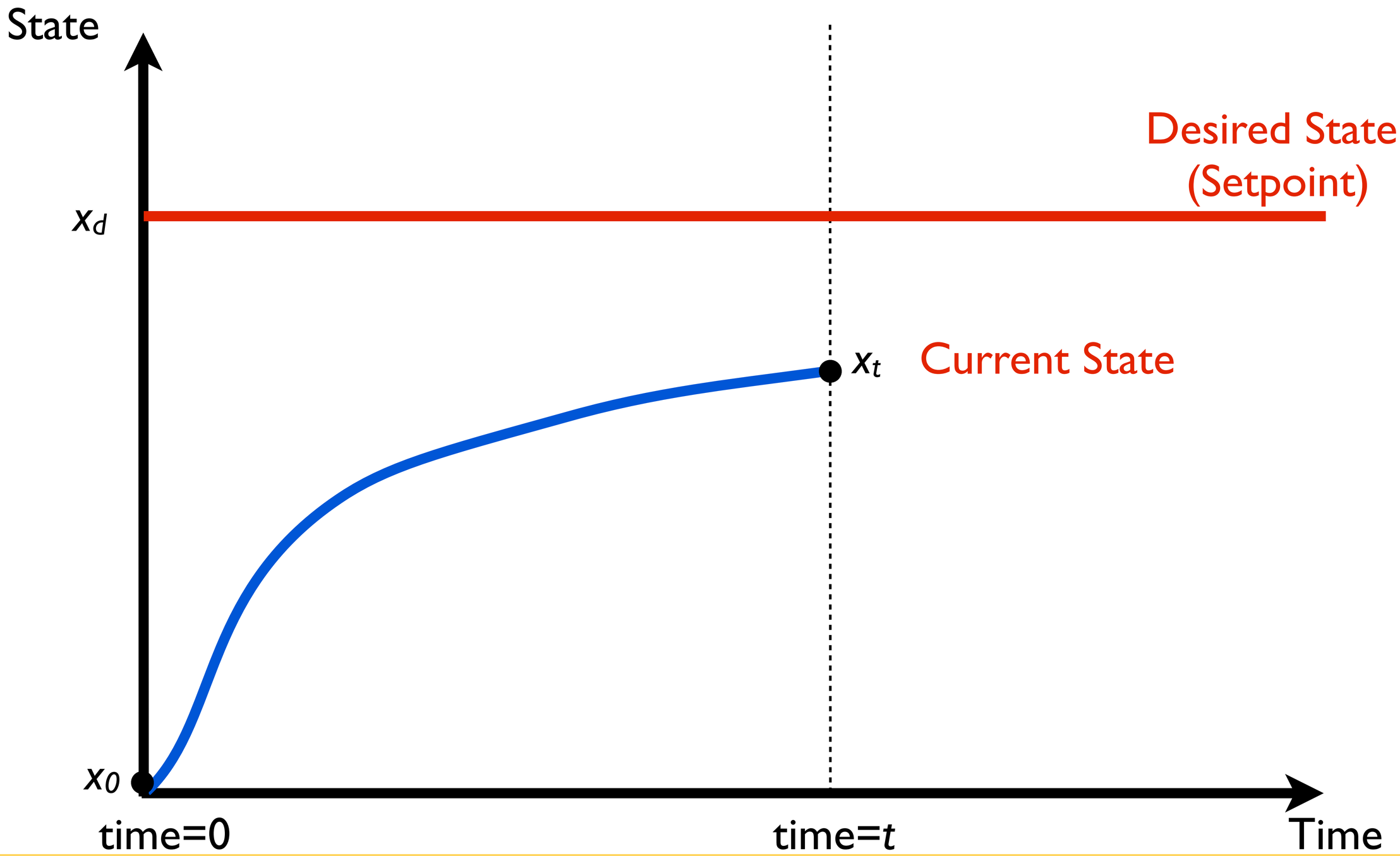
P	I	D
$K_p e(t)$	$K_i \int_0^t e(\alpha) d\alpha$	$K_d \frac{de(t)}{dt}$
Current	Past	Future



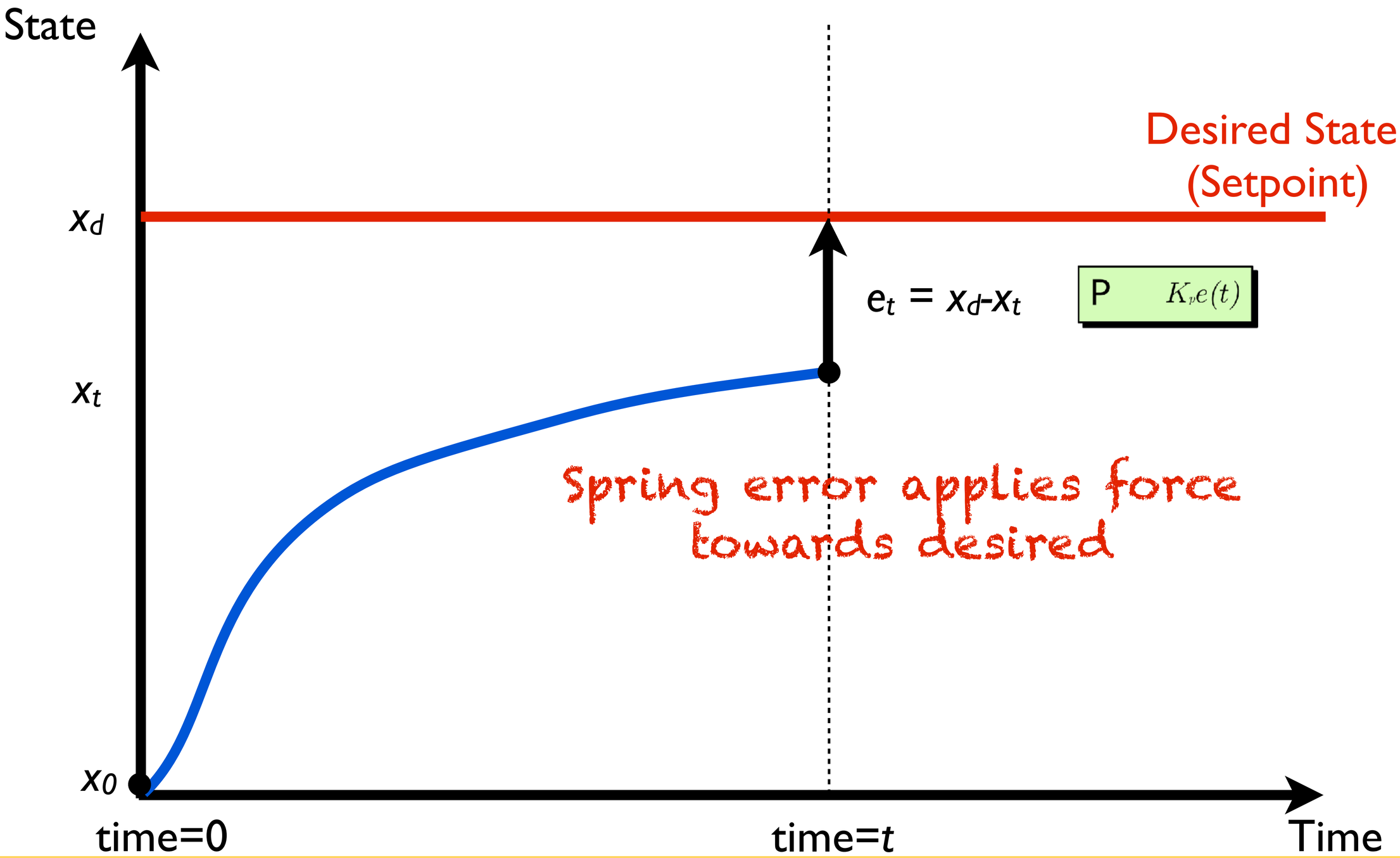
Slide borrowed from Michigan Robotics autorob.org

Consider PID wrt. state over time

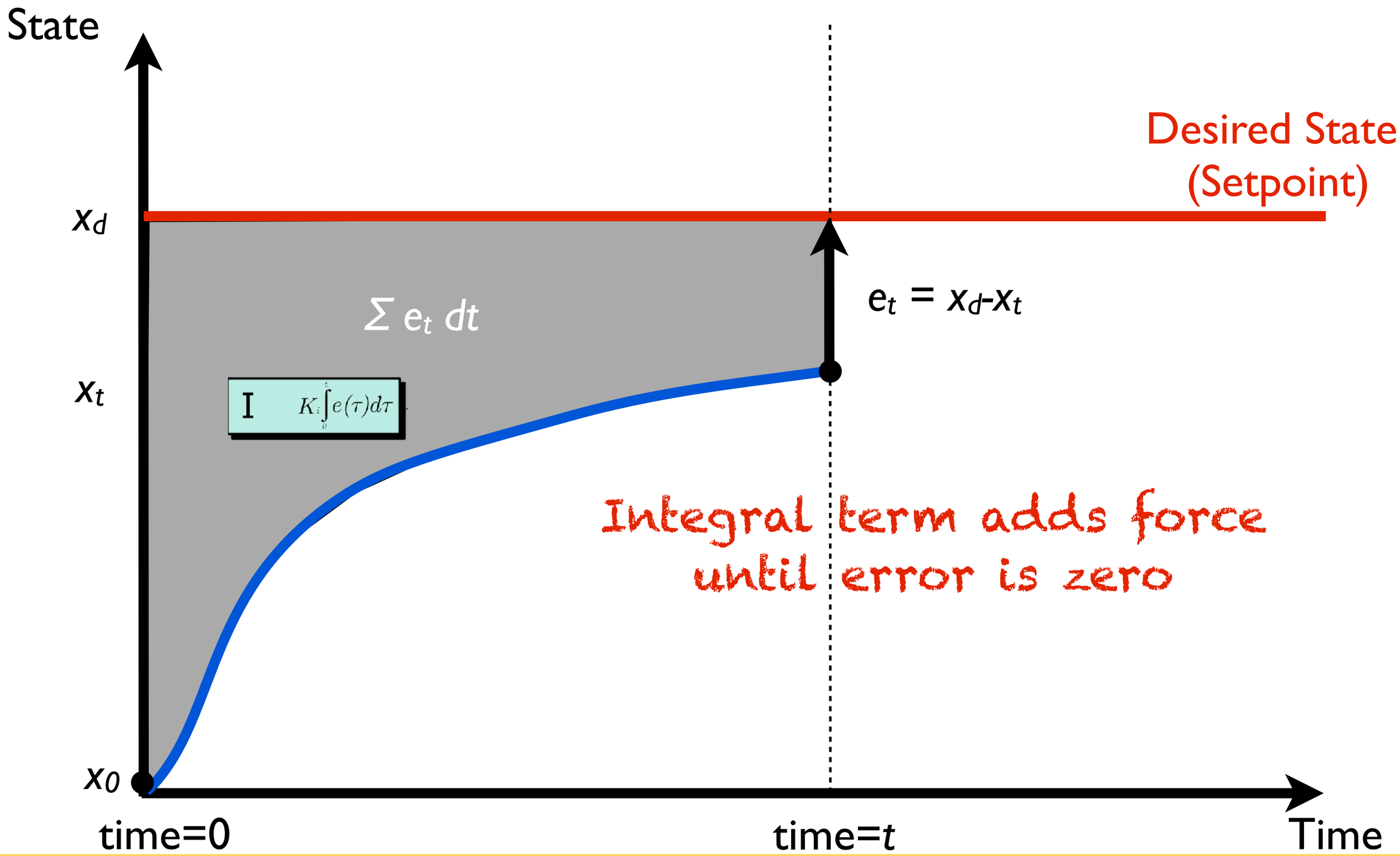
Slide borrowed from Michigan Robotics autorob.org



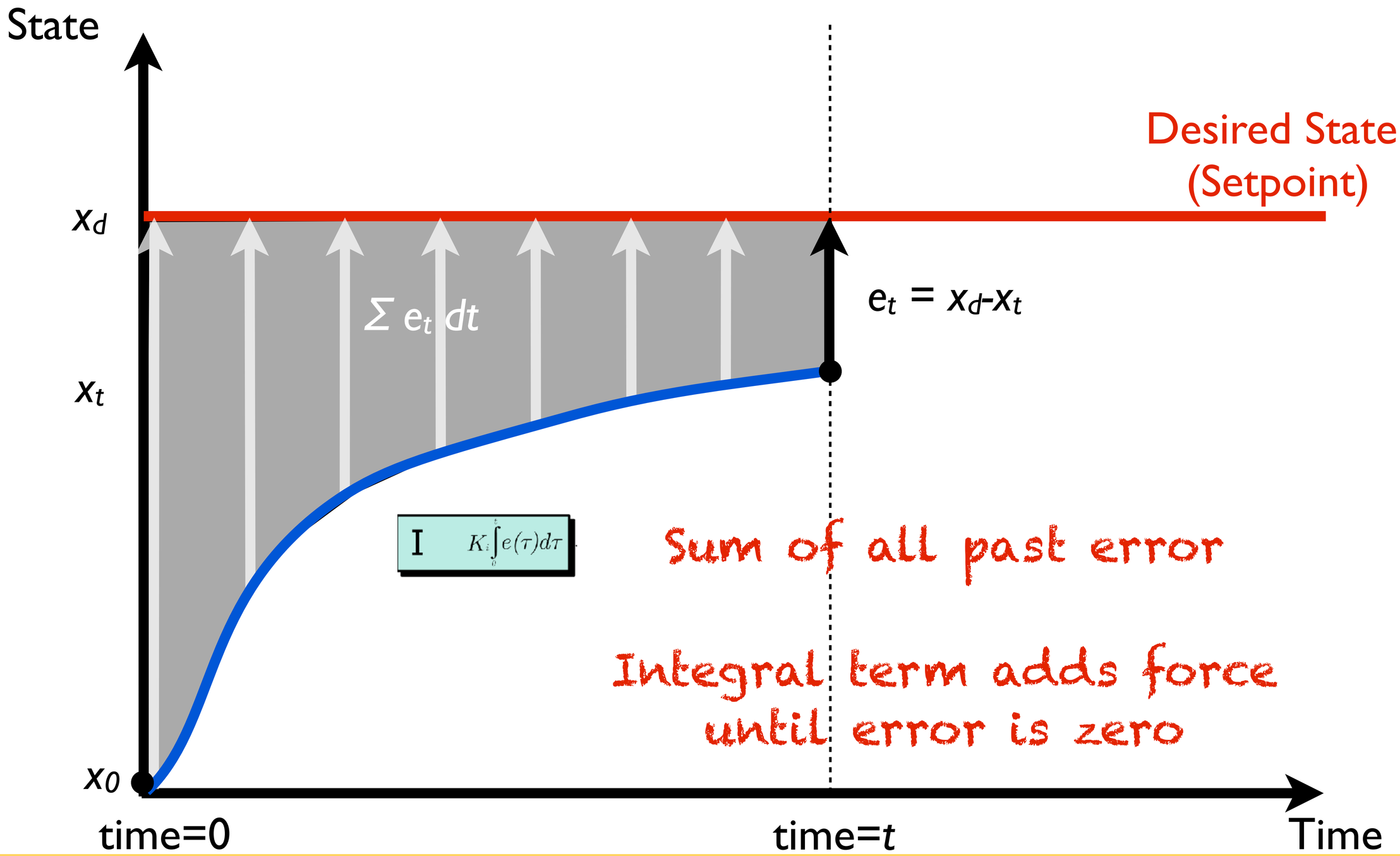
Slide borrowed from Michigan Robotics autorob.org



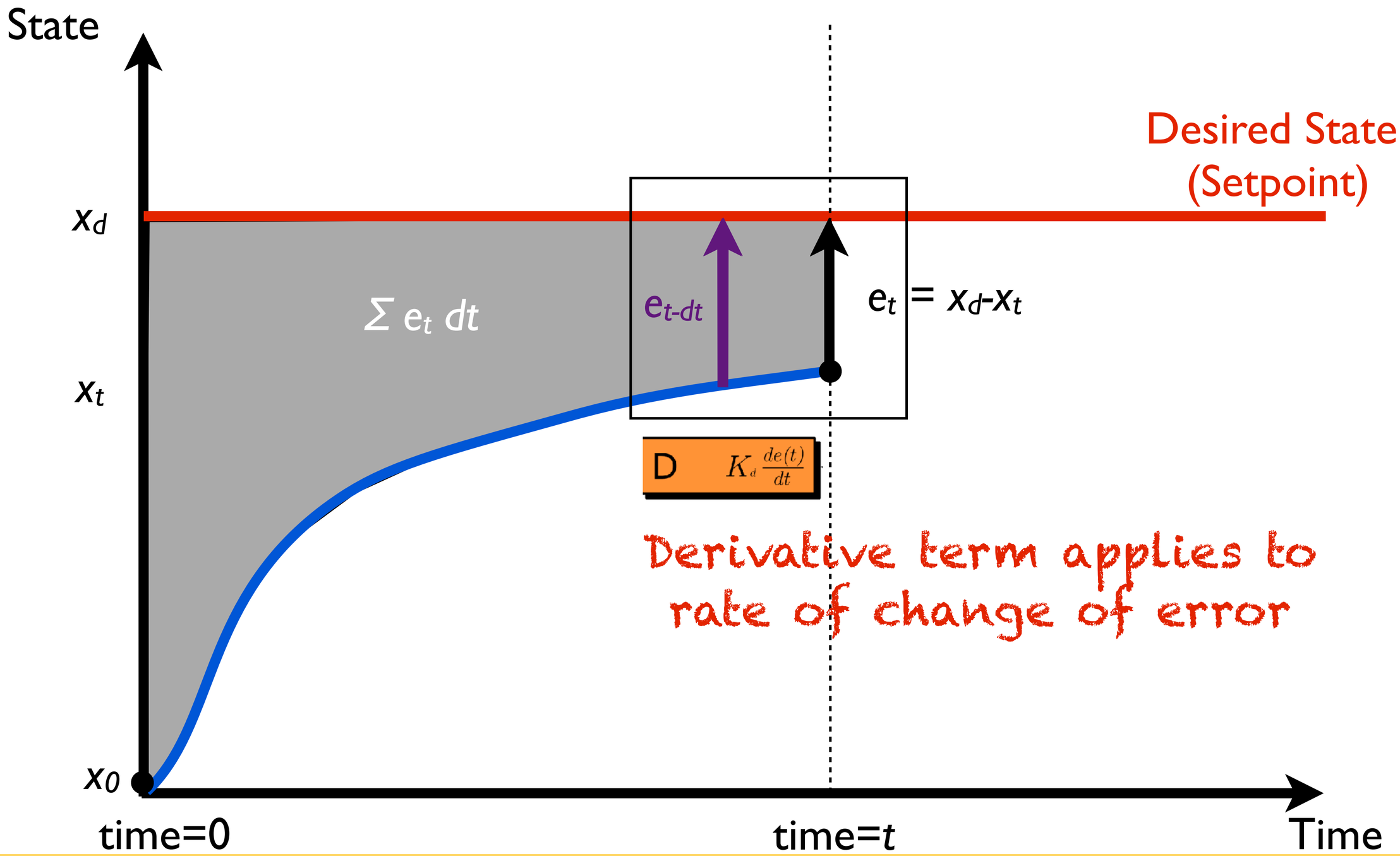
Slide borrowed from Michigan Robotics autorob.org



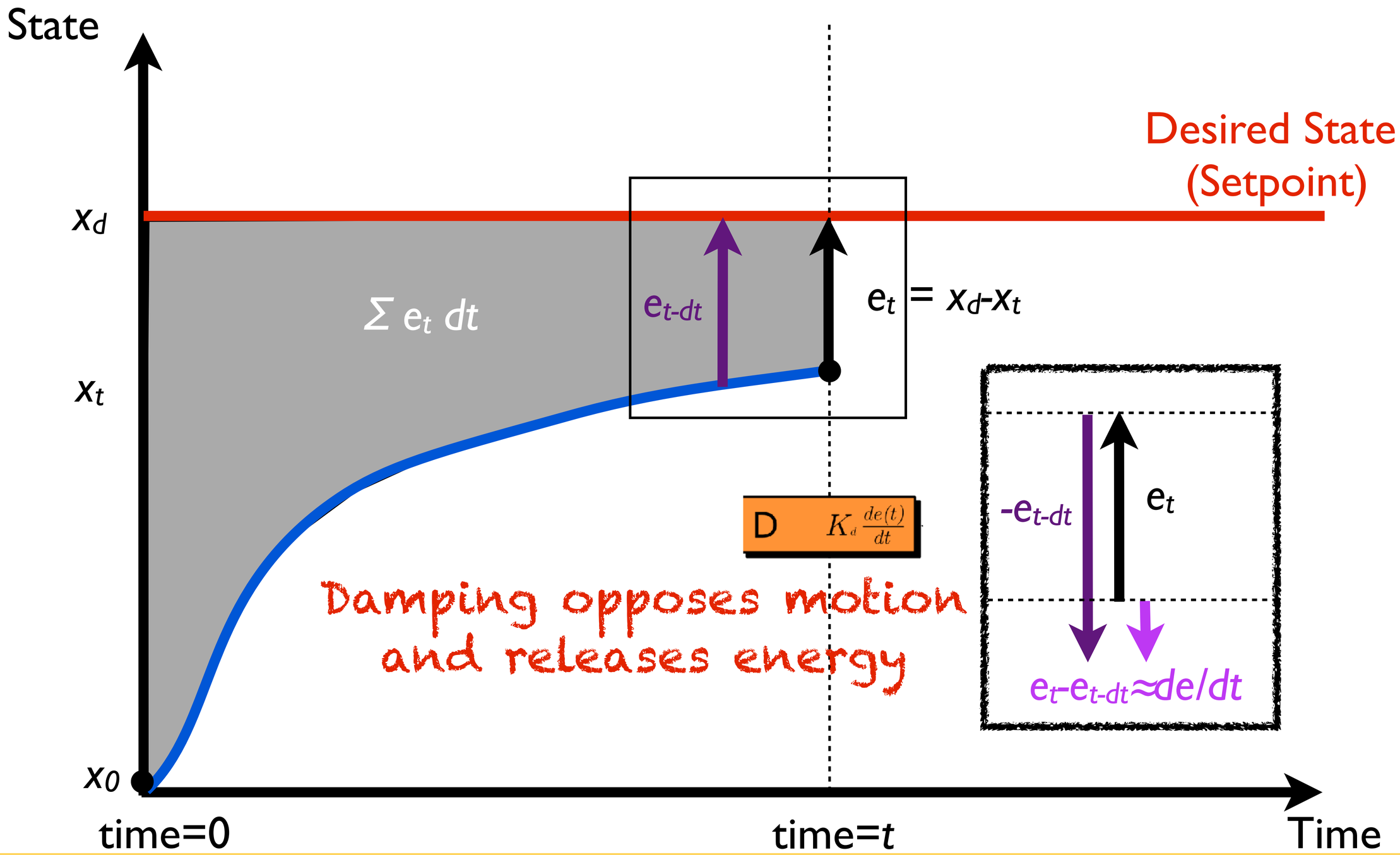
Slide borrowed from Michigan Robotics autorob.org



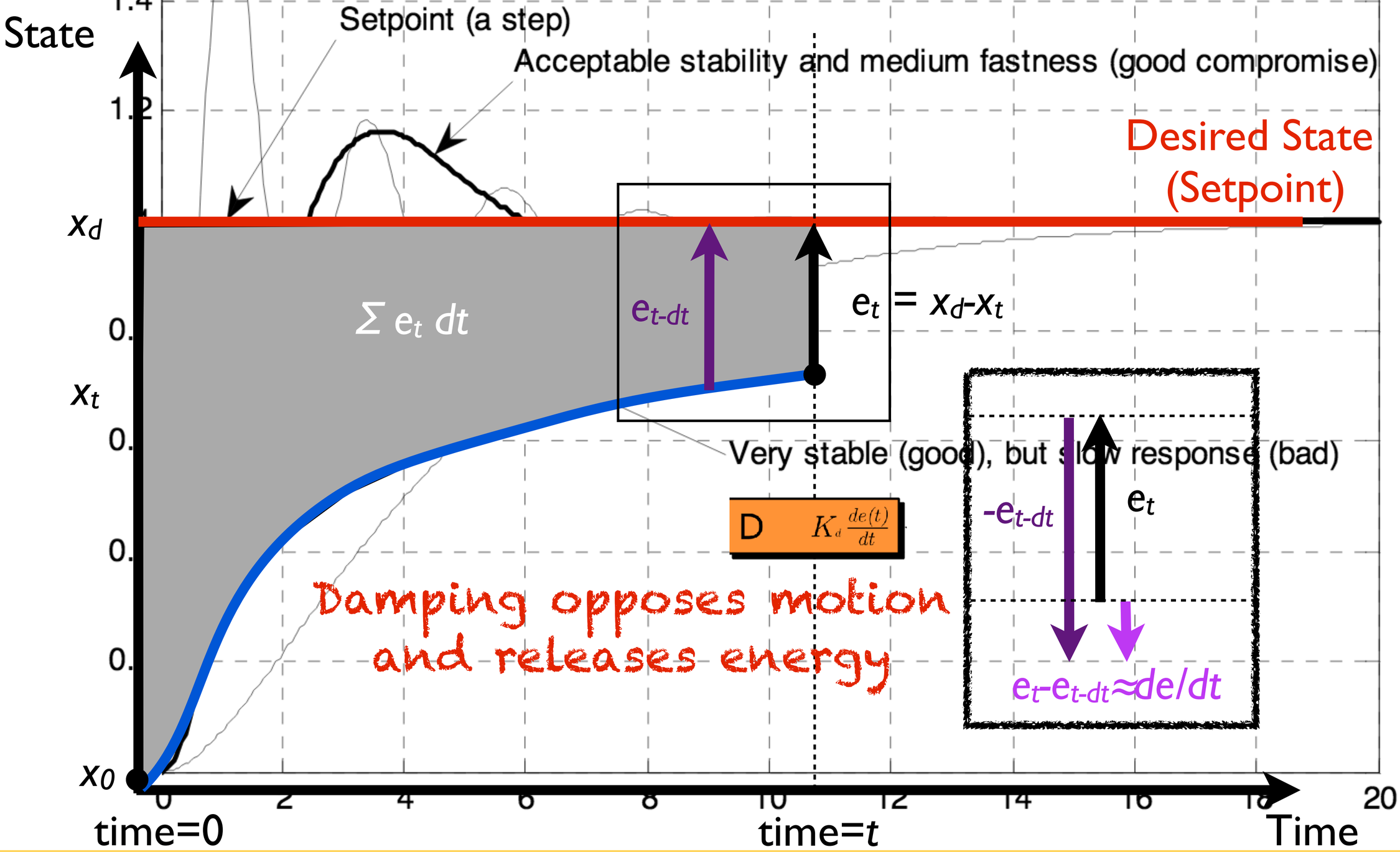
Slide borrowed from Michigan Robotics autorob.org



Slide borrowed from Michigan Robotics autorob.org

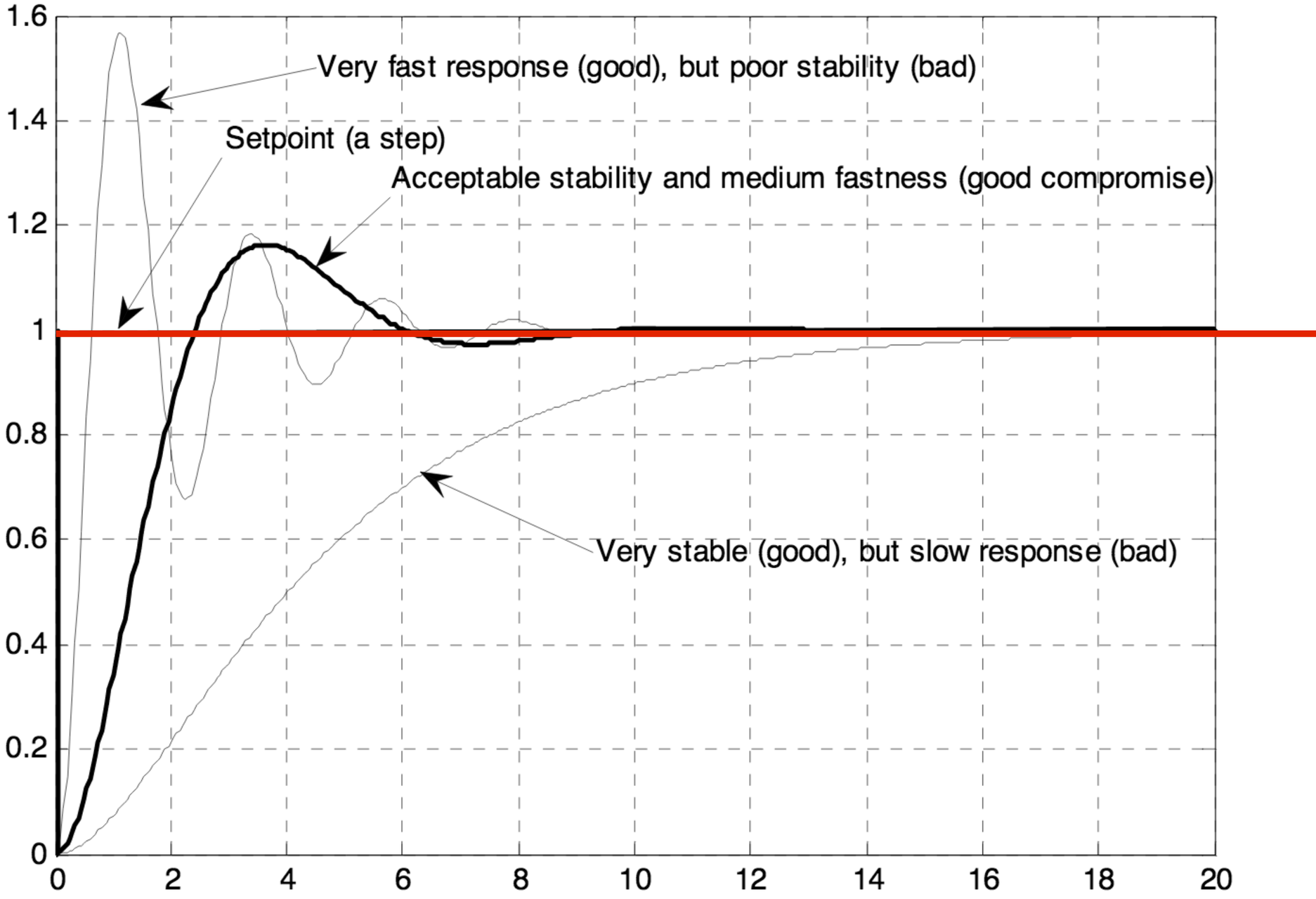


Slide borrowed from Michigan Robotics autorob.org



Slide borrowed from Michigan Robotics autorob.org

PID Converenge



Slide borrowed from Michigan Robotics autorob.org

PID as a spring and damper model

Slide borrowed from Michigan Robotics autorob.org

PID Control

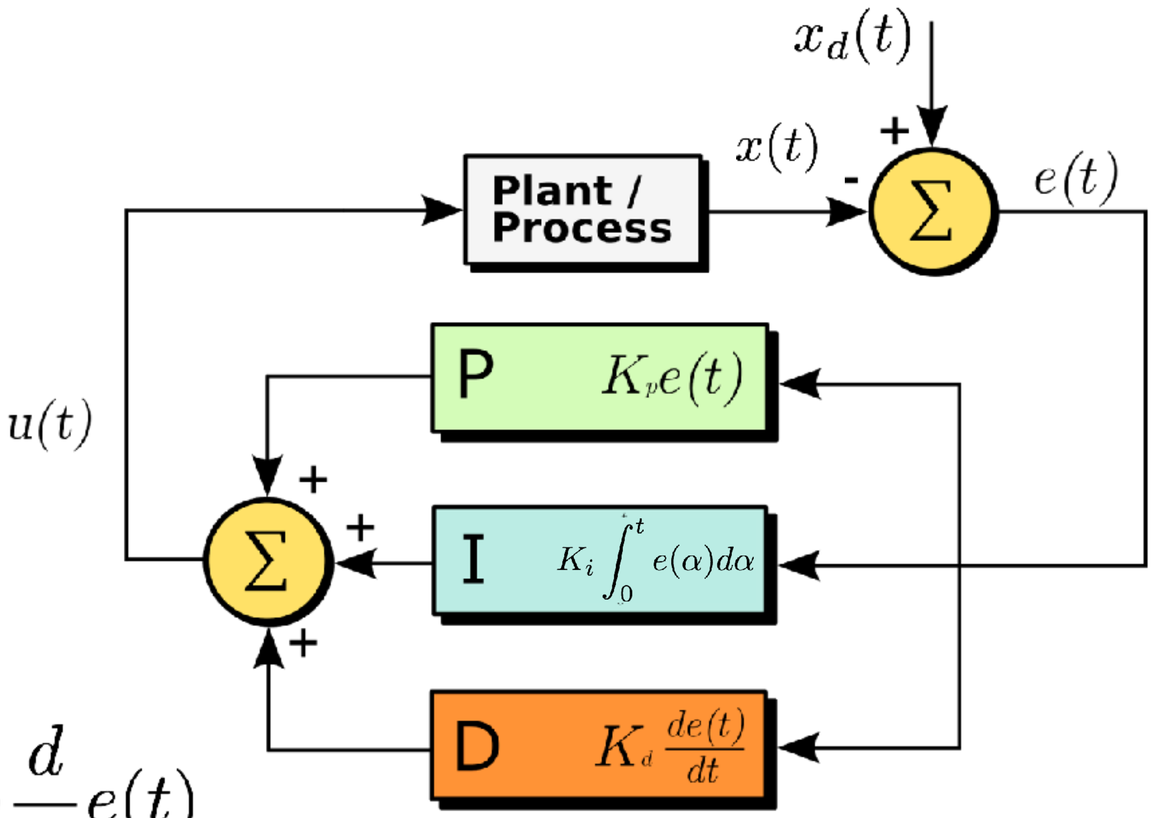
Error signal:

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

Control signal:

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

P	I	D
$K_p e(t)$	$K_i \int_0^t e(\alpha) d\alpha$	$K_d \frac{de(t)}{dt}$
Current	Past	Future



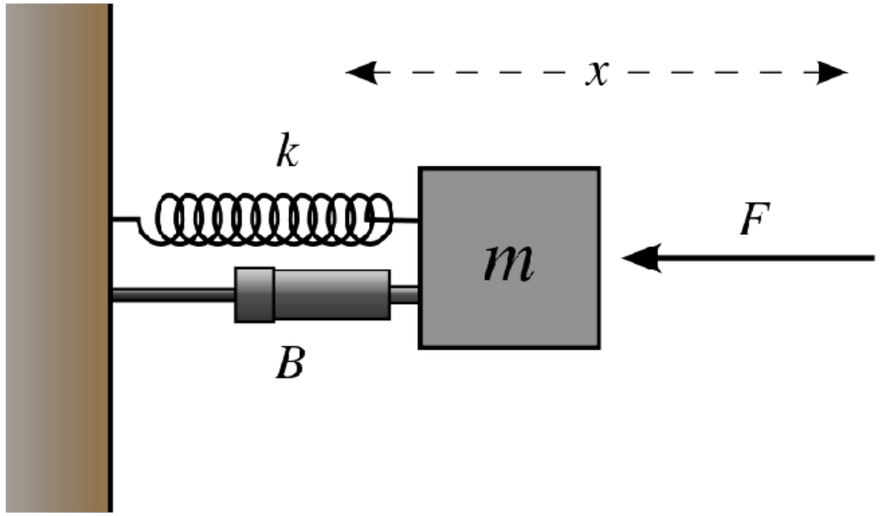
Slide borrowed from Michigan Robotics autorob.org

Hooke's Law

P $K_p e(t)$

- Describes motion of mass spring damper system as

$$F = -kx$$



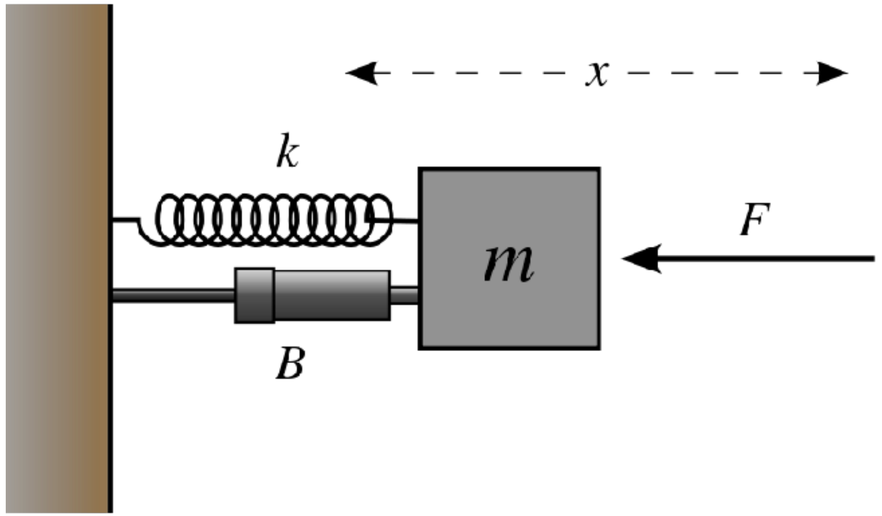
Robert Hooke
(1635-1703)

Slide borrowed from Michigan Robotics autorob.org

Hooke's Law

P $K_p e(t)$

- Describes motion of mass spring damper system as



$$F = -kx$$

force moving spring towards rest

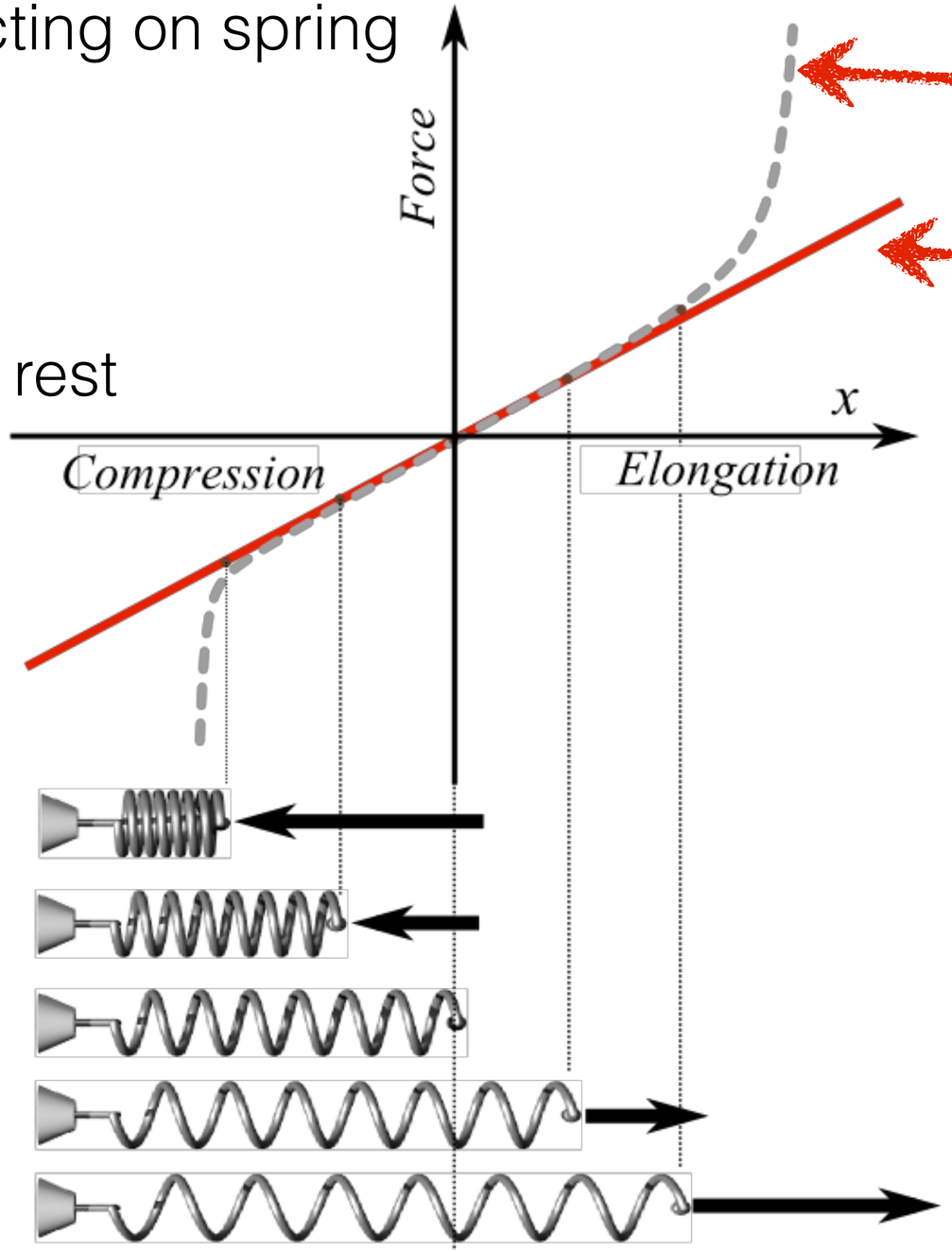
spring stiffness

distance from rest displacement

Slide borrowed from Michigan Robotics autorob.org

Vertical: Force acting on spring

Horizontal: Position from rest

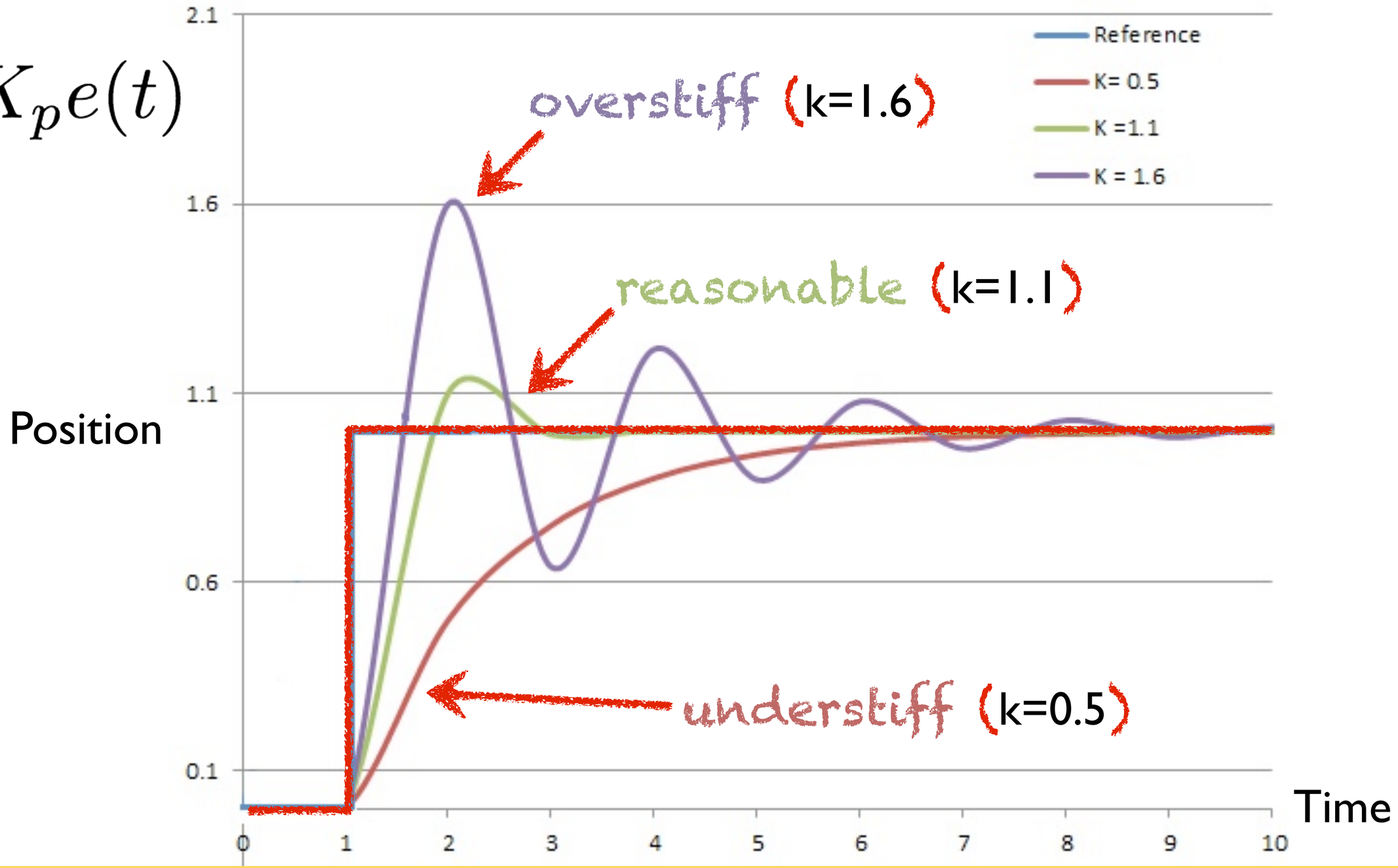


actual force relation

linear approximation (-kx)

Slide borrowed from Michigan Robotics autorob.org

$K_p e(t)$



Slide borrowed from Michigan Robotics autorob.org

PID Control

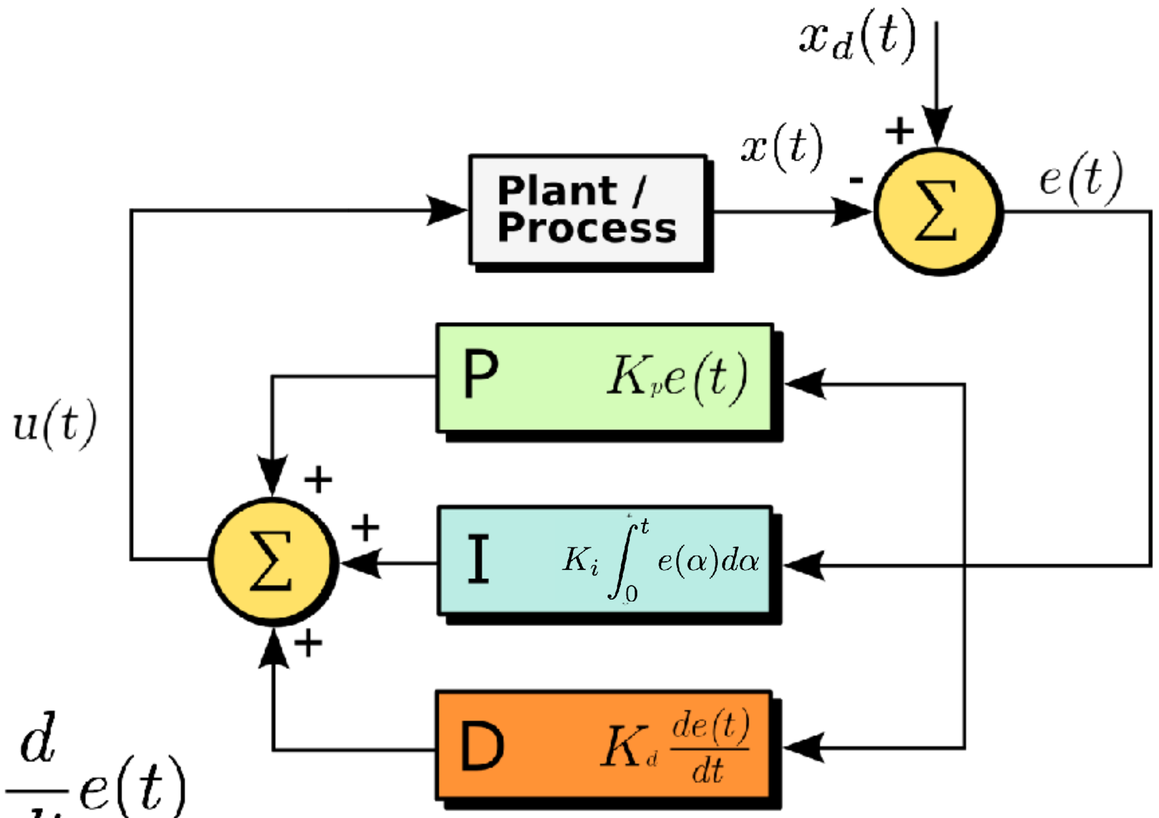
Error signal:

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

Control signal:

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

P	I	D
$K_p e(t)$	$K_i \int_0^t e(\alpha) d\alpha$	$K_d \frac{de(t)}{dt}$
Current	Past	Future



Slide borrowed from Michigan Robotics autorob.org

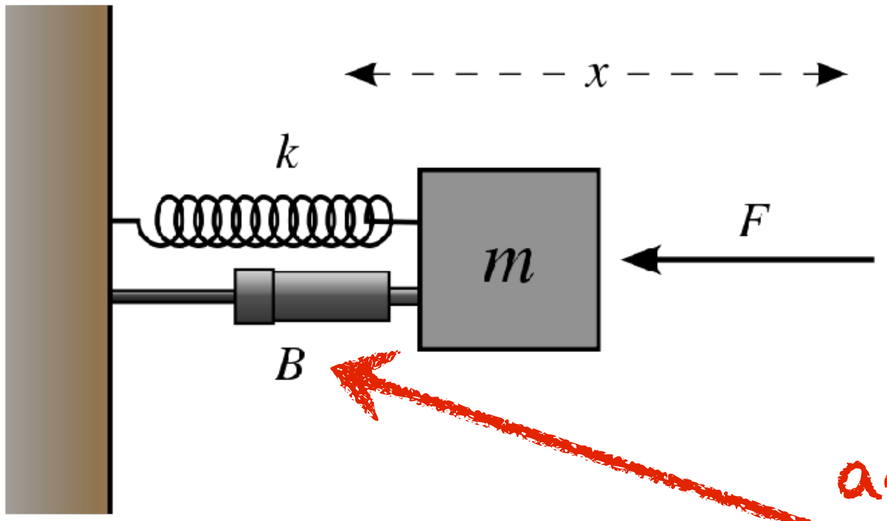
Spring and Damper

P $K_p e(t)$

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

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

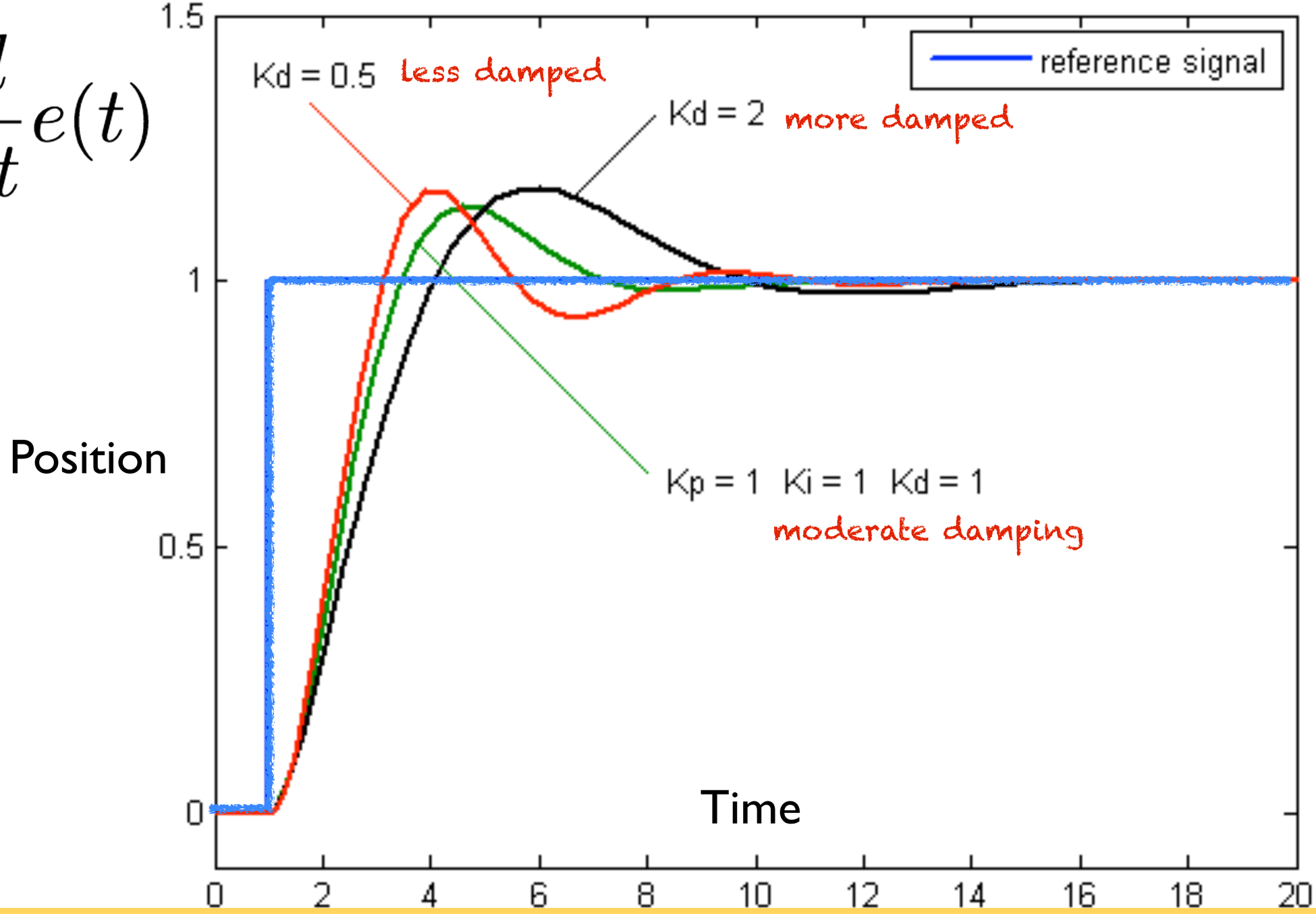
assuming constant set point, velocity is derivative of error



add damper to release energy

Slide borrowed from Michigan Robotics autorob.org

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



Slide borrowed from Michigan Robotics autorob.org

PID Control

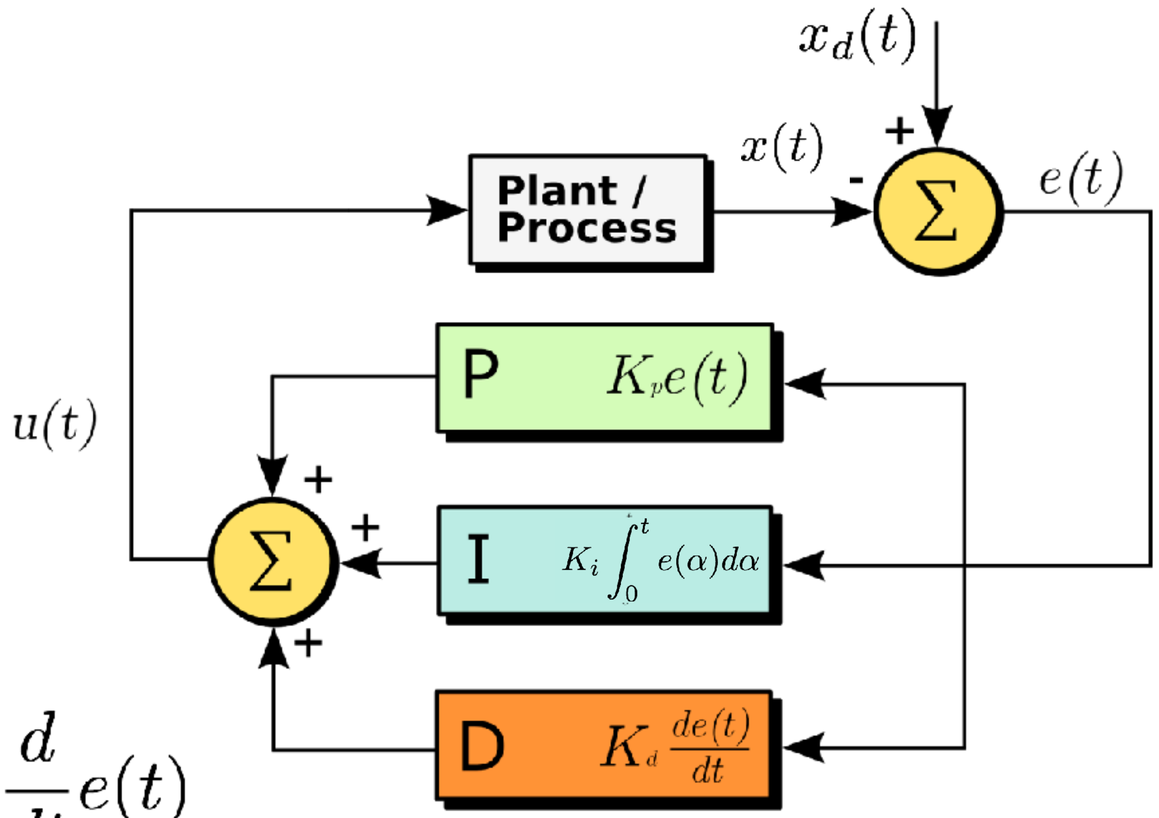
Error signal:

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

Control signal:

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

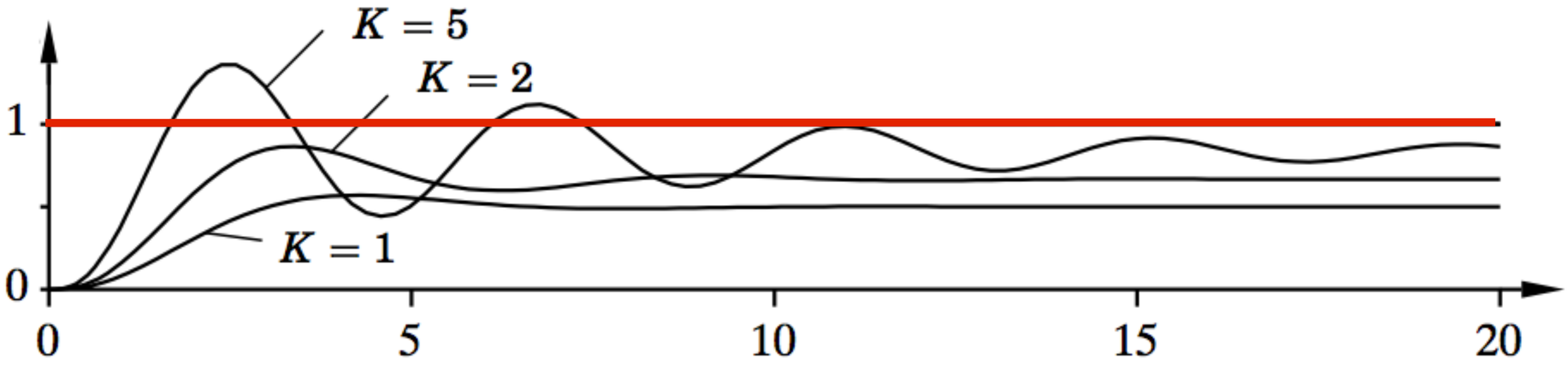
P	I	D
$K_p e(t)$	$K_i \int_0^t e(\alpha) d\alpha$	$K_d \frac{de(t)}{dt}$
Current	Past	Future

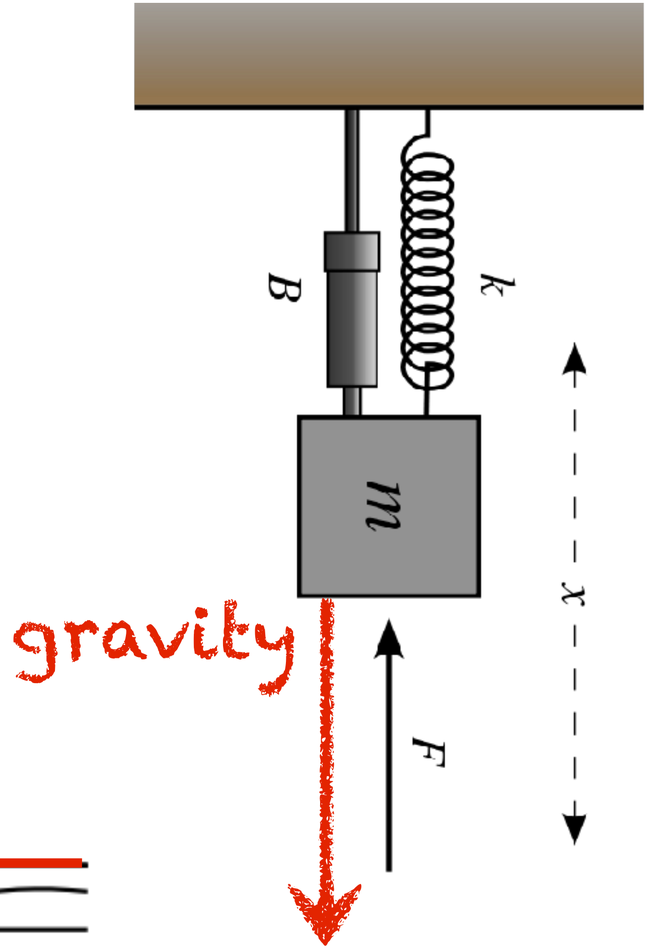


Slide borrowed from Michigan Robotics autorob.org

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

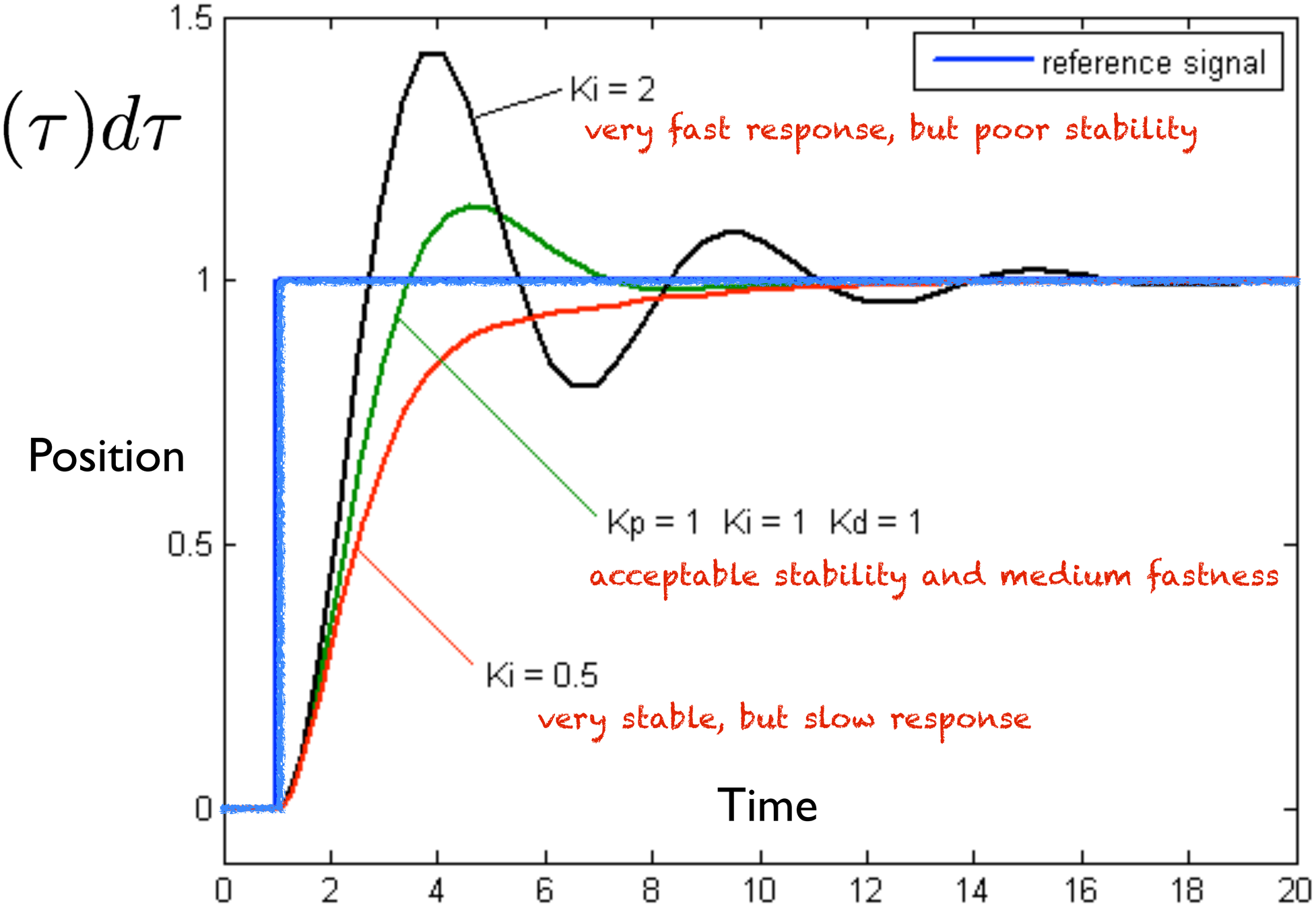




Slide borrowed from Michigan Robotics autorob.org

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

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



Slide borrowed from Michigan Robotics autorob.org

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.

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

P	$K_p e(t)$
I	$K_i \int_0^t e(\alpha) d\alpha$
D	$K_d \frac{de(t)}{dt}$

Slide borrowed from Michigan Robotics autorob.org

Some tips 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

Slide borrowed from Michigan Robotics autorob.org

Next Lecture
Mobile Robotics - I

Atlas Gets a Grip | Boston Dynamics - https://www.youtube.com/watch?v=-e1_QhJ1EhQ