Bachelor's Thesis

**Bachelor's degree in Industrial Technology Engineering**

# Safe feeding strategies for a Physically Assistive Robot

**MEMORY**

| | |
|---|---|
| **Author:** | Maria Vila Abad |
| **Advisor:** | Gerard Canal and Guillem Alenyà |
| **Tutor:** | Cecilio Angulo |
| **Call:** | June 2018 |

ETSEIB

Escola Tècnica Superior
d'Enginyeria Industrial de
Barcelona

Institut de Robòtica i
Informàtica Industrial

UPC

# Acknowledgements

I owe my deepest gratitude to Gerard Canal for guiding me through this project and for being always there since the very beginning. Without his enthusiasm, encouragement and support this thesis would hardly have been completed. I express my warmest gratitude Guillem Alenyà for all the useful advices and for giving me this perfect opportunity. I would also like to express my gratitude to Sergi Foix for the numerous explanations and the help provided, and to Cecilio Angulo for his predisposition and attention.

Finally I would like to thank all the volunteers who spent some time participating in the experiments with users and to all the members of the perception and manipulation group for useful discussion and good moments.

# Abstract

With aging societies and the increase of handicapped people the demand for robots that can help nursing humans on-site is increasing. Concretely, according to World Health Organization (WHO) by 2030 more than 2 billion people will need one or more assistive products. With this perspective it becomes vital to develop assistive technology products as they maintain or improve disabled people's functioning and independence. One of the most important activities that a person needs to be able to perform in order to feel independent is self-feeding.

The main objective of this thesis is to develop software that controls a robot in order to feed a disabled person autonomously. Special attention has been given to the safety and naturalness of the task performance. The resulting system has been tested in the Barrett WAM® robot.

In order to fulfill this goal an RGB-D camera has been used to detect the head orientation and the state of the mouth. The first detection has been realized with the OpenFace library whereas the second one has been realized with the OpenPose library. Finally, the depth obtained by the camera has been used to identify and cope with wrong detections.

Safety is an essential part of this thesis as it exists direct contact between the user and the robot. Therefore, the feeding task must be completely safe for the user. In order to achieve this safety two different types of security have been considered: passive safety and active safety. The passive safety is achieved with the compliance of the robot whereas active safety is achieved limiting the maximum force that is obtained with a force sensor. Some experiments have been carried out to determine which is the best setup for the robot to ensure a safe task performance.

The designed system is capable of automatically detecting head orientation and mouth state and decide which action to take at any moment given this information. It is also capable of stopping the robot movement when certain forces are reached, return to the previous position and wait in this position until it is safe to perform that action again.

A set of experiments with healthy users has been carried out to validate the proposed system and the results are presented here.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

**DOF** Degrees Of Freedom . . . . . . . . . . . . . . . . . . . . 17

**EU** European Union . . . . . . . . . . . . . . . . . . . . 14

**FSM** Finite-State Machine . . . . . . . . . . . . . . . . . . . . 32

**HRI** Human-Robot Interaction . . . . . . . . . . . . . . . . . . . . 22

**IRI** Institut de Robòtica i Informàtica Industrial . . . . . . . . . . . . . . . . . . . . 23

**RGB-D camera** camera that gives normal RGB image and its corresponding depth image . . . . . . . . . . . . . . . . . . . . 15

**ROS** Robot Operating System . . . . . . . . . . . . . . . . . . . . 20

**SCI** Spinal Cord Injury . . . . . . . . . . . . . . . . . . . . 19

**WHO** World Health Organization . . . . . . . . . . . . . . . . . . . . 5

# 1. Introduction

According to WHO [19], over a billion people, about 15% of the world's population, has some sort of disability. More concretely, between 110 and 150 million adults have significant difficulties in functioning.

Nowadays, more than one billion people need one or more assistive product but according to WHO by 2030 more than 2 billion people will need one or more assistive products. Rates of disability are increasing mainly due to population aging and increase in chronic health conditions. As an example in Figure 1.1 can be observed the population pyramid comparing European Union (EU) population in 2008 and the 2060 projection made by the Eurostat [2].

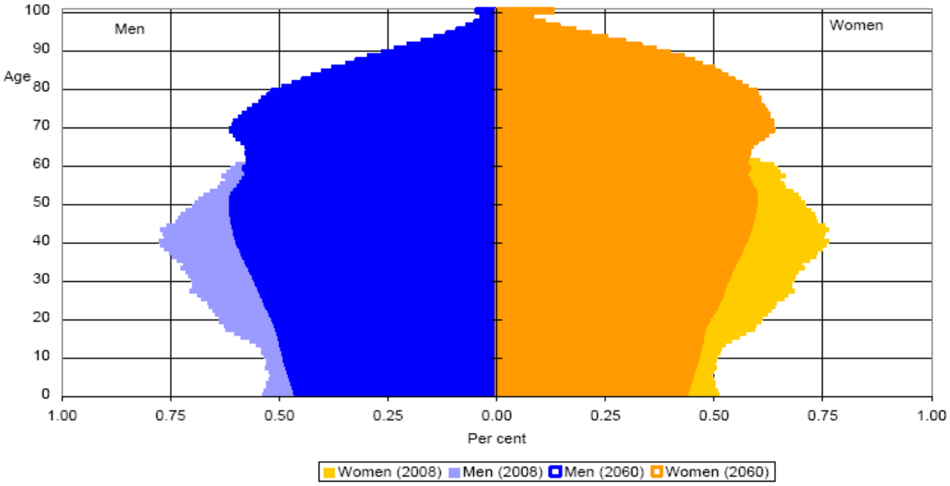


**Figure 1.1:** Population pyramid of EU in 2008 and expected in 2060

With this perspective it becomes vital to develop assistive technology products as they maintain or improve disabled people's functioning and independence. One of the most important activities that a person needs to be able to perform in order to feel independent is self-feeding.

As will be explained in the state of the art, currently there does not exist any robot with sensors to detect the user's state and autonomously feed him in a safe way. Consequently, there exists the need of a robot capable of developing this task. Therefore, the aim of this project is to develop software that controls a robot in order to feed a disabled person autonomously. Concretely, in this project the software will be tested in the Barret WAM® robot but it is compatible with any kind of robot given that it has enough degrees of freedom to perform the feeding action.

A vital importance factor that has to be taken into account is the security of the user while the task is being executed. A review on the current regulation for robots and robotic devices of personal care will be made in order to understand the available standards. The robot's behavior will be designed in order to comply with these standards.

In order to achieve this project's objective two sensors will be used. The first sensor is a camera that gives normal RGB image and its corresponding depth image (RGB-D camera) that will be fixed on the gripper of the robot. The RGB-D camera will be used to obtain information about the state of the mouth and if the person is looking to the robot. The second one is a force sensor that will be fixed on the robot's base of the gripper. It will be used to learn if there exists resistance to the robot's movement.

The trajectories the robot will follow will be learnt from demonstration. Learning from Demonstration means teaching the robot a task repetitively in order for him to learn what it means to perform this task by generalizing from observing several demonstrations.

With this implementations the robot will be able to decide which action to take and when to feed the person based on the sensor's inputs received. The main focus of this project lies on feeding the person in a secure way. Therefore food grasping is out of the scope of this project.

# 2. Objectives

The main goal of this project is to program a robot so it is capable of feeding a disabled person autonomously. The robot will be able to grasp the food (the food needs to have a creamy texture), enter the mouth safely and exit it also in a safe way. The focus of this project will be in the following aspects:

- Detect the face orientation of the user
- Detect whether the mouth is open or not
- Perform all movements safely, specially the ones where the robot can potentially interact with the user. The robot will never be able to impact with the user with a harmful force.
- Perform all the movements in a natural way

Given that the focus of the project is mainly the safety of the user, it is important to remark that the robot will go to a predetermined mouth position so knowing the exact position of the mouth in robot space is not under the scope of this project. Neither is under the scope of this project to know the location of the food or its state. So the robot will move to a predetermined food position and will not stop to feed the user once the food has been finished.

# 3. Related work

With aging societies and the increase of handicapped people the demand for robots that can help nursing humans on-site is increasing. For this purpose many robots have been proposed. They can be divided into the following groups:

- **Manually operated eating systems:** with these systems the user has to approach the spoon to the desired position but the system offers a stabilizer to help those who have difficulties moving accurately. On example is the Neater eater [10]. It offers a spoon with an arm that has to be fixed to the table so the user just has to move the arm to approach the spoon to the food and to his mouth.

- **Forearms stabilizers:** these systems consist on an arm support. The user's arm is partially or completely fixed to the support and it helps reduce the vibration of the person stabilizing the arm movement. Some examples are the Jaeco arm support [5] or the Mas arm support [9].

- **Electrically operated eating systems:** these systems offer a spoon fixed to an arm that can move autonomously. Part of this system also have a dish attached so the arm is capable to grasp the food autonomously. Some examples are the Bestic arm [18], My spoon [15] or Mealbuddy [7].

In order to be able to use manually operated systems or forearms stabilizers the user needs to have at least partial control of the upper limbs. Therefore, a considerable part of the population that needs feeding assistance is not able to feed with these systems as they can not be able to move their arms or are not able to move them as properly as needed. As the objective of this project is to be able to feed a high percentage of the population needing feeding assistance, only electrically operated eating systems have been taken into account.

Self-feeding assistant robots have been developed with commercial and research objectives. In table 3.1 some existing electrically operated meal assistant robots are presented. This table also shows the Degrees Of Freedom (DOF) each robot has and the type of the input each robot has in order to be controlled.

The most used piece of cutlery is the spoon as for example it is the case of [15] or [7] because it is safer than the fork. However there are some meal assistant robots as [6] that have a gripper thus they are also able to hold a bottle or glass.

An extremely important factor for the meal assistant robot is to have the capability of entering the mouth, tilting the spoon and exiting the mouth autonomously. This is due to the fact that depending on the degree of disability the person may not be able to do it autonomously or it can be extremely challenging for them to reach the spoon or unload the food. There are few prototypes that offer these capabilities. To be able to offer these capabilities in a completely autonomous manner the face has to be tracked because the position of the mouth has to be known as well as its state. Some examples of robots that offer this capabilities are [11] or [12].

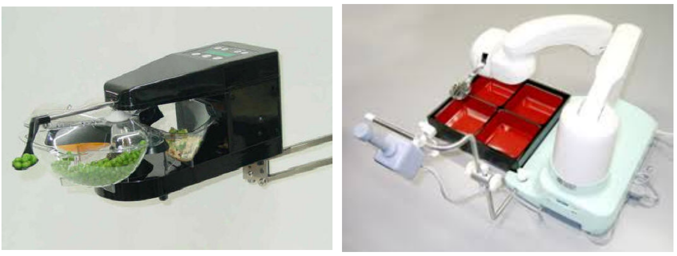
| Feeder robot | DOF | Research/Commercial | Input type |
|---|---|---|---|
| My Spoon$^{TM}$ [15] | 5 | Commercial | Keyboard/joystick |
| Bestic® Arm [18] | 4 | Commercial | Keyboard/joystick |
| Meal Buddy [7] | 4 | Commercial | Keyboard/joystick |
| Mealtime Partner [8] | 2 | Commercial | Keyboard/joystick |
| The voice bot [3] | 4 | Research | Voice commands |
| Meal-Assistance Robot by Yamaguchi University [16] | 2 | Research | Eye Interface |
| ASIBOT [6] | 5 | Research | Keyboard/joystick |
| Georgia Institute of Technology [12] | 7 | Research | GUI/Kinect |

**Table 3.1:** Comparison of existing self-feeding robots

A key factor is the control method used. As seen in table 3.1 the majority of self-feeding robots use a keyboard or joystick control. For example [15] uses a chin operated joystick whereas [7] or [8] use hand operated buttons. These methods offer a safer task development as the user is the one controlling the robot movement and there is little possibility of software errors. On the other hand, these methods can not be used by a subgroup of people as they requires a mechanical actioning. There exist other control methods that do not require mechanical actioning as for example voice commands like in [3] or eye tracking as in [16]. These methods allow the vast majority of people to use these devices but they are not as safe as a mechanical actioning as there can occur detection problems.

The most recent prototypes of feeding assistant robot offer a more automatic process. This is achieved by tracking the person's face, specially his mouth. For example in [12] face tracking is performed using a Kinect camera and ARtags. This presents the problem that the person should wear an ARtag on its forehead and it does not detect the state of the mouth so there is still the need to have a manual control to tell the robot when to start the feeding action. A more complete type of tracking and controlling is the one presented in [11]. In this prototype there are 5 types of inputs: a camera, a microphone, a current sensor, a force sensor and a joint encoder. This allows the robot to know the mouth position and detect various types of anomalies produced during the feeding task. However, there is no information about the state of the mouth. The prototype presented in [13] solves this problem as it includes mouth tracking and mouth state detection. This prototype also offers control through Electroencephalography to control the user's intentions. However, this prototype does not support any anomalies control.

**(a)** Bestic arm **(b)** Mealbuddy



**(c)** Mealtime partner **(d)** My Spoon

**Figure 3.1:** Different state of the art self-feeding robots

A subgroup of the elderly or handicapped population as for example people with Spinal Cord Injury (SCI) level C4 and above can not move the neck or they can not move it easily. As a result the vast majority of robots discussed above are not able to feed these people. There is just one prototype that offers face tracking and mouth state control but it does not have any anomalies control so it does not offer a safe performance. Therefore there exist the not covered need of a completely safe self-feeding robot that do not require any type of mechanical actioning.

# 4. Resources

In order to accomplish this project's objective some hardware and software resources will be needed. The most important hardware resource is the robot itself: in this case the Barrett WAM® robot. The other hardware resources used are an RGB-D camera and a force sensor. The most important software resources used are Robot Operating System (ROS) and C++. In this chapter all the relevant hardware and software resources are presented.

## 4.1 Robot Operating System

The ROS is a set of software libraries and tools that help build robot applications. Therefore, it offers a message passing interface that provides inter-process communication. Groups of ROS-based processes that are running and connected to the same network are represented in a graph where each process is a node and messages between processes are represented as edges. Messages in ROS have a specific type that is defined by the programmer. The type of an edge of the graph mentioned before is given by the type of the messages that are sent through that edge. The node sending messages in an edge and the node receiving them in the same edge must use the same type of messages. There are three types of communications between nodes (a communication would be an edge in the graph mentioned previously):

1. **Service:** A service is a type of communication that allow nodes to send requests and receive responses. In a service there is always a node offering the service (which is called the server) and other nodes that will send requests to it. When a request is received by the server it will compute the response and send it back to the node requesting it. Service calls are blocking which means that the caller will block until the server returns a response.

2. **Topic:** A topic is a type of communication that allows nodes to send and receive messages continuously. In a topic there is always one node (known as the publisher) that publishes messages in the topic and one or more nodes that are subscribed to the topic (known as the subscribers). The subscribers will receive all the messages published in the topic.

3. **Action:** An action is a type of communication that allows two nodes to send and receive messages continuously with feedback information. In an action there are always two nodes: the action client and the action server. Between these two nodes there are five topics: goal, cancel, status, result and feedback. The action

client publishes in the goal and cancel topics and is subscribed to the the status, result and feedback topics; the opposite happens with the action server. When the action client publishes in the goal topic the action server starts working towards that goal. While doing it, the action server is publishing status and feedback messages so the action client knows its state. If the action client wishes to cancel the goal it has to publish in the cancel topic. Finally, when the action server has completed the goal it publishes in the result topic to let it know to the action client.

With this communications it is possible to have a wide variety of nodes which allows to have a considerable degree of modularity.

Finally ROS offers a wide variety of tools to support introspecting, debugging, plotting and visualizing the state of the system being developed. The most known tools are rviz and rqt. Rviz is a tool that provides three dimensional visualization of the robot and many sensor data types which allows the user to be able to identify a wide variety of problems, as well as to know the robot's state in real time. Rqt is a Qt-based framework for developing graphical interfaces for the robot. The most common rqt plugins are rqt_graph (which shows the graph of your current system) and rqt_plot (which allows to plot anything that can be represented as a number and varies over time).

## 4.2 Programming language

Nodes in ROS can be written in a variety of programming languages but usually they are written in Python or C++. It is important to remark that nodes in the same network do not need to be written in the same programming language.

Python is an interpreted programming language whereas C++ is a compiled one. In a compiled language, programs are translated once running the source code through a compiler. This results in a very efficient code because when it is executed it will only load and execute. Oppositely, when an interpreted language program is executed it must be parsed, interpreted, and executed. For this reason, interpreted programs are usually less efficient than compiled programs.

The programming language chosen for this project is C++ because it contributes to have a more efficient program.

## 4.3 Barrett WAM® robot

The feeding task is performed by the Barret WAM® robot shown in Figure 4.3. It is a 7 DOF robot as can be seen in Figure 4.2. It has a generally spherical workspace of 2 meters in diameter as shown in Figure 4.1. The joint ranges exceed those for conventional robotic arms.

The high number of degrees of freedom allows the robot to reach any point of the workspace with the end plate in any orientation. All joints have human-like kinematics and it is possible to have different degrees of stiffness. All this aspects ease the Human-Robot Interaction (HRI), specifically when it involves physical contact as in the case of the feeding task.

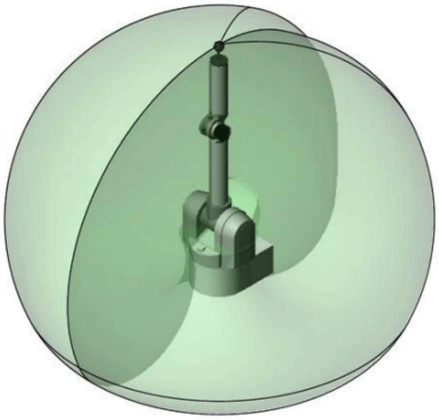


**Figure 4.1:** Isometric view of the Barret WAM workspace

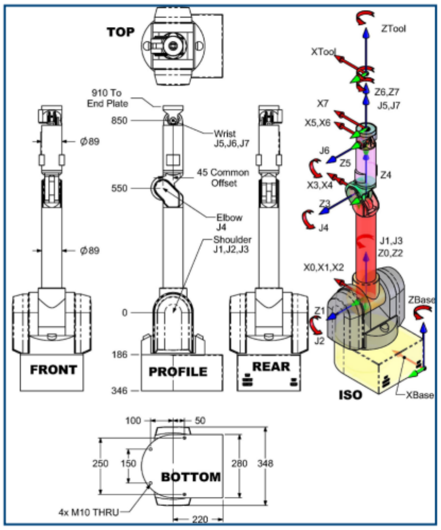


**Figure 4.2:** Barret WAM robot schematics

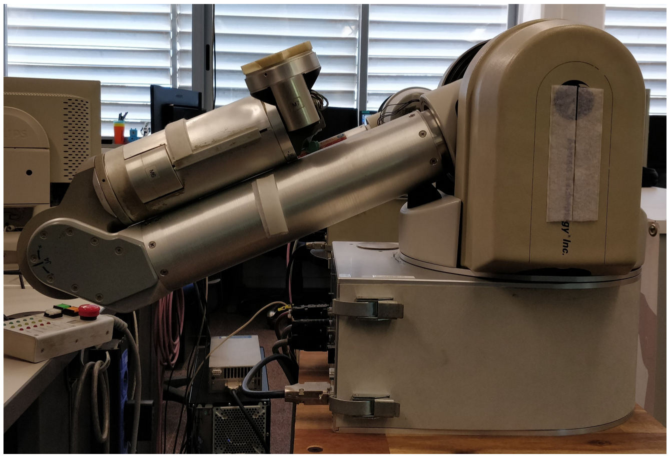


**Figure 4.3:** Barret WAM robot at IRI

The Barret WAM® robot is capable of lifting 3kg of load. As we will only need to lift a gripper, the camera used, a spoon and the food on the spoon the specifications are adequate. It has a maximum endtip velocity of 3m/s which is sufficient for our task.

Moreover, the WAM robot has the controller shown in Figure 4.4. It is of vital importance as it allows to switch off the WAM at any moment offering an emergency shutdown of the robot. It offers two types of safety switch off:

- Pushing the red button: the WAM will be immediately shutdown cutting off all power so it will fall by its own weight
- Pushing Shift+Idle: the WAM's power will be gradually decremented so it will slowly fall

In addition, it is used to start the WAM avoiding any involuntary switch on. Therefore the controller is in charge of allowing the robot to switch on and off.

**Figure 4.4:** WAM controller

More information about the specifications of the robot can be found in its webpage [17].

The WAM requires a gripper that needs to be attached to the last joint in order to be able to grasp the spoon. A custom-made 3D printed gripper shown in Figure 4.5 was developed in the Institut de Robòtica i Informàtica Industrial (IRI) laboratories for the feeding task.

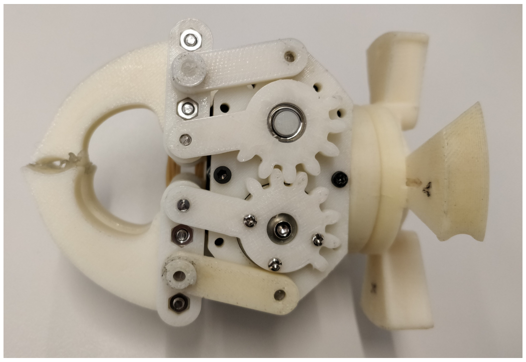


**Figure 4.5:** 3D printed gripper

## 4.4 Camera

In order to perform an accurate face recognition we need to use a camera that gives normal RGB image and its corresponding depth image (RGB-D camera). There exist many RGB-D cameras like the Microsoft Kinect$^{TM}$ One, the ASUS Xtion Pro and the Creative Senz3D$^{TM}$. The first two cameras have a larger depth detection range but have a lower precision on closer objects. On the other hand, the third camera has a smaller detection range but has a better depth detection on closer objects. We plan to install the camera close to the mouth in order to avoid camera visibility obstructions by the WAM. For this reason the best camera is the Creative Senz3D$^{TM}$ camera shown in Figure 4.6. Moreover it is small enough to be fixed on the gripper and to be able to move it naturally.

**Figure 4.6:** Creative Senz3D camera

The Creative Senz3D$^{TM}$ uses the time-of-flight technology to obtain the depth image. This technology sends light signals that are reflected by the objects in the field of view and the camera lenses gather the reflected light. Knowing the speed of light and the delay of the light signals (or time-of-flight), the distance traveled is calculated. With this technique edges are sharper than with other techniques and it provides images quicker, although images have a lower resolution. More specifically, 320x240 is the maximum depth image size in the Creative Senz3D$^{TM}$.

## 4.5 Force sensor

During the feeding task there exists the need to have information about the resistance against the robot's movement. This information consists on the force and torque that the end plate receives in the three axis. Depending on their values it is possible to evaluate if the robot must be stopped for security reasons or if it is safe to continue the movement. In order to obtain this information a force sensor is needed.

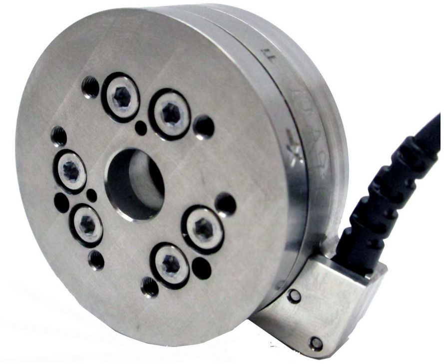


**Figure 4.7:** ATI mini40 f/t sensor

The force sensor chosen in this project is the ATI$^{TM}$ mini40 f/t sensor which can be observed in Figure 4.7. It is a small sensor tan can be perfectly attached between the WAMs end plate and the base of the gripper. It is a 6-DOF sensor which means that it provides force and torque data for the three axis.

The limiting forces in the feeding task are very low, concretely between -7.1N and 4.5N in the Y axis, so there exists the need of a high resolution sensor. Concretely, the ATI sensor offers a near-zero noise distortion and a resolution of 1/200N in the Y axis. Moreover it has a sensing range from -20N to 20N in the Y axis which is ideal for our application. More information about the sensors specifications can be found in [1].

# 5. Regulation for personal care robots

Personal care robots are a new technology expected to increment the quality of our life in the foreseeable future. However, unlike industrial robots, they require special safety measures as they are going to be in direct contact with their human users. Therefore, institutions are increasingly focusing on the legal challenges proposed by the robotics sector. Currently, there exist an international regulation that proposes the safety standards for personal care robots: ISO 13482:2014.

## 5.1 ISO 13482:2014

ISO 13482:2014 [4] is an international regulation that states the security standards for robots and robotic devices in personal care. A robot is a driven and programmable mechanism of two or more axis with a certain degree of autonomy (within his surroundings) whose objective is to perform planned tasks. A service robot is a robot that conducts beneficial tasks for humans or devices excluding industrial automation applications. A personal care robot is a service robot that performs actions that directly contribute to the improvement of quality life of human beings (excluding medical applications). The aim of this regulation is to specify the contact conditions between human and robot. Particularly, it specifies requirements and guidelines for the inherently safe design, protective measures, and information for the usage of personal care robots. The following three types of personal care robots are taken into account in this regulation:

1. **Mobile servant robot:** Personal assistant robot that is capable of travelling in order to carry out tasks that involve interaction with human beings like object manipulation or information exchange.

2. **Physical assistant robot:** Personal assistant robot that physically assists a user in order to carry out required task providing a complement or an increase of the personal abilities.

3. **Person carrier robot:** Personal assistant robot whose purpose is to transport human beings to a planned destiny.

In this project a physical assistant robot is implemented. In particular it is a not-restricted physical assistant robot as it is not fixed to a human when it is being operated.

### 5.1.1 Verification and validation procedures

Each danger has to be verified and validated with a specific procedure. The possible verification and validations procedures are listed below.

- **A (Inspection):** Inspection of the state of the personal assistance robot or the equipment and structures using human senses without any specialized inspection equipment. When the robot is not operating inspection is usually done visually or acoustically.

- **B (Practice test):** Verification of the state of the personal assistance robot and its equipment in normal and anomalous conditions. For example: functional tests, cyclic tests or performance tests.

- **C (Measurement):** Comparison between the real values of the properties of the personal assistance robot and its specified limits.

- **D (Inspection during operation):** Inspection (like method A) of the functionalities of the personal assistance robot or its equipment when it is operating in normal or anomalous conditions.

- **F (Program examination):** Structured revision or inspection through the software code design and the related specifications. A code inspection or verification of the software code should follow this method.

- **G (Revision of the risk evaluation based in tasks):** Structured revision or inspection through risk analysis, risk estimation and corresponding documentation.

- **H (Design schemes examination and its corresponding documents):** Structured revision and inspection through the design schemes and its corresponding documents.

### 5.1.2 Significant dangers of the personal assistant robots

There are many dangers of this regulation that refer to the robot executing the feeding task. Many of them refer to the robot being secure in any assistant task and those have not been taken into account as the robot used is considered to comply with the established regulation. The dangers that have been deeply studied are the ones that are more related to the task in hands.

#### 5.1.2.1 Stress dangers due to usage and postures

This problem, when related to the feeding task, includes the stressing posture of the person required to operate the robot. If not solved it can cause musculoskeletal disorders.

In order to prevent this problem ergonomic principles described in the ISO 14738 have to be applied. In order to have a secure design the person have to be able to maintain a good posture when operating the robot and be able to use all robots commands without stress (usually this is achieved by not directly connecting them to the robot). Other complementary measures are the usage of shock absorption mechanisms and posture supports.

The instruction manual has to include information about the correct way of operating the robots control commands and the robot itself.

In order to verify and validate this problem one or more of the following methods can be used: A, C, D or H.

#### 5.1.2.2 Dangerous physical contact during the human-robot interaction

This problem, when related to the feeding task, includes object detection failure of objects related to the security inside the operating space. If this problem is not solved when the robot is operated it can cause collisions with objects related to the security. This danger takes into account harmful levels of physical reaction during the tactile interaction. If this problem is not solved it can cause cuts, amputations, crushing or entrapment. Finally this danger also includes tactile interactions with robot components not planned for tactile interaction. If not solved it can cause injuries made by blunt objects, entrapment or crushing.

In order to prevent this problem first of all one has to identify the functions that guarantee the person's security during the tactile interaction. When doing this process the following aspects have to be considered:

- Detection of people in the space reachable by the robot.
- During the tactile interaction the physical reaction between the robot and the person has to be planned to be as low as possible.
- The design has to be made in order to avoid physical interactions between the person and parts of the robot not designed for tactile interaction.

The design has to be made in order to decrease: the friction between robot and skin, the shear force, dynamic collisions, torques, center of gravity arches, weight transfer and supports on people. The following complementary measures can also be taken into account:

- Software limits to mark the robots area of operation in order to restrict the movement of the robot to a defined volume or to avoid the robot entering a certain volume.

- A limit or various limits to the speed have to be fixed in order to avoid any harm to human beings. Only authorized people can change those limits. The robot must always move in a speed below the fixed limit.

- Quantitative force or secure contact torque limits have to be analyzed through ergonomic experimentation. Force control must be obtained through a contact sensor. If there is an undesired contact between the person and the robot the following requirements have to be fulfilled:
  - A fast enough answer must exist in order to always maintain the limits below the security force limit.
  - Bring the robot to a secure state after the accidental contact.

The instruction manual has to include information about the tasks where tactile interaction is foreseen and the limitation of groups of users, environmental conditions, etc. It also has to include instructions on how to operate the robot in order to avoid injuries and warnings of possible injuries if the instructions are not followed properly.

In order to verify and validate this problem one or more of the following methods can be used: C, D F or G.

#### 5.1.2.3 Dangerous autonomous actions

This problem, when related to the feeding task, includes the dangerous actions during the development of any autonomous task. If not solved it can cause too many problems.

In order to solve this problem the robot has to be designed so he makes the correct decision at any given situation and in case an erroneous decision is made the robot can not cause any unacceptable harm. The damage caused by incorrect decisions can be decreased by increasing the reliability of the decision or limiting the effect of an inaccurate decision.

The design has to satisfy that the operating scenarios where there exist a high risk of suffering damage due to an incorrect action are restricted. In addition unique identifiers have to be used for the objects related with the security, movement routes, ... The following complementary methods can also be used:

- Increase the reliability of the sensors and algorithms to a level where unacceptable dangers will not appear

- Algorithms capable of calculating and supervising the probability of the correctness of a decision. Decisions with high uncertainty have to be reevaluated using alternative approaches or additional information. If after the reevaluation there still exists unacceptable uncertainty external aid has to be searched or a security stop has to be performed

- Decisions should be verified through different detection rules

The instruction manual has to include usage limits that must exclude situations where decisions could cause a risk of unacceptable harm considering previsible bad usages. The usage information should also inform about the detection and the decision taking competence of the robot. Finally, the instruction manual should incorporate instructions on how to avoids damages due to inaccurate actions and decisions.

In order to verify and validate this problem one or more of the following methods can be used: B, C, D F or G.

# 6. Feeding strategies

This chapter is devoted to explain the feeding strategies developed in this thesis. It will begin with a discussion of the best spoon positioning. Then it will explain the robot behavior chosen, followed by the explanation of its implementation. Finally it will discuss if the behavior presented complies with the regulation explained in chapter 5.

## 6.1 Positioning of the spoon

The positioning of the spoon relative to the gripper is a key factor because it can ease the task and make it more natural or, on the other hand, make it difficult to reach all corners of the dish or be more invasive for the person that is going to be fed. Three different positioning of the spoon were considered.

The first position considered is the one that has a 180 degrees angle between the hand and the spoon handle as seen in Figure 6.1(a). Although in this position it is easy to reach all corners of the dish and the hand almost never contacts the side of the bowl, it is more invasive for the person because when the hand is approximating the mouth the person is going to see all the robot moving towards him. Another factor that has to be considered is if it seems natural to see the robot moving with this positioning; in this case it seems natural but not entirely.

The second position considered is the one that has 90 degrees angle between the hand and the spoon handle as seen in Figure 6.1(b). Despite that fact that in this position the movement of the robot seems more natural, it is harder to reach certain corners of the dish and the hand frequently contacts the side of the bowl which can cause the overturning of the dish. However, it is a less invasive approach of the spoon to the mouth.

Finally the third position considered is with an intermediate angle, specifically, a 135 degrees angle between the hand and the spoon handle as seen in Figure 6.1(c). This positioning is less invasive for the person than the 180 degrees option but more invasive than the 90 degrees option. With this positioning the hand sometimes contacts the side of the bowl but not as many times as with the 90 degrees option. It is hard to reach certain corners of the dish but not as hard as with the 90 degrees option. Finally it is difficult to move the robot naturally with this positioning.

Between the facts that differentiate the different positioning of the spoon, the most important one is the comfort of the user. This is due to the fact that the robot will only be used if the user is comfortable with it. That is why the key aspects are the naturality of movement and the invasivity of the robot. The 135 degrees option can seem the best one because it has a good balance between comfort for the robot to reach and obtain

**(a)** 180 degrees **(b)** 90 degrees **(c)** 135 degrees

**Figure 6.1:** Spoon positioning relative to the gripper

the food and invasivity of the robot. Nevertheless the movements of the robot with this positioning don't seem as natural as what it is desired. For those reasons the positioning chosen is the one that has 90 degrees between the hand and the spoon handle.

However, with this positioning we have some problems to reach and obtain the food. In particular it is problematic to reach certain parts of the bowl and the gripper frequently touches the edge of the bowl. The main causes of these problems are that the end of the spoon is not far enough from the end of the gripper and the spoon handle is excessively wide. Both causes can be solved with a longer spoon.

## 6.2 Robot's behavior

The robot behavior is controlled through a Finite-State Machine (FSM). A FSM is a model that can store the current state (the current status of a system) and can change it if certain conditions are fulfilled. In order to change from one state to another one a transition is activated. A transition is a set of actions that are executed when certain conditions are fulfilled or when an event is received.

In this application when the robot is in a state it stays in a determined position. On the other hand, when the robot is executing a transition it performs a defined operation. In this particular case transitions are activated through six boolean variables and there are four states and nine transitions which are listed below. The FSM controlling the robot behavior can be observed in Figure 6.2.

### 6.2.1 States

- **Home:** when the robot is in this state it is in its initial position so it is completely folded and resting on the support.
- **Position 1:** when the robot is in this state it has de gripper more or less 30 cm above the dish containing the food.

- **Position 2:** when the robot is in this state it has the end of the spoon more or less 5cm in front of the mouth of the user. The spoon is completely horizontal and loaded with food.
- **Mouth:** when the robot is in this state the spoon is inside the mouth of the user.

### 6.2.2 Transitions

1. **Home to position 1:** this transition moves the robot from home to position 1
2. **Get food:** this transition is in charge of filling the spoon with food. In order to accomplish it the robot starts at position 1, gets the food from the dish and returns to position 1. When this transition is finished the spoon full variable changes its value from false to true
3. **Position 1 to position 2:** this transition moves the robot from position 1 to position 2
4. **Enter mouth:** this transition is in charge of entering the spoon inside the user's mouth. It starts in position 2 and ends when the spoon is inside the mouth. When this transition is finished the spoon full variable changes its value from true to false
5. **Wait gravity:** when this transition is executed the robot waits in gravity mode for 0.5 seconds. When the robot is in gravity mode it stays in the same position compensating the gravity force. However if it is moved it follows the movement naturally without opposing it with a reaction force. In this mode the robot is completely safe
6. **Exit mouth:** this transition moves the robot from inside the mouth to position 2
7. **Exit mouth 2:** this transitions moves the robot from the current position to position 2
8. **Position 2 to position 1:** this transition moves the robot from position 2 to position 1
9. **Throw food:** this transition is in charge of emptying the spoon so when it finishes the spoon full variable changes its value from true to false. To achieve its objective it starts in position 1 then it rotates the spoon 100 degrees and it goes back to position 1
10. **Position 1 to home:** this transition moves the robot from position 1 to home

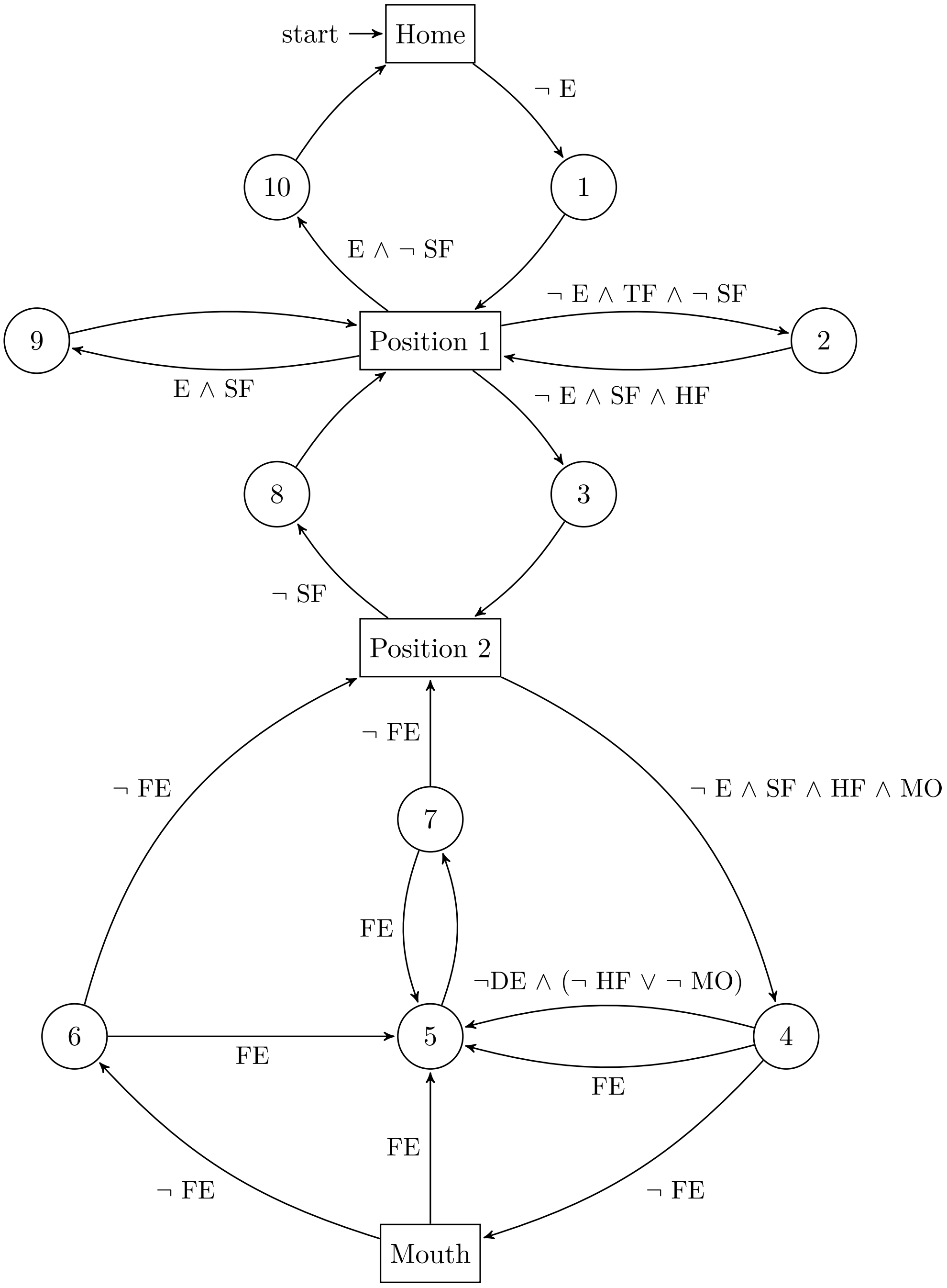


**Figure 6.2:** Robot's behavior state machine

### 6.2.3 Variables

- **E (Exit):** if this variable is true the robot is going to end the feeding procedure. Its default value is false
- **TF (Timer finished):** when this variable is true the 15 seconds timer has finished. This timer is started every time the robot finishes transition 7. Its default value is true
- **SF (Spoon full):** when this variable is true the spoon is loaded with food. Its default value is false
- **HF (Head forward:)** this variable indicates if the user is looking to the gripper. Its default value is false
- **MO (Mouth open):** this variable is true when the user has the mouth open. Its default value is false
- **FE (Force exceeded):** this variable indicates if the limiting force has been exceeded. Its default value is true
- **DE (Depth exceeded):** this variable indicates if the limiting depth has been exceeded. When the limiting depth is exceeded it is not possible to claim the veracity of the MO and HF variables.

## 6.3 ROS implementation

This section is devoted to explain the code that has been developed for this project and the ROS nodes and libraries that have been used. In order to achieve the desired robot behavior several ROS nodes are needed. Some of them had already been developed at IRI, some of them are open-source nodes that can be downloaded on-line and the rest of them have been developed for this project. Aside from these ROS nodes two face detection libraries have been used to decide whether the mouth is open or not and the head orientation. Furthermore, a C++ library has been implemented to decide which action has to be executed at every moment.

### 6.3.1 Face detection

Face detection is an essential part of this project as it provides information about the head orientation and the state of the mouth. As described in section 5.2 this information is needed for the FSM.

#### 6.3.1.1 Facial landmarks

Facial landmarks are defined points of the face as can be seen in Figure 6.3. Concretely there are 68 facial landmarks among which 20 belong to the mouth. Face detection libraries are based on detecting facial landmarks.

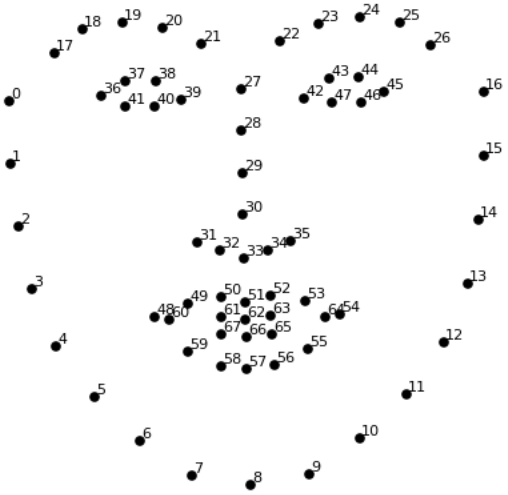


**Figure 6.3:** Facial landmarks

In order to decide if the mouth is open or closed these facial landmarks have been used. The mouth state has been decided with a comparison between the lip size and the space between the lips. Being key_point_x.y the key point of the landmark x in the y axis of the picture we obtain the following variables:

$$LSS = (key\_point\_50.y + key\_point\_51.y + key\_point\_52.y)/3$$

$$LSI = (key\_point\_61.y + key\_point\_62.y + key\_point\_63.y)/3$$

$$LIS = (key\_point\_65.y + key\_point\_66.y + key\_point\_67.y)/3$$

$$LII = (key\_point\_56.y + key\_point\_57.y + key\_point\_58.y)/3$$

LSS is the average position of the top part of the user's superior lip in the y axis whereas LSI is the average position of the bottom part of the user's superior lip in the y axis. In the same way, LIS is the average position of the top part of the user's inferior lip in the y axis whereas LII is the average y position of the bottom part of the user's inferior lip in the y axis.

We decide that the mouth is open enough for the robot to introduce the spoon without causing any damage if the sums of the lips width is 1.4 times bigger than the space between the lips. This value was found empirically as it gives a good performance. Therefore the mouth is open enough if:

$$((LSI - LSS) + (LII - LIS)) * 1.4 > (LIS - LSI)$$

It is important to comment that the mouth detection is only considered successful if the 20 mouth landmarks have and average success probability higher than 50%.

#### 6.3.1.2 Head orientation

The head orientation is represented through the Euler angles as can be observed in Figure 6.4. For the feeding task it is important to know whether the person can look at the spoon or not (either directly or sideways). This is due to the fact that the spoon will go from position 1 to position 2 only if the person can look at the spoon but there is no need for the person to be exactly oriented towards the gripper. This is important because feeding requires the user's attention, thus if the user is not looking at the camera it is not safe to perform the feeding task.

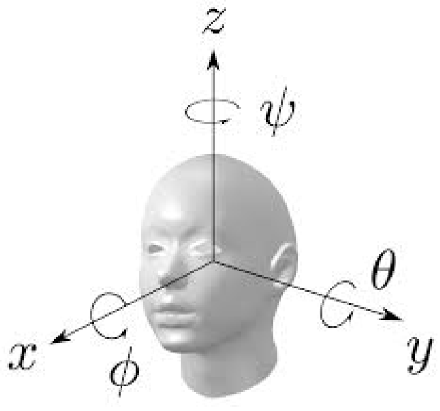


**Figure 6.4:** Head Euler angles

With the Euler angles it is possible to determine if the person can look at the gripper or not as there are some angles from which the person can not see the gripper. The limiting angles have been obtained empirically and can be observed in table 6.1.

| | Minimum value [rad] | Maximum value [rad] |
|---|---|---|
| $\phi$ | -1.35 | 0 |
| $\theta$ | -0.5 | 0.5 |
| $\psi$ | -0.6 | 0.6 |

**Table 6.1:** Limiting Euler angles

#### 6.3.1.3 Face recognition libraries

For this particular task two face detection libraries were taken into account. What makes them stand out in front of other face detection libraries is their high accuracy rate. Furthermore these libraries are freely available for free non-commercial use. The two face detection libraries that have been considered for this project are the following:

- **OpenFace:** OpenFace is the first toolkit capable of face landmark detection, head pose estimation, facial action unit recognition and eye-gaze estimation. It uses convolutional neural networks and computer vision algorithms as explained in [20] to solve the tasks mentioned above. Moreover, this tool is capable of real-time performance.

  For this project face landmark detection is needed as it is possible to extract from it whether the mouth is open or not. This library also offers the head pose estimation tool. In conclusion, it offers all the information that is needed from the face detection and with a good detected frames per second rate.

- **OpenPose:** OpenPose represents the first real-time multi-person system to jointly detect human body, hand and facial landmarks. The face landmark detection is produced using multiview bootstrapping as explained in [14]. Furthermore, this library is capable of real-time performance. It offers two versions: one that uses the GPU and one that uses the CPU and is claimed to have a better accuracy but has a lower frame rate.

  As it has been mentioned before for this project face landmark detection is needed as it is possible to extract from it whether the mouth is open or not. The problem of this library is that it does not offer head orientation which is also needed for this project.

OpenFace was the first library tested as it offers all the tools needed for this project and does not require any additional hardware. Therefore if this library would have had enough accuracy it would have been enough to perform all the face detection tasks.

After applying the transformation from facial landmarks to mouth state explained above, the accuracy of the mouth landmarking was not good enough to suit the project's needs. Concretely, the upper lip is always detected properly but the lower lip is not always accurately detected. In order to decide whether the results obtained with this library are acceptable or not, the library has been tested with three videos manually labeled and the results shown in Figure 6.5 have been obtained.

As can be observed in Figure 6.5 OpenFace with the mouth state transformation has an 8.1% rate of not detecting the mouth when it is open and a 8.4% rate of not detecting it when it is closed. The not detection rate is not negligible but does not represent a problem as when the mouth is not detected it is computed as if the mouth was closed not incurring in false positives. The problem that can be clearly observed is

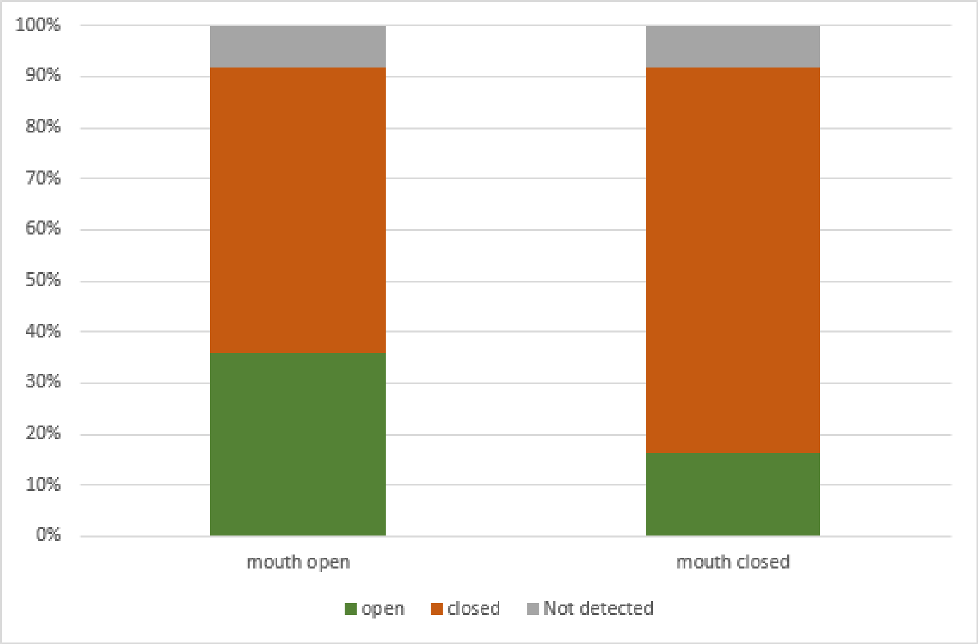


**Figure 6.5:** OpenFace accuracy in mouth state detection

that there is a false positive rate of 16.5%. Deciding that the mouth is open when it is actually closed entails an extremely high risk because the robot will try to feed the person and so the spoon will impact with the user's mouth. Finally, another problem can be observed: the high rate of false negatives, concretely a 56%. In this application a low rate of false negatives would not be a problem as the robot would just wait for the mouth to open which would not cause any harm to the user. However, given the high rate of false negatives the robot would spend a lot of time waiting for the person to open it's mouth and so the user has to wait too much time. This could end up with fatigue and the user not using the product.

For this reasons facial landmark detection of OpenFace is not accurate enough for this application. Thus, it will not be used.

The next tested tool is the head orientation tool of OpenFace. After applying the head looking transformation the results obtained are satisfactory and so this tool can be used.

There is still a need for a facial landmark detection. The next tool tested is OpenPose with the CPU version as it offers a better accuracy and it does not require additional hardware. After applying the transformation from facial landmarks to mouth state explained above, the frame rate is observed to be too low, of 0.5 concretely. This frame rate is not acceptable as in two seconds a person can open and close the mouth several times which transforms the result obtained in not relevant.

The next tool tested is OpenPose with the GPU version. This requires an NVIDIA graphics card with at least 1.6GB available. After applying the transformation from facial landmarks to mouth state explained above, a better result is observed.

In order to test the OpenPose library and to be able to compare the results with the ones obtained with OpenFace the same experiment has been performed. Concretely, this library has been tested with the same videos than the ones used to test OpenFace. The results obtained can be observed in Figure 6.6.

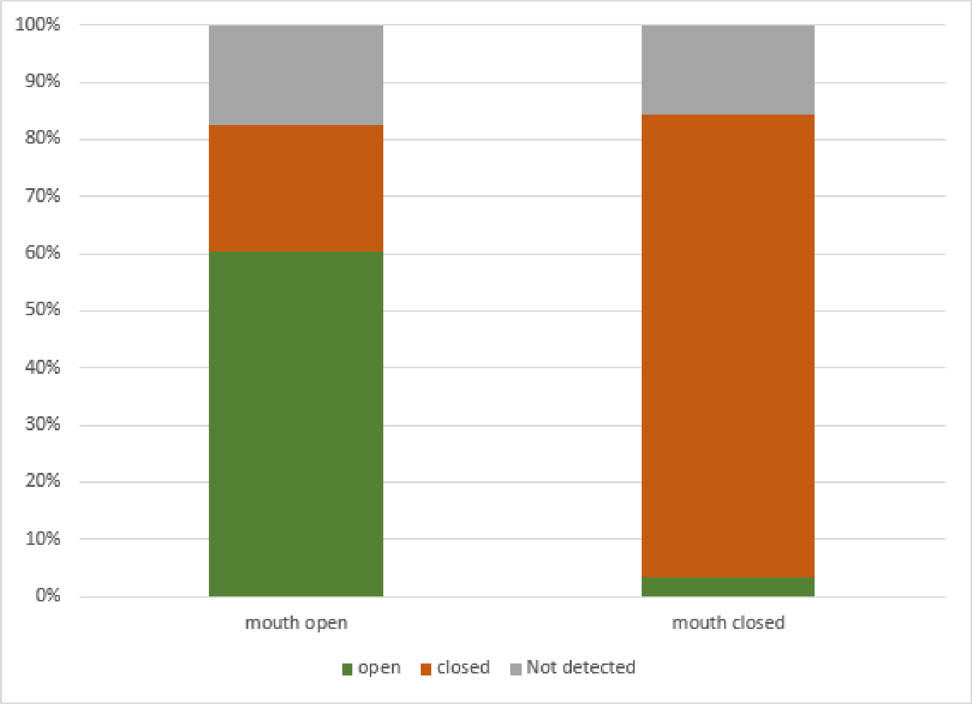


**Figure 6.6:** OpenPose accuracy in mouth state detection

We can observe that OpenPose has only a 3.23% of false positives which is a great improvement from the results obtained with OpenFace. It also has a lower rate of false negatives, concretely a 22.13%. On the other hand, OpenPose has a higher rate of not detecting the mouth both in the case where the mouth is open and the case where the mouth is closed. In this project not detecting the mouth's state is treated as if the mouth was closed. This way is chosen as it presents a safer approach.

The highest risk derived from mouth detection are false positives as they cause an impact of the spoon with the user's face. The rate of false positives is very low if OpenPose is used. False negatives do not present a risk but they may cause the user to stop using the robot because of the high amount of waiting time. With OpenPose the number of false negatives is not negligible but it is lower than with OpenFace. In conclusion it is safe to use OpenPose to detect the mouth state. However, the number of false positives is not 0 so there exists the need to ensure that if an impact is produced it will not be harmful for the user.

#### 6.3.1.4 Depth camera

After some testing with the face detection libraries it has been observed that when the camera is close to the face the accuracy rate drops. Therefore it remains clear that at a certain distance the results obtained with these libraries are not reliable. This is where the depth sensor of the RGB-D camera comes in handy.

After some trial and error a limit from which the face detection libraries are not reliable has been set. Concretely, when the camera is at 40cm or closer to the face, the results are not reliable enough.

#### 6.3.1.5 ROS nodes

In order implement all the previous aspects in the ROS framework some ROS nodes have been developed. These nodes send and receive information from each other using some topics. In Figure 6.7 can be observed the ROS nodes and the topics used.

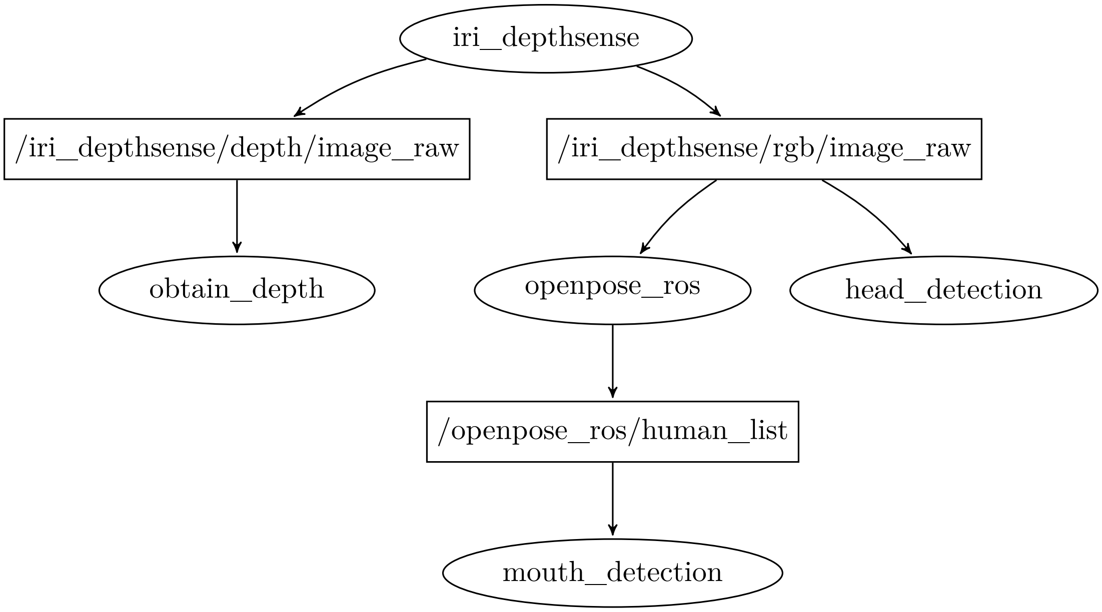


**Figure 6.7:** ROS nodes and topics for face detection

Therefore the nodes used and implemented are the following ones:

- **iri_depthsense_camera:** this is a ROS node developed at IRI. It is in charge of obtaining the image of the RGB-D camera and publishing it in the respective topics. Concretely, it publishes the RBG image in the topic /iri_depthsense/rgb/image_raw and the depth image in the topic /iri_depthsense/depth/image_raw.
- **obtain_depth:** this is a custom made node that is subscribed to the /iri_depthsense/depth/image_raw topic where it obtains the depth images. This node is used to obtain the depth of a determined point of an image. Therefore, in this particular case it is used to obtain the depth of user's face in order to know if the information from the face detection libraries is reliable or not. To do so, this node offers a service where it returns the depth of the given point of an image.
- **head_detection:** this is a custom made node that is subscribed to the /iri_depthsense/rgb/image_raw topic where it obtains the RGB images. This node calls the OpenFace library in order to obtain the head orientation in Euler angles. With this information it decides if the person can be looking at the gripper (either directly or sideways). It offers a service where, when requested, it returns whether the user is looking at the gripper or not.
- **openpose_ros:** this is a node available on-line. It is subscribed to the /iri_depthsense/rgb/image_raw topic where it obtains the RGB images. Every time this nodes receives an image it calls the OpenPose library where it obtains the facial landmarks and it publishes them together with their probability to be correct in the topic /openpose_ros/human_list.
- **mouth_detection:** this is a custom made node that is subscribed to the topic /openpose_ros/human_list. Every time a message is published in that topic it computes whether the mouth is open or not. It offers a service where, when requested, returns if the mouth is open or not or if it is not defined.

### 6.3.2 Force sensing and limiting

Another key factor in the feeding task is the safety of the movement. This is why it is of vital importance to have a force sensor in the end effector of the WAM. When the limiting force is exceeded it changes to gravity mode in order to not cause any harm to the user.

#### 6.3.2.1 Force limits

When we try to find the limiting force we encounter major problem: the RGB-D camera is fixed on the gripper after the force sensor. This causes the force sensor to not give the force of the end effector of the spoon as the camera has a not negligible weight and so it causes a moment of inertia.

To solve this problem different force limits have been established for different movements. For this particular project just three trajectories have a force limit as they are the ones with a potential higher risk. These three trajectories are: enter mouth, exit mouth and exit mouth 2. However, exit mouth and exit mouth 2 have the same force limits as they execute a similar trajectory.

For these limits it has only been considered the force produced in the Y axis as it is the axis where the force is going to be executed if the spoon impacts with the persons face and if the person holds back the spoon and does not let it exit the mouth. The force limits that can be observed in Figure 6.2 have been obtained after a trial and error process.

| | Minimum value [N] | Maximum value [N] |
|---|---|---|
| Enter mouth | -1.5 | 4 |
| Exit mouth and exit mouth 2 | -7.1 | 4.5 |

**Table 6.2:** Force limits for the enter mouth, exit mouth and exit mouth 2 trajectories

It is important to comment that when entering the mouth if the spoon impacts with the user's face the resulting force will be a negative one in the Y axis. On the other hand, when exiting the mouth if the person holds back the spoon the resulting force will be positive one in the Y axis.

When exiting the mouth the spoon is slightly turned upwards to generate a more natural trajectory. This causes and increase of the moment of inertia produced by the camera and thus the limiting force when executing this trajectory has to be higher.

#### 6.3.2.2 ROS nodes

In order to implement the a force limiting tool in thr ROS framework a IRI made node has been used and a node has been developed. They interact through a topic called /netft_data. This implementation can be observed in Figure 6.8.

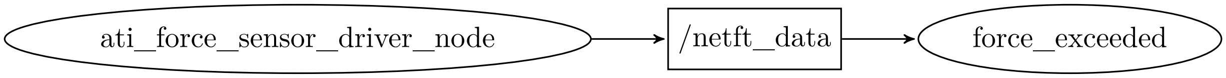


**Figure 6.8:** ROS nodes and topic for force limitation

Therefore, the nodes used and implemented are the following ones:

- **ati_force_sensor_driver_node:** this is a node developed at IRI that obtains the readings from the force sensor. It transforms the force readings into Newtons and the torque readings into Newton per meter and publishes them into the /netft_data topic.

- **force_exceeded:** this is a custom made node that reads the force sensor information from the /netft_data topic. It offers a topic where, when requested and given the movement the robot is performing, it returns if the force has been exceeded or not.

### 6.3.3 Robot movement

In order to feed the person we need to be able to move the robot to the desired position following safe trajectories. This objective can be achieved through different ways.

#### 6.3.3.1 Trajectories

There exist two basic ways to go to a certain point. The first one is using inverse kinematics. With this technique the joint rotations are computed from the end effector position and rotation in WAM space. This computation is made using simple mechanics as the distances between joints are known. The second technique is learning from demonstration. With this technique the developer teaches the robot how to move from a position to another one and the robot will reproduce those exact same movements.

As the WAM has 7 DOF there exist many positions with which the robot has the same end effector position and rotation. This presents a big problem for the inverse kinematics approach as the robot can be following one trajectory with certain rotations and suddenly change to a different solution making an abrupt movement that can scare or harm the user. Moreover, with this approach it is not possible to determine how the WAM is going to move and thus it can follow unnatural trajectories or hit some objects that should not be reached.

On the other hand, learning from demonstration trajectories can be more natural as the developer is the one to decide how are they going to be. However, if only the trajectory is to be learned in joints, it can not be adapted to different positions. Although solutions such as DMP or ProMP exist in the IRI laboratory, they have been discarded for the moment to ensure a robust and safe behavior using only joint trajectories.

The main groups of users of this project are disabled people and elderly people. This being said, these people are going to have reduced mobility which implies that the user's mouth position is not going to change constantly. Moreover, the position of the food can be fixed at a certain point. Taking into account these considerations it becomes clear that the most important thing is to make the task as natural as possible without scaring the user at any moment. As the user is assumed to be always at the same place and the food is going to be at a determined position, it is not essential to have trajectories that depend on the positions of the mouth and the food. This is why for these project the trajectories are going to be learned from demonstration.

Another factor to take into account is the compliance. A robot is non-compliant if the end effector is designed to have predetermined positions or trajectories. No matter what kind of external force is applied the robotic end effector will follow the exact same path every time. On the other hand, a compliant end effector can reach several positions and apply different forces on given objects. In this project the robot compliance is desired as it offers a passive security that assures a safer task development.

However, this compliance comes at a cost. There exist a trade-off between compliance and precision. When the compliance increases the robot becomes less precise and the other way around. This tasks needs to be as compliant as possible because a passive security is desired when treating directly with people. It is specially important during the enter mouth and exit mouth trajectories as during those trajectories the spoon is in constant contact with the user. Nevertheless, this task also needs to be very precise during certain trajectories especially during the food grasping and entering the mouth. Generally it needs to be very precise during those trajectories where the spoon is full because if this were not the case the food could be easily spilled. In the section 7.1 the performance of the robot with and without compliance will be tested.

#### 6.3.3.2 ROS nodes

In order implement all the previous aspects in the ROS framework some ROS nodes have been developed. These nodes send and receive information from each other using some topics, actions and services. In Figure 6.9 can be observed the ROS nodes, topics and services used.

Therefore, the nodes used and implemented are the following ones:

- **iri_wam/iri_wam_controller:** this is a node developed at IRI. This node is in charge of communicating with the internal controllers of the WAM robot in order to move it. It offers the action /follow_joint_trajectory which moves the robot to a certain position given a trajectory made of the joint states of several points. It also offers the service /hold_on which changes the robot state from gravity to normal state when requested. Moreover, it offers the service /joints_move which moves the robot to a certain position.

- **iri_wam_robot_state_publisher:** this is a node developed at IRI which is in charge of communicating with the internal controllers of the WAM robot in order to obtain information about its current state. Concretely, it offers the service /joints_move which, when requested, returns the state of all the joints of the WAM robot.

- **wait_gravity:** this is a custom made node for this project which is in charge of setting the WAM robot in gravity mode. Concretely, it offers a service which, when requested, sets the robot in gravity mode for one second.

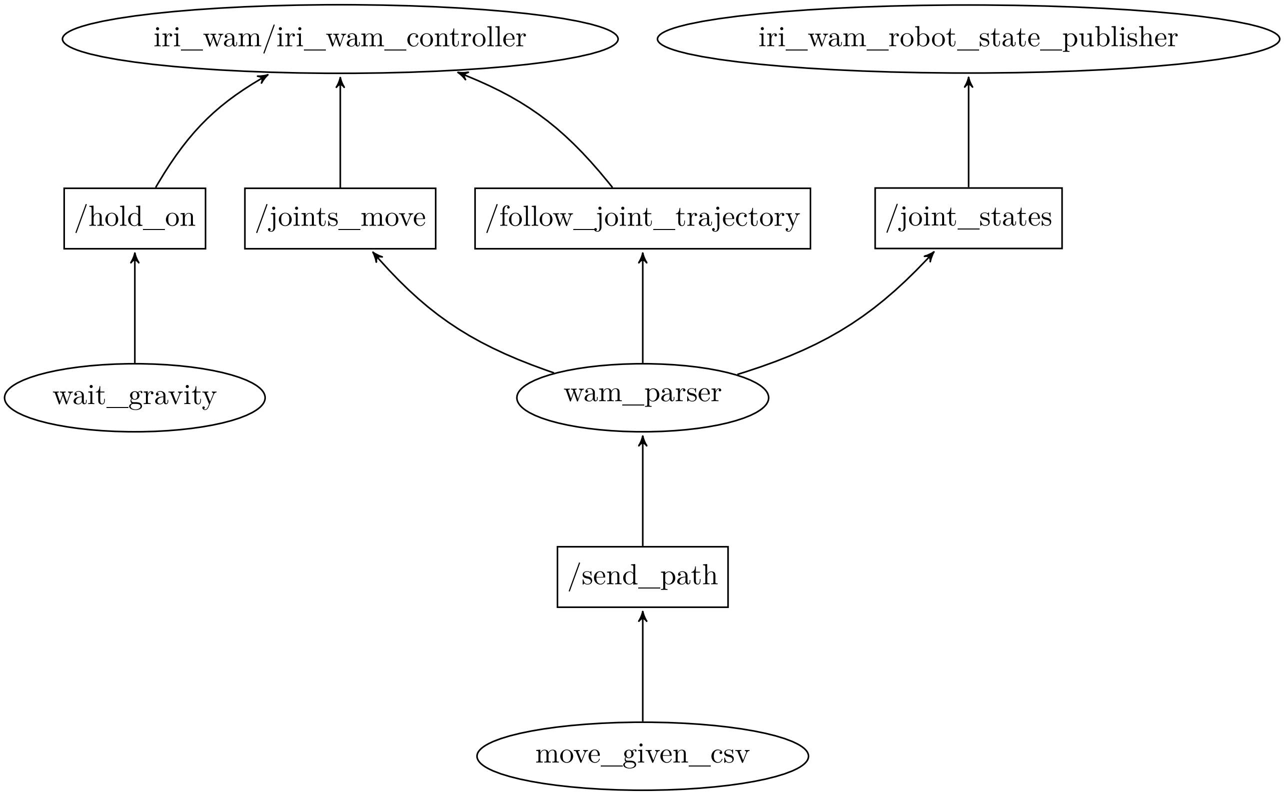


**Figure 6.9:** ROS nodes, topics, actions and services for moving the WAM robot

- **wam_parser:** this is a custom made node that offers an action which, when requested, move the robot to a certain position that is given through a csv file path or as a ending point. It has two different type of movements:
  - It moves the robot from the actual position (that is obtained reading from the topic /joint_states) to position 2. In order to do so it forms a 10 point trajectory and publishes this goal to the /follow_joint_trajectory action server.
  - It reads the csv file containing a trajectory formed with several points. Each point contains the joint states and the time where it has to be executed. First of all it calls calls the server /joints_move with the first position of the trajectory. When this position has been reached this node publishes the trajectory read from the csv to the /follow_joint_trajectory action server.
- **move_given_csv:** this is a custom made node that offers an action which, given a trajectory name, calls the /send_path action with the corresponding csv file path. With this proceeding the WAM is going to move following a desired

trajectory.

It is important to comment that the /follow_joint_trajectory action only moves the robot if it is in the first position of the trajectory. If this is not the case it aborts the action. This is why it is needed to send a service that moves the robot to the first position of the trajectory. However, there exists a problem with this proceeding as the service /joints_move can not be compliant whereas the action /follow_joint_trajectory can be compliant. This means that if the compliant method is used it will only be compliant during the action execution.

### 6.3.4 Communication and decision making

The nodes explained previously need to be connected together and called when needed. This is why the feeding_communication node is essential. This node is a custom made node which in charge of receiving all inputs previously mentioned, deciding which action to take and calling the adequate nodes. This node communicates to the other ones through actions and services as can be seen in Figure 6.10.

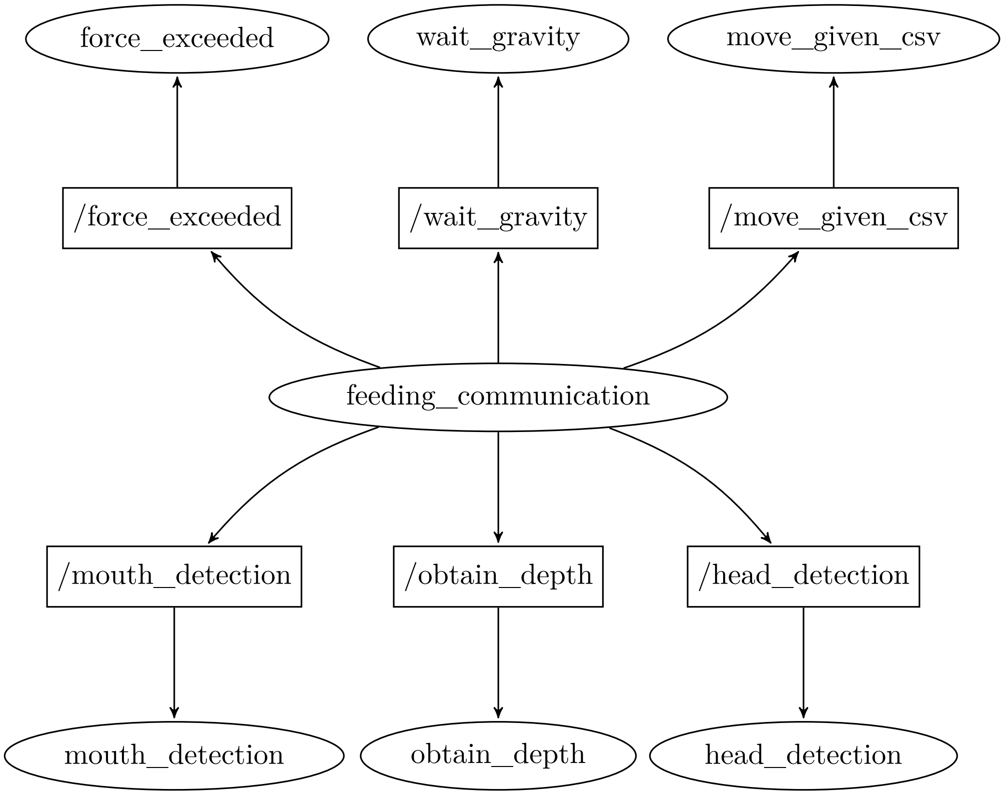


**Figure 6.10:** ROS nodes, actions and services for communicating all nodes

The feeding_communication node calls a custom made C++ library which is in charge of deciding which action to take at any moment. This library stores the current state and state of all variables and, with this information decides which action to take at any moment.

In order to obtain all the information that the library needs the feeding_communication node requests this information to the following services:

- **/force_exceeded:** this service tells if the force is being exceeded during certain trajectories.
- **/mouth_detection:** this service tells if the mouth is open, closed or if the detection probability is too low to assure a correct detection.
- **/obtain_depth:** this service tells the depth of a given 2D point in the same coordinates of the depth image.
- **/head_detection:** this service tells if the person can look at the spoon directly or sideways only judging by the head orientation.

The feeding_communication node gives all the information obtained with these services to the C++ library and asks which action to make. If the library decides that the robot has to remain in gravity mode the feeding_communication node will request the /wait_gravity service. On the other hand, if the C++ library decides that a movement has to be performed the feeding_communication node calls the /move_given_csv action. The C++ library also offers information about actions needing to be canceled.

Finally, it is important to comment that to activate the iri_wam_bringup node has to be called. This node is in charge of switching on the WAM and making available all the topics, services and actions that it has.

## 6.4 Regulation compliance

As it has been seen in chapter 5 personal care robots are regulated by ISO 13482:2014. The dangers explained in that chapter are the most influential in the feeding task. In the following sections it will be explained how the regulation is complied for each of these dangers.

### 6.4.1 Regulation compliance of stress dangers due to usage postures

In this prototype the spoon goes to a predefined mouth position. This implies that the user's mouth should be positioned at a certain point. However, this predefined position could be easily changed if the user can not seat in a comfortable position complying with this condition. Moreover, the user can seat in a chair that adapts to his needs and to the ergonomic principles described in ISO 14738.

The robot command is not wireless but it can be moved to any position the user needs it to be. This is due to the considerable length of the wire that connects the command to the robot. This guarantees that the user will be able to use the robot command without stress.

For those reasons, this prototype does not require the user to be in a stressing posture and thus it complies with this part of the regulation.

### 6.4.2 Regulation compliance of dangerous physical contact during the human-robot interaction

The most important aspect is to plan the lowest possible physical reaction between the robot and the user. In this prototype the limiting force has been set to the lowest value that allows the task development without being stooped by no other cause than contact. This is why the physical interaction is as low as possible.

Before starting the feeding task the user should be placed in the defined position and not move any part of the body except his face during the feeding task. This fact should be included in the instructions manual. If the user complies with these conditions the robot itself will never impact with the person because the spoon will be the only part that will impact with the user. As the spoon is the only part designed for tactile interaction we can assure that this aspect is complied.

In this prototype the only person expected to be inside the space reachable by the robot is the user. This fact should be included in the instructions manual. The user is constantly being detected and the robot will only approach the user if it is looking at the robot.

Moreover, all complementary measures are also fulfilled with this prototype:

- There exist software limits that mark the are of operation so the robot will only move in the defined volume. This is achieved by predefining the trajectories and not letting the robot move wherever it wants.

- The WAM driver has a speed limit and when it is exceeded the robot stops and slowly falls.

- Force and torque limits have been analyzed empirically using a force sensor as described in the regulation. This analysis has been developed in every trajectory where the robot can potentially impact with the user. These trajectories are the enter mouth and exit mouth trajectories. As will be explained in the experiments, when this limit is exceeded a fast enough answer exists so that the force is always under a harmful force. After an accidental contact is produced the robot will return to position 2. This is a secure position as in this place the robot can not impact with the user and will only move if the user opens the mouth.

With all this aspects we can conclude that the contact during the human-robot interaction is not harmful and therefore the prototype complies with the regulation.

### 6.4.3 Regulation compliance of dangerous autonomous actions

As explained before, the robot behavior is implemented using a FSM. If the variables that the FSM uses are right the robot will make the correct decision at any given situation. In case of an erroneous value of a variable due to an erroneous detection the robot can not cause any unacceptable harm as there exists a limiting force.

The damage caused by incorrect decision is decreased by choosing OpenPose over OpenFace as it has a higher accuracy and limiting the harm made by an incorrect decision through force limits.

The design restricts the scenarios where there exist potential risk of suffering damage to two. It is the lowest number of scenarios as in both cases there exists tactile interaction between the user and the robot and thus its risk can not be eliminated.

Moreover, this prototype also fulfills one complementary measures. The reliability of the algorithms has been increased to a level where unacceptable risks will not appear. The unacceptable risk of this prototype is a harmful contact between the robot and the user.

Taking into account all the previous aspects we can determine that during any autonomous action the robot will not harm the user and thus the prototype complies with this part of the regulation.

# 7. Experiments and results

In order to test the proposed robot behavior and its implementation, different tests have been performed. The first tests have been developed to analyze the safety of the system and thus they have been developed without real users. When the safety of the system has been confirmed more test have been carried out but this time with real users.

## 7.1 Limiting force

As said before the first tests have been developed to test the system security. In this tests both the active and passive security are being analyzed. The passive security is the one offered by the robot being in compliance mode. On the other hand, the active security is the one achieved by controlling and limiting the force obtained with the force sensor.

### 7.1.1 Setup

To perform this experiment a picture of a person opening the mouth has been fixed on a wood panel. This wood panel is strong enough to support the robot's force without moving or bending. This setup can be observed in Figure 7.1.

In order for the spoon elasticity to not affect the results the spoon has been turned 180 degrees so the spoon handle will be the one impacting the wood panel and not the other way round. The spoon has been fixed to the gripper with tape so it can not slip through it.

This experiment consists on the robot moving towards the picture with the same movement that it performs when entering the user's mouth. However, in this experiment the robot will impact with the wood panel. The resulting force will be recorded.

The tests developed in this experiment are going to analyze the robot response in the four different setups:

- **No-compliance mode and not manually limited force**
- **No-compliance mode and manually limited force**
- **Compliance mode and not manually limited force**
- **Compliance mode and manually limited force**

It is important to remark that in these experiments manually limited force stands for the force being limited through the code as explained in the section 6.3.2.1. In order to decide the best setup for the feeding task it will be taken into account the reaction time and the maximum force achieved as the most important aspect is to be able to ensure that the task is safe. Moreover, it will also be taken into account the robot's precision as it is important to maintain the spoon horizontally, especially when it is filled with food.

**Figure 7.1:** Setup of the limiting force experiment

### 7.1.2 Maximum force achieved

In order to test the maximum force in the four setups they have been performed while recording the force produced. Before doing these experiments, the force sensor has been tested in gravity mode with the robot in the same position as when it is performing the enter mouth trajectory. With this test it has been observed that the sensor noise is of about 0.15 N in the three axis.

It is important to remind that the direction of the force while the spoon is entering the mouth is Y. This is why the force produced in this axis has more relevance than the ones produced in the other axis.

In Figures 7.2, 7.3, 7.4 and 7.5, we can observe that there are three force peaks in Fy. The first peaks is of positive force and is caused by the robot moving from home to position 1. The second peak is also of positive force and is caused by the robot getting the food from the bowl. Finally, the third peak is of negative force and is caused by the robot impacting with the wood panel. The first two peaks are not relevant in this experiment as the robot is not impacting with the user when performing those trajectories. On the other hand, the third peak is going to be deeply studied as it gives information about the force that the robot is going to make if it impacts with the user's face. Not only it is important because it represents the impact of the robot with the user but because it is done on the face which is a sensitive part of the human body.

In Figure 7.2 it can be observed the resulting force of the impact between the robot and the wood panel when the robot is in no-compliance mode and the force is not manually limited. In this situation it can be observed that the force in the y axis decreases drastically in a low period of time. It is important to remind that the sensing range of the force sensor is from -20 to 20 N thus the force that exceed this limit are not reliable. However, seeing the tendency we can say that the force performed to the wood panel will continue increasing over time. In conclusion, with this mode the limiting force are exceeded by far and this increase is not stopped by the resistance of the wood panel. So it becomes clear that it is not safe to feed a person with this setup.

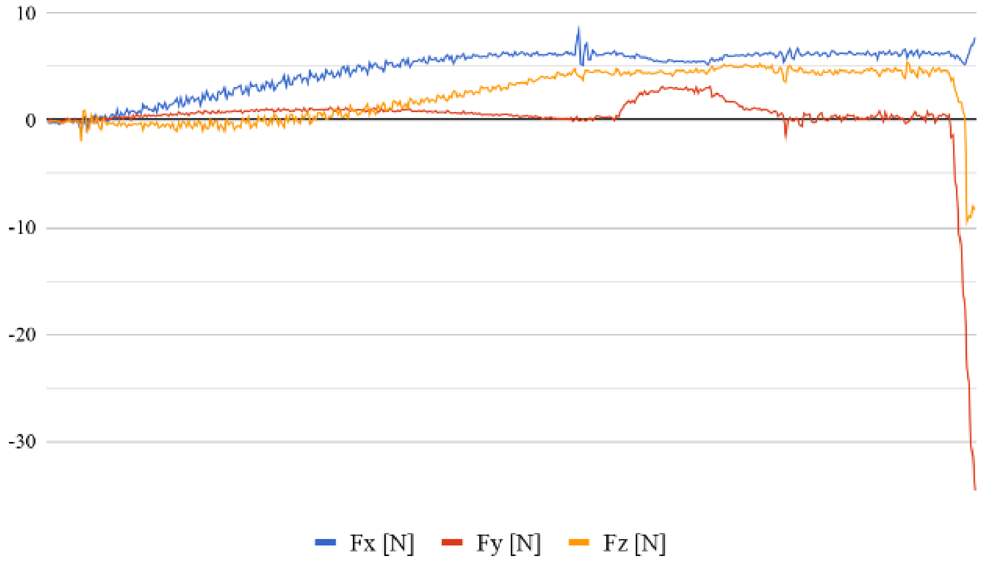


**Figure 7.2:** Measured force in no-compliance mode and not manually limited force setup

In Figure 7.3 it can be observed the resulting force of the impact between the robot and the wood panel when the robot is in a non-compliant mode and the force is manually limited. In this situation we can observe that force in the y axis when the spoon handle impacts the wood panel decreases up to -5.87N. The fact that this number is smaller than the minimum limit imposed is due to the force of the impact being bigger than the minimum limit imposed and the reaction time of the WAM. After reaching this point it increases and stays for a small period of time between -1N and -1.4N. This is due to the robot being in gravity mode so it does not represent a safety problem. After this the force increases again up to 0N and stays there. It can be observed that once the robot reacts to the impact the force generated are not harmful for a person so it is completely safe. Furthermore the minimum force achieved is of -5.87N which does not represent a big danger as a person can withstand this force on his face. In conclusion, this is a safe setup.

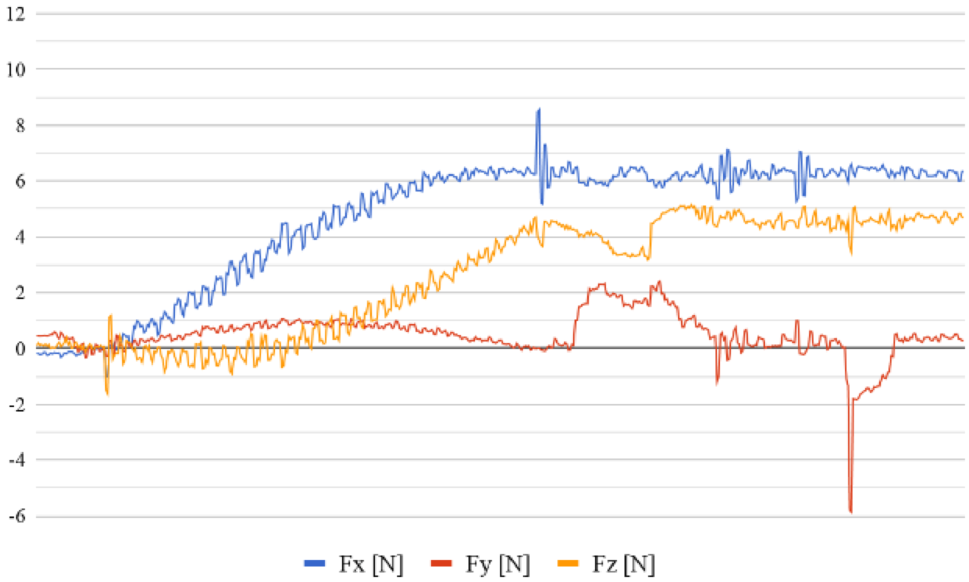


**Figure 7.3:** Measured force in no-compliance mode and manually limited force setup

In Figure 7.4 it can be observed the resulting force of the impact between the robot and the wood panel when the robot is in compliance mode and the force is not manually limited. It is important to remark that the force in the y axis of this graph has been normalized. This normalization has been performed in order to be able to easily compare the decrease in the Fy with the decreases of the Fy in graphs 7.3, 7.2 and 7.5. In this situation we can observe that the force in the y axis decreases up to -5.35N when the handle of the spoon impacts the wood panel. After the first impact the force increases to -3.68N. We can observe that after that the robot tries to reach the desired position again but this time the force produced is higher with a peak of -4.39N. After that the force increases again and remains between -3.9N and -4.19N. When the trajectory where the robot enters the mouth of the person finishes, the force increases to -2.88N as the robot is not trying to move to a determined position. With these results we can conclude that this setup is safe for the person because after a first impact the force remains at a maximum of -4.39N which is not a harmful force for a user's face.

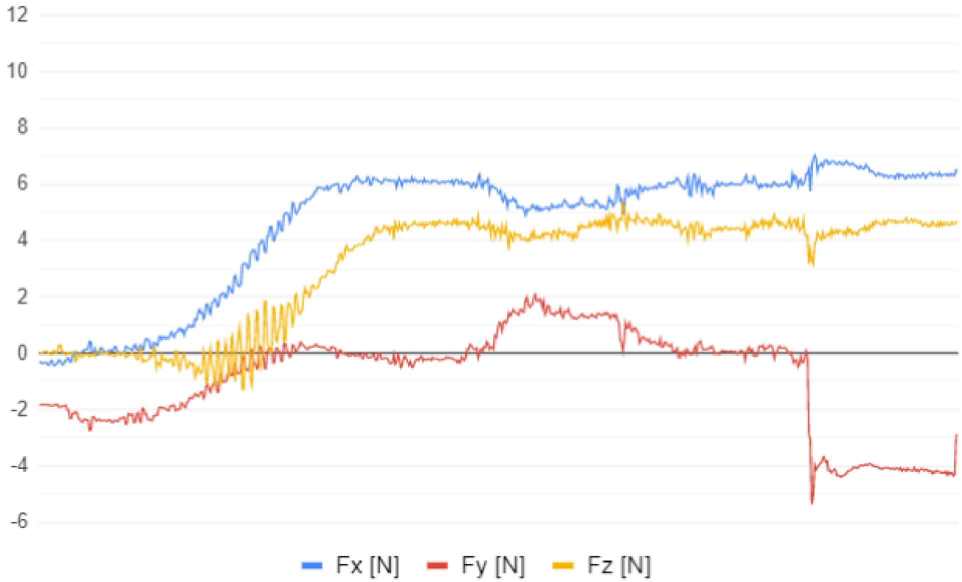


**Figure 7.4:** Measured force in compliance mode and not manually limited force setup

The last setup is the one where the robot is in compliant mode and the force is manually limited. In Figure 7.5 we can be observe the resulting force of the impact between the robot and the wood panel with this setup. In this situation we can observe that force in the y axis when the spoon handle impacts the wood panel decreases up to -5.57N. After that, it increases a bit and decreases again but now up to -6.4N. This is due to the fact that when the spoon handle impacts the wood panel the compliance stops the robot from performing more force on the panel. This is why the robot increases the force applied. However, after this increase the robot tries to reach the desired position again which causes the second decrease of force. After this the force increases drastically and stays between -1.7N and -2.6N. This increase is due to the fact that the limiting force has been exceeded and the robot has reacted to it so instead of trying to reach the final position it remains in gravity mode. We can observe that after that the force increases again up to 0N and stays there. This is due to the fact that the robot is not in gravity mode anymore and is pulling back. In conclusion, we can affirm that the robot with this setup is safe as after the first two impacts the robot remains at forces that are not harmful for the user's face. However, we can also observe that having an active and passive safety is not essential as the two safeties behave similarly in this particular case. Both safeties have an initial peak and then decrease to values that are not harmful for the user's face.

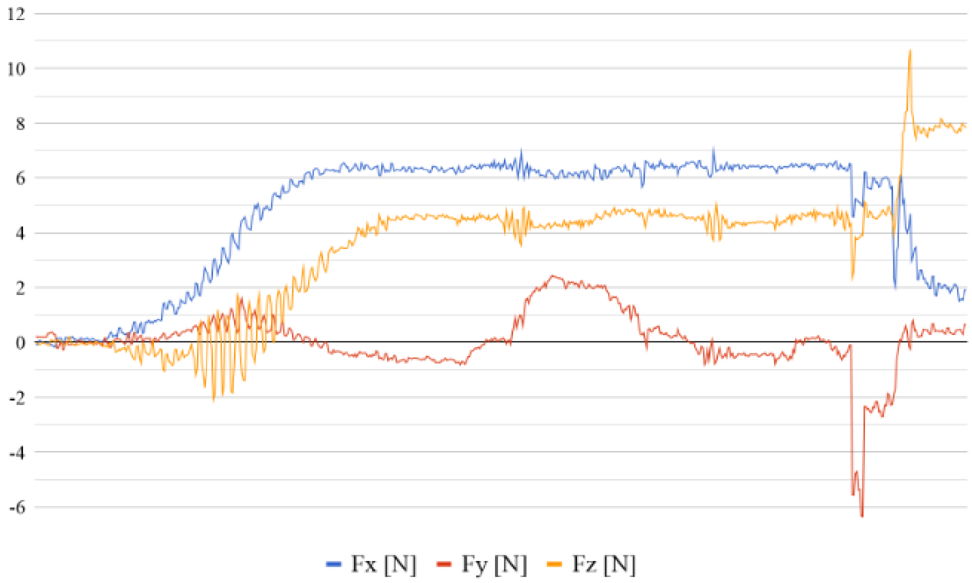


**Figure 7.5:** Measured force in compliance mode and manually limited force setup

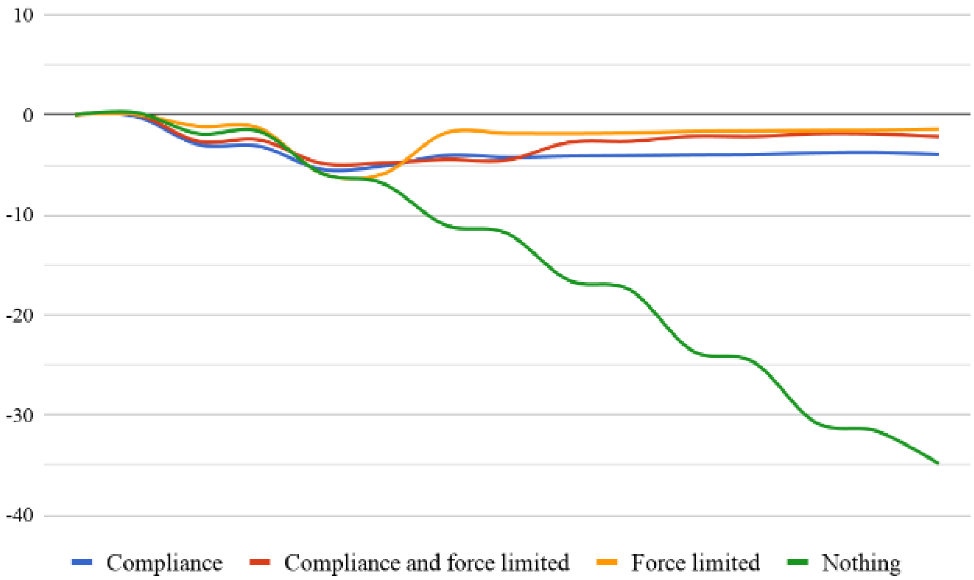


**Figure 7.6:** Comparison of the force in the y axis for the four setups during the impact

The moment of the impact is the most interesting in this experiment as it is where the minimum forces are reached. In Figure 7.6 a comparison between the four setups when the spoon handle impacts the wood panel can be observed. In this Figure all forces start at 0N to have a better comparison between them.

First of all, we can observe in Figure 7.6 that during the first period of time the setups that are compliant increase a little bit faster than the ones that are not compliant. After this first increase, the setups with compliance reach a peak of -4.8N whereas the setup without compliance and the force limited reaches a peak of -5.8N and the non-compliance and not force limited setup continues increasing. The no-compliance and force limited setup has an increase of force after its peak and remains in -1.6N as it enters the gravity mode stage. On the other hand, the compliance and force limited setup also reaches the gravity mode stage but it takes longer for the robot to reach it. To sum up, the compliance and force limited setup has a higher peak but takes longer to reach the gravity phase whereas the no-compliance and force limited setup has a lower peak but reaches faster the gravity phase. Finally, the compliance and not force limited setup remains in -4N as it is trying to reach the desired position.

### 7.1.3 Reaction time

From the same experiment, the reaction time of the robot when it is exceeding the limiting forces can be. For every setup the experiment has been carried out several times. The resulting reaction times can be observed in table 7.1.

| | Reaction time [s] |
|---|---|
| No-compliance mode and not manually limited force | $\infty$ |
| No-compliance mode and manually limited force | 0.225 |
| Compliance mode and not manually limited force | 0.3 |
| Compliance mode and manually limited force | 0.25 |

**Table 7.1:** Average reaction time for the four setups

The moment in which the force is considered to be exceeded is when the force sensor in the y axis is less than -1.5N. It has been considered that the robot starts to react when the force increases drastically. The sensing rate of the force sensor is 20Hz. Knowing the number of readings between the first reading where the force exceeds the limiting force and the first reading where the force has increased drastically we can compute the reaction time using the following equation:

$$t_r = number\_of\_readings/20$$

In table 7.1 it can be observed the reaction times with the different setups. The no-compliance mode and not manually limited force has an $\infty$ reaction time because the force never ceases to increase as the robot is constantly trying to reach the desired position. Between the other setups we can observe that the ones with compliance have a higher reaction time. This is due to the fact that there are two trials with high forces where the robot to reach the desired position. After this two trials the force decreases. When the force is just manually limited there is just one trial so the robot reaches faster higher values of force.

### 7.1.4 Precision

As explained in section 6.3.3.1 there exists a trade-off between compliance and precision. This is why the robot's precision is affected by the compliance of the robot. However, manually limiting the force does not affect the precision.

After doing some testing it has become clear that if the robot is compliant, the food is spilled more easily. This happens because the spoon is not completely horizontal and it moves more than needed.

With this result it becomes clear that the compliance in its current state is not precise enough. In order to obtain both compliance and precision a different controller should be studied. This controller would have to allow compliance only on selected components. So it would allow compliance in the y axis but not the rotation of the end-effector.

### 7.1.5 Discussion

With the previous results we can conclude that there exists a need to have a passive or active safety as if it is not the case the robot will reach harmful values of forces. We have then three possible setups: one with passive and active safety, one with active safety and one with passive safety.

Passive safety, in other words, the robot being in compliance mode offers a safer operation as the robot reaches lower forces. However, the difference of peak forces between compliance and limiting force is only of 1N so it is not a determining factor. On the other hand, passive safety has a longer reaction time compared to active safety as it has a rebound, but the difference is, again, not deciding. The least important factor is the precision as it does not affect the safety of the user. But in this factor there exist a great difference between compliance and no-compliance setups as the ones without compliance have good precision whereas the compliance setups have less precision which may cause the food to be spilled. Considering these factors, it becomes clear that the best setup is the one that has the force limitation and is non-compliant.

The chosen setup offers a safe task development with a low reaction time which is perfect for our task. It is important to note that the chosen setup only has active safety.

## 7.2 Experiment with users

Once the safety of the robot has been tested and it has been determined that it is safe for a person to be fed by the robot, it is possible to perform experiments with users.

The data of this experiment has been collected from 104 execution with 10 able-bodied participants. Three participants were female and seven participants were male.

It is important to comment that before doing the experiments the users are asked to fill the image rights form shown in Appendix A. Thereby, the images obtained during the tests with people can be presented.

### 7.2.1 Setup

In this experiment a bowl containing yogurt is placed in the predefined location of the food. The user is placed in front of the food and his mouth in its predefined location. The chair used can move upwards or downwards to adjust to the user's height. This setup cam be observed in Figure 7.7.

Before starting the test the functioning of the robot performing the feeding task is explained to the users. After the explanation the user is asked to try the execution a reduced number of times. During these trials the bowl containing the yogurt is moved away so the user can feel more confident as does not have to worry about food spilling. With these trials the user learns where is the predefined location of the mouth. Knowing this the user can adjust the height of the chair and can position his mouth in the predefined location.

**Figure 7.7:** Setup of the experiment with users

After this initial setup the yogurt is placed again in the predefined food location. At this moment the experiment is started to record. It is important to comment that in order to obtain accurate results the experiment is recorded with rosbag. Rosbag is a tool offered by ROS that enables to record the state of the selected topics during a desired amount of time. In this experiment the topics recorded with rosbag are the RGB image, the force sensor readings and the joint states topic. With this three topics all the relevant information can be accurately obtained after the experiments have finished.

During the first test with each user, the user was asked to behave as they wished. After those first test some specific tests were held in order to evaluate the reaction time of the robot and the maximum force produced. Some of this experiments are:

- When the robot finished the food grasping the user is not looking at the robot, and some seconds later, the user turns the head to look at the robot.
- When the robot is in position 2 the user turns the head and with the head turned opens the mouth. Then he/she starts to turn the head again but this time to the robot with the mouth still open.
- When robot is at position 2 the user opens the mouth and some seconds later he closes it so the robot impacts with the user's face.
- When the spoon is inside the mouth of the user the user bites the spoon and does not let it exit the mouth.

This experiment has allowed the testing of the following things: head detection reaction time and accuracy, mouth detection reaction time and accuracy and force limiting.

### 7.2.2 Head orientation

This section is dedicated to analyze the reaction time of the robot to the changes in head orientation and its accuracy. As said before during all the tests the image topic has been recorded using rosbag. A later extraction of reaction times and accuracy has been performed. An example of the head orientation detection can be observed in Figure 7.8.

The accuracy has been counted as the number of times the robot detects the head in the correct orientation. To do so it has been counted the number of times the spoon has moved without the head being oriented towards it, the number of times the head is oriented towards the food but the spoon has not moved and the number of times the robot has acted correctly.

The reaction time has been counted as the difference of time between the head being oriented towards the spoon and the starting of the robot movement.

Using these proceedings the results shown in Table 7.2 have been obtained.

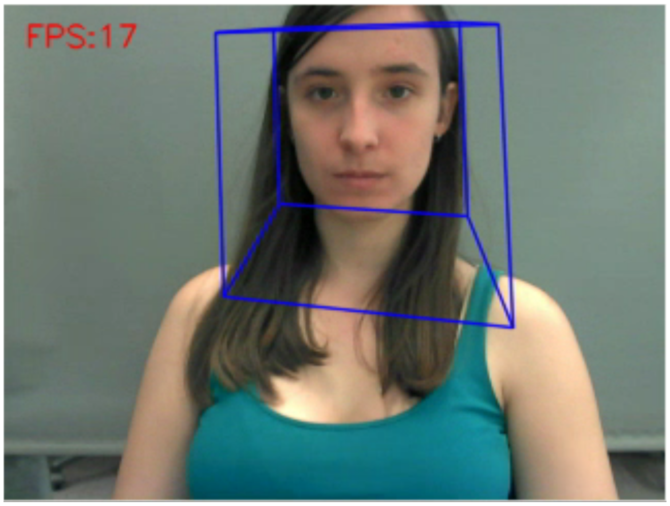


**Figure 7.8:** Head orientation detection using OpenFace

| | |
|---|---|
| Reaction time | 0,46 seconds |
| Accuracy | 100% |

**Table 7.2:** Head orientation reaction time and accuracy

We can observe that the accuracy is perfect as the robot always reacts as expected. Regarding the reaction time a lower value would be better but there is always a delay between a change and the detection of that change. However, a delay of less than half a second is enough to detect the problem before causing any harm to the user. The user will not get tired of waiting half a second every time he has to be fed so it does not present a major problem. The risks in this detection would come with a lower accuracy but as it is not the case we can conclude that it is completely safe and recommendable to use OpenFace for the head orientation detection.

### 7.2.3 Mouth detection

This section is dedicated to analyze the reaction time of the robot to the changes in the mouth state and its accuracy. As in the previous section the only required information to extract these values is the image topic and it has been recorded using rosbag. A later extraction of reaction times and accuracy has been performed. An example of the facial landmark detection made by OpenPose can be observed in Figure 7.9.

The robot actions that have been counted to calculate the accuracy are the following ones:

- If the user opens the mouth and the robot immediately starts moving towards it the accuracy increases

- If the user opens the mouth and the robot does not move towards it the accuracy decreases
- If the person has the mouth closed and the robot starts moving towards it the accuracy decreases

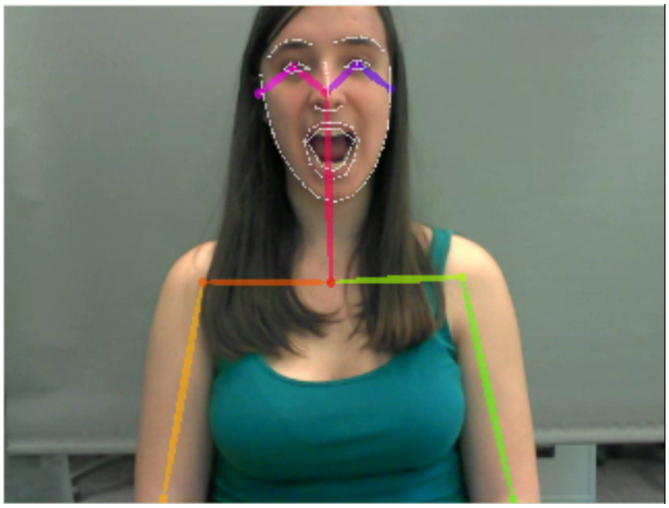


**Figure 7.9:** Face detection using OpenPose

The reaction time has been counted as the difference between the oppening of the mouth and the robot starting moving towards it. The results obtained following these procedures can be observed in Table 7.3.

| | |
|---|---|
| Reaction time | 0,44 seconds |
| Accuracy | 84,47% |

**Table 7.3:** Mouth state reaction time and accuracy

As explained in the previous section a lower reaction time would be better. However, this delay allows the detection of the problem before causing any harm to the user. Moreover, the user will not be tired of waiting half a second every time the spoon has to enter his mouth.

The biggest problem we encounter in the mouth state detection is the accuracy. This detection is crucial as if it is not done properly the spoon can impact with the user's face. Although the precision is not perfect it is important to remark that there was a 0% of false positives so the robot never started to move towards the user's mouth when it was actually closed. The 15,53% of errors were due to the robot detecting the mouth closed when it was actually open. It is important to remark that the 91% of this errors were produced in the same two users. One of this users has a big beard which may confuse the algorithm. With this results we can determine that it is safe to use this algorithm to detect the mouth state.

### 7.2.4 Force limiting

This tests have been performed with the setup chosen in the previous experiment which is no-compliance and force limited. Two aspects have been further studied in this experiment: the force of the impact between the user's face and the spoon and the resulting force of the user biting the spoon when it is exiting the mouth.

#### 7.2.4.1 Impact between the user and the spoon

In order to obtain the results for this test the only topic used is the force sensor topic as it has all the needed information. In the first experiment the spoon handle was the one impacting but in this experiment the impact is produced by the spoon. As the spoon can bend easily we want to test if the setup chosen is still valid.

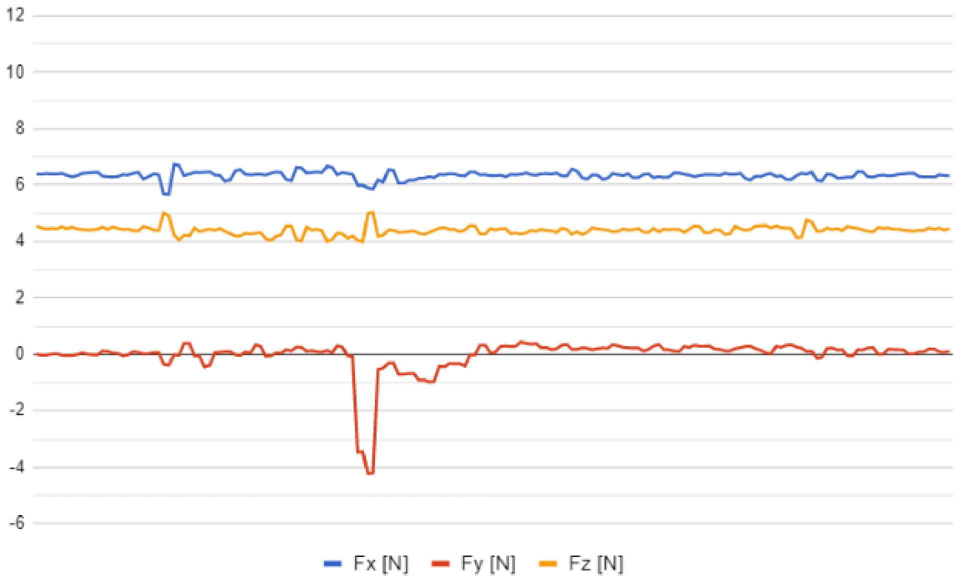


**Figure 7.10:** Impact between the spoon and the user's face

As we can observe in figure 7.10 the results obtained are extremely similar to the ones obtained in the previous experiment. There is a first impact produced at -3.45N.

We can observe that the force remains constant for a small period of time. This is due to the small bending that the spoon experiences. After the spoon has bent the force decreases again arriving at a peak of -4.22N. At this point the force limit acts which produces a drastic increase of force. The robot remains in gravity mode for one second and the force remains around 0.7N during this period. After this the robot retreats which causes the force to increase again and remain around 0N.

It is important to remark that the maximum impact force between the user's face and the spoon has been produced at -6.7N which is still not harmful for a person. All users reported that the impact was not harmful. The reaction time of the robot in this circumstance has not changed from the previous experiment. We can then conclude that it is safe for a person to be fed only with active safety.

#### 7.2.4.2 Spoon retained by the user

In the first experiment the only force analyzed was the impact force as it is the one with the biggest potential to harm the user. However, it is also interesting to analyze the forces during the trajectory exit mouth when the person retains the spoon.

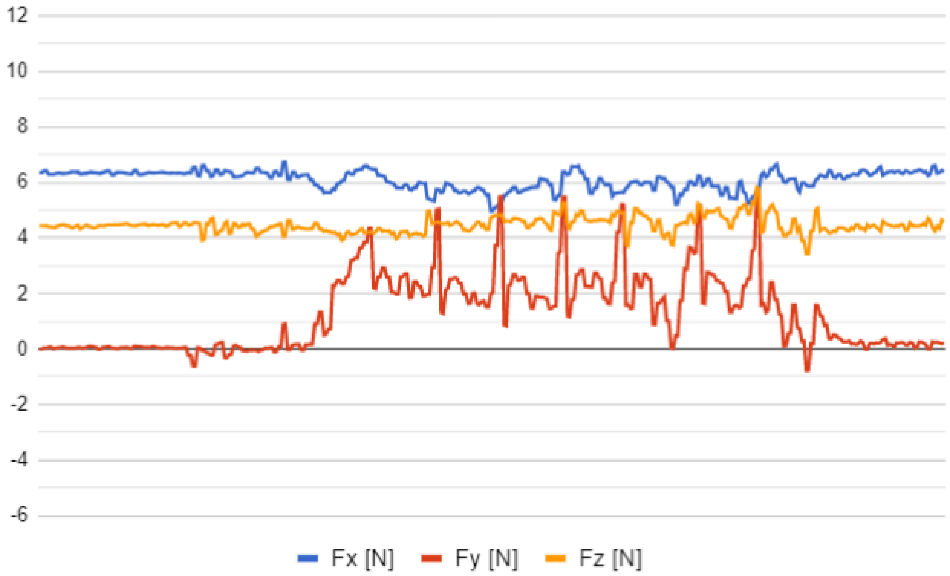


**Figure 7.11:** Forces resulting of user retaining the spoon

At the start of this trajectory the robot tries to exit the mouth but as the user is biting the spoon the force increases. When the force limit is reached the robot changes its state to gravity mode and so the force decreases. But after one second of being in this mode the robot tries to exit the mouth again which causes another force peak and the robot responds with the same behavior. The robot is going to repeat this procedure until the user frees the spoon.

In figure 7.11 we can observe the force graphic for this situation with the explained robot behavior. As can be observed there are several force peaks in the y axis, the higher one with a force of 5.48N. We can also observe that when the robot remains in gravity mode the force vary from 0.2N to 2.7N. From this graphic we can determine that in this situation the robot has tried to exit the mouth 7 times and during the eighth trial it has succeeded.

We can then conclude that the robot has behaved as expected. Moreover, the peak forces are not high so the person does not have to endure a lot of force before the robot stops applying it. To sum up, it is safe for the user to retain the spoon as the robot is going to let him do it and will only exit the mouth when the user has truly freed the spoon.

### 7.2.5 Discussion

With this results we can conclude that the robot performing the feeding task with this implementation results in a safe and accurate performance. In Figure 7.12 can be observed the robot performing the feeding task in the three stages where the robot is closer to the person. In Figure 7.12(a) the robot is approaching the user's mouth whereas in Figure 7.12(b) the robot is exiting the mouth. Finally, in Figure 7.12 the robot is impacting with the user's face.

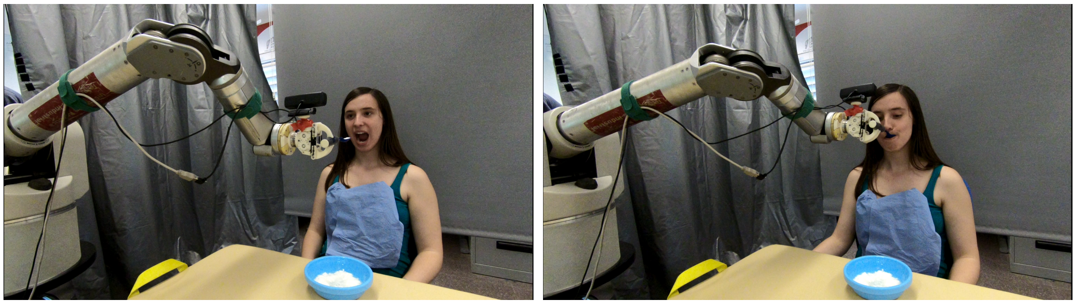
The head accuracy is perfect so the spoon will never approach the user if it is not looking at the robot. We have also determined that the mouth state accuracy is not perfect but the there is a 0% of the robot approaching the mouth when it is actually closed. This is why those detections are completely safe.

However, the percentage of the robot not approaching the mouth when it is actually open is not negligible. This causes a waiting period longer than expected. Moreover, the reaction time of both head detection and mouth state detection are noticeable for a person. This is why the waiting time should be improved.

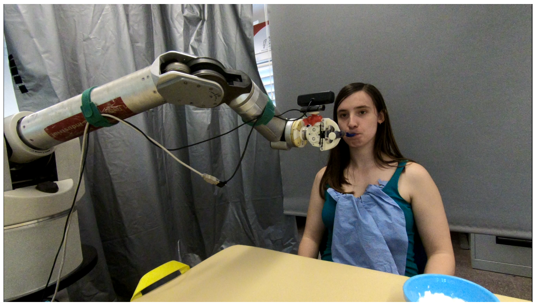
If the user closes the mouth when the depth limit has been exceeded the robot will impact the user's face. As a reminder: the depth limit indicates the limiting distance from which the head orientation detection and the mouth state detection have a high rate of wrong detections. This could yield a problem if the impact were not safe. But as analyzed in the two experiments the impact is completely safe and not harmful for the user.

Moreover, it is also safe for the user to retain or move the spoon once it is inside his mouth. The robot is going to try to exit the mouth but will never reaching a harmful force.

The most important thing to conclude with this experiment is that all users felt comfortable with the prototype. For example one of them said that it felt as if his mother was feeding him.



**(a)** Entering mouth **(b)** Exiting mouth



**(c)** Impact with user's face

**Figure 7.12:** Robot in different stages of the feeding task

# 8. Budget

In this chapter the cost of the hardware, the software and human resources are presented. It is important to remark that it is just an estimation so the real cost of the project may differ from the calculated.

## 8.1 Hardware cost

The following hardware resources have been used in this project: two computers, one force sensor one Creative Senz3D and one WAM robot. The prizes of these components are listed in Table 8.1. In order to calculate the prize per hour of each resource the following equation has been used:

$$Price\_per\_hour = Unit\_price/(Amortitzation\_period * 250 * 8)$$

In this formula the 250 belongs to the number of working days in a year and the 8 of this formula belongs to the number of working hours each day has.

| **Resource** | **€/unit** | **Amortization period [years]** | **Price per hour[€/h]** |
|---|---:|---:|---:|
| WAM robot | 97.500 | 10 | 4,875 |
| Creative Senz3D | 149 | 4 | 0,018625 |
| Force sensor | 6.000 | 3 | 1 |
| PC | 1.500 | 4 | 0,1875 |

**Table 8.1:** Price of each hardware resource

In Table 8.2 we can observe the number of hours each hardware resource has been used and the total cost of all the hardware resources.

| **Resource** | **Units** | **Hours of usage** | **Cost [€]** |
|---|---:|---:|---:|
| WAM robot | 1 | 70 | 341,25 |
| Creative Senz3D | 1 | 90 | 1,68 |
| Force sensor | 1 | 50 | 50 |
| PC | 2 | 400 + 70 | 88,13 |
| Total | | | 481,06 |

**Table 8.2:** Hardware resources budget

## 8.2 Software cost

The software resources used are free to use for a non-commercial application so the cost of the software for this project is 0€. However, if this project wants to be used for commercial purposes this prize would change. Concretely, all resources are also free for commercial purposes unless the OpenFace and the OpenPose libraries. A license of OpenFace costs 12.887,69€ a year whereas a yearly license of OpenPose costs 21.484€.

## 8.3 Human resources cost

I have been working in this project for approximately 20 hours a week during 5 months. This is a total of 400 hours. The supervisor has dedicated 5 hours a week during the same 5 months which is a total of 100 hours. Finally, the users of the experiments have dedicated a total of 10 hours. In Table 8.3 the salary of the engineer, the project manager and users are presented as well as final costs.

| **Role** | **€/hour** | **Hours** | **Salary [€]** |
|---|---|---|---|
| Project Manager | 40 | 100 | 4.000 |
| Industrial engineer | 20 | 400 | 8.000 |
| Tester | 15 | 10 | 150 |
| Total | | 520 | 12.150 |

**Table 8.3:** Human resources budget

## 8.4 Total cost

Taking into account all previously specified costs we can observe the total budget in Table 8.4. An extra 5% has been added to the budget to cover any factor that may have not been considered.

| **Concept** | **Estimated Cost (€)** |
|---|---|
| Hardware | 481,06 |
| Software | 0 |
| Human resources | 12.150 |
| **Subtotal** | 12.631,06 € |
| Contingency (5%) | 631,55 € |
| **Total** | **13.262,61 €** |

**Table 8.4:** Total budget

# 9. Conclusions

A prototype of a robot capable of feeding a person autonomously in a safe and natural manner has been proposed and implemented in this thesis. The user of this prototype is thought to be a disabled or elderly person and thus the robot has to be able to develop the task on its own. The main focus of this project has been feeding the user in a safe and natural manner. Therefore, food grasping has not been deeply studied. This prototype has been successfully tested in the WAM robot.

Two aspects of the user's face have been detected. The first one is head orientation which has been achieved with the library OpenPose. This information allows the robot to know if the person is looking at the robot either directly or sideways. The second aspect is mouth state detection which is an essential part of this application as the robot has to know whether the mouth is open or not. This detection has been achieved with the library OpenFace. As has been seen, both detections are fast enough for this application. Moreover, OpenPose offers a perfect accuracy and OpenFace offers a high rate of accuracy and an extremely low rate of false positives.

The security of the prototype has been deeply studied as it is an essential part of this thesis. The chosen security setup is a setup with only active security as it offers a safe task development without loosing precision. To achieve this active security a force sensor has been used. The force has been limited during the potentially harmful trajectories in order to guarantee a safe task development.

A study on the current regulation on physically assitive robots has been conducted. The design of the prototype has been made in order to comply with this regulation. Moreover, the final design complies with this regulation.

A first set of tests has been carried out to test the safety of the prototype and to decide the best security setup. Once the security has been checked, tests with users have been carried out resulting in a satisfying robot behavior in all of them. These tests have shown that the prototype is capable of feeding people autonomously with a natural approach and without causing any harm.

A robot like the one presented can have a significant impact on the life of the disabled and elderly, especially on those that can not self-feed. This impact is achieved through a gain of independence and thus, an empowerment of this group of people. This is why a robot like the one presented can have an unlimited social impact.

# 10. Future work

Many extensions and improvements could be included in the system. The most notorious one is face tracking. Currently, the location of the mouth is obtained but the robot moves to a predefined position. Some solutions to this issue would be using DMP to move the robot. However, at the moment it is not reliable enough for this application because a wrong movement could have harmful consequences.

A further improvement would be to also detect the food position, state and to move consequently. In order to detect the food position and state the Creative Senz3D camera mounted on the gripper could be used. Although the gripper could cause some occlusions, there is enough range to be able to locate the food. In order to implement this improvement a computer vision algorithm able to detect the location and the state of the food should be implemented. Moreover, to achieve this improvement it is also needed to move the robot to a desired position without having the trajectory previously defined. Finally, it should be made a further study of the strategies the spoon should follow to grasp the food depending on its state.

Another possible improvement is to use the depth obtained with the camera to detect whether the spoon is inside the mouth or not. Because of the sensing ranges of the camera the current setup can not be used for this purpose. In order to achieve it a different camera able to detect the depth of closer objects should be used. As an alternative to this solution the camera could be fixed in a different part of the robot so it will always remain further enough to obtain the correct depth. The problem with this solution is that the gripper con easily hide part of the mouth and thus make the system unusable.

In this prototype services are used to move the robot to the first position of the trajectory. This presents a big problem as services are blocking so even if the limiting force is exceeded the service will not stop. In order to not use these services custom trajectories should be created autonomously during the task execution. This has been implemented in the exit mouth 2 trajectory and could be applied to the rest of trajectories.

Currently, it is possible to only have active security because the user position is predefined and it is been indicated that only one person can be inside the space reachable by the robot when the task is being executed. However, in order to obtain a more flexible prototype in a less controlled environment passive security is needed. The problem that passive security presets is precision. Nevertheless, precision is especially needed in the rotation of the last joint because a small change in it can cause food spilling. In order to achieve this the robot driver should be changed to allow to have different compliances in different components.

In conclusion, if all these improvement were applied the safety of the resulting prototype would improve.

# Bibliography

[1] *ATI: F/T Sensor: Mini40.* `https://bit.ly/2IXauj2`. [Online: last accessed May 2018].

[2] *Eurostat: Population structure and ageing.* `https://bit.ly/2x3TL88`. [Online: last accessed April 2018].

[3] B. House, J. Malkin, and J. Bilmes. “The VoiceBot: A Voice Controlled Robot Arm”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI ’09. Boston, MA, USA: ACM, 2009, pp. 183–192.

[4] *Robots and robotic devices – Safety requirements for personal care robots.* Standard. International Organization for Standardization, Feb. 2014.

[5] *JAECO MultiLink Arm with Elevation Assist.* `https://bit.ly/2s4X9tP`. [Online: last accessed May 2018].

[6] A. Jardón, C. A. Monje, and C. Balaguer. “Functional Evaluation of Asibot: A New Approach on Portable Robotic System for Disabled People”. In: *Applied Bionics and Biomechanics* 9.1 (2012), pp. 85–97.

[7] *Meal buddy System.* `https://bit.ly/2s0SQAg`. [Online: last accessed May 2018].

[8] *Mealtime Partner Dining System.* `https://bit.ly/2LiDSOC`. [Online: last accessed May 2018].

[9] *Mobile Arm Support MAS Table Mount.* `https://bit.ly/2ICBksr`. [Online: last accessed May 2018].

[10] *Neater Eater.* `https://bit.ly/2x0Sw9E`. [Online: last accessed May 2018].

[11] D. Park et al. “Multimodal execution monitoring for robot-assisted feeding”. In: *Intelligent Robots and Systems (IROS).* 2017.

[12] D. Park et al. “Towards assistive feeding with a general-purpose mobile manipulator”. In: *IEEE International Conference on Robotics and Automation - workshop on Human-Robot Interfaces for Enhanced Physical Interactions.* 2016.

[13] C. J. Perera, T. D. Lalitharatne, and K. Kiguchi. “EEG-controlled meal assistance robot with camera-based automatic mouth position tracking and mouth open detection”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 1760–1765.

[14] T. Simon et al. “Hand Keypoint Detection in Single Images using Multiview Bootstrapping”. In: *CVPR*. 2017.

[15] I. Sumio. “Meal support robot "My spoon"”. In: *The Robotics Society of Japan Journal* 21.4 (2003), pp. 378–381.

[16] K. Tanaka, S. Mu, and S. Nakashima. “Meal-Assistance Robot Using Ultrasonic Motor with Eye Interface”. In: *Int. J. Automation Technol.* 8.2 (2014), pp. 186–192.

[17] *The WAM Arm.* `https://bit.ly/2IFFZhl`. [Online: last accessed April 2018].

[18] J. Jiménez Villarreal and S. Ljungblad. “Experience Centred Design for a Robotic Eating Aid”. In: *Proceedings of the 6th International Conference on Human-robot Interaction*. HRI ’11. Lausanne, Switzerland: ACM, 2011, pp. 155–156.

[19] *World Health Organization: Disability and health.* `https://bit.ly/2wZ22ds`. [Online: last accessed April 2018].

[20] A. Zadeh, T. Baltrusaitis, and L. P. Morency. “Convolutional experts constrained local model for facial landmark detection”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017.

# Appendix A

**Image release form**

I hereby grant Institut de Robòtica i Informàtica Industrial (IRI) permission to use my likeness in photographs, video recordings or electronic images in any and all of its publications, including website entries, without payment or any other consideration. I understand and agree that these materials will become the property of the organization and will not be returned. I hereby irrevocably authorize the organization to edit, alter, copy, exhibit, publish or distribute these images for purposes of publicizing the organization's programs or for any other lawful purpose. In addition, I waive the right to inspect or approve the finished product, including written or electronic copy, wherein my likeness appears. Additionally, I waive any right to royalties or other compensation arising or related to the use of my image. I hereby hold harmless and release and forever discharge the organization from all claims, demands, and causes of action which I, my heirs, representatives, executors, administrators, or any other persons acting on my behalf or on behalf of my estate have or may have by reason of this authorization.

I am 18 years of age and am competent to contract in my own name. I have read this release before signing below and I fully understand the contents, meaning and impact of this release.

(Signature)　　　　(Date)

(Printed Name)