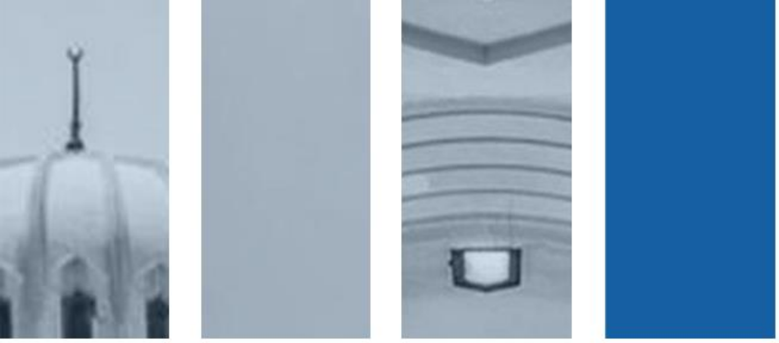


ROTARY INVERTED PENDULUM SUBMISSION

Program: MCTA

Course Code: MCT411

Course Name: Hybrid Control Systems

Examination Committee

Dr. Mohamed Ibrahim Awad
Dr. Diaa Emad Abdel Fattah
Eng. Hesham Salah

Ain Shams University
Faculty of Engineering
International Credit Hours Engineering Programs (I-CHEP)
Fall Semester – 2023

Program: MCTA

Course Code: MCT411

Course Name: Hybrid Control Systems

AIN SHAMS UNIVERSITY
I-Credit Hours Engineering Programs
(i.CHEP)

Student Personal Information for Group Work

Student Names:	**Student Codes:**
Ahmed Hisham Hassabou	19P4007
Omar Ahmed Elsayed	19P8638
Eslam Elhusseini	19P9651
Nourhan Emad Sayed	19P8403
Gamal Hazem	19P8288

Table of Contents

LIST OF FIGURES

Real Time Optimal Control for Rotary Inverted Pendulum

Ahmed Hassabou, Nourhan Emad, Omar Ahmed, Gamal Hazem and Eslam Elhusseini
Ain Shams University, Faculty of Engineering, Mechatronics and Automation

Abstract: Due to its highly nonlinear, multivariable, and intrinsically unstable dynamics, the rotary inverted pendulum system provides a substantial difficulty. This system is an important tool for teaching current control theory and acts as a great testbed for assessing various control design strategies. The fundamental goals of this research are to build a genuine rotary inverted pendulum using the Euler-Lagrange technique, and to construct a controller program capable of autonomously self-erecting and balancing the rotary inverted pendulum. The study describes a feasible method for applying real-time control to allow the pendulum to self-erect from a lower position and maintain its vertical-upright balance. For the self-erecting phase, we use an energy-based Proportional-Derivative (PD) controller, and for the balancing phase, we use a Linear Quadratic Regulator (LQR) controller. To demonstrate the efficiency of these control measures, we present complete data from both computer simulations and experimental tests. Both simulations and experiments show that the suggested control mechanism for the rotary inverted pendulum system is efficient and effective.

Introduction

The Rotary Inverted Pendulum (RIP) is a challenging task in the field of control systems, used to assess and illustrate the effectiveness of control algorithms. This system is distinguished by its simple structure, multidimensional nature, and intrinsic instability, as well as several nonlinear characteristics.

Prior research has explored various approaches to balance the rotary inverted pendulum. Some studies have applied linear control theory [1], energy-based control methods [2], and balancing sequences to raise the pendulum from its stable equilibrium. Zhong and Rock [3] have employed a linear quadratic regulator to optimize control gains in feedback controllers. For a single inverted pendulum, a straightforward bang-bang method, as described by Astrom and Furuta [4], can be utilized to regulate the system's energy content.

The goal of this research is to present an energy-based control strategy for swinging a pendulum from a lower position to a stable upright condition. This method is based on a precise nonlinear mathematical model of a rotating inverted pendulum. Following that, the LQR control approach is used as the principal controller to stabilize the rotating inverted pendulum, guaranteeing that the pendulum remains upright, and the arm remains horizontal. This is accomplished using state feedback control, which converts the linear system's unstable poles to stable poles.

All tests were carried out in real-time utilizing a physical rotary inverted pendulum system constructed by the team, as well as simulations using the mathematical model of the rotary inverted pendulum.

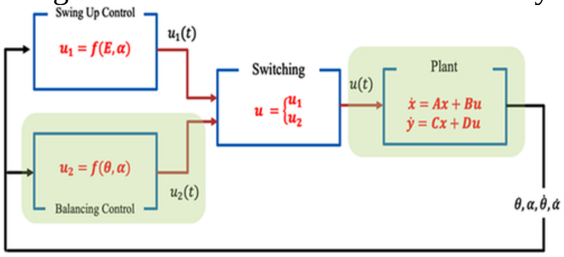


Figure 1: Rotary inverted pendulum control strategy

MODELING THE ROTARY INVERTED PENDULUM

The RIP system is made up of numerous critical components, including a controller, an arm, a pendulum, an actuator (particularly, a DC motor), and two incremental rotary encoders. The major goal of this system is to use the controller to keep the pendulum upright atop the rotary arm by modifying the position of the arm, which is supported by a base. The DC motor provides the power to rotate the arm, while the two rotary encoders detect and monitor the angular locations of both the pendulum and the arm.

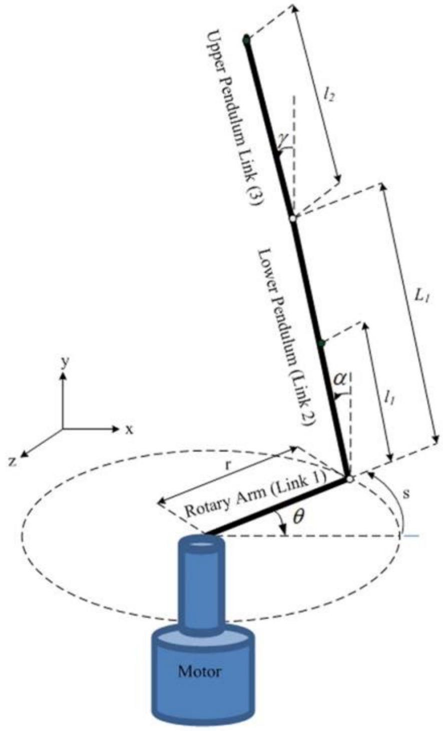


Figure 2: Reference frames and parameters of rotary inverted pendulum

Figure 2 shows a schematic illustration of a rotating inverted pendulum system. The design reduces the pendulum to a concentrated mass positioned at its halfway. In the above picture, displacement of the pendulum is represented by an angle along the y-axis, with representing the angular position in the x-axis plane. The velocity of the pendulum's center of mass is examined to construct a mathematical model of the system. In the modelling process, the following assumptions are critical:

- The system is initially in equilibrium, which is to say, the starting conditions are assumed to have zero angular position and velocity.
- The pendulum's movement is restricted to a minimal angular displacement from the vertical axis to be consistent with a linear model.
- The pendulum can receive a slight disturbance, which is a common scenario in control systems to assess stability and response.

Position: $z = L_p \sin\alpha$, $\dot{x} = L_a \cos\theta$, $\dot{y} = L_a \sin\theta$, $\tan\gamma = \dfrac{l_p \sin\theta}{L_a}$, $\grave{L_a} = \sqrt{(l_p \sin\alpha)^2 + L_a{}^2}$

$x = \grave{L_a}\cos(\theta - \gamma)$, $y = \grave{L_a}\sin(\theta - \gamma)$

$x = L_a \cos\theta + l_p \sin\alpha \sin\theta$, $y = L_a \sin\theta - l_p \sin\alpha \cos\theta$

$$\therefore \text{Velocity:}\quad v = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -L_a\dot{\theta}\sin\theta + l_p\dot{\alpha}\cos\alpha\sin\theta + l_p\dot{\theta}\sin\alpha\cos\theta \\ L_a\dot{\theta}\cos\theta - l_p\dot{\alpha}\cos\alpha\cos\theta + l_p\dot{\theta}\sin\alpha\sin\theta \\ l_p\dot{\alpha}\sin\alpha \end{bmatrix}$$

l_p: Distance to the center of gravity of the pendulum

L_a: Length of the arm , m_p: Mass of the pendulum

j_a: Inertia of the arm , j_p: Inertia of the pendulum

θ: Angle of the arm , α: Angle of the pendulum

b_a: Damping of the arm , b_p: Damping of the pendulum

When setting system requirements, the goal is to have a quick settling time with little overshoot in the pendulum's reaction to disturbances. The Rotary Inverted Plant model is built in line with available literature and resources in the public domain [5][6]. The model has two main components for estimating the velocity of the accumulated mass of the Pendulum. Lagrange's equation is used to determine the kinematic equations of motion:

Lagrange equation: $L = K_E - P_E \rightarrow (1)$, $\dfrac{d}{dt}\left(\dfrac{dL}{d\dot{\theta}}\right) - \dfrac{dL}{d\theta} + \dfrac{dR}{d\dot{\theta}} = \tau \rightarrow (2)$, $\dfrac{d}{dt}\left(\dfrac{dL}{d\dot{\alpha}}\right) - \dfrac{dL}{d\alpha} + \dfrac{dR}{d\dot{\alpha}} = 0 \rightarrow (2^*)$

Kinetic Energy: $K_E = \dfrac{1}{2}\left(m_p L_a{}^2\dot{\theta}^2 + j_a\dot{\theta}^2 + m_p l_p{}^2 \sin^2\alpha\,\dot{\theta}^2 + m_p L_p{}^2\dot{\alpha}^2 + j_p\dot{\alpha}^2\right) + m_p L_a l_p \dot{\theta}\dot{\alpha}\cos\alpha \rightarrow (3)$

Potential Energy: $P_E = -m_p g l_p \cos\alpha \rightarrow (4)$

$R = \dfrac{1}{2}b_a\dot{\theta}^2 + \dfrac{1}{2}b_p\dot{\alpha}^2$ Let α be very small $\cong 0$

$\dfrac{dR}{d\dot{\theta}} = b_a\dot{\theta} \rightarrow (a)$, $\dfrac{dR}{d\dot{\alpha}} = b_p\dot{\alpha} \rightarrow (b)$

$\dfrac{d}{dt}\left(\dfrac{dL}{d\dot{\theta}}\right) = m_p L_a{}^2\ddot{\theta} + j_a\ddot{\theta} + m_p L_a l_p \ddot{\alpha}$, $\dfrac{d}{dt}\left(\dfrac{dL}{d\dot{\alpha}}\right) = m_p L_p{}^2\ddot{\alpha} + j_p\ddot{\alpha} + m_p L_a l_p \ddot{\theta}$

$\dfrac{dL}{d\theta} = 0$, $\dfrac{dL}{d\alpha} = m_p g l_p \alpha$

$$(m_pL_a{}^2 + j_a)\ddot{\theta} + b_a\dot{\theta} + (m_pL_al_p)\ddot{\alpha} = \tau \rightarrow (1^{**})$$

$$(m_pL_p{}^2 + j_p)\ddot{\alpha} + b_p\dot{\alpha} + (m_pL_al_p)\ddot{\theta} - (m_pgl_p)\alpha = 0 \rightarrow (2^{**})$$

From (2^{**}): $\ddot{\alpha} = \dfrac{-b_p\dot{\alpha} + (m_pL_al_p)\ddot{\theta} - (m_pgl_p)\alpha}{m_pl_p{}^2 + j_p}$ We will assume b_p very small

Put in (1^{**}): $\ddot{\theta} = -\left(\dfrac{b_ax_2}{x_3x_4}\right)\dot{\theta} - \left(\dfrac{x_2x_5}{x_3x_4x_6}\right)\alpha + \left(\dfrac{1}{x_3x_4}\right)\tau \rightarrow (1^{***})$

$$\ddot{\alpha} = \left(\frac{b_ax_2{}^2}{x_3x_4x_4}\right)\dot{\theta} + \left(\frac{x_2{}^2x_5x_6 + x_5x_3x_4x_6{}^2}{x_3x_4x_6{}^3}\right)\alpha - \left(\frac{x_2}{x_3x_4x_6}\right)\tau \rightarrow (2^{***})$$

$x_2 = m_pL_al_p$, $x_3 = \dfrac{x_4x_6 - x_2{}^2}{x_4x_6}$, $x_4 = m_pL_a{}^2 + j_a$, $x_5 = m_pgl_p$, $x_6 = m_pL_p{}^2 + j_p$

Therefore, the concluded DC Motor equation is:

$$\tau = \frac{n_mn_gk_tk_g(V_m - k_gk_m\dot{\theta})}{R_m}$$

Motor Parameters Notations:

τ = Motor torque , $\mathrm{k_t}$ = Torque constant

$\mathrm{k_g}$ = Motor gear ratio , $\mathrm{n_g}$ = Gearbox efficiency

$\mathrm{k_m}$ = Back EMF constant , $\mathrm{n_m}$ = Motor efficiency

$\mathrm{R_m}$ = Armature resistance , $\mathrm{V_m}$ = Armature voltage

let $G_1 = \dfrac{n_mn_gk_tk_g}{R_m}$ and $G_2 = \dfrac{n_mn_gk_tk_g{}^2k_m}{R_m}$

The Implemented State Space Model According to the previously defined equations:

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\\ \dot{\alpha}\\ \ddot{\alpha}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & A_1 & A_2 & 0\\ 0 & 0 & 0 & 1\\ 0 & A_3 & A_4 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ \alpha\\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix}0\\ A_5\\ 0\\ A_6\end{bmatrix}V_m \quad , \quad \begin{bmatrix}\theta\\ \alpha\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ \alpha\\ \dot{\alpha}\end{bmatrix}$$

$$\begin{bmatrix}\dot{\theta}\\ \dot{\alpha}\\ \ddot{\theta}\\ \ddot{\alpha}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & A_1 & A_2 & 0\\ 0 & A_3 & A_4 & 0\end{bmatrix} * \begin{bmatrix}\theta\\ \alpha\\ \dot{\theta}\\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ A_5\\ A_6\end{bmatrix}V_{m,} \begin{bmatrix}\theta\\ \alpha\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix} * \begin{bmatrix}\theta\\ \alpha\\ \dot{\theta}\\ \dot{\alpha}\end{bmatrix}$$

Such that:

$A_1 = \dfrac{-b_ax_2}{x_3x_4}G_2$, $A_2 = \dfrac{-x_2x_5}{x_3x_4x_6}$, $A_3 = \dfrac{b_ax_2{}^2}{x_3x_4x_6}G_2$

$A_4 = \dfrac{x_2{}^2x_5x_6 + x_3x_4x_5x_6{}^2}{x_3x_4x_6{}^3}$, $A_5 = \dfrac{G_1}{x_3x_4}$, $A_6 = \dfrac{-x_2}{x_3x_4x_6}G_1$

SYSTEM IDENTIFICATION

The lack of a datasheet might be a significant hindrance. Datasheets offer critical specifications and operational characteristics needed for component integration and use. As a result, in the absence of these tools, we must rely on empirical procedures such as parameter estimation to obtain these values. To approximate the characteristics of the component or system, testing and observation are utilized.

Identifying known variables and constraints is an important step in parameter estimation, particularly when utilizing tools like MATLAB's Parameter Estimator. In the context of the specified model transfer function, adherence to physical principles and practical issues is crucial. For example, the concept that a motor's moment of inertia (J) would be negative is fundamentally nonsensical; so, constraints must be placed to ensure that the estimated values are plausible. Additionally, certain model variables may be readily visible or accessible via experiments, contributing to a more realistic and trustworthy parameter estimation process. Armature resistance (R), for example, is commonly directly or accurately approximated. By utilizing known values as constraints, the optimization problem becomes more well-defined, and the parameter estimation work becomes more tractable. Finally, defining known variables and constraints increases the realism and reliability of parameter estimation in MATLAB's Parameter Estimator tool.

Motor Parameters Modeling

The Simulink block diagram for a motor encapsulates the dynamic interactions between its electrical and mechanical components. At the core is a voltage source, representing the input signal applied to the motor, which could be a step, sinusoidal, or another relevant input. The electrical section includes blocks for the armature resistance (R) and the back electromotive force (Kb), reflecting the electrical properties of the motor. The torque generation block incorporates the torque constant (Kt), illustrating the conversion of electrical signals into mechanical torque. On the mechanical side, components like inertia (J) and damping (D) represent the motor's resistance to changes in rotational motion. The block diagram may also account for external mechanical loads if present. This comprehensive block diagram serves as a powerful tool to simulate and analyze the dynamic behavior of the motor under varying conditions and control strategies within the Simulink environment.

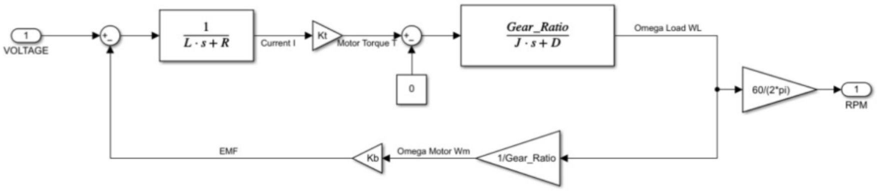


Figure 3: Motor Modeling in Simulink

Data Collection

Data collection during transient response is crucial in parameter estimation, especially in dynamic systems since it gives critical insights into the system's behavior throughout the interval of transition from one state to another. Transient response captures the dynamic properties of a system when subjected to changes in inputs or starting conditions, providing information on its time-dependent reaction and stability. Transient response is important because it reveals the underlying dynamics of the system, such as temporal constants, settling periods, and overshoots, which are essential for appropriate parameter estimation.

During transient periods, the system adapts, and the observed data covering these adjustments allows for the derivation of model parameters pertaining to inertia, damping, and other dynamic features.

Collection Methodology

A Quadrature Encoder is used by an Arduino code to operate a DC motor and offer precise movement feedback. The design connects the motor and encoder to certain pins and populates an array with a series of PWM (Pulse Width Modulation) values that control the speed and direction of the motor. The use of a timer interrupt set to occur every 5 milliseconds, which continually logs encoder position changes and indicates motor speed. When the setMotorPWM function is called, the motor's PWM value is updated depending on the array index, and when negative values are discovered, the motor's direction is flipped. When all PWM values have been iterated over, the motor is shut off. The data is then entered into an excel spreadsheet to compute the motor's velocity. The data is then sent into MATLAB's parameter estimator tool.

Parameter Estimation Sequence

In our experimental setup, we applied both +12V and -12V to the DC motor, systematically recording time elapsed, Pulse Width Modulation (PWM) values, and Revolutions Per Minute (RPM) readings. This data was then meticulously logged into an Excel file, serving as a comprehensive dataset for subsequent analysis. Upon importing the dataset into MATLAB, we leveraged the Parameter Estimator in Simulink for the purpose of refining our DC motor model. Within the Simulink environment, we constructed a model incorporating essential parameters like the motor torque constant, back emf constant, armature resistance, inertia, and inductance. Initializing the model with proposed parameter values, we initiated an iterative parameter estimation process, minimizing a cost function to align model predictions with experimental data. The convergence of this process was guided by well-defined convergence criteria. The estimated parameters extracted from this optimization endeavor were subsequently utilized in formulating the Linear Quadratic Regulator (LQR) matrix. This LQR controller, designed within Simulink, aimed to optimize the performance of the system based on the newly estimated model parameters. To validate our results, we conducted further experiments and analyzed the performance of the LQR controller, ensuring its efficacy in stabilizing the DC motor under varying conditions. This comprehensive approach ensures the accurate representation of the DC motor model and the effective utilization of estimated parameters in designing an optimized LQR controller.

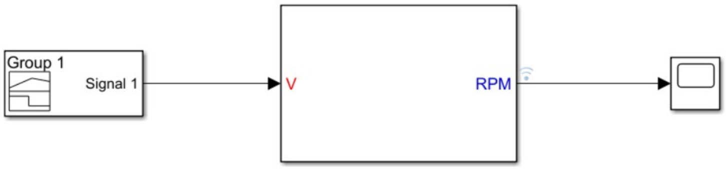
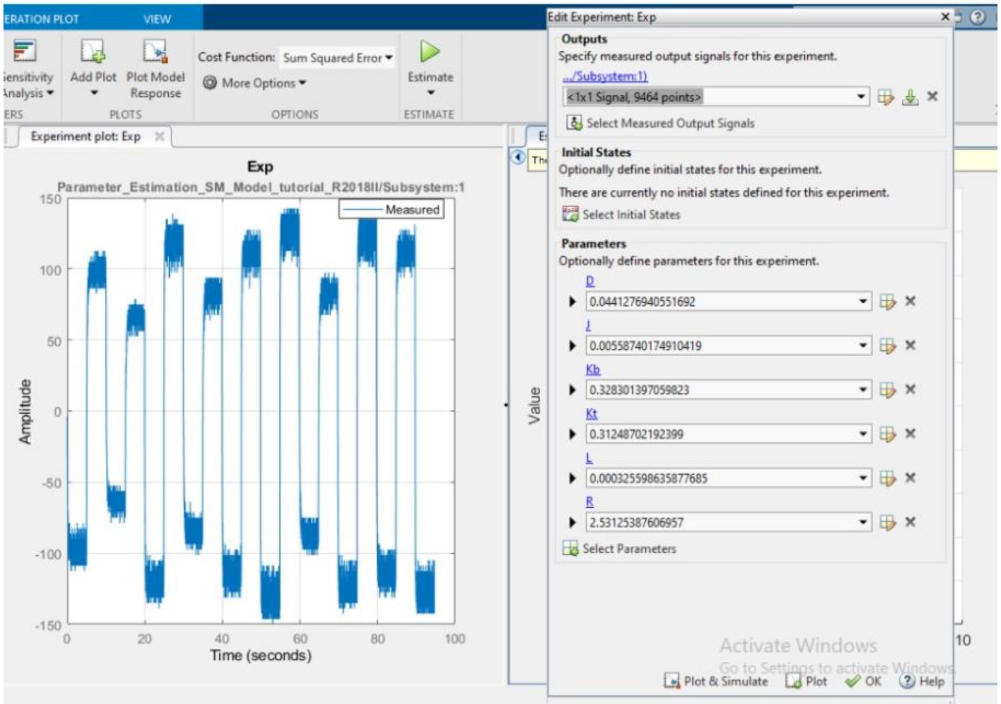


Figure 4: Motor Transfer Function



Edit Experiment: Exp

Outputs

Specify measured output signals for this experiment.

.../Subsystem:1)

<1x1 Signal, 9464 points>

Select Measured Output Signals

Initial States

Optionally define initial states for this experiment.

There are currently no initial states defined for this experiment.

Select Initial States

Parameters

Optionally define parameters for this experiment.

D: 0.0441276940551692

J: 0.00558740174910419

Kb: 0.328301397059823

Kt: 0.31248702192399

L: 0.000325598635877685

R: 2.53125387606957

Select Parameters

Plot & Simulate | Plot | OK | Help

Figure 5: Estimated Parameters After Simulation

Results

The following table summarizes the key parameters and values derived from the empirical testing and parameter estimation process:

Symbol	Description	Unit	Value
K_t	Motor torque constant	$\frac{\text{Nm}}{\text{A}}$	0.312
K_m	Back EMF constant	Vs	0.328
R_m	Armature resistance	Ω	2.53
K_g	System gear ratio	None	26:1
η_m	Motor efficiency	%	69%
η_g	Gearbox efficiency	%	90%
B_{eq}	Viscous damping coefficient	$\frac{\text{Nm}}{s}$	0.044
J_{eq}	Moment of inertia at the load	$Kg.m^2$	0.005587
m_p	Pendulum mass	Kg	0.06
m_{arm}	Arm mass	Kg	0.035
l_p	Pendulum COG length	m	0.09
l_{arm}	Arm COG length	m	0.09
r	Arm Length	m	0.18
J_p	Pendulum moment of inertia	$Kg.m^2$	0.005832
J_{arm}	Arm moment of inertia	$Kg.m^2$	0.005587

Therefore, using the physical values from Table 1, the state space form is populated as,

$$\begin{bmatrix}\dot{\theta}\\ \dot{\alpha}\\ \ddot{\theta}\\ \ddot{\alpha}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 1.1858 & -438.7442 & 0\\ 0 & 9.0638 & -71.4132 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \alpha\\ \dot{\theta}\\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0.5720\\ 0.0931\end{bmatrix} V_m$$

The Rotary Inverted Pendulum plant model design has four states, one controlled input and four outputs. All the four outputs are not measurable and so we want to do an observability test to make sure that the plant is completely state observable with the outputs which can be measured. We have two encoders to measure the pendulum angle α and the motor arm link angle θ.

$$Y = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \alpha\\ \dot{\theta}\\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix} V_m$$

With this set of measured outputs available, we form the observability matrix which is, $[CCACA^2CA^3]^T$. We used MATLAB© command $rank(obsv(A, B))$ to get the rank of the observability matrix; the matrix is full rank and hence we conclude that the system is completely state observable with the measured outputs of the pendulum angle a, and motor arm link angle, q. We also form the controllability matrix as, $[BABA^2BA^3B]$. Again, we have used MATLAB© command $rank(ctrb(A, B))$ to get the rank of the controllability matrix; it shows that the matrix is full rank and hence is controllable. The open loop plant has poles at $[0, \ -438.7447, 2.9786, \ -2.9782]$. It is observed that plant has a right half plane pole which makes the system unstable and hence we need to design a controller to stabilize it.

CONTROLLER DESIGN

Pole Placement

Following the identification of the open loop poles of the system, which included a right half plane pole indicating instability, we applied the pole placement technique to design a controller for stabilizing the system.

To ensure a robust and stable control system, specific constraints were set for the controller design. The desired performance criteria included a settling time of 2 seconds and an overshoot limit of no more than 5%. These constraints were crucial to ensure the system's response was both quick and precise without excessive oscillations.

In addition to modifying the existing poles, we introduced two extra poles as part of the controller design for this 4th order system. These additional poles were strategically placed further into the left half of the s-plane, at -10ω_n and -15ω_n, where ω_n is the natural frequency of the system. This placement was done to enhance the system's damping characteristics and to further ensure the stability and responsiveness of the control system.

The implementation of the pole placement technique, with the specified constraints and the addition of the extra poles, resulted in all closed-loop poles being in the left half plane. The closed loop poles at

[-43.4715 + 0.0000i, -28.9810 + 0.0000i, -2.0000 + 2.0974i, -2.0000 - 2.0974i]

This was a clear indication that the designed controller successfully stabilized the system. The new pole positions, combined with the added gain, effectively shifted the system's dynamics to a stable and desired performance region, aligning with our predefined control objectives.

LQR Controller

Following the exploration of the pole placement technique, we shift our focus to the Linear Quadratic Regulator (LQR), a pivotal control strategy in modern control theory. The LQR approach offers an optimal solution based on a predefined cost function, striking a balance between system performance and control effort.

The state-space representation of our rotary inverted pendulum system, characterized by matrices A, B, C, and D, serves as the foundation for the LQR design. We consider the state vector X = [Θ, Θ_dot, α, α_dot], which encapsulates the rotational arm's angle θ, the pendulum's angle α, and their respective angular velocities.

Central to the LQR methodology is the formulation of a cost function, typically denoted as $J=\int(X^TQX+u^TRu)$ dt. This function quantifies the system's state deviations and control energy expenditure, with Q and R matrices dictating the respective weightings. Our choice of these matrices is pivotal; a higher weighting in Q emphasizes state accuracy, while a larger R value penalizes excessive control effort.

In our system, the matrices Q and R were judiciously selected after careful consideration of the system dynamics and desired performance criteria. The matrix Q was chosen to prioritize the accuracy of the pendulum's angle α and rotational arm's angle θ, while R was set to regulate the control input magnitude.

Utilizing MATLAB's LQR function, the optimal gain matrix K was computed. This gain matrix is pivotal in formulating the control law $u=-Kx$, which drives the system towards a desired state efficiently. The LQR controller's primary objective is to minimize the cost function, thereby ensuring optimal performance under the constraints imposed by our system dynamics.

The implementation of the LQR controller was conducted through MATLAB/Simulink simulations. These simulations were instrumental in observing the system's behavior under the LQR regime, allowing for fine-tuning of Q and R to achieve an optimal balance between state error minimization and control effort. Comparative analysis with the previously discussed pole placement strategy revealed that the LQR approach yielded a more balanced system response. While pole placement provided us with the tools to place the system poles at desired locations, LQR offered a systematic and optimal solution considering the trade-offs between state performance and energy consumption. This was particularly evident in the system's response to disturbances and its steady-state behavior.

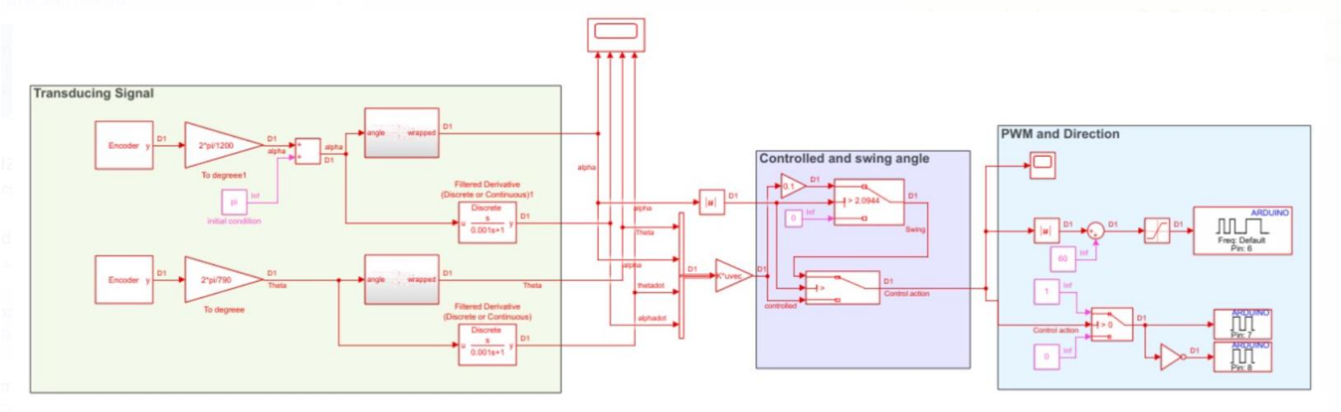


Figure 6 shows The Implementation of HIL in Simulink

CONTROLLER VALIDATION

Prior to the real-world deployment of the Linear Quadratic Regulator (LQR) controller for the rotary inverted pendulum, a rigorous validation process is implemented within the Simulink environment. This process is critical to ensure the controller's efficacy and safety in managing the dynamics of the physical system.

Hardware Specification and Simulation Integration

The validation begins with the meticulous integration of the pendulum system's hardware specifications into Simulink using the Simscape Multibody Link tool. This integration allows for a high-fidelity simulation, where the CAD model of the pendulum is translated into a virtual representation that retains the physical parameters and constraints.

This virtual prototype interacts within a simulated environment that replicates real-world physics, providing a baseline for controller evaluation.

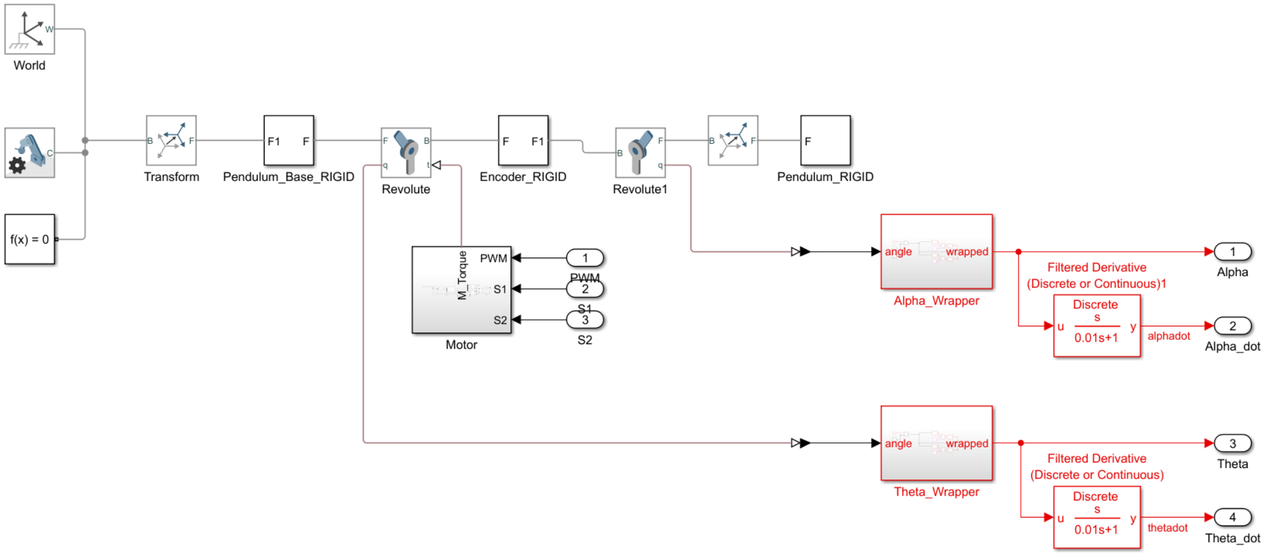


Figure 7: Simulink model showcasing the hardware specifications of the rotary inverted pendulum, integrated using Simscape Multibody Link tool.

Motor Characterization and Parameter Estimation

The motor, being the actuator of the system, is characterized within Simulink with parameters obtained from empirical estimations. This model includes the motor's electrical dynamics, such as resistance and inductance, and mechanical properties like torque constants. The accuracy of this motor model is crucial as it directly affects the fidelity of the control action in the simulation.

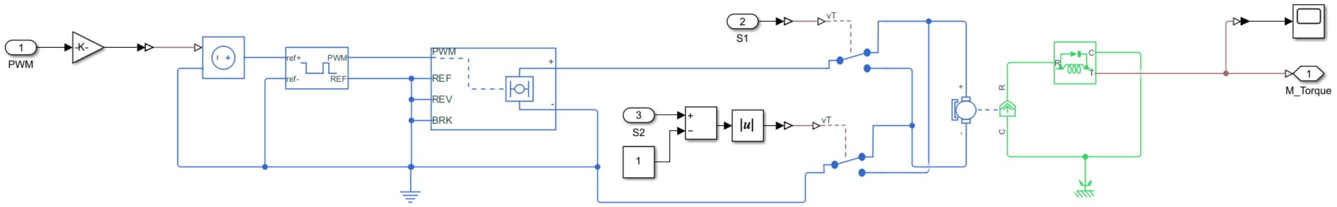


Figure 8: Detailed representation of the motor model in Simulink, parameterized using values obtained from empirical estimations.

Angle Normalization

To address the periodic nature of angular measurements, an 'Angle Wrapper' block is implemented. This block normalizes the angle of the pendulum within a range of [−π, π], ensuring consistent angle representations for control purposes. It mitigates the issue of angle discontinuity that occurs when transitioning between 2π and 0 radians.

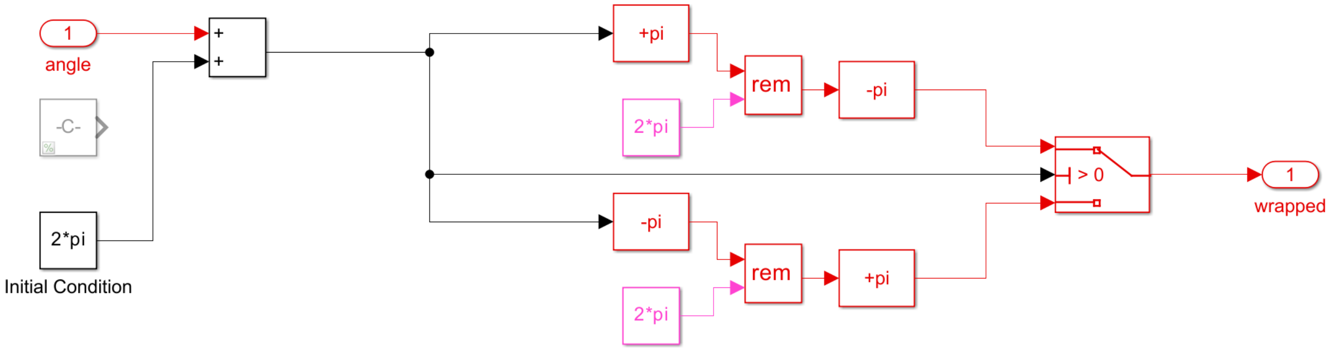


Figure 9: The Angle Wrapper block in Simulink, normalizing the pendulum's angular measurements for consistent control feedback.

Control Strategy: Swing-Up and Stabilization

The control strategy encompasses two distinct phases: swinging up the pendulum to the upright position and then stabilizing it. The 'Swing-up' algorithm utilizes an energy-based control method that injects the necessary energy into the system to bring the pendulum to an upright position. Once the pendulum reaches the vicinity of this unstable equilibrium, the LQR controller takes over to stabilize it. This transition is managed by a State-flow chart, which orchestrates the switch based on the angle and control timing criteria.

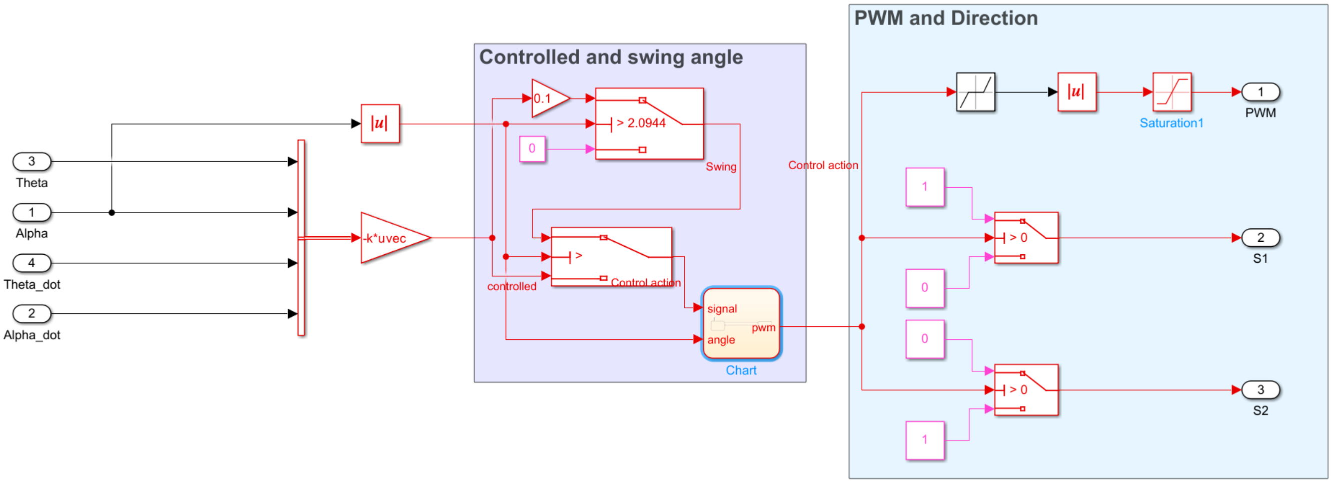


Figure 10: Control strategy implementation in Simulink, featuring the swing-up sequence followed by the LQR stabilization phase.

Closed-Loop System and Sensitivity Analysis

The closed-loop system is constructed by feeding the state vector, which includes the pendulum angles and their derivatives, into the LQR controller. The output, which is the PWM signal for the motor, is processed through a 'PWM and Direction' block that accounts for actuator saturation and directionality of the control input.

A 'Sensitivity Calculator' subsystem is introduced to analyze the responsiveness of the system to the control inputs. This subsystem calculates the sensitivity of the pendulum's angular position and velocity to variations in the PWM signal, providing insight into the robustness and precision of the control algorithm.

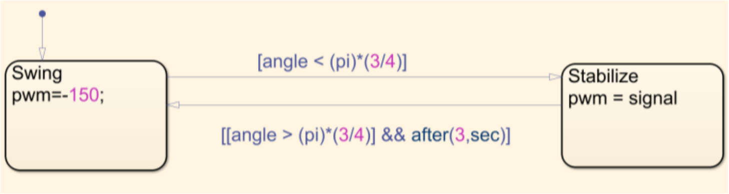


Figure 11: Stateflow diagram depicting the transition logic between the swing-up phase and the stabilization phase of the pendulum.

Simulink Model Overview

The overall Simulink model encapsulates the entire system, including the hardware representation, motor model, angle normalization, control strategy, and sensitivity analysis blocks. This comprehensive model is subjected to a series of tests that simulate various operational scenarios and disturbances.

The controller's performance is evaluated against key metrics such as settling time, overshoot, and steady-state error. Sensitivity analysis further validates the controller's ability to maintain system stability against model uncertainties and external disturbances. The simulation results offer a predictive understanding of how the controller will behave when applied to the physical system.

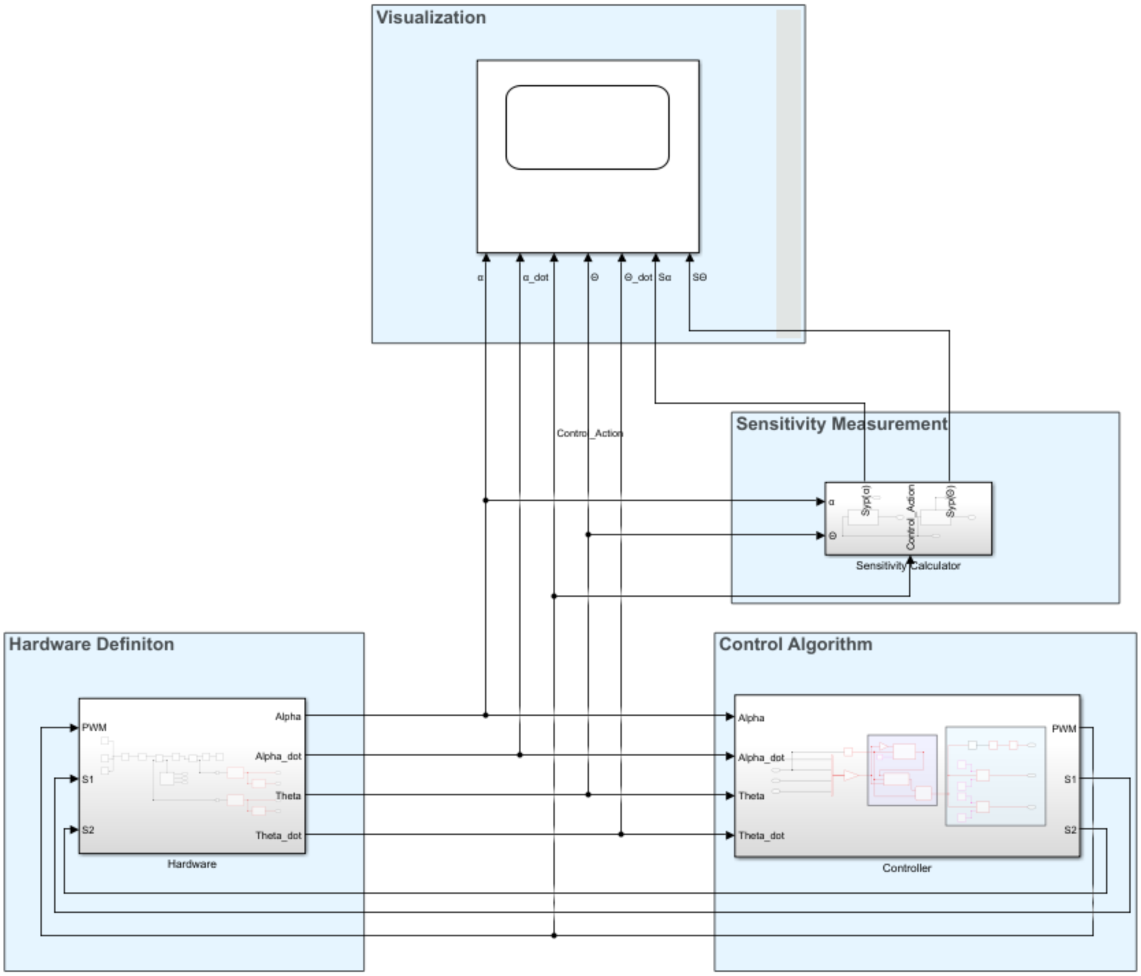


Figure 12: Comprehensive overview of the Simulink model, incorporating the hardware representation, motor dynamics, control strategy, and sensitivity analysis.

Simulations and Results

To ascertain the robustness and reliability of the LQR controller for the rotary inverted pendulum, we conducted a series of simulations under varied conditions:

Scenario 1: Ideal Conditions

Under the first scenario, the pendulum system was allowed to reach its stabilized state without external disturbances, and the motor model was assumed to have an ideal response with no deadzone effect. The results, depicted in fig.9, showcased the system's behavior under optimal conditions. The pendulum angle α and arm angle θ settled to their target states with minimal overshoot and steady-state error, affirming the controller's efficacy in a disturbance-free environment.

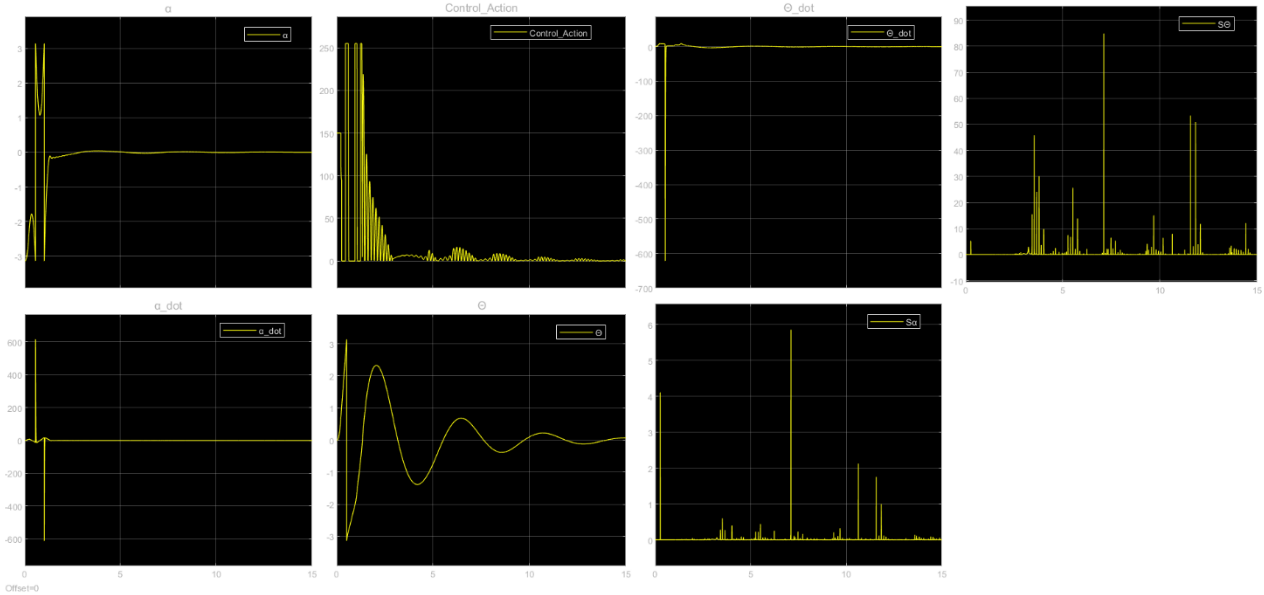


Figure 13: Simulation results under ideal conditions without disturbance and deadzone effects, demonstrating the pendulum's response with LQR control.

Scenario 2: Deadzone Effect with Harsh Disturbance

The second scenario introduced a realistic motor model with a deadzone effect, where PWM signals within the range of (-40,40) do not result in actuation. Additionally, a harsh disturbance was applied to challenge the controller's disturbance rejection capabilities. The outcomes, illustrated in fig.10, revealed the system's resilience in the face of significant perturbations. Despite the deadzone nonlinearity and external force, the LQR controller managed to re-stabilize the pendulum, demonstrating robust performance with increased settling time and transient deviations.

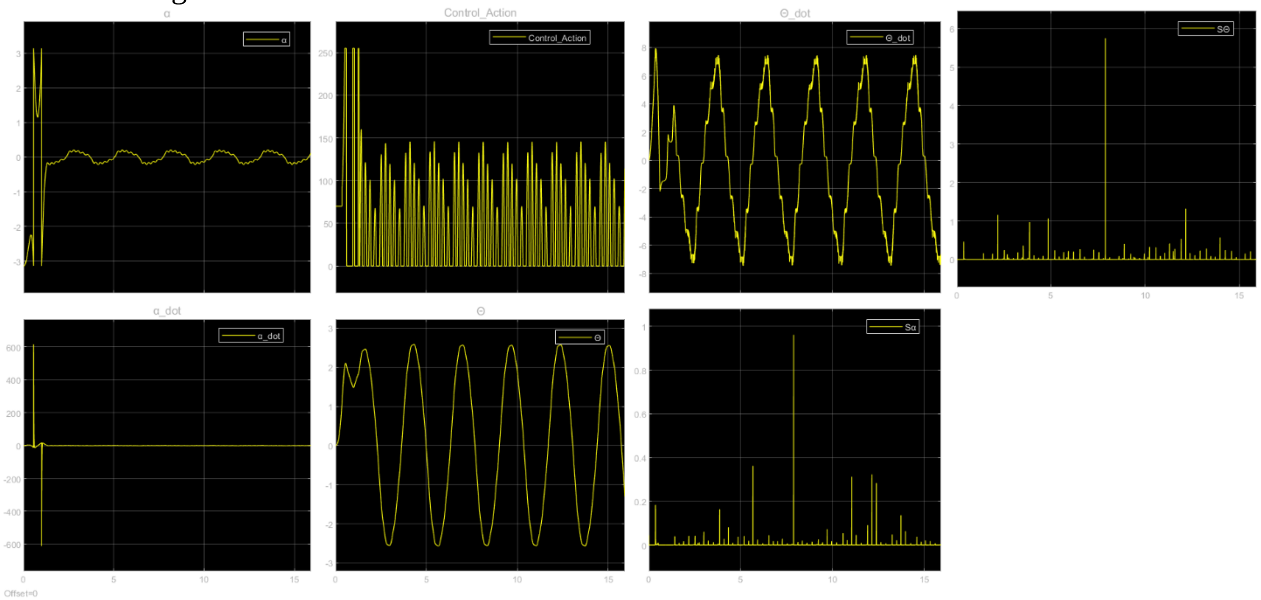


Figure 14: Simulation results with the implementation of motor deadzone effects and a harsh disturbance applied, illustrating the robustness of the LQR controller.

Scenario 3: Deadzone Effect with Slight Disturbance

In the third scenario, we maintained the deadzone effect and introduced a slight disturbance to the system. The results, shown in fig.11, demonstrated the controller's precision in a more typical operational setting. The system experienced minor fluctuations following the disturbance; however, the LQR controller promptly compensated, returning the pendulum to a stable state efficiently. This scenario underscores the controller's adaptability to subtle external influences and actuator imperfections.

The simulation results across the three scenarios provide valuable insights into the system's behavior under varying conditions. Scenario 1 confirmed the theoretical stability of the controller in an ideal setting. Scenario 2 and 3 introduced real-world complexities such as actuator deadzones and external disturbances. The LQR controller's performance in these scenarios suggests a high degree of robustness, with effective disturbance rejection and recovery dynamics. The deadzone effect, while introducing a nonlinearity to the control signal, was well-handled by the controller, indicating its suitability for implementation on physical systems with similar characteristics.

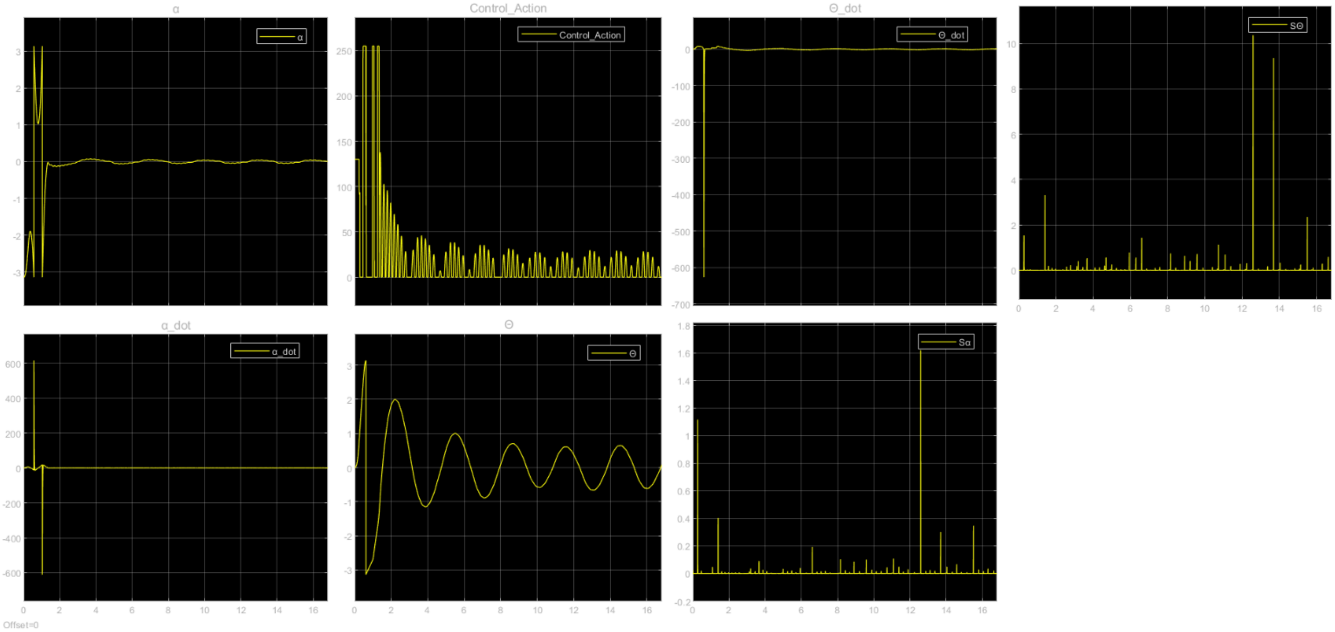
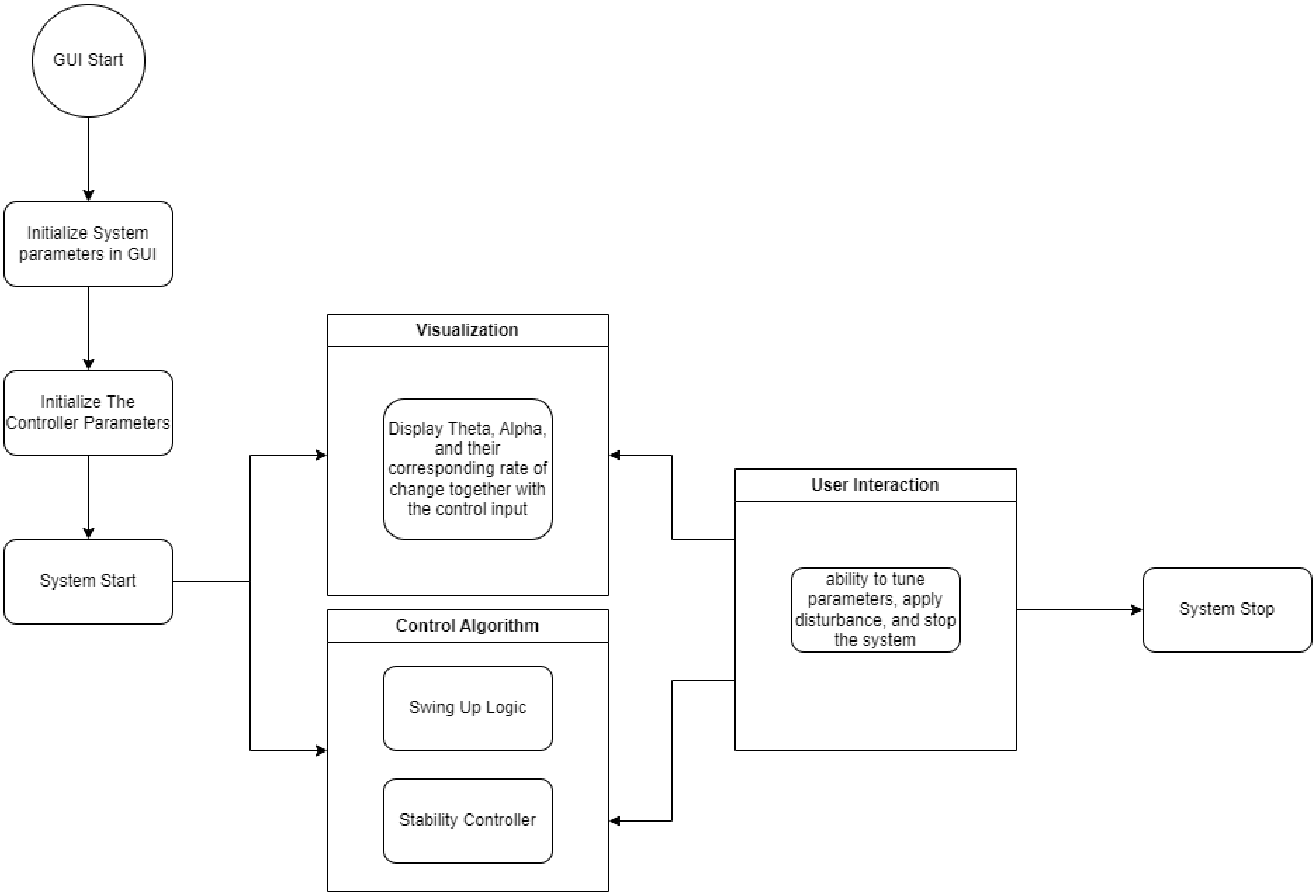


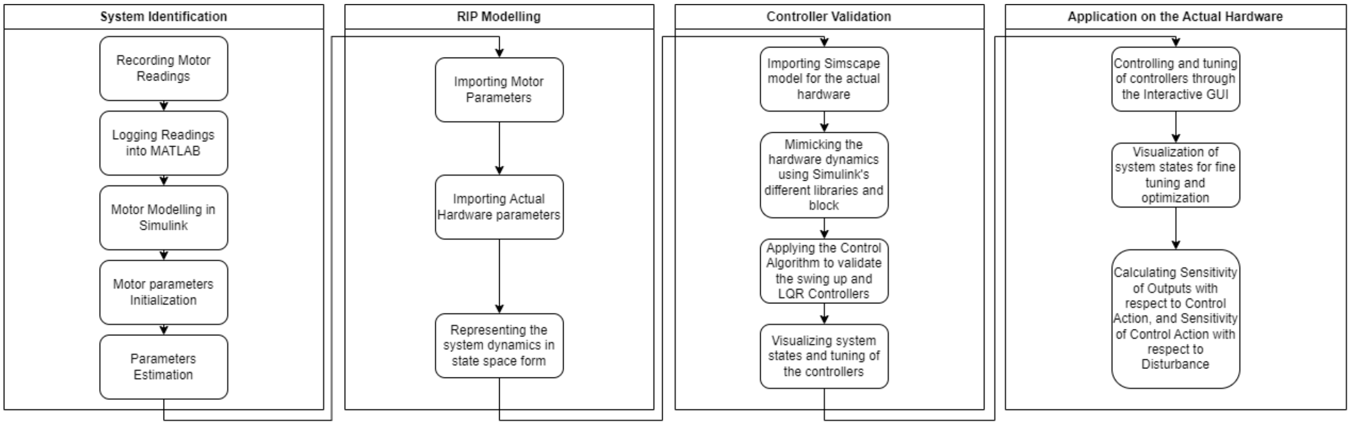
Figure 15: Simulation results reflecting the LQR controller's performance under slight disturbance with deadzone effect, showcasing its adaptability and precision.

FLOWCHARTS

Project Flowchart


GUI Start
Initialize System parameters in GUI
Initialize The Controller Parameters
System Start
Visualization
Display Theta, Alpha, and their corresponding rate of change together with the control input
Control Algorithm
Swing Up Logic
Stability Controller
User Interaction
ability to tune parameters, apply disturbance, and stop the system
System Stop

Progress Flowchart


System Identification
Recording Motor Readings
Logging Readings into MATLAB
Motor Modelling in Simulink
Motor parameters Initialization
Parameters Estimation
RIP Modelling
Importing Motor Parameters
Importing Actual Hardware parameters
Representing the system dynamics in state space form
Controller Validation
Importing Simscape model for the actual hardware
Mimicking the hardware dynamics using Simulink's different libraries and block
Applying the Control Algorithm to validate the swing up and LQR Controllers
Visualizing system states and tuning of the controllers
Application on the Actual Hardware
Controlling and tuning of controllers through the Interactive GUI
Visualization of system states for fine tuning and optimization
Calculating Sensitivity of Outputs with respect to Control Action, and Sensitivity of Control Action with respect to Disturbance

HARDWARE IMPLEMENTATION

CAD Isometric View

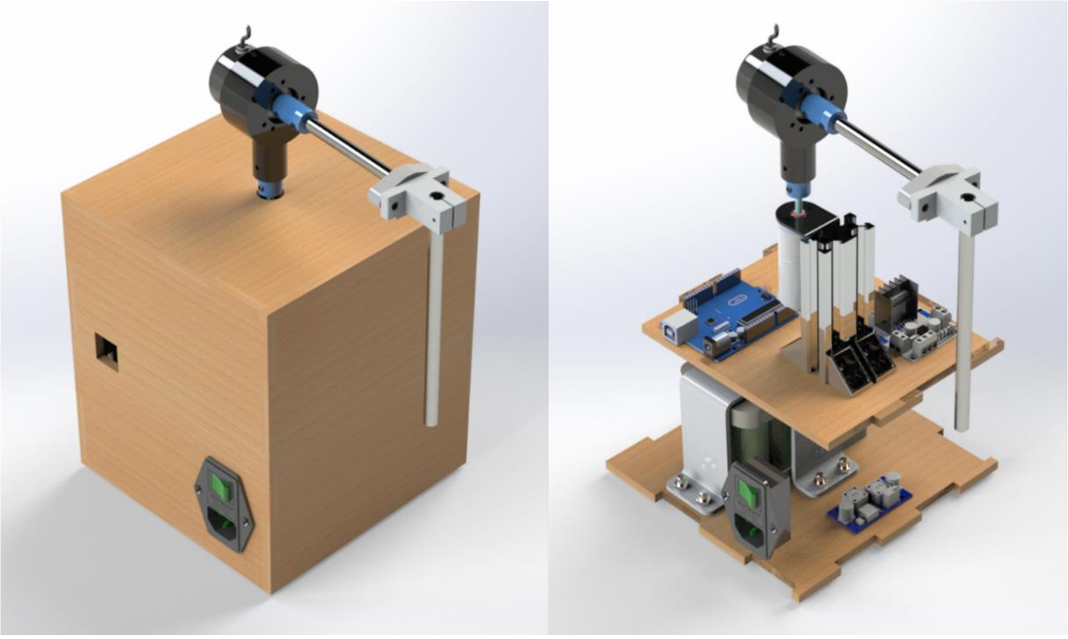


Figure 16: CAD Isometric View

CAD Side View

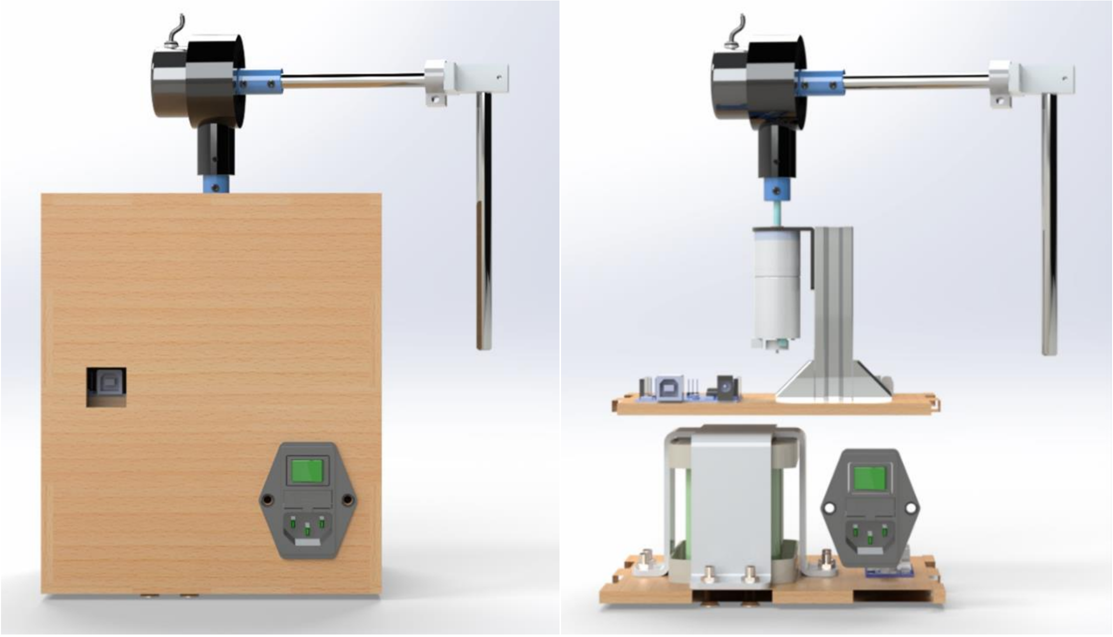


Figure 17: CAD Side View

Real Hardware

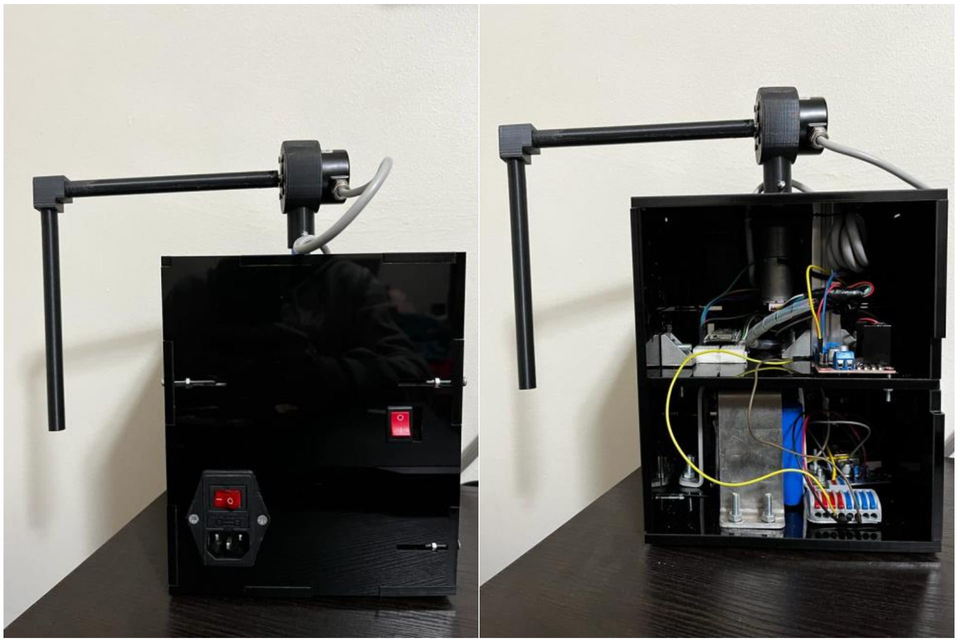


Figure 18: Real Hardware

Electrical Circuit

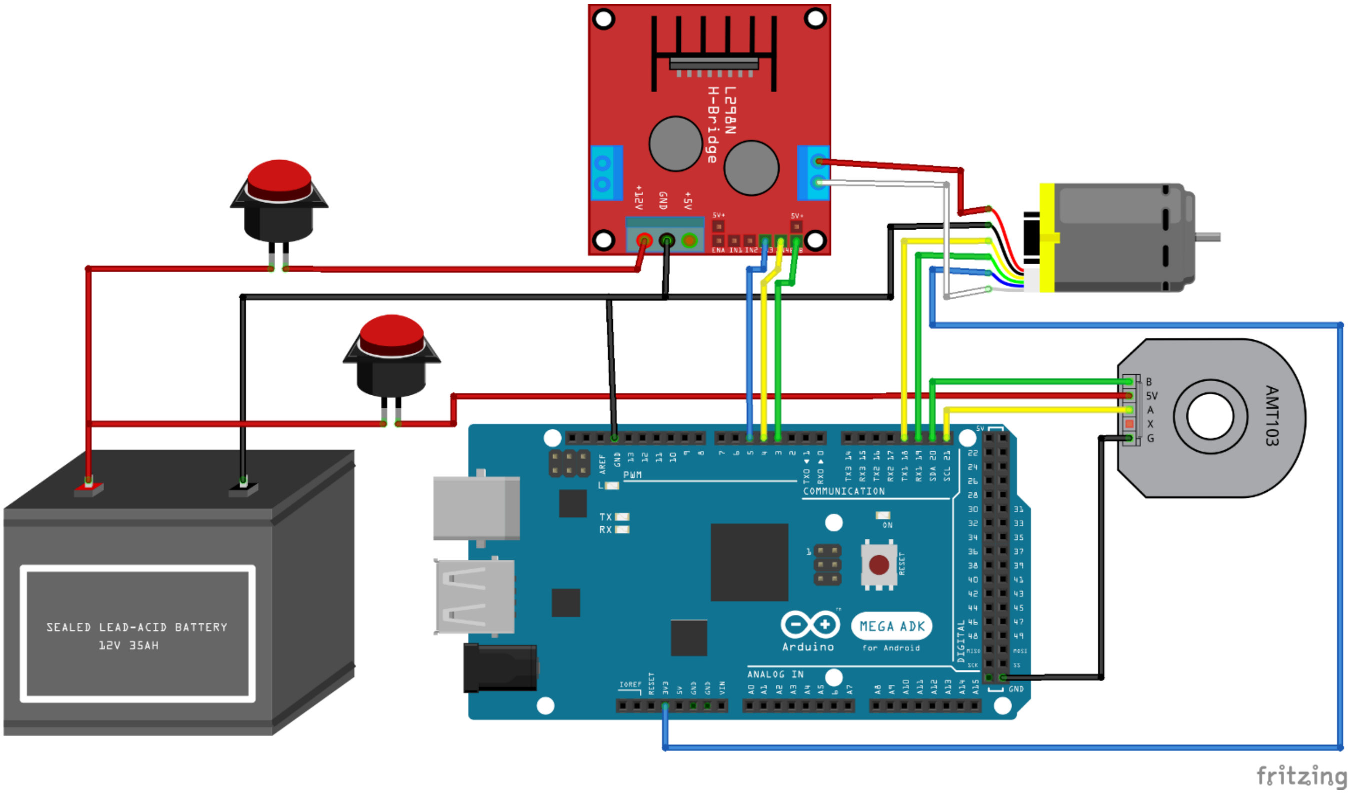
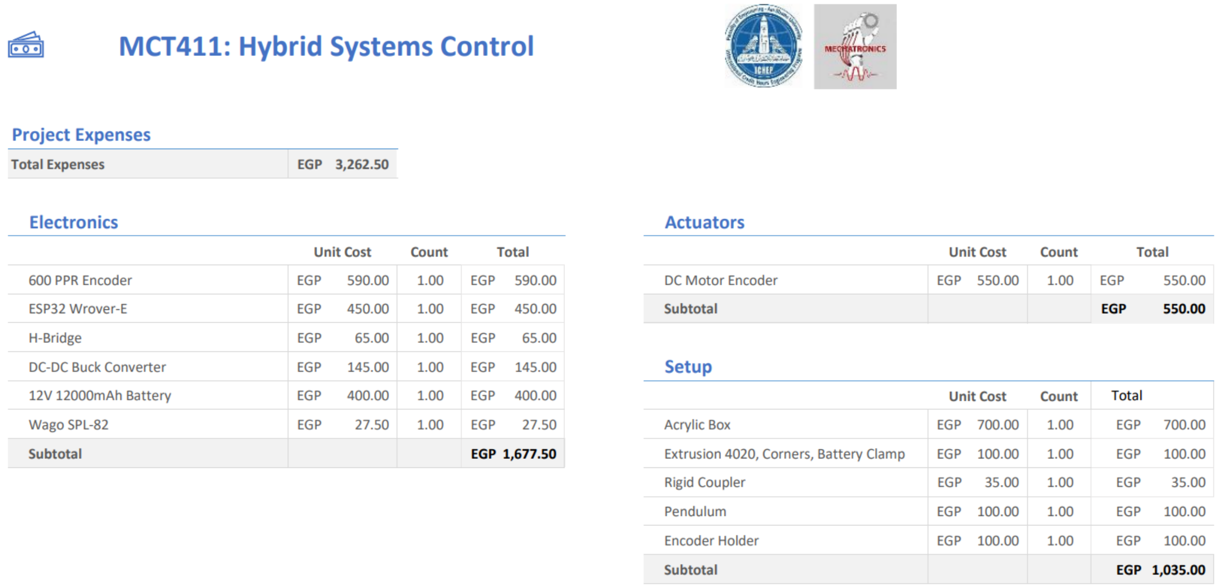


Figure 19: Electrical Circuitry

List of Components



MCT411: Hybrid Systems Control

Project Expenses

Total Expenses	EGP 3,262.50

Electronics

	Unit Cost	Count	Total
600 PPR Encoder	EGP 590.00	1.00	EGP 590.00
ESP32 Wrover-E	EGP 450.00	1.00	EGP 450.00
H-Bridge	EGP 65.00	1.00	EGP 65.00
DC-DC Buck Converter	EGP 145.00	1.00	EGP 145.00
12V 12000mAh Battery	EGP 400.00	1.00	EGP 400.00
Wago SPL-82	EGP 27.50	1.00	EGP 27.50
Subtotal			EGP 1,677.50

Actuators

	Unit Cost	Count	Total
DC Motor Encoder	EGP 550.00	1.00	EGP 550.00
Subtotal			EGP 550.00

Setup

	Unit Cost	Count	Total
Acrylic Box	EGP 700.00	1.00	EGP 700.00
Extrusion 4020, Corners, Battery Clamp	EGP 100.00	1.00	EGP 100.00
Rigid Coupler	EGP 35.00	1.00	EGP 35.00
Pendulum	EGP 100.00	1.00	EGP 100.00
Encoder Holder	EGP 100.00	1.00	EGP 100.00
Subtotal			EGP 1,035.00

Figure 20: BOM

Files and Results

REFERENCES

1. Dorf, R.C. and R.H. Bishop, 1998. Modern Control Systems. 8th Ed, Addison-Wesley, 2725 Sand Hill Rd, Menlo Park, CA., ISBN: 0201-30864-9, pp: 855.
2. Astrom, K.J. and K. Furuta, 1996. Swinging up a Pendulum by Energy Control. Paper presented at IFAC 13th World Congress, San Francisco, Californiapp:1-6. http://www.control.utoronto.ca/~broucke/ece1653s/Intro/ast_fur96.pdf
3. Zhong, W. and Rock, 2001. Energy and passivity-based control of the double inverted pendulum on cart. Proceeding of the IEEE Conference on Control Applications, Sep. 5-7, Mexico, pp: 896-901. http://www.tf.uni-kiel.de/etit/ART/paper/2001/ieee_cca_isic_zhong.pdf
4. Astrom, K.J. and K. Furuta, 2000. Swinging up a pendulum by energy control. Automatica, 36: 287-295. http://www.ece.ucsb.edu/~hespanha/ece229/references/AstromFurutaAUTOM00.pdf
5. Wiki, Furuta Pendulum. https://en.wikipedia.org/wiki/Furuta_pendulum
6. Ratiroch, A., Anabuki, P., Hirata, H.: Self-tuning control for rotational inverted pendulum by eigenvalue approach. Proceedings of TENCON, D, pp. 542–545. Chiang Mai, Thailand (2004)