# WrightEagle2008 3D Team Description Paper

Xue Feng, Tai Yunfang, Xie Jiongkun, Zhou Weimin, Ji Dinghuang, Zhang Zhiqiang, Xiaoping Chen

AI Center, Department of Computer Science,
University of Science and Technology of China, Hefei,China
{henry519, tyf, devilxjk, zwm, jidhuang, zqzh}@mail.ustc.edu.cn, xpchen@ustc.edu.cn
WWW Home Page: http://wrighteagle.org

**Abstract.** Since RoboCup 2006, we named WrightEagle3D have put the research priorities on humanoid robot and we have improved a lot last year. This paper describes the development and main features of our team in order for the participation of RoboCup2008 which will be held in Suzhou, China. The main emphasis is agent architecture, debug tools and robot skills.

## 1 Introduction

The 3D league takes a big step to the RoboCup's ultimate long-term goal in RoboCup2007. The big step is the simspark project [1] which is a generic simulator for physical multi-agent simulations. With simspark, we can experiment with different sensors, actuators or morphologies of agents and study team behavior with a set of given agents. Take the real humanoid robots as a comparison, simspark has the advantages in costs and efficiency. Now the agent uses the HOAP-2 based model [2] which has 4 DOF of each arm and 6 DOF of each leg.

WrightEagle3D was born in 2004 whffch is also the year that simulation3D league came into existence. We are frequent visitor of RoboCup3D league. We won 5th place in RoboCup2004, 7th place in RoboCup2005 and 2th place in RoboCup2006 when the agent model is sphere-based. And we won 1st place in RoboCup2007 which uses the humanoid-based model that has mentioned before.

In this article, section 2 is the description of the agent architecture. Section 3 is the description of agent skills such as walk, kick, get up, etc. Section 4 describes the development tools for the team binary. Section 5 briefly describes the decision that applied on the current 2vs2 match. Section 6 briefly describes the future work based on humanoid robot.

## 2 Architecture

Lots of tdps and articles mentioned this part. Our architecture is quite similar to that of 2D's which has a world model layer. Mainly, the whole agent Architecture consists of those parts:

1. Network Layer: for the communication with server.
2. World Model Layer: all the information needed for agent.
3. Decision Layer: like the human being's brain, for decision making.
4. Action Layer: currently, the agent's skills are composed of primitive actions. Those actions will be talked in the next section.
5. Path Planning: this is an important part for generating a series of actions when given a destination point.

As foundation of the whole architecture, world model maintains and keeps refreshing data that indicates agents' conditions, ball information, field information, game state, etc. Like the 2D's realization, when the network layer receives a new message that contains the new vision sense, perception sense, game state, etc, it has the responsibility of updating and refreshing world model and provide services to the upper layer.

## 3 Skills

The humanoid model have 20 degrees of freedom. So there comes lots of new problems for us to conquer. Such as the joint control, step pattern generation, etc.

In order to perform a soccer game, a humanoid robot must have the ability to perform some basic behaviors, even some advanced ones. Up till now, WrightEagle3D's robot has the ability to walk, turn, slide-move, getup and kick ball, etc. We have conducted a lot of studies and experiments on those behaviors, and have come up with some ideas on how to control every joint's move.

Now we are trying to discover some algorithms to help move the more precisely, for it is very important in biped locomotion. Our behavior architecture is base on state machine, here are the details:

### 3.1 State Machine

We use Turning's state machine as the controller of our agent's actions. It maintains a set of states, a set of executions, and a set of conditions. In each cycle, it transfer the state to a new state according to the conditions space. The set of executions managers what to do of each state.

### 3.2 Walk

The goal of walking behavior is that the robot can walk faster, more stably and more precisely. We have made progresses in these factors, especially in precision. In our walking procedure, the state of one foot is divided into three parts: Raise, Land and Support. And the whole walking behavior is divided into four states, see Figure 1.

And Figure 2 shows how the system works:

| Left Leg | Right Leg | State |
|---|---|---|
| Raise | Support | RS |
| Land | Support | LS |
| Support | Raise | SR |
| Support | Land | SL |

**Fig. 1.** Foot States

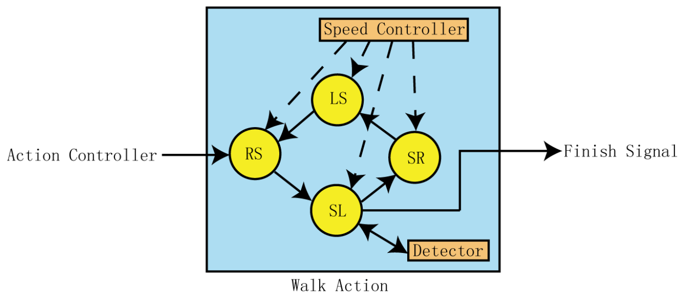


**Fig. 2.** Walk states transfer

Just as the graph has shown above, when the "walk command" was sent to the action Walk, Walk set the default state as RS, and then after state RS is finished, it will transmit to the next state, and so on. Walk will automatically adjust the speed of walking and after it detected that the robot is about to reach the destination, it will slow down and finally stop the Walk action. So the system can execute another action.

The 2D dimension control of the leg is simple and has been used in RoboCup2007. In the past, we tried lots of algorithms for the robot balance, such as zmp [3, 4], machine learning, etc. But nearly all of them was not as good as the expected. Then we used a system called "search and find parameter" which depends on the development tool "Command" that will be introduced at the "development tool" section.

### 3.3 Turn

The prevailing view to the turning action asserts that it's the first job to make the action as quickly and stable as possible. We agree this foundation also, but we want to add up to our own thinking. As is known to all, it's critical to get hold of every moment of the time, for the vertiginous situation in the game. So it's essential to change the action as agile as possible. For example, when a robot is turning, when the ball go to the nearby his foot, thus if he can make a quick change from turn to kick he can get another opportunity to get a goal.

There are some similarities between walk and turn, but they are really different matters, one calls for quick and stable, the other need agile and suitable instead. In the flounder to the algorithms and methods to turning action, we try to use the ZMP-stability [3, 4], neural-network modulating to make it works. Although it's a hard job, we will try our best to do research on it.

### 3.4 Slide-Move

Slide-Move is another important action. With this action, the robot can subtly adjust its position and direction, for preparing to kick the ball. Slide-Move action is quite similar to Walk action. But an accurate localization process is added, so the robot can go to the desired position and desired direction.

### 3.5 GetUp

It's difficult for a person to getup in a random situation, just the same in the simulation 3D. In the base of the previous research, we consider to make this goal as our destination. In RoboCup2007, almost all the teams in the world take the solution of making the robot get up after falling a preconceived state, like falling toward the chest or the back. But we want to get some break-ins from the rigid state.

By the tools of dynamics and the mechanics, we do some mathematical work on the robot, with the anticipation of controlling in a more exact method. However we think it's essential to make some basic actions through some body-weary job such as parameters modulating, for the consideration of the stability and safety. Altogether, it's a great challenge for us to make the robot could stand up similar to actual human beings, but it's also a chance to bring mankind to a higher altitude.

### 3.6 KickBall

Kick-ball is also an important action, with this action robot can shoot ball, pass ball, intercept ball, etc. At the beginning, our action of kicking was based on changing parameters manually to achieve better performance. In China Open 2007, we saw many actions of kick based on manually changing parameters which were beyond our thinking. After that we thought that we should not staying on changing parameters manually.

However when we tried to use physical model to form actions of kicking, we could not get good effect. Then we tried to use changing parameters manually to get good effect at first. When we kicked very well we studied the model in these actions. And then based on the model we tried to improve the actions. And we have got much better effect.

Now we think that we have the ability to intercept, but our aim is to control ball more exactly, like passing ball to teammate.

## 4 Development Tool

To make the research of simulation 3D more convenient, we have developed three tools: Commander, LogPlayer, TCServer. These tools is mainly for windows, because we prefer to using Visual Studio 2005 to write code. Also we don't have many good enough computers for everyone to run server, these tools can make up for it.

### 4.1 Commander

The tool of Commander was developed before RoboCup 2007. It was the first tool that our team developed for Simulation 3D with humanoid model.

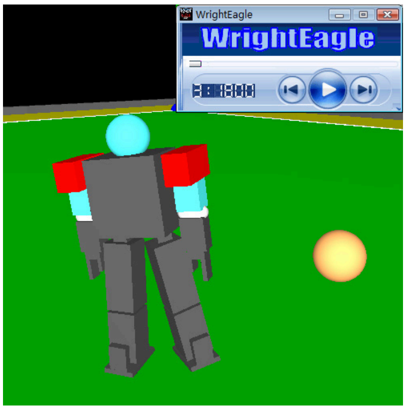


**Fig. 3.** Commander Screen Shot

It has three features, Figure3 is a screen shot of Commander:

1. It has a physics kernel implemented by ODE [7] just like rcssserver0.5.5 so that we can apply some algorithm on it, such as search, learning, etc.
2. It provides a GUI interface so that we can manually set the angular velocity of each joint, and click a button to simulate a cycle. It is just for the beginners to know the performance of each joint.
3. It is a substitution of sparkmonitor of server0.5.5 on windows.

However, as the server changes, the tool of Commander can not correctly simulate the robot's action, and the log player can not display the new server's log file, too.

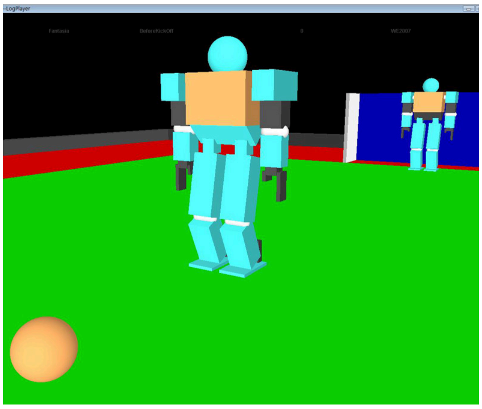

**Fig. 4.** LogPlayer Screen Shot

### 4.2 LogPlayer

Before China Open 2007 in Oct. 2007, we developed a new log player which could display the log file of the new server. Figure 4 is a screen shot of LogPlayer.

Not like the log player of the Commander, this log player is more like the monitor spark of the server. It can display the team names, the score and the time.

For it is designed for windows ,we needn't change to linux to display the log file, and also everyone in our team could display the log file in their own computer.

### 4.3 TCServer

Our team wants to introduce simulation 3D to more students in our school, and let more students who was interested in it take part in our research. So we developed the tool: TCServer. Figure 5 is a screen shot of TCServer.

The TCServer is mainly designed for windows. It includes two parts:

1. The server part. We ignore many parts of the server, just reserve the physics part. Comparing to the physics kernel of Commander, it is more accuracy and more rcssserver0.5.6-like.
2. The agent module. We use a dll solution instead of network solution for avoiding the delay of the network.

## 5 Decisions

The decision part is the soul of a team. It gets the information from the world model, and decides what to do with the current situation. It just works like a

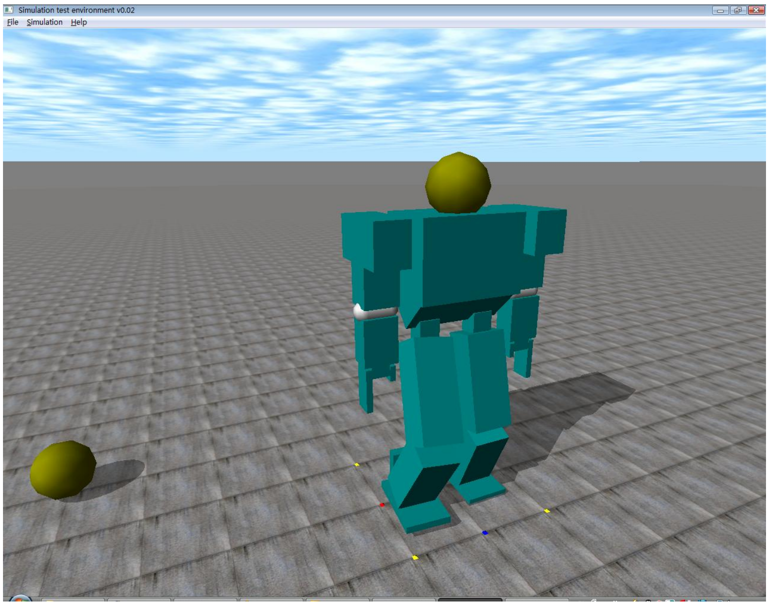


**Fig. 5.** TCServer Screen Shot

person's brain. This year, we have a great progress in our action part, the player can walk faster , get up quicker , and kick the ball more precisely . Based on that, we do a lot changes to make the decision more efficient

We have two kinds of play mode; one mode is that we have a midfield and a goalkeeper, the other is one midfield and one defense on the field. According to our test, the later mode is more efficient, the midfield and the defense can both attack and defend, depending on where their enemies and the ball are and their pose. When a player work as a defense, he will walk to the ball, and kick the ball away or to a palace where is advanced for his teammate to get the ball, if it is hard for him to kick the ball away, he will just walk to a position where he can block off the way his enemy shoot. When he gets the ball, he will choose to shoot, or kick to somewhere his teammate can get the ball, once his teammate gets the ball, will also decide to shoot or pass the ball. This kind of low pass between two players can be a very good teamwork, which can be used to get rid of enemies' defense or get to a better place for shooting. We also use three players in one team to test our decision for teamwork, the more player there is, the more complicated it is. Along with the progress of the agents control, we will do better teamwork, and we are still working on the cooperation of multiplayers.

## 6 Conclusion and Future Work

In this paper, we described several aspects of our team. We gave an architecture of the agent, the humanoid-robot skills, the development tool for the team and the decisions for the 2vs2 match.

Lots of goals should be achieved in the future. First, the controlling of joints. There are lots of research and studies in the field of Robotics [6]: The forward kinematics, reverse kinematics, dynamics, etc. But in simulation, it is hard to control in velocity aspect, not to mention the real robots because of the noise. The current server setting is not add the noise. But the precision of sense sent by the server is restricted to 2 after the radix point. So after computation use the robotics, the precision of foot position is about 2cm. And it affects the velocity computation. However, we have solved the 2D leg controlling which has been used in walk in RoboCup2007 and resulted in a good walk.

Second, step pattern generation. Currently, there are two models of walking: ZMP-based model and Inverse Pendulum model [3, **?**,**?**]. But the lack of sensors results in the difficulty of step pattern generation. There are two force resistance perceptors on each foot. But it seems that there is a bug in its realiztion. Walking in 3D dimension is much harder than walking in 2D dimension. We see lots of stable 2D walking in the match. But no team can walk obliquely. This is a very challenging work.

Third, path planning. How to make the robot walk through obstacles such as enemies, goal and etc. However, the sum players on the field is only 4 currently, so with search algorithm, it is easy to keep away from enemies. But when there are more players, the computation costs more expensive and 20ms is not enough. We need discover a new algorithm to solve this problem.

## References


1. Oliver Obst and Markus Rollmann: Spark, A Generic Simulator for Physical Multi-agent Simulations, 2005
2. RoboCup 3D Soccer Simulation League: HOPE-1 Manual
3. Miomir Vukobratovi6, B.Borovac, DSurla and D.Stoki6: Biped Locomotion: Dynamics, Stability, Control and Application, 1990
4. M.Vukohratovii. and J.Stepanenko: On the Stahility of Anthropomorphic Systems, 1972
5. Sugihara, T., Nakamura, Y. and Inoue: Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control, 2002
6. John J.Craig: Introduction to Robotics Mechanics and Control, 2002
7. Russell Smith: OPEN DYNAMICS ENGILE USER GUIDE, 2004
8. Chen Xiaoping, Liu Jinsu, Xue Feng, Lai Xinghua: WrightEagle2007 3D Team Description Paper, 2007