Warthog Robotics @Home Team Description Paper 2022

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 and Roseli A. F. Romero

Abstract— **This paper aims to describe the Antares system focusing on what was developed in the last year. This robot and it's auxiliary software and tools are developed inside the Warthog Robotics group of the Robotics Center (CRob) of the University of São Paulo. It's being used in presentations and fairs apart from competing in the Latin American Robocup@Home Competition (LARC/CBR) for the past five years. Being the first four years focused on the development of a functional hardware and the last year on the insertion of this system into a more educational environment where more students can learn its inner-works and particularities. We present Antares's basic software skills which are used in a state machine algorithm to perform complex tasks, also our efforts to implement a docker image able to run all of the robots skills and tools, a new implementation of the manipulator which is still under development and a server database to upload and tag photos for the object detection software.**

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 pourpose was to facilitate research and experiments about how is the interaction between humans and robots. The program, wich 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 which varies from a Hokuyo laser to a Kinect 2 for world perception, those are connected to a a ultratop NUC for the software processing and control. A more in depth description of the system hardware can be found in our previous work [1]

The software is divided into simple skills, such as speech, object detection, locomotion, face recognition, and manipulation which are all Robotics Operating System (ROS) nodes that are 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 right order for a complex task to be completed. This architecture helps us to develop the skill in separate environments without much hassle during integration. In the last year the skills, itself, were only improved in its efficiency and readability, that said for more in depth expiation of their implementation we relate to our previous work [2].

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: arthur.demarchi@wr.sc.usp.br

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

The following sections describe the new process of database generation for the object detection skill training, the newly improved and still in development robotics's arm manipulator and our efforts to implement a docker image to run inside the Antares system.

II. HARDWARE

For this year, the research related to the robot's hardware was focused on the manipulator, which consists of a completely new arm design. The project starts with the need for a more robust arm with smoother movements compared to the previous one. In addition, its actuator has a better grip due to the way it was developed. The actuator can be seen in figure 3:

In figure 3 it is possible to see that the side claws are deformed in order to fit better in round objects as well. In

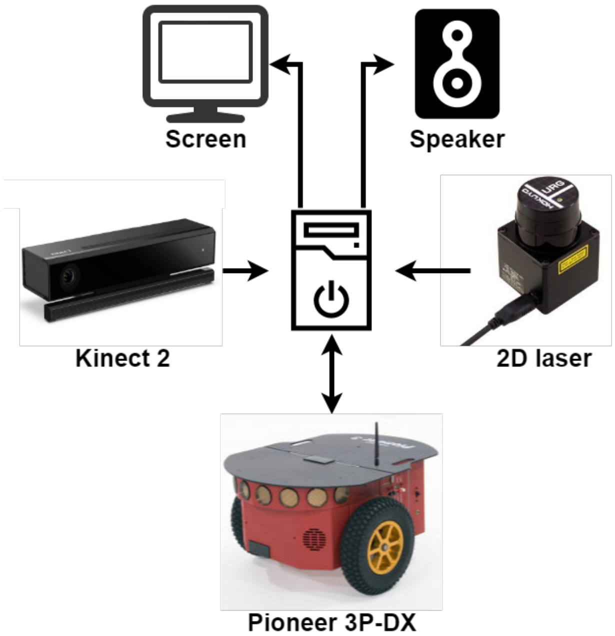

Fig. 2: Antare's hardware system.

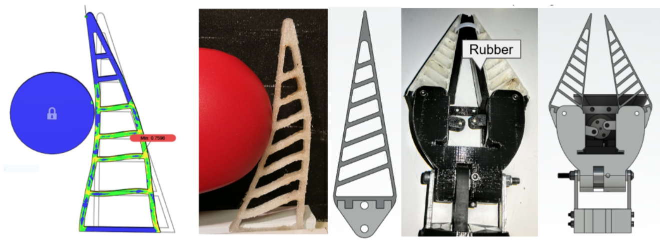

Fig. 3: Antare's actuator's hook.

figure 4, we can also see the model developed in *Fusion Autodesk Sofware*:

The new arm is composed of 6 rotation joints, giving the robot a greater degree of freedom compared to the previous arm and the use of stepper motors in these joints aims to provide the desired smoothness, but the height of the implementation still follows [3] as in the previous version.

A virtual model has already been developed for use in the online competition of the year 2021 [2], presenting a very satisfactory performance where the robot managed to obtain the first place in the general classification.

That said, with the return to face-to-face competitions, the challenge is being:

- Assemble the arm correctly;
- Adjust the rest of the robot's electronics and mechanics for the new arm;
- Perform the correct movement of the arm, in a robust and smooth way;

If there is slack in the schedule, it is still expected to control the position of the arm with encoders already designed and attached to the robot, thus expecting a more precise movement.

After carrying out all these steps, it is expected that the robot can present a satisfactory performance in handling objects, which can contribute to a good placement in the general classification.

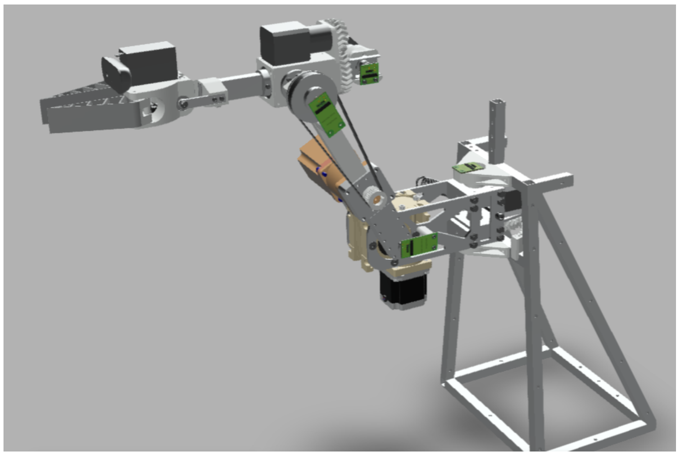
Fig. 4: Antare's robotic arm manipulator.

III. SOFTWARE

A. *Docker*

One of the most important aspects in the software development this year was the full implementation of the Docker tool in our teams, docker is a tool developed to provide a OS-level virtualization in the form of containers and utilises the kernel features like *cgroups* and *namespaces* to isolate processes [4]. The docker images generated this year made possible not only to the robots to start running some of its skills inside a docker terminal but also for the new developers of the team to quickly start working on the software without the need to install all the dependencies, providing a standard environment for the development and execution of the system's different modules. This has ended the problem with different software libraries versions and hidden dependencies.

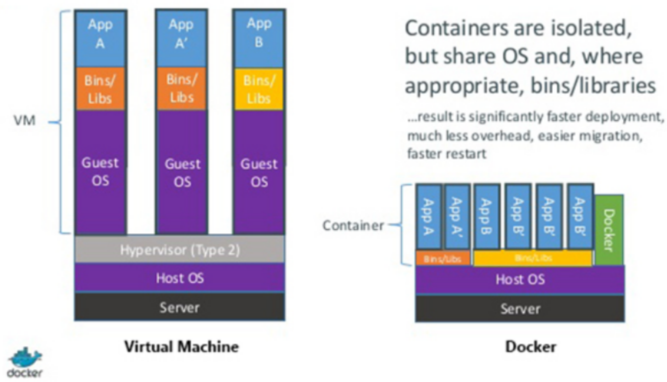

Fig. 5: Example of how docker share the machine resources versus a virtual machine. Image acquired from [5]

B. *Object Detection*

The Object detection skill of the system is a challenge of haste every year since the objects that the system need to recognize may change. This means that the machine needs to be retrained with a new set of images to perform ideally. As there are no available free datasets with all the items we may need, the usual process is that after the item discovery phase during the competition the team gathers the maximum amount of photos they can and label them with bounding boxes to the respective items that need to be recognized.

This process takes a lot of time from the team, time that is essential in the tuning of the robot in other skills as well, so the development of a process that is as fast as possible and can be done simultaneously by as many people as possible give us a head start to work in more pressing matters.

The process has three major roles, *labelers*, photographers, a dataset leader and a machine learn specialist.

Firstly the dataset leader will buy the objects that need to be recognized and bring them with a physical machine, for backup and upload of the photos to a *Cmputer Vision Annotation Tool - CVAT* server with him to the room. The dataset leader then proceeds to place the table in its position and mark 9 (nine) positions for the photographers to stand on while taking the pictures. This positions must be in three different distances and three different angles forming a cone in front of the table.

The dataset leader also need to prepare the room with multiple tables, facing different walls, with different tablecloths so that the process is simultaneous and provides pictures with different backgrounds. The light of the room must also be controlled and vary during the photo session.

The dataset leader will then define the combination of objects that each photographer will picture. Those combinations vary in size from one single object to 10 different objects at a time.

Then the photographers, with there own personal cameras, will one by one take the combination assigned to them, position the combination on the table forming a simple disposition of items and take a picture from each one of the nine positions marked in the ground, after taking those nine photos the photographer will change the disposition rotating the items and take the photos again, for a third time and now creating occlusions if possible the objects are rotated and the nine pictures are taken.

During this process we guarantee that the following aspects are not constant:

- Photographer
- Camera
- Camera Openness
- Light
- Background wall
- Tablecloth
- Object angles
- Object position
- Object Number
- Object Occlusion

From time to time the dataset leader will stop the session, gather all photos in the backup, begin a upload routine, vary the light of the room and re-randomize the combinations.

Fig. 6: Example of labeled images.

While that's happening, as photos are uploaded, the *labelers* already have access to them via the CVAT server and can begin labeling them. The labeling process is done throught the CVAT tool with bounding boxes, much likely to any image labeling tool, despite it being well documented with a lot of team's guidelines on how to do it, there is no need to a explain it in depth since labeling is a well know process in every supervised machine learning project, figure 6 exemplifies the tool utilization.

After labeling the labels and images can be exported from CVAT to the machine in which the machine learn specialist will perform the training.

As stated in section III-A the usage of docker was extensive this year, this includes the script and image to put the CVAT server online, which will facilitate this process for the following years and can facilitate for other teams to use the same tool and process.

The machine learn specialist then can choose a implementation of one of the state of the art architectures for object detection: MobileNet [6], ResNet [7], Inception [8] or Yolo [9]. Our to go decision this year was Yolo since its performance last year with images that were taken in a earlier version of this process granted us the first place in the object detection task.

The advantages of this process are that it only needs two students with actual understanding of how datasets works, since CVAT is easy to use even for those of other areas, and that all the phases can happen simultaneously. Also the amount of photos that can be categorized is enormous and it only depends on how many volunteers you can get to become your photographers and labelers with a low effort of the team itself.

IV. CONCLUSIONS

In this paper, we presented our team's effort to improve our robot hardware and mostly software organization, distribution and processes. The implementation of docker has proved itself valid since more graduate members are now able to influence the coding and the object detection dataset generation has already given results since its first version try in the online competition of 2021. For future work we expect to manufacture the new manipulator design and transfer more of our tools into containers inside our main docker image.

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] casa.com.br, "Mesas e cadeiras para uma sala de jantar cheia de estilo," 2017. [Online]. Available: https://casa.abril.com.br/moveis-acessorios/mesas-e-cadeiras-para-uma-sala-de-jantar-cheia-de-estilo/

[4] B. B. Rad, H. J. Bhatti, and M. Ahmadi, "An introduction to docker and analysis of its performance," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 17, no. 3, p. 228, 2017.

[5] E. Preeth, F. J. P. Mulerickal, B. Paul, and Y. Sastri, "Evaluation of docker containers based on hardware utilization," in *2015 international conference on control communication & computing India (ICCC)*. IEEE, 2015, pp. 697–700.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.