In the name of God

S4M

Robo Cup 3D Simulation League

Team Description Paper 2013

Mohammad Amin Haghpanah, Farzad Jafari, Mostafa Najafi, Sepand Seraj

S4M.Helli3@gmail.com

Abstract. This paper describes the ideas, and research carried out through novel algorithms used by the S4M 3D 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, turning, kicking and path planning which will be briefly explained in this paper.

1. Introduction

Allameh Helli 3 High School is a branch of Iran's NODET (National Organization for Development of Exceptional Talents) Schools which has been participating in Robo Cup competitions since 2008 and has achieved great success during this short time. The High School has had a number of teams compete in the junior soccer and junior rescue leagues. Following these successes, work on Robo Cup simulation leagues kicked off in the form of introducing two teams to the 3D soccer simulation league which have begun researching since 2011. Some members in these teams already have experience from previously competing in the junior soccer and junior rescue leagues and a selected group of the two teams formed the S4M team.

The main challenge in 3D soccer simulation is the control of a humanoid robot with 22 DOF (Degrees Of Freedom) 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 and A.I 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 expenses. Thus the Robo Cup organizations 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: in section 2 the system architecture is discussed, followed by agent’s behavior and skills in section 3 which is followed by the conclusion and the future plans 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 by parsing incoming messages into the agent’s world. 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 sent back to the server. It should also be noted that we used team Asfaar’s original base and mounted are own skills on this base and we also used the Nexus team’s Inverse kinematics.

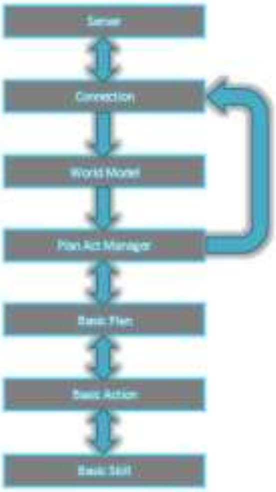


Fig. 1. Overall base architecture

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 and shooting skill functions are.

3.1 Localization

Localization can be done through several methods the methods using more than one flag have been solved for a long time and can be solved by several equations so we only explain localization methods with one flag or less.

3.1.1 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 so we try to use this method as rarely as possible.

The way

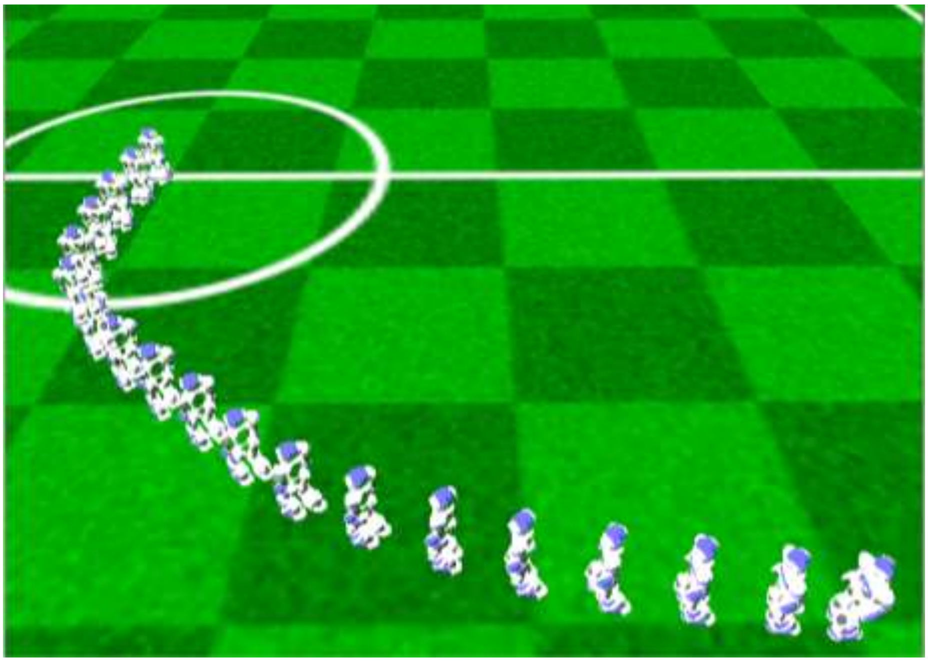
3.2 Walking

Creating a stable, fast and at the same time flexible walking is perhaps the biggest problem in 3D soccer simulation. We can find a lot of inspiration in the way human walks. The human changes direction during a single step so we implemented the same way of walking on our robot which is a lot faster than going in a straight line and turning.

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 on the ground going backwards and shifting the body forward while the other foot is in the air and stays in a specific location. In order to achieve stability during these phases we need to satisfy specific constraints.



3.3 Path Planning

One of the most important things in 3D soccer simulation is to find a path for the robot at any moment to be able to reach its destination without hitting any obstacles along the way. In order to do so first we need to know the exact location of the obstacles and build a path based on their location. The path planning function's inputs are starting point, final point, starting angle, final angle and the location of the obstacles and its output is an array of location which will be then given to the go to point function.

Our approach to the problem is to divide the whole problem in to two layers:

3.3.1 First Layer

In this layer we solve the problem without the obstacles and only based on the starting and the final angle.

3.3.1.1 Solution

Let A be the starting point, B the final point, α the starting angel and β the final angle.

The first thing we do is dividing the path in to arc and a line for doing so we find the point that lines crossing A with α angel which will be called PA and B with β which will be called PB cross each other and we name it P then if

PA > PB we split PA into two parts one of which equal to PB and we name that point Q.

Now we find the curve that PA and PB are tangential to on Q and B. the output of this layer is the QMB arc and the AQ line. (Fig.2)

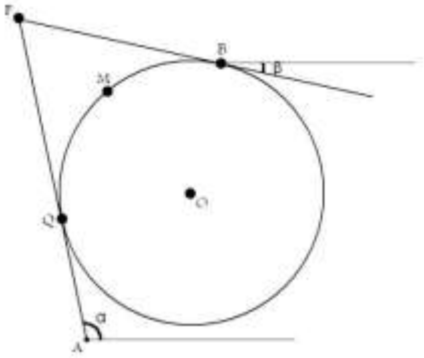


Fig. 2. Solution

If the line crossing A with α angel and B with β crossed each other the output path will look like Fig.4 but if they did not cross each other because of exceptional angels the path will look like Fig.4.

In case the lines did not cross each other we put a point called M in the middle of A and B then we split the problem in two problems first is to go from A to B with the final angel of α+10 and from M to B with final angel β and each of these parts can be solved in the normal mode.

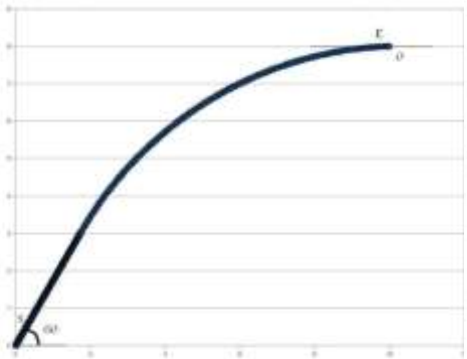


Fig 3. Normal mode

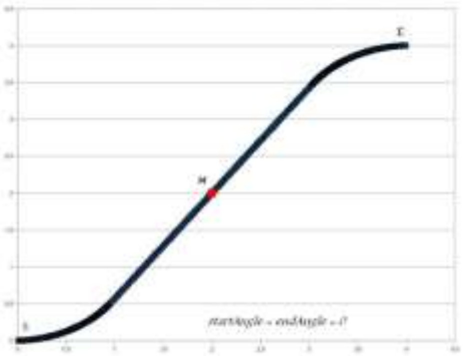
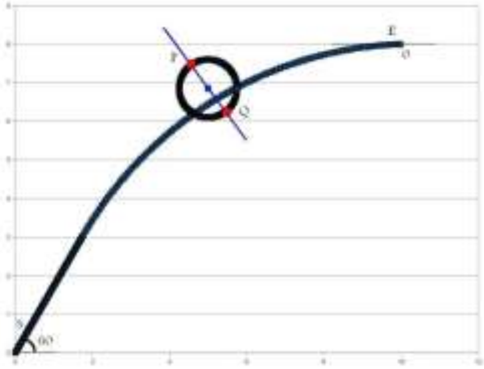
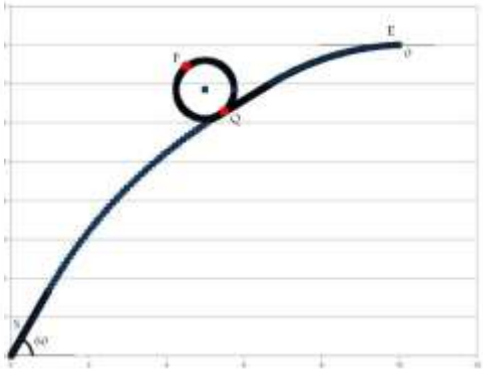


Fig 4. Exceptional mode

3.3.2 Second Layer

The first thing we do in this layer is to assign an area to every obstacle which is a circle centered by the obstacle itself. Then we find those obstacles that the path made in the second layer goes through their area then we draw a vertical line from the selected obstacle. If the obstacle was located on the line part of the path we draw the line vertical to the path and if the obstacle was located on the arc part of the path we connect draw the line connecting the obstacle to the center of the circle that the arc is located on will cross the obstacle's circle at Q and P points then by connecting all the Ps and Qs to each other we make a graph as shown in the picture.





The weight of every edge is the distance of the path connecting them to each other which will be calculated by calling the first layer with the position of the two ends of the edges and if the path connecting two of these points crossed any obstacles we will remove it.

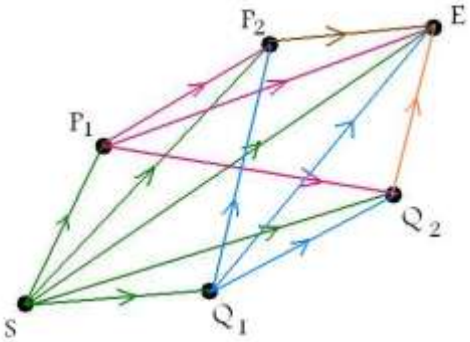
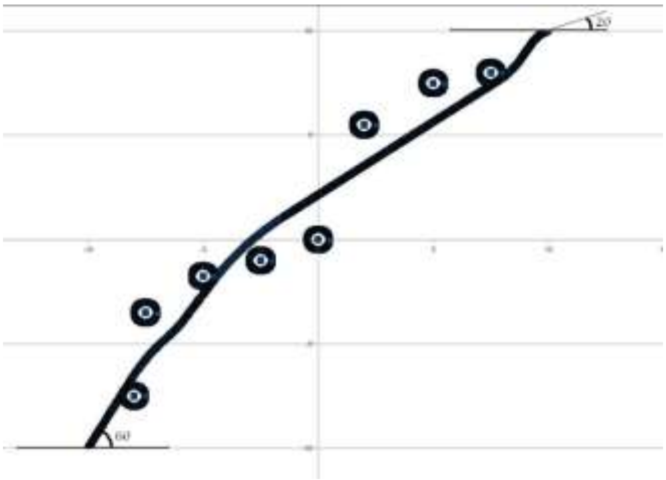


Fig 5. Graph connecting all Ps and Qs

Then we find the shortest way for getting from A to B by the edges and in order to do so we use the Dijkstra's algorithm then we name the points given by the Dijkstra's algorithm k_1, k_2, …, k_i then we give every k_i to k_{i+1} to the first layer which will then create the final path of a series of arcs and lines that we will then give them to the GTP function and as the path planning function is called every cycle we are always sure that we will never hit and obstacle.



3.4 Shooting

Shooting is a key skill for playing football. The important factors of a shoot are its direction and power.

In order to find out the direction of the shoot we need to find the slope of the line connecting the ball to our target (*S* line) by using this formula $(y_{target} - y_{ball}) / (x_{target} - x_{ball})$ which is the direction the foot will be moving at.

The shoot is composed of three phases in the first phase we find the B point which is z centimeters behind the ball and also located on the *S* line as well.

In the second phase we create an X-Y trajectory for the foot to get to the B point by sending both foot's position and the B point to the Path Planning function.

In the third phase we find the C point which is also z centimeters away from the ball but in front of it and also located on the *s* line.

After getting to the B point we move the foot to the C point with maximum power so that we make sure it moves in the right direction.

As we haven't worked on the power aspect of shooting we move the foot from the B point to the C point with maximum power.

4. Conclusion and Future plans

Humanoid soccer simulation is still a very new project which has a lot of room for improvement. Being a student team we have the advantage of starting our researches at a younger age therefore we have more time to perfect it. We are going to make our walk more stable so we are going to consider more factors for our walk like ZMP, COM and etc. we are also going to write plans for goalkeeper, defense, offense so they will be able to make decisions at any moment considering the situation. We are also going to work on shoot's precision and power as well.

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. 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
4. ShuujiKajita, Fumio Kanehiro, Kenji Kaneko,KazuhitoYokoiand Hirohisa Hirukawa, The 3D Linear Inverted Pendulum Model: A simple modeling for a biped walking pattern generation,Proceedings of the 2001 IEEE/RSJInternational Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001
5. Patrick MacAlpine, Daniel Urieli, ShivaramKalyanakrishnan and Peter Stone, UT Austin Villa 3D Simulation Soccer Team Description Paper 2011
6. ShahriarAsta, Tuna Sonmez, OnuralpUlusoy, AlperAlimoglu, Mustafa Ersen,OzdemirSozmen, and SanemSariel-Talay,beeStanbulRoboCup 3D Simulation LeagueTeam Description Paper 2011
7. Mahdi Alehdaghi, Hossein Ansari, Armin Ildermi, Sina Molavipour, Mohammad J. Shokri. Nexus3D Base Code
8. Xu Yuan, Tan Yingzi. Layered Ominidirectional Walking Controller for the Humanoid Soccer Robot.
9. Michael Angelo A. Pedrasa. Path Planning Algorithm for Soccer Playing Robots Based on Repeated Modification of Bezier Polynomials, PHILIPPINE ENGINEERING JOURNAL PEJ 2005; Vol. 26 No. 1:1–20