Ain Shams University
Faculty of Engineering

MCT411s
Fall 25

HYBRID CONTROL

Project Report

Name	ID	Contribution
Amr Khaled Abd Elsadik	2100524	20%
Hala Ehab Fawzy	2100708	20%
Youssef Ahmed Abd-El-Fattah	2101059	20%
Arwa Ramadan Abdelfattah	2100647	20%
Mario Sameh Samir	2100364	20%

Submitted to:

Eng.: Hesham Salah

Eng.: Mohab Ahmed

Table of Contents

Introduction 1
Objectives 1
Mechanical Design 2
Modelling of the pendulum system 2
Assumptions 2
System Equations 3
Controller design and Requirements 5
Methodology 5
Controller Requirements 5
LQR Design 5
Parameter Estimation 6
Simulation results 6
Estimation Iterations 6
Validation Results 7
Estimated Parameters 8
Simulink HIL 8
Simulation 13
Simulation Results 13
Simulation vs implementation 14
GUI 14
Conclusion 14
References 15

Table of Figures

Figure 1 Furuta Pendulum Diagram 1
Figure 2 Mechanical Design 2
Figure 3 Step =50........ 6
Figure 4 Step =150........ 7
Figure 5 Ramp =50 7
Figure 6 Estimated Parameters 8
Figure 7 Whole System........ 8
Figure 8 Encoder Subsystem 9
Figure 9 Motor Subsystem........ 9
Figure 10 MotorEncoder........ 10
Figure 11 Emergency Stop........ 10
Figure 12 Wrapper 11
Figure 13 Swing up........11
Figure 14 swing up function 12
Figure 15 Simulation model........ 13
Figure 16 Simscape Multibody........ 13
Figure 17 APP GUI 14

Introduction

In this project, a Furuta pendulum system is modeled and controlled using modern control techniques, specifically pole placement and/or Linear Quadratic Regulator (LQR), as required. The system is developed and tested in MATLAB/Simulink, including Hardware-in-the-Loop (HIL) implementation. A physical prototype is constructed and integrated with sensors and actuators to validate the controller's performance against real-world uncertainties.

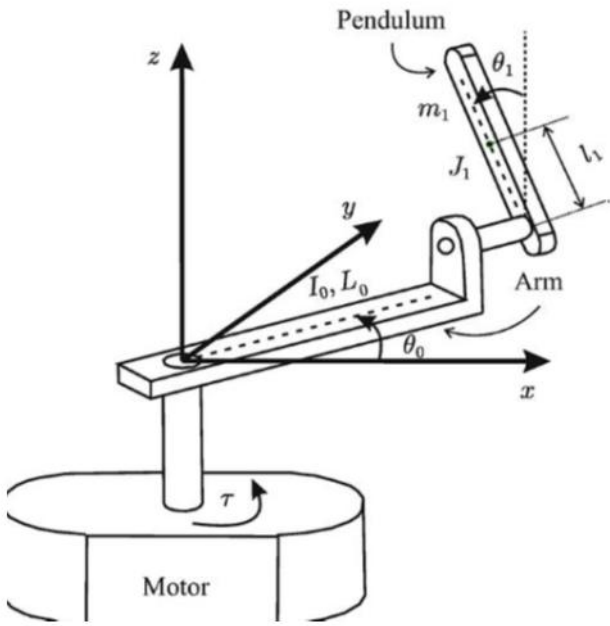

Figure 1 Furuta Pendulum Diagram

The mechanical structure of the Furuta pendulum is composed of two arms. The first arm, actuated by an electric motor, rotates parallel to the horizontal plane. The second arm, the pendulum, hangs at the tip of the first arm and rotates without actuation around the axis of the first arm.

Objectives

- Design a physical model for Furuta pendulum
- Design a controller using LQR technique to control the pendulum
- Implement the system in Simulink with HIL.
- Design a Gui Displaying System angles and actuator Signal

Mechanical Design

The Physical platform of the Furuta pendulum is mainly composed of main microcontroller (Arduino Uno), base, geared Dc motor with encoder, Dc motor driver, power supply, 600 ppr encoder, encoder bracket, pendulum rod, as shown in Figure.3. The rotating arm is connected to the motor and moves in a circle in the horizontal plane, driving the pendulum rod through coupling. The encoder can measure the angle of the pendulum rod in real time, and an angle signal transmits data in real time

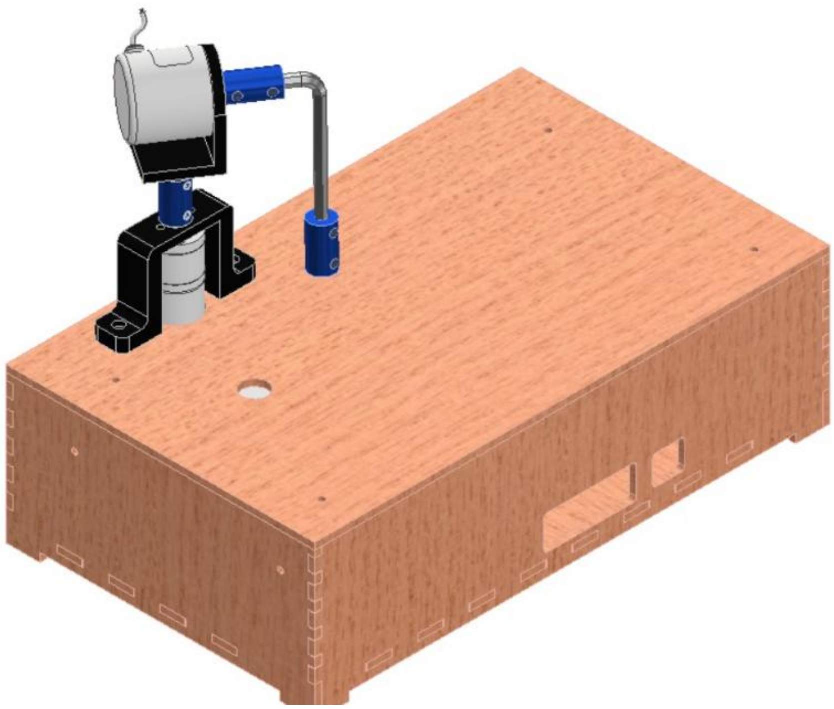

Figure 2 Mechanical Design

Modelling of the pendulum system

[1]The pendulum is displaced with an angle while the direction of is in the x-direction of this illustration. So, mathematical model can be derived by examining the velocity of the pendulum center of mass.

Assumptions

1- The system starts in a state of equilibrium
2- The pendulum does not move more than a few degrees away from the vertical to satisfy a linear model
3- System should withstand small external disturbances

L	Length to Pendulum's Center of Mass
m	Mass of Pendulum Arm
r	Rotating Arm Length Vx Velocity
Θ	DC motor load gear angle (radians)
α	Pendulum Arm Deflection (radians)
h	Distance of Pendulum Center of mass from ground
Jcm	Pendulum Inertia about its center of mass
Vx	Velocity of Pendulum Center of mass in the x-direction
Vy	Velocity of Pendulum Center of mass in the y-direction

System Equations

There are two components for the velocity of the Pendulum lumped mass. So,

$$V_{Pen.\ center\ of\ mass} = -L\cos\alpha(\dot{\alpha})\hat{x} - L\sin\alpha(\dot{\alpha})\hat{y} \qquad (1)$$

$$V_{arm} = r\dot{\theta} \qquad (2)$$

The equations (1) and (2) can solve the x and y velocity components as,

$$V\,x = r\dot{\theta} - L\cos\alpha(\dot{\alpha}) \qquad (3)$$

$$V\,y = -L\sin\alpha(\dot{\alpha}) \qquad (4)$$

Potential Energy: The only potential energy in the system is gravity. So

$$V = P.E._{Pendulum} = mg\,h = mg\,L\cos\alpha \qquad (5)$$

Kinetic Energy: The Kinetic Energies in the system arise from the moving hub, the velocity of the point mass in the x-direction, the velocity of the point mass in the y-direction and the rotating pendulum about its center of mass.

$$T = K.E._{Hub} + K.E._{Vx} + K.E._{Vy} + K.E._{Pendulum} \qquad (6)$$

The moment of inertia of a rod about its center of mass is,

$$J_{cm} = (1/12)MR^2 = (1/12)M(2L)^2 = (1/3)ML^2 \qquad (8)$$

So, complete kinetic energy

$$T = (1/2)Jeq\,\dot{\theta}^2 + (1/2)m\left(r\dot{\theta} - L\cos\alpha(\dot{\alpha})\right)^2 + (1/2)m(-L\sin\alpha(\dot{\alpha}))^2 + (1/2)\,J\,cm\,\dot{\alpha}^2 \qquad (9)$$

The Lagrangian can be formulated as

$$L = T - V = (1/2)Jeq\,\dot{\theta}^2 + (2/3)mL2\dot{\alpha}^2 -$$
$$mLr\cos\alpha(\dot{\alpha})(\dot{\theta}) + (1/2)\,mr2\dot{\theta}^2 - mgL\cos\alpha \quad (10)$$

The two generalized co-ordinates are and. So, another two equations are,

$$\frac{\delta}{\delta t}\left(\frac{\delta L}{\delta \dot{\theta}}\right) - \frac{\delta L}{\delta \theta} = T_{output} - B_{eq}\dot{\theta} \qquad (11)$$

$$\frac{\delta}{\delta t}\left(\frac{\delta L}{\delta \dot{\alpha}}\right) - \frac{\delta L}{\delta \alpha} = 0 \qquad (12)$$

Solving the equations and linearizing about = 0, equations become,

$$\left(J_{eq} + mr^2\right)\ddot{\theta} - mLr\ddot{\alpha} = T_{output} - B_{eq}\dot{\theta} \qquad (13)$$

$$\frac{4}{3}mL^2\ddot{\alpha} - mLr\ddot{\theta} - mgL\alpha = 0 \qquad (14)$$

The output Torque of the motor which act on the load is defined as,

$$T_{output} = \frac{\eta_m \eta_g K_t K_g (V_m - K_G K_m \theta)}{R_m} \qquad (15)$$

Finally, by combining the above equations, the following state-space representation of the complete system is obtained.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{bd}{E} & \frac{-cG}{E} & 0 \\ 0 & \frac{qd}{E} & \frac{-bG}{E} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ c\frac{\eta_m \eta_g K_t K_g}{R_m E} \\ b\frac{\eta_m \eta_g K_t K_g}{R_m E} \end{bmatrix} V_m \qquad (16)$$

Here, $a = J_{eq} + mr^2$, $b = mLr$, $c = 4/3mL^2$, $d = mgL$

$$E = ac - b^2, \quad G = \frac{\eta_m \eta_g K_t K_m K_g^2 - B_{eq} R_m)}{R_m}$$

Controller design and Requirements

Methodology

The Furuta Pendulum is a non-linear and inherently unstable system, making it an interesting control problem. The linearization of the Furuta Pendulum system is typically done around an equilibrium point to apply linear control techniques like LQR. LQR is an optimal control method for linear systems, finding the best state-feedback gain (K) by minimizing a quadratic cost function that balances state errors (Q matrix) and control effort (R matrix). The methodology involves defining system dynamics (A, B matrices)

Controller Requirements

- Control the angle θ_1 to be zero. So, the pendulum stands in the upright position.
- Overcome any disturbances that may affect the pendulum. So, the pendulum should maintain its upright position and don't fall.

LQR Design

1- System Modelling

Begin by modeling the system dynamics in state-space form: x represents the state vector, u is the control input, A is the state matrix, and B is the input matrix

$$\dot{x} = Ax + Bu$$

2- Cost Function

Define a quadratic cost function that reflects the system's performance. The typical cost function is given by: $\int_0^\infty (X^T * Qx + U^T * Ru)dt$

Q and R are positive semi-definite weighting matrices that influence the importance of the state and control efforts in the performance

3- Calculate the Gain matrix (K)

4- Implement the controller

$$u = -K * X$$

5- Testing and Tuning

Simulate the controlled Furuta Pendulum system to evaluate its performance under different initial conditions and disturbances. Tune the weighting matrices Q and R if necessary to achieve the desired output

Parameter Estimation

The identification of system parameters plays an essential role in system modeling and control. The parameter estimation may be represented as an optimization problem. Firstly, the initial values of the DC motor parameters are extracted using the dynamic model through measuring the values of voltage, current, and speed of the motor. Then, these values are used as an initial value for Simulink design optimization. The experimental input/output data can be collected using a suggested microcontroller-based circuit that will be used later for estimating the DC motor parameters by building a Simulink model. Two optimization algorithms are used, the pattern search and the nonlinear least square. The results show that the nonlinear least square algorithm gives a more accurate result that almost approaches the actual measured speed response of the motor.

Simulation results

Estimation Iterations

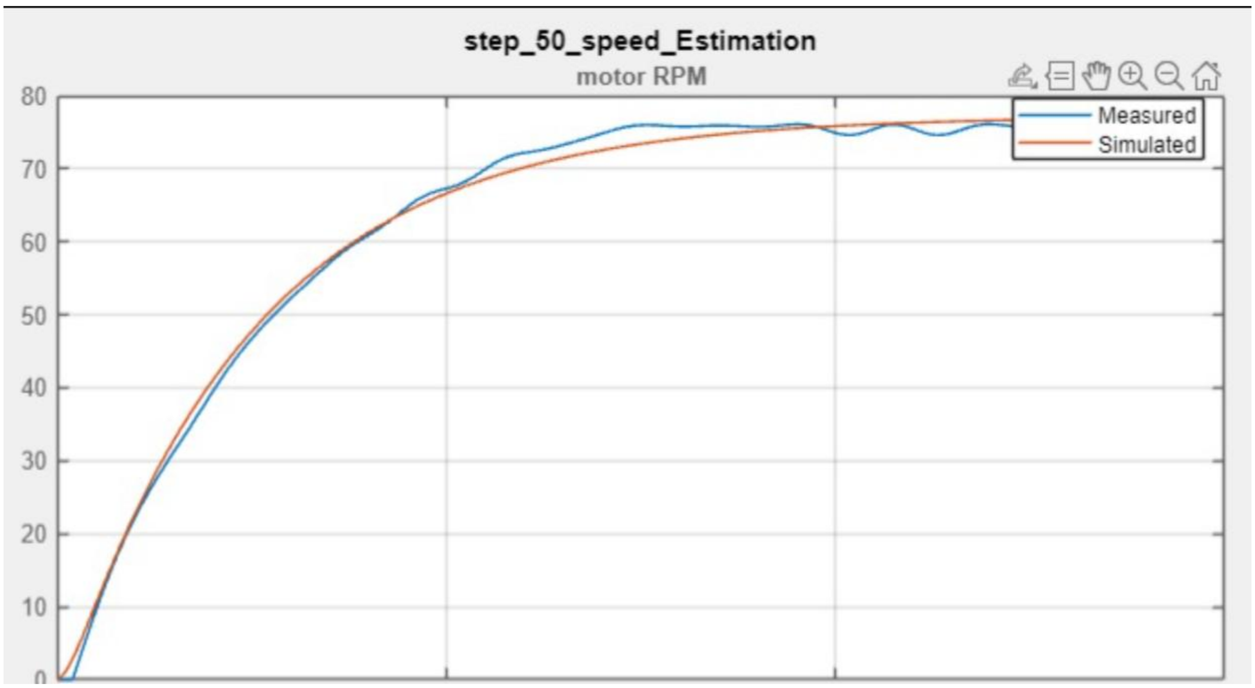

Figure 3 Step =50

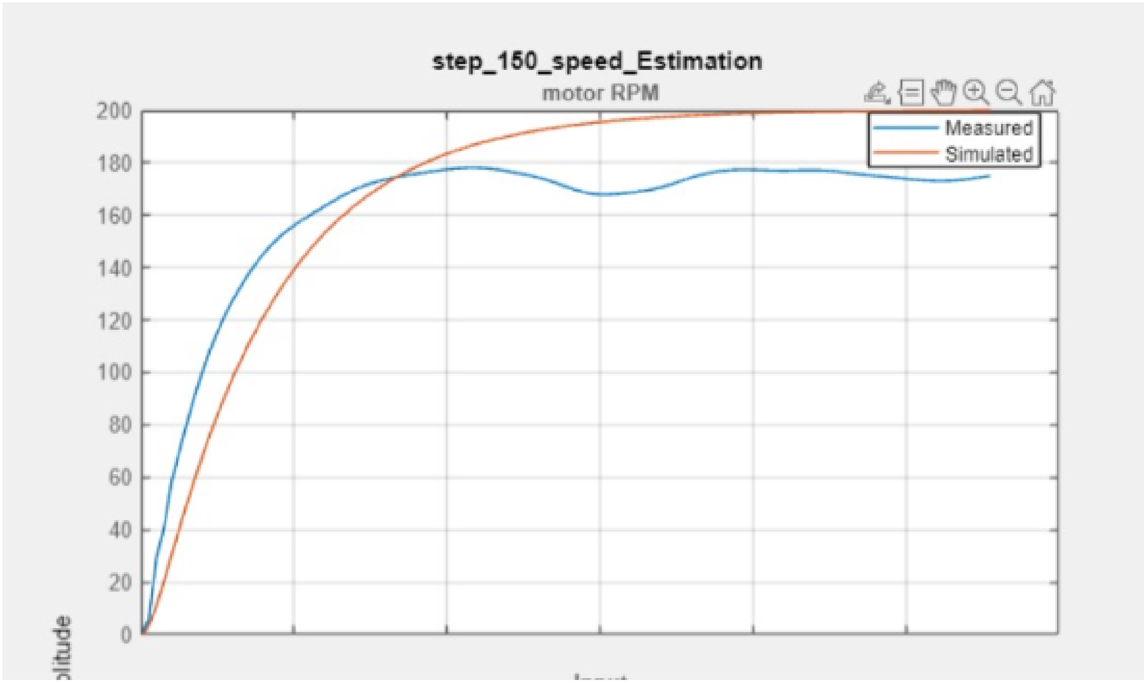

Figure 4 Step =150

Validation Results

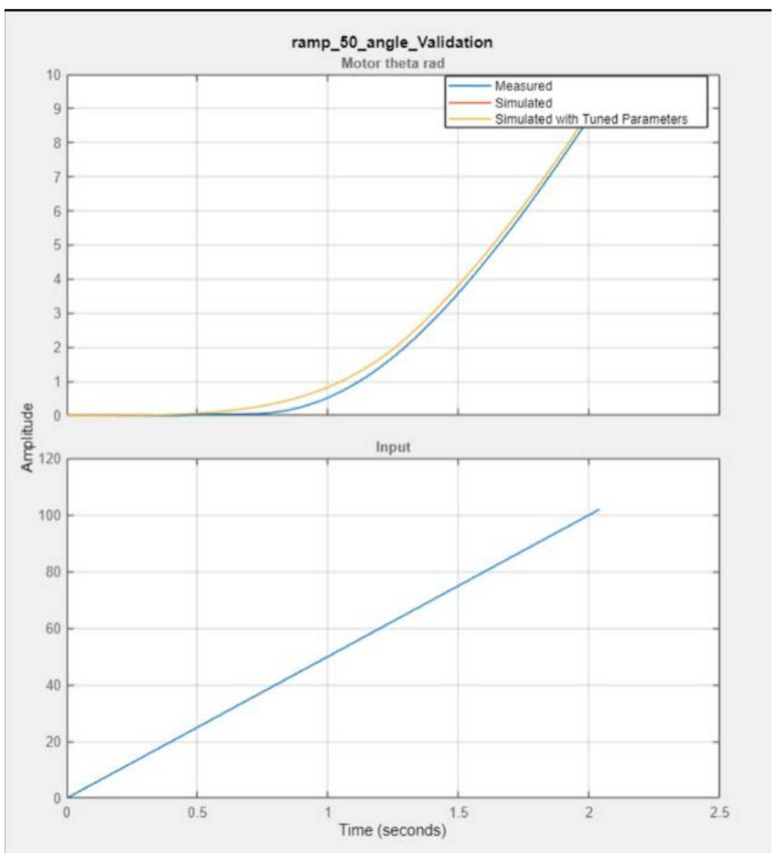

Figure 5 Ramp =50

Estimated Parameters

```
Estimation result(s):
    B1 = 0.0039055
    I = 0.00035491
    Kv = 0.078249
    H = 0.0054872
```

Figure 6 Estimated Parameters

Simulink HIL

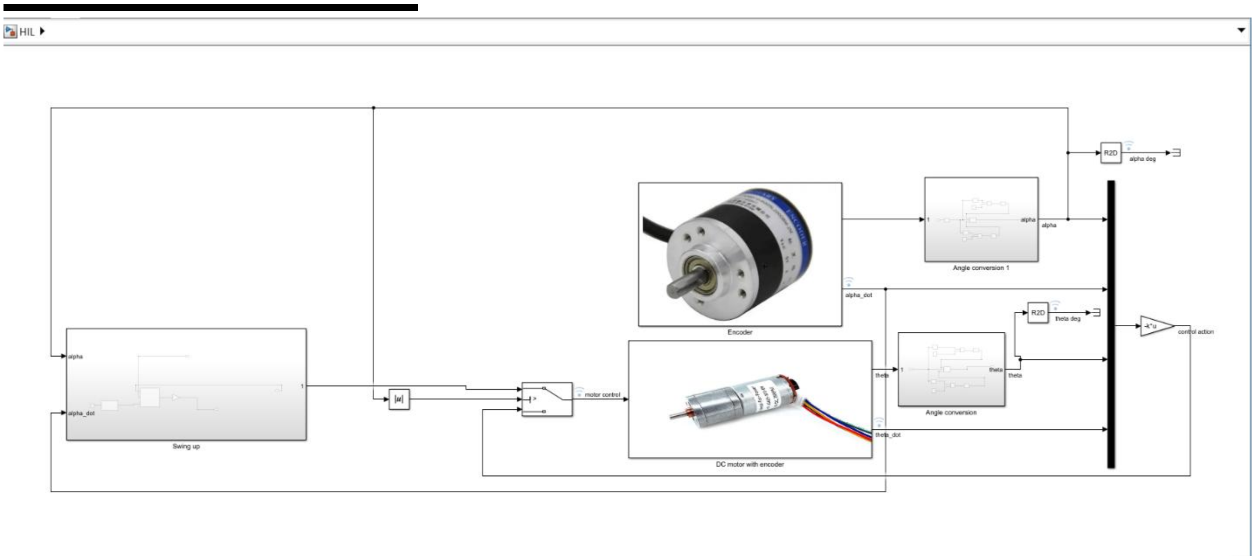

Figure 7 Whole System

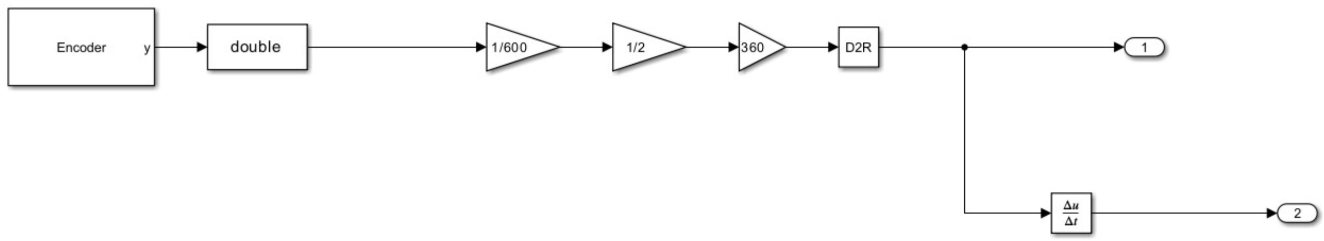

Figure 8 Encoder Subsystem

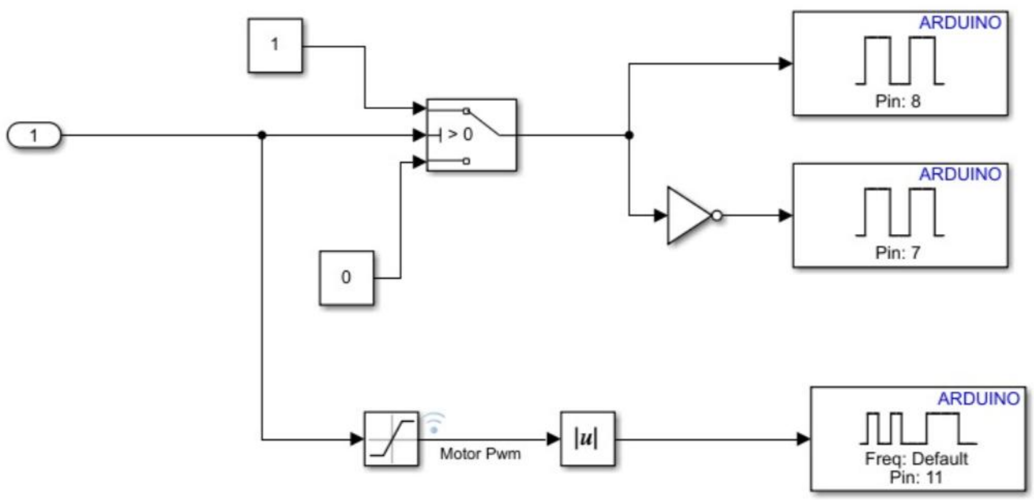

Figure 9 Motor Subsystem

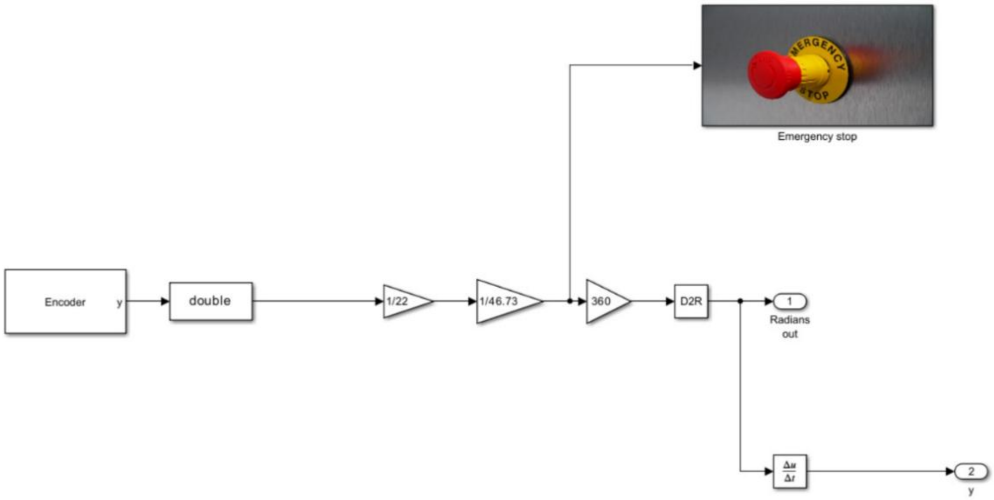

Figure 10 MotorEncoder

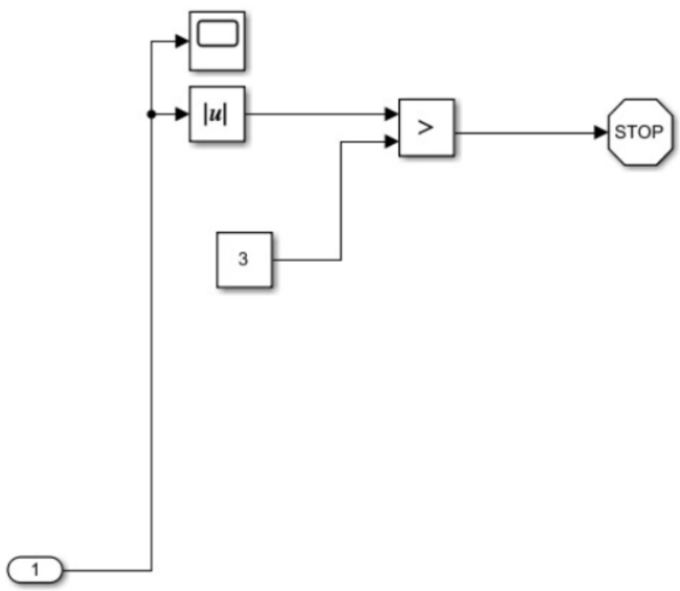

Figure 11 Emergency Stop

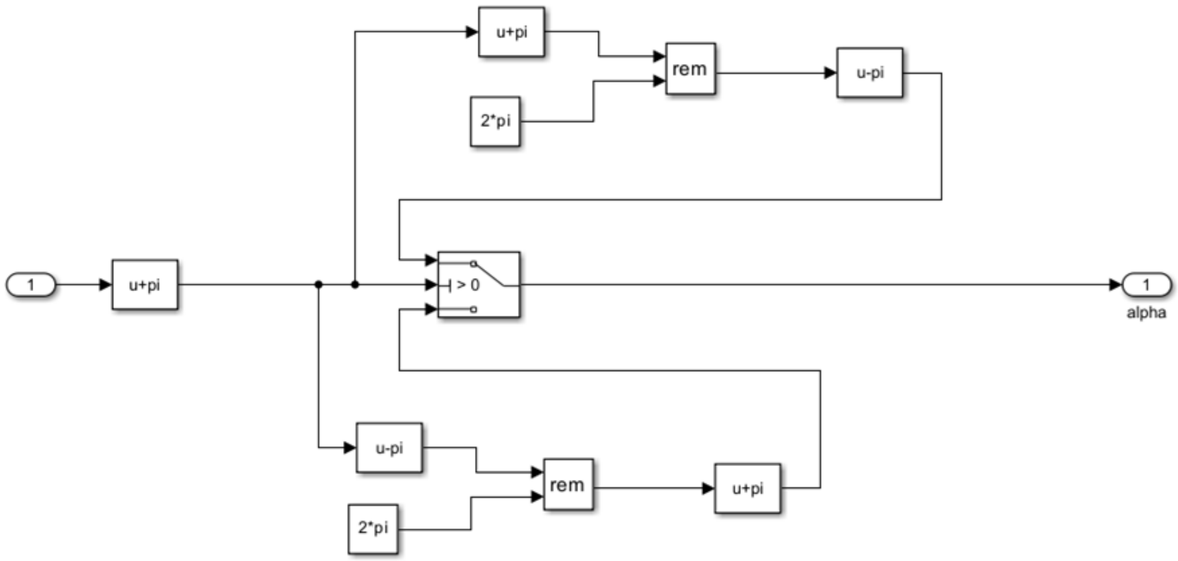

Figure 12 Wrapper

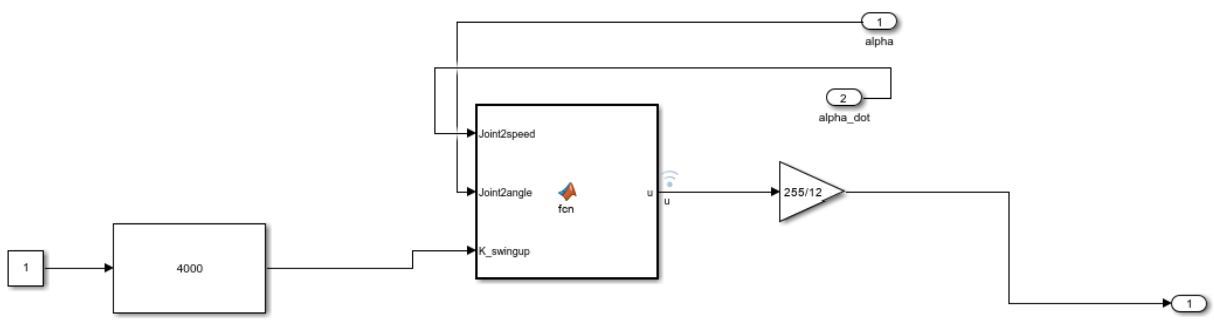

Figure 13 Swing up

For swingup we used the Total energy control system approach

```
function u = fcn(Joint2speed,Joint2angle,K_swingup)

i=1;
g = 9.81;
m=0.010143116119298235; %Kg
I=5.737507722492961 4*1e-6; %Kg.m^2
l=27.156641173388582*1e-3; %[m]

if Joint2angle>2*pi
    Joint2angle = Joint2angle-(2*pi);
elseif Joint2angle < 0
    Joint2angle = Joint2angle +2*pi;
end

S=sign(Joint2speed*cos(Joint2angle));

if Joint2speed * cos(Joint2angle) >=0
    delta=1;
else
    delta=-1;
end


E = 0.5*(I+m*l^2)*Joint2speed^2 + m*g*l*(cos(Joint2angle)-1);
E_norm =E/(m*g*l);
E_ref = 0;
E_err = E_ref-E;

u = K_swingup*(E_err)*delta;

% if Joint2speed ==0
%     i=i+1;
% end
% if mod(i,2)
%     u = K_swingup*E_err*delta;
% else
%     u = -K_swingup*E_err*delta;
% end
if Joint2angle <=(pi/11) && Joint2angle >=-(pi/11)
    u =0;
end
end
```

Figure 14 swing up function

Simulation

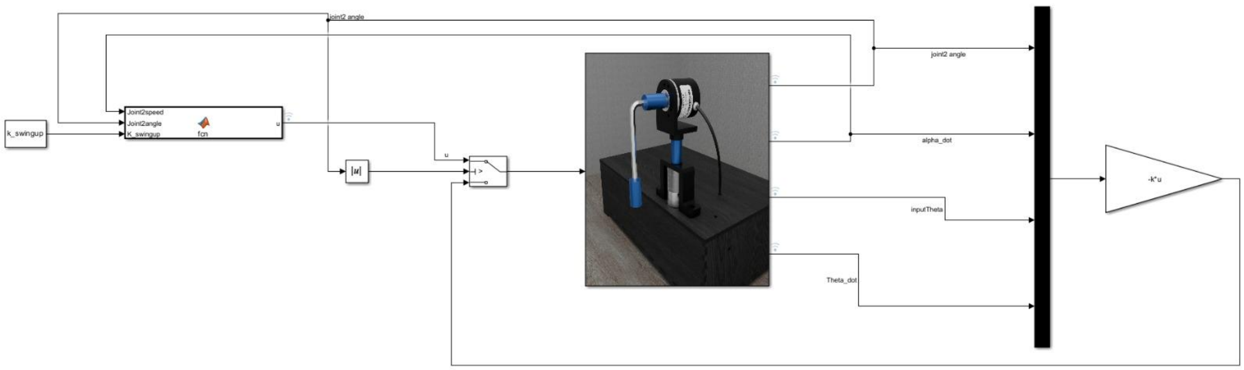

Figure 15 Simulation model

Simulation Results

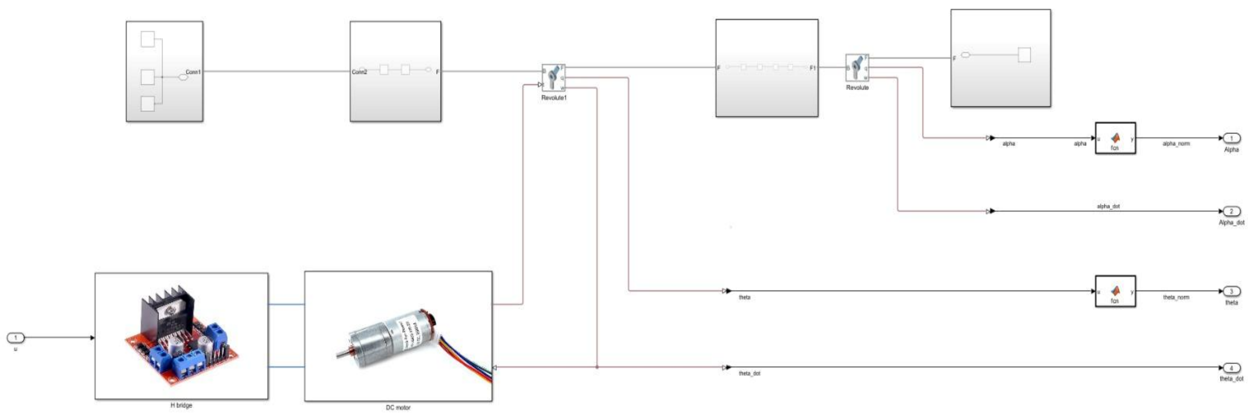

Figure 16 Simscape Multibody

Simulation vs implementation

GUI

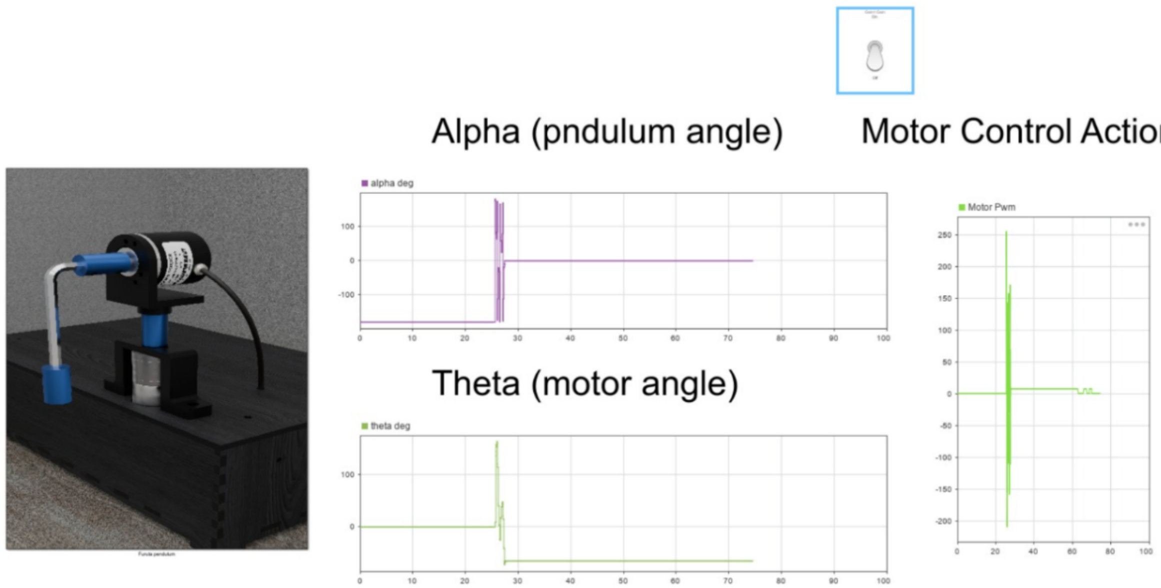

Figure 17 APP GUI

Conclusion

-The Furuta pendulum project successfully demonstrated the application of modern control techniques to a highly nonlinear and unstable system. A mathematical model of the system was developed and linearized around the upright equilibrium point, enabling the design of a stabilizing controller using LQR and/or pole placement methods.

-Tunning of parameters was one of the major problems that has been solved by trial and error for different parameters until applying the suitable performance of the pendulum movement.

References

[1] "Modeling and Control of a Rotary Inverted Pendulum Using Various Methods, Comparative Assessment and Result Analysis," [Online]. Available: https://www.researchgate.net/publication/224178913_Modeling_and_control_of_a_rotary_inverted_pendulum_using_various_methods_comparative_assessment_and_result_analysis.