

# SELF-BALANCING ROBOT

Capstone Project

Presented by

Mohammad Saydeh



# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

---

Comparison and Discussion

---

Conclusion & Future Perspectives

# BACKGROUND: SELF-BALANCING ROBOTS

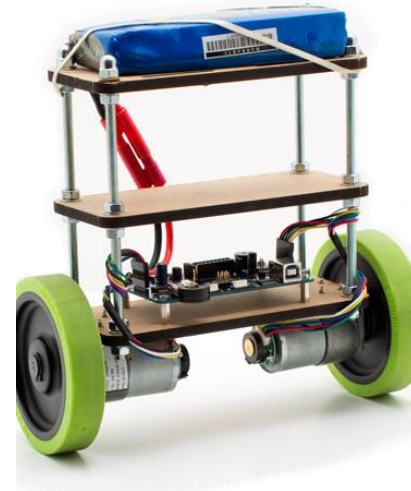
Require control systems

Invention started from 1980s

Non-linear & unstable

Sustainable Design

Used in multiple fields



Balanduino



Segway



Hover Boards



Legway



# MOTIVATION

Newly and continuously growing field

Considered a sustainable and multi-purpose robot

Using Simscape platform to implement a modeled version of the robot

Possibility of comparing different control algorithms

# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

---

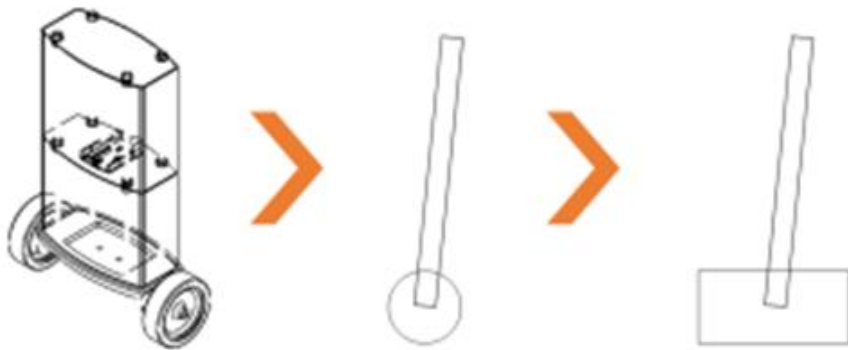
Comparison and Discussion

---

Conclusion & Future Perspectives

---

# PRINCIPLE OF INVERTED PENDULUM IP



Other than TWSBRs, IP's applications:

- Human Walking Robots
- Earthquake resistant building design
- Missile Launchers

# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

---

Comparison and Discussion

---

Conclusion & Future Perspectives

# LITERATURE REVIEW

## An & Li 2013

### LQR & PID Simulations' Comparison

- Controlling self-balance
- PID's feedback: either tilt angle or tilt angle rate
- LQR: involves combining tilt angle tilt angle rate and position
- LQR achieved steady state faster than PID

## Rahman et al., 2018

### LQR PID & FLC Comparison

- Implemented on a TWSBR using ROS and Gazebo
- PID gave the most stable response in the real time Pitch angles plot
- LQR was the faster
- FLC was non-stable due to inefficient tuning

## ETH Zurich University, 2022

Design and implement of a new form of a self balancing robot ( 2whld robot – 4whld robot, and quadruped)

### LQR PID & FLC Comparison

- Implemented on a two-wheeled inverted pendulum mobile robot
- Feedback composed of tilt angle and position
- FLC showed less overshoot and faster response but consumed higher energy
- LQR showed faster response and less overshoot than PID

## Bature et al., 2014

### LQR & PID Comparison

- TWSBR MATLAB simulation and real implementation
- Both met the specifications: less than 200 ms setting time & less than 5 degrees tilt
- PID had higher overshoot but less steady state error
- LQR had less overshoot and minimal steady state error

## Jiménez et al., 2020



# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

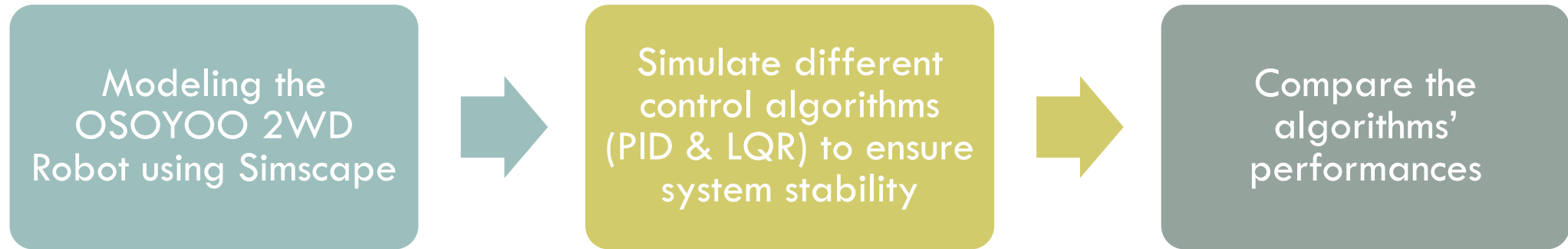
---

Comparison and Discussion

---

Conclusion & Future Perspectives

# PURPOSE



# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

---

Comparison and Discussion

---

Conclusion & Future Perspectives

# CONTROL METHODS

## PID

- SISO
- Minimize error
- Can be tuned using  $K_p$ ,  $K_i$  &  $K_d$  parameters
- $K_p$ : Present
- $K_i$ : Past
- $K_d$ : Future

## LQR

- MIMO/SIMO
- Based on States
- Feedforward & Feedback Controls
- Governed by gains that minimize the cost function :
  - $J = \int \{x'Qx + u'Ru + 2 * x'Nu\} dt$

# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

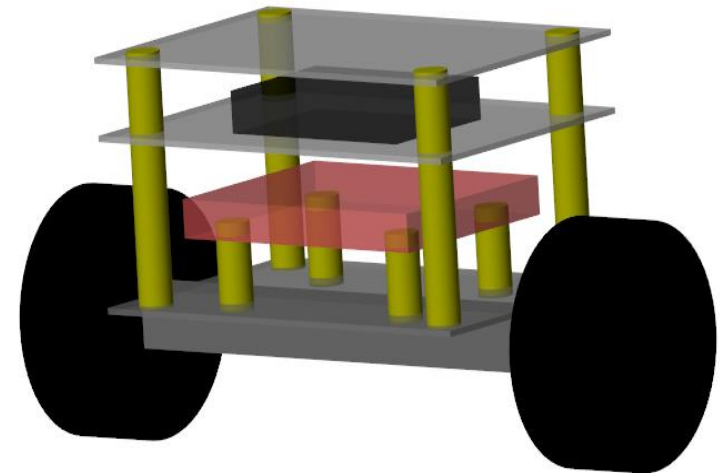
---

Comparison and Discussion

---

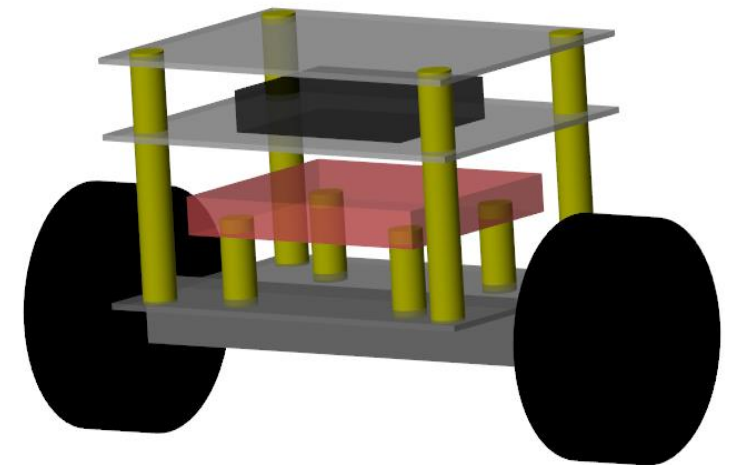
Conclusion & Future Perspectives

# MODELING OF THE 2WD ROBOT

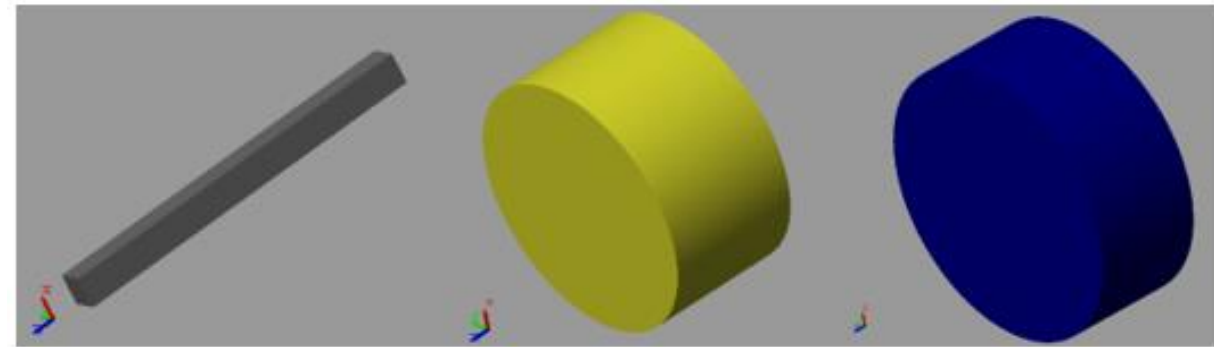
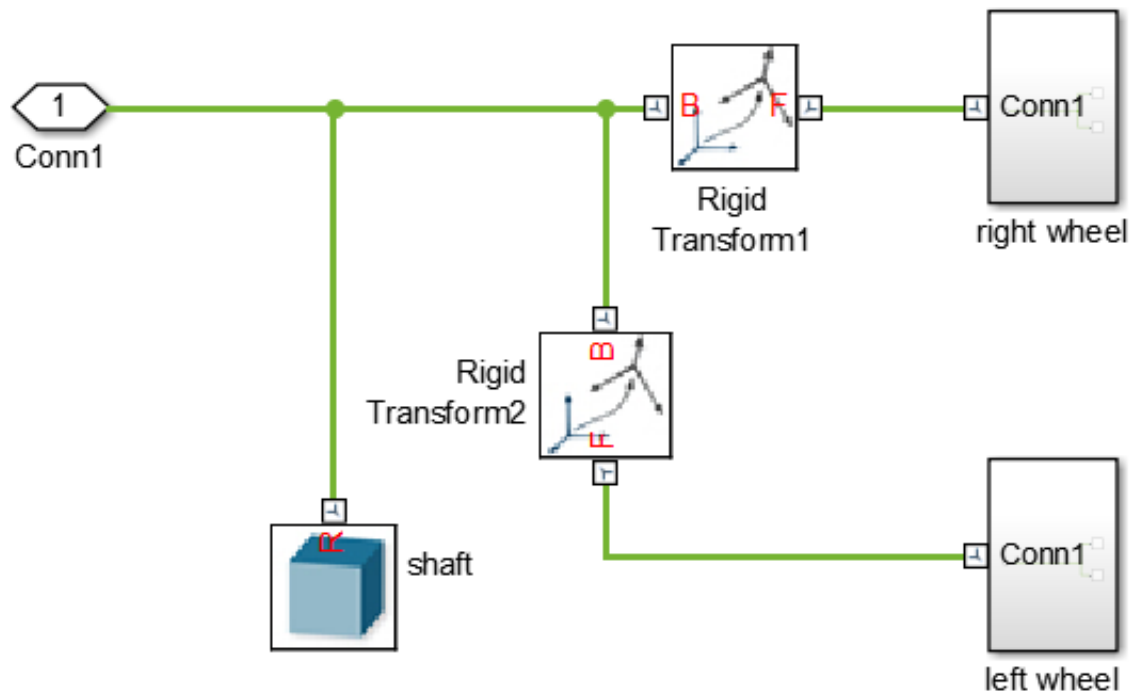


# MODELING: CART

Shaft			Unit
Shape	Brick		
Dimensions	[1.5,1.5,18]		cm
Mass	345		g
Left & Right Wheels			Unit
	Wheel Body	Wheel Tire	
Shape	Cylinder		
Radius	2.36	3.2	cm
Length	2.57	2.6	cm
Mass	23.4	25.6	g



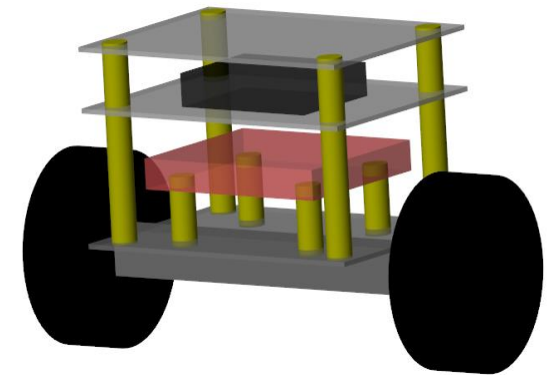
# MODELING: CART



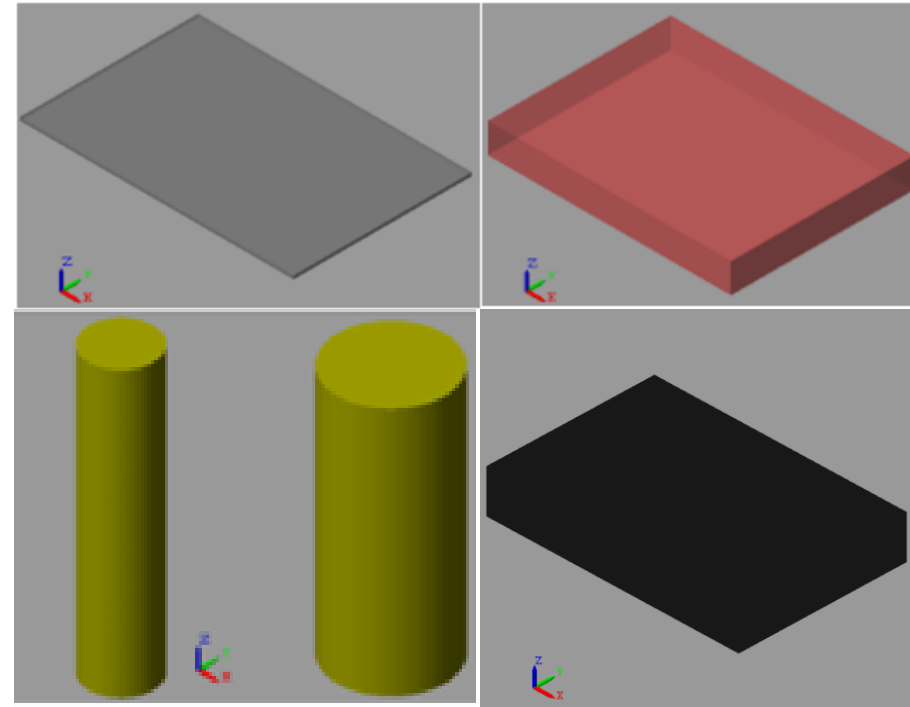
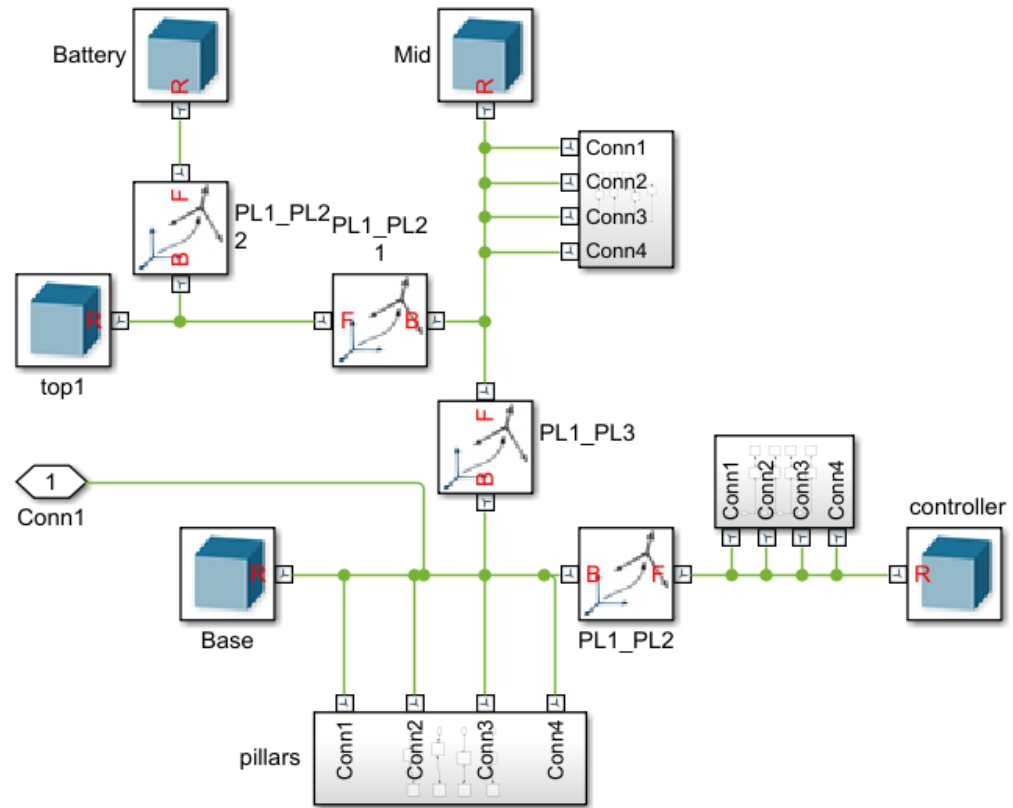


# MODELING: CHASSIS

Levels						Unit
	Base	Controller	Middle	Top	Battery	
<b>Shape</b>	Brick					
<b>Dimensions</b>	[12.58,8.2,0.16]	[8.22 6.19 1]	[12.58,8.2,0.16]	[12.58,8.2,0.16]	[6,4,1.1]	cm
<b>Mass</b>	107.8	69.6	63	63	105	g
Rods					Unit	
	1 <sup>st</sup> Set		2 <sup>nd</sup> Set			
<b>Shape</b>	Cylinder					
<b>Radius/rod</b>	0.5		0.5		cm	
<b>Length/ rod</b>	4.4		2.3		cm	
<b>Mass/ rod</b>	6.1		2.9		g	
<b>Number/Set</b>	4		8		Rods	



# MODELING: CHASSIS



# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

---

Comparison and Discussion

---

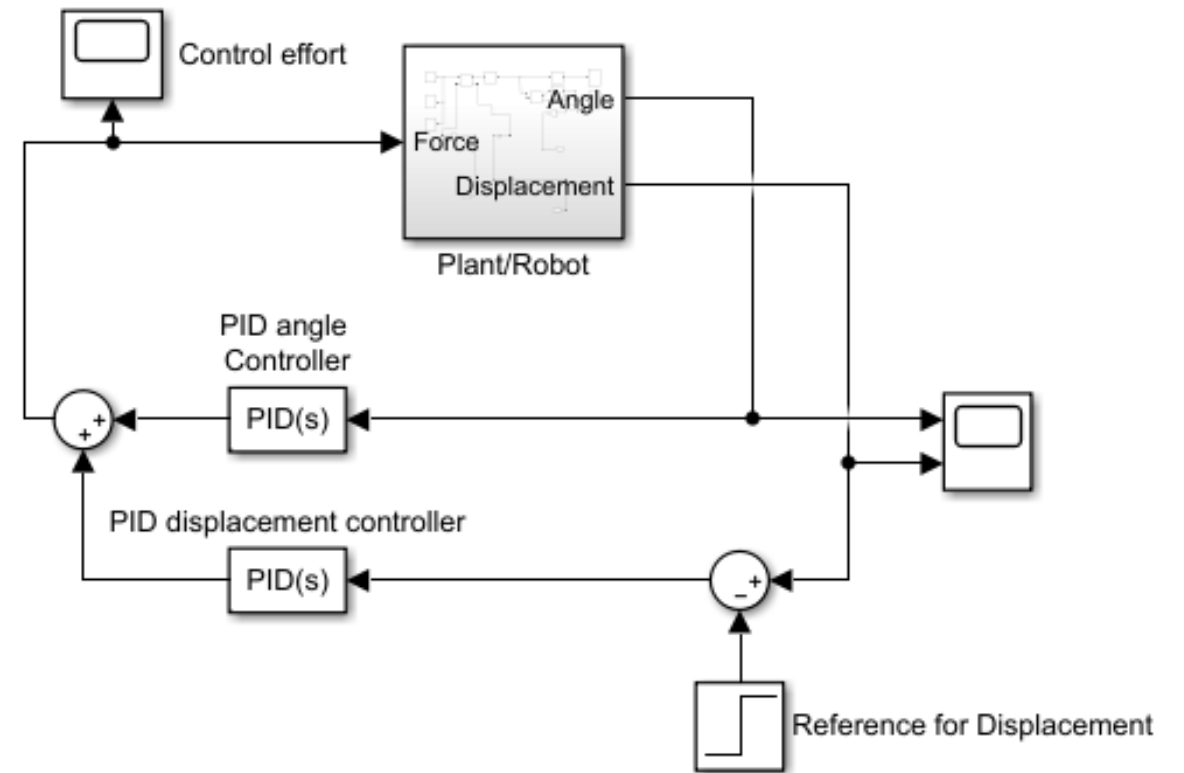
Conclusion & Future Perspectives

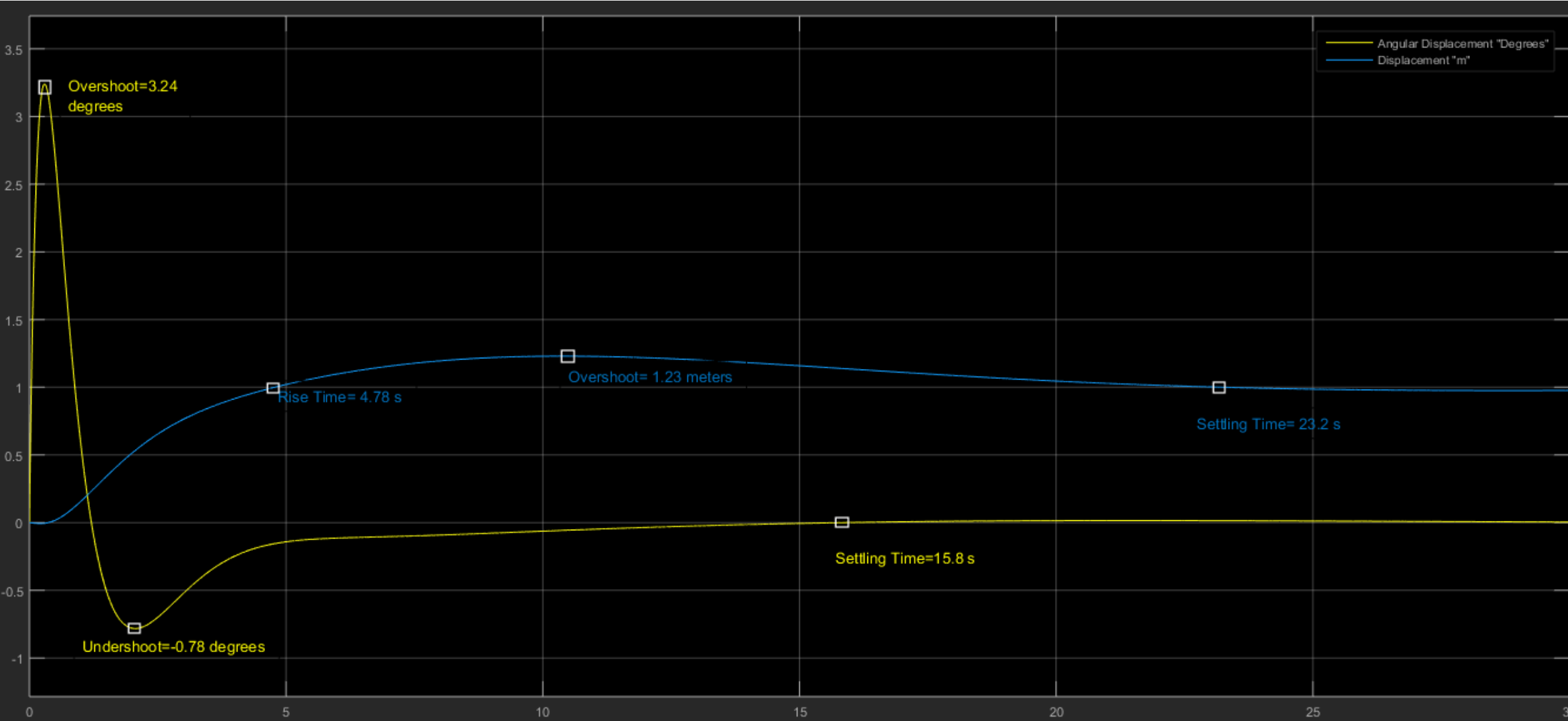


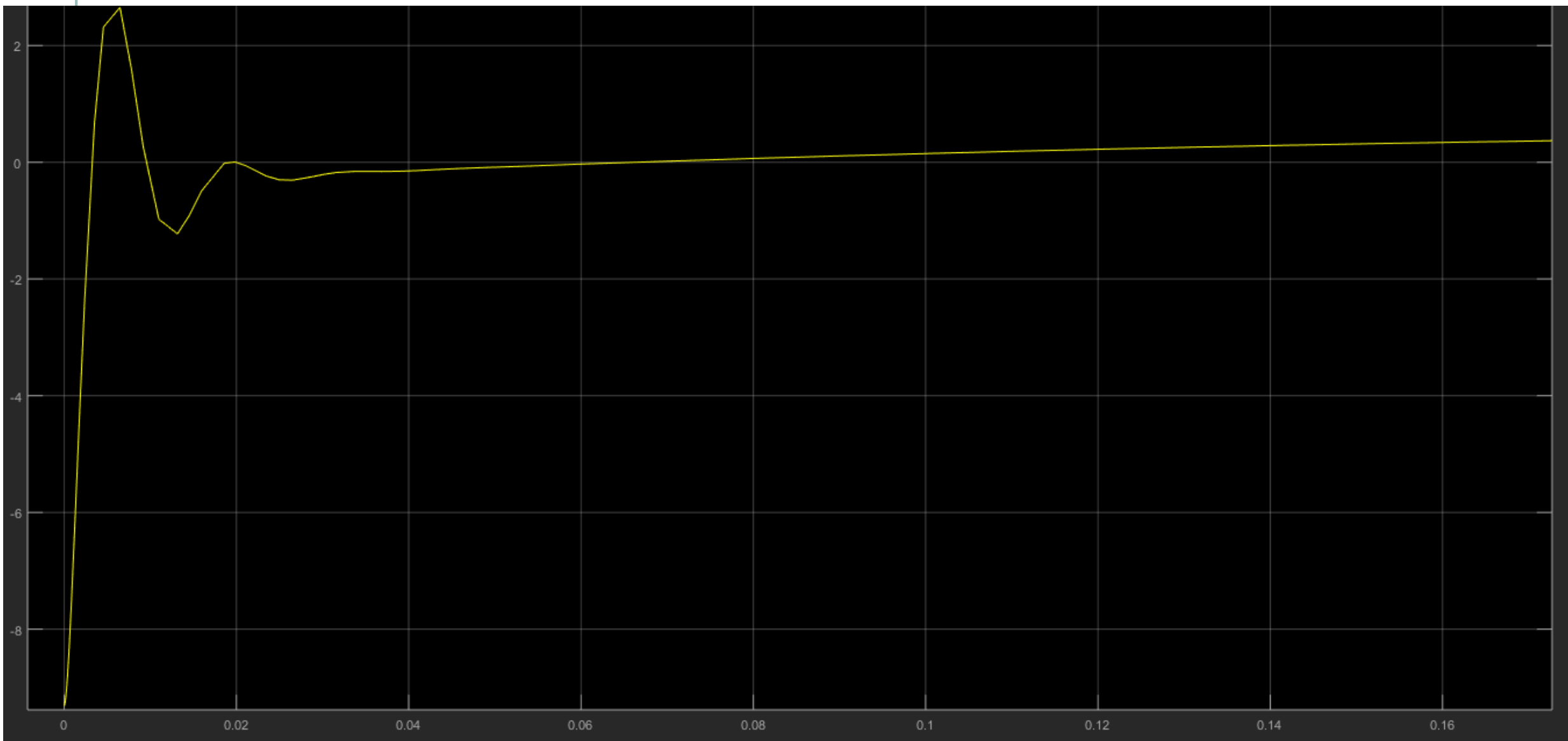
# **SIMULATION: PID**

# SIMULATION: PID

- Recording “angle”, “displacement”, and “control effort” responses with respect to time
- The robot is controlled via displacement force “f”
- PID Feedback is received from position sensor & gyroscope “p” & “q”
- $K_p$ ,  $K_d$  and  $K_i$  gains are adjusted in each of the PID controllers







# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

---

Comparison and Discussion

---

Conclusion & Future Perspectives





# SIMULATION: LQR



# LQR CONTROLLER SPECS

**Margins Goal**

Name: MarginsGoal1

**Purpose**  
Enforce specific gain and phase margins (disk margins for MIMO feedback loops).

**Feedback Loop Selection**  
Measure margins at the following locations:  
SelfbalseftunerMoo/Sum5/1  
+ Add signal to list

Measure margins with the following additional loops open:  
+ Add loop opening location to list

**Desired Margins**  
Gain margin: 6 dB  
Phase margin: 40 deg

**Step Tracking Goal**

Name: StepTrackingGoal1

**Purpose**  
Make specific closed-loop step response closely match the desired response.

**Step Response Selection**  
Specify step-response inputs:  
SelfbalseftunerMoo/Step1/1[xref]  
+ Add signal to list

Specify step-response outputs:  
SelfbalseftunerMoo/Sum1/1  
+ Add signal to list

Compute step response with the following loops open:  
+ Add loop opening location to list

**Desired Response**  
Specify as  
 First-order characteristics  
 Second-order characteristics  
 Custom reference model

Time constant: 10  
Overshoot (%): 2

**Reference Tracking Goal**

Name: TrackingGoal1

**Purpose**  
Follow reference commands with prescribed performance and fidelity. Limit cross-coupling in MIMO systems.

**Response Selection**  
Specify reference inputs:  
SelfbalseftunerMoo/Step1/1[xref]  
+ Add signal to list

Specify reference-tracking outputs:  
SelfbalseftunerMoo/Subsystem/1  
+ Add signal to list

Evaluate tracking performance with the following loops open:  
+ Add loop opening location to list

**Tracking Performance**  
Specify as  
 Response time, DC error, and peak error  
 Maximum error as a function of frequency

Response Time: 1 s  
Steady-state error (%): 0.1  
Peak error across frequency (%): 100

**Options**  
Enforce goal in frequency range: [0 Inf] rad/s  
Adjust for signal amplitude: No

Apply goal to  
 All models  
 Only models: [1 2]

**Poles Goal**

Name: PolesGoal1

**Purpose**  
Constrain the dynamics of the closed-loop system, specific feedback loops, or specific open-loop configurations.

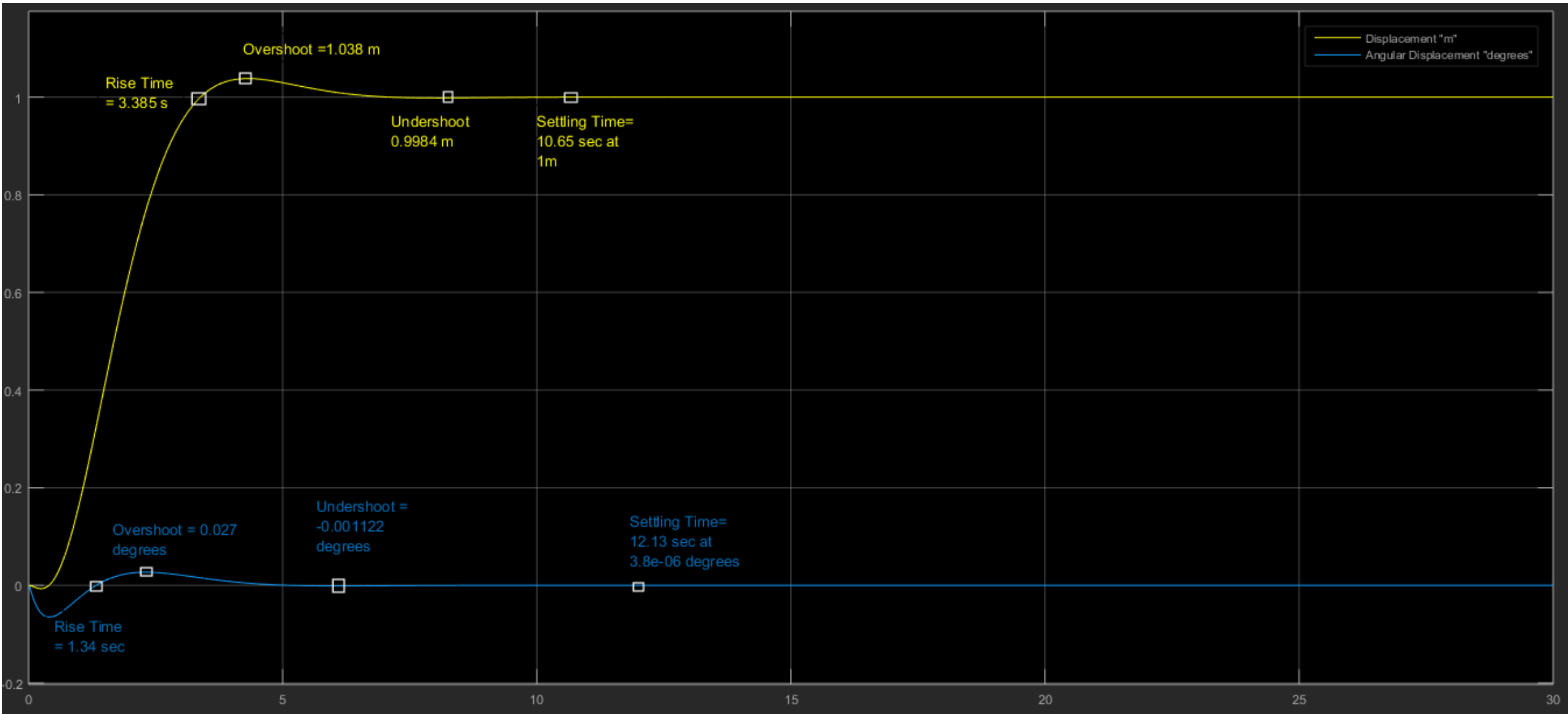
**Feedback Configuration**  
Compute poles of:  
 Entire system  
 Specific feedback loop(s)

Compute poles with the following loops open:  
+ Add loop opening location to list

**Pole Location**  
Keep poles inside the following region:  
Minimum decay rate: 0  
Minimum damping: 0.7  
Maximum natural frequency: 45

**Options**  
Enforce goal in frequency range: [0 Inf] rad/s  
Apply goal to:  
 All models  
 Only models: [1 2]

OK Apply Cancel ?



# CONTROL EFFORT



# LQR

SIMO system

State Variables:  $x, \dot{x}, \theta, \dot{\theta}$

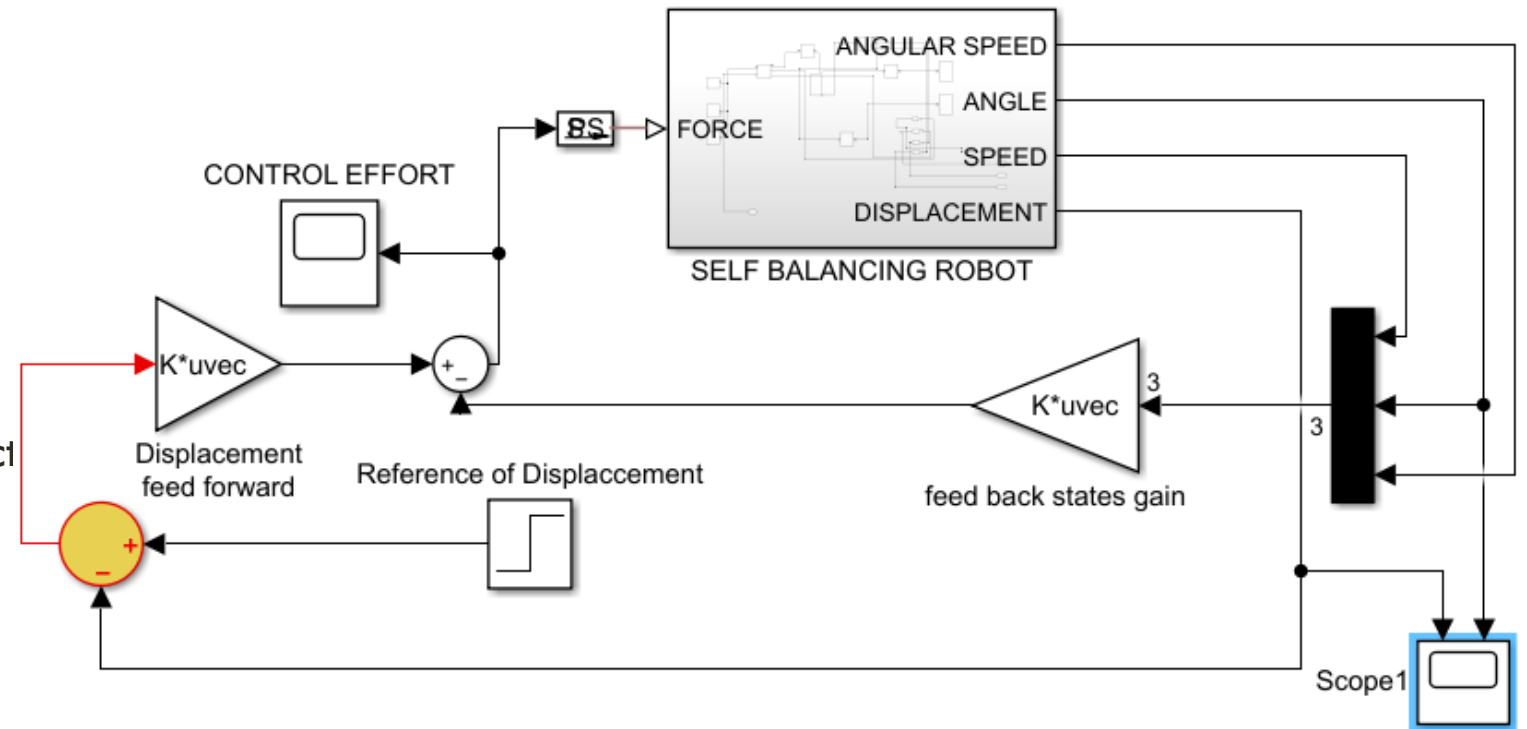
Additional sensors

Consists of feed forward & feedback controls

Gains are calculated based on the "Control Analysis" in MATLAB based on specific contract

$u = -Kx$  minimizes the cost function

$$J = \int \{x'Qx + u'Ru + 2 * x'Nu\} dt$$



# LQR WITH MOO

$x$  and  $\emptyset$  are observed

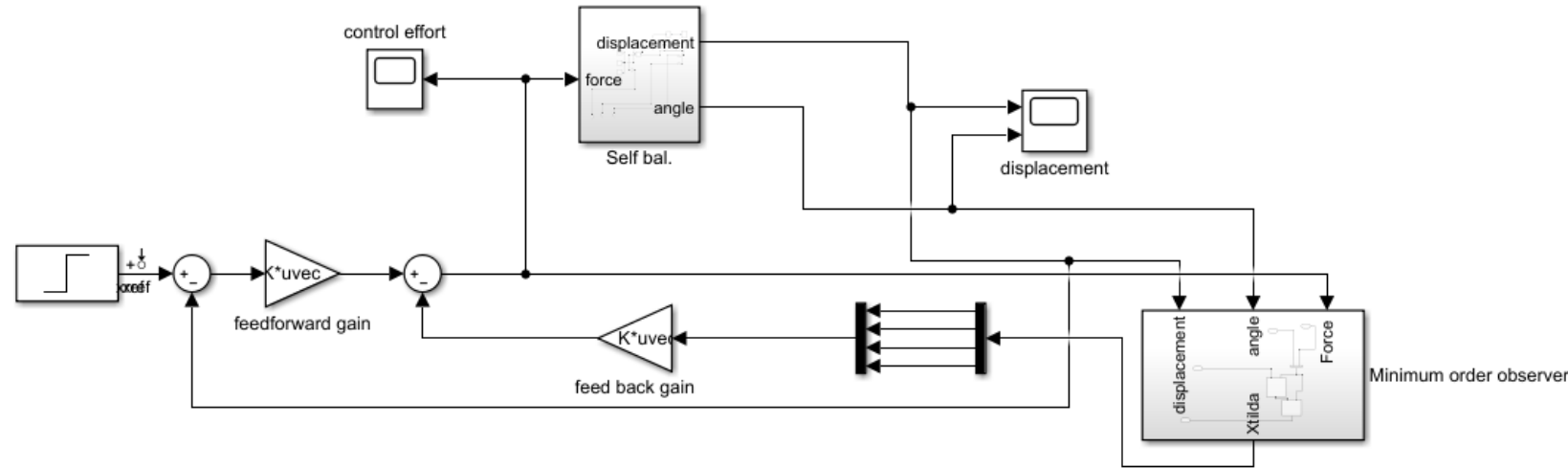
$\dot{x}$  and  $\dot{\emptyset}$  are to be estimated

“Ethatilda” block gets the state variables required to acquire the estimated states by:

$$\dot{\tilde{\eta}} = \hat{A}\tilde{\eta} + \hat{B}y + \hat{F}u$$

The above “Ethatilda” and the measured states “ $y$ ” are then fed to the transformation block which will output the estimated states “xtilda”

$$\tilde{x} = \hat{C}\tilde{\eta} + \hat{D}y$$



# INVERTED PENDULUM TRANSFER FUNCTIONS

Result after applying newton's second law then linearization on Pendulum and car separately

$$(I + ml^2) \ddot{\theta} - mgl \theta = ml\ddot{x}$$

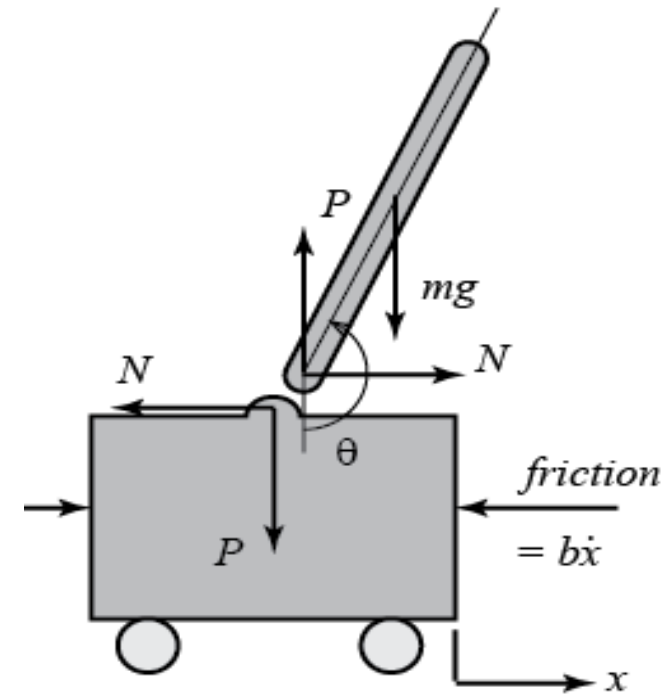
$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\theta} = F$$

Laplace Transfrom

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(l + ml^2)}{q}s^2 - \frac{mgl(M + m)}{q}s - \frac{bmgl}{q}} \left[ \frac{\text{rad}}{\text{N}} \right]$$

$$\frac{X(s)}{U(s)} = \frac{\frac{q}{(I + ml^2)s^2 - gml}}{s^4 + \frac{b(l + ml^2)}{q}s^3 - \frac{mgl(M + m)}{q}s^2 - \frac{bmgl}{q}s} \left[ \frac{\text{m}}{\text{N}} \right]$$

Where  $q = [(M+m)(I+ml^2) - (ml)^2]$





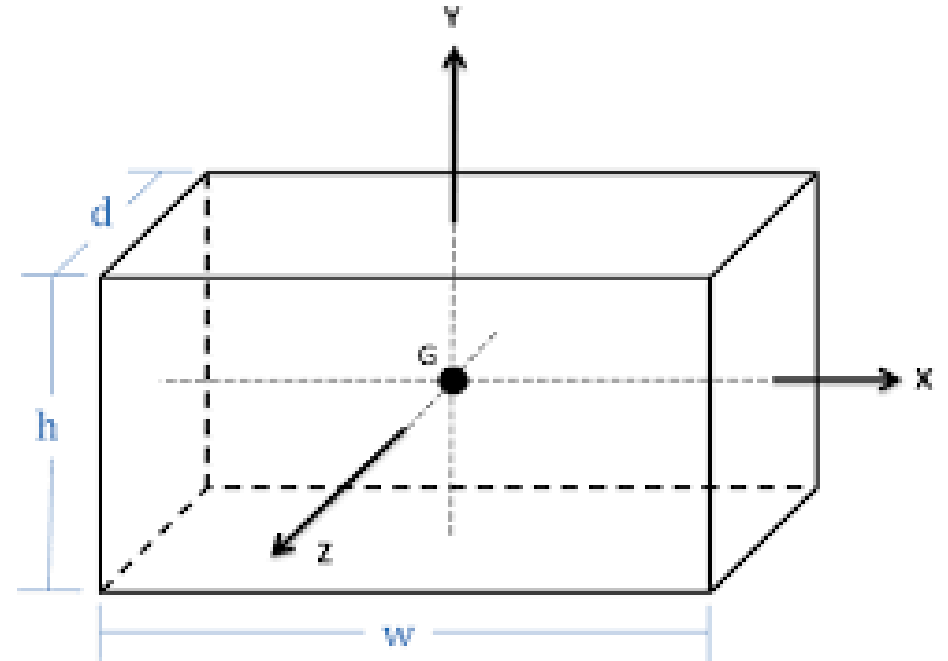
# INVERTED PENDULUM STATE SPACE MODEL

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\emptyset} \\ \ddot{\emptyset} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I + ml^2)b}{I(M + m) + Mml^2} & \frac{0}{I(M + m) + Mml^2} & \frac{m^2 gl^2}{I(M + m) + Mml^2} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{-mlb}{I(M + m) + Mml^2} & \frac{mgl(M + m)}{I(M + m) + Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \emptyset \\ \dot{\emptyset} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0}{I(M + m) + Mml^2} \\ 0 \\ \frac{ml}{I(M + m) + Mml^2} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \emptyset \\ \dot{\emptyset} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

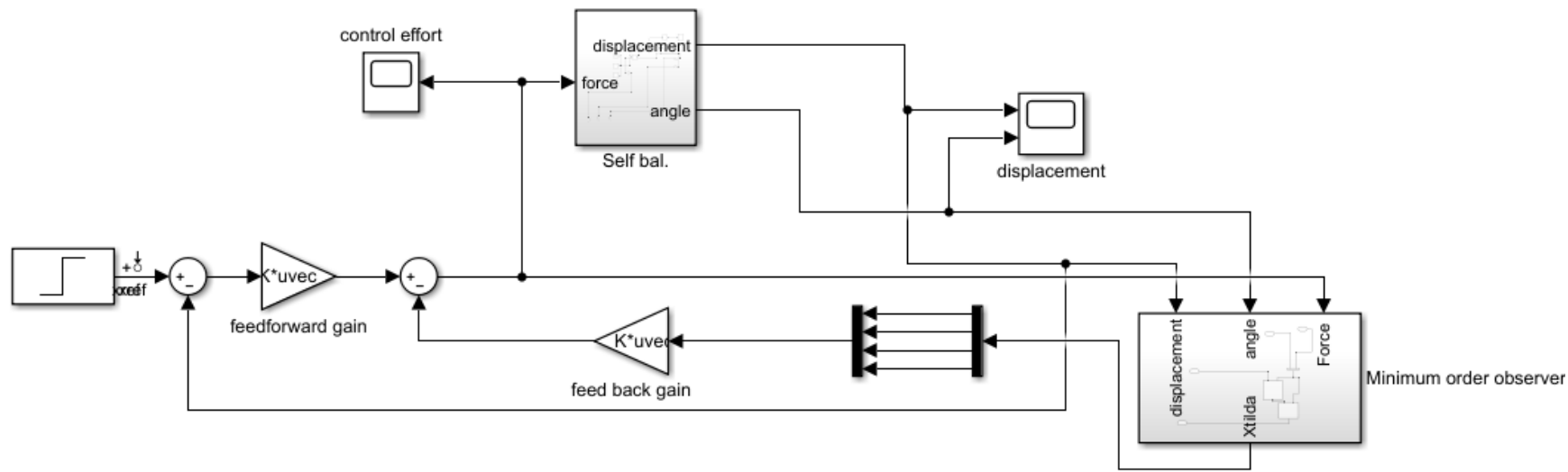
# ROBOT'S CHARACTERISTICS

	Label	Value	Unit
Mass of the Cart	M	0.5500	Kg
Mass of the Chassis	m	0.4696	Kg
Length to chassis center of mass	l	3.59	cm
Coefficient of friction of the cart	b	0.1	N/m/sec
Mass moment of inertia of the Chassis	I	0.0004648 7	Kg.m <sup>2</sup>



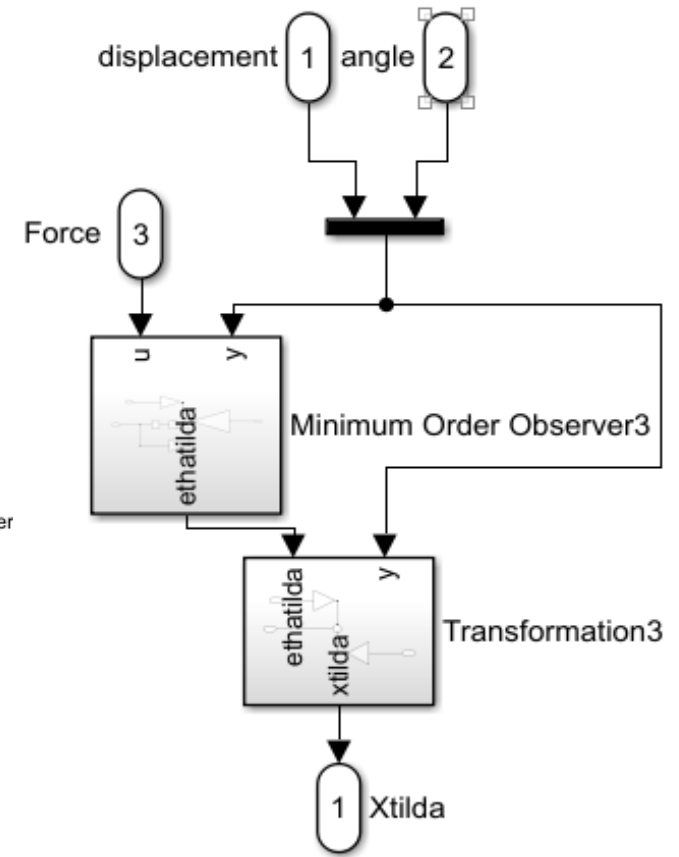
$$I = \frac{1}{12}m(h^2 + d^2)$$

# LQR WITH MOO



$$\dot{\tilde{\eta}} = \hat{A}\tilde{\eta} + \hat{B}y + \hat{F}u$$

$$\tilde{x} = \hat{C}\tilde{\eta} + \hat{D}y$$



Minimum Order Observer

# LQR CONTROLLER SPECS

**Margins Goal**

Name:

**Purpose**  
Enforce specific gain and phase margins (disk margins for MIMO feedback loops).

**Feedback Loop Selection**  
Measure margins at the following locations:  
   
+ Add signal to list

Measure margins with the following additional loops open:  
+ Add loop opening location to list

**Desired Margins**

Gain margin:  dB  
Phase margin:  deg

**Step Tracking Goal**

Name:

**Purpose**  
Make specific closed-loop step response closely match the desired response.

**Step Response Selection**  
Specify step-response inputs:  
   
+ Add signal to list

Specify step-response outputs:  
   
+ Add signal to list

Compute step response with the following loops open:  
+ Add loop opening location to list

**Desired Response**  
Specify as  
 First-order characteristics  
 Second-order characteristics  
 Custom reference model

Time constant:   
Overshoot (%):

**Reference Tracking Goal**

Name:

**Purpose**  
Follow reference commands with prescribed performance and fidelity. Limit cross-coupling in MIMO systems.

**Response Selection**  
Specify reference inputs:  
   
+ Add signal to list

Specify reference-tracking outputs:  
   
+ Add signal to list

Evaluate tracking performance with the following loops open:  
+ Add loop opening location to list

**Tracking Performance**  
Specify as  
 Response time, DC error, and peak error  
 Maximum error as a function of frequency

Response Time:  s  
Steady-state error (%):   
Peak error across frequency (%):

**Options**  
Enforce goal in frequency range:  rad/s  
Adjust for signal amplitude:

Apply goal to  
 All models  
 Only models:

**Poles Goal**

Name:

**Purpose**  
Constrain the dynamics of the closed-loop system, specific feedback loops, or specific open-loop configurations.

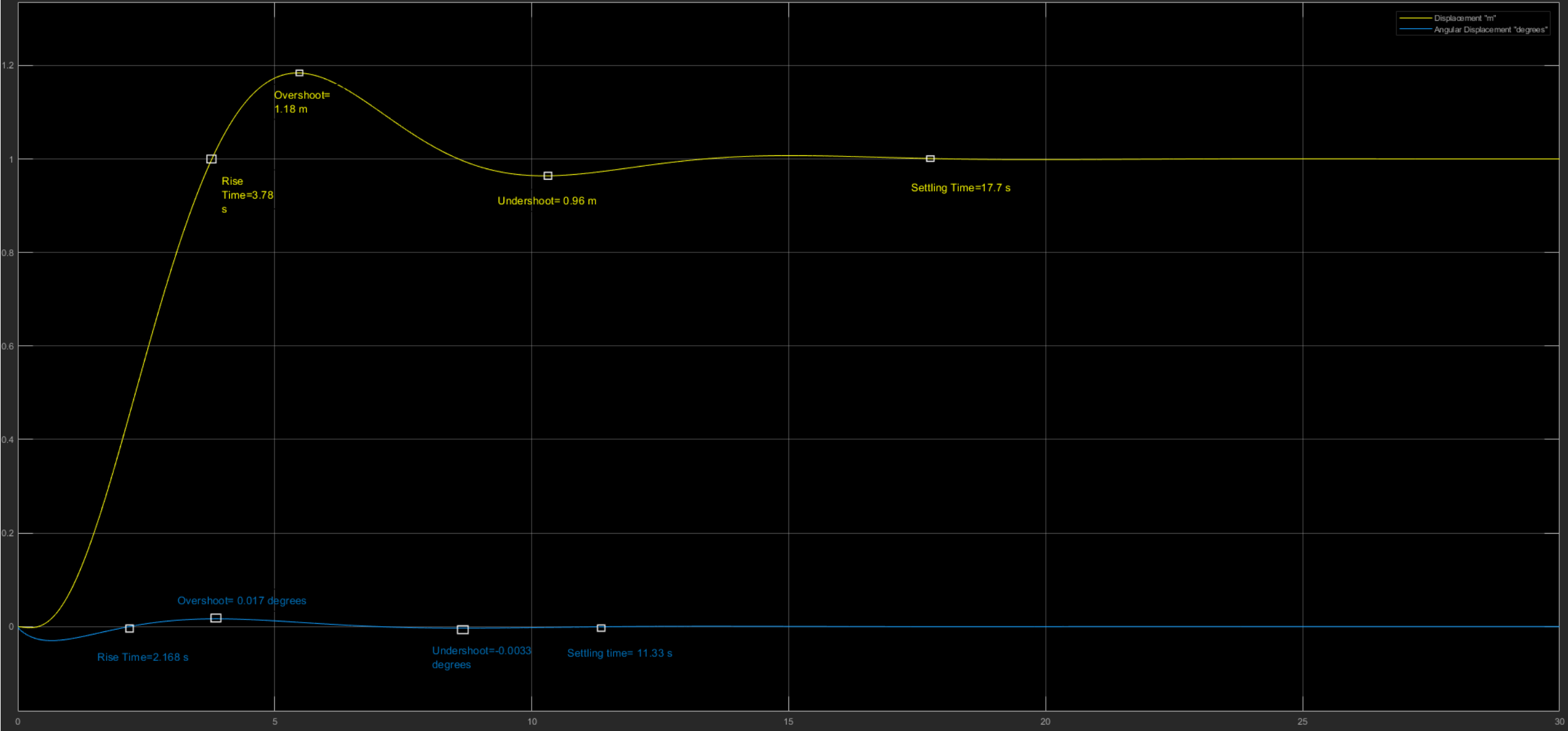
**Feedback Configuration**  
Compute poles of:  
 Entire system  
 Specific feedback loop(s)

Compute poles with the following loops open:  
+ Add loop opening location to list

**Pole Location**  
Keep poles inside the following region:  
Minimum decay rate:   
Minimum damping:   
Maximum natural frequency:

**Options**  
Enforce goal in frequency range:  rad/s  
Apply goal to:  
 All models  
 Only models:

OK Apply Cancel ?



— Displacement "m"  
— Angular Displacement "degrees"

Overshoot=  
1.18 m

Rise  
Time=3.78  
s

Undershoot= 0.96 m

Settling Time=17.7 s

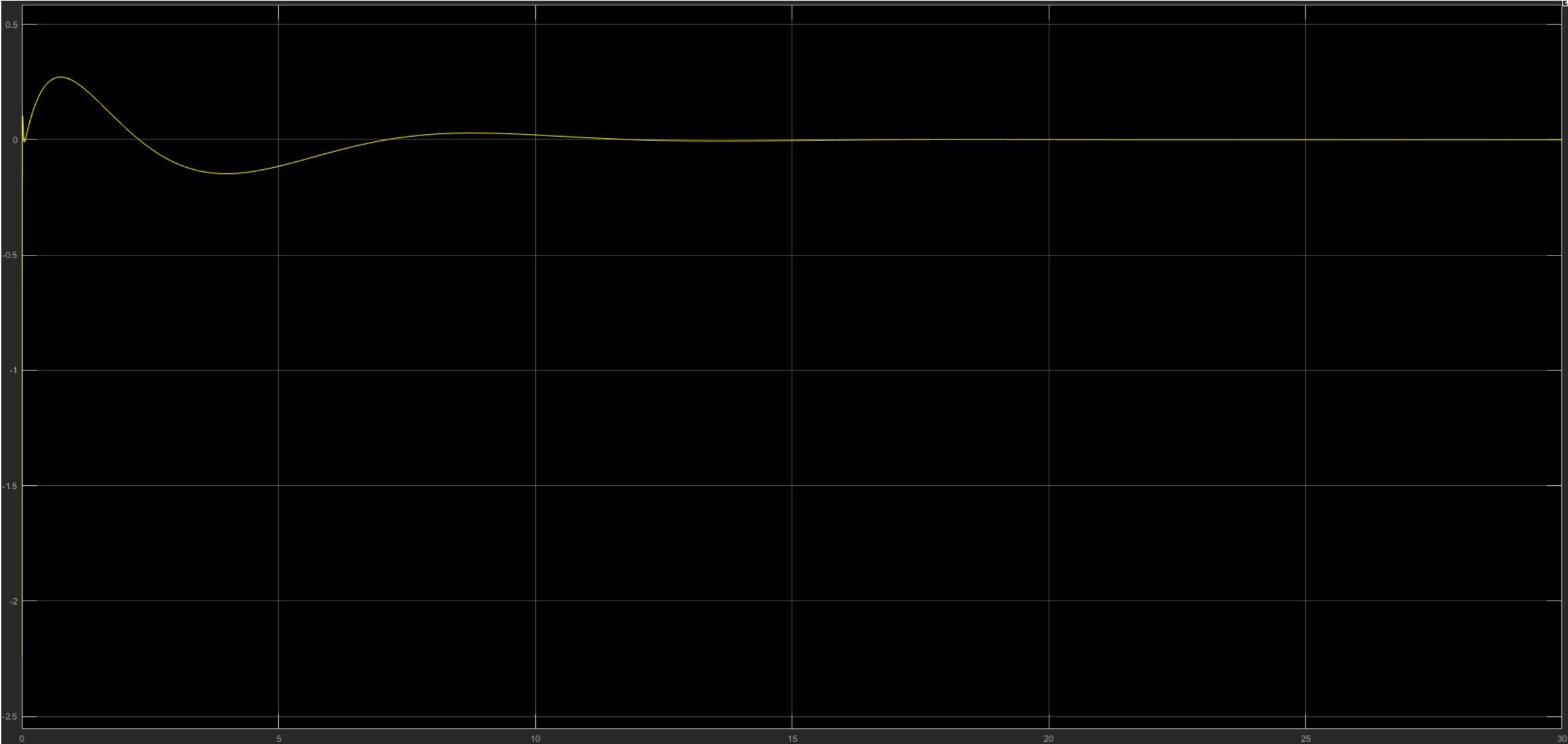
Overshoot= 0.017 degrees

Rise Time=2.168 s

Undershoot=-0.0033  
degrees

Settling time= 11.33 s

# CONTROL EFFORT



# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

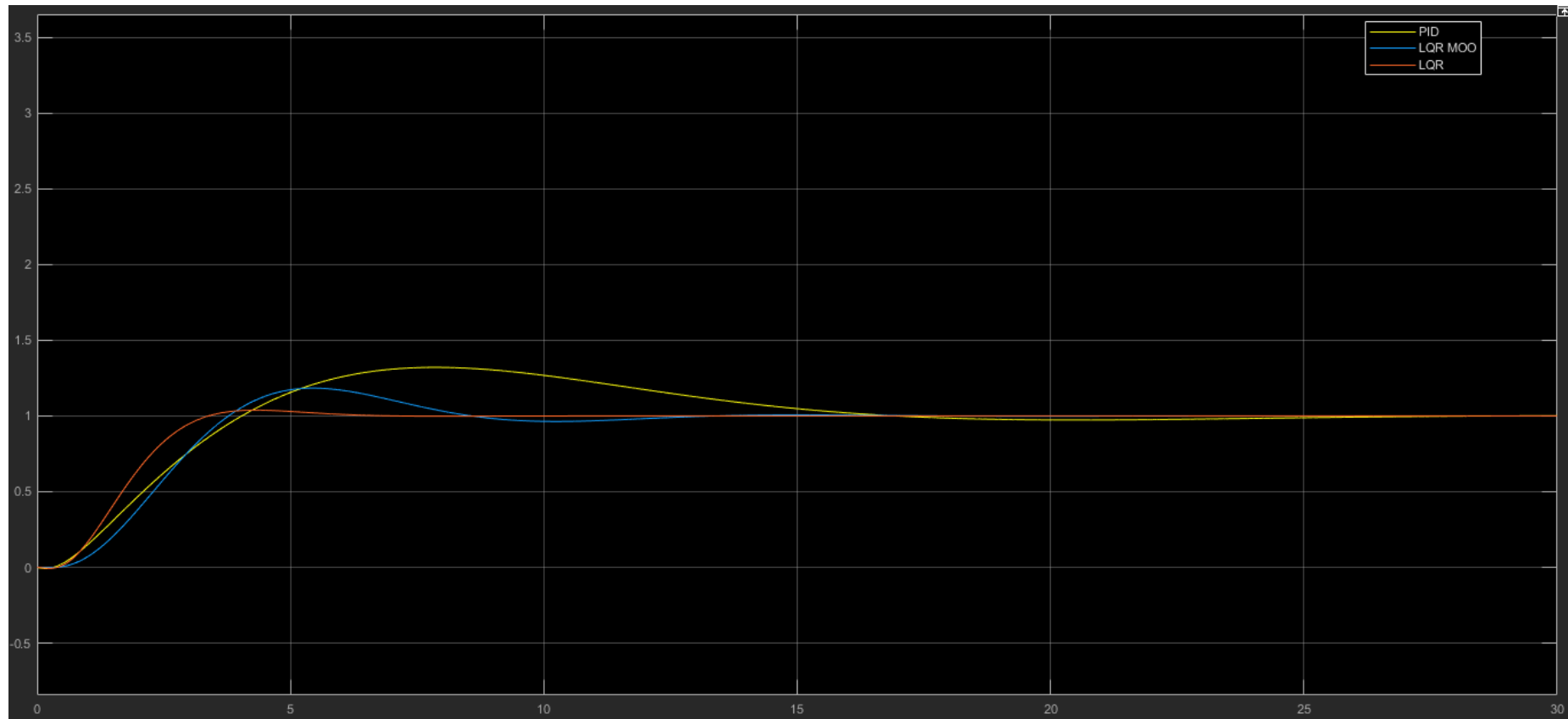
---

Comparison and Discussion

---

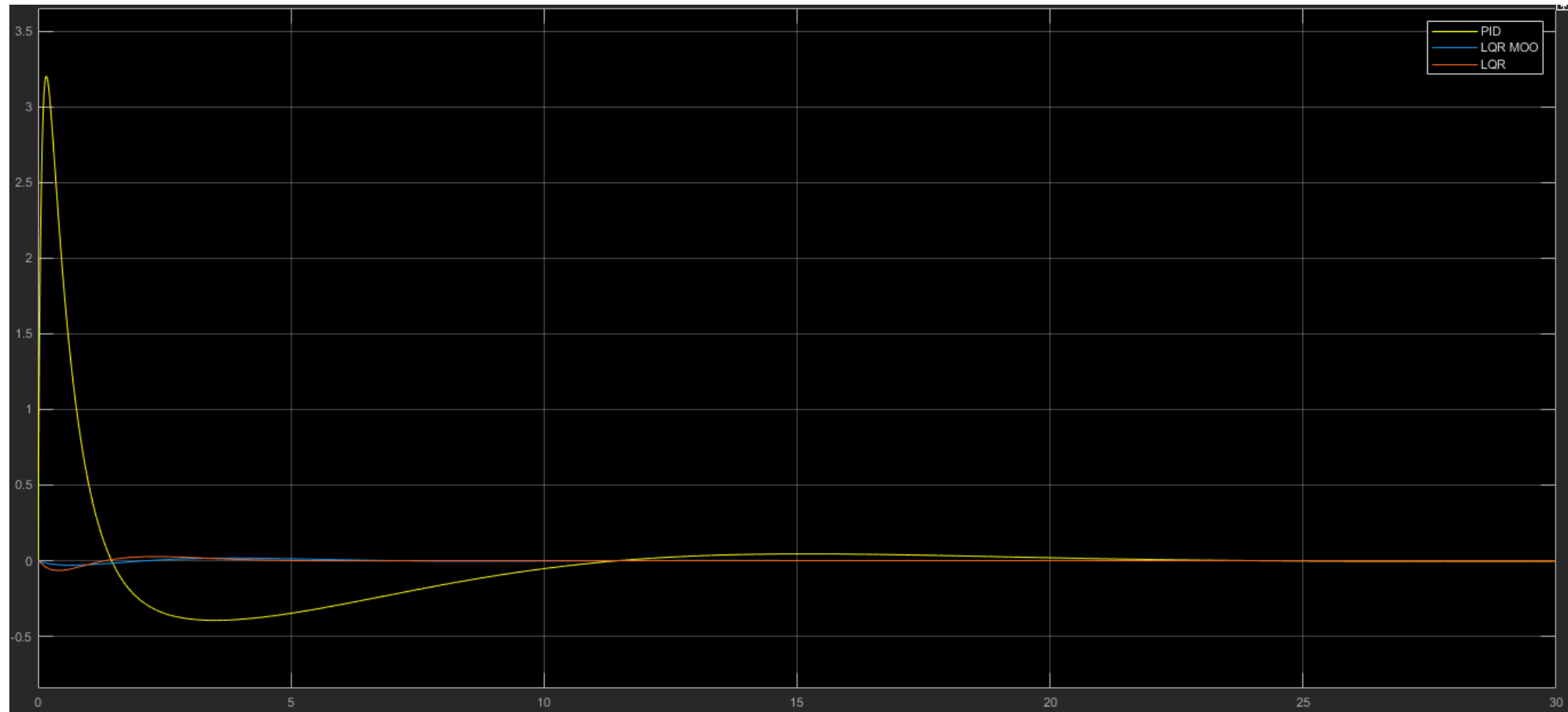
Conclusion & Future Perspectives

# DISPLACEMENT

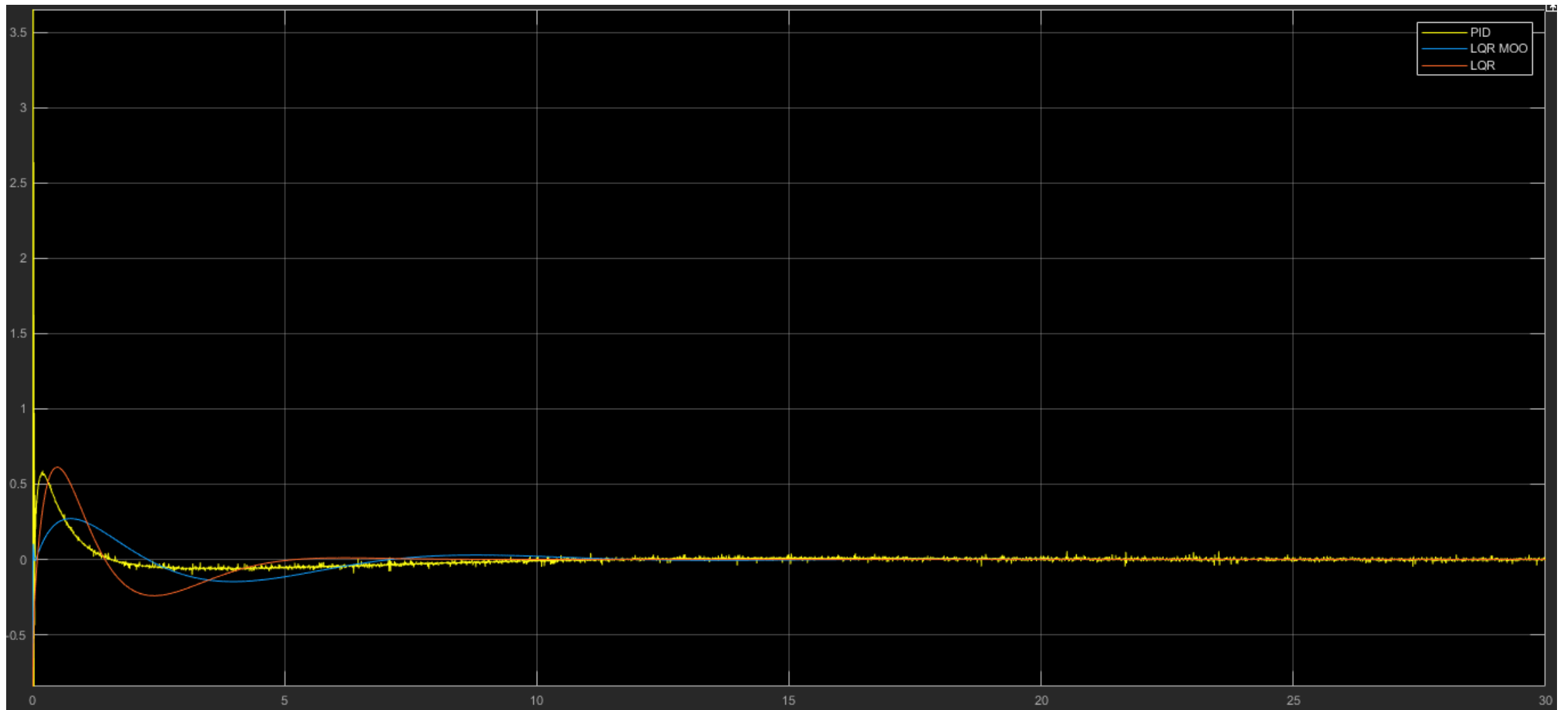




# ANGLE



# CONTROL EFFORT



# VIDEO

The screenshot displays the MATLAB R2018a software interface. At the top, the title bar reads "MATLAB R2018a". Below it is a ribbon menu with tabs for "HOME", "PLOTS", "APPS", "MECHANICS EXPLORERS", and "VIEW". A search bar on the right contains the text "Search Documentation" and the user's name "Mohammad".

The main workspace is divided into several sections. On the left, there is a "Mechanics Explorers" pane with a tree view showing a model structure: "SelfbalseltunerMoo" containing "Plant\_Robot", "Self\_bal", and "SELF\_BALANCING\_ROBOT". Below this is a "Current Folder" pane showing the path "C:\Users\mohammad\Desktop\LIB SBR\mysbr.m".

The central area is a 3D plot window titled "Mechanics Explorer-SelfbalseltunerMoo". It shows a 3D model of a self-balancing robot (a two-wheeled robot with a balancing pole) in a coordinate system with X, Y, and Z axes. The robot is positioned in the center of the plot. A simulation timeline at the bottom of the plot window shows "T = [0,30]" and "Time 30".

At the bottom of the interface, there are panes for "Workspace" and "Command Window".

# OUTLINE

---

Background & Motivation

---

Principle of Inverted Pendulum

---

Literature Review

---

Purpose

---

Control Methods

---

Modeling of the 2WD Robot

---

Simulation: PID

---

Simulation: LQR

---

Comparison and Discussion

---

Conclusion & Future Perspectives

# CONCLUSION & FUTURE PERSPECTIVES

- LQR somehow gave the most promising results
- However, LQR is the most expensive in terms of sensors
- There is no optimal controller that meets all user requirements
- User must compromise based on his application and choose the best controller

## Future Perspectives

- Real implementation on the actual robot
- Performance comparison between actual and empirical results

**THANK YOU**