

Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Elettronica, Telecomunicazioni e Tecnologie dell'informazione

Ciclo **XXVIII**

Settore Concorsuale di afferenza: **09/E3**

Settore Scientifico disciplinare: **ING-INF-01**

TITOLO TESI

**Advanced Interfaces for HMI in Hand Gesture
Recognition**

Presentata da: **Simone Benatti**

Coordinatore Dottorato

Prof. **Alessandro Vanelli Coralli**

Relatore

Prof. **Luca Benini**

Esame finale anno 2016

Advanced Interfaces for HMI in Hand Gesture Recognition



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Simone Benatti

Dipartimento dell'Ingegneria Elettrica e dell'Informazione "Guglielmo
Marconi" -DEI
University of Bologna

This dissertation is submitted for the degree of
Doctor of Philosophy

Università' di Bologna

March 2016

I would like to dedicate this thesis to my loving parents.

Acknowledgements

Luca Benini and Elisabetta Farella, I thank you for leading me all the way through the PhD. You allowed me to work on what I'm interested in, you helped me with your experience and your advices, giving me every resource for my research and your support during these 3 years.

I would like to thank Jan Rabaey, who hosted me in Berkeley for my internship and gave me the opportunity to collaborate to groundbreaking projects and to work with inspiring colleagues. I thank also the guys of our team here in Bologna, especially Bojan Milosevic and Filippo Casamassima. They are great colleagues and friends, I am lucky to be part of a such exciting group.

A special thank to Emanuele Gruppioni, Laura Cavazza and Fred Burghard who inspired me with their knowledge of prosthetics, orthopedics and embedded design.

Finally, I owe a huge debt of gratitude to all my family and to Cinzia that have helped me in so many different ways.

Abstract

The present thesis investigates techniques and technologies for high quality Human Machine Interfaces (HMI) in biomedical applications. Starting from a literature review and considering market SoA in this field, the thesis explores advanced sensor interfaces, wearable computing and machine learning techniques for embedded resource-constrained systems. The research starts from the design and implementation of a real-time control system for a multifinger hand prosthesis based on pattern recognition algorithms. This system is capable to control an artificial hand using a natural gesture interface, considering the challenges related to the trade-off between responsiveness, accuracy and light computation. Furthermore, the thesis addresses the challenges related to the design of a scalable and versatile system for gesture recognition with the integration of a novel sensor interface for wearable medical and consumer application.

Table of contents

List of figures	xiii
List of tables	xvii
1 Introduction	1
1.1 Thesis outline	4
2 Embedded Setup for Hand Gestures Pattern Recognition	7
2.1 Overview	7
2.2 EMG Signal	9
2.2.1 Signal Description	9
2.2.2 Signal Acquisition	11
2.3 Sensor placement	12
2.4 Pattern Recognition Algorithms	14
2.4.1 Overview of the SVM	14
2.5 Acquisition Protocol description	16
2.6 Experimental Results	17
2.6.1 Single arm position classification	18
2.6.2 Different Arm Position Classification	19
2.6.3 Multisession Classification Performance	20
2.7 Discussion	22
2.8 Transient Elimination	25
2.8.1 HMM	27
2.8.2 Hybrid Classification	29
2.9 Real Time Requirements	30
2.10 Conclusions	32
3 Prosthetic Controller Design	35
3.1 Overview	35

3.2	System Architecture	38
3.2.1	Controller Board	39
3.2.2	Embedded SVM Implementation	49
3.2.3	PC Application	51
3.2.4	Control Strategy	52
3.3	Experimental Results	55
3.4	Conclusion	60
4	Versatile EMG based Pattern Recognition Platform	61
4.1	Overview	61
4.2	Active and Passive Electrodes	63
4.3	Architectural Overview	66
4.3.1	Chip level Design	66
4.3.2	System level design	67
4.3.3	Signal Pre-processing	70
4.4	Experimental Results	73
4.4.1	DWT classification	75
4.5	Complete System Validation	79
4.6	Experimental Results	80
4.7	Conclusion	86
5	Advanced Signal Conditioning	89
5.1	Overview	89
5.2	Biopotential Acquisition and Processing	92
5.3	DC Offset Compensation	92
5.3.1	System Description	95
5.3.2	Experimental Results	96
5.4	PLI Removal	99
5.4.1	ECG Signal	100
5.4.2	EEG Signal	103
5.5	Real Time PLI Removal	107
5.5.1	HW Techniques for PLI Removal	108
5.5.2	SW Techniques for PLI Removal	109
5.6	PLI Methods Overview	110
5.7	Experimental Results	113
5.7.1	Off-line analysis setup	113
5.7.2	Sensitivity to Power Line Amplitude	114

5.7.3	Sensitivity to Power Line Frequency Variability	114
5.7.4	Evaluation on the Cerebro platform	117
5.8	Conclusion	119
6	Conclusions	121
	References	123

List of figures

1.1	Research outline	3
2.1	EMG and muscular contraction	10
2.2	Ottobock active EMG sensor	11
2.3	Test to target muscle involved in hand movements	12
2.4	Activation Pattern	13
2.5	SVM algorithm diagram	16
2.6	Sensor positioning	17
2.7	Acquisition session	18
2.8	Accuracy for a single training session.	19
2.9	Difference of accuracy between classification in the position of the training and in other arm position.	20
2.10	single vs multiple position training	21
2.11	Single vs double training session.	22
2.12	Single vs double training session in different positions.	22
2.13	Support vectors number in different testing conditions.	23
2.14	Different Kernel Computational time vs number of SVs	24
2.15	Transient errors	25
2.16	Classification approach.	26
2.17	Error Statistics	29
2.18	Comparison of the average accuracies for each gesture of the two classifiers.	30
2.19	Effect of the initial transient duration boundary on the gestures classification accuracy.	31
3.1	Diagram of the system architecture	38
3.2	Block diagram of the interface between the MCU and the DC motor driver (top) and example motor current absorption curves (bottom).	40
3.10	SVM static allocation in the MCU's Flash memory	49

3.11	EMG data thresholding and segmentation	52
3.12	Diagram of the communication between the PC and the embedded board.	53
3.13	FSM for classification mode	54
3.14	Hand control strategy as executed during a gesture sequence.	55
3.15	Memory footprint of the SVM models (top) and computation requirements expressed in clock cycles (bottom) for the two analyzed setups.	58
3.16	Computation time at each input for the different setups and training datasets.	59
4.1	Block diagram of the commonly used active sensors setup (top row) and the proposed solution (bottom row).	66
4.2	Block diagram of the Cerebro ASIC [94].	68
4.3	Block diagram of the proposed device.	69
4.4	Photo of the assembled PCB (top view). The dimension of the circuit is 85×50 mm.	70
4.5	Raw data acquired from the Cerebro ADC (top) and the processed signals (bottom).	71
4.6	EMG signals (top) and their spectrum(bottom) for the tested system: (a) Cerebro, (b) Ottobock, (c) MYO. The signals represent gesture contractions and were acquired in sequence from the same user, placing the electrodes in the same position	73
4.7	Accuracy of the classifier signal of data acquired with dedicated AFE (a) and commercial active sensors (b)	74
4.8	Wavelet decomposition scheme	77
4.9	Prediction labels for DWT featured signal.	78
4.10	Computation time for the online classification with SVM models of varying number of support vectors, as measured on the Cortex M4 MCU.	80
4.11	Sensor placement and position occurrence in best accuracy classification	81
4.12	Gesture recognition accuracy (a) and number of SVs (b) for varying numbers of input channels and recognized gestures. The reported curves are the average of the 4 users	83
4.13	(a) Recognition accuracy and number of SVs for different SVM input sample rates. (b) Recognition accuracy of the SVM algorithm for different number of input channels, using EMG signals acquired with the MYO armband (dashed line) and our device (continuous line).	84
4.14	Breakdown of the power consumption of the microcontroller for three different applications: (a) 8 channels data stream at 1KHz, (b) 7 gestures recognition at 25Hz and (c) 7 gestures recognition at 1KHz.	85

5.1	Block diagram of the active sensors configuration.	95
5.2	Block diagram of the proposed solution.	95
5.3	EMG signal DC offset (a) and drift (b)	97
5.4	Two channels acquiring the same EMG signal, with only the initial compensation (gray) and with periodic compensation (blue).	98
5.5	Comparison of the FFT of the PLI filtering techniques (top to bottom): notch filter, time domain subtraction, regression subtraction and sinusoidal modeling. 101	
5.6	Raw ECG signals (top) and their FFTs (bottom) for the Cerebro platform (a) and for the AD7194 acquisition board (b).	102
5.7	Color-coded power spectra of consecutive 90-s epochs (4 s window) with Hanning window. Data was sampled with 250 Hz. Spectra are color coded on a logarithmic scale, 30 dB corresponds to an input signal with an amplitude of 100 μ V.	104
5.8	Illustration of the electrodes arrangement on the head of the patient with the denoted patient ground G. The pairs of electrodes are split in colors where the gray ones are referenced to the right ear (R2) and the the white ones to the left ear (R1).	105
5.9	First pair of EEG channels comparing the wearable EEG device with the state-of-the-art in USZ. Brain waves are obtained after proper filtering and decomposing of the EEG signals in delta, theta, alpha, and beta frequency bands.	106
5.10	Frequency response of the two electrode pairs. FFTs are very similar in amplitude except for the notch filter at 50 Hz of the USZ acquisition system. 107	
5.11	Signal acquisition block diagram.	110
5.12	Block diagram of the adaptive PLI filter.	113
5.13	Sensitivity to PLI amplitude.	115
5.14	Sensitivity to power line frequency variations when f_{PLI} is known.	115
5.15	Sensitivity to power line frequency variations when f_{PLI} is unknown.	116
5.16	Convergence behavior of the APF.	117
5.17	PLI removal for ECG signals. Top row: raw and filtered signals (units: seconds and mV), bottom row: frequency spectrum (units: Hz and dB); from left to right: notch, SM, RS, APF.	118
5.18	PLI removal for EMG signals. Top row: raw and filtered signals (units: seconds and mV), bottom row: frequency spectrum (units: Hz and dB); from left to right: notch, SM, RS, APF.	118

List of tables

2.1	Confusion matrix for single training session.	17
3.1	Healthy Subjects	57
3.2	INAIL Patients	57
3.3	NINAPRO Subjects	57
4.1	Comparison with state-of-the-art AFE ASICs.	64
4.2	Comparison of the Cerebro AFE with two commercial solutions	68
4.3	Signal preprocessing resource requirements	72
4.4	Confusion matrix for recognition with passive sensors.	75
4.5	Confusion matrix for recognition with active sensors	75
4.6	Comparison between DWT and Envelope Detection.	78
4.7	System Specs	88
5.1	Comparison with state-of-the-art AFE ASICs.	94
5.2	Average offset with different acquisition techniques.	99
5.3	Computational and memory requirements.	119

Chapter 1

Introduction

The acronym HMI (Human Machine Interaction) represents the connection layer between users and electronic devices, a multimodal approach including the hardware, the software and the methodologies that enable information exchange and control of machines. HMI has been defined by the ACM as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them, because human-computer interaction studies a human and a machine in communication. " (cit) The HMI can be investigated in many ways and generates strong interest in wearable systems.

In fact, the rapid growth of the miniaturized and efficient electronic systems enables the application of these paradigms to the development of unobtrusive systems to interact with electronic devices, opening interesting application scenarios. The smartphones, smartwatches or armbands that are invading the market, are clear examples of the revolution in the modality of interaction between humans and objects.

The recognition of gestures represents the interface used by these devices to interact with the users. The interaction and the recognition express at various levels but in all these kind of systems the acquisition, processing and optimization of the energy budget are the fundamental challenges. The sensors are the interface that bridge the gap between the physical world and the embedded system while the algorithms are the key to decode and regulate the interaction of the user with the system. All these elements must take in account the resource constraints that impact on the system design. Among the gestures used in HMI, the hand gestures are the preferred natural way of control and interaction of humans both for human-to-human and human-to machine, representing fast and effective physical interaction to receive information and send commands to object.

Since the sensing technologies and sensors are the first elements in the architecture of a HMI system, the use of sensors interfaced directly with the muscle represents the most

physiological approach considering that the muscular tissue is the final actuator in the control of hand movements. The most important signal, acquired by the muscular activity, is the Electromyogram (EMG), widely studied in clinical and diagnostic applications. The EMG can be acquired and analyzed with high end bench top medical devices but nowadays its acquisition and processing represents a challenge when done with resource constrained systems.

In fact, the small size needed for unobtrusive solutions limits the size and the number of the components that can be used and the dynamic use conditions of wearable devices need robustness features not needed in a bench top setup. Indeed, the interface between the muscles and the sensors is affected by many noise sources, like power line interferences, crosstalk, perspirations and skin contact impedance.

The muscular pattern recognition based on machine learning techniques is a widely studied field of research. The application of statistical learning algorithms, both supervised or unsupervised, reaches good performance in terms of accuracy of the recognition even though these algorithms are computationally demanding and most of them are not suitable on an embedded system. Most of the research focuses on the tuning or on the comparison of pattern recognition algorithms without considering the constraints of an embedded system in terms of memory and computation capabilities.

The upper limb prosthetics is a major research field for hand gesture recognition. The restitution of the functionality of the hand is a major goal in rehabilitation and healthcare applications. The study of the HMI in these application copes mainly with the interfacing and sensors issues and with the performance of the recognition, as the prosthetic systems must be as much as possible reliable and robust. In the commercial SoA systems used in prosthetics [10], where the requirements of reliability and robustness are strict, the muscular activity recognition is based on simple threshold detection. The gestures are encoded in predefined bursts of flexions and extensions of the residual wrist muscles. This method does not provide a natural interface while requiring high level of concentration during the use of the prosthesis and a long learning curve [50]. The recovery of the hand functionality with a natural gesture control is a major task in the research field of HMI for EMG gesture recognition. The outcomes of these research can be applied also out of the rehabilitation scenario, exploiting the EMG based HMI for a generic user interface. For instance, the Thalmic Labs released the Myo on the market, an armband for EMG gesture recognition, interfaced with a PC via BlueTooth low energy. The sensor armband sends the data to a PC where they pattern recognition algorithm runs. This wearable device is ergonomic and well designed but the gesture recognition needs a PC platform and the wearable part of the device is, *de facto*, just an acquisition system that streams EMG data to a high end platform where a

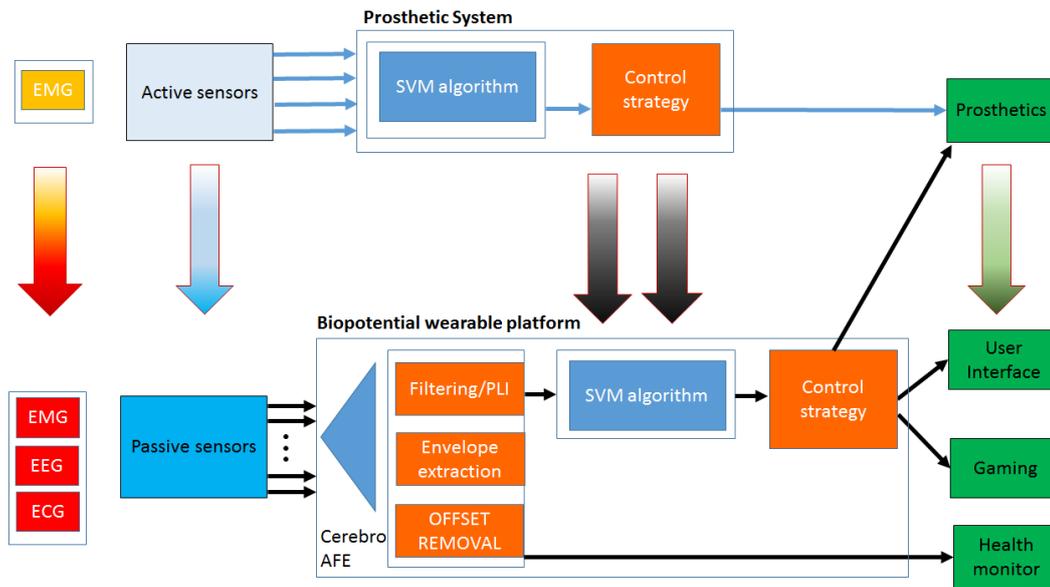


Fig. 1.1 Research outline

semi-supervised pattern recognition algorithm runs.

The study of a complete and efficient system for EMG hand gesture recognition requires a multilevel approach ranging from the chip level to the algorithm optimization and to the system level design. The work presented in this dissertation focuses on the analysis and on the multilevel design of a complete HMI for hand gesture recognition, proposing an efficient approach to overcome the main limits of the State of Art (SoA) in this field.

The first part of the thesis describes a 2 year long project in collaboration with INAIL-Centro protesi, one of the most important center in Europe for prosthetics and rehabilitation. The aim of the project was the development of a myoelectric prosthetic controller based on machine learning algorithms. Initially we collected a rich EMG dataset of EMG from the hand gestures of 10 healthy subjects. We acquired data on 10 days sessions to understand how the variability of the signal affects the gesture recognition and to test the performance of the classification algorithm. Then we designed the complete prosthetic system starting with a board design based on COTS (components off the shelf) and developing the software and the firmware to control the prosthesis. We based the design of the system on the I-LIMB artificial hand, one of the most advanced commercial poliarticulated hand. The Support Vector Machine (SVM) algorithm, considered in literature the most robust and efficient classifier for these kind of applications, was embedded on the microcontroller platform and used to design ah high level control strategy for the prosthesis. The complete system was tested on 5 healthy subjects and on 4 INAIL patients. Furthermore the performance of the system was tested on the NINAPRO database that represents the benchmark dataset in the

analysis of movement with EMG signal. The final outcome of the project was the first prototype of a hand prosthesis controlled by machine learning algorithms embedded on a low performance platform.

The work continued, in collaboration with ETHZ, to extend the application scenarios and to optimize our EMG HMI. We developed a multisensor wearable platform based on Cerebro, an Analog Front End (AFE) designed for biopotentials acquisition. This platform performs the pattern recognition using low cost passive electrodes instead of the active high cost sensors used in prosthetics.. Reaching the same performance of the high end system we designed a scalable and flexible platform. The system was compared with the Myo Armband and reaches the same performance but without the use of a PC platform, since the computing is done totally onboard.

Finally, starting from our cerebro platform we analyzed some advanced signal processing and optimization techniques for the acquisition of other biopotentials, like ElectroEncephaloGraphy (EEG) or ElectroCardioGraphy (ECG). We enter in details of the DC offset compensation and the real time Power Line Interference (PLI) removal. Fig. 1.1 shows the architecture evolution of our work, starting from the diagram of the prosthetic controller (FIg 1.1 High) and coming to the more general wearable biopotential platform (Fig 1.1 Low). The following paragraph presents the thesis outline with the description of the content for each chapter.

1.1 Thesis outline

In **Chapter 2** we start the analysis of the problem of EMG pattern recognition for hand gesture classification. We started by collecting a robust dataset of EMG signals targetting the variability of the acquisition among days. In fact, we collected data from 10 healthy subjects repeating the acquisitions over 10 days and with 4 different arm position. We investigated how the performance of the SVM classifier on a single day training drops among multiple session classification and varying the arm position. We compared the mean classification accuracy with one or two separated training sessions keeping into account the number of SV calculated by the model training. We profiled the implementation of the SVM on an ARM Cortex microcontroller to verify that our implementation is compliant with reat time requirements for a prosthetic system. Finally we tested a combined approach of SVM and HMM used to minimize the performance loss due to the misclassification in EMG signal rise and fall transient.

Chapter 3 describes the complete implementation of the prosthetic system. We described the board designed to control the prosthesis, we addressed the issues of the embedded im-
plementation of the SVM algorithm and we shown how the use of an high level FSM to

control the prosthesis can improve the performance of the simple classification algorithm. In fact, we considered that the final goal of the prosthesis is the correct execution of the end-to-end intended hand movement and not to simply reach an high classification accuracy. We tested the system on 5 healthy subject and on 4 INAIL patients. Furthermore we verified our system with an excerpts of the NINAPRO database that is the most complete dataset for EMG pattern recognition. We concluded that our system outperforms the Ninapro reaching its performance using a 4 sensors setup instead of the 10 sensors used by the Ninapro. This part concludes the INAIL project.

The **Chapter 4** describes the Cerebro EMG acquisition, comparing active and passive electrodes based pattern recognition. We developed a system for EMG acquisition using low cost passive electrodes and an AFE designed at ETHZ for biopotential acquisition. We described the board designed for the wearable devices, the AFE and we described the study and the implementation of the wearable application, profiling the optimizations in terms of number of sensors, positioning, sampling rates and the tradeoff between gestures, accuracy and computational charge. We also compared our system with the Myo armband that represents the SoA of EMG gesture recognition interfaces. We obtain the same performance of the Myo in terms of accuracy but we performed the computation onboard in real time, while the Myo armband requires a PC platform to run the pattern recognition algorithm.

Chapter 5 explores the use of the low cost cerebro platform considering the signal quality improvement. Firstly we described the jump compensation of the EMG signal that is performed with a closed loop interaction between the analog and digital part of the system. Finally we analyzed the PLI removal showing the limits of traditional regression-subtraction and sinusoidal moelling filters in real time online applications. Moreover we proposed the profiling and the implementation of an efficient real time adaptive filter for PLI removal on EMG and ECG signals.

In **Chapter 6** we draw the conclusions showing the direction of our ongoing research and the future work.

Chapter 2

Embedded Setup for Hand Gestures Pattern Recognition

2.1 Overview

Assistive technologies in the last years are boosting interesting research efforts to enhance quality of life. In fact, the availability of accurate sensing technologies at a relatively low price and the possibility to exploit the power of low-cost low-power and fast microcontrollers enable a whole new branch of applications, where wearable smart sensors and embedded systems can be used not only for monitoring but also for the implementation of sensor fusion techniques or complex machine learning algorithms and for provision of stimuli through actuation. An application field such as the control of prostheses can greatly benefit from this rapid technology evolution.

Recently, multi-finger active prostheses of the upper limb have appeared at commercial level (e.g. Touch Bionics i-Limb, RSL Steeper BeBionics 3, Otto Bock's SensorHand) enabling a larger set of gestures w.r.t. previous prostheses, therefore asking for an adequate strategy for their control. State of the art technologies for the feed-forward control of active hand prostheses are controlled via surface EMG in a way that forces the user to learn to associate contraction of what remains of the muscle to unrelated postures of the prosthesis, e.g., sequences of wrist flexion and extension correspond to various gestures. While at one side these techniques grant a good reliability and short activation time (within 30ms for the detection of the activity and less than 300ms for the classification), on the other side the control strategy is non-natural, requiring focus and a non-trivial learning curve for the user. It would be desirable instead to command the prosthesis movement by activating the muscle as to move the phantom limb.

Luckily, scientific literature recently proved that machine learning applied to EMG signals could be beneficial in prosthetics to bring the control towards more intuitive and natural strategies. In fact, convincing results have been shown both on healthy subjects [48] and amputees [37]

Initially, research on EMG signal based HMI was focused on the comparison between the various machine learning techniques applied to the patter recognition [9]. In particular, approaches based on LDA [116], ANN [12] and SVM [82] classifiers were investigated as the most promising solutions. These studies explore classification of hand gestures on a number of subjects varying from 3 to 6, with a number of gestures ranging from 4 to 9. These approaches compare different classification algorithms, reaching high levels of accuracy (around and beyond 90%), and exploring the use of feature extraction techniques to improve the performance. The high accuracy obtained with all the selected classifiers suggests that the different algorithms are mostly equivalent for the recognition of hand gestures and it is not possible to find a classifier or a feature set that definitively outperforms the others.

The setup and the number of electrodes used for the EMG signal acquisition are further key points with high impact on the design of an EMG hand gesture recognition system. The work in [59] compares the accuracy of systems based on implanted intramuscular and surface EMG electrodes. Results confirm that there is no significant difference between the two approaches in the classification of hand gestures. Passive surface electrodes are compared with active EMG sensors in [53] and [42]. These work analyze some simple hand movements (wrist flexions and extensions and hand pronations and supinations), classifying the EMG signals from 2 couples of active and passive electrodes placed in the same way. In this experiment, the data is collected from 5 subjects and classified using the Auto-Regressive (AR) model. The results show no significant classification difference between the two classes of sensors (85% versus 87% of accuracy). The impact of the number of used electrodes was explored with different outcomes. In [59], results show that on 6 subjects there is no significant improvement with more than 4 sensors, while in [14] the best accuracy tested on 12 subjects is reached with a 7 electrodes configuration and this confirms that to obtain good performance a multichannel approach is desired.

All the described systems rely on high-end data acquisition platforms and perform off-line processing. They do not focus on optimization for resource constrained platforms such as wearables. It has been proven that pattern recognition algorithms can obtain high gesture classification accuracy, but the design of real time efficient systems is still a challenge.

This chapter describes the preliminary analysis done in an on-going work towards a real-time embedded implementation of an EMG based control of an active hand prosthesis by use of machine learning techniques and in particular by use of a Support Vector Machine.

Starting from the lesson learnt by literature, this chapter faces, as first step, the variability on classification results due to the changes in placements of the EMG sensors that occurs in real life. In fact, the wearer limb is subject day by day to physical differences due to swelling, fatigue, perspiration that can cause misplacements of the sensors w.r.t. the desired position. Furthermore the same gesture can be performed in various positions and orientation of the arm (along the body, lifted up, etc.). These changes affects the classification performance and must be addressed properly. Starting from a data collection repeated in different days on 10 healthy subjects, we investigated the reliability of the gesture recognition in an EMG signal controlled hand prosthesis. The analysis takes into account that our final target is an embedded implementation. The analysis is based on three main elements: (i) the selection of a correct signal acquisition chain, (ii) the analysis of the physiological best placement of the EMG sensors to grant a robust classification, and (iii) the analysis of the performance of the system through days, evaluating the difference of performance when the sensors are placed and removed in different sessions, as in the real use of the device for the classification of natural movements. The activation of the same gesture is considered in multiple combinations of the arm orientation and position to better study their influence on the classification performance and consequently improve the robustness of the classifier. Some works show experiments with a high number of subjects and features to test the accuracy of their classification algorithm. These results reach accuracy ratio near 100% but they are not applicable in real scenarios, because they do not take into account the variability of the signal caused by the placement of the EMG sensors. The misplacement of the EMG electrodes among different sessions and the positions of the arm during the use of the prosthesis affect the classification performance. This work tries to address the problem analysing the best placement of four EMG sensors to maximize the recognition performance. Furthermore, we analyse the variability of training data along different days and caused by different positions of arm and forearm while performing the same gesture and we validate the proposed setup for the embedded implementation that we are describing in the next chapter of the dissertation.

2.2 EMG Signal

2.2.1 Signal Description

The EMG signal measures the electrical activation of the muscular fibres. Muscle tissue conducts electrical potentials similarly to the nerves and these electrical potentials are named muscle action potentials (AP). APs are generated by the passage of Na⁺ ions and K⁺ ions

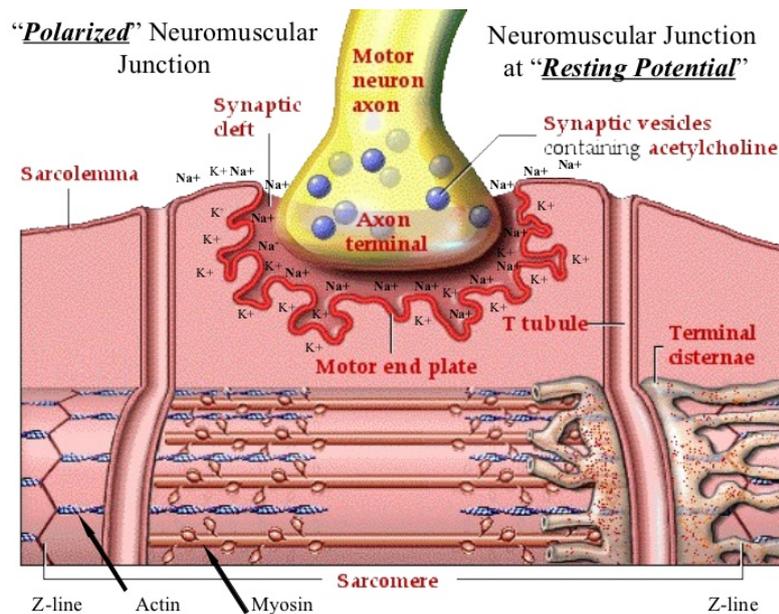


Fig. 2.1 EMG and muscular contraction

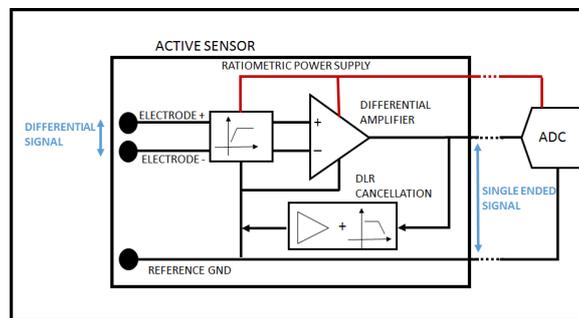
along nerve cell membranes. These reactions are the electrical messages sent out by the brain to muscles to start contractions. The Na^+ ions flow through the cellular membrane via the Na^+ channels. As a result of the influx of Na^+ ions, the cell membrane depolarizes and the nerve impulses propagate towards the target muscle cells (Fig. 2.1). The flow of the Na^+ ions into the nerve cell is an essential step in the conduction of action potential in nerve fibers and along axons and causes a release of Ca^{++} ions, causing cross-bridge binding and the muscle sarcomere contraction. [17]

When EMG electrodes are placed on the skin surface the signal is composed by all the AP of the cells underlying the electrodes that are aligned parallel to the muscle fibers. The surface EMG sensors are made by two conductive plates each one connected to the inputs of a differential amplifier that can sense the action potential of muscular cells.

The amplitude of this signal is 20mV (-10 to +10) depending on the diameter of the muscle fiber, the distance between the active muscle fiber and the detection site and the properties of the electrode. This kind of signals are also very noisy and difficult to manage even if the maximum bandwidth does not exceed 2kHz. The main causes are the noise due to motion artefacts, fiber crosstalk, electrical equipment and the floating ground of the human body, not referred to a solid ground potential.



(a) Ottobock sensor



(b) Internal architecture

Fig. 2.2 Ottobock active EMG sensor

2.2.2 Signal Acquisition

The typical EMG acquisition in prosthetics applications is based on active sensors, expansive analog sensors that provide a high-quality signal. Ottobock sensors [7] represent the commercial solution for EMG acquisition for high-end prosthetics, both in research and industrial applications. These sensors perform a full-analog signal conditioning based on a bandpass discrete filter, an instrumentation amplifier (IA) with a high gain stage and an offset cancellation feedback circuit that requires the use of a dedicated metal plate as the reference electrode for each sensor. Since we started from a SoA prosthetic setup, we decided to use the Ottobock 13E200 (Figure 2.2), a family of pre amplified sensors with single ended output [7]. In Ottobock sensor the EMG signal is amplified and integrated to reach an output span of 0 - 3.3V, ideal for the single ended stage of an embedded microcontroller ADC. The bandwidth of the Ottobock sensor is 90-450Hz with a further notch filter for the 50Hz. This is because the sensors for the classification of the gestures do not need extensive frequency information but a clear low noise signal. The analog signals are acquired from an embedded board based on ARM CORTEX M4 microcontroller. The internal 16-bit ADC



Fig. 2.3 Test to target muscle involved in hand movements

samples the data and an external Bluetooth interface sends it to a laptop. The embedded custom solution is preferable with respect to a data logger solution in this application because the final goal is to implement a complete embedded real time system. The data collected by the PC are managed in Matlab, for the signal processing and the pattern recognition.

2.3 Sensor placement

The position of sensors is a critical task in using a limb prosthesis. Muscles in healthy subjects are placed in known position and it is possible to have little differences in the amplitude of the activation signal or in the position of EMG sensor. Amputees presents specific characteristics in muscular structure and difference of amplitude of activation signal if compared with healthy subjects and, for this reason, it is important to evaluate subject by subject the possibility for an amputee to use an active prosthesis. The positioning strategy is tested on healthy subject in this chapter and the test on amputees is presented in the next chapter with the complete implementation.

Our strategy started from a previous work of [36], that placed 5 sensors on an elastic strip, with an equal distance each other's. The idea was to consider a sort of general pattern of the forearm muscle activation, without focusing attention on the anatomical structure of the arm. The risk of such approach is to lose the contribution of one or more sensors if misplaced. We propose an approach, based on a four sensors configuration, in two steps: a theoretical one that considers the muscles involved in hand movement and a practical one that verifies the good placement of the electrodes and the information given from all sensors.

The theoretical approach starts from the analysis of the muscular tissues of the limb. Signals have maximum quality where the muscle is wider, and the muscular fibres are spread. In the upper limb, the optimal zone is the proximal third. In the forearm, the muscles are divided in four groups, and the use of surface EMG sensors requires that muscle near the skin surface must be preferred for the classification. We can divide the forearm muscles in two groups: muscles in the internal part of the forearm (flexor radialis carpi, palmares longus, flexor carpi ulnaris, flexor superioris digitalis), involved in flexion movements, and

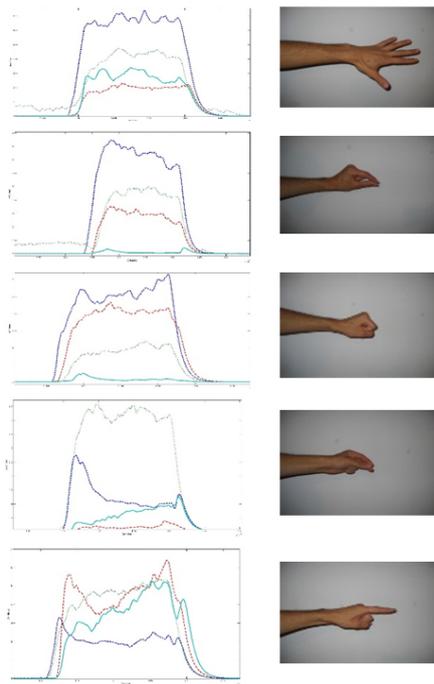


Fig. 2.4 Activation Pattern

muscles placed in the external part of the forearm (extensor communis digitorum, extensor digiti minimi, extensor carpi ulnaris), involved mainly in extension movements.

Nevertheless, the theoretical selection of muscles is not sufficient, because in the practical case the electrode catches a zone of many muscular fibres, introducing noise and requiring to verify the usability and the integrity of the target signal. Forearm was divided in different zones considering the dimension of the selected Ottobock sensors. The zones are numbered with progression, starting from the flexor carpi radialis, with counter clockwise sequence. Sensors are placed on the muscle primarily involved in movement selected. The muscles are found with tactile analysis and the corresponding number is used to place the sensor on the elastic strip used in this experiment (Figure 2.3).

By placing sensors on the flexor carpi radialis, flexor carpi ulnaris, extensor digitorum communis and extensor carpi ulnaris we obtained a good differentiation in classification pattern. The assumption for a pattern recognition control system is that the set of signals and features describing a given state of muscular activation are different from one state of activation to another. Figure 2.4 shows the gesture of the hand and the corresponding activation pattern acquired by the four sensors. The amplitude of each of the four sensors is a clear discriminant among patterns, even if the presented solution is not intended as the best placement, because the final target of this application is the use of the board on transradial amputees, and for these patients the placement strategy must be tuned subject by subject.

2.4 Pattern Recognition Algorithms

The recognition and decoding of the user's intended hand movement are major challenges in the development of a HMI system for prosthetics. The classification of hand gesture with pattern recognition methods is a well studied field of research and there are many contribution by signal processing and machine learning communities. Literature analysis seems to demonstrate that pattern recognition approach is a good way to restore a natural gesture control overcoming the limitations of the interfaces used in SoA prosthetics control. SVM [30] is the most accurate algorithm for pattern recognition with these kind of signals. In the following paragraph we will give an overview of the supervised learning approach and we will describe the intuitions and the key concepts of the classical logistic regression approach and the main differences with the SVM.

2.4.1 Overview of the SVM

The supervised learning approaches define a model from a dataset of given input data $(x^{(i)}, y^{(i)})$ where $x^{(i)}$ are the vectors of the input data and $y^{(i)}$ are the labels. The definition of the model is made by the tuning of a vector of parameters θ through the minimization of an opportune cost function (training). The model is represented by a boundary that separates the data in two or more classes (classification or prediction). In detail we can define the hypothesis function as

$$h_{\theta}(x) = g(\theta^T x)$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

This is named sigmoid function or logistic function and gives the estimated probability that the output $y = 1$ with a given new input x .

$$h_{\theta}(x) = P(y = 1|x; \theta)$$

Observing the sigmoid function we see that if $z = \theta^T x$ is greater than 0 the output of $g(z)$ is greater than 0.5 otherwise is

During the classification stage, if we have a input x for which $h_{\theta}(x) > 0.5$ (i.e. $\theta^T x > 0$) the corresponding y will be equal to 1, otherwise it will be equal to 0. The tuning of the

parameter θ is obtained solving an optimization problem by minimizing the following cost function with the input data of the training dataset :

$$Cost(h_{\theta}(x, y)) = \frac{1}{m} \sum_{i=1}^m \left[-y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right]$$

The SVM was developed in the 90s by the computer science community in the framework of the statistical learning theory. The basic idea of this supervised learning algorithm is to improve the performance of the classification obtained with logistic regression methods through the application of the *large margin* classification. [30]. The offline training phase of the algorithm uses labeled instances of data to calculate the optimal separation hyperplane (*maximum margin* hyperplane) between two classes of data through the solution of a convex optimization problem. Such separation plane is represented by a set of data vectors, named the Support Vectors (SVs), which belong to the borders of the two classes and they are used to classify new data instances. A diagram of SVM training and recognition is illustrated in

When the two classes are not linearly separable, the input data space can be mapped to a higher-dimensional space through a kernel function to allow an effective separation [97]. The Kernel used is based on a Radial Basis Function (RBF) and use a gaussian function expressed by

$$\exp\left(-\frac{\|x - l^{(n)}\|^2}{2\sigma^2}\right) \quad (2.1)$$

where x is the input vector, l^i is the i support vector and σ is the variance.

Having two possible classes, denoted as Cl_1 and Cl_2 , the formula of the decision function to classify a new input instance is:

$$f(\mathbf{x}) = \sum_{i=1}^{N_{SV}} y_i \alpha_i K(\mathbf{x}, \mathbf{s}_i) - \rho \quad f(\mathbf{x}) > 0, \mathbf{x} \in Cl_1 \quad f(\mathbf{x}) < 0, \mathbf{x} \in Cl_2 \quad (2.2)$$

where $\mathbf{x} \in \mathbb{R}^{N_F}$ is the input features vector, $\mathbf{s}_i \in \mathbb{R}^{N_F}$, $i = 1, \dots, N_{SV}$ are the support vectors, α_i are the support values, with y_i denoting the class they reference ($y_i = +1$ for Cl_1 , $y_i = -1$ for Cl_2) and $K(\cdot, \cdot)$ denotes the kernel function.

The SVM is a binary classification algorithm, but it can be extended for the application in a multiclass scenario by reducing the problem to multiple binary classification problems for each class. In our application, the input for the SVM classifier is the 4-dimensional vector of the EMG signals acquired by the electrodes ($N_F = 4$) and we used the Radial Basis Function (RBF) kernel. We classify up to 5 gestures using the *one-vs-one* approach, where each class is tested against every other in order to implement the multiclass classifier.

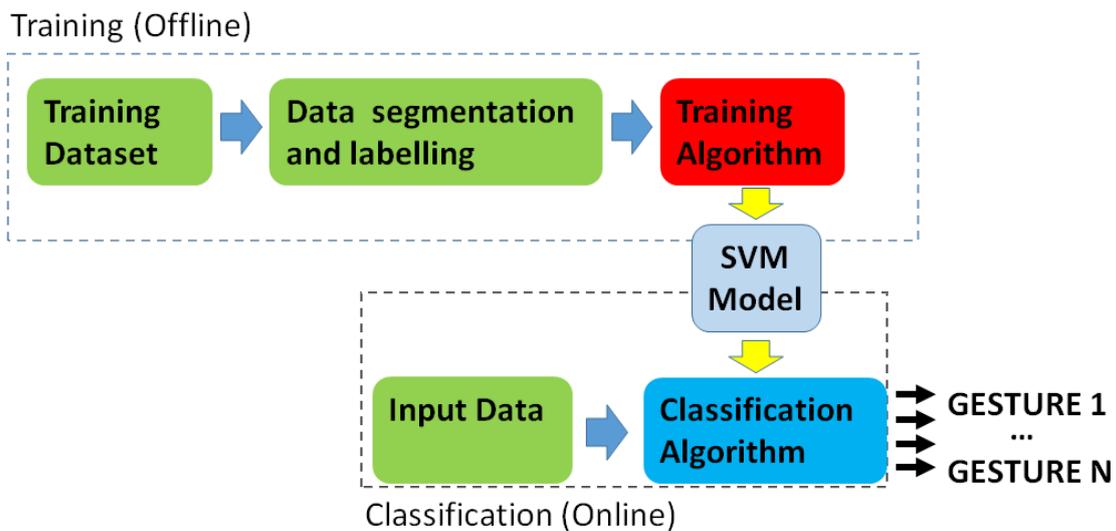


Fig. 2.5 SVM algorithm diagram

2.5 Acquisition Protocol description

The collection of the dataset follows an acquisition protocol described in this paragraph. Selected movements are the most common hand gesture used in daily life, to ensure a good quality of social reintegration. Gesture selected are shown in Figure 2.4 (closed hand, open hand 2-finger pinch, 3-finger pinch, point index). The classification includes also the rest position of the hand, recorded between two subsequent gestures. There are 9 subjects involved in the dataset acquisition. Gestures are acquired in 10 sessions collected 1 for day in 10 days not necessarily consecutive.

The sequence of gestures is repeated and scrambled in four arm positions (Figure 2.6a). Each acquisition session is divided in 4 steps, one for each position of the forearm. The collected sequence is composed by 10 repetitions of muscular contraction 3 second long. Between each contraction there are 3 seconds of rest. One file for each acquisition is captured. Subjects wear an elastic strip with the 4 EMG sensors. Sensors positions are tuned with a simple procedure described below. An arm as shown in Figure 2.6. The operator ideally traces a line on the axial direction of the forearm (Fig. 2.6b-left). Then the operator places Sensors 1 and 2 at 30mm respectively on the left and on the right side of the line at the proximal third of the forearm. The operation is repeated for sensors 3 and 4 with the arm flex in the position shown in 2.6b-right by considering the ideal line and placing sensor 3 and 4 at 30mm of distance symmetrically at the two sides of the line, at the proximal third of the forearm. Once the sensor are placed, a “good positioning” signals trace is acquired to avoid

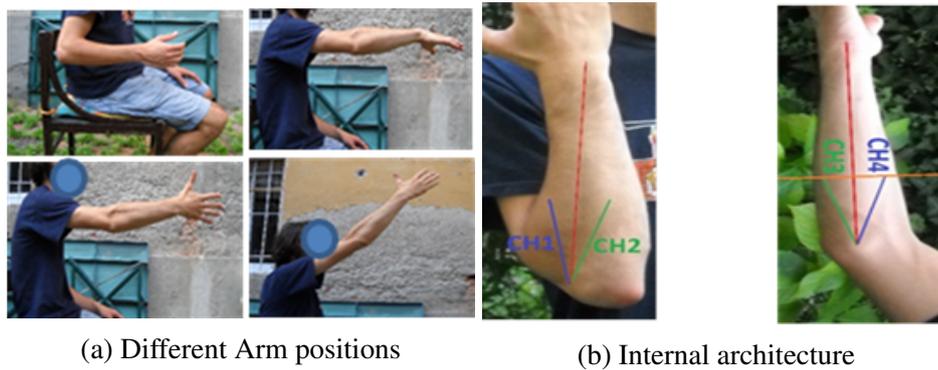


Fig. 2.6 Sensor positioning

Table 2.1 Confusion matrix for single training session.

ACCURACY %		REST	PREC	OPEN	POWER	POINT
99,4	REST	77185	118	0	339	0
91,6	PREC	149	24992	0	1688	445
75,8	OPEN	125	416	10088	2689	0
95,7	POWER	186	162	6	14745	302
82,3	POINT	82	756	0	1820	12371

misplacement of the strip. The test is made with the gestures of hand open and closed, which correspond to a non-zero signal for all the sensors.

To avoid that the subject learns the sequence of gestures and loses the naturalness in movements, the sequence are scrambled and each acquisition in a different arm position has a different pattern of gesture. For the same reason, the positions of the arm are not always in the same sequence: for example in session 1 sequences are captured in distal position and in proximal position, in session 2 these sequence is inverted, and scrambled with different pattern of gesture in other sessions. A typical acquisition trace is shown in Fig. 2.7. The mean number of samples for typical acquisition session is 140k sample

2.6 Experimental Results

The implementation of a pattern recognition system in a real application for the gesture classification has three main issues: the correct placement of the EMG sensors, the variability of the activation pattern due to the various positions of the arm and the variation of muscular contraction among following acquisition sessions. The importance of this aspect is acknowledged in literature, where other works address specifically the problem of placement and the difference between surface and intramuscular sensors [60] and propose their placement

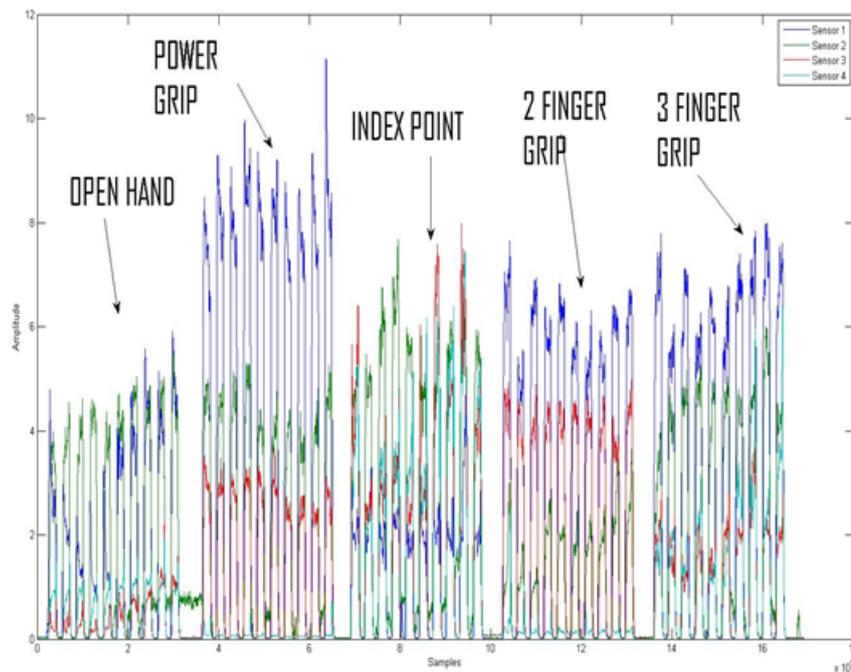


Fig. 2.7 Acquisition session

schemes. The training of the SVM is a critical phase since the quality of the training and the resulting matrix strictly affects the fidelity of the recognition. In turn, the segmentation phase performed on data collected is therefore crucial to augment the quality of the training. However data preparation, thresholding and segmentation are subjective operations based on the knowledge of the signal and on experience [76].

2.6.1 Single arm position classification

Our first purpose for the experiments is to verify the performance of the proposed placement scheme, to guarantee that a good classification is possible. For this application, we considered only the three fingers precision grip, because many commercial prosthetic systems, in the operating configuration, admit to choose between two or three finger precision grips. In the initial test, we considered only one session dataset (i.e. 10 gestures per position) per each patient. The training set is the 25% of the session dataset (i.e. 3 gestures per repetition) obtained with manual segmentation based on an empirical threshold. The training stage is performed offline on MATLAB libSVM. After the creation of the model, the classification gives the mean accuracy for each gesture. We selected the proximal position of the arm

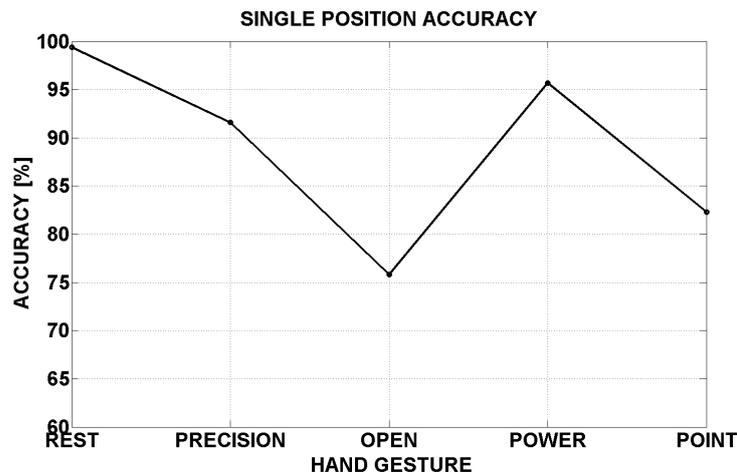


Fig. 2.8 Accuracy for a single training session.

because it is natural and typically used in the EMG test on pattern recognition accuracy. The data presented are the result of the mean values of accuracy for all subject involved in the experiment. Confusion matrix (Table 1) shows that the rest position, the power grip and the precision grip are the gestures recognized with higher accuracy.

The open hand and the index point gestures are recognized with lower accuracy because it is more difficult for the subject to repeat them exactly. Average classification accuracy is between 75% and 99.4%, as shown in Figure 2.8. This data validates the positioning strategy proposed and the setup used in our approach, because results meet the performance declared in literature on this topic.

2.6.2 Different Arm Position Classification

The variability in arm position is another important issue, when an EMG classification system is proposed for upper limb prostheses. Scheme et al. (2010) and Fougner et al. (2011) start to analyse this problem and propose a solution based on the placement of two inertial sensors used to detect the exact position of the limb. The position of sensors is on the arm and forearm and the combination of their information is used to detect the arm position. This approach is interesting, however it presents some practical problems if integrated in user daily routine. The solution presented is not integrated; the EMG board and inertial sensors are on separate boards and need separate power supply and a communication infrastructure. Furthermore, the system requires frequent calibrations due to the misplacement of inertial nodes on the body.

In this section, we evaluate the performance improvements coming by the use of an additional training session, which combines samples of the patterns from the different arm

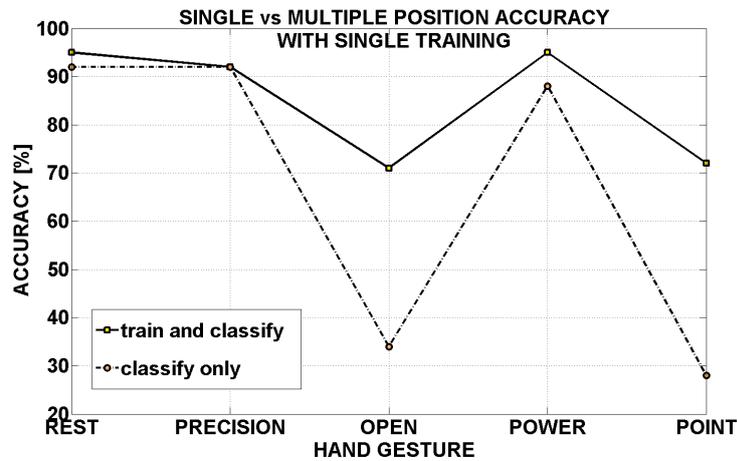


Fig. 2.9 Difference of accuracy between classification in the position of the training and in other arm position.

positions without modifying the test setup. The use of an excessively large training sets in traditional machine learning approach is not recommended, because usually the number of support vectors created by the algorithm is too high to guarantee a strong classification and a computational charge suited for an embedded implementation.

In the case study, we identified the most common positions of the arm in which a grip action of the hand is required. Positions are shown in Figure 2.6a. Initially a training session is performed for each patient only in the proximal position of the forearm and the classification performance is evaluated for the other limb position in the same acquisition session without further training.

Figure 2.9 plots the difference of accuracy in classification with arm positions not included in the training, showing the decreasing of performance compared with the classification in the position used for training. Even in this case the open hand and the point gestures suffer lack of performance. To verify the efficiency of the training made for the different positions of the arm, in Figure 2.10 the recognition accuracy obtained by the SVM classifier is shown when training on one position compared with the training in multiple positions. This strategy gives major benefits to the classification of the two gestures with major recognition errors because includes in the training set the gesture with more variability.

2.6.3 Multisession Classification Performance

The activation patterns of EMG sensors is strongly dependent from the position and the orientation of the electrodes[115]. Small displacements of the sensors among the sessions can create big differences in the EMG traces and consequently reduce significantly the accuracy

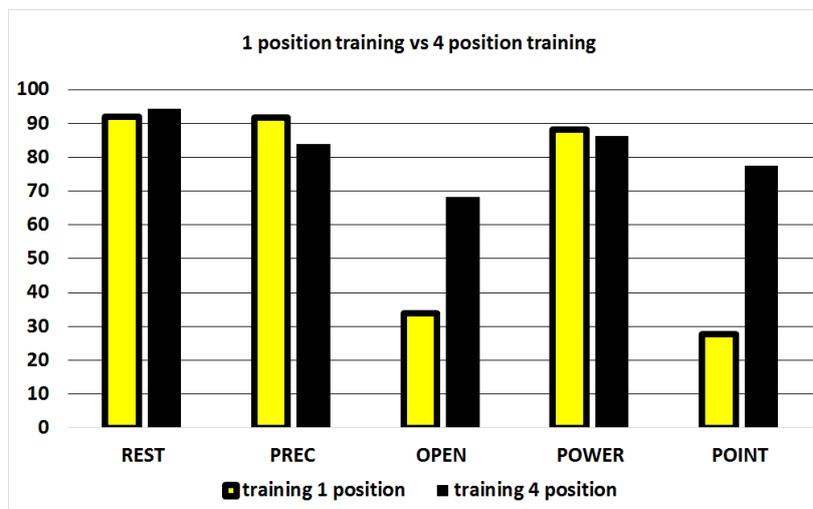


Fig. 2.10 single vs multiple position training .

of the classification. The difference of classification performance in this work considering the training made on a single session or on two different sessions. The work of [91] uses an 8 electrodes system and evaluates the performance of the system in 3 different sessions. The proposed system has 8 electrodes and the set of gestures chosen targets HCI applications, instead of our daily use grasping types. Nevertheless the results of this work can confirm the decreasing trend of the performance caused by misplacement. The SVM in this experiment is trained by merging the support vectors determined in the previous tests and the vectors of an additional session. With this procedure, it is possible to observe the behaviour of the recognition accuracy dependent from the two sessions. The first test uses the complete training session of the last paragraph to evaluate the accuracy of the classifier among the different sessions. The sequences of the movements are coded in the acquisition protocol and for each trace the different movements can be located. The prediction of the SVM algorithm is used on all the traces, and the mean value of accuracy are collected for each patient. The single training session cannot satisfy minimum requirements for reliability of classification. The differences of accuracy in the 5 gestures are shown in Figure 2.11.

The procedure to place the electrodes, described in session 2, is standardized and repeated for all subjects; however still little inconsistency in position of the electrodes can cause big differences in muscular activation pattern. To cope with this issue, we evaluate the use of multi-stage training and compare the accuracy per arm positions in single versus double training session. The extended training merges two training sessions to create a unique training model. This kind of training method enhances the performance of the classification accuracy as shown in Figure 2.12.

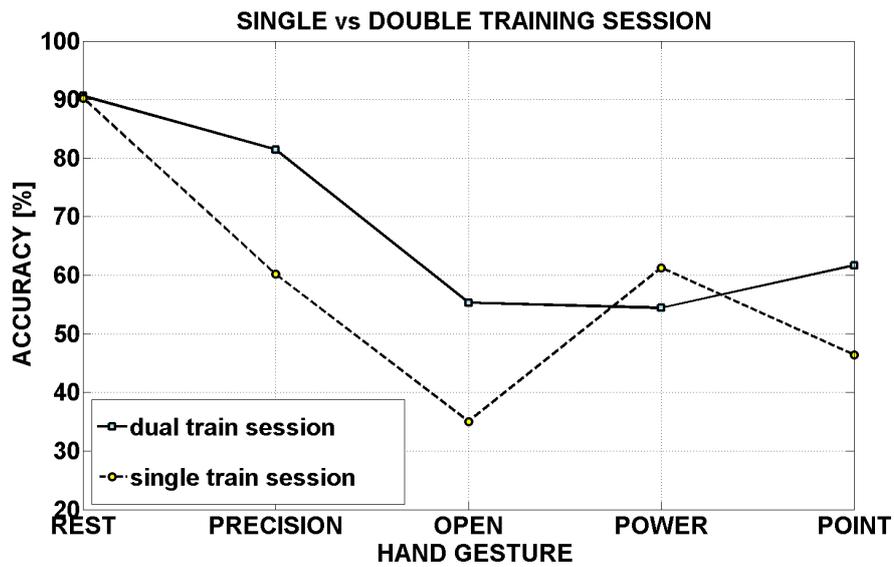


Fig. 2.11 Single vs double training session.

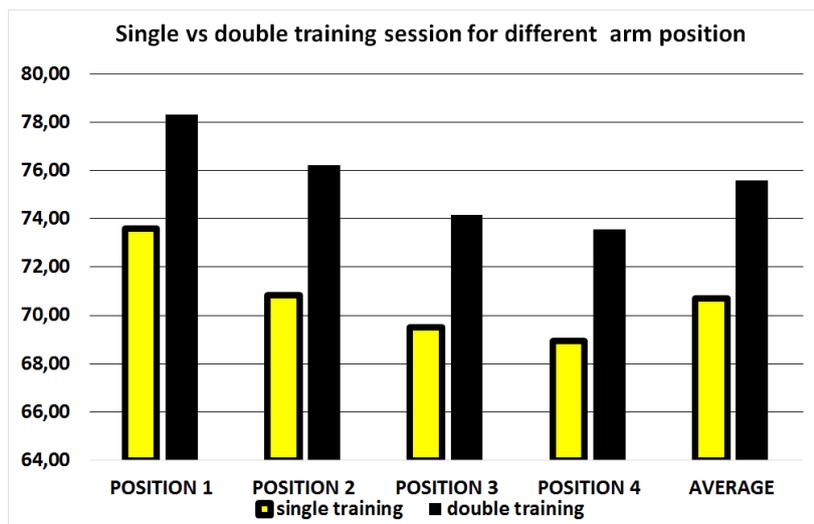


Fig. 2.12 Single vs double training session in different positions.

2.7 Discussion

The results of the experiment show that a robust classification for EMG signals cannot be achieved without considering the issues coming from the electrodes placement, which can present differences, even if slight, from one use to another and/or day by day. Furthermore, this work outlined the differences in classification accuracy, which occur for the same gesture performed in diverse positions of the arm (parallel to the ground, proximal to the body, lifted upwards, etc.). The idea behind this work is that multiple sessions training is necessary

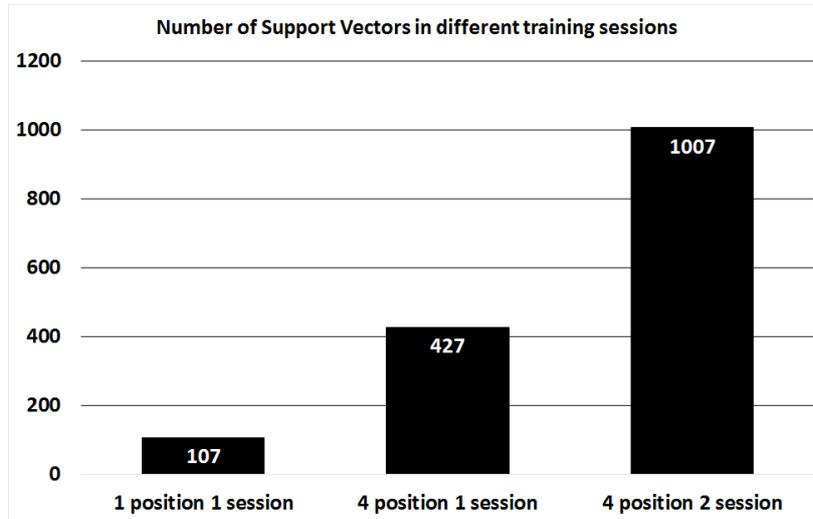


Fig. 2.13 Support vectors number in different testing conditions.

for a realistic application and a SVM with light signal filtering can give good accuracy in recognition of activation of muscular patterns.

The proposed setup is ideal for an embedded implementation, because the hardware setup is simple and the signal processing is light weight, suitable on a microcontroller. The EMG signal is filtered and de-noised directly by the sensors, and the hardware signal conditioning allows to have well differentiated patterns even with a four sensors setup and small digital filtering.

The SVM training algorithm is quite heavy from the computational point of view and runs on a PC application offline, but it is possible to implement the classification on embedded platform maintaining response time compatible with the use of a prosthesis. The computational time of the prediction function is in the most part dependent from the number of Support Vectors present in the model. More support vectors indicates a complex model due to overlapping of different patterns. The choice of an appropriate training set and the tuning of the algorithm parameters can reduce the number of support vectors. Obviously the model becomes more complicated when multiple sessions and multiples arm positions are considered.

Figure 2.13 shows the number of support vectors in different cases of the experiment.

The platform used for the implementation of the SVM classification algorithm is an ARM Cortex M4, with 100MHz clock. The execution time of the classification routine is measured to understand if the usage of an embedded low cost platform can satisfy the specification of the system. The test is performed with different models, changing the kernel type and the number of the support vectors, to evaluate the trade-off between complexity and

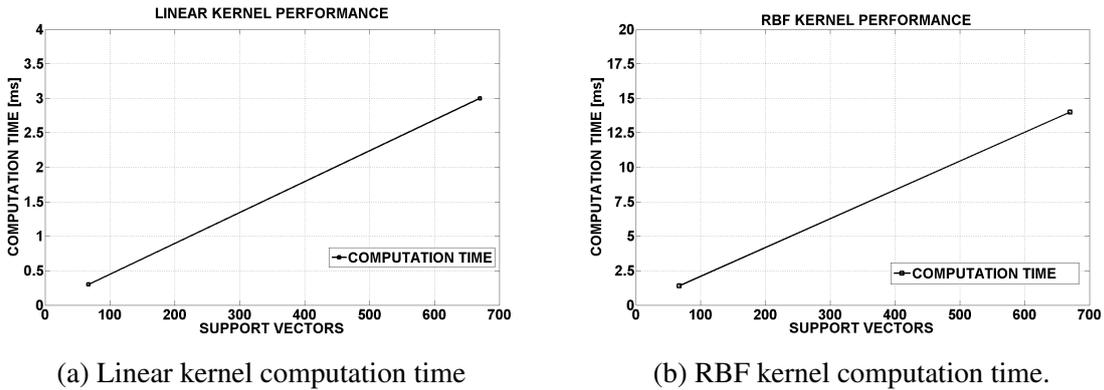


Fig. 2.14 Different Kernel Computational time vs number of SVs

response time of the algorithm. Figure 2.14a and 2.14b shows the computation time obtained with FPU (Floating point unit) calculations. The response time allows to use the embedded platform to implement an active control of the prosthesis, with linear or RBF (Radial Basis Function) kernels. The number of support vectors can be reduced with an appropriate training sample selection and with the tuning of model made on the single subject. In this experiment we did not perform optimization of the parameters or differentiated selection of the samples in creating the training models. This choice is intended to show the starting point in implementing an embedded application and wants to contribute in evidencing and focusing on the problems and the issues that become critical in the implementation of this kind of devices.

2.8 Transient Elimination

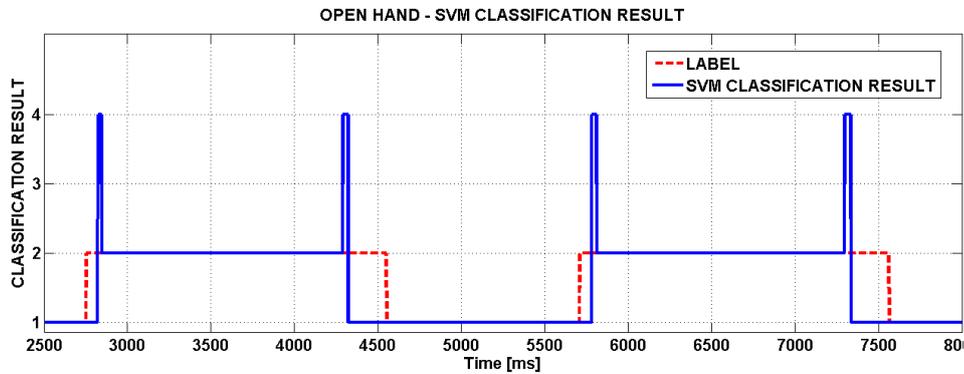


Fig. 2.15 Transient errors

Although the classification accuracies obtained with SVM is around 90%, in a real-world situation the accuracy typically drops with use. Losing control of a prosthesis would not only be undesirable but unsafe (for example when the prosthesis is used to hold a cup of hot coffee).

To get insight on how to improve the classification accuracy, we tried to determine which traits of the signal are more likely to be misclassified. A myoelectric signal can in fact be partitioned in two different states with respect to the activity of the muscle from which it is generated. The steady state corresponds to the periods of time during which the muscle is maintained in a constant contraction/relaxation, while the transient corresponds to the periods of time during which the muscle goes from rest to a voluntary contraction or vice versa [34]. From what we observed, the steady-state data can be classified much more accurately than transient data; this observation is also confirmed by the work of Englehart et al. [48]. In real-time or on-line applications, sliding windows with significant overlap and voting strategies can minimize the classifier output error [89]. Such approach can improve classification accuracy; however the system loses flexibility since the windows length [47] are fixed and require a great amount of extra-computations. To improve the classification accuracy while maintaining the application suitable for a real-time embedded implementation, a time-series classifier with a low computational overhead can be used to detect transitions allowing SVM classification only on steady state gestures. In this paragraph we present a hybrid pattern recognition approach using Hidden Markov Models (HMM) and SVM. It focuses on the advantage of such combined approach in gesture recognition for prosthetics, independently from further improvements to classifier performance obtained by parameters

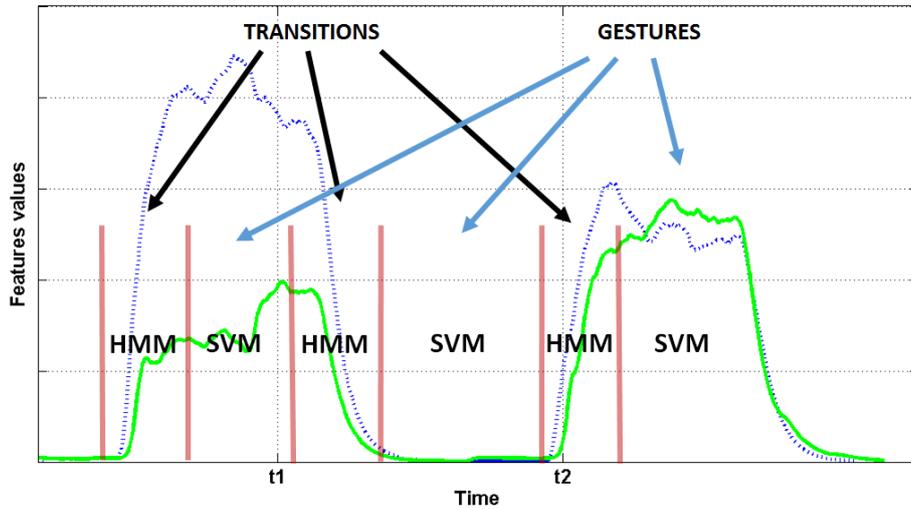


Fig. 2.16 Classification approach.

optimization or feature extraction. HMMs is widely used in speech recognition [66] where the signal can be viewed as a piecewise stationary signal while the SVM, that lacks the ability to model temporal dependencies, can be successfully used to classify the gestures in steady state. HMM is in fact a statistical model widely used in applications, whose capability to model spatio-temporal series is preferable to other signal processing approaches. It well captures dependencies between consecutive measurements.

The adoption of the hybrid classifier granted an increase in accuracy of the gesture recognition and the low computational complexity of HMMs allows the described implementation to be suitable for an embedded system. We therefore explore a solution based on HMM to distinguish between steady-state data from transient data and then use SVM only to classify the EMG signal during steady-state.

In literature there are some interesting works on the HMM used in EMG classification. [39] presents good classification accuracy around 90% on a dataset with 12 subjects. The limit of this work is that the recognition is performed only on a few set of gestures, i.e. pronation, supination and rotations of the wrist, with the aim to map them on different gestures on the prosthesis. Such approach is not suitable for a natural gesture interface, where the intended gestures shall correspond to the effective ones on the artificial limb. The work [67] combines the HMM and MLP (Multi-Layer Perceptron) to improve robustness of the classification. Once again, the system classifies prono-supinations and rotations of the wrist not usable in natural gesture prosthesis. Furthermore, acquisitions from only one subject compose the dataset in a single session so that it is difficult to understand the robustness of the system.

Our final target is the design of a natural control system that can be implemented on a real world prosthetic arm and for this reason we excluded from our dataset the prono-supination and wrist rotations, while we focused on grasping and opening movements. Aiming to have a better understanding on the impact of transients on the classification error, we get insight on the role of transients in the generation of errors during classification; in particular, more than 60% of the error is generated during transients classification. In Fig. 2.16 our classification approach is presented. The combined adoption of the HMM and of the SVM allowed us to obtain an increase of 8% in the overall accuracy with an average increase of 12.5% for gestures classification.

2.8.1 HMM

The HMM is a statistical algorithm used in modelling generative sequences characterized by an underlying process generating an observable sequence. The HMM is a framework of the Markov processes that are used to describe the temporal evolution of a system S that can be in one of N states S_1, S_2, \dots, S_N .

The system change its state depending on the probability associated to each state even though the states in many cases are not directly observable. Hence they must be considered as hidden random variables that generate an observable output. The HMM describes observations $O = \{o_1, o_2, \dots, o_t\}$ with their states $Q = \{q_1, q_2, \dots, q_t\}$. The two fundamental hypothesis for the HMM are:

1. The state of the system and any given instant t depends only on the state at instant $t - 1$

$$p(q_t | q_{t-1}, o_{t-1}, q_{t-2}, o_{t-2}, \dots, q_1, o_1) = p(q_t | q_{t-1}) \quad (2.3)$$

2. The observations O at any instant t depends only on the state at the instant t

$$p(o_t | q_t, q_{t-1}, o_{t-1}, q_{t-2}, o_{t-2}, \dots, q_1, o_1) = p(o_t | q_t) \quad (2.4)$$

The basic elements to define the HMM problem are:

- A set of N hidden states, $S = \{s_1, s_2, \dots, s_N\}$. Often these states are related to physical phenomena.
- A set of M distinct observation symbols $V = \{v_1, v_2, \dots, v_M\}$. These represent the values of the observations as output of the system.

- The state transition matrix $A = \{a_{ij}\}$, where a_{ij} is the probability of making a transition from state s_i to state s_j : $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$, with $1 \leq i, j \leq N$ and where q_t denotes the state at time t .
- The observation symbol probability distribution matrix $B = \{b_j(k)\}$, where $b_j(k)$ is the probability of emitting v_k in state s_j at time t :
 $b_j(k) = P(o_t = v_k | q_t = s_j)$, with $1 \leq j \leq M$; $1 \leq k \leq M$ and where o_t denotes the observation at time t .
- The initial state distribution matrix, $\Pi = (\pi_i)$ where π_i is the probability that s_i is the initial state: $\pi_i = P(q_1 = s_i)$, with $1 \leq N$.

Following the convention, an HMM can be expressed as:

$$\lambda = (A, B, \pi) \quad (2.5)$$

The transition probability matrix (A) and the observation symbol probability distribution matrix (B) that describe the HMMs are computed using the Matlab function *hmmestimate*. Before training the HMM, each training set is quantized into 12 levels; the same is done with the test-sets before the classification. With 4 states and 12 possible emissions for each state, matrix A results to be 4x4 and matrix B 4x12. The definition of the matrices is performed offline with Matla, while the on line recognition is calculated with the multiplication of the A and B matrices and does not affect the real time computing requirements.

2.8.2 Hybrid Classification

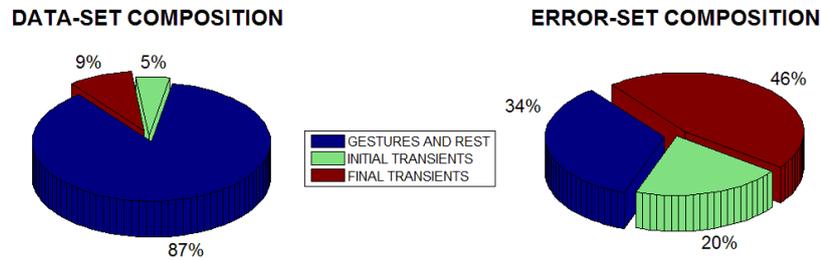


Fig. 2.17 Error Statistics

The general average accuracy obtained with the gesture classification in the first hand position is 84.0%. It has to be considered that the cross-validation of the dataset is not performed in this test to point out the principal source of misclassification. By comparing the classification result with the label, it can be seen that the errors are concentrated at the beginning and at the end of each repetition, generally corresponding to the initial and final transients. The fact that during the transients the probability of a misclassification is much higher is put in evidence by Fig. 2.17. The first pie represents the distribution of all the samples of the datasets; as it can be seen, the samples that belong to the transients correspond to a small percentage of the total. In the second pie the distribution of all the samples that were misclassified by the SVM is shown; even though the two transient phases together represent only the 14% of the datasets, 66% of the error is concentrated in these two phases. The second aim of the experiment is to implement a hybrid HMM-SVM classifier and evaluate its performance. Following the same procedure as the one described previously, each dataset is labelled and split in two parts: training set (30% of the dataset), and test set. Both classifiers are trained with the same training set but using different labels. The SVM, as previously described, is trained to distinguish between the various gestures, so a label with values from 1 to 5 corresponding to the hand gestures is used. The HMM classifier is instead used to discern between 4 classes, corresponding to 4 phases: rest, initial transient, gesture and final transient. The output of the HMM is then used to activate the SVM classifier during the phases of rest and gesture and deactivate it during the transients. A four-state HMM is trained, with states 1, 2, 3 and 4 denoting the rest, initial transient, gesture and final transient states, respectively. Regarding the classification, the Viterbi algorithm [51] is chosen to recover the single most likely state sequence from a sequence of observations associated with the trained HMM. The classifier correctly detects all of the 1260 gestures tested; due to the noise, false positives are generated in a rate of one for every 126 gestures that are correctly

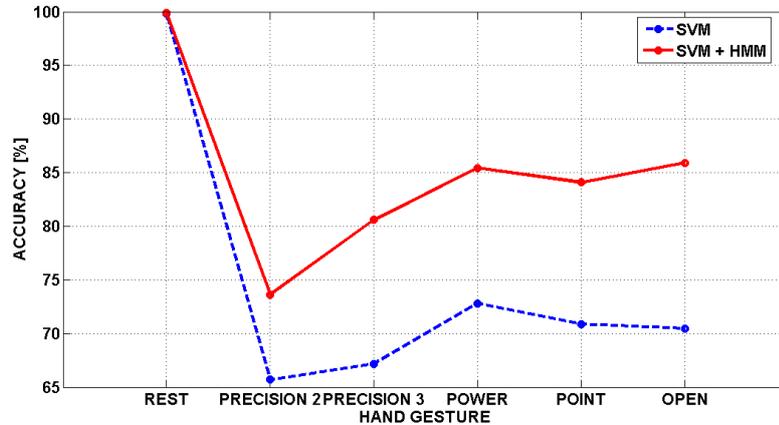


Fig. 2.18 Comparison of the average accuracies for each gesture of the two classifiers.

classified.

Following the same procedure described for the previous experiments, a SVM classifier is trained on the same training-set used for the training of the HMM. The SVM is then used to classify the test-sets, but instead of classifying all the samples, it is activated only for the samples classified as “gesture” or “rest” by the HMM. The HMM acts as a trigger for the SVM, turning on the SVM classification only when the HMM classification is "gesture" or "rest". The HMM is "always on" but the output of the classification is given by the SVM output when a stable gesture is detected. This avoid glitch in the classification, since an output from the SVM is required when the HMM detects a steady state of the gestures.

The accuracy in the classification of these samples is then computed and compared with the accuracy previously obtained. The general average accuracy is 91.8%. Figure 2.18 shows a comparison between the average classification accuracies obtained for each gesture with the SVM classifier (blue dashed line) and the average classification accuracies obtained for each gesture with the hybrid HMM-SVM classifier (red continue line). The values presented are the result of the mean values of accuracy for all the subjects involved in the experiment.

2.9 Real Time Requirements

The goal of the previous experimental phase was the identification of the causes of error in SVM classification of EMG signals. In general, a classification error derives from the confusion of two classes. This can happen either because the input of the classifier does not contain enough information to enable the classifier to discern between the two classes or because the classifier itself is unable to exploit all the information brought by its input signal.

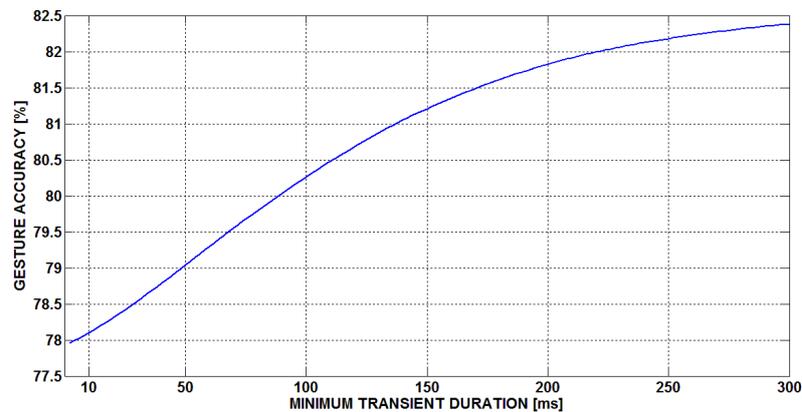


Fig. 2.19 Effect of the initial transient duration boundary on the gestures classification accuracy.

The type of classifier, the selection of the training-set, the number of classes and the choice of the sensors and features to use are all factors that can affect directly the overall classification accuracy. For instance, by adding a single posture to the classification with respect to [23], the overall classification accuracy drops significantly; it is in fact comprehensible that, when adding new classes while maintaining the same number of features the probability of misclassifications grows.

What emerges from the results of this analysis, however, is that a considerable portion of the misclassifications happens when trying to classify the EMG signals during transients. During these phases, the EMG signals are in fact quite similar for all the hand gestures and, when using classification based on features related only to a specific time instant, the identification of the posture generating the signal is often prone to error.

In particular, in the described experiment, the 66% of all the classification errors occurred during the transients. The exclusion of the transients' samples from the classification can therefore bring a drop in the error-rate, which is quite desirable in prosthetics, where robustness plays a very important role.

The exclusion of the transients from the SVM classification can be operated by means of a time-series classifier. In our case the HMM was preferred to other more sophisticated algorithms because of its simplicity. The SVM classification is in fact relatively heavy from the computational point of view and has to be coupled with a computationally-light time-series algorithm to allow an embedded implementation that can maintain a response time compatible with the use of a prosthesis.

The results of the second experiment show that with the HMM-SVM approach, a consistent increase in the classification accuracy is achieved. In particular, while the accuracy of the rest

classification remained over the 99%, the average accuracy in the other gestures classification gained more than 12.5%.

By excluding the initial transients from the SVM classification, however, an additional delay in the control of a hand prosthesis can be introduced whenever the muscles monitored from the EMG sensors start a contraction. To limit this delay it is possible to impose a boundary to the duration of the initial transients detected by the HMM.

In general, minimizing the control delay is a priority of a prosthetic system; however, it is also important to preserve the robustness of the control system, to guarantee precision in the use of the prosthesis during every day activities. The upper boundary on the duration of the initial transients can therefore be chosen taking into account the trade-off between the classification accuracy and the introduced delay because our work targets strict real-time requirements, since in the final system the gesture recognition enables the artificial limb movement.

Figure 2.19 represents this trade-off, having on the x-axis the initial transient duration boundary and on the y-axis the gesture classification accuracy (the rest classification is excluded from the accuracy calculation since it is not affected by the changes in duration of the initial transients). In this application, we impose an upper boundary of 300ms, corresponding to the maximum prosthesis reaction time that can be considered acceptable by the amputee for a usable prosthesis control [23].

2.10 Conclusions

In this analysis we investigated the classification accuracy in a real application of an EMG interface. We analyzed the acquisition interface, intended as the choice of the kind of sensor used, placement strategy, the training policies. We proved that 4 sensors correctly placed and a light signal processing can give high classification accuracy, comparable with systems with more sensors and a signal processing with higher computational cost. It is not possible now to obtain good performance without considering the difference of classification in different arm positions and among multiple sessions, if EMG surface sensors, which can be removed, are used. The strategy of a double training session, in different days, is compatible with the clinical scenario, because it is normal for a patient with an upper limb prosthesis to have periodical check-up of the prosthesis with technicians and doctors. Moreover we get insights on the causes of misclassifications. The majority of the classification errors occurred during the transients between rest position and gesture. By further analyzing this issue, we saw that high accuracy cannot be reached by a time independent algorithm and the adoption

of a combined time-series approach, such as the use of HMM, improves the classification accuracy.

Chapter 3

Prosthetic Controller Design

3.1 Overview

There are an estimated 2 millions of hand amputees in the United States and approximately the same in Europe. More than 200.000 new amputation surgeries are performed each year and approximately 10.000 children receive amputations resulting in a lifelong disability [11].

Amputee patients are supported by a long standing research and development of prosthetic devices, which can be divided in passive and active ones. The passive prostheses, firstly developed, have only a cosmetic purpose and do not support any of the hand functionalities.

Since the hand prosthesis acts as functional replacements of the missing limb in amputees, technological advancements over the years spent efforts to give significant improvements in the restoration of the functionalities of a normal human hand. The first example of such advancement was the body-powered prostheses, capable of restoring basic tasks such as opening and closing a terminal device [19]. In such devices, motion is transmitted to the prosthesis mechanically, controlling the artificial hand with the abduction of the shoulder or the healthy wrist flexion. These devices are often used because they are simple, robust, and relatively inexpensive, even though they do not re-establish a complete natural control of the hand.

Electrically powered or active prostheses [105] are advantageous w.r.t. body-powered ones because they require less user effort, as movement is actuated with DC motors [61]. They can be controlled through a variety of means such as force sensors [28], acoustic interfaces [111] and EMG signals [99]. Such devices restore some functionality to amputees, but their control is typically limited to only one or two Degrees of Freedom (DoF).

For aesthetic and functional reasons, the research on prosthetic arms addresses multiple challenges regarding shape, size and control of the artificial hand to provide devices acceptable by the amputees. The development challenges of underactuated anthropomorphic

prosthesis varies from hand design [73] to fingers' actuation [122], in search of the best tradeoff between the complexity of the design and its usability and reliability.

Recently, the biomechatronic design of these prostheses was refined and on the market we have advanced and reliable multifinger prostheses. The Bebionic 3 [1] is a poliarticulated myoelectric hand with embedded control, which features 14 different hand positions. Each finger is independently driven by a DC motor and, exploiting its robust design, the hand can carry up to 45 kg. A dedicated wireless link allows clinicians to monitor the hand and program the user-dependent parameters, such as selection of the control sequences and grip thresholds.

The Michelangelo Hand [4] is modeled after the physiology of the human hand and it is powered by separated motors for the positioning of the thumb and the control of the fingers. The thumb, index and middle fingers are actively driven, while the other two fingers follow passively the hand movements. The controller can reach up to 7 different hand positions and thumb independent actuation allows a fine control of the palm shape.

In the I-limb hand [2] all five digits move independently, bending at the natural joints to fit around the shape of the grasped objects. Its controller can be programmed to perform the gesture in accordance with a selected contraction burst of the residual forearm muscles. The I-limb can be interfaced and controlled by a dedicated smartphone application. This mode of operation gives access to all the 24 possible grip patterns, that can be further customized by the user. These SoA devices are enabling complex functionalities, including proportional grasps and configurable gestures, but their control strategies are still based on non-intuitive codification of gestures in muscle contractions. Even though they offer high reliability and robustness in control, these systems basically use simple thresholds to detect the muscular activity and the prosthesis commands are encoded in predefined sequences of extensions and flexions of the residual forearm muscles [50]. This approach does not offer a natural interface, it requires a long learning curve and a high level of concentration during usage.

To restore a more intuitive control of the artificial hand, pattern recognition and machine learning techniques are investigated [49]. Gesture recognition myoelectric control is based on the assumption that the muscular activation patterns are repeatable among different instances of a same gesture of the hand. Hence, different gestures can be classified from their muscular activity and the scientific literature proposes a wide analysis of the best sensor configurations [118] [53] [42], classification algorithms [44, 48] [14] and actuation controls [98]. These approaches were analyzed in the previous chapter of the dissertation. However, most of the reported experimental evaluations focus only on the recognition accuracy over all the samples of the collected dataset and do not consider the possible drawbacks of the proposed strategies on the deployment of a complete solution for real time embedded prosthetic control.

Recently, an alternative research solution based on the use of *soft synergies* has been proposed [52, 25]. This solution develops the design of a robotic hand with only opening and closing capabilities, but that adapts its grasp on the object being manipulated. Its primary goal is to restore a correct proportional control of the grasping force, assuming that the round grip is the most used movement to grasp objects. The hand realized by [38] further refines this approach with the principle of *adaptive synergies* and features 19 joints controlled by a single actuator. The control of the device is moved from the EMG-based contraction codification or gesture recognition to the versatility of the mechanical design of the hand. It uses the EMG signal for the detection of the hand activation and as the proportional control of the grasping force. This solution is robust and powerful even if the hand can only open and close its adaptive grasp and the finger movements are not independent.

Another interesting approach compares linear and non linear regression methods for a proportional and simultaneous control of multiple DoF of the prosthesis [63, 58]. The aim of these works is to propose a method to overcome the limitations of a controller that can manage only one DoF at time. The regression approach differs from the classification mainly because it provides a continuous output instead of a discrete output associated to a recognized gesture. Nevertheless, the presented setup is based on a large array of sensors and the experiments included only flexion, extension and rotation of the wrist.

There is a lack of complete devices for embedded gesture recognition and control of prosthetics, which is due to the difficulties to perform system-level development and address problems ranging from signal acquisition and processing, embedded system design, applied prosthetics and machine learning. The work presented in this paper presents the development and implementation of an intuitive embedded prosthesis controller, as the result of the 2 years collaboration between 4 institutes with multidisciplinary competences, including the University of Bologna, FBK and the INAIL prosthetic center, one of the most important European institutes for prosthetics and rehabilitation.. Our work analyzes the EMG signal acquisition and integrates the use of SVM-based gesture recognition with a real-time controller to provide accurate and robust actuation of a poliarticulated prosthetic hand. The proposed approach is implemented and run on an embedded microcontroller and it is evaluated for its end-to-end gesture recognition and timely actuation capabilities.

Moreover, we validated our approach by applying it to data from the NINAPRO database [16] that represents the largest and most complete dataset for EMG gesture recognition analysis. The proposed system is designed with Commercial Of The Shelf (COTS) components and open source development tools, to make it easy to replicate all the results and to provide an useful platform for future research and product development.

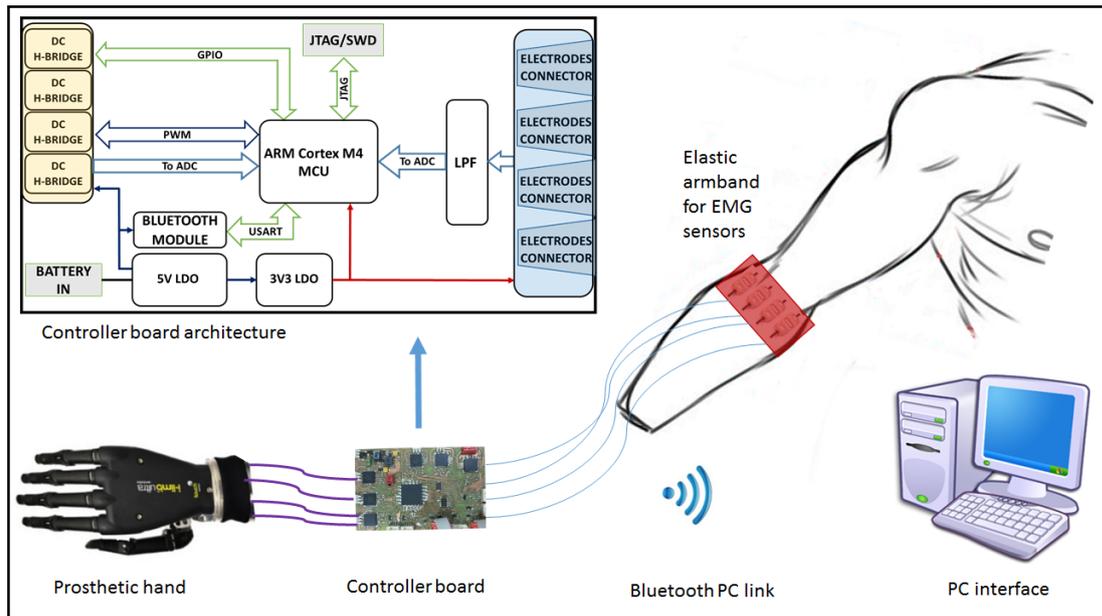


Fig. 3.1 Diagram of the system architecture

3.2 System Architecture

The proposed system is an embedded controller which acquires EMG signals from the patient's forearm for a real-time gesture-based control of a poliarticulated prosthetic hand. Offline studies of EMG gesture recognition report recognition accuracy on all collected samples, regardless of their collocation during a gesture. When actuating a prosthesis, only one output per executed gesture is needed and the gesture decision should be made as soon as possible at the start of its execution, while the classification of subsequent samples becomes unnecessary. However, the transient phase at the onset of a gesture is more difficult to classify when compared to a stable on-going contraction. Hence, a robust implementation of a gesture controller has to cope with the initial uncertainty in the recognition and to provide a timely decision for a correct actuation of the hand.

The proposed solution integrates sample-level SVM classification with a high level FSM to produce accurate and robust control of the prosthesis. For mechanical constraints, the prosthetic hands start every gesture from a rest position, i.e. open hand. Hence, we included this transition to be performed between each executed gesture and used it to improve the robustness and usability of the system. Moreover, during the on-set of a gesture, the output of the sample-level recognition is analyzed with a majority voting approach to limit the errors due to the signal transitions and to converge to a decision within a specified time window.

Fig. 3.1 presents the architecture of the proposed setup, which is composed by: (1) an elastic armband with 4 EMG sensors, (2) the main control board responsible for data acquisition and classification, prosthesis actuation and Bluetooth communication, (3) a poliarticulated prosthetic hand and (4) a PC software for data acquisition, recognition algorithm training and customization of system parameters.

3.2.1 Controller Board

The main component of the proposed setup is a custom embedded controller board, whose block diagram is illustrated in Fig. 3.1 (top). It is designed on a 4-layer PCB and includes a 32 bit MCU, circuitry for EMG signal acquisition, for the actuation and control of the hand prosthesis and a Bluetooth transceiver for the communication with a host device. The board tolerates a single-voltage power supply from 5.5 to 8V, to easily fit on commercial prosthetic systems that use standard battery cells varying from 7.2 to 8V. The on-board linear regulators provide stable output voltages of 3.3V and 5V employed for the different sub-systems.

The system is based on the Kinetis K40 MCU by NXP. It has an ARM Cortex M4 core, which includes a Floating Point Unit (FPU) with DSP instructions and fits the design of a low power system with high computational resources required for on-board real-time gesture recognition. Furthermore, ARM provides the CMSIS dedicated library for an optimized execution of math functions, which was employed in our embedded implementation of the SVM classifier. The presence of two independent 16-bit SAR ADCs (*ADC0* and *ADC1*) and of a dedicated PWM output peripheral allows the control of the proposed prosthetic hand minimizing the need for external components and hence the board complexity.

Even if the EMG sensors have a higher bandwidth, we use *ADC0* to acquire the 4 channels at a sampling frequency of 500Hz, which has been shown to be sufficient for gesture recognition applications [22]. On each channel, an RC low pass filters with cutoff frequency at 1kHz is used to minimize the high frequency electrical noise. A further resistive voltage divider protects the ADC's inputs, limiting the signal span to the 0 – 3.3V range. The ADC peripheral can also be configured to oversample the data using an internal accumulation register that stores up to 32 consecutive samples and provides their mean value as output.

The DC motors powering the finger actuation of the artificial hand are controlled by an integrated H-bridge driver (MC33926 by NXP). The connection scheme between the MCU and the driver is shown in Fig. 3.2 (top) and uses GPIO signals to configure the IC and a dedicated PWM output to chopper-control the H-bridge. In addition, to provide feedback on the status of the motors, the system uses the current monitor provided by the integrated driver, which outputs a voltage proportional to the current being drawn. The Programmable

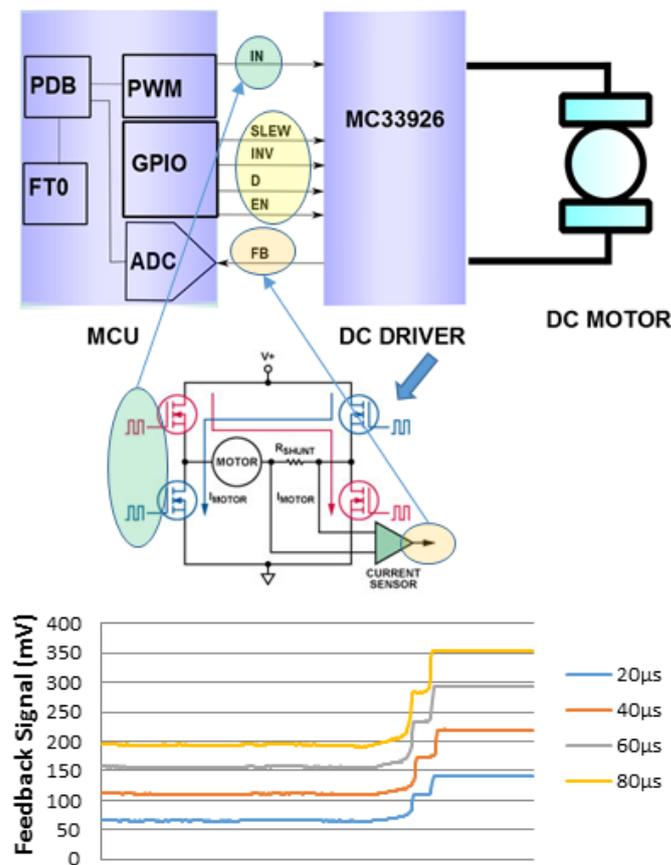


Fig. 3.2 Block diagram of the interface between the MCU and the DC motor driver (top) and example motor current absorption curves (bottom).

Delay Block (PDB) available in the MCU was hence used to synchronize the PWM output and the *ADC1* to measure such voltage only on the high level of the PWM wave, when the H-bridge and the DC motors are powered.

A finger movement is completed when it is completely open, completely closed or if it is halted in grasping an object. In such situations, the DC motor increases its current consumption, which is measured by the provided monitor and the *ADC1*. This signal and a threshold trigger are used to stop the motors without the use of an absolute encoder or other means of sensing the position of the fingers. A typical curve of the voltage provided by the H-bridge is reported in Fig. 3.2 (bottom) for different values of the PWM period.

The hardware configuration is completed with a Bluetooth (BT) transceiver (SPBT2632C2A by STM), allowing the system to communicate with external devices. The BT standard was chosen for its easy use and to facilitate communication with a variety of host devices (PCs or tablets). This bi-directional wireless interface is used to stream the acquired EMG data or store on the device customized parameters and settings. Data streaming to a host device is

employed in order to test the system and to acquire instances of gestures for offline analysis and for the training of the classification algorithm. The trained recognition models and further system settings are then sent back to the embedded device and stored in the MCU's Flash memory.

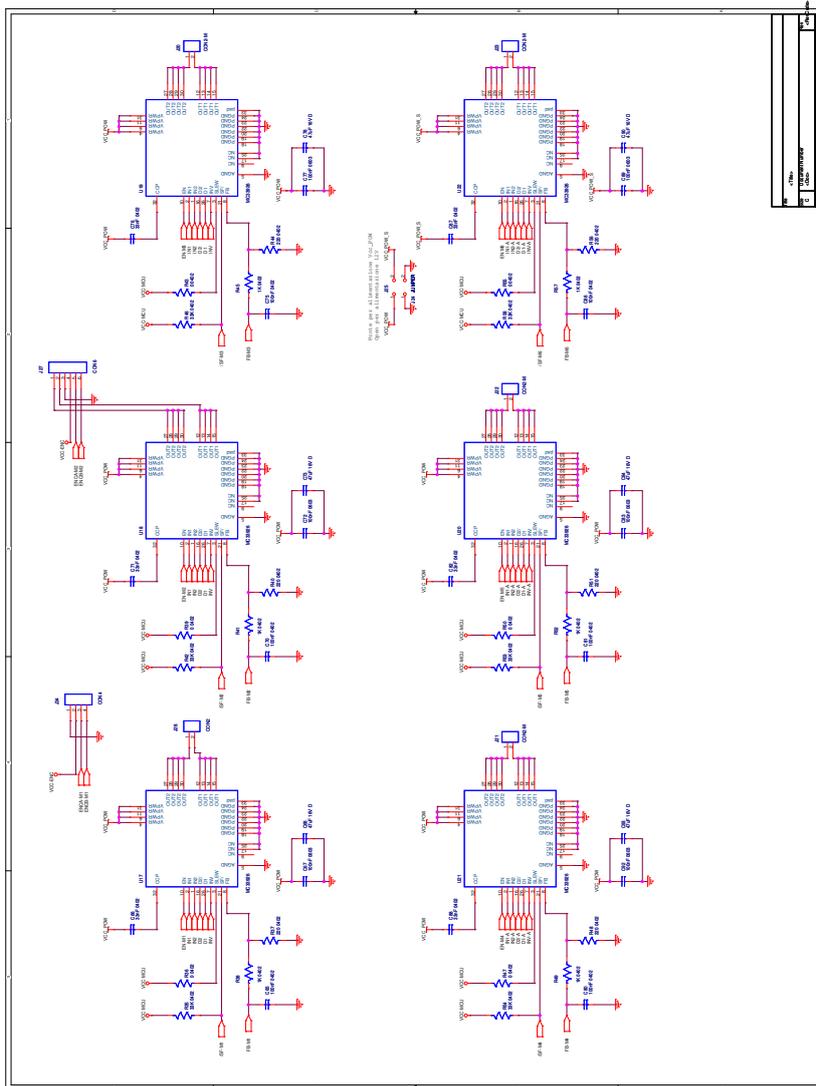


Fig. 3.3 Schematic sheet 1

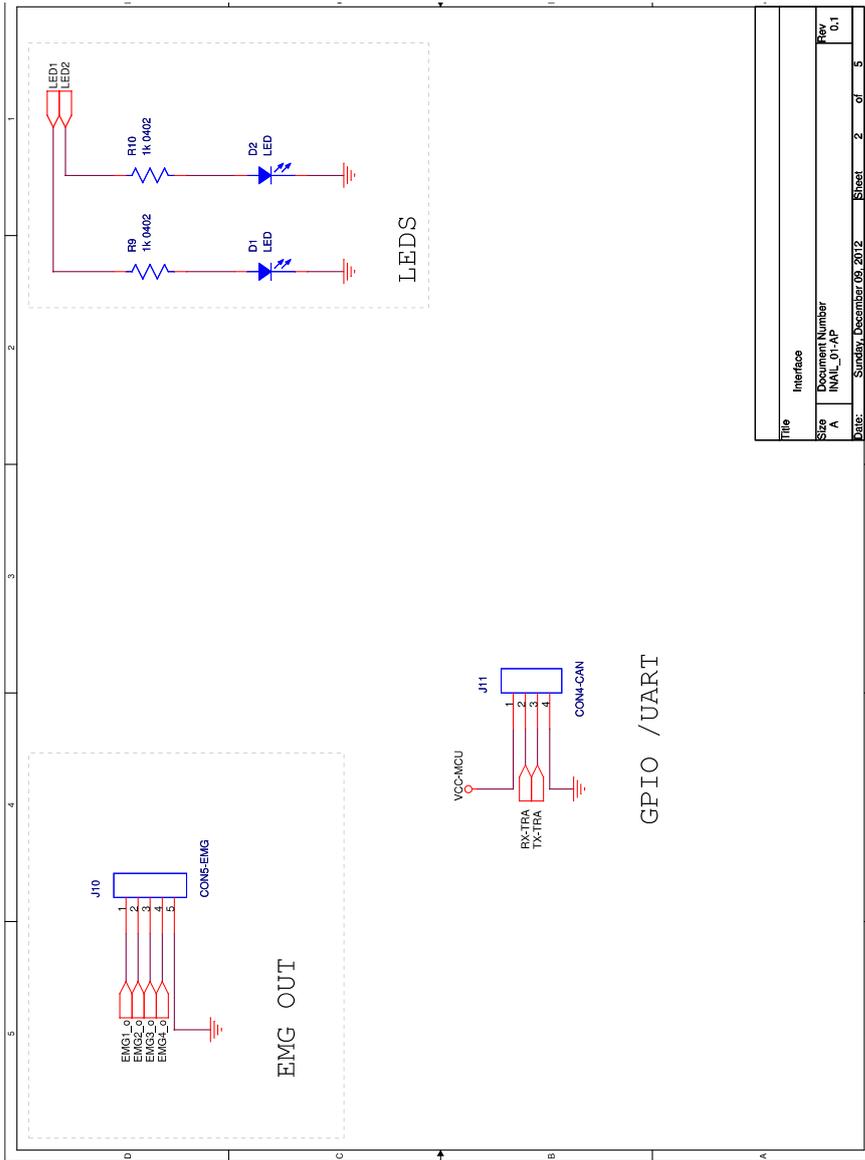


Fig. 3.4 Schematic sheet 2

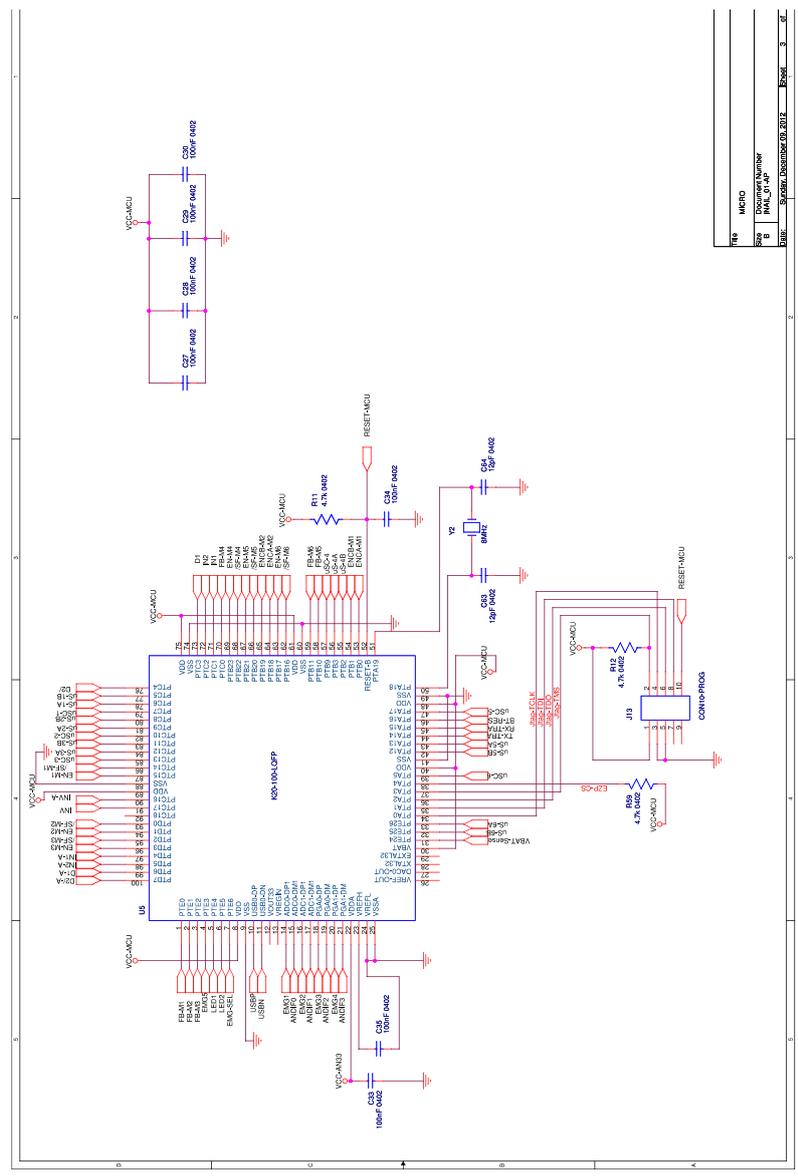


Fig. 3.5 Schematic sheet 3

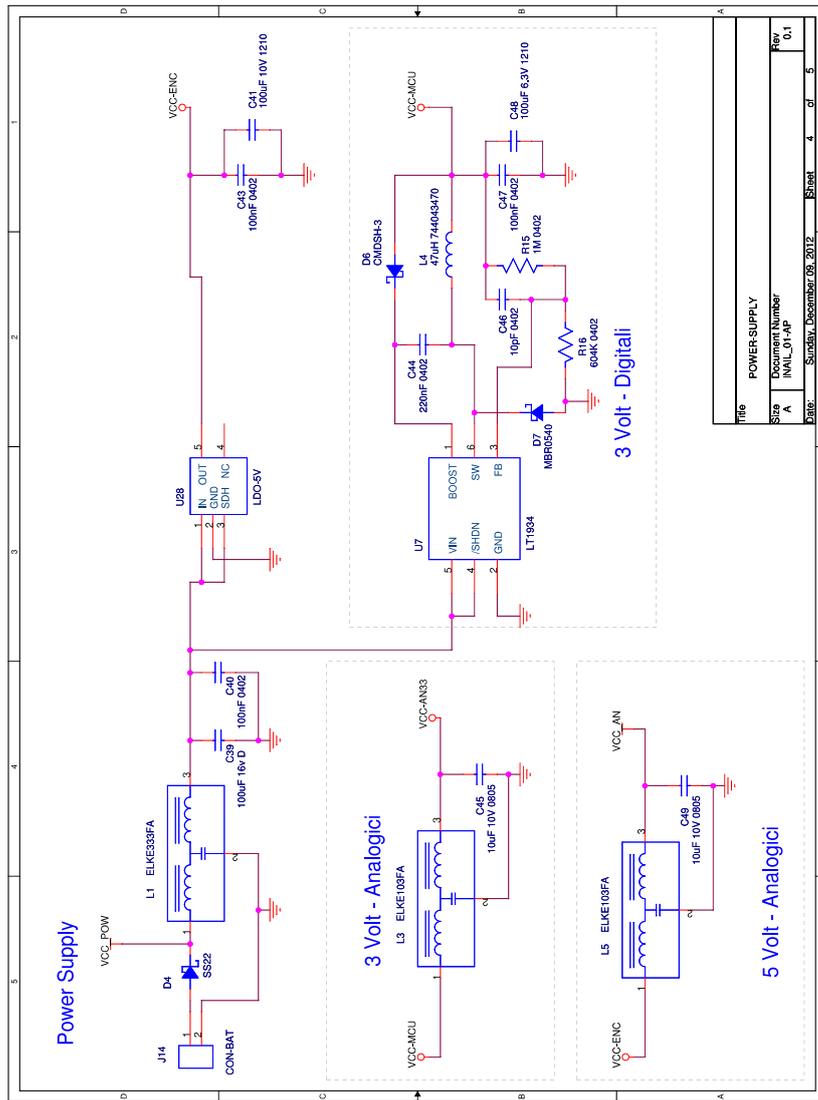
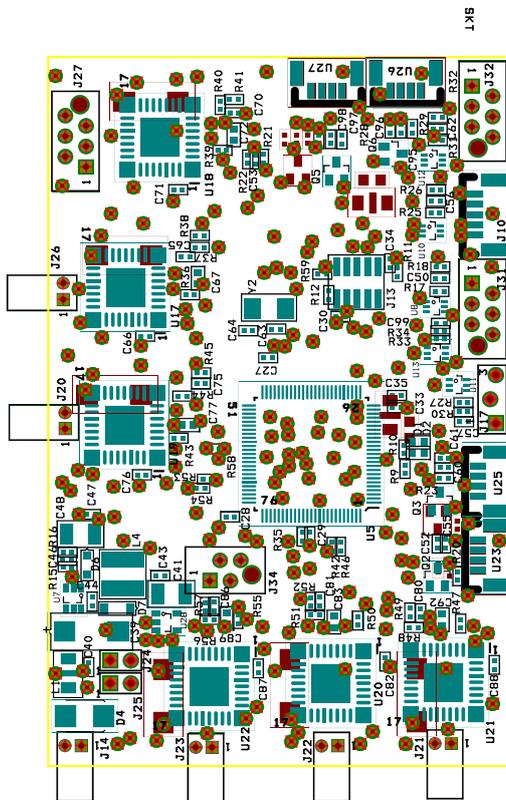
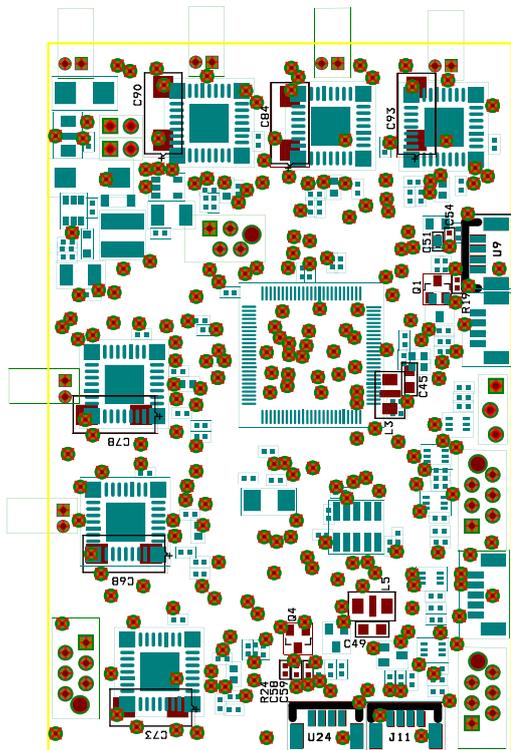


Fig. 3.6 Schematic sheet 4



DRILL CHART				
SYM	DIAM	TOL	QTY	NOTE
X	0.500 mm		248	
+	0.800 mm		34	
◇	0.900 mm		7	
■	1.800 mm		4	
TOTAL			293	

Fig. 3.8 PCB TOP



DRIFT CHAR1				
NOTE	YTQ	JOT	MAID	MY2
	845		mm 002.0	X
	34		mm 008.0	+
	1		mm 008.0	◇
	4		mm 008.1	■
	Σ32		JATOT	

Fig. 3.9 PCB BOTTOM

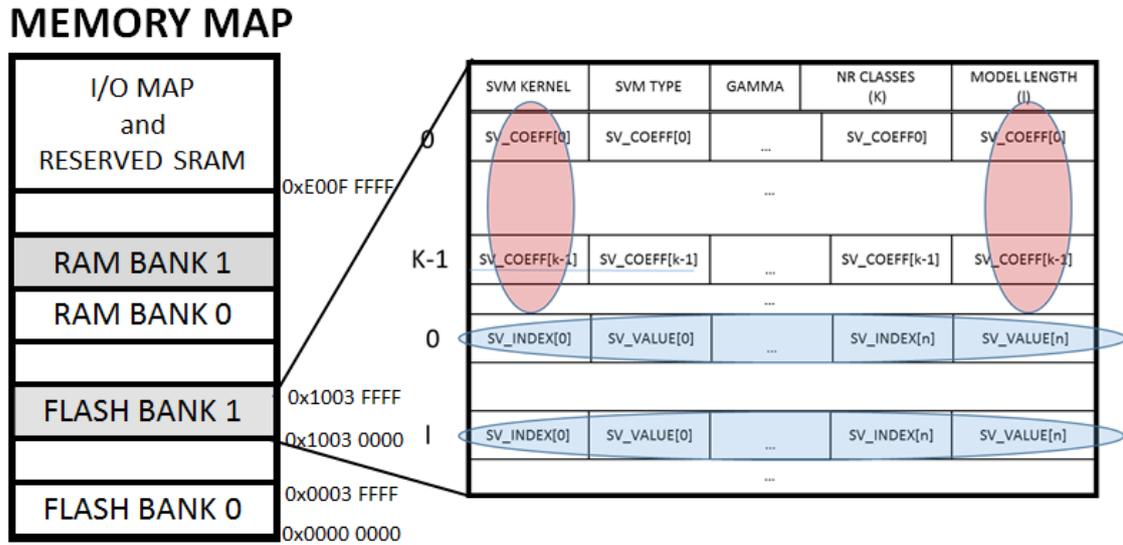


Fig. 3.10 SVM static allocation in the MCU's Flash memory

3.2.2 Embedded SVM Implementation

The training of a SVM classifier for the recognition of gestures requires the availability of a collection of labeled instances of input vectors. Moreover, it is computationally demanding, but it should be performed only at the setup of the recognition algorithm, hence it can be performed offline on a PC without imposing particular limitations. The calculated models are then stored on the embedded microcontroller where the classification algorithm is executed for a real time recognition of the performed gestures. The trained models are composed by a set of SVs which define the separation hyperplanes. Well separable gestures lead to models with a low number of SVs, while more complex situations lead to an increase of SVs needed to correctly separate the classes. The total number of SVs (N_{SV}) defines the algorithm's real-time computational and memory costs: they all need to be stored and iterated through within the decision function during real-time recognition.

The libSVM is an open source multiplatform library implementing the SVM algorithm, which includes the training and classification functions [69]. It is designed for high-end devices, where it is executed on top of an operating system, which dynamically manages the allocated data and the needed memory. In the original libSVM implementation, gesture models are dynamically allocated and initialized with the C malloc function. This dynamic memory allocation simplifies code development and optimizes resource usage, but its deployment is not recommended for embedded systems [3]. Hence, we modified the implementation of the libSVM functions to use a model that is statically stored and allocated in the Flash memory of the employed Kinetis MCU.

The Flash memory of the Kinetis microcontroller is divided in 2 separated banks of 256 kBytes each. The *Bank 0* stores the code and the vector table for the program execution. The *Bank 1* is available for non-volatile data storage and in our application it is used to store system parameters and the trained SVM models. We take advantage of the pointer manipulation and arithmetic functions offered in C to optimize the embedded implementation of the library. Moreover, we stored the SVs in a compact memory area to better exploit the structure of the libSVM's prediction function, based on nested *for* cycles iterating through the SVs. Fig. 3.10 shows the MCU's memory structure and the static allocation of the stored parameters.

The C function $svmpredict(x,y)$ implements the SVM decision function for the classification of new input data. Its first step is the computation of the RBF kernel mapping of the input vector and the SVs, which produces the array $K\langle \mathbf{x}, \mathbf{s}_i \rangle$. For an efficient implementation, the libSVM at model definition pre-computes the products between the class labels and the support values and stores their values in memory, denoted as the SV coefficients $C_i = \alpha_i y_i$. Hence, during the rel-time classification, part of the decision function had been already computed since it does not depend on the input vector. These pre-computed coefficients allow to execute the complete decision function between two classes just with 2 nested *for* loops that calculate the decision function for each combination of labels. Finally, the predicted label is the one which has won the major number of binary classifications. The pseudo-code of the implementation is reported in Algorithm 1.

The memory requirements of the algorithm can be expressed by the following formula:

$$M = Header + N_{SV}(N_F + N_{Cl} - 1) \quad (3.1)$$

where there is the contribution of a fixed header of 20 Bytes containing the algorithm's type and configuration parameters and a variable component depending on the number of SVs (N_{SV}), on the number of input data features (N_F) and on the number of recognized classes (N_{Cl}). Such variable component has to be multiplied by the size of the used data type to compute the final memory footprint (e.g. 4 Bytes for single precision float).

In our embedded implementation, the model parameters and the SV values and coefficients are stored in different memory areas, but maintaining the dependencies between the relative vector positions it is possible to access them in an efficient way. The vectors are correctly addressed exploiting the pointer algebra to substitute the allocation of a dynamic list of structures with the indexing of the Flash sector where the SVs and coefficients are stored.

Algorithm 1 Multiclass SVM implementation

```

1:  $C_i = y_i \alpha_i$ 
2:  $v[N_{CI}] = \{0, 0, \dots, 0\}$ 
3: function Class = svmpredict( $\mathbf{x}$ )
4: for i = 1 to  $N_S$  do
5:    $K\langle \mathbf{x}, \mathbf{s}_i \rangle = \exp\left(-\frac{\|\mathbf{x} - \mathbf{s}_i\|_2}{2\sigma^2}\right)$ 
6: end for
7: for j = 1 to  $N_{CI}$  do
8:   for k = j+1 to  $N_{CI}$  do
9:      $f(\mathbf{x}) = \sum_{i=1}^{N_S} C_{i(jk)} K\langle \mathbf{x}, \mathbf{s}_i \rangle - \rho$ 
10:    if  $f(\mathbf{x}) > 0$  then
11:       $v(j)++$ 
12:    else
13:       $v(k)++$ 
14:    end for
15:  end for
16: Class  $\leftarrow$  index of  $\max(v)$ 

```

3.2.3 PC Application

The PC application, written in C# on Windows, allows to connect to the system via BT and to perform a set of test and tuning operations. In particular, the embedded system implements three operation modes: Streaming Mode (SM), Update Mode (UM) and Classification Mode (CM). The transmission of a dedicated control string from the PC to the board manages the transitions between the operation modes and the CM is the default mode used for stand-alone control of the prosthesis. SM is used to stream EMG data to the PC, useful to test the positioning and functionality of the system and to collect example gestures for a personalized training of the SVM algorithm. Finally, UM allows to update the SVM model stored on the device.

The combination of the controller board and the PC application provides a complete instrument for the setup and everyday use of a gesture controlled prosthetic hand. The personalized setup of the system is performed under the supervision of a clinician and it requires the user to correctly position the sensors, test their activation and to collect a training dataset composed by a few repetition of each gesture to recognize. Each gesture is repeated 5 times, interleaving a few seconds of muscular relaxation and the PC application allows to visually inspect, segment (manually or automatically with a threshold) and label the gestures. Moreover, once the desired dataset has been collected, the training of the SVM is performed on the PC using again the libSVM to tune the classifier on the user.

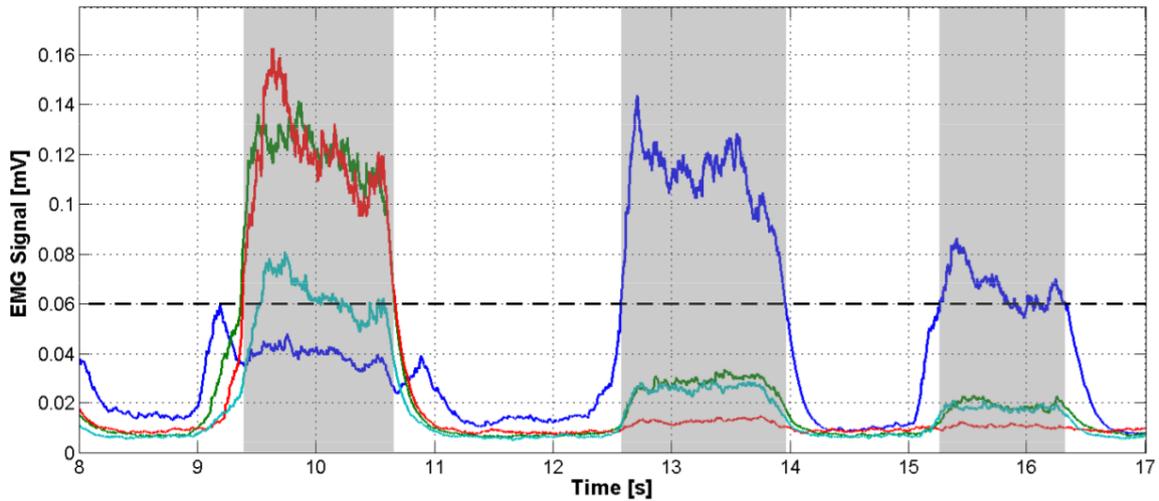


Fig. 3.11 EMG data thresholding and segmentation

An example of collected gestures, with threshold segmentation is showed in Fig. 3.11. When the SVM model is computed, the PC interface sets the system in UM and manages the model transfer to the MCU. This operation allows to correctly store the personalized model in the Flash memory of the MCU and use it for real-time classification. The communication protocol sends 3 types of packets containing the following:

1. the general configuration values: SVM type, gamma parameter, the number of classes and the number of features.
2. the model parameters: the rho parameter, the total number of SVs, and the SVs for each class.
3. the SVs and their coefficients, sending one packet for each SV.

When a packet has been sent, the interface waits the ACK sent from the board before sending the new packet. Each packet has its checksum to check the integrity of the transmitted data. Fig. 3.12 shows the transmission scheme between the PC interface and the embedded board. After the first packet is received, the embedded firmware computes the total memory occupation of the model, influenced by the number of input channels and the number of SVs and correctly allocates the needed memory. During the prosthesis control, the classification algorithm and the prosthetic controller run in real time on the embedded board, hence the interaction with the PC is not required.

3.2.4 Control Strategy

For mechanical reasons, prosthetic hands execute the various grasps and movements starting always from a reset configuration, normally the open hand position. After the execution

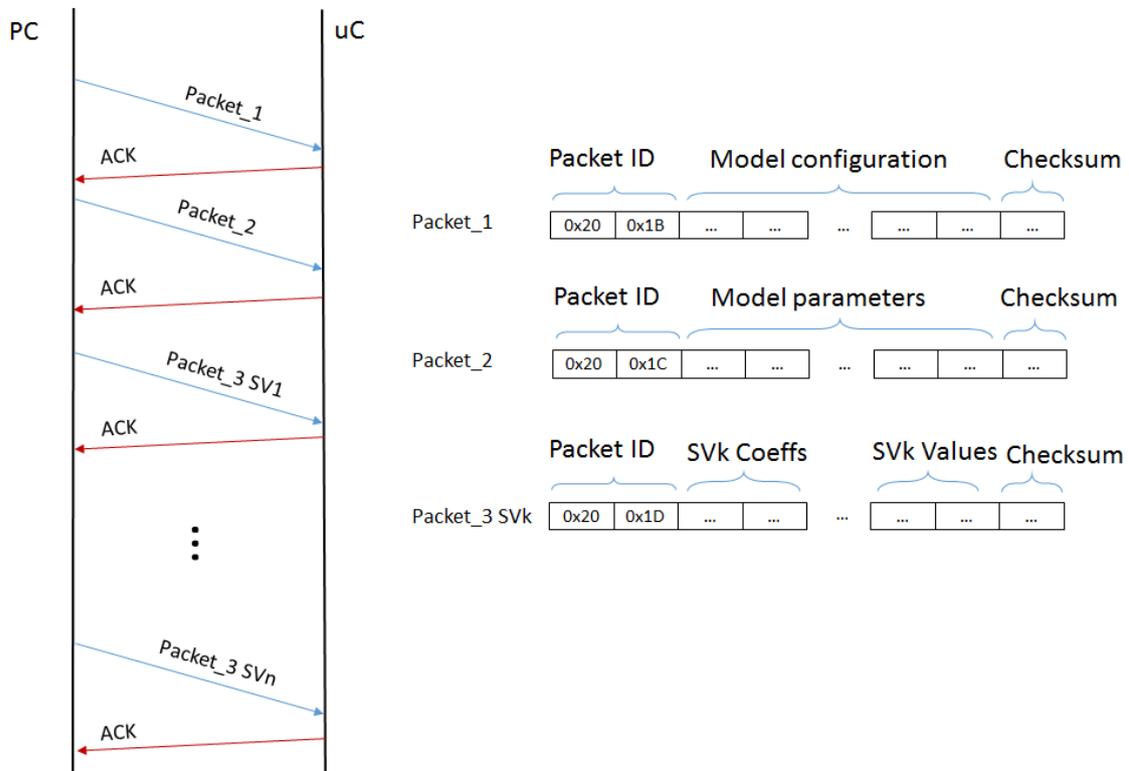


Fig. 3.12 Diagram of the communication between the PC and the embedded board.

of a gesture, it is therefore necessary to return to the reset configuration before performing another movement. This strategy allows to control the prosthesis movement using only the current feedback provided by the integrated driver, since the system always starts from a known configuration.

The diagram of the Finite State Machine (FSM) that controls the the system in CM is represented in Fig. 3.13. The ADC peripheral extracts the mean value of 16 consecutive samples of the 4 EMG channels, providing therefore a filtered signal without the need for further signal processing. When the acquired signals pass the activation threshold, the spike removal block acts as a time trigger to avoid the activation on spurious spikes due to movement artifacts or external noise. The contraction spikes lasting for less than 100ms are filtered because they are not related to voluntary hand movements. If the signal lies over the threshold for longer, the system starts the classification of the incoming samples using the SVM models stored on-board.

Since the on-set of a gesture contains the transient phase from the rest to the intended contraction, the recognition of the performed gesture is more difficult at the beginning and gets more reliable as the contraction reaches a steady state. However, a timely response of the prosthesis requires to start the actuation and execute the intended movement as soon as the

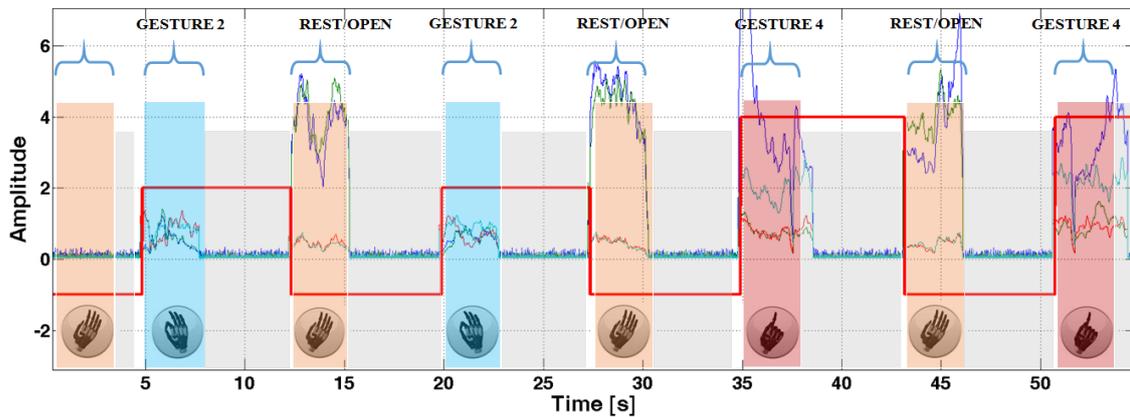


Fig. 3.14 Hand control strategy as executed during a gesture sequence.

because there are no other possible positions reachable after a gesture has been performed. Hence, it is possible to exclude the open hand gesture from the training set and simply detect any muscle contraction to exit the current gesture and reach the rest position. We evaluated the approach both including and leaving out the open hand gesture from the training dataset and we compared the results in terms of performance and efficiency.

3.3 Experimental Results

When considering real-time control of a prosthesis, the sample recognition accuracy does not evaluate the complete system, which is better evaluated by its ability to timely perform the intended user movement, with robustness to spikes, false contractions and missclassifications. We define this metrics as the *end-to-end* success ratio.

With our experiments, we demonstrate that using a proper control strategy on top of the classification algorithm improves greatly the accuracy and the robustness of the final gesture recognition. We tested the system on the data coming from 5 healthy subject and on 4 upper limb amputees of the INAIL center, using the 4 EMG channels as input vectors ($N_F = 4$). Firstly, we acquired a sequence of 10 repetitions of 5 gestures (power grasp, precision grasp, index pointed, open hand and rest position) to collect the training set, then we segmented and labeled the data to train the SVM models. We evaluated the sample-level SVM accuracy using cross-validation on the collected dataset and, finally, we tested the system on the end-to-end execution of the gestures.

The final embedded system is fully implemented and tested, however, to facilitate performance evaluation on a wide number of subjects and on benchmark datasets, we replicated the functionality of the recognition algorithm and of the controller in a simulated environment

using Matlab. Hence, we tested the system on the collected data and on data from the NINAPRO database, a recently released extensive collection of hand gestures for EMG recognition.

The end-to-end classification is evaluated on 10 repetitions of sequences composed by a gesture followed by an open hand gesture. In the collected data, the users performed a sequences with 10 repetitions of power grasp, pinch grasp and pointing index, collecting a total of 60 consecutive movements. On this sequence, we evaluated the number of wrong gestures as executed by the proposed control strategy. To have a comparison with the classical classification accuracy, an error of 1 in 10 gestures is equivalent to an error rate of 10% (i.e. 90% accuracy), while an error of 1 in 60 gestures corresponds to an error rate of 1.66% (98.4% accuracy).

Tables 3.1 and 3.2 show the results of the end-to-end execution accuracy and the resulting number of SVs in the trained models on healthy and amputee subjects. The first column shows the cross-validation recognition accuracy calculated on all the samples of the training dataset, which confirms that the classifier performance is aligned with the SoA. Under the columns labeled *COMPLETE*, we show the number of SVs and the end-to-end error rates using a SVM model that includes the open hand gesture to be recognized ($N_{Cl} = 5$). In the *REDUCED* case, we excluded the open hand gesture and simply assume such gesture is performed when a contraction is detected after any other gesture ($N_{Cl} = 4$). We can observe that the error rate is minimal and maintained with both approaches, while the mean number of SVs decreases from 296 to 179 for healthy subjects and from 534 to 278 for amputees.

The Ninapro database collects up to 52 different hand gestures from 27 subjects, recorded with an accurate setup using a hand-tracking glove and 10 Ottobock EMG sensors on the forearm. From such dataset, we selected the same gestures as the ones used in our application and we constructed again an interleaved dataset inserting open gestures between the others. As in our setup, we used the 10 EMG channels as the input vectors ($N_F = 10$) and we selected data from healthy subjects to be compared with our healthy subjects.

The evaluation of the proposed control strategy on such data is reported in Table 3.3, showing similar values of accuracy and of end-to-end classification rates for the complete and reduced approaches. Our solution with only 4 EMG sensors shows comparable results to the one obtained with the Ninapro dataset, where EMG acquisition is based on an array of 10 sensors, hence more complex and expensive.

Maintaining a low complexity is a key point in the development of an efficient embedded system and its design and implementation should be tailored to optimally employ the hardware resources. The number of EMG channels, and hence the dimensionality of the input data vectors, beside the impact on the system cost has a major impact on the amount of the

Table 3.1 Healthy Subjects

Subject	COMPLETE			REDUCED	
	Accuracy	SVs	FSM errors	SVs	FSM errors
S1	88.01	178	0	124	0
S2	89.76	312	0	209	0
S3	89.21	229	0	155	0
S4	86.34	404	0	204	0
S5	83.49	311	0	206	0
MEAN	87.37	296	0	179	0

Table 3.2 INAIL Patients

Subject	COMPLETE			REDUCED	
	Accuracy	SVs	FSM errors	SVs	FSM errors
S1	94.86	166	0	55	0
S2	93.65	262	0	38	0
S3	81.38	393	1	290	0
S4	86.43	1316	1	729	1
MEAN	89.09	534	< 1	278	<1

Table 3.3 NINAPRO Subjects

Subject	COMPLETE			REDUCED	
	Accuracy	SVs	FSM errors	SVs	FSM errors
S1	90.46	142	0	116	0
S2	93.49	116	4	107	4
S3	92.89	132	0	67	0
S4	92.15	226	0	169	0
S5	90.15	158	0	114	0
MEAN	91.83	155	< 1	112	< 1

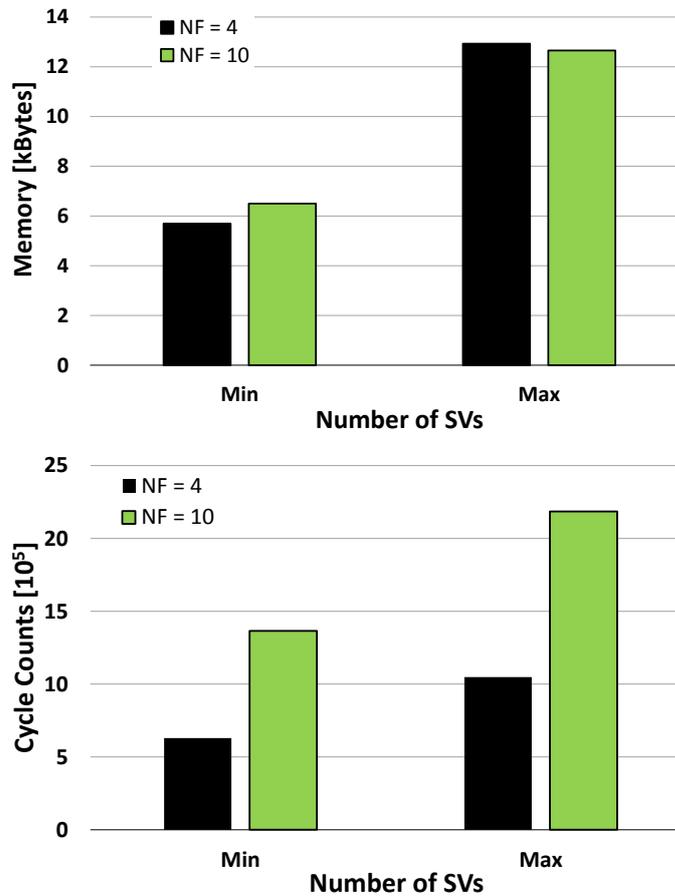


Fig. 3.15 Memory footprint of the SVM models (top) and computation requirements expressed in clock cycles (bottom) for the two analyzed setups.

required memory and on the computational load of the recognition algorithm. The proposed SVM is implemented using single precision floating point representation for the SVs, their coefficients and the input signals. This directly impacts on the hardware resources needed for model storage and real-time execution, which were evaluated on the ARM Cortex M4.

The memory footprint of the application is dominated by the SVM model composed by the SVs and the pre-computed coefficients C_i and it is given by the Eq. 3.1. The minimum and maximum memory requirements given by the trained models for the two analyzed setups (our and NINAPRO healthy subjects) are reported in Fig. 3.15 (top). The two solutions have very similar memory occupation and this is given by their combinations of input features and the resulting SVs. Our solution has a smaller number of input features ($N_F = 4$) and results in models with a higher number of SVs ($N_S = 178 \div 404$). In contrast, the NINAPRO setup has a higher number of input channels ($N_F = 10$) and results in models with a lower number

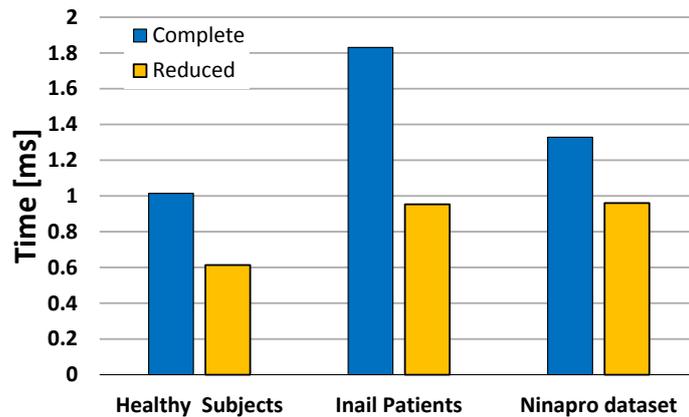


Fig. 3.16 Computation time at each input for the different setups and training datasets.

of SVs ($N_S = 116 \div 226$). Overall the two have similar memory occupation of up to 12 kB, which can be handled by the proposed MCU equipped with 64 kB of Flash.

The run-time computational load of the application is composed by the signal acquisition, the SVM prediction and the control FSM. The SVM prediction dominates the workload and sets the requirements for real-time execution. Fig. 3.15 (bottom) shows its execution time, expressed in CPU cycle counts, as measured using the MCU's internal CYCCNT register. We analyzed the two setups, having input vectors of 4 and 10 and we again measured the performance for the best and worst cases (min and max number of SVs). The computational times are again influenced by the number of input features, the number of recognized gestures and the resulting number of SVs, but this time the obtained combinations do not result in similar requirements. In fact, the proposed approach outperforms the NINAPRO setup and leads to a reduced computation time in all combinations.

Fig. 3.16 shows the quantitative differences of the computational load of the algorithm with healthy subjects, amputees and with Ninapro data, considering the complete and reduced datasets. To compare the complete and reduced approaches, we consider the mean execution times of the 3 classes of subjects on the proposed MCU. The SVM reaches the best performance with the healthy subjects, because the activation patterns of the different gestures are well separated due to the good condition of the forearms muscles. With the Ninapro healthy subjects the SVM reaches performance comparable to our setup, even though the execution is slightly slower.

The test on amputee patients demonstrates that the computation of the classifier's output requires a significant increase of time with the complete dataset, due to the larger SVM model. In fact, the compromised residual muscles of the amputees produce more confused activation patterns and the SVM algorithm needs more SVs to define the separation boundary between the classes. Nevertheless, with the proposed control strategy, it is possible to reduce

the number of SVs of the model by the elimination of the open hand gesture. In this case, we obtain a significant reduction of the number of SVs and consequently of the computation time that becomes comparable to the performance obtained on healthy subjects with a redundant sensor setup. The experimental evaluation shows that the proposed hardware and the software implementation are tailored to meet memory and real-time processing constraints of the application. Moreover, the MCU's rich set of peripherals and the possibility to synchronize the ADC and PWM make it the optimal choice for such system.

3.4 Conclusion

At the best of our knowledge, the described system represents the first example of a hand prosthesis controlled by a natural gesture interface. The analysis of a robust setup, the embedded implementation of a powerful machine learning algorithm and the design of an efficient control strategy allowed the implementation of the complete system. The performance of the system are measured on the final gesture recognition and to maximize the robustness, exploiting the robustness of the SVM and of an upper level FSM. If we analyze our results using the classical metrics of the percentage accuracy of the classification we would reach results of classification accuracy that are around 97% on real data. These values of accuracy are not reachable in embedded systems simply using the whole classification algorithm without very large models and risks of overfitting. The combined approach gives very good performance and maintains flexibility in the design of the control strategy enabling further techniques to gain robustness and improve the reliability of the system. We designed the system using only COTS because we believe that this setup can be easily reproduced and can be useful for ongoing research on a really deployable the upper limb prosthesis.

Chapter 4

Versatile EMG based Pattern Recognition Platform

4.1 Overview

Conditioning and processing of biological signals represent interesting challenges for wearable electronics in health applications. Information gathering from these signals requires complex hardware circuitry and dedicated computation resources. The design of innovative analog front-end integrated circuits, combined with efficient signal processing algorithms, allows the development of platforms for monitoring, activity and gesture recognition based on embedded real-time systems. This chapter describes the outcome of a collaboration between the University of Bologna and the ETHZ for the development of an efficient wearable biosignal processing platform.

Learning from the experience coming from the design of the prosthetic controller, we present the description and the performance of an EMG pattern recognition system based on the combination of low cost passive sensors, an innovative analog front-end and a low power microcontroller. The performance of the proposed system matches state-of-the-art high-end active sensors, opening the way to the development of affordable and accurate wearable devices. In the former part we analyze and compare the performance of this system with the active electrodes based setup used in previous chapter for the pattern recognition prosthetic system, while in the latter part we focus on the analysis of the embedded platform, in terms of sensors number and placement strategies and in terms of the trade-off between computation time and classification accuracy. The work focuses on the multi-level design and characterization of the system, integrating the hardware and software components to

develop a wearable device capable of acquiring and processing EMG signals for real-time gesture recognition.

Some attempts in the direction of an embedded platform for hand gesture recognition with on-board computing capabilities are performed in recent work [70, 117]. These works present an architecture based on an ARM Corex-A8 DSP processor and offer an open source platform on which it is possible to run pattern recognition algorithms. The achieved computation time (0.58 ms) matches the real time requirements, but the test dataset includes only one subject with just the recognition of the wrist flexions and extensions and the work does not cope with power consumption or with the issues of EMG acquisition and preprocessing. Furthermore, the used platform is *de-facto* a miniaturized PC and has a power consumption of 1 W limiting its use for low power wearable devices.

For the development of a wearable and low-power system, targeting high accuracy, the most promising approach seems to be the synergy between a low power AFE and a microcontroller, merging the system flexibility with a good signal quality and maintaining a good trade-off between power consumption and computing capabilities.

Some commercial AFE solutions are available on the market [102], [13]. However, the scientific community is also interested in building highly efficient AFEs. In [113], an 8-channel AFE ASIC for EEG acquisition is reported. The ASIC operates efficiently but the platform does not provide signal processing capabilities. A 16-channel ASIC for neural data acquisition is reported in [55]. It has a low power consumption of 2.21 mW, but the proposed AFE introduces a considerable IR-noise ($5.4 \mu V_{rms}$) and the ADC only achieves 7 ENOB, hence the system has limited performance. The solution published by [29] is further simplified where only the AFE is designed at chip level. The ADC responsible for biomedical data acquisition is placed off-chip and controlled by an external dedicated DSP. An implantable AFE solution is presented by [78] and a multichannel programmable IC is introduced in [71], but the authors only present and evaluate the AFE design, with no integration of ADC. An active-electrode solution supporting the use of dry electrodes is presented by [109] and a biopotential front end is presented in [101], again without the integration of an ADC and with only the evaluation of the AFE.

The only efficient solution integrating the AFE with ADC and a microprocessor on the same chip is published by [103] to build a multi-sensor biomedical ASIC. For this purpose, an ARM Cortex M0 processor is used with dedicated hardware accelerators optimized for energy efficient execution of biomedical signal processing algorithms. However, the limited number of sensors (3) affects the system flexibility and limits the possibilities to use this solution in other applications beyond ECG. Table 4.1 summarizes the comparison of the aforementioned systems with the proposed solution. Our goal is to develop a wearable

solution, which preserves the data acquisition quality of active sensors, while providing a configurable and scalable solution with processing capabilities. With this approach, we can dramatically reduce the cost of system scaling, for instance bringing from 600\$ to tens \$ the insertion of a new channel in the system, thus providing a versatile and scalable platform with the additional benefits of on-board processing capabilities.

Furthermore, we compare our device and recognition outcomes with the performance of a recently introduced commercial low-cost wearable system, the MYO armband from Thalmic Labs [6], that represents the most interesting commercial solution for wearable EMG gesture recognition. This is a wearable and low cost dev equipped with EMG and inertial sensors. It connects to a PC or tablet via Bluetooth Low Energy (BLE) and allows both raw data streaming and the use of a proprietary library for gesture recognition. The signal processing is performed on the host platform and the used algorithms are not documented, but they can recognize 5 simple hand gestures using the 8 EMG channels and a 3-axes accelerometer. This system represents an innovative platform and the first commercial system for EMG hand gesture recognition. Nevertheless, the device presents low flexibility in terms of possible applications because it lacks embedded computing capabilities and cannot be used as a stand-alone system. With respect to it, we perform the recognition directly on board targeting at the same time recognition accuracy and real-time performance. Our proposed platform allows the classification of 7 gestures with a classification accuracy of up to 92%, with a power consumption of $29.7mW$. The results presented here demonstrate the robustness of our approach, reaching the performance of state-of-the-art systems on a miniaturized wearable low-power and low-cost flexible platform. The test results demonstrate a similar performance for our fully embedded and wearable system with on-board computing and the commercial armband with the classification performed on a PC.

4.2 Active and Passive Electrodes

Advances in sensing and processing technologies have led to miniaturized wearable devices suitable for a wide range of applications, including Human-Computer interaction, fitness and health [83]. A small size is required for such systems to be unobtrusive and successfully adopted, posing strict constraints on their technical specifications. In particular, it limits the battery size/duration and the processing capabilities embeddable in the system, demanding for a high energy efficiency and an optimal use of the available resources.

Since EMG is a superposition of a low-amplitude signal (e.g. $V_{EMG} \leq 10mV$), it is affected by an offset of up to two or three orders of magnitude higher and noise. The offset is mainly caused by the tissue-to-electrode interface, which results in the superposition of a

Table 4.1 Comparison with state-of-the-art AFE ASICs.

	ADSI1298 TI[102]	ADASI000Yazicioglu AD[13]	[113]	Gosselin [55]	Bohorquez [29]	Muller [78]	Lopes [71]	Xu [109]	Teng [101]	Helleputte [103]	Proposed ASIC [94]
Applications:	ECG, EEG	ECG	ECG, EEG	EEG	ECG, EEG	EEG	EEG	EEG	EMG, ECG	ECG	EMG, ECG, EEG
Technology [µm]:	-	-	0.5	0.18	0.18	0.065	0.35	0.18	0.35	0.18	0.13
Supply [V]:	3.0	3.3	3.0	1.8	1.5	0.5	3.3	1.8	1.5	1.2	3.3
Power [mW]:	9.5	21	0.2	2.21	-	0.005	5.9 ^a	0.7	0.324 ^a	0.345	15
Electrode channels:	8	5	8	16	1	2	16	8	3	3	8
Int. offset compensation:	no	no	yes	yes	no	yes	no	yes	yes	yes	yes
Tolerated offset [mV]:	±425	±300 ^b	±45	±450	-	±50	-	±250	±350	±400	±300
IR-noise (1-100 Hz) [µV _{rms}]:	0.53	0.94	0.59	5.4	3.4	3.6	2.9	1.75	21	0.62 ^c	0.82
Dynamic range [dB]:	115	107	-	-	-	-	60	84	97	110	108

^a power consumption of ADC not included

^b highest gain setting

^c for a signal bandwidth of 1-150 Hz

non-constant half-cell potential to the EMG signal. Furthermore, the tissue-to-electrode impedance is frequency-dependent, time-varying and depending on the used electrodes: it ranges between 100Ω and $10M\Omega$. To correctly handle this low-amplitude signals, a high-gain, low-noise Analogue Front End (AFE) is required. The AFE needs to be robust to offset drift and to provide a high input impedance to prevent that contact impedance affects the measurement.

In advanced wearable health applications, such as prosthetics or robotic hand control, high accuracy and robustness of the system are essential combined to unobtrusiveness. In this cases, active EMG electrodes are used and the analog signal conditioning is performed by the sensor's internal circuitry as in the Ottobock 13E200 sensor used as a reference. Active sensors can be connected to a simple low-precision ADC, typically included in a low-cost microcontroller (MCU), which can process the conditioned EMG signal. The block diagram of this solution is illustrated in the top row of Fig. 4.1. This architecture does not require advanced signal processing algorithms for signal conditioning and acquisition, providing a robust platform for hand gesture recognition, but it is limited by the cost of the active electrodes (hundreds of Euros), which lead to expensive devices with a small number of electrodes [5].

Active EMG sensors are used in [27], while the system proposed in [98] is based on an array of passive sensors and a external PC to run the algorithm. The signal acquisition, conditioning and processing in those systems is performed with expensive commercial data-loggers or dedicated diagnostic hardware, supported by high-end interfaces, hence they are not suited for embedded applications and wearable low cost devices.

The open challenge we address in this part is the design and implementation of a embedded wearable system that can process EMG signals in real time to recognize hand gestures, targeting the scalability and the versatility of the design. We propose the implementation of a real time system for the acquisition, conditioning and classification of EMG signals, based on Cerebro, an innovative AFE developed in our lab [95], interfaced with a 32-bit ARM Cortex M4 MCU. The block diagram of the proposed solution is illustrated in the bottom row of Fig. 4.1. The system implements a pattern recognition algorithm based on Support Vector Machines (SVMs) to classify hand gestures. It complies with the real-time requirements for the control of hand prosthesis ($300ms$ of response time), while facing the computational and memory constraints of such low-resources embedded system. We will show the use of the proposed system for the recognition of hand gestures to control a prosthesis, with performance comparable to a high-end active sensor-based system [24]. Furthermore, the system is versatile, scalable and can be applied to a vast range of signals and applications.

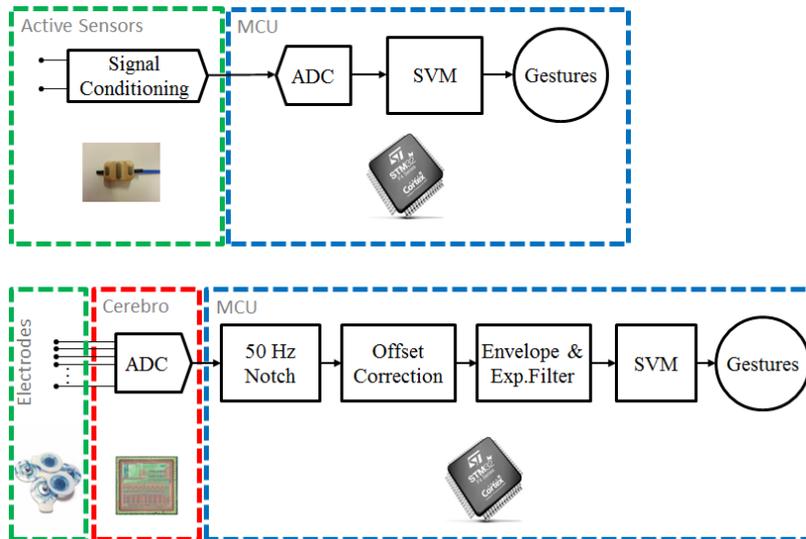


Fig. 4.1 Block diagram of the commonly used active sensors setup (top row) and the proposed solution (bottom row).

4.3 Architectural Overview

The proposed system combines hardware and software elements to form an efficient and flexible solution. Its block diagram is proposed in the bottom row of Fig. 4.1. We use low-cost disposable electrodes for the acquisition of EMG bio-potentials from the user's forearm. They are acquired via the Cerebro AFE and fed into a MCU, which implements the processing steps. Here, the acquired signals are digitally conditioned to reduce interferences and eliminate offset. They are then fed to a SVM classifier to recognize the performed hand gesture. The use of a programmable MCU allows a high flexibility, lower than an FPGA but more simple, extremely inexpensive and more usable for most developers. Furthermore, the combination of a single multichannel AFE with low cost sensors allows the scalability of the system, increasing the number of sensors without losing the low cost requirements.

4.3.1 Chip level Design

The task of an analog front-end (AFE) for biomedical-signals is to amplify the desired signal to an appropriate level for analog-to-digital conversion while suppressing all unwanted signals such as electrode DC-offset, mains interference and circuit noise. The Cerebro analog front-end depicted in Fig. 4.2 The AFEs of the 8 differential channels consist of an instrumentation amplifier and a 3.2 kHz low-pass filter, followed by a multiplexer to a shared 13 ENOB Sigma-Delta ADC, which samples each channel at 8 kHz. . Each channel consists of a variable-gain instrumentation amplifier (IA) followed by a first order active

RC low-pass filter. The IA is chopper-stabilized to remove the flicker noise from the signal band. Measurements show $0.82 \mu\text{V}_{\text{rms}}$ noise in a 100 Hz bandwidth [94]. Two current-mode digital-to-analog converters (DACs) compensate DC-offset at the IA input: the first DAC removes the input referred offset of the IA itself whereas the second, chopped DAC removes the differential input offset caused by the electrode-to-skin contact.

A single-loop 3rd order delta-sigma modulator with 3-level quantizer provides 80 dB of dynamic range for the accurate conversion of the eight input channels. Running at 16.384 MHz, the modulator is operated at an oversampling ratio of 64 with each channel sampled at 32 kS/s. A first comb filter decimation stage followed by a de-multiplexer and a second comb filter decimation and compensation stage provides 8 kS/s, 16 bit, high resolution sampled data at the AFE output.

Based on the decimated output samples of the ADC, the microcontroller controls the on-chip DC-tracker for each channel. When the DC-offset drifts above the linear voltage range at the input of the IA, the value sent back to the IA is incremented/decremented to compensate for the offset drift thereby closing the loop. The DAC DC-compensation values for each channel can be added to the corresponding received samples in the microcontroller to reconstruct the true DC signal acquired from the electrodes. This solution differs from the classical approach where a large external AC-coupling capacitor is used to create a system high-pass with a corner frequency below 1 Hz. With the implemented scheme, an effective dynamic range of 108 dB is achieved [94]. The Cerebro ASIC further contains a low-impedance patient ground (PGND) for setting the input common mode and a single-ended input, a 12-bit auxiliary ADC for internal temperature measurement or acquisition of an additional input. A 3-wire SPI interface is employed to connect to Cerebro. Table 4.2 compares the Cerebro with two commercial state of the art biomedical signal acquisition chips, revealing that it operates with similar performance as the commercial products. What differentiates Cerebro from other data acquisition ASICs is the internal offset compensation that can be digitally controlled. A detailed description of Cerebro is reported in [95].

The block diagram of Cerebro is depicted in Fig. 4.2. To mitigate flicker noise, the predominant type of noise in the signal band, the instrumentation amplifiers are chopper-stabilized. By the use of DACs and a DC tracking circuitry, the time-variant DC-offset is compensated and an effective dynamic range of 108 dB is achieved.

4.3.2 System level design

The proposed device is a multi-functional biopotential acquisition and processing system, whose high-level block diagram is shown in Fig. 4.3. The focus of our work is the system-

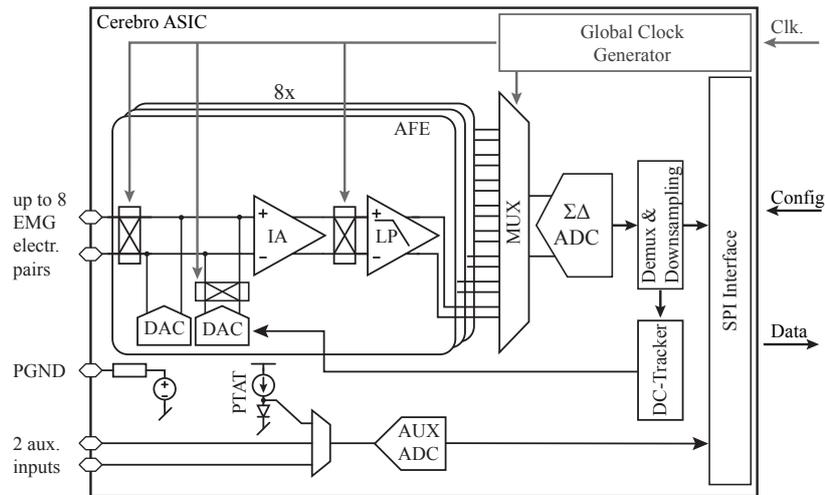


Fig. 4.2 Block diagram of the Cerebro ASIC [94].

Table 4.2 Comparison of the Cerebro AFE with two commercial solutions

	Cerebro	TI ADS1298	AD ADAS1000
Technology	130nm	-	-
Supply voltage [V]	1.2/3.3	1.8/3	1.8/3.3
Nr. Channels	8	8	5
Max. Offset [mV]	± 300	± 300	± 900
Int. Offset Compensation	Yes	No	No
IR Noise [nV/\sqrt{Hz}]	82	31	95

level design of the device, from the hardware development to the EMG-based gesture recognition application.

This device is composed of the multichannel Cerebro AFE [94], which is interfaced via SPI to an ARM microcontroller used for data acquisition and on-board gesture recognition SVM. This architecture combines the performance of a dedicated AFE with a general purpose microcontroller, leveraging system performance and flexibility for various biomedical applications. Additional inertial and pressure sensors have been added in order to enable in future developments sensor fusion techniques for an accurate gesture recognition and motion tracking. The acquired data and the resulting gestures can also be locally stored on a SD card or transmitted by a Bluetooth module to a host device. The raw data transmission is necessary for the training of the gesture recognition algorithm, resulting in the SVM models that are then uploaded and stored on the device for on-line use. We adopted a standard Bluetooth 2.1 module, since it allows to stream the acquired data at high rates and provides a reliable real-time acquisition system, which is not achievable using a Bluetooth Low Energy

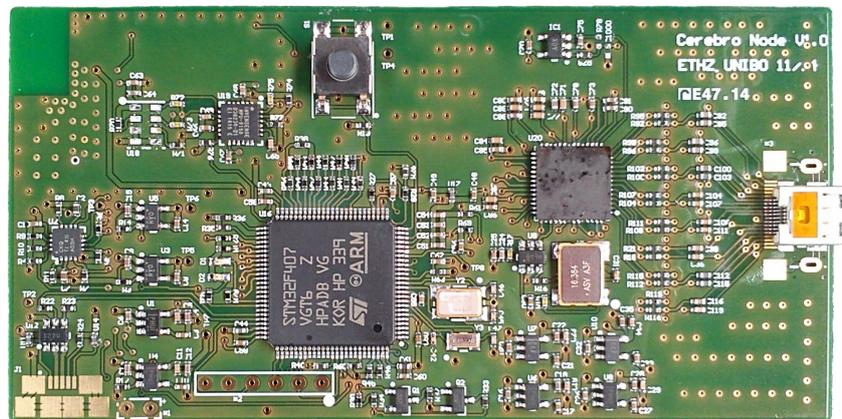


Fig. 4.4 Photo of the assembled PCB (top view). The dimension of the circuit is 85×50 mm.

in use (battery or USB) and manages the recharging of the battery. Starting from this system supply, the different sections of the board (Bluetooth, sensors, analog) are provided with separate low-dropout voltage regulators. This flexible solution for controlling the power management improves the integrity of the acquired analog signals and allows us to switch-off or duty-cycle submodules of the board that are not required for a targeted biomedical application and thus enhancing battery lifetime. The main features of the device are summarized in Table 4.7.

4.3.3 Signal Pre-processing

The electrical activity of the moto-neuron cells induces muscular contractions. Muscle tissue conducts the APs, in the same way of nerves, which is detected and measured by EMG sensors. Electrodes are placed on the skin surface and capture the action potentials of the muscular fibers underlying them. As mentioned, the classical EMG active sensor is made by two metal plates, each one is connected to the input of a precision instrumentation differential amplifier that senses the action potential of muscular fibers. The EMG signal is differential with zero mean and an amplitude varying from $\pm 25 \mu V$ to $\pm 10 mV$, depending on the dimension and the depth of the muscles contracting underneath the electrodes. The acquired signal is a combination of the EMG potentials, a time-varying offset and noise. Noise and interference sources, which are all in the useful signal band, include variations in the reference potential induced by the floating ground potential of the human body, electrostatic noise collected by the input stage of the amplifier and mains interference.

An example of 4 EMG channels acquired from the user's forearm with a battery-powered Cerebro acquisition board is shown in the top plot of Fig. 4.5.

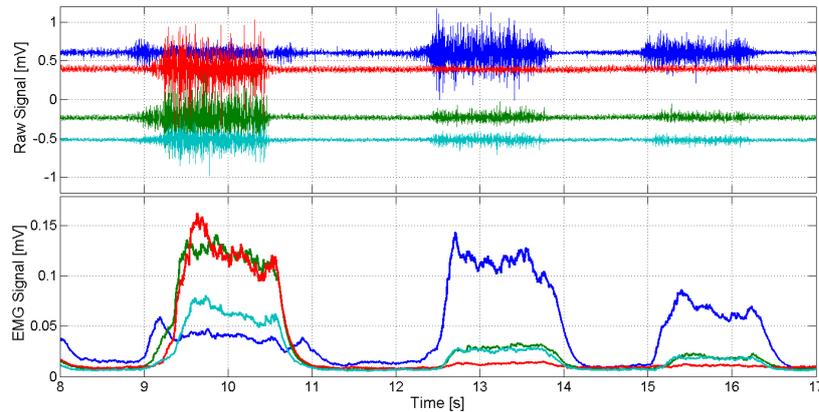


Fig. 4.5 Raw data acquired from the Cerebro ADC (top) and the processed signals (bottom).

Ottobock active sensors extract the EMG envelope via hardware, with analog signal conditioning circuitry based on an integrator and on a notch filter to reduce the mains interference. Our first goal is to extract the same information from the EMG signal with a digital signal preprocessing, without external HW components. To extract an enveloped signal for the SVM classifier we preprocessed the EMG samples acquired with the AFE with the following blocks:

Notch Filter. The first step of the signal processing chain is the elimination of the mains interference, which is a sinusoidal component in the range 50-60Hz, depending on the geographical location. It is caused by the AC frequency of the mains powerline and persists also when the system is powered by a battery. Solutions to eliminate the mains interference vary from notch filters to interference cancellation via time-domain or regression subtraction [18]. For EMG signals this is not a critical problem, given its slightly higher amplitudes when compared to other biosignals (e.g. EEG).

Considering the nature of the EMG signals and the limited hardware resources, we choose to implement a notch filter, which offers good performance at a very limited computational and memory cost. In particular, we implemented a second order notch filter with a Q-factor of 60. To deal with small variations of the interference frequency, we perform an initialization phase during which we compute the Fast Fourier Transform (FFT) of the acquired signal and find the peak frequency corresponding to the mains interference.

Offset Correction. The time-varying offset should be removed from the filtered signal. To correctly track the offset, its estimate must be periodically updated. This is done with a moving average approach: at each sample we compute the average of the last N_O samples and subtract the result from the current sample to obtain an offset-free signal. The mean is computed in an incremental way: we keep a buffer of the last N_O samples and compute

Table 4.3 Signal preprocessing resource requirements

	Clock Cycles	Time [μs]	Memory [Bytes]
Notch	110	0.655	30
Offset	35	0.209	140
Envelope + LPF	26	0.155	8

their sum, removing at each iteration the last item in the buffer and adding the new one. We evaluated the effect of the size of the sliding window on the estimation of the offset and we found that a window of $N_O = 128$ samples has the best trade-off between accuracy and memory requirements.

Envelope and filtering. Once the offset has been removed, we extract the envelope of the sampled differential signal, by replacing each sample with its absolute value. To eliminate high frequency noise, the signal is further low-pass filtered. This is achieved by an exponential filter, which has the form:

$$y_n = \alpha y_{n-1} + (1 - \alpha)x_n \quad (4.1)$$

where x_n is the current sample and y_n and y_{n-1} are the current and the previous filtered samples. The coefficient α was set to 0.99 and an example of the resulting signal is plotted in the bottom graph of Fig. 4.5. The FFT of such signal is plotted in Fig. 4.6(a), and it can be compared with the FFT of a signal acquired by a high-end commercial EMG electrode (Ottobock 13E200), shown in Fig. 4.6(b). The summary of the computational and memory costs of the proposed preprocessing algorithm is shown in Table 4.3 for the ARM Cortex M4 MCU employed in this study. The proposed solution can easily be employed to sample all the 8 available channels at their maximum sample rate of 8KHz, leaving processing time and memory for the gesture recognition stage.

After the comparison of our signal quality with the one acquired with state-of-the-art active sensors [7], we complete the evaluation comparing our signal with the one provided by a recent wearable EMG system. In Fig. 4.6 we compare the acquired EMG signal and its frequency spectrum for the three acquisition systems, (a) Cerebro AFE, (b) Ottobock active sensors and (c) MYO device. The three signals were subsequently acquired from the same user, replicating the same acquisition setup for the three systems. The active sensors integrate a hardware amplifier, a low pass filter with a cut-off frequency of 450 Hz and extract the envelope of the signal. They provide an amplified signal in the range 0-3 V, which is directly acquired by the internal ADC of the microcontroller, with a sampling frequency of 500 Hz. The MYO provides a programming interface to receive EMG signals from its 8

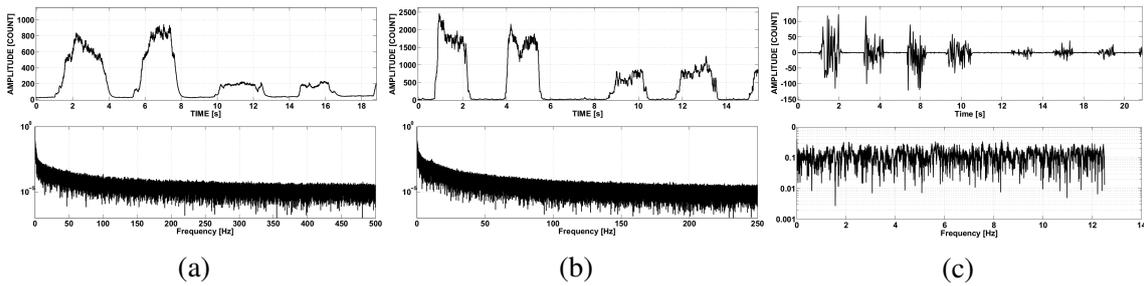


Fig. 4.6 EMG signals (top) and their spectrum (bottom) for the tested system: (a) Cerebro, (b) Ottobock, (c) MYO. The signals represent gesture contractions and were acquired in sequence from the same user, placing the electrodes in the same position

sensors at approximately 25 Hz. We presume that internally the signals are amplified, filtered and sampled at a higher frequency, but no information is available yet.

Fig. 4.6 clarifies that the FFT extracted from the signals acquired with the Ottobock sensors and with our platform are equivalent, while the limited bandwidth provided by the MYO does not allow extraction of information in the frequency domain.

4.4 Experimental Results

In the proposed experiment, one healthy subject was sensorized with 4 EMG passive sensors placed on the forearm, following the placement strategy described in [24]. The sampling frequency of each channel was 1 KHz and the selected gain was 512. The EMG signals were collected for the most important hand movement of daily life activities: open hand (OPH), power grip (POG), 2-fingers precision grasp (PCG), pointed index (PIX) and rest position (RST). Each contraction was held on for 3 seconds and followed by 3 seconds of rest position. Each gesture was repeated 10 times. Data are acquired from the board and sent to a PC where the segmentation and the labeling operation were performed in Matlab. The SVM model was trained using 30% of the collected dataset, while the remaining data was used to test its accuracy. The purpose of the performed experiments was the verification of the accuracy of the proposed setup comparing it with an equivalent high-end commercial front end. Fig. 4.7 shows an example of acquired EMG signal and the relative SVM classifier output. The accuracy was measured as the difference between the SVM output value and the true label of the gesture annotated by hand. The classification accuracy on the acquired dataset is 92.36%. From the output of the classifier we can notice that the misclassifications are focused on the transient phase of gestures, where the muscular activation pattern is affected by a high variability.

Fig. 4.7 (b) shows the 4 gestures acquired with the Ottobock active sensors. The data was

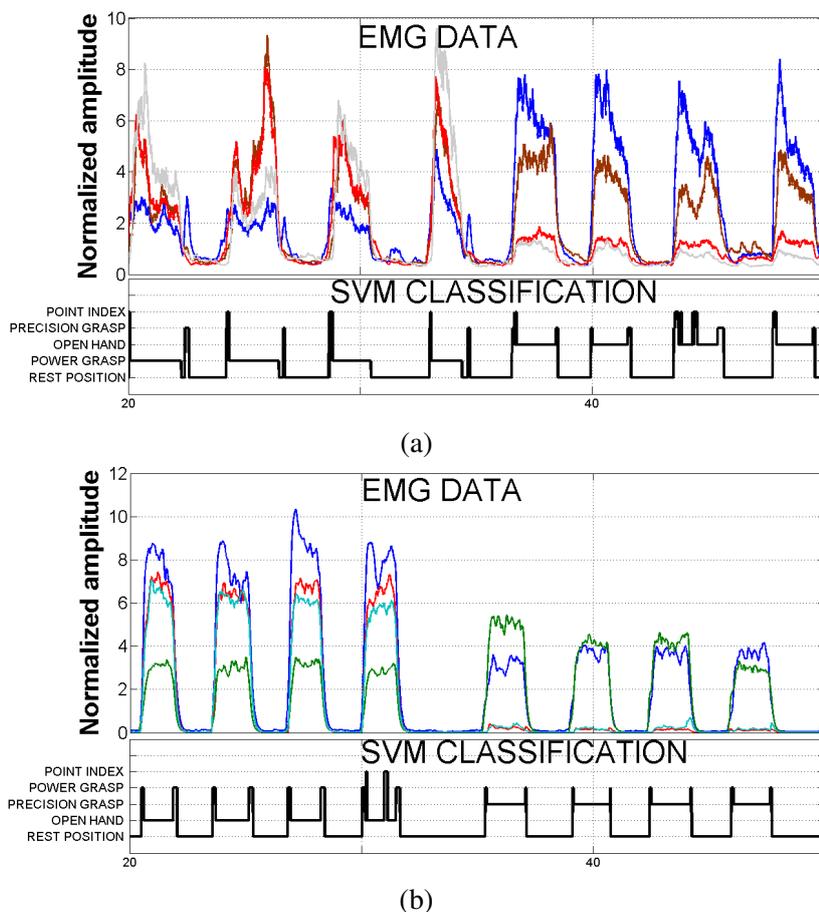


Fig. 4.7 Accuracy of the classifier signal of data acquired with dedicated AFE (a) and commercial active sensors (b)

acquired with a microcontroller system connected to 4 EMG active sensors and segmented with the same procedures used for the previous tests. Tables 4.4 and 4.5 show the confusion matrices of the two classification sessions.

The two prediction outputs are similar and present comparable misclassification on the transient of the muscular contraction. The overall accuracy obtained with the active sensing solution is 91.06%. The signal acquired with the proposed system is slightly more noisy than the one acquired with the active sensors, but it is suitable for an accurate gesture recognition via SVM, achieving equal classification rates. Given the higher noise of the proposed AFE-based solution, the SVM training creates 113 support vectors (SV) for the AFE solution and 59 SV for the active sensor solution. The number of SV influences the memory requirements and the computational time needed for the on-line classification of gestures. The computation time, measured on the embedded implementation of the SVM as in [24], is 1.5ms for the

Table 4.4 Confusion matrix for recognition with passive sensors.

	RST	POG	OPH	PCG	PIX
RST	77197	0	0	73	0
POG	1342	16823	0	718	729
OPH	227	0	13872	2816	886
PCG	667	1	390	13902	74
PIX	1102	1377	0	1648	17016

Table 4.5 Confusion matrix for recognition with active sensors

	RST	OPH	PCG	POG	PIX
RST	78817	0	868	3368	0
OPH	1342	10570	0	2449	2327
PCG	0	0	15775	66	0
POG	0	2	59	17653	0
PIX	0	905	0	3043	11565

active sensor solution and 2.8ms for the AFE, which is adequate for real-time processing of a gesture recognition application. This demonstrates that the proposed solution with the Cerebro AFE and the MCU is suitable for an accurate embedded gesture recognition system.

4.4.1 DWT classification

In the previous section we compared the results of the EMG recognition with active and passive electrodes, concluding that we are aligned with the Ottobock sensors interface. To give further validation of our feature extraction method for EMG signal, we compared the accuracy of the proposed method for training and classification with the SVM with a very popular feature extraction technique for EMG raw signal: the Discrete Wavelet Transform (DWT) feature extraction. DWT is the most accurate feature in the time-frequency domain and gives good performance with the pattern recognition of the EMG signal [87]. The Wavelet Transform is based on the Multiresolution Analysis (MRA) of a given signal through a scalable modulated sliding window.

Wavelet Transform overview

The Wavelet Transform provides a Time-Frequency representation of a non-stationary signal, giving information on the frequency content of a signal in the time domain. Formally, the

Continuous Wavelet Analysis (CWT) of a given function $f(t)$ for the scale factor s and the translation τ is given by:

$$\gamma(s, \tau) = \int f(t) \Psi_{s, \tau}(t) dt$$

The wavelets are the set of basis functions $\Psi_{s, \tau}(t)$ and are obtained by scaling and translating a function $\Psi(t)$, called *mother wavelet*

$$\Psi_{s, \tau}(t) = \frac{1}{\sqrt{s}} \Psi\left(\frac{t - \tau}{s}\right)$$

As the value of s is increased, the wavelet is dilated ranging the analysis from the higher to the lower frequencies. The final result is the value of the transformation. For each value of s the wavelet is shifted increasing the value of τ . The implementation of this analysis requires to use a Wavelet that can be translated and scaled not continuously but in discrete steps, as showed below:

$$\Psi_{j, k}(t) = \frac{1}{\sqrt{s_0^j}} \Psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right)$$

with j and k integer values. The results of the transformation performed with discrete wavelets is a series of coefficients, named wavelet decompositions. Although this discretized Wavelet transform can be computed by a computer, it is highly redundant and computational demanding because of the all combinations of s and τ . Basically the CWT is a correlation between a given signal and a test function (the wavelet) at different scales. In the discrete case (the DWT), the signal is passed through a cascade of both low pass filter (LPF) and high pass filter (HPF) to obtain information on high and low frequency bands, remembering that filtering a signal is equal to perform the convolution of the signal with the impulse response of the filter. To calculate both the components we introduce the scaling function ϕ

$$\phi(st) = \begin{cases} 1 & \text{if } 0 < st < 1, \\ 0 & \text{otherwise} \end{cases}$$

The wavelet functions $\Psi()$ are associated to the high pass components while the scaling functions $\phi()$ are related to the low pass filtering. The filtering changes the signal resolution and the scale is changed by downsampling the original signal.

The Wavelet can be considered as a band-pass filter and the series of scaled wavelet can be considered as a filter bank. Using a finite number of values for the decomposition we

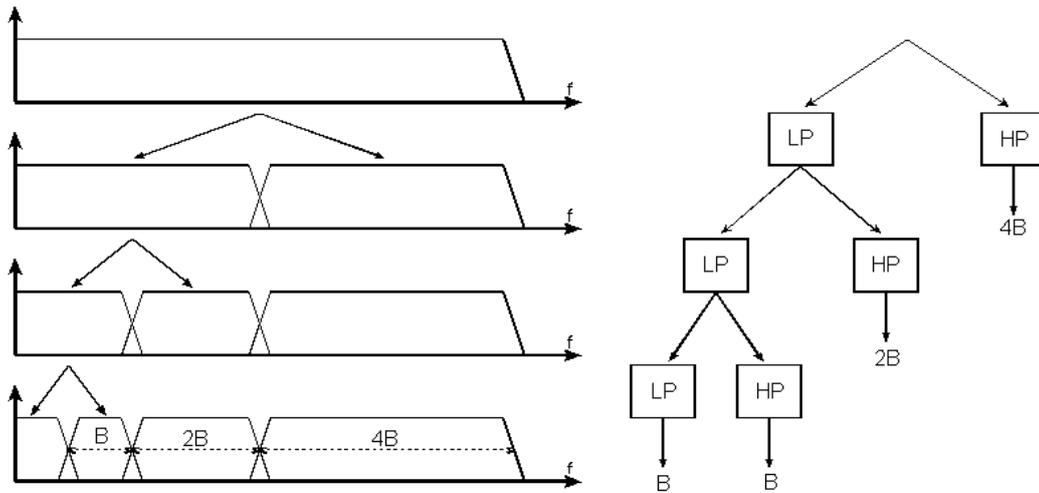


Fig. 4.8 Wavelet decomposition scheme

extract information in time-frequency domain by the recursive filtering of the given signal. Fig. 4.8 shows the sequences of high and low pass filtering sequence performed to obtain the DWT coefficients. For a DWT decomposition of level n , the coefficients of the LPFs are named *Detail Coefficients* (D_n) while the coefficients of the HPFs are named *Approximation Coefficients* (A_n).

Performance Comparison

The data are represented as a $(j \times 4)$ matrix, where j is the number of samples and 4 is the number of EMG channels. The D_n of the DWT are calculated on a fixed time window of the EMG signal (typically 256 or 512 samples). During the extraction of the DWT feature, the detail level used for this signal is set to 4 for each EMG channel and the resulting feature matrix $((j \times 4) \rightarrow (j \times 16))$ is composed by 16 columns. The energy of the coefficients is calculated and normalized on the entire window (i.e. the columns of the input matrix). The formula to calculate the energy of a vector is

$$E = \sum_{n=0}^N x_n^2 \quad (4.2)$$

In this experiment we compared the classification accuracy obtained with the DWT and with the envelope extraction to ensure that we have the same performance using the same training/classification datasets (without crossvalidations or other classifier optimization techniques). The idea of these test is to confirm that the two techniques act on the same data with the same accuracy using the SVM classifier. The accuracy of this classification is 81.3%. Figure 4.9 shows the classification label as output of the SVM predict function.

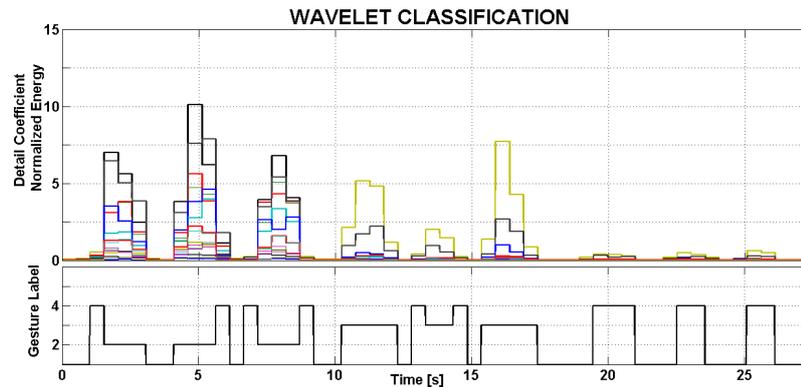


Fig. 4.9 Prediction labels for DWT featured signal.

Classification accuracy of the two methods is comparable, and this is a further demonstration that the envelope calculation can be used in this application. The SVM is executed on the microcontroller and with the Gaussian Kernel the model for enveloped signal with 58 SV is executed in $1.4ms$. The SVM for the DWT model has 14 SV instead of the 58 of the enveloped model, but the feature matrix is 16 columns because the detail level of coefficient produces 4 column per each vector. This means that the computation time of the SVM predict function is the same in both cases. Furthermore the classification with the DWT requires the time for the acquisition of the signal windows ($128ms$) and the calculation of the DWT ($3ms$). Table 1 summarizes the comparison between the two methods in terms of accuracy and computation time showing the number of SV of each model and the time of execution of the entire classification. In the column of the execution time the 3 terms of the sum represent the time of acquisition of the sample or of the windows, the time spent to compute the envelope or the DWT feature and the time to execute the SVM respectively.

METHOD	ACCURACY	SV	EXECUTION TIME [ms]
ENVELOPE	83.9%	58	$1 + 0.1 + 1.4$
DWT	81.3%	14	$128 + 3 + 1.4$

Table 4.6 Comparison between DWT and Envelope Detection.

4.5 Complete System Validation

After the validation of the AFE and of the signal acquisition chain, we present the complete embedded solution for real-time EMG-based hand gesture recognition.

The system combines the accuracy of a custom analog front end with the flexibility of a low power and high performance microcontroller for on-board processing. Our system achieves the same accuracy of high-end and more expensive active EMG sensors used in applications with strict requirements on signal quality. At the same time, due to its flexible configuration, it can be compared to the few wearable platforms designed for EMG gesture recognition available on market. We demonstrate that we reach similar or better performance while embedding the gesture recognition on board, with the benefit of cost reduction. To validate this approach, we collected a dataset of 7 gestures from 4 users, which were used to evaluate the impact of the number of EMG channels, the number of recognized gestures and the data rate on the recognition accuracy and on the computational demand of the classifier. As a result, we implemented a SVM recognition algorithm capable of real-time performance on the proposed wearable platform, achieving a classification rate of 90%, which is aligned with the state-of-the-art off-line results and a 29.7mW power consumption, guaranteeing 44 hours of continuous operation with a 400mAh battery. The pre-amplified signals coming out from active ottobock sensors have a good signal-to-noise ratio and noise cancellation characteristics, but such devices are expensive enclosed in a separate housing with a big form factor. Thus, they do not allow the design of a low cost scalable and miniaturized system, because the addition of input channels multiplies the needed analog circuitry and impacts on the final cost and dimension of the system.

Although a number of solutions has been proposed, also at chip level [113, 55, 29, 78, 71, 109, 101, 103], to promote scalability and integration in a EMG gesture recognition systems, currently there are no works presenting a complete system-level approach for signal acquisition and processing. In particular, the signal processing capabilities of all systems found in literature are limited and target low-power sensor nodes, where most of the necessary digital signal processing is performed on an external device in software, e.g. PC or smart phone.

Targeting a multilevel design approach. development of a high-performance wearable platform for hand gesture recognition needs we present in this section the results of the mplementation and validation of our versatile platform. The architecture, first introduced in [20], is based on the Cerebro AFE [94] interfaced with an ARM Cortex M4 microcontroller and merges the optimized design of an analog ASIC with the easy-to-use, low-power and high-performance commercial microcontroller. The achitecture of the AFE allows the system to be used for the acquisition of a wide range of biomedical signals, while the versatility of

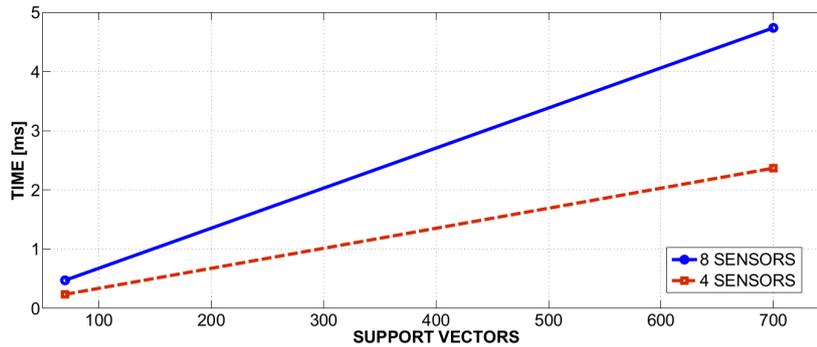


Fig. 4.10 Computation time for the online classification with SVM models of varying number of support vectors, as measured on the Cortex M4 MCU.

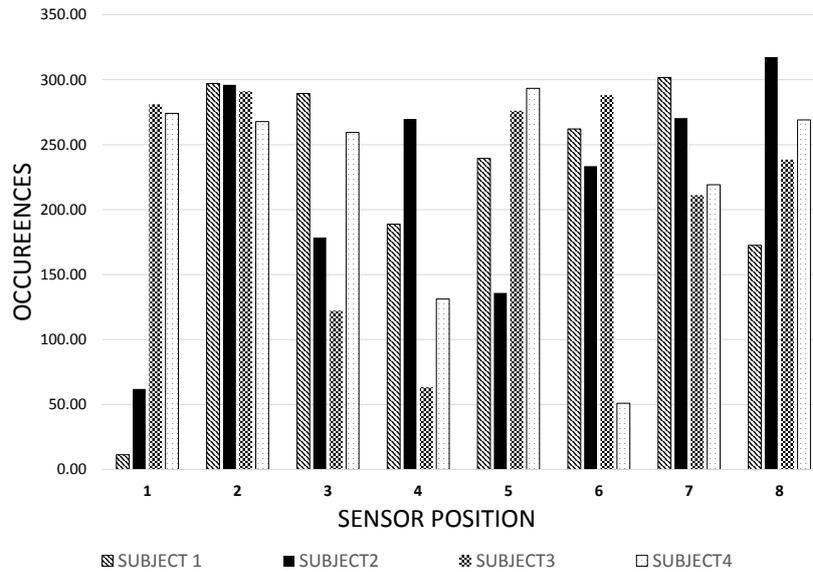
the microcontroller allows advanced on-board signal processing optimized for the application in use with a low-cost low-power platform. This synergy allows the design of an highly-configurable and scalable system able to provide a high EMG signal acquisition quality, comparable to the state-of-the-art active sensors and an on-board real time signal processing for pattern recognition.

4.6 Experimental Results

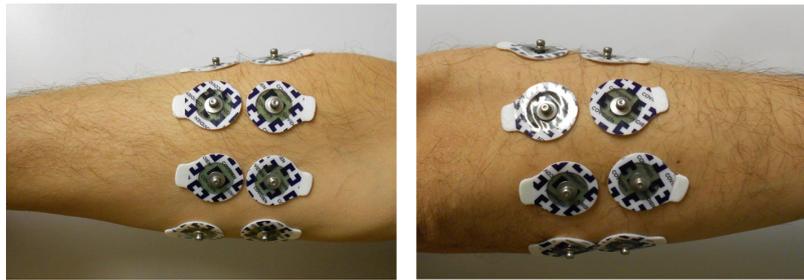
Even in this application, we used a Gaussian RBF kernel to cope with the variability of the EMG input signals. For the classification stage, we implemented an embedded version of the SVM classifier, with the adapted libSVM to avoid the dynamic memory allocation that is not suitable for the limited resources of an embedded microcontroller.

The memory and computational footprints of the recognition algorithm are directly influenced by the number of support vectors in the trained model, which becomes an important evaluation factor for a resource-limited platform as ours. In particular, the time needed to classify a new instance is linearly proportional to the number of SVs in the model, as shown in Fig. 4.10, where the runtime computational time was measured on the MCU of the proposed system for 40 and 700 vectors respectively. Since SVs are stored as floating point vectors, if d is the dimension of the SVs and n is their number, the cost in memory of the classification algorithm is $4 \times d \times n$ Bytes. In our application, the maximum number of SVs obtained from the algorithm training is less than 300, which corresponds to 9600 Bytes and can be easily stored in the MCU's 1 Mb of FLASH memory.

The evaluation of the trade-off between the complexity of the system and its performance is a key element in the design of a wearable device. Firstly, we verified the setup of our platform to be confident that it is aligned with the SoA in terms of recognition accuracy.



(a) Sensor occurrences in the configuration with the best accuracy for the subjects involved in the experiment.



(b) The placement of EMG electrodes on the subjects' forearms.

Fig. 4.11 Sensor placement and position occurrence in best accuracy classification

Hence, we made a detailed analysis of the system scalability in terms of the number of sensors used and recognized gestures, with relation to the computational and power costs. Finally, we acquired the EMG data from the MYO and we tested on them the accuracy of the same SVM algorithm implemented in our platform. All the data was acquired and tested on a PC using Matlab and an implementation of the SVM algorithm equivalent to the one we implemented for the embedded processor.

The EMG gesture data was acquired from four healthy subjects using the 8 fully differential channels of the Cerebro AFE at 1KHz. The EMG electrodes were placed on the subjects' forearm as shown in Fig. 4.11b. The set of 6 performed gestures was: power grip, precision grasp, open hand, pointed index and the flexion/extension of the wrist. The selected movements are the most used in the daily life activities and represent the more common movements that is possible to set on the control of a poliarticulated prosthesis. We

choose to evaluate also the flexion/extension of the wrist because is very used in literature and it is the main movement used in the control of commercial polyarticulated prostheses. During the acquisition sessions, the subjects repeated each gesture 4 times, they maintained the gesture contractions for 3 seconds and separated each gesture with 3 seconds of hand relaxation (rest position). The final dataset is composed by 7 classes: the 6 gestures and the rest position. With the collected gestures, we trained an SVM model for each subject using 10% of the dataset, with the vector of acquired and pre-processed EMG channels as input. The recognition accuracy was computed applying the resulting SVMs to the continuous stream of acquired gestures, which was opportunely labelled. The average accuracy in the recognition of 7 gestures is 89.2% (with a maximum of 92%), with a model composed of 123 SVs. This value is comparable with the systems presented in literature and obtained with the high-end interface described in previous sections.

To better evaluate the interaction between the system complexity and the recognition capability, we tested the accuracy in the recognition of subsets of a varying number of gestures (from 4 to 7), using varying numbers of input channels (from 4 to 8). Starting from the acquired dataset, for each gesture and input channel dimension, we tested all the possible combinations of sensor configurations and of recognized gestures. The gesture recognition accuracy and the number of SVs in the trained models were used as performance metrics, to evaluate the precision and the computational costs for each case.

This analysis can provide a robust estimation of the system performance, eliminating the dependencies from the selection of a gesture set or from a subject-tuned sensor configuration that can modify the final accuracy of the whole system. The results are reported in Fig. 4.12, showing the averaged recognition accuracy (a) and number of SVs (b) for the different configurations. We can see that the accuracy increases slightly when number of sensors increases and gesture number decrease. The total difference in accuracy from 4 to 8 sensors is less than 2%. This trend is confirmed also by [59]. The SV number increases when the number of sensors decreases and this is mostly due to the difficulties of the SVM algorithm in the creation of the decision boundary caused by the reduction of the information contained in vectors.

Furthermore, for each permutation of the classified gestures, the configuration of the sensors that achieved the best accuracy was traced to verify if it is possible to find a pattern for the best sensor placement common for all subjects. The sensors are numbered from 1 to 8 starting from the sensor placed near to the elbow. In Fig. 4.11a the columns on the x-axis represents the number of occurrences of each sensor in the configurations that reach the best accuracy, for each subject involved in the experiment. For instance, if the subject 1 reach the highest accuracy of the 4 gesture dataset with the configuration of sensor 1-4-7,

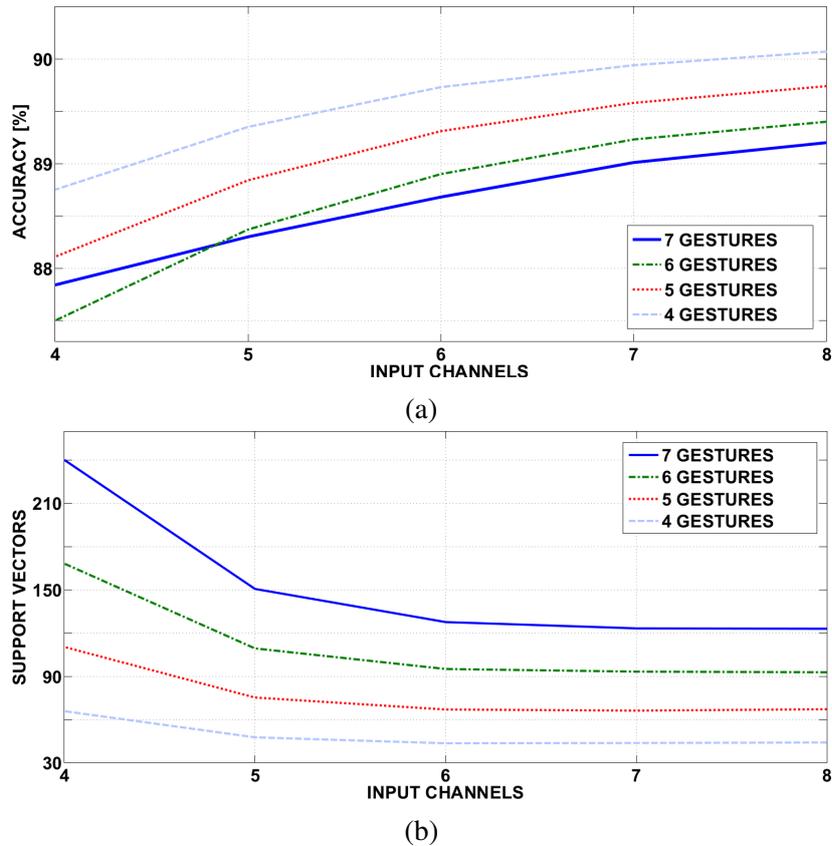


Fig. 4.12 Gesture recognition accuracy (a) and number of SVs (b) for varying numbers of input channels and recognized gestures. The reported curves are the average of the 4 users

these numbers are counted on the graph. This comparison allows to view if a certain sensor is mostly involved in the best classification rate for all subjects. A common pattern is not recognizable because the surface EMG electrodes collect crosstalk between near muscular fibers and the variability in the muscular structure confirms that a person-tuned setup is required .

Using the collected data, we also characterized our system in terms of the signal frequency used for the SVM classification. The Cerebro AFE has a minimum sampling frequency of 1KHz, hence the data is first acquired and pre-processed with that sampling rate. Then, we decimated the acquired samples, varying the sample rate from 25 to 1KHz, to evaluate the impact of the data rate on the SVM classification. The result is presented in Fig. 4.13a, where it is possible to notice that the average accuracy is not significantly affected by the data rate, while the number of SVs on the other hand is halved at 25Hz, reducing considerably the computational time and power consumption. A lower sampling frequency cannot be used since the data pre-processing algorithm cannot be adapted for such low frequencies.

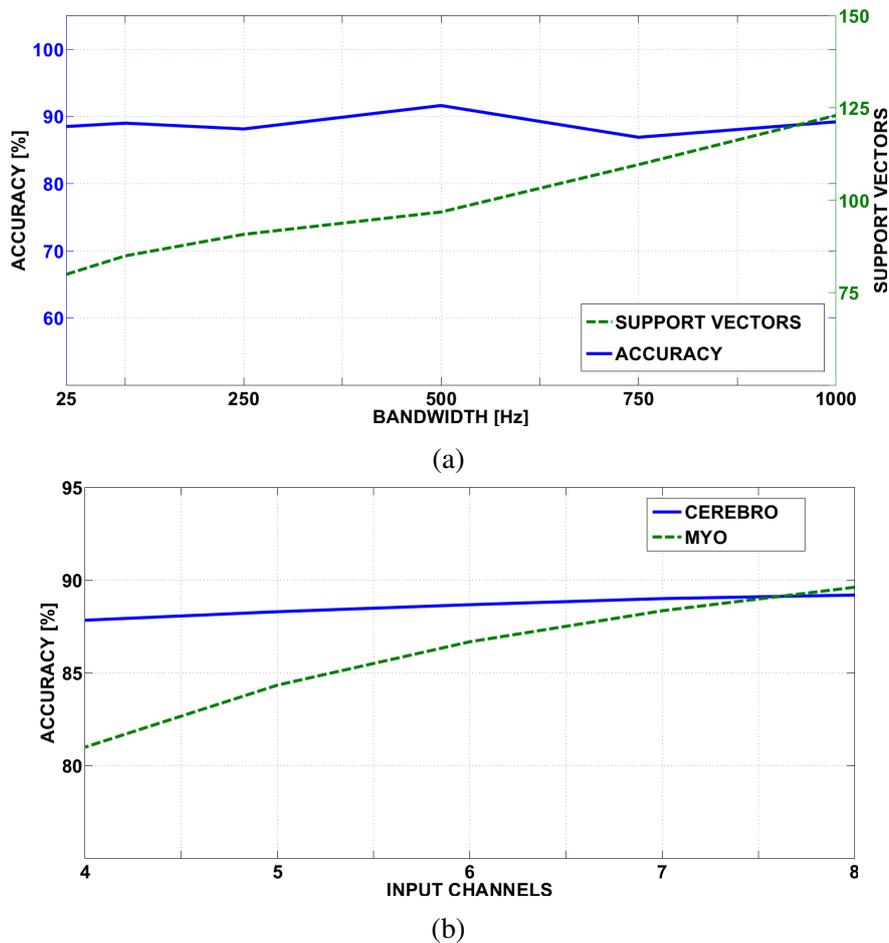


Fig. 4.13 (a) Recognition accuracy and number of SVs for different SVM input sample rates. (b) Recognition accuracy of the SVM algorithm for different number of input channels, using EMG signals acquired with the MYO armband (dashed line) and our device (continuous line).

Indeed, the envelope extraction described in the previous sections is based on an exponential filtering and does not work if the number of samples is too low. An approach where the down-sampling is performed after the pre-processing did not bring significant improvement in power consumption since in this case most of the energy is spent to read data from the ADC. To compare our system with the MYO armband, we used it to collect a dataset of gestures having the same characteristics described above. We acquired the 7 gestures dataset exploiting the data streaming function provided by the MYO armband. The only difference is that the output data rate of the 8 MYO channels is fixed at 25Hz. There is no information on how this signal is acquired, but we suppose that the EMG signals are internally amplified and filtered by an analog circuit. The 50Hz filter is implemented in HW on the analog circuitry

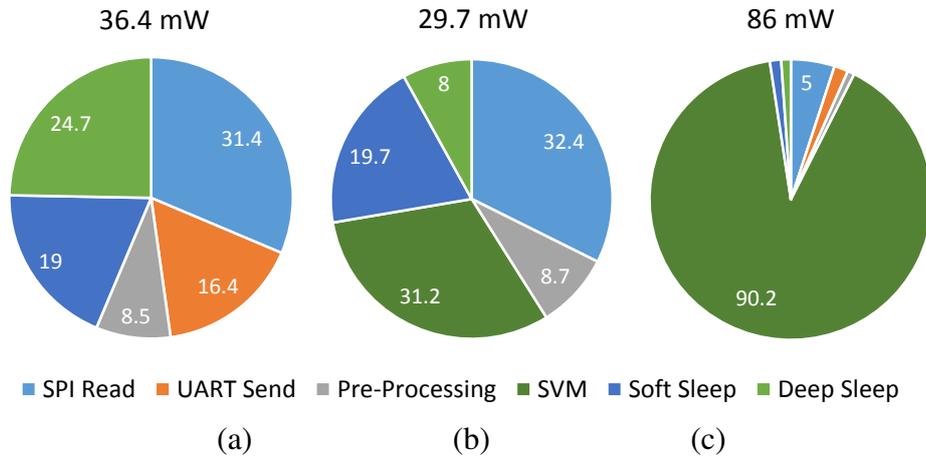


Fig. 4.14 Breakdown of the power consumption of the microcontroller for three different applications: (a) 8 channels data stream at 1KHz, (b) 7 gestures recognition at 25Hz and (c) 7 gestures recognition at 1KHz.

of the MYO to clean up the signal from the mains interference. To match this configuration and compare results with MYO, we sampled and pre-processed the EMG signals from our Cerebro platform at 1KHz and then downsampled the signal to 25Hz. Both datasets were compared using the proposed SVM recognition algorithm, with the EMG signals as input, the first 10% of the gestures for training and the rest for testing. Figure 4.13b shows the classification performance in the two cases for different numbers of signals used as input. As we can notice, with 8 sensors the accuracies are comparable, but the MYO system is less robust to the decreasing of sensors showing a drop of performance of 9.5% passing from 8 to 4 sensors compared to a drop of 1.5% for our solution. Moreover, our system is capable to perform the recognition algorithm on-board, while the MYO needs an external device on which to rely for the actual data processing.

An important feature of our system is its flexibility in terms of computational capabilities and power consumption. According to the application needs, it is indeed possible to configure different parameters, such as: the number of differential channels used, the number of detectable gestures, and whether to process data on-board or stream it through the Bluetooth interface. Bluetooth streaming allows to collect the EMG signals necessary for the SVM model creation during the training phase, which is performed on the PC. Once the model has been generated, it is transmitted back to the device and stored in the MCU's internal FLASH memory, turning off the radio to save power. To analyze the power consumption of the proposed device, we partitioned the system into its three main components: the data acquisition interface, the microcontroller and the radio module. The power consumption of

the Cerebro AFE is constant and it is equal to 15mW. The average power consumption of the Bluetooth is proportional to the amount of data sent and can reach a maximum of 16 mW, when streaming 8 channels at 1KHz. The power consumption of the microcontroller can be extremely variable according to the computational load, we thus analyzed the energy consumption during the 3 main phases of the program execution: data acquisition, computation and sleep.

During data acquisition from the AFE, the SPI interface, the interrupt controller and the DMA are active and used to transfer sampled data to the internal memory. The energy consumption is proportional to the number of used channels and the sampling rate, which for our application was 5.2mW reading 8 channels at 1KHz.

After acquisition, data gathered from the AFE is pre-processed with the digital filtering approach described in the previous section and this operation requires up to 0.7mW. The classification algorithm requires much more power than the pre-processing step, in this case the power needed to classify data is affected mainly by the number of SVs and the number of used channels. In Fig. 4.14 we show the breakdown of the average power consumption of the microcontroller for three application scenarios: data streaming via Bluetooth, SVM classification at 1KHz and at 25Hz. When the desired computation is over, the microcontroller is programmed to go in a sleep state, which was chosen considering its power consumption and break-even time. The implemented deep sleep state has transition times of $13\mu s$ and consumes only 1.1mW.

In total, the power consumption of the device is influenced by the application and the system configuration. With 8 active channels, we found a 36.4 mW power consumption for the data streaming application at 1KHz, 86 mW for SVM recognition of 7 gestures at 1KHz and 29.7 mW for the recognition at 25Hz. To lower the energy consumption it is thus possible either to reduce the sampling frequency or to reduce the number of detectable gestures.

4.7 Conclusion

In this chapter, we presented a wearable device for EMG signal acquisition and processing, with an on-board real-time SVM algorithm for the classification of hand gestures. The system integrates Cerebro, an innovative AFE for the acquisition of biopotential signals, and an ARM Cortex microcontroller for efficient digital processing. Our work, adopting a multilevel design approach, ranges from the chip design of the AFE to device system-level design and integration. The optimized design of the analog ASIC provides a high quality in the acquisition of a wide range of biomedical signals, while the versatility of the microcontroller allows advanced on-board signal processing optimized for the application in use.

To validate the proposed system, we compared it with two different classes of EMG devices: high-end active sensors (Ottobock 13E200) and a low-cost wearable device (MYO). The first ones provide a high quality EMG signal and are used for medical and prosthetics applications, while the latter one is a recently introduced armband for EMG-based gesture recognition for interactive applications. Our device, combining the Cerebro AFE and digital signal processing, is capable to provide signals with the high quality comparable with the active sensors, while being low-cost, wearable and flexible.

We used the EMG signal for the classification of hand gestures and implemented on the microcontroller a real-time SVM recognition algorithm. With this approach we obtained a good classification accuracy of up to 92%, with limited computational costs of the algorithm.

The presented recognition approach allows scalability of the number of used channels and the high flexibility of the presented platform allows a versatile optimization for energy efficiency. While maintaining the recognition accuracy and the number of recognized gestures, we are able to scale the number of used channels and the algorithm frequency, thus saving energy and achieving a power consumption of 29.7 mW, fully compliant with the energy constraints of a wearable device. This platform reaches the same performance of the benchmark wearable system for natural gesture recognition, while adding onboard real time computing and reducing the cost of the whole system.

The next important challenge that we intend to face is the use of the device for clinical tests, also with other biopotentials, taking advantage of its flexibility in terms of wearability and signal processing. This must cope with the issues related to the deployment of a wearable node, mostly related to environmental noise sources and floating ground. Indeed, in the next chapter we will present a detailed analysis of these two problems, proposing the solutions implemented on our wearable platform.

Table 4.7 System Specs

AFE:	
Acquisition channels:	8 EMG channels
Acquisition bandwidth:	1 kHz
Input-referred noise:	0.82 μ V
Input range:	\pm 300 mV
Power consumption:	15 mW
Microcontroller:	
Type:	ARM Cortex M4
Operating frequency:	168 MHz (210 DMIPS)
Flash:	1Mb
RAM:	192Kb
Maximum power cons.:	86 mW
Sleep mode power:	1.1 mW
Embedded platform:	
Dimensions:	85x50x6 mm
Total power cons.	29.7 mW

Chapter 5

Advanced Signal Conditioning

5.1 Overview

Conditioning and processing of biological signals represent interesting challenges for wearable electronics in health applications. Information gathering from these signals requires complex hardware circuitry and dedicated computation resources. The design of innovative analog front-end integrated circuits, combined with efficient signal processing algorithms, allows the development of platforms for monitoring, activity and gesture recognition based on embedded real-time systems which have great impact in many healthcare and rehabilitation scenarios. For instance, every year in EU and US over 4 million deaths are caused by cardiac diseases. More than 100 millions of people conduct their life after a heart attack and there are 30 million of people suffering from arrhythmia and other cardiac and cardiovascular disorders [106]. On the other hand, there are more than 1 Billion people suffering from neurological diseases, even if the international community was seriously underestimating in traditional epidemiological and health statistical methods its effects. The health cost for treatments associated with these diseases is about 450.5 B\$ per year [107]. Hence the monitoring of hearth rate, blood pressure, oxygen saturation, brain activity and other physiological parameters can help minimize this cost and enhance the quality of life for a significant part of the world's population. During hospitalization, monitoring these parameters is relatively simple and can be obtained with high-end bench-top diagnostic systems. To extend the control and the diagnostic capabilities out of hospitals it is necessary to provide unobtrusive and low cost systems, which should be equipped with adequate sensor interfaces, sufficient computational resources and with optimized power management strategies. One open challenge in this field is the design of unobtrusive systems that can be used in different applications of human biosignal analysis. Their goal is to be able to run the needed algorithms for monitoring and diagnosing the life parameters, while providing adequate communication interfaces

and a prolonged battery life. The design of a wearable system for biosignal monitoring presents many challenges, since it involves integrated circuit design, mixed analog-digital signal acquisition techniques, digital signal processing, low power algorithms and adaptive filtering techniques. Mobile and long-term recording of biomedical signals such as ECG, EMG and EEG can improve diagnosis and monitor the evolution of several widespread diseases. However, it requires specific solutions, such as wearable devices that should be particularly comfortable for patients, while at the same time ensuring medical-grade signal acquisition quality, including Power Line Interference (PLI). In this chapter we describe the acquisition of other biopotentials with the wearable node presenter in Chapter 4. We will enter in the details of two of the major issues related to wearable biopotential acquisition: the DC offset due to the capacitive coupling of the electrodes and the PLI removal. In the first part we describe how to remove the DC offset during an acquisition of EMG signal with our node, while in the last part of the chapter we describe the techniques to remove the PLI with adaptive and non-adaptive filters on EEG and ECG signal and their impact on the distortion of the acquired samples.

In recent years, there have been numerous research and commercial efforts in the design of wearable biopotentials measurement systems. There are already several low-cost devices on the market, like the ones from OMRON [81], Philips [86] and VIVAGO [8]. All these devices offer limited computational resources and are not designed for applications requiring flexibility e.g. in terms of multi-modality or number of channels. Their counterparts is represented by novel wireless portable and quite compact systems, such as the g.MOBILab+ by G.TEC [56], which at one side are more flexible, enabling multi-modal multi-channel biosignal recording, however being expensive and often requiring the use of proprietary software. At research level, physiological signals monitoring systems have appeared since the MIThrill2003 prototype [45], which represents a milestone in wearable computing platforms. It is based on a PDA connected with a sensor board for biopotential acquisition. The embedded sensor board is equipped with a 3-axes accelerometer, temperature sensor and an analog front-end for EMG-ECG acquisition. Sampled data is transmitted from the sensor board to the PDA for processing. Another important similar project is the AMON platform [15], a monitoring system composed by a wrist-worn device capable of measuring blood pressure, O_2 saturation, ECG and body temperature, which sends acquired data to a base station for remote storage, processing and support functionalities. Furthermore, there is a 2-axis accelerometer to correlate vital parameters to user activity. The project [79] proposes a system to measure and collect biological data for up to 9 hours in extreme environments. The board is based on a PIC microcontroller and can collect ECG, heart rate, blood pressure and saturation, body temperature and movements. The user can interact with the acquisition

system using a PDA, thus obtaining an unobtrusive system. However, the main limitation of all these architectures is the need to transmit data from sensors to a PDA or a base station. The sensor nodes are not equipped with sufficient computational resources, thus the data transmission limits the bandwidth of the processed signals and has a significant impact on power consumption. The recent advances in embedded system integration enable on-board signal processing required to improve the quality of the signals and possibly to perform part of the signal processing, thus opening the possibility to optimize the power consumption. These are the opportunities we want to exploit in this work. A wireless system performing on-board processing is presented in [32]. The system is equipped with a custom DSP and a TI MSP430 microcontroller. The DSP is used to perform Independent Component Analysis (ICA) and adaptive filtering to detect heart rates and cardiac arrhythmia. Although the system is well optimized for low power operation, it is based on a custom DSP independent from the low power commercial microcontroller, therefore limiting scalability in more complex and diverse applications. [85] presents a neck-band system for cardiac activity monitoring, which implements a CWT based BPM algorithm and an ECG derived respiration rate monitor. The data can be stored in an SD card or transmitted by a low power radio. The digital platform is an MSP430 and the ASIC analog front end offers great performance in terms of power consumption (21 μ V), but it is connected to the microcontroller with an analog interface. An analog back-end interface can be affected by additional noise and reduces the system scalability, when compared to digital interfaces. Recently, wearable systems started also to cope with EEG signal acquisition and monitoring. Several wireless and portable EEG monitoring systems have been published so far in literature. Some implementations exploit discrete components [100] and [40], while others rely on fully integrated systems, especially for the implementation of low-noise analog front-end circuitries. An example of a fully implemented wireless EEG sensor node is presented in [31] which uses the analog front-end published in [112]. All these systems are strongly oriented to a single application scenario therefore lacking in flexibility. Furthermore, they cannot be used in applications requiring multimodality, where data fusion from heterogeneous sensors is required. The lesson learned by these inspiring approaches is that we must join the design of a high performance AFE to allow the acquisition of the principal biopotentials (EEG, ECG and EMG) with an efficient microcontroller with integrated DSP functions. The AFE must have a digital back-end with SPI or I2C to provide a faster communication with the microcontroller, which must have sufficient computational resources to locally execute algorithms for filtering and information extraction, without data transmissions to a base station.

5.2 Biopotential Acquisition and Processing

The biopotential acquisition in the wearable devices uses the AC coupling as a preferred solution even though this setup is prone to errors in the signal acquisition. The decoupling capacitors cut the DC component of the signal as a High Pass filter. This approach is needed because the reference potential of the body is floating and differential measure is required. The high pass filter, the differences in the contact impedance of the electrodes and the floating ground potential cause a DC offset of the differential signal. This offset can saturate the signal span bringing saturation of the input and information content loss. Furthermore, even though the instrumentation amplifiers exploit a high Common mode Reject Ratio (CMRR) input stage, the differences in the values of non-ideal decoupling capacitors decrease significantly the CMRR giving rise to external noise interference, especially due to the PLI. In this chapter we describe the techniques used in the Cerebro platform to cope with these two major problems of the wearable biopotential acquisition. In the first part we present the HW solution implemented in Cerebro for the DC offset compensation while in the second part we focus on the on-board real-time PLI filtering on a low-power bio-potential acquisition wearable system. We analyze in depth basic and advanced PLI filtering techniques and evaluate them in a wearable real-time processing scenario, assessing performance on EMG and ECG signals. Our experiments prove that most PLI removal algorithms are not usable in this challenging context, because they lack robustness or they require off-line processing and large amounts of available data. On the other hand, adaptive filtering techniques are robust and well-suited for lightweight on-line processing. We substantiate this finding with off-line analysis and comparison, as well as with a complete embedded implementation on our low-power low-cost wearable device.

5.3 DC Offset Compensation

Recent technological advances and the application push from the consumer market have created a growing demand for wearable, low-cost and low-power devices with advanced bio-sensing capabilities. The processing of body-related potentials is becoming one of the main challenges in this field, leading towards efficient and unobtrusive applications. For this reason, the research on AFEs and on advanced energy efficient computing platforms is evolving both at silicon and at system level.

One of the most interesting biopotentials is the surface EMG signal that allows to detect the muscular activity of the human body [9]. EMG is widely adopted in human machine interaction applications, such as prosthesis control [20], even if it is affected by

high variability and ground loop problems. For this reasons, in prosthetics and medical applications, where a high-quality signal is required, expansive analog sensors are adopted. As mentioned in previous chapters, Ottobock sensors [7] represent the commercial solution for EMG acquisition for high-end prosthetics, both in research and industrial applications. These sensors perform a full-analog signal conditioning based on a bandpass discrete filter, an instrumentation amplifier (IA) with a high gain stage and an offset cancellation feedback circuit that requires the use of a dedicated metal plate as the reference electrode for each sensor.

An alternative approach, mostly used at research level, is based on passive low-cost electrodes, which requires to place a common reference electrode on the user's body in a neutral position (e.g. on the elbow when collecting forearm signals). In this case, there is only one common reference, which is not adequate for all the channels. Each channel is therefore affected by an offset dependent on the input impedance and it is influenced by its position and the physical connection between the electrode and the skin. The input impedance changes every time we place the electrodes on the skin and it changes during an acquisition due to skin condition or electrode movement.

With the common reference approach, a digital signal processing stage is required to obtain a well conditioned, zero mean differential signal, suitable for further analysis. Nevertheless, a correct offset removal is not possible if the signal is unbalanced and near the saturation voltage of the input amplifier. In the case of saturation, a digital offset removal on the acquired signal via mean subtraction is not possible without a loss in the signal content.

The approach based on dedicated ASICs is becoming more promising for wearable biosignal acquisition platforms, principally for the reduction of costs and external components. The Texas Instruments *ADS1299* and the Analog Devices *AD7194* are commercially available solutions, proposing AFEs with high CMRR, 24-bit resolution and a digital back-end suitable for integration with low power microcontrollers (MCUs). Nevertheless, the design of these components lacks flexibility because they are oriented to simplify the design of ECG systems with dedicated internal registers and circuits.

The literature of research on ASIC design is rich of inspiring solutions for EMG and other biopotential signal acquisitions [90, 26, 104]. These works are focused on performance optimization in terms of CMRR and ADC resolution or bandwidth, but they do not consider the DC offset removal necessary for the signal processing in prosthetics.

In fact, the offset due to input impedance and motion artifacts can be orders of magnitude higher than the biomedical signal being recorded. This tends to saturate a high gain IA causing losses of information content of the EMG signal. To overcome these problems, in [113] a pair of servo loops are used, combining a 16-element DAC and a fine analog

Table 5.1 Comparison with state-of-the-art AFE ASICs.

	Yazicioglu [113]	Bohorquez [29]	Muller [78]	Cerebro [94]
Channels:	8	1	8	8
Offset comp.:	analog fine servo	AC coupling	comp. DAC	comp. DAC
IR-noise [μV_{rms}]:	0.59	3.4	3.6	0.82
Tol. offset [mV]:	± 45	-	± 50	\pm 300

^a 1-100Hz signal bandwidth

feedback to remove DC offset up to ± 45 mV. An alternative approach is AC coupling performed at the input of the IA [29]. A comparison of the biomedical data acquisition ASIC used in this work with other state-of-the-art AFE ASICs is shown in Table. 5.1. One of the challenges for fully integrating the interface to a pair of electrodes is the DC offset due to input impedance and motion artifacts, which can be orders of magnitude higher than the biomedical signal being recorded and saturate the IA.

In this part, we show the digitally controlled feedback to cancel DC offset for the EMG signal acquisition. The DC removal is performed by the Cerebro internal DAC used to adjust the reference of each channel removing DC offset. In our approach, we couple the AFE with a MCU, which initializes and triggers when necessary the DC compensation. Each channel is independently analyzed and its offset is estimated and removed directly at the input using a binary-weighted 13-bit DAC. Its LSB is $75 \mu\text{V}$ so that the overall offset correction range is ± 300 mV.

Based on the analysis of collected signals, we estimated the offset evolution in time and we developed smart offset update policies to ensure it is always correctly removed. For this purpose, we periodically check the characteristics of the signals, to ensure we perform the offset estimation and removal when no contractions affect the captured signal.

The proposed solution presents two significant improvements w.r.t. the SoA. Firstly, we acquire a fully balanced and zero mean EMG signal, allowing to employ the full span of the ADC input while avoiding saturation and loss of signal content. Moreover, there is no need for a reference electrode for each channel, thus simplifying the system architecture, the AFE input stage and reducing the overall sensor area for a more scalable solution. The closed loop feedback of the digital part on the analog represents an innovative approach in biopotential signal conditioning and it is the main contribution of this work.

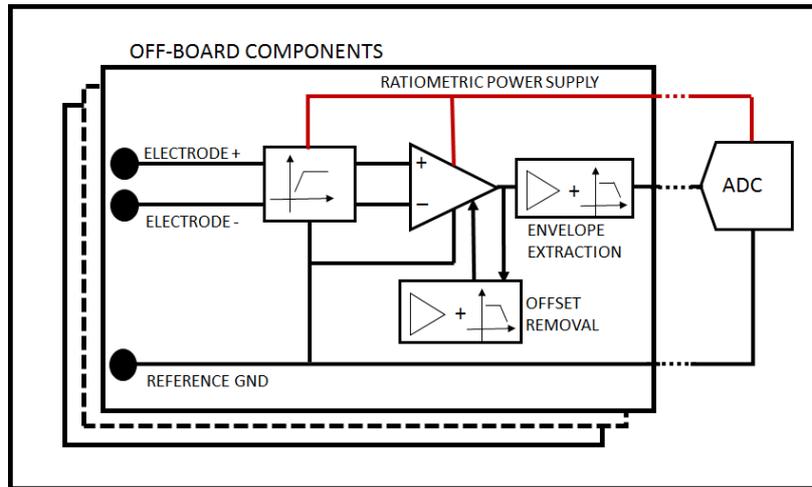


Fig. 5.1 Block diagram of the active sensors configuration.

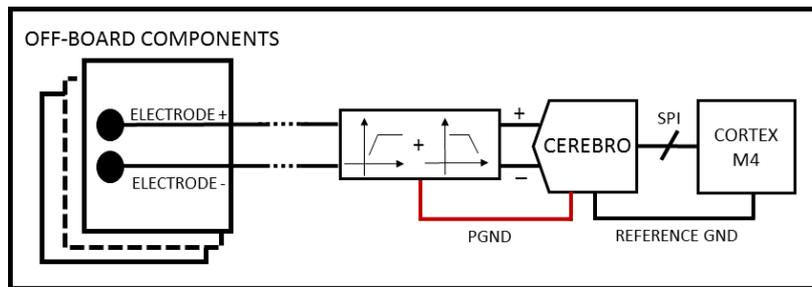


Fig. 5.2 Block diagram of the proposed solution.

5.3.1 System Description

The classical approach in active sensors is the use of 2 electrodes for the differential signal and a reference electrode for each sensor [7], as shown in Fig. 5.1. To detect the desired depolarization, the DC component is removed by a decoupling capacitor that allows to acquire the voltage change produced during the contraction. The AC component of the signal is the input of an IA in which the gain is set by a resistor net. In this solution, the offset compensation is based on an integrator circuit which sets the averaged mean value of the IA output as its reference. Finally, the envelop extraction is made by another opamp with a low-pass configuration. This approach removes the DC offset of the signal, but it needs several off-board analog components and requires to distribute the power supply and the ground reference to this off-board circuitry to have a ratiometric measure. Furthermore, the system is not easily scalable because for each new input it requires to add all the compensation and conditioning circuitry.

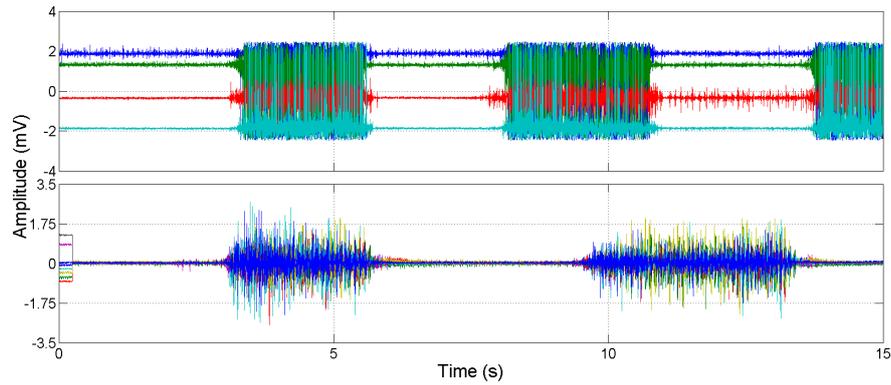
The proposed approach is illustrated in Fig. 5.2 and reduces significantly the off-board components, limiting them to the 2 differential electrodes. Each channel has passive low and

high pass filters mounted on the PCB board to decouple the input signal and reduce the low frequency noise. These filters are referred to the Patient Ground (PGND), a balanced voltage that can be set directly by the Cerebro ADC to modify the common mode voltage of the input. A significant improvement of the CMRR can be realized if a driven-right leg (DRL) is integrated on-chip to generate the patient ground [114]. The DLR is an electric circuit, often added to biosignal amplifier, designed to reduce the Common Mode Interference. An operational amplifier takes as input the signal coming from the inputs of the differential amplifier and The patient ground in the DRL circuit is derived from the common-mode voltage of the IA. Indeed, a DRL circuit is also used in practice to reduce the common-mode requirements of the IA. The algorithm to estimate and compensate the offset is executed in real time on the MCU and drives the internal DAC of Cerebro that compensate the DC offset. First, it checks if a contraction is in progress in the acquired EMG signal and eventually it waits until no contraction is detected. For this purpose, the standard deviation of the signal is computed and compared with a threshold. The DC offset is estimated as the mean value of each channel and it is computed on windows of $0.25s$ (250 samples at $1KHz$ sampling rate). If the calculated value exceeds the resolution of the DAC, a proportional controller uses the AFE gain and ADC resolution to compute the number of DAC steps to apply to compensate the estimated offset. Since the DAC is not extremely precise and in the acquisition chain it is placed before the AFE, this approach is eventually repeated until the offset is within the DAC resolution. In our tests we needed a maximum of 3 iterations ($750ms$) to perform the correction. This technique can be applied to other biopotential acquisition scenarios, allowing an effective offset removal, simplifying the conditioning circuitry and reducing the number of on-body electrodes.

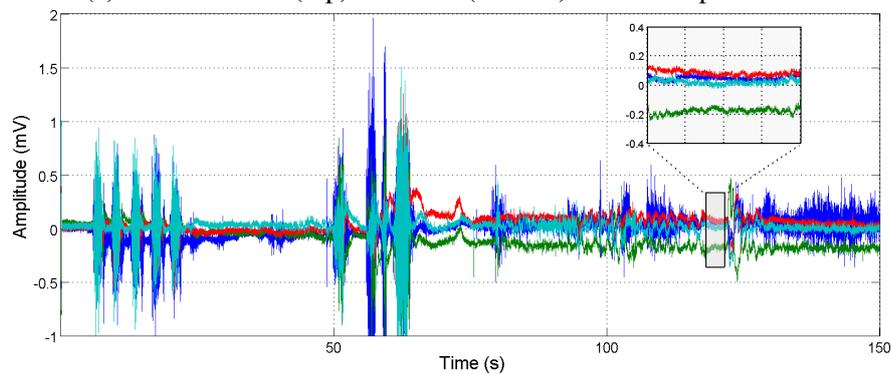
5.3.2 Experimental Results

The proposed approach was developed and tested with EMG signals collected with eight differential pairs of disposable electrodes placed in a circular cuff on the forearm of four healthy subjects. The signals were sampled at $1KHz$ with a gain of 32. This data collection involving human subjects was approved by the Institutional Review Board.

Figure 5.3a shows a sample of raw acquired data with no compensation performed (top) and data acquired when applying the proposed offset compensation approach (bottom). In the first case, every channel has a different offset, which is not known a priori since it is affected by a number of factors such as electrodes location, skin humidity, and body potential. Moreover, a high offset can cause the saturation of the acquired signal. When the offset compensation is applied, it is instead possible to exploit the full span of the ADC avoiding saturation and the related loss of information.



(a) EMG without (top) and with (bottom) offset compensation.



(b) Long time acquisition with offset compensation applied only during the initial phase.

Fig. 5.3 EMG signal DC offset (a) and drift (b)

The DAC implemented in Cerebro allows positive or negative fixed steps of $75\mu V$, regardless of the AFE gain. Thus, our compensation algorithm is able to reduce the offset within the $\pm 75\mu V$ range. Unfortunately, the signal offset is not stable in time, as can be seen in Fig. 5.3b where a longer acquisition is reported showing that some of the channels (the green one) drift in time. It is thus not sufficient to compensate the offset only at the beginning of each acquisition and we periodically check and remove it with the algorithm described above.

To evaluate the offset variations in time, we collected data from 4 subjects during different activities (walking, running, writing, working), with acquisitions of up to 30 min. From the analysis of the collected data we measured that in the worst case scenario the offset of a channel exceeds $\pm 150\mu V$ (two times the DAC resolution) after 30 seconds. We thus decided to execute the offset compensation routine every 30s.

It is crucial to perform the offset compensation only when no muscle contractions are in progress. We thus compute the standard deviation of the EMG signal and if it is higher than

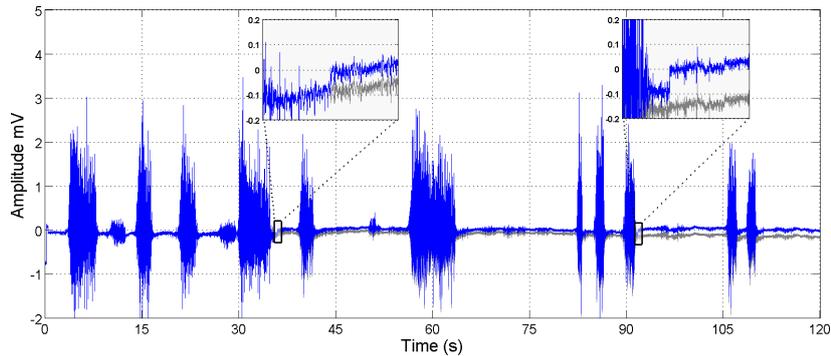


Fig. 5.4 Two channels acquiring the same EMG signal, with only the initial compensation (gray) and with periodic compensation (blue).

an empirically-set threshold, a muscle contraction is detected and the algorithm waits for its end to estimate the offset and to perform the compensation.

To evaluate the effects of periodic offset compensation, we connected two acquisition channels to the same pair of electrodes and collected again 10 minutes of signal from 4 users. In this case, on one channel the offset estimation and removal was applied only at the beginning, while on the second one we applied it periodically, every 30s. One sample of this acquisition is shown in Fig. 5.4 where the periodically corrected channel is plotted in blue. It is possible to notice that at 30s, when the compensation is triggered, a contraction is in progress and the algorithm waits for it to end to compensate the offset. At 60s there is no need to compensate the offset since the mean of the signal is within the DAC resolution of $75\mu V$ so the DAC settings are not modified. At 90s another contraction is in progress and the algorithm waits again to correctly compensate the offset.

Table 5.2 summarizes the overall performance of the proposed approach and shows the average offset on 10 minute acquisitions when no, initial and periodic compensation is applied. The initial and periodic approaches were applied to the same signal acquired connecting two channels on the same electrodes. The results show how the proposed periodic compensation reduces drastically the channel offset.

The proposed system, suitable for wearable acquisition of biopotential signals, was tested with forearm EMG signals commonly used for hand gesture recognition and prosthesis control. We collected signals of up to 30 minutes from 4 healthy subjects while they were performing various everyday activities. From this analysis, we observed that the EMG signals exhibit a random initial offset, which changes in time during the acquisitions. Thus, we implemented an algorithm which periodically checks the offset of each channel only if no contraction is in progress and sets the Cerebro ADC to automatically compensate the estimated offset. The results show that our approach correctly removes the DC offset from the acquired signals and allows the development of flexible and scalable wearable devices.

Table 5.2 Average offset with different acquisition techniques.

	No Comp. (mV)	Initial Comp. (mV)	Periodic Comp. (mV)
Subj. 1	1.62	0.29	0.05
Subj. 2	1.65	0.20	0.08
Subj. 3	1.33	0.26	0.08
Subj. 4	1.62	0.13	0.02
Mean	1.56	0.22	0.06

5.4 PLI Removal

In this sections we demonstrate the performance of our platform with two kinds of vital signs: the acquisition of an Electrocardiogram (ECG), based on a 3 leads configuration, and of an Electroencephalogram (EEG), using 2 fully differential channels. We implemented, profiled and compared four approaches for the PLI noise filtering. Furthermore, we show the performance of the acquisition system and we compare the results with a commercial state-of-the-art AFE (AD7194) for the ECG and a non-portable hospital device for multichannel bio-potential acquisition for the EEG. In the first part the systems are tested offline with different PLI methods to validate the approach of the cerebro system by comparison with other SoA commercial devices. In the second part we go through the real time advanced PLI removal comparing different techniques and showing an efficient adaptive filter with embedded implementation.

Different approaches have been presented in literature so far for the removal of PLI. We tested different PLI filtering algorithm with offline data to evaluate and compare the performance of acquisition of our system w.r.t some selected high-end systems. The simplest approach is a notch filter, which is a stop-band filter that allows to attenuate the frequency of a narrow band. The rejected band depends on the quality factor Q of the filter: with a $Q = 50$ the notch filter provides 10 dB attenuation at the frequency $f_{pli} \pm 0.5$ Hz. This approach has the advantage to be easily implemented and to have low computational requirements, but it introduces distortion in the signal power spectrum.

More advanced approaches have been developed in literature to overcome the limitations of the notch filter and to accurately separate the PLI from the EEG signal. In particular, there are methods based on time-domain subtraction [68], regression subtraction [18] and sinusoidal modeling [120]. These methods all share the basic approach, which consists in

the estimation of the sinusoidal interference and its removal from the acquired signal. Brief summaries of these techniques are listed below.

The time-domain subtraction method first divides the signal in linear and non-linear segments which is performed by setting a threshold on the second derivative of the signal. Then, in the linear segments, the signal is averaged and the PLI is estimated, which is then also removed from non-linear segments.

Regression-subtraction or time-correlated powerline interference subtraction estimates the amplitude and phase of the PLI and then subtracts it from subsequent samples. This approach models the interference as two quadrature sinusoids with the same frequency and uses blocks of data to estimate it with a least-squares fit.

The sinusoidal modeling approach models the interference by a set of harmonically related sinusoids modulated by low order time polynomials. The polynomial coefficients can be estimated by minimizing the quadratic error between the signal and the model in a given interval. Then the estimated PLI is subtracted from the original signal to obtain a noise free signal.

Time domain method relies on the separation of linear and non-linear segments, which is easily applicable to ECG signals, but is not useful in other cases. Regression-subtraction and sinusoidal modeling, on the other hand, use short intervals of signals (0.5 – 1.5 s) to estimate the sinusoidal source, which is then effectively removed from the original signals, preserving their other characteristics.

5.4.1 ECG Signal

The ECG signal is one of the most important biosignals that can provide a great amount of information in medical and fitness applications. It senses the electrical activity of the heart during its muscular contractions. During the heartbeat, the muscular cells on the heart surface depolarize their membrane. The resulting potential differences can be detected using surface electrodes placed in a proper configuration and a low noise signal amplifier. The typical frequencies of ECG signals go from 0.01 to 250 Hz and the amplitude is lower than 5 mV.

As all the biosignals, the ECG is difficult to manage because it is a low amplitude signal affected by different sources of noise (e.g. power line interference, baseline wander, ground loop noise, muscular contraction and respiration artifacts). For these reasons, in the design of an ECG signal detection system and, in general, for the design of a wearable device for biopotentials measurement, the system level hardware and software design is extremely important. All applications relying on the acquisition of biopotential signals share the common need to reduce noise and interferences by digital post-processing. The most

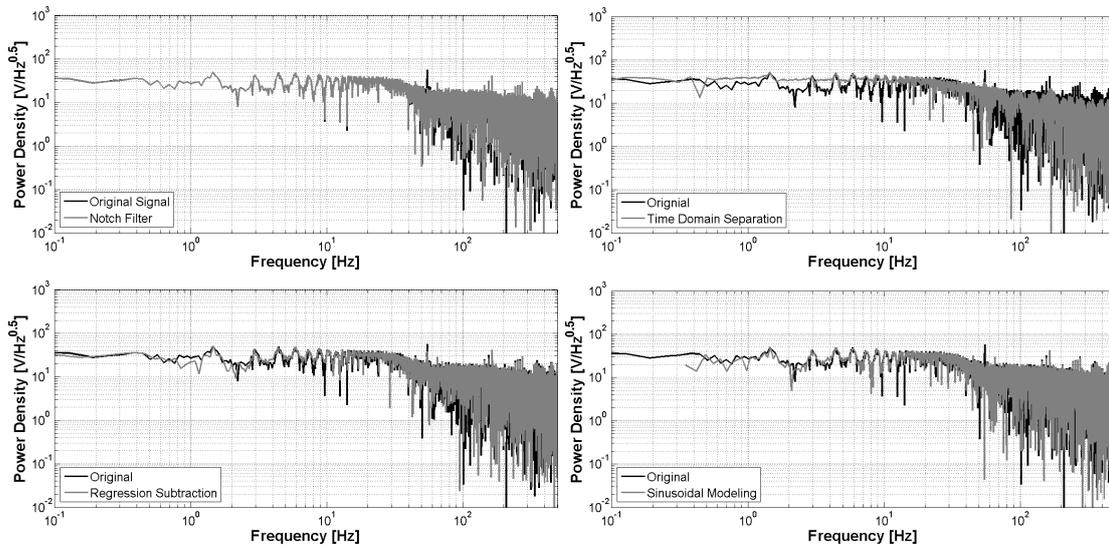


Fig. 5.5 Comparison of the FFT of the PLI filtering techniques (top to bottom): notch filter, time domain subtraction, regression subtraction and sinusoidal modeling.

common sources of interference are: powerline interference (PLI), baseline wander (drift), movement and breathing artefacts, changes in the electrode impedance and intra-channel interference. In particular, PLI introduces a 50/60 Hz sine interference to the signal. It is always noticeable, even when the system is battery-powered [96], that its accurate removal is a critical but required task.

To test the capabilities of our system and to compare the different de-noising approaches, we used a 3-lead ECG acquisition scenario. In this set-up, we used one differential channel of the AFE to acquire the ECG signal. We placed two disposable electrodes on the wrists of the user and an additional electrode was placed on the ankle as the reference potential (patient ground).

The ECG signal was acquired with our platform and with a reference state-of-the-art AFE, which is the Analog Devices AD7194 chip. We used the development board provided for this chip, which is equipped with 8 analog channels and can be connected to a PC via USB for the data acquisition. Our device sent the sampled data to a PC via Bluetooth and all the collected data was stored and processed on the PC. The two systems have been configured in the most similar way possible, setting the acquisition frequency at 1 kHz and the gain at 8 for our device, while the ADC was set to sample at 960 Hz with a gain of 128.

The signals acquired from the two devices are plotted in Fig. 5.6, along with their frequency spectrums. From the plots we can observe that the two systems provide signals of comparable quality. The frequency spectrum of the two acquired signals is very similar. The Cerebro ASIC acquired signal exhibits a strong PLI component at 50 Hz (the exact measured

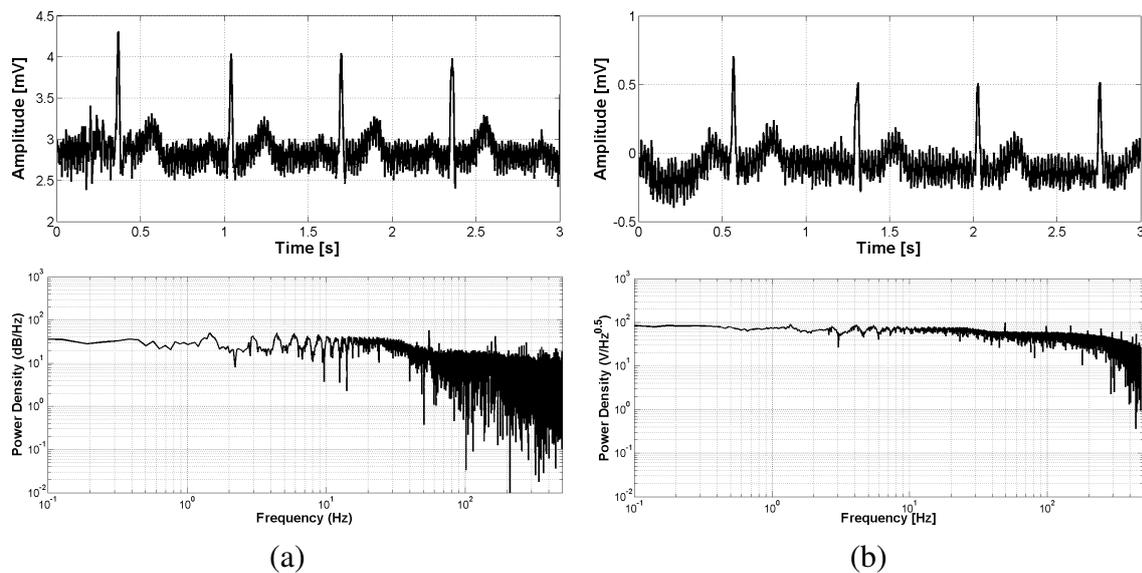


Fig. 5.6 Raw ECG signals (top) and their FFTs (bottom) for the Cerebro platform (a) and for the AD7194 acquisition board (b).

frequency of the PLI was 55 Hz). The ADC has an internal filter, which reduces the PLI contribution. Using the data acquired with Cerebro, we implemented and compared the PLI filtering techniques described in the previous Section. Ideally, the perfect filtering technique should remove the PLI component and leave the rest of the signal frequency spectrum as it is.

The result of the PLI removal for the considered approaches is shown in Fig. 5.5, where we plotted the frequency spectrum for the raw and the filtered signals acquired with our system. From the plots we can see that all the considered algorithms remove the PLI component, but they alter the rest of the frequency spectrum in different ways. We can note that the notch filter removes also frequency components close to the PLI, while the time-domain approach alters considerably the spectrum of the output signal, for both lower and higher frequencies.

The regression-subtraction method is the one that alters less the frequency spectrum of the signal, only reducing the PLI component. Also the sinusoidal modeling approach is very precise and removes correctly the PLI without additional changes in the signal. It is worth noting that the signals also present interferences at frequencies multiple of the main PLI component (e.g. 100 Hz), which can be removed applying the considered approaches also for those frequencies.

To summarize the results of this comparison, we computed the SNR of the filtered signals, obtaining 21.9 dB for the notch filter, 28.3 dB for the time-domain subtraction, 29.5 dB for the regression-subtraction and 30.4 dB for the sinusoidal modeling approach. The latter two methods deliver considerably better signal quality and are the ones chosen to be used in our

system. In particular, the regression subtraction method has been implemented and used for the final ECG and EEG applications, since it is the one that preserves better the frequency response of the system.

5.4.2 EEG Signal

EEG represents a collection of electrical voltages recorded at different locations on the scalp of patients. Electrical characteristics of these signals show a typical bandwidth range from 0.5 to 100 Hz with a peak amplitude of about $100\ \mu\text{V}$. Such signals are generated by millions of underlying neurons that fire asynchronously and are responsible for the brain activity. Hence, the EEG recording does not contain the activity of single neurons but the averaged activity of millions of neurons. For this reason, raw (unprocessed) EEG signals do not show any kind of regularity in the time-domain. However, after proper band-pass filtering of the EEG signals, e.g. extracting delta, theta, alpha and beta frequency bands, more regular patterns can be identified, especially in the lower frequency bands. These filtered EEG bands are of high interest because they are strictly correlated with the states in the brain such as wakefulness, sleep, or even with some severe diseases including epilepsy and neoplasms [77], [33] and [80].

EEG evaluation is thus an important tool to learn about brain functioning. The understanding of brain functions, however, is currently limited to clinical environments and may not accurately reflect brain activities in the real world. Furthermore, long recordings are feasible only during sleep as the EEG amplifiers are large, inconvenient for patients, heavy, and need to be plugged in, making it unable for patients to move more than a few meters.

On the contrary, a wearable EEG device is not restricted to these limitations, exploiting paradigms of integration, low power operation, and small device size [35]. This increased degree of freedom for the wearable EEG device allows to record biological signals also outside of the clinical laboratory, increases the interests in the research field that is currently restricted to medical use cases. Such a portable device has countless applications with high market potential, ranging from early detection of diseases to the monitoring of well-being habits and cognitive behavior.

To provide a satisfactory wearable EEG device, it is essential to build it such that its performance is comparable to those obtained in the state-of-the-art clinical devices. We therefore compare our system with the commonly used bench-top clinical device. It is shown that we obtain similar performance in field tests.

A spectrogram is shown in Fig. 5.7, which gives a better insight on the overall performance of the wearable EEG device. The color map on the right hand-side has the dimension decibel and indicates the power of the recorded signal. A $100\ \mu\text{V}$ sinusoidal signal is sampled

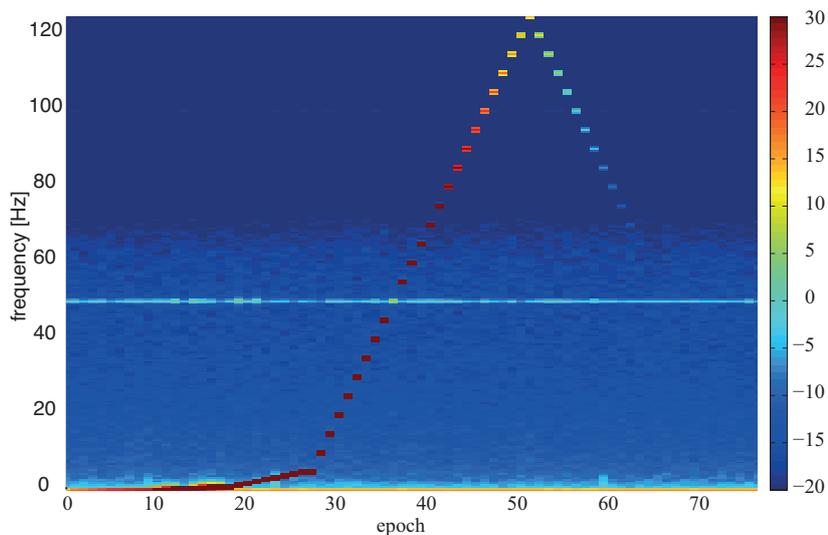


Fig. 5.7 Color-coded power spectra of consecutive 90-s epochs (4 s window) with Hanning window. Data was sampled with 250 Hz. Spectra are color coded on a logarithmic scale, 30 dB corresponds to an input signal with an amplitude of $100\mu\text{V}$.

with the device at a sampling frequency of 250 Hz while the input sinusoidal signal frequency is increased over time from 0 to 250 Hz. Each EEG channel of the device is low-pass filtered at a cutoff frequency of 66 Hz. The dots on the illustration represents the sinusoidal amplitude changes (from grey to white) indicating the signal attenuation by the low-pass filter. The spectrogram also shows signals aliased back in the Nyquist band which were generated by feeding the EEG device with sinusoidal signals with frequencies higher than half of the sampling frequency. The horizontal gray line at 50 Hz is the residual of the mains interference being successfully suppressed to a certain degree and does not significantly disturb the EEG signal. This spectrogram illustrates that the Cerebro AFE is well-suited to record EEG signals with amplitudes below $100\mu\text{V}$. In the frequency band of interest, i.e., from 0.5 to 100 Hz, a signal-to-noise ratio of more than 35 dB is observed.

In order to provide a useful wearable EEG device, it is essential to build it such that its performance is comparable to those obtained in the state-of-the-art in clinical use. For this reason, a direct comparison with the commonly used state-of-the-art device in hospitals is performed in this paper to show that the wearable EEG device achieves similar performance.

The specification of the wearable EEG device is performed taking the recorded EEG signals from both devices, i.e., our wearable EEG device with limited hardware resources and the state-of-the-art recording device in operation at the University Hospital of Zurich (USZ) which has no limitations in power consumption and hardware resources. The EEG

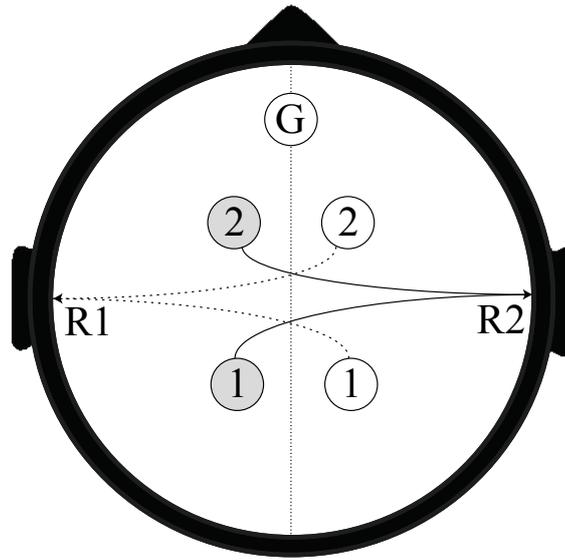


Fig. 5.8 Illustration of the electrodes arrangement on the head of the patient with the denoted patient ground G. The pairs of electrodes are split in colors where the gray ones are referenced to the right ear (R2) and the the white ones to the left ear (R1).

signals are simultaneously acquired from the same patient at a sampling frequency of 250 Hz with the wearable EEG device and at 256 Hz with the USZ device.

Fig. 5.8 illustrates the placement of electrodes on the scalp of the patient. During the recording, a test subject is connected to different EEG channels, where two out of four channels are connected to the wearable EEG device (numbered gray electrodes), while the remaining two are connected to the state-of-the-art device used in USZ (numbered white electrodes). The pairs of electrodes connected to both devices are placed next to each other in order to record EEG signals from the common source.

The current measurement setup in use at USZ consists of polygraphic amplifiers provided by Artisan (Micromed, Mogliano Veneto, Italy) and the recording is performed using the Rembrandt Datalab (Embla System, Broomfield, CO, USA). Before analog-to-digital quantization is done, the analog EEG signals are high-pass (-3 dB at 0.16 Hz) and low-pass filtered (-3 dB at 67.2 Hz), as indicated in [77].

In the wearable EEG device, the analog front-end [92] samples the analog signals at 250 Hz. After the analog-to-digital quantization, digital high-pass (-3 dB at 0.16 Hz) and low-pass filtering (-3 dB at 66 Hz) is performed on the digitized signals.

After acquiring the EEG signals from the two acquisition systems, a post-processing filter has been applied in order to highlight major cerebral waves of the patient. As illustrated in Fig. 5.9, the EEG Channel-1 of both devices deliver similar EEG signal patterns for the different EEG bands of interest (Channel-2 shows very similar behaviour, not shown). For

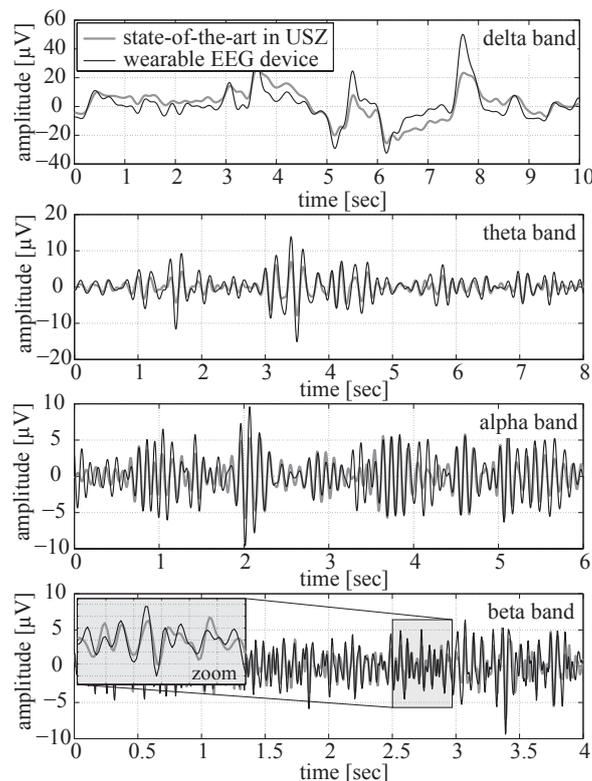


Fig. 5.9 First pair of EEG channels comparing the wearable EEG device with the state-of-the-art in USZ. Brain waves are obtained after proper filtering and decomposing of the EEG signals in delta, theta, alpha, and beta frequency bands.

both devices, waves are obtained according to the following scheme: delta low-pass filter at 4 Hz, theta band-pass filter between 4 and 7 Hz, alpha band-pass filter between 7 and 15 Hz while beta is band-pass filter between 15 to 30 Hz.

FFT plots of the two acquired channels are shown in Fig. 5.10, which provides a different perspective of the measured EEG. Frequency responses of both devices overlap over the whole Nyquist frequency band. The only noticeable difference is that the USZ device uses a notch filter to cancel mains interference while the wearable EEG device handle this problem by applying the regression subtraction approach and subtracting the estimated interference signals [92].

Furthermore, Cerebro offers higher scalability w.r.t. comparable commercial devices or other research prototypes and higher flexibility in terms of multi-modality. Therefore, we think that the proposed wearable platform has high potential to be used not only for the monitoring of vital signs, but also for biomedical real-time signal processing. The Bluetooth interface allows to connect the Cerebro board to mobile devices paving the way to the development of efficient user interfaces for clinician and patients. In the next paragraphs

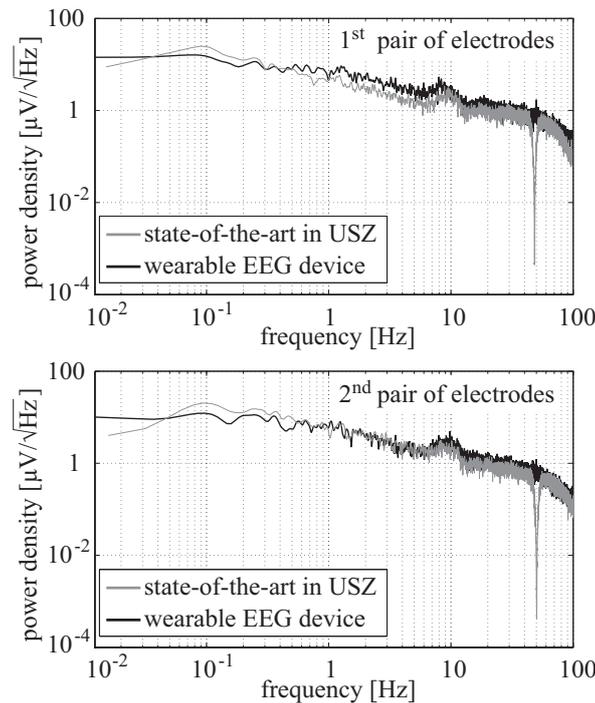


Fig. 5.10 Frequency response of the two electrode pairs. FFTs are very similar in amplitude except for the notch filter at 50 Hz of the USZ acquisition system.

we will analyze an on-line real-time solution for advanced PLI removal overcoming the limitations that affect most of the common PLI filtering techniques.

5.5 Real Time PLI Removal

The PLI is due to the capacitive coupling between the subject and nearby electrical appliances and mains wiring [41] and represents a major common source of interference. The main frequency of the PLI is nominally at 50 Hz in Europe and 60 Hz in USA. However, it is non-stationary and frequency and amplitude variations are often detected and mainly originated from the AC power system. In particular, its frequency has variations of ± 2 Hz, while the amplitude is heavily influenced by the system in use and the environment [74, 46].

Many solutions have been developed to reduce the interference in the acquired biomedical signals. Useful ways to reject the interferences include the use of active electrodes with integrated analog filters, the shielding and connection to ground of electrodes, subjects and nearby electrical appliances. In spite of these solutions, a significant residual interference remains. Moreover, the proposed solutions are often invasive and not suitable for wearable solutions.

Wearable sensor nodes are usually equipped with limited computational resources, thus the most common approach to reject the PLI from biomedical signals is the use of analog filters. This solution introduces non linear phase shifts, skewing the signal and it is costly in terms of component count and board space. Thus, additional signal processing techniques are necessary to filter the noise and achieve a robust output. The desired solution should remove the PLI even if it is non-stationary, while minimally affecting the frequency signal spectrum of interest. Furthermore, for a successful application in wearable solutions, it should be suitable for real-time implementation in resource constrained devices, requiring low computational complexity and low memory cost.

We evaluate the real-time parameter monitoring with digital PLI filtering. We compared the four major classes of approaches for the PLI noise filtering, evaluating them with simulated signals and interference for an accurate study in a wide range of situations. Moreover, we implemented and profiled them on the embedded wearable platform evaluating accuracy and real-time performance. Based on this analysis, we demonstrate a highly effective PLI removal on our wearable smart platform in two applications: the acquisition of a 3-lead ECG, and the acquisition and processing of EMG signals.

5.5.1 HW Techniques for PLI Removal

Several solutions for the removal of the PLI have been studied in the past decades. One of the approaches is the design of an analog front end with high Common Mode Rejection Ratio (CMRR). Hardware solutions have been developed to increase the actual CMRR by equalization of the cable shield and the use of a Driven Right Leg (DRL) circuit [75]. The DRL improves the CMRR by a closed loop compensation that provides a CM-canceling signal to the subject's body. This reduces the CM gain and boosts the CMRR, but it is affected by stability issues, which limit the effective improvement. The digitally-assisted DRL tunes the DRL loop's gain at the power line frequency, but needs an additional notch filter increasing power consumption. The common-mode feedback (CMFB) technique [110] improves CMRR by feeding the CM voltage back to the input of each pre-amplifier. The feedback loop is based on a summing amplifier and high value compensation capacitors to efficiently extract the CM noise. All these solutions require increased area, extra components and lead to an increased power consumption with a reduction of the common mode interference that is hardly higher than 40 dB [57].

5.5.2 SW Techniques for PLI Removal

For a more robust PLI removal, digital signal processing techniques are employed. The most common approach used to remove the PLI is a digital notch filter at the power line frequency (50 or 60 Hz) [62, 84, 88]. It has the advantage to be easily implemented and it has low computational requirements. However, the notch filter is not a good candidate to remove the PLI from biomedical signals with time-varying disturbances: to avoid signal distortion around the PLI frequency it needs a narrow notch frequency band, which leads to ineffective filtering when PLI frequency deviations are present.

A different approach is the time domain subtraction for PLI elimination [68]. This method first divides the signal in linear and non-linear segments by evaluating a threshold on its second derivative. In the linear segments, the signal is averaged and the PLI is estimated, which is then removed from both linear and non-linear segments. This approach is limited to the processing of ECG signals, which exhibit a periodic alternation between linear and non-linear segments. In fact, it is not suitable for the denoising of EMG or EEG recordings since the linear and non linear segments in those cases cannot easily be identified in the presence of the PLI. Furthermore, the time domain subtraction alters considerably the spectrum for both lower and higher frequencies.

To overcome this limitation, newly developed techniques include the Sinusoidal Modeling (SM) [121] and Regression Subtraction (RS) [93], which accurately separate the PLI from biopotential signals. These methods share the same base approach, which consists in the estimation of the sinusoidal interference and its removal from the acquired signal. They estimate the amplitude and phase of the PLI and they reject the interference while minimally affecting the spectrum of the signal. On the other hand, these techniques lose their effectiveness when the frequency of the PLI is not constant and its deviation from the nominal value is not known. To overcome this drawback, it is necessary to calculate the Fast Fourier Transform (FFT) of the signal to localize with precision the value of the fundamental PLI frequency. This solution is not suitable for a real time implementation due the high computational cost to perform the FFT with adequate precision.

Another approach is the use of adaptive interference cancellation, which can track amplitude, phase and frequency of the PLI [108]. A drawback of this solution is that it requires a reference signal to estimate the frequency variations, which may not always be available in practice. To overcome this limitation, reference-free adaptive methods have been proposed in literature [119, 72]. Such methods are unfortunately sensitive to the ratio between the PLI and the signal amplitude in the interference frequency band. Therefore, they are not general and the parameters must be tuned case by case. An innovative adaptive filtering method was introduced by Keshtkaran et al. to track time fluctuations of the PLI's

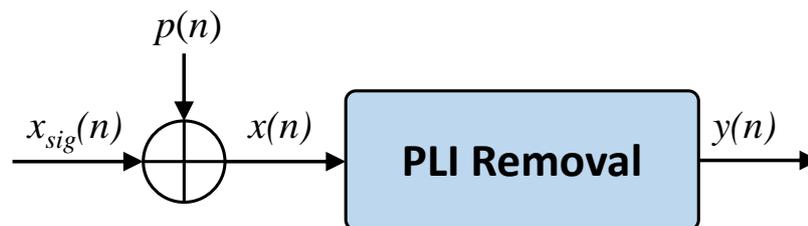


Fig. 5.11 Signal acquisition block diagram.

frequency, amplitude and phase [64, 65]. It can be used to process ECG, EEG and EMG signals without any reference signal, with the advantages of low computational complexity and low memory requirements, hence making it suitable for real time implementation in smart wearable solutions.

5.6 PLI Methods Overview

This section details the major digital filtering techniques and their parameters. In the rest of this paper, we denote: $x_{sig}(n)$ as the true biomedical signal, $p(n)$ as the PLI, $x(n)$ as the contaminated signal and $y(n)$ as the filtered output signal (see Fig. 5.11).

Notch Filter. The notch filter is a stop band filter that allows to attenuate component frequency in a narrow band. The transfer function of a real second order IIR filter is given in the following equation:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{b_0 + b_1z^{-1} + b_2z^{-2}}$$

where a_0, a_1, a_2 are the feed-forward and b_0, b_1, b_2 are the feedback coefficients of the filter. The transfer function is implemented with the difference linear equation as follows:

$$a_0y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n)(n-1) - a_2y(n)(n-2)$$

In our implementation, we use a second order IIR filter centered at the PLI frequency with a bandwidth of 1Hz. In our experiments, the quality factor was set at 50 in order to reduce the PLI interference at approximately the same level of the ExG signal and thus avoid distortion of the bio-potentials in the frequency spectrum.

Sinusoidal Modeling. SM allows to remove the PLI and the baseline wonder (BW) from the biomedical signal. This method models PLI and BW with a set of sinusoids modulated

by low order time polynomials. It is expressed as follows:

$$p_{est}(t) = \sum_{i=0}^I (a_0^{(i)} + a_1^{(i)}t + a_2^{(i)}t^2) \sin(2\pi i f_{PLI} t) + (b_0^{(i)} + b_1^{(i)}t + b_2^{(i)}t^2) \cos(2\pi i f_{PLI} t)$$

where I is the total number of signal components in the model. The BW corresponds to the component with $i = 0$, while the PLI harmonics are modeled by the remaining ones ($i \neq 0$). The sinusoids represent the harmonic character of the PLI signal with the fundamental frequency ($i = 1$) corresponding to the PLI frequency (f_{PLI}) in the analysis window. Moreover, the model is extended to include the DC component ($i = 0$), whose time variations correspond to the BW signal component.

The coefficients of the model can be estimated by minimizing the quadratic error between the signal $x(t)$ and the modeled PLI:

$$\min \left(\sum_{n=1}^N |x(t_n) - p_{est}(t_n)|^2 \right)$$

where t_n are the sampling instants in the analysis window. Finally, the estimated model is subtracted from the original signal to obtain a noise free signal $y(t) = x(t) - p_{est}(t)$.

We implemented this method to remove the first component of the power line interference ($i = 1$) with $f_{PLI} = 50Hz$ and a window size of $1.5s$ as recommended in [121]. In our implementation, we do not include the DC component in the model, since Cerebro is equipped with an internal DAC used to adjust the reference of each channel and remove any DC offset. The algorithm can be used in two modalities: in the first one, denoted as offline mode, the PLI is estimated and then subtracted from the same analysis window. In the second method, denoted as online mode, the PLI is estimated in one signal window and then it is subtracted from the subsequent window. The latter modality allows to use this approach in real-time applications.

Regression Subtraction. The acquired signal $x_n = x_{sig}(n) + p(n)$ is the superposition of the biomedical signal $x_{sig}(n)$ and the PLI components, modeled as:

$$p_i(n) = A_i \cdot \sin\left(2\pi \frac{f_{PLI,i}}{f_s} n + \phi_i\right)$$

with amplitude A , phase ϕ and harmonic number i . This method considers a window of N samples multiple of $(f_s/f_{PLI,i})$ where the phase and amplitude of the PLI are estimated in two steps. In the first step, in order to estimate ϕ_i , the algorithm projects the acquired sequence $x(n)$ onto a locally generated cosine with known phase:

$$\frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos\left(2\pi \frac{f_{PLI,i}}{f_s} n + \varphi_{est,i}\right) = \frac{1}{2} A_i \sin(\varphi_i - \varphi_{est,i})$$

Following, the method executes an iterative binary search to find the estimated phase $\varphi_{est,i}$. In the second step, the amplitude estimation ($A_{est,i}$) is found by a projection onto an in-phase sinusoidal signal, as follows:

$$A_{est,i} = \frac{2}{N} \sum_{n=1}^{N-1} x(n) \sin\left(2\pi \frac{f_{PLI,i}}{f_s} n + \varphi_{est,i}\right)$$

The estimated PLI noise is found using ($A_{est,i}$) and ($\varphi_{est,i}$) as follows:

$$p_{est,i}(n) = A_{est,i} \cdot \sin\left(2\pi \frac{f_{PLI,i}}{f_s} n + \varphi_{est,i}\right)$$

Finally, the estimated noise is subtracted from the original signal to obtain a noise free signal. This method was implemented in software to remove the first component of the PLI, with $f_{PLI} = 50\text{Hz}$ and a window size of 1s . RS can also be applied in two modalities, online and offline, as described for SM.

Adaptive PLI filter.

In our evaluation, we focus on the adaptive PLI filter (APF) proposed in [64] and [65], due to its capability to process heterogeneous vital signs and their low computational and memory requirements. This approach iteratively estimates the fundamental frequency of the PLI and then generates its other harmonics. At each sample, the estimated PLI is subtracted from the noise-affected biosignal in order to reject the PLI. The APF distinguishes two methods for the estimation of amplitude and phase of the interference: in APF LMS the amplitude and phase of each harmonic are obtained using a least mean square algorithm, while the APF RLS uses a simplified recursive least squares algorithm to approximate the PLI parameters. Beside the parameter estimation algorithm, the two APF approaches are equal.

Fig. 5.12 shows the functional blocks of the APF algorithm. The bandpass filter is used to preprocess the signal to enhance the fundamental harmonic of the PLI and to obtain a robust estimation of its frequency. This filter is also useful for the attenuation of lower frequency artifacts and signal components, which may negatively affect the frequency estimation. If the nominal power line frequency is known to be at 50Hz , the bandpass filter can be set to $45\text{-}55\text{Hz}$, but it can be further customized to accommodate both 50Hz and 60Hz powerline frequencies.

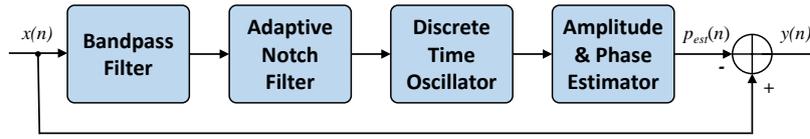


Fig. 5.12 Block diagram of the adaptive PLI filter.

The next stage consists in an Adaptive Notch Filter (ANF) for frequency estimation. It is implemented through a lattice algorithm [43] to obtain high performance in the instantaneous estimation of the PLI frequency with low complexity and suitability for real-time finite precision implementation. Next, the discrete-time oscillators and the amplitude phase estimator are used to generate the estimated PLI sinusoid. Defining the estimation error as $e(n) = x(n) - p_{est}(n)$, the APF LMS method minimizes the cost function:

$$E(n) = \mathbb{E}|e(n)|^2$$

while the APF RLS minimizes the cost function:

$$E(n) = \sum_{n=1}^{N-1} \lambda_a^{(N-1-n)} e^2(n)$$

In this case, λ_a is the forgetting factor and $0 \ll \lambda_a < 1$. Finally, the estimated interference $p_{est}(n)$ is subtracted from the input signal $x(n)$ to obtain a noise-free signal $y(n)$. In our experiments, we tested both approaches to remove the first component of the PLI and we followed the guidelines presented in [64] for a correct adjustment of the parameters in each block.

5.7 Experimental Results

In this section, we present an extensive analysis using off-line ECG data and simulated PLI to quantitatively evaluate the methods under various signals and parameters conditions. Furthermore, the algorithms were all implemented on the microcontroller of our wearable device and they were tested on real-time acquisition and filtering of ECG and EMG signals.

5.7.1 Off-line analysis setup

To analyze the characteristics of the aforementioned algorithms under various signal conditions, we used ECG data taken from the PTB Diagnostic ECG Database, which is part of the Pysionet project [54]. The biopotentials are sampled at 1 kHz with 16-bit resolution

over a $\pm 16.384\text{mV}$ range and have variable duration (from 30s to 2min). In each test, we added a synthetic PLI containing 1 harmonic to the ECG signal, with variable frequency and amplitude as specified in each simulation. To denote the signal to noise ratios (SNRs) of the input and output signals we used SNR_{IN} and SNR_{OUT} , defined as:

$$SNR_{IN} = 20 \log_{10} \frac{\|x_{sig}\|_2}{\|p\|_2}$$

$$SNR_{OUT} = 20 \log_{10} \frac{\|y\|_2}{\|x_{sig} - y\|_2}$$

In medical grade ECG setup no specific biomedical requirements apply to SNR. Hence, it is necessary a comparison between the wearable device and the state-of-the-art non-portable device, used in clinical applications. In [21] we compared our wearable ExG platform with a polygraphic amplifiers provided by Artisan (Micromed, Mogliano Veneto, Italy) in particular for EEG acquisition. We verified that the data acquired by the two systems has the same temporal evolution and frequency spectrum.

5.7.2 Sensitivity to Power Line Amplitude

To achieve a proper interference cancellation, the algorithm should work robustly under several SNR_{IN} conditions. To evaluate this feature, we simulated signal sequences whose SNR_{IN} ranged from -20dB to 20dB. For each SNR_{IN} value, we added a 50Hz sinusoidal interference to the signal and each method was applied to remove such interference, computing the resulting SNR_{OUT} . Figure 5.13 shows the SNR_{OUT} for each SNR_{IN} at the sampling rate of 1 kHz.

In this test, executed with a constant PLI frequency, we can observe that both online and offline modalities reach the same performance in removing the PLI for all SNR_{IN} values. In fact, high values of SNR_{OUT} are noticed for both RS (36dB) and SM (31dB) approaches. The APF RLS and APF LMS achieve a constant SNR_{OUT} of 35dB and 29dB respectively, indicating that the performance of the APF methods is robust with respect to SNR_{IN} . The notch filter demonstrates to be the worst method in this comparison, with $SNR_{OUT} = 25\text{dB}$.

5.7.3 Sensitivity to Power Line Frequency Variability

In real life scenarios, the frequency of the PLI is not constant at exactly 50Hz, hence it is important to evaluate the performance of these techniques with regard to PLI frequency

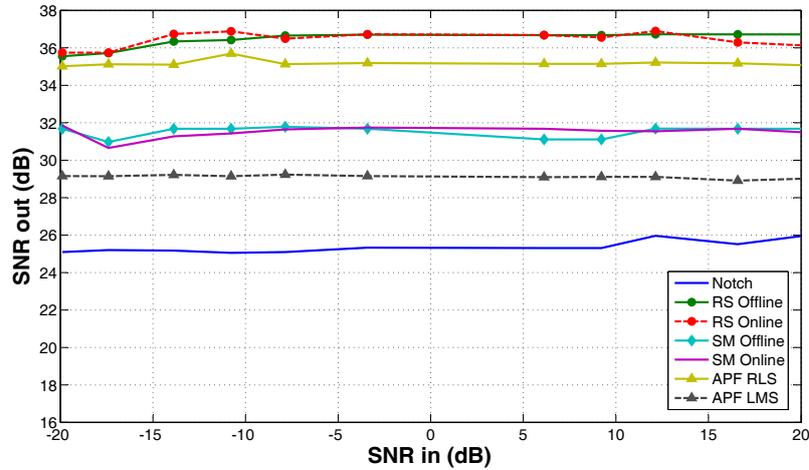
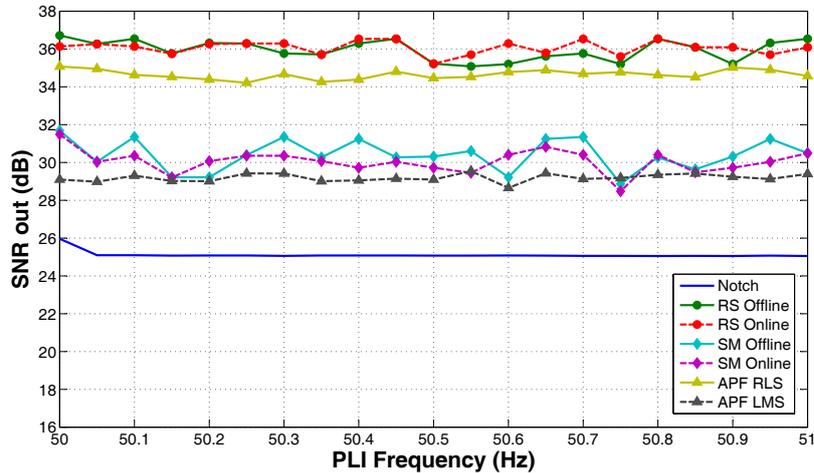


Fig. 5.13 Sensitivity to PLI amplitude.

Fig. 5.14 Sensitivity to power line frequency variations when f_{PLI} is known.

variations. The notch filter, SM and RS, all need to know a priori the fundamental frequency of the interference (f_{PLI}). On the other hand, the APF approach can automatically detect the value of the frequency interference and no a priori setting of its nominal value is required.

To test the capabilities of the algorithms to remove the PLI under varying powerline frequencies, we used as the input of each method synthetic sequences with SNR_{IN} equal to $6dB$ and PLI frequency ranging between 50 and $51 Hz$. We computed the output SNRs to evaluate the performance of the algorithms. In this test, we assumed to know a priori the frequency value of the powerline interference and set f_{PLI} correctly in each trial. Figure 5.14 shows the SNR_{OUT} achieved by each method. We can observe that SM, RS and APF, all achieve a SNR_{OUT} of up to $29dB$. The notch filter is again the worst method, with $SNR_{OUT} = 25dB$.

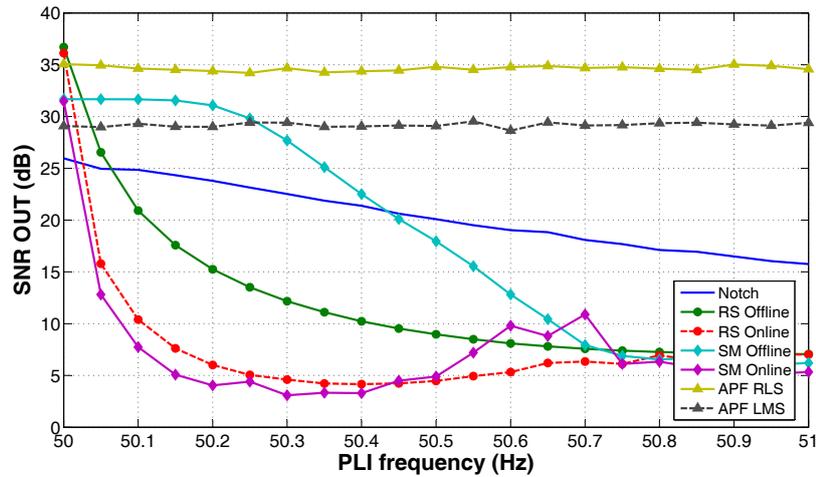


Fig. 5.15 Sensitivity to power line frequency variations when f_{PLI} is unknown.

The accurate value of the PLI is unknown in a real scenario, and may also change over time, thus it is important to test the performance of the algorithms when f_{PLI} is not known. Hence, we performed the same experiment as described before, but with f_{PLI} always set to 50 Hz. Figure 5.15 shows the resultant SNR_{OUT} . Here, we can observe that for a frequency deviation smaller than $0.01Hz$ the notch filter, SM online and RS (online and offline) correctly remove the PLI from the biomedical signals, with $SNR_{OUT} > 30dB$. For a frequency deviation higher than $0.01Hz$, all these algorithms lose their effectiveness. In fact, we can observe an SNR_{OUT} lower than $30dB$ for all of them. SM offline is adequately robust to achieve an SNR_{OUT} of up to $30dB$ only for frequency deviations lower than $0.3Hz$.

Moreover, the performance of SM and RS online deteriorates rapidly if the number of samples in a window used to estimate the interference is not an exact multiple of the number of samples in a PLI period. In the non-multiple sampling case there is a variance of the initial PLI phase between two consecutive windows, therefore the PLI estimated in one signal window and then subtracted from the subsequent window is not correctly aligned.

The notch filter, on the other side, shows a lower but slowly declining SNR_{OUT} , hence resulting more robust to frequency deviations. The best performance is achieved with the APF approaches, in particular the APF RLS shows high values of $SNR_{OUT} (\geq 35dB)$ consistently achieved for varying PLI frequencies deviations.

SM and RS algorithms need an accurate estimate of the f_{PLI} to achieve good results. The frequency of the interference is estimated by computing the FFT of the input signal and localizing the strongest bin in the 45 - 55 Hz range. As resulted from experimental analysis, high performance is achieved when the estimate is accurate with a resolution of $0.01Hz$. The number of samples required to achieve such resolution depends on the signal's sampling period. When sampling at $1kHz$, we need 100000 samples, which implies that we have to

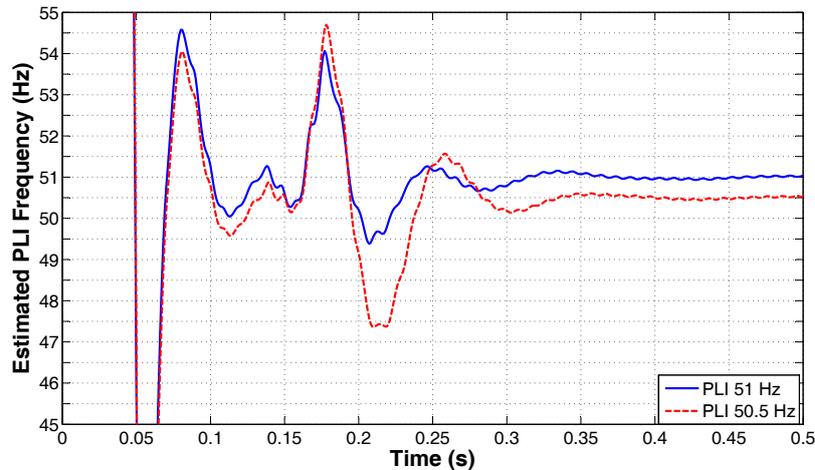


Fig. 5.16 Convergence behavior of the APF.

wait more than 1 minute to collect data for a PLI frequency estimate at the required level of resolution. This automatically rules out the use of these algorithms for real-time continuous monitoring: not only the FFT would be very demanding in terms of computation and storage, but also the PLI frequency can fluctuate significantly during the 1 minute interval, thereby fundamentally limiting the achievable accuracy.

In contrast, the APF methods can effectively track the variations in the PLI frequency. To demonstrate the fast convergence of this approach, two synthetic sequences with main frequencies of 50.5Hz and 51Hz were tested. In Figure 5.16 we can observe the frequency convergence of the APF RLS, where the frequency estimates converge to the actual fundamental frequency (i.e. 50.5Hz and 51Hz) in less than 400ms , while maintaining an SNR_{OUT} of 35db .

5.7.4 Evaluation on the Cerebro platform

To demonstrate the capabilities of our system and to evaluate the performance of the algorithms in a real-life application scenario, we collected ECG and EMG signals with the Cerebro platform. Data was collected from three subjects with acquisitions of up to 10 min. The signals were sampled at 1kHz with a gain of 32.

For the ECG signals we used a 3-lead acquisition setup, where one differential channel of Cerebro was used to acquire the signal. We placed two disposable electrodes on the wrists of the user and an additional electrode was placed on the right ankle as the reference potential. The EMG signal was acquired with one differential pair of electrodes placed on the forearm of the subjects and an additional electrode was placed on the elbow as the reference potential. Using the data acquired with Cerebro, we compared the filtering performance of

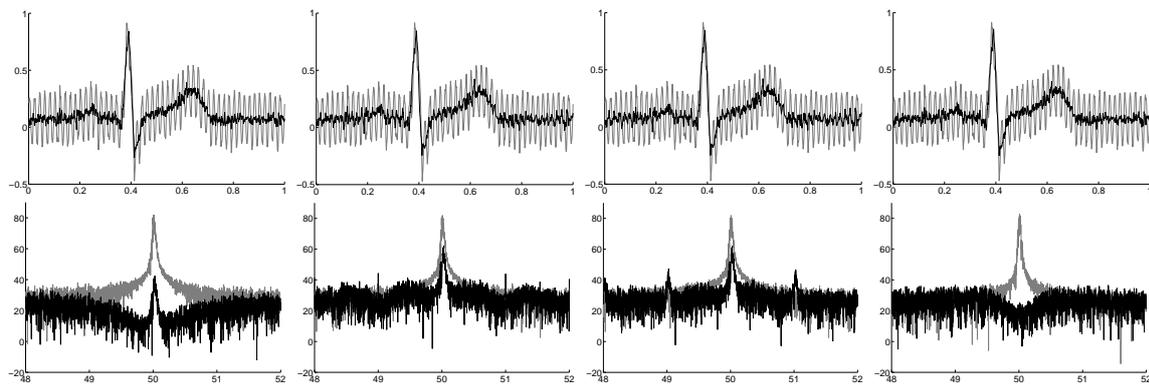


Fig. 5.17 PLI removal for ECG signals. Top row: raw and filtered signals (units: seconds and mV), bottom row: frequency spectrum (units: Hz and dB); from left to right: notch, SM, RS, APF.

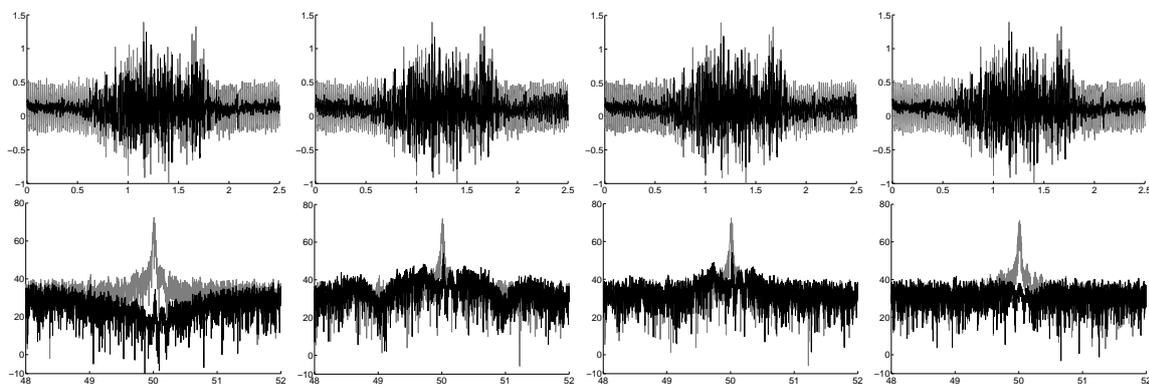


Fig. 5.18 PLI removal for EMG signals. Top row: raw and filtered signals (units: seconds and mV), bottom row: frequency spectrum (units: Hz and dB); from left to right: notch, SM, RS, APF.

the considered algorithms, evaluating the online versions of the RS and SM algorithms and the RLS version of the APF, which outperforms the LMS one.

The result of the PLI removal from the ECG and EMG signal is shown in Fig. 5.17 and 5.18, where we plotted the raw and the filtered signals along with their frequency spectrum. Ideally, the perfect filtering technique should remove the PLI component and leave the rest of the signal spectrum unchanged. The visual difference in the filtered signals is minimal, but from the spectrum plots we can note that the notch filter, RS online and SM online fail to adequately remove the interference. The notch filter removes also frequency components close the PLI, while the RS and SM present minimal alterations of the surrounding frequency spectrum, but they do not completely remove the PLI. The APF RLS delivers considerably better signal quality, indeed it is very precise to remove the PLI without additional changes in the signal's spectrum.

Table 5.3 Computational and memory requirements.

Algorithm	Computational Cost (ms)	Memory (byte)
Notch	0.002	48 byte
Sinusoidal Modeling	79	84040
Regression Subtraction	8.3	8160
APF RLS	0.036	184

Finally, we evaluated the computational cost of the algorithms. For the SM and RS approaches we calculated the computational cost to filter the signal in a window of N samples equal to 1500 (1.5s) and 1000 (1s) respectively. The notch and APF are iterative approaches applied at each new sample. Computational and memory costs are summarized in Table 5.3. Here we can note that the APF has the lowest computational cost, requiring only $36\mu s$ at each sample, making it a preferred choice for real-time implementation. The notch filter is also characterized by a low computational cost, but it introduces distortion in the signal, therefore it is not a good candidate to remove the PLI from biomedical signals. SM and RS are much more demanding with respectively 79ms and 8.3ms needed to process each window. Moreover, they have considerably higher memory requirements and they do not perform as well as the APF RLS in the continuous monitoring scenario characterized by frequency fluctuations of the PLI signal.

5.8 Conclusion

In this chapter, we presented two techniques to address two of the main issues of biosignal acquisition and processing. The DC offset cancellation is performed on AC coupled signals enabling the digital closed loop control of the analog front end for an combined HW/SW approach. The internal Cerebro DAC correctly reference the signal to acquire and eliminate the need for an additional reference electrode for each differential input. The MCU interface is responsible to estimate the offset and to dynamically control the AFE settings in a closed digital-to-analogue loop. The solution was tested with forearm EMG signals commonly used for hand gesture recognition and prosthesis control. We collected signals of up to 30 minutes from 4 healthy subjects while they were performing various everyday activities. From this analysis, we observed that the EMG signals exhibit a random initial offset, which changes in time during the acquisitions. Thus, we implemented an algorithm which periodically checks the offset of each channel only if no contraction is in progress and sets the Cerebro ADC to automatically compensate the estimated offset. The results show that our approach correctly removes and DC offset from the acquired signals and allows the development of

flexible and scalable wearable devices. Furthermore we evaluated PLI filtering techniques for a wearable device, targeting their on-board implementation and considering platforms with adequate processing capabilities to acquire high quality heterogeneous biopotential signals. In particular, we considered the acquisition of EMG and ECG signals and we described the implementation and performance of four main PLI removal methods. For quantitative and qualitative analysis of their characteristics, we performed extensive simulation using real ECG data and simulated interference with various signal conditions. The different approaches were also implemented and profiled on the embedded wearable system, evaluating them in terms of accuracy to reject the PLI and real-time performance. In particular, we demonstrate that the APF RLS takes only a few hundreds of milliseconds to autonomously tune its parameters and converge to the input PLI frequency, while the other approaches need several minutes for accurate frequency estimation. Furthermore, APF RLS has the best performance in terms of output SNR and it best preserves the frequency characteristics of the targeted signals, while having minimal computational cost.

Chapter 6

Conclusions

The work introduced in this dissertation focused on the analysis and the design at hardware and software level for an embedded system for hand gesture recognition. The multilevel design draws from supporting knowledge on both the machine and the human side, having the final target of an interface that is modular, flexible and not limited to a single use scenario. We started from the analysis of the SoA finding that the main open challenges in wearable hand gesture recognition are the difficulties to embed complex algorithms on a real-time resource-constrained platform and the reliability of the recognition. Hence we got insight into the sensor interfaces and the algorithms used for hand gesture recognition. We collected a robust dataset of EMG signal to analyze, characterize and validate our setup. From the hardware perspective, we started by developing an embedded system designed especially for the prosthetics employing state of the art solutions with advanced sensing and processing features. Then we adapted and modified the SVM algorithm to run on an embedded low power platform and we designed the whole firmware and software interface to control the artificial hand. The main goal of our implementation is the robustness of the classification, since this is the principal requirement of a prosthesis and the main limitation of the SoA in this field hence we developed a control strategy to improve the robustness of the recognition and we tested the system on healthy subjects, on amputees and also on data coming from a benchmark dataset. The outcome of this part of the research is the first embedded controller for prosthetics based on machine learning technique.

Continuing the evolution of the hardware and the software solutions, we target a more generic system for hand gesture recognition and biopotential acquisition, addressing flexibility and scalability. Exploiting the development of a dedicate high performance AFE, we investigated the sensor interface comparing the SoA commercial solution with a custom research solution and we designed a new hardware to have a multisensor wearable platform. We profiled the system to find the best tradeoff between number of sensors, sampling rates number of

recognized gestures and computational charge.

This solution was compared with the SoA of commercial HMI for hand gesture recognition, represented by an armband that streams the EMG data to a PC that runs a pattern recognition algorithm. We outperform this solution, having the same gesture recognition accuracy but with the advantage of a versatile system that runs the pattern recognition on board. In the last chapter of the dissertation we analyzed some optimization techniques to cope with some of the main problems of a biopotential wearable acquisition system, like the DC offset cancellation and the PLI removal. In fact the signal quality is the key element to perform an efficient signal processing and to gather information from the biosignals. The analog interface of a wearable system is prone to many causes of noise that can affect the signal quality, invalidate the whole signal conditioning chain. Firstly we presented the approach to remove the DC offset due to the floating ground potential of a wearable system without adding external circuitry to the system (like the DLR circuit used in ECG acquisition system). Finally we presented the main approaches to PLI removal considering the issues of a real time environment and showing the results of the filter implemented on our system.

References

- [1] BEBIONICS. <http://bebionics.com>.
- [2] I LIMB. <http://www.touchbionics.com>.
- [3] IEC-62304 International Standard on Medical device software Software life cycle processes . http://www.iso.org/iso/catalogue_detail.htm?csnumber=38421.
- [4] Michelangelo Hand. <http://http://www.ottobockus.com/prosthetics/upper-limb-prosthetics/solution-overview/michelangelo-prosthetic-hand/>.
- [5] MICHELANGELO HAND. <http://www.living-with-michelangelo.com/gb/home/>.
- [6] MYO ARM BAND. <http://www.thalmic.com/>.
- [7] Ottobock Sensor 13E200. <http://www.ottobock.com/>.
- [8] VIVIAGO. <http://vivago.com>.
- [9] Myoelectric control systems—a survey. *Biomedical Signal Processing and Control*, 2(4):275 – 294, 2007.
- [10] Michelangelo Hand. <http://www.ottobockus.com/prosthetics/upper-limb-prosthetics/>, 2011.
- [11] National Institute of Standards and Technology. http://www.nist.gov/tip/wp/pswp/upload/239_limb_prosthetics_services_devices.pdf, 2011.
- [12] M.R. Ahsan, M.I. Ibrahimy, and O.O. Khalifa. Electromyography (emg) signal based hand gesture recognition using artificial neural network (ann). In *Mechatronics (ICOM), 2011 4th International Conference On*, pages 1–6, May 2011.
- [13] Analog Devices. Low Power, Five Electrode Electrocardiogram (ECG) Analog Front End. *ADAS1000 datasheet*, 2012.
- [14] Alex Andrews, Evelyn Morin, and Linda McLean. Optimal electrode configurations for finger movement classification using emg. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 2987–2990. IEEE, 2009.

- [15] U. Anliker, J.A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E.B. Schenker, F. Catarisi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic. AMON: a wearable multiparameter medical monitoring and alert system. *Information Technology in Biomedicine, IEEE Transactions on*, 8(4):415–427, 2004.
- [16] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Elsig, A.-G. Mittaz Hager, O. Deriaz, C. Castellini, H. Muller, and B. Caputo. Characterization of a benchmark database for myoelectric movement classification. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 23(1):73–83, Jan 2015.
- [17] John V Basmajian and CJ De Luca. Muscles alive. *Muscles alive: their functions revealed by electromyography*, 278:126, 1985.
- [18] A Bazhyna, II Christov, A Gotchev, IK Daskalov, and K Egiazarian. Powerline interference suppression in high-resolution ECG. In *Computers in Cardiology, 2003*, pages 561–564. IEEE, 2003.
- [19] J.T. Belter, B.C. Reynolds, and A.M. Dollar. Grasp and force based taxonomy of split-hook prosthetic terminal devices. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 6613–6618, Aug 2014.
- [20] S. Benatti, B. Milosevic, F. Casamassima, P. Schonle, P. Bunjaku, S. Fateh, Q. Huang, and L. Benini. Emg-based hand gesture recognition with flexible analog front end. In *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*, pages 57–60, Oct 2014.
- [21] S. Benatti, B. Milosevic, M. Tomasini, E. Farella, P. Schönle, P. Bunjaku, G. Rovere, S Fateh, and L. Huang, Q.and Benini. Multiple biopotentials acquisition system for wearable applications. In *Biomedical Engineering Systems and Technologies Conference (BIOSTEC), 2015 INSTICC*, Jan 2015.
- [22] Simone Benatti, Filippo Casamassima, Bojan Milosevic, Elisabetta Farella, Philipp Schonle, Schekeb Fateh, Thomas Burger, Qiuting Huang, and Luca Benini. A versatile embedded platform for emg acquisition and gesture recognition. *Biomedical Circuits and Systems, IEEE Transactions on*, 9(5):620–630, 2015.
- [23] Simone Benatti, Elisabetta Farella, L Benini, and E Gruppioni. Analysis of robust implementation of an emg pattern recognition based control. *BIOSIGNALS 2014*, 2014.
- [24] Simone Benatti, Elisabetta Farella, Luca Benini, and Emanuele Gruppioni. Analysis of robust implementation of an EMG pattern recognition based control. In *Intl. Conf. on Bio-inspired Systems and Signal Processing, BIOSIGNALS*, pages 45–54, 2014.
- [25] Antonio Bicchi, Marco Gabiccini, and Marco Santello. Modelling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 366(1581):3153–3161, 2011.

- [26] William Biederman, Daniel J Yeager, Nathan Narevsky, Jaclyn Leverett, Ryan Neely, Jose M Carmena, Elad Alon, and Jan M Rabaey. A 4.78 mm² fully-integrated neuromodulation soc combining 64 acquisition channels with digital compression and simultaneous dual stimulation. 2015.
- [27] S. Bitzer and P. van der Smagt. Learning EMG control of a robotic hand: towards active prostheses. In *IEEE Intl. Conf. on Robotics and Automation, ICRA.*, 2006.
- [28] Jeremy Blum and HS Byram Hills. Using force sensors to effectively control a below-elbow intelligent prosthetic device, 2008.
- [29] J.L. Bohorquez, M. Yip, A.P. Chandrakasan, and J.L. Dawson. A Biomedical Sensor Interface With a Sinc Filter and Interference Cancellation. *IEEE Journal of Solid-State Circuits (JSSC)*, 46(4):746–756, April 2011.
- [30] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of the Workshop on Computational learning theory*, pages 144–152, 1992.
- [31] L. Brown, J. van de Molengraft, R.F. Yazicioglu, T. Torfs, J. Penders, and C. Van Hoof. A low-power, wireless, 8-channel EEG monitoring headset. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 4197–4200, Aug 2010.
- [32] D. Buxi, T. Berset, M. Hijdra, M. Tutelaers, D. Geng, J. Hulzink, M. van Noorloos, I. Romero, T. Torfs, and N. Van Helleputte. Wireless 3-lead ECG system with on-board digital signal processing for ambulatory monitoring. In *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE*, pages 308–311, Nov 2012.
- [33] G. Buzaki. *Rhythms of the Brain*. Oxford University press, London, 2nd edition, 2006.
- [34] THOMAS W Calvert and AE Chapman. The relationship between the surface emg and force transients in muscle: simulation and experimental studies. *Proceedings of the IEEE*, 65(5):682–689, 1977.
- [35] A.J. Casson, D. Yates, S. Smith, J.S. Duncan, and E. Rodriguez-Villegas. Wearable electroencephalography. *Engineering in Medicine and Biology Magazine, IEEE*, 29(3):44–56, May 2010.
- [36] Claudio Castellini, Emanuele Gruppioni, Angelo Davalli, and Giulio Sandini. Fine detection of grasp force and posture by amputees via surface electromyography. volume 103, pages 255–262, 2009.
- [37] Claudio Castellini and Patrick van der Smagt. Surface EMG in advanced hand prosthetics. *Biological Cybernetics*, 100(1):35–47, 2009.
- [38] M.G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi. Adaptive synergies for the design and control of the pisa/iit soft hand. *Int. J. Rob. Res.*, 33(5):768–782, April 2014.

- [39] Adrian DC Chan and Kevin B Englehart. Continuous myoelectric control for powered prostheses using hidden markov models. *Biomedical Engineering, IEEE Transactions on*, 52(1):121–124, 2005.
- [40] Xun Chen and J. Wang. Design and implementation of a wearable, wireless EEG recording system. In *Bioinformatics and Biomedical Engineering, (iCBBE) 2011 5th International Conference on*, pages 1–4, May 2011.
- [41] M.F. Chimene and Ramon Pallas-Areny. A comprehensive model for power line interference in biopotential measurements. *Instrumentation and Measurement, IEEE Transactions on*, 49(3):535–540, Jun 2000.
- [42] Ying-Han Chiou, Jer-Junn Luh, Shih-Ching Chen, Jin-Shin Lai, and Te-Son Kuo. The comparison of electromyographic pattern classifications with active and passive electrodes. *Medical engineering & physics*, 26(7):605–610, 2004.
- [43] N.I. Cho and S.U. Lee. On the adaptive lattice notch filter for the detection of sinusoids. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 40(7):405–416, Jul 1993.
- [44] Rubana H Chowdhury, Mamun BI Reaz, Mohd Alauddin Bin Mohd Ali, Ashrif AA Bakar, Kalaivani Chellappan, and Tae G Chang. Surface electromyography signal processing and classification techniques. *Sensors*, 13(9):12431–12466, 2013.
- [45] R. DeVaul, M. Sung, J. Gips, and A. Pentland. MITHril 2003: applications and architecture. In *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 4–11, Oct 2003.
- [46] Roger C Dugan, Mark F McGranaghan, Surya Santoso, and H Wayne Beaty. *Electrical power systems quality*. McGraw-Hill, 2003.
- [47] K. Englehart and B. Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, 50(7):848–854, July 2003.
- [48] Kevin Englehart, B Hudgin, and Philip A Parker. A wavelet-based continuous classification scheme for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, 48(3):302–311, 2001.
- [49] PEng Erik Scheme MSc and PEng Kevin Englehart PhD. Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. *Journal of rehabilitation research and development*, 48(6):643, 2011.
- [50] D. Farina, Ning Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O.C. Aszmann. The extraction of neural information from the surface emg for the control of upper-limb prostheses: Emerging avenues and challenges. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 22(4):797–809, July 2014.
- [51] G David Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

- [52] Marco Gabiccini, Antonio Bicchi, Domenico Prattichizzo, and Monica Malvezzi. On the role of hand synergies in the optimal choice of grasping forces. *Autonomous Robots*, 31(2-3):235–252, 2011.
- [53] B Giroux and M Lamontagne. Comparisons between surface electrodes and intramuscular wire electrodes in isometric and dynamic conditions. *Electromyography and clinical neurophysiology*, 30(7):397–405, 1990.
- [54] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, Jun 2000.
- [55] B. Gosselin, A.E. Ayoub, J.-F. Roy, M. Sawan, F. Lepore, A. Chaudhuri, and D. Guitton. A mixed-signal multichip neural recording interface with bandwidth reduction. *Biomedical Circuits and Systems, IEEE Transactions on*, 3(3):129–141, June 2009.
- [56] GTEC g.MOBILab. <http://www.gtec.at/Products/Hardware-and-Accessories/g.MOBILab-Specs-Features>.
- [57] M.A. Haberman and E.M. Spinelli. A multichannel eeg acquisition scheme based on single ended amplifiers and digital dnl. *Biomedical Circuits and Systems, IEEE Transactions on*, 6(6):614–618, Dec 2012.
- [58] Janne M Hahne, Felix Biebmann, N Jiang, H Rehbaum, Dario Farina, FC Meinecke, Klaus-Robert Muller, and LC Parra. Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 22(2):269–279, 2014.
- [59] L.J. Hargrove, K. Englehart, and B. Hudgins. A comparison of surface and intramuscular myoelectric signal classification. *Biomedical Engineering, IEEE Transactions on*, 54(5):847–853, May 2007.
- [60] L.J. Hargrove, K. Englehart, and B. Hudgins. A comparison of surface and intramuscular myoelectric signal classification. volume 54, pages 847–853, 2007.
- [61] Hanna Heger, S Millstein, and GA Hunter. Electrically powered prostheses for the adult with an upper limb amputation. *Journal of Bone & Joint Surgery, British Volume*, 67(2):278–281, 1985.
- [62] K. Hirano, S. Nishimura, and S.K. Mitra. Design of digital notch filters. *Communications, IEEE Transactions on*, 22(7):964–970, Jul 1974.
- [63] Ning Jiang, Kevin B Englehart, and Philip A Parker. Extracting simultaneous and proportional neural control information for multiple-dof prostheses from the surface electromyographic signal. *Biomedical Engineering, IEEE Transactions on*, 56(4):1070–1080, 2009.
- [64] Mohammad Reza Keshtkaran and Zhi Yang. A fast, robust algorithm for power line interference cancellation in neural recording. *Journal of Neural Engineering*, 11(2), 2014.

- [65] M.R. Keshtkaran and Zhi Yang. A robust adaptive power line interference canceler vlsi architecture and asic for multichannel biopotential recording applications. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 61(10):788–792, 2014.
- [66] T. Kinjo and K. Funaki. On hmm speech recognition based on complex speech analysis. In *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, pages 3477–3480, Nov 2006.
- [67] Jangwoo Kwon, Chulkyu Shin, EungHyuk Lee, Younghwan Han, and SeungHong Hong. Hybrid hmm-mlp classifier for prosthetic arm control purpose. In *TENCON '96. Proceedings., 1996 IEEE TENCON. Digital Signal Processing Applications*, volume 1, pages 21–24 vol.1, Nov 1996.
- [68] Chavdar Levkov, Georgy Mihov, Ratcho Ivanov, Ivan Daskalov, Ivaylo Christov, and Ivan Dotsinsky. Removal of power-line interference from the ECG: a review of the subtraction procedure. *BioMedical Engineering OnLine*, 4(1):50, 2005.
- [69] libSVM. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2011.
- [70] Jun Liu, Fan Zhang, and H.H. Huang. An open and configurable embedded system for emg pattern recognition implementation for artificial arms. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 4095–4098, Aug 2014.
- [71] C.M. Lopez, D. Prodanov, D. Braeken, I. Gligorijevic, W. Eberle, C. Bartic, R. Puers, and G. Gielen. A multichannel integrated circuit for electrical recording of neural activity, with independent channel programmability. *IEEE Trans. on Biomedical Circuits and Systems*, 6(2):101–110, April 2012.
- [72] S.M.M. Martens, M. Mischi, S.G. Oei, and J.W.M. Bergmans. An improved adaptive power line interference canceller for electrocardiography. *Biomedical Engineering, IEEE Transactions on*, 53(11):2220–2231, Nov 2006.
- [73] Bruno Massa, Stefano Roccella, Maria Chiara Carrozza, and Paolo Dario. Design and development of an underactuated prosthetic hand. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3374–3379. IEEE, 2002.
- [74] Christopher D. McManus, Karl-Dietrich Neubert, and Eckehart Cramer. Characterization and elimination of AC noise in electrocardiograms: A comparison of digital filtering methods. *Computers and Biomedical Research*, 26(1):48 – 67, 1993.
- [75] A.C. Metting van Rijn, A. Peper, and C.A. Grimbergen. High-quality recording of bioelectric events. *Medical and Biological Engineering and Computing*, 28(5):389–397, 1990.
- [76] Bojan Milosevic, Elisabetta Farella, and Luca Benini. Continuous gesture recognition for resource constrained smart objects. In *UBICOMM 2010, The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 391–396, 2010.

- [77] R. Moore and J. Lopes. IFCN standards for digital recording of clinical EEG. the international federation of clinical neurophysiology. In *Electroencephalogr Clin Neurophysiol Suppl.* PubMed, 1998.
- [78] R. Muller, S. Gambini, and J.M. Rabaey. A 0.013 mm², 5 μW, DC-Coupled Neural Signal Acquisition IC With 0.5 V Supply. *IEEE Journal of Solid-State Circuits (JSSC)*, 47(1):232–243, Jan 2012.
- [79] C.W. Mundt, K.N. Montgomery, U.E. Udoh, V.N. Barker, G.C. Thonier, A.M. Tellier, R.D. Ricks, B.B. Darling, Y.D. Cagle, N.A. Cabrol, S.J. Ruoss, J.L. Swain, J.W. Hines, and G.T.A. Kovacs. A multiparameter wearable physiologic monitoring system for space and terrestrial applications. *Information Technology in Biomedicine, IEEE Transactions on*, 9(3):382–391, Sept 2005.
- [80] P. Nunez and R. Srinivasan. *Electric fields of the brain: the neurophysics of EEG.* Oxford University Press, London, 2nd edition, 2006.
- [81] Omron R7. <http://www.omron-healthcare.com/eu/en/our-products/blood-pressure-monitoring/r7>, 2004.
- [82] Mohammadreza Asghari Oskoei, Student Member, and Huosheng Hu. Support vector machine-based classification scheme for myoelectric control applied to upper limb.
- [83] Alexandros Pantelopoulos and Nikolaos G Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Trans. on Systems, Man, and Cybernetics*, 40(1):1–12, 2010.
- [84] Soo-Chang Pei and Chien-Cheng Tseng. Elimination of ac interference in electrocardiogram using iir notch filter with transient suppression. *Biomedical Engineering, IEEE Transactions on*, 42(11):1128–1132, Nov 1995.
- [85] J Penders, J van de Molengraft, M Altini, F Yazicioglu, and C Van Hoof. A low-power wireless ECG necklace for reliable cardiac activity monitoring on-the-move. *Proc. of the Intl. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2011.
- [86] Philips MX40. <http://www.usa.philips.com/healthcare-products/HC862115/intellivue-mx40-wearable-patient-monitor>, 2011.
- [87] A Phinyomark, C Limsakul, and P Phukpattaranont. Application of wavelet analysis in emg feature extraction for pattern classification. *Measurement Science Review*, 11(2):45–52, 2011.
- [88] J. Piskorowski. Digital q -varying notch iir filter with transient suppression. *Instrumentation and Measurement, IEEE Transactions on*, 59(4):866–872, April 2010.
- [89] F. Riillo, L.R. Quitadamo, F. Cavrini, E. Gruppioni, C.A. Pinto, N. Cosimo Pastò, L. Sbernini, L. Albero, and G. Saggio. Optimization of emg-based hand gesture recognition: Supervised vs. unsupervised data preprocessing on healthy subjects and transradial amputees. *Biomedical Signal Processing and Control*, 14(0):117 – 125, 2014.

- [90] Lait Abu Saleh, Wjatscheslaw Galjan, Jakob M. Tomasik, Dietmar Schroeder, and Wolfgang H. Krautschneider. A 130nm asic for emg signal acquisition to control a hand prosthetic. pages 149–153. SciTePress, 2012.
- [91] T. Scott Saponas, Desney S. Tan, Dan Morris, Jim Turner, and James A. Landay. Making muscle-computer interfaces more practical. In *CHI*, pages 851–854. ACM, 2010.
- [92] P. Schonle, F. Schulthess, S. Fateh, R. Ulrich, F. Huang, T. Burger, and Q. Huang. A DC-connectable multi-channel biomedical data acquisition ASIC with mains frequency cancellation. In *ESSCIRC (ESSCIRC), 2013 Proceedings of the*, pages 149–152, Sept 2013.
- [93] P. Schonle, F. Schulthess, S. Fateh, R. Ulrich, F. Huang, T. Burger, and Q. Huang. A dc-connectable multi-channel biomedical data acquisition asic with mains frequency cancellation. In *ESSCIRC (ESSCIRC), 2013 Proceedings of the*, pages 149–152, Sept 2013.
- [94] Philipp Schönle, Felix Schulthess, Schekeb Fateh, Roger Ulrich, Fiona Huang, Thomas Burger, and Qiuting Huang. A DC-connectable multi-channel biomedical data acquisition ASIC with mains frequency cancellation. In *ESSCIRC*, pages 149–152, 2013.
- [95] Philipp Schönle, Felix Schulthess, Schekeb Fateh, Roger Ulrich, Fiona Huang, Thomas Burger, and Qiuting Huang. A DC-connectable multi-channel biomedical data acquisition ASIC with mains frequency cancellation. In *ESSCIRC*, pages 149–152, 2013.
- [96] et al. Serrano, Roberto E. Power line interference in ambulatory biopotential measurements. *Proceedings of the 25th Annual International Conference of the IEEE*, 4, May 2003.
- [97] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [98] P. Shenoy, K.J. Miller, B. Crawford, and R. P N Rao. Online electromyographic control of a robotic prosthesis. *Biomedical Engineering, IEEE Transactions on*, 55:1128–1135, 2008.
- [99] S. Sudarsan and E. Chandra Sekaran. Design and development of {EMG} controlled prosthetics limb. *Procedia Engineering*, 38:3547 – 3551, 2012. International Conference On Modelling Optimization And Computing.
- [100] T.J. Sullivan, S.R. Deiss, Tzyy-Ping Jung, and G. Cauwenberghs. A brain-machine interface using dry-contact, low-noise EEG sensors. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, May 2008.
- [101] Shin-Lian Teng, R. Rieger, and Yu-Bin Lin. Programmable ExG Biopotential Front-End IC for Wearable Applications. *IEEE Trans. on Biomedical Circuits and Systems*, 8(4):543–551, Aug 2014.

- [102] Texas Instruments. Low-Power, 8-Channel, 24-bit Analog Front-End for Biopotential Measurements. *ADS1298 datasheet*, 2012.
- [103] N. Van Helleputte, M. Konijnenburg, J. Pettine, D.-W. Jee, H. Kim, A. Morgado, R. Van Wegberg, T. Torfs, R. Mohan, A. Breeschoten, H. de Groot, C. Van Hoof, and R.F. Yazicioglu. A 345 μ W Multi-Sensor Biomedical SoC With Bio-Impedance, 3-Channel ECG, Motion Artifact Reduction, and Integrated DSP. *IEEE Journal of Solid-State Circuits (JSSC)*, 50(1):230–244, Jan 2015.
- [104] Nick Van Helleputte, Mario Konijnenburg, Julia Pettine, D-W Jee, Hyejung Kim, Alonso Morgado, Roland Van Wegberg, Tom Torfs, Rachit Mohan, Arjan Breeschoten, et al. A 345 μ W multi-sensor biomedical soc with bio-impedance, 3-channel eeg, motion artifact reduction, and integrated dsp. 2015.
- [105] Paul Ventimiglia. *Design of a human hand prosthesis*. PhD thesis, Worcester Polytechnic Institute, 2012.
- [106] WHF . <http://www.world-heart-federation.org/cardiovascular-health/global-facts-map/>, 2012.
- [107] WHO. http://www.who.int/mental_health/neurology/neurological_disorders_report_web.pdf, 2012.
- [108] B. Widrow, Jr. Glover, J.R., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, Jr. Eugene Dong, and R.C. Goodlin. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692–1716, Dec 1975.
- [109] Jiawei Xu, S. Mitra, A. Matsumoto, S. Patki, C. Van Hoof, K.A.A. Makinwa, and R.F. Yazicioglu. A Wearable 8-Channel Active-Electrode EEG/ETI Acquisition System for Body Area Networks. *IEEE Journal of Solid-State Circuits (JSSC)*, 49(9):2005–2016, Sept 2014.
- [110] Jiawei Xu, R.F. Yazicioglu, B. Grundlehner, P. Harpe, K.A.A. Makinwa, and C. Van Hoof. A 160 μ W 8-channel active electrode system for eeg monitoring. *Biomedical Circuits and Systems, IEEE Transactions on*, 5(6):555–567, Dec 2011.
- [111] Shumpei Yamakawa and Takuya Nojima. A proposal for a mmg-based hand gesture recognition method. In *Adjunct Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST Adjunct Proceedings '12, pages 89–90, New York, NY, USA, 2012. ACM.
- [112] R.F. Yazicioglu, P. Merken, R. Puers, and C. Van Hoof. A 200 μ W eight-channel EEG acquisition ASIC for ambulatory EEG systems. *Solid-State Circuits, IEEE Journal of*, 43(12):3025–3038, Dec 2008.
- [113] R.F. Yazicioglu, P. Merken, R. Puers, and C. Van Hoof. A 200 μ W Eight-Channel EEG Acquisition ASIC for Ambulatory EEG Systems. *IEEE Journal of Solid-State Circuits (JSSC)*, 43(12):3025–3038, Dec 2008.
- [114] H.J. Yoo and C. van Hoof. *Bio-Medical CMOS ICs*. Integrated Circuits and Systems. Springer, 2010.

- [115] Aaron J Young, Levi J Hargrove, and Todd A Kuiken. Improving myoelectric pattern recognition robustness to electrode shift by changing interelectrode distance and electrode configuration. *Biomedical Engineering, IEEE Transactions on*, 59(3):645–652, 2012.
- [116] Haoshi Zhang, Yaonan Zhao, Fuan Yao, Lisheng Xu, Peng Shang, and Guanglin Li. An adaptation strategy of using lda classifier for emg pattern recognition. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 4267–4270, July 2013.
- [117] Xiaorong Zhang, He Huang, and Qing Yang. Real-time implementation of a self-recovery emg pattern recognition interface for artificial arms. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 5926–5929, July 2013.
- [118] Xu Zhang, Xiang Chen, Zhang-yan Zhao, You-qiang Tu, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. Research on gesture definition and electrode placement in pattern recognition of hand gesture action semg. In *Proceedings of the 1st International Conference on Medical Biometrics, ICMB'08*, pages 33–40, Berlin, Heidelberg, 2007. Springer-Verlag.
- [119] A.K. Ziarani and A. Konrad. A nonlinear adaptive method of elimination of power line interference in ecg signals. *Biomedical Engineering, IEEE Transactions on*, 49(6):540–547, June 2002.
- [120] Miroslav Zivanovic and Miriam González-Izal. Simultaneous powerline interference and baseline wander removal from ECG and EMG signals by sinusoidal modeling. *Medical engineering & physics*, 35(10):1431–1441, 2013.
- [121] Miroslav Zivanovic and Miriam González-Izal. Simultaneous powerline interference and baseline wander removal from ECG and EMG signals by sinusoidal modeling. *Medical Engineering & Physics*, 35(10):1431 – 1441, 2013.
- [122] Loredana Zollo, Stefano Roccella, Eugenio Guglielmelli, M Chiara Carrozza, and Paolo Dario. Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. *Mechatronics, IEEE/ASME Transactions On*, 12(4):418–429, 2007.