In the name of God

Invictus3D RoboCup 3D Simulation League Team Description Paper 2016

Amirhosein Izadjoo, Farzad Jafari Aghdam, Behdad Ahmadi Neishaboori, Mojtaba karimi Nia, MohammadHosein Musavi, MohammadReza Gomar, Ardeshir Shojaie Nasab

Shahid Beheshti University

Sbceinvictus3d@gmail.com

Abstract. This paper describes the ideas, and research carried out through novel algorithms used by the Invictus3D soccer simulation team aiming to optimize the performance of humanoid soccer agents. Our agent's performance is currently based on several physical and AI algorithms used for localization and skills such as walking and kicking which will be briefly explained in this paper.

1. Introduction

Shahid Beheshti University has had a team in soccer simulation 2d.Following these successes, work on RoboCup 3D soccer simulation league kicked off in the form of introducing a team to the 3D soccer simulation league which have begun researching since 2014.The main challenge in 3D soccer simulation is the control of a humanoid robot while satisfying both the physical constraints of the robot and the official rules of a soccer game. In order to achieve this goal a number of physical, A.I and learning algorithms have been used which can be optimized through the usage of A.I methods. By using simulated robot agents instead of actual NAO robots we obtain several advantages such as being able to experiment on the robots in shorter time and less money expenses. Thus the RoboCup organization's goal of using 11 humanoid soccer bots to defeat an 11 man football team can be achieved much sooner.

This paper is structured as follows: section 2 the system architecture is discussed, followed by agent's behavior and skills in section 3 and future works in section 4.

.2. System Architecture

Our overall system architecture (Fig. 1) can be broken-down into several modules that interact with each other. The server communication module is in charge of establishing a two-way connection between the simspark simulation server and our agent while decoding incoming messages into the agent's world model and encoding outgoing ones so that they are ready to be sent to the server. Our agent layer is responsible for the agent's basic skills and behaviors including localization, standing up, walking and kicking. Our agent's final actions during a cycle are determined using information provided by the server which each agent can access through its own world model. These actions are returned to the connection class ready to be encoded and sent back to the server.

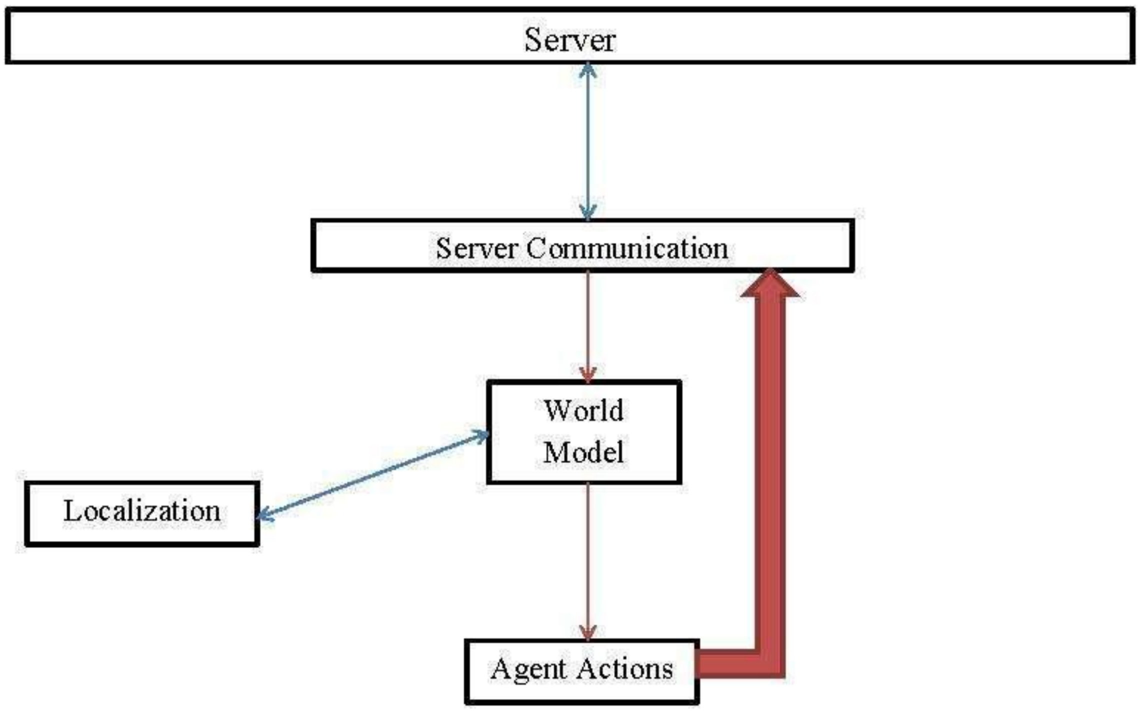

Fig. 1. Overall base architecture

3. Agent Behavior and Skills

In this section we will discuss the methods and algorithms used in order for our agent to localize itself and other objects. We will also discuss how walking skill functions are.

3.1 Localization

This section describes the different methods we used for self-Localization and ball localization

3.1.1 Self-Localization

These methods are used by our agent in order to obtain its position. We prefer the first method and the second method is used if our agent cannot see more than two flags.

3.1.1.1 Localization with two flags or more

Our flags heights are always constant therefore we can always obtain our agent's location by using the coronal plane. If there are more than two flags in our agent's sight we will pair them up and use this process on each pair and finally in order to obtain our position we will use weighted average on the results of this process.

$$(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2 = r_1^2 \quad (x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2 = r_2^2$$

3.1.1.2 Localization with one flag and body direction

The server sends us the position of a flag. We use these flag positions and a gyroscope to find out what our body direction is. Then we use this body direction and flag position to localize our agent's position. Using a gyroscope for a long time will result in a high error.

3.1.2 Ball localization

In order to score a goal we have to obtain the location of the ball. There are only two situations, we either have the ball in our sight or we don't. In the first condition the server gives us the coordinates of the ball and in the second condition we try to keep the ball in our sight while trying to have a better view on court so that we can see more flags when looking at the ball.

3.2 Walking

Creating a stable, fast and at the same time flexible walking is perhaps the biggest problem in 3D soccer simulation. Currently we use a static walking method based on physical factors such as Center of Mass (COM) and Zero Moment Point (ZMP) which was first proposed by [1] for biped humanoid robots during an IEEE Robotics convention. By satisfying the constraints noted in this method we receive a set of trajectories, afterwards we use inverse kinematics to find out the required angular velocity in order to achieve these trajectories.

3.2.1 Walking cycle

A complete walking cycle is composed of two phases: double-support and single support. The double support phase is when both feet are on the ground and during the single support phase one foot remains steady on the ground while the other foot swings from the rear to the front. In order to achieve stability during these phases we need to satisfy specific constraints which will be discussed in the next section.

3.2.2 Trajectories

In order to achieve a stable walking a set of constraints need to be fulfilled which lead to specific trajectories. This will be discussed in this part. Note that the trajectories we obtain are only for the sagittal plane since we are using a static walking method which was first proposed by [1].

We obtain the following constraint for our ankle trajectory:

$$\theta_a(t) = \begin{cases} q_{gs}(k), & t = kT_c \\ q_b, & t = kT_c + T_d \\ -q_f, & t = (k+1)T_c \\ -q_{ge}(k), & t = (k+1)T_c + T_d \end{cases}$$

$$x_a(t) = \begin{cases} kD_s, & t = kT_c \\ kD_s + l_{an}\sin q_b + l_{af}(1-\cos q_b), & t = kT_c + T_d \\ kD_s + L_{ao}, & t = kT_c + T_m \\ (k+2)D_s - l_{an}\sin q_f - l_{ab}(1-\cos q_f), & t = (k+1)T_c \\ (k+2)D_s, & t = (k+1)T_c + T_d \end{cases}$$

$$z_a(t) = \begin{cases} h_{gs}(k) + l_{an}, & t = kT_c \\ h_{gs}(k) + l_{af}\sin q_b + l_{an}\cos q_b, & t = kT_c + T_d \\ H_{ao}, & t = kT_c + T_m \\ h_{ge}(k) + l_{ab}\sin q_f + l_{an}\cos q_f, & t = (k+1)T_c \\ h_{ge}(k) + l_{an}, & t = (k+1)T_c + T_d \end{cases}$$

And the following constraints are obtained for our hip trajectory.

$$z_{\mathrm{h}}(t)=\begin{cases} H_{\mathrm{h\,min}}, & t=kT_{\mathrm{c}}+0.5T_{\mathrm{d}} \\ H_{\mathrm{h\,max}}, & t=kT_{\mathrm{c}}+0.5(T_{\mathrm{c}}-T_{\mathrm{d}}) \\ H_{\mathrm{h\,min}}, & t=(k+1)T_{\mathrm{c}}+0.5T_{\mathrm{d}}. \end{cases}$$

$$x_{\mathrm{h}}(t)=\begin{cases} kD_{\mathrm{s}}+x_{\mathrm{ed}}, & t=kT_{\mathrm{c}} \\ (k+1)D_{\mathrm{s}}-x_{\mathrm{sd}}, & t=kT_{\mathrm{c}}+T_{\mathrm{d}} \\ (k+1)D_{\mathrm{s}}+x_{\mathrm{ed}}, & t=(k+1)T_{\mathrm{c}}. \end{cases}$$

$$x_{\mathrm{h}}(t) = \begin{cases} kD_{\mathrm{s}}+\dfrac{D_{\mathrm{s}}-x_{\mathrm{ed}}-x_{\mathrm{sd}}}{T_{\mathrm{d}}^2(T_{\mathrm{c}}-T_{\mathrm{d}})}\left[(T_{\mathrm{d}}+kT_{\mathrm{c}}-t)^3-(t-kT_{\mathrm{c}})^3\right. & \\ \quad \left.-T_{\mathrm{d}}^2(T_{\mathrm{d}}+kT_{\mathrm{c}}-t)+T_{\mathrm{d}}^2(t-kT_{\mathrm{c}})\right]+\dfrac{x_{\mathrm{ed}}}{T_{\mathrm{d}}}(T_{\mathrm{d}}+kT_{\mathrm{c}}-t)+\dfrac{D_{\mathrm{s}}-x_{\mathrm{sd}}}{T_{\mathrm{d}}}(t-kT_{\mathrm{c}}), & t\in(kT_{\mathrm{c}},kT_{\mathrm{c}}+T_{\mathrm{d}}) \\ kD_{\mathrm{s}}+\dfrac{D_{\mathrm{s}}-x_{\mathrm{ed}}-x_{\mathrm{sd}}}{T_{\mathrm{d}}(T_{\mathrm{c}}-T_{\mathrm{d}})^2}\left[(t-kT_{\mathrm{c}}-T_{\mathrm{d}})^3-(T_{\mathrm{c}}+kT_{\mathrm{c}}-t)^3-(T_{\mathrm{c}}-T_{\mathrm{d}})^2(T_{\mathrm{c}}+kT_{\mathrm{c}}-t)\right. & \\ \quad \left.-(T_{\mathrm{c}}-T_{\mathrm{d}})^2(t-kT_{\mathrm{c}}-T_{\mathrm{d}})\right]+\dfrac{D_{\mathrm{s}}-x_{\mathrm{sd}}}{T_{\mathrm{c}}-T_{\mathrm{d}}}(T_{\mathrm{c}}+kT_{\mathrm{c}}-t)+\dfrac{D_{\mathrm{s}}+x_{\mathrm{ed}}}{T_{\mathrm{c}}-T_{\mathrm{d}}}(t-kT_{\mathrm{c}}-T_{\mathrm{d}}), & t\in(kT_{\mathrm{c}}+T_{\mathrm{d}},kT_{\mathrm{c}}+T_{\mathrm{c}}). \end{cases}$$

Note that on level ground Θ_h is constant and equal to: 0.5

3.2.3 Inverse kinematics

We need to pass our obtained trajectories to the inverse kinematics function in order to receive our joints angular velocities. This function receives input in the form of a matrix containing the foot position compared to the torso therefore we must convert our global foot position to match this requirement. In order to do this we subtract the global hip position from the global foot position.

3.3 Kicking

After discussing different types of kicking, we concluded that the best way to develop a kick Is using learning or optimization methods. So we checked different learning and optimization methods (RL , Imperialist Competitive Algorithm , PSO , Genetic Algorithm) and we came up with Genetic Algorithm. our shot has three steps which the first two are static and the third step will be determined by using genetic algorithm. To determine the third step we had about 2048 chromosomes as our starting population (21 of these shots were written by team members and the rest of the shots were made randomly). We applied genetic algorithm for 112 generations and after this we got to a point that the algorithm didn’t made any better chromosomes.

4 Future works

We are currently working on our walking system using genetic algorithm. But it’s not completed yet. But we believe that it’ll be completed before the competition. Also we’re currently working on our skill changing and plans.

References

1.Huang, Qiang, Planning Walking Patterns for a Biped Robot, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, june 2001, VOL. 17, NO. 3

2. Spong, Mark, NOVEMBER 2005, Robot Modelling and Control, -----------, Wiley, 1st Edition

3. Jazar, Reza, Theory of Applied Robotics: Kinematics, Dynamics, and Control, --------, SPRINGER, 2nd Edition

4. J.J. Alcaraz-Jim´enez, D. Herrero-P´erez, and H. Mart´ınez-Barber´a, Motion Planning for Omnidirectional Dynamic Gait in Humanoid Soccer Robots, JOURNAL OF PHYSICAL AGENTS, JANUARY 2011, VOL. 5, NO. 1

5. Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yok oiand Hirohisa Hirukawa, The 3D Linear Inverted Pendulum Model: A simple modeling for a biped walking pattern generation, Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001

6. Patrick MacAlpine, Daniel Urieli, Shivaram Kalyanakrishnan and Peter Stone, UT Austin Villa 3D Simulation Soccer Team Description Paper 2011

7. Shahriar Asta, Tuna Sonmez, Onuralp Ulusoy, Alper Alimoglu, Mustafa Ersen, Ozdemir Sozmen, and Sanem Sariel-Talay, beeStanbul RoboCup 3D Simulation League Team Description Paper 2011