

Project Team:

Tyler Chesney
Keegan Penso
Quincy Owyang
Mojit Bhardwaj
Jack Russo

MECA 482

Furuta Pendulum

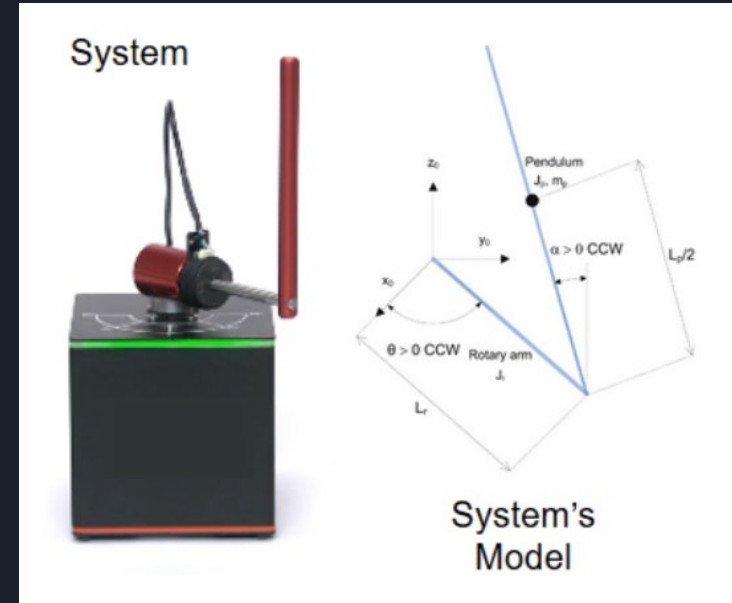
Spring 22'



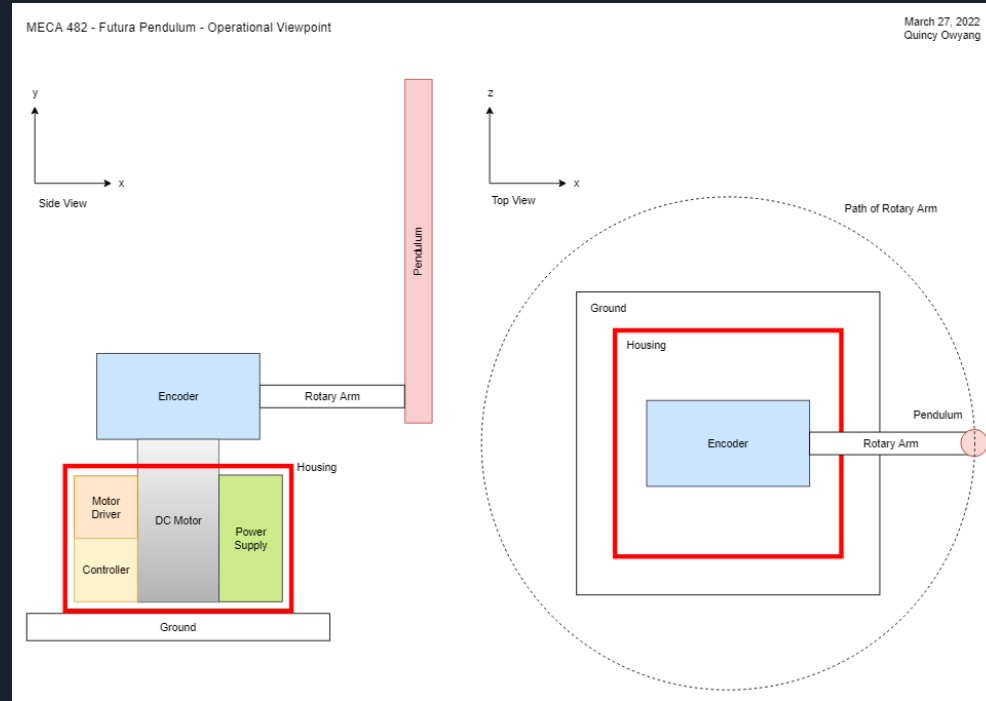
Introduction

The Furuta pendulum, or rotational inverted pendulum, consists of a driven arm which rotates in the horizontal plane and a pendulum attached to that arm which is free to rotate in the vertical plane.

Our goal is to design a system that is able to stabilize itself in the upright-vertical position.

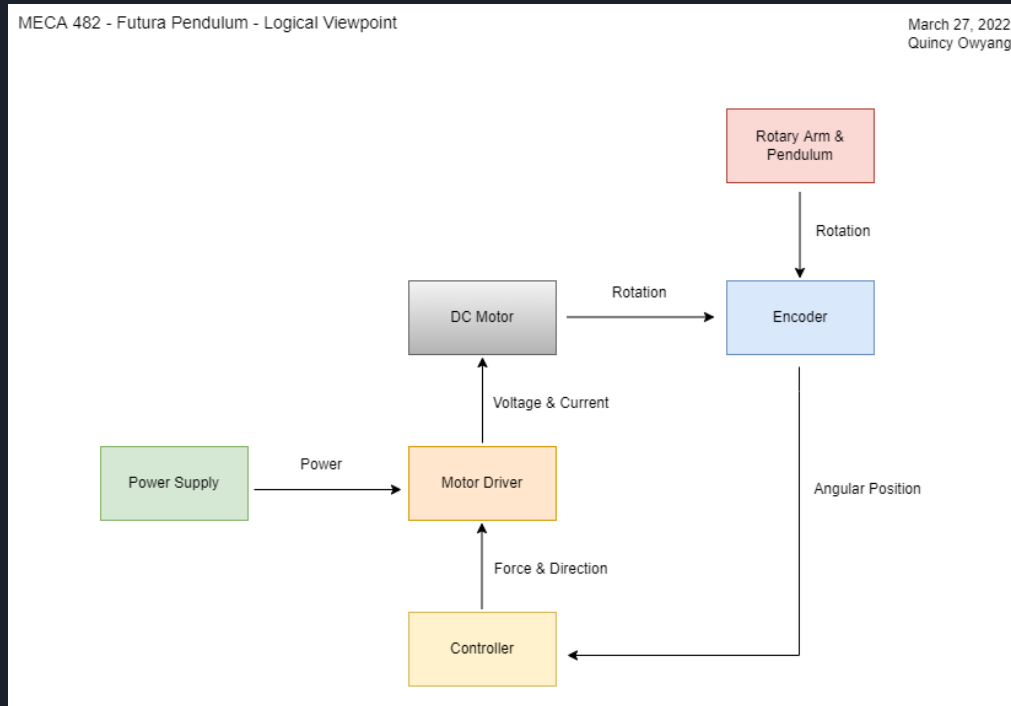


Operational Viewpoint



This viewpoint shows the physical space and movements of components of the pendulum

Operational Viewpoint



This logical viewpoint shows the interactions between components in the system.

System Model

L_r - Arm length

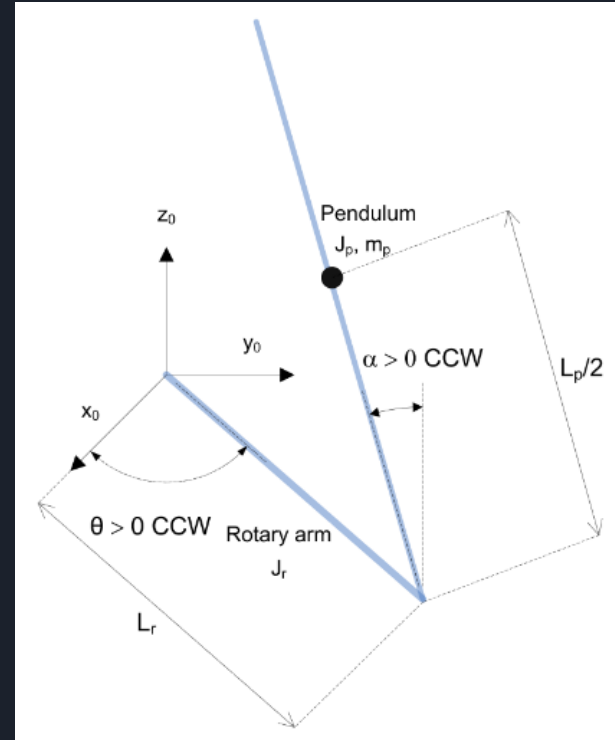
J_r - Arm Moment of Inertia

θ - Arm Angle

L_p - Pendulum Length

J_p - Pendulum Moment of Inertia

α - Pendulum angle





Linearized Equations of Motion

$$(m_p L_r^2 + J_r) \ddot{\theta} - \frac{1}{2} m_p L_p L_r \ddot{\alpha} = \tau - B_r \dot{\theta}$$

$$-\frac{1}{2} m_p L_p L_r \ddot{\theta} + \left(J_p + \frac{1}{4} m_p L_p^2 \right) \ddot{\alpha} - \frac{1}{2} m_p L_p g \alpha = -B_p \dot{\alpha}$$

State Space Equations

$$A = \frac{1}{J_T} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{4}m_p L_p^2 L_r g & -\left(J_p + \frac{1}{4}m_p L_p^2\right) B_r & -\frac{1}{2}m_p L_p L_r B_p \\ 0 & \frac{1}{2}m_p L_p g (J_r + m_p L_r^2) & \frac{1}{2}m_p L_p L_r B_r & -(J_r + m_p L_r^2) B_p \end{bmatrix}$$

$$B = \frac{1}{J_T} \begin{bmatrix} 0 \\ 0 \\ J_p + \frac{1}{4}m_p L_p^2 \\ \frac{1}{2}m_p L_p L_r \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

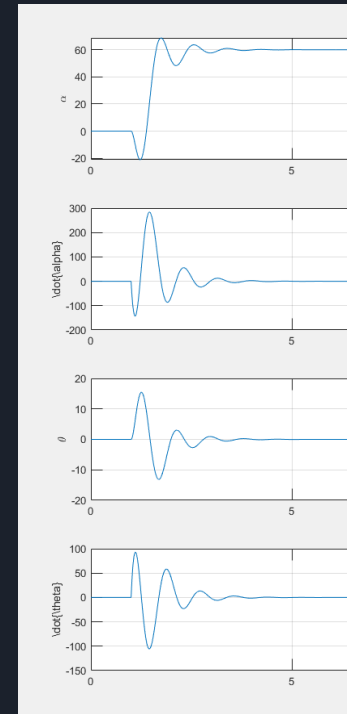
$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

MATLAB Code

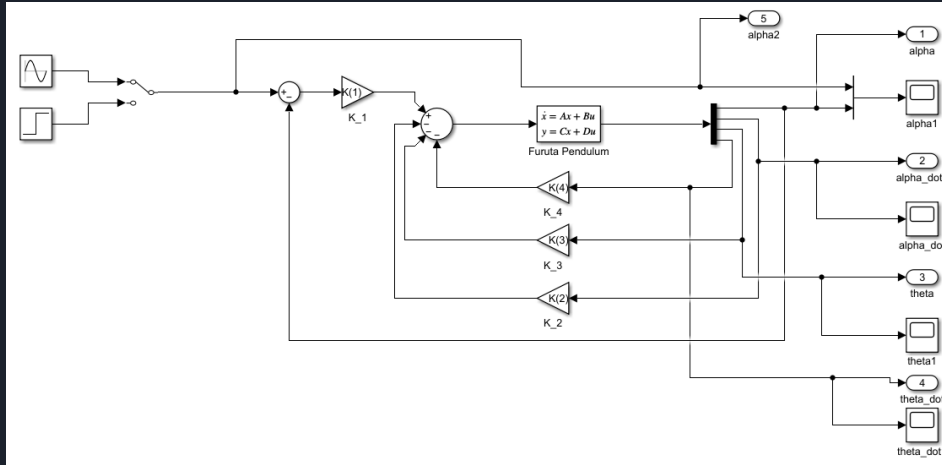
```

1  %% List of parameters
2
3  L_r = 0.36; %length of radial arm
4  L_p = 0.58; %length of the pendulum
5  m_p = 0.31; %mass of pendulum (kg)
6  m_r = 0.53; %mass of the radial arm.
7  B_p = 0.025; %damping of the pendulum.
8  B_r = 0.01; %damping of the radial arm.
9  fre = 0.1; %the frequency of the sin wave input
10 |
11 J_r=((m_r+m_p)*L_r^2)/3;
12 J_p=(m_p*L_p^2)/3;
13 J_T=J_p*m_p*L_r^2+J_r*m_p*L_p^2;
14 g=9.81; % gravity
15
16 %% Matrices
17
18 A=[0 0 1 0;
19   0 0 0 1;
20   0 0.25*m_p^2*L_p^2*L_r*g -(J_p+0.25*m_p*L_p^2)*B_r -0.5*m_p*L_p*L_r*B_p;
21   0 0.5*m_p*L_p*g*(J_r+m_p*L_r^2) 0.5*m_p*L_p*L_r*B_r -(J_r+m_p*L_r^2)*B_p];
22
23 B=1/J_T*[0;0;J_p+0.25*m_p*L_p^2;0.5*m_p*L_p*L_r];
24
25 C=[1 0 0 0;
26   0 1 0 0];
27 D=[0;0;0;0];
28
29 P=[-17.1 8.34 -2.87 0];
30

```



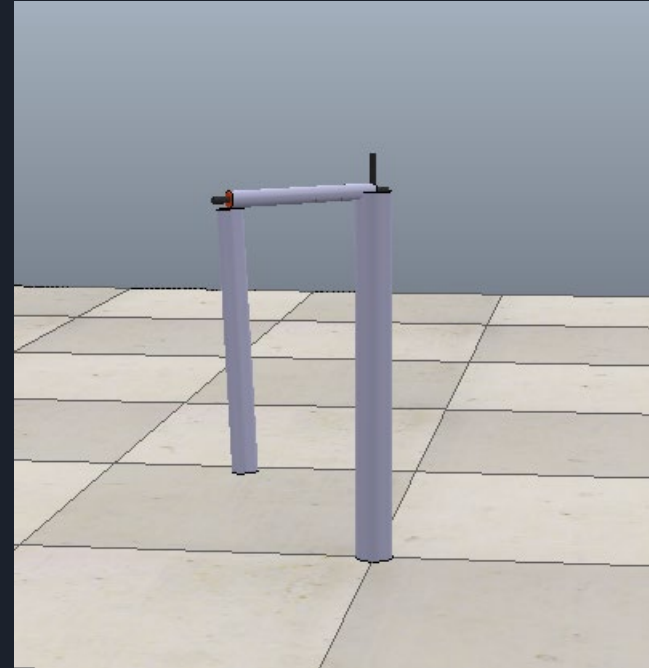
SIMULINK Diagram



Vikash Gupta (2022). Full State Feedback of Furuta Pendulum
(<https://www.mathworks.com/matlabcentral/fileexchange/25585-full-state-feedback-of-furuta-pendulum>), MATLAB Central File Exchange. Retrieved May 19, 2022.

Simulation Results

```
Child script "/Frame"
1  -- MECA 482 --
2
3  local Pendulum, RotatingArm, Frame, Revolute, Revolute2;
4  local theta_correct, alpha_correct, thetadot, alphadot;
5
6  function sysCall_init()
7      -- do some initialization here
8      init_parameters();
9      Pendulum = sim.getObjectHandle("Pendulum");
10     RotatingArm = sim.getObjectHandle("RotatingArm");
11     Frame = sim.getObjectHandle("Frame");
12     Revolute1 = sim.getObjectHandle("Revolute1");
13     Revolute2 = sim.getObjectHandle("Revolute2");
14
15     --Initializing time and state
16     init_measure_state();
17 end
18
19 function sysCall_init()
20     -- do some initialization here
21     --Setting model parameters
22
23     --Getting objects handles
24
25     --Initializing time and thetadot
26
27 end
28
29 function sysCall_actuation()
30     -- put your actuation code here
31     electro_actuation();
32     --No limit to the velocity
33     velocity=sign(torque)*1000;
34     -- Actuation Joint Setting
35     -- The setJointForce receive the abs of a torque
36     sim.setJointForce(Revolute1,math.abs(torque));
37     -- We don't put any limit to velocity since we control the joint with torque
38     sim.setJointTargetVelocity(Revolute1, velocity);
39 end
40
```



Due to complications interfacing MATLAB and CoppeliaSim, an accurate simulation could not be produced



References

- [1] Vikash Gupta (2019). Full State Feedback of Furuta Pendulum (<https://www.mathworks.com/matlabcentral/fileexchange/25585-full-state-feedback-of-furuta-pendulum>), MATLAB Central File Exchange. Retrieved May 19, 2022.
- [2] Hernández-Guzmán Victor Manuel and Silva-Ortigoza Ramón, Automatic control with experiments. Cham, Switzerland: Springer, 2019.
- [3] Jacob Apkarian, Michel Lévis and Hakan Gurocak, SRV02 Rotary Servo Base Unit User Manual. Ontario, Canada: Quanser, 2011.