FRA-UNIted — Team Description 2017

Thomas Gabel, Steffen Breuer, Constantin Roser,
Roman Berneburg, Eicke Godehardt

Faculty of Computer Science and Engineering
Frankfurt University of Applied Sciences
60318 Frankfurt am Main, Germany
{tgabel|godehardt}@fb2.fra-uas.de,
{sbreuer|croser|bernebur}@stud.fra-uas.de

Abstract. The main focus of FRA-UNIted's effort in the RoboCup soccer simulation 2D domain is to develop and 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. This paper outlines the history and architecture of the FRA-UNIted team and highlights current educational and research efforts.

1 Introduction

The soccer simulation 2D team FRA-UNIted is a continuation of the former Brainstormers project which has ceased to be active in 2010. The ancestor Brainstormers 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. Over the years, a number of sister teams emerged (e.g. the Tribots and Twobots) participating in real robot leagues. 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 throughout the first decade of the new millennium.

While the real robot teams mentioned were closed down entirely, the 2D team has been in suspend mode since 2010 and 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, as well.

As a continuation of our efforts in the ancestor project, 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, [1]) methods for various elements of FRA-UNIted's decision making modules – and their integration into the competition team – has been and is our main focus.

Moreover, the extended use of the FRA-UNIted framework in the context of university teaching has moved into our special focus. So, we aim at employing the 2D soccer simulation domain as a fundament for teaching agent-based programming, foundations of multi-agents systems as well as applied machine learning algorithms.

In this team description paper, we disdain from presenting approaches and ideas we already explained in team description papers of the previous years. Instead, we focus on recent changes and extensions to the team as well as on reporting partial results of work currently in progress. 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.

1.1 Design Principles

FRA-UNIted relies on the following basic principles:

– There are two main modules: the world module and the decision making module.
– Input to the decision module is the approximate, complete world state as provided by the soccer simulation environment.
– The soccer environment is modelled 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).

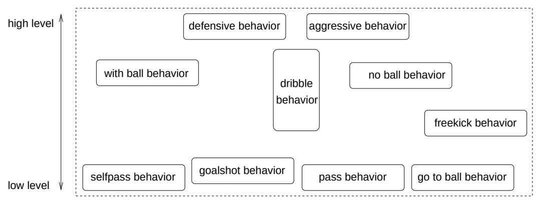

Fig. 1. The Behavior Architecture

1.2 The FRA-UNIted Agent

The decision making process 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 Figure 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". Nevertheless, there is no strict hierarchical sub-divisioning. Consequently, it is also possible for a low-level behavior to 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 so doing, call the "defensive behavior" delegating responsibility for action choice to it.

2 Current Efforts

This section is meant to reflect some of our research and development activities throughout the recent year.

2.1 Refactoring the Code Base

After reestablishing the team in 2015, we intended to also use it more actively in teaching (including student projects, bachelor theses, and student-based workgroups). Since the team was created in 1998, almost 20 years ago, and suspended since 2010, the team code was not state of the art from a technical point of view. From a beginner's and many students' point of view, many structures appeared grown and thus in parts unclear and confusing. Therefore, our main focus was on a rework and clean-up of the team code that would provide these students an easier access to the team. In the refactoring process the team code was reorganized in a whole new file structure, which is more understandable and easier to use. A single class per file policy was enforced as well as a strict object-oriented software re-design and a warning-free coding policy to enable easier debugging. Specifically, to improve the team and make working with it easier, we extended it in the following ways: (1) We designed a new build process that formats the compilation output, making it easier to see errors that occur during compilation. (2) To simplify debugging we implemented a debug mechanism that starts all agents, but one of them in the `gdb` with a set break-point. (3) When testing with extended logging, we identified performance problems. We solved them by generating a file system in the main memory and writing the log data into it. At the end of the match the files are moved to their default location.

2.2 Learning to Dribble

It has been one of our team's traditions to replace hand-coded player skills by those learnt using reinforcement learning [2–6]. Our current effort includes making our players learn autonomously to dribble as fast as possible into any given direction, keeping ball possession, avoiding collisions, and moving ahead as gently as possibly. This is work in progress which is why more details on this subproject will be available and worth reporting at a later point of time. We hope to employ the RL-based neural dribble behavior during the next tournaments.

2.3 Improving Neck and View Behaviors

As a result of reviewing our team's playing strategy and comparing it to strategies of several opponent teams, we found that improvements regarding our own way of influencing the players' neck and view behavior need to be applied. The first step of doing so was done by merging both, the neck and the view controller, to a combined "NeckAndView" behavior. Our goal in developing the combined behavior is to be able to match up neck- and view-controlling to a point where they deliver the best results, helping us to improve our team's competitiveness. This will be achieved by a combined decision making process that allows our players to decide which neck orientation and view field fit together best at any given time.

This is work in progress. At the moment, the combined controller acts like the two former ones with addition of a functionality that allows us to perform emergency requests in order to look at specified absolute angles on the field. The result of these requests is calculated by firstly checking, if the desired angle is reachable by only changing the player's neck orientation and then applying the needed neck angle to the player. Should this not be possible, the needed view angle will be calculated and applied alongside the newly calculated neck angle. Due to distortion, a critical point (180 degrees behind the player) arises where the method will fail to deliver the desired results.

2.4 Repeatability of Multi-Agent Competition Results

Recently, we published the results of empirical studies that aimed at quantifying the progress made in soccer simulation 2D during the previous years: We focused on both, soccer simulation's so-called "first stable period" from 2003 to 2007 [7] as well as on the more recent "second stable period" from 2010 to 2015 [8][1].

In line with this and as a continuation of our efforts to systematically evaluate the recent progress made in soccer simulation, we again performed an extensive empirical study involving all of the released 18 binaries from the RoboCup 2016 tournament[2]. The study comprised approximately 85.000 matches and aimed at finding the ground truth of the team performance of last year's world championships tournament. This is similar to the work of Budden et al. [9] who specifically focused on different tournament formats and answered the question which tournament formats do bring about results that are most aligned to the "ground truth" of the participating teams' strengths.

Here, we present an excerpt of our findings. The full set of empirical results is currently in the process of being finalized and prepared for being published. Our results allow for concluding:

- RoboCup 2016 has witnessed a situation where the two finalist teams (Gliders and Helios) were as close to each other as never before, at least as far as the RoboCup tournaments from 2009 onward as well as between 2003 and 2007 [7] are considered. A standard game between these two competitors ends in a draw in 58% of the cases. Figure 2 (right) shows that, to this end, the second closest situation was in 2012 where, however, the share of

[1] Note that a "stable period" refers to the time period during which the Soccer Server software underwent no changes.

[2] Binaries retrieved from chaosscripting.net.

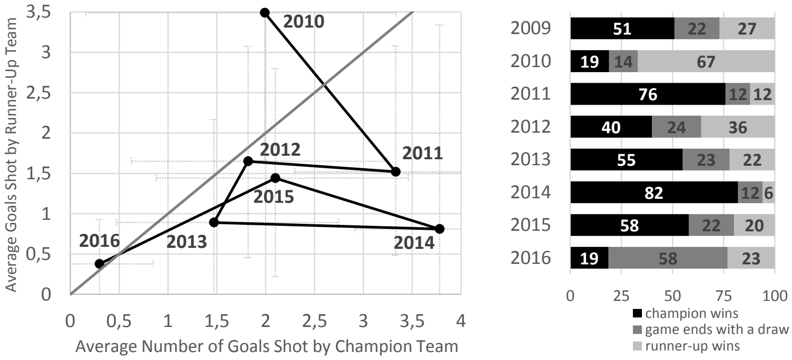

Fig. 2. Evaluation of RoboCup soccer simulation 2D final matches over the years.

r^{gt}	Team	Percentage of Matches			Avg.	Goals		r^*		r^*_{16}		r^*_8	
		Won	Drawn	Lost	Points	Shot	Received		$r^* - r^{gt}$		$r^*_{16} - r^{gt}$		$r^*_8 - r^{gt}$
1	HELIOS	80.0	16.0	4.0	2.560	3.010	0.159	2	-1	2	-1	2	-1
2	Gliders	80.2	13.5	6.3	2.541	3.082	0.353	1	1	1	1	1	1
3	Oxsy	77.6	13.5	8.8	2.464	2.731	0.391	5	-2	5	-2	5	-2
4	FRA-UNIted	53.4	31.5	15.2	1.916	1.173	0.398	11	-7	11	-7		
5	CSU-Yunlu	48.8	21.7	29.5	1.682	1.536	0.927	4	1	4	1	4	0
6	MT	47.3	21.9	30.8	1.638	1.693	1.156	7	-1	7	-1	7	-2
7	MarliK	43.3	28.1	28.6	1.580	1.149	0.753	10	-3	10	-3		
8	Hermes	42.9	25.8	31.4	1.543	1.356	0.960	9	-1	9	-1		
9	Ri-one	37.6	27.9	34.5	1.407	1.054	0.959	3	6	3	6	3	3
10	Shiraz	38.4	14.9	46.7	1.301	1.039	1.743	6	4	6	4	6	1
11	HfutEngine	34.7	23.3	42.0	1.273	1.159	1.280	17.5	-6.5				
12	FURY	28.3	27.1	44.6	1.120	0.686	1.364	8	4	8	3	8	0
13	Ziziphus	27.5	21.3	51.2	1.038	1.027	1.577	15	-2	15	-3		
14	ITAndroids	24.2	19.8	56.0	0.923	0.914	1.707	13	1	13	0		
15	CYRUS	18.6	18.9	62.5	0.747	0.793	2.166	12	3	12	2		
16	HillStone	19.8	13.0	67.1	0.725	1.424	3.235	14	2	14	1		
17	FCP-GPR	9.9	17.1	73.0	0.469	0.491	2.208	16	1	16	0		
18	LeftEagle	5.3	9.2	85.5	0.250	0.529	3.508	17.5	0.5				
							Sum of Absolute Values		47		36		10

Table 1. Comparing the ground truth of RoboCup 2016 as generated from a series of 85.000 matches to the competition results generated on-site in Leipzig.

draws was as low as 24%. The parity of the champion and runner-up is also supported by the average score (averaged over 500 matches) of 0.300:0.378 with a standard deviation of 0.546:0.638 which is by far the lowest number of goals shot in recent years as shown in Figure 2 (left).

– In [9], the authors use the L_1 distance to capture the difference between the ranking r^* resulting from a RoboCup tournament and the ranking r^{gt} resulting from a massively repeated simulation of matches using published binaries (ground truth):

$$d_1(r^*, r^{gt}) = \sum_{i=1}^{n} |r^*_i - r^{gt}_i|$$

Considering only the top eight teams from the respective year's RoboCup tournament, the authors found that the d_1 distance was 12 in 2012 and 12 in

2013. In 2014, given the changed tournament format (which has been kept ever since), that distance value has dropped to 4. For last year's tournament (RoboCup 2016), however, we find that d_1 takes a value of 10. It remains an open question whether this value would have been lower or higher under a different tournament scheme.

– Table 1 provides an overview over the discrepancy between the ranking from RoboCup 2016 and the ground truth using the continuous evaluation scheme [9]. If the focus on the teams on the top 8 ranks is dropped, we find that the ranking error takes a value of 36 on the set of the teams that made it to the main round (top 16) and even 47 on the full set of all 18 participants.

3 Summary

In this team description paper we have outlined the characteristics of the FRA-UNIted team participating in RoboCup's 2D Soccer Simulation League. We have stressed that this team is a continuation of the former Brainstormers project, pursuing similar and extended goals in research, development as well as for teaching purposes. Specifically, we have put emphasis on our most recent research and development activities.

References

1. Sutton, R., Barto, A.: Reinforcement Learning. An Introduction. MIT Press/A Bradford Book, Cambridge, USA (1998)
2. Riedmiller, M., Merke, A.: Using Machine Learning Techniques in Complex Multi-Agent Domains. In Stamatescu, I., Menzel, W., Richter, M., Ratsch, U., eds.: Adaptivity and Learning. Springer (2003)
3. Gabel, T., Riedmiller, M.: Learning a Partial Behavior for a Competitive Robotic Soccer Agent. KI Zeitschrift **20** (2006) 18–23
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), Honolulu, USA, IEEE Press (2007) 68–75
5. Gabel, T., Riedmiller, M., Trost, F.: A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach. In: RoboCup 2008: Robot Soccer World Cup XII, LNCS, Berlin, Springer Verlag (2008, to appear)
6. Riedmiller, M., Gabel, T., Hafner, R., Lange, S.: Reinforcement Learning for Robot Soccer. Autonomous Robots **27** (2009) 55–74
7. Gabel, T., Riedmiller, M.: On Progress in RoboCup: The Simulation League Showcase. In: J. Ruiz-del-Solar, E. Chown, P. Plger, editors, RoboCup 2010: Robot Soccer World Cup XIV, LNCS, Singapore, Springer (2010) 36–47
8. Gabel, T., Falkenberg, E., Godehardt, E.: Progress in RoboCup Revisited: The State of Soccer Simulation 2D. In: S. Behnke D. Lee, S. Sariel, R. Sheh, editors, RoboCup 2016: Robot Soccer World Cup XX, LNCS, Leipzig, Springer (2016)
9. Budden, D., Wang, P., Obst, O., Prokopenko, M.: RoboCup Simulation Leagues: Enabling Replicable and Robust Investigation of Complex Robotic Systems. IEEE Robotics and Automation Magazine **22** (2015) 140–146