Team Description Paper UFPel PinguimBots @Home LARC/CBR 2022

Kristofer Stift Kappel, Ariston Souza Monteiro, Arthur Alves Cerveira, Felipe Machado,
Vinicius Alves Hax, Isabele Scherdien, Arthur Sandim, Pedro Henrique Mesquita,
Felipe de Souza Marques and Paulo Roberto R. Ferreira Jr

***Abstract*— The Home-Environment Technological-Agent (Theta) is the robot developed by the UFPel PinguimBots @Home team. The frame of the robot is a modified wheelchair, manufactured by the company Freedom Veículos Elétricos, which supports the project. To compete at the league and be able to accomplish the tasks, Theta is equipped with a set of sensors and processing devices. Thereby, the equipped hardware enables Theta to have multiple functions to execute the competition's tasks. Some of the functions Theta is capable of are mapping, navigation, self-localization, skeleton tracking, object and face detection, and Human-Robot Interaction. Due to the Covid-19 outbreak, the team is not currently working on upgrading the physical aspects of the robot, but mainly focusing on recognition and simulation tasks.**

I. INTRODUCTION

The recent growth of robotics and artificial intelligence over the past years led to the searching of solutions for a wide array of problems. Most of these challenges are related to the automation of domestic tasks, seeking the increase of comfort in the home environment.

Aiming to approaches these challenges, The Home-Environment Technological-Agent (Theta) is a domestic robot designed by the UFPel PinguimBots@Home team in partnership with the company Freedom Veiculos Elétricos LTDA. In order to push the state-of-art of service robotics and autonomous wheelchairs at the same time, Theta is designed over a wheelchair provided by the company.

To compete at LARC/CBR, Theta has additional sensors and processing devices which enables the completion of the competition tasks. A YDLIDAR X4 and a pair of rotary encoders enable Theta to perform localization and mapping. A Microsoft Kinect provides RGB and Depth information to accomplish tasks related to computer vision. Lastly, Human-Robot Interaction (HRI) is a result of two different interfaces, voice interaction and feedback through facial emotions. A microphone and a speaker are responsible for voice interaction. Furthermore, the visual feedback is displayed on a monitor positioned vertically. The initial version of Theta can be seen in Figure 1.

Fig. 1: Theta version 1.0

Regarding processing devices, Theta has a laptop computer and a Arduino UNO unit. The laptop runs Ubuntu 16.04 and ROS Kinetic [1]. Thus, ROS is essential to run and traverse information through multiple applications on a robotic system. Some of the tasks designated to ROS are mapping, localization, navigation, speech recognition, voice synthesis, computer vision, and changing the facial expressions. The main usage of the Arduino UNO is to converter and transfer odometry data from the rotary encoders to the computer, and transfer the velocity messages generated by the computer to the controller of the motors.

This paper is organized as follows: Section II briefly describes the team. Section III presents Theta's construction and navigation modules. Section IV introduces the approaches to computer vision tasks. Section V elucidates the Human-Robot Interaction. Section VI talks about the simulation. Section VII draws conclusions. Lastly, Theta's hardware and software are summarized.

The work of the team was financed and supported in part by Freedom Veículos Elétricos LTDA.

Programa de Pós-Graduação em Computação (PPGC), Centro de Desenvolvimento Tecnológico (CDTec), Universidade Federal de Pelotas (UFPel), Pelotas, RS, Brazil kskappel@inf.ufpel.edu.br

II. UFPel PinguimBots @Home

PinguimBots[1] is established at the Universidade Federal de Pelotas (UFPel), at Rio Grande do Sul, Brazil. The first appearance of the @Home team in a competition was at LARC/CBR in 2019, in which the team earned fourth place. Furthermore, in 2020 the team competed the second time at LARC/CBR in the @Home virtual event.

The PinguimBots@Home team was born out of a partnership between UFPel and a wheelchair manufacture company called Freedom Veiculos Elétricos LTDA. The main goal is to research and develop technologies for robotics on a domestic environment, taking advantage of the task-solving format of the competition, and further applying those technologies to actual wheelchairs.

To achieve easier integration between the technologies developed and the wheelchairs, the base frame which composes Theta consists of a slightly modified wheelchair manufactured by the company. Consequently, it retains the joystick which sits originally on one of the armrests of the wheelchair. Hence, every velocity message passes through the joystick. This decision aims to simplify future integration with Freedom products. In return, Freedom supports the team by providing a workplace, a part of the equipment, and materials.

The teams is currently in a process of redesign, both related to the members that make up the team as the overall structure of the robot. This decision relies on the fact that after the COVID-19 outbreak most of the member are having their first touch with the physical robot. As the team get used to the robot many ways to upgrade it start appearing. Given this, at the time, most of the changes still under development and testing

III. THETA'S DESCRIPTION

The current section introduces details relative to Theta. The main source of processing power is a laptop computer aboard of the robot. The specifications of the computer are as follows: an Intel Core 2 Duo T6670 processor, 2 GB RAM DDR2 800 Mhz, with an Intel Graphics 4500M GPU. The notebook's operating system is a Ubuntu 16.04 and runs ROS Kinetic Kame [1]. ROS plays a crucial role when working on a robotic project, once it provides features like message-passing, language independence and modularity for the concurrent development of different modules that will be interconnected.

Subsection III-A gives a brief introduction about Theta's structure and base construction. Furthermore, subsection III-B explains navigation and movement.

A. Base Frame

Theta is a robot designed over the frame of a wheelchair, manufactured by Freedom Veículos Elétricos, company which supports the team. After modifications to reshape the wheelchair to have a robot-like shape, as can be seen in Figure 1, Theta can no longer carry a human. However the design focuses on completing home-environment tasks. The frame has been updated to a smaller and lighter version. However, due to the pandemic we were unable to display an updated picture.

Theta is equipped with two pairs of wheels. A pair of motorized wheels on the back, and a pair of swivel wheels on the front to grant stabilization to the robot. One of the design changes to be tested in the future is switching the robot to front wheel drive and evaluating changes in stability and traction.

The joystick that sits on top of one of the armrests of a conventional Freedom wheelchair remains as a movement interface on the robot. Thus, making it simple for future integration of technologies with other wheelchair. Therefore, the wheelchair can be controlled in two different ways.

The first is directly by the joystick. Placed on a spot of easier access, Theta can be moved effortless by interacting with the joystick.

At the same time, the computer can control the wheelchair frame by publishing velocity message unto a certain rostopic. The Arduino unit then receives the message and translates it to PWM signal and sends it to the joystick controller which acts upon the wheels.

The same Arduino UNO is responsible for receiving pulses generated by both rotary encoders, translate it to odometry message and publish those messages unto the appropriate rostopic.

At the moment, this arduino module that controls the chair is being redesigned, the code is being refactored, and document, and a new circuit is under development.

Regarding energy to sustain the devices, Theta drains it from a standard Freedom wheelchair battery. Also, a fail-safe button, placed on an accessible spot, cuts the power off from the motors and activates the breaks in case of an emergency.

B. Navigation, Localization and Path Planning

The navigation task consists of, given a map that has the geometries of the environment, and the measurements of the robot, it needs to move from a starting position to a destiny point, while localizing itself in the path [2].

To create such map of the environment, called occupancy grid map, Simultaneous Localization and Mapping (SLAM) is often used. The team uses the node slam-mapping, from Gmapping ROS package to map the scenario. It takes a laser scan, given by the YDLIDAR X4 and odometry, given by the rotary encoders, as inputs, and outputs a occupancy grid map. An example is shown in Figure 2.

To localize itself, Theta uses Adaptive Monte Carlo Localization (AMCL). AMCL is a classical localization algorithm available at the ROS Navigation Stack. It is a probabilistic localization algorithm that uses a particle filter to estimate the position and orientation of a robot on a known 2D map, as shown in Figure 3.

Theta's path planning rely on the move-base node, which is a configurable node that serves as an interface for the

[1] `http://www.pinguimbots.com/`

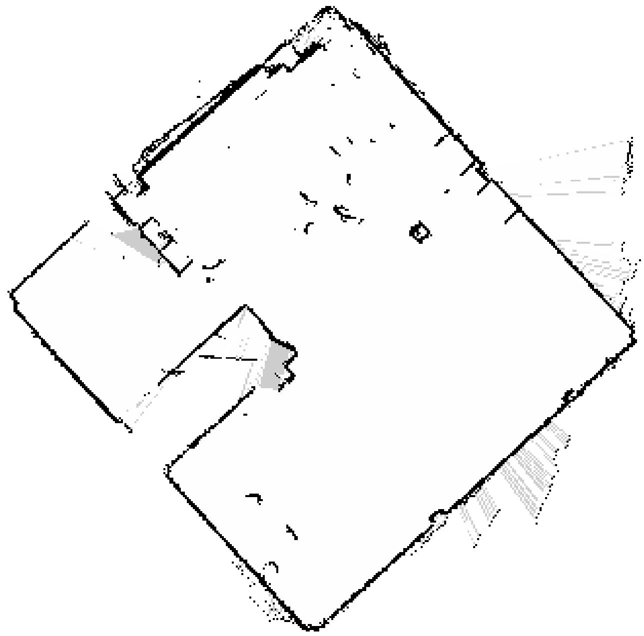


Fig. 2: A map from Freedom's lab obtained in real-time using gmapping package.

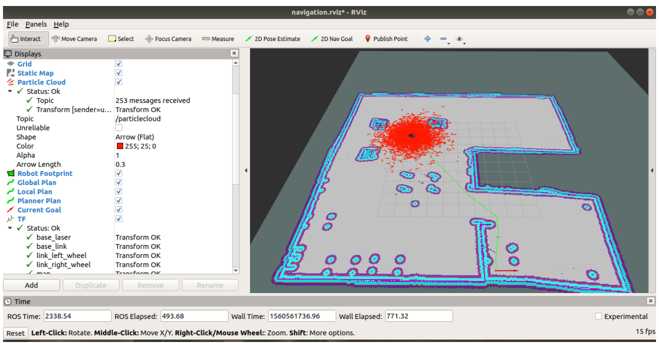


Fig. 3: Localization using Adaptive Monte Carlo Localization

robot to connect with other functionalities within the ROS Navigation Stack, such as the global and local planners.

RViz is a core tool to visualize instances, such as, the occupancy grid, costmaps, robot frame, and sensors readings, and to interact with parameters on a graphical fashion. RViz is also used to simulate and estimate the robot's pose during navigation. An example of the RViz interface running AMCL can be seen in Figure 3.

IV. Robot Vision

Computer Vision is essential for a domestic robot since it enhances the ability of the robot to perceive its surroundings and aids in decision making.

The following subsections will display some of Theta's computer vision capabilities.

A. Object detection

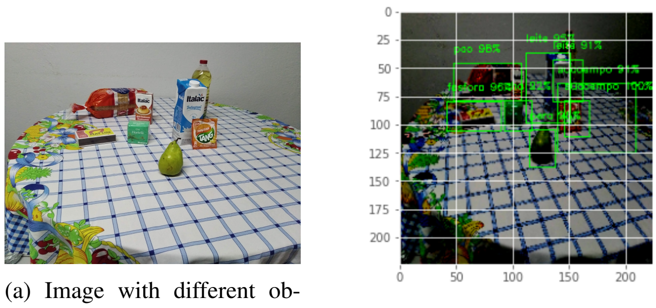
Object detection is an important skill for the robot to be able to perform tasks like manipulating objects or recognizing scenes.

Regarding object detection and recognition, in the 2020 LARC/CBR edition the team applied machine learning with a ResNet 18 Neural Network using the PyTorch framework. The approach was based and enhanced on top of the one presented on [3]. In preliminary tests it presented satisfactory results. However, the performance during the competition presented errors such as low precision rate, shifted bounding boxes, which led us to try a different approach.

Hence, the team changed to an approach based on an R-CNN model [4]. As the basis, the team used the content available on [5], which utilizes the Keras library over the TensorFlow Platform. Furthermore, the application available on [5] was adapted to learn multiple classes during training.

The model consists of an R-CNN model pre-trained over the imagenet dataset. The input is a RGB image with a 224x224 dimension.

The training dataset is made up from images gathered to train the model for the task of the 2020 LARC/CBR edition. Thus, the images contained objects from fifteen different classes (Figure 4). Later, images from the 2021 edition were also added to the training dataset.



(a) Image with different objects used for training.

(b) Objects after classification.

Fig. 4: Images before and after classification.

After a few tests training with different sets of hyper-parameters, the following values presented suitable results: an initial learning rate of 0.0001, 100 epochs, and a batch size of 16 samples. It is worth noticing that the tests are ongoing and the team aim to enhance the results. The Loss and Accuracy curves can be seen on Figure 5. On the tests, after one hundred epochs the loss value stopped significantly decreasing.

B. Face Detection

Face Detection is especially useful to identify whether a potential operator is nearby, as well as looking for known users of facial expressions.

To detect faces on the image, either having masks or not, a Deep Neural Network (DNN) was applied. We fed the DNN with images containing labeled faces. The model is capable os identifying faces in new images and classify them in two groups: having a not on or not.

We utilized a MobileNet as a base for the face extraction model through two models, MTCNN for images and OpenCV DNN for videos. The structure of MTCNN is

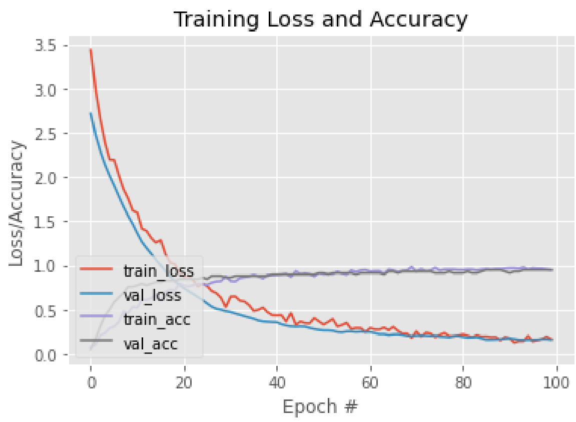


Fig. 5: Loss and accuracy curves for training and validation datasets.

composed of three stages, a P-net that gather the potential candidate windows, followed by an R-Net that regressively selects the border of those windows. Lastly, O-Net gives the location of the faces found. Moreover, OpenCV DNN uses SSD Caffe from the Python library DNN, built over a ResNet10.

Regarding the classification task the team used the Keras framework and a dataset of aout 1350 images where half had mask and half didn't. Thus, the masks were artificially generated with Python[2].

C. Follow Me

Following someone is an important function for a service robot on a domestic environment and is a typical @Home task. Through Kinect it is possible to obtain an RGB image from the camera and depth information from the infrared sensor, useful for detecting a person and its position. OpenNI enables communication from the Kinect through ROS and Nite allows to access Kinect's functions such as skeleton tracking and gesture recognition.

After calibrating with a human user through the detection of a characteristic pose, Nite publishes a set of transforms indicating a set of axis detected for that specific user. Through the transforms it is possible to get information about the current position of the user and follow them from a safe distance.

V. Human-Robot Interaction

To proper interact with a human, the robot must be capable of maintaining a level of communication with him, this can be done using either visual, or audible feedbacks. However, this communication must occur fluently, without causing the user frustration or anxiety when giving a command to the robot or waiting its feedback.

1) Facial Expressions: A set of animated facial expression is the main way for Theta to communicate emotions or statuses to the user through visual feedback. To keep the interaction fluent, Theta can change its own emotional state, e.g. being happy when completing a task, sad when something goes wrong, or a thinking face when processing the next step. Additionally, Theta is able to present its current status using visual feedbacks, such as, a loading screen when booting up or an sleep screen when it has not interacted with anyone in a while.

To engage the user to keep interacting with Theta, the changes on the facial expressions follow an Affective Loop [6]. By constantly changing to reaction that better fits the appropriate situation the user tends to feel encouraged to interact.

Fig. 6: Some of the facial expressions that theta can reproduce

Theta's face module subscribes to a specific topic in which other nodes can publish to. This way, Theta can provide visual feedback despite the task that is reporting the current status.

We are constantly adding new faces to fulfill needing of new tasks regarding human-robot communitacion.

2) Speech Recognition and Synthesis: Theta uses two different speech recognition tools: the Python library SpeechRecognition with the Google Speech = Recognition engine, which is engaged whenever Theta has internet access, and the open source tool Vosk, which works offline.

The Google Speech Recognition (GSR) engine has a superior performance when compared to Vosk. However, Vosk has an adaptable recognition model.

After evaluating the team decided to use the "small" Vosk model, which allows the dictionary to be extended with new words. This is a desirable feature once some questions may

[2] `https://github.com/prajnasb/observations`

contain, e.g., words in other languages.

In the LARC/CBR @Home speech recognition task the team usually have a list of possible questions. To enhance the recognition in this task, a Levenshtein Distance algorithm is applied to better match the outputted text to one of the possible questions. The algorithm works by calculating how many changes are necessary on a text to transform it into another. This process is done for every question on the list and the one with the smaller number o changes needed is probably the right question.

Theta is able to answer the desired question outloud using Google's voice synthesizer (GTTS). GTTS is a library that requests from the Google API and return an audio file with the synthesised phrase. The audio is then played using playsound library.

During the research and implementation stage, the team tested an array of other tools and libraries for voice recognition and synthesis. One of the main concerns was a hot-word detection tools, such as Porcupine from Picovoice. Thereby, Theta would not need to hear and process a large sequence of audio, but only start listening when a keyword is recognized. At the end, Porcupine was discarded due to low adaptability on its free version, not allowing a custom keyword to be set.

Picovoice also has a speech-to-intent engine called Rhino. Rhino enables the possibility of recognizing custom commands, e.g., "Turn on the living room light". This can be done by going to the online console, following Picovoice's syntax, training and exporting an YAML file and feeding it to the engine. The implementation was impracticable due to the command structure and the recognition being comparable to Vosk in empirical tests.

Tests with voice synthesis involved LianeTTs and Festival. LianeTTs has Brazilian Portuguese output and Festival has multilingual options, including Portuguese. There are multiple voices available for Festival. However, the tested voices, Festival and LianeTTs default ones, produce an over-robotized voice, which makes the HRI less friendly.

VI. CONCLUSIONS

This technical description paper details The Home-Environment Technological-Agent (Theta), a service robot developed by the UFPel PinguimBots @Home team. Theta is designed over a wheelchair manufactured by Freedom, a company that supports the project. Theta is currently able to perform tasks, such as, navigation, face and object detection, following a person, speech recognition and synthesis.

As future works we intend to redesign Theta to fix flaws found is the current robot frame, equip it with a manipulator, add more sensors, and improve HRI. Finally, as the main research topic, we wish to transfer what was developed on Theta, to an autonomous wheelchair that is able to navigate autonomously and support the user on a domestic environment.

REFERENCES

[1] "Ros documentation," http://wiki.ros.org/, 2018, [Accessed: 10-06-2019].

[2] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System.* " O'Reilly Media, Inc.", 2015.

[3] "Jeremy fix's deep learning lectures," https://teaching.pages.centralesupelec.fr/deeplearning-lectures-build/index.html, 2020, [Accessed: 24-08-2021].

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[5] "R-cnn object detection with keras, tensorflow, and deep learning," https://www.pyimagesearch.com/2020/07/13/r-cnn-object-detection-with-keras-tensorflow-and-deep-learning/, 2020, [Accessed: 24-08-2021].

[6] K. Höök, "Affective loop experiences–what are they?" in *International Conference on Persuasive Technology*. Springer, 2008, pp. 1–12.

VII. SOFTWARE AND HARDWARE DESCRIPTION

In this section, we summarize Theta's hardware and software features.

A. Hardware

- The base consists on a modified wheelchair manufactured by Freedom.
- A pair of rotary encoders.
- A Kinect camera .
- A YDLIDAR.
- An Arduino UNO which communicates the rotary encoders to the computer and the latter to the joystick controller.
- A monitor, a microphone and a speaker to perform HRI.
- A notebook with an Intel Core 2 Duo T6670 processor, 2 GB RAM DDR2 800 Mhz, Intel Graphics 4500M GPU.
- A Failsafe Button that cuts the power off from the motors and activate the breaks.

B. Software

- Ubuntu 16 running ROS Kinect Kame
- OpenCV, TensorFlow and Keras for image recognition tasks.
- ROS Navigation Stack and gmapping.
- Python SpeechRecognition library and Vosk for Speech Recognition tasks.
- OpenNI and Nite to perform skeleton tracking.