UT Austin Villa 3D Simulation Soccer Team 2020

Bo Liu[†], Yifeng Zhu[†] and Patrick MacAlpine[‡] and Peter Stone[†]

[†]The University of Texas at Austin, USA
{bliu,yifengz,pstone}@cs.utexas.edu
[‡] Microsoft Research, USA
patmac@microsoft.com

Abstract. This paper describes the research focus and ideas incorporated in the UT Austin Villa 3D simulation soccer team entering the RoboCup competitions in 2020.

1 Introduction

In this paper, we describe the agent our team UT Austin Villa is currently developing for participation at the 2020 RoboCup 3D Simulation Soccer competition. The main challenge presented by the 3D simulation league is the low-level control of a humanoid robot with more than 20 degrees of freedom. The simulated environment is a 3-dimensional world that models realistic physical forces such as friction and gravity, in which teams of humanoid robots compete with each other. Thus, the 3D simulation competition paves the way for progress towards the guiding goal espoused by the RoboCup community, of pitting a team of 11 humanoid robots against a team of 11 human soccer players. Programming humanoid agents in simulation, rather than in reality, brings with it several advantages, such as making simplifying assumptions about the world, low installation and operating costs, and the ability to automate experimental procedures. All these factors contribute to the uniqueness of the 3D simulation league.

The approach adopted by our team UT Austin Villa to decompose agent behavior is bottom-up in nature, comprising lower layers of joint control and inverse kinematics, on top of which skills such as walking, kicking and turning are developed. These in turn are tied together at the high level of strategic behavior. Details of this architecture are presented in this paper, which is organized as follows. Section 2 provides a brief overview of the 3D humanoid simulator. In Section 3, we describe the design of the UT Austin Villa agent, and elaborate on its skills in Section 4. In Section 5, we draw conclusions and present directions for future work.

2 Brief Overview of 3D Simulation Soccer

2007 was the first year of the 3D simulation competition in which the simulated robot was a humanoid. The humanoid used in the 2007 RoboCup competitions in Atlanta, U.S.A., was the Soccerbot, which was derived from the Fujitsu HOAP-2 robot model.[1] Owing to problems with the stability of the simulation, the Soccerbot was replaced by the Aldebaran Nao robot[2] at the 2008 RoboCup competitions in Suzhou, China. The robot has 22 degrees of freedom: six in each leg, four in each arm, and two in the neck and head. Figure 1 shows a visualization of the Nao robot and the soccer field during a game. The agent described in the following sections of this paper is developed for the Nao robot.

Each component of the robot's body is modeled as a rigid body with a mass that is connected to other components through joints. Torques may be applied to the motors controlling the joints. A physics simulator (Open Dynamics Engine[3]) computes the transition dynamics of the system taking into consideration the applied torques, forces of friction and gravity, collisions, etc. Sensation is available to the robot through a camera mounted in its head, which provides information about the positions of objects on the field every third cycle. This information has a small amount of noise added to it and is also restricted to a 120° view cone. The visual information, however, does not provide a complete description of state, as details such as joint orientations of other players and the spin on the ball are not conveyed. Apart from the visual sensor, the agent also gets information from touch sensors at the feet and accelerometer and gyro rate sensors. The simulation progresses in discrete time intervals with period 0.02 seconds. At each simulation step, the agent receives sensory information and is expected to return a 22-dimensional vector specifying torque values for the joint motors.

Since 2007 was the year the humanoid was introduced to the 3D simulation league, the major thrust in agent development thus far has been on developing robotic skills such as walking, turning, and kicking. This has itself been a challenging task, and is work still in progress. High-level behaviors such as passing

[1] http://jp.fujitsu.com/group/automation/en/services/humanoid-robot/hoap2/
[2] http://www.aldebaran-robotics.com/eng/
[3] http://www.ode.org/

Figure 1. On the left is a screenshot of the Nao agent, and on the right a view of the soccer field during a 11 versus 11 game.

and maintaining formations are beginning to emerge, and are now beginning to play more of a role in determining the quality of play in addition to the proficiency of the agent's skills.

For a more in-depth history of the 3D simulation league see [1].

3 Agent Architecture

At intervals of 0.02 seconds, the agent receives sensory information from the environment. Every third cycle a visual sensor provides distances and angles to different objects on the field from the agent's camera, which is located in its head. It is relatively straightforward to build a world model by converting this information about the objects into Cartesian coordinates. This of course requires the robot to be able to localize itself for which we use a particle filter incorporating both landmark and field line observations [7, 14]. In addition to the vision perceptor, our agent also uses its accelerometer readings to determine if it has fallen and employs its auditory channels for communication.

Once a world model is built, the agent's control module is invoked. Figure 2 provides a schematic view of the control architecture of our humanoid soccer agent.

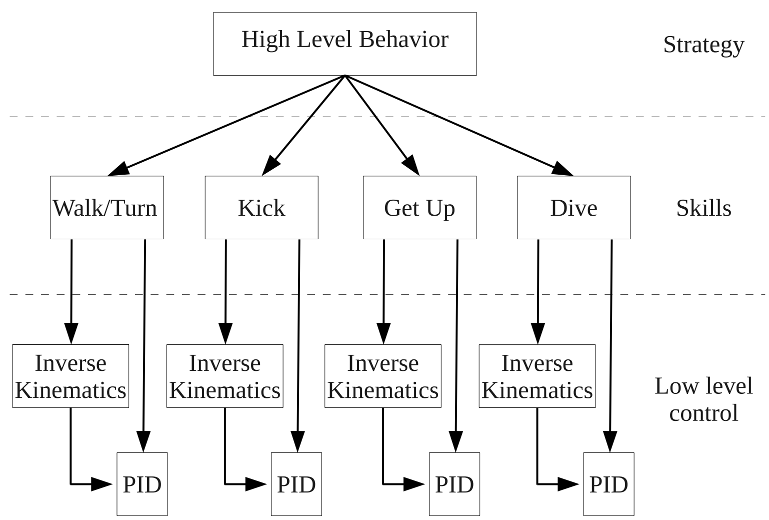

Figure 2. Schematic view of UT Austin Villa agent control architecture.

At the lowest level, the humanoid is controlled by specifying torques to each of its joints. We implement this through PID controllers for each joint, which take as input the desired angle of the joint and compute the appropriate torque. Further, we use routines describing inverse kinematics for the arms and legs. Given a target position and pose for the foot or the hand, our inverse kinematics routine uses trigonometry to calculate the angles for the different joints along the arm or the leg to achieve the specified target, if at all possible. The PID control and inverse kinematics routines are used as primitives to describe the agent's skills, which are discussed in greater detail in Section 4.

Developing high-level strategy to coordinate the skills of the individual agents is work in progress. Given the capabilities of the current set of skills, we employ a high-level behavior for 11 versus 11 games as follows. We instruct the player closest to the ball to go to it while other field player agents assume set formational positions on the field computed using Delaunay triangulation [2] based on offset positions from the ball. Our predefined formations, as well as role assignment to determine which agent should go to which position in the formation, are described in [11]. Our role assignment functions minimize the makespan (time for all agents to reach assigned target positions on the field) and are computed quickly and efficiently in polynomial time using SCRAM role assignment algorithms [17]. We have also developed and employed a marking system that incorporates an extension to SCRAM role assignment for prioritized role assignment [19]. When deciding where to kick the ball for a pass, agents use a learned neural network scoring function to choose a location to kick the ball [24], and then broadcast this location so that teammates may alter their assigned formation positions and move toward the anticipated destination of the kick [14]. Unlike field players that are interchangeable and can be assigned to any field player role position on the field, the goalie is instructed to stand a little in front of our goal and, using a Kalman filter to track the ball, attempts to dive and stop the ball if it comes near [26].

4 Player Skills

Our plan for developing the humanoid agent consists of first developing a reliable set of skills, which can then be tied together by a module for high-level behavior. Our foremost concern is locomotion. Bipedal locomotion is a well-studied problem (for example, see Pratt [28] and Ramamoorthy and Kuipers [29]). However, it is hardly ever the case that approaches that work on one robot generalize in an easy and natural manner to others. Programming a bipedal walk for a robot demands careful consideration of the various constraints underlying it.

We experimented with several approaches to program a walk for the humanoid robot, including monitoring its center of mass, specifying trajectories in space for its feet, and using machine learning techniques to optimize a series of fixed key frame poses for the agent to cycle through in order to walk and turn in different directions [34]. After deciding that an omnidirectional walk gave us the best chance for quickly moving and turning, we chose to use a double linear inverted pendulum model based omnidirectional walk engine that was designed by our standard platform league team for use on the physical Nao robots. This walk engine, and associated optimization of parameters for the walk, are described in [12].

When invoking the kicking skill, the agent chooses from several different kicks as described in [27] and [13]. During kicking inverse kinematics is used to control the kicking foot such that it follows an appropriate trajectory through the ball. This trajectory is defined by set waypoints, ascertained through machine learning, relative to the ball along a cubic Hermite spline. For the 2014 competition we were able to use learning by observation to develop and integrate new longer kicks that can travel 20 meters as described in [4] and [14]. In 2015 we extended agents' repertoires of kicks to include variable distance kicks for accuracy, and in doing so were able to execute set plays [16]. Kick selection was later tuned for height in 2016 [20]. Fast walk kicks, which take less than 0.25 seconds to execute, were added for the 2017 competition [23].

Two other useful skills for the robot are falling (for instance, by the goalie to block a ball) and rising from a fallen position. We programmed the fall by having the robot bend its knee, by virtue of which it would lose balance and fall. Our routine for rising is divided into stages. If fallen face down, the robot bends at the hips and stretches out its arms until it transfers weight to its feet, at which point it can stand up by straightening the hip angle. If fallen face up, the robot uses its arms to push its torso up, and then rocks its weight back to its feet before straightening its legs to stand. We optimized the rising movements of the robot for speed and stability as described in [13]. We also optimized goalie dives to stop shots on goal [24].

Skills for getting up, walking, and kicking are optimized using an overlapping layered learning approach [22]. Layered learning is a hierarchical machine learning paradigm that enables learning of complex behaviors by incrementally learning a series of sub-behaviors [30]. Overlapping layered learning is an extension to the paradigm that allows learning certain behaviors independently, and then later stitching them together by learning at the "seams" where their influences overlap. By using an overlapping layered learning approach we ensure that newly learned skills by an agent will work together with previously learned skills (the agent is able to stably transition between skills).

In 2019, skills were optimized to significantly reduce the probability of self-collisions, and strategy changes were introduced to support a new pass mode added to the competition—these changes were keys to winning the 2019 RoboCup 3D simulation competition [25].

Videos of some our agent's skills are available at our team's homepage.[4]

5 Conclusions and Future Work

This paper has presented a high-level view of the architecture and design of the UT Austin Villa agent. A fairly comprehensive and in-depth description of our 2011 agent, which is the base for this year's agent, is given in a technical report [26].

The simulation of a humanoid robot opens up interesting problems for control, optimization, machine learning, and AI. While the main emphasis thus far has been on getting a workable set of skills for the humanoid, for which considerable headway has been made, there is now a shift in the league to working

[4] `http://www.cs.utexas.edu/~AustinVilla/sim/3Dsimulation/`

on higher level behaviors as well. To expedite the progress being made in the 3D simulation domain, and promote participation and research efforts within the league, UT Austin Villa has released a base set of the team's code[5] to serve as a starting point for members of the research community [21]. A humanoid soccer league with scope for research at multiple layers in the architecture offers a unique challenge to the RoboCup community and augurs well for the future. There are numerous vistas that research in the 3D humanoid simulation league is yet to explore; these provide the inspiration and driving force behind UT Austin Villa's desire to participate in this league.

UT Austin Villa has been involved in the past in several research efforts involving RoboCup domains. Kohl and Stone [10] used policy gradient techniques to optimize the gait of an Aibo robot (4-legged league) for speed. Stone *et al.* [31] introduced Keepaway, a subtask in 2D simulation soccer [3, 9], as a test-bed for reinforcement learning, which has subsequently been researched extensively by others (for example, Taylor and Stone [32], Kalyanakrishnan *et al.* [8], and Taylor *et al.* [33]). Most recently the team has used the 3D simulation domain to explore learning walks for bipedal locomotion (MacAlpine *et al.* [12], Farchy *et al.* [5], and Hanna *et al.* [6]). Additionally, UT Austin Villa has used RoboCup as a testbed for ad hoc teamwork research by creating drop-in player challenges where robots programmed by different teams play soccer with each other without pre-coordination [15, 18]. We are keen to continue our research initiative in the 3D simulation league.

Our initial focus for the 2020 competition will be on further optimizing our set of skills, to realize faster walks, better formation strategy, etc. Banking on a reliable set of skills, we will seek to continue developing higher level behaviors such as passing, intercepting balls, and marking opponents. We also hope to incorporate deep learning into these efforts and make these relatively computation expensive deep learning methods run during the competition.

References

1. H. Akiyama, K. Dorer, and N. Lau. On the progress of soccer simulation leagues. In R. A. C. Bianchi, H. L. Akin, S. Ramamoorthy, and K. Sugiura, editors, *RoboCup-2014: Robot Soccer World Cup XVIII*, Lecture Notes in Artificial Intelligence, pages 599–610. Springer, 2015.
2. H. Akiyama and I. Noda. Multi-agent positioning mechanism in the dynamic environment. In *RoboCup 2007: Robot Soccer World Cup XI*. Springer, 2008.
3. M. Chen, E. Foroughi, F. Heintz, Z. Huang, S. Kapetanakis, K. Kostiadis, J. Kummeneje, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang, and X. Yin. Users manual: RoboCup soccer server — for soccer server version 7.07 and later. *The RoboCup Federation*, August 2002.
4. M. Depinet, P. MacAlpine, and P. Stone. Keyframe sampling, optimization, and behavior integration: Towards long-distance kicking in the robocup 3d simulation league. In R. A. C. Bianchi, H. L. Akin, S. Ramamoorthy, and K. Sugiura, editors, *RoboCup-2014: Robot Soccer World Cup XVIII*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 2015.

[5] UT Austin Villa base code release: `https://github.com/LARG/utaustinvilla3d`

5. A. Farchy, S. Barrett, P. MacAlpine, and P. Stone. Humanoid robots learning to walk faster: From the real world to simulation and back. In *Proc. of 12th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2013.
6. J. Hanna and P. Stone. Grounded action transformation for robot learning in simulation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, February 2017.
7. T. Hester and P. Stone. Negative information and line observations for monte carlo localization. In *IEEE Int. Conf. on Robotics and Automation*, May 2008.
8. S. Kalyanakrishnan, Y. Liu, and P. Stone. Half field offense in RoboCup soccer: A multiagent reinforcement learning case study. *Proceedings of the RoboCup International Symposium 2006*, June 2006.
9. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. RoboCup: A challenge problem for AI. *AI Magazine*, 18(1):73–85, 1997.
10. N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2004.
11. P. MacAlpine, F. Barrera, and P. Stone. Positioning to win: A dynamic role assignment and formation positioning system. In X. Chen, P. Stone, L. E. Sucar, and T. V. der Zant, editors, *RoboCup-2012: Robot Soccer World Cup XVI*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 2013.
12. P. MacAlpine, S. Barrett, D. Urieli, V. Vu, and P. Stone. 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)*, July 2012.
13. P. MacAlpine, N. Collins, A. Lopez-Mobilia, and P. Stone. UT Austin Villa: RoboCup 2012 3D simulation league champion. In X. Chen, P. Stone, L. E. Sucar, and T. V. der Zant, editors, *RoboCup-2012: Robot Soccer World Cup XVI*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 2013.
14. P. MacAlpine, M. Depinet, J. Liang, and P. Stone. UT Austin Villa: RoboCup 2014 3D simulation league competition and technical challenge champions. In R. A. C. Bianchi, H. L. Akin, S. Ramamoorthy, and K. Sugiura, editors, *RoboCup-2014: Robot Soccer World Cup XVIII*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 2015.
15. P. MacAlpine, K. Genter, S. Barrett, and P. Stone. The RoboCup 2013 drop-in player challenges: Experiments in ad hoc teamwork. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014.
16. P. MacAlpine, J. Hanna, J. Liang, and P. Stone. UT Austin Villa: RoboCup 2015 3D simulation league competition and technical challenges champions. In L. Almeida, J. Ji, G. Steinbauer, and S. Luke, editors, *RoboCup-2015: Robot Soccer World Cup XIX*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 2015.
17. P. MacAlpine, E. Price, and P. Stone. SCRAM: Scalable collision-avoiding role assignment with minimal-makespan for formational positioning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, January 2015.
18. P. MacAlpine and P. Stone. Evaluating ad hoc teamwork performance in drop-in player challenges. In G. Sukthankar and J. A. Rodriguez-Aguilar, editors, *Autonomous Agents and Multiagent Systems, AAMAS 2017 Workshops, Best Papers*, volume 10642 of *Lecture Notes in Artificial Intelligence*, pages 168–186. Springer International Publishing, 2017.

19. P. MacAlpine and P. Stone. Prioritized role assignment for marking. In S. Behnke, D. D. Lee, S. Sariel, and R. Sheh, editors, *RoboCup 2016: Robot Soccer World Cup XX*, Lecture Notes in Artificial Intelligence. Springer, 2017.
20. P. MacAlpine and P. Stone. UT Austin Villa: RoboCup 2016 3D simulation league competition and technical challenges champions. In S. Behnke, D. D. Lee, S. Sariel, and R. Sheh, editors, *RoboCup 2016: Robot Soccer World Cup XX*, Lecture Notes in Artificial Intelligence. Springer, 2017.
21. P. MacAlpine and P. Stone. UT Austin Villa RoboCup 3D simulation base code release. In S. Behnke, D. D. Lee, S. Sariel, and R. Sheh, editors, *RoboCup 2016: Robot Soccer World Cup XX*, Lecture Notes in Artificial Intelligence. Springer, 2017.
22. P. MacAlpine and P. Stone. Overlapping layered learning. *Artificial Intelligence*, 254:21–43, January 2018.
23. P. MacAlpine and P. Stone. UT Austin Villa: RoboCup 2017 3D simulation league competition and technical challenges champions. In C. Sammut, O. Obst, F. Tonidandel, and H. Akyama, editors, *RoboCup 2017: Robot Soccer World Cup XXI*, Lecture Notes in Artificial Intelligence. Springer, 2018.
24. P. MacAlpine, F. Torabi, B. Pavse, J. Sigmon, and P. Stone. UT Austin Villa: RoboCup 2018 3D simulation league champions. In D. Holz, K. Genter, M. Saad, and O. von Stryk, editors, *RoboCup 2018: Robot Soccer World Cup XXII*, Lecture Notes in Artificial Intelligence. Springer, 2019.
25. P. MacAlpine, F. Torabi, B. Pavse, and P. Stone. UT Austin Villa: RoboCup 2019 3D simulation league competition and technical challenge champions. In S. Chalup, T. Niemueller, J. Suthakorn, and M.-A. Williams, editors, *RoboCup 2019: Robot Soccer World Cup XXIII*, pages 540–552. Springer, 2019.
26. P. MacAlpine, D. Urieli, S. Barrett, S. Kalyanakrishnan, F. Barrera, A. Lopez-Mobilia, N. Ştiurcă, V. Vu, and P. Stone. UT Austin Villa 2011 3D Simulation Team report. Technical Report AI11-10, The University of Texas at Austin, Department of Computer Science, AI Laboratory, December 2011.
27. P. MacAlpine, D. Urieli, S. Barrett, S. Kalyanakrishnan, F. Barrera, A. Lopez-Mobilia, N. Ştiurcă, V. Vu, and P. Stone. 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)*, June 2012.
28. J. Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Computer Science Department, Massachusetts Institute of Tehcnology, Cambridge, Massachusetts, 2000.
29. S. Ramamoorthy and B. Kuipers. Qualitative hybrid control of dynamic bipedal walking. *Robotics: Science and Systems*, II, 2007.
30. P. Stone. Layered learning in multiagent systems: A winning approach to robotic soccer. 2000.
31. P. Stone, R. S. Sutton, and G. Kuhlmann. Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 13(3):165–188, 2005.
32. M. E. Taylor and P. Stone. Behavior transfer for value-function-based reinforcement learning. pages 53–59, July 2005.
33. M. E. Taylor, S. Whiteson, and P. Stone. Comparing evolutionary and temporal difference methods for reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1321–28, July 2006.
34. D. Urieli, P. MacAlpine, S. Kalyanakrishnan, Y. Bentor, and P. Stone. On optimizing interdependent skills: A case study in simulated 3D humanoid robot soccer. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, May 2011.