Development of an EMG Controlled Robotic Arm to Restore Upper Limb Function

- Manipulating the Surroundings -

P3 Project Report

Group 363

Aalborg University
Robotics
Department of Electronic Systems

AALBORG UNIVERSITY
STUDENT REPORT

Robotics
Aalborg University
http://www.aau.dk

Title:
Development of an EMG Controlled Robotic Arm to Restore Upper Limb Function

Theme:
Manipulating the Surroundings

Project Period:
3rd Semester Fall 2015

Project Group:
15gr363

Participant(s):
Biranavan Pulendralingam
David Cerny
Emil Blixt Hansen
Rasmus Andersen
Steffen Madsen

Supervisor(s):
Anders la Cour-Harbo
Romulus Lontis

Copies: 3

Page Numbers: 75

Date of Completion:
December 14, 2015

Abstract:

The goal for this project was to develop a robotic solution for replacing functionality of upper limbs of people suffering from tetraplegia. This endeavour was motivated by identifying the issue through an analysis on topic of tetraplegia and related matters. For this task, a CrustCrawler robotic arm, controlled by accelerometers and EMG signals emitted by facial muscles, was utilized. The product was delivered through establishing a mathematical model for the arm, programming user's interface, setting up the processing and designing a control system algorithm. The interface alone and the final solution were also tested in three different cases of daily routines. Results and procedures are documented in relevant sections.

Preface

The following report is the product of a 3^{rd} semester project group, written by students of Robotics at Aalborg University. The topic for this project is a robotic solution for people suffering from tetraplegia (paralysis of the whole body except the part above neck). It describes the utilization of EMG muscle signals to move a robotic arm.
This report comes with a CD containing electronic copy of the report, video footage, full code, and MATLAB files. The report uses Harvard method for referencing the sources. In certain places, the annotations for sine and cosine functions (*sin* and *cos*) are reduced to *s* and *c*, in order to save space in long equations.
We would like to thank our supervisors, Anders la Cour-Harbo and Romulus Lontis, for insightful guidance throughout the project. Our thanks belongs also to Karl Damkjær Hansen, who constructed the CrustCrawler arm for us, and offered us a practical guidance about the arm. Also he created the Dynamixel library, containing useful functions e.g. for setting the torque or position, which simplified our work. Lastly we would like to give our thanks to John Hansen for providing the EMG control box.

Aalborg University, December 14, 2015

Biranavan Pulendralingam
<bpulen14@student.aau.dk>

David Cerny
<dcerny14@student.aau.dk>

Emil Blixt Hansen
<ebha14@student.aau.dk>

Rasmus Andersen
<rean14@student.aau.dk>

Steffen Madsen
<smad14@student.aau.dk>

Contents

Chapter 1

Introduction

A paralysis of the whole human body is an serious issue. The body is rendered immobile due to damage in a nervous system or directly in a spinal cord. Studies say that complete or partial paralysis is not a small problem, but affects a considerable number of people all around the world. Having a partial or complete paralysis means that the patient has a reduced muscle activation he or she can carry out, or no activation at all. Paralysed people around the world, depending on their condition, are divided in group of paraplegics (paralysis from waist down), or tetraplegics (paralysis of the whole body from neck down). Although there are many different causes for paralysis, as a stroke or a sickness (e.g. multiple sclerosis, or amyotrophic lateral sclerosis), the further analysis is mostly concerned by paralysis caused by spinal cord injury, as it is more relateable for majority of people. [Melton 2015]

A survey, conducted in the United States of America, shows that 1 out of 50 people is suffering from partial or complete paralysis of muscles, meaning that there is approx. 6 million (1.3 million caused by spinal cord injury) in USA affected by these conditions. The same survey also states that this number is 33% higher than estimations from previous years. That way it suggests that the amount of people suffering from paralysis in USA is steadily growing. Also the venture into depth with spinal cord injuries revealed that around 47% of affected people end up with tetraplegia. [Christopher and Reeve 2015] [Alabama 2002]

Another survey, that is from Australia, gathers statistics about people suffering from spinal cord injuries. This survey shows that there is approximately 12 000 Australians that sustained spinal cord damage, and 350 to 400 new cases are recorded each year. In 15% of the total amount of cases a person ends up with complete tetraplegia, and in 38% of the cases a person has incomplete tetraple-

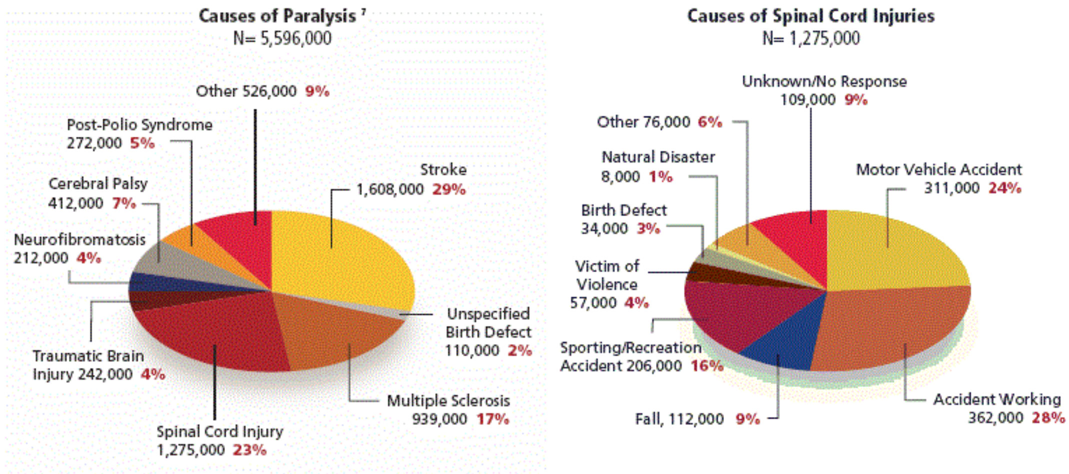


Figure 1.1: These pie-charts show what are the causes for paralysis and spinal cord injury in USA. [Christopher and Reeve 2015]

gia, which means a loss of movement in all four limbs, although certain nerves are still operational and allows a person to make certain moves. [Rebuildinglives 2014]

Statistics from World Health Organization provide global key facts that there are between 250 000 and 500 000 new cases of spinal cord injury every year. [WHO 2013]

From the previous presented surveys, [Christopher and Reeve 2015], [Alabama 2002] and [Melton 2015], and charts in figure 1.1 it can be concluded that the number of paralysed people around the world is high, and the annual arise of new cases is considerable as well. This means there is a big group of patients with great limitation in their lives. It can also be presumed that a majority of them are dependent on daily care of another person, which might cause economical problems.
Most of the probable causes of complete or partial paralysis are strokes and spinal cord injuries. Namely the spinal cord injuries are usually caused by car accidents or accidents at workplaces. That means that such injury are not caused by rare circumstances. Thus there is a relative high probability of being injured, if one is not vigilant enough.

Complete tetraplegia means the patient has no activation of the muscles from neck down. This project sets its focus on complete tetraplegia but where the patient still has an absolute control of the area of neck and above. Otherwise the product of this project would not work. The goal of this project is to find a robotic solution, controlled by electromyography (EMG) signals, which are read from the face muscles of a patient. The concern of this issue is divided into different areas

that were chosen to be further investigated and analysed.
The goal of this project is to find a robotic solution, which would make daily routines of the affected people easier and thus make them more self-dependent. The aim is to utilize a provided robotic arm (CrustCrawler), combined with EMG electrodes to carry out ordinal daily tasks.

Chapter 2

Tetraplegia and Related Assistive Technology

This chapter analyses issues of tetraplegics' life struggles, assistance technologies and muscle functionality. This research is made to acquire necessary background that would be needed to understand in order to designing a solution suited for patients with tetraplegia.

2.1 Every Day Life

The following section contains a description of how patients with spinal injury, such as tetraplegia, meet and experience everyday life. This is done in order to describe how physical challenges can affect the mental health of the patients. This is done in order to analyse the problem, and to set a motivation for a development of this project.

Tetraplegia is a condition that often is caused by an accident. This means that people with tetraplegia in most cases are unprepared for the condition before they are diagnosed, and that their ability to move their bodies from the neck down is gone from one day to another. Because of this unprepared, physical transformation, tetraplegia patients are forced to sudden radical changes of lifestyles. The changes are related not only to physical but also the psychological health of the patients.

When a person is hit by a severe accident, he/she will first of all go through a tough emotional time. [Johnson 2010]
Even though it can vary from person to person what they feel, it is common to refer to the Kübler-Ross model, which also is known by the name "The Five Stages of Grief" when there are strong negative feelings involved. [Kuebler-Ross 1969]

The five stages of grief are denial, anger, bargaining, depression and acceptance. Kübler-Ross model is based on interviews with relatives of patients who just died. When the model first was published in 1969 Kübler-Ross only meant it as a model of a person who just lost a close-relative, e.g. wife. Nonetheless it has been accepted as a model for all various of losses, from death, to job or illness.

The physical life that the tetraplegia patients knew before, is replaced by a life in a wheelchair. This new condition is forcing the patient to live with decreased physical freedom, therefore many of the tasks that the patients were carrying out before, without thinking, are suddenly now, if not impossible then hard to do even with helping tools and caretakers or an environment that is adapted to the decreased freedom. Even a task as simple as eating is a challenging task that therefore requires more time, thinking and concentration than it would have had required before the patient got a spinal injury.

These radical changes of lifestyle are mentioned in an article brought by Sygeplejesken by Sanne Angle. She describes how patients with spinal injury meet every day life. The article takes point in a study made by Angle consisting of 6 women and 6 men in an age between 17-70, who all have in common that they are new spinal injury patients. The purpose of the study is to follow the patients from injury start and 2 years forward, and see how their view on life changes as time goes by. From the study, Angle found 6 similar stages, that the patients went through, in a process of moving forward and finding meaning in the new forced lifestyle. [Angle 2009]

1. **Survive physically and rediscover the will to live**
   The patients first of all have to survive physically, and rediscover the will to live. Here the will to live is fragile, and it is hard for the patient to see the future, and for some of the patients, death is preferable. Here family and professional help plays an important role, because those are the ones who stimulate a picture of a future. This stage is the base for the process of rediscovery the meaning of life for the patients.

2. **See a possible future**
   The patient is starting to regain hope. Here a rehabilitation program is set and the patient is learning how to deal with the new lifestyle. The patient now has something to work for, and a rehabilitation program that he/she can construct their everyday life from. This stage is dependent on how the professional help supported the patient through the rehabilitation program, and towards the life he/she wants to live.

3. **Progressive training**
   There is progress in the rehabilitation and some patients can via an adapted

environment reach some of the goals they had. In this stage there are hopes and dreams for a better future.

4. Experience the possibilities narrow down
The signs of improvement are not as radical as in the beginning stage. Many of the results the patients are achieving are due to technology. Therefore the hope the patient had in stage 3 is challenged. This stage is hard and it can take a long time, compared to the other stages, to get by.

5. Exploiting the limited options
The patient defies the feeling of emptiness, and is getting aware of the values that surrounds him/her. The patient finds some value and meaning, from the life before the injury. This could be due to the surroundings, that have adapted to the patient's handicap, e.g. a well adapted housing. The dreams and hopes for the future are getting adapted to what is possible, which makes the uncertainty of the future disappear. Due to this the frustrations are getting smaller and the feeling of well-being is getting bigger.

6. Signify the limited possibilities value
The patient is feeling well, and the big changes in lifestyle are becoming routine. The rehabilitation is getting in the background compared to the everyday life. The patient still meets challenges such as environment that is not adapted for his/her handicap or new health problems. However the spinal injured is, in this stage, living the life in acceptance of his/her limitation of everyday life.

As stage 5 describes, the patient finds values and meaning in an environment that is adapted for him/her and thereby gets a better well-being. A big part of this adapted environment consisting of assistive technology (AT), which purpose is to enable the patient to carry out activities as he/she did before the injury. These AT's help the patient to increase independence in the everyday life.
In the following case Susan Peterson is a patient suffering from tetraplegia.

2.1.1 Susan Peterson Case

Susan Peterson is a woman who survived a car accident in 2008. Susan was a massage therapist and was living a healthy life by going to the gym 6 times a week. However, the accident in 2008 gave her a cervical spinal injury, which made her paralysed from the neck down. She says that she misses the little things like scratching her head and she dreams of going down the beach and eating food, before she wakes up and she is back in the wheelchair.
Peterson states the numerous different things she needs to adapt to. This includes caretakers, medication and a wheelchair operated by her chin. She is also learning

how to operate a computer by her voice, so she can control the TV and other kind of electronic devices. [Ross 2015]

As a summary for this section, it can be concluded that the everyday life of patients with tetraplegia, completely changes the form of everyday life he/she had before the injury. A task as simple as eating is an impossible task to carry out without some assisting help. There can also be concluded that the initial days after getting a spinal cord injury are difficult, and it is first when the patients starts to adapt their dreams and hopes to what is possible for them, that they can start to live everyday lives. Additionally, it can be concluded that the patients need an interaction between an adapted environment and AT, in order to carry out activities as they did before the injury and to increase independence.

2.2 Existing Methods and Alternatives

A research has been made to summarise what kind of help is available nowadays for a patient suffering from paralysis. The less self-dependent the patient is, the greater is the need for nursing, and the more expensive it is to take care of the patient. The research also explains alternative solutions and projects that could reduce the nursing expenses, and make people suffering from tetraplegia, being more self-dependent.

One of the most common ATs that can be offered to paralysed people is nursing assistance at home or having them hospitalized at care homes. However this is an expensive solution. The less self-dependent the patient is, the more expensive it is to take care of him/her.
Genworth is a company who specializes in life-insurances. They have done a comparison and estimated the average cost for care-service in USA. Having a caretaker come home and provide help costs annually from \$44 000,- to \$46 000,- (293 056,- DKK to 306 376,- DKK) depending on extent of the required assistance. If the patient's state has great need of attention, it is likely that he or she will end up in a care home, for which the costs annually are between \$80 000,- to \$91 000,- (532 828,- DKK to 606 092,- DKK), depending on whether he/she wishes to have a private room or shared room. [Genworth 2015]

In Europe, the costs are not as high as in the USA (as illustrated in figure 2.1). E.g. in UK the annual cost for home care is about £11 000,- (111 836,- DKK) if the patient needs care for 14 hours a day. Home care for 24 hours a day costs about £30 000,- (305 009,- DKK) a year. Being accommodated in a care home with nursing services costs about £37 500,- (381 261,- DKK) a year. [Buisson 2014]
Such prices can create an economical pressure on a paralysed individual. It

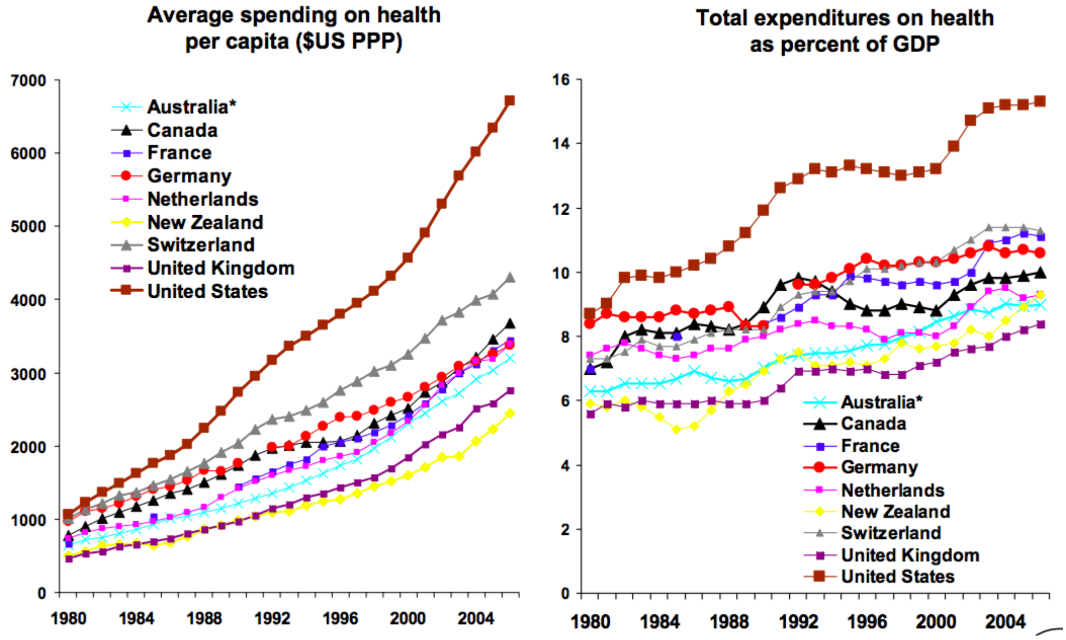


Figure 2.1: Charts compare total spending on health care between USA, European, and other countries. [Sanjay Basu 2012]

is presumed that the care, however helpful it might be, is going to be degrading for patients in certain intimate situations, which might influence them mentally. Therefore, there are efforts to develop helpful gadgets for paralysed people. Solutions that increase patient's self-dependency can reduce care costs and somewhat renew the feeling of self-worth of the particular person.
This project focuses mainly at issues of tetraplegics. Things like plain wheelchair are not mentioned in following text since it does not return any self-dependency to people with tetraplegia as they are not capable of moving it without external help.

One proposal for AT that can help patients be more independent in their daily lives, is the Tongue Drive System (TDS). This system allows the user to access their environment through tongue movement. The tongue has many properties, not only for eating and ingestion, but it is also not prone to fatigue as fast as other muscles. It does not require a lot of thinking to move, as it is very natural and can move reflexively. TDS is a wireless and wearable device that records the position of the tongue and can transmit it to the users surroundings. Through moving the tongue to predefined locations within the oral area, the assistive device can issue commands and send them to a computer or smartphone. TDS is a non-invasive device that can allow the user to control a mouse or keypad on a computer screen, or control a wheelchair with the use of only the tongue. Since this project is still in testing and development stage, there is no information about actual costs of the product, although sources suggest that the solution is going to be low-cost. [Jeonghee Kim 2013] [Satapathy 2014]

A study from Brown University describes their researchers' project of a robotic arm being controlled by human brain activity. With the advanced development of brain-computer interface, their test subjects, suffering from tetraplegia, were able to move the robotic arm by thinking about moving their own arm, and successfully carried out simple tasks (like picking up a cup of coffee, taking a sip, and putting it back). This might be a revolutionary improvement in paralysed peoples lives, however the project is still in an early stage and there is no actual purchasable product on the market. [Orenstein 2012]

Last thing to mention is Stephen Hawking's computer, which is a good example of a gadget that returns a part of self-dependency to a person. Stephen Hawking is a famous and very successful theoretical physicist of this era. He suffers from amyotrophic lateral sclerosis, which rendered his whole body immobile, making him even unable to speak. He still can make minor muscle contractions, especially in facial area. Stephen received a computer from the company Intel, that was made specially for him and is attached on his wheelchair. He controls the machine by a single cheek muscle.
There is a software keyboard on the screen of his computer, and a cursor is slowly scanning it by row or column. Hawking is then able to select the desired button by twitching his cheek, which is detected by an infra-red switch mounted on his glasses. He can also talk this way, when he sends a build-up sentence to an included speech synthesizer. Intel still researches ways to make the interface faster and are thinking of implementing more functions. Although the Assistive Content-Aware Toolkit platform (ACAT) was developed originally for Hawking, Intel has released the platform as open-source, making it available for any developer to create different solutions for computer interfaces, which can have interesting implementations in the future. [Hawking 2015] [Prasad 2015]

Those three examples shows that there is an initiative in the world to help paralysed people. However all of the mentioned alternative solutions are still in stage of development. It concludes that there is still room for new ideas. And that is what this project is going to try to offer.

2.3 Muscles and EMG

Tetraplegia is a condition that results in a paralysis of patient's torso and limbs. Because of this, the patients are not able to perform all the daily routines they could before this condition. As chapter 1 states, the goal of this project is to control a robotic manipulator, using EMG signals sent from a contraction of facial muscles, and thereby enable a tetraplegia patient to perform selected daily routines, with the help of the manipulator, EMG signals and accelerometer. This chapter investigates the muscles in the face along with what EMG is.

2.3.1 EMG Signals

EMG is an abbreviation for electromyography which is a technique to read the electric activity produced in the skeletal muscles. [Robertson 2013]
To further understand what EMG is, the following example will demonstrate what happens to a muscle when it performs a contraction.
Firstly, the brain sends impulse signals to the muscle that tells it to perform a concentric contraction (the muscles shortens). The way a muscle contracts is described with the Cross-bridge theory, by Huxley. A muscle contains two proteins, myosin and actin. In figure 2.2 it can be seen that the myosin attaches to the actin and thereby contracts the muscle. [Staff 2014]
When this happens, an electrical activity is generated in the muscle, which is what the electrodes can read as EMG signals.

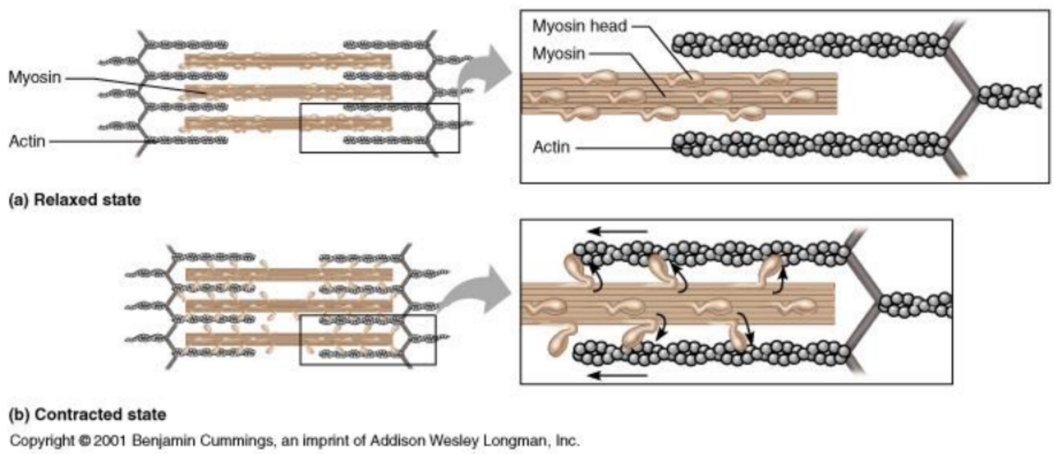


Figure 2.2: The Cross-bridge theory where the myosin connects with the actin to contract the muscle. [Zee 2015]

The EMG is a signal that has a very fast frequency (depending on how fast the program is coded to read) which is clear from a graph (as in figure2.3). When a muscle is relaxed, the EMG signals with a small amplitude can still be read. However, when the muscle starts to contract the amplitude grows. The growth of the amplitude varies depending on how much force is needed. In figure 2.3 the relationship between the force applied and the amplitude in the EMG signal can be seen.

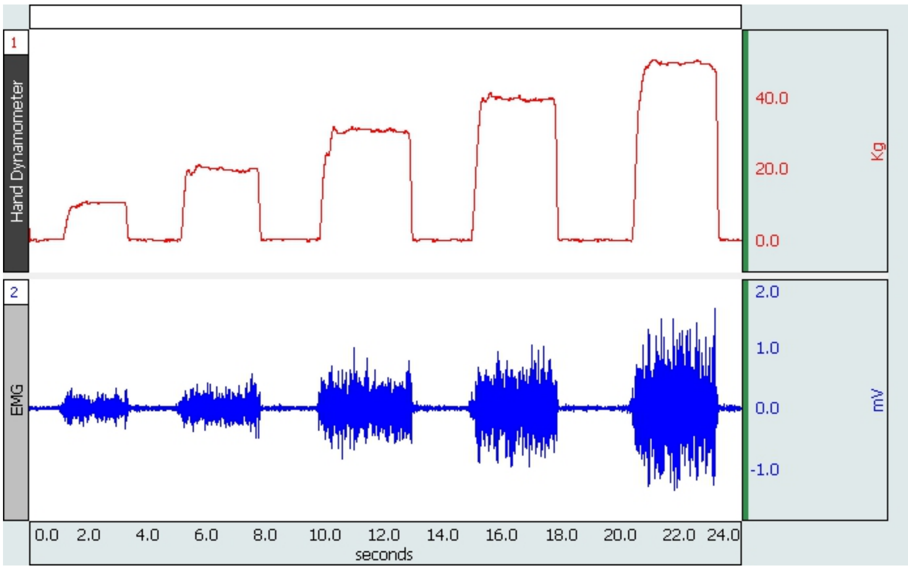


Figure 2.3: The relationships between the force applied and the EMG signal read. [BIOPAC 2015]

2.3.2 The Face Muscles

The human body contains different types of muscles. These can be categorised into two groups:

- Striated muscles.
- Non-striated muscles.

The striated muscles are the most common muscles in the body. These muscles can be controlled voluntarily and are often attached to the skeleton. Some striated muscles go under the category cardiac muscles and cannot be controlled voluntarily. The heart is an example of a cardiac muscle.

The non-striated muscles are the muscles, which cannot be controlled voluntarily. These are found in e.g. veins.

The muscles selected for this project must be a striated non-cardiac muscle so the patient can control them with ease. [Zee 2015]

The face contains over 20 muscles, therefore a selection is made to locate muscles which are easy to control and to read. [Taylor 2015]

Figure 2.4 shows the muscles suitable for this project. The selected muscles are the Masseter muscle and the Frontalis muscle. The Masseter muscle was selected because it is one of the main muscles, which is used while chewing, and for that reason is easy to control. The Masseter is also a surface muscle and therefore the EMG signal can be read with the electrodes. However, for this task there is a need of two channels of EMG signals and since the two Masseters operate parallel with each other, as they are located on both sides of the head and are connected to the jaw, another muscle is needed. [Zee 2015]

Therefore another easy-to-control muscle, the Frontalis, was selected. This muscle is normally only used to control facial expression and is therefore a muscle a patient will has good control over.

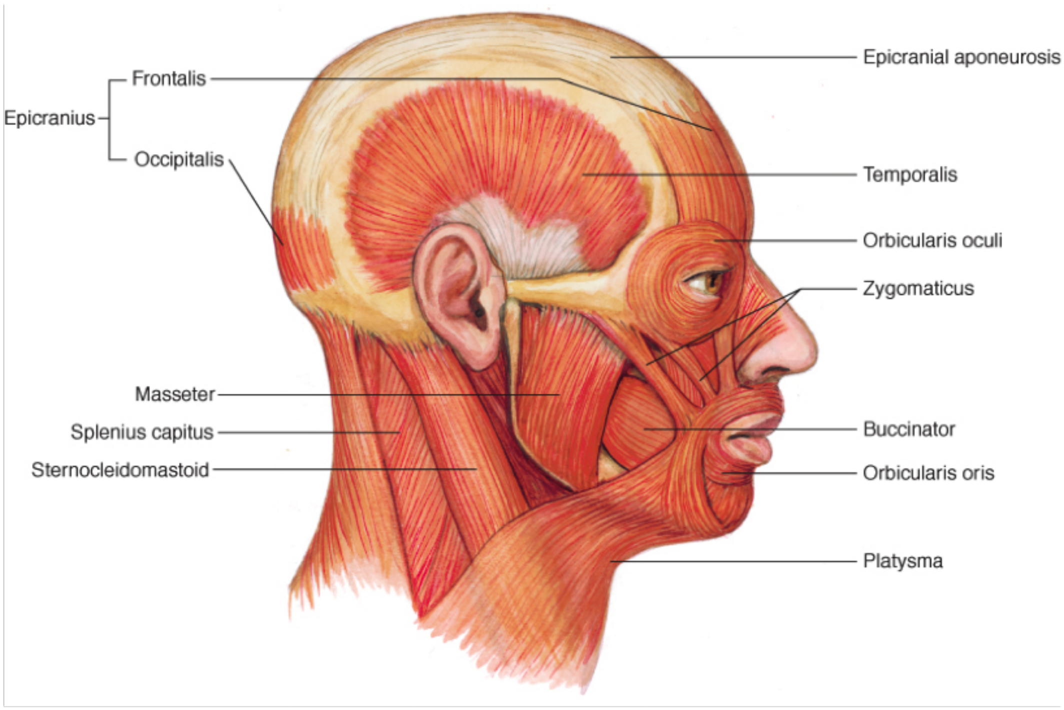


Figure 2.4: The muscles of the neck and jaw of a person. [Envisage 2014]

In order to implement the EMG signals in CrustCrawler's controls, certain counter-measures have to be applied, since the raw signal oscillates, due to its steep amplitudes (figure 2.3). To develop a robust solution, the concept of grey zones needs to be applied. This idea, along with signal processing, is described in section 5.1.

To collect the EMG signals, an EMG-reader has been provided and is described also in section 5.1.

Chapter 3

Problem Formulation

Diseases and conditions, such as tetraplegia, are known to impair the patient's ability to move and control their limbs. That reduces or completely disables the patient's ability to interact with their surroundings and making them very reliant on helpers to assist with everyday tasks. The tasks that the patient was able to perform before is with this kind of disease or condition, if not impossible, then very complicated to do without help. An assist can be offered in form of a human help or assistive technology.

This project focuses on using a CrustCrawler Pro Series arm with help of two channels of EMG signals and accelerometer. EMG can be recorded using surfaces from any superficial muscle which is translated to movement by the CrustCrawler. The processing is happening inside of a control box provided by university. The control box also contains the accelerometer, which is planned to be utilized as well.

Specifically the electrodes will be placed on the facial muscles Masseter and Frontalis. The end user of the product is considered to be paralysed from the neck down, thus the solution is focused on patients with intact facial muscles. Since the patients neck is not paralysed, it is possible to use the accelerometers by placing the control box on the right side of the head. When the patient does simple movements for example moving the head forward or backward, it is a possibility to transmit this specific response from the accelerometers into a command for the manipulator.

Grey zones are applied in order to make the EMG signals more reliable. With the grey zone it is not necessary to keep a constant tension on the muscle thus making the control of the CrustCrawler more robust. This method also makes the system more safe, because the signal needs to peak through the grey zone and into on-zone before activating the robot. So the patient can still do small movements like chewing.

Since the final solution will only have two EMG channels and accelerometer to control 5 servos, a mode selection system will be used to help the patient control the robot in an easy and reliable fashion. The mode selection can be combined with the accelerometers.

To control the CrustCrawler an ArbotiX-M micro-controller will be used. For safety, the connection between the EMG channels and the ArbotiX-M micro-controller will be wireless, so the current the CrustCrawler will draw, is not directly connected to the patient. This is done in order to limit the risk of electrocution.
The goal of the project is to complete 3 cases:

1. Pick and place.
2. Pressing a light-switch.
3. Drinking assistance.

This leads to the research question:

How can a CrustCrawler, controlled by ArbotiX-M micro-controller, be implemented as an assistive device that helps tetraplegia patients perform 3 daily routines, using accelerometer and EMG signals measured from contractions of patient's face muscles?

Chapter 4

Requirements Specifications

The solution to the problem, described in chapter 3, is obtained by a 3 degrees of freedom (DoF) manipulator that is able to help patients with tetraplegia. The manipulator in this project has to be able to carry out certain cases that represent daily life complications tetraplegia patients may experience. However, during the testing phase the manipulator will be controlled by a test subject, not suffering from tetraplegia. By 5 electrodes and an accelerometer attached to the subject's face, the test subject is able to control the robot via head movement and contractions of 2 facial muscles. When the test subject contracts a muscle, the electrodes pick up an electrical signal in form of EMG signals. These EMG signals are the input to the system. The purpose of this system, is to interpret the EMG signals and translate them to a safe desired motion, performed by the manipulator. Thereby it is possible for the patient to control the manipulator with muscle contractions. Figure 4.1 illustrates this purpose of the overall system where the EMG signals are the input and a controlled motion is the output.

This chapter explains the requirement specification of the solution and the cases the manipulator has to perform. Those are found in section 4.2 and section 4.3. This is done in order to create a basic understanding of the system and give a clear overview of what the solution is required to do.

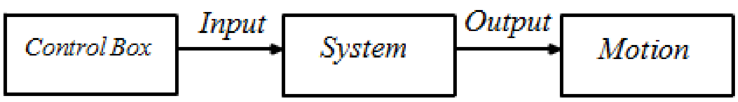


Figure 4.1: State of the system.

4.1 General Description

The system has an input in form of EMG signals and data received from an accelerometer, and an output in form of a desired, and controlled motion, performed by the manipulator.

This is illustrated in figure 4.1, which describes the state of the system. Further details on how the subsystems work are described in chapter 5. The system starts with the EMG signals received from the muscle contractions, which are the input into the system. Inside the system box different mathematical equations are applied (described in subsection 5.3.4) to make the CrustCrawler perform the desired movement. Then the system sends out the desired output and the CrustCrawler moves. This is further explained in chapter 5.

4.2 Selected Cases

In order to prove that the delivered system can work in real-life application, the manipulator needs to be tested to carry out certain real everyday-life situations that a person can face. 3 different cases have been selected, in order to simulate basic daily struggles of a person suffering from tetraplegia. Satisfying those will prove that the system can be implemented usefully in real-life scenario.

4.2.1 Pick and Place

Picking up things and putting them down somewhere else, is a basic functionality of an upper limb. If the interface of the manipulator reaches such level, that it will be possible to control all the manipulators servos using just 2 EMG channels and an accelerometer, this case will be considered a success. Namely the task to pick up a little box in a marked location and transport it to another marked location should receive higher attention. The reason for this is, that this test is an examination of fundamental functionality. The test will be undergone by 3 different people to see, if conditions differ for other individuals.

4.2.2 Pressing a Light-switch

In the modern age, pressing buttons might be as fundamental as grabbing and placing. Namely pressing a light-switch is a specific case, which could allow the solution to be made easier and more intuitive than trying to control each servo separately. E.g. the end-effector can be set in a given position and can be controlled by modifying the Cartesian coordinates of the end-effector, rather than trying to get into the right position by setting values to each joint.

4.2.3 Drinking Assistance

Last case is about solving a specific pick and place scenario, where it is to be tested, if it is possible to pick up a bottle of water and bring it to patient's face, using the manipulator exclusively. The requirement is to pick up the bottle and bring it to a desired position, from which it is possible to conveniently drink from the bottle. In this scenario it can be tested if pre-programmed movement functions can be utilized, since the position from which a person is able to start drinking, can be presumed to be always the same.

If the system is to be considered viable, those cases must be successfully carried out. Each case is picked also for testing out different aspects of the interface.

4.3 User Requirements

Here the requirements of the user are set. Since this project focusses on restoring upper limb dysfunction, the user of the final solution will have to be able to control at least 2 face muscles (Masseter and Frontalis). Since the final solution is based on visual feedback, the user also has to be able to see.

4.4 Design Requirements

This section sets the requirements for the hardware and the software used in the final solution.

The solution has to be controlled using only 2 channels of EMG signals, and an accelerometer with 3 axes. These EMG signals will read from the patient's face. To avoid an accident, where the electronics on the patient could fry, the final solution needs to have wireless communication with the manipulator for safety. This means that even if a situation with high current in unwanted places of the system happens, the current will never reach the user. The wireless communication should be with XBee modules to ensure a fast connection.

Since the final solution will only have limited input signals to control 3 servos and the end-effector, a mode selection system is necessary. This mode selection system should allow the user to control the CrustCrawler depending on the case. For instance it could be convenient to be able to move each actuator separately or move the end-effector in Cartesian space, but in the case of drinking from a bottle, it could also be convenient to have predefined functions which can be accessed easily to minimize the user input and speed up the operating process. Due to the mode selection the solution has to be able to be controlled via different modes, e.g. cartesian space and joint space. This means that a mathematical model of the kinematics for the CrustCrawler must be established. This kinematic must be implemented on a micro-controller.

In order to have a controlled motion of the manipulator, a linear control algorithm should be implemented on the micro-controller. A linear mathematical control model will be established and implemented on the micro-controller.

4.5 Performance Requirements

The performance requirements ensure that the solution will be able to handle the loads it encounters in the cases.

The solutions should be able to lift a typical bottle of water. Assuming a bottle filled with water will weight approximately 0.5 kg, all the actuators of the solution should be able to lift 0.5 kg in a fully stretched position at the outer bounds of the manipulator's reach. This means that the actuator with the heaviest load will have to produce a torque of:

$$\tau = 0.5kg \cdot g \cdot (0.22m + 0.27m) \approx 2.41Nm \tag{4.1}$$

where g is referring to the gravitational acceleration constant $(9.82m/s^2)$, and $0.22m + 0.27$ m refer to a length of the manipulator, when straightened out and horizontal. This calculation considers the two links to have a total weight of 0 kg.

4.6 Final List of Requirements

1. The final solution should be operated with EMG signal, obtained from two face muscles, and an accelerometer.
2. The user must be able to move his/her neck, in order to control the accelerometers effectively.
3. The signal between the CrustCrawler and the user is wireless.
4. Wireless communication is with XBee modules.
5. The final solution should be able to lift a bottle of water, equal to 0.5 kg.
6. The final solution should be able to be controlled using forward and inverse kinematic, implemented on a micro-controller.

7. The final solution must be able to operate in a variation of modes; cartesian space, joint space and a predefined point.

8. The servos of the final solution should have a linear control algorithm implemented.

9. The final solution has a mode selection system implemented for easy transition between the modes.

Chapter 5

System Description

In the following chapter the subsystems are explained. This considers both the physical hardware and the software.

In figure 5.1 a chart of the subsystems is showed along with their connections. The chart is divided up into two parts, the real world and the system.

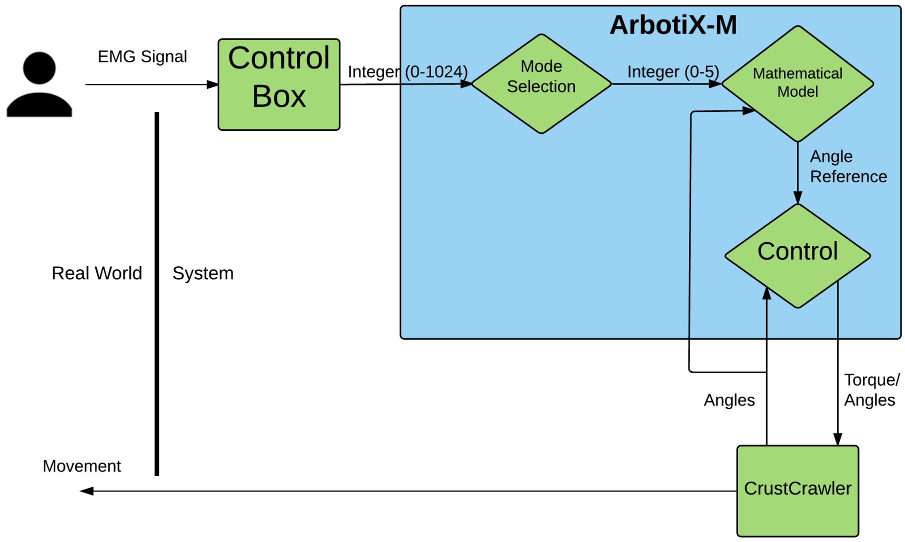


Figure 5.1: An overview of the subsystems. Rectangular boxes stand for hardware and diamond boxes stand for software.

The real world is the patient in his/her environment while the system represents all the different subsystems running in the ArbotiX-M and the CrustCrawler. The system contains the following 5 levels:

Control Box

The control box, which is explained in section 5.1.

Mode Selection
Section 5.2 describes how the interface works.

Mathematical Model
Here in subsection 5.3.4 the kinematics are used.

Control
In section 5.4 the control system theory is introduced, and the controller design of the CrustCrawler is explained.

CrustCrawler
The CrustCrawler, and its mathematical model is explained and examined in section 5.3, while the micro-controllers and circuit boards attached to the CrustCrawler are explained in section 5.5.1.

The control box is the place, where the EMG signals from the muscle contractions are recorded and received. From the control box, signals containing a value from 0-1024, are sent to the ArbotiX-M which contains a program with mode selection, where the patient can choose what he/she wants to do. Then the mode function send what the patient chose to the mathematical model.

According to what the patient picks in the Mode selection function, the math is done to perform the given action, which could be e.g. moving in Cartesian coordinates which includes both forward and inverse kinematics. Then the mathematical models outputs an angle reference, depending on the action needed to be performed.

The control function receives the values from the mathematical model, and converts them into reference torques for the servos on the CrustCrawler. These signals are then sent to the servos. The servos move a little and then send its current angle back to the controller. Here the current position is withdrawn from the initially set position, so it can be calculated how far the servos are off. The angles can also be sent to the mathematical modelling if necessary for new calculations.
The arrow from the CrustCrawler to the real world is made to show that the movement of the CrustCrawler can be seen, so the patient can react accordingly.

5.1 EMG Control Box

This section gives an explanation on how the EMG and accelerometer data are received and perceived by the ArbotiX-M micro-controller. This includes what range is used to get data that are easily translated into movement for the robot.

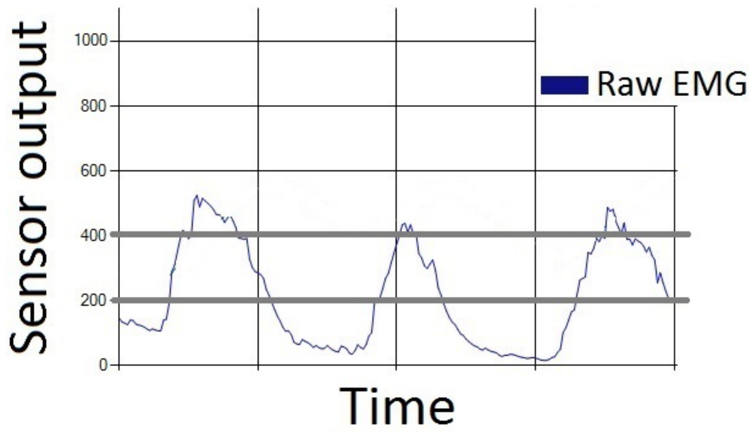


Figure 5.2: Example of a grey zone application. In this case, the servo would start moving, if the signal value crosses the upper threshold of 400. It would stop the movement, if the signal value drops under the lower threshold of 200.

Only how the signal is processed is looked into as the EMG control box itself is a standalone device and out of the scope for this project.

5.1.1 Signal Processing

The raw sensor values from the EMG channels can be described as a combination of sinus functions with high and low frequencies and amplitudes. In order to utilise those signals, whose amplitude is jumping up and down a lot, a threshold system needs to be designed. It would not be a robust solution to simply create a threshold on one solitary value, since the EMG curve could cross it up and down multiple times, if trying to turn the movement on/off, resulting in stutter movements of the robot. To eliminate that, a grey zone has been implemented, which basically utilizes different threshold value for turning the movement on and different one for turning the movement off. In other words, the signal value needs to cross a certain threshold value to activate the movement, but in order to deactivate it, the value needs to drop under different threshold, which is located lower than the original one. Low enough to prevent the EMG signal from accidentally dropping under the threshold. In that way, the stutter effect is eliminated. For graphical explanation (see figure 5.2). This threshold system with a grey zone method, is also used for the accelerometer values. This is used so small natural movement will not make the program switch stated.

Other method, than grey zones, would be e.g. applying a low-pass filter to the raw EMG readings. Low-pass filter helps to smooth the raw sensor signal so the amplitude does not jump up and down as much as it would with the raw signal, i.e. making the low-frequency part of the signal more significant compared

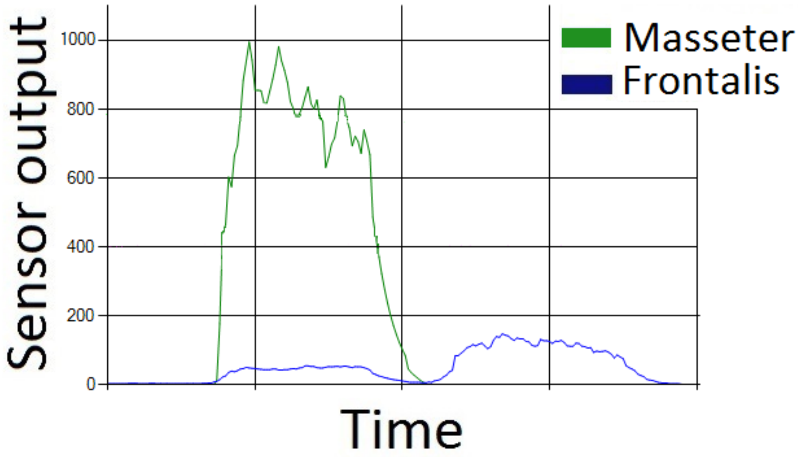
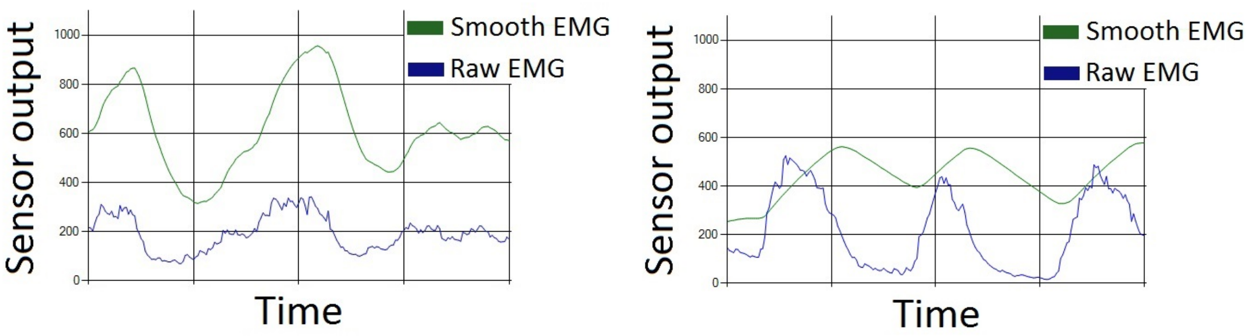


Figure 5.3: EMG readings recorded in the processing program. EMG1 curve stands for Masseter muscle, EMG2 plots Frontalis signals.



(a) 8% Rise

(b) 1.5% Rise

Figure 5.4: Comparison of two EMG low-pass frequency filters

to the rest of the signal (see figure 5.4 for comparison). This would reduce the risk of unwanted motion, if only one threshold between moving and not-moving was implemented. However, applying the filter also makes all the curve slopes less steep, which prolongs the rise- and drop time of the curve, which will not be convenient.

A simple low-pass filter formula could look something like this:

```
SmoothValue = SmoothValue − (LPF \cdot (SmoothValue − RawValue))
```

Where LPF is a constant in the interval (0;1] that specifies how much the amplitude can rise and fall between the smoothed and the raw value. The larger LPF is, the closer becomes the smoothed signal to the original raw signal.

In the end, it has been chosen to work with the grey zones, because it has been considered convenient for this project to preserve fast rise time, and drop time.

5.1.2 EMG Signal Test

In order to give those thresholds appropriate values, a test of EMG readings from the Masseter and the Frontalis muscles has been undertaken to see, how the readings from those two differ. As it was expected, both muscles emit signals of different strength and amplitude. That implies that each muscle needs their own grey zone. One for both would not work sufficiently.

The Masseter muscle provides very strong initial signals. When contracted, it is easy to surpass a value of 400. However it is very hard to hold the value this high over extended time. This is why the lower threshold needs to be very low.
Frontalis muscle produces weaker, but more stable reading, i.e. the initial overshoot is not that big. The grey zone for Frontalis is not as big as the one for Masseter. A visual comparison can be seen in figure 5.3.

To find a suitable solution, it would not be sufficient enough to only test one person. Therefore a test has been made on 4 persons. The electrodes are placed on the same position on the Masseter and the Frontalis on all 4 persons. These positions are illustrated in figure 5.5.

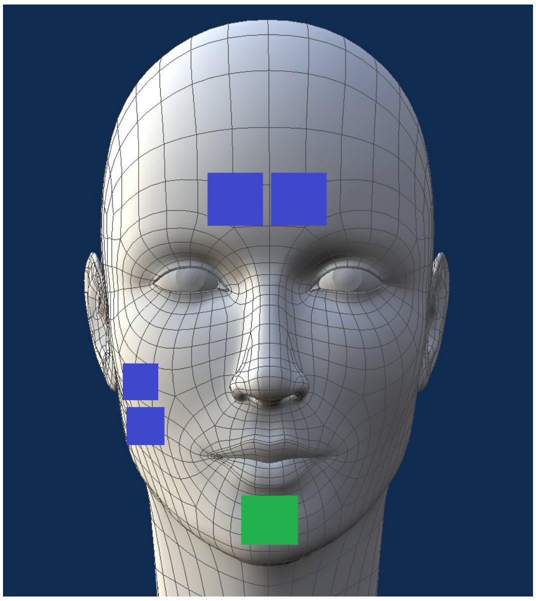


Figure 5.5: Illustration of where the 5 electrodes shall be placed. The 4 blue squares are the desired positions on the two muscles and the green is the reference which both channels should be connected to. [ImageShack 2015]

The test result can be seen in table 5.1, which shows that the 3 of the test subjects can control the manipulator with small changes to the initial values. It is only one respondent who has somewhat lower values. The accelerometer values were not tested, since the placement of the control box is always the same, hence the values will be same.
Before first use, the patient will therefore needs to calibrate the upper and lower boundaries for the EMG signals, for it to work correct.

	Masseter Lower	Masseter Upper	Frontalis Lower	Frontalis Upper
Emil	25	350	40	90
Rasmus	13	117	13	30
Biranavan	25	350	25	60
David	25	350	25	70

Table 5.1: An overview of the test result of the EMG signals.

It is also required of the patient that the pad of the electrodes are in clear contact with surface of the skin. otherwise it makes the signals harder to read and difficult to attach on the face.

This section described the EMG signal processing that occurs in ArbotiX-M, the control box, and the way to utilise EMG readings in this project. With these findings, it is possible now to receive and work with the signal values, as well as implementing the right algorithm, that utilises those values, in the code.

5.2 Mode Selection

For interface of the robotic arm, the patient is able to utilise several outputs that the EMG control box can provide. Among those are the two channels of EMG signals from Masseter and Frontalis muscles. This control box also contains a accelerometer which can be utilised as well, in order to make the final solution more convenient and user friendly. It is then decided that both EMG channels can be used for control of the motion itself, while outputs from a accelerometer can be used to navigate through a menu of different modes the software offers. These modes are the following:

Joint movement
: The patient should be able to control each joint individually, in order to get the CrustCrawler out of a difficult situation or to recover if it enters singularity.

Cartesian movement
: For a patient, the easiest way to navigate the CrustCrawler, is to do it in the

Cartesian space, therefore this mode is essential for the patient.

Predefined movement

For easy use, the CrustCrawler needs to have some predefined points. This is for an example, the point in front of the patients mouth, so drinking is not a difficult task.

Horizontal movement

For sake of convenience, one more mode is implemented, where the end-effector first goes from the position it was in before, to a horizontal plane, by rotating the 3^{rd} joint. Afterwards, the patient has the possibility to either rotate the whole manipulator around an axis, perpendicular to the base plane, or to rotate the end-effector vertically (around z-axis of the 2^{nd} joint), while keeping it in the horizontal position it reached initially. In other words, the angle of 3^{rd} joint becomes a function of 2^{nd} joint's angle.

Pause mode

Pause mode is a way to use the facial muscles without making the CrustCrawler move.

With the above kind of movements the patient would have the needed control over the CrustCrawler to perform the desired tasks. However, since it is two facial muscles that are being used, the patient will encounter certain problems when he/she wants to eat, drink or talk, since the Masseter is especially used for that. Therefore the pause mode is added. This mode will allow patients to eat, drink and talk without the CrustCrawler moving.

The accelerometer can be utilized by attaching the control box to the patient's head, thus making him/her able to control the output of the accelerometer by head movements. In this project, it is expected that the control box is wielded according to figure 5.7. The accelerometer of the control box provide three outputs, which are changing dependently on the position of the control box. When the box is attached to the patient's head as desired, the output of accelerometer changes, when the patient tilts his/hers head. A higher intensity of the head movement provides of course higher values, than a slow tilting does. Through testing, it was investigated which axis of the accelerometer can be used to control certain action, and the appropriate thresholds, crossing which the desired action is performed, were designed. When the control box is mounted on the head, as in figure 5.7, the z-axis of the accelerometer readings are most useful for recognizing the head tilting to the left, or right, because the value changes of that accelerometer are the most significant, when performing these actions. For the same reason, readings of the x-axis accelerometer were chosen for tilting head forwards and backwards. The significance of readings the accelerometer, considering those 4 actions, can be seen in figure 5.6, which shows the logic behind choosing z-axis for horizontal head movement, and x-axis for vertical head movement.

This gives the patient 4 different simple actions, provided by the accelerometer only, he/she can utilize for controls. Counting in also the EMG channels, the patient has 6 different inputs at his/hers disposal. The utilization of them is described further.

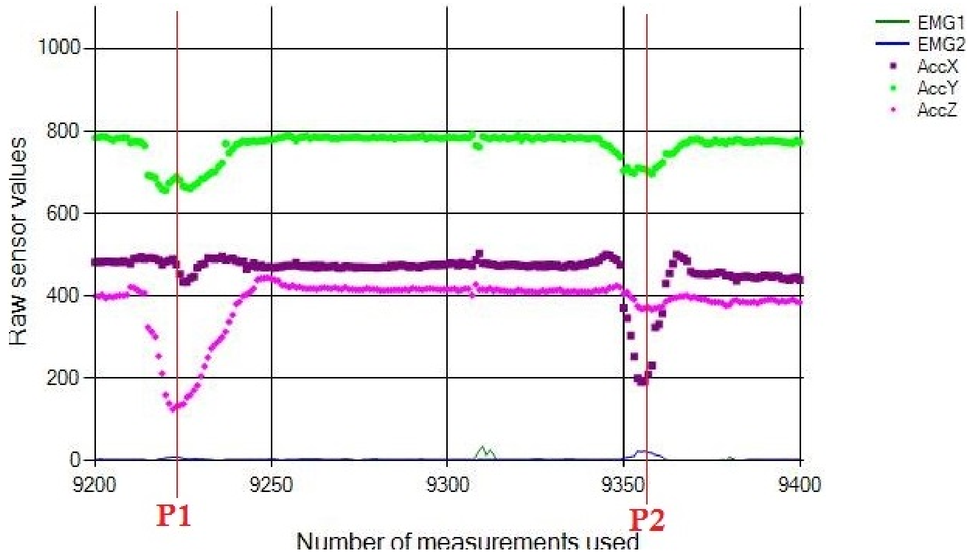


Figure 5.6: Plot of the accelerometer readings, when the control box is correctly attached to the head. At point P1 the change of values is caused by the patient flicking his/hers head right. At P2 the patient has flicked his/hers head forwards. Flicking the head in opposite directions would cause the reading curves to deviate in opposite directions as well.

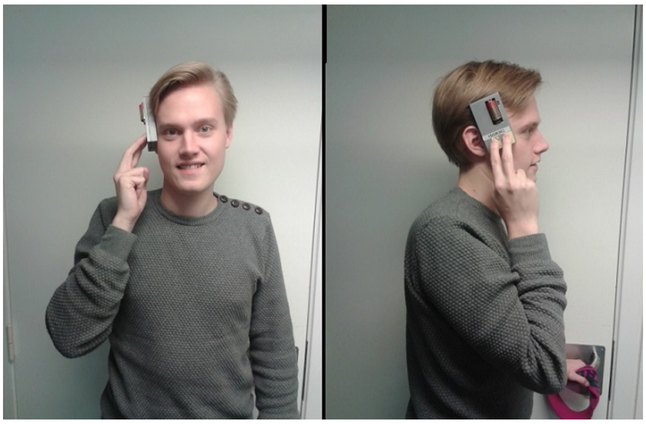


Figure 5.7: Correct position of the control box, for which the trigger thresholds were calibrated.

After the control box is in the position, the patient can start navigating in the interface by simple head movements. One has to start by deciding on a mode he/she wants to enter. The way a person changes between the five modes is by flicking his/hers head to the left, or right. By that, it is possible to skim through the mode selection. If the desired mode is selected, nodding the head forwards confirms the selection and the program enters the mode. Every mode then contains similar interface of its own. In the joint space, the actions a patient can do, are the following:

Changes between joints
In order to move the desired joint, the patient needs to be able to toggle between them. This is done again by flicking the head to the left, or the right side. The two servos in the end-effector will be considered as one joint, so it can open or close. flicking the head forwards aborts the joint movement mode and returns the patient back to mode selection.

Move the joints
The way the patient is going to operate the servo, is by contracting the Masseter muscle to move it clockwise and the Frontalis to move it counter-clockwise.

The Cartesian movement also has an inner interface, that works similarly. The positive and negative direction belongs to the world coordinate system when moving in the Cartesian space.

Change axes
With the two channels of EMG signals, it is not possible to move in a 3D space, therefore the patient needs to choose which axis he/she would like to move along at one time. This is done by flicking the head left or right,

in order to toggle between moving the end-effector along one of the X, Y, or Z axes. Like in the joint movement, flicking the head forwards makes the program jump back to mode selection.

Move along the axes
Contraction on the Masseter moves the CrustCrawler's end-effector in the positive direction. Contraction on the Frontalis will thereby move the end-effector in the negative direction.

Predefined positions mode has been created to carry out repetitive tasks with ease, e.g. drinking. When the patient already grasps a glass of beverage, it can be presumed that his/hers drinking position always can be the same. For that reason, it is easier to have a predefined function that places the end-effector with the beverage in the "drinking position", than to have the patient to manually create that set-up by himself/herself. By default, there are two predefined positions (Home and Mouth) included in the program. "Home" is moving the CrustCrawler to its initial position (see the kinematics in section 5.3.4) and "Mouth" is the mentioned placement of the end-effector in the position to drink. Patient can switch between predefined positions by flicking the head left and right. By flicking the head forwards, the selected position is initialized. When the CrustCrawler reaches the selected position, it automatically switches to Pause mode, thereby the patient can drink/eat right away.

The pause mode can be also accessed from the mode selection manually. This mode simply immobilizes the arm, so the Patient does not have to be afraid of undesired behaviour of the arm, that could be possibly caused by eating, talking, laughing, or head movements, like nodding or turning. The pause mode can be aborted by flicking the head forwards, and then flicking it backwards. This action brings up the mode selection again.

The design of this mode selection is intuitive because the way the patient chooses the options is recurring throughout the program. This makes it easy to get used to navigating in the system. With the interface working, it is now possible to utilize it to control the robot's movements.

5.3 CrustCrawler and Mathematical model

The robotic manipulator used for this project, is a 3 DoF robot. The robot is from the company CrustCrawler, and is from a series called Pro-series. The Pro-series is a concept that enables the buyer to order separate units and then build their own robotic manipulator. The units can be bought in a set or in single pieces. This concept is well known, mostly from LEGO-mindstorm. This is ideal for customers that want to custom-build their own robotic manipulator, meeting their desired length and torque requirements. On figure 5.8 shows the manipulator constructed for this project.

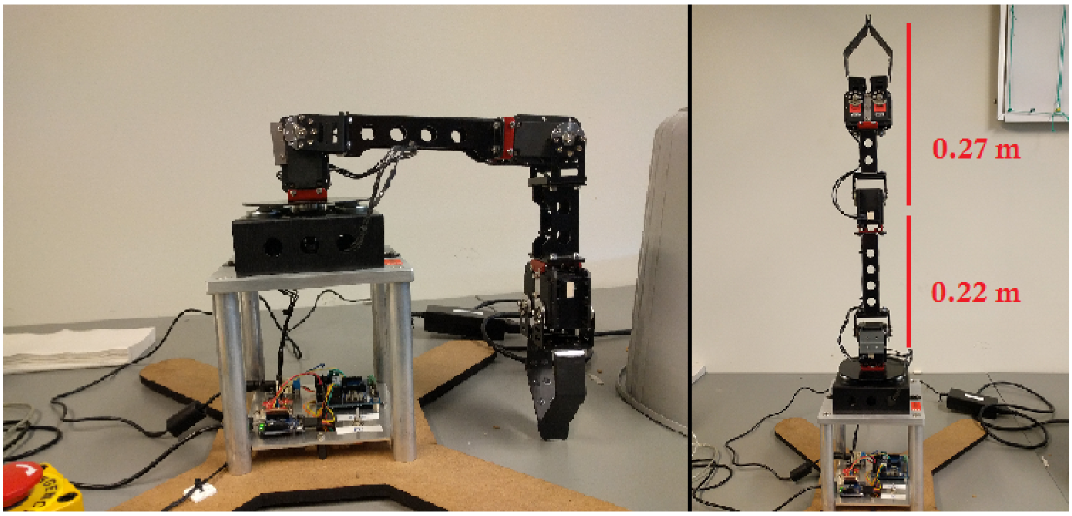


Figure 5.8: CrustCrawler in the setup for this project.

This section gives a description of the CrustCrawler robot, and furthermore establishes a mathematical model of the robot, in form of a kinematic model. Some of the knowledge used in the description of the robot, will be used in the mathematical model.

5.3.1 Workspace

As seen in figure 5.8, the length of the 1^{st} link is 0.22 m, and the length of the 2^{nd} link is 0.27 m. This gives the manipulator a maximum reach of approx. 0.49 m, if both links are placed in a straight line. Figure 5.9 and 5.10 shows a sketch of the robot, and its maximum reach workspace from front and top view.

5.3.2 Servos

On the given CrustCrawler for this project there are a variation of different servos. In total, there is 5 modules that enable the possibility of movement for the robotic manipulator.

- MX-106 servo (2^{nd} joint)
- 2 × MX-64 servo (1^{st} and (3^{rd} joint)
- 2 × MX-28 servos (Gripper)

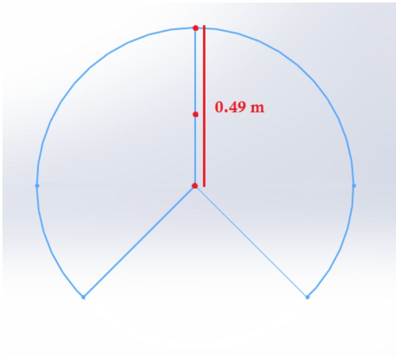


Figure 5.9: Front view.

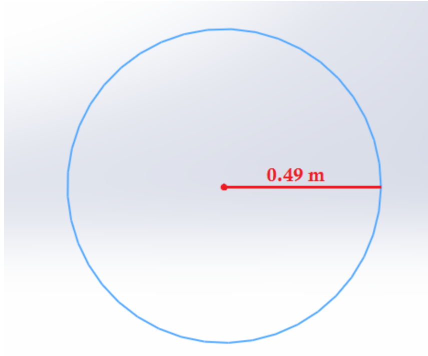


Figure 5.10: Top view.

Figure 5.11: MX-28

Figure 5.12: MX-64

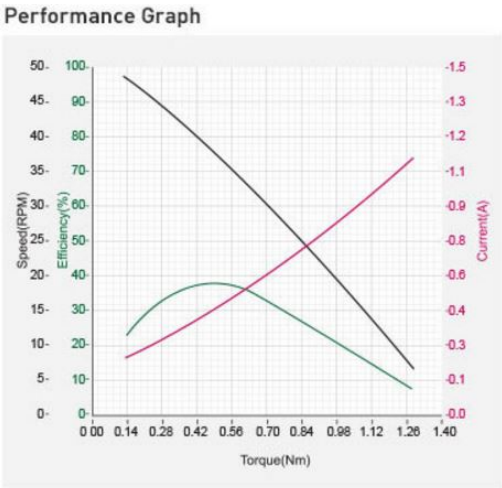
Figure 5.13: MX-106

Since the CrustCrawler is custom-build, pricing is a factor, when building the manipulator.

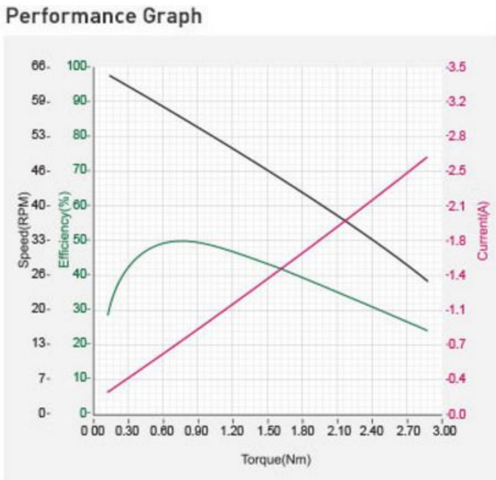
- The turn-table cost $640,- (4343,- DKK).
- The MX-106 cost $500,- (3511,- DKK).
- The MX-64 cost $305,- (2141,- DKK).
- The MX-28 cost $230,- (1615,- DKK).

There are two MX-28 servos, each of them are connected to a jaw, which is forming the gripper. The manipulator is connected to an EMG control box through an ArbotiX-M micro-controller. This makes it possible to move the manipulator with EMG signals measured from surface muscles (Masetter and Frontalis). When looking at figure 5.11, 5.12 and 5.13, it is noticeable that all of the servos look alike.

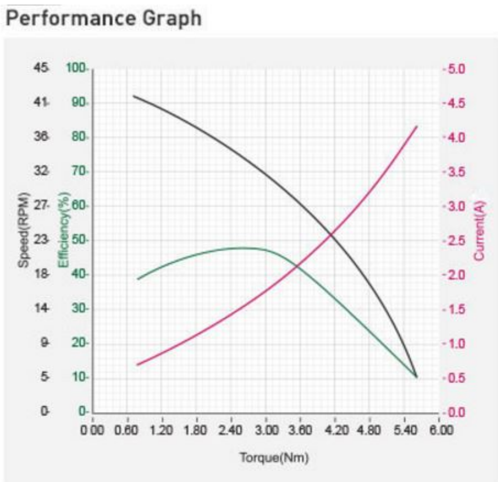
To understand the differences between the servos, it is necessary to look at their datasheets. The datasheets contain all the relevant information about each servo. A lot of the information is the same on all of the datasheet. Each datasheet contains



(a) Datasheet for the MX-28. [Envisage 2014]



(b) Datasheet for the MX-64. [Envisage 2014]



(c) Datasheet for the MX-106. [Envisage 2014]

Figure 5.14

a graph showing the connection between torque and current/speed/efficiency.

The speed graph in figure 5.14b shows that MX-64 is the only servo with an almost linear decrease. In figure 5.11 and 5.13 the MX-28 and MX-106 are more curved. MX-64 is also the one with the most linear growth in use of current. It is important to notice that the MX-106 figure has a bigger torque range, since it is capable of producing a higher torque.

When comparing the three servos, three torque values are estimated from figures 5.14a, 5.14b and 5.14c. In each torque value, the servos will be compared in efficiency, current and speed.

- MX-28 peak-point is around 0.49 Nm.
- MX-64 peak-point is around 0.75 Nm.
- MX-106 peak-point is around 2.70 Nm.

The table 5.2 makes it possible to understand the difference between the three servos.

The table shows that the MX-106 is better at producing torque compared to MX-64 and MX-28. The MX-28 torque-value is not high enough to be compared with the MX-106 since it is not capable of producing a torque above 1.26 Nm. The MX-106 graph in figure 5.14c, does not show the values for a torque of 0.49 Nm. This is why some of the values illustrated in figure 5.2 are assumed, because it is not known how the graphs progress. The MX-28 is not desirable to replace, because of its low weight which is of an advantage. Furthermore it would increase the expenses to replace it with one of the other servos.

5.3.3 Explanation of Resolution

The Dynamixel-MX servos have a 360° position control with a 12-bit (4096) resolution. This is the way of translating the degree system for the servo, i.e. if the input value of the joint angle in the program is changed by 11.4, then the real angle is

Servo	MX-28	MX-64	MX-106
Wait (g)	77	135	153
Dimension (mm)	35.6 x 50.6 x 35.5	40.2 x 61.1 x 41	40.2 x 65.1 x 46
Operation voltage (V)	10-14.8	10-14.8	10-14.8
Stall Toque (N.m)	2.5 at 12V	6.0 at 12V	8.4 at 12V
Stall current (A)	1.4 at 12V	4.1 at 12V	5.2 at 12V
No Load Speed (RPM)	55 at 12V	63 at 12V	45 at 12V
Effecientcy (%) when Touque 0.49NM	37	57	can be assumed 32
Current (A) when Touque 0.49NM	30	15	can be assumed 12
Speed (RPM) when Touque 0.49NM	76	90	can be assumed 95
Effecientcy (%) when Touque 0.75NM	32	50	39
Current (A) when Touque 0.75NM	41	18	15
Speed (RPM) when Touque 0.75NM	57	86	91
Effecientcy (%) when Touque 2.70NM	not counterable	26	48
Current (A) when Touque 2.70NM	not counterable	69	32
Speed (RPM) when Touque 2.70NM	not counterable	43	73

Table 5.2: Comparison of the MX-28, MX-64 and MX-106 servos.

going to change by 1 degree (see table 5.3), which means that the servo is operating with precision of approx. For detailed values see table 5.3 This is a translation that is used for programming a movement of a servo. Instead of specifying the actual degrees in the program, the position must be specified in the resolution units of the given servo.

12-bit resolution unit	Degrees
1	0.088
11.4	1
1024	90
2048	180
3072	270
4096	360

Table 5.3: The resolution of Dynamixel servos.

5.3.4 Kinematic Model

A kinematic model, is a mathematical model of movement without taking force into account. The purpose of this model is to easily transition between joint space and Cartesian space. Joint space is when all 3 angles of the CrustCrawler are known and with that as input can give a position in 3-dimensional space. Going from a position in the 3 dimensional space to 3 angle values is necessary as the patient of the final solution should be able to operator in both joint space and Cartesian space. The kinematics that describes the transition from joint space to Cartesian is called forward kinematics while going from Cartesian space to joint space is the inverse kinematics.

Forward Kinematic

The purpose of this section is to establish a forward kinematic model of the robot. The model has to determine a position from 3 given joint angles. The model will be used in 5.3.5 to verify that the inverse model, established in section 5.3.4, is correct. In order to make the modelling of the forward kinematics less complex, 6 frames will be placed on different parts of the robot and the relation between them will be described. These 6 frames will only be used to determine 6 matrices that can be used as the forward kinematic model.

A frame is placed on each of the 3 axes of the servos of the robot as illustrated in figure 5.15. These 3 frames are called the 1^{st}, 2^{nd} and 3^{rd} frame. The 3 other frames are called the Base frame, the 0^{th} frame and the end-effector frame. The

Base frame is static while the other frames are moving with respect to the Base, except from the 0^{th} frame which has a pure translation, as seen from the Base frame, and therefore does not move either.

The frame of the end-effector is placed between the jaws of the gripper, and it is the position of this frame's origin, as seen from the Base frame, the forward kinematic model determines. The frames and the dimensions between the frames are illustrated in figure 5.16.

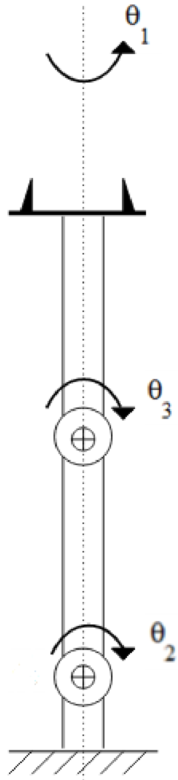


Figure 5.15: Illustrates the CrustCrawler robot and the 3 axes which the robot can rotate around.

When the axes are identified, the frames determined, and the dimensions between the frames measured, the Denavit-Hartenberg procedure can be used to calculate the forward kinematics and obtain the transformation matrix, ${}^{B}_{E}T$. [Craig 2014]

From the Denavit-Hartenberg parameters the transformation matrix ${}^{B}_{E}T$ is determined, by finding ${}^{B}_{0}T$, ${}^{0}_{1}T$, ${}^{1}_{2}T$, ${}^{2}_{3}T$, ${}^{3}_{E}T$ and multiplying them.

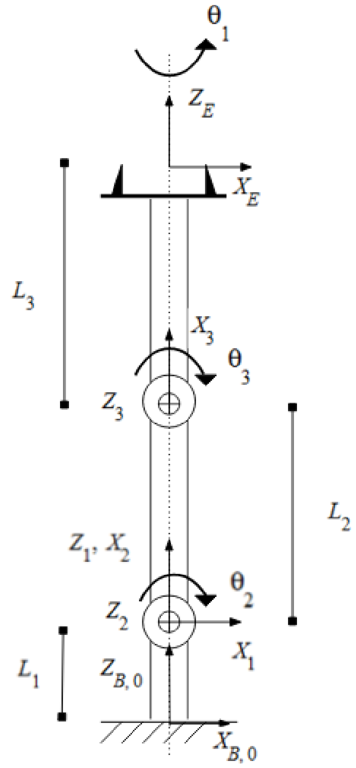


Figure 5.16: Illustrates the five frames from the 0th frame to the frame of the end effector. Here $L_1 = 110, L_2 = 220$ and $L_3 = 270$.

$$
{}^{B}_{0}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 110 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^{1}_{2}T = \begin{bmatrix} c\left(\theta_2 - \frac{\pi}{2}\right) & -s\left(\theta_2 - \frac{\pi}{2}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\left(\theta_2 - \frac{\pi}{2}\right) & -c\left(\theta_2 - \frac{\pi}{2}\right) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 220 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad
{}^{3}_{E}T = \begin{bmatrix} 1 & 0 & 0 & 270 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{B}_{E}T =^{B}_{0} T^{0}_{1}T^{1}_{2}T^{2}_{3}T^{3}_{E}T \tag{5.1}
$$

With this equation 5.1 it is possible to go from joint space to Cartesian space.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1^{st}	0	0	140	θ_1
2^{nd}	$\frac{\pi}{2}$	0	0	$\theta_2 - \frac{\pi}{2}$
3^{rd}	0	220	0	θ_3
E	0	270	0	0

Table 5.4: Denavit-Hartenberg parameters of the CrustCrawler robot.

Inverse Kinematic

Since the forward kinematic model cannot be directly used to give the robot a desired pose and position, an inverse kinematic model is determined. This model takes a X_P, Y_P and a Z_P position taken from the Cartesian space as input and from that can calculate 3 joint angles. In the inverse kinematics, the amount of degrees each joint has to rotate to give the end-effector a given position in Cartesian space is determined as a θ_i value for each of the joint axes. It can therefore be assumed that a specified position X_P, Y_P, Z_P is known.

For this project a geometric approach is used to obtain the set of joint angles. The different approaches are explained further in this sub-subsection.

θ_1 is a rotation around the Z-axis of the 0^{th} frame, as illustrated in figure 5.17. Due to this X_P and Y_P are the only parameters that play a role when determining θ_1.

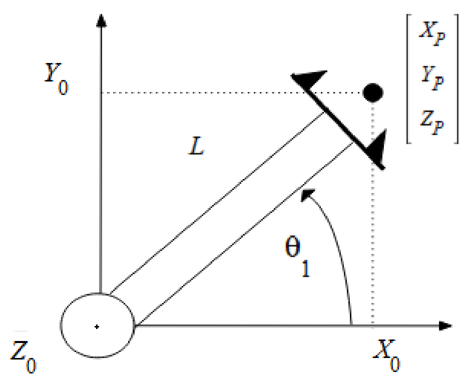


Figure 5.17: Illustrates the CrustCrawler seen from above and the rotation of θ_1.

θ_1 is found by the specified position X_P, Y_P, where an equation for X_P and Y_P can be made, and θ_1 can be isolated.

$$
\begin{aligned}
X_P &= cos(\theta_1) \cdot L \\
Y_P &= sin(\theta_1) \cdot L \\
&\Downarrow \\
\frac{Y_P}{X_P} &= tan(\theta_1) \\
&\Downarrow \\
\theta_1 &= arctan\left(\frac{Y_P}{X_P}\right)
\end{aligned} \tag{5.2}
$$

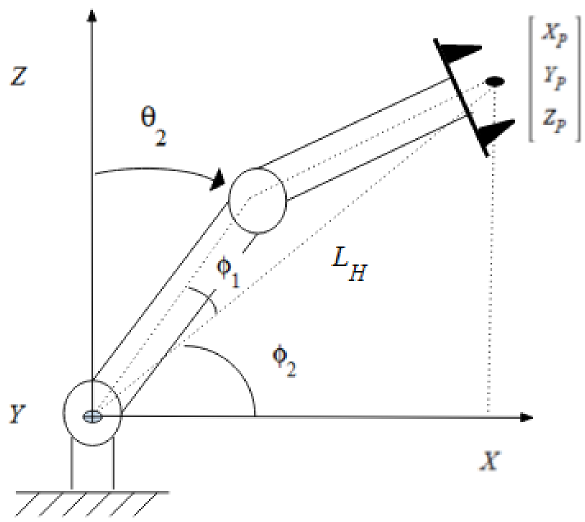


Figure 5.18: Illustrates the CrustCrawler seen from the side and a θ_2 rotation.

Figure 5.18 illustrates a θ_2 rotation. The length of the links and the specified point X_P, Y_P, Z_P are known, a geometric analysis of figure 5.18, is used to obtain ϕ_1 and ϕ_2. The equation for ϕ_1 is found by using the cosine relations. The equation for ϕ_2 can be found by using the same procedure as equation 5.2 since the length, L_H, of the vector to the end effector can be calculated.

$$\phi_1 = arccos\left(\frac{L_1^2 + X_P^2 + Y_P^2 + Z_P^2 - L_2^2}{2 \cdot L_1 \cdot \sqrt{X_P^2 + Y_P^2 + Z_P^2}}\right) \tag{5.3}$$

$$\phi_2 = arctan\left(\frac{Z_P}{\sqrt{X_P^2 + Y_P^2 + Z_P^2}}\right) \tag{5.4}$$

$$\Downarrow$$

$$\theta_2 = \frac{\pi}{2} - \left(arccos\left(\frac{L_1^2 + X_P^2 + Y_P^2 + Z_P^2 - L_2^2}{2 \cdot L_1 \cdot \sqrt{X_P^2 + Y_P^2 + Z_P^2}}\right) + arctan\left(\frac{Z_P}{\sqrt{X_P^2 + Y_P^2 + Z_P^2}}\right)\right) \tag{5.5}$$

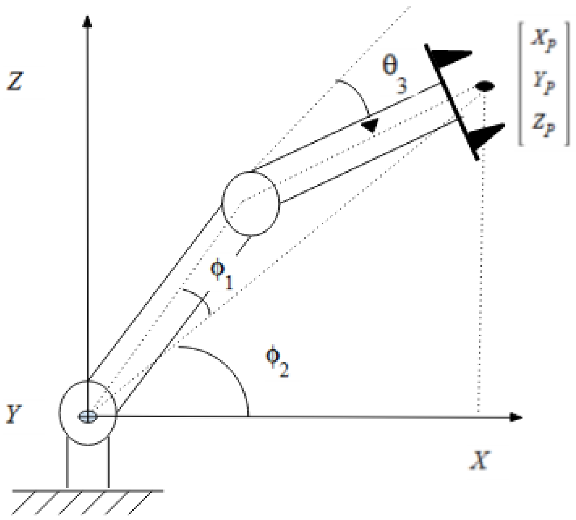


Figure 5.19: Illustrates the CrustCrawler seen from the side and the rotation of θ_3.

A geometric solution for θ_3 is obtained by figure 5.19 using the cosine relation and subtracting the value from π:

$$\theta_3 = \pi - arccos\left(\frac{L_1^2 + L_2^2 - (X_P^2 + Y_P^2 + Z_P^2)}{2 \cdot L_1 \cdot L_2}\right) \tag{5.6}$$

Thus, the inverse model of the robot and the equations for the 3 joint values are the following:

$$\theta_1 = arctan\left(\frac{Y_P}{X_P}\right) \tag{5.7}$$

$$\theta_2 = \frac{\pi}{2} - \left(arccos\left(\frac{L_1^2 + X_P^2 + Y_P^2 + Z_P^2 - L_2^2}{2 \cdot L_1 \cdot \sqrt{X_P^2 + Y_P^2 + Z_P^2}}\right) + arctan\left(\frac{Z_P}{\sqrt{X_P^2 + Y_P^2 + Z_P^2}}\right)\right) \tag{5.8}$$

$$\theta_3 = \pi - arccos\left(\frac{L_1^2 + L_2^2 - (X_P^2 + Y_P^2 + Z_P^2)}{2 \cdot L_1 \cdot L_2}\right) \tag{5.9}$$

5.3.5 Verification of the Inverse Model

This subsection contains a test of the inverse model established in subsection 5.3.4, which verifies that the model is correct. The forward model, established in section 5.3.4, is used as a reference for the inverse kinematics. The test is done as following:

1. A specified set of joint values θ_s is chosen.
2. A position P_d is calculated by the set of joint values and the forward kinematics model of the robot.
3. The position obtained in previous item is put in the inverse model of the robot, and a set of 3 joint values θ_o is obtained.
4. The new set of joint values is put in the forward model and a position P_t is obtained. If $P_d = P_t$ in all the test the inverse model is verified.

Test Nr:	$\theta_s(\theta_1, \theta_2, \theta_3)$	P_d	$\theta_o(\theta_1, \theta_2, \theta_3)$	P_t	$P_d = P_t$
1	$(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2})$	$(0, 220, -270)$	$(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2})$	$(0, 220, -270)$	✓
2	$(1, 2, 2)$	$(-2.3, -3.6, -268)$	$(-2.14, 4.3, -2)$	$(-2.3, -3.6, -268)$	✓
3	$(0.33, 0.5, 1)$	$(18.9, 28.6, 57.2)$	$(0.33, 0.5, 1)$	$(18.9, 28.6, 57.2)$	✓
4	$(0.5, \frac{\pi}{2}, 1)$	$(321, 175.4, -227.2)$	$(0.5, \frac{\pi}{2}, 1)$	$(321, 175.4, -227.2)$	✓
5	$(\pi, \frac{-\pi}{2}, \frac{\pi}{2})$	$(220, 0, 270)$	$(0, -0.2, \frac{\pi}{2})$	$(220, 0, 270)$	✓

Table 5.5: A test to verify that the inverse model from subsection 5.3.4 is correct.

In table 5.5, all the tests showed that $P_d = P_t$, which means that the inverse model, made in subsection 5.3.4, is verified. The test did also show that not all $\theta_s = \theta_o$, which is because there exists more than 1 solution to each inverse kinematic problem, as illustrated in figure 5.20. It applies that if a point P_d is within workspace of the robot, not in its maximum reach and if Z and Y is not equal to 0, there exist 4 solutions to the inverse kinematic problem. These 4 solutions are illustrated in table 5.6. However the different solutions will not be taken into account while creating the software for this project, for it has been considered not necessary.

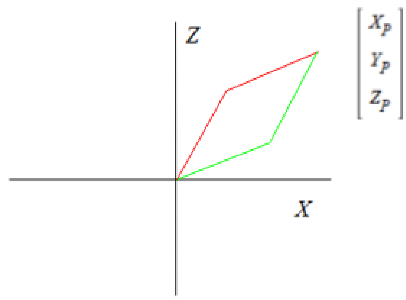


Figure 5.20: Illustrates how there can be more than 1 solution, to a inverse kinematic problem.

Number of solution	θ_1	θ_2	θ_3
1	θ_1	θ_2	θ_3
2	θ_1	$\phi_1 - \phi_2 + \frac{\pi}{2}$	$-\theta_3$
3	$\theta_1 + \pi$	$-\theta_2$	$-\theta_3$
4	$\theta_1 + \pi$	$\phi_2 - \phi_1 - \frac{\pi}{2}$	θ_3

Table 5.6: Shows the different solutions, and how to obtain them, taking point in solution 1 which is the solution obtained in subsection 5.3.4.

This section has introduced the technical information about the CrustCrawler, which is to be used in the final product implementation. The section has also presented the mathematical model of the CrustCrawler, which is necessary in more complicated moving patterns, like linear movement of the end-effector.

5.4 Control System

A control system is a device, set of devices, or in this case an algorithm which commands and regulates behaviour of other system. First general type of a control system is an open-loop, which is illustrated in figure 5.21. In this case of open-loop control, the output is generated on a given input, which comes to the system. E.g. a mobile robot that is not equipped with sensors and needs to get to a certain point. Knowing the speed of the robot and the distance between the robot and the point, it is possible to calculate the time the robot needs to drive to get to the point. This type of command is called open-loop control system, because the amount of time the robot drives is not based on the actual position of the robot. If something makes the robot go out of course, or it accelerates due to sudden change of the environment (e.g. slope down), or a calculation mistake occurs, the robot would go out of course which would make the robot miss the desired destination. A open loop system has no way to compensate through the errors, and make adjustments on its own. Open-loop control is great for systems which do not change too much, or if accuracy is not important.

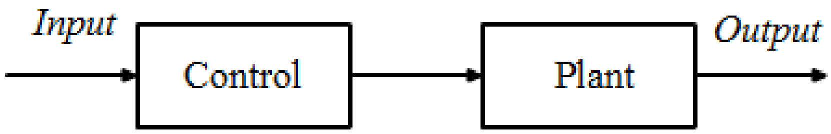


Figure 5.21: Standard flowchart for open-loop control system.

A closed-loop control system means that the system is regulated by a feedback of the output which goes back to the controller. This type of system is illustrated in figure 5.22. The given input (reference signal) goes to a system (plant), which derives an output through a given transfer function, and the gain of a controller. That output is then read by a sensor, and by comparing to the initial reference value, an error can be calculated. The error value is then received by control of the system, and further input is modified accordingly.

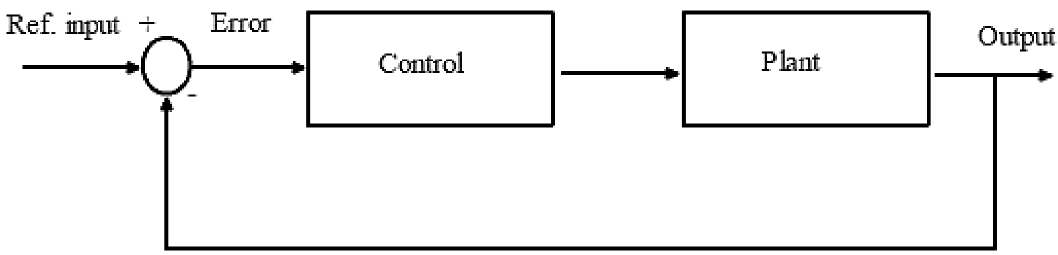


Figure 5.22: Standard flowchart for close-loop control system.

The purpose of this section is to create and implement a Single Input Single Output (SISO) control system for the CrustCrawler robot. This is done in order to make a custom controller. When the controller is implemented in the ArbotiX-M a test is made for each joint and evaluated.

5.4.1 Model of the Plant

As illustrated in figure 5.22, a plant is needed in a model of the control-loop. The plant consists of a mathematical established function that relates the input of the plant to the output, this function is called a transfer function of the plant. The plant is an overall system that can consist of numerous subsystems, as illustrated in figure 5.23. The plant of the control-loop, used for this project consists of two subsystems; an electrical system that receives a voltage as an input. A mechanical system that receives a torque (τ) as an input and returns an angle (θ), as an output. However, in this project it is assumed that the electrical system is provided with a voltage instantly, it is therefore expected that the electrical system does not have a significant impact on the control system. Due to this assumption the electrical subsystem is ignored in the modelling of the control system and only the mechanical subsystem is modelled. The mechanical model can later be used to make an approximated stability check for each joint of the robot, and thereby determine a value for the controller.

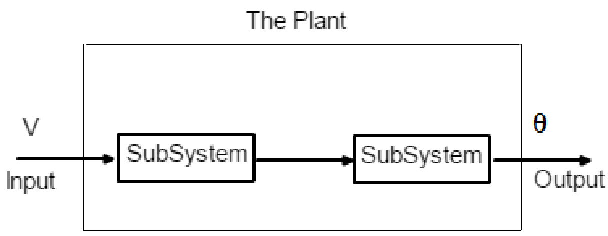


Figure 5.23: The whole plant and the subsystems within it, inputs and outputs.

While the electrical system is ignored, the input to the plant is now a torque, and the output is an angle. It is known that the algebraic model of the plant has to be a function that relates the input to the output, which means that the plant has to be equal to the following:

$$\frac{\theta}{\tau} = Plant \tag{5.10}$$

It is assumed that the load of the mechanical system consists of a rotating mass, the inertia and a bearing friction (viscous damping). By Newton's 2^{nd} law it is known that sum of torque is equal to the inertia multiplied by the angular velocity:

$$\sum \tau = J \cdot \ddot{\theta} \tag{5.11}$$

Furthermore, it is assumed that torques affecting the system are a τ_M which is created by the servos, τ_D which is created by the viscous damping D and τ_L created by the gravity and a external load. The torques are illustrated in figure 5.24 and equation 5.12.

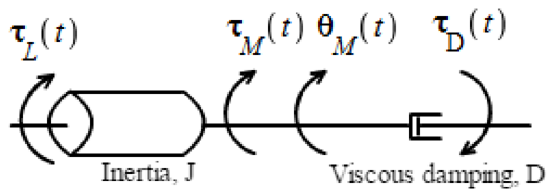


Figure 5.24: A free body diagram (F.B.D.) of a mechanical system. [NISE 2008]

$$\tau_J - \tau_D - \tau_L = J \cdot \ddot{\theta} \tag{5.12}$$

While the torque τ_D, increases with the speed, equation for τ_D can be set up [NISE 2008]:

$$\tau_D = D \cdot \dot{\theta} \tag{5.13}$$

Equation 5.12 is put in the Laplace domain. An equation 5.14 is obtained and a 2^{nd} order transfer function for the algebraic relation between the input and the output is isolated:

$$T_J - T_L - D \cdot s = J \cdot \Theta \cdot s^2 \Longrightarrow \frac{\Theta}{T_J - T_L} = \frac{1}{J \cdot s^2 + D \cdot s} \tag{5.14}$$

Now when the transfer function for the plant is identified, a full open-loop block diagram of the plant, can be established and illustrated in figure 5.25.

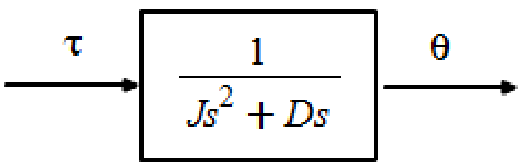


Figure 5.25: The open-loop of the plant.

5.4.2 Identification of the Constants J and D

The model established in this section is a linear approximation to a non-linear system. Meaning that the model are assumed not to be dependent on the force and the movement of the robot. J and D are due to this considered as constants. In this sub section J and D are identified.

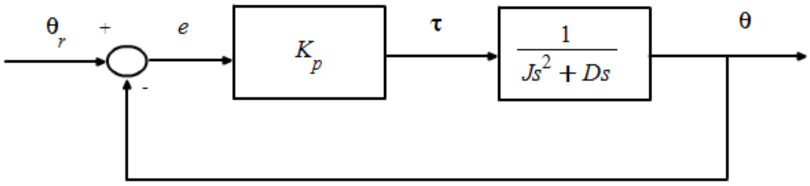


Figure 5.26: The closed-loop system.

In order to identify J and D the transfer function of the closed-loop system, illustrated in figure 5.26, must be found:

$$\frac{\Theta}{\Theta_r} = \frac{K_p \frac{1}{Js^2+Ds}}{1 + K_p \frac{1}{Js^2+Ds}}$$
$$\Downarrow$$
$$\frac{\Theta}{\Theta_r} = \frac{K_p}{Js^2 + Ds + K_p} \tag{5.15}$$

Equation 5.15 can now be set equal to the general equation of a 2^{nd} order function [NISE 2008]:

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{\frac{K_p}{J}}{s^2 + \frac{D}{J}s + \frac{K_p}{J}} \tag{5.16}$$

It is now known that:

$$\omega_n^2 = \frac{K_p}{J} \tag{5.17}$$

$$2\zeta\omega_n = \frac{D}{J} \tag{5.18}$$

Where the natural frequency, ω_n can be found from the following equation for the peak time [NISE 2008]:

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \tag{5.19}$$

The damping ratio, ζ can be found by knowing the percentage overshoot, $\%OS$ of a step response [NISE 2008]:

$$\zeta = \frac{-ln(\%OS/100)}{\sqrt{\pi^2 + ln^2(\%OS/100)}} \tag{5.20}$$

By measuring the step response directly from the robot, values for J and D can be found. The movements, from where a step response is obtained, are movements from the home position of a joint and a 90° rotation. The motion for the 3^{rd} joint is illustrated in figure 5.27.

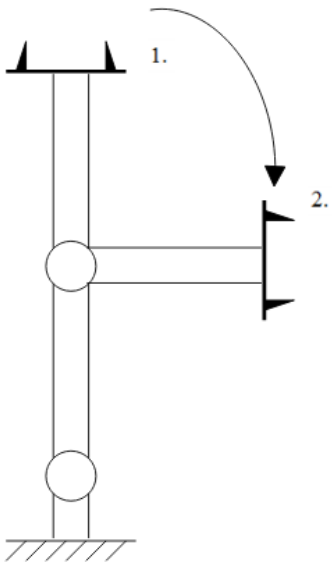


Figure 5.27: Movement used to obtain a step response for joint 3.

5.4.3 Example: Identification of J and D for the 3^{rd} Joint

In order identify J and D for the 3^{rd} joint, the damping ratio, ζ and the natural frequency ω_n has to be calculated. In order ζ, $\%OS$ is determined by figure 5.28 showing the step response for the 3^{rd} joint:

$$\%OS = 3.3032\% \tag{5.21}$$

Thereby the value ζ is found by equation 5.20:

$$\zeta = 0.7349 \tag{5.22}$$

Next ω_n is determined by using the peak time T_p which is determined from the step response and is ilustrated by the vertical dotted line in figure 5.28. T_p is estimated to be:

$$T_p = 0.3984\ s \tag{5.23}$$

Now ω_n can be identified using equation 5.19

$$\omega_n = 11.69 \tag{5.24}$$

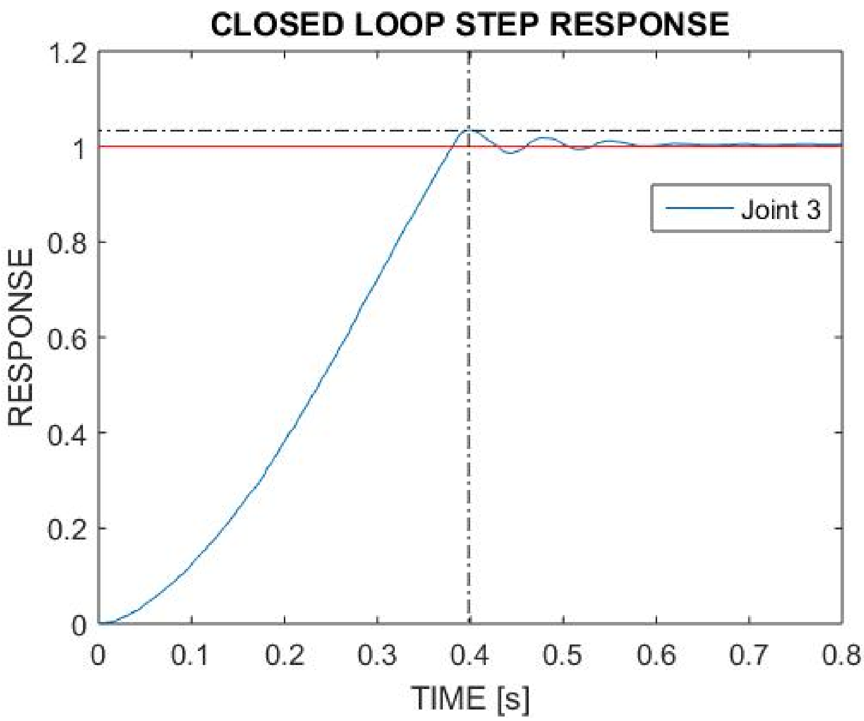


Figure 5.28: Step response for the joint 3.

When ω_n and ζ are found, J and D can be identified. J can be found using equation 5.17, where a K_p controller is used ($K_p = 1$):

$$J = 0.0074\ kg \cdot m^2 \tag{5.25}$$

Knowing J and ω_n, D can be identified using equation 5.18:

$$D = 0.12264\ Nm \cdot s \tag{5.26}$$

The different J and D- values identified for the 3 joints are illustrated in table 5.7.

	joint 1	joint 2	joint 3
J	0.0148 $kg \cdot m^2$	0.0137 $kg \cdot m^2$	0.0074 $kg \cdot m^2$
D	0.1728 $Nm \cdot s$	0.1859 $Nm \cdot s$	0.1264 $Nm \cdot s$
K_p	1	1	1

Table 5.7: The identified J and D for the 3 joints.

Verification of Model

This section looks into verification of the mathematical model. A step response plot, is made by the model for each joint, and compared to a step response, measured from the robot. If the step response of the model is an approximation of the step response measured from the robot, then the model is verified.

As it is illustrated in figure 5.29, all the models are an approximation of the real time measurement. The model is faster than the real live measurement. However, the approximation of the model is evaluated to be useful for further work.

5.4.4 Identification of P-gains

The controller used for this project is chosen to be a P-controller, although a P-controller produces a constant steady-state error. However it is evaluated that while the robot solution is controlled as a joystick, precision is not considered a significant aspect for the robot solution. Here the stability of the systems are investigated and the gain of the controllers determined.

Stability of the Systems

The investigation of the stability of the systems, are obtained by bode plots. Bode plots are obtained by an open-loop transfer function, and can be used to estimate a closed-loop response. A Bode plot is a graph of the frequency response of a system. When the controller is added, it changes the Bode plot, and there by changing the closed-loop response. A bode plot exist of two plots, a plot expressing the magnitude of the frequency, this is the magnitude plot and another one expressing the phase frequency response, this is called the phase plot. From a bode plot it is possible to investigate the stability of a system, by looking at the phase margin. Figure 5.30 illustrates how to obtain the phase margin, where the magnitude plot intersect with a 0 dB a line is drawn to the phase plot. The difference of response form where this line intersect with phase plot and -180, is the phase margin. A system is considered unstable if the phase margin is 0 or negative. The goal for the phase margin for this project is to have a phase margin between 45° and 60°.

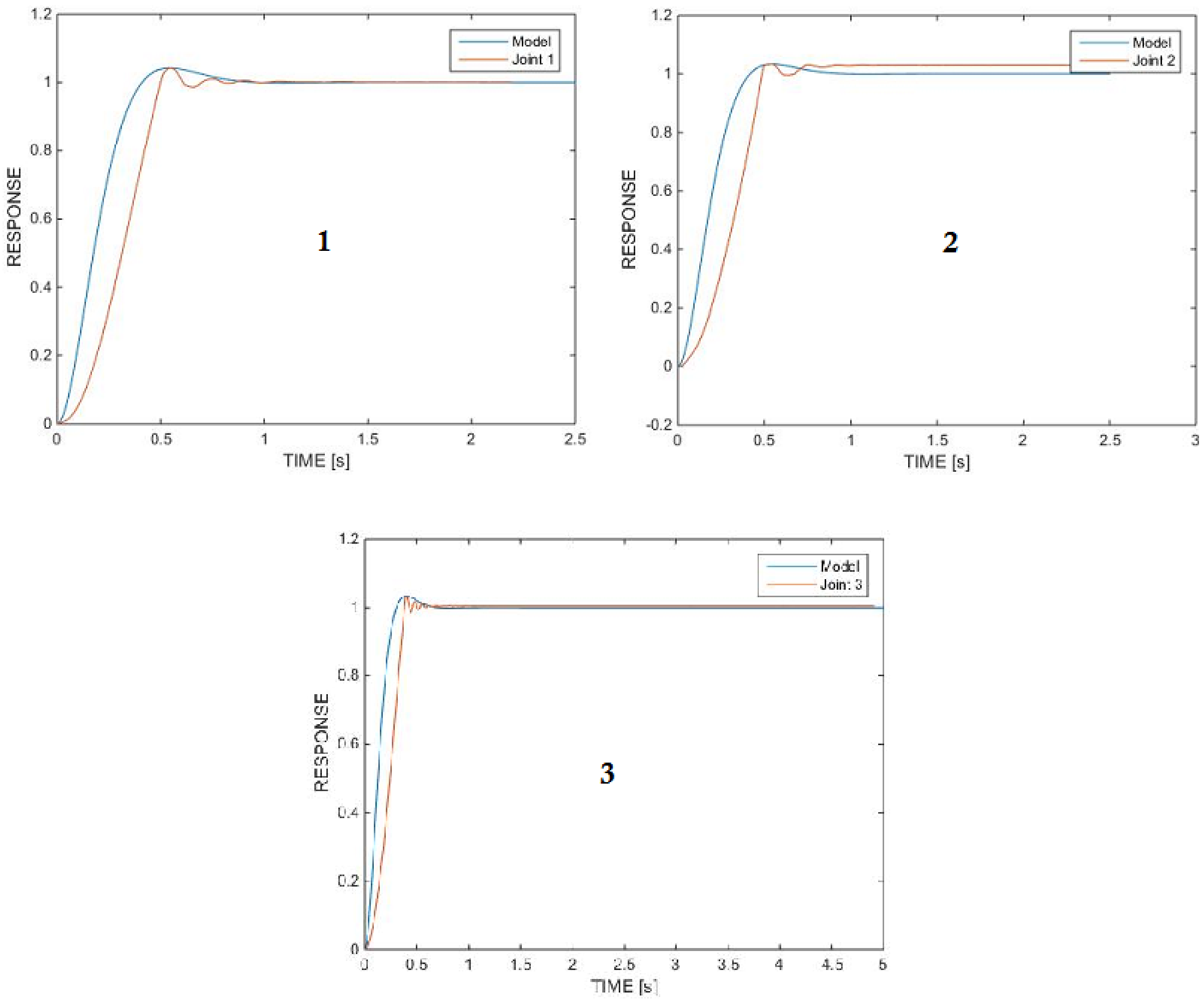


Figure 5.29: Verification model of joint 1, 2, and 3.

According to figure 5.31 the bigger phase margin, the smaller overshoot the given system has. But the phase margin also influences the settling time of the system proportionally (i.e. the bigger phase margin, the longer settling time). That is why it is important to find a phase margin that offers the fastest settling time with an acceptable overshoot.

The phase margin is designed such that it is possible to specify the gain of the controller, in this case the P-gain, K_p. The design process begins by investigating a bode plot of an open-loop transfer function for an arbitrary gain (usually 1). Then the desired P-gain can be found by reading the current phase margin, and searching for a magnitude value (in dB), in which the phase margin would change

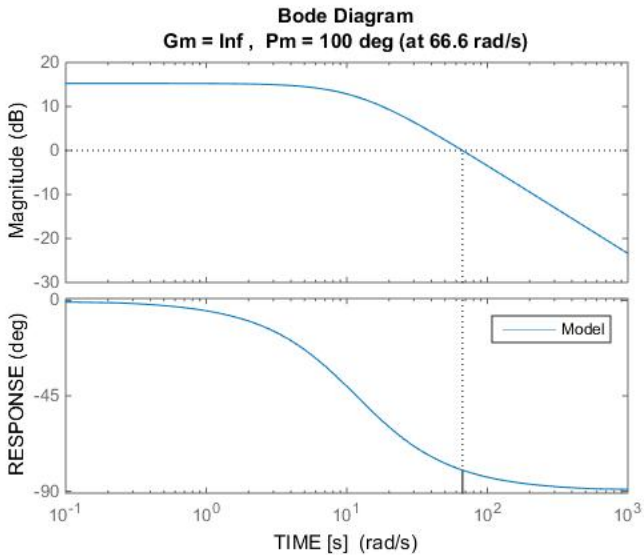


Figure 5.30: Example of how to determine a Phase margin.

to the desired value (e.g. 45°). Then the following equation is obtained:

$$20log_{10}(K_p) = x \tag{5.27}$$

Where K_p is the desired P-gain. All there is left to do, is to solve the equation for K_p.

$$K_p = 10^{\frac{x}{20}} \tag{5.28}$$

Stability Test for the 1^{st} Joint

In order to identify the phase margin for where the controller for the 1^{st} joint is stable, a Bode plot is made by the linear dynamic model for the 1^{st} joint, with a $K_p = 1$. ed in figure 5.33, the phase plot for the controller is not in between 45° and 60° however the response of the phase plot never intersect with -180. Due to this the system of the 1^st joint is stable no matter which P-gain is chosen for the controller. While a phase margin between 60° and 45° is desired for this project, and figure 5.33 illustrated that if the $K_p = 1$ the phase margin 65.7° this means that a two K_p values needs to be found. One K_p value that makes the phase margin 60° and another on making the phase margin 45°. To get a phase margin on 60°, 2.36 dB is found on figure 5.32 and put in equation 5.28.

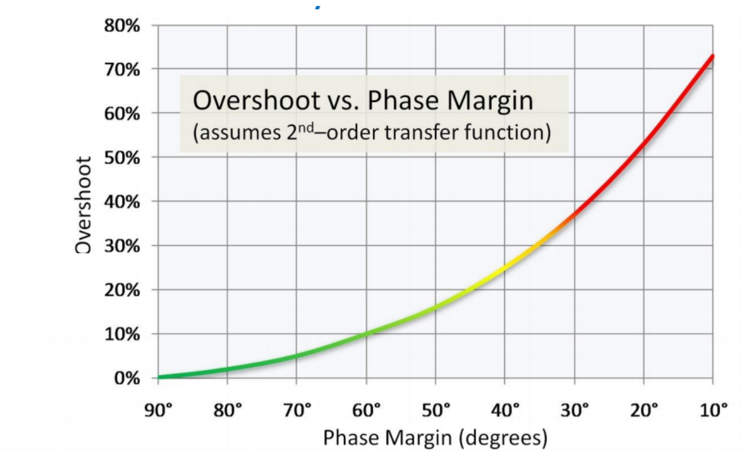


Figure 5.31: Overshoot - Phase margin dependence. [Nielsen 2015]

Phase margin	joint 1	joint 2	joint 3
45 deg	2.798	3.5481	3.051
60 deg	1.341	1.682	1.43

Table 5.8: The identified P-gains for a phase margin on 45 and 60 deg .

$$K_p = 10^{\frac{2.36dB}{20}} \tag{5.29}$$
$$\Downarrow$$
$$K_p = 1.312 \tag{5.30}$$

The same is done to find a K_p for which the phase is 45°, this time where 9.27 dB found on figure 5.32:

$$K_p = 10^{\frac{9.27dB}{20}} \tag{5.31}$$
$$\Downarrow$$
$$K_p = 2.907 \tag{5.32}$$

Now two new K_p values are obtained, for the 1^{st}, the other K_p values are illustrated table 5.8. The bode plot obtained for joint 2 and joint 3 are illustrated in appendix A.

Test of P-gain

I order to make the controllers work, P-gain, for each controller of the joints are determined. The results from table 5.8 are used to determine the P-gain, by a test

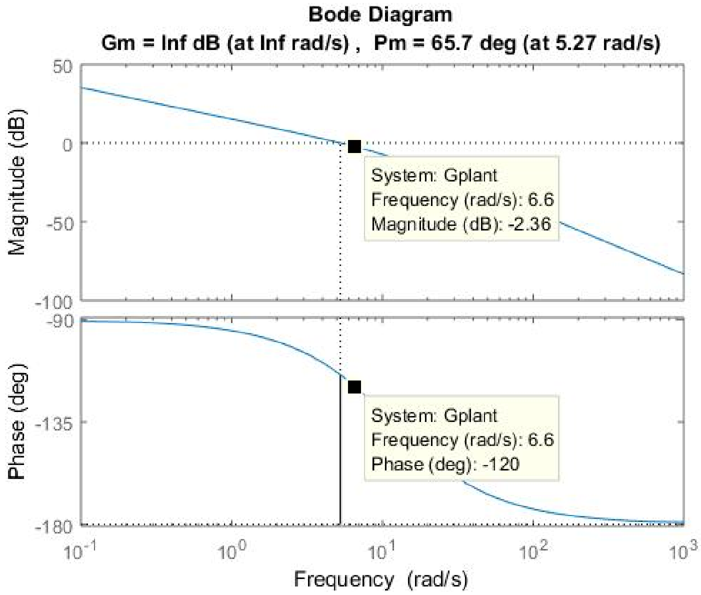


Figure 5.32: A bode plot for a the model for the joint 1 with $K_p = 1$.

done on the CrustCrawler. The criteria for an optimal P-gain, is a gain that make the robot move in a controlled speed, and is able to carry out a motion without oscillating. The test description is the following:

1. A P-gain is chosen for each joint such that the phase margin for each joint is 45°.
2. A program that make the robot move from the home-position to the position illustrated in figure 5.34, and back to the home position runs and -0.10 is subtracted from all P-gains after every movement cycle.
3. The above mentioned is done until the the P-gain is on a level where the phase margin is 60°.
4. The movements are evaluated and a P-gain for each joint are chosen.

The test showed that for joint 1, with P-gains above 2.2 gave oscillating movements. These oscillating were not apparent in the movement for a P-gain in the ranges of 1.3 - 1.9. For joint 3 the oscillating started with a P-gain above 3. An oscillation was present for movements of joint 2 when the P-gain was in the range of 1.682 - 2, it was clear that this oscillating was due the robot not being assembled firmly. In the range of 2 - 3.548 oscillating was not present for the joint 2.

The motion illustrated in figure 5.34, is chosen while it is within the area of normal operation for the robot. The P-gains obtained in this section are adapted for this motion. While the inertia of the robot is depending on the rotation of joint

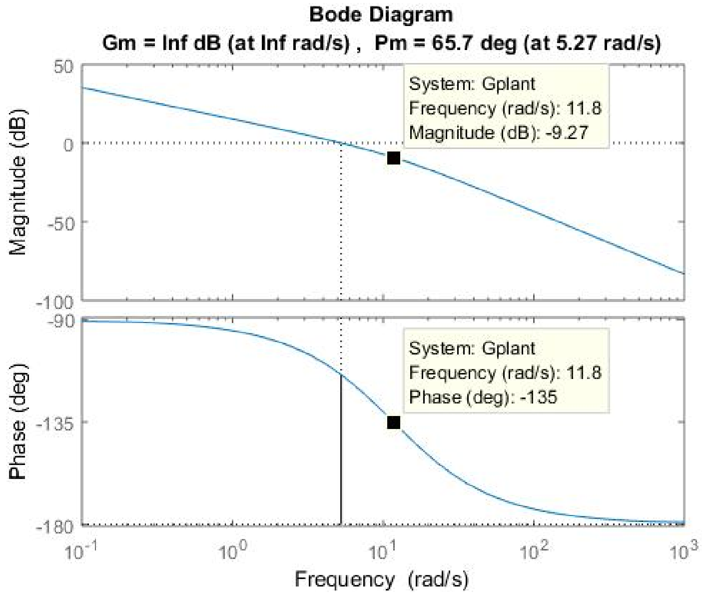


Figure 5.33: A bode plot for a the model for the joint 1 with $K_p = 1$.

2 and joint 3, and the external load τ_L, the behaviour outside the normal operation is not certain as non-linearities can happen. The model shown here is just a linear approximation to the non-linear system.

5.5 Electronics and software implementation

This section consists of a list of the electronic components used in this project, their functions, and the main functions of the software, implemented on the micro-controller.

5.5.1 Micro-controllers

In this subsection the electronics attached to the CrustCrawler are described. In figure 5.35 the four boards can be seen. This configuration includes ArbotiX-M micro-controller, UartSBee V4, RS-485 Breakout and a power divider. [Hansen 2015]

1. ArbotiX-M is a robot controller that works like an Arduino and can be programmed using the Arduino Integrated Development Environment (IDE). From this controller the software will be running.
2. UartSBee V4 is the board that takes the USB signal and turns it into a serial connection the ArbotiX-M controller can read. The UartSBee and ArbotiX-M

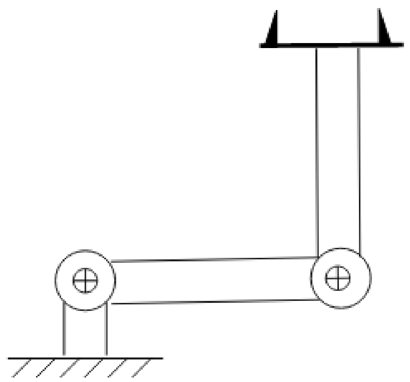


Figure 5.34: The position from where the robot goes from the home position. $\theta_1 = 90°$, $\theta_2 = 90°$ and $\theta_3 = -90°$.

are connected with a FTDI cable (USB to serial converter).

3. The RS-485 Breakout is needed because the ArbotiX-M supports TTL serial connection but the CrustCrawler does not, so the RS-485 Breakout is added for communication conversion.

4. The power divider connects the servo connections to the RS-485 Breakout as well as distributing power to all 5 servos.

There are two other essential things added to the system. The first one is the Xbee/PC switch. This is added because the Xbee and USB connection uses the same serial connections, thus can not be used at the same time. Therefore a switch is added, so the user does not need to mount and unmount the Xbees and UartSBee.
The second thing, is the red emergency stop button. When pressed, it cuts the power to the servos but not the ArbotiX-M micro-controller, thus the user must be aware that when unpressed the CrustCrawler will start running again if the power to the micro-controller is not cut.

The main reason that a micro-controller is used is because of its cost-effectiveness and small size. Compared to PC, the cost of the used micro-controller is very low (approx. 281,- DKK), and it is also a lot smaller than regular PC. Also, no other program is running than what the programmer has uploaded. If ROS, for instance, was used instead, then it will run on a PC with a Linux OS, like Ubuntu, installed. With the ROS solution, the OS can have an impact on how the system behaves, because the processor is handling many different tasks at the same time. With the micro-controller, the software uploaded through the Ardunio IDE, will be the

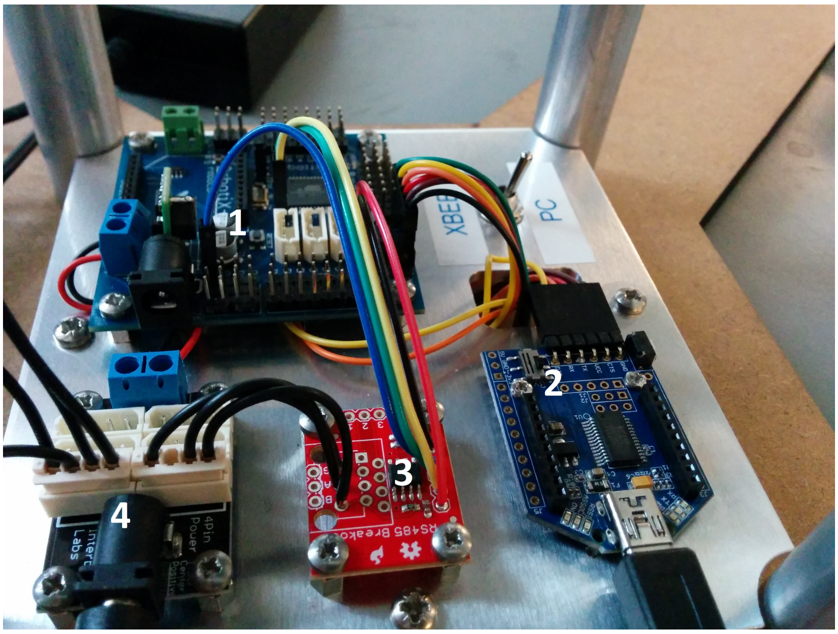


Figure 5.35: The four boards attached to the CrustCrawler

only program running and therefore lowers the risk of the OS prioritizing other processes and making an error that would affect the program.

5.5.2 Software Implementation

This subsection introduces what software is used to program the ArbotiX-M micro-controller as well as what software is used to handle the EMG signal and make the CrustCrawler Arm move accordingly.

The ArbotiX-M micro-controller is Arduino compatible but not a true Arduino. It support the Arduino variation of C++ which is what all the software for this project is written in. Because the ArbotiX-M micro-controller is not a true Arduino the Arduino IDE can be used with some modifications. Trossen Robotics have already made this ready to download (see [TrossenRobotics 2015]). Furthermore only the Arduino IDE version 1.0.6 can be used as the newer versions of the IDE handles the Arduino core differently which creates complications with the ArbotiX-M micro-controller. Furthermore a Dynamixel mx library is used a serial connection to the UartsBee V4 on the CrustCrawler. All development software is set up and ready to make the CrustCrawler move by writing to the specified.

Package Filtering and Trajectory Planning

When an EMG signal is read by the control box it is sent to the ArbotiX-M microcontroller using two XBee modules. These two modules communicate at a rate of 115200 bits per second (bps) over a standard serial connection (TX and RX). The control box sends a message-package at a frequency of 100 Hz. Among other information a message-package includes 5 analogue inputs - two EMG channel values and 3 accelerometer values. Furthermore it contains a startbyte, two digital outputs which tell if the accelerometer is enabled and if it is set to $\pm$ 1.5 G or $\pm$ 6 G, and a digital input to indicate low battery. In total, a message-package adds up to 24 bytes.
In order to ensure that the CrustCrawler is reading a message-package from the beginning the startbyte is used to determine the beginning of a message-package. The startbyte is constant at $0x7E$ and can therefore be compared with a simple if-statement. A message-package has variation in its values, which also can be $0x7E$. Therefore it is not enough to only check one byte but since the first 3 bytes are constant. The 3 first bytes will be checked, and if the message-package has those values right, the rest of the message-package is assumed to be received successfully. The EMG channels and raw sensor values of the accelerometer are read as 8-bit (range 0 - 255) values through the serial connection with the XBee. Therefore the EMG values are sent as high bytes and low bytes as they are 10-bit values (range from 0 - 1023). Since the raw sensor values are split into 2, they have to be added up in a way that makes it easier to translate to motion parameters. For this, 2 arithmetic operators are used - bitshifting(«) and bitwise logic operator OR(|). This way it is only necessary to do 3 steps which can be written as a single line of code. The first step is to store the high 8 bit value in a 10 bit (or higher) variable. The second step is to move all the bits in the new variable 8 positions to the left. This is done using the bitshift operator. For example $0011 << 2 = 1100$. The last step is to use the | operator to sign the low byte to the variable. The operator compares each position of the bytes with each other and if both bits are 0 the result will be 0 otherwise it will be 1. For example $0100|0101 = 0101$. This can be written as

```
RawEMG = HighByteEMG << 8 | LowByteEMG;
```

After the raw sensor values have been obtained, they can be translated to movement functions for the CrustCrawler. For simplicity in the program a class called CrustCrawler is created and contains functions to move the arm. The object contains methods to move the CrustCrawler in 2 different ways - point-to-point (PTP) and linear movement (LIN). The linear movement is supported if a need for it arises in the future but since it is not used in this project an explanation is left out. The PTP movement is created by making a 3^{rd} order polynomial as trajectory for each joint, thus the position of the robot to a given time can be calculated. It is implemented as follows:

```
float m_gotoTime; //default value is 7 seconds
float a0[3], a1[3], a2[3], a3[3];
for (int i = 0; i < 3; i++)
{
  a0[i] = (float)posStart[i];
  a1[i] = 0.0f;
  a2[i] = 3.0f/(t_f*t_f)*(posEnd[i]-posStart[i]);
  a3[i] = -2.0f/(t_f*t_f*t_f)*(posEnd[i] - posStart[i]);
}
```

The controller described in section 5.4 is then used to carry out the new position:

```
float newPos[3];
unsigned long t_s = millis();
while ((millis() - t_s) <= m_gotoTime)
{
  for (int i = 0; i < 3; i++)
  {
    float t = (float)(millis() - t_s);
    newPos[i] = a0[i] + a2[i]*t*t + a3[i]*t*t*t;
  }
  SetJointPositionCustomController(newPos[0], newPos[1], newPos[2]);
}
```

For each position calculated the custom controller described in section 5.4 can be used. It takes the desired position as input, subtracts the current position and multiplies it with a K_p values to get a torque. This torque is then sent to the servo and the loop repeats until the error becomes constant. Since there will always be a steady-state-error when using a P-controller the percentage of the error has to be under 3%, 3 times before the program moves on.

```
while (errorCount[0] < 2 || errorCount[1] < 2 || errorCount[2] < 2)
{
  for(int i = 0; i < 3; i++)
  {
    float measuredPosition = GetJointPosition(i+1);
    float error = pos[i] - measuredPosition;
    float torque = error * m_Kp[i];
    if (torque > 4600)
      torque = 4600;
    else if (torque < - 4600)
      torque = -4600;
```

```
    SetTorque(i+1, torque);

    if (abs(error/pos)*100 < 3)
      errorCount[i]++;
    oldError[i] = error;
  }
  delay(1);
}
```

Thus the robot is able to move to a predefined position.

When the robot moves in the Cartesian mode, the inverse kinematic model is used from section 5.3.4. They are used to calculate the angles and the SetJointAngle(angle1, angle2, angle3) is used to go to the desired location. Note that the custom controller for this project is only used for PTP movement with predefined points. Whenever the user moves the CrustCrawler arm whether it is in Cartesian mode or joint mode, the internal PID controller of each servo is used.

All the code, including the code to handle the switching between modes, can be found on the CD in Appendix B.

Chapter 6

Integration and Test

This chapter introduces a functionality of the final solution and describes tests that were undertaken to investigate the stability and reliability of the system.

The patient controls the CrustCrawler with a control box attached to his/her head that provides readings from the accelerometer of the box. patient's Masseter and Frontalis muscles are also connected to the control box, which obtains EMG signals from those two muscles. The patient can then control the robot, using these 5 signals.

The robot has 5 different modes the patient can switch between. A mode selection is carried out by head-flicking left or right. Confirming and entering the selected mode is done by flicking head forwards. The navigation inside certain mode is generally carried out in the same way. Left/right head movements are used for switching between options (joints/axes/ ...). When in joint-, or Cartesian mode, contracting the hooked up muscles will then cause either the selected servo to move (Masseter - clockwise, Frontalis - counter-clockwise), or it will make the end-effector to move along the selected axis (Masseter - positive, Frontalis - negative). Moving head forwards then aborts the mode and takes the patient back to mode selection. For the predefined positions mode, the forward head movement executes the selected position and makes the program jump in the pause mode afterwards. For safety purposes, the abortion of pause mode is done by moving the head forwards, followed by moving the head backwards (see section 5.2 for more detailed controls description).

6.1 Interface Calibration

The first test was carried out only with the mode selection program, without the CrustCrawler connected. The purpose was to examine and calibrate the thresholds, and debug the program, before trying to use it to move the CrustCrawler. The movement functions inside the program were replaced by printing messages to a computer screen, so it can be seen, how the flow behaves. This has been done in order to prevent health hazards to people and environmental damage, as well as damage to the robot itself, if a serious bug was encountered. Carrying out this test helps the future problem-solving, namely pinpointing errors, since it proves, that there can be no big error in the program. The setup for the test, was done in the following steps:

- The mode selection program was transferred to a different Arduino board.
- The movement function calls were changed to serial print.
- A 2^{nd} board's RX was connected with the 1^{st} one's TX connection. They were also connected to each others ground.
- The electrodes were placed on the test subjects face.

After few tries, the signal thresholds were set to ideal values, and the program was able to work without any major issues.

6.2 Custom Controller vs. Internal PID

As you can read in section 5.4, an effort was made in delivering own design of a system controller. After finishing the linear design, a test was undertaken to find out, if the controller design is better, than the internal PID control, which the servos contain. The custom controller has undergone certain tests to see, if the design is stable, and applicable. The test included number of applications that the robot can carry out(e.g. PTP movement, linear movement, ...). Using the custom control, the robot was able to carry out PTP movements with success. However, any other application, like the linear movement, was met with failure, since the robot does not move smooth, but more stutter-like and unpredictably. This happens because the designed model is linear and is applied in the real world, where non-linearities exist.

For that reason, the custom controller is used only, when PTP movement is involved. In praxis it means that the custom controller is used only when predefined position is called (i.e. Home, or drinking position). For all other actions, the internal PID controller is utilised.

6.3 Motion Control

After finishing the testing and calibration of the interface, it was possible to replace the serial print commands with movement function calls, and upload the program to the ArbotiX-M board, in order to make a test-run with the CrustCrawler. Utilising just internal PID controller, it was possible to effectively control the CrustCrawler and move it around with the designed interface.

6.4 Final Test

The final testing is, where the overall usability and stability of the system is examined. The whole test is a combination of 3 case tests, which were described earlier in section 4.2. The purpose of this test is to see, if a patient can operate the whole system with ease. The test is considered a success if all 3 cases are carried out without problems. The set of cases was picked carefully, so the patient is forced to utilise all the modes, that were designed (joint movement for pick and place, Cartesian movement for pressing the switch, and combination of Cartesian movement, or horizontal, and predefined positions for drink assistance). If all 3 tests are successful, it proves that:

- The controls are intuitive enough for the patient to effectively utilise them to carry out various tasks.
- All the modes in the interface have been debugged successfully, and are able to operate without complications.

Footage of all the tests can be seen on the CD in Appendix B.

6.4.1 Pick and place

In this test, the patient is to attempt to pick up an object (little box, approx. 5x5x5 cm) placed on location 1, put it to location 2 afterwards. Both locations are made of 9x9 cm squares, and the distance between them is 20 cm. The distance between the center of robot's base, and center of the squares is approx. 32 cm. You can see this setup in figure 6.1 This test is to be carried out by using joint movement only, so the basic functionality can be tested. The test is considered successful, if the patient's succeeds in navigating the robot without any mistake and if he is able to pick up the object and place it in the designated position.

This test was carried out by 3 different people. The results are visible in table 6.1. It states the time how long it took to different patients to carry out the task, and a fail counter, which states how many times the given patient failed, while performing the task. All 3 patients were able to carry out the test on the first try. The time it took to carry out the task varies but Emil had more experience with using the interface. On the other hand, Rasmus and David tried the interface for the first time. This indicates that the CrustCrawler can be even more effective overtime when the patient gets trained.

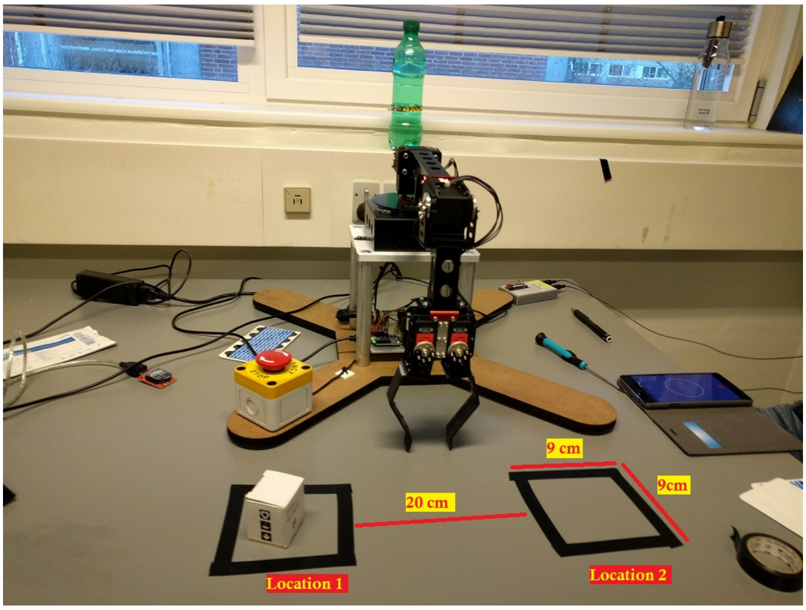


Figure 6.1: Setup used for the Pick and place scenario.

	Fail counter	Time
Rasmus	0	3 min 45 sec
Emil	0	2 min 10 sec
David	0	3 min 1 sec

Table 6.1: Test results for the Pick and Place.

6.4.2 Pressing a light-switch

This test is carried out in slightly different setup. The table with the robot arm is moved to a wall with an actual light-switch, and the task is to navigate the robot, using Cartesian movement, to press the switch, and control the light. This test proves that the patient is able to control the robot with Cartesian movement as well, and it shows that the it is physically possible for the robot to actually press the button. Lastly it proves that the patient can manoeuvre the CrustCrawler precisely enough to hit an object with a width of 5 mm.

6.4.3 Drinking assistance

This last test combines few modes together, which makes the task more complex. This test was carried out in a setup which you can see on figure 6.2. The patient is required to pick up a bottle from point 2, navigating there with use of horizontal movement (most convenient one), and then run the predefined function to bring the bottle swiftly closer to patient's face. If the patient is able to do this without

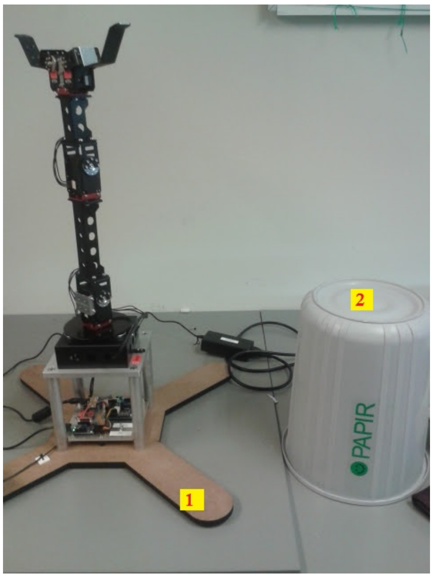


Figure 6.2: Setup for the drink assistance. Point 1 and 2 in the figure are for the orientation purposes for description.

dropping the bottle, with utilising the 2 mentioned modes, it proves that the usability of the solution is convenient enough for a person to operate it effectively.

This chapter presented and examined all the tests that were made, in order to find and troubleshoot any hidden complications, so the final product could be delivered. All tests mentioned here, except the custom controller, were carried out with satisfying results.

Chapter 7

Conclusion

The purpose of this project was to develop a robotic arm controlled by EMG signals, which could be able to replace upper-limb functionality, thus assist people suffering from tetraplegia with their daily tasks. It can be concluded that tetraplegia is a serious and overlooked issue and since paralysis from a spinal cord injury is not a rare issue, it means that to sustain irreversible spine damage is viable as well. The muscles of the face were analysed, in order to find 2 suitable muscles that can provide EMG input to the CrustCrawler.

This led to establishing of 3 cases and a problem formulation. These were created in order to test the system and to be able to set it into real-life scenarios. For the patient to control the CrustCrawler an EMG control box was provided. The box offers readings of 2 EMG channels and an accelerometer.

In order to increase the robustness of the interface, grey zones were implemented. These were chosen over low-pass filters because the grey zones preserve the slopes and peak points of the EMG curve. It was a requirement for the solution to be able to move in different modes, e.g. joint and Cartesian, therefore 2 separate mathematical models were created. 1 for conversion from joint to Cartesian space and another from Cartesian to joint space. This was implemented on a micro-controller along with a control system, that allows the CrustCrawler to be position-controlled.

The 3 case tests consisted of pick and place test, pressing a light-switch and drinking assistance. The pick and place test was carried out on 3 different persons with 2 of them trying it for the first time. This was done in order to see if the interface is intuitive, user-friendly and to see if the patient's performance increases over time. The drink assistance also proved that the CrustCrawler is able to lift a bottle filled with water, which equals to 0.5 kg. All 3 tests were carried out with

satisfying results, and the objectives were met.
From above mentioned information it can be concluded that all the points, raised from the requirements specification, were fulfilled.

Bibliography

Alabama, University of (2002). *Spinal Cord Injury Facts & Statistics*. English. URL: http://www.sci-info-pages.com/facts.html.

Angle, Sanne (2009). "Kampen for et menigsfyldt liv". In: *Sygeplejesken* 17.

BIOPAC (2015). *EMG & FORCE*. BIOPAC Systems, Inc. URL: http://www.biopac.com/?app-advanced-feature=emg-force.

Buisson, Laing & (2014). *Care home or home care?* English. URL: https://www.moneyadviceservice.org.uk/en/articles/care-home-or-home-care#home-care-costs.

Christopher and Dana Reeve (2015). *Paralysis Facts & Figures*. English. Christopher & Dana Reeve foundation. URL: http://www.christopherreeve.org/site/c.mtKZKgMWKwG/b.5184189/k.5587/Paralysis_Facts__Figures.htm.

Craig, John J. (2014). *Introduction to Robotics: Mechanics and Control*. Third. ISBN 10: 1-292-04004-1 ISBN 13: 978-1-292-04004-2. Pearson Education Limited.

Envisage (2014). *The 5 Muscle Groups That Control Your Face & How to Treat Them Well*. URL: http://www.envisagerenew.com/blog/5-muscle-groups-that-control-your-face/.

Genworth (2015). *Compare Long Term Care Costs Across the United States*. English. Genworth. URL: https://www.genworth.com/corporate/about-genworth/industry-expertise/cost-of-care.html.

Hansen, Karl Damkjaer (2015). *CrustCrawler Documentation*. URL: https://github.com/AalborgUniversity-ControlLabs.

Hawking, Stephen (2015). *The Computer*. English. URL: http://www.hawking.org.uk/the-computer.html.

ImageShack (2015). URL: http://imageshack.com/f/132/facetest2aqq4.jpg.

Jeonghee Kim Hangue Park, Joy Bruce Erica Sutton Diane Rowles Deborah Pucci Jaimee Holbrook Julia Minocha Beatrice Nardone Dennis West Anne Laumann Eliot Roth Mike Jones Emir Veledar Maysam Ghovanloo (2013). "The Tongue Enables Computer and Wheelchair Control for People with Spinal Cord Injury". In: *Science Translational Medicine* 5.213. URL: http://stm.sciencemag.org.zorac.aub.aau.dk/content/5/213/213ra166.full.

Johnson, Glen (2010). *Emotional Stages of Recovery*. URL: http://www.tbiguide.com/emotionalstages.html.

Kuebler-Ross, Elisabeth (1969). *On Death and Dying*. Scribner.

Melton, Newsome & (2015). *Spinal cord injury FAQs*. English. Newsome Melton. URL: http://www.brainandspinalcord.org/spinal-cord-injury/complete-vs-incomplete-spinal-injury.html.

Nielsen, Kirsten Moelgaard (2015). *Control design 1*. From presentation on 9th lecture.

NISE, NORMAN S. (2008). *Control Systems Engineering*. John Wiley &Sons, Inc.

Orenstein, David (2012). *People with paralysis control robotic arms using brain-computer interface*. English. Brown university. URL: https://news.brown.edu/articles/2012/05/braingate2.

Prasad, Sai (2015). English. Intel. URL: https://01.org/acat.

Rebuildinglives (2014). *Facts about Spinal Cord Injuries*. English. THE ALLIANCE OF STATE SPINAL INJURY ORGANISATIONS. URL: http://rebuildinglives.com.au/facts/.

Robertson, D. Gordon E. (2013). *Research Methods in Biomechanics, Second Edition*. Human Kinetics Publishers.

Ross, Vicki (2015). *A Dramatic Change of Life*. URL: http://www.christopherreeve.org/site/c.mtKZKgMWKwG/b.5048467/k.4ABB/A_Dramatic_Change_of_Life.htm.

Sanjay Basu MD, PhD (2012). *U.S. versus European healthcare costs: the data*. English. URL: https://epianalysis.wordpress.com/2012/07/18/usversuseurope/.

Satapathy, Sangita (2014). *TDS*. English. URL: http://www.slideshare.net/satapathysangita/tongue-drive-system-sangita-satapathy.

Staff, Healthwise (2014). *Brain & Nervous System Health Center*. WebMD. URL: http://www.webmd.com/brain/electromyogram-emg-and-nerve-conduction-studies?page=1.

Taylor, Tim (2015). *Muscles of the Head and Neck*. InnerBody. URL: http://www.innerbody.com/anatomy/muscular/head-neck.

TrossenRobotics (2015). English. Trossen Robotics. URL: http://www.trossenrobotics.com/p/arbotix-robot-controller.aspx.

WHO (2013). *Spinal cord injury*. English. World Health Organization. URL: http://www.who.int/mediacentre/factsheets/fs384/en/.

Zee, Mark de (2015). Associate Professor at Department of Health Science and Technology, Aalborg University.

Appendix A

Bode Plots

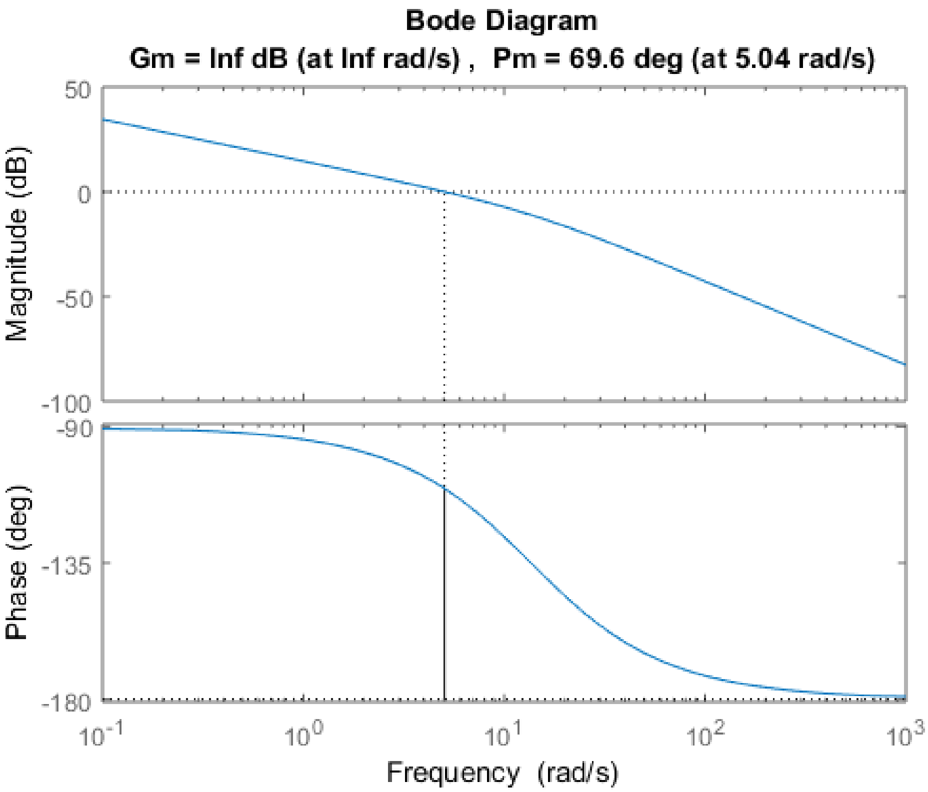


Figure A.1: Bode plot of joint 2

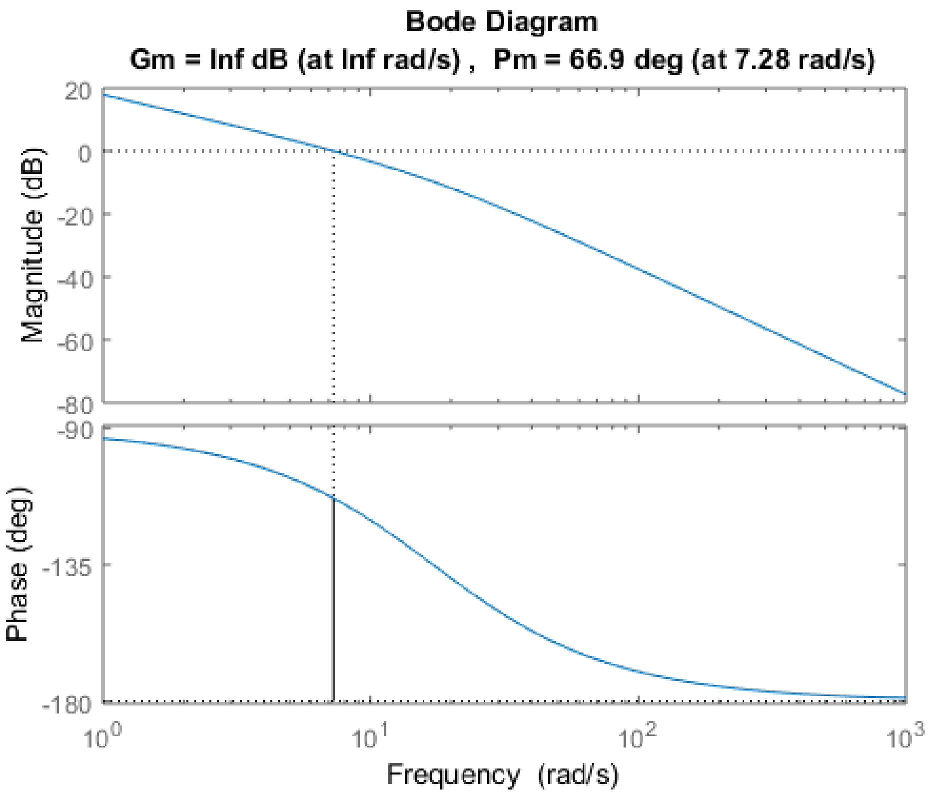


Figure A.2: Bode plot of joint 3

Appendix B

CD

The CD contains the following:

- Source code.
- Test video footage.
- MATLAB files.
- Copy of the report.