Tsinghua Hephaestus 2014 AdultSize Team Description

Mingguo Zhao, Sotirios Stasinopoulos, Ning Jiang, Qingqiu Huang, Yiqian Chang, Yilun Chen, Xu Zhang, Yifan Hou, Qi Yan, Yaotang Chen, Kaiyuan Xu, Peng He, Yang Liu, Ziang Yan, Qianhui Wu, and Sibo Jia

Tsinghua University, Beijing, China
mgzhao@mail.tsinghua.edu.cn

Abstract. This document describes both hardware and software specifications and practical functions of the humanoid robot THU-Strider, developed by team Tsinghua Hephaestus as a platform for research in bipedal locomotion, robot self-localization and multi-robot cooperation. The robot will also be used to participate in Humanoid League(AdultSize) of RoboCup 2014 Brazil.

1 Introduction

The Tsinghua Hephaestus is a RoboCup Humanoid League team running at Dept. of Automation, Tsinghua University, China, since July 2006. Before that, we had three years participation experience in RoboCup Four Legged League and got the fifth place in the Technical Challenges of RoboCup2006 Bremen. Our current research interest is focused on bipedal locomotion[1][2][3][4], robot self-localization[5][6] and multi-robot cooperation in dynamic environment[7]. The humanoid team had taken part in the RoboCup2007 and RoboCup2008 both in KidSize and TeenSize. In the RoboCup2008, our TeenSize team got the 2nd place and our KidSize team went into the Round Robin2. Moreover, our TeenSize team got the 3rd place in Robocup 2009 and 2010. From 2011 and on we started to participate in Adultsize. We got the 3rd place in RoboCup2011, 2rd in Robocup2012 and 3rd in Robocup 2013. Now we have been getting prepared for Robocup2014. THU-Strider is an AdultSize humanoid soccer robot developed as a platform for Tsinghua Hephaestus RoboCup team in RoboCup 2011. The main goal is to develop a light weight and fast walking soccer robot. A passive dynamic walking based powered walking-Virtual Slope Walking is been developed for gait generation and some mechanical refinements are ongoing. This document will give a general view of the robot.

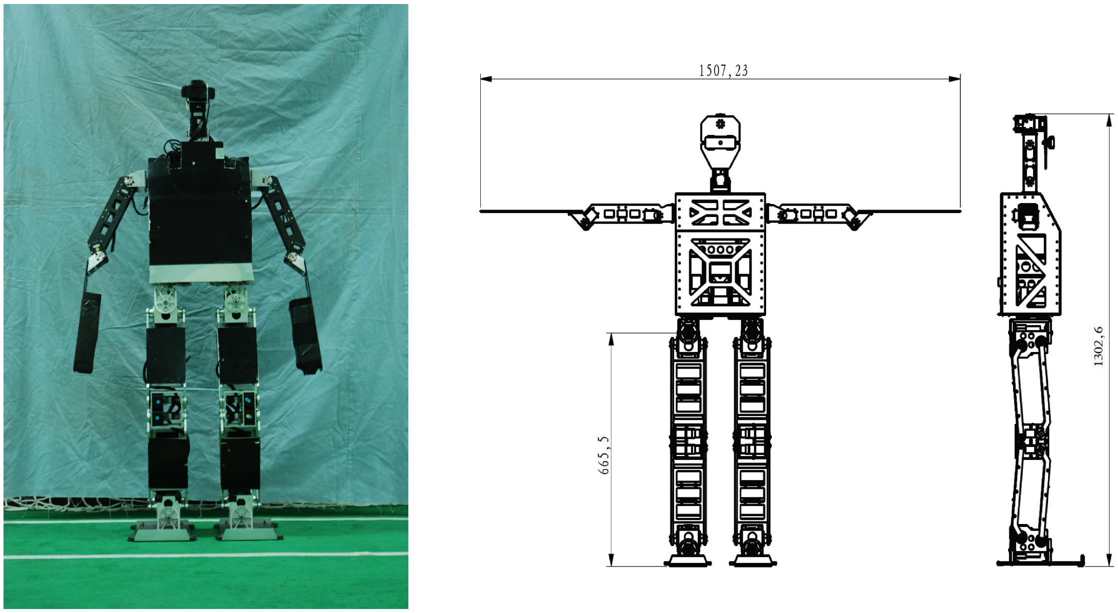

Fig. 1. 1) THU-Strider robots 2) Robot Dimension

2 The Robot Design

Fig 1-1) shows our THU-Strider robot in practice. The robot has a height of 1320 mm, and weights 18.10 Kg, including batteries. The detailed dimensions are shown in Fig. 1-2). The robot has 18 DOFs: 5 in each leg, 3 in each arm, 2 in the head. For THU-Strider, 10 Vstone V3310 servo motors are used as actuators for legs and 8 Robotis EX106+ for arms. We use NI SingleBoard RIO-9606 as the motion controller, connected with all motors. And use an Intel NUC DC32171YE as the Main Controller of the robot, with three USB ports, one for the motion-controller, another for the gyro and the other for the camera. Specifically, an Mti-28A83G25 Gyro is mounted on top of the robot head, while a Logitech USB camera accompanied by a fish-eye lens located in the robot head is employed as our visual sensor. Buttons and LEDs, located on the back, are set to control and indicate the robot state. The motors of upper body are connected in series on a RS-485 bus and lower body (leg) motors are connected in series on a LVTTL bus. The details of the control system are shown in Fig. 2.

3 Software Architecture

The software architecture is developed on Robot-Operating-System(ROS) and the distribution used is ROS Groovy. The whole software system consists of three main modules:

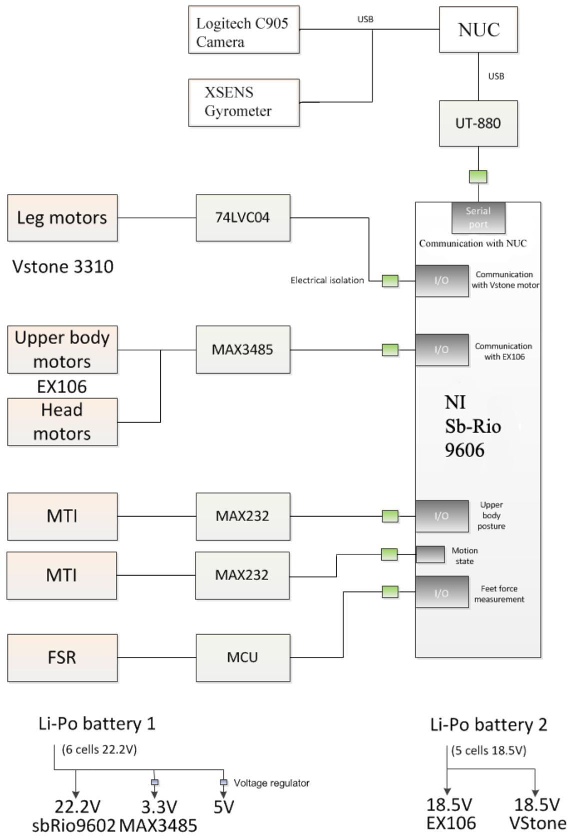

Fig. 2. Control System Architecture

Cognition, Behavior and Motion. Each module runs several different nodes in parallel to realize a specific function. Modules are arranged so that they are independent from each other. Both Cognition and Motion interchange data with Behavior through ROS messages or services. Cognition is responsible for information gathering from both camera and Gyro, self-localization and perception. Motion is responsible for gait planning and motor controlling, while Behavior acts as the brain of the robot, analyzing the data from Cognition and sending orders to Motion. Module configurations and data flows are shown in Fig. 3.

Image Grabber grabs images from the vision sensor and generates related information of the image and the pose of the camera.

Image Processor processes the incoming images grabbed by Image Grabber, and yields information needed for Localization and Behavior Control.

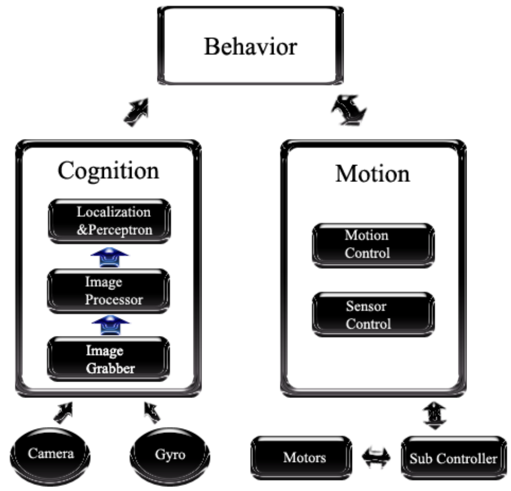

Fig. 3. Software Architecture

Localization&Perception: implements the Particle Filter localization algorithm, manages position information of robots and the ball, so as to be used by Behavior Control.

Behavior controls the game process and makes behavior decisions.

Motion Control manages all the actuators of the robot, and controls locomotion or any other actions of the robot according to the requests from Behavior.

Sensor Control manages other sensors, and interacts with the Sub-Controller.

4 Vision

A Logitech USB camera accompanied by a fish-eye lens is employed as our visual sensor. The Vision module has two tasks: object recognition and relative position estimation. The object recognition process is based on the results of color segmentation. However, we do not use the entire image to retrieve features for object recognition; the raw image is first undistorted to eliminate the distortion effects of the wide-angle fish-eye lens, using the extrinsic calibration parameters obtained through the OpenCV calibration procedure, and the best undistorted image part is kept. Then, we resize the original image into a smaller one and use it for the detection of our objects, thus saving computational resources. This procedure takes full advantage of the tools the newly applied ROS middleware provides us with.

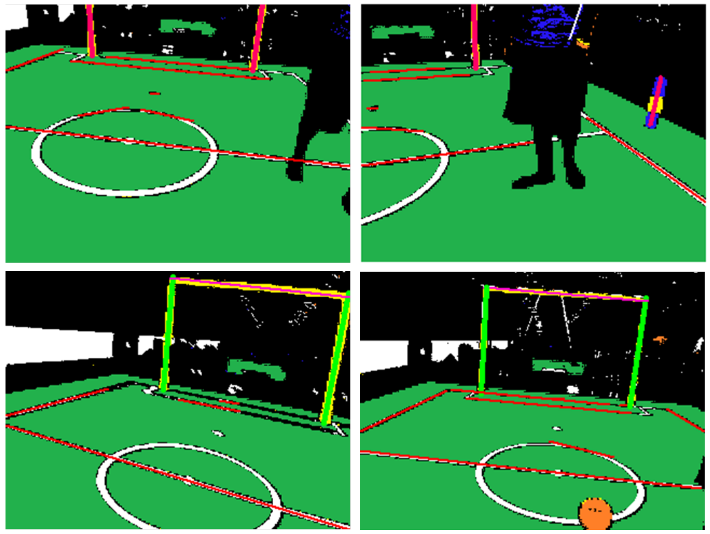

Fig. 4. Result of Recognition

4.1 Vision Sensor

A Logitech 2-MP Portable Webcam C905 and a Focusafe 1/2" 1.55mm IR MP fisheye lens are used to provide the visual input of our robot. This combination yields a eld of view of about $180(horizontal) \times 135(vertically)$, but after the calibration procedure, the used field of view is about $130° \times 90°$. The robot has to rotate its head to widen this field. The connection to the main controller via USB 2.0 provides real-time image series of a resolution of 800×600 of 30 fps, but as explained earlier, the resized images used for object detection have a resolution of 480×360.

4.2 Color Segmentation

After images are captured, the RGB pixels are mapped to 8 color space simultaneously to minimize the storage cost, using a monochromatic image, reducing greatly the storage space required and avoiding the waste of bandwidth for the transmission of the image to the object detecting modules.

4.3 Feature Extraction

Objects are distinguished by colors. As the features, clusters of different colors were extracted along the scan line from top to bottom or left to right. And the process would end if no more special colors appear. The special color found calls further object recognition in related areas. For instance,consecutive orange pixels call the process of ball detector.

Single special colored point would not be considered as a target object, in case there was some expected error spot in the image especially during walking or other dynamic situations.

4.4 Object Recognition

Features, characteristic series of colors or a pattern of colors, are used for object recognition, e.g., a sequence of some orange pixels is an indication of a ball. Fig. 4 shows the recognized ball, goal and obstacles in the image.

Ball detector After Color Segmentation, Breath First Search (BFS) is used in order to find candidate regions of the ball. Then for each candidate regions, the convex set of the region is calculated and three points are sampled randomly on its edge. At last, a circle is fitted with these points and the fitting error is evaluated. If the error is less than a certain threshold, this region is accepted as a ball.

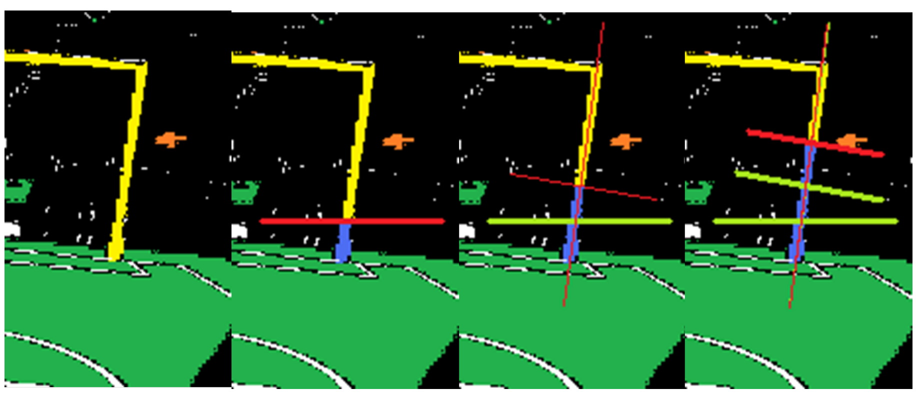

Fig. 5. Goal detector

Goal detector Goal is a reasonably large rectangle area with yellow pixels. We use an algorithm based on moment-method just like linear regression searching the goal from the bottom of the frame and checking the minimum eigenvalue of the inertia matrix simultaneously. We stop when the eigenvalue increases greatly and suddenly. Then we check the direction of the goal and the existence of a goalkeeper.

Field and Obstacle detector After the field is detected using its characteristic green color and already known shape, we look for black or dark colored objects that interrupt the field pattern and categorize them accordingly, calculating their width and distance.

Field Line detector Moment-method is also used in the Field Line detector, which starts searching for a random white point, and uses linear regression simultaneously. It stops when we find descending quality of regression. Then use a hypothesis inference to find all points on the line. The complexity of all the algorithms is $\mathcal{O}(n)$ (n is the number of points for a special color). The result of recognition is shown in fig. 4.

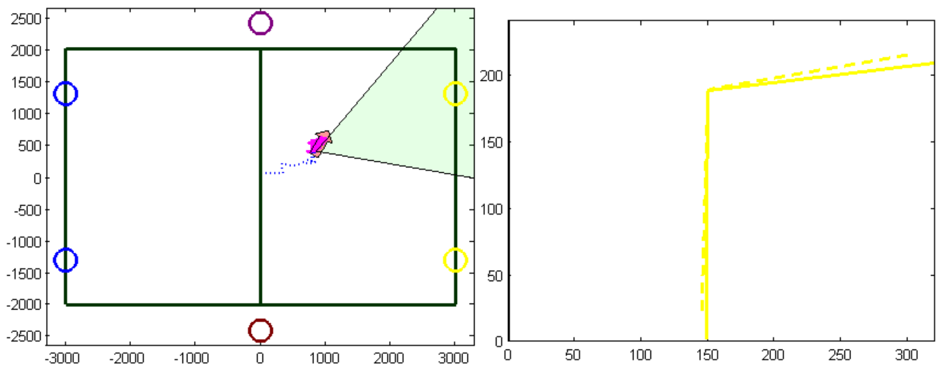

Fig. 6. A Demo of Particle Filter

4.5 Localization&Perception

Starting this year, we are using the information obtained through our visual and gyro sensors to implement our robot localization algorithm. The algorithm is based on a Particle Filter (Monte Carlo Localization), integrating odometry data from the movement commands given to the robot as well. In that way, we are able to attain an estimation of the position of our robot and other objects, while performing our tasks.

5 Behavior Control

The data provided by the sensors and localization modules is used to plan a more complex behavior series. And the module of Behavior Control takes charge of this task. The main task is separated into subtasks until they can be described as a set of basic behaviors which can be executed by the robot. All this is done by a hierarchical state machine described in XABSL (Extensible Agent Behavior Specification Language). The basic motion actions are transferred to and interpreted by the motion module, while other basic actions are processed in further modules.

It can output the following variables: 1) A motion request of basic behaviors to inform the motion module of the robot's next action. 2) A head motion request of head mode to inform the motion module of the robot's next head action. 3) 3 LED's state. An XABSL behavior specification is comprised by a set of behavior modules called options and a set of different simple actions called basic behaviors. Each option consists of various numbers of states or subordinate options. Each state has two parts of information, decisions and actions. Decisions describe the conditions whether to jump out or stay in the current state according to the input variable, while the actions consist of the outputs such as the basic behaviors, LEDs etc.

6 Gait Planning

We have developed gait planning for our robot in both passive and active way. The implementation of passive forward walking is applying Virtual Slope Walking in the sagittal plane with the Lateral Swing Movement for lateral stability[1]. The forward walking speed of THU-strider is 0.6cm/s. The sideward walking and turning is realized by carefully designing the key frames. The entire above gait is generated by connecting the key frames with smooth sinusoids. The design of active gait, including kicking gait and small steps, is based on the ZMP theory. The process can be briey divided into three steps: rstly, set the position and speed of a part of motors with a preset motion; secondly, set the desired ZMP trajectory, by which a close form restriction of all motors would be attained; nally, solve those restriction functions to get the position and speed of the other motors. The result is the speed and position of each motor. To actualize this algorithm naturally, the kicking movement is divided into several parts, then each part is divided into many frames to be executed. A Matlab program is designed to do the calculation frame by frame. After the calculation, one feasible strategy of kicking movement based on ZMP will be obtained as a matrix, each line records the position of motors at a certain time.

Gait planning is mainly done by the ROS program running on the NUC. When we get basic motion requests from a higher inhierarchy module, motion is translated into instructions for each joint actuator. Instructions keep being sent out frame by frame via Singleboard-RIO 9606 to every motor so as to make robot move as expected.

7 Conclusion

Our AdultSize robot THU-Strider is a completely autonomous humanoid robot, with 1 camera, 1 gyro and 18 actuators integrated on body, controlled by a NUC-SingleBoardRIO system. In this paper we present the specifications and functions of the THU-Strider, as well as some related works on vision, localization, gait planning and control.

Tsinghua Hephaestus commits to participate in RoboCup 2014 in Brazil and to provide a referee knowledgable of the rules of the Humanoid League.

References

1. M. Zhao, J. Zhang, H. Dong, Y. Liu, L. Li and X. Su, Humanoid Robot Gait Generation Based on Limit Cycle Stability, *In the proceedings of the RoboCup Symposium* 2008., LNAI 5399, pp. 403-413, 2009.
2. M. Zhao, H. Dong and N. Zhang, The instantaneous leg extension model of Virtual Slope Walking, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3220-3225, Oct. 2009.
3. H. Dong, M. Mingguo Zhao, J. Zhang andN. Zhang, Hardware design and gait generation of humanoid soccer robot Stepper-3D, *Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 828-838, 2009.
4. H. Dong, M. Zhao and N. Zhang, High-speed and energy-efficient biped locomotion based on Virtual Slope Walking, *Autonomous Robots*, vol. 30, no. 2, pp. 199-216, Jan. 2011.
5. Y. Cai, M. Zhao, Z. Shi and W. Xu, "An Experimental Comparison of Localiza tion Methods in Robocup Four Legged League", *Robot*, 2007, Vol. 29, No. 2, pp167-170.
6. Z. Liu, Z. Shi, M. Zhao and W. Xu, Mobile robots global localization using adaptive dynamic clustered particle filters, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1059-1064, 2007.
7. Z. Liu, M. Zhao, Z. Shi and W. Xu, Multi-robot Cooperative Localization through Collaborative Visual Object Tracking, *In the proceedings of RoboCup 2007 Symposium*, LNAI 5001, pp. 41-52, May. 2008.