MECA 482 F21 Group 5

Final Design Project

Inertia Wheel Pendulum

12/17/2021

Project Team:
Sam Kelly
Sean Murray
Victor Alvarez
Xavier Andrade Rubio
John Voter

Introduction

- The Inertia Wheel Pendulum is a common project used for teaching purposes of nonlinear control systems

- This system has two degrees of freedom, one being the entire pendulum the second being the inertia wheel

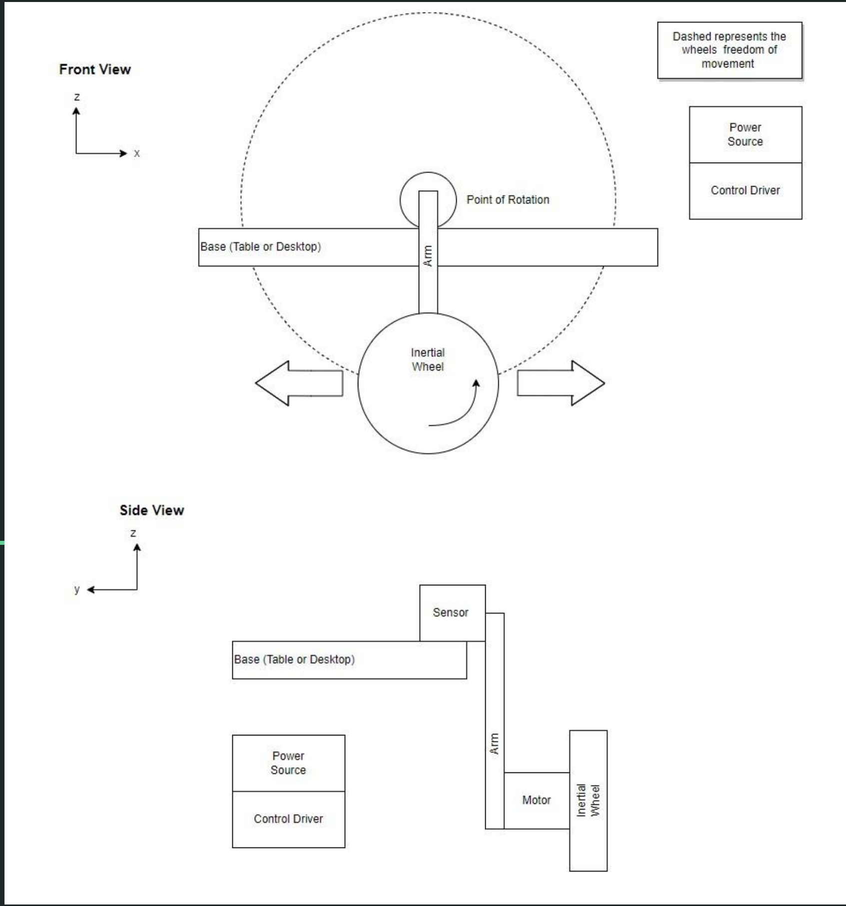
- Our goal is to design a system able that is able to stabilize itself in the upright-vertical position through only the inertia of the wheel on the end of the pendulum

Operational Viewpoint

This viewpoint shows the physical location and connections between components along with their movement in relation to one another.

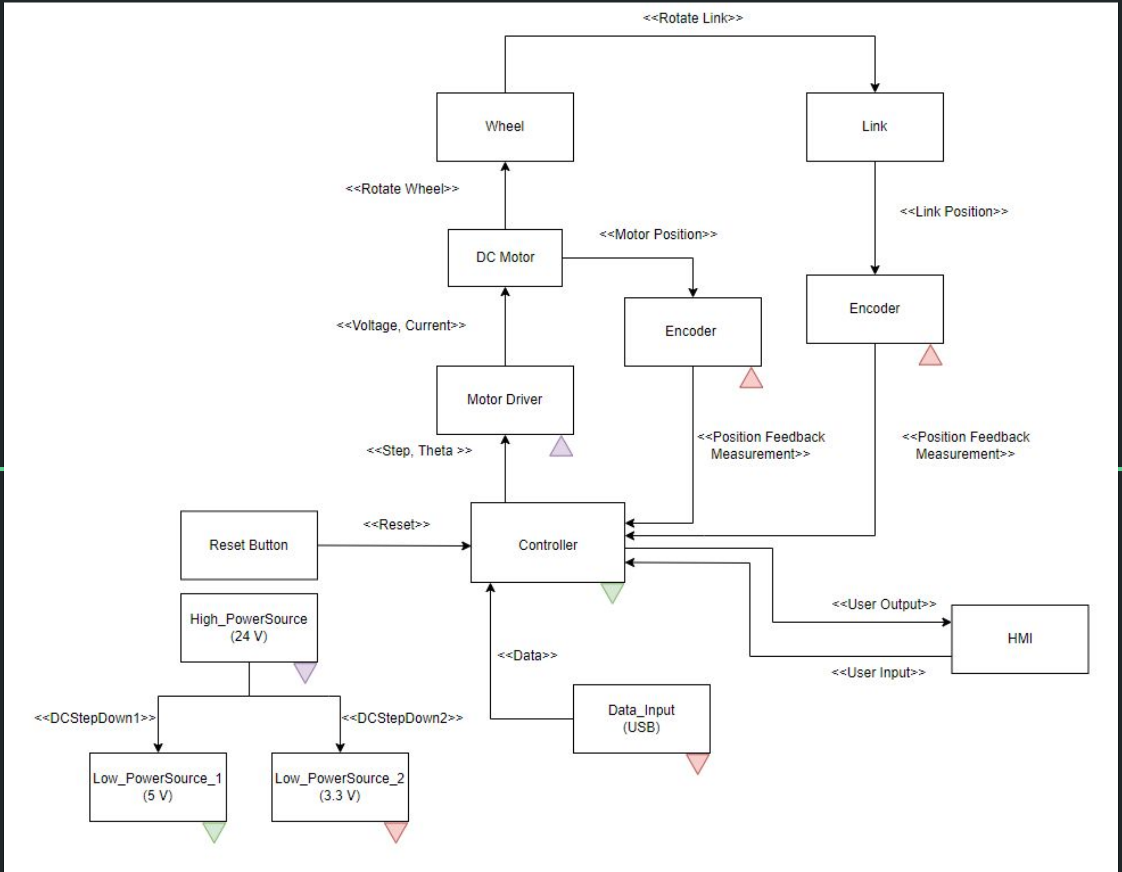
Front view shows if someone were to look head on the system at rest.

Side view shows the system at operational equilibrium from the side plane.

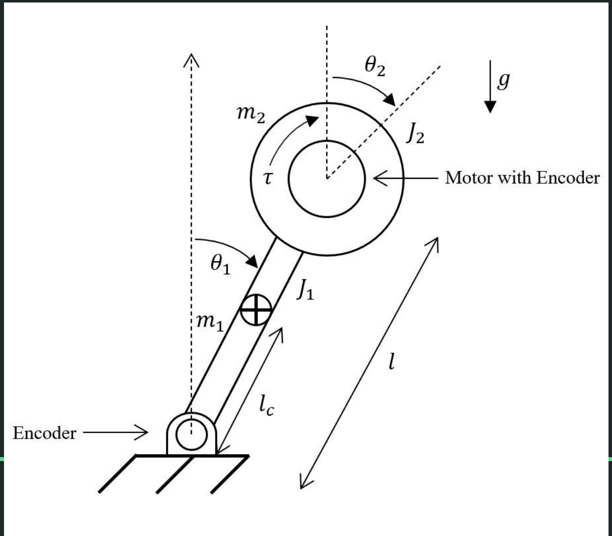


Logical Functional Viewpoint

This diagram serves to display the interaction between components within the system. It would also be useful as a wiring diagram if a physical circuit was to be built.



System Model



Free Body Diagram

- τ is torque applied at the wheel.
- lc is the location of the pendulum's center of mass.
- l is the pendulum's length.
- J1 is the pendulum' inerti when rotating around its center of mass.
- J2 is the wheel inertia (plus the motor rotor inertia).
- g is the gravity acceleration.

State Space Representation

$\dot{\theta} = A\theta + Bu$	Eq. 1
$y = C\theta$	Eq. 2
$\dot{\theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$	Eq. 3
$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$	Eq. 4
$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\overline{m}g}{J_2(1 - m_1 l_c^2 + m_2 l^2 + J_1 + J_2)} & 0 & 0 \\ \frac{-\overline{m}g}{J_2(1 - m_1 l_c^2 + m_2 l^2 + J_1 + J_2)} & 0 & 0 \end{bmatrix}$	Eq. 5
$B = \begin{bmatrix} 0 \\ \frac{1}{J_2} + \frac{m_1 l_c^2 + m_2 l^2 + J_1 + J_2}{J_2(1 - m_1 l_c^2 + m_2 l^2 + J_1 + J_2)} \\ \frac{-m_1 l_c^2 + m_2 l^2 + J_1 + J_2}{J_2(1 - m_1 l_c^2 + m_2 l^2 + J_1 + J_2)} \end{bmatrix}$	Eq. 6

MECA 482 F21

MATLAB Code

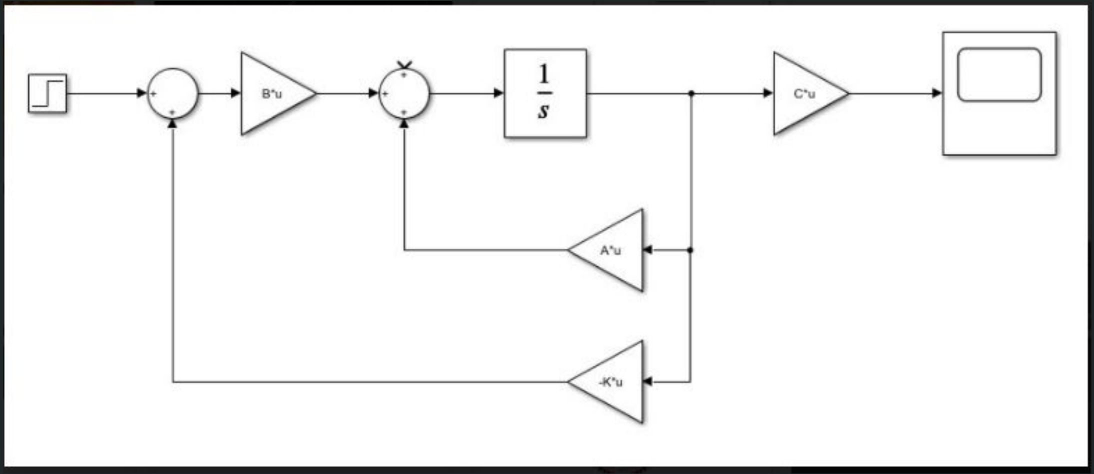
```
% Define system matrices
g = 9.81
r = 0.5
m1 = 2
m2 = 1
l = 1
lc = 0.25*l
mbar = m1*lc+m2*l
J1 = (1/3)*m1*l^2
J2 = (1/2)*m2*r^2

A = [0 1 0; mbar*g/(J2*(1-m1*lc^2+m2*l^2+J1+J2)) 0 0; -mbar*g/(J2*(1-m1*lc^2+m2*l^2+J1+J2)) 0 0]
B = [0; 1/J2+(m1*lc^2+m2*l^2+J1+J2)/(J2*(1-m1*lc^2+m2*l^2+J1+J2)); -(m1*lc^2+m2*l^2+J1+J2)/(J2*(1-m1*lc^2+m2*l^2+J1+J2))]
C = [1 0 0];
D = 0;

check1 = A(2,1)
check2 = A(3,1)
check3 = B(2,1)
check4 = B(3,1)
% Create state space object
sys = ss(A,B,C,D);

% Check open-loop eigenvalues
E = eig(A);

% Desired closed-loop eigenvalues
P = [0, -0.5, -1.5];

% Solve for K using pole placement
K = place(A,B,P);

% Check for closed-loop eigenvalues
Acl = A-B*K;
Ecl = eig(Acl);
detAcl = det(Acl)

% Closed-loop system
syscl = ss(Acl, B, C, D);

Kr = 1/dcgain(syscl);
syscl_scaled = ss(Acl, B*Kr, C, D);

% Step response of the system
%step(syscl);
step(syscl_scaled);
%step(sys)
```

MECA 482 F21

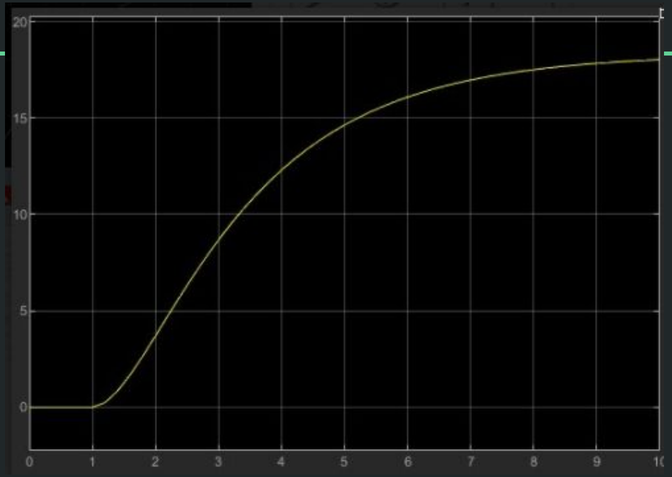
Simulink Block Diagrams

Closed-Loop Feedback

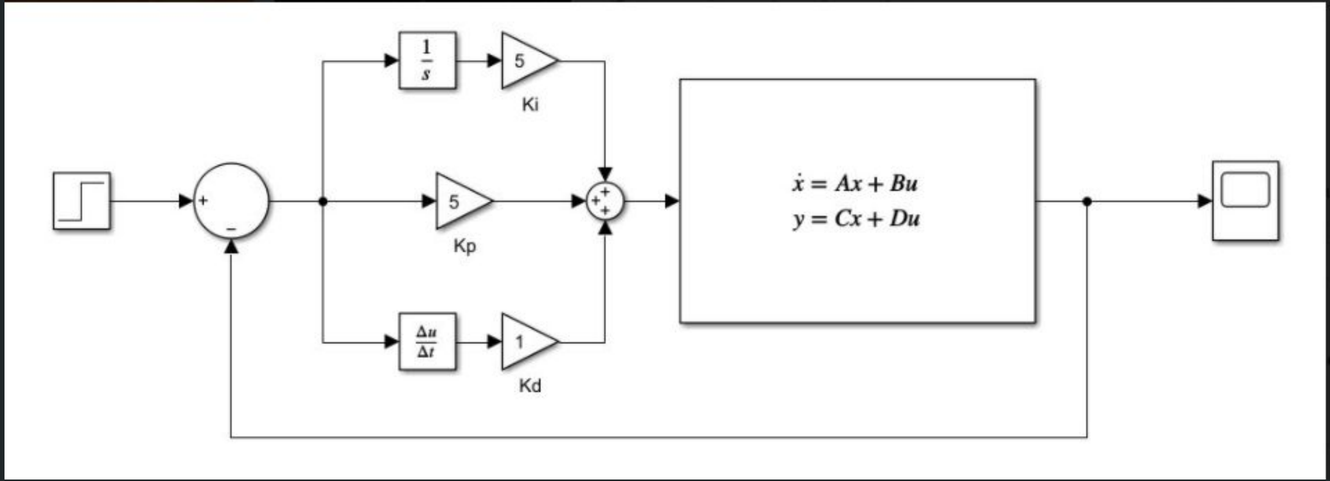
Controller



Corresponding Step Response of the System

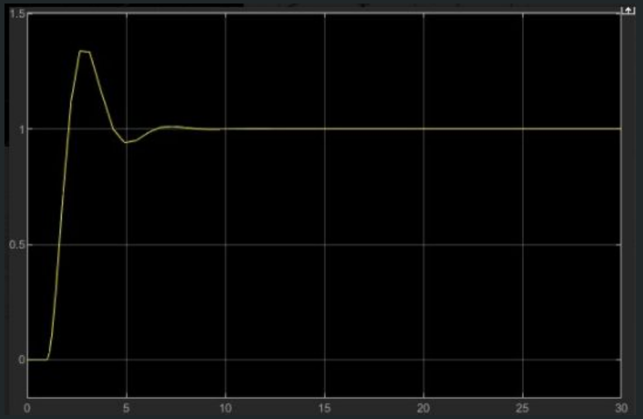


Simulink Block Diagrams

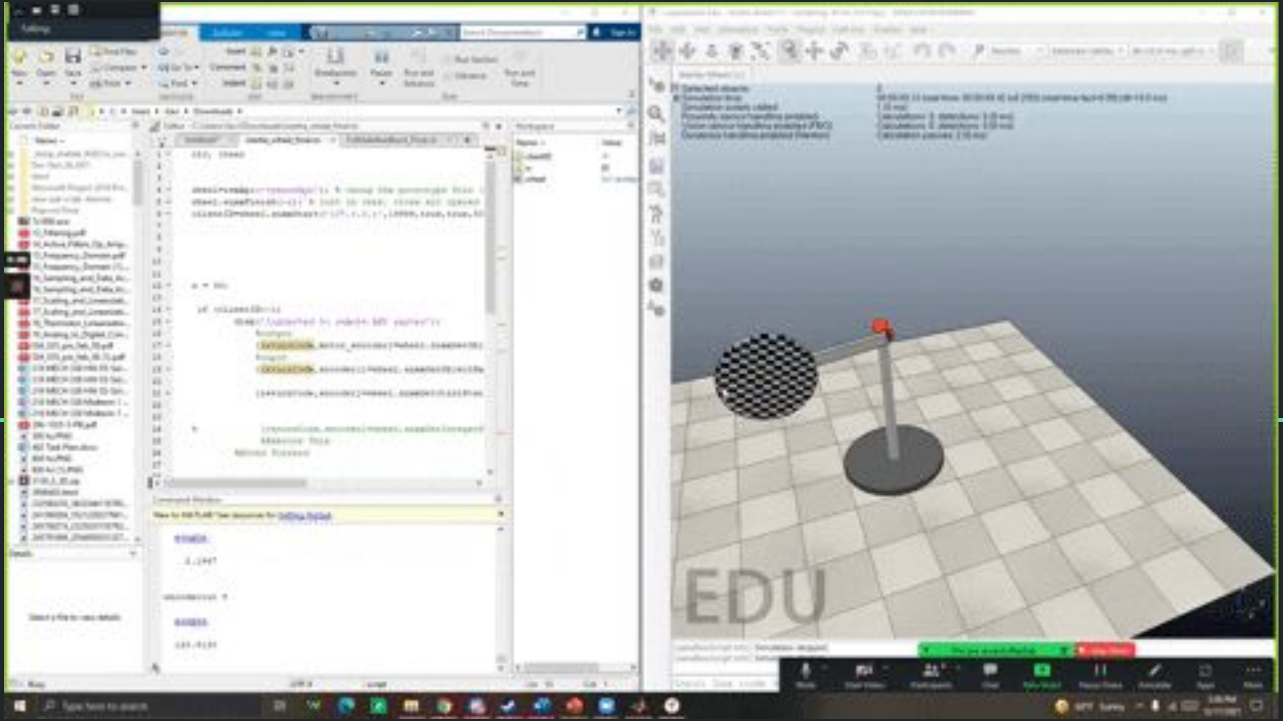


Proportional Integral Derivative Controller

Corresponding Response of the Above System (Not used in CoppeliaSim)



CoppeliaSim Simulation



References

[1] Victor Manuel Hernández-Guzmán and Ramón Silva-Ortigoza
"Automatic Control with Experiments"
[2] Norman Nise
"Control Systems Engineering", 7th Edition