FRA-UNIted — Team Description 2026

Thomas Gabel, Berkan Eren, Tobias Schierstein, Jorge Vanegas, Eicke Godehardt

Faculty of Computer Science and Engineering
Frankfurt University of Applied Sciences
60318 Frankfurt am Main, Germany
{tgabel|godehardt}@fra-uas.de,
{berkan.eren|tobias.schierstein|vanegas}@stud.fra-uas.de

Abstract. The main focus of FRA-UNIted's effort in the RoboCup soccer simulation 2D domain is to apply machine learning techniques in complex domains. In particular, we are interested in applying reinforcement learning methods, where the training signal is only given in terms of success or failure. In this paper, we review some of our recent efforts taken during the past year, putting a special focus on a novel approach to learn how to utilize body-checks during normal play as well as a new method to estimate other players' stamina usage.

1 Introduction

The soccer simulation 2D team FRA-UNIted is the continuation of the former Brainstormers project which has ceased to be active in 2010. The ancestor project was established in 1998 by Martin Riedmiller, starting off with a 2D team which had been led by the first author of this team description paper since 2005. Our efforts in the RoboCup domain have been accompanied by the achievement of several successes such as multiple world champion and world vice champion titles as well as victories at numerous local tournaments. Our team was re-established in 2015 at the first author's new affiliation, Frankfurt University of Applied Sciences, reflecting this relocation with the team's new name FRA-UNIted.

The underlying and encouraging research goal of FRA-UNIted is to exploit artificial intelligence and machine learning techniques wherever possible. Particularly, the successful employment of reinforcement learning (RL, [5]) methods for various elements of FRA-UNIted's decision making modules – and their integration into the competition team – has been our main focus. Moreover, the extended use of the FRA-UNIted framework in the context of university teaching has moved into our special focus.

In this team description paper, we refrain from presenting approaches and ideas we already explained in team description papers of the previous years [1, 2]. Instead, we focus on topics that have moved into our focus in the course of the past year. As you will see, over the past year, the idea to body-check opponents and stamina assessments have increasingly become topics of our interest. We start this team description paper, however, with a short general overview of

the FRA-UNIted framework. Note that, to this end, there is some overlap with our older team description papers including those written in the context of our ancestor project (Brainstormers 2D, 2005–2010) which is why the interested reader is also referred to those publications, e.g. to [3, 4].

1.1 Design Principles

FRA-UNIted relies on the following basic principles:

- There are two main modules: the world module and decision making
- Input to the decision module is the approximate, complete world state as provided by the soccer simulation environment.
- The soccer environment is modeled as a Markovian Decision Process (MDP).
- Decision making is organized in complex and less complex behaviors where the more complex ones can easily utilize the less complex ones.
- A large part of the behaviors is learned by reinforcement learning methods.
- Modern AI methods are applied wherever possible and useful (e.g. particle filters are used for improved self localization).

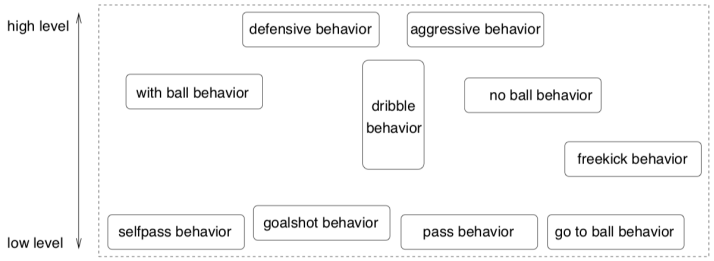


Fig. 1. The Behavior Architecture

1.2 The FRA-UNIted Agent

The decision-making process of the FRA-UNIted agent is inspired by behavior-based robot architectures. A set of more or less complex behaviors realize the agents' decision making as sketched in Fig. 1. To a certain degree this architecture can be characterized as hierarchical, differing from more complex behaviors, such as "no ball behavior", to very basic, skill-like ones, e.g. "pass behavior". There is no strict hierarchical sub-divisioning, which is why a low-level behavior may call a more abstract one. For instance, the behavior responsible for intercepting the ball may, under certain circumstances, decide that it is better to not intercept the ball, but to focus on more defensive tasks and, in doing so, call the "defensive behavior" and delegating responsibility for action choice to it.

2 An RL Approach to Utilize Body-Checks

The approach presented here is based on a Bachelor thesis conducted within FRA-UNIted [6], which explored the feasibility of learning deliberate body-checking as an explicit reinforcement learning (RL) behavior in the 2D architecture. Motivation for this study arose from previous observations that RL and neural-network-based approaches have shown particular promise for close-range, localized behaviors. Body-checking represented an unexplored interaction primitive that, if successfully learned, could potentially yield small, but meaningful advantages in tightly contested situations. Reinforcement learning was therefore chosen as the primary method to explore this hypothesis [5, 3]. The learned policy ultimately did not surpass the handcrafted heuristic baseline in terms of consistent and meaningful disruption. Nevertheless, the investigation provided valuable insight into the structural challenges of modeling physically grounded interaction in the RoboCup 2D simulator and helped clarify practical limitations and design trade-offs when integrating RL modules into FRA-UNIted.

2.1 Problem Formulation

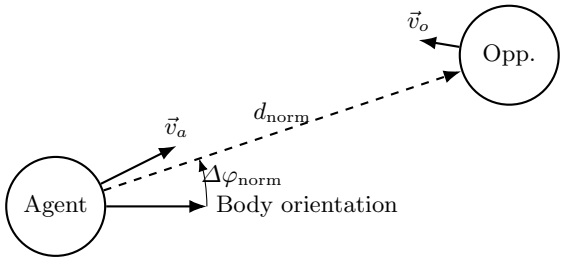
The body-check task was defined as a localized interaction between a learning agent and a selected opponent. The intended objective was to align with the opponent, close down distance, establish contact, and maintain it long enough to influence the opponent's movement dynamics. Contact itself was not modeled as a terminal condition. Episodes were allowed to continue beyond the first contact in order to observe post-contact interaction effects. To reduce tactical complexity and isolate the physical interaction problem, training and evaluation were conducted in controlled `PlayOn` scenarios generated via the coach infrastructure. A fixed set of manually defined starting configurations ensured reproducibility and limited environmental variance.

2.2 State and Action Representation

The state representation was deliberately kept compact and relative. It consisted of a five-dimensional, rotation- and translation-invariant feature vector including: relative distance to the selected opponent, angular deviation between agent orientation and opponent, closing velocity, opponent speed, and a velocity-alignment feature.

The compact representation was chosen to maintain interpretability of learned Q-values and to focus strictly on pairwise interaction geometry. However, in retrospect, the feature set likely provided insufficient information to reliably model all sub-components of the task. Additional geometric or dynamical features may have been beneficial.

The action space consisted of discrete movement primitives (directional dash and turn actions). The design intention was to keep the action space simple and interpretable. We found that idle and backward dash actions proved counterproductive during early exploration phases, as purely exploratory action selection frequently resulted in non-progressive behavior.



$$v_{\text{align}} = \cos\phi = \frac{\vec{v}_a \cdot \vec{r}}{\|\vec{v}_a\|\|\vec{r}\|}$$

Fig. 2. Schematic representation of the relative state features used for body-check learning: normalized distance d_{norm}, normalized angular deviation $\Delta\varphi_{\text{norm}}$, agent and opponent velocities, and the derived velocity-alignment feature v_{align}.

2.3 Reward Design

Reward design evolved significantly during development and became increasingly complex. Initially, behavioral cloning of the heuristic policy was attempted but did not yield satisfactory results. Subsequently, heuristic behavior was indirectly approximated through reward shaping by incentivizing alignment and forward acceleration toward the opponent. Contact detection was implemented agent-side via distance thresholds. Although the simulator provides collision-related information, geometric distance-based detection proved more reliable and consistent for logging and reward assignment. Consequently, distance-based contact detection was chosen as the dominant reward signal. In retrospect, the reward formulation became unnecessarily complex. Multiple shaping components were introduced in an attempt to replicate heuristic behavior, likely complicating the learning dynamics and contributing to instability.

2.4 Policy Comparison

Three policy variants were evaluated: a uniform random baseline, a handcrafted heuristic strategy, as well as a neural value-based policy trained via Q-learning.

The heuristic policy reliably generated contact but did not explicitly optimize post-contact influence. The neural policy did not consistently exceed heuristic performance under the defined evaluation criteria.

2.5 Structural Insights

The investigation revealed that body-checking implicitly consists of several sequential sub-problems: alignment, closing distance, establishing contact, and maintaining contact. Not all of these stages may be equally suitable for value-based RL under limited state representations. Our key insights include:

– Altough establishing contact is a requirement, it is not equivalent to meaningful disruption.
– Physical interaction effects are relatively weak and sensitive to small geometric variations.
– The state representation likely contained too little information.
– Reward shaping became overly complex, hindering stable learning.
– RL is not uniformly suited for all sub-components of the body-check task.

Overall, the structure and individual components required for a learning-based body-check module became clearly identifiable through this work. Even though the resulting policy did not outperform the heuristic baseline, the study provided insight into the practical limits and integration challenges of reinforcement learning within the robotic soccer context.

3 Towards a Reliable Stamina Assessment from Player Observations

3.1 Motivation and Goals

Currently, agents use all given data from the Soccer Server observed from the field. Better decision-making requires more reliable data about each agent. Gaining the ability to compute a reliable stamina estimation improves the accuracy of decision-making. Furthermore, gaining an advantage compared to other agents whose decisions are not influenced by a stamina factor is crucial. The goal is to make a reliable assessment of the stamina value of each player, providing additional data on the agents. Not only single values of the agents but also values for the team can be extracted, such as whether the whole team has a low stamina value versus the own team. Then, strategic choices can be optimized as a result of knowing this additional information.

Problem Description: Looking closer at the task, many issues arise when considering that a reliable stamina assessment is needed from plain player observations. While the actions themselves are deterministic for the player, a uniform distribution noise is added to movements during each tick the agent moves, as defined by the ServerParam::PLAYER_RAND Default = 0.10. This noise is not only added along the magnitude of the velocity vector, but is a new vector in any direction. This makes single values not accurate, with a deviation of a maximum of 10 percent by default. Another critical phase is the simulation of the recovery of stamina, as done by the Soccer Server. During this simulation, careful consideration have to be made when the agents are close to the thresholds of the recovery values.

3.2 Concept

Observing a single value is not accurate enough for a reliable stamina assessment; for this reason, we simulate the stamina for each player from the start to the

end of the game. We apply the Law of Large Numbers, as when observing many values, the distribution approaches the mean, and the mean for the uniform distribution used is $\mu_0 = 0$. This eliminates the deviations from the noise each movement undergoes. Another innovation is the special consideration for stamina recovery, taking into account when the stamina is close to thresholds for recovery. With this, the stamina assessment is very reliable with parameters to optimize the needed accuracy.

Movement: Understanding the movement of the agents is important for evaluating where the stamina is used. Acceleration is a simple addition to the velocity of the agent, creating the new velocity, as seen in 1. Moving an agent is the same as adding the new velocity to the current point, moving to the new point, shown in Equation 2.

$$(u_x^{t+1},\, u_y^{t+1}) = (v_x^t,\, v_y^t) + (a_x^t,\, a_y^t)\colon \text{accelerate} \tag{1}$$

$$(p_x^{t+1},\, p_y^{t+1}) = (p_x^t,\, p_y^t) + (v_x^{t+1},\, v_y^{t+1})\colon \text{move} \tag{2}$$

The stamina only affects the acceleration vector, thus obtaining the acceleration vector is key to reversing the stamina usage.

3.3 Stamina

With the formulas from Section 3.2, we can start reversing the steps to get closer to the stamina value.

$$(v_x^{t+1}, v_y^{t+1}) = (p_x^{t+1}, p_y^{t+1}) - (p_x^t, p_y^t) \qquad \text{(velocity at } t+1\text{)} \tag{3}$$

$$(a_x^t, a_y^t) = (u_x^{t+1}, u_y^{t+1}) - (v_x^t, v_y^t) \qquad \text{(acceleration at } t\text{)} \tag{4}$$

First, calculating the velocity the agent had while performing the movement, seen in Equation 3, is crucial because a decay is applied to the velocity after each movement. After obtaining the velocity, the acceleration for the given velocity can be calculated based on the previous velocity the player had, as in Equation 4.

After getting the acceleration vector, we need to consider how the acceleration vector is calculated from the stamina usage.

$$\text{power\_rate} = \text{dir\_rate} \times \text{player\_effort} \times \text{player\_dashPowerRate} \tag{5}$$

$$(a_x^t,\, a_y^t) = \text{Power} \times \text{power\_rate} \times \big(\cos(\theta_t),\, \sin(\theta_t)\big) \tag{6}$$

Formula 6 illustrates the calculation from the server for the acceleration vector. It is important to know that stamina is referred to as Power; the relationship is one-to-one in the current Soccer Server version. In previous versions, where a negative stamina dash was possible, a negative stamina would have been needed for calculation; however, only positive powers from 0 to 100 are possible, as defined in the server parameters. The power_rate from Equation 5 is a combination of player stats, as well as the angle of the acceleration relative to the body.

3.3.1 Reversing Stamina Consumption

Using Equation 6 and computing the Euclidean distance, which only obtains the magnitude, the orientation of the vector is not relevant for the stamina consumption in this step. We obtain the equations 7 and 8.

$$\text{Power} = \frac{\sqrt{(a_x^t)^2 + (a_y^t)^2}}{\text{power\_rate}} \tag{7}$$

$$= \frac{\sqrt{(a_x^t)^2 + (a_y^t)^2}}{\text{dir\_rate} \times \text{player\_effort} \times \text{player\_dashPowerRate}} \tag{8}$$

For the equation to be solved, four variables have to be known. First, the acceleration vector, which can be calculated from Equation 4. Then, two of the variables are player parameters. In the FRA-UNIted agent, each agent is already assigned the correct player type, as given by the Soccer Server, so this is already known. Lastly, the dir_rate needs to be calculated, which can also be done using the acceleration vector while knowing the body direction from the agent before the movement.

3.3.2 Stamina Recovery Simulation

During stamina recovery simulation, noise has to be considered. Agents often stop spending stamina when it reduces the stamina recovery value. Here, special care must be taken to avoid a butterfly effect during this critical phase of stamina simulation. Apart from this potential butterfly effect, the code from the Soccer Server can be copied and used for the simulated stamina recovery in the same manner.

3.4 Coach

The coach is the best starting point for obtaining the most reliable data. The coach receives global see messages whenever needed. These messages contain the current position and the delta change of the position, which is the velocity. Both of these data points are the only needed observations for performing a full simulation of the stamina of each observed agent.

3.5 Initial Results

Initial results from 11 agents on the field show that stamina usage has been simulated, and the difference from the true stamina capacity is illustrated in Figure 3.

In the beginning of a game, the simulated value is closer to the true value, while it diverges from the true capacity as time progresses. This is due to parameter settings safeguarding critical moments when stamina is close to recovery thresholds. Another point of mention is the reset of stamina during halftime of

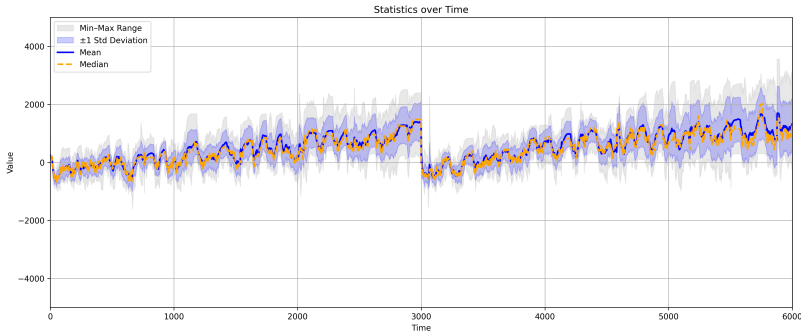


Fig. 3. Deviation of modelled and true stamina statistics.

the game. With this resetting of stamina, deviations start from zero again and a more reliable assessment is possible. With these results, it is shown that reliable stamina values can be extracted while keeping the butterfly effect to a minimum.

4 Conclusion

In this team description paper we have outlined the characteristics of the FRA-UNIted team participating in RoboCup's 2D Soccer Simulation League. After a brief presentation of the fundamental architecture of the agents, we emphasized that our recent work has returned to reinforcement learning tasks as well as to approaches for an improved opponent and teammate modelling.

References

1. Gabel, T., Breuer, S., Sommer, F., Godehardt, E.: FRA-UNIted - Team Description 2019 (2019), Supplementary material to RoboCup 2019: Robot Soccer World Cup XXIII, LNCS, Sydney, Australia
2. Gabel, T., Eren, B., Vanegas, J., Godehardt, E.: FRA-UNIted - Team Description 2025 (2025), Supplementary material to RoboCup 2025: Robot Soccer World Cup XXIX, LNCS, Salvador, Brazil
3. Gabel, T., Riedmiller, M., Trost, F.: A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach. In: L. Iocchi, H. Matsubara, A. Weitzenfeld, C. Zhou, editors, RoboCup 2008: Robot Soccer World Cup XII, LNCS. pp. 61–72. Springer, Suzhou, China (2008)
4. Riedmiller, M., Gabel, T.: On Experiences in a Complex and Competitive Gaming Domain: Reinforcement Learning Meets RoboCup. In: Proceedings of the 3rd IEEE Symposium on Computational Intelligence and Games (CIG 2007). pp. 68–75. IEEE Press, Honolulu, USA (2007)
5. Sutton, R., Barto, A.: Reinforcement Learning. An Introduction. MIT Press/A Bradford Book, Cambridge, USA (1998)
6. Vanegas, J.: Körpereinsatz im simulierten 2D-Roboterfußball: Entwicklung eines Body-Check-Verhaltens mit Methoden des Reinforcement Learning (2026), Bachelor Thesis, Frankfurt University of Applied Sciences