UTBots@Home 2020 - Pionner 3-AT current and future developments

Felipe Pierre Conter[1], Jonathan Cerbaro[2], Dieisson Martinelli[2],
Marlon de Oliveira Vaz[3], João Alberto Fabro[1] and André Schneider de Oliveira[2]

***Abstract*— This TDP describes the hardware and software solutions currently applied to allow the Pioneer robot from UTFPR to execute domestic tasks related to the Robocup@Home initiative. The Pioneer 3-AT robot (a.k.a. "Apollo") has sensors, such as a laser scanner and a Kinect 3D sensor, that provide information necessary for navigation, object recognition, environment mapping and auto-localization. An innovative human-robot interaction interface, using different facial expressions based on simulated emotions, is also presented. The robot is capable of executing some Robocup@Home tasks, such as voice synthesis and recognition, object recognition and navigation. All the programming uses ROS (Robot Operating System), so that the same control programs can be executed in both robots with minimum alterations. Also, we present what is planned for this year edition, which will happen remotely, and the improvements planned for the future.**

I. INTRODUCTION

Robocup@Home is a competition that aims to foster the development of robots that will help people in domestic environment, being the biggest competition for service robots on the planet [1][2]. In order to perform household tasks, the robotics fields of human-robot interaction, navigation, map construction on dynamic environments and computer vision are developed, among many others [1].

To accomplish these tasks, a Pioneer research robot is being used: a Pioneer 3-AT (a.k.a. "Apollo"), presented in Figure 1. The applications are running with ROS (Robot Operating System) and a set of software libraries and tools that help to develop applications for robotics [3]. The control system of the robots is responsible for acquiring and interpreting various sensors and, from that data, make decisions to fulfill tasks.

In this document, the partial results and expectations of projects under development for the robot is presented. Section 2 presents a description of the robot's hardware, including specifications and characteristics. The software already developed, and what is still in development, is described in section 3. Conclusions and future perspectives are presented in section 4.

[1]J. A. Fabro is a professor with the Graduate Program on Applied Computing - PPGCA, at the Informatics Department - DAINF - at UTFPR - Federal University of Technology, Parana, Brazil - Campus Curitiba. F. P. Conter is a master student within this Graduate Program. fabro@utfpr.edu.br, felipeconter.cc@gmail.com

[2]A. S. de Oliveira is a professor with CPGEI - Graduate Program of Electrical Engineering and Industrial Informatics, UTFPR, Brazil - Campus Curitiba. J. Cerbaro and D. Martinelli are master students within this Graduate Program. andreoliveira@utfpr.edu.br, jonathancerbaro@alunos.utfpr.edu.br, dmartinelli@alunos.utfpr.edu.br

[3]M. O. Vaz is a professor with IFPR - Federal Institute of Parana, Campus Pinhais, Pinhais. He is also a doctorate student in CPGEI. marlon.vaz@ifpr.edu.br

II. HARDWARE

A. Robot Specifications

The robot is based on a Pioneer 3-AT research robot: it weights only 12 Kg, with 12 Kg of payload, and maximum speed of 0.7 m/s. Its autonomy is of 2 hours, with 3, 12 volts, standard batteries. Its dimensions are: 50 cm wide, 50 cm deep and 28 cm high. The robot connects to an external notebook via a serial-USB interface allowing its connection to a ROS system. Figure 1 presents this robot.

A redesign is being planned, and a Jetson TX1 [4] will be coupled to the robot, in order to process computer vision and other GPU-focused processes. The ideia is to integrate the Jetson TX1 to the ROS network on the external notebook. Later, the notebook can be replaced by a small sized computer and integrated into the robot design.

Fig. 1. Pioneer AT (a.k.a. "Apollo") with its support for the Kinect sensor, and the tablet that is used to present simulated emotions.

B. "Neck" Support

On the robot's surface is a support constructed of aluminium (as seen in Figure 1. This support is 1.2 meters long, allowing the positioning of the tablet and the Kinect sensor to a height comfortable enough to allow interaction with people (1.5 meters approximately).

C. Sensors

A Kinect depth sensor has been incorporated. With this sensor it is possible to create a depth image, identify silhouettes of persons and objects, among other functions [5]. In order to establish the communication between the Kinect sensor and ROS the freenect package was used. The data generated by Kinect and the laser sensor is published in topics in ROS, making the data available to algorithms used in this project. "Appolo" has a YDLIDAR X4, which is a low cost LIDAR sensor capable of 360 degrees distance measurements [6].

D. Manipulator

The original model of the manipulator used is a Beckman Coulter ORCA Robotic Arm, a planar manipulator that has three rotational joints and can be seen in Figure 2. To make the structure respond to the desired commands, a process of retrofitting was carried out, where the motors and encoders were connected in microcontrollers that allow the open programming of the mechanism. The robot is currently under remodeling and the addition of this arm to the Pioneer 3-AT robot is planned.

In addition to the three motors that define the degrees of freedom of the structure, each joint is also coupled to a quadrature encoder. This sensor emits digital pulses to the microcontroller when the joint is in motion, either by driving the motor or by the action of external forces.

The quadrature encoder emits two pulses slightly out of phase. This allows to infer the direction in which the motor is rotating: The microcontroller triggers an interruption every time a pulse is received by the encoder at output A. If output B is already at a high level, it means that the direction of rotation is clockwise. If output B is at a low level, then this pulse will occur shortly thereafter and therefore the direction of rotation is counterclockwise.

Joints 1 and 2, respectively related to the manipulator's shoulder and elbow, are controlled by an Arduino that has an H-bridge for each engine. Joint 3, related to the handle, and the gripper are controlled in the same way by another Arduino. Each joint has a PID control system that aims to maintain the angle of the joint in the desired position in opposition to external forces, in addition to allowing smoother movements between one position and another. Arduinos are connected to a single USB port, external to the mechanism, via a hub. Through this connection, ROS commands are received.

Customized ROS messages were implemented for sending and receiving robot information. While the microcontrollers receive the messages containing the desired angle in the topic of the respective joint, it sends messages allowing the monitoring of the pose of the structure.

Fig. 2. The manipulation arm mounted in another robot. The plan is to add this manipulator to the Pioneer 3-AT.

As the encoders do not have memory of the location where the joint stopped and also do not allow monitoring the movements that occur when the structure is off, the robot needs to perform a startup routine every time the system is powered. This makes all the joints assume an initial position and the control is done from these already known angles.

III. SOFTWARE

A. ROS - Robotic Operating System

The robot has a control notebook with the "kinect" version of ROS, on the operating system Ubuntu 16.04 LTS. ROS allows the application of several tools, libraries and conventions that simplify the development of complex and robust tasks for robots[3]. In this system, the core process, *roscore*, functions as a server that manages threads and nodes. Topics are data buses through which nodes exchange messages, and multiple nodes may subscribe to each topic (receiving its data) or publishing to it (sending data). Nodes are computational processes, that process data coming from some topics, and publish results to other topics.

B. Emotion Simulation

Emotions are simulated by the robot through faces, displayed according to the present situation, for a better visualization of internal states of the robot, and better interaction between the robot and people.

Emotions were based on Plutchik's wheel of emotions[7]. A tablet running the ROS Image Viewer application displays

Fig. 3. Examples of faces developed.

these faces by subscribing to a ROS topic. The emotions system is based on *ros_display_emotions* package [8]. Figure 3 depicts some of the faces developed to express different emotions during the interaction with people. Figure 4 illustrates an emotion being displayed by the robot through the tablet.

C. Simultaneous Localization and Mapping

Odometry errors can lead to uncertainty in navigation. It is important that the robot is able to correct its location based on the feedback from sensors in real time. SLAM (Simultaneous Localisation and Mapping) [9] algorithms can achieve this. Currently, the YDLIDAR readings and the robot's wheels odometry are feeding the standard ROS navigation stack and *move_base* is used to send navigation tasks to the robot.

By constructing a map of the environment at the same time as it is updating the robot's position, the robot estimates it's position and updates wheel velocity. To accomplish this, SLAM has a number of tasks: extraction of reference points, data association, state estimation, status update, and reference point update. Parameters adjustments and tests under different SLAM configurations are crucial for improving navigation.

Fig. 4. Simulated emotion presented on the tablet (face) of the robot.

D. Voice recognition and synthesis

For speech synthesis, a ROS package called *espeak_ros* is being used. This package works by running a ROS node that subscribes to the topic *speak_line*, and performs voice synthesis based on the specified parameters. The configuration of these speech parameters is done through the *dynamic_reconfigure* package, which allows you to change parameters on ROS nodes without having to restart them. The configuration of the speech parameters encompasses several characteristics, such as speech speed, volume, pitch (lower or higher), the space between words, age and intonation variation. By altering these parameters, the speech characteristics can be defined in accordance with the emotion displayed on its "face".

E. Object Recognition and Manipulation

Object recognition can be accomplished through different approaches. Simple techniques such as blob detection can help identify objects that have a well defined color and shape, with a strong difference in intensity from the background. In a simulated environment, with well defined objects such as plain colored cubes or spheres, segmentation of these objects was achieved through blob detection. However, since the spectrum of household objects varies in shape, colors, transparency, and many other characteristics, machine learning may perform better than traditional image processing techniques.

The latest experiment in the simulation environment V-REP [10] includes a table with a white cube on top of it, and a youBot robot, by automation company KUKA [11]. This robot was selected for the simulated experiments because of it's open platform, the presence of a manipulation arm, and a good model already available in V-REP. The scene in V-REP includes a 5x5 floor. The robot was modified by adding a laserscan Hokuyo URG-04LX-UG01 sensor [12] and a V-REP sensor model called "Blob detection camera". The Hokuyo sensor was placed on the front side of youBot base, and the camera positioned on the tip of the arm, just before the grippers, as can be seen on figure 5.

The problem set in V-REP is the following: youBot has to position itself next to the table, with a predefined

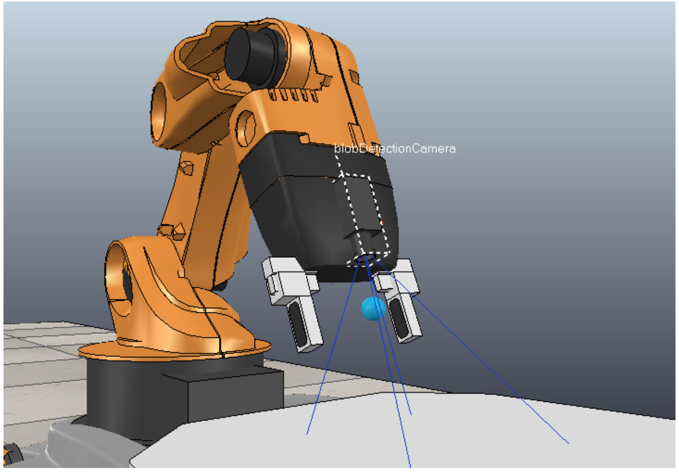

Fig. 5. Blob detection camera position on youBot arm.

orientation (positioned towards the table). The table has a white cube on its top. Once arriving to it's destination, the robot has to position it's camera on top of the table and detect the cube, using the blob detection camera sequence of filters to segment the object, as displayed on figure 6. Robot movement and positioning was managed by ROS code running outside V-REP.

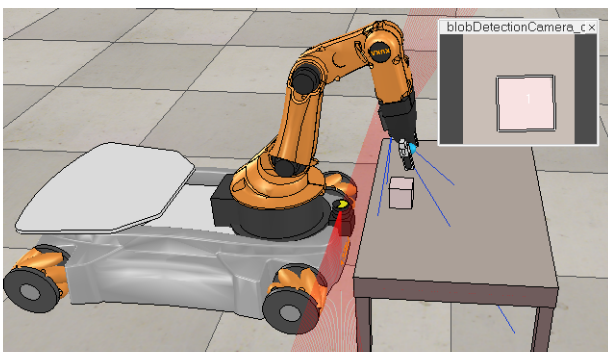

Fig. 6. Arm positioning and blob detection.

Object detection in images is the first step towards object recognition (classification and definition of object's position inside a 2D image). Currently, YOLOv3 [13] is being tested for these tasks and once the robot redesign is complete, it will run in the Jetson TX1 in the Pioneer 3-AT. Once the object detection occurs in the RGB image, the second step is to estimate the object's position relative to the robot, so that object manipulation can be performed. This year, UTBots@Home will present a tutorial on object detection with YOLO.

IV. CONCLUSION AND FUTURE WORK

The team is capable of fulfilling several tasks of the Robocup@Home category successfully. Among the tasks under development are the more accurate recognition of voice commands, and the integration of SLAM and manipulation tasks. The remote characteristic of this year's event allowed the planning of a redesign of the Pioneer 3-AT robot, which will allow the execution of GPU-based algorithms such as convolutional neural networks. Object recognition is expected to be the main advance this year. For the future, research projects focused on the manipulation arm and in the human-robot interaction can result in new ideas and approaches for related tasks.

ACKNOWLEDGMENT

The authors would like to thank the financial support of UTFPR.

REFERENCES

[1] Official homepage: Robocup@Home. http://www.robocupathome.org/. Accessed June 2019.

[2] L. IOCCHII, D. HOLZ, J.R. del SOLAR, K. SUGIURA, T. van der ZANT, RoboCup@Home: Analysis and results of evolving competitions for domestic and service robots. Artificial Intelligence. 229, C (December 2015), 258-281. DOI: https://doi.org/10.1016/j.artint.2015.08.002

[3] About ROS Official Homepage. http://www.ros.org/about-ros/. Accessed June 2019.

[4] Jetson TX1 Module. https://developer.nvidia.com/embedded/jetson-tx1. Accessed September 2020.

[5] MICROSOFT, Develop Network: Kinect Sensor. http://msdn.microsoft.com/en-us/library/hh438998.aspx. Accessed June 2019.

[6] Lidar Scanner Ydlidar X4. https://acroname.com/sites/default/files/assets/ydlidar_g4_datasheet_0.pdf. Accessed June 2019.

[7] R. PLUTCHIK, H. KELLERMAN (Ed.). Emotion: theory, research and experience. New York: Academic press, 1986.

[8] P. S. PALAR. ROS package ros_display_emotions. https://github.com/kpiatan/ros_display_emotions. Accessed September 2020.

[9] S. RIISGAARD, M. R. BLAS, SLAM for Dummies. http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf. Accessed June 2019.

[10] Coppelia Robotics V-REP: Create. Compose. Simulate. Any Robot. http://www.coppeliarobotics.com. Accessed June 2019.

[11] KUKA YouBot. Mobile manipulator for research and education. https://bit.ly/2r1uyql. Accessed June 2019.

[12] Scanning Rangefinder Distance Data Output/URG-04LX-UG01 Product Details — HOKUYO AUTOMATIC CO., LTD. https://www.hokuyo-aut.jp/search/single.php?serial=166. Accessed June 2019.

[13] REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement. arXiv, 2018.