FURUTA PENDULUM

Major Task

Name	ID
Mahmoud Yasser Mahmoud	2001200
Mustafa Mahgoub Sobhy	2001131
Hussein Hamed Hussein	2001767
Hossam Ahmed Selim	2001830

Faculty of Engineering Ain Shams University

Hybrid Control Systems (MCT411s)

Table Of Contents

List of Figures

INTRODUCTION

Overview:

The Furuta Pendulum, an archetypal system in control theory, represents a fascinating challenge due to its inherent instability. This system, consisting of a pendulum attached to a rotary arm, requires careful application of advanced control strategies to achieve stability in its upright position.

Motivation:

Control systems play a critical role in modern engineering applications, from robotics to aerospace systems. The Furuta Pendulum serves as an excellent testbed for honing control techniques, enabling engineers to explore methods such as pole placement, robust control, and disturbance rejection in a practical, hands-on manner.

Objectives:

This project aims to:

- Develop and implement a controller capable of stabilizing the pendulum at its upright position.
- Ensure the system remains stable despite external disturbances and inherent nonlinearities.
- Employ MATLAB/SIMULINK for modeling, simulation, and Hardware-in-the-Loop (HIL) validation to bridge the gap between theory and practice.

SYSTEM OVERVIEW

The Furuta Pendulum Design:

The Furuta Pendulum, also referred to as the rotary inverted pendulum, is a classic example of a nonlinear, underactuated system. It consists of two main components:

- **Rotary Arm:** A horizontal arm driven by a motor, capable of controlling rotational motion.
- **Pendulum:** A vertical rod attached to the end of the rotary arm, free to swing within the vertical plane.

The rotary arm's angular position (θ_1) is monitored and controlled using a motor equipped with an encoder. Simultaneously, a second encoder measures the pendulum's angular position (θ_2). The interplay between these two components creates a system that is both highly dynamic and challenging to stabilize, particularly when the pendulum is in its unstable upright position.

Functional Requirements:

The primary aim is to stabilize the pendulum in its upright position ($\theta_2=0$) through precise manipulation of the rotary arm's motion. This involves addressing the following challenges:

1. **Upright Stability:** Ensuring the pendulum maintains balance at $\theta_2=0$.
2. **Disturbance Rejection:** Countering external forces and inherent system nonlinearities that could destabilize the pendulum.

3. **Fast Response:** Achieving stability quickly, with minimal overshoot or oscillations, to ensure robust and reliable system performance.

METHODOLOGY

Modeling the System:

Form paper: https://www.researchgate.net/publication/228421122_On_the_Dynamics_of_the_Furuta_Pendulum

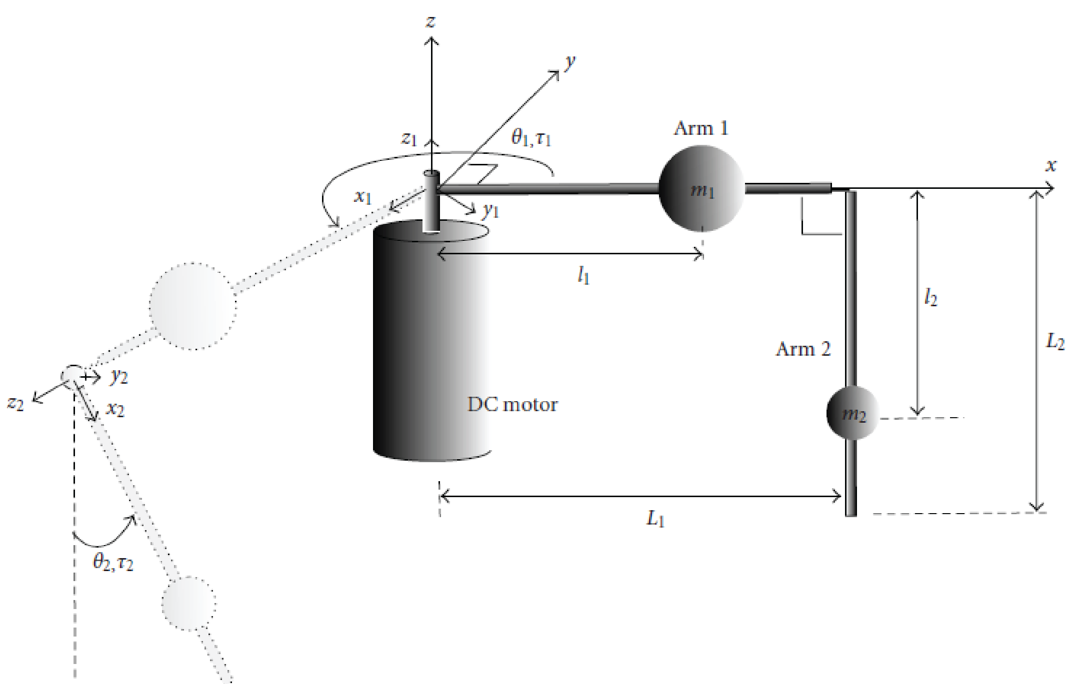


Figure 1:Schematic of the single rotary inverted pendulum system

Actuation and Dynamics:

The Furuta Pendulum system incorporates a DC motor that applies torque (τ1) to the first arm (Arm 1). While Arm 1 is actively controlled, the connection between Arm 1 and Arm 2 is passive, allowing Arm 2 to rotate freely.

The physical characteristics of the system are described as follows:

- **Arm Dimensions:** The lengths of Arm 1 and Arm 2 are L1 and L2, respectively.
- **Mass Distribution:** The masses of Arm 1 and Arm 2, denoted as m1 and m2, are concentrated at points located l1 and l2 from their respective axes of rotation.
- **Inertia Properties:** Each arm has a moment of inertia about its center of mass, represented as J1 for Arm 1 and J2 for Arm 2.

Damping

Rotational joints within the system experience viscous damping, characterized by the coefficients b1 and b2:

- b1: Damping due to motor bearings acting on Arm 1.
- b2: Damping introduced by the pin coupling between Arm 1 and Arm 2.

Moments of Inertia

The system's moments of inertia are determined using the parallel axis theorem:

1. The total moment of inertia of Arm 1 about its pivot is:

$$\widehat{J_1} = J_1 + m_1 {l_1}^2$$

2. The total moment of inertia of Arm 2 about its pivot is:

$$\widehat{J_2} = J_2 + m_2 {l_2}^2$$

When Arm 2 is in its equilibrium position (hanging vertically downward), the motor rotor experiences a combined moment of inertia:

$$\widehat{J_0} = \widehat{J_1} + m_2 {l_1}^2 = J_1 + m_1 {l_1}^2 + m_2 {l_1}^2$$

Torque on motor:

$$\tau_1 = \tau_m \times N$$

$$\omega_o = \frac{\omega_m}{N}$$

$$\tau_m = k_t \times I$$

$$\tau_o = N \times k_t \times I$$

$$\tau_1 = N \times k_t \times \left(\frac{V - K_e \times \omega_m}{\sqrt{R^2 + (\omega \times L)^2}}\right)$$

$$\tau_1 = N \times k_t \times \left(\frac{V - K_e \times \omega_o \times N}{\sqrt{R^2 + (\omega \times L)^2}}\right)$$

$$\tau_1 = N \times k_t \times \left(\frac{V - K_e \times \dot{\theta}_1 \times N}{\sqrt{R^2 + (\omega \times L)^2}}\right)$$

Where, N is gear ratio.

And $l_2 = L_2/2$

State space:

$$\text{term}_a = J_0 J_2 - m_2^2 l_1^2 l_2^2$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{g m_2 l_2 l_1}{\text{term}_1} - \dfrac{b_o J_2}{\text{term}_1} - \dfrac{\left(\dfrac{J_2}{\text{term}_1} \cdot \dfrac{N K_t}{\sqrt{R^2 + (wL)^2}}\right)^2 N^2 K_t^2}{\sqrt{R^2 + (wL)^2}} & \dfrac{-b_o m_2 l_2 l_1}{\text{term}_1} \\ 0 & \dfrac{g m_2 l_2 J_o}{\text{term}_1} - \dfrac{b_o m_2 l_2 l_1}{\text{term}_1} - \dfrac{\left(\dfrac{m_2 l_2 l_1}{\text{term}_1} \cdot \dfrac{N K_t}{\sqrt{R^2 + (wL)^2}}\right)^2 N^2 K_t^2}{\sqrt{R^2 + (wL)^2}} & -\dfrac{b_o J_o}{\text{term}_1} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{J_2}{\text{term}_1} \cdot \dfrac{N K_1}{\sqrt{R^2 + (wL)^2}} & \dfrac{m_2 l_2 l_1}{\text{term}_1} \\ \dfrac{m_2 l_2 l_1}{\text{term}_1} \cdot \dfrac{N K_1}{\sqrt{R^2 + (wL)^2}} & \dfrac{J_0}{\text{term}_1} \end{bmatrix}$$

Controller Design:

Before design control

1. check stability:
   a. After getting parameters from parameter estimation and SolidWorks CAD and get eigenvalues of A matrix:

$$\text{Eigenvalue } 1 = 0 \quad \text{Eigenvalue } 2 = 12.34$$

$$\text{Eigenvalue } 3 = -13.44 \quad \text{Eigenvalue } 4 = -5.21$$

As there is eigenvalue in positive side so system is unstable

2. Check controllability of the system and observability
   a. To check if the system is controllable or not, we must check whether the rank of the controllability matrix is full-rank or not.
      i. $Controlabilty_{Matrix} = [B \quad AB \quad A^2B \quad A^3B]$
      ii. Using A and B matrices we got the

$$Co\ Matrix = 1.0e6 \times \begin{bmatrix} 0 & 0 & 0 & -8 & -1 & 49 & 3 & 998 \\ 0 & 0 & 0 & 133 & 1 & -137 & 17 & -21416 \\ 0 & -8 & -1 & 49 & 3 & 998 & -6 & -7790 \\ 0 & 133 & 1 & -137 & 17 & -21416 & -133 & 42874 \end{bmatrix}$$

      iii. The Rank of Controllability Matrix is 4 Full Rank.
      iv. <u>The System is Controllable.</u>

   b. To check if the system is Observable or not, we must check the rank of observability matrix is full rank or not.
      i. $$Ob\ Matrix = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

ii. Using A and C matrices we got

$$Ob = 1.0e+04 \times \begin{bmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0010 & -0.0006 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0164 & 0.0006 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0010 & -0.0006 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0164 & 0.0006 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0061 & 0.0032 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0168 & -0.0039 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0061 & 0.0032 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0168 & -0.0039 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.1224 & -0.0123 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.6261 & -0.0784 \end{bmatrix}$$

iii. The Rank of Observability Matrix is 4 Full Rank.

iv. <u>The System is Observable.</u>

3. We use LQR to control the System:

LQR Technique:

Overview:

The Linear Quadratic Regulator (LQR) is a powerful and commonly used approach for designing optimal state feedback controllers. It is particularly well-suited for systems like the Furuta Pendulum, where the goal is to stabilize the pendulum in its upright position while balancing the trade-off between minimizing control effort and reducing state deviations.

Control Problem:

The LQR method seeks to minimize a quadratic cost function that captures both state deviations and control input efforts. The cost function, J, is expressed as:

$$J = \int_{0}^{\infty} (x^T Q x + u^T R u) \; dt$$

Where:

- Q: A positive semi-definite matrix that penalizes deviations in the system states from their desired values.
- R: A positive definite matrix that penalizes large control inputs to limit energy usage and prevent excessive actuation.

Optimal Control Law:

The objective is to determine the optimal state feedback gain matrix, KKK, that minimizes JJJ. The resulting control law is given by:

$$u=-Kx$$

Here:

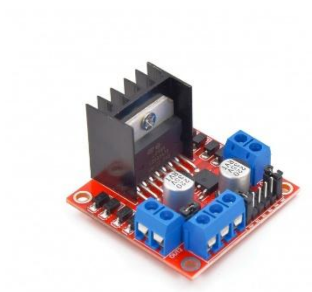
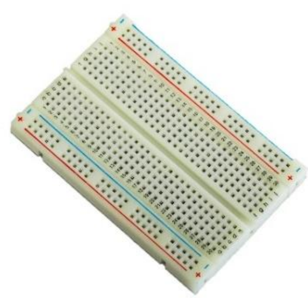
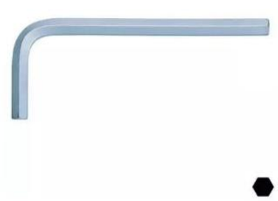
- u: The control input.
- K: The optimal gain matrix that maps the current state xxx to the required control input.

This feedback mechanism ensures the system responds optimally, stabilizing the pendulum in its upright position while efficiently balancing state deviations and control efforts.

In MATLAB, the lqr() function is used to compute the optimal state feedback gain matrix K for a given system. The function takes the system's state-space matrices and weighting matrices as inputs and returns the gain matrix K.

Hardware and Software

Hardware Components:

DC motor with encoder		**H-bridge**	
Breadboard		**Power supply**	
Esp32		**Encoder**	
Pendulum		**Coupler**	

Circuit Topology:

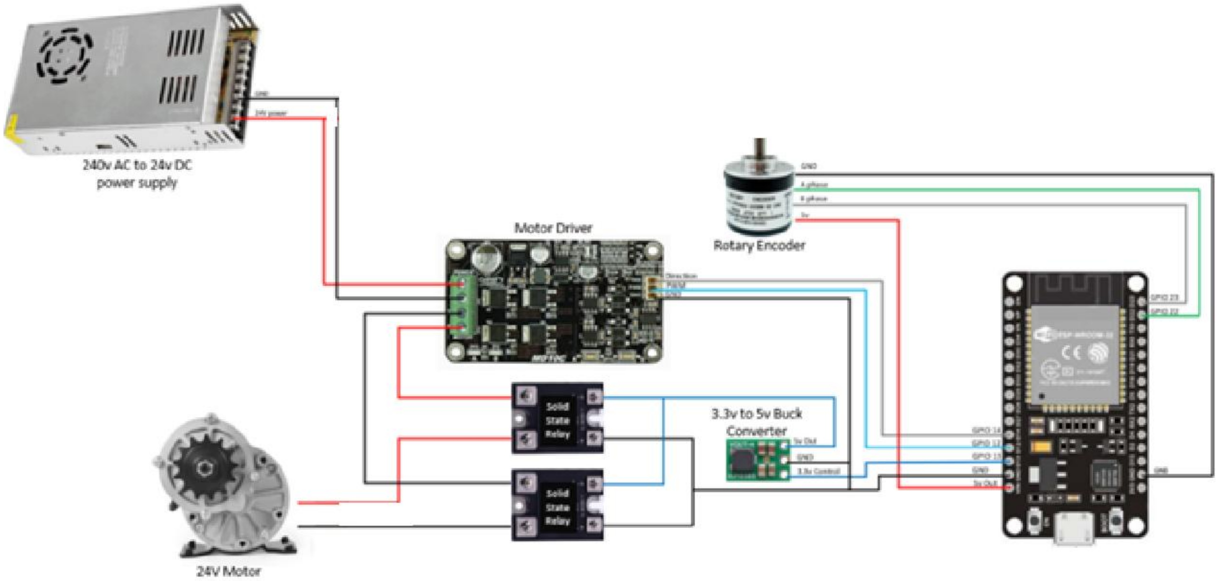


Figure 2:Circuit Topology

Mechanical Assembly:

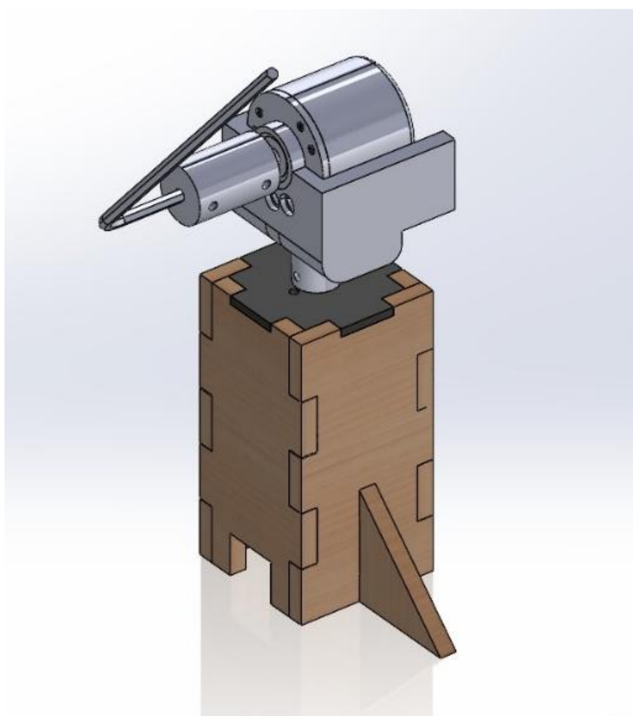
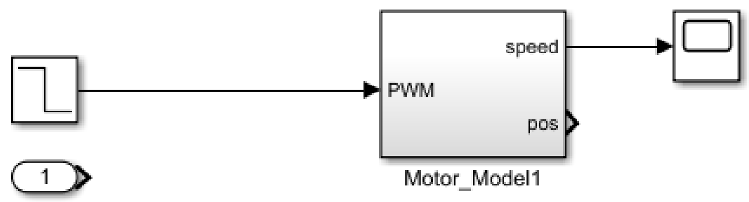


Figure 3: Mechanical Assembly

MATLAB/SIMULINK Implementation:

First, we start by developing a model for the DC motor:



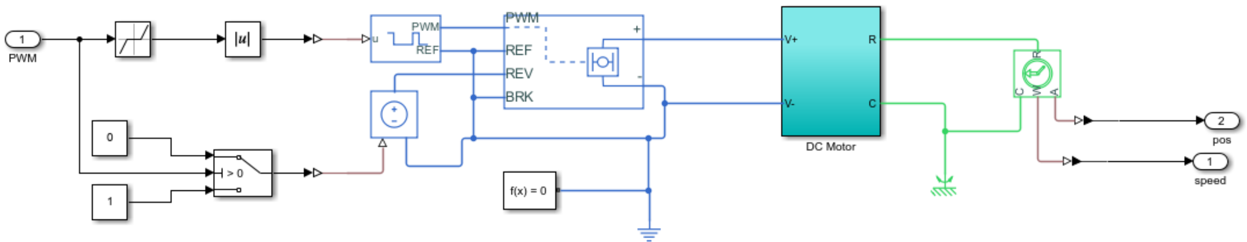
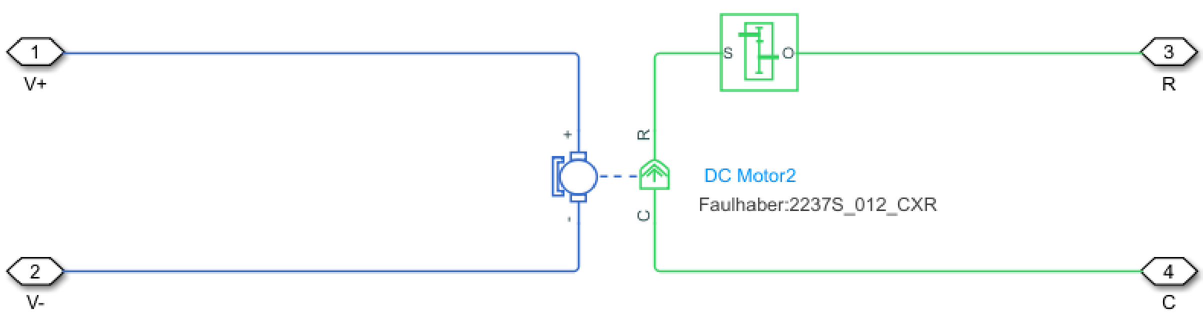


Figure 4:DC Motor Model



Next, we perform parameter estimation to determine the motor's parameters.

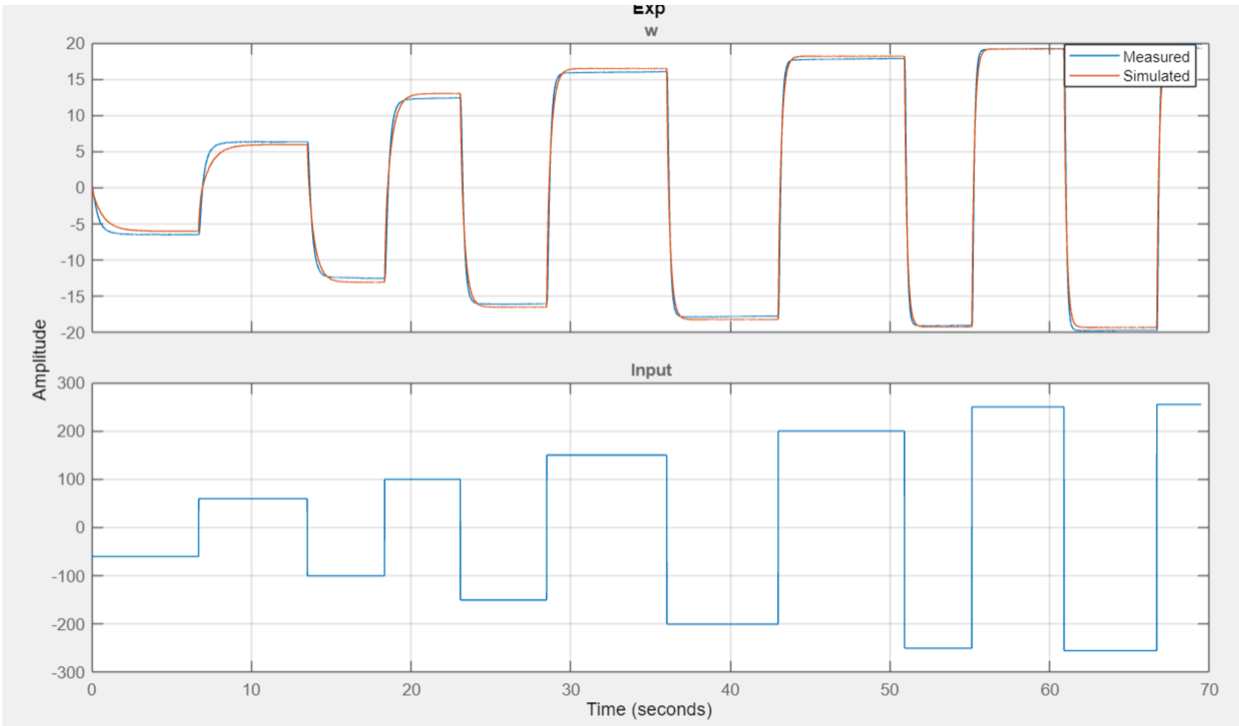
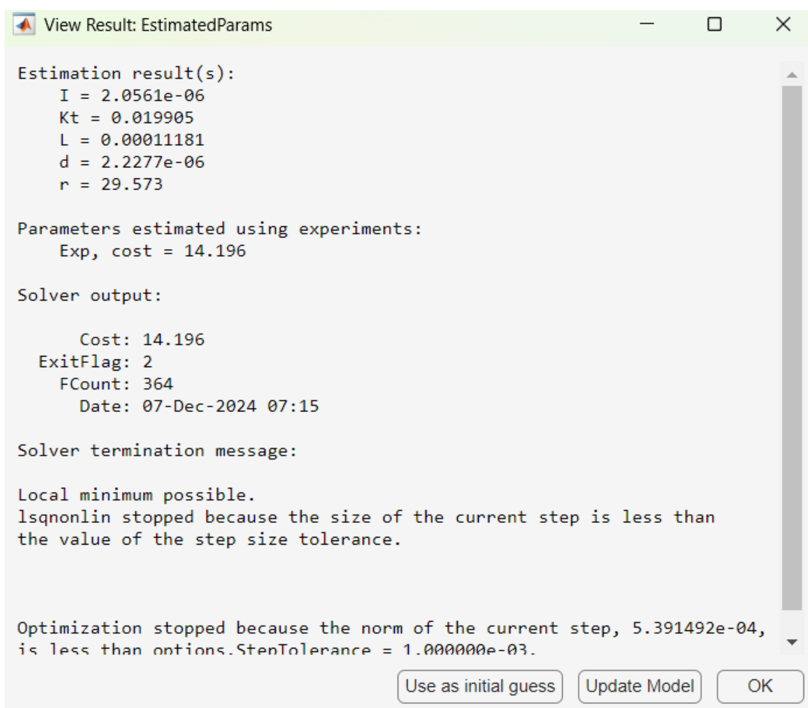


Figure 5: Parameter Estimation



View Result: EstimatedParams

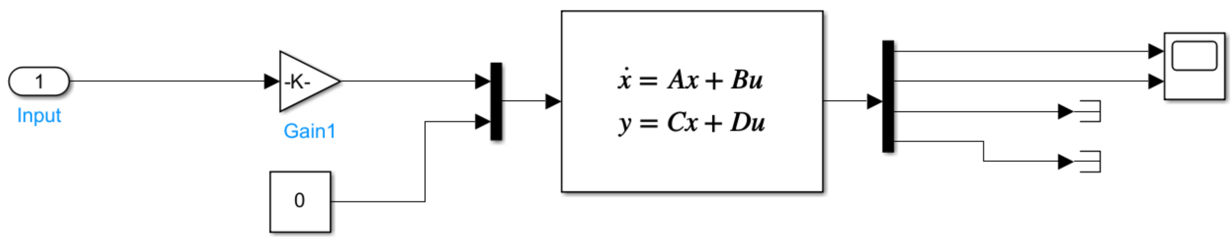
```
Estimation result(s):
    I = 2.0561e-06
    Kt = 0.019905
    L = 0.00011181
    d = 2.2277e-06
    r = 29.573

Parameters estimated using experiments:
    Exp, cost = 14.196

Solver output:

      Cost: 14.196
  ExitFlag: 2
    FCount: 364
      Date: 07-Dec-2024 07:15

Solver termination message:

Local minimum possible.
lsqnonlin stopped because the size of the current step is less than
the value of the step size tolerance.


Optimization stopped because the norm of the current step, 5.391492e-04,
is less than options.StepTolerance = 1.000000e-03.
```

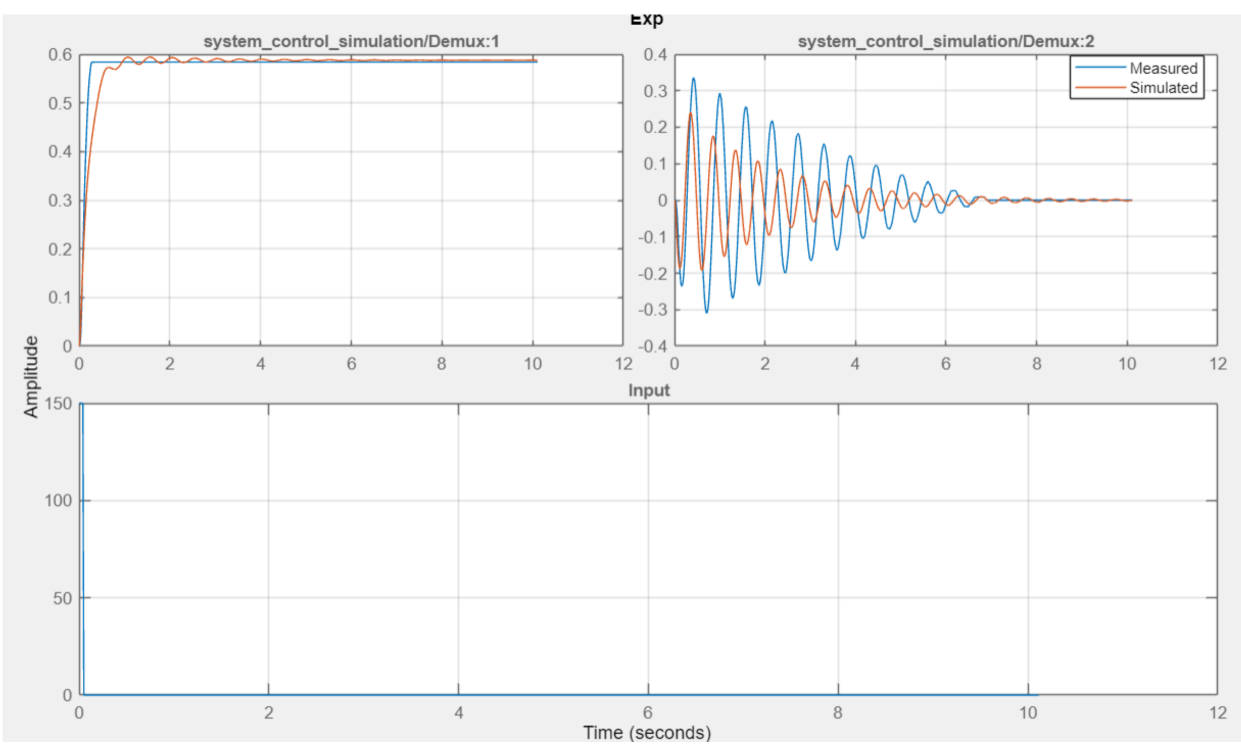
Use as initial guess | Update Model | OK

Figure 6:Estimation Results

We obtain parameters such as masses and lengths from the CAD model, verifying them through physical measurements as well.

Then, we build system models in Simulink and perform parameter estimation to determine values like damping and inertia for accuracy. (To ensure the system remains within the linearized region, we apply a small impulse (t = 0.05) to the motor while the pendulum is at rest in the downward position)





It's not entirely accurate, but it works—likely due to the effects of linearization.

Next, we implement the system model and parameters in MATLAB, initially assuming gain values to design an LQR controller and calculate the matrix K.

```
l1 = 0.07;                           % Length of Arm(1)
L2 = 0.125;                          % Length of Arm(2)
I=2.05613759e-06;                    % Moment of inertia of motor
J=6.1912e-10;                        % Moment of inertia of Arm(1) at its center
J2_p=2e-6;                           % Moment of inertia of Arm(2) at its center
b_theta1 = 0.0077;                   % Damping coefficient for Arm(1)
b_theta2 = 5e-5;                     % Damping coefficient for Arm(2)
m_1 = 0.133;                         % mass of Arm(1)
m_2 = 0.02;                          % mass of Arm(2)
r = 29.5729;                          % motor coil resistance
L = 1.1181*1e-4;                     % motor coil inductance
g = 9.81;                            % gravity
Kt = 0.02;                           % motor constant

% Substituting l1 = l1/2 and l2 = L2/2
l2 = L2 / 2;                        % center of mass of Arm (2)

J0 = (26^2*I+J) + m_2*l1^2;  % Moment of inertia of Arm 1 and motor
J2 = J2_p + m_2*l2^2;       % Moment of inertia of Arm 2

s = 1; % pendulum up (s=1) pendulum down (s=-1)
z=sqrt(r^2+(62832*L)^2);
d=2.2277e-06;
term_1 = J0 * J2 - m_2^2 * l1^2 * l2^2;

% Compute terms for B matrix
B31 = J2 / term_1;
B31 = B31 * 26*Kt/z;
B32 = s*m_2 * l2 * l1 / term_1;
B41 = s*m_2 * l2 * l1 / term_1;
B41 = B41 * 26*Kt/z;
B42 = J0 / term_1;

% Define the B matrix (inputs are volt and τ2)
B = [0    0;
     0    0;
     B31  B32;
     B41  B42];

% Compute terms for A matrix
A32 = g * m_2^2 * l2^2 * l1 / term_1;
A33 = -b_theta1 * J2 / term_1;
A33 = A33 - B31*(26^2*Kt^2/z);
A34 = s*-b_theta2 * m_2 * l2 * l1 / term_1;
A42 = s*g * m_2 * l2 * J0 / term_1;
A43 = s*-b_theta1 * m_2 * l2 * l1 / term_1;
A43 = A43 - B41*(26^2*Kt^2/z);
A44 = -b_theta2 * J0 / term_1;

% Define the state-space A matrix
A = [0,    0,    1,    0;
     0,    0,    0,    1;
     0,   A32,  A33,  A34;
     0,   A42,  A43,  A44];
```

```
% Output matrix C (identity matrix for state observation)
C = eye(4);  % Full-state observation

% Construct the D matrix (zero matrix)
D = zeros(4, 2);  % 4 outputs and 2 inputs

%LQR design
Q=diag([20 100 10 100]);% state matrix [theta alpha thetadot alphadot]
R=7;
K = lqr(A, B(:,1), Q, R);
```

SIMULATION AND TESTING

Control Performance in Simulink:

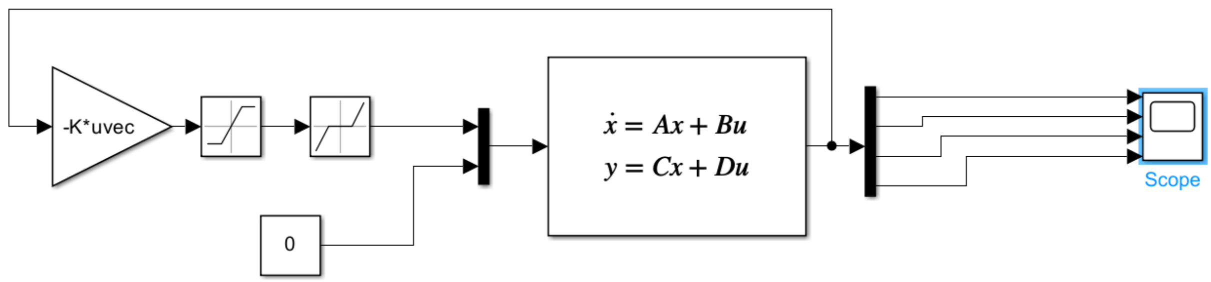
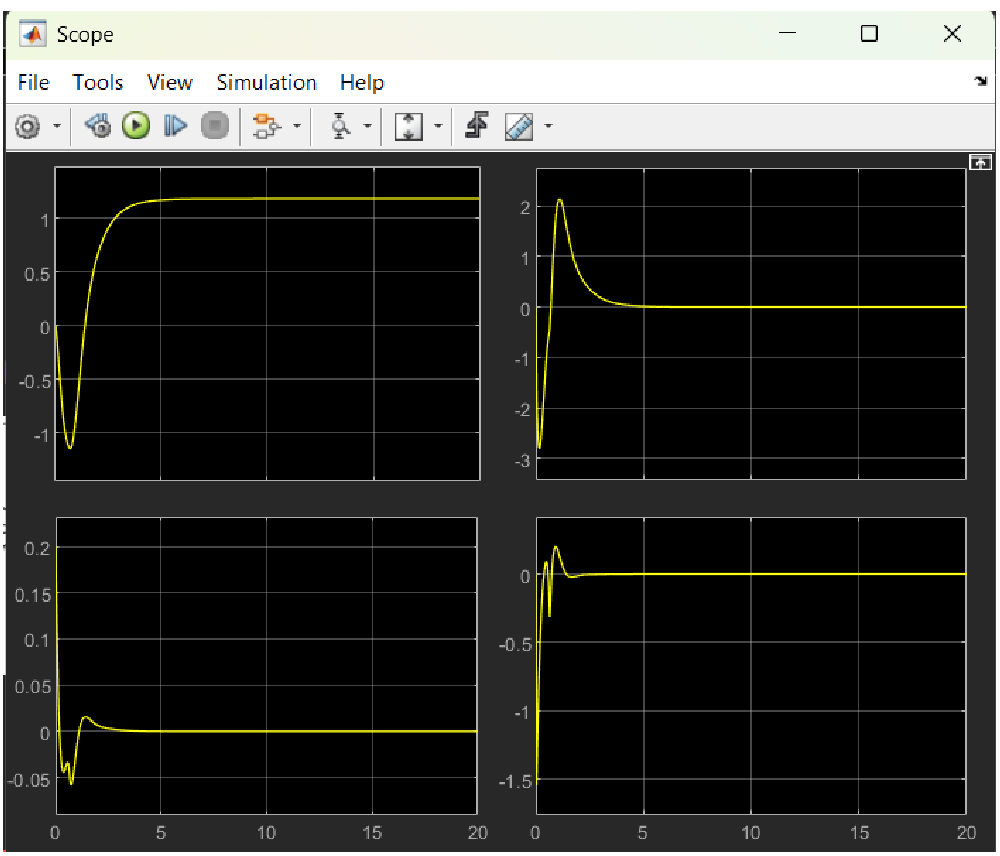
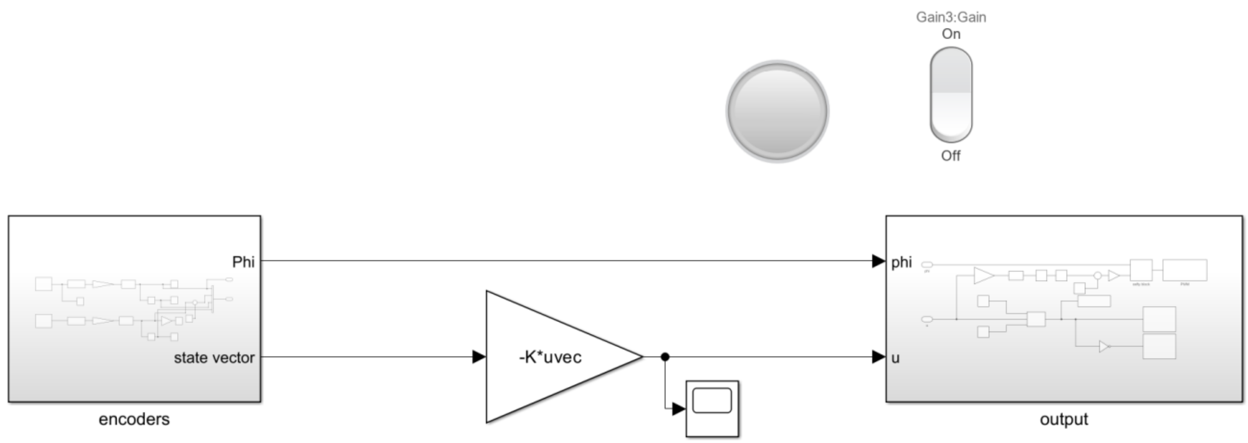
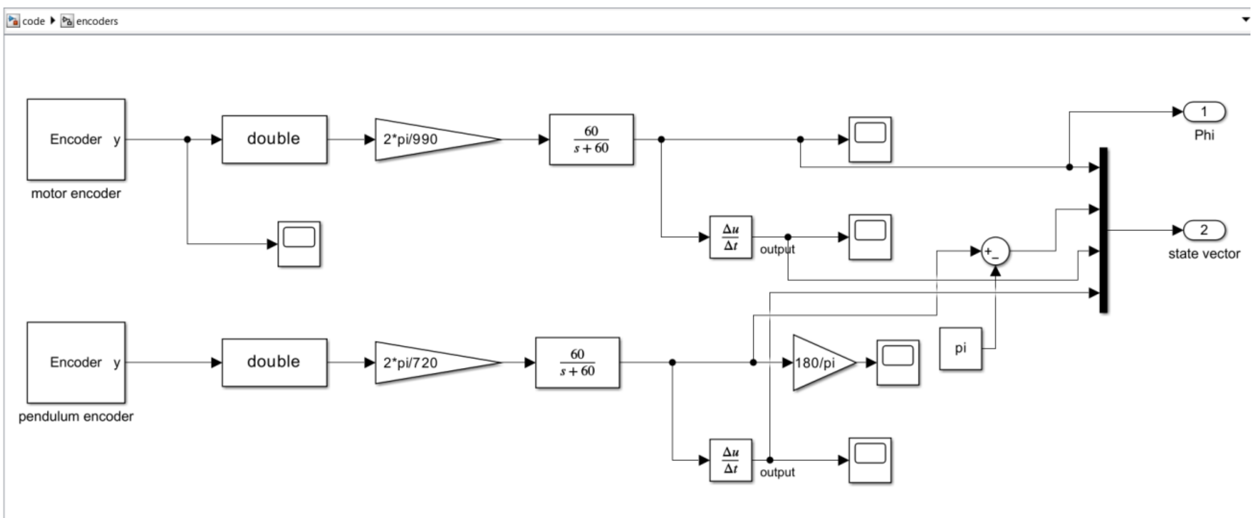


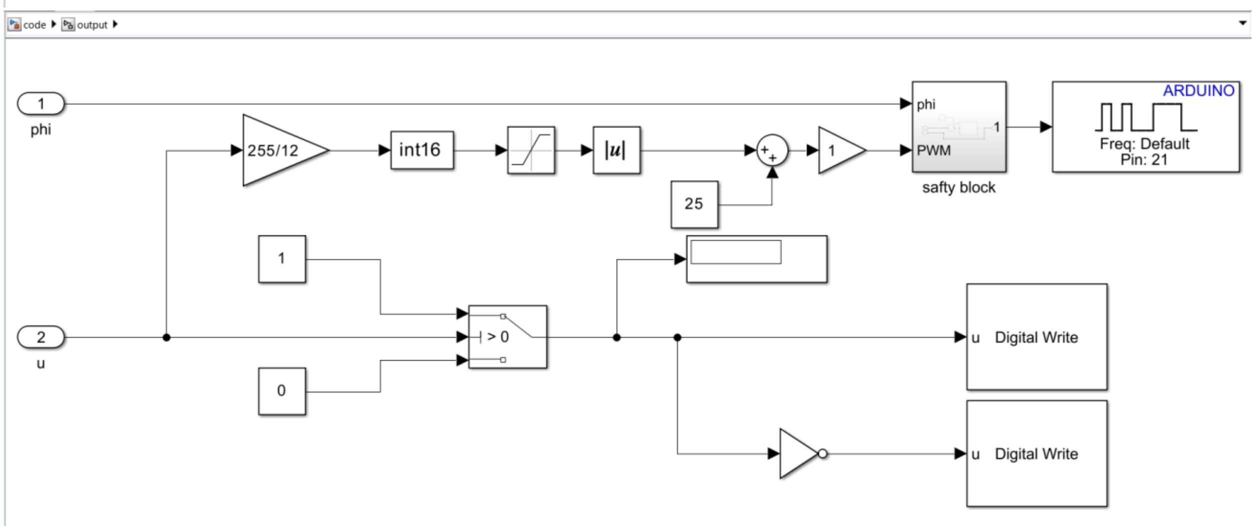
Figure 7:System Model



Hardware-in-the-Loop (HIL) code:

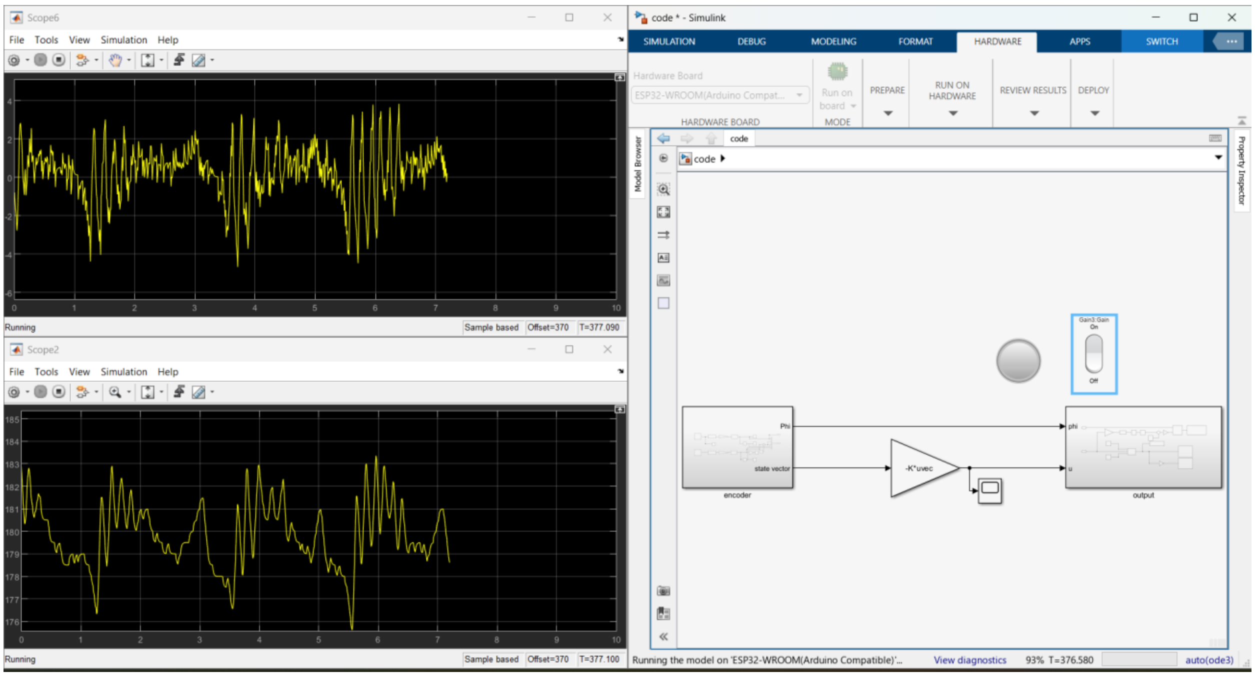


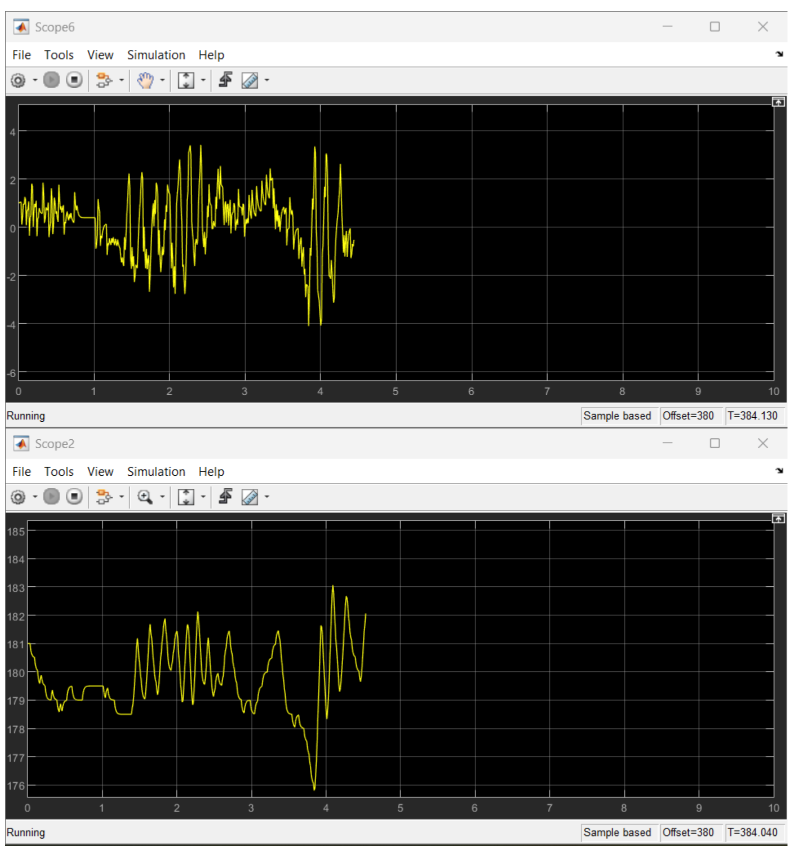




Results and Analysis

Stability Assessment





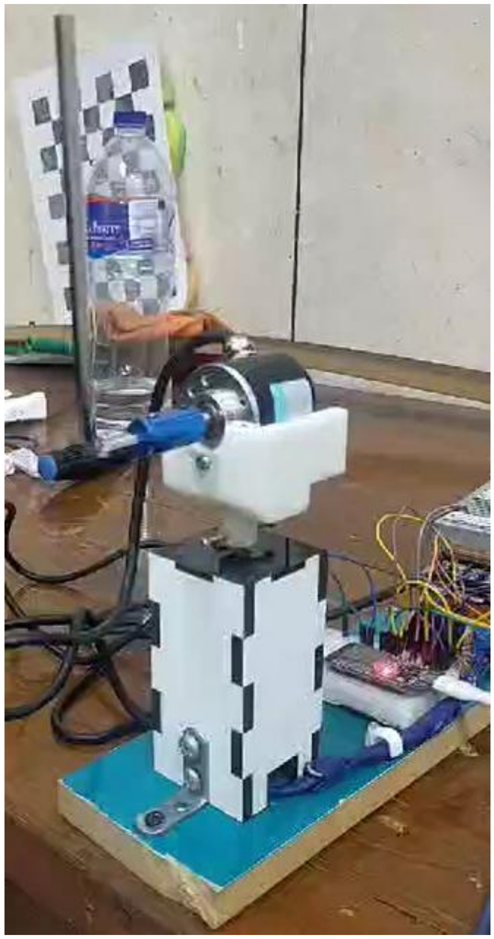


Figure 8:Real System

Response to Disturbances

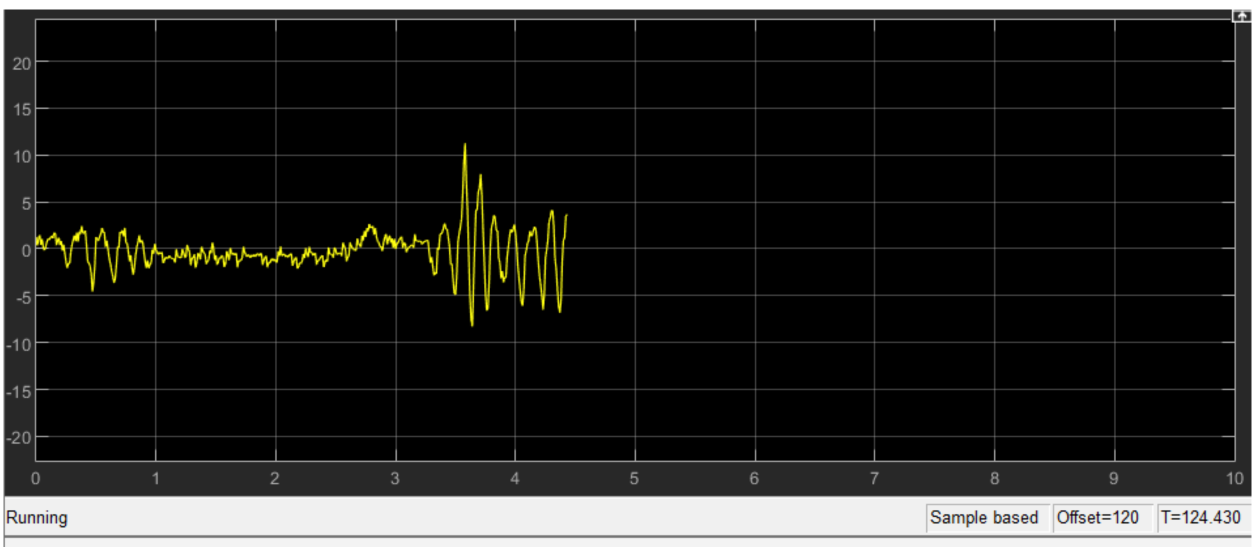


Figure 9:Disturbance Rejection

CHALLENGES

1. Dead zone of motor
2. Hysteresis in motor
3. Backlash in motor
4. Get accurate model to system
5. Tuning of controller

CONCLUSION

The Furuta Pendulum project serves as an illustrative example of the complexities inherent in controlling nonlinear, underactuated systems. Through a combination of modeling, simulation, and hardware implementation, we successfully demonstrated the stabilization of the pendulum in its upright position using advanced control strategies such as LQR.

While the system met its stability and disturbance rejection objectives, challenges such as motor dead zones, hysteresis, and backlash highlighted areas for further optimization. Additionally, refining the accuracy of the system model and enhancing the controller's tuning remain pivotal for improving performance.

PROJECT LINK:

https://drive.google.com/file/d/1tvSFMBx72GJj9IXAS10luPX_5w33Nhrb/view?usp=drive_link