ISSN 2395-8618

Implementation on an FPGA of Perceptron Algorithm for Pattern Classification and Recognition in Electromyographic Digital Signal Processing

J. Brandon Mañón Juárez, Eusebio Ricárdez Vázquez, Pablo Manrique Ramírez

***Abstract*—This paper presents the design and synthesis of a perceptron as an pattern recognition model for its implementation for digital electromyographic signal processing for the classification of movements from wrist and arm. In order to reduce the implementation hardware and accelerate operations for pattern recognition and digital signal processing, the perceptron algorithm, digital signal filtering, segmentation and characteristics extraction are designed with Verilog HDL and implemented at hardware level in a Field Program Gate Array. After obtaining and characterizing the signal, from registered signal values in MATLAB, the simple perceptron training algorithm is used to obtain the synaptic weights of the perceptron, which are then implemented in the FPGA design for the perceptron module in hardware. After implementing the complete system and performing the tests, the signals obtained are analysed with their classification and the percentage of success, with results of 70% and 80% person A and 60% and 80% for persond B for contraction and extension, respectively; it should be noted that the complete system was only tested on two people.**

***Index Terms*—Electromyography, digital signal processing, artificial neural network, FPGA.**

I. Introduction

Within the scope of computer engineering are some systems that process information whose variables are continuous in time and which can be modeled computationally either by software or hardware, making better use of the resources that comprise them. Some systems that are frequently modeled through computation are the biological ones; a particular case is those used for the analysis of muscular ailments or the control of prosthesis through electromyographic signals (EMG) [5] which are electrical signals that the muscles produce in response to the electrical signal sent by the brain to the motor neurons that make up the muscle and cause the movement.

An electromyographic signal acquisition, conditioning and processing system is defined as the set of elements that interact with each other to capture the electrical signals produced by the muscular movement of some part of the body, then the signal is prepared to have it or obtain information with specific characteristics and at the end calculations or operations are performed to perform some action from the results of the signal. The signal processing system is usually embedded in a single computing device that consists of the necessary elements for each stage. Despite the progress in terms of computing and information processing capabilities, the needs and requirements for the design and implementation of EMG signal processing systems require devices that, despite their high speed, low energy consumption, resource optimization and reduced device size.

Some prototype projects that process EMG signals present areas of opportunity in terms of design, on the one hand in hardware design to reduce size and implementation space, improve energy consumption by reducing the number of devices to a minimum and replacing analog filters with digital ones, just to mention a few, and on the software side it is suggested to accelerate the processing capacity, optimize the implementation algorithms, as well as the implementation of pattern recognition and classification stages. This work is aimed at the integration of digital signal processing techniques to cover part of the areas of opportunity in the prototypes as described by [3] and [11] as well as the acceleration of EMG signal processing through the implementation of a field programmable gate array (FPGA), in which pattern recognition and classification are also implemented by means of an artificial neural network model. It is emphasized that the application of the system is for general purpose use so it can be implemented in different use cases.

II. Related Work

Digital signal processing systems applied to electromyography are developed according to the needs of the case study or biomedical application and the techniques used vary according to the requirements and design specifications, however, many systems have common characteristics and specifications in both design and implementation. Cases such as [1] and [6] where an EMG signal acquisition bracelet is used, what varies is the type of device, respectively, while one is commercial and integrates more types of sensors,

Manuscript received on 21/05/2023, accepted for publication on 20.08/2023.

The authors are with the Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, Mexico ({jmanonj2023, ericardez,pmanriq}@cic.ipn.mx).

ISSN 2395-8618

the other is designed only for acquisition by electrodes, but both acquire the information wirelessly and is processed in MATLAB.

In the designs of [4,7,9] the analog conditioning stage is designed, in the first case a robotic arm is controlled by pattern recognition through connection to a BeagleBone development board, in the second case an avatar of a mobile device game is controlled to perform muscle rehabilitation in a playful and supervised way, for the third case the system is only used to analyze the behavior of the EMG signal, however, in all three cases the conditioning stage covers a large size and space by requiring many electronic components in the analog bandpass filter stage.

In the prototypes that perform digital signal processing using an FPGA there are several techniques, for example, [2] who proposes two architectures of FIR (finite impulse response) type digital filters, the first in series and the second in parallel where it is highlighted that the parallel architecture, despite using more FPGA resources, the processing speed and filtering time are better compared to the serial architecture.

In [10], a specific way of performing digital processing of electromyographic signals is defined, although it uses an analog conditioning stage, the digital processing has stages that perform complex calculations such as Fast Fourier Transform, dimensionality reduction, segment variance extraction, as well as its mean value. The prototype obtains a percentage higher than 66% of success in the classification of the three movements, it is highlighted that although some stages are executed in parallel, the process control is by means of a state machine which resembles the sequential execution of the processing.

III. Methods and Materials

Based on the characteristics of a digital signal processing system, a design proposal is defined that takes into account techniques that cover the areas of opportunity of the existing prototypes; the diagram of the proposal is shown in Fig. 1.

The EMG signal is obtained from the muscle movement of wrist contraction or rotation by two electrodes placed on the user's upper limb, it is taken to the analog conditioning stage in which a gain is obtained in the signal by means of an instrumentation amplifier. Then band-pass filtering is performed for 60 Hz that corresponds to the noise of the power supply installation and the signal is conditioned with a displacement and amplification for processing in a digital device. In the digital stage the analog to digital conversion is performed, each sample is transferred to the FPGA to pass through the digital bandpass filter and start the segmentation of the ranges with considerable information of the EMG signal. From each segment two characteristics are obtained, absolute value and wavelength, which in the simple perceptron stage allow to obtain a movement classification that is performed.

A. Analog Signal Conditioning

The analog signal conditioning stage is divided into three parts: preamplification, band-reject filter and displacement. Considering [7], the instrumentation amplifier used is the INA128 to which, in order to obtain an EMG signal gain of approximately 2.5, a gain resistor $R_G = 33K\Omega$ is connected. Likewise, the band reject filter is also taken from [7] and the values of its components are defined as shown in Fig. 2, for which, the filter capacitor $C_0 = 68nF$, the quality factor $Q = 5$, the filter resistor $R_0 = 3.9K\Omega$, and the resistor for the quality accuracy $R_Q = 39K\Omega$.

For signal displacement and amplification, an amplifier in differential configuration is used to meet, from Equation 1, the displacement required for the EMG signal:

$$V_o = 1.5V_{sEMG} + 1.5V. \tag{1}$$

From Equation 1, value 1.5 is factorized and taken to Equation 2, which corresponds to the differential amplifier, so now $R_{10}/R_9 = 1.5$, and it is the gain factor:

$$V_o = \frac{R_{10}}{R_9}(V_2 - V_1). \tag{2}$$

Fig. 3 shows the analog signal conditioning circuit, in the signal displacement stage, is used the differential amplifier with gain of 1.5, and proposing for $R_{10} = R_8 = 18K\Omega$ in Equation 2, is obtained $R_{11} = R_9 = 12K\Omega$. From the circuit a module is designed for PCB implementation and connection to the development board for digital processing.

B. Analog to Digital Conversion

The EMG signal has a frequency between 50 and 500 Hz, according to [10], it concentrates its highest energy between 50 and 250 Hz, taking into account the Nyquist theorem for signal sampling, it is defined that the sampling frequency is 1100 Hz, i.e. $2.2f_{sEMG}$. The analog to digital converter of the SAMD21 microcontroller is used to acquire the EMG signal, the conversion is configured with a resolution of 10 bits per sample.

Also, due to the programming model of the microcontroller and the distribution of its pins, a data transfer format to the FPGA is required, so shift registers are implemented through which the analog to digital conversion bits are received and then the processing result bits are transmitted to the microcontroller. Fig. 4 shows the block diagram of the microcontroller elements and the FPGA modules that interact in the processing.

C. Digital Filter

The digital filter is designed using MATLAB's Filter Designer Toolbox, in which the filter type, cutoff frequencies and, since it is a digital filter operating binary values, the word size and the format of the binary values are defined. After designing the filter, the code is generated in Verilog corresponding to the hardware to be configured in the FPGA.

ISSN 2395-8618

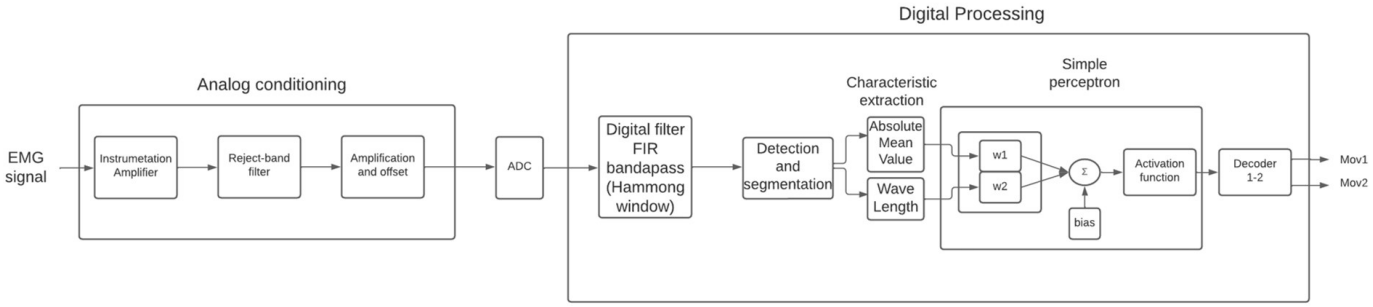

Fig. 1. Digital signal processing for EMG signal block diagram

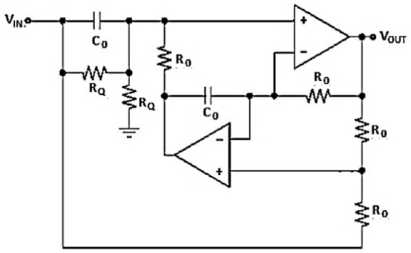

Fig. 2. Band-reject filter

The word format assigned for the values is 22 bits of fixed point type, of which 10 are for the fractional part, 11 for the integer part and one sign bit.

D. Signal Segmentation and Characteristic Extraction

A signal segmentation stage is required in which significant and not false movements are identified, so it is convenient to identify at what moment a movement is carried out with the continuous comparison of the ranges of the signal samples. Similarly, when identifying a motion, a significant number of samples should be obtained from which to obtain features, in this case the mean absolute value and wavelength of the segment. Equations 3 and 4 define how to obtain the number of samples required from the signal:

$$N_{min} = \frac{T_{min}}{T_s}\,, \tag{3}$$

$$N_{max} = \frac{T_{max}}{T_s}\,. \tag{4}$$

As a function of the minimum and maximum times of the EMG signal (T_{min} and T_{max}) and the sampling period (T_s).The number of samples in the project is 50, where two signal cycles and a tolerance of six samples are taken into account due to the speed of signal capture.

Fig. 5 shows the signal segmentation algorithm for subsequent implementation with Verilog.

At the start of segmentation each sample is processed in the absolute mean value and wavelength, as suggested [8], extraction modules to obtain a segment characteristic value. Equation 5 corresponds to the absolute mean value (AMV) and Equation 6 to the segment wavelength (WL):

$$AMV = \frac{1}{N}\sum_{i=1}^{N}|x_i|\,, \tag{5}$$

$$WL = \sum_{i=1}^{N}|x_i - x_{i-1}|\,. \tag{6}$$

From the above equations N represents the number of samples, x_i is the value of a given current sample at the output of the filter, x_{i-1} is the previous sample stored in an accumulator register. Each equation is described in Verilog with which its equivalent abstraction in hardware is obtained, as shown in Fig. 6 and Fig.7.

E. Characteristic Classification

After extracting the features of each segment, the results are routed to the inputs of the simple perceptron for pattern recognition and motion classification to be performed. The simple perceptron is defined according to Equations 7 and 8:

$$\widetilde{y} = \sum_{i=1}^{N} x_i w_i + \theta\,, \tag{7}$$

$$\widehat{y} = \begin{cases} 1 & : \widetilde{y} > 0 \\ 0 & : \widetilde{y} \leq 0 \end{cases}\,, \tag{8}$$

where $\widetilde{y}$ corresponds to the prediction value as a function of the features x_i and the synaptic weights w_i, θ corresponds to the prediction trend threshold shift. Finally $\widehat{y}$ corresponds to the prediction or decision that is made as a result of the perceptron activation function.

Weights w_i are obtained by training the perceptron, i.e., they are defined as a synaptic learning threshold which is stored after learning or training the perceptron:

$$\widetilde{y} = AMV * w_1 + WL * w_2 + \theta\,. \tag{9}$$

In Equation 9, the perceptron model is defined considering the characteristics obtained in the segmentation. By defining

ISSN 2395-8618

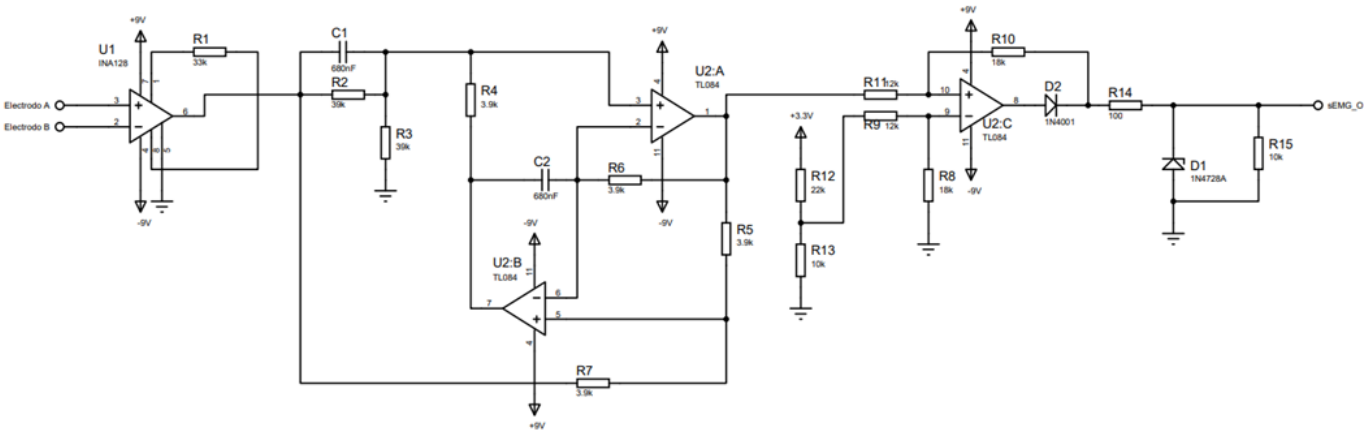

Fig. 3. Signal conditioning scheme

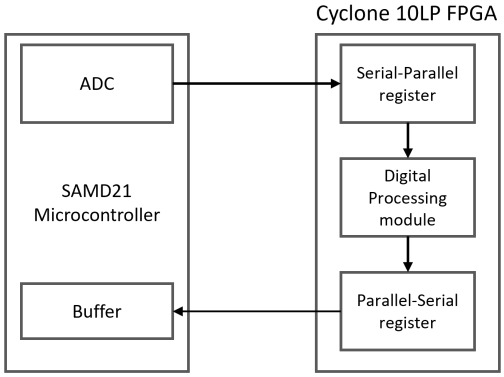

Fig. 4. FPGA and microcontroller interconnection

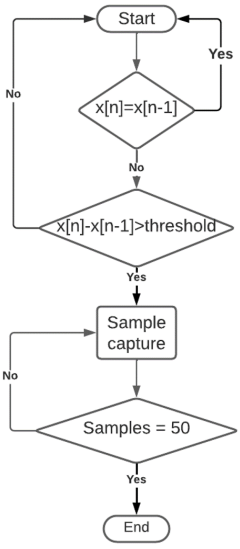

Fig. 5. Sample capture flow diagram

the equations in Verilog, the weights are obtained by being described in hardware as a ROM memory with stored values for each feature. Finally, by obtaining the hardware equivalent of the perceptron model, the classification is decoded into a two-bit label where 01 represents arm contraction and 10 represents wrist rotation. Fig. 8 shows the RTL diagram corresponding to the perceptron module designed with Verilog.

F. Perceptron Training and Module Implementation

The implementation process consists of two stages, the first one has to do with the training of the perceptron and the second one with its application to pattern recognition. In the training stage, the device is connected to the computer via USB, a MATLAB script is executed in the computer which begins to acquire the values of the filtered and segmented signal from the FPGA. The perceptron training algorithm coded in MATLAB is taken from the below description.

– Random values are assigned for the synaptic weights of the and the threshold of the perceptron.
– The first step is to choose the vector $[x_j]$ of features i that enters the perceptron.
– If the prediction value obtained with Equation 7 is different from that corresponding to the label $y(x_j)$ of feature vector j, the weights are modified as indicated by the Equations 10 to 13. Otherwise, the weights do not change:

$$w_i^{new} = w_i^{old} + \Delta w_i, \tag{10}$$

$$\Delta w_i = y(x_j)x_i, \tag{11}$$

$$\theta_i^{new} = \theta_i^{old} + \Delta_i, \tag{12}$$

$$\Delta\theta_i = y(x_j). \tag{13}$$

– If the training completion criterion is not met, it is necessary to return to the vector assignment and repeat the process.

ISSN 2395-8618

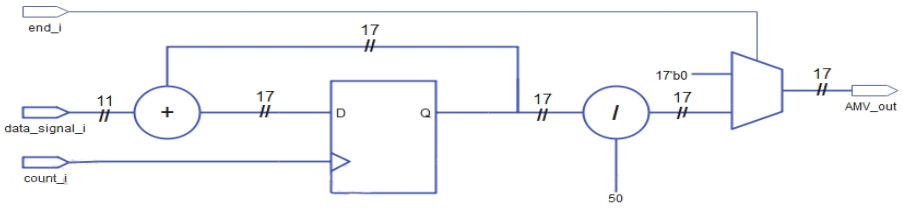

Fig. 6. AMV module RTL scheme

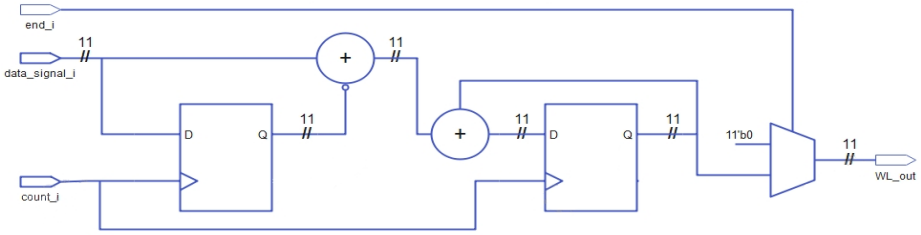

Fig. 7. WL module RTL scheme

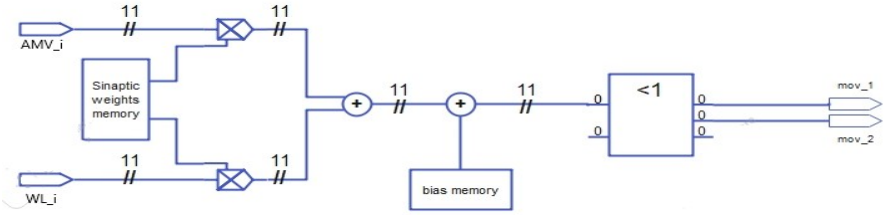

Fig. 8. Perceptron RTL diagram

When the synaptic weights are defined, they are formatted in 22-bit fixed-point format in order to load those values into a .hex format file that is in turn defined in the initialization configuration of the ROMs that make up the simple perceptron module in Verilog. After the training procedure and definition of synaptic weights, the Verilog project must be recompiled to reload the design in the FPGA and reconfigure it with the trained perceptron.

In the application to pattern recognition, the device no longer requires connection to a computer because the system becomes independent as it already has the configuration of digital processing and training of the perceptron in the FPGA. The user again has to put on the electrodes, however, it is only to recognize the movements that are made, the device displays two LEDs the result of the classification label that identifies the movements.

IV. Results and Analysis

In the digital processing stage, the ideal digital filter shows stability in the behavior depending on the specifications given, however, when performing an adjustment and transition of the values to be processed in fixed-point binary format, the behavior of the filter is altered, as shown in Fig. 9.

Despite the behavior of the digital filter, the signal is attenuated in the range of frequencies higher than 500 Hz, as shown in Fig. 10.

In the implementation to obtain the segmented signal, the acquisition of the information is obtained from five people for six seconds for each user sample and for each of the two movements to be recognized; the values are automatically recorded in a text file. The values acquired by the device and stored in the text files are plotted in MATLAB, Fig. 11 shows the data graphs of the movement performed by all the sample users and grouped in one register for wrist contraction and rotation. Likewise, with the segmented signal obtained previously, the classification of both movements can be seen when executing the training algorithm.

The synthesis of the main project for the digital processing stage reports the FPGA resource consumption that, for the Cyclone 10 LP device model 10CL016YU256C8G whose capacity is 15,408 logic elements uses 1,147 logic elements, that is, less than 10% of the FPGA for the implementation of the complete module, so it is defined that it is possible that at least 5 equal modules can be implemented in the FPGA.

Training the perceptron with the segmented values allows finding a hyperplane of separation for the movements. In a MATLAB script the data is sorted to assign classification

ISSN 2395-8618

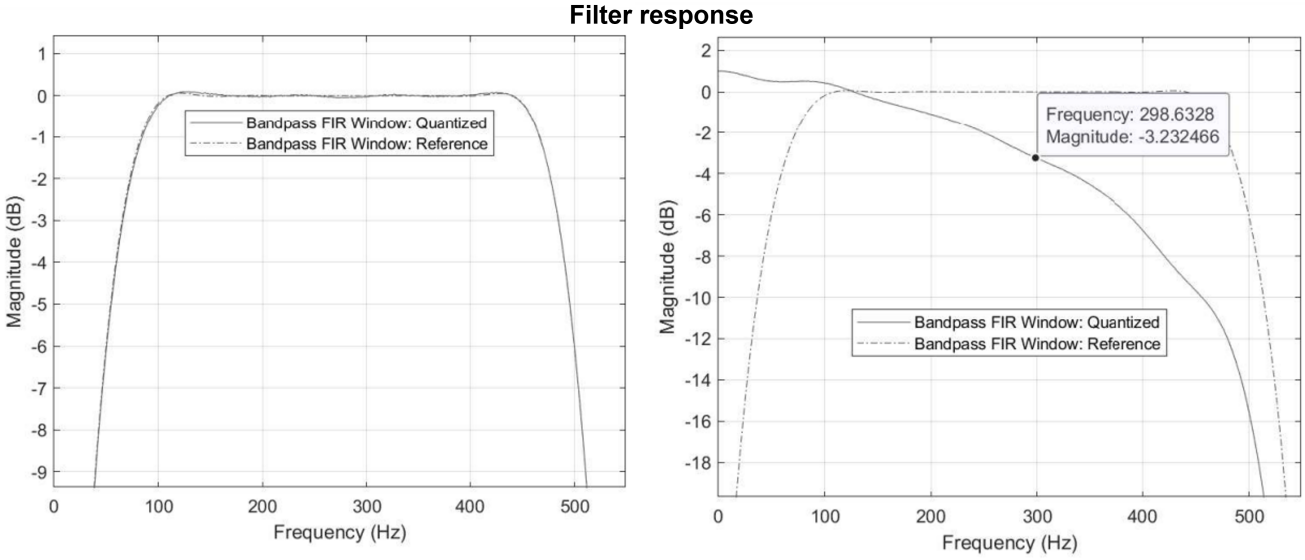

Fig. 9. Digital filter design response (a) Without modifying word, (b) With modifying word

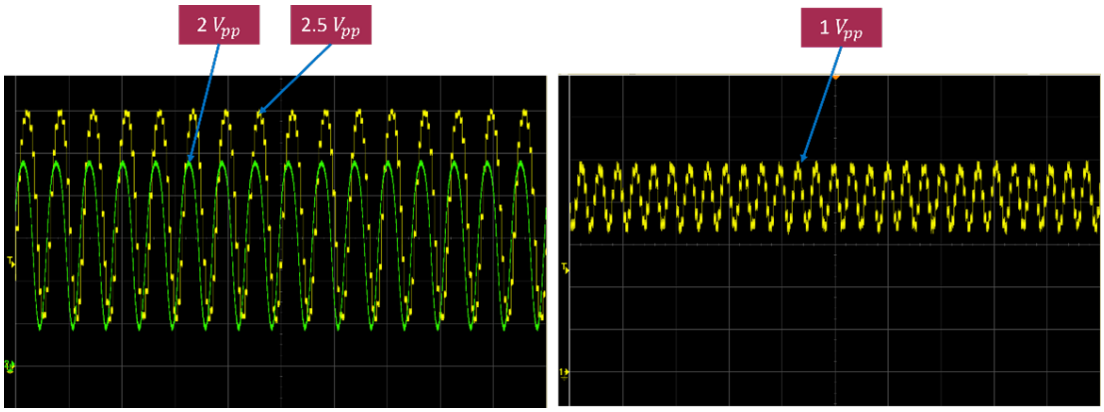

Fig. 10. Signal filtering

according to the movements and subsequently the perceptron training algorithm is executed with which the synaptic weights w_1 and w_2 are obtained after a separation hyperplane is found for the registered values. With the training samples, the epochs or iterations performed by the program are close to 300, i.e., with the path of each segment and the iterative change of weights to find the correct classification, approximately 300 value changes are required. Fig. 12 shows the final result of the perceptron training which identifies that it was possible to train and classify with the implemented system. Triangles indicate contraction and asterisks indicate wrist rotation.

Regarding the percentage of prediction successes of the device, when implemented independently and tested in two people, of ten repetitions for contraction of which in person A only seven prediction were asserted and for person B only six. Thus, it is determined that, for the first test, the percentage was 65 percent correct. For the wrist rotation movement, in person A and person B the assertion was eight of ten movements. of the movements in this case the percentage of prediction is 80%. Since the system was tested on only two people, it is necessary to use the system on more people to be more certain of the percentage of performance as well as the factors that influence a change in response and accuracy.

V. Conclusions and Future Work

The application of mathematical models and basic machine learning algorithms like perceptron for artificial neural networks by means of computation at the software or hardware level is possible based on the knowledge of the technologies used. When software interacts with hardware, great computing power is achieved. In the development of the project, digital

ISSN 2395-8618

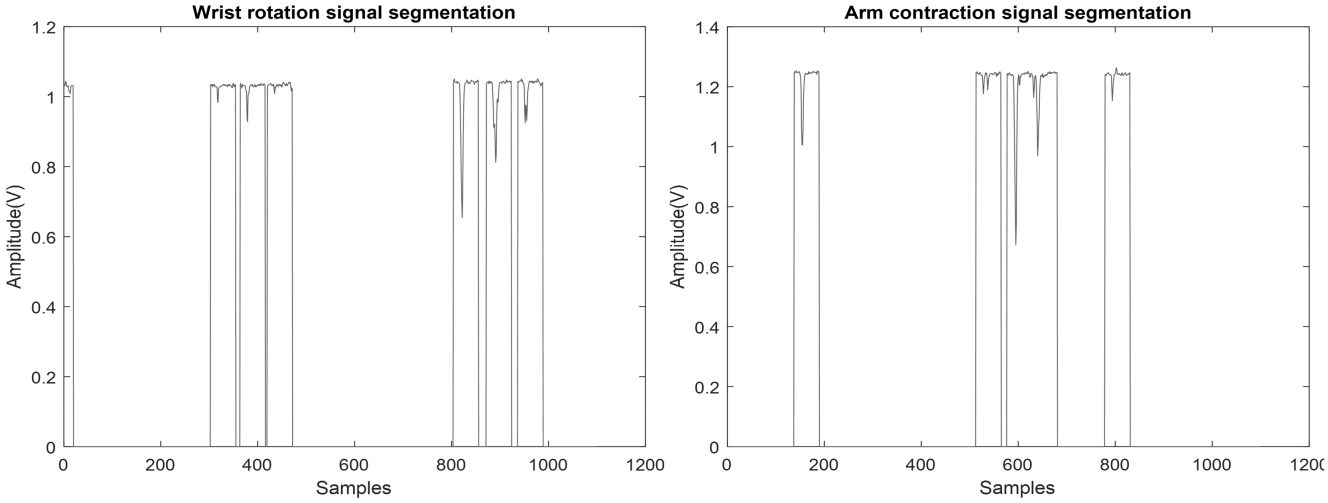

Fig. 11. Movements of signal graphics

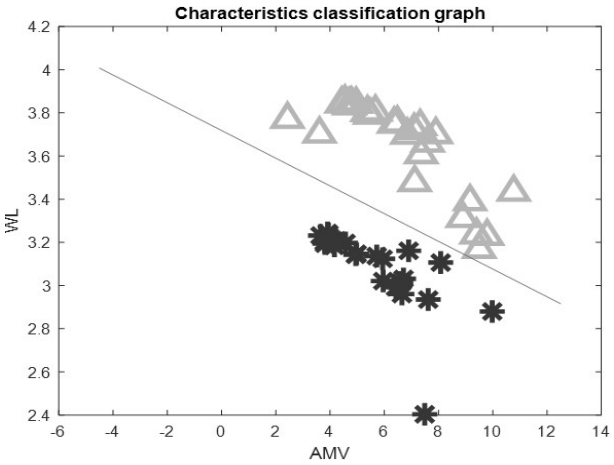

Fig. 12. Training and classification

signal processing algorithms are implemented at software and hardware level, it is identified that both implementations may differ in accuracy, however, the results are similar. It leaves a wide panorama of opportunities in the perspective that computer engineering is not only software development but also hardware development and that both can interact for the optimization of the computation of real world models.

In the case of signal processing and segmentation algorithms, as well as the simple perceptron model, it is observed that for some algorithms that are represented in a flowchart and that are executed sequentially when implemented with structured programming languages, they can also be implemented at the hardware level, which causes their operation to be independent of the previous execution of a program segment and at the same time interact with other stages, which gives greater speed for processing.

The influence of the word format stands out in the accuracy of the results, so improvements can be made by parameterize the data format in general for the entire system, i.e., defining from the analog to digital conversion, the design of digital filters and digital processing, the format that the values should have to avoid loss of information and accuracy.

The development board has a wireless communication module so that, in order to improve the system, it can be implemented to transfer the signal acquisition without requiring USB communication.

From the report of resources used by the module, it is defined that it is possible to implement replicas or at least increase the number of perceptrons to be able to classify more movements. Similarly, it is possible that the firmware necessary for the training of the perceptron is carried out in the microcontroller and not depend on a computer.

REFERENCES

[1] A. M. Arias, "Prototipo de mano robótica controlado mediante señales electromiográficas con un dispositivo comercial," *Computación y Sistemas*, vol. 25, no. 2, pp. 307–315, 2021.

[2] Q. I. Artetxe, "Procesado digital de señales mediante circuitos integrados reconfigurables," *Filtros de respuesta impulso finita (FIR)*, Universidad del País Vasco, España, 2014.

[3] M. M. Basáñez, *Diseño e implementación de un sistema de adquisición, procesamiento y clasificación de señal mioeléctrica para prótesis transhumeral de tres grados de libertad*, INAOE, México, 2008.

[4] M. A. Cantillo, "Procesamiento de señales EMG en un sistema embebido para el control neuronal de un brazo robótico," *Revista Colombiana de Tecnología Avanzada*, vol. 32, no. 2, pp. 139–147, 2018.

[5] R.H. Chowdhury, M.B. Reaz, M.A. Ali, A.A. Bakar, K. Chellappan, and T.G. Chang, "Surface Electromyography Signal Processing and Classification Techniques," *Sensors,* vol. 13, pp. 12431–12466, 2013.

ISSN 2395-8618

[6] C. M. Gómez, *Desarrollo de un sistema inalámbrico asistencial para la monitorización a distancia y clasificación de señales electromiográficas en pacientes con movilidad reducida*, Universidad de Antioquia, Colombia, 2022.
[7] M. J. Hernández, *Sistema de rehabilitación muscular de antebrazo mediante un videojuego para dispositivos Android*, Instituto Politécnico Nacional, Mexico, 2017.
[8] P. Geethanjali and K. K. Ray, "EMG based man–machine interaction—A pattern recognition research platform," *Robotics and Autonomous Systems,* vol. 62, no. 6, pp. 864–870, 2014.
[9] G. D. Proaño, *Sistema de adquisición de señales EMG de superficie multicanal para prótesis de miembro superior*, Universidad Politécnica Salesiana, Ecuador, 2019.
[10] L. D. Reyes, "Implementación en FPGA de un clasificador de movimientos de la mano usando señales EMG," *Redes de Ingeniería,* vol. 6, 1, pp. 85–94, Colombia (2015).
[11] L. E. Tenesaca, *Desarrollo del prototipo de un sistema de control para prótesis de mano asistida mediante señales electromiográficas*, Universidad Politécnica Salesiana, Ecuador, 2020.