Intelligent Recognition System of Myoelectric Signals of Human Hand Movement

Sergio A. Vicario Vazquez[1(✉)], Outmane Oubram[1], and Bassam Ali[2]

[1] Facultad de Ciencias Químicas e Ingeniería, Universidad Autónoma del Estado de Morelos, Av. Universidad 1001 Col. Chamilpa, 62209 Cuernavaca, Mexico
vicario17_@hotmail.com, oubram@uaem.mx

[2] Facultad de Ingeniería, Universidad Autónoma de Yucatán, Av. Industrias no Contaminantes por Periférico Norte, Apdo. Postal 150, Cordemex, Mérida, Yucatán, Mexico
baali@correo.uady.mx

Abstract. This paper presents the recognition of myoelectric signal's algorithm design thru artificial neural network architecture, for the manufacture of a prototype of human hand prosthesis. At the beginning of the project the myoelectric sensor was designed to help capture the signals that correspond to the movement of each finger of a human hand. A database was generated with captured myoelectric signals, which was used for the training of artificial neural networks (ANN), obtaining the weights and bias. The performance of the architecture was evaluated with statistical criteria for the validation of ANN, comparing between simulated data and experimental data. It was found, that the best architecture in this project has 7 neurons in the hidden layer, one in the output layer and 96% correlation coefficient, this architecture is the number 7 in Table 1 which contains a performance report learning algorithm of the different architectures proposed.

Keywords: Myoelectric signals · Intelligent recognition · Prosthesis
Artificial neural network · Myoelectric sensor

1 Introduction

Prosthesis is an artifact developed in order to improve the quality of life of people who by some accident lost some limb of their body. Through the advancement of technology in the last decades prostheses have becoming more advanced. The human body is able to generate electrical signals in its muscles, these signals are known as bioelectric signals. The bioelectrical signals are divided into different types depending on the origin of this signal. The myoelectric signals are those generated by the contraction of some muscle of any extremity such as the arms and legs, can be measured with a suitable equipment and thus use the information that these provide us in the design of prostheses. The work of Parimal performs a micro controlled system, based on the microcontroller 68HC11 (Parimal et al. 1988). The system amplifies the myoelectric signals, filters them, digitizes them and the control algorithm decides the movement of a robot of two degrees of freedom. Another similar work in which, unlike the previous

C. Brito-Loeza and A. Espinosa-Romero (Eds.): ISICS 2018, CCIS 820, pp. 97–112, 2018.
https://doi.org/10.1007/978-3-319-76261-6_8

one, a myoelectric prosthesis is activated (Romero et al. 2001). In systems oriented to assistance in industrial environment, is the work of López, di Sciasco, Orosco, Ledesma, Echenique, and Valentinuzzi (López et al. 2006), in this paper, a myoelectric sensor was designed, which is responsible for capturing, amplifying and filtering the signals generated by the contraction of the muscles responsible for the movement of each finger. The recognition of the myoelectric signals is performed through neural network architecture, allowing a more reliable result. This architecture could be developed and validated thanks to the Matlab tool, to later generate a function which represents the operation of said neural network, obtaining a signal recognition algorithm and entering it in the Arduino Mega 2560 development board in order to obtain a Prototype of prosthesis of a human hand shown in Fig. 1.

Fig. 1. Prototype of prosthesis of a human hand.

2 Experimental System

The development methodology of this research project begins with the design of a myoelectric sensor for the acquisition of the signals. A database was created from the myoelectric signals generated by the contraction of the muscles responsible for the movements of each finger of a human hand.

2.1 Myoelectric Sensor Design

The parameters of the myoelectric signals to be captured are:

- The frequency of the myoelectric signal is 50 and 200 Hz.

- Sleep state: is the state where the excitable cells maintain a potential −90 mV < V < −50 mV.
- Active state: in this state excitable cells have an electrical potential between −55 mV < V < 30 mV.

A myoelectric sensor was designed with the following characteristics:

- Pre-amplification
- High pass filter
- Low pass filter
- Amplification and signal coupling
- Selection of cables for electromyography (EMG)

An instrumentation amplifier with biomedical applications AD620 was used, which requires less components to adjust the gain and presents a rather high CMRR (Common Mode Rejection Ratio) value at low frequencies. To calculate the components of this stage, the formula recommended in the AD620 circuit data sheet was used:

$$G = 1 + \frac{49.9\,\mathrm{K}\Omega}{R_G}$$

In this case it is desired to calculate R_G, since it was postulated as $G = 250$, therefore the result is the following:

$$R_G = \frac{49.4\,\mathrm{K}\Omega}{250 - 1} = 198.88\,\Omega;\ \frac{R_G}{2} = \frac{200\,\Omega}{2} = 100\,\Omega$$

High Pass Filter: This was calculated for a cut-off frequency of 5 Hz with a commercial capacitor of $C = 0.1$ µF.

$$F_C = \frac{1}{R * 2\pi * c};\ R = \frac{1}{2\pi * (0.1\,\mu F) * (5\,\mathrm{Hz})} = 318.3\,k\Omega$$

$$R_6 = 610\,k\Omega // 610\,k\Omega = 305\,k\Omega$$

Inverter Amplifier: The purpose of this amplifier that is connected between the R_6 of the AD620 is to provide patient safety also helps to maintain a stability in the signals obtained in the EMG.

$$\frac{V_0}{V_1} = -\frac{R_F}{R_1};\ \frac{V_0}{V_1} = -\frac{470\,\mathrm{K}\Omega}{12\,\mathrm{K}\Omega} = 39.16$$

Low pass filter: A Butterworth low pass filter of order 2 was designed with a Sallen-key topology, with a quality factor of 0.71, gain 1, and with a cutoff frequency of 980 Hz. Obtaining the following results.

$$R_1 = 22\,\mathrm{K\Omega};\, R_2 = 1.2\,\mathrm{K\Omega};\, C_1 = 10\,\mathrm{nF};\, C_2 = 100\,\mathrm{nF}$$

Once the amplification problem of the small signals has been solved, it is necessary to design additional stages for conditioning the system, which are aimed at filtering and cleaning the signal being collected and amplifying the filtered signal. The circuits used are shown in Fig. 2.

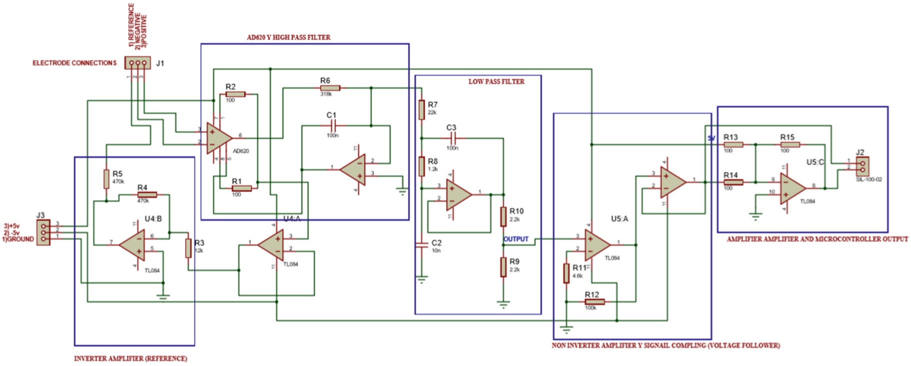

Fig. 2. Sensor of myoelectric signals with all its stages.

2.2 Acquisition

When the myoelectric sensor was obtained, the EMG was performed, in other words, to capture signals from the muscles responsible for the movement of the fingers on the hand. The EMG consists of placing surface electrodes on the required extremity as shown in Fig. 3, these sensors are connected by wires to the input of the myoelectric sensor, and the output of the myoelectric sensor is connected to the Arduino Mega 2560 which sends the data to a PC.

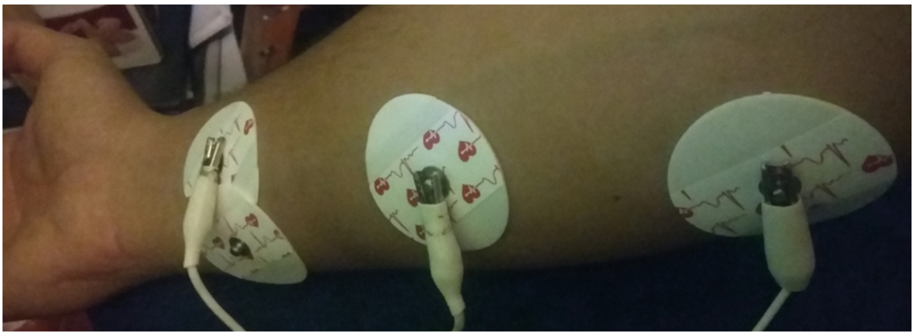

Fig. 3. Placement of the surface electrodes for the realization of EMG.

With the help of the Matlab tool the signals obtained were plotted and saved, in order generate a database necessary for the training of the neural network. In this work the movements on which they worked were the contraction of each of the fingers and the full fist at the same time, namely, the movement of opening and closing the complete hand. The signals are shown in the following Figs. 4, 5, 6, 7, 8 and 9.

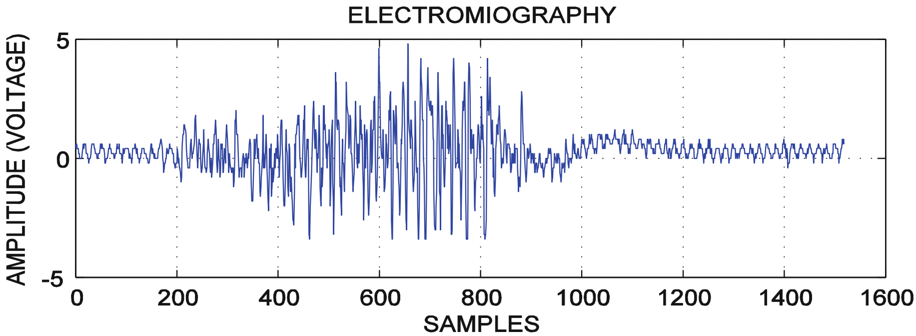

Fig. 4. Myoelectric signal produced by the full fist.

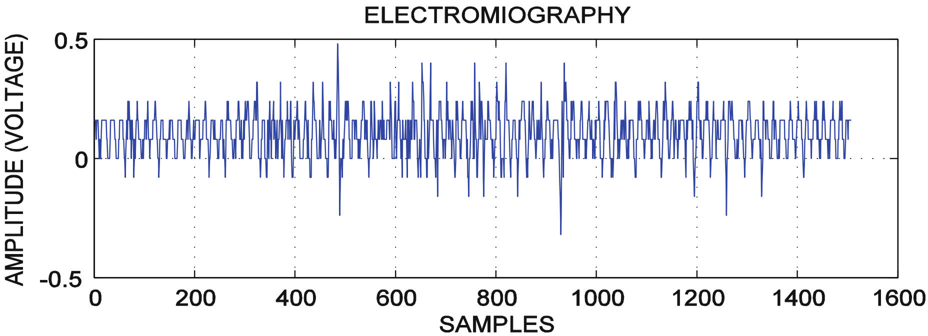

Fig. 5. Myoelectric signal produced by the index finger

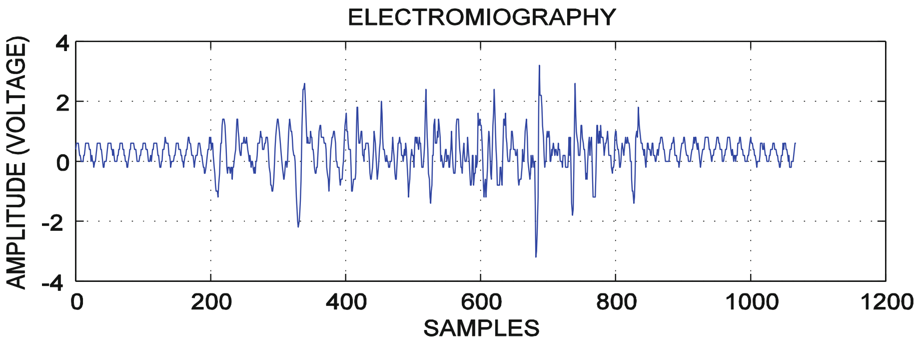

Fig. 6. Myoelectric signal produced by the middle finger

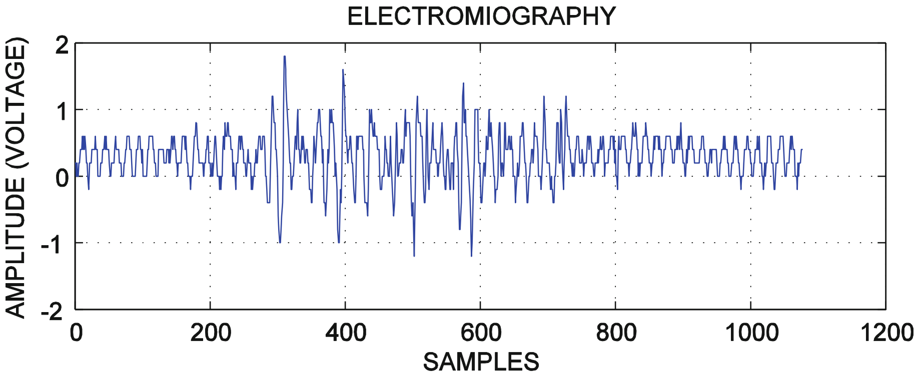

Fig. 7. Myoelectric signal produced by the ring finger

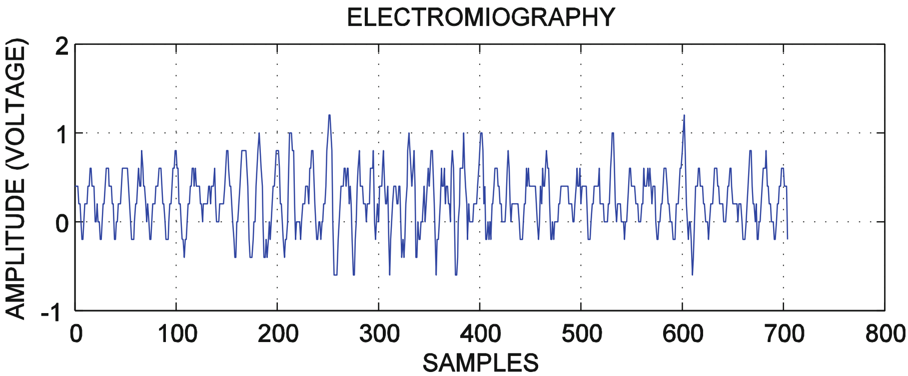

Fig. 8. Myoelectric signal produced by the little finger

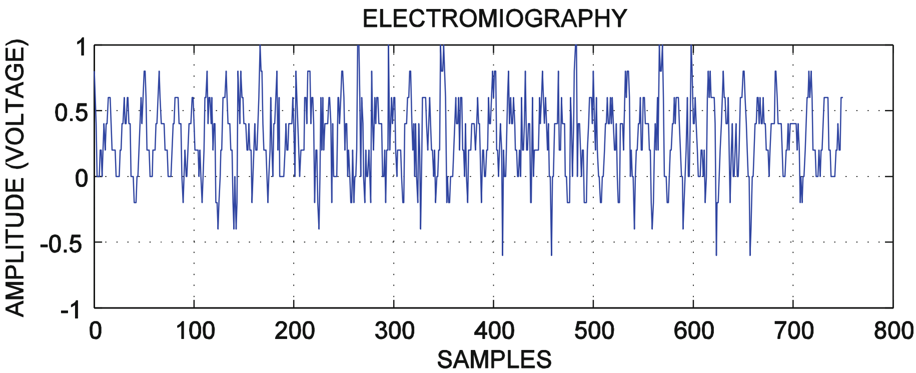

Fig. 9. Myoelectric signal produced by the thumb.

3 Artificial Neuronal Networks

An Artificial Neuronal Networks (ANN) is defined as a nonlinear mapping system. They consist of a number of simple processors connected by connections with weights. Processing units are called neurons. Each unit receives inputs from other nodes and generates a simple scalar output that depends on the available local information, and stored internally or arriving through the weighted connections.

Simple artificial neurons were introduced by McCulloch and Pitts in 1943. An artificial neural network is characterized by the following elements:

1. A set of processing units or neurons.
2. An activation state for each unit, equivalent to the output of the unit.
3. Connections between units, generally defined by a weight that determines an input signal to the unit.
4. A propagation rule, which determines the effective input of a unit from the external inputs.
5. An activation function that updates the new activation level based on the actual input and previous activation.
6. An external entry that corresponds to a term determined as bias for unit.
7. A method for gathering information, corresponding to the learning rule.
8. An environment in which the system will operate, with input signal and even error signals.

An abstract and simple model of an artificial neuron (Fig. 10) composed of a set of inputs $X = (x_1, x_2, x_3, \ldots, x_i)$ which simulates the function of the dendrites; A vector of $W = (w_1, w_2, w_3, \ldots, w_i)$ which emulates the function of a synapse; An action threshold or also called bia θ; An activation function which is considered the equivalent of soma and an output (Pedro and Inés 2004).

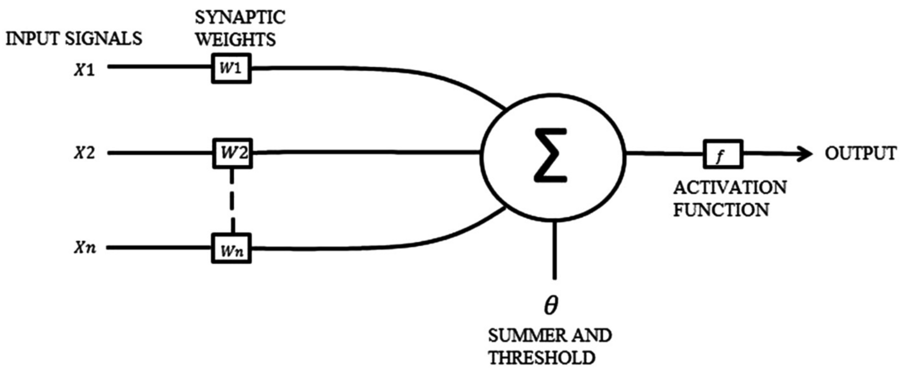

Fig. 10. Natural elements of a neuron.

The descriptions of the components that make up an artificial neuron are:

- Input vector: this vector $X = (x_1, x_2, x_3, \ldots, x_i)$ contains the information that the neuron is going to process; Where "i" is the number of inputs to the neuron.
- Vector of synaptic weights: Each of the elements of the input vector $X = (x_1, x_2, x_3, \ldots, x_i)$ is multiplied by an adjustable value called the synaptic weight and is represented by the vector $X = (x_1, x_2, x_3, \ldots, x_i)$. This vector is adjusted during training, in order to minimize the error of the neuron output with respect to the expected output.
- Threshold or bia: It is denoted by the symbol θ and is considered as an additional weight that receives an input with a value equal to unity.

Activation or transfer function: The rule that establishes the effect of the total input u_t on the activation of the unit is called the activation function F_k (Pedro 2010).

Some of the most used functions are the linear, sigmoid and hyperbolic tangent (Sergio 2017). The equations are listed as follows:

$$f(x) = x$$

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

Starting from a series of random synaptic weights, the learning process looks for a set of weights that allow the network to develop a certain task. Most of the training methods used in neural networks with forward connection consist of proposing an error function that measures the current performance of the network as a function of the synaptic weights. The goal of the training method is to find the set of synaptic weights that minimize (or maximize) the function. There are two types of learning, if the network learns during its normal operation or if the disconnection of the network until the process terminates on-line and off-line networks, respectively (Pedro and Inés 2004).

- Learning algorithms are listed below:
- Descending batch gradient (traingd).
- Gradient batch and traingdm.
- Variable learning rate (traingdx).
- Conjugate gradient algorithms.
- Scaled conjugate gradient (trainscg).
- BFGS algorithm (trainbfg).
- Levenberg-Marquardt (trainlm).

After the training were submitted to different statistical processes in order to compare the performance of the different ANN architectures obtained. Some of the statistical equations used to compare groups of data generated with artificial neural networks are described below. The output of the ANN is OUT_{SIM} the simulated data, the experimental data are named with OUT_{EXP} and n represents the number of values of the samples (Bassam 2014).

Coefficient of determination measures the degree of dependence between variables, taking the value 0 in case of null correlation or the value 1 in case of total correlation. Equivalent to the square of the correlation coefficient.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(OUT_{\exp(i)} - OUT_{sim(i)}\right)^2}{\sum_{i=1}^{n} \left(OUT_{\exp(i)} - OUT_{\exp}\right)^2}$$

The Root Mean Square Error (RMSE) is a measure of the degree of dispersion of the data with respect to the average value. It is known as the standard deviation for a discrete probability distribution:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(OUT_{sim(i)} - OUT_{\exp(i)}\right)^2}{n}}$$

MPE (Mean Percentage Error) is the mean of the percentage error. It is a simple metric, used to see if the forecast error has a positive or negative bias. It is also said that the forecast is underestimated or overestimated

$$MPE = \frac{\sum_{i=1}^{n} \left(\frac{OUT_{\exp(i)} - OUT_{sim(i)}}{OUT_{\exp(i)}}\right)}{n} \times 100(\%)$$

4 Artificial Neural Network Design and Training

Once the signals were captured and a database was generated, the data was subsequently debugged. This was done by submitting the database to a statistical process which gave us the variables that were considered for the input vector of RNA, these variables are: variance, standard deviation, mean square value, RMS value, symmetry coefficient and kurtosis. For the training of the neural network we used the variables derived from the database debugging and the Matlab package, which offers us a series of predefined functions which facilitate the training process (Demuth et al. 2007). We performed different tests dividing the database into 3 in different percentages, that is, 60% for training, 20% validation and 20% for test. Before training, it was considered to define different criteria, such as the number of hidden layer neurons, the learning factor among others. The Levenberg-Marquardt algorithm was used for the training of the tests, since in comparative studies (Sergio 2017) it has shown a better performance with respect to the algorithms of backpropagation, descending gradient, among others.

An algorithm was proposed which helped to calculate the following parameters:

- Number of neurons in the hidden layer.
- Learning factor.
- Linear correlation coefficient.
- The RMSE.
- Synaptic weights and bias.
- Storage of results of each training.

The artificial neural network which has the following characteristics (Fig. 11):

- Network type: perceptron.
- 8 inputs, 7 neurons in the hidden layer, one neuron in the output layer and one output.
- Hyperbolic function in the hidden layer of RNA.
- Linear function in the RNA output layer.
- A Levenberg-Marquardt optimization algorithm.

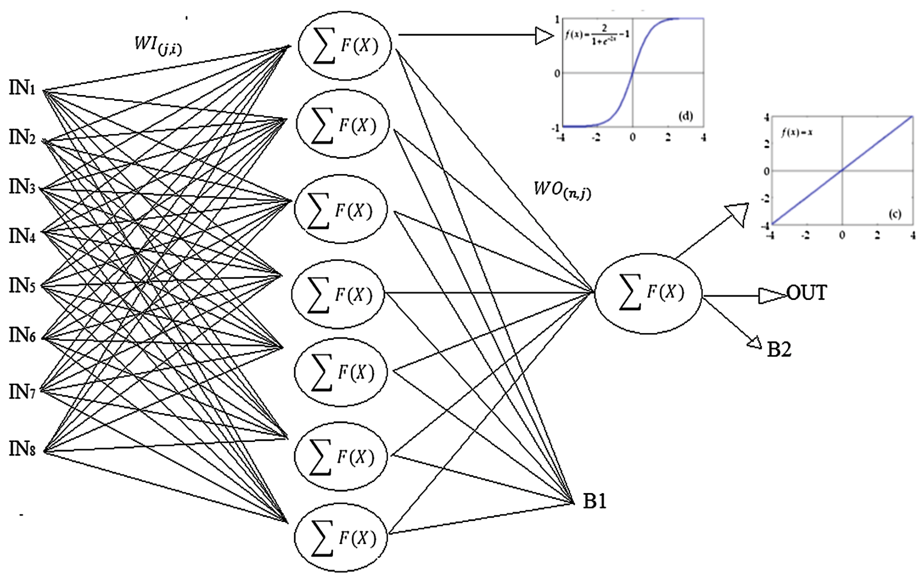

Fig. 11. Neural network graphical model.

In the Table 1 an extract of a report generated from the training is presented, in the Fig. 12 is shown an outline of which are the stages that were followed in the work. This report was obtained using the data base of the myoelectric signals and with the 8 variables derived from the debugging of the database with the help of statistical methods. In this example the architecture No. 7 exhibits the best correlation (0.9622). The cases were calculated the R^2 (test), RMSE (global and test) and the C_R of the variables. The global RMSE includes 100% of the training data and the test data only 20%.

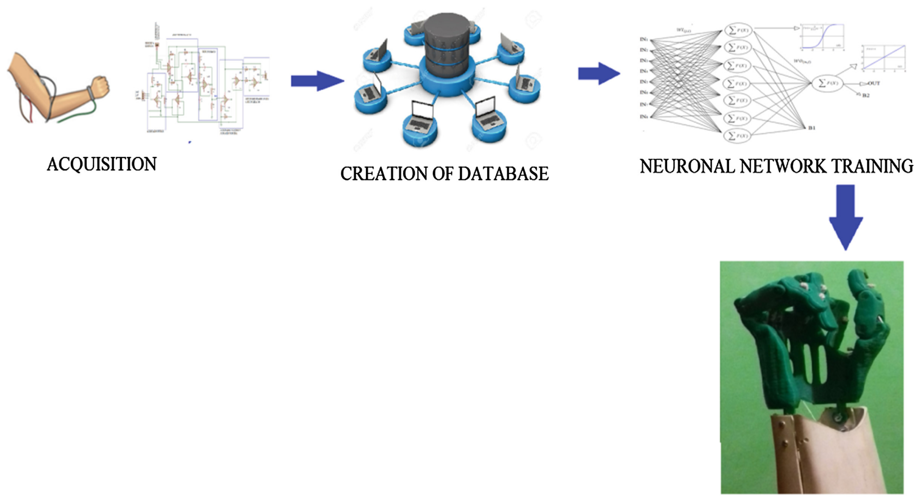

Fig. 12. Diagram of the stages by the project is constituted.

Table 1. Report extract generated by the learning algorithm.

Number of architecture	Architecture ANN	Numbers of neurons	Epoch	Root Means Square Error (*RMSE*)	Means Percentage Error (*MPE*)	R^2 Coefficient of determination	Best linear equation
1	8–1–1	1	1000	0.554209372	17.36174307	0.9004	$0.81T + 0.71$
2	8–2–1	2	1000	0.285012061	12.64856695	0.9502	$0.88T + 0.43$
3	8–3–1	3	1000	0.283046879	13.77088346	0.9504	$0.89T + 0.04$
4	8–4–1	4	1000	0.27890845	11.34057548	0.9516	$0.89T + 0.35$
5	8–5–1	5	1000	0.213592679	11.66547187	0.9628	$0.92T + 0.3$
6	8–6–1	6	1000	0.230296651	10.53668406	0.9599	$0.94T + 0.2$
7	8–7–1	7	1000	0.21687368	10.19372467	0.9622	$0.93T + 0.27$
8	8–8–1	8	1000	0.264669891	12.85161597	0.9539	$0.9T + 0.33$
9	8–9–1	9	1000	0.275548412	11.76859909	0.9517	$0.92T + 0.28$
10	8–10–1	10	1000	0.267190521	11.66423641	0.9535	$0.94T + 0.22$

After completing the training of the network with the architecture No. 7, the validation stage was started in which the database compiled in the electromyography was tested. The objective is to compare the experimental and simulated outputs, in order to bring the ANN architecture to a development card (Arduino) and thus give an application to the ANN.

5 Results and Implementation of the Neural Network

The results obtained with this architecture were satisfactory obtaining the following parameters:

- *RMSE*: 0.2168 (Root Means Square Error).
- *MPE*: 10.194% (Percentage Error).
- Alignment: 0.93*T* + 0.27.
- Synaptic weights.
- Pathways.

The linear correlation between the measured and the estimated ANN obtained in test 7 is shown in Fig. 13, the values of *WI, WO, B1* and *B2*, respectively, are shown in Tables 2 and 3.

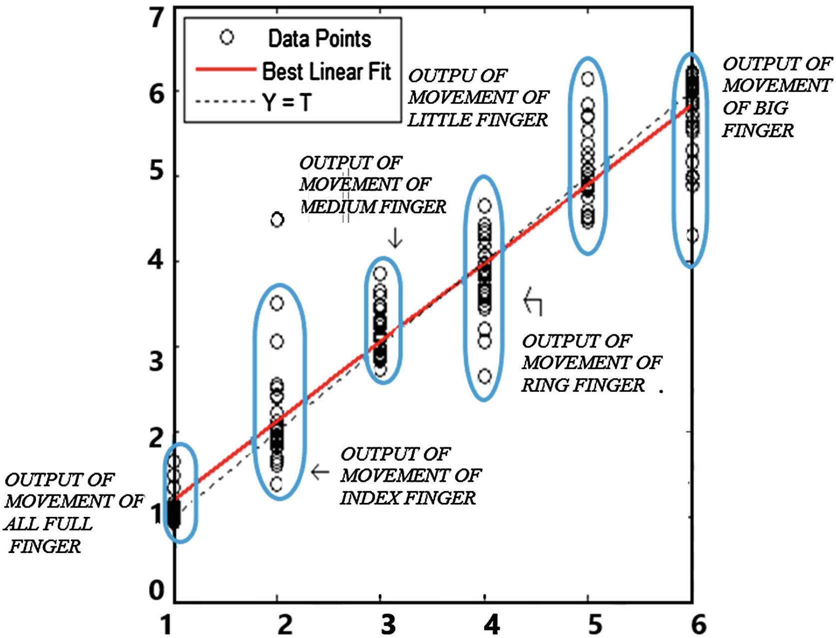

Fig. 13. Graph of linear correlation between the experimental and simulated results.

Table 2. Synaptic weights of architecture #7, of the 8 inputs for the 7 neurons of the hidden layer.

Synaptic Weight Matrix WI							
0.8475	2.2854	2.1476	0.1976	−0.5232	−3.9432	−2.5756	−3.3085
3.7104	−9.9741	−10.029	−5.5328	−3.5506	−5.6621	−7.5008	−6.102
14.8244	13.2162	−0.6775	4.2638	−3.1937	−3.8813	−6.7063	0.6947
−7.9776	−15.0781	3.212	6.7119	1.209	1.3704	9.7667	1.7998
1.5851	0.6765	−0.6552	2.0935	−1.1748	1.1137	−1.5177	−0.7433
−3.9121	−41.0433	−11.7778	8.4399	−1.998	10.4343	−22.3041	0.3034
6.7575	3.4153	1.9836	−19.1374	1.4781	−6.1953	8.4232	−3.6328

Table 3. Synaptic weights and paths of the hidden layer and the layer of exit.

Synaptic weights and bias of the hidden layer and exit layer							
Vector WO	−1.9619	1.0649	1.2106	1.004	1.8337	−1.2558	0.9283
Vector B1	6.8032	7.7847	−3.5825	3.3002	5.0344	10.8505	2.6925
Vector B2	1.4902						

5.1 Equation Resulting from ANN

Finally we present the proposed equation obtained from architecture #7, given by:

$$\boldsymbol{OUT} = \sum_{j=1}^{7}\left[\boldsymbol{WO}_{(\boldsymbol{n},\boldsymbol{j})}\left(\frac{2}{1+\boldsymbol{e}^{-2\left(\sum_{i=1}^{8}\left(\boldsymbol{WI}_{(j,i)} * \boldsymbol{In}_{(i)}\right)+\boldsymbol{B}2\right)}}-1\right)\right]+\boldsymbol{B}1$$

WHERE:

- $WI_{(j,i)}$ = Matrix of synaptic weights of the inputs.
- $WO_{(n,j)}$ = Matrix of synaptic weights of the hidden layer.
- $In_{(i)}$ = Input data.
- $B1$ = Paths of the hidden layer.
- $B2$ = Path of the output layer.

5.2 Application with Arduino

The final application of the previous processes, an ARDUINO MEGA 2560 development board was used, it was possible to carry out a demonstrative application in which the proposed ANN equation was entered in a simplified way. This application was achieved with an algorithm show in Fig. 14 hat has the purpose of making a selection of the input data, which were obtained from the database debugging, it is worth mentioning that each input data is a result of a Statistical process applied to each signal, produced by the movement of each of the fingers. After performing the process of data entry, these are introduced to the ANN equation obtaining an output value which will trigger the movement of one of the fingers of the prototype of the prosthesis.

In order to validate the proper functioning of the training and the result of the neural network, an arduino program was performed, in which a part of the database, the synaptic weights and the bias are entered.

The program that was developed consists of three steps which will be explained in detail:

- Step one: the program generates a random number, that number represents a row of the database which is an 8 × 24 matrix (columns per row), the data in that row is entered into the resulting function of the training of the neural network.
- Step two: data entered into the function of the neural network results in a predicted output according to the row that was selected each result obtained will generate an action.

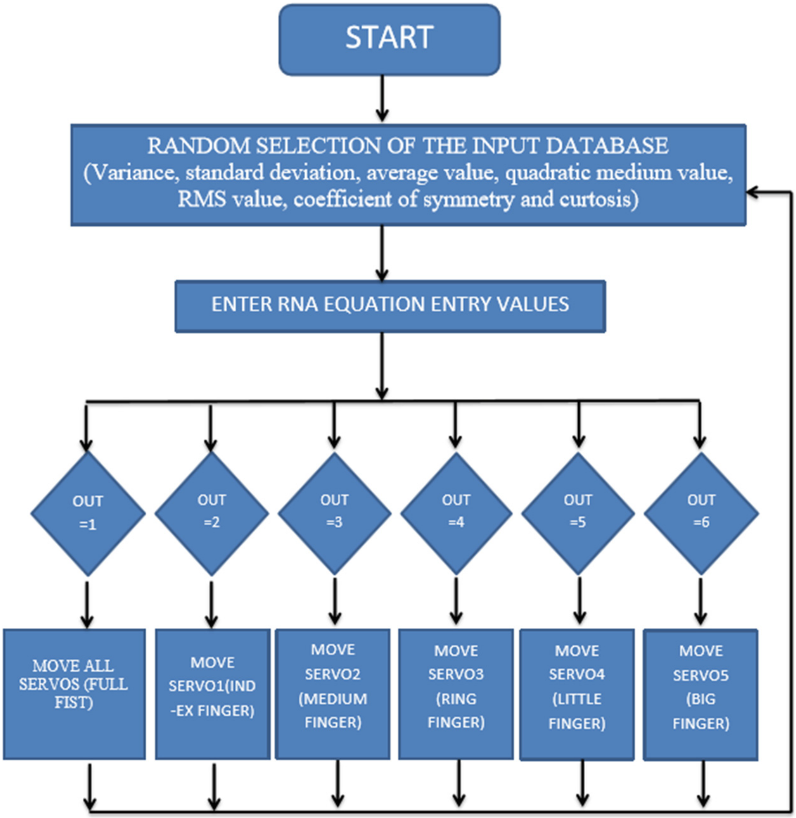

Fig. 14. Algorithm used for the application in ARDUINO MEGA 256.

- Step three: each action is based on the calculated output, it is worth mentioning that these outputs have an error rate of 10.94% (*MPE*). The order of the actions is as follows.

- *Out* = 1: All servos are actuated, imitating how the complete fist closes (Fig. 15a).
- *Out* = 2: the servo is actuated by mimicking the contraction of the index finger (Fig. 15b).
- *Out* = 3: the servo two is actuated by mimicking the contraction of the medium finger (Fig. 15c).
- *Out* = 4: the servo three is actuated by mimicking the contraction of the ring finger (Fig. 15d).
- *Out* = 5: the four servo is actuated by mimicking the contraction of the little finger (Fig. 15e).
- *Out* = 6: the five servo is actuated by mimicking the contraction of the thumb (Fig. 15f).

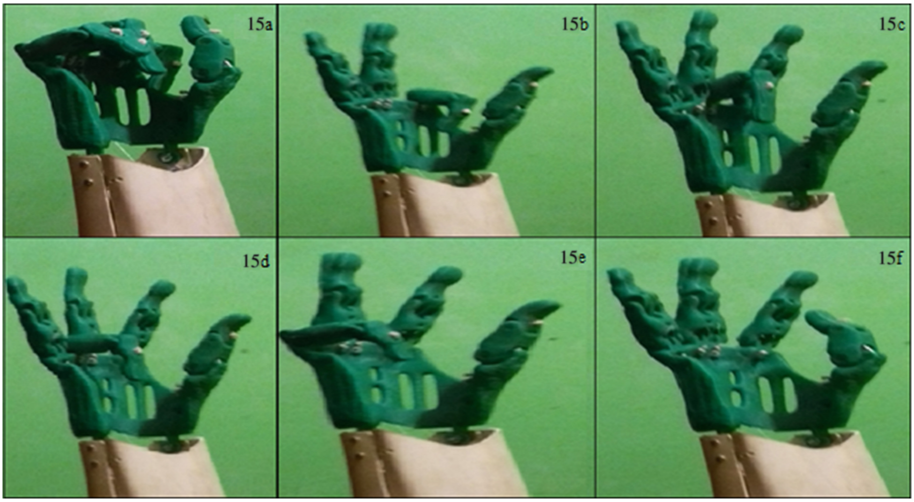

Fig. 15. (a) *Out* = 1 full closed fist, (b) *Out* = 2 contracted index finger, (c) *Out* = 3 contracted middle finger, (d) *Out* = 4 contracted annular finger, (e) *Out* = 5 contracted pinky finger, (f) *Out* = 6 contracted thumb.

6 Conclusions

Through the design of a myoelectric sensor it was possible to obtain myoelectric signals, the contraction of the muscles responsible for the movements of each finger, of a human hand, as well as to develop a way of selection and characterization of myoelectric signals. Using neural networks we can obtain a selection of signals with a small degree of error, with this we offer a good performance of the prototype of the prosthesis.

References

Parimal, P., Allen, S., Rapach, G.: Microcontroller based prosthetic device using myoelectric signal. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1427–1428 (1988)

López, N., di Sciasco, F., Orosco, E., Ledesma, N., Echenique, A., Valentinuzzi, M.: Control Mioelétrico de un brazo robótico. In: IV Congreso Iberoamericano sobre Tecnologías de Apoyo a la Discapacidad. Vittoria-Espiritu Santo-Brasil (2006). ISBN: 8496023451

Alonso, A., Hornero, R., Espino, P., de la Rosa, R., Liptak, L.: Entrenador Mioeléctrico de Prótesis para Amputados de Brazo y Mano. Universidad de Valladolid (2001)

Pedro, V.I., Inés, G.L.: Redes de Neuronas Artificiales, Un enfoque Práctico. Pearson Education, Madrid (2004)

Pedro, C.P.: Inteligencia Artificial con Aplicaciones en Ingeniería. Alfaomega, México D.F. (2010)

Bassam, A., Conde-Gutiérrez, R.A., Castillo, J., Laredo, G., Hernández, J.A.: Direct neural network modeling for separation of linear and branched paraffins by adsorption process for gasoline octane number improvement. Fuel **124**, 158–167 (2014)

Sergio, V.V.A.: Adquisición, Procesamiento e Interpretación de Señales Mioeléctricas para el Diseño y Fabricación Robótica, tesis de licenciatura. Universidad Autónoma del Estado de Morelos, Cuernavaca (2017)