Optimisation of Kick Latency for Enhanced Performance of Robots in the RoboCup Three-Dimensional League through Proximal Policy Optimisation (PPO)

University of the Witwatersrand

School of Computer Science & Applied Mathematics

Humbulani Colbert Nekhumbe
2340639

Supervised by Dr Pravesh Ranchod

July 25, 2024

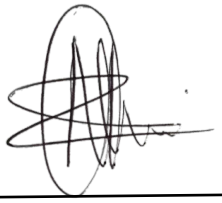
A dissertation issued to the University of the Witwatersrand's faculty of science in Johannesburg, as a partial satisfaction of the prerequisites for obtaining a Masters of Science degree.

Abstract

This study aimed to enhance the kicking ability of Nao robots in the three-dimensional RoboCup simulation by addressing a crucial challenge observed in the University of Witwatersrand RoboCup team. The focal challenge revolved around a noticeable delay and slow movement manifested by the robot during ball kicks, leading to vulnerabilities in ball possession against opposing teams. To surmount this challenge, the implementation of Proximal Policy Optimisation (PPO), a methodology pioneered by OpenAI, was advocated. The precise objective was to optimise kick parameters, with a primary emphasis on curtailing kick latency. This optimisation aimed to ensure swift and accurate execution across various kicking scenarios, encompassing actions like propelling the ball into the opponent's territory to bolster ball possession and thwart adversary manoeuvres. Harnessing the iterative advancements embedded in PPO, the successor to Trust Region Policy Optimisation (TRPO), the endeavour was to refine the kicking behaviour of Nao robots. This optimisation process significantly reduced the observed kick delay, and this made the robot more agile and effective at competing in the complex three-dimensional RoboCup simulation environment. The study's outcomes highlighted substantial progress in reducing kick latency and improving the adaptability of robotic soccer players, opening up possibilities for further exploration in reinforcement learning for autonomous agents.

Declaration

I, Humbulani Colbert Nekhumbe, affirm that I am the sole author of this research study which is being presented for MSc in Robotics at the esteemed University of the Witwatersrand. I confirm that this study has not been previously presented to any other academic institution for consideration for another academic qualification.



02/07/2024

Acknowledgements

I would like to express my deepest gratitude to God, whose unwavering guidance and blessings have been my source of strength throughout this academic journey. I am truly thankful for the wisdom and inspiration provided during moments of challenge.

I extend my sincere appreciation to my supervisor, Pravesh Ranchod, for his invaluable guidance, support and mentorship. His expertise and commitment have been instrumental in shaping the direction of my research and I am grateful for the opportunity to learn under his guidance.

A special thanks to the Wits RoboCup team, particularly Michael Beukman and Branden Ingram, for their generous assistance and patience. Their willingness to share their knowledge, walk me through the intricacies of the Wits RoboCup code-base and provide valuable insights has been crucial to the success of this project.

I am deeply indebted to my fiancee, Ronewa Tshitetete, for her unwavering encouragement and constant support. Her belief in my abilities and her motivation played a pivotal role in keeping me focused and determined throughout this endeavour. I am fortunate to have her by my side. Lastly, I want to express my gratitude to my newborn daughter, Oritonda Aliana Nekhumbe. The prospect of becoming a father has been a powerful motivator, urging me to work diligently and complete this study as she arrive to this world. I look forward to the joy of parenthood and the opportunity to be a present and devoted father to her.

To everyone who has contributed to this journey, thank you for being an integral part of my academic and personal growth.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The RoboCup is a global initiative that fosters research in robotics and artificial intelligence through robot football competitions. It is an extremely significant platform for improving robotic system capabilities in dynamic and competitive scenarios [da Silva 2019]. This study focuses on the optimisation of kicking behaviour in RoboCup events, with a focus on the difficulty of kick delays.

The impetus for this research originates from close observations of the University of Witwatersrand RoboCup squad during matches. The main difficulty is a robot's visible latency and sluggish movement during the kicking phase, which leads to ball possession being lost to the opposition. This issue has a direct impact on the robots' overall performance on the field. It has an impact on the team's ability to score goals and, eventually, win games. As a result, the purpose of this study is to improve the overall performance of robotic football players by reducing the latency in their kicking behaviour.

Existing research, such as Depinet *et al.* [2014], highlights the complexity of kicking in the three-dimensional RoboCup simulation league. The difficulties include providing smooth transitions between walks, achieving precision requirements, and coordinating several joints and keyframes for effective kicking. Machine learning techniques have emerged as effective solutions for solving these difficulties. In this context, kick latency is identified as a crucial component affecting robot performance.

This study investigated specific strategies and approaches, with a primary focus on lowering kick latency through the use of machine learning technologies, including Proximal Policy Optimisation (PPO). By improving the kicking characteristics, this study will considerably improve robot performance in RoboCup competitions. The final goal was to enable these robots to compete more effectively in the complex and challenging environment of the three-dimensional RoboCup simulation by resolving the observed kick latency and enhancing overall kicking behaviour.

The outcome of this study demonstrates the efficiency of the Proximal Policy Optimisation (PPO) algorithm in increasing the kicking performance of robotic agents in the RoboCup simulation. The latency figure 5.7 shows that integrating rewards customised to reduce kick latency resulted in a significant and consistent drop in the time it takes

robots to execute kicks. This positive change represents faster responses to kicking activities. Additionally, the increasing trend of the Max Distance Reward demonstrated that robotic agents had improved abilities to kick farther distances. These strong data support the study's strategy, demonstrating that robots not only kicked with lower latency but also with greater effectiveness. This represents a big step forward in building adept decision-making capabilities for robots in dynamic contexts, adding to the overarching goal of enhancing robotic systems in competitive scenarios.

Chapter 2

Background

2.1 RoboCup

The focus of this study is applicable in the three-dimensional simulation RoboCup, where Nao robot agents compete with 11 agents per team and two teams per game [Depinet *et al.* 2014]. The robots are equal in size and approximately 0.5 metres tall. The playing field is approximately $30mx20m$, which is a scale model of the human soccer field. In recent years, the teams that won the RoboCup three-dimensional league won because of their fast and strong movements, which were not an easy challenge.

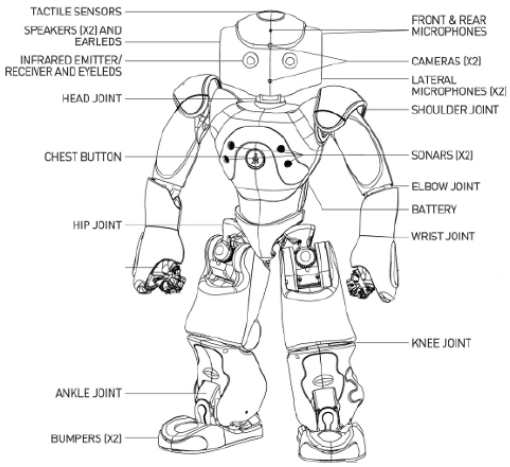


Figure 2.1: Physical NAO robot - [Teixeira *et al.* 2020]

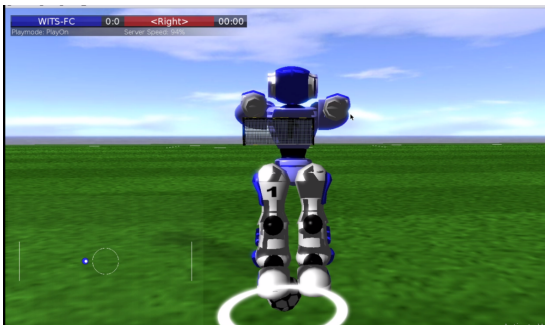


Figure 2.2: Virtual Nao robot

2.1.1 3D Soccer Simulation

The RoboCup 3D simulation league employs SimSpark as its physical multiagent simulator, founded on the Open Dynamics Engine library [Lau 2023]. This simulation league enhances realism by incorporating an additional dimension and more intricate physics, utilising SimSpark1 as the official RoboCup 3D simulation server. In the league's initial stages, a spherical agent model took precedence, and later advancements introduced humanoid models, starting with the Fujitsu HOAP-2 robot in 2006. This transition aimed to enhance low-level control of humanoid robots, emphasising essential behaviours such as walking, kicking, turning, and standing up [Reis 2023]. The comprehensive interaction of these components is illustrated in Figure 2.4, providing an overview of how different elements harmonise within the simulation environment. Simswift represents the agents running on the server, highlighting the interconnectedness of these components.

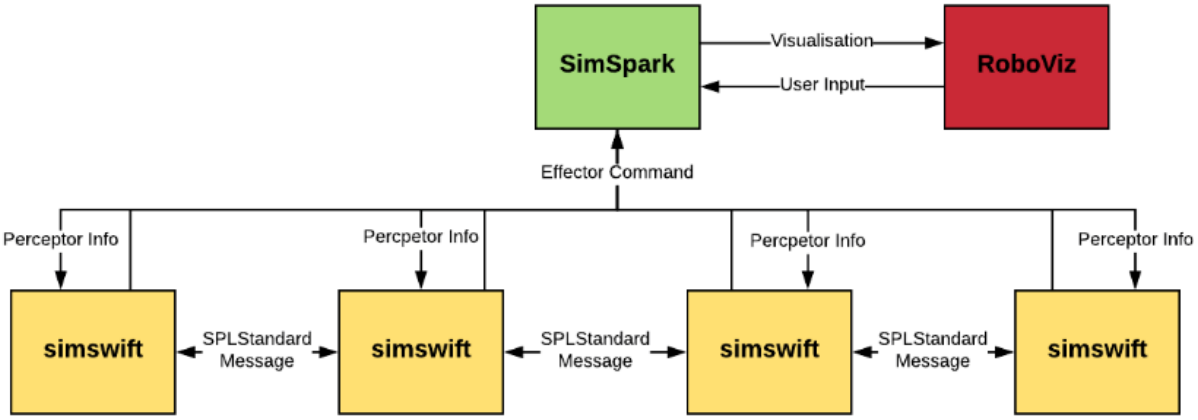


Figure 2.3: Components of RoboCup Environment

2.1.2 SimSpark

SimSpark, a versatile physical multiagent simulator system, stands as the official simulation server for the RoboCup 3D Simulation League. Setting itself apart, SimSpark offers flexibility for non-soccer simulations, enabling the creation and execution of custom environments through its scene description language [de Almeida Martins 2023]. The SimSpark simulated soccer environment adheres to specific dimensions: the pitch spans $1,181$ by 787 inches, goals have measurements of 83.07 by 23.62 inches with a height of 31.50 inches, the penalty area covers 236.22 by 70.87 inches, and there's a centre circle with a 78.74-inch radius. The soccer ball used in the simulation has a 1.57-inch radius and a mass of 26 grams. The field's distinctive features include strategically positioned markers in each corner and a goal post [de Almeida Martins 2023]. Figure 2.4 provides a visual representation of the soccer field dimensions and the marker positions. Each marker, with a unique ID and fixed location, plays a crucial role in agent localization. The agents can precisely ascertain their positions by observing the field lines in conjunction with a subset of these markers.

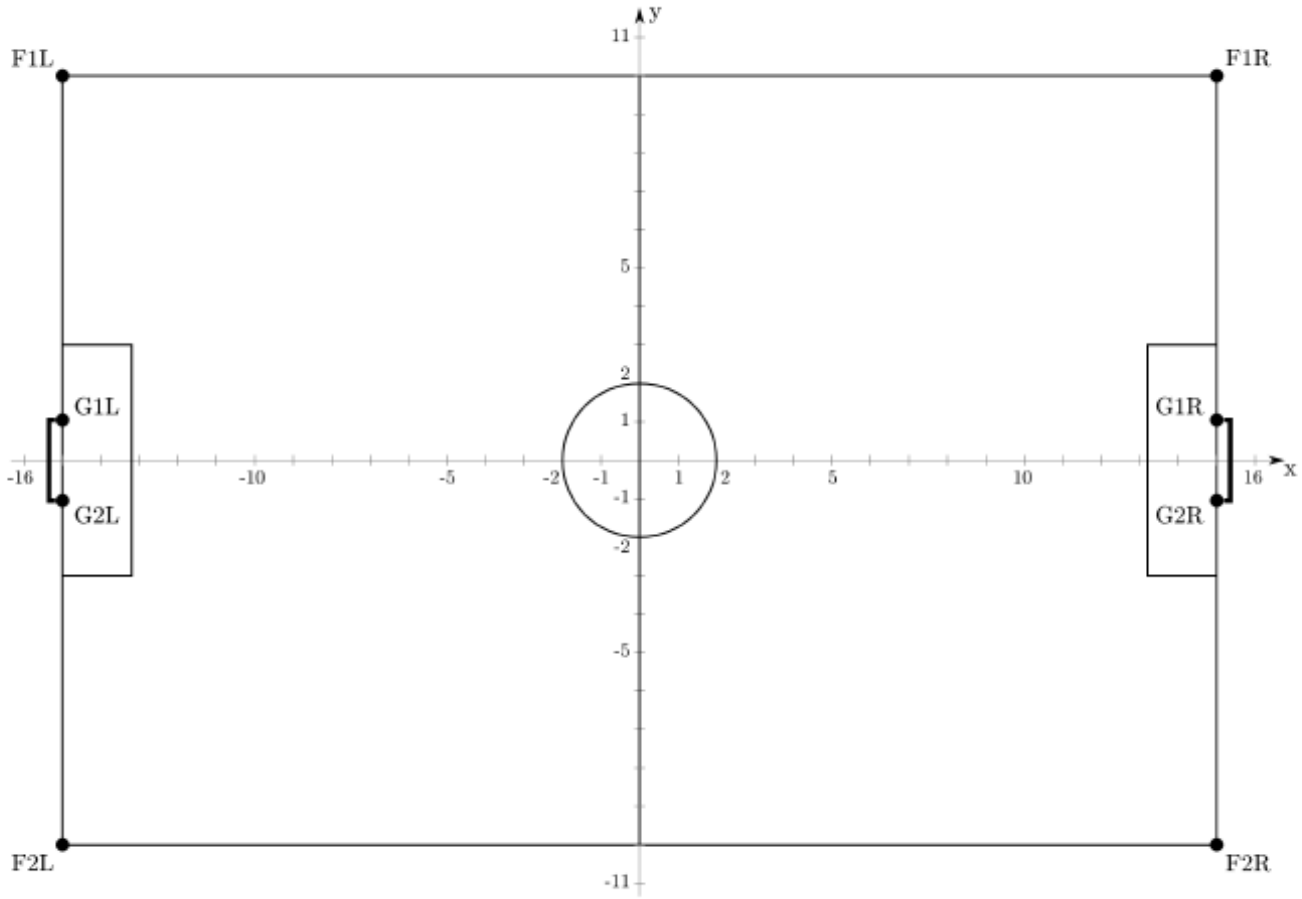


Figure 2.4: Simulated soccer field - [Lau 2023]

2.1.3 RoboViz

RoboViz, the official monitor and visualisation tool of the RoboCup 3D Simulation League [Lau 2023], is actively developed and maintained by the magmaOffenburg team. This tool facilitates 3D rendering of the environment and agents and allows for the visualisation of simple drawings, such as lines and circles. With its ability to render pathing trajectories, agent positions, and ball locations, RoboViz proves invaluable for debugging and implementing new strategies in the highly competitive RoboCup 3D Simulation League. Figure 2.5 illustrates the interface of RoboViz [Lau 2023].

Figure 2.5: RoboViz interface

2.1.4 Competition Setup

During official competitions, teams are provided with a client to execute their agents. Subsequently, these agents establish connections to a proxy, which acts as an intermediary linking to the simulation server. Concurrently, a monitor connects to the server, facilitating game rendering and enabling visualiation. The schematic representation of this setup architecture is presented in Figure 2.6.

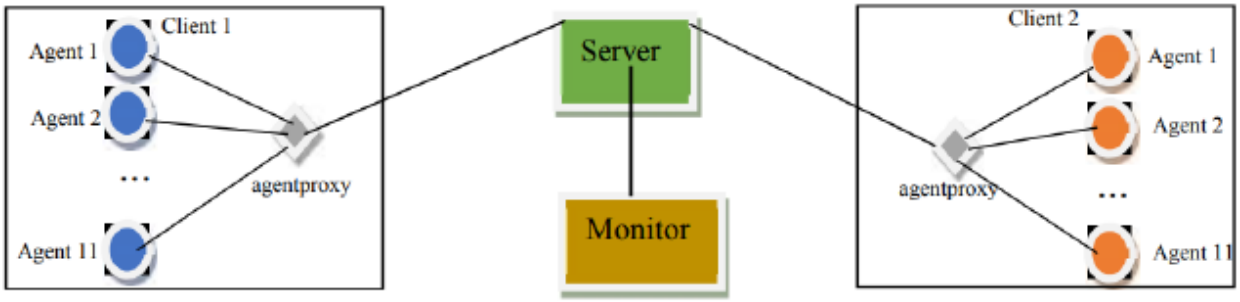


Figure 2.6: Competition environment setup of RoboCup - [Lau 2023]

2.2 Reinforcement Learning (RL)

RL refers to a paradigm of machine learning that allows an agent to acquire knowledge via trial and error in a dynamic environment while using feedback from its own actions and experiences. It employs incentives and penalties as indicators of negative and positive behaviour. The primary objective of RL is to identify an acceptable model of action that maximises the agent's cumulative total reward. The agent figures out which actions yield the greatest reward by trying them out without knowing which action to perform. Agents with reinforcement learning can perceive the features of their environment and are able to influence the environment through the actions they perform [Sutton and Barto 2018]. In Figure 2.7, the procedure within reinforcement learning is depicted. The key phrases listed below describe the fundamental elements of a reinforcement learning problem:

- **Agent** The physical object that takes an action on the environment,
- **Action** is a set of all possible moves that the agent can choose from,
- **State** This is the current moment event which the agent is in,
- **Environment** Physical world that the moves of the agent will be made on,
- **Reward** The feedback that is received back after the action is taken by the agent,
- **Policy** This is a method to map agent's state to actions,
- **Value** This is the anticipated benefit the agent will obtain after acting in a specific state,
- **On-policy** assesses the decision-making methodology,
- **Off-policy** This one assesses a different policy from the one that produced the data.

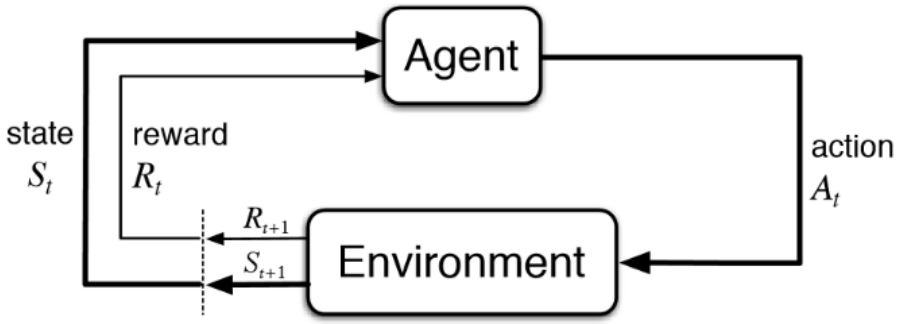


Figure 2.7: The procedure within reinforcement learning [de Sousa Pereira 2020]

2.2.1 Types of models

RL can be categorised into two distinct types: model based methods and model free methods. In the model free method, the agent maximises the expected reward from the actual observations without any prior information. It is only interested in the corresponding reward and cannot estimate the next state when choosing an action. In the model based technique, the agent first learns the model by performing an action and then observes the next state and the immediate reward. The reduced number of iterations is used during the learning process to build a model that is used to simulate the next episodes. The model-based approaches are fast because they do not require as many samples since they can select from the model [de Sousa Pereira 2020]. Figure 2.8 illustrates the relationships between model-based and model-free approaches.

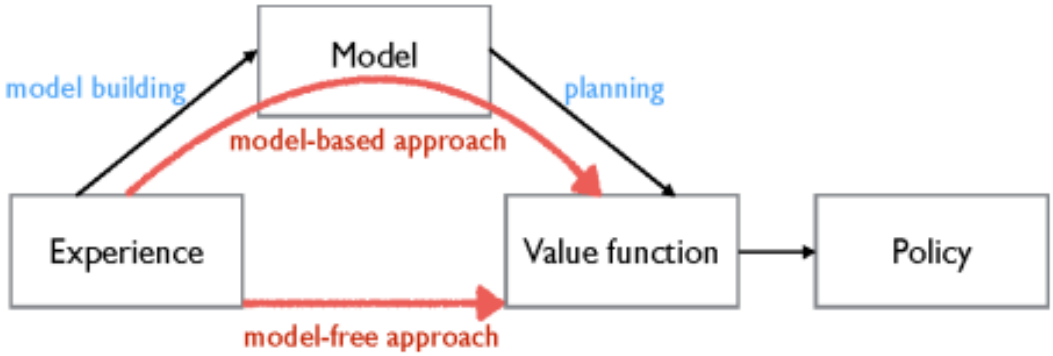


Figure 2.8: The relations of model-free and model-based approaches [de Sousa Pereira 2020].

2.3 Policy Optimisation

Policy Optimisation optimises policy based on user-specified policy and produces stable models as a result. Policy-based algorithms create a policy called $\pi_\theta(a|s)$ that calculates the likelihood that agents will take action a and then perform state s. By enhancing the function's parameter that created the policy, the policy is optimised for the job for which it was trained. The policy is updated only if it's based on the latest policy, since the optimisation is performed only on-policy. There are several ways that the policy-based models use to improve how they update the policy and θ. Figure 2.9 shows the Taxonomy of algorithms in modern RL.

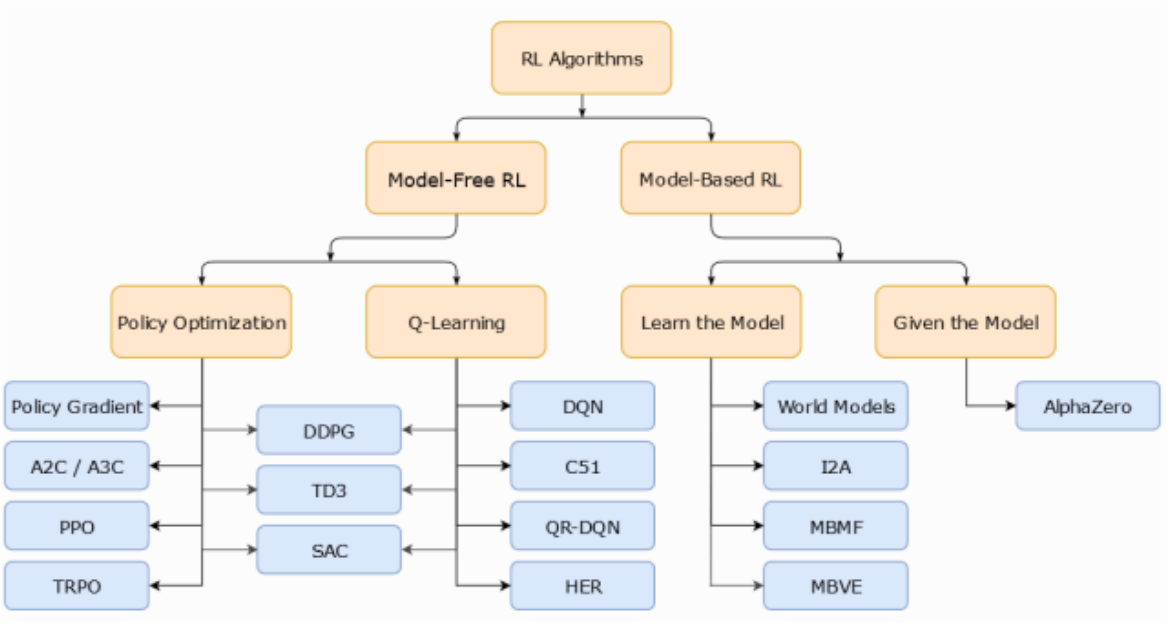


Figure 2.9: Modern RL's algorithm taxonomy - [Sousa 2021]

2.3.1 Policy Gradient

The Policy gradient algorithm aims to optimise a policy by optimising θ directly as per the reward function given by

$$J(\theta) = \sum_{s \in S} D^{\pi}(s) \sum_{a \in A} \pi_{\theta}(a \mid s) X^{\pi}(s, a)$$

where $D^{\pi}(s)$ represent a stationary Markov chain distribution for the policy π, where X^{π} represent the benefit of carrying out action a on state s based on policy π_{θ}. The target function is reformatted by policy gradient, which ignores the state distribution [Sousa 2021]. To update the parameter θ, the Monte-Carlo and REINFORCE policy gradient algorithm use the predicted return by Monte-Carlo methods with the use of episode samples [Xu and Vatankhah 2013].

2.3.2 TRPO: Trust Region Policy Optimisation

Schulman *et al.* [2015] states that TRPO was designed with the goal of improving the stability of training by reducing the variation in policy between iterations. This is done by placing a divergence boundary on the size of each iteration's policy update. This algorithm is better suited for optimising non-linear policies that are very large, such as neural networks. It updates the policies by making the biggest improvement in performance while adhering to a particular restriction on how similar the old and new policies may be. The KL-divergence, which is a measure of the distance between the probability distributions, describes the constraints.

TRPO differs from the normal policy gradient because the usual policy gradient maintains a tight proximity in parameter space between the old and new policies. Small variations in parameter space can cause extraordinarily significant differences in performance, making the normal policy gradient more dangerous than using large step sizes with vanilla policy gradients. TRPO improves performance quickly and monotonically. TRPO trains a stochastic policy using the on-policy method. It is examined using sampling operations while utilising its stochastic policy's most recent iteration. The initial condition and training technique determine how random the action selection is, and as the update rule encourages it to make use of benefits it has already received, the policy gradually becomes less random. This may result in the policy getting stuck in local optima.

2.3.3 PPO: Proximal Policy Optimisation

PPO is a policy gradient algorithm that builds upon Trust Region Policy Optimisation (TRPO). PPO introduces modifications to TRPO by integrating a penalty constraint into the objective function to tackle convergence issues. PPO differentiates itself from other policy gradient methods and TRPO through several key aspects. One of the notable modifications introduced by PPO is the integration of a penalty constraint into the objective function, addressing convergence issues encountered by TRPO. By incorporating this penalty constraint, PPO achieves more stable and reliable policy updates, enhancing the optimisation process. In terms of the policy update mechanism, PPO utilises a trust region policy optimisation approach. This involves constraining the policy update through a clipping mechanism embedded in the objective function. The clipping mechanism ensures that the updated policy remains within a specified range, preventing large policy changes that could potentially lead to instability [Schulman *et al.* 2017]. This feature distinguishes PPO from other methods by providing a smooth balance between exploiting the advantages of the current policy and exploring alternative actions, ultimately facilitating the discovery of robust and reliable solutions.

Hämäläinen *et al.* [2020] state that one of the advantages of PPO lies in its simplicity and ease of implementation compared to other policy gradient algorithms. It exhibits superior sample efficiency, allowing for more efficient use of collected experiences during training. Additionally, PPO offers strong numerical performance and exhibits stable convergence properties. Because of these features, PPO is a good choice for many reinforcement learning tasks. It lets experts train useful policies with fewer computing needs and better guarantees of convergence. The clipping mechanism and the trust region policy optimisation approach work together to make PPO a well-known algorithm in the field of reinforcement learning. It makes policy updates stable and reliable. Its simplicity, efficiency, and robust convergence properties contribute to its popularity and successful application in various domains.

Objective Function

The objective of PPO is to update the policy parameters θ to maximise the expected reward. The policy update is performed by optimising the following objective function:

$$\theta_{n+1} = \arg\max_{\theta} \mathop{\mathrm{E}}_{s,a\sim\pi_{\theta_n}} \left[L\left(s, a, \theta_n, \theta\right) \right],$$

The objective function $L(s, a, \theta_n, \theta)$ is defined as:

$$L\left(s, a, \theta_n, \theta\right) = \min\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_n}(a \mid s)} A^{\pi_{\theta_n}}\{s, a), \operatorname{clip}\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_n}(a \mid s)}, 1-\epsilon, 1+\epsilon \right) A^{\pi_{\theta_n}(s,a)} \right),$$

Here, $\pi_\theta(a|s)$ represents the current policy, $\pi_{\theta_n}(a|s)$ represents the policy from the previous iteration and $A^{\pi_{\theta_n}}(s, a)$ is the advantage function that estimates the advantage of taking action a in state s over a baseline. The hyperparameter ϵ controls the extent of the policy update.

Policy Update

PPO employs a trust region policy optimisation approach to ensure stable policy updates. By introducing a clipping mechanism in the objective function, PPO constrains the policy update to a specified range. This ensures that the updated policy remains close to the previous policy, preventing large policy changes that may lead to instability. The clipping mechanism allows the policy update to strike a balance between exploiting the advantages of the current policy and exploring alternative actions. It provides a smooth transition between policy improvement and exploration, enabling PPO to find robust and reliable solutions.

Exploration and Exploitation

In reinforcement learning, the exploration-exploitation trade-off is a critical aspect. PPO maintains a level of exploration throughout the training process to avoid getting trapped in local optima. By sampling actions based on the most recent stochastic policy, PPO explores different action choices and gathers diverse experiences. However, as the policy update progresses and the advantages of certain actions become apparent, PPO gradually shifts towards exploitation by reducing the level of randomness in action selection. This balance between exploration and exploitation is essential for PPO to discover high-quality policies while maximising the expected rewards. PPO incorporates entropy regularisation as an exploration strategy. By adding an entropy term to the objective function, PPO encourages the policy to produce actions with higher entropy or unpredictability. This promotes exploration by preventing the policy from becoming too deterministic and encouraging it to explore different actions and states. The balance between exploitation and exploration is crucial in PPO, as it allows the algorithm to navigate complex and diverse environments effectively, leading to more robust and adaptive policies. By carefully managing the exploration-exploitation trade-off through the trust region policy optimisation approach and entropy regularisation, PPO strikes a balance between exploring new possibilities and exploiting existing knowledge. This enables it to discover effective policies and achieve convergence to optimal or near-optimal solutions in a wide range of reinforcement learning tasks.

2.4 Objective

The primary objective of this research is to develop a Proximal Policy Optimisation (PPO) method for optimising the kicking behaviour of the robot Nao in three-dimensional RoboCup. The proposed PPO method aims to improve the robot's kicking accuracy and decision-making process by fine-tuning its policy parameters. By leveraging PPO's ability to efficiently learn from past experiences while exploring new strategies, the research seeks to enhance the overall performance of the robot in complex soccer scenarios.

2.5 Significance of the project

This study proposes the PPO method to improve the performance of the robot Nao by optimising its kicking behaviour in a complex and dynamic environment. By leveraging the advantages of PPO, such as stable policy updates and efficient convergence, this study aims to enhance the robot's capability to adapt and learn in real-time scenarios, since kicking is one of the many skills that are important for robots to play soccer successfully. It is important to find the best kicking parameters for the Nao robot in the three-dimensional RoboCup simulation league so that the team is able to score goals in the three-dimensional RoboCup simulation league competitions. They also allow the team to win some games or competitions, which is the main reason why the teams compete against each other. In summary, this is important because it will improve the team's performance in all future games.

2.6 Structure of the project

These are the remaining points of the plan. In Chapter 3, related work is covered. Chapter 4 presents the methodology, Chapter 5 the research plan and Chapter 6 the conclusion of this paper.

Chapter 3

Related Work

3.1 Introduction

The earlier research on the kicking behaviour of robots in the three-dimensional RoboCup simulation competition is covered in this chapter. These include various methods that were used to improve the kicking behaviour of a robot in the RoboCup three-dimensional League by optimising the kicking parameters. Various observations were made, such as the kicking delay, kicking accuracy and kicking the ball at a long distance, which were observed in the University of Witwatersrand robot football team during RoboCup play against other teams in the three-dimensional RoboCup simulation League.

3.2 Review of related work

The study by Teh [2012] introduced a novel method developed by rUNSWIft to improve the balance and accuracy of kicking in RoboCup. The researchers recognised that kicking the ball with accuracy and balance is a challenging task for Nao robots in RoboCup. They devised a technique where the Nao robot approaches the ball and turns towards the goal during the kicking motion. This approach, called the omnidirectional kick, was specifically designed for the Aldebaran Nao robot and aimed to advance the performance of rUNSWIft's striker robot in the RoboCup Standard Platform League.

The implementation of this kicking style took into account various factors such as speed, balance, precision and dependability. The kinematics module was used to understand the robot's posture by considering its centre of gravity and body tilt. The result was a dynamic omnidirectional kick that enabled faster targeting and increased force without the need to manipulate the hip joint or sacrifice power. Dribbling kicks were also incorporated to enhance speed and maneuverability, albeit at the cost of distance and power. Additionally, modifications were made to the front-get-up exercises, incorporating new inward arm swings to improve their reliability. In the 2012 RoboCup Standard Platform League, rUNSWIft's performance demonstrated the effectiveness of their approach. Al-

though they achieved a third-place finish, the competition was strong, showcasing the progress made by numerous top teams over the years [Teh 2012].

The study by MacAlpine *et al.* [2015] focused on the University of Texas Austin Villa RoboCup team, which achieved victory in the 2015 RoboCup three-dimensional League by scoring 87 goals. The research highlighted the changes made to the team's approach between 2014 and 2015, which led to their success in winning all of their games during the competition and league. One significant improvement made by the Texas University Austin Villa team was the addition of 13 new shot options for their robots to choose from in 2015. In the previous year, the team only had four kick options available, including a quick short kick and high and low kicks that could cover distances of approximately 5m and 15m respectively. These kicks were effective but limited in their range.

The team also participated in a robot kicking accuracy competition where teams had their robots shoot a ball into the centre of the field from various distances ranging from 3m to 12m in 1m increments. By employing learned accurate kicks, the University of Texas Austin Villa team won the kicking accuracy competition by achieving the least error in kicking distance. The kick optimisation process in their approach utilised CMA-ES. This optimisation method enabled the agents to generate different directional kicks using simple curves. The parameters for the kicks were learned through an optimisation task where the agents were trained to approach the ball from at least 10 different angles. In summary, the University of Texas Austin Villa team's accomplishment in the 2015 RoboCup three-dimensional League was attributed to the expansion of their kick options, the utilisation of CMA-ES for kick optimisation and the training of agents to perform accurate kicks from various angles [MacAlpine *et al.* 2015].

The kick optimisation study conducted by Depinet *et al.* [2014] utilised the CMA-ES strategy, which had previously been employed to optimise the walking behaviour of the robot. The optimisation process involved a population size of 200 and after 400 iterations of CMA-ES, a stride that was 5 meters longer than the initially observed kick was achieved. Two additional fitness factors were considered in the optimisation process, resulting in slight variations between the two kicks. The first fitness parameter focused on accuracy, incorporating a Gaussian penalty based on the variance between the intended and achieved angles. This fitness function aimed to produce a robust and accurate kick.

The second fitness parameter emphasised the distance the ball travelled through the air. The objective was to shoot the ball over the heads of opponents from a long distance. In order to account for the agents' height, a reference height of 0.5 meters above the ground was used to determine the rewards for this fitness function. The optimisation results with this feature generated a powerful kick that travelled over 11 meters in the air. The provided equation represents the fitness function for the air distance parameter, including a setback penalty, a reward for hitting the goal and a reward based on the ball's final location and twice the air distance.

In Abdolmaleki *et al.* [2016], the kick controller for a virtual Nao robot was optimised using the CREPS-CMA technique. Different optimisation functions were were selected. A comparison was made between the CREPS-CMA algorithm and the contextual standard REPS, generating 50 new samples for each iteration. The kick was optimised over 1000 iterations, considering ideal kick-off distances ranging from 2.5 meters to 12.5 meters. The findings indicated that the contextual tasks were effectively learned by the CREPS-CMA approach, whereas REPS did not demonstrate the same level of success. The experimental setup involved placing the robot in various positions relative to the ball, facing it as it approached, positioning itself and executing kicks towards the target using the optimised kick controller. It was observed that the nonlinear policies outperformed the linear policies in terms of performance.

de Sousa Pereira [2020] employed the TD3 and SAC algorithms, along with stable baseline frameworks, to optimise corner kicks in the three-dimensional RoboCup simulation league. The neural network algorithms controlled the kicking agent, which aligned and kicked the ball using a controlled kick within the penalty area. The optimisation process focused on obtaining rewards at the end of each episode, considering the final position of the ball. The robots had various constraints in their action spaces to prevent undesired behaviours. A three-layer neural network with 256 nodes in the first and last layers and 128 nodes in the middle layer, was utilised for the optimisation. The results demonstrated that the optimised robot achieved positive rewards and exhibited improved performance in kicking the ball closer to the goal line and the intended teammate. The crossing agent, responsible for crossing the ball into the box, also showed stability and found optimal values for successful kicks.

In a related work, Barrett *et al.* [2010] investigated controlled kicking under uncertainty. They developed a kick engine and a kick selection algorithm for humanoid robots in the 2010 RoboCup. The kick engine allowed variable distance and direction kicks, reducing the need for precise alignment. The kick decision engine considered the agent's position uncertainty, kick outcome and time constraints, resulting in fast ball movement across the field.

Furthermore, Melo and Máximo [2019] employed PPO methods to optimise the walking capabilities of a humanoid robot. They created two optimisation tasks, one with a focus on forward speed and the other on stability and locomotion. The tasks utilised episode termination conditions based on reaching a finish line or a predetermined episode duration. The study showed significant improvements in both forward speed and stability after several hours of training.

These related works are relevant to the current research as they explore optimisation techniques for specific behaviours in humanoid robots. While de Sousa Pereira [2020] focused on corner kicks, Barrett *et al.* [2010] investigated general kicking strategies and Melo and Máximo [2019] addressed walking capabilities. This proposal builds upon these studies by using PPO to optimise the kicking accuracy of Nao robots in the RoboCup three-dimensional league.

Chapter 4

Research Methodology

4.1 Introduction

In this section, we provide an overview of the research methodology adopted for investigating and enhancing the kicking performance of RL-driven agents within the RoboCup simulation environment. The discussion encompasses key elements such as the formulation of research questions and the detailed setup of the Proximal Policy Optimisation (PPO) algorithm specifically tailored for RoboCup. This methodology forms the foundation for our exploration into optimising kicking behaviour and understanding the implications of reducing decision-making latency in the dynamic context of RoboCup.

4.2 Research Questions

The primary focus of this study is to investigate the effectiveness of Proximal Policy Optimisation (PPO) in the context of RoboCup. To accomplish this objective, this study seeks to address the following research questions:

1. **Optimisation Objective:** How can Proximal Policy Optimisation (PPO) be leveraged to enhance the kicking performance of RL agents in the RoboCup simulation?

2. **Impact of Latency Reduction:** What is the effect of reducing decision-making latency on kicking performance and how does this influence the overall behaviour of the RL agent?

4.3 PPO Setup for RoboCup

The training of Nao agents for RoboCup soccer involves the utilisation of the Proximal Policy Optimisation (PPO) algorithm, adapted to suit the specific requirements of the task. PPO is a reinforcement learning algorithm that facilitates effective learning by iteratively optimising policy functions. In the context of RoboCup, the PPO algorithm enables agents to interact with the environment, observe states and execute actions based on learned policies.

4.3.1 State Representation

The state representation is a critical component of the learning process, providing the agents with a comprehensive view of the environment. It includes a multitude of features extracted from the RoboCup simulation, encompassing ball positions, player coordinates, the spatial arrangement of teammates and opponents and other relevant game-related information. This rich state representation serves as the foundation for the agents' decision-making process, enabling them to contextualise their actions within the dynamic soccer environment.

4.3.2 Action Space

The action space in this context is precisely determined by the target joint angles and joint velocities of the Nao agents within the RoboCup environment. Specifically, the RL agents can output commands related to adjusting these joint parameters. This focused definition ensures a more accurate representation of the agent's control capabilities, centering on the manipulation of joint angles and velocities. The constrained action space allows for precise motor control, enabling the agents to influence the robot's movements with detailed adjustments. Within this refined space, agents can learn sophisticated motor control strategies, contributing to effective in-game actions such as strategic ball handling and goal-oriented decision-making.

4.3.3 Iterative Training Process

The training process unfolds iteratively, with agents engaging in interactions with the environment, receiving feedback in the form of rewards and adjusting their policies accordingly. This cyclical process of exploration and exploitation enables the agents to progressively refine their strategies, converging towards more optimal policies. The training duration, represented by the number of time steps, influences the depth of learning and the adaptability of agents to diverse game scenarios.

4.3.4 Policy Evaluation and Analysis

The analysis of trained policies involves a comprehensive examination of the agent's behaviour in various situations. Performance metrics, including the defined rewards and key game-play statistics, are logged and analysed to gauge the efficacy of the learned policies. This analysis extends beyond mere quantitative metrics, delving into qualitative aspects of game-play to ensure a holistic evaluation of the agent's soccer-playing capabilities.

4.4 Environment Setup

The experiments are conducted within the context of the RoboCup soccer simulation. The environment used encapsulates the complexities of the soccer domain, providing the agent with a rich sensory input and the ability to execute a variety of actions. To

ensure a comprehensive exploration of the agent's capabilities, we configure the environment with specific parameters. The number of time steps and wait time parameters play a pivotal role in determining the duration of each experiment and the temporal dynamics the agent must adapt to. Additionally, the agent's perception was enhanced by stacking consecutive frames, enabling it to capture temporal dependencies crucial for decision-making.

Under the Evaluation Metrics section, the reward specifications are detailed and guide the agent's learning process. Each reward, including the kick latency reward, backswing effectiveness, and distance optimisation, is meticulously defined to represent distinct aspects of desirable kicking behaviour. These rewards collectively form the basis for evaluating the agent's performance. The inclusion of Kick Latency Reward is a focal point, aimed at minimising decision-making delays during kicking actions. The experiments are initiated with the same reward specifications from previously trained models for kick behaviour with PPO, providing a starting point for the agent's learning process. The models are loaded from a checkpoint, ensuring consistency in the starting conditions for each experiment. This approach facilitates a controlled exploration of the impact of the Kick Latency latency reward, as any observed differences in performance can be attributed to the introduction or absence of this specific reward. The utilisation of previously used reward specifications aligns with a transfer learning paradigm, leveraging prior knowledge to expedite learning in the RoboCup soccer environment.

The experimental setup is designed to support an iterative approach to reinforcement learning. After each experiment, the agent's performance is scrutinised, and insights are used to refine the reward specifications. This iterative cycle of experimentation and refinement is integral to the agent's learning process, ensuring adaptability to the dynamic nature of the RoboCup soccer simulation. By systematically varying reward configurations and analysing the corresponding outcomes, we aim to uncover optimal combinations that synergise with Kick Latency Reward, ultimately enhancing the agent's proficiency in executing swift and effective kicks.

4.5 Reward Function

To guide the learning process, we designed a reward function that provides feedback to the agents based on their actions and the current state. The reward function aims to incentives desirable behaviours, such as scoring goals, successfully passing the ball and exhibiting effective teamwork. It penalises undesirable behaviours, such as own goals, collisions and unsuccessful passes. The specific formulation of the reward function depends on the particular objectives and requirements of the RoboCup soccer domain. We will carefully design the reward function to ensure a balance between encouraging optimal play and providing informative feedback to guide the learning process.

Reward	Magnitude
Keep Backswing Reward	True
Fall Penalty Magnitude	100
Max Distance Reward	True
Cumulative Distance Reward	True
Z-distance Weighted Reward	10,000
Ball Distance Magnitude Reward	1
Backswing Magnitude Reward	3
Backswing Stop Y Reward	True
Penalise Y Distance Reward	True

Table 4.1: Integrated Rewards and Magnitudes

The table 4.1 details these rewards along with their respective magnitudes. This comprehensive set of rewards, each addressing specific aspects of kicking behaviour, plays a synergistic role in optimising the agent's performance.

4.5.1 Algorithm

PPO works like other policy gradient methods in that it calculates the output probabilities in the forward pass based on various parameters and uses the backward pass to calculate the gradients to improve the probabilities. Below 4.5.1 is the working PPO algorithm:

Algorithm 1 PPO-clip - [Schulman *et al.* 2017]

Input: original policy coefficients θ_0 original value function coefficients ϕ_0

for iteration $= 0, 1, 2, \ldots$ do

gather a group of paths $\mathcal{D}_{\text{iter}}$ by running the policy $\pi_{\text{iter}} = \pi(\theta_{\text{iter}})$ in the environment.

Calculate the cumulative rewards (rewards-to-go) for each time step in the paths.

Estimate the advantages (advantage approximations) based on the current value function $V_{\phi_{\text{iter}}}$ using a suitable method.

Improve the policy by maximising the PPO-Clip objective:

$$\theta_{\text{iter}+1} = \text{argmax}_\theta \left(\frac{1}{|D_{\text{iter}}| \cdot T} \sum_{\text{path} \in D_{\text{iter}}} \sum_{t=0}^{T} \min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{iter}}}(a_t|s_t)} \cdot \text{advantages}, g(\epsilon, \text{advantages}) \right) \right)$$

(typically using stochastic gradient ascent with Adam or a similar optimisation algorithm)

Approximate the value function by minimising the mean squared error through regression:

$$\phi_{\text{iter}+1} = \text{argmin}_\phi \left(\frac{1}{|D_{\text{iter}}| \cdot T} \sum_{\text{path} \in D_{\text{iter}}} \sum_{t=0}^{T} (V_\phi(s_t) - \text{rewards})^2 \right)$$

(typically using a gradient descent algorithm)

end for

4.6 Logging and Analysis Infrastructure

The experiments are logged and analysed using the Weight and Biases (WandB) platform, a comprehensive tool for experiment tracking and result visualisation. WandB serves as a centralised repository for storing experiment configurations, training trajectories and performance metrics. It facilitates collaborative research efforts and fosters a deeper understanding of the agent's behaviour by providing a unified platform for experiment management.Weight and Biases (WandB): WandB is an experiment tracking platform that offers a suite of tools for machine learning practitioners. It enables seamless logging of experiments, tracking of metrics and visualisation of results. WandB's interactive dashboards provide a holistic view of experiment outcomes, making it easier to discern patterns and make data-driven decisions. By integrating WandB into our experimental setup, we enhance the reproducibility and transparency of our research, creating a robust foundation for iterative experimentation and analysis.

4.7 Performance Evaluation

In assessing the performance of trained Nao agents in the RoboCup environment, a suite of evaluation metrics is employed. These metrics encapsulate various aspects of agent behaviour, providing a nuanced understanding of their proficiency in soccer game-play. Evaluation metrics revolve around reward structures, crucial in shaping the learning trajectory of the RL agent. Key rewards include:

Kick Latency Evaluation

Central to the evaluation is the Kick Latency metric, a key parameter designed to minimise delays in kicking actions. The reduction of latency is crucial for real-time responsiveness, impacting the agent's ability to make timely decisions during game-play. The Kick Latency Evaluation involves measuring the time elapsed between the decision to kick and the actual execution, with lower latency values indicative of more agile and responsive agents.

back-swing Management

The management of back-swing, represented by the Keep back-swing Reward, plays a pivotal role in refining kicking strategies. A well-controlled back-swing contributes to more accurate and powerful kicks. The evaluation of back-swing management involves assessing how effectively agents utilise and control their back-swing during kicking actions, promoting precision in ball interactions.

Spatial Proficiency

Metrics such as Max Z-Height, Max Distance, Cumulative Distance and Cumulative Z-Height provide insights into the spatial awareness and navigation capabilities of the agents. Max Z-Height measures the maximum vertical height reached by a kicked ball, indicating the agent's ability to vary the trajectory of its kicks. Max Distance and Cumulative Distance quantify the spatial reach of kicks, reflecting the agent's proficiency in covering ground efficiently. Cumulative Z-Height extends the spatial analysis to the vertical dimension, offering a comprehensive perspective on the agent's spatial behaviours.

Strategic Decision-making

The assessment of strategic decision-making involves metrics like back-swing Stop Y and Target Latency. back-swing Stop Y evaluates the precision of back-swing control concerning the Y-axis, contributing to accurate ball targeting. Target Latency measures the time delay in targeting decisions, providing insights into the agent's ability to make informed and strategic choices during game-play.

4.8 Robustness Evaluation

To evaluate the robustness of the PPO algorithm, tests are conducted under various scenarios and conditions. This includes testing with different reward specifications. By examining the algorithm's performance across these diverse conditions, its ability to adapt and perform consistently is assessed. The comprehensive methodology presented above provides a framework for evaluating the performance of Proximal Policy Optimisation (PPO) in the context of RoboCup. While low Kick Latency is desirable, it must be balanced with effective back-swing management and spatial proficiency to ensure a well-rounded soccer-playing capability. The interdependence of these metrics reflects the nuanced nature of evaluating agent performance, emphasising the need for a comprehensive and multifaceted assessment approach. From kick latency to spatial proficiency and strategic decision-making, these metrics collectively contribute to a comprehensive understanding of the agents' soccer-playing capabilities.

4.9 Conclusion

In conclusion, this methodology establishes a structured framework for training and evaluating RL agents within the RoboCup simulation environment. With a focused inquiry into refining kicking behaviour for robotic agents, the approach aligns with specific research questions. The integration of well-defined evaluation metrics offers a systematic means to derive insights into the nuanced dynamics of agent performance. As this research unfolds, the goal is to not only enhance the immediate context of RoboCup game-play but also to contribute broader understandings applicable to real-world scenarios involving robotic decision-making and spatial navigation. This methodology lays the groundwork for advancing the capabilities of RL-driven robotic systems, bridging the gap between simulation and practical application.

Chapter 5

Results

5.1 Introduction

In this section, we present the outcomes of a comprehensive set of experiments designed to optimise kicking behaviour for robotic agents in the RoboCup simulation environment. The experiments were conducted using the Proximal Policy Optimisation (PPO) algorithm, exploring various scenarios to understand the impact of different reward structures on the learning process.

5.2 Training Scenario 1: PPO without Kick Latency Reward

In the initial training scenario, the primary objective was to comprehensively understand the fundamental learning dynamics of the Proximal Policy Optimisation (PPO) algorithm when applied to train kicking behaviour. This foundational phase aimed to assess how well the agent could learn and optimise its kicking policy without the influence of latency reduction. This deliberate exclusion of kick latency rewards provided a baseline understanding of the kicking capabilities before introducing latency-related enhancements.

5.2.1 Training Focus and Objective

The primary focus of this initial training scenario was to evaluate the inherent learning capabilities of the RL agent in acquiring an effective kicking policy. By employing the PPO algorithm, the training aimed to leverage reinforcement learning principles to teach the agent to perform kicks optimally. The omission of kick latency rewards during this phase allowed for an unadulterated examination of the agent's ability to adapt and learn, providing valuable insights into the baseline kicking performance.

5.2.2 Rewards Utilised for Encouraging Kicking Behaviour

In the initial training scenario without kick latency rewards, specific reward structures were strategically employed to assess the fundamental learning capabilities of the Proximal Policy Optimisation (PPO) algorithm in enhancing kicking behaviour. One of the key rewards employed was the Max Distance Reward, designed to incentivise the RL agent to maximise the distance covered by the kicked ball. The observed upward trend in the Max Distance Reward graph indicates the RL agent's progressive improvement in kicking the ball over greater distances. This upward trajectory suggests successful adaptation, demonstrating increased proficiency in achieving longer kicks, aligning with the overarching training objective. Refer to Figure 5.1 for a visual representation of the Max Distance Reward graph.

Another crucial metric is the total reward, a composite measure comprising contributions from max distance, cumulative distance and max height rewards. The upward trend in the Total Reward graph reflects the cumulative success across multiple reward components. This positive trajectory implies that the RL agent effectively combines different aspects of kicking behaviour, such as distance and height, to maximise the overall reward. Figure 5.2 provides a visual representation of the total reward graph.

5.2.3 Max Distance Reward

Figure 5.1 illustrates the continuous improvement in the RL agent's ability to achieve longer kicks. The positive trend is indicative of successful adaptation and proficiency in maximising kicking distance. This distance is similar to the kick currently employed in the Robocup team, which can reliably achieve a distance of approximately 5 units.

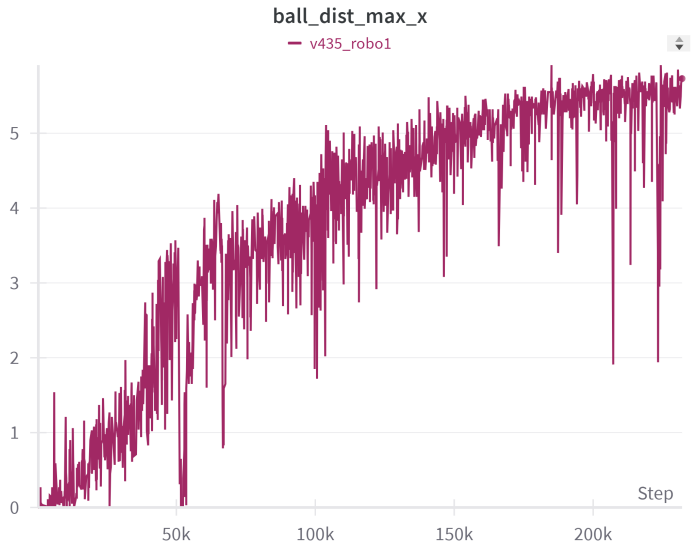


Figure 5.1: Max Distance Reward Graph

5.2.4 Total Reward

Figure 5.2 provides a holistic view of the RL agent's performance, showcasing the cumulative effect of max distance, cumulative distance and max height rewards. The upward trend underlines the agent's success in integrating various kicking aspects for enhanced performance. This upward trend also validates the design choice of PPO, which is able to improve the agent's performance through experience.

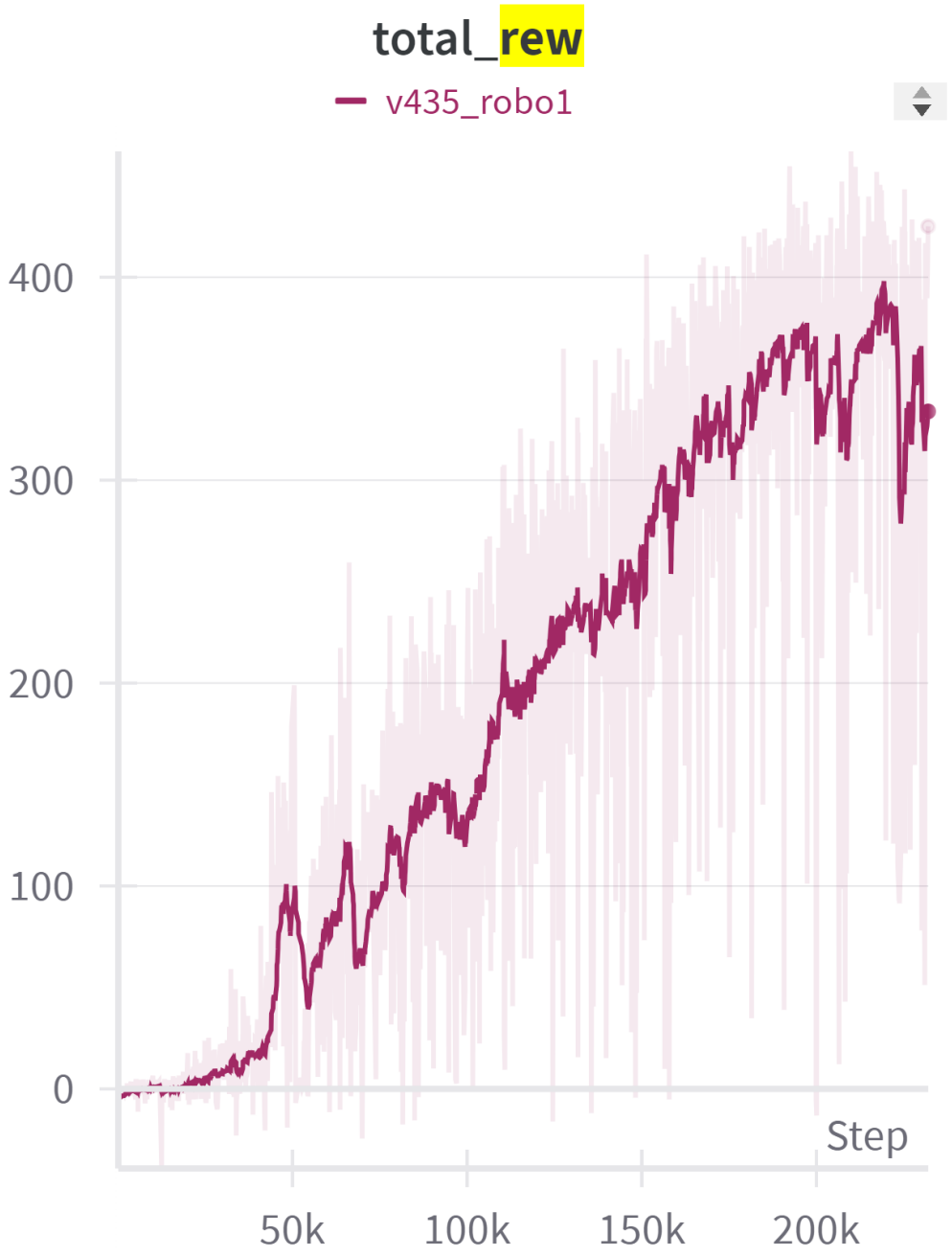


Figure 5.2: Total Reward

The delay in the RL model's ability to start learning and getting rewarded for reducing latency is intrinsic to the nature of reinforcement learning. Early in the training process, the agent explores a wide range of actions and their consequences, which often leads to suboptimal and random behaviours. This exploration phase is crucial as it allows the agent to gather diverse experiences and understand the environment's dynamics. Over time, as the agent receives feedback through rewards and penalties, it starts to identify patterns and adjust its actions to maximise the cumulative reward. The initial phase, where rewards remain low or constant, reflects this period of exploration and trial-and-error learning. It takes a significant number of steps before the agent's learning algorithm can effectively discern the actions that lead to desired outcomes, such as reduced kick latency. This gradual improvement is a hallmark of RL, where substantial training is often required before noticeable progress is made in the targeted behaviours.

5.2.5 Ball Distance Final Euclidean (Euc) Reward

The Ball Distance Final Euclidean (Euc) reward serves as a critical metric for evaluating the RL agent's precision in ball placement relative to the target after completing the kicking action. Contrary to the previous understanding, the Euc reward calculates the Euclidean distance between the agent and the ball at the end, indicating how much the ball moved in the y dimension vs the x dimension. The upward trend in the corresponding Figure 5.3 indicates positive progress in the RL agent's ability to consistently position the ball closer to the desired target. This improvement underscores the accuracy and consistency achieved by the RL agent, crucial for real-world applicability.

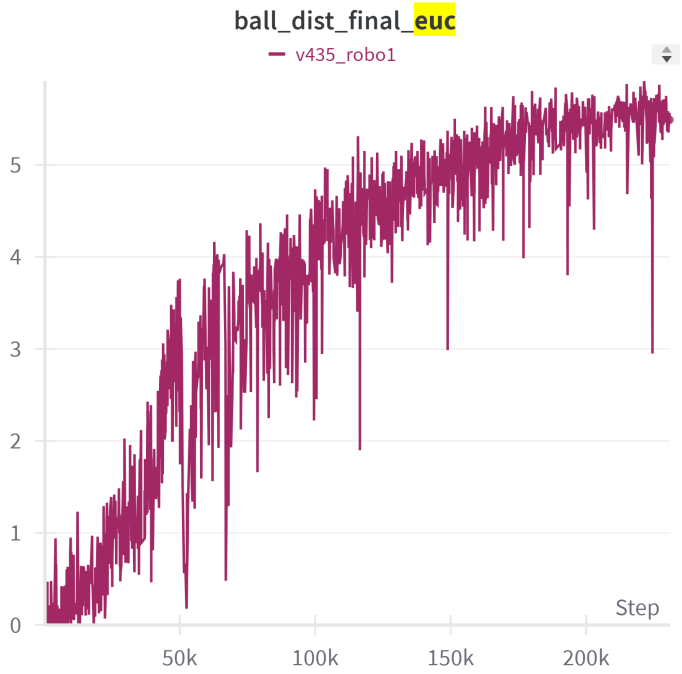


Figure 5.3: Ball Distance Final Euclidean (Euc) Reward without kick Latency

Implications and Analysis:

- **Accuracy Improvement**: The upward trend in the Euclidean reward signifies progress in the agent's ability to kick further, which indirectly suggests enhanced accuracy in reaching the desired target. Importantly, the values closely aligning with the max x numbers indicate a high level of accuracy in achieving the desired target location. This alignment demonstrates the agent's proficiency in consistently hitting the target, reflecting a positive trajectory towards improved accuracy in kicking.

- **Consistent Performance**: The consistent upward movement in the Euclidean reward highlights the agent's sustained improvement in kicking distance over training episodes. This consistency underscores the reliability of the agent's performance in consistently achieving longer kicks, contributing to its overall effectiveness on the field.

- **Integration of Distance Metrics**: The Euclidean distance metric provides a comprehensive assessment of spatial dimensions, allowing for effective learning across different dimensions. By considering both horizontal and vertical components of the kicking action, the agent gains a more holistic understanding of its kicking capabilities. This integration ensures that the agent learns and adapts to various spatial dimensions, leading to a more versatile and adaptable kicking behaviour.

In summary, the applied rewards successfully incentivised the RL agent's adaptation and improvement in achieving longer kicks, integrating various kicking aspects, and enhancing accuracy in ball placement. A detailed analysis of these reward components provides a comprehensive understanding of the RL agent's kicking behaviour.

5.2.6 Exclusion of Kick Latency Rewards

The intentional exclusion of kick latency rewards in this phase served a strategic purpose. By meticulously observing and analysing the RL agent's kicking behavior without latency reduction, the study aimed to address the first research question: How can Proximal Policy Optimisation (PPO) be leveraged to enhance the kicking performance of RL agents in the RoboCup simulation? This initial training phase establishes a baseline, allowing for a thorough examination of the agent's baseline kicking capabilities.

Importantly, this baseline becomes a crucial reference point for subsequent experiments where kick latency rewards are introduced. The detailed analysis of the RL agent's learning dynamics, as presented in the following sections, contributes to a comprehensive understanding of its baseline kicking abilities. Moreover, this baseline becomes instrumental in evaluating the effectiveness of kick latency rewards, providing a clear benchmark for comparison in subsequent scenarios.

5.3 Training Scenario 2: PPO with Kick Latency Reward and Additional Rewards

In the second training scenario, the Proximal Policy Optimisation (PPO) algorithm was employed with a focus on reducing decision-making latency, addressing the critical question of how latency reduction influences kicking performance. This scenario aimed to optimise the kicking behaviour of RL agents in the RoboCup simulation, aligning with the overarching optimisation objective.

5.3.1 Optimisation Objective: Leveraging PPO for Enhanced Kicking Performance

The incorporation of kick latency rewards, coupled with additional rewards, facilitated a nuanced exploration of Proximal Policy Optimisation's (PPO) capabilities. Notably, the Max Distance Reward maintained its significance as one of the pivotal metrics. The discernible upward trend illustrated in Figure 5.4 stands out, signifying the RL agent's consistent improvement in achieving longer kicks. This upward trajectory implies a commendable adaptation on the part of the agent, showcasing heightened proficiency in covering substantial distances during kicking actions. The noteworthy aspect is the maximisation of the Max Distance Reward between 6 and 7, indicative of consistently impressive kicking performance.

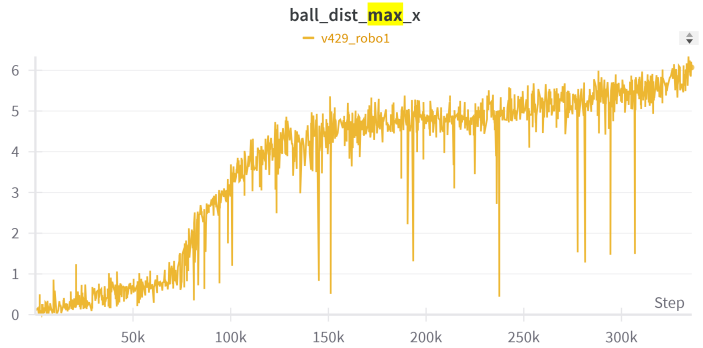


Figure 5.4: Max Distance Reward with Kick Latency

This positive trend in the Max Distance Reward not only underscores the adaptability of the RL agent but also highlights the efficacy of incorporating kick latency rewards in the training paradigm. Despite the added complexity introduced by kick latency considerations, the RL agent not only maintains but enhances its ability to execute powerful and well-directed kicks. This observation supports the notion that kick latency rewards, far from hindering performance, contribute to a more refined and efficient decision-making process, ultimately enhancing the overall effectiveness of the RL agent in the RoboCup environment.

5.3.2 Total Reward

The upward trend observed in the total reward, as depicted in Figure 5.5, carries significant implications for the experiment. A positive trend in the Total Reward aligns with the overarching objective of reinforcement learning, where the agent strives to maximise cumulative rewards over time. In this context, the RL agent's consistent improvement in the Total Reward indicates a successful learning trajectory. It implies that, despite the incorporation of kick latency rewards and other additional complexities, the agent adapts and refines its strategies, leading to more favourable outcomes in terms of the overall reward. This positive trend serves as a key validation of the training paradigm's effectiveness, showcasing the RL agent's ability to not only navigate the dynamic RoboCup environment but also to make decisions that result in increasingly positive cumulative rewards. The upward trajectory in the Total Reward reinforces the notion that the introduced rewards and training methodologies contribute synergistically to the agent's proficiency, enhancing its overall performance.

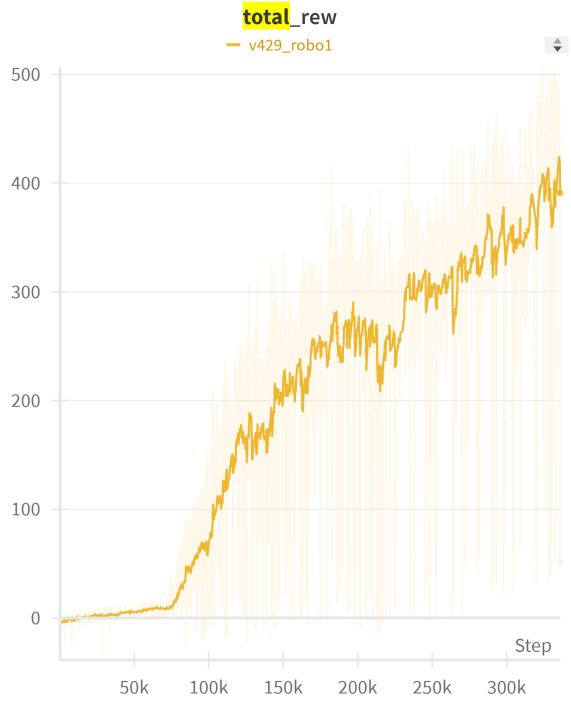


Figure 5.5: The Total Reward with Kick Latency

The delay in the RL model's ability to start learning and getting rewarded for reducing latency is intrinsic to the nature of reinforcement learning. Early in the training process, the agent explores a wide range of actions and their consequences, which often leads to suboptimal and random behaviours. This exploration phase is crucial as it allows the agent to gather diverse experiences and understand the environment's dynamics. Over time, as the agent receives feedback through rewards and penalties, it starts to identify patterns and adjust its actions to maximise the cumulative reward. The initial phase, where rewards remain low or constant, reflects this period of exploration and

trial-and-error learning. It takes a significant number of steps before the agent's learning algorithm can effectively discern the actions that lead to desired outcomes, such as reduced kick latency. This gradual improvement is a hallmark of RL, where substantial training is often required before noticeable progress is made in the targeted behaviours.

5.3.3 Impact of Latency Reduction: Effect on Kicking Performance and Overall Behaviour

The training scenario also aimed to answer the second research question concerning the impact of reducing decision-making latency on kicking performance and the resulting behaviour of the RL agent. Figures depicting the Max Distance Reward, Total Reward and Euclidean Distance Reward manifest upward trends throughout training.

The significance of these trends lies in the following considerations:

- **Enhanced Kicking Distance**: The persistent increase in the Max Distance Reward affirms that, despite latency considerations, the RL agent consistently refines its kicking strategy to achieve longer distances. This implies a successful adaptation to reduced decision-making time.
- **Holistic Performance**: The rising trend in the Total Reward graph indicates that, even with the added complexity of kick latency rewards, the RL agent adeptly balances various performance aspects. This holistic improvement suggests that the agent can optimise its behaviour across multiple dimensions.
- **Precision in Ball Placement**: The upward trend in the Euclidean Distance Reward signifies the RL agent's growing precision in placing the ball accurately concerning the target as shown in Figure 5.6. This is crucial not only for distance but for achieving targeted and precise kicks.

In conclusion, the upward trends in max distance reward, total reward and Euclidean Distance Reward underscore the robustness of the RL agent in adapting its kicking behaviour. These trends are indicative of successful training outcomes, addressing both the primary optimisation objective and the nuanced exploration of latency reduction's impact.

5.4 Solving for Kick Latency: A Comprehensive Analysis

5.4.1 Kick Latency Reward Dynamics

Addressing the pivotal challenge of kick latency, the training scenario strategically introduced the Kick Latency Reward. This reward, depicted in Figure 5.7, initially remained constant at zero, representing a penalty for the constant latency observed in Figure 5.8. However, around 150K steps, a gradual increase in the reward became evident, coinciding with a downward trend in latency observed in Figure 5.8. This increase in the kick latency reward signifies a notable evolution, highlighting the RL agent's ability to

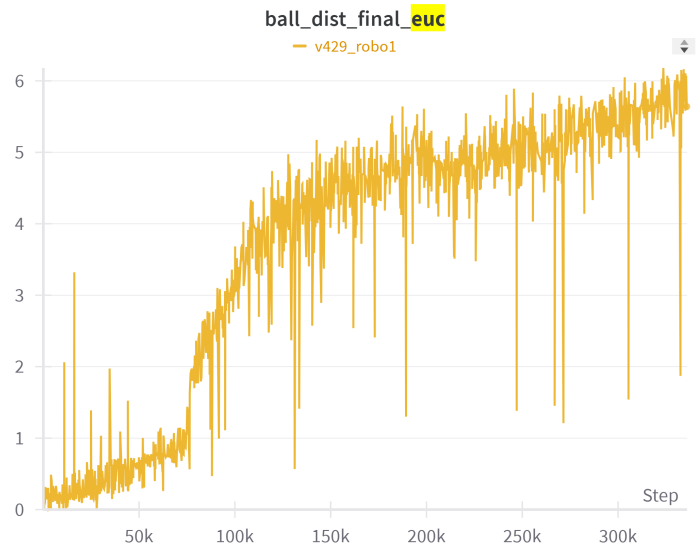


Figure 5.6: Ball Distance Final Euclidean (Euc) Reward without kick Latency

adapt and optimise decision-making latency effectively. It demonstrates a sophisticated learning mechanism within the RL agent, wherein the agent learns to reduce latency through successive training episodes, as indicated by the rising trend of the kick latency reward.

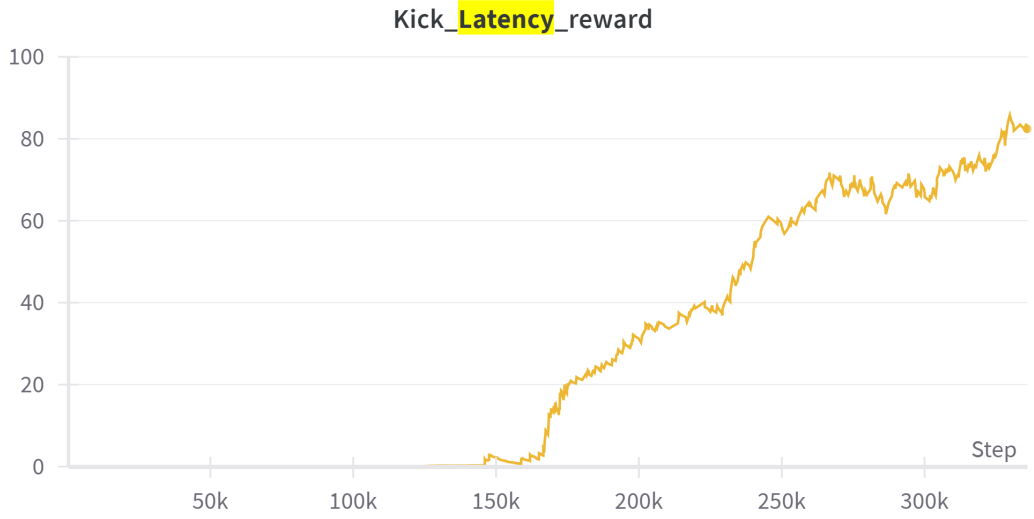


Figure 5.7: The kick Latency reward

5.4.2 Latency Trends

Complementing the analysis of the kick latency reward, Figure 5.8 illustrates the reduction of latency throughout the training process. Initially constant at 0.6, latency began decreasing around step 122K, gradually approaching 0.1. This substantial reduction underscores the effectiveness of the kick latency reward in mitigating decision-making delays. In contrast, Figure 5.9 shows latency remaining constant at 0.6 throughout the training process.

The stark difference from the decreasing trend observed with the Kick Latency Reward underscores the pivotal role played by this reward mechanism. The introduction of the Kick Latency Reward not only prevented latency from fluctuating around certain values but induced a consistent reduction, affirming its effectiveness in promoting swift and timely decision-making during kicking actions. This comparative analysis reinforces the significance of the Kick Latency Reward in fostering more responsive and efficient behaviour in RL-driven agents, a tangible and consistent trend towards more efficient decision-making.

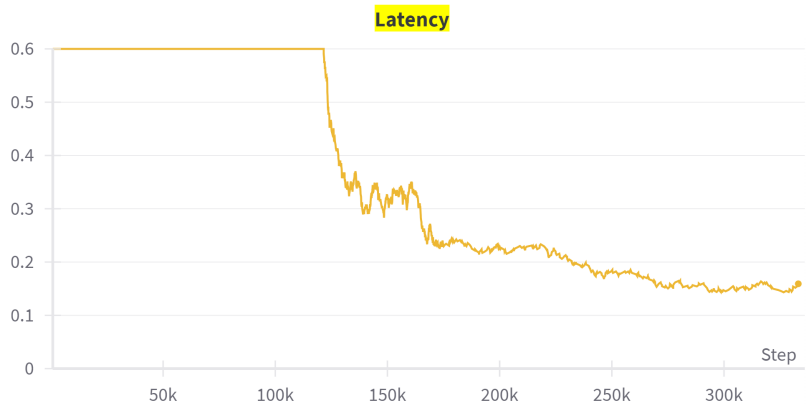


Figure 5.8: Kick Latency with the Kick Latency reward

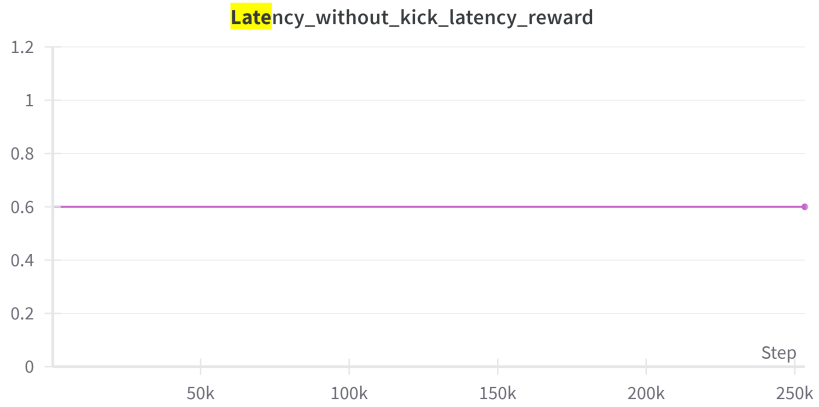


Figure 5.9: Kick Latency without kick latency reward

The reduction of kick latency directly enhances the overall kick and ball distance due to several interconnected factors. Primarily, latency reduction implies that the robotic agent can make quicker and more precise decisions about when and how to execute a kick. When decision-making latency is minimized, the agent can time its kicks more accurately, ensuring that the optimal force and angle are applied to the ball. This precise timing is crucial for achieving maximum power transfer from the robot to the ball, resulting in longer kicks. Additionally, a reduced latency in executing kicks allows for smoother and more coordinated movements, preventing any delays or interruptions that might otherwise dissipate the energy intended for the kick. Consequently, the force applied to the ball is more consistent and efficient, propelling the ball further. The combination of timely, powerful kicks and the efficient transfer of energy leads to an overall improvement in kick performance, as reflected in increased ball distance. This underscores the importance of latency reduction not only for the speed of actions but also for the effective application of physical forces in robotic soccer simulations.

5.4.3 Approximate KL Divergence

Approximate KL Divergence hovering around zero, with minimal fluctuations and rare outliers, as depicted in Figure 5.10, indicates the promising training dynamics of the Proximal Policy Optimization (PPO) algorithm. This stability suggests effective policy adaptation, crucial for maintaining conservative updates close to the existing policy.The small fluctuations observed suggest that the agent is able to adapt its policy effectively without encountering major divergences, leading to a stable training trajectory. Such behaviour aligns with smooth policy transitions and underscores the model's stability and robust learning process. The ability to maintain KL Divergence close to zero reflects a balanced and reliable learning paradigm, successfully navigating the delicate equilibrium between exploration and exploitation.

5.4.4 Policy Gradient Loss

The stability observed in the policy gradient loss, illustrated in Figure 5.11, ranging consistently between -0.05 and 0, indicates the effective adaptation of the Proximal Policy Optimization (PPO) algorithm to the expanded reward structure. This stability reflects the optimization of the model's policy parameters, suggesting that the agent fine-tunes its strategy efficiently without encountering significant fluctuations. Maintaining a narrow range between -0.05 and 0 implies subtle updates to the policy, preventing drastic changes that might result in policy divergence. Such controlled adaptation aligns with the PPO algorithm's goal of making cautious policy updates to ensure stable learning.

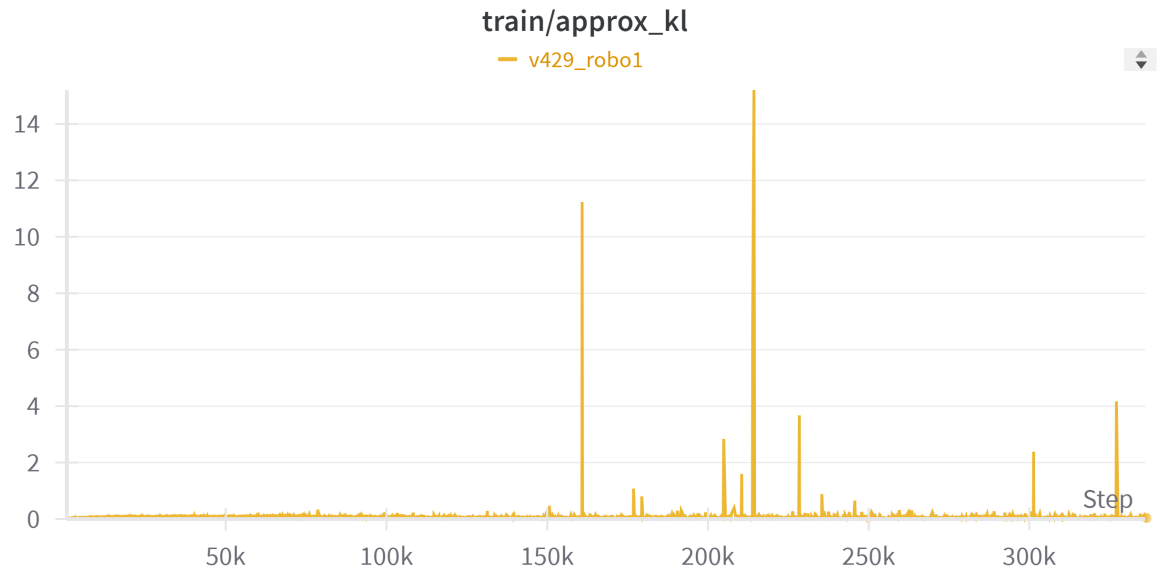


Figure 5.10: Approximate KL Divergence

5.4.5 Synergistic Impact on Kicking Proficiency

The collective impact of integrated rewards is vividly portrayed in the Max Distance Reward, Total Reward and Euclidean Distance Reward figures. The amalgamation of these rewards, each with its nuanced focus, synergistically contributes to the overall proficiency of the RL agent. This holistic approach resonates with real-world applicability, mirroring the success of historical strategies employed by the RoboCup team.

The Euclidean Distance Reward now calculates the Euclidean distance between the agent and the ball at the end of the kicking action. This metric offers a comprehensive assessment of spatial dimensions, considering both horizontal and vertical components of the kicking action. The upward trend in the corresponding Figure 5.3 signifies positive progress in the RL agent's ability to consistently position the ball closer to the desired target. Notably, the consistent and upward trends in these figures validate the RL agent's adept adaptation to the reward structure. The achieved synergy by incorporating diverse aspects of kicking behaviour into the learning process is underscored, marking a substantial step forward in developing a well-rounded and adaptable learning trajectory for the RL agent.

5.4.6 Reinforcement of Desired Latency Reduction Behaviour

The consistent increase in the Kick Latency Reward, coupled with the adept maximisation of rewards by the RL agent, serves as a reinforcement mechanism for the reinforcement learning process. Beyond mere numerical increments, these rewards emphasise the desired behaviour of minimising latency and optimising kicking performance. This highlights the efficacy of the chosen reward structure in steering the learning process toward the intended objectives. The success in solving for kick latency is evident not

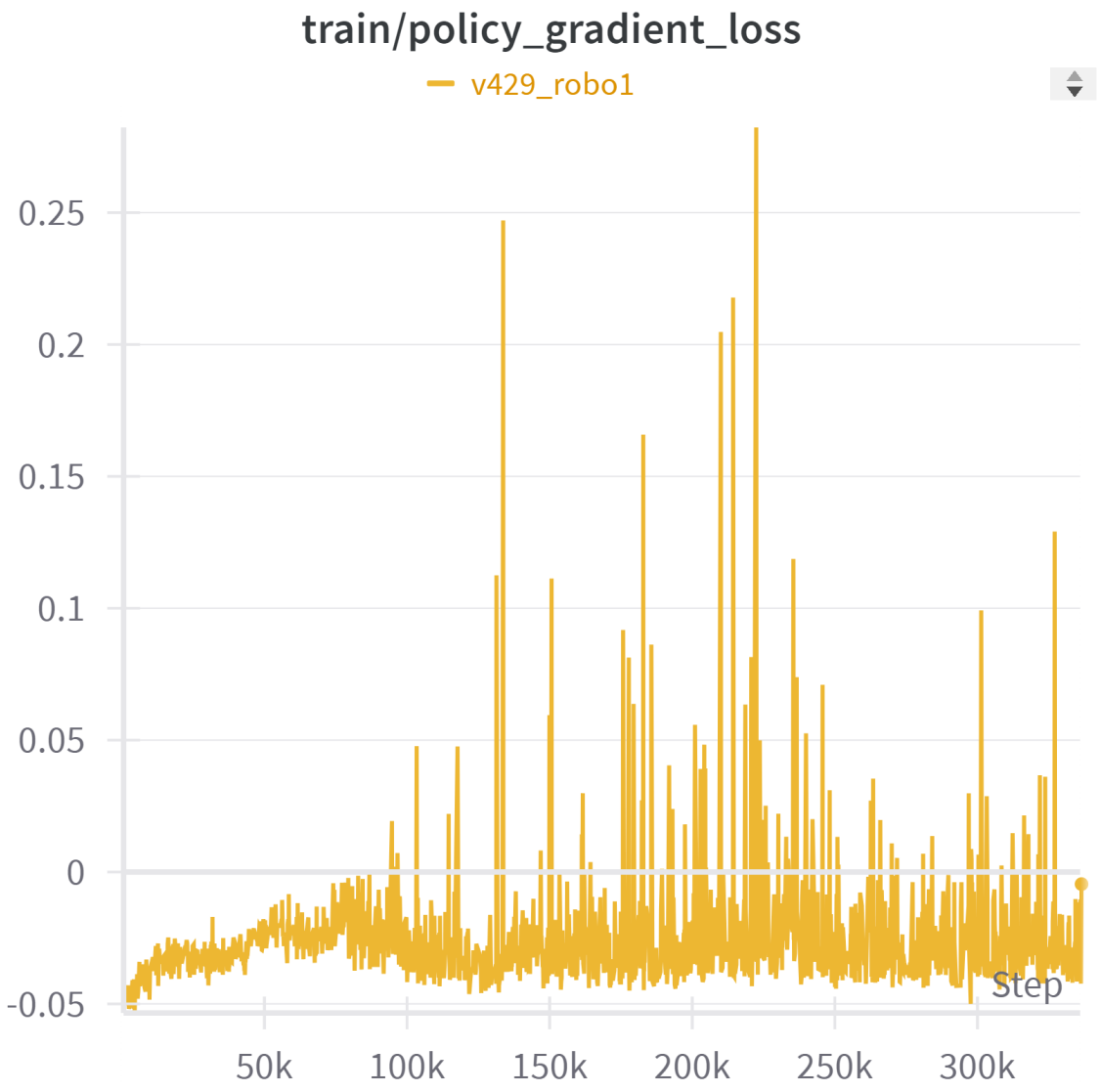


Figure 5.11: Policy Gradient Loss

only in the decreasing trend of latency but also in the maximisation of the associated reward. This comprehensive approach, incorporating a spectrum of rewards, showcases the adaptability and effectiveness of RL agents in dynamic and competitive environments such as the RoboCup simulation.

5.5 Conclusion

In conclusion, our investigation into reinforcement learning scenarios for enhancing kicking performance in the RoboCup simulation has yielded promising outcomes. Scenario 1, employing Proximal Policy Optimisation (PPO), demonstrated the adaptability of the RL agent, showcased by the positive trends in essential performance metrics such as max distance reward and total reward. Building upon this success, Scenario 2 which introduced a pivotal element of kick latency reduction. The remarkable ascent of the Kick Latency Reward, coupled with the integration of historically successful additional rewards, exemplifies the effective reduction of decision-making latency and the creation of a comprehensive reinforcement landscape, aligning with proven RoboCup strategies. Model evaluation, a crucial aspect of our study, delved into the adaptability and stability of the learned policies. Analyses of value loss, approximate KL divergence, policy gradient loss and learning rate provided a nuanced understanding of the RL agent's behaviour. These insights, complemented by visual representations, contribute to a comprehensive comparison across scenarios, facilitating a holistic assessment of the RL agent's capabilities in the challenging domain of RoboCup kicking.

Chapter 6

Conclusion

6.1 Introduction

In the pursuit of enhancing the performance of RoboCup simulation agents, this dissertation has undertaken a comprehensive exploration of reinforcement learning techniques, particularly focusing on the Proximal Policy Optimisation (PPO) algorithm. The research aimed to optimise kicking behaviour in the simulated environment, addressing critical challenges such as decision-making latency. Through meticulous experimentation and analysis, the study has uncovered valuable insights into the dynamics of training scenarios, the impact of rewards and the effectiveness of kick latency reduction. The findings provide a solid foundation for understanding the complexities involved in training RL agents for soccer-playing tasks.

6.2 Research Findings

The research findings reveal the efficacy of employing PPO in refining kicking behaviour, elucidating the nuances of various rewards and the pivotal role they play in the learning process. Notably, the analysis of kick latency reduction showcases the successful mitigation of decision-making latency, a crucial factor in real-time scenarios. The investigation into different reward components, such as max distance, total reward and Euclidean distance, demonstrates a positive learning trajectory, signifying the adaptability and proficiency of RL agents in acquiring diverse kicking skills. The exploration of clip fraction, explained variance and other evaluative metrics provides further insights into the stability, adaptability and learning patterns of the trained agents.

6.3 Future Research

As we conclude this dissertation, it opens avenues for future research in the realm of RoboCup simulation and reinforcement learning. One promising avenue is the exploration of diverse RL models to address specific aspects of soccer-playing tasks. For instance, employing different models to optimise long-distance kicks with accuracy, improve dribbling skills or enhance the goalkeeping capabilities of agents would be a fruit-

ful direction. The investigation into varied RL architectures could contribute to a more comprehensive skill set for the RoboCup team, potentially leading to more strategic and versatile game-play. Additionally, further research could delve into the integration of multi-agent systems, simulating team dynamics and cooperation, to achieve a more holistic approach to RoboCup competitions.

6.4 Conclusion

In conclusion, this dissertation underscores the potential of reinforcement learning, particularly PPO, in advancing the capabilities of RoboCup simulation agents. The detailed analysis of rewards, kick latency reduction and evaluative metrics provides a robust understanding of the learning dynamics. The insights gained not only contribute to the specific domain of RoboCup but also lay the groundwork for broader applications of RL in complex, real-world scenarios. As we look ahead, the suggested avenues for future research promise to further elevate the performance of RL agents, making them more adept and adaptable in soccer-playing tasks. This research journey marks a significant stride in the ongoing quest to push the boundaries of artificial intelligence in competitive and dynamic environments.

References

[Abdolmaleki *et al.* 2016] Abbas Abdolmaleki, David Simões, Nuno Lau, Luis Paulo Reis, and Gerhard Neumann. Learning a humanoid kick with controlled distance. In *Robot World Cup*, pages 45–57. Springer, 2016.

[Barrett *et al.* 2010] Samuel Barrett, Katie Genter, Todd Hester, Michael Quinlan, and Peter Stone. Controlled kicking under uncertainty. In *The Fifth Workshop on Humanoid Soccer Robots at Humanoids*. Citeseer, 2010.

[da Silva 2019] Tiago Rafael Ferreira da Silva. Humanoid low-level skills using machine learning for robocup. 2019.

[de Almeida Martins 2023] Henrique Manuel Neiva de Almeida Martins. Fcportugal-machine learning for a flexible kicking robotic soccer skill. 2023.

[de Sousa Pereira 2020] Ventura de Sousa Pereira. Fcportugal-multi-robot action learning. 2020.

[Depinet *et al.* 2014] Mike Depinet, Patrick MacAlpine, and Peter Stone. Keyframe sampling, optimization, and behavior integration: Towards long-distance kicking in the robocup 3d simulation league. In *Robot Soccer World Cup*, pages 571–582. Springer, 2014.

[Hämäläinen *et al.* 2020] Perttu Hämäläinen, Amin Babadi, Xiaoxiao Ma, and Jaakko Lehtinen. Ppo-cma: Proximal policy optimization with covariance matrix adaptation. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020.

[Lau 2023] Nuno Lau. Fc portugal: Robocup 2022 3d simulation league and technical challenge champions. *RoboCup 2022:: Robot World Cup XXV*, 13561:313, 2023.

[MacAlpine *et al.* 2015] Patrick MacAlpine, Josiah Hanna, Jason Liang, and Peter Stone. Ut austin villa: Robocup 2015 3d simulation league competition and technical challenges champions. In *Robot Soccer World Cup*, pages 118–131. Springer, 2015.

[Melo and Máximo 2019] Luckeciano Carvalho Melo and Marcos Ricardo Omena Albuquerque Máximo. Learning humanoid robot running skills through proximal policy optimization. In *2019 Latin american robotics symposium (LARS), 2019 Brazilian symposium on robotics (SBR) and 2019 workshop on robotics in education (WRE)*, pages 37–42. IEEE, 2019.

[Reis 2023] Luis Paulo Reis. Coordination and machine learning in multi-robot systems: Applications in robotic soccer. *arXiv preprint arXiv:2312.16273*, 2023.

[Schulman *et al.* 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[Schulman *et al.* 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Sousa 2021] Bruno Miguel da Silva Barbosa Sousa. Fcportugal-machine learning for creating robotic soccer setplays. 2021.

[Sutton and Barto 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Teh 2012] Belinda Teh. *Dynamic Omnidirectional Kicks on Humanoid Robots*. PhD thesis, Honours Thesis, The University of New South Wales, 2012.

[Teixeira *et al.* 2020] Henrique Teixeira, Tiago Silva, Miguel Abreu, and Luís Paulo Reis. Humanoid robot kick in motion ability for playing robotic soccer. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 34–39. IEEE, 2020.

[Xu and Vatankhah 2013] Yuan Xu and Hedayat Vatankhah. Simspark: An open source robot simulator developed by the robocup community. In *Robot Soccer World Cup*, pages 632–639. Springer, 2013.