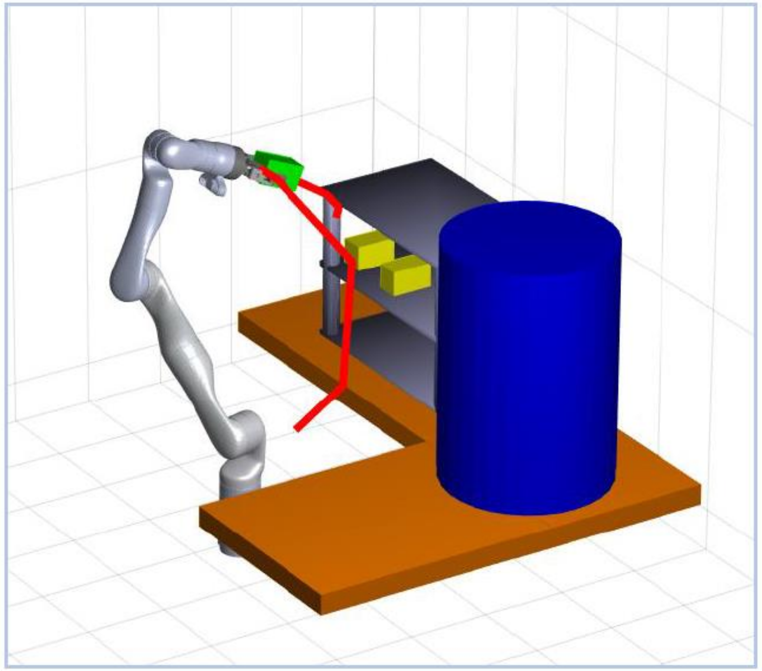
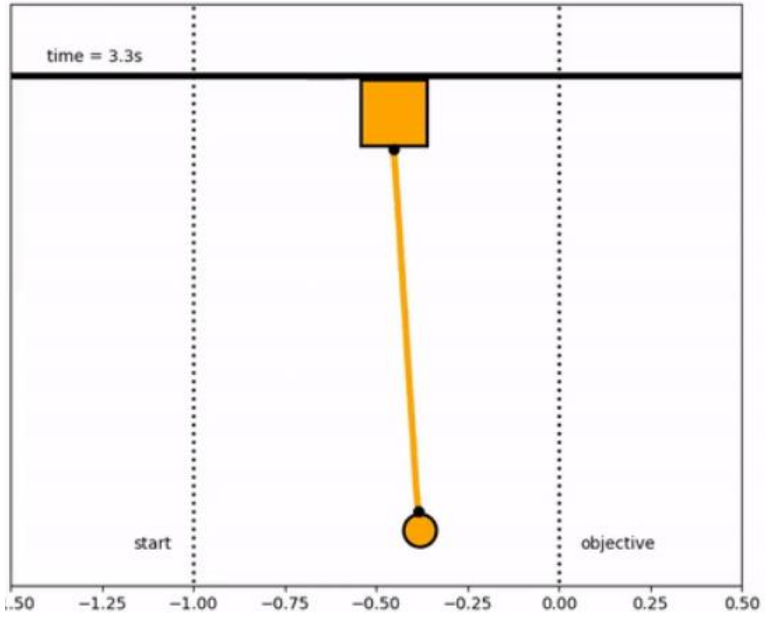
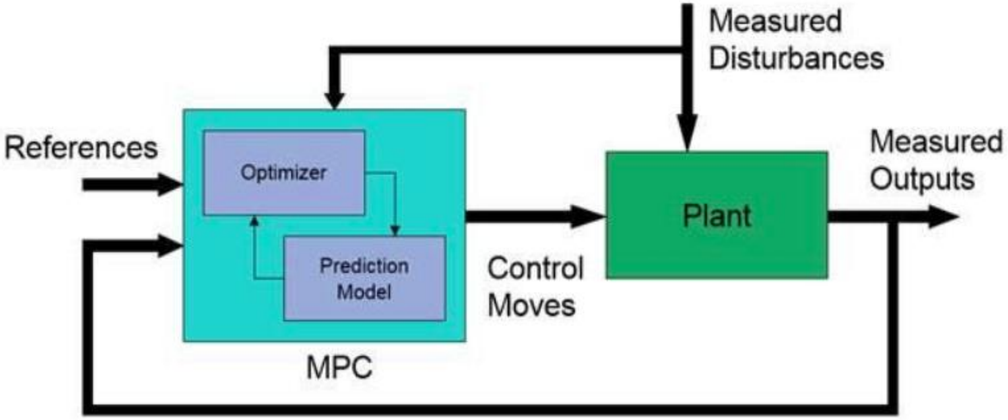
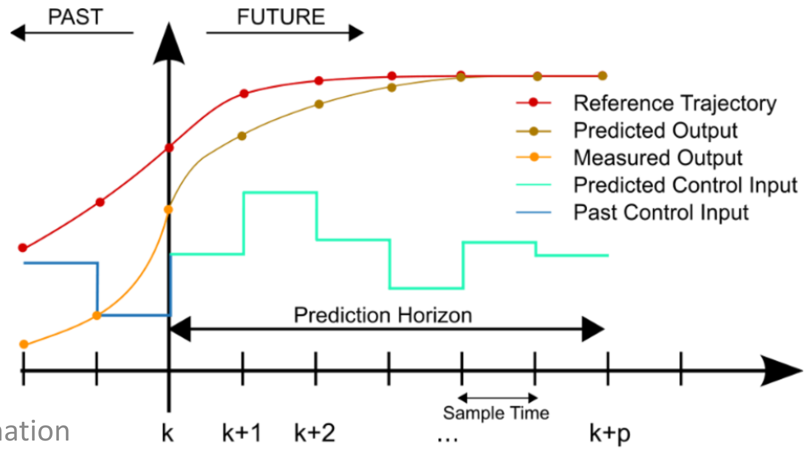




Model Predictive Control

MPC





Asst. Prof. Claudia F. YAŞAR YTU Control and Automation Engineering Department-2023

MATLAB Drive

	Intelligent Control Systems KOM5101	Preparation + Homework	Matlab Drive
1	Introduction to intelligent control systems (knowledge-based vs data-driven systems)	Select the project from https://github.com/mathworks/MathWorks-Excellence-in-Innovation#mathworks-excellence-in-innovation-projects	https://drive.matlab.com/sharing/c1f9073b-a0b0-4966-95b0-c107691878da
2	Computational thinking tools	Work with the Virtual Hardware and Labs for Control. Solve the following Labs Lab4_PositionAnalysis.mlx Lab3_PositionControl.mlx Lab2_VehicleModel.mlx Lab1_CruiseControl.mlx	https://drive.matlab.com/sharing/77e65af2-6ffd-4709-a0bd-c36e0fbe50df
3	Dynamical systems modelling	1. Study and Obtain the state space model of a crane system. 2. Study and Obtain the state space model of the Lateral Vehicle Dynamics bicycle model with two degrees of freedom, lateral position and yaw angle.	https://drive.matlab.com/sharing/31e0ba39-b3f8-402c-a428-6a5b9d620081
4	Model Predictive Control MPC	Study and work with the MPC models explained. Use the MPC Toolbox of Matlab and the apmonitor server. Learn how to work with the drivingScenarioDesigner. Program the MPC algorithms using Simulink and Live scripts. Modify Models and MPC parameter and settings.	https://drive.matlab.com/sharing/398fa9fa-4650-4316-ab2b-0d228b24f48c

Model Predictive Control (MPC Toolbox will be used)

• Linear MPC design example using MPC Toolbox and Simulink: Will be used to teach students how to design linear MPCs using MPC Designer App.
Un example that uses an automated driving application will be used.

Linear MPC Overhead-crane with a State Space Model: We will design and program a model predictive controller for an overhead crane with a pendulum mass. The system has to meet specific control objectives by tuning the controller and using the state space model of the crane system.
Un example of a simulation and optimization will be accomplished for the pendulum system.

Using the APMonitor Server

The APMonitor Modeling Language is optimization software for mixed-integer and differential algebraic equations. It is coupled with large-scale solvers for linear, quadratic, nonlinear, and mixed integer programming. Modes of operation include data reconciliation, real-time optimization, dynamic simulation, and nonlinear predictive control. It is freely available through MATLAB, Python, or from a web browser interface.

APM.IMODE – APMonitor Option

A Global Options | F M S C Local Options

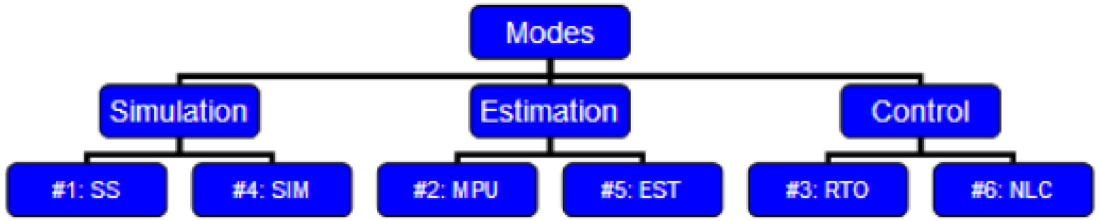
```
Type: Integer, Input
Default Value: 3
Description: Model solution mode
             1=ss, Simulation (steady state)
             2=mpu, Model parameter update (steady state)
             3=rto, Real time optimization (steady state)
             4=sim, Dynamic simulation (simultaneous)
             5=est, Dynamic estimation (simultaneous)
             6=ctl, Dynamic control (simultaneous)
             7=sqs, Dynamic simulation (sequential)
             8=sqe, Dynamic estimation (sequential)
             9=sqo, Dynamic control (sequential)
```

Non-linear Model Core

The core of all modes is the non-linear model. Each mode interacts with the nonlinear model to receive or provide information. There are 6 modes of operation for the APMonitor software.

1. Steady-state simulation (SS)
2. Model parameter update (MPU)
3. Real-time optimization (RTO)
4. Dynamic simulation (SIM)
5. Moving horizon estimation (EST)
6. Nonlinear control / dynamic optimization (CTL)
7. Sequential dynamic simulation (SQS)
8. Sequential dynamic estimation (SQE)
9. Sequential dynamic optimization (SQO)

Modes 1-3 are steady state modes with all derivatives set equal to zero. Modes 4-6 are dynamic modes where the differential equations define how the variables change with time. Modes 7-9 are the same as 4-6 except the solution is performed with a sequential versus a simultaneous approach. Each mode for simulation, estimation, and optimization has a steady state and dynamic option.



APMonitor FV, MV, SV, CV Classification

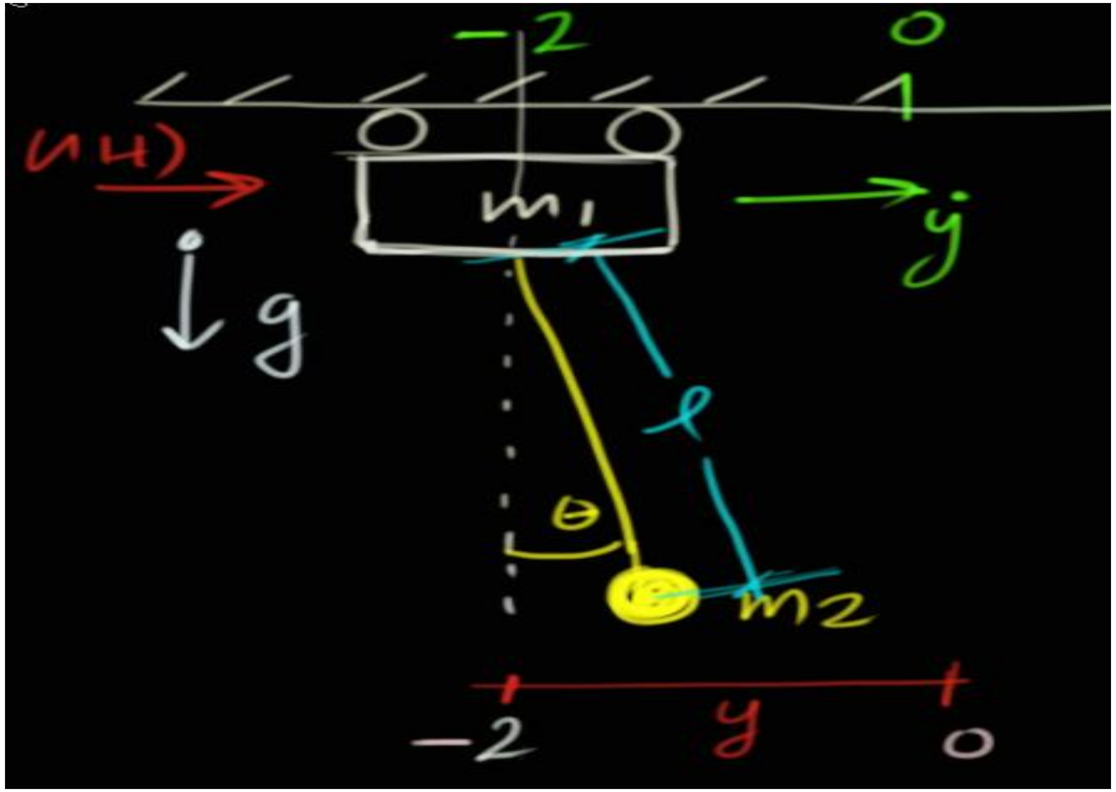
Parameters and variables can be specified as feedforward variables (FV), manipulated variables (MV), state variables (SV), and controlled variables (CV). These are 4 basic types of variables that are used in the APMonitor simulations. The FV and MV types are specified from the available declared *parameters*. These are values that are fixed in steady-state calculations. The SV and CV types are specified from the available declared *variables*.

- Parameters list in model
  - Feedforward Variables (FV)
  - Manipulated Variables (MV)
- Variables list in model
  - State Variables (SV)
  - Controlled Variables (CV)

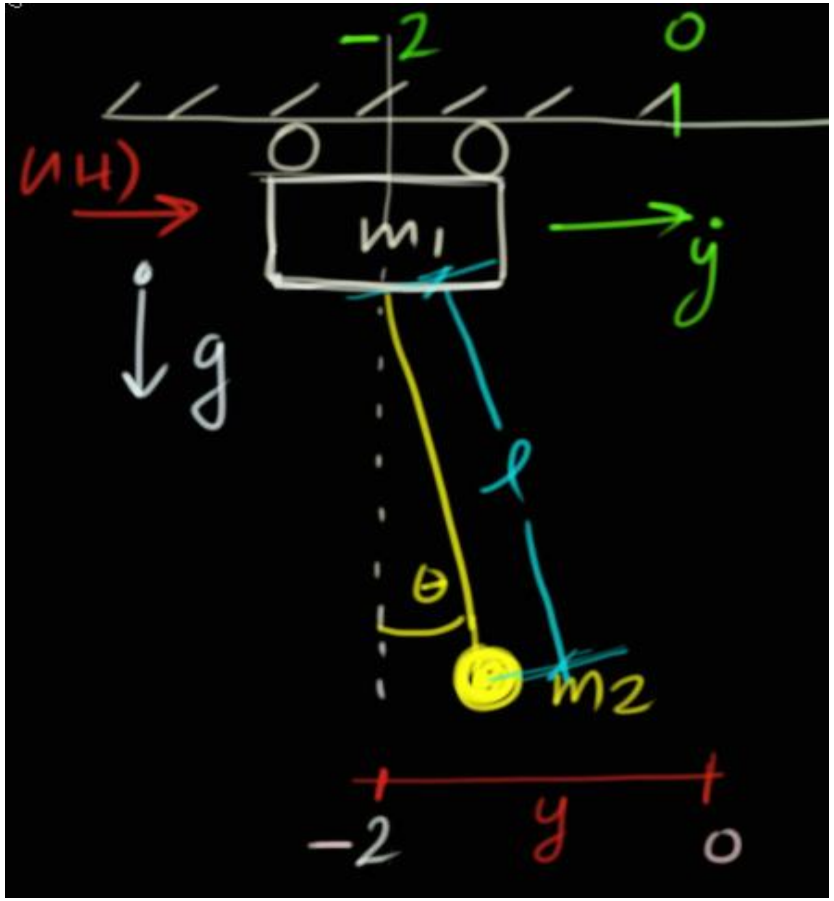
Linear MPC Overhead-crane with a State Space Model

Design a model predictive controller for an overhead crane with a pendulum mass. Meet specific control objectives by tuning the controller and using the state space model of the crane system. Simulate and optimize the pendulum system.

State Space Model



State Space Model



Input: $u(t)$ force applied to the cart

Outputs: y cart position

θ angle of the pendulum

$$x = \begin{bmatrix} \Delta y \\ \Delta v \\ \Delta\theta \\ \Delta\dot{\theta} \end{bmatrix}$$

$$y = \begin{bmatrix} \Delta y \\ \Delta\theta \end{bmatrix}$$

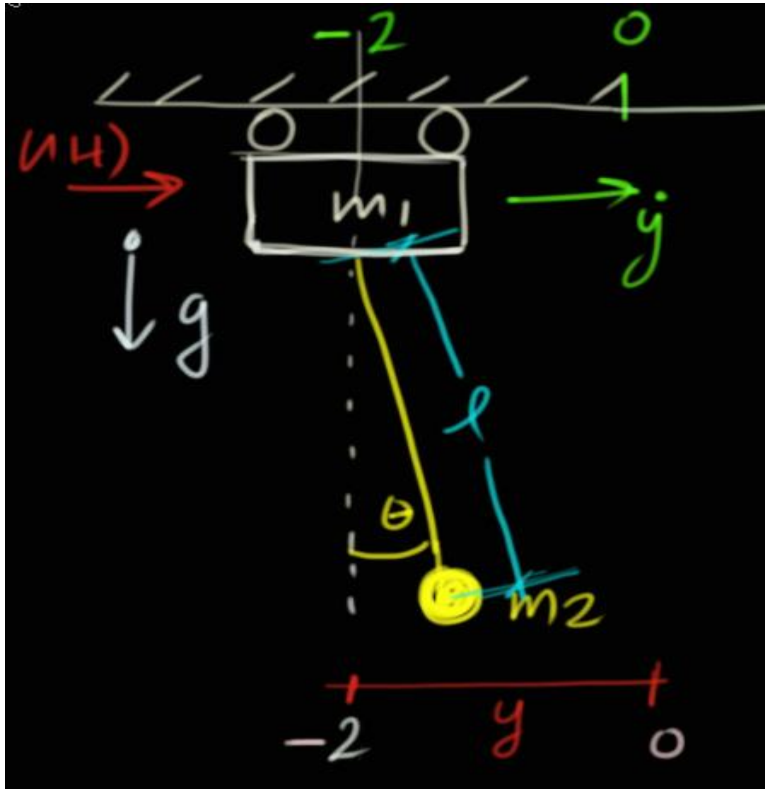
Δy Cart's position

$\Delta v = \Delta\dot{y}$ velocity of the cart

$\Delta\theta$: Angle of the pendulum

$\Delta\dot{\theta} = \Delta q$ Rate of the angle change

State Space Model



$$\dot{x} = \begin{bmatrix}\Delta\dot{y}\\ \Delta\dot{v}\\ \Delta\dot{\theta}\\ \Delta\ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & C_5 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -C_8 & 0\end{bmatrix}\begin{bmatrix}\Delta y\\ \Delta v\\ \Delta\theta\\ \Delta\dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ C_6\\ 0\\ -C_7\end{bmatrix}\Delta u(t)$$

$$y = \begin{bmatrix}\Delta y\\ \Delta\theta\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\Delta y\\ \Delta v\\ \Delta\theta\\ \Delta\dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}\Delta u(t)$$

State Space Model

$$\dot{x} = \begin{bmatrix} \Delta\dot{y} \\ \Delta\dot{v} \\ \Delta\dot{\theta} \\ \Delta\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & C_5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -C_8 & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta v \\ \Delta\theta \\ \Delta\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ C_6 \\ 0 \\ -C_7 \end{bmatrix} \Delta U(t)$$

$$y = \begin{bmatrix} \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta v \\ \Delta\theta \\ \Delta\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Delta U(t)$$

$$C_1 = m_2 l \,;\quad C_2 = m_1 + m_2$$

$$C_3 = m_2 g l \,;\quad C_4 = J + m_2 l$$

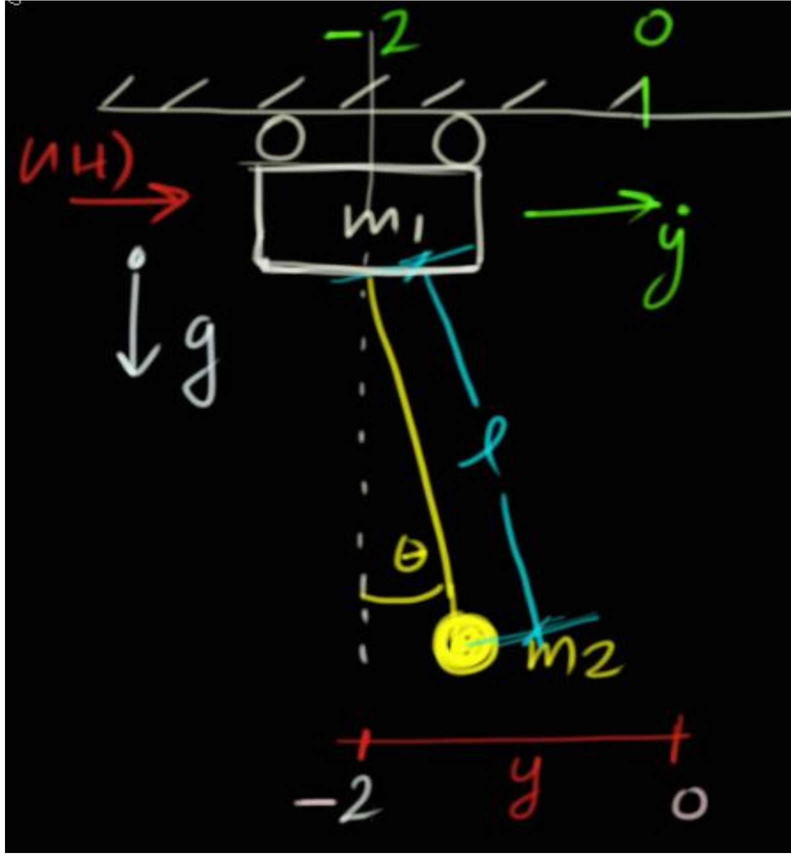
$$C_5 = \frac{C_1 C_3}{C_2 C_4 - C_1^2} \,;\quad C_6 = \frac{C_4}{C_2 C_4 - C_1^2}$$

$$C_7 = \frac{C_1}{C_2 C_4 - C_1^2} \,;\quad C_8 = \frac{C_3 C_2}{C_2 C_4 - C_1^2}$$

Note: $\dot{y} = v$

$\dot{\theta} = q$

Linear MPC Overhead-crane with a State Space Model



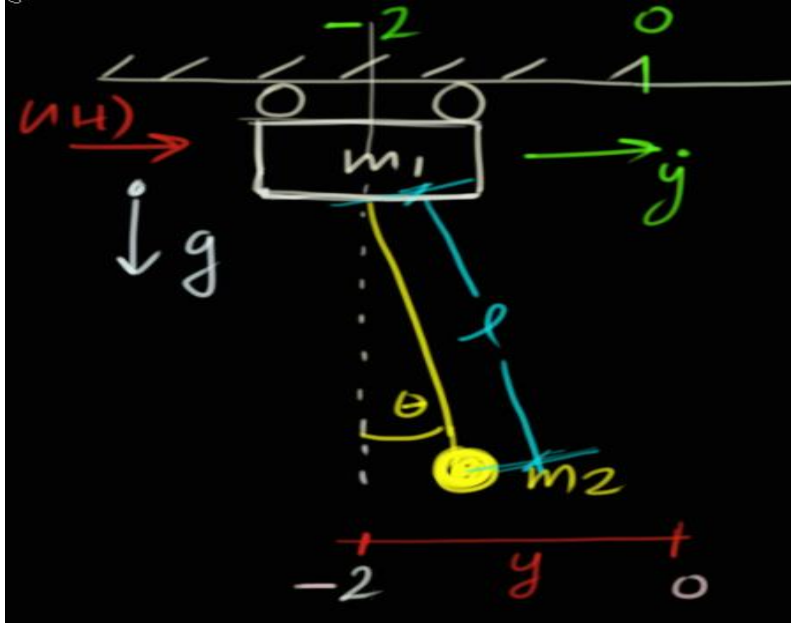
- The objective of the controller is to **adjust the force on the cart to move the pendulum mass to a new final position**.
- Ensure that initial and final velocities and angles of the pendulum are zero.

- The position of the pendulum mass is i**nitially at -2 and it is desired to move it to the new position of 0 within 6 seconds.**
- Demonstrate controller performance with changes in the pendulum position and that the final pendulum mass remains at the final position without oscillation.

---Evaluate the results when you increase the pendulum's mass---

https://apmonitor.com/do/index.php/Main/ControlTypes

Using Notepad++



Parameters

m1 = 10 ! kg
m2 = 1 ! kg
l=1 ! m
g=10 ! m/s^2
J=10 ! kg. m^2 Inertia moment
u = 0 ! degree of freedom for optimizer ---Force applied to the cart
final ! loaded in new_pendulum.csv time vs. final values

Variables

y = -2
v = 0
theta = 0
q = 0

Intermediates

c1=m2*l
c2=m1+m2
c3=m2*g*l
c4=J+m2*l
c5=c1*c3/(c2*c4-(c1^2))
c6=c4/(c2*c4-(c1^2))
c7=c1/(c2*c4-(c1^2))
c8=c3*c2/(c2*c4-(c1^2))

$$C_1 = m_2 l \; ; \; C_2 = m_1 + m_2$$

$$C_3 = m_2 g l \; ; \; C_4 = J + m_2 l$$

$$C_5 = \frac{C_1 C_3}{C_2 C_4 - C_1^2} \; ; \; C_6 = \frac{C_4}{C_2 C_4 - C_1^2}$$

$$C_7 = \frac{C_1}{C_2 C_4 - C_1^2} \; ; \; C_8 = \frac{C_3 C_2}{C_2 C_4 - C_1^2}$$

Equations

$y = v
$v = c5*theta + c6*u
$theta = q
$q = -c8*theta - c7*u

Objective Function

minimize final * (y^2 + v^2 + theta^2 + q^2)   ! custom objective function
minimize 0.001 * u^2 ! minimizing motor movement (force) with a small penalty of 0.001

Asst. Prof. Claudia F. YAŞAR YTU Control and Automation Engineering Department-2023

Using Notepad++

```
Parameters
 m1 = 10 ! kg
 m2 = 1 ! kg
 l=1  ! m
 g=10  ! m/s^2
 J=10 ! kg. m^2  Inertia moment
 u = 0 ! degree of freedom for optimizer ---Force applied to the cart
 final ! loaded in new_pendulum.csv  time vs. final values
 Variables
 y = -2
 v = 0
 theta = 0
 q = 0
Intermediates
 c1=m2*l
 c2=m1+m2
 c3=m2*g*l
 c4=J+m2*l
 c5=c1*c3/(c2*c4-(c1^2))
 c6=c4/(c2*c4-(c1^2))
 c7=c1/(c2*c4-(c1^2))
 c8=c3*c2/(c2*c4-(c1^2))
 Equations
 $y = v
 $v = c5*theta + c6*u
 $theta = q
 $q = -c8*theta - c7*u
 Objective Function
 minimize final * (y^2 + v^2 + theta^2 + q^2)    ! custom objective function
 minimize 0.001 * u^2  ! minimizing motor movement (force) with a small penalty of 0.001
```

$$\dot{x} = \begin{bmatrix}\Delta\dot{y}\\ \Delta\dot{v}\\ \Delta\dot{\theta}\\ \Delta\ddot{\theta}\end{bmatrix} = \begin{bmatrix}0&1&0&0\\0&0&C_5&0\\0&0&0&1\\0&0&-C_8&0\end{bmatrix}\begin{bmatrix}\Delta y\\ \Delta v\\ \Delta\theta\\ \Delta\dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\C_6\\0\\-C_7\end{bmatrix}\Delta u(t)$$

$$y = \begin{bmatrix}\Delta y\\ \Delta\theta\end{bmatrix} = \begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}\Delta y\\ \Delta v\\ \Delta\theta\\ \Delta\dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}\Delta u(t)$$

$$dy = v$$

$$dv = C_5\theta + C_6 u$$

$$d\theta = \dot{\theta}$$

$$d\dot{\theta} = dq = -C_8\theta - C_7 u$$