Intelligent systems for control
TI0090

Modelling of a dynamic system

Student name:David Machado Couto Bezerra

Contents

1 Introduction **3**

2 System Description **3**
- 2.1 Lagrangian Mechanics . . . 4
- 2.2 Euler-Lagrange Equations . . . 4
- 2.3 State-Space Model . . . 5
- 2.4 Model Simulations . . . 5

3 System Linearization **6**
- 3.1 Linearization Process . . . 7
- 3.2 State-Space Model . . . 7
- 3.3 Simulation of the Linear Model . . . 8

4 System Analysis **9**
- 4.1 BIBO Stability . . . 10
- 4.2 Lyapunov's Stability . . . 10
- 4.3 Controllability and Observability . . . 11
  - 4.3.1 Controllability . . . 12
  - 4.3.2 Observability . . . 12

5 Theoretical Background **13**
- 5.1 PID Controller . . . 13
  - 5.1.1 Tuning Methods . . . 13
- 5.2 Cascade Controller . . . 15
- 5.3 Fuzzy Controller . . . 16
- 5.4 Linear Quadratic Regulator (LQR) . . . 16
- 5.5 Model Predictive Control (MPC) . . . 17

6 Control System Design **17**
- 6.1 System Specifics and Structure . . . 18

6.2 Tuning the System . . . 18
6.3 PD Controller Response . . . 18
6.4 Cascade PD Control . . . 19
6.5 Fuzzy PD Controller . . . 20
6.6 Linear Quadratic Regulator (LQR) . . . 21
6.7 Model Predictive Control (MPC) . . . 22

7 Controlled System Comparison **23**
7.1 PD Controller Performance . . . 23
7.2 Cascade PD Controller Performance . . . 25
7.3 Fuzzy PD Controller Performance . . . 25
7.4 LQR Performance . . . 25
7.5 MPC Performance . . . 31

8 Conclusion **31**

References **31**

1 Introduction

The inverted pendulum is a widely recognized example in control theory, known for illustrating the challenges of stabilizing a nonlinear system. Due to its inherent instability, it requires a dynamic control strategy to keep the pendulum upright, typically by adjusting the movement of the base cart. As such, it has become a foundational model for testing control strategies and is often used as an educational tool to teach key concepts like feedback control and state-space representation.

Historically, the problem of stabilizing the inverted pendulum has been studied extensively. The first documented solution was presented by Roberge in his MIT thesis in the 1960s [6], marking the beginning of numerous studies and innovations aimed at developing robust control strategies. Over the decades, a variety of methods have been explored, including proportional-integral-derivative (PID) controllers, linear quadratic regulators (LQR), and model predictive control (MPC). As one of the simplest yet most fundamental robotic systems to control, it has played a crucial role in advancing control theory [1].

Beyond academic research, the inverted pendulum has found practical applications in several fields. It is used in robotics to stabilize bipedal and autonomous systems, in transportation for improving vehicle dynamics, and in structural engineering to develop earthquake-resistant designs. Moreover, it has influenced the development of medical devices designed to assist with balance and support for patients with motor difficulties [2]. These varied applications demonstrate the importance of the inverted pendulum as an inspiration for control strategies in both theoretical and practical domains.

2 System Description

The inverted pendulum system consists of a pendulum mounted on a cart, where the cart's movement controls the pendulum's stability. The system is modeled using Lagrangian mechanics, an image for the system it's displayed on Figure 1.

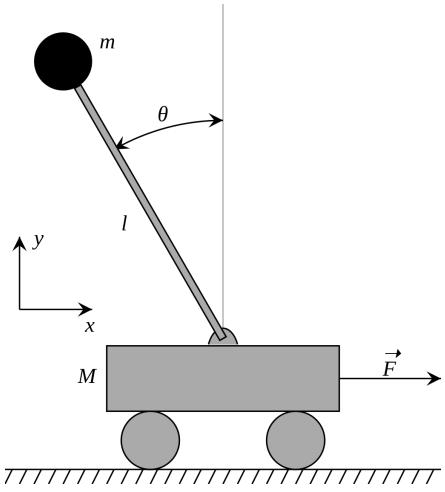

Figure 1: The free-body dynamics of an inverted pendulum

2.1 Lagrangian Mechanics

The main :

- **Potential Energy (U):**

$$U = mgl\cos\theta$$

  where m is the mass of the pendulum, g is the acceleration due to gravity, l is the length of the pendulum, and θ is the angle of deviation.

- **Kinetic Energy (T):**

$$T = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{\theta}\dot{x}\cos\theta$$

  where M is the mass of the cart and $\dot{x}$ is the velocity of the cart.

- **Lagrangian (L):**

$$L = T - U = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{\theta}\dot{x}\cos\theta - mgl\cos\theta$$

In this model:

- $f(t)$ force applied to the cart;
- g is the acceleration due to gravity;
- m is the mass of the pendulum;
- M is the mass of the cart.

2.2 Euler-Lagrange Equations

The solutions for Euler-Lagrange equations, derived from the Lagrangian, define the system's dynamics:

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = 0$$

with q being either θ or x.

- Equation for $\ddot{x}$ (cart acceleration):

$$\ddot{x} = \frac{u - ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta}{M+m}$$

- Equation for $\ddot{\theta}$ (angular acceleration):

$$\ddot{\theta} = \frac{g\sin\theta - u\cos\theta - \frac{d_{mf}}{m\cdot L}\dot{\theta}}{l}$$

The term $\frac{d_{mf}}{m\cdot L}$ for damping was added so that the pendulum would have the tendency to stop after an period of time which is natural and also $u = f(t)$ being the input for the system in newtons.

2.3 State-Space Model

The state-space model for the inverted pendulum system is defined by the following state variables:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

where:

- $x_0 = \theta$ is the pendulum angle,
- $x_1 = \dot{\theta}$ is the pendulum angular velocity,
- $x_2 = x$ is the cart position, and
- $x_3 = \dot{x}$ is the cart velocity.

The state-space model is governed by the following system of differential equations:

$$\dot{x_0} = x_1 \tag{1}$$

$$\dot{x_1} = \frac{g \sin x_0 - u \cos x_0 - \frac{d_{mf}}{m \cdot l} x_1}{l} \tag{2}$$

$$\dot{x_2} = x_3 \tag{3}$$

$$\dot{x_3} = \frac{u - ml\dot{x_1} \cos x_0 + ml x_1^2 \sin x_0}{M + m} \tag{4}$$

With these states equations defined it can be use simulations to achieved the response of the non-linear model.

2.4 Model Simulations

The simulations presented here were conducted using Python, following the structure of the function code provided in Listing 1.

In this case, no control input is applied, so the cart remains stationary ($u = 0$). The purpose of this simulation is to observe the natural behavior of the pendulum and analyze why an external control system is necessary for stabilization. By allowing the pendulum to move freely without any external intervention, we can clearly see the system's inherent instability and the need for control mechanisms to maintain an upright position as shown by the Figure 2.

In the plot, the pendulum begins with an initial state of $x = [\pi/4, 0, 0, 0]$. As the simulation progresses, the system tends to move toward the state $x = [\pi, 0, 0, 0]$, where the angle π indicates that the pendulum has fully rotated downward, reaching the inverted position.

Listing 1: Model simulation function template

```
def normal_pd(state, t):
    ds = np.zeros_like(state)
    _th, _th_d, _x, _x_d = state

    damping_term = V * _th_d / (m * L)

    # No control applied, the cart remains stationary
    u = 0

    ds[0] = _th_d
    ds[1] = (G * np.sin(_th) - u * np.cos(_th) - damping_term) / L
    ds[2] = 0
    ds[3] = 0
    return ds
```

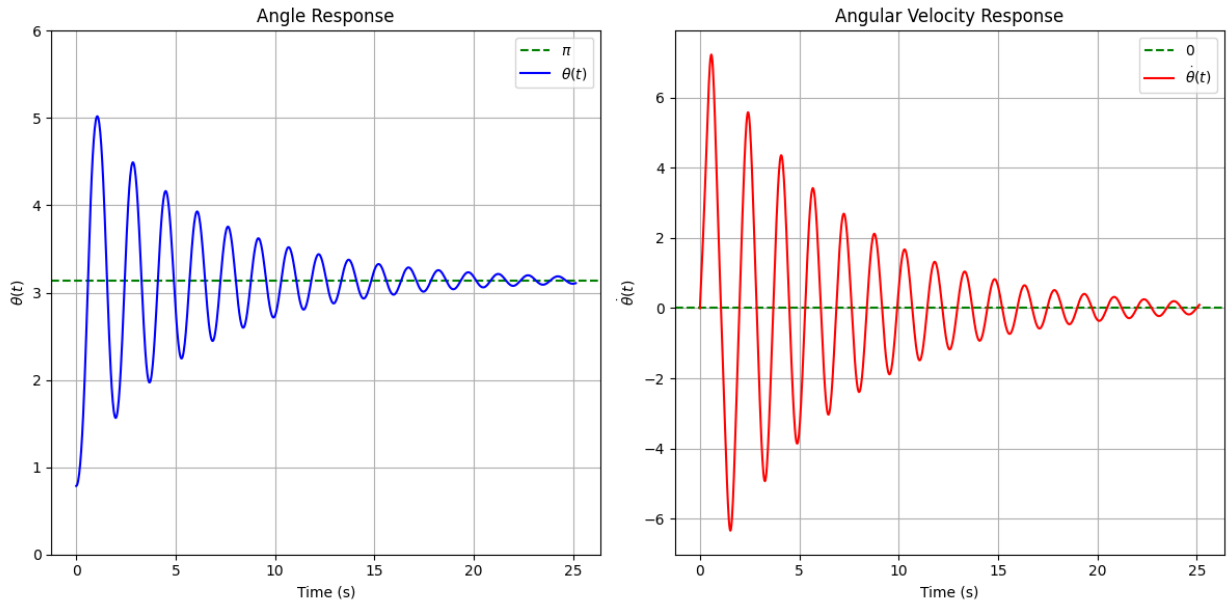

Figure 2: Nonlinear model response

3 System Linearization

The process of linearization is essential for simplifying the analysis and control design of nonlinear systems. By linearizing the inverted pendulum system around a convenient steady-state operating point, we can approximate the system's behavior using linear techniques, making it more manageable for control design and analysis.

3.1 Linearization Process

To linearize the system, we consider small deviations around the equilibrium point. For the inverted pendulum system, the steady-state corresponds to:

- Control input: $u = 0$,
- Pendulum angle: $\theta = 0$,
- Steady-state condition: $[\theta, \dot{\theta}] = [0, 0]$.

The linearization process requires approximating the nonlinear functions $\sin(\theta)$ and $\cos(\theta)$ using their respective Taylor series expansions around the equilibrium point. First, the Taylor series expansion for $\sin(\theta)$ around $\theta = 0$ is given by:

$$\sin(\theta) = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \cdots$$

For small values of θ (i.e., near the equilibrium point), the higher-order terms become negligible, allowing us to approximate $\sin(\theta) \approx \theta$.

Similarly, the Taylor series expansion for $\cos(\theta)$ around $\theta = 0$ is expressed as:

$$\cos(\theta) = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \cdots$$

Again, for small angles, the higher-order terms are insignificant, resulting in the approximation $\cos(\theta) \approx 1$.

These approximations for $\sin(\theta)$ and $\cos(\theta)$ enable us to linearize the system dynamics around the equilibrium point.

$$\dot{x_0} = x_1 \tag{5}$$

$$\dot{x_1} = \frac{g x_0 - u - \frac{d_{mf}}{m \cdot l} x_1}{l} \tag{6}$$

$$\dot{x_2} = x_3 \tag{7}$$

$$\dot{x_3} = \frac{u}{M + m} \tag{8}$$

3.2 State-Space Model

The linearized state-space representation of the system is derived from the linearized dynamics. The state-space model is given by:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g}{l} & -\frac{d_{mf}}{m \cdot l} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ -\frac{1}{l} \\ 0 \\ \frac{1}{M+m} \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

This model captures the linearized dynamics of the pendulum for small deviations around the equilibrium point. The state vector $x = [\theta, \dot{\theta}, x, \dot{x}]$ consists of the pendulum angle and angular velocity, and u represents the control input applied to the cart.

3.3 Simulation of the Linear Model

To evaluate the accuracy of the linearized model, we compare its behavior with that of the original nonlinear model. The response of the system to a step input of amplitude A is plotted, and the Root Mean Square Error (RMSE) is used as a metric for comparison.

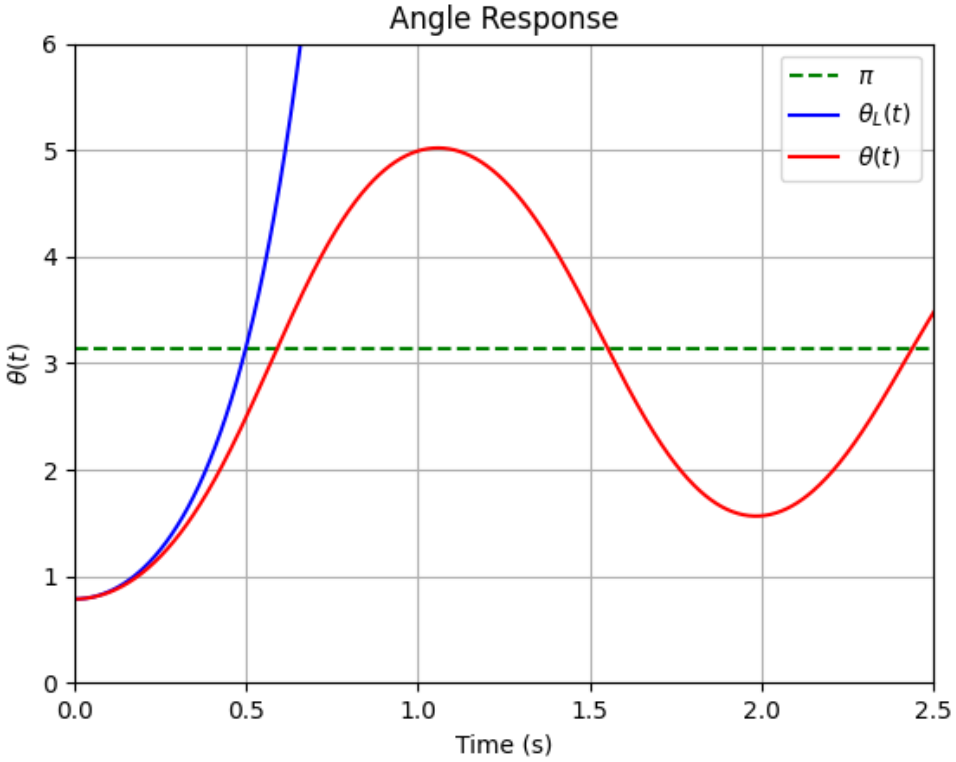

Figure 3: Step response of the linearized system

The RMSE values over different time intervals are as follows:

- **RMSE** for $t \in (0, \frac{\pi}{4})$: 1.74316.
- **RMSE** for $t \in (0, \pi)$: 31583.68.

These results demonstrate significant differences between the linearized and nonlinear models, particularly over larger angular displacements. This discrepancy highlights the limitations of the linear approximation, which is only valid for small deviations around the equilibrium point. Since the model has been linearized, we can now derive the system's transfer function:

$$G(s) = \begin{bmatrix} \frac{1}{(M+m)\,s^2} \\ \frac{-\frac{1}{l}}{s^2 + \frac{d_m f}{ml}\,s - \frac{g}{l}} \end{bmatrix}$$

The first transfer function describes the cart's position response to the applied force, while the second represents the pendulum's angle response to the same input. Both functions provide a linearized view of the system's behavior. Using these transfer functions, the step response plot reveals key performance metrics, including steady-state value, overshoot, rise time, and settling time, enabling a straightforward comparison between the linear and nonlinear models. .

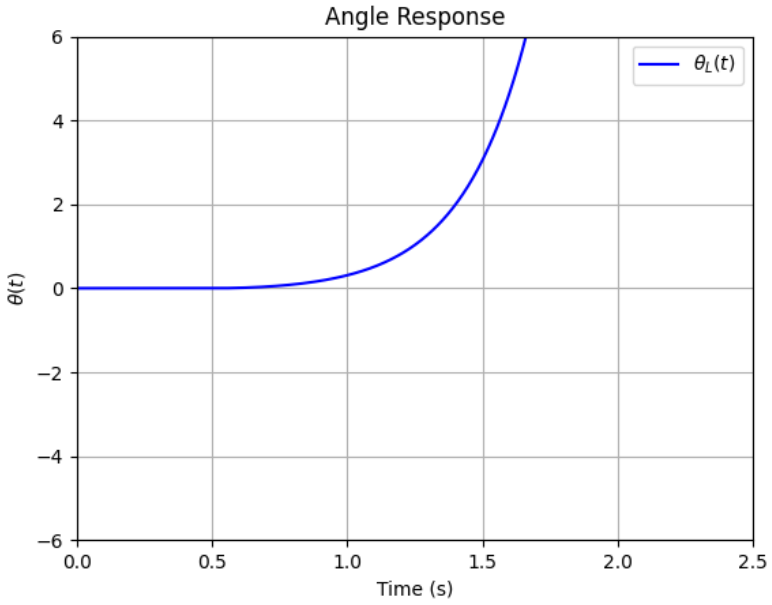

Figure 4: Step response of the pendulum

Since the step response was exponential no information can be derived in terms of rise time, overshoot, settling time, undershoot, peak, peak time and steady state value.

4 System Analysis

In this section, we provide the theoretical background for the main properties of the dynamical models and justify the importance of controlling the system. The analysis covers BIBO stability and Lyapunov's stability, which are critical in determining the behavior of the system under different

conditions. On Figure 5, there is the map of poles and it can be seen the presence of positive real part poles where it characterizes an positive exponential in the time domain and therefore the system is not stable.

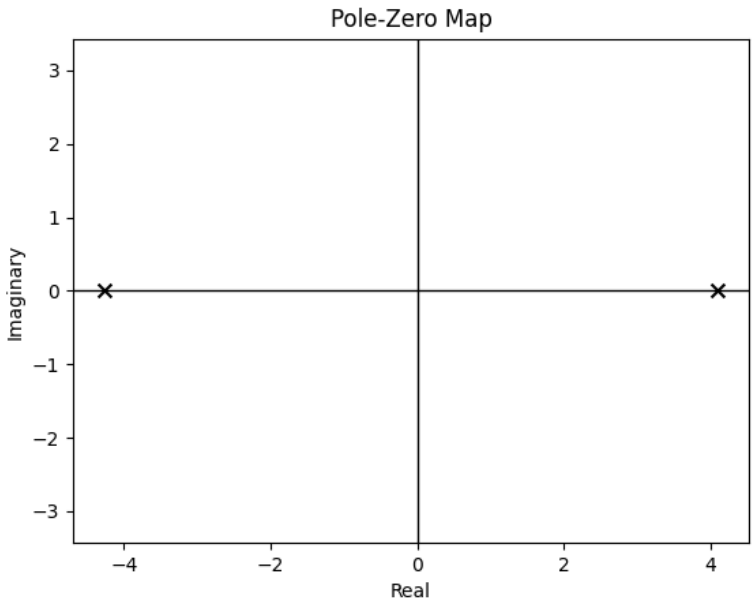

Figure 5: Poles map of the system

4.1 BIBO Stability

The system demonstrated instability under the BIBO (Bounded Input, Bounded Output) criterion. An exponential response to a step input indicates that the system fails to meet the BIBO stability condition as showed on the previous Figure 4, which requires that bounded inputs produce bounded outputs.

4.2 Lyapunov's Stability

Lyapunovâs stability theory is a powerful tool for determining the stability of a system based on energy functions. According to Lyapunovâs direct method:

[Lyapunov's Direct Method] Let $\dot{x} = f(x)$ for a continuous function f and a domain D around the origin. If there exists a continuously differentiable function $V(x)$ such that:

$$V(x) > 0 \quad \text{for all } x \in D \setminus \{0\}, \quad \text{and } V(0) = 0,$$
$$\dot{V}(x) = \nabla V \cdot f(x) \leq 0 \quad \text{for all } x \in D \setminus \{0\}, \quad \text{and } \dot{V}(0) = 0,$$

then the origin is stable in the sense of Lyapunov. If $\dot{V}(x) < 0$, the origin is asymptotically stable. Furthermore, if $\dot{V}(x) \leq -\alpha V(x)$ for some $\alpha > 0$, the origin is exponentially stable.

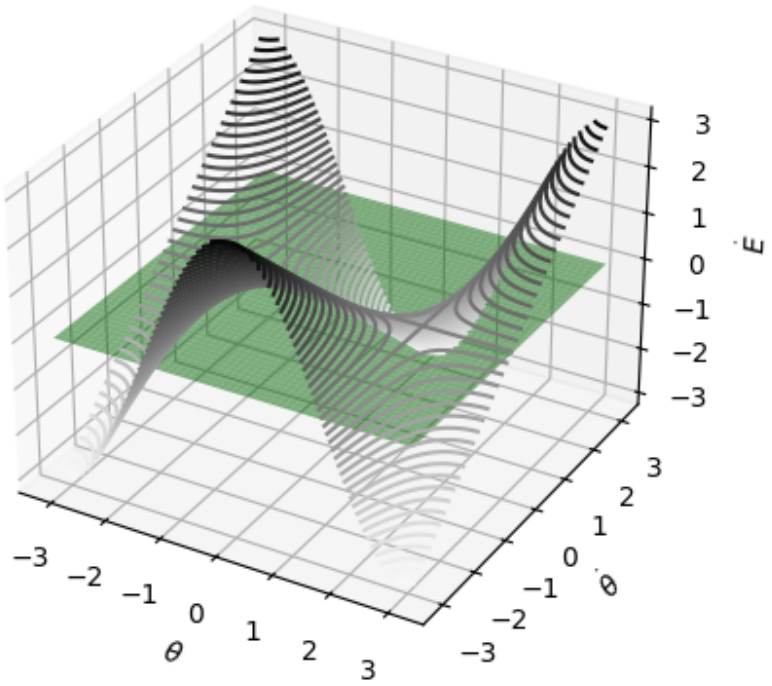

Figure 6: Lyapunov's function derivative

In the case of the inverted pendulum, Lyapunov's method is often applied using an energy-based function. A commonly used function for this system is the total mechanical energy of the pendulum:

$$E(\theta, \dot{\theta}) = \frac{1}{2} m l^2 \dot{\theta}^2 + mgl \cos\theta$$

This energy function represents the kinetic and potential energy of the pendulum. If the total energy decreases over time, the system is stable in the sense of Lyapunov. In the context of the inverted pendulum, the plot of the previous function shows that the system is not stable around $[0, 0]$ since there is values of the energy derivative bigger than zero (Figure 6) and so the tendency of the system is to go there.

4.3 Controllability and Observability

In this section, we define and compute the controllability and observability of the inverted pendulum system. These concepts are crucial in determining whether it is possible to fully control the system

using inputs and whether the complete state can be reconstructed from the output.

4.3.1 Controllability

A system is controllable if it is possible to move from any initial state to any desired final state within a finite time interval by applying appropriate input signals. The controllability matrix $\mathcal{C}$ is given by:

$$\mathcal{C} = [B\ AB\ A^2B\ \dots\ A^{n-1}B]$$

If the rank of $\mathcal{C}$ equals the number of states n, then the system is fully controllable:

$$\text{rank}(\mathcal{C}) = n$$

4.3.2 Observability

A system is observable if it is possible to deduce the complete state of the system from its output signals. The observability matrix $\mathcal{O}$ is given by:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

If the rank of $\mathcal{O}$ equals the number of states n, then the system is fully observable:

$$\text{rank}(\mathcal{O}) = n$$

In the case of the inverted pendulum model, both controllability and observability criteria are satisfied, with the ranks of $\mathcal{C}$ and $\mathcal{O}$ matching the number of states, indicating that the system is both controllable and observable, the code used to prove this statement is displayed on Listing 2.

Listing 2: Controllability and Observability Calculation

```
from numpy.linalg import matrix_rank
import control as ct

# Controllability
controllability_matrix = ct.ctrb(A, B)
is_controllable = matrix_rank(controllability_matrix) == len(A)
print(f"Controllability: {'YES' if is_controllable else 'NO'} (Rank: {matrix_rank(
    controllability_matrix)})")

# Observability
observability_matrix = ct.obsv(A, C)
is_observable = matrix_rank(observability_matrix) == len(A)
print(f"Observability: {'YES' if is_observable else 'NO'} (Rank: {matrix_rank(
    observability_matrix)})")
```

5 Theoretical Background

This section provides an overview of classical control strategies, focusing on the PID controller, cascade controller, and fuzzy controller. Each control method is discussed in terms of its control law, as well as the advantages and limitations it brings to general control systems. The decision to employ these strategies depends on the system's dynamics and specific requirements such as disturbance rejection, ease of implementation, and adaptability to nonlinearity.

5.1 PID Controller

The Proportional-Integral-Derivative (PID) controller is one of the most widely used control methods in industry. It combines proportional, integral, and derivative actions to correct system errors, offering a comprehensive approach to control by considering the present, past, and future errors in the system [4]. The control law is defined as:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{d}{dt} e(t)$$

In this equation, K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. The proportional term addresses the current error magnitude, the integral term accumulates the past error to correct steady-state offsets, and the derivative term reacts to the rate of change of the error, damping oscillations and preventing instability.

Advantages: The PID controller provides robust and well-rounded control, compensating for steady-state errors through the integral action and reacting quickly to disturbances with the derivative action. It is effective in a wide range of systems, from simple processes to more complex applications requiring fine control.

Disadvantages: The integral term can cause overshoot and make the system unstable if not properly tuned, especially in systems sensitive to accumulated errors, like highly dynamic or nonlinear systems. Additionally, PID controllers can be overly sensitive to noise, as the derivative term amplifies high-frequency fluctuations.

5.1.1 Tuning Methods

Tuning a PID controller is critical for achieving the desired system response. Several methods are available for tuning PID controllers, each with its own advantages and applications. Some of the most common tuning methods include the Ziegler-Nichols method and the Internal Model Control (IMC) method.

Ziegler-Nichols Tuning Method: The Ziegler-Nichols tuning method, first introduced by J.G. Ziegler and N.B. Nichols in 1942, is one of the earliest and most widely used methods for tuning PID controllers. It assumes the system is of first order and aims to provide a simple approach to finding the appropriate PID parameters. The method involves the following steps:

1. Set K_D and K_I to zero;
2. Increase K_P from zero until the system starts to oscillate with a consistent frequency.

Ziegler–Nichols method[1]

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$0.8\overline{3}T_u$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$0.125T_u$	–	$0.10K_uT_u$
classic PID[2]	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_uT_u$
Pessen Integral Rule[2]	$0.7K_u$	$0.4T_u$	$0.15T_u$	$1.75K_u/T_u$	$0.105K_uT_u$
some overshoot[2]	$0.3\overline{3}K_u$	$0.50T_u$	$0.3\overline{3}T_u$	$0.6\overline{6}K_u/T_u$	$0.1\overline{1}K_uT_u$
no overshoot[2]	$0.20K_u$	$0.50T_u$	$0.3\overline{3}T_u$	$0.40K_u/T_u$	$0.06\overline{6}K_uT_u$

Figure 7: Ziegler-Nichols tuning method table.

3. Measure the ultimate gain K_u and the oscillation period T_u.
4. Set the control constants according to the Ziegler-Nichols table.

The Ziegler-Nichols tuning table provides the appropriate values for K_p, K_i, and K_d, depending on whether the system is tuned for P, PI, or PID control, as shown in Figure 7. While Ziegler-Nichols can be effective in many cases, it may not provide optimal performance for disturbance rejection or systems with complex dynamics.

Internal Model Control (IMC) Tuning Method: The Internal Model Control (IMC) tuning method provides better disturbance rejection compared to Ziegler-Nichols. IMC uses a model of the system to predict its behavior and adjust the control accordingly. The key advantage of IMC is its ability to handle unexpected changes and provide smoother control performance.

The IMC tuning process is based on the idea of designing a controller G_c^* using a low-pass filter f with two key parameters: the desired closed-loop time constant τ_c and a filter parameter r. The IMC tuning procedure involves breaking the system model into its time delays and stable parts, then designing the controller accordingly (Figure 8).

Genetic Algorithm Tuning: Another modern approach to tuning PID controllers is through genetic algorithms (GA). GA-based tuning evaluates a population of potential solutions, evolving them across generations to minimize the cost function (e.g., cumulative error). In this approach, the fitness function is defined as the sum of errors in the pendulum angle and cart position. The fitness function is

$$\text{fitness} = \sum_i (\theta_i - \theta_r)^2 + \sum_i (x_i - x_r)^2$$

Genetic algorithms operate using selection, crossover, and mutation. The best individuals (solutions) are selected based on their fitness scores, and new solutions are generated by combining the best-performing ones. Small random mutations are introduced to maintain diversity in the

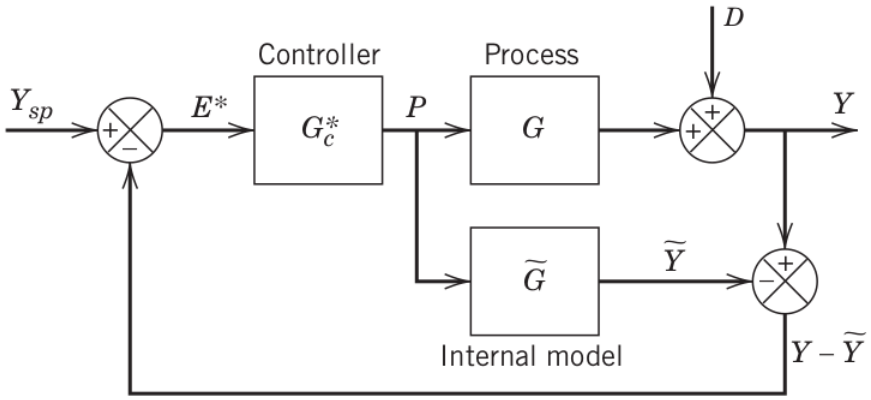

Figure 8: Internal Model Control (IMC) design and structure.

population, helping to explore the solution space effectively. This method is particularly useful for complex systems where traditional tuning methods may not yield optimal results.

The primary challenge in controlling the inverted pendulum is its inherent instability. Both the IMC and Ziegler-Nichols methods rely on assumptions of a stable, first-order system, making them unsuitable for the inverted pendulum. While Ziegler-Nichols was initially applied to the pendulum's PD controller, this approach was based on a misunderstanding,i was unaware at the time that the method requires system stability.

5.2 Cascade Controller

The cascade controller introduces a multi-loop control structure, typically composed of an outer loop and an inner loop, where one loop's control action influences the other. The outer loop controls the primary variable (e.g., position), while the inner loop manages a secondary variable (e.g., velocity or force) [5]. The control laws for the two loops are given by:

$$u_1(t) = K_{p1}e_1(t) + K_{i1}\int e_1(t)dt + K_{d1}\frac{d}{dt}e_1(t) \quad \text{(Outer loop)}$$

$$u_2(t) = K_{p2}e_2(t) + K_{i2}\int e_2(t)dt + K_{d2}\frac{d}{dt}e_2(t) \quad \text{(Inner loop)}$$

Cascade control is useful in systems where multiple variables are dynamically coupled. In this structure, the outer loop influences the inner loop, which can respond more rapidly to disturbances. The inner loop typically manages faster dynamics, while the outer loop provides overall control.

Advantages: Cascade control improves disturbance rejection and can handle interacting variables more effectively. By breaking the control problem into two loops, the controller can manage different timescales and system behaviors more precisely, leading to better performance in coupled or multi-variable systems.

Disadvantages: Designing and tuning a cascade controller can be complex, as it requires an accurate understanding of the system dynamics. Both loops must be carefully tuned, which can be time-consuming, and errors in tuning one loop can destabilize the entire system.

5.3 Fuzzy Controller

Fuzzy control is a method that uses fuzzy logic to handle uncertainty and nonlinearity in systems. Unlike traditional controllers that rely on precise mathematical models, fuzzy controllers operate using a set of heuristic rules and membership functions to make control decisions [3]. The control action is dynamically adjusted based on the system's behavior, using fuzzy inference to determine the appropriate outputs.

In a fuzzy control system, membership functions define how much a particular input belongs to different fuzzy sets (e.g., small, medium, large). Control rules are then applied based on these fuzzy sets to adjust the output. For example, a rule might be: "If the error is small and the rate of change of the error is medium, then the control action should be moderate."

Advantages: Fuzzy control is highly effective for managing nonlinear systems and systems where the dynamics are not well understood or subject to frequent changes. It is adaptive and does not require an exact mathematical model, making it flexible and robust in handling uncertainties.

Disadvantages: The implementation of fuzzy control is more complex than traditional control methods, as it requires defining appropriate membership functions and rules. Tuning a fuzzy controller can also be difficult and time-consuming, especially in systems with many inputs and outputs.

5.4 Linear Quadratic Regulator (LQR)

The Linear Quadratic Regulator (LQR) is an optimal control technique widely used in systems where the goal is to minimize a cost function that balances the trade-off between minimizing state errors and control effort. It provides an efficient way to control linear systems by finding the optimal control law through feedback.

The cost function for LQR is expressed as:

$$J(t) = \int_0^\tau \left(x(t)^T Q x(t) + u(t)^T R u(t) \right) d\tau$$

In this equation, $x(t)$ represents the system's state vector, $u(t)$ is the control input, and Q and R are symmetric, positive semi-definite weighting matrices. The matrix Q penalizes deviations in the state, while R penalizes the control effort, allowing designers to balance the relative importance of state accuracy versus the cost of control actions [7].

LQR computes the optimal feedback gain matrix K by solving the continuous-time algebraic Riccati equation (CARE). This feedback gain K is used in the control law to compute the optimal control input as follows:

$$u(t) = -Kx(t)$$

Here, K is the optimal gain matrix that ensures minimal control effort while maintaining the desired system performance by driving the state vector $x(t)$ toward zero or another target state.

Advantages of LQR: LQR guarantees optimal performance for linear systems by ensuring both stability and efficiency in control. It provides a systematic way to tune the balance between control effort and state accuracy through the Q and R matrices. Moreover, LQR's formulation inherently ensures that the system remains stable, given the correct tuning of these matrices.

Disadvantages of LQR: One of the main limitations of LQR is that it assumes a linear system model. For nonlinear systems, LQR may not yield optimal results unless the system can be accurately linearized around an operating point. Additionally, LQR requires that the system be observable; otherwise, effective pole placement may not be possible. In cases where the system is not fully observable, a Kalman filter may be employed to estimate the states needed for feedback.

5.5 Model Predictive Control (MPC)

Model Predictive Control (MPC) is a more advanced control strategy designed to handle multivariable control problems, where future behavior and constraints need to be explicitly considered. Unlike LQR, which applies a static feedback gain, MPC continuously optimizes the control input over a finite prediction horizon, solving an optimization problem at every time step to find the best control actions.

The primary objective of MPC is to minimize state tracking errors and control effort over a future prediction horizon. The cost function in MPC typically takes a similar form to LQR, penalizing deviations from the desired trajectory and control effort:

$$J = \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k \right)$$

N represents the prediction horizon, x_k is the predicted state at time step k, and u_k is the control input. Like LQR, the matrices Q and R serve as tuning parameters, but MPC also handles constraints on states and inputs, making it more versatile for real-world applications. At each time step, MPC solves a constrained optimization problem to minimize the cost function while respecting any operational constraints. Once the optimal solution is found, only the first control action is applied, and the process is repeated at the next step using updated state measurements, making MPC a receding horizon control method.

Advantages of MPC: One of the key strengths of MPC is its ability to handle multivariable systems with constraints, making it highly suitable for complex systems with multiple control objectives. It is particularly effective for systems where constraints on states or inputs are critical, as it explicitly accounts for these constraints in its optimization process.

Disadvantages of MPC: The main drawback of MPC is its computational complexity, as it requires solving a potentially large-scale optimization problem at each time step. This makes it computationally intensive and may not be suitable for systems with very fast dynamics unless sufficient computational resources are available. Furthermore, like LQR, MPC relies on having an accurate model of the system. If the model is not well-calibrated, the performance of the controller may degrade.

6 Control System Design

This section provides a detailed description of the control system design for the inverted pendulum system. It outlines the control objectives, manipulated and controlled variables, and explains the control structure. Additionally, it describes the tuning strategies applied to the controllers, including PD, cascade PD, and fuzzy PD controllers.

6.1 System Specifics and Structure

The primary goal of the control system is to stabilize the inverted pendulum in an upright position with the pendulum angle $\theta = 0°$ and to minimize the cart's movement around its initial position, $x = 0$. The system must maintain stability even in the presence of external disturbances, ensuring that the cart's displacement remains small and controlled during the pendulum's stabilization. To achieve this, the control system is designed to have a fast response time with minimal overshoot and oscillations, as excessive swings in the pendulum would make control difficult.

In this setup, the force applied to the cart is the manipulated variable, and the controlled variables are the pendulum angle θ and cart position x. The system operates as a Single Input, Multiple Output (SIMO) system, where the single input (force on the cart) affects both the pendulum's angle and the cart's position. The set point is defined as $[\theta = 0°, \dot{\theta} = 0, x = 0, \dot{x} = 0]$, with the goal of keeping the pendulum upright and the cart close to its initial position.

6.2 Tuning the System

For the controller design, a PD controller was selected due to its effectiveness in managing rise time, overshoot, and oscillations. The PD controller is suitable for systems like the inverted pendulum, where precise control is needed to balance the pendulum and minimize oscillatory behavior. In this case, a mixed tuning strategy was employed: the controller constants for the cart were tuned manually, while the ZieglerâNichols method was applied to tune the pendulum controller.

For the cart system, the controlled transfer function is expressed as:

$$\frac{K_p + K_d s}{s^2 + K_p + K_d s}$$

The stability conditions for this system are that $K_p > 0$ and $K_d > 0$, with an additional condition $K_d^2 \geq 4K_p$ to ensure that the system does not exhibit excessive oscillations. Using the ZieglerâNichols tuning method, the optimal controllers for the pendulum were determined. Although the pendulum is inherently unstable and ZieglerâNichols tuning alone might not work effectively, by first adjusting the cart control parameters manually and then applying Zieglerâs method to the pendulum, the overall system response was satisfactory.

For the cascade control strategy, a genetic algorithm was used to perform the tuning of both the inner and outer loops. The genetic algorithm was configured with a population size of 50, a mutation rate of 0.1, and ran for 250 generations. The tuning of the fuzzy controller was conducted through a trial-and-error approach, using the PD values from the standard PD controller as a reference point.

6.3 PD Controller Response

The response of the system using the PD controller was analyzed by applying the tuned values for the controller constants. The resulting system behavior was observed to meet the design objectives, with minimal overshoot and oscillations, as well as a fast response time. The chosen controller constants were $K_p = 54.4$ and $K_d = 4.08$ for the pendulum, with $Kx_p = 1$ and $Kx_d = 4$ for the cart. The system response is shown in Figure 9.

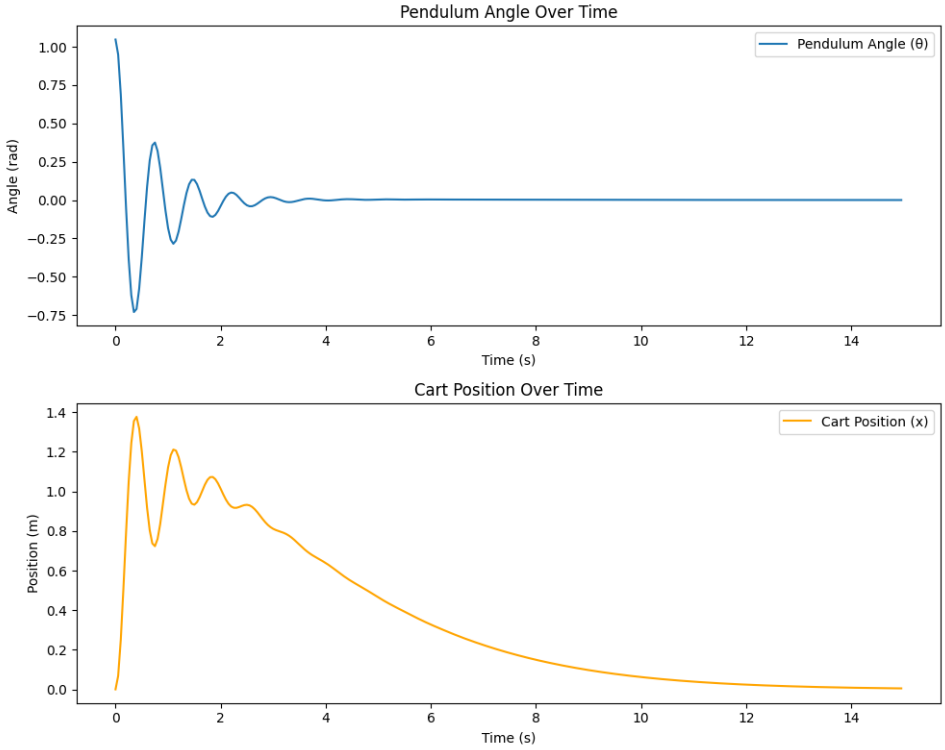

Figure 9: System response for the PD controller with $K_p = 54.4$, $K_d = 4.08$ and $Kx_p = 1$, $Kx_d = 4$.

6.4 Cascade PD Control

The cascade PD control strategy was also explored as part of the controller design. This approach involves two nested control loops: an outer loop that controls the cart position and an inner loop that controls the pendulum angle. The outer loop applies a PD controller to the cart position, while the inner loop applies a PD controller to the pendulum angle.

For the cart position control, the PD control law is:

$$u_1 = Kx_p \cdot x + Kx_d \cdot \dot{x}$$

The control law for the pendulum angle in the inner loop is:

$$u_2 = K_p \cdot (\theta - u_1) + K_d \cdot (\dot{\theta} - \dot{u_1})$$

This cascade control structure allows for better handling of dynamic interactions between the cart and the pendulum. The system response for the cascade PD controller is shown in Figure 10, the parameters used were $K_p = 31$, $K_d = 5$, $Kx_p = -1$ and $Kx_d = -1$.

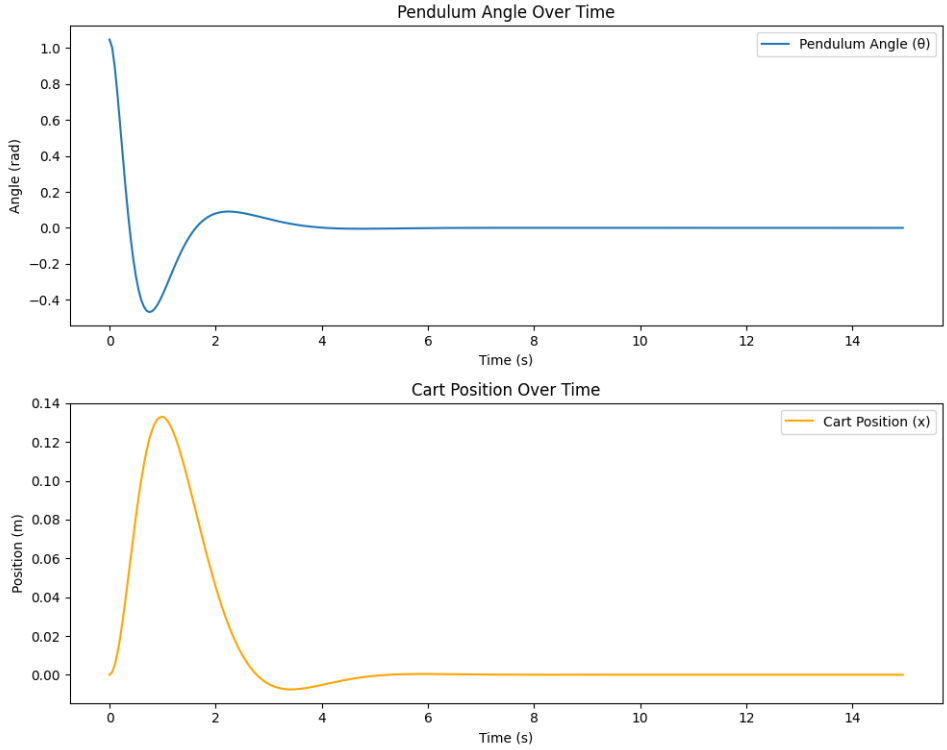

Figure 10: System response for the cascade PD controller.

6.5 Fuzzy PD Controller

Finally, a fuzzy PD controller was implemented to handle the nonlinearities and uncertainties present in the inverted pendulum system. The fuzzy PD controller uses fuzzy logic to dynamically adjust the proportional (K_p) and derivative (K_d) gains based on the current error and rate of change of error.

In this approach, fuzzy sets and membership functions are defined for both the error in the pendulum angle and cart position. The outputs of the fuzzy inference system determine the values of K_p and K_d, which are then applied to the system. Fuzzy rules govern the behavior of the controller, such as "if the error is negative and the rate of change is negative, then K_p and K_d should be medium." The table of the rules implemented for the controller of the pendulum are shown in Figure

Also there are the rules for the cart are

Figures 13 and 14 show the membership functions for the pendulum and cart, respectively:

The fuzzy PD controller proved effective in adapting to different conditions and handling nonlinearities. The system response for the fuzzy PD controller is shown in Figure 15.

Error	Delta Error	Kp	Kd
Negative	Negative	Medium	Medium
Negative	Zero	High	High
Negative	Positive	Medium	High
Zero	Negative	Medium	Medium
Zero	Zero	High	Medium
Zero	Positive	Medium	Medium
Positive	Negative	Medium	Medium
Positive	Zero	High	High
Positive	Positive	Medium	High

Figure 11: Fuzzy Rules for Kp and Kd based on angle error

Cart Error	Cart Delta Error	Kp	Kd
Negative	Negative	Low	Low
Negative	Zero	Low	Medium
Negative	Positive	Low	Medium
Zero	Negative	Low	High
Zero	Zero	Low	Low
Zero	Positive	Low	High
Positive	Negative	Low	Medium
Positive	Zero	Low	Medium
Positive	Positive	Low	Low

Figure 12: Fuzzy Rules for Kp and Kd based on position of the cart error

6.6 Linear Quadratic Regulator (LQR)

For the control of the inverted pendulum system, an LQR controller was implemented to ensure optimal performance by minimizing the cost function, which balances both the state error and the control effort. The weighting matrices Q and R were manually tuned through a trial-and-error process, based on the specific dynamics of the system:

$$Q = \text{diag}(30, 1, 35, 1), \quad R = \text{diag}(50)$$

These matrices were chosen to prioritize the pendulum's angular stability (with higher weights on the pendulum angle and position) while maintaining control over the cart's movement. The tuned parameters resulted in satisfactory system performance, stabilizing the pendulum in an upright position with minimal movement of the cart.

The resulting system response using LQR is shown in Figure 16, which demonstrates the effectiveness of the control strategy in minimizing oscillations and ensuring rapid stabilization.

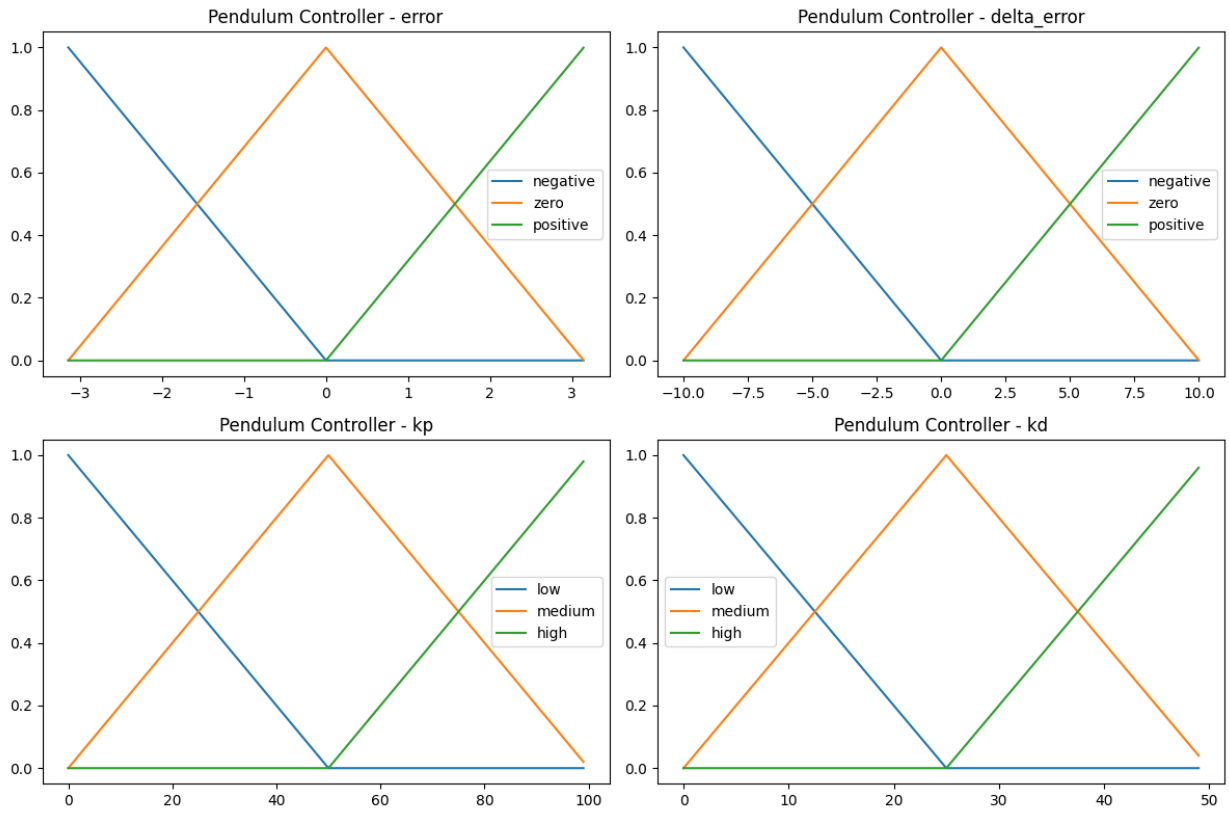

Figure 13: Membership functions for the pendulum.

6.7 Model Predictive Control (MPC)

MPC was implemented for the inverted pendulum system to optimize control inputs over a future time horizon, handling constraints on both the states and control inputs. The system dynamics were discretized using a zero-order hold (ZOH) method to allow for the control actions to be computed at each time step. The discretized state-space matrices A_{zoh} and B_{zoh} were used to model the system dynamics.

The cost function minimized by the MPC controller includes terms for both state tracking errors and control effort. The weighting matrices Q and R used in the optimization problem were manually tuned, with values chosen through trial and error to balance pendulum stabilization and control effort:

$$Q = \text{diag}(100, 1, 100, 1), \quad R = 1$$

The MPC optimization problem is solved at each time step using a prediction horizon of $N = 20$. The optimization process minimizes the cost function while ensuring the constraints on the system dynamics and control inputs are met. The first control input from the solution is applied, and the process repeats at the next time step with updated state measurements.

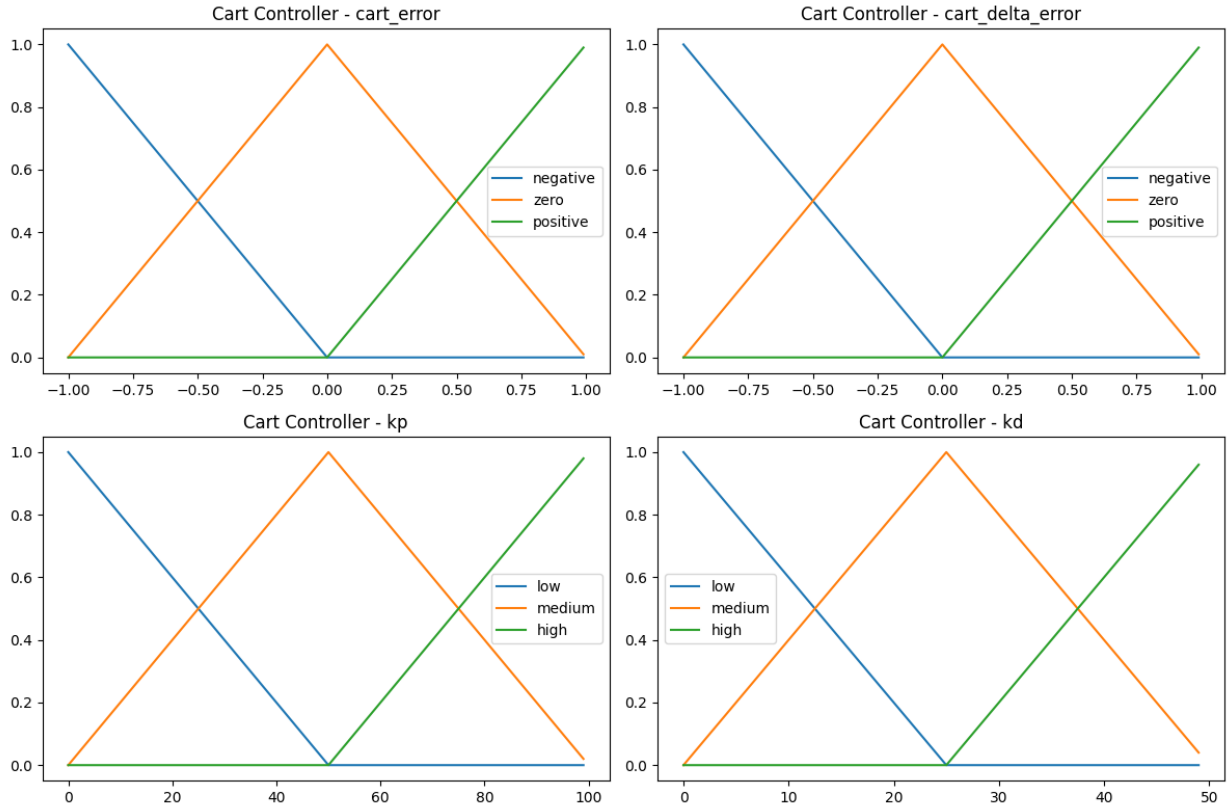

Figure 14: Membership functions for the cart.

The system response using the MPC strategy is shown in Figure 17. The results demonstrate that MPC effectively stabilizes the pendulum while controlling the cart position, adhering to the constraints imposed during the optimization process.

7 Controlled System Comparison

In this section, the performance of the different controller strategies PD, cascade PD, fuzzy PD, LQR, and MPC are compared in terms of their ability to track a set-point and reject disturbances. The set-point for the system was defined as $\text{floor}(t/6)$, causing the reference position to increase periodically over time. A disturbance function was introduced as $0.5 \times \left(\sin(t) \cdot e^{-\cos(t)} - \frac{t}{4} - e^{-t}\right)$ to simulate external perturbations affecting the system. The controllers were evaluated based on their response to both the set-point and the disturbance.

7.1 PD Controller Performance

The PD controller, as shown in Figure 18, performed not so well in tracking the set-point because of the slower response and it struggled to reject disturbances effectively. When the disturbance

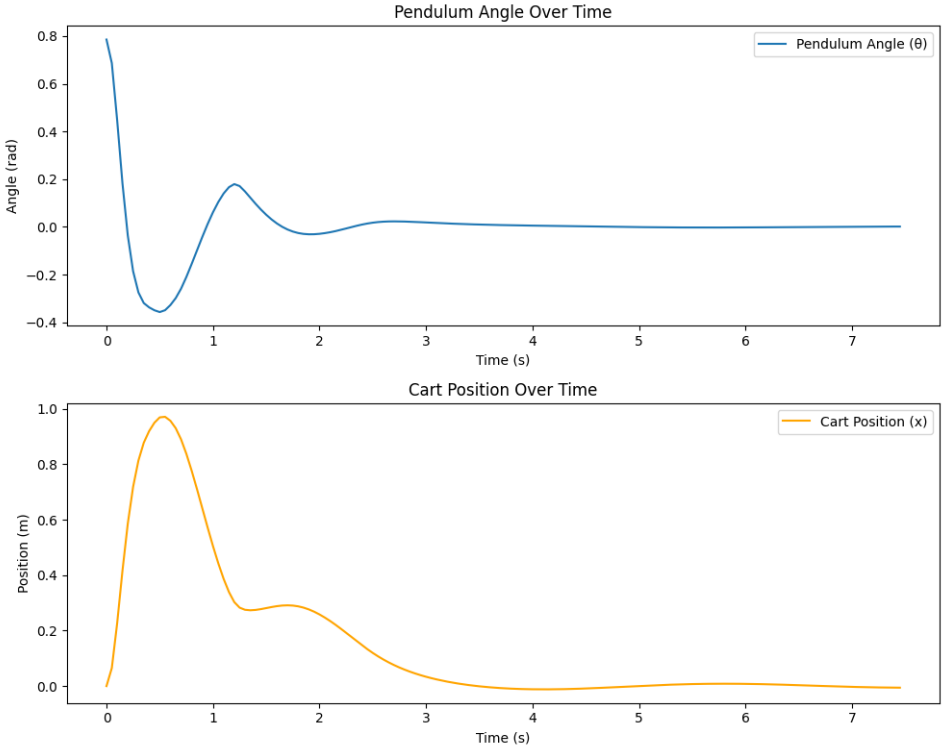

Figure 15: System response for the fuzzy PD controller.

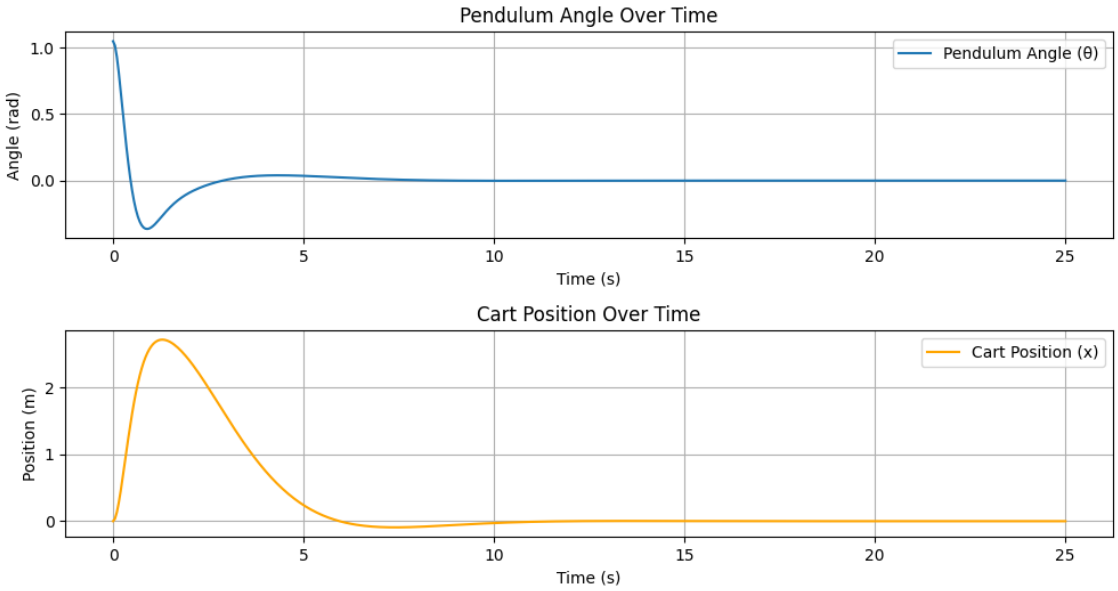

Figure 16: System response using LQR with tuned weighting matrices.

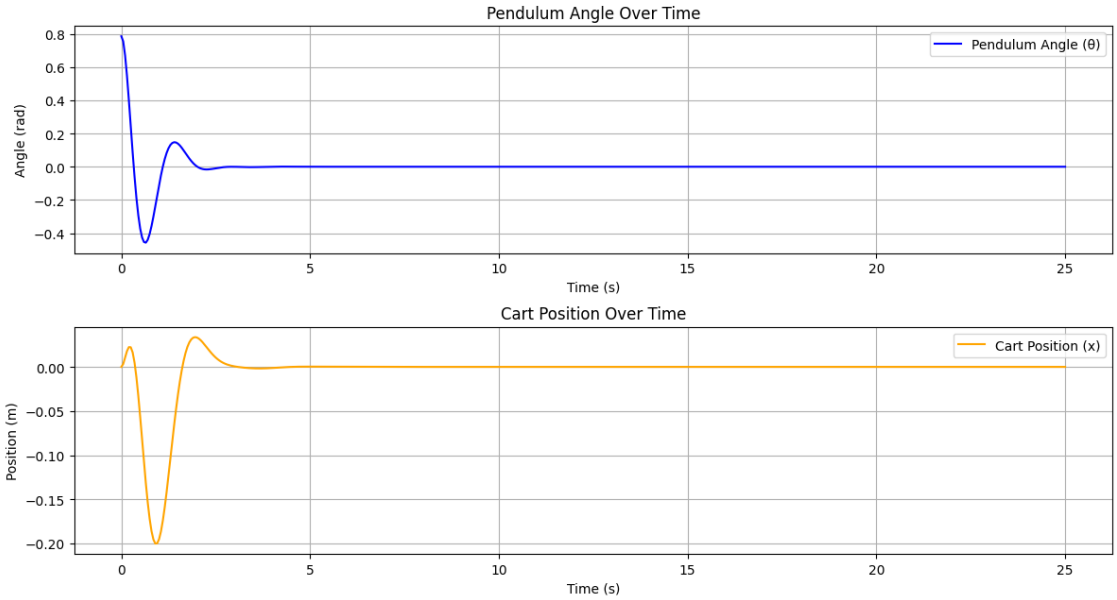

Figure 17: System response using MPC with discretized dynamics and horizon length $N = 20$.

was applied, the cart exhibited noticeable deviations from the set-point before recovering. Also the control action had enormous value going from 400 newtons to - 400 newtons.

7.2 Cascade PD Controller Performance

The cascade PD controller, illustrated in Figure 19, showed better disturbance rejection compared to the standard PD controller. Set-point tracking was also improved, with reduced oscillations and faster convergence to the desired position. The control action range also was decreased showing more potential.

7.3 Fuzzy PD Controller Performance

As shown in Figure 20, the fuzzy PD controller performed well in handling both set-point tracking (showed some overshoot) and disturbance rejection. In the beginning it had high value of the control action, but quickly fixed to be stable, this may be due to the set of rules defined and so it can be fixed but afterwards it was an good minimal control.

7.4 LQR Performance

The LQR controller, as seen in Figure 21, was good in both tracking and control action as expected, although it could be a bit faster on the tracking.

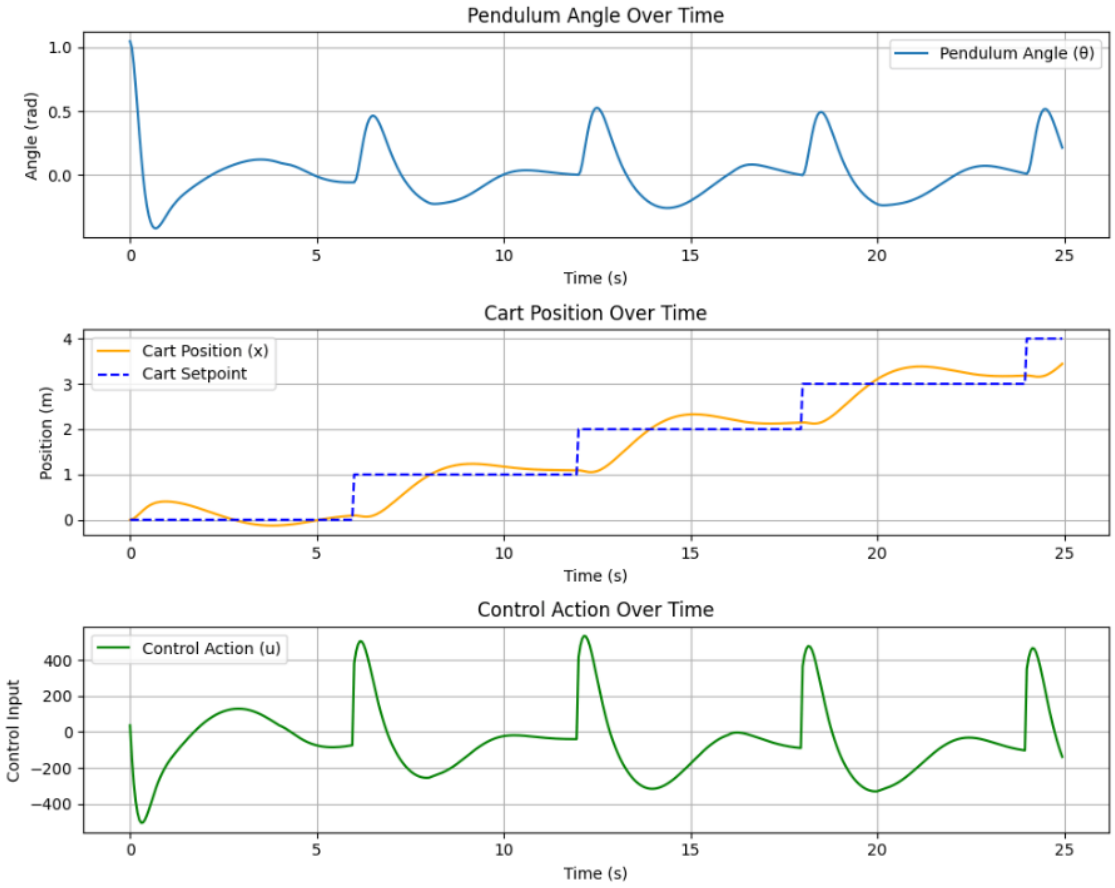

Figure 18: Set-point tracking and disturbance rejection using the PD controller.

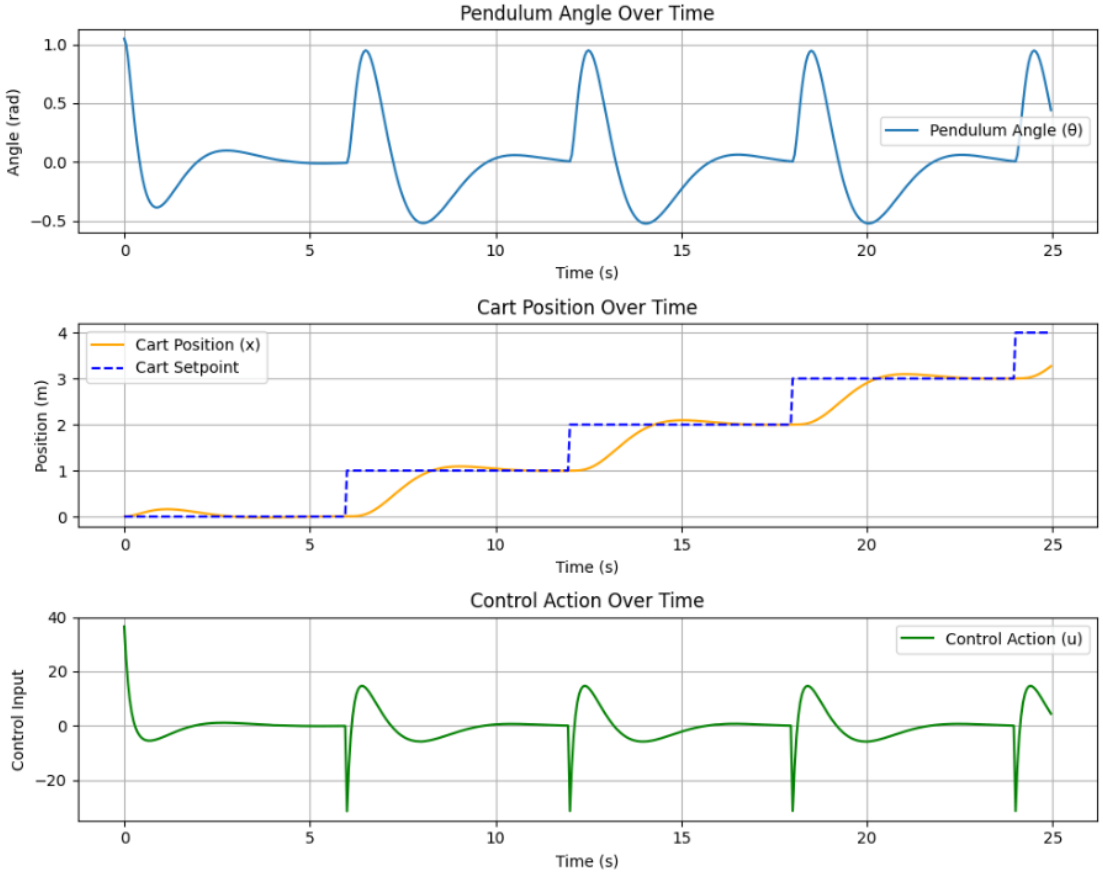

Figure 19: Set-point tracking and disturbance rejection using the cascade PD controller.

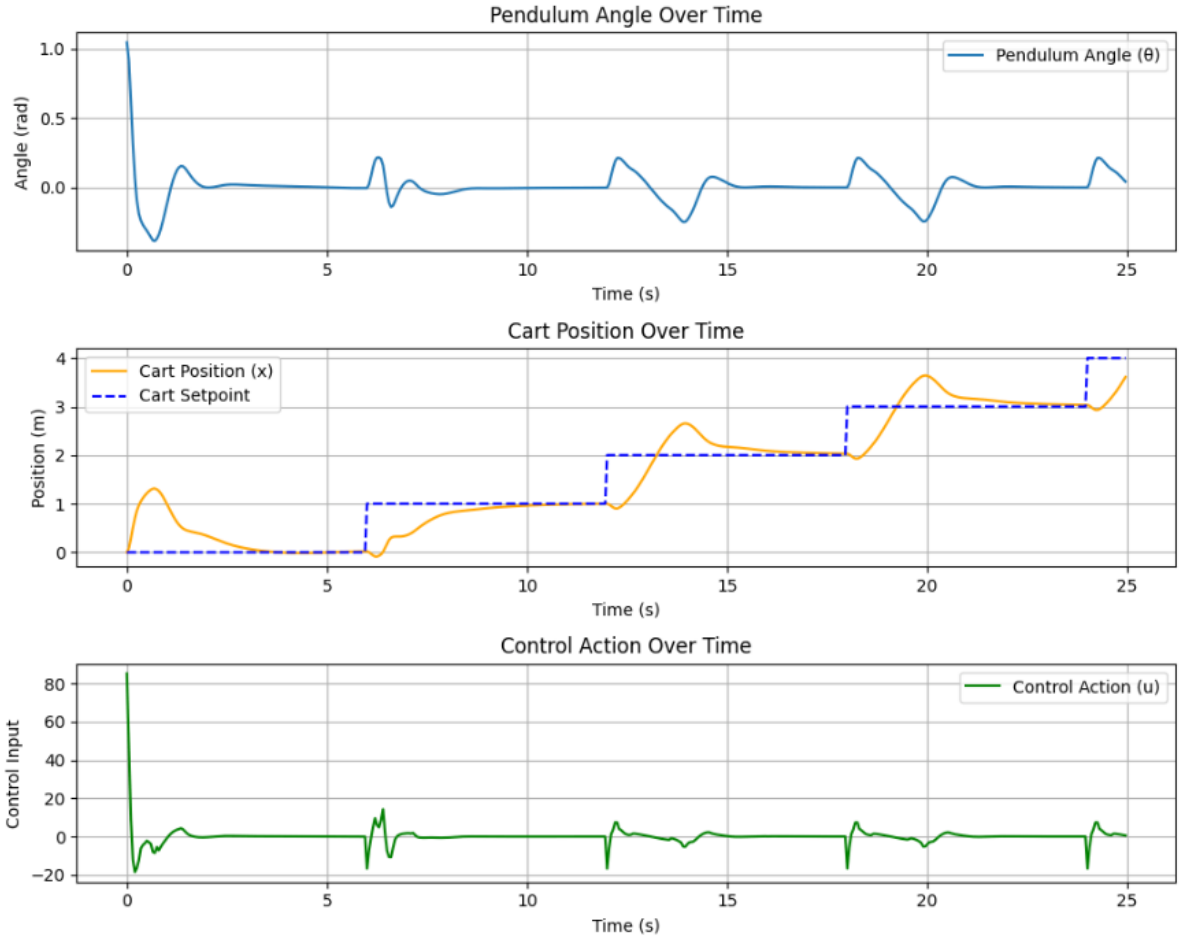

Figure 20: Set-point tracking and disturbance rejection using the fuzzy PD controller.

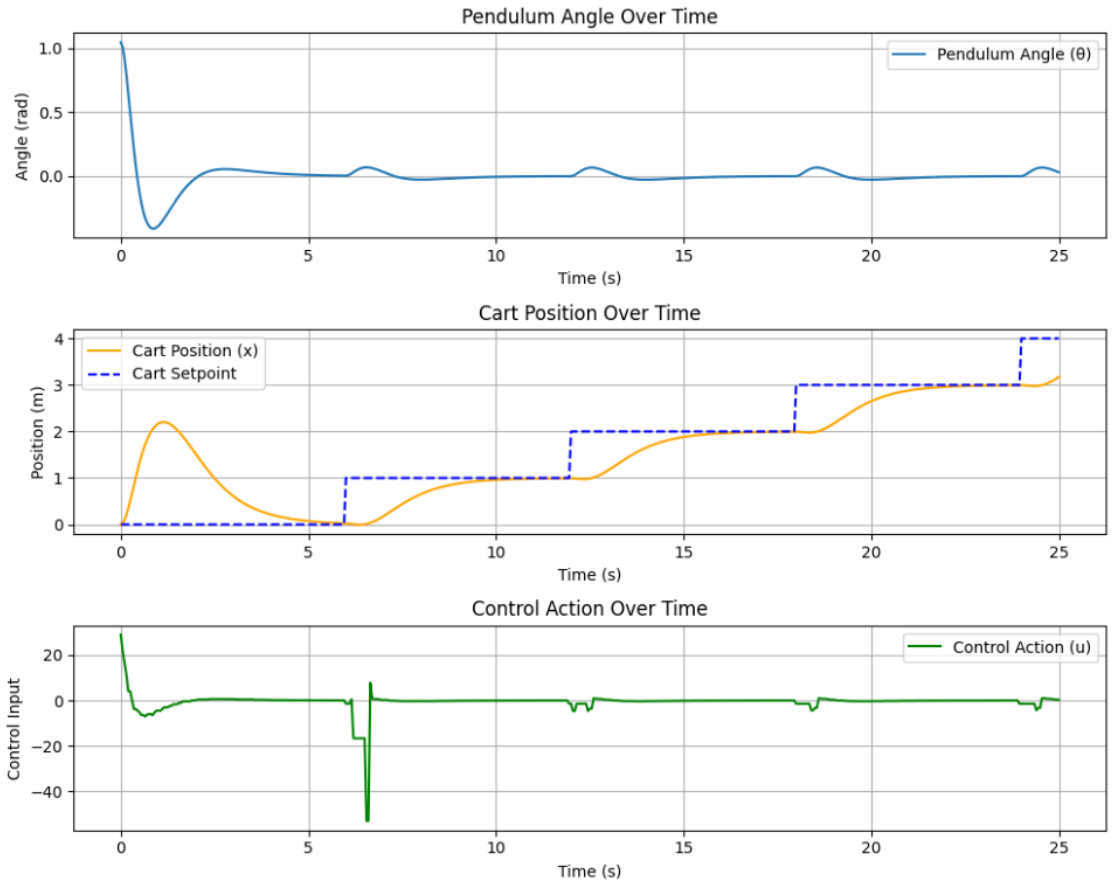

Figure 21: Set-point tracking and disturbance rejection using the LQR controller.

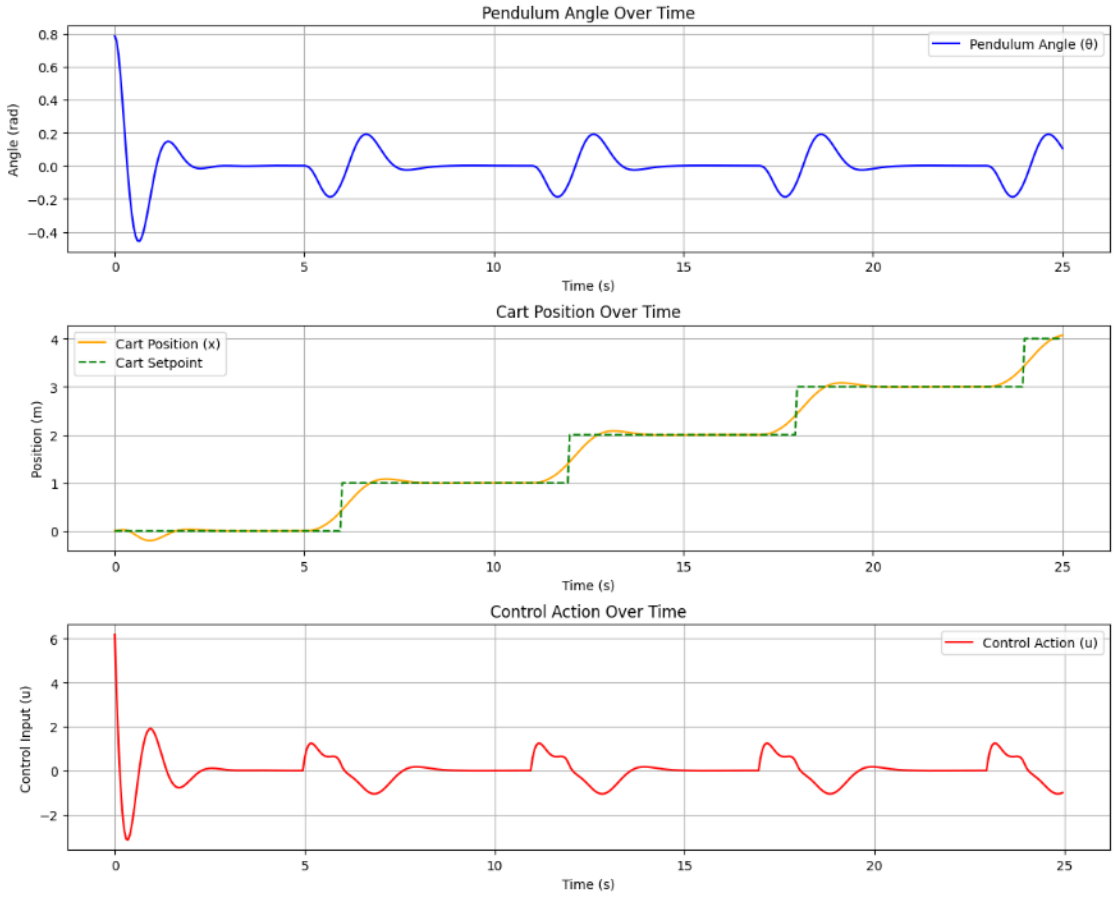

Figure 22: Set-point tracking and disturbance rejection using the MPC controller.

7.5 MPC Performance

Figure 22 shows the perfect performance of the MPC controller with the most minimal and yet effective control of all, showing quality of the method.

8 Conclusion

MPC offered the best performance, with precise tracking and minimal control effort, but at a high computational cost. LQR was a reliable alternative with good control, though slower than MPC. Fuzzy PD showed promise but needs further tuning. PD and cascade PD, while simpler, struggled with slower response and higher energy use, making them less effective for this system.

References

[1] O. Boubaker. The inverted pendulum benchmark in nonlinear control theory: A survey. *International Journal of Advanced Robotic Systems*, 10, 2013.

[2] Ioannis Kafetzis and Lazaros Moysis. Inverted Pendulum: A system with innumerable applications. 2017.

[3] Adriano Kossoski, Fernanda Cristina Correa, Angelo Marcelo Tusset, and JosÃ© Manoel Balthazar. Pid-fuzzy control design for a nonlinear inverted pendulum. *Journal of Advanced Intelligent Control*, 9(1), 2024.

[4] Alan V. Oppenheim. Signals and systems. MIT OpenCourseWare: Massachusetts Institute of Technology, 2011. Accessed: 2024-06-09.

[5] Fuat Peker, Ibrahim Kaya, Erdal Cokmez, and Serdal Atic. Cascade control approach for a cart inverted pendulum system using controller synthesis method. In *2018 26th Mediterranean Conference on Control and Automation (MED)*, pages 1–6, 2018.

[6] J. K. Roberge. The mechanical seal. Bachelor's thesis, Massachusetts Institute of Technology, 1960.

[7] Elisa Sara Varghese, Anju K Vincent, and V Bagyaveereswaran. Optimal control of inverted pendulum system using pid controller, lqr and mpc. *IOP Conference Series: Materials Science and Engineering*, 263(5):052007, nov 2017.