HfutEngine3D Soccer Simulation Team Description Paper 2020

Xu Zhang , Zhuolun Li , Haipeng Zhong , Xinpeng Hu, Hao Wang , Baofu Fang

School of Computer and Information
Hefei University of Technology, China

Abstract. This paper simply describes the architecture of HfutEngine3D team. This year we use the released code form UT and add ourselves decision information. Additionally, we use CMA-ES method and closed-loop control method of an omnidirectional walking model of humanoid robots which based on Cerebellar Mode Articulation Controller (CMAC) to achieve a steady and fast omnidirectional walk.

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, 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 3th place in 2010 in RoboCup,and the 6th in 2012. And we got the 5th in RoboCup China Open 2017. In 2018 we reached 2rd place in RoboCup China Open. 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 process based on Cerebellar Mode Articulation Controller to achieve a steady and fast omnidirectional walk. 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, op-ponents 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.

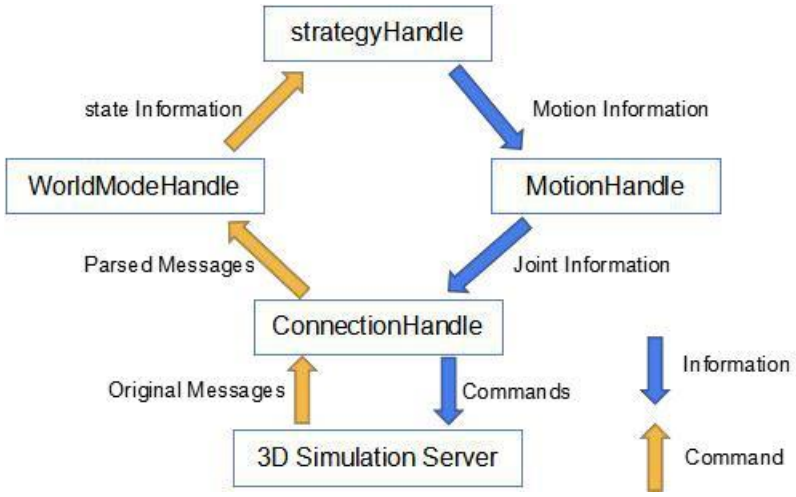

Fig.1. HfutEngine3D main running flow in about one cycle

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 length-ways 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).

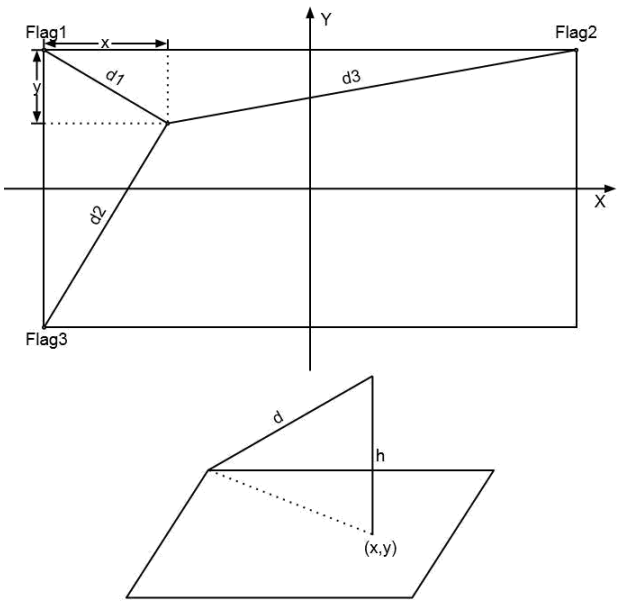

Fig.2. Self-Location and Calculating the Hight

In Fig.2, parameters 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 ycoordinate. Moreover, we get the x-coordinate by another triangle with field length, d1, d3. 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 Filtering

In the early versions of server, client can get completely accurate visual information,so we can get accurate coordinates. While after some updates, the noise jamming is added. The visual information got from server is no longer accurate.In order to reduce the error and get the accurate coordinates of position, we use Kalman filtering algorithm to process the data.Comparing the coordinates before and after filtering, we believe that the filtering is effective.

3.3 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.

3.4 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:

$$\bar{x} = \frac{g}{z_0} x \tag{1}$$

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.

3.5 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. Dynamicly calculate the acceleration based on the distance to target andangle.
2. Detect the change of target, dynamicly 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.

$$a = 1.67607lg(DistanceToTarget) \tag{2}$$

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 pre-dict the max velocity. Here are three groups of data:(d1,Vm1), (d2,Vm2), (d3,Vm3).

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

3.6 Actions

We add some actions in the latest version, such as kicking balls from the front or side. With the help of 3D Linear Inverted Pendulum, robots can make some complex actions. To cooperate with these actions, we also add some simple Articial Intelligence to help robots determine where and when to do the actions.

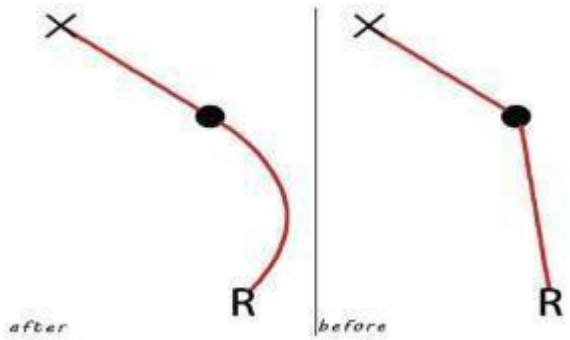

Fig.3. Dynamic gait planning

As an example, in Fig.3, we can see in the previous version, when robot(the big R in Fig.3) gets the ball's position and the target position, it adjusts its facing angle then walks directly to the ball. In our recent version, robot can get the path curve to the ballwith its facing angle, the ball's position and the target position. That will save time to get to the target and increase the accuracy of shoots.

3.7 Decisions

We make some functions for robots to decide their next movements. These functions are divided into two categories, individual decisions and team decisions. Robot relies on its self-location and the ball's position to decide what individual decision it takes.
Team decisions are used to control formation transform and other movements. To sum up, team decisions govern the formation, and individual decisions decides the speci al movement of individual robots.

3.8 Dynamic Role Assignment

Our team adopts the typical 2-3-5 formation and designs the real-time running target of five forwards based on KM algorithm. From the nature of bipartite graph, it can be seen that there will not be two nodes matching the same running position nodes at the same time in the points of forward players. Therefore, in order to let the robot not be limited to the thought of mechanization and improve the cooperation between them to achieve dynamic running, we decided to combine km algorithm and strategy design. First, the positions of five forwards and five running points are obtained. The first point set is the position of the forwards, and the second point set is the position of the running points, which constitutes the right bisection graph. Then, the weight of each edge is set to the opposite of the distance between the two connected positions, and the KM algorithm is used to find the maximum matching of the bipartite graph.

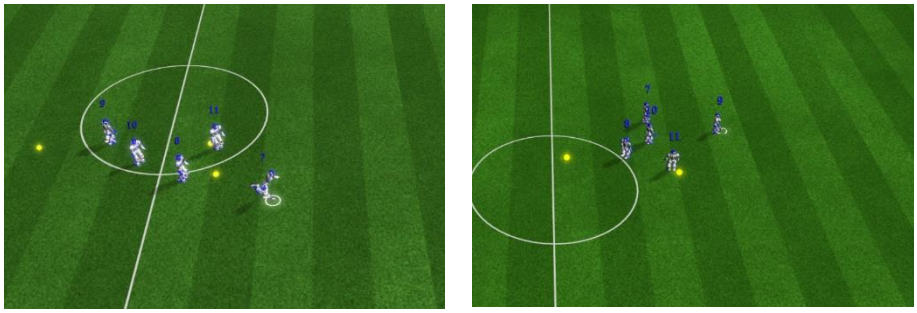

Fig.4. Example of Dynamic Role Assignment

3.9 Toolkits Development

We have developed a multi-threaded optimization script tool, in which users can perform multiple optimization tasks for agent kicking skills at the same time. Users can open multiple optimization tasks by running scripts, and then it will get the score of each task, and generate the action parameters of the next generation of population based on the highest score, so as to reduce the chance of the final performance when the robot runs a set of parameters, and multiple robots can also improve the optimization efficiency at the same time. Similarly, when optimizing robot walking, we can run multiple robot threads directly in the same arena to achieve the same effect.

Fig.5. Multi-Threaded Optimization

4 Experiments in CMAC

This year we use a closed-loop control method of an omnidirectional walking model of humanoid robots which based on Cerebellar Mode Articulation Controller (CMAC).The cubic spline interpolation method were used to plan the foot space trajec-tory, and built the Double Linear Pendulum (D-LIP). Subsequently, the trunk and foot poses were applied to compute the current joint angle value of the robot legs. Mean-while, the sensor information according to the CMAC model can be modified the angle value in the Inverse Kinematics to achieve a steady and fast omnidirectional walk.

5 Future Works

At present, we are studying how to judge the behavior of a robot in real-time during the game, including the state of dribbling and preparation before long-distance kicking, and how to deal with the corresponding behavior differently. So far, we are carrying out this research, which is progressing smoothly.

After that, we plan to continue to work on the optimization of complex robot skills. We also want to design new skills for robots, such as side kicks and other actions that are more in line with human behavior. At the same time, we will continue to improve our multi-threaded optimization tools. We still have a long-range goal to achieve.

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 ReinforcementLearning 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 coordinationin 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 linearinverted 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. InComputer Engineering and Applications, 2004(8), Aug, 2004
11. Baofu Fang, Hao Wang, Hongliang Yao, Jin Yang, Jin Zhou. The Application ofQ Learning In RoboCup. In Proceedings of 2004 China RoboCup, Oct, 2004.
12. Runmei Zhang, Hao Wang, Honghang Yao, Baofu Fang. Influence Diagrams andIts Application in Robocup. In Acta Simulata Systematica Sinica, 2005(1), 2005.
13. Baofu Fang, Hao Wang. The Survey of HfutEngine2005 Robot Simulation SoccerTeam De-sign. In Journal of Hefei University of Technology, Vol.29(9), Sep 2006.
14. Jianqing Gao, Hao Wang, Lei Yu. A Fuzzy Reinforcement Learning Algorithm andIts 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-AgentDefensive 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 LearningTechnical Based on Support Vector Machine Classification and Its Application in RoboCup. In Journal of Harbin University of Technology. 39(S1), Jun 2007
18. Mike Depinet, Patrick MacAlpine, and Peter Stone. Keyframe Sampling, Optimization, and Behavior Integration: Toward Long-Distance Kicking in the RoboCup 3D Simulation League. Proceeding of the RoboCup International Symposium 2014, July 2014
19. Hansen N. The CMA evolution strategy: A tutorial[J]. arXiv preprint arXiv:1604.00772, 2016.
20. Hansen N. The CMA Evolution Strategy: A Comparing Review[M]// Towards a New Evolutionary Computation. Springer Berlin Heidelberg, 2006:75-102.
21. Urieli D, Macalpine P, Kalyanakrishnan S, et al. On optimizing interdependent skills: A case study in simulated 3D humanoid robot soccer[C]// International Conference on Autonomous Agents and Multiagent Systems. DBLP, 2011:769-776.
22. Macalpine P, Stone P. Overlapping layered learning ☆[J]. Artificial Intelligence, 2018, 254:21-43.
23. 薛建. 基于 CMA-ES 算法的足球仿人机器人步态研究与实现[D]. 合 肥工业大学, 2014.
24. Zhou Z, Zhou Y, Wang J. Omnidirectional walk design of humanoid robots using layered learning method based on CMA-ES[C]// Advanced Information Management, Communicates, Electronic and Automation Control Conference. IEEE, 2017:464-468.