Tampere University

Aitor Toichoa Eyam

EMOTION-DRIVEN HUMAN-COBOT INTERACTION BASED ON EEG IN INDUSTRIAL APPLICATIONS

Faculty of Engineering and
Natural Sciences
Master's Thesis
May 2019

ABSTRACT

Aitor Toichoa Eyam: Emotion-driven human-cobot interaction based on EEG in industrial applications
Master's Thesis
Tampere University
Automation Engineering, Factory Automation and Robotics
May 2019

Currently, the world is facing its fourth industrial revolution which has been conceptualized into the term of "Industry 4.0". Among all the features that include this idea, becomes essential to keep humans in the loop of the processes. Taking into account the human factor inside the industry is one of the most difficult aspects to manage optimally.

Different from machines, human beings' behaviour is complicated to parametrize and anticipate. Combined with the fact that as time goes by industries are becoming more complex, it is necessary to develop adaptive systems.

One of the duties of the adaptive systems is to make machines adjust to situations that could be modified quickly or several times without decreasing the performance of the system. As a consequence, this concept is being applied to the robotic field too.

As the industries are evolving, more robots are being introduced in them sharing the workplace with humans. Consequently, the concept of Human-Robot Interaction (HRI) has become a great topic to improve. The level of success of this interaction is closely related to the level of trust in the same one. Human-Robot trust relation can be increased in several ways related to the robot, to the human or to the environment that surrounds the interaction.

In order to increase trust in HRI, among other parameters, it has been developed the collaboratives robots (cobots). Cobots are robots mean to work in a collaborative way with humans, changing the patterns of interaction established with robots.

However even if cobots are able to work hand-to-hand with humans, they sill need to understand in a better way human needs to become their colleagues.

The main aspect that defines human beings is their emotional state. Humans' cognitive state is what characterize mostly all the decisions that they made during a day. Therefore, understanding humans' emotional response is becoming a key target for the industrial field.

In seek of understanding humans emotions, there have been developed several ways to analyse the emotional state of a person by gesture recognition, speech recognition, electroencephalography, etc. Depending on the technique of analysis, there have been developed devices that can detect those emotions.

The aim of this thesis work is to develop a system that acts as a bridge between humans and cobots influencing positively to their interaction making a cobot adapt its behaviour to the emotional state of a human while performing a collaborative task.

By achieving this, major issues related to trust in HRI can be favoured, having impacts both in industry as in social fields.

Keywords: human-robot interaction, human-cobot interaction, cobot, collaborative work, trust, human-robot trust, emotions, emotion-driven, electroencephalography, EEG, industrial applications, human-in-the-loop, human-machine interface, adaptive systems.

PREFACE

First of all, I would like to thank my family because is due to their effort and love I had the great opportunity to do an exchange to Tampere, Finland and having one of the best experiences that I had in my life. I also thank their support through the whole year making me feel close to home.

Secondly, I want to thank Prof. José Luis Martínez Lastra and Dr. Borja Ramis Ferrer for giving me the opportunity and confidence for joining the FAST-Laboratory team, and having the chance of performing a project with such characteristics as this one, being the biggest project that I have done so far.

Thirdly, I want to thank all the members of the FAST-Lab for creating an incredible atmosphere to work seriously and comfortably at the same time that feeling close to each other.

Finally but not least important, I would like to thank all the friends that I have done during my stay in Finland. I will never forget all the moments, anecdotes and experiences that we have lived together. They have been my little family here in Tampere, creating deep connections that will remain forever in my heart.

With all the love, energy, experience and lucidity!

Tampere, 20th May 2019

Aitor Toichoa Eyam

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

API	Application Programming Interface
BCI	Brain-Computer Interaction
ECG	Electrocardiogram
EEG	Electroencephalography
HCI	Human-Computer Interaction
HMI	Human-Machine Interface
HRI	Human-Robot Interaction
HRC	Human-Robot Collaboration
IP	Internet Protocol
JSON	JavaScript Object Notation
LAN	Local Area Networks
LoA	Levels of Automation
PAD	Pleasure-Arousal-Dominance
SDK	Software Development Kit
TCP	Transmission Control Protocol
VR	Virtual Reality

1. INTRODUCTION

This chapter presents the main concepts that any reader will need to understand the context in which this document is developed. By the end of the section, the reader will be aware of the challenges that this Master Thesis is facing and the proposed plan to solve them.

1.1 Thesis background

Nowadays the world is living an exponential growth of technology which is leading the history of civilization to its denominated 4th industrial revolution, also know with the name of "*Industrie 4.0*" [1]. This change in the industry concept is related to diverse aspects such as, Cyber-Physical Production Systems (CPPS) which relates to the connection of the virtual and the physical world, smart factories, intelligent production which will focus on the development and application of Human-Computer Interaction (HCI), customization of manufacturing process and products making machines adapting to humans needs, etc.[2]. These developments answer to the necessity of improving the efficiency of the systems, produce a cleaner industry environment using technologies which reduce the negative impact on the atmosphere and facilitate user's use, comprehension and trust on technology [3].

With the development of smartphones, tablets and other personal devices, the human race is physically and emotionally more linked to technology than ever before. But, unlike technology progression, human being's way of thinking does not work either grow in an exponential way. Due to this fact, scientists, entrepreneurs and engineers are creating different ways to improve the understanding between technology and its users [4]. They have developed algorithms that use the information that the users create while using their devices. Improving their service and offering to the users a better personal experience[5]. Some examples of this progression can be easily found on search engines, advertisements, or social networks [5], [6].

Similarly to the ideas that are being applied to give a better experience to the users of devices such as smartphones, adapting their characteristics to them, it is starting to be introduced and developed in the factories. The main objective behind this is to keep humans in the loop of the processes[7], reaching a scenery in which both humans and machines are an inherent part of the equation. Therefore, terms like HCI or Human-Robot Interaction (HRI) are being thoroughly employed in the industries. These technologies enhance the trust and understanding between humans and machines in order to improve the performance and employees' work-life quality[8].

HCI concept has opened a new road on how to design tools. While working, instead of being the user the one that has to adapt to the machine, is the machine the one that must be adjusted to the user's necessities. Giving life to the concept of adaptative automation systems [8]. Each individual has a completely different pattern of behaviour, skills, needs and limitations related to its own background, that is why nowadays there is a tendency to change from a generalist way of design to a particular one, taking into account user's preferences and background [9].

One field in which it is possible to see how this new idea of making technology adjust to the users is being implemented is the robotic field. Robots are getting each day more in human lives, sharing working area or even assisting them on daily duties [10], [11]. As this tendency is becoming more relevant as technology is advancing, it is really important to satisfy two objectives: produce robots with the capacity to understand and assist better human beings and produce safe robots. So as to achieve the first objective, one of the new approaches that are being developed is to generate robots with the ability to recognize and interpret human emotions giving an optimal response taking them into account [12]. In the second place, to fulfil the second objective, the production companies are developing the new era of robots denominated "Collaborative robots" or just *cobots* [13], [14]. The aim of the cobots is to share the same work area as a human and work in a collaborative way with them without being considered as an unknown agent for the people that surround them [15].

For the sake of making robots assists better humans, diverse techniques are being applied in order to detect human's emotions. So as to recognize mental states, there is interest in applying techniques different from the industrial field, such as the electroencephalography (EEG) [16]. EEG technology is becoming important due to the fact that is a non-invasive method and is based on internal signals of the body which are harder to control voluntarily [17]. To apply this technology outside a medicine atmosphere, various companies are developing headsets that measure brainwaves using EEG technology. These devices are being applied on fields such as videogames [18], [19], robot manipulators [20], drone control [21], etc.

By applying the techniques mentioned in the previous paragraphs on the factories, the approach by which industrial applications are performed will change fitting more into the Industry 4.0 scope. The industry will become more customized and adaptive making easier to work for the employees, achieving better efficiency, results and creating a more healthy environment.

1.2 Problem statement

As introduced in the previous point, nowadays society is heading a new chapter in its evolution in which technology is starting to be more personalized and suitable to each individual, solving labour accidents, communication issues and personal dysfunctionalities.

The influence of the ideas brought by the Industrie 4.0 concept is changing the actual situation of the production and fabrication industries from simple to complex systems [22], [22]. The evolution of the production plants from manual to automated systems has made inevitable the introduction of robots and other machines into the working place. As a result, as more machines are introduced in the workplace it increases the necessity of creating more systems of Human Machine Interface (HMI) to track their performance and state [8]. These HMI act as bridges between human and machines, allowing the exchange of information among both partners. The miscommunication between both actors could lead to an increase in user's distrust, reductions in productivity or even labour incidents [8].

In hence of avoiding problematics regarding the interactions between human and machines. It must be taken into consideration the concept of adaptative automation, producing an industrial environment that adjusts its components to the users that interact with them [23], [24].

Nowadays, even if there have been developed systems that improve the interaction between human and machines, such as robots, there is still work to do in the field. A reflection of this necessity is that even if its know that mental states such as stress, boredom or fatigue are reducing productivity and causing accidents while working with robots [25], HMI systems are not adjusting their characteristics to those mental states [8]. As a consequence HRIs are not achieving their best level of performance, generating a reduction in trust towards robots.

Just as proposed in [8], in order to personalize the interactions between robots and humans, it is necessary to develop adaptative systems that can measure the characteristics of a person and use them to influence the behaviour of the robots with which this person is interacting. So as to fulfil that objective, it is necessary to find a proper answer to the following question:

- How to positively adjust the interaction between a human and a cobot?

1.3 Objectives

In order to give an optimal answer to the previous question, the problem proposed can be approached by defining different objectives. By accomplishing these goals one by one the challenge will be solved. The following list presents the main targets of the thesis work:

- Select an optimal emotion detection method.
  - This method must be accurate enough so the mental states are not confused leading to an error.
- Analysis of user's emotional reactions.
  - By knowing how the user reacts to different stimuli is possible to generate an emotional profile which will help to know how to adapt the system to a user.
- Design the use case for the human-cobot interaction.
  - The task that will make both actors interact between them must be collaborative. This means that they must interact with each other in order to perform it, being both essential for the task.
  - In addition, the task must be representative, being able to extrapolate its characteristics to other fields in which HRI could happen.
- Creation of an HMI application that allows sharing user's cognitive state to the cobot.
  - The application must be able to interpret the detected emotions and correlate them in a way that the cobot interpret them an change its behaviour.

1.4 Challenges and limitations

Diverse challenges could be faced in the development of the work being able to prejudice its results. Some of those challenges could be seen below:

- Emotion detection.
  - The emotion detection patterns and technologies are quite new nowadays so it might be challenging to be able to detect the cognitive state of the user.
  - Even if it is possible to detect the cognitive state, it could be the case that the detection will not be accurate enough to serve as input to the cobot.

- Analysis of emotions
    - As emotions are not constant but fluctuate, it might be challenging to analyse them properly in order to extract a pattern profile of the user.

It is important to remark that this thesis work is one of the first researching projects trying to assess the presented question in the exposed way. As a consequence, it does not pretend to generate a final commercial application to be applied directly on the industries, but the first idea of an application with margins to be improved in the future.

1.5 Document structure

This document is composed of six chapters. The first one is the introduction to the ideas behind the development of thesis work. Secondly, the second chapter presents the background concepts that are related to the topic of this project. Following, during the third chapter is presented the proposal to achieve the objective and the selection of techniques and technologies needed to implement it. Then, the fourth chapter is focused on the implementation of the work follows the presented objectives to achieve. The fifth chapter presents the results of the implementation and its conclusions. And lastly, the sixth chapter presents the future work.

2. STATE OF ART

In this section, it can be found the patents, literature and technology review that serve as base and context of this document. Among it, it will be described the most important concepts, ideas and information that must be understood in order to start the resolution of the problems and challenges exposed in the previous chapter.

This article is divided into five parts. The first part is dedicated to the concept of human cobot interaction and how this relation is trying to solve several problems that can be found in nowadays industry. In the second part, it is going to be discussed the different existent systems to detect emotions and how they have been applied. The third part will be about what is EEG technology and its application on engineering fields. Then, in the fourth section will be considered how industrial applications could be influenced by this type of technology. Later in the fifth part is focused on the seek of patents or possible works that have already solved the problem that concerns this document. Finally, the sixth part will provide a summary of what has been reviewed in the section.

2.1 Human-Robot Interaction

The use of robots in industrial production processes is rising. Every day, the figures of incorporated robots at the production lines are greater. Inevitably, humans and robots have to maintain interactions to achieve work objectives.

These interactions require both components to have characteristics that permit them to work towards the same goals. However, as the interactions are performed by at least a human and a robot, it is critical to ensure trust between each other. Moreover, another relevant aspect to take into consideration is the possible hazards that might appear from the interaction.

2.1.1 Human-Robot Trust

Nowadays HRI is a fundamental key in the industrial and production field. Inside the factories, systems are becoming more automated, consequently, more robots are being installed inside the plants having more operational control in the processes [8]. As a result robots' functionalities are increasing allowing them to be used in diverse types of operations. Considering the increment on the operation control of the robots, the role humans in the factories is changing to have supervisor duties over automation systems such as planning, instructing, controlling, intervening or training [26]. Therefore, the interactions between humans and robots are increasing, being relevant that robots have optimal skills ensuring high performance in stressful situations or increases in the workload [27].

The characteristics of HRI affect directly to the production results thus to the final product quality [28]. Due to that reason, it becomes extremely relevant to focus on how these interactions are handled regarding their quality, performance and reliability.

One of the aspects that has more impact on how HRI are managed and their outcomes is the concept of human-robot trust [23], [27]. Human-robot trust is linked with the confidence of humans in automation systems, a relation that has been widely studied [29]–[31]. Although both concepts of trust are related, there might be differences between them [27]. Regarding human-robot trust, it can be divided into various factors related to three elements: human, robot and environment [27]. The following figure shows some of the factors related to humans, robots and environment that have an influence on the quality of trust in the interaction.

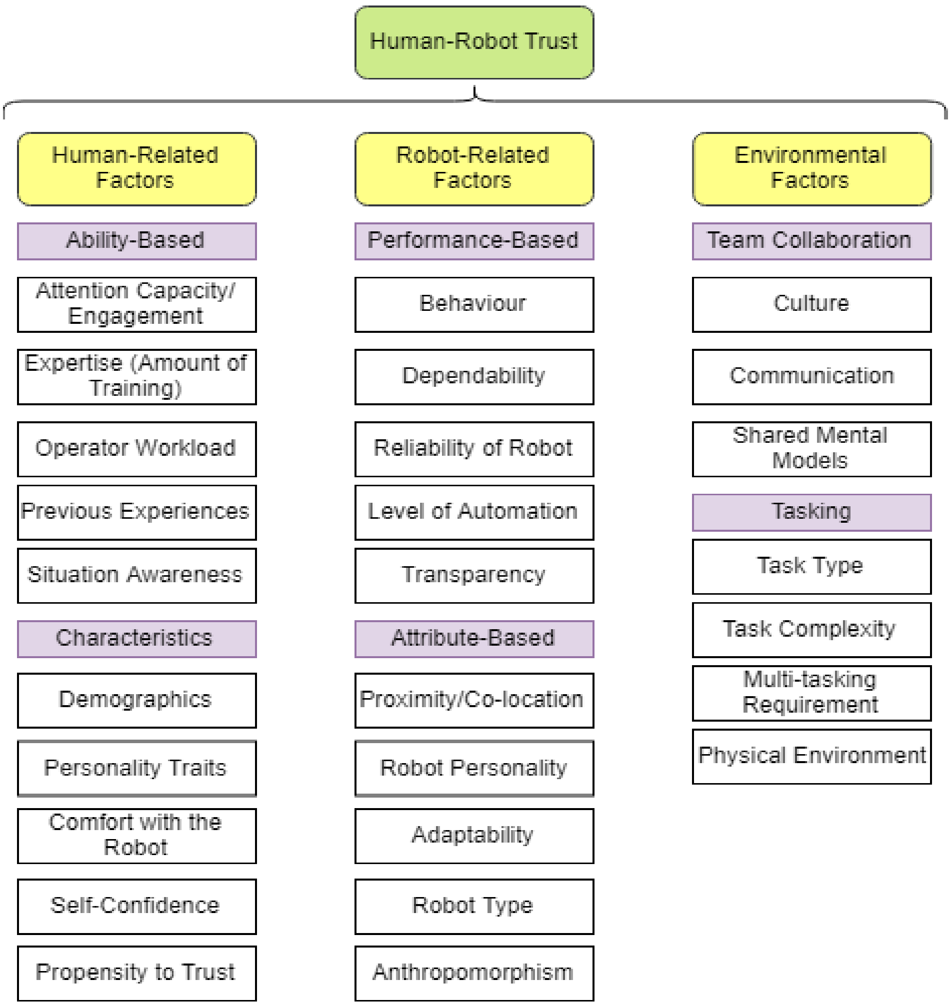


Figure 1. *. Factors of trust development in HRI (Adapted from [27])*

Observing the characteristics from the table, it is possible to see that there are several aspects that influence trust in HRI, being some of them more related with the cognitive aspect of the human and others closer to the characteristics of the robot or the process itself.

Regarding the human being, earns relevance taking a look in emotional aspects such as the comfortability with the robot. If there is a lack of comfort over the robot which the user has to work with, there will be a decrease in the trust interaction [23]. In addition, unpredictable variables such as workload and stress must be controlled to improve the interaction [23]. Now, linking with the robot aspects, the trust will be incremented depending on the Level of Automation (LoA) of the robot, and if the human has skills to adjust it [23]. This is related to the idea that the robot should behave as the user expect it to do it, incrementing its predictability and dependability [23], [32]–[34]. Moreover, robots with similar anthropology to human beings tend to impact positively to the user's comfort, principally when environmental factors as the task require it [35], [36]. Related with the environment, the robot must be aware of itself and its movements, knowing where it is located in case [23], [32], [33].

Finally, regarding the literature review, it seems a critical aspect developing adaptative automation systems in the human-robot team operations field [24], [37]–[39].

2.1.2 Hazards and ergonomics

Unlike machines, humans are in a certain way unpredictable. It is complicated to determine in a perfect and specific way their behaviour or their response to a certain situation. People can act by their free will evoking circumstances out of the outline. Human factors can be considered as a double-edged sword. On the one hand, they are extremely important in order to solve problems and situations that machines are not able to solve in a proper way. But on the other hand, they can be the main motive for accidents [40].

While working with robots, accidents can be categorized into three main groups [41]:

- Engineering failures: mechanical, electrical or software problems that can provoke a failure on the robot increasing its speed or acceleration, performing unpredicted abrupt movements, not stopping it, etc.

- Human behaviour: Poor performance of the work, service violations while working with the robot, breach of safety measures, disobeying of orders, bad attitude, fail of attention, lack of cognitive perception, etc.

- Environmental conditions: Installations' quality, clean atmosphere, optimal positioning of elements, level of noise, workplace measures, etc.

Among these factors, the ones that cause more accidents in different types of industrial applications are the environmental conditions and the human behaviour [42, p.], being the last one the cause of 90% of accidents occurred [25], [43]. The causes of these type of accidents have been intensively studied in seek of an answer. Nowadays, there are more studies that point to emotional aspects such as fatigue, stress and repetition as the main cause of the accidents. It is said that 48.8% of the accidents are strongly related to the abovementioned features [25].

As the characteristics that are causing problems are related to the concept of ergonomics, companies are investing more effort and resources to improve in that field. The concept of ergonomics have been defined by *McCormick and Sanders*, such as: "Ergonomics applies information about human behaviour, abilities and limitations and other characteristics to the design of tools, machines, tasks, jobs and environments for productive, safe, comfortable and effective human use" [44]. This concept has been undervalued among managers being just seen as a tool to improve the safety and health of the employees. But, at the same time is improving the personal state of workers while performing their duties, it is enhancing the quality of the process, final product, productivity and it is reducing costs [45].

It is known that without a correct psychological condition and mindset, the performing of tasks will never be as optimal as it could have been. In fact, [45] makes a correlation of studies about how social, psychological and cognitive ergonomics factors can have a straight impact on quality. Amid these studies, the one performed by *Elkman* [46], takes into consideration, in assembly tasks, various ergonomics problems such us: physical demands, assembly difficulty, and psychologically demanding works. Of these categories, the one with a higher value of quality deficiency was related to psychological concerns; with a result of 70% of failures in the performed tasks.

With all these data, seems more than necessary to find solutions to reduce the number of hazards related to human conditions.

2.1.3 Human-Cobot Interaction

Focusing on the robotic industry, this situation is trying to be addressed by the development of HRI. Each day this field is earning a bigger role through factories, creating more advances to enhance safety, performance and comfortability. Some of the improvements that have been implemented in the systems of the robots are, the placement of sensors in order to detect the position of a human avoiding the possible impact, reduction of movements' velocity and acceleration, collision detection, reduction of the working range, etc. [41]. By the application of these characteristics, robots start to become a more feasible option to work in a collaborative way with employees, without the fear of having hazards.

In this context is where merges the concept of Human-Robot-Collaboration (HRC). HRC refers specifically to the collaboration between robots and humans to achieve a common objective. In the other part, HRI is a more generic term which includes the act of collaboration and other concepts.

As it has been mentioned in the *Introduction* chapter, technology is closer than ever to society. Because of that, HRC is starting to be implemented outside the industry field, getting closer to atmospheres governed by human skills such as offices, reception of establishments, hospitals, homes, etc. The nearer robots are to humans, the more they need to adapt their characteristics to understand human behaviour and to not be seen as a hazard for them. Due to that reason, it raises the idea of developing a new class of robots; robots that can perform collaborative tasks with humans as colleagues. This new era of robots is called *Co*llaborative *ro*bots or just *cobots.*

A *cobot* is a robot meant to work in a collaborative way with humans, sharing the same workplace and acting as their colleagues. The main difference of this concept in comparison with robots resides in that they are no longer placed inside cells. Since the cells isolating the robots from the frequented zones by the workers have been eliminated, the *cobots* must have incorporated tools that allow them to perform their main functions without being seen as a danger in the industry or outside it. As mentioned before, developers are creating the *cobots* with slower performance speed than robots, a less robust appearance, lighter weight, cameras and with several contact sensors in their links and joints so when a person touches them they can stop their movement. These specifications give them versatility, being able to adapt to different work situations, environments and persons; always trying to assist in the best possible way to humans [47].

Cobots still having most of the abilities that robots have. They are still being able to perform dangerous, difficult, monotonous, dirty or boring tasks that lead workers to non-optimal mindsets. With this adaptability, it is trying to leave *cobots* to do the part of the work that causes problematics to humans; letting to the employees the parts of the process that can only be performed by human beings.

There are several types of *cobots*, each of them are meant to assist in a different type of situations. Researching between the biggest developers of robots, it can be found some examples of the best and most used *cobots* nowadays [48]:

- *Sawyer*, is a *cobot* developed by *Rethink Robotics*. Some of their main functionalities are applied to pick-and-place tasks, co-packing and packaging, stamping and quality inspection [49].

Figure 2 . *Sawyer Rethink Robotics cobot [49].*

- *YuMi*, created by *ABB robotics* [50], is a dual-arm robot developed for automation applications such us assembly process. Even if this has been the main application, it has been used for different tasks such as, making paper aeroplanes or conducting a philharmonic orchestra.

Figure 3. *YuMi ABB Robotics cobot [50]*

- *UR3, UR5* and *UR10* are *cobots* offered by *Universal Robots* [51]. Each of them with a different payload, these *cobots* have abilities like screwing, soldering, using tools, painting, etc. These abilities made them suitable for assisting *BMW* assembly process.

Figure 4. *UR3, UR5, UR10, Universal Robotics cobots [51]*

The development of *cobots* has been a game-changer on the working concept. In order to introduce them into the industry and other fields, it is necessary to define a common goal for the worker and the *cobot*. The performance of this common goal needs to use a combination of human and *cobot* abilities, but the *cobot* ones should be adapted to the human needs [52]. To apply this in a proper way, the user should specify to the *cobot* her or his needs, so it can perform an action or another one. This exchange of information is necessary in order to improve the communication issues between robots and workers lived nowadays.

The main consequence of the lack of perfect harmony between humans and robots is that people are relatively unpredictable. The environment that surrounds a human being is dynamic and the emotional factors that determine it fluctuate through the day. Therefore, is extremely complex for a robot to predict, without human feedback, which is the ideal action that must be carried out to assist a person. In turn, it is complicated for a person to communicate with a robot instantaneously and without programming in order to correct it or transmit information on-live. If a communication improvement in both directions is achieved the effectiveness in performing tasks would improve abruptly.

One of the main targets for *cobots* while working with humans is to be able to process and understand their requirements, that commonly they are used to be linked to their emotional state. Having said that, the following question enters into the debate: is the actual technology applied on *cobots* enough to be considered a safe and real colleague? Or is it necessary to improve the communication between both actors?

In order to improve that communication deficiency, it raises the possibility of using devices able to detect and analyses the emotional state of a user and transmit a certain type of information to the *cobot*. Currently, it can be relevant for the industries detecting the levels of fatigue, concentration, stress, fear, etc. These characteristics are strongly correlated with the emotional response of an individual, therefore it becomes relevant to detect them. By interpreting the employee's emotions, it is possible to know their cognitive state and link it to a *cobot*. In this way, the *cobot* will know how to adapt its performance to its colleague, for example by regulating its working velocity, stopping its performance of an action that endangers the worker, modifying its path, etc.

2.2 Emotion-driven systems

Nowadays, the detection of mental states of a person is being studied in a deep way. The reason behind this is that the identification of these states will help to achieve a better personal interaction between users and devices. So as to give the user the best experience, different approaches have been developed to identify how the user is feeling while interacting with the device. Some of the most commonly used methods on HCI devices are based on:

- Facial recognition: Detecting feature variations in the face's muscles. The data is usually extracted from eyebrows, eyes, nose and mouth [53]–[55].

- Speech recognition: Distinguishing speaker's acoustic and lexical characteristics used in the speech [56], [57].

- Body language: Identifying shapes and postures of a person. Normally obtained from the combination of head, arms, hands, torso and legs placements [58].

- EEG: Interpreting the apparition and intensity of brainwaves in the different brain lobes [59]–[61].

- Electrocardiogram (ECG): Measuring the cardiac frequency of a person, inter and within beats, and other parameters from the ECG [62].

Before explaining briefly the abovementioned technologies to detect emotions, it is relevant to define and analyse two of the most used approaches to classify human emotions.

2.2.1 Emotion-classification methodologies

Usually, there are two ways of classifying emotions. The first one is substantiated in distinguishing basic emotions. As with the primary colours, these emotions are the base to create derivated ones. One example of this type of classification is proposed by Plutchik, identifying eight basic emotions: fear, anger, joy, sadness, acceptance, disgust, expectancy, surprise [63]. The second methodology uses a dimensional space to classify the emotions depending on which are its axes values. This approach was first proposed by J. A. Russell defining a bipolar space composed by the dimensions: valence and arousal [64].

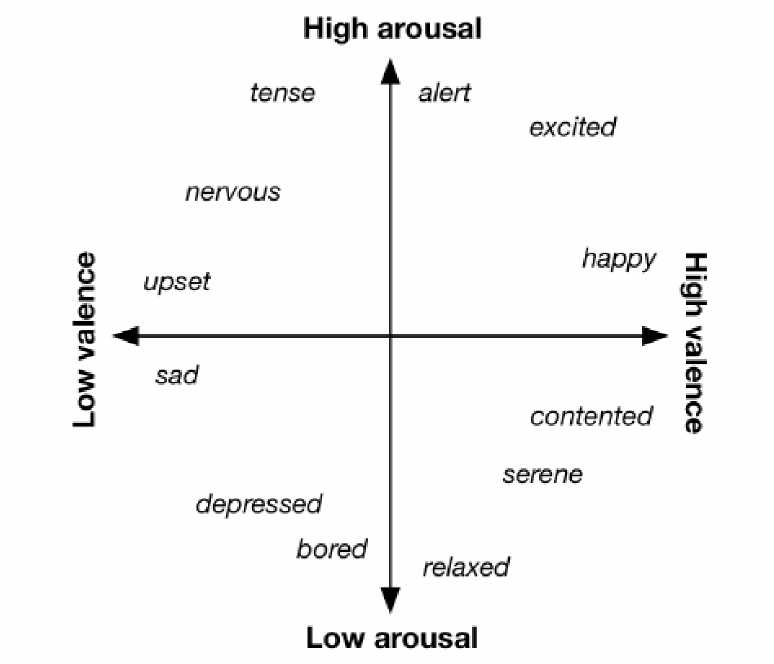


Figure 5. *Valence-Arousal Model [65]*

The main problem of this approach is that if some emotions have a similar value of *Valence* and *Arousal*, such as fear, stress or anger, the method is not able to distinguish between them. So as to solve the commented issue, the method was upgraded by Mehrabian and Russell to a three-dimensional approach composed by Pleasure-Arousal-Dominance (PAD).

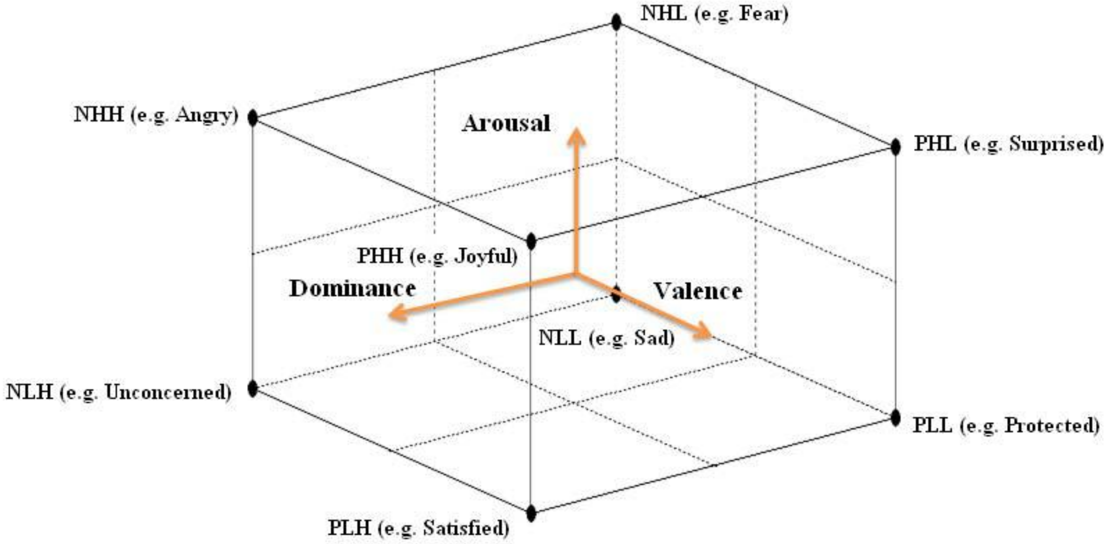


Figure 6. *PAD Model [60]*

Being the plane defined by *Pleasure-Arousa*l the same as the *Valence-Arousal*. With the new dimension, *Dominance*, it is possible to solve the problem suggested before. Now, if there are emotions with similar *Pleasure* and *Arousal*, they could be differentiated because each of them will have a different Dominance rate.

Between the two ways of classifying emotions explained, the dimensional method is widely used on research due to its ease of use, adaptability and capacity to distinguish among emotions different than the basic ones in a specific way.

2.2.2 Facial recognition

When thinking about detecting human emotions the first thing that usually comes into the mind is the face of a person. The face of a human being is composed of 43 muscles. Combining these muscles, it is possible to create different countenances. As a popular proverb says: “Face is the reflection of soul”, this makes reference to the fact that usually, everything that affects human beings giving them an emotional input, is reflected instantaneously and subconsciously with a facial expression. Due to this reason, facial recognition is one of the most used tools in the analysis of the human emotional state.

This method of emotion recognition is based on basic emotions methodology. There are several lists of basic emotions depending on the researchers that enlisted them [64]. But the common point among them is that they are describing all the human emotions with the combination of these basic ones.

So as to detect the emotional state of a person with a device, is necessary to detect the variations in the face's muscles. One of the most used methods to fulfil this goal is to track the movements of eyebrows, eyes, nose and mouth, creating a map of points [56], [66]. Then, is analysed the motion of the movements and compared with a database [64], [66].

There are some issues related to this approach of detection. The first one is related to the use of basic emotions for the classification. It is expected that the description of the basic emotions fits athwart all cultures of the earth, but as it has been discovered, not all cultures answer with the same countenance to the same stimulus [53]. This creates problematics because as a consequence, devices should modify their characteristics to perform correctly with people of one and another part of the globe. The second issue is related to the control of facial muscles. As has been mentioned, the recognition is based on the movements of the features. Even if most of the people do not have the ability naturally, is possible to train to have control of face muscle. With this skill, is possible to fake expressions even if a person is feeling the opposite emotion. A perfect example of this could be actors and actresses that must imitate expressions of different characters.

2.2.3 Speech recognition

The most common and dominated method of communication among humans is verbal. That is why when is time to express ideas, feelings or information, speech is the chosen way to do it.

As simple as it could seem, verbal communication hides more characteristics than it resembles. Properties such as frequency, pitch, resonance, fluency, vocabulary, can define the age, sex, utterance and more details of the speaker. Consequently, the study of speech recognition is being developed intensively so it can be one of the best tools to improve HCI.

In order to distinguish between emotions through the speech, most of the approaches use characteristics related to the sound such as the pitch, energy and resonance to classify the arousal level of the emotions. In the other side, as is hard to differ the valence with the same properties, linguistic related features as fluency and lexical are used to fulfil this objective [54]. Once analysed the main attributes, it is necessary to use an optimal database to complete the classification.

The issues that are facing this approach are related to the accuracy of the detection. Just as it results easy to classify the arousal of emotion it results more complicated to distinguish the valence to complete the classification [67]. In addition, the databases must be quite accurate and as with the facial expressions, speech can be controlled to emulate emotional states.

2.2.4 Body language

Even being the most used way of subconscious communication, unlike verbal communication, body language is not a commonly dominated pattern of communication. Therefore is a powerful way to detect emotions in human beings.

While performing any activity such as conversations, walking, sitting, or being in a social atmosphere, people express their feelings through words and gestures. The different gestures and body positions that people are adopting in these situations is reflecting unconsciously how they feel about what they are doing, thinking or feeling. Is that huge the impact of the body language that is said that in an interaction, 65% of the emotional output is due to non-verbal communication [57].

The non-verbal communication is composed of several aspects such as body positioning, gestures, facial features, gaze movements, etc. But the most common way to detect a person’s utterance is by studying the shapes and postures that are adopting by combining head, arms, hands, torso and legs placements.

Different from other types of recognition, it is quite important to differentiate between gender and age. There are some body language patterns that are more characteristic in one gender than in the other, and in an age range than in other.

The detection of emotional states is usually achieved by the use of cameras that distinguish body postures and an emotion model database which relates those positions to the emotional states.

Similar to the previous methods, an issue to face is the cultural differences and the ability to control this characteristic in purpose.

2.2.5 Electroencephalography

EEG is a non-invasive neurophysiological exploration based on the recording of the bioelectrical signals produced by the brain. All the data created by the brain activity is registered in a graph which has the name of electroencephalogram. In order to detect brain signals, it is necessary to have an electroencephalogram system. This system is composed of several electrodes which must be placed on the scalp in order to measure the electrical activity of the brain. This electrical activity generates brainwaves which can be categorized depending on their frequency in four types, delta, theta, alpha, beta and gamma [68].

- Delta waves [0.5, 4] Hz: This state is commonly achieved while sleeping on the dreamless state, being the main brainwave in this state. Psychologically is linked to the personal unconscious.

- Theta waves [4, 8] Hz: Theta rhythm is generated in REM phase and dreaming state while sleeping, on deep meditation states and flow state. It is said that is the door to consciousness and is related to deep memories stored in the subconscious.

- Alpha waves [8, 12] Hz: Is the most common brainwave due to the fact that is generated in the awake state by being focused and relaxed. By closing the eyes alpha waves are generated because it gives the information to the brain that it is possible to be relaxed.

- Beta waves [12, 25] Hz: These waves used to be created in moments of high attention, alertness, thinking, or moments of analysis. Is typical in task-oriented moments situations. As it has a great range, it used to be divided into three categories:
    - Low beta [12, 15] Hz: Related to an idle state.
    - Beta [15, 23] Hz: Linked with task-oriented work, high engagement or processing.
    - High beta [23, 38] Hz: Associated with complex reasoning, stress or anxiety or new experiences.

- Gamma waves [25, 45] Hz: The brain is flooded with gamma waves while performing voluntary movements, processing of information, multi-tasking, great inspiration or even enlightenment moments. Great meditators are able to generate gamma waves while meditating.

Regarding the activity performed, the brain of a person will be flooded by different types of waves and in several lobes. According to this, if different activities produce various patterns of brain activity, different emotions will evoke a similar outcome. This hypothesis has been tested and confirmed by several studies and correlated with the PAD model.

Following the PAD model, it has been discovered a correlation between the *Pleasure (Valence)* factor and the activation of the hemispheres of the brain. This activation is correlated with the apparition of alpha waves in each hemisphere, which are related to brain activity [59], [61], [69], [70]. Concerning the arousal axe, as beta waves are linked to an alert state of mind and alpha waves to a dominant relaxed mindset, the ratio between beta and alpha will determine the arousal levels of an individual [60], [61], [69]. Finally, in order to determine the dominance state, it is necessary to observe the combination of an increase in the beta/alpha ratio in the frontal lobe and an increase in the beta activity at the parietal lobe [60], [61], [69], [70].

By the use of these methodologies, different emotional states are recognised using EEG.

2.2.6 Electrocardiogram

Similar to EEG, ECG is a non-invasive procedure through which is possible to measure the electrical activity of the heart [71]. With each heartbeat, an electric signal is generated. The electrocardiograph is the instrument in charge of recording those signals. It uses several electrodes placed in different positions in order to detect the beats and register them into a graph. By the analysis of this graph is possible to detect a great amount of information about the patient.

So as to detect emotions by the use of ECG, several techniques have been applied which measures the cardiac frequency, inter and within beats and other parameters from the ECG graph. After extract these data, they have been performing experiments searching for reaction patterns, correlate and analyse the data, and classify the emotions using the valence-arousal model or other methods such as the local binary and ternary patterns [62].

ECG has in its side that is quite difficult for most of the people to control their heartbeats, so it's a method that, opposite to facial expression or body language, is hard to fake and hide emotions from it.

2.2.7 Emotion-driven applications examples

As it has been mentioned, HCI is a field that is earning more weight in all fields of society. The reflection of this reality can be seen with the number of instruments that are being developed in the shake of improving this area. Some examples of nowadays advances are shown next.

- Roboy [72]: It is a robot that tries to emulate human behaviour in order to improve HRI. The project was created in the University of Zurich and later moved to the

Technical University of Munich where it still in development. With several characteristics on mechatronics, control, cognition and more areas, *Roboy* is able to understand the users, maintain conversations, learn about its interactions and give an emotional response to the inputs that it is receiving.

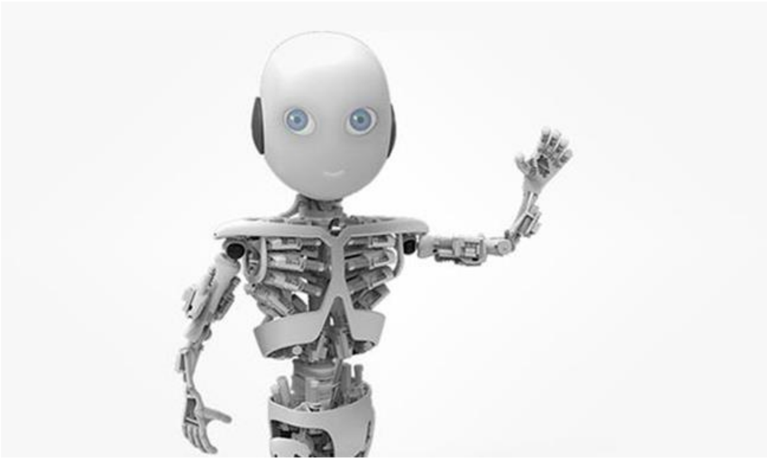


***Figure 7**. Roboy Humanoid [72]*

- Affectiva [73]: Is a company focused on the detection, measurement and analysis of human states in order to create applications that improve HCI. By the use of artificial intelligence, verbal language analysis, facial recognition, deep learning and huge databases, they are able to determine emotions and give an answer to them. Some of its applications are related to improving marketing aspects such as commercials; analysis of reactions in political debates, gaming or monitoring attention, drowsiness and other human characteristics while driving.

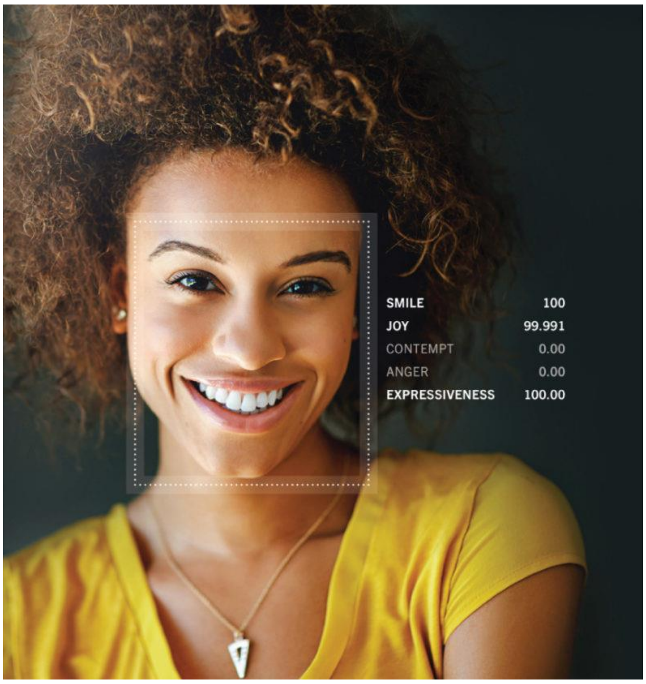


Figure 8. *Affectiva Face-Emotion Detection [73]*

- Pepper [74]: Developed by *SoftBank* Robotics, *Pepper* is a humanoid robot able to recognize basic human emotional states, give a proper answer to that specific state and perform interactions with people. It is being used in multiples companies as a reception assistant, tour guide and other employs and in educational projects.

Figure 9. *Pepper Robot [74]*

- NAO [74] Created by *SoftBank* Robotics as *Pepper*, *NAO* is a programmable robot created with the aim of increasing the advances in HRI, helping people of all backgrounds to get used to interactions with robots. As it could be programmed to fulfil a great number of tasks, this adaptability has made *NAO* a great tool for educational, healthcare and even entertainment porpuses.

Figure 10. *NAO Robot [74]*

2.3 EEG technology

As it has been mentioned before, EEG is a non-invasive technique that allows visualizing the electrical behaviour of the brain. It uses multiple electrodes disposed on the scalp. Each of these electrodes is measuring the voltage generated by the neuronal activity in a specific position. Usually, the electrodes are distributed along the scalp following the 10-20 system [75]. This is an international methodology used to place the electrodes all over the scalp in a normalized way, achieving a communion between all the studies performed with EEG technology.

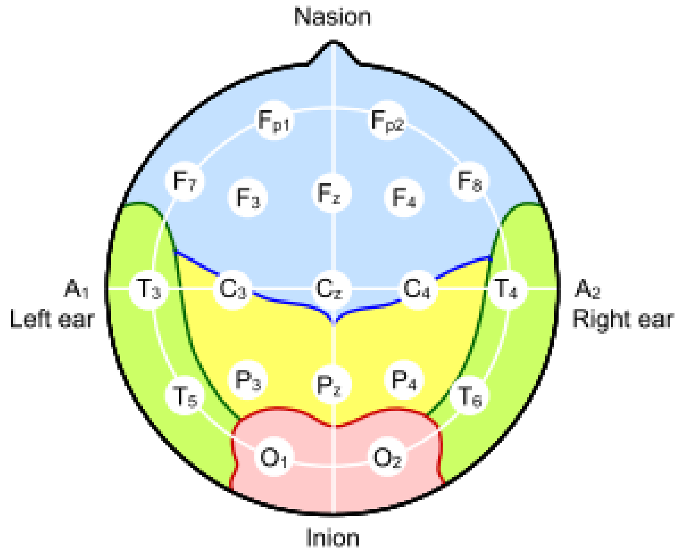


Figure 11. *10-20 System [59]*

In order to fulfil the objectives proposed in this document, it is necessary to find an EEG device for monitoring the brain activity an interpret the emotions of the user.

By researching on the internet, it is possible to find several companies that are developing EEG headsets. In order to know which of the headsets offered in the market fits better in the approach of this master thesis, a study between some of the most relevant companies nowadays has been made. Among the selected companies, a brief description of what is offering each of them and what are their strong and weak characteristics is presented.

- Neurosky [76]: The headset offered by the company is based on one EEG dry electrode placed on the forehead at the FP1 position of the 10-20 system. Neurosky is offering diverse software that records brainwaves data, raw signals, and metrics for attention and meditation states. One of the bad aspects of the offered headset is that it only has one EEG electrode and in a first analysis, the more electrodes the more accuracy of the signals.

Figure 12. *Neurosky Headset [76]*

- OpenBC [77]I: This company sells 3D printed headsets which are possible to print in a particular printer too. The headset has 35 different locations to place the electrodes, following the 10-20 system, and is possible to track about 16 EEG channels. Is it possible to get their open source software to get raw data for free, but some other parts of the headset such as hardware and electrodes have to be ordered separately. OpenBCI devices are not offering a straight detection of mental states, so in order to know them, it would be necessary for more computational work.

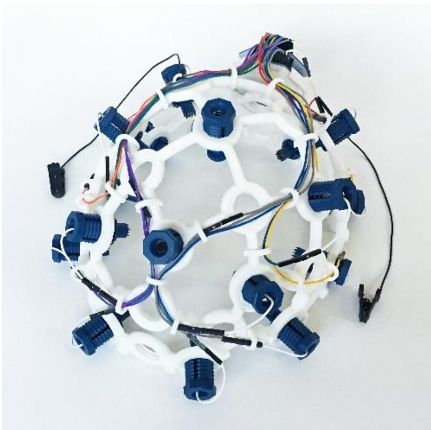


Figure 13. *OpenBCI – Ultracortex "Mark IV" EEG Headset [77]*

- Emotiv [78]: Mainly focused on the development of headsets, Emotiv has three different models, all of them following the 10-20 positioning system: Insight, Epoc+ and Epoc Flex. Insight has 5 fixed locations for the dry electrodes. The Epoc+ headset has 14 EEG channels and it uses saline electrodes. Finally, there is the Epoc Flex with 34 sensors in its structure. In order to extract data, Emotiv has different software which may allow mental commands training, raw data monitoring or even emotional detection. The emotions detected are stress, engagement, interest, focus, excitement and relaxation. But in order to obtain high accuracy data is necessary to buy a PRO license that increases the costs.

Figure 14. *Emotiv Headsets: Insight, Epoc+, Epoc Felx (from left to right) [78]*

- <u>Cognionics</u> [79]: Cognionics has a broad range of headsets with different amount of electrodes each. In order to increase the accuracy, the headsets use different types of sensors, dries or salines depending if they are placed on the skin or on the hair. Several types of software allow data acquisition and creation of applications in various programming languages. The main problem is that in comparison to the other options the devices are quite expensive ($6,500 for the one with 8 channels).

Figure 15. *Cognionics Headset [79]*

- <u>ANT Neuro</u> [80]: Unlike the exposed options, ANT Neuro is more focused on the development of medical applications. Due to that, their EEG caps have more locations for the electrodes in order to achieve the best accuracy possible, their aesthetic is more similar to a cap and the cost of the products are more expensive. Moreover, the offered software are not that focused to develop different applications or even to detect emotional states.

Figure 16. *ANT Neuro Cap [80]*

Later in the document, it will be selected one of the options that had been introduced as a possibility to apply EEG technology to fulfil the objectives presented in the first chapter.

2.4 Industrial applications

There have been plenty of changes during the evolution of the industry [3]. From handmade products to automated procedures, from small factories to giga-factories, or from human-human interactions to the concept of HRC [30], [52], [81]. As it was mentioned in the previous chapters, the industry is evolving in a fast way, and the new technologies described previously are introducing a new change in the industry globe. Same as it happened with other technologies such as robots, new technologies will need time to evolve and generate a positive impact in the industry [10], [82].

There are always various approaches for performing, and the technologies described in the previous sections are leading to these patterns. For example, the integration of cobots in the factory lines is having an effect on the behaviour of the employees and in their duties inside work [26]. The workload that right now a single person is experimenting could be reduced by the assistance of a cobot performing a collaborative task [83]. As a result of the new perspective of a more friendly and adaptative atmosphere is being generated with the application of cobots in the industrial lines [83].

In addition, the use of adaptive systems is generating changes in how human-machine interactions are interpreted [8]. With the perspective of keeping human in the loop, there are being developed systems capable of adapt autonomously to changes in the environment, user state, etc [7].

The actual context in which industrial applications are being developed is having more variables to take in consideration becoming a more complex [7], as a result, the technology that surrounds these applications are adopting different approaches to assess the challenges that coming from that context. Some of the characteristics that are being taking into are considering cognitive and emotional aspects [22].

2.5 Patents landscape

Once all the technologies and methodologies that serve as foundation for this master thesis have been explained, it seems interesting to seek for possible patents and applications related to this work. In order to do that, research among patents has been done. The application *Derwent Innovation* [84] has been used to fulfil that propose.

So as to seek for the most accurate and relevant results that concern the development of this thesis, two types of searches were performed. The first one was more focused on discovering patents related to the detection of emotions and use on HMI applications. In the other hand, the second research was centred on Brain-Computer Interaction (BCI), use of EEG and robot-control applications. In order to search on the application, it is necessary to use specific keywords that must be correlated with the topic that concerns the research. Because of that, on the first search, the keywords used were: "*Emotion detection"* and *"Human-Computer Interaction"*; and in the second one: *"EEG"* and *"Robot brain control"*.

Combining both searches, it was necessary to filter some of the results that were offered by the application. The reason for this filtering is that the program is searching for mostly all of the patents that have something in relation to the specific keywords that were introduced. As a result, there are a great number of patents that are not of the interest of this thesis work, even if they are related to the keywords. An easy example to understand this is that just by searching for *"EEG"* there were hundreds of medical applications related to that topic but that they are not related with the understanding of emotions or with the control of applications, machinery or other devices.

Once the filtering of results was applied, it was possible to extract the relevant data needed for the analysis of information. With this data, there is the chance to see in which parts of the world this topic of study has been more developed and where there are not investing too many resources investigating it. In addition, it is offered a graph in which it can be seen the progression through the time of how many patents have been done in the previous years. Finally, there are some bar graphs about the top assignees and inventors that more patents have on the field. These graphics are going to be commented in the next paragraphs.

First of all, in the picture below, it can be seen a map in which there are coloured on red, orange and pink the countries that have created patents in the fields of the research. The stronger the colour the more patents the country has developed. As it can be appreciated on the legend, China is the region that has invested more in this area with 78 patents, followed by the United States of America with 47 patents and by the Republic of Korea with 9 patents.

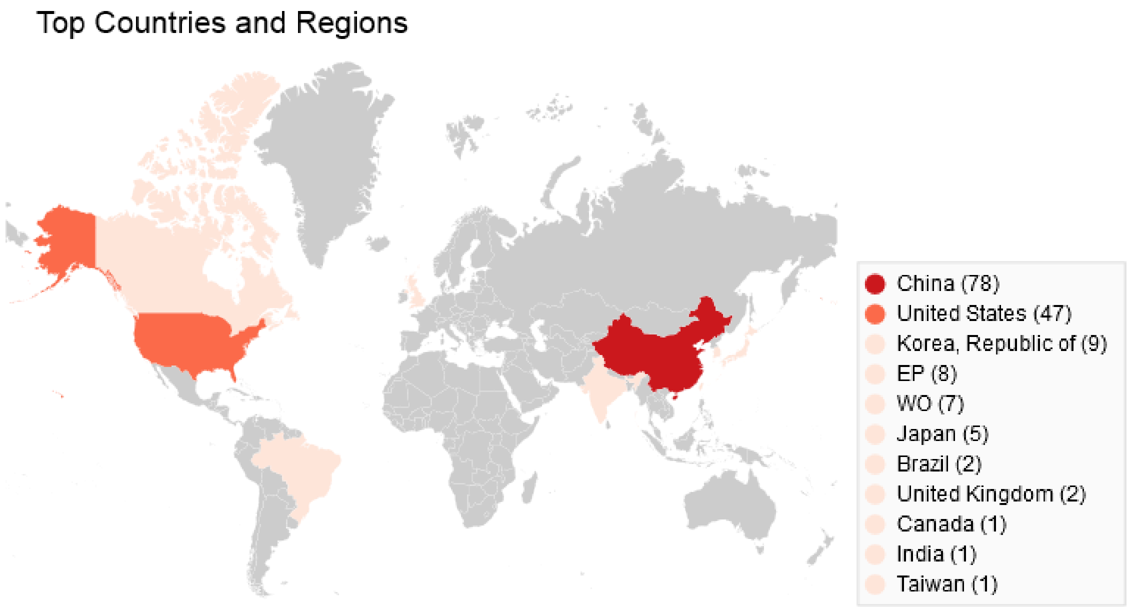


Figure 17. *Countries and regions with more patents.*

Secondly, one of the most relevant graphs shared by *Derwent Innovation* is the one that represents how the amount of patents on the area has evolved over the last years. Here it is easy to recognize how the need for creating systems with the ability to understand human behaviour in order to adapt their characteristics, has increased tremendously since the last 10 years. Since the year 2011, the tendency is growing, and even if in the last year, 2018 it has been a decrease, it is more than feasible that it starts to increase again during the next years.

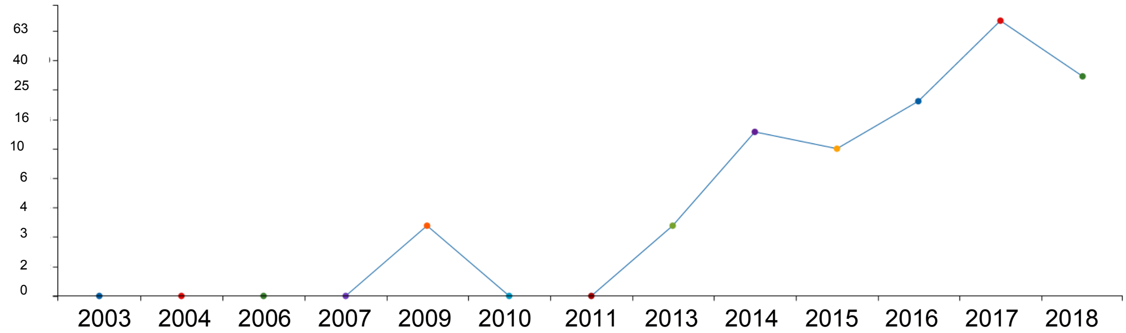


Figure 1. *. Number of patents developed each year*

The following bar graphs will show the assignees and the inventors with more patents in their names. Starting with the assignee's graph, just by looking at the first five the weight of China in this area of investigation gets remarkable, being the first two with more patents. Almost the same effect happens on the inventors' graph, where it can be a relationship between the inventors and the assignees that have more presence.

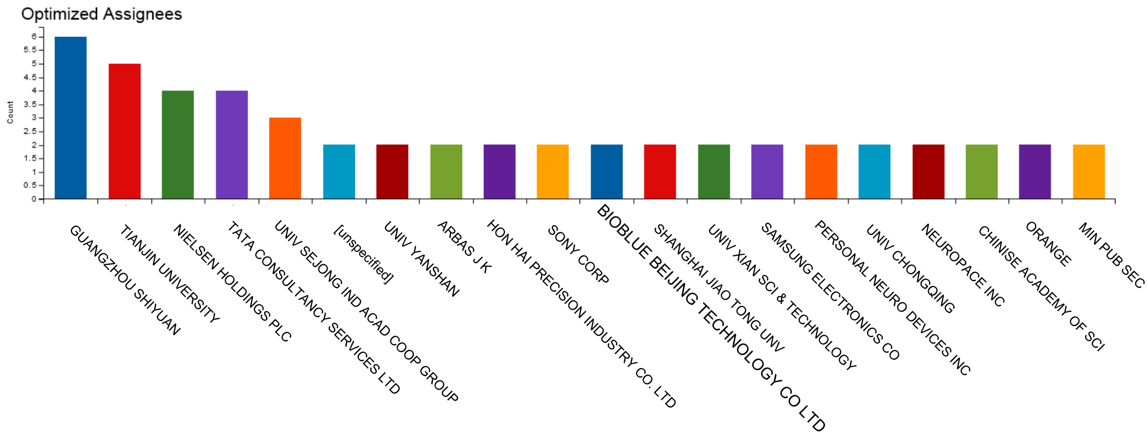


Figure 18. *Assignees' Patents Graph*

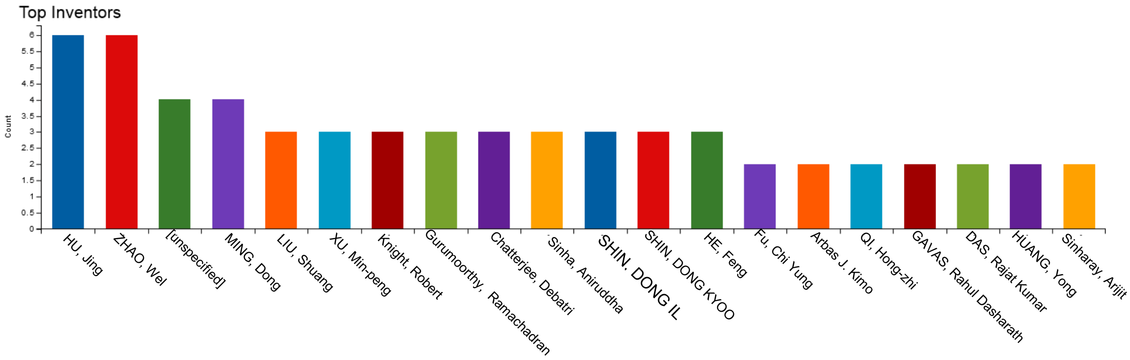


Figure 19. *Inventors' Patents Graph*

Finally, the last an probably most relevant graph extracted from the application is the "Patents Map". The map is developed with the information of all the patents included in the research, creating a topographical representation. Each patent is presented on the map as a point. So if two patents have a similar topic, their points will be close on the map. On the contrary, if their topics are not similar, they will be close to each other. The greater the number of points on a zone the greater the "mountain". The same in the opposite way, the fewer points, the flatter the topographical representation will be. In addition, it is added a "collective" name for the different zones in which there are patents gathered. These names try to simplify in a few words the topics of the patents gathered.

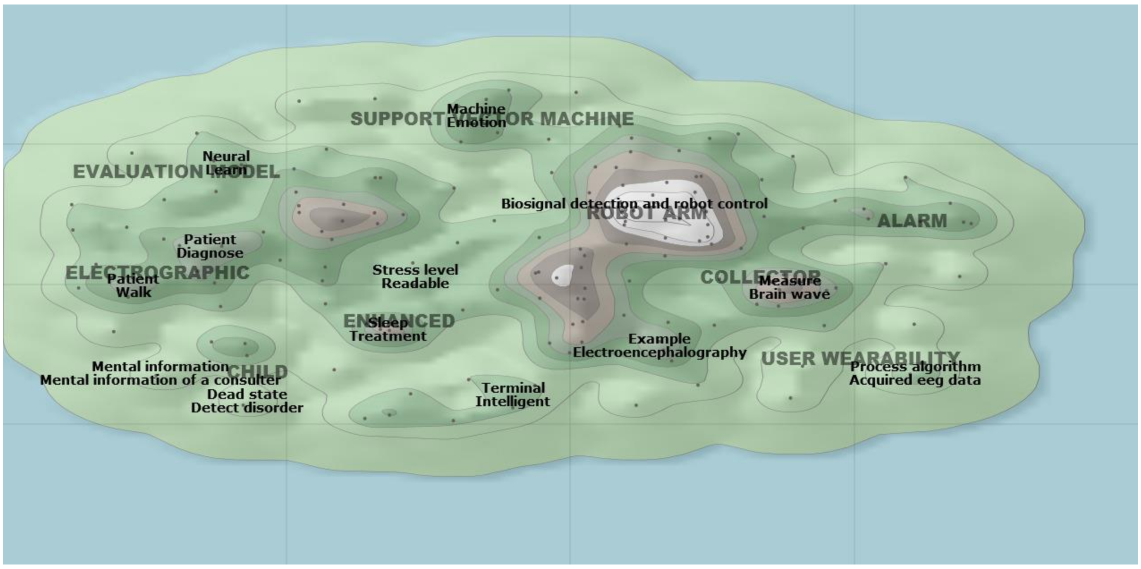


Figure 20. *Patents Map*

This topographical representation very representative because just with a glance, it is possible to see which areas have been invested more time and resources on researching. In this case, the topic with more patents is the *“Biosignal detection and robot control”* which fits with the areas that this document is in contact with. To finish, it is important to mention that, even if there has been more investigation in this topic than in the others and it is correlated with the one that it is being developed in this text; it has not been created a solution for the challenges exposed in the first section of the document.

2.6 Summary

Along with this chapter, it has been explained the necessary related work to understand the basis in which this document is developed. The aim of this work is to provide a solution to the problems related to HRI. Most of these problems have as a common denominator the human factor which can be it solution too and is strongly related to human’s emotional states. It has been seen that improving human-robot trust is a great tool to achieve a positive influence on HRI. In addition, trust in HRI can asses several issues related with human’s emotional state while working with robots. Being stress and fatigue the cognitive states that more accidents provoke between human and robots, it is important to provide solutions that can handle this issue. As human and robots are working like a tandem, it earns relevance to make changes on both sides. First, in some cases such as collaborative tasks, robots are being substituted by cobots, which are collaborative robots with special characteristics that allow them to work properly and safely with humans. In the second place, there have been developed different methodologies to detect and interpret human emotions, which are the nucleus of the next evolution of the industry.

The goal of this work is to influence positively in a human-cobot interaction by using adjusting the behaviour of the cobot using the cognitive state of the human.

Following, some tables are presented to compile some of the main ideas presented during this chapter.

Table 1. *Characteristic of emotion recognition methodologies*

	Detected by	Classification Approach	Ease to control consciously	Dependency of an interaction
Facial recognition	Variations in face's muscles (eyebrows, eyes, mouth…)	Basic emotions	Easy	Middle
Speech recognition	Sound and linguistic characteristics (pitch, resonance, vocabulary, grammar…)	Basic emotions	Easy	High
Body language	Body positioning, gestures, facial features…	Basic emotions	Middle	Middle
EEG	Measuring brain's electrical activity via electrodes	Valence-Arousal/PAD	Difficult	Low
ECG	Measuring heart's electrical activity	Valence-Arousal/PAD	Difficult	Low

Table 2. *Characteristics of emotion recognition devices*

	Headset model	**Number of channels**	**Type of electrodes**	**Mental states detection**	**Application Development Software**	**Price**
Neurosky	*Mindwave Mobile 2*	1	Dry	Attention and meditation	OpenAPI	$99
OpenBCI	*Ultracortex "Mark IV"*	16 max.	Dry or gel	No	3rd Party Software based on signal processing and data analysis	$599
Emotiv	*Insight*	5	Semi-dry polymer	Excitement, engagement, relaxation, stress, focus	Based on JSON and WebSockets, Supports: Java, C#, C++, NodeJS, Python…	$299
	Epoc+	14	Saline	Excitement, engangement, relaxation, stress, focus		$799
	Epoc Flex	32	Saline	No		$1,699
			Gel			$2,099
Cognionics	*Quick-8*	8	Dry or saline	No	OpenAPI	$6,500
	Quick-20	20	Dry or saline			$14,600
	Quick-30	30	Dry or saline			$22,000
ANT Neuro		32-256 max.	Dry	No	SDK based on C++ And toolboxes based on SDK (Python, Matlab…)	~$25,000+

3. PROPOSAL AND SELECTION

During this chapter is going to be exposed firstly a proposal which integrates several concepts referenced in along the previous chapter. This proposal will provide the foundations of a use case for building an implementation which expands the presented idea to a real scenery. Secondly, there will be performed a selection of ideas and technologies that will be used as tools for developing the proposal. Finally, using the selected tools, it will be explained how the proposal has been implemented to fulfil the goals

3.1 Proposal statement

As it has been defined in the introduction chapter, this thesis work is following the objective of proposing and developing an approach to make cobots adopt to human's emotional states influencing favourably the HRI aspects.

In order to fulfil the objective of this thesis work, it is necessary to find a proper use case that represents the relevant requirements of the system that will be developed.

First of all, the user will be working with a cobot in a collaborative way. Defined previously, a cobot is a robot meant for work with a human in a collaborative way. One of its main objectives is to enhance the trust of a person while working with a robot [85]. The cobot must assist in the best possible way to the user achieving a common goal.

Secondly, the interaction between the user and cobot should be realised through a collaborative task. This means that both roles, user and cobot, are essential for performing the task. In addition, the task itself should be representative of being able to extrapolate the benefits and characteristics of the task to other fields.

While performing the collaborative, user's cognitive state will suffer some variations. These changes might have an influence on human-cobot interaction. So as to discover how to interpret these mental state variations, it would be necessary to find out a way of tracking those modifications.

Finally, in order to conduct the mental state influence in a beneficial way, the cognitive state should be given as an input to the cobot modifying its characteristics adapting them to the user. It must be found a communication pattern between user and cobot in order to share the cognitive state to the cobot so it can modify its characteristics to match properly the user needs.

The use case that defines the basic specifications that should have a system that tries to cope with the propositions explained in the previous paragraphs, must be composed of the following elements: cobot, user, cognitive detection device and HMI. A diagram showing the main interactions between the mentioned elements is proposed in the next figure.

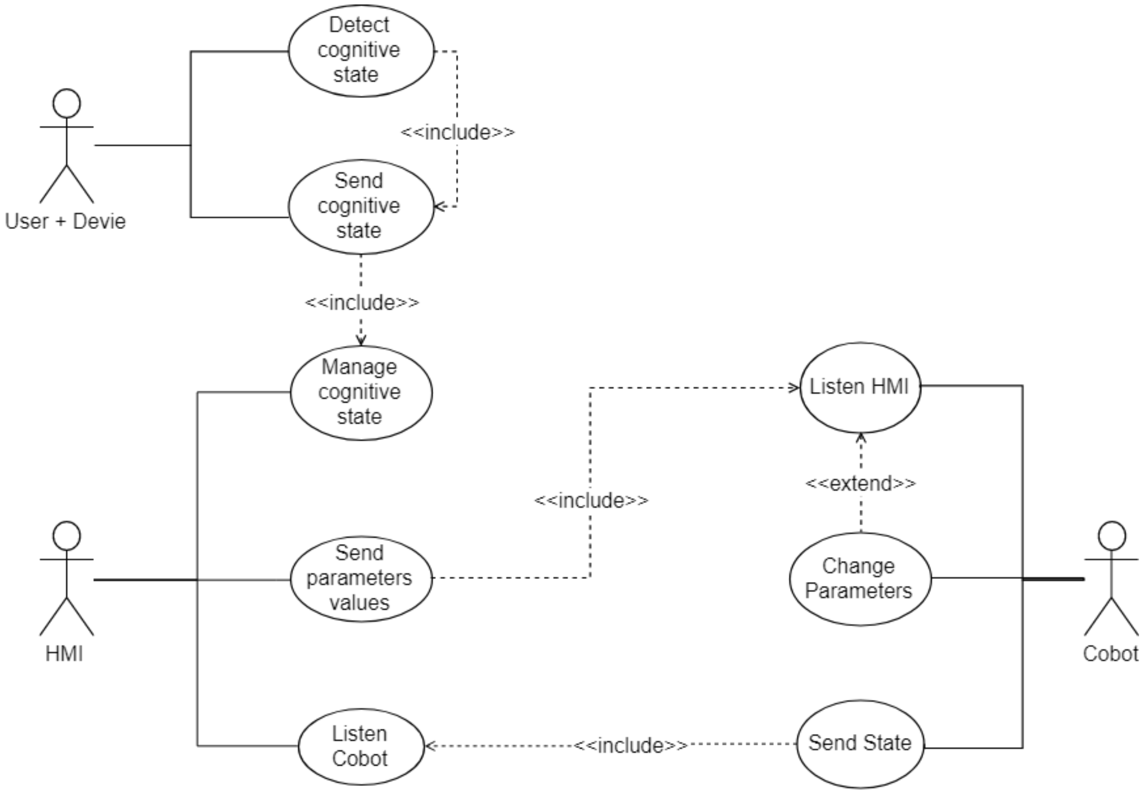


Figure 21. *Proposal Use Case Diagram*

3.2 Scenario

Following the defined use case, this section provides the selection of elements that will be used to implement the explained proposal. In order to select properly the components for the implementation, a representative scenario has been developed. This scenario is comprised of different parts: task, actors, device and HMI.

- Task

  This is the task through which the actors will interact in order to solve it. The task itself should be a representative work that in a simplified way encapsulates a great number of situations that can be observed in the industry and even out of it. This work must be collaborative, which means that it could not be performed in a proper way just by one of the actors, it necessarily needs both of them to be solved.

s

- Actors

  There are two main actors in this scenery, the user and the cobot.

  - User: The user represents the employee that in a hypothetical case will be working with a cobot in collaborative work. As a human being, the emotional state of the worker could fluctuate while working on the task with the cobot or by other external reasons. These emotions will be generated automatically by the user and they will be sent to the cobot.

  - Cobot: Is the collaborative robot in charge to assist the user to perform the specific task mentioned before. As a cobot, it has to have specific characteristics to work in an optimal way without being seen as an unknown agent for the user generating distrust. In addition, it will be in charge to receive the emotional data generated by the user and perform the proper actions as a response.

- Device

  Is the component that measures the cognitive state of the user and sends the extracted information to the HMI. This device will be based on one of the emotion detection systems explained in the previous chapter.

- HMI

  The HMI is going to be the element that connects and allows the interaction between user and cobot. Acting as a bridge, it will use as an input the emotions provided by the device, interpret the data and send the information that the cobot will need to adjust its actions to the user emotional state.

With these four components, task, actors, device and HMI, the scenario that composes the proposed implementation is defined. Each of these parts is indispensable to complete the objectives. If any of them fails the execution of the plan will not be performed correctly.

In the following parts of this chapter, are going to be selected and explained in a detailed way each of the elements that compose the scenario.

3.3 Collaborative task selection

Once the scenery has been described, it is necessary to find a proper representative task to solve. The selected task must be collaborative in order to fit perfectly in the scope of the thesis, that is why is essential to make a research of several works that certainly needs to be done by more than one agent and that in specifically one must be human.

3.3.1 Proposals and selection

Searching for different collaborative works in which it is needed the interaction of a human being, several options have been found:

- Packing and wrapping boxes

  The process will consist of wrapping the present within or outside a box and tie a ribbon on it. Both, the cobot and the worker will wrap the presents but only the human will be able to tie the ribbon around the present because it is a process similar to tying the laces of the shoes, process that neither machines or robots still able to perform in a proper way nowadays. This option was discarded due to the fact that the process is not representative enough, and the wrapping part can be perfectly handled only by a human or by a robot, so it is not collaborative enough.

- Car wrapping vinyl

  It is a task in which is necessary accuracy and certain artistic skills. This task was proposed because the artistic component it can be only performed by a human being and in terms of accuracy machines have already overcome humans. But there are several problems with this task such us, the workspace is too big for the cobot, and there are people that are able to make the wrapping by their own without a second hand. Because of those reasons this option was discarded too.

- Welding process

  Welding has always been considered a difficult activity that requires experience, precision and patience. Actually, there are robots in charge of welding in diverse industrial tasks, but mostly all of these tasks are repetitive, easy to program and too dangerous for humans. In the other side, there are other types of welding such as reparations works, artistic welding, architecture welding or tooling and die welding, that are mostly performed by humans due to its complexity of programming it into a robot, creativity or needed of adaptation. That is why this option was exposed. The cobot could be offering the welding material to the human at the specific location and the second one will use the tools to perform the joints properly. Even if this process could be seen as collaborative, it will require a cobot able to handle extremely high temperatures. In addition, testing this option would be harder because, as mentioned before, welding in a proper way requires tones of skills and experience in which are not in the field of the author of the thesis.

- Folding envelopes and stamp them

  Similar to the packing and wrapping presents process, this case has a part which could be done perfectly by a machine or by a human alone and a part that can only be accomplished adequately by a person. Folding an envelope is easy and does not mandatory requires more than one person to do it, even if two people could make it faster. Stamping a stamp is hard for a cobot because is not an action easily to predict. In each stamp, the wax spreads in a completely different way so the user has to adapt the pressure of the stamp in order to close properly the envelope. As this case does not fit the scope of being a completely collaborative task it has been refused.

- Assembly process

  There are several types of assembling processes, that it is why it could be possible to describe a real collaborative task by assembling an object. In order to fulfil the objective, the employee and the cobot must be working at the same time on the object, performing each of them a completely necessary task for the assembly that the one could not do it as well as the other.

  With this description, there are proposed two options.

  The first option is to assembly a little crane. The crane is composed of several elements (wheels, axes, bolts...), among them there are some ropes. The ropes have to be tied to the wheels in order to recreate the lifting and dropping movement of a crane. As it was said on the previous options, the act of tying a rope is difficult to execute by the cobot so it is something that only a person can do. In the process, the main role of the cobot would be holding the different parts of the crane in a specific position, allowing the user to introduce pieces into their holes, fasten the bolts and tie without problems of losing the position of the pieces. The role of the user is to fit the parts following the instructions.

Figure 22. *Crane*

  The second option consists of making the assembly of a small wooden box. The box is comprised of six tables that must be joined by using bolts. Similar to the crane's process, the cobot's role is to join the pieces fixing their position for the user to screw the bolts. This option was proposed by a colleague from the department where this master thesis was developed [86].

Figure 23. *Wooden Box*

> Both process fully complies the purpose of the collaborative work, both actors have a relevant role in the process. They are working at the same time and the task is really representative and can be extrapolated to other situations among the industry or outside it. So it seems suitable to select an assembly process as a collaborative task.

Selected the assembly process as the type of collaborative work that will be performed by the user and the cobot, it is necessary to select a cobot in order to know which of the presented assemblies would be the best option.

3.4 Cobot selection

As it has been mentioned before it is necessary to select a cobot which works side by side with the user in order to perform the collaborative task.

The selection of the cobot is linked with the task that is going to be performed. Since it has been chosen as a collaborative job an assembly process, the selected cobot must be able to assist in this type of process in an optimal way.

Assembly processes usually require multitasking, combining movements in a synchronous or asynchronous way, coordinate operations, precision and accuracy. The reason is that if the operation has to be repeated many times, the movements of the cobot must be the same ones each time the operation is performed. It can be performed a robots segregation depending on the number of arms in, single-arm or dual arm robots. By reviewing its application in assembly processes, it seems that dual-arm robots have a great number of advantages upon the single-arm robots [87]. Some of the advantages that have dual-arm robots over single-arm robots are linked to the abovementioned properties of the assembly process. Dual-arm robots can perform multiple tasks, for example, one arm can be holding a piece and the other one screwing; they can work with both arms simultaneously or separately, in an asynchronous or synchronous way; and moreover, a dual-arm robot can replace two single-arm robots [88]. As a consequence, in pursuance of selecting a cobot for this project, searching among the actual commercial cobots, it seems that the only dual-arm cobot fully developed and available nowadays is the ABB 14000, also know as *YuMi (Figure 3)*.

Being the brand new development of ABB robotics, *YuMi* simulates the upper body part of a human being, having a torso and two arms. Both arms are padded ensuring the safety of the users what allows it to adopt higher velocity in each movement. In addition, it has contact sensors which give it the capacity to stop its motion in case of a collision or a possible collision.*YuMi* is specifically made to handle tiny components in collaborative works, that is why it has a payload of 500 g. Its 7 axes per arm give it incredible adaptability and flexibility in order to solve different working conditions. Moreover, by using the teachpendant tool, it is easy to teach it movements driving its extremities defining a path point by point. This last feature helps to simplify and reduce the programming load.

All these characteristics make YuMi a safe and optimal option to work with the user to fulfil the same goal in a collaborative way. So as to achieve the aim of this work,*YuMi* will be influenced by the emotions of the user adjusting some of its parameters in order to perform the task in the best way for the user.

3.4.1 Collaborative task sequence

Once is known which it is going to be the cobot used in the project it is possible to create the sequence of steps to assembly both the crane and the wooden box. The sequence is going to take into account the user, the cobot and its interface. In the case of YuMi, as it is a cobot from ABB Robotics, the interface is the ABB’s teach pendant.

First of all, is going to be presented the components that compose the crane.

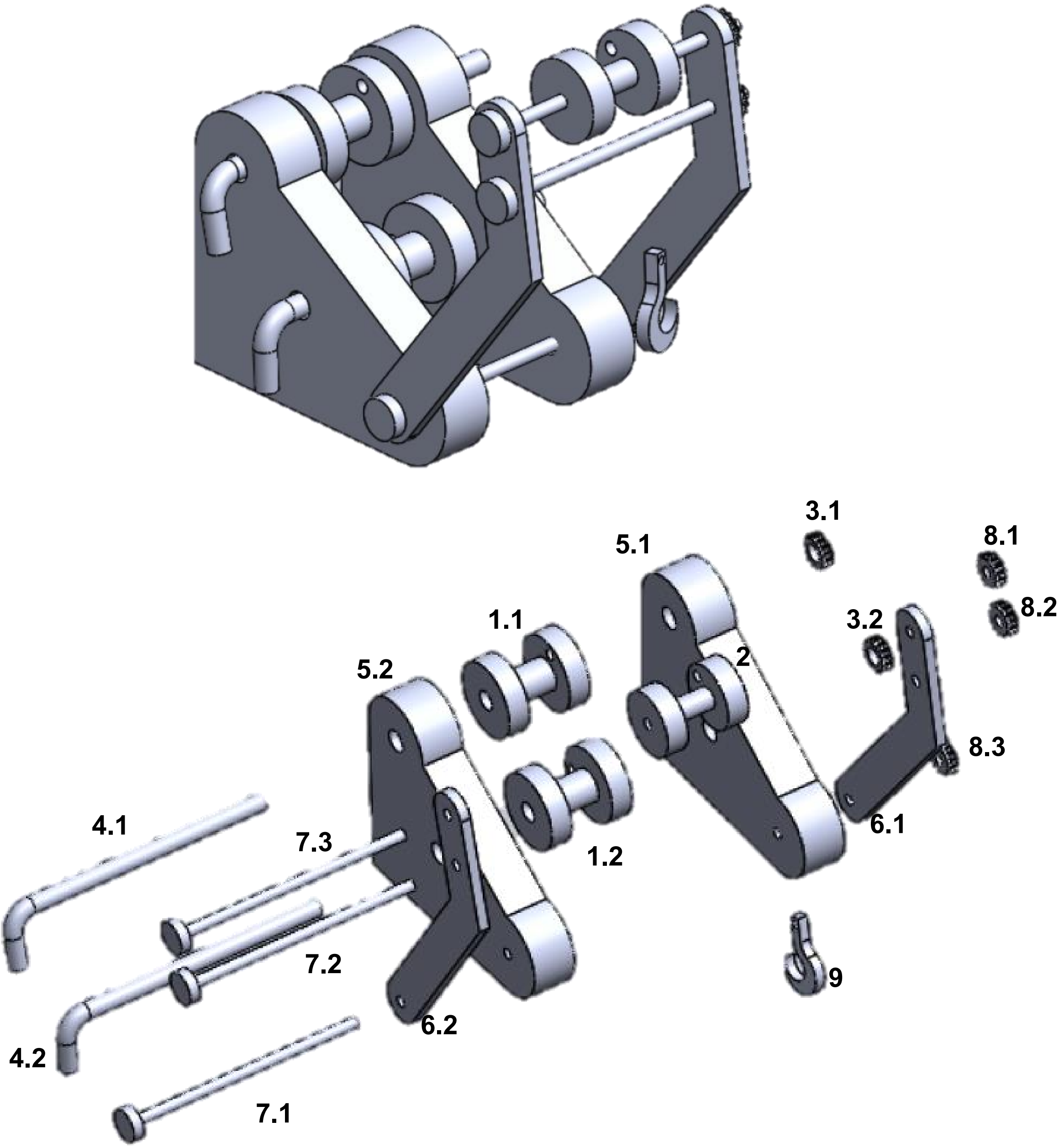
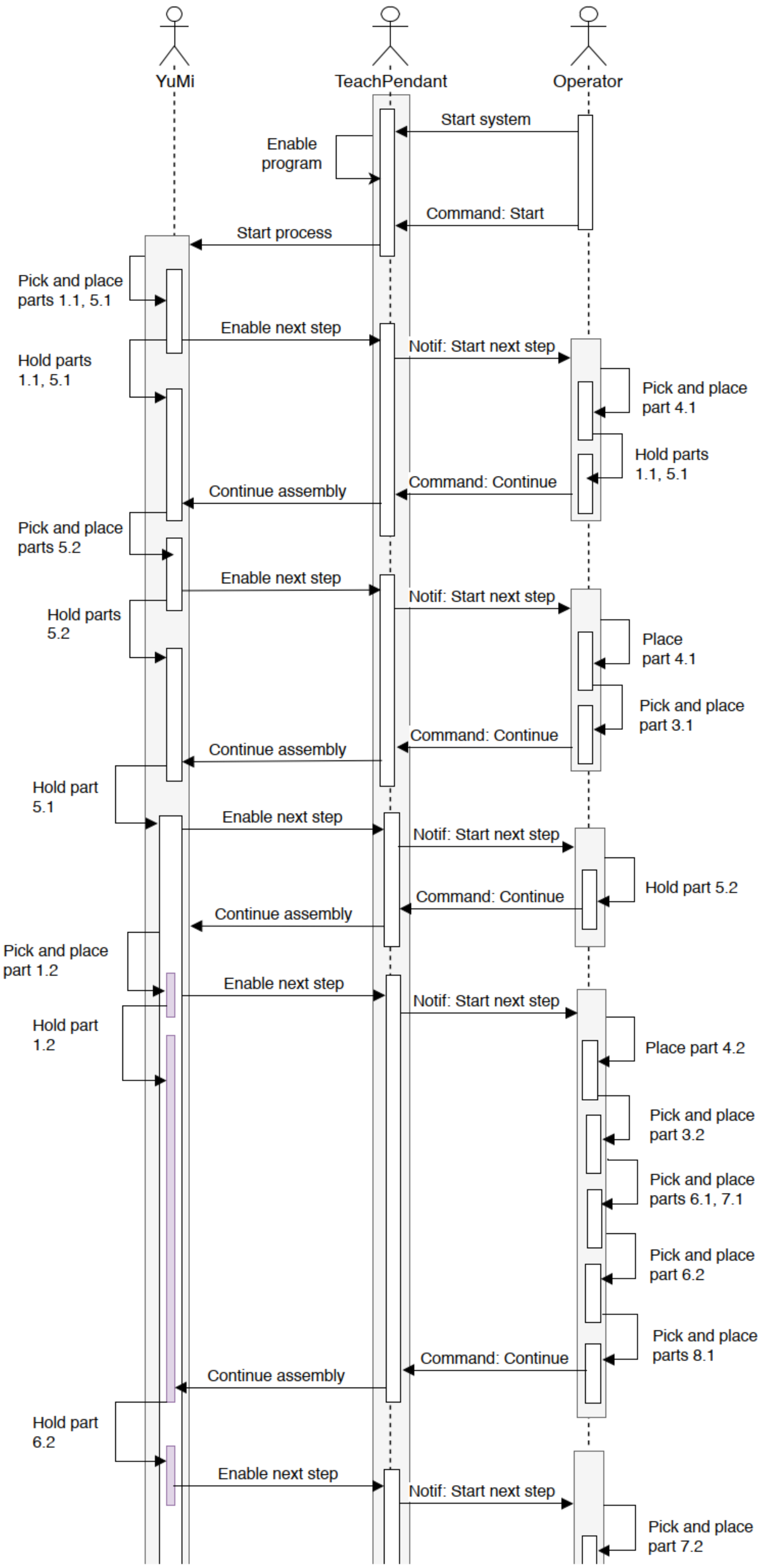


Figure 24. *Crane 3D model & Parts*

Next, it is going to be presented the sequence diagram of how it would be the assembly process of the crane with YuMi.


YuMi
TeachPendant
Operator
Start system
Enable
program
Command: Start
Start process
Pick and place
parts 1.1, 5.1
Enable next step
Notif: Start next step
Hold parts
1.1, 5.1
Pick and place
part 4.1
Hold parts
1.1, 5.1
Command: Continue
Continue assembly
Pick and place
parts 5.2
Enable next step
Notif: Start next step
Hold parts
5.2
Place
part 4.1
Pick and place
part 3.1
Command: Continue
Continue assembly
Hold part
5.1
Enable next step
Notif: Start next step
Hold part 5.2
Command: Continue
Continue assembly
Pick and place
part 1.2
Enable next step
Notif: Start next step
Hold part
1.2
Place part 4.2
Pick and place
part 3.2
Pick and place
parts 6.1, 7.1
Pick and place
part 6.2
Pick and place
parts 8.1
Command: Continue
Continue assembly
Hold part
6.2
Enable next step
Notif: Start next step
Pick and place
part 7.2

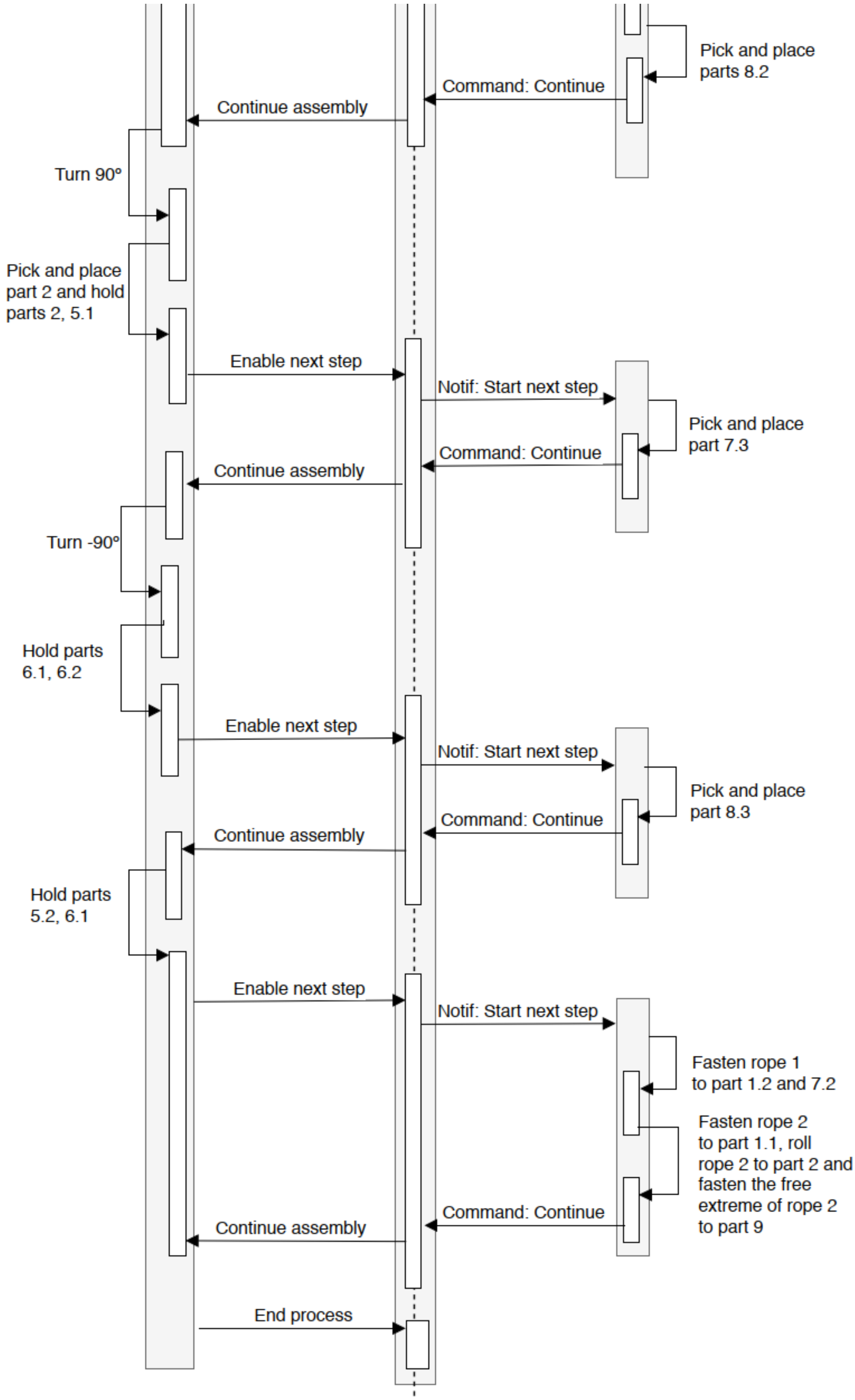


Figure 25. *Crane Assembly Sequence Diagram*

Secondly, a three dimensional model of the wooden box with its explosion of components can be seen in the next figure.

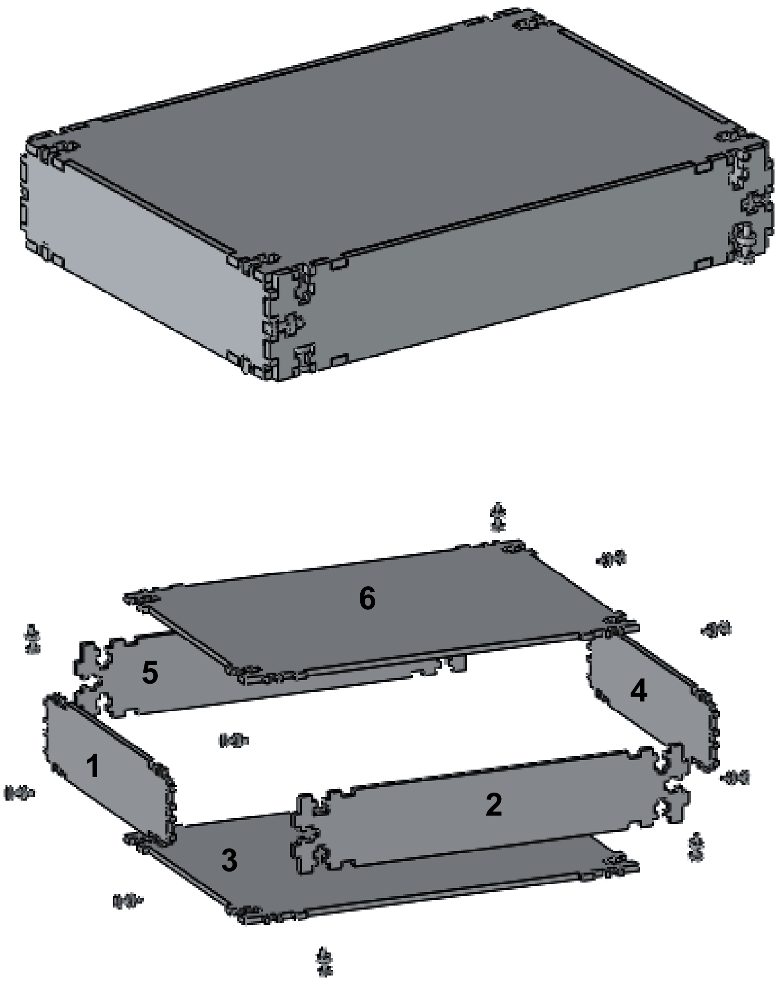


Figure 26. *Wooden Box 3D Model & Parts*

Following, is proposed a sequence of the required steps to assembly for the box.

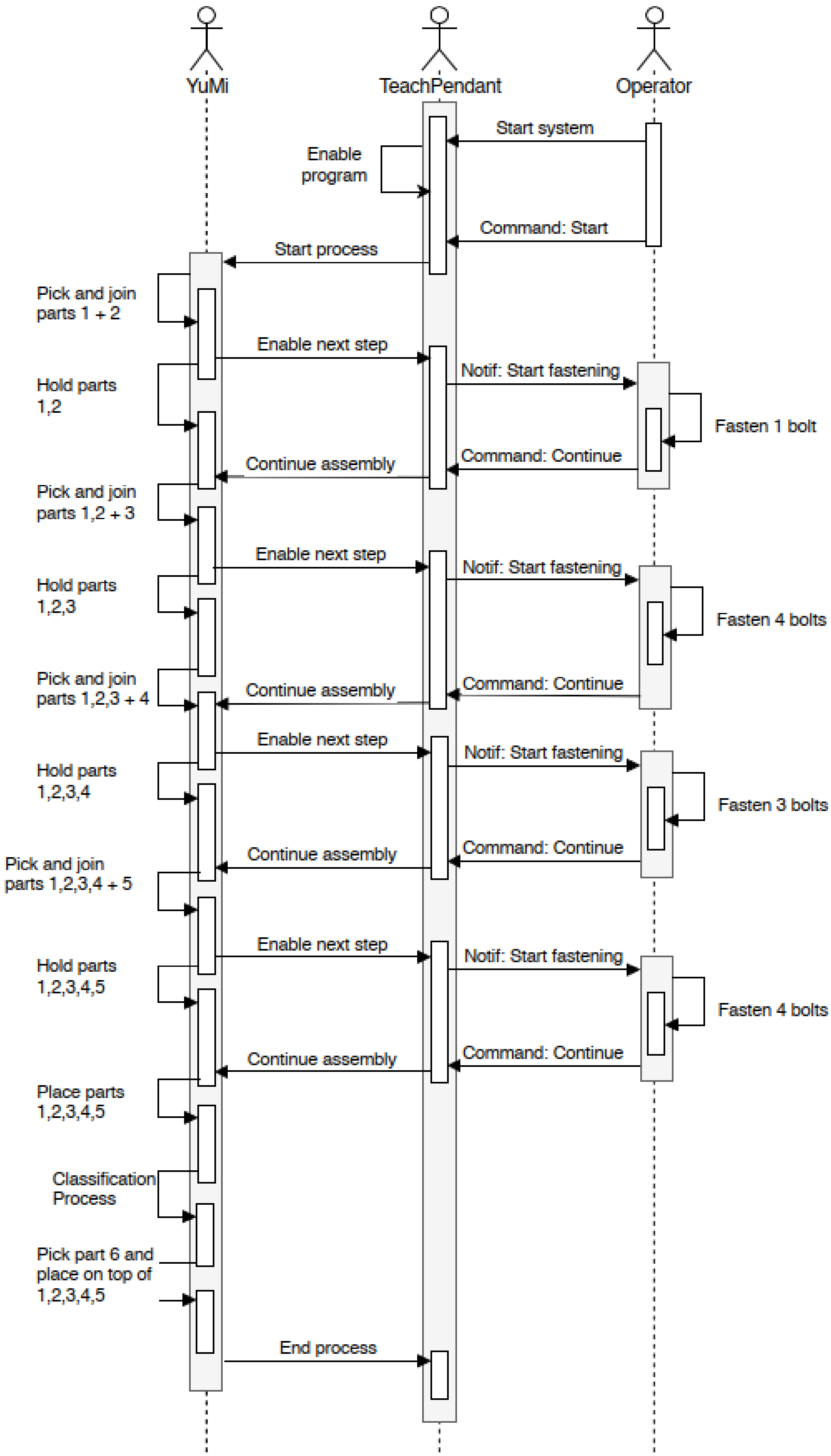


Figure 27. *Wooden Box Assemby Sequence Diagram*

Once both processes were designed, they were tested in order to decide which one would be better to use for the development of this thesis work.

During the preparation of the crane process, a problem came up. The crane was developed by 3D printing and the tolerance achieved by the printer were not accurate enough. While assembling the parts, there were always incidences, for example trying to fit some bars into their respective holes or with the screwing, being some pieces stuck in the process. So as to avoid this kind of incidents the crane should be developed with another material such as metal which will increase the cost of the process. Due to that reason and because of the reduction of the assembly complexity, the wooden box was selected as the best option to apply in this project.

The wooden box is composed of six tables that must be screwed in order to configure the appropriate shape. YuMi's role is the same as with the crane, holding the position of the tables and waiting for the worker to screw the bolts. In addition, this process has a better industrial scope because, after the assembly, YuMi must classify, using the camera placed in its right hand, different products. These products are mobile phones which are drawn in paper sheets, so depending on the model of the phone YuMi will place them in the assembled box or in another place.

In this way, the whole process is composed of two parts, the assembly and the classification. The first part is the one that is collaborative. The user and YuMi will assembly five of the six faces that compose the box. The sixth face is the top one, which will be placed once the classification is finished closing the box. As the top face is not going to be screwed, it is just necessary to fix the four bottom vertices in order to define the structure of the box. So as to define a point is necessary to define three coordinates, to fix a vertex is necessary to use three bolts. As a result, it will be necessary to screw a total of twelve bolts.

During the process, the duty of the user will be screw the bolts and supervise if YuMi is placing the sides correctly. In case that when YuMi is fixing its position the box’s faces are not properly allocated, is the user’s responsibility to place them correctly for the screwing. This part represents a task that the cobot is not able to do and only the human will know when is performed in a proper way. By the cobot’s side, its duty is to fix the position of the faces without losing its balances while the human is applying force screwing the bolts. That is the part that only the cobot is able to accomplish because a human will have a great reaction for each movement losing the position.

The simplicity of this process makes it a perfect representative of what it has to be a collaborative work. As well, inside the simpleness, there is a complex abstraction of all the diverse procedures that can be reflected in this example.

3.5 Emotion detection technique and device selection

This section is focused on the selection of a method and a device for detecting the emotional states of the user while working on the task.

3.5.1 Emotion detection technique selection

In the second chapter of this document diverse techniques to detect a human's emotional state have been introduced. A summary of their main characteristics was shown in *table 1.* It is necessary to find which is the best option among all the methods for the proposed scenery.

Regarding the use case proposed of an operator working side by side with a cobot, some of these approaches may not seem appropriate. In this particular situation, the worker will not be holding a conversation with the cobot. So as to detect his emotional state by speech recognition it is necessary a speaking interaction, this methodology is dismissed. Concerning body language, besides it is something that can be useful in order to know which employee is tired or who still with energy, in this context it is not the most accurate because the working positions will be always quite similar, even it is possible to perform the collaborative activity sit. These facts reduce the possibilities of detecting specific postures related to emotions. The case of facial recognition is similar to the one of body language. Both cases could determine in some situations emotions that the operator is experimenting but at the same time, a factory line is not a too emotional demanding atmosphere. Therefore, the facial expression of an employee could be considered as neutral even though he could be stressed or tired. In addition, both features can be controlled by the employees so it might be quite difficult to know when the expressions or gestures are forced or not. As body language and facial expressions have their similarities, ECG and EEG have them too.

Both are approaches difficult to control voluntarily by the user, they act in a subconscious way and they do not need a great emotional interaction to be detected. Hence, both of them could be valid strategies for the proposed case. In order to know which one is the best option for this specific case, it is necessary to explain how emotions are classified.

Once known the PAD approach, it is possible to determinate which system suits better for the use case, ECG or EEG.

Amjad M.R. et al., published in 2016 a research experiment comparing signals provided from both sources ECG and EEG [66]. The conclusion of the investigation was that ECG signals are strongly connected with the arousal dimension, but they do not represent adequately the behaviour of the emotions in the valence dimension. On the other hand, EEG achieved better results detecting mental states in both factors, arousal and valence. Therefore, as in the proposed use case the experimented emotions will be associated with more than one factor of the PAD method, EEG will perform a better service than ECG.

Taking into consideration the number of researches, investigations and conclusion that has the EEG approach by its side, and the negatives aspects mentioned of the other possibilities, it seems adequate to choose an EEG device to determine the emotions of an employee while working with a cobot in order to perform a collaborative task

3.5.2 Emotion detection device selection

Chosen EEG as the detection method, it is necessary to select among all the devices that are able to detect the electrical activity of the brain by the use of this technology.

In the second chapter of the document, there was advanced various type of caps and headsets that are able to fulfil that objective. In *table 2*, presented in chapter 2, there is a summary of the mentioned devices and their most relevant properties for the proposed scenery.

The most relevant characteristics to select a headset will be the quality of the data, the ease of creating an application and the detection of mental states. By analysing the captured data on *table 2*, it is possible to extract the conclusion that the headsets provided by *Emotiv* will be the best option to achieve the proposed objectives. The reason behind this selection is that these headsets have a different range of channels, which will ensure accurate brain data, is possible to develop an application using different types of programming languages and two of the headsets provide different mental states.

Having said that, between the three headsets (*Insight, Epoc+* and *Epoc Flex)*, the one that suits better in the previously mentioned features is the *Epoc+*. With 14 channels, detection of several emotions, software that allows the creation of applications in several programming languages and an affordable price of $799, is the headset that better fits in the scope of the thesis.

Further information that supports this selection is that *Epoc+* offers optimal EEG signals which have been tested and used in several studies proving its quality [60], [61]. In the second place, the Emotiv headsets are able to provide six different emotions on-live mode, being this one of the main objectives of this thesis. This detection reduces the computational problematics related to the detection of emotions by using the brainwaves data. In the third place, there has been carried on various successful studies and researches with Emotiv headsets related with EEG signal quality, emotion detection and even mind commands, that give them credit to their performance [20], [60], [89], [90]. Finally, there are some communities of developers that are actively creating applications that use these headsets by utilising the Emotiv Application Programming Interface (API).

3.6 Human-Machine Interface

In order to achieve the goals defined in the previous chapters is necessary to develop an HMI application that connects the user emotions with YuMi's actions so they can be adapted properly. The HMI systems that will be developed is going to be composed of three different parts:

- Device: Mentioned at the beginning of this chapter, this is the component in charge of detecting the brainwaves and the emotional state of the user. Previously selected, the device is going to be the *Epoc+* headset. Via a USB connector, the headset is able to connect to a laptop and send it all the data that is reading from the user's brain. Using the software provided by *Emotiv* is possible to access to the emotional state data, named as "performance metrics", that the user is feeling on-live.

- *ToEymotions Application*: While the headset is sending data to the laptop, it is necessary to develop a code which receives this data and manage it in a proper way. The code should make calculations and classify the data to then send which parameters of YuMi have to be modified to adapt to the user's emotions. *Emotiv* has developed an API which serves as a helpful tool to create applications using their headsets. This API allows having access to mostly all the data that can be shown in the software of the company such us stream data, face expressions, head motion movements, mental commands, etc. The API has been developed in three programming languages, C++, C# and NodeJS. Because of the amount of information that was available about applications using *Emotiv*'s headsets made in C++ and the advantages that give programming using C++ with *Qt* [60], the selected language for programming was C++.

- YuMi (RobotStudio): The last part of the HMI is the cobot. YuMi must receive the information that the C++ application will send, and modify the pertinent parameters. In order to adjust YuMi's behaviour is required to use *ABB*'s programmation software called *RobotStudio*. With this software is possible to write the code that later will be applied on YuMi creating the best possible behaviour.

Described the three components comprising the HMI application is fundamental to specify how they will be connected between them. There are two links, one between the device and the c++ application on the laptop, and another one between the c++ application and the cobot. The first link is made by the USB connector that allows reading from the laptop the generated brainwaves. And the second link will be performed by the use of TCP socket communication, acting the C++ application as the client and YuMi as the server. In order to see in a better way how the three parts interact with each other, a use case diagram has been developed.

4. IMPLEMENTATION

Along this section, it is going to be defined the implementation of the presented proposal. This implementation is divided into two main parts, the gathering of data and the creation of the application.

4.1 Gathering of data

The application that will be developed will need as inputs the emotional values of the user while working with the cobot. But in order to provide optimal emotional feedback to the cobot, first is necessary to assess the accuracy and veracity of the headset, and to extract relevant information about how the user emotions vary in light of a stimulus. So as to achieve these two objectives different tests have been carried out.

Before entering into the core of the experiments, it is relevant to mention that as commented previously, the selected headset to detect the emotions has been the *Emotiv's Epoc+*. This headset provides seven emotional values, that can be defined as following [78]:

- Interest: Grade of attraction or aversion to the experimented stimulus.
- Stress: Level of ease or comfort that is evoking the situation.
- Relaxation: Capacity to achieve a peaceful state of mind.
- Excitement: Instantaneous arousal degree that is creating a stimulus.
- Engagement: State of absorption in the performed action.
- Long-term Excitement: Similar to excitement metric but in longer periods of time, it usually measures minutes.
- Focus: Degree of fixed attention on a particular job avoiding disturbances.

All these metrics are scaled in the *EmotivPro* software [78] in a 0 to 100 scale, being 0 the lowest level of the metric and 100 the highest one. As is mentioned by the company, the algorithms that they use creates different scaled values for each individual user registered in their platform. This means that for example, an excitement value of 60 for a person that gets easily excited is not the same as a value of 60 for a person that remains calm in every situation. At the same time, the more times that is used the headset the more accurate will be the metrics because the system would have more data to learn how the same subject behaves.

As a consequence of this, the experiments and the training have been applied to one specific user in order to have a more accurate emotional profile of the subject. Having more precise data will allow creating better feedback for YuMi so it can adapt better to the subject situation.

In addition, in order to know if the headset is correctly placed or not there is a diagram that allows verifying if the sensors are correctly placed having good, regular or bad contact quality with the scalp.

Once knowing the definition of the emotions that are going to be read through the tests, and how the system will interpret the data, it would be easier to analyse what is experiencing the user and extract conclusions.

4.1.1 Valence-Arousal Test

Motivated by the Valence-Arousal methodology to classify emotions, the aim of this first experiment was to track the reactions of the subject while exposed to different types of videos. These videos were selected to evoke variations specifically in the *Interest*, *Excitement*, *Focus* and *Relaxation*. Following the descriptions of the metrics, it can be defined as a hypothesis that the *Interest* metric should represent the Valence and the *Excitement* the Arousal [91].

The first four videos were considered as "relaxation" videos, so following with the Valence-Arousal approach, they were trying to evoke low levels of *Excitement* and middle levels of *Interest*. In order to see if the values given by the headset matched the feelings of the user, the subject was asked to define his state on target metrics. The definition consisted of four classification levels: *low, mid-low, mid-high* and *high*. Being *low* not feeling the emotion at all and *high* feeling completely that mental state.

There were different conclusions extracted from this first group of videos. The first one was that the *Excitement* metric was quite accurate with the values defined by the user. In most of the videos, the metric was defined as *mid-low* and the values recorded by the headset were in a low range too. This could match with the Valence-Arousal approach as these videos where evoking a calm and relaxed feeling which are related to a low arousal level. Regarding the *Interest,* the subject addressed a *mid-low* feeling being the metric varying around 50 in this first group of videos. Regarding the *Focus* metric, it was addressed as *mid-low* and *mid-high* in different videos, being possible to see the variations on the metric through the videos but they never reached high values.

The second group of videos was selected to induce high arousal and valence responses in the user. The content of the videos was related to highlights moments of different sports. In comparison to the first group of videos, considered a “low-arousal” class, the results were as expected. The *Excitement* was addressed as *high* by the subject in the moments of each highlight and *mid-high* between them. That reaction was completely verified watching the metric arriving at values around 90 in the highlights and around 50 in the meantime. It was interesting too to see how the *Interest* was defined as *high* by the user during the videos and the metric itself is around 80 most of the time even if the *Excitement* was lower. About the *Focus*, it was pointed as *high* by the user too, and as the *Interest,* it remained in high values around 75 even if the *Excitement* was low. The *Relaxation* was defined as *mid-low* by the user, but opposite as with the first videos the values were higher.

The third and last group of videos was focused on increasing the grade of aversion feelings on the subject. Here there was a bit more confusion on the metrics. The first video was pointed as *mid-low* in *Excitement* terms by the user due to the fact that they were generating a great degree of sadness and compassion in the subject. The metrics record low levels of the metric. At the same time, it was addressed as *mid-low* interest due to the great grade of aversion, but the *Interest* metric remained in middle levels. The *Focus* was defined as *mid-low*, matching with the registered metrics. The second video generated a greater value of *Excitement*, addressed as *high* arousal by the user due to irritation, what is completely verified on the registered data, but the grade of *Interest* was pointed as *mid-low* due to the aversion and the metrics show a middle-high grade of *Interest* through the video. The *Focus* was defined as *mid-high* which fits with the obtained values. In both videos, the *Relaxation* was addressed as *low* but the values suffer plenty of variation.

One of the most important aspects of this last experiment was the *Stress* metric. In this last case, it worked perfectly and it addressed high values, what fits with the grade of aversion that the subject was suffering.

The extracted conclusion of this experiment was that the *Excitement* metric represents very well the arousal of the approach, but the *Interest* can be confused, so it is necessary more data in order to verify that it represents the emotional valence. Regarding other metrics, the *Relaxation* is not completely related to the relaxation at the moment itself so it could be confusing to using it. But as a method to know if the user is relaxed or not it is possible to use the *Stress* metric. Regarding the *Focus* and the *Engagement*, it is necessary to perform more experiments to know how to apply them and which are they optimal ranges in the subject.

4.1.2 Crane assembly Test

The aim of this experiment was to use see the subject's emotions evolution by performing an assembly task that could be perfectly applied in an industrial context. In fact, the employed element was the discarded collaboration task that has been mentioned previously in the document. In particular, it has been selected the assembly of the crane because as it represents an industrial component, the assembly is divided into steps. These steps are always the same, avoiding any change in the way of resolving the problem, tending to repetition, which could have an effect on the *Engagement, Focus* and *Stress* metrics.

In addition, the use of a process with similar characteristics to the one that has been selected to test the future application allows seeing how the user could react with a problem similar to the one that will have to solve with the cobot. This situation makes the emotional reactions more similar to the ones that the user will have working with the real task so the extracted conclusions will be more relevant.

The crane was joined four times in a row following the same steps trying to emulate the situation of an assembly line.

It was possible to see in the first test that the S*tress* level increased as the time is passing and the crane is being assembled. Regarding the *Engagement* metric, it tends to be at a mid-high level, around 65 which seems accurate to the subject perception and it fits with what is representing which is the level of absorption by performing a task. About the *Focus*, it is on a mid-low level but is increasing constantly through the experiment.

On the second test, the *Stress* metric reaches higher levels than in the first experiment. The *Engagement* as in the first test, vary in the range of 60-70, but in this case, tends to decrease a little bit through time. The *Focus* metric is a little bit higher than in the first experiment what it is normal as the *Stress* is higher too, so as there is more *Stress* in order to perform a task in a good way it is needed more *Focus*.

Regarding the third experiment, the values of the *Stress* at the beginning are lower than in the second experiment, in the middle of the test reaches similar values but lower ones. *Engagement* seems the same as in the previous tests. And the *Focus* increase in comparison to the other experiments.

It is in the fourth experiment when it is possible to see the drop on the metrics. *Stress* levels are lowered to levels between 75-56 when in the previous experiments where reaching values between 95-75. The *Engagement* values are lowered to, remaining to vary in a "constant" way, but now on the range of 50-40. That could be caused because as the process is more dominated the immersion on it is lowered. Finally, regarding the *Focus*, it is reaching moderate values than in the rest experiments too.

More experiments with a different task have to be performed in order to confirm the hypothesis of the more times the user is performing a task the emotions correlated with the high arousal and negative valence are lowering. Which in this case means that the *Stress* metric becomes lower.

It would be interesting to make more tests in order to see how many times a task should be performed by the subject in order to learn it in a real way or at least to achieve a certain level of dominance on it.

4.1.3 Box assembly alone

In order to give proper emotional feedback to the cobot, it still necessary to record more data about how the emotions of the subject are evolving through the experiments. In this case, the subject has been exposed to the challenge of make the assembly of the wooden box but instead of performing it with the cobot in a collaborative way, the task will be performed only by the user. This will provide a better knowledge of the user's emotions facing an industrial product. In addition by tracking its emotions and comparing them with the ones that will be recorded working with YuMi, it would be possible to see which is the difference between working alone or with the cobot.

The box is was assembled ten times, following the same steps and recording the emotions. Once all the assemblies had finished, it was calculated for all the metrics, the maximum, the minimum and the average. Calculating these statistical values makes easier to analyse the evolution of the subject's mental state through the experiments and if there is an actual process of learning the task that generates a reduction of the considered negative emotions as the stress.

By analysing the calculate parameters of the ten experiments is possible to extract the following impressions. It seems that there is a correlation in most of the experiments between *Stress*, *Excitement,* and *Interest*. Whenever there is an increase of one them there is a similar increase in the rest. This phenomenon between the first two metrics could be related by the fact that the *Excitement* represents the arousal, so if there is a stressful situation, which is an high-arousal emotion, it seems normal that the *Excitement* increases too.

In the other hand, there is possible to see an improvement in the *Stress* metrics as the number of experiments is increasing. Comparing the *Stress* levels in the firsts four experiments with the levels of the next five experiments, there is a decrease from the first one to the last one. This decrease can be noticed since the fifth experiment.

Similar to the *Stress*, *Engagement* suffers an average improvement in the last experiments. The increase of the metric is lower than the decrease of the *Stress*, but it still being a great new that the *Engagement* increases as the *Stress* gets lower, and the process is being dominated.

But opposite to the previous two, *Focus* metric is becoming lower through the experiments, what resembles the same aspect that happened in the crane experiment. The more dominated that is the task the less focus required to perform it.

4.1.4 Box assembly with the cobot

So as to finalize the extraction of emotional data from the user in order to apply it to the application, the last experiment consisted of assembly the box with the help of YuMi in a collaborative way. In the following picture, it can be seen the required elements that compose the scene for assemblying the box with the cobot.

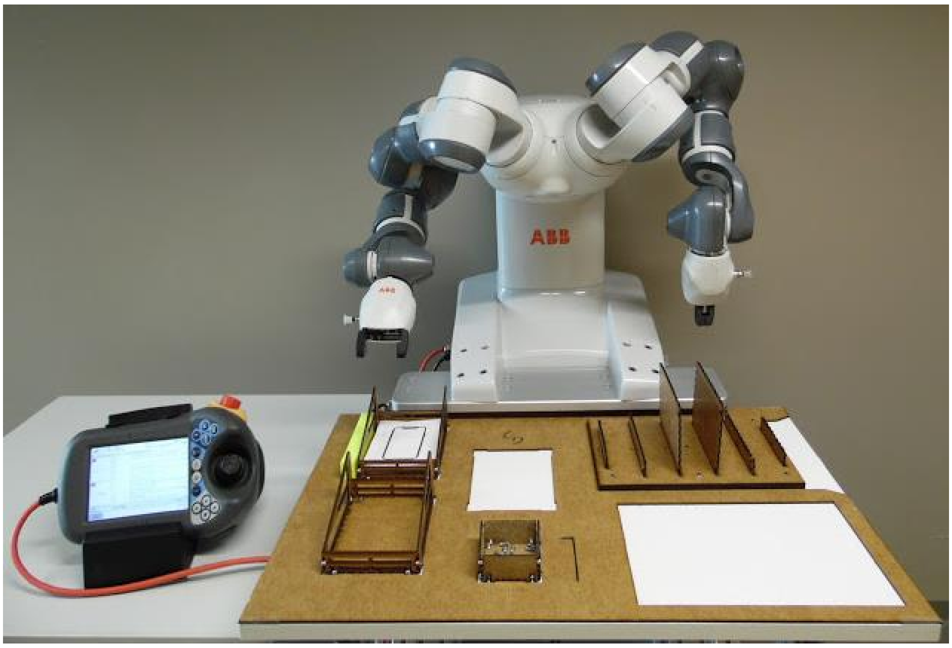


Figure 28. *Real Box Assembly Scenario*

In this case, the box has been assembled thirteen times in order to have at least the same amount of data as in the previous test making it easier to compare them, and having extra information. As in the previous case, it has been calculated the maximum, minimum and average values of each metric of the experiments.

Before comment the conclusions revealed from these experiments it is relevant to mention the following statement. In order to have a better analysis of the data, this experiment has been analysed in two different ways. The first one is calculating the mentioned statistical values taking into consideration the whole time of the assembly. In the other hand, the second way has been performed by just taking into consideration the working time of the user. This difference has been done in order to compare properly the assembly alone by the user and the assembly using a cobot. In the first case, the user is all the time working, not having idle times that release its stress. But in the second scenario, the user has idle times while is waiting for the cobot to pick, locate and hold the pieces for him to fasten them. That is why, in order to make a fair comparison of working time, it has been considered to eliminate the idle time to calculate the statistical parameters too.

Commented that, and by analysing the data is possible to obtain several conclusions.

Similar to the other experiments, here is possible to watch the same correlation between the metrics *Stress*, *Excitement,* and *Interest*. Considering the ideal time, the *Stress* levels have been reduced through the experiments, is a great difference between the fourth case and the second one (in the third case there was an error in the device and the data was not recorded). The same happens when without idle time, but the difference is even higher.

In *Engagement* terms, there is an increase through experiments in both cases, considering and without considering the idle time. And as in the assembly alone case, the *Focus* metric in both cases, idle and no idle time, decrease in a certain way as the process is becoming more dominated or learned.

4.1.5 Gathering data conclusion

In order to finish with relevant conclusions, it is necessary to compare the data between experiments. The following table presents the average values of the emotions considering the average values of the first ten experiments assembling the box alone and with YuMi.

Table 3. *Tests Average Emotion comparision*

Box Assembly	ENG	INT	REL	STR	FOC	EXC	LEXC
Alone	0.6401	0.6356	0.2919	0.5213	0.5247	0.4067	0.4079
with Yumi (considering idle time)	0.6653	0.6286	0.3263	0.4737	0.4998	0.4278	0.4301
with Yumi (just working time)	0.6637	0.6325	0.33513	0.4848	0.5111	0.43182753	0.4475

Starting by comparing the box assembly alone with the one with YuMi considering the idle time; it results that the average *Stress* of the subject is lower working with YuMi than alone. This can be related to the fact of sharing responsibilities with the cobot and not having to do everything the user. The *Engagement* metric is higher too in the case of working with a cobot, but the *Focus* is lower.

But as it had been mentioned before, in order to compare them in a more accurate way, it is necessary to extract the idle time of the equation. Using this case, the *Stress*, even being higher than in the case with the idle time, still being quite lower than to the assembly alone. Which confirms that for the subject is less stressful work with a cobot than alone. Is necessary to add that is normal that extracting the idle time the *Stress* increase because usually in the idle time there is time to relax. As a good point for the case without idle time, the *Focus* still lower than in the assembly alone but is higher than in the assembly alone. This fact continues following the cases that if a task is more difficult, the task requires more *Focus*.

The *Excitement* and the *Interest* metrics, in this cases of assemblies, seems to have a lower relevance than in other cases due to the fact that they seem related to the *Stress* that earns more weight as in this experiments it can be seen how as the user is relaxed the *Stress* decreased and when its less relaxed then it is increased.

Once all these data have been analyzed it seems that the most important metrics to take into consideration and with more accuracy in this field are the *Stress* as a principal metric and the *Engagement* and *Focus* as secondary metrics. At the same time, it is possible to extract the relevant ranges in which the subject emotions work. These ranges have been used to defining four classes, *low, mid-low, mid-high, high*.

4.2 HMI development

As it has been mentioned in previous chapters, the HMI is composed of three elements, the *Emotiv Epoc+* headset, which will detect the emotions, the *ToEymotinons Application* which is in charge of using the emotions provided by the headset, calculating the emotional feedback, and communicating with YuMi, and YuMi itself which will receive the emotional feedback and modify its parameters adjusting to the user's emotion needs.

The presented elements must interact with each other exchanging different type of information. Each element conforming the HMI has a different way of connection to share their data. The *Epoc+* uses Bluetooth to connect to the USB port which receives its signals. This USB is the one that allows the laptop to access to the data detected by the headset by using *Emotiv* software or API. Then, regarding the exchange of information between the *ToEymotions* application and YuMi, the selected option of communication must allow the transference of information in a not complicate and fast way and it must be applicable to any kind of robot or cobot interface, avoiding software of specific customers. A great alternative that provides these features, is the creation of a Transmission Control Protocol / Internet Protocol (TCP/IP) socket connection.

TCP/IP is one of the most widely applied protocols in networks, sharing the field with the Internet and Local Area Networks (LANs) [92]. One of its main objectives is ensuring the transference of information between the hosts, a client and a server [92]. Usually, this protocol is used when the client and server have different remote network systems. In order to fulfil the communication between both participants, client, and server, it must be defined a socket for each of them. A socket is an endpoint which is determined by a port number and an IP address. With the creation of a socket for the client, another one for the server and the definition of a listening port, a unique communication channel is created between them [93].

In this particular case, it has been decided to define the cobot as the server and *ToEymotions* as the client. The idea behind this is to represent that is the cobot the one that guides the interaction deciding when to have emotional feedback.

The following scheme shows the abovementioned different type of connections between the elements that compose the HMI.

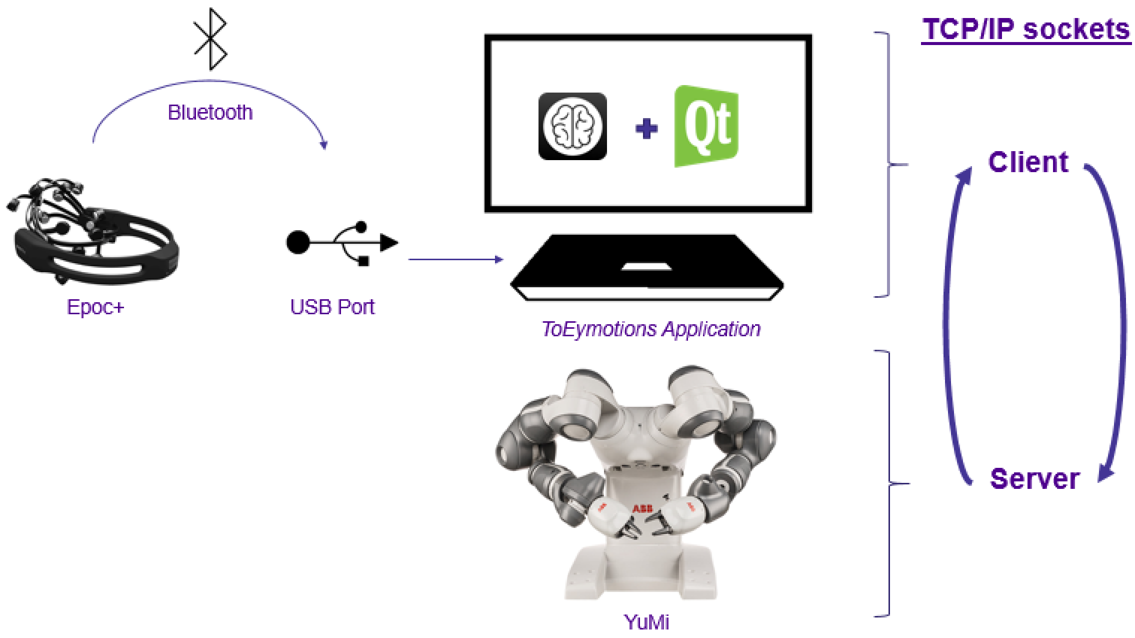


Figure 29. *Connections between HMI elements*

4.2.1 ToEymotions Application

ToEymotions is the name of the developed application in charge of managing the emotions detected by the headset and providing emotional feedback to the cobot. The application can be divided into three main blocks: one in charge of receiving the emotion values, classify them and preparing the feedback for the cobot; a second block responsible of maintaining the communication with the cobot; and a third block, developed in the cobot, focused on handling the communication with the second block and modifying the parameters that will allow YuMi to adapt its behaviour to the user feelings. The parameters that are going to be modified are the velocity of the cobot and the time that the user has to perform one step of the assembly.The first two blocks have been coded using multithreading. By using multithreading technique is possible to constantly listen to the server in case it needs to communicate and simultaneously receives the data coming from the headset handling the emotions.

Introduced how the application is structured, a continuation is going to be explained deeply how each of the blocks is designed.

- <u>Block 1: Emotional feedback</u>

  This block is the core of the application. It commands the reception of data coming from the headset, classifies the emotional values and decides the feedback that will be sent to the cobot.

  In order to obtain data from the headset, as the employed headset is the *Epoc+* it is necessary to use the Emotiv API, *Cortex*. This API allows capturing diverse data flow coming from the device and integrate them on applications. Using WebSockets and JavaScript Object Notation (JSON), *Cortex* makes possible to access to raw EEG data, user's facial expressions, mental commands, head motion parameters, and performance metrics [94]. In addition, there are several examples of how to use *Cortex* API correctly. Within the examples, there is a static library which provides the source code to use subscribe and use the streaming information from the headset. The static library is composed of the following codes:

  - *CortexClient*: Uses WebSockets to establish a connection with *Cortex* and defines mostly all of the methods linked to the *Cortex APIs*.
  - *HeadsetFinder*: Looks for headsets connected to the device were the application Is running and provides a list of them.
  - *Headset: Saves the characteristics that define a found headset.*
  - *SessionCreator*: Creates an Emotiv session linked with a headset.
  - *DataStreamExample*: Defines methods for making possible the reception of data streams.

  So as to subscribe to the emotion data stream, it is mandatory to have access to an Emotiv PRO license. Later, to run an example using this library, it is necessary to create a "cortex app" on their webpage [78], save the "client id" and "client secret" provided and modify the "SessionCreator" file setting those parameters plus the "EmotivID" and password. In addition, when subscribing to the data streams, the license code must be specified; this will give permission to get the emotional values.

  It is relevant to mark that the performance metrics or emotion data stream is received every ten seconds as a JSON array. Which means that every ten seconds there will be a new JSON array with the mental state information.

Regarding the developed part which receives the emotion data stream and prepares the emotional feedback for the cobot, it is structured as the next diagram shows.

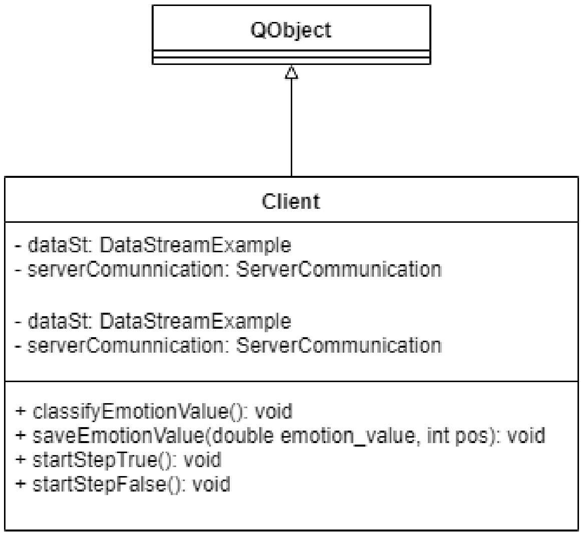


Figure 30. *Block 1 Class Diagram*

An explanation of which is the responsibility of the main element of the diagram is exposed next:

- *saveEmotionValue*: Saves the arriving emotional values from *DataStreamExample* composing an array of emotions. Each position of the array corresponds to a different emotion type. In addition, each time a new emotion value arrives, it checks the emotion type and adds the value with the previous value of the same emotion. This sum is saved in another array, creating an array of the added values

- *classifyEmotionValue*: Depending on the number of JSON arrays received, this method calculates the average value of all the emotion types. Taking into consideration *Stress* as the main mental state and *Engagement* and *Focus* as additional ones, it checks if their average values are in one of the following five levels or sublevels. If they are, the velocity parameter is set.

1. Stress is LOW → VEL = HIGH
    1. IF: Engagement *MID-LOW* & Focus *MID-LOW* → VEL = MID

2. Stress is MID-LOW → VEL = HIGH
    1. IF: Engagement *LOW* & Focus *MID-LOW* → VEL = MID

3. Stress is MID-HIGH → VEL = MID
    1. IF: Engagement *MID-HIGH* & Focus *MID-HIGH* → VEL = HIGH
    2. IF: Engagement *HIGH* & Focus *MID-LOW* → VEL = HIGH
    3. IF: Engagement *LOW* & Focus *LOW* → VEL = MID

4. Stress is HIGH → VEL = SLOW
    1. IF: Engagement *MID-HIGH* & Focus *MID-LOW* → VEL = MID
    2. IF: Engagement *HIGH* & Focus *LOW* → VEL = MID

5. Stress is SUPER-HIGH → VEL = SLOW
    1. IF: Engagement *HIGH* & Focus *HIGH* → VEL = MID

Once the velocity is set, the delay parameter is defined as follows.

IF VEL = HIGH → DELAY = LOW

IF VEL = MID → DELAY = MID

IF VEL = SLOW → DELAY = HIGH

- *startStepTrue*: Points that the cobot has started a step of the assembly, as a result, the calculation of the emotional feedback must start.

- *startStepFalse*: Points that the cobot has finished a step of the assembly which means that the emotional feedback must finish. In addition, it resets the arrays that saved emotional values.

- Block 2: Communication with the cobot

The aim of this block is to communicate with the cobot, which is acting as the server, listening to its messages and sending information about the emotions when is needed. This section is comprised of the elements exposed in the next diagram.

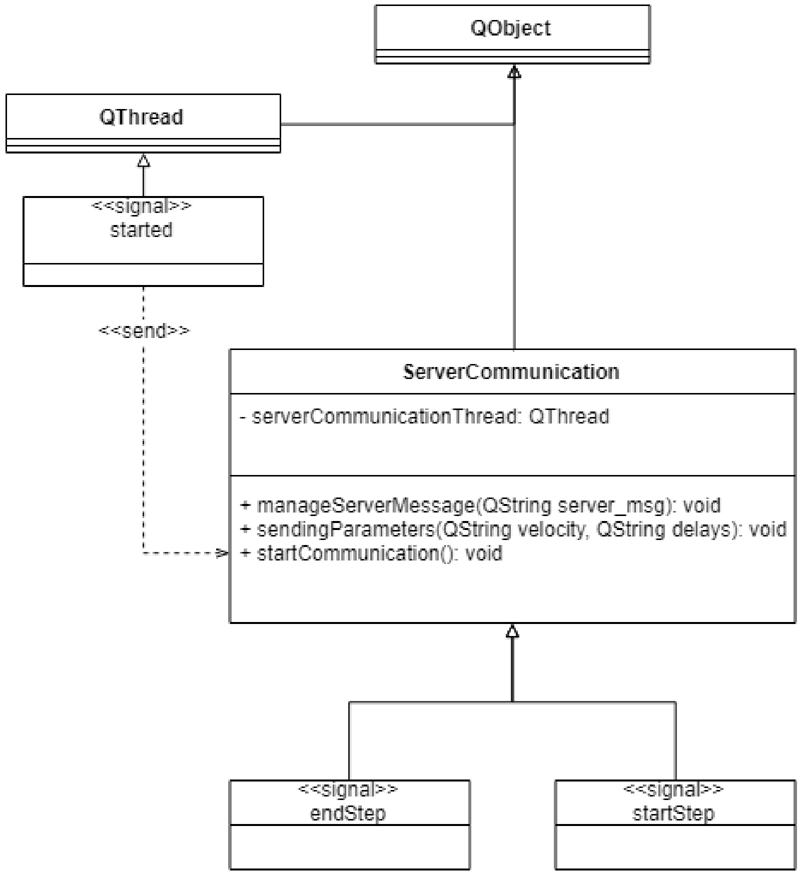


Figure 31. *Block 2 Class Diagram*

- *startCommunication*: This method is in charge of creating the client socket and maintaining the communication loop with the cobot. The code has been adapted from [95]. It verifies if the cobot has sent a message and if it does, it calls *manageServerMessage*.

- *manageServerMessage*: Receives as a parameter the message of the cobot, and depending on its content it emits *startStep* signal, to inform that a step of the assembly has already started, or *endStep* signal, which informs that the step is finished so it has to finish the emotional feedback too.

- *sendingParameters*: Receives as parameters the values of velocity and the delay coming from block 1 and it sends them to the cobot.

- <u>Block 3: Modification of cobot's parameters</u>

Finally, the third block is in charge of handling the exchange of messages with the block 2 and modify the velocity and delay when a received message indicates it. The code of this block has been created in *RobotStudio*. The script, apart of having the sections related to the sockets and the emotional feedback which have been developed for this Master thesis, it has the parts related with the required movements that YuMi has to perform in order to assemble the wooden box. The programming of these movements has been performed by [96].

In order to generate the server socket that allows the interaction with block 2 properly, first is necessary to connect to the cobot's controller. The connection has been performed connecting a router to the WAN port of the cobot and the laptop to the router. The connection has been done in this way because as the cobot is acting as a server, in order to bind the created socket to an IP address and a port number, *RobotStudio* only allows using any public Local Area Network(LAN) address or the controller service port address [50].

So as to keep the process of emotional feedback on the loop instead of the programming of the cobot movements for the assembly, is presented a class diagram with the methods that are integrated into the emotional process.

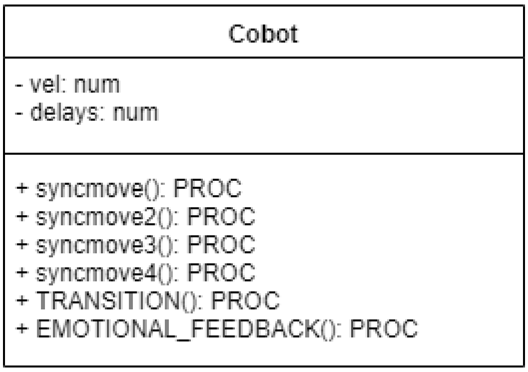


Figure 32. *Block 3 Class Diagram*

- *syncmoveX*: These functions allow YuMi to perform synchronous movements with both. Once the arms arrive at the positions where it should remain holding the pieces until the user ends the fastening, two things are done: a message is sent to the block 2 informing that a step has started and the *TRANSITION* method is called.

- *TRANSITION:* Makes the cobot wait in the "fastening" position for a certain *"delay"* time. This time delays the start of the next step and it gets its value depending on the emotional feedback. It represents the time that the user has to screw the bolts. When the *delays* time has finished, it means that the step has ended and it calls EMOTIONAL_FEEDBACK.

- *EMOTIONAL_FEEDBACK:* Adapts the characteristics of the cobot to the user's conditions. It sends a message to block 2 to announce that the step has finished so it can receive the feedback. Depending on the reply message received the *velocity* and *delay* are adapted to a value or another.

The application just takes into account the emotional values of the user while performing the task. The reason is that it was observed while gathering data from the test, that it is at that moment when the user usually suffers the relevant variations in his emotional state. During the transition between steps, as the user does not has to conduct any action related to the assembly, the emotions tend to smooth their values, becoming less significant.

Having been explained the functionality of the *ToEymotion* application by blocks, it is possible to show how the three blocks are linked and how they are interacting between them in order to fulfil its purpose which is make the cobot adapt to the user emotional state while performing a collaborative task making the user feel more comfortable and trust more in the cobot. The relationship between blocks and how the emotional feedback will be introduced in the collaborative process, can be seen in the following diagrams.

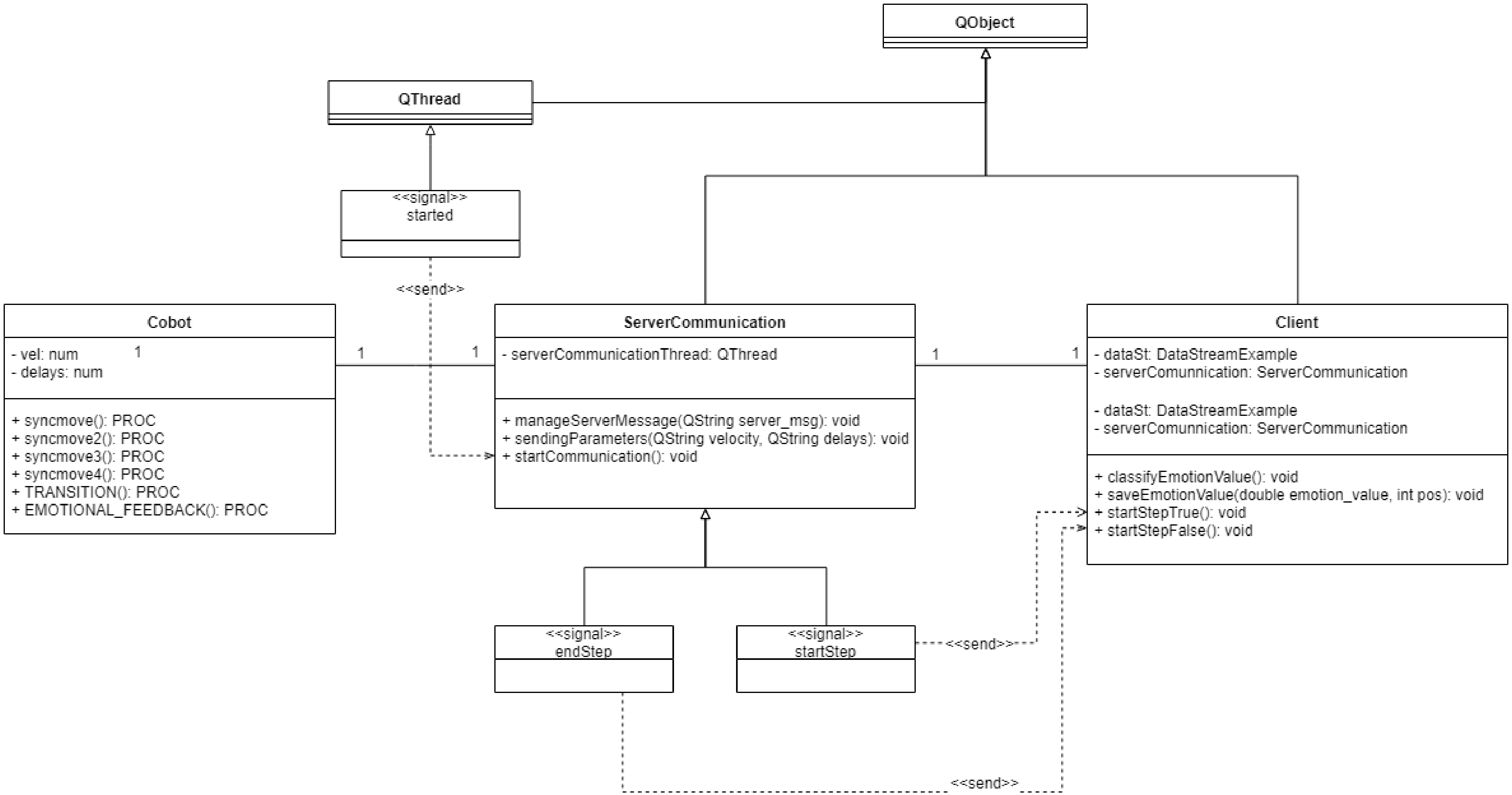


Figure 33. *ToEymotions Class Diagram*

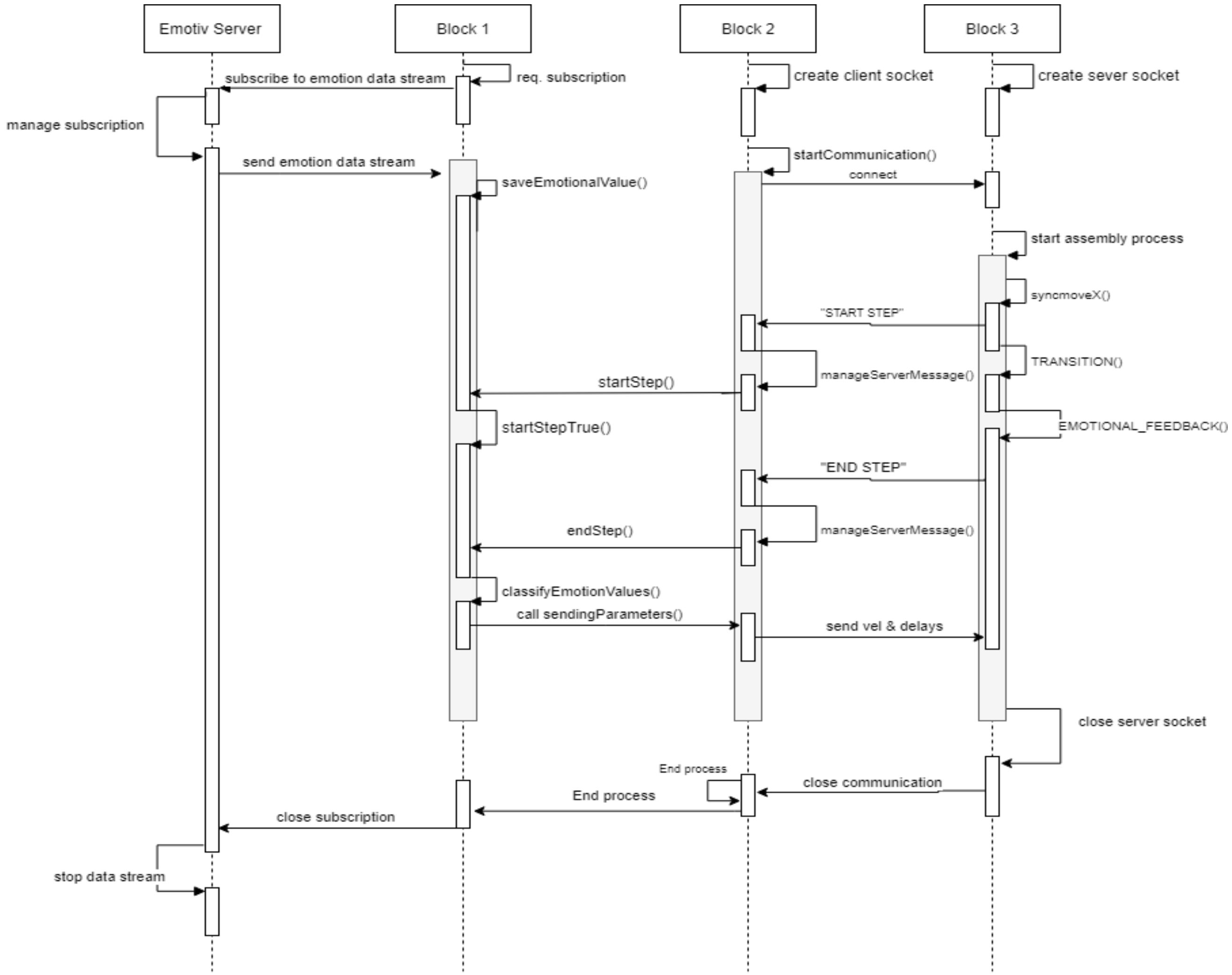


Figure 34. *ToEymotions Sequence Diagram*

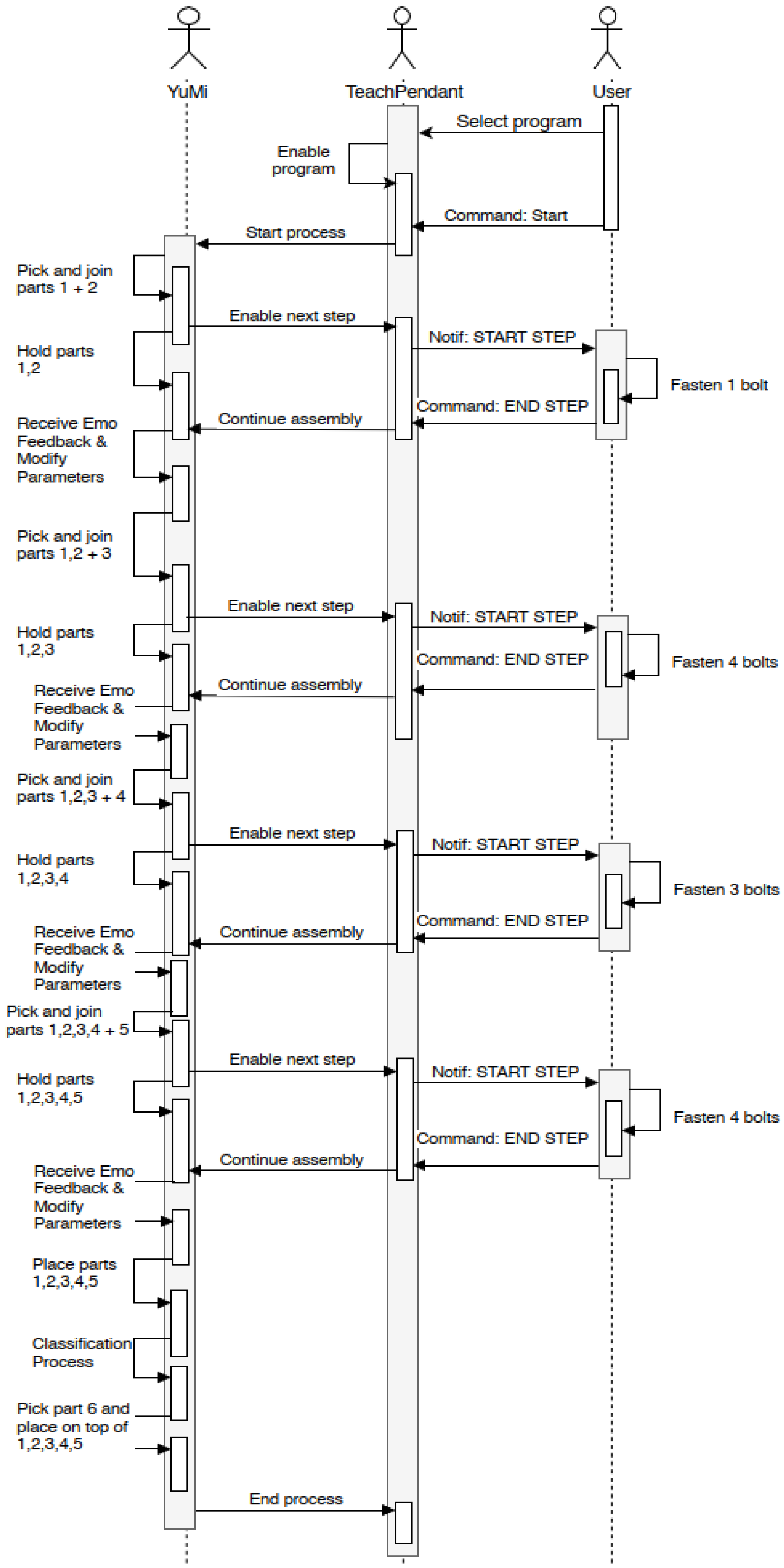


Figure 35. *Box Assembly with Emotional Feedback Sequence Diagram*

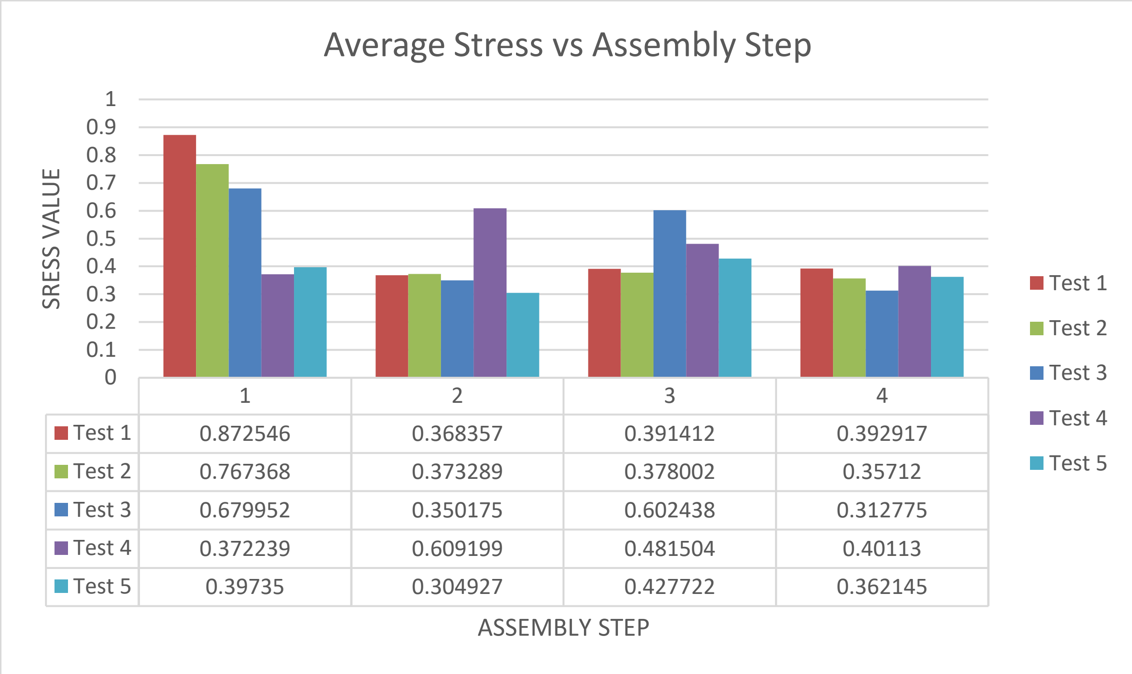
5. CONCLUSIONS AND RESULTS

The aim of the development of this thesis work is to propose and implement a system that allows a cobot to adapt its behaviour to a user by using his emotions affecting positively to his trust and comfort with the interaction. In order to fulfil this goal it has been proposed to generate a scenario in which is employed an EEG headset to measure the user's mental state, and the HRI is performed by conducting an assembly collaborative task. Finally, it has been elaborated the *ToEymotions* application which manages the emotional data and provides it as feedback to the cobot changing its performance.

So as to demonstrate that using the mental state of a user to influence the cobot's conduct is an optimal way of influencing beneficially the HRI, the developed system has been tested following the same patterns of the experiments made for gathering data.

Figure 36. *Human-Cobot Interaction with EEG headset*

By assembling the wooden box using the *ToEymotions* application, it is possible to see how the modification of YuMi's velocity and delay to fasten the bolts affects the stress of the user. When it is detected a high average value of stress during a step of the assembly, and the classification algorithm defines that YuMi's velocity should decrease and the delay increase, in the next step, the stress is lowered.



	1	2	3	4
Test 1	0.872546	0.368357	0.391412	0.392917
Test 2	0.767368	0.373289	0.378002	0.35712
Test 3	0.679952	0.350175	0.602438	0.312775
Test 4	0.372239	0.609199	0.481504	0.40113
Test 5	0.39735	0.304927	0.427722	0.362145

Figure 37. *Stress Evolution Through Tests*

In addition, the stress value tends to be reduced through the assembly, which means that the intervention on YuMi lowers the stress values as the interaction advance. Moreover, comparing the different test, it can be appreciated that in the same step, the stress values are decreasing. The first times that the user is exposed to the assemble challenge is more stressed, but as YuMi modifies its parameters the user is having a better emotional reaction, in stress terms, to the same step of the process.

As is exposed in [37], employees' emotional states have a straight influence in their capacities to interact correctly with a robot. Moreover, the effect of mental states such as stress or fatigue can degrade trust [37]. Just as is proposed in [8] the solution for the previous situation might be in generating adaptative systems which consider user's capabilities.

Taking into consideration these facts, and the proposed question that motivates the development of this thesis work: how to positively adjust the interaction between a human and a cobot? *ToEymotions* application offers a solution for the by tracking the emotions, considered as unpredictable [37], and adjusting the cobot's parameters, which results into a general decrease of the user stress values through the interaction, which seems related to an increase in trust [8], [37]. An improvement in trust in HRI can be linked with an increase in productivity and in the product quality [85].

During the development of the thesis, regarding the gathering of data or the implementation, there were found some limitations that allowed only to make the work for one user. The first limitation was that the functionalities offered by *Emotiv* are adjusted to each user. Each time the headset is used, the data is saved creating a personal cognitive profile. Thus, it requires time using the headset to have values accurate to the user emotional state. The second limitation is related to the tests. In order to adjust the cobot to the user mental state, the ranges that define the emotional classification should be the most likely related to the user. Therefore, is necessary to make individually test with all the users that will work with the cobot. So as to ensure first that the application works properly and because of time-related issues, the application has been developed for one user. However, it would have been great to have data from more users for testing the application.

Finally, it has been used for the implementation of the project a commercial headset which shares emotional data, but it could be used any device that provides mental states.

6. FUTURE WORK

First of all, this thesis work is a researching project that generates the first steps for the development of a new methodology to improve the HRI. Adjusting a cobot's behaviour to the human needs by using mental states, is an idea that has never been applied before. As a consequence, there could be more possible patterns to resolve the objectives proposed apart from the one exposed in this document.

Since this project is comforming a new path on the way of seen HRI, there are still a great number of ideas and possibilities that can serve as complementaries tools generating synergies.

An example of synergy could be the combination of virtual reality (VR) as a training tool for employees [97] and an EEG headset in order to extract the necessary information to create better profiles of the employees that will work with the cobots. The better emotional profile the better cobots will respond to human needs.

Another case could be attaching a camera to the cobot so it can recognise the user that is going to work with it. By recognising the user using face recognition, the cobot could access to all the data that can affect the interaction between both actors and upload its characteristics beforehand.

This application could find an application reconfiguring how employees are distributed in the working place. By tracking their cognitive state and knowing their real-time adaptations of the cobots, if the performance of the interaction is decreasing, it could be suggested a change in the workplace between employees to another one that allows them to increase their performance.

Moreover, as it has been mentioned in the previous chapter, the project has been developed using a commercial headset. However, it is possible to create a headset which adapts better to the conditions that the user is exposed to. There are several communities, such as *OpenEEG* [77] that are generating instructions for developing and generate EEG devices. After producing the headset, it will be necessary to apply any of the methodologies for detecting emotions mentioned in the second chapter of the project to know the emotions of the user.

Regarding the work exposed in this document, in order to continue researching this field, there is some further work that can be done. In the first place, it will be necessary to test *ToEymotions* with more users to collect more data generating a greater analysis of how a big group of users reacts to the process. In the second place, it would be great to improve the correlation between emotions improving the algorithms. In the third place, it could be interesting to find out other variables that can affect the HRI to take them in consideration. Finally, create a Graphical User Interface (GUI) could be helpful for the user to handle the application in an easier way.

REFERENCES

[1] N. Tbn, “SMART MANUFACTURING FOR THE FUTURE INDUSTRIE 4.0 Future Markets.”

[2] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, Aug. 2014.

[3] K. Zhou, Taigang Liu, and Lifeng Zhou, “Industry 4.0: Towards future industrial opportunities and challenges,” in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 2147–2152.

[4] K. Chen and D. C. Yen, “Improving the quality of online presence through interactivity,” *Inf. Manage.*, vol. 42, no. 1, pp. 217–226, Dec. 2004.

[5] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, “Finding High-quality Content in Social Media,” in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, New York, NY, USA, 2008, pp. 183–194.

[6] E. Agichtein, E. Brill, S. Dumais, and R. Ragno, “Learning User Interaction Models for Predicting Web Search Result Preferences,” in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2006, pp. 3–10.

[7] M. Gil, M. Albert, J. Fons, and V. Pelechano, “Designing human-in-the-loop autonomous Cyber-Physical Systems,” *Int. J. Hum.-Comput. Stud.*, vol. 130, pp. 21–39, Oct. 2019.

[8] V. Villani, J. Czerniak, L. Sabattini, A. Mertens, and C. Fantuzzi, “Measurement and classification of human characteristics and capabilities during interaction tasks,” *Paladyn J. Behav. Robot.*, vol. 10, pp. 182–192, Jan. 2019.

[9] L. Uden, “Human–Computer Interaction and Learning,” in *Encyclopedia of the Sciences of Learning*, N. M. Seel, Ed. Boston, MA: Springer US, 2012, pp. 1464–1469.

[10] M. Edwards, “Robots in industry: An overview,” *Appl. Ergon.*, vol. 15, no. 1, pp. 45–53, Mar. 1984.

[11] J. Broekens, M. Heerink, and H. Rosendal, “Assistive social robots in elderly care: a review,” *Gerontechnology*, pp. 94–103, 2009.

[12] J.-M. Fellous, “From Human Emotions to Robot Emotions,” p. 11.

[13] J. E. Colgate, J. Edward, M. A. Peshkin, and W. Wannasuphoprasit, *Cobots: Robots For Collaboration With Human Operators*. 1996.

[14] A. Vysocky and P. Novak, "HUMAN – ROBOT COLLABORATION IN INDUSTRY," *MM Sci. J.*, vol. 2016, no. 02, pp. 903–906, Jun. 2016.

[15] Å. Fast-Berglund, F. Palmkvist, P. Nyqvist, S. Ekered, and M. Åkerman, "Evaluating Cobots for Final Assembly," *6th CIRP Conf. Assem. Technol. Syst. CATS*, vol. 44, pp. 175–180, Jan. 2016.

[16] B. Zhang, Jianjun Wang, and T. Fuhlbrigge, "A review of the commercial brain-computer interface technology from perspective of industrial robotics," in *2010 IEEE International Conference on Automation and Logistics*, 2010, pp. 379–384.

[17] L. Shu *et al.*, "A Review of Emotion Recognition Using Physiological Signals," *Sensors*, vol. 18, p. 2074, Jun. 2018.

[18] M. Palaus Gallego, E. Muñoz Marrón, R. Viejo Sobera, and D. Redolar Ripoll, "Neural Basis of Video Gaming: A Systematic Review," May 2017.

[19] M. Ahn, M. Lee, J. Choi, and S. C. Jun, "A review of brain-computer interface games and an opinion survey from researchers, developers and users," *Sensors*, vol. 14, no. 8, pp. 14601–14633, Aug. 2014.

[20] A. P. Garcia, I. Schjølberg, and S. Gale, "EEG control of an industrial robot manipulator," in *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*, 2013, pp. 39–44.

[21] O. Raheja, K. Pawar, M. Saraf, and S. Maiti, "Mind Controlled Assistive Quadrotor Drone," vol. 7, no. 3, p. 5, 2018.

[22] M.-P. Pacaux-Lemoine, D. Trentesaux, G. Zambrano Rey, and P. Millot, "Designing intelligent manufacturing systems through Human-Machine Cooperation principles: A human-centered approach," *Comput. Ind. Eng.*, vol. 111, pp. 581–595, Sep. 2017.

[23] T. Sanders, K. E. Oleson, D. R. Billings, J. Y. C. Chen, and P. A. Hancock, "A Model of Human-Robot Trust: Theoretical Model Development," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 55, no. 1, pp. 1432–1436, Sep. 2011.

[24] E. de Visser and R. Parasuraman, "Adaptive Aiding of Human-Robot Teaming: Effects of Imperfect Automation on Performance, Trust, and Workload," *J. Cogn. Eng. Decis. Mak.*, vol. 5, no. 2, pp. 209–231, Jun. 2011.

[25] J. Yeow, P. K. Ng, K. Sin Tan, T. Chin, and W. Yin Lim, "Effects of Stress, Repetition, Fatigue and Work Environment on Human Error in Manufacturing Industries," *J. Appl. Sci.*, vol. 14, pp. 3464–3471, Sep. 2014.

[26] J. Y. C. Chen, M. J. Barnes, and M. Harper-Sciarini, “Supervisory Control of Multiple Robots: Human-Performance Issues and User-Interface Design,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 4, pp. 435–454, Jul. 2011.

[27] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. C. Chen, E. J. de Visser, and R. Parasuraman, “A Meta-Analysis of Factors Affecting Trust in Human-Robot Interaction,” *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 53, no. 5, pp. 517–527, Oct. 2011.

[28] M. Zamora, E. Caldwell, J. Garcia-Rodriguez, J. Azorin-Lopez, and M. Cazorla, “Machine Learning Improves Human-Robot Interaction in Productive Environments: A Review,” in *Advances in Computational Intelligence*, 2017, pp. 283–293.

[29] J. D. Lee and K. A. See, “Trust in Automation: Designing for Appropriate Reliance,” *Hum. Factors*, vol. 46, no. 1, pp. 50–80, Mar. 2004.

[30] R. Parasuraman and V. Riley, “Humans and Automation: Use, Misuse, Disuse, Abuse,” *Hum. Factors*, vol. 39, no. 2, pp. 230–253, Jun. 1997.

[31] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “A model for types and levels of human interaction with automation,” *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 30, no. 3, pp. 286–297, May 2000.

[32] C. D. Kidd and C. Breazeal, “Effect of a robot on user perceptions,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Sendai, Japan, 2004, vol. 4, pp. 3559–3564.

[33] V. Groom and C. Nass, “Can robots be teammates? Benchmarks in human-robot teams,” *Interact. Stud.*, vol. 8, pp. 483–500, Oct. 2007.

[34] T. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots,” *Robot. Auton. Syst.*, vol. 42, no. 3–4, pp. 143–166, Mar. 2003.

[35] J. Goetz, S. Kiesler, and A. Powers, “Matching robot appearance and behavior to tasks to improve human-robot cooperation,” presented at the Proceedings - IEEE International Workshop on Robot and Human Interactive Communication, 2003, pp. 55–60.

[36] B. R. Duffy, “Anthropomorphism and the social robot,” *Robot. Auton. Syst.*, vol. 42, no. 3, pp. 177–190, Mar. 2003.

[37] K. E. Schaefer, “The Perception and Measurement of Human-Robot Trust,” p. 359.

[38] M. Cummings and S. Guerlain, “Developing Operator Capacity Estimates for Supervisory Control of Autonomous Vehicles,” *Hum. Factors*, vol. 49, pp. 1–15, Mar. 2007.

[39] J. Chen, P. Durlach, Sloan, and Bowens, "Human robot interaction in the context of simulated route reconnaissance missions," *Mil. Psychol.*, vol. 20, pp. 135–149, Jul. 2008.

[40] P. Millot, "Toward Human-Machine Cooperation," in *Informatics in Control, Automation and Robotics: Selected Papers from the International Conference on Informatics in Control, Automation and Robotics 2007*, J. Filipe, J. A. Cetto, and J.-L. Ferrier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 3–20.

[41] O. Ogorodnikova, "Methodology of safety for a human robot interaction designing stage," in *2008 Conference on Human System Interactions*, 2008, pp. 452–457.

[42] B. C. Jiang and C. A. Gainer, "A cause-and-effect analysis of robot accidents," *J. Occup. Accid.*, vol. 9, no. 1, pp. 27–45, Jun. 1987.

[43] P. Huang, X. Xuan, R. Qiu, and G. Liu, "Statistical Analysis on Production Safety Accidents of Heavy Casualties of the Period 2001-2011 in China," *Procedia Eng.*, vol. 45, pp. 950–958, 2012.

[44] M. S. Sanders and E. J. McCormick, *Human factors in engineering and design*. New York: McGraw-Hill, 1993.

[45] M. Zare, M. Croq, F. Hossein-Arabi, R. Brunet, and Y. Roquelaure, "Does Ergonomics Improve Product Quality and Reduce Costs? A Review Article," *Hum. Factors Ergon. Manuf. Serv. Ind.*, vol. 26, Apr. 2015.

[46] J. A. E. Eklund, "Relationships between ergonomics and quality in assembly work," *Appl. Ergon.*, vol. 26, no. 1, pp. 15–20, Feb. 1995.

[47] M. Seeger, "COBOT SAFETY FEATURES IMPLEMENETED FOR MANUFACTURING," p. 8.

[48] F. Tobe, "Why Co-Bots Will Be a Huge Innovation and Growth Driver for Robotics Industry," *IEEE Spectrum: Technology, Engineering, and Science News*, 30-Dec-2015. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/industrial-robots/collaborative-robots-innovation-growth-driver. [Accessed: 22-Apr-2019].

[49] "Rethink Robotics | Smart Collaborative Robots | Advanced Automation Technology," *Rethink Robotics*. [Online]. Available: https://www.rethinkrobotics.com/. [Accessed: 22-Apr-2019].

[50] "ABB Group - Leading digital technologies for industry." [Online]. Available: https://new.abb.com. [Accessed: 22-Apr-2019].

[51] “Collaborative robotic automation | Cobots from Universal Robots.” [Online]. Available: https://www.universal-robots.com/. [Accessed: 22-Apr-2019].

[52] A. Maria Bauer, D. Wollherr, and M. Buss, “Human-Robot Collaboration: a Survey.,” *J Humanoid Robot.*, vol. 5, pp. 47–66, Mar. 2008.

[53] N. Fragopanagos and J. G. Taylor, “Emotion recognition in human-computer interaction,” *Neural Netw. Off. J. Int. Neural Netw. Soc.*, vol. 18, no. 4, pp. 389–405, May 2005.

[54] A. Fernández-Caballero *et al.*, “Smart environment architecture for emotion detection and regulation,” *J. Biomed. Inform.*, vol. 64, Dec. 2016.

[55] F. Abdat, C. Maaoui, and A. Pruski, “Human-Computer Interaction Using Emotion Recognition from Facial Expression,” in *2011 UKSim 5th European Symposium on Computer Modeling and Simulation*, 2011, pp. 196–201.

[56] M. M. H. E. Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognit.*, vol. 44, pp. 572–587, 2011.

[57] K. Hartmann, I. Siegert, D. Philippou-Hübner, and A. Wendemuth, “Emotion Detection in HCI: From Speech Features to Emotion Space,” *12th IFAC Symp. Anal. Des. Eval. Hum.-Mach. Syst.*, vol. 46, no. 15, pp. 288–295, Jan. 2013.

[58] F. Noroozi, C. Corneanu, D. Kamińska, T. Sapiński, S. Escalera, and G. Anbarjafari, “Survey on Emotional Body Gesture Recognition,” *IEEE Trans. Affect. Comput.*, vol. PP, Jan. 2018.

[59] D. O. Bos, *EEG-based Emotion Recognition The Influence of Visual and Auditory Stimuli*. .

[60] Y. Liu and O. Sourina, “EEG-based Dominance Level Recognition for Emotion-Enabled Interaction,” presented at the Proceedings of the 2012 IEEE International Conference on Multimedia and Expo, 2012, pp. 1039–1044.

[61] R. Ramirez and Z. Vamvakousis, “Detecting Emotion from EEG Signals Using the Emotive Epoc Device,” in *Brain Informatics*, 2012, pp. 175–184.

[62] S. Tivatansakul and M. Ohkura, “Emotion Recognition using ECG Signals with Local Pattern Description Methods,” *Int. J. Affect. Eng.*, Dec. 2015.

[63] “Emotion, Theories of | Internet Encyclopedia of Philosophy.” [Online]. Available: https://www.iep.utm.edu/emotion/. [Accessed: 20-May-2019].

[64] J. A. Russell, “Affective space is bipolar,” *J. Pers. Soc. Psychol.*, vol. 37, pp. 345–356, Mar. 1979.

[65] D. Graziotin, X. Wang, and P. Abrahamsson, “Understanding the Affect of Developers: Theoretical Background and Guidelines for Psychoempirical Software Engineering,” Jul. 2015.

[66] A. M. R. AlzeerAlhouseini, I. F. Al-Shaikhli, and M. A. Dzulkifli, “Emotion Detection Using Physiological Signals EEG & ECG,” *Int. J. Adv. Comput. Technol.*, p. 10, 2016.

[67] P. Ekman *et al.*, “Universals and cultural differences in the judgments of facial expressions of emotion,” *J. Pers. Soc. Psychol.*, vol. 53, no. 4, pp. 712–717, Oct. 1987.

[68] K. Blinowska and P. Durka, “Electroencephalography (EEG),” in *Wiley Encyclopedia of Biomedical Engineering*, American Cancer Society, 2006.

[69] B. Reuderink, C. Mühl, and M. Poel, “Valence, arousal and dominance in the EEG during game play,” *IJAACS*, vol. 6, pp. 45–62, 2013.

[70] E. Yurci, “EMOTION DETECTION FROM EEG SIGNALS : CORRELATING CEREBRAL CORTEX ACTIVITY WITH MUSIC EVOKED EMOTION,” 2014.

[71] D. M. Mirvis and A. L. Goldberger, “C H A P T E R 13 Electrocardiography,” p. 42.

[72] “Welcome 2.0 - HELLO MY NAME IS ROBOY.” [Online]. Available: https://roboy.org/. [Accessed: 29-Apr-2019].

[73] “Home - Affectiva : Affectiva.” [Online]. Available: https://www.affectiva.com/. [Accessed: 29-Apr-2019].

[74] “SoftBank Robotics - Group | Global Site,” *SoftBank Robotics - Group | Global Site*. [Online]. Available: https://softbankrobotics.com/. [Accessed: 29-Apr-2019].

[75] U. Herwig, P. Satrapi, and C. Schönfeldt-Lecuona, “Using the International 10-20 EEG System for Positioning of Transcranial Magnetic Stimulation,” *Brain Topogr.*, vol. 16, no. 2, pp. 95–99, 2003.

[76] “EEG - ECG - Biosensors.” [Online]. Available: http://neurosky.com/. [Accessed: 20-May-2019].

[77] “OpenBCI - Open Source Biosensing Tools (EEG, EMG, EKG, and more).” [Online]. Available: https://openbci.com/. [Accessed: 20-May-2019].

[78] “Homepage,” *Emotiv*. [Online]. Available: https://www.emotiv.com/. [Accessed: 20-May-2019].

[79] “Cognionics Dry EEG Headsets and Dry Electrode Sensors,” *CGX*. [Online]. Available: https://www.cognionics.net. [Accessed: 20-May-2019].

[80] “ANT Neuro | inspiring technology for the human brain.” [Online]. Available: https://www.ant-neuro.com/. [Accessed: 20-May-2019].

[81] P. K. Choubey, K.-S. Chung, M. Kim, J. Lee, and R. R. Srivastava, “Advance review on the exploitation of the prominent energy-storage element Lithium. Part II: From sea water and spent lithium ion batteries (LIBs),” *Miner. Eng.*, vol. 110, pp. 104–121, Aug. 2017.

[82] E. Heer, “Robots in modern industry,” *Astronaut. Aeronaut.*, vol. 19, pp. 50–59, Sep. 1981.

[83] A. P. Calitz, P. Poisat, and M. Cullen, “The future African workplace: The use of collaborative robots in manufacturing,” 2017.

[84] “Login | Derwent Innovation.” [Online]. Available: https://www.derwentinnovation.com/login/. [Accessed: 20-May-2019].

[85] I. Maurtua, A. Ibarguren, J. Kildal, L. Susperregi, and B. Sierra, “Human–robot collaboration in industrial applications: Safety, interaction and trust,” *Int. J. Adv. Robot. Syst.*, vol. 14, no. 4, p. 1729881417716010, Jul. 2017.

[86] R. Bejarano, B. Ramis Ferrer, W. M. Mohammed, Martinez Lastra, and J. L. Martinez Lastra, “Implementing a Human-Robot Collaborative Assembly Workstation,” 2019.

[87] P. Tsarouchi *et al.*, “Robotized Assembly Process Using Dual Arm Robot,” *Procedia CIRP*, vol. 23, pp. 47–52, Jan. 2014.

[88] J. Krüger, G. Schreck, and D. Surdilovic, “Dual arm robot for flexible and cooperative assembly,” *CIRP Ann.*, vol. 60, no. 1, pp. 5–8, Jan. 2011.

[89] A. Rajendra Desai, “EEG-based evaluation of cognitive and emotional arousal when coding in different programming languages,” 2017. .

[90] M. Strmiska, Z. Koudelkova, and M. Zabcikova, “Measuring Brain Signals Using Emotiv Devices,” vol. 13, p. 6, 2018.

[91] Y. Lyu, S. Garti, and M. Honkanen, “Emotiv Epoc+ live metrics quality validation,” p. 10.

[92] L. Irish and K. J. Christensen, “A ‘Green TCP/IP’ to reduce electricity consumed by computers,” in *Proceedings IEEE Southeastcon ’98 “Engineering for a New Era,”* 1998, pp. 302–305.

[93] W. R. Stevens and K. R. Fall, “TCP/IP Illustrated, Volume 1: The Protocols,” p. 1059.

[94] “Introduction – Cortex API Documentation.” [Online]. Available: https://emotiv.github.io/cortex-docs/#introduction. [Accessed: 20-May-2019].

[95] “sloankelly — Bitbucket.” [Online]. Available: https://bitbucket.org/sloankelly/. [Accessed: 16-May-2019].

[96] “R. Bejarano, B. Ramis, W. Mohammed, J. Martinez, ‘Implementing a Human-Robot Collabora-tive Assembly Workstation’, 2019.”

[97] L. Amezua Hormaza, W. M. Mohammed, B. Ramis Ferrer, R. Bejarano, and J. L. Martinez Lastra, “On-line Training and Monitoring of Robot Tasks through Virtual Reality,” Jul. 2019.

APPENDIX A: TOEYMOTIONS CODE

This section presents the main programming code of the three blocks that comprise *ToEymotions*.

6.1 Block 1: Emotion Feedback

- Client Class Header

```
#ifndef CLIENT_H
#define CLIENT_H


#include <QObject>
#include <QString>
#include <QWebSocket>
#include <QTimer>
#include <QThread>
#include <QList>
#include <QVector>
#include <list>
#include <ws2tcpip.h>
#include "DataStreamExample.h"

class ServerCommunication;


//----------------EMOTION COMMUNICATION-------------

class Client : public QObject
{
    Q_OBJECT

public:
    explicit  Client(DataStreamExample  &dataSt,  ServerCommunication
&serverCommunication, QObject *parent = nullptr);
    void classifyEmotionValue();

public slots:

    void saveEmotionValue(double emotion_value, int pos);
    void startStepTrue();
    void startStepFalse();

public:

    QString set_vel_msg;
    QString delays;
    double emotion_value;
```

```
    double emo_array[7];


  private:

    DataStreamExample &dse;
    ServerCommunication &sc;
    bool start_step = false;
    int emo_count = 0; //Number of emotion arrays that are received while performing the step
    int pos; // Variable to count when emo_array is full
    double ave_emo;
    double emo_sum[7];
    double emo_ave[7];

  };


  #endif // CLIENT_H
```

- Main .cpp (Emotional Feedback section)

```
  #include "client.h"
  #include "servercommunication.h"

  #include <QCoreApplication>
  #include <iostream>
  #include <string>
  #include <ws2tcpip.h> //Winsows Sockets is the API/Framework that windows uses to access to network sockets
  #include <QString>
  #include <QtDebug>
  #include <QThread>


  #pragma comment (lib, "ws2_32.lib") // Link the compiler with the windsock library


  using namespace std;


  //--------------------EMOTIONAL FEEDBACK SECTION------------------
  //---------------------------(BLOCK 1)---------------------------


  Client::Client(DataStreamExample &dataSt, ServerCommunication &serv-erCommunication, QObject *parent) :
      QObject(parent), dse(dataSt), sc (serverCommunication)
  {
      //timer = new QTimer(this);
```

```
      connect(&dataSt,   &DataStreamExample::sendEmotionValue,   this,
&Client::saveEmotionValue);

      connect(&serverCommunication,   &ServerCommunication::startStep,
this, &Client::startStepTrue);
      connect(&serverCommunication,     &ServerCommunication::endStep,
this, &Client::startStepFalse);


  }

  void Client::startStepTrue()
  {
      start_step = true;
      cout << "---------START STEP--------\n";

  }

  void Client::startStepFalse()
  {
      start_step = false;
      cout << "----------END STEP------\n";

      classifyEmotionValue();
      for(int i=0;i<=6;i++)
             {
                 emo_sum[i] = 0; // Restart values
             }
             emo_count = 0; // Restart values
  }

  void Client::saveEmotionValue(double emotion_value, int pos)
  {
      // HERE IS WHERE I AM SAVE THE EMOTIONAL VALUES INTO AN ARRAY

      this->emotion_value = emotion_value;
      this->pos = pos;

      emo_array[this->pos] = this->emotion_value;

      if(start_step == true && pos==6)
      {
         for(int i=0;i<=6;i++)
         {
            // cout << "FOR ARRAYS \n";
             emo_sum[i] = emo_sum[i] + emo_array[i];
             switch (i)
             {
                 case 0:
                      cout << "\n INTEREST -> " << "ACTUAL VALUE: " <<
emo_array[i] << " -- SUM: " << emo_sum[i] <<"\n";
                 break;
```

```
                case 1:
                     cout << "\n STRESS -> " << "ACTUAL VALUE: " <<
emo_array[i] << " -- SUM: " << emo_sum[i] <<"\n";
                break;

                case 2:
                     cout << "\n RELAXATION -> " << "ACTUAL VALUE: "
<< emo_array[i] << " -- SUM: " << emo_sum[i] <<"\n";
                break;

                case 3:
                     cout << "\n EXCITEMENT -> " << "ACTUAL VALUE: "
<< emo_array[i] << " -- SUM: " << emo_sum[i] <<"\n";
                break;

                case 4:
                     cout << "\n ENGANGEMENT -> " << "ACTUAL VALUE: "
<< emo_array[i] << " -- SUM: " << emo_sum[i] <<"\n";
                break;

                case 5:
                     cout << "\n LONG TERM EXCITEMENT -> " << "ACTUAL
VALUE: " << emo_array[i] << " -- SUM: " << emo_sum[i] <<"\n";
                break;

                case 6:
                     cout << "\n FOCUS -> " << "ACTUAL VALUE: " <<
emo_array[i] << " -- SUM: " << emo_sum[i] <<"\n";
                break;
            }

        }
        emo_count++;
        return;
    }

  }


  void Client::classifyEmotionValue()
  {
      cout << "\n --------------------------------CLASSIFYING EMO-
TIONAL VALUE--------------------------------\n" << endl;

      cout << "NUMBER OF EMOTIONAL ARRAYS CONSIDERED: " << emo_count <<
"\n";

      //Calculation of the average value of each metric
      for(int i=0; i<=6; i++)
      {
          emo_ave[i] = emo_sum[i]/emo_count;

          switch (i)
          {
```

```
                case 1:
                    cout << "\nSTRESS -> " << " AVERAGE: " << emo_ave[i]
<<"\n";
                break;

                case 4:
                    cout  <<  "ENGANGEMENT  ->  "  <<  "  AVERAGE:  "  <<
emo_ave[i] <<"\n";
                break;

                case 6:
                    cout << "FOCUS -> " << " AVERAGE: " << emo_ave[i]
<<"\n";
                break;
            }


        }

        // The classification is using STRESS as main emotion and ENGANGE-
MENT and FOCUS as side ones
        /**
          0 -> INT
          1 -> STR
          2 -> REL
          3 -> EXC
          4 -> ENG
          5 -> LEX
          6 -> FOC

          ----------

          HIGH -> 100% SPEED - (This are example values)
          MID  -> 50% SPEED - (This are example values)
          SLOW -> 30% SPEED - (This are example values)
          **/

    // THE FOLLOWING RANGES ARE ADAPTED FOR THE USER

        if(emo_ave[1]>=0 && emo_ave[1]<0.239)
        {
            cout << "\n 1st CASE " << "-STRESS IS LOW- " << "\n";
            set_vel_msg = "HIGH";
            // If user not ENG neither FOC the speed must be reduced,
because he could be distracted
            if((emo_ave[4]>=0 && emo_ave[4]<=0.517) && (emo_ave[6]>=0 &&
emo_ave[6]<=0.352) )
            {
                set_vel_msg = "MID"; // User is not focused
                cout << "--Case 1.1-- \n";
            }
        }
        else if(emo_ave[1]>=0.239 && emo_ave[1]<0.485)
        {
```

```
          cout << "\n 2nd CASE " << "-STRESS IS MID-LOW- " << "\n";
          // STRESS IS MID-LOW
          set_vel_msg = "HIGH";

          // IF: Is a bit stressed but ENG and FOC are low, is not
focused so I change the speed to mid (not high)
          if((emo_ave[4]>=0 && emo_ave[4]<=0.517) && (emo_ave[6]>=0 &&
emo_ave[6]<=0.51) )
          {
              set_vel_msg = "MID";
              cout << "--Case 2.1-- \n";
          }

      }
      else if(emo_ave[1]>=0.485 && emo_ave[1]<0.64)
      {
          cout << "\n 3rd CASE " << "-STRESS IS MID-HIGH- " << "\n";
          // STRESS IS MID-HIGH
          set_vel_msg = "MID";

          // MH & H Engagement and MH & H Focus, means that despite the
stress user is focus or enganged in the task ("flow state")
          if((emo_ave[4]>=0.664 && emo_ave[4]<=1) && (emo_ave[6]>=0.51
&& emo_ave[6]<=1) )
          {
              set_vel_msg = "HIGH"; // User is focused
              cout << "--Case 3.1-- \n";
          }
          // H Engagement and ML & MH Focus, means that despite the
stress user is focus or in "flow state"
          else    if((emo_ave[4]>=0.775    &&    emo_ave[4]<=1)    &&
(emo_ave[6]>=0.352 && emo_ave[6]<=1))
          {
              set_vel_msg = "HIGH";
              cout << "--Case 3.2-- \n";
          }
          // This means that there is not focus and there could be a
problem with middle level of stress
          else    if((emo_ave[4]>=0    &&    emo_ave[4]<=0.517)    &&
(emo_ave[6]>=0 && emo_ave[6]<=0.352) )
          {
              set_vel_msg = "SLOW"; // User is focused
              cout << "--Case 3.3-- \n";
          }
      }
      else if(emo_ave[1]>=0.64 && emo_ave[1]<=0.859)
      {
          cout << "\n 4th CASE " << "-STRESS IS HIGH- " << "\n";
          //STRESS IS HIGH
          set_vel_msg = "SLOW";
          // MH & H Engagement and MH & H Focus, means that despite the
stress user is focus or in flow state
          if((emo_ave[4]>=0.664 && emo_ave[4]<=1) && (emo_ave[6]>=0.51
&& emo_ave[6]<=1) )
```

```
          {
              set_vel_msg = "MID"; // User is focused
              cout << "--Case 4.1-- \n";
          }
          // H Engagement and ML & MH Focus, means that despite the
stress user is focus or in flow state
          else    if((emo_ave[4]>=0.775    &&    emo_ave[4]<=1)    &&
(emo_ave[6]>=0.352 && emo_ave[6]<=1))
          {
              set_vel_msg = "MID";
              cout << "--Case 4.2-- \n";
          }
      }

      else if(emo_ave[1]>=0.859 && emo_ave[1]<=1)
      {
          cout << "\n 5th CASE " << "-STRESS IS SUPER HIGH- " << "\n";
          //STRESS IS SUPER HIGH
          set_vel_msg = "SLOW";
          // Condition: SUPER STRESS but SUPER ENG and SUPER FOC, time
to MID
          if((emo_ave[4]>=0.78 && emo_ave[4]<=1) && (emo_ave[6]>=0.67
&& emo_ave[6]<=1) )
          {
              set_vel_msg = "MID"; // User is focused
              cout << "--Case 5.1-- \n";
          }
      }

      if(set_vel_msg == "HIGH")
      {
          delays = "LOW";
      }
      else if(set_vel_msg == "MID")
      {
          delays = "MID";
      }
      else if(set_vel_msg == "SLOW")
      {
          delays = "HIGH";
      }

      // Comunicate it to server communication that the classification
is done
      sc.sendingParameters(set_vel_msg, delays);

      cout << "---------CLASSIFICATION ENDS----------\n" << endl;


  }
```

6.2 Block 2: Communication with the cobot

- ServerCommunication Class Header

```
#ifndef SERVERCOMMUNICATION_H
#define SERVERCOMMUNICATION_H

#include <QObject>
#include <QString>
#include <QWebSocket>
#include <QTimer>
#include <QThread>
#include <QList>
#include <QVector>
#include <list>
#include <ws2tcpip.h>
#include "DataStreamExample.h"

class Client;

//----------------------SERVER COMMUNICATION--------------------

class ServerCommunication : public QObject
{
    Q_OBJECT

public:
    explicit ServerCommunication(QThread &serverCommunicationThread,
QObject *parent = nullptr);

    void manageServerMessage(QString server_msg);
    void sendingParameters(QString velocity, QString delays);

public slots:
    void startCommunication();

signals:
    void startStep();
    void endStep();


private:
    QWebSocket mySocket;
    QString server_msg;
    QString client_msg;
    QThread &sct; // Server Communication Thread
    QString delays;
    SOCKET sock;
    bool velChanged;
    QString velocity;

};
```

```
#endif // SERVERCOMMUNICATION_H
```

- Main.cpp (Server Communication section + main)

```
//-----------------SERVER COMMUNICATION SECTION-----------
//-------------------------(BLOCK 2)---------------------


ServerCommunication::ServerCommunication(QThread &serverCommunica-
tionThread, QObject *parent) : QObject(parent), sct(serverCommunica-
tionThread)
{

    connect(&serverCommunicationThread, &QThread::started, this,
&ServerCommunication::startCommunication);
    connect(&serverCommunicationThread, &QThread::finished, this,
&QObject::deleteLater);
    connect(this, &QObject::destroyed, &serverCommunicationThread,
&QThread::deleteLater);

}

// METHOD CONNECTED TO QTHREAD FOR MULTITHREADING
void ServerCommunication::startCommunication()
{

    cout << "START COMUNNICATION WITH SERVER \n";
    // allocate resources using new here
    // DataStreamExample dataStream;
     //Client client(dse);
     string ipAddress = "192.168.125.1"; // Is the IP address of the
server we want to connect. It is "127.0.0.1" because it was the local
machine.
     int port = 54000; // Listening port on the server. The applica-
tion is listening this port and server to

     // INITIALIZE WINSOCK

     WSADATA data; //WinSock structure - data Structure
     WORD ver = MAKEWORD(2, 2); // Version number of WinSock(?)
     int wsResult = WSAStartup(ver, &data); //To start the the WinSock
     // data is filled if wsResult returns 0, if not it will return
an error
     if(wsResult != 0)
     {
         cerr << "Can't start WinSock, Err #" << wsResult << endl; //
The # is for saying "number"
         return;
     }

     // CREATE SOCKET
```

```
        sock = socket(AF_INET, SOCK_STREAM, 0);
        // Check if the socket is valid. If the value return by sock is
negative it will be invalid
        if(sock == INVALID_SOCKET)
        {
            // WSAGetLastError() = It is for getting the last error code
as the name sais
            cerr << "Can't create socket, Err #" << WSAGetLastError() <<
endl;
            WSACleanup();
            return;
        }

        // FILL IN A HINT STRUCTURE

        // Hint structure will say WinSock what server and what port we
want to connect it
        sockaddr_in hint;
        hint.sin_family = AF_INET; // Specify Family
        hint.sin_port = htons(port); // Specify Port number (port was
defined before)
        inet_pton(AF_INET, ipAddress.c_str(), &hint.sin_addr); //the
"c_str()" is to convert to a Cstyle string; &hint... is the address of
the buffer where the internet IP address goes
        /* (A buffer is a temporal storage memory of info that allows us
to transfer data between
         * functional unities with different transfer characteristics.
         * It is like a "translator". One is spanish the other finnish,
so the buffer "allows" the communication
         * by translating the words that one is saying to the other. The
translator will be the storage space).
        */

        // CONNECT TO A SERVER


        // Connection of the socket to a remote server
        int connResult = ::connect(/*client.*/sock, (sockaddr*)&hint,
sizeof(hint));
        if (connResult == SOCKET_ERROR)
        {
            cerr << "Can't connect to server, Err #" << WSAGetLastError()
<< endl;
            closesocket(/*client.*/sock);
            WSACleanup();
            return;
        }
        else
        {
            cout << "CLIENT> I am connected\n" << endl;

        }
```

```
        int buf_len = 4096;
        char* buf = new char[buf_len];

        while(true)
        {
            // Wait for response
            ZeroMemory(buf, 4096); // Cleaning the buffer
            //qDebug() << "1";
            int bytesRecieved = recv(sock, buf, 4096, 0);
            if(bytesRecieved == SOCKET_ERROR) cout << "Error: #" << WSA-
GetLastError() << endl;
            //qDebug() << "2";
            // if client disconnect "bytesRecieved" = 0, if there is an
error = -1, if recieved somethinf = +positive
            // Echo response to console - if I have recieved anything;
            if(bytesRecieved > 0)
            {
              //  qDebug() << "3";
                server_msg  =  QString::fromStdString(string(buf,  0,
bytesRecieved)); // Saving what Server says into server_msg
                cout << "SERVER> " << string(buf, 0, bytesRecieved) <<
endl; // Print out what Server said
                manageServerMessage(server_msg); // Manage what server
said
              // Echo response

              // The string contents: contents of buffer, former index
and number of characters=bytesRecieved
            }else{
                cout << "NO BYTES RECIEVED" << endl;
            }

        }
    }


    void ServerCommunication::manageServerMessage(QString server_msg)
    {
        this->server_msg = server_msg;

        if(this->server_msg == "START STEP")
        {
            cout << "Starting emotional feedback...\n" << endl;

           // Emit the signal to call the method that will save the
values on an array-list
            emit startStep();
           return;
        }
        else if(this->server_msg == "END STEP")
        {
           cout << "End of emotional feedback. \n" << endl;
           emit endStep();
```

```
        }
        else
        {
            cout << "Messsage not clear\n" << endl;
            return;
        }
    }

    void     ServerCommunication::sendingParameters(QString     velocity,
QString delays)
    {
        cout << "\n --SETTING COBOTS PARAMETERS--\n";
        this->velocity = velocity;
        this->delays = delays;
        cout << "\nCHANGING VELOCITY T0 : " << velocity.toStdString() <<
"\n";
        cout << "CHANGING DELAYS T0 : " << delays.toStdString() << "\n";

    // SENDING PARAMETERS TO COBOT
        send(sock, velocity.toStdString().c_str(), velocity.size(), 0);

    }

    //-----------------------MAIN---------------------
    int main(int argc, char *argv[])
    {

        QCoreApplication a(argc, argv);
        DataStreamExample dse;
        QThread sct;
        ServerCommunication sc(sct);
        Client client(dse,sc);


        const QString license = "WRITE USER LICENSING"; //For high-reso-
lution PM and Raw EEG it must be PRO License


        qInfo() << "";
        qInfo() << " ######### AITOR IS USING TOEYMOTIONS #########";
        qInfo() << "";


        dse.start("met", license);


        sc.moveToThread(&sct);

        sct.start();


        return a.exec();

        }
```

6.3 Block 3: Modification of cobot's parameters

- Right Arm

```
MODULE MainModule
    !Classification targets
    CONST robtarget pick:=[[172.60,-287.02,52.11],[0.723557,0.00798275,0.690005,0.0171397],[1,0,2,4],[-106.454,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget inspect:=[[196.44,-261.26,218.56],[0.5266,-0.47944,0.496271,-0.496534],[0,0,1,4],[-106.465,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget packing:=[[497.01,-122.70,71.36],[0.754079,0.0324416,0.655717,0.0186784],[1,0,2,4],[-93.4555,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget buffer:=[[376.86,-243.83,158.65],[0.78616,0.0170034,0.616827,0.034475],[1,0,2,4],[-103.335,9E+09,9E+09,9E+09,9E+09,9E+09]];
    !Box faces targets
    VAR robtarget F1:=[[251.15,45.99,111.40],[0.0361444,0.709161,-0.703849,0.0195104],[1,2,0,4],[175.456,9E+09,9E+09,9E+09,9E+09,9E+09]];
    VAR robtarget F5;
    !Fitting targets
    CONST robtarget F12_R :=[[454.09,-54.20,154.83],[0.998033,0.0589704,-0.0212458,-0.00131195],[1,0,2,4],[-136.196,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F32_R :=[[447.16,-79.05,174.37],[0.693291,0.0629267,0.0327608,-0.717157],[1,0,0,4],[-166.073,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F43_R :=[[478.22,-85.35,181.44],[0.674665,0.0806442,0.0524449,-0.731829],[1,1,0,4],[-166.077,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F54_R :=[[295.89,-7.95,194.29],[0.474717,-0.50631,0.509383,-0.508746],[1,1,0,4],[-176.833,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F65_R :=[[426.03,13.22,187.27],[0.262985,0.670715,-0.62968,0.290659],[1,2,0,4],[-174.995,9E+09,9E+09,9E+09,9E+09,9E+09]];
    !Grasp from one side to avoid colision on final position
    !Other Variables
    VAR cameratarget mycameratarget;
    VAR bool LoadPr:= TRUE;
    CONST string myjob := "job1.job";
    VAR robtarget p1;
    !Multitask synchronization variables
    PERS tasks multi_task{2} := [["T_ROB_R"], ["T_ROB_L"]];
    VAR syncident sync1;
    VAR syncident sync2;
    VAR syncident sync3;
    VAR syncident sync4;
    VAR syncident sync5;
```

```
    VAR syncident sync6;
    VAR syncident sync7;
    VAR syncident sync8;
    !VAR syncident sync9;
    VAR syncident sync9;
    !VAR syncident sync11;
    VAR syncident sync10;
    VAR syncident sync11;
    VAR syncident sync12;
    VAR syncident sync13;
    VAR syncident sync14;
    VAR syncident sync15;
    VAR syncident sync16;
    VAR syncident sync17;
    VAR syncident sync18;
    VAR num delays;
    VAR num gripperforce;

    !VARIABLES FOR EMOTIONAL FEEDBACK
    VAR socketdev emotion_socket;
    VAR socketdev server_socket;
    VAR string message;
    !VAR speeddata velocity;
    PERS num vel;

    !Procedures - Functions
    PROC EMOTIONAL_FEEDBACK()
        SocketSend emotion_socket, \Str:= "END STEP";
        !Change for continue signal (button, sensor, gesture recog-
nition)
        !ADD EMOTIONAL FEEDBACK HERE
        SocketReceive emotion_socket, \Str:= message;
        TPWrite message; !Write received message on FlexPendant
        TPWrite "ACTUAL SPEED 1): " + ValToStr(vel);
        !Velocity value
        TEST message

        CASE "HIGH":
            !Set velocity to HIGH level v = 100%
            vel := 100;
            SpeedRefresh vel;
            vel := CSpeedOverride(\CTask);
            TPWrite "CHANGING SPEED (R) TO: " + ValToStr(vel);
            !velocity.v_tcp:= vmax;
            TPWrite "vel = HIGH";
            !Set delays to LOW level delays = 40s
            delays := 40;

        CASE "MID":
            !Set velocity to MID level v = 50%
            vel := 50;
            SpeedRefresh vel;
            vel := CSpeedOverride(\CTask);
            TPWrite "CHANGING SPEED (R) TO: " + ValToStr(vel);
```

```
                TPWrite "vel = MID";
                !Set delays to MID level delays = 60s
                delays := 60;

            CASE "SLOW":
                !Set velocity to LOW level v = 30%
                vel := 30;
                SpeedRefresh vel;
                vel := CSpeedOverride(\CTask);
                TPWrite "CHANGING SPEED (R) TO: " + ValToStr(vel);
                TPWrite "vel = SLOW";
                !Set delays to HIGH level delays = 80s
                delays := 80;

            DEFAULT:
                !Set velocity to MID level
                !Set delays to MID level
                TPWrite "DEFAULT VALUES ";
                vel := 75;
                SpeedRefresh vel;
                vel := CSpeedOverride(\CTask);
                TPWrite "CHANGING SPEED (R) TO: " + ValToStr(vel);

                delays := 60;
            ENDTEST
            TPWrite "DELAYS: " + ValToStr(delays);
            WaitTime(3);
        ENDPROC

        PROC TRANSITION()
            !User can wait the delays or if it has finish presh the button
to continue assembly
            WaitTime(delays);
            EMOTIONAL_FEEDBACK;
        ENDPROC
        PROC CLASSIFY()
            vel := 100;
            SpeedRefresh vel;
            vel := CSpeedOverride(\CTask);
            TPWrite "CHANGING SPEED (R) TO: " + ValToStr(vel);
            !-----------------------------------------------
            MoveL inspect, v2000, fine, tool0;
            CamReqImage RightCAM;
            CamGetResult RightCAM, mycameratarget;
            !vision inspect
            !pick
            MoveL Offs(pick, 0, 0, 180), v1000, fine, tool0; !Align with
picking point (elevated)
            MoveL pick, v100, fine, tool0;
            g_VacuumOn1;
            WaitTime (2);
            SetDO custom_DO_7,1;
            !lift
            MoveL Offs(pick, 0, 0, 20), v20, fine, tool0;
```

```
        MoveL Offs(pick, 0, 0, 40), v50, fine, tool0;
        MoveL Offs(pick, 0, 0, 180), v80, fine, tool0;
        SetDO custom_DO_7,0;
        If mycameratarget.val1=1 THEN
            MoveJ Offs(pick, 210, 170, 180), v100, fine, tool0;
            MoveL Offs(pick, 210, 170, 30), v100, fine, tool0;
        ELSEIF mycameratarget.val2=1 THEN
            MoveJ Offs(pick, 210, 170, 180), v1000, fine, tool0;
            MoveL Offs(pick, 210, 170, 30), v100, fine, tool0;
        ELSEIF mycameratarget.val3=1  THEN
            MoveJ Offs(pick, 220, 20, 140), v1000, fine, tool0;
            MoveL Offs(pick, 220, 20, 30), v100, fine, tool0;
        ENDIF
        !Drop
        g_VacuumOff1;
        g_BlowOn1;
        WaitTime (1);
        g_BlowOff1;
        p1 := CRobT();
        MoveL Offs(p1, 0, 0, 100), v2000, fine, tool0;
    ENDPROC
    PROC syncmove()
        SyncMoveOn sync2, multi_task;
        MoveL F12_R \ID:=10, v50, fine, tool0;
        SocketSend emotion_socket, \Str:= "START STEP";
        TRANSITION;
        SyncMoveOff sync3;
        UNDO
            SyncMoveUndo;
    ENDPROC
    PROC syncmove2()
        SyncMoveOn sync5, multi_task;
        MoveL F32_R \ID:=20, v50, fine, tool0;
        SocketSend emotion_socket, \Str:= "START STEP";
        TRANSITION;
        SyncMoveOff sync6;
        UNDO
            SyncMoveUndo;
    ENDPROC
    PROC syncmove3()
        SyncMoveOn sync12, multi_task;
        MoveL F43_R \ID:=30, v50, fine, tool0;
        SocketSend emotion_socket, \Str:= "START STEP";
        TRANSITION;
        SyncMoveOff sync13;
        UNDO
            SyncMoveUndo;
    ENDPROC
    PROC syncmove4()
        SyncMoveOn sync16, multi_task;
        MoveL F54_R \ID:=40, v50, fine, tool0;
        SocketSend emotion_socket, \Str:= "START STEP";
        TRANSITION;
        SyncMoveOff sync17;
```

```
        !Because the collaborative part is over
        vel := 100;
        SpeedRefresh vel;
        vel := CSpeedOverride(\CTask);
        TPWrite "CHANGING SPEED (R) TO: " + ValToStr(vel);

        UNDO
            SyncMoveUndo;
    ENDPROC
    PROC Loadcam()
        IF LoadPr=TRUE THEN
            CamSetProgramMode RightCAM;
            CamLoadJob RightCAM, "job1.job";
            CamSetRunMode RightCAM;
            LoadPr:=FALSE;
        ENDIF
    ENDPROC
   PROC main()
        !**************************************************
        !EMOTIONAL COMMUNICATION
        !**************************************************

        !Creating communication
        SocketCreate server_socket;
        SocketBind server_socket, "192.168.125.1", 54000;
        SocketListen server_socket;
        SocketAccept server_socket, emotion_socket;

        !**************************************************
        !Box Assembly
        !**************************************************
        F5 := Offs(F1,0,50,0);
        delays := 40;
        gripperforce := 20;
        g_Init \maxSpd:=10 \holdForce:=10 \Calibrate;
        g_SetForce 10;
        g_SetMaxSpd 10;
        MoveJ Offs(F1, 0, 0, 100), vmax, fine, tool0;
        g_GripOut;
        MoveL F1, v50, fine, tool0;
        g_GripIn \holdForce:=gripperforce;
        MoveL Offs(F1, 0, 0, 100), vmax, fine, tool0;
        MoveJ RelTool(F12_R, 0, -50, 0), vmax, fine, tool0;
        WaitSyncTask sync1, multi_task;
        syncmove;
        WaitSyncTask sync4, multi_task;
        WaitSyncTask sync7, multi_task;
        MoveL RelTool(F32_R, 70, 0, -50), vmax, fine, tool0;
        WaitSyncTask sync8, multi_task;
        MoveL RelTool(F32_R, 0, 0, -50), vmax, fine, tool0;
        syncmove2;
        WaitSyncTask sync9, multi_task;
        MoveJ RelTool(F43_R, 50, 0, 0), v100, fine, tool0;
        syncmove3;
```

```
        g_GripOut;
        MoveL RelTool(F43_R, 0, 0, -50), v100, fine, tool0;
        WaitSyncTask sync10, multi_task;
        MoveL RelTool(F43_R, 0, -300, -50), v100, fine, tool0;
        MoveJ Offs(F5, 0, 0, 100), vmax, fine, tool0;
        MoveL F5, v50, fine, tool0;
        g_GripIn \holdForce:=gripperforce;
        MoveL Offs(F5, 0, 0, 100), vmax, fine, tool0;
        WaitSyncTask sync11, multi_task;
        MoveJ RelTool(F54_R, 100, 0, 0), v100, fine, tool0;
        syncmove4;
        WaitTime(delays);
        g_GripOut;
        MoveL RelTool(F54_R, 0, 0, -250), vmax, fine, tool0;
        WaitSyncTask sync14, multi_task;
        MoveJ inspect, vmax, fine, tool0;
        WaitSyncTask sync15, multi_task;
        !**************************************************
        !Classification
        !**************************************************
        Loadcam;
        FOR i FROM 1 TO 3 DO
           CLASSIFY;
        ENDFOR
        MoveL inspect, v2000, fine, tool0;
        WaitSyncTask sync18, multi_task;

        !**************************************************
        !EMOTIONAL COMMUNICATION
        !**************************************************

        !Closing communication
        SocketClose emotion_socket;

    ENDPROC
ENDMODULE
```

- Left Arm

```
MODULE MainModule
    !Faces targets
    VAR robtarget F2:=[[217.50,270.42,97.01],[0.0378795,-0.69461,-0.717906,0.0263026],[-1,2,1,4],[109.144,9E+09,9E+09,9E+09,9E+09,9E+09]];
    VAR robtarget F3;
    VAR robtarget F4;
    VAR robtarget F6;
    CONST robtarget HomePos:=[[36.36,252.47,230.83],[0.0230078,-0.650791,-0.758609,0.021297],[0,2,1,4],[103.371,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget packing_drop:=[[187.23,-79.29,86.12],[0.101524,-0.993924,0.0127726,-0.0405671],[-1,-2,1,4],[149.436,9E+09,9E+09,9E+09,9E+09,9E+09]];
    !Fitting targets
    CONST robtarget F12_L:=[[364.28,28.48,373.22],[0.0448571,-0.692267,-0.720223,0.0057613],[-1,1,0,4],[155.125,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F32_L:=[[247.16,-4.13,271.85],[0.709806,0.0450996,0.70216,0.033366],[-1,1,1,5],[155.814,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F43_L:=[[460.02,95.38,135.53],[0.694197,0.0088047,0.0374715,-0.718755],[-1,-1,2,4],[172.968,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F54_L:=[[279.09,84.09,241.05],[0.715549,0.0470472,0.696597,0.0229988],[-1,-1,-1,4],[176.863,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget F65_L:=[[271.86,94.38,67.61],[0.343411,0.572924,0.627282,-0.40043],[-1,1,1,4],[135.686,9E+09,9E+09,9E+09,9E+09,9E+09]];
   !Multitask synchronization variables
    PERS tasks multi_task{2} := [["T_ROB_R"], ["T_ROB_L"]];
    VAR syncident sync1;
    VAR syncident sync2;
    VAR syncident sync3;
    VAR syncident sync4;
    VAR syncident sync5;
    VAR syncident sync6;
    VAR syncident sync7;
    VAR syncident sync8;
    VAR syncident sync9;
    VAR syncident sync10;
    VAR syncident sync11;
    VAR syncident sync12;
    VAR syncident sync13;
    VAR syncident sync14;
    VAR syncident sync15;
    VAR syncident sync16;
    VAR syncident sync17;
    VAR syncident sync18;
```

```
VAR num gripperforce;

!VARIABLES FOR EMOTIONAL FEEDBACK
PERS num vel;
!Procedures - Functions
PROC syncmove()
    SyncMoveOn sync2, multi_task;
    MoveL F12_L \ID:=10, v50, fine, tool0;
    SyncMoveOff sync3;
    SpeedRefresh vel;
    vel := CSpeedOverride(\CTask);
    TPWrite "CHANGING SPEED (L) TO: " + ValToStr(vel);

    UNDO
        SyncMoveUndo;
ENDPROC
PROC syncmove2()
    SyncMoveOn sync5, multi_task;
    MoveL F32_L \ID:=20, v50, fine, tool0;
    SyncMoveOff sync6;
    SpeedRefresh vel;
    vel := CSpeedOverride(\CTask);
    TPWrite "CHANGING SPEED (L) TO: " + ValToStr(vel);

    UNDO
        SyncMoveUndo;
ENDPROC
PROC syncmove3()
    SyncMoveOn sync12, multi_task;
    MoveL F43_L \ID:=30, v50, fine, tool0;
    SyncMoveOff sync13;
    SpeedRefresh vel;
    vel := CSpeedOverride(\CTask);
    TPWrite "CHANGING SPEED (L) TO: " + ValToStr(vel);

    UNDO
        SyncMoveUndo;
ENDPROC
PROC syncmove4()
    SyncMoveOn sync16, multi_task;
    MoveL F54_L \ID:=40, v50, fine, tool0;
    SyncMoveOff sync17;
    SpeedRefresh vel;
    vel := CSpeedOverride(\CTask);
    TPWrite "CHANGING SPEED (L) TO: " + ValToStr(vel);

    UNDO
        SyncMoveUndo;
ENDPROC
PROC main()
    !**************************************************
    !Box Assembly
    !**************************************************
    F6 := Offs(F2,0,-50,80);
```

```
F4 := Offs(F2,0,50,0);
F3 := Offs(F2,0,-100,80);
gripperforce := 20;
g_Init \maxSpd:=10 \holdForce:=10 \Calibrate;
g_SetForce 10;
g_SetMaxSpd 10;
MoveJ HomePos, vmax, fine, tool0;
MoveL Offs(F2, 0, 0, 100), vmax, fine, tool0;
g_GripOut;
MoveL F2,v50,fine,tool0;
g_GripIn \holdForce:=gripperforce;
MoveL Offs(F2, 0, 0, 100), vmax, fine, tool0;
WaitSyncTask sync1, multi_task;
MoveJ RelTool(F12_L, 0, -60, 0), vmax, fine, tool0;
syncmove;
WaitSyncTask sync4, multi_task;
g_GripOut;
MoveL RelTool(F12_L, 0, 0, -40), v100, fine, tool0;
WaitSyncTask sync7, multi_task;
MoveJ RelTool(F12_L, 150, 0, 0), vmax, fine, tool0;
MoveL F3,vmax,fine,tool0;
g_GripIn \holdForce:=gripperforce;
MoveL Offs(F3, 0, 0, 50), vmax, fine, tool0;
MoveJ RelTool(F32_L, -50, 0, 0), vmax, fine, tool0;
WaitSyncTask sync8, multi_task;
syncmove2;
g_GripOut;
MoveL RelTool(F32_L, 0, 250, 0), vmax, fine, tool0;
MoveJ Offs(F4, 0, 0, 100), vmax, fine, tool0;
MoveL F4,v50,fine,tool0;
g_GripIn \holdForce:=gripperforce;
WaitSyncTask sync9, multi_task;
MoveL Offs(F4, 0, 0, 200), vmax, fine, tool0;
MoveJ RelTool(F43_L, -200, 0, 0), vmax, fine, tool0;
syncmove3;
!WaitSyncTask sync11, multi_task;
WaitSyncTask sync10, multi_task;
MoveL RelTool(F43_L, -150, 0, 0), v100, fine, tool0;
MoveL RelTool(F43_L, -150, 0, 0 \Rx:= -45), v100, fine, tool0;
MoveL RelTool(F43_L, -150, 0, 0 \Rx:=-45 \Rz:= -120), v100,
fine, tool0;
MoveL RelTool(F43_L, -150, 0, 0 \Rz:= -120), v50, fine, tool0;
WaitSyncTask sync11, multi_task;
MoveJ RelTool(F54_L, 0, 50, 0), vmax, fine, tool0;
syncmove4;
WaitSyncTask sync14, multi_task;
MoveJ RelTool(packing_drop, 0, 0, -180), v100, fine, tool0;
MoveL packing_drop,v50,fine,tool0;
g_GripOut;
MoveL RelTool(packing_drop, 0, 0, -100), vmax, fine, tool0;
WaitSyncTask sync15, multi_task;
MoveJ HomePos, v100, fine, tool0;
WaitSyncTask sync18, multi_task;
!**************************************************
```

```
        !Close box
        !**************************************************
        MoveJ Offs(F6, 0, 0, 70), vmax, fine, tool0;
        MoveL F6, v50, fine, tool0;
        g_GripIn \holdForce:=gripperforce;
        MoveL Offs(F6, 0, 0, 70), vmax, fine, tool0;
        MoveJ Offs(F65_L, 0, 0, 50), vmax, fine, tool0;
        MoveL F65_L, v50, fine, tool0;
        g_GripOut;
        MoveL RelTool(F65_L, 0, 0, -50), v50, fine, tool0;
        MoveL HomePos, vmax, fine, tool0;
    ENDPROC
ENDMODULE
```