

BahiaRT 2023: Team Description Paper for RoboCup 3D Soccer Simulation League

Gabriel Sousa, Wesley Silva, Alan Vitor Nascimento, Glenda Santana, Calvin Albuquerque, 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
`teambahiarth@gmail.com`
<http://www.acso.uneb.br/bahiarth/>

Abstract. This paper describes the significant enhancements in team BahiaRT since RoboCup 2022. We describe our work about setplays learning from demonstration. We also our efforts to enhance our team's skills such as walking and kicking. We are also launching a new initiative to develop a new base code from scratch using software engineering approaches and programming best practices.

1 Introduction

Bahia Robotics Team (BahiaRT) has experienced a continuous evolution in RoboCup competitions in recent 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. In 2022, we also won 3rd place in the Technical Challenge. We were also Champions(in 2015) and runner-up(2016 and 2021) of the Portuguese RoboCup Open competition in the last years. We won nine editions of the Brazilian Robotics Competition from 2013 to 2021 and 2nd place in 2022.

The delay in finishing some vital research projects explains the performance decrease in the last two years. We made the last enhancement on robots' skills over five years ago. We are now working on kicking optimization using recent state-of-the-art deep learning approaches. We finished a long-term project about setplays learning from demonstration described in section 2. In section 3, we describe the current status of our walking optimization project. We plan to finish this project before RoboCup 2023 and run with a new walking skill during the competition. Also, our project about kicking optimization is described in section 4.

Since 2013, BahiaRT has used a source base code, named FCPBase 2013, as part of an international scientific cooperation partnership with team FC Portugal from the Universities of Aveiro and Porto in Portugal. However, last year, we launched a new project to develop our base code from scratch. The old base presents some bugs that are hard to identify and fix. So we decided it was time

to use a completely new base code. In section 5, we describe the current status of the new base code development. We are not sure if it will be ready for RoboCup 2023, but we expect to get an earlier version before it.

2 Learning Setplays from Demonstration

The solution to the problem of learning setplays from demonstrations of human spectators was divided into three stages, as illustrated in Figure 1. In the first stage, volunteers are expected to watch the Multi-Agent Systems (MAS) BahiaRT using a version of *RoboViz* modified in this work, taking breaks in situations that they consider that the robots simulated in BahiaRT had an unwanted collective behavior, or underperform. The scenes captured in stage 1 are taken to stage 2, where Learning from Demonstration (LfD) takes place. To support the first stage, we selected a subset of features present in the BahiaRT world model [14].

In the second stage, the Strategy Planner (SPlanner) [4] tool was changed so that it can be used effectively as a demo generator in the Soccer Simulation 3D (3DSSIM) environment, making it able to start a new *setplay* from the scene captured in *RoboViz*. The BahiaRT setplays collecting toolkit used to collect and build a setplays dataset is described in a previous work [16].

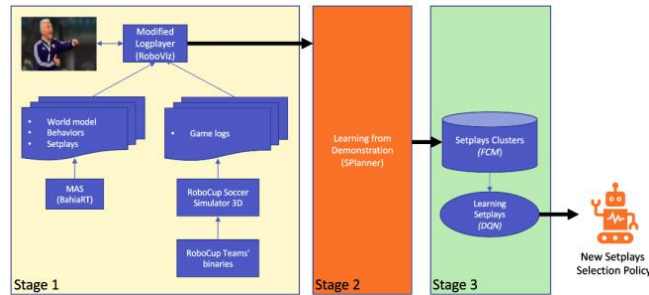


Fig. 1: Learning Setplays from Demonstration: complete solution split into three stages.

We developed a fuzzy clustering engine in the third stage to organize the dataset by setplays similarity. This is important to turn the dataset able to be used by agents in real-time and to deal with semantic equivalence[15][12].

The clustered dataset is used by a deep reinforcement learning solution based on algorithm Deep Q-Network (DQN) [6] to learn a setplays selection policy.

FCPortugal Setplays Framework (FSF) [7] uses a *setplays* manager based on the approach of Case-based Reasoning (CBR) [18]. This approach builds a case history from the agents' use of *setplays*. The team BahiaRT extends the FSF to support setplays execution [11]. However, the CBR solution is not scalable

to a large dataset of setplays. We used a Deep Reinforcement Learning (DRL) strategy to learn a new setplays selection policy to choose one of the dataset clusters. So, the CBR applies to the setplays in the selected cluster. The complete solution is exhibited in Figure 2a.

(a) BahiaRT’s new architecture using the learned setplays selection Policy implemented in a Deep Q-Network. (b) BahiaRT’s training architecture uses BahiaRT Gym

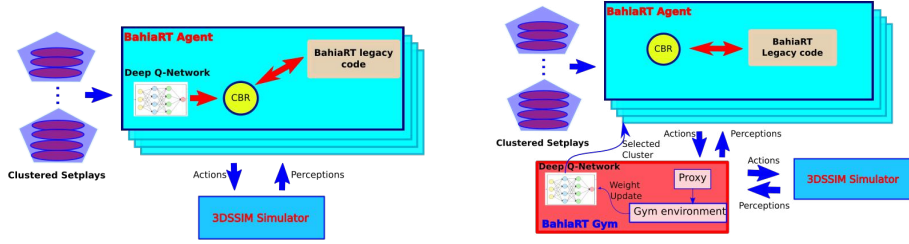


Fig. 2: DQN solution to learn and execute a new setplays selection policy.

The strategy uses the DQN algorithm using a Deep Q-Network to represent the learned policy. The DQN receives the clustered dataset and the properties from BahiaRT’s world model as input and generates a cluster number as output. The CBR loads the setplays definitions of this cluster and selects the best setplay to use in the current situation.

We use Open AI Gym [3] for training the Deep Q Network and the DQN implementation in stable baselines 3 [9]. Figure 2b shows the complete training architecture.

As Open AI Gym does not offer a 3DSSIM environment, we developed our environment named BahiaRT Gym [16]. This environment is uncoupled from the BahiaRT team’s code. Any 3D Soccer Simulation team can use BahiaRT Gym for DRL experiments. Both single-agent and multi-agent training are available. All perceptions sent by the simulator to the agents and the actions sent from agents to the simulator are public in BahiaRT Gym for use in the observation space or reward calculus. BahiaRT Gym also connects the agents to send exploratory actions during training. The observation space used for training the setplays selection policy is defined from the results of preliminary experiments [14] and the results of the setplays files collected using the BahiaRT Setplays Collecting Toolkit [16].

The complete results of the assessment of the trained setplays selection policy are published in a previous work [13]. The results showed evidence that using LfD to collect demonstrations of setplays using a crowdsourcing strategy and training a setplays selection policy can enhance the team’s overall performance. However, we detected some bugs in the execution of setplays by the

team BahiaRT that interfered with the results. We are currently working on fixing these bugs to repeat the training process. We estimate to finish the new training before RoboCup 2023.

3 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 3.



Fig. 3: Cart Table Model[15].

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 walking[2]. The second reason is that 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 coefficients β_0 , β_1 and β_2 of the equation (1) are established through reinforcement learning training using the Open IA Gym tool to define optimal values for them. A better view of this

training cycle can be seen in figure 4. A point that should be highlighted is that logic was created to analyze the height of the vertical CoM, this height value is obtained by understanding the limits of the NAO model, taking the maximum value of the height of its CoM in vertical, when the robot is completely standing, and when the robot is at the lowest point of its CoM in vertical when the legs are open forming a triangle, this value was calculated obtaining a minimum value of 0.17 and a maximum height value of 0.22. A negative weight was created in the reward when the CoM value exceeds these thresholds.

With the Vertical CoM value set. 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 Gauss elimination to solve tridiagonal equations.

$$F(t) = \beta_0 + \beta_1 \cos\left(\frac{2\pi t}{L}\right) + \beta_2 \cos\left(\frac{4\pi t}{L}\right) \quad (1)$$

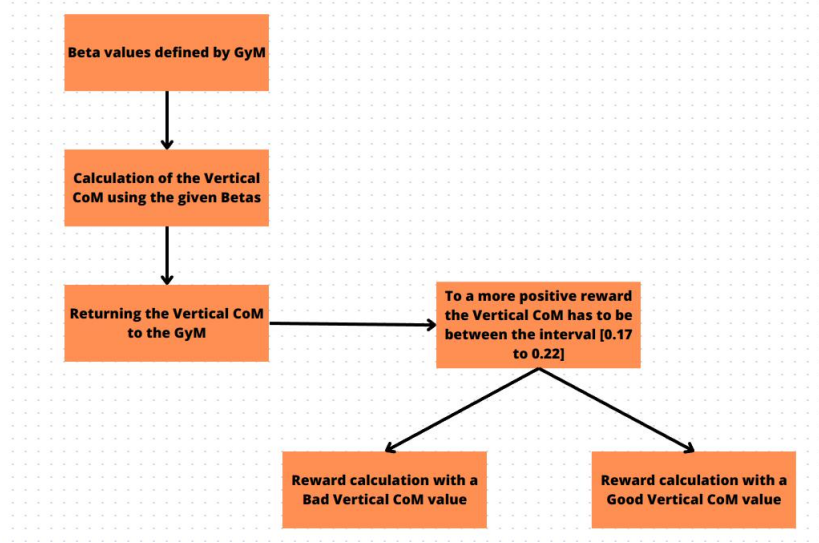


Fig. 4: Reinforcement Learning Vertical CoM Model

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 scratch using reinforcement learning(see Figure 5).

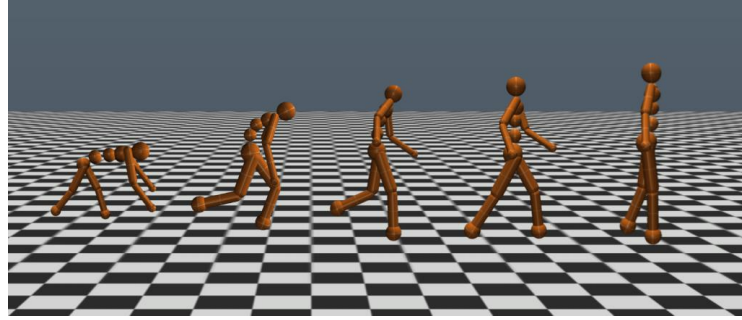


Fig. 5: Reinforcement Learning Walking[15].

4 Kicking optimization

Optimizing the BahiaRT team kicking skills is one of the projects in development for 2023. 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 to make them faster and more reliable to use. Still, ever since we started searching for state of the art in humanoid robot-kicking skills development while also working on the BahiaRT Gym environment, we decided to create new skills from scratch using Deep Reinforcement Learning. Throughout our research, it became clear that DRL is the way to go when dealing with learning new skills for our humanoid robots since many other teams participating in RoboCup have been using it for the past few years as well, such as FCPortugal[1] and MagmaOffenburg[17]

Our idea is to use 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[10] alongside PyTorch[8] and use the output as speed values for the joints as the action space. While the details about the neural network algorithm are still in discussion since this is ongoing work in its early development stages, we already have an idea for our first reward model, which prioritizes stability and speed over accuracy at first. However, it still needs some work and testing. Nonetheless, we expect to have new and optimized kicks by the RoboCup 2023 event.

5 New Base Code

Since 2013, our team has been working on the same code base, which was implemented many times by the authors who passed in our team without any convention or standardization, causing bugs and making the code faulty. Aiming to solve these two problems and build a stronger, cleaner code utilizing modern C++, we decided to implement a new code base.

This is a general structure of the code:

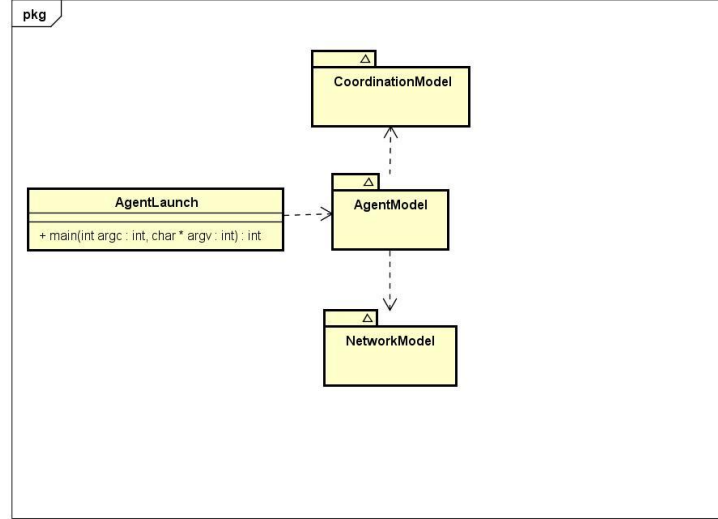


Fig. 6: Shows the high-level architecture of the new base code

Above there are three central namespaces AgentModel, NetworkModel, and CoordinationModel.

AgentLaunch is a class responsible for initializing the agents in the server jointly with the main function.

In the AgentModel we work with the classes which build a world model, parsing the messages sent by the server, as well as S-expressions and world state, including the agent's perception of the world, which encompasses the world object's position, like agents, ball, flags, and goalposts, beside the capture of the general progress of the match: game time, scoreboard and play mode. On this namespace, the single action decision is also made by the agent.

NetworkModel is a piece in which the connection between the agent and server is established. Here are the sockets of the protocol TCP/UDP. Alternative connections to RoboViz (for debugging purposes and training) and BahiaRT Gym (for training purposes) are also supported by NetworkModel.

Finally, the CoordinationModel refers to coordination between our team's agents, as the strategies used in different scenarios, or the team formation.

5.1 Network Model

The Network Model is responsible for providing basic and low-level socket communication, primarily enabling a reliable TCP/UDP channel between the target server and the Network Model's client. This involves managing and handling

message exchanges for both sending and receiving. Although the sockets for the local inter-process communication field already have a big and well-structured variety of open-source libraries accessible to use, the team decided to develop an exclusive library from Unix native sockets to ensure proper management of the communication between server and team agents. The ultimate aim was to create a code base that is fully understandable and easy to maintain.

The Network Model is structured around the NetworkManager namespace, which contains the classes Socket, TcpSocket, and UdpSocket. The structure inside NetworkManager can be illustrated by Figure 7.

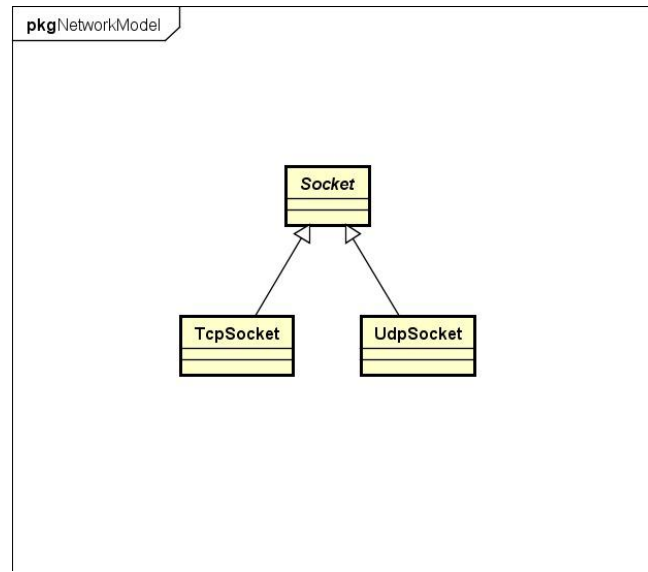


Fig. 7: Communication architecture of the new base code

The NetworkManager namespace uses inheritance from pure virtual classes. The Socket class is a pure virtual class that defines pattern functions and data members inherited by the TcpSocket and UdpSocket classes.

The TcpSocket class establishes a TCP connection between agents and the endpoint. To achieve this, the class handles sending and receiving messages to ensure smooth and efficient communication. During matches, the agents must rely on the TcpSocket class to interact with the server, as it provides the necessary functions to establish a reliable end-to-end connection between the server and agents and coordinate message exchanges.

Similarly, the UdpSocket class stands in charge of providing a UDP transmission channel between agents and the server. Unlike TCP, UDP does not establish a connection before transmitting data, making it a faster protocol and optimizing data transfer. During test matches, the UdpSocket functions must

be required by agents for sending messages to the RoboViz, which is used for debugging purposes.

Testing the new Network Model also enables us to fix an old bug in our legacy base code. We used to send the message size as an integer hard-copied in the TCP buffer. But the server can not support this format. Only the agentProxy could deal with this kind of message. For this reason, our team couldn't connect the server without using agentProxy. Now, we are converting the message size to a string type, and the team can connect the server even without using agentProxy.

5.2 World Model

The WorldModel is a package inside the AgentModel namespace in the first-level architecture (figure 6). Its primary and general objective is to concisely interpret, build and serve to store the world's information, which will feed the yet-to-be-created classes. It is currently under active development by the team. As of now, it has three different main modules.

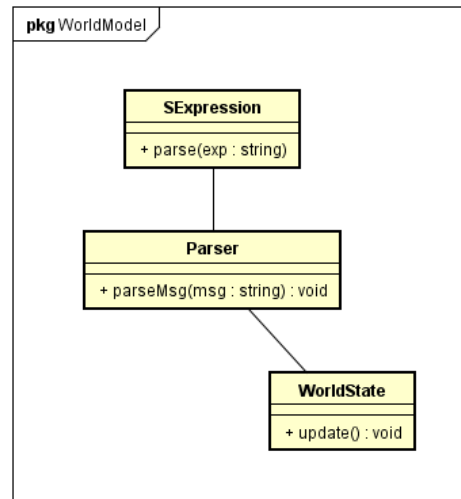


Fig. 8: Initial class diagram proposal for the World Model package.

The first module is called SExpression - a data structure that is able to receive and translate the server message from and to a token/value tree. Following, there is the Parser that reads the s-expression data and respectively updates the elements inside the last class, which is called WorldState - this class is the holder of the game's current information.

Even though research has revealed ready-made, open-source s-expression libraries, the team decided to create an authorial one from scratch. It should enforce code awareness, reasonable programming practices, and a better main-

tainable, flexible, and adaptable resulting module for possible upcoming parsing structure and code usage changes.

Acknowledgements

This project is partly funded by FAPESB/IC. We thank team FCPortugal for their cooperation and partnership.

References

1. Abreu, M., Silva, T., Teixeira, H., Reis, L., Lau, N.: 6d localization and kicking for humanoid robotic soccer. *Journal of Intelligent & Robotic Systems* 102 (06 2021)
2. Argollo, E.M.: Desenvolvimento de um Módulo para Execução do Movimento de Corrida em um Agente Humanoide Simulado. Bachelor Thesis, Universidade do Estado da Bahia (UNEB), Salvador, BA, Brazil (2017)
3. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI Gym. *arXiv.org* (2016), <http://arxiv.org/abs/1606.01540>
4. Cravo, J., Almeida, F., Abreu, P.H., Reis, L.P., Lau, N., Mota, L.: Strategy planner: Graphical definition of soccer set-plays. *Data & Knowledge Engineering* 94, 110–131 (Nov 2014), <http://www.sciencedirect.com/science/article/pii/S0169023X14000950>
5. Kajita, S., Nagasaki, T., Kaneko, K., Hirukawa, H.: Zmp-based biped running control. *IEEE Robotics Automation Magazine* 14(2), 63–72 (2007)
6. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. *Nature* 518(7540), 529–533 (Feb 2015), <http://www.nature.com/articles/nature14236>
7. Mota, L., Lau, N., Reis, L.P.: Co-ordination in RoboCup’s 2D simulation league: Setplays as flexible, multi-robot plans. In: 2010 IEEE Conference on Robotics, Automation and Mechatronics. pp. 362–367 (Jun 2010)
8. 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>
9. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research* (2021)
10. 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>

11. 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/>
12. 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)
13. Simões, M.A.C.: Aprendizagem por Demonstração de Planos Coordenados em Sistemas Multiagentes. Ph.D. thesis, Universidade Federal da Bahia, Salvador, BA, Brazil (Jul 2022)
14. 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)
15. 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>
16. Simões, M.A., Mascarenhas, G., Fonseca, R., dos Santos, V.M., Mascarenhas, F., Nogueira, T.: BahiaRT Setplays Collecting Toolkit and BahiaRT Gym. *Software Impacts* 14, 100401 (Nov 2022), <https://linkinghub.elsevier.com/retrieve/pii/S2665963822000938>
17. Spitznagel, M., Weiler, D., Dorer, K.: Deep reinforcement multi-directional kick-learning of a simulated robot with toes. In: 2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). pp. 104–110 (2021)
18. von Wangenheim, C.G., v Wangenheim, A.: Raciocínio baseado em casos. Manole, Barueri (2003)