Karachi Koalas3D Simulation Soccer Team
Team Description Paper for World RoboCup2012

Sajjad Haider[1], Mary-Anne Williams[2], Saleha Raza[1], Benjamin Johnston[2], Shaukat Abidi[1], Usman Sharif[1] and Ali Raza[1]

[1]Artificial Intelligence Lab, Institute of Business Administration, Karachi, Pakistan
[2]Innovation and Enterprise Research Lab, University of Technology, Sydney, Australia
http://www.karachikoalas.org

sajjad.haider@khi.iba.edu.pk, mary-anne.williams@uts.edu.au,

Abstract. This paper describes the progress made by Karachi Koalas since its participation in the 2011 RoboCup Soccer. We have improved upon our partial Fourier series and evolutionary algorithms based bipedal gait by adding dynamic skill switching capabilities. Gyro based stability routines have also been added to the gait which makes locomotion more stable. The strategy code is based on dynamic role switch and fuzzy rules. Currently we are working on intelligent path planning and collision avoidance methods and are exploring the application of machine learning and computational intelligence techniques to develop strategy of our soccer team.

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). In 2011, we participated in IranOpen and World RoboCup (held in Istanbul, Turkey) and reached the top 16 groups. 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. Since 2008, it has formed a joint Standard Platform League team, named WrightEagleUnleashed [1], 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 [2-9].

The rest of the paper describes the development environment and code architecture of Karachi Koalas, advancements made in the overall architecture and in locomotion, localization and team behavior and the further high priority tasks we aim to finish before the competition.

2 Development Environment

We are the only team (to the best of our knowledge) in 3D simulation league which has done all the development in C#/Mono. KarachiKoalas has used TinMan library [10] 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 the recently released debugging tool RoboViz [11]. 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 (rcsserver3d) by providing higher level abstraction of preceptors and actuators for communication with the server. After RoboCup 2011, there have been some major enhancements in our architecture. This includes addition of '*Behavior*' layer. The induction of this new layer now allows us to build our strategy in terms of high-level behaviors instead of low-level skills, as was the case previously, and facilitates us in building more sophisticated strategies. The low-level intricacies of every behavior are hidden in its internal implementation. 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. Locomotion and localization are two key components of *AgentModel* and have been described in detail in the following sections. Overall coordination among agents is performed by *AgentCollaboration* module that gathers an agent's 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

1

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

adopting a suitable formation and dynamically assigning different roles to each player. 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.

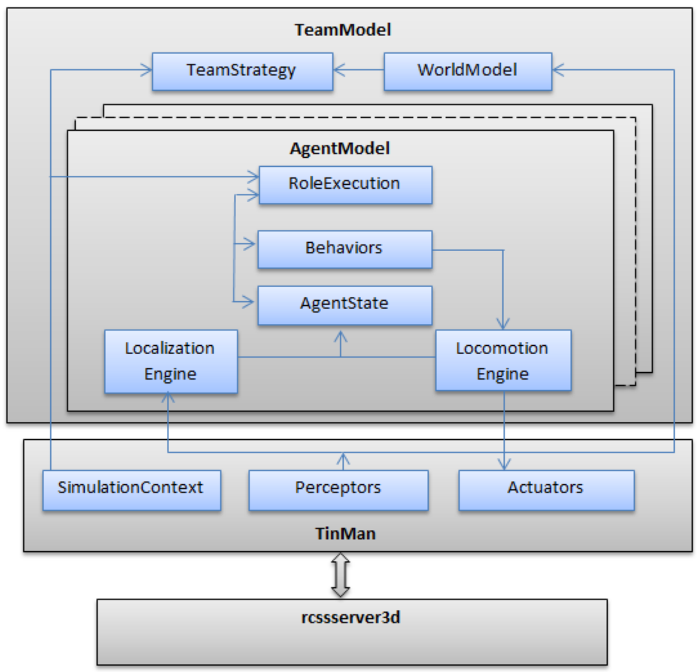

Fig. 1. Software Architecture

4 Locomotion

Our locomotion efforts were focused on the development of the following skill sets:

- Forward, backward, turn and sidewalks
- Getup from back and belly and diving behavior of the goal keeper
- Forward, side and angular kicks

The gait was developed by optimizing partial Fourier series via evolutionary algorithms. The general form of PFS used in our development is as follows:

$$f(t) = a_0 + \sum_{n=1}^{N} a_n \sin(\frac{2\pi nt}{L} + \phi_n)$$

where N is the number of frequencies, a_0 is the offset, a_n represents amplitudes, L is the period and φ_n represents phases.

This is the same mechanism which was adopted last year, but for this year, we have further evolved the walk and have made it more stable and faster. As a result, the speed of our locomotion is now comparable to the other top teams in simulation 3D league. In addition, a significant effort has been made in understanding the kinematics of the walk and incorporating dynamic adjustment in our locomotion engine. This has resulted in our agent having the capability to adjust their path dynamically without making a complete stop as was the case in our last year implementation.

Kicks have also been refined to optimize the time taken by an agent to kick the ball. In addition, based on the current game situation, agents intelligently decide to kick or dribble the ball.

5 Localization

Our localization module is based on particle filter and works in the following manner. If more than one marker is available through preceptor/vision sensors then it uses the triangulation method to compute the location and orientation of Nao. It also reinitializes/resamples particles within the neighborhood of Noa's position and orientation in this step. If only one marker is available, then first it predicts the position of each particle using the motion equation and then updates the position of Nao using the weighted average of particles' coordinates. Particles whose coordinates and orientation are more consistent with the available preceptor information are assigned more weight. In case no flag is available then motion equation is applied to each particle and the position of Nao is updated as a weighted average of particles' coordinates.

When a player loses the vision of the ball, it retains the previous ball position for some time. In addition, a player who sees the ball and is also confident of its own position, that is, is seeing at least two flags, announces the ball position through the message passing mechanism of SimSpark. This helps the 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. Ball positions obtained either through vision or message broadcast are smoothed by averaging out past few values so that the factor of noise in sensor readings can be minimized.

6 Strategy

For this year, our core focus has been on building effective strategy. As of now, our strategy module is capable of team formation, dynamic role switching, adversary/situation awareness and priority-based fuzzy rules. The latest enhancement in our software architecture now allows us to build game strategy using pre-defined behaviors of soccer agents. This behavior layer provides a higher level abstraction to build the strategy and release our strategy code from dealing with the low-level locomotive skills. These behaviors are called by the agents as per their designated roles. Four different types of roles are defined for the players: goalie, defenders, attacker and midfielders. Defenders are further divided into left and right defenders depending upon their default positions. The attackers/midfielders role is dynamic and any player among this set can become an attacker if it satisfies certain conditions. The attacker who is in the best position to approach the ball is selected as the main attacker, The remaining players then take the supporter role automatically and position themselves accordingly to provide support to the main attacker. The coordination among players is performed via broadcasted messages. This mechanism enables KarachiKoalas agents to communicate and coordinate with each other by broadcasting information about their current position and the role.

State machines in the form of priority fuzzy rules are defined for each role. These rules take inputs from a player's vision and the messages it receives from other team members. The rules are defined after careful analysis of several training matches. Our current focus is on the implementation of efficient path planning and collision avoidance techniques to enable the agents to intelligently move in the field and approach and dribble the ball. Similarly, the enhancement of strategy via application of machine learning techniques is another area of active research and development.

References

1. Anshar, M and Williams, M-A: Extended Evolutionary Fast Learn-to-Walk Approach for Four-Legged Robots. Journal of Bionic Engineering, 255-263 (2007).
2. Anshar, M and Williams, M-A: Evolutionary Robot Gaits. In: International Conference on Intelligent Unmanned Systems(2007)
3. 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)
4. Stanton, C. and Williams, M-A.: A Novel and Practical Approach towards Color Constancy Mobile Robots. In: RoboCup 2005: Robot Soccer World Cup VIII Series: Lecture Notes in Computer Science, Lecture Notes in AI, 444-451 (2006)
5. Stanton, C. and Williams, M-A.: Grounding Robot Sensory and Symbolic Information using the Semantic Web. 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.) 757-765 (2004)

6. 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)
7. 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)
8. 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)
9. http://www.wrighteagleunleashed.org
10. http://code.google.com/p/tin-man/
11. https://sites.google.com/site/umroboviz/