ME155C Final Project Report
Inverted Pendulum Control: Swing-up, Balance, and Catching

Raaghav Thirumaligai, Cole Giusto, Tien Nguyen

June 8, 2025

Abstract

This project focuses on the identification, modeling, and control of an inverted pendulum on a cart system. Through experimental identification, control design, and closed-loop testing, we implemented robust swing-up, balancing, and catching controllers. System identification methods included logarithmic sine sweeps and parametric identification with MATLAB's tfest. Controller design employed energy-based swing-up methods and robust LQR/LQG balancing strategies. Experimental validations confirm effective performance, achieving stable equilibrium transitions and pendulum catching.

1 Introduction

This project investigates the control of an inverted pendulum on a cart, specifically targeting swing-up, balancing, and catching tasks. The inverted pendulum, a classic control problem, requires precise and robust feedback control strategies due to its inherent instability. Previous literature includes methods involving energy-based swing-up and state-space feedback stabilization. The report is structured with system identification in Section 2, controller design in Section 3, closed-loop results in Section 4, and conclusions in Section 5.

2 System Identification

2.1 Process Description

The experimental setup consists of an asymmetric metal rod mounted to a motor-driven cart, with encoders providing measurements of pendulum angle (θ) and cart position (x). The control input is the motor voltage (u), and the primary controlled outputs are cart position and pendulum angle.

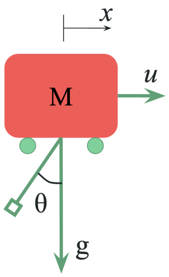

Figure 1: Experimental setup: Cart-pendulum system.

2.2 Identification Methods

Non-parametric identification utilized a logarithmic sine sweep (0.3 – 3 Hz). Parametric identification was done using MATLAB's `tfest` function, performing 30 experimental trials.

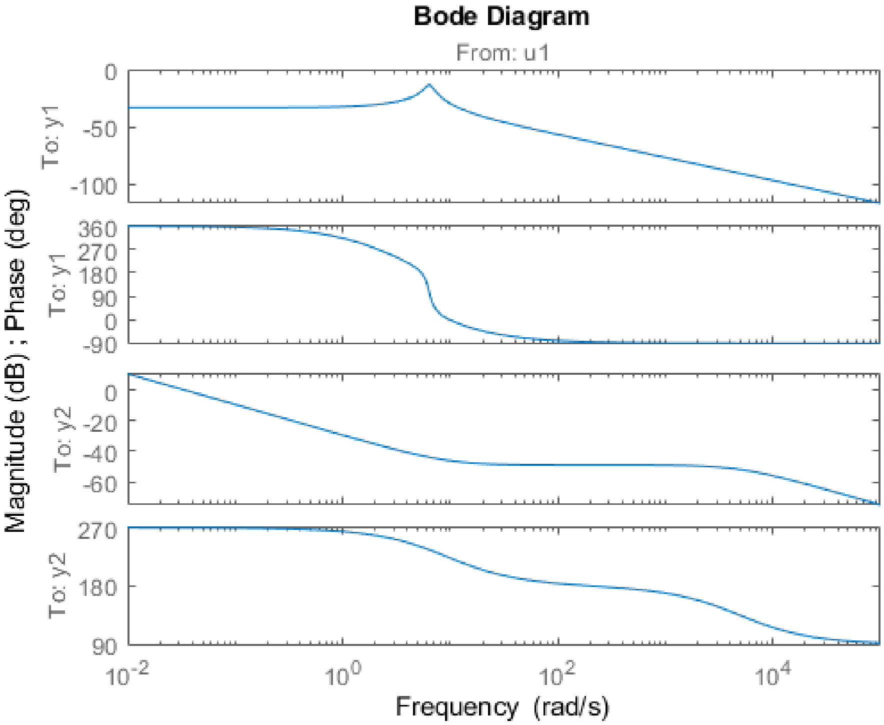

Figure 2: Bode plot: Identified frequency response for cart dynamics and pendulum dynamics downwards.

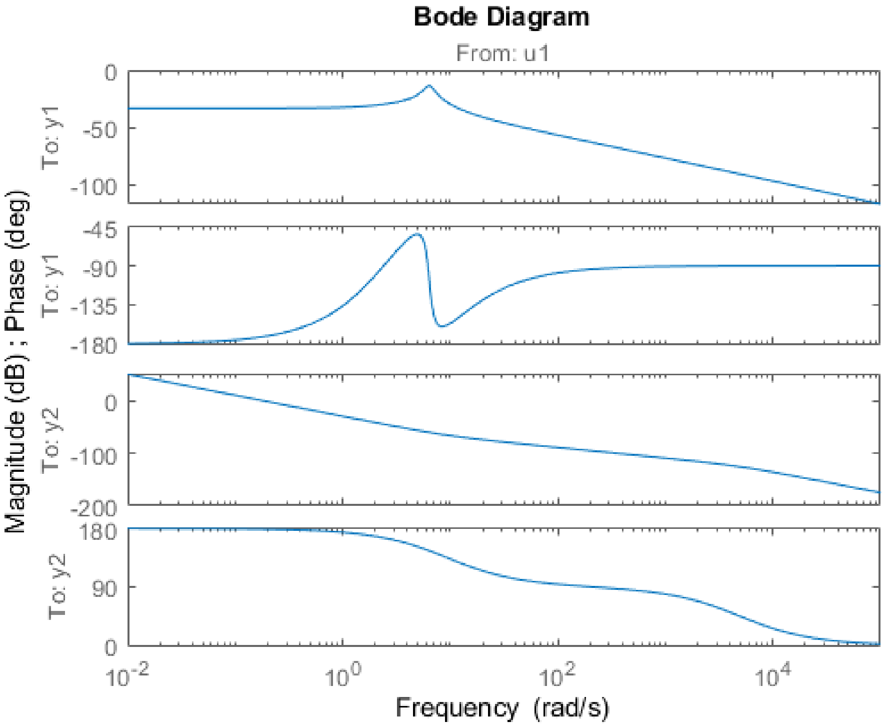

Figure 3: Bode plot: Identified frequency response for cart dynamics and pendulum dynamics upwards.

3 Controller Design

3.1 Swing-Up Controller

The swing-up controller uses an energy addition method to drive the pendulum toward the upright position. Around every zero crossing of the pendulum angle θ, the controller applies a force to move the cart in the direction opposite to the pendulum's velocity in order to increase the energy in the pendulum. This continues until the pendulum is able to fully "swing up". Additionally, a simple proportional controller is used to regulate the cart's position to stay near zero meters, preventing excessive cart movement during the swing-up phase. The activation is limited to a range $\theta \in [-30°, 30°]$ to try and reach the inverted state with a small velocity.

Figure 4: Swing-up controller activation.

3.2 Balancing Controller

The balancing controller employs LQR/LQG methods to stabilize the pendulum in the inverted (unstable) equilibrium within a narrow angular region.

3.3 Catching Controller

Similar to balancing, catching employs LQR/LQG but aims to return the pendulum reliably to the stable downward equilibrium.

3.4 State Machine

Control modes transition between swing-up, balancing, and catching using a robust state machine architecture. This state machine was implemented in Simulink using logic gates, comparing the pendulum angle and velocity to determine the current state and enable the correct controller.

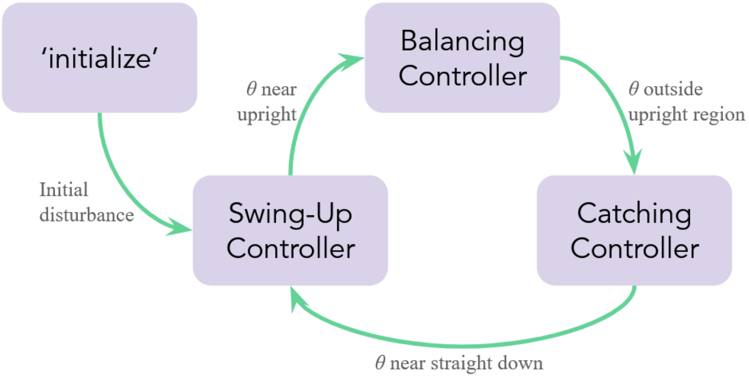

Figure 5: State machine for mode switching between controllers.

4 Closed-Loop Performance

Closed-loop experiments validated the controllers' performance. The step response showed quick stabilization and effective handling of mode transitions.

um toward the u

Figure 6: Closed-loop step response for cart and pendulum positions.

5 Conclusions and Future Work

The controllers successfully addressed swing-up, balancing, and catching objectives, validating theoretical designs with robust experimental performance. Further work includes enhancing failure logic and addressing steady-state error.

References