

# HfutEngine3D Soccer Simulation Team Description Paper 2014

Qingyuan Zhang, Chaowei Dong, Weixin Xie, Gang Sun, Jinlei Pan

School of Computer and Information  
Hefei University of Technology, China

**Abstract.** This paper simply describes the architecture of HfutEngine3D team. In order to control a biped robot with high degree of freedom to get faster and more stable, we divide our team into four parts: InformationHandle, MotionHandle, WorldmodelHandle, StrategyHandle. Additionally, it introduces what we use in Matlab to simulate robot motion to get ZMP and CoM, which proves the stability of robot's action, and the implementation of 3D linear inverted pendulum.

## 1 Introduction

In RoboCup China 2005, we came into RoboCup 3D Simulation League for the first time. Early 3D league was sphere form and we focused on the accuracy of calculation of three-dimensional physical virtual circumstance. At the same time, research of the Middleware SPADES (System for Parallel Agent Discrete Agent Simulation) was also very important. We get 10th place in RoboCup China 2005, 12th place in 2006. In 2007s Valentine's Day, new version of server was released, which included new Fujitsu HOAP-2 simulation robot instead of old sphere robot. The new server brought many changes as well as new challenges such as, Joint Control, State Detect and etc.. After months hardwork, our team featured some new controlling ideas and its humanoid motion worked very well. HfutEngine3D got 7th in RoboCup China Open 2007 in Oct.. We also got 3rd in RoboCup Iran Open 2008 and 5th in RoboCup China Open 2008. In July 2008, we attended in RoboCup 2008 in Suzhou. It was the first time for us attended in RoboCup. We have advanced into the top 16. Later we got the 4th place in 2009, and the 3th place in 2010 in RoboCup.

This paper introduces HfutEngine3Ds features and implementation. Section 2 briefly describes the teams main modules. Section 3 introduces some of our team's characteristic. Section 4 tries to show our experiment in Matlab about ZMP and CoM. The last section is about our plan of future work.

## 2 Team Architecture

According to Peter Stones layer learning method, we design four learning modules for the team. They are InformationHandle, MotionHandle, WorldmodelHandle and StrategyHandle.

The InformationHandle takes charge of communication with server, it includes network controlling, message parsing and command queue building. WorldmodelHandle contains several states updating and some calculating of motions key parameters like whether robot is falling down. MotionHandle is designed to control Joints to finish quite complex task. StrategyHandle is the brain of robot which is based on non-goalie idea and dynamic assignment.

As 3D server is working in C/S mode, we get message from InformationHandle module firstly. After parsing the message, WorldmodelHandle module updates all of the states including joints state, game state and object state(ball, itself, teammates, opponents and etc.). StrategyHandle module analyzes current situation and then chooses one tactic with the best benefit. To achieve the strategy, the robot should also make a series of joints commands to perform motions finished by MotionHandle module, joints commands will be put into the command queue. At last, InformationHandle module gets command from command queue. Fig.1. describes HfutEngine3D's running flow. Generally, the running flow is based on sense-think-act cycle.

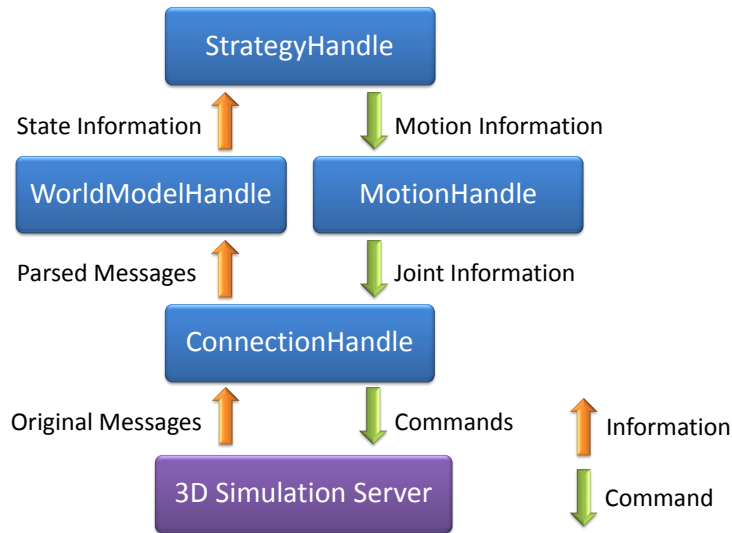


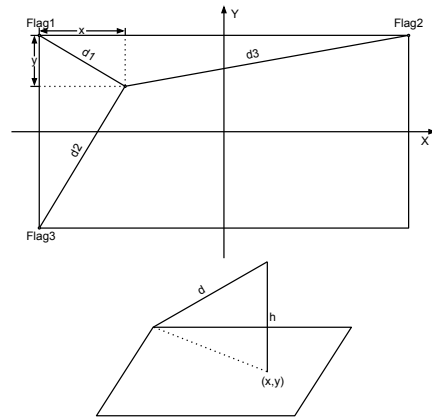
Fig. 1. HfutEngine3D main running flow in about one cycle

### 3 Team Characteristic

#### 3.1 Self-Localization

We only need three position vectors about the relationship between flags and our robot to calculate its self-position. The robots x-coordinate will be calculated

by lengthways and two flags, while the y-coordinate will be given by transverse two flags (It is known that any three flags of field s arrange have two ones in lengthways and two ones in transverse, so we choose the closest three flags generally).



**Fig. 2.** Self-Location and Calculating the Hight

In Fig.2, parameters  $d_1$ ,  $d_2$  and  $d_3$  denote the distance between robot and Flag1, Flag2 and Flag3, which are included by vision information. We can draw a triangle with field width,  $d_1$  and  $d_2$  in level. With the triangle, we can calculate the y-coordinate. Moreover, we get the x-coordinate by another triangle with field length,  $d_1$ ,  $d_3$ . At last, we use the x-coordinate and the y-coordinate to get robots height. It can also be obtained by forward kinematics when the robot is standing.

### 3.2 Kinematics

The detailed parameters of the robot are quite important to the robot's development. With these parameters we can construct forward kinematics and inverse kinematics. In our team, forward kinematics is used to get the position of joints, even the height of robot. While inverse kinematics is often used to calculate the joint angle after walking gait planning.

### 3.3 3D Linear Inverted Pendulum

In the latest version of HfutEngine, we use 3D linear inverted pendulum to plan the Walking pattern. It can make robots walking faster and more stable without falling down too often. The following equation is the key to the planning of walking pattern:

$$\ddot{x} = \frac{g}{z_c}x \quad (1)$$

The equation of y-coordinate is the same as the one below. We can use this equation to calculate the x-coordinate and y-coordinate of the torso, moving foot and holding foot as the time changes. Then, use inverse kinematics functions to get the angle of each joint, and then apply these angles that have calculated, the robot can be able to walk more perfectly.

### 3.4 Shift Velocity Walking

Sometimes, our robots may directly walk to a random point. What we need to do is just controlling the accelerating and decelerating work period. The following is the characteristics of walking what we want our robot to make:

1. Dynamically calculate the acceleration based on the distance to target and angle.
2. Detect the change of target, dynamically switch different action.
3. Predict possible stance in all walking period including walking distance, velocity and the slowdown point.

To smoothing the shift process, we construct a logarithm function which is based on distance.

$$a = 1.67607 \lg (DistanceToTarget) \quad (2)$$

When calculating the slowdown point, we need to predict the max velocity the robot can reach. After analyzing from experimental data, we use a quadratic function to predict the max velocity. Here are three groups of data:  $(d_1, V_{m1})$ ,  $(d_2, V_{m2})$ ,  $(d_3, V_{m3})$ .

$$V_m = V_{m1} + \frac{V_{m2} + V_{m1}}{d_2 - d_1}(d - d_1) + \frac{\frac{V_{m3} - V_{m2}}{d_3 - d_2} - \frac{V_{m2} - V_{m1}}{d_2 - d_1}}{d_2 - d_1}(d - d_1)(d - d_2) \quad (3)$$

### 3.5 Toolkits Development

We develop a trainer. In order to operate it, we modify the source code of rcssserver3D first. The tool is connected to the simulation server through a TCP socket. By using the tool we can not only get the position of robots and ball, but also draw each angle of joints of robots, move robots or ball to a new position just by click the mouse, change the playmode of game, save the joints information of the robot to a logfile and replay it. It is very useful for us to train the skills of the robot. Fig.3 shows the trainer tool. It was developed by VS2008. It can be executed in Windows or in Linux just by the wine tool.

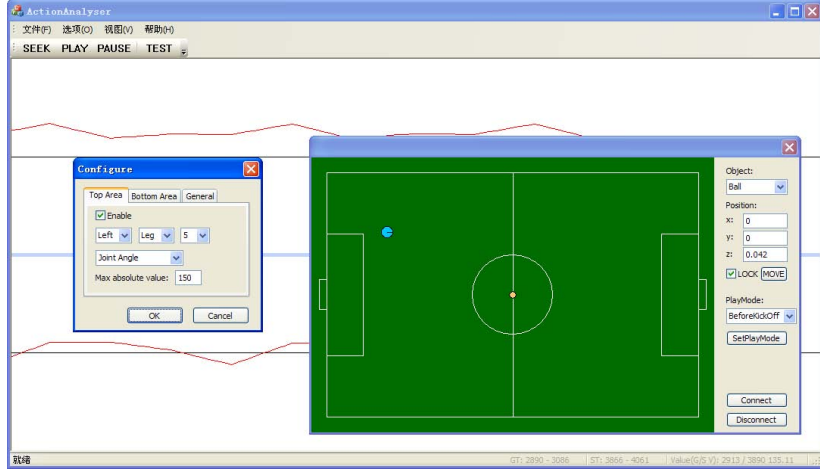


Fig. 3. HfutEngine3D trainer

## 4 Experiments in Matlab

In the current 3D Simulation League, what we focus on is the humanoid biped robot motion control and the vision procession. ZMP(Zero-Moment Point) and CoM(Center of Mass) are two very important parameters in detecting stability of motion. Here we simplify HfutEngine3Ds up-body into cube, it will become easier to calculate ZMP and CoM.

ZMP and CoM can be calculated by

$$X_{ZMP} = \frac{\sum_{i=1}^n m_i(z_i + g)x_i - \sum_{i=1}^n m_i x_i z_i - \sum_{i=1}^n I_{iy} \Omega_{iy}}{\sum_{i=1}^n m_i(z_i + g)} \quad (4)$$

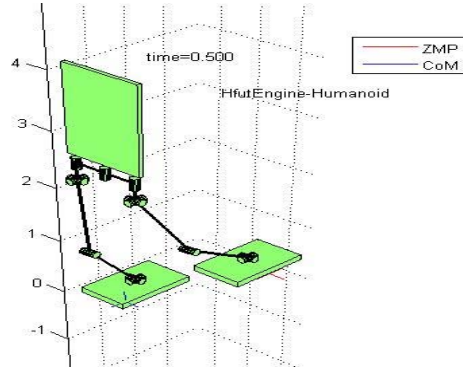
$$Y_{ZMP} = \frac{\sum_{i=1}^n m_i(z_i + g)y_i - \sum_{i=1}^n m_i y_i z_i - \sum_{i=1}^n I_{ix} \Omega_{ix}}{\sum_{i=1}^n m_i(z_i + g)} \quad (5)$$

$$O_Z = \frac{\sum_{i=1}^n O_i m_i}{\sum_{i=1}^n m_i} \quad (6)$$

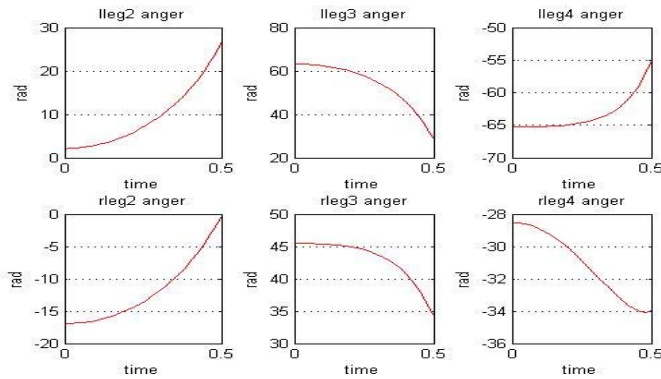
In MatLab, we construct a biped robot with the help of Shuuji Kajita<sup>1</sup>. Fig.4. simulates the motion of the robot and calculates the ZMP and the CoM. Fig.5.

<sup>1</sup> Shuuji Kajita, Dr, Eng, Humanoid research group, Intelligent Systems Research Institute National Institute of Advance Industrial Science and Technology (AIST), METI

logs the six joints of leg which are changing, and Fig.6 show the changing of ZMP and CoM.



**Fig. 4.** Robots motion with ZMP and CoM calculation



**Fig. 5.** The six joints of legs angles

After our painstaking debugging, the robots can walk, stand up, turn or shoot very well.

## 5 Future Works

Based on teammates hardwork, we believe HfutEngine3D will have a bright future. Currently, the most important work for us is to keep developing our tools for the robot debug and design a good algorithm for planning the cooperation of robots. We still have a long-range goal to achieve.

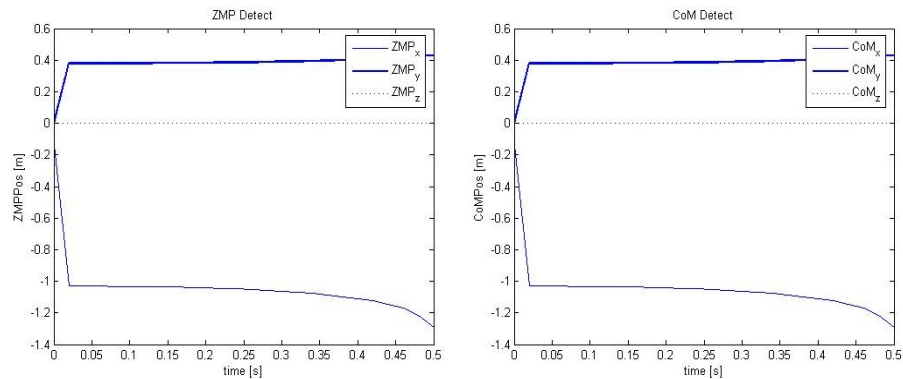


Fig. 6. Calculation of the ZMP and the CoM

## References

1. Kajita, S. and Tani, K.. Experimental study of biped dynamic walking. In *IEEE Control Systems*, Vol.16, No.1, pp. 13-19, February 1996.
2. Kajita, S.. *Humanoid Robots*. Tsinghua University Press, 2007.
3. R.Tedrake, T.W.Zhang, H.S.Seung. Stochastic Policy Gradient Reinforcement Learning on a Simple 3D Biped. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2004)*, pp. 2849-2854, 2004.
4. Guestrin, C., Venkataraman, S., Koller, D.. Context-specific multi-agent coordination and planning with factored MDPs. In Proc. 8th Nation. Conf. on *Artificial Intelligence*, Edmonton, Canada (2002).
5. Kok, J.R., Spaan, M.T.J., Vlassis, N.. Non-communicative multi-robot coordination in dynamic environments. In *Robotics and Autonomous Systems 50 (2005)*, pp. 99, 114.
6. Kajita, S. ; Kanehiro, F. ; Kaneko, K. ; Yokoi, K. ; Hirukawa, H. The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, pp. 239 .
7. Zhulin An, Jiangjiang Yu, Hao Wang. RoboCup Simulation League Goalie Design. In *Proceedings of 1st Austria Open of RoboCup*, 2003.
8. Baofu Fang, Hao Wang, Jia Liu, Chenwen Su. Team Strategy of HfutAgent. In *Proceedings of 2003 Master China RoboCup*, Aug, 2003.
9. Cheng Wang, Hao Wang, Baofu Fang. Multi-agents Action Selection Using Coordination Graph Based on Value Rule. In *Computer Engineering and Applications*, 2004(19).
10. Lei Yu, Hao Wang, Cheng Wang. Studies on Strategy of Pass in RoboCup. In *Computer Engineering and Applications*, 2004(8), Aug, 2004.
11. Baofu Fang, Hao Wang, Hongliang Yao, Jin Yang, Jin Zhou. The Application of Q Learning In RoboCup. In *Proceedings of 2004 China RoboCup*, Oct, 2004.
12. Runmei Zhang, Hao Wang, Honghang Yao, Baofu Fang. Influence Diagrams and Its Application in Robocup. In *Acta Simulata Systematica Sinica*, 2005(1), 2005.
13. Baofu Fang, Hao Wang. The Survey of HfutEngine2005 Robot Simulation Soccer Team Design. In *Journal of Hefei University of Technology*, Vol.29(9), Sep 2006.

14. Jianqing Gao, Hao Wang, Lei Yu. A Fuzzy Reinforcement Learning Algorithm and Its Application in RoboCup Environment. In *Computer Engineering and Applications*, 2006(6).
15. Yang Liu, Hao Wang, Baofu Fang, Hongliang Yao. Application of the method of support vector regression in RoboCup. In *Journal of Hefei University of Technology*, Vol.30(10), Oct 2007.
16. Baofu Fang, Bingrong Hong, Hao Wang, Dunqiao Bao, Long Li. A Muti-Agent Defensive Strategy Based on Monte Carlo Method. In *Journal of Harbin University of Technology*, 39(S1), Jun 2007.
17. Hao Wang, Yang Liu, Baofu Fang. The Research of Reinforcement Learning Technical Based on Support Vector Machine Classification and Its Application in RoboCup. In *Journal of Harbin University of Technology*. 39(S1), Jun 2007.