

RoboFEI@Home 2016 Team Description Paper

Andrey A. Masiero, Douglas de Rizzo Meneghetti, Leonardo Contador,
Lucas Vasconcelos, Flavio Tonidandel and Plinio T. Aquino Junior

Centro Universitário FEI,
Av. Humberto Alencar de Castelo Branco, 3972, 09850-901, SBC, SP, Brazil
<http://www.fei.edu.br/robofei>, amasiero@fei.edu.br

Abstract. This paper presents the description of the RoboFEI@Home team as it stands for the Robocup 2016 in Leipzig, Germany. It describes all mechanical, electrical and software modules, designed for RoboFEI's robot, which is called Judith, a robot based on Peoplebot platform. Judith was built to attempt Human-Robot Interaction (HRI) on social scenarios, which is the main research of our team. A framework for behavior analysis and adaptation have been constructed in the last year to be implement on Judith and many other robots.

Keywords: Robocup@Home, RoboFEI, Peoplebot, Autonomous Robot

1 Introduction

This paper describes the hardware and software aspects of the RoboFEI@Home team, designed to compete in the RoboCup 2016 @Home League.

The group has a long tradition in Robotic Soccer and the first time we took part in a competition was in 1998, when Prof. Reinaldo Bianchi was a member of the group that developed the FutePOLI Team, which competed in the First Brazilian Micro Robot Soccer Cup, held in São Paulo, Brazil. Since then, Bianchi started the development of soccer playing robots at the Centro Universitário FEI, competing in the Very Small Size category and after, in the Very Small Size league. In 2004, during the First Brazilian Competition on Intelligent Robots (a competition where the IEEE Robot Competition was held together with the First Brazilian RoboCup), our team became Brazilian Champion in the IEEE Very Small category. In the same competition, our first team developed for the RoboCup Small Size League became vice-champion. In 2006, Very Small Size team became Champion again, and our first 2D RoboCup Simulation team was the Brazilian champion.

The institution RoboCup Small Size League team, called RoboFEI, won for the first time the Brazilian Robocup in 2010, and is currently the Brazilian champion, winning the championship 4 times in a row (2010, 2011, 2012 and 2013). This team takes part in the RoboCup World Competition since 2009, and the best result we had was in 2012, when we stayed among the 8 top teams. After developing robotic soccer players for the last 17 years, we developed a

team to compete in the RoboCup Humanoid League. The development of this team started in 2012, with students designing and building a humanoid robot from scratch. Last year, the RoboFEI-HT team competed our first RoboCup World Competition, held in João Pessoa, PB, Brazil, with 4 humanoid robots: two Newton (developed in the institution, being the pieces of one of them, made in a 3D printer) and two humanoid robots based on Darwin-OP. At RoboCup 2014, the team stayed among the 16 top teams and in the same year, the team competed the Latin American Robotics Competition (LARC 2014) and became champion in the LARC RoboCup Humanoid Kid Size league.

Now, the institution has establish important positions on small size and humanoid league, we decide to drive our researches into new paths for collaborating on social and service robots. Last year, we applied for Latin American @Home competition and we won the 3rd place. We focus on researches applied in Human-Robot Interaction, Social and Service Robots, Behavior Analysis, People Modeling and Profiling, and all questions for improving people and robots coexistence.

2 Hardware Design

Our team has one robot, called Judith (see Fig. 1). Judith is a robot based on Peoplebot robot [1] and KUKA youBot arm [8]. As shown in Fig. 1, Judith has: 1 Microsoft Kinect for Xbox 360; 2 Sonar Arrays; 2 Infrared Sensors; 1 iPad 2; 1 Microphone Yoga HT-320A; 1 Touch Sensors Array; 2 Speakers; 1 Emergency Button; and 1 Hokuyo Laser URG-04LX-UG01.

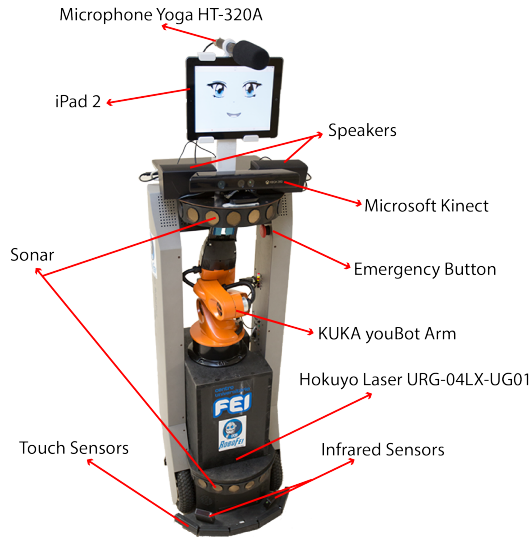


Fig. 1. RoboFEI's Judith

Working with two different ready robots present some challenges. First of all, each one has electric source with different voltage. KUKA youBot arm needs 24v to work while Peoplebot works with 12v. To put Peoplebot and KUKA youBot arm together, we need to build a circuit for connecting both batteries kit together. This circuit is important to keep all parts of robots safe in case of electric damage on overcharge. It also helps to make both robots work at the same time.

Peoplebot has a vertical grip which needs to be removed for attaching KUKA youBot arm. To attach the new arm, we built a support for keeping it on the right height to manipulate objects on the table. KUKA youBot arm weighs 16.53 pounds so we made a wood support to hang up with this weight. With all mechanical and electrical project setup, we need to connect both robots on main computer.

Here we face another problem, because our computer has only USB ports as a device input options or Ethernet and WiFi connections. Peoplebot can interface with a serial port RS232 and KUKA youBot arm interfaces with a real-time EtherCAT. To solve this problem, we use EtherCAT connection for KUKA youBot arm, and for Peoplebot platform needs a RS232 to USB converter for helping interface with computer.

As main computer we have been using an ASUS Ultrabook 14" Touch-Screen Laptop with Intel Core i5, 4GB Memory, 500GB Hard Drive. It has only 3 USB ports supporting all devices. To enable more ports, we have used a USB hub to increase number of devices interfaced. As iPad need a annual license to develop programs, we are changing for an Android tablet which allow us to develop a interactive face for our robot Judith.

Platform used by Peoplebot turns the transportation very difficult for us. In that way, we have worked on a module platform using 3D printer and aluminum parts. We want to accomplish this project until Robocup 2016 in Germany.

3 Software Design

This section present all details about RoboFEI@Home software. A stable version of the software is available on GitHub¹. Follow sections are divided as: (I) Software Architecture of Judith; (II) Face and People Recognition; (III) Object Detection; (IV) Speech Recognition and Synthesize; (V) Navigation; (VI) Object Manipulation.

3.1 Software Architecture

All Judith software is developed using Ubuntu Linux 14.04 LTS [20] and ROS Indigo Igloo [16]. The initial architecture of Judith is composed by three layers. Each layer is responsible for one part of the code. First layer receives all data from sensors, extract the features and publish it for AI and Controller nodes

¹ https://github.com/OpenFEI/rfh_judite

(Second Layer). AI and Controller nodes take the features and run algorithms for computer vision, location, planning, pattern recognition, among others. In the end, second layer send to third layer what is the velocity of the motors, position of manipulator or face express by robot. Third layer controls all the actuators drivers and actions. A graphic view of the architecture is presented on Fig. 2.

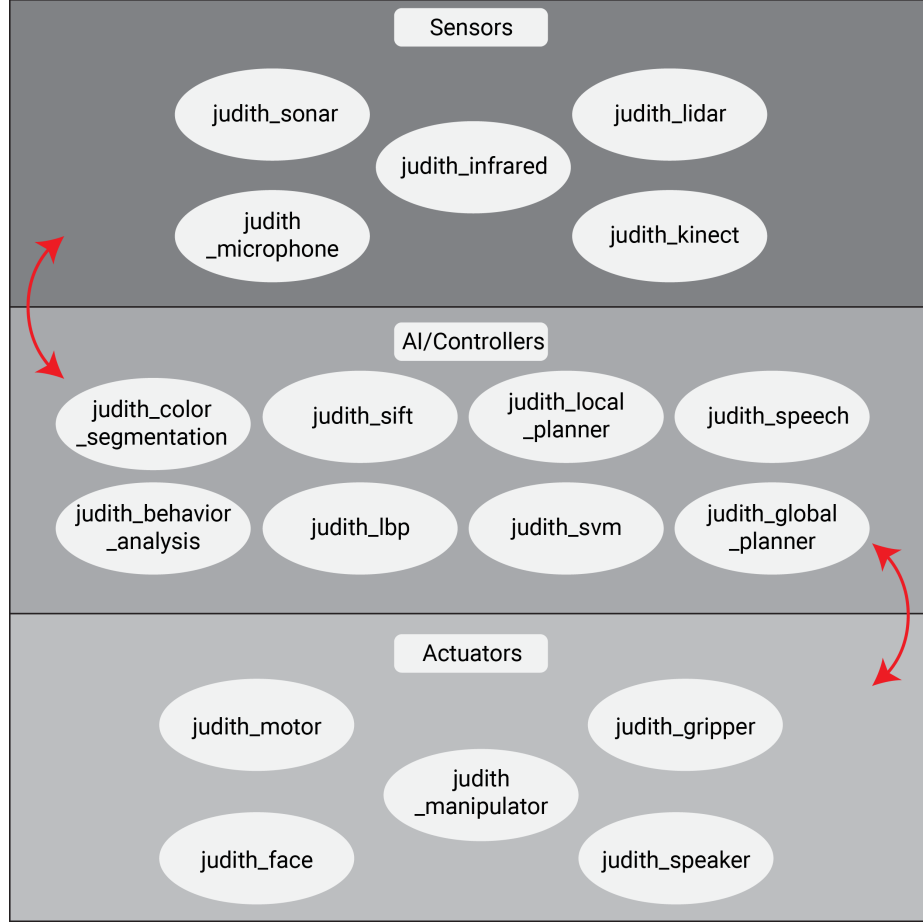


Fig. 2. Software Architecture of Judith

Sensors and Actuators layers has been developed using C++ [21], due to hardware drivers communication, except by `judith_face` node. Face's node is developed in Java [12] language to create an application on Android [11] Platform. AI/Controllers layer is developed using Python [22] as programming language to

all algorithms. All decisions made for the best performance of our robot during task execution and also for training new members to the team.

3.2 Face and People Recognition

We understand when you recognize a person and treat it by the name, you make the interaction more comfortable for her/him. In that way, we develop some nodes that can detect and recognize a person by their face. To make it possible, we acquire an image using a camera (Webcam Logitech C920 or Microsoft Kinect X360) and convert the frame into a OpenCV [5] Image Matrix. This information is a input of the LBP algorithm which has been widely used for face recognition due to its computational performance [2, 23, 19, 24, 18]. LBP algorithm may also help on identify gender, age and facial expression which can be useful for making an adaptive behavior of the robot during interaction.

Beyond facial recognition this set of nodes is responsible for following a single person, without external interference during process. To perform this task, we decide to combine three techniques. First is face detection followed by extraction of the color from person t-shirt [17, 14, 4]. With the t-shirt color defined, we perform a color segmentation using OpenCV [13, 15, 6]. The last technique is to extract a skeleton position using Nite/OpenNI library for Microsoft Kinect [9]. With this information robot can follow the direction of a certain person and also to identify the distance between them.

3.3 Object Detection

3.4 Speech Recognition and Synthesize

Speech is one of the most important way to interact with a human. Due to that, we use CMU Pocketsphinx [10] to make Judith's speech recognition and also synthesize a female voice on our robot.

3.5 Navigation Stack

To execute tasks on environment the robot needs to know where it is and how to arrive into another point of the place. The first step is create an environment map. For performing this task, we use GMapping and SLAM to create the map with laser Hokuyo URG-04LX-UG01 helps. We save the map bag using a node from ROS called map_server.

After that, we run the bag saved and execute the algorithm *Adaptive Monte Carlo Localization* (AMCL) [7] with data from the laser make the robot capable to locate its position on the map. With position known, we can give the robot a point to go on the map. Next step is to use an algorithm A* to make the route between current position and desire position. After that, for each movement of the robot, it executes an *Dynamic Window Approach*(DWA) algorithm for planning the current position movements.

3.6 Object Manipulation

For manipulating an object, we first use object detection node (see sec. 3.3) to know its current position. Then we execute a few movements preview determined to put the gripper on robot view. After that, we use PCL algorithm [3] to segment the gripper and get object distance. At the end we execute planning to move KUKA youBot arm joints for getting closer and grab the object.

4 Conclusions

We have described all development of our robot Judith. We believe that all of these tasks can help on home, health and office services. To improve the acceptance of these robots, we have been working into a behavior analysis framework developed using ROS. With this framework, we pretend:

- Identifying people’s emotion;
- Creating user behavior profile;
- Improving approach to start a interaction.

With this TDP, we hope to participate on Robocup 2016 in Germany and try to advance during the competition and learning with all teams, as we did to win third place on XIV Latin American Robotic Competition in 2015. Hope see you all there.

References

1. Activmedia Robotics, L.: Pioneer 2/peoplebotTM. Operations Manual, Version (2001)
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28(12), 2037–2041 (2006)
3. Aldoma, A., Marton, Z.C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S., Vincze, M.: Point cloud library. *IEEE Robotics & Automation Magazine* 1070(9932/12) (2012)
4. Baggio, D.L.: Mastering OpenCV with practical computer vision projects. Packt Publishing Ltd (2012)
5. Bradski, G., et al.: The opencv library. *Doctor Dobbs Journal* 25(11), 120–126 (2000)
6. Culjak, I., Abram, D., Pribanic, T., Dzapo, H., Cifrek, M.: A brief introduction to opencv. In: *MIPRO, 2012 Proceedings of the 35th International Convention*. pp. 1725–1730. IEEE (2012)
7. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI* 1999, 343–349 (1999)
8. Group, K.R.: KUKA youBot Robot. <http://www.youbot-store.com/> (2016), [Online; accessed 29-January-2016]
9. Guide, O.U.: Openni organization, november 2010. Last viewed 18, 15 (2011)

10. Huggins-Daines, D., Kumar, M., Chan, A., Black, A.W., Ravishankar, M., Rudnicky, A.I.: Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on. vol. 1, pp. I-I. IEEE (2006)
11. Inc., G.: Google Android. <http://developer.android.com/intl/pt-br/google/index.html> (2016), [Online; accessed 9-February-2016]
12. Joy, B., Steele, G., Gosling, J., Bracha, G.: Java (tm) language specification. Addison-Wesley, June (2000)
13. Kang, S.K., Nam, M.Y., Rhee, P.K.: Color based hand and finger detection technology for user interaction. In: Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on. pp. 229–236. IEEE (2008)
14. Laganière, R.: OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision. Packt Publishing Ltd (2011)
15. Oliveira, V., Conci, A.: Skin detection using hsv color space. In: H. Pedrini, & J. Marques de Carvalho, Workshops of Sibgrapi. pp. 1–2 (2009)
16. Open Source Robotics Foundation, O.: ROS Indigo Igloo. <http://wiki.ros.org/indigo/Installation> (2015), [Online; accessed 9-February-2016]
17. Pulli, K., Baksheev, A., Korniyakov, K., Eruhimov, V.: Real-time computer vision with opencv. Communications of the ACM 55(6), 61–69 (2012)
18. Samadi, F.: Human age-group estimation based on anfis using the hog and lbp features. Electrical and Electronics Engineering: An International Journal (ELEIJ) 2(1), 21–29 (2013)
19. Shan, C.: Learning local binary patterns for gender classification on real-world face images. Pattern Recognition Letters 33(4), 431–437 (2012)
20. Sobell, M.G.: A practical guide to Ubuntu Linux. Pearson Education (2014)
21. Stroustrup, B.: The C++ programming language. Pearson Education India (1986)
22. VanRossum, G., Drake, F.L.: The Python Language Reference. Python software foundation Amsterdam, Netherlands (2010)
23. Yang, Z., Ai, H.: Demographic classification with local binary patterns. In: Advances in Biometrics, pp. 464–473. Springer (2007)
24. Ylioinas, J., Hadid, A., Pietikainen, M.: Age classification in unconstrained conditions using lbp variants. In: Pattern Recognition (ICPR), 2012 21st International Conference on. pp. 1257–1260. IEEE (2012)