To appear in *Proceedings of the RoboCup International Symposium 2018* (RoboCup 2018), Montreal, Canada, June 2018.

UT Austin Villa: RoboCup 2018 3D Simulation League Champions

Patrick MacAlpine†⋆ and Faraz Torabi‡ and Brahma Pavse‡ and John Sigmon‡ and Peter Stone‡

† Microsoft Research, USA
patmac@microsoft.com
‡The University of Texas at Austin, USA
faraztrb@utexas.edu, brahmasp@utexas.edu, johnsigmon@utexas.edu, pstone@cs.utexas.edu

Abstract. The UT Austin Villa team, from the University of Texas at Austin, won the 2018 RoboCup 3D Simulation League, winning all 23 games that the team played. During the course of the competition the team scored 143 goals without conceding any. Additionally, the team won the RoboCup 3D Simulation League goalie challenge. This paper describes the changes and improvements made to the team between 2017 and 2018 that allowed it to win both the main competition and goalie challenge.

1 Introduction

UT Austin Villa won the 2018 RoboCup 3D Simulation League for the seventh time in the past eight years, having also won the competition in 2011 [1], 2012 [2], 2014 [3], 2015 [4], 2016 [5], and 2017 [6] while finishing second in 2013. During the course of the competition the team scored 143 goals and conceded none along the way to winning all 23 games the team played. Many of the components of the 2018 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 2018 UT Austin Villa agent, the base foundation of which is the team's 2011 championship agent fully described in a team technical report [7], but instead focuses on changes made in 2018 that helped the team repeat as champions.

In addition to winning the main RoboCup 3D Simulation League competition, UT Austin Villa also won the RoboCup 3D Simulation League goalie challenge. This paper also serves to document the goalie challenge and the approach used by UT Austin Villa when competing in the challenge.

The remainder of the paper is organized as follows. In Section 2 a description of the 3D simulation domain is given. Section 3 details changes and improvements to the 2018 UT Austin Villa team: variable distance fast walk kicks and a

⋆ The first author did the majority of the work for this publication while a postdoc at the University of Texas at Austin.

passing strategy incorporating deep learning, while Section 4 analyzes the contributions of these changes in addition to the overall performance of the team at the competition. Section 5 describes and analyzes the goalie challenge, while also documenting the overall league technical challenge consisting of both the goalie challenge and a free/scientific challenge, while Section 6 concludes.

2 Domain Description

The RoboCup 3D simulation environment is based on SimSpark [8], 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 X 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.

3 Changes for 2018

While many components developed prior to 2018 contributed to the success of the UT Austin Villa team including dynamic role assignment [9], marking [10], and an optimization framework used to learn low level behaviors for walking and kicking via an overlapping layered learning approach [11], the following subsections focus only on those that are new for 2018: variable distance fast walk kicks and a passing strategy incorporating deep learning. A performance analysis of these components is provided in Section 4.1.

3.1 Variable Distance Fast Walk Kicks

This section discusses an improvement to fast walk kicks which were first introduced for the 2017 competition. A fast walk kick is the ability of an agent to approach the ball and kick it without having to first stop and enter a stable standing position. The amount of time it takes for agents to approach and kick the ball is an important consideration as kick attempts that take longer to perform give opponents a better chance to stop them from being executed.

For the 2017 competition the UT Austin Villa team made large improvements by incorporating fast walk kicks and reducing kicking times [6]. In 2017 two different fast walk kick distances were optimized: one for long distance and a shorter distance lower height kick that would not accidentally travel over the goal when taking a shot. New for the 2018 competition, fast walk kicks were optimized for several distances in 1 meter increments from 18 meters down to 5 meters. Kicks were optimized in discrete distances in a similar manner to how the team previously optimized slower variable distance kicks [4] as opposed to learning a kicking skill that adjusts its distance [12]. Having a larger set of distances to kick the ball to provides better passing options for team play.

The UT Austin Villa team specifies kicking motions through a periodic state machine with multiple key frames, where each key frame is a parameterized static pose of fixed joint positions. Figure 1 shows an example series of poses for a kicking motion. The joint angles are optimized using the CMA-ES [13] algorithm and overlapping layered learning [11] methodologies. Kicking motion angle positions were learned for every joint—except for those controlling the position of the robot's head as we wanted to ensure it stayed looking at the ball—over each of 12 contiguous simulation cycles resulting in ≈ 260 parameters being optimized for each kick distance.

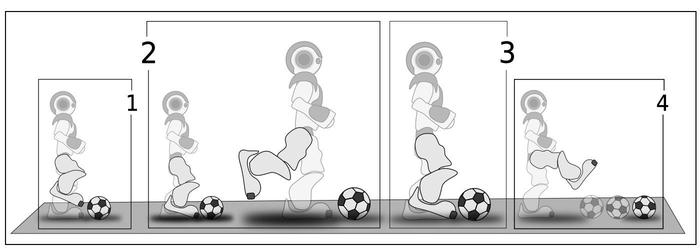


Fig. 1: Example of a fixed series of poses that make up a kicking motion.

During learning the robot runs through an optimization task where it performs ten kick attempts beginning from different positions behind the ball, with these kick attempt starting positions being at various offset angle positions one meter from the ball. For each kick attempt the robot walks toward a specific offset position behind the ball from which to execute the kicking motion—the X and Y offset positions behind the ball from which to start the kick are parameters of a kick that are also learned. Once the offset position behind the ball is reached, the robot kicks the ball toward a target position that is the desired kick distance away from the starting position of the ball in the forward direction (toward the opponent's goal) of the field. At the conclusion of a kick attempt a *fitness* value—how good the kick attempt was—is computed, and the overall fitness for a kick is the average fitness of all kick attempts using that kick. The fitness function for a kick attempt at a particular target distance is as follows:

$$fitness_{\text{dist}} = \begin{cases} -(\text{targetDistance}^2) & : \text{ Penalty} \\ -(\text{kickDistanceFromTarget}^2) & : \text{ Otherwise} \end{cases}$$

A penalty condition is one of the following: the agent fell over, the agent ran into or missed the ball, or the kick attempt took too long (over 12 seconds to make contact with the ball) and timed out. The fitness an agent receives when there is a penalty is the same as if the ball did not move during a kick attempt. A perfect kick's fitness is 0. The relative difference in fitness between kicks does not matter as CMA-ES only uses ordinal ranking of fitness values during learning.

Each variable distance fast walk kick was optimized with CMA-ES by running 300 generations with a population size of 300. The resulting fitness for most of the different distance kicks was greater than -1, meaning the average squared error of distance was less than a meter.

Longer distance kicks were learned first using initial parameter seed values from our longest 2017 pre-existing fast walk kick which can travel close to 20 meters. Kicks were learned in descending order of distance, and as new shorter distance kicks were learned they were then used as seeds for even shorter kicks.

3.2 Deep Learning Passing Strategy

Before the 2018 competition, we used the hand-tuned heuristic scoring function shown in Equation 1 to decide where to kick the ball for a pass. The equation rewards kicks that move the ball towards the opponent's goal, penalizes kicks that move the ball near opponents, and rewards kicks that move the ball near a teammate. All distances in Equation 1 are measured in meters. A primary reason for Equation 1's effectiveness is that it *efficiently* evaluates the value of different kicking locations.

$$\texttt{score}(target) = \begin{array}{l} -\|opponentGoal - target\| \\ \forall opp \in Opponents, -.5 * \max(64 - \|opp - target\|^2, 0) \\ -.5 * \max(64 - \|closestOpponentToTarget - target\|^2, 0) \\ + \max(10 - \|closestTeammateToTarget - target\|, 0) \end{array} \quad (1)$$

While efficient and successful, Equation 1 is potentially very limited. Firstly, it does not capture the specific positions of players from the kick target. Secondly, the heuristic's restrictive nature forces us to use a different hand-tuned scoring function to handle set plays such as kick-offs. In an effort to tackle these limitations, we used a deep learning based approach for RoboCup 2018.

In our approach, we determine the value of potential passing locations by training a value network. While we evaluate the performance of our network in regular gameplay scenarios, we have trained our network using a supervised learning problem formulation with only indirect kick data against various teams in the league.

Let the total data set S of size m be $\{(x^i, y^i)\}_{i=1}^m$. A single input, x^i, to the network is a 49 dimensional feature vector representing the state of the game ie: the play mode, the coordinates of 22 player locations, ball location, and potential pass location. The output, y^i, of the network is a single scalar value between $[0, 1]$ that denotes the value of the potential pass location. During our data collection process, we determine a single y^i by repeatedly restoring the state according to x^i ten times. In each of these restorations, the team receives a reward of $+1$ if it scores a goal within 20 seconds, else it receives a reward of 0. The average reward of these ten runs is y^i. Naturally, for each configuration of player and ball locations, there are many valid passing locations; hence, there are many training examples for a single configuration. Here, a valid location is one that is at most 20 m away from the initial ball position and is within the field bounds.

Furthermore, the data was augmented in the following manner:

1. The input into the network is organized in a canonical representation. Specifically, we sort players based on the x coordinates from the left to right of the field.
2. We also pre-process the data to ensure symmetry, which augments our data. Along the y axis, we ensure that inputs into the neural network are such that the y coordinate of the ball is positive by flipping all the y coordinates of the input if the y coordinate of the ball is negative. This allows us to reduce the number of possible data examples by half, which allows us to converge faster.

Training Given that a large network can overfit and be computationally expensive, the best size neural network was based on two factors - its potential to overfit and its compliance with the 20 ms cycle time constraint. Table 1 shows the various fully connected network capacities tested along with their computational cost related metrics.

Table 1: The average range of time taken, max time taken, and max packets missed during a *single* forward pass for different networks. Time units are in milliseconds. Bold indicates selected network.

	Neurons per layer	Avg. Range of Time Cost	Max Time Cost	Max Missing Packets
1	128 128 64 32 1	0.015 - 0.04	0.095	0
2	256 128 64 32 1	0.03 - 0.06	0.134	0
3	**128 128 64 32 16 1**	**0.017 - 0.05**	**0.102**	**0**
4	512 256 128 64 32 1	~ 0.15	~ 0.2	5

Ultimately, we employed network 3 (in bold) for RoboCup 2018, since its large size would enhance the network's ability to represent complicated functions as well as not cause any agent to miss packets.

Below are the training specifics for network 3:

– Training was offline with the data collected by the method described earlier.

- Data set size: $\sim$ 4600 states. Nearly $\sim$ 772000 training examples after augmentation. Network was explicitly trained to handle indirect kicks.
- Training/Test split: 90% and 10%.
- Update Algorithm: Backpropagation.
- Loss Function: Mean Squared Error of the predicted values and true values for a given kick location.
- Optimizer: Adam Optimizer [14].
- Epochs: 10000.
- Architecture: 5 hidden layers with $128, 128, 64, 32, 16, 1$ neurons respectively.
- Activation function: Leaky ReLU.
- Weight initialization: Xavier.
- Learning rate: 0.00001.
- Regularization parameter: 0.00025.
- Mini-batch gradient descent: 64 batch size.
- Deep Learning Framework: Tensorflow

Once the network is trained, it performs online evaluation of potential passing locations, and an agent kicks to the location with the highest value.

4 Main Competition Results and Analysis

In winning the 2018 RoboCup competition UT Austin Villa finished with a perfect record of 23 wins and no losses.[1] During the competition the team scored 143 goals while conceding none. Despite finishing with a perfect record, 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 six teams' released binaries from the competition are shown in Table 2.

Table 2: UT Austin Villa's released binary's performance when playing 1000 games against the released binaries of all other teams at RoboCup 2018. This includes place (the rank a team achieved at the 2018 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	2.648 (0.047)	939-4-57	2708/60
FCPortugal	3	4.572 (0.055)	997-0-3	5071/499
BahiaRT	4	6.734 (0.057)	1000-0-0	6735/1
KgpKubs	5	6.586 (0.052)	1000-0-0	6586/0
Miracle3D	6	5.878 (0.048)	1000-0-0	5878/0
ITAndroids	7	9.104 (0.058)	1000-0-0	9104/0

UT Austin Villa finished with at least an average goal difference greater than 2.6 goals against every opponent. UT Austin Villa's strong defense and use of

[1] Full tournament results can be found at `http://www.cs.utexas.edu/~AustinVilla/?p=competitions/RoboCup18#3D`

marking [10] limited opponent scoring opportunities, and half the opponents were unable to score any goals against UT Austin Villa. The only team to score more than 100 goals during the 1000 games played against UT Austin Villa was FCPortugal with 499, and of those 452 (over 90%) were scored from a kickoff set play the FCPortugal team developed that allowed for an almost immediate and unblockable shot on goal. Additionally, UT Austin Villa won all but 60 games that ended in ties, and 4 games that ended in losses, out of the 6000 that were played in Table 2 with a win percentage greater than 93% against all teams. These results show that UT Austin Villa winning the 2018 competition was far from a chance occurrence. The following subsection analyzes the contribution of the new variable distance fast walk kicks and deep learning passing strategy components (described in Section 3) to the team's dominant performance.

4.1 Analysis of Components

To analyze the contribution of new components for 2018—variable distance fast walk kicks and a deep learning passing strategy (Section 3)—to the UT Austin Villa team's performance, we played 1000 games between a version of the 2018 UT Austin Villa team with each of these components turned off—and no other changes—against each of the RoboCup 2018 teams' released binaries. Results comparing the performance of the UT Austin Villa team with and without using these components are shown in Table 3.

Table 3: Different versions of the UTAustinVilla team when playing 1000 games against the released binaries of all teams at RoboCup 2018. Values shown are average goal difference with values in parentheses being the difference in performance from the team's released binary.

Opponent	Released Binary	No Var. Dist. Walk Kicks	No Deep Learn Pass Str.
UTAustinVilla	0[a]	-0.073 (-0.073)	-0.114 (-0.114)
magmaOffenburg	2.648	2.525 (-0.123)	2.441 (-0.207)
FCPortugal	4.572	4.478 (-0.094)	4.458 (-0.114)
Miracle3D	5.878	6.139 (+0.261)	6.133 (+0.255)
KgpKubs	6.586	6.371 (-0.215)	6.746 (+0.160)
BahiaRT	6.734	6.828 (+0.094)	6.655 (-0.079)
ITAndroids	9.104	8.982 (-0.122)	9.113 (+0.009)

[a] Games were not played, but assumed to be an average goal difference of 0 in expectation with self play.

Results are mixed in terms of improved performance against the other teams' released binaries when using variable distance walk kicks and our deep learning passing strategy. Both new components help against the top three teams (UTAustinVilla, magmaOffenburg, and FCPortugal), however, which is good as improved performance is more important against better teams. It might be the case that a larger set of passing location options coupled with a better decision on where to pass the ball is beneficial against more skilled teams, while against less skilled teams the best strategy is just to kick the ball as far as possible down the field and then run after it.

4.2 Additional Tournament Competition Analysis

To further analyze the tournament competition, Table 4 shows the average goal difference for each team at RoboCup 2018 when playing 1000 games against all other teams at RoboCup 2018.

Table 4: Average goal difference for each team at RoboCup 2018 (rows) when playing 1000 games against the released binaries of all other teams at RoboCup 2018 (columns). Teams are ordered from most to least dominant in terms of winning (positive goal difference) and losing (negative goal difference).

	UTA	mag	FCP	Bah	Kgp	Mir	ITA
UTAustinVilla	—	2.648	4.572	6.734	6.586	5.878	9.104
magmaOffenburg	-2.648	—	0.376	2.710	2.567	4.853	4.171
FCPortugal	-4.572	-0.376	—	1.804	2.298	4.598	2.826
BahiaRT	-6.734	-2.710	-1.804	—	0.581	1.761	1.266
KgpKubs	-6.586	-2.576	-2.298	-0.581	—	0.527	0.207
Miracle3D	-5.878	-4.853	-4.598	-1.761	-0.527	—	0.022
ITAndroids	-9.104	-4.171	-2.286	-1.266	-0.207	-0.022	—

It is interesting to note that the ordering of teams in terms of winning (positive goal difference) and losing (negative goal difference) is strictly dominant—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. UT Austin Villa is dominant in terms of relative goal difference, however, as UT Austin Villa has a higher goal difference against each opponent than all other teams against the same opponent.

5 Technical Challenges

During the competition there was an overall technical challenge consisting of two different league challenges: free and goalie challenge. 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 5 shows the ranking and cumulative team point totals for the technical challenge as well as for each individual league challenge. UT Austin Villa won the goalie challenge and finished third in the free challenge resulting in a third place finish in the overall technical challenge. The following subsections detail UT Austin Villa's participation in each league challenge.

5.1 Free Challenge

During the free challenge, teams give a five minute presentation on a research topic related to their team. Each team in the league then ranks the presentations

Table 5: Overall ranking and points totals for each team participating in the RoboCup 2018 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		Free		Goalie	
Team	Rank	Points	Rank	Points	Rank	Points
magmaOffenburg	1	42	1	25	3-5	17[a]
FCPortugal	2	32	2	15	3-5	17[a]
UTAustinVilla	**3**	**30**	**3**	**5**	**1**	**25**
KgpKubs	4	21	—	—	2	21
BahiaRT	5	17	—	—	3-5	17[a]
ITAndroids	6	5	—	—	6	5

[a] Results released from the competition awarded tied teams the points total for the highest rank of the range they finished within. As determining points totals for tied teams was not explicitly specified in the rules, we contend the proper award is instead the average points total across the range of tied for places as is the case for the Standard Platform League technical challenge rules (`https://spl.robocup.org/wp-content/uploads/downloads/Challenges2013.pdf`) from which the 3D Simulation League scoring system was derived. Teams receive 13 points using the average points total for the range of positions they are tied for, with resulting overall scores being magmaOffenburg 38, FCPortugal 28, and BahiaRT 13.

with the best receiving a score of 1 votes, second best a score of 2, 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 free challenge is the team that receives the lowest score. Table 6 shows the results of the free challenge in which UT Austin Villa was awarded third place.

Table 6: Results of the free challenge.

Team	Votes
magmaOffenburg	14
FCPortugal	16
UTAustinVilla	**18**

UT Austin Villa's free challenge submission[2] presented the team's use of deep learning to develop a passing strategy discussed in Section 3.2. The magmaOffenburg team talked about learning model-free behaviors [15], and the FCPortugal team presented a hybrid ZMP-CPG based walk engine for biped robots [16].

5.2 Goalie Challenge

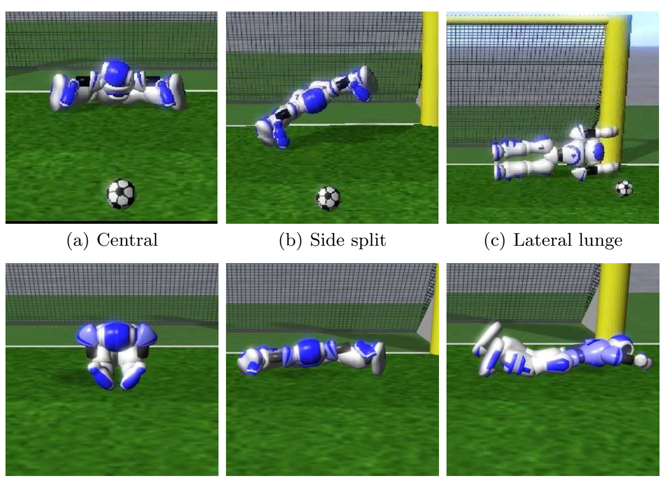
A goalie challenge[3] was held where a goalie faces 12 shots from random starting positions on the field, and then is given a score for the percentage of shots the

[2] Free challenge entry description available at `http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/AustinVilla3DSimulationFiles/2018/files/UTAustinVillaFreeChallenge2018.pdf`

[3] Framework for running the goalie challenge at `https://github.com/magmaOffenburg/magmaChallenge`

goalie is able to stop. Starting positions of shots range in one meter increments from 3 to 15 meters in the forward direction from the goal, and in one meter increments from 0 to 9 meters toward each side of the goal. Target locations for shots are either the center or toward either side of the goal. There are two different shot speeds: slow and fast, and an initial Z velocity as an integer from 0-5 meters per second is added to a shot to determine its height. Given the different shot starting positions, target locations, and velocities, there are a total of 8892 possible shots. Some of the possible shots go over the goal and miss, however, and so for the competition only the shots that will score on an empty goal (8316 possible different shots) are used. At the beginning of the challenge a random seed is selected to determine which 12 shots will be used during the challenge. If after the conclusion of the challenge more than one team has the same score, those teams face a second set of different shots to serve as a tie breaker.

The UT Austin Villa team's goalie positions itself to block shots and has three separate goalie diving behaviors for if the ball is kicked straight at, a little to the side, and further to the side of the goalie as described in [7]. Figure 2 shows screenshots of these dives. The diving behaviors consist of a series of fixed poses parameterized by different joint angles. Prior to this year's competition the team's diving behaviors were only hand-designed and hand-tuned. Once on-site at the competition the team decided to optimize these goalie dives for the goalie challenge. Using a training task consisting of a subset of 360 shots chosen to be well distributed across the set of all possible challenge shots, 84 joint angle parameters for the goalie dives were optimized across 200 generations of the CMA-ES [13] algorithm with a population size of 150. After learning, the new goalie dives were able to stop 46.6% of all 8000+ possible shots as compared to being able to stop only 36.4% of shots before learning. These new goalie dives were also added to and used by the goalie during the final rounds of the main RoboCup competition.



(a) Central (b) Side split (c) Lateral lunge

(d) Central optimized (e) Side split optimized (f) Lateral lunge optimized

Fig. 2: Screenshots of the original hand-tuned (a-c) and optimized (d-f) goalie diving behaviors.

Results of the goalie challenge are shown in Table 7. UT Austin Villa won the challenge by saving 50% of the shots the goalie faced which is twice as many as any of the other teams competing in the challenge.

Table 7: Scores for each of the teams competing in the goalie challenge.

Team	Score	Score Tie Breaker
UTAustinVilla	**0.50**	—
KgpKubs	0.25	0.08
BahiaRT	0.25	0.00
FCPortugal	0.25	0.00
magmaOffenburg	0.25	0.00
ITAndroids	0.08	—

6 Conclusion

UT Austin Villa won the 2018 RoboCup 3D Simulation League main competition as well as the goalie challenge.[4] Data taken using released binaries from the competition show that UT Austin Villa winning the competition was statistically significant. The 2018 UT Austin Villa team also improved from 2017 as it was able to beat the team's 2017 champion binary by an average of 0.171 (+/- 0.042) goals across 1000 games.

In an effort to both 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 [17].[5] This code release provides a fully functioning agent and good starting point for new teams to the RoboCup 3D Simulation League (it was used by two other teams at the 2018 competition: KgpKubs and Miracle3D). Additionally the code release offers a foundational platform for conducting research in multiple areas including robotics, multiagent systems, and machine learning.

Acknowledgments

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-1637736, IIS-1651089, IIS-1724157), ONR (N00014-18-2243), FLI (RFP2-000), DARPA, Intel, Raytheon, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. 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.

[4] More information about the UT Austin Villa team, as well as video from the competition, can be found at the team's website: `http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/#2018`

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

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 Soccer 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 Soccer 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 Soccer 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 Soccer World Cup XXI. LNAI. Springer (2018)
7. 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)
8. 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
9. 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)
10. MacAlpine, P., Stone, P.: Prioritized role assignment for marking. In: RoboCup 2016: Robot Soccer World Cup XX. LNAI. Springer (2016)
11. MacAlpine, P., Stone, P.: Overlapping layered learning. Artificial Intelligence **254** (2018) 21–43
12. Abdolmaleki, A., Simoes, D., Lau, N., Reis, L.P., Neumann, G.: Learning a humanoid kick with controlled distance. In: RoboCup 2016: Robot Soccer World Cup XX. LNAI. Springer (2016)
13. Hansen, N.: The CMA Evolution Strategy: A Tutorial. (2009) `http://www.lri.fr/~hansen/cmatutorial.pdf`.
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
15. Baur, M., Christmann, K., Dorer, K., Fischer, J., Grüßinger, M., Schmider, C., Weiler, D.: The magmaOffenburg 2018 RoboCup 3D simulation team. In: RoboCup 2018 Symposium and Competitions: Team Description Papers. (2018)
16. Kasaei, S.M., Simões, D., Lau, N., Pereira, A.: A hybrid zmp-cpg based walk engine for biped robots. In: ROBOT 2017: Third Iberian Robotics Conference, Cham, Springer International Publishing (2018) 743–755
17. MacAlpine, P., Stone, P.: UT Austin Villa robocup 3D simulation base code release. In: RoboCup 2016: Robot Soccer World Cup XX. LNAI. Springer (2016)