beeStanbul
RoboCup 3D Simulation League
Team Description Paper 2011

Shahriar Asta, Tuna Sonmez, Onuralp Ulusoy, Alper Alimoglu, Mustafa Ersen, Ozdemir Sozmen, and Sanem Sariel-Talay

Artificial Intelligence and Robotics Laboratory
Istanbul Technical University
Computer Engineering Department
Istanbul, TURKEY
http://air.cs.itu.edu.tr/beestanbul
sariel@itu.edu.tr

Abstract. The main objective of the beeStanbul project is to develop an efficient software system to correctly model the behaviors of simulated Nao robots in a competitive environment. Several AI algorithms are being used and developed for contributing to robotics research while also advancing the quality of competitions in 3D Simulation League. This team description paper presents important aspects of the overall system design and outlines the methods used in different modules.

1 Introduction

The beeStanbul project from the Artificial Intelligence and Robotics laboratory (AIR lab) at Istanbul Technical University (ITU) is the first initiative from ITU to participate in RoboCup competitions. Earlier projects in the AIR lab were mainly on cooperative multirobot systems. This challenging project was initiated in 2009 to apply the experience, gained from earlier research on multirobot systems [1–3], to competitive environments as well.

The beeStanbul team consists of undergraduate and graduate students from the Computer Engineering Department of ITU. The main goal of the team is to contribute for the main objective of the RoboCup project by an efficient design of a software system which will be successful during the main competition. The designed software system serves as a basis to apply several high-level intelligence, reasoning and learning methods.

beeStanbul team participated in the RoboCup 2010 competitions for the first time and had a change to score several goals during the competitions. The software architecture of the system has been revised and several promising motion types have already been developed since 2010. With the completion of the high-level modules, the team is expected to achieve the project's main goal.

The organization of the rest of the paper is as follows. Section 2 presents the software system architecture for simulated Nao robots in the SimSpark simulation environment. Motions and behaviors available for an agent are presented in

Sections 3 and 4 respectively. The developed team-strategy and planning methods are discussed in Section 5, followed by the conclusion in Section 6.

2 System Architecture

The overall software system consists of several modules that interact with each other (Fig. 1). The Server Layer perfoms a two-way communication with the SimSpark server, decodes incoming messages and encodes outgoing messages. In order to carry out these operations, the SimSpark utilities and rcssnet library, provided by SimSpark, are used.

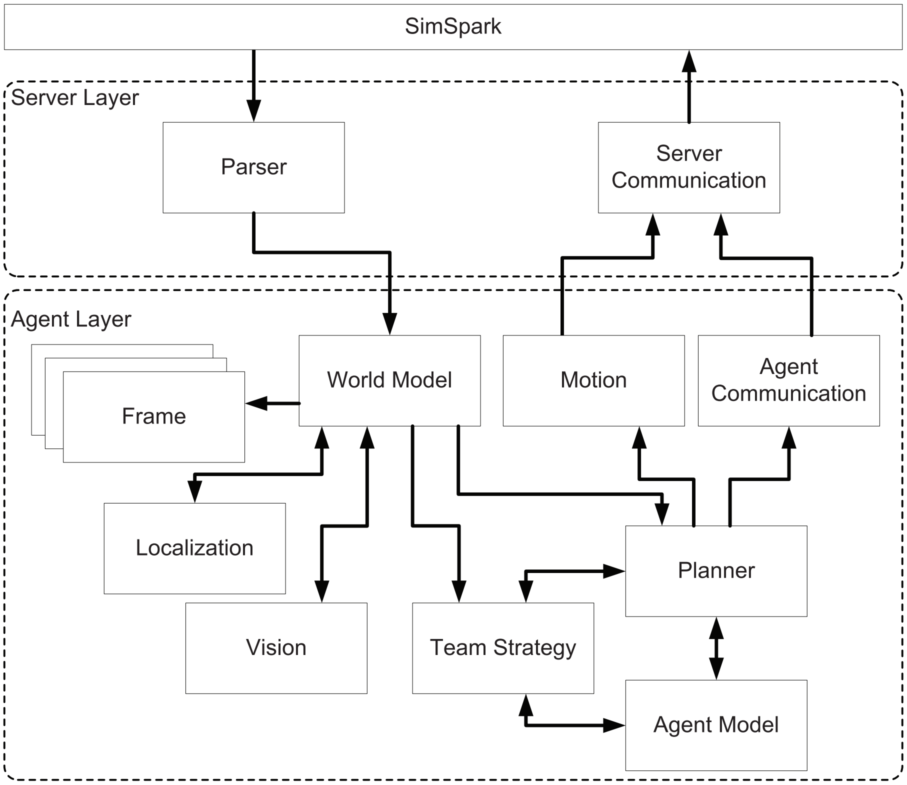

Fig. 1. Overall Software Architecture

The Agent Layer is responsible for performing the main functionalities of a robot. Each agent maintains its own world model for the environment and the agent model for its own state. The Localization module is responsible for determining the correct pose of the robot given an observation history. Localization is performed by using Monte Carlo Localization Algorithm [4]. A short term memory (in accordance with the observation history) of the robot is maintained by ten consecutive frames in the world model. The Vision module is responsible for determining the positions of the observed objects in the environment. Based on the observation history, a semantic analysis of the objects (including the opponents) and predictions are made. The robot decides on a behavior based on its agent and world models, the selected team strategy and the assigned role for itself. The motion command for the corresponding behavior is sent to the Server Layer to generate the desired effect. Simultaneously, either informative or query messages might be sent to teammates based on the selected team role. The planner for the goalie is different than that of the field players.

3 Motions

Robots use seven main body motion types for scoring or keeping a goal. These include straight walks (e.g., forward and side walk), diagonal walk, inward and outward turn, rotate, stand-up and kick. The first five motion types basically use the same approach with an oscillatory behavior while the last two have different phases during their executions. There are three head motions which can be executed simultaneously with the body motions. These include track object, search and reset.

3.1 Motions in Coronal Plane

The Partial Fourier Series (PFS) model is used for motions in coronal plane. The PFS model is previously developed by [5] for the forward walk motion of a simulated Nao robot. The PFS model decomposes a bipedal walk periodic function into a set of oscillators. Each oscillator has four components namely, offset, amplitude, period and phase delay. The period is common for all the oscillators to maintain synchronization among all the joints of the robot. The general formulation is given as follows:

$$JointValue = C + A * sin((2 * \pi * t)/T + \phi) \tag{1}$$

where C is the offset, A is the amplitude, T is the period and ϕ is the phase delay. Joint angular velocities are then set to:

$$\Delta Joint = (targetAngle - currentAngle) * speed \tag{2}$$

In order to achieve the desired gait quality, speed and stability for each motion type, one needs to find the optimal parameter values for each component of the oscillator. In our work, this has been achieved [6] through applying nature-inspired algorithms, mainly, Genetic Algorithm and Simulated Annealing. Simulated Annealing with restarts (SAR), in particular has a better performance in comparison to the original Simulated Annealing or Genetic Algorithm. Therefore, the best parameters are obtained by using SAR. These best parameters produce the desired motion trajectories for each walk motion (Figure 2).

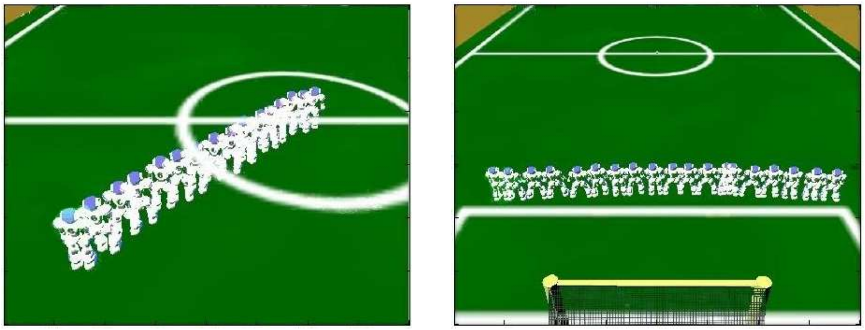

Fig. 2. Two optimized motions in coronal plane: (left) sidewalk, (right) diagonal walk

The objective of the optimization process is to produce the desired motion, with minimum deviation from the expected trajectory, maximum speed and stability. Our experiments show that the PFS model can be applied to produce motion types in coronal plane for a simulated NAO humanoid robot.

All the motion types are then integrated, through a motion transition procedure which performs the necessary calculations to smoothly switch between two arbitrary motion types. Also, the motion model provides odometry data which is then used as a measurement step in localization. Forward kinematics model has been used to find the location of the foot end-effector with respect to torso. Mathematical simplicity of the PFS model helps finding the step length and height. Together with localization methods like Monte Carlo Localization, odometry provides a reliable source of location information even when there are no observed landmarks.

3.2 Kick

The kick motion involves four phases, namely, *expansion*, *preparation*, *execution* and *wrap-around*. In all these phases, the support leg is shifted outwards in order to provide space and stability for the action. Some oscillations are then assigned to each joint of the kicking foot at each phase. In the *expansion* phase, the kicking leg moves along the y axis and is raised from the ground. In the *preparation* phase, the kicking leg is moved backwards along robot's x axis. The purpose is to gather adequate kicking force. In the *execution* phase, a value is assigned to each joint angle and the angular velocity is set to its maximum value. This way, the leg is moved with maximum force and speed towards the ball. Finally, in the *wrap-around* phase, after the ball is kicked, the robot puts both support and kicking legs to their initial positions. Particle Swarm Optimization [7] method has been employed to optimize the parameters of kick motion and necessary joint values at each phase. The future work includes a new learning method for kick motion with a generic alignment approach.

4 Behaviors

There is a layered motion selection architecture for motions, behaviors, plans and cases in the team strategy. Modularity and ease of maintenance could be achieved by using such a hierarchy. While motions operate on the lowest level, behaviors are constucted as sequences of low-level motions. Some behavior examples include *move-to-target*, *turn-to-target* and *dribble-to-target*. On top of these behaviors, static plans are constructed in the form of behavior sequences. The team strategy component uses these plans in a modular way. A sample instantiation of a plan is given in Figure 3.

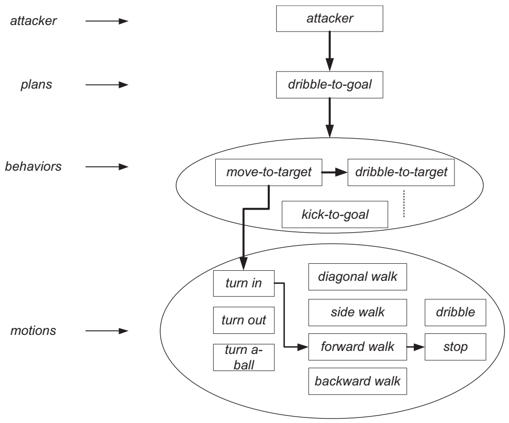

Fig. 3. A sample instantiation of the *dribble-to-goal* plan.

5 Team Strategy and Planning

Since one of the players is directly assigned to the goalie role, it has its own strategy as a FSM. The roles of the remaining robots are determined based on the Case Base Reasoning (CBR) method [8]. In each case, a play-based [9] high-level behavior selection is performed.

A case represents a world situation, the corresponding role assignment and the behavior sequences for these roles. Since the environment is distributed in nature and the available information for a robot is limited with its own view, a distributed implementation of CBR is developed. Behavior sequences represent the behaviors of the agents who are represented in the case. Due to noise and uncertainty in the environment, rather than considering single points of positions, an abstraction is made for the locations. A limited number of cases are hand-coded and new cases are derived during a game. Case-based strategy algorithm runs on the robot which has the attacker role. A standing robot that is closest to ball becomes the attacker.

Two different planning methods will also be developed and integreted into the CBR framework. One of them uses a Hierarchical Task Network (HTN) approach [10] and the other models the problem as a DEC-POMDP and solves it by using dynamic programming approach [11].

6 Conclusion

This team description report outlines different parts of the developed software system for beeStanbul robots. The beeStanbul is an ongoing project and the low-level modules of the architecture were designed and implemented. The future focus will be on the use of efficient planning and team strategies and the integration of learning capabilities for robots.

References

1. Sariel, S.: An Integrated Planning, Scheduling and Execution Framework for Multi-Robot Cooperation and Coordination. Phd thesis, Istanbul Technical University, Turkey (2007)
2. Sariel, S., Balch, T., Erdogan, N.: Multiple traveling robot problem: A solution based on dynamic task selection and robust execution. IEEE/ASME Transactions on Mechatronics **14**(2) (2009) 198–206
3. Sariel, S., Balch, T., Erdogan, N.: Naval mine countermeasure missions: A distributed, incremental multirobot task selection scheme. IEEE Robotics and Automation Magazine **15**(1) (2008) 45–52
4. Thrun, S., Fox, D., Burgard, W., Dallaert, F.: Robust monte carlo localization for mobile robots. Artif. Intell. **128** (May 2001) 99–141
5. Picado, H., Gestal, M., Lau, N., Reis, L.P., Tomé, A.M.: Automatic generation of biped walk behavior using genetic algorithms. In: Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence. IWANN '09, Berlin, Heidelberg, Springer-Verlag (2009) 805–812
6. Asta, S., Sariel-Talay, S.: Nature-inspired optimization for biped robot locomotion and gait planning. In: Proceedings of the 6th European Event on Nature-inspired Techniques in Scheduling, Planning and Timetabling. evoSTIM (2011) 434–443
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks IV. (1995) 19421948
8. Ros, R., Arcos, J.L., Lopez de Mantaras, R., Veloso, M.: A case-based approach for coordinated action selection in robot soccer. Artif. Intell. **173** (June 2009) 1014–1039
9. Mcmillen, C., Veloso, M.: Distributed, Play-Based Coordination for Robot Teams in Dynamic Environments. In: RoboCup 2006: Robot Soccer World Cup X. Springer-Verlag, Berlin, Heidelberg (2007) 483–490
10. Hayashi, H., Tokura, S., Ozaki, F.: Towards real-world htn planning agents. In Jain, L., Nguyen, N., eds.: Knowledge Processing and Decision Making in Agent-Based Systems. Volume 170 of Studies in Computational Intelligence. Springer Berlin, Heidelberg (2009) 13–41
11. Wu, F., Zilberstein, S., Chen, X.: Online planning for multi-agent systems with bounded communication. Artificial Intelligence **175**(2) (2011) 487 – 511