HfutEngine3D Soccer Simulation Team Description Paper 2010

Dong Li, Hao Wang, Baofu Fang, Guoqiang Shi

School of Computer and Engineering
Hefei University of Technology
Hefei 230009, China
sgqxc@hotmail.com

Abstract. This paper simply describes the architecture of HfutEngine3D team. In order to control a biped robot with a high degree of freedom to get faster and more stable movement, we design our team in four parts: InformationHandle, MotionHandle, WorldmodelHandle, StrategyHandle. Additionally, it will introduce what we are using in Matlab to simulate Agent motion to get ZMP and CoM, which prove the stability of Agent's action.

Key words: architecture, biped robot, kinematics, ZMP, CoM

1. Introduction

In RoboCup China 2005, we came into RoboCup 3D Simulation League for the firsr time. Early 3D league was sphere form and we focused on the calculation's accuracy of three-dimensional physics virtual circumstance. At the same time, research of the Middleware SPADES (System for Parallel Agent Discrete Agent Simulation) was also very important then. We get 10th place in RoboCup China 2005, 12th place in 2006. In 2007's 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 etc. After months' hard work, our team featured some new controlling ideas and its humanoid motion worked very well. In Oct. HfutEngine3D got 7th in RoboCup China Open 2007. 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 to attend in RoboCup. We have advanced into the top 16.

This paper introduces HfutEngine3D's features and implementation of our team. Section 2 briefly describes team's main modules. Section 3 introduces our some of our team characteristic. Section 4 tries to show our experiment in Matlab about ZMP and CoM. The last section is our future work's planning.

2 Team Architecture

According to Peter Stone's Layer Learning method, we've designed four learning modules for the team. They are Information-Handle, Motion-Handle, Worldmodel-Handle and Strategy-Handle.

The Information-Handle takes charge of communication with server. It includes network controlling, message parsing and command queue building. Worldmodel-Handle contains several states updating and some calculating of motions' key parameters like whether robot is falling down, etc. Motion-Handle is designed to control Joints to finish fairly complex movement. Strategy-Handle is the brain of robot. Our Strategy-Handle is based on 'non-goalie' idea and dynamic role assignment.

As 3D server working in C/S mode, we firstly take Information-Handle module to get message server. After parsing the message, Worldmodel-Handle module updates all of the states including joints state, game state, object state (ball, itself, teammate, opponent etc). Strategy-Handle module decides current situation and then chooses one tactics with the best benefit. To achieve the strategy, Agent should also make a series of joints commands to perform motions finished by Motion-Handle module. Joints commands will be put into the command queue. At last, Information-Handle module gets command from command queue. Fig.1. describe HfutEngine3D running flow. Generally, our running flow is based on sense-think-act cycle in Fig.2.

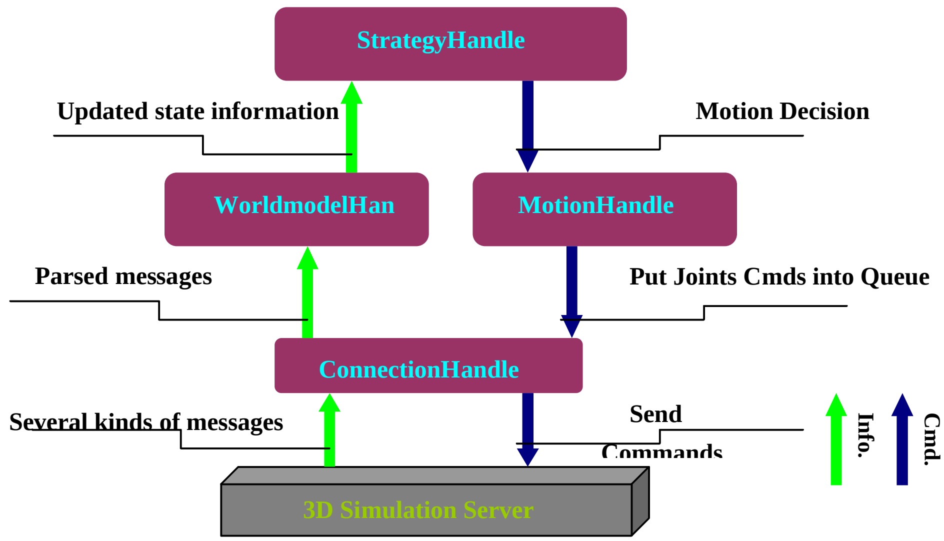

Fig.1. HfutEngine3D main running flow in about one cycle

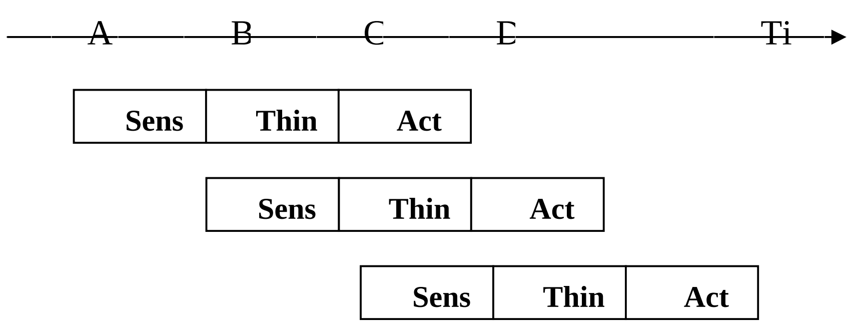

Fig.2. Sense-Think-Act cycle

3 Team Characteristic

3.1 Self-Localization

We only need three position vectors about the relationship between flags and our robot to calculate our self-localization. In our method, robot's x-coordinate will be calculated by lengthways, two flags and y-coordinate will be given by transversing 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).

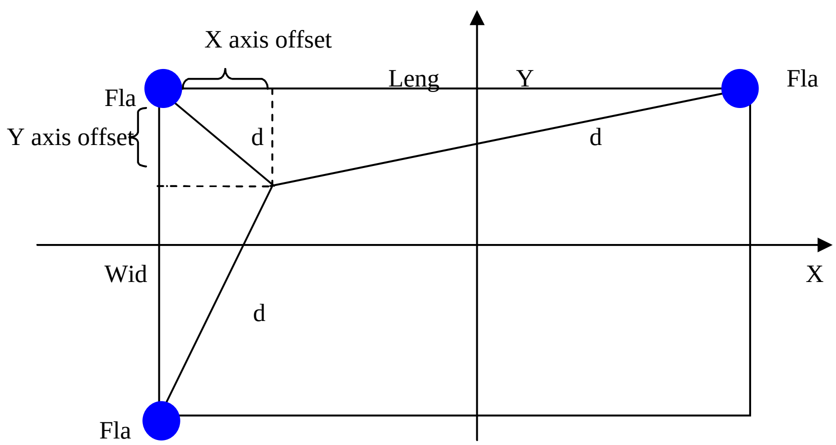

Fig.3. Self-Location

In fig.1, parameters d1, d2 and d3 denote the distance between robot and flag1, flag2 and flag3, which are included by vision information. We can draw a triangle with field width, d1 and d2 in level. With the triangle, we can calculate the y-coordinate. Moreover, we get the x-coordinate by another triangle with field length, d1, d3. At last, we use x-coordinate and y-coordinate to get robot's height which is shown in Fig.4. We can also get it by using forward kinematics when the robot is standing.

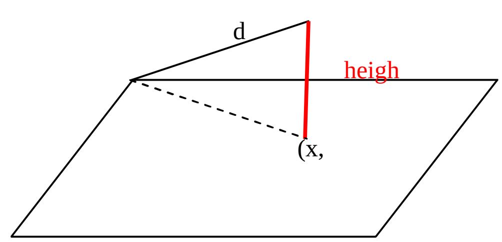

Fig.4. Calculate Height

3.2 Kinematics

The detailed parameters of the robot are quite important to the agent 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. Inverse kinematics is often used after walking gait planning to calculate the joint angle.

3.3 Shift Velocity Walking

In field, our robot will walk directly to a random point. What we need to do is just to control the accelerated and decelerate work period. The following is the characteristics of walking we want our robot to make:

(1) Dynamic calculation the acceleration based on the distance to target and angle.

(2) Detect the change of target, dynamic switch different action.

(3) Predict possible stance in all walking period including walking distance, velocity and the slowdown point.

To smoothing the period of shift process, we construct a logarithm function which takes distance as parameters.

$$a = 1.67607 \times \log_{10}\left(DistanceToTarget\right) \quad . \tag{1}$$

When calculating the slowdown point, we need to predict our max velocity our robot can reach. After analyzing from experimental data, we use a quadratic function to predict our max velocity. Here we use three groups of data: (d1, Vm1), (d2, Vm2), (d3, Vm3): d->Vm.

$$V_m = V_{m1} + \frac{\left(V_{m2} - V_{m1}\right)}{d_2 - d_1} \times \left(d - d_1\right) + \frac{\left(\left(V_{m3} - V_{m2}\right)/_{d_3 - d_2} - \left(V_{m2} - V_{m1}\right)/_{d_2 - d_1}\right)}{\left(d_2 - d_1\right)} \times \left(d - d_1\right) \times \left(d - d_2\right) \quad . \tag{2}$$

3.4 Toolkits Development

For the convenience of robot’s motion design, we develop our action toolkits. The envisage-glance and side-glance of robot based on ‘soccerbot065.rsg’ are given by toolkits, therefore we can directly design its posture. After finishing the design, the parameters will be converted into parameter table which our robot can use in on-line match. The analyzing part is used to give preliminary suggestion of motion designing just like which period the robot’ performing becomes abnormal etc. Fig.5 shows our toolkits.

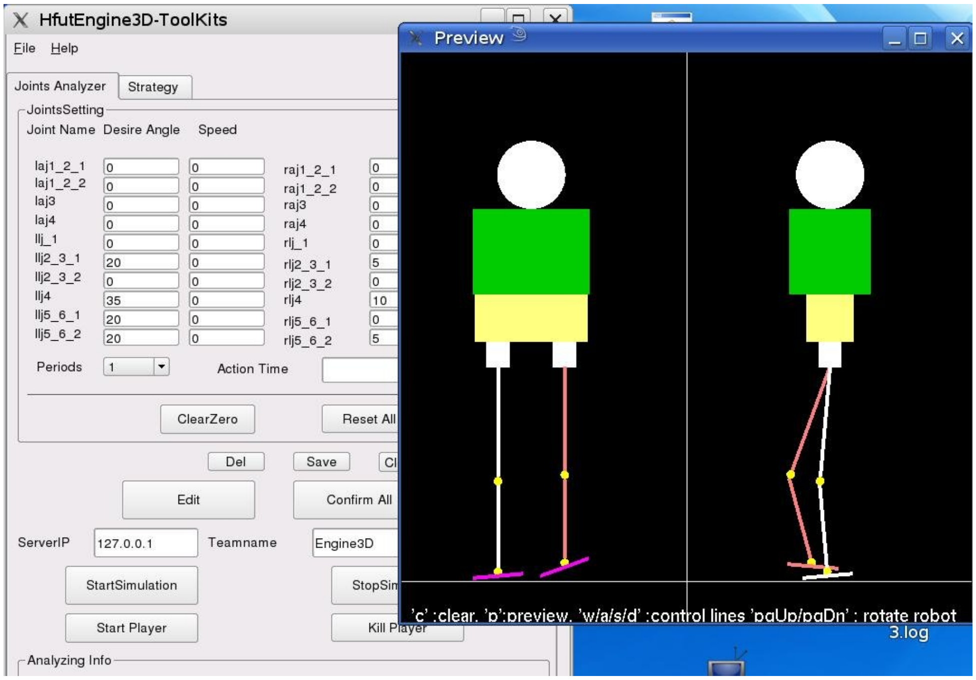

Fig.5. HfutEngine3D toolkits

We also develop a trainer. To use it we should modify the source code of rcssserver3D first. The tool connects to the simulation server through a TCP socket. By using the tool we can get the position of robots and ball, draw each angle of joints of robots, move robots and 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, and other functions. It is very useful for us to train the skills of our robot. Fig.6 shows the trainer tool. It was developed by VS2008. It can be executed in Windows and in Linux just by the wine tool.

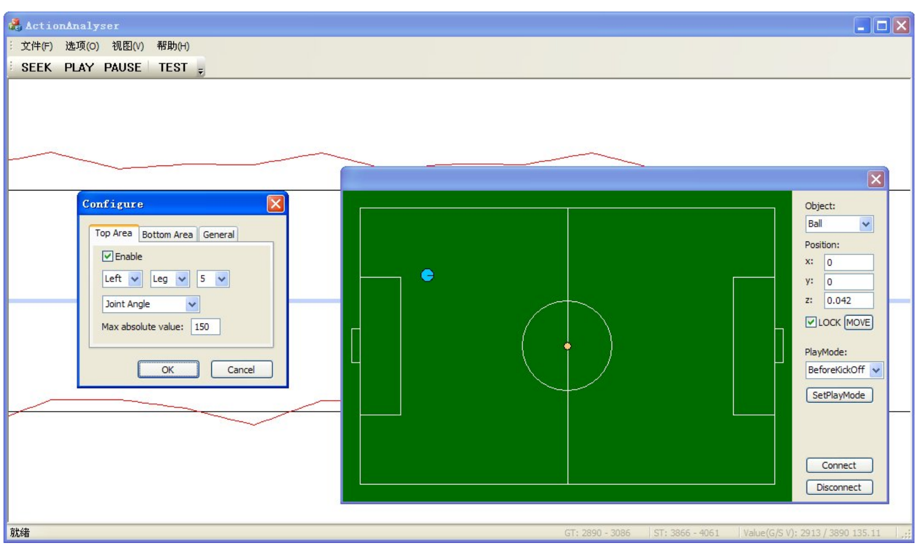

Fig.6. HfutEngine3D trainer

4 Experiments in Matlab

In the current 3D Simulation League, what we focus on is humanoid biped robot motion control and vision procession. ZMP (Zero-Moment Point) and CoM (Center of Mass) are two very important parameters in detecting stability of motion. Here we simplify HfutEngine3D's up-body into cube, it become easier to calculate ZMP and CoM. ZMP and CoM can be calculated by

$$X_{ZMP}=\frac{\sum_{i=1}^{n}m_i\left(Z_i''+g\right)x_i-\sum_{i=1}^{n}m_i x_i'' z_i-\sum_{i=1}^{n}I_{iy}\Omega_{iy}''}{\sum_{i=1}^{n}m_i\left(z_i''+g\right)} \tag{3}$$

$$y_{ZMP}=\frac{\sum_{i=1}^{n}m_i\left(z_i''+g\right)y_i-\sum_{i=1}^{n}m_i y_i'' z_i-\sum_{i=1}^{n}I_{ix}\Omega_{ix}''}{\sum_{i=1}^{n}m_i\left(z_i''+g\right)} \tag{4}$$

$$O_z=\left(\sum_i O_i\times m_i\right)\Big/\sum_i m_i \tag{5}$$

In MatLab, we construct a biped robot with the help of Shuuji Kajita[1]. Fig.7. simulates robot's motion and calculates the ZMP CoM.Fig.8. Logs six joints of leg are changing. And Fig.9 & Fig.10 show the ZMP and CoM's changing.

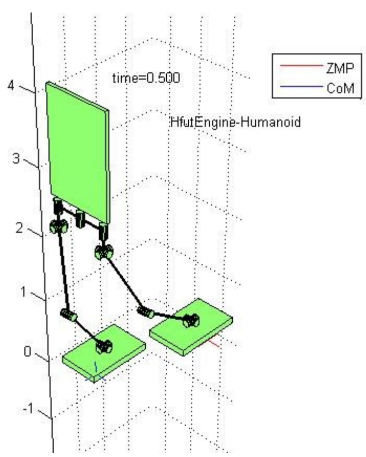

Fig.7.robot's motion with ZMP and CoM calculating

[1] Shuuji KAJITA, Dr, Eng, Humanoid research group, Intelligent Systems Research Institute National Institute of Advance Industrial Science and Technology (AIST), METI

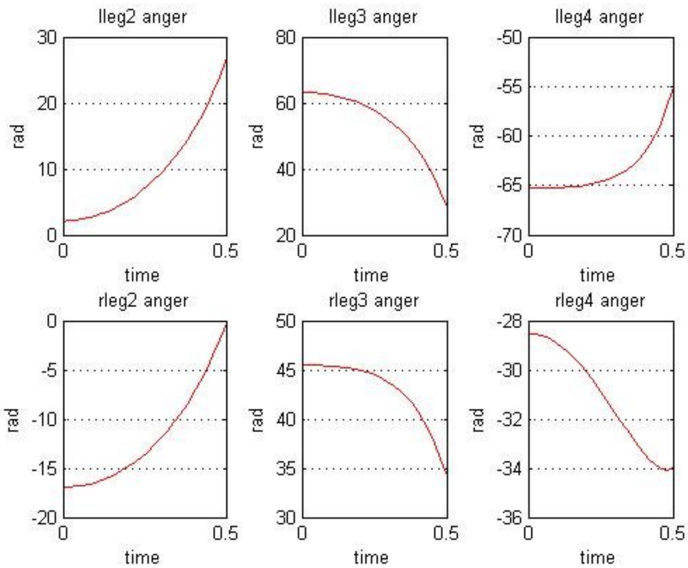

Fig.8.six joints of leg's angles

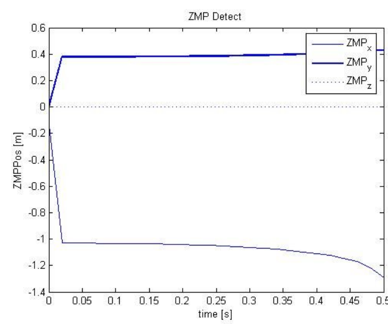

Fig.9.ZMP computation

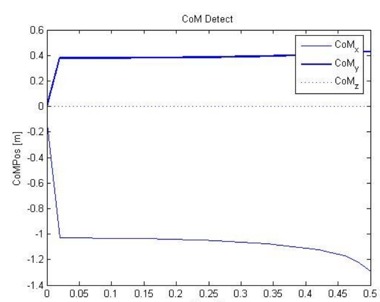

Fig.10. CoM computation

After painstaking debugging, HfutEngine3D performs walking, standing up, turning and shooting very well.

5 Future Works

Based on teammates' hard work, 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 work planning algorithm for robots' cooperation. Restricted vision perceptor will be used in the future. Much of time should be spent in it. We still have a long-range object to realize.

6 References

1. Kajita, S. and Tani, K., ``Experimental study of biped dynamic walking,'' 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," 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 multiagent 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. Robotics and Autonomous Systems 50 (2005)，pp. 99, 114

6 AN Zhu-lin, YU Jing-jing, WANG Hao. Robocup Simulation League Goalie Design. Proceedings of 1st Austria Open of RoboCup. 2003.

7. Fang Bao-fu Wang Hao Liu Jia, and Su Chen-wen. Team Strategy of HfutAgent. Proceedings of 2003 Master China RoboCup. Aug, 2003.

8. Wang Cheng, Wang Hao, Fang Bao-fu. Multi-agents' Action Selection Using Coordination Graph Based on Value Rule, Computer Engineering and Applications.2004(19).

9 Yu Lei, Wang Hao, Wang Cheng, Studies on Strategy of Pass in RoboCup, Computer Engineering and Applications, 2004(8), Aug,2004.

10 Fang Bao-fu, Wang Hao, Yao Hong-Liang, Yang Jin and Zhou Jin. The Application of Q Learning In RoboCup. Proceedings of 2004 LeeXeeun China RoboCup.Oct, 2004

11. Zhang Run-mei, Wang Hao, Yao Hong-Hang, Fang Bao-fu. Influence Diagrams and Its Application in Robocup. Acta Simulata Systematica Sinica,2005(1),2005.

12. Fang Bao-fu, Wang Hao, etc. The Survey of HfutEngine2005 Robot Simulation Soccer Team Design, Journal of Hefei University of Technology .Vol.29 (9). Sep, 2006.

13. Gao Jian-qing, Wang Hao, Yu Lei. A Fuzzy Reinforcement Learning Algorithm and Its Application in RoboCup Environment. Computer Engineering and Applications, 2006(6).

14. LIU Yang, WANG Hao, FANG Bao-fu, Yao Hong-liang. Application of the method of support vector regression in RoboCup. Journal of Hefei University of Technology Vol.30 (10). Oct, 2007.

15. Fang Bao-fu, Hong Bing-Rong, Wang Hao, Bao Dun-Qiao and Li Long. A Muti-Agent Defensive Strategy Based on Monte Carlo Method. Journal of Harbin University of Technology. 39(S1), Jun, 2007.