Reinforcement Learning for Dynamic Stability

Alli Ajagbe, Noyonica Chatterjee, Surabhi Tannu

Introduction and Related Work

In the realm of robotics, achieving dynamic stability is an important objective, important for the seamless operation of various robotic systems of diverse applications. From industrial settings employing harbour cranes in transportation processes to the domain of autonomous robots with inverted pendulum dynamics navigating unpredictable environments, the ability to adapt by itself and maintain balance in the face of disturbances is of utmost importance. Our project aims to empirically compare different reinforcement learning algorithms (policy, model-free, model-based) to test dynamic stability, mainly in the cartpole problem.

Drawing inspiration from the cartpole problem, we can extrapolate real-life implications such as in the case of harbor cranes, and segway personal transporters, which exhibit dynamics similar to the cartpole system. Through the study and optimization of dynamic stability in cart pole systems using RL algorithms, we have the potential to significantly improve the stability and efficiency of such operations.

Prior research in this field has laid the essential groundwork, shedding light on optimization techniques, trajectory stabilization, and the application of RL algorithms for stability control. Works such as that of Posa, Kuindersma, and Tedrake (2016), exploring optimization and stabilization of trajectories for constrained dynamical systems, alongside studies by Xi and Chen (2020) on stability control of biped robots using hybrid RL methods, provide valuable insights into the complexities and potential solutions in achieving dynamic stability. Moreover, contributions by Lucian et al. (2018) delve into the performance, stability, and applicability of RL-based control systems, further enriching our understanding of this domain.

Problem and Proposed Work

The cart pole system is a classic problem in control theory and reinforcement learning. It consists of a cart that can move along a frictionless track, with a pole attached to the cart via a hinge. The goal is to balance the pole upright by moving the cart left or right.

Our project is motivated by the imperative need to find a robust RL algorithm capable of ensuring dynamic stability in diverse robotic systems with inverted pendulum mechanisms. We are comparing and analyzing various algorithms across different types, to find the one that most suits the dynamic stability problem, with a particular focus on the cartpole system. To address this challenge comprehensively, our methodology encompasses a multifaceted approach. Initially, we focus on a literature review, looking into existing research and methodologies pertaining to dynamic stabilization and RL algorithms. This foundational exploration provides us with necessary insights, based on which we can take our subsequent steps.

Further, we transition to the implementation phase, where we compare and analyse existing RL algorithms based on predefined metrics, within the Gym Environment provided by OpenAI. Leveraging a diverse array of RL approaches, including policy-based, model-free and model-based methods, we aim to explore the efficacy of various strategies in achieving dynamic stability. Through iterative experimentation, we endeavour to identify the most effective algorithms for our intended application scenario.

Pertinent to why an empirical analysis is of importance in our project; comparative analysis helps in understanding the strengths and weaknesses of RL approaches in dynamic stability problems. A comparison will also help us understand the algorithm's generalization - the ability of the RL algorithm to apply learned policies to unseen situations - and its transferability - how well knowledge learned in one setting can be applied to another. Evaluating

generalization and transferability assesses RL algorithms' ability to adapt and perform well. All these are crucial when selecting the most appropriate approach for a given dynamic stability problem.

Additionally, performing empirical analysis will give us a deep understanding of the dynamic systems we are working with, in our case the cartpole. This includes understanding the dynamics of the environment/system we are trying to stabilize, including its state space, transitions, and control inputs. This will also help us understand and implement RL algorithms, especially in real-life scenarios. Further, we will be defining evaluation metrics to measure convergence speed, stability, and adaptability, which will aid in the comparison of our chosen algorithms.

We have chosen to focus on the implementation of algorithms for the cartpole balancing problem based on its various real-life extrapolations extending beyond the fields of robotics. Some of these are:

- Inverted Pendulum: Many real-world applications involve stabilizing inverted pendulums, which are essentially variations of the cartpole system. Some examples are bipedal robots, stabilizing cranes, etc.
- Financial Markets: The dynamics of financial markets, with assets fluctuating constantly, can sometimes be conceptualized as a balancing act similar to the cartpole system. Traders and investors employ various strategies to maintain stability and achieve desirable outcomes.

Proposed Evaluation

In our proposed system we have defined the following parameters for evaluation.

Convergence Speed: This metric refers to how quickly our RL algorithm converges towards an optimal or near-optimal policy within the given number of episodes. We will analyze the rate at which the algorithms improve their performance over the training episodes, measuring the speed of learning and adaptation.

Stability: Stability in this context relates to the ability of an RL algorithm to consistently maintain balance and prevent the cartpole system from falling over throughout the training process. This will be a measure of deviation from the starting point from the beginning to the end of the experiment.

Average Cumulative Reward: Cumulative reward represents the total sum of rewards obtained by the RL agent in an episode with a maximum possible value of 500 amongst the 10000 runs, in the cartpole environment. A higher cumulative reward indicates that the RL agent is effectively balancing the pole.

References

1. Posa, M., Kuindersma, S., & Tedrake, R. (2016). Optimizations and stability analysis for locomotion planning and control. The International Journal of Robotics Research, 35(7), 1167-1181.

2. Xi, W., & Chen, X. (2020). A novel biped robot dynamic balance control method based on deep reinforcement learning. Robotics and Computer-Integrated Manufacturing, 63, 101896.

3. Lucian, M., et al. (2018). Performance of reinforcement learning in the design of control systems. Journal of Control Engineering and Applied Informatics, 20(2), 60-69.

4. Han, B.-C., Kim, H.-C., & Kang, M.-J. (2023). Comparison of value-based Reinforcement Learning Algorithms in Cart-Pole Environment. *International Journal of Internet Broadcasting and Communication*, 15(3), 166-175. http://dx.doi.org/10.7236/IJIBC.2023.15.3.166

Timeline (Tasks with green color have been completed)

Week	Task 1	Task 2	Task 3

1-2	Identifying dynamic stabilization algorithms: Q-learning, PPO, and others from credible sources.	Conducting literature review on RL modules for stabilization across existing databases.	Identifying an illustrative scenario for simulation - CartPole from OpenAI Gym.
3-4	Setting up a development environment with required tools and libraries for RL and initiating algorithm coding.	Exploring the implementation of advanced algorithms, comparing theoretical and practical advantages.	Employing OpenAI Gym for basic stabilization simulations and refining algorithms for optimal performance.
5-6	Defining evaluation metrics including convergence speed, stability, and adaptability for fair comparison.	Employing Weights & Biases (Wandb) for experiment tracking, model performance comparison, and initial result logging.	Comparing algorithm performances based on predefined metrics for stability, efficiency, and adaptability.
7-8	Optimizing models according to testing feedback.	Leveraging Wandb for tracking experiments and identifying top models.	-
9-10	Conduct final evaluations, comparing models against predefined metrics and documenting the efficiency of different algorithms.	Create a comprehensive report outlining the methodology, results, and insights derived from studying dynamic stability in systems similar to CartPole Dynamics.	-

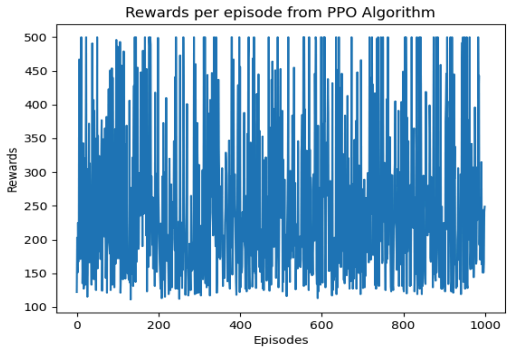
Work Done So Far

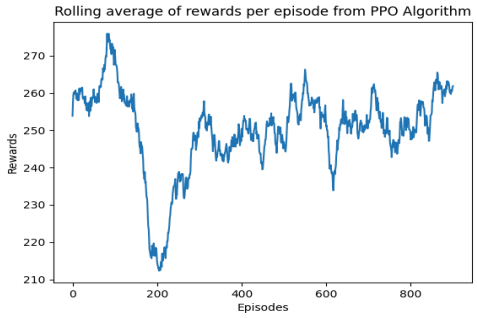
Literature Review: We have conducted an extensive literature review, gathering insights from about 14 relevant studies. For our literature review, we maintained an Excel sheet to document our findings and gain a comprehensive understanding of the methodologies employed, the scale at which the experiments were conducted, and possible adaptations for our project. In our Excel, we maintain 7 columns namely - Title, Author, Methodology Employed, Points for our project (1,2,3), and Link.

Title	Author (s)	Methodology Employed	Points for Our Project	Points for Our Project 2	Points for Our Project 3	Link
			Idea 1	Idea 2	Idea 3	
Optimization and Stabilization of Trajectories for Constrained Dynamical Systems	M. Posa, S. Kuindersma, and R. Tedrake	The methodology in this literature employs a combination of trajectory optimization and control techniques to synthesize and stabilize complex trajectories for robots subject to contact constraints. Specifically, the authors introduce a trajectory optimization algorithm called DIRCON, which extends the direct collocation method to naturally incorporate manifold constraints. They then adapt the classical time-varying linear quadratic regulator to produce a local cost-to-go in the manifold tangent plane. Finally, they utilize quadratic programming to descend the cost-to-go while incorporating unilateral friction and torque constraints.	Manifold constraints to define feasible states, ensuring stable motions despite contact constraints.	Trajectory optimization techniques like DIRCON to generate dynamic stability trajectories	Quadratic programming to optimize trajectories	https://agile.seas.harvard.edu/publications/optimization-and-stabilization-trajectories-constrained-dynamical-systems

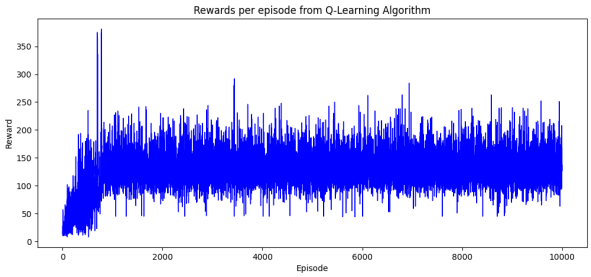
Setup: Leveraging the existing Gymnasium library, we have set up our experimentation framework and simulation environment that models the dynamics of the cartpole system. We plan on using this environment to evaluate the performance of the RL algorithms as it allows for consistent and approximately reproducible experimentation under controlled conditions with the right amount of stochasticity.

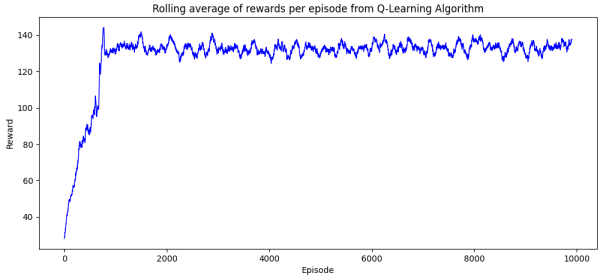
PPO Algorithm Implementation: We have successfully implemented the Proximal Policy Optimization algorithm - a policy gradient method following the Actor-Critic framework. We have used 10000 as the maximum number of training episodes and 1000 for testing and evaluation. Following the completion of the training and testing phases, our implementation generates a plot of the rewards per episode, and the rolling average of the rewards with a window size of 100.





Q-Learning Implementation: In addition to PPO, we have also implemented the Q-Learning algorithm for the cartpole environment with the plots in correspondence to that of the PPO algorithm. For comparison, we have also compared the training time for both algorithms. The time plot comparison can be found here.





More figures from our work done so far can be accessed via our GitHub repo.

Challenges and Limitations - Limited Computational Resource

Many research papers and related works we have reviewed utilised external graphics cards and cloud computing resources from AWS and Google Cloud to run experiments spanning over 20000 episodes. However, we are unable to avail of such a provision. One workaround we identified is leveraging Kaggle's provision of the T4 and the P100 GPUs (even though we are still constrained by the weekly limit of 30 hours). Due to the finite nature of this and the demand from other users (for T4), we decided to use a benchmark of 10000 episodes for all our RL algorithms. This limitation might not allow us to capture the full nuances and variations of the learning process from all the algorithms - especially those that require hyperparameter tuning like DQN. While our results might not be fully robust for generalisation, we believe we can still provide meaningful empirical evaluations of the RL algorithms that are not far-fetched from the results we would have gotten otherwise.

Next Steps

Building upon our previous work of algorithm selection, literature review and implementation setup, our immediate next steps involve model development for the other algorithms and preliminary testing. Succinctly, the algorithms to be implemented post-midsem are Deep Q-Learning, Q-Learning with Prioritized Experience Replay (QPER), and SARSA. This catalogue helps us cut through the identified RL types from policy-based as in the case of PPO, model-free as for Q-Learning and model-based as for SARSA. Leveraging Weights & Biases (Wandb) for experiment tracking and hyperparameter tuning for the deep learning models, we'll log results and compare algorithm performances based on predefined metrics for stability, efficiency, and adaptability. Finally, we will document our findings,