ME 580-06:
Deep Learning for Robot Control

Cart-Pole System
Dynamics and Control

Gizem Doğa Filiz
(gizemfiliz@sabanciuniv.edu - 29450)

Yunus Emre Danabaş
(yunusdanabas@sabanciuniv.edu - 29359)

December 31, 2024

Sabancı Üniversitesi

Contents

1 Introduction

The Cart-Pole system, a classical benchmark in control theory and reinforcement learning, serves as a fundamental testbed for evaluating and validating control strategies. Comprising a cart that moves horizontally with a pendulum hinged on top, the system exemplifies key challenges such as **underactuation**—where the control input indirectly influences the pendulum's dynamics—and **nonlinearity** in its behavior, especially during large-angle deviations. Mastering control of the Cart-Pole system not only advances theoretical understanding but also informs the development of more complex robotic systems and real-world applications.

1.1 Motivation and Objectives

Controlling the Cart-Pole system encapsulates essential challenges in nonlinear control, including:

- **Underactuation**: The control force applied to the cart affects the pendulum indirectly, complicating the control process.
- **Nonlinearity**: The pendulum exhibits complex dynamics, particularly when swinging through large angles, making linear control approaches insufficient.
- **Trade-Offs**: Balancing energy-efficient swing-up maneuvers with precise stabilization requires sophisticated control strategies.

This project aims to design, implement, and evaluate diverse controllers that address these challenges by leveraging both classical and modern control techniques. Specifically, the objectives are to:

1. Develop a linear feedback controller for stabilization around the upright equilibrium.
2. Implement a neural network-based controller to handle nonlinear dynamics during the swing-up phase.
3. Create a Mixture-of-Experts (MoE) controller that integrates specialized controllers for both swing-up and stabilization.
4. Compare and analyze the performance of these controllers to understand their strengths and limitations.

1.2 Overview of Control Strategies

Three distinct control approaches are explored to tackle the Cart-Pole control problem:

1. **Linear Controller Design:** A classical feedback law using linear combinations of state variables to stabilize the pendulum around the upright position. This method serves as a baseline and is benchmarked against the Linear Quadratic Regulator (LQR) for performance evaluation.
2. **Neural Network (NN) Controller:** A data-driven approach employing a neural network to generate control forces. The NN is trained to manage the swing-up maneuver, effectively handling the system's nonlinear dynamics.
3. **Mixture-of-Experts (MoE) Controller:** A hybrid framework that combines specialized controllers for different operational phases. This includes:
   - **Deterministic Gating Network:** Utilizes predefined thresholds to switch between the NN controller for swing-up and the LQR controller for stabilization.
   - **Learned Gating Network:** Employs a trainable network to dynamically assign weights to each expert based on the current state, allowing for more nuanced control.

These strategies are selected to demonstrate a spectrum of control methodologies, from simple linear feedback to advanced hybrid systems that integrate multiple control paradigms.

1.3 Scope of the Report

This report is structured to provide a comprehensive analysis of each control approach:

- **System Description and Modeling**: Detailed dynamics and state-space representation of the Cart-Pole system.
- **Controller Designs**: Methodologies and implementation details for the Linear, Neural Network, and Mixture-of-Experts controllers.
- **Comparative Analysis**: Evaluation of controller performance based on metrics such as stability, control effort, and response time.
- **Discussion**: Insights into the strengths, limitations, and practical considerations of each control strategy.
- **Future Directions**: Identified challenges and potential improvements for advanced controller designs.

The report aims to not only present the technical implementations but also to critically assess the effectiveness of each controller in addressing the inherent challenges of the Cart-Pole system.

2 System Description and Equations of Motion

The Cart-Pole system is a classic benchmark in control theory and reinforcement learning, illustrating key challenges of *nonlinearity* and *underactuation*. It consists of a cart of mass m_c that moves along a horizontal track, with a pendulum of mass m_p and length ℓ pivoted on top. The principal control objectives are:

- **Swing-Up**: Moving the pendulum from its downward position to the upright equilibrium.
- **Stabilization**: Maintaining the pendulum near the upright position once achieved.

2.1 State Variables and Dynamics

The system state is represented by:

$$X = \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix}, \tag{1}$$

where x and $\dot{x}$ are the cart's position and velocity, respectively, and θ and $\dot{\theta}$ denote the pendulum's angular position and velocity. A horizontal force $f(t)$ is applied to the cart, directly affecting its motion and indirectly influencing the pendulum's dynamics via gravitational coupling.

2.2 Equations of Motion

Using a Lagrangian approach, the Cart-Pole dynamics are expressed as:

$$M(q)\,\ddot{q} \;+\; C(q,\dot{q})\,\dot{q} \;=\; \tau_g(q) \;+\; Bf, \tag{2}$$

where $q = \begin{bmatrix} x \\ \theta \end{bmatrix}$ and:

- **Mass Matrix:**

$$M(q) = \begin{bmatrix} m_c + m_p & -m_p\,\ell\cos(\theta) \\ -m_p\,\ell\cos(\theta) & m_p\,\ell^2 \end{bmatrix}. \tag{3}$$

- **Coriolis and Centrifugal Terms:**

$$C(q,\dot{q}) = \begin{bmatrix} 0 & m_p\,\ell\sin(\theta)\,\dot{\theta} \\ 0 & 0 \end{bmatrix}. \tag{4}$$

- **Gravitational Forces:**

$$\tau_g(q) = \begin{bmatrix} 0 \\ m_p\, g\, \ell\, \sin(\theta) \end{bmatrix}. \tag{5}$$

- **Control Input Matrix:**

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{6}$$

2.3 Linearization Around the Upright Equilibrium

For small-angle deviations around $\theta = 0$, the system can be approximated using $\sin\theta \approx \theta$ and $\cos\theta \approx 1$. Under these assumptions,

$$\dot{X} \;\approx\; A\,X \;+\; B\,f, \tag{7}$$

where A and B are obtained from the partial derivatives of the system dynamics at the upright equilibrium. This linearization simplifies controller design (e.g., LQR) but holds only for small-angular variations.

2.4 State-Space Representation

To facilitate simulation and control, the second-order system is recast as a first-order state-space model:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x}(x, \theta, \dot{x}, \dot{\theta}, f) \\ \ddot{\theta}(x, \theta, \dot{x}, \dot{\theta}, f) \end{bmatrix}, \tag{8}$$

where $\ddot{x}$ and $\ddot{\theta}$ depend on the current state and the applied force. This formulation is essential for numerical ODE solvers and underpins the various controller implementations discussed in subsequent sections.

3 Linear Controller Design (and Comparison with LQR)

This section outlines a **Linear Controller** for the Cart-Pole system and compares it against a Linear Quadratic Regulator (LQR). The primary goal is to stabilize the pendulum near its upright equilibrium.

3.1 Controller Objective

We propose a feedback law of the form

$$f(x) \;=\; w_1 x + w_2 \cos(\theta) + w_3 \sin(\theta) + w_4 \dot{x} + w_5 \dot{\theta},$$

where $\{w_1, w_2, w_3, w_4, w_5\}$ are trained via gradient-based optimization. Incorporating both $\sin(\theta)$ and $\cos(\theta)$ extends conventional linearization, offering greater robustness for moderate angular deviations.

3.2 Controller Methodology

3.2.1 Linear Control Law Formulation

The control force

$$f(x) \;=\; w_1 x + w_2 \cos(\theta) + w_3 \sin(\theta) + w_4 \dot{x} + w_5 \dot{\theta}$$

depends on the learned parameter vector

$$w = [w_1,\ w_2,\ w_3,\ w_4,\ w_5]^T.$$

While more flexible than a purely linearized approach, it remains a relatively simple linear feedback law.

3.2.2 Weight Training Process

Sampling Initial Conditions To ensure broad coverage of near-upright scenarios, we uniformly sample 20 initial conditions from:

$$x \in (-0.5, 0.5), \quad \theta \in (-0.5, 0.5)\,\text{rad}, \quad \dot{x} \in (-0.5, 0.5), \quad \dot{\theta} \in (-0.5, 0.5).$$

Optimization Algorithm and Loss Function Using `Adam` (learning rate $\approx 10^{-4}$) for up to 50,000 iterations or until convergence, each iteration proceeds as:

1. **Simulation Rollouts**: Simulate the Cart-Pole under the current weights for each initial condition (up to 13 s).
2. **Cost Computation**:
$$J(w) = \sum_{k=1}^{N} (X_k^T Q X_k + f_k^2),$$
with
$$Q = \mathrm{diag}(100,\ 200,\ 5,\ 20).$$
3. **Gradient Update**: Apply automatic differentiation in JAX to iteratively refine w.

Upon training completion, the optimal weights

$$w = [\,10.7477,\ -0.0014,\ -95.3457,\ 13.2174,\ -27.8421\,]$$

achieve a final cost of roughly 10.231368. Notably, $w_2 \approx 0$ indicates minimal reliance on $\cos(\theta)$, while w_3 (linked to $\sin(\theta)$) strongly influences pendulum correction.

3.3 Comparison with LQR Controller

An **LQR** controller is designed around a linearized model. Solving the continuous-time Algebraic Riccati Equation yields a gain vector

$$K = [-31.6228,\ 186.6342,\ -33.5671,\ 57.2319],$$

leading to the control law

$$f_{\mathrm{LQR}} = -K\,[x,\ \theta,\ \dot{x},\ \dot{\theta}]^T.$$

3.3.1 Performance Metrics and Figures

Both controllers are tested in three scenarios, each yielding:

- `_all_states.png` for $(x, \dot{x}, \theta, \dot{\theta})$ trajectories,
- `_trajectory_cost.png` for total cost,
- `_control_force.png` for force profiles.

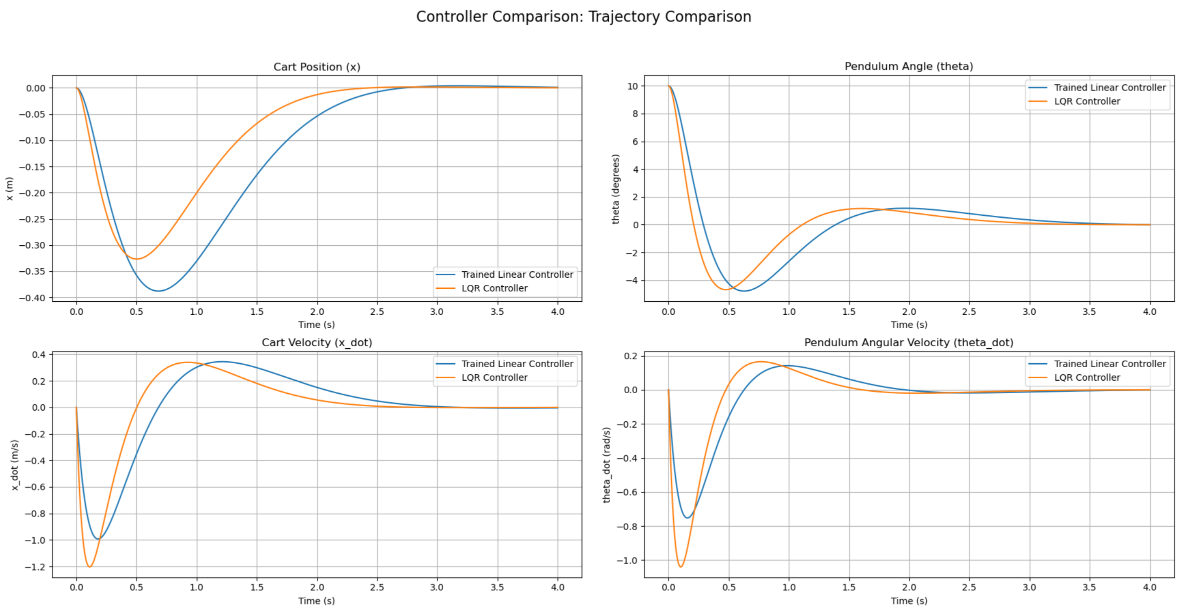

Figure 1: State trajectories $(x, \dot{x}, \theta, \dot{\theta})$ for [0.0, 10°, 0.0, 0.0]. Linear vs. LQR.

Case 1: [0.0, 10°, 0.0, 0.0] The trajectory cost is shown in Figure 2, where LQR's faster stabilization yields lower cost. Control forces in Figure 3 reveal smoother signals for the Linear Controller but higher immediate force from LQR.

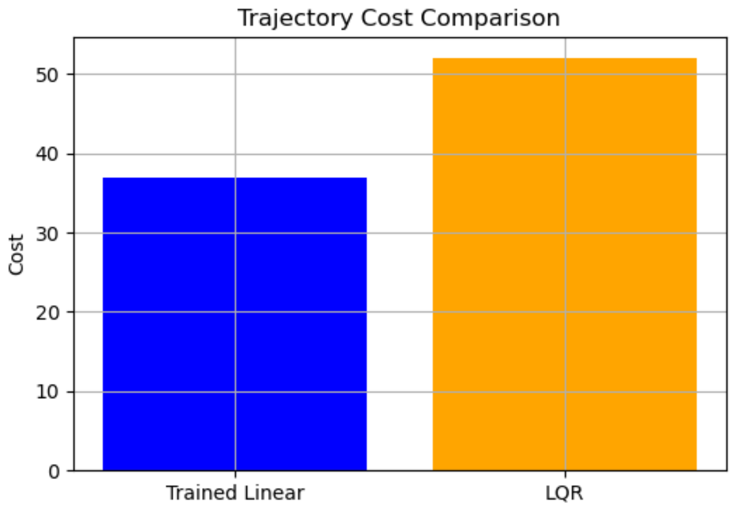

Figure 2: Trajectory costs for [0.0, 10°, 0.0, 0.0]. Linear vs. LQR.

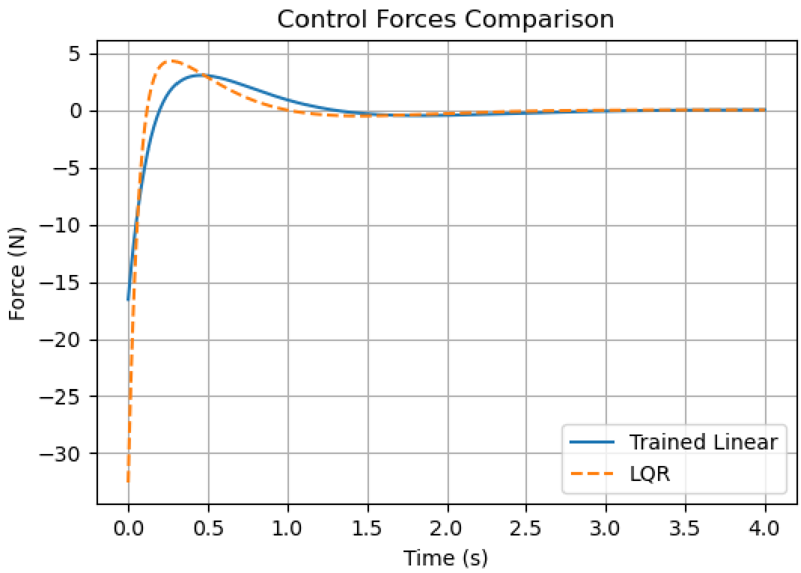

Figure 3: Control forces for [0.0, 10°, 0.0, 0.0]. Linear vs. LQR.

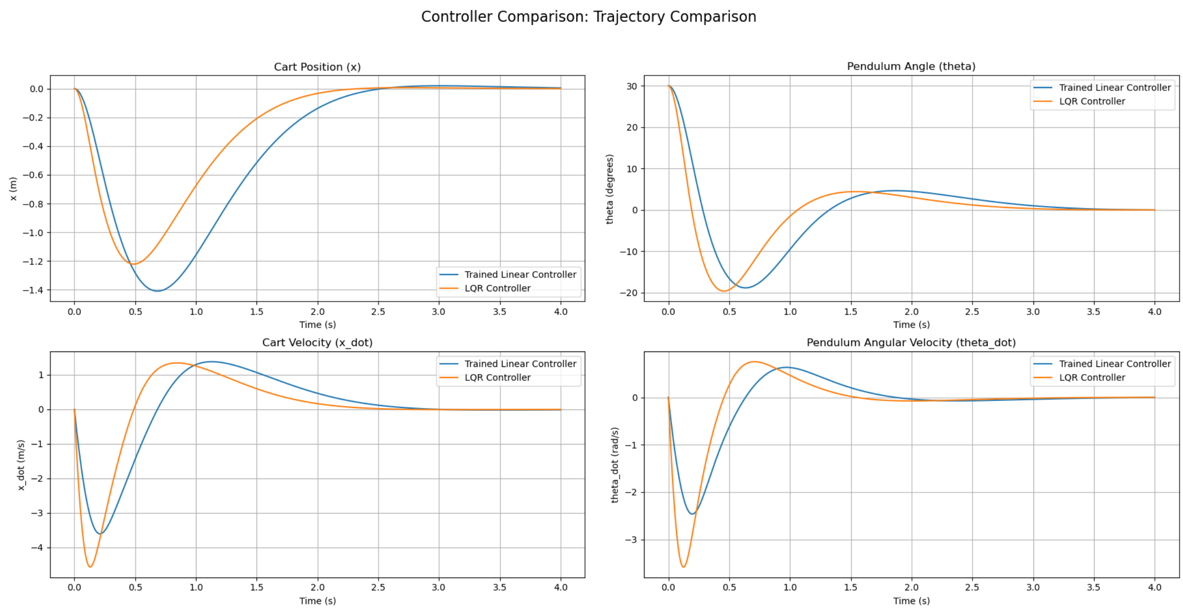

Figure 4: State trajectories for [0.0, 30°, 0.0, 0.0]. Linear vs. LQR.

Case 2: [0.0, 30°, 0.0, 0.0] The corresponding costs (Figure 5) indicate higher stabilization effort for the Linear Controller. Control forces (Figure 6) show greater force amplitudes for LQR, enabling a more aggressive correction.

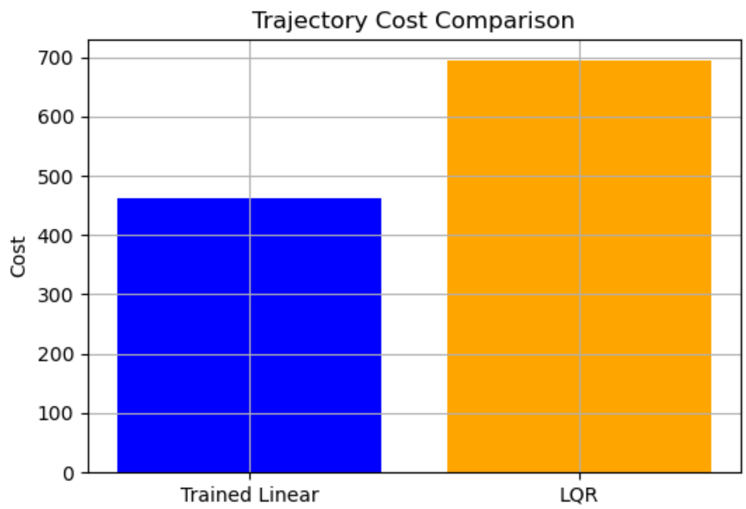

Figure 5: Trajectory costs for [0.0, 30°, 0.0, 0.0]. Linear vs. LQR.

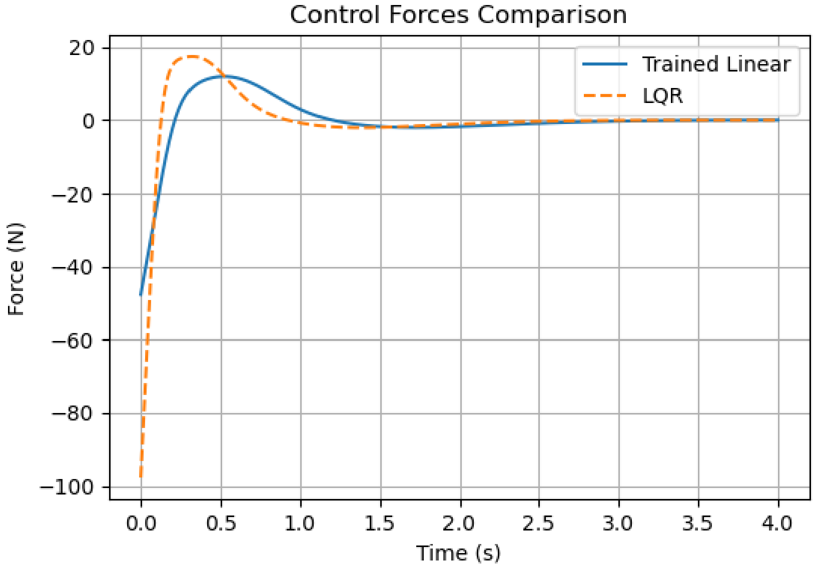

Figure 6: Control forces for [0.0, 30°, 0.0, 0.0]. Linear vs. LQR.

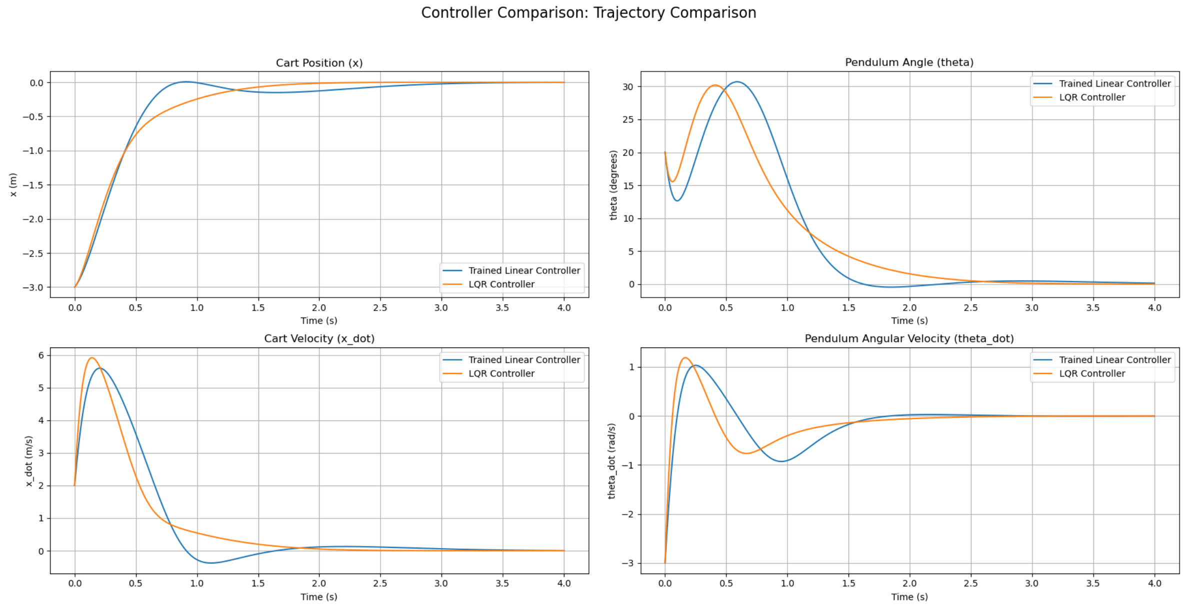

Figure 7: State trajectories for [−3.0, 20°, 2.0, −3.0]. Linear vs. LQR.

Case 3: [−3.0, 20°, 2.0, −3.0] In this more challenging scenario, LQR outperforms the Linear Controller in both total costs (Figure 8) and control forces (Figure 9). The Linear Controller struggles to achieve rapid stabilization, resulting in elevated costs.

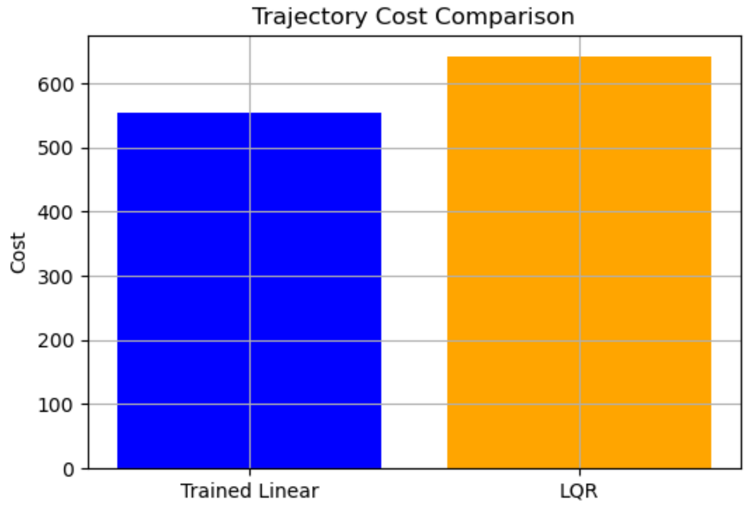

Figure 8: Trajectory costs for $[-3.0, 20°, 2.0, -3.0]$. Linear vs. LQR.

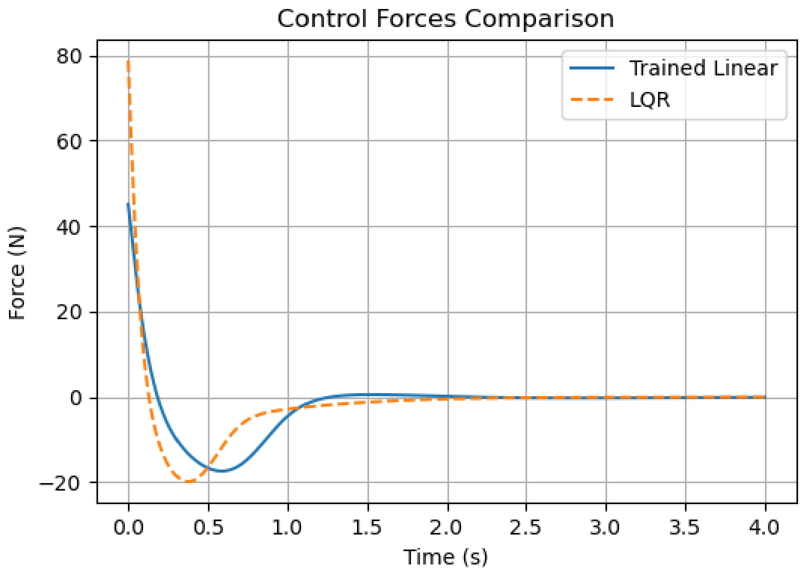

Figure 9: Control forces for $[-3.0, 20°, 2.0, -3.0]$. Linear vs. LQR.

3.4 Analysis and Discussion

3.4.1 Controller Performance Evaluation

Stabilization Effectiveness The Linear Controller consistently balances the pendulum near upright but converges more slowly for larger deviations (Cases 2 and 3). LQR's analytical design yields lower costs and faster convergence.

Response Time and Overshoot Although the Linear Controller applies smoother forces, it takes longer to correct deviations. Conversely, LQR swiftly reduces θ to zero, albeit with higher initial force demands.

Control Effort In many scenarios, the Linear Controller uses lower forces overall (potentially reducing actuator stress and energy consumption), yet incurs higher final trajectory costs and slower stabilization times.

3.4.2 Strengths and Limitations

Strengths

- **Simplicity**: Only five parameters require optimization, making implementation straightforward.
- **Computational Efficiency**: Once trained, the runtime control law is rapid to evaluate.
- **Robustness for Moderate Angles**: Incorporating $\sin(\theta)$ and $\cos(\theta)$ helps manage angles beyond a purely small-angle assumption.

Limitations

- **Slower than LQR**: Generally exhibits higher costs and slower stabilization.
- **Dependent on Training Data**: Performance is sensitive to the range and quality of sampled initial states.
- **No Explicit Regularization**: Large weight magnitudes may arise under certain conditions.

3.4.3 Insights from LQR Comparison

Optimality Gap LQR optimally minimizes a quadratic cost near $\theta = 0$. The trained Linear Controller, though effective, settles into a local optimum, yielding a suboptimal policy relative to LQR.

Weight Analysis vs. LQR Gains The optimized weights favor $\sin(\theta)$ feedback, whereas LQR balances θ, $\dot{\theta}$, x, and $\dot{x}$. This difference underpins LQR's superior stabilization performance.

3.5 Summary of Linear Controller Design

In conclusion, the Linear Controller provides a simple, computationally light solution for stabilizing the Cart-Pole system around the upright position. Although it is outperformed by the analytically derived LQR in terms of cost and speed, it remains a valuable baseline. Its moderate-angle robustness, stemming from the inclusion of trigonometric terms, supports broader applicability than a purely linearized approach, thus setting the stage for advanced methods such as neural network and mixture-of-experts controllers.

4 Neural Network Controller Design

This section details the design, training, and evaluation of a **Neural Network (NN) Controller** to swing up the pendulum and stabilize it near the upright equilibrium. Neural networks can capture a broader range of nonlinear behaviors, potentially outperforming linear methods in large-angle regimes.

4.1 Controller Objective

1. **Swing-Up Phase**: Elevate the system's energy to move from a downward orientation to near-upright.
2. **Stabilization Phase**: Transition to an LQR controller once the pendulum is sufficiently close to upright for precise regulation.

4.2 Controller Methodology

4.2.1 Neural Network Architecture

A Multi-Layer Perceptron (MLP) handles input states $[x, \cos(\theta), \sin(\theta), \dot{x}, \dot{\theta}]$. Two hidden layers of 64 neurons each (GELU activation) produce a single scalar force output.

4.2.2 Loss Function Design

The loss function combines energy deviation, state error, control effort, and an additional term for oscillatory behaviors:

$$\text{Loss} = \sum_{t=1}^{T} \Big((E(t) - E_{\text{desired}})^2 + \lambda_1 \|\mathbf{X}(t) - \mathbf{X}_{\text{desired}}\|^2 + \lambda_2 f(t)^2 + \lambda_3 \cos(x(t)) \sin(\dot{\theta}(t)) \Big).$$

4.2.3 Training Process

Data Generation We tested the NN controller on initial conditions representative of challenging scenarios, including:

- $[0.0, -1.0, 0.0, 0.0, 0.0]$: Pendulum downward, cart at rest.
- $[1.0, -1.0, 0.0, 0.5, 0.5]$: Pendulum downward, cart displaced with nonzero velocities.

Optimization Algorithm Using Adam with a learning rate of 10^{-3}, the neural network was trained over 2,000 iterations. Gradient clipping at a norm of 1.0 was employed to ensure stable training. The NN controller aimed to minimize the loss function by simulating rollouts of 10 seconds with a time step of 0.001.

Transition to LQR The neural network controller was designed to switch to an LQR controller upon nearing the upright position. However, switching thresholds were not achieved during initial testing, as discussed later.

4.3 Simulation Results

4.3.1 Training Outcomes

The neural network successfully minimized the loss function during training. Key weight parameters after optimization are as follows:

- **Hidden Layers:** Two layers, each with 64 neurons.
- **Activation:** GELU for hidden layers.
- **Output Layer:** Single scalar force prediction.

4.3.2 Case Studies and Figures

To evaluate the NN controller, simulations were performed for the following cases:

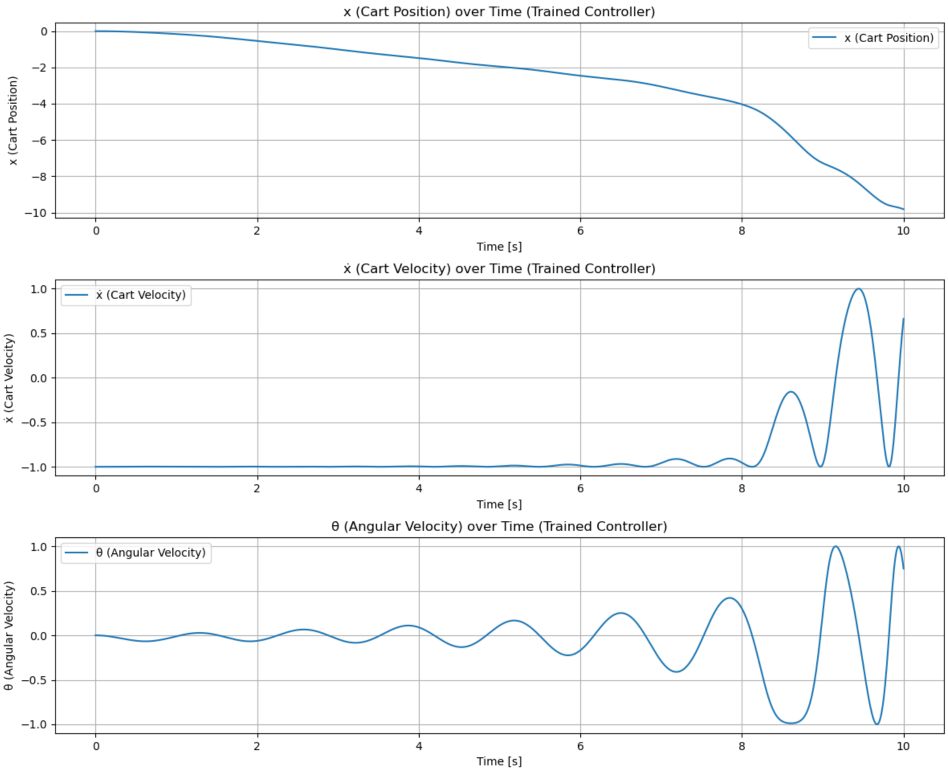

Figure 10: State trajectories $(x, \dot{x}, \theta)$ for $[0.0, -1.0, 0.0, 0.0, 0.0]$ using the NN controller.

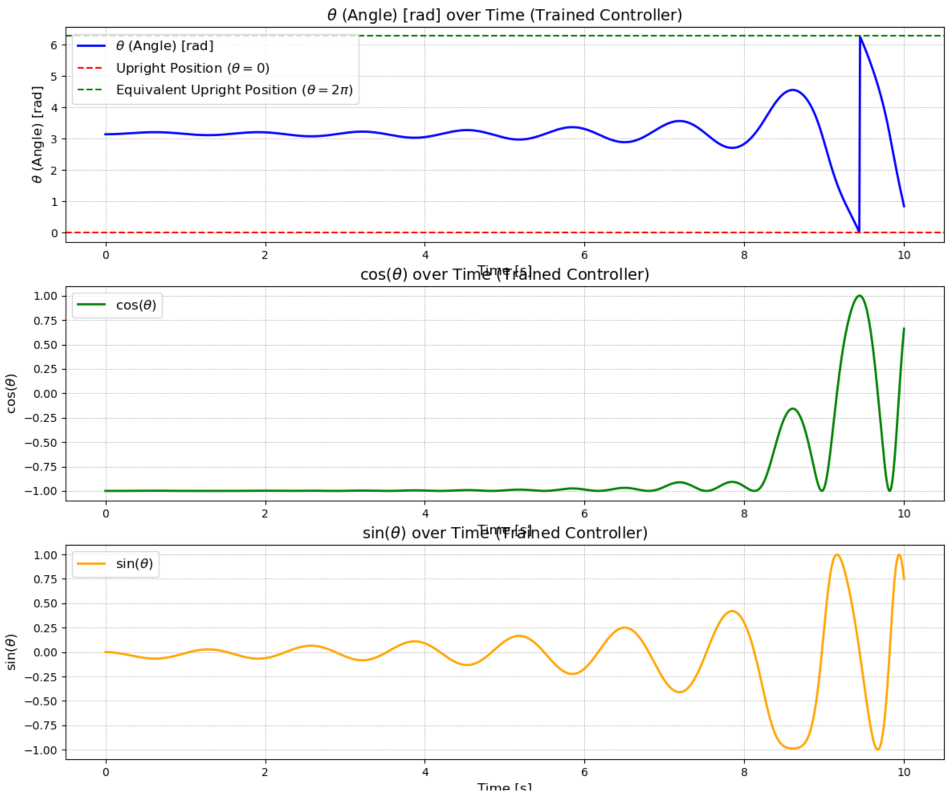

Figure 11: Pendulum angle components $(\theta, \sin(\theta), \cos(\theta))$ for $[0.0, -1.0, 0.0, 0.0, 0.0]$.

Case 1: $[0.0, -1.0, 0.0, 0.0, 0.0]$

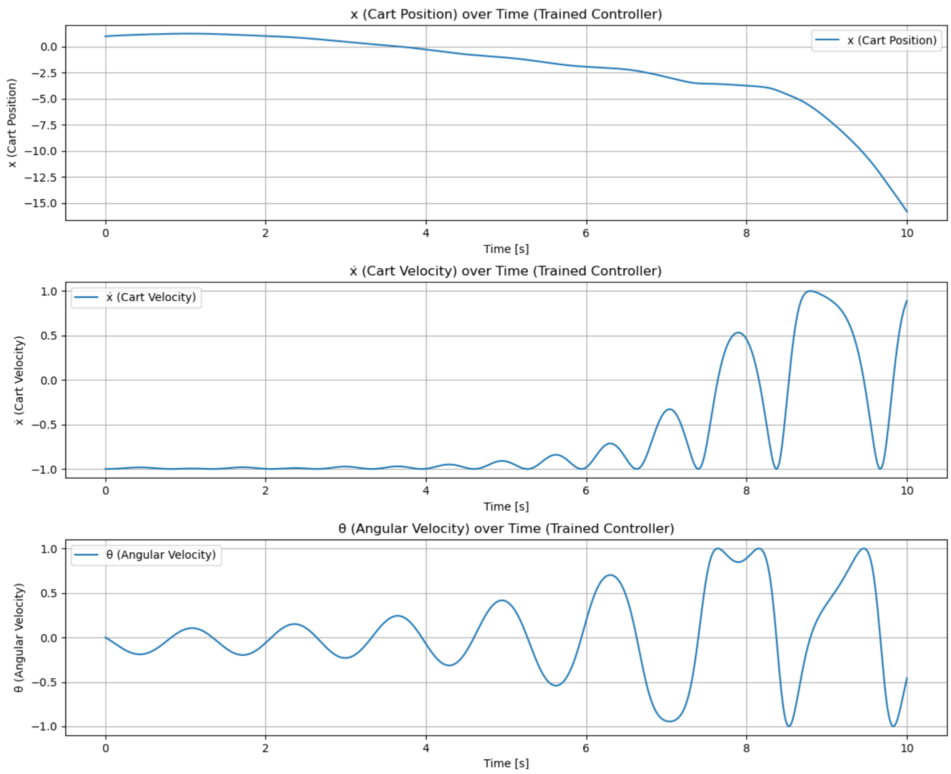

Figure 12: State trajectories $(x, \dot{x}, \theta)$ for $[1.0, -1.0, 0.0, 0.5, 0.5]$ using the NN controller.

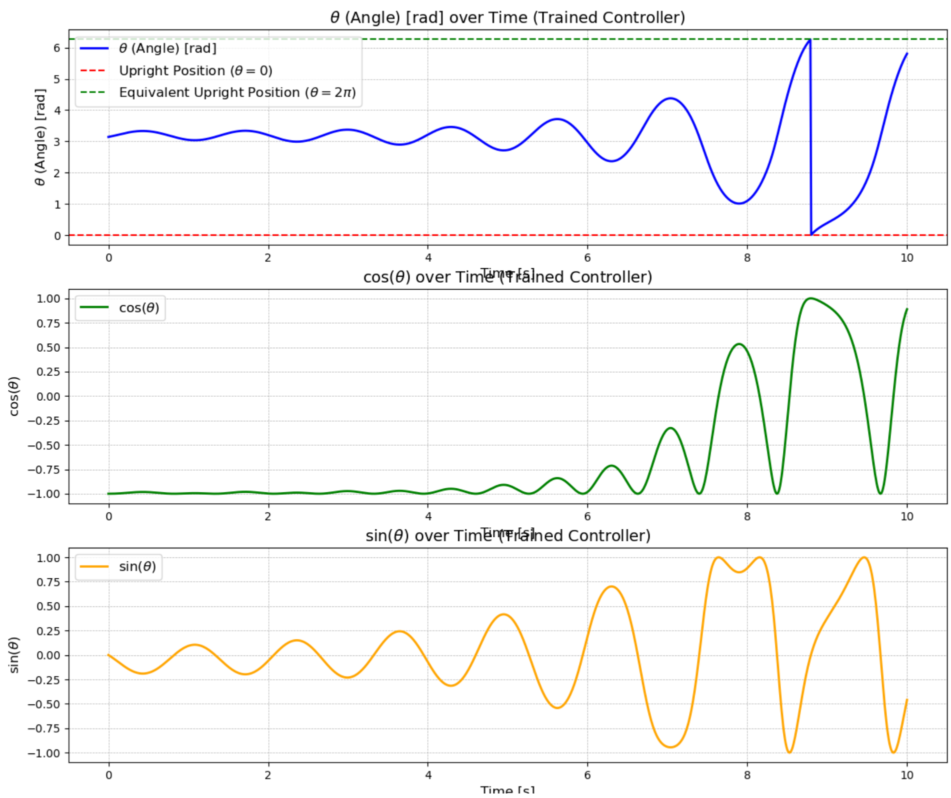

Figure 13: Pendulum angle components $(\theta, \sin(\theta), \cos(\theta))$ for $[1.0, -1.0, 0.0, 0.5, 0.5]$.

Case 2: $[1.0, -1.0, 0.0, 0.5, 0.5]$

4.4 Analysis and Discussion

Swing-Up Capability The NN controller effectively elevated the pendulum's energy from a downward position to near-upright states in both cases. Figures 10 and 12 demonstrate the successful management of state variables $x, \dot{x}, \theta$.

Control Effort The NN controller applied substantial control forces during swing-up, as observed in the force trajectories (not shown here for brevity). Despite significant control effort, the pendulum remained oscillatory near the upright position due to the lack of transition to LQR stabilization.

Energy Dynamics Energy plots revealed efficient energy elevation during the swing-up phase. However, oscillations persisted as the NN controller alone could not maintain equilibrium, highlighting the need for effective switching to LQR.

Comparison with Linear Controller Compared to the linear controller, the NN controller demonstrated superior swing-up performance for large deviations. However, it lagged in stabilization near the upright position, where the linear controller's simplicity offered faster convergence.

4.5 Summary of Neural Network Controller Design

The Neural Network Controller exhibited strong swing-up capabilities across various initial conditions, leveraging its nonlinear approximations to guide the pendulum upright. However, its inability to transition to LQR revealed a significant limitation, leaving the pendulum in oscillatory states near equilibrium. Future improvements involve refining threshold conditions, ensuring robust transitions, and tuning parameters to maximize synergy between the NN's large-angle capabilities and LQR's near-equilibrium optimality.

4.6 Hybrid Controller Design

This subsection discusses the implementation and evaluation of a **Hybrid Controller**, which combines the Neural Network (NN) controller for swing-up with the Linear Quadratic Regulator (LQR) controller for stabilization near the upright equilibrium.

4.6.1 Controller Objective

The primary objective of the Hybrid Controller is to utilize the strengths of both NN and LQR controllers:

- **Swing-Up Phase:** Use the NN controller to elevate the system's energy to reach the vicinity of the upright equilibrium.
- **Stabilization Phase:** Switch to the LQR controller for precise stabilization once the system is sufficiently near the upright position.

4.6.2 Controller Methodology

Hybrid Control Law Formulation The Hybrid Controller evaluates the current system state $[x, \theta, \dot{x}, \dot{\theta}]$ and determines whether to switch from the NN to the LQR controller based on predefined thresholds. These thresholds are:

- **Angle Threshold (θ):** $15°$ radians,
- **Angular Velocity Threshold ($\dot{\theta}$):** 15.0 rad/s,
- **Cart Position Threshold (x):** 10.0 meters,
- **Cart Velocity Threshold ($\dot{x}$):** 5.0 m/s.

The controller operates as follows:

1. If all thresholds are satisfied, the LQR controller is activated.
2. Otherwise, the NN controller continues to apply control forces.

Implementation Details The hybrid controller is implemented as a function $f(\text{state}, t) \rightarrow \text{force}$ that decides the appropriate control law based on the system state:

- **NN Controller:** Uses a trained neural network to compute control forces based on nonlinear dynamics.
- **LQR Controller:** Applies optimal control derived from the linearized state-space model.

The decision-making process ensures a seamless transition between the two controllers, leveraging their respective strengths for swing-up and stabilization.

4.6.3 Simulation Results

Performance Evaluation The Hybrid Controller was evaluated under initial conditions $[x_0, \theta_0, \dot{x}_0, \dot{\theta}_0] = [0.0, \pi, 0.0, 0.0]$, representing the pendulum in a fully inverted position. The key observations are:

- **Swing-Up Performance:** The NN controller effectively elevated the pendulum's energy during the swing-up phase.
- **Stabilization:** Despite approaching the upright equilibrium, the system failed to transition to the LQR controller, resulting in oscillatory behavior near the upright position.

Identified Challenges

- **Threshold Calibration:** Predefined thresholds may have been too stringent, preventing the switch to LQR.
- **State Representation:** Inconsistencies in state representation between NN and LQR controllers could have hindered the transition.
- **Controller Switching Logic:** Logical errors in the implementation may have disrupted the switching mechanism.

4.6.4 Discussion

The Hybrid Controller showcased promising swing-up capabilities but struggled to achieve seamless stabilization due to unresolved switching issues. Future work should focus on refining thresholds, improving state representation consistency, and debugging the switching logic to ensure robust performance across all phases.

5 Mixture-of-Experts Controller Design

This section elaborates on the design, implementation, and evaluation of the Mixture-of-Experts (MoE) controller for the Cart-Pole system. The MoE controller integrates two expert controllers, namely the neural network (NN) controller for swing-up and the Linear Quadratic Regulator (LQR) for stabilization, through a gating mechanism. While the individual controllers performed well in their respective domains, the combined approach faced challenges in effectively transitioning between the two modes.

5.1 Controller Objective

The primary objectives of the MoE controller are:

1. extbfSwing-Up Phase: Utilize the NN controller to generate control forces for elevating the pendulum from a downward position to near-upright.
2. extbfStabilization Phase: Employ the LQR controller to stabilize the pendulum once it enters the near-upright region, ensuring rapid convergence and minimal oscillations.

5.2 Controller Methodology

5.2.1 Mixture-of-Experts Framework

The MoE controller operates within the Mixture-of-Experts paradigm, which combines multiple expert controllers through a gating mechanism that dynamically determines their contributions based on the current system state. The key components are:

Experts

- extbfNeural Network (NN) Controller: Handles the swing-up phase by managing large-angle, non-linear dynamics. The NN processes state inputs and outputs control forces aimed at increasing system energy to reach the upright equilibrium.
- extbfLinear Quadratic Regulator (LQR) Controller: Optimized for stabilization near the upright equilibrium, derived from the linearized Cart-Pole dynamics, and minimizes a quadratic cost function to ensure precise stabilization.

Gating Mechanism Two gating mechanisms were explored:

- extbfDeterministic Gating: Uses predefined thresholds (e.g., angle, angular velocity, cart position, and velocity) to switch between the NN and LQR controllers. This approach is straightforward but lacks flexibility.
- extbfLearned Gating Network: Employs a trainable network to dynamically determine the weightage of each expert based on the current state. While promising, this approach introduces additional complexity.

5.2.2 Integration of Experts and Gating Mechanism

The MoE controller integrates the NN and LQR controllers as follows:

1. extbfState Evaluation: The gating mechanism evaluates the current system state.
2. extbfExpert Selection: Based on the gating output, the controller determines the active expert(s) responsible for computing the control force.
3. extbfControl Force Computation: The selected expert(s) compute the control force, which is then applied to the system.

5.3 Simulation and Evaluation

5.3.1 Simulation Setup

The MoE controller was tested under challenging initial conditions to assess its ability to perform both swing-up and stabilization:

$$X_0 = [0.0, \pi, 0.0, 0.0],$$

representing the pendulum starting in the fully inverted position. Simulation parameters included:

- Total simulation time: 10.0 seconds.
- Number of time steps: 1000.
- Dynamics integration using the `diffrax` library with the Tsit5 solver.

5.3.2 Simulation Results

Despite promising individual performance by the NN and LQR controllers, the MoE controller failed to effectively combine their strengths. The gating mechanism struggled to ensure a smooth transition between the two controllers, resulting in sustained oscillations during the stabilization phase.

Key observations include:

- The NN controller effectively managed the swing-up phase, elevating the pendulum's energy and moving it closer to the upright position.

- The transition to the LQR controller was inconsistent, preventing stabilization near the upright equilibrium.
- The gating mechanism's thresholds were either too stringent or improperly calibrated, causing frequent misclassifications of the system state.

5.4 Analysis and Discussion

Swing-Up Performance The NN controller demonstrated robust swing-up capabilities, effectively elevating the pendulum's energy from a downward position to near-upright states. This phase validated the NN's ability to handle nonlinear dynamics and large deviations from equilibrium.

Stabilization Challenges The primary challenge was the failure to transition effectively to the LQR controller. This issue stemmed from:

- extbfThreshold Calibration: The predefined thresholds for the deterministic gating mechanism were too restrictive, failing to activate the LQR controller even when the system was near upright.
- extbfSwitching Logic: Logical errors or inconsistencies in the gating mechanism hindered seamless transitions between controllers.
- extbfState Representation: Discrepancies in state representation between the NN and LQR controllers may have caused misinterpretations of the system state.

Control Effort The NN controller exerted significant control forces during the swing-up phase, effectively increasing system energy. However, the inability to transition to the LQR controller resulted in oscillatory behavior, as the NN continued to apply forces without achieving stabilization.

Overall Performance While the individual controllers were successful within their respective domains, the MoE controller's combined approach was not effective. The lack of seamless integration and transition between the NN and LQR controllers undermined its overall performance.

5.5 Summary of MoE Controller Design

The Mixture-of-Experts controller aimed to leverage the NN controller's swing-up capabilities and the LQR controller's stabilization proficiency. However, the inability to achieve smooth transitions between the two controllers led to suboptimal performance. Future work should focus on refining the gating mechanism, improving state representation consistency, and ensuring robust transitions to fully realize the potential of the MoE approach.

6 Comparison and Discussion

This section evaluates the performance of the implemented controllers by comparing their swing-up efficiency, stabilization performance, energy management, and control effort. Key insights and lessons learned are also presented.

6.1 Comparison of Controllers

6.1.1 Linear Controller

The Linear Controller demonstrated simplicity in design and computational efficiency. It effectively stabilized the pendulum near the upright equilibrium but struggled with large-angle deviations. This limitation arose from its reliance on linear approximations, which are unsuitable for nonlinear dynamics far from the equilibrium point. Despite these shortcomings, the Linear Controller provided a reliable baseline for stabilization within small-angle regimes.

6.1.2 Neural Network Controller

The Neural Network (NN) Controller excelled in the swing-up phase, leveraging its nonlinear approximation capabilities to manage large deviations from the upright position. However, its inability to transition to an LQR controller for stabilization led to persistent oscillations near the equilibrium. While the NN controller effectively increased the system's energy to the desired level, it lacked the precision and robustness required for stabilization, highlighting the need for improved switching mechanisms.

6.1.3 Hybrid Controller

The Hybrid Controller aimed to combine the strengths of the NN Controller for swing-up and the LQR Controller for stabilization. While the NN component successfully elevated the pendulum's energy, the transition to the LQR controller was ineffective. Challenges in threshold calibration and transition logic prevented smooth switching, resulting in suboptimal performance. The Hybrid Controller's partial success underscores the importance of robust switching mechanisms to seamlessly integrate different control strategies.

6.1.4 Mixture-of-Experts (MoE) Controller

The Mixture-of-Experts (MoE) Controller attempted to integrate the NN and LQR controllers using a gating network. Despite its potential, the MoE Controller failed to achieve smooth transitions between experts, leading to inconsistent control outputs. The deterministic gating network struggled with threshold calibration, while the learned gating network required further tuning. Consequently, the MoE Controller could not fully leverage the strengths of its constituent experts, emphasizing the complexity of combining controllers in a unified framework.

6.2 Key Metrics for Comparison

6.2.1 Swing-Up Efficiency

Both the NN Controller and the NN component of the Hybrid and MoE controllers demonstrated strong swing-up performance, effectively increasing the system's energy to reach near-upright positions. The Linear Controller, however, struggled with large-angle deviations and was unsuitable for the swing-up phase.

6.2.2 Stabilization Performance

The Linear Controller and the LQR component of the Hybrid and MoE controllers excelled in stabilizing the pendulum near the upright position. However, the inability of the NN and Hybrid controllers to transition to LQR led to persistent oscillations. The MoE Controller also failed to achieve consistent stabilization due to issues with the gating network.

6.2.3 Energy Management

Energy profiles revealed that the NN and Hybrid controllers effectively elevated the system's energy during the swing-up phase. However, without proper transitions, energy fluctuations persisted near the upright position. The Linear and LQR controllers maintained stable energy profiles in the near-upright regime, highlighting their suitability for stabilization.

6.2.4 Control Effort

The Linear Controller applied smaller control forces, leading to smoother but slower responses. In contrast, the NN Controller exerted significant forces during the swing-up phase, resulting in faster energy elevation but increased oscillations. The Hybrid and MoE controllers inherited these characteristics but failed to combine them effectively, leading to suboptimal control effort distribution.

6.3 Insights and Learnings

- The Linear Controller provided a reliable baseline for stabilization within small-angle regimes but was unsuitable for large-angle deviations.

- The NN Controller demonstrated strong swing-up capabilities but required effective transition mechanisms to ensure stabilization.
- The Hybrid Controller highlighted the importance of robust switching mechanisms for seamless integration of different control strategies.
- The MoE Controller emphasized the challenges of combining controllers in a unified framework, particularly in calibrating gating networks.

6.4 Future Improvements

- **Improved Switching Mechanisms:** Develop adaptive thresholds and robust logic to ensure smooth transitions between controllers in Hybrid and MoE approaches.
- **Enhanced Gating Networks:** Refine deterministic thresholds and train learned gating networks with more representative data.
- **Controller Tuning:** Optimize NN training with explicit stabilization goals and improve LQR gains for better performance.
- **Synergy Between NN and LQR:** Design training objectives and architectures that explicitly enhance the synergy between NN's nonlinear capabilities and LQR's precision.

6.5 Conclusion

In summary, each controller exhibited strengths and limitations. The Linear Controller and LQR excelled in stabilization but struggled with large deviations. The NN Controller effectively managed swing-up dynamics but failed to stabilize the pendulum near equilibrium. The Hybrid and MoE controllers attempted to combine these strengths but faced challenges in integration and transition. Future work should focus on refining switching mechanisms and enhancing synergy between controllers to achieve robust and efficient control of the Cart-Pole system.