

**Computer Vision and Neural Machine Interface**

**For Upper Limb Prostheses**

A graduation project thesis submitted to the

Faculty of Engineering at Cairo University

For the Bachelor degree of

Systems and Biomedical Engineering

***Submitted by:***

Abdlerahman Hamza ***Under supervision of:***

Ghadir Moemen Dr. Meena AbdelMaseeh

Hanna Nabil

Mostafa Elwafa

Acknowledgments

First of all, this work would not have been completed if it were not for God’s will to allow it to be so. And for that we are greatly thankful.

We would have not been able to give all this hard work and dedication if it was not for the continuous support and guidance from ***Dr. Meena AbdelMaseeh***, our project supervisor.

We are deeply grateful for ***ITIDA*** and ***ITAC*** for being our financial sponsers.

We would also like to thank ***Dr. Maged Ghonemia*** for supporting us personal, and providing any needed materials during our work.

It was a great pleasure working within a professional environment, and for that we would like to thank all ***stakeholders of IHUB*** for their dedicated support.

We would like to thank ***Dr. Ahmed Morsy*** for providing the MYO armband (wireless EMG) to our project.

Another thank you from us to the ***Rehabilitation Lab*** staff for providing necessary equipment.

We are grateful for ***Dr. Tamer Basha*** for advising us on practical approaches to the problem.

We are very grateful to all of you, and we wish that we have made you proud.

Abstract

Amputation has always been one of the oldest most crippling health problems in the history of mankind. There were many different approaches to manage the problem and improve amputees’ quality of life. However, all solutions have faced great challenges.

Due to the great variety of amputation types and levels, as well as different personal requirements for each amputee, it is unlikely to have one solution to fit all. Especially, when it comes to upper limb amputation as it is highly dependent on the personal life and daily activities of the amputee, it becomes more difficult to generalize.

In Egypt, the main challenges facing amputees are the high cost of motorized prostheses and incompatibility of the healthcare system with current advances worldwide. Due to the recent advances in the electromechanical systems, motorized prostheses are globally available for most amputees. The challenge, however, remains in designing a suitable control system.

In this work, the proposed system integrates multiple modules and fuses different data streams to provide a suitable control system with an acceptable performance, at an affordable cost for the Egyptian market.

Table of Contents

[Problem definition: 5](#_Toc519200396)

[Industry status: 6](#_Toc519200397)

[State of the art research: 7](#_Toc519200398)

[Problem analysis: 8](#_Toc519200399)

[Objectives and outcomes: 9](#_Toc519200400)

[I. Myoelectric prosthesis: 11](#_Toc519200401)

[II. Computer vision in Grasping Technology: 14](#_Toc519200402)

[Fusion hypothesis: 18](#_Toc519200403)

[Goal: 18](#_Toc519200404)

[System Criteria: 18](#_Toc519200405)

[Approach: 19](#_Toc519200406)

[System main parts: 19](#_Toc519200407)

[2-Computer vision part: 19](#_Toc519200408)

[3- The integration algorithm: 19](#_Toc519200409)

[Flow of information: 20](#_Toc519200410)

[Integration algorithm: 20](#_Toc519200412)

[Limitations: 22](#_Toc519200413)

[Amputee assistance in corrective actions: 22](#_Toc519200414)

[Publicly available datasets: 24](#_Toc519200415)

[EMG signal Datasets from NINAPRO Database: 24](#_Toc519200416)

[Subjects and ethical requirements: 25](#_Toc519200417)

[Acquisition setup: 25](#_Toc519200418)

[Acquisition protocol: 27](#_Toc519200419)

[Signal processing: 29](#_Toc519200420)

[Data Records: 30](#_Toc519200421)

[Amsterdam Library of Object Images (ALOI): 31](#_Toc519200422)

[Technical details: 33](#_Toc519200423)

[Materials and Tools: 34](#_Toc519200424)

[Problems : 34](#_Toc519200425)

[Installation and usage of Open-MYO: 40](#_Toc519200426)

[I.EMG module: 45](#_Toc519200427)

[Methodology: 46](#_Toc519200428)

[Preprocessing: 48](#_Toc519200429)

[Features Extraction: 49](#_Toc519200430)

[Classification: 50](#_Toc519200431)

[II. Computer Vision module: 52](#_Toc519200432)

[III. Result: 55](#_Toc519200433)

[Procedure of EMG Training: 55](#_Toc519200434)

[Procedure of EMG Training: 56](#_Toc519200435)

[Final Result: 56](#_Toc519200436)

[1- Solved problems: 58](#_Toc519200437)

[2- Unsolved problems and suggested solutions: 64](#_Toc519200438)

[3- Future sight: 64](#_Toc519200439)

[Technical skills: 66](#_Toc519200440)

[Soft skills: 66](#_Toc519200441)

[Gantt chart: 67](#_Toc519200442)

Chapter One

INTRODUCTION

Introduction

# Problem definition:

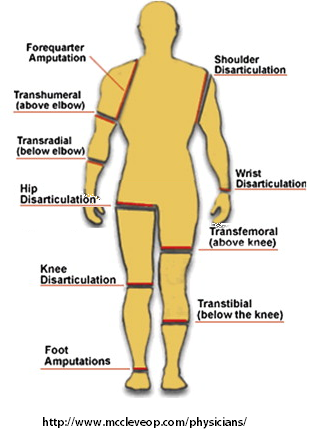


Figure : Types of amputations

Amputation is one of the oldest health problems as it has been prevalent in many societies through the ages. It comes in many different types and forms. It can be mainly categorized into upper body amputation and lower body amputation. These two categories are approached differently. While the challenge with the lower body is maintaining the stability and force control, it is a different story for the upper body amputation. It is a story of dexterous control. How to achieve the required accuracy for such complex movement and perform the action? Especially, when addressing the amputation of the hand, as the most sophisticated movements are required. This shall be the scope of this work: transradial amputation.

A transradial amputee is someone who went through a surgical separation of the radius and ulna of the lower arm (between the elbow and the wrist). In developing countries 1 in 10 is in need of an assistive device. For Egypt, there are above half a million (the largest number in the Middle East) according to National Center for Health Statistics. This number is increasing as most frequent causes of upper limb amputation such as trauma, diabetes and cancer are prevalent.

# Industry status:

There are four main aspects to be considered on attempting to substitute their loss: functionality, usability, appearance and cost. Appearance is the most easily solved problem. However, there is a tradeoff between the remaining three aspects. For example, higher functionality comes at a higher cost and more sophisticated control system, which in turn decreases usability.

Overall, the choice is made individually for each patient based on their circumstances, for example: their specific activities and their economical state. One more element to be considered is the environment of the patient and the healthcare system capabilities, i.e. what is or isn't supported by this system. Mainly, amputees can choose between active and passive prostheses. Passive prostheses aesthetically replace a limb but cannot move. For active prostheses there are two main categories:

1- Body powered prostheses, such as a hook or a prehensor.

2- Motorized prostheses, such as i-limb. They are normally controlled myoelectrically, i.e., using the bioelectrical signals from the residual muscles in the affected limbs.

Motorized prostheses are more cosmetically appealing than traditional body powered prostheses, and generally superior to other types in functionality. Through the times past, many approaches were developed as attempts to focus on recovering functionality. Starting with body powdered prostheses that provide only one degree of freedom, and ending with the most sophisticated motorized prostheses of the modern age. Advanced prosthetic hands can dramatically improve users' quality of life by enabling them to carry out daily living activities.

Due to the advancement of electromechanical technologies, powerful prosthetic hands are now widely available in the market with various configurations suitable to different users. Currently available motorized prostheses have enabled the performance of any desired action with an accuracy mimicking that of a real hand. However, the problem remains in how to obtain the control signal. Over the course of history, there were many different approaches. The leading idea was to use myoelectric control as it should offer intuitive control. Nevertheless, this approach’s weakness lies in the fact that it is highly dependent on myoelectric signals. Recent research reveals that this signal weakens over time due to muscle atrophy after the amputation which makes it unreliable as a long term solution for transradial amputees. The main challenge for transradial amputation is a control signal for such a dexterous movement.

# State of the art research:

Globally, the current most promising research addresses this issue by targeted muscle reinnervation surgery. Unfortunately, this solution does not seem feasible in Egypt. There are two main reasons for the unlikeliness of applying it:

1- The healthcare system in Egypt does not support the requirements of this operation such as the prolonged follow up and rehabilitation period. Also there is a lack of expertise and training.

2- Egypt is a middle income country; this surgery is not common due to its large expenses. Therefore, amputees go for low functionality instead.

As a result, Egyptian amputees are likely to be excluded from the future of motorized prostheses, unless a reasonable control mechanism can be developed.

An altered approach is to use a hybrid control system consisting of different branches contributing with the myoelectric signal to obtaining the required final control signal. Many configurations were introduced to combine information from a myoelectric signal with another signal, such as electroencephalography, electro-oculogram and computer vision.

We opted to design a hybrid solution for an intelligent prosthetic hand that fuses data feed from computer vision with neural machine interface to decode the user’s intent. This solution has the capability to overcome the previously mentioned issues and it would be feasible in Egypt as well as more affordable.

# Problem analysis:

 For most amputees, full function of the hand is unlikely to be restored. Luckily, with good problem analysis, it is possible to divide the elephant task and focus on restoring the most commonly needed functionality. It was found that restoring the ability to grasp and move objects can be of great benefit in increasing an amputee’s quality of life. Recently, researchers started to direct their attention to intelligent prostheses. Inspired by the concept of muscle memory of the natural hand, the intelligent prosthetic hand has the ability to decide which movements to carry out without the requirement of complex EMG analysis. This novel method of control gives the patient the ability to grasp different objects including objects he/she has never seen before, which is the most challenging control task. And one of its great advantages is that it requires less training from the amputee as it is mainly dependent on the artificial intelligence system to decode the user’s intent and carry out the movement, as opposed to getting all the information needed from bio-signals controlled by the amputee.

Figure : Grasp types

# Objectives and outcomes:

1- The goal of this work is to enable transradial amputees to use a simple affordable, yet efficient, computer vision system to grasp and move common objects with motorized prostheses. We provide a prototype for a hybrid solution that integrates myoelectric control and computer vision to get an adequate result.

2- The system will open the opportunity for governmental insurance to be a good choice for amputees as well as benefit the Egyptian healthcare organizations.

3- Improve the Egyptian investment: the project can result in business environment with stronger awareness of the challenges in the medical field and concrete ways to overcome them.

4- Fill the gap between the research and industrial market by making such system realizable.

5- Improve the quality of life for transradial amputees in Egypt.

Chapter Two

LITARETURE REVIEW

Literature Review

# I. Myoelectric prosthesis:

A myoelectric prosthesis is one which is controlled by the electrical activity of a muscle, i e. by a myoelectric signal (MES). Most myoelectric prostheses are powered by electricity from a battery, in which case the myoelectric signal controls the flow of energy from the battery to an electric motor.

Electrically powered prostheses with myoelectric control have several advantages over other types of prostheses: the user is freed of straps and harnesses required of body powered and mechanical switch control; the MES is noninvasively detected on the surface of the skin; the controller can be adapted to proportional control with relative ease; and muscle activity required to provide control signals is relatively small and can resemble the effort required of an intact limb.

Despite the efforts of the media and others to convince us that a myoelectric prosthesis is something new from the space age, I must tell you that the first myoelectric prosthesis we know of was created in the period 1944-1948 by Reinhold Reiter, a physics student at Munich University.

Because the transistor had not been invented, Reiter was forced to use vacuum tubes in the electronic system, and it was not feasible to make the system portable Instead, this prosthesis was designed for use at a factory bench, powered from the nearest outlet.

Even at this early date Reiter recognized the need to obtain maximum information from the myoelectric signal: his system controlled both opening and closing of an electric hand from a single muscle.

Reiter's system did not gain clinical or commercial acceptance. More regrettably, his work was not published, although it was described briefly in a report on a 1948 Hannover trade fair, printed in an obscure German medical newspaper. It was not discovered by the researchers who succeeded him until 1969, by which time his ideas had been reinvented.

In the years beginning about 1957, researchers in many countries invented myoelectric control. Groups led by Bottomley in England; Herberts in Sweden; Kato in Japan; Kobrinski in Moscow; Reswick, Lyman and Childress in the USA, to extend the initial work in each country was independent: they each invented Reiter's work.

In any event, they all were impaired in this work by two factors First, what they were attempting to accomplish was well beyond the capability of available technology. Second, they did not understand adequately the needs and desires of the amputees on whose behalf they were working.

In 1965 there were collaboration between The University of North Dakota and the Ontario Crippled Children's Center to fit the first all Canadian myoelectric prosthesis. The control system was contained in three boxes worn in a belt around the waist. The prosthesis featured an electric wrist although this was replaced quickly by an electric hook, figure 3.

Shortly after this, they designed necessary electronic circuitry to permit operation of this electric hook in a normally closed, voluntary opening mode what is now referred to as a cookie crusher mode. But this work did not result in hardware which could be obtained and used readily by the average prosthetics.

After a long conversation with some amputees, they realized that a hand is not a pair of pliers, nor an arm a device to place the pliers in space. Comfort and even cosmetic were accepted as critical requirements, figure 4.

Understanding of the process by which the myoelectric signal is generated improved significantly in this period, providing important new insight into how to use this signal for control.



Figure : Electric hook.

Figure : Cosmetic hook.

By the mid 70's Otto Bock was established firmly as the predominant supplier of myoelectric controls and electrically powered components for prostheses.

Also in the early 1970, pattern recognition based approach to myoelectric control was developed. It used amplitude-based features and a simple statistical classifier to achieve reasonable accuracy, but used many myoelectric channels with cumbersome instrumentation, and required a large computing facility and lots of processing time.

In the 1980s, the pattern recognition approach was refined somewhat by extracting more information from fewer (two to four) myoelectric channels. This allowed greater accuracy, but the computing facilities of the day were incapable of achieving this task in real time.

In the early 1990s, the accuracy of the pattern recognition approach was improved again with the use of artificial neural network classifiers.

In recent years, there has been a steep rise in the quality of prostheses for patients with upper limb amputations which allow for an increase in the degrees of freedom of hand designs and a larger number of available grip patterns with little added complexity for the wearer.

In 2003, Kevin Englehart publishes a study [2] which represents an ongoing investigation of dexterous and natural control of upper extremity prostheses using MES. The scheme described within uses pattern recognition to process four channels of MES, with the task of discriminating multiple classes of limb movement. The method does not require segmentation of the MES data, allowing a continuous stream of class decisions to be delivered to a prosthetic device.

Another recent improvement in prosthetic hand design instead employs electroneurographic (ENG) signals, requiring an interface directly with the peripheral nervous system (PNS) or the central nervous system to control a prosthetic hand. While ENG methods are more invasive than using surface EMG for control, an interface with the PNS has the potential to provide more natural control.

Despite the recent progress in design and control strategies, however, prosthetic hands are still far more limited than the actual human hand.

# II. Computer vision in Grasping Technology:

In the case of using computer vision it was shown that object shapes can be quantized such that appropriate grasp types and sizes can be determined.

Došen et al [11] demonstrated a dexterous hand with an integrated vision based control system. The user controlled the prosthesis hand and the activation of the camera with myoelectric signals. A simple object detection method was used, in conjunction with distance information, estimated via ultrasound. This structure allowed them to approximate the size of the object of interest. The calculated size was then introduced to a rule-based reasoning algorithm to select the appropriate grasp.

Marković et al [12] demonstrated a semi-autonomous control mechanism in which stereo-vision provided depth information. In addition, their solution offered artificial proprioceptive feedback, via visual feedback to the user, about the grip aperture size by using augmented reality.

They incorporated sophisticated algorithms for image segmentation, 3-dimensional point cloud generation and geometrical model fitting.

With such improvements, the process of identifying the object size and the appropriate grasp became significantly faster.

Marković further exploited a data fusion technique to control a prosthetic hand. A plethora of modalities, namely, myoelectric recording, computer vision, inertial measurements and embedded prosthesis sensors (position and force) were utilized to provide realtime simultaneous, proportional and semi-autonomous control. The shape, the size and the orientation of objects were estimated with RGB-D imaging and integrated with prosthesis orientation and user behavior via inertial sensing. This setting was integrated into a prosthetic wrist, but only palmar and lateral grasps were considered.

Saxena *et al* [13] provide the capability of grasping novel (unseen) objects for robotic hands by utilizing a stereo camera. Without building a 3-dimensional model, they estimated the 3-dimensional location of the best grasp by triangulation. The grasp location estimator algorithm was trained on synthetic images in a supervised learning regime.

Kootstra et al [14] developed an early cognitive vision architecture for grasping unknown objects. Without any segmentation or preprocessing, they were able to generate two- and three-finger grasps based on contours and surface structure provided by stereo cameras.

Lenz et al [15] introduced RGB-D images to a two-step cascade deep learning system. Given the image of an object to grasp, firstly a small deep network determined the suitable grasping points for the object; based on its position, size and orientation.

Then, a second network was trained to pick the best candidate among the grasping spots that were identified by the first network.

Kopicki et al [16] provided a one-shot learning mechanism for recognizing the most appropriate grasp for novel objects. They generated thousands of grasp candidates for images taken by a depth camera and optimized the combination of two learned model types: a contact model and a hand configuration model.

Nazarpour [6] proposed Computer vision-based assistive technology solutions which can revolutionize the quality of care for people with sensorimotor disorders. The goal of this work was to enable trans-radial amputees to use an efficient computer vision system to grasp and move common household objects with a two-channel myoelectric prosthetic hand.

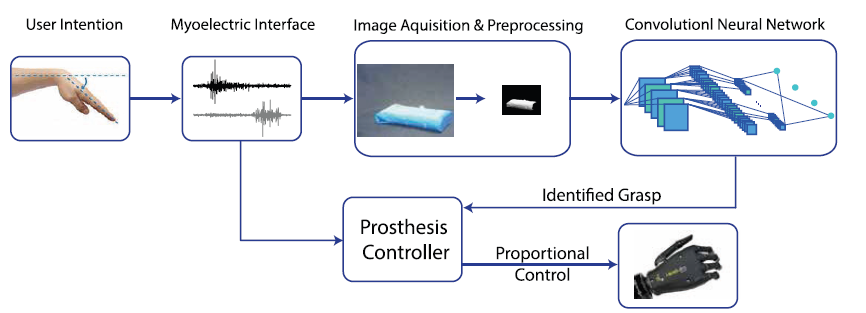


Figure : Nazarpour overall control structure. This figure is cited from [6].

Nazarpour developed a deep learning-based artificial vision system to augment the grasp functionality of a commercial prosthesis. His main conceptual novelty is that we classify objects with regards to the grasp pattern without explicitly identifying them or measuring their dimensions. The overall control structure is illustrated in figure 5.

Chapter Three

PROPOSED SOLUTION AND SYSTEM DESIGN

Proposed solution and system design

# Fusion hypothesis:

We propose that a hybrid solution fusing data streams from computer vision with neural machine interface to decode the user’s intent can produce a suitable affordable solution for Egyptian transradial amputees.

# Goal:

To create an online system with human in loop and use empirical evaluation to attain an adequate result.

# System Criteria:

The proposed system should offer intuitive control, have an acceptable accuracy and it should be a long term solution.

# Approach:

Our approach is to use artificial intelligence to decode user’s intent. Recently, there have been great improvements in the field of artificial intelligence. And despite its disadvantages of undeterministic results, high requirements and being expensive computationally, we adapted this approach as it better fits the problem addressed. Due to the biovariability between subjects, and the requirement of the system to scale this problem has too many factors to have a hard coded solution.

# System main parts:

The system constitutes of three main parts, each assigned with a specific task:

1-Neural machine interface (NMI):

Using electrical signals from the muscles acquired from the surface (surface EMG) as a control signal generated by the amputee and interpreted by machine learning.

It is used to provide control to the amputee over the system, and give the ability of corrective actions.

## 2-Computer vision part:

A fully automated part to process images of chosen objects and determine the suitable grasp type.

## 3- The integration algorithm:

An algorithm to control the interactions and flow of information between the human controlled part (NMI) and the fully automated part (computer vision part). Thus, giving the amputee assisted action control as described below.

## Flow of information:

# 

Figure : System block diagram

# Integration algorithm:

The integration algorithm uses four classes of EMG signals as command orders by the amputee as follows:

***Class 0:*** Reactivate system (restart).

***Class 1:*** Confirm action.

***Class 2:*** Cancel action.

***Class 3:*** Put system in rest mode.

These four commands are used to control a sequential list of actions depending on the current state of the system as follows:

***State 0:*** System on rest mode, listening for an EMG activation signal. When the signal is received, it moves to state 1.

***State 1:*** Processing object image using the computer vision part to determine the suitable grasp type. Then preshaping the prosthetic hand in the form of the grasp according the determined type. If the system receives a confirmation signal it moves forward to the next state.

***State 2:*** Preforming the grasp and keeping it firm until next signal.

***State 3:*** Releasing the object, putting the prosthetic hand back to rest position and system automatically goes back to rest state, until another EMG activation signal is received.

During the sequential actions described above, if a cancellation signal is received at any time, the system repeats the previous state.

EMG classes 0 and 3 are independent of current system state, as class 0 is used to restart the system from any state into state 1 of determining the grasp type. And class 3 puts the system on rest mode, which is useful in case the amputee accidentally activated the system or changed his mind and does not wish to proceed with the action.

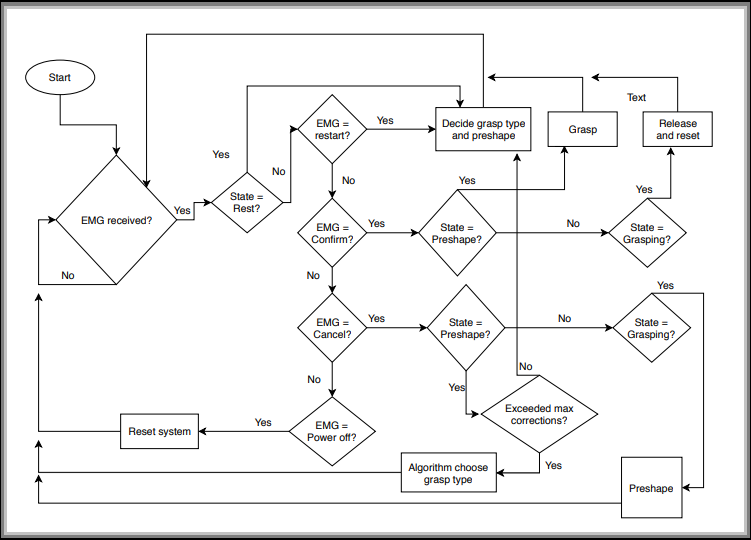


Figure : Integration algorithm flowchart

# Limitations:

Due to the requirements of the system to be affordable and simple in design, there were some tradeoffs in the design. This led to the following limitations:

1. The current system does not offer proportional control for the amputee to control the size of the grip or the force applied on the object.
2. The system does not offer feedback control.

These limitations will be further discussed in chapter 6 as part of the future sight for this project.

# Amputee assistance in corrective actions:

Since the system has high uncertainty due to dependency on artificial intelligence, it was suitable to add an assisting way for the amputee in corrective action. In a special scenario, if an error presists and the grasp type is not suitable for the desired object despite repeated calls to the computer vision module, the integration algorithm handles the case. It removes the mistakenly predicted grasp type and chooses at random from remaining types. It repeats this process until an action is confirmed by the amputee marking a correct grasp type. Then, the algorithm returns to its original state of communicating with the computer vision model.

Chapter Four

PUBLICLY AVAILABLE DATASETS

Publicly Available Datasets

## EMG signal Datasets from NINAPRO Database:

Different datasets from NINAPRO project were used, which is an ongoing work that aims to aid research on advanced hand myoelectric prosthetics with publicly available datasets.

The mentioned databases are obtained by jointly recording multi-modal data, including e.g. surface electromyography (sEMG) signals, hand kinematics, hand dyanamics while the subjects perform a predefined set of up to 53 movements.

Dataset 1 includes data from 27 intact subjects.

Dataset 2 includes data from 40 intact subjects.

Dataset 3 includes data from 11 transradial amputees (with amputation levels as represented in the figure at the end of the page).

Dataset 4 includes 10 intact subjects recorded with "Cometa" electrodes.

Dataset 5 includes 10 intact subjects recorded with two Thalmic Myo armbands, putting them on the same forearm simultaneously.

Dataset 6 contains repeatability data from 10 intact subjects repeating data acquisitions 2 times per day for 5 days.

Dataset 7 contains data from 20 intact subjects and 2 transradial amputees.

## Subjects and ethical requirements:

Before the data acquisition began, each subject was given a thorough written and oral explanation of the experiment itself, including the associated risks; the subject would then sign an informed consent form.

## Acquisition setup:

The acquisition setup included several sensors, designed to record hand kinematics, dynamics and the corresponding muscular activity. The sensors were connected to a laptop responsible for data acquisition.

Hand kinematics was measured using a 22-sensor CyberGlove II dataglove (CyberGlove Systems LLC, The CyberGlove is a motion capture data glove, instrumented with joint-angle measurements. It uses proprietary resistive bend-sensing technology to accurately transform hand and finger motions into real-time digital joint-angle data. The device returns 22 8-bit values proportional to these angles for an average resolution of less than one degree depending on the size of the subject’s hand. In addition to the CyberGlove, a standard commercially available 2-axis IS40 inclinometer was fixed to the subject’s wrist to measure the wrist orientation. This inclinometer has a range of 120° and a resolution of less than 0.15°.

Hand dynamics was measured using the Finger-Force Linear Sensor (FFLS) employing strain gauge sensors to detect flexion and extension forces of all fingers, plus abduction and adduction of the thumb. This sensor is characterized by high signal repeatability, minimal drift over time, almost perfect linearity and virtually no hysteresis (both parameters have a maximum deviation of 0.3%).

Muscular activity was measured using either OttoBock or Delsys double-differential sEMG electrodes. In the first configuration ten MyoBock 13E200-50 electrodes were used, providing an amplified, bandpass-filtered and Root-Mean-Square (RMS) rectified version of the raw sEMG signal. The electrodes’ amplification gain was set at about 14,000. These electrodes are already widely used in prosthetics; they employ frequency shielding and filtering in order to avoid low and high frequency interferences emitted, for example by 50–60 Hz power sources, mobile phones or security systems. These electrodes were fixed on the forearm using an elastic armband. In the second configuration we used 12 Trigno Wireless electrodes (Delsys, Inc, www.delsys.com), each one equipped with a self-contained rechargeable battery and with an operational range of 40 m (the setup also includes a wireless receiving base station). sEMG signals are sampled at a rate of 2 kHz with a baseline noise of less than 750 nV RMS. These electrodes also integrate a 3-axes accelerometer sampled at 148 Hz and were fixed on the forearm using their standard adhesive bands. A hypoallergenic elastic latex–free band was placed around the electrodes to keep them fixed during the acquisition.

Particular care was taken in the placement of the electrodes on the forearm, since this is usually regarded as a crucial step for data usability. We decided to combine two methods which are common in the field, that is, a dense sampling approach and a precise anatomical positioning strategy. The electrodes are positioned as: eight electrodes are equally spaced around the forearm at the height of the radio-humeral joint; two electrodes are placed on the main activity spots of the flexor digitorum superficialis and of the extensor digitorum superficialis; in the second configuration only, two electrodes were also placed on the main activity spots of the biceps brachii and of the triceps brachii. The main activity spots were identified by palpation. This positioning of the electrodes also gives the opportunity to improve inter-subject classification results by applying linear and non-linear spatial registration algorithms.

Data from all sensors were transmitted to the laptop used for data acquisition in different ways. Data from the CyberGlove were transmitted over a Bluetooth-tunneled serial port at slightly less than 25 Hz; data from the inclinometer, the FFLS and the Otto-Bock sEMG electrodes were acquired at 100 Hz using a National Instruments data acquisition card (NI-DAQ PCMCIA 6024E, 12-bit resolution); the Delsys base station received the sEMG and accelerometer streams via an ad-hoc wireless network and relayed the data via a standard USB connection to the laptop. Each data sample provided by each sensor was associated to an accurate timestamp using Windows performance counters.

## Acquisition protocol:

Preceding the experiment, each subject is requested to give informed consent and to answer questions including age, gender, height, weight and laterality. In the case of amputees, we also note the date, type and reason of the amputation, remaining forearm percentage, information about the use of prostheses (cosmetic, kinematic, myoelectric), type and degree of phantom limb sensation and DASH (Disability of the Arm, Shoulder and Hand) score. The remaining forearm percentage is computed as the ratio between the length of the amputated forearm and the length of the contralateral forearm from the elbow to the wrist, rounded to the tens. Subsequently, subjects were made to sit at a desk on an office chair, adjusted to match the maximum comfort, and comfortably resting their arms on the desktop. A laptop in front of the subject provided visual stimuli for each task, while also recording data from the measurement devices.

The experiment is divided into one training part and three exercises addressing different types of movements, interrupted by rest time in order to avoid muscular fatigue. The training phase consisted of a condensed mix of the exercises, in order for the subjects to become familiar with the experiment.

The details of the acquisition procedure depend on the kinematics or dynamics acquisition setup. During the exercises performed using the Cyberglove II, the intact subjects were asked to mimic movies of movement shown on the screen of the laptop with their right hand, while amputated subjects were asked to mimic the movements with the missing limb as naturally as possible.

Since the main aim of kinematics data is to permit movement classification, all the subjects were asked to concentrate on mimicking the movements rather than on exerting high forces. The set of movements was selected from the hand taxonomy, robotics, and rehabilitation literature, with the aim of covering the majority of the hand movements encountered in activities of daily living (ADL). Each movement repetition lasted 5 s, and it was alternated with a rest posture lasting 3 s. The sequence of movements was not randomized in order to encourage repetitive, almost unconscious movements.

During the exercises performed using the FFLS, subjects were instructed to repeat nine force patterns by pressing with one or more right hand digits on the device. An initial calibration phase was performed to establish the rest and maximal voluntary contraction (MVC) force levels for all fingers, and training was performed before each force pattern. (The MVC for the amputated limb was estimated according to the sensation of the subject.) The force levels requested for each finger were represented as colored bars on the screen. During the exercise, the stimulus increased up to 80% of the maximal voluntary contraction force established during calibration, and then it decreased to 0% following a squared-sinusoidal pattern.

Also in this case, intact subjects were asked to execute the experiment with their right hand, while amputated subjects were asked to think to repeat the movements as naturally as possible with the missing limb. It is important to remark that amputees cannot, in general, produce any reliable ground truth due to the inability to operate any sensor with the missing limb. In related literature, this fundamentally unsolvable problem has been circumvented either by

1. Instructing the subjects to execute a task bilaterally while recording the ground truth from the intact limb.
2. Or by asking them to follow a visual stimulus.

There is no consensus on the best procedure, so each subject was left free to choose after a short training phase, which resulted in only two subjects undergoing bilateral execution. As a result of this, for the rest of the amputees, the database contains only the stimulus as ground truth. Analyses with the stimulus as ground truth have, anyway, already been successfully performed

## Signal processing:

Several signal processing steps were performed, These steps included synchronization, relabelling and (for the Delsys electrodes) filtering. The raw data are available upon request.

* Synchronization:

High-resolution timestamps were used to synchronize the data streams. Specifically, all streams were super-sampled to the highest sampling frequency (2 kHz or 100 Hz, depending on the used sEMG electrodes) using linear interpolation (real-valued streams) or nearest-neighbor interpolation (discrete streams).

* Relabeling:

The movements performed by the subjects may not perfectly match with the stimuli proposed by our software due to human reaction times and experimental conditions. The resulting erroneous movement labels have been corrected by applying a generalized likelihood ratio algorithm offline, which realigns the movement boundaries by maximizing the likelihood of a rest-movement-rest sequence. Both the original labels and the new labels are included in the files.

* Filtering:

The Delsys electrodes are not shielded against power line interferences, which can affect the recoded signal in particular cases. Therefore, prior to synchronization, the Delsys sEMG signals were cleaned from 50 Hz (and harmonics) power-line interference using a Hampel filter.

## 

## Data Records:

For each subject and exercise, the database contains one file in Matlab format with synchronized variables. The variables included in the files are:

* subject: subject number;
* exercise: exercise number;
* emg: sEMG signal of the electrodes; columns 1–8 include the signal from the electrodes equally spaced around the forearm; columns 9 and 10 include the signal from the electrodes located on the main activity spots of the muscle Flexor Digitorum Superficialis and of the muscle Extensor Digitorum Superficialis; when available, columns 11 and 12 include the signal from the main activity spots of the muscle Biceps Brachii and of the muscle Triceps Brachii;
* acc (36 columns): (x,y,z)-axis acceleration values of the 12 electrodes;
* glove (22 columns): uncalibrated signal from the 22 sensors of the Cyberglove. The raw data are declared to be proportional to the angles of the joints in the CyberGlove manual.
* inclin (2 columns): inclinometer (roll,pitch) values;
* stimulus (1 column): the original label of the movement repeated by the subject;
* restimulus (1 column): the a-posteriori refined label of the movement;
* repetition (1 column): stimulus repetition index;
* rerepetition (1 column): restimulus repetition index;
* force (6 columns): force values;
* forcecal (2×6 values): maximal force values (minimal and maximal force values for each sensor).

## Amsterdam Library of Object Images (ALOI):

ALOI is a color image collection of one-thousand small objects, recorded for scientific purposes. In order to capture the sensory variation in object recordings, we systematically varied viewing angle, illumination angle, and illumination color for each object, and additionally captured wide-baseline stereo images. We recorded over a hundred images of each object, yielding a total of 110,250 images for the collection. See Technical Details for a description of the acquisition setup.

Details have been published in: J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, The Amsterdam library of object images, Int. J. Comput. Vision, 61(1), 103-112, January, 2005.

Each object was recorded with only one out of five lights turned on, yielding five different illumination angles (conditions l1-l5). By switching the camera, and turning the stage towards that camera, the illumination bow is virtually turned by 15 (camera c2) and 30 degrees (camera c3), respectively. Hence, the aspect of the objects viewed by each camera is identical, but light direction has shifted by 15 and 30 degrees in azimuth. In total, this results in 15 different illumination angles.

Furthermore, combinations of lights were used to illuminate the object. Turning on two lights at the sides of the object yielded an oblique illumination from right (condition l6) and left (condition l7). Turning on all lights (condition l8) yields a sort of hemispherical illumination, although restricted to a more narrow illumination sector than true hemisphere. In this way, a total of 24 different illumination conditions were generated, conditions c[1..3]l[1..8].

Illumination Color is varied in 12 configurations.

Each object was recorded in frontal view, with all five lamps turned on. Illumination color temperature is changed from 2175K to 3075K. Cameras were white balanced at 3075K, resulting in objects illuminated under a reddish to white illumination color, conditions i110, i120, ..., i250.

Object Viewpoint is varied from 72 directions.

The frontal camera was used to record 72 aspects of the objects by rotating the object in the plane at 5 degree resolution, conditions r0..r355. This collection is similar to the COIL collection.

By turning the object to view the second camera, a center image (configuration c), a right image (configuration r) and a left image (configuration l) could be made. The combination of left-center and center-right images yields two pairs of 15 degree baseline stereo, whereas the combination of the left-right pair yields a 30 degree baseline stereo pair.

## Technical details:

* Light Controller: Dimmer Osram HT 1-10 DIM,Transformer Osram HT 150/230/12L
* Cameras: Sony DXC390P 3CCD, settings: gain 6dB
* Lenses: Computar, 12.5-75mm, 1:1.2, settings: f=5.6mm, zoom=48mm (objects 1-750) ,zoom=15mm (objects 751-1000).
* Rotation Stage: Parker Hannifin Corporation 20505RT,(The rotation table is set up for 800 steps per revolution,Maximum speed is 5.22 rpm.)
* Rotation Stage Controller: Parker Hannifin Corporation, power supply P25L,stepper driver L50, stepper motor SY563T.
* Frame Grabber: Matrox Electronic Systems Ltd. CORONA\_II PCI framegrabber.

Chapter Five

SYSTEM IMPLENTATION

AND

USED MATERIALS

System implementation and used materials

# I.EMG module:

The surface MES is an effective and important system input for the control of powered prostheses. This control approach, referred to as myoelectric control, has found widespread use for individuals with amputations or congenitally deficient upper limbs.

Clinical evaluations of myoelectrically controlled prostheses indicate that the three major factors that determine the acceptance rates by the users are: the type of prosthesis, the degree of user training, and the control strategy. It is the third factor that we consider here.

A myoelectric control system is described that offers exceptional performance with regard to three important aspects of controllability: the accuracy of movement selection, the intuitiveness of actuating control, and the response time of the control system.

Accuracy is essential to faithful realization of a user’s intent. It must be as high as possible, although it is difficult to define the threshold of acceptability, as no definitive clinical trials have addressed this issue.

An intuitive interface to the control system relieves the mental burden of the user. In this regard, a control system should be capable of “learning” the muscle activation patterns.

The response time of a control system should not introduce a delay that is perceivable by the user.

The criteria we used suggested a pattern-recognition-based approach to myoelectric control. Two things are needed for this to be possible:

1. More information must be extracted from the MES about the active muscle state. The manner in which one might extract more information from the MES could involve one or both of the following approaches.

• Use multiple channels of MES, providing localized information at a number of muscles sites.

• Develop a feature set that extracts as much information as possible from the MES that serves to discriminate different classes of movement.

1. A classifier, capable of exploiting this information, must be constructed. The role of the classifier is to assimilate and exploit the information it receives, and decide from which class the information originates.

## Methodology:

The construction of a continuous classifier and the acquisition of data used to evaluate it are described herein. The control problem can be defined for any set of motions. It was decided to investigate a four-class problem involving hand and wrist control, as those with below-elbow limb deficiencies represent a large proportion of prosthetic users. Twelve channels of myoelectric data were acquired using electrodes.

Pattern recognition was performed on analysis windows that may be up to 256 ms in duration (a longer record may lead to more delay). For each analysis window, a feature set was computed, and these features provided to a pattern classifier. Figure 8 illustrates the MES windowing strategy in the continuous classifier.

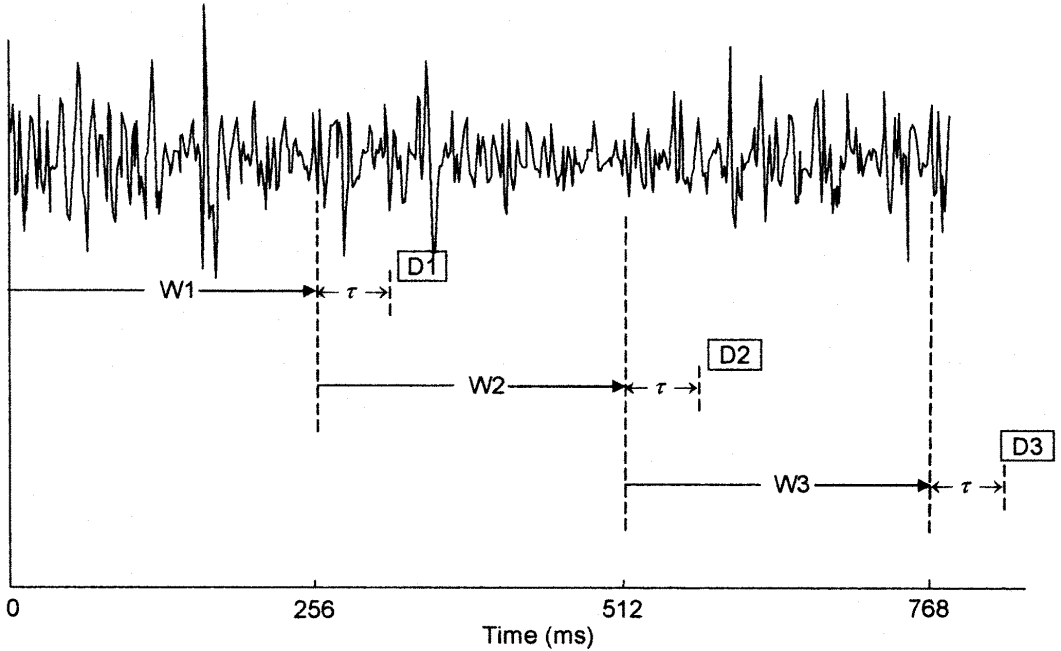


Figure : Windowing of MES data in the continuous classifier, just single channel is shown. Successive analysis windows (W1, W2, and W3) are adjacent and disjoint. For each analysis window, a classification decision. This figure is cited from [2]

The continuous classifier acts upon a sliding window of data, producing a class decision (an estimate of the intended motion) from each window. The simplest approach is to use adjacent, disjoint analysis windows of the MES. This is equivalent to incrementing the window position by an amount equal to its size, as illustrated in figure 8.

In this scheme, each analysis window is equal to 256 ms. Therefore, decisions are made at 256-ms intervals. It is clear from figure 1 that processing (feature extraction and classification) occurs in only a portion of the time spent acquiring data, implying that a processing system will be underutilized. Consider a scheme that fully utilizes the computing capacity of a given system: as soon as a decision is generated, begin processing the data of the most recent samples. This is analogous to incrementing the -sample analysis window as show in figure 9.

This produces a decision stream that is as dense as possible, and makes the decisions closed to each other. This will lead to smoother motion

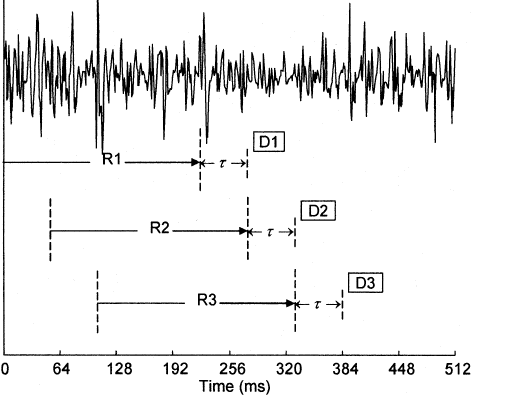


Figure : Windowing schema that produce decision stream that is as dense as possible. This figure is cited from [2]

### Preprocessing:

As surface electrodes are used, there are two main issues produced which disturb the original MES: Crosstalk and electrode movements.

Muscle crosstalk occurs when the MES from one muscle interferes with that of another limiting reliability of the signal of the muscle being tested.

Muscle crosstalk and electrode movements produce noise which is considered low frequency components.

So, a high-pass Butterworth filter with corner frequency at 20 Hz is utilized for preprocessing. This configuration is expected to eliminate artifacts due to electrode movement and reduce the crosstalk by attenuating contributions from muscles distant from the electrode.

Empirically, also we found that signal normalization can help in some scenarios. Signal amplitude of each channel is divided by its corresponding mean.

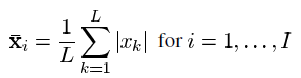
### Features Extraction:

Feature Extraction plays a key role in the overall classification process because extracted features from raw data represent characteristics of the signal and they affect the performance of a classifier. In the EMG motion classification task, feature extraction is also essential for improving the performance of a classifier.

According to [3], Time-domain feature set showed the best classification accuracy compared to the feature sets using Discrete Wavelet Transform, Wavelet Packet Transform and Empirical Mode Decomposition.

Time domain feature set was created directly from raw EMG data. That means that any transformation was not required. In this study, four features are selected:

1. Mean Absolute Value: An estimate of the mean absolute value of the signal x in segment i which is L samples in length is given by



Where *Xk* is the *k*-th sample in segment *i* and *I* is the total number of segments in the record.

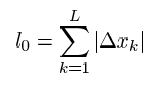
1. Zero Crossings: A simple frequency measure can be obtained by counting the number of times the waveform crosses zero.

D:\4th\GP\finallllllll\1.png

1. Slope Sign Changes: A feature that may provide another measure of frequency content is the number of times the slope changes sign. Given three consecutive samples *Xk-1, Xk and Xk+1*, the slope sign change is incremented if



1. Waveform Length: A feature that provides information on the waveform complexity in each segment is the waveform length. This is simply the cumulative length of the waveform over the segment, defined as



.

The resultant values indicate a measure of waveform amplitude, frequency, and duration all within a single parameter.

Feature set is computed on each of twelve channels, and then concatenated to form a 48-dimensional feature vector. This feature vector was then provided to the classifier.

### Classification:

Two classification algorithms were used in this study, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM).

KNN is more general version of the nearest neighbor technique bases the classification of an unknown sample on the votes of k of its nearest neighbor rather than on only it’s on single nearest neighbor, figure 10.

If the costs of error are equal for each class, the estimated class of an unknown sample is chosen to be the class that is most commonly represented in the collection of its K nearest neighbor’s sample.

Among the various methods of supervised statistical pattern recognition, the Nearest Neighbor is the most traditional and powerful one; it does not consider a priori assumptions about the distributions from which the training examples are Arranged. Also KNN has no model other than storing the entire dataset, so there is no learning required.

A new sample is classified by calculating the Euclidean distance to the nearest training case. It is common to select k small and odd. If K is very low, say k=1, the model will be sensitive to the noise and this can lead to overfitting. If the k value is very large, the model may underfit the data.

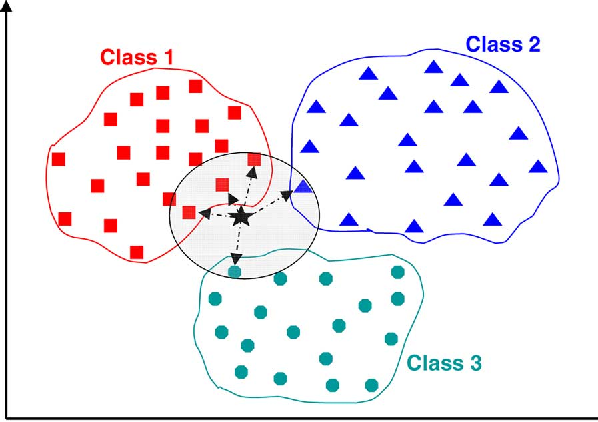


Figure : KNN starts at te test point, it is labeled by the majority vote of the samples around.

In this case k=5, and the test point will belong to class 1.

Regarding the SVM, it is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

The regularization parameter (often termed as C parameter in python’s sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.

For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

In this study, linear support vector classification (LinearSVC) is used with kernel ‘linear’ and implemented in terms of ‘liblinear’. Also one versus reset is used as the multi class strategy with C parameter equal to one.

## II. **Computer Vision module:**

We set out to translate the advances in deep learning in the robotics and computer vision research for control of hand prostheses. This solution can identify the appropriate grasp type for objects according to a learned abstract representation of the object rather than the explicitly-measured object dimensions. This key concept is illustrated in figure 11.

In this way, objects are not classified based on the object category or identity, but based on the suitable grasp pattern. We predict that a deep network trained for grasp recognition can extract high level and grasp related features from objects and discard other unnecessary details. This approach is therefore conceptually different from object recognition in which object details matter.



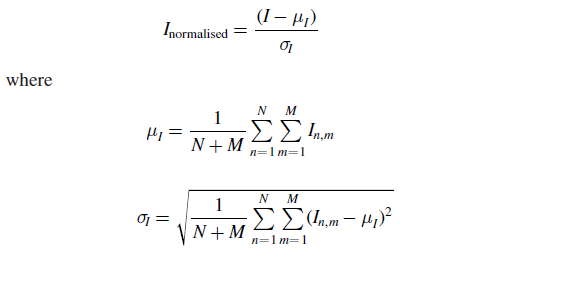
Figure : Grasp recognition. This figure is cited from [6].

To learn this abstract representation, we use convolutional neural network (CNN) architecture. There is mounting evidence that CNN based structures can learn and classify visual patterns efficiently if provided with a large amount of training (labeled) samples.

In ALOI dataset, there are 1000 objects, 250 of them have been photographed at a second zoom rate. We discarded these 250 objects. For each of the remaining 750 objects, the database includes 72 pictures, taken at 5\_ intervals against a black background.

We first selected 473 of the objects in four different classes of pinch,tripod, palmar wrist neutral and palmar wrist pronated. Otherobjects were either not graspable or could be picked withmore than one grasp type. Then later due to imbalanced class distribution, we prefer to sample the dataset. Some instances are deleted from the over represented class, under sampling. This leads to use about 356 objects.

In preprocessing, each image *I,* was first converted to grey scale and was downsampled to an *N* = 36 by *M=* 48 image. Empirically, we found that image normalization, prior to the CNN setting, improved the final accuracy. Therefore, each image was normalized by:



We used pre-trained CNN from [6]. Figure 12 shows the implemented two layers CNN architecture. We used five kernels of size 5×5 and the resultant feature maps were sub-sampled by max-pooling by a factor of two. We applied the maxpooling operation to ensure salient elements in each feature map are retained.

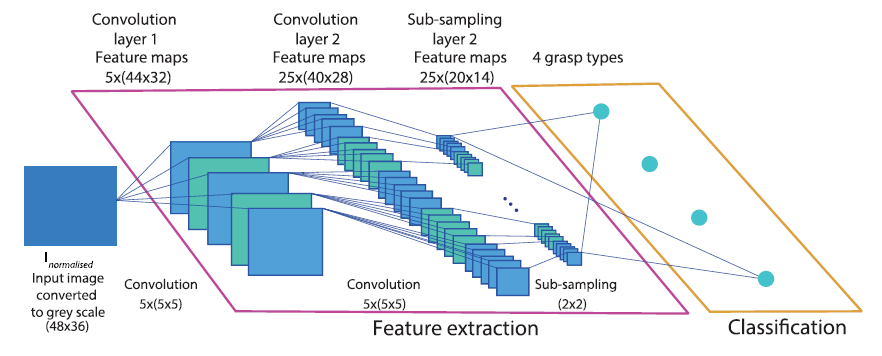


Figure : Two layers CNN architecture. This figure is cited from [6].

Many activation functions are tested like Scaled exponential linear unit, sigmoid, Tanh, rectified linear unit (ReLU) and leak ReLU. Empirically it is found that the ReLU function, figure 6, results in the highest performance and hence we used it in this study.



Figure : ReLU activation function.

Training was carried out through back propagation using the adaptive momentum estimation, Adam algorithm, for optimizing the learned filters within each iteration. Over-fitting is avoided by using Dropout regularization which randomly shuts down some neurons in each iteration. Here, the percentage of dropped neurons is equal to 20%. We built the model by Keras package using TensorFlow backend in python.

# Materials and Tools:

Due to the following reasons we had to run our systems on two different platforms First was PC Running Windows and Second was Raspberry-Pi Running Rasperian (Linux)

## Problems :

We had a problem to get the whole process working on Raspberry as it was not possible to install Tensorflow on Raspberry as its " Arm based" which is not officially supported by TensorFlow team.

So we attempted to install Tensorflow on Windows but still no success in acquiring EMG on Windows as we were using Open-Myo library for acquiring EMG which supports only Linux,

So after several Trials which comes with a price that I had re-install Python Package on my Windows  machine (as my python was 32-bit ) I got to acquire EMG from Windows using MYO-python Library  
Luckily after that I found a [repository](https://github.com/samjabrahams/tensorflow-on-raspberry-pi) that unofficially  helps with installing Tensorflow on Raspberry, so I got it installed on Raspberry-Pi

So now we got our Code to work on PC (windows) and Raspberry-Pi (Linux)

We have Working Online Scenario (or semi-online) on PC (Windows) on this [Repository](https://github.com/hananabilabd/Computer-Vision-and-Neural-Machine-Interface-for-Upper-Limb-Prostheses): <https://goo.gl/AwFrQ7> with its Customizable GUI and working Online Scenario on Raspberry on this [Repository](https://github.com/hananabilabd/EMG-Classification-Visualization-using-MYO-ArmBand-Raspberry-Pi/blob/master/README.md): <https://goo.gl/C4Wr7u> also with its customizable GUI

Note that we are Acquiring EMG on Both are with Different Bluetooth Devices and different acquiring protocols so it may differ in Frame speed or something as on Raspberry um using Raspberry Bluetooth for Acquiring and on Windows um using the MYO Bluetooth Dongle which may be little faster and disconnects after a longer period than on Raspberry

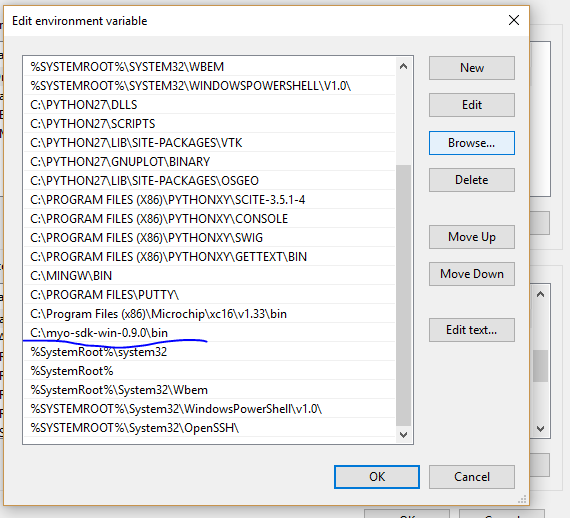
Discovered that slowness of frames while operating on MYO on Linux, was because we accessed the Filtered EMG, moreover the Raw EMG gives double or about 3 times the number of frames produced by Filtered EMG Now we have discovered the issue which was causing the slowness of the frames we get from Linux

Now we got to Acquire EMG from Raspberry-Pi and we got to do training and testing (Online scenario) on Linux

So Project is Created on Windows Using MYO ArmBand with MYO-Python

**Installation and usage of MYOPython** (https://github.com/NiklasRosenstein/myo-python)

1. you need to install myo-python library with the command PIP
2. $ pip install myo-python
3. you need to put myo-sdk into Environment Variables like this



4. You need to run ArmBand Manager and choose to connect the myo

5. Open any of the examples codes , it should run then

The Myo-Python module only works on Windows,

**Installation TensorFlow on Windows**

Unfortunately until Now TensorFlow supports Python 3.5.x or 3.6.x 64-bit only on Windows untill now

It is strongly recommended to download Winpython Packagefrom (https://winpython.github.io/) or https://sourceforge.net/projects/winpython/files/?source=navbar) if you don't have python 3

Installing it with native pip

To install TensorFlow, start a terminal. Then issue the appropriate pip3 install command in that terminal. To install the CPU-only version of TensorFlow, enter the following command:

```cmd

$ pip3 install --upgrade tensorflow

```

To install the GPU version of TensorFlow, enter the following command:

```cmd

$ pip3 install --upgrade tensorflow-gpu

```

**Installation PyQt4 on Windows**

1. if You already have PyQt5 just Run :

```cmd

$ pip uninstall PyQt5

```

2. download the package from this website (https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyqt4) according to your python Version (python 2 or 3 )

3. cd to directory of the downloaded package

4.

```cmd

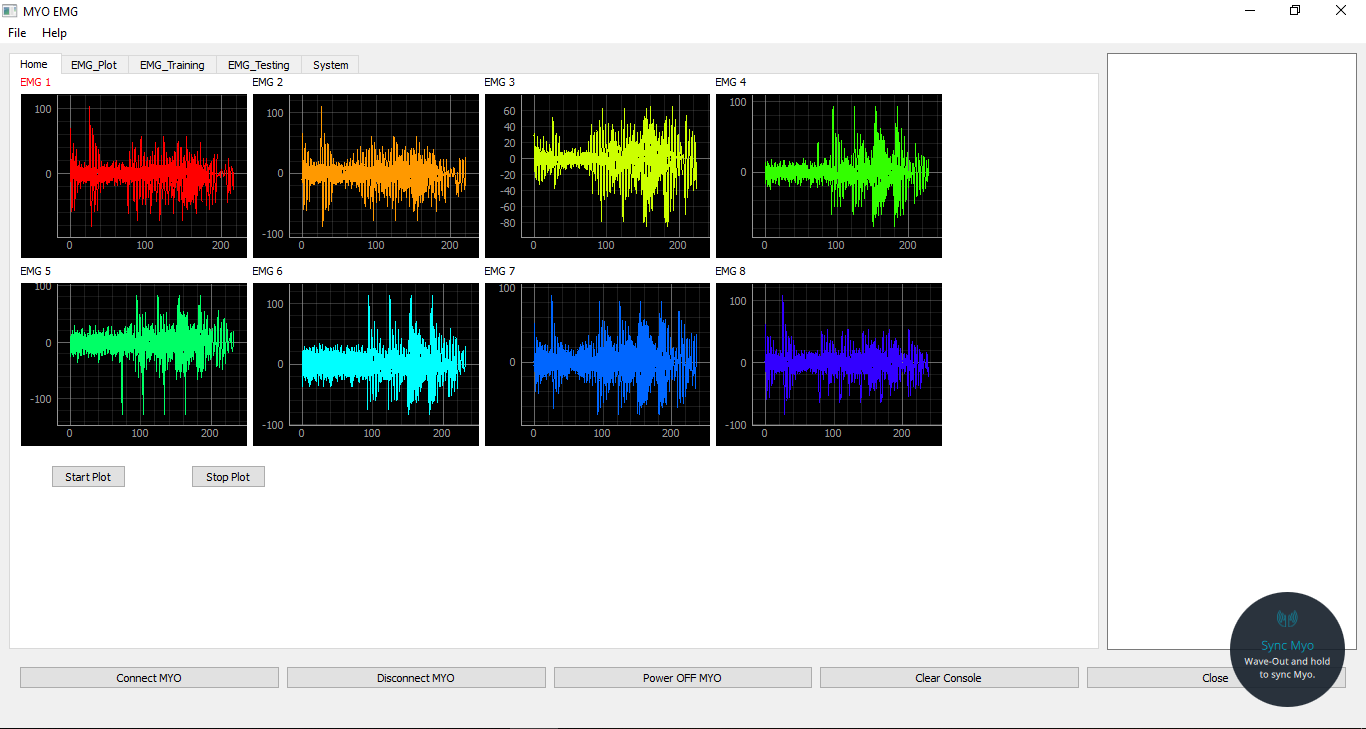
$ pip install packageName.whl

```

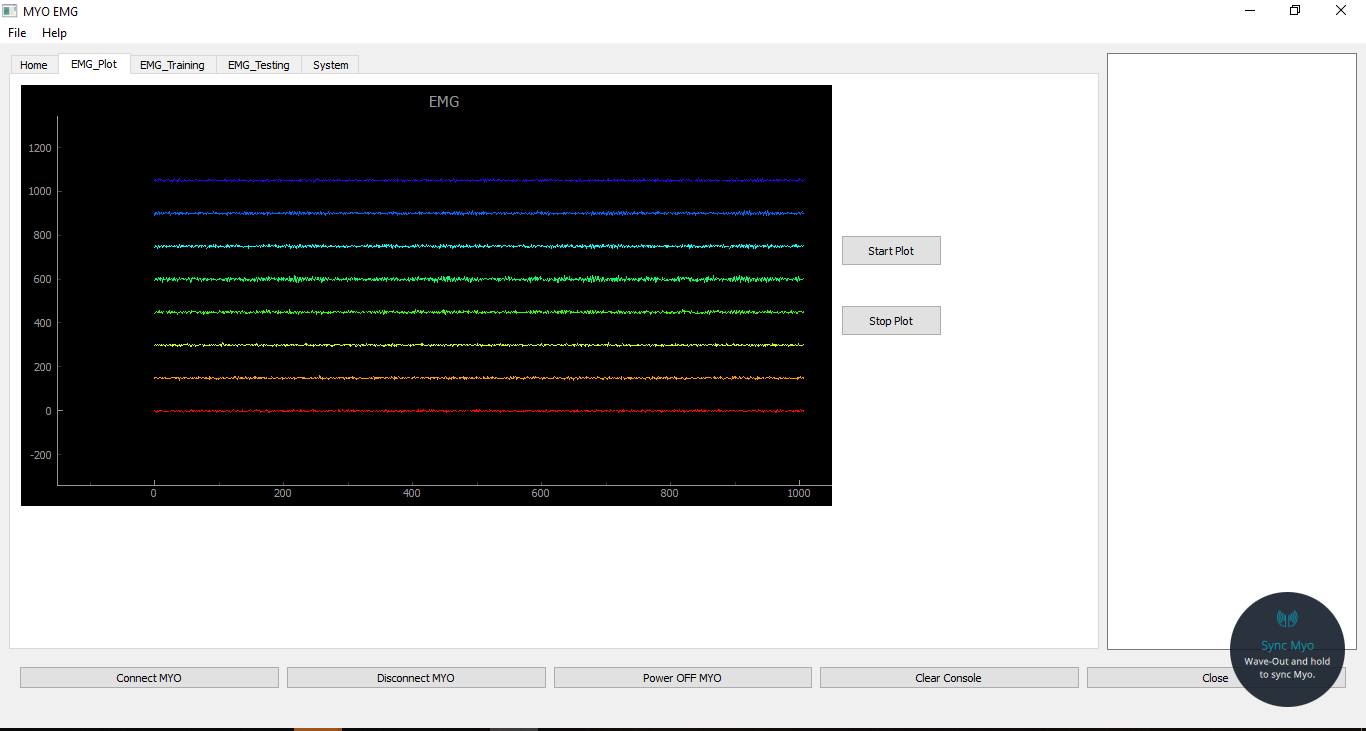
**Video** (https://youtu.be/Z\_KbUj3OziE)

**Screenshots**

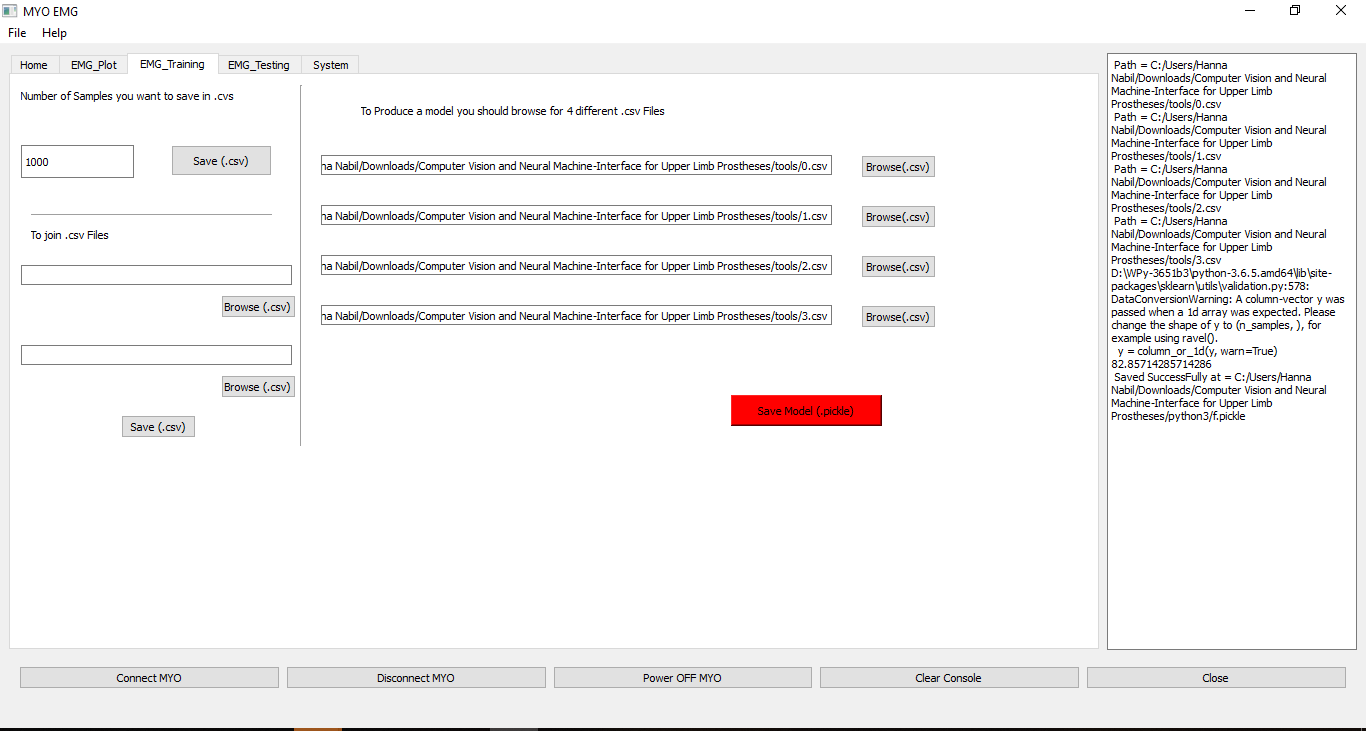
EMG Plot Graph1:



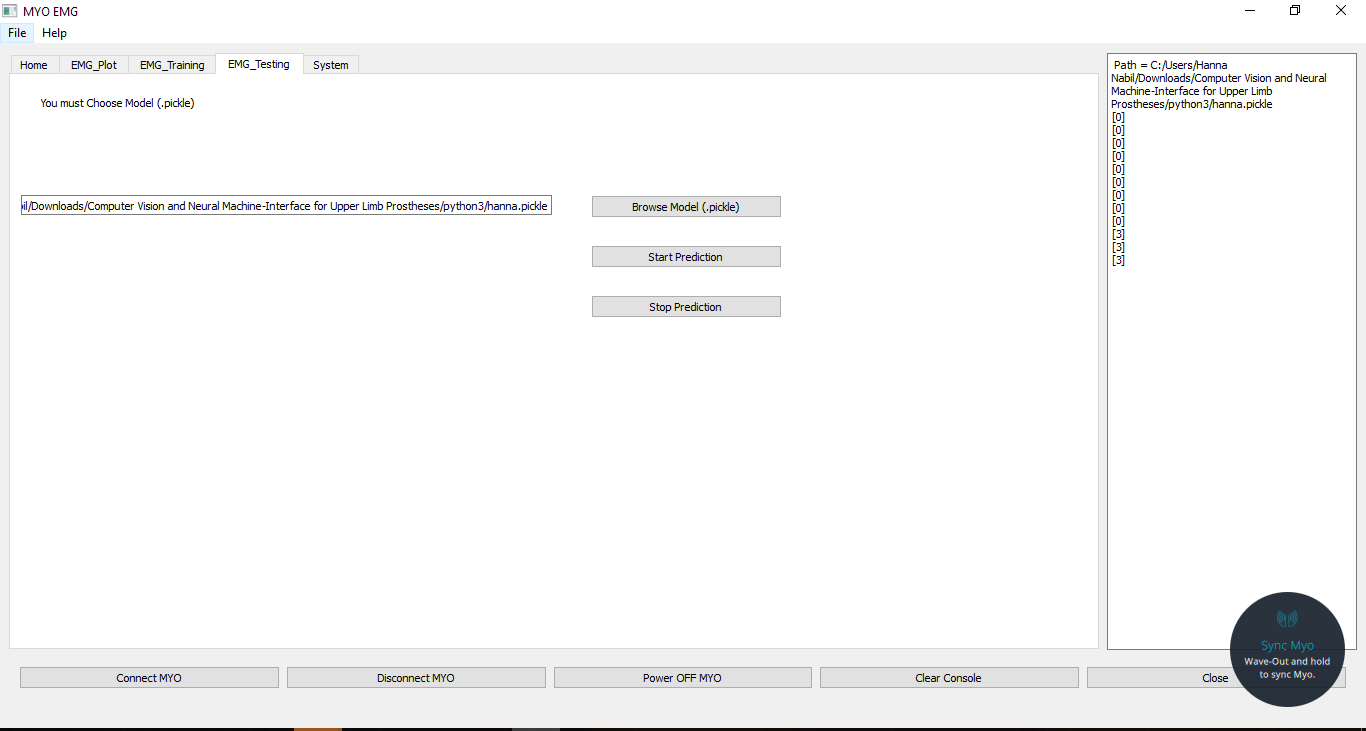
EMG Plot Graph2:



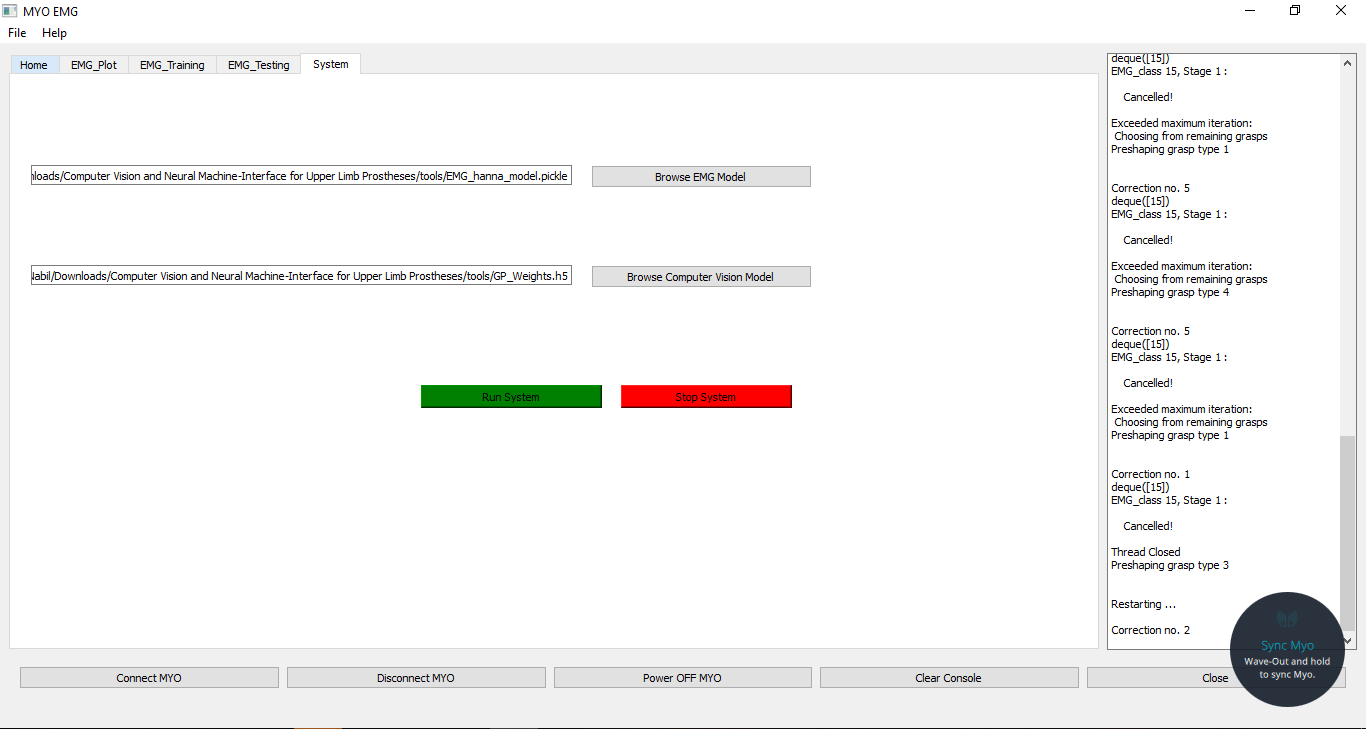
EMG Training:



EMG Testing:



Computer Vision & EMG system Test:



Project is Created on Raspberry-Pi Using MYO ArmBand with Open-Myo

Python module to get data from a Myo armband using a generic Bluetooth LE interface.

## Installation and usage of Open-MYO:

This module works with generic Bluetooth LE antennas (e.g. CSR V4.0, Cambridge Silicon Radio or the Bluetooth interface integrated in the Raspberry Pi 3/Raspberry Pi Zero W). This module does not work with the Bluetooth antenna included with the Myo armband because it uses a propietary protocol from Bluegiga. For further information, kindly refer to: (https://github.com/Alvipe/Open-Myo)

This module requires the bluepy (https://github.com/IanHarvey/bluepy) Python module to work. To install it, run:

``$ sudo pip install bluepy``

All code using the bluepy module must run with root permissions. To run the example code, execute:

``$ sudo python GP.py``

The Open Myo module only works on Linux , as the bluepy module is only available for Linux.

**Installation PyQt4 on Linux**

```cmd

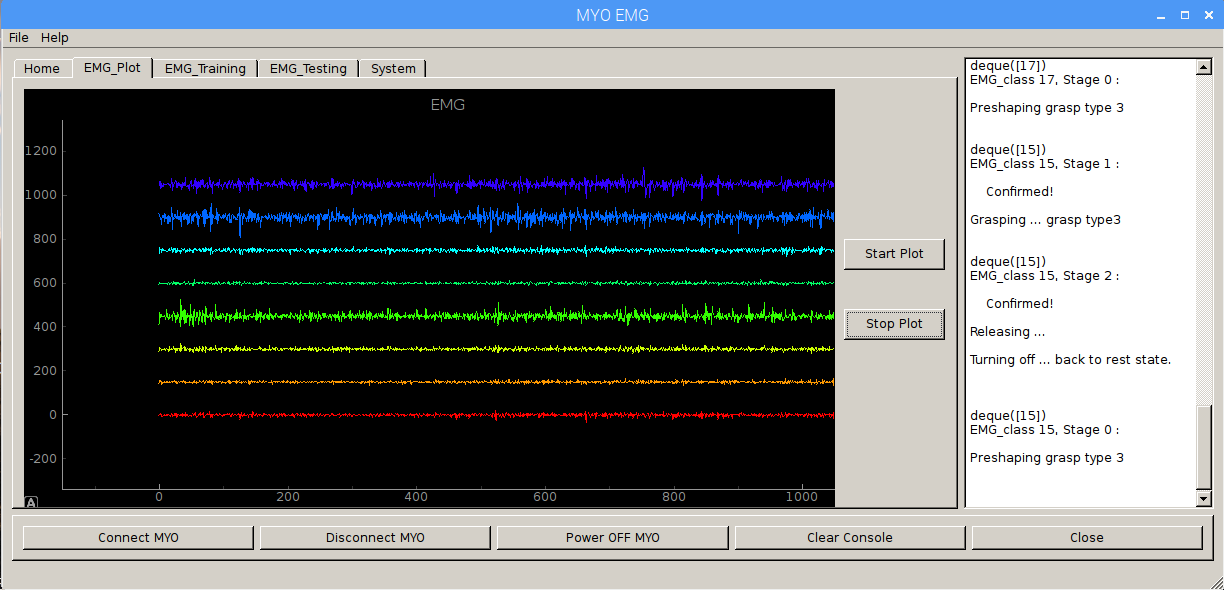
$ sudo apt update

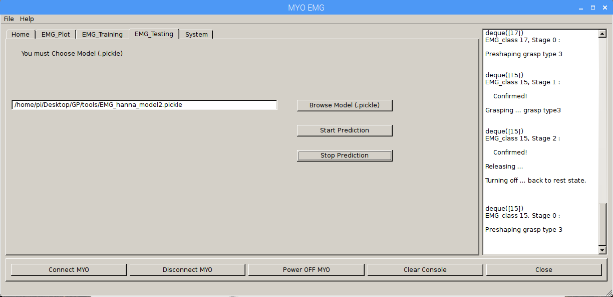
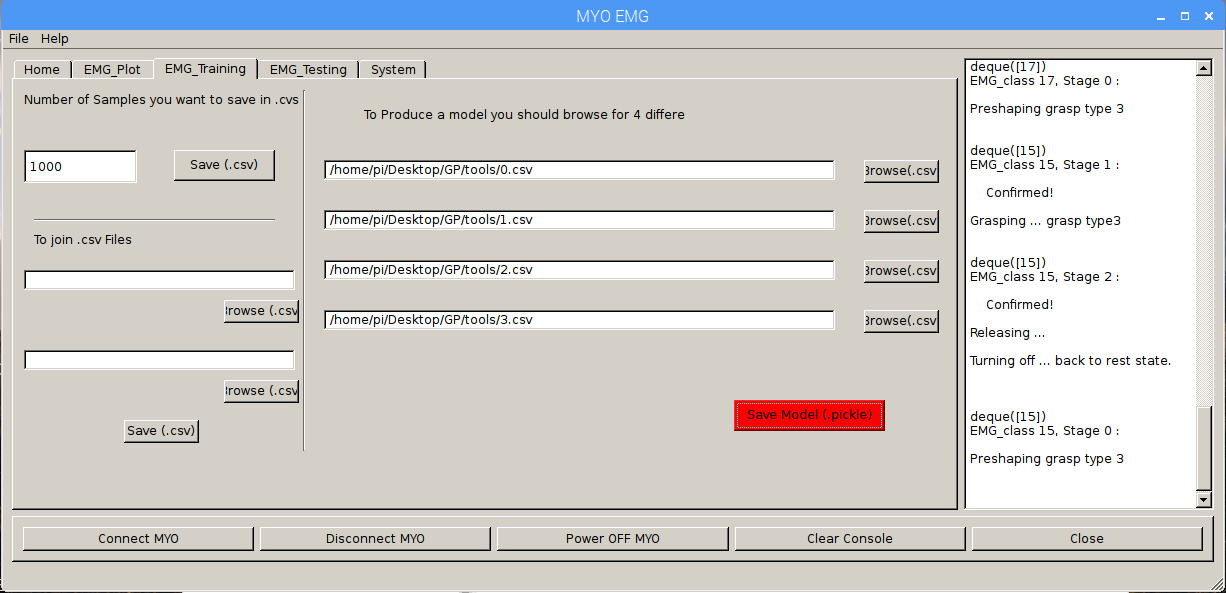
$ sudo apt install pyqt4-dev-tools

```

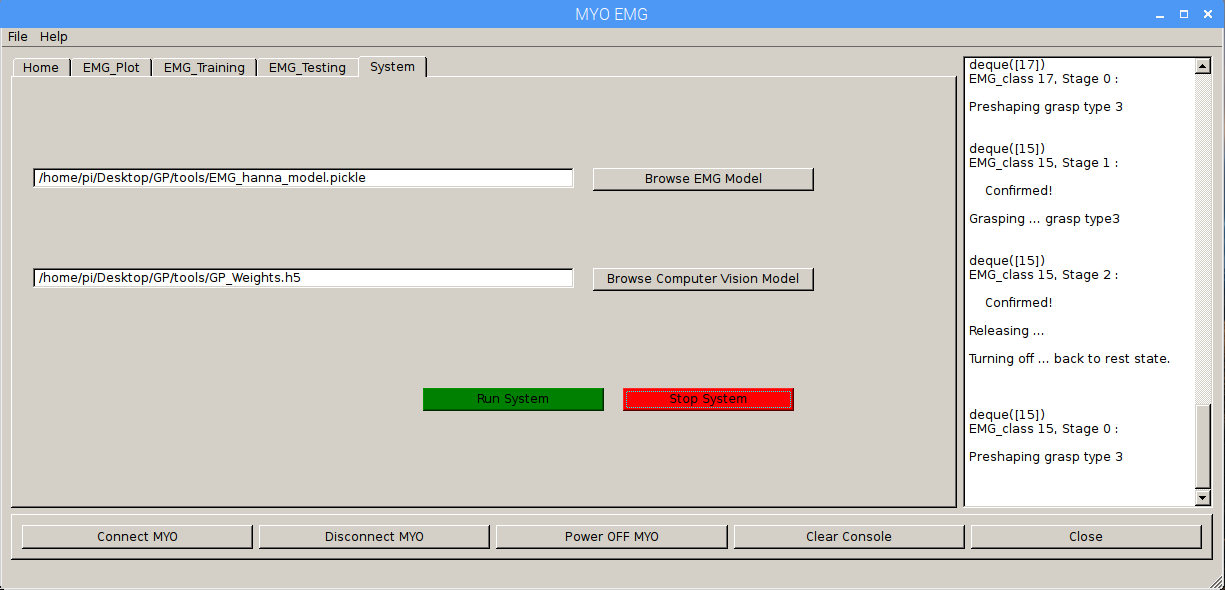
**YouTube Video** (<https://youtu.be/xmgbvkJEDqI>)

**Screenshots**

EMG Plot Graph:

EMG Training and Testing:

Computer Vision & EMG system Test:



## III. Result:

* For EMG, after applying this method on Ninapro Dataset, we got a reasonable accuracy, around 75 % in most subjects. Figure14 show accuracy in some subjects.

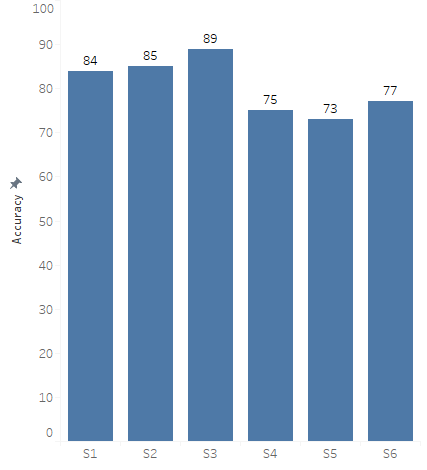


Figure 14: EMG result.

* Regarding the computer vision, in Within-object cross validation (WOC)., we evaluated the ability of the proposed structure in classifying previously seen objects. The training set included 90% (65 of 72) of the views for each object in each grasp class. The remaining 10% of the views for each object were allocated to the testing set.

In Between-object cross-validation (BOC). To be able to identify the appropriate grasps for unseen objects, we carried out the BOC test. In the BOC scheme, an object and its views were either wholly seen or unseen.

After applying the CNN architecture on ALOI dataset, we got accuracy around 90% in WOC and 71% in BOC. Figure15 illustrates our result.

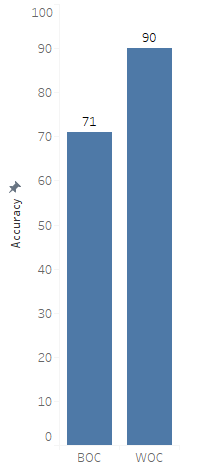


Figure 15: EMG result.

* On online scenario, We Had a working Solid System that Takes EMG Signals on Real-time From MYO Armband which Gives us 200 Samples (Each sample have 8 Values as we are dealing with 8 electrodes) so we Put this EMG Signal in a thread that runs in a loop , then Output of EMG Acquiring is passed to Predict Function actually we pass every 512 samples at first time to predict Function and we then Do the Overlapping , so wait for every 128 samples and pass them to predict Function and delete the oldest 128 from EMG array in predict Function , so we take new 128 samples and delete the oldest 128 samples , after that we make pre-processing (Filtration) and we pass it to get predictors and Finally we have a result class out of 4 classes.

We then use this output class to do the Fusion between computer Vison and the Resulted EMG.

Let’s Talk about we can reach to get someone to produce a class in EMG.

## Procedure of EMG Training:

1. Open EMG Training Widget.
2. Put number of samples in the textbox.
3. Browse for the location you want to put the output .csv file in.
4. Click save.

Now you need to repeat this step 4 times with any desirable movements you want to classify between them.

After that we will have 4 .csv Files

We Then Browse for these 4 .csv files and we Click to produce (.pickle)

When we have the (.pickle) the trained model of our 4 Movement.

## Procedure of EMG Training:

We now have two options:

\*First is to run the EMG prediction only.

\*Second Option is to run the whole system.

So with the first option to run the EMG only you will have to browse for the (.pickle) File we had save from Training Stage , Once we browse for our file we can start Prediction and the classes got printed in the right panel of the GUI

When we go with the second Option we will have to browse for the computer Vision model and the EMG Model (.pickle) and then we can start the system so the system will produce EMG class and the EMG class will Fused with the Decision from the Computer Vision

Note that we had a separate Thread for Each Module of EMG Prediction or System or even EMG Plotting.

### Final Result:

So, to sum up with the amputee wearing MYO we can predict his hand movement and with fusion with camera we can predict grasp type

Chapter Six

CHALLENGING PROBLEMS AND FUTURE SIGHT

Challenging problems and future sight

# Solved problems:

1. The system takes some time in carrying out any computations and does not receive EMG signals meanwhile.

**Solution:** Using a multithreaded integration algorithm to perform all actions simultaneously and using a queue to restore and retrieve EMG classes for the communication between threads.

1. System accuracy depends on two artificial intelligence branches with large room for error.

**Solution:** Allow user to correct actions and design algorithm to assist with corrections. The algorithm detects repeated corrections and acts accordingly.

1. **Imbalanced Classes in Computer Vision:**

Most real-world classification problems display some level of class imbalance, which is when each class does not make up an equal portion of your data-set. It is important to properly adjust your metrics and methods to adjust for your goals. If this is not done, you may end up optimizing for a meaningless metric in the context of your use case.

For example, you may have a 2-class (binary) classification problem with 100 instances. A total of 80 instances are labeled with Class-1 and the remaining 20 instances are labeled with Class-2. This is an imbalanced dataset and the ratio of Class-1 to Class-2 instances is 80:20 or more concisely 4:1.

You can have a class imbalance problem on two-class classification problems as well as multi-class classification problems. Most techniques can be used on either.

Most classification data sets do not have exactly equal number of instances in each class, but a small difference often does not matter.

There are problems where a class imbalance is not just common, it is expected. For example, in datasets like those that characterize fraudulent transactions are imbalanced. The vast majority of the transactions will be in the “Not-Fraud” class and a very small minority will be in the “Fraud” class.

In computer vision module, we noticed that after we selected the intended objects, Imbalanced class distribution problem is occurred. We found that for objects the CNN could not predict them correctly, the fault prediction was the same class.

When we visualize the distribution of the four classes, they appeared as in figure 16.

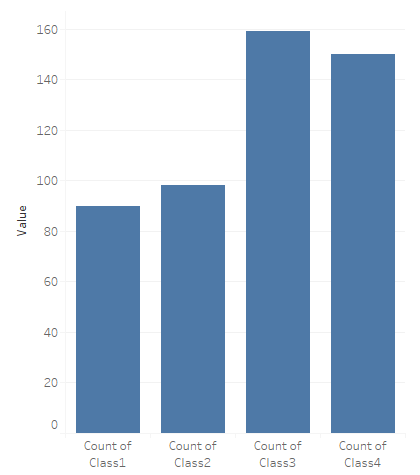


Figure 16: Classes distribution of the computer vision categories.

From figure 16, it is clearly that Class 3 and Class 4 has many objects compared to class 1 and class 2.

There are many methods to deal with imbalanced classes like: collect more data, resampling the dataset, changing performance matrix, generate synthetic samples or try penalized models.

Our choice was to resampling the dataset. You can change the dataset that you use to build your predictive model to have more balanced data.

This change is called sampling your dataset and there are two main methods that you can use to even-up the classes:

1. You can add copies of instances from the under-represented class called over-sampling (or more formally sampling with replacement), or

2. You can delete instances from the over-represented class, called under-sampling.

These approaches are often very easy to implement and fast to run. We went on the under-sampling way.

1. Overfitting in Computer Vision:

Supervised machine learning is best understood as approximating a target function (f) that maps input variables (X) to an output variable (Y).

An important consideration in learning the target function from the training data is how well the model generalizes to new data. Generalization is important because the data we collect is only a sample, it is incomplete and noisy.

In machine learning we describe the learning of the target function from training data as inductive learning.

Induction refers to learning general concepts from specific examples which are exactly the problem that supervised machine learning problems aim to solve. This is different from deduction that is the other way around and seeks to learn specific concepts from general rules.

Generalization refers to how well the concepts learned by a machine learning model apply to specific examples not seen by the model when it was learning.

The goal of a good machine learning model is to generalize well from the training data to any data from the problem domain. This allows us to make predictions in the future on data the model has never seen.

There is a terminology used in machine learning when we talk about how well a machine learning model learns and generalizes to new data, namely overfitting and underfitting.

Overfitting and underfitting are the two biggest causes for poor performance of machine learning algorithm.

Overfitting refers to a model that models the training data too well. See figure 17.

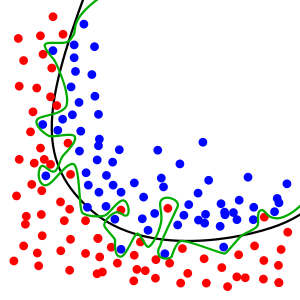


Figure 17: the red line represent the overfitting model. While the black one generalize the model. Cited from Wekipedia.

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize

Briefly, Overfitting is Good performance on the training data, poor generalization to other data.

There are many methods to avoid overfitting such as L1 regularization, L2 regularization, Data Augmentation, and Dropout regularization.

We used dropout regularization. Dropout is a regularization technique for neural network models proposed by Srivastava, et al [9].

Dropout is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.

As a neural network learns, neuron weights settle into their context within the network. Weights of neurons are tuned for specific features providing some specialization. Neighboring neurons become to rely on this specialization, which if taken too far can result in a fragile model too specialized to the training data. This reliant on context for a neuron during training is referred to complex co-adaptations.

You can imagine that if neurons are randomly dropped out of the network during training, other neurons will have to step in and handle the representation required to make predictions for the missing neurons. This is believed to result in multiple independent internal representations being learned by the network.

The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to overfit the training data.

1. Understanding the EMG data:

At first we’ve encountered some problems at understanding the EMG dataset and its structure and we didn’t know how we should access it and use it according to our project.

We had read the official documentation of it but there were some keys and terminologies we didn’t get from the first time like some keys called (‘stimulus’ , ‘repetition’) in ninapro dataset.

Solution: we had used the graphs to visualize the data so we could got a lot of information about it which made us understand the data easily and enabled us to use the the data as we need

1. Accessing the datasets:

After using google colab for our deep learning model to increase the performance, we had faced a problem with accessing the datasets we needed.

**Solution:** we’ve uploaded the datasets to github so it can be faster accessed by google colab and to be used in serialization process.

we’ve used a digital ocean droplet which is a VPS with very high download and upload speeds which enabled us to deal with downloading and uploading the dataset in very little time.

1. EMG data acquisition protocol

After successfully implementing the offline scenario of EMG classifier using ninapro dataset, we’d needed to make emg dataset ourselves so we can use it in a real time Emg classifier so we had needed to choose our acquisition protocol.

**Solution:** we had decided to use MYO armband ,which is a gesture recognition device worn on the forearm and manufactured by Thalmic Labs, to acquire the EMG data, we had tried with different subjects and different protocols to get reasonable accuracy.

At first we’ve encountered some problems with libraries used to get the EMG data from MYO cause some of them aren’t compatible with our hardware and some of them have bugs and issues which aren’t solved cause they no longer have support from their developers. But finally we’ve founded a good library to deal with acquisition the data.

1. Using overlapping to increase the accuracy of EMG classifier

at first, we didn’t got the reasonable accuracy that we had expected, and to solve that we found out that we can use window overlapping to increase the accuracy.

-there’s a tradeoff between getting a high accuracy and the time, so we had needed to choose a reasonable window length and overlapping factor.

So We've tried different overlapping factors like (256,128,64) ,And we stick with

1. windows length:512ms

2. overlapping factor: 128

as it gives us a reasonable accuracy with reasonable time.

- Needing for a parallel processing for the deep learning model

1. We had needed a GPU to train the deep learning models so we can take advantages of parallel processing to save time,but unfortunately we didn't have a suitable GPU for that.

**Solution:** we've used Google Collab to solve that problem which is a free cloud service with a supported free GPU which anyone can use to develop deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV

# Unsolved problems and suggested solutions:

1. The zooming effect in the computer vision branch significantly affects the computer vision’s ability to determine grasp type. This result is coherent with the fact that grasp type depends on object size in real life.

**Proposed solution:** Integrate a distance sensor with the system to estimate the size of the object and preprocess images respectively.

1. Computer vision branch handles only photos taken in ideal conditions.

**Proposed solution:** Use a multilayered network to handle more variability in photography conditions.

# Future sight:

1. Add proportional control to adjust force exerted on object as well as size of grip. One idea to apply this addition is to increase the number of EMG classes to six classes and assign two of them for increasing and decreasing force and size.
2. Add feedback sensors to the system to give information similar to that of a real hand, i.e. temperature of object, smoothness or roughness of surface … etc.

Chapter Seven

GAINED SKILLS

Gained skills

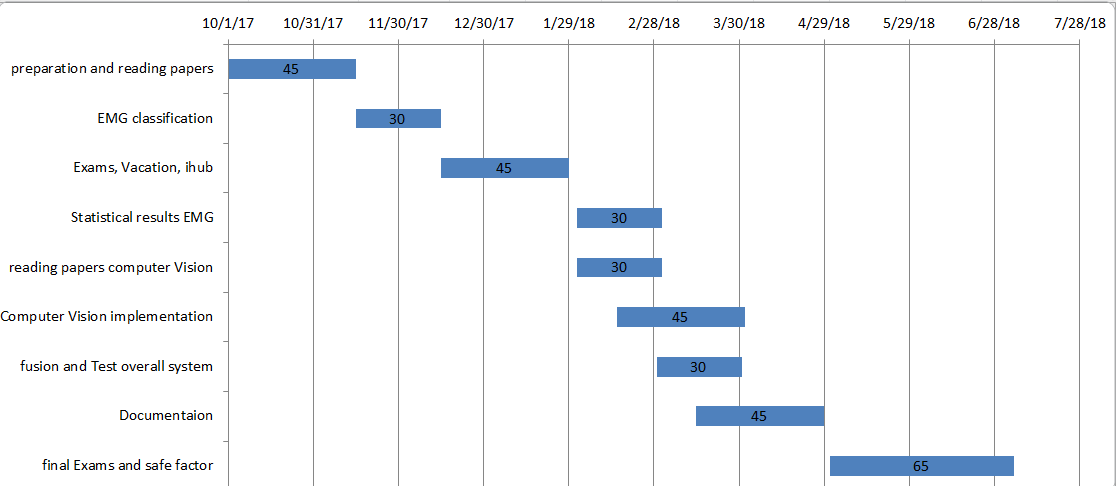
# Technical skills:

1. Choosing the suitable programming tools.
2. Professional collaborative programming.
3. Using Python and related tools.
4. Utilizing Jupyter notebook for documented programming.
5. Machine learning application in real life problems.
6. Statistical analysis.
7. System design.
8. Empirical evaluation.
9. Deep learning application and tools.
10. Good programming practices.
11. Code documentary and maintenance.

# Soft skills:

1. Project management.
2. Time management.
3. Teamwork.
4. Resources management.
5. Research and guided self-learning.
6. Working in a multidesplinary environment.
7. Professional and formal correspondence.
8. Technical writing.
9. Working within a process.
10. Professional follow-up and feedback.

# Gantt chart:



Chapter Eight

REFERENCES

References

[1] Manfredo Atzori, Arjan Gijsberts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mittaz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto & Henning Müller. “Electromyography data for non-invasive naturally-controlled robotic hand prostheses”

[2] K. Englehart and B. Hudgins, “A robust, real-time control scheme for multifunction myoelectric control,”

[3] Sungtae Shin, Reza Langari and Reza Tafreshi, “ a performance comparison of EMG classification methods for hand and finger motion’’

[4] Mohammad Tafhim Khan and M. Tahseenul Hasan, “Comparison between kNN and SVM for EMG Signal Classification ’’

[5] Adenike A. Adewuyi, Levi J. Hargrove and Todd A. Kuiken “Evaluation EMG features and classifer selection for application to partial hand prothesis control ’’

[6] Ghazal Ghazaei, Ali Alameer, Patrick Degenaar, Graham Morgan

and Kianoush Nazarpour “Deep learning-based artificial vision for

grasp classification in myoelectric hands’’

[7] Meena AbdelMaseeh, Tsu-Wei Chen, and Daniel Stashuk “Extraction and Classification of Multichannel Electromyographic Activation Trajectories for Hand Movement Recognition’’

[8] Jan Mark Geuserbroek, Gertjan J.Burghouts and Arnold W.M. Smeulders “The Amsterdam Library for Object Images”

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov “a simple way to prevent neural network from overfitting”

[10] Alberto Cairo “The functional Art ’’

[11] Došen S and Popović D B 2011 Transradial prosthesis: artificial vision for control of prehension Artif.

[12] Marković M, Došen S, Cipriani C, Popović D B and Farina D 2014 Stereovision and augmented reality for closed-loop control of grasping in hand prostheses J. Neural Eng.

[13] Saxena A, Driemeyer J and Ng A 2008 Robotic grasping of novel objects using vision Int. J. Robot.

[14] Kootstra G, Popović M, Jørgensen J A, Kuklinski K, Miatliuk K, Kragic D and Krüger N 2012 Enabling grasping of unknown objects through a synergistic use of edge and surface information Int. J. Robot. Res.

[15] Lenz I, Lee H and Saxena A 2015 Deep learning for detecting

robotic grasps Int. J. Robot. Res.

[16] Kopicki M, Detry R, Adjigble M, Stolkin R, Leonardis A and Wyatt J L 2015 One-shot learning and generation of dexterous grasps for novel objects Int. J. Robot. Res.