UURT Humanoid Robot Team Description Paper for Humanoid KidSize League of RoboCup 2013

R. Shabani, M. Sheikhlou, F. Tofigh*, D. Fazli, S. Rabiei, and R. Ghasemzadeh

Robotic Lab.
Faculty of Engineering
Urmia University - Urmia 57153, Iran
uurt@urmia.ac.ir
http://www.uurt.ir/

Abstract. This paper describes humanoid robot systems of Urmia University Robotic Team (UURT) for the upcoming KidSize competition RoboCup2013. Experiences of the last competitions lead us to do more modification on our software design in order to act more reliable and easy to manage. Implementing fast and real-time control, making decisions by the robots in contact with each-other to manage the game and localization are our main goals.

Statement of commitment

The team commits to participate in RoboCup 2013 and to provide a referee knowledgeable of rules of humanoid league.

1 Introduction

Including different research fields like machine vision techniques, stability problems, machine intelligence, etc. humanoid robots are one of the challenging and interesting fields in robotic world.

UURT is a student robotic team of Urmia University, established in 2005 working on Deminer robots, Soccer 2D simulation, Humanoid and Aerospace researches and has awarded some successes in national robotic competitions in Iran. Working on humanoid robots commenced on December 2010 by working on Robotis' "Bioloid Premium Kit"[1].Soon, has been found that alterations needed to control this kit so mechanical design team, designed a new structure using connection links of Bioloid kit.

This paper provides a brief overview of our robots after experiencing RoboCup2011, Istanbul, Trukey, IranOpen2011 and IranOpen2012 for the next competition.

---

* f.tofigh@gmail.com

[1] http://www.robotis.com/

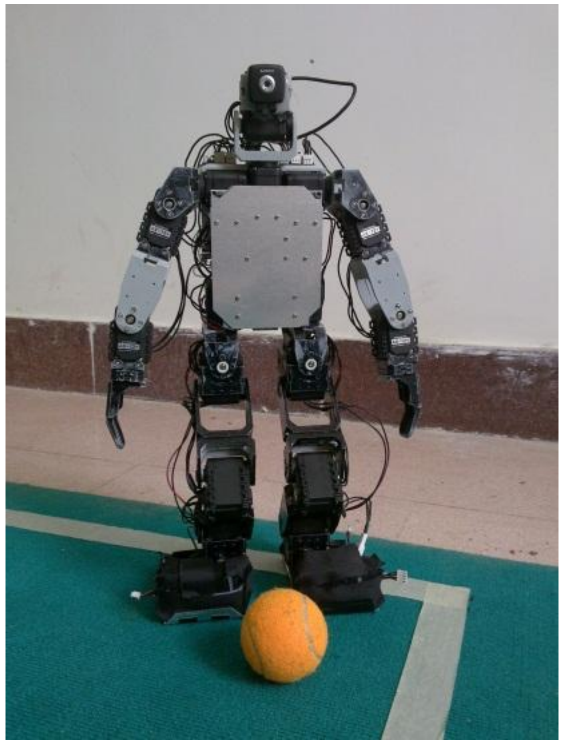

Fig. 1. UURT Humanoid KidSize Robot

2 Researches

The main research interests are:

- Hard real-time control;
- Machine Learning techniques and application;
- AI planning;
- Performance analysis and Optimization.

3 Mechanical and Electrical Designs

The basic set of the "Bioloid Premium Kit" was enhanced by changing the complete mechanical structure and using Robotis Dynamixel MX-28 servos and a 6 degree of freedom Inertial Measurement Unit (IMU) sensor. The Robot has 20 actuated degrees of freedom, 6 per leg, 3 per arm and 2 degrees of freedom for neck. The robot is controlled by RoBoard RB-110 based on the Vortex86DX, 32 bit x86 1000MHz CPU with 256MB RAM[2]. It is equipped with VIA VT6655

[2] http://www.roboard.com/

Chipset wireless adapter. RB-110 is powered by a Li-Polymer battery providing 2 Ah, which gives us 40 minutes of run time. A combination of Gyroscope and acceleration sensors has been used to facilitate the control of robot to have a stable motion. All sensors are monitored using ATMega328p, ATMEL microcontroller, and the results send to RoBoard RB-110 using RS-232 serial port.

The actuators are Dynamixel MX-28 and AX-12 and the camera is μCam Serial JPEG Camera Module connected to RB-110 using full duplex TTL port and equipped with a pan/tilt. Actuators are powered by an additional Li-Polymer battery and communicate with RB-110 through TTL port of communication. Physical specifications of robots are 47 cm height and 1.9 kg weight which is classified in KidSize according to RoboCup rules.

4 Software Design

The software of the robot, consists of 7 parts. 4 programs prepare robot to run the pilot. these 4 tools use to designing the motions, simulation, design and export decision making for pilot and test Image processing algorithms. further more, a library of image processing module and a hardware module to communicate with servos implemented which has been linked to the pilot program.

4.1 Pilot program

This program is used to make decisions according to the outputs of sensors and image processing. the pilot program should be able to make the right decision in different conditions.

The most important features considered in the pilot program is reliability and fault tolerance. All hardware errors tried to be covered in software. The pilot program, has to detect all hardware errors and fix them.

Main section of pilot program is decision making which is a finite state machine. This machine includes some internal states to implement defined controlling operations. Moving among states based on external trigger like finishing of processing an image frame or a sensor event.

4.2 Image Processing Software

To have an environmental perception a vision system is essential. The main goal of vision system is to detect objects of interest, such as ball, goals and robots. The input of this system is the image frames acquired by camera and are fed to image processing algorithm.

Considering the overhead of new mechanical balance subsystem, we have had to reduce image processing load. All per-processor filters are removed and frequency of image grab is reduced by 40 percent. Considering all these changes we have lead to have a less-precise vision system.

In the processing stage, at first RGB color space is converted to HSV color space. Then special colors have been searched filtering special ranges in Hue

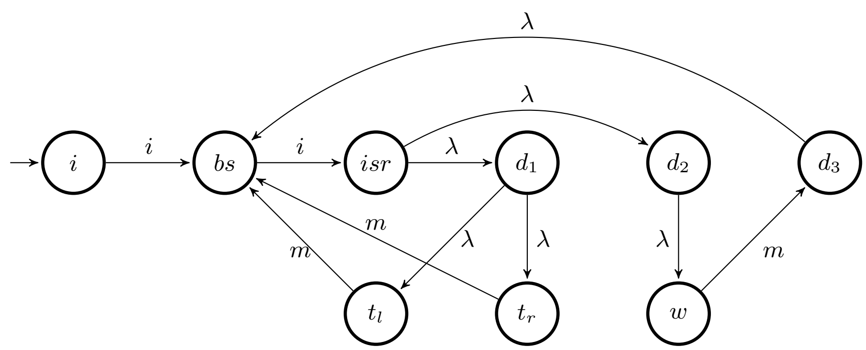

Fig. 2. Decision diagram

and Value matrices. These ranges themselves, are obtained by a small GUI tool **"Vision Control"**. Output is an XML1-compatible file containing HSV ranges of colors and information about ball, camera and field. Then this file is used by vision algorithms as an input reference. This makes program more flexible in different lighting conditions.

An additional detection algorithm using Hough transform is used to identify the lines on the field[1]. By knowing the lines we are able to detect specific marks on the field. Based on the found objects and an internal perception model, the robot will try to localize itself on the field[2,3].

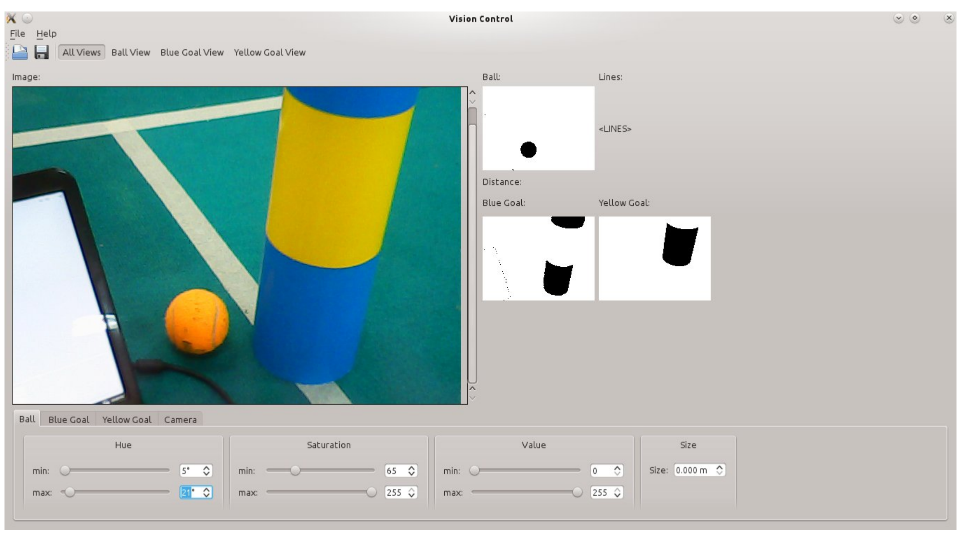

Fig. 3. Running Vision Control program - All segmentation

4.3 Behavior Engine

Considering the problems of coding finite state machine, such as test and debug, modification and etc, a graphical software has been developed which gives the ability of designing a machine or applying any modification more easier. the output of this program is the source code for controller.

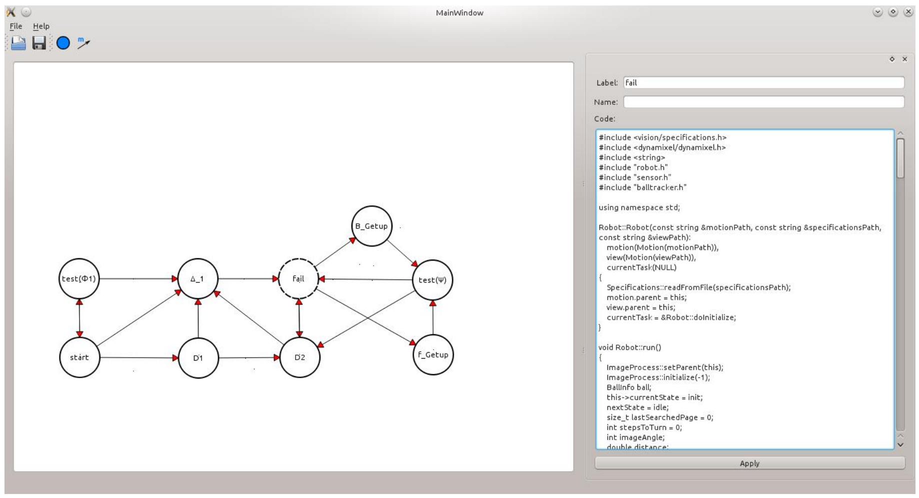

Fig. 4. Behavior engine

4.4 Motion Control

Motion control is the part that converts high-level commands issued by behavior engine to a sequence of predefined packets for motors. These packets have been modified using Robomotion program by Robotis company. A compiler has been developed to convert `mtn` files to a more flexible language that gait planner can interact with easily. This format includes a higher-level description of actuator positions and hides details of low-level hardware communication protocol from gait planner.

To manage the communication with hardware, a library interface is used to control the timing and sequences. Indeed, this layer gives a high-level interface to pilot program to set it free from the details of hardware. This layer implemented as a independent library.

5 Gait Planning

The ZMP theory is used to generate stable walking pattern. The ZMP is defined as the point on the ground about which the sum of all the moments of the active forces equals zero. If this point remains in the support polygon, then the robot can have some control over the motion of itself by applying force and/or torque to the ground. Walking is a periodic phenomenon, including two phases: single-support phase and double-support phase. In the single-support phase, compatible hip and feet trajectories are designed based on ZMP theory; then, respective leg joint angles are calculated using inverse kinematics model of the 6DoF robot's legs and in the double-support phase the ZMP is transferred from one foot to another.

6 Conclusion and Future work

As explained in previous sections of this paper lots of hardware and software tools were developed by UURT to reach good and successful achievements. Currently, we are focused on implementing dynamic real-time control on our robots and also decision making by the robots in the game in contact with each other.
Team progress reports are available on our homepage.

References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing (2nd Edition). Prentice Hall (2002)
2. Rath, C.: Self-localization of a biped robot in the robocup domain. Master's thesis, Institute for Software Technology, Graz University of Technology
3. Chang, S.H., Chang, W.H., Hsia, C.H., Ye, F., Chiang, J.S.: Efficient neural network approach of self-localization for humanoid robot. In: Pervasive Computing (JCPC), 2009 Joint Conferences on. (dec. 2009) 149 –154