WrightEagle 2009 SPL League

Feng Wang, Guoqiang Jin, Jinsu Liu and Xiaoping Chen
Multi-Agent Systems Lab., Department of Computer Science and Technology,
University of Science and Technology of China, HeFei, 230027, China
xpchen@mail.ustc.edu.cn
http://www.wrighteagle.org

October 30, 2008

Abstract

This paper discribes the WrightEagle SPL Team, which was founded in 2000, and took part in the Aibo's game last 8 years, and the Nao's game in 2008.

1 Introduction

WrightEagle is a robot soccer team developed by Multi-Agent Systems Lab of University of Science and Technology of China(USTC). Now it consists of four groups, which are 2D simulation group, 3D simulation group, MSRS group, and SPL(4-legged and 2-legged) group. WrightEagle SPL team was founded in 2000. We are the first Chinese legged robot team that has ever entered RoboCup. In last 8 years, our team participated in the Aibo's competitions of every year. In 2008, WrightEagle SPL took part in the Nao's game at RoboCup2008 held in Suzhou. We are still the first Chinese team on 2-legged robot in SPL.

WrightEagle team consists of four groups as described above, all of which are expected to be related to the research on and development of Nao robots. Every group consists of PhD. students, master students, and undergraduate students. We will do our best to further our researches on Nao league.

Here are the team members: Aijun Bai, Dongpeng Chen, Min Cheng, Kangping Fu, Dinghuang Ji, Jianmin Ji, Guoqiang Jin, Haoxiang Li, Huan Lin, Fei Liu, Jinsu Liu, Yunfang Tai, Feng Wang, Jiongkun Xie, Feng Xue, Ke Shi, Limin Zhao, Zhiqiang Zhang, Zongzhang Zhang and Ming Zhong.

2 Technology report

2.1 Motion Control

Precise control in the robot behavior planning is an open problem due to the jumpiness of vision, unruly locomotion, and various kinds of delay in the whole system. This problem always occurs when Nao is finding or reaching the ball, where searching ball or extra turning frequently occurs. When the ball is close around, the robot has to adjust its gesture (e.g to bend over) to find the ball again, which further aggravates this problem. In our implementation in RoboCup2008, we used simple algorithms which just depends on the vision module for localization. We have tried to obtain more accurate information, but it results in longer decision cycles. We planned to do following work to address this problem on physical 2-legged robot, Nao, and the work is in progress.

1. Design various walking engines, first make it walk steadily, then consider the velocity and other elements;
2. Optimize walking engines using machine learning approaches to find stable, fast, robust and flexible gaits;
3. Design cases for automated odometry calibration based on leg model;
4. Construct a model for robot motion, consider kinds of delay effect and predict control of robot walking; System,

Based on above supports, we adopt a method of *Planning with Control* to gain a precise control. That is, use the information from perception module and odometry calculated in realtime to estimate the current state of the robots and the target, and simultaneously predict the state of the nearly future according to motion model, then base on all of these is a small-area and short-time path planning to approach the target, where planning segment is to be switched if obvious excursion from preconcerted path occurs.

A new layer is introduced between upper planning module and lower processing module.Thus skills, such as approaching ball, turning to goal gap and so on, developing on this layer become not so hard but more robust.

2.2 Vision

WrightEagle develops a simple and effective vision module to recognize ball and goal specialize to the digital camera on Nao.

As a result of the fact that we can't set the auto-exposition parameter of the camera, we take these two cases:

Following is the description of the Figure 1.

1. Normal images, different color can be separated easily.

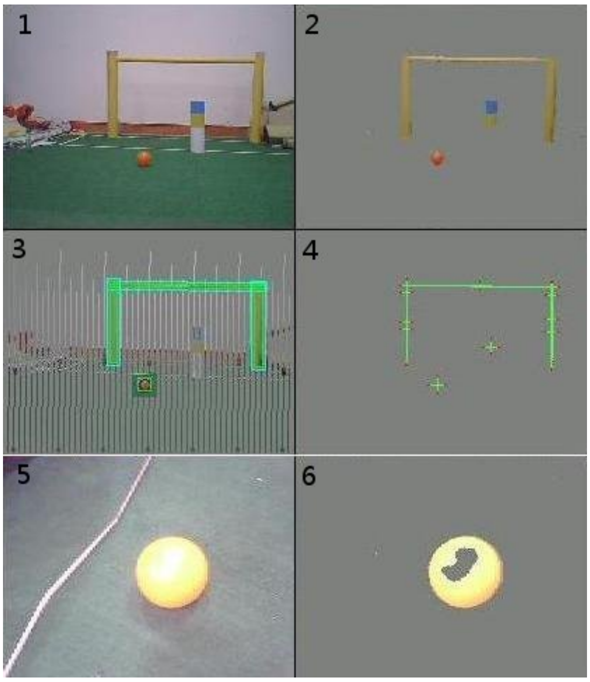

Figure 1: images

- Normal, the image is of high quality, simple segmentation algorithm is enough to handle this, so we use a simplified version of the AIBO's vision module. See the first four images of figure 1.
- Abnormal, when Nao looks down, the saturation of the image is very low because of the autocompensation, as a result, the color changed sharply, using only AIBO's vision module is not enough, we give a specialized version. See the last two images of figure 1.

2. Segmentation results using an off-line color table, we can recognize orange, yellow, blue and green (not shown).
3. Goal and small ball recognized and vertical scan line.
4. When find the interested color, continue finding the boundary of the color.
5. When Nao looks down, the brightness of the image is high and the saturation is low, the color is different from the normal case.
6. Segmentation results using the specialized version when Nao looks down.

The input of the vision module is the raw image of YVYU format and the DH matrix of the camera, using the angle calculating from the DH matrix to detect the correspond case and launching different methods. As following:

1. Normal Case:
    - Image segmenting: using a color table obtained off-line.
    - Object detecting: finding the right color on the scan line. If orange color is detected, using ball detector to find the orange area, recognize ball and filter fake ball.
    - Ball recognition: When detecting the orange area, handle it differently as big ball or small ball.

- Goal recognition: matching the shape and location of the color area detected as goalposts or beam, to decide the left and right goalpost. The result is also accurate when Nao sees just one goalpost.

2. Ball recognition when looked down: Because the light changed sharply, the color table is useless. But the image only contains green ground (background) and ball (the object), and there is a big difference between these. So we using a specialized color table to segment the image, also consider the hole appeared in the color area. see Figure 2.

Figure 2: Y U V grey tones of the 5th image of Figure 1

2.3 Decision-Making

Although we did little on decision-making in our Nao team last year duo to too tight schedule, we did a lot in our simulation 2D team, which has won the champion in RoboCup2006 and the 2nd place in RoboCup2007 and RoboCup2008.

When building our 2D team, we especially concern on proper methods for decision-making in each simulated cycle. We are trying to model the whole decision process as a Partial Observable Markov Decision Process (POMDP). When the probabilities of observation are transformed to the probabilities of state-transition of the next state, the model is simplified to a Markov Decision Process (MDP).

- **State** S**:** In each state, the ball is controlled by one player. When the ball is free, the agent directly predicts the possible state of the ball under control, based on the interception model which is presently quite mature.

- **Action** A**:** We have defined some actions such as Pass, Dribble, Shoot and etc. Each action has its own formal model, to compute the possible state and the probability of state transition.

- **Reward** R**:** The basic idea is to discriminate the several main scenarios, appointing different goals to each. The highest reward is achieved when the goal is reached; otherwise a small reward will be given according to state point sensitivity.

A basic decision process is as follows. The agent analyzes the situation, makes sure the current state and confirms the goal state. The action generator generates many actions as candidates. Each candidate has its reachable state and the transformation probability. The obtained values of the states are computed by an approximate method iteratively. At last, we can get the best decision result based on the current state.

Though, as is known that most of decision-making in this league is base on rules or logic, empirically, we believe that introduction of some calculating or probability mechanism is quite considerable. The adopting of MDP in our 2D simulation achieves a lot, but when porting it into physical robot, we come up against two problem: firstly, lower supporting items like perception, world state, skills and so on should be well modeled, which is actually a hard and long-term job; secondly, how to simplify and optimize the process to cope with big amount of computation in such resource constrained systems like Nao robots.

A simplified version of decision-making with probability has been tried out. Though the result is not good enough, we will continue the work on the aspect.

3 Related Research in the Lab

We have conducted a series of research work on the MDPs(markov decision processes) and Motion Control while took part in the previous RoboCup competitions. [fan07] gave a new real-time dynamic programming algorithm BIRTDP to solve the MDP problems. [fengwu07] described the Correlation-MDPs used to solve the DEC-POMDP problems in multi-agent planning, and some experiments are carried out on 2D simulation team, it will be furthered to the SPL team. [haitao07] implements a new approach to calculating and calibrating the odometry for Quadruped robots Aibo, it helps WrightEagle Aibo team got the 2nd place in RoboCup2008, and it will be adjusted to the Nao robots. [jinsu08] introduces online ZMP sampling search algorithms, which helps a lot in motion control and will be used in our SPL team for RoboCup2009.

References

[fan07] Changjie Fan and Xiaoping Chen. Bounded Incremental Real-Time Dynamic Programming. IEEE Proceedings of FBIT 2007, Jeju Island, Korea, 2007.

[haitao07] Haitao He and Xiaoping Chen. A Model-Based Approach to Calculating and Calibrating the Odometry for Quadruped Robots. Proceedings of RoboCup Symposium 2007. Atlanta, America, July 2007

[fengwu07] Feng Wu and Xiaoping Chen. Solving Large-Scale and Sparse-Reward DEC-POMDPs with Correlation-MDPs. Proceedings of RoboCup Symposium 2007. Atlanta, America, July 2007.

[jinsu08] Jinsu Liu and Manuela Veloso. Online ZMP Sampling Search for Biped Walking Planning. IROS 2008.