

SEU-UniRobot Team Description Paper

Haidan Gao, Chuan Liu, Peng Wu, Shiyu Ying, and Wei Zhou

RoboCup Research Group, Southeast University, Nanjing, China haidangao@163.com

Abstract. This document describes the current system of kid size humanoid robot designed by our team, SEU-UniRobot, for the RoboCup 2017 competition in Nagoya, Japan. We have participated in the last competition in Leipzig, Germany and had an over-all update afterwards. Thus some changes implemented in robotic mechanical structure and software architecture will be demonstrated in this paper. Moreover, we look forward to accumulating more experience and getting a better result.

1 Introduction

In this paper, we describe the robot system for the autonomous competition in the KidSize group of Humanoid League. We have made lots of improvements both in hardware and software system for the RoboCup 2017 competition, hoping for better grades at the second show in Humanoid League.

This year we changed the team name SEU-UniRobot, last year we use the name SEU-Jolly.

Four brand new robots from SEU-UniRobot named Jolly1, Jolly2, Jolly3 and Jolly4 are fully autonomous robots, who are fixed to the limitation in sizes and other aspects. More details will be specialized in following parts.

2 Overview of the System

A photograph of our robots is showed in Fig.1. And the Table.1 just tells the specification of our robots, each of which consists of a USB camera, a computer board, an Inertial Measurement Unit (IMU), 16 servo motors, a battery and several user interfaces such as buttons without electronic compass. In Fig.2 we demonstrate the software system. The robot continuously locates where he is by processing the information from camera and IMU. At the same time, he takes actions to search for the soccer at intervals. Once he finds the soccer, relative commands given by the strategy part will be sent to the body control process for next behavior. Then, servo motors decode and execute these commands quickly. By combining various simple commands, the robots can realize corresponding actions required for soccer competition.

3 Computer System

3.1 Hardware

Our main computer board (GB-BX) with Core i3-5010 CPU is typical of the high computational capability which is very significant for the robots. And a 60GB



Fig.1. Robot Jolly1

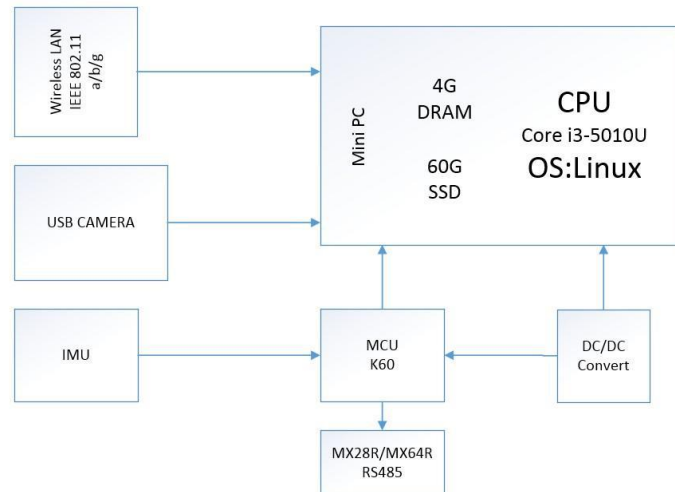


Fig.2. Hardware system

Weight	3500 g(Including Batteries)
Height	630 mm
Velocity(Forward)	0.2 m/s
Walking Directions	All Directions
CPU Board	GB-BX(Core i3H-5010, 1.7 GHz)
OS	Linux(Ubuntu 14.04)
Interface	Ethernet x 1, USB x 2, Push button x 2
Servo motor	MX-28R x 8, MX-64R x 8
Battery	ACE(14.8v 3300 mAh)

Table 1. Hardware System

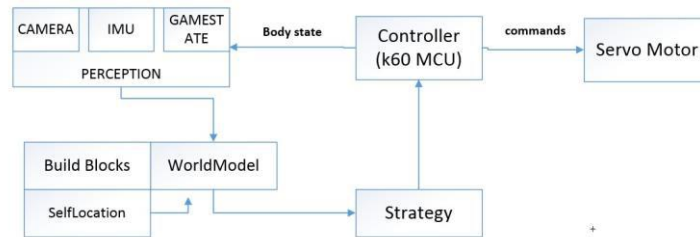


Fig.3. Architecture of software system

SSD is attached to the board for higher I/O speed. All of these are set to support the robot to react quickly in the fierce competitions. With this in mind, all software modules we develop including perception and control are executed on it.

3.2 Development Environment

We use Ubuntu 14.04 32 bit as our development environment both on the robots and our computers. We make use of the router to connect with the robots, which supports for both LAN and WLAN connection. So we just edit and compile source code in our computers, debugging too of course. The WLAN provides us with an easy solution to sending files to robots so that we can code everywhere we like without any limitations. However, a direct connection between our computers and robots is also accessible with the Ethernet port when necessary. Considering these all, every teammate is able to try their own idea over and over again.

4 Software System

4.1 Architecture

Figure 3 shows the architecture of our software system. It is mainly composed of following three parts: perception, strategy and control. The perception part gets information from sensors and game controller. Images are captured by a USB camera and processed in the computer board. Useful information would be sent to the World Model as the sharing parts among processes. Based on these information, blocks like soccer and goal would be built up for self-location and strategies. Since external and internal information both prepared, the strategy part would generate relevant actions for the competition.

The body control tasks are operated in the dedicated control part. It not only controls the body according to the commands from the strategy part, such as walk or kick; but also periodically sends the status of the robot (e.g. posture) to the World Model. All these operations depend on sending relevant commands to servo motors by control process. Meanwhile, an IMU is used for gyro feedback and posture estimation. All of these are written in C++.

4.2 Body Control

Two different kinds of servo motors are applied on our robots, MX-64R for the legs and MX-28R for other parts. The maximum torque of MX-64R is 64 kg-cm, which is sufficient to support robots' actions very well. Under the ZMP planning for motion, the robots get fast and stable mobility.

4.3 Image Processing

This process has been promoted this year. Now it is divided into three parts: soccer, ball and site recognition. Site recognition, as a newly added method, plays an important role in image processing. According to the color table, green points belonging to the site will be taken. Then, field boundaries will be extracted after convex processing, which will be used in ball and goal recognition afterwards (shown in Figure 4 and Figure 5).

As for goal recognition, based on field boundaries, goal post inside the field could be detected. Then, we can extract the edge of it by vertical and horizontal scanning lines. However, in order to reduce false recognition, we will refer to the distance measurement by camera according to different width between goal posts in different observation points.

Finally, in terms of soccer recognition, based on the Harr-like feature, we use the Adaboost algorithm to learn what the soccer is like. To improve our algorithm, we apply the particle filter to minimize the area which we need to search for soccer. In Fig.6 we give an example about the capture of soccer. The resolution of the images

can be selected from either 640x480 or 320x240. Our algorithm runs at 20 fps with onboard computer.

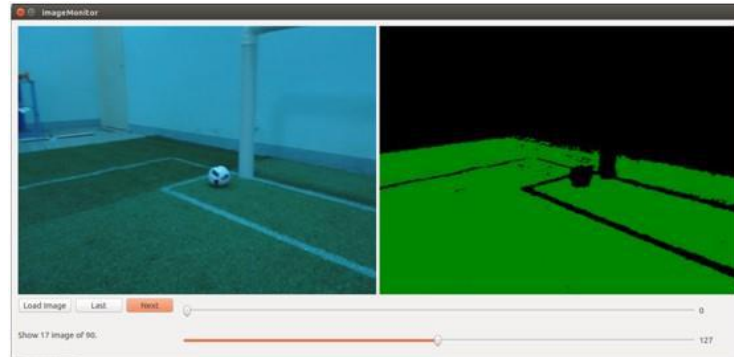


Fig.4. Green points detection

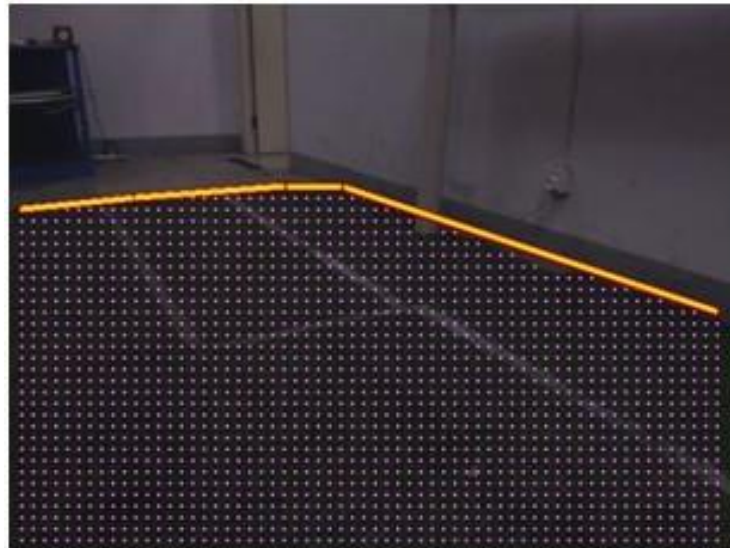


Fig.5. Site recognition



Fig.6. Capture of Soccer

4.4 Self-Location

Based on last competition's experience, we give high priority to Monte Carlo particle filter algorithm. And in our latest system, we take goal's distance and orientation as the feature for robot's self-location. The number of particle is 60 and the final location is generated by K-means algorithm. In Fig.7 we give an example of our new self-location system. Since the electronic compass made serious drift and we were unable to recognize the belonging of goals, which resulted in many own goals in RoboCup 2016 competition, we would make efforts to solve this problem and improve the preciseness of self-location this year.

4.5 Strategy Part

We use two nested finite state machine for strategy control. External state machine is responsible for robotic start-up, standing up, kicking the ball and localization. Internal state machine is responsible for searching the ball, posture adjustment and dribble. Fig.8 and Fig.9 show the both state machine.

5 Conclusions

In this paper, we give a brief introduction to our robots which are newly born for the RoboCup 2017 competition in Nagoya, Japan. Although this competition is still a big challenge for us, we believe that our robots would pay back what we have devoted!

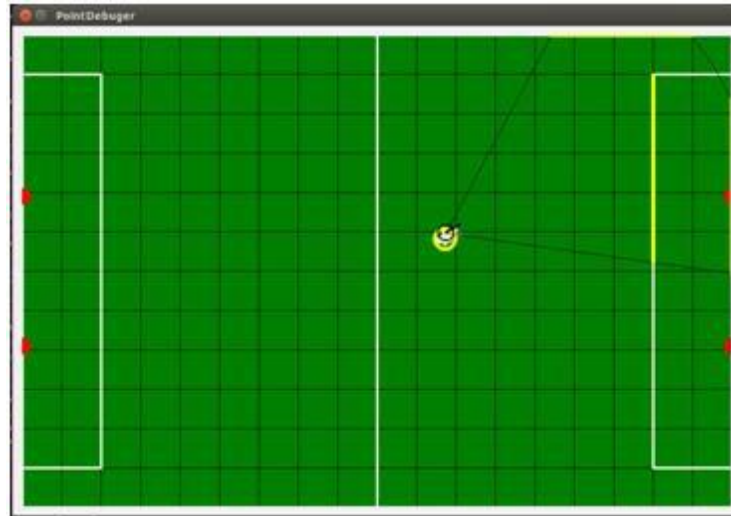


Fig.7. Self-location

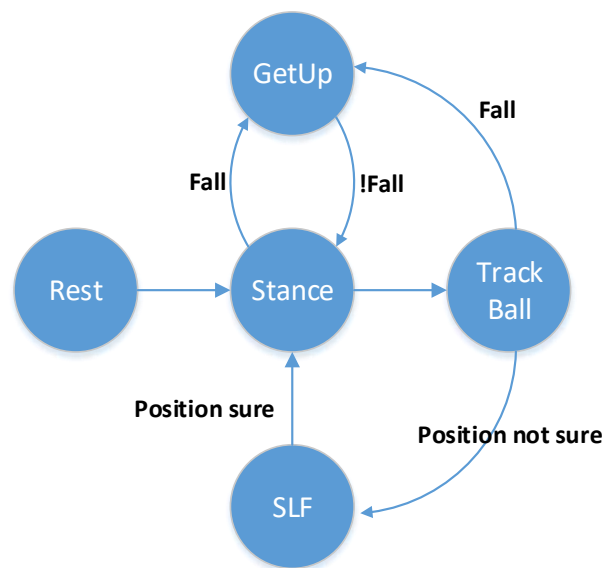


Fig.8. External State Machine

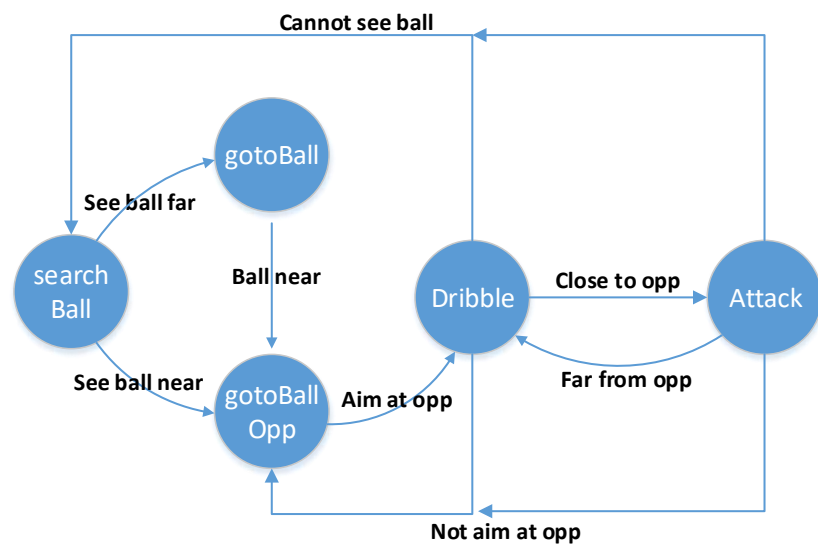


Fig.9. Internal State Machine (Track Ball Machine)