Escola Tècnica Superior d'Enginyeria

Electrònica i Informàtica La Salle

Treball Final de Màster

Màster Universitari en Enginyeria de Telecomunicació

DEVELOPMENT OF AN EXPERIMENTAL PLATFORM FOR THE USE IN THE STUDY OF EMPATHY BETWEEN HUMAN AND ROBOTS

Alumne	Professor Ponent
Fernando Martín Valcárcel	Raquel Ros Espinoza

ACTA DE L'EXAMEN
DEL TREBALL FI DE MÁSTER

Reunit el Tribunal qualificador en el dia de la data, l'alumne

D. Fernando Martín Valcárcel

va exposar el seu Treball de Fi de Máster, el qual va tractar sobre el tema següent:

DEVELOPMENT OF AN EXPERIMENTAL PLATFORM
FOR THE USE IN THE STUDY OF
EMPATHY BETWEEN HUMAN AND ROBOTS

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

Abstract

Interactions between human and robots are being more frequent as their role is constantly expanding because of the latest technologies. For these interactions to be as smooth as possible and integrate them on everyday life, a bond between human and robot should be formed. To see this, human and robot empathy should be studied in detail from different perspectives.

This work is centered around the objective of developing an experimental platform that could help the study of empathy between human and robot by giving them a challenge to solve together, the selected one was a maze. Apart from the structure for the maze to be solved, this tool will collect and provide data of different aspects of the test, like emotion of the human in each moment or the number of times that robot and human clash in its decisions.

After doing the evaluation, while this platform still has limitations and a lot of room of improvement, it could be used as a steppingstone to the study of empathy in challenging situations between human and robots. This is possible due to the freedom on setting different parameters of the challenge, such as maze size or behavior of the robot on negotiations which enables the study of the bond in different situations.

Acknowledgements

This is to thank all the people who put up with me and my circumstances.

Index

Acronyms

BT: BlueTooth.

CNN: Convolutional Neural Network.

HRI: Human Robot Interaction

IoT: Internet of Things

IR: Infrared

SVM: Support Vector Machine.

TFM: *Trabajo Final de Master.*

UI: User Interface.

1 Introduction

Nowadays, technology's role in human society is steadily increasing more and more, to the point of being part in all aspects of life. This is no exception for the Robotics area.

Until now, the predominant role of robots in society was for heavy industry that needed to be automated, but now it starting to take a new step. For example, in Japan, there are already robots working as information guides or waiters.

Interactions between people and robots will start to be prevalent in the future, and as such, society will need to find a way to make these interactions as natural and comfortable as possible in order to encourage adoption of such systems in our daily life.

That's why the focus on this work is on the so-called *Social Robots*, entities that are designed to interact with humans.

A more formal definition was done by Cynthia Brezel: *"We refer to this class of autonomous robots as social robots, i.e, those that people apply a social model to in order to interact with and to understand"* [1].

The most important hurdle that a social robot encounters is the perception of human's in which the robot is seen as a tool instead of a companion. In order to overcome that, there are several studies that focus on a system that could define the components in a bond between human and robots and how to strengthen them.

For example, in a study related to the one cited above, they introduced the concept of a component needed in social intelligence, an attention system, which the robot should mimic from the human's way of reading different cues. In other words, giving the robot the ability to process external information like a human should improve the interactions between them. [2]

On the other hand, there are models that directly propose mirroring the current emotion of human, not only to help the interaction but to "*trigger a beneficial cycle that might allow users to reduce their stress levels*" [3].

As a last example, there are also approaches in deepen the bond between human and robot by using the same behavior that a real-life pet companion should do. This kind of robot design approach is seen in some products targeted mostly to children.

In conclusion, for robots to be accepted as much as possible in society daily life, empathy has to be studied from as much as different perspectives as possible.

1.1 Goal

This work’s objective is the design and development of an experimental platform that will enable the study of a cooperative interaction model based on empathy between social robots and humans at a reflective level. This platform will contribute to do that by providing different kind of metrics that could be used to study the subject level of attachment and trust through applied elements of the process of realizing empathy; in other words, extract information from a challenging situation given to a robot-human pair in which they will have to interact with each other to overcome it.

The scenario designed for this study is a maze challenge in which the human and the robot will have to cooperate in finding a path to the exit. There will be two different modes for solving the maze, one by turns where each one takes turns in deciding where they go without having to convince the other party and a cooperative mode in which the two of them should find some sort of agreement on the decision of which path to take.

This kind of challenge was used because it encouraged the interaction between both human and robot, establishing some sort of relationship.

The platform will be measuring different aspects of the test, such as time in each negotiation phase, times a human or robot agrees with each other, emotion felt by the human in every moment, and much more that will be explained later in detail. It will also provide a set of parameters that can change the behavior of the robot and used to measure a different kind of interaction.

1.2 Organization

In Chapter 2, the state of the art is addressed. We will be showing a classification of different kind of social robots depending on the environments where they are used, a brief explanation on empathy and several related works about empathy will be reviewed.

In Chapter 3, all the development related work for this study will be described, starting from an introduction to an overall system, a detailed explanation on the negotiation strategies that the robot will be using on the experiment and the elements used such as the software, robot and interfaces. It will also have a technical explanation of the components that are part of the experimental platform.

In Chapter 4, the evaluation method of the platform, such as time responses, flexibility, etc, and its tests results will be described in depth.

And finally, Chapter 5 and 6 will contain this work’s conclusions and future lines of work that could expand the utility of the platform by giving other kind of inputs, improving the behavior of the robot, etc.

2 State of the Art

2.1 Social Robots

In the Introduction was explained what a social robot is, so this section will be focused on their classification and application in the actual market.

First, all robots can be classified depending on their autonomy degree, which can be:

- Teleoperated. It is fully controlled by an operator and has no autonomy. This kind of robots usually make use of Virtual or Augmented reality and are used for tasks that are dangerous or that require actions that are not possible for humans. An example of this can be seen on figure 1.
- Semi-autonomous: It uses operator inputs, but it can solve some situations on its own. For example, Rovers that are used in space missions are meant to be autonomous but can receive instructions from humans if there is a dire situation that the robot is not prepared to deal with, like getting stuck in a weird way.
- Autonomous: They do not use any kind of operator input in order to solve any situation. For example, Ava's *Ava 500,* which is a telepresence robot that is used for company meetings and is able to navigate autonomously through the different rooms.

Figure 1. ***A teleoperated robot using Virtual Reality*.**

Using the above classification, social robots are designed to have a certain amount of autonomy, if not all, as they must adapt to the human behavior while interacting with them.

Focusing on social robots, they can be classified depending on the environment in which they have a role in. There are four different types: Retail or Service, Home, Health and Education. In this section we have grouped all three last types together.

2.1.1 Retail/Service

In this environment, it is important that the robot can navigate through the scene autonomously. This kind of robots can be used to:

- *Keeping track of inventory*. They can help locate items that could have been misplaced by the customers. An example of that is Simple Robotics *Tally,* which uses computer vision and machine learning to identify product placement, presentation, pricing and availability.
- *Guiding customers on premises*. Robots can be tour guides on a museum or moving information points in airports and train stations. They can also process orders and payments. These kinds require a wide set of languages and good communication skills. An example of that is Softbank's *Pepper,* which has a long range of expressions, uses speech recognition and an autonomous navigation.
- *Retrieving products for customers*. Not only they could locate peculiar items requested by the customer but also get them from inaccessible places. Depending in which kind of establishment they are used, they could also suggest related items. An example is Lowe's *LoweBot*, which is shown in figure 2, this robot, apart from the above also monitors inventory in real time so its data can be used for detecting customer patterns.
- *Delivering products*. They should be able to not only deliver the item but also to interact with the customer in the process. An example is Domino's Pizza DOM, which has an in-built heat and cold system to preserve the food while it does the delivery.

Figure 2. *LoweBot* is an example of Business Social Robot.

2.1.2 Home, Health and Education

In this environment Social robots have the main role of companionship. Isolation and loneliness are becoming a very serious problem in the most developed countries as recent studies have pointed that it could cause long term illness such as diabetes, high blood pressure and mental disarray.

While a robotic companion cannot replace human interaction, it can be used as a relief in some cases:

- *Medicine*. Robots can help staff shortages and act as a companion for the patients. Even some can lift the patients from their bed, for example, RIKEN's *ROBEAR*, which has torque sensors and capacitance-type tactile sensors made of rubber that allow gentle movements, ensuring that the robot can perform those kinds of tasks without endangering them.
- *Elderly*. This kind of robots should be aimed for a well-aged niche and their function will be not only companionship but also health monitoring, cognition and motor stimulation. An example of this is Luvozo PBC's *SAM,* which provides frequent check-ins and non-medical care for residents in assisted living and skilled nursing facilities.
- *Children*. They should mean not only a distraction for the kids of the house when no adult is around but also as an educational device to help with their development. An example of this is Emotix's *Miko,* which can see, hear, sense and remember children's moods and it uses conversational learning to help in tasks for educational purposes.
- *In General*. Apart from their main companionship role, they are also a domestic hub that can interact with all the IoT devices on the home while also giving real time information of the outside world. An example of this is Blue Frog Robotic's *Buddy* which is shown on figure 3. This companion is totally autonomous and apart from all mentioned above, it can also be used as surveillance, as it has movement detection.

Figure 3. Blue Frog Robotic's *Buddy* is an example of Companion Social Robot.

2.2 Empathy

Currently, there are several different definitions of empathy, but for this work we are going to use first the definition of Hoffman [7], that defines it as "*a psychological process that makes a person have "feelings that are more congruent with another's situation than with his own situation.*".

Empathy has different levels of complexity from low to high and in the robotics field all of them need to be considered at the same time, so A. Paiva illustrated the elements involved in an empathic encounter [8] which are the observer, the target, an event that happens and it is witnessed by the observer, the emotion that the target feels to that event, the context of the event and mediating factors such as how close the observer is to the target. A diagram of interactions is shown on figure 4.

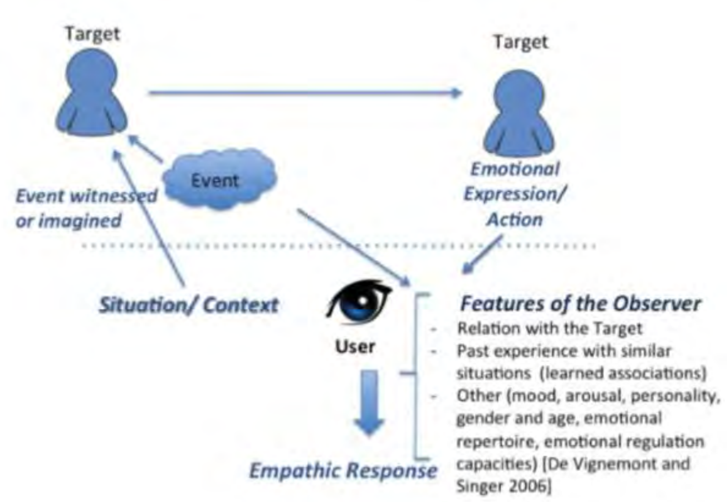


Figure 4. Elements involved in an emphatic encounter

Apart from that, there are three different processes that happen on empathic situations and were defined by Boukricha [9]:

- *"The empathy mechanism as the process by which an empathic emotion arises. "*
- *"The empathy modulation as the process by which both an empathic emotion is modulated, and a degree of empathy is determined. "*
- *"The empathic responses as the process by which an empathic emotion is expressed/communicated and actions are taken. "*

2.3 Empathy in Robotics

There has been few written about robot empathy but there are a couple of works that could be relevant to this study and will be reviewed in this section.

The first one is about the human interaction with artificial intelligence in doing collaborative drawing pieces [4]. A web application called DuetDraw was implemented in order to test this kind of collaboration, which figure 5 illustrates a diagram of the different steps. Its findings can be summarized in these points:

- Users were significantly more content with the AI when it gave detailed instructions.
- While users always wanted to lead the task, they also wanted the AI to explain its intentions but only when the users wanted it to do so.
- Although users rated the AI relatively low in predictability, controllability, and comprehensibility, they enjoyed their interactions with it during the task.

That study is similar to this one as it measures the collaborative aspect of human and AI, with the difference that this work is centered on the application in the robotics field.

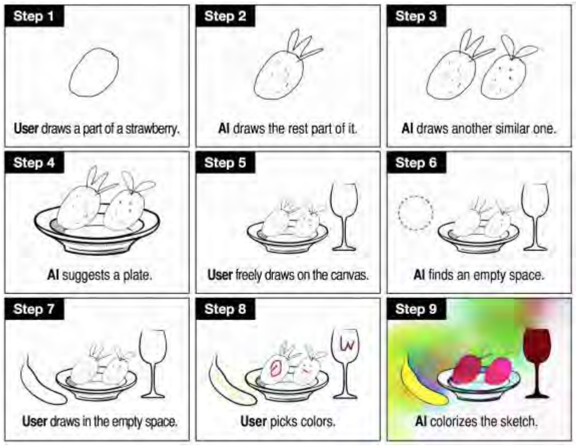


Figure 5. Sample diagram of *DuetDraw* in Lead mode

Another work related that is relevant to this study is one in which a system model for teleoperation called *collaborative control* [5] has been designed. In it, a human and a robot have a collaborative relation that is relegated to the situations in which the Robot needs any kind of advice, as can be seen on figure 6.

Results pointed out that *collaborative control* enabled the robot to perform better than if it were left on its own. Specifically, it was found that "*in situations where the robot does not know what to do, or in which it is working poorly, a simple human answer (a single bit of information) is often all that is required to get the robot out of trouble*".

Figure 6. Control's collaborative exploration mode test

Finally, one last study that is related to this work is one about the investigation of the influences in attitudes and emotions on human–robot interactions centered on the negative ones, anxiety and communication avoidance behavior [6].

A simplified version of the designed model of this negative attitudes and anxiety towards robots can be seen in figure 7.

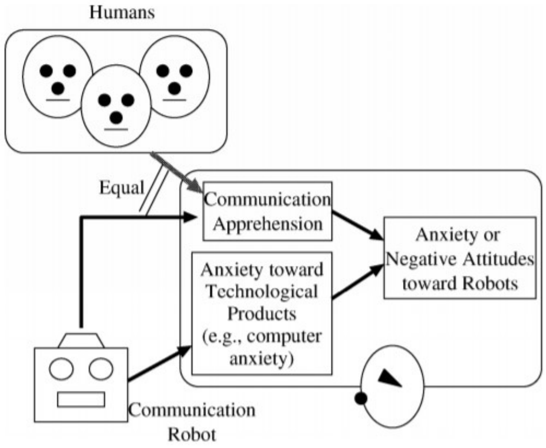


Figure 7. Hypotetical model of negative attitudes and anxiety towards robots

The most relevant finding for our work in this study is that "*repeated interaction with robots can change human behavior toward robots such as communication avoidance behavior*".

3 A framework for the study of empathy

As said in previous chapters, the main goal is to provide a platform that will be used in the study of how empathy is developed between a robot and a human. This chapter will describe in detail all about it.

First, to give an overall picture, the experiment consists of three essential elements: a human, a robot and a computer. The human will have to try to solve a maze that is partially shown on the screen with the help of a social robot created by the company MINT, called Robobo. Nor the human nor the robot can see more than the adjacent blocks that have been discovered.

This robot is comprised by a robot base, a Smartphone and it is controlled by a behavior model that was developed and integrated in the testing platform.

Robobo's role is to be part on the solution of the maze, as a companion for the human. It will be showing its decisions about which path to take in the computer screen but also by showing emotions that could led to the humanization of the robot. This is done with several cues which are movement, lightning, making noises or showing face emotions.

Finally, apart from controlling the robot and displaying the maze, the computer's role is also to manage the whole logic of the experimental platform: from robot behavior to recording logs and getting information of the human emotional state. In figure 8 is shown how the test environment is meant to be used, to the right is the computer which is displaying the maze and to the right the robot can be seen smiling.

In order to give different kind of experimental testing scenarios that could provide more data to an empathy study, there were designed two different strategies of the maze challenge that have different levels of interaction with the robot. They are going to be explained in detail in the following section.

Figure 8. Photo of an experiment setup

3.1 Game strategies

3.1.1 Taking turns

The first mode is the simplest: a turn-based strategy. In this case, human and robot will decide on turns which path to go at every stage of the challenge. These decisions are completely independent, which means that the robot could choose to go to the right path in the first turn and the human choose to return to the starting block at the next turn.

This mode could be used for the following points:

- Measure wherever if the human is more comfortable without having to directly negotiate with the robot.
- Measure wherever if it is more efficient not having to negotiate.
- Determine if there is any kind of bond between robot and human without having a direct interaction, as their actions do not have to be approved by the other party.

3.1.2 Cooperative

In the second mode, first both will express on each turn where they want to go. If they choose the same direction, then that path is taken. If each one wants to go to different directions, then a negotiation phase is triggered and human and robot will have to get to an agreement on where to go.

The negotiation could have one or more steps. In every one of them, human and robot will have to choose between a set of different arguments that can be used to persuade the other part into yielding the decision.

Although having the human freedom to negotiate would be ideal, the processing constraints and difficulty in developing a machine learning AI that could learn negotiation patterns led us to narrow the negotiation arguments into three different ones.

Human and robot can rely on the following arguments to communicate with each other in the negotiation phase:

- *Leadership. "I've decided the most, let me lead"*: Frequency of one's actions until now is higher than the other, so it is natural that its decisions are the ones that have to be taken.
- *Fairness. "You chose the most, now let me!"*: Most actions have been taken by the other part, so if this is cooperation, the decisions number should be even.
- *Failure Rate. "Your decisions took us to dead ends"*: Other one's previous decisions turned out to be dead ends so they should not be regarded anymore.

Each one will have available only the arguments that apply to its current situation, for example, if no dead end has been yet discovered, this option will not appear. Apart from that, on each round both will also have the option to directly yield to the other's decision or to toss a coin. This coin will directly decide by luck which one of the decisions is taken without having to enter in any argument.

The robot will also consider the human's feeling at the negotiation time. If the human is angry then the robot will yield its decision in order to avoid confrontation. This could potentially deepen the connection with the human by showing an understanding of the situation.

For each negotiation step, apart from its decisions being shown on the screen (see figure 9), the robot will be doing a set of actions which involves movement, lightning, sounds and expressions in order to state its argument. If a step ends without agreement, on the next one the robot actions will be done with more emphasis. These behaviors are needed for the bond between robot and human to deepen by humanizing the robot in the taking of decisions.

To illustrate this, we can use the following situation as an example. Both are in the middle of the maze, the human´s decisions have been chosen the most but some of them ended in dead ends. Then, the next turn takes place, here is what could happen:

- [*Robot rotates its wheels to face left*]
- Robot: I want to go left!
- Human: I want to go right.
- Robot: Ok. Then let's negotiate. I think that you chose the most, now let me!
- Human: I've decided the most, let me lead
- [*Robot rotates its wheels to face left, goes forward and back a couple of times*]
- Robot: Your decisions have taken us to dead ends.
- Human: Ok, let's do it your way / Toss a coin then.

This example of a negotiation phase had two different steps and ends in letting the human choose between tossing a coin or yielding its decision to the robot, as it had more arguments in the negotiation. In Figure 9 can be seen a sample of the negotiation phase screen.

The following measures be extracted from the testing of this strategy:

- Number of times human/robot yields or agree
- Emotion of the human through each negotiation phase
- Time spent on each negotiation
- Times a coin toss is forced

The data above could help a study to determine if the robot actions contribute to strengthen the bond between human and robot or see in which situations the human is more willing to yield or quantify what is the impact of the robot yielding a decision on the interactions between human and robot.

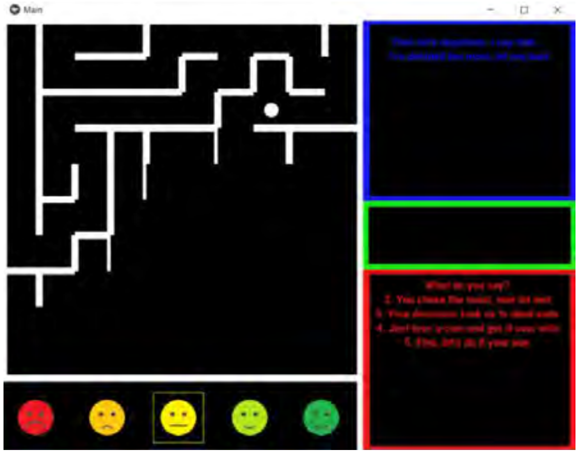


Figure 9. On the right side of the GUI, the negotiation screen: above the robot's decisions, below the human's

3.2 Robobo, the Robot

Robobo, as seen in Figure 10, is a social robot that is oriented to the educational environment. It is meant to be a combination between a simple mobile base and a Smartphone in top, where it should not only get some feedback from the human but also use it as an output device for sounds or face expressions.

Figure 10. Robobo

The target of this robot is being a learning tool that uses some of the latest technologies while being affordable and having an ample library resource for different languages like Python, JavaScript and Scratch and a platform like ROS. This was beneficial to this work as the experimental platform is coded in Python, which made the integration with the robot pretty easy.

The behavior of the robot is programmed on a computer which should communicate with the Smartphone docked on the base. In the figure 11 below, a diagram of the different components of the robot and how they connect with each other is shown.

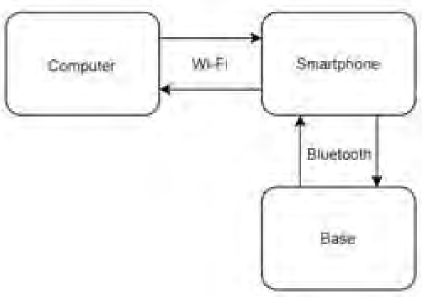


Figure 11. Robobo components connectivity diagram

In the following subsections we will talk more in depth about the role for each one of the components that make this Robot whole.

3.2.1 Robot base

It's the part of the robot that oversees different kind of movements, led lightning and has sensors to avoid collisions. In figure 12 a schematic of the base is shown. In this section we will be describing a bit each of the components.

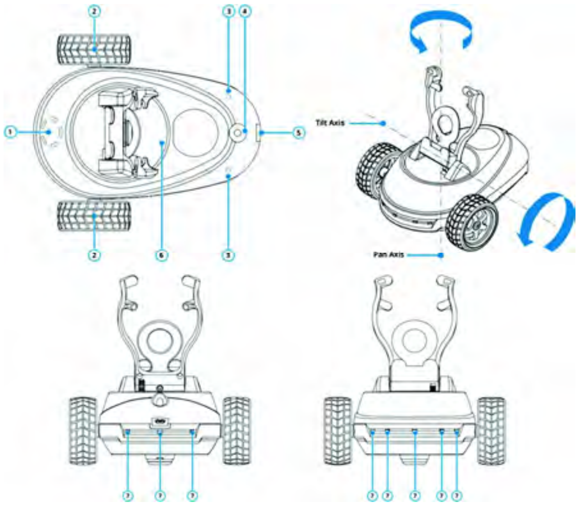


Figure 12. Robobo base schematic

To do the basic movement of the base, it counts with two wheels that can be managed independently. Each wheel can have its own direction, speed and time that are moving. They are indicated with a 2 on the upper left corner of figure 12.

In order to give more expression to the robot, the dock where the Smartphone should be placed can do Pan and Tilt movements. The screen panning can be done from -160 to 160 degrees while the tilting accepts ranges from 5 to 105 degrees as seen on the upper right corner of figure 12.

The last of the base outputs are the LEDs, which are 5 on the front forming a circle and two on the back, which are showed on the upper left side of figure 12 and marked with a 1 and a 3 respectively. These LEDs can be switched on and off independently and its color can be chosen from eight different presets.

As inputs, it has proximity IR sensors at the front and at the back of the base, which are indicated on both lower images of figure 12.

And finally, the USB cable input, which is used for charging the robot base is in the back and showed in the lower left corner of figure 12.

3.2.2 Smartphone

This part has three different functions: to gather inputs, express outputs and serve as a medium of communication between the brain and the robotic base.

The inputs that the phone can gather and send to the computer interface are the following:

- Acceleration of the robot using the gyroscope.
- Screen tap counter.
- Face sensor recognition that uses the Smartphone camera.
- Blob and Brightness detection using the same as above.
- QR Code reading.
- Voice detection using the microphone.
- Battery level of the phone and the base.

The outputs that the robot can express via Smartphone are:

- Speech by using text to speak.
- It can play musical notes and predefined sounds such as mumbles or laughs.
- Facial emotions. There are nine different expressions, which are Happy, Laughing, Surprised, Sad, Angry, Normal, Sleeping, Tired and Afraid. All of them are showed in figure 13.

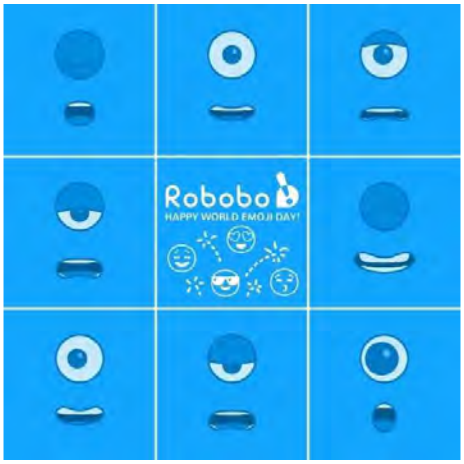


Figure 13. Robobo facial emotions

For the connection to be made between the Smartphone and the base to be done, It is required that the Robobo App is installed and running. Currently, this app is only available for Android and can be found on the Google Play Store.

In this app, the main functionality is the one that lets the user connect to a Robobo Base from the list of associated devices, as can be seen on figure 14. In this screen, the app shows the local IP address, which should be used on the setup of the testing platform. Once it is connected, if the user press Start, the configuration screen is replaced with the Robobo face and now the device is ready to be receiving commands.

IP DEL DISPOSITIVO:
192.168.1.137

SELECCIONA TU BASE ROBOBO:
ROB-GIE
ROB_30
ROB_P09

☐ Master ROS Local

Master ROS Remoto:
http://localhost:11311/

☐ Multirobot

Nombre Robobo:
Robobo2

Figure 14. Robobo Android App main and configuration screen.

Apart from that several configurations of the robot can be set up, such as:

- Language
- Camera Calibration
- Remote Host & Robot Name
- Firmware Update
- Configuration Values Restore

Finally, the last option is the one for folding the mobile phone stand. This needs to be used after the end of the testing in order to avoid possible damage of folding the stand manually.

3.2.3 Robobo Computer Interface

From all the different options about the usage of an interface, that could be on Scratch, Python or ROS. The option used as an interface between the experimental platform and the robot is the one done in Python. This interface consists of a set of python libraries released by the Robobo Project that includes a set of functionalities to interact with the robot. The main actions are:

- Connection with the Base & Initialization (*connect, disconect*)
- Wheel Movements (*moveWheelsByTime*, *moveWheelsByDegrees*, …)
- Mobile Phone Movement (*movePanTo*, *moveTiltTo*, …)
- LED Management (*setLedColorTo*)
- Sound Playing (*playNote, playSound, sayText*, …)
- Showing an Emotion (*setEmotionTo*)
- Tracking Elements (*setActiveBlobs, readClapCounter, readLastNote, …*)

In order to use this kind of functions, a set of different python utils were also provided:

- *Acceleration*: X,Y,Z axis of the base
- *Blob & BlobColor*: Color recognition for use in image processing
- *Color*: LED inputs
- *ConnectionState:* Of the PC to the mobile phone
- *Emotions*: The ones showed on the screen by the robot
- *Face*: Position and distance of the user face from the robot
- *IR*: Position arrays, which make easier to use the actions mentioned above
- *LED*: Position arrays, same as the IR, they are defined so it can be used in an intuitive way
- *Message*: Text to talk parser
- *Note*: Musical scales
- *Orientation*: Of the mobile phone in the base
- *QRCode*: Detection of this kind of codes
- *Sounds*: Premade by Robobo
- *StatusFrequency*: Of the messages coming from the Robot
- *Tap*: Of the user on the phone screen
- *Wheels*: Position arrays

For a better integration with the Main Program, a new Robot class Interface was created with several functions that contained bundles of the robot actions. It will be explained with detail in the 3.3 section.

3.3 Software architecture

This part of the experimental platform is comprised of several different components. In Figure 15 can be seen the relation between each of the scripts and the external agents, the user and the robot.

For the sake of getting a better understanding, the following subsections are a way to separate each part from a logic perspective and the last subsection will get a bit more into the code development part.

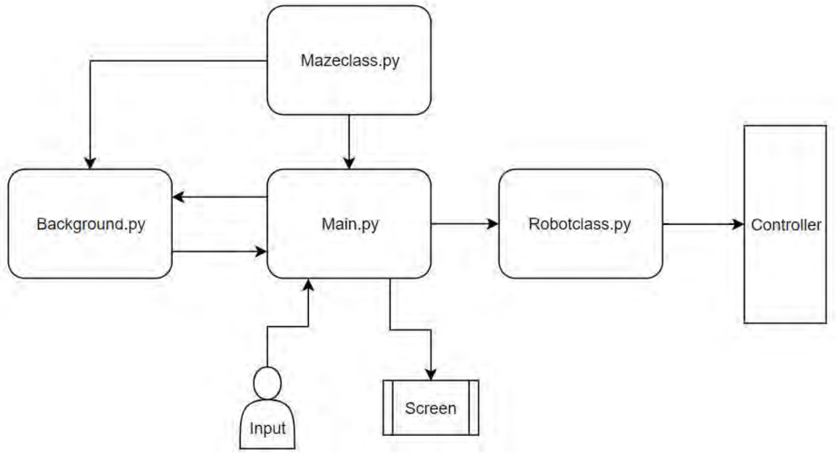


Figure 15. Interaction chart between scripts

First, we will put on a list all the functionalities that are needed in the platform:

- Maze generation and management
- Maze solving algorithm that the robot uses
- Robot position management on the maze map
- User input translation into map array positions
- Negotiation algorithm that manages the arguments
- Movement translation into robot movements
- Framework that manages sets of robot cues
- Log writing of each move done by the robot and the user and other info
- User Interface for the display of the maze, human emotions and robot/system/human negotiation information
- Adaptation depending on the maze mode selected by the user
- Use of different configuration parameters that could change the whole behavior of the test

3.3.1 Maze generation and solving

In order to define and generate the maze, a depth-first search algorithm is used. This algorithm goes block by block of a maze starting from a random point and selecting random neighboring cells which has not yet visited, creating a path. This is done recursively until all cells are visited.

The solution of the maze is saved internally in the maze object and can be used to study the difficulty of the maze generated, but it is not used by the robot as having one actor of the test knowing the correct answer invalidates the point of the test.

In addition to the random generator of the maze, there is also an option to define manually a pre-made maze depending on the mode of the experiment. This is because, in order to have a more significant study sample, all the different participants must go through the same experiment conditions.

For the robot to resolve the maze, the algorithm implemented into the its behavior is a variation of the right-hand rule that is adapted to conflicting situations in which two different parties could make decisions that affect each other negatively.

To explain this further first we will start with the standard right-hand rule algorithm or, as it is also known, wall follower algorithm. It states that if all the walls of the maze are connected, the exit is guaranteed to be found by keeping one hand in contact with one wall of the maze and then traverse it, as it is shown on figure 16. This is not the most optimized solving algorithm, but it is better that way to make the logic of the robot seem more human like.

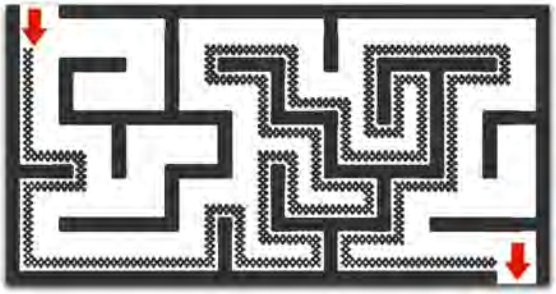


Figure 16. Example of a right-hand rule algorithm

As the maze is comprised by blocks, the algorithm had to be adapted to it. In order to do that, the robot will try to go first to the block below, if it is not possible, then to the right, if not, up, and finally, to the left.

This algorithm should be enough if the robot was solving the maze alone, but when another party gets involved in the decisions and could mess with the following of the algorithm, then it is no longer a valid. For example, if the robot goes left and in the next turn the human chooses to go right, ending in the same spot as the turn before, with this algorithm the robot will want to go left again, starting an infinite loop until the human yields. This behavior is something that could impact negatively on the HRI.

In order to overcome this issue a State was defined for each block in the maze. This state will contain on information that the robot will use to decide its next move in the maze. There are six different types of states:

- *Unknown*: The robot has not been in that block before
- *Current*: Where the robot is
- *Pass*: A block in which the robot has already passed and could be part of the solution
- *Discard*: A block that only belonged to a Dead-End situation
- *Solution*: Part of the solution, determined when the robot gets to the exit
- *Conflict*: A block that was *Passed* but had to be walked again because of direct user imposition.

Having all this information, we will proceed to give an explanation of the solving algorithm:

1. Get the number of walls of the current block
2. Get how many of the neighboring blocks are *Discard* or *Passed*
3. If the sum of the walls and the discard walls is three, that means that we are in the process of backtracking from a dead-end choice and we have to convert all *Conflict* blocks into *Unknown* so we can consider them again
4. Else if the sum of walls, *Discard* and *Passed* blocks are three, then it means that we are not backtracking but there is only one possible choice
5. *Now that we know what we are doing, it is time to determine where to go*
6. If there is not a wall below and it is not the last block of the column, then check if the block below complies with ALL these conditions to select it:
    a. It is not *Discard*
    b. It is not *Pass* or we are backtracking
    c. It is not *Conflict* or we are backtracking
7. Else if there is not a wall on the right and it is not the last block of the row, then check the conditions mentioned above but for the right block
8. Else if there is not a wall above and it is not the first block of the column, then check the conditions mentioned above but for the above block
9. Else select the left block as an answer

With this algorithm, in the example mentioned before, then the robot will follow the path taken by the human but if they encounter a dead end, the path that was taken before and not completed should still be available for the decision making.

One important thing that has to be mentioned is that the algorithm is not bullet-proof and if the human really wants to mess with the robot's decision, it could cause a crash in its reasoning. For example, if the human makes constant backtracking decisions without any kind of logic.

3.3.2 Robot interface

While interacting with the robot, at a code level an interface that bundles different commands for *cues* is needed in order to simplify it. For example, the robot can be beckoned to do a "Dead End" behavior, in which there is no escape but to go backwards, so the interface sends the order of changing the face to "*Afraid*", play a "*Doubtful*" sound, set the LEDs to an orange color and to tilt the mobile phone.

The following are the behaviors that the robot is prepared to have, from this interface:

- *Presentation*: Robobo's self-introduction to the human by saying its name, doing a little bow and showing a happy expression
- *Start*: At the beginning of the trial it does a whole spin while it lets out a laugh and lights its LEDs to green.
- *Win*: At the end of the trial, when reaching the goal, it moves the Smartphone with a happy expression, making sounds and lighting its LEDs to blue.
- *Auto*: When the maze has only one way to go and the decision is automatic, it just moves its head from each side to the other like searching for something.
- *Dead End*: Triggered by having no other way out than to backtrack, the robot lets out a scared sound while bowing its head.
- *Backtrack*: While having to pass through passages that have been visited before, it mumbles and has a tired expression.
- *Waiting*: For the user decision, it just tilts its head to the human and making a noise.
- *Deciding*: When the robot is making the choice, it says "hmm" while moving the face from each side to the other.
- *Negotiating*: An action to convince the human of its decision. Depending of which round it is the negotiation, the action will be with a corresponding intensity

In table 1 below a summary of which robot cues each behavior uses is shown.

Table 1. Cues used by the robot on each Behavior.

Cues > v Behavior	Facial Emotion	Sounds	Movement	LEDs	Tilt/Pan
Presentation	X	X		X	X
Start	X	X	X	X	
Win	X	X		X	
Auto		X		X	X
Dead End	X	X		X	X
Backtrack	X	X		X	
Waiting		X			X
Deciding		X	X		X
Negotiating	X	X	X	X	X

3.3.3 Main and User Interface

Finally, as a Main part, as its name implies, it will have to be the one that glue all the core scripts together and it is the one which is executed. It oversees the behavior of the different test modes and the User Interface (UI).

The UI was done by using the Kivy framework, who made possible an easy implementation of simple graphics, which were chosen that way in order to bring more of the subject attentions to the robot instead of what was happening on the screen.

Figure 17. Screenshot of the User Interface while doing a turn-based test.

In the main panel, the user and the robot (ie. players) are represented with a white circle. As shown on figure 17, players will only be able to see the walls of the blocks where it has passed before and their adjacent ones in all eight different directions so they could give a better situation to the human.

In the panel directly below, the user can state at any time what its current emotion is in five different levels, from Angry to Very Happy. This selection has to be done by clicking on the corresponding face with the mouse. Apart from logging this input for further study, it also is used in the cooperative test mode, if the user is angry, the robot will directly yield to avoid conflict.

The right panel is divided into three different sections:

- Robot UI, which uses blue color and is on the upper side.
- System UI, which uses green color and is on the center.
- Human UI, which uses red color and is on the bottom.

The reason behind it is that by using differentiated sections and colors, the user can understand the situation better without having to use long phrases printed on the screen. The contents of each section will vary depending on the experiment mode used.

First, at the start of the test the Robot UI will show a menu where the user can choose which mode they are going to play or to see the Robot making a short presentation of itself.

In turn mode, on the Robot UI will be shown a "*Deciding…*" message each time while the robot is doing the action of choosing a path. When it is the turn of the human, a message showing "*Your Turn!*" will be visible.

In cooperation mode, it is used to display the current negotiation round options: The ones that the robot has taken will appear in the Robot UI and the ones that are available by the user will appear on the Human UI. There is a mechanic for the coin toss in which a message will appear on the System UI saying, "Tossing Coin" and a "*Your turn/My turn*" message will be intermittently appear and disappear between the Robot and Human UI. When the toss of the coin is finished, the message assigning the turn will be the one remaining on its corresponding panel.

In both modes, when the maze ends, a congratulatory message will be shown on the System UI and the user should be able to retry the test.

As for the user inputs, the human has to enter its direction choice via keyboard arrows when it is its turn to choose the path and whenever is a negotiation phase, the user will have to do a selection of different options that will be appearing on the human UI which will have assigned the number keys in the keyboard.

3.3.4 Negotiation logic

In order to implement the negotiation phase on the experimental platform, the following logical steps have been developed:

1. Two lists should contain the three different argumentations for each human and robot
2. Those lists should also contain the coin toss and yield options.
3. There should be a counter that quantifies the negotiation phases won by each one and a counter for each dead end that was discovered because of their decisions.
4. All the elements of the lists should be available at the start of the negotiation phase.
5. In each negotiation step, each option will be checked to see if they comply with the current state of the test and discarded if they don't. For example, if the robot has taken the most decisions, then the second argument (*Fairness*) will be discarded in their list while in the human's the first (*Leadership*) will be the one eliminated.
6. There should be a way to get the first argument available for the robot, which will be shown on the robot UI screen.
7. On the other hand, for the human, all the list of usable options should be able to be gathered in order to show it on the human UI screen.
8. After both parties have chosen their arguments, they are discarded so they can't be used in the next negotiation step.
9. Step 6 to 8 should be repeated until one of them chooses to toss the coin or to yield. Currently, with only three arguments, there could be at most two negotiation steps.
10. Negotiation phase win counter of the one that is chosen is updated.
11. Negotiation phase ends. Whenever is a next one, it should start again from step 4.

3.3.5 Configurable options

The experimental platform should be able to support the use of configurable options in order to make the test conditions as broader as possible. These are the options:

- *Maze Size*. The screen should be able to auto adjust on how many blocks the tester wants to define. This number could be increased or decreased depending of what kind of challenge the tester wants to give to the subject.
- *Start Select*. The one running the platform should be able to choose who starts on the test. This could give completely different outcomes as it changes who will have the initiative in both the turn mode and the negotiation.
- *Pre-negotiation turns*. Number of turns that the robot yields if there is no agreement in the negotiation mode.
- *Coin Percentage*. Probability that the robot will take the coin choice over the yield in negotiation mode if it runs out of arguments.
- *Auto Mode*. This is a mode in which if there is only one way, the robot automatically takes it and no user action is needed. This could be used in bigger mazes in order to not make it repetitive to the human for making a lot of superfluous inputs and center them on the real decision times.

3.3.6 Metrics

For the study of the test, the following information should be tracked through the whole challenge:

- Test mode selected
- Detailed information of each move of the robot and the human, which should contain:
  - Turn number
  - Which turn is it
  - Starting point and path selection
- If there is a change in the human feelings, original emotion and final one
- If there is a negotiation phase, show who won, why and how much time that it took
- Duration of the whole test
- Average duration of negotiation phase
- Average emotion through the whole test
- Total decision count of negotiations that each one won
- Total number of negotiation phases and steps

3.3.7 Robot phrase dictionary

To give a more diverse experience to the human, a robot phrase dictionary should be created so it can be used in the moments of the challenge that require more interaction, like the negotiation phase.

These phrases will be defined in a python file called robotext and, not only they can be edited, but the program should be able to manage automatically the use of new phrases added or deleted.

The dictionary has phrases for the following situations:

- When the robot thinks there is only one way to go.
- On the negotiation screen, the argument's text.
- When the robot is waiting for and input of the user.
- Agreement on the direction to take before negotiation phase.
- Decision from the robot in the turn mode

3.3.8 Development and debugging mode

In this subsection we describe different additions to the code in order to develop new functionalities or debug unwanted situations.

The maze class has a debug mode in which the whole maze, goal, and solution are drawn on screen by using the *tkinter* library. This was used at the development stage in order to check the reliability and scalability of the maze without having to implement the whole frontend logic.

The robot interface includes a debug mode in which the behavior of the robot is simulated on the terminal screen, useful for development as it takes of the 3rd party element that is the robot, cutting in waiting time on debugging.

Apart from that, there is also a developer mode, where the user can access a terminal menu to give direct commands to the robot. These commands range from direct movements of the wheels or other cues to the execution of the different robot behaviors. This mode is hidden to the test subjects and, as its name implies, was only used for development purposes.

3.4 Face Emotion Recognition

As stated in the first point of this chapter, one of the inputs that the robot takes into account for the negotiation phases is the emotion of the person who is working with. The first approach for this was facial recognition.

Because of the movements that the robot will be doing through the maze solving and its limited processing capability, it had to be done via an external source. In this case we choose to do the video gathering and processing from the computer where the test is done.

Creating facial recognition software from scratch could be considered a full study case, so what was done is the adoption of already existing model. The one that was more promising was a Python script made by Amine Horseman that will be explained below. [10]

3.4.1 Script Technical Explanation

This python scripts were made in order to demonstrate that the CNN architecture performs better than SVM on facial recognition and if the use of landmarks, HOG and sliding window improves its accuracy, as seen in Figure 18.

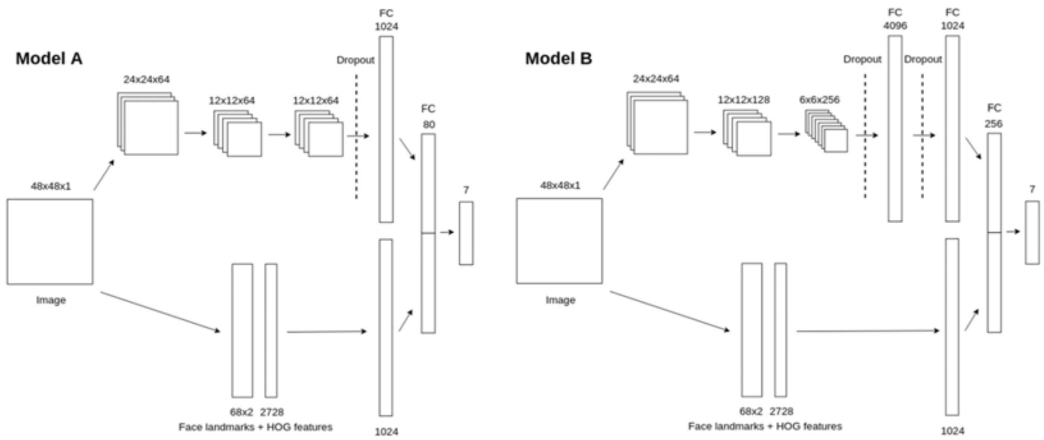


Figure 18. Diagram of the two different CNN models used.

For the training of the model, a dataset called Fer2013 is used. This dataset contains 30000 images of facial expressions that are classified in seven different categories, one for each emotion: Neutral, Happy, Sad, Fear, Surprised, Angry and Disgust.

The biggest issue that this dataset brought was that a good percentage of the images were misleading, in the shape of cropped pictures, wrong landmarks and non-human faces. The best results that were achieved with this dataset were 75.2% as stated in [11].

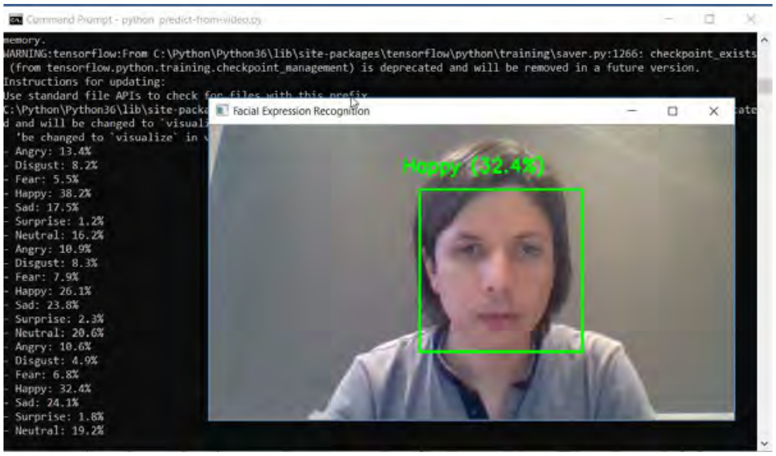
The author of the script that we are using claims that its accuracy is 75.2% using only 5 expressions, and 61.4% when the whole set is involved.

The following are settings that are used for the training and can be modified con the parameters.py file or by command:

- Use of face landmarks that delimit the expression area to be analyzed, improving the accuracy.
- Use of Histogram of Oriented Gradients (HOG) that should improve object detection by using technique counts occurrences of gradient orientation in localized portions of an image.
- Use of Sliding Window algorithms in order to evade using the full picture by detecting the face earlier.

It is possible to change all the parameters in that file, but it is not recommendable unless the user has a deep knowledge on image processing.

After the training, the script is ready to do recognition on both still images and video feed, an example of the script in action is shown on Figure 19 below.



```
memory.
WARNING:tensorflow:From C:\Python\Python36\lib\site-packages\tensorflow\python\training\saver.py:1266: checkpoint_exists
 (from tensorflow.python.training.checkpoint_management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
C:\Python\Python36\lib\site-packa
d and will be changed to `visual
  'be changed to `visualize` in 
- Angry: 13.4%
- Disgust: 8.2%
- Fear: 5.5%
- Happy: 38.2%
- Sad: 17.5%
- Surprise: 1.2%
- Neutral: 16.2%
- Angry: 10.9%
- Disgust: 8.3%
- Fear: 7.9%
- Happy: 26.1%
- Sad: 23.8%
- Surprise: 2.3%
- Neutral: 20.6%
- Angry: 10.6%
- Disgust: 4.9%
- Fear: 6.8%
- Happy: 32.4%
- Sad: 24.1%
- Surprise: 1.8%
- Neutral: 19.2%
```

Figure 19. Screenshot of the recognition software on action.

In addition to above, there is a training mode that uses hyperparameters. This training consists in several different evaluations with a modification on each evaluation of the following variables:

- Learning rate
- Learning rate decay
- Decay step
- Optimizers (momentum, adam, rmsprop, adagrad, adadelta)
- Optimizer parameters

This kind of training should add reliability to the final outcome.

3.4.2 Testing the facial recognition model

The main goal for this test was to make sure that this script could be relied for the emotion recognition. In order to have a conclusive testing, instead of a live video feed, the script was modified to analyze the same recording in all the different test cases and to show not only the recognized emotion on the screen but also the percentage of it that the program considered it was above from all the other possible emotions.

These test cases were different levels of training optimization by hyperparameters, starting with no extra evaluations (first column) and ending in twenty (last column), in increments of five for each one. Depending on the number of evaluations, the training could take up to several hours.

The video record used for the testing was done with as much visibility of the subject as possible and good lighting. It consisted of three rounds of the different emotions that were supposedly supported by the recognition software.

The order was: Neutral →Happy → Sad → Fear → Surprised →Angry →Disgust

Notation used in Table 2 on next page:

- X → Emotion not recognized
- ~ → Emotion partially recognized but not maintained during the duration of it
- O →Emotion recognized correctly
- Score was calculated with the following formula:
    - Number of **O** + Number of **~** * 0.5
    - Max Score is 21, the total number of emotions tested

Through the execution of the test, for some reason, the hyperthread testing using 10 evaluations failed in all recognitions. It was ran three different times and seemed like a code limitation.

The analysis and conclusion of the test data have been stated on the section *3.5.3. Analysis and Integration on main system*.

Table 2. Emotion recognition test results.

Evaluations > v Emotion	1	5	10	15	20
Neutral 1	X	X	X	X	X
Happy 1	~	~	X	~	~
Sad 1	~	~	X	O	O
Fear 1	X	X	X	X	X
Surprised 1	X	X	X	X	X
Angry 1	X	~	X	~	O
Disgust 1	X	X	X	X	X
Neutral 2	X	X	X	X	X
Happy 2	~	~	X	~	~
Sad 2	~	~	X	O	O
Fear 2	X	X	X	X	X
Surprised 2	X	X	X	X	X
Angry 2	X	~	X	~	O
Disgust 2	X	X	X	X	X
Neutral 3	X	X	X	X	X
Happy 3	~	~	X	~	~
Sad 3	~	~	X	O	O
Fear 3	X	X	X	X	X
Surprised 3	X	X	X	X	X
Angry 3	X	~	X	~	O
Disgust 3	X	X	X	X	X
Score	**3**	**4.5**	**X**	**5.5**	**7.5**

3.4.3 Analysis and Integration on main system

After the testing phase, it became clear that, even if the script was a good design, the image database that it was using contained several data that was making the script not giving acceptable results for its integration in the project.

A better database with a huge quantity comprised of well documented expressions data and landmarks was found: The 10k US Adult Faces Database by Wilma Bainbridge.

This could make a reliable facial recognition training for the script, but in order to make it work several changes should be done to the code, plus the heavy testing and given the time constraints it had to be put aside and made a future line of work.

In conclusion, rather than automatically assessing the user emotion, we opted for manual input by the user through the test process.

4 System Evaluation

In this chapter, we will cover different aspects from the experimental platform that were tested. These tests consist in the run of five different iterations, which means solving a maze from start to finish, of what we will define as a *standard* setup plus the alteration of the aspect that is going to be tested.

These iterations will be run in both turn and cooperation modes so in total there would be ten different runs for each aspect tested.

Before going to the actual testing data, the definition of *standard* setup will be explained. It contains the following:

- Maze size: Seven by seven blocks, which proved to be the most comfortable size by one of the evaluations explained later.
- Maze used: random generated, as using the same one for all the different iterations could alter the outcome on the decision taken by the human.
- Coin toss: Fifty percent. This means that if the robot has no other argument in the negotiation phase, it will decide between tossing a coin and yielding with a 50% chance of each one.
- Robobo usage: No. For some aspects of the test, like solving a maze of 20x20, having the robot doing the cues could stretch the length of the test exponentially, so the measures are done with the only the programmed behavior of the robot being shown in the screen using the debug mode.
- First negotiation yields: Two. This parameter's purpose is to define how many times the robot will yield at first if a negotiation phase must take place. It is used to give a feedback that can be used by the robot for future negotiation phases because at the start no one has done any decision so none of them could have an argument on their favor in the first negotiation.
- Initiative: Robot. This is defined so that the Robot could have a mayor role in the maze solving. When the human has the initiative, there is a chance that the robot could not show any decision at all, for example, if they always agree. In that case if the robot had the initiative, at least the human would see the robot's decision and not just "*I agree*".

The main goal for the evaluation will be to detail the importance of each tested aspect, its data taken from both turn and cooperation modes, compare them and finally do an evaluation of the resulting data.

4.1 Robot interaction

The most important aspect to be evaluated in this work is the actual role of the robot in the experimental platform. The data to be measured in this test will be the following:

- Time that takes to go from the start of the maze to the finish. This metric will be useful to know whether this kind of challenge will be usable for a study. If it takes a long time, it could cause a negative impact on the bond between human and robot because of being repetitive while if it is very short it could not be enough to the formation of it. Also, from a study execution point of view, if it takes too long to get only one subject sample, the cost of the study goes up.
- How much the cooperation between human and robot impacts in the test time. Which will be useful to use in a comparison between that mode and the turn based. If the increment of time between them is low, then it could mean that there is really no effective time for any bonding to be done and could be considered the same as taking turns to decide where to go without considering the other party.
- The average negotiations that take place in the same iteration. If this number is too low, then it means that the maze generated by the system is not suitable for the test of interactions between human and robot.

Table 3. Turn based robot interaction test results.

Measure > v Iteration	Total Turns	Robot Time	Human Time	Total Time	Human %
Iteration 1	31	86 s	31 s	117 s	26.50 %
Iteration 2	40	112 s	50 s	162 s	30.86 %
Iteration 3	33	88 s	29 s	117 s	24.79 %
Iteration 4	24	60 s	14 s	74 s	18.91 %
Iteration 5	39	121 s	46 s	167 s	27.54 %
Average	33	93.4 s	34 s	127.4 s	25.72 %

Table 4. Cooperation strategy robot interaction test results.

Measure > v Iteration	Total Turns	Total Negotiations	Robot Time	Human Time	Total Time *	Human %
Iteration 1	29	4	116 s	72 s	188 s	40.45 %
Iteration 2	38	6	155 s	97 s	252 s	38.49 %
Iteration 3	25	2	81 s	33 s	114 s	28.95 %
Iteration 4	44	7	202 s	134 s	336 s	39.88 %
Iteration 5	42	6	183 s	120 s	303 s	39.60 %
Average	35.6	5	147.4 s	91.2 s	238.6 s	37.47 %

* Note: Coin toss time was added to both of them equally.

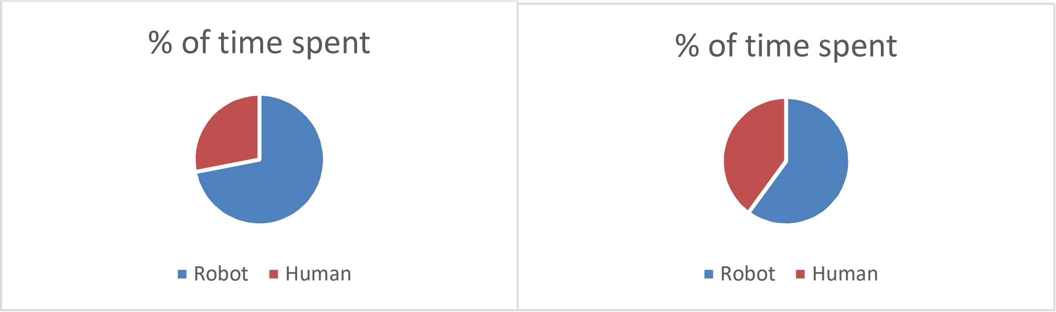


Figure 20. Average % of time spent of each mode, to the left turn based, to the right cooperation.

From table 3 and 4 we can extract a comparison table between turn and cooperation mode. In figure 20 on the left side is shown that most of the time in turn mode is being consumed by the robot. This is because this is a mode in which the decisions are being taken fast, but the human must wait until the robot finishes its decision cues before being able to express its next movement. It does not help that the input of the user is just pressing the arrow keys, so the numbers are not contrastable.

But what it really can be contrasted is that percentage of time spend by the human with the one in the cooperation strategy. In the same figure to the right can be seen that the percentage grows in cooperation, meaning that the HRI goes up when they must negotiate.

In terms of total time, it was expected that the negotiation strategy was going to take more time than the turn based. This is mainly because of two reasons.

First, in the negotiation strategy for each movement both human and robot must provide its decision. This is the minimum interactions case scenario in which they agree and move on to the next block.

Second, every time they disagree, at least one extra input will be required to select the negotiation argument. In the worst-case scenario, both can still disagree and go to a second step in the negotiation. And even there if one of them select to toss the coin then there is extra time added because of the tossing coin animation.

4.2 Maze size

This evaluation was done to determine the optimum size of the maze that should provide a reasonable test duration. As if it was long, it would become tedious to the subject and this could pose a risk to the bond of human and robot.

The sizes that were chosen are 5x5, 7x7, 10x10, 15x15, 20x20.

Table 5. Turn based robot interaction test results.

Measure > v Iteration	Turn Mode
5x5	11
7x7	30
10x10	50
15x15	102
20x20	254

What has been tested is that a 7x7 maze is the most optimum maze as the average between thirty and forty turns does not last enough to impact negatively on the human perspective.

If we take in consideration the turn duration in the past evaluation and extrapolate the results to this turn count, the time cost of the mazes starting by the 10x10 is not realistic for a study if it wants to use a sample big enough to draw conclusions.

Another thing that was detected in the evaluation is that the 20x20 is the maximum that the maze could be generated without having to scale the player circle on the map, as it is shown on figure 21, which can barely fit in a block.

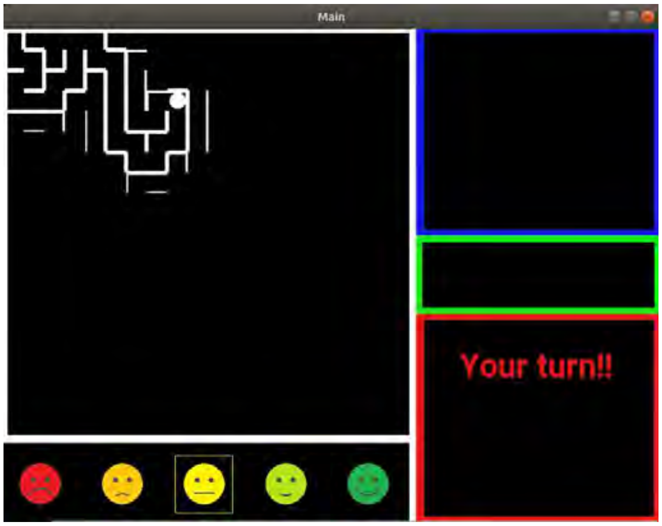


Figure 21. 20 by 20 maze.

5 Conclusions

Empathy between robots and humans is still a world in which there are a lot of open questions and the test runs confirmed that this testing platform could give a new point of view which will contribute to the study of ways of bonding between robot and humans.

Providing a challenge to a human with a robot as a companion is not always enough to build a bond between both of them. It can serve as an example the turn-based strategy, where both parties are solving the maze together but with very few interactions.

Giving them the means of interaction, in this case by negotiating at each step if they were not on agreement, is able to provide us data on how to tackle the HRI and improve it from the robot side. As an example, there was a point in the development which revealed that if the robot gives always the same answer to the human, in terms of cues such as saying, *"This way!"* on each turn, the human could grow tired of the robot behavior and see the solving of the maze as a chore. That was what motivated the use of a dictionary that could select between different phrases for each situation.

The configurable options enable the use of different testing parameters, such as initiative, dialog and robot behavior that could be considered into the study of behavior models by taking test samples modifying these parameters. For example, it could be studied if having a robot that is more prone to yield its decision contributes more to the HRI than having a behavior in which it won't never give up. This is possible thanks to the tossing coin parameter, which dictates the probability in which the robot will yield its decision or not and force the toss of a coin.

The strong point of this work is that it could set the base for an even greater testing platform, in which if the future work lines that are stated here are added, it could give the necessary data for a complete study.

But on the same line, this could be considered as a weak point as it has a considerable room for improvement in terms of negotiation model and robot interaction. Also, the lack of actual experimentation with other subjects was detrimental, as their feedback would have been part of an improvement for the platform.

6 Future work lines

About future work lines, the first that comes to mind is the adoption of the facial recognition by using a better image database to train the algorithm. There was already one with the landmarks ready to be used but because of time constrains it couldn’t be fitted in: The 10k US Adult Faces Database by Wilma Bainbridge. This could allow a better input of the user’s feelings, as it will be in real-time.

Other future line is the improvement of the negotiation algorithm by using formulas that consider several different environment inputs and use them as weighted variables, such as current test time, amount of decisions, disagreements and different thresholds of action. For example, the robot could make the decision to yield in a negotiation if the challenge have been on for a long time or to never give up depending on how much closer it gets to the exit.

Adding to the one mentioned before, it would also be interesting to have the robot showing emotions that were adapting to the choices made on previous negotiation rounds, such as frustration for not being able to follow the desired path for a successive number of times.

Another step in the negotiation line would be the voice recognition of the human in the whole process and not having to input manually. It could also add a flexibility component by giving free new options to the human and the use of a machine learning method for the robot to learn new ways of negotiation.

Finally, one other future work could be the increase on the difficulty of the maze, giving more open spaces where the robot and the human get a greater sense of being lost. For example, not having a predefined exit or having a different shaped maze, like circular or non-geometrical.

7 References

[1] Towards Sociable Robots. Cynthia Breazeal. MIT Media Lab.

[2] A context-dependent attention system for a Social Robot. Cynthia Breazeal and Brian Scassellati. MIT Artificial Intelligence Lab.

[3] Robot mirroring: A framework for self-tracking feedback through empathy with an artificial agent representing the self. Monica Perusquía-Hernandez, David Antonio Gómez Jáuregui, Marisabel Cuberos-Balda, Diego Paez-Granados. NTT Communication Science Lab, Univ. Bordeaux, ESTIA, Swiss Federal School of Technology in Lausanne.

[4] Understanding the User Experience of Co-Creation with Artificial Intelligence. Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, Bongwon Suh. Seoul National University, 2018.

[5] Collaboration, Dialogue, and Human-Robot Interaction. Terrence Fong, Charles Thorpe, Charles Baur. The Robotics Institute, Carnegie Mellon University, Pittsburgh.

[6] Prediction of Human Behavior in Human-Robot Interaction Using Psychological Scales for Anxiety and Negative Attitudes Towards Robots. Tatsuya Nomura, Takayuki Kanda, Tomohiro Suzuki, Kensuke Kato.

[7] Martin L. Hoffman. 2001. Empathy and Moral Development: Implications for Caring and Justice. Cambridge University Press.

[8] Empathy in Virtual Agents and Robots: A Survey. ANA PAIVA, INESC-ID and Instituto Superior Técnico, University of Lisbon, Portugal IOLANDA LEITE, KTH Royal Institute Technology HANA BOUKRICHA and IPKE WACHSMUTH, Faculty of Technology, Bielefeld University

[9] Hana Boukricha. 2013. Simulating Empathy in Virtual Humans. Ph.D. Dissertation. Bielefeld University.

[10] https://github.com/amineHorseman/facial-expression-recognition-using-cnn

[11] Facial Expression Recognition using Convolutional Neural Networks: State of the Art. Pramerdorfer & al. 2016 (https://arxiv.org/abs/1612.02903)

8 Appendix A. Library Requirements

The following is a list of all the python libraries needed to run the scripts on this TFM. As the facial emotion recognition is not integrated on the main program, it can be run without the libraries required for it.

8.1 Main Program

- Kivy
- Pygame
- Cython
- Opengl
- Pypiwin32
- Libgl1-mesa-dev
- Libgles2-mesa-dev
- Zlib1g-dev

8.2 Facial Emotion Recognition

- Tensorflow
- Tflearn
- Numpy
- Argparse
- Hyperopt
- Pymongo
- Networkx
- Dlib
- Imutils
- Opencv3
- Scipy
- Pandas
- Skimage

9 Appendix B. System Requirements

Requirements needed to run the system:

- The computer and the phone have to be connected to the same Wi-Fi local network
- This network doesn’t need internet connection
- Due to connectivity/protocol issues, a Unix SO is required. Ubuntu is recommended
- Python3 installed with pip to install all the library dependencies
- Robobo application has to be installed on the phone, because of this, it has to be an Android phone
- The Robobo base is autonomous, so it has to be charged beforehand

10 Appendix C. Setup and Instructions

In order for this experiment to be used by general public, on this appendix can be found the Setup and steps used in the different parts of the experimental platform.

10.1 Main Program

To run the main program (maze & robot):

```
python main.py
```

The configuration files are in const_def.py and the variables of interest that can be modified by the user are:

- *ROBOT_IP*: is the one that appears on the Robobo App.
- *MAZE_SIZE*: size of the maze in squares.
- *DEBUG_MODE*: if this is set to *True* then the robot will not be used and all its actions will appear on terminal.
- *START_SELECT*: This will select which one is the one who has the first action, can be "*ROBOT*" or "*HUMAN*".
- *TIME_TURN*: To adjust the time that the robot turns, as it can depend on the model.
- *TIME_STRAIGHT*: To adjust the time that the robot goes straight.
- *TILT_SPEED*: To adjust the speed which the robot tilts the mobile phone.
- *PAN_SPEED*: To adjust the speed which the robot pans the mobile phone.

10.2 Facial Emotion Recognition

To run the program, first it needs to be trained and evaluated, using this command:

```
python train.py –train=yes –evaluate=yes
```

This will do a standard training of the model and will return an auto evaluation done by the program that will state the current accuracy of the emotion recognition.

After the training, the script can do recognition on both still images and video feed with the following commands:

(Still Images)
```
python predict.py –image path/to/image.jpg
```

(Live video)
```
python predict-from-video.py
```

So when executing the training, you have to pass the max_eval parameter. In this example, the training does 10 consecutive evaluations:

```
python optimize_hyperparams.py –max_evals=10
```

11 Appendix D. Sample of test log

The following is a sample of the logs.txt that is generated after a cooperation mode test.

```
-------------------------------------------------------------
COOPMODE
1|Agreement: Yes. Move => [1,0]
2|Agreement: Yes. Move => [2,0]
3|Agreement: Yes. Move => [2,1]
4|Agreement: Yes. Move => [3,1]
5|Agreement: Yes. Move => [3,2]
6|Agreement: Yes. Move => [3,3]
7|Agreement: Yes. Move => [2,3]
8|Agreement: No. Move => [2,4]
Negotiation|2.15s|Win: Human|Move => [2,4]
9|Agreement: Yes. Move => [2,5]
10|Agreement: Yes. Move => [2,6]
11|Agreement: Yes. Move => [3,6]
12|Agreement: Yes. Move => [3,7]
13|Agreement: Yes. Move => [4,7]
14|Agreement: Yes. Move => [4,6]
15|Agreement: No. Move => [5,6]
Negotiation|4.22s|Win: Human|Move => [5,6]
Emotion: Neutral => Smile
16|Agreement: Yes. Move => [5,7]
17|Agreement: Yes. Move => [6,7]
18|Agreement: Yes. Move => [6,6]
19|Agreement: Yes. Move => [6,5]
20|Agreement: Yes. Move => [6,4]
21|Agreement: Yes. Move => [6,3]
22|Agreement: Yes. Move => [6,2]
23|Agreement: No. Move => [6,3]
Toss Coin
Negotiation|14.78s|Win: Robot|Move => [7,2]
Emotion: Smile => Neutral
24|Agreement: Yes. Move => [7,3]
25|Agreement: Yes. Move => [7,4]
26|Agreement: Yes. Move => [8,4]
27|Agreement: Yes. Move => [8,5]
28|Agreement: Yes. Move => [9,5]
29|Agreement: Yes. Move => [9,6]
30|Agreement: Yes. Move => [8,6]
```

```
31|Agreement: Yes. Move => [8,7]
32|Agreement: Yes. Move => [8,8]
33|Agreement: No. Move => [8,9]
Toss Coin
Negotiation|14.78s|Win: Robot|Move => [7,8]
Emotion: Neutral => Angry
34|Agreement: Yes. Move => [7,7]
35|Agreement: Yes. Move => [7,6]
36|Agreement: Yes. Move => [7,5]
37|Agreement: Yes. Move => [7,6]
38|Agreement: Yes. Move => [7,7]
39|Agreement: Yes. Move => [7,8]
40|Agreement: Yes. Move => [7,9]
41|Agreement: Yes. Move => [8,9]
42|Agreement: Yes. Move => [9,9]
Finish|Total: 351.33s|AverageEmotion: Sad(2.46)
------------------------------------------------------------
```