A comparison between deep reinforcement learning methods applied to the control of a mobile robot in a Very Small Size Soccer environment

Natã Ismael Schmitt
Computer Science
Federal University of Santa Maria
Av. Roraima - Camobi, Santa Maria
email: nischmitt@inf.ufsm.br

Bruno Silva de Castro
Control and Automation Engineering
Federal University of Santa Maria
Av. Roraima - Camobi, Santa Maria
ORCID: 0009-0006-4696-435X

Anselmo Rafael Cukla
Control and Automation Engineering
Federal University of Santa Maria
Av. Roraima - Camobi, Santa Maria
email: anselmo.cukla@ufsm.br

Gabriel Tarnowski
Mechatronics Engineering department
Universidad Nacional de Misiones
Obera, Misiones, Argentina
email: gabriel.tarnowski@fio.unam.edu.ar

Leonardo R. Emmendörfer
Department of Electrical Energy Processing
Federal University of Santa Maria
Av. Roraima - Camobi, Santa Maria
email: leonardo.emmendorfer@ufsm.br

Thiago A. N. De Andrade
Department of Statistics
Federal University of Santa Maria
Av. Roraima - Camobi, Santa Maria
email: thiagoan.andrade@gmail.com

***Abstract*—Reinforcement Learning is a field of Artificial Intelligence that studies the development of algorithms that learn to make decisions optimally. In this research, we push the boundaries of knowledge by exploring deep learning techniques for robot control in soccer games. Our approach considers a controlled environment, with the IEEE Very Small Size Soccer robotic competition as a reference. The research's main contribution is comparing the Soft Actor-Critic (SAC) and Deep Deterministic Policy Gradient (DDPG) control algorithms evaluated in a virtual environment with dynamically altered game complexity. Specifically, those algorithms are used to model the actions of a player acting as an attacker in the robotic competition. Changes in the virtual environment consider the Curriculum Learning method, which enables a gradual and effective learning curve during training. The results underscore the SAC algorithm's superiority over DDPG in this specific context, highlighting its greater effectiveness and adaptability. From a broader perspective, the findings of this research contribute to a better understanding of how deep learning techniques can be optimized for applications in competitive robotics.**

I. Introduction

Reinforcement Learning (RL) is probably the area of machine learning that attracts the most fascination among non-experts. This is because research in this area is often associated with robot training [1]. The peak popularization of this field of knowledge occurred in 2013 when the British company DeepMind built an algorithm capable of learning to play Atari games (https://atari.com/) without prior knowledge. It is a much broader field whose origins of study and research date back to the 50' [2].

In short, RL is the field of Artificial Intelligence that studies the development of algorithms that learn to make optimized decisions based on a process of trial and error under a mechanism of rewards and punishments. Through a virtual environment with controlled parameters, the algorithm (agent) performs specific actions to achieve a specified objective. A reward function is used to assign a numerical value, which tells the agent whether its action contributed to or hindered the pursuit of the goal. In this way, the algorithm learns which actions, in which states, contribute to a positive reward, always seeking to maximize the gain of future rewards. Deep Reinforcement Learning (DRL) [3] is an essential variation of RL, which employs the use of deep neural networks to solve this problem.

Over the years, several works applying DRL techniques in different contexts have been published. A search on *Google Scholar* for the term *Deep Reinforcement Learning* returns more than a million articles and other academic contributions, highlighting the extent of research in this domain. One of the applications in emerging studies includes using DRL techniques to control robots in football matches within the scope of the *IEEE Very Small Size Soccer* (IEEE VSSS) competition.

IEEE VSSS robot football is a type of robotics competition that combines several technologies that cooperate so that the robots can score against the opponent, based on the rules of the Brazilian Robotics Competition (CBR) [4].

A game comprises two teams of 3 differential wheel robots, which are externally controlled. Its dimensions are a maximum of 8 cm wide, 8 cm long, and 8 cm high. The arena is 1.5 meters long and 1.3 meters wide. An orange golf ball with well-defined weight and characteristics is used for the match. The VSSS is widely used as a source of research, bringing scientific advances to robotics and bringing together groups that aim to achieve the best robots and control methods. In Brazil, the main attraction event is the *Latin American Robotics Competition* (LARC), with the VSSS and many other types of competitions taking place annually.

The control of the robots is done via a computer, which, connected to the camera (positioned above the field), monitors the movements of these agents in a match. The captured image is then processed with computer vision algorithms, extracting variables such as the ball's position and direction and speed of all robots on the field, among others. This information is used by the game control algorithm, which will make decisions based on the current state. Differential wheel robots can be controlled through the speeds of each of their wheels so that the algorithm can generate the final speed of the wheels of each robot and, consequently, control its movements.

This research aims to use DRL techniques to study robot control in football matches within the scope of the IEEE VSSS competition. More specifically, training *Soft Actor-Critic* (SAC) [5] and *Deep Deterministic Policy Gradient* (DDPG) [6] control algorithms, which were evaluated in a virtual environment with dynamically changing game complexity. For this, the open source simulator `RSoccer` [7] was used through its implementation as a library of the `Python` programming language.

In addition to this introduction, this paper is structured as outlined below. Section II lists some essential works related to the research topic. Section III presents methodological details that guided the achievement of the results of this research, including theoretical aspects about reinforcement learning, a computational framework centered on the open source simulator `RSoccer` [7], the definition of learning premises and rewards, among others. Detailed analysis of the scenarios under investigation is presented in Section IV. Section V addresses final considerations and future proposals.

II. Related works

From both theoretical and applied perspectives, researchers have dedicated themselves to advancing the field of DRL. In this Section, mention is made of some exponent works, emphasizing those that have direct links with the findings presented in this study or are closely related to the lines of investigation currently under development. The main objective of the connections is to provide the fundamental theoretical framework for the scope of this research and to anticipate future directions within the scope of DRL applied to robotic control.

The research carried out by [8] presents a comprehensive overview of the use of RL in robotic manipulation tasks, mapping existing knowledge gaps and suggesting future directions for investigation. The authors undertook a systematic review, employing a broad sweep of scientific literature, focusing on electronic databases recognized for disseminating essential results, such as *IEEE Xplore* and *Google Scholar*. In this process, 42 articles were selected and analyzed, published since 2013, based on specific inclusion parameters.

In a similar perspective, [9] perform a comprehensive literature review, covering publications in the period 2015 and 2022, to understand and apply DRL algorithms. These authors direct their focus to significant advances in the use of RL in robotics, highlighting the challenges linked to the transition from strategies learned in simulations to real-world applications. This study included the analysis of more than 150 academic works, using research platforms such as *Google Scholar*, *IEEE Xplore*, and *ArXiv*

In a seminal paper, [10] introduced the reinforcement learning methodology known as *Curriculum Learning*. This approach, which prioritizes progressive training in steps of increasing complexity, is inspired by the effective learning processes observed in humans and animals. Our research uses the contribution of [10] to adapt the training environment in the scenarios considered.

Recently, in [11], they conducted a study in which two types of neural networks are compared to control the formation of robots in a VSSS environment considering the DDPG methods and *Double Deep Q-Network* (DDQN) [12]. The function of neural networks is to introduce a controller that acts as a type of technician who coordinates the role of the robots, be it an attacker, defender, or goalkeeper during the match.

The work presented by [13] explores the application of two neural network algorithms, specifically DDPG and SAC, to move differential robots in a virtual environment containing obstacles. The main focus of this study is to analyze the robot's movement on the playing field, where the objective is for it to move between specific points on the map without colliding with obstacles. Our investigation is heavily inspired by the contributions of [13], similarly exploring the effectiveness of these algorithms in the specific context of robotic navigation.

In addition to the previously mentioned articles, we would like to highlight other research at the frontier of knowledge in robotic computing. We refer to the works of [14], [15], [16], [17], [18] and [19]. These studies span a wide variety of subareas in robotics research, including multi-robot trajectory planning, path planning, and smoothing, as well as decentralized and probabilistic methods for avoiding collisions between multiple robots. They also explore using reinforcement learning in football competitions with small and very small robots. This work and its future developments have benefited significantly from the contributions of these researchers.

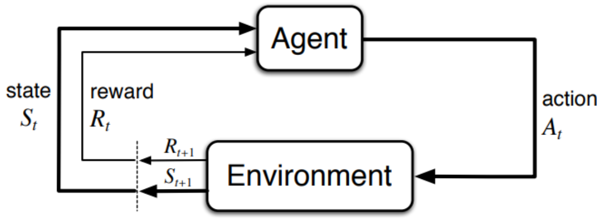
III. Methodology

A. Reinforcement Learning

This work follows the data flow generally applied in Reinforcement Learning, as shown in Figure 1 [20]. Here, the environment and the agent are presented. The agent receives initial data from the environment, such as state S_t and R_t, and generates an action A_t, which is executed in the environment, generating new states and rewards S_{t+1} and R_{t+1}, which are processed again by the agent, completing the cycle.

This process is aligned with the Markov Decision Process (MDP) properties, where each decision is based only on the current state, thus respecting Markov properties. In a simplified way, the MDP can be represented as MDP(S, A, P, R), where S represents the set of states of the environment, A is the set of actions available to the agent, P is the transition probability function and R is the reward function ([21], [22] and [1]).

Figure 1. Execution of Reinforcement Learning algorithms



B. `rSoccer` Overview

To carry out the Reinforcement Learning tests used, the library `rSoccer` [7] was used, an environment with simulated physics aimed at training VSSS games, based in the library `GYM` [23], widely used in the literature to evaluate the performance of reinforcement learning algorithms. The library is available in the `Python` programming language, and its environment was modified to allow the player to be trained with progressive difficulties and personalized rewards. Furthermore, the `rSoccer` library supports the rendering of images for real-time visualization of environments through a GUI (Graphical User Interface), simplifying the observation of game progress in the environment and allowing an empirical evaluation of the behavior of agents.

The VSSS environment includes an out-of-the-box reward function that evaluates changes between the previous and current state of the environment and assigns a numerical value to indicate the importance of the agent's action in achieving the goal. This reward comprises three functions: a ball potential function, a movement function, and an energy expenditure function. The ball potential function compares the previous and current position of the ball to determine whether it has approached the enemy goal, returning a positive value proportional to the ball's displacement in the direction of the goal and calculated by Equation 1. The movement function provides feedback on the agent's action, considering the agent's approach to the ball, encouraging the robot's initial learning, and is defined by Equation 2. Lastly, the reward is given to agent's energy expenditure to avoid unnecessary movements and discard non-optimal paths to reach the ball. This reward is given by the Equation 3.

$$R_p = \frac{bp_t - bp_{t-dt}}{dt} \tag{1}$$

$$\vec{RB} = ||(B_x - R_x, B_y - R_y)|| \tag{2}$$

$$E_r = -\sum_{p=0}^{E}(|w_0| + |w_1|) \tag{3}$$

With the three functions described, the final reward function is achieved by including weights for each function so the user can balance and contribute more appropriate weights to the algorithms. In Equation 4 W_b, W_m and W_e are introduced, representing weights for ball potential, agent movement, and energy spent, respectively. Note that if the goal indication for the reward function is given by adding or subtracting G_p, which is normally a higher value, the algorithm learns the importance of the goals.

$$R = \begin{cases} W_b \cdot R_p + W_m \cdot M_r + W_e \cdot E_r + G_p & \text{Goal} \\ W_b \cdot R_p + W_m \cdot M_r + W_e \cdot E_r & \text{Didn't goal} \\ W_b \cdot R_p + W_m \cdot M_r + W_e \cdot E_r - G_p & \text{Opponent goal} \end{cases} \tag{4}$$

C. Attacker Simulation Ambient

This subsection covers the theoretical aspects that makeup one of this research's main contributions: the modification of the learning environment, emphasizing progressive training and stages of increasing complexity.

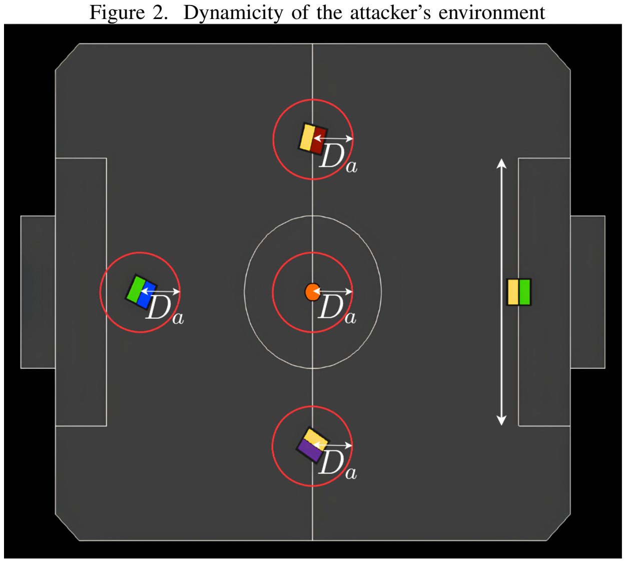
Aiming to take advantage of Curriculum Learning techniques [10], discrete environments were replaced by a single environment, which will have its difficulty dynamically changed, allowing the agent smooth training, as its learning curve will dictate the difficulty of the environment. Therefore, we adopted a number ranging between 0 and 1 for the difficulty criterion, with 0 being the least complexity and 1 being the most significant difficulty. A difficulty of 0 means the environment with the robots in a fixed starting position and immobile. Based on the average rewards from the previous 100 episodes, we define the current difficulty of the environment to calculate D_a, whose effect is described in Figure 2. From this, Equation 5 describes the calculation of D_a, representing the difficulty of the environment, and which values D_a admits. In this equation, $r_m = \frac{\max_r R}{\lambda}$, R is the set of averages among the agent's last 100 rewards, which we select the maximum value of R for use in this equation. λ is an arbitrary number representing the reward function's maximum value, which was empirically defined as 450 from previous training. Note that in 5, the minimum value of D_a is 0.1.

$$D_a = \begin{Bmatrix} 0.1 & r_m \leq 0.6 \\ 0.25 & 0.6 < r_m \leq 0.7 \\ 0.55 & 0.7 < r_m \leq 0.8 \\ 1 & r_m > 0.8 \end{Bmatrix} \tag{5}$$

D. Agent Observations

As inspiration from the work of [13], the standard `rSoccer` observations in this striker training environment have been replaced by those described in Table I. Observations that previously reported absolute coordinates of x and y now report distance and angle as a way of representing the observed positions always in relation to the agent so that the neural network has data that occurs more frequently because it is centered on the agent's position, and no longer globally in the field, thus facilitating the generalization of the network. It was also decided to reduce the amount of data reported about other robots, such as linear and angular velocities, since this information is not of great relevance to the robot's objective, in addition to being proof that few observations are needed for the agent satisfies the requirements of the environment. A visual

Figure 2. Dynamicity of the attacker's environment



description for a better understanding of the observations can be seen in Figure 3.

Observations
Distance between agent and ball
Agent to ball angle
Distance between agent and opponent goal
Agent angle to opponent goal
Distance between agent and M_i
Agent angle for M_i
Sine of the agent angle
Cosine of the agent's angle
Linear velocity in X of the ball
Linear velocity in Y of the ball
Linear speed in X of the agent
Linear velocity in Y of the agent
Agent angular velocity

Table I
OBSERVATIONS FOR THE ATTACKING AGENT USED IN THIS WORK

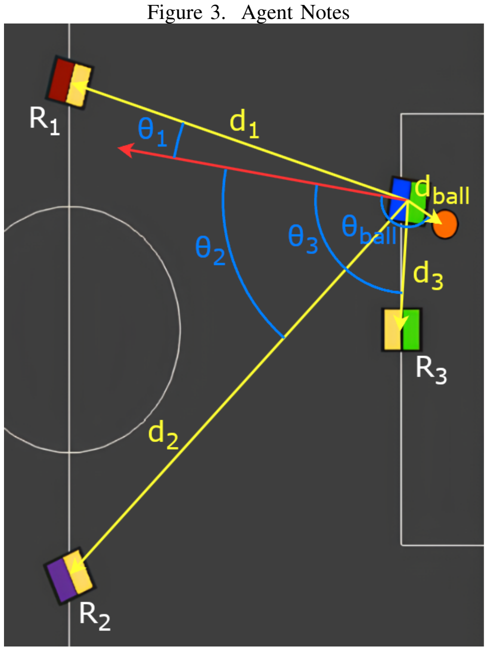
E. Hyperparameters

The hyperparameter scheme adopted in this research was based on the work of [13]. The authors made the source codes available in a repository on GitHub, facilitating their application in different contexts by other researchers. Although the main focus of the study is not robotic programming, its contributions were precious for developing the programming environment presented in this work. Table II below presents the hyperparameters considered in this work.

Hyperparameter	Description	Used Value
a	Learning rate	2.5e-5
Batch size	Number of samples per training batch	256
Experience memory	Replay buffer size	5e5
σ	Noise level	0.1
γ	Future discount factor	0.99
τ	Soft update rate for target networks	0.999

Table II
SOFT ACTOR CRITIC HYPERPARAMETERS

Figure 3. Agent Notes



F. Reward Function

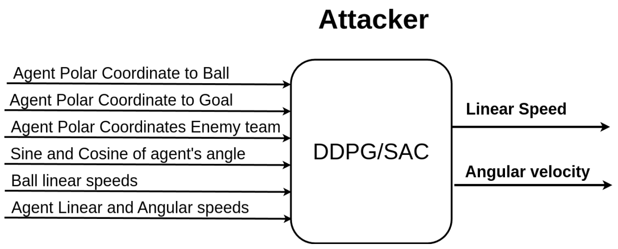
This work considers a modification of the reward function to improve agent learning. The first change was in the weights so that the agent better prioritizes the occurrence of goals. An increase in the weight of the ball gradient was also made to make the robot learn more quickly that the ball must follow a particular path. The energy was increased to make it more obvious to the agent when choosing a shorter route. The weights used for training are found in Table III, whose values were explained previously.

Observation	Quantity
W_b	2
W_m	0.25
W_e	1e-3
G_p	500

Table III
WEIGHTS USED IN THE REWARD FUNCTION OF THE ATTACKING ENVIRONMENT

In Figure 4, we see the inputs and outputs for the attacking agent, which come from the data presented in Table I. Each of the inputs is part of the state S of the environment at time t, and their outputs make up the action A at time t.

Figure 4. Attacking agent input and output diagram



Attacker
Agent Polar Coordinate to Ball
Agent Polar Coordinate to Goal
Agent Polar Coordinates Enemy team
Sine and Cosine of agent's angle
Ball linear speeds
Agent Linear and Angular speeds
DDPG/SAC
Linear Speed
Angular velocity

IV. Results and discussions

In this section, the numerical evaluation results are presented for the study that compared the performance of two DRL algorithms, namely SAC and DDPG, applied to the control of a mobile robot in the context of IEEE VSSS.

It is observed that SAC surpassed DDPG in terms of obtaining rewards and effectiveness in carrying out the proposed tasks. As can be seen in Figures 5 and 6, comparing the rewards obtained by the SAC and DDPG methods, respectively.

The graph in Figure 5 highlights four regions with different colors, each with a different value for D_a, representing training steps with different difficulty levels. Such complexity levels are determined as described in Equation 5. Generally, it can be inferred that as D_a increases, the performance metric stabilizes, suggesting that a larger D_a could be associated with a more stable or effective learning policy in the long run.

Similarly, the image in Figure 6 shows the performance obtained by the DDPG model, considering different learning stages. Contrary to what was found for the SAC algorithm, performance is highly affected, decreasing as complexity increases. In the graphs presented in both Figure 5 and Figure 6, the X axis indicates the training steps. In contrast, the Y axis represents the Simple Moving Average (SMA) calculated from the last 100 episodes.

Figure 5. SAC algorithm training graph

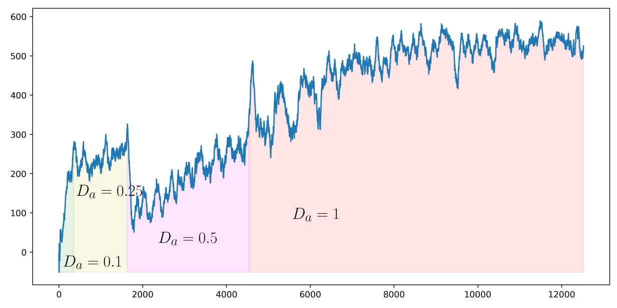
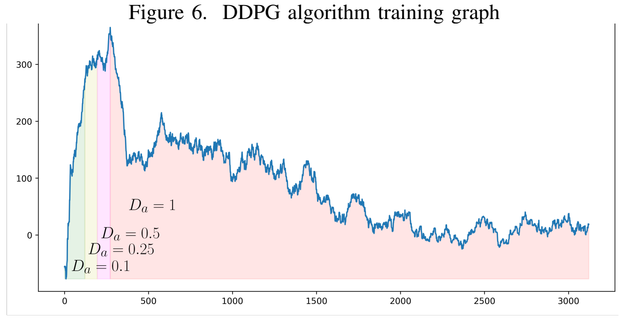


Figure 6. DDPG algorithm training graph



The experimental results showed that SAC achieved significantly superior performance in the attacker control task. Not only was SAC able to score more effectively against virtual opponents, but it also showed greater adaptability and generalization ability to different game configurations and obstacles. On the other hand, DDPG, although it showed good initial performance, faced difficulties adapting to incremental changes in the complexity of the environment, resulting in less consistent performance.

The superiority of SAC can be attributed to its ability to explore the space of actions more efficiently and states thanks to its maximum entropy policy. This allows for a more comprehensive exploration of the game environment, leading to more robust and adaptive strategies. Furthermore, the SAC framework, which emphasizes reward-based learning and error minimization, has proven to be particularly suitable for the dynamic and unpredictable environment of IEEE VSSS.

The results suggest that applying *Curriculum Learning* techniques may have played a crucial role in improving the learning process, facilitating the progressive adaptation of robots to increasingly complex challenges. This hypothesis, however, requires a more detailed investigation for confirmation. This simulates a more natural and effective learning environment, where agents master simple tasks before progressing to more difficult ones.

The results also suggest the importance of a well-defined reward function in reinforcement learning. The functionality of dynamically adjusting the environment's difficulty and reward based on the agent's performance proved to be fundamental in encouraging the robot to learn effective control and navigation strategies.

V. Conclusion

This paper applies Deep Reinforcement Learning (DRL) algorithms to control robots in the *IEEE Very Small Size Soccer* (IEEE VSSS) robot soccer modality. Specifically, *Soft Actor-Critic* (SAC) and *Deep Deterministic Policy Gradient* (DDPG) control algorithms were considered and modified to act as attackers, with performances evaluated in a virtual environment with dynamically changed game complexity. The learning mechanism takes into account *Curriculum Learning* [10] methodology. This methodology emphasizes progressive training in stages of increasing complexity, taking inspiration from the efficient learning processes observed in humans and animals. In summary, the main findings of the research are listed below:

- DRL algorithms demonstrated effectiveness in controlling robots within the scope of the IEEE VSSS competition.
- As an attacker, the SAC algorithm not only outperformed DDPG, but it did so with a level of efficiency that demands respect, as evidenced by its superior accumulated rewards.
- The DDPG algorithm exhibited a learning curve that began with progressive improvement, followed by a decline as the complexity of the environment increased. In contrast, despite requiring a more extended training period, the SAC algorithm demonstrated superior performance under environmental conditions of varying complexity.
- The superiority of SAC can be primarily attributed to its maximum entropy policy, which promotes a more effective exploration of the action space.

- Considering that the primary success metric defined was the ability to score goals in all episodes with the fewest possible attempts, it was found that none of the algorithms achieved absolute success. Therefore, the need for future algorithm improvements is identified to achieve this goal.

The results listed above suggest a broad spectrum of future contributions. Among the main lines of investigation underway by the proponents of this research, the following stand out:

- Explore specific algorithms to improve goalkeeper training in robotics, considering different reinforcement learning techniques.
- Expand the scope of performance comparison between different algorithms, such as SAC and DDPG, to include new approaches, focusing on their capabilities as attacking players.
- Investigate *path planning* algorithms that integrate reinforcement learning knowledge for better real-time navigation between obstacles, fulfilling the objectives of robot football.
- Deepen the study of the maximum entropy policy used by SAC to understand how this approach can be explored or improved in different competitive robotics configurations.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html.

[2] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd. O'Reilly Media, Nov. 2022, p. 861, ISBN: 978-1098125974.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, *Playing atari with deep reinforcement learning*, 2013. arXiv: 1312.5602 [cs.LG].

[4] *Very small size soccer rules*, https://www.cbrobotica.org/wp-content/uploads/2021/05/vssRules3x321.pdf, Acesso: 18/12/2023.

[5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, *Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor*, 2018. arXiv: 1801.01290 [cs.LG].

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, *et al.*, *Continuous control with deep reinforcement learning*, 2019. arXiv: 1509.02971 [cs.LG].

[7] F. B. Martins, M. G. Machado, H. F. Bassani, P. H. M. Braga, and E. S. Barros, "Rsoccer: A framework for studying reinforcement learning in small and very small size robot soccer", in *RoboCup 2021: Robot World Cup XXIV*, R. Alami, J. Biswas, M. Cakmak, and O. Obst, Eds., Cham: Springer International Publishing, 2022, pp. 165–176, ISBN: 978-3-030-98682-7.

[8] H. Nguyen and H. La, "Review of deep reinforcement learning for robot manipulation", in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 590–595. DOI: 10.1109/IRC.2019.00120.

[9] D. Han, B. Mulyana, V. Stankovic, and S. Cheng, "A survey on deep reinforcement learning algorithms for robotic manipulation", *Sensors*, vol. 23, no. 7, p. 3762, 2023. DOI: 10.3390/s23073762.

[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning", in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 41–48, ISBN: 9781605585161. DOI: 10.1145/1553374.1553380. [Online]. Available: https://doi.org/10.1145/1553374.1553380.

[11] C. H. C. Pena, M. G. Machado, M. S. Barros, *et al.*, "An analysis of reinforcement learning applied to coach task in ieee very small size soccer", in *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, 2020, pp. 1–6. DOI: 10.1109/LARS/SBR/WRE51543.2020.9307069.

[12] M.-W. Han and P. Kopacek, "Neural networks for the control of soccer robots", in *ISIE'2000. Proceedings of the 2000 IEEE International Symposium on Industrial Electronics (Cat. No.00TH8543)*, vol. 2, 2000, 571–575 vol.2. DOI: 10.1109/ISIE.2000.930360.

[13] J. C. de Jesus, V. A. Kich, A. H. Kolling, R. B. Grando, M. A. d. S. L. Cuadros, and D. F. T. Gamarra, "Soft actor-critic for navigation of mobile robots", in *Journal of Intelligent & Robotic Systems*, vol. 102, 2021. DOI: 10.1007/s10846-021-01367-5.

[14] A. Zolanvari, M. Shirazi, and M. Menhaj, "A q-learning approach for controlling a robotic goalkeeper during penalty procedure", in *2019 II International Congress on Science and Engineering*, Hamburg-Germany, 2019, pp. 1–12.

[15] H. Ali, G. Xiong, H. Wu, B. Hu, Z. Shen, and H. Bai, "Multi-robot path planning and trajectory smoothing", in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2020, pp. 685–690.

[16] W. Luo, W. Sun, and A. Kapoor, "Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates", in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 372–383.

[17] H. Zhu, B. Brito, and J. Alonso-Mora, "Decentralized probabilistic multi-robot collision avoidance using buffered uncertainty-aware voronoi cells", *Autonomous Robots*, vol. 46, pp. 401–420, 2022. DOI: 10.1007/s10514-021-10029-2.

[18] F. Martins, M. Machado, H. Bassani, P. Braga, and E. Barros, "Rsoccer: A framework for studying reinforcement learning in small and very small size robot soccer", in *RoboCup 2021: Robot World Cup XXIV*,

R. Alami, J. Biswas, M. Cakmak, and O. Obst, Eds., ser. Lecture Notes in Computer Science, vol. 13132, Cham: Springer, 2022. DOI: 10.1007/978-3-030-98682-7_14.

[19] L. da Silva Costa and F. Tonidandel, “Multi-robot path planning with safety based control applied to the small size league robots”, in *RoboCup 2023: Robot World Cup XXVI*, C. Buche, A. Rossi, M. Simões, and U. Visser, Eds., ser. Lecture Notes in Computer Science, vol. 14140, Cham: Springer, 2024. DOI: 10.1007/978-3-031-55015-7_7.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018, ISBN: 0262039249.

[21] M. Ororbia and G. Warn, “Structural design synthesis through a sequential decision process”, in *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ser. 40th Computers and Information in Engineering Conference (CIE), Available: https://doi.org/10.1115/DETC2020-22647, ASME, Virtual, Online, 2020, V009T09A045. DOI: 10 . 1115 / DETC2020-22647.

[22] R. BELLMAN, “A markovian decision process”, *Journal of Mathematics and Mechanics*, vol. 6, no. 5, pp. 679–684, 1957, ISSN: 00959057, 19435274. [Online]. Available: http://www.jstor.org/stable/24900506 (visited on 05/15/2024).

[23] G. Brockman, V. Cheung, L. Pettersson, *et al.*, *Openai gym*, 2016. arXiv: 1606.01540 [cs.LG].