Design and Implementation of a Prosthesis System Controlled by Electromyographic Signals Means, Characterized with Artificial Neural Networks

David Tinoco-Varela 1,*,†, Jose Amado Ferrer-Varela 2,†, Raúl Dalí Cruz-Morales 1,† and Erick Axel Padilla-García 3,†

1 Engineering Department, Superior Studies Faculty-Cuautitlán, National Autonomous University of Mexico, UNAM, Cuautitlán Izcalli 54714, Mexico
2 Superior Studies Faculty-Cuautitlán, ITSE, National Autonomous University of Mexico, UNAM, Cuautitlán Izcalli 54714, Mexico
3 Academy of Robotics Engineering, Polytechnic University of Atlacomulco, Atlacomulco 50465, Mexico
* Correspondence: dativa19@hotmail.com; Tel.: +52-1-55-39705942
† These authors contributed equally to this work.

Abstract: Around the world many people lose a body member for many reasons, where advances of technology may be useful to help these people to improve the quality of their lives. Then, designing a technologically advanced prosthesis with natural movements is worthy for scientific, commercial, and social reasons. Thus, research of manufacturing, designing, and signal processing may lead up to a low-cost affordable prosthesis. This manuscript presents a low-cost design proposal for an electromyographic electronic system, which is characterized by a neural network based process. Moreover, a hand-type prosthesis is presented and controlled by using the processed electromyographic signals for a required particular use. For this purpose, the user performs several movements by using the healthy-hand to get some electromyographic signals. After that, the obtained signals are processed in a neural network based controller. Once an usable behavior is obtained, an exact replica of controlled motions are adapted for the other hand by using the designed prosthesis. The characterization process of bioelectrical signals was performed by training twenty characteristics obtained from the original raw signal in contrast with other papers in which seven characteristics have been tested on average. The proposed model reached a 95.2% computer test accuracy and 93% accuracy in a real environment experiment. The platform was tested via online and offline, where the best response was obtained in the online execution time.

Keywords: neural networks; EMG signals; prostheses; embedded system

1. Introduction

In 2017, around 57.7 million people in the world suffered the amputation of a limb due to traumatic cases [1]. This means that affected people see the quality of their lives diminished, being of great importance to design prostheses that may be set to each particular situation. Moreover, the design of prostheses which can be controlled and can react in a “natural” way, regarding the required complex motions of mechanisms, is of paramount interest.

For this purpose, the use of bioelectrical electromyographic (EMG) signals is proposed as element of control. EMG signals are generated by every muscular contraction. Then, the user could control the prosthesis in the same way that would react if had the body member.

From above, development of bio mechatronic prosthesis with EMG signals is presented in this manuscript, being a way to get each movement and particular bioelectrical representation that can be characterized to obtain signals that can be used as commands.
To treat EMG signals, an adequate amplitude range was obtained by amplifying them after having acquired values directly through an electrode. Thus, the signal amplitude varies from µV to a low range of mV (lower than 10 mV), and according to Trimedica [2], the usable energy of a signal is bounded in the frequency range of 0–500 Hz with dominant energy in the range of 50–150 Hz, both ranges above the electrical noise level. To obtain a filtered sensory system, a converter was proposed for the raw signals (without processing) to adequate them into suitable signals for characterization and classification, i.e., subject the acquired signal to a cleaning process.

It is worth to mention that the main drawback of a system that characterizes EMG signals is the sensitivity of the usable signal profiles which depend on different factors, such as the muscle contraction intensity, the electrodes position, and on the skin type and condition, additionally, measurements are affected by the collection data system conditions, the electrode quality, sensor design, environment noise, etc. Moreover, instrumentation, software, sensors, and specific devices to face this set of factors would make it hard to develop a low-cost experimental platform. To face with this set of factors, the system can be characterized by obtaining as much as possible information of signal samples that belongs to the training dataset of the artificial neural network (ANN).

In literature, EMG signals have been used in several experiments, for instance, EMG signals have been used to detect emotion’s reaction signals as in [2], to create musical sounds as in [3] and to develop smart clothes [4]. Also, EMG signals have been used to get control parameters of robotic arms [5], to operate MP3 players [6], and mobile phones [7], to characterize gestures [8], and for applications in human-machine interfaces [9].

Gesture or manual movement classification is one of the goals of this work since it seeks to develop a system that adequately classifies the EMG signals of the user, and to use that classification to control a hand prosthesis. Some different types of classification that can be reviewed from the literature, and some of those works related to this proposal are the following. The most common are by using support vector machines [10] random forest [11], k-nearest neighbors (k-NN) [12], wavelet transform [13], decision trees [14], fuzzy logic [15,16] and artificial neural networks [8], among others.

Artificial neural networks have become very popular for characterization and classification, and they have been used in different projects to classify hand movements to design prostheses, where a fluid drives the “natural” movement of the human body [17–19]. Regarding EMG signal characterization, raw signals and extraction of features can be used as input elements [20], where pattern classification success depends on the features selection. Vectors composed by characteristics of the original EMG signal have been used as inputs to train neural networks, where the main used values are the mean absolute value (MAV), variance (VAR), standard deviation (SD), zero crossing (ZC), and the root mean square (RMS). However, there are some other characteristics of the EMG signals to be used, such as waveform length (WL), mean absolute value slope (MAVS), mean frequency (MNF), median frequency (MDF), slope sign change (SSC), root mean square (RMS), wavelet packet transform (WPT) coefficients, auto regression (AR) coefficients, Willson amplitude (WA), fast Fourier transform (FFT) coefficients, cepstrum coefficients (CC), and short-time Fourier transform (STFT) coefficients. In this manuscript, 20 characteristics of the EMG signals are extracted and used as inputs for an ANN, to enrich the characterization and classification process, unlike other works that typically use just 7 characteristics.

It is worth to mention that all the design, filtering, and sensor characterization have been carried out by using a low-cost system, inferring that no expensive devices are used for its implementation. Then, this proposal leads to an affordable solution, compared to other proposals in literature, where the production costs are more expensive. Moreover, our system reached a 95.2% of precision in the computer evaluation and 93% in the real test. Additionally, it was tested for online and offline experiments, where the best system response was obtained by online execution time.

This manuscript is organized as follows, in Section 2, the preliminaries of the project realization are presented; in Section 3, the proposed design of the system is stated, where the
characteristics extraction of the raw EMG signal, the input vector of used neural network, and the signal process development are described. In Section 4, experiments and data acquisition are shown, analyzing the obtained results by showing a comparison with other similar proposals. Finally, in Section 5 the conclusions to analyze the improvements obtained and future work are stated.

2. Preliminaries

This section presents the preliminaries for better understanding of the proposed prosthesis. Focusing on artificial neural networks and electronic filters.

2.1. Filters

Electronic filters allows passing or suppressing a certain set of frequencies, such as low-pass (signals with high frequencies are suppressed), high-pass (signals with low frequencies are suppressed) and band-pass (allows the passage of a set of frequencies, delimited by a lower and an upper limit). For instance, in Figure 1 a low-pass filter was used to decrease the presented noise of the signal, allowing only the one belonging to the required signal.

![Figure 1. Using a low-pass filter to remove noise from a sinusoidal signal.](image)

Depending on the type of components used, there are passive and active filters. The passive ones only contain resistive, capacitive, and inductive elements, while the active ones contain active components, that are typically operated and tuned by using operational amplifiers. Regarding active filters, three of the most used active filters are: Butterworth, Bessel, and Chebyshev. If an ideal low-pass filter existed, all the active filters would eliminate signals above the cutoff frequency, and perfectly pass signals below the cutoff frequency. However, in a real situation, many trade-offs should be considered to get the optimum performance for a given application.

In this work, the Butterworth filter is used to obtain the required signals of the user body, since they are easy to handle at many real time applications [21,22]. Then, these obtained signals are used on the artificial neural network.

2.2. Artificial Neural Networks

The simplest model of a biological neuron is the well-known perceptron, introduced by Frank Rosenblatt, which emulates a neuron behavior by reproducing its main characteristics. In this model, the information reception through different inputs $P_1, P_2, \ldots, P_n$, as they were dendrites to stimulates the neuron nucleus, and the final value depends on an active response (1) or inactive response (0) of the neuron. Thus, the mathematical representation of the neuron activation function can be defined as

$$S = \sum_{i=1}^{n} P_i W_i - b$$

\[
\text{Output} = \begin{cases} 
0 & \text{if } S \leq 0 \\
1 & \text{if } S \geq 1 
\end{cases}
\]

where $W_1, \ldots, W_n$ are the synaptic weights, and $b$ is the bias of the neuron. A schematic representation of perceptron can be seen in Figure 2.
In this work, a supervised learning was implemented for the training process of the neuron, i.e., the output values for a set of input data are known by the designer. Then, if the expected output \( z \) is different from the obtained output \( y \), an error \( e = z - y \neq 0 \) is obtained and the synaptic weights of the neuron are adjusted as result of this error. Such that, \( W'_i = W_i + \gamma eX_i \) where \( \gamma \) is the learning factor, and \( X_i \) are the inputs.

This adjustment is performed while \( e \neq 0 \). When \( e = 0 \) in all cases, a trained neuron is obtained. From this neuron, that is the minimum unit, more complex configurations can be built by interconnecting them, giving way to artificial neural network architectures.

An ANN is a computational scheme that emulates the learning process of a biological brain. Also, the ANN are connectionist systems, i.e., interconnection of different independent neurons is performed, until a network of neurons communicated by means of synaptic weights is arrange, a neural network general scheme can be seen in Figure 3, which basically consists of the input layer, the output layer, and the hidden layers. The neurons number in the input layer is conditioned by the number of elements in the input vector with which was trained. The number of neurons in the output layer corresponds to the classes or classification sets number that the problem contains. Finally, the neurons number in the hidden layers will depend on the nature and complexity of the addressed problem.

Before an ANN can classify a behavior or structure, it must be trained, which means that a user must told it what to learn. To carry out this training, it is necessary to enter many phenomena or problem known cases to be learned, the network will adjust itself in such way that it will learn that for a set of inputs, there is a certain output. The cases with which a network is trained are stored in a dataset, and the ANN will obtain all the information from such dataset.

These schemes have been used mainly in classification problems [23], object detection [24], image processing [25], and to estimate sustainable transport demand [26], controllers [27,28], melanoma detection [29], among others.
3. Materials and Methods

This section presents the entire design and development process of the proposed system, from the electronic to the artificial learning part and its implementation. The neural network that controls the prosthesis needs to be fed by a database to work and be able to learn to replicate the desired movements. In this case, the signals database is captured from the user who performs different muscle contractions movements. For this reason, it is necessary to design an electromyographic signal sensor that obtain and adapt such signals to be processed and stored, in order to be used for training the neural network. The design of this sensor is presented.

3.1. Electromyographic Sensor Design

To design an electronic device capable of sensing small electromagnetic variations, the obtained EMG signals were amplified, filtered, and conditioned to be used by an Arduino board.

To replicate the movements for a hand, different types of sensing methods can be used for this purpose, such as cyber gloves [30], cameras [31], and electromyography [32]. The electromyography was the used method in this paper. Despite the available EMG sensors on the market, such as Myoware and Oymotion, in this proposal the own EMG sensor is made to obtain a raw signal for the Arduino development board. To design this sensor, five stages are considered to condition the signal: pre-amplification, signal filtering (low-pass and high-pass filters), final amplification, and offset. Each stage in the signal cleaning process is described as follows.

3.2. Signals Conditioning

3.2.1. Pre-Amplification

This stage is the first contact with the signal and the subsequent processes. Thus, instrumentation amplifiers are used, and their working conditions are design by using ideal amplifiers conditions, as infinity input impedance, zero impedance output, high ratio rejection, and highly gain voltage, all this to prevent added noise to the amplified signal. An amplifier of this type can be purchased commercially, however, an own amplification system is designed to adapt it to the needs of the project, this is done by using LM833 operational amplifiers.

The general amplification circuit, based on the EMG sensor, is divided in two amplification stages, due to, if all the amplification is carried out in a single stage, saturation would be generated in the output, and consequently an EMG signal loss.

Figure 4 shows the amplification circuit designed with LM833, which serves as a signal pre-amplifier.

![Schematic of the amplifier circuit with LM833.](image-url)
Where $V_3$ is the output signal given by,

$$V_3 = (V_2 - V_1) \left( \frac{R_4}{R_3} \right) \left( 2 \left( \frac{R_2}{R_1} \right) + 1 \right)$$  \hspace{1cm} (1)$$

where $V_2$ and $V_1$ are the circuit’s input voltages, $R_1, R_2, R_3, R_4$ are resistances for amplification setting.

From Equation (1), it is possible to notice that the gain depends on $R_2$, but the variation depends on $R_1$, if $R_3 = R_4$ is considered.

### 3.2.2. Third Order Low-Pass Butterworth Filter

A raw signal is susceptible to noise and contamination, so it should be conditioned before using it, for that reason a pre-amplification stage is proposed where a Butterworth active filter is used to filtering and discarding frequencies and values that do not give a representative value on the input signal. Later, the following must be considered: higher order filters, use a higher number of operational amplifiers, resistors, and capacitors. Considering this, it is decided to design a third order filter, which needs less components, and has an adequate cutoff frequency response. A cutoff frequency of 500 Hz is established to eliminate noise caused by electronic equipment and electronic components of the sensor itself. Figure 5 shows the designed filter. A simulation of this filter is performed by using $R = 12 \text{ kΩ}$, $C_2 = 47 \text{ µF}$, $C_3 = 4.7 \text{ µF}$, and a LM833N amplifier where is obtained that the real cutoff frequency is at 471 Hz, since the EMG frequency interest range is between 0 to 500 Hz and the greatest amount of information is between 50 Hz and 150 Hz, this cutoff frequency is acceptable. This filter has a 61 dB drop per decade.

![Figure 5. Schematic diagram of a 3rd order low-pass Butterworth filter.](image)

### 3.2.3. Third Order High Pass Filter

A high pass filter should be designed for better signal matching and handling. Then, a cutoff frequency of 20 Hz is established, this cutoff frequency is selected to eliminate the noise caused by the movement of the cable that connects the electrode and the skin. The electrical signals from both noise sources have most of their energy in the frequency range from 0 to 20 Hz. The designed circuit is shown in Figure 6. Simulation of this filter is carried out to obtain that the real cutoff frequency is 22.7 Hz, the filter has a drop of $-60.3 \text{ dB}$ per decade.

![Figure 6. 3rd order high-pass butterworth filter.](image)
3.2.4. Non-Inverting Amplifier

In the second amplification stage, a non-inverting amplifier is made, which is designed to deliver a minimum gain of 1 and a maximum gain of 30 according to the $R_2$ potentiometer value of 20 KΩ, the amplifier design is shown in Figure 7.

![Non-inverting amplifier schematic.](image)

**Figure 7.** Non-inverting amplifier schematic.

3.2.5. Offset (Non-Inverting Summing Amplifier)

Once the signals are amplified and filtered, an EMG signal is available, which consists of positive and negative voltages, but is not compatible with the chosen microcontroller device, Arduino. To adapt this signal and avoid microcontroller damage, one last conditioning signal stage is necessary, consisting of a non-inverting summing amplifier. An offset circuit that equalize all the resistances, raises the potentiometer’s voltage, this allows placing the signal at 2.5 V as a new reference, obtaining an EMG signal ready to be processed by the Arduino, the designed offset circuit from 0 V to 9 V is shown in Figure 8.

![Designed offset circuit.](image)

**Figure 8.** Designed offset circuit.

3.3. Neural Network Design

Considering that the amplified signals are obtained and adapted, they are stored within a dataset and used for training the neural network, however, for a better neural network learning and classification, is necessary to extract certain signal characteristics so that the ANN training be adequate and functional. Due to this, using different techniques and algorithms, 20 main characteristics are extracted from each user signals obtained. When these characteristics are obtained, a representative vector is created, and stored in a dataset training and therefore feeds the neural network input.

**Extraction of Features**

The extraction of characteristics is recommended since they are chosen according to the needs of training and operation of the neural network, for this reason the characteristics extracted from the raw EMG signals and the way to obtain them are described. The 20 characteristics extracted from the raw EMG signal to create the input vectors, in order to
train the neural network. Table 1 shows the selected characteristics to be extracted, then a brief explanation of the way in which each of them are calculated is shown in this section.

Table 1. EMG signal characteristics, for a hybrid vector of 20 times, frequency, and time-frequency.

| 20 Characteristics Vector |
|---------------------------|
| Mean absolute value       |
| Zero crossing             |
| Slope sign change         |
| Waveform length           |
| Autoregressive model AR (5): first, second, and third coefficients |
| Fast Fourier Transform: Energy, periodogram, and mean power |
| Short time Fourier transform, Window 1: Energy, spectrogram, and mean power |
| Short time Fourier transform, Window 2: Energy, spectrogram, and mean power |
| Short time Fourier transform, Window 3: Energy, spectrogram, and mean power |
| Wavelet transform approximation: Coefficient 1 and variance; coefficient 2 and variance; coefficient 3 and variance; coefficient 4 and variance; coefficient 5, variance |

Mean absolute value (MAV): It is the sum of each segment $\bar{x}_i$ of $N$ samples of a signal. Where $x_k$ is the $k$-th sample in segment $i$, and $I$ is the total number of segments.

$$\bar{x}_i = \frac{1}{N} \sum_{k=1}^{N} |x_k|, \quad i = 1, 2, \ldots, I$$

Zero crossing (ZC): It is the number of times the waveform crosses zero. To obtain the result, a comparison is made between two consecutive sections of the signal $x_k$ and $x_{k+1}$, such that:

$$x_k > 0 \text{ and } x_{k+1} < 0$$
$$x_k < 0 \text{ and } x_{k+1} > 0$$

A work related to this concept is presented in [33], which is focusing on EMG signals.

Slope sign change (SSC): Similar to ZC, but now measuring sign changes between signal elements.

$$x_k > x_{(k-1)} \text{ and } x_k > x_{(k+1)}$$
$$x_k < x_{(k-1)} \text{ and } x_k < x_{(k+1)}$$

Waveform length (WL): It is the sum of the differences between values of consecutive samples of the EMG signal, up to the last sample. It is obtained as follows.

$$wl = \sum_{k=1}^{N} |\Delta x_k|$$

where $x_k = x_k - x_{(k-1)}$.

P-order auto regressive model AR(P): It is used for a random process representation of a model in which the main variable depends on its past values and it is possible to obtain the coefficients of the auto-correlation function $a_i$, $P$ gives the order, and $e(k)$ is the element error.

$$x(k) = \sum_{i=1}^{P} a_i x(k-i) + e(k)$$

Fast Fourier Transform (FFT): Fourier transform allows a signal to be broken down into its individual components, in such way, it is possible to obtain information that would otherwise be hidden itself within the signal. It is defined as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi}{N} kn}, \quad k = 0, \ldots, N - 1$$
where $x_n$ is the sampled signal for specific discrete time. A fixed period time sampling was taken to obtain the estimate of the power spectral density, and thus have the characteristics of energy and average power.

**Short time Fourier transform (STFT):** It separates the original signal into sections, each of these sections is analyzed by means of the Fourier transform, and each section characteristics are extracted independently. This analysis can be obtained as follows:

$$STFT(x(t)) = \frac{1}{\sqrt{a}} \psi \left( \frac{t-b}{a} \right)$$

where $w(t)$ is the window function.

Spectrogram consists of acquiring a certain number of samples through a temporary window, with a specific size. The frequency content is determined for these samples, and they are represented in a three-dimensional graph. Performing this same operation on the total length of the signal. This process gives us frequency information and the variation of energy. For this study, the short Fourier transform of 3-time windows of the EMG signal is obtained, using a standard rectangular window, from each, two characteristics are extracted, the energy and the average power, elements that compose the input vector of the ANN to train the system.

**Discrete Wavelet Transform (DWT):** Wavelet continuous transform (CWT), express a continuous $x(t)$ signal as an expansion of coefficients proportional to the product between the signal $x(t)$ and the scaled and displaced version of the mother wavelet $\psi_{(a,b)}$.

$$CWT(a,b) = \int_{-\infty}^{\infty} x(t) \psi_{(a,b)}(t) dt \quad a,b \in R$$

$$\psi_{(a,b)} = \frac{1}{\sqrt{a}} \psi \left( \frac{t-b}{a} \right)$$

where $a$ is the scale parameter to compress or expand and $b$ is the translation parameter that moves the Wavelet. Discrete wavelet transform (DWT) is obtained by discretizing the CWT. The values $a = 2^{-j}$ and $b = k2^{-j}$ are usually used to discretize for parameters $a$ and $b$, hence the mother Wavelet takes the form:

$$\psi_{(j,k)}(t) = 2^{j/2} \psi(2^j t - k) \quad (2)$$

Thus, the Wavelet $\psi(t)$ with a scale function $\phi(t)$, is associated, which allows us to approximate any function $x(t)$ with these functions, by means of the expression:

$$x(t) = \sum_k \sum_j c_{(j,k)} \phi(t) + \sum_k \sum_j d_{(j,k)} \psi(t) \quad j,k \in Z \quad (3)$$

where $c_{(j,k)}$ are the scale or approximation coefficients, and $d_{(j,k)}$ represents the Wavelet or detail coefficients of the original signal $x(t)$, with respect to the scale function $\phi(t)$ and the Wavelet function $\psi(t)$.

The behavior of the DWT is like a “filter banks”. Such filters correspond to a low-pass and a high-pass filter, when the original signal passing through them, the output coefficients $c_{(j,k)}$ and $d_{(j,k)}$ respectively, are obtained. A decomposition of the signal can be obtained at different levels by passing the scale coefficients obtained from the previous filtering by an identical pair of filters, thus obtaining the coefficients of the next level.

The Wavelet selected for this article is Daubechies D5, which will provide us with five approximation coefficients and five detail coefficients, for each of the 5 approximation coefficients the variance calculation has been performed.

### 3.4. Experimental Platform

The experimental platform is shown in Figure 9, and it consists of an Arduino Nano board, this card has an ATmega328 microcontroller of AVR architecture. The analog to digital converter has a resolution of 10 bits, with a conversion range voltage of 0 V to
5 V. Pins 5, 6, 9, 10, and 11 are used for servomotor control, and analog pin 2 for EMG signal acquisition.

![Experimental platform for a bionic hand prosthesis.](image1)

**Figure 9.** Experimental platform for a bionic-hand prosthesis.

To obtain the EMG signals, low-cost self-adhesive silicone gel electrodes are used, which are reusable, 10 of these worth approximately 7.5 USD. A complete cleaning must be carried out in the area to be placed to keep the skin clean each time a user wears it.

In Figure 10 the electronic circuit schematic built for the control of the hand is presented. Additionally, a complete amplification circuit for sending filtered and conditioned signals to the microprocessor is performed by using a perforated phenolic prototype board.

![Electrical diagram connection of experimental platform for human-hand emulation.](image2)

**Figure 10.** Electrical diagram connection of experimental platform for human-hand emulation.

4. Results

This section shows the implementation and tests carried out with the EMG signals sensor. To carry out the tests, Simulink/MATLAB software version 2020, Mathworks, (MATLAB Central, Mexico) was used together with the Toolbox Simulink Support Package for Arduino Hardware, with this it is possible to carry out oscilloscope simulations and correctly adjust the signals from electrodes. This is done by adjusting the three potentiometers with the resistive values described in the design of the high-pass and low-pass filters.

4.1. Implementation and Testing of EMG Sensor with MATLAB

Users must place the electrodes on the arm to obtain the system signals and send them to the computer that are displayed through an oscilloscope made in Simulink/MATLAB.

Figure 11 shows the used position to extract the EMG signals, where the V1 and V2 electrodes must be placed in the muscle in which the EMG signals are acquired, and the GND electrode in a nearby area. It is important to emphasize that it must be an area where the muscle is not working. The proper location for the electrodes is between the innervation zone (or motor unit) and the tendinous insertion, or belly of the muscle [34].
Once the EMG signals are obtained, they are displayed on the oscilloscope designed in Simulink, adjustments are made to adapt the values and the signal can be acquired properly. The final values are as follows, for a gain of 31 in the preamp stage, potentiometer is adjusted to 27.64 Ω and for a gain of 21 in the final amplification, potentiometer is adjusted to 13.6 KΩ, giving a total gain of 735. It is also important to notice that the final applied offset is 1.5 V with a potentiometer resistance of 3.3 KΩ. Then, the gains are tuned, tests are carried out that consist of moving the fingers of the hand with small force and the acquired signals can be observed in MATLAB oscilloscope as shown in Figure 12.

![Figure 11. Location of the electrodes on the arm.](image1.png)

4.2. Data Acquisition

Once the signal is acquired satisfactorily, the integrated microcontroller development environment (IDE) is used to create a program that process the muscles signals sent by the electrodes. The acquired signal is discretized by the acquisition system and it is important to guarantee that the signals characteristics can be reconstructed, for that reason, the Nyquist theorem is used, this theorem indicates that the sampling frequency must be at least twice than the frequency of the signal that will be reconstructed. The frequency limits that the designed EMG sensor circuit admits go from 22.7 Hz to 472 Hz, where

\[
f_{\text{sample}} \geq 2f_{\text{signal}} = 2 \times 472 \text{ Hz} = 944 \text{ Hz}
\]

The sampling velocity of created program is 1 kHz, this signal is acquired each time it is requested by MATLAB, the number of samples sent is 60,000 in 60 s, these are sent through the serial port. Each sample generates 2 bytes, so a total of 120,000 bytes are sent over 60 s.

![Figure 12. EMG signal acquired by the developed sensor and displayed in the virtual oscilloscope designed in Simulink. The signal is traced in real time.](image2.png)
A Simulink/MATLAB interface was created, which is responsible for receiving the sampled signal by the Arduino board, this interface stores the information in matrix structures. Each hand movement generates a particular signal, these are stored in a labeled matrix to identify which movement was generated, these are: closed hand, grip, index, middle and ring finger movement. These movements are shown in Figure 13.

![Figure 13. Positions of the hand and fingers that are characterized.](image)

We generated a database or “dataset” with the signals of each of the described movements, 50 samples of each movement are collected, with 500 sampling points each. Once obtained, sequences of the same movement are generated and saved as a single vector. Figure 14 shows the obtained sequences.

![Figure 14. Samples sequences related to each hand movement to be characterized, obtained from the sensor and plotted by Simulink.](image)

A windowing process is applied to each of these signals (Figure 15), which consists of delimiting only the area of interest and cutting the linear part that does not have information of the muscular effort to be analyzed. In this way, all 500 samples are saved, and the storage is optimized by reducing each signal to only the part of interest. Such signals are stored in arrays for later analysis within a neural network.

Finally, dataset is generated with each structured vector from the extracted characteristics of each movement signal, previously defined and the input array for the neural network is created. In total, 250 signals are obtained, which are divided into 250 vectors of 20 characteristics each, therefore, there is a $20 \times 250$ matrix.
4.3. Training of the Neural Network

This section describes the neural network process training with the dataset obtained and described in Section 4.2.

For the hand movements characterization, a supervised neural network for pattern recognition is created, with a trainlim training function. The network has a 20 neurons input layer, automatically adjusted by the algorithm according to the dataset, there is a 15 neurons hidden layer defined in the program, with a sigmoid activation function and a 5 neurons output layer with a gradient descent function. In Figure 16, the network architecture is shown, as well as its training parameters.

The original dataset was divided in two subsets: 60% for training data and 40% for validation, discarding a test subset. From these sets, the generated confusion matrix is observed in Figure 17. It can be seen that in the characterization of these 5 movements, an adequate response was obtained, which achieves a success rate of 95.2%.
Figure 17. Confusion matrix related to the training and validation of the characterized EMG signals.

Notice that before select the training function (TF) mentioned previously, it was necessary to compare and choose from different TF. It is difficult to obtain the best neural network architecture to solve a problem, however, tools like MATLAB has efficient training function which can be used to determine the best suited to our problem. In this case, five TF were analyzed in order to determine which one would be used on the prosthesis. The different TF were implemented in a feedforward neural network.

To make the selection process of the best TF, a database with the hand movements signals has been used, the dataset has been divided into 60% training, 20% validation and 20% testing. The executed TF’s and the data obtained from each are shown in Table 2.

Table 2. Training characteristics and validation of results to be used in the prosthesis.

| TF from MATLAB | Algorithm                  | Best Performance of Mean Square Error | Epoch of the Best Validated Performance | Accuracy |
|----------------|----------------------------|---------------------------------------|----------------------------------------|----------|
| trainlm        | Levenberg-Marquardt       | 0.010917                              | 12                                     | 95.6%    |
|                | BFGS                      |                                       |                                        |          |
| trainbfg       | Quasi-Newton Polak-Ribiére | 0.04938                               | 157                                    | 78.8%    |
| traincgp       | Conjugate Gradient        | 0.035416                              | 27                                     | 88.8%    |
| trainoss       | One Step Secant Variable  | 0.029079                              | 48                                     | 91.2%    |
| traingdx       | Learning Rate Backpropaga-| 0.043967                              | 26                                     | 90.8%    |

Table 2 shown that the Levenberg-Marquardt algorithm corresponding to the trainlm TF, presents the best precision results, being the algorithm used in this approach.

Figure 18 shows the performance metrics for each of the TFs, and Figure 19 shows the confusion matrices for each of these TFs.
Figure 18. Evaluated TF’s performance where blue line is the training, green line the validation, red line the test, and dot line the best validation performance for 5 case studies: (a) Trainlm, (b) Trainbfg, (c) Traingcp, (d) Trainoss and (e) Traingdx.

Figure 19. Evaluated TF’s confusion matrix. (a) Trainlm, (b) Trainbfg, (c) Traingcp, (d) Trainoss and (e) Traingdx.
4.4. Emulation of a Human Hand

With the obtained signals and the hand movements characterized by the previously described neural network, a human hand prosthesis was created to verify the operation of the ANN that emulates the hand movements. In order to physically check the movements generated by the neural network, a prosthesis is developed that consists of 5 servomotors, 5 fingers with three phalanges each, which mimic the fingers and hand movements. The neural network sends the necessary signals to the servomotors to replicate the indicated movements, this developed prosthesis is shown in Figure 20.

![Figure 20. Bionic-hand prosthesis which emulates human-hand movements.](image)

Real environment operational test is carried out by performing 90 random movements, taken from the previously described in Section 4.2. Where 84 movements were correctly replicated and 6 were incorrectly performed, obtaining a 93.3% of assertiveness and 6.6% of error. This is according with the efficiency percentages obtained in the training and validation.

The EMG signal characteristics extracting, and neural network training methods are compared with others to analyze and demonstrate that the used method has an adequate response and is at least as effective as the most efficient known methods. This comparison is shown in Table 3.

This proposal shown similar efficiency to the others, despite having more features per input vector, the efficiency has not been significantly increased. This is interesting, since it could be deduce that there is a maximum number of characteristics that significantly influence the way in which neural networks learn the behavior of EMG signals, a problem that will be investigated in a future work. Notice that the acquisition and experimentation system is a low-cost designed, even so compared to other prototypes and authors that used more expensive EMG sensors our present better response. This fact can lead to the conclusion, that hardware may not have such an influence on the EMG signals characterization success and that the efficiency rate depends only on the neural network training and learning.
Table 3. Comparison of acquisition systems.

| Reference | Acquisition Device | Hand Gestures | Algorithm Characterization | Extracted Features or Training Data | Acquisition Frequency | Success Average Rate |
|-----------|--------------------|---------------|-----------------------------|-------------------------------------|-----------------------|----------------------|
| [35]      | Biopac MP100 data acquisition system | Left, right, up, down | Back propagation with Levenberg-Marquardt | 7 features: MAV, RMS, VAR, SD, ZC, SSC, WT, and db2 with 4 levels | 1 kHz | Between 88.4% and 89.2% |
| [36]      | °                  | Close hand, flex hand, extend the hand and fine grip | Convolutional NN Images | Images | 8 kHz | 83.7% ± 13.5%, 71.2% ± 20.2%, 82.6% ± 13.9%, and 74.6% ± 15% for each movement |
| [37]      | MYO armband        | Fist, Wave In, wave Out, Fingers Spread, and Double Tap | feedforward ANN | 5 features: MAV, SSC, WL, RMS, and Hjorth parameter (HP) | ° | 98.7% |
| [38]      | BIOPAC-MP100C data acquisition system | Left, Right, Up and Down | Levenberg-Marquardt and scaled conjugate gradient based back-propagation | 7 features: Moving Average, RMS, VAR, SD, ZC, SSC and WL | 1 kHz | 88.4% |
| [39]      | MyoWare Muscle Sensor (AT-04-001) | Cylindrical Grasp, Supination (Twist Left), Pronation (Twist Right), Resting Hand and Open Hand | Back propagation used Levenberg-Marquardt | 4 features: MAV, median, WL and RMS | ° | 80% |
| [40]      | MyoWare Muscle Sensor | Rock, scissors, paper, one, three, four, good, okay, finger gun, and rest | Multilayer perceptron, support vector machine (SVM), random forest (RF), and a logistic regression (LR). | 6 features: RMS, VAR, MAV, SSC, ZC, and WL | ° | Maximum 94%, depending on the methodology |
| [41]      | labVIEW and Biokit | Open and closing, thumb flexion, index flexion, middle and ring finger flexion. | Levenberg-Marquardt | 6 features: MAV, RMS, MNF, ZC, SSC and SD. | 1 kHz | ° |
| Our project | Own | Closed hand, grip, index, middle, ring | Back propagation with Levenberg-Marquardt | 20 features: explained above. | 1 kHz | 95.2% and 93% in real test. |

4.5. Online Execution

Online system testing is performed by using the MATLAB Online tool in which the same offline code is used and the same data is loaded in order to verify the behavior of our system in an online environment. Which will be used as a node of the Internet of things in a future work.

To obtain the system’s efficiency in the online environment, about 70 random tests are carried out, where a better behavior in temporal response is obtained. Table 4 shows online and offline tests, the time showed on it, is a mean value.

It can be seen in Table 4 that the online execution time is better than offline execution, in this case, the algorithm was executed in a computer with 6 GB DDR RAM memory and an AMD A8-6410 processor running at 2.4 GHz, which can be an important factor to obtain such results. Notice that the online execution process controls the designed prosthesis in the same way as the offline execution, which is an important fact related to our system.
Table 4. Identification of 9 random manual gestures executed online and offline comparison.

| Movement               | Online Execution Time [s] | Offline Execution Time [s] |
|------------------------|---------------------------|----------------------------|
| Trial 1: Index         | 1.095399                  | 2.352354                   |
| Trial 2: Grip          | 0.874528                  | 2.300595                   |
| Trial 3: Closed hand   | 1.015186                  | 2.208586                   |
| Trial 4: Ring finger   | 1.043519                  | 2.342219                   |
| Trial 5: Middle        | 0.953655                  | 2.816689                   |
| Trial 6: Ring finger   | 1.008239                  | 2.234787                   |
| Trial 7: Closed hand   | 1.069840                  | 2.185594                   |
| Trial 8: Grip          | 1.003771                  | 2.324529                   |
| Trial 9: Index         | 0.893527                  | 2.065108                   |

5. Conclusions

This manuscript presents an entire EMG signal acquisition system design, which considers the integration of sensor, filtering, and information transmission systems, selection of the adequate algorithm, also shows the EMG signals characterization by neural networks and the online and offline prosthetic hand control using such characterizations.

The neural network training is carried out with vectors that contain 20 extracted features from the raw EMG signals, achieving a 95.2% accuracy in practices and validation, and 93% accuracy in a real environment experiment. Although there is a high response reliability percentage and more features were used for training, the response has not been improved significantly compared to other proposals that only use 6 or 7 features in their training vectors. Which leads to question; there exist an optimal evaluating features number to obtain high efficiency without sacrificing time and processing resources?

The system was evaluated with online and offline executions using MATLAB toolbox, in order to have a homogeneous behavior, observing that the offline execution had better execution time, but both of them control adequately the designed prosthesis.

The development of a low-cost hardware system goal is fulfilled, the performance shown is better than some high-cost professional hardware systems. Which is an advantage for prosthesis implementation and encourage their use in poor or developing countries.

Figure 9 shows the entire system developed, i.e., the electronic circuit, the prosthesis and the power supply.

Future Work

The proposed solution can be extended to explore the sensor channels number, which it may has a direct influence with the extracted features number.

The extracted features number optimization is proposed, and the creation of a prosthesis with materials that not only replicate the hand movements, but also be used by a patient in the future as a real hand, this include a better prosthetic hand development, made with additive manufacturing.

There are different methods to do a system recognition, as shown in [42], that would be worthy to test and compare the performance of those methods versus the proposed in this manuscript.

Author Contributions: Conceptualization, J.A.F.-V. and D.T.-V.; methodology, J.A.F.-V. and D.T.-V.; software, J.A.F.-V.; validation, J.A.F.-V., E.A.P.-G. and R.D.C.-M.; formal analysis, J.A.F.-V. and D.T.-V.; investigation, J.A.F.-V. and D.T.-V.; writing—original draft preparation, J.A.F.-V.; writing—review and editing, D.T.-V., E.A.P.-G. and R.D.C.-M.; funding acquisition, D.T.-V. and R.D.C.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by UNAM PAPIME PE103322, UNAM PAPIME PE105822 and FESC CI2254.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Supporting data is available via request to authors.
Conflicts of Interest: The authors declare no conflict of interest.

References

1. McDonald, C.L.; Westcott-McCoy, S.; Weaver, M.R.; Haagsma, J.; Kartin, D. Global prevalence of traumatic non-fatal limb amputation. Prosthetics Orthot. Int. 2021, 45, 105–114. [CrossRef] [PubMed]
2. Wagner, J.; Kim, J.; André, E. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In Proceedings of the 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, 6 July 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 940–943.
3. Dubost, G.; Tanaka, A. A wireless, network-based biosensor interface for music. In Proceedings of the 2002 International Computer Music Conference (ICMC 2002), Gothenburg, Sweden, 16–21 September 2002.
4. Tröster, G. The agenda of wearable healthcare. Yearb. Med. Inform. 2005, 14, 125–138. [CrossRef]
5. Crawford, B.; Miller, K.; Shenoy, P.; Rao, R. Real-time classification of electromyographic signals for robotic control. In Proceedings of the 20th National Conference on Artificial Intelligence, Online, 9 July 2005; Volume 2, pp. 523–528.
6. Fistre, J.; Tanaka, A. RealTime EMG Gesture Recognition for Consumer Electronics Device Control; Sony CSL Paris Open House 10/2002: Paris, France, 2002.
7. Costanza, E.; Perdomo, A.; Inverso, S.A.; Allen, R. EMG as a subtle input interface for mobile computing. In Proceedings of the International Conference on Mobile Human-Computer Interaction, Online, 26 June–1 July 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 426–430.
8. Khanna, A.; Muthukumaraswamy, S.A. Cost-effective system for the classification of muscular intent using surface electromyography and artificial neural networks. In Proceedings of the 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 2, pp. 366–371.
9. Song, M.S.; Kang, S.G.; Lee, K.T.; Kim, J. Wireless, skin-mountable EMG sensor for human–machine interface application. Micromachines 2019, 10, 879. [CrossRef]
10. Toledo-Pérez, D.C.; Rodríguez-Reséndiz, J.; Gómez-Loeno, R.A.; Jauregui-Correa, J. Support vector machine-based EMG signal classification techniques: A review. Appl. Sci. 2019, 9, 4402. [CrossRef]
11. Sangjun, O.; Mallipeddi, R.; Lee, M. Real Time Hand Gesture Recognition Using Random Forest and Linear Discriminant Analysis. In Proceedings of the 3rd International Conference on Human-Agent Interaction, Daegu, Korea, 21–24 October 2015; pp. 279–282.
12. Benalcázar, M.E.; Jaramillo, A.G.; Zea, A.; Páez, A.; Andaluz, V.H. Hand gesture recognition using machine learning and the Myo armband. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece, 28 August–2 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1040–1044.
13. Haider, I.; Shahbaz, M.; Abdullah, M.; Nazim, M. Feature extraction for identification of extension and flexion movement of wrist using EMG signals. In Proceedings of the 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 3–6 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 792–795.
14. Joshi, A.; Monnier, C.; Betke, M.; Sclaroff, S. Comparing random forest approaches to segmenting and classifying gestures. Image Vis. Comput. 2017, 58, 86–95. [CrossRef]
15. Khezri, M.; Jahan, M.; Sadati, N. Neuro-fuzzy surface EMG pattern recognition for multifunctional hand prostheses control. In Proceedings of the 2007 IEEE International Symposium on Industrial Electronics, Vigo, Spain, 4–7 June 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 269–274.
16. Chan, F.H.; Yang, Y.S.; Lam, F.; Zhang, Y.T.; Parker, P.A. Fuzzy EMG classification for prosthetic control. IEEE Trans. Rehabil. Eng. 2000, 8, 305–311. [CrossRef]
17. Gini, G.; Arvetti, M.; Somlai, I.; Folgheraier, M. Acquisition and analysis of EMG signals to recognize multiple hand movements for prosthetic applications. Appl. Biosci. Biomech. 2012, 9, 145–155. [CrossRef]
18. Raurale, S. Acquisition of EMG signals to recognize multiple hand gestures for prosthetic robotic hand—A review. Int. J. Curr. Eng. Technol. 2014, 4, 65–70. [CrossRef]
19. Owies, R.J.; Rihani, R.; Alkhawaja, A. ANN-based EMG classification for myoelectric control. Int. J. Med. Eng. Inform. 2014, 6, 365–380. [CrossRef]
20. Hudgins, B.; Parker, P.; Scott, R.N. A new strategy for multifunction myoelectric control. IEEE Trans. Biomed. Eng. 1993, 40, 82–94. [CrossRef]
21. Rastogi, N.; Mehra, R. Analysis of butterworth and chebyshev filters for ECG denoising using wavelets. IOSR J. Electron. Commun. Eng. (IOSR-JECE) 2013, 6, 37–44. [CrossRef]
22. Jadhav, S.A.; Misal, S.B.; Mishra, A.; Murugkar, A. Designing of stepped impedance Butterworth and Chebyshev filters for wireless communication. In Proceedings of the 2017 IEEE Applied Electromagnetics Conference (AEMC), Aurangabad, India, 19–22 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–2.
23. Li, J.; Zhang, Z.; Zhu, X.; Zhao, Y.; Ma, Y.; Zang, J.; Li, B.; Cao, X.; Xue, C. Automatic Classification Framework of Tongue Feature Based on Convolutional Neural Networks. Micromachines 2022, 13, 501. [CrossRef] [PubMed]
24. Fayyaz, M.A.B.; Johnson, C. Object detection at level crossing using deep learning. Micromachines 2020, 11, 1055. [CrossRef] [PubMed]
25. Egmont-Petersen, M.; de Ridder, D.; Handels, H. Image processing with neural networks—A review. Pattern Recognit. 2002, 35, 2279–2301. [CrossRef]

26. Barreda-Luna, A.A.; Rodríguez-Reséndiz, J.; Flores Rangel, A.; Rodríguez-Abreo, O. Neural Network and Spatial Model to Estimate Sustainable Transport Demand in an Extensive Metropolitan Area. Sustainability 2022, 14, 4872. [CrossRef]

27. Cabrera-Rufino, M.A.; Ramos-Arregui, J.M.; Rodríguez-Reséndiz, J.; Gorrostieta-Hurtado, E.; Aceves-Fernandez, M.A. Implementation of ANN-Based Auto-Adjustable for a Pneumatic Servo System Embedded on FPGA. Micromachines 2022, 13, 890. [CrossRef]

28. Sanchez-Reyes, L.M.; Rodriguez-Resendiz, J.; Salazar-Colores, S.; Avecilla-Ramírez, G.N.; Pérez-Soto, G.I. A High-accuracy mathematical morphology and multilayer perceptron-based approach for melanoma detection. Appl. Sci. 2020, 10, 1098. [CrossRef]

29. Gao, W.; Fang, G.; Zhao, D.; Chen, Y. A Chinese sign language recognition system based on SOFM/SRN/HMM. Pattern Recognit. 2004, 37, 2389–2402. [CrossRef]

30. Liu, K.; Kehtarnavaz, N. Real-time robust vision-based hand gesture recognition using stereo images. J. Real-Time Image Process. 2016, 11, 201–209. [CrossRef]

31. Benalcázar, M.E.; Motoche, C.; Zea, J.A.; Jaramillo, A.G.; Anchundia, C.E.; Zambrano, P.; Segura, M.; Palacios, F.B.; Pérez, M. Real-time hand gesture recognition using the Myo armband and muscle activity detection. In Proceedings of the 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), Salinas, Ecuador, 16–20 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.

32. Toledo-Perez, D.; Rodriguez-Reséndiz, J.; Gómez-Loenzo, R.A. A study of computing zero crossing methods and an improved proposal for EMG signals. IEEE Access 2020, 8, 8783–8790. [CrossRef]

33. Jamal, M.Z. Signal acquisition using surface EMG and circuit design considerations for robotic prosthesis. Comput. Intell. Electromyogr. —Anal. —Perspect. Curr. Appl. Future Challenges 2012, 18, 427–448.

34. Ahsan, M.R.; Ibrahimy, M.I.; Khalifa, O.O. Electromyography (EMG) signal based hand gesture recognition using artificial neural network (ANN). In Proceedings of the 2011 4th International Conference on Mechatronics (ICOM), Kuala Lumpur, Malaysia, 17–19 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1–6.

35. Abu, M.A.; Rosleesham, S.; Suboh, M.Z.; Yid, M.S.M.; Kornain, Z.; Jamaluddin, N.F. Classification of EMG signal for multiple hand gestures based on an artificial neural network. Sensors 2021, 22, 225. [CrossRef]

36. Poliakov, A.; Pakhalin, V. Fast User Activity Phase Recognition for the Safety of Transfemoral Prosthesis Control. Facta Univ. Ser. Mech. Eng. 2020, 18, 135–151. [CrossRef]