AUTMan Humanoid TeenSize Team Description Paper

RoboCup 2016 Humanoid Robot League, Leipzig, Germany

Soroush Sadeghnejad[1], Jacky Baltes[2], Sepehr Ramezani[1], Mohammad Karimi[1], Ariyan Valaei[1], Mohammad Javadi[1], Jamillo Santos[2], Tikwai Poon[2], Behnam Yazadnkhou[1] , Danial Alihosseini[1] , Mohsen Seyyed Hosseini[1], Sepehr Behjou[1]

[1]*Bio-Inspired System Design Lab*
Mechanical Engineering Department,
Amirkabir University of Technology (Tehran Polytechnic),
No. 424, Hafez Ave., Tehran, Iran. P. O. Box 15875-4413.
http://autman.aut.ac.ir

[2]Autonomous Agents Laboratory
University of Manitoba, Winnipeg, Canada, R3T 2N2
http://aalab.cs.umanitoba.ca

Abstract. This document introduces AUTMan (**A**mirkabir **U**niversity of **T**echnology and University of **Man**itoba), a joint team participating in the Humanoid League Teen Size competition at RoboCup 2016 in Leipzig, Germany. This team was founded in 2013 and is a collaboration between two teams from Amirkabir University of Technology (Tehran Polytechnic), Iran and University of Manitoba, Canada. Our humanoid Teen Size research is mainly based on both universities' previous experiences in participating in the RoboCup humanoid league. A brief history of Team AUTMan and its' research interests and directions for future research are given in this paper. Our main research interests within the scope of humanoid robots are: (a) design of different basic postures for walking gaits, (b) push recovery and active balancing, (c) running robots, (d) accurate localization, and (e) strategic reasoning for soccer.

Keywords. RoboCup 2016, humanoid joint team, walking gaits, active balancing and push recovery, running robots, localization, strategic reasoning.

1 Introduction

Challenge problems involving sporting events have been an important part of evaluating performance in robotics and AI for more than twenty years. More recently, it has become a major focus for humanoid robotics, since advances in hardware and software, decreases in associated costs, and the goal of having robots function in domains intended to be inhabited by humans have made humanoid robots spurred research in this area. Consequently, a variety of humanoid robot competitions have become prominent.

The highest profile competitions are soccer playing robots. RoboCup is an important challenge to promote artificial intelligence, machine learning, and system design technologies. RoboCup is pursuing the goal: "By the year 2050, develop a team of fully autonomous humanoid robots to win against the human world cup champion team" [1]. With the introduction of the Humanoid Robot League in 2002, RoboCup tries to promote the development of useful and practical humanoid robots for society. Rules are by updated every year to guarantee that the competitions provide meaningful benchmark results for research as well as remain entertaining for participants and spectators. Improvements in the hardware and software of the robots led to a push towards larger humanoid robots in more complex environments by the humanoid league community [2]. This move towards the Teen and Adult sized sub-leagues of the RoboCup humanoid league, causes issues for teams, especially new teams. Developing a team of at least 11 fully capable AdultSize humanoid robots to achieve the RoboCup goal in 2050 is unrealistic for many teams due to financial, organizational, human resources, and maintenance issues [3]. The humanoid league community feels that the establishment of new rules and by encouraging teams towards forming joint teams are the ways of promoting this goals. The cooperation of Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran and University of Manitoba, Winnipeg, Canada, is one of the most successful joint teams. Despite having several previous initiatives directed at suitable cooperation protocols between different teams, there are several valuable lessons for both teams which are crucially important for the future of the humanoid league. Besides having difficulty in coping with suitable infrastructures, we believe that such collaboration is vital for the future of RoboCup. Due to this facts, we continue the cooperation for our team in 2016, and briefly explain about the future of our joint team in the next sections.

2 Recent Achievements

Team AUTMan Teen Size participated with great success in previous competitions. In RoboCup 2015, Hefei, China, the team placed 3rd in the general TeenSize League and at the same time placed 2nd in the Technical Challenge together with Team WF Wolves & Taura Bots, a joint team from Germany and Brazil. In RoboCup IranOpen 2015, Tehran, Iran, our joint team placed 1st in the Technical Challenge. Also in RoboCup 2014, Joao Pessoa, Brazil the team placed 1st in the Technical Challenge together with Team Nimbro from University of Bonn, Germany. Also the team won the 3rd Place in the Humanoid Teen Size Robot Sprint Challenge in International FIRA RoboWorld Cup 2014, Beijing, China. Both teams also extensively published their research results [1, 2, 4, 5].

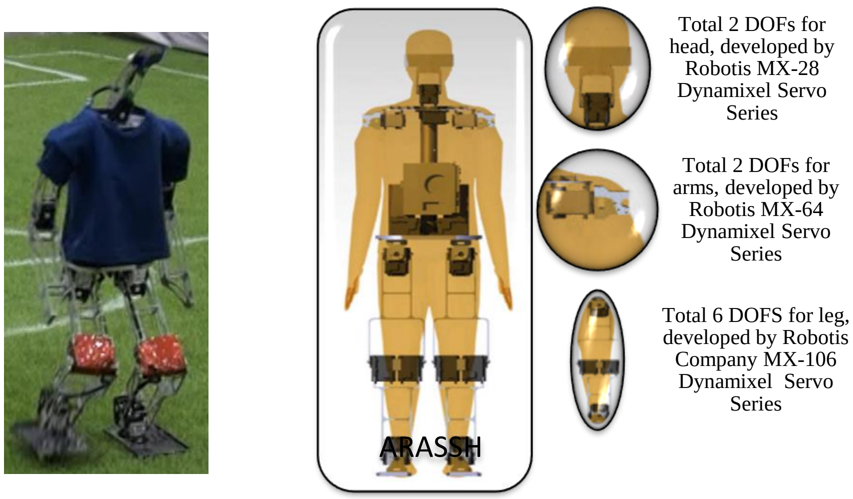
3 Development for 2016

In 2016, the goal of the collaboration between these teams and also participation in RoboCup 2016 is three-fold like previous year. First, the implementation of improvements for our teen sized humanoid robots framework and software by combining the extensive experiences from both teams. The main research thrust for the new robot design is to focus on designing of different basic postures for walking gaits, push recovery and active balancing by careful studying of different methods. The other goal is to continue to demonstrate the feasibility of joint teams in RoboCup and the third one is to modify our previous TeenSize humanoid platform robot.

4 Hardware Design

We will use our previous year developed robot, ARASH, with some modifications. An Anthropomorphic Robot Augmented with Sense of Human (ARASH) is a robust robot with 100cm in height and 7.5Kg in weight which we focused on weight reduction and also nimble reaction features. We designed the robot based on the anthropomorphic data, collected from different human. ARASH has 20 degrees of freedom (DOF) in total. Figure 1 shows the distribution of these degrees of freedom in the robot total structure. As explained in detail in [3], to reduce the risk of robot performance and to increase the robot wide range of motion and fast reaction and to reach to a higher performance, we used two Dynamixel MX-106 servo series in the knee joints.

To control Dynamixels servo actuators and getting sensory information fusion of inertial measurement unit, the OpenCM9.04, an open-source controller with a 32bit ARM Cortex-M3 from ROBOTIS.Co [6], is utilized as a low level controller and as Device Communication Manager (DCM). It communicates with the upper layer controller (PC) using a 1 MBps serial interface. Communication of controller and actuators is based on Half-Duplex TTL serial communication at 1Mbps.



(a) (b)

Figure 1. ARASH, (a) general overview, (b) total robot structure degrees of freedom distributions

The inertial measurement unit (GY-80 [7] IMU) includes internal 3 axis gyro, 2 axis accelerometer and 3 axis magnetometer sensors and it provides 3D posture of robot's center of mass (COM) in real-time. Free RTOS real-time operating system [8] is used to execute different tasks. In the new version of ARASH, a MAXData QutePC-3001 [9] mini embedded board is used as the main controller. The high performance and low power consumption are the important factors for using this kind of main boards, as a main processor in humanoid robots. A 1inch lcd is used to monitor some parameters of robot.

5 Software Development

This section described the main software components of our robot.

5.1 Software Architecture

The team's software architecture is based on ROS [10] and has focused on the modularity of the system and is developing a high-level framework as a ROS package. In 2016, we have gone even made significant progress in the development of our ROS framework. Another focus of this framework is the modularity and ease of access for other humanoid teams. So we have developed shell scripts and commands to help the user of the open framework focus on the algorithm or the framework of the code.

5.2 Remote Robot Control

Formerly, all the connections with the robots have been established using a simple SSH connection and all the logging and simulation processes were forced to be run on the robot while the robots are on the field. This is simple to implement but puts a heavy processing load on the main computer. So all these process are moved consuming tasks to the remote pc and the robots only handles the network connection while running. With this approach the remote control of the robot is more robust and also GUI-based.

5.3 Robot Connection

With improvement of Localization and Behavior nodes, the perception of teammate robots are used to increase the validity of the perception of each robot. Furthermore, the online logging of the robots require reliable UDP connection package. So, a new Robot Connection package is developed for reliable connection even in weak network coverage which is not far from mind in RoboCup matches.

5.4 Motion

The walking method is based on the linear inverted pendulum model (LIPM) [11], [12]. Utilizing this method, the three dimensional position of the COM of the robot in two phases (single support and double support) are calculated. Meanwhile, the path of the swinging foot in the single support phase in Z (upward) and X (forward) directions are estimated by trigonometric and polynomial functions of time respectively, whose general forms are as follows:

$$Z(t) = Asint + Bcost + C$$
$$X(t) = A_n t^n + A_{n-1} t^{n-1} + \ldots + A_1 t + A_0 \quad (1)$$

The coefficients of each function, and consequently the degree of the polynomial for forward motion, are determined based on the desired characteristics of the gait or, in other words, based on the number of assumptions (known positions, velocities and accelerations of specific points) which constitute a particular path. The actual position of each motor is then obtained using inverse kinematics. It is noteworthy that the stability conditions of the robot have also been included in the inverted pendulum formulations. Therefore, stability could be further achieved by designing a proper double support phase and also by suitably tuning the various walking parameters. Next, the above mentioned algorithm is generalized to other directions so that the walking of the robot turns into an Omni directional one. Additional intricacies, such as push recovery, are also taken into account using different strategies. To ensure the reliability of the proposed gait pattern, a model of the robot has been simulated in a SimMechanics environment of MATLAB software [13], which is illustrated in Figure 3. The foot, shin, thigh and center of mass of the robot, distinguished by different colors, have been modeled in this simulation. Mechanical characteristics of the robot, like types of joints, types of links and controllers of motors, have all been accurately modeled so that the simulation can be conveniently utilized and updated both for the current work and for future augmentations and also for other works which are based on LIPM. A feature which is going to be added to the walking strategy is making it a closed-loop pattern by utilizing feedback from the IMU and motors. This will help the robot have a better control of its stability in comparison with an open-loop pattern in different situations such as on an uneven terrain. Also unlike the walking program of last year which had been implemented on a microcontroller [11], the walking program is going to be transferred to a PC, along with other tasks (such as vision, behavior, etc.) for RoboCup 2016.

5.5 Cognition

As it was explained previously and in [3], our cognition module processes input in several layers. In the first step we construct a lookup table by selecting random pixels based on our desired color labels.

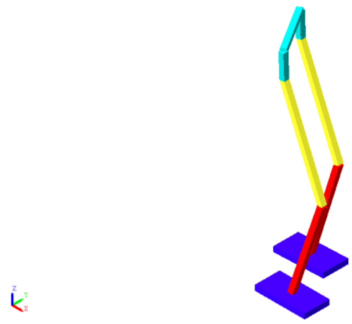


Figure 2 a robot model generated in SimMechanics environment of MATLAB software

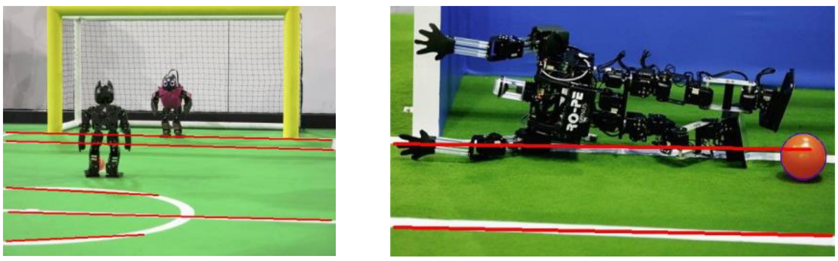
Then each pixel is labeled to its corresponding color. Afterward the vertical scan is done and the result would be a segmented picture with vertical rectangles, where each of them belongs to a specific color class.

Regioning: The main purpose of this step is to make shapes out of the found segments. Actually segments which have potential to construct a same shape would be connected. Segment connection is done based on criterions such as same color label, distance and length of each rectangle. The others and the individual segments would be omitted.

Edge detection: In this step, we extract edge spots from the image, based on the spot's edge magnitude and gradient. Since it was not efficient to apply the edge detection algorithm to the whole image, we applied our implemented edge detector on the coordinates of shapes, explained in the previous step.

Feature Extraction and Object Recognition: In this step, region features are determined by shape and color of the regions which were explained completely in [3]. Since the RoboCup rules have been changed our color-based algorithms were not efficient by their own anymore. So beside our previous implementations, we also implemented edge-based algorithms such as CHT and LHT, optimized with our new ideas which are used in involved cases for a more accurate result (Figure 4).

Localization: As humans model their environment, our robots must model their environments to be aware of it and make appropriate decisions in a timely manner. So we use localization module to provide a model of environment. Using perception data provided by vision we can use relative distances to our self in order to find our location in field based on statistical analysis. As is mentioned, the localization module just provides a model and send it to behavior module for decision making; so we must put all object specifications in this model and send it to the behavior module. This message is called world model. This objects can be lines, obstacles, goal, ball and so on. Specifications can be locations relative to the center of the field, angle, velocity and etc. Like many other teams, we use statistical methods too in order to estimate the location and other specifications of an object. In order to have more precise estimation, many filters are applied such as Monte Carlo, Unscented Kalman Filter and Particle Filter. Since this filters are time consuming, so we are working on methods to make these filters quicker. At last our created model is sent periodically to other robots in field; so each robot have a map merger module that is run every time that a map is received.



(a) (b)

Figure 3 Line and ball detection using edge-based methods

A copy of this created model is sent to the simulator module; so as is described in previous sections this data is sent to the network to have remote simulation on the client.

5.6 Behavior

This behavior module determines how to achieve high level goals in our architecture. As mentioned before, the localization module sends the behavior module a model of environment and the behavior module uses this data to make proper decisions. In order to have simplicity and independent nodes, this module is divided into three layers: (a) the high level decision maker, (b) the decision executor and monitor, and (c) the velocity controller.

High level decision maker: The highest level in the behavior module, just sends commands to the lower layer. This layer consists of state machines which are divided into two nodes. One of them controls the body and the other one controls the head. The body state machines implement behaviors for ball finding, goal finding, etc. For all of these commands, we need a path to get to that location. So we use a path planner to create a path based on the model received from the localization module and that plan is sent to the decision executor and monitor layer to track its execution. The control of the head is based on attention control. Sometimes we need to urgently update location of the ball, whereas other times we want to update our location. In order to update the required data, this node orders commands and sends them to the lower layer. All of these behaviors are different for different roles; in other word the states are role based. By combination of these states we create our desired tactics. This layer has another node which is communicating with other robots and may request a role change from other robots.

Decision executer: This layer executes the commands received from the upper layer. This layer consists of two ROS nodes (head and body) similar to the high level decision maker. This layer make decision for the speed of the robot in x, y and theta and creates strategies for head movement to achieve the goals introduced by higher layer. We have another node for object avoidance. This node is aside the body and head node in the decision layer. We are sensitive to objects near us in this node and if we detect an object in front of us, we ignore the body velocity commands and make proper decisions to avoid objects.

Velocity controller: This layer is the lowest layer in the behavior module and we use it to smooth out the body and head trajectories. A PID controller is used in order to accomplish this task.

6 Conclusion and Acknowledgment

This report described the technical plans and works done by the AUTMan Humanoid Teen Size Robot Joint Team for its entry in the RoboCup 2016 Humanoid Teen Size League. AUTMan team's focus, for the second year of RoboCup competition, has been on modifying our Teen Size Platform, localization, motion behavior, and vision module due to our past and relevant experience in various RoboCup leagues. We look forward to continuing and expanding our above research with the new humanoid robots. For further information, please refer to our publications. The reader can also find more pictures and videos of our robot on our official websites.

REFERENCES

1. Gerndt, R., Seifert, D., Baltes, J.H., Sadeghnejad, S., Behnke, S.: Humanoid Robots in Soccer: Robots Versus Humans in RoboCup 2050. Robotics & Automation Magazine, IEEE 22, 147-154 (2015)

2. Shangari, T.A., Shamshirdar, F., Heydari, M.H., Sadeghnejad, S., Baltes, J., Bahrami, M.: AUT-UofM Humanoid TeenSize Joint Team; A New Step Toward 2050's Humanoid League Long Term RoadMap. Robot Intelligence Technology and Applications 3, pp. 483-494. Springer (2015)

3. Sadeghnejad, S., Baltes, J., Ramezani, S., Karimi, M., Karimi, M., Valaei, A., Javadi, M., Ahmadi, A., Hosseinmemar, A., Santos, J.: AUT-UofM Humanoid TeenSize Team Description Paper.

4. Shangari, T.A., Shamshirdar, F., Azari, B., Heydari, M., Sadeghnejad, S., Baltes, J.: Real-time Ball Detection and Following Based on a Hybrid Vision System with Application to Robot Soccer Field.

5. Baltes, J., Sadeghnejad, S., Seifert, D., Behnke, S.: RoboCup Humanoid League Rule Developments 2002–2014 and Future Perspectives. RoboCup 2014: Robot World Cup XVIII, pp. 649-660. Springer (2015)

6. Ha, I., Tamura, Y., Asama, H., Han, J., Hong, D.W.: Development of open humanoid platform DARwIn-OP. In: SICE Annual Conference (SICE), 2011 Proceedings of, pp. 2178-2181. IEEE, (Year)

7. Dharmawan, A., Putera, C.A.L.: Purwarupa Sistem Integrasi Quadcopter dan Mobile Robot. IJEIS-Indonesian Journal of Electronics and Instrumentation Systems 2, 97-108 (2012)

8. Kulkarni, S.S., Padgilwar, R.S.: Automation over Internet & Android app using RTOS. IJCER 3, 43-48 (2014)

9. Hennerkes, W.A.: MAXDATA Main Menu. Maxdata, pp. 27-27. Springer (1990)

10. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source Robot Operating System. In: ICRA workshop on open source software, pp. 5. (Year)

11. Tamiz, M., Bararian, E., Shangari, T.A., Karimi, M., Shamshirdar, F., Heydari, M.H., Jegarian, M., Parizi, A.T., Jannatifar, M., Iravani, E.: AUTMan Humanoid KidSize Team Description Paper.

12. Kajita, S., Matsumoto, O., Saigo, M.: Real-time 3D walking pattern generation for a biped robot with telescopic legs. In: Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, pp. 2299-2306. IEEE, (Year)

13. Shaoqiang, Y., Zhong, L., Xingshan, L.: Modeling and simulation of robot based on Matlab/SimMechanics. In: Control Conference, 2008. CCC 2008. 27th Chinese, pp. 161-165. IEEE, (Year)