Karachi Koalas 3D Simulation Soccer Team
Team Description Paper for World RoboCup 2014

Sajjad Haider[1], Mary-Anne Williams[2], Saleha Raza[1], Xun Wang[2], Syed Ali Raza[1] ,
Muhammad Anshar[2], Nima Ramezani[2], Usman Arif[1], Shamaila Mahmood[1], Maya Knox[2],
Michelle Yousuf[2], Tahir Pasha[2]

[1]Artificial Intelligence Lab, Institute of Business Administration, Karachi, Pakistan
[2]Innovation and Enterprise Research Lab, University of Technology, Sydney, Australia
http://karachikoalas.iba.edu.pk
sajjad.haider@khi.iba.edu.pk, mary-anne.williams@uts.edu.au,

Abstract. This paper describes the research and development efforts made by Karachi Koalas as it aims to participate in World RoboCup 2014 competition. We have made several changes to our robot team that participated in 2013 World RoboCup Soccer. A major addition to team is the inverse kinematics based Omni-directional walk. The walk has significantly improved our players' ability to move in the ground. Work is currently in progress to make changes in the code architecture and locomotion capabilities to accommodate heterogeneous robots.

1 Introduction

Karachi Koalas team was formed in the mid of 2010 as a result of a strong and further evolving scientific partnership between University of Technology, Sydney (UTS) and Institute of Business Administration, Karachi (IBA). The history of Karachi Koalas shows a constant improvement in team performance. For the last three years, we have been participating in RoboCup competitions. In 2011 and 2012, we reached the top 16 and top 10 groups, respectively. Last year in 2013, we were successful to secure 5th position in the competitions. In addition, it won two awards at IranOpen RoboCup 201: "Best Presentation" and "Most Improved Team". UTS has a strong commitment to the RoboCup competition and has been a frequent participant in the Standard Platform League starting from 2003. It won the Australian RoboCup Championship competition in 2004 and was the top International Team in 2004 at Robot Soccer World Cup where it came first in the Soccer Challenges and second in the Soccer Games. In 2008, it formed a joint Standard Platform League team, named WrightEagleUnleashed [15], with University of Science and Technology China, which was the Runner-Up of 2008. Several papers have been published by the team members on RoboCup related research topics that demonstrate its commitment and contribution to the advancement of RoboCup [1-12].

The rest of the paper describes the development environment and code architecture of Karachi Koalas. In addition, it also provides an overview of the advancements made in locomotion, localization and team/agent behavior and a list of high priority tasks we aim to finish before the competition.

2 Development Environment

We are the only team in 3D simulation league which has done all the development in C#/Mono. KarachiKoalas has used TinMan library [16] that handles low-level communication with the server. TinMan's execution on Linux has been made possible through Mono[1]. This flexibility provided us an ideal platform to build our code simultaneously in Windows and Linux environments. We have also made extensive use of RoboViz [17]. The tool is great for the dynamic placement of ball and agents as well as getting insight of agents' internal states and beliefs via text annotations and different types of graphical displays.

3 Software Architecture

We have developed a modular architecture that is built on top of the TinMan library. TinMan supports low-level interfacing with the RoboCup server (rcssserver3d) by providing higher level abstraction of preceptors and actuators for communication with the server. Fig. 1 provides a high level view of the overall software architecture. The actuators and preceptor layer exposed by TinMan are used by our *AgentModel* and *TeamModel*. *AgentModel* is responsible to handle the functioning of an individual agent. This includes maintaining current state of the agent in *AgentState*, localizing it in the field using *localization engine* and enabling it to exhibit different behaviors via *Behavior* layer. Behavior layer is an intermediate layer that in turn calls the locomotion engine to exhibit low-level locomotive skills. The *Behavior* layer allows us to build our strategy in terms of high-level behaviors and facilitates us in building more sophisticated strategies. The low-level intricacies of every behavior are hidden in its internal implementation. Locomotion, localization and collision avoidance are three key components of *AgentModel* and have been briefly explained in the following section. Overall coordination among agents is performed by *AgentCollaboration* module that gathers an agent's

[1] Mono is an open source implementation of .NET framework which enables .NET applications to be developed and executed on Linux.

state from *AgentState* and game/world state from *WorldState* and applies different heuristics to devise a suitable strategy. *TeamStrategy* module deals with the execution of a certain strategy by adopting a suitable formation via *TeamFormation* module and dynamically assigning different roles to each player via *DynamicRoleAssignment* module. Agents are then responsible to exhibit these roles using *RoleExection*. The Simspark server supports direct communication among agents through its messaging interface. This interface has also been exploited by the *AgentCollaboration* module that in turn uses *SimulationContext* of TinMan to receive and broadcast messages.

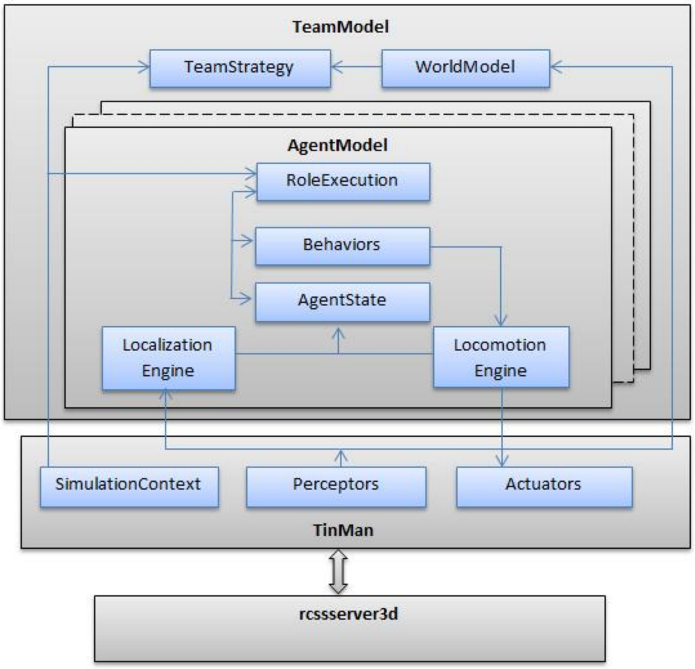

Fig. 1. Software Architecture

4 Team/Agent Skills

This section provides a brief overview of the ongoing research and development to improve Karachi Koalas' skill sets. Most of the skills have been successfully integrated into the code while the remaining few will hopefully be added before the start of the competition.

4.1 Locomotion

A major improvement in our team has been the development of inverse kinematics based omnidirectional walk. Previously, our walk was based on Partial Fourier Series (PSF) which had its own limitations specially in terms of achieving omni-directionality. For our new walk, we have designed an inverse kinematics based module for the leg joints of robot which is based on the solution provided by B-Human team [13]. The Nao implementation in the Simulation 3D environment has six joints for each leg of the robot. The hip YawPitch joint of both legs, however, are interlocked. Thus, they cannot be set to different values and setting any one of them sets the other automatically. The inverse kinematic module, therefore, outputs the joint angle values of eleven joints while taking position and orientation matrices of both legs as input. Our walk engine is basically a variation of [13] as proposed in [14].

A higher level behavior module tells the walk engine the target point. As the new omnidirectional walk has the ability to move in any directional without turning and (virtually) with the same velocity, we have designed different behaviors to utilize this feature. We have set a threshold value that decides whether to turn towards a target point or not before approaching it. It is important to note that the target point (mostly the ball position) changes continuously during the game and thus extra time is required to keep facing the target position. If the distance to the target position is less than the defined threshold then we reach the target without turning. Thus, unlike [14], we are utilizing the ability of omni-directional walk to reach any position without changing the orientation.

4.2 Localization

Our localization engine uses both marker and lines for self-localization. The engine only requires one landmark and a connected line. To identify the right line among the set of lines, Cartesian system of the world model is converted into the Cartesian system of the robot. Subsequently, the length of line is determined to find its end points in world model and then the Cartesian system is switched back to that of the world model. By using the endpoints of that line and the point of the marker, the position of the robot is calculated.

To further improve the localizer accuracy and to remove noise, we have also added a Kalman filer layer that smoothes the localizer value and ballPolar estimate provided by the server.

Message passing mechanism has also been used to gain understanding of the environment. A player, who sees the ball and is also confident of its own position, announces the ball position through the message passing mechanism. This helps other team members, who are not seeing the ball, in their decision-making. For instance, a defender who has gone too far and is not seeing the ball directly can get the message from the goalie that the ball is behind and it can use back walk to get closer to the ball.

4.4 Dynamic Role Assignment

The role assignment layer does dynamic role assignment which means that any player (except player # 1) can take any role during a match depending upon the game situation. During a match, players maintain a particular formation and each point in the formation defines a particular role. These roles are dynamically assigned to the players in accordance with their current position and the positions of ball and other teammates. The dynamic role assignment module applies dynamic programming to determine the best formation that require minimum distances to be covered by the players to reach their desired target. This minimization ensures that roles are optimally assigned to the players and the formation is efficiently maintained.

References

1. Ali Raza, Usman Shareef and Sajjad Haider, On Learning Coordination Among Soccer Agents, In Proceedings of IEEE International Conference on Robotics and Biomimetics , Guanzhou, China (2012)
2. Anshar, M and Williams, M-A: Extended Evolutionary Fast Learn-to-Walk Approach for Four-Legged Robots. Journal of Bionic Engineering, 255-263 (2007).
3. Anshar, M and Williams, M-A: Evolutionary Robot Gaits. In: International Conference on Intelligent Unmanned Systems (2007)
4. Asma Larik and Sajjad Haider, Rule-Based Behavior Prediction of Opponent Agents using Robocup 3D Soccer Simulation League Logfiles, IFIP Advances in Information and Communication Technologies , 381, pp. 285-295 (2012)
5. Chris Stanton, Edward Ratanasena, Sajjad Haider and Mary-Anne Williams, Perceiving Bumps and Touches from Proprioceptive Expectations, In Proceedings of 2011 RoboCup Symposium, Istanbul, Turkey (2011)
6. Karol, A. and Williams, M-A.: Robot soccer Distributed Sensor Fusion for Object Tracking. In: RoboCup 2005, Robot Soccer World Cup VIII Series: Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, 504-511 (2006)
7. Karol, A., Nebel, B., and Williams, M-A.: Case-Based Game Play in the RoboCup Four-Legged League: Part I The Theoretical Model. In: RoboCup 2003: Robot Soccer World Cup VII Series: Lecture Notes in Computer Science,: Lecture Notes in Artificial Intelligence, Vol. 3020 Polani, D.; Browning, B.; Bonarini, A.; Yoshida, K. (Eds.) 739 -748 (2004)
8. Gärdenfors, P. and Williams, M-A, Building Rich and Grounded Robot World Models from Sensors and Knowledge Resources: A Conceptual Spaces Approach. In:2nd International Symposium on Autonomous Mini-Robots for Research and Edutainment, 34-45 (2003)
9. Mendoza, R., Johnston, B., Yang, F., Huang, Z., Chen, X., and Williams, M-A.: OBOC: Ontology based object categorisation for robots. In: Fourth International Conference on Computational Intelligence, Robotics and Autonomous Systems (2007)
10. Sajjad Haider, Shaukat Abidi and Mary-Anne Williams, On Evolving a Dynamic Bipedal Walk using Partial Fourier Series, In Proceedings of IEEE International Conference on Robotics and Biomimetics , Guanzhou, China (2012)

11. Saleha Raza, Sajjad Haider and Mary-Anne Williams, Teaching Coordinated Strategies to Soccer Robot via Imitation, In Proceedings of IEEE International Conference on Robotics and Biomimetics , Guanzhou, China (2012)
12. Saleha Raza and Sajjad Haider, Building Soccer Skills via Imitation In Proceedings of 3rd International Conference on Advanced Topics in Artificial Inteligence , Singapore, 2012.
13. Graf, C., Hartl, A., Rofer, T., and Laue, T, A robustclosed-loop gait for the standard platform league humanoid. In Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2009 IEEE-RAS International Conference on Humanoid Robots, 30 – 37 (2009)
14. Patrick MacAlpine, Samuel Barrett, Daniel Urieli, Victor Vu and Peter 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 (2012)
15. http://www.wrighteagleunleashed.org
16. http://code.google.com/p/tin-man/
17. https://sites.google.com/site/umroboviz/