

Nexus3D Team Description Paper

Mahdi Abolfazli Esfahani, Mahdi Alehdaghi, Naser Erfani, Hamid Sajjadi, Soroush Farokhnia, Messiah Abolfazli, Reza Etemadi

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

nexus@um.ac.ir

Abstract. Nexus3D has participated in a varieties of soccer simulation competitions since 2003. These competitions are a platform in which we test and enhance our research in the field of AI and robotics and also compare our results and achievements to the state of art. In this paper we present a brief description for our current work on Nexus3D team. We have used inverse kinematics for walking and movements of robot and CMA-ES and genetic algorithm for our optimization.

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.

It has lots of features such as being cheap and easy-to-use which has made this platform more popular in recent years and also it has another powerful feature, it has three-dimensional environment which has made this platform a good resource and platform for studying humanoid robot control and strategic 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.

Currently, our main focus is on robot's stability control and movement together with team strategy and team positioning.

Our source code is implemented from the scratch and is extended with our latest base release. in this paper, first we explain our localization in section 2 and after that

we speak about our agents skills and behaviors in section 3 . finally we speak about decision making and discuss our new method for positioning in section 4.

2 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.

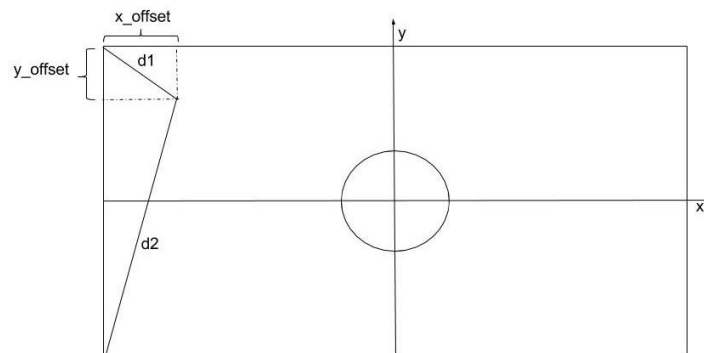


Fig. 1. Localization using two flags

3 Skills and Behaviors

3.1 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.

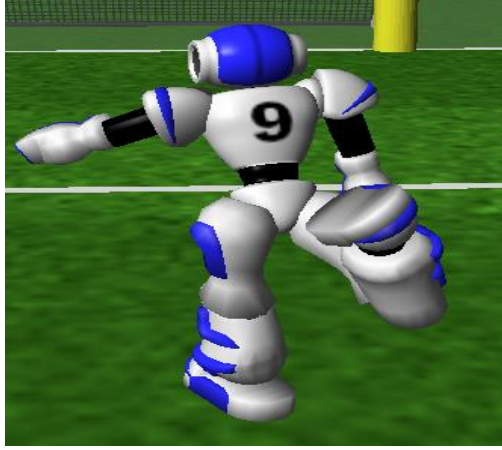


Fig. 2. Initial state for optimizing shoot

3.2 Walk

3.1.1 Walking System

In studies we have done in the past, we used to work on solving kinematic equations of the NAO robot to make it appropriate to move in a definite trajectory. In the studies we have done currently we used an algorithm named CMA-ES, which stands for CMA-ES stands for Covariance Matrix Adaptation Evolution Strategy, to realize the best optimum speed for each of the joints of NAO robot which makes it able to walk in a fast and stable way.

3.1.1.1 Zero of Moment (ZMP)

A point which the summation of the vertical forces that are going to the sole of the robot do not have any torque. Actually at this point the summation of all the forces is zero. We call the robot is dynamically balanced if the point is in the convex polygon between sole and field.

By using the set of equation (1) the ZMP position can be found by using set of equation (1) where m_i is the mass of each link; \ddot{x} , \ddot{y} and \ddot{z} are the acceleration along x, y and z axis; x_i , y_i and z_i are the position of each link. I_{iy} is the inertia tensor along y axis and variables $\ddot{\omega}_{iy}$ and $\ddot{\omega}_{ix}$ are the absolute angular acceleration angular along y and x.

$$\begin{cases} x_{zmp} = \frac{\sum_{i=1}^n m_i(\ddot{z}+g)x_i - \sum_{i=1}^n m_i\ddot{x}z_i - \sum_{i=1}^n I_{iy}\cdot\ddot{\omega}_{iy}}{\sum_{i=1}^n m_i(\ddot{z}+g)} \\ y_{zmp} = \frac{\sum_{i=1}^n m_i(\ddot{z}+g)y_i - \sum_{i=1}^n m_i\ddot{y}z_i - \sum_{i=1}^n I_{ix}\cdot\ddot{\omega}_{ix}}{\sum_{i=1}^n m_i(\ddot{z}+g)} \end{cases} \quad (1)$$

3.1.1.2 Center of Polygon (COP)

Calculating center of polygon would be very suitable and powerful which is center of for and this work will be done by using forces which go to the sole of the robot from ground.

If the robot is stable the center of forces is equal to the ZMP which can be calculated by using equation 2

$$COP = \frac{\int P F_n(x, y) dp}{\int F_n(x, y) dp} (2)$$

where P is each point of the sole and is the vertical force in that point. As we know, Nao robot has four pressure sensors in the sole. Thus, equation 2 would become as equation 3 where p_i is the location of the sensor and f_i is the value of force in that point

$$COP = \frac{\sum_{i=1}^4 p_i \cdot f_i(x_i, y_i)}{\sum_{i=1}^4 f_i(x_i, y_i)} (3)$$

3.1.1.3 Fitness Function

As we wanted a stable walking system to move correctly and do our jobs in a correct way, we have a function which has a significant and important role. to achieve this important goal ZMP and (COP) have to be very close to each other. So as we want to have a stable walk, the subtraction of these two values should very low to have a stable walk.

$$Fitness_1 = \sum_{i=0}^{simulation\ cycle} (Z_i - C_i) (4)$$

3.3 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.

3.3.1 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.

4 Decision Making

AS we have 11 players in a team it makes team strategies and decision important for handling different situations because this platform is multi agent and because this platform is multi agent, decision process becomes complicated. As you know we have lots of possible actions in order to determine the best action among all of these possible actions for a given situation, we use a method which has two phase for making decision. In this method, the first phase of this two phase method for making decision, for this

we have to search one of these search is for finding superlative actions in different classifications, and after finding these actions we do another search but in this time we want to find the best action between them. It looks complicated to compare different types of actions which belongs to different types of actions. but for comparing them in a correct way we compare their common parameters in them. In Fig.3 you can see whole the work in a diagram.

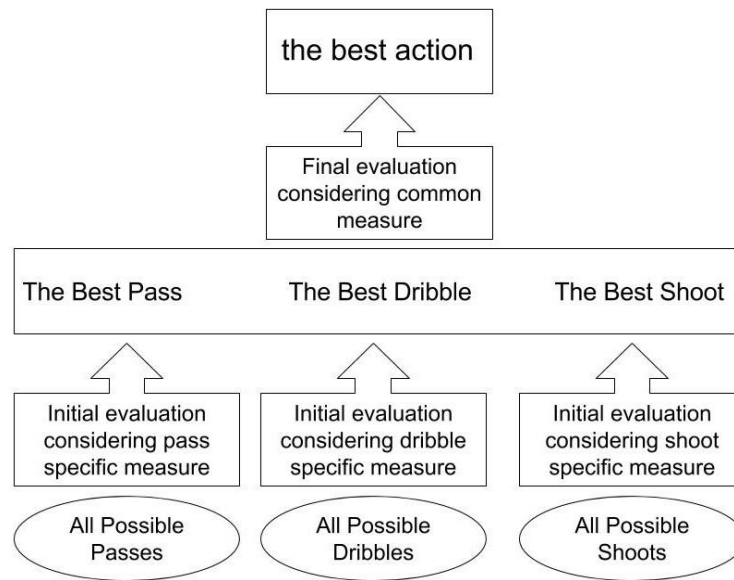


Fig. 4. two phase decision making mechanism

4.1 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. Therefore 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.

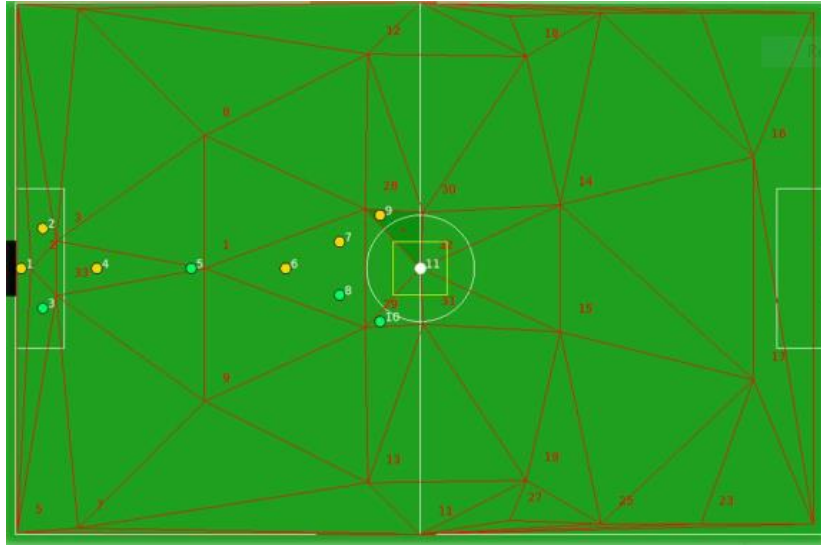


Fig. 5. Setting formation in FEdit

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

5 References

1. M. Alehdaghi, M. Abolfazli Esfahani, N. Erfani, S. Farokhnia, H. Ghayour, S. Askari, H. Sajjadi, "Nexus 2016 3D Soccer Simulation Team Description"
2. A. Ildermi, M. Alehdaghi, I. Kiarazm, M. Niazi, A. Gharib, M. Shokri, "Nexus 2011 3D Soccer Simulation Team Description"
3. M. Alehdaghi, A. Ildermi, et. al, "Nexus3D Base Code"
4. 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
5. Daniel Joseph Paluska, "Design of a Humanoid Biped for Walking Research", Master's thesis, 2000
6. Jerry Pratt and Gill Pratt, "Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation", Conference on Climbing and Walking Robots, 1999
7. Knight, R, and Nehmzow, U., "Walking Robots, A Survey and a Research Proposal", Technical Report
8. Krister Wolff and Peter Nordin. "An Evolutionary Based Approach for Control Programming of Humanoids"