To appear in *Proceedings of the RoboCup International Symposium 2016* (RoboCup 2016), Leipzig, Germany, July 2016.

UT Austin Villa: RoboCup 2016 3D Simulation League Competition and Technical Challenges Champions

Patrick MacAlpine and Peter Stone

Department of Computer Science, The University of Texas at Austin, USA
{patmac,pstone}@cs.utexas.edu

Abstract. The UT Austin Villa team, from the University of Texas at Austin, won the 2016 RoboCup 3D Simulation League, winning all 14 games that the team played. During the course of the competition the team scored 88 goals and conceded only 1. Additionally the team won the RoboCup 3D Simulation League technical challenge by winning each of a series of three league challenges: free, keepaway, and Gazebo running challenge. This paper describes the changes and improvements made to the team between 2015 and 2016 that allowed it to win both the main competition and each of the league technical challenges.

1 Introduction

UT Austin Villa won the 2016 RoboCup 3D Simulation League for the fifth time in the past six years, having also won the competition in 2011 [1], 2012 [2], 2014 [3], and 2015 [4] while finishing second in 2013. During the course of the competition the team scored 88 goals and only conceded 1 along the way to winning all 14 games the team played. Many of the components of the 2016 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 2016 UT Austin Villa agent, the base foundation of which is the team's 2011 championship agent fully described in a team technical report [5], but instead focuses on changes made in 2016 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 technical challenge by winning each of the three league challenges: free, keepaway, and Gazebo running challenge. This paper also serves to document these challenges and the approaches used by UT Austin Villa when competing in the challenges.

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 2016 UT Austin Villa team (including those for marking, getting open, tuning kick selection for height, indirect kick set plays, and directional kicks), 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 league challenges that were used to determine the winner of the technical challenge, and Section 6 concludes.

2 Domain Description

The RoboCup 3D simulation environment is based on SimSpark [6], 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.

The 2016 RoboCup 3D Simulation League competition included a couple key changes from the previous year's competition. The first of these was a rule change to make previously direct kick-ins—awarded to the opposing team when a player is the last to touch the ball before it goes out of bounds over the sideline—indirect (another player other than the player who took the kick must touch the ball before a goal can be scored). This rule was put in place to encourage both passing and teamwork, and to make the rules for kick-ins more similar to their equivalent in human soccer of indirect throw-ins. Coupled with this rule change, and also to match human soccer, an indirect kick is awarded to the opposing team if a player taking a kick (direct or indirect) touches the ball twice in a row.

The second change to the RoboCup 3D Simulation League was to add an automated referee for calling charging fouls. Players are deemed to have committed a charging foul, and are beamed outside the field of play as punishment, if when colliding with an opposing player their force (velocity component in the direction of the opposing player) exceeds a tunable threshold. Charging is not called on a player if the player has recently touched the ball—thus deemed to be going for the ball—or if two players are charging into each other at the same time. Calling charging fouls improves play by reducing the number of collisions while at the same time better matching the rules of human soccer.

3 Changes for 2016

While many components contributed to the success of the UT Austin Villa team, including dynamic role assignment [7] and an optimization framework used to learn low level behaviors for walking and kicking via an overlapping layered learning approach [8], the following subsections focus only on those that are new for 2016. A performance analysis of these components is provided in Section 4.1.

3.1 Marking

During the 2015 RoboCup competition it was noticed that teams had developed good kicking/passing abilities, and then were exploiting these abilities through set plays by passing the ball to open teammates in good positions to take a shot on goal. To counteract set plays, UT Austin Villa developed and employed a marking system for the 2016 RoboCup competition to cover and defend against opponents in dangerous offensive positions from receiving passes.

The marking system implemented by the UT Austin Villa team is a sequential process encompassing the following three steps:

1. Decide which players to mark
2. Select which roles to use for marking purposes
3. Use prioritized role assignment to assign players to positions

In the first step, hand-coded heuristics are used to decide which opponents are in dangerous offensive positions and should be marked. Next, from the team's default formation positions, computed as offset positions from the ball using Delaunay triangulation [9], a set of formation positions are selected to be replaced with positions needed for marking. The set of formation positions to be replaced, computed by the Hungarian algorithm [10], is that which minimizes the sum of distances between positions for marking and the formation positions that are being replaced by the marking positions. Finally, prioritized role assignment [11], an extension of SCRAM role assignment [7], is used to assign agents to move to the team's desired role positions (both those for formation and marking purposes). A full description of the marking system can be found in [11].

3.2 Getting Open

In 2015 UT Austin Villa optimized variable distance kicks [4] that allow for the ball to be kicked different distances in one meter increments. These kicks provide passing options as robots can select from many potential targets to kick the ball to. Each potential kick location is given a score according to Equation 1, and the location with the highest score is chosen as the location to kick the ball to. Equation 1 penalizes kick locations that are farther from the opponent's goal, penalizes kicks that have the ball end up near opponents, and also rewards kicks for landing near a teammate. All distances in Equation 1 are measured in meters.

$$\texttt{score}(location) = \begin{array}{l} -\|centerOfOpponentGoal - location\| \\ \forall opp \in Opponents, -\max(25 - \|opp - location\|^2, 0) \\ +\max(10 - \|closestTeammateToTarget - location\|, 0) \end{array} \quad (1)$$

Ideal locations for kicking the ball are those near a teammate who is in an offensive and open (no opponents are close by) position on the field to receive a pass and take a shot on goal. New for the RoboCup 2016 competition, rather than waiting for a teammate to broadcast a location the ball is going to be kicked to, and only then moving to that position [3], a robot assuming the center forward or striker role [2] in the team's formation moves toward the position that it calculates its teammate closest to the ball is anticipated to potentially kick the ball to using Equation 1. As Equation 1 gives higher values for locations away from opponents, this behavior results in the striker dynamically moving to open offensive positions on the field rather than the previous behavior of standing at a fixed location relative to the ball—possibly close to opponents marking that player—as it waits to hear from a teammate where the ball might next be kicked.

3.3 Tuning Kick Selection for Height

In addition to allowing for more precise passing, variable distance kicks [4] are also useful for taking shots on goal. Generally speaking, the greater the distance a kick travels the longer and higher the ball may travel in the air, and possibly over the goal, when shooting. During the 2015 RoboCup competition, to try and prevent accidentally shooting the ball over the goal, kicks for shooting on goal were limited to those traveling no more than 7 meters in distance beyond the goal line. Despite this attempt at limiting the distance for kicks when shooting, a number of shots still missed their target by flying above the goal.

Table 1: The maximum forward distance measured across 100 kicks for different kick types (kicks for different distances) during which the ball is above the height of the goal. All values are in meters.

Kick Type (Distance in Meters)	10	11	12	13	14	15	16	17	18	19	20
Max Above Goal Height Distance	4.00	4.50	6.25	10.50	6.00	7.50	8.75	9.75	10.50	15.50	16.00

For the 2016 RoboCup competition, the UT Austin Villa team took a more calibrated approach to preventing shots from going over the goal. Instead of estimating 7 meters as the difference between a kick's distance and the maximum forward distance the ball can travel at a height above the goal for all kick types, the maximum forward distance the ball is above the height of the goal was directly measured across 100 kick attempts for each kick type. Table 1 shows these measurements for different kicks for an agent with a type four body model (an agent with the standard Nao model but with toes added). Using these measurements, the kick with maximum power that will not fly above the goal—the kick with the maximum distance, but whose forward above goal height distance is less than the ball's distance to the goal—is selected when shooting on goal.

From the data in Table 1 it can be seen that there is not a constant offset difference between the distance of a kick and the maximum forward distance that the ball can be above the height of the goal. Furthermore, as for many of the kick types this difference is less than the 7 meters estimated value used during the 2015 competition, it is not surprising that shots went over the goal during last year's competition.

3.4 Indirect Kick Set Plays

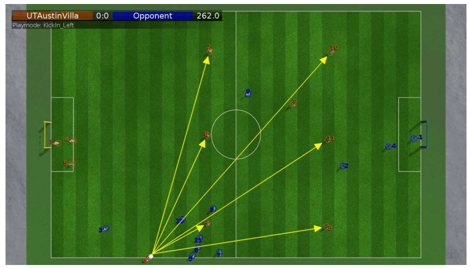

Fig. 1: Indirect kick formation for set plays during a kick-in. Yellow lines represent potential passes to teammates in the formation.

With this year's rule changes (outlined in Section 2) making kick-ins indirect, and double touches on kicks result in an indirect free kick for the opposing team, it becomes practical to use set plays for indirect kicks. During indirect kicks the team switches to a formation shown in Figure 1, where three players are spread in a line across the field near the halfway line, and another three players are spread in a line across the field in a more forward offensive position. The player taking the kick, now with more options from the formation as to which teammate to pass the ball to, chooses to kick the ball to the best location as decided by Equation 1 in Section 3.2. The player taking the kick also waits 10 seconds from the start of an indirect kick before kicking the ball so as to give its teammates time to move to their assigned positions in the indirect kick formation.

3.5 Directional Kicks

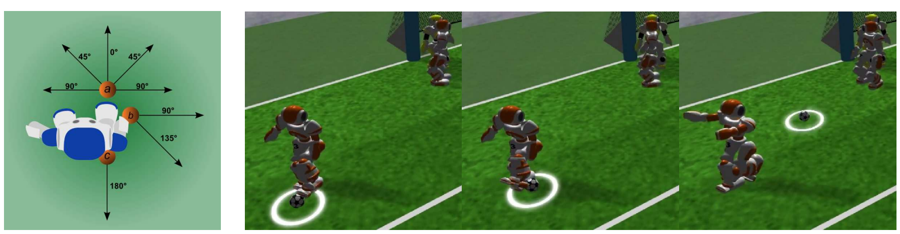

Fig. 2: Possible directions to kick the ball with respect to the placement of the ball at *a*, *b*, and *c* (left image). Agent with its back to the goal taking a shot before scoring at RoboCup 2016 using a 180° "hook" kick where the ball is pulled backward and then kicked behind the agent (right images).

Directional kicks incorporating inverse kinematics created for the 2011 competition [1], but whose use was later discontinued due to poor performance, were reintroduced and integrated into agents with a type four body model (the standard Nao model but with toes added) using overlapping layered learning [8]. Unlike a standard forward kick, where an agent must take the time to line up behind the ball in the direction that it is to be kicked, directional kicks allow for an agent to quickly approach and kick the ball from multiple angles and relative ball positions (example ball positions and kick directions are shown in Figure 2).

4 Main Competition Results and Analysis

In winning the 2016 RoboCup competition UT Austin Villa finished with a perfect record of 14 wins and no losses.[1] During the competition the team scored 88 goals while only conceding 1. 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 eight 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 2016. This includes place (the rank a team achieved at the 2016 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)
FUT-K	2	1.809 (0.036)	888-3-109	1872/63
FCPortugal	3	2.431 (0.040)	954-1-45	2452/21
BahiaRT	4	3.123 (0.040)	985-0-15	3123/0
magmaOffenburg	5	3.921 (0.049)	996-0-4	3926/5
KgpKubs	8	7.728 (0.046)	1000-0-0	7729/1
ITAndroids	6	9.022 (0.053)	1000-0-0	9024/2
HfutEngine3D	9	10.192 (0.056)	1000-0-0	10192/0
Miracle3D	7	11.126 (0.059)	1000-0-0	11126/0

UT Austin Villa finished with at least an average goal difference greater than 1.8 goals against every opponent. Additionally UT Austin Villa only lost 4 games out of the 8000 that were played in Table 2 with a win percentage greater than 88% against all teams. This shows that UT Austin Villa winning the 2016 competition was far from a chance occurrence. The following subsection analyzes some of the components described in Section 3 that contributed to the team's dominant performance.

4.1 Analysis of Components

Table 3: Average goal difference and goals against (with percentage of goals against scored off set plays in parentheses) achieved by versions of the UT Austin Villa team with and without marking when playing 1000 games against the top three teams at RoboCup 2016.

	Average Goal Difference		Goals Against (Set Play %)	
Opponent	Marking	No Marking	Marking	No Marking
UTAustinVilla	0.001	-0.336	290 (18.62%)	667 (61.67%)
FUT-K	1.809	1.613	63 (25.40%)	288 (66.67%)
FCPortugal	2.431	2.361	21 (23.81%)	160 (81.88%)

1000 games were played between a version of the UT Austin Villa team with marking (Section 3.1) turned off and each of the RoboCup 2016 teams' released

[1] Full tournament results can be found at `https://robocup.info/xml/RC2016_SS3D_tournament.xml`

binaries. Only the top three teams from the competition showed any appreciable difference in the number of goals scored against the team without marking when compared to the same team with marking (no other teams were able to score more than 6 goals even with marking turned off). Results against the top three teams are shown in Table 3. Against all three teams the average goal difference went down when not using marking. Additionally both the number of goals against, and the percentage of goals scored off set plays,[2] increased drastically without marking. These results show marking provides a substantial defensive improvement, and furthermore is very effective in defending against set plays.

Table 4: Average goal difference achieved by different versions of the UT Austin Villa team when playing 9000 games against all teams at RoboCup 2016 (1000 games against each opponent).

UTAustinVilla	NoGettingOpen	NoKickHeightTuning	NoIndirectKickSetPlays	DirectionalKicks
5.484	5.430	5.331	5.459	3.815

Table 4 shows the average goal difference achieved by the following different versions of the UT Austin Villa team when playing 9000 games against all teams at RoboCup 2016 (1000 games against each opponent).

UTAustinVilla Released binary (no directional kicks).
NoGettingOpen Agent assigned to striker role does not try and get open.
NoKickHeightTuning Kick selection not tuned for height when shooting.
NoIndirectKickSetPlays No indirect kick set plays.
DirectionalKicks Uses directional kicks.

As the removal of most components in Table 4 decreased average goal difference, these components were beneficial to the team. Having the player assigned to the striker role move to get open (Section 3.2) slightly improves performance and additionally increased the average number of goals scored against all opponents from 5.476 to 5.526. Tuning which kick is selected when taking a shot based on height (Section 3.3) also increased the average goal difference. This improvement is likely due to no longer kicking the ball over the goal as the average shot scoring percentage against all opponents increased from 29.69% without tuning to 32.05% with kick height tuning: the ability to kick the ball with just the right amount of power such that it flies into the goal—but not over it—is a valuable skill during games. While using indirect kick set plays (Section 3.4) showed only a very slight increase in average goal difference, the average scoring percentage on indirect kicks jumped from 27.10% to 34.78% when using set plays.

The only change that hurts performance is the use of directional kicks. It was known during the competition that choosing to pass the ball with a directional kick instead of using a longer kick to take a shot on goal decreased performance, and so directional kicks were disabled during the semifinals and finals rounds. Directional kicks do however increase the average number of pass attempts in a game from 13.020 to 15.838, and they raise the average possession (a teammate is closer than any opponent to the ball) percentage time from 56.71% to 59.81%. Directional kicks may become more useful if teams' goalie behaviors improve resulting in the need to be more selective when choosing when to shoot on goal.

[2] Goals within 20 secs of corner kicks, 25 secs of kick-ins and indirect free kicks, and 35 secs from start of kickoffs. Times reduced by 5 secs against UTAustinVilla opponent.

4.2 Additional Tournament Competition Analysis

Table 5: Average goal difference for each team at RoboCup 2016 (rows) when playing 1000 games against the released binaries of all other teams at RoboCup 2016 (columns). Teams are ordered from most to least dominant in terms of winning (positive goal difference) and losing (negative goal difference).

	UTA	FUT	FCP	mag	Bah	ITA	Kgp	Mir	Hfut
UTAustinVilla	—	1.809	2.431	3.921	3.123	9.022	7.728	11.126	10.192
FUT-K	-1.809	—	0.708	2.424	1.477	6.692	5.799	8.795	7.535
FCPortugal	-2.431	-0.708	—	0.756	1.022	4.343	5.127	7.204	5.535
magmaOffenburg	-3.921	-2.424	-0.756	—	0.154	3.624	4.080	5.908	4.742
BahiaRT	-3.123	-1.477	-1.022	-0.154	—	3.416	4.661	6.067	4.556
ITAndroids	-9.022	-6.692	-4.343	-3.624	-3.416	—	0.163	1.939	1.096
KgpKubs	-7.728	-5.799	-5.127	-4.080	-4.661	-0.163	—	0.253	1.391
Miracle3D	-11.126	-8.795	-7.204	-5.908	-6.067	-1.939	-0.253	—	0.800
HfutEngine3D	-10.192	-7.535	-5.535	-4.742	-4.556	-1.096	-1.391	-0.800	—

To further analyze the tournament competition, Table 5 shows the average goal difference for each team at RoboCup 2016 when playing 1000 games against all other teams at RoboCup 2016. 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

For the third straight year there was an overall technical challenge consisting of three different league challenges: free, keepaway, and Gazebo running 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 6: Overall ranking and points totals for each team participating in the RoboCup 2016 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		Keepaway		Gazebo Running	
Team	Rank	Points	Rank	Points	Rank	Points	Rank	Points
UTAustinVilla	**1**	**75**	**1**	**25**	**1**	**25**	**1**	**25**
FCPortugal	2	53.33	2	20	3	15	2	18.33
magmaOffenberg	3	36.67	3	15	4	10	3	11.67
FUT-K	4	30	5	5	2	20	4	5
BahiaRT	5	15	4	10	5	5	—	—

Table 6 shows the ranking and cumulative team point totals for the technical challenge as well as for each individual league challenge. UT Austin Villa earned the most points and won the technical challenge by taking first in each of the league challenges. The following subsections detail UT Austin Villa's participation in each league challenge.[3]

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 top five presentations with the best receiving 5 votes and the 5th best receiving 1 vote. Additionally several respected research members of the RoboCup community outside the league vote, with their votes being counted double. The winner of the free challenge is the team that receives the most votes. Table 7 shows the results of the free challenge in which UT Austin Villa was awarded first place.

Table 7: Results of the free challenge.

Team	Votes
UTAustinVilla	**37**
FCPortugal	35
magmaOffenburg	33
BahiaRT	31
FUT-K	14

UT Austin Villa's free challenge submission[4] was on the team's marking system [11] discussed in Section 3.1. The FC Portugal team presented research on a learning algorithm, originally designed to learn a parameterized kick for kicking the ball different distances [12], to also allow for the ball being at different relative positions with respect to the robot. The magmaOffenburg team divulged work on learning a kick while walking in raw actuator space [13]. The BahiaRT team discussed an extension to their existing defensive system [14], and the FUT-K team talked about their approach to preventing long passes by the opponent.

5.2 Keepaway Challenge

In the course of the keepaway challenge,[5] a group of three agents on one team attempts to maintain possession and keep the ball away from a single opponent agent for as long as possible. Additionally, the ball must be kept within a slowly shrinking square area on the field, which is eventually reduced in size to zero after five minutes. During a keepaway task attempt, if the opponent agent touches the ball, or the ball leaves the allowed keepaway area on the field, a keepaway attempt is considered over, and the time that has passed from the beginning of the attempt is recorded as the score for that attempt.

[3] Videos of the keepaway and Gazebo running challenges can be found at `http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/#2016challenges`

[4] Free challenge entry descriptions available at `http://chaosscripting.de/files/competitions/RoboCup/WorldCup/2016/3DSim/freeChallenge/`

[5] Details and framework for the keepaway challenge at `https://github.com/magmaOffenburg/magmaChallenge#keep-away-challenge`

Fig. 3: Keepaway challenge triangle (left image) and square (right image) formations. Red box is keepaway area. Yellow lines show where ball will next be passed. Blue line shows agent movement using second level kick anticipation (moving to new position in formation once pass has occurred).

UT Austin Villa created two formations for use in the keepaway challenge shown in Figure 3. In the first formation, agents align themselves in a triangle shape. The agent on the ball waits until the opponent gets near, and then that agent passes the ball to the teammate for which the angle between the opponent and teammate as measured from the ball is the largest (the ball is passed to the teammate with the least chance of the opponent intercepting the pass). In the second formation, agents assume a square shape with teammates in position to receive a pass at 90° angles to the ball. The same passing strategy as used with the triangle formation is used with the square formation, but with the addition that kick anticipation [4] is extended to the second level—a player anticipates and runs to the location of a second pass from the location that the next (first) pass is broadcast as going to by the player kicking the ball—so that teammates will be in their designated formation positions after a pass has occurred. Which formation to use is chosen randomly at the beginning of a keepaway attempt.

Table 8: Time in seconds for each of the teams competing in the keepaway challenge.

Team	Attempt 1	Attempt 2	Attempt 3	Average of Best Two Attempts
UTAustinVilla	**235.39**	**152.56**	**223.55**	**229.47**
FUT-K	108.95	38.04	140.09	124.52
FCPortugal	35.40	33.04	80.78	58.09
magmaOffenburg	35.76	20.86	24.86	30.31
BahiaRT	17.50	17.26	13.52	17.38

Table 8 shows the results of the keepaway challenge where teams were ranked by the average time of a team's best two out of three attempts. UT Austin Villa won the challenge with an average time of over three minutes. Each of UT Austin Villa's keepaway attempt times were greater than all other teams' attempts.

5.3 Gazebo Running Challenge

Ongoing work within the RoboCup community is the development of a plugin[6] for the Gazebo [15] robotics simulator to support the RoboCup 3D Simulation League. As such, a challenge[7] was held where robots attempt to walk forward

[6] `https://bitbucket.org/osrf/robocup3ds`

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

as fast as possible for 20 seconds in the Gazebo simulator without falling. In preparation for the challenge UT Austin Villa optimized fast walking parameters for the team's omnidirectional walk engine [16] within the Gazebo simulator using the CMA-ES algorithm [17]. Walk engine parameters were optimized for 300 generations of CMA-ES with a population size of 150.

Table 9: Speed in meters per second for each of the teams competing in the Gazebo running challenge.

Team	Run 1	Run 2	Run 3	Run 4	Run 5	Average
UTAustinVilla	**1.01**	**1.20**	**1.03**	**1.20**	**1.16**	**1.12**
FCPortugal	0.62	0.59	0.67	0.63	0.61	0.63
magmaOffenberg	0.44	0.26	0.43	0.44	0.40	0.39
FUT-K	0.00	0.00	0.00	0.00	0.00	0.00

Results of the Gazebo running challenge are shown in Table 9. Each participating team performed five running attempts and were scored by the average forward walking speed across all attempts. UT Austin Villa won the challenge with all five of the team's runs having a speed of over 1 m/s. Each of UT Austin Villa's running attempt speeds were greater than all other teams' attempts.

6 Conclusion

UT Austin Villa won the 2016 RoboCup 3D Simulation League main competition as well as all technical league challenges.[8] Data taken using released binaries from the competition show that UT Austin Villa winning the competition was statistically significant. The 2016 UT Austin Villa team also improved dramatically from 2015 as it was able to beat the team's 2015 champion binary by an average of 0.561 (+/-0.029) 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 [18].[9] 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 the new KgpKubs team at the 2016 competition). Additionally the code release—which was awarded second place for the HARTING Open Source Prize at RoboCup 2016—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 in 2016 is supported in part by grants from NSF (CNS-1330072, CNS-1305287), ONR (21C184-01), and AFOSR (FA9550-14-1-0087). Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by UT Austin in accordance with its policy on objectivity in research.

[8] More information about the UT Austin Villa team, as well as video highlights from the competition, can be found at the team's website: `http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/#2016`

[9] 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., 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)
6. Boedecker, J., Asada, M.: Simspark–concepts and application in the robocup 3d soccer simulation league. In: SIMPAR-2008 Workshop on the Universe of RoboCup Simulators. (2008) 174–181
7. 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)
8. 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)
9. Akiyama, H., Noda, I.: Multi-agent positioning mechanism in the dynamic environment. In: RoboCup 2007: Robot Soccer World Cup XI. LNAI. Springer (2008)
10. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2** (1955) 83–97
11. MacAlpine, P., Stone, P.: Prioritized role assignment for marking. In: RoboCup 2016: Robot Soccer World Cup XX. LNAI. Springer (2016)
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. Dorer, K., Kurz, V.: Learning a kick while walking in raw actuator space. (2016)
14. da Silva, C., Soares, A., Argollo, E., Simões, M.A.C., Frias, D., de Souza, J.R.: Módulo cooperativo de defesa para um time de futebol de agentes humanoides autônomos. In: XIII Workshop de Trabalhos de Iniciação Científica e Graduação da Escola Regional de Computação Bahia - Alagoas - Sergipe. (2015)
15. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: Intelligent Robots and Systems (IROS). (2004)
16. 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)
17. Hansen, N.: The CMA Evolution Strategy: A Tutorial. (2009) `http://www.lri.fr/~hansen/cmatutorial.pdf`.
18. MacAlpine, P., Stone, P.: UT Austin Villa robocup 3D simulation base code release. In: RoboCup 2016: Robot Soccer World Cup XX. LNAI. Springer (2016)