UTBots@Home 2022 - Pionner 3-AT (A.K.A. Apollo) current and future developments

Piatan S. Palar[1], Marlon O. Vaz[2], Bruno Bock Silva[3], Gustavo F. Lewin[3], Gustavo F. Armênio[4], Adrian C. Silva[5], Emanuel H.V. de Amorim[5], Gabriel M. C. Costa[5], Hermann J. R. Reuter[5], Felipe P. Conter[6], André S. de Oliveira[1] and João A. Fabro[7]

***Abstract*— This TDP describes the hardware and software solutions currently applied to allow the Pioneer robot from UTFPR (Federal University - Paraná) 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 sensor, that provides information necessary for navigation, object and people 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 control programs can be executed in the robot using multiple processing units (such as an NUC onboard computer, Arduino and Jetson Nano). Also, we present the current solutions in development for the person recognition and manipulation tasks, as well as 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 research robot is under continuous development by team UTBots@Home at UTFPR[1]: "Apollo", presented in Figure 1. This robot is composed of a mobile base (Pioneer 3-AT), a manipulator with 3 degrees of freedom, a LiDAR range finder, a screen to present the "face" of the robot, speakers for verbal communication, and a depth/RGB camera.

The applications are running with ROS (Robot Operating System), a set of software libraries and tools that help to

[1]HomePage of the Team: https://laser.dainf.ct.utfpr.edu.br/doku.php?id=utbots_at_home

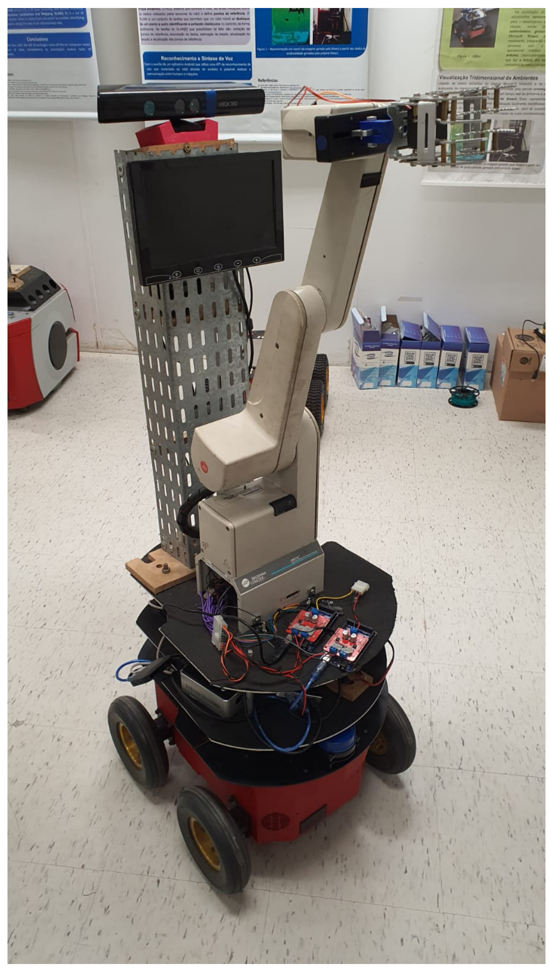

Fig. 1. Pioneer 3-AT (a.k.a. "Apollo") with its support for the Kinect sensor, the LCD display for simulated emotions and the manipulator.

[1]A. S. de Oliveira is a professor with CPGEI - Graduate Program of Electrical Engineering and Industrial Informatics, UTFPR, Brazil - Campus Curitiba. Piatan Palar is a PhD student within this Graduate Program. andreoliveira@utfpr.edu.br, emailpiatan@alunos.utfpr.edu.br

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

[3]B. Bock Silva, G. F. Lewin are students of Electronics Engineering in UTFPR. {brunobock,lewin}@alunos.utfpr.edu.br

[4]G. F. Armênio is a student of Computer Engineering in UTFPR.. gustavofardoarmenio@alunos.utfpr.edu.br

[5]A. C. Silva, E. H. V. de Amorim, G. M. C. Costa, H. J. R. Reuter are students of Mechatronics Engineering in UTFPR. {adriansilva, emanuelamorim,gabrielmelocosta, hermannreuter}@alunos.utfpr.edu.br

[6]F. P. Conter is with UNILA-Latin American Integration Federal University - Foz do Iguaçu - Paraná - Brazil. felipeconter.cc@gmail.com

[7]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. He is also the leader of the Laboratory of Embedded Systems and Robotics at UTFPR (LASER-http://laser.dainf.ct.utfpr.edu.br) fabro@utfpr.edu.br

develop applications for robotics [3], that run on a the NUC embedded computer placed inside the robot. The control system of the robots is responsible for acquiring and interpreting various sensors and, from that data, make decisions to fulfill the 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.

II. HARDWARE

A. Robot Specifications

The robot is based on a Pioneer 3-AT research robot: this base 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 embedded computer via a serial-USB interface allowing its connection to a ROS system (Figure 2).

A redesign is being planned, and a Jetson Nano [4] will be coupled to the robot, in order to process computer vision and other GPU-focused processes. The idea is to integrate the Jetson Nano to the ROS network on the embedded NUC computer [5].

Fig. 2. Side view of the robot subsystems mounted on top of the Pioneer 3-AT base.

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 LCD Display and the Kinect sensor to a height comfortable enough to allow interaction with people (1.5 meters approximately). In addition to that, an Android device is needed to capture the voice commands.

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 [6]. 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. "Apollo" has a YDLIDAR X4, which is a low cost LIDAR sensor capable of 360 degrees distance measurements [7].

D. Manipulator

The original model of the manipulator used in the robot is a Beckman Coulter ORCA Robotic Arm, a planar manipulator that has three rotational joints and can be seen in Figure 1. To make the structure respond to the desired commands, a process of retrofitting was carried out, where the motors and encoders were connected to Arduino Mega microcontrollers that allow the open programming of the mechanism. This arm is fixed on the Pioneer 3-AT robot.

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.

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 is controlled via a Nuc computer with the "Melodic" version of ROS, on the operating system Ubuntu 18 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, a core process (*roscore*) manages processes (nodes) and topics. 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).

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 (Figure 3).

Emotions were based on Plutchik's wheel of emotions[8]. A LCD screen running the ROS Image Viewer application displays these faces by subscribing to a ROS topic. The emotions system is based on *ros_display_emotions* package [9]. Figure 4 illustrates an emotion being displayed by the robot through the screen.

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 Localization and Mapping) [10] 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.

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

For object recognition, an apparatus that allows capturing of a large number of images of each object has been developed. This system has two Logitech C920 cameras (stereovision) and an ASUS Xtion depth sensor, which allows the capture of 2,600 RGB images by the camera and 2,600 point cloud images. Due to the way the images are collected, it is possible to create 3D objects by applying only the texture. Therefore, it is possible to generate a large dataset of every object, as well as a general 3D format of it.

The object 3D pose estimation method is based on the intersection of 3D arrays in the environment's 3D space. Its most significant singularity is that it requires no prior knowledge about the 3D shape of the target object. The detection of the object occurs in an earlier step, using the YOLOv3 neural networt [11], integrated with ROS by the *darknet_ros* package [12]. Since YOLOv3 works only with the RGB layer, the object's 3D shape information does not influence directly the object detection, which allows for the detection of objects with varying shapes, such as apples and bananas. The intention is to run this processes on the Jetson in the Pioneer 3-AT.

Furthermore, upon training the YOLOv3 model, a variety of objects can be grouped under a certain class - for example, we can label the same class to different instances of apples, with different sizes, poses and colors (green and red apples).

YOLOv3 can even identify new instances of objects: instances never seen before by the robot. By identifying a point in the center of the visible face of the object, and a point on the RGB-D sensor, a 3D array is defined. By moving the robot, other 3D arrays are defined. The method makes use of those 3D arrays to estimate a point that represents the object's 3D center. Experiments have shown that this method performs with a mean euclidian distance of 9.4 cm between the true 3D center of the object and the estimated coordinate.

F. Person Recognition

To perform the person recognition tasks, the RBG layer of the Kinect is evaluated by YoloV3, which detects and counts the individual persons present in the frame, defining their bounding boxes.

For most advanced tasks such as the identification of a specific person amongst a crowd, the idea is to perform a second evaluation inside each bounding box, utilizing a different dataset, to check if the person detected is the one the robot is looking for.

G. Manipulation

In a new project involving optimization and alternative robotic kinematics techniques, the retrofit process of the

Fig. 3. Examples of faces developed.

manipulator arm was carried out. The manipulator has five rotational degrees of freedom and a claw-like end effector for manipulating objects.

Considering that the mean error of the position estimate is sufficiently big for the range of grip of the claw, it is planned to fix a Realsense F200 RGB-D sensor (which records distances at a resolution of 480p [13]) near the arm's handle, through a 3D printed support. Thus, the camera can observe the operation of the claw, and perform new position estimates using the same methods as the arm moves. The movement creates new challenges related to the position of the camera in the 3D space and perspective issues, that must be considered.

The attached RealSense F200 is enough to detect different surfaces in the manipulator's working space. This data will be sent as a PointCloud via ROS network.

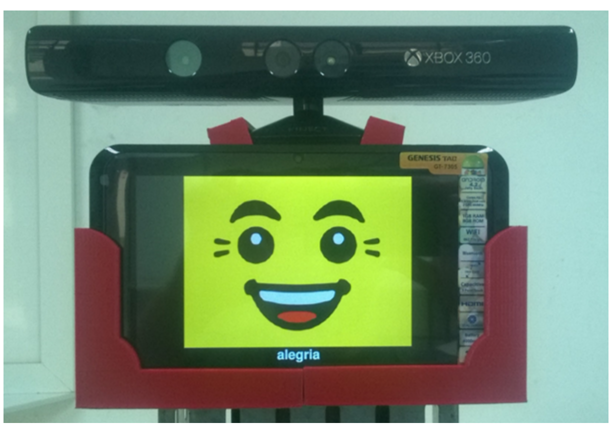

Fig. 4. Simulated emotion presented on the face of the robot.

The manipulator movements are measured through quadrature encoders, limited by limit switch sensors and are also sent through the ROS network.

Setpoint angle commands for each joint are sent through ROS where forward and inverse kinematics calculations are computed using Clifford's Algebra. Such techniques seek to implement more efficient calculation tools using the algebraic advantages of the dual quaternion, which are a subset of Clifford's Algebra. Clifford's Algebra is also used to relate the objects detected by the camera to points and planes that constitute a collision and manipulation control system.

H. Simulation

A scene containing the Apollo robot and a set of obstacles was developed for the CoppeliaSim simulation platform (version 4.2.0), which allows the interfacing of all ROS nodes at an entirely virtual environment.

A model of the Pionner 3-AT was developed by the means of 3D modeling software and exported to CoppeliaSim.

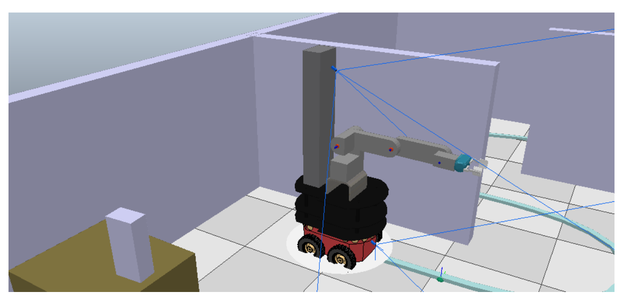

Fig. 5. Apollo's simulation on CoppeliaSim.

Actuation of the motors vary according to the linear and angular velocity commands read in subscribed ROS topics.

As for the LIDAR model, a pre-existing model was used (*hokuyo*). It allows the detection of distances to nearby obstacles.

The robotic arm Beckman Coulter ORCA was also replicated in a 3D model. In this case, 3 joint motors were applied (shoulder, elbow and wrist). They receive angular values from ROS topics responsible for the movement of the arm's joints.

CoppeliaSim's built-in model BaxterGripper was used in order to simulate the real gripper. This model is subscribed to a ROS topic responsible for its closure and opening.

Aiming to simulate the operator's vision and reproduce the executed processes with the highest precision, an RGB camera was placed following all dimensions used in the physical robot. Figure 5 shows the 3D model of the robot.

IV. CONCLUSION AND FUTURE WORK

The team is capable of fulfilling several core tasks of the Robocup@Home category successfully. Among the work under development are more accurate recognition of voice commands, more sophisticated person recognition methods and position estimate corrections, as well as improvements in the manipulator arm. Furthermore, correctly integrating the robot's subsystems is an ongoing challenge as its effectiveness, and complexity, grows. Finally, the simulation ambient has been an important addition that enables multiple simultaneous testing without overstressing the robot.

ACKNOWLEDGMENT

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

REFERENCES

[1] Official homepage: Robocup@Home. Available at: http://www.robocupathome.org/. Accessed June 2022.

[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. Artifi cial Intelligence. 229, C (December 2015), 258-281. DOI: https://doi.org/10.1016/j.artint.2015.08.002

[3] About ROS Official Homepage. Available at: http://www.ros.org/about-ros/. Accessed June 2022.

[4] Jetson Nano Module. Available at: https://developer.nvidia.com/embedded/jetson-nano-developer-kit. Accessed June 2022.

[5] Intel Nuc. Available at: https://www.intel.com.br/content/www/br/pt/products/details/nuc.html. Accessed June 2022.

[6] MICROSOFT, Develop Network: Kinect Sensor. Available at: http://msdn.microsoft.com/en-us/library/hh438998.aspx. Accessed June 2022.

[7] Lidar Scanner Ydlidar X4. Available at: https://acroname.com/sites/default/files/assets/ydlidar_g4_datasheet_0.pdf. Accessed June 2022.

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

[9] P. S. PALAR. ROS package ros_display_emotions. Available at: https://github.com/kpiatan/ros_display_emotions. Accessed September 2021.

[10] S. RIISGAARD, M. R. BLAS, SLAM for Dummies. Available at: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf. Available at: Accessed June 2022.

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

[12] BJELONIC, M., YOLO ROS: Real-Time Object Detection for ROS. Available at: https://github.com/leggedrobotics/darknet_ros. Accessed June 2022.

[13] Available at: Intel Realsense. Available at: https://www.intel.com.br/content/www/br/pt/architecture-and-technology/realsense-overview.html. Accessed September 2020.