Nexus3D Team Description Paper

Reza Etemadi, Hamid Sajjadi, Arvin Fathi

Nexus Simulation and Modeling Lab, Department of Computer Engineering, Ferdowsi University
of Mashhad, Iran

nexus@um.ac.ir

Abstract. Nexus 3D is one of the oldest team participating in different 3D soccer simulation competetions since 2003. These competetions are a source of motivation for doing more research in the field of AI and robotics and testing the result of it with the state of art and other teams across the world. In this paper we try to present a description for our current work on Nexus3D team. We have moved to Austin villa base, and used our own algorithms for different porpuses on their base.

Keywords: Soccer Simulation, Nao, Simspark, Humanoid, Robotics, Inverse Kinematics

1 Introduction

Robocup soccer simulation is a platform which is client-server multi agent.in this platform server for communicating with each one of the clients sends the state of the world to them and they reply with specific commands at each cycle.

Being cheap and easy-to-use, and also exciting competitions, motivates researchers all over the world to participate and contribute to the state of art of robotic and AI decision making and has made it more popular in recent years. 3D soccer simulation has a three-dimensional environment which has made this platform ideal for studying humanoid robot control and decision making. The current developments of 3D soccer simulation server uses a humanoid robot model known as Nao as soccer playing agents, which can be controlled by a low-level interface so teams can improve and develop robot's control decisions.

We have moved our code and algorithms to UT Austin Villa Robocop's 3D simulation team base code and implemented our own algorithms based on it. In section 2 we explain what is UT Austin Villa base code and why we moved on it, then we describe our localization in section 3, and after that we speak about our agents' skills and behaviors in section 4. Finally, we speak about decision making and discuss our new method for positioning in section 5.

2 UT Austin Villa Base Code

UT Austin Villa 3D soccer simulation team is one of the most prominent teams in 3D soccer simulation competitions. They are successful both in the match field and research field. To help the community of 3D soccer simulation and contribute to it, they have released their base code in 2016. The base code includes basic communication, walk and skills. But optimized walking engine or advanced skills such as diving and long kicks are not included in the source code.

After its release, many teams decided to move on their base code and start their team from there. Nexus3D is not an exception. Our team was in a state that improving significantly needed some in-depth changes in our source code and our approach to implementing skill selection. Also, our walk was fast but not as stable as we wanted, especially when we wanted to go behind the ball to use a skill. These problems are where UT Austin Villa has reasonable result and functioning. As a result, we decided to move on their base code and work on more advanced problems instead of putting plenty of time and energy on changing our own base.

In following sections, we explain some of the algorithms we have moved from our own base code, or have implemented from scratch, on UT Austin Villa base code.

3 Localization

Robots get relative positions of all observable objects through their cameras. Knowing the position of 8 landmarks around the field, we can use landmark methods for agent's localization. All we need is to map the relative coordination system to a global one.

From different methods available for this purpose, we use a two-flag localization method in which the transform matrix is generated by using two different flags. We can easily calculate the position of agent on a plane which is created by the position of two flags and the agent itself. Then we need rotation matrix of this plane to map 2D positions on it. By using forward kinematics, we can estimate the height of agent and with having the height of agent and two flags, we can calculate the orientation of plane, therefore we can find its rotation matrix. By using reversed rotation matrix, we can map 2D coordination on plane to the global 3D coordination.

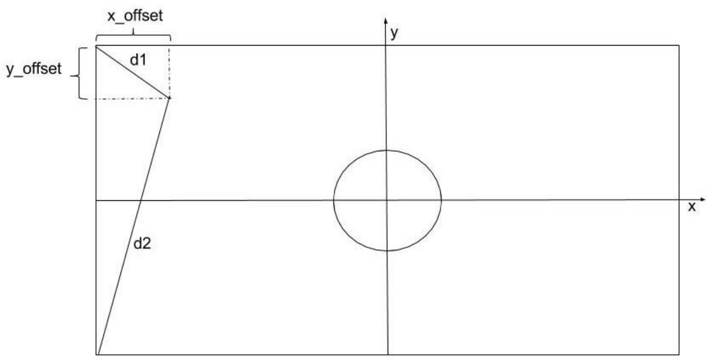

Fig. 1. Localization using two flags

4 Skills and Behaviors

4.1 Shoot

Shooting skill should be able to kick the ball to a given position which can be far away from the shooting position. To achieve a desirable shoot two factors should be considered, precision and power. We focused on the power of the shoot and tried to optimize it in a way that the maximum power is reached and for this purpose we used CMA-ES algorithm.

As the initial state for optimization, the robot is standing on one foot as stance foot and the other foot is ready to swing and kick the ball, shown in fig. 2, then the optimization process is performed on a basic and inefficient joint movements to enhance the velocity and angles of them.

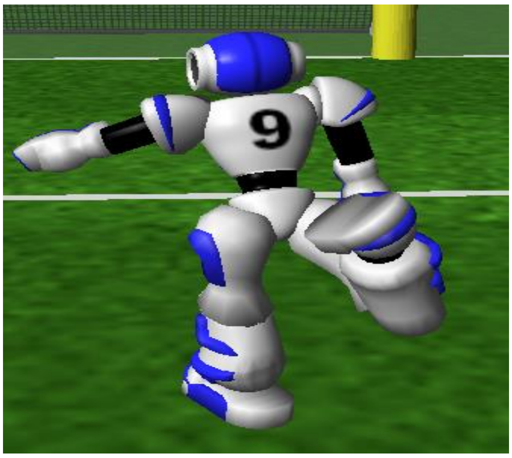

Fig. 2. Initial state for optimizing shoot

4.2 Behaviors

We have all type of skills some of them are low level and on the other hand some of them are high level, combination of some of these low level skills together creates a concept called behavior. Although behavior is combination of some of low level skills it is a high level skill and it need to gain some information about environment. For example one of behaviors we have is dribble which uses two skills one of them is walking and the other one is kicking which depend on the state of the environment. Actually

When we call dribble behavior, agent moves toward the ball and kick it in an anticipated direction. By repeating this sequence, dribble behavior is achieved.

4.2.1 Interception

Intercepting the ball can be tricky sometimes. To prevent agents from collisions and to acquire best results we need to rate our agents according to their success probability. To calculate the probability and therefore the score we used some parameters like ball average speed, opponent average speed, our player average speed and distance of each agent from the ball. Another important parameter which can be missed but plays a significant role in intercepting is the angle between the ball and our goal. With these parameters we can calculate if opponents reaches the ball sooner than our agent. If so the interception position will be set according to opponent speed and its direction. Otherwise the ball will be interception point.

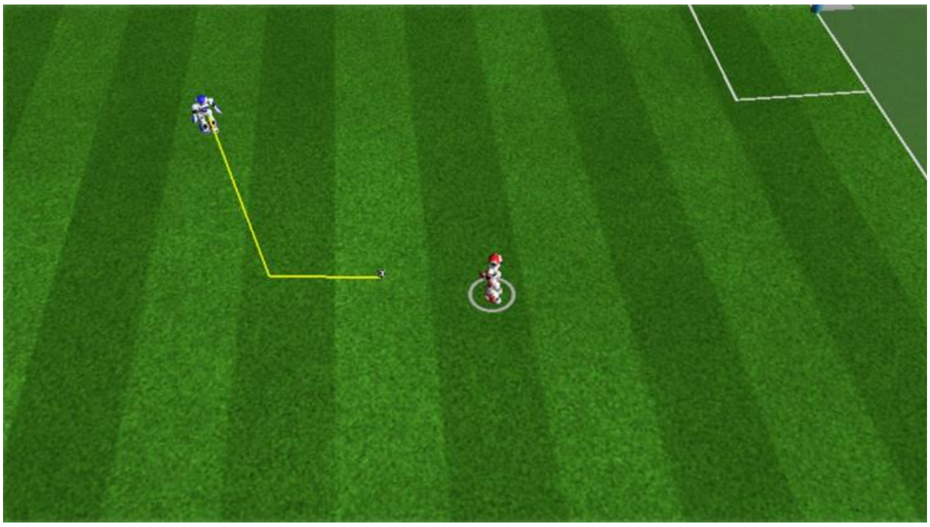

Fig. 3. Yellow line shows calculated position for interception

Sometimes the distance to ball is not the only parameter promising a good interception. To avoid risks, another important factor is the angle of player to ball. If the distance between agent to ball angle and ball to goal angle is not high, it means that our player is directly going toward opponent and ball so it can clear the ball with more stability. Therefore the angle can play a more important role in prioritizing this agent to maybe a closer agent to the ball.

5 Positioning

One of our main problems was setting formations based on the current flow of the game. We needed to create a formation file for different states in the game based on position of the ball. It was both time consuming and static. Another major problem with this method is that it is based on the location of the ball and opponent's position had no impact on the positioning of our players at all. For the first problem, we decided that we need a new method to implement formation and positioning of our agents. We used FEdit to do so. FEdit is an open source software for 2D soccer simulation and allow teams to set some important formations static.

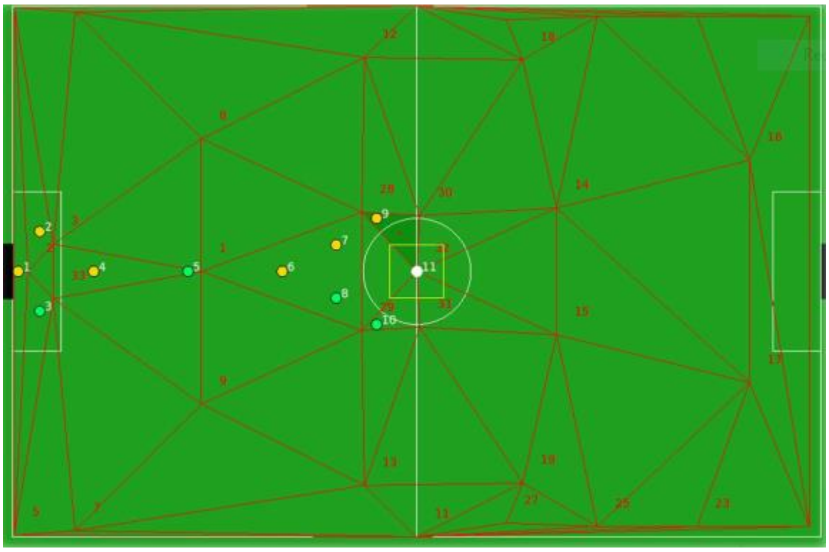

Fig. 5. Setting formation in FEdit

FEdit uses Delaunay Triangulation algorithms to guess the best position for other states. We used FEdit and changed it so it fits the 3D soccer simulation field size and used it to decide the positions for our roles and assign agents to those roles.

Fig. 6. Using the same formation in 3D simulation

FEdit solved the issue about creating formation files and its time consuming process. But as we mentioned there was another problem. Opponent players positions had no impact in our players positioning. Because of that we implemented a marking algorithm which makes players in the middle, that are not too close to the ball or in our defensive area, always stay on the line between opponent player and our goal.

6 References

1. M. Alehdaghi, M. Abolfazli Esfahani, N. Erfani, S. Farokhnia, R. Etemadi, H. Sajjadi, Me. Abolfazli, "Nexus 2017 3D Soccer Simulation Team Description"
2. P. MacAlpine, P. Stone "UT Austin Villa RoboCup 3D Simulation Base Code Release", Proceedings of the RoboCup International Symposium (RoboCup 2016), Leipzig, Germany, July 2016.
3. M. Alehdaghi, M. Abolfazli Esfahani, N. Erfani, S. Farokhnia, H. Ghayour, S. Askari, H. Sajjadi, "Nexus 2016 3D Soccer Simulation Team Description"
4. A. Ildermi, M. Alehdaghi, I. Kiarazm, M. Niazi, A. Gharib, M. Shokri, "Nexus 2011 3D Soccer Simulation Team Description"
5. M. Alehdaghi, A.Ildermi, et. al, "Nexus3D Base Code"
6. V. Salmani, M. Naghibzadeh, F. Seifi, A. Taherinia, "A Two Phase Mechanism for Agent's Action Selection in Soccer Simulation", The Secod World Enformatika Conference, WEC'05, Istanbul, Turkey, pp. 217-220 February 2005
7. Daniel Joseph Paluska, "Design of a Humanoid Biped for Walking Research", Master's thesis, 2000
8. Jerry Pratt and Gill Pratt, "Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation", Conference on Climbing and Walking Robots, 1999

9. Knight, R, and Nehmzow, U., "Walking Robots, A Survey and a Research Proposal", Technical Report

10. Krister Wolff and Peter Nordin. "An Evolutionary Based Approach for Control Programming of Humanoids