algorithms

MDPI

Article

Electromyography Gesture Model Classifier for Fault-Tolerant-Embedded Devices by Means of Partial Least Square Class Modelling Error Correcting Output Codes (PLS-ECOC)

Pablo Sarabia [1], Alvaro Araujo [1], Luis Antonio Sarabia [2] and María de la Cruz Ortiz [3,*]

[1] B105 Electronic Systems Lab, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2] Department Mathematics and Computation, Faculty Sciences, University of Burgos, Pza. Misael Bañuelos s/n, 09001 Burgos, Spain
[3] Department Chemistry, Faculty Sciences, University of Burgos, Pza. Misael Bañuelos s/n, 09001 Burgos, Spain
* Correspondence: mcortiz@ubu.es; Tel.: +34-947259571

Abstract: Surface electromyography (sEMG) plays a crucial role in several applications, such as for prosthetic controls, human–machine interfaces (HMI), rehabilitation, and disease diagnosis. These applications are usually occurring in real-time, so the classifier tends to run on a wearable device. This edge processing paradigm imposes strict requirements on the complexity classifier. To date, research on hand gesture recognition (GR) based on sEMG uses discriminant classifiers, such as support vector machines and neural networks. These classifiers can achieve good precision; they cannot detect when an error in classification has happened. This paper proposes a novel hand gesture multiclass model based on partial least square (PLS) class modelling that uses an encoding matrix called error correcting output codes (ECOC). A dataset of eight different gestures was classified using this method where all errors were detected, proving the feasibility of PLS-ECOC as a fault-tolerant classifier. Considering the PLS-ECOC model as a classifier, its accuracy, precision, and F1 are 87.5, 91.87, and 86.34%, respectively, similar to those obtained by other authors. The strength of our work lies in the extra information provided by the PLS-ECOC that allows the application to be fault tolerant while keeping a small-size model and low complexity, making it suitable for embedded real-time classification.

Keywords: electromyography; fault tolerant; gesture recognition; wearable; edge computing; sensitivity; specificity; class modelling; partial least squares; error correcting output; diagonal modified confusion entropy

check for updates

Citation: Sarabia, P.; Araujo, A.; Sarabia, L.A.; Ortiz, M.d.l.C. Electromyography Gesture Model Classifier for Fault-Tolerant-Embedded Devices by Means of Partial Least Square Class Modelling Error Correcting Output Codes (PLS-ECOC). *Algorithms* **2023**, *16*, 149. https://doi.org/10.3390/a16030149

Academic Editor: Maryam Ravan

Received: 30 January 2023
Revised: 2 March 2023
Accepted: 5 March 2023
Published: 7 March 2023

1. Introduction

The growing interest in the application of gesture recognition (GR) using surface electromyography (sEMG) signals has been evident in recent years. A search on the SCOPUS database using the keywords (“Electromyography” or “EMG”) and “Gesture Recognition” yielded 884 references dating back to 1993. This is a highly interdisciplinary field, with the majority of papers falling under the categories of “Computer Science” and “Engineering” with 637 and 557 references, respectively, with 351 being common to both. Other areas with a significant number of references include “Mathematics,” “Physics and Astronomy,” and “Medicine” with 169, 144, and 124 references, respectively. Of those, none, 18, and 9 do not belong to either “Computer Science” or “Engineering”.

There are numerous medical applications for this technology for both people with disabilities and healthy individuals, as discussed in [1]. EMG signals have also been used in the design [2] and control of prostheses [3]. Applications in robot therapy were presented in [4], and [5] describes an elbow rehabilitation system using a remote-controlled intelligent robot. The use of EMG signals for disease diagnosis is an area with many applications, including the progression of primary lateral sclerosis [6], dysphagia [7], patellofemoral pain syndrome [8], neuromuscular disorders [9], and the evaluation of mobility loss in patients with musculoskeletal disorders [10].

Some of the uses of sEMG-based GR mentioned before can be grouped in human–machine interface (HMI). sEMG HMI allows the user to control or command other systems, such as wheelchairs, prosthetics, drones, robot arms, or exoskeletons. This use has been extensively studied by specific reviews on the topic. These reviews showed the necessity of low-complexity classifiers [11–14], which translates into reduced preprocessing, lower latency, and lower power consumption. These characteristics are critical when talking about wearable devices, as shown in review [15].

One of the great challenges facing GR systems is their use in wearable devices. A wearable device is a type of embedded system that is designed to be worn on the body, such as smartwatches, fitness trackers, and clothing embedded with sensors. These devices have several advantages, including convenience, portability, and personalization, but also have some limitations, such as limited battery life, privacy, security concerns, and limited computational power. The processing that takes place at the end devices is what is known as edge computing.

Edge computing has several advantages, such as privacy, given that no data is uploaded to a third-party cloud server; lower and reliable latency, as data are not required to be uploaded, and the latency does not depend on the network availability; and lower cost or simplicity because the system does not require a modem or connection to a network. On the other hand, the resources are limited by the embedded systems, and the use of batteries limits the use of more complex algorithms, such as complex neural networks (NNs) [16]. Therefore, this leaves room for simpler lineal models, such as the one presented in this work.

To ensure continuity of operation, a fault-tolerant system can be implemented, which allows the system to continue functioning correctly in the presence of hardware or software faults by detecting, diagnosing, and recovering from errors in real-time. This is achieved by using techniques such as redundancy, error detection and correction, diversity, and isolation. They can be combined to accomplish different levels of fault tolerance depending on the specific requirements of the system and application. As wearables are becoming more and more important in different fields, their fault-tolerance feature is increasingly important in many critical systems, such as aerospace, military, medical, and industrial systems, where a *failure* could have serious consequences. In these applications, the lack of robustness could lead to material or even personal damage.

The contribution of the proposed method is its capability of detecting errors by using a gesture multiclass model based on partial least square (PLS) class modelling with the use of an encoding matrix called error correcting output codes (ECOC) while keeping a small size model and low complexity, therefore, maintaining its suitability for real-time classification on a wearable device. To achieve fault tolerance, error detection was implemented. An *error* is defined as a deviation from the expected or correct behaviour of the system, in this case, the difference between the gesture made and the obtained label. In the context of HMI, an *error* can lead to a wheelchair driving into an obstacle because of a gesture misclassification or a drone crashing because the GR classifier confused left withright. In the proposed class-modelling approach, if an *error* is detected, it can take action, such as asking the user to reposition the electrodes or to repeat the gesture, preventing the system from executing any incorrect actions based on the sEMG input signal. Latency and robustness are critical in real-time classification [17]. This work also addresses the unresolved issue of automating the obtention of useful features because there are dozens of them that are usually chosen by hand during the design of the classifier and have a great impact on the outcome of the classifier.

This paper is organized as follows: Following this introduction, state-of-the-art approaches are discussed in Section 2, and the elements of the problem are described in Section 3. This includes the notation and experimental procedure to obtain the EMG signals (Section 3.1) and a description of the structural elements of the PLS-ECOC procedure

(Sections 3.2–3.8). The results are shown in Section 4, and a discussion of them in Section 5 shows the advantages of modelling the gesture and the accuracy of gesture classification, comparing it with those found in several papers. This paper ends with conclusions and several references.

2. State-of-the-Art Approaches

In gesture recognition using sEMG signals, computational techniques from machine learning (ML) are employed to assign a recorded signal to a gesture class. Specifically, a matrix $\mathbf{X}$ of sEMG signals with N rows corresponds to N gestures grouped into K classes $(C_1, \ldots, C_K)$. There are two general approaches to addressing this issue: discriminant techniques and class modelling.

The first one is a purely discriminative approach. In the training phase, the ML technique constructs a mathematical model and decision rule to assign each gesture class a subset $\wp_k$, $k = 1, \ldots, K$ of the signal space in such a way that they are disjoint, and their union is the total. As a result, in the prediction phase, a new signal will necessarily be assigned to one of the classes and only one. These types of techniques are commonly known in pattern recognition as discriminant or classification techniques.

The performance of a discriminant model is measured with the expected classification accuracy, namely, the percentage of correct decisions in prediction or variations of this metric. Examples of common purely discriminant methods include linear or quadratic discriminant analysis (LDA, QDA [18]), regularized discriminant analysis (RDA) [19], partial least squares discriminant analysis (PLS-DA) [20], classification and regression trees (CART) [21], or support vector machines (SVM) originally developed for two-class classification [22], generalized to the multiclass situation [23], and the case of unlabelled data [24]. Neural networks (NNs) have also been widely developed and improved since their inception [25], with the backpropagation algorithm [26] allowing for computational feasibility. NNs offer flexibility, allowing them to adapt their structure to improve classification accuracy. For example, the convolutional neural networks (CNNs) proposed in reference [27] have been applied to handwritten digit classification.

The second approach for GR using EMG signals is known as the class-modelling technique. It also builds K subsets, named K-class models, which are represented by $\wp_k$, $k = 1, \ldots, K$, but they are not necessarily disjoints, and their union may not cover the entire signal space. As a result, an object(in this case an sEMG signal) can belong to one or more classes (gestures) or none at all. To evaluate the performance of the class-modelling technique, the sensitivity and specificity of each class model are measured. In this context, the sensitivity of a class model refers to its ability to correctly identify its corresponding gesture (usually measured as the rate of correctly classified objects within the class model, $\wp_K$), while its specificity refers to its ability to correctly reject objects that do not belong to the class model (measured as the rate of correctly classified objects outside the class model).

Among the class-modelling methods, some stand out: Soft independent models of class analogy (SIMCA) [28] have been widely used for product authentication, quality assurance, fraud detection, unequal class models or unequal dispersed classes (UNEQ) [29], or an adaptation of SVM [30]. Two-class modelling using PLS can be seen in reference [31] and for K-classes in [32].

The advantage over the usual class discriminant classifiers used in sEMG gesture recognition problems and the proposed class modelling approach is that it enables the system to be fault tolerant. This is because the class modelling approach does not limit the answer to a unique label. Another advantage is the extra information provided by the sensitivity and specificity of each class model that allows the engineer to adapt the design by including extra channels to detect the difference between the confused gestures or to remove certain gestures if they are being problematic prior to the deployment of the system. This possibility in discriminant classifiers would not be available, and the only possible action would be to add more data to the dataset with the objective of improving its accuracy.

In the field of gesture recognition using EMG signals, class modelling has never been used. In the literature reviewed, only gesture classification has been performed. Publications powered by pattern recognition methods have been reviewed in references [33–35]. The most commonly used are SVM [36,37] and NNs [38]. After the signal acquisition, among all the modules that compose a gesture recognition system, perhaps the most important and critical ones are feature extraction and the classification algorithm. In this regard, the algorithms and results of these references are summarized below.

The review by Nazmi et al. [35] of works from 1999 to 2013 shows that the classifiers used are linear discriminant analysis, SVM, adaptive network-based fuzzy inference system, NNs, and fuzzy logic. The range of accuracy is [73.00, 98.87] with a mean of 90.60% and a standard deviation of 7.06%. Moreover, Jaramillo-Yáñez et al.'s review [33] from 2013 to 2019 showed that the classifiers used are support vector machines, feedforward neural networks, linear discriminant analysis, convolutional neural networks with several variants, k-nearest neighbours coupled with several other procedures binary tree-support vector machines, vector autoregressive hierarchical hidden Markov models, Gaussian mixture models and hidden Markov models, quadratic discriminant analysis, fuzzy logic, recurrent neural networks, generalized regression neural networks, and one vs. one classifier. Among them, the most commonly used machine learning algorithms are support vector machines, feedforward neural networks, and linear discriminant analysis. The accuracy of the reviewed publications ranges from 71.00% to 99.78% with a mean of 91.57% and a standard deviation of 6.35%. Similar accuracy is reported (63.74% to 99.23% with a mean of 89% and a standard deviation of 10.11%) by Dhumal and Sharma's [34] revision between 2017 and 2021. A more recent comparison evaluated five machine learning techniques in classifying daily gestures [10]. It reported an accuracy ranging from 47.7% to 83.6% with a mean value of 66.54% and a standard deviation of 14.95%. This revision included SVN, a random forest, a decision tree-based algorithm, a convolutional neural network, and a recurrent neural network.

The analysis of the abovementioned reviews also shows that in order to capture and describe the complexity and variability of sEMG signals, it is considered necessary to handle "deep learning" methods. This trend can be observed in related fields such as gesture recognition based on electroencephalogram (EEG) signals. In Zhang et al.'s revision [38], a taxonomy of deep learning models was made according to their functionality in different scenarios of brain activity. After a very extensive description of applications, the authors indicated that it is still very challenging to produce classification results in real-time. In [39], GR performance was improved by using a graph sequence neural network applied to an HMI problem. In [40], six features were obtained using the correlation and canonical correlation analysis between filtered signals; after that, a selection was conducted. However, these advances have not yet been translated into their equivalent sEMG signals.

Phinyomark and Scheme [41] made a very systematic analysis of the use of deep learning for classifying sEMG signals and define two categories of computational procedures:

- Feature engineering. Under this heading, the methods to find the best combination of features in a specific problem are grouped. They are parallel computing procedures, either native or classics parallelized. Then, conventional classifiers, such as support vector machines, linear discriminant analysis, k-nearest neighbours, random forests, multilayer perceptron neural networks, etc., are applied.
- Feature learning. This heading includes methods with a special emphasis on "deep learning". In general, though, deep learning models can be roughly grouped into three main categories: unsupervised pretrained networks, convolutional neural networks, and recurrent neural networks. These three categories of models have already been used to analyse sEMG signals, as shown in the reviews of the preceding paragraphs.

The authors concluded their extensive review by saying: "A key challenge and impediment to the clinical deployment of deep learning methods is their high computational cost (i.e., long training times and high computational complexity). Because of the stringent power and size restrictions of prosthetic components, most devices are built using embedded systems".

On the other hand, the use of an SVM classifier for gestures in EEG and sEMG signals is still interesting, as shown by Quitadamo et al.'s review [14]. Up to nine different variants of SVM classifiers were reviewed in several applications. The review showed that the accuracy of the 76 publications revised ranged from 18.83% to 100% with a mean of 80.8% and a standard deviation of 13.32%. Taking into account only the sEMG GR publications, the accuracy increases to a range from 73% to 100% with a mean of 91.1% and a standard deviation of 6.38%. The review concludes that "SVMs result to be among the most versatile classifiers for pattern recognition . . . and furthermore resulted to be particularly suitable for online implementations".

Therefore, along with the trend to use deep learning, there is still interest in exploring the feasibility of class modelling with PLS-ECOC that simultaneously reduces the complexity associated with feature engineering and that associated with the deep learning approach. In addition, PLS-ECOC distinguishes between *failure* and *error*.

However, in all cases, the methods used are independent of the structure of **X**, the sEMG signals, and the encoding used to describe the classes, which is denoted by **Y**. In engineering, particularly in the field of signal transmission, redundancy has been used to distinguish, at the receiver, two signals despite having lost information during transmission. This idea has been used in this work to encode the gesture classes and construct the matrix **Y** by assigning each class a code (a vector of ones and minus ones) using an ECOC matrix. Little attention has been paid to the codification of classes in GR sEMG publications. Only ref. [14] presents the effect of two codifications (one vs. one and one vs. all) which have been proven to be inefficient [32].

The proposal in this work is to use PLS to find the linear relationship between the signals and the encoded classes, explaining the maximum variance in **X**, in **Y**, and the maximum correlation between them. PLS constructs *nvl* latent variables, linear combinations of **X** variables and **Y** variables under the condition of having the maximum correlation. In this way, PLS applies a reduction of the signal space but links to the classes of the gestures to be distinguished. This characteristic makes PLS an advance in the treatment of EMG signals. Since ancient times [42], the need to reduce the dimension of the signal space has been recognized, in most cases, as a second stage after obtaining features [43]. In [44], using several classifiers, a comparison has been made between feature selection and dimensionality reduction, including PCA. However, in general, PCA is performed independently of the classifier and, therefore, of the classes to be modelled. In a review made, only one PLS application with sEMG signals has been found. It is devoted to model concurrent EEG and EMG data collected in a Parkinson's disease study [45], but it was not a classification of gestures.

The method, PLS-ECOC [32], provides the sensitivity and specificity matrix of the constructed model. This matrix is evaluated using diagonal modified confusion entropy (DMCEN) [46]. According to the classification of methods to preprocess sEMG signals, PLS-ECOC can be considered an embedded one because the construction of latent variables and the model of classes are built simultaneously in the same model [47].

It is also noteworthy that PLS-ECOC is robust to sEMG signals that differ from those used to train the class model since, in addition to the equations for class prediction, the model provides a closed enclosure in the sEMG signal space, the PLS-box, so that a signal outside of it will be declared an outlier.

3. Materials and Methods

This section presents the methodology followed to develop the hand gesture modelling system (HGM) based on surface electromyography signals (sEMG) and partial least squares multiresponse. It has been divided into eight subsections, each of them providing enough information to understand the method and the results section. A complete schematic of the operations followed to achieve HGM can be seen in Figure 1.

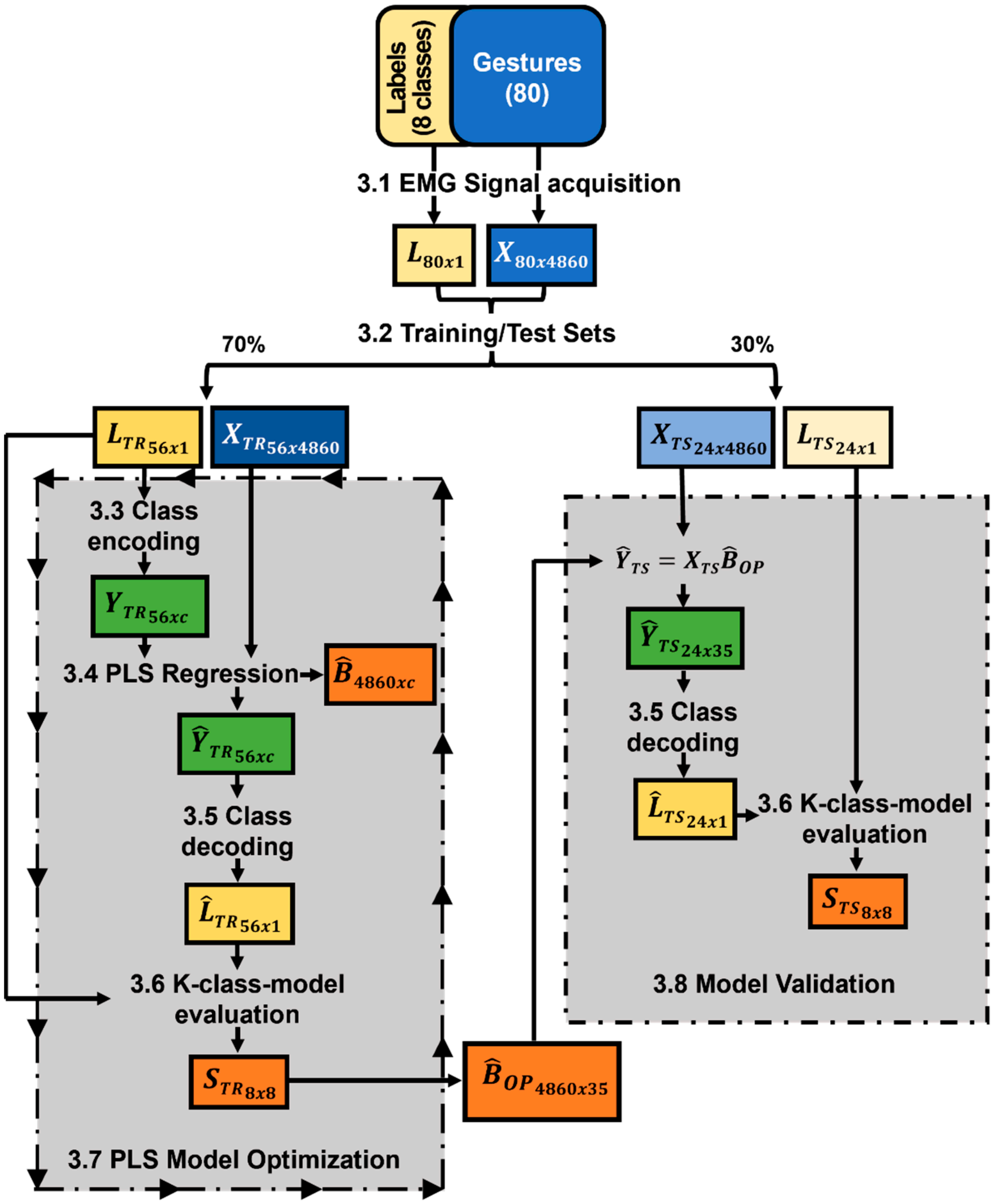


Figure 1. Schematic of the hand gesture modelling method based on sEMG and PLS-ECOC. Steps 3.1 to 3.8 are addressed by the subsections of this section. The arrows box indicates iteration of the included steps in it. "*c*" stands for the number of binary learners in class codification and changes in each iteration. See the text above for an explanation of the other details.

This figure is meant to be a guide to the reader showing the processing and flow of data, and it follows the same notation and subsection names and numbering used in the rest of this paper. In it, each rectangle includes the numerical matrix resulting from applying the operation indicated in the access arrow and describes in detail in the subsection that bears the name and numbering indicated. The second subscript of the matrices indicates the dimension of each of them.

For example, the vector of the class labels, $\mathbf{L}$, and the signals sEMG, $\mathbf{X}$, are divided according to the procedure described in Section 3.2 into the data to be used for training, $\mathbf{L}_{TR}$ and $\mathbf{X}_{TR}$, and for prediction, $\mathbf{L}_{TS}$ and $\mathbf{X}_{TS}$, respectively.

Applying the coding procedure explained in Section 3.3 returns $\mathbf{Y}_{TR}$, the matrix of the encoded response, which, together with $\mathbf{X}_{TR}$, allows the coefficients of the PLS regression, $\hat{\mathbf{B}}$, to be obtained. With this model, the calculated code matrix, $\hat{\mathbf{Y}}_{TR}$, is obtained, which is decoded into $\hat{\mathbf{L}}_{TR}$ to compare it with the vector of the labels $\mathbf{L}_{TR}$. Afterwards, the matrix of sensitivities and specificities, $\mathbf{S}_{TR}$, is obtained in training. This process is iterated according to the procedure described in Section 3.7 to obtain the optimal PLS model, $\hat{\mathbf{B}}_{OP}$. This model is applied to the test set, $\mathbf{X}_{TS}$, and the vector of labels in prediction, $\hat{\mathbf{L}}_{TS}$, is obtained. More details on the method can be consulted in [32].

The K-class model consists of K models, one for each class modelled. Class models are built and validated as a whole. Consequently, if a new class was added, the model would need to be built and validated again, obtaining a new (K + 1)-class model.

I sEMG signals were recorded into K class, $C_1, C_2, \ldots, C_K$, each of them associated with each gesture type. The number of signals recorded for the j-th class is denoted by I_j, so $I = \sum_{j=1}^{K} I_j$.

The result of applying the K-class model is summarized in the confusion matrix **N**

$$\mathbf{N} = \begin{pmatrix} n_{11} & n_{12} & \cdots & n_{1m} & \cdots & n_{1K} \\ n_{21} & n_{22} & \cdots & n_{2m} & \cdots & n_{2K} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ n_{j1} & n_{j2} & \cdots & n_{jm} & \cdots & n_{jK} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ n_{K1} & n_{K2} & \cdots & n_{Km} & \cdots & n_{KK} \end{pmatrix}, \quad (1)$$

where usual gesture classification algorithms verify the following: n_{jm} is the number of gestures belonging to class C_j which are inside the class model built for class C_m.

- $I_j = \sum_{m=1}^{K} n_{jm}$, which means that the sum of the j-row elements of matrix **N,** Equation (1), is equal to the number of elements in the j-th class.
- $I = \sum_{j=1,m=1}^{K} n_{jm}$, which means that the sum of all the elements of matrix **N,** Equation (1), is equal to the total I signals recorded.

However, when using class modelling, each signal can be assigned to no class or more than one class. So, one or both of the above equations might not be met. This distinction between usual classifiers and the class modelling presented here will prove to be critical.

The relative frequency matrix, **F**, Equation (2), can be obtained from Equation (1) as follows:

$$\mathbf{F} = \left(f_{jm}\right) = \left(n_{jm}/I_j\right),\ j = 1,\ \ldots,\ K,\ m = 1, \ldots, K. \quad (2)$$

As a consequence of the matrix **N** structure and the use of the class modelling algorithm, each of the rows of the **F** matrix, Equation (2), may not add 1, unlike a classification method.

F matrix has a relevant statistical meaning. The assignation of an sEMG signal to a class is applying a hypothesis test. Given K classes, this decision is a family/set of $K(K-1)$ hypothesis tests such as those in Equation (3). It is composed of $K-1$ tests for each of the K classes. The null Hypothesis H_0 is the same for each $j = 1, \ldots, K$, but H_1, the alternative hypothesis, is different for each one.

$$\begin{array}{l} \mathrm{H}_0\text{: EMG signal belongs to gesture } C_j. \\ \mathrm{H}_1\text{: EMG signal belongs to gesture } C_m,\ m = 1, \ldots, K,\ m \neq j. \end{array} \quad (3)$$

TEST matrix, Equation (4), symbolically summarizes these tests. Each diagonal term contains the null hypothesis (H_0), and the rest of the terms of the matrix contain the corresponding alternative hypothesis (H_1).

$$\mathbf{TEST} = \begin{pmatrix} \mathrm{H}_0 : \mathrm{C}_1 & \mathrm{H}_1 : \mathrm{C}_1 & \cdots & \mathrm{H}_1 : \mathrm{C}_1 & \cdots & \mathrm{H}_1 : \mathrm{C}_1 \\ \mathrm{H}_1 : \mathrm{C}_2 & \mathrm{H}_0 : \mathrm{C}_2 & \cdots & \mathrm{H}_1 : \mathrm{C}_2 & \cdots & \mathrm{H}_1 : \mathrm{C}_2 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \mathrm{H}_1 : \mathrm{C}_j & \mathrm{H}_1 : \mathrm{C}_j & \cdots & \mathrm{H}_0 : \mathrm{C}_j & \cdots & \mathrm{H}_1 : \mathrm{C}_j \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \mathrm{H}_1 : \mathrm{C}_K & \mathrm{H}_1 : \mathrm{C}_K & \cdots & \mathrm{H}_1 : \mathrm{C}_K & \cdots & \mathrm{H}_0 : \mathrm{C}_K \end{pmatrix}. \tag{4}$$

Hypothesis test theory defines α_j as the significance level and $1 - \beta_{mj}$ as the power of each test. These parameters are related to sensitivity and specificity of the test. Note that notation on sensitivity and power of the test vary and can be confusing, see references [35,36,48]; that is why they are defined below.

The column j-th of **TEST,** Equation (4), corresponds to the $K - 1$ hypothesis tests of C_j class. Using the **F** matrix, it is possible to calculate the significance level and power of the test and to define the sensitivity and specificity:

- Sensitivity, $(1 - \alpha_j) = f_{jj}$ of **F**, Equation (2). This means the probability of correctly assigning an sEMG signal of class C_j to the class model of C_j annotated as s_{jj} in Equation (5).
- Specificity, $\left(1 - \beta_{mj}\right) = \left(1 - f_{mj}\right)$ *when* $m \neq j$ corresponds to the rest of the elements of the columns of **F,** Equation (2). This means the probability of correctly not assigning an sEMG signal of class C_m to the class model of C_j annotated as s_{jm} in Equation (5).

To sum up, sensitivity is the proportion of gestures that are assigned correctly to the modelled class, and specificity is the proportion of gestures correctly rejected from the modelled class.

With this notation, matrix **F** is transformed into matrix **S** of sensitivities and specificities, Equation (5), that characterizes the performance of the K-class model.

$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1j} & \cdots & s_{1K} \\ s_{21} & s_{22} & \cdots & s_{2j} & \cdots & s_{2K} \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{j1} & s_{j2} & \cdots & s_{jj} & \cdots & s_{jK} \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{K1} & s_{K2} & \cdots & s_{Kj} & \cdots & s_{KK} \end{pmatrix} = \begin{pmatrix} f_{11} & 1-f_{12} & \cdots & 1-f_{1j} & \cdots & 1-f_{1K} \\ 1-f_{21} & f_{22} & \cdots & 1-f_{2j} & \cdots & 1-f_{2K} \\ \vdots & \vdots & & \vdots & & \vdots \\ 1-f_{j1} & 1-f_{j2} & \cdots & f_{jj} & \cdots & 1-f_{jK} \\ \vdots & \vdots & & \vdots & & \vdots \\ 1-f_{K1} & 1-f_{K2} & \cdots & 1-f_{Kj} & \cdots & f_{KK} \end{pmatrix}. \tag{5}$$

3.1. Acquisition of EMG Data

The acquisition setup is composed of an embedded device (Section 3.1.1) and a computer. The methodology for the acquisition is described in Section 3.1.2. Data description is included in Section 3.1.3. A detailed description, including code, can be found in the previous work [37] of the authors.

3.1.1. Hardware Description

The data were acquired using an embedded system, Figure 2, based on a 4-channel 24-bit ADC (AD7124-4) connected by serial peripheral interface (SPI) to a microcontroller (MCU) STM32L486RG microcontroller and running YetiOS [49] as the operating system. The sampling rate was set to 500 Hz with a hardware lowpass filter with a cutoff frequency of 130 Hz. The signals were transmitted via serial to the computer where they were processed using MATLAB [50].

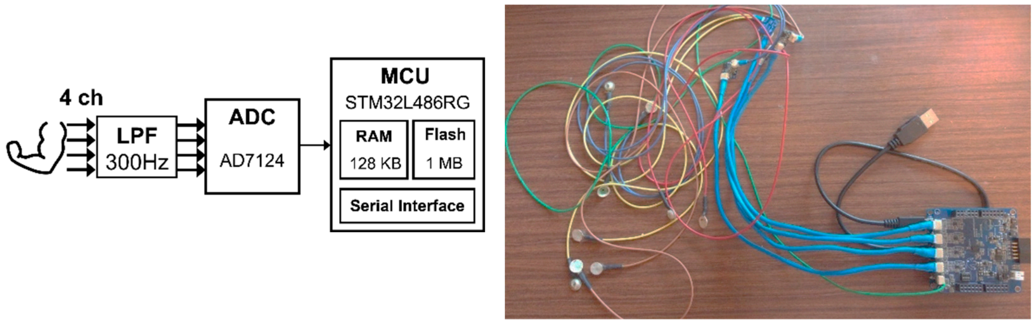
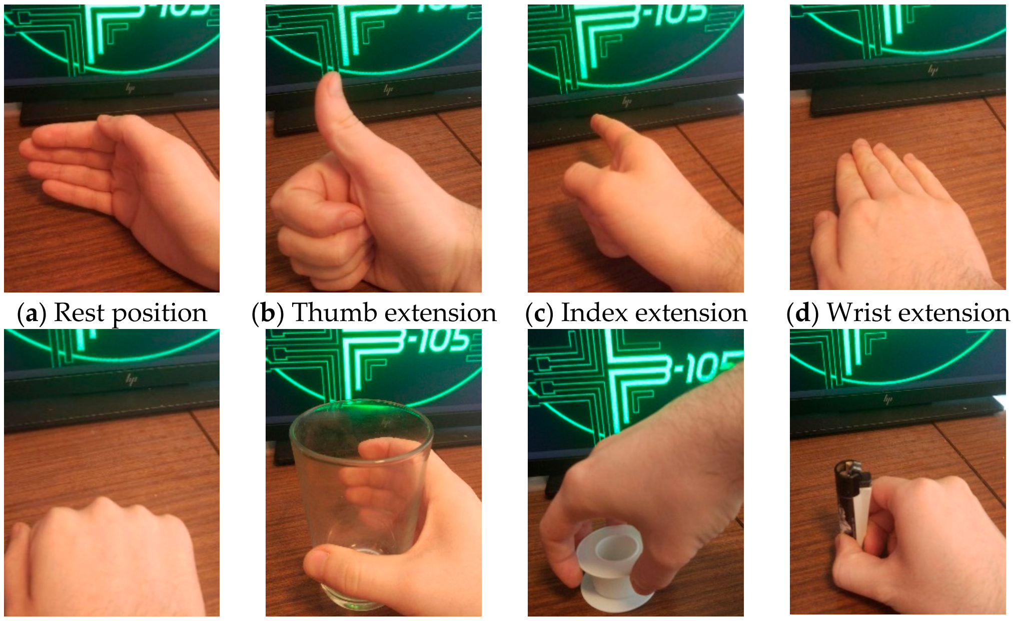


Figure 2. Hardware acquisition system, left block design, and right picture of the system.

3.1.2. Methodology for Acquisition

The study conducted within this work does not require ethical approval since no part of the experiments has any possible effect on the subjects' bodies, harmful or otherwise. The system developed only gathers user motion data through EMG measurements. Moreover, the data were recorded and used in this publication under informed consent of the subject. The subject was a 23-year-old male.

Figure 3 shows the 8 gestures recorded that emulate representative prosthetic hand movements. These particular movements are the same as those used in one of the most employed databases in the field, Ninapro DB6 [51].


(**a**) Rest position (**b**) Thumb extension (**c**) Index extension (**d**) Wrist extension

(**e**) Wrist contraction (**f**) Circular grasp (**g**) Clamp grasp (**h**) Precision tripod grasp

Figure 3. Eight different gestures recorded. The classes $C_1, C_2, \ldots, C_8$ of the text are the pictures (**a**–**h**), respectively.

Electrodes are placed on the skin of the forearm of the subject after applying conductive gel. Then, the subject was asked to repeat the gestures shown in Figure 3 10 times, each of them for 3 s, and they were subsequently stored in the computer, processed, and saved in matfiles using MATLAB [50].

3.1.3. Data Description

The result of the acquisition is a vector of around 1620 voltage measurements for each channel. Channel 3 was faulty during the acquisition, so, it was removed. Each of these vectors is concatenated, leading to a vector of 4860 coordinates. A total of 80 gestures (10 repetitions of each class) were acquired, producing a matrix **X** of size 80 × 4860 that will be used in this paper. This dataset is fully available in [52].

3.2. Training and Test Sets to Evaluate the Prediction Abilitty of the PLS2-CM Model

The evaluation of prediction capability can be done through a test set independent of the training set or by cross-validation (CV), also named k-fold procedure, that includes leave-one-out cross-validation. Despite its popularity, CV use has been criticized [53], and it has recently been formally proved that it is not a correct procedure to estimate prediction capability [54]. It has therefore been decided to divide dataset **X** into two subsets: $\mathbf{X}_{TR}$ for training and $\mathbf{X}_{TS}$, the test set, for the evaluation of predictive capability. The Kennard-Stone algorithm [55], the most relevant for this task, was used by means of 'kennardstone' function of PLStoolbox [56]. The method initially selects the pair of sEMG signals with the largest distance and ranked as most representative. Then, in each following step, the remaining samples with the greatest distance from the already selected samples are chosen and added to the bottom of the previous rank list. This procedure is repeated until a predefined number of gestures had been chosen and ranked in each class. The method assures a uniform distribution of the sEMG signals selected, and it also includes boundary gestures of the class. Remaining gestures are assigned to test subset. In each class, 70% has been assigned for training and 30% for test subset. Resultantly, training matrix $\mathbf{X}_{TR}$ has a size of 56 × 4860 and 24 × 4860 for prediction $\mathbf{X}_{TS}$.

3.3. Class Encoding

Error correcting code (ECOC) is the method of choice to encode the K classes by means of c binary learners. This codification is defined by the matrix **M** of K rows and c columns whose elements are $+1$ or -1. A simple example, Equation (6), of **M** matrix was built to encode 4 classes using a code of length $c = 7$.

$$\mathbf{M} = \begin{pmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{pmatrix}. \tag{6}$$

Each row of **M** represents the class code, so in Equation (6), the objects of the third class will be assigned the code or vector $\begin{pmatrix} -1, & -1, & +1, & +1, & -1, & -1, & +1 \end{pmatrix}$. The same can be performed for the other three classes. Using these codes, it is possible to obtain the matrix **Y** formed by the class codes of all the objects. Each column of **M** is a binary learner that can compare between two groups of classes or superclasses $\mathbb{N}$ and $\mathbb{R}$. Continuing with the example of Equation (6), the first column (binary learner) divides the classes into two superclasses: the first one comprises the class $\mathbb{N} = C_1$ and left the rest in the second, $\mathbb{R} = C_2 \cup C_3 \cup C_4$. Another example could be the second binary learner that groups C_1 and C_4 in the first superclass, $\mathbb{N} = C_1 \cup C_4$, and C_2 and C_3 in the second, $\mathbb{R} = C_2 \cup C_3$.

A series of changes that do not affect the codification can be deduced from the aforementioned description of **M**: permutations of columns, sign change in the whole column, and whole columns of -1 or $+1$ can be discarded. Therefore, the maximum length to encode K classes is $c_{max} = 2^{K-1} - 1$. For example, in the case of Equation (6), with $K = 4$ classes, $c_{max} = 7$.

Selecting the adequate binary learners and their number c for a specific problem is challenging. As evaluated with several datasets in reference [46] and given that our dataset has $K = 8$ classes, the proposed approach was to use function 'designecoc' (Statistics and Machine Learning Toolbox of MATLAB [50]). This function assigns positive or negative elements with equal probability for each element in **M** and uses a random generator to obtain the encoding, so the size of c is a random variable. However, its mean can be estimated by $10 \log_2(K)$, which in our case, means $c \cong 30$. Applying this codification to the training label matrix $\mathbf{L}_{TR}$ with dimensions (56×1) will result in matrix $\mathbf{Y}_{TR}$ with dimensions $(56 \times c) \cong (56 \times 30)$ containing the encoding for each gesture, see Figure 1.

3.4. Partial Least Squares Regression

Partial least squares regression (PLS) is a statistical tool that is very extended at the moment. It was introduced by Wold [57] in the 1970s and was successively developed [58,59]. This tool was used to relate $\mathbf{X}_{TR}$ with $\mathbf{Y}_{TR}$, see Figure 1.

PLS sequentially builds pairwise lineal combinations of variables, one from $\mathbf{X}$ and one from $\mathbf{Y}$, defined by the $\mathbf{r}$ and $\mathbf{q}$ vectors, respectively, that solve

$$\begin{gathered} max_{\mathbf{r},\mathbf{q}}\{var(\mathbf{X}_{TR}\mathbf{r})[corr(\mathbf{X}_{TR}\mathbf{r},\mathbf{Y}_{TR}\mathbf{q})]^2 var(\mathbf{Y}_{TR}\mathbf{q})\} = max_{\mathbf{r},\mathbf{q}}\{[(\mathbf{X}_{TR}\mathbf{r})^{\mathrm{T}}\mathbf{Y}_{\mathrm{TR}}\mathbf{q}]^2\} \\ \text{subject to } \|\mathbf{r}\| = \|\mathbf{q}\| = 1, \end{gathered} \tag{7}$$

where *var* and *corr* stand for variance and correlation, respectively, of the new variables $\mathbf{X}_{TR}\mathbf{r}$ and $\mathbf{Y}_{TR}\mathbf{q}$ that are named latent variables of PLS model.

Maximization of the product $var\ (\mathbf{X}_{TR}\mathbf{r})[corr(\mathbf{X}_{TR}\mathbf{r},\mathbf{Y}_{TR}\mathbf{q})]^2 var(\mathbf{Y}_{TR}\mathbf{q})$ tends to look for directions of large variance in both **X**- and **Y**-spaces (with more information), avoiding those of small variance (probably noise). In addition, the criterion includes the term $corr\ (\mathbf{X}_{TR}\mathbf{r},\mathbf{Y}_{TR}\mathbf{q})$ that helps in avoiding directions in the signal space with small correlation with the class codification.

PLS is particularly useful when the variables $\mathbf{X}_{TR}$ and/or $\mathbf{Y}_{TR}$ are very correlated or are colinear. In our case, $\mathbf{X}_{TR}$, the sEMG signals, present both characteristics: They are not independent, and they do not define 4860 dimensions, and neither do $\mathbf{Y}_{TR}$, the gesture codes, define approximately 30 dimensions. PLS is very efficient in finding the subjacent structure in the data by using a reduced number of latent variables, *nvl*. These latent variables define a subspace of $\mathbf{X}$ of *nvl* dimensions that explains the variability in the object codes, $\mathbf{Y}_{TR}$, that, in our problem, is the variability between gesture classes. The value of *nvl* is adjusted between one and K to obtain the best response using the method described in Section 3.6.

This step of procedure returns a linear regression model described by $4860 \times c$ coefficients along with the PLS-box [60], which is a closed region in the space of the sEMG signals. The complement set to the PLS-box identifies signals to which the model cannot be applied because they are significantly different from the training matrix. PLS-box is defined by two critical values fixed in this work to a confidence level of 0.95. As all the responses are fitted together, PLS provides a common PLS-box for the c binary learners.

The result of applying the model to sEMG signals ($\mathbf{X}_{TR}$ or $\mathbf{X}_{TS}$) will return, respectively, the predicted code ($\widehat{\mathbf{Y}}_{TR}$ and $\widehat{\mathbf{Y}}_{TS}$, respectively), see Figure 1. For this task, the function 'pls' of PLSToolbox [56] has been used.

3.5. Decoding, Class Assignation

Based on the encoding section, Section 3.3, for each binary learner, g_i, $i = 1, \ldots, c$ and its two superclasses $\mathbb{R}$ y $\mathbb{N}$, there are two sets of predicted values, $g_i(\mathbb{R})$ and $g_i(\mathbb{N})$, that should be close to -1 and 1, respectively. The distribution of these predicted values is calculated using the univariate kernel density [61]. Fixing the probabilities of γ_i and δ_i (that can be different for each g_i), the distributions previously fitted are used to compute critical values $CV_i(\mathbb{R})$ and $CV_i(\mathbb{N})$ and with the conditions established in Equation (8).

$$\begin{aligned} P\{\hat{y}_i \in\ g_i(\mathbb{R})\ |\hat{y}_i \le\ CV_i(\mathbb{R})\} = \gamma_i \\ P\{\hat{y}_i \in\ f_i(\mathbb{N})\ |\hat{y}_i \le\ CV_i(\mathbb{N})\} = \delta_i, \end{aligned} \tag{8}$$

where P stands for probability. Notice that the definitions in Equation (8) imply that γ_i would be a large value close to one, whereas δ_i would be close to zero. In this application, $\gamma_i = 0.99$ and $\delta_i = 0.01$ for all g_i, $i = 1, \ldots c$, have been fixed.

For each EMG signal in the PLS-box, if $\hat{y}_i \le\ CV_i(\mathbb{R})$, then the i-th coordinate on the decoding vector will be -1, and if $\hat{y}_i > CV_i(\mathbb{N})$, the coordinate will be +1. For each binary learner, the objects whose predicted values are between the corresponding critical values present two possibilities: (i) if $CV_i(\mathbb{R}) < CV_i(\mathbb{N})$, the signal will not be assigned, neither +1 nor -1, and (ii) when $CV_i(\mathbb{R}) > CV_i(\mathbb{N})$, the signal is assigned to both +1 and -1 (intersection). Finally, the signal is inside the i-th class model if the codeword of the i-th class in $\mathbf{M}$ is one of the decoding vectors related to it. To summarize, the sEMG signal could be assigned to none, one, or more classes or gestures. In our example, see Figure 1, the purpose is to obtain the $\hat{\mathbf{L}}_{TR}$ and $\hat{\mathbf{L}}_{TS}$ label matrix from the $\widehat{\mathbf{Y}}_{TR}$ and $\widehat{\mathbf{Y}}_{TS}$ predicted codes. However, based on the premise that any sample measured can only belong to one class, because it is not possible to perform two gestures at the same time, it is possible to obtain not valid results and *errors* when one sample belongs to none, two, or more classes. For this reason, predicted labels $\hat{\mathbf{L}}_{TR}$ and $\hat{\mathbf{L}}_{TS}$ will have one more possible class (K + 1) than the original (K) classes in label matrix $\mathbf{L}$.

3.6. K-Class Model Evaluation

Evaluation of a K-class model implies studying $K \times K$ values of its corresponding sensitivity $\mathbf{S}$ matrix. Any given problem with more than $K > 2$ makes it practically impossible to work with. In our case, it would be 64 values of sensitivity and specificity. Several metrics have been proposed to address this problem [62,63]. These metrics, in most of the cases, are obtained from the sensitivity of every individual class model, pair-wise specificities, efficiency or the total sensitivity, total specificity, and total efficiency of all class models, including convex combination of individual sensitivities and specificities. In particular, the mean of both is named accuracy in reference [48], but in [35], accuracy is the proportion of true assignations (positives or negatives). In general, these indexes are less sensible to changes in $\mathbf{S}$ matrix. For this reason, they are not well suited for a systematic comparison of class models, for example, in an optimization.

One alternative is to apply entropy concept as a measure of the order/information within the states of a system. A K-class model could be seen as a system whose states are the classes, and the information is provided by the sensitivities and specificities for each class model (K models in total). One K-class model will be better if it has less entropy according to Shannon's entropy definition. In other words, a better class model will provide more information and better sensitivities and specificities if it successfully assigns sEMG signals to the model of each class and rejects the ones that do not belong to it, thus reducing the uncertainty or entropy of the system. In Shannon's notation, followed in this paper, the entropy value will be 0 if all elements of $\mathbf{S}$ are ones, and it will be perfectly precise, leaving no room for uncertainty.

The development of this idea for confusion matrix can be found in references [64,65]. The first one proposes, for the first time, a measure of the order/information generated by a classification method, called confusion entropy (CEN), inspired by Shannon's entropy. The second solves a deficiency of CEN by proposing modified confusion entropy (MCEN).

MCEN was evaluated and generalized in [46], defining a new index, diagonal modified confusion entropy (DMCEN), for sensitivity and specificity to evaluate K-class models. It also introduces one modification to solve MCEN's almost unresponsiveness to differences in sensitivity by explicitly accounting for the contribution of sensitivity against specificity in the parameter w. This parameter can take values from zero to one; in our work, has was chosen $w = 0.5$.

A detailed description of DMCEN is out of the scope of this paper. Qualitatively, the idea is that for the computation of Shannon's entropy of a C_j class, is necessary to define the probability that the system is in class C_j in relation to another class C_m, for this it is considered as reference for all the decisions that involve both classes. In other words, to evaluate one K-class model, is necessary to take into account its whole $\mathbf{S}$ matrix.

In this work, DMCEN was calculated using an ad-hoc MATLAB code, available in reference [66], that calculates global DMCEN and for each class given an $\mathbf{S}$ matrix. DMCEN can take values from zero to one, and the lower DMCEN, the better the K-class model is.

3.7. PLS-ECOC Model Optimization

ECOC matrix, $\mathbf{M}$, has a random origin, as explained in Section 3.3, and is not related to $\mathbf{X}_{TR}$, but with $\mathbf{M}$**,** the response $\mathbf{Y}_{TR}$ is built. Therefore, the PLS model that relates $\mathbf{X}_{TR}$ and $\mathbf{Y}_{TR}$ depends on $\mathbf{M}$. To select the best $\mathbf{M}$ matrix, the steps from Section 3.3 to Section 3.6 were repeated 280 times, aiming to obtain the best $\mathbf{S}$ matrix. The PLS model associated with it is called PLS_{OP} and is the one that will be used in Section 3.8 for the evaluation with the test subset, $\mathbf{X}_{TS}$. This process is illustrated in Figure 1 as the left grey rectangle.

3.8. Model Validation

To validate the model, the calculated label matrix $\hat{\mathbf{L}}_{TS}$, DMCEN and $\mathbf{S}_{TS}$ was obtained using the model PLS_{OP} and test data $\mathbf{X}_{TS}$ as its input. Another interesting metric is the use of $\hat{\mathbf{L}}_{TS}$, extra class, the one assigned to errors to obtain the overall *error* and *failure* rate of the system. Error rate in a system is defined as a known misclassification per total signals predicted, and that would be the sEMG labelled as *error* in $\hat{\mathbf{L}}_{TS}$. *Failure* rate is the wrongly labelled sEMG signals in $\hat{\mathbf{L}}_{TS}$ compared to $\mathbf{L}_{TS}$ minus the number of errors over the total of signals predicted. This process is illustrated in Figure 1 as the right grey rectangle.

4. Results

PLS stands out when it comes to model data that have high collinearity because it uses latent variables. To evaluate the collinearity of the data, principal component analysis (PCA) was carried out with $\mathbf{X}_{TR}$ and $\mathbf{Y}_{TR}$ of PLS_{OP}. The result of this analysis shows that 4860 dimensions of $\mathbf{X}_{TR}$ can be represented with 5 to 8 components, explaining 88.2% to 93.3% of the variance, respectively. Similar results were obtained when analyzing $\mathbf{Y}_{TR}$, where the number of principal components is 7, much smaller than the original dimension, 35, and explaining all the variance. In addition, in this case, 15.2% of the pairwise correlation coefficients between $\mathbf{Y}_{TR}$ columns are, in absolute value, greater than 0.5. These results, specific to our problem, indicates the need to use a regression on latent variables, such as PLS.

Optimization of the PLS model, Section 3.8, generates 280 ECOC matrices; therefore, 280 matrices $\mathbf{Y}_{TR}$ result in 280 PLS models. Each of these iterations may have a different c, codeword length, of the ECOC matrix and a different DMCEN index. The optimization looks to minimize the DMCEN index in the training subset, $\mathbf{X}_{TR}$, obtaining a PLS_{OP} optimum model with seven latent variables. DMCEN = 0.031, and codeword length $c = 35$. All codeword lengths and DMCEN values obtained are depicted in Figure 4. Codeword length varies from 30 to 39, greater than the expected 30 (Section 3.3), with a median value of 36. DMCEN varies from 0.031 to 0.217 with a median and mean value of 0.12 with a very symmetric distribution.

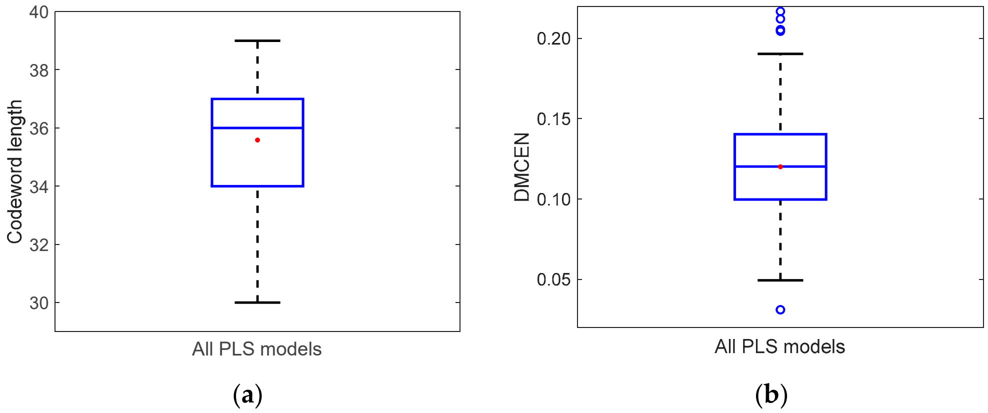


Figure 4. Some characteristics of the 280 repetitions made in the optimisation step. (**a**) Codeword length of ECOC matrices. (**b**) DMCEN values for PLS models built with these ECOC matrices.

Sensitivity and specificity in the training matrix $\mathbf{S}_{TR}$ are shown in Table 1. The meaning of this matrix was thoroughly discussed at the beginning of Section 2. There are two elements in Table 1 that differ from one (the best possible result); both cases will be commented. First, the element marked with (*), 0.86, means that one C_3 sample was wrongly assigned to the C_1 class, so the C_1 class model is not completely specific against C_3. Second, the element marked with (**), 0.71, means that two C_6 samples were assigned to C_5, so the C_5 class model is not completely specific against C_6. From this analysis, we can say that the C_1 and C_5 class models are unable to fully reject signals from C_3 and C_6, respectively.

Table 1. Sensitivity and specificity matrix $\mathbf{S}_{TR}$ for the test dataset $\mathbf{X}_{TR}$.

True Class	**Model of Class**							
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	1	1	1	1	1	1	1	1
C_2	1	1	1	1	1	1	1	1
C_3	0.86 *	1	1	1	1	1	1	1
C_4	1	1	1	1	1	1	1	1
C_5	1	1	1	1	1	1	1	1
C_6	1	1	1	1	0.71 **	1	1	1
C_7	1	1	1	1	1	1	1	1
C_8	1	1	1	1	1	1	1	1

(*) Specificity of C_1 class model against C_3. (**) Specificity of C_5 class model against C_6.

Applying the PLS_{OP} model to the test matrix $\mathbf{X}_{TS}$, the sensitivity and specificity matrix $\mathbf{S}_{TS}$ is obtained, Table 2, and DMCEN associated with $\mathbf{X}_{TS}$ is 0.209. Again, we are going to comment on the three elements of $\mathbf{S}_{TS}$ that differ from the one in Table 2. First, the element marked with (*), 0.67, means that one out of the three signals from the C_3 class was wrongly rejected from the C_3 class model. Therefore, the class C_3 model sensitivity lowered to 0.67. The C_3 class model also wrongly accepted two signals belonging to C_8, thus reducing its specificity against C_8 to 0.33, marked as (**). Lastly, the element marked as (***), 0.33, means that the C_5 class model failed to reject two signals from the C_6 class, so its specificity against C_6 reduced to 0.33. From these results, we can say the C_3 class model is having problems to model the C_3 signals like its own and is failing to completely reject all signals from C_8. The C_5 class model is presenting the same flaw as in the training set wrongly assigning C_6 signals to itself.

Table 2. Sensitivity and specificity matrix $\mathbf{S}_{TS}$ for the test dataset $\mathbf{X}_{TS}$.

True Class	**Model of Class**							
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	1	1	1	1	1	1	1	1
C_2	1	1	1	1	1	1	1	1
C_3	1	1	0.67 *	1	1	1	1	1
C_4	1	1	1	1	1	1	1	1
C_5	1	1	1	1	1	1	1	1
C_6	1	1	1	1	0.33 ***	1	1	1
C_7	1	1	1	1	1	1	1	1
C_8	1	1	0.33 **	1	1	1	1	1

(*) Sensitivity of C_3 class model. (**) Specificity of C_3 class model against C_8. (***) Specificity of C_5 class model against C_6.

To understand the benefits of modelling sEMG signals as an alternative to usual classifiers, the assignation matrix, Table 3, of the test dataset $\mathbf{X}_{TS}$ is shown. This matrix was obtained by applying the PLS_{OP} model and decoding it. Each row of the matrix represents one sEMG signal from $\mathbf{X}_{TS}$ and is composed of a vector of zeros and ones indicating the class assignment made by the model. Each signal sEMG belonging to test dataset ($\mathbf{X}_{TS}$**)** is a row of Table 3, and each 1 means that the class model, the column, accepted the signal.

Table 3. Assignation matrix for the sEMG classes of the test dataset $\mathbf{X}_{TS}$. A 1 means assigned class and 0 not assigned.

EMG Signal	True Class	Class Assigned by the PLS_{OP} Model							
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
1	C_1	1	0	0	0	0	0	0	0
2	C_1	1	0	0	0	0	0	0	0
3	C_1	1	0	0	0	0	0	0	0
4	C_2	0	1	0	0	0	0	0	0
5	C_2	0	1	0	0	0	0	0	0
6	C_2	0	1	0	0	0	0	0	0
7	C_3	0	0	0	0	0	0	0	0
8	C_3	0	0	1	0	0	0	0	0
9	C_3	0	0	1	0	0	0	0	0
10	C_4	0	0	0	1	0	0	0	0
11	C_4	0	0	0	1	0	0	0	0
12	C_4	0	0	0	1	0	0	0	0
13	C_5	0	0	0	0	1	0	0	0
14	C_5	0	0	0	0	1	0	0	0
15	C_5	0	0	0	0	1	0	0	0
16	C_6	0	0	0	0	0	1	0	0
17	C_6	0	0	0	0	1	1	0	0
18	C_6	0	0	0	0	1	1	0	0
19	C_7	0	0	0	0	0	0	1	0
20	C_7	0	0	0	0	0	0	1	0
21	C_7	0	0	0	0	0	0	1	0
22	C_8	0	0	0	0	0	0	0	1
23	C_8	0	0	1	0	0	0	0	1
24	C_8	0	0	1	0	0	0	0	1

In this analysis, an *error* is defined as a signal that belongs to a number of gestures different from one, a *failure* is defined as a signal that is assigned solely to the wrong class, and a success is defined as a signal that is correctly assigned to its class.

For example, Row 7 of Table 3 shows an sEMG signal from the C_3 gesture class that was not assigned to any class, resulting in an *error* and reducing the sensitivity of the C_3 class model, as indicated by the (*) label in Table 2

Rows 17 and 18 show two signals from the C_6 class that have been assigned to two classes, the correct one (C_6) and an incorrect one (C_5). This results in two *errors*, but the C_6 class model sensitivity is not affected. However, incorrect assignment of the signals to C_5 reduces the specificity of the C_5 class model against the C_6 class, as indicated by the (***) label in Table 2.

Similarly, rows 23 and 24 show two signals that are correctly assigned to the C_8 class but are also incorrectly assigned to the C_3 class, which reduces the specificity of the C_3 class model against the C_8 class and adding two *errors* to the overall results.

To sum up, the assignation matrix, Table 3, shows that there were 5 *errors* where a signal was assigned to a number of gestures different than one, 19 successes where a signal was correctly assigned to its class, and 0 *failures*. It also highlights the origin of the changes in specificity and sensitivity in the **S** matrix, Table 2.

5. Discussion

Differences between a classifier and a class-modelling classifier such as PLS-ECOC were stated in the Introduction and State-of-the-Art Approaches Sections. To summarize, the main difference is that a class-modelling classifier can assign one gesture to none or more than one class. This feature combined with the PLS-box was used in this work to implement a fault-tolerant classifier that can detect *errors*, increasing the robustness of the whole system. This feature could not be achieved with previous classifiers used in sEMG GR applications.

To fully evaluate the extra information provided by a class-modelling classifier, the DMCEN [46] metric was used as described in Section 3.6. The usual metrics (accuracy, precision, or F1 score) will fall short because they assume that each object can belong to only one class. However, for the sake of comparison with other classifiers in the sEMG GR field, it seems reasonable to express the results also in the usual metric format: accuracy, precision, and F1 score. To this purpose, the *errors* (gestures assigned to none or two or more classes) were assigned to their most probable class. Note that with this transformation, the class-modelling algorithm loses its fault-tolerant capability.

Considered PLS-ECOC as a classification method, the confusion matrix in training $\mathbf{N}_{TR}$ = (n_{ij}), is diagonal, with n_{ii} = 7 and n_{ij} = 0 if i $\neq$ j. Therefore, accuracy, precision, and F1 are all equal to 100%.

In prediction, with the test set, the confusion matrix is shown in Table 4. The accuracy, precision, and F1 values are 87.5, 91.87, and 86.34%, respectively. They have been calculated in accordance with ref. [63] to maintain consistency with the definitions of sensitivity and specificity given in the introduction of Section 3.

Table 4. Confusion matrix $\mathbf{N}_{TS}$ for the test dataset $\mathbf{X}_{TS}$.

True Class	**Model of Class**							
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	3	0	0	0	0	0	0	0
C_2	0	3	0	0	0	0	0	0
C_3	0	0	2	0	0	0	0	1
C_4	0	0	0	3	0	0	0	0
C_5	0	0	0	0	3	0	0	0
C_6	0	0	0	0	2	1	0	0
C_7	0	0	0	0	0	0	3	0
C_8	0	0	0	0	0	0	0	3

However, in the field of sEMGs, there is no agreement on the definition of accuracy, and this ambiguity makes it difficult to compare results. Section 3.5 of ref. [33] is devoted to this question. The authors find that one work of the 56 studied defines "the recognition accuracy" (a gesture is considered a true positive, i.e., the gesture is recognized correctly when the model determines what gesture was performed and when this gesture was performed by a person). Eleven other papers use the concept of "classification accuracy" because they only took into consideration what gesture was performed by a person as a true positive, and the remaining papers do not show what they consider a true positive. This ambiguity is transferred to the calculation of accuracy. In our case, both definitions coincide with each other and with our calculation because the rest position (no gesture, in other papers) has been considered as a class in the model.

Other definitions of accuracy used in the analysis of sEMG signals are the ratio of correctly predicted gestures (true positives plus true negatives) to the total gestures [67,68], obtaining 98.44% with our data. In [33], accuracy is also defined as the arithmetic mean of sensitivity and specificity, which, in our case, would be 87.5%.

In any case, the accuracy of PLS-ECOC transformed into a classifier is in the range of values found in the literature described in Section 2. Specifically, using the means and standard deviations obtained from the accuracies reported in references [14,33–35] and assuming that the data come from a normal distribution, the tolerance limits state that we can be 95.0% confident that 99.0% of the distribution lies above 73.62, 52.76, 63.40, and 71.92%, respectively. The value obtained, 87.5%, with PLS-ECOC transformed into a classifier is inside all four tolerance intervals.

Another important aspect of the proposed class-model is that it does not require human intervention in adjusting it. Most approaches in the sEMG GR field require the selection of features to make the classifier work or to adjust the parameters of the classifier. By contrast, the proposed method is totally automated, simultaneously adjusting feature extraction, dimensionality reduction in the *nvl* latent variables, and classification performance using the DMCEN index. It only needs a labelled dataset to work with. This approach should provide better flexibility, reducing design time. Another advantage from the designer's point of view is the information that the PLS-ECOC provides; the sensitivity and specificity matrix **S** can give important insights on which gestures are being confused, providing a useful debug tool to fix or improve the overall system design. For example, it can highlight which electrodes should be placed differently or that some gestures cannot be used as commands for the system.

The last part of this discussion will focus on the applicability of the presented method for a wearable, real-time application. Three metrics are evaluated: the size of the model, latency, and scalability. The size of the PLS_{OP} model is extremely important for wearable devices, as it directly impacts performance and limits usability. Any lineal model has a parameter size of $nvc \times nc \times c$ where nvc is the number of voltage values recorded, nc is the number of channels, and c is the codeword length. The computation required is also proportional to the size of the model because obtaining the classification value will require one multiplication and one addition per parameter. This size is comparable with the SVM algorithm. Translated into memory parameters, it will occupy 664 kB, a size that could not be fitted into the ram memory of a low-power MCU. However, the current implementation uses a 3 s gesture as an input. So, the size will fold by 30 in a real case scenario where windowing of the order of the hundredth of milliseconds is used. Latency, another key parameter, is composed of the time to acquire one gesture to be evaluated (3 s) and the time to process it. Being a linear model, the processing time in an MCU will be low because it is only composed of a multiplication and an addition per parameter. The last parameter is scalability. PLS-ECOC codification makes the size of the model increase logarithmically ($c \cong 10 \log_2 K$) compared to SVM usual encoding, one vs. one, which increases quadratically ($c = K(K-1)/2$), being, in both cases, K, the number of classes, providing a good base for more complex HMI systems that will require a large number of gestures to be functional. All these characteristics should be tested by implementing PLS-ECOC in a real embedded device and evaluating its performance.

Another hint that there is room for improvement in the reduction of the codeword, c, length is the fact that the observed correlation in Figure 4 is 0.2 between DMCEN and c. This result implies that a larger codeword (and model) does not translate into better overall classification characteristics of the model.

6. Conclusions

The proposed PLS-ECOC model for gesture recognition using sEMG signals can provide a *failure*-free result in the prediction phase. While keeping comparable results in accuracy, precision, and F1 of 87.5, 91.87 y 86.34%, respectively, to the current state-of-the-art classifiers. This highlights the effectiveness of the proposed method as a fault-tolerant classification system. To the best knowledge of the authors, this type of fault-tolerant design has not been studied before, and it opens up new possibilities for developing classification applications in critical systems where safety is a key requirement, such as controlling a robotic arm.

The proposed method also offers great flexibility for designers because it does not require any exploratory analysis to obtain features, which can be time-consuming and will be specific to each dataset. The PLS-ECOC method simultaneously adjusts feature extraction, dimensionality reduction in the *nvl* latent variables, and classification performance using the DMCEN index. Dimensionality reduction performed simultaneously with the adjustment of the model provides a great base for future developments. It is more robust and adapts better to differences between subjects, electrode placement, or acquisition systems. However, this paper was only evaluated on one subject, and further studies with more subjects, electrode positions, and acquisition systems should be carried out. The sensitivity and specificity matrix **S** provides a powerful tool for system designers to better understand where gestures are confused and to make decisions to improve the design.

DMCEN is as simple as the accuracy metric used in discriminant classifiers. However, it extends its validity to class modelling classifiers, proving its utility for optimizing the PLS model.

The PLS-ECOC lineal model provides good characteristics for further development because of its computational simplicity. However, in its current state, it is too big to fit in a low-power MCU that could be used in a wearable design. To address this issue, a couple of future work lines are proposed: reduction of the codeword length by reducing the ECOC output matrix and windowing the sEMG signals, which would be a required step to implement the method in a real-time application.

Another improvement that should be taken into account is to use DMCEN optimization for the other two criteria (minimizing errors and minimizing the length of the codeword) either through a desirability function that introduces "a priori" weight to the criteria [69] or by applying a multicriteria optimization and using the Pareto front of the optimal solutions to obtain the optimal solution "a posteriori" [70]. As the relationship between the three criteria and the experimental data is not described by any deterministic function, a heuristic method, such as a genetic algorithm [71], could be used in the optimization process.

Author Contributions: Conceptualization, P.S., A.A. and L.A.S.; methodology, P.S. and L.A.S.; software, P.S., M.d.l.C.O. and L.A.S.; validation, P.S., L.A.S. and M.d.l.C.O.; formal analysis, P.S. and L.A.S.; investigation, P.S.; data curation, P.S., M.d.l.C.O. and L.A.S.; writing—review and editing, P.S., A.A., M.d.l.C.O. and L.A.S.; funding acquisition, P.S., M.d.l.C.O. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Consejería JCyL number BU052P20 and cofinanced with European funds. This research was partially funded by the Spanish Ministry of Science and Innovation and the European Union under Grant PID2020-116011RB-C22 MyGait and Comunidad de Madrid under grant P022/BMD-7236 MINA-CM. Projects).

Data Availability Statement: Sarabia Ortiz, Pablo (2023), "sEMG 4 channel 8 gestures dataset", Mendeley Data, v1 http://dx.doi.org/10.17632/2f982h6vg3.1 (accessed on 4 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hakonen, M.; Piitulainen, H.; Visala, A. Current state of digital signal processing in myoelectric interfaces and related applications. *Biomed. Signal Process. Control* **2015**, *18*, 334–359. [CrossRef]
2. Dunai, L.; Novak, M.; García Espert, C. Human Hand Anatomy-Based Prosthetic Hand. *Sensors* **2021**, *21*, 137. [CrossRef]
3. Khushaba, R.N.; Kodagoda, S.; Takruri, M.; Dissanayake, G. Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals. *Expert Syst. Appl.* **2012**, *39*, 10731–10738. [CrossRef]
4. Krebs, H.; Palazzol, J.; Hogan, N. Rehabilitation robotics: Performance-based progressive robot-assisted therapy. *Auton. Robot.* **2003**, *15*, 7–20. [CrossRef]
5. Bouteraa, Y.; Ben Abdallah, I.; Ibrahim, A.; Ahanger, T.A. Development of an IoT-Based Solution Incorporating Biofeedback and Fuzzy Logic Control for Elbow Rehabilitation. *Appl. Sci.* **2020**, *10*, 7793. [CrossRef]
6. Santos Silva, C.; Oliveira Santos, M.; Gromicho, M.; Pinto, S.; Swash, M.; de Carvalho, M. Electromyographic findings in primary lateral sclerosis during disease progression. *Clin. Neurophysiol.* **2021**, *132*, 2996–3001. [CrossRef]
7. Roldan-Vasco, S.; Orozco-Duque, A.; Orozco-Arroyave, J.R. Swallowing disorders analysis using surface EMG biomarkers and classification models. *Digit. Signal Process.* **2023**, *133*, 103815. [CrossRef]
8. Catelli, D.S.; Kuriki, H.U.; Polito, L.F.; Azevedo, F.M.; Negrão Filho, R.F.; Alves, N. Patellofemoral pain syndrome: Electromyography in a frequency domain analysis. *J. Phys. Conf. Ser.* **2011**, *313*, 012004. [CrossRef]

9. Subasi, A. Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. *Comput. Biol. Med.* **2013**, *43*, 576–586. [CrossRef]
10. Salinas, S.A.; Elgalhud, M.A.T.A.; Tambakis, L.; Salunke, S.V.; Patel, K.; Ghenniwa, H.; Ouda, A.; McIsaac, K.; Grolinger, K.; Trejos, A.L. Comparison of Machine Learning Techniques for Activities of Daily Living Classification with Electromyographic Data. In Proceedings of the 2022 International Conference on Rehabilitation Robotics (ICORR), Rotterdam, The Netherlands, 25–29 July 2022. [CrossRef]
11. Dillen, A.; Steckelmacher, D.; Langlois, K.; De Beir, A.; Marusic, U.; Vanderborght, B.; Nowé, A.; Meeusen Romain, F.; Ghaffari, O.; Romain, K.D.P. Deep learning for biosignal control: Insights from basic to real-time methods with recommendations. *J. Neural Eng.* **2022**, *19*, 011003. [CrossRef]
12. Zheng, M.; Crouch, M.S.; Eggleston, M.S. Surface Electromyography as a Natural Human–Machine Interface: A Review. *IEEE Sens. J.* **2022**, *22*, 9198–9214. [CrossRef]
13. Xiong, D.; Zhang, D.; Zhao, X.; Zhao, Y. Deep Learning for EMG-based Human-Machine Interaction: A Review, IEEE/CAA. *J. Autom. Sin.* **2021**, *8*, 512–533. [CrossRef]
14. Quitadamo, L.R.; Cavrini, F.; Sbernini, L.; Riillo, F.; Bianchi, L.; Seri, S.; Saggio, G. Support vector machines to detect physiological patterns for EEG and EMG-based human–computer interaction: A review. *J. Neural Eng.* **2017**, *14*, 011001. [CrossRef]
15. Diab, M.S.; Rodriguez-Villegas, E. Embedded Machine Learning Using Microcontrollers in Wearable and Ambulatory Systems for Health and Care Applications: A Review. *IEEE Access* **2022**, *10*, 98450–98474. [CrossRef]
16. Lu, Y.; Le, V.L.; Kim, T.T.H. A 184-μW Error-Tolerant Real-Time Hand Gesture Recognition System With Hybrid Tiny Classifiers Utilizing Edge CNN. *IEEE J. Solid-State Circuits* **2023**, *58*, 530–542. [CrossRef]
17. Jiang, Y.; Song, L.; Zhang, J.; Song, Y.; Yan, M. Multi-Category Gesture Recognition Modeling Based on sEMG and IMU Signals. *Sensors* **2022**, *22*, 5855. [CrossRef]
18. Morrison, D.F. *Multivariate Statistical Methods*, 3rd ed.; McGraw-Hill: New York, NY, USA, 1990.
19. Friedman, J.H. Regularized discriminant analysis. *J. Am. Stat. Assoc.* **1989**, *84*, 165–175. [CrossRef]
20. Barker, M.; Rayens, W. Partial Least Squares for discrimination. *J. Chemom.* **2003**, *17*, 166–173. [CrossRef]
21. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 1984.
22. Vapnik, V.N. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998.
23. Sun, M. A multi-class support vector machine: Theory and model. *Int. J. Inf. Technol. Decis. Mak.* **2013**, *12*, 1175–1199. [CrossRef]
24. Schölkopf, B.; Williamson, R.; Smola, A.; Shawe-Taylor, J.; Platt, J. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems 12, Proceedings of the 29 December 1999 Conference*; Solla, S., Leen, T., Müller, K.-R., Eds.; MIT Press: Cambridge, MA, USA, 2000; pp. 582–588.
25. McCulloch, W.; Pitts, W. A logical calculus of the ideas imminent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
26. Werbos, P. Beyond Regression. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 1974. Available online: https://www.researchgate.net/publication/35657389 (accessed on 4 March 2023).
27. Le Cun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
28. Wold, S. Pattern recognition by means of disjoint principal components models. *Pattern Recognit.* **1996**, *8*, 127–139. [CrossRef]
29. Derde, M.P.; Massart, D.L. UNEQ: A disjoint modelling technique for pattern recognition based on normal distribution. *Anal. Chim. Acta* **1986**, *184*, 33–51. [CrossRef]
30. Tax, D.M.; Duin, R.M. Support Vector Data Description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
31. Ortiz, M.C.; Sarabia, L.A.; García-Rey, R.; Luque de Castro, M.D. Sensitivity and specificity of PLS-class modelling for five sensory characteristics of dry-cured ham using visible and near infrared spectroscopy. *Anal. Chim. Acta* **2006**, *558*, 125–131. [CrossRef]
32. Valencia, O.; Ortiz, M.C.; Sánchez, M.S.; Sarabia, L.A. Simultaneous class-modelling in chemometrics: A generalization of Partial Least Squares class modelling for more than two classes by using error correcting output code matrices. *Chemom. Intell. Lab. Syst.* **2022**, *227*, 104614. [CrossRef]
33. Jaramillo-Yánez, A.; Benalcázar, M.E.; Mena-Maldonado, E. Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review. *Sensors* **2020**, *20*, 2467. [CrossRef]
34. Dhumal, S.; Sharma, P. EMG Pattern Recognition: A Systematic Review. In *Lecture Notes in Networks and Systems ISMS 2021*; Garg, L., Sisodia, D.S., Kesswani, N., Vella, J.G., Brigui, I., Xuereb, P., Misra, S., Singh, D., Eds.; Springer International Publishing: Cham, Switzerland, 2023; Volume 521, pp. 120–130. [CrossRef]
35. Nazmi, N.; Rahman, M.A.A.; Yamamoto, S.I.; Ahmad, S.A.; Zamzuri, H.; Mazlan, S.A. A Review of Classification Techniques of EMG Signals during Isotonic and Isometric Contractions. *Sensors* **2016**, *16*, 1304. [CrossRef]
36. Aviles, M.; Sánchez-Reyes, L.M.; Fuentes-Aguilar, R.Q.; Toledo-Pérez, D.C.; Rodríguez-Reséndiz, J. A Novel Methodology for Classifying EMG Movements Based on SVM and Genetic Algorithms. *Micromachines* **2022**, *13*, 2108. [CrossRef]
37. Sarabia Ortiz, P. Design and Evaluation of Electromyography Signal Processing Techniques Using Resource-Constrained Devices. Master's Thesis, E.T.S.I. Telecomunicación, UPM, Universidad Politécnica de Madrid, Madrid, Spain, 2020. Available online: https://oa.upm.es/72388/ (accessed on 4 March 2023).
38. Zhang, X.; Yao, L.; Wang, X.; Monaghan, J.; McAlpine, D.; Zhang, Y. A Survey on Deep Learning-based Non-Invasive Brain Signals: Recent Advances and New Frontiers. *J. Neural Eng.* **2021**, *18*, 1002. [CrossRef]

39. Cai, S.; Li, H.; Wu, Q.; Liu, J.; Zhang, Y. Motor Imagery Decoding in the Presence of Distraction Using Graph Sequence Neural Networks. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 1716–1726. [CrossRef] [PubMed]
40. Jia, H.; Su, Z.; Dua, F.; Zhan, Y.; Caiafa, C.F.; Solé-Casals, J. Improving Pre-movement Pattern Detection with Filter Bank Selection. *J. Neural Eng.* **2022**, *19*, 066012. [CrossRef] [PubMed]
41. Phinyomark, A.; Scheme, E. EMG Pattern Recognition in the Era of Big Data and Deep Learning. *Big Data Cogn. Comput.* **2018**, *2*, 21. [CrossRef]
42. Englehart, K.; Hudgins, B.; Parker, P.A.; Stevenson, M. Classification of the myoelectric signal using time-frequency based representations. *Med. Eng. Phys.* **1999**, *21*, 431–438. [CrossRef]
43. Arozi, M.; Caesarendra, W.; Ariyanto, M.; Munadi, M.; Setiawan, J.D.; Glowacz, A. Pattern Recognition of Single-Channel sEMG Signal Using PCA and ANN Method to Classify Nine Hand Movements. *Symmetry* **2020**, *12*, 541. [CrossRef]
44. Mendes Junior, J.J.A.; Freitasa, M.L.B.; Siqueira, H.V.; Lazzarettia, A.E.; Pichorima, S.F.; Stevan, S.L., Jr. Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach. *Biomed. Signal Process. Control.* **2020**, *59*, 101920. [CrossRef]
45. Chen, X.; Liu, A.; Wang, Z.J.; Peng, H. Corticomuscular Activity Modeling by Combining Partial Least Squares and Canonical Correlation Analysis. *J. Appl. Math.* **2013**, 401976. [CrossRef]
46. Valencia, O.; Ortiz, M.C.; Sánchez, M.S.; Sarabia, L.A. A modified entropy-based performance criterion for class-modelling with multiple classes. *Chemom. Intell. Lab. Syst.* **2021**, *217*, 104423. [CrossRef]
47. Baldomero-Naranjo, M.; Martínez-Merino, L.I.; Rodríguez-Chía, A.M. A robust SVM-based approach with feature selection and outliers detection for classification problems. *Expert Syst. Appl.* **2021**, *178*, 115017. [CrossRef]
48. Tannemaat, M.R.; Kefalas, M.; Geraedts, V.J.; Remijn-Nelissen, L.; Verschuuren, A.J.M.; Koch, M.; Kononova, A.V.; Wang, H.; Bäck, T.H.W. Distinguishing normal, neuropathic and myopathic EMG with an automated machine learning approach. *Clin. Neurophysiol.* **2023**, *146*, 49–54. [CrossRef]
49. Rodriguez-Zurrunero, R.; Tirado-Andrés, F.; Araujo, A. YetiOS: An Adaptive Operating System for Wireless Sensor Networks. In Proceedings of the 2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops), Chicago, IL, USA, 1–4 October 2018; pp. 16–22. [CrossRef]
50. *MATLAB*, version 9.9.0.2037887 (R2020b); The Mathworks, Inc.: Natick, MA, USA, 2022.
51. Palermo, F.; Cognolato, M.; Gijsberts, A.; Müller, H.; Caputo, B.; Atzori, M. Repeatability of grasp recognition for robotic hand prosthesis control based on sEMG data. In Proceedings of the 2017 International Conference on Rehabilitation Robotics (ICORR), London, UK, 17–20 July 2017; pp. 1154–1159. [CrossRef]
52. Sarabia Ortiz, P. *"sEMG 4 Channel 8 Gestures Dataset", Mendeley Data*, version 1. 2023. [CrossRef]
53. Esbensen, K.H.; Geladi, P. Principles of Proper Validation: Use and abuse of re-sampling for validation. *J. Chem.* **2010**, *24*, 168–187. [CrossRef]
54. Bates, S.; Hastie, T.; Tibshirani, R. Cross-validation: What does it estimate and how well does it do it? *arXiv* **2022**, arXiv:2104.00673v4. [CrossRef]
55. Kennard, R.W.; Stone, L.A. Computer aided design of experiments. *Technometrics* **1969**, *11*, 137–148. [CrossRef]
56. *PLSToolbox*, Version 8.9.2 R23162; Eigenvector Research, Inc.: Manson, WA, USA, 2021.
57. Wold, S. *Chemometrics and Bruce: Some Fond Memories*; Lavine, B.K., Brown, S.D.D., Booksh, K.S., Eds.; 40 Years of Chemometrics—From Bruce Kowalski to the Future; ACS Symposium Series; American Chemical Society: Washington, DC, USA, 2015. [CrossRef]
58. de Jong, S. SIMPLS: An alternative approach to partial least squares regression. *Chemom. Intell. Lab. Syst.* **1993**, *18*, 251–263. [CrossRef]
59. Rosipal, R.; Krämer, N. *Overview and Recent Advances in Partial Least Squares in Subspace, Latent Structure and Feature Selection*; Saunders, C., Grobelnik, M., Gunn, S., Shawe-Taylor, J., Eds.; Statistical and Optimization Perspectives Workshop, SLSFS; Springer: Berlin/Heidelberg, Germany, 2005; pp. 34–51. Available online: https://link.springer.com/chapter/10.1007/11752790_2 (accessed on 4 March 2023).
60. Ruiz, S.; Ortiz, M.C.; Sarabia, L.A.; Sánchez, M.S. A computational approach to partial least squares model inversion in the framework of the process analytical technology and quality by design initiatives. *Chemom. Intell. Lab. Syst.* **2018**, *182*, 70–78. [CrossRef]
61. Wand, M.P.; Jones, M.C. *Kernel Smoothing, Monographs on Statistical an Applied Probability, 60*; Springer-Science-Business Media: New York, NY, USA, 1995.
62. Cuadros-Rodríguez, L.; Pérez-Castaño, E.; Ruiz-Samblás, C. Quality performance metrics in multivariate classification methods for qualitative analysis. *Trends Anal. Chem.* **2016**, *80*, 12–624. [CrossRef]
63. Ballabio, D.; Grisoni, F.; Todeschini, R. Multivariate comparison of classification performance measures. *Chemom. Intell. Lab. Syst.* **2018**, *174*, 33–44. [CrossRef]
64. Wei, M.; Yuan, X.Y.; Hu, Q.H.; Wang, S.Q. A novel measure for evaluating classifiers. *Expert Syst. Appl.* **2010**, *37*, 3799–3809. [CrossRef]
65. Delgado, R.; Núñez-González, J.D. Enhancing Confusion Entropy (CEN) for binary and multiclass classification. *PLoS ONE* **2019**, *14*, e0210264. [CrossRef]

66. Sánchez, M.S.; Valencia, O.; Ruiz, S.; Ortiz, M.C.; Sarabia, L.A. DMCEN a MATLAB Function to Evaluate the Entropy Improvement Provided by a Multivariate K-Class-Model. Available online: https://www.mathworks.com/matlabcentral/fileexchange/112175-dmcen?s_tid=srchtitle (accessed on 4 January 2023).
67. Tuncer, S.A.; Alkan, A. Classification of EMG signals taken from arm with hybrid CNN-SVM architecture. *Concurr. Computat. Pract. Exper.* **2022**, *34*, e6746. [CrossRef]
68. Gopal, P.; Gesta, A.; Mohebbi, A. A Systematic Study on Electromyography-Based Hand Gesture Recognition for Assistive Robots Using Deep Learning and Machine Learning Models. *Sensors* **2022**, *22*, 3650. [CrossRef] [PubMed]
69. Derringer, G.C.; Suich, R. Simultaneous optimization of several response variables. *J. Qual. Technol.* **1980**, *12*, 214–219. [CrossRef]
70. Ortiz, M.C.; Sarabia, L.A.; Herrero, A.; Sánchez, M.S. Vectorial optimization as a methodological alternative to desirability function. *Chemometr. Intell. Lab. Syst.* **2006**, *83*, 157–168. [CrossRef]
71. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; Wiley: Hoboken, NJ, USA, 2001.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.