Platform for the interpretation of the Spanish Sign Language Alphabet based on the Myo Armband Device

Carlos Santiago Viera Betancor, Valentín de Armas Sosa, Félix B. Tobajas Guerrero

Institute for Applied Microelectronics (IUMA)

University of Las Palmas Gran Canaria (ULPGC), Spain

***Abstract*— According to the Spanish National Institute of Statistics, more than a million people suffer from hearing problems in Spain, of which 98% use the oral language to communicate, leaving the Spanish Sign Language limited to less than 10%. These communication difficulties can translate into a feeling of isolation, causing important effects in the daily lives of people who suffer from this problem. Combining the capacity of commercially available gesture recognition devices with the infrastructures offered by IoT and the powerful low-cost hardware development platforms available in the market, this project aims to develop a platform capable of interpreting letters from the Spanish Sign Language alphabet. For this, the Myo Armband device will be used, and thanks to its electromyographic sensors (EMG), in combination with a gyroscope and an accelerometer, will allow the detection of hand movements.**

Keywords: Spanish Sign Language, Electromiography, Myo Armband, Support Vector Machines

I. INTRODUCTION

Internet of Things has acquired an important presence in the field of accessibility, offering solutions to the elderly o people with some type of disability, with the main objective of facilitating their lives, trying to reduce their limitations as much as possible. Thus, in this project a solution is proposed for people with hearing or speaking problems, facilitating their communication through sign language, based on IoT devices. According to the Spanish National Institute of Statistics, more than a million people suffer from hearing problems in Spain, of which 98% use the oral language to communicate, leaving the Spanish Sign Language limited to less than 10%. These communication difficulties can translate into a feeling of isolation, causing important effects in the daily lives of people who suffer from this problem.

The main objective of this project consists of the design and implementation of an HW / SW platform that allows the interpretation of the fingerprint alphabet of the Spanish Sign Language in real time, using the Myo Armband device [1] and the Arduino Nano 33 IoT development platform [2].

For the development of this platform, three main objectives are to be met:

- Detection of symbols from the EMG sensors and the IMU of the Myo Armband device and, through a BLE connection with the Arduinno Nano 33 IoT device, extract the information from the sensors for further processing.
- Development of the code necessary for the Arduino Nano 33 IoT device to process the data received from the Myo Armband device and perform a classification of the detected symbols using a Support Vector Machine (SVM) classifier.
- Representation of the detected symbols.

II. MYO ARMBAND

The Myo Armband device is a bracelet that, using EMG sensors, measures the electrical activity of the muscles to detect gestures made by the user's hands. By using its 9-axis Inertial Measurement Unit (IMU), it can detect movement, orientation, and rotation of the forearm. In the case of this project, the raw data of the EMG and IMU sensors will be obtained, to later process them

Thanks to Myo's software development kit (SDK), together with its BLE specification, which was made public in a header file format in C ++ language, it is possible for developers to access, from another device, through a BLE communication, to the raw data (in RAW) transmitted by the Myo device.

EMG sensors monitor the electrical signals that circulate under the skin, produced by the muscles, being able to detect any small electrical activation that they produce. The Myo device extracts data from these sensors at a sample rate of 200 Hz. Since it is a band with surface sensors, they cannot identify exact muscle movements. Together, however, the eight components can be used to identify certain finger gestures. Figure 1 shows the position of the 8 EMG sensors as well as the device itself. On the other hand, the Myo bracelet can extract data from the IMU at a sample rate of 50 Hz.

Figure 1: Myo Armband device

III. LIBSVM

Among the most powerful and well-known free software tools for implementing SVM algorithms is LIBSVM. Developed by Chih-Jen Lin et al, this library offers the possibility of using different kernels when implementing SVM classifiers, giving the option of integrating the library directly into the user application or making use of the precompiled programs included [3].

Specifically, in this paper, a C-SVM classifier is used, where C is a regularization parameter for the penalty rate between negative predictions, improving the precision of the model. This parameter varies between 0 and infinity, and the smaller its value, a high bias and low variance is achieved, which means that the classifier gives good results with poorly classified data points. If, on the other hand, the value of C is higher, a low bias and a high variance are achieved since it tries to avoid misclassified points. When working with a C-SVM classifier with the RBF kernel function, the gamma parameter is also available, which defines the extent of the influence of a single training example. To obtain the optimal C and gamma values, a cross-validation technique can be used, however, in this paper we start from the default value, with C = 100 and gamma = 0.0001.

IV. Platform development

To establish the BLE connection between the Myo Armband device and the Arduino Nano 33 IoT device, we start from the BLE specification of the Myo device, released by Thalmic Labs once they finished supporting the device. The specification is available on GitHub as a C ++ header file, named myohw.h. Thus, starting from this header file, a library has been created that contains the necessary functions to establish a BLE connection between the previously mentioned devices.

Once the connection was established, the data selection was carried out from the information available in the characteristics of the Myo device. Thus, the BLE characteristics to which the Arduino Nano 33 IoT device will subscribe will be EMGData0Char and IMUDataChar. Actually, there are four characteristics to obtain data from EMG sensors, since these sensors have a sampling frequency of 200 Hz,and cannot send enough data through a single characteristic. Therefore, the readings are divided into four characteristics, each containing two sequential samples.

However, the IMU has a sample rate of 50 Hz, so if you want to build data frames that include the EMG sensors and the IMU data, you must lower the sample rate of the EMG sensors. at 50 Hz. The way to do this is to subscribe to only one of the EMG data features and, from this, only get one of the two samples it contains.

The values of the EMG sensors are preprocessed based on a full wave rectification, which basically consists of calculating the absolute value of the value of each sensor, achieving that the readings of each symbol are distributed in a smaller area of space of characteristics, allowing the SVM classifier to construct better decision limits. EMG signals have both positive and negative components, but what measures muscle activation is the amplitude of the signal. Table 1 shows the parameters considered for the development of this initial platform.

BLE Characteristic	Info
EMGData0Char	8 EMG sensors
IMUDataChar	Roll, pitch, yaw
	Acelerometer X, Y, Z
	Gyroscope X, Y, Z

Table 1: Extracted parameters

To verify the final platform, 8 letters of the alphabet have been used, following the relationship shown in Table 2. For each symbol, 50 samples are obtained, of which 45 become part of the training dataset and the remaining 5 to the test dataset.

Inserted character	Assigned letter
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

After the samples are taken, the dataset is entered into the LIBSVM svm-train tool, which generates a model. We pass this model through the ArduinoSVM tool that generates the support vectors and the necessary methods to integrate the prediction function in our program.

Once the model is integrated into our platform, the program is ready, presenting two options to the user through a menu: obtaining samples or predicting a sample, as shown in Figure 2.

```
--------------------------------------------------------
MENU: Select an option
  1: Capture sample
  2: Predict sample
--------------------------------------------------------
Enter the symbol ID you want to capture:
0 1:17.89 2:-46.28 3:-6.23 4:0.68 5:0.18 6:0.56 7:3.25 8:5
--------------------------------------------------------
```

Figure 2: Main menu

V. RESULTS AND CONCLUSIONS

In this section the results of each classification realized through the development is presented. The main evaluation criteria for classifiers is through the accuracy level, which represents the total number of correct classifications made

Once the previous model had been generated, its effectiveness was tested by evaluating the test dataset, using the svm-predict command. Thus, Figure 3 shows the result of the prediction, obtaining a 100% correct result.

```
>svm-predict test_s.txt initialModel results.txt
Accuracy = 100% (40/40) (classification)
```

Figure 3: Dataset testing results

Verified the effectiveness of the model and having it integrated into the final platform, the prediction of samples from the developed program is tested. To do this, select the second menu option, and when it is ready, representing a symbol, press any key to extract a sample and predict the class to which it belongs, as shown in Figure 4.

```
--------------------------------------------------------
MENU: Select an option
  1: Capture sample
  2: Predict sample
--------------------------------------------------------
Press any key to start sample prediction...
Captured sample:
1:-22.63 2:14.70 3:-2.63 4:-0.21 5:-0.28 6:0.79 7:3.44 8:5.37 9:3.
The predicted symbol is: 0
--------------------------------------------------------
```

Figure 4: Example of sample prediction

According to the results obtained in each experiment, it can be established that this work has achieved the main goal of creating and developing an interactive platform, based on the Myo Armband device, which can translate from the Spanish sign language alphabet into written text.

REFERENCES

[1] Myo Armband (2019). Myo Armband. Recuperado de: https://support.getmyo.com/hc/en-us/articles/202648103-Myo-Gesture-Control-Armband-tech-specs.

[2] Nano 33 IoT (2021). Arduino. Recuperado de: https://docs.arduino.cc/hardware/nano-33-iot.

[3] Chang C, Lin C. “LIBSVM: A library for support vector machines”. National Taiwan University. 2011. [Online] Available: https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf [Último acceso: 10-Abril-2021].