

MCT411: Hybrid Control

Fall 2025

Furuta Pendulum

Final Report - Team 14



Name	ID
Abdelghaffar Essam Abdelghaffar	2100428
Ingy Ahmed Sherif	2100648
Kareem Saleh Hassan	2100909
Mariam Mohamed El Sebaey	2100260

Submitted to: Dr. Mohamed Ibrahim
Dr. Mohamed Hamdy.
Eng. Mohab Ahmed.
Eng. Hesham Salah.

Contribution

Name	Contribution	
Abdelghaffar Essam	Mechanical Design, LQR Implementation	25%
Ingy Ahmed	LQR Implementation, Simulation Setup	25%
Kareem Saleh	Mechanical Design, Swing-up Implementation	25%
Mariam El Sebaey	Swing-up Implementation, Simulation Setup	25%

Table of Contents

1. Introduction	3
2. Physical System and Mathematical Modeling	4
2.1 Physical System Overview	4
2.2 System Parameters.....	5
2.3. System Model.....	6
2.4. Controller Requirements.....	8
2.5. Swing-Up Controller (Energy Control).....	8
2.6. Balancing Controller (LQR)	9
2.7. Implementation Details.....	13
3. Simulation Results.....	16
4. Real-Life Results	17
5. Conclusion.....	19
6. References.....	20

Table of Figures

Figure 1. Furuta Pendulum.....	3
Figure 2. Full CAD Model.....	5
Figure 3. System Variables.	6
Figure 4. Swing-up Model in Simulink.....	9
Figure 5. Exported Model.	11
Figure 6. Linearization I/O.....	11
Figure 7. Operating Point Setting.	12
Figure 8. Full Simulink Model.....	13
Figure 9. State Diagram.....	14
Figure 10. Futura Pendulum Model with Motor.	15
Figure 11. Simulation Results.....	16
Figure 12. Real-Life Results.	18

1. Introduction

The Furuta Pendulum is an example of a nonlinear, underactuated mechatronic system with fast dynamics. The movement of this device consists of two arms: a horizontally mounted arm driven by a motor and a vertical pendulum arm attached to its end, where the rotary axes of both arms are perpendicular to each other, as shown in Figure 1.

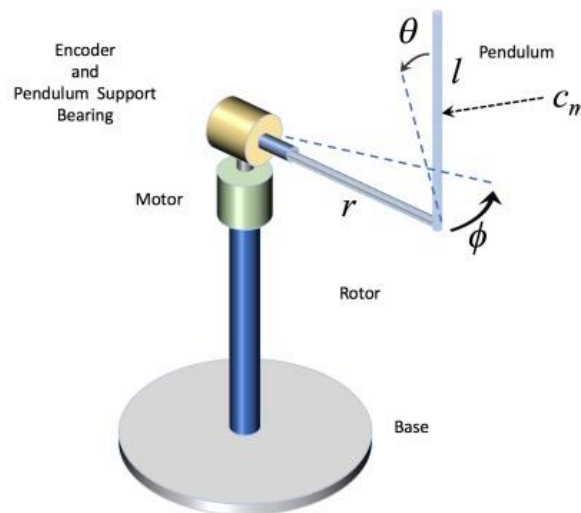


Figure 1. Furuta Pendulum.

The primary challenge of this project is to control a system that has only one exciting torque applied to the first degree of freedom while managing the other degrees of freedom. The control objective is divided into two distinct stages:

1. **Swing-Up Phase:** The goal is to move the pendulum from its naturally stable downward position to the upright equilibrium position.
2. **Balancing Phase:** Once the pendulum reaches the upright position, the controller must regulate the angle θ_1 to be zero and maintain this position despite external disturbances.

For this project, we have implemented an **Energy-Based Swing-Up** controller to sequentially add energy to the system by considering the energy of the pendulum in the upright equilibrium position vs. the actual current energy. Once the pendulum is near the upright position, the control is handed over to a **Linear Quadratic Regulator (LQR)** for stabilization, as required by the project constraints. The entire system is modeled and tested using **MATLAB/Simulink** with **Hardware-in-the-Loop (HIL)** implementation to verify the robustness of the physical prototype.

2. Physical System and Mathematical Modeling

This section details the hardware requirements and the analytical approach used to describe the Furuta Pendulum's dynamics.

2.1 Physical System Overview

The Furuta Pendulum setup used in this project consists of a compact, high-fidelity mechatronic system designed for both simulation and Hardware-in-the-Loop (HIL) testing. The full design is shown in Figure 2, and main components of the physical setup are:

- **Actuation and Sensing:**
 - A DC motor drives the horizontal arm (table) and is controlled via a H-bridge motor driver, which allows bidirectional control of the motor torque.
 - A 600 PPR is mounted on the motor shaft to measure the horizontal arm angle θ_1 .
 - The vertical pendulum arm is instrumented with a second high-resolution encoder to measure the pendulum angle θ_2 .
- **Microcontroller Interface:**
 - An ESP32 microcontroller is used for data acquisition, signal processing, and motor control.
 - The ESP32 interfaces with the H-bridge and encoders, providing real-time communication with MATLAB/Simulink for HIL implementation.
- **Power Supply:**
 - The system operates from a 12 V DC power supply, providing sufficient current for the motor and control electronics.
- **Integration with MATLAB/Simulink:**
 - The setup is connected via USB or serial communication to the host computer.
 - The Simulink model sends control commands to the ESP32 and reads encoder feedback, allowing real-time control and visualization of the pendulum dynamics.



Figure 2. Full CAD Model.

2.2 System Parameters

The Swing Up Controller gain calculations is based on the following physical parameters extracted from the setup:

- **Pendulum mass:** $m_r = 0.018 \text{ kg}$.
- **Pendulum COM to rotation axis:** $l_r = 0.0638 \text{ m}$
- **Pendulum mass moment of inertia:** $J_r = 9.83 \times 10^{-5} \text{ kg.m}^2$
- **Distance from table rotation axis to pendulum pivot:** $l_t = 0.07 \text{ m}$

To account for the non-linearities and internal dynamics of the system, we utilized parameter estimation to identify the following DC motor and damping characteristics:

- **Rotor moment of inertia:** $J_{motor} = 0.0024 \text{ kg.m}^2$
- **Armature inductance:** $L_a = 9.1523 \times 10^{-4}$
- **Motor constants:** $k_t = k_b = 0.048 \text{ V/rpm}$
- **Motor resistance:** $R_a = 5.8466 \Omega$
- **Frictions:** $b_{motor} = 0.0055 \frac{\text{N.m.s}}{\text{rad}}$, $b_{pendulum} = 3.82 \times 10^{-7} \frac{\text{N.m.s}}{\text{rad}}$,
 $b_{table} = 6.24 \times 10^{-4} \frac{\text{N.m.s}}{\text{rad}}$

Identifying these friction coefficients through estimation rather than theoretical calculation was essential for the LQR state-space representation, as it allows the controller in stabilization phases.

2.3. System Model

After applying Lagrangian equations, the following dynamic equations are obtained [1]:

$$(J_r + m_r l_r^2) \ddot{\theta}_2 - m_r l_t l_r \cos \theta_2 \ddot{\theta}_1 + b_p \dot{\theta}_2 - \frac{1}{2} (J_r + m_r l_r^2) \sin 2\theta_2 \dot{\theta}_1^2 - m_r g l_r \sin \theta_2 = 0$$

$$(J_t + m_r l_t^2 + (J_r + m_r l_r^2) \sin^2 \theta_2) \ddot{\theta}_1 - m_r l_t l_r \cos \theta_2 \ddot{\theta}_2 + m_r l_t l_r \sin \theta_2 \dot{\theta}_2^2 + (J_r + m_r l_r^2) \sin 2\theta_2 \dot{\theta}_2 \dot{\theta}_1 + b_t \dot{\theta}_1 = T$$

Where:

- θ_1 : table angle as shown in figure 3.
- θ_2 : rod angle
- T : motor torque

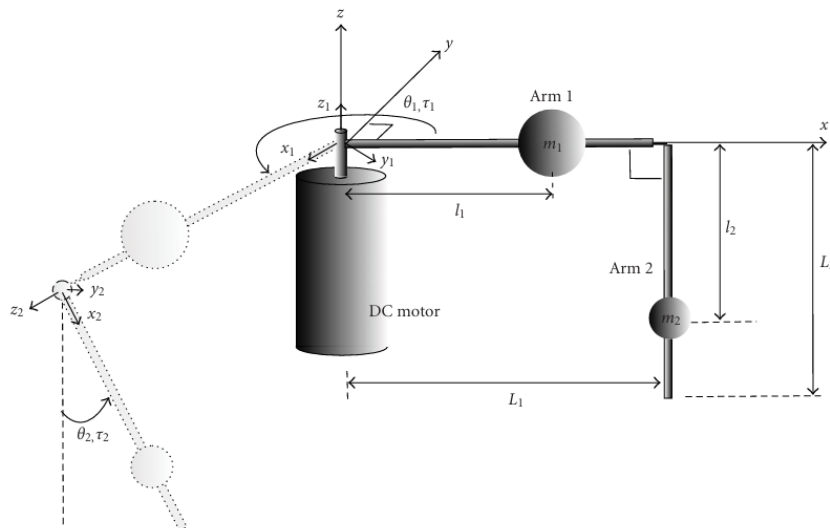


Figure 3. System Variables.

Linear-Time-Invariant (LTI) Differential Equations:

$$J_r^e \ddot{\theta}_2 - m_r l_t l_r \ddot{\theta}_1 + b_p \dot{\theta}_2 - m_r g l_r \sin \theta_2 = 0$$

$$J_t^e \ddot{\theta}_1 - m_r l_t l_r \ddot{\theta}_2 + b_t \dot{\theta}_1 = T_t - T_c$$

Where:

- $J_r^e = J_r + m_r l_r^2$
- $J_t^e = J_t + m_t l_t^2$

$$V(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + K_b \omega(t)$$

Where:

- $V(t)$: Applied terminal voltage.
- $\omega(t)$: Angular velocity of the motor shaft ($\dot{\theta}_1$).

$$T(t) = K_t i_a(t)$$

Where:

- K_t : Torque Constant
- $T(t)$: Torque

$$J_{motor} \frac{d\omega(t)}{dt} + b_{motor} \omega(t) = T(t)$$

∴ State vector reordered as:

$$x = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix}$$

and the **A** and **B** matrices written in terms of their elements.

$$\Delta = J_r J_t^e - m_2^2 L_1^2 l_2^2$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ A_{31} & A_{33} & A_{32} & A_{34} \\ 0 & 0 & 0 & 1 \\ A_{41} & A_{43} & A_{42} & A_{44} \end{bmatrix}$$

Where the elements are:

$$\begin{aligned} A_{31} &= 0 \\ A_{32} &= \frac{gm_t^2 l_t^2 L_t}{\Delta} \\ A_{33} &= -\frac{b_t J_t^e}{\Delta} \\ A_{34} &= -\frac{b_p m_p l_p L_t}{\Delta} \\ A_{41} &= 0 \\ A_{42} &= \frac{gm_p l_p J_r^e}{\Delta} \\ A_{43} &= -\frac{b_t m_p l_p L_t}{\Delta} \\ A_{44} &= -\frac{b_r J_r^e}{\Delta} \end{aligned}$$

$$B = \begin{bmatrix} 0 \\ B_{31} \\ 0 \\ B_{41} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{J_t^e}{\Delta} \\ 0 \\ \frac{J_p^e}{\Delta} \end{bmatrix}$$

2.4. Controller Requirements

Requirement	Description
Swing-Up	Move pendulum to upright position
Stabilization	Maintain pendulum upright using LQR
Sampling	≥ 0.1 kHz
Torque limits	Motor saturation constraints
Robustness	Against small disturbances

2.5. Swing-Up Controller (Energy Control)

The swing-up phase sequentially adds energy to the system by considering the energy of the pendulum in the upright equilibrium position and its actual energy. Energy is supplied through oscillating movements of the horizontal arm, which depend on the pendulum's position and velocity. It's implemented in Simulink as shown in Figure 4.

Energy Equation:

The normalized Energy E is determined by:

$$E = \frac{1}{2} J_r \dot{\theta}_1^2 + m_r g l_r (1 + \cos \theta_1)$$

The control action can be determined by:

$$u = \text{sat}(k_{\text{swing}}(E_r - E)\dot{\theta}_1 \cos \theta_1)$$

Where:

- E_r : The desired energy in the up-right equilibrium position, which is equal to:

$$E_r = 2m_r g l_r$$

- k_{swing} : Tunable coefficient that determines how fast the pendulum reaches the vicinity of the equilibrium
- $\dot{\theta}_1 \cos \theta_1$: This term ensures the torque is applied according to the direction of movement $\dot{\theta}_1$

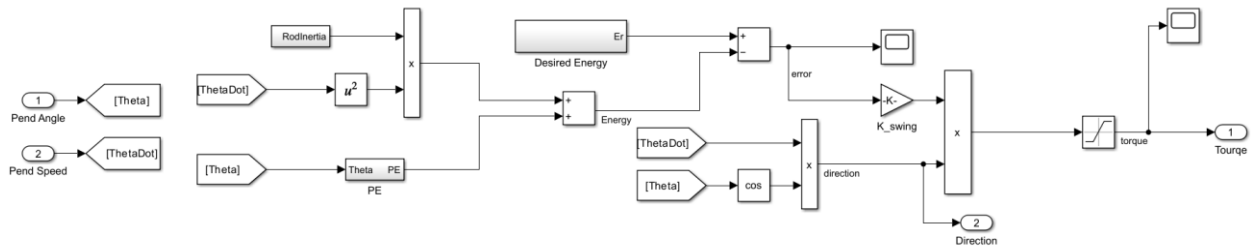


Figure 4. Swing-up Model in Simulink.

2.6. Balancing Controller (LQR)

a) System Modeling of the Furuta Pendulum

The Furuta pendulum is a nonlinear, underactuated mechanical system consisting of a rotary arm driven by a motor and a pendulum attached to the arm, free to rotate in a vertical plane. The system exhibits strong nonlinear coupling between the rotary arm and pendulum dynamics, particularly near the upright equilibrium position.

A Simscape Multibody (SimMechanics) model of the Furuta pendulum was constructed to represent the system physically. The model includes:

- Geometric properties and link masses
- Moments of inertia
- Center-of-mass locations
- Damping coefficients to model friction and energy dissipation

This model provides a realistic simulation of the system without requiring manual derivation of differential equations.

b) Linearization and State-Space Representation

To enable LQR controller design, the nonlinear model must be linearized about a specific operating point. The control objective in this work is stabilization of the pendulum in the upright equilibrium position, which is inherently unstable.

Rather than performing manual analytical linearization, the complete nonlinear Simulink model—constructed based on the Lagrange-derived equations and including actuator dynamics—was linearized using **Simulink Model Linearizer**. This approach allows the linear model to account for system interconnections and non-ideal effects present in the simulation.

The system was linearized around the upright equilibrium, resulting in the continuous-time state-space model:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

with the state vector defined as:

$$x = [\theta_1 \quad \dot{\theta}_1 \quad \theta_2 \quad \dot{\theta}_2]^T$$

where u represents the motor torque input.

The obtained matrices A, B, C, D describe the local linear dynamics of the system near the upright configuration.

c) Concept of Linear Quadratic Regulator (LQR)

The Linear Quadratic Regulator is an optimal state-feedback control strategy designed to stabilize linear systems while minimizing a predefined performance index. Unlike classical pole-placement methods, LQR determines the feedback gains by solving an optimization problem, ensuring a systematic trade-off between performance and control effort.

The LQR problem is formulated by defining the quadratic cost function:

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt$$

where:

- $\mathbf{Q} \geq 0$ penalizes state deviations,
- $\mathbf{R} > 0$ penalizes control effort.

For the Furuta pendulum, this formulation allows prioritizing stabilization of the pendulum angle while preventing excessive motor torque.

d) Optimal Control Law

The optimal control input that minimizes the cost function is given by the state-feedback law:

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$$

where the gain matrix is \mathbf{K} .

e) Closed-Loop System Behavior

Applying the LQR feedback gain modifies the system dynamics to:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x}$$

The eigenvalues of the closed loop system matrix lie in the left half of the complex plane, ensuring asymptotic stability of the upright pendulum position.

f) Control Design Approach Used in This Work

The LQR controller was designed using the following approach:

1. Construct a Simscape Multibody (SimMechanics) model of the Furuta pendulum with defined physical parameters and estimated damping coefficients. Then add actuation and sensing to the joints as shown in Figure 5.

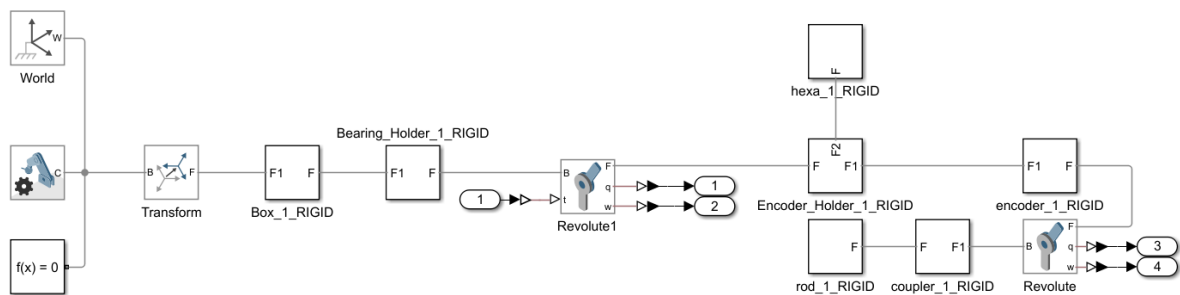


Figure 5. Exported Model.

2. Linearize the complete Simulink model around the upright equilibrium using Simulink Model Linearizer.

- Model Input and Outputs

Edit model I/Os...

Select linearization I/Os by right clicking on the desired line in your Simulink model.

Active	Block	Port	Signal Name	Bus Element	Configuration
<input checked="" type="checkbox"/>	Trial_5/Inport	1			Input Perturba
<input checked="" type="checkbox"/>	Trial_5/Subsystem	1	Motor Angle		Output Measu
<input checked="" type="checkbox"/>	Trial_5/Subsystem	2	Motor Speed		Output Measu
<input checked="" type="checkbox"/>	Trial_5/Subsystem	3	Pen. Angle		Output Measu
<input checked="" type="checkbox"/>	Trial_5/Subsystem	4	Pen. Speed		Output Measu

Help Highlight Refresh Signal Names Delete

Figure 6. Linearization I/O.

- Operating point as the initial conditions are set to be the upright position where $\alpha = 0$ (or at time = 0 if the model is upright by default as shown in Figure 7).

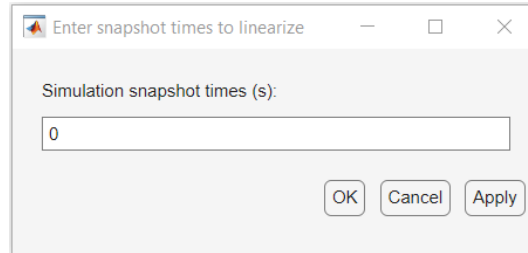


Figure 7. Operating Point Setting.

- Obtain the state-space matrices A, B, C, D.
- Design an LQR controller by selecting suitable weighting matrices Q and R.
 - The weighting matrix Q was selected to heavily penalize deviations in the pendulum angle, reflecting the primary objective of stabilizing the pendulum in the upright position. Lower weights were assigned to the rotary arm states to allow necessary motion for stabilization.
 - The control weighting R was chosen to limit excessive motor torque and avoid actuator saturation.
- The state-feedback gain was applied to the system using the LQR control law.

g) Results

$$\text{System Matrix (A)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1.3323 \times 10^{-12} & -491.2776 & 96.3299 & -0.0531 \\ 0 & 0 & 0 & 1 \\ 9.3259 \times 10^{-12} & -350.1650 & 169.7849 & -0.0934 \end{bmatrix}$$

$$\text{Input Matrix (B)} = \begin{bmatrix} 0 \\ 12991.5328 \\ 0 \\ 9259.8962 \end{bmatrix}$$

$$\text{State Weighing Matrix (Q)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

$$R=100$$

State Feedback Gain Matrix (K) = $[-0.0100 \quad -0.0857 \quad 1.5880 \quad 0.1478]$

$$u(t) = -[-0.0100 \quad -0.0857 \quad 1.5880 \quad 0.1478] \begin{bmatrix} \theta \\ \dot{\theta} \\ \alpha \\ \dot{\alpha} \end{bmatrix}$$

2.7. Implementation Details

The Stateflow chart is responsible for managing the controller phases and generating a live status signal for monitoring. It allows a discrete supervisory logic on top of continuous Simulink dynamics and acts as the interface between the continuous dynamics of the Furuta pendulum and the discrete logic of the control strategy. The full model and blocks details are shown in figures 8, 9 and 10.

1. States:

1. **Swing_Up:** The pendulum is in the downward stable position and energy is being injected to reach the upright position. (Default state)
2. **LQR:** The pendulum is near the upright equilibrium, and the linear controller stabilizes it.
3. **Recovery:** Handles situations where the pendulum deviates significantly from upright or hardware safety conditions are triggered.

2. Transitions

1. **Swing_Up → LQR:** Triggered when the pendulum angle is sufficiently close to upright (threshold-based) and the energy converges to the desired level.
2. **LQR → Recovery** (optional): Triggered if the pendulum falls outside a predefined stability region.
3. **Recovery → Swing_Up:** Resets the control cycle if safe conditions are restored.

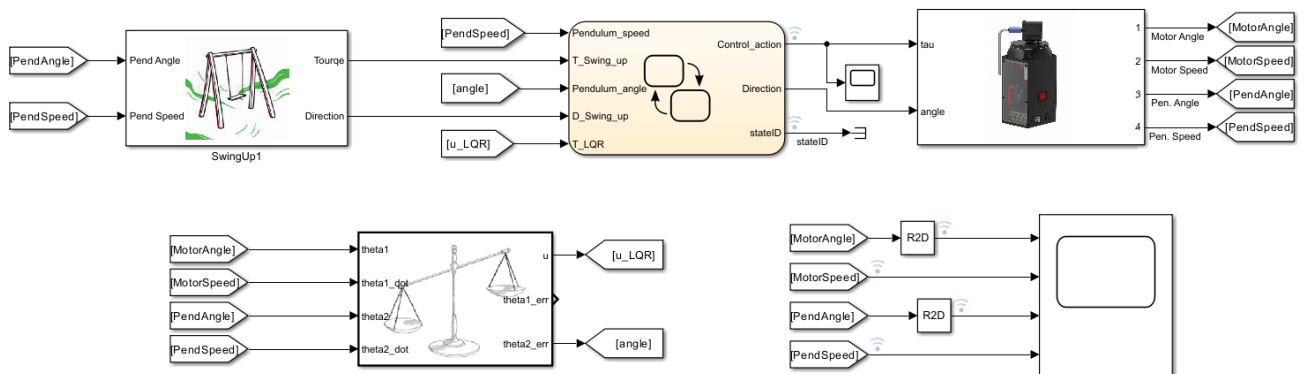


Figure 8. Full Simulink Model.

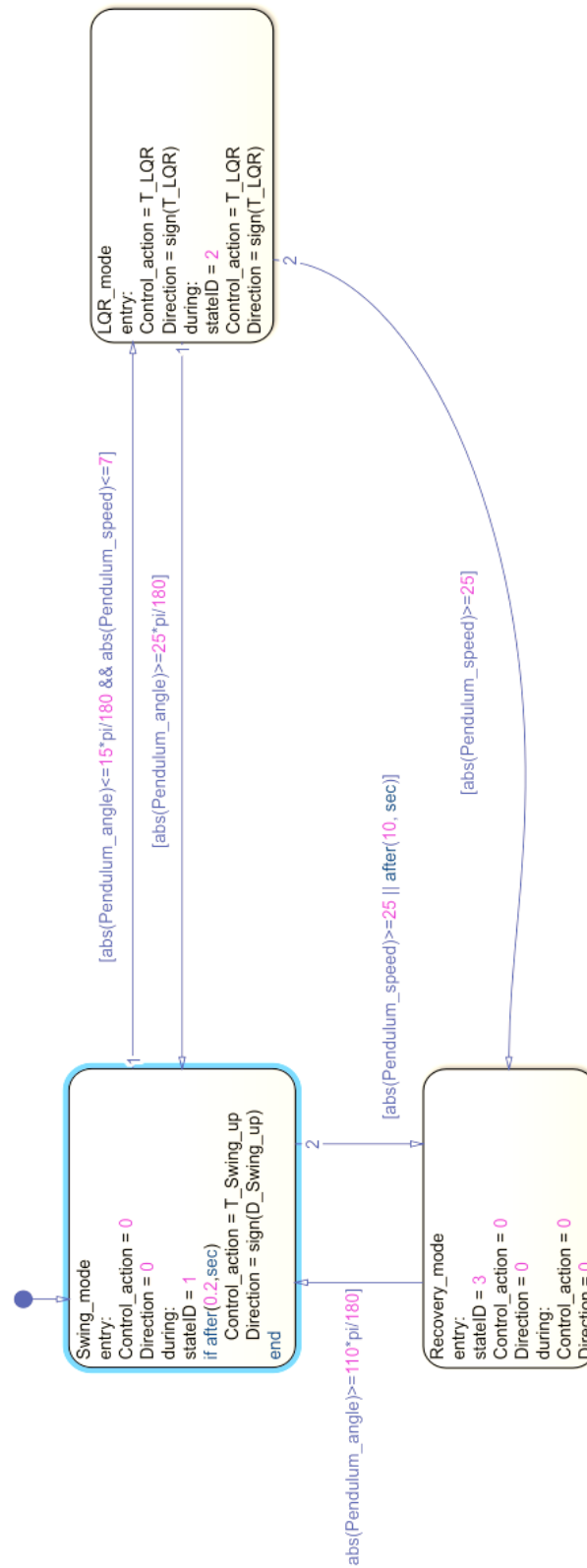


Figure 9. State Diagram.

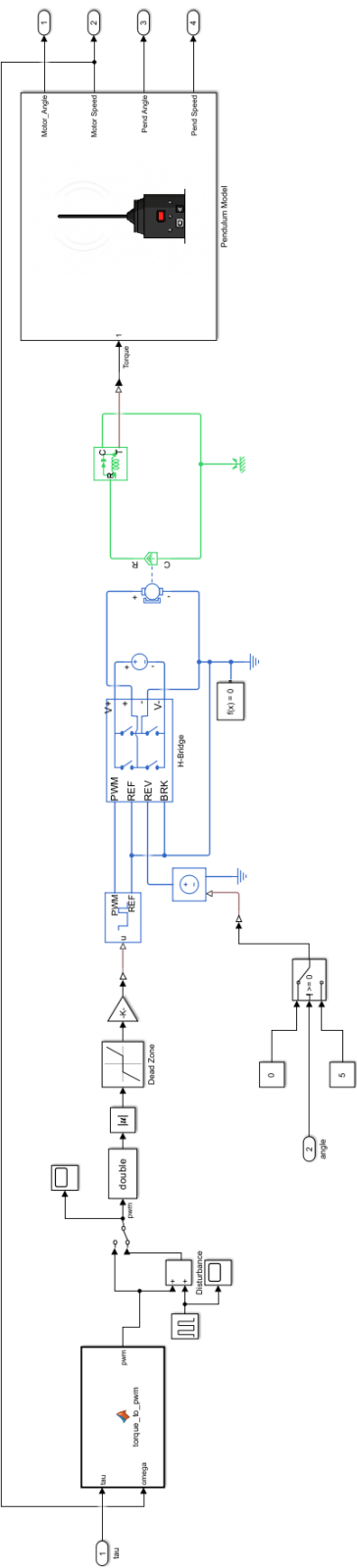


Figure 10. Futura Pendulum Model with Motor.

3. Simulation Results

The closed-loop behavior of the Furuta Pendulum was analyzed through MATLAB/Simulink simulations. The key outputs were recorded for the duration of the simulation, including the pendulum angle (θ_2), pendulum angular velocity ($\dot{\theta}_2$), motor angle (θ_1), and motor angular velocity ($\dot{\theta}_1$).

Figure 11 shows the time responses of these signals when a small initial disturbance is introduced to the system. It can be observed that:

- The pendulum successfully transitions from the downward to the upright position (swing-up phase) and stabilizes under the LQR controller.
- The motor angle and velocity adjust accordingly to supply the necessary torque while maintaining stability.
- The responses demonstrate smooth behavior with minimal overshoot and settling times compatible with the control objectives.

These results validate the designed swing-up and balancing control strategy in simulation before HIL implementation.

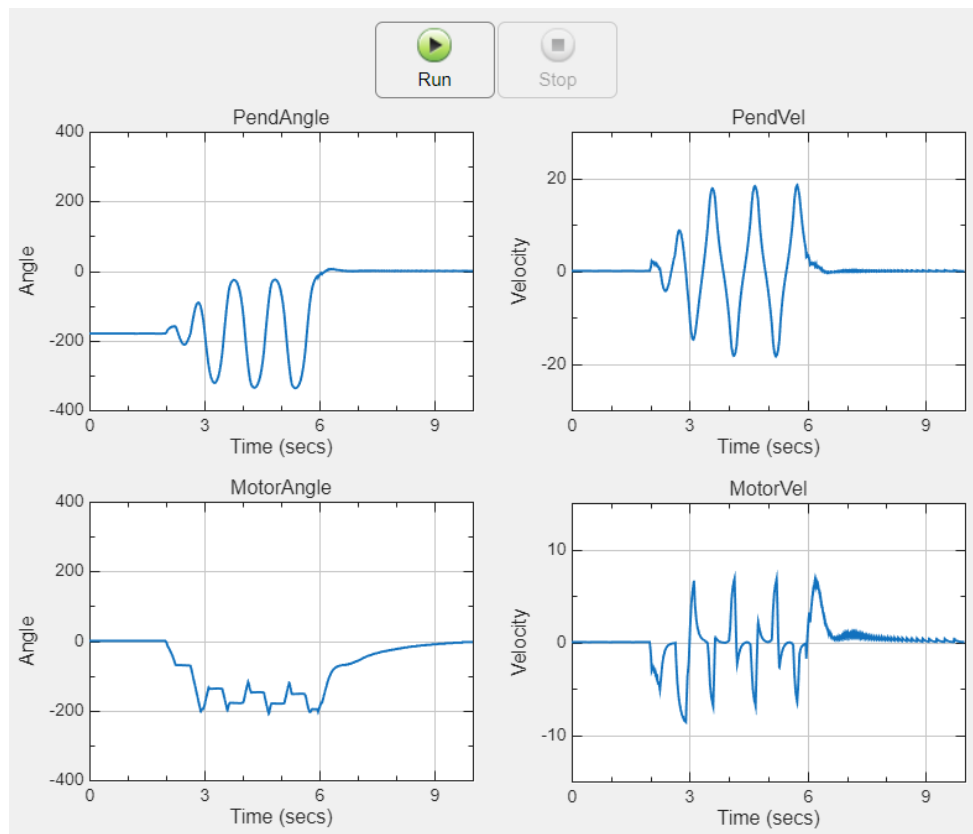


Figure 11. Simulation Results.

4. Real-Life Results

The performance of the Furuta Pendulum was evaluated both in simulation and on the physical setup under identical control parameters. The comparison between simulated and real-world results highlights the effectiveness of the implemented controllers as well as the limitations of the simulation model.

Observations:

1. Overall Behavior:

- In both simulation and real-life experiments, the pendulum successfully transitions from the downward stable position to the upright equilibrium (swing-up) and remains stabilized under the LQR controller.
- The control strategy shows consistent qualitative behavior in both cases, indicating that the system model and parameter estimation provide a reasonable approximation of the real system dynamics.

2. Oscillations:

- Real-life measurements exhibit slightly higher oscillations compared to simulation results, as shown in Figure 12.
- This discrepancy is mainly due to unmodeled effects in the physical system, such as joint backlash, external disturbances, and high-frequency friction not captured in the simplified model.

3. Settling Time:

- The pendulum takes longer to settle in the actual setup compared to simulation.
- Factors contributing to this include additional damping effects, motor nonlinearities, and minor delays in the actuation system that are not fully accounted for in the simulation.

4. Damping and Friction:

- Real-life damping appears more effective, possibly due to unmodeled air resistance, internal mechanical damping, and slight stiction in the joints.
- While the simulation includes estimated friction coefficients, the real system contains additional complexities that slightly alter transient responses.

5. Control Signal:

- The control torque applied in real life follows the same trend as in simulation but may have slightly higher peaks or variations due to actuator limitations and measurement noise.
- The overall magnitude and direction of control actions remain consistent, validating the LQR and swing-up controller design.

6. Model Simplifications:

- The simulation model is simplified to allow linearization and controller design, which excludes certain higher-order dynamics and unmodeled parasitic effects present in the real system.
- Despite these simplifications, the key dynamic behavior of the pendulum is well captured, confirming the reliability of the modeling and control approach.

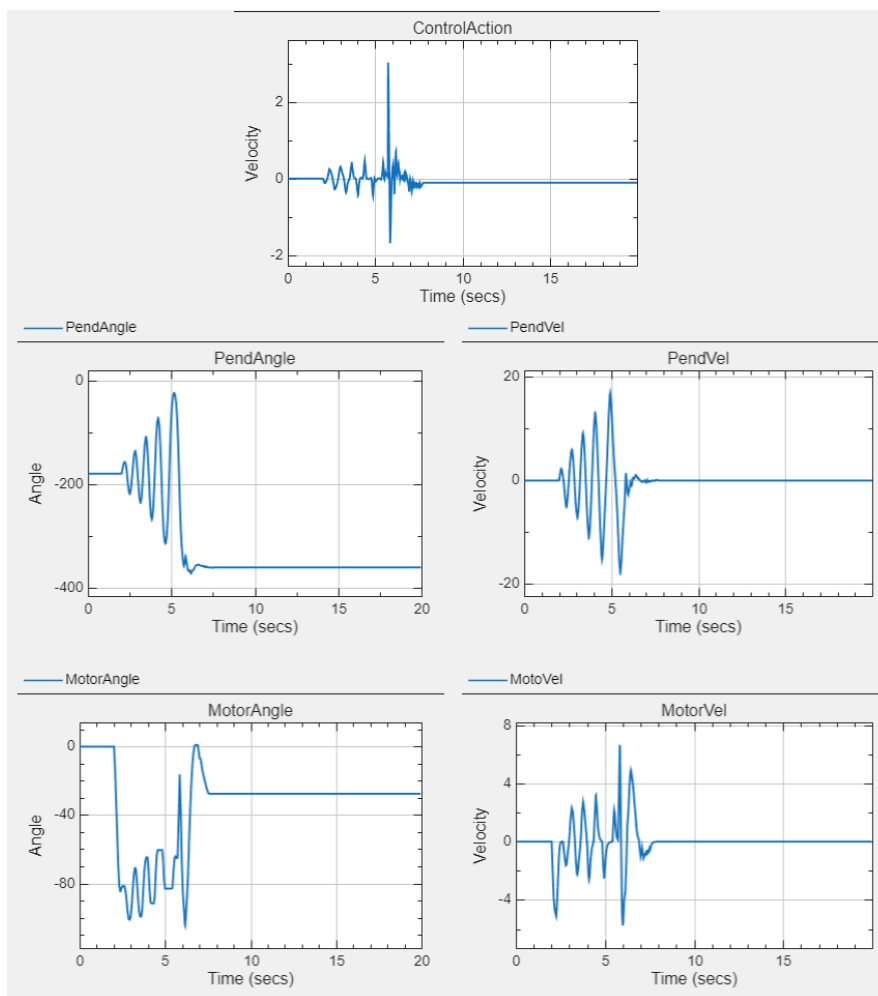


Figure 12. Real-Life Results.

5. Conclusion

While the real system exhibits slightly higher oscillations, increased settling time, and minor variations in control signals compared to simulation, the overall system response is in good agreement with the predicted behavior. This validates both the energy-based swing-up strategy and the LQR stabilization approach. Further refinements to the model, including more detailed friction, damping, and motor nonlinearities, could reduce these discrepancies in future implementations.

6. References

- [1] B. Yao, "Modeling of Rotary Inverted Pendulum," 2022. [Online].