Warthog Robotics @Home Team Description Paper 2023

Arthur Demarchi, Pedro A. Pizarro, Maurício Z. Varella,
Roberto K. Y. Junior, Gustavo S. Stéfano, Guilherme V. Nardari,
Rafael G. Lang, Ivan N. da Silva, Rhayna Christiani V. M. Casado
and Roseli A. F. Romero

Abstract— **This paper presents the Warthog Robotics Antares project, which has been developed since 2017. Antares is the first and yet only domestics services robot developed by the Warthog Robotics group from the Robotics Center (CRob) of the University of São Paulo. It's been used in presentations and fairs apart from competing in the Latin American Robocup@Home Competition (LARC/CBR) for the past seven years. Since the beginning, we focused on developing good and functional hardware and software, applying computer and engineering technologies for the best robot's performance. We show in this paper our new hardware and software improvements: a new robotics arm manipulator, the operation of the navigation system, and our software's skills update.**

I. INTRODUCTION

The group of service robots program of Warthog Robotics was founded with its first, and yet only, robot system in 2017 named Antares. Its purpose was to facilitate research and experiments about the interaction between humans and robots. The program, which initially started with a single project to mostly graduate students, now is a comprehensive area in which any graduate or under-graduate student can participate.

The system is composed of a *Pioneer P3-DX* base as the locomotion provider, a robotic arm as a manipulator, and sensors that vary from a Hokuyo laser to a Kinect 2 for world perception, connected to an ultratop NUC for the software processing and control. An in-depth description of the system's hardware is in our previous work [1].

The software division is into simple skills, such as speech, object detection, locomotion, face recognition, and manipulation. All these skill are Robotics Operating System (ROS) nodes used in a publisher/subscriber protocol while a state machine algorithm controls the flow of the system in a way that each node (skill) is executed in the correct order to complete complex task. This architecture helps us to develop each skill in separate environments without much hassle during integration. For an in-depth explation of their implementation we relate to our previous work [2].

The following sections describe a summary of the robot's components: the newly improved and still-in-development robotics arm manipulator, the operation of the well-hardware-and-software-integration navigation system, and our work on refining and updating the software's skills.

All authors are with Warthog Robotics, São Carlos School of Engineering and Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, São Paulo, Brazil.
* Corresponding Author: rhayna.casado@wr.sc.usp.br

Fig. 1: Real robot (left) and CAD model (right)

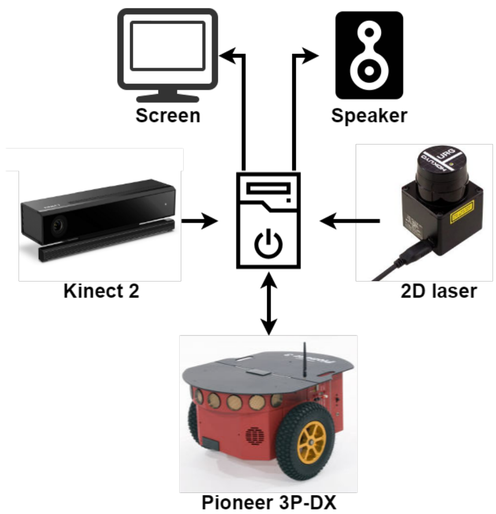

Fig. 2: Antares' hardware system.

II. SUMMARY

This section presents the Antares' system, providing a hardware and software summary. Therefore, the robot's hardware, developed over the years, consists of:

- **Central Processing Unit (CPU)** - A "Ultratop Liva ZePlus", with 4GB RAM "Impact DDR4" hooked to a USB hub. The CPU controls all sensors and actuators directly, except the manipulator, which has an *Arduino Nano* unit intermediating the communication between the motors and CPU.
- **Pioneer P3-DX** - A mobile base with a differential drive system and embedded controller that allows the robot's navigation through the skill-set "WRNavigation" built on top of *RosAria*.
- **Microsoft Kinect 2** - The leading sensor responsible for people tracking, gesture and objects recognition, and camera capabilities.
- **Laser Sensor** - A Hokuyo (URG-04LX-UG01) is used mainly for navigation and map generation. This sensor is localized at the bottom of the robot and is responsible for precision 2D mapping.
- **Microphone and Speaker** - An external speaker and a microphone with noise canceling cover both at the same *p2* connection of the CPU.
- **Screen** - A display with HDMI data input and a USB power input. It is mainly used as the robot's face through the "WRFace" software, based on the "homer_robot_face" *ROS* package [3] [4]. A 3D printing malleable box protects the screen and its PCB.
- **Manipulator** - A projected and team-made robotics arm that holds and transports objects. It is attached to the front of the robot. The manipulator is connected to the CPU by an *Arduino Nano*.

The components harmonize with software skills, which integrate via ROS. These skills are:

- ***WRFace*** - The face software, based mainly on the "homer_robot_face" node, portrays a likable face on the robot's display that can react according to what is occuring in the human-robot interaction.
- ***WRParser*** - The parser software is responsible for translating the audio caught by the microphone to text and understanding the provided commands. For that, the it isolates the keywords that represent the action that the robot must take.
- ***WRSpeech*** - The speech software generates audio that simulates human speech based on a string of characters. Hence, the robot can interact with humans and answer their questions.
- ***WRVision*** - The vision software takes part in face recognition and object recognition skills, being the major consumer of camera data and providing other software with the attributes, location, and classification of objects and people in the environment.
- ***WRFollow*** - The follow software is responsible for tracking people, recognizing a need-following agent, and setting goals for the navigation system. Therefore, it increases human-robot interaction, as the robot can recognize people and help them as they do their daily-basis activities. The *Microsoft Kinect 2* data is the tracking system base.
- ***WRNavigation*** - The navigation software works in mapping, motor control, obstacle avoidance, and other skill sets. It is fully integrated with the *Gazebo* and uses mainly (but not only) the laser sensor, *Hokuyo (URG-04LX-UG01)*, data. The navigation software base is an implementation of *SLAM* for 2D Lidar data.

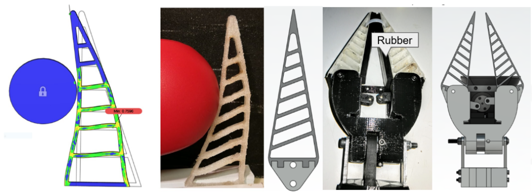

Fig. 3: Antares' actuator's hook.

III. HARDWARE

Over this year, we continued working on the manipulator. We focused on researching and improving our newest arm design. This project aims to achieve a more robust and more efficient arm with smoother movements, also using an actuator with a better grip. Figure 3 presents the actuator.

Figure 3 also shows that the side claws deform to fit better round objects. In Figure 4, we can also see the model developed in *Fusion Autodesk Software*.

The new arm has six rotation joints with stepper motors in each joint. For that, it gives the robot a greater degree of freedom than our previous manipulator. Also, these stepper motors aim to provide the desired smoothness. In addition, the arm's project includes attached encoders that control the manipulator's position, providing more precise movement.

In the 2021 online competition, we used a virtual model of the manipulator, which presented a very satisfactory performance achieving first place in the general classification [2]. Therefore, nowadays, we developed and advanced its hardware structure, firmware, and software integration. Hence, we're striving to maintain the characteristics and particularities of the simulated arm on our physical structure, mainly working on its electronics and mechanics.

The mechanical structure and the electronic features improved altogether in this new arm design were well-thought and projected to each element has its importance and contribution for a better manipulation performance of Antares. Therefore, we expect to bring a different design and efficiency to the competition and keep upgrading our hardware structure.

IV. SOFTWARE

A. Navigation

The navigation system is one of the main parts of the robot, as it must be capable of moving independently. To properly move, the system needs to understand its current

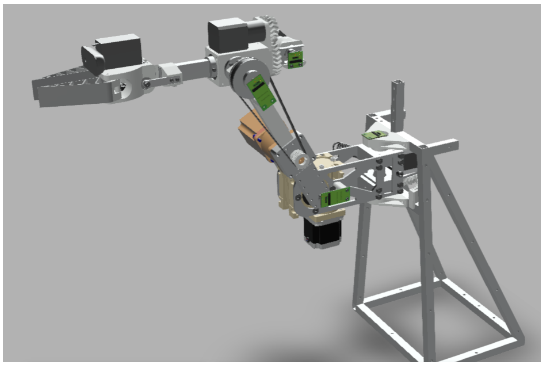

Fig. 4: Antares' robotic arm manipulator.

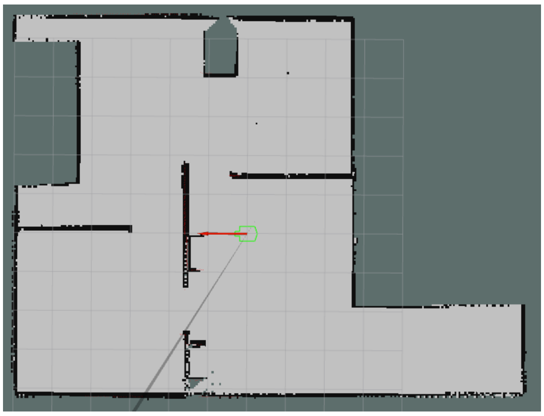

Fig. 5: RViz map.

position, have a goal position, create a route between these positions, and avoid obstacles that may show up.

Therefore, the hardware working is crucial, as it will provide the needed information for the navigation software process and do the actual moving. For that, a laser sensor located in the front of the robot is responsible for generating an environment map. For the map visualization, we use the ROS tool RViz. Figure 5 shows an example of these generated maps. With the environment map, the system can determine the robot's location and where it should go. Additionally, with the correct labels, it can correspond to each space as a different room.

To facilitate the development and improvement of the navigation system, we opted to use a simulated model in Gazebo, as presented in Figure 6. With this simulation, it was possible to create different home environments with challenges to our system so that we could understand how it responds to each of them. Then, we worked on improving most of our weaknesses and are still looking to keep developing those in remain.

By understanding its current location and the goal position, the system finds the best way for the robot to follow. Primarily, the path is short, with a possibility of going, which means there's an actual route between the positions, not a dead-end route.

Also, our system can recognize unexpected and moving obstacles that may show in the way, which in real life could

Fig. 6: Simulated Antares navigating.

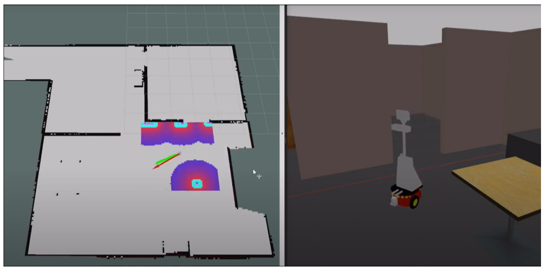

Fig. 7: Simulated Antares locating obstacles.

be a living thing such as an animal. Locating these obstacles is crucial because it ensures the robot is safe for usage. When it finds some block, expected or not, it dodges its route by circling the obstacle and then returns to its previous path. Figure 7 shows the RViz map generated when the simulated Antares finds a block on its way.

B. *Update*

One of the most meaningful aspects of software development this year was revisiting our previous code and analyzing its performance and quality. Therefore, we worked on keeping them updated, clean, and easy for other people to use and study the code. Then, using Python's library, we changed minor aspects in the code, which doesn't change how it works but makes it faster.

As we focused on minor changes, it didn't affect Antares' hardware. However, it's evident how these changes impacted our team and robot, as now it is notable which are the robot's weaknesses and how we should overcome them. Thus, we are willing to target our efforts towards these aspects, to always bring a better version of the robot to competitions and educational fairs.

V. CONCLUSIONS

In this paper, we presented our team's effort to improve our robot hardware and software organization, distribution, and processes. The manipulator manufacturing and implementation has been a great project that brings progress to Antares, as it can catch objects from different forms. Also, the navigation system and software update add knowledge for locating and solving errors that are extremely ennobling to the team. For future work, we expect to keep understanding our weaknesses and improving our other software skills.

References

[1] G. J. Acra, G. S. Stéfano, C. M. Ranieri, G. V. Nardari, A. H. Pinto, J. P. Belo, D. C. Tozadore, B. d. A. Iury, B. A. Mello, I. Arenales, *et al.*, "Warthog robotics@ home team description paper," *Sistema Olimpo*, 2019.

[2] F. P. Sanches, G. V. Nardari, J. P. R. Belo, T. Somensi, K. M. Sakiyama, R. Z. Lopes, B. B. d. Silva, R. G. Lang, I. N. d. Silfa, and R. A. F. Romero, "Warthog robotics@ home team description paper," *Sistema Olimpo*, 2021.

[3] V. Seib, J. Giesen, D. Grüntjens, and D. Paulus, "Enhancing human-robot interaction by a robot face with facial expressions and synchronized lip movements," 2013.

[4] V. Seib, S. Manthe, R. Memmesheimer, F. Polster, and D. Paulus, "Team homer@ unikoblenz—approaches and contributions to the robocup@ home competition," in *Robot Soccer World Cup*. Springer, 2015, pp. 83–94.