UTBots@Home 2023 Team Description Paper - UTFPR

Marlon O. Vaz[1], Bruno Bock Silva[2], Gustavo F. Lewin[2], João H. C. Pires[2], Gustavo F. Armênio[3], David Segalle[3], Enzo H. Gaio[3], Victor B. Errera[3] Renzo T. da Rosa [4], Larissa B. L. da Silva [5], Luiz G. S. Greca[5], Felipe P. Conter[6], André S. de Oliveira[7] and João A. Fabro[8]

***Abstract*— This Team Description Paper presents an overview of the current solutions and ongoing developments by the UtBots@Home team for the challenges presented by the Robocup@Home initiative. The team utilizes Apollo, a Pioneer 3-AT mobile base, or Zeus, a Pioneer LX base, equipped with various sensors and peripheral devices connected to a ROS (Robot Operating System) network. This network, running on portable computers, facilitates communication and data processing. The ROS-based software enables the robot to perform a wide range of functionalities, including navigation, environment mapping, auto-localization, emotion simulation for human-robot interaction, voice recognition and synthesis, object and people recognition, 3D and 2D person tracking, face recognition, and object manipulation. The team is continually working to enhance the robot's capabilities and has ongoing projects, such as a contraption for generating object 2D and 3D information datasets, developing a new arm, and conducting research in Natural Language Processing, which paves the way for new functionalities and improvements to competition tasks.**

I. INTRODUCTION

Since 2014, the UTBots@Home[1] team has been dedicated to the ongoing development of service robots designed to assist humans in household tasks. Our efforts are driven by the challenges presented by the Robocup@Home competition [1] and extend beyond its scope, in a variety of fields such as Human-Robot Interaction and Cooperation, Navigation and Mapping in Dynamic Environments, Computer Vision and Object Recognition under Natural Light Conditions, Object Manipulation, Adaptive Behaviors and Environmental Intelligence.

Over the years, the competition has served as a platform for numerous academic research endeavors at UTFPR. Master's and doctoral projects have played a pivotal role in enhancing the capabilities of the team's robots and pushing the boundaries of their respective research domains.

Current research and advances focuses on various topics, including but not limited to enhancing the ability to learn new objects, striving for more precise 3D point estimation of objects, exploring person re-identification techniques, leveraging natural language processing (NLP) to enhance human-robot interaction, the development of a new and robust robotic arm, and developing adaptive behavior for complex tasks.

This Team Description Paper (TDP) provides an overview of the team's current solutions and ongoing research for various challenges in the competition. Section 2 provides a description of the approaches developed by the team, with software and hardware details. Section 4 focuses on the team's current research endeavors. Finally, section 5 concludes with a summary and offers insights into future perspectives.

II. OVERVIEW OF OUR APPROACHES

To perform a variety of tasks with multiple types of input data and output responses, the team utilizes several sensors, an LCD display, speakers, an adapted robotic arm and two mobile robotic bases, Apollo and Zeus. To process all the information and coordinate responses accordingly, the use of portable computers in a Robot Operation System (ROS) [2] network is essential. Our robot Apollo can be seen in Figure 1. The following subsection highlight the main technologies embedded in our robot.

A. ROS - Robotic Operating System

The robot is controlled by a ROS network made of two connected portable computers, an Intel NUC [3] with the "Noetic" version of ROS, on the operating system Ubuntu 20.04 LTS, and a NVIDIA Jetson Nano [4] board with a "Melodic" version of ROS. ROS allows the application of several tools, libraries and conventions that simplify the development of complex and robust tasks for robots [2]. Topics are data buses through which nodes exchange messages, and

[1] Marlon O. Vaz is a professor with IFPR - Federal Institute of Parana, Campus Pinhais, Pinhais. He is also a PhD. student at CPGEI/UTFPR. marlon.vaz@ifpr.edu.br

[2] Bruno Bock Silva, Gustavo F. Lewin and João H. C. Pires are students of Electronics Engineering at UTFPR - Federal University of Technology - Parana, Brazil - Campus Curitiba. {brunobock,lewin, joaopires}@alunos.utfpr.edu.br

[3] David Segalle, Enzo H. Gaio, Victor B. Errera and Gustavo F. Armênio are students of Computer Engineering at UTFPR. {davidsegalle,enzoholzmanngaio,verrera gustavofardoarmenio}@alunos.utfpr.edu.br

[4] Renzo T. de Rosa is a student of Information Systems at UTFPR. renzotognella@alunos.utfpr.edu.br

[5] Larissa B. L. da Silva and Luiz G. S. Greca are students of Mechatronics Engineering at UTFPR. {larissabialski, luizgreca}@alunos.utfpr.edu.br

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

[7] André S. de Oliveira is a professor with CPGEI - Graduate Program of Electrical Engineering and Industrial Informatics, UTFPR andreoliveira@utfpr.edu.br

[8] João A. Fabro is a professor with the Graduate Program on Applied Computing - PPGCA, Informatics Department - DAINF - at UTFPR. He is also the leader of LASER - Laboratory of Embedded Systems and Robotics (http://laser.dainf.ct.utfpr.edu.br) fabro@utfpr.edu.br

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

multiple nodes may subscribe to each topic (receiving its data) or publishing to it (sending data).

B. Robot Navigation

In order to perform the navigation of the robot in complex environments, we have two mobile bases available, Apollo and Zeus. Both can be connected easily to the ROS network.

Apollo is a Pioneer 3-AT research robot [5]: 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 base platform does not come equipped with a LiDAR sensor, so a YDLIDAR X4 sensor is employed in conjunction with the base. The YDLIDAR X4 is an affordable LiDAR sensor capable of performing 360-degree distance measurements [6].

Zeus is a Pioneer LX model [7], that weights 60 kg, can carry up to 60 kg of payload, travel at a maximum speed of 1.8 m/s, and has a autonomy of 13 hours of continuous operation.The dimensions of the Pioneer LX robot are: 50 cm wide, 70 cm deep and 45 cm high. This base comes with its own LiDAR sensor, a SICK LMS300 laser range finder with 250 degrees of view and 15m of range [7].

Fig. 1. Side view of the robot subsystems mounted on top of the Pioneer 3-AT base.

In robot navigation, odometry errors can lead to substantial uncertainty. 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) [8] algorithms can achieve this. Currently, the LiDAR sensor readings and the robot's wheels odometry are fed to the standard ROS navigation stack and *move_base* is utilized 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.

C. Voice recognition and synthesis

For speech recognition, a ROS package called *whisper_cpp_ros*[2] was developed. Aimed at robust and efficient speech recognition capabilities based on machine learning, this package utilizes 2 programs: (i) *Silero Voice Activity Detection* (VAD) for voice activity detection, and (ii) *whisper.cpp*, a C++ implementation of Whisper, a well-established speech recognition system. Within the ROS system, the *voice activity detection* (VAD) node efficiently detects voice activity in audio streams, allowing the program to focus its processing resources on relevant speech segments. The integration of *whisper.cpp* further enhances the system, providing automatic speech recognition and real-time transcription of spoken language.

Regarding speech synthesis, a ROS node was developed that utilizes the *Mimic3* speech synthesis engine, integrated with the robot's system to produce natural sounds. Leveraging the capabilities of ROS, the program processes textual inputs and accurately synthesizes them into realistic and expressive speech. The speech synthesis algorithms of *Mimic3* contribute to the system's ability to mimic human-like intonation, cadence, and emotions, enhancing the overall user experience and engagement with robotic systems.

D. 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 2).

Emotions were based on Plutchik's wheel of emotions [9]. A LCD screen running the ROS Image Viewer application displays these faces by subscribing to the custom ROS $\backslash$*face_emotion* topic. The emotions system is based on the custom *ros_display_emotions* package [10], which runs a ROS node that subscribes the $\backslash$*emotion* topic and publishes a image message to the $\backslash$*face_emotion* topic.

[2]Available at: https://github.com/UtBotsAtHome-UTFPR/whisper_cpp_ros

Fig. 2. Examples of faces developed.

E. Robot Vision

The Kinect V1 depth sensor captures RGB-D images, which offer visual and depth information essential for generating point clouds and enabling various functionalities [11]. In order to establish the communication between the Kinect sensor and ROS the *freenect_stack*[3] package was used.

1) Object Recognition: For object recognition, the *darknet_ros* package [12] performs the integration of the YOLOv3 neural network [13] with ROS. Upon training the YOLOv3 model, a variety of objects can be recognized as units of certain classes. The *darknet_ros* package works by subscribing to the Kinect RBG image topic, so it can perform real-time detection, publishing bounding-box and class messages. Running the model in real-time in the Jetson Nano board guarantees high accuracy classifications with a mean of 1.5 frames per second, taking advantage of graphical acceleration of processes. An example of custom objects detection can be seen in Figure 3.

For manipulation tasks, it is essential to estimate the object 3D position as well. RGB-D images, used alongside an object detector, can provide distance information of a certain detected object. A detector outputs a bounding box for a certain object in a RGB image and matching the RGB and Depth images means that the distances of the pixels of an object can be extracted. A statistic method applied on the multiple distances of the frame results in a unique distance value for the object and, transformed in a 3D cartesian point through trigonometric calculations, a 3D location in relation to the robot. Experiments with different statistics showed a maximum error of 7.6 cm, considering the Kinect sensor's poor accuracy.

[3]Available at: https://github.com/ros-drivers/freenect_stack

2) Person Detection, Tracking and Pose: To detect people in the robot's point of view, *darknet_ros* can also be applied, with weights trained to detect people, as Figure 4 shows. It can adittionaly be used to segment people from the original frame, cropping the bounding box.

For tasks such as following a person and identifying gestures, the custom *mediapipe_track* [4] package implements MediaPipe Pose [14] with ROS messages. MediaPipe Pose is responsible for processing body pose information and tracking the general skeleton pose of a person, in the manners presented in Figure 5. The *mediapipe_track* package enhances this functionality by performing tracking and generating

[4]Available at: https://github.com/UtBotsAtHome-UTFPR/mediapipe_track

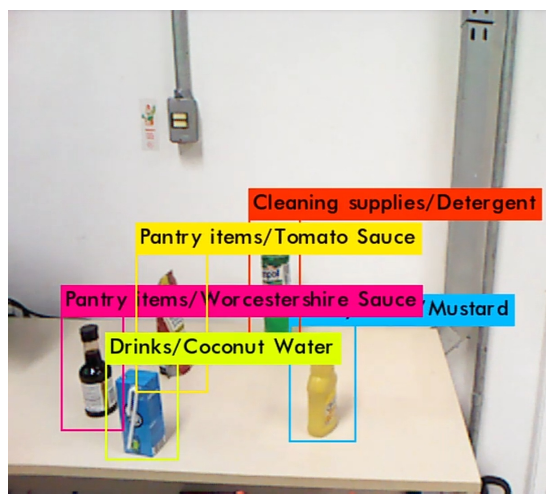

Fig. 3. YOLO v3 model detecting a custom set of objects real-time with ROS.

3D position estimates. It achieves this by subscribing to Kinect RGB and Depth messages, including cropped bounding boxes, and can subsequently publish the processed data for consumption by other nodes.

3) Face Recognition: In order to solve personal recognition tasks, a K-nearest neighbours (KNN) algorithm was chosen. This algorithm is divided in 3 phases, image gathering, training and recognition.

During the image gathering phase, images are obtained from the RGB layer of the Kinect and processed using the *face_recognition* Python library to locate, crop, and save the face of the subject to be trained. For the training phase, the same library is used to load all the captured images and map them to a space. K-Nearest Neighbors (KNN) algorithms are applied to create clusters of points representing expected classes, enabling identification of unknown data points based on their proximity to these clusters. To speed up the process, a Ball Tree algorithm is employed to efficiently find the nearest points in the space. In the final stage of recognizing people, the Kinect is used again to capture images, extract the face, and identify its location within the space, using the same logic as image gathering and training. Utilizing the Ball Tree, the unknown data point is quickly analyzed, and it is assigned to a group representative of an individual or categorized as an unknown subject.

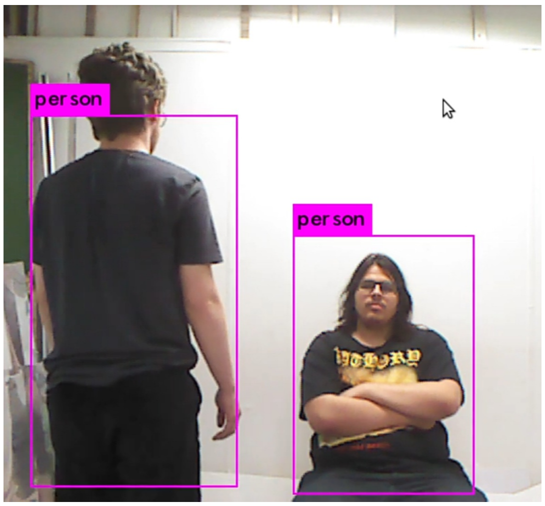

Fig. 4. YOLO v3 model detecting people in real-time with ROS.

Fig. 5. MediaPipe Pose detecting the person's skeleton with ROS.

Figure 6 shows a user being recognized after a capture and training.

F. Manipulation

The manipulator used in the robot is a Beckman Coulter ORCA Robotic Arm [15], a planar manipulator that has three rotational joints and can be seen in Figure 1. A claw-like structure, coupled to a high torque servo motor, was attached to the extremity of the arm. 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 for open programming of the mechanism.

In addition to the three motors that define the degrees of freedom of the structure, each joint is also coupled to a quadrature encoder, that allows for the microcontrollers to independently control each joint in motion. To acomplish

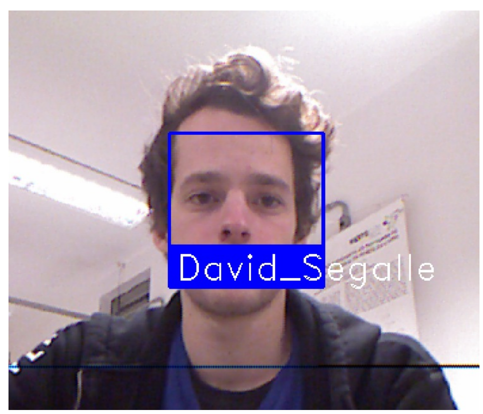

Fig. 6. Image showing a user being recognized and his position displayed in the image.

that, it allows to infer the direction in which the motor is rotating.

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. Both Arduinos are connected to USB ports of the NUC computer, 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.

Through the set-point angle commands, 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.

G. Simulation

A scene containing the Apollo robot and a set of obstacles was developed [16] for the CoppeliaSim simulation platform (version 4.2.0) [5], which allows the interfacing of all ROS nodes at an entirely virtual environment. Figure 7 shows the 3D model of the robot.

The robotic arm Beckman Coulter ORCA was replicated in a 3D model, and put together with the Pioneer 3-AT mobile base model already available at the simulator. 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. As for the LIDAR sensor, a pre-existing model was used (*hokuyo*). It allows the detection of distances to nearby obstacles.

Aiming to simulate the operator's vision and reproduce the executed processes with the highest precision, a RGB camera was placed following all dimensions used in the physical robot.

[5]CoppeliaSim robotics simulator can be download, free of charge for non-commercial applications at http://www.coppeliarobotics.com

III. CURRENT RESEARCH

A. Rapid object RGB-D dataset gathering

The classes detected and the accuracy of the detection depends on the number and quality of labeled images of the dataset used for training. To improve the dataset quality, an external apparatus that allows capturing of a large number of images of each object has been developed [17]. 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, each one viewed by a slightly different angle and distance.

Therefore, it is possible to generate a large dataset of every object, as well as a general 3D format of it. This information can be used to quickly add a new object to the detector and the 3D format can be of use to fine-tune the estimation of a 3D point for object manipulation.

B. Natural Language Processing and Understanding

In order to improve the human-robot interactions, it is valuable to increase the robot's interpretation of speech. In this context, the use of Natural Language Processing (NLP) comes in handy, with the capability of extracting intention and turning information into entities. The use of such functionalities brakes through the limitations of standard phrases and commands and makes the robot capable of processing synonyms and distinct phrase constructions for a given context. To implement such capabilities in the robot's interactions, research and development is being conducted with Rasa [18], an open source solution based on machine learning commonly used to implement chatbots.

C. New Robotic Arm

In order to achieve more refined and versatile robotic manipulation, the team is currently developing a new set of arms and grippers. The arm is being constructed as

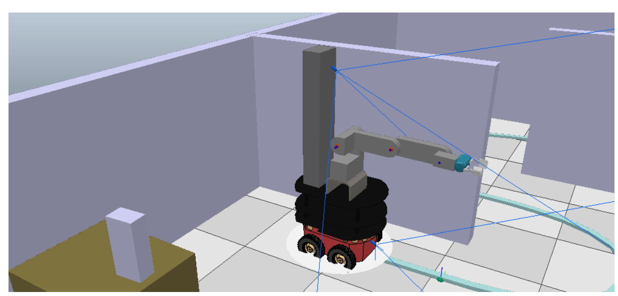

Fig. 7. Apollo's simulation on CoppeliaSim.

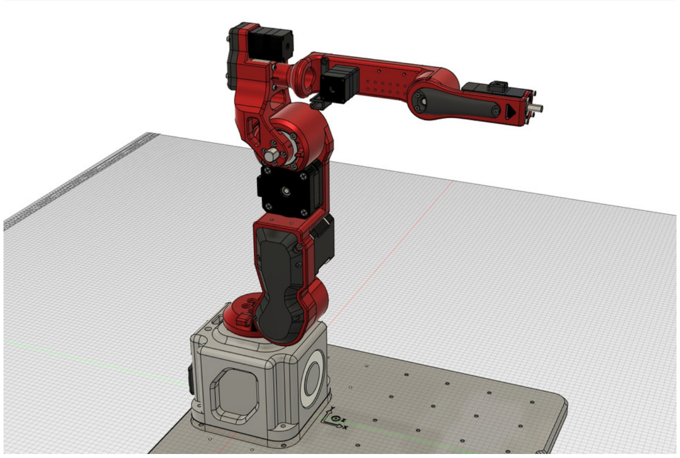

Fig. 8. 3D model of the new arm in development

an adapted version of the open-source Dummy-Robot arm, originally developed by Zhihui[19]. As for the grippers, the plan is to create a set of custom grippers that can be easily swapped among them, allowing for modularity and distinct applications (Figure 8).

IV. CONCLUSION AND FUTURE WORK

The team has demonstrated the ability to successfully accomplish several core tasks in the Robocup@Home category and has made rapid progress in developing more complex functionalities and robot behavior control in recent years. Noteworthy developments include emotion simulation using *ros_display_emotions*, person tracking, and 3D pose estimation with *mediapipe_track*, which are innovative contributions to the competition. Ongoing projects offer promising insights into the team's future evolution, such as the contraption for RGB-D dataset generation of objects and a new robust arm, both contributing to advancing object manipulation tasks that remain challenging for most teams. Additionally, the use of NLP with Rasa opens up new horizons for a more sophisticated human-robot interaction. The team continuously strives to expand the robot's capabilities, exploring innovative frontiers to advance research across various fields of knowledge. An important a goal is to bring credibility to Latin America in the realm of service robotics and technology as a whole.

ACKNOWLEDGMENT

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

REFERENCES

[1] Robocup, "Robocup@Home,"athome.robocup.org.`https://athome.robocup.org/`(accessed Jun., 2023).

[2] Open Robotics, "ROS/Introduction, "wiki.ros.org.`http://wiki.ros.org/ROS/Introduction`(accessed Jun., 2023).

[3] Intel "What is Intel® NUC?,"intel.com.br. `https://www.intel.com.br/content/www/br/pt/products/details/nuc.html`. (accessed Jul. 2023).

[4] NVIDIA, "Jetson Nano Developer Kit,"developer.nvidia. `https://developer.nvidia.com/embedded/jetson-nano-developer-kit`. (accessed Jul., 2023).

[5] Pioneer 3-AT Operations Manual, Rev. 6.4 , Adept MobileRobots, Amherst, NH, USA, 2013.

[6] YDLIDAR, "X4 Datasheet,"01.13.000000, 2018.

[7] Pioneer LX User's Guide, Rev. D , Adept MobileRobots, Amherst, NH, USA, 2017.

[8] S. S. Riisgaard; G. F. Blas, "SLAM for Dummies,"ocw.mit.edu. `http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf`. (accessed Jun., 2023).

[9] R. Plutchik; H. Kellerman, "Emotion: theory, research and experience,". New York: Academic press, ed. 1, 1986.

[10] Palar, P. S, "ROS package ros_display_emotions,"github.com. `https://github.com/kpiatan/ros_display_emotions`. (accessed Jun., 2023).

[11] MICROSOFT, "Develop Network: Kinect Sensor,"msdn.microsoft.com. `http://msdn.microsoft.com/en-us/library/hh438998.aspx` (accessed Jun., 2023).

[12] M. Bjelonic, "YOLO ROS: Real-Time Object Detection for ROS,"github.com. `https://github.com/leggedrobotics/darknet_ros` (accessed Jun., 2023).

[13] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement,"arXiv preprint arXiv:1804.02767, 2018.

[14] MediaPipe, "Pose landmark detection guide,"developers.google.com. `https://developers.google.com/mediapipe/solutions/vision/pose_landmarker/`(accessed Jul. 2023).

[15] "BECKMAN COULTER ORCA Manipulator,"American Laboratory Trading, `https://americanlaboratorytrading.com/lab-equipment-products/beckman-coulter-orca-control-module-with-robotic-arm-13916` (accessed Jul., 2023).

[16] J. Cerbaro, D. Martinelli, "UTBot@Home simulation environment"`https://github.com/UtBotsAtHome-UTFPR/P3AT-D`. (accessed Jul., 2023).

[17] J. Fabro, M. Vaz, A. Oliveira, "Design and Development of an Automated System for creation of Image Datasets intended to Allow Object Identification and Grasping by Service Robots,". January 2020, 1-6. DOI: `https://www.doi.org/10.21528/CBIC2019-136`

[18] Rasa Technologies Inc., "Open Source Natural Language Processing (NLP),"rasa.community. `https://rasa.community/open-source-nlu-nlp/` (accessed Jul., 2023)

[19] P. Zhihui, "Dummy-Robot: Super compact smart robotic-arm,"github.com. `https://github.com/peng-zhihui/Dummy-Robot` (accessed Jul. 2023)