Scorpius Robocup 2013 Team Description Paper

Sadegh Mokari, Amir Hossein Seyri, Fateme Darvishvand, Elnaz Soleymani, Sina Mir Hejazi, Pantea Habibi, Saleh Khazayi

{mokari, seyri, darvishvand, e.soleymani, m.hejazi, p.habibi, s.khazayi}@aut.ac.ir

Abstract. In this paper, we will describe Scorpius soccer simulation team researches and recent activities in order to qualify for competitions ahead. Most of these researches focused on rewriting the new base code and procedure of using the optimization unit to generate skills and improve quality of these skills.

1 Introduction

Soccer simulation 3D challenges using humanoid robots have started its activities since 2007. At the same time many Iranian students started their researches on this branch of robotic science. Scorpius and Parsian were two teams that participate in many Robocup challenges. Since 2012 our team has started its activities with old members within some new members. These studies focused on writing a new base code and solving basic problems for optimizing procedures and approaches. These researches will be proposed at next parts. This paper is organized as follows. At the first, we will briefly describe team structure, updating mechanisms and kinematics calculations. Then optimization unit and optimization process on humanoid robots skills such as walking, kicking ball and diving will be proposed. The next section introduces multi agent decision process. At the last part we summarized the researches and discussed the future activities.

2 Agent Architecture

In this section we describe base code structure. Agent structure has been designed on simulation process that starts from receiving data from server and updating World-Model. Then decision starts depending on team strategy and agent's decision. At the end results are being sent to simulation server.

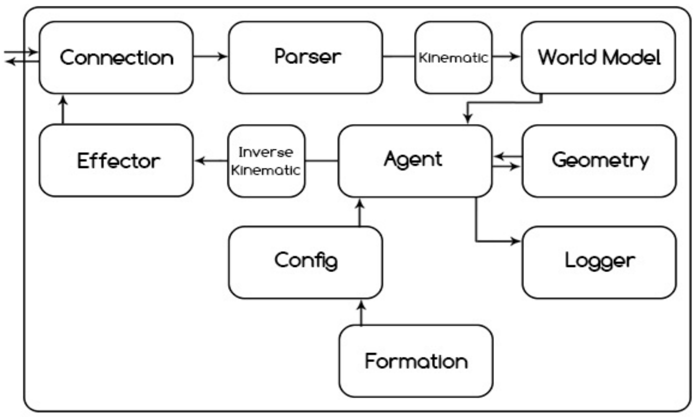

Fig. 1. Scorpius Agent Architecture: Connection unit connects agent to the server, and passes data form server to the parser unit. Then WorldModel will be updated using these data. Also in order to obtain coordinates of each body section such as hands, legs and torso, it uses Kinematic unit. The agent makes a decision using WorldModel data, formation and geometry tools and sends it to the server by using some classes such as Effector and Connection.

Updating Process

In this section we have a brief description of how to update the environment by using the information of simulated sensors. Generally the most important part of each process is updating the location. For this action we need to neutralize the effect of the rotation of neck and body angels on the fundamental information. For doing this, first we neutralize the effect of the neck rotation by using the kinematic matrix of camera's position from body's position, then we neutralize body's rotation matrix with the information of Gyroscope sensor. Now we can update all information according to this two rotation matrixes. By this information and the position of fixed points we can estimate the position of robot and other moving objects.

To calculate the inverse of body's rotation for neutralizing it, we have to multiply the previous rotation to the new rotation since the information of Gyroscope sensor is obtained according to previous frame. These operations face some errors after several successive intervals; therefore we use the position of fixed points and lines to decrease this kind of errors. Also the information of another sensor combines with the information of vision sensors and Gyroscope sensors. We use accelerometer sensor's data for distinguishing if the robot is fallen on the ground. When the robot is standing and has balance this parameter is equal to the gravity acceleration. When the robot loses its balance and perches on the verge of falling, this parameter starts to decrease so that if this value be smaller than the threshold the robot means it has fallen. By observing this sensor's data we can obtain the kind and direction of falling.

Kinematic Module

This module will do the kinematic calculations. In forward kinematic part the important point is finding the center of mass. In this base code, forward kinematic is been used for calculating the position of joints. For calculating position of Torso, head position is been calculated, and then Torso position is been obtained by multiplying head position to rotation matrix, and Torso position will be calculated.

Now by multiplying mass of every unit to its position and calculating average of them, we can calculate center of mass.

$$CoM = \sum_{i=1}^{n} \frac{m_i \times \boldsymbol{r_i}}{\Sigma_{i=1}^{n} m_i} \qquad (1)$$

Also in inverse kinematic, equations were been simplified and we have closed form equations. This unit is been designed to facilitate generation and optimization skills.

3 Behaviors

Behaviors are divided into two general categories: group and individual skills.

Individual skills are necessary for multi agent environments. Therefore, first the robots must achieve a significant level of individual skills. Thus, the main approach has been based on the design and optimization skills so far. In our approaches skills are been considered as an optimizing problem. Generally each skill can be modeled as a multi objective multi constraint optimization problem. These objectives can be different for each skill.

Real world space model is been used for robot parts to represent the skills. The basic pattern can be considered for developing the skill by the optimizer module in order to reduce the search space.

In the next part walking is been described as an example for these skills and its optimization's process is been discussed.

3.1 Walking skill.

Walking pattern can be generated easily. We generally distinguish Walking from Running in this way: only one foot at a time leaves its contact with the ground and also there is a period of double-support. In contrast, walking begins when both legs are off the ground with each step. So we can generate this pattern with three stages, at first stage body initializes. At two next stages one foot stands on the ground and the other foot moves forward. These stages are defined by side of swinging foot. For walking straight forward there is an important fact that 'speed of walking' and 'walking stability' should be optimized together. Also there are some constraints in robots such as velocity and angle limits of each effector.

Walking frequency is considered constant; therefore the number of gait periods is constant. In optimization, in each step of walking, trajectories of robot's heap and swinging foot are changed and coded.

Main point of these trajectories is that, in walking, foot must come up from ground and after moving forward it comes down; thus, in this movement sum of trajectory changes of swing foot in z axis is zero. This point is considered in gait coding. Simulation process for gait optimization is very slow and time consuming, so we use parallel evolution with many robots. However this approach decreases the time of evaluation effectively but the process of optimization is very slow due to long time of simulation process. So we use a unit called 'offline agent' to approximate walking fitness and compensate the constraints.

Offline agent

This unit calculates all of inverse kinematics and does body control calculation. We use prediction of joints next angle to reach this goal. With this method we will not need simulation server for simulating agent's joints.

According to small noise of effectors, this unit makes good predication for effector motions. Because we code body motions to real world positions for calculation of walking speed, we can add ankles changes in one period of walking. Also we use inverse kinematics equation for calculating angle of each effector and evaluate constraints of effector velocities limits and effector angles limits. We can evaluate the skill with satisfying limitation of effector velocities and angles. Also we can obtain fitness of the skill.

This unit can be used for optimization agent's skills without using simulation server. This approach helps us to optimize skills about hundred times faster than when we use the simulation server.

3.2 Goalkeeper agent skills

This agent, according to the rules of the game, has more freedom than other agents, such as using the body to touch the ball. In addition to walking skill that will be generally described in future, on occasions this agent should be able to dive to make a Barrier in front of the goal so the opposite team cannot score a goal. These skills are described in the upcoming section, also the quality of getting in proper place with the ball position and a simple decision for this agent, is described in the end.

Diving

One of the team's defensive skills is diving. The issue arises when the players can perform better in front of the opponent's move that is not like a shot. These skills can also be used for more advanced goalie to practice defensive actions, the case that if more time is can raise his. Otherwise, it simply does. Diving is divided into two parts: Diving straight and diving directly into the surrounding. Diving straight in if he gets down on his hands, is a defensive action for all of the players.

Diving straight such that Fall on the front of the body, will be done by the goalkeeper to defend the goal.

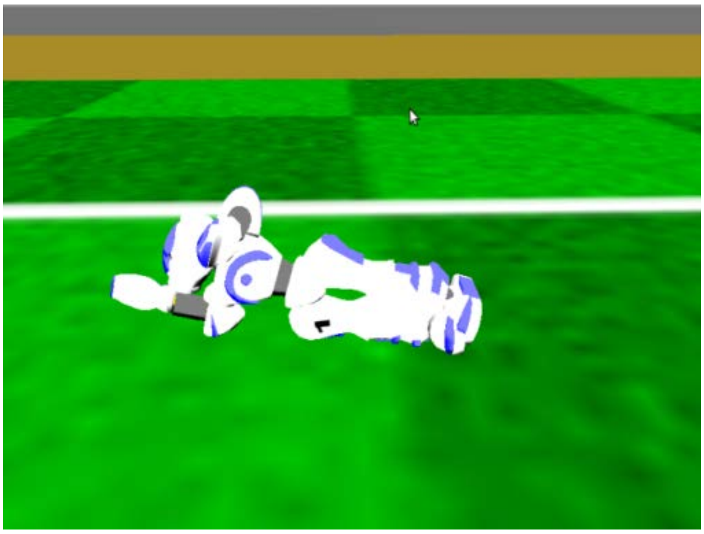

Fig. 2. Diving

3.3 Decision making and positioning

For proper positioning we need to classify these: Mathematical concepts, the goal of goalkeeper, the goalkeeper functions, the original location of the goalkeeper, the allowable amount of coming out of the goal, ranging defense, ...

In mathematical concepts, to put the goalkeeper in the best position when opponent team attacks, we use the "cosine theorem”. For locating the ball or opposing player, by the use of flags around Gridiron and Drawing, two imaginary circles with a radius R (the distance between two flags = R) are obtained by two points that One point is inside the Gridiron, that is the position of the ball or players. Goalkeeper tasks can be more than the defensing of the goal. For example when the ball is in our possession, the goalkeeper can come forward to attack the middle of Gridiron.

Primary location of the goalkeeper (half a meter ahead of the goal) can be evaluated properly.

4 Conclusions and Future activities

With the development of the optimizer the multi-objective, many of activities such as walking in all the ways, and Turning robots have been developed and optimized. Also work on appropriate optimization of shoot module is being done. Multifactor decision modules have a very simple form.

The future activities of team will be conducted on Optimization, sequential thinking skills and team playing with the multifactorial Thinking.

About multifactorial Thinking, agent movements are divided into two categories: Movement without the ball such as ‘being in the right place’, and Movement with ball in possession such as ‘passing the ball’.

References

1. *A simple method for decision making in robocup soccer simulation 3d environment.* **K.N. Maleki, M.H. Valipour,R.Y. Ashrafi, S. Mokari and MR. Jamali and C. Lucas.** s.l. : Avances en Sistemas e Informatica, 2008.
2. *A uniform biped gait generator with offline optimization and online adjustable parameters.* **L. Yang, C.M. Chew, T. Zielinska, and A.N. Poo.** 5, 2007, Robotica, Vol. 25, pp. 549–565.
3. *Scorpius Team Description Paper Soccer Simulation 3D league.* **K.N. Maleki, M.H. Valipour,R.Y. Ashrafi, S. Mokari and MR. Ja-mali and C. Lucas.** Singapore : s.n., 2010.
4. *Polytechnic-Parsian RoboCup 2010 Team Description Paper.* **Jahangiri, Soheil and Khalifeh, Mohammad Taghi and Moradi, Karim and Nikan, Hojjat and Nadi, Abolfazl and Sarvari, Zabihollah and Vatankhah, Hedayat.** 2010.