A Baseline Approach for Goalkeeper Strategy using Sarsa with Tile Coding on the Half Field Offense Environment

Victor G. Ferreira Barbosa
Computer Engineering
UNIVASF
Juazeiro, Brazil
victorgfb80@gmail.com

Rosalvo Ferreira de Oliveira Neto
Computer Engineering
UNIVASF
Juazeiro, Brazil
rosalvo.oliveira@univasf.edu.br

Roberto V. L. Gomes Rodrigues
Computer Engineering
UNIVASF
Juazeiro, Brazil
hrodeberthus@gmail.com

***Abstract*—Much research in RoboCup 2D Soccer Simulation has used the Half Field Offense (HFO) environment. This work proposes a baseline approach for goalkeeper strategy using Reinforcement Learning on HFO. The proposed approach uses Sarsa with eligibility traces and Tile Coding for the discretization of state variables. Two comparative studies were conducted to validate the proposed baseline. First, a comparative study between the Agent2D's goalkeeper strategy and a random decision strategy was performed. The second comparative study verified the performance of the proposed approach against a random decision strategy. Wilcoxon's Signed-Rank test was used for measuring the statistical significance of performance differences. Experiments showed that the Agent2D's goalkeeper strategy is inferior to a random decision, and the proposed baseline delivers a performance superior to a random decision strategy with a confidence level of 95%.**

***Index Terms*—RoboCup, Reinforcement Learning, Sarsa, Tile Coding**

I. Introduction

The simulated environments have improved research in the area of Robotics and Artificial Intelligence. An example of these environments is the RoboCup 2D Simulated Soccer League (RoboCupSoccer). Much research in RoboCup 2D Soccer Simulation has used the Half Field Offense (HFO) environment [1]. It has been used in studies of attack [2] [3] and defense strategies [2]. The two main advantages of HFO over the RoboCupSoccer Server with a full match are: 1) Simulation speed, since it allows the execution of "matches" with only half the field, the experiments can be carried out more quickly and focused on the objective of the study and 2) easy to capture events and variables, because it has libraries that allow the researcher to directly access information such as goal, the capture of the ball by the defense, among other relevant information during the experiment. The HFO Environment simulates matches within the RoboCup 2D soccer server. To perform an experiment in the HFO, it is necessary to specify the number of Agents (Players) and their profiles, if offense or defense. It is also necessary to specify whether a player follows the base team strategy or a developed strategy for evaluation. The HFO Environment uses the Agent2D as the base team, the 2012 RoboCup 2D champion team [4]. Fig. 1 shows the HFO architecture.

The authors would like to thank CNPq for the financial support.

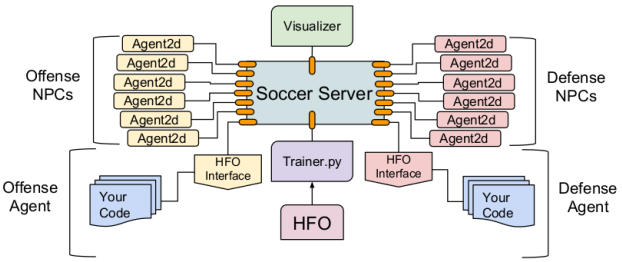

Fig. 1. HFO architecture [1].

A standard methodology is to compare the performance of a new approach or algorithms with base-teams performance [5]. This study analyzes the performance of the Agent2D's goalkeeper strategy in HFO and proposes a new strategy to be used as a baseline. The soccer simulation represents a stochastic environment. Thus reinforcement learning algorithms can be used to build a goalkeeper strategy. This type of learning can be classified as an intermediary between supervised and unsupervised learning [6]. This work presents an efficient goalkeeper strategy.

The remaining of this paper is organized as follows. Section 2 presents the problem definition. Section 3 presents some background on Reinforcement Learning that were used to build the proposed approach, which is described in Section 4. Section 5 shows the experimental methodology. Section 6 presents the experimental results. Section 7 presents related works. Finally, Section 8 concludes this paper and proposes future works.

II. Problem Definition

Fig. 2 shows the position of the goalkeeper and the players during an attack. As can be seen from Fig. 2, there is a significant number of possibilities of actions that a goalkeeper can take. This study considered that a goalkeeper could take two

actions: Move or Intercept. The move is the displacement of the goalkeeper forward, backward, or sideways. The intercept is going towards the ball to try to catch it. The Agent2D base team provides these actions. These actions have been considered in studies in this area [7]. The problem of defining what action the goalkeeper should take during a game of robot football can be defined as a Reinforcement Learning problem.

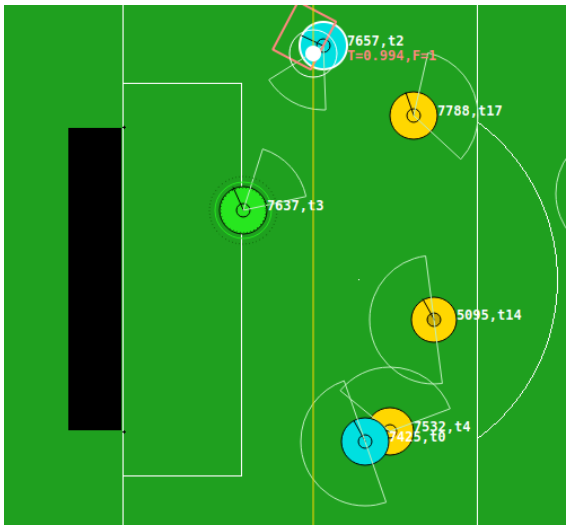

Fig. 2. Position of the goalkeeper and the players during an attack. The goalkeeper is green. The defense agents are yellow, and the offense agents are blue.

III. Definitions and Background

A. Reinforcement Learning

An intelligent agent is an autonomous entity which acts upon an environment making decisions for achieving goals [8]. It perceives the environment through its sensors, and the agent manipulates this perception through the concept of state, which represents a configuration of the environment at a given time. According to Russell and Norvig [9], in stochastic environments and with a Markov transition model, the decision process is called the Markov Decision Process (MDP), composed of four elements [10]:

- S is a set of states;
- A is a set of actions;
- $P^a_{ss'}$ is a transition function = P(S'|S, A), representing the probability of reaching state S' if action A is applied while in state S;
- R is a reward function.

In MDP, the agent learns through a process of trial and error, using a reward function (R) to guide the agent with, as feedback, a reward value. The agent's behavior is modeled through the interaction between the agent and the environment. At each time step t, the agent is in a state (S) and maps its perceptions of the environment to performs an action (A) that will move the agent to a new state S'. The mapping of state S to state S' is done by the transition function ($P^a_{ss'}$). The mapping of all actions to states is called the Policy (π) and defines how the agent behaves. The MDP's objective is to build an optimal Policy that maximizes expected cumulative reward [8]. The cumulated reward or return G_t at a given time step is expressed as the sum of future rewards, as seen in (1). However, it is essential to take into account the time weight of the rewards. For example, the most recent rewards should have a more significant influence on the cumulative reward. For this, a discount factor (γ) in G_t is introduced, as seen in (2). Equation (3) shows the standard cumulative reward using the discount factor. Therefore, the agent's goal is to maximize (3).

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} \tag{1}$$

$$G_t = R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} ... \tag{2}$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{3}$$

To maximize the accumulated reward, the agent needs to measure the quality associated with each state, so every state must have a utility value. This utility value is always associated with a policy. Two functions can be used to calculate the utility of a state given a policy (π): the state-value function v_π and the action-value function q_π.

1) State-value function: The state-value function for policy (π) is the expected return from starting from state s at time t. It is denoted as $v_\pi(s)$, as seen in (4) and (5).

$$v_\pi(s) = E_\pi\left[G_t \mid S_t = s\right] \tag{4}$$

$$= E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]. \tag{5}$$

2) Action-value function: The action-value function for policy (π) is the expected return from starting from state s at time t taking action a, as seen in (6) and (7). It is denoted as $q_\pi(s, a)$, which is called Q-values.

$$q_\pi(s,a) = E_\pi\left[G_t \mid S_t = s, A_t = a\right] \tag{6}$$

$$= E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]. \tag{7}$$

The agent's final goal in MPD is to find a policy (π) that maximizes one of the two values: $v_\pi(s)$ or $q_\pi(s, a)$. The optimal state-value $v_*(s)$ is defined by $\max_\pi v_\pi(s)$, and the optimal action-value $q_*(s, a)$ is $\max_\pi q_\pi(s, a)$. Thus, we can write the optimal $v_*(s)$ and $q_*(s, a)$ by (8) and (9). They are known as the Bellman equation.

$$v_*(s) = (max_a R^a_s + \gamma \sum_{s' \in S} P^a_{ss'} v_*(s')) \tag{8}$$

$$q_*(s,a) = R^a_s + \gamma \sum_{s' \in S} P^a_{ss'} max_{a'} q_*(s', a') \tag{9}$$

Reinforcement Learning (RL) can solve Markov Decision Processes [8]. In the literature, there are several approaches to building policies using Reinforcement Learning. Among the most popular, we can highlight the Q-Learning and Sarsa algorithms described in the next section.

B. Q-Learning and Sarsa

Q-learning is a reinforcement learning technique used for learning the optimal policy in a Markov Decision Process. It is a value iteration algorithm. The Bellman equation is the basis of the value iteration algorithms for solving MDP [11]. If there are N possible states, there will be N Bellman's equation, one for each state. The goal of Q-learning is to find the optimal policy by learning the optimal Q-values for each state-action pair $q_*(s, a)$ instead of learning the transition model $P^a_{ss'}$. The Q-values for each state-action pair is stored in a table called a Q-table. The dimensions of the table are the number of actions by the number of states. Fig. 3 shows the Q-Learning algorithm, where the α parameter is the learning rate.

Q-Learning algorithm

Algorithm parameters: $\alpha \in (0, 1]$, For each state s and action a, initialize randomly Q(s,a)
foreach *episode* **do**
  Initialize S;
  foreach *step of episode* **do**
    Choose A from S using policy derived from Q (e.g., ε-greedy);
    Take action A, observe R, S';
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$;
    $S \leftarrow S'$;
  end foreach
end foreach

Fig. 3. Q-Learning algorithm.

The main difference between Sarsa and Q-Learning is that the Sarsa algorithm chooses the current action and the next action using the same policy, so the update rule is $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', a) - Q(S, A)]$.

C. Tile Coding

In a complex environment with continuous variables, MDPs have infinite combinations of states and actions, so it is necessary to carry out the discretization of continuous variables. However, there is a lack of generalization when the discretization process is carried out [12]. The success of RL in such cases critically depends on effective discretization or so-called quantization approach [8]. There are a variety of methods with this goal; among the best known, we can highlight Tile Coding, which is a linear discretization method. Tile Coding provides a balance between representational power, computational cost, and ease of use [13]. It works as following.

The space of each variable is divided into partitions called tiling. Each tiling is represented by an interval [ε I, ε I + 1], where ε represents the offset value, and i is the number of tiling. For example, suppose a scalar variable X [0, 1], and that we are using 2 tilings with ε = 0.15. The range of each tiling would be [0, 1] and [0.15, 1.15]. Each tiling is made up of subpartitions called tiles. Each tile is a receptive field for one binary feature. The number of tiles is defined by their width (w), representing the resolution of the discretization.

The main advantage of Tile Coding with multiple tilings is the increase of the generalization power [14]. To illustrate this feature, let's consider Fig. 4. If we discretize the variable X without multiple tilings, the two inputs p (0.2) and q (0.3) would be coded as (1, 0, 0, 0) and (0, 1, 0, 0) respectively, and they do not share any information. However, with two tilings and using the offset value ε =0.15 , the two inputs p and q would be coded as (1, 0, 0, 0, 1, 0, 0, 0) and (0, 1, 0, 0, 1, 0, 0, 0) respectively. In this way, the two entries would share a bit of information.

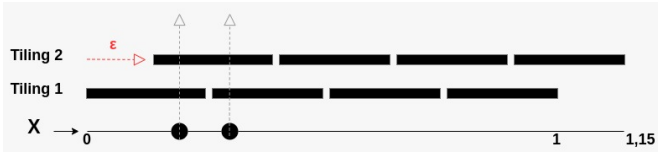

Fig. 4. Tile Coding.

IV. Proposed Approach

This section describes the proposed approach to be used as a baseline for the goalkeeper strategy. The approach uses Sarsa, one of the most popular reinforcement learning algorithms. There are two significant challenges when developing a solution using Sarsa. The first is world modeling, which consists of identifying which states and actions should be considered. The second is the way of discretizing continuous variables. This work proposes modeling of the world using a reduced number of variables and Tile Coding with only one tile per tiling to discretize the variables.

A. State Variables

In RL, the states are represented implicitly by the set of state variables [15]. In this study, the agent state is composed of the following state variables in a cartesian plane:

- Ball position (X, Y);
- Goalkeeper position (X, Y);
- Opponent position (X, Y).

Fig. 5 shows the state variables in the field of RoboCup Simulation 2D. All state variables were normalized in the range [-1, 1] follow the scale showed in Fig. 6.

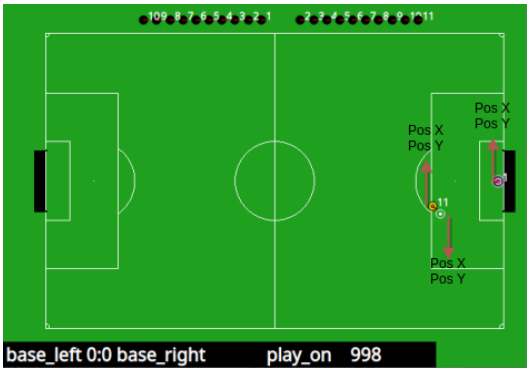

Fig. 5. State variables in the field of RoboCup Simulation 2D.

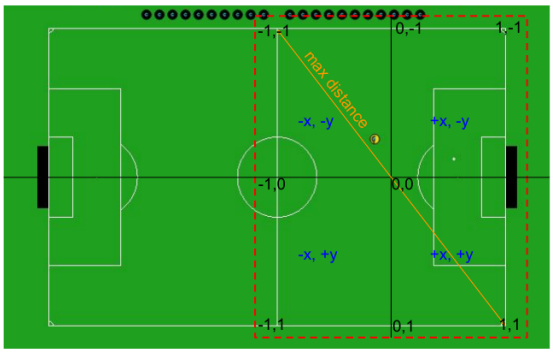

Fig. 6. State variables normalized.

B. Actions

The goalkeeper follows the general Agent2D policy, one of the most popular base times of RoboCup 2D. In our modeling, the agents may take two actions:

- Move, which is an automated action that moves the agent towards the best position guided by the Helios strategy.
- Intercept, which is an automated action that moves the agent towards the ball position.

C. Tile Coding

The state variables were discretized by Tile Coding. In this study, we used the tiling organization based on [14] description, in which use only a tile per tiling with overlap, since any function representable with an N-tiling organization is also representable with a single-tiling organization [14]. The codification is defined as follows: four tilings and one tile per tiling. The width of each tile was equal to 0.7. Fig. 7 shows the overlap between tiling. Each tile was coded as a bit, so each variable was discretized as a 4-bit binary number.

D. Sarsa

The proposed approach used Sarsa with Eligibility Traces [8], a technique widely used in RL to decrease learning time by memorizing newly visited state/action pairs. The stopping criterion used in training was the convergence in the average cumulative success rate in 10 episodes. Table I shows the used rewards.

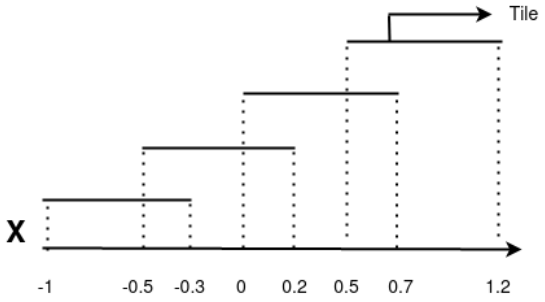

Fig. 7. Tile Coding setting used.

TABLE I
USED REWARDS

Reward	Event
+1	for defense
-1	for goal scored
0	otherwise

V. EXPERIMENTAL METHODOLOGY

This section presents the experimental methodology. Two comparative studies were conducted to validate the proposed baseline. The first compared the performance of the Agent2D's goalkeeper strategy with an arbitrary decision, in which it randomly chooses between Intercept or Move. The second comparative study verified the performance of the proposed approach against a random decision strategy. In order to compare the performance differences, the non-parametric Wilcoxon's Signed-Rank Test was used [16]. The two studies were evaluated in 30 experiments, and each experiment consisted of 50 episodes. Fig. 8 shows the experimental methodology. The components of the experimental methodology will be described below.

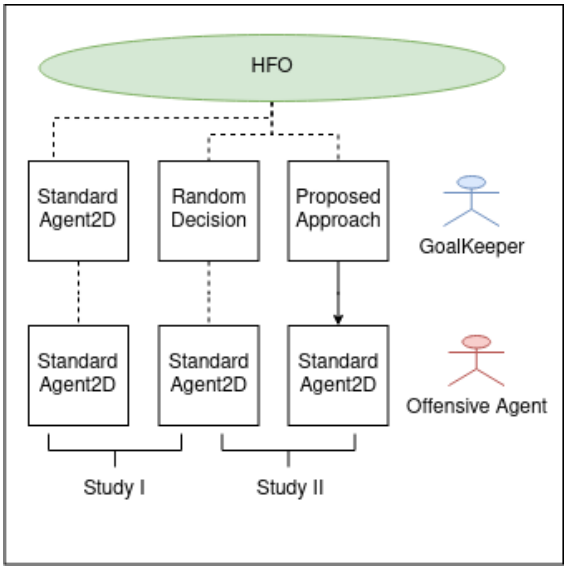

Fig. 8. Experimental Methodology.

TABLE II
DECISIONS IN THE FLOWCHART

Decision	Conditional statements
A	Is the ball in the penalty area?
B	Can the interception take place outside the penalty area?
C	Is there a teammate close to the ball and there is no close opponent?
D	Can the opponent arrive faster than the agent?
E	Is the opponent closer to the ball than the agent?
F	Is there a teammate closest to the ball who can reach it?
G	Is the ball too close?
H	Can the opponent arrive faster than the agent?

A. HFO (Half Field Offense)

The experiments were carried out in HFO. In the attack, only one player was used following the strategy of the base team. The defense was composed only by the goalkeeper. Three goalkeeper strategies were used: 1) the Agent2D's standard goalkeeper strategy, 2) a random decision, and 3) the proposed approach. It is essential to highlight that the standard Agent 2D player positioning strategy file was used in the experiments. This file contains the information needed to execute the Move and Intercept actions. The use of the standard strategy allows the reproducibility of this study.

B. Performance measurement metrics

The performance evaluation metrics used in both studies was the goalkeeper's success rate in 50 episodes. An episode in the HFO occurs when:

- **Goal.** The offense scored a goal.
- **Captured.** The defense gained control of the ball.
- **Out of Bounds.** The ball left the playfield
- **Out of Time.** No agent has approached the ball in the last 100 timesteps.

The success rate in 50 episodes was calculated as follow: (Captured + Out of Bounds + Out of Time)

C. Base team strategy

Fig. 9 shows the base team's goalkeeper strategy. Table II describes the decisions in the Flowchart.

D. Hypothesis Test

In order to compare the performance differences in these studies, the non-parametric Wilcoxon's Signed-Rank Test was used [16]. The configurations of tests used in this study were:

1) Test Study I:

- Null Hypothesis : $\mu 1 = \mu 2$
- Alternative Hypothesis : $\mu 1 \geq \mu 2$

where

- $\mu 1$ represents the mean of the success rate for Random Decision;
- $\mu 2$ represents the mean of the success rate for the Base Team;

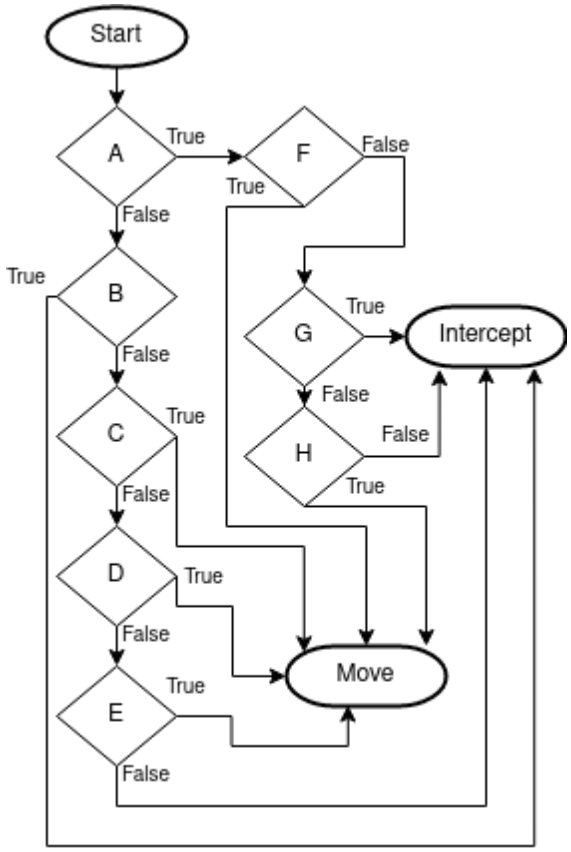

Fig. 9. Base team's goalkeeper strategy.

2) Test Study II:

- Null Hypothesis : $\mu 1 = \mu 2$
- Alternative Hypothesis : $\mu 1 \geq \mu 2$

where

- $\mu 1$ represents the mean of the success rate for Proposed Approach;
- $\mu 2$ represents the mean of the success rate for Random Decision;

E. Setup Experimental

In order to avoid possible problems caused by programs already installed or differences of hardware, all experiments were performed on the same machine with 8Gb RAM memory, 4 CPU cores and Ubuntu 20.04 operating system. Table III shows Sarsa parameters, based on [7].

TABLE III
SARSA PARAMETERS

Parameter	Value	Description
γ	0,9	Discount Factor
α	0,1	Learning Rate

VI. RESULTS

In this section, the results of the two experiments are presented as described in the previous section.

TABLE IV
SUCCESS RATE OF BASE TEAM - STUDY I

AVG	MAX	MIN	STD
8,03 (16,07%)	13 (26,00%)	3 (6,00%)	2,31 (4,62%)

TABLE V
SUCCESS RATE OF RANDOM - STUDY I

AVG	MAX	MIN	STD
9,30 (18,60%)	15 (30,00%)	5 (10,00%)	2,58 (5,15%)

TABLE VI
HYPOTHESIS TEST OF STUDY I

Base Team	Random	Difference	p-value
8,03	9,30	1,27	0,04128

TABLE VII
SUCCESS RATE OF PROPOSED APPROACH - STUDY II

AVG	MAX	MIN	STD
18,73 (37,47%)	27 (54,00%)	11 (22,00%)	3,81 (7,63%)

A. First Study

Fig. 10 shows the box plot for the base team strategy and random decision in the thirty experiments. Each experiment consisted of 50 episodes. As can be seen in Tables IV and V, the strategy's average success rate with a random decision was higher than the Agent2D's base team (9,3 vs. 8,03). Results show that Random Strategy has a higher success rate than the base team approach with a significance of 95% since the p-value is smaller than 0.05, as shown in Table VI. The result indicates that using the Agent2D's standard goalkeeper strategy would underestimate the performance comparison of Reinforcement Learning algorithms. The randomized strategy was better than the baseline strategy because solutions based on rules, such as the baseline strategy, they do not work well in non-deterministic and stochastic environments.

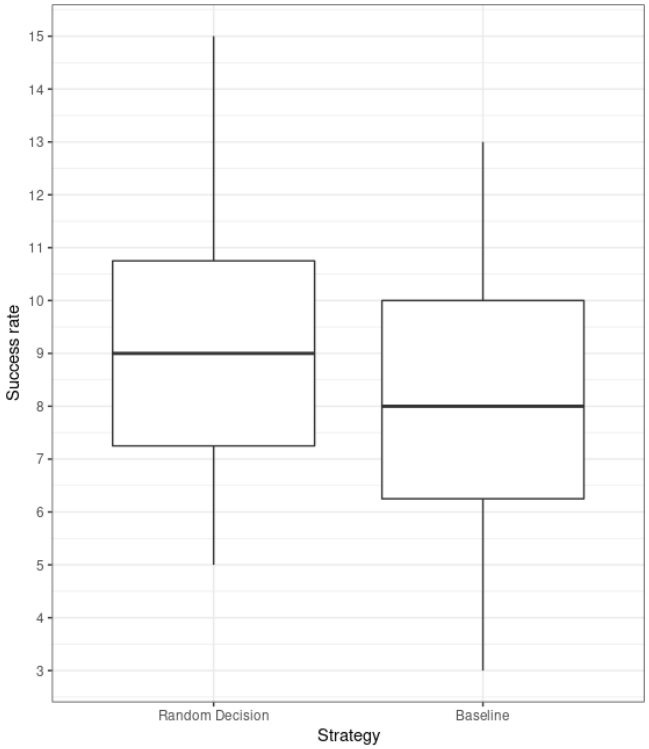

Fig. 10. BoxPlot of First Study. Comparison between the base team strategy and random decision.

B. Second Study

Fig. 11 shows the average cumulative success rate in 10 episodes of the proposed approach throughout the training. As can be seen in Fig. 11, the curve's stabilization ensures the convergence of the algorithm. Fig. 12 shows the box plot for the proposed approach and random decision in the thirty experiments. As shown in Table VII, the strategy's average success rate with the proposed approach was higher than a random decision (18,73 vs. 9,30). Table VIII shows the results of the hypothesis test. Since the p-value is smaller than 0.05, the results indicate that the Proposed Approach has a higher success rate than the Random Decision approach, with a confidence level of 95%. The second experimental study results show that the proposed solution is an efficient Goalkeeper strategy for games in the 2D Simulation League. Thus it is more appropriate to be used as a baseline in research in this area.

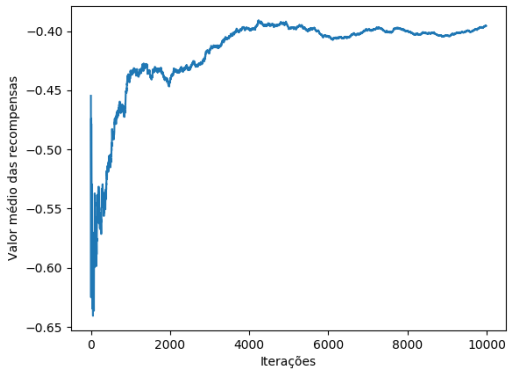

Fig. 11. Average cumulative success rate in 10 episodes of the proposed approach.

VII. RELATED WORKS

There are many types of research on RoboCup. However, according to the literature survey made during the creation of

TABLE VIII
HYPOTHESIS TEST OF STUDY II

Proposed Approach	Random	Difference	p-value
18,73	9,30	9,43	0.0001

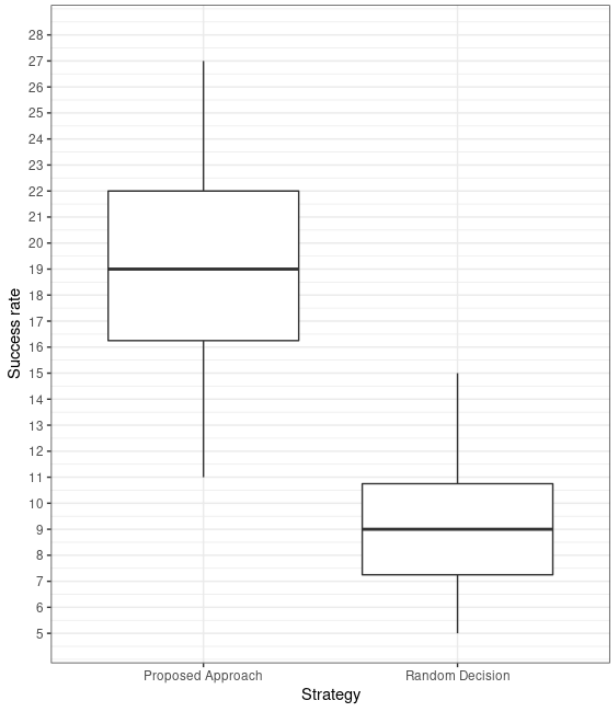

Fig. 12. BoxPlot of Second Study. Comparison between the proposed approach and random decision.

this paper, there are no works that propose a baseline approach for goalkeeper strategy using Sarsa with Tile Coding on HFO. Among researches found it is possible to highlight:

Xiong et al. [17] developed a new passing strategy based on the q-learning algorithm. The study used the Agent2D as the base team. To validate the proposed passing strategy, the authors carried out a comparative study between the team with the new passing strategy and UvA Trilearn, a base team from the University of Amsterdam. Experiments showed that the new passing strategy delivers a performance superior. The results showed that the proposed strategy reached a proportion of success in the number of passing 11,97% superior.

In Silva et al. [18], the authors proposed a data mining solution for defense strategy. The solution was composed of two artificial neural networks. One to predict passing probabilities and the other to predict the opponent's kick. The study used the Expertinos as the base team, a base team from the Federal University of Itajubá, Brazil. To validate the proposed defense strategy, the authors carried out a comparative study between the Expertinos and the ITAndroids. The results showed that the proposed approach reduced the number of goals scored by 17%.

In 2019, Zolanvari et al. [19] proposed a penalty defense strategy for the goalkeeper. The proposed strategy used Q-learning. The study was carried out in RoboCup Small Size soccer robots instead of Simulation 2D. To validate the proposed penalty defense strategy, the authors carried out a comparative study between a simple hard-coded algorithm and their solution based on reinforcement learning. The results extracting from real experiments showed that the RL approach was more successful in solving the problem than the hard-coded algorithm.

In [20], Heusden investigated two penalty defense strategy for the goalkeeper. Both used Deep Q learning [21] applied to discrete state and action spaces. First, he investigated the use of an intermediate rewards function based on the distance between the goalkeeper and a line representing the trajectory of the ball. Second, he investigated the use of Deep Q learning combined with transfer learning. The study used four distinct phases to evaluate transfer learning. Each phase used the information learned in the previous tasks to guide the learning in the current phase. The study was carried out in the HFO Environment. The state-space consisting of five possible bins for distances, and five possible bins for angles were used. The results showed that the first approach had a success rate 49% higher than the second approach. The study also reports that it is not possible to create a penalty defense strategy without using an intermediate reward function or learning transfer.

VIII. Conclusion

This work presented an efficient goalkeeper strategy for the Half Field Offense Environment. The proposed solution is composed of Sarsa with eligibility traces and Tile Coding for the discretization of state variables. This work's main goal was to develop a baseline approach for goalkeeper strategy using Reinforcement Learning on HFO since this simulation environment has served as a benchmark in this research area. This study showed that the Agent2D's goalkeeper strategy has a low success rate on average, around 16%, which makes its comparison with recent Reinforcement Learning algorithms such as Deep Q-Networks and Dueling Double-Deep Q Networks, underestimated. Among the main contributions of this study, the highlights are: 1) it showed that the Agent2D's goalkeeper strategy is inferior to a random decision, and 2) proposed a new baseline approach for goalkeeper strategy. The strategy's average success rate with the proposed approach was higher than a random decision (18,73 vs. 9,30). Thus, the proposed approach is more appropriate to be used as a baseline in research in this area. As future work, we intend to expand this study to build new baselines such as faults, penalties and cornering.

References

[1] M. Hausknecht, P. Mupparaju, S. Subramanian, S. Kalyanakrishnan, and P. Stone, "Half field offense: An environment for multiagent learning and ad hoc teamwork," in *AAMAS Adaptive Learning Agents (ALA) Workshop*, Singapore, May 2016. [Online]. Available: http://www.cs.utexas.edu/users/ai-lab?hausknecht:aamasws16

[2] M. J. Hausknecht and P. Stone, "Deep reinforcement learning in parameterized action space," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.04143

[3] M. Hausknecht and P. Stone, "On-policy vs. off-policy updates for deep reinforcement learning," in *Deep Reinforcement Learning: Frontiers and Challenges, IJCAI Workshop*, July 2016.

[4] H. Akiyama, "Agent2d base code," 2010, [Online; accessed 3-agosto-2020]. [Online]. Available: https://osdn.net/projects/rctools/

[5] J. P. F. Nascimento, R. F. de Oliveira Neto, and L. J. M. Amorim, "An efficient kick strategy for agents in the 2d simulation league," in *8th Brazilian Conference on Intelligent Systems, BRACIS 2019, Salvador, Brazil, October 15-18, 2019*. IEEE, 2019, pp. 461–466. [Online]. Available: https://doi.org/10.1109/BRACIS.2019.00087

[6] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014.

[7] F. L. da Silva, R. Glatt, and A. H. R. Costa, "Simultaneously learning and advising in multiagent reinforcement learning," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, K. Larson, M. Winikoff, S. Das, and E. H. Durfee, Eds. ACM, 2017, pp. 1100–1108. [Online]. Available: http://dl.acm.org/citation.cfm?id=3091280

[8] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2018. [Online]. Available: https://books.google.com.br/books?id=6DKPtQEACAAJ

[9] S. J. Russell and P. Norvig, *Artificial Intelligence: a modern approach*, 3rd ed. Pearson, 2009.

[10] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[11] J. M. Porta, N. A. Vlassis, M. T. J. Spaan, and P. Poupart, "Point-based value iteration for continuous pomdps," *J. Mach. Learn. Res.*, vol. 7, pp. 2329–2367, 2006.

[12] A. A. Sherstov and P. Stone, "Function approximation via tile coding: Automating parameter choice," in *Abstraction, Reformulation and Approximation*, J.-D. Zucker and L. Saitta, Eds. Springer Berlin Heidelberg, 2005, pp. 194–205.

[13] R. S. Sutton, "Generalization in reinforcement learning: Successful examples using sparse coarse coding," in *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, D. S. Touretzky, M. Mozer, and M. E. Hasselmo, Eds. MIT Press, 1995, pp. 1038–1044.

[14] A. A. Sherstov and P. Stone, "Function approximation via tile coding: Automating parameter choice," in *Abstraction, Reformulation and Approximation, 6th International Symposium, SARA 2005, Airth Castle, Scotland, UK, July 26-29, 2005, Proceedings*, ser. Lecture Notes in Computer Science, J. Zucker and L. Saitta, Eds., vol. 3607. Springer, 2005, pp. 194–205. [Online]. Available: https://doi.org/10.1007/11527862_14

[15] "The relation between reinforcement learning parameters and the influence of reinforcement history on choice behavior," *Journal of Mathematical Psychology*, vol. 66, pp. 59–69, 2015.

[16] D. Rey and M. Neuhauser, *Wilcoxon-Signed-Rank Test*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1658–1659.

[17] X. Li, W. Chen, J. Guo, Z. Zhai, and Z. Huang, "A new passing strategy based on q-learning algorithm in robocup," in *International Conference on Computer Science and Software Engineering, CSSE 2008, Volume 1: Artificial Intelligence, December 12-14, 2008, Wuhan, China*. IEEE Computer Society, 2008, pp. 524–527.

[18] F. L. S. Silva, J. P. de Almeida Barbosa, and G. S. Bastos, "2d behavior based soccer team - a neural network approach," in *XXI Congresso Brasileiro de Automática - CBA2016, Vitória - ES, Brasil*, 2016, pp. 1941–1946.

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

[20] R. van Heusden, "Making a robot stop a penalty - using q learning and transfer learning."

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602