[1]WrightEagle 2003 – Sony Legged Robot Soccer Team

Xiang Li, Zefeng Zhang, Xueliang Liu, Qinrui Li, Kai Xu, Yanpeng Feng, Lei Jiang, Xiaoping Chen

Computer Science Department, Univ. of Sci. and Tech. of China, Hefei, China, 230027
{xlee,zhangzf,lxliang,kaixu,ypfeng,snowave}@mail.ustc.edu.cn, qrli@hotmail.com,xpchen@ustc.edu.cn

Abstract. WrightEagle – the only Chinese team in Sony legged robot league of RoboCup, has come to its third time to participate the competition as an experienced team with well preparation.

1. Introduction

WrightEagle was developed by Multi-Agent System Laboratory of USTC (University of Science and Technology of China), and here comes its third version of year 2003. Using AIBO, the four legged robot made by Sony Corporation as the research hardware platform, it aims at investigating in the field of top front technology of MAS.

2. Team Members

All team members are from MAS Laboratory of USTC. The team leader is Professor Xiaoping Chen, the director of MAS Lab and AI Research Center, USTC. Main members include Dr. Xiang Li (Ph. D student), Mr. Zefeng Zhang (master student), Mr. Xueliang Liu(master student), Mr. Qinrui Li(undergraduate student), Ms. Lei Jiang (master student), Mr. Kai Xu (undergraduate student) and Mr. Yanpeng Feng (undergraduate student).

3. Architecture

Based on the experience of the last two years, five modules are contained in our new team design: vision, locomotion, localization, communication and behaviors decision. Vision, locomotion and communication serve as the fundamental layers for higher modules. Localization module gets the analyzed image result from vision module and

[1] This work is supported by the National High-Tech Project (No. 2001AA422200) and the National Natural Science Foundation of China (No. 69875017).

body displacement from locomotion module, and tries to calculate robot position on the field for behaviors decision module which also get information from vision module, send out command to locomotion module and use communications among robots for better team performance.

4. Vision Module

The object of the vision module of this year is better accuracy, better adaptability to bad lighting condition, and to try to provide more information of the field.

The accuracy means both color discerning accuracy and distance estimation accuracy. To find out objects, we still mainly depend on colors, because the good color filtering process can filter out 95% of the non-important information. It's a pretty faster approach to get the profile of objects than approaches based on outline. And when more accuracy is required, outline approach can be applied at much lower cost. For distance estimation, we've introduced angle base calculation to replace the interpolation between samples. Angle is more accurate than size. The error that we've measured is within 100mm most of the time

Lighting condition can significantly affect the vision. We've chosen to use color sample. That makes it quite easy to get the color in special lighting condition. It's also a fast approach. By letting much of the computation pre-calculated, it can save much CPU time for further computation. YUV is still the only color space we consider about.

We are now developing ways to recognize complex objects like the field and dogs. This is designed to be a high level above color filter process while the low level information is still available. Starting from the simpler, we're trying to reconstruct the field in memory based on the vision information, without marks.

5. Locomotion Module

With the lessons of locomotion design in the last two years, we aim at the stabilization and multiformity of robot locomotion design. We divide the locomotion module into two components, the walk design and the fixed action design as we did last year. The fixed actions, such as ball shooting, getting up or swinging the ball are all based on experience. We simply make the action description files and loaded them into memory after the boot up of the robot. The design of fixed actions is mostly focused on tuning the action till they are more accurate and get more chance to succeed.

Considering that the instability occurs when the robot is switching between different walk types when heavy computation takes place, we shift this computation to the middle period of the computation. That is to say, instead of computing the body stance for each frame at the beginning or at the end of each step, we make such computation when the robot is much easier to 'stand' on the ground. This method also brings us the advantage of managing the stable body stance when performing different walks because they mostly require the same middle body stance.

We also designed the cripple walk command and included them into the locomotion module. The difference between normal walk command and cripple walk command is that, besides performing the normal walk, the robot must also try not to lose the ball it controls which involves using its mouth and its front legs to retain the ball.

Another way to increase stabilization is to insert more frames where all four legs are on ground. This is also possible because we always calculate the body position on the way before we calculate each leg related to the body which drives it. All we should do is to insert a leg moving segment between the switch between two pairs of legs in air and use four legs on ground to drive robot body forward.

To improve the efficiency and speed of robot walk, we are trying to maximize the extent in its possible range while walking. By searching in a pre-defined locomotion range space and send the appropriate command, we can drive the robot more quickly and reduce the frequency of wasteful leg switch.

6. Communication Module

Because communication is a very important characteristic of multi-agent system, we pay much attention to it. We comprehend wireless network better after using it at last year.

We establish point to point connection to communicate between our robot players. Each robot is a server and a client, it can send and receive data. We use a flow control to satisfy the limitation of bandwidth in game. We prepare using CRC code for finding out mistake in communication.

The information that robot can get by wireless network includes roles, other plays' location, ball location from other plays, ball control and tactics of whole team (attack or defense). Robots can know the situation of match and do some tactical cooperation depending the information. Because the identification of player is not enough in vision module, information communication can avoid crashing into each other in match. When a player has controlled the ball, the tactics of whole team will become attacking. It will send information to others. Other player will know the ball has been controlled by teammate. It will walk to opponent's half-court and prepare tactical cooperation.

7. Localization Module

Last two years, we were using a simple way to achieve self-localization. It's fast but not very satisfying. So, new approach, Monte-Carlo localization, is introduced to our team.

Monte-Carlo localization is a probabilistic approach. In this method, the current location of the robot is modeled as the density of a set of samples. Each sample can be seen as the possible location and pose of the robot. Therefore, such each sample is a vector representing the robot's x/y-coordinates in millimeters and its heading direction in radians. Localization module updates sample set by using information of seen landmarks and goals received from vision module and information of performed action received from action module.

Because of the limitation of vision, information of landmarks and goals is not always very accurate, especially for far landmarks. We introduce a method to make adjustment on the distance estimation of far landmarks by using information of near ones. It improved accuracy of localization.

This new localization module supplies more reliable self-localization results and has a fairly good speed.

8. Behaviors Decision

Behaviors Decision module has modeled three world state information: self world model, team world model and self memory world model. Self world model stores all the information from localization module, vision module and observer sensors etc which directs robot to make decision by itself. Single robot's decision is based on the decision theory and conditional state machine. For single robot, the whole decision process is a state transition process, but in some situation robot needs to used decision theory to justify the best action.

Self memory world model stores some important information in last five to ten decision cycles. Its main function is to make robot more smart and more heuristic.

Because different area in the field stands for different chance for attack and defense, we divide the field into different grid space to justify a robot's behavior, in default case we also has a default grid to make default decision. Different grid has different properties such as attacker coefficient, defense coefficient, grid transition list etc. Besides we can use grid to define a robot's fundamental attack and defense area when cooperation .

Behaviors decision of this year is focused on multi-agent communication and cooperation. So each robot has team world model to store other robot's information and command from other robots. Using team information robots can find ball faster and more effective to attack and defense.

In multi-robot communication and cooperation, we focused on next problems:
1.When encountered emergence condition, the robot can react as immediately as possible. So we use "Scenario" to describe all this situation, and in the specific condition we match this condition with the "Scenario" library as quickly as possible and notice other robot to do the right thing. In this situation, we need to address problem such as "Role Assignment", "Scenario Match", "Parameter Selection ", "Scenario Description" etc.
2. In order to advance single robot's skill, we use other robot's information. We will analyze the effect of the communication against no-communication. Other robot's information and self observation information will be combined and we have a mechanism to justify the selection of the best using information.
3. If all things go well, we will also design a multi-agent decision problem focused on conflict . We will address the decision problem with the concept of "conflict". All the robot's decision is about to address a conflict. So we will design an architecture about conflict: how conflict is generated and how conflict is solved etc.

9. Conclusion

These five modules serve as different parts of team WrightEagle2003. Compared with team of last two years, we have made much improvement in low-level layers of team design --- faster and more reliable robot movement and more robust vision system. It is the guarantee of better performance of our new soccer team to Padova.

Recent Articles

1. Xiang Li, Zefeng Zhang, Lei Jiang, Xiaoping Chen: WrightEagle 2002 – Sony Legged Robot Team. Computer Science Department, University of Science and Technology of China, Hefei, China, 2002.
2. Zefeng Zhang, Xiaoping Chen: A fast cheap robot vision system, The fifth academic conference of AI in China, Computer Science Department, University of Science and Technology of China, Hefei, China, 2003.