CprE 488 – Embedded Systems Design

Lecture 7 – Embedded Control Systems

Joseph Zambreno and Phillip Jones
Electrical and Computer Engineering
Iowa State University

www.ece.iastate.edu/~zambreno   www.ece.iastate.edu/~phjones
rcl.ece.iastate.edu

If everything seems under control, you're just not going fast enough. – Mario Andretti

Motivation for Controls

- We often need a way to direct a system to a given goal (i.e., setpoint)
  – A car's cruise control: Reach and maintain a given speed
  – Quadcopter control: Maintain a stable hover
  – Building heating system: Reach and maintain a given temperature

- To this end, we will focus on feedback control techniques
  – PID (main focus)
  – State Space (touch upon)

- There are additional control techniques, such as feedforward, but this is typically used in combination with feedback, and beyond the scope of this course.

Motivation for Controls (cont.)

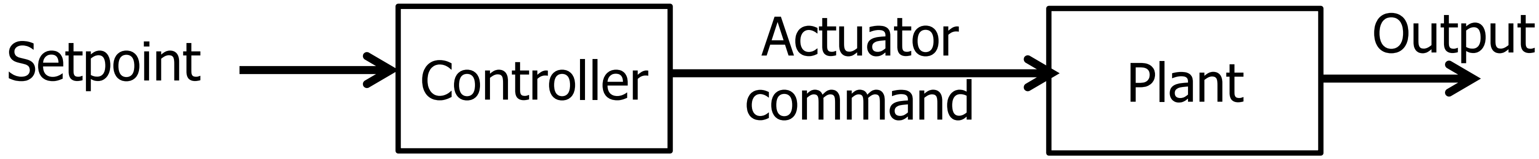
- Simple inverted pendulum on a chart:
  - https://www.youtube.com/watch?v=9KU39-V16Bk
- Triple inverted pendulum on chart (free fall):
  - https://www.youtube.com/watch?v=cyN-CRNrb3E
- Triple inverted pendulum on chart (controlled fall):
  - https://www.youtube.com/watch?v=SWupnDzynNU
- Human vs robot dog (Boston Dynamics):
  - https://www.youtube.com/watch?v=W1LWMk7JB80
- Handle (Boston Dynamics):
  - https://www.youtube.com/watch?v=-7xvqQeoA8c
- Atlas: Back-flip (Boston Dynamics):
  - https://www.youtube.com/watch?v=fRj34o4hN4I
- Parkour Atlas (Boston Dynamics):
  - https://www.youtube.com/watch?v=tF4DML7FIWk
- Construction (Boston Dynamics):
  - https://www.youtube.com/watch?v=-e1_QhJ1EhQ

Terminology

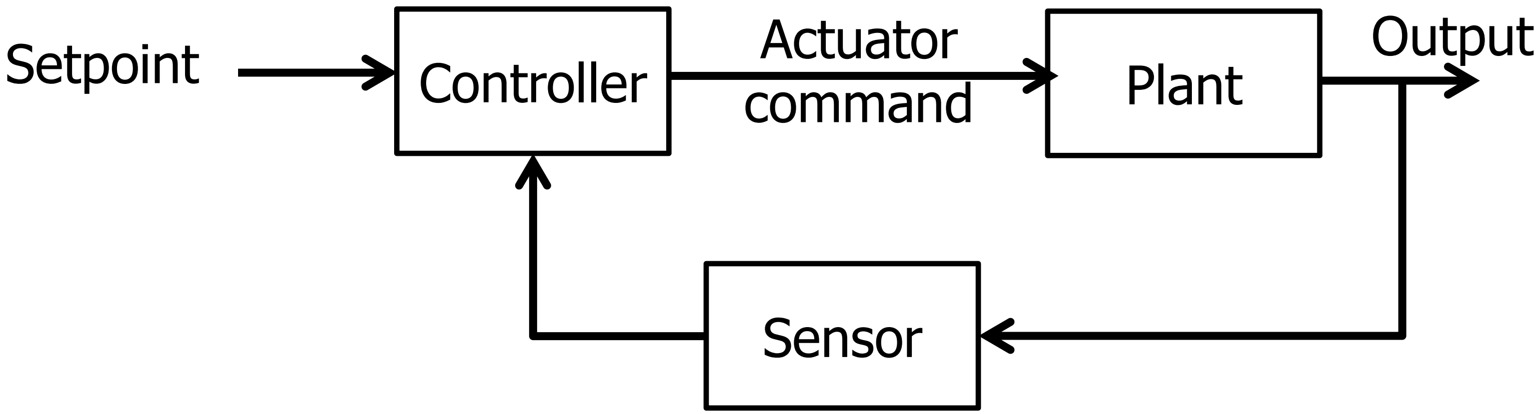
- **Plant/Process**: system being controlled
  - Car, Plane, Building, Quadcopter
- **Setpoint**: goal-value of the quantity being controlled
  - Speed, Temperature, Height
- **Sensor**: mechanism for measuring quantities of the system
  - Thermometer, Barometer, Tachometer, Encoder, Accelerometer
- **Actuator**: mechanism to enact change on the plant
  - Servo, Valve, Muscle, Motor
- **Controller**: mechanism to process sensors, and command actuators
  - Microprocessor, FPGA logic, Analog circuit
- **Control Law**: Rules that map sensor signals to actuator commands
  - On-off, P, PD, PI, PID, State-space, …

Terminology

- **Open-Loop**: Control system uses a controller to obtain the desired response with no feedback.



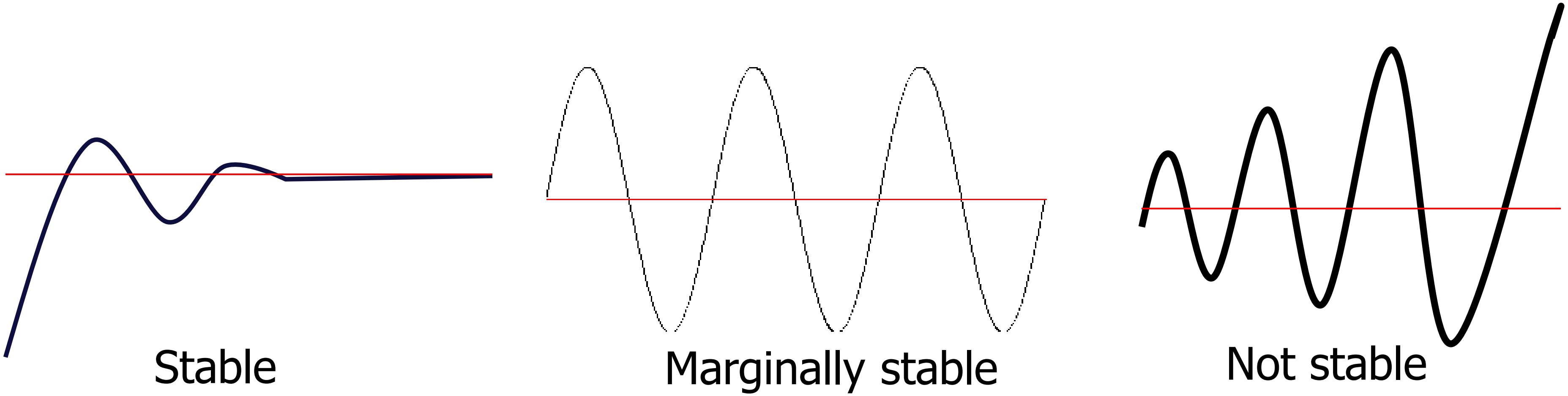
- **Closed-Loop:** Control systems use a controller with feedback to compare the actual output to the desired plant response.



- Control of Mobile Robots (Georgia Tech): **Great 6-week intro!!!**
  - https://www.youtube.com/playlist?list=PLp8ijpvp8iCvFDYdcXqqYU5IbI_aOqwjr

Typical Controller Metrics

- **Stability**: (e.g. bounded oscillation of system output)



- For a stable controlled system
  - **Disturbance Rejection**: How well does the system hold setpoint in the presence of a disturbance (e.g., shoving a quadcopter)
  - **Command tracking**: How well does the system respond to changes in the controller setpoint
    - Rise time
    - Settling time

Examples

- Watt's fly ball governor (1788)

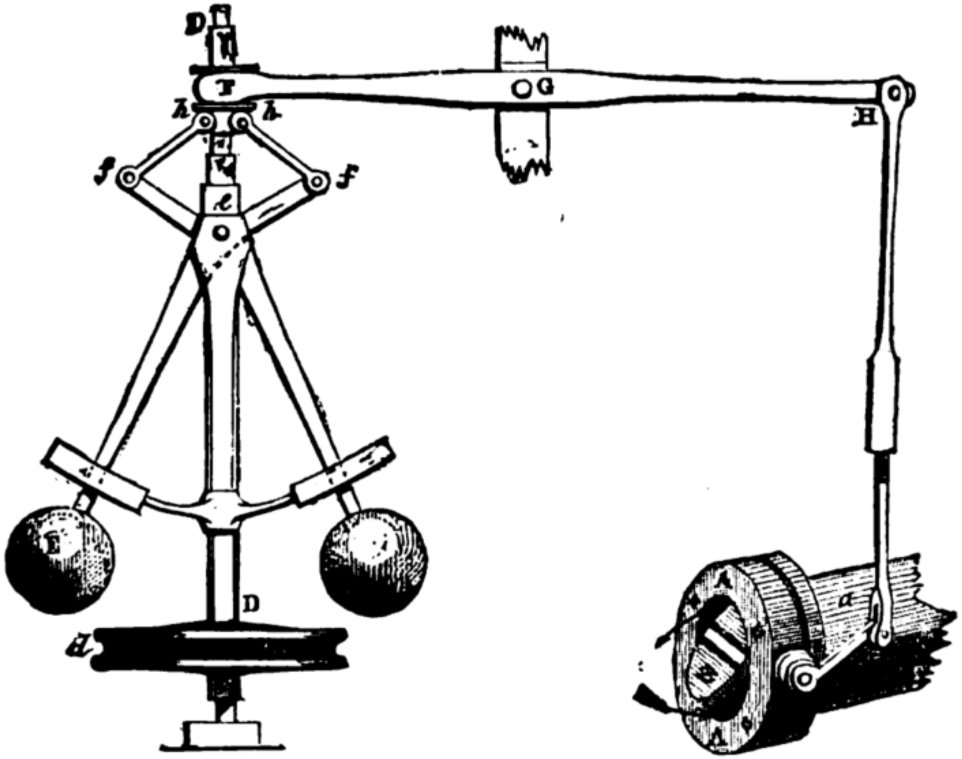
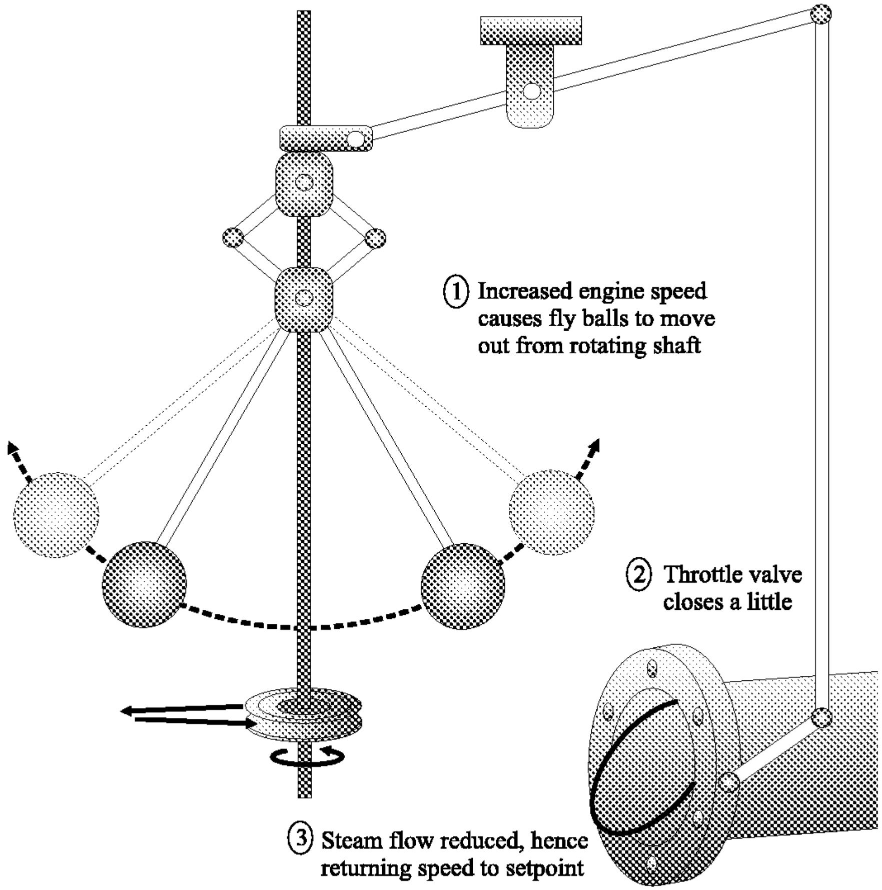


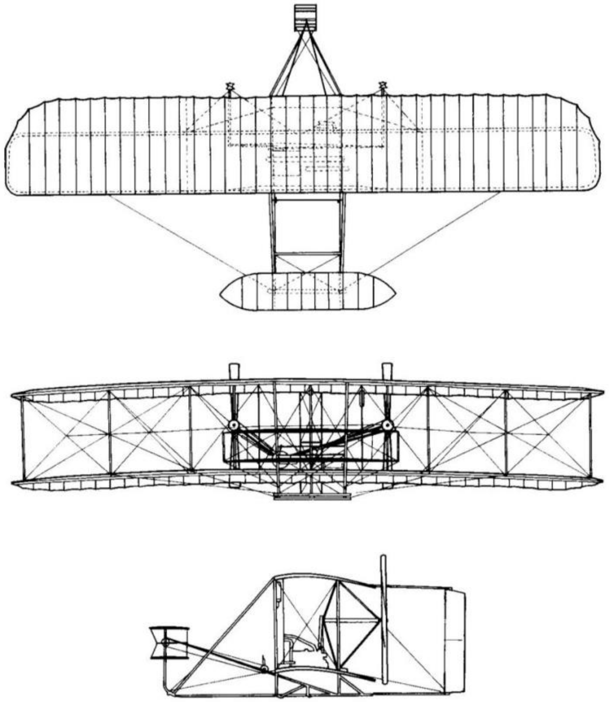
FIG. 4.—*Governor and Throttle-Valve.*



- 1868: James Clerk Maxwell publishes the first theoretical study of steam engine governors. By that time, there were more than 75,000 governors installed in England.

Examples (cont.)

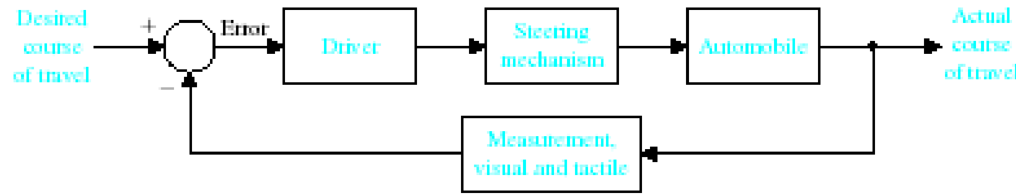
- Orville and Wilbur Wright made the first successful experiment with manned flight (1905)



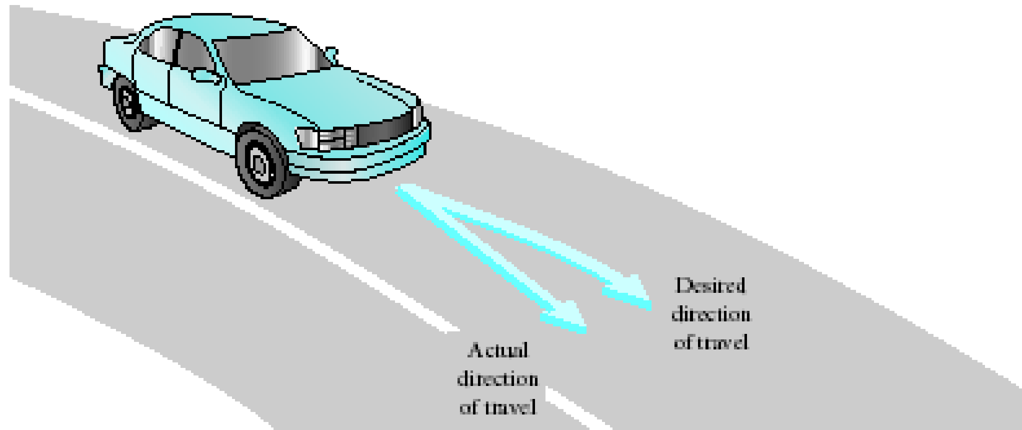
- Their main insight was that the airplane itself had to be inherently unstable, which would give the pilot more control and render the overall flying system (pilot and machine) stable
- The first autopilot was developed by Sperry Corp. in 1912

Examples

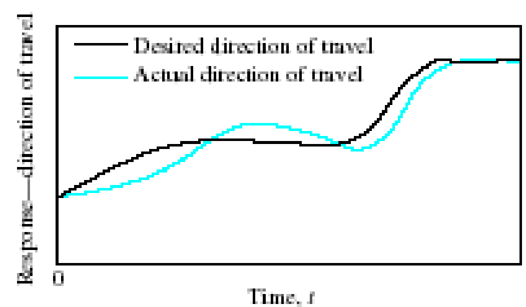
- Automobile steering control system.
- The driver uses the difference between the actual and the desired direction of travel to generate a controlled adjustment of the steering wheel.
- Typical direction-of-travel response.



(a)



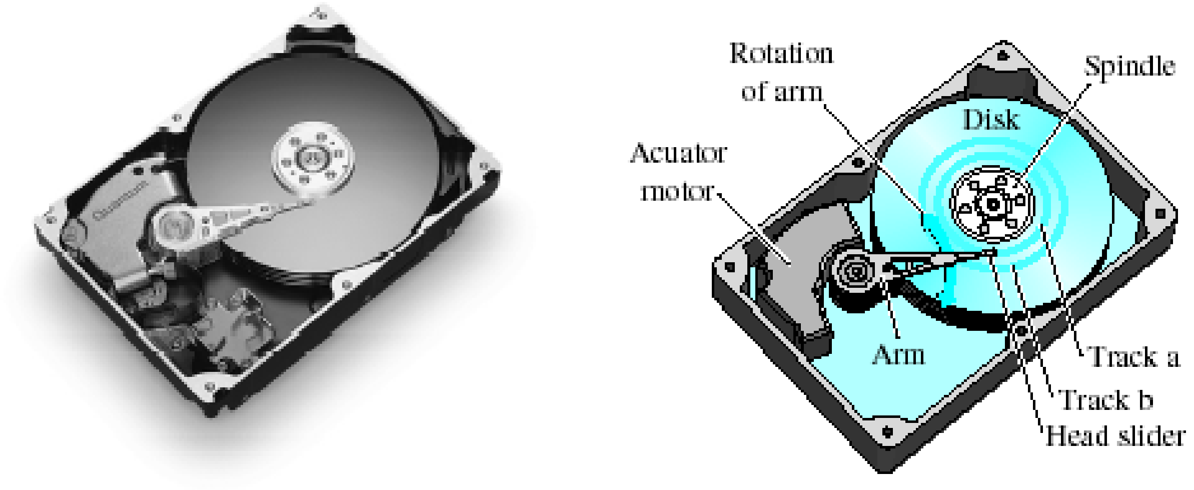
(b)



(c)

Examples (cont.)

- Hard drive head control

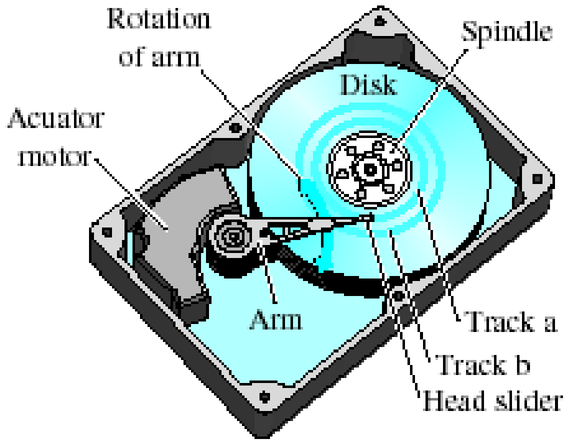


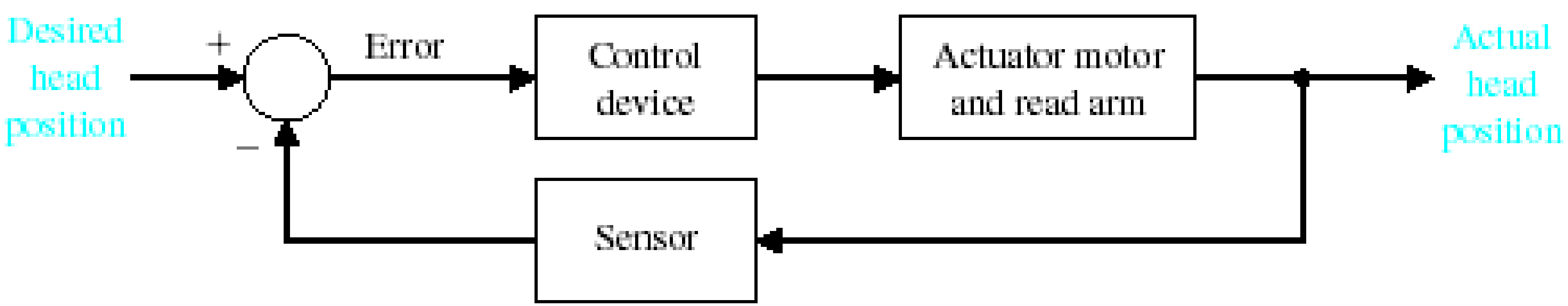
(a) (b)

(a) A disk drive ©1999 Quantum Corporation. All rights reserved.
(b) Diagram of a disk drive.

Examples (cont.)

- Hard drive head control





PID control

- Continuous-time and Discrete-time form

$$u(t) = K_P e(t) + K_I \int_0^t e(t)\,\mathrm{d}t + K_D \frac{\mathrm{d}e(t)}{\mathrm{d}t}$$

$$u[n] = K_P e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

—$u(t), u[n]$ is the correction given by the controller to the system at time t or discrete sample n;
—$e(t), e[n]$ is the error between the set point and current state of the system under control at time t or discrete sample n;
—K_P, K_I, and K_D scale the error, integral (sum) of error, and derivative (difference) of the error, respectively.

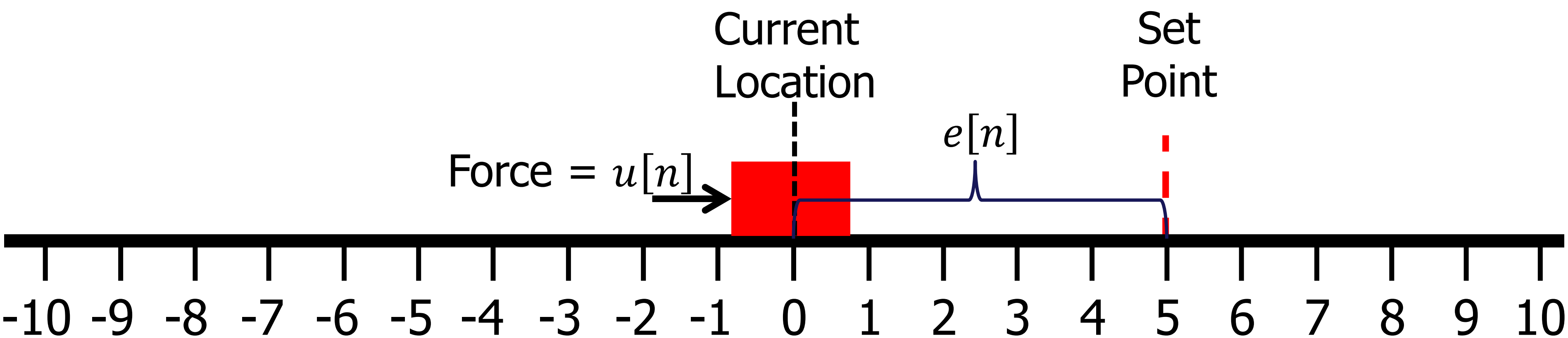
PID control: Example setup

$$\underbrace{u[n]}_{\text{Command sent to actuator}} = K_P \underbrace{e[n]}_{\text{Current error}} + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: P Control

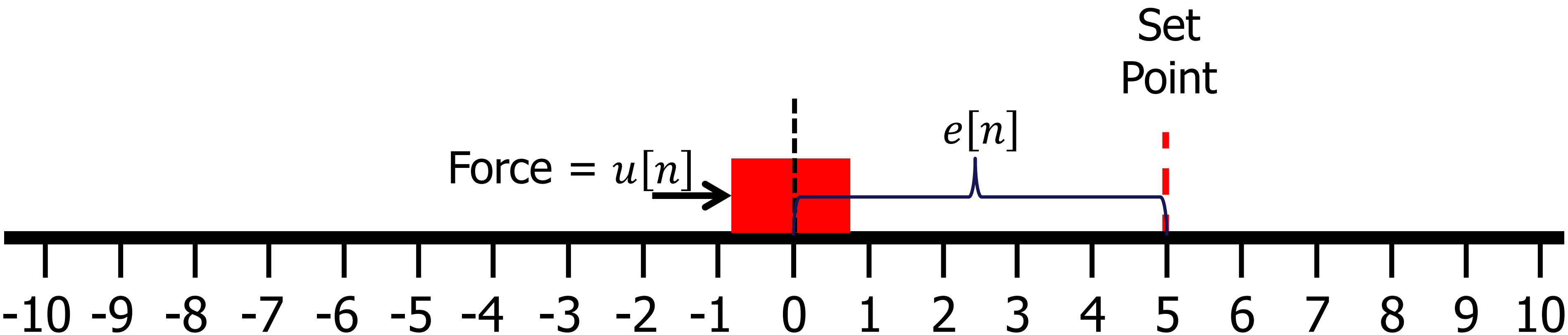
$$u[n] = K_P e[n] + \cancel{K_I \sum_{j=0}^{n} e[j]} + \cancel{K_D(e[n] - e[n-1])}$$

0 0

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: P Control

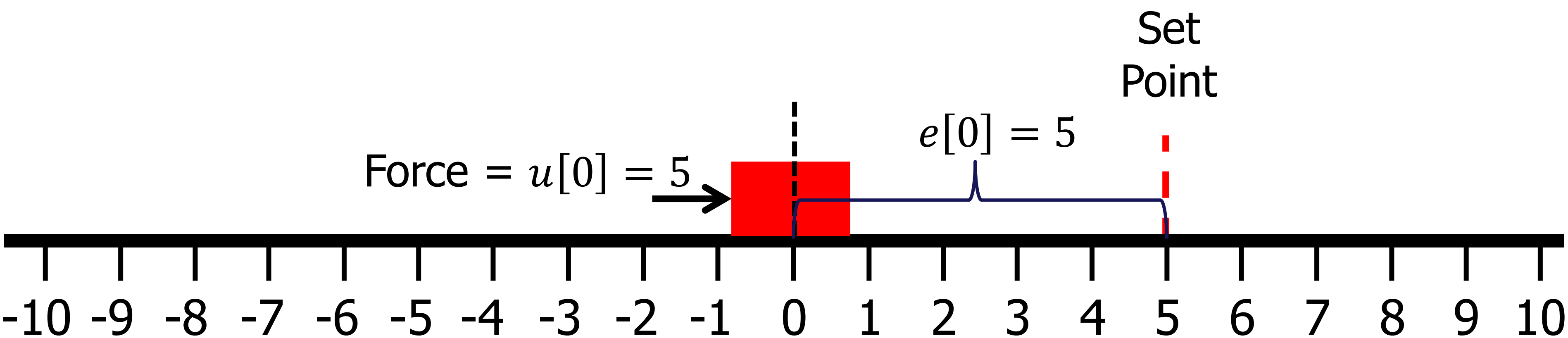
$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

n =0: $e[0]$= 5, $u[0]$=5; n =1: $e[1]$= ?, $u[1]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: P Control

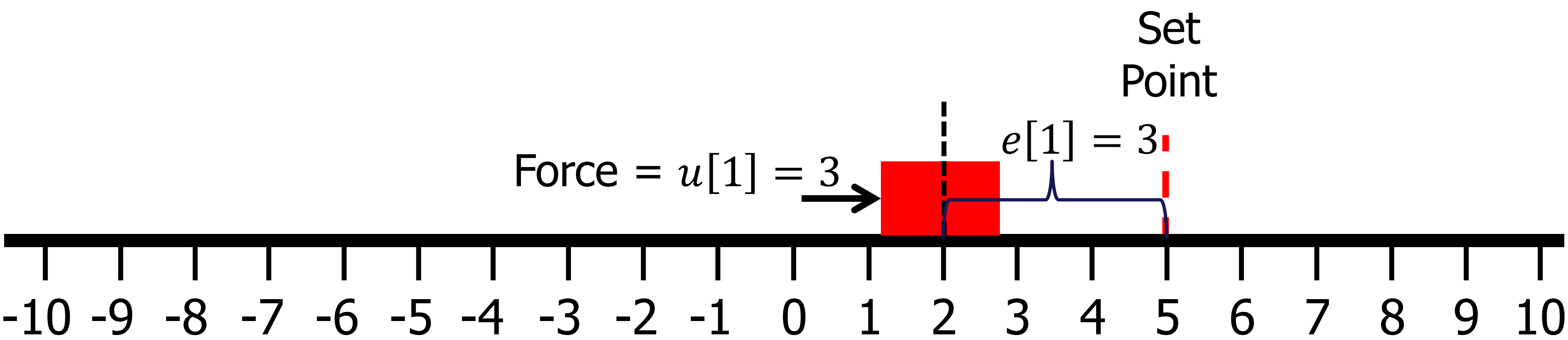
$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

n =0: $e[0]$= 5, $u[0]$=5; n =1: $e[1]$= 3, $u[1]$=3; n =2: $e[2]$= ?, $u[2]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: P Control

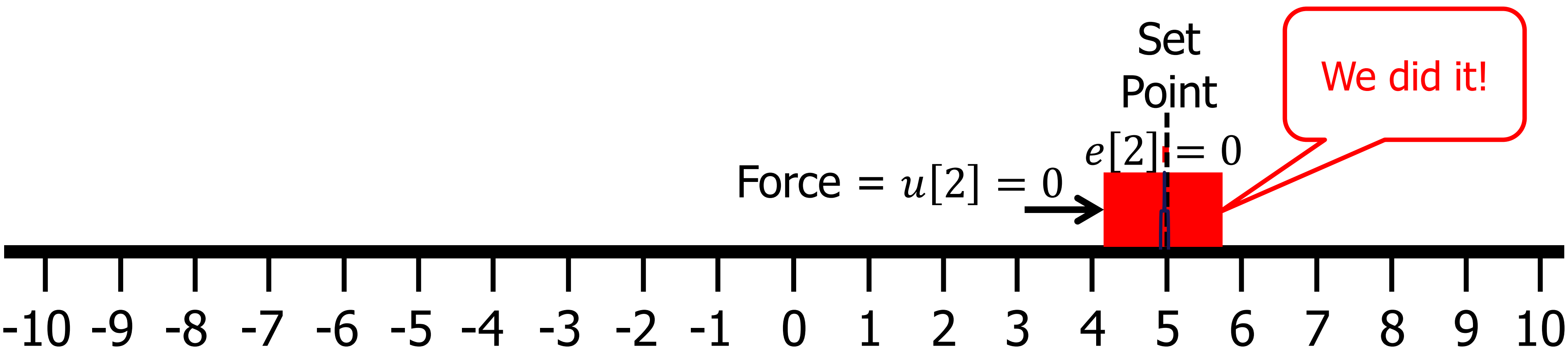
$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

n =0: $e[0]$= 5, $u[0]$=5; n =1: $e[1]$= 3, $u[1]$=3; n =2: $e[2]$= 0, $u[2]$=0;
n =3: $e[3]$= ?, $u[3]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: P Control

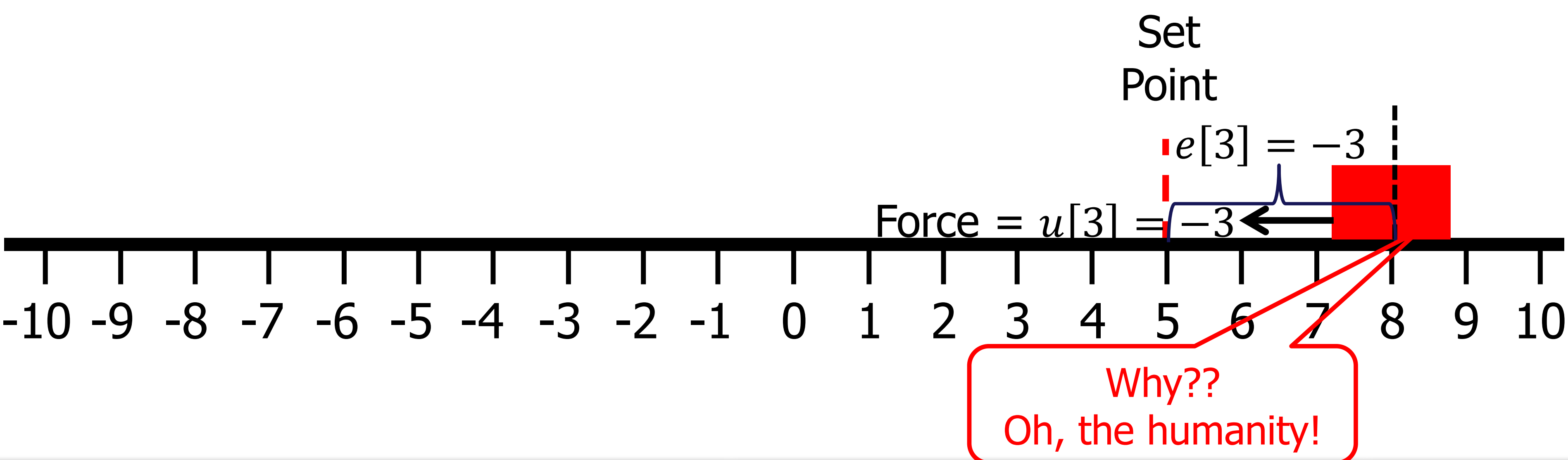
$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{i=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

n =0: $e[0]$= 5, $u[0]$=5; n =1: $e[1]$= 3, $u[1]$=3; n =2: $e[2]$= 0, $u[2]$=0;
n =3: $e[3]$= -3, $u[3]$=-3;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: P Control: Earth to Moon

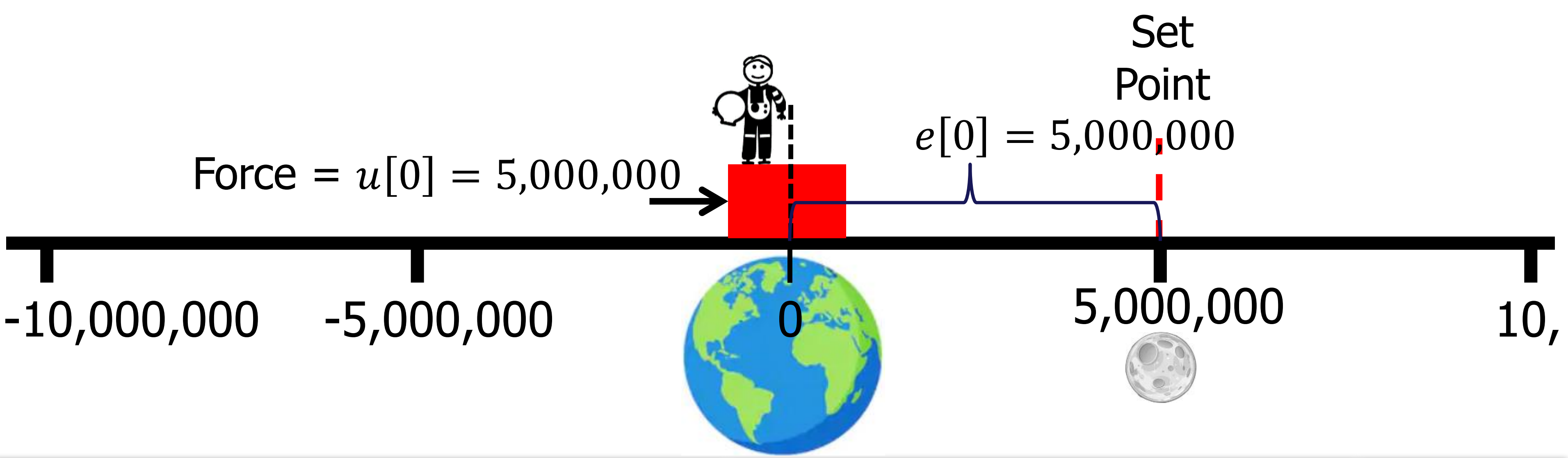
$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

n =0: $e[0]$= 5,000,000 , $u[0]$=5,000,000;

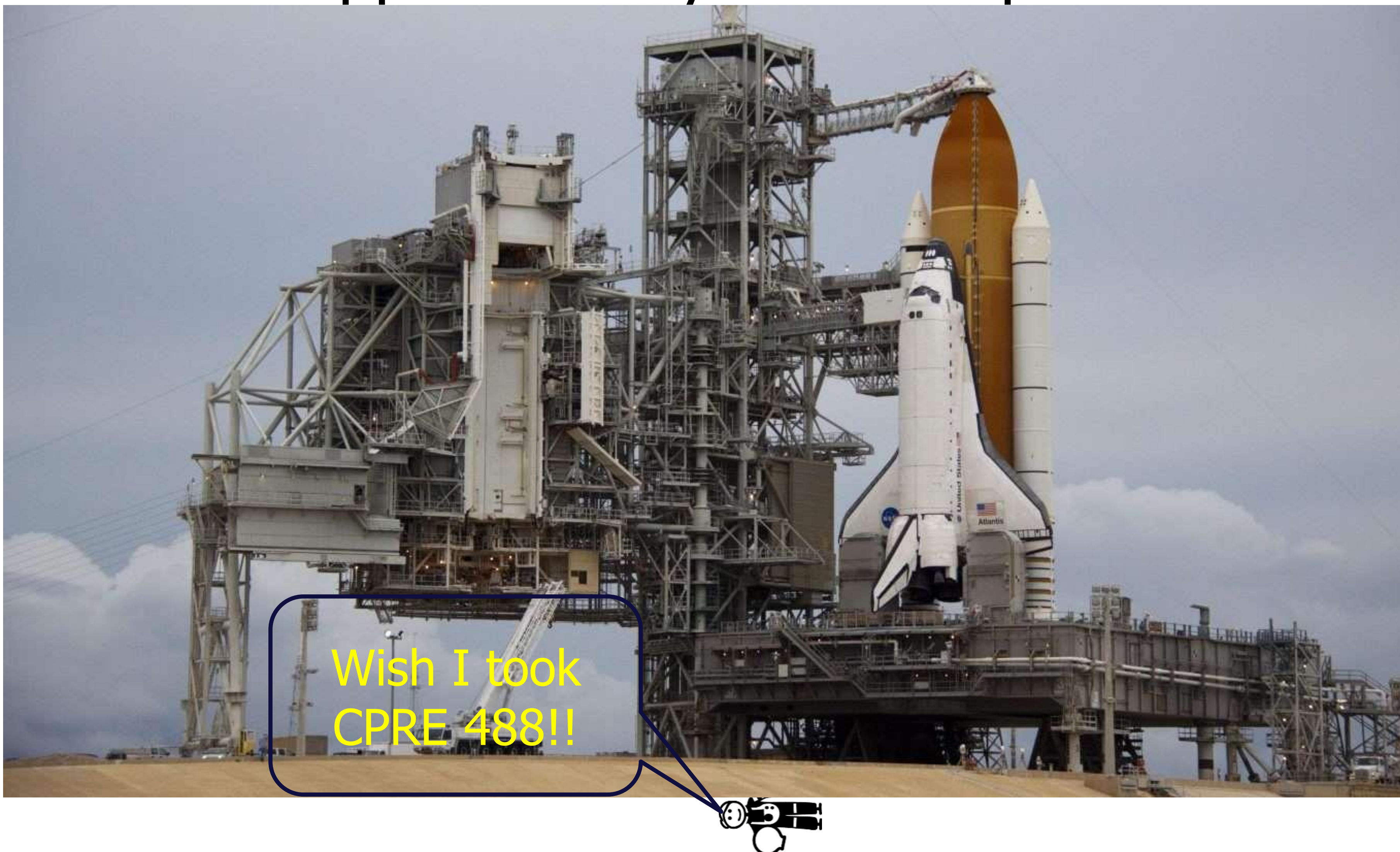
Goal: Have the red block move from location Earth to location Moon

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



Approximately 1 million pounds!!



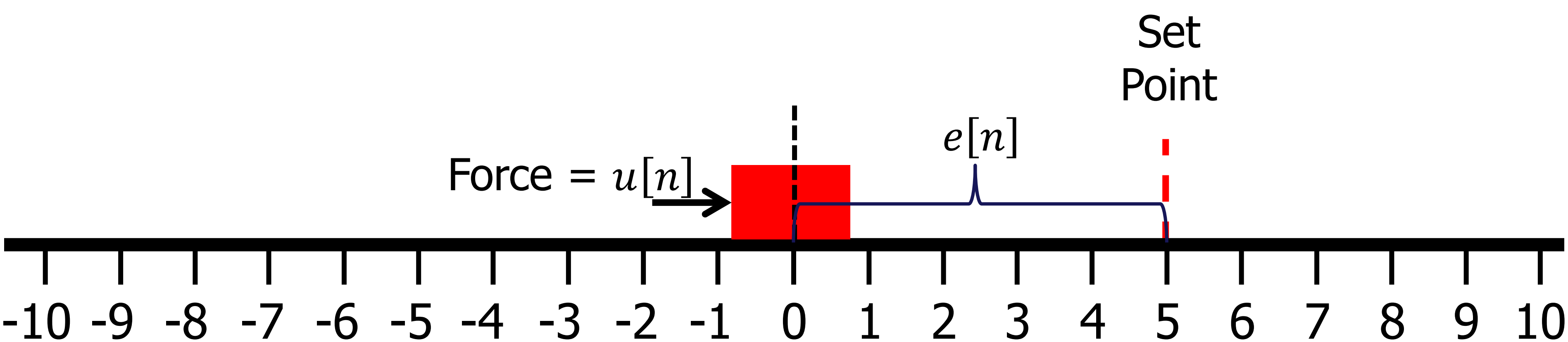
PID control: PD Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

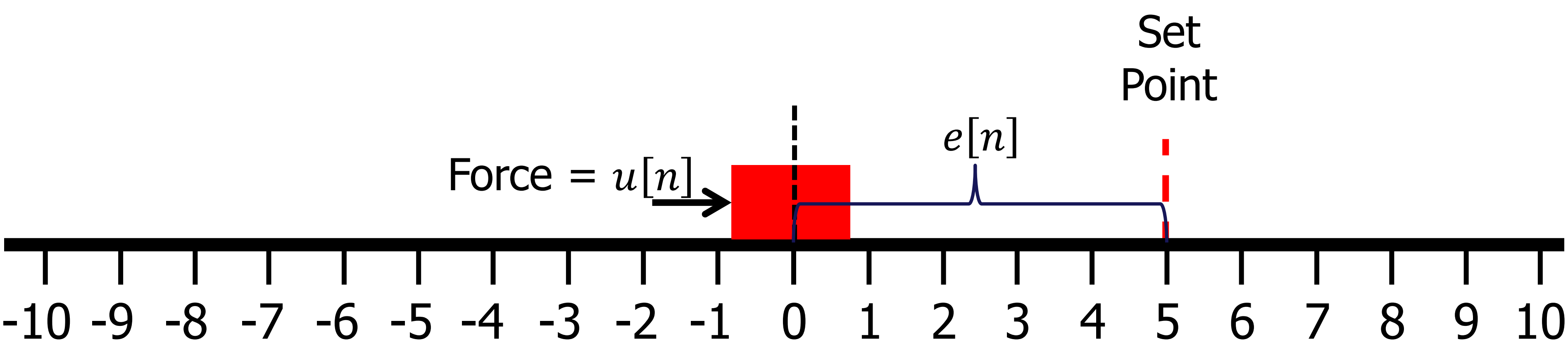
$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

n=0: $e[0]$=?, $(e[0] - e[-1])$=?, $u[0]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

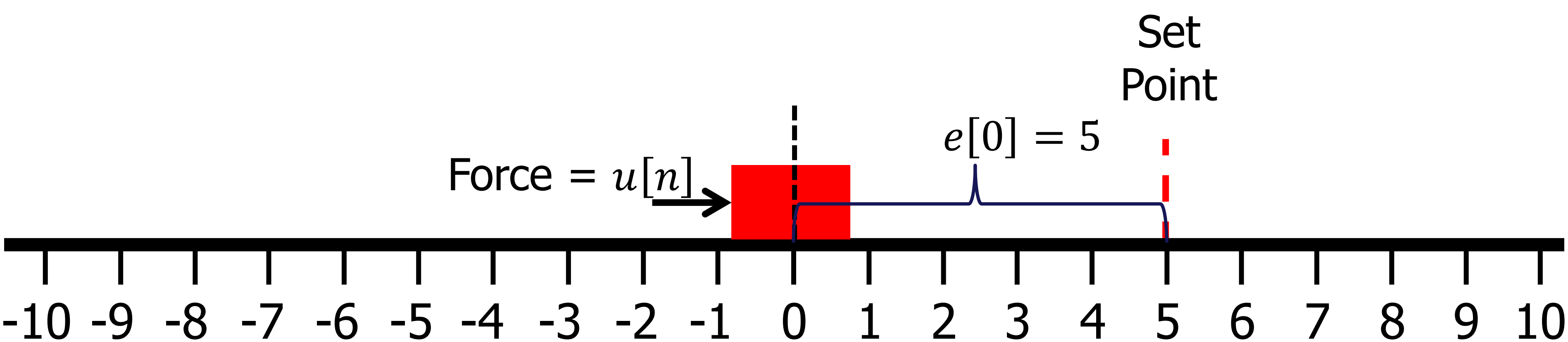
$$u[n] = \overset{1}{K_P}\,\underset{5}{e[n]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{5}{e[n]} - \overset{5}{e[n-1]})}_{0}$$

n=0: $e[0]$=?, $(e[0] - e[-1])$=?, $u[0]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

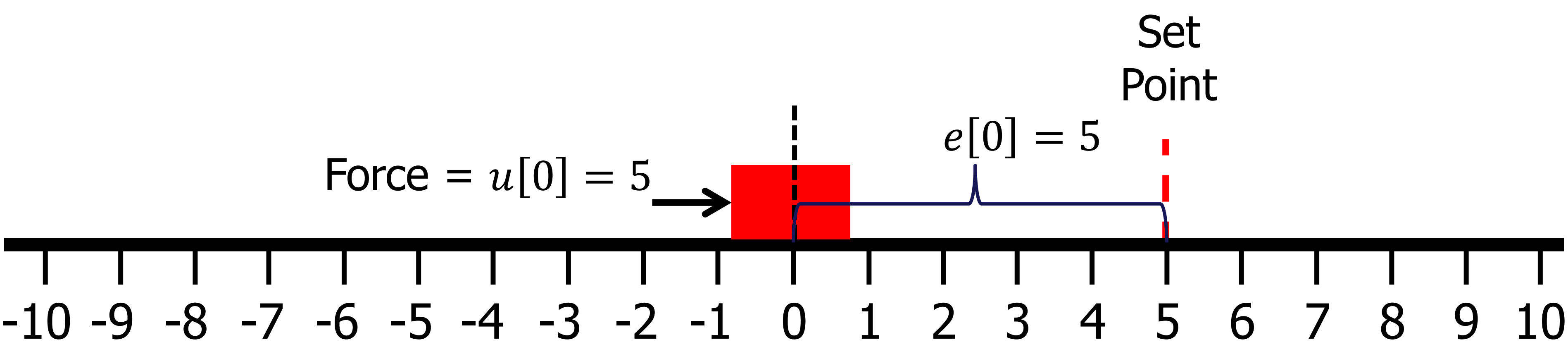
$$u[n] = \overset{1}{K_P}\underset{5}{e[n]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{5}{e[n]} - \overset{5}{e[n-1]})}_{0}$$

n=0: $e[0]$=5, $(e[0] - e[-1])$=0, $u[0]$=5;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

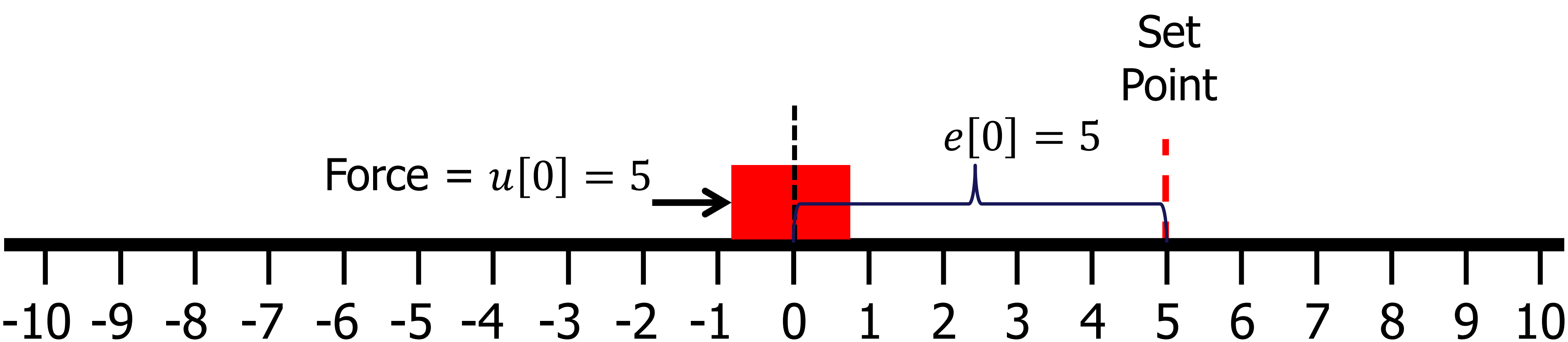
$$u[n] = \overset{1}{K_P}\,\underset{5}{e[n]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{5}{e[n]} - \overset{5}{e[n-1]})}_{0}$$

n=0: $e[0]$=5, $(e[0] - e[-1])$=0, $u[0]$=5; n=1: $e[1]$=?, $(e[1] - e[0])$=?, $u[1]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

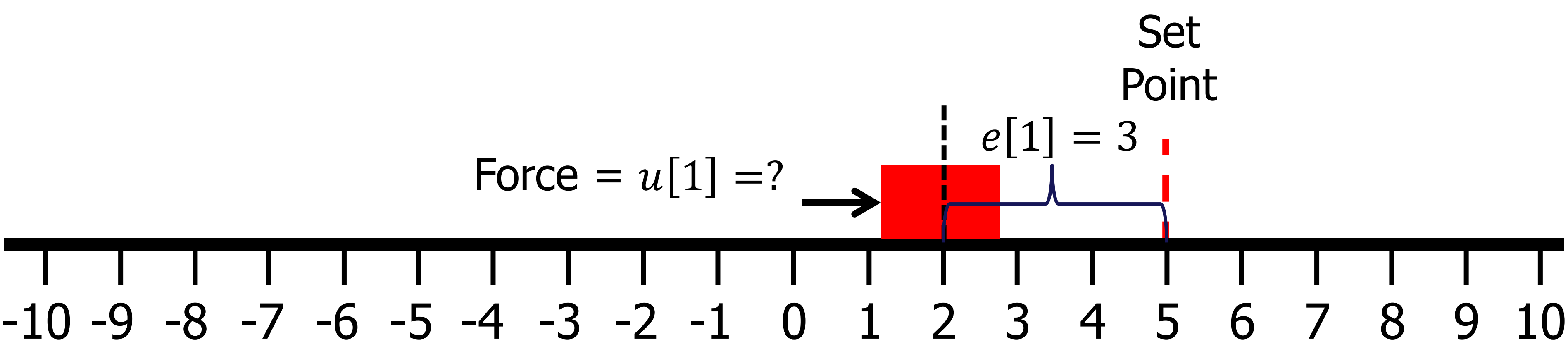
$$u[n] = \overset{1}{K_P}\,\underset{3}{e[1]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{3}{e[1]} - \overset{5}{e[0]})}_{-2}$$

n=0: $e[0]$=5, $(e[0] - e[-1])=0$, $u[0]$=5; n=1: $e[1]$=?, $(e[1] - e[0])=?$, $u[1]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

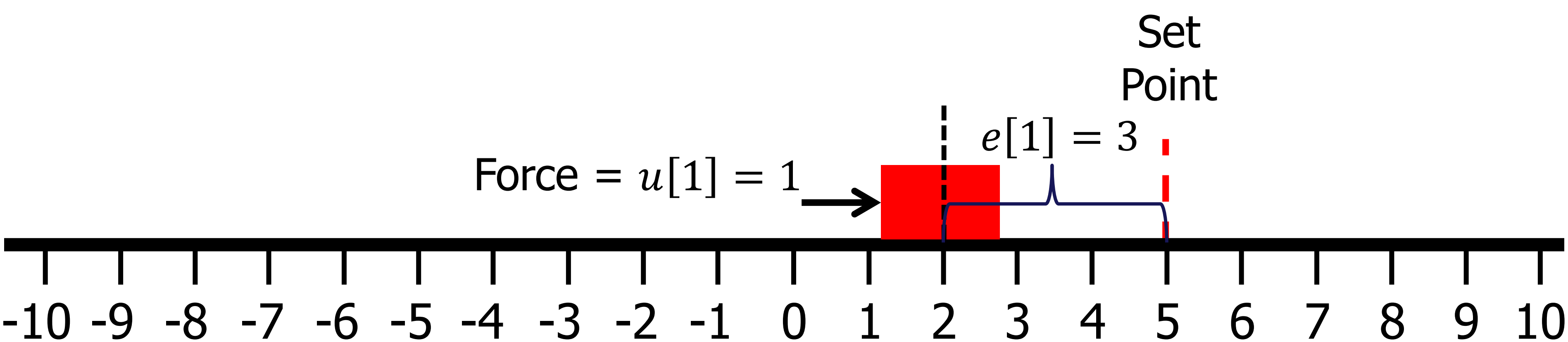
$$u[n] = \overset{1}{K_P}\,\underset{3}{e[1]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{3}{e[1]} - \overset{5}{e[0]})}_{-2}$$

n=0: $e[0]$=5, $(e[0] - e[-1])=0$, $u[0]$=5; n=1: $e[1]$=3, $(e[1] - e[0]) = -2$, $u[1]$=1;
n=2: $e[2]$=?, $(e[2] - e[1]) =$?, $u[2]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

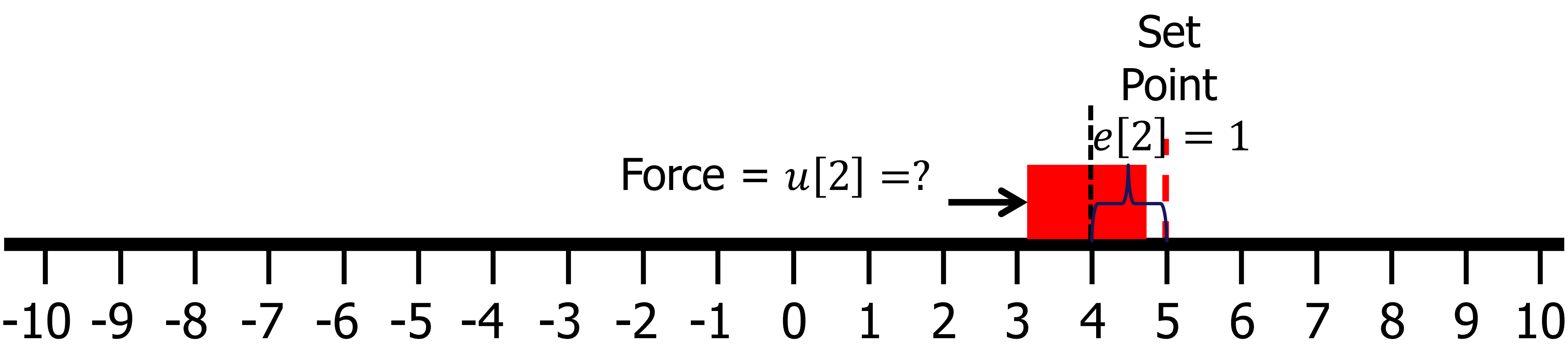
$$u[n] = \overset{1}{K_P}\,\underset{1}{e[2]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{1}{e[2]} - \overset{3}{e[1]})}_{-2}$$

n=0: $e[0]$=5, $(e[0] - e[-1])$=0, $u[0]$=5; n=1: $e[1]$=3, $(e[1] - e[0])= -2$, $u[1]$=1;
n=2: $e[2]$=?, $(e[2] - e[1])=$?, $u[2]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

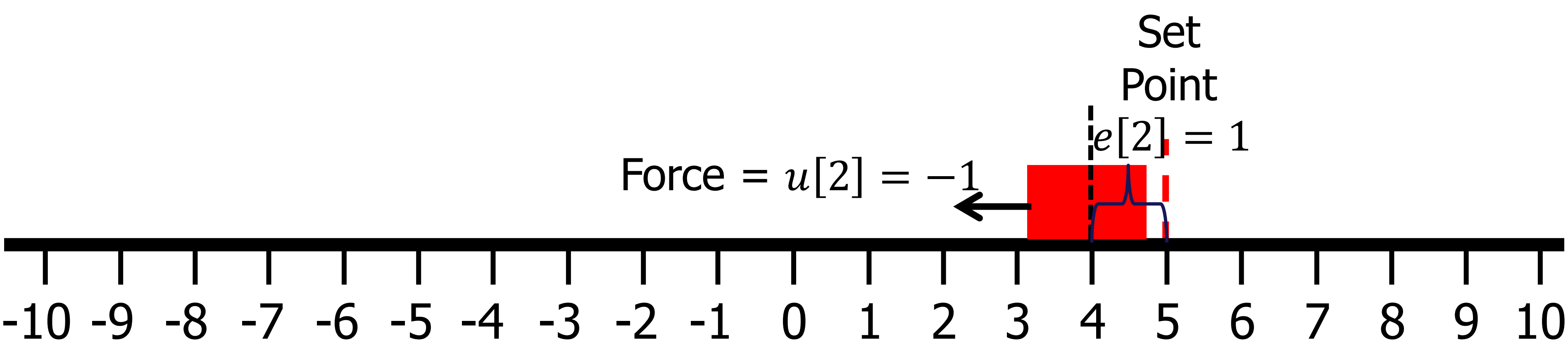
$$u[n] = \overset{1}{K_P}\,\underset{1}{e[2]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{1}{e[2]} - \overset{3}{e[1]})}_{-2}$$

n=0: $e[0]$=5, $(e[0] - e[-1])$=0, $u[0]$=5; n=1: $e[1]$=3, $(e[1] - e[0])= -2$, $u[1]$=1;
n=2: $e[2]$=1, $(e[2] - e[1])= -2$, $u[2]$=-1; n=3: $e[3]$=?, $(e[3] - e[2])=$? , $u[3]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

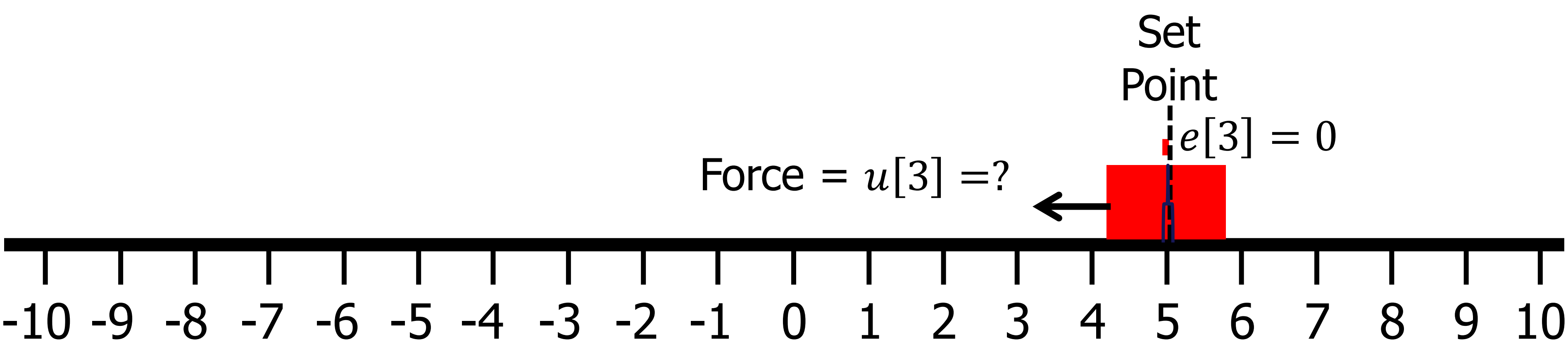
$$u[n] = \overset{1}{K_P}\underset{0}{e[3]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{0}{e[3]} - \overset{1}{e[2]})}_{-1}$$

n=0: $e[0]$=5, $(e[0] - e[-1])$=0, $u[0]$=5; n=1: $e[1]$=3, $(e[1] - e[0])= -2$, $u[1]$=1;
n=2: $e[2]$=1, $(e[2] - e[1])= -2$, $u[2]$=-1; n=3: $e[3]$=?, $(e[3] - e[2])$= ?, $u[3]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

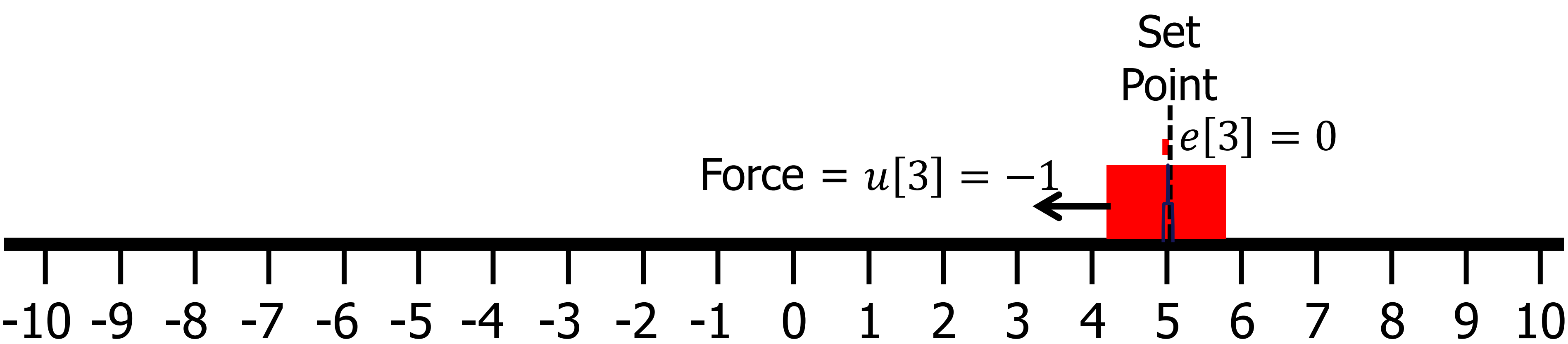
$$u[n] = \overset{1}{K_P}\,\underset{0}{e[3]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{0}{e[3]} - \overset{1}{e[2]})}_{-1}$$

n=0: $e[0]$=5, $(e[0] - e[-1])$=0, $u[0]$=5; n=1: $e[1]$=3, $(e[1] - e[0])= -2$, $u[1]$=1;
n=2: $e[2]$=1, $(e[2] - e[1])= -2$, $u[2]$=-1; n=3: $e[3]$=0, $(e[3] - e[2])=-1$, $u[3]$=-1;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

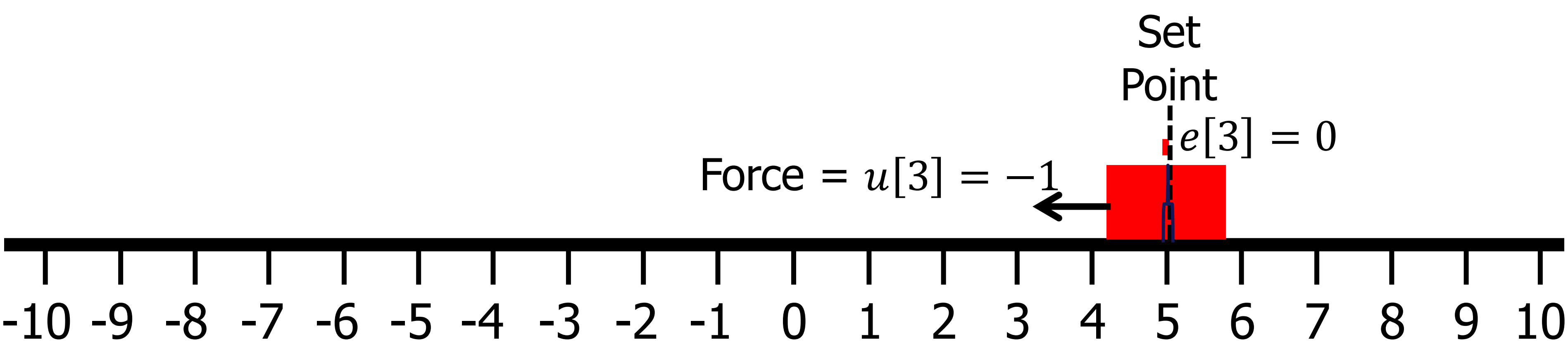
$$u[n] = \overset{1}{K_P}\,\underset{0}{e[3]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{0}{e[3]} - \overset{1}{e[2]})}_{-1}$$

n=3: $e[3]$=0, $(e[3] - e[2])$=−1, $u[3]$=-1;n=4: $e[4]$=?, $(e[4] - e[3])$=?, $u[4]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

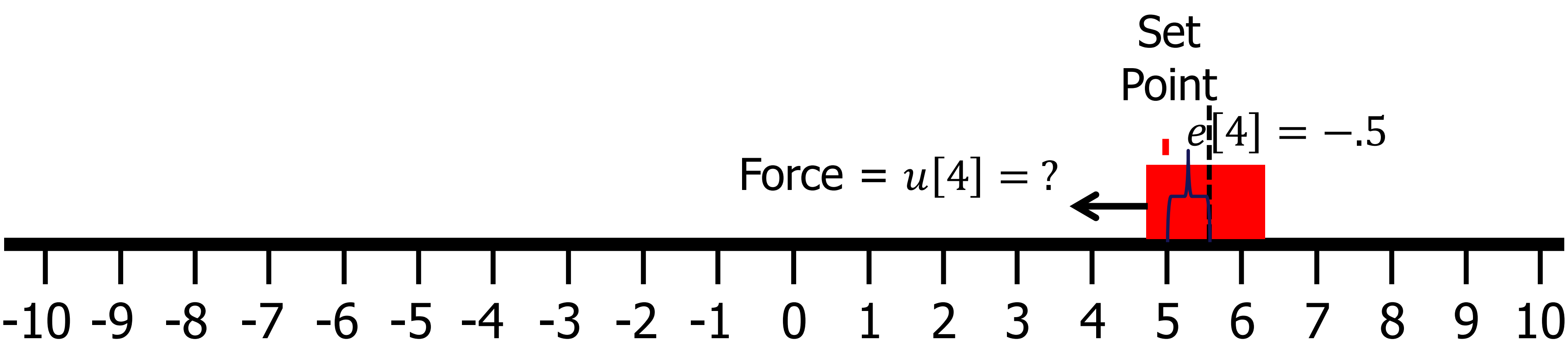
$$u[n] = \overset{1}{K_P}\,\underset{-0.5}{e[4]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{-0.5}{e[4]} - \overset{0}{e[3]})}_{-0.5}$$

n=3: $e[3]$=0, $(e[3] - e[2])$=-1, $u[3]$=-1;n=4: $e[4]$=?, $(e[4] - e[3])$=?, $u[4]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

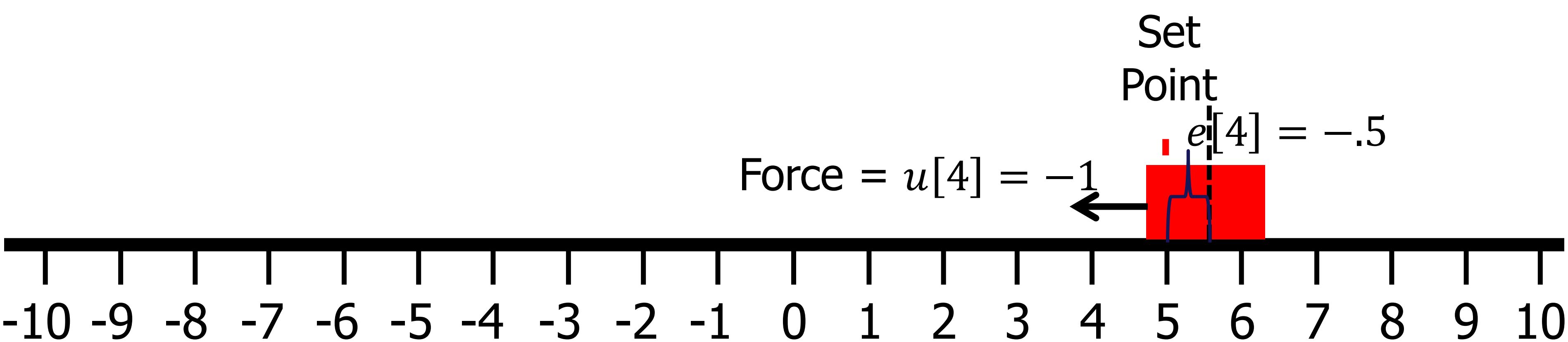
$$u[n] = \overset{1}{K_P}\underset{-0.5}{e[4]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{-0.5}{e[4]} - \overset{0}{e[3]})}_{-0.5}$$

n=3: $e[3]$=0, $(e[3] - e[2])$=-1, $u[3]$=-1;n=4: $e[4]$=-.5, $(e[4] - e[3])$=$-.5$, $u[4]$=-1;
n=5: $e[5]$=?, $(e[5] - e[4])$=?, $u[5]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

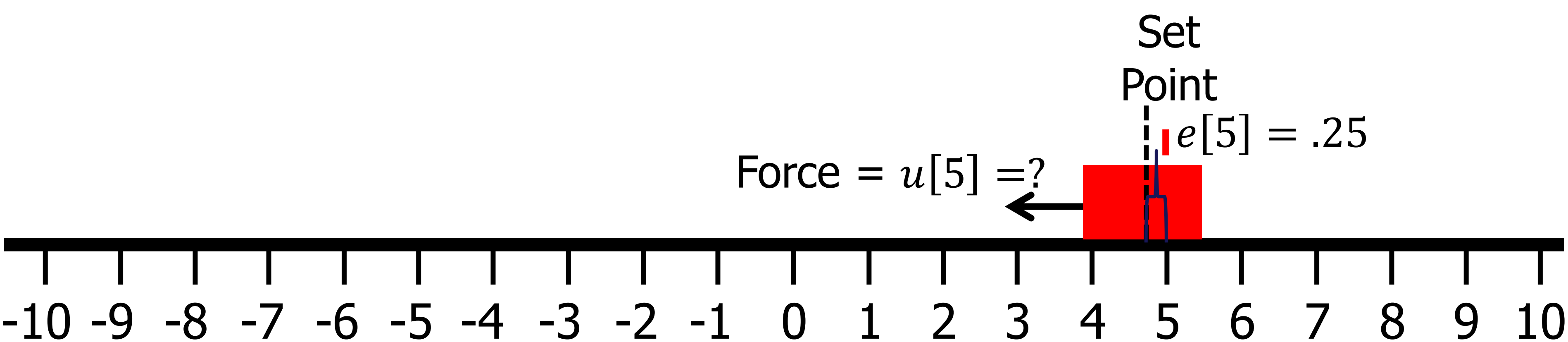
$$u[n] = \underset{0.25}{\overset{1}{K_P}} e[5] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D} \underbrace{(\overset{0.25}{e[5]} - \overset{-0.5}{e[4]})}_{0.75}$$

n=3: $e[3]$=0, $(e[3] - e[2])$=-1, $u[3]$=-1;n=4: $e[4]$=-.5, $(e[4] - e[3])$=$-.5$, $u[4]$=-1;
n=5: $e[5]$=?, $(e[5] - e[4])$=?, $u[5]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PD Control

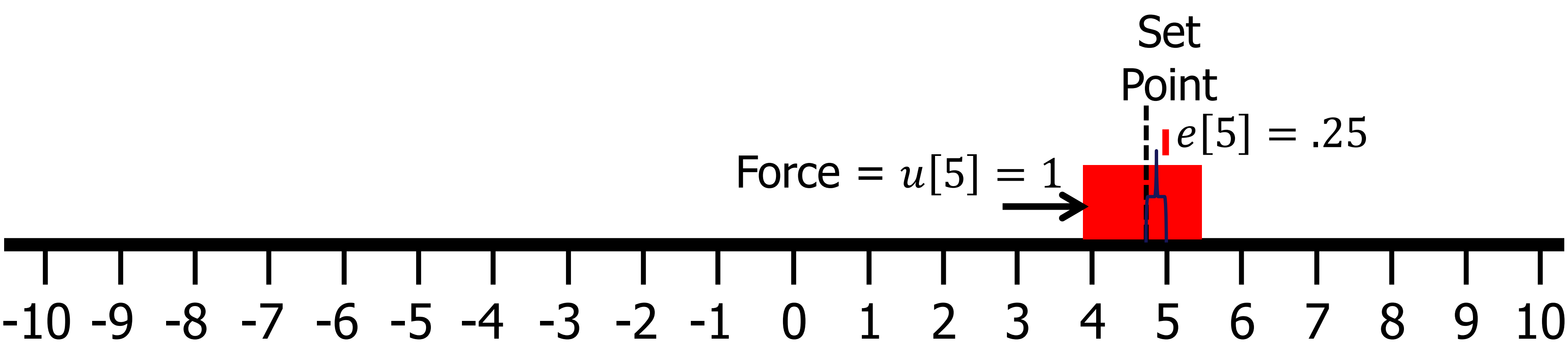
$$u[n] = \overset{1}{K_P}\,\underset{0.25}{e[5]} + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}\underbrace{(\overset{0.25}{e[5]} - \overset{-0.5}{e[4]})}_{0.75}$$

n=3: $e[3]$=0, $(e[3] - e[2])$=-1, $u[3]$=-1;n=4: $e[4]$=-.5, $(e[4] - e[3])$=$-.5$, $u[4]$=-1;
n=5: $e[5]$=0.25, $(e[5] - e[4])$=0.75, $u[5]$=1;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a level surface with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



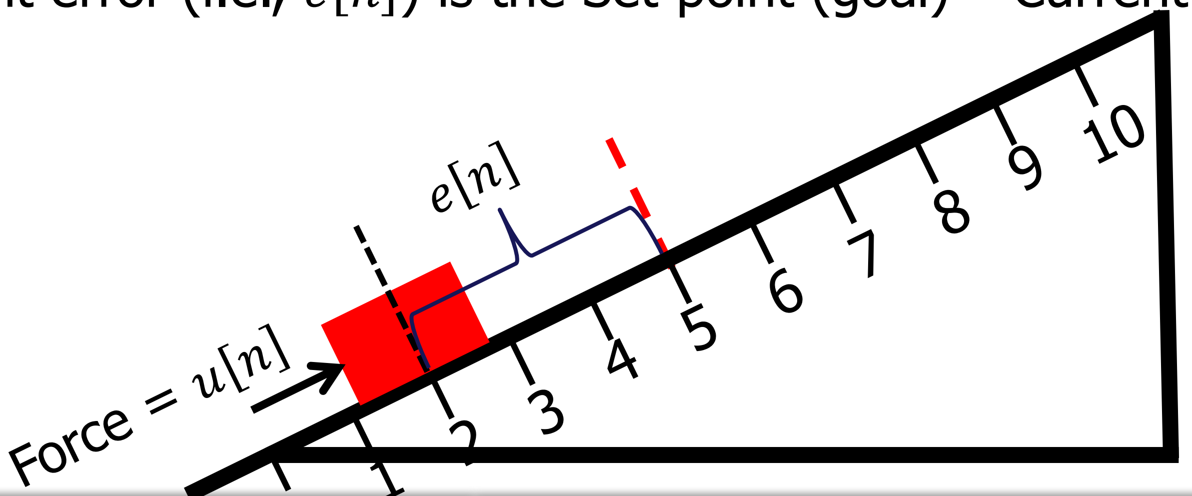
PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

<u>Forces acting on the block:</u>

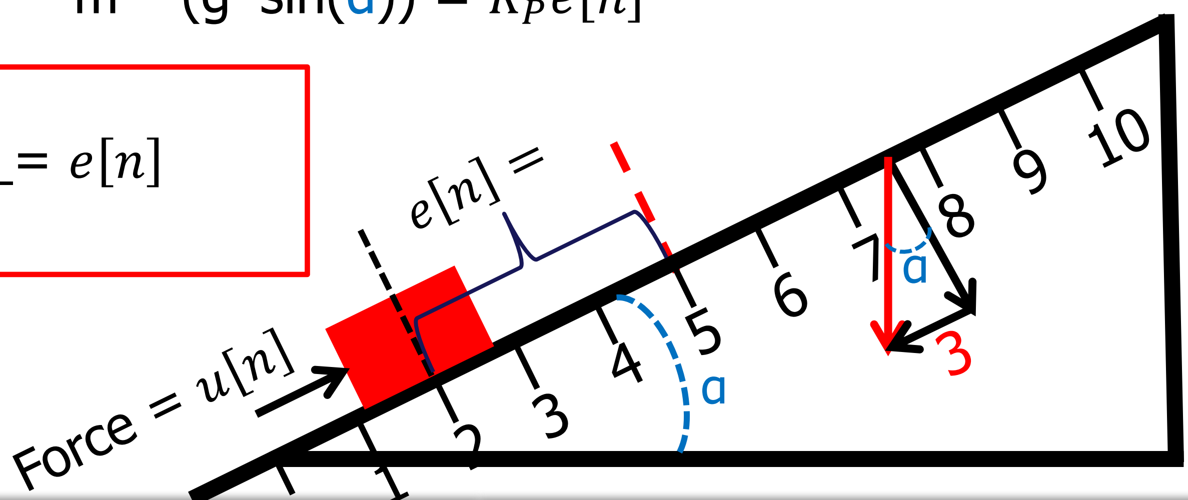
<u>Gravity acting in direction of ramp</u>
F = m*a = m * (g*sin(α))

=

<u>Control law (i.e. u[n])</u>
$u = K_P e[n]; \textit{when block is not moving, and } K_I = 0$

m * (g*sin(α)) = $K_P e[n]$

$$\frac{m * (g*\sin(\alpha))}{K_P} = e[n]$$



PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

<u>Forces acting on the block:</u>

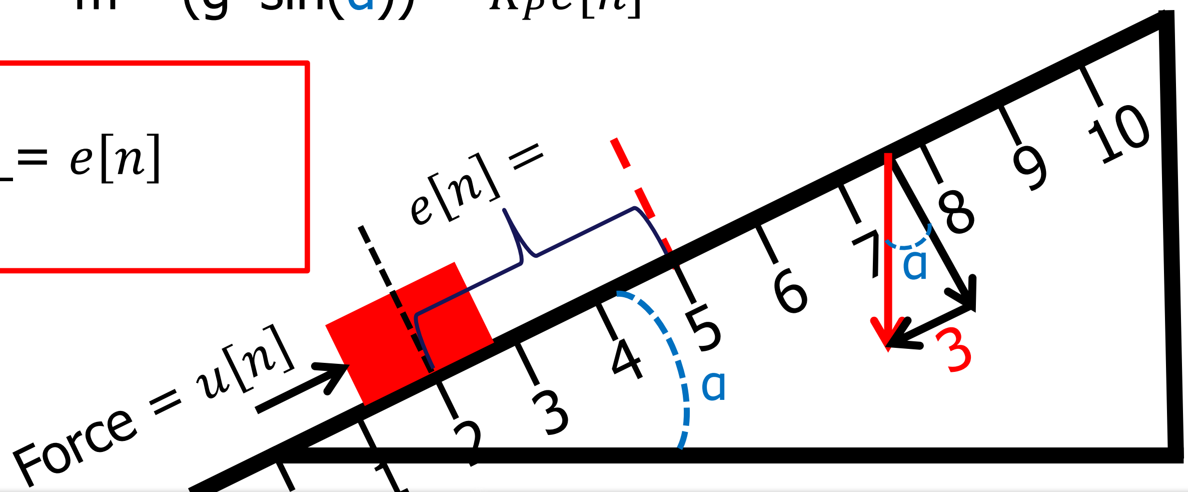
<u>Gravity acting in direction of ramp</u>
F = m*a = m * (g*sin(α))

=

<u>Control law (i.e. u[n])</u>
$u = K_P e[n];$ *when block is not moving, and* $K_I = 0$

m * (g*sin(α)) = $K_P e[n]$

$$\frac{m * (g*\sin(\alpha))}{K_P{=}1} = e[n]$$



PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

Forces acting on the block:

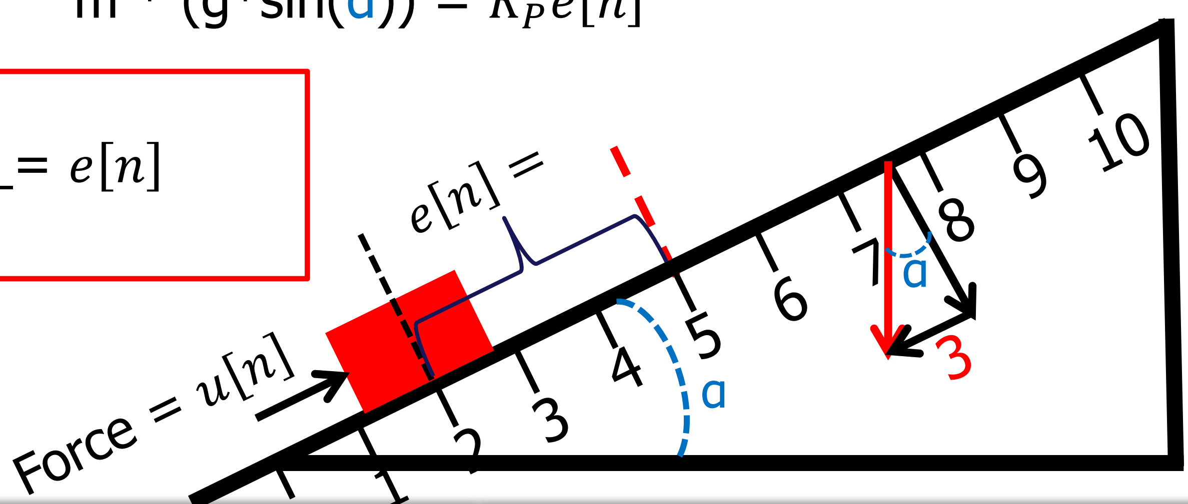
Gravity acting in direction of ramp
F = m*a = m * (g*sin(α))

=

Control law (i.e. u[n])
$u = K_P e[n]; \textit{when block is not moving, and } K_I = 0$

m * (g*sin(α)) = $K_P e[n]$

$$\frac{m * (g*\sin(\alpha))}{K_P = 2} = e[n]$$



PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

Forces acting on the block:

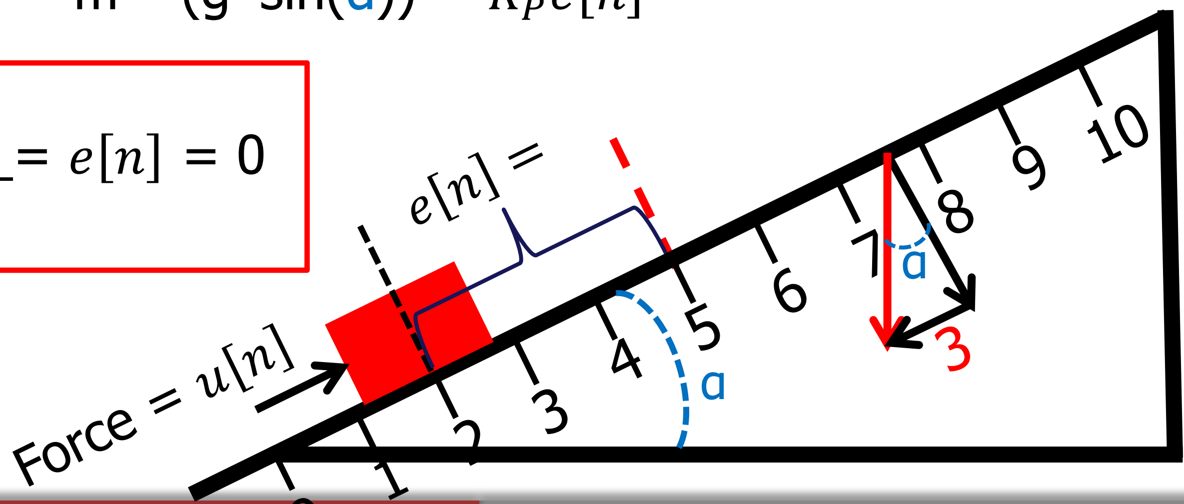
Gravity acting in direction of ramp
F = m*a = m * (g*sin(α))

=

Control law (i.e. u[n])
$u = K_P e[n]; when\ block\ is\ not\ moving,$
$and\ K_I = 0$

m * (g*sin(α)) = $K_P e[n]$

$$\frac{m * (g*\sin(\alpha))}{K_P=2} = e[n] = 0$$



PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

<u>Forces acting on the block:</u>

<u>Gravity acting in direction of ramp</u>
F = m*a = m * (g*sin(α))

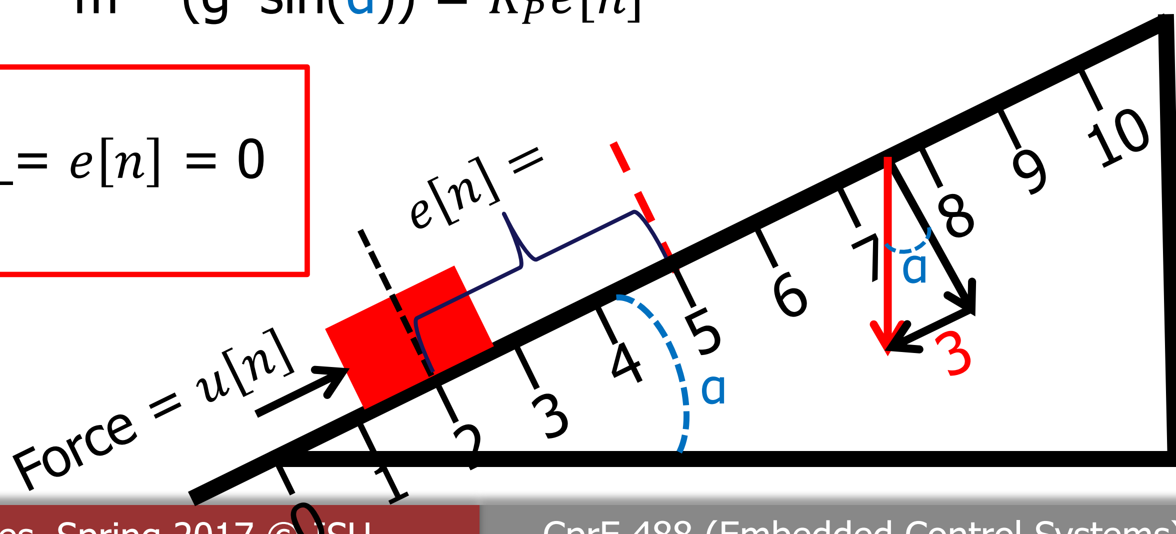
=

<u>Control law (i.e. u[n])</u>
$u = K_P e[n]$; *when block is not moving, and* $K_I = 0$

m * (g*sin(α)) = $K_P e[n]$

$\dfrac{\text{m * (g*sin(α))}}{K_P = \infty} = e[n] = 0$

That's a problem!!!



PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D}(e[n] - e[n-1])$$

Forces acting on the block:

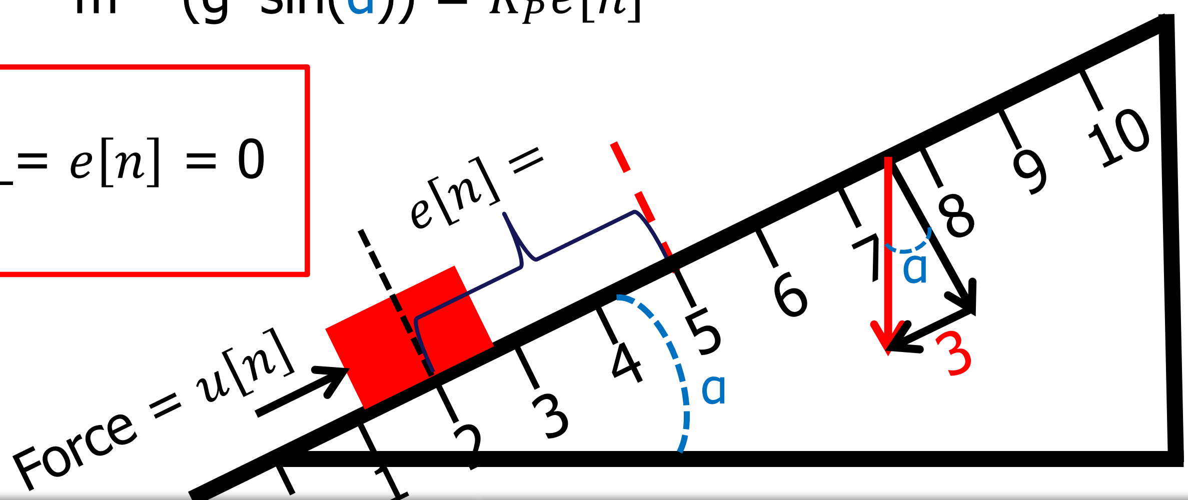
Gravity acting in direction of ramp
F = m*a = m * (g*sin(α))

=

Control law (i.e. u[n])
$u = K_P e[n]; \textit{when block is not moving, and } K_I = 0$

m * (g*sin(α)) = $K_P e[n]$

$$\frac{m * (g*\sin(\alpha))}{K_P=\infty} = e[n] = 0$$

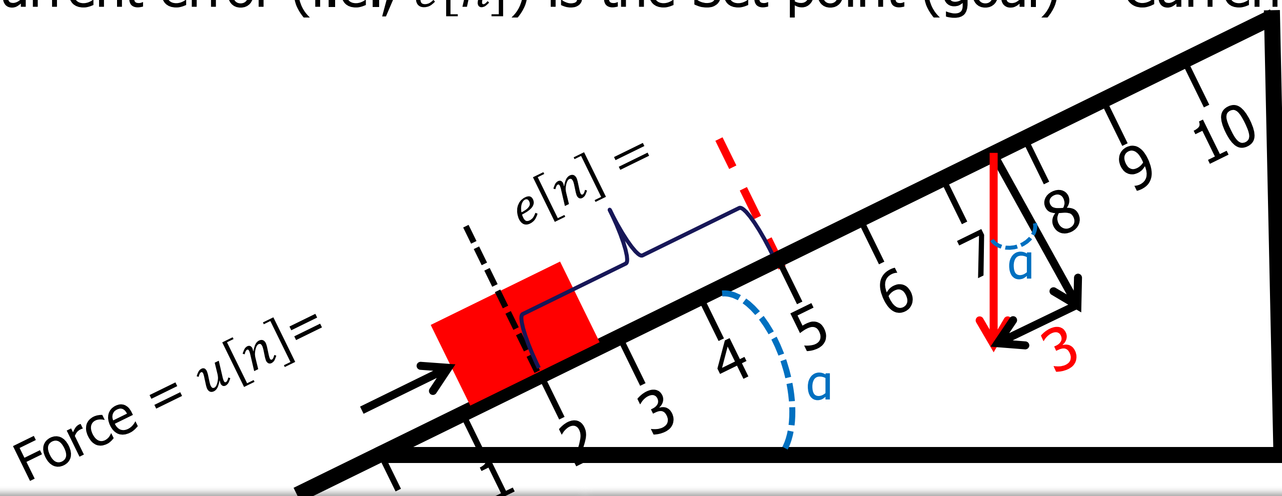


$$u[n] = \overset{1}{K_P} e[n] + K_I \sum_{j=0}^{n} e[j] + \overset{1}{K_D} (e[n] - e[n-1])$$

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



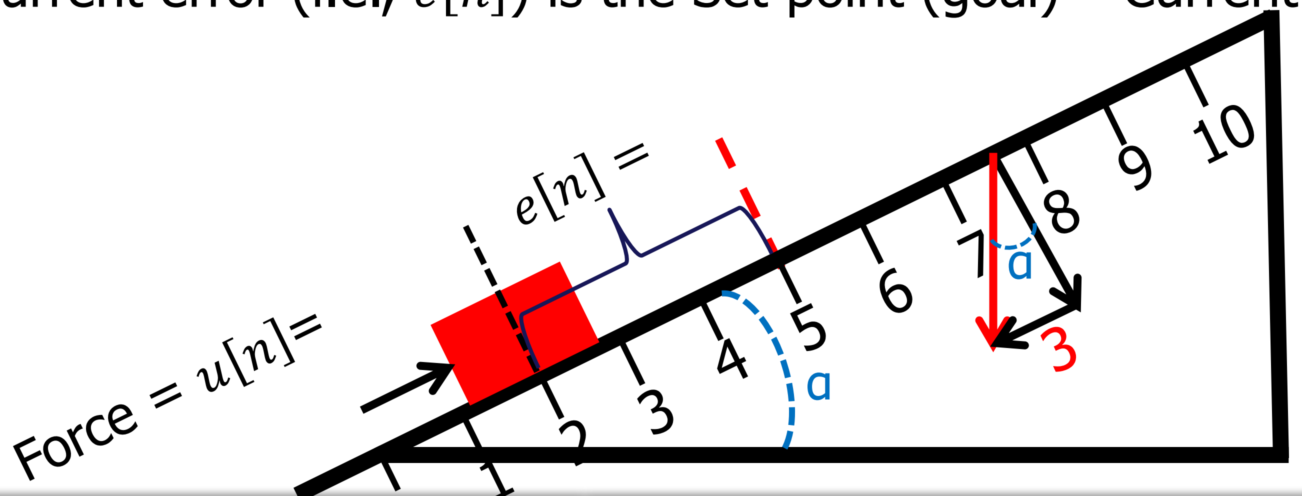
PID control: PID Control

$$u[n] = \overset{1}{K_P} e[n] + \overset{.5}{K_I} \sum_{j=0}^{n} e[j] + \overset{1}{K_D} (e[n] - e[n-1])$$

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

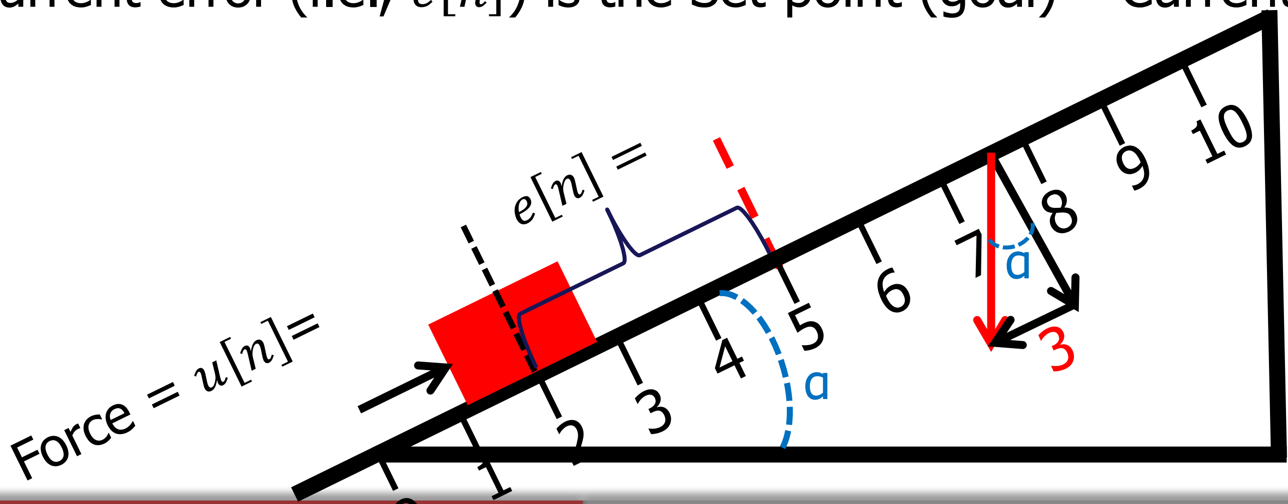
$$u[n] = \overset{1}{K_P} e[n] + \overset{.5}{K_I} \sum_{j=0}^{n} e[j] + \overset{1}{K_D} (e[n] - e[n-1])$$

n=0: $e[0]$=?, $esum[0]$ =?, $(e[0] - e[-1])$=?, $u[0]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

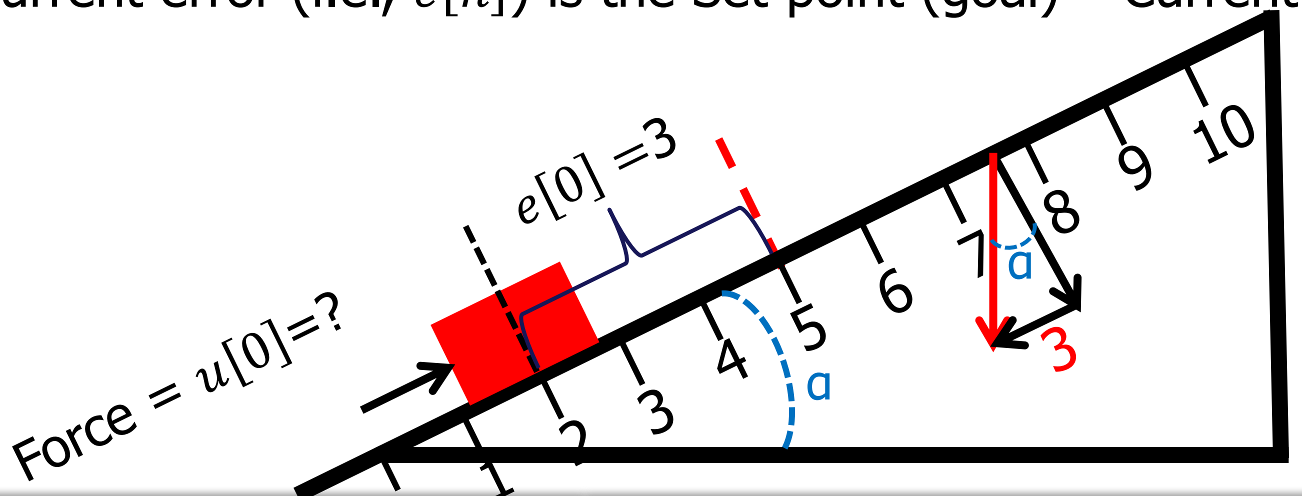
$$u[n] = \overset{1}{K_P} e[n] + \overset{.5}{K_I} \sum_{j=0}^{n} e[j] + \overset{1}{K_D} (e[n] - e[n-1])$$

n=0: $e[0]$=?, $esum[0]$ =?, $(e[0] - e[-1])$=?, $u[0]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

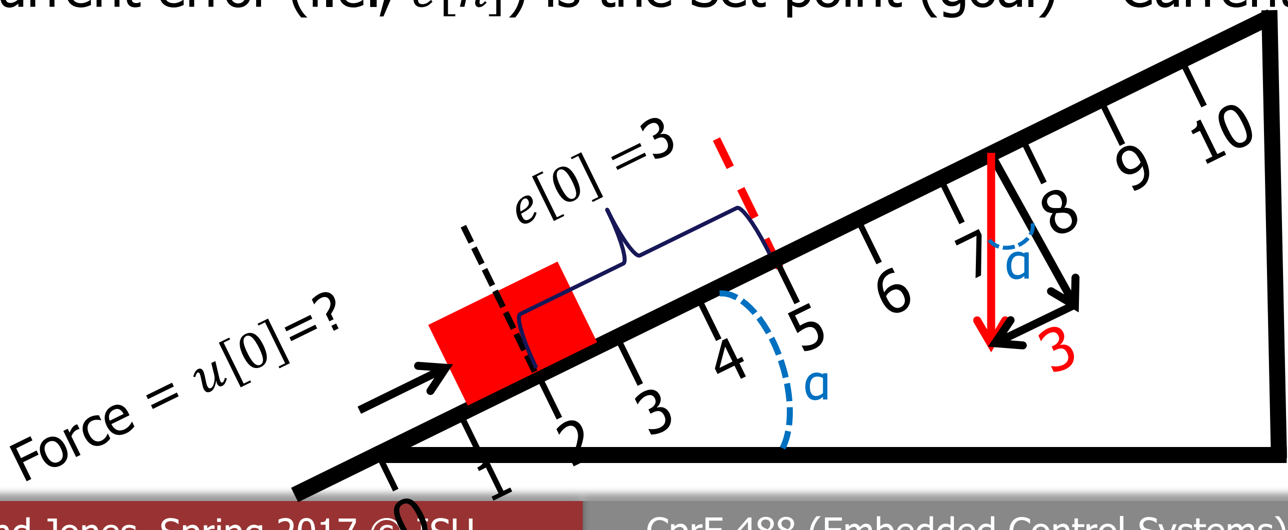
$$\underbrace{u[0]}_{4.5\,=} = \overset{1}{K_P}\underbrace{\overset{3}{e[0]}}_{3} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\underbrace{\overset{3}{e[0]}}_{pre\_sum=0}}_{.5*3\,=\,1.5} + \overset{1}{K_D}\underbrace{(\overset{3}{e[0]} - \overset{3}{e[-1]})}_{0}$$

n=0: $e[0]$=?, $esum[0]$ =?, $(e[0] - e[-1])$=?, $u[0]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{4.5\ =}{u[0]} = \underbrace{\overset{1}{K_P}\,\overset{3}{e[0]}}_{3} + \underbrace{\overset{.5}{K_I} \sum_{j=0}^{n} \overset{3}{e[0]}}_{.5*3\ =\ 1.5} + \overset{1}{K_D}\underbrace{(\overset{3}{e[0]} - \overset{3}{e[-1]})}_{0}$$

pre_sum=0

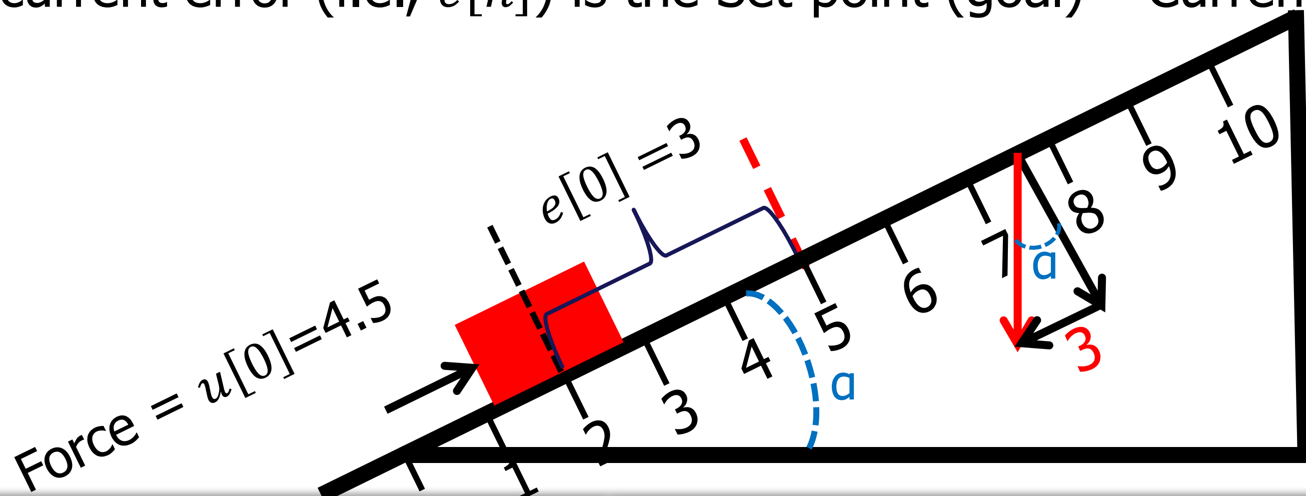
n=0: $e[0]$=3, $esum[0] = 3$, $(e[0] \ - e[-1])$=0, $u[0]$=4.5;

n=1: $e[1]$=?, $esum[1] =?$, $(e[1] \ - e[0])$=0,? $u[1]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{3.5\ =}{u[1]} = \overset{1}{K_P}\underbrace{\overset{2}{e[1]}}_{2} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\overset{2}{e[1]}}_{.5*5\ =\ 2.5} + \overset{1}{K_D}\underbrace{(\overset{2}{e[1]} - \overset{3}{e[0]})}_{-1}$$

pre_sum=3

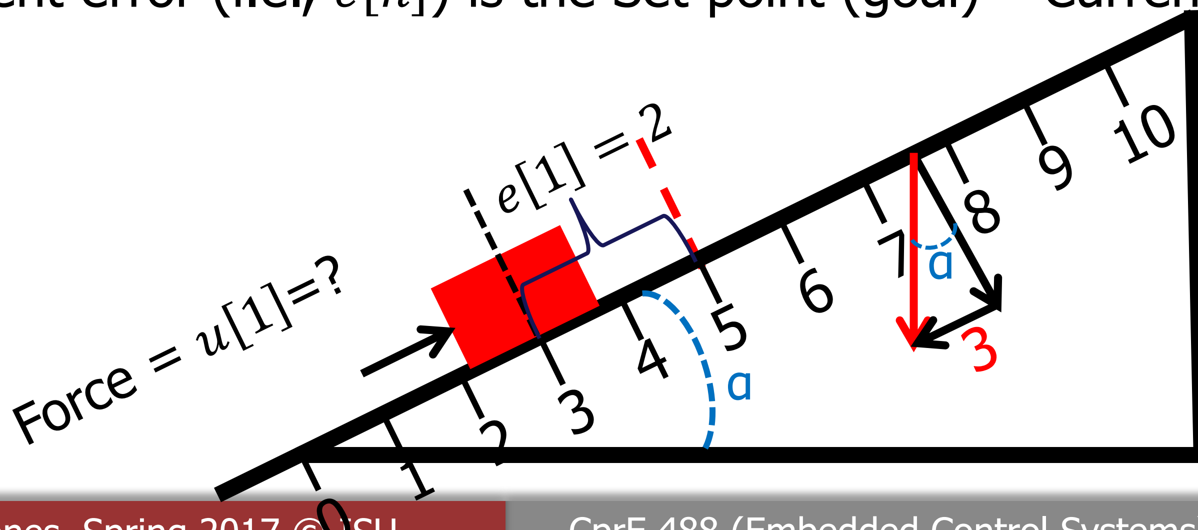
n=0: $e[0]$=3, $esum[0] = 3$, $(e[0] - e[-1])$=0, $u[0]$=4.5;

n=1: $e[1]$=?, $esum[1]$ =?, $(e[1] - e[0])$=0,? $u[1]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underbrace{u[1]}_{3.5} = \overset{1}{K_P}\underbrace{\overset{2}{e[1]}}_{2} + \overset{.5}{K_I}\underbrace{\sum_{j=0}^{n} \overset{2}{e[1]}}_{\text{pre\_sum}=3,\ .5*5\,=\,2.5} + \overset{1}{K_D}\underbrace{(\overset{2}{e[1]} - \overset{3}{e[0]})}_{-1}$$

3.5 =

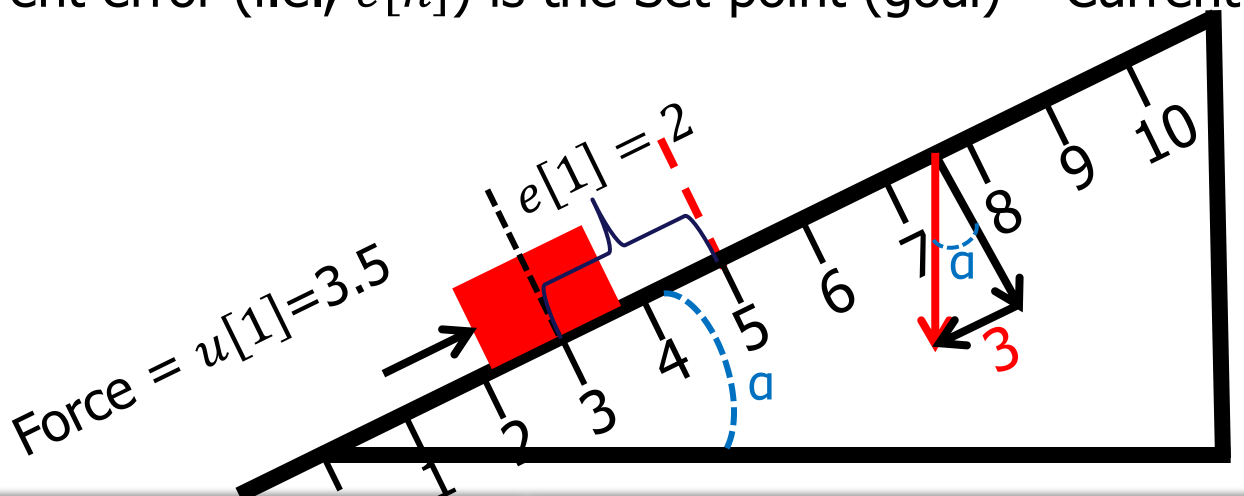
n=0: $e[0]$=3, $esum[0] = 3$, $(e[0] - e[-1])= 0$, $u[0]$=4.5;

n=1: $e[1]$=2, $esum[1] = 5$, $(e[1] - e[0])= -1$, $u[1]$=3.5;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a <u>ramp</u> with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{3.5\ =}{u[1]} = \overset{1}{K_P}\underbrace{\overset{2}{e[1]}}_{2} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\overset{2}{e[1]}}_{\text{pre\_sum}=3;\ .5*5\,=\,2.5} + \overset{1}{K_D}\underbrace{(\overset{2}{e[1]} - \overset{3}{e[0]})}_{-1}$$

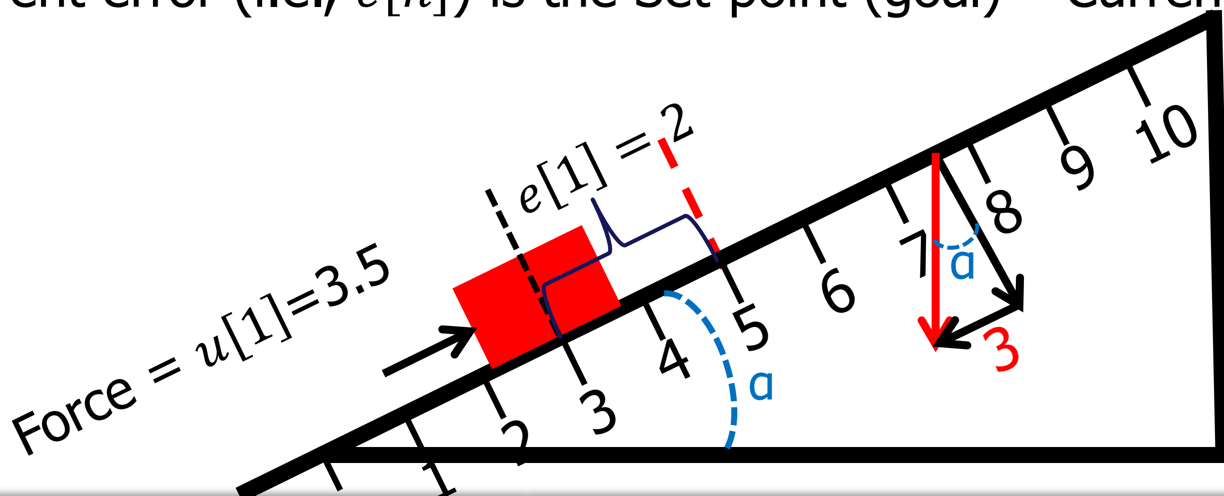
n=1: $e[1]$=2, $esum[1] = 5$, $(e[1] \; - e[0]) = -1$, $u[1]$=3.5;

n=2: $e[2]$=?, $esum[2]$ =?, $(e[2] \; - e[1])$=?, $u[2]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$u[2] = \overset{1}{K_P}\,\underbrace{\overset{1}{e[2]}}_{1} + \underbrace{\overset{.5}{K_I} \sum_{j=0}^{n} \overset{1}{e[2]}}_{\text{pre\_sum}=5} + \overset{1}{K_D}\,\underbrace{(\overset{1}{e[2]} - \overset{2}{e[1]})}_{-1}$$

3 = 1 .5*6 = 3 -1

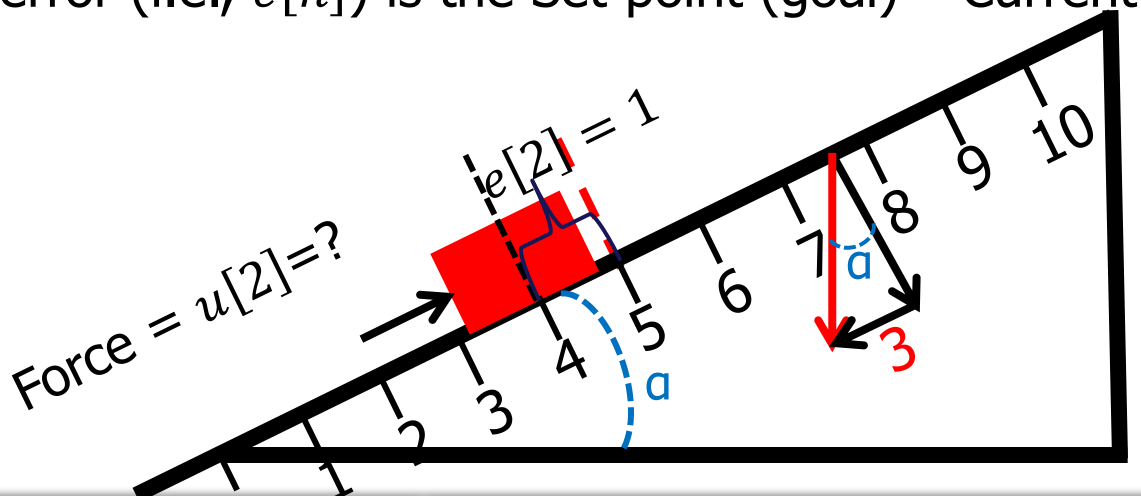
n=1: $e[1]$=2, $esum[1] = 5$, $(e[1] - e[0]) = -1$, $u[1]$=3.5;

n=2: $e[2]$=?, $esum[2]$ =?, $(e[2] - e[1])$=?, $u[2]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{3}{u[2]} \underset{=}{=} \overset{1}{K_P}\underbrace{\overset{1}{e[2]}}_{1} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\overset{1}{e[2]}}_{.5*6\,=\,3} + \overset{1}{K_D}\underbrace{(\overset{1}{e[2]}-\overset{2}{e[1]})}_{-1}$$

pre_sum=5

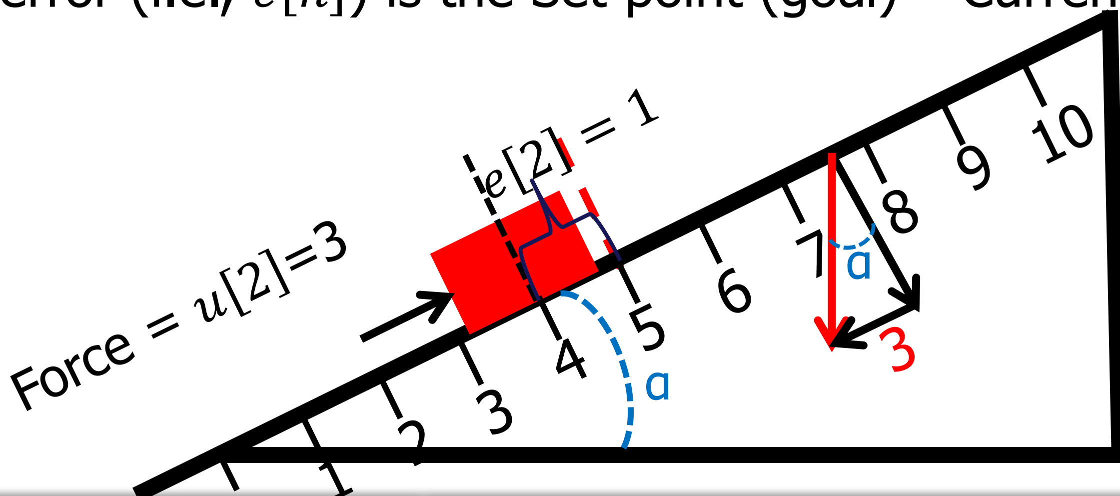
n=1: $e[1]$=2, $esum[1] = 5$, $(e[1] - e[0]) = -1$, $u[1]$=3.5;

n=2: $e[2]$=1, $esum[2] = 6$, $(e[2] - e[1]) = -1$, $u[2]$=3;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\overset{}{u[2]} = \overset{1}{K_P}\underbrace{\overset{1}{e[2]}}_{1} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\overset{1}{e[2]}}_{\text{pre\_sum=5}\quad .5*6\,=\,3} + \overset{1}{K_D}\underbrace{(\overset{1}{e[2]} - \overset{2}{e[1]})}_{-1}$$

$$3 \quad =$$

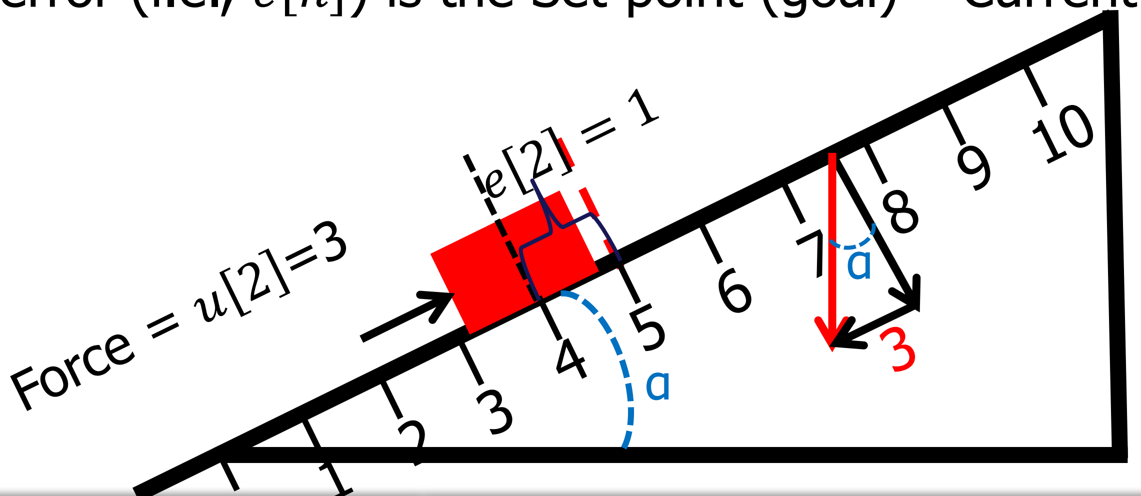
n=2: $e[2]$=1, $esum[2] = 6$, $(e[2] - e[1]) = -1$, $u[2]$=3;

n=3: $e[3]$=?, $esum[3]$ =?, $(e[3] - e[2])$=? $u[3]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a <u>ramp</u> with <u>NO</u> friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{2}{u[3]} \underset{=}{=} \overset{1}{K_P}\underbrace{\overset{0}{e[3]}}_{0} + \underbrace{\overset{.5}{K_I} \sum_{j=0}^{n} \overset{0}{e[3]}}_{\substack{\text{pre\_sum}=6 \\ .5*6\ =\ 3}} + \overset{1}{K_D}\underbrace{(e[3] - \overset{0}{e[2]}\overset{1}{)}}_{-1}$$

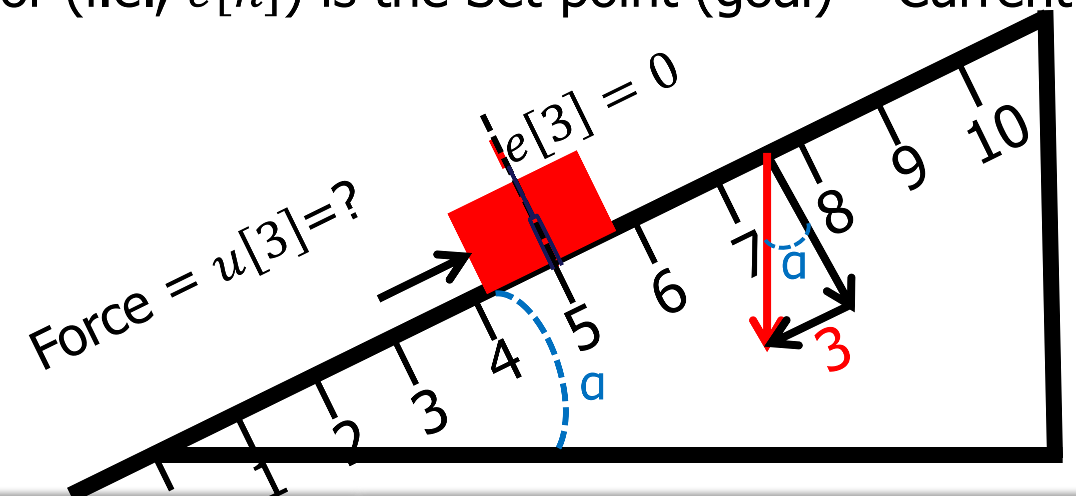
n=2: $e[2]$=1, $esum[2] = 6$, $(e[2] - e[1]) = -1$, $u[2]$=3;

n=3: $e[3]$=?, $esum[3] =?$, $(e[3] - e[2])$=? $u[3]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{2\ =}{u[3]} = \overset{1}{K_P}\underbrace{\overset{0}{e[3]}}_{0} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\overset{0}{e[3]}}_{\substack{\text{pre\_sum=6} \\ .5*6\ =\ 3}} + \overset{1}{K_D}\underbrace{(\overset{0}{e[3]} - \overset{1}{e[2]})}_{-1}$$

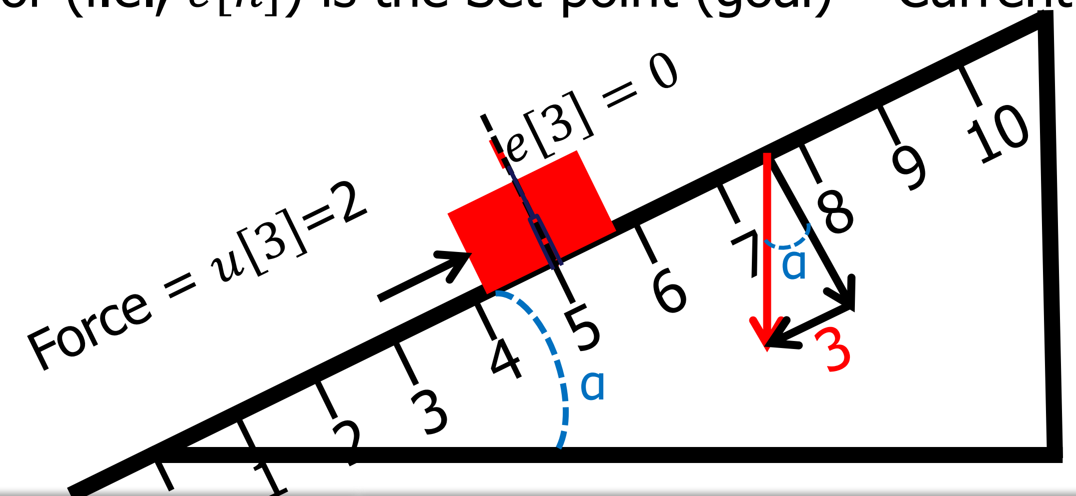
n=2: $e[2]$=1, $esum[2] = 6$, $(e[2] \ - e[1]) = -1$, $u[2]$=3;

n=3: $e[3]$=0, $esum[3] = 6$, $(e[3] \ - e[2]) = -1$ $u[3]$=2;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{2}{u[3]} \underset{=}{=} \overset{1}{K_P}\underbrace{\overset{0}{e[3]}}_{0} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\overset{0}{e[3]}}_{\text{pre\_sum}=6,\ .5*6\,=\,3} + \overset{1}{K_D}\underbrace{(\overset{0}{e[3]} - \overset{1}{e[2]})}_{-1}$$

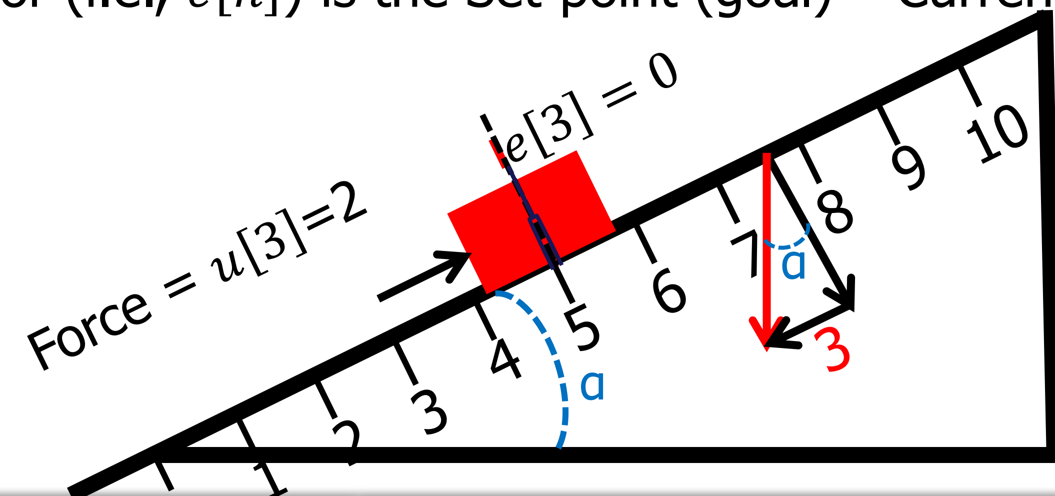
n=3: $e[3]$=0, $esum[3] = 6$, $(e[3] \; - e[2]) = -1$ $\quad u[3]$=2;

n=4: $e[4]$=?, $esum[4]$ =?, $(e[4] \; - e[3])$=? $\quad u[3]$=?;

Goal: Have the red block move from location 0 to location 5

Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



PID control: PID Control

$$\underset{2}{u[4]} \underset{=}{=} \overset{1}{K_P}\underbrace{\overset{0}{e[4]}}_{0} + \underbrace{\overset{.5}{K_I}\sum_{j=0}^{n}\overset{0}{e[4]}}_{\text{pre\_sum}=6;\quad .5*6\,=\,3} + \overset{1}{K_D}\underbrace{(\overset{0}{e[4]}-\overset{1}{e[3]})}_{-1}$$

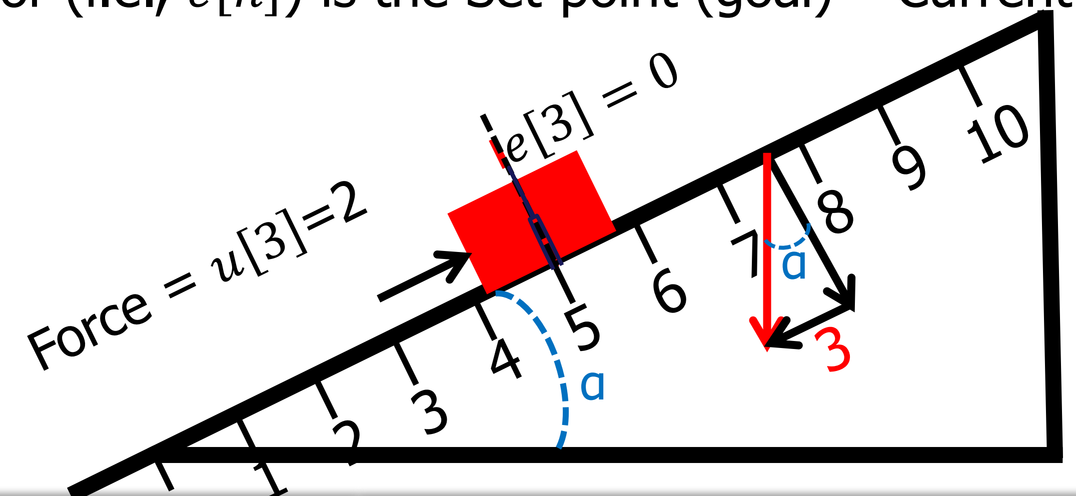
n=3: $e[3]$=0, $esum[3] = 6$, $(e[3] \ - e[2]) = -1$ $u[3]$=2;

n=4: $e[4]$=?, $esum[4]$ =?, $(e[4] \ - e[3])$=? $u[3]$=?;

Goal: Have the red block move from location 0 to location 5

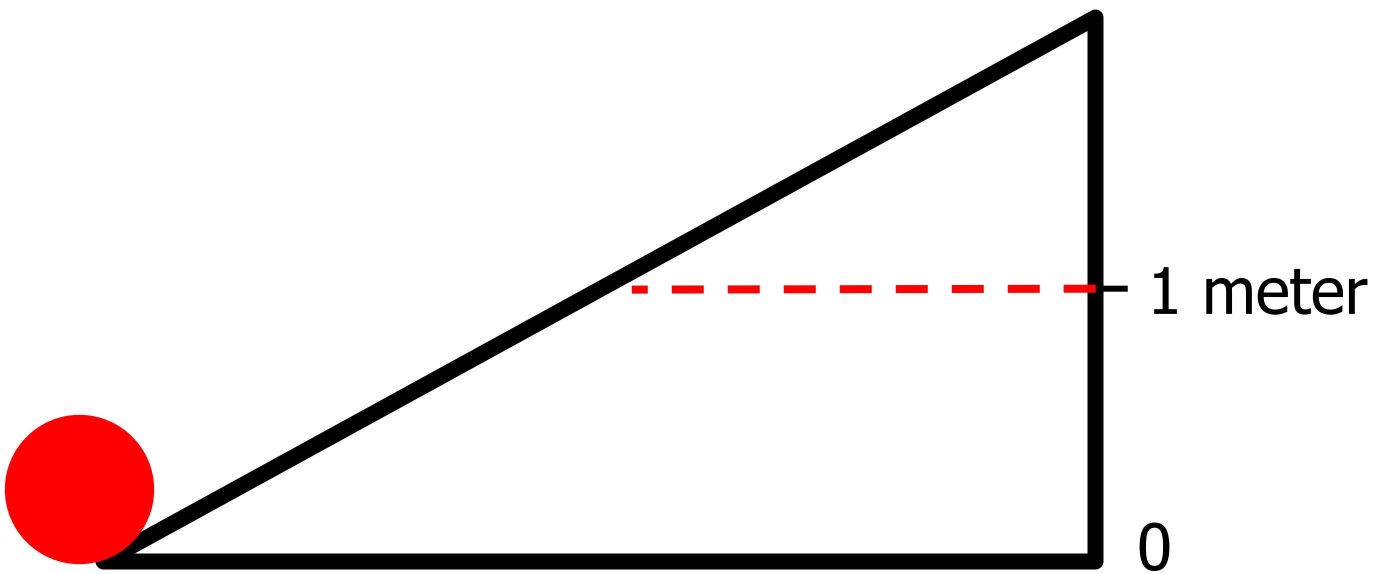
Set up:

- Red block is on a ramp with NO friction
- Let the output of the controller (i.e., $u[n]$) be force applied to the block
- The current error (i.e., $e[n]$) is the Set-point (goal) – Current Location



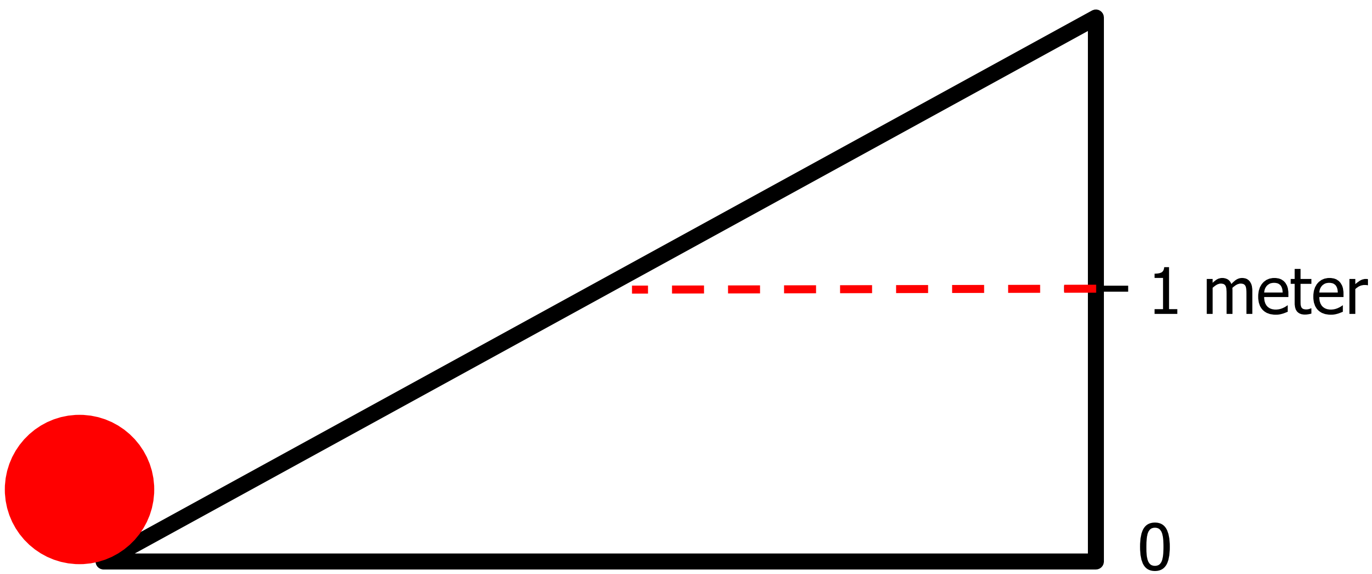
PID Plot Analysis

- Practice intuition for PID tuning



PID Plot Analysis

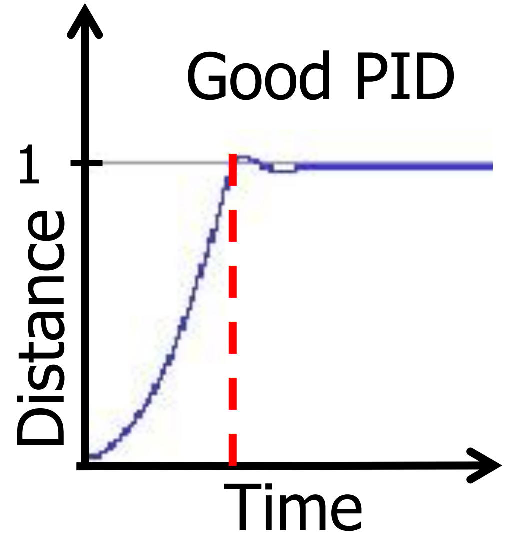
- Practice intuition for PID tuning

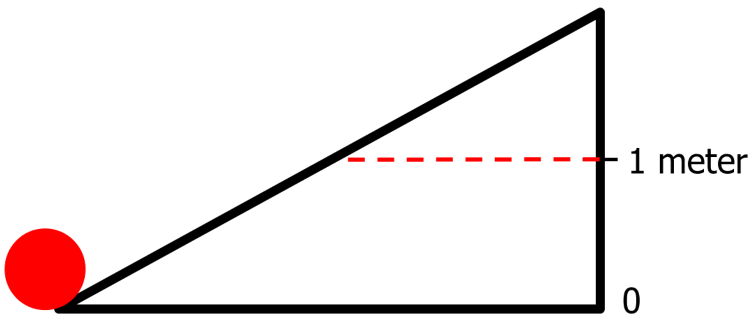


PID Plot Analysis (cont.)

https://sites.google.com/site/fpgaandco/pid-demo

- Car max angle
- P = 30, I=1, D=2.2
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

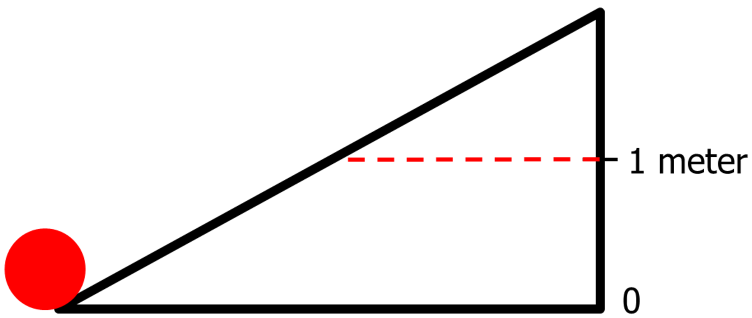


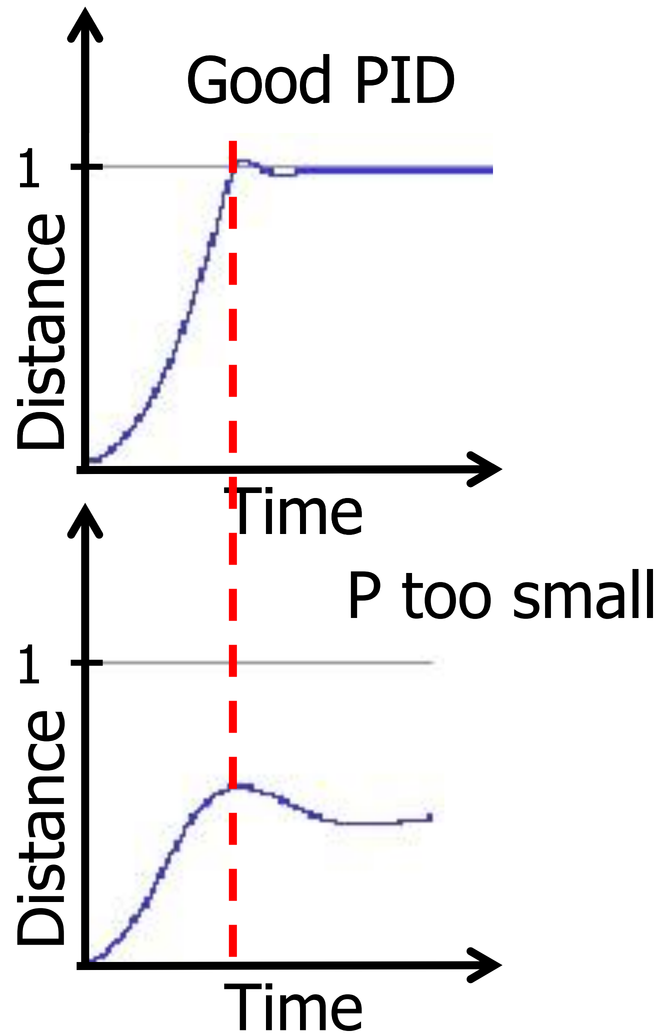


PID Plot Analysis (cont.)

https://sites.google.com/site/fpgaandco/pid-demo

- Car max angle
- P = 30, I=1, D=2.2
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

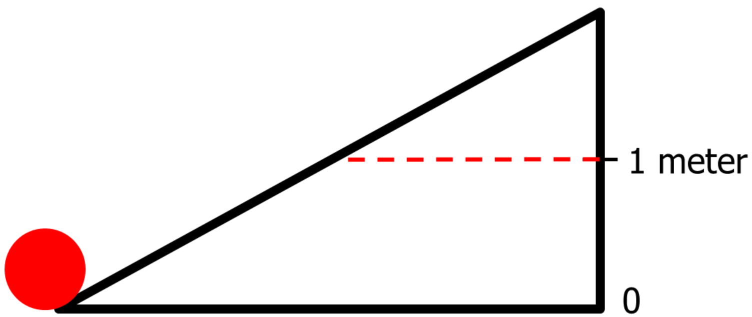


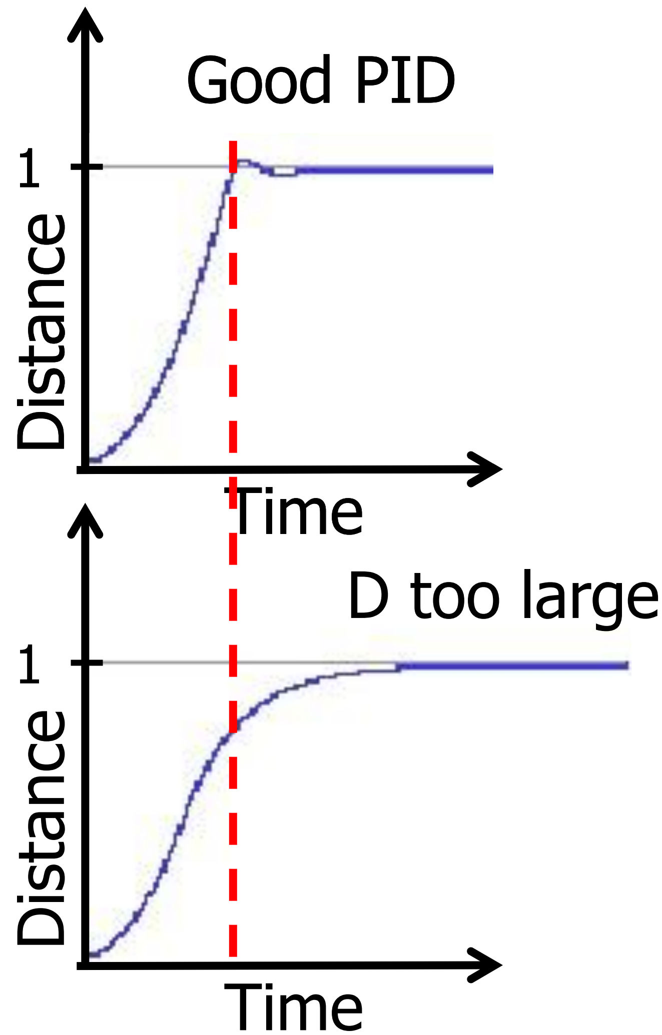


PID Plot Analysis (cont.)

https://sites.google.com/site/fpgaandco/pid-demo

- Car max angle
- P = 30, I=1, D=2.2
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

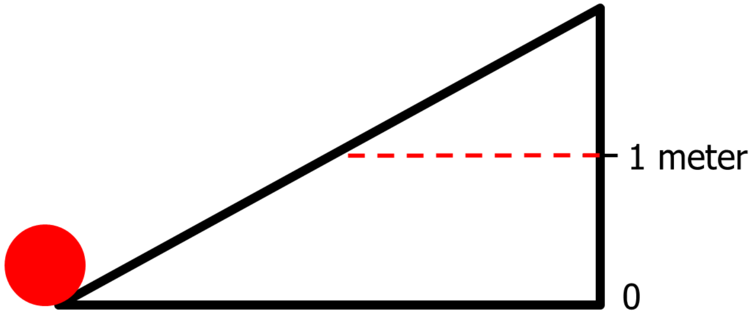


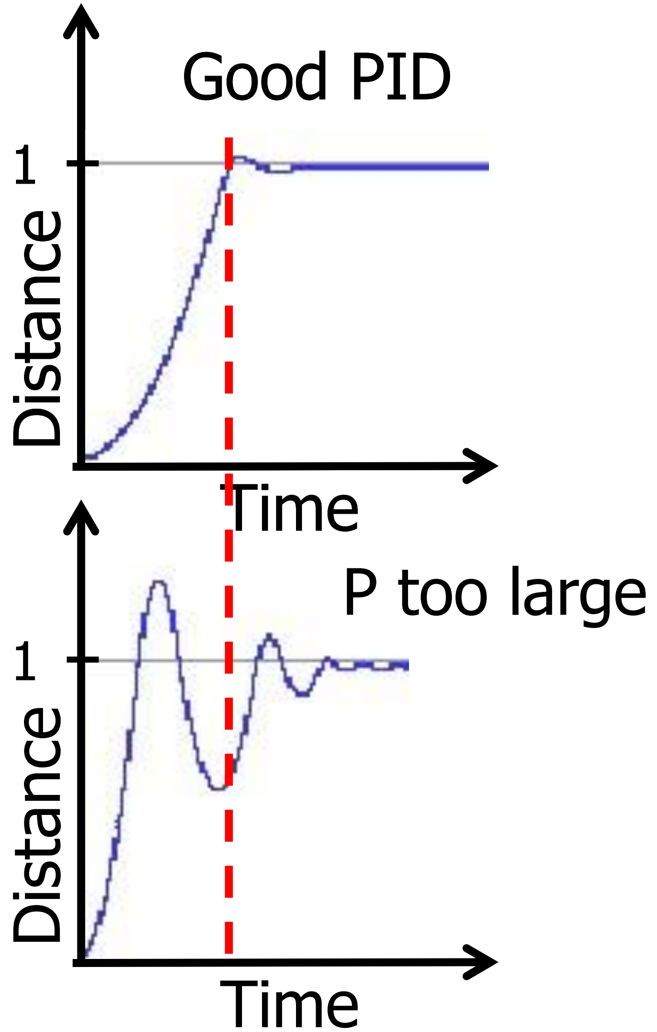


PID Plot Analysis (cont.)

https://sites.google.com/site/fpgaandco/pid-demo

- Car max angle
- P = 30, I=1, D=2.2
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

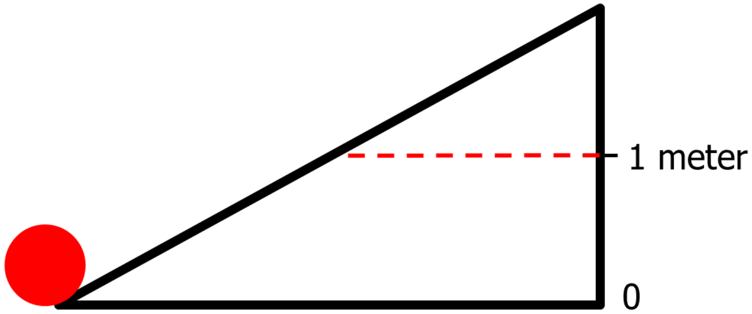


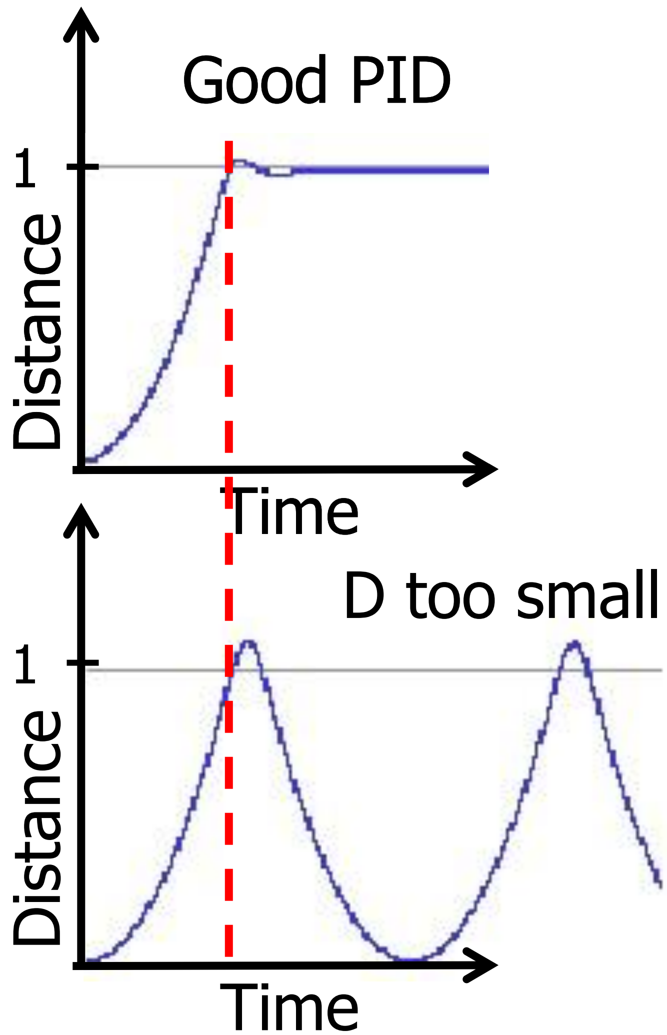


PID Plot Analysis (cont.)

https://sites.google.com/site/fpgaandco/pid-demo

- Car max angle
- P = 30, I=1, D=2.2
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

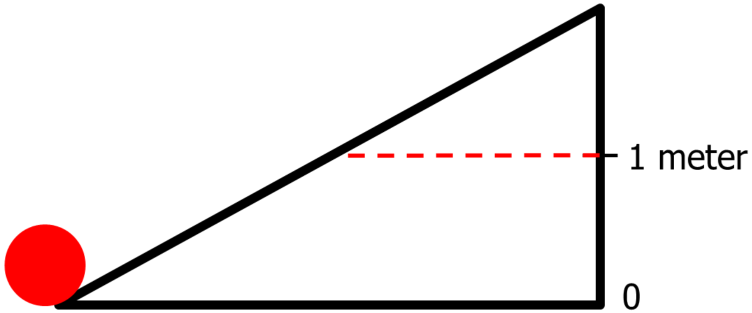


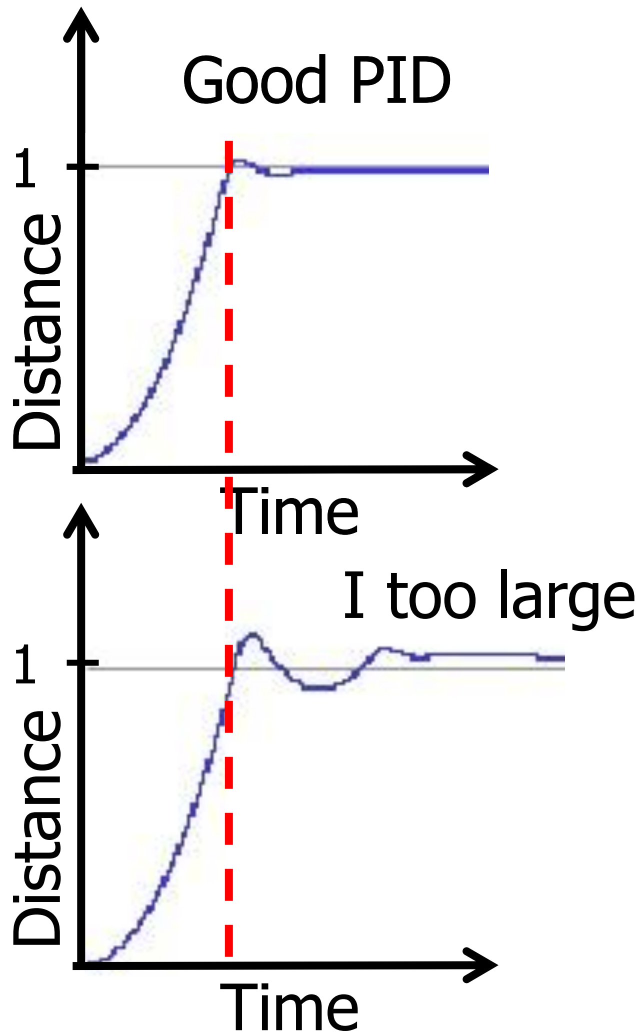


PID Plot Analysis (cont.)

https://sites.google.com/site/fpgaandco/pid-demo

- Car max angle
- P = 30, I=1, D=2.2
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

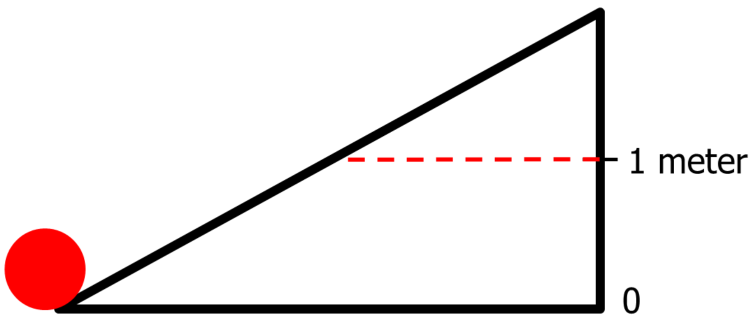




PID Plot Analysis (cont.)

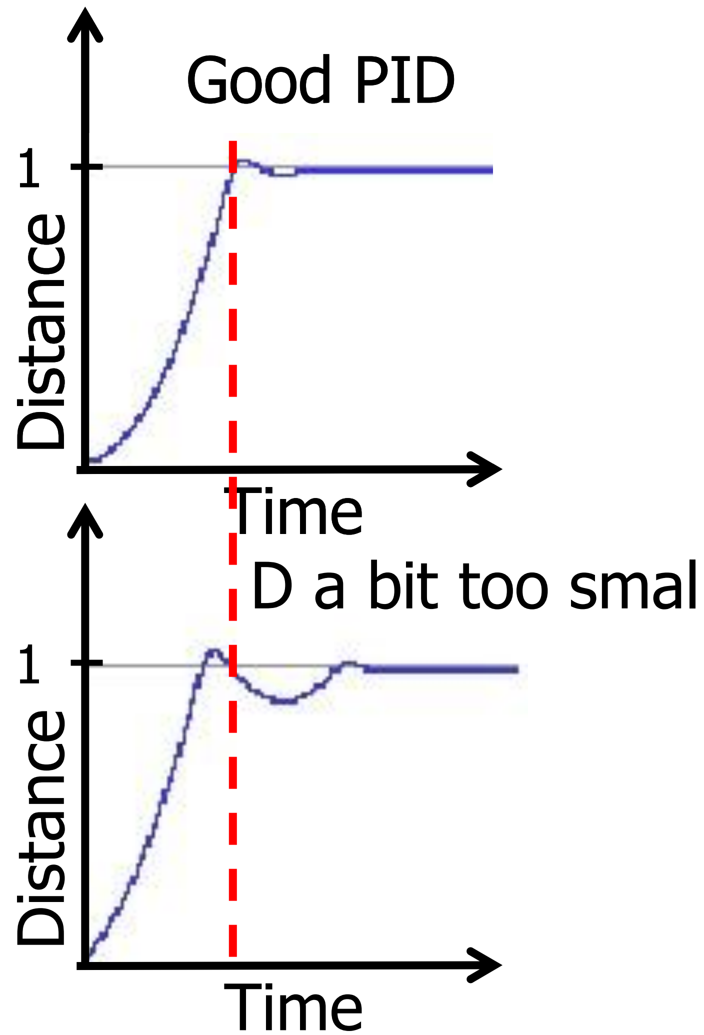
https://sites.google.com/site/fpgaandco/pid-demo

- Car max angle
- P = 30, I=1, D=2.2
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

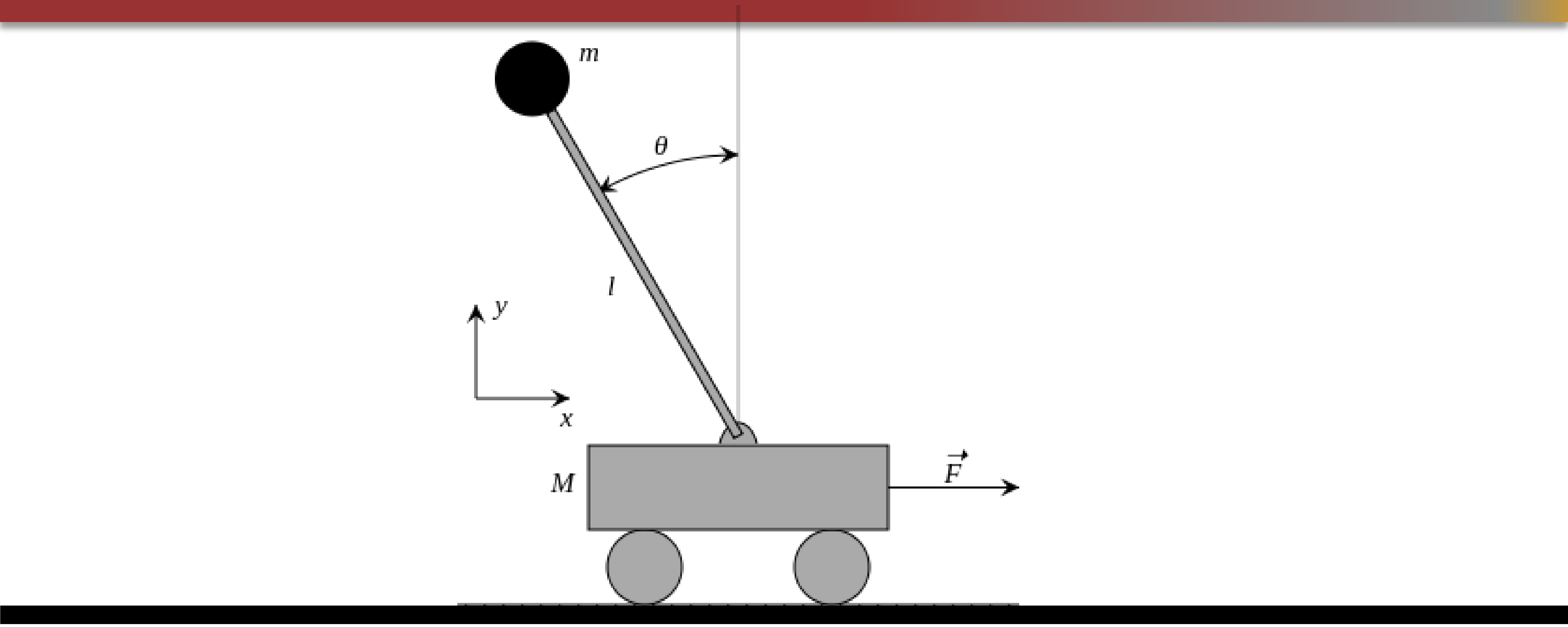


https://sites.google.com/site/fpgaandco/pid-demo

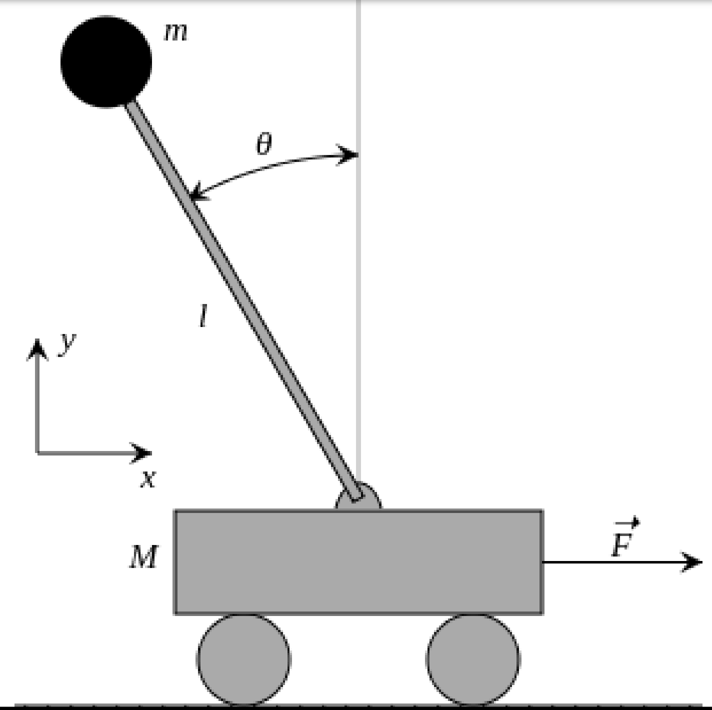
- Car max angle
- P = 30, I=1, D=1.5
- M = .2 Kg, Damping force = 0, Motor force limit 1 N

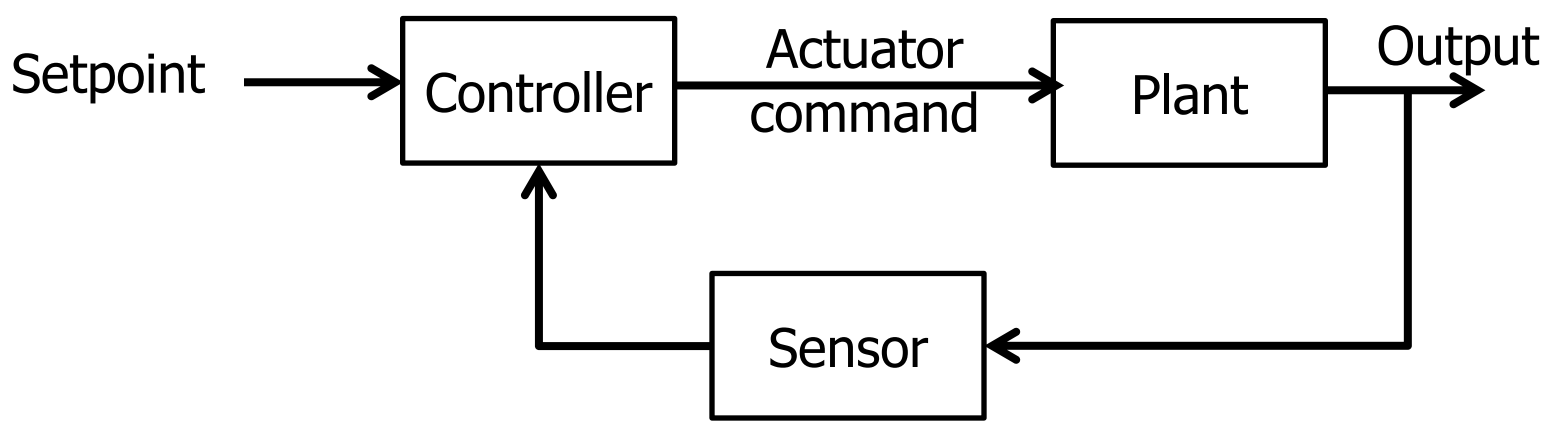


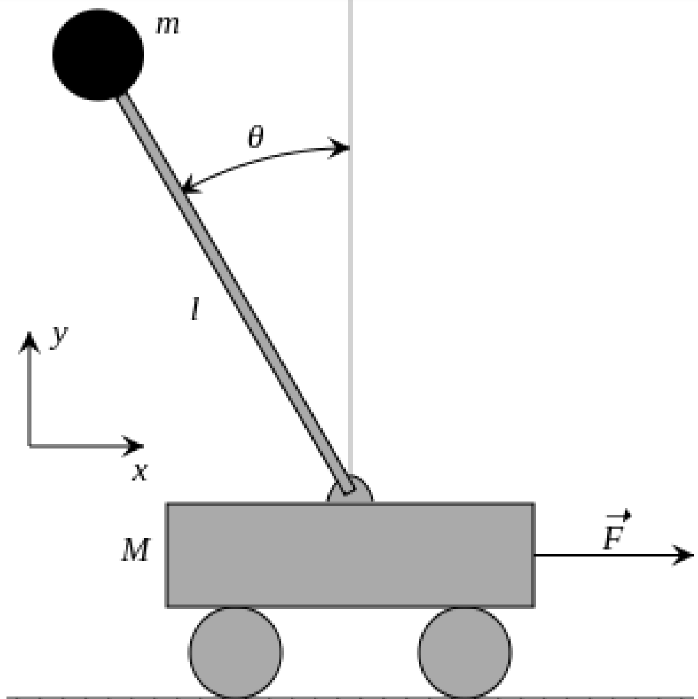
Inverted Pendulum

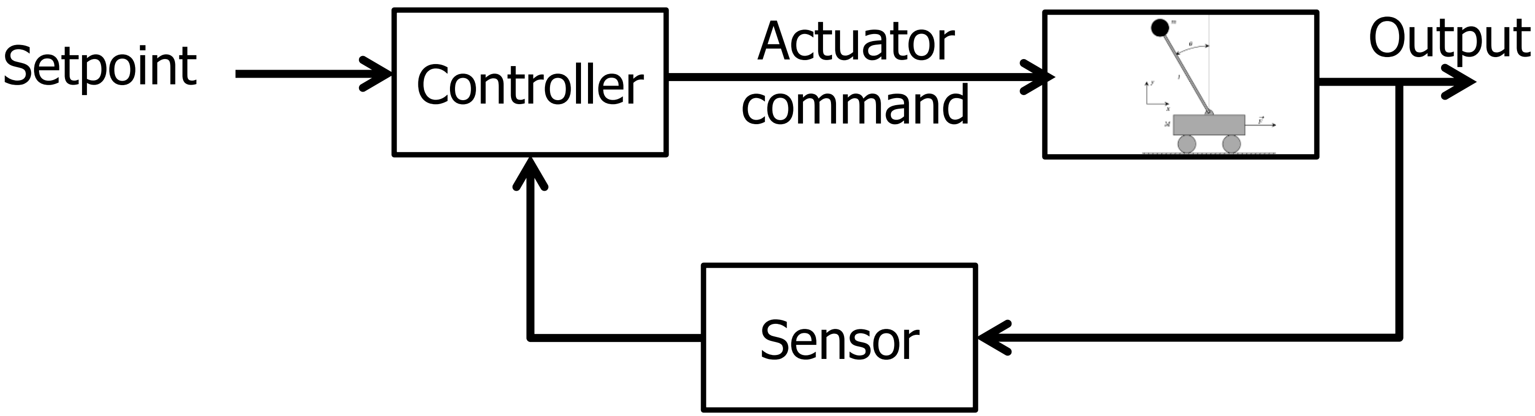


Inverted Pendulum (cont.)

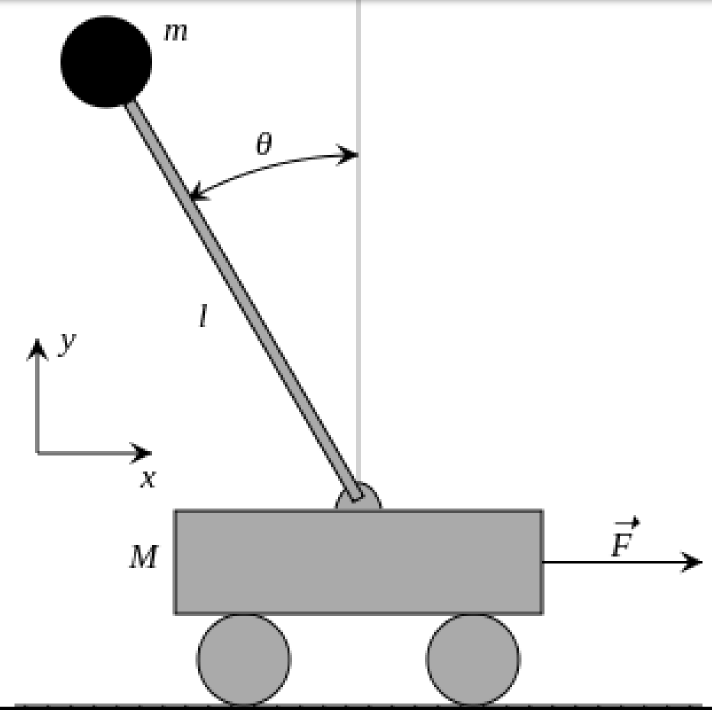


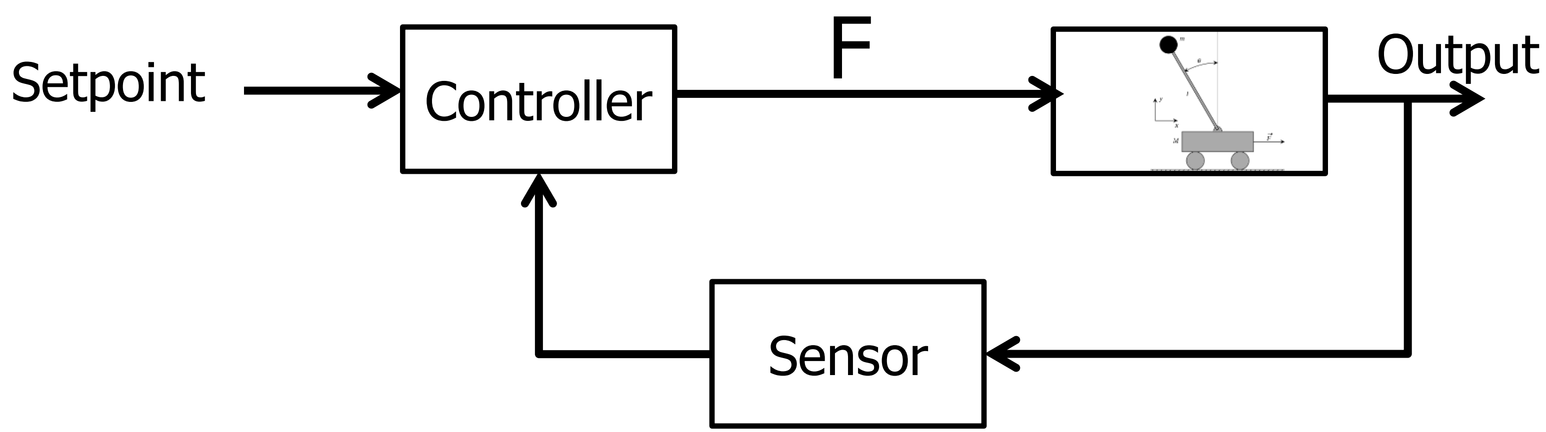



m
θ
l
y
x
M
F

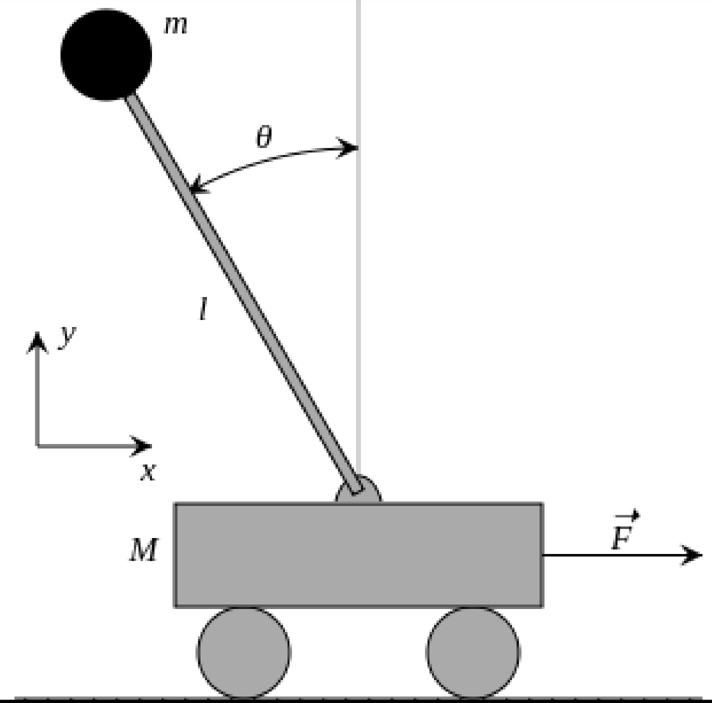

Setpoint
Controller
Actuator command
Output
Sensor

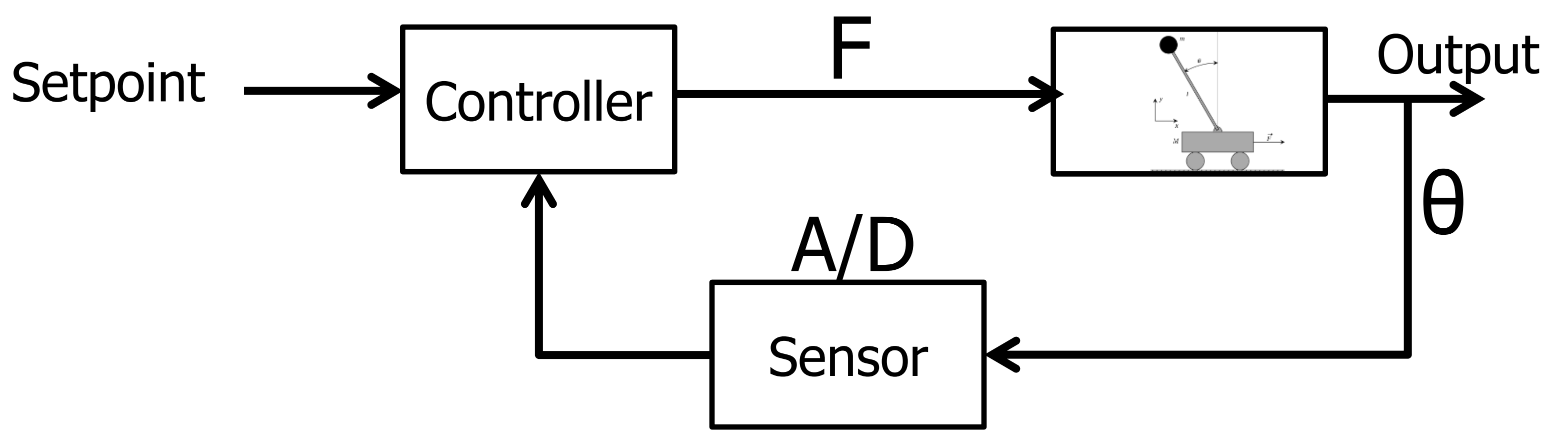
Inverted Pendulum (cont.)





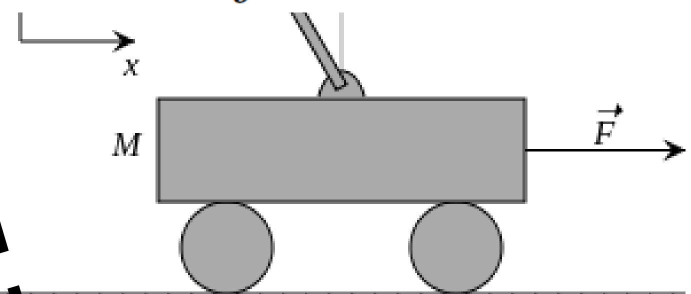
Inverted Pendulum (cont.)

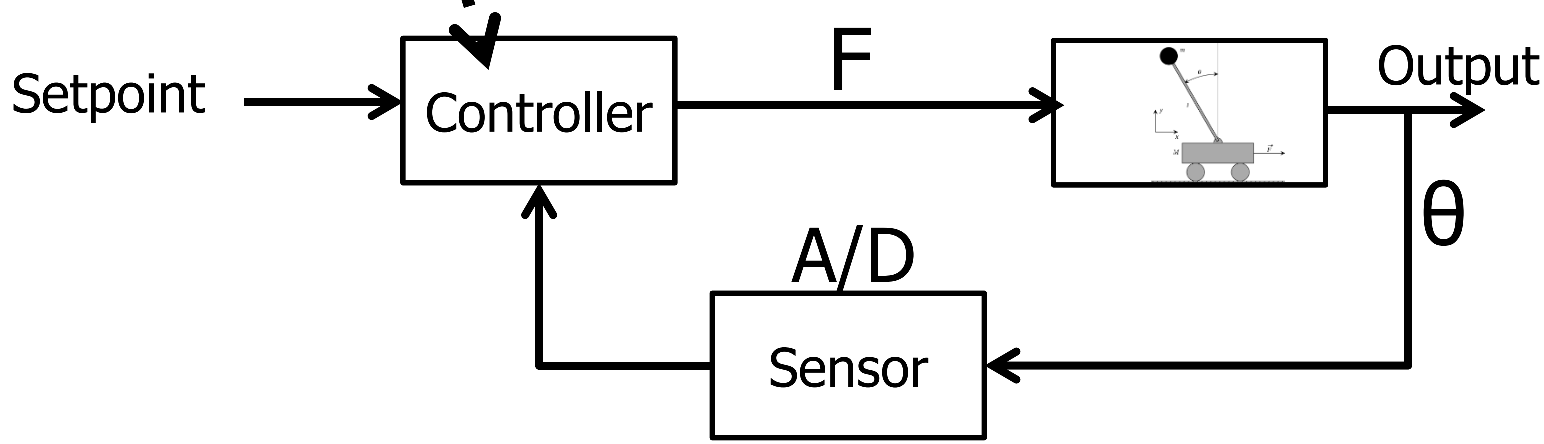




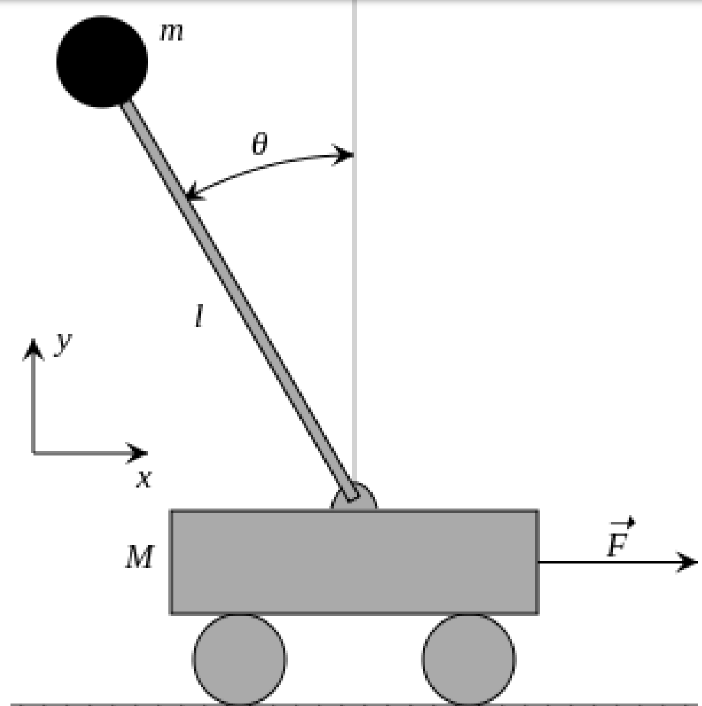
Inverted Pendulum (cont.)

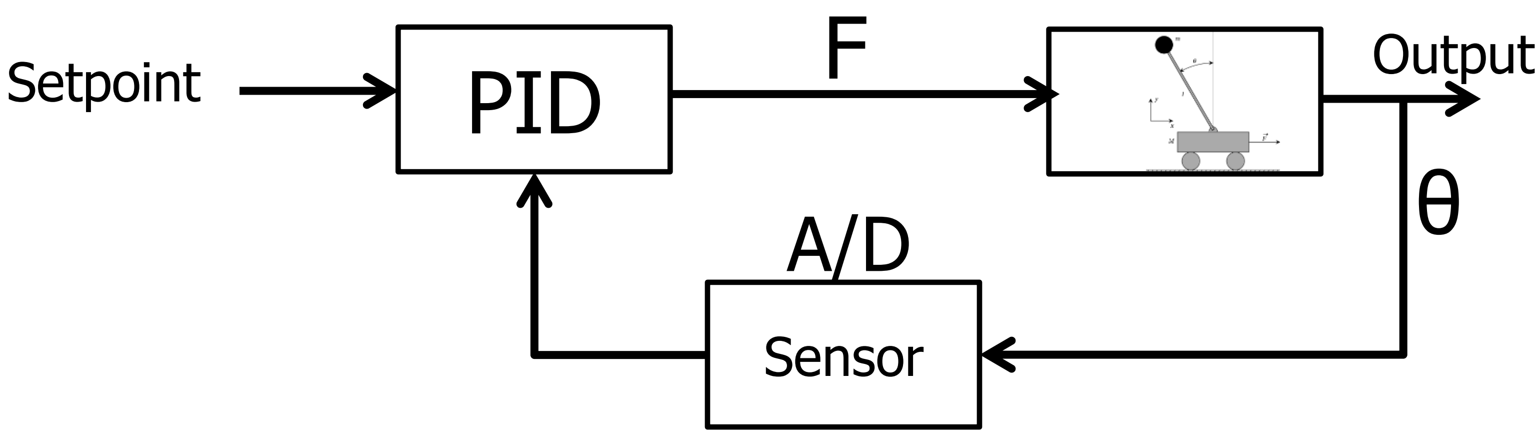
$$u[n] = K_P e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$



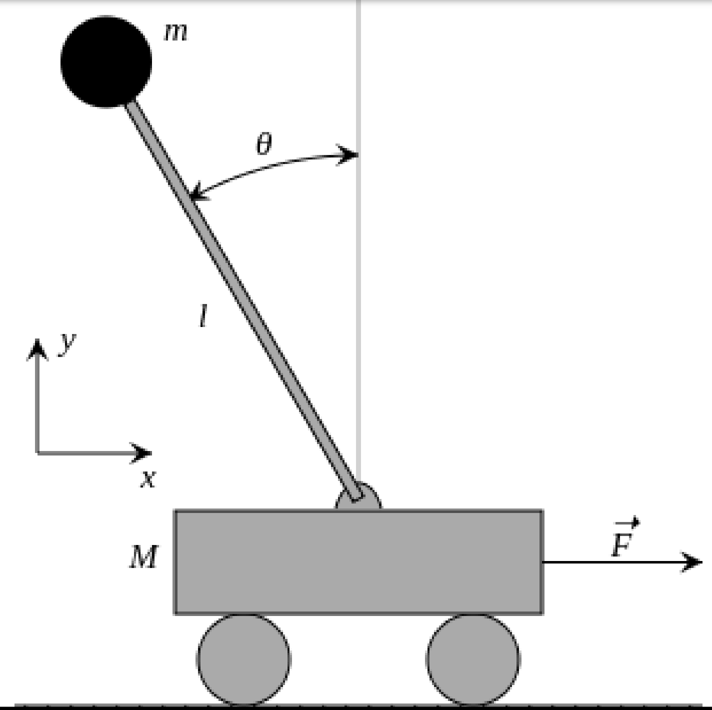


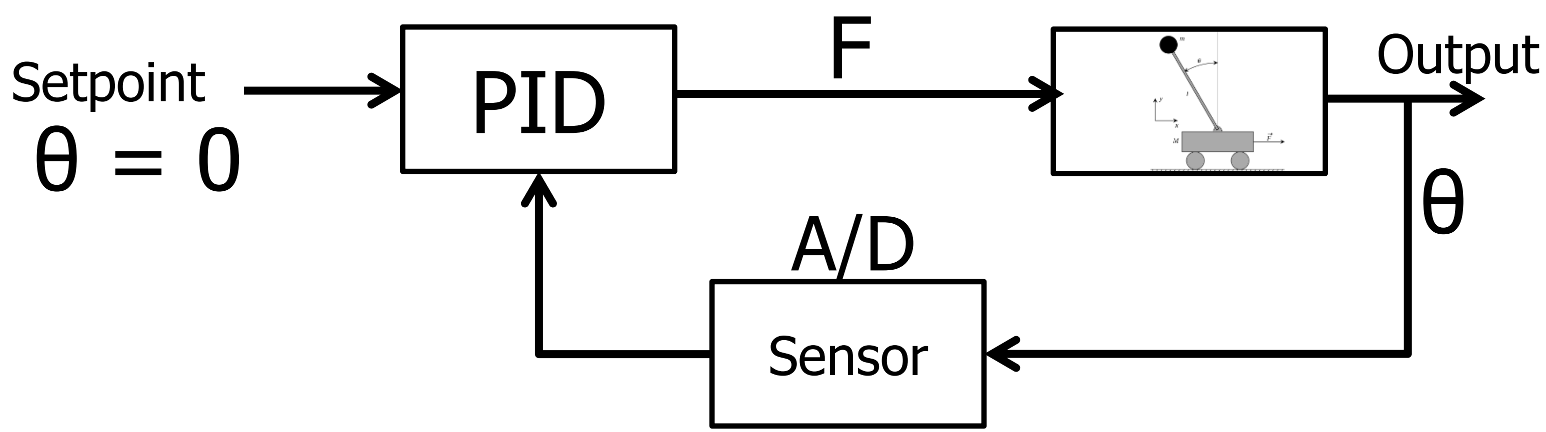
Inverted Pendulum (cont.)



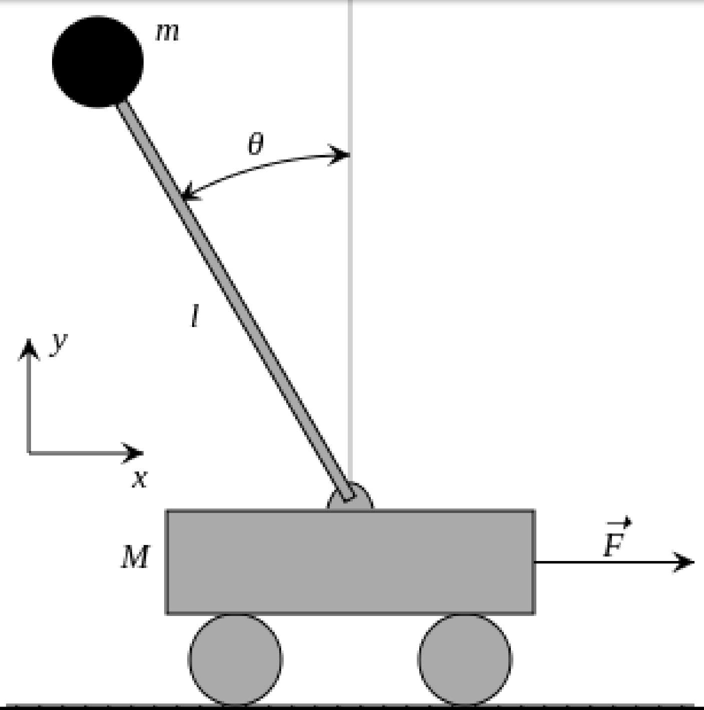


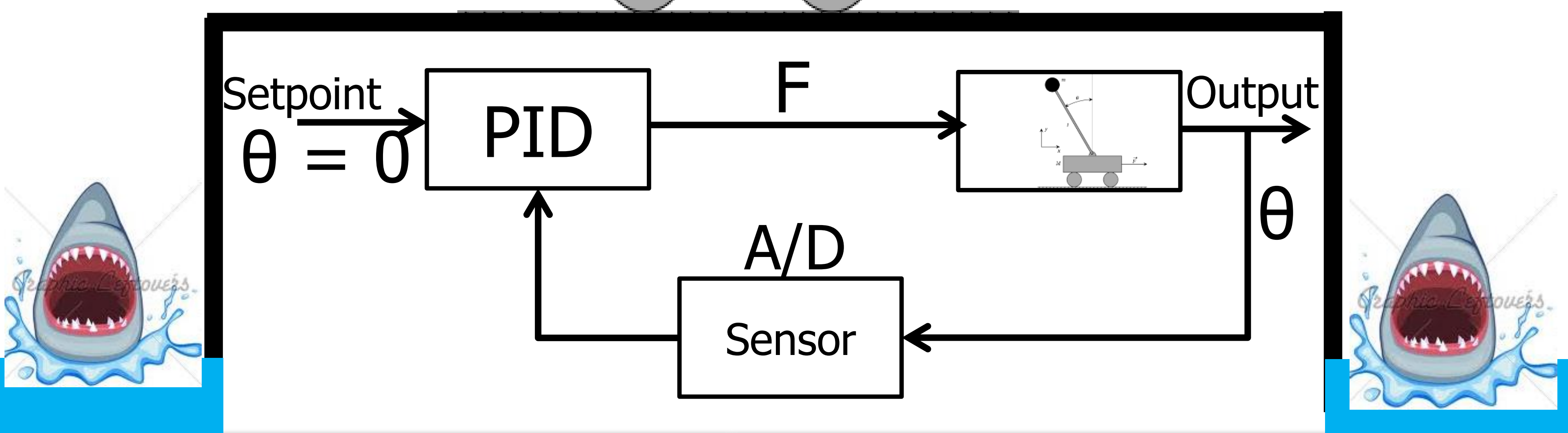
Inverted Pendulum (cont.)



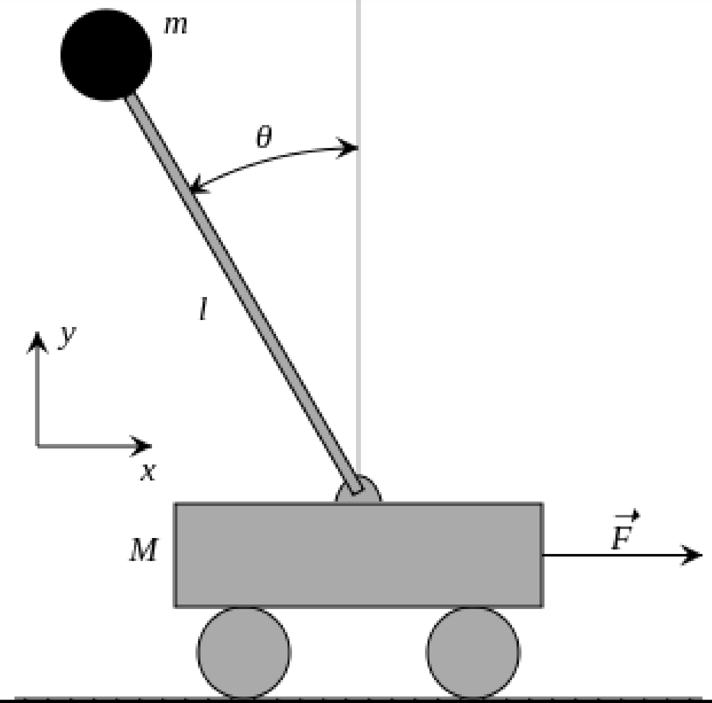


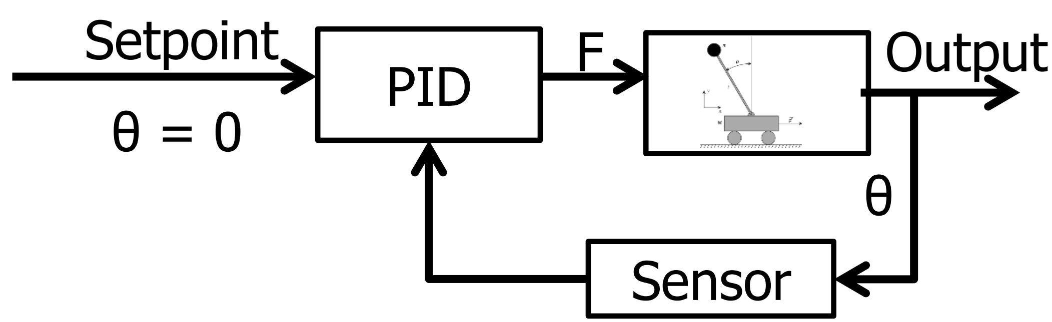
Inverted Pendulum (cont.)



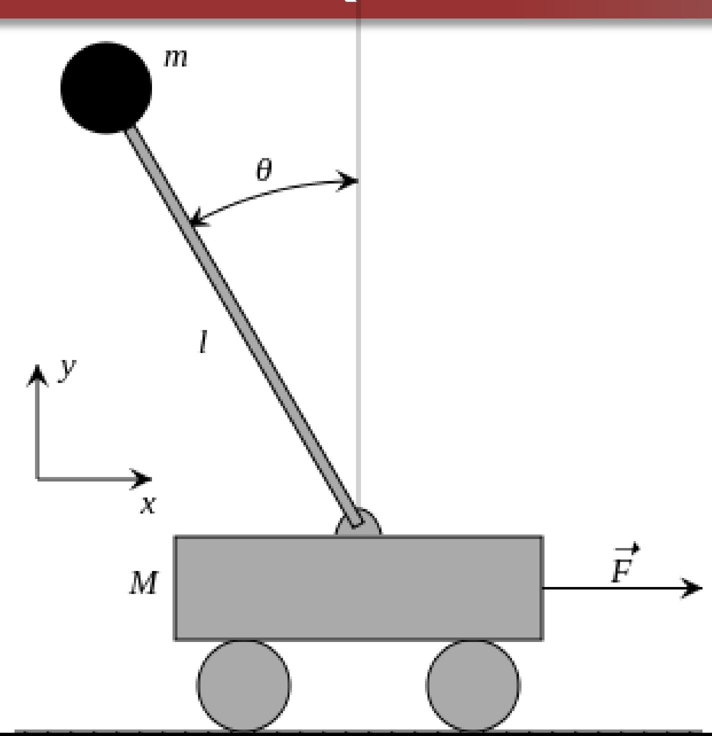


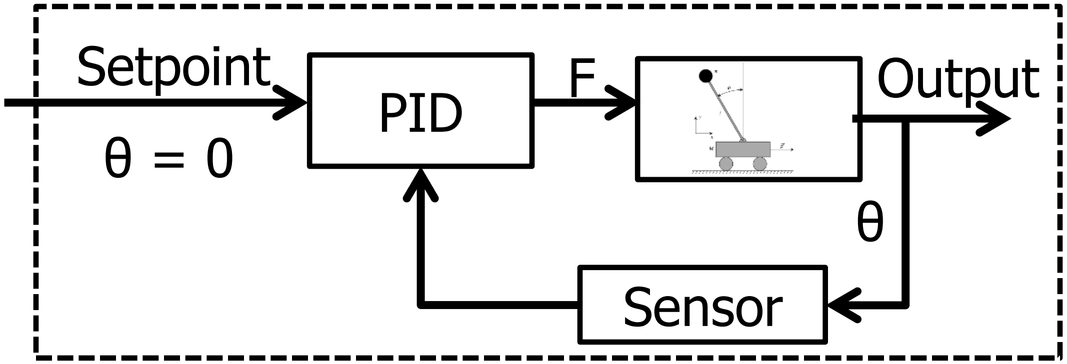
Inverted Pendulum (cont.)



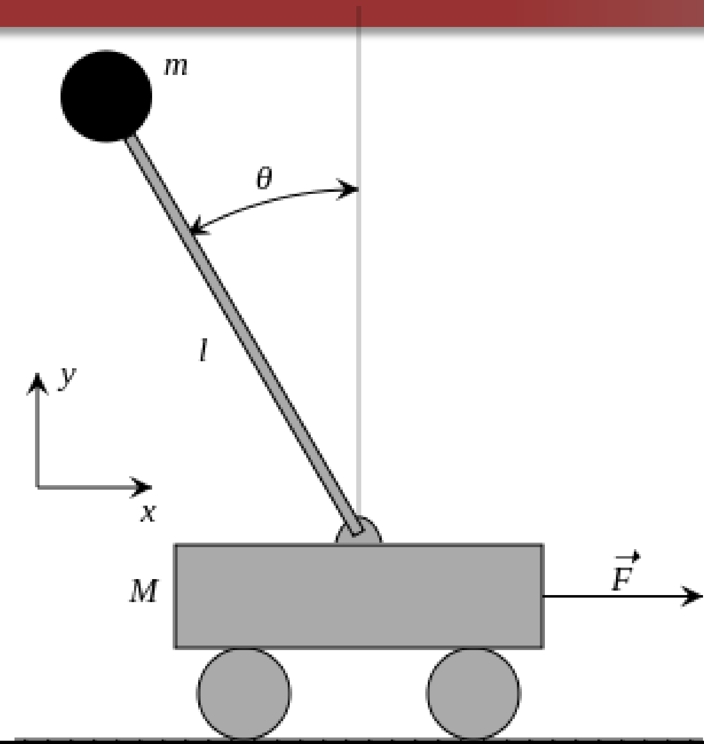


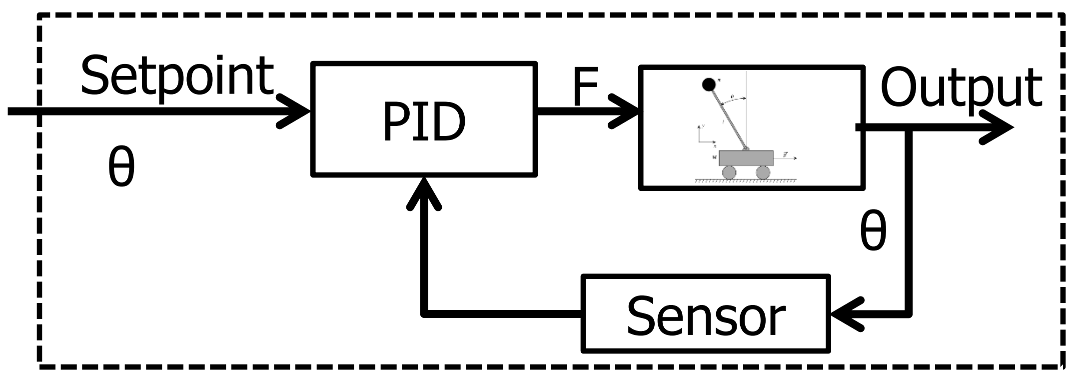
Inverted Pendulum (Nested PID)



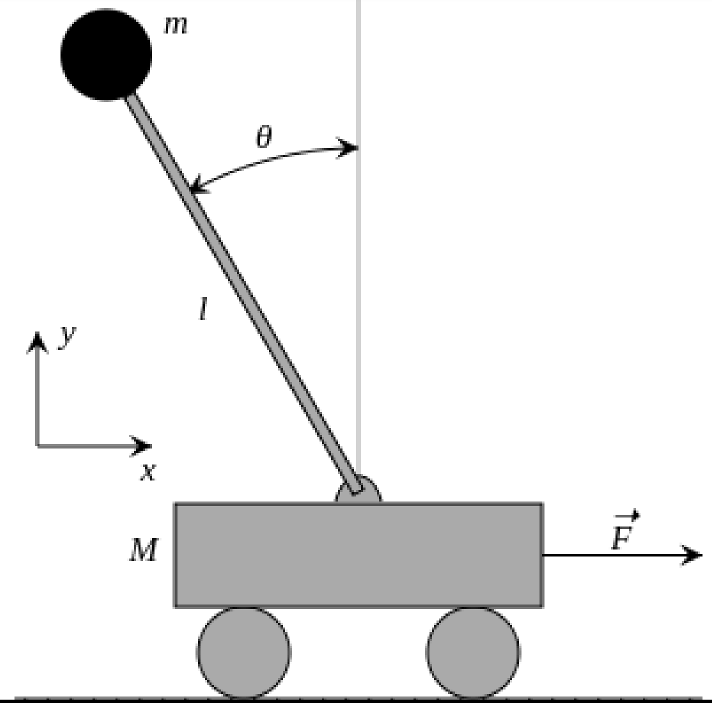


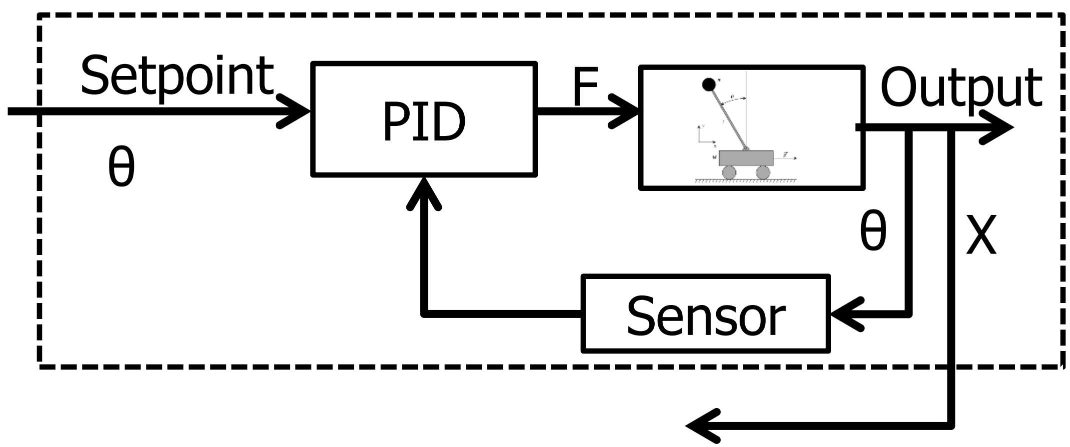
Nested PID



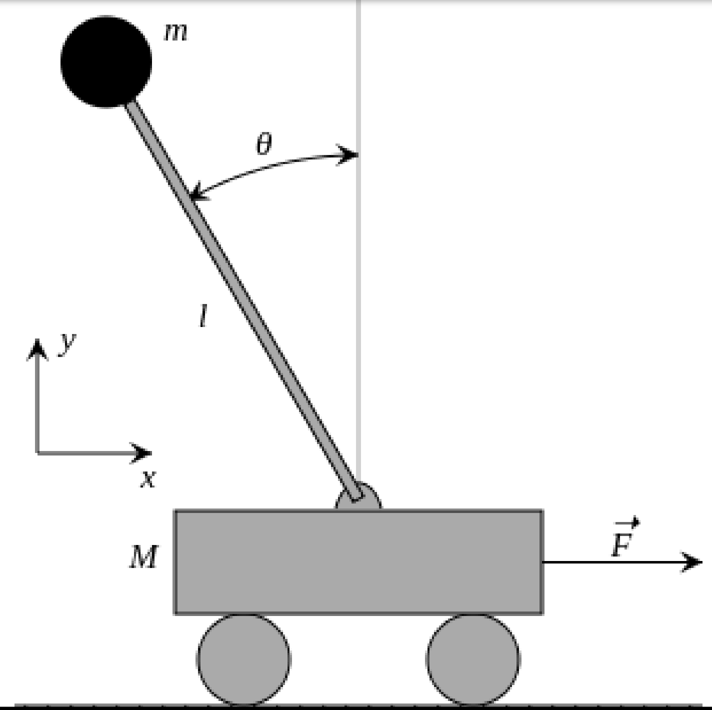


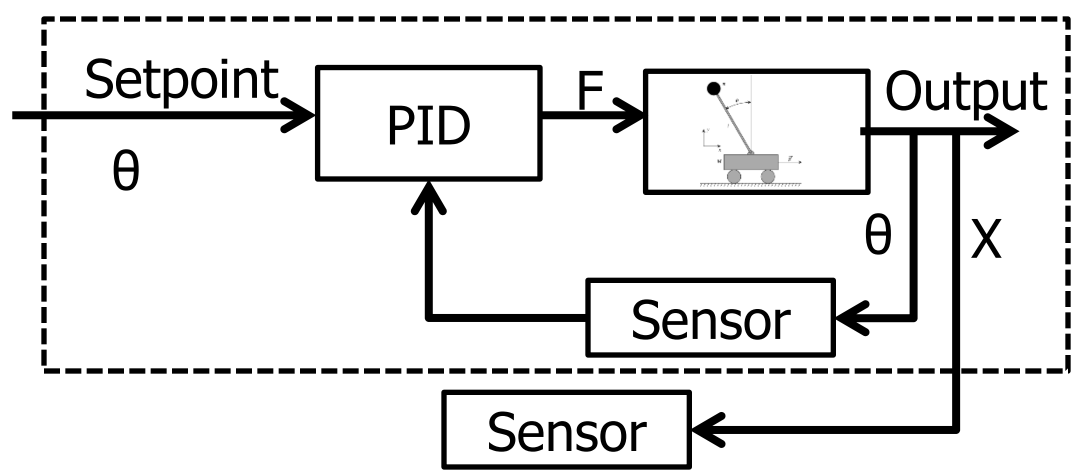
Nested PID (cont.)



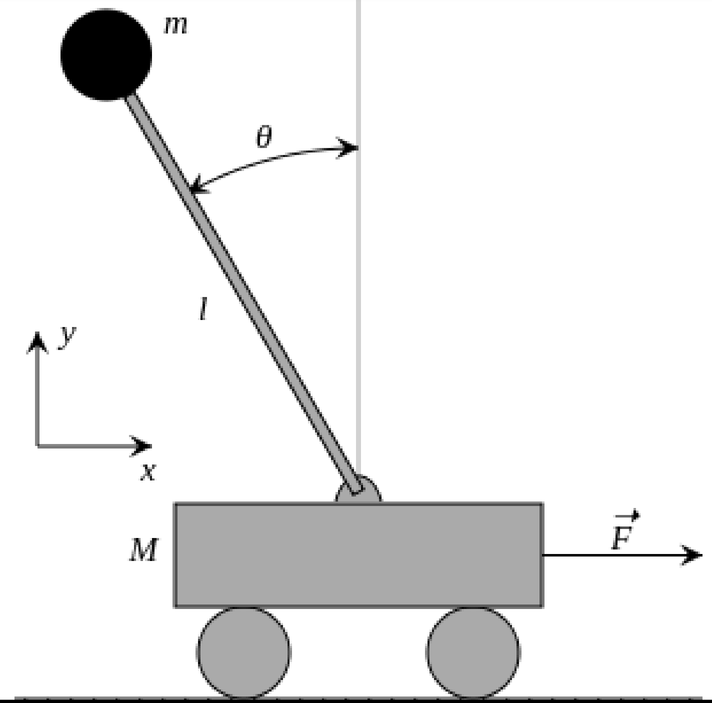


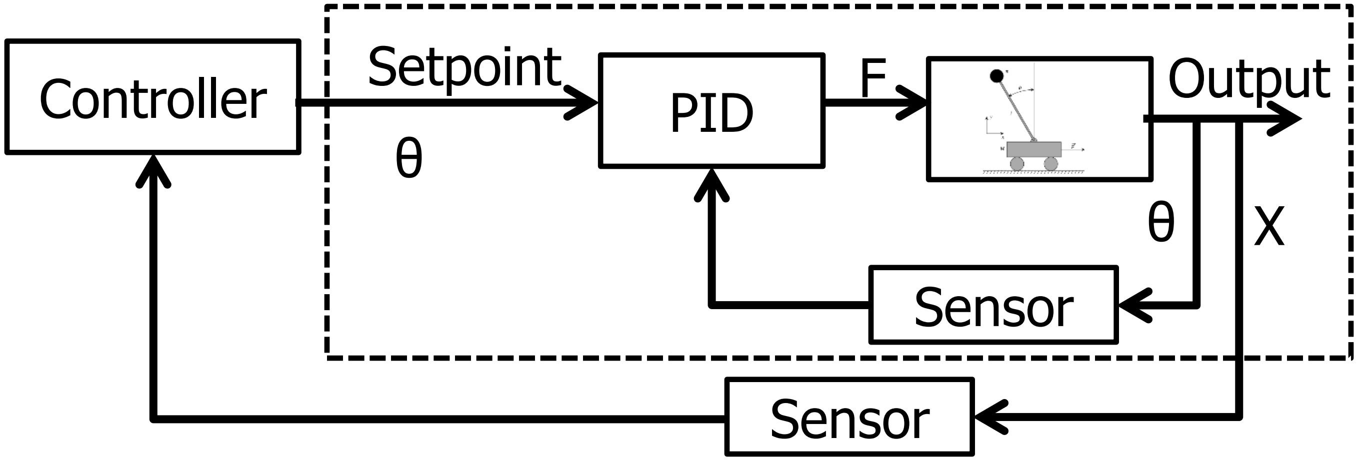
Nested PID (cont.)



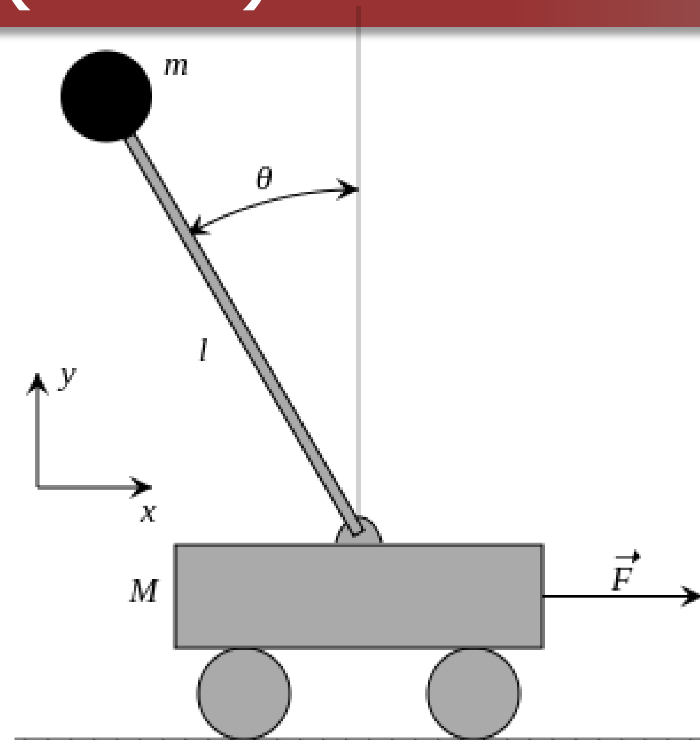


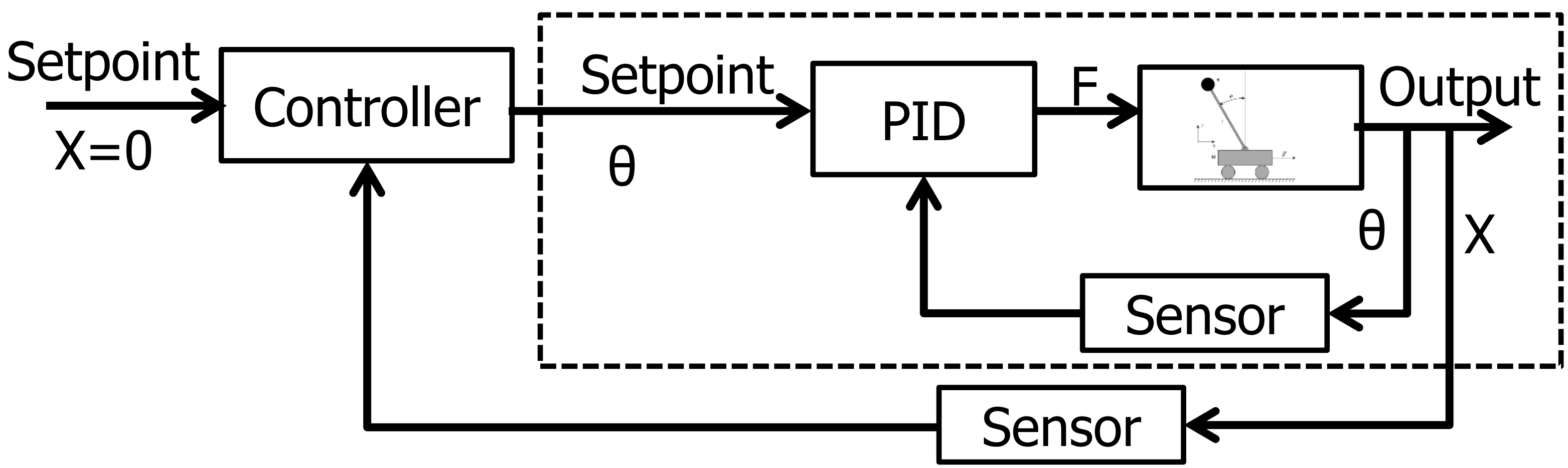
Nested PID (cont.)



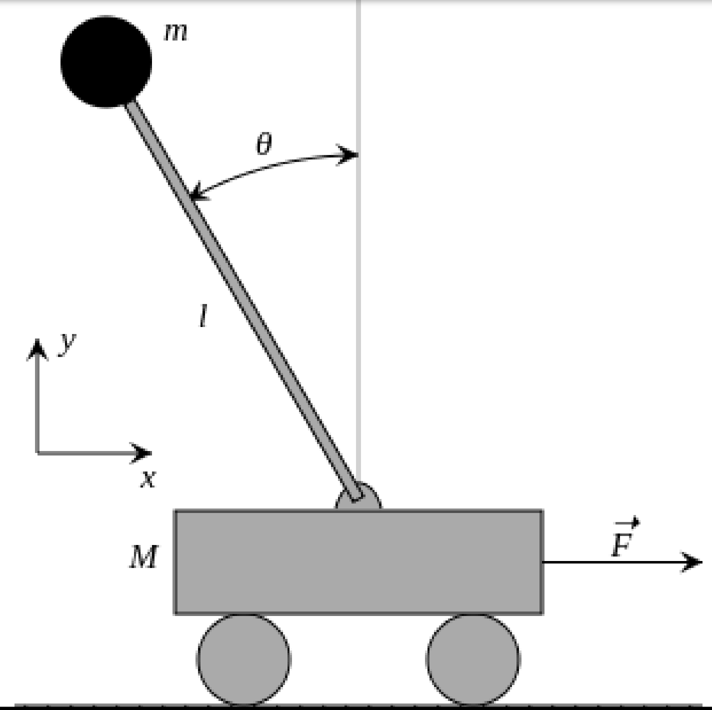


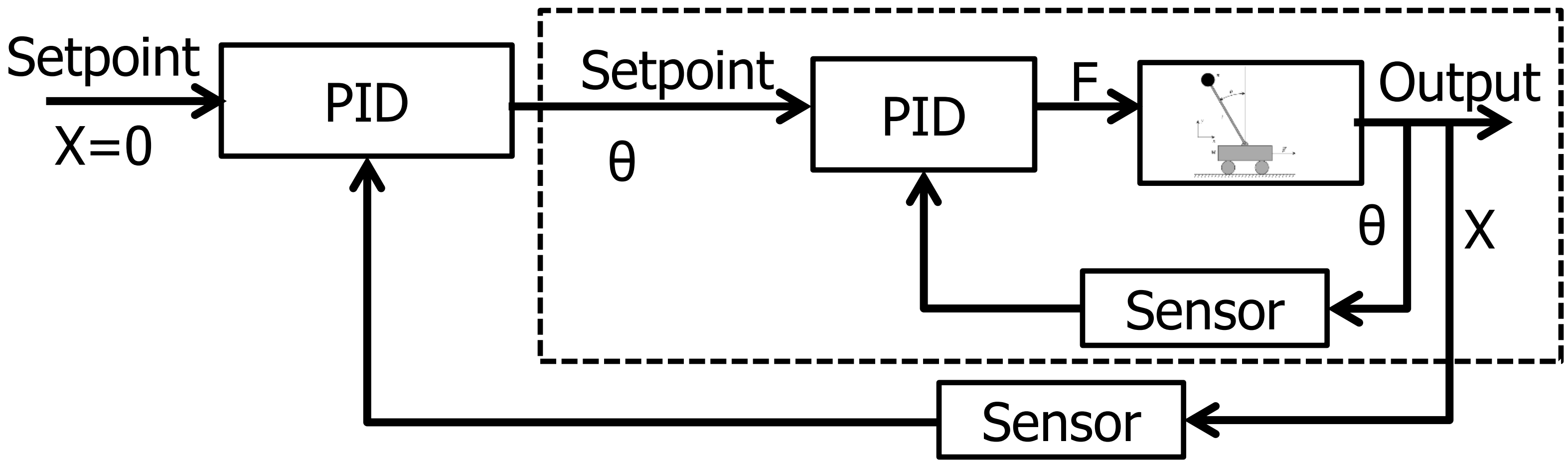
Nested PID (cont.)



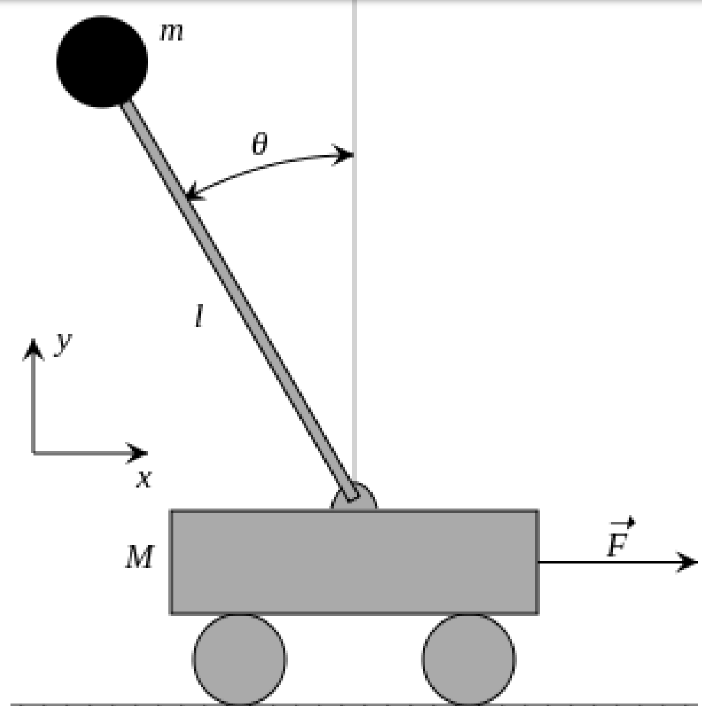


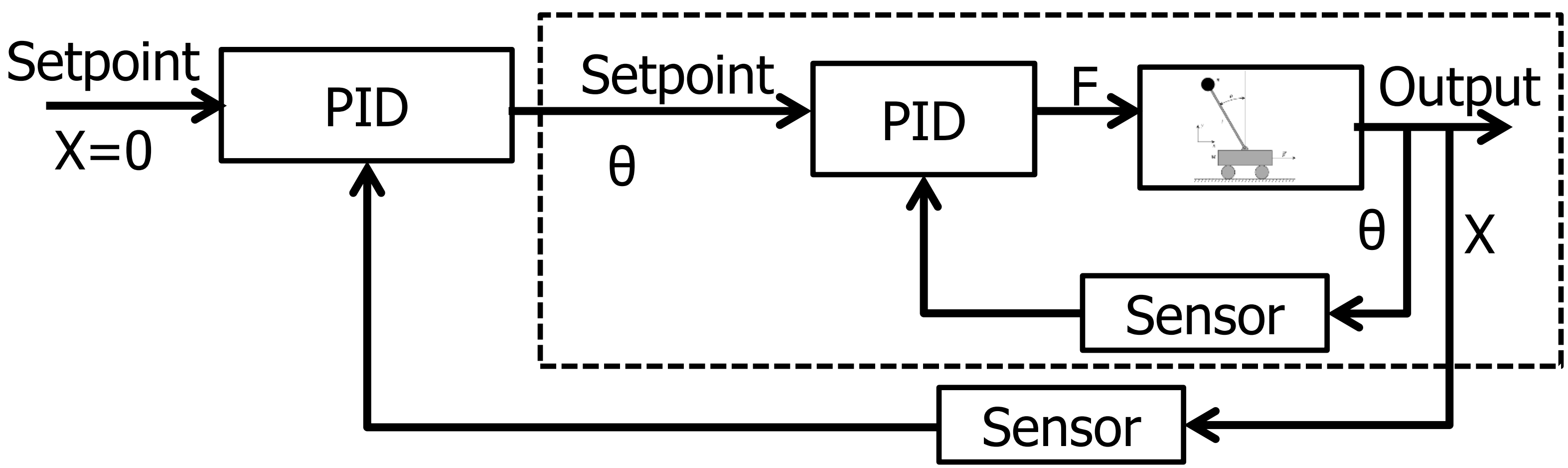
Nested PID (cont.)



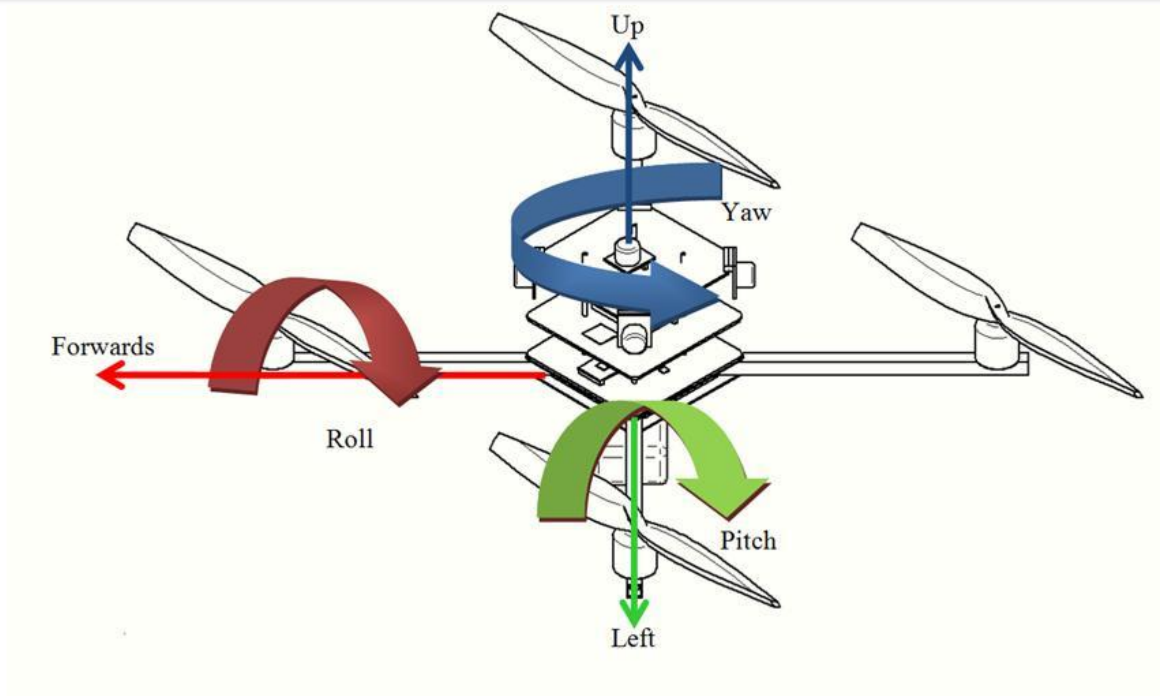


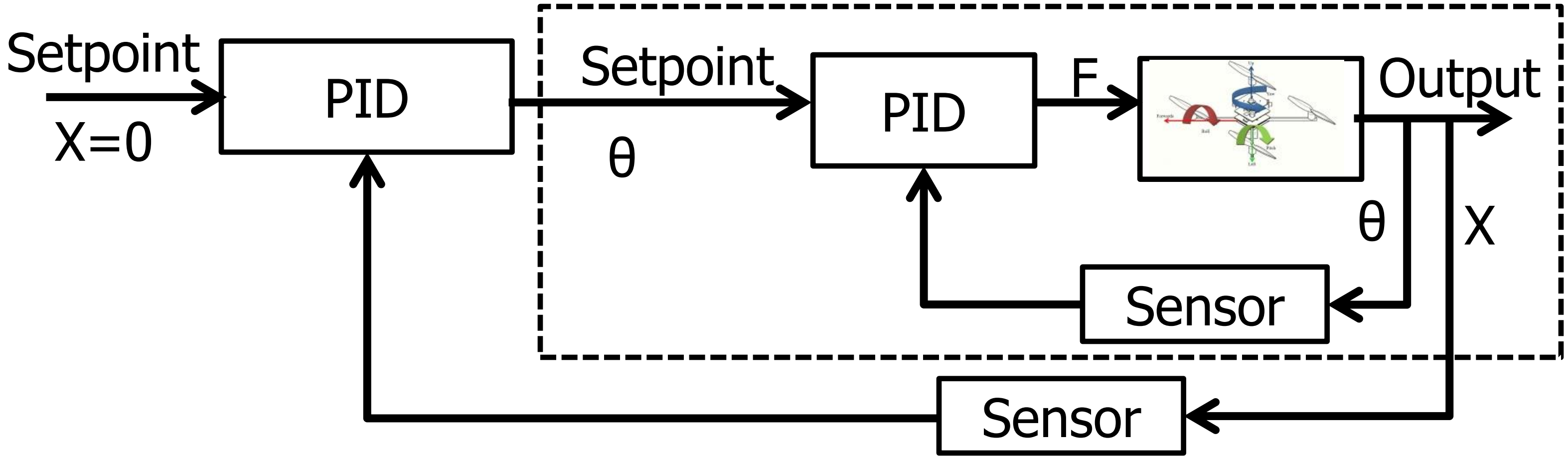
Relation to Quadcopter (Nested PID)





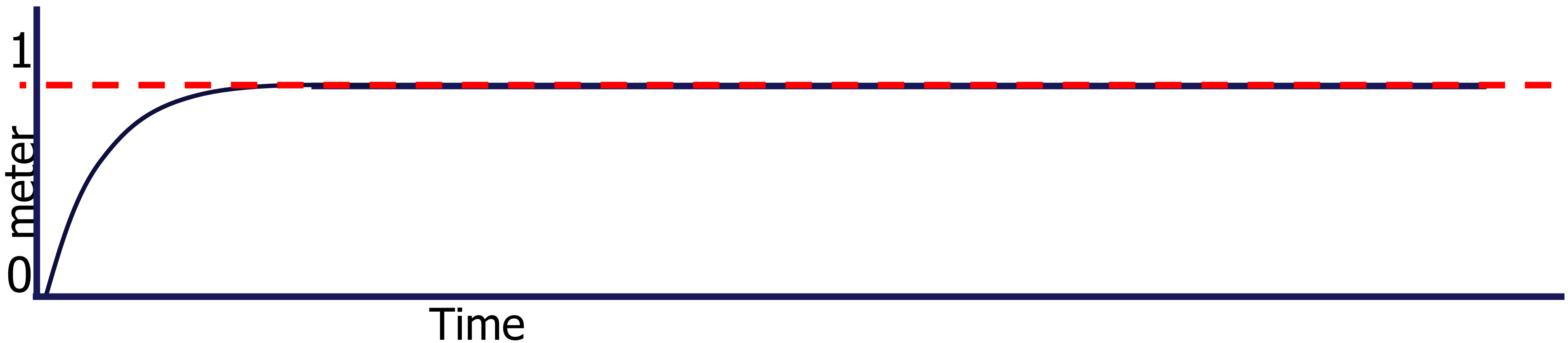
Relation to Quadcopter (Nested PID)





Revisiting the D Constant

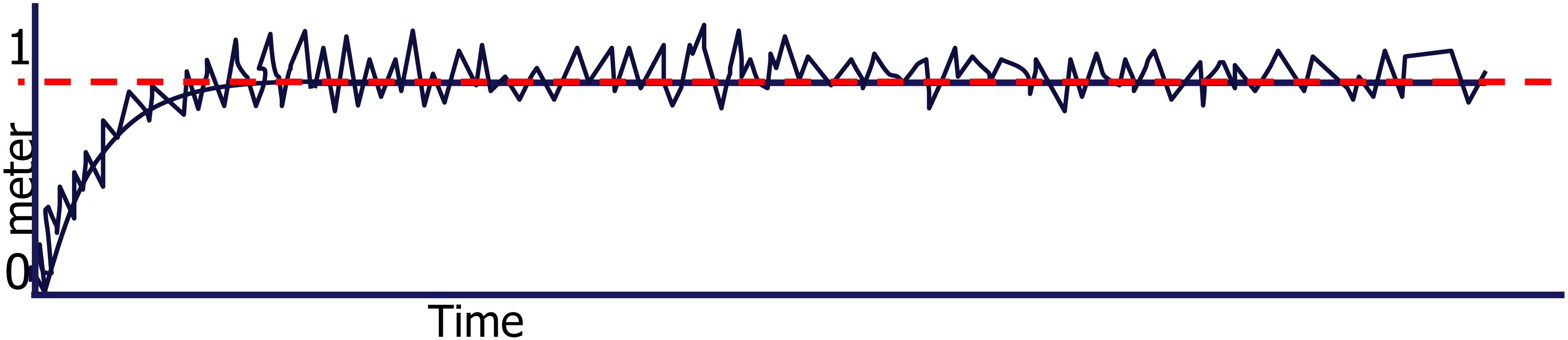
- A large D constant will dampen the system, helping to keep it stable, but causing it to be slow in reacting.
- Are there any issues we need to be concerned with in a real system for a large D constant?



$$u[n] = K_P e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

Revisiting the D Constant (cont.)

- A large D constant will dampen the system, helping to keep it stable, but causing it to be slow in reacting.
- Are there any issues we need to be concerned with in a real system for a large D constant?
- A large D constant will amplify the noise from the sensor which will case the controller to give large spikes of compensation.



$$u[n] = K_P e[n] + K_I \sum_{j=0}^{n} e[j] + K_D(e[n] - e[n-1])$$

PID Tuning Techniques

- There are a few PID tuning techniques, more like rules of thumb (http://en.wikipedia.org/wiki/PID_controller)
  - Manual tuning (Dr. Jones does **not** recommend this manual approach)
    1. Set KI and KD to 0 and increase KP until system oscillate, then turn down some
    2. Increase KI until steady state error is removed
    3. To reduces overshoot and settling time increase D
  - Ziegler–Nichols: heuristic method (Dr. Jones does **not** recommend, unless you are very experienced with Controls, and even then does not recommend)
    1. Set KI and KD to 0
    2. Based on the value of KP that causes the system to oscillate (i.e. KU) and the corresponding oscillation period (PU), compute KP, KI, KD using table

Ziegler–Nichols method

Control Type	K_p	K_i	K_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$1.2K_p/P_u$	-
PID	$0.60K_u$	$2K_p/P_u$	$K_pP_u/8$

PID Tuning Techniques

- Dr. Jones **recommend** approach
  1. Set KI and KD to 0, and increase KP until system starts to overshoot & Oscillate.
  2. Increases KD to reduces overshoot and settling time
  3. Increase KI to remove static error.

Model-based Control

- Controller developed based on a mathematical model of plant
  - Benefits?
  - Draw backs?

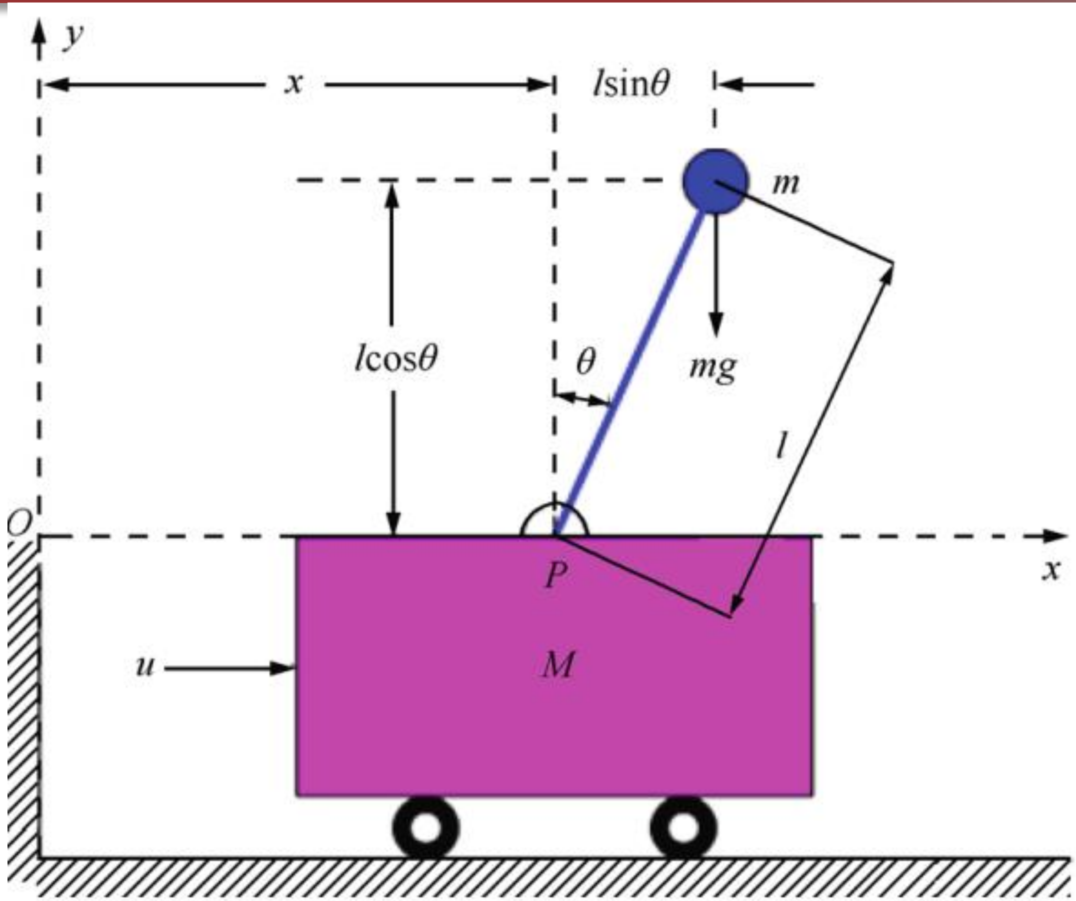
Simple Car Model (point mass)

- Velocity of car = x
- Acceleration of car = x’
- Mass of car = m
- Force acting on care = u (i.e. from gas petal)
- Scaling constant based on car measurements: c

Linear Force (X)

$$x' = \frac{c}{m}u$$

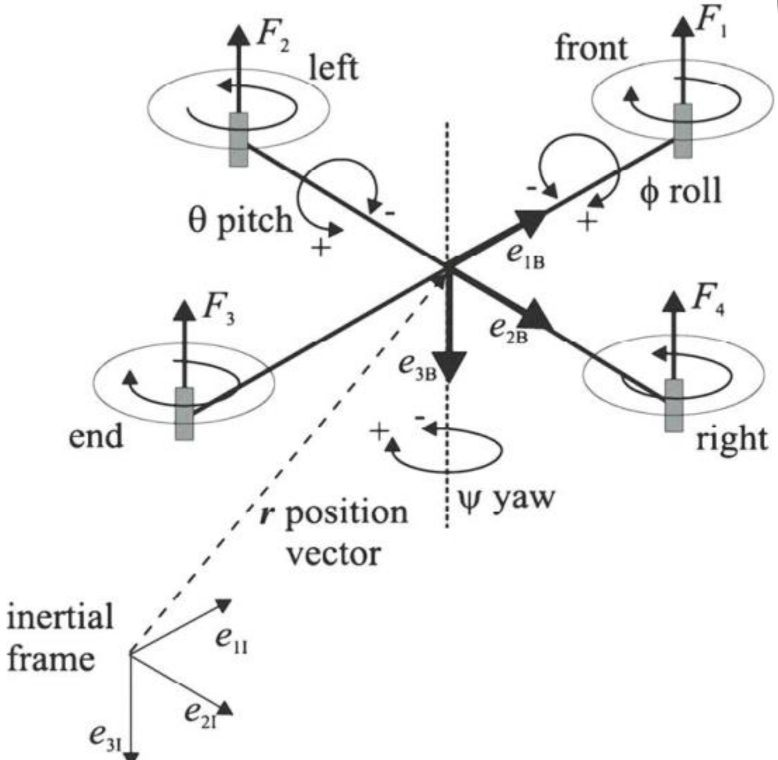
Inverted Pendulum Model



Linear Force (X) $$\ddot{x} = \frac{u + ml(\sin\theta)\dot{\theta}^2 - mg\cos\theta\sin\theta}{M + m - m\cos^2\theta}$$

Rotational Forces (θ) $$\ddot{\theta} = \frac{u\cos\theta - (M+m)g\sin\theta + ml(\cos\theta\sin\theta)\dot{\theta}^2}{ml\cos^2\theta - (M+m)l}$$

Quadcopter Model



Rotational Forces (Φ/θ/Ψ)

$$\begin{cases} \ddot{\phi} = -\dot{\psi}\dot{\theta}C\phi + \frac{lC\psi}{I_{xx}}u_2 - \frac{lS\psi}{I_{yy}}u_3 \\ \quad + \frac{(I_{yy}-I_{zz})}{I_{xx}}(\dot{\psi}-\dot{\theta}S\phi)\dot{\theta}C\phi \\ \ddot{\theta} = \frac{\dot{\psi}\dot{\phi}}{C\phi} + \dot{\phi}\dot{\theta}t\phi + \frac{lS\psi}{C\phi I_{xx}}u_2 + \frac{lC\psi}{C\phi I_{yy}}u_3 \\ \quad - \frac{(I_{yy}-I_{zz})}{I_{xx}}(\dot{\psi}-\dot{\theta}S\phi)\frac{\dot{\phi}}{C\phi} \\ \ddot{\psi} = \dot{\phi}\dot{\psi}t\phi + \frac{\dot{\phi}\dot{\theta}}{C\phi} + \frac{lS\psi t\phi}{I_{xx}}u_2 + \frac{lC\psi t\phi}{I_{yy}}u_3 \\ \quad + \frac{l}{I_{zz}}u_4 - \frac{(I_{yy}-I_{zz})}{I_{xx}}(\dot{\psi}-\dot{\theta}S\phi)\dot{\phi}t\phi \end{cases}$$

Linear Forces (X/Y/Z)

$$\begin{cases} \ddot{x} = -(S\theta C\phi)u_1/m \\ \ddot{y} = (S\phi)u_1/m \\ \ddot{z} = -(C\theta C\phi)u_1/m + g \end{cases}$$

Attitude Control of a Quadrotor with Optimized PID Controller: https://www.researchgate.net/publication/271285250

Model-based Control (cont.)

- Design a controller based on a mathematical model of the plant
- State-space
  - x_k – state of system vector at time k
  - u_k – input vector of system at time k
  - y_k – output vector of system at time k
- Choose u_k to obtained desired y_{k+1}

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k$$

Model-based Control (cont.)

- Design a controller based on a mathematical model of the plant
- State-space
  - x_k – state of system vector at time k
  - u_k – input vector of system at time k
  - y_k – output vector of system at time k
- Choose u_k to obtained desired y_{k+1}

Matrix based off of the physics of the plant (i.e. math-model of the plant)

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k$$

Model-based Control (cont.)

- Design a controller based on a mathematical model of the plant
- State-space
  - x_k – state of system vector at time k
  - u_k – input vector of system at time k
  - y_k – output vector of system at time k
- Choose u_k to obtained desired y_{k+1}

Actuator matrix (i.e. math-model of how u_k gets translated into actuator commands)

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k$$

Model-based Control (cont.)

- Design a controller based on a mathematical model of the plant
- State-space
  - x_k – state of system vector at time k
  - u_k – input vector of system at time k
  - y_k – output vector of system at time k
- Choose u_k to obtained desired y_{k+1}

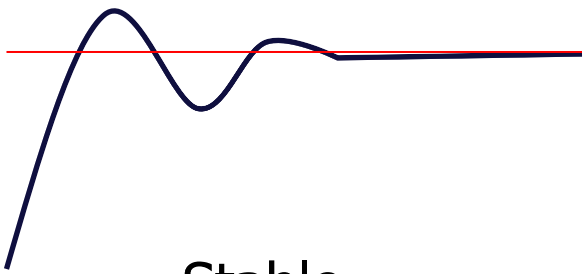
Sensor matrix (i.e. express what plant states you can observe with sensors)

$$x_{k+1} = Ax_k + Bu_k$$

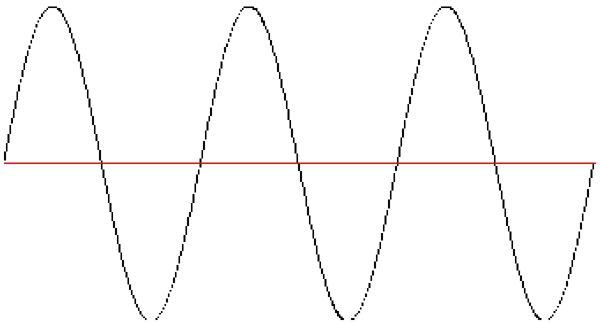
$$y_k = Cx_k$$

Typical Controller Metrics

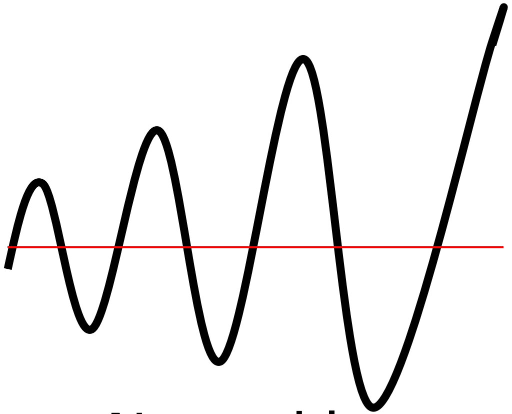
- **Stability**: (e.g. bounded oscillation of system output)



Stable



Marginally stable



Not stable

- For a stable controlled system
  - **Disturbance Rejection**: How well does the system hold setpoint in the presence of a disturbance (e.g., shoving a quadcopter)
  - **Command tracking**: How well does the system respond to changes in the controller setpoint
    - Rise time
    - Settling time

Control Systems Summary

- **PID (<u>no plant model available</u>)**
  - <u>Benefits:</u>
    - Very useful for controlling many commonly found systems
    - Do not need much knowledge of the plant being controlled
  - <u>Drawbacks:</u>
    - Only can control a single input single output (SISO)system
    - Can lead to hand tuning many constants.
    - Tuning even more challenging when dependencies exists
- **PID (<u>with plant model</u>)**
  - <u>Benefits:</u>
    - Easy to gain intuition for how constants impact system
    - There are tools that can computed constants (as a starting point)
  - <u>Drawbacks:</u>
    - If you have a plant model, then there are more advanced controllers you can use (e.g., state space observer models)

Control Systems: Next Steps?

- **Control of Mobile Robots** (Georgia Tech): **Great 6-week intro!!!**
  - https://www.youtube.com/playlist?list=PLp8ijpvp8iCvFDYdcXqqYU5Ibl_aOqwjr

- **Signal & Systems**
  - **"The Scientist and Engineer's Guide to Digital Signal Processing"**
    - **A great hands-on minimum math approach to Signals & Systems, and Digital Signal Processing:** https://www.dspguide.com/pdfbook.htm
  - **"Introduction to Signals & Systems":** https://web.stanford.edu/~boyd/ee102/
    - **Stephen Boyd, Stanford**

- **Linear Dynamical Systems (i.e., Applied Linear Algebra)**
  - https://ee263.stanford.edu/archive/ **(Stephen Boyd, Stanford)**

- **Iowa State University:**
  - **EE 224: Signals & Systems I, EE 324: Signal & Systems II**
  - **EE 475: Control Systems I**
  - **EE 476: Control Systems II (mostly Lab)**

Acknowledgments

- These slides are inspired in part by material developed and copyright by:
  - Maxim Raginsky (University of Illinois)
  - Magnus Egerstedt (Georgia Tech)

PID control: P control