Kor Kai Team Description Paper 2011

Saran Lertpradit, Tapana Voravoothiwat, Nathakorn Nophakunvijai, Putti Thaipumi, Panjamapong Sermsawatsri, Thanakorn Chindanonda, Tanaphon Tangngamjit

Department of Computer Engineering
Department of Mechanical Engineering
Chulalongkorn University
Phayathai Rd., Bangkok, 10330, Thailand
tapana.v@gmail.com

Abstract. In this paper, we present an overview of a kid-size humanoid team "Kor Kai" from Chulalongkorn University, Thailand. This year we have applied new techniques and approaches as described in the paper.

Introduction

Kor Kai, formerly known as Chibi Dragon, is a team consists of students in Engineering Innovator Club established under Faculty of Engineering, Chulalongkorn University. Since 2005, we have participated in Robocup competition. Later on 2008, we won first place in both SSL and Rescue league under the names of Plasma-Z and Plasma-RX respectively. In 2011, we are currently pursuing the challenge of humanoid league.

During the development process, we focused on developing well-structured design result to simplicity of implementation and debugging process. The issues from previous competitions are carefully reviewed and analyzed in order to improve our work.

An overview of software structure, our system is separated into four independent processes, which are vision process, localization process, strategy process, and control process. C++ and IPC are main tools for implemented our software. These processes are executed on Intel Atom processor which equip onboard with the robot.

Low level hardware control is done with our custom made servo controller board, which is designed to control up to 16 channels of servo motors. Two servo controller boards are used separately to control head and body of the robot. The

servos movement commands are passed through serial communication from Atom processor board to this level for execute.

Hardware

The robot named “01” is built based on Kondo KHR-1 HV. It consists of 18 servos, accelerometers, gyroscopes, Atom processor board, and 2 servo controller boards.

Logitech Webcam Pro 9000 camera is mounted on the 2 degrees of freedom neck. Atom processor board on the back of the robot executes the computer vision, localization, strategy, and motion control system. Figure 1 shows the outline of 01.

Software Architecture Overview

Our software runs on Linux operating system. The design separates workload into four processes, communicating with each other via UNIX socket.

Apart from the four main processes, simulation software is built for multi-purpose testing by sending simulated input messages to the desired process, then monitor on the output for debug and testing. While the simulator is not used for debug, it can be used as a graphical user interface displaying feedback received from the robot.

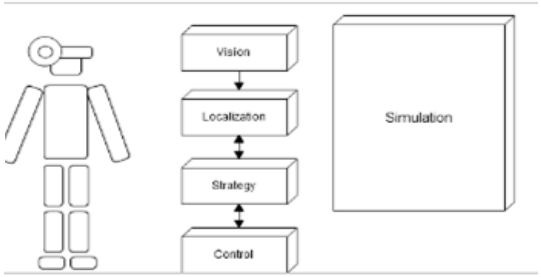

Fig.1 Software Architecture

Image Processing

We use one CMOS camera (Logitech QuickCam Pro 9000 2.0 Mega Pixels CMOS Camera) as an image capture device. The image resolution is 640 by 480 in RGB format. Our vision algorithms are mainly based on colors detection. We first

apply color threshold by using a lookup table. Then segment the image and apply pattern recognition to identify the objects in the image. Hough Transform is also used to extract the field line. After the objects coordinate in the image frame are obtained, these data are passed to Localization Process.

Localization

The localization process computes the absolute location of objects on the field, which are robots and ball. Landmark poles are used as references of the robot's location. Formerly, our localization algorithm was not accurate because no filter was applied to the calculation. To improve the accuracy, this time, the particle filter method is used..

Particle filter method produces an amount of particles. Each represents an absolute position of robot. The process is divided into 3 main steps, which are control modeling, sensor modeling, and re-sampling.

Control modeling is the method using control data of the robot to compute the position change of each particle. Accuracy of this method output mainly depends on accuracy of the control data.

Sensor modeling uses images from camera to compute absolute position and probability of each particle.

Re-sampling is the method of creating new particles. The method chosen is to create new particles at the most probable particle. The particles created will disperse themselves in control and sensor modeling steps in the next loop of the program.

Strategy

Strategy process decides robot behaviors, which are high-level behavior (game plan and strategy) and low-level behavior (walking and turning).

After the process has been successfully computed, it returns control data to the control process in order to perform the movement.

Controls

Motion control process controls movement of the robot. Gathering desired movement from strategy process, then compute to a set of servos joint angle value. The trajectory of the feet is divided in to 3 axes. Each dimension is generated using waveform generator online [3]. Movement Direction and speed are simply control by changing the amplitude of the waveforms [4]. The parameters of the waveforms can be obtains by using machine learning algorithm for more robust locomotion. 2-axis gyroscope inside robot is also used to maintain stability.

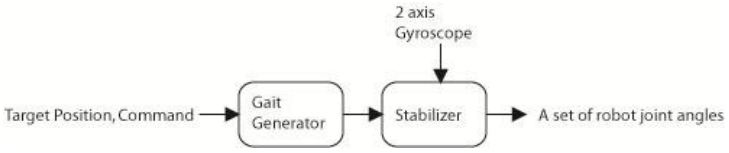

Fig.2 robot control process

Conclusion

As this is the second time participating Robocup Soccer Humanoid League, the problems and difficulties during the implementation process are the main factors that we concern. The software architecture is improved to solve data flow issues.

More implementation approaches has been applied this year, including lines detection, inter-process communication, machine learning, and particle filter. These result in better software performance. The processor, control board, camera, robot design, and servo motors are improved this year.

However, there are some issues needed to be improved, such as robot walking speed, image processing frame rate, and many other issues. And it is our intention to apply the improvement in all aspects.

Reference

1. Laue T.; Jeffry de Haas T.; Burchardt A.; Graf C.; Röfer, “Efficient and Reliable Sensor Models for Humanoid Soccer Robot Self-Localization” in *The 2009 IEEE-RAS Intl. Conf. On Humanoid Robots*, 2009.

2. D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte Carlo Localization: Efficient Position Estimation for Mobile Robots,” in *Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, 1999.

3. Sven Behnke, "Online Trajectory Generation for Omnidirectional Biped Walking," in *Proc. of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida*, 2006.

4. Felix Faber and Sven Behnke, "Stochastic Optimization of Biped Walking using Gyro Feedback and Phase Resetting," *in Proc. of the IEEE/RSJ Int. Conf. on Humanoid Robots (Humanoids)*, 2007.