

Team Description Paper for Team I-KID Robocup2013

Bi Junxi¹, Tan Xiaofeng¹, Guo Yunkai¹,
Xie Pengcheng¹, Zhang Jiwen², Lou Xiaoping¹

¹ Beijing Information Science & Technology University, Beijing, China

Email: louxiaoping@bistu.edu.cn

Web: ikid.bistu.edu.cn

² Golden Robot and Manipulator Co. Ltd., Beijing, China

Email: zjwts@jinnuoman.com

Web: www.jinnuoman.com

Abstract. In this Team Description Paper, we describe the main changes of our humanoid robot for RoboCup 2013 Netherlands. We mainly illustrate the new designed hardware of our robot, the improvements of field line detection, world modeling, behaviors control and the new IMU module that aim to improve the performance of the robot.

1 Introduction

Team IKID is organized by a group of challenging undergraduates and graduates in BISTU (Beijing Information Science & Technology University) where Team Water, who won the first place of Mid-size league in the RoboCup 2010 Singapore and RoboCup 2011 Turkey, comes from [1]. We devoted ourselves to the research field of humanoid robot since 2010, and the first time we took part in RoboCup is China Open 2011, and become one of excellent teams in China with fast pace. After that, with little experience in world champion, our team scored 12 goals and finally reached the elimination round in RoboCup 2012 Mexico. For RoboCup China Open 2012, we also won several honors. Based on these field competitions, we are stepping to upgrade the hardware as well as relevant algorithms including vision processing, world modeling, behavior and motion control. This year, we also designed a type of new robot by ourselves with great enhancement in computing capability and phased-out the GRM-Lite which is not competitive for the time being.

2 Overview of the hardware of the newly designed robot

The newly designed robot makes a significant improvement with the computing speed. PC104 computer is replaced with a cortex-A8 based CPU Samsung S5pv210. The processor clocked at 1GHz does not only exceed in frequency, but also in power consumption. 512MB memory combined with 1G FLASH is sufficient to load any algorithms for soccer, such as fast image processing, particle filter based world modeling. High data load on USB bus with YUV space image, which means noise vulnerable for transfer, is exchanged with smaller sized JPEG alternatives. While on PC104 computer, the time consumption for JPEG decompression is unable to stand, S5pv210 is capable to decode the image at little cost, powered by its hardware JPEG codec.

The mother board where the CPU is seated, is composed of a microcontroller as a gait controller, USB ports, an Ethernet for updating program, a power conversion module, a RS232C interface for transferring data from the main board and a RS485 interface for receiving and sending commands to Robotis Dynamixel. The new controller structure is depicted as Fig 1.

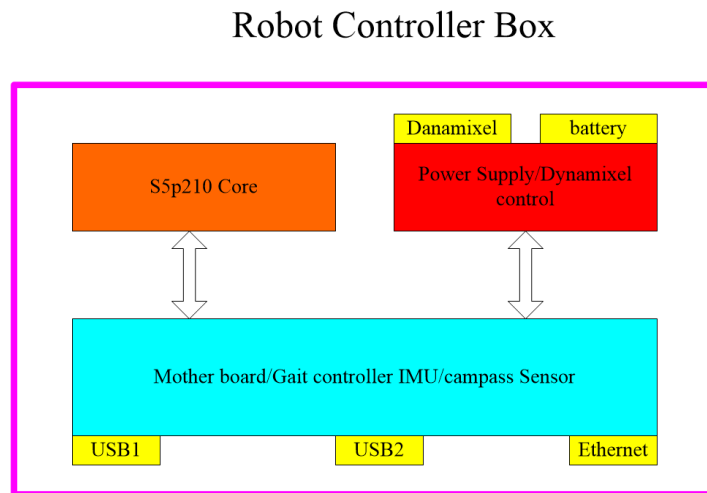


Fig. 1. The framework of hardware

Instead of multi-boards stack mounting method with PC104 standard, which makes it heavy, less robust and hard to be cued, we adopted an encapsulation of this core CPU with backplane into a standard box. Therefore, the robot controller is exchangeable as a

whole unit rather than stand-alone bare PCBs. The standard controller unit will greatly improve the robustness, and make fast treatment possible during a game play in emergency. Mechanical structures are also updated to fit this controller unit, but few are changed for DOF configurations.

Though equipped with an IMU module in previous type, the robot is unable to make full use of its output due to noise and interference. This year we introduce Unscented Kalman Filter to handle this problem [4]. As compass is introduced, it is also preprocessed by a UKF, despite of its much simpler model than IMU.

Due to current sensor technologies, one cannot expect a MEMS IMU with high precision enough to correct joints' state for every control cycle. Instead, we are able to adjust the gait parameter for a relative long period in face of perturbations. Such parameter includes the inclination of torso during single support period, the side swing of upper body as well as foot lifting height. Tuning happen in every footstep cycle rather than control cycle, the response frequency and accuracy is sufficient in such condition. Therefore, by large amount of experiments on real robot, we achieve a faster walking ability with probably 20% improvement while still maintain stability.

3 Improvements of algorithm

3.1 Algorithms Overview

An intelligent robot can be abstracted as a sensing-motion coordinating system, the data stream flows from external sensors to internal descriptions and then external motion execution again. Therefore the software architecture is organized as hierarchical style depicted by Fig 2 which can be split into 4 parts: vision processing, world modeling, behavior control, and motion generation. The following sections interpret each of them, especially their improvements.

3.2 Line detection algorithm improvements

Previous field line detecting algorithm which is based on Sobel derivative operator is noise vulnerable, especially within an image filled with other bright colors, such as illumination, and the threshold of white color is also time-consuming for calibration; in addition, the procedure is quite independent with other object detectors, such as balls and gates, therefore, recognition of field line is extremely unstable with high computing density. As a result, in previous localization algorithms, landmark poles still play an

essential role rather than field line, but this year we developed a novel procedure for robust white line detection to improve our localization quality in an acceptable runtime.

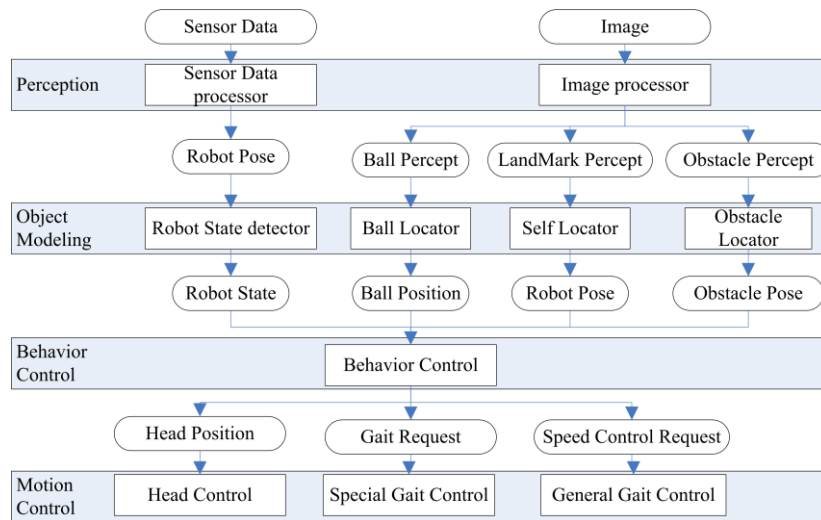


Fig. 2. Structure of Decision-Making Algorithms

Firstly, line scan direction is changed from horizontal to vertical, enabling us to detect the field borders. Within this region, where outer side white color is kept away, we can continue the scan to find any white line point candidates, and clustering them as white region. This procedure does not exclude ball and gate pole clustering either. Additionally, we adopt Run-Length-Coding to get a better understanding of its contour, which is quite different from the 4-direction connecting procedure as described last year. Filtered by experience, a single region forms a line spot as stated in [2].

A line segment clustering procedure is then executed to combine them together. To ensure that two segments are on the same line, their direction and distance should be close enough. This can be done by Hesse Normal Form and the clustering Algorithm stated in [2]. The center circle is also clustered with a shorter length below the predefined threshold. Radius which is calculated from two short line segments in a projected ground can be an judgment of a real circle.

Finally as stated in our previous algorithm, cross point is calculated as landmarks, which is the final usage of line detection. Since line region is relatively thin than other objects, several predefined rules are used for further filtering that cannot be described in detail. Combined with open source code from B-human in Standard League and

Darwin-OP [3] from Upenn, robust line detection is realized, and can act as essential role in self localization.

3.3 World Modeling

Particle Filter acts as an essential role in our world modeling procedure. In order to improve this key algorithm to reach a much more precise and stable result, we updated the particle filter algorithm in two aspects: elimination of the unnecessary resampling, and a new strategy to dealing with kidnapping and localization fault recovery.

The resampling used to take place every cycle of iteration, and increase the risk of losing particle diversity, and thus make a worse description of the distribution. To fix this shortcoming, current resampling happens depending on the variance of the importance weights.

Instead of resampling every cycle the robot maintains the importance weight in memory and updates them as follows if the variance of important weights of the particles is within the threshold.

$$w_t^{[m]} = \begin{cases} 1 & \text{if resampling took place} \\ p(z_t | x_t^{[m]}) w_{t-1}^{[m]} & \text{if no resampling took place} \end{cases}$$

This is based on the concept that the variance of weights relates to the efficiency of the sample based representation [5]. Resampling should happen only when the particles suffer from representing inefficiency, as depicted as Fig 3.

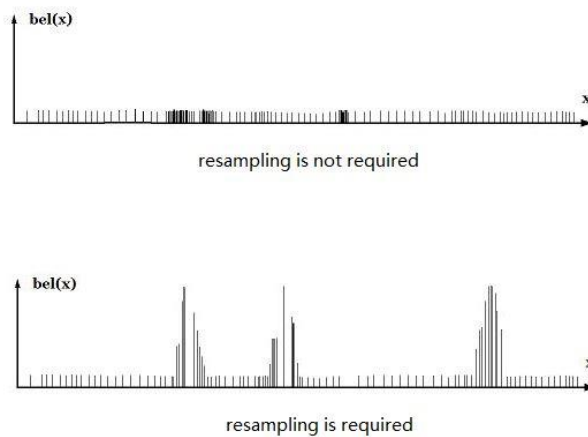


Fig. 3. Resampling required time [5]

In addition, Dealing with kidnapping and localization fault recovery should be paid special attention within a play field involved by robot handlers as well as referees. The algorithm should recover in a short time. The solution we previously adopted is the injection of a certain amount of random particles for each iteration. And this year we add those particles with changing amount and distribution much similar as stated in [5].

During this algorithm, a random sample is added with probability

$$\max\{0.0, 1.0 - w_{fast}/w_{slow}\}$$

Where

$$w_i = w_i + \alpha_i(w_{avg} - w_i), (i = \text{slow, fast})$$

$$0 \leq \alpha_{slow} \ll \alpha_{fast}$$

w_{avg} stands for the average importance weight of the particles.

We also make full use of compass with Kalman filter described in section 2 to assist localization due to the released new convention. Odometry is also updated from gait parameter (x, y, θ) to embedding with IMU measurements. With the technologies we developed this year, self-localization is greatly improved, and supplies a better resource for behavior control than ever before.

3.4 Behavior considering stochastic distribution

We discover several problems which are cumbersome when coding for an intelligent behavior for football playing, most of which are due to ignorance of probabilities embedded with a self-controlled robot. Sometimes, the robot hesitates just before the gate with a proper excellent position for goal. At other times, when robot is going back after a goal, the robot speed is always low with unnecessary modulation of his angle toward the target position for a kick off, but finally, it failed to localize himself back to the half field in time.

One of the solutions may be limiting the bouncing times among decision states, whereas it breaks a long term tuning with large error. Actually, the problem arises from the inherent stochastic observation errors. Current decision making process is heavily dependent on a single estimation of the robot posture as well as the state of football, whereas the world modeling result is the belief expressed by its density function. General method would be a single estimation generated by expectation calculation, or

density extraction which replaces this distribution with a certain value. In order to take into this observation uncertainty, we brought in a particle behavior method. The method simply execute sensor-actuator closed-loop controlling algorithm for every particle generated from particle filters. As a single particle stands for a possible state of the robot, we can make the decision for every such state, and then make an expectation across these particles.

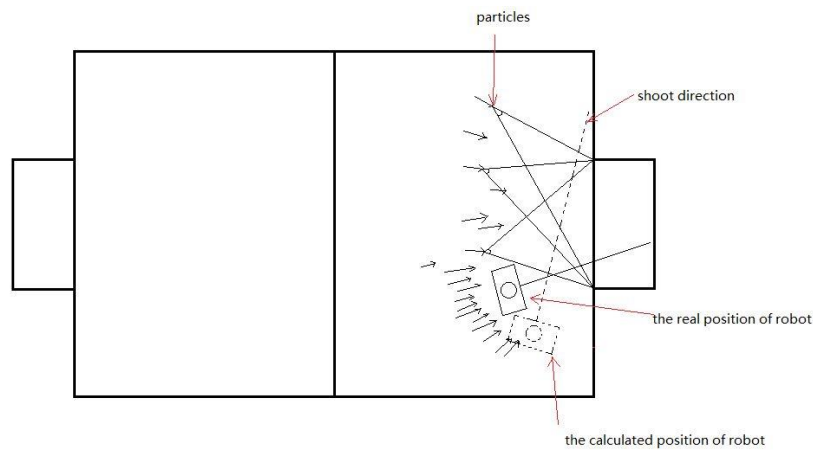


Fig. 4. An illustration of estimation failure, but proper pose for a goal with single particles

Figure 4 illustrates such a particle behavior process whether a kick action should be issued. On one hand, robot may start to kick when he saw both of the opponent gate poles even if a localization procedure is not complete and reliable; On the other hand, the final result is not only a kick command, but also a combination of kick action and possibilities to goal, which will supply reference for higher behavior.

Another improvement lies in a splitting of head control and motion control [2]. Though similar head control modes as stated in are also used in our previous version of code, however, the current head control strategy and motion command is strictly combined; therefore, the environment acquiring strategy is not based on information quality, but through pre-programmed style.

This year we do not only introduce the independent head control loop into our framework, but also try to make it much more advanced by evaluating every marked objects, including the gate pole, and the mark pole (which is inexpertly abandoned for RoboCup 2013, and we will make use of field line landmarks instead). The behavior

dynamically regulate expect to ball or self-locations, and head scan mechanism will try to establish such expect by focusing its eyes to different objects.

4 Conclusion

In this paper we described the hardware of our newly designed robot and the algorithms that we will use to improve the performance of the robot. With these changes, we can trust that our robot will have a better show in RoboCup 2013 Netherlands. Our goal is not only winning in this game, but to share the knowledge with other teams from all over the world.

IKID will provide a referee knowledgeable of the rules of the Humanoid League during RoboCup 2013 Netherlands and will participate in this game as schedule.

References.

1. Lou Xiaoping, Zhang Jiwen, Zhang Haitao, Zhu LianQing.: I-KID: Team Description for RoboCup2012(2012)
2. Thomas Rofer, Tim Laue, et al: B-Human Team Report and Code Release 2011(2011), <http://www.b-human.de/>
3. The code of Team DARwIn, <https://fling.seas.upenn.edu/~RoboCup/wiki/>
4. St-Pierre, M.; Gingras, D. Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system. In Intelligent Vehicles Symposium, 2004 IEEE(2004)
5. Sebastian Thrun, Wolfram Burgard, Dieter Fox.: Probabilistic Robotics. The MIT Press Cambridge, Massachusetts, London (2005)