UT Austin Villa: RoboCup 2021 3D Simulation League Competition Champions

Patrick MacAlpine[†], Bo Liu[‡], William Macke[‡], Caroline Wang[‡], and Peter Stone[†‡]

[†]Sony AI, USA
patrick.macalpine@sony.com
[‡]The University of Texas at Austin, USA
bliu@cs.utexas.edu, wmacke@cs.utexas.edu, clw4542@cs.utexas.edu, pstone@cs.utexas.edu

Abstract. The UT Austin Villa team, from the University of Texas at Austin, won the 2021 RoboCup 3D Simulation League, winning all 19 games the team played. During the course of the competition the team scored 108 goals while conceding only 5. Additionally the team finished second in the overall RoboCup 3D Simulation League technical challenge by finishing second in both the fat proxy and scientific challenges. This paper details and analyzes the results of the 2021 competition, and also presents a new deep RL learning framework that was presented during the scientific challenge.

1 Introduction

UT Austin Villa won the 2021 RoboCup 3D Simulation League for the ninth time in the past ten competitions, having also won the competition in 2011 [1], 2012 [2], 2014 [3], 2015 [4], 2016 [5], 2017 [6], 2018 [7], and 2019 [8] while finishing second in 2013 (there was no official competition in 2020, however the team also won the Offenburg Open replacement competition that year). During the course of the competition the team won all 19 games it played and scored a total of 108 goals while conceding only 5. Many of the components of the 2021 UT Austin Villa agent were reused from the team's successful previous years' entries in the competition. This paper is not an attempt at a complete description of the 2021 UT Austin Villa agent, the base foundation of which is the team's 2011 championship agent fully described in a team technical report [9].

In addition to winning the main RoboCup 3D Simulation League competition, UT Austin Villa took second place in the RoboCup 3D Simulation League technical challenge by taking second in each of the two league challenges: fat proxy and scientific challenges. This paper serves to document these challenges as well as the main competition.

The remainder of the paper is organized as follows. In Section 2 a description of the 3D simulation domain is given highlighting differences from the previous year's competition. Section 3 provides an overview of the 2021 UT Austin Villa team and its key components, while Section 4 analyzes the overall performance of the team at the competition. Section 5 describes and analyzes the fat proxy challenge, while also documenting the results of the overall league technical challenge consisting of both the fat proxy and scientific challenges. Section 6 provides details about a new deep RL learning framework that was presented during the scientific challenge, and Section 7 concludes.

2 Domain Description

The RoboCup 3D simulation environment is based on SimSpark [10, 11], a generic physical multiagent system simulator. SimSpark uses the Open Dynamics Engine (ODE) library for its realistic simulation of rigid body dynamics with collision detection and friction. ODE also provides support for the modeling of advanced motorized hinge joints used in the humanoid agents.

Games consist of 11 versus 11 agents playing two 5 minute halves of soccer on a 30 $\times$ 20 meter field. The robot agents in the simulation are modeled after the Aldebaran Nao robot, which has a height of about 57 cm, and a mass of 4.5 kg. Each robot has 22 degrees of freedom: six in each leg, four in each arm, and two in the neck. In order to monitor and control its hinge joints, an agent is equipped with joint perceptors and effectors. Joint perceptors provide the agent with noise-free angular measurements every simulation cycle (20 ms), while joint effectors allow the agent to specify the speed/direction in which to move a joint.

Visual information about the environment is given to an agent every third simulation cycle (60 ms) through noisy measurements of the distance and angle to objects within a restricted vision cone (120°). Agents are also outfitted with noisy accelerometer and gyroscope perceptors, as well as force resistance perceptors on the sole of each foot. Additionally, agents can communicate with each other every other simulation cycle (40 ms) by sending 20 byte messages.

In addition to the standard Nao robot model, four additional variations of the standard model, known as heterogeneous types, are available for use. These variations from the standard model include changes in leg and arm length, hip width, and also the addition of toes to the robot's foot. Teams must use at least three different robot types, no more than seven agents of any one robot type, and no more than nine agents of any two robot types.

In 2019 a pass play mode was introduced to the RoboCup 3D simulation league to encourage more passing and teamwork. The pass play mode allows players some extra time on the ball to kick and pass it during which time the opponent is prevented from interfering with a kick attempt. A player may initiate the pass play mode as long as the the player is within 0.5 meters of the ball and no opponents are within a meter of the ball. Once pass play mode for a team has started the players from the opponent team are prevented from getting within a meter of the ball. The pass play mode ends as soon as a player touches the ball or four seconds have passed. After pass mode has ended the team who initiated the pass mode is unable to score for ten seconds—this prevents teams from trying to take a shot on goal directly out of pass mode. However, new for this year's competition, a team may score before 10 seconds after their pass play mode has ended if multiple players from the team have touched the ball with at least one touch coming after the ball has traveled beyond the area where opponents were not allowed to enter when pass mode was active—this allows a player to take a quick shot on goal and score after receiving a teammate's pass out of pass mode.

One significant change for the 2021 RoboCup 3D Simulation League competition was in how robots are penalized for a couple of types of fouls. Previously, if robots committed touching fouls (a group of three or robots touching at the same time), or illegal defense fouls (more than three robots inside their own goal area), robots were beamed (moved) to the sideline outside of the field of play. Both touching and illegal defense fouls were created to prevent crowding that could inhibit play and potentially make the simulator unstable due to handling a large amount of robots colliding at the same time. While enforcing these fouls is currently necessary for smooth play in RoboCup 3D simulation games, moving robots off the field for having committed the fouls looks very unnatural as they are not part of normal soccer, and could even be exploited by a team committing a foul to their advantage (e.g. a team could purposely commit an illegal defense foul during their own goal kick to-https://www.overleaf.com/project/6164550b981e83657848d0a8 have their robot moved to a forward position just outside the field to receive a pass from a goal kick). Now, instead of moving robots all the way off the field after having committed these fouls, robots are only slightly repositioned to have less of an effect on game play: robots are moved to as close a position on the field as possible to their current position where they are not touching other robots, and for illegal defense fouls robots are also moved to a position outside of their own goal area.

Figure 1 shows images of the Nao robot and soccer field during a game.

Fig. 1. A screenshot of the Nao humanoid robot (left), and a view of the soccer field during a 11 versus 11 game (right).

3 2021 UT Austin Villa Team

The UT Austin Villa team was largely unchanged from the the previous RoboCup competition in 2019, with many components developed prior to 2021 contributing to the success of the team including dynamic role assignment [12], marking [13], and an optimization framework used to learn low level behaviors for walking and kicking via an overlapping layered learning approach [14].

The primary changes to the team for this year's competition were related to pass mode: a bug was fixed that would sometimes cause an agent to take a shot on goal directly out of pass mode, and logic was added to no longer need to wait 10 seconds before trying to score after pass mode has ended in order to account for the pass mode rule change in this year's competition. The decision to keep the agent almost the same as that which was used in the previous competition was twofold. First, instead of focusing on the competition, the team decided to dedicate more time and resources to research and development of a deep RL framework detailed in Section 6. Second, keeping the team almost the same allows for it to serve as a benchmark for league progress as discussed in Section 4.

4 Main Competition Results and Analysis

In winning the 2021 RoboCup competition UT Austin Villa finished with a perfect record of 19 wins.[1] During the course of the competition the team scored 108 goals while conceding only 5. Despite the team's strong performance at the competition, the relatively few number of games played at the competition, coupled with the complex and stochastic environment of the RoboCup 3D simulator, make it difficult to determine UT Austin Villa being better than other teams by a statistically significant margin. At the end of the competition, however, all teams were required to release their binaries used during the competition. Results of UT Austin Villa playing 1000 games against each of the other eleven teams' released binaries from the competition are shown in Table 1.

UT Austin Villa finished with at least an average goal difference greater than 1.6 goals against every opponent. Additionally, UT Austin Villa's win percentage was greater than 94% against all teams except for a 78.1% win percentage against magmaOffenburg. These results show that UT Austin Villa winning the 2021 competition was far from a chance occurrence.

As mentioned in Section 3, the UT Austin Villa team was largely unchanged from the previous competition, and can thus serve as a benchmark for the progression of the league by looking at the team's relative performance against opponents between the previous and current competitions. Analysis from the 2019 competition showed that UT Austin Villa had an average goal difference of at least 2.4 goals and a winning percentage greater than 91% against all opponents [8]. Also, among the six common opponents between the 2019 and 2021 competitions (magmaOffenburg, WrightOcean, HfutEngine, FCPortugal,

[1] Full tournament results can be found at `http://www.cs.utexas.edu/~AustinVilla/?p=competitions/RoboCup21#3D`

Table 1. UT Austin Villa's released binary's performance when playing 1000 games against the released binaries of all other teams at RoboCup 2021. This includes place (the rank a team achieved at the 2021 competition), average goal difference (values in parentheses are the standard error), win-loss-tie record, and goals for/against.

Opponent	Place	Avg. Goal Diff.	Record (W-L-T)	Goals (F/A)
magmaOffenburg	2	1.612 (0.048)	781-68-151	2073/461
WrightOcean	4	2.989 (0.047)	963-4-33	3181/192
Apollo3D	3	3.119 (0.053)	941-16-43	3690/571
HfutEngine	5	3.835 (0.049)	995-1-4	4055/220
FCPortugal	6	4.106 (0.062)	975-8-17	6045/1939
Miracle3D	7	5.819 (0.048)	1000-0-0	5820/1
BahiaRT	9	6.806 (0.060)	1000-0-0	6809/3
KgpKubs	10	7.337 (0.052)	1000-0-0	7337/0
ITAndroids	8	8.031 (0.058)	1000-0-0	8033/2
MIRG	7	12.193 (0.049)	1000-0-0	9193/0
WITS-FC	11	10.552 (0.054)	1000-0-0	10552/0

BahiaRT, and ITAndroids), the opponents improved by an average goal difference of 0.97 when playing against UT Austin Villa during this year's competition. The significant overall relative improvement in performance by teams from the previous competition is a strong sign of the league progressing, and suggests that to repeat again as champions in future competitions UT Austin Villa will likely need to return focus toward improving the team's performance.

4.1 Additional Tournament Competition Analysis

To further analyze the tournament competition, Table 2 shows the average goal difference for each team at RoboCup 2021 when playing 1000 games against all other teams at RoboCup 2021.

Table 2. Average goal difference for each team at RoboCup 2021 (rows) when playing 1000 games against the released binaries of all other teams at RoboCup 2021 (columns). Teams are ordered from most to least dominant in terms of winning (positive goal difference) and losing (negative goal difference).

	UTA	mag	Apo	Wri	Hfu	FCP	Mir	ITA	Bah	Kgp	MIR	WIT
UTAustinVilla	—	1.612	3.119	2.989	3.835	4.106	5.819	8.031	6.806	7.337	9.193	10.552
magmaOffenburg	-1.612	—	1.399	1.407	2.386	4.101	4.481	5.099	6.157	5.771	9.947	8.288
Apollo3D	-3.119	-1.339	—	0.379	0.996	3.321	3.130	4.816	3.201	4.512	6.717	6.483
WrightOcean	-2.989	-1.407	-0.379	—	0.349	1.332	2.961	3.738	2.735	4.237	7.520	6.181
HfutEngine	-3.835	-2.386	-0.996	-0.349	—	0.557	2.113	4.075	1.952	3.661	6.398	6.000
FCPortugal	-4.106	-4.101	-3.321	-1.332	-0.557	—	0.664	0.567	0.544	1.125	3.454	2.772
Miracle3D	-5.819	-4.481	-3.130	-2.961	-2.113	-0.664	—	0.857	0.589	1.712	5.581	3.962
ITAndroids	-8.031	-5.099	-4.816	-3.738	-4.075	-0.567	-0.857	—	0.340	0.819	3.461	3.738
BahiaRT	-6.806	-6.157	-3.201	-2.735	-1.952	-0.544	-0.589	-0.340	—	0.606	2.462	1.387
KpgKubs	-7.337	-5.771	-4.512	-4.237	-3.661	-1.125	-1.712	-0.819	-0.606	—	0.610	0.154
MIRG	-9.193	-9.947	-6.717	-7.520	-6.398	-3.454	-5.581	-3.461	-2.462	-0.610	—	0.161
WITS-FC	-10.552	-8.288	-6.483	-6.181	-6.000	-2.772	-3.962	-2.306	-1.387	-0.154	-0.161	—

It is interesting to note that the ordering of teams in terms of winning (positive goal difference) and losing (negative goal difference) is transitive—every opponent that a team wins against also loses to every opponent that defeats that same team. Relative goal difference does not have this same property, however, as a team that does better against one opponent relative to another team does not always do better against a second opponent relative to that same team (e.g. UTAustinVilla has a higher average goal compared to magmaOffenburg when playing Apollo3D but not MIRG).

5 Technical Challenges

During the competition there was an overall technical challenge consisting of two different league challenges: scientific and fat proxy challenges. For each league challenge a team participated in, points were awarded toward the overall technical challenge based on the following equation:

$$\texttt{points}(rank) = 25 - 20 * (rank - 1)/(numberOfParticipants - 1)$$

Table 3. Overall ranking and points totals for each team participating in the RoboCup 2021 3D Simulation League technical challenge as well as ranks and points awarded for each of the individual league challenges that make up the technical challenge.

	Overall		Scientific		Fat Proxy	
Team	Rank	Points	Rank	Points	Rank	Points
magmaOffenburg	1	50	1	25	1	25
UTAustinVilla	**2**	**40**	**2**	**20**	**2**	**20**
BahiaRT	3	25	4	10	3	15
FCPortugal	4	20	2	20	—	—
WITS-FC	5	10	—	—	5	10

Table 3 shows the ranking and cumulative team point totals for the technical challenge as well as for each individual league challenge. UT Austin Villa finished second in both the scientific challenge and fat proxy challenge resulting in a second place finish in the overall technical challenge. The following subsections detail UT Austin Villa's participation in each league challenge.

5.1 Scientific Challenge

During the scientific challenge, teams give a five minute presentation on a research topic related to their team. Each team in the league then ranks the presentations with the best receiving a score of 4 (based on four teams participating in the challenge), second best a score of 3, etc. Additionally several respected research members of the RoboCup community outside the league rank the presentations, with their scores being counted double. The winner of the scientific challenge is the team that receives the highest score. Table 4 shows the results of the scientific challenge in which UT Austin Villa tied for second place.

Table 4. Results of the scientific challenge.

Team	Score
magmaOffenburg	43
FCPortugal	36
UTAustinVilla	**36**
BahiaRT	25

UT Austin Villa's scientific challenge submission[2] presented research on using a deep RL framework to learn running behaviors which is presented in detail in Section 6. The other teams participating in the scientific challenge also presented interesting work:[3] magmaOffenburg talked about learning a multi-directional kick using deep RL, FCPortugal discussed work on 6D localization [15], and BahiaRT presented a custom OpenAI Gym environment for skills optimization in the 3D Soccer Simulation League.

5.2 Fat Proxy Challenge

While strategy and teamwork is important for success in the RoboCup 3D Simulation League [12, 13], historically the teams with the best low level skills such as fast and stable walks and long and quick kicks have performed the best. Creating these low level skills serves as a barrier for entry to new teams in the league, however, as the skills can be difficult to develop with teams often employing machine learning techniques to generate them [16].

As a way to make it easier for new teams to join the league, and also to allow teams to focus on high level strategy without needing to worry about low level skills, a fat proxy[4] was created that controls the low level motions of the robots for walking, kicking, and getting up after having fallen over. The fat proxy processes all communication between agents and the simulation server, and receives messages from agents that include high level `dash` and `kick` commands for walking and kicking the ball that are similar to the same commands in the RoboCup 2D Simulation League:

```
dash <forward/backward speed> <left/right speed> <turn_angle>
kick <power> <horizontal_angle> <vertical_angle>
```

When the fat proxy receives a `dash` command from an agent, it translates the command to torques to apply to the robot's joints to have the robot walk in the direction specified in the `dash` command using the magmaOffenburg team's walk engine embedded inside the fat proxy. In the case of a `kick` command, the fat

[2] Scientific challenge entry description available at `https://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/AustinVilla3DSimulationFiles/2021/files/UTAustinVillaScientificChallenge2021.pdf`

[3] All participating teams' scientific challenge entry descriptions available at `http://archive.robocup.info/Soccer/Simulation/3D/FCPs/RoboCup/2021/`

[4] `https://github.com/magmaOffenburg/magmaFatProxy`

proxy sends a message to the simulation server to propel the ball in the direction specified in the command assuming that the agent that sent the `kick` command is close to the ball. By controlling the low level motions of the robots, the fat proxy levels the playing field such that robots from different teams all have the same set of skills.

The only change made to the UT Austin Villa agent to participate in the fat proxy challenge was to map the teams own high level commands used as input for the team's walk engine [17] and kicks [18] to that of the fat proxy's `dash` and `kick` commands. The team used the same strategy and formations for the fat proxy challenge as were used during the main competition.

Four teams participated in the fat proxy challenge which consisted of a round robin tournament where every team played every other team using the fat proxy. Teams were ranked by how many points they received (3 points for a win, 1 point for a tie, and 0 points for a loss), with the tie breaker when teams have the same number of points being the number of goals a team scored minus the number of goals they conceded. Results of the fat proxy challenge are shown in Table 5.

Table 5. Overall rank, goals scored, goals conceded, and points for each team participating in the fat proxy challenge

Team	Rank	Goals Scored	Goals Conceded	Points
magmaOffenburg	1	27	4	9
UTAustinVilla	**2**	**15**	**8**	**4**
BahaiRT	3	13	12	4
WITS-FC	4	0	31	0

UT Austin Villa took second in the challenge with magmaOffenburg winning the challenge. A noticeable difference during the challenge between UT Austin Villa and magmaOffenburg was that the UT Austin Villa team was not as stable when walking and often fell over. A likely reason for the instability is that the distribution of high level motions normally sent to the UT Austin Villa team's walk engine are different from the distribution of walk trajectories that the magmaOffenburg team's walk engine embedded in the fat proxy is tuned for. For future iterations of the fat proxy challenge the UT Austin Villa team's performance could be improved by attempting to constrain the walk commands to be closer to those normally used by the magmaOffenburg team in the magmaOffenburg team's code release.[5]

6 Deep RL Framework

Previous work from FCPortugal [19, 20] and ITAndroids [21] demonstrated that Offline Deep Reinforcement Learning is capable of learning faster walking/running behaviors than previously hand designed policies. Continuing with this line of

[5] `https://github.com/magmaOffenburg/magmaRelease`

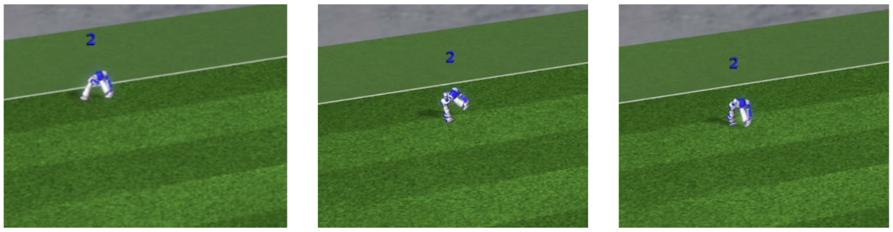

Fig. 3. Preliminary result: the agent learns to take a large step forward.

work, UT Austin Villa has been working on facilitating the use of Deep RL algorithms with the Robocup 3DSim league. As the majority of contemporary deep RL frameworks are written in Python, the UT Austin Villa team developed a custom OpenAI Gym [22] environment in Python that connected to the 3DSim platform via network sockets. The environment follows the same definition of the state and action spaces as FCPortugal [19]. The reward is -1 if the agent falls; else, 10 * forward distance of agent from time t to $t+1$. The OpenAI Gym environment wraps the 3DSim environment, enabling applying Python deep RL libraries to the existing C++ simulation software.

FCPortugal and ITAndroids used the on-policy RL algorithm, Proximal Policy Optimization (PPO) [23], to learn faster walking/running behaviors. As an on-policy RL algorithm, PPO has relatively high data requirements, which ITAndroids addressed by using parallel actors to gather data. UT Austin Villa instead investigated using the off-policy algorithm, Soft Actor Critic (SAC) [24]. Off-policy RL algorithms have demonstrated better sample-efficiency than on-policy algorithms in a variety of domains. UT Austin Villa used the implementation of SAC provided by Tianshou [25]. All experiments were performed with a single actor.

```
class Robo3DSimEnv(gym.Env):
    def __init__(self):
        # initialize the 3D simulation
        ...
    def reset(self):
        # reset the environment
        ...
        return state
    def step(self, action):
        # simulate to the next state
        ...
        return next_state, reward, done, info
```

Fig. 2. The environment APIs.

In preliminary experiments, SAC has been able to learn simple walking behaviors that allow it to move forward small amounts before falling over (See Figure 3). The preliminary learning curves are also provided in Figure 4, and the hyperparameters used for SAC are summarized in Table 6. UT Austin is hopeful that with further training SAC will be able to learn a full running behavior that can be integrated into existing play strategies.

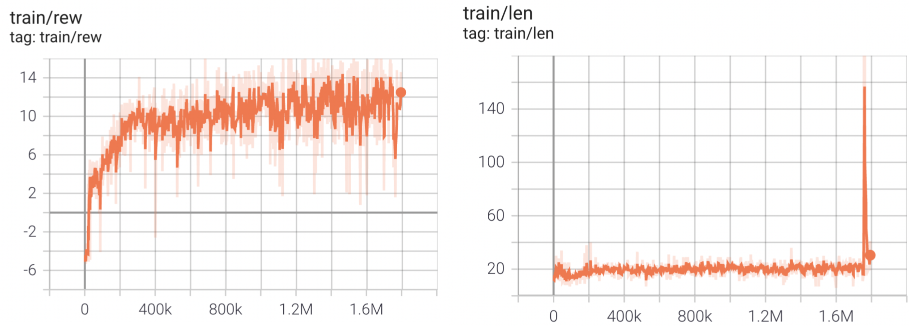

Fig. 4. The average return (**left**) and episode steps (**right**) over training.

Hyperparameters	Value
Time-step per epoch	10K
Number of epochs	2K
Learning rate of actor	0.0003
Learning rate of critic	0.0003
Learning rate of the entropy regularizer α	0.0001
Replay buffer size	1M
Critic update parameter τ	0.005
Initial α in SAC	0.2
Number of hidden layers in the neural network	128
Number of neurons in each hidden layer	2
Learning batch size	256

Table 6. Hyperparameters of the SAC algorithm for training a fast walk.

7 Conclusion

UT Austin Villa won the 2021 RoboCup 3D Simulation League main competition and finished second in the overall league technical challenge.[6] Data taken using released binaries from the competition show that UT Austin Villa winning the competition was statistically significant.

In an effort to make it easier for new teams to join the RoboCup 3D Simulation League, and also provide a resource that can be beneficial to existing teams, the UT Austin Villa team has released their base code [26].[7] This code provides a fully functioning agent and good starting point for new teams (it was used by seven out of the other eleven teams at the 2021 competition: Apollo3D, HfutEngine, KgpKubs, Miracle3D, MIRG, WITS-FC, WrightOcean). Additionally the code release offers a foundational platform for conducting research in multiple areas including robotics, multiagent systems, and machine learning.

[6] More information about the UT Austin Villa team, as well as video from the competition, released binaries, and team publications, can be found at the team's website: `http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/#2021`

[7] Code release at `https://github.com/LARG/utaustinvilla3d`

Acknowledgments

Thanks to members of the magmaOffenburg teams for creating the fat proxy for the fat proxy challenge.

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CPS-1739964, IIS-1724157, NRI-1925082, FAIN-2019844), the Office of Naval Research (N00014-18-2243), Future of Life Institute (RFP2-000), Army Research Office (W911NF-19-2-0333), DARPA, Lockheed Martin, General Motors, and Bosch. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research. Patrick MacAlpine is an employee of Sony AI America and is supported by Sony.

References

1. MacAlpine, P., Urieli, D., Barrett, S., Kalyanakrishnan, S., Barrera, F., Lopez-Mobilia, A., Ştiurcă, N., Vu, V., Stone, P.: UT Austin Villa 2011: A champion agent in the RoboCup 3D soccer simulation competition. In: Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012). (2012)
2. MacAlpine, P., Collins, N., Lopez-Mobilia, A., Stone, P.: UT Austin Villa: RoboCup 2012 3D simulation league champion. In: RoboCup-2012: Robot Soccer World Cup XVI. LNAI. Springer (2013)
3. MacAlpine, P., Depinet, M., Liang, J., Stone, P.: UT Austin Villa: RoboCup 2014 3D simulation league competition and technical challenge champions. In: RoboCup-2014: Robot World Cup XVIII. LNAI. Springer (2015)
4. MacAlpine, P., Hanna, J., Liang, J., Stone, P.: UT Austin Villa: RoboCup 2015 3D simulation league competition and technical challenges champions. In: RoboCup-2015: Robot World Cup XIX. LNAI. Springer (2016)
5. MacAlpine, P., Stone, P.: UT Austin Villa: RoboCup 2016 3D simulation league competition and technical challenges champions. In: RoboCup 2016: Robot World Cup XX. LNAI. Springer (2016)
6. MacAlpine, P., Stone, P.: UT Austin Villa: RoboCup 2017 3D simulation league competition and technical challenges champions. In: RoboCup 2017: Robot World Cup XXI. LNAI. Springer (2018)
7. MacAlpine, P., Torabi, F., Pavse, B., Sigmon, J., Stone, P.: UT Austin Villa: RoboCup 2018 3D simulation league champions. In: RoboCup 2018: Robot World Cup XXII. Lecture Notes in Artificial Intelligence. Springer (2019)
8. MacAlpine, P., Torabi, F., Pavse, B., Stone, P.: UT Austin Villa: RoboCup 2019 3D simulation league competition and technical challenge champions. In Chalup, S., Niemueller, T., Suthakorn, J., Williams, M.A., eds.: RoboCup 2019: Robot World Cup XXIII. LNAI. Springer (2019) 540–52
9. MacAlpine, P., Urieli, D., Barrett, S., Kalyanakrishnan, S., Barrera, F., Lopez-Mobilia, A., Ştiurcă, N., Vu, V., Stone, P.: UT Austin Villa 2011 3D Simulation Team report. Technical Report AI11-10, The Univ. of Texas at Austin, Dept. of Computer Science, AI Laboratory (2011)

10. Obst, O., Rollmann, M.: SPARK A Generic Simulator for Physical Multiagent Simulations. In: Multiagent System Technologies Proceedings of the MATES 2004. Volume 3187 of Lecture Notes in Artificial Intelligence. Springer (2004) 243–257
11. Xu, Y., Vatankhah, H.: Simspark: An open source robot simulator developed by the RoboCup community. In: RoboCup 2013: Robot World Cup XVII. Springer (2014) 632–639
12. MacAlpine, P., Price, E., Stone, P.: SCRAM: Scalable collision-avoiding role assignment with minimal-makespan for formational positioning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15). (2015)
13. MacAlpine, P., Stone, P.: Prioritized role assignment for marking. In: RoboCup 2016: Robot World Cup XX. LNAI. Springer (2016)
14. MacAlpine, P., Stone, P.: Overlapping layered learning. Artificial Intelligence **254** (2018) 21–43
15. Abreu, M., Silva, T., Teixeira, H., Reis, L.P., Lau, N.: 6d localization and kicking for humanoid robotic soccer. Journal of Intelligent & Robotic Systems **102** (2021)
16. MacAlpine, P., Depinet, M., Stone, P.: UT Austin Villa 2014: RoboCup 3D simulation league champion via overlapping layered learning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15). (2015)
17. MacAlpine, P., Barrett, S., Urieli, D., Vu, V., Stone, P.: Design and optimization of an omnidirectional humanoid walk: A winning approach at the RoboCup 2011 3D simulation competition. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12). (2012)
18. Depinet, M., MacAlpine, P., Stone, P.: Keyframe sampling, optimization, and behavior integration: Towards long-distance kicking in the robocup 3d simulation league. In: RoboCup-2014: Robot World Cup XVIII. LNAI. Springer (2015)
19. Abreu, M., Lau, N., Sousa, A., Reis, L.P.: Learning low level skills from scratch for humanoid robot soccer using deep reinforcement learning. In: 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), IEEE (2019) 1–8
20. Abreu, M., Reis, L.P., Lau, N.: Learning to run faster in a humanoid robot soccer environment through reinforcement learning. In: RoboCup 2019: Robot World Cup XXIII. Lecture Notes in Artificial Intelligence. Springer (2019)
21. Melo, L.C., Melo, D.C., Maximo, M.R.O.A.: Learning humanoid robot running motions with symmetry incentive through proximal policy optimization. Journal of Intelligent & Robotic Systems **102** (2021)
22. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. arXiv preprint arXiv:1606.01540 (2016)
23. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. ArXiv **abs/1707.06347** (2017)
24. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J., Krause, A., eds.: Proceedings of the 35th International Conference on Machine Learning. Volume 80 of Proceedings of Machine Learning Research., PMLR (2018) 1861–1870
25. Weng, J., Chen, H., Yan, D., You, K., Duburcq, A., Zhang, M., Su, H., Zhu, J.: Tianshou: A highly modularized deep reinforcement learning library. arXiv preprint arXiv:2107.14171 (2021)
26. MacAlpine, P., Stone, P.: UT Austin Villa RoboCup 3D simulation base code release. In: RoboCup 2016: Robot World Cup XX. LNAI. Springer (2016)