# Karachi Koalas 3D Simulation Soccer Team
# Team Description Paper for World RoboCup 2013

Sajjad Haider[1], Mary-Anne Williams[2], Saleha Raza[1], Benjamin Johnston[2], Shaukat Abidi[2], Ali Raza[1], Abdun Nafay[1], Nabeel Rajput[1], Safeer-e-Hussain[1], Sojharo[1]

[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 2013 RoboCup Soccer competitions. We have made several changes to our robot team that participated in 2012 World RoboCup Soccer. The localizer has been improved and now robots utilize both markers and lines to estimate their location in the ground. The localizer is further strengthened by adding a Kalman filter layer that sits on top of it and provides a smooth estimate of a robot's location and the ball position. An enhanced collision avoidance mechanism has also been added to the code. In addition, work is currently under progress to have a Voronoi diagram based team formation and dynamic programming based role assignment.

## 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 and 2012, we participated in World RoboCup competition and reached the top 16 and top 10 groups, respectively. 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 [13], 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 [14] 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 [15]. 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 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

---

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

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.

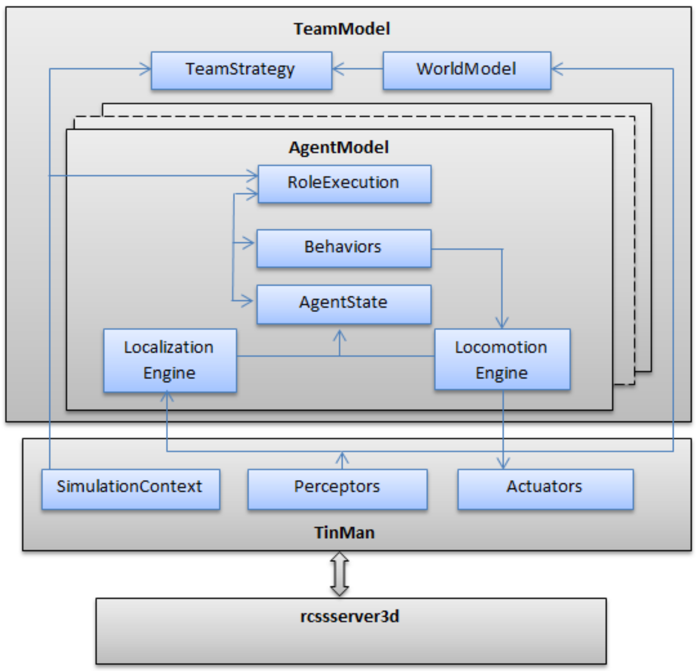


**Fig. 1.**Software Architecture

## 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*

We are still using partial Fourier series (PFS) based locomotion engine. The engine consists 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:

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

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

So far we have not made any significant changes to our walk engine but we aim to improve it before the competition. We, however, have managed to improve our kicking skills and now have couple of kicks having range of 10+ meters.

*4.2 Localization*

Significant improvements have been made to our localizer this year. In the past, the localizer was working on markers data using triangulation. The major improvement this year has been the use of lines and incorporation of Kalman filter that sits on top of the localization layer.

Localization with lines helps a robot in not moving its head in search of a pair of landmark. As a result, the robot can focus more on the ball and does not lose sight of it. The routine 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.

### 4.3 Collision Avoidance

In the previous years, we had used ad-hoc procedures to avoid collision among the team members. One of the major improvements this year has been a Potential Field based approach for obstacle avoidance. Attractive fields are assigned to a goal to be achieved by a robot and repulsive fields are assigned to obstacles in the path of the robot. The sum of all the attractive and the repulsive forces are calculated to provide an obstacle free path to the robot. The robot moves in the direction of the resultant force to avoid collision with the obstacles (team members).

### 4.4 Team Formation

We have used Voronoi diagram to make a flexible positioning system for our agents. Voronoi diagram takes a finite set of points and generates a region for every point which is known as Voronoi cell. Using Voronoi diagram we generate the Voronoi Cell for every agent except Goalie and Attacker. For this purpose, we have used a modified version of the Fortune's algorithm. It calculates a region for each agent and the center point of that region becomes the optimal position for the agent in the given situation. The cells are calculated for every cycle. If there is a change in the position of the agents then every agents move towards the center of its region. The implementation of this approach is still a work in progress but the initial testing of the algorithm has produced successful results.

### 4.5 Dynamic Role Assignment

In previous years, we had fixed formation and players were assigned specific roles. It is obvious that it was not an ideal solution and had many limitations such as no one taking a key position in case the corresponding player is thrown out by the server. This year we have opted for 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. http://www.wrighteagleunleashed.org
14. http://code.google.com/p/tin-man/
15. https://sites.google.com/site/umroboviz/