Ararabots@home 2022 Team Description Paper

Edson T. Matsubara, Luiz H. S. Santos, Marina P. Barboza, Nicolas S. Y. Kaihara, Rafael L. P. da Silva,
Rafael T. Nantes, Raphaella B. Jacques, João V. da S. Santos, Arthur H. A. Farias
Mauro B. de Paula, Gabriel L. da Silva
Laboratório de Inteligência Artificial & Faculdade de Computação
Universidade Federal de Mato Grosso do Sul
Campo Grande, Brasil
ararabots@gmail.com

***Abstract*—In this paper, we provide a full description of the project in the progress of AraraBox, an autonomous system for domestic use developed by a new team of the Artificial Intelligence Laboratory of the Federal University of Mato Grosso do Sul. This paper describes the work of AraraBots during the year to develop an autonomous robot, qualified for the tasks in this category. We discuss various techniques to overcome the challenges posed by the competition and the functions implemented or to be implemented, such as facial and object recognition, recognition and emulation, object object manipulation, and navigation and mapingation.**

I. Introduction

A participation in RoboCup Competition phase in Brazil, hosted in Latin American Robotics Competition (*LARC/CBR*), is needed to help develop Robotics, Artificial Intelligence and related areas by sharing knowledge and learning from other more experienced competitors. Since 2019, our research team has been working on the development of a autonomous low-cost domestic robot aiming to perform all the tasks from RoboCup@Home category and this paper provides information on how we plan on doing this.

Em 2019, we participated in the competition at Rio Grande-RS, we will take a closer look at the challenges we had to face in order to reach 7th place, analyzing the issues found and describing ways to solve them, applying not just previous robotics knowledge, but also machine learning algorithms, computer vision, natural language processing, etc. The last ones being core areas of research of our laboratory.In 2020 and 2021, due to difficulties encountered during the pandemic, our team was unable to participate in the competition, in this category.

In this paper we discuss and show partial results and expectations of projects under development and to be developed. First we will give a brief history of our team and workplace, then we explain the features and software included in this project, with the goal to enhance our skills and refined our system to be more competitive alongside the other teams.

II. Ararabots Team Brief Background

Created in 2010, the Ararabots team emerged inside our AI research lab named LIA(Laboratório de Inteligência Artificial) [1], hosted at UFMS, in a way to put in practice some techniques used inside the lab and expand our work into robotics. The main goal was to develop new technologies in programming and hardware development. Since 2011 our team participated in almost every *LARC/CBR* competition hosted around our country, mostly in categories like *IEEE VSSS* or *IEEE SEK* achieving third place, on both in 2012.

Fig. 1. Throphies acquired in 2012 *LARC/CBR*

We continued to participate in *VSSS* in the following years, in 2018 along with *VSSS* we returned with *SEK* category and we achieved 9th place and 15th place in each category. In 2019, together with *VSSS*, we reacched the 8th place,in that regard, this results shows that we try to get better every competition. In 2019 yet, the team competed for the first time at Robocup@home, accordingly to our experience, we will try to use state-of-the-art algorithms and develop the new technologies that integrates them.

III. Robot Features

This section describes separately each module of our project, and its features and peculiarities. Initially the project was divided into five parts: vision system, navigation, question answering, voice recognition and voice emulation. Each one of them are more specified below.

A. AraraBox System

We decided to use *Python* as the base programming language since we already have a previous knowledge and still use in another category along with the *ROS system* [2] [3]. That said, we plan for *ROS* to manage resources for every vital part of our project without deadlocks and transmission problems using the tools available through the system.

It receives camera images for object and face recognition, point cloud from Xbox Kinect in order to map the ambient, voice commands and output values to move the motors and the gripper accordingly and voice emulation data in response to commands.

It is a challenge to continue our goal of developing a low-cost robot, we are looking for the best ways to perform the tasks so they are cost-effective and work optimally so we can help make these technologies more accessible.

B. Vision System

The vision system uses a high definition 720p webcam to acquire frames that are used in a python-developed subsystem, that makes a threaded camera read of frames, trying to diminish I/O blocking operations freezing the system, and does series of processing steps in order to do tasks like object detection, people counting.

1) ***Object Detection****:* We intend to use MediaPipe [4] to solve Object Detection task. Created by Google engineers, Mediapipe is an open-source framework that offers cross-platform, customizable ML solutions for live and streaming media.

The main reason we opted for MediaPipe is because it's fast and have great performance in limited hardware.

2) ***Face Recognition****:* For the face recognition task, we intend to implement by using an open-source GitHub library called "Face Recognition", by ageitgey [5]. The repository is created using the C++'s dlib face recognition library built with machine learning tools to solve real world problems. With this library, we can easily find faces in a picture and even the facial features of the face detected. We can also manipulate those features for things like Instagram filters.

For the machine learning, we can separate a folder of known faces with pictures with faces with the file name of the person we want the machine to recognize. When combined with the webcam on the Ararabox, we can loop through the frames of the video and keep detecting the faces as we can do with a single picture.

C. Voice Recognition

For voice recognition we decided to use DeepSpeech, an open source Speech-To-Text engine, using a model trained by machine learning techniques based on Baidu's Deep Speech research paper. Project DeepSpeech uses Google's TensorFlow to make the implementation easier.

We intend to use the version 0.9.3 to transcribe the audio captured by a microphone to a text file, which it´s intend to be used in the area of question answering.

D. Question Answering

In this section we will describe our approach to solve the question answering challenge proposed. Our idea to accomplish this task is to combine the power of current Natural Language Processing(NLP) models, and googlesearch API [6] to contextualize the question. The model we choose to use in this case is the *Huggingface's Transformers* [7] which provides state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, T5, CTRL...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over thousands of pretrained models in 100+ languages and deep interoperability between PyTorch and TensorFlow 2.0. Figure III-D shows the base architecture.

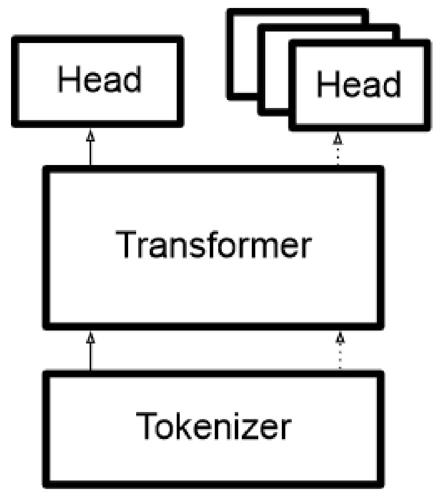

Fig. 2. Base Transformers [7] architecture.

We intend to use the RoBERTa base model with fine-tuning in SQuAD2. That being said, SQuAD2 is a datased containing 100000 questions plus 50000 unanswerable questions. Therefore, our robot will learn to answer all of those questions and identify which ones are unanswerable. That said, we will fine tune to our own dataset, based on the command generator published by the organization on GitHub.

Here is an example of this network in use. It was provided the first paragraph of Brazil's page on Wikipedia and asked a question about a topic that was approached in the summarized text. The result can be seen in figure III-D.

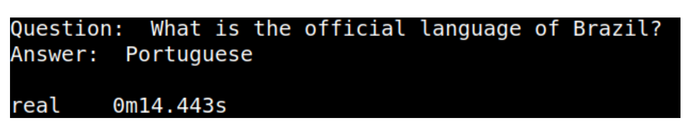

Fig. 3. RoBERTa base model example.

We are still studying how to improve the model to do the computation in a shorter time.

E. Robot Voice Emulation

This section is still being studied and we intend to use a text-to-speech *ROS* package or Google text-to-speech API to emulate voice dialogues to produce a better human robot interaction. This task will use a ROS node to translate phrases into voice, answering received voice commands.

F. Mapping and Navigation

The navigation of a robot in an unknown environment is a problem already known and solved in the literature through a technique known as SLAM (Simultaneous Localization and Mapping) [8]. As the nervous system of the robot was used *ROS*, so it is kind of immediate that the same would be used as a SLAM tool. *ROS* offers some tools already implemented that perform this task very efficiently, one of them gmapping, which was the first approach used. However the chosen *ROS* version (Melodic Morena) at the time still did not support gmapping, so we tried another tool, called RTAB-Map [9].

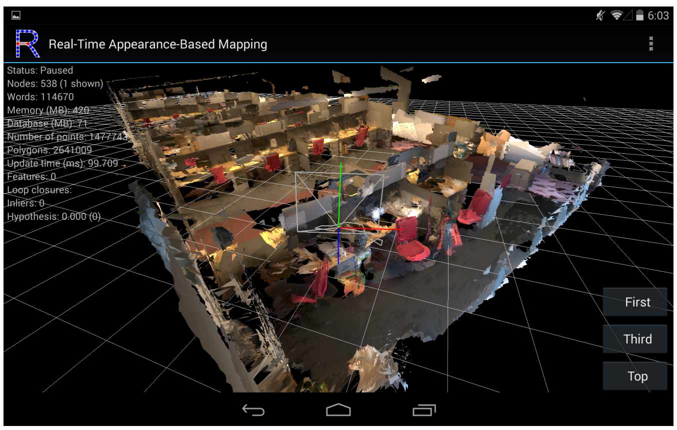

Fig. 4. RTAB-Map mapping tool example

The RTAB-Map (Real-Time Appearance-Based Mapping), a RGB-D SLAM approach based on a global loop closure detector with real-time constraints. The loop closure detector uses a bag-of-words approach to determinate how likely a new image comes from a previous location or a new location. When a loop closure hypothesis is accepted, a new constraint is added to the map's graph, then a graph optimizer minimizes the errors in the map. It's proved to be efficient given the mapping and localization functions already implemented, and totally compatible with the *ROS* version. To obtain the point cloud used by the RTAB-Map we used Kinect, more specifically the XBOX-360 version.

Fig. 5. the Xbox 360 Kinect RGBD sensor

G. Hardware Design and Implementation

This section explains aspects related to the robot hardware, below hardware components of the robot are detailed in a succinct way:

AraraBox has an Arduino Uno equiped with ATMEGA328P [10] processor, which is the microcontroller responsible for the two hoverboard motors used to move the robot, the motors have an encoder and a driver. The circuit power is supplied by external power, by 3 Makita Lithium-Ion batteries of 18V and 3.0Ah each; A Thinkpad Lenovo notebook, with intel core i5 and 8GB RAM memory; an X-BOX 360 Kinect as RBGD sensor; a Samsung Galaxy Tab4 Tablet to emulate a face and give a better human–computer interaction; an external Microphone to get voice commands; a Speaker to reproduce voice; a Microsoft Lifecam Cinema 720p HD.

About the motors, they have four controller pins that will be used to control the movement of the robot: start/stop, brake, rotation and speed setting. Their functions are described below:

- start/stop: this signal is responsible of starting the motor controller.
- brake: when active(ON), the motor completely stops it's rotation acting like a brake.
- rotation: this signal is used to indicate the rotation direction of the motor.
- speed setting: this signal is used with PWM(Pulse Width Modulation) to control the motor's current speed.

The signals start/stop, brake and rotation are active in low voltage (voltage level of 0 to 0.5 V indicates the signal is ON and 4 to 5 V indicates it is OFF).

An Arduino Uno board will be used to send signals to the controllers pins. It will receive packets with commands codified in bytes from the main computer, decode and transform its message into electric signals to control the motors. All communication board-PC will be done using serial communication.

Since we are not using a pre-assembled base and structure, we made a design cheap, simple and functional with room for improvements. The base assembly is completed and working, and we are building the remaining structure.

H. Robot Manipulator

The robot manipulator will be developed using servomotors, stainless steel pieces produced with a CNC machine and 3D-printed parts. Projects, now, includes a 5 degrees of freedom (DOF) robotic arm with a gripper able to manipulate objects using some techniques imitating the human arm as perfect as possible.

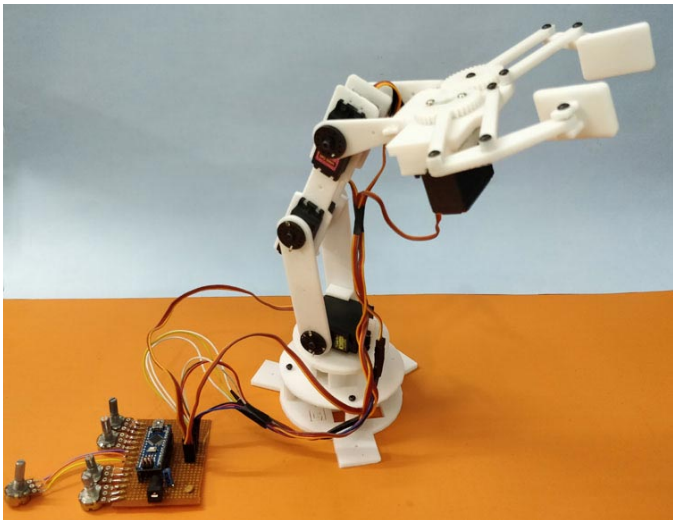

Fig. 6. 3D Printed Robot Arm, used in some gripping projects using RL in our LAB

For gripping tasks the robot can use some inverse kinematic [11] algorithms to calculate joint positions to reach some point in space. These calculated by other robot systems. For motion and control we should use an Arduino UNO with some servo motor shield.

IV. Conclusion

Using what was described in this TDP, AraraBox, alongside the Ararabots@home members, will try to solve the problems proposed at the Robocup@home rule book for 2020. Development is incremental and as future work we can say a fine-tuning of things developed during the year, and start to build own ML algorithms for the robot.

Acknowledgment

The team greatfully thanks the teachers Dr. Edson Takashi Matsubara, Dr. Eraldo Rezende and Me. Glauder Guimarães for the support and space assigned to the work during the year and to the Faculty of Computing and the Federal University of Mato Grosso do Sul for the opportunity of representing them through Ararabots. We also thanks all students named as authors for all the work in progress in this project.

AraraBox Hardware and Software Description Summary and Team Information

The hardware and software used will be briefly described in the subsections below. We intend to use algorithms that solve some functionalities such as navigation; global task planning; recognition of voices, objects and faces; sensors; control brushless motors among others techniques and apparatus. We covered both, low and high controls together aiming the best human-robot interaction.

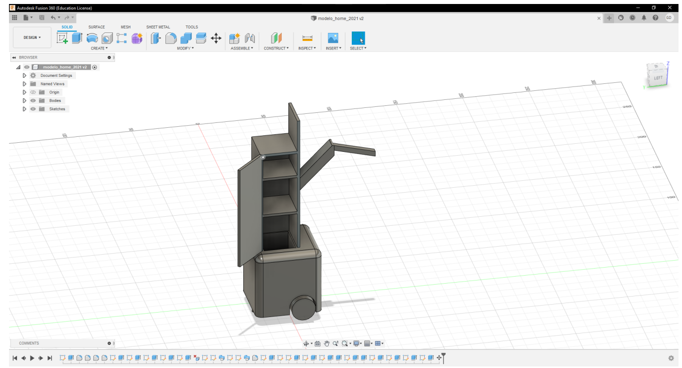

Fig. 7. Robot prototype.

A. Hardware Description

AraraBox has two drive wheels and one free wheels, positioned in the rear to give balance. The body still in development and all the definitive electronic parts will be protected and well built to avoid short-circuits and modular to be easily manipulated.

- Base: One Arduino Mega 2560; Two hoverboard motors with drivers, One Thinkpad Lenovo notebook, with intel core i5 and 8GB RAM memory; Three Makita Lithium-Ion batteries of 18V and 3.0Ah each.
- Torso: One Analog switch button; One Xbox-360 Kinect; One Microsoft LifeCam CInema 720p HD.
- Arm: Undefined.
- Head: One Samsung Galaxy Tab 4 tablet; One microphone; Two speakers.

B. Software Description

Firmware(low level) control is running on a Arduino Mega board and the AraraBox software system is based on Robot Operating System(ROS) alongside with Machine Learning techniques.

- Operational System: Ubuntu 20.04.4 LTS (Focal Fossa).
- Middleware: ROS Noetic Ninjemys
- Localization, navigation and Mapping System: SLAM; ROS RTAB-Map
- Object Detection: MediaPipe
- Face Recognition: Face Recognition library
- Question answering: RoBERTa
- Speech Recognition: DeepSpeech.
- Speech Generation: Undefined.

References

[1] LIA, "Lia facom - homepage," 2018, accessed on 2022-06-28. [Online]. Available: http://lia.facom.ufms.br/ararabots/

[2] ROS, "Ros - robot operating system," 2018, accessed on 2022-06-28. [Online]. Available: https://www.ros.org/

[3] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3. Kobe, Japan, 2009, p. 5.

[4] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," *ArXiv*, vol. abs/1906.08172, 2019.

[5] "Face recognition," 2017, accessed on 2022-06-28. [Online]. Available: https://github.com/ageitgey/face_recognition
[6] "googlesearch," 2020, accessed on 2022-06-28. [Online]. Available: https://github.com/Nv7-GitHub/googlesearch
[7] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.
[8] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
[9] M. L. Introlab, "Rtab-map," 2019. [Online]. Available: http://introlab.github.io/rtabmap/
[10] A. Atmega328, "P datasheet complete, 2016-06," *Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_datasheet.pdf[Hämtad: 2016-09-29]*, 2016. [Online]. Available: http://www.atmel.com/Images
[11] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 298–303.