Balancing an Inverted Pendulum using LQR and PD Control in MuJoCo

Dhriti Madireddy Vianca Varma Parth Bangla Abhignan Muppavaram

***Abstract*—Balancing an inverted pendulum is a classic control problem that tests the effectiveness of feedback control strategies. In this work, we explore and compare Linear Quadratic Regulator (LQR) and Proportional Derivative (PD) control strategies in stabilizing an inverted pendulum mounted on a robotic arm using the MuJoCo physics engine. We discuss challenges in implementing reinforcement learning (RL), the shift to classical control, and analyze results derived from simulated trials.**

***Index Terms*—LQR, Proportional Derivative, MuJoCo, Inverted Pendulum, Reinforcement Learning**

I. Introduction

The inverted pendulum problem represents a foundational benchmark in control systems due to its instability and non-linear dynamics. In the League 3 task of our project, we were tasked with stabilizing an inverted pendulum mounted at the end of a multi-jointed robotic arm. Our initial goal was to use Reinforcement Learning (RL) techniques, but due to technical difficulties and inconsistent results, we transitioned to classical control methods.

II. Method Explanation

To control the inverted pendulum mounted on a robotic arm, we implemented a hybrid control approach combining Linear Quadratic Regulator (LQR) and Proportional Derivative (PD) controllers. These controllers were assigned to different joints of the arm based on their influence on pendulum stability and required control characteristics.

A. Proportional Derivative (PD) Control

PD control was used to maintain the position of the shoulder and elbow joints. The objective was to keep these joints in their initial neutral position while allowing the base joint to actively correct the pendulum's balance. The proportional term (P) provides a restoring torque proportional to the error between the current joint position and the initial desired position. The derivative term (D) introduces a damping effect that reacts to the joint's angular velocity, reducing oscillations and improving stability. These gains were manually tuned through iterative testing to ensure that the upper segments of the arm remained stable without adding unnecessary noise to the system.

This method acts as a soft constraint mechanism to prevent the shoulder and elbow from deviating significantly, which might otherwise introduce noise and instability to the pendulum dynamics. PD control is particularly effective for fine-tuning joint responses without demanding heavy computation or complex optimization.

Identify applicable funding agency here. If none, delete this.

B. Linear Quadratic Regulator (LQR) Control

The base joint, being the most influential in stabilizing the pendulum, was controlled using an LQR. The LQR controller is derived by defining a state-space representation of the system and solving the Riccati equation to find an optimal feedback gain matrix K. The control signal is given by the linear law:

$$u = -Kx$$

where x is the state vector containing the pendulum's angle and angular velocity. The matrices Q and R were used to weight the state errors and control effort, respectively. Higher weights in Q penalize deviations from the upright position, while the weights in R penalize excessive torque application.

We experimented with various Q and R configurations to balance responsiveness and energy efficiency. Adding damping through an extra term in the control law such as $-0.1\dot{\theta}$ helped reduce overshooting and promote stability.

C. Environment Setup and Integration

The MuJoCo simulation was interfaced using a custom Gymnasium wrapper, where the control logic was executed in each simulation step. State observations were extracted from joint angles and velocities. The control torques were computed and applied at every step, followed by updating the simulation.

For the constrained configuration, we added realistic limits on joint torque, joint angles, and angular velocities to better reflect physical feasibility. This ensured the simulated system adhered to hardware-like behavior. Additionally, we implemented noise and damping in the physics to further model real-world variability.

Through the combination of these methods, we designed a controller that uses the base for active balancing via optimal feedback (LQR) and stabilizes the remaining joints passively via damped proportional feedback (PD). This separation of concerns allowed us to simplify control design while preserving effectiveness in dynamic stabilization tasks.

III. Results

We evaluated three different configurations of control methods in the MuJoCo simulation environment:

- LQR + PD (Unconstrained)
- LQR + PD (Constrained)
- PPO (Reinforcement Learning Agent)

The table below summarizes the performance of each setup across multiple runs:

TABLE I
Performance Comparison of Control Configurations

Config	Avg Count	Max Count	Torque Limited	Velocity Clamped	Joints Clamped
1	30	113	No	No	No
2	50	72	Yes	Yes	Yes
3	2	4	Yes	No	No

A. LQR + PD (Unconstrained)

This configuration performed decently with an average balance count of 30 and a maximum count of 113. However, the absence of torque, velocity, and joint limits enabled the robotic arm's base joint to spin indefinitely in an unrealistic manner. This behavior created centrifugal force to hold the pendulum upright, which is physically infeasible. Tuning the Linear Quadratic Regulator and Proportional Derivative parameters required several hours of manual trial and error. Despite identical parameter settings, results varied across runs due to sensitivity in the physics engine. The major drawback of this method was that it bypassed the real challenge of controlled stabilization by relying on unbounded rotation that would be unachievable in real-world robotic systems. Thus, while the numbers appeared promising on paper, they were not reflective of practical success.

B. LQR + PD (Constrained)

By introducing realistic physical constraints including torque caps, joint angle limits, and velocity clamping, the performance and realism of the simulation improved significantly. The average balance count rose to 50 with a maximum of 72. The robot now responded by gently correcting the pendulum's motion instead of relying on brute-force spinning. The torque applied by the controller remained within allowable ranges, making the system not only more stable but also more efficient. In multiple trials, this constrained setup showed far less variance in outcome compared to its unconstrained counterpart. This configuration modeled hardware-specific limitations, making it a more appropriate solution for deployment in actual robotic hardware. Furthermore, the trade-off between speed and feasibility was acceptable, as the system became more interpretable and safe for realistic control tasks.

C. PPO (Reinforcement Learning Agent)

The reinforcement learning model consistently underperformed, achieving an average balance count of 2 and a maximum of 4. Setting up the Gymnasium wrapper for MuJoCo was challenging, and implementing a reward function that led to effective learning was even harder. Despite many iterations and significant training time, the agent frequently failed to generalize and often froze or overreacted, leading to the pendulum falling. Training times for each model were significantly longer than the classical controllers, and even slight deviations in reward shaping resulted in diverging behaviors. The agent struggled to find a balance between overcorrection and underreaction, often destabilizing the pendulum within a few simulation steps. These limitations revealed the gap between theoretical potential and practical success in reinforcement learning when domain knowledge is lacking.

IV. Analysis

The comparison highlights both the strengths and limitations of model-based and learning-based approaches. Classical control methods demonstrated predictable, tunable behavior and provided a clear understanding of the system's responses. Although the unconstrained configuration produced higher maximum counts, the absence of realistic constraints made the results physically inaccurate and unsuitable for real-world implementation. Conversely, the constrained controller, while slower to respond and slightly less effective in peak balance counts, delivered consistent and physically plausible behavior.

The constrained Linear Quadratic Regulator and Proportional Derivative setup revealed a compelling balance between control effort and performance. It showed that physically grounded parameters, such as capped joint angles and torque limits, can lead to smoother correction behavior and more sustainable performance across trials. This configuration also showed better resilience to the simulation noise and floating-point variations inherent in MuJoCo.

Working with the reinforcement learning agent illuminated the complexity of deploying such methods in practice. Setting up the environment from scratch, configuring observations and actions, and designing a reward function required deep technical understanding. Training was time-consuming and results were highly variable, with limited interpretability. Without proper intuition into the agent's learning process, progress felt opaque and frustrating. It became evident that reinforcement learning introduces a high barrier to entry due to its dependence on hyperparameter tuning, reward shaping, and environment configuration.

Moreover, nondeterministic simulation factors such as noise, floating-point errors, and threading inconsistencies led to different outcomes for identical initial conditions. This emphasizes the need for robustness and repeatability in control systems, especially when translating simulations into real hardware applications. We observed that even in the best constrained setup, small perturbations in initial velocity or simulation time step could compound over time, highlighting the delicate nature of such dynamical systems.

In conclusion, while reinforcement learning holds potential for more adaptive behavior in the long term, our experiments show that classical control strategies like Linear Quadratic Regulator and Proportional Derivative—when implemented with proper physical constraints—offer a more stable, explainable, and practically applicable solution for inverted pendulum stabilization in robotics. Future efforts may integrate reinforcement learning as a fine-tuning layer built on top of these reliable foundations, provided improvements in reward design, training reliability, and system interpretability.

Acknowledgment

The authors would like to thank Professor Donghyun Kim for guidance throughout the project and the CS403 teaching staff for their support and feedback. We also acknowledge the open-source tools and documentation that facilitated our use of MuJoCo and reinforcement learning frameworks.

V. Related Works

The inverted pendulum has been a significant research topic in control theory due to its complex dynamics and inherent instability, making it a valuable platform for evaluating various control strategies. Traditional methods such as PID controllers, praised for their simplicity and real-time applicability, have been extensively studied. Ang *et al.* [1] provided a comprehensive analysis of PID controllers, highlighting their robustness and ease of implementation. Nevertheless, PID controllers face challenges when dealing with nonlinearities and external disturbances, limiting their effectiveness in more demanding applications such as the inverted pendulum.

Advanced control methodologies, particularly Linear Quadratic Regulator (LQR) approaches, have shown considerable improvements by systematically minimizing predefined cost functions that balance stability and control efforts. Wan *et al.* [2] demonstrated the superior capability of LQR controllers to maintain stable and precise control in dynamic environments. Further research by Chacko and Abraham [3] expanded on LQR methodologies by focusing on optimal selection of weighting matrices, significantly enhancing system response characteristics such as overshoot, response time, and overall stability.

Recent advancements have explored hybrid methods combining traditional and modern control strategies, integrating the simplicity of classical controllers with the mathematical rigor of LQR. Such hybrid methods frequently incorporate gravity compensation, improving robustness and precision by explicitly addressing gravitational torque disturbances. These studies underscore the potential of integrated approaches to substantially improve the effectiveness of inverted pendulum stabilization systems.

Our work builds on these insights, integrating PD and LQR controllers with gravity compensation. This hybrid approach aims to leverage the strengths of each control methodology while addressing identified limitations in existing research. Thus, our study contributes to ongoing efforts by providing a thorough comparative analysis and demonstrating the clear advantages of combining classical and optimal control techniques.

References

[1] K. H. Ang, G. Chong, and Y. Li, “PID control system analysis, design, and technology,” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[2] T. Wan, Y. Li, and H. Wang, “Design of LQR Controller for the Inverted Pendulum,” *Advanced Materials Research*, vol. 926–930, pp. 1909–1912, 2014.

[3] P. Chacko and A. Abraham, “On LQR controller design for inverted pendulum stabilization,” *International Journal of Dynamics and Control*, 2023. [Online]. Available: ResearchGate