BahiaRT 2022: Team Description Paper for RoboCup 3D Soccer Simulation League

Gabriel Sousa, Jadson Nobre, Rafael Fonsêca, Vitor Manoel dos Santos, Wesley Silva, Felipe Mascarenhas, Ana Patricia Magalhães, Jorge Campos, Josemar Rodrigues de Souza, Robson Marinho da Silva, and Marco A C Simões

Bahia State University (UNEB/ACSO), Salvador, BA, Brazil
teambahiart@gmail.com
http://www.acso.uneb.br/bahiart/

Abstract. This paper describes the significant enhancements in team BahiaRT since RoboCup 2021. We describe our work about setplays learning from demonstration. We describe our working for learning new skills, such as kicking and walking. We summarize the projects we still working on at the end of the paper.

1 Introduction

Bahia Robotics Team (BahiaRT) has experienced a continuous evolution in RoboCup competitions in the last years. In the world championship, BahiaRT ranked fifth in 2014, sixth in 2017, and fourth in 2015, 2016, 2018, and 2019. In 2021, the team ranked in the 9th position. We were also Champions(in 2015) and runner-up(2016 and 2021) of the Portuguese RoboCup Open competition in the last years. We won the previous nine editions of the Brazilian Robotics Competition and the previous eight editions of the Latin American Robotics Competition.

The delay in finishing some vital research projects explains the performance decrease in RoboCup 2021. We made the last enhancement on robots' skills more than five years ago. We are now working on kicking optimization using recent state-of-the-art deep learning approaches. In section 2, we present the BahiaRT Gym, a customized environment that allows the use of several reinforcement learning algorithms using the OpenAI Gym framework[3]. We are also finishing a long-term project about setplays learning from demonstration described in section 3. We describe some work in progress in section 4. we plano to integrate Some partial results from these work in progress in the team we will use for the next competition. The expectation is to observe a good performance increase during the RoboCup 2022.

Since 2013, BahiaRT uses a source base code, named FCPBase 2013, as part of an international scientific cooperation partnership with team FC Portugal from Universities of Aveiro and Porto in Portugal.

2 BahiaRT Gym

The BahiaRT Gym is our custom environment created using the OpenAI Gym toolkit[3], which is made for comparing and developing reinforcement learning algorithms. Though the toolkit already provides us with many pre-built environments, none of those was ideal for our work, nor could be used with our server. We decided to create one environment from scratch with the idea to make it useful, not only to our team but to any other team, no matter the coding language used.

In order to do so, we have drawn the process architecture shown in Fig.1, where the gym environment can connect to the server as a monitor, receiving information about many aspects of the simulation, including objects positions, game time, play modes and so on, while also being able to send back trainer commands, which can control some of these aspects. The environment also has a connection to the training agents, sending them the calculated optimized actions that they should execute and receiving a custom message in return. At the moment, we've been using this as a synchronization flag to check if the actions were correctly received, but it can be used in other ways such as sending back to the environment any data generated by the team. An important feature of BahiaRT Gym is the BahiaRT Proxy that we use to connect the agents and the server to each other. It creates two threads for each agent, sending the agent actions and receiving the agent perceptions, as shown in Fig.1. In this way, the proxy is able to intercept any message sent by both the server and the agents, and it is used to relays the agents perceptions to the gym using TCP sockets.

The demonstration environment that everyone will be able to use once we release the BahiaRT Gym is a very simple one, where the agent has to decide whether to stand still, carry the ball or try to kick it to the goal. Although at the time it only uses the ball speed and the distance between it and the agent as observations, this environment is still open for improvements before its release.

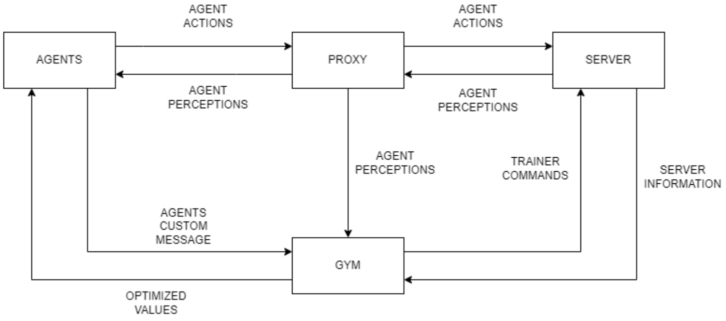


Fig. 1. BahiaRT Gym Process Architecture

3 Strategy: intensive use of setplays

In the last years, we have been designing an effective solution for BahiaRT players using setplays in most situations during a game. We hypothesize that we can extract intuitive knowledge from soccer experts or fans and use it for training a robot soccer team[12].

We decided to use the toolkit provided by team FCPortugal: the Strategy Planner (SPlanner)[4] and the FCPortugal Setplays Framework (FSF)[6]. We have first used SPlanner and FSF when we have designed our kickoff setplay[9]. We have developed both a hand-coded version and a SPlanner-generated version of the kickoff setplay. We noticed that SPlanner and FSF lack some crucial features for the humanoid robot's soccer and, in particular, for 3D Simulation. So we added many resources, such as defensive setplays, opponents to be used as a reference in setplays, new behaviors, etc[10].

The resulting setplays will compose a large dataset that we need to organize in clusters to use considering the real-time requirements of 3D Simulation. We developed a fuzzy clustering strategy to organize the set of setplays into clusters[13].

When the clusters are available, we plan to use a deep reinforcement learning approach, using the BahiaRT Gym (see Section 2). The proposal is to learn a policy to select the best cluster of setplays to be used in any instantaneous situation during the game. When the group is selected, the agents use a Case-based Reasoning (CBR) approach present in the FSF to choose the specific setplay in the cluster. The following subsections describes the modifications we made in the FSF and team BahiaRT to adapt the CBR strategy to this project.

3.1 Using Case-based Reasoning (CBR) for setplays selection

Using setplays intensively also concerns the choice of collective plays and appropriate moments of the match to use them. To guarantee these characteristics, we implement our version of the CBR algorithm.

The CBR is an artificial intelligence approach that solves inferred problems by retrieving and adapting known solutions used in past problems. In CBR, knowledge is represented by cases – structures that store the characteristics of the problem and its solution – and its life cycle is composed of 4 stages [14]:

1. Recovery: when past cases are retrieved and compared to the new problem;
2. Reuse: where the most similar past case solution is transposed to the presented problem;
3. Review: concerns the evaluation of the results obtained with the applied solution;
4. Retention: a process in which new knowledge is produced, through the storage of information produced in the form of a case.

Our implementation of CBR aims to select the most appropriate setplay from a set of predefined setplays and given the characteristics of the match at a given moment. Figure 2 shows the CBR life cycle we use in our team. The input of the CBR system is a game snapshot containing information about the match: opponent's name, game score, ball and player positioning, game play mode, and game time.Some case completion information is also used: data on the success rate and the number of uses of the case in current and past games. The process of recovery and selection of the best solution begins. The comparison of cases with the new problem is made using the global-local principle[2], supported by a system of weights, distributed among the attributes that characterize a case.

After selecting the best case, the setplay used in its solution is applied to the new problem. The application result is analyzed and then a new case is generated and stored in the case base. To improve the intelligence of the CBR, so that the choices of setplays are more assertive in terms of their success rate and the advantage provided to the team, it is necessary to build a solid base of cases. To build this base and validate the CBR implementation, we used a supercomputer, capable of executing several game instances, thus expanding the use of CBR and the knowledge stored in the case base.

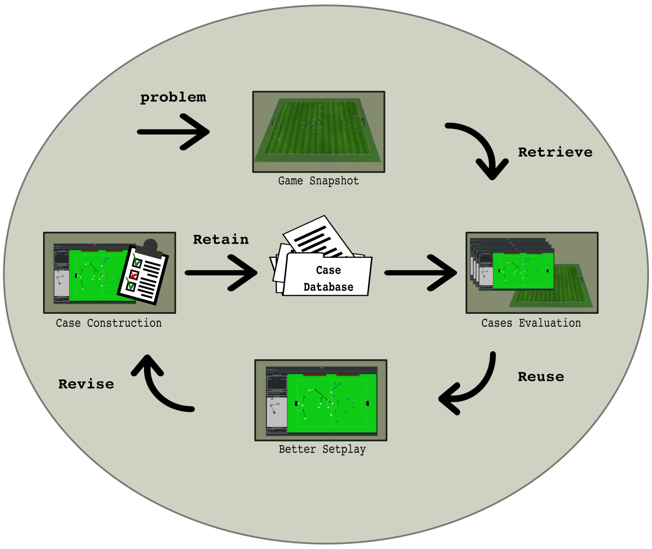


Fig. 2. Life cycle of BahiaRT's CBR algorithm.

3.2 Changing BahiaRT's implementation of FSF

To adapt the FSF to our team and still support the new features of the setplays [11], some modifications to the framework were necessary. The update in the FSF allowed, for example, the use of pass-based multi-flow setplays. Multiflow setplays have branches derived from the main branch from a given action. As in previous versions of the FSF, only the main branch of a setplay was evaluated. It was necessary to create mechanisms to change the branch based on certain conditions verified from the execution of the action that derives the pass to choose a new branch.

The FSF update also includes the ability to represent players from the opponent team, making the team able to perform setplays and defensive actions such as marking opponents.

Finally, the methods that evaluate the completion of actions defined in the framework were revised and corrected, to correctly verify the state of each type of action.

3.3 Learning setplays from demonstration

We are working on a reinforcement learning engine to use the dataset schema created to support learning setplays from demonstration[13]. We will use the modified tools described previously in this Section to populate the dataset to apply the clustering solution presented to keep similar setplays together in a group.

We developed a toolkit for collecting setplays to the dataset. The idea is to use a *crowdsourcing* approach to build a large and heterogeneous dataset.

Collecting setplays to dataset Before starting the installation process, one must check if their machine meets the system's usage requirements:

- Install git and docker in a modern Linux distribution
- Use of Docker Desktop in Windows 10 or + is also possible, but we haven't tested this setup. If you want to try using the toolkit in Windows, install and set up Docker Desktop, WSL2, and an X Server for Windows.
- Check if your computer uses an NVIDIA Graphic Card and install updated NVIDIA drivers. Make sure your OS is really using your NVIDIA drivers.
- Just update your graphic card drivers, if you do not use an NVIDIA Graphic Card.

Using *Docker* in a Windows 10 machine requires the XServer service to be started before running the container.

In order to successfully install the environment, a few steps must be followed:

- Install Docker in a Debian-based Linux distribution.
- Set up the toolkit.
- Run the toolkit.

For more detailed information on the process, including the required commands, access the toolkit documentation [1].

After completing the installation process, one must run the container using the toolkit, which will open two windows on their computer screen: the RoboViz

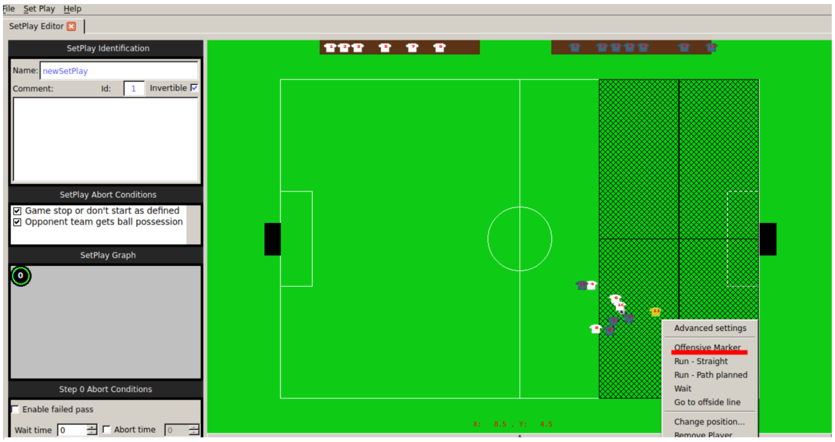
[1] https://bitbucket.org/bahiart3d/setplaysdataset/src/master/README.md

initial screen and the Logplayer. On the Logplayer, a game simulation can be started by selecting one of the game files inside the *games* folder.

Once selected, the RoboViz initial screen turns into a soccer field with a scoreboard on the top left corner. The Logplayer window turns into a game controller, displaying a few tools: advance, rewind, pause, reset, and start a demonstration.

The game files simulate a half-match played by the BahiaRT team. As game progresses, the match eventually unfolds into a position with the potential for a setplay to be designed. Whenever this happens, click on the *start a demonstration* button, and answer a quick set of questions regarding the setplay specifics.

Afterwards, the main setplay creating tool opens on the computer screen: the Setplay Editor (Figure 3). Such an environment mirrors the previously chosen position, allowing the creation of conditions and play steps using the players' set of actions.



Name: newSetPlay
Comment: Id: 1 Invertible ☑

☑ Game stop or don't start as defined
☑ Opponent team gets ball possession

□ Enable failed pass
Wait time 0 □ Abort time 0

X: 8.5 , Y: 4.5

Fig. 3. Setplay Editor environment.

Additionally, to create proper and accurate setplays, one must look at the soccer field as a Cartesian Plane. The numbers on the bottom of the image above represent the user's mouse's X and Y coordinates on the graph.

Creating those set strategies can help algorithms increase their in-game knowledge by strengthening their pattern-recognition ability and widening their tactful repertoire. Once designed, the setplay can be saved by clicking *file* on the top left corner of the screen, where you must select *export setplay* and then save it in the *setplays* folder.

After generating various setplays, create a zip folder to store them and then send it to us through our submission form[2].

[2] https://formfaca.de/sm/fB2AVqE8m

For more detailed information, check the repository usage instructions [3].

We also deployed a video tutorial containing all information about setup and usage of the BahiaRT setplays collecting toolkit[4]. Even after the end of this project we plan to let this form available to continuously collect setplays and update the dataset periodically. The dataset and all the codes related to this project will be publcly availalble soon.

We expect to use the result of this project in RoboCup 2022 to check if the use of the setplays demonstrated by humans can enhance the overall performance of our team.

4 Work in Progress

At this time, we are working on three research projects that can benefit the BahiaRT global performance. We are considering walking and kick optimization, and decision-making under imprecise conditions as our main goals. In the following subsections, we shortly describe these initiatives.

4.1 Walking optimization

The first goal of this optimization is to change the current walking model of the BahiaRT team from the cart table to the inverted pendulum model (IPM). In the cart table model, the Center of Mass (CoM) in the single support phase, when the robot is balanced on only one foot, describes the motion as a car of mass m, representing the agent's CoM, balanced on a table of negligible mass, as you can see in the Fig 4.

The first reason for changing the cart table model to the IPM is the height of the CoM. In the cart table, the CoM height is always the same not changing during the movement of the robot. The result of this invariance is that the robot knee must be bent during the movement to keep the hip height constant. this movement can double the energetic cost of the walking[1]. The second reason is because the cart table is a 2D model and the IPM is a 3D model, that is, the IPM can make movements of walking sideways with more speed and fluidity.

To make this change, we need to change aspects of the calculation of the walking model. We will use the planned feet positions and Zero-moment point (ZMP) trajectory to calculate the vertical and horizontal CoM trajectories to generate a 3D CoM trajectory. We need to calculate the vertical CoM trajectory using a Fourier basis function(see equation (1)). The position and acceleration are discretized with a small time step to generate the horizontal CoM. So, a tridiagonal system for the given IPM differential equation is solved[5]. We will use the Thomas algorithm to solve this tridiagonal system. this algorithm is a simplification of the a Gauss elimination to solve tridiagonal equations.

$$F(t) = \beta 0 + \beta 1 cos(\frac{2\pi t}{L}) + \beta 2 cos(\frac{4\pi t}{L}) \tag{1}$$

[3] https://bitbucket.org/bahiart3d/setplaysdataset/src/master/USAGE.md

[4] https://youtu.be/h_s8rA2IS88

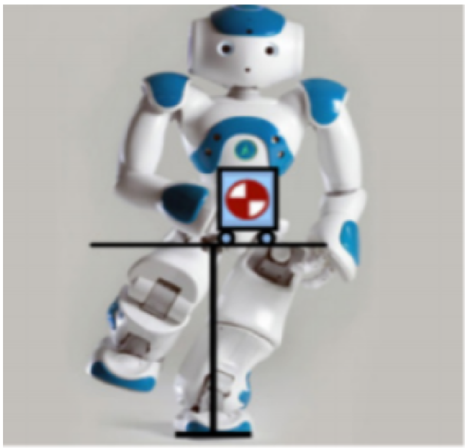


Fig. 4. Cart Table Model[13].

After the implementation of the IPM, we will use reinforcement learning to optimize the parameters of the IPM for better results and efficiency. Then, we intend to do a walking model from the scratch using reinforcement learning(see Figure 5).

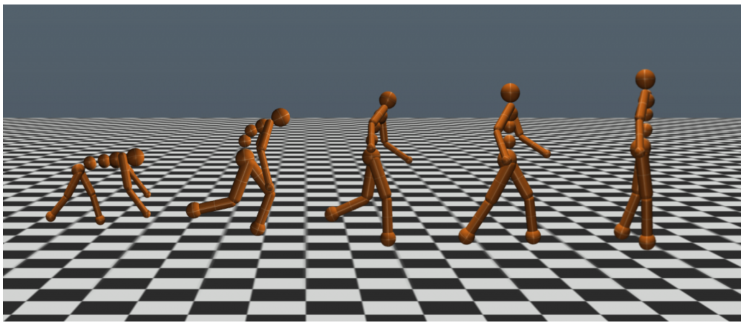


Fig. 5. Reinforcement Learning Walking[13].

4.2 Kick optimization

The optimization of the BahiaRT team kicking skills is one of the projects in development for 2022. There are currently two kick types, one static for long distances that takes a few seconds to prepare and one dynamic for medium distances, which is a little bit faster, but usually leads to more falls. The original idea was to optimize the kicking parameters for both kicks in order to make them faster and more reliable to use. But ever since we started working on the BahiaRT Gym environment, we decided to create new kicking skills from scratch, using the agents' perceptors, such as the vision of objects, the force sensors on the feet and the joints velocities, as parts of the observation space. We intend to feed this information into a neural network using Stable Baselines 3[8] alongside PyTorch[7] and use the output as speed values for the joints as the action space. The reward and details about the neural network algorithm are still in discussion since this is an ongoing work in its early development stages, but we expect to have new and optimized kicks by the RoboCup 2022 event.

4.3 Kick selection

The Kick selection project has the purpose of optimizing the process of deciding which kick must be used in each game situation. It aims to achieve better results in the match, keeping possession of the ball, scoring more goals and spending less time on this process by making better choices.

BahiaRT has at least three different kicks, and the decision process of which one of these will be used in each situation is defined by a hand-coded algorithm. This solution has a lot of negative points, and one of them is the amount of variables to be considered in the decision-making process, such as position of allies and opponents, ball position, orientation of the agents, and a lot of other variables, which increases a lot the complexity of the hand-coded algorithm. Reinforcement learning can be a better option to solve this problem, because it can deal with a larger number of variables and approximate an optimal or sub-optimal policy for selecting the better kick. In this way, it will be possible to define an improved decision kick policy, which is the purpose of this project.

The main points and variables here are the different kick types that the agent may choose, called action space. The variables that are going to be used as observation space are: positions of all agents, their orientation and the position of the ball. Another important point is the choice of episode starting conditions to guarantee randomness in the tests. The ending conditions and rewards for each episode are relevant to assure the convergence of the reinforcement learning algorithm. Most part of the environment has already been modeled, but tests are still required to validate what has been done and to fix any problems.

Acknowledgements

This project is partly funded by CNPq/PIBIC, FAPESB/IC, UNEB/PICIN. We thank team FCPortugal for their cooperation and partnership.

References

1. ARGOLLO, E.M.: Desenvolvimento de um modulo para a execuÇao do movimento de corrida em um agente humanoide simulado (2017)

2. Bergmann, R.: Learning of knowledge-intensive similarity measures in case-based reasoning
3. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. arXiv preprint arXiv:1606.01540 (2016)
4. Cravo, J., Almeida, F., Abreu, P.H., Reis, L.P.: Strategy planner: Graphical definition of soccer set-plays. Data & Knowledge Engineering 94, 110–131 (2014)
5. Kajita, S., Nagasaki, T., Kaneko, K., Hirukawa, H.: Zmp-based biped running control. IEEE Robotics Automation Magazine 14(2), 63–72 (2007)
6. Mota, L., Fabro, J.A., Reis, L.P., Lau, N.: Collaborative behavior in soccer: The setplay free software framework. In: Robot Soccer World Cup. pp. 709–716. Springer (2014)
7. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
8. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research 22(268), 1–8 (2021), http://jmlr.org/papers/v22/20-1364.html
9. RAMOS, C.E.d.R.: Planejador Multiagentes para criação de jogadas ensaiadas em um time de futebol de robôs simulados. Bachelor thesis, Universidade do Estado do Bahia, Salvador, Bahia, Brazil (2017), in Portuguese.
10. Simoes, M.A.C., Nobre, J., Sousa, G., Souza, C., Silva, R.M., Campos, J., Souza, J.R., Nogueira, T.: Strategy Planner: Enhancements to support better defense and pass strategies within an LfD approach. In: 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). pp. 46–52. IEEE, Ponta Delgada, Portugal (Apr 2020), https://ieeexplore.ieee.org/document/9096188/
11. Simões, M.A., Nobre, J., Sousa, G., Souza, C., Silva, R.M., Campos, J., Souza, J.R., Nogueira, T.: Generating a dataset for learning setplays from demonstration. SN Applied Sciences 3(6), 1–20 (2021)
12. Simões, M.A.C., Nogueira, T.: Towards Setplays Learning in a Multiagent Robotic Soccer Team. In: 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). pp. 277–282 (Nov 2018)
13. Simões, M.A.C., da Silva, R.M., Nogueira, T.: A Dataset Schema for Cooperative Learning from Demonstration in Multi-robot Systems. J Intell Robot Syst (Dec 2019), https://doi.org/10.1007/s10846-019-01123-w
14. Watson, I., Marir, F.: Case-based reasoning: A review. The knowledge engineering review 9(4), 327–354 (1994)