Real-time Pattern Recognition for Prosthetic Hand

Author:

Mario Alejandro Benítez López

Advisors:

Carlos Francisco Rodríguez Herrera, Dr. Eng.

Jonathan Camargo Leyva, M.Sc.

Master of Science Thesis

Department of Mechanical Engineering

Faculty of Engineering

Universidad de los Andes

July 2019

1 CONTENT

2	Abstract.....	4
3	Acknowledgments.....	5
4	Nomenclature	6
5	List of Figures	8
6	List of Tables	9
7	Introduction	10
8	Previous Work.....	12
9	State of the Art.....	15
9.1	Grasping and Manipulation in Robotics.....	15
9.2	Control of Myoelectric Hands	17
10	Objectives.....	20
11	Methodology.....	21
11.1	Grasping Design	21
11.1.1	Palm of the hand.....	21
11.1.2	Thumb	23
11.1.3	Fingertips.....	26
11.2	Pattern Recognition	28
11.2.1	EMG Feature Set	28
11.2.2	EMG Data Acquisition	29
11.2.3	ANN Deployment	30
11.3	Actuation and Electronic Systems.....	32
11.3.1	Actuation system	32
11.3.2	Electronic system	35
12	Results and Discussion	38
12.1	Real-Time Pattern Recognition	38
12.2	Robotic Arm	42
13	Cost of the Hand	45
14	Conclusions	46
15	Future Work	47
16	Bibliography	48
17	Annexes.....	52

17.1	Archives Repository	52
17.2	Arduino Pattern Recognition Code	52
17.3	PCB Circuit Schematic	61

2 ABSTRACT

Control of myoelectric prosthetic hands is still an open problem, currently, most commercial prostheses use direct, proportional or Finite State Machine control for this purpose. However, as mechanical design advances, more dexterous prostheses with more degrees of freedom (DOF) are created, then a more precise and intuitive control for the user is required. State of the art has focused in the use of pattern recognition as a control strategy with promising results. Studies have shown similar results to classic control strategies with the advantage of being more intuitive for the user. Many works have tried to find the algorithms that best follows the user's intention. However, deployment of these algorithms for real-time classification in a prosthesis has not been widely explored. This project addresses this problem by deploying and testing in real-time an artificial neural network (ANN). The ANN was trained to classify three different motions: no grasp, precision grasp and power grasp in order to control a two DOF trans-radial prosthetic hand with electromyographic signals acquired from two channels. Static and dynamic tests were made to evaluate the ANN under those conditions, 95% and 81% accuracy scores were reached respectively. Our work shows the potential of pattern recognition algorithms to be deployed in microcontrollers that can fit inside myoelectric prostheses. On the other hand, a prototype of a prosthetic hand that is able to physically replicate the classified actions was developed.

3 ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the following people that have supported me during the realization of this thesis:

To my advisors Carlos Pacho and Jonathan for accepting to assess me in this work, and for their continuous support, patience and guidance throughout this project.

To my dear Lloit who always believed I could carry out this project and encouraged me in every stage of it.

To my friends of the master's program that contributed with many brilliant ideas for the project.

To foundation Materialización 3D that welcomed me with open arms and helped me to conduct tests with amputees.

To all the volunteers that helped me to validate the results of this project.

To my mother and grandmother who have not only supported me in this project but in every step of my life.

4 NOMENCLATURE

F	Force
F_{req_g}	Required grasping force
$\bar{F}_{ext_j}$	External force
$\bar{g}$	Gravitational force
G	Predicted grasp
$\bar{G}$	Grasp Matrix
$\bar{G}^+$	Pseudoinverse Grasp Matrix
$DASDV$	Difference Absolute Standard Deviation
$Delay_{PR}$	Delay associated to Pattern Recognition classification
LW	Window length to study of the EMG signal
L_{pull}	Required pulling length to close the hand
L_i	Output value of output neuron
MAV	Mean Absolute Value
N_p	Number of probabilities for average calculation
N_w	Number of threads in the worm
N_{wg}	Number of teeth in the worm gear
$N(G)$	Null space of the Grasp Matrix
P	Power
$P(G_i)$	Predicted probability of a grasp
$\bar{P}(i)$	Mean of the predicted probabilities of a grasp
rms	Root Mean Square
$\bar{r}_{F_{ext_i}}$	Position of the external force relative to center of mass of the object
$\bar{r}_{\lambda_i}$	Position of the contact force relative to center of mass of the object
SP	Sampling Period of the EMG signal
SSC	Slope Sign Change
t_{req_g}	Required grasping time
T	Torque
$T_{out_{gb}}$	Output torque of the gear box
T_{req_g}	Required grasping torque

V	Linear velocity
$V_{out_{wg}}$	Output linear velocity of the string pulled by the worm gear
V_{req_g}	Required velocity of the string when grasping
VAR	Variance
WL	Waveform length
ZC	Zero Crossings
$\bar{\gamma}$	Parametric arbitrary vectors
$\emptyset_{cyl}$	Diameter of the cylinder on top of the worm gear
$\emptyset_{wg}$	Diameter of the worm gear
ω_{wg}	Angular velocity of the worm gear
$\omega_{out_{gb}}$	Output angular velocity of the gear box
$\bar{\lambda}$	Applied contact force

5 LIST OF FIGURES

Figure 1. Hand designed by Camargo grasping an object. [15]	12
Figure 2. Prosthesis prototype developed by Sánchez [16].....	13
Figure 3. Prosthesis prototype grasping an object controlled by an amputee. [20]	14
Figure 4. Movement constrains of the contact models in grasping. a) Hard Finger with No Friction. b) Hard Finger with Friction. c) Soft Finger with Friction.	15
Figure 5. Planar model of a precision grasp including gravity g and a grasp force λ on second contact point within the friction cone. [22].....	16
Figure 6. Palm pieces of the hand in previous design.....	21
Figure 7. CAD simulated finger shafts clearance	21
Figure 8. Pieces of the new palm design.....	22
Figure 9. Palms of the prosthesis, old (down) and new (up).	22
Figure 10. Possible approaches for grasping in previous prototype. [17]	23
Figure 11. DOF of each phalange in the thumb. AA: Adduction-Abduction. FE: Flexion-Extension. IP: Interphalangeal joint. MCP: Metacarpal joint. CMC: Carpometacarpal joint [44]	24
Figure 12. Pulleys for actuation of the thumb.	25
Figure 13. Grasping sequence of the old thumb.....	26
Figure 14. Grasping sequence of the new thumb.....	26
Figure 15. Curing of the fingertips in an oven.	27
Figure 16. New fingertips in distal phalanges.	28
Figure 17. Actions instructed the subjects to complete. a) No grasp b) Precision grasp c) Power grasp ..	30
Figure 18. Muscles where EMG signals were recorded. a) Forearm muscles were used for AB and HM subjects. b) Upper arm muscles were used for THA subject.	30
Figure 19. Closing mechanism of the fingers. [20].....	32
Figure 20. Experimental set-up for determining required force for closing the thumb and the fingers. ..	33
Figure 21. Actuation system for closing the prosthetic hand.	34
Figure 22. Protoboard for EMG acquisition and Teensy 3.2.....	36
Figure 23. PCB designed for the prosthetic hand.	36
Figure 24. Available grasps for the prosthesis. a) Precision grasp b) Power grasp	43
Figure 25. Finger motor velocity for 5 different tests.....	43
Figure 26. Thumb motor velocity for 5 different tests.	44

6 LIST OF TABLES

TABLE 1: Required forces to close the thumb and the fingers.	33
TABLE 2: Possible parameters for the actuation system.	35
TABLE 3: Accuracy percentage of training stage of the ANN	38
TABLE 4: Accuracy percentage of static test.....	38
TABLE 5: Accuracy percentage of dynamic tests.	39
TABLE 6: Confusion matrix of subject 4 for his dynamic test.	40
TABLE 7: Comparison of results with similar studies. All of them recorded 2 EMG channels.	41
TABLE 8: Classification accuracies and processing time for a 200 ms window length and a sampling time of 5 ms with different <i>NP</i>	41
TABLE 9: Classification accuracies and processing time for a 150 ms window length and a sampling time of 5 ms with different <i>NP</i>	41
TABLE 10: Classification accuracies and processing time for a 150 ms window length and a sampling time of 3.75 ms with different <i>NP</i>	42
TABLE 11: Costs of the prosthetic hand prototype.....	45

7 INTRODUCTION

Amputation is an acquired condition that results in the loss of a limb such as fingers, arms, feet, toes, legs, and it can be caused by disease, trauma or congenital defects [1]. Amputation of a limb severely affects the life of a person physically and emotionally. Physically, the person loses functionality due to the missing limb that no longer performs the tasks it was used to do. In extreme cases, when multiple and higher levels of amputation are present the person will no longer be capable of working, walking and even taking care of herself. Emotionally, a grief sensation over the loss of the limb is normal in all the amputees, how they overcome this feeling depends on several factor such as level of disability, adequacy of cosmesis, cultural issues, presence of social supports, support of their loved ones, among others [2]. However, when the person is not able to effectively cope with the amputation, the grief could become in clinical depression that must be treated [3]. Currently, the main treatment for amputations is the use of a prosthesis, a device that is worn by the amputee and it is located where his missing limb should be. The objective of wearing a prosthesis is to restore some of the functionality lost by the user or at least being an aesthetical limb that hides the amputation, for this reason prostheses typically resembles the form of the lost limb.

There are no official statistics of how many amputees are Colombia, but due to the many years of civil war many of Colombian civils and soldiers have suffered of amputations. For instance, according to a study conducted by the Universidad Javeriana and the Military Hospital [4], from 2012 to 2014 1 in every 4 soldiers injured in combats suffered an amputation for a total of 251 amputations. Even though the high number of amputations in the country the access to prostheses is very limited, especially for low-income persons. The main reasons for this phenomenon are the lack of local companies that develop prostheses and the high cost of imported prostheses that health insurances do not cover. Thus, most of the access is achieved thanks to foundations that donate prostheses for free such as *Gilete*, *Materialización 3D*, *Cirec*, *Mahavir Kmina*, *Enabling the Future*, among others. On the other side, soldiers have special insurances granted by the government that cover the cost of the prostheses, but still those treatments are very expensive reaching 150 million Colombian pounds [5].

In order to quantify the magnitude, distribution and causes of amputations, data from a study that estimates limb loss in the United States from 2005 to 2050 is analyzed [6]. For instance, for 2005, 1.6 million of people were living with an amputation, and projections for 2050 say that this number will more than double, up to 3.6 million. Likewise, in 2020, 2.2 million of amputee persons are expected which will represent 0.7 % of the total population of the country. Also, there are more cases of lower limb amputations than upper limb amputations cases, with a ratio of 2:1. Finally, the main cause of amputation in 2005 was associated with diabetes and vascular diseases accounting for more than half of the total amputations (54%), trauma follows close (45%) and cancer and congenital causes stay behind (<1%). In future years diabetes is projected to keep growing as the main cause of amputation as diabetes cases would triplicate for 2050.

As is shown from the previous statistics, limb amputation is a condition that affects a considerably percentage of the population. Nowadays, there is a large market of prostheses that aims to ensure availability of prostheses for this population. As amputation cases are increasing, the market follows this trend, for 2025, it is expected to reach a size of USD 1760 million [7]. This is not particularly surprising, since the most advanced prostheses can go up to 120000 USD [8]; however, as it was earlier stated, in Colombia costs and availability limit access to this type of prostheses. Thus, part of the motivation of this

project is to develop a prototype of a functional low-cost hand prosthesis that helps to address these problems.

Hand amputation drastically reduces the capability of a person to do its daily activities such as working, studying, practicing sports, eating, among others. For this reason, many prosthetic hands are being developed and their ultimate goals are to dexterously replicate the movements of a human hand and make the control of the prosthesis an intuitive process for the user. Hand prostheses can be classified in two types, body powered and external powered. As its name suggests body powered prosthetics use the motion of other parts of the body, commonly the elbow and shoulder, to generate grasps. However, these type of prostheses have numerous limitations as stated by the users in several studies, some of these are: control of the prosthesis is not intuitive, it requires a large amount of physical effort to use, there is no sensory feedback to the user, prosthesis is not aesthetic [9].

On the other hand, external powered prostheses use the energy of an external battery to power motors that generate the movements of the hand. For this system to work, a control scheme that accurately follows the intention of the user is needed. The most common scheme is based on electromyography (EMG), prostheses that use this control scheme are known as myoelectric. EMG is a technique that records the electric potentials generated by the muscles in response to an electrical stimulus that comes from the motor nerves. The main requirements for myoelectric prostheses, according to the users, fall into three main categories: EMG sensing, control and feedback [10]. EMG sensing refers to the ability of the prosthesis to efficiently predict the user's intention. This means it could be able to identify different grasp types, determine speed and force of the desired grasp. Also, wrist motion and grasp type should be simultaneously distinguishable, and it must classify and actuate in less than 300 milliseconds for the user not to perceive a significant delay [11]. Control requirements aim to physically perform what the EMG sensing predicts, adding the constrain of maintain a stable grasp of the objects. Feedback must be an intuitive and adjustable system that continuously provides information about grasp, position and force.

On the other side, researchers have found the following challenges as the most relevant to overcome for robotic hands in the present time [12]:

- It is still difficult to robustly, adaptively and dexterously control a robotic hand given the high mechanical complexity of the devices
- It is not clear how to implement complex tasks using the robotic hand equipped with limited sensing capability in un-structural environment
- It is still an open question to represent and transfer human's manipulation skills to a robotic hand which has the similar configuration space

All these challenges and requirements are used as a roadmap for researchers and engineers to develop prostheses that successfully fill the requirements of amputees. As it is complicated to address all of them in one single project, we focused on guarantee part of the EMG sensing requirements while contributing to overcome the first challenge.

8 PREVIOUS WORK

The development of a low-cost prosthetic hand has been the focus of several undergraduate final career projects of the mechanical engineering department at Los Andes University in recent years. Usually, each student continues the project where the previous student finished and the goal of each project is to improve a feature of the hand such as design, manufacturing or control. It is worth mentioning that none of these projects has reached a fully functional prototype that an amputee person can wear. This section summarizes these projects and their biggest contributions to the development of the prosthetic hand.

At the beginning, the projects focused on the study and design of a robotic hand which established the basis for the posterior design of a prosthetic hand. In 2001, Duran studied the grasping action of human hands and subsequently designed a robotic hand based on his findings [13]. Then, in 2003, Flower built a new robotic hand, but focused his work on the actuation system required for the hand to grasp objects and validated the stability of the grasps performed by the hand [14].

At a second stage, the control of the robotic hand through muscular contractions is set as the main objective of the projects. In 2009, Camargo designed and built a new robotic hand that is controlled by mechanomyographic signals of the muscles [15]. He also introduced an important feature to the mechanical design of the hand, the under-actuation of the fingers which reduced the number of actuators required for closing the hand. Figure 1 shows a sequence of movements of the hand during the grasping of an object.

Figure 1. Hand designed by Camargo grasping an object. [15]

Afterwards, in 2014 Sánchez [16] resumed the project with a new objective to develop a prosthetic hand controlled by electromyographic signals. Thus, she redesigned the hand and builds a socket to fit inside all the components that the hand requires to work. Another significant contribution is the use of 3D printing as the manufacturing process of the hand which allows to use more complex geometries during the design of the hand. She also implemented a feedback system to the user, based on a vibration motor

located in the arm of the user that vibrates accordingly to the force exerted by the robotic hand. Figure 2 shows the prosthetic hand prototype Sánchez built.

Figure 2. Prosthesis prototype developed by Sánchez [16].

In 2015, Velandia studied the dynamic response of the robotic hand by modelling it as a mass-spring-damper system. He also designed and tested a PID controller and a sensor of the force exerted by the motors that actuate the hand [17]. In the same year, Arciniegas proposed the use of brain signals for the control of a virtual hand, for this purpose he used pattern recognition algorithms such as Artificial Neural Networks, Support Vector Machines and Bayesian Networks [18]. His results are based on the accuracy of the algorithms in predicting the user's intention, although they were not very good, they showed the potential of pattern recognition.

It is noteworthy that besides the mechanical engineering department the electrical engineering department also worked in projects related to prosthetic hands. However, they took a different approach, focusing on how to control the hand through the electromyographic signals generated by the muscle contractions of the user. In 2004, they proposed two algorithms for the offline classification of these signals: An Artificial Neural Network and a Multivariable Learning Data Analysis [19]. They reached an average accuracy of 85 and 81 % respectively which showed the potential of these algorithms for this application.

Finally, the author of this project previously worked on the development of the prosthetic hand for his undergraduate thesis in 2016. His job mostly consisted on designing components of the hand in order to be a step closer to a prosthetic hand that can be used by a person with trans-radial amputation [20]. Thus, he designed a socket where all the actuation and electronic systems could be fitted and did preliminary tests with an amputee subject. Figure 3 shows the amputee subject controlling through electromyographic signals the prototype developed.

Figure 3. *Prosthesis prototype grasping an object controlled by an amputee.* [20]

9 STATE OF THE ART

9.1 GRASPING AND MANIPULATION IN ROBOTICS

Prosthetic hands, robotic hands and industrial grippers must be capable of effectively manipulate and grasp objects to successfully carry out the task they are designed to do. Thus, when designing any of these devices it is necessary to understand the kinematic and dynamic models that describe the grasp. The action of grasping an object is divided in 3 stages: approaching, contact interaction and movement [21]. Approaching refers to how the hand moves towards the object and this could be represented by the position and orientation relative to the object. If they are not appropriate, grasping could not be completed. Contact interaction studies how the hand closes around the object to perform the intended grasp. The final stage of movement concerns on how the object is transported or manipulated by the hand after being grasped.

Another important concept to understand in the grasping of an object is how to model the interaction between the fingers of the hand or gripper and the object to be grasped. There are several models for this purpose where each model assumes a different condition of movement and rotation on the contact [22]. However, only the three most relevant are introduced, they are shown in the Figure 4. The first model is Hard Finger with No Friction, its first assumptions is that the object is very slippery, then the object can freely rotate and move relative to the finger except in the normal direction of the contact. Second assumption is that there is no significant strain in the surface of the finger and the object, so contact patch is very small and it can be modeled as a single point. Second model, the Hard Finger with Friction accounts for a significant friction between the object and the finger although keeps the hard finger assumption. Thus, the new condition implies that there is no relative movement of the object on the surface of the finger due to the friction. Finally, Soft Finger with Friction assumes that the finger or object surfaces go through a considerable strain and the contact becomes a surface instead of a point. As a result, the finger exerts a moment of friction on the object normal to the contact surface which does not allow the object to rotate in this direction [22].

Figure 4. Movement constraints of the contact models in grasping. a) Hard Finger with No Friction. b) Hard Finger with Friction. c) Soft Finger with Friction.

Grasps can be classified in precision and power grasps depending on what parts of the hands interact with the object. For a precision grasp the object is grasped with the fingertips as shown in Figure 5 and it is typically used to execute tasks that require fine motor skills such as writing, eating, picking up small objects, using a cell phone, etc. Power grasp requires the whole hand to grasp the object and its function is to move the object from one place to another. There is a big difference in modelling both grasps, power grasp implies that the object and the hand become one single body and there is no movement one relative the other, so only one the combined model is necessary to move the object. In contrast, precision grasp

requires to understand and solve three models for the manipulation of the object at the same time. The first one is the model of the hand itself, the second one corresponds to the contact between the fingertips and the object, and the third one is the model of the object itself. Even though precision grasp model is much more complex for a human or computer to process it allows the hand to perform more complex tasks as previous mentioned.

Figure 5. Planar model of a precision grasp including gravity $\bar{g}$ and a grasp force $\bar{\lambda}$ on second contact point within the friction cone. [22]

When a hand or a gripper completes the grasp an object with no slipping, it had to fulfill two different conditions. The first one is that the exerted contact forces in all the contact points lie within the friction cone which guarantees no slipping of the object relative to the contact point if no external force is applied. The geometry of this cone only depends on the static friction coefficient, the bigger it is the wider will be the aperture of the cone and more forces could lie within the cone. The second condition is that the sum of the exerted forces by the hand or the gripper counteract the sum of the external forces if there is any, thus equilibrium is reached as stated in Newton's first law. By looking at these conditions the first one can be considered the general solution and the second one as the particular solution, so only when they are combined a stable grasp will be performed. Both conditions are typically summarized as one condition known as graspability. Mathematically, graspability exists when the null space of the grasp matrix is nontrivial as shown in equation 1 [23].

$$\text{If } N(\bar{G}) \neq 0 \rightarrow \exists \bar{\lambda} \quad (1)$$

$$\bar{\lambda} = -\bar{G}^+ \bar{g} + N(\bar{G}) \bar{\gamma} \quad (2)$$

Thus, it does exist at least one set of forces $\bar{\lambda}$, defined by equation 2 that can be exerted by the fingers such that grasp is able to complete and sustain. By looking at the definition of lambda, the first term corresponds to the particular solution and the second one to the general solution in accordance with previous definitions. Also, from equation 2 it can be seen that the forces are heavily dependent on the

grasp matrix $\bar{G}$, its mathematical definition is rather complex since it is given by the orientation and position of the fingers and the center of mass of the object as well as the contact model between the fingers and the object. For this reason it is not shown here, but it can be found in reference [22].

9.2 CONTROL OF MYOELECTRIC HANDS

The idea of using electromyographic (EMG) signals to control a prosthetic hand is not recent, it has been explored as early as 1945 [24], for this reason, many research has been conducted on this topic since then. Commercial application of myoelectric prosthetics in the 1960s and 1970s was achieved thanks to advances in size of the battery and motor which allowed to effectively package the components inside the prosthesis [25]. From then till now, the adequate processing of the EMG signal to actuate the hand has been one of the main challenges to address. It is even possible to classify myoelectric prostheses according to the control scheme used to process this signal. Nowadays, the most widely schemes used by commercial prostheses are Direct Control (DC), Proportional Control (PC), Finite State Machine (FSM) and On-Off control. More recent techniques and still on development are Pattern Recognition (PR) and Regression Control (RC).

The simplest scheme is On-Off control and it works by defining a threshold of the EMG signal to choose a direction of actuating the motors. Proportional myoelectric control maps the intensity of the EMG signal, that depends on the level of contractions in the muscles, to the power sent to the motors which is reflected on the speed of grasping. Direct control is similar to PC because it maps the EMG signal to the power sent to the motors. However, unlike PC that maps EMG intensity from all the channels into one single power value sent to each motor, DC maps EMG intensity individually, so each EMG channel controls one motor. Finite State Machine control compares the EMG signal with thresholds that correspond to different states which are mapped into a predefined posture of the hand [26]. It is worth highlighting that FSM is a sequential control because it switches between states while PC and DC control is simultaneous since they are constantly mapping the EMG signal into a movement like a human hand does.

Commercial control schemes, such as PC, DC and FSM, that aim to control multiple degrees of freedom (DOF) or physically represent multiple grasping states are less intuitive to control for the users. The main reason is that the more DOF or states to control or represent in the hand, the more EMG channels and variety of contractions are required. Otherwise, there will not be enough information to differentiate the movements or states. This requirement implies that the user must learn to perform different types of contractions in different muscles that he is not used to perform in his daily life, therefore for the user to master the control the prosthesis requires a lot of training [27][28].

Pattern Recognition takes a different approach to the previous schemes, it computes a set of features from the EMG signal that an algorithm process and classify to a corresponding grasp or wrist movement. For this purpose, it is required a previous training of the algorithm by feeding it many EMG data for each grasp or wrist actions, so it learns to classify the data by itself. Most of the current PR prostheses are based on Machine Learning algorithms such as Linear Discriminant Analysis, Nives-Bayes classifier, k-Nearest-Neighbors, Neural Networks, Support Vector Machines, Convolutional Neural Networks, among others. Recently, developments on PR algorithms have proved the capability of the algorithms to estimate hand posture and grip force [29]. Currently, state of the art has focused on finding the optimal algorithms and set of features that follow the users' intention with the best accuracy [30][31]. Features are divided according to their domain, time or frequency, studies have shown that frequency domain features are not

recommended for PR classification. Instead, domain features are better suited for classifying EMG signals due to their better accuracy when tested [31]. Also, their computational cost is very small which allows microcontrollers to quickly compute them. The following time domain features are the most widely used for PR in hand prosthetics [30]:

$$rms = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (3)$$

$$MAV = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (4)$$

$$ZC = \sum_{i=1}^{N-1} \text{sgn}(x_i, x_{i-1}) \quad (5. a)$$

$$\text{sgn}(x, y) = \begin{cases} 1 & \text{if } (x \cdot y) < 0 \\ -1 & \text{if } (x \cdot y) > 0 \end{cases} \quad (5. b)$$

$$SSC = \sum_{i=2}^{N-1} \text{sgn}(m(x_i), m(x_{i-1})) \quad (6. a)$$

$$m(x_i) = x_i - x_{i-1} \quad (6. b)$$

$$WL = \sum_{i=1}^{N-1} |x_i - x_{i-1}| \quad (7)$$

Nowadays, PR is recommended to be accompanied by a Targeted Muscle Reinnervation (TMR) surgery which allows PR to access more neural information and therefore, a better control of the prosthesis [32]. TMR is a surgical procedure that connects the residual nerves of the amputated limb to muscles that no longer are functional due to amputation, usually the pectoral muscles [33]. Thus, the resultant EMG signals generated by the contraction of the targeted muscles corresponds to attempts of the amputee to move his missing limb which results in an easier and more intuitive control [34]. When compared DC and PR in subjects that have undergone TMR, PR outperformed DC in tests that evaluate the ability of amputees to complete tasks with his prosthesis such as SHAP and Clothespin Relocation Task tests [35]. Applicability of TMR goes from shoulder disarticulation to transradial amputations and it has even shown its potential in control of powered lower limb prostheses [32]. Although, the main objective of TRM is to improve the control of prostheses, it has also been proven as an alternative treatment to phantom limb pain and neuromas which are complications associated with limb amputation [36].

Several PR systems for prosthetic hands have been developed in research environments and currently, there are only two available on the market: *Infinite Biomedical Technologies* [37] and *COAPT* [38]. It is noteworthy that they do not sell prosthesis with PR integrated, instead they sell microcontrollers with the PR system which are adapted to a myoelectric prosthesis. However, its clinical application is still limited due to the failure of the PR algorithms when confronted with real-life conditions. Conditions such as sweating, movement of the EMG acquisition system and mobility of the user. All of them add noise to the

EMG signal which makes it harder for the algorithm to find the patterns that it was trained to recognize [39]. Motion artifact is an additional source of noise generated when the skin-electrode interface is affected by the contraction of the muscles [40]. Recent studies have proposed new PR algorithms to overcome these issues with promising results [41]. Since the nineties the application of PR to control myoelectric prosthesis has been proposed and evaluated in many studies [11]. Nonetheless, most of them develop PR algorithms in off-line conditions, but in order to reduce the gap between research and clinical applications more PR algorithms need to be evaluated in real-time.

For contributing to this purpose, in this study, we tested in real time an Artificial Neural Network (ANN) deployed on a microcontroller. We selected an ANN as the classification algorithm due to its architecture which makes it quick to train it offline and easy to program inside a microcontroller. Static and dynamic real-time classification tests were made with seven volunteer subjects: five able body (AB), one with hand malformation (HM) and one with trans-humeral amputation (THA). We use EMG signals from remnant muscles in the stump, since TMR is not yet available in Colombia. Similar studies have been found [42][43], but the deployment in real time on a microcontroller is still an open problem. The methodology section will explain in detail our steps to achieve this goal.

10 OBJECTIVES

- Develop and validate a pattern recognition algorithm embedded in a microcontroller that can classify in real time the user's intention.
- Design a new actuation mechanism for the thumb that increases the variety of objects the hand can grasp.

11 METHODOLOGY

11.1 GRASPING DESIGN

As the section Previous Work showed, several functional prototypes of robotic hands have been designed, built and tested. In this project, we use the last design by the author in 2016 as starting point [20], and we modified it according to the objectives of the project and some mechanical problems stated in the previous projects. For this reason, only relevant changes in design are shown in this section, for a detailed explanation of hand design reader is invited to check [16], [20].

11.1.1 Palm of the hand

The palm design consisted of three pieces which are joint by bolts as seen in Figure 6, it can also be appreciated that a clearance between the pieces exists as highlighted by the red circle. Although, in the CAD design there is no such assembly clearance, it does exist due to limitations in assembling tolerances caused by low quality 3D printing. Clearance is a problem when closing the hand since the shafts of the fingers do not fit tightly to their mounting holes located between pieces 1 and 2 as seen in Figure 7. Thus, when the motor pulls the pulleys that are mounted in the shafts a force is exerted on them which sometimes remove the shafts from their mounting holes and in turn the fingers from the hand.

Figure 6. Palm pieces of the hand in previous design.

Figure 7. CAD simulated finger shafts clearance

Hence, it was required a new design for the palm to address this problem that also considered the manufacturing limitations. In this manner, the first required modification was to locate the mounting holes only in one piece, instead of being located between pieces. Thus, clearance will only depend on tolerance given by 3D printing and not assembling. Following this idea, piece 1 was divided in two halves, and each half was merged with piece 2 or 3 as seen in Figure 8.

Figure 8. Pieces of the new palm design

The second modification consisted on extend the mounting holes to the lateral faces of the pieces, so the shafts could be easily placed by pushing them or if needed, removed with a magnet. Also, it is required to seal the faces of the holes in case the palm is turned vertically, and gravity pushes out the shafts. A soft silicone made of *Ecoflex* glued to the face of the holes was used for this purpose. Figure 9 shows a physical comparison of the old and the new palm.

Figure 9. Palms of the prosthesis, old (down) and new (up).

11.1.2 Thumb

The previous prototype could only perform power grasps when the hand approached the object with an especial orientation which limited the ability of the hand to grasp objects. Such orientation required the object to be in front of the hand and aligned with the palm, then it should move towards it as seen in Figure 10. Due to the design of the thumb that is always opposed to the palm, when objects approach perpendicularly or from behind the hand the thumb obstructs them from reaching the palm and being grasped.

Figure 10. Possible approaches for grasping in previous prototype. [17]

With the purpose of solving the approach problem and getting the hand to make the two desired grasps, a new thumb was designed. Ideally, a mathematical grasping model of the previous prototype is required to properly understand the dynamics of the grasp, and make design changes based on it. However, its complexity is very high considering that is a 3D-model with multiple fingers and contact points depending on the object. Since modelling is not the core of the project, a different approach to design the new thumb was taken but keeping in mind some insights from grasp modelling. The idea was trying to replicate as much as possible the movement of a human thumb that has 5 DOF [44], as depicted in Figure 11, by actuating only one DOF.

Figure 11. DOF of each phalange in the thumb. AA: Adduction-Abduction. FE: Flexion-Extension. IP: Interphalangeal joint. MCP: Metacarpal joint. CMC: Carpometacarpal joint [44]

For this purpose, the 3 DOF of the joints associated with Flexion-Extension (FE) were turned in 2 DOF as the Interphalangeal (IP) and the Metacarpal (MP) joints were merged in one joint. Also, the 2 Adduction-Abduction DOF of the MCP and the Carpometacarpal joints were reduced to a fixed orientation of the metacarpal and a change in the orientation of the FE DOF between the merged phalange and the metacarpal. AA DOF were neglected because they only add mechanical complexity to the hand and its actuation was not found to be critical when grasping an object. Furthermore, the new resting position of the thumb is on the same plane that the other fingers, so now the objects can approach perpendicularly and from behind the hand when grasped. The actuation of the thumb is done by a DC motor connected to a worm gear which pulls a string that goes through a system of pulleys inside the phalanges. Then, when motor pulls the strings the thumb closes, and when the motor turns in opposite direction strings are released and a couple of tendons made of resin that act as springs return the thumb to its resting position. Finally, given the orientation of the thumb, which is not aligned with the motor, power is not directly transmitted, instead some power would be lost in friction of the string with one of the internal faces of the palm. Consequently, an additional pulley was located inside the palm which considerably reduces the friction force as seen in Figure 12.

Figure 12. Pulleys for actuation of the thumb.

Another advantage of the new design over the old one is that the new thumb is more efficient in terms of time, control and power consumption. When the old thumb grasps it needs to first open to then close because its resting position is closed, as seen in the sequence of the Figure 13. Instead, the new thumb only needs to close the thumb when grasping as shown in Figure 14 sequence. If the same actuation system is used, the new thumb saves time since it does not need to open, therefore the actuation time of the motor as well as the power required would be less. Also, control of the thumb is easier because the motor needs to turn in only one direction instead of two as the old thumb does, so there is no need for the controller to know when the thumb is completely open to change the direction of the motor to close it.

In order to guarantee the structural integrity of the hand, a PID controller will regulate the voltage applied to the motor by constantly measuring the force it exerts to the thumb. This is achieved by indirectly measuring strain of a piece located between the thumb and the motor, so one end of the piece is tied to the string that pulls the pulleys and the other end is tied to the string that the motor pulls. This control system is the same as the previously designed for closing the other four fingers, further details of PID controller and the geometry of the piece can be found in references [17] and [20] respectively.

Figure 13. Grasping sequence of the old thumb

Figure 14. Grasping sequence of the new thumb

11.1.3 Fingertips

The fingers and the palm are 3D printed with a filament made of Polylactic Acid (PLA) which is a material with low friction coefficient and relative high stiffness. Thus, the contact model for the hand is assumed to be Hard-Finger with no friction, so only normal forces can be exerted by the fingers and the palm. This represents a problem when precision grasp is performed because there are only three contact points that exert force in the object. By looking at equilibrium equations 8.a and 8.b, it is clear that the three contact normal forces are not enough to counteract the external forces and its respective momentums.

$$\sum_{i=1}^3 \bar{\lambda}_i + \bar{g} + \sum_{j=1}^M \bar{F}_{ext_j} = 0 \quad (8.a)$$

$$\sum_{i=1}^3 \bar{\lambda}_i \bar{r}_{\lambda_i} + \sum_{j=1}^M \bar{F}_{ext_j} \bar{r}_{F_{ext_j}} = 0 \quad (8.b)$$

Because of this, as a mechanism to increase friction and create a contact surface, fingertips made of resin that resemble the human fingertips were designed. Previous projects have already introduced this feature to the hand by sticking fingertips made of resin to the front of the phalanges and the palm, as seen in Figure 9. Fingertips were manufactured by pouring an *Ecoflex* silicone on 3D printed molds with the desired form, to later be cured inside an oven at 90°C for 10 minutes and stuck to the hand with *Loctite* adhesive joint [20]. However, this methodology was designed for manufacturing the fingertips required for power grasps, but it does not work for precision grasp. A different approach is needed because the fingertips for precision grasp must have an especial geometry to be stuck to the phalanges, since they must be on top of the distal phalanges where the object contacts the finger. Therefore, if a fingertip was made by the old process it would require building a mold of a piece that completely covers the distal phalange which would be difficult to pour as well as remove. Instead, fingertips were made by pouring the top of the phalange inside a recipient full of liquid resin, then cured as the old process does, so a thin layer forms around the phalange, as shown in Figure 15. This process was repeated until the thickness of the layer reached around 2 mm. Finally, the adhesive joint of the frontal fingertips was changed because the fingertips easily detached from phalanges when using *Loctite*. *Sil-Poxy* a silicone adhesive is used as substitute, it works better because it is made specifically for bonding *Ecoflex* to plastics, ceramics and other surfaces.

Figure 15. Curing of the fingertips in an oven.

Figure 16. New fingertips in distal phalanges.

By adding these fingertips, now exists friction force and a contact surface which are enough to counteract the external forces as can be seen in equilibrium equations 9.a and 9.b.

$$\sum_{i=1}^3 \bar{\lambda}_i + \sum_{i=1}^3 \bar{f}_f + \bar{g} + \sum_{j=1}^M \bar{F}_{ext_j} = 0 \quad (9.a)$$

$$\sum_{i=1}^3 \bar{\lambda}_i \bar{r}_{\lambda_i} + \sum_{i=1}^3 \bar{f}_f \bar{r}_{f_f} + \sum_{i=1}^3 \bar{M}_f + \sum_{j=1}^M \bar{F}_{ext_j} \bar{r}_{F_{ext_j}} = 0 \quad (9.b)$$

11.2 PATTERN RECOGNITION

11.2.1 EMG Feature Set

Given the high variability of the EMG signal it is not possible to use the raw data values as input to the PR algorithms. Hence, extracting features of the signal contained in a moving window has been widely used in PR [30][41]. The optimal window width has been found to be between 150-250 milliseconds [45]. Thus, a 200 milliseconds width window was employed, so 40 EMG values are recorded inside. As previously seen, multiple sets of features have been proposed and they can be classified in time and frequency domain features. A comparison between these domains have shown better results for time based features, so frequency based features are avoided in this study [31]. Also, preliminary results found that features that compute the mean value or the sum of the values inside the window reduce the performance of the ANN in real time. Thus, the following set of features that study the variation of the EMG inside the window were proposed:

$$VAR = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (10)$$

$$WL = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \quad (11)$$

$$DASDV = \sqrt{\frac{1}{N-1} (x_{i+1} - x_i)^2} \quad (12)$$

11.2.2 EMG Data Acquisition

AB subjects were asked to execute precision and power grasps as well as no grasping at all, these actions are presented in Figure 1. Given the physical condition of the HM and THA individuals they could not perform the desired grasps, so a different set of actions to acquire the data from was required to train the ANN. These actions were selected as contractions of the remnant muscles that the subject could complete in a natural way and with a constant EMG signal pattern that could be differentiable from the other performed actions. For this purpose, HM and THA volunteers were requested to conduct different combinations of contractions on their remnant muscles with varying intensity until the previous established requirements were fulfilled.

For data acquisition, subjects continuously performed three times each desired action for 15 seconds. A pause between recordings to avoid muscular fatigue was made, also AB subjects were told to execute grasps and contractions in the most natural and comfortable way possible. During these tests, EMG data was recorded at a sampling frequency of 200 Hz generating around 9000 data samples for each grasp which is used to train the ANN. Data is recorded on real-time using the PLX-DAQ extension made for *Arduino* which exports data from the serial monitor directly to *Microsoft Excel* where it is saved as a .csv archive to be later used during training phase. They performed these actions in two static and one dynamic tests. For the static tests, the subjects were instructed to only execute the action and no other movement of the arm. While the first static test EMG data of each grasp is acquired to train the ANN, the second static test is used to evaluate the classification accuracy of the ANN in real time. For the dynamic test, the same actions from previous tests are executed in addition to extension and flexion of the forearm as well as raising and lowering the arm. The objective of the last test is to evaluate the performance of the ANN while grasping and performing typical movements, such as reaching or moving objects, at the same time, as this has been found to be one of the main causes for PR to fail [39].

Figure 17. Actions instructed the subjects to complete. a) No grasp b) Precision grasp c) Power grasp

Two channels of EMG signals are recorded with five surface self-adhesive gelled electrodes, two for each channel and one as reference. For the AB and HM subjects the electrodes pairs are placed on the *extensor carpi radialis* and the *brachioradialis* muscles respectively. These muscles were selected because preliminary tests showed a high electrical activity when these individuals performed the desired grasps. For the THA subject the *biceps brachii* and the *triceps brachii* muscles are selected due to the higher level of amputation and the significant electrical activity they generate. Placement of the electrodes on the selected muscles is shown in Figure 2. We designed and implemented our own custom circuit for acquisition of EMG signals with the purpose to easily integrate it with the ANN in real time. The circuit is a half wave rectifier with low pass filter of 10 Hz that amplifies the signal around 5000 times.

Figure 18. Muscles where EMG signals were recorded. a) Forearm muscles were used for AB and HM subjects. b) Upper arm muscles were used for THA subject.

11.2.3 ANN Deployment

The ANN was deployed on a *Teensy 3.2* microcontroller using an *Arduino* compiler. This board features a 32-bit M4-Cortex ARM processor which grants enough computational power for the network to quickly classify the input signal. The architecture of the ANN was established through a grid search during training

phase to find the best parameters without significantly increasing the complexity of the network. It consists of one hidden layer of ten neurons with the activation function *ReLU* and three output neurons with activation function *Softmax*. It is important to notice that each subject that uses this PR algorithm must have his own customized ANN because EMG signal patterns are different for each person. The process to train and to deploy an ANN for a new user is very simple in fact. First, EMG data is recorded, then, data is pre-processed to be fed into the ANN, it is trained in *Python* by using the *Scikit Learn* machine learning library, next the matrix of weights of the ANN and the normalizing parameters are copied to the PR code in *Arduino*. It is remarkable that after recording the EMG data, deploying a functional real-time PR ANN takes less than 15 minutes.

The microcontroller uses the following algorithm to determine the current grasp intention: every 5 ms, one new measure of EMG is acquired per channel, each one is added to the 200 ms wide windows that collect the EMG values. Then, the required features are calculated and normalized with the mean and variance from the training data. Next, features are fed into the hidden layer of neurons of the ANN to make the forward pass to the output layer. There, the *Softmax* function predicts the probability of each grasp to be the one intended by the user for that instant of time as shown in equation 13. Subsequently, the algorithm finds the mean probability of the previous 40 predictions made by the ANN for each grasp according to equation 14. Finally, the algorithm selects the grasp with the maximum mean probability as the predicted one as it can be seen in equation 15. The final step is made in order to avoid sudden changes of prediction that could be a prediction error for that instant of time. If this mistake was not addressed, the time needed to physically complete the grasp will increase as the prosthesis will suddenly stop executing the desired grasp to start another. This whole process takes less than one millisecond for the *Teensy* microcontroller to complete which shows its outstanding computational power.

$$P(G_i) = \frac{e^{L_i}}{\sum_{i=1}^3 e^{L_i}} \quad (13)$$

$$\bar{P}(G_i) = \frac{\sum_{i=1}^{40} P(G_i)}{40} \quad (14)$$

$$G = \max(\bar{P}(G_0), \bar{P}(G_1), \bar{P}(G_2)) \quad (15)$$

It is noteworthy that the classification delay of the PR algorithm depends on the window length, the sampling period and the number of probabilities used to compute the average in equation 14 as seen in equation 16.

$$Delay_{PR} = LW + SP * N_p \quad (16)$$

Although, for most of the project taking 40 probabilities and sampling at 5 milliseconds showed high accuracies scores of the PR algorithm, the classification delay was found to be above the 300 milliseconds threshold. For this reason, several tests were conducted to find out if by using different parameters the classification delay could be reduced without compromising the classification accuracy.

Lastly, the whole Real-Time PR *Arduino* code with explanatory comments can be consulted on the second annex. Also, the reader is invited to check the folder *Pattern Recognition* of the archives repository where training data, training codes and deployed ANN for each test subject can be found.

11.3 ACTUATION AND ELECTRONIC SYSTEMS

One of the main restrictions when designing a prosthesis is the workspace, it is very limited, and many components must be fitted inside. Thus, making all these components as small as possible is mandatory because the only room for actuation system, force sensors, electronics and battery is located inside the socket of the prosthesis which is designed to be as big as the forearm.

11.3.1 Actuation system

All the fingers of the hand must be closed to complete precision or power grasp, so an actuation system that performs this operation is required. As it was shown before, closing of the thumb is done by pulling a string, for the remaining fingers, it is also used the same system. The main difference with the thumb is that a special sub-actuated mechanism allows to close the four fingers by only pulling one string, it can be appreciated in Figure 19.

Figure 19. Closing mechanism of the fingers. [20]

Hence, if the hand needs to be closed, two actuators are required, one for each string. With the intention of finding an appropriate actuator it is required to first establish our force and velocity requirements when closing the hand. It is complicated to analytically estimate the force required to close each finger due to the tendons between the phalanges that act as springs when they are bend. For this reason, closing force of the thumb and the four fingers was found experimentally. Experimental set-up consisted on tying a support to the pulling string of each finger where masses were added until all the phalanges closed. Later, the masses are weighed with an electronic balance, as seen in Figure 20. The corresponding results are shown on TABLE 1.

Figure 20. Experimental set-up for determining required force for closing the thumb and the fingers.

TABLE 1. Required forces to close the thumb and the fingers.

Finger	Required force [kgf]	Force [N]
Thumb	0.550	56.1
Four fingers	5.700	581.0

Thus, selection of the actuators is based on grasping speed, exerted force, and size requirements. Grasping speed is important because on this depends great part of the delay perceived by the user when he controls the hand. If the hand takes too much time to complete the intended grasp by the user, he will notice that there is a significant gap between his command intention and physical completion of the action. This could lead to a decrease of prosthesis functionality and it has been shown to be one of the main concerns of the users [46]. As a reference of how much time should take the hand to fully close, most advanced commercial prostheses take from 0.3 to 1 seconds [47]. For our study, considering that improving actuation is not the focus of the project, we set 1.5 seconds as closing time, a goal we expect to be easy to reach with commercial actuators. On the other hand, the exerted force by the actuator must be at least equal or higher than the required to close the thumb and the fingers. Otherwise, hand would not close completely and when grasping an object, the exerted force may not be enough to sustain the grasp. It is worth highlighting that a trade-off between force and speed exists, as they are related by the power that the actuator can deliver as seen in equation 17. Then, if too much force is exerted the speed will decrease and vice versa, consequently actuator selection must be done with special attention to this constrain.

$$P = FV \quad (17.a)$$

$$P = T\omega \quad (17.b)$$

By reviewing the requirements and constrains, especially size limitation, the selected actuators are micromotors. First, the dimensions of the selected micromotors are 10 x 12 x 35 mm which fit perfectly to the available space in the socket. Likewise, the micromotors of the selected reference come attached to a gear box, so it is possible to select a gear box configuration that provide the required force and speed. Additionally, in order to pull the strings of the fingers, the motors are coupled to a worm gear system which increases the applied torque and reduces the output speed. The system can pull the string because on top of the gear, there is a cylinder where the string can be tied, in this way, when the gear rotates, the string will wind around the cylinder, as depicted in Figure 21.

Figure 21. Actuation system for closing the prosthetic hand.

Now, to select the appropriate configuration of the gear box the required torque and speed must be found. First, the torque of the thumb and the fingers are different, so the biggest one is used as selection parameter, computation of these torques can be found on equation 18. As the force required to close the fingers is 10 times bigger than the needed for the thumb and the cylinder diameters are expected to be similar, the finger torque is used.

$$T_{req_g} = F_{req_g} \phi_{cyl} \quad (18)$$

Then, to find the required speed, time and distance are required, first, we assumed that both, thumb and fingers, should close at the same time, 1.5 seconds. Second, the length of the string needed to fully close thumb and fingers was measured. Considering that velocity will not be constant during grasping due to required acceleration and deacceleration at the beginning and the end. The required velocity must be over dimensioned to account for this condition because we do not know the dynamic model to close the hand. As a first approach, the required velocity is calculated twice the nominal value as seen in equation 19.

$$V_{req_g} = 2 \frac{L_{pull}}{t_{req_g}} \quad (19)$$

With this information the appropriate gear box and worm-gear parameters can be selected by making some calculations showed from equations 20 to 22. Micromotors fabricant provide information of output torque and angular velocity for each gear box configuration, so the reduction does not need to be considered.

$$T_{out_{wg}} = \frac{N_{wg} T_{out_{gb}}}{N_w} \quad (20)$$

$$\omega_{wg} = \frac{N_w \omega_{out_{gb}}}{N_{wg}} \quad (21)$$

$$V_{out_{wg}} = \frac{\phi_{cyl} \omega_{wg}}{2} \quad (22)$$

Another limitation for selection process is that only a few combinations of possible parameters for the worm gear can work together. Specially, on such a limited space as the available in the socket. These parameters are worm and worm gear diameters, worm gear teeth and number of inputs of the worm. For facilitating this selection process, the worm-gears were designed with the automatic tool of the CAD software *Autodesk Inventor*. The main advantage of this tool is that besides computing the design parameters, it also provides a CAD model of the worm-gear which could be later 3D printed. Table 2 shows several parameter combinations for the actuation system, only the parameters of the highlighted row fulfills the speed and torque requirements, so these are going to be used in the prosthesis.

TABLE 2. Possible parameters for the actuation system.

$T_{out_{gb}}$ [kg*cm]	N_{wg}	N_w	$\omega_{out_{gb}}$ [rpm]	ω_{wg} [rpm]	ϕ_w [mm]	ϕ_{cyl} [mm]	$V_{out_{wg}}$ [mm/s]	V_{req_g} [mm/s]	$T_{out_{wg}}$ [kg*cm]	T_{req_g} [kg*cm]
2.2	36	1	300	8.33	28.8	17	7.42	10	39.6	4.84
1.1	36	1	600	16.7	28.8	17	14.8	10	19.8	4.84
2.2	36	2	300	16.7	28.8	17	14.8	10	19.8	4.84
1.1	36	2	600	33.3	28.8	17	29.7	10	9.9	4.84
2.2	32	1	300	9.4	22.4	12	5.89	10	35.2	3.42
1.1	32	1	600	18.8	22.4	12	11.8	10	17.6	3.42
2.2	30	1	300	10.0	24	18	9.42	10	33.0	5.13
1.1	30	1	600	20.0	24	18	18.8	10	16.5	5.13

11.3.2 Electronic system

Many electronic components such as resistors, diodes, H-Bridges, capacitors, integrated circuits, among others are required for the optimal functioning of the prosthesis. As a first prototype the components required for EMG acquisition were placed together on a protoboard that was used throughout the conducted tests, as depicted in Figure 22.

Figure 22. Protoboard for EMG acquisition and Teensy 3.2.

Nevertheless, protoboard dimensions (22x18x1 cm) are too big for being placed inside the socket, so it was necessary to create a smaller PCB circuit that could fit the available space. Besides including the EMG acquisition components, the new PCB also includes the *Teensy* and the electronic components of the actuation system. Detailed schematic circuit can be consulted on Annex 3, and the editable schematic and PCB layout is found on folder *Electronics* of the archives repository. The PCB dimensions are 8.6x6x1.6 cm which would fit without problem. A top view of the PCB is shown in Figure 23.

Figure 23. PCB designed for the prosthetic hand.

Finally, a lithium polymer (LiPo) battery is selected to power the hand due to its multiple advantages that suit our requirements when compared to other batteries. First, it has a high energy density which reduces battery weight, it also supports larger discharge rates and it has a better safety performance [48]. Nominal voltage of the battery is 7.4 V which is the closest voltage available offered by distributor to 6 V, the recommended voltage for the micromotors. When selecting the amperage, the battery dimensions play

a significant role because the larger the battery the bigger the amperage. For this reason, it is important to find the expected consumed power by the prosthesis during a day such that the battery can provide the required power, but it still can fit inside the socket. As a first approach to dimension the required power throughout the day, the current required during grasping was measured several times. An average current of 152 and 34 milliamperes for the fingers and the thumb respectively was registered. By adding both currents and the *Teensy* consumption (32 mA), and 20 mA from the EMG and force sensor circuits a total of 238 mA would be consumed in an hour. In this way, a 2200 mAh battery was selected its dimensions are 10.4x1.4x3.5 cm which is expected to last around 9 hours of operation if the user continuously performs grasps.

12 RESULTS AND DISCUSSION

12.1 REAL-TIME PATTERN RECOGNITION

Accuracy of the classifier for each subject and grasp were found for three different conditions: ANN training, static tests and dynamic tests, results are shown from Tables 3 to 5.

TABLE 3: Accuracy percentage of training stage of the ANN

Subject	Training results		
	No grasp	Precision grasp	Power grasp
AB 1	100.0	100.0	100.0
AB 2	100.0	95.0	98.3
AB 3	100.0	96.6	95.6
AB 4	100.0	99.0	97.3
AB 5	99.6	99.1	100.0
Average AB	99.9	97.9	98.2
HM	100	96.3	96.7
THA	100	100	100

TABLE 4: Accuracy percentage of static test

Subject	Static tests		
	No grasp	Precision grasp	Power grasp
AB 1	100.0	96.1	98.8
AB 2	100.0	91.7	99.7
AB 3	85.0	94.3	100.0
AB 4	100.0	100.0	98.8
AB 5	89.0	99.5	84.9
Average AB	94.8	96.3	96.4
HM	99.7	98.6	84.2
THA	100	74.9	100

TABLE 5: Accuracy percentage of dynamic tests. ¹

Subject	Dynamic test		
	No grasp	Precision grasp	Power grasp
AB 1	83.3	98.7	96.5
AB 2	96.3	49.2	96.2
AB 3	47.0	52.0	100.0
AB 4	77.1	85.0	99.2
AB 5	72.9	83.6	97.2
Average AB	75.3	73.7	97.8
THA	74.2	52	100

Average offline accuracy scores higher than 97% were achieved for the three conditions predicted by the ANN as seen in Table 3 for all the subjects, even 100% was reached for two of them. It is an excellent result for the classifier considering that the architecture of the network is very simple: only 1 hidden layer with 10 neurons. ANN with more layers and neurons were trained increasing from 1 to 2 % the accuracy for the subjects with worst results. However, increasing the complexity of the network would also have increased the processing time required for the microcontroller to predict the grasp. This could lead to a delay in the sampling rate of the EMG signal if the required time to process the network is higher than the sampling time. Thus, the real-time sampling frequency would not be the same as the one used for acquiring the training data, which could cause the ANN to fail. Further work to find the optimal architecture of the network that accurately classifies the grasps and the microcontroller could process without additional delays is suggested.

Table 4 shows a slight decrease in the static test accuracy of the classification algorithm compared to the training results. Yet, it is possible for the microcontroller to predict with an average accuracy of 95.8%. Which means that the user would be able to reach an object and once there complete and maintain the desired grasp.

Table 5 shows a significant decrease of the ANN accuracy for the conditions of no grasp and precision grasp when compared to the first and second tests. One of the main reasons for these differences is the movement artifact noise added to the EMG signal when the forearm is continuously extended and flexed. For instance, when the arm is flexed the *brachioradialis* muscle, one of the two EMG channels, is contracted and the electrodes move and come closer which is known to add noise to the signal [40]. Also, as the muscle is contracted, it automatically generates a voltage that adds to the current EMG value for that channel which leads to prediction errors. The muscle of the second channel, the *extensor carpi radialis*, is barely affected by this movement, so no relative movement between electrodes, nor muscle contraction is presented. Since the ANN recognizes the patterns of variation of the EMG signal for each grasp, the noise causes an additional variation of the EMG signal which can be confused by the ANN with a grasp with a higher variation pattern. The effect of this type of noise can be verified in the confusion matrix of the AB subject 4 shown in Table 6. This matrix not only shows the accuracy for each grasp but also shows the predicted grasp when the PR algorithm predicts wrongly.

¹ HM subject was not able to participate when dynamic tests were conducted.

TABLE 6: Confusion matrix of subject 4 for his dynamic test.

Intended	Predicted		
	No grasp	Precision grasp	Power grasp
No grasp	77.1	22.9	0
Precision grasp	0	85.0	15.0
Power grasp	0	0.8	99.2

Movements of the EMG acquisition systems were tested during arm lifting and lowering, but there were no significant changes in the signal, and consequently in the predicted grasp. It is worth highlighting that the AB subjects 1 and 4, who felt comfortable exerting bigger forces than the other subjects when grasping, obtained better results in the dynamic tests. A possible reason for this is that despite the chosen set of features, the ANN is less susceptible to the noise previously described when higher mean values of the EMG are achieved.

A remarkable classification accuracy for the power grasp is achieved for all the users in static and dynamic tests. Since this grasp requires bigger forces to be completed, higher EMG values are recorded, around an additional 70 to 100% for all the subjects when compared to precision grasp. Then, the power grasp can be easily differentiated by the ANN in static and dynamic tests since there is a clearer boundary. Even the noise of the dynamic tests is small compared to the EMG values and does not affect the predictions as much as in no grasp and the precision grasp. This result grants that the user would be able to continuously grasp and move any object at its will.

THA subject results were lower than AB and HM subjects for both tests, especially for precision grasp. The main reason for this is that it is difficult for the THA subject to perform and sustain two different types of contractions given his physical condition. Even during experiments, subject stated that he only could easily replicate the contractions for power grasp. Therefore, several contractions intended by the subject to be precision grasps are in fact patterns corresponding to power grasps. This difficulty should not be surprisingly because the user is not familiar with contracting the muscles of his stump in his daily life. Thus, better results could be achieved if the subject undergoes over a muscular strengthening of his stump that allows him to easily perform the appropriate muscle contractions. Also, a deeper training of the ANN to find the muscle contractions with clear pattern boundaries that the subject could replicate without difficulty is required.

Additionally, it is noteworthy that during the tests the more time the individuals had to familiarize with the PR algorithm the better results were obtained. This was more significant for the THA subject as it could be expected.

When compared to similar real-time PR algorithms our ANN shows similar results as shown in Table 7. However, other studies have different conditions like more actions to classify, real-time grasping of the prosthesis or different deploying conditions, so comparison must be done carefully. Yet, the obtained results are very good considering that similar accuracies were achieved even though the computational

power of a computer is much bigger than the one the *Teensy* can provide. Still, it is significant that the required time for a prediction to be made by the microcontroller is less than 1 ms.

TABLE 7: Comparison of results with similar studies. All of them recorded 2 EMG channels.

Research	Possible actions	Average accuracy	Deployment	Physical grasping
Khezri [42]	6	95.3-98	Computer	Yes
Matrone [43]	4	66.7-93.8	Computer	Yes
This study	3	75.4-98.3	Microcontroller	No

Tables 8 to 10 show the accuracy score of the PR algorithm when varying the window length and sampling periods as well as the required processing time for each action. These results prove that the delay could be reduced up to 188 milliseconds without significantly decreasing the accuracy of the algorithm. Also, it is remarkable that the time required to predict each grasp is different, requiring the most time for the power grasp. This is due to the fact that the EMG values recorded for power grasp are the highest, then when the respective features are computed, and the forward pass is performed large numbers will arise and therefore longer computational times.

TABLE 8: Classification accuracies and processing time for a 200 ms window length and a sampling time of 5 ms with different N_p

LW = 200 ms SP = 5 ms	N for average			LW = 200 ms SP = 5 ms	Processing time [ms]		
	10	20	40				
0	1	0.993	1	0	1.3	1.3	1.4
1	0.992	0.987	1	1	1.6	1.6	1.6
2	0.97	0.995	1	2	1.9	1.9	1.9
Delay [ms]	250	300	400	Mean [ms]	1.60	1.60	1.63

TABLE 9: Classification accuracies and processing time for a 150 ms window length and a sampling time of 5 ms with different N_p .

LW = 150 ms SP = 5 ms	N for average			LW = 150 ms SP = 5 ms	Processing time [ms]		
	10	20	40				
0	0.99	1	1	0	1.2	1.3	1.3
1	1	1	1	1	1.4	1.5	1.5
2	0.88	0.636	0.906	2	1.6	1.6	1.7
Delay [ms]	200	250	350	Mean [ms]	1.40	1.47	1.50

TABLE 10: Classification accuracies and processing time for a 150 ms window length and a sampling time of 3.75 ms with different N_p

LW = 150 ms SP = 3.75 ms	N for average			LW = 150 ms SP = 3.75 ms	Processing time [ms]		
	10	20	40				
0	0.991	1	1	0	1.49	1.55	1.68
1	0.983	0.982	1	1	1.91	1.87	1.96
2	1	0.980	1	2	2.18	2.25	2.36
Delay [ms]	187.5	225	300	Mean [ms]	1.86	1.89	2.00

12.2 ROBOTIC ARM

An under-actuated two degrees of freedom robotic hand was designed and built for the prosthesis. It is actuated by two motors each one connected to a worm drive mechanism which pulls the strings of the hand, the first motor actuates the thumb and the second one the remaining fingers. The degree of aperture of the hand is measured indirectly by the deformation of a piece tied between the strings of the second motor and its controlled fingers. Next, a PID controller that uses the deformation as control variable regulates the speed of the motor while closing the hand to guarantee its structural integrity. Then, when the user wants to open the hand, the motors release the strings and tendons made of resin work as springs that apply a restorative force. Figure 24. Available grasps for the prosthesis. a) Precision grasp b) Power grasp shows the two possible grasping configurations of the hand: precision and power grasps. Although, a hand with more available grasps could be designed only two were chosen in order to reduce the number of actuators, material and manufacture costs, physical control complexity and weight. Nevertheless, it is expected that the two selected grasps will allow the amputee to complete precision tasks such as writing and eating since tools like pencils and utensils could be grasped with the precision grasp of the robotic hand. Likewise, force tasks like reaching and carrying objects could be accomplished with the power grasp.

Figure 24. Available grasps for the prosthesis. a) Precision grasp b) Power grasp

With the purpose of verifying whether the hand will be able to move fast enough for the user to not perceive a delay several tests were conducted. These tests aimed to estimate the time required for the motor to reach their nominal speed when closing the hand, this is a condition in which we expect the user to notice the movement of the hand. Five tests for each motor of the hand were conducted, their results are shown in Figure 25. Angular velocity of the motors was measured using an encoder coupled to the extended input shaft of the motor, the *Arduino* code for acquiring this velocity can be found on the *Electronic circuits* folder of the archives repository.

Figure 25. Finger motor velocity for 5 different tests.

Figure 26. Thumb motor velocity for 5 different tests.

In average 70 and 80 milliseconds are needed for the finger and thumb motors respectively to reach their nominal speed. In this manner, by adding 188 milliseconds which is the least possible computational time required to properly classify the EMG signal we would obtain an overall delay of 268 milliseconds. Therefore, we expect the user to not feel any significant delay when he performs an action with the hand since we are below the perceivable 300 milliseconds delay [11].

Finally, additional tests that include the physical action of grasping are required to fully validate the results. These tests will show the total elapsed time for the prosthesis to complete the intended grasp. The last conducted tests tried to evaluate these conditions by controlling the hand through PR and actuating the motors at the same time, but they failed. The main reason for this is that the supplied voltage by the battery to the circuit does not remain constant when the motors are actuated because the motor load varies and so does the required power which is directly linked to the supplied voltage. Therefore, the input voltage to the amplifiers also varies as well as the resultant output EMG signal. Since, the PR classifies the EMG signals by studying its variation, adding the noise variation generated by the motors will mislead the algorithm when classifying. Results showed that no matter the intended grasp by the subject all were classified as power grasp which is not surprising considering that it is the grasp with most variation on the EMG signal. In order to solve this problem, an additional 9V battery dedicated exclusively to power the EMG amplifiers was included, but it did not fix it. We believe that problem lies on the shared ground by the circuits, on account of this it is expected that it could be solved by isolating the actuation and EMG circuits PR. Due to time limitations it was not possible to complete this solution and it is left as future work.

13 COST OF THE HAND

An important consideration for the prosthetic hand prototype is that manufacturing costs remain low, so in a future it could be potentially acquired by low-income amputees or easily build by foundations. The total expenses for manufacturing the hand are shown in Table 11 in Colombian pounds:

TABLE 11: Costs of the prosthetic hand prototype.

Category	Description	Cost [COP]
Electronic	PCB force sensor	9100
	Hall sensor and resistors	2500
	Microcontroller <i>Teensy 3.2</i>	70000
	Global PCB	104900
	Instrumentation Amplifiers	52000
	Surface resistors, capacitors and diodes	15000
	Operational Amplifiers	2000
	Surface H-Bridge	9000
	Surface voltage regulators	3000
	LiPo Battery	72300
	Electrodes set	10000
	Subtotal	349800
Mechanical	Bolts and shafts	10000
	Finger pulleys 3D printing	60000
	Mechanism pulleys 3D printing	74000
	Phalanges and palm 3D printing	352000
	Socket 3D printing	9000
	Polycarbonate sheet	8000
	Micromotor thumb	57300
	Micromotor fingers	57300
	Fingertips, tendons and palm molds 3D printing	105000
	<i>Ecoflex</i> 200 g	40000
	Worm-gears 3D printing	80000
	Sailing strings	10000
	Subtotal	862600
	Total [COP]	1212400
	Total [USD]	377

14 CONCLUSIONS

An Artificial Neural Network was deployed on a microcontroller and tested to follow the intention of able body and amputee subjects in real time. Average classification accuracies of 95% and 81% for static and dynamic tests respectively were reached. Thus, it is possible to conclude that EMG pattern recognition algorithms with low complex architectures can be deployed on microcontrollers with good results. Further tests that rate the performance of the pattern recognition algorithm using the criteria of succeeding in tasks instead of the accuracy of classification are required to fully validate the clinical applicability of the proposed algorithm.

The prosthetic hand prototype can physically perform power and precision grasps thanks to the improvements made on grasping mechanism of the thumb, fingertip design and the assembling of the palms of the hand. Furthermore, it was shown that it is possible to select the parameters of Pattern Recognition so the user should not experience a delay when actuating the hand. However, the combination of Pattern Recognition and physical grasping in real time remains a challenge to overcome.

15 FUTURE WORK

A new action to classify, besides the three proposed should be added to the training phase of the ANN in order to simplify the potential usage of the prosthesis to the user. The current architecture of the PR algorithm requires the user to maintain a constant contraction of the muscles when he wants to exert a grasp for the ANN to correctly predict it. Otherwise, if the subject relaxes his muscles the ANN will understand that the subject wants to open the hand and switch to this class. Then, if the user wants to continuously grasp an object it will eventually cause him fatigue due to the constant contraction which is known to degenerate the EMG signal and consequently the PR algorithm will probably fail [49]. Thus, the new action proposed to predict is the aperture of the hand. In this way, even if the subject stops the contraction of the hand it will not predict it as the intention to open it, it will be understood as a resting position of the user while the hand must sustain the initial classified grasp. Then, the hand will open only when the new contraction trained for this purpose is performed by the user.

To fully validate the clinical applicability of the proposed algorithm and our prosthetic hand prototype the following steps are suggested. A strategy that isolates the actuation and EMG acquisition circuits is required as it was previously stated. In this manner, it will be possible to conduct additional tests that include the physical action of grasping to fully validate the results. These tests will show the total elapsed time for the prosthesis to complete the intended grasp and therefore if a perceivable delay for the user exists. Moreover, the accuracy of the deployed algorithm could decrease because of the increasing number of tasks that the microcontroller needs to do. Some of these are measuring grasping force, updating the PID controller, activating/deactivating the motors that actuate the fingers, among others. Considering that the microcontroller executes tasks in a sequential way, it could lead to a delay of the sampling time of the EMG signal. As previously described this delay may affect the performance of the ANN. Also, further tests with more amputee persons are required to validate the obtained results, especially with trans-radial amputees since they are the final users of the designed prosthetic hand. Those tests should not evaluate the performance of the prosthetic hand by the accuracy score, instead they must assess it by looking into the tasks it can complete, for this purpose, the application of clinical tests is recommended. For instance, the SHAP and Clothespin Relocation Task tests [35] that evaluate the capability and required time to complete hand function activities required in the daily life of a person such as writing a letter, turning over cards, picking up small and big objects, simulated eating, stacking objects, among others.

16 BIBLIOGRAPHY

- [1] Stanford Health Care, "Amputation Causes," 2019. [Online]. Available: <https://stanfordhealthcare.org/medical-conditions/bones-joints-and-muscles/amputation/causes.html>. [Accessed: 07-Nov-2019].
- [2] A. M. Lai, W. D. Stanish, and H. I. Stanish, "The young athlete with physical challenges," *Clin. Sports Med.*, vol. 19, no. 4, pp. 793–819, 2000.
- [3] C. G. Bhuvaneshwar, L. A. Epstein, and T. A. Stern, "Reaction to Amputation," *Prim. Care Companion J. Clin. Psychiatry*, vol. 9, no. 4, pp. 303–308, 2007.
- [4] El País.com.co, "El 26% de los soldados heridos en combate terminan con amputaciones, dice informe," *El País.com.co*, Cali, 09-Jun-2015.
- [5] J. G. Mercado, "Cada 19 horas hay un mutilado en Colombia," *EL TIEMPO*, 11-Aug-2012.
- [6] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Trivison, and R. Brookmeyer, "Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050," *Arch. Phys. Med. Rehabil.*, vol. 89, no. 3, pp. 422–429, 2008.
- [7] Grand View Research Inc., "Robotic Prosthetics Market Analysis By Technology (Microprocessor Controlled, Myoelectric Prosthetics), By Extremity (Lower Body, Upper Body Extremity), By Region, And Segment Forecasts, 2014 - 2025," 2017.
- [8] S. Chu, "I-Limb Ultra Revolution: New Bionic Hand For iPhone," *iDigital Times*, 2013. [Online]. Available: <http://www.idigitaltimes.com/i-limb-ultra-revolution-new-bionic-hand-iphone-video-354836>.
- [9] R. Sacchetti *et al.*, "Literature Review on Needs of Upper Limb Prosthesis Users," *Front. Neurosci.*, vol. 10, no. May, pp. 1–14, 2016.
- [10] B. Peerdeman *et al.*, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *J. Rehabil. Res. Dev.*, vol. 48, no. 6, pp. 719–738, 2011.
- [11] B. Hudgins, P. Parker, and N. S. Robert, "A New Strategy for Multifunction Myoelectric Control," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, 1993.
- [12] Universität Bielefeld, "HUMANOIDs 2018 Workshop 'From Robotic Dexterous Manipulation to Manual Intelligence,'" 2018. [Online]. Available: <https://ni.www.techfak.uni-bielefeld.de/ICHR2018WS/home>. [Accessed: 07-Dec-2019].
- [13] L. A. Durán, "Estudio de la mecánica de sujeción de objetos en la mano humana como requerimiento para el diseño de prótesis," Universidad de los Andes, 2001.
- [14] A. Flower, "Desarrollo del prototipo de una mano robótica prensil," Universidad de los Andes, 2004.
- [15] J. Camargo and A. Polanco, "Mano robótica prensil controlada por señal mecanomiográfica," Universidad de los Andes, 2009.
- [16] J. Sanchez and J. Camargo, "Prótesis funcional de mano con control y retroalimentación de fuerza," Universidad de los Andes, 2014.

- [17] J. F. Velandia and J. Camargo, "Sistema de control retroalimentado para prótesis," Universidad de los Andes, 2015.
- [18] J. R. Arciniegas, "Miembro superior mediante herramientas de comunicación BCI," Universidad de los Andes, 2015.
- [19] C. A. Sánchez and J. Prat, "Método de clasificación de la señal EMG del antebrazo basado en redes neuronales artificiales y su aplicación sobre una prótesis virtual," Universidad de los Andes, 2004.
- [20] M. A. Benítez and J. Camargo, "Prótesis funcional de mano con control y retroalimentación de fuerza," Universidad de los Andes, 2016.
- [21] SIRSLab Siena-Robotics-Systems-Lab. Prattichizzo, *Lecture 1.1: Introduction to the course*. 2018.
- [22] P. Domenico and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, 1st ed., B. Siciliano and O. Khatib, Eds. Berlin: Springer-Verlag Berlin Heidelberg, 2008, pp. 671–700.
- [23] SIRSLab Siena-Robotics-Systems-Lab. Prattichizzo, *Supplementary material 3.2: Graspability*. 2018.
- [24] L. McLean and R. N. Scott, "The Early History of Myoelectric Control of Prosthetic Limbs (1945-1970)," in *Powered Upper Limb Prostheses*, A. Muzumdar, Ed. Berlin: Springer, 2004, pp. 1–15.
- [25] R. R. Caldwell and D. F. Lovely, "Commercial Hardware for the Implementation of Myoelectric Control," in *Powered Upper Limb Prostheses*, A. Muzumdar, Ed. Berlin: Springer, 2004, pp. 55–71.
- [26] P. Geethanjali, "Myoelectric control of prosthetic hands: State-of-the-art review," *Med. Devices Evid. Res.*, vol. 9, pp. 247–255, 2016.
- [27] N. Jiang, H. Rehbaum, I. Vujaklija, B. Graimann, and D. Farina, "Intuitive, online, simultaneous, and proportional myoelectric control over two degrees-of-freedom in upper limb amputees," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 3, pp. 501–510, 2014.
- [28] S. A. Dalley, H. A. Varol, and M. Goldfarb, "A method for the control of multigrasp myoelectric prosthetic hands," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 1, pp. 58–67, 2012.
- [29] Y. Yamanoi, S. Morishita, R. Kato, and H. Yokoi, "Development of myoelectric hand that determines hand posture and estimates grip force simultaneously," *Biomed. Signal Process. Control*, vol. 38, pp. 312–321, 2017.
- [30] J. Camargo and A. Young, "Feature selection and non-linear classifiers: effects on simultaneous motion recognition in upper limb," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. PP, no. c, pp. 1–1, 2019.
- [31] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7420–7431, 2012.
- [32] T. A. Kuiken *et al.*, "Targeted Muscle Reinnervation for the Upper and Lower Extremity," *Tech. Orthop.*, vol. 32, no. 2, pp. 109–116, 2018.
- [33] T. A. Kuiken, G. A. Dumanian, R. D. Lipschutz, and L. A. Miller, "The use of targeted muscle reinnervation for improved myoelectric prosthesis control in a bilateral shoulder disarticulation amputee," *Prosthetics Orthot. Int.*, no. 28, pp. 245–253, 2004.

- [34] L. Resnik, H. H. Huang, A. Winslow, D. L. Crouch, F. Zhang, and N. Wolk, "Evaluation of EMG pattern recognition for upper limb prosthesis control: A case study in comparison with direct myoelectric control," *J. Neuroeng. Rehabil.*, vol. 15, no. 1, 2018.
- [35] L. J. Hargrove, L. A. Miller, K. Turner, and T. A. Kuiken, "Myoelectric Pattern Recognition Outperforms Direct Control for Transhumeral Amputees with Targeted Muscle Reinnervation : A Randomized Clinical Trial," *Sci. Rep.*, no. August, pp. 1–9, 2017.
- [36] G. A. Dumanian *et al.*, "Targeted Muscle Reinnervation Treats Neuroma and Phantom Pain in Major Limb Amputees A Randomized Clinical Trial," vol. 270, no. 2, 2019.
- [37] Infinite Biomedical Technologies, "IBT Products," 2019. [Online]. Available: <https://www.i-biomed.com/products.html>.
- [38] COAPT, "Pattern Recognition 101," 2019. [Online]. Available: <https://www.coaptengineering.com/technology.html>.
- [39] O. W. Samuel *et al.*, "Intelligent EMG Pattern Recognition Control Method for Upper-Limb Multifunctional Prostheses: Advances, Current Challenges, and Future Prospects," *IEEE Access*, vol. 7, pp. 10150–10165, 2019.
- [40] C. J. De Luca, L. Donald Gilmore, M. Kuznetsov, and S. H. Roy, "Filtering the surface EMG signal: Movement artifact and baseline noise contamination," *J. Biomech.*, vol. 43, no. 8, pp. 1573–1579, 2010.
- [41] H. Jahani Fariman, S. A. Ahmad, M. Hamiruce Marhaban, M. Alijan Ghasab, and P. H. Chappell, "Hand movements classification for myoelectric control system using adaptive resonance theory," *Australas. Phys. Eng. Sci. Med.*, vol. 39, no. 1, pp. 85–102, 2016.
- [42] M. Khezri and M. Jahed, "Real-time intelligent pattern recognition algorithm for surface EMG signals," *Biomed. Eng. Online*, vol. 6, pp. 1–12, 2007.
- [43] G. Matrone, C. Cipriani, M. C. Carrozza, and G. Magenes, "Two-channel real-time EMG control of a dexterous hand prosthesis," in *2011 5th International IEEE/EMBS Conference on Neural Engineering, NER 2011*, 2011, pp. 554–557.
- [44] L. Y. Chang and Y. Matsuoka, "A Kinematic Thumb Model for the ACT Hand," no. May, pp. 11–14, 2006.
- [45] L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, "Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 2, pp. 186–192, 2011.
- [46] F. Cordella *et al.*, "Literature review on needs of upper limb prosthesis users," *Front. Neurosci.*, vol. 10, pp. 1–14, 2016.
- [47] I. Vujaklija, D. Farina, and O. Aszmann, "New developments in prosthetic arm systems," *Orthop. Res. Rev.*, vol. Volume 8, pp. 31–39, 2016.
- [48] A&S Power, "The advantages and disadvantages of lithium polymer battery," 2017. [Online]. Available: <https://www.szaspower.com/industry-news/The-advantages-and-disadvantag.html>.
- [49] J. Song, J. Jung, and Z. Bien, "Robust EMG Pattern Recognition to Muscular Fatigue Effect for

Human-Machine Interaction,” in *Mexican International Conference on Artificial Intelligence*, 2006, pp. 1190–1199.

17 ANNEXES

17.1 ARCHIVES REPOSITORY

All the archives used during this project can be found on the following link, they are sorted in the folders CADs, Electronic Circuit and Pattern Recognition.

<https://drive.google.com/open?id=1hVrgffxnaVhRRPnp5rbJdpL7o4S6avwN>

For further questions about the archives mail the author at: ma.benitez834@uniandes.edu.co

17.2 ARDUINO PATTERN RECOGNITION CODE

The following code includes everything required to: read EMG signal, compute features, classify user's intention with ANN, actuate the hand. However, it does not include force control and the hand will continuously grasp until the user changes his intention to no grasping. If he does not change it, structural integrity of the hand may be affected, since the motors will indefinitely continue pulling and something will eventually break, so the code must be used with caution. Integration with force control is the next step, previous work concerning this topic can be found on [17], [20]. This code can be copied and pasted directly on an *Arduino* compiler, it was previously tested, and it should work. It requires three libraries that are not installed by default, but they can be easily found on *Arduino* library manager or its official web page.

```
////////////////////////////////////
// Code for Real-time Pattern Recognition of EMG signal //
// by a Neural Network for Control of Prosthetic Hand //
// Author: Mario Benítez //
// E-mail: ma.benitez834@uniandes.edu.co //
// Year: 2019 //
// GNU General Public License v3.0 //
////////////////////////////////////

#include <BasicLinearAlgebra.h> // Not
#include <Metro.h>
#include <PID_v1.h>

const int pinEMG0 = A0; // Declares the pin used to measure the channel 0 of EMG
const int pinEMG1 = A1; // Declares the pin used to measure the channel 1 of EMG
const int enableF = 18;
const int abrir = 19;
const int cerrar = 20;
const int enableT = 21;
const int abrirT = 22;
const int cerrarT = 23;
int EMG0; // Saves the value of EMG in the channel 0
int EMG1; // Saves the value of EMG in the channel 1
int sensorValue;
int sensorValue2;
const int nce = 3; // Neuronas capa escondida
const int ncs = 3; // Neuronas capa salida
const int in = 3; // Número de entradas a la capa escondida
double dotpr = 0;
double pp;
double mpv; // max probability value
int G = 0;
double rms1 = 0; double sqr1;
```

```

double rms2 = 0; double sqr2;
const int ww = 200; // Wide of the window in ms
const int st = 5; // Sample time of the EMG signal
const int nm = int(ww / st); // Number of values measured within the window
int c = 0; // Counts
int intemg1, intemg2; // Value of the integral over the windows for each channel
int sintemg1, sintemg2; // Value of the square integral over the windows for each
channel
double mean1, mean2; // Value of the mean over the windows for each channel
double var1, var2; // Value of the variance over the windows for each channel
int ssc1, ssc2, m1, m2; // Variables used for computing the slope sign change for each
channel
int wfl1, wfl2; // Variables for the value of the waveform length over the window for
each channel
double aac1, aac2; // Variables for the value of the average amplitude change over the
window for each channel
double dasdv1, dasdv2; // Variables for the difference absolute standard deviation
value over the window for each channel
const int nf = 3; // Number of features used for each channel
const int lfv = nf * 2; // Length of the features vector
double mp0, mp1, mp2, maxmp; // Mean of each grasp in the selected window and maximum
mean probability

char *grasp[] = {"No grasp", "Precision grasp", "Power grasp"};

Metro elapsedTime = Metro(5);
Metro printGrasp = Metro(5);
Metro actuationTime = Metro(200);

using namespace BLA;
// Matrix that saves the weights of the hidden and output layers of neurons
BLA :: Matrix <10, lfv> WHL = { 0.212, -0.171, -0.124, -0.347, -0.471, -0.402, -0.747,
0.062, -0.017, 0.101, 0.082, -0.692, 0.204, 1.939, -1.307, 0.775, 3.936, -1.507, -
0.029, -0.582, 0.303, 0.011, 2.496, -2.173, 1.096, -1.904, -1.485, 0.683, 0.627,
0.197, -0.349, -0.924, -0.631, 0.536, 0.261, 0.115, 0.418, 0.044, 0.276, -0.187, -
0.163, -0.172, 0.119, 0.198, 0.199, -0.031, -0.018, -0.016, -0.067, -0.007, -0.075,
0.008, 0.039, 0.026, -4.013, 0.122, -1.544, -3.198, -6.478, 1.620};
BLA :: Matrix <3, 10> WOL = {0.275, 0.560, 0.313, -3.352, -2.042, -0.320, 0.087,
0.026, 0.032, 6.496, -0.161, 0.283, -4.077, 0.172, 2.092, 1.556, -0.173, -0.031, -
0.076, -6.901, -0.109, -0.838, 3.754, 3.191, -0.040, -1.258, 0.110, 0.024, 0.024,
0.437};
//
// Bias values hidden and output layer
BLA :: Matrix <10, 1> BHL = { -0.6832, 1.7123, 0.4094, 0.4215, -1.3428, 1.5341, -
1.4623, -1.1987, -1.7016, -8.6795}; // Bias values hidden layer
BLA :: Matrix <3, 1> BOL = { -2.8830, 6.5733, -2.1301}; // Bias values output layer
//
BLA :: Matrix <2 * nm, 1> inval = {17, 16, 16, 17, 17, 17, 17, 16, 16, 18, 17, 17, 17,
19, 17, 17, 17, 18, 17, 17, 17, 16, 16, 17, 17, 17, 17, 16, 16, 18, 17, 17, 17, 19,
17, 17, 17, 18, 17, 17, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2}; // input EMG values vector

BLA :: Matrix <lfv, 1> fv = {0, 0, 0, 0, 0, 0}; // features vector
//BLA :: Matrix <22,1> fv = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; // features
vector

BLA :: Matrix <10, 1> inHL; // input vector to the hidden layer of neurons
BLA :: Matrix <3, 1> inOL; // input vector to the output layer of neurons
BLA :: Matrix <10, 1> fol; // features to output layers
BLA :: Matrix <3, 1> onn; // output values of the neural network

// Parameters for standard scaler

```



```

forwardpass();

determinegrasp();
// The following lines are used to print variables if needed
// Serial.print("Agarre: ");
// Serial.print(G);
// Serial.print(" ");
// Serial.println(grasp[G]);
// Serial.print(" EMG0: ");
// Serial.print(EMG0);
// Serial.print(" EMG1: ");
// Serial.println(EMG1);
//      printinval();
//      printfv();
//      printnfv();
//      printprob();
//      printprobmat();

}

void acqemg() {
    EMG0 = analogRead(pinEMG0);
    EMG1 = analogRead(pinEMG1);
}

void arrangedata(int nv1, int nv2) { // Arrange the data to add the current new value
    for (int i = 0; i < nm; i++) {
        if (i < (nm - 1)) {
            inval(i, 0) = inval(i + 1, 0);
            //Serial.println("Cumple condicion");
        }
        else {
            inval(i) = nv1; // Only replaces the new value
        }
    }
    for (int j = nm; j < inval.GetRowCount(); j++) {
        if (j < (inval.GetRowCount() - 1)) {
            inval(j, 0) = inval(j + 1, 0);
        }
        else {
            inval(j) = nv2; // Only replaces the new value
        }
    }
}

void forwardpass() {
    inHL = WHL * nfv + BHL; // Matrix multiplication to find input values to neurons in
hidden layer
    for (int i = 0; i < 10; i++) { // Goes over the hidden layer of neurons
        fol(i, 0) = relu(inHL(i, 0));
    }
    inOL = WOL * fol + BOL; // Matrix multiplication to find input values to neurons in
output layer
    softmax();
}

void normalization () { // Normalizes the features with the mean and variance from
training set
    for (int i = 0; i < fv.GetRowCount(); i++) {
        nfv(i, 0) = (fv(i, 0) - mss(i, 0)) / vss(i, 0);
        //Serial.println(nfv(i,0));
    }
}

```

```

void softmax () { // Softmax function for the output layers
    double sumexp = 0; // Sums the exponentials of the output layer
    //double cexp = 0; // Current neuron exponential output over sum of exponentials
    double mev = 0; // Max exponential value
    for (int i = 0; i < ncs; i++) {
        sumexp += exp(inOL(i, 0)); // Sum of exponentials
    }
    for (int j = 0; j < ncs; j++) {
        cexp(j, 0) = exp(inOL(j, 0)) / sumexp; // Current exponential output neuron
calculation
        // if (cexp(j, 0) > mev) {
        //     mev = cexp(j, 0);
        //     G = j;
        // }
    }
    //Serial.println("}");
    arrangeprobabilities();
}

void arrangeprobabilities() {
    int c1 = 0;
    int c2 = 0;
    for (int i = 0; i < probmat.GetColCount(); i++) {
        if (i < (probmat.GetColCount() - 1)) {
            probmat(0, i) = probmat(0, i + 1);
            probmat(1, i) = probmat(1, i + 1);
            probmat(2, i) = probmat(2, i + 1);
            c1++;
        }
        else {
            probmat(0, i) = cexp(0, 0); // Only replaces the new value
            probmat(1, i) = cexp(1, 0);
            probmat(2, i) = cexp(2, 0);
            c2++;
        }
    }
}

void determinegrasp() {
    mp0 = 0; mp1 = 0; mp2 = 0;
    maxmp = 0;
    for (int i = 0; i < probmat.GetColCount(); i++) {
        mp0 += probmat(0, i);
        mp1 += probmat(1, i);
        mp2 += probmat(2, i);
    }
    mp0 = mp0 / probmat.GetColCount();
    mp1 = mp1 / probmat.GetColCount();
    mp2 = mp2 / probmat.GetColCount();
    //Serial.println(mp0);
    //Serial.println(mp1);
    //Serial.println(mp2);
    if ((mp0 > mp1) && (mp0 > mp2)) {
        G = 0;
        //Serial.println(G);
    }
    else if ((mp1 > mp0) && (mp1 > mp2)) {
        G = 1;
        //Serial.println(G);
    }
    else if ((mp2 > mp0) && (mp2 > mp1)) {
        G = 2;
    }
}

```



```

        //Serial.println(G);
    }
}

////////////////////////////////////
//          Physical actuation of the hand          //
////////////////////////////////////
// It must be first calibrated according to connections to H Bridge and motors
void actuateHand(){
//  if (G == 0) {
//      digitalWrite(enableF, HIGH);
//      digitalWrite(abrir, HIGH);
//      digitalWrite(cerrar, LOW);
//      digitalWrite(enableT, HIGH);
//      digitalWrite(abrirT, HIGH);
//      digitalWrite(cerrarT, LOW);
//  }
    if (G == 1 || G == 0) {
        digitalWrite(enableF, HIGH);
        digitalWrite(abrir, HIGH);
        digitalWrite(cerrar, LOW);
        digitalWrite(enableT, HIGH);
        digitalWrite(abrirT, HIGH);
        digitalWrite(cerrarT, LOW);
    }
    if (G == 2) {
        digitalWrite(enableF, HIGH);
        digitalWrite(abrir, LOW);
        digitalWrite(cerrar, HIGH);
        digitalWrite(enableT, HIGH);
        digitalWrite(abrirT, LOW);
        digitalWrite(cerrarT, HIGH);
    }
}

////////////////////////////////////
//  Activation functions for the Neural Network  //
////////////////////////////////////
double sigmoid(double x) {
    double sigm = 1 / (1 + exp(-x)); // Value of the sigmoid function
    return sigm;
}

double relu(double x) { // Definition of ReLU function
    if (x <= 0) {
        return 0;
    }
    else {
        return x;
    }
}

// For hyperbolic tangent simply use tanh

////////////////////////////////////
//  Functions computing features of EMG signal  //
////////////////////////////////////
void mean() {
    mean1 = 0; // mean of channel 1 vector
    mean2 = 0; // mean of channel 2 vector
    for (int i = 0; i < nm; i++) {
        mean1 += inval(i, 0); // Temporal value of mean 1, it is used to avoid creating an
        extra variable
    }
}

```

```

    }
    for (int j = nm; j < inval.GetRowCount(); j++) {
        mean2 += inval(j, 0); // Temporal value of mean 2
    }
    mean1 = mean1 / nm;
    mean2 = mean2 / nm;
}

void variance () {
    var1 = 0; // Variance of the channel 1
    var2 = 0; // Variance of the channel 2
    for (int i = 0; i < nm; i++) {
        var1 += (inval(i, 0) - mean1) * (inval(i, 0) - mean1); // Temporal value of
variance 1, it is used to avoid creating an extra variable
    }
    for (int j = nm; j < inval.GetRowCount(); j++) {
        var2 += (inval(j, 0) - mean2) * (inval(j, 0) - mean2); // Temporal value of
variance 2
    }
    var1 = var1 / (nm);
    var2 = var2 / (nm);
    fv(0, 0) = var1;
    fv(0 + nf, 0) = var2;
}

void waveformlength() {
    wfl1 = 0;
    wfl2 = 0;
    for (int i = 0; i < (nm - 1); i++) {
        wfl1 += abs(inval(i + 1, 0) - inval(i, 0));
    }
    for (int j = nm; j < (inval.GetRowCount() - 1); j++) {
        wfl2 += abs(inval(j + 1, 0) - inval(j, 0));
    }
    fv(1, 0) = wfl1;
    fv(1 + nf, 0) = wfl2;
}

void dasdv() { //Function that computes the difference absolute standard deviation
value
    dasdv1 = 0;
    dasdv2 = 0;
    for (int i = 0; i < (nm - 1); i++) {
        dasdv1 += ((inval(i + 1, 0) - inval(i, 0)) * (inval(i + 1, 0) - inval(i, 0)));
    }
    for (int j = nm; j < (inval.GetRowCount() - 1); j++) {
        dasdv2 += ((inval(j + 1, 0) - inval(j, 0)) * (inval(j + 1, 0) - inval(j, 0)));
    }
    dasdv1 = sqrt(dasdv1 / (nm - 1));
    dasdv2 = sqrt(dasdv2 / (nm - 1));
    fv(2, 0) = dasdv1;
    fv(2 + nf, 0) = dasdv2;
}

void intemg() { // Function that computes the integral of the values within the window
    intemg1 = 0;
    intemg2 = 0;
    for (int i = 0; i < nm; i++) {
        intemg1 += inval(i, 0);
        intemg2 += inval(nm + i - 1, 0);
    }
}

```

```

void sintemg() { // Function that computes the square integral of the values within
the window
    sintemg1 = 0;
    sintemg2 = 0;
    for (int i = 0; i < nm; i++) {
        intemg1 += (inval(i, 0) * inval(i, 0));
        intemg2 += (inval(nm + i - 1, 0) * inval(nm + i - 1, 0));
    }
}

void computefeatures() { // Function used to compute all the required features
// Only the required features are computed but it is easy to change by uncommenting
mean();
variance();
// slopesignchange();
waveformlength();
// averageamplitudechange();
dasdv();
}

////////////////////////////////////
// Functions used to print important variables //
////////////////////////////////////
void printinval() { // Function used to print optimally the features vector
    Serial.print("Input EMG= ");
    for (int i = 0; i < inval.GetRowCount(); i++) {
        if (i == 0) {
            Serial.print("{ ");
            Serial.print(inval(i, 0), 0);
            Serial.print(", ");
        }
        else if ((i > 0) && (i < (inval.GetRowCount() - 1))) {
            Serial.print(inval(i, 0), 0);
            Serial.print(", ");
        }
        else if (i == (inval.GetRowCount() - 1)) {
            Serial.print(inval(i, 0), 0);
            Serial.println("}");
        }
    }
}

void printfv() { // Function used to print the features vector
    Serial.print("FV= ");
    for (int i = 0; i < fv.GetRowCount(); i++) {
        if (i == 0) {
            Serial.print("{ ");
            Serial.print(fv(i, 0), 4);
            Serial.print(", ");
        }
        else if ((i > 0) && (i < (fv.GetRowCount() - 1))) {
            Serial.print(fv(i, 0), 4);
            Serial.print(", ");
        }
        else if (i == (fv.GetRowCount() - 1)) {
            Serial.print(fv(i, 0), 4);
            Serial.println("}");
        }
    }
}

```

```

void printnfv() { // Function used to print optimally the features vector
    Serial.print("NFV= ");
    for (int i = 0; i < nfv.GetRowCount(); i++) {
        if (i == 0) {
            Serial.print("{ ");
            Serial.print(nfv(i, 0), 4);
            Serial.print(", ");
        }
        else if ((i > 0) && (i < (nfv.GetRowCount() - 1))) {
            Serial.print(nfv(i, 0), 4);
            Serial.print(", ");
        }
        else if (i == (nfv.GetRowCount() - 1)) {
            Serial.print(nfv(i, 0), 4);
            Serial.println("");
        }
    }
}

void printprob() {
    Serial.print("Predicted probability: { ");
    for (int j = 0; j < ncs; j++) {
        Serial.print(cexp(j, 0), 5);
        Serial.print(",");
        if (j == (ncs - 1)) {
            Serial.println("}");
        }
    }
}

void printprobmat() {
    Serial.println("Matrix of probabilities: ");
    Serial.print("[ ");
    for (int i = 0; i < probmat.GetColCount(); i++) {
        Serial.print(probmat(0, i), 3);
        Serial.print(",");
    }
    Serial.println("]");
    Serial.print("[ ");
    for (int j = 0; j < probmat.GetColCount(); j++) {
        Serial.print(probmat(1, j), 3);
        Serial.print(",");
    }
    Serial.println("]");
    Serial.print("[ ");
    for (int k = 0; k < probmat.GetColCount(); k++) {
        Serial.print(probmat(2, k), 3);
        Serial.print(",");
    }
    Serial.println("]");
}

```

17.3 PCB CIRCUIT SCHEMATIC

Caution: Never power the *Teensy 3.2* with a voltage input higher than 6 V.