UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
ENGENHARIA DE COMPUTAÇÃO

ÂNGELO CARDOSO LAPOLLI

Low-Power Hardware Design for Neural Spike Detection and Compression within Invasive Brain Machine Interface Systems

Final Report presented in partial fulfillment of the requirements for the degree of Computer Engineer

Prof. Dr. Renato Perez Ribas
Advisor

Porto Alegre, June 2014

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitora de Graduação: Prof. Sérgio Roberto Kieling
Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb
Coordenador do curso: Prof. Marcelo Götz
Bibliotecário-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“Start where you are.
Use what you have.
Do what you can.”
Arthur Ashe

ACKNOWLEDGEMENTS

I would like to offer my special thanks to CEA Grenoble LETI group for supporting most of this work research. In particular, I am thankful to Prof. Dr. Rodolphe Héliot for his inspiration and for guiding me throughout this project.

I am grateful to UFRGS professors and staff for providing academic excellence within international standards. Notably, I wish to express my appreciation to Prof. Dr. Renato Perez Ribas for his assistance and valuable advice.

I would like to specially thank my brother and friends for contributing with a comprehensive an cheerful environment. Above all, I am greatly thankful to my parents, Aldo and Rosa, for their endless love and life guidance.

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

ADC	*Analog-to-Digital Converter*
ASIC	*Application-Specific Integrated Circuit*
AV	*Absolute Value*
BMI	*Brain-Machine Interface*
CMOS	*Complementary Metal-Oxide-Semiconductor*
CNS	*Central Nervous System*
CS	*Compressed Sensing*
DD	*Discrete Derivatives*
DWT	*Discrete Wavelet Transform*
IC	*Integrated Circuit*
iEEG	*Intra-Cortical Electroencephalography*
MEA	*Micro-Electrode Array*
MST	*Minimum Spanning Tree*
NEO	*Nonlinear Energy Operator*
PCA	*Principal Component Analysis*
PNS	*Peripheral Nervous System*
RIP	*Restricted Isometry Property*
RTL	*Register-Transfer Level*
SNR	*Signal-to-Noise Ratio*
SPC	*Super-Paramagnetic Clustering*
VHDL	*VHSIC Hardware Description Language*
VHSIC	*Very High Speed Integrated Circuit*
WCSS	*Within-Cluster Sum of Squares*

LIST OF FIGURES

ABSTRACT

Invasive brain-machine interface systems benefit from intra-cortically implanted microelectrode arrays for neural signal recording with improved signal-to-noise ratio and spatial resolution. Given the fragility of the biological inner tissues close to these sensors, most of this data processing is delegated to an external unit looking to avoid hazardous heating from electric energy dissipation. In those conditions, multiple-cells activity monitoring generates great amount of data which is to be wirelessly transmitted in a way to avoid cables passing through the skull. As this transmission at the incoming data rate presents dangerous power levels, data compression approaches must be employed to guarantee safety by reducing the bandwidth which directly affects energy consumption.

With that in mind, this work proposes digital hardware implementations for neural spike detection and compression using compressed sensing. The proposed method is validated considering neural data processing performance and the respect of the constraints imposed by the related physiological context. The final hardware specifications for a CMOS 0.18 μm fabrication process indicate a power of approximately 500 nW within an area of 0.07 mm^2 for the processing of a single recording channel. The solution provides an estimated 96.5% data rate reduction which is shown to be greater than the achieved by related work.

Keywords: Brain machine interface, spike sorting, neural data compression, compressed sensing.

Circuito de Baixa Potência para Detecção e Compressão de Impulsos Nervosos em Sistemas de Interface Cérebro Máquina

RESUMO

Sistemas invasivos de interface cérebro máquina beneficiam-se de matrizes de microeletrodos implantadas intra-corticalmente para a captura de sinais nervosos com melhor relação sinal-ruído e resolução espacial. Dada a fragilidade dos tecidos biológicos internos próximos a esses sensores, grande parte do processamento desses dados é delegada a uma unidade externa procurando evitar um aquecimento danoso pela dissipação de energia elétrica. Nessas condições, o monitoramento da atividade de múltiplas células gera grande quantidade de dados a ser transmitida via comunicação sem fio de forma a evitar cabos passando através do crânio. Visto que essa transmissão na taxa dos dados de entrada apresenta níveis de potência perigosos, abordagens de compressão de dados precisam ser utilizadas para garantir a segurança pela redução da largura de banda a qual afeta diretamente o consumo de energia.

Por esse motivo, esse trabalho propõe implementações digitais em hardware para detecção de impulsos nervosos e compressão via amostragem compressiva. O método proposto é validado considerando o desempenho do processamento de dados nervosos e o respeito das restrições impostas pelo contexto fisiológico relacionado. As especificações finais de hardware para um processo de fabricação CMOS de 0.18 μm indicam uma potência de aproximadamente 500 nW em uma área de 0.07 mm^2 para o processamento de um único canal de captura. A solução proporciona uma redução da taxa de dados estimada em 96.5%, a qual é mostrada ser superior às obtidas por trabalhos relacionados.

Palavras-chave: interface cérebro máquina, *spike sorting*, compressão de dados nervosos, amostragem compressiva.

1 INTRODUCTION

Historically, the integrated circuit (IC) industry has mainly focused its efforts on the improvement of computational performance. More recently, low-power driven design has emerged as a major concern given the increasing interest in portable applications. The technological advances in this sense have stimulated the development of wearable and implantable electronic devices with major applications in entertainment, sports, and health care. Specifically, in biomedical engineering, they can be used for physiological signals monitoring [3], and for the restoration of sensorimotor functions within brain-machine interface (BMI) systems [4] [5].

1.1 Brain-Machine Interface Systems

A BMI is a direct communication link between the nervous system and a device. Typically, BMI systems are used for medical purposes where neural data is recorded and decoded in order to control an actuator such as wheel chairs [4] or prosthetic limbs [5]. In invasive BMI systems, data recording is done by means of intra-cranial electroencephalography (iEEG) [6]; in turn, the decoding consists in classifying neural spikes according to their source [7].

Neural information is transmitted throughout the body by low energy electrical signals. Therefore, its capture from externally placed sensors results in restricted signal-to-noise ratios (SNRs) by virtue of the attenuation caused by the multiple layers of biologic tissues in between [6]. With that in mind, cortically implanted micro-electrode arrays (MEAs) are used to achieve better SNR and spatial resolution in a procedure known as iEEG. Beyond that, the collected data is wireless transmitted to the exterior to avoid cables passing through the skull. That approach, however, impose stringent power density limitations to the implanted circuitry due to its proximity to the more fragile inner cells [8]. For that reason, compression methods are usually used to reduce wireless transmission rate and, consequently, energy dissipation [9].

Once the neural signal is recorded, it must be interpreted looking for relevant information relative to the control of the actuator. This data is encoded by characteristic electrical waveforms called neuronal spikes or action potentials. The detection of voluntary actions by their association with neuronal spike firing patterns is used to command the actuator [7]. An important step of this process is known as spike sorting, it consists in detecting the presence of action potentials and in grouping them by their source neuron. Given that each neuron generates a characteristic spike shape, they are individually extracted from the signal received from the sensors and then a clustering algorithm is applied to this data set.

In short, a BMI system is composed by sensors, and hardware for communication and data processing. With the current sensor technology, good quality signal recording and usability require cortically implanted MEAs along with a wireless transmission module. Being so, eventual data compression and spike sorting is distributed among the implanted nodes and an external processing unit which is directly connected to the actuator.

1.2 Motivation

The human nervous system coordinates the physiological activities, conveys sensory information throughout the body, and issues orders for muscular movements [10]. Since all these functions are done by means of the propagation of neural spikes, BMI systems as described above have a very diverse range of possible medical applications. Moreover, the challenge of meeting the constraints imposed on circuits by this kind of system stimulates the study of IC design techniques.

The association of mechanical prosthesis with a BMI can bring hope to amputees in restoring their sensorimotor functions; in such a way, it improves their quality of life and increases their autonomy. Additionally, neural data recording and processing can be done with the intention of monitoring neurological disorders such as epilepsy [3] being, in that case, capable of predicting and preventing seizures. In the light of neuroscience research, the access to this data is valuable for modeling human neural computation.

When considering IC design, even with the recent accelerated rate of advances in the fabrication process, the fulfillment of the strong power and area constraints of the cortically implanted MEAs and associated hardware is still a challenge to overcome. Researchers have been lately proposing ideas to address this problem [11] [12], but no definitive solution has yet been found. Under those circumstances, this research topic is open to new discoveries involving not only BMIs in specific, but also novel low-power and low-area circuit design techniques.

1.3 Objective

Based on a previous study on the feasibility of the use of compressed sensing (CS) [13] in BMIs [14], the purpose of this work is to propose a solution for the compression of neural data using this method. The idea is to design a digital application-specific integrated circuit (ASIC) performing spike detection, extraction, and compression. Within this context, different design techniques are to be studied towards meeting the hardware design constraints. Simulated data is to be used for modeling the input signal emulating recording, amplification, filtering, and analog-to-digital conversion. The evaluation of the proposed processing approach will consider the output data rate reduction besides the spike detection and the potential classification accuracy.

1.4 Structure of the Report

The first part of this report introduces the physiological aspects relevant to the development of the circuit, it covers the general structure of the nervous system and the characteristics of neural signals. After that, state-of-the-art techniques for spike sorting and the CS compression method are presented along with related hardware solutions. In a third moment, mathematical modeling and simulation serve to optimize parameters as to reduce hardware power and area maintaining satisfactory performance levels. The final hardware architecture and its design methodology are then described in the following section. Afterwards, the results of logic synthesis and physical layout are shown and discussed in comparison with the existing similar solutions. Finally, the conclusions taken from this work are presented with considerations referring to future works.

2 PHYSIOLOGICAL BACKGROUND

The development of any system involving the processing of biological signals must consider the related physiological context to ensure safety and adequate performance. In the first part of this section, the general structure of the human nervous system is described; the second part then approaches the generation and the characteristics of neural data.

2.1 The Nervous System Structure

The human nervous system conveys and processes sensory information to coordinate body activities, it monitors external and internal environmental conditions in order to react preserving proper organism functioning. This system comprises neurons and supporting cells that constitute the brain, the spinal cord, and nerves through which neural data is propagated [10]. The brain, in particular, integrates most of this data.

The input to the nervous system occurs in specialized sensory receptors that encode physical information from the surroundings resulting in internally comprehensible electrochemical signals. These are then propagated to processing centers which interpret them and issue commands to muscles and visceral glands by the same mean. This process creates the perception of senses such as sight, hearing, taste, smell, and touch; in addition, it allows, for example, the coordination of breathing, heart beating, and body movement [10].

The processing centers mentioned above constitute the central nervous system (CNS), it includes the brain and the spinal cord. The communication paths linking sensory receptors and effectors to the CNS compose the peripheral nervous system (PNS) which comprises the nerves. The elementary signals propagated through this network are called spikes or action potentials, they are generated within neurons and represent sensorimotor information [10].

Within this structure, the brain is the main processing unit. Excepting basic reflexive responses, this organ is major coordinator of all other nervous system functions. This is where neural information is integrated and motor control commands are originated, it is hence the richest source of data in the communication with the nervous system [10]. Nevertheless, this essential organ is protected by structures such as the skull and the cerebrospinal fluid which attenuate neural signals when performing external recordings.

The importance of the nervous system functions justifies the medical interest in the communication with this system. Although its extension over the whole body, the integration of its processing activity on the brain makes of this organ a centralized source of neural data. Still, the brain protection structures significantly affect external signal recordings SNRs and spatial resolutions; on the other hand, when considering iEEG, care must be taken not to damage internal cells [8].

2.2 Neural Data

Neural data refers to all the information propagated throughout the nervous system pathways. These pathways are mainly formed by specially structured cells called neurons which generate the signals that represent this data. This generation involves electrochemical processes that define the characteristics of these signals and, consequently, impact on the design of BMI systems.

The representation of neural data is in the form of electrical impulses called neuronal spikes or action potentials, spike firing is name given to the neuronal action of emitting such signals. Spike firing patterns within sets of neurons are associated with specific sensorimotor information related to the desired BMI application [7]. For that purpose, action potentials are detected and classified according to their source neuron; the analysis of the characteristics of these signals are hence important to define the methods for performing such tasks.

In the interest of describing the generation process of spikes and, in consequence, its characteristics, the anatomy of neurons must be taken into account. As illustrated by Figure 2.1, an neuron is composed of a cell body (soma), several branch projections known as dendrites, and a single nerve called axon ending on presynaptic terminals. The cell body is the metabolic center of the cell and it contains genetic information, dendrites are the responsible for the incoming of neural data whereas the axon conducts messages to the other neurons through the presynaptic terminals [10]. The specific structure of each neuron is specialized according to the type of information it transmits, that differentiation affects its spike shapes allowing their identification.

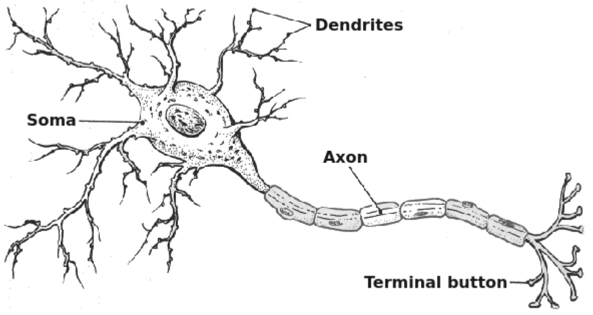


Figure 2.1: Neuron anatomy [1].

Briefly, neural spikes are a sequence of potential variation across the membrane of a neuron due to changes in its ionic permeability triggered by chemical substances called neurotransmitters. When not firing, this difference of electrical charge is of approximately -70 mV being called resting potential. By the action of neurotransmitters coming from adjacent neurons, the permeability of the membrane changes causing ions to move increasing this potential. If this value reaches a certain threshold of excitation around -60 mV, the potential quickly increases up to about 40 mV in which is called depolarization phase; it is followed by a repolarization when the difference of electrical charge rapidly decreases to the resting potential eventually undershooting it [15]. The resulting shape of a neural spike is illustrated by the Figure 2.2. This process provokes as well the release of neurotransmitters which can result in the firing of other neurons [15].

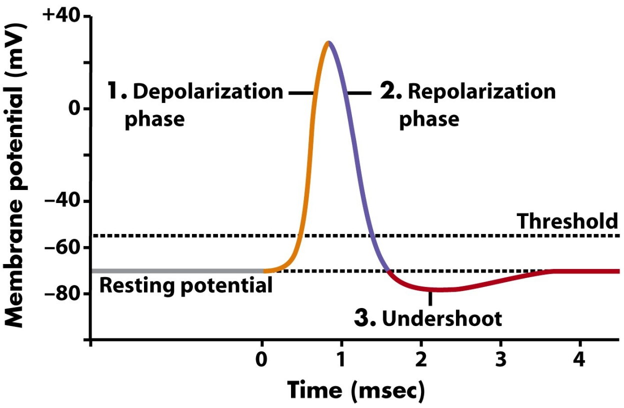


Figure 2.2: Neural spike phases [2].

In summary, the decoding of neural data needed in BMIs involves the detection and the source identification of neural spikes. The fact that these signals have a characteristic shape in consequence of their electrochemical generation process within neurons must be explored in the interest of detecting them. Considering the identification of their source neuron, the differences in their shape resulted from the structural variation of the cells must be explored.

3 RELATED WORK

The development of neural data processing solutions is currently an active research topic due to the challenge of combining high performance and low calculation complexity. Thus, the purpose of this section is to present the most successful approaches on that direction. The first subsection is dedicated to spike sorting whereas the second presents the CS compression method which have inspired this work. Finally, related hardware solutions are described.

3.1 Spike Sorting

The first neuroscience researches relied on the analysis of a single neuron activity. With the advances in sensors and in computational processing performance, present-day recording techniques allow simultaneous monitoring of multiple cells [7]. For that purpose, spike sorting is performed, *i.e.*, the identification and the classification of neuronal spikes from the neural signal received from sensors. This processing can be subdivided into four steps: raw data treatment, spike detection, feature extraction, and spike clustering. Some state-of-the-art techniques for each one is individually discussed hereinafter. Although the circuits to be proposed only include the spike detection among these steps, the definition of a complete spike sorting procedure is important to be able to evaluate the potential performance of the solution.

3.1.1 Raw Data Treatment

Raw data incoming from micro-electrodes is a low-amplitude (in the order of μV) noisy electrical signal, amplification and filtering are hence necessary before any other processing. After that, the signal can be digitalized if required by the nature of the subsequent circuitry.

Amplifiers used in intra-cortical neural recordings are specially designed circuits looking to respect power, area, and noise specifications. The main challenge of their design is to combine low input-referred noise and minimum energy consumption [16]. Following amplification, filtering is performed as to suppress low frequency offsets and to attenuate high frequency noisy appearance. For that end, a band pass filter is typically applied between 300 Hz and 3 kHz [17]. On this sense, in light of avoiding distortions affecting the spike shapes, the Butterworth filter type is usually used considering its flat frequency response within the passband.

The resulting signal from amplification and filtering must be sampled and quantized in the case of a digital data processing approach. Experimental observations of neuronal signals indicate significant frequency content up to about 8 kHz [18]. In consequence, according to Nyquist-Shannon sampling theorem, a minimum sampling frequency of 16 kHz is recommended; in practice, this value typically ranges from 20 kHz to 30 kHz [19]. In view of quantization, uniform quantizers are generally used with resolutions varying from 6 to 24 bits. However, some studies such as [20] show that the use of non-uniform approaches may reduce these values by lowering quantization noise. [20] suggests the use of an optimal quantizer assuming a normal probability distribution function of the signal.

Even though raw data treatment is usually neglected on neural data processing, it can significantly affect the outcome of spike sorting. Whereas appropriate amplification and filtering can enhance data acquisition SNR, the parameters of the digitalization may improve spike sorting results.

3.1.2 Spike Detection

From the resulting signal of raw data treatment, the second step of spike sorting aims at detecting and extracting action potentials. The detection generally involves the analysis of local amplitude and frequency information having as the most usual approaches for hardware implementation the Absolute Value (AV) and the Nonlinear Energy Operator (NEO) methods [21]. In turn, the extraction consists on the definition of a data window to represent individual spikes.

The AV technique is a simple detection method based on setting a threshold to the amplitude of the waveform. This value Thr is defined in [22] from an estimate of the standard deviation of the background noise $\tilde{\sigma}_N$ according to:

$$Thr = 4\tilde{\sigma}_N \text{ with } \tilde{\sigma}_N = median\left\{\frac{|x|}{0.6745}\right\}, \tag{3.1}$$

where x represents the whole input signal. The spike detection occurs whenever the value of the signal is greater than Thr. Previous work on the evaluation of this method [21] indicates that although it presents low calculation complexity, its detection accuracy degrades significantly with the decrease of SNRs.

In contrast with the AV method, the NEO approach presents better results with lower SNRs [21]. This technique is based on the definition of the NEO operator Ψ [23] given by:

$$\Psi[x(n)] = x^2(n) - x(n+1)\cdot x(n-1), \tag{3.2}$$

this value is large whenever the signal is both high in power and high in frequency which characterizes the presence of a spike. This operator is applied for each sample of the input signal and, similarly to the AV method, the result is compared to a threshold defined as:

$$Thr = C \cdot \frac{1}{N} \cdot \sum_{n=1}^{N} \Psi[x(n)], \tag{3.3}$$

where C is a scale factor and N is a predetermined number of samples. These parameters must be chosen experimentally as to optimize the detection being C generally set to 8 [21].

Once the presence of a neuronal spike is identified, it is extracted from the signal represented by a data window of preset length. This value is determined based on the estimated duration of the action potential waveform which, for mammalians, is generally between 1 ms and 1.2 ms [19]. In addition to that, spike alignment is required, *i.e.*, the definition of the position of this window relative to the input signal. This is normally set in regard to the local maximum amplitude close to the detection instant. For instance, in [22], 64 samples at 24 kHz are extracted for each action potential, then its maximum amplitude is determined from a cubic spline interpolation and aligned to the twentieth position. Differently, [14] considers 32 samples recorded at the same frequency and the maximum amplitude among these values is put at the twelfth. In any case, the output of this processing step is a set of independent data vectors representing individual spikes.

Spike detection is a crucial step within spike sorting since being responsible for identifying and extracting the elementary neural data element. Effectively, the number of successfully classified spikes at the end of the processing is limited to the number of correct detections; besides, the presence of false positives may impact on the distinction of classes. Beyond all that, this step also represents a reduction on the volume of data which can be used to improve hardware energy efficiency similarly to a compression approach.

3.1.3 Feature Extraction

The objective of the third step of spike sorting is to reduce the representation of the detected action potentials to the essential information required to differentiate them by their source neuron. With that intention, most solutions use Principal Component Analysis (PCA) whereas some authors defend the use of the Discrete Derivatives (DD) [21] or the Discrete Wavelet Transform (DWT) [22].

PCA is a statistical data analysis method which transforms a set of observations into a space where its components are in a decreasing order of variance. Assuming that the components with the highest variance values compile the information needed for the classification of the spikes, some feature extraction techniques reduce the original data set to the first two or three components of its PCA representation.

The DD is a simple method proposed in [24] which compiles a spike representation s into its slopes at different data points with configurable δ time-scales as in:

$$dd_{\delta}(n) = s(n) - s(n - \delta). \tag{3.4}$$

In [21], δ is set to 1, 3, and 7; it concludes that this method results in high classification accuracy for low computational complexity.

In contrast with the Discrete Fourier Transform in which a signal described in time is decomposed into sinusoidal functions defined exclusively in the frequency domain, the DWT uses functions defined in both spaces. As a consequence, the transform holds information about these two domains with adjustable resolutions. In [22], a four-level decomposition using Haar wavelets is performed for spike sorting purposes, it shows better classification results in comparison with the PCA approach for the analyzed methodology.

In brief, feature extraction serves to adapt the data set to the classification algorithms. By doing so, it can reduce the processing time of the following step and, more importantly, improve classification results. Even though this step can be potentially used to reduce data volume and, consequently, overall power dissipation, it usually represents high computational complexity for its implementation within implanted circuitry.

3.1.4 Spike Clustering

At last, the final stage consists in classifying the spikes into groups according to their features similarity expecting each one to contain action potentials fired by the same neuron. This task is done by clustering methods such as the k-means [25] and the Super-Paramagnetic Clustering (SPC) [26].

The fundamental idea of the k-means method is to classify a set of observations x into k clusters S minimizing Within-Cluster Sum of Squares (WCSS) given by:

$$\sum_{i=1}^{k}\sum_{x\in S_i}\|x_j-\mu_i\|^2. \tag{3.5}$$

From an initial set of k cluster centers, the standard algorithm alternates between two steps: the assignment of the observations to the cluster with the closest center in Euclidean distance and the update of these centers to the centroid of each resulting cluster. Eventually, the classification converges to a final result. This technique has the disadvantage of being supervised considering that the number of clusters must be priorly determined; on the other hand, it classifies the whole observation set. In [14], the definition of the cluster centers is performed by means of a Minimum Spanning Tree (MST); this tree is initialized with a random point of the data set and it is iteratively built by inserting the closest point to any point already belonging to it. Then, a threshold is set on the distances to create clusters; finally, their centers are used for the k-means initialization.

Alternatively, the SPC was developed based on ferromagnetic interactions described by the Potts model. Firstly, it assigns random clusters for each observation and it defines a value to represent the interaction strength among them. From this value, it derives the probability of each observation to change its cluster in the case its nearest neighbors do so. Then, it randomly chooses an observation to change its classification causing others to eventually change as well based on a Monte Carlo approach. This last step is iteratively repeated with the purpose to find a called super-paramagnetic phase in which the classification converges. For more details, check the original description of the method in [26] or its adapted implementation for spike clustering in [22]. In opposition to the k-means technique, SPC is an unsupervised approach; it can, nevertheless, leave unclassified observations.

The choice of an algorithm for spike clustering is of course essential to the outcome of the spike sorting process; still, all the preceding steps influence on the overall final performance. In essence, the success of a spike sorting technique is strictly related to the adaptation among the chosen methods for the realization of each step.

3.2 Compressed Sensing

The compression of neural data within the nodes of implanted hardware on a iEEG recording lowers the overall power consumption by reducing wireless data transmission rate. The use of CS for that end has been analyzed in [14] in terms of spike sorting results in which the step of spike clustering is directly applied on the compressed data. Considering that no decompression is performed in that case, this method could be as well classified as a feature extraction technique. In this subsection, the original CS method will be firstly presented followed by its adaptations for neural data processing.

CS was originally proposed in [13], it was conceived with the intention of allowing sparse signals to be recovered from representations with reduced sampling rates. The compression consists into representing a signal $x \in \mathbb{R}^m$ as a projection in a subspace given by:

$$y = \Phi \cdot x \text{ with } \Phi \in \mathbb{R}^{m\times N} \text{ and } m < N, \tag{3.6}$$

where Φ is called encoding matrix and $y \in \mathbb{R}^m$ is the compressed signal. The goal of this method is to determine an encoding matrix and a reconstruction algorithm for reliably recovering the original signal. Considering the matrix, it has been shown important the respect of the Restricted Isometry Property (RIP) [27] [28]. In turn, the reconstruction involves a linear system with infinite solutions; in that context, from the assumption of sparsity, algorithms such as Orthogonal Matching Pursuit and Iterative Hard Thresholding have shown efficient recovery with none or insignificant errors [29] [30].

In view of neural data compression, [14] shows that the use of a random binary encoding matrix containing the uniformly distributed elements ± 1 is efficient for spike sorting assuming the compression of individual spikes. In addition to that, spike clustering is shown feasible to be applied directly over the compressed data dismissing reconstruction algorithms. [14] suggests the use of $N = 32$ and $m = 6$ being N the number of samples which represent the original spikes and m its equivalent in the compressed space. The low calculation complexity of this method has stimulated the development of this work hardware implementation.

3.3 Neural Data Processing on Hardware

Several hardware solutions have been proposed for performing neural data processing associated to intra-cortically implanted MEAs. Namely, [11] presents a neural recording micro-system with an embedded ASIC for spike detection. Moreover, [12] introduces a 64-channel chip for spike detection, alignment, and feature extraction. Regarding the CS implementation, [9] describes a circuit which processes this compression technique on generic biological signals.

The system in [11] supports 256 neural recording sites composed by 32-channels where spike detection is individually performed for each signal source. This processing is realized on 5-bits converted data at 20 kHz by the definition of a threshold based on the calculation of its mean and its standard deviation. Additionally, the option of selecting the desired sites for data transmission reducing the overall bandwidth is provided. [11] claims that the ASIC fabricated in a 0.5 μm process composed by an 32-channel spike detection core, an analog-to-digital converter (ADC) and a storage unit can achieve an average data compression ratio of 92% while consuming 2.4 mW on a 6 mm^2 area.

[12] implements several spike sorting steps within a 64-channel implanted hardware. More specifically, it assumes an input data stream containing 8-bits samples at 24 kHz on which NEO detection and DD feature extraction are applied. Concerning alignment, it sets the points of the maximum derivative to the eleventh sample within 48-sample long spikes. The resulting circuit fabricated in a 90 nm process dissipates 2.03 μW per channel on an area of 0.06 mm^2 providing an estimated 91.25% compression ratio.

Looking to later evaluate this work hardware architecture, [9] CS compression hardware implementation serves for comparison purposes. Given that [9] considers generic biological signals, its data compression is performed without prior spike detection. Within its circuit, a CS binary encoding matrix is created by means of pseudo-random binary bit sequence generators; then, the compression is calculated with the support of XOR operators and accumulators. The final ASIC fabricated in a 90 nm CMOS process consumes 1.9 μW within 0.09 mm^2 considering a single recording channel, its compression ratio determined directly from the encoding matrix dimensions is of 90%.

These solutions were presented for sharing the goal of reducing wireless transmission bandwidth through implanted circuit processing. Their specifications are taken as parameters at the end of this report for evaluating this work approach.

4 MODELING AND SIMULATION

Having analyzed some state-of-the-art spike sorting techniques, this section evaluates the performance of their adapted implementation on hardware. The final goal is to define a complete spike sorting procedure optimizing parameters for the minimization of the circuit power and area maintaining satisfactory results in detecting and classifying action potentials. In the first subsection, the methodology for modeling and simulating the solutions is presented; then, each subsection is dedicated to the optimization of a different processing step.

4.1 Methodology

The results presented in the sequence of this section derive from mathematical modeling and simulation performed through Matlab; in the Appendix A, examples of these models are provided. For this end, a set of simulated neural data published by Quiroga, R. Q. was used in which spikes from three different classes were inserted at random times with a normalized peak amplitude of 1 (unitless). The noise levels were determined from their standard deviation σ_N set as 0.05, 0.1, 0.15, and 0.2 relative to this amplitude. Looking to reproduce real recording conditions, the data was firstly generated at 96 kHz and then downsampled to 24 kHz. A more detailed description of this data set is available in [22].

Regarding raw data treatment, the amplification noise is considered implied within the input signal; besides that, a Butterworh fourth order band-pass filter between 300 Hz and 3 kHz is assumed for noise suppression. As for spike detection, both the AV and the NEO spike detection methods are performed extracting 32-samples long spikes and setting the maximum amplitude among the detection instant and the subsequent 19 samples at the twelfth data point. Finally, for classification purposes, k-means clustering is performed being initialized by the construction of a MST.

4.2 Quantization

The choice of the quantization approach directly affects the power and the area of the circuit by determining the word bit width. As [20] has shown, a non-uniform quantization may reduce this parameter for the same spike sorting performance level. Thus, the following simulations verify this conclusion in the context of this work procedure. Considering that the quantization noise may affect both spike detection and clustering results, the outcomes of the standard AV and NEO detection techniques were analyzed along with the classification accuracy. In respect to feature extraction, PCA decomposition was performed prior to the classification. For different bit resolutions, an uniform quantizer and a non-uniform optimal quantizer assuming a Gaussian probability distribution function were compared. In both cases, the full-scale potential was set to 3.33 (unitless).

Firstly, the spike detection was evaluated. For this purpose, similarly to [20], the detection probability was defined as the number of correctly detected spikes divided by the total number of spikes in the input signal. The false alarm probability was defined as well as the number incorrect detections related to the number of samples not containing a spike normalized to the spike length. The Figure 4.1 and the Figure 4.2 exhibit these results for the AV and the NEO detection methods respectively, the AV false alarm probabilities were omitted for presenting negligible values being less than 1% for most cases and less than 4% in all cases.

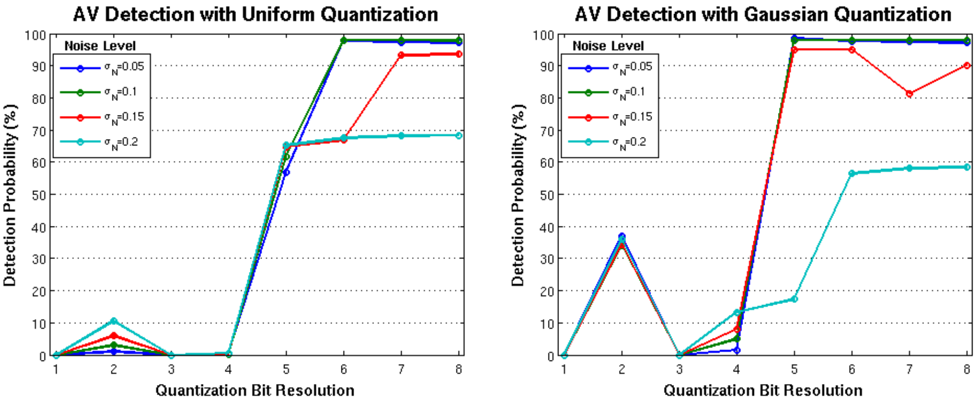


Figure 4.1: AV detection evaluation for different quantizations.

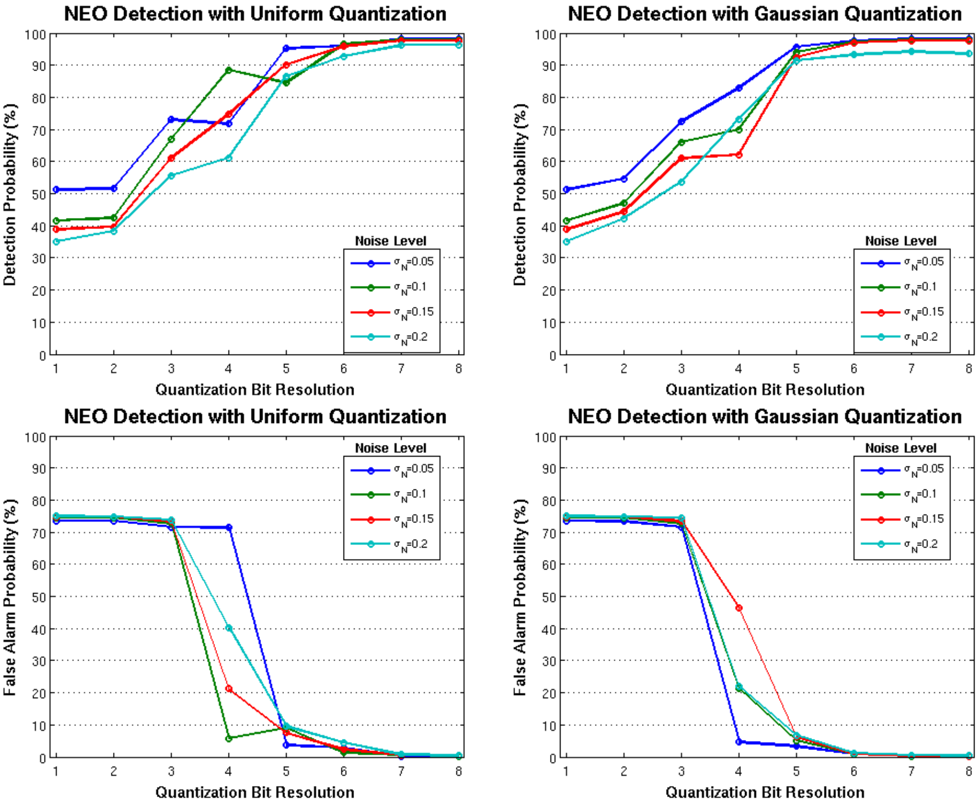


Figure 4.2: NEO detection evaluation for different quantizations.

In the plots of Figure 4.1, the highest detection probabilities considering all noise levels occur using at least a 7 bits and a 6 bits resolutions for the uniform and the Gaussian quantizations respectively. As the performance of the AV technique is known to significantly degrade with the decrease of SNRs, the results for the 0.2 noise level are disregarded for this method analysis allowing to have the 5 bits resolution as the optimal setting for the Gaussian quantization. Being so, [20] conclusions are sustained, *i.e.*, a non-uniform quantization approach can be used to reduce up to 2 bits in resolution maintaining high performance levels for the AV spike detection.

Analyzing the Figure 4.2, the same resolution is effective for NEO detection when using the Gaussian quantization; it represents a 1 bit reduction for similar detection probability compared to the uniform quantizer. In all cases, the increase of this parameter beyond this value does not provide considerable performance improvements. Regarding the false alarm probability, a 5 bits resolution shows again satisfactory results. As expected, the NEO method performs well in this configuration even for the highest simulated noise level in opposition to the AV approach.

Looking to verify the impact of quantization on spike clustering, resolutions ranging from 5 to 8 bits were evaluated in relation to their effect on the classification outcome. The results are shown in Figure 4.3 and in Figure 4.4 in respect to the AV and the NEO spike detection method respectively, they represent the correct classification probability defined as the number of correctly detected spikes which were also successfully classified divided by the total number of spikes in the input signal.

The classification performance is correlated with the detection results given that the number of successfully classified spikes is limited to the number of correct detections, this quantity also affects the precision in which the clustering algorithm determines the features to characterize each spike class. Indeed, the results in Figure 4.3 and in Figure 4.4 show similarities with their respective detection probability. Thereby, the classification step was not significantly affected directly from the quantization resolution, it was actually influenced by the propagation of its impact on the detection. In fact, the optimal quantization bit resolutions considering the classification are the same as the obtained from the detection evaluation. Henceforth, the sequence of this work will consider a 5 bits non-uniform quantization assuming a Gaussian probability distribution function as it represents the minimum bit resolution which maintains high performance levels.

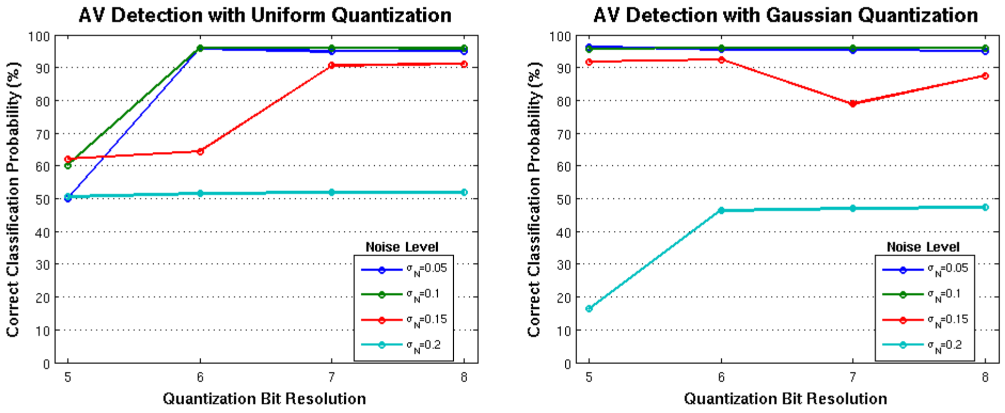


Figure 4.3: Spike clustering evaluation posterior to AV detection for different quantizations.

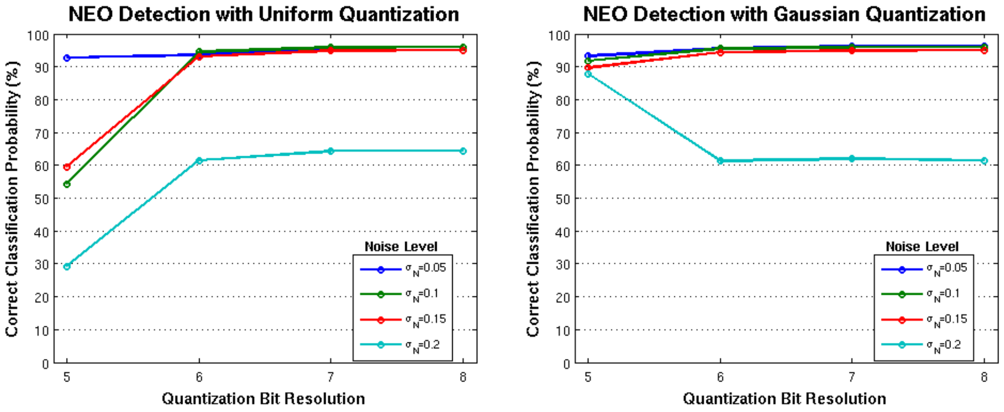


Figure 4.4: Spike clustering evaluation posterior to NEO detection for different quantizations.

4.3 Threshold Setup Time

The spike detection techniques described and selected in section 3 present low calculation complexity, they are based on the definition of thresholds from the characteristics of the input data. When performing real-time spike sorting, these values must be set from the recent signal history such as to estimate its attributes. On this direction, the objective of this subsection is to define a setup time representing the length of signal observation for the determination of these thresholds. In view of hardware design, the minimization of this duration may reduce storage requirements.

From that perspective, the threshold estimate using different setup times were evaluated according to its error rate in comparison with a full signal length calculation. Thinking on an implementation based on digital counters, the durations were set according to the lengths of complete counting cycles using different bit widths at the sampling frequency (24 kHz). 60 seconds long simulated signals were used for the evaluation assuming that their properties are sufficient to represent the ones from real-time recording on the same channel; in the worst case in which this premise is false, it would be necessary to recompute the threshold for each 60 s. The plots in Figure 4.5 illustrates the threshold estimate error rate for the hardware adapted AV detection method and for the NEO approach.

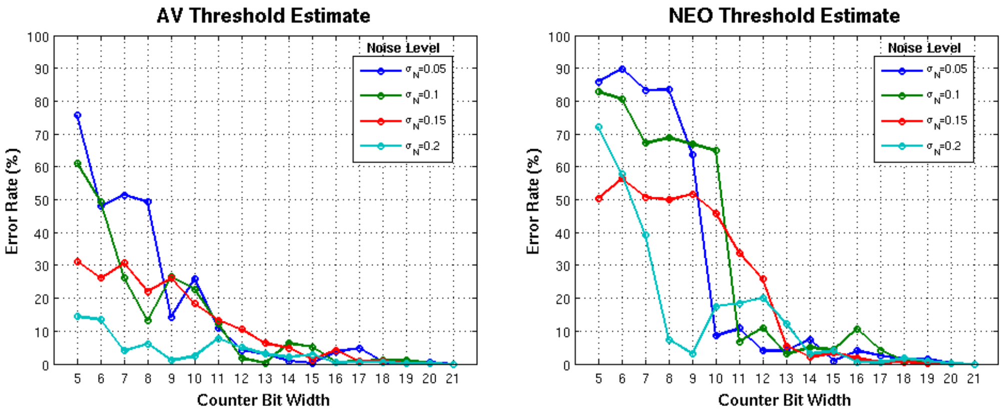


Figure 4.5: Threshold estimate error for different setup time defined from digital counter bit widths.

The results show that for the AV technique the threshold converges quicker than the NEO approach with the increase of the setup time. Effectively, in order to obtain a error rate inferior than 10% for the simulated signals, a 13 bits wide counter is sufficient for the first detection method whereas a 14 bits one is required for the other. Aiming to standardize this parameter, the threshold setup time will be set to 683 ms (14 bits digital counter cycle length at 24 kHz). In this way, the resulting detection performance is identical to the ones in Figure 4.1 and in Figure 4.2 which consider the whole signal for threshold computation.

4.4 CS Encoding Matrix

Essential key to the CS method, the encoding matrix must be herein defined aiming the preservation of the characteristics serving to distinguish spike classes. Taking into account that these optimal features are still unknown, the choice of the encoding matrix in this work is based on experimentation. This matrix was chosen to be hard-coded within the circuit logic for saving silicon area opposing to SRAM storage and real-time generation [9]; in consequence, the selection of a single matrix was required.

In accordance with [14], several random matrices in $\{-1, 1\}^{6\times 32}$ were generated and tested in respect to the differences among classes of the resulting compressed spikes representation. On this direction, the matrix with the highest value for the division of the minimum distance between two cluster centers and the maximum distance of a spike from its own cluster was selected. For each matrix, this factor was averaged regarding different noise levels. Looking to eliminate the influence of the detection step, the compression was performed over the true spikes. Additionally, the compressed action potentials were classified after a PCA decomposition and the matrices which presented more than 10% of classification errors were discarded.

The Figure 4.6 compares the spike clustering outcome for uncompressed and compressed spikes. The results show that the CS compression did not degrade the results; in fact, it has even allowed a greater correct classification probability at the highest noise level. This fact suggests that the performed CS method besides reducing data volume can as well improve the spike sorting procedure when carefully choosing an encoding matrix.

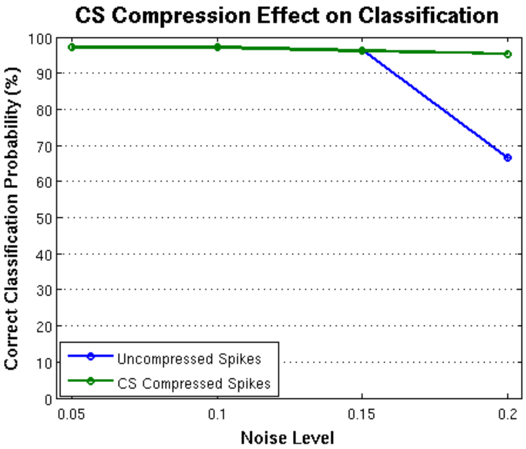


Figure 4.6: Spike clustering evaluation from uncompressed and CS compressed spikes.

4.5 Feature Extraction

Prior to the application of a clustering algorithm to classify spikes, their features are usually extracted by their decomposition using PCA or DWT. Beyond that, as stated in section 3, the CS compression can be as well solely considered as a feature extraction technique. In this subsection, clustering results are compared using those three approaches. Again, this process was performed over the true spikes as to exclude the spike detection impact. The classification correctness in the uncompressed domain without any transformation, using PCA, and using DWT was firstly evaluated; then, the same was done in the CS compressed domain.

Surprisingly, the classification performance did not present any difference among these features extraction techniques; the correct classification probability was the same as the illustrated in Figure 4.6. Since the results were inconclusive regarding the best approach, the PCA decomposition will be arbitrarily chosen for this step in the sequence of this work.

4.6 Integrated Spike Sorting

Combining all the previously chosen techniques, the full spike sorting procedure can be summarized as:

- Filtering: Butterworth fourth order band-pass filter between 300 Hz and 3 kHz;
- Quantization: non-uniform Gaussian quantization with a 5 bits resolution;
- Spike detection: AV or NEO method with 683 ms for threshold setup, 32 samples per spike with the maximum value at the twelfth data point;
- Compression: CS with fixed encoding matrix in $\{-1, 1\}^{6\times 32}$;
- Feature extraction: PCA decomposition;
- Spike clustering: k-means clustering initialized from the construction of a MST.

This whole processing was simulated for performance evaluation and for generating the circuit expected results thinking on its validation. The Figure 4.7 illustrates the overall spike sorting performance; additionally, it includes the PCA representation of the first two principal components of the compressed spikes distinguishing each class by colors.

From these results, the AV and the NEO detection techniques can be compared. In respect to spike detection, as concluded by [21], the second approach shows better adaptation for higher noise levels; nonetheless, it generally results in greater false alarm probability in all cases. When analyzing spike clustering, both implementations present much lower correct classification probability for the 0.2 noise level. Although this can be explained for the AV technique by its low detection performance, the same cannot be assumed for the NEO case; such explanation would require further analysis which is left future work. These two methods are chosen for hardware implementation within two distinct circuits.

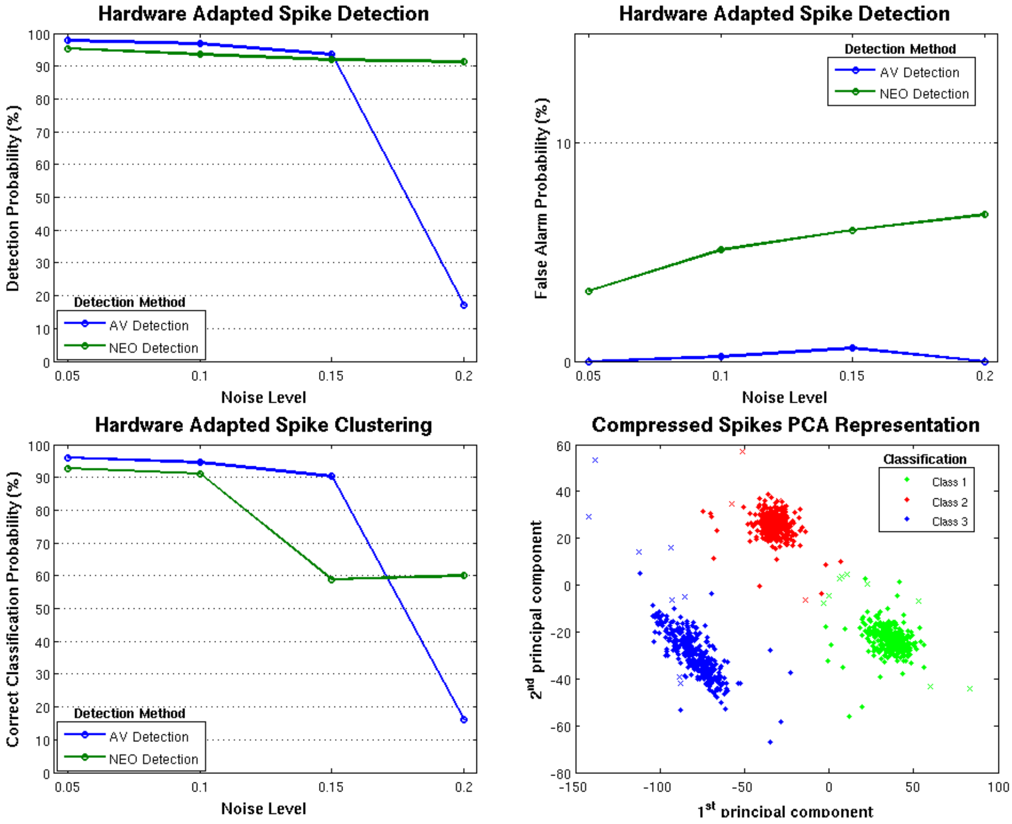


Figure 4.7: Hardware adapted spike sorting evaluation. The PCA representation is for the AV detection at the lowest noise level, correctly classified spikes are represented by a point, others are represented by ’X’ marks.

5 PROPOSAL AND METHODOLOTY

In this section, the architecture of the circuits for the previously defined spike detection and CS compression approach is specified. At first, the BMI system which these circuits are supposed to compose is described along with the constraints that must be respected in order to actually implement the solution. After that, the hardware architectures for each processing step are individually presented. At last, their design methodology is defined.

5.1 BMI System General Structure

In short, the BMI system herein assumed is composed by intra-cortically implanted MEAs with associated hardware for real-time raw data treatment, spike detection, and CS compression. The resulting data is then to be wirelessly sent to a central processing unit responsible for feature extraction, for spike clustering, and for the control of an actuator. This structure is illustrated by the Figure 5.1.

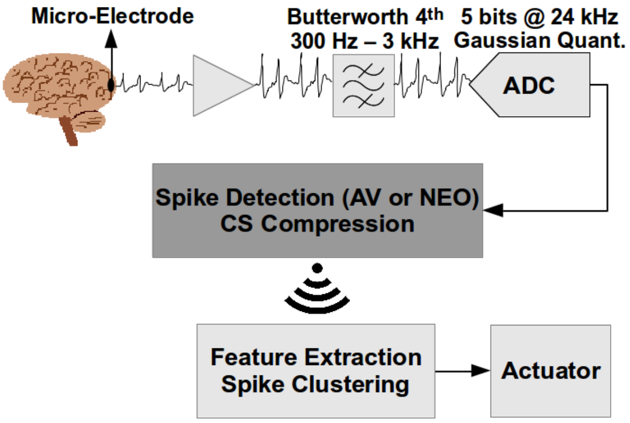


Figure 5.1: BMI system topology.

The implanted circuitry is limited in power density for not damaging internal cells. On that direction, performing spike detection and compression may reduce the overall dissipated energy by decreasing the wireless data transmission rate. However, for the same reason, this processing unit must respect rigorous constraints. Effectively, the whole implanted hardware must present a power density inferior to 800 μW/mm^2 [8].

On top of that, this circuit must fit on the base area of the MEA. Taking the the Utah array [31] as a model, this area is limited to 0.16 mm^2 per recording channel which must be shared with the amplifier, the filter, and the ADC. From this perspective, the area of the circuit to be designed must be of the order of 0.01 mm^2.

5.2 Circuit Architecture

Both the AV and the NEO spike detection techniques were chosen to be implemented given that the former presents very low computational complexity and the latter is adapted to higher noise levels. As a result, two circuits are to be designed only differing on their detection method; they share the same global architecture which is shown in Figure 5.2. In brief, they receive neural data 5 bits samples at 24 kHz from which spikes must be detected, extracted, and compressed. For this end, they are composed by three main units: a shift buffer, the Spike Detector, and the CS Compressor. The first simply stores incoming data being accessible to the other two blocks. The Spike Detector calculates its corresponding threshold, controls spike alignment, and informs the presence of an action potential. Finally, the CS Compressor multiplies the encoding matrix by the detected spikes. The implementation details of these two last units will be individually described in the sequence.

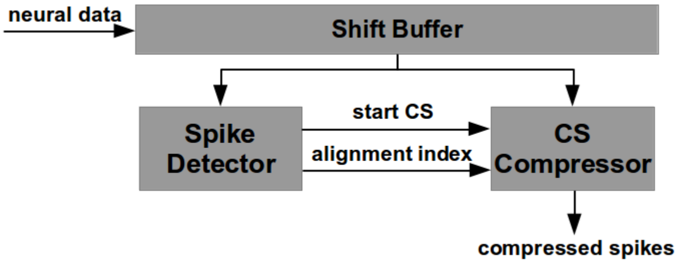


Figure 5.2: General circuit architecture.

5.2.1 Spike Detectors

The chosen spike detection approaches can be subdivided into two processing phases: threshold computation and real-time detection. As already discussed, the first one occurs within 683 ms of signal observation controlled by a 14 bits digital counter at 24 kHz; during this period, the threshold calculation differs between the AV and the NEO detection technique. After that, the second phase simply compares the input signal with this value. Following all that, the spike alignment is processed whenever an action potential is detected. The topology of the hardware implementations described in the sequence are illustrated in Figure 5.3.

The AV threshold definition presented in section 3 can be rewritten as:

$$Thr = \frac{4}{0.6745} \cdot median\{|x|\} \approx 5.9303 \cdot median\{|x|\}. \tag{5.1}$$

Considering the quantization, $|x|$ contains only integer values between 0 and 16. From this premise, its median is as well restricted within this range with 0.5 steps as this operator averages the two mid-samples in the case of an even number of elements. In order to detect spikes, the threshold must be inferior than the maximum signal amplitude (15); therefore, the median is assumed to be at most 2.5. From this perspective, the median operator is implemented by 3 registers that accumulate the number of samples which respect $|x(n)| = 0$, $|x(n)| = 1$, and $|x(n)| = 2$. At the end of the threshold setup time, these values are processed for the determination of the median which serves to select the corresponding threshold value through a multiplexer. Afterwards, during the real-time detection phase, this value is constantly compared with every incoming sample.

In respect to the NEO threshold calculation, a combinatory hardware implementation of the NEO operator was created. During the threshold setup time, the results of this operator on the incoming samples are accumulated within a register; at the end, they are averaged and multiplied by the threshold multiplication factor set to 8. As this factor and the number of samples taken into consideration are both powers of two, this processing simply involves shifting binary words; note that the disregard of remainders does not affect the the spike detection as the NEO operator on integer values can only result in integer values as well. The following phase consists in calculating the NEO operator for every sample and comparing it to this precomputed threshold.

Once an action potential is detected, the spike alignment starts. For that purpose, an independent unit keeps track of the maximum spike amplitude position within the shift buffer; this processing considers the sample at the detected instant and the following 19. Finally, the CS Compressor is signalized to start its processing receiving the information regarding the alignment.

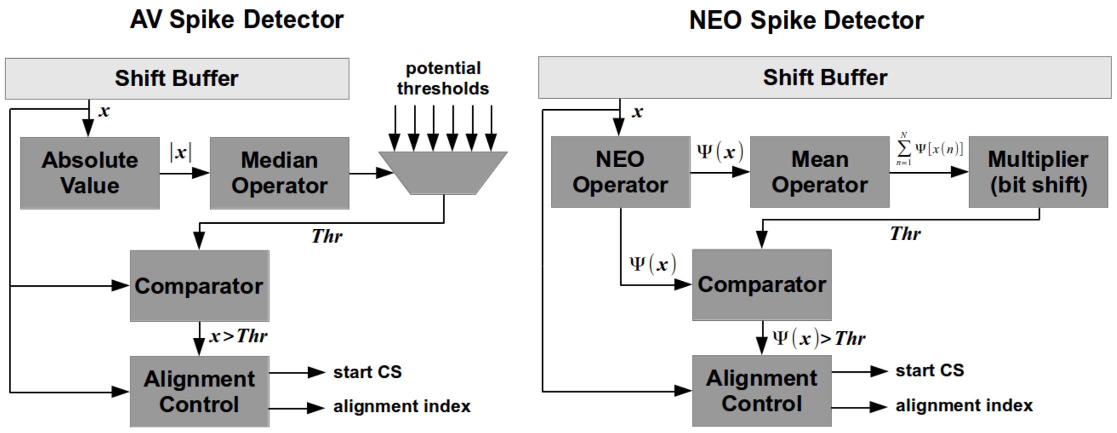


Figure 5.3: Spike detectors.

5.2.2 CS Compressor

Following the definition of the spike alignment, the CS Compressor processes its multiplication with the encoding matrix. As this matrix is only composed by ± 1 elements, this calculation consists in simply summing samples eventually inverting their signs. In this sense, by using the binary values 0 and 1 to respectively represent the matrix elements +1 and -1, the multiplications can be efficiently implemented by means of XOR operations as in [9]. Despite this whole processing could be performed using combinatory logic, looking to reduce the circuit area, it was sequentially implemented in 32 clock cycles with the support of registers to accumulate partial results. Furthermore, since the compressed spikes values may overflow the input sample word length, 2 extra bits were inserted for their representation. Even so, this approach does not eliminate this possibility which is signalized by an additional output. This architecture is shown in Figure 5.4.

The presence of a single CS Compressor precludes simultaneous compressions. On that account, the spike detection must be deactivated during this processing which causes the system to neglect the occurrence of some overlapping spikes. The compression was chosen to be performed within the time of arrival of the next 32 samples after the detection instant. Given that by the time the spike alignment is complete 20 new samples have already come, this processing is restricted to last $(32 - 20)/24\,kHz = 0.5\,ms$; thus, the

32 compression cycles must be realized at 64 kHz.

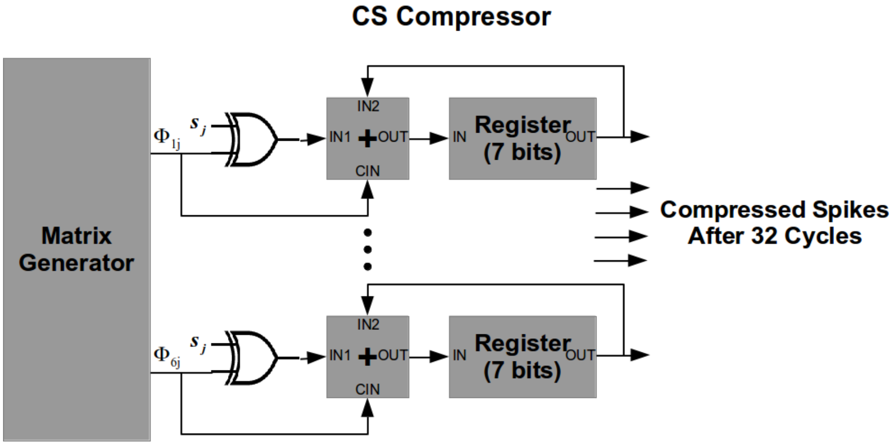


Figure 5.4: CS Compressor. The overflow control is omitted.

5.3 Design Methodology

The methodology used for designing the circuits was divided in three main phases: behavioral description, logic synthesis, and physical layout. They all included a verification step supported by Matlab modeling results. Figure 5.5 globally illustrates this flow.

The first phase consisted in the describing the chosen architectures at the Register-Transfer Level (RTL) by means of the hardware description language VHDL. This process occurred in a modular bottom-up approach where simple hardware elements such as comparators, counters, registers, and accumulators were first described, validated, and later combined to build the more complex structures. At the end, a complete functional verification was performed using the simulation software ModelSim.

After that, logic synthesis was realized within Cadence Encounter RTL Compiler using the X-FAB CMOS 0.18 μm standard-cell library. On that occasion, the intent was to optimize energy efficiency which was estimated with the switching activity information obtained through ModelSim. The best results were then selected for the physical layout. The scripts used in this phase are available in the Appendix B.

At last, the physical layout consisted in floorplanning, powerplanning, placement, clock tree synthesis, routing, and verification. This phase was performed looking to minimize area respecting design rules, timing constraints, and functionality. For that end, it was used Cadence Encounter Digital Implementation System.

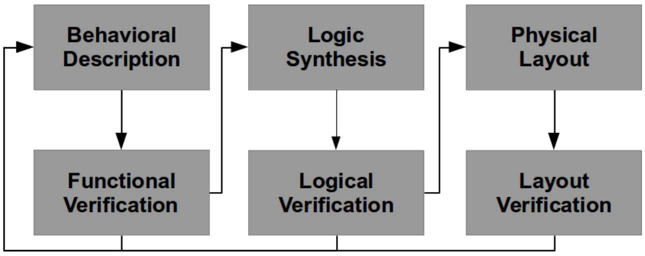


Figure 5.5: Design flow.

6 RESULTS AND DISCUSSION

This section presents the logic synthesis and physical layout results for the proposed circuits along with their design techniques. Beyond that, it discusses the solution comparing it to the related work described in section 3.

6.1 Logic Synthesis

Logic synthesis was performed targeting the minimization of energy consumption. Considering the chosen IC fabrication process node, the dynamic power was assumed to be the determinant factor for the overall circuit energy efficiency. Still, high transistor threshold voltage was used in order to optimize leakage power. On the other hand, the clock gating technique was applied to reduce dynamic power.

Standard-cells composed by transistors with high threshold voltage provide low leakage power at the expense of reduced transition speed. In circuits where processing performance is critical, multiple threshold voltages are used according to the timing restrictions of each processing path. As the herein proposed circuits are power critical, this parameter was maximized for their whole implementation. Considering that the supply voltage greatly impacts on the energy efficiency as well, they were designed using the lowest supported supply voltage (1.8 V) for the chosen fabrication process with the highest threshold voltage available (0.8 V).

In respect to architectural modifications, clock gating disables parts of the clock tree when the corresponding synchronous logic is not in use. By these means, it decreases the number of signal transitions and consequently reduces the dynamic power. This technique was automatically implemented by the synthesis software, its impact is shown by the Table 6.1. As expected, it has not presented any effect on the leakage power whereas it has reduced dynamic power in 30,72% and in 23.84% for the circuits with the AV and the NEO spike detectors respectively.

Table 6.1: Post-synthesis power estimation with and without clock gating (CG).

Detection Method	**AV**		**NEO**	
Optimization	**without CG**	**with CG**	**without CG**	**with CG**
Leakage Power	0.91 nW	0.92 nW	0.83 nW	0.84 nW
Dynamic Power	721.8 nW	500.06 nW	756.27 nW	575.97 nW
Total Power	722.71 nW	500.98 nW	757.1 nW	576.81 nW

All the synthesized circuits were validated regarding their functionality; Figure 6.1 and Figure 6.2 show simulation snapshots which illustrate this step. Afterwards, as the synthesized circuits with the clock gating technique were the ones that presented the lowest power, their respective netlists were selected for the physical layout.

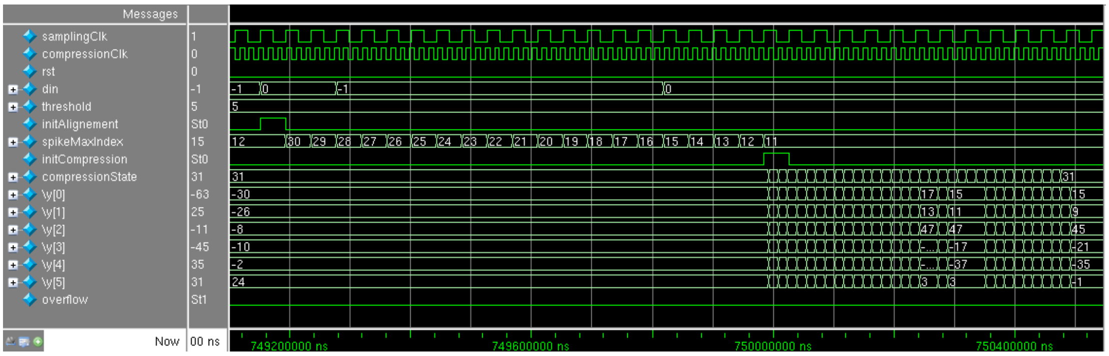


Figure 6.1: Post-synthesis simulation snapshot of the circuit using the AV spike detection method.

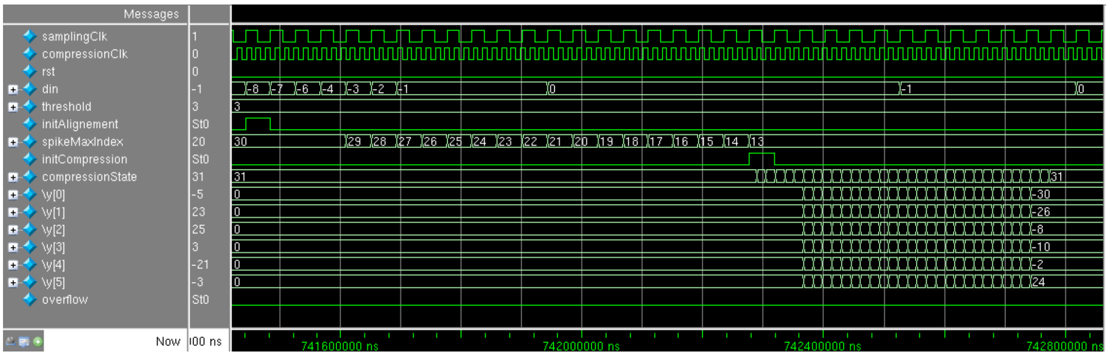


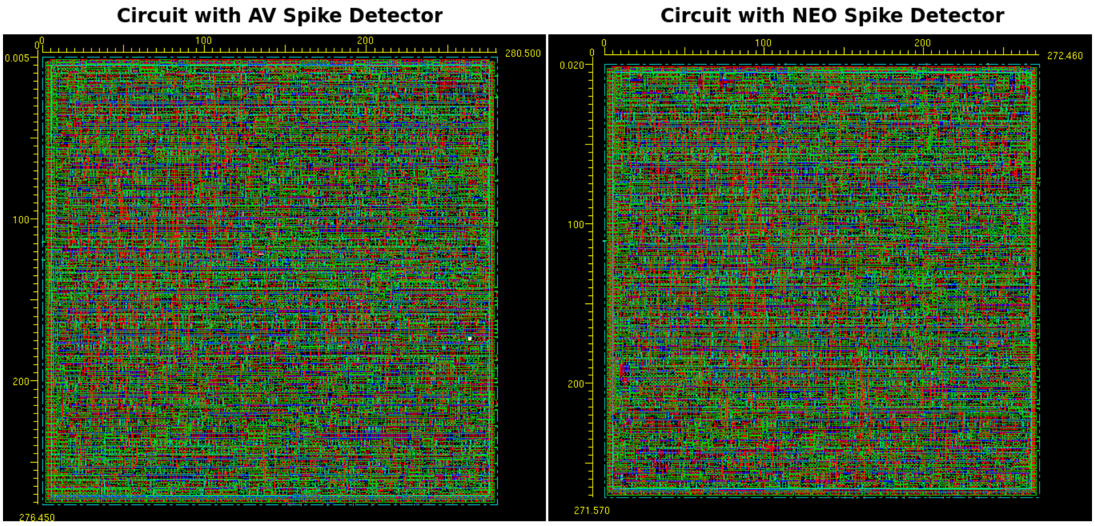
Figure 6.2: Post-synthesis simulation snapshot of the circuit using the NEO spike detection method.

6.2 Physical Layout

The physical layout was designed looking to minimize area. Since the proposed solutions are supposed to share a restricted surface with additional circuitry, they were designed as IPs without any input or output pad.

At first, the floorplanning was set as to obtain 80% of core utilization maintaining the aspect ratio close to 1; 5 μm were reserved at each side for the power rings and evenly spaced pairs of rows were created to allocate the cells. After that, the powerplanning consisted in creating power rings and horizontal stripes. Then, the standard-cells composing the circuit logic were automatically placed along with filler cells. Over this design, the clock tree synthesis was performed followed by timing verification and routing. Finally, extra metal was inserted in order to respect the metal density constraint for the fabrication process.

The resulting layouts were verified regarding design rules and connectivity. The final circuits are illustrated in Figure 6.3 where their dimensions are indicated. The area obtained for circuit performing AV spike detection was of approximately 0.078 mm^2 whereas the one with the NEO spike detector presented 0.074 mm^2.



Circuit with AV Spike Detector

Circuit with NEO Spike Detector

Figure 6.3: Final layouts with dimensions indicated in μm.

6.3 Discussion

Given that the designed circuits respect the specifications for real implementation, this subsection analyzes these solutions and compares them with the related approaches presented in section 3. For that purpose, both hardware parameters and the final data rate reduction are taken into account.

Although the decision of designing a circuit performing AV spike detection was taken for its low computational complexity expecting it to result in significant power and area minimization, its comparison with the circuit with the NEO spike detector shows little difference regarding these aspects. Therefore, considering the better adaptation of the latter to high noise levels, it is recommended as the best alternative. With respect to data compression, both circuits are equivalent; assuming an average firing rate of 100 spikes/s, the input rate of 120000 bps (24000 samples/s x 5 bits/sample) is reduced to 4200 bps (100 spikes/s x 6 words/spike x 7 bits/word) representing a data rate reduction of 96.5%.

The comparison of this solution with the existing related work is shown by the Table 6.2. It is noteworthy that the herein proposed circuits present the lowest power specifications with areas only greater than [12] where the fabrication process allows higher logic density. More importantly, the data rate reduction is higher than any other referenced solution. For instance, assuming the wireless transmission energy consumption at 3 nJ/bit [32], this work circuits result in a transmission power of 12.6 μW per channel whereas the other referenced hardwares achieve a minimum of 92.16 μW. This difference derives from the CS methodology and the reduced bit width thanks to the optimized quantization approach.

Table 6.2: Solutions comparison considering a single channel. The average spike firing rate is estimated at 100 spikes/s.

Reference	[11]	[12]	[9]	This work (AV)	This work (NEO)
Process	500 nm	90 nm	90 nm	180 nm	180 nm
Area	0.19 mm^2	0.06 mm^2	0.09 mm^2	0.078 mm^2	0.074 mm^2
Power	75 μW	2.03 μW	1.9 μW	0.5 μW	0.58 μW
Power Density	681.81 μW/mm^2	33.83 μW/mm^2	21.11 μW/mm^2	6.41 μW/mm^2	7.84 μW/mm^2
Data Rate Reduction	92%	91.25%	90 %	96.5%	96.5%

7 CONCLUSION

This work proposed low-power hardware implementations for neural spike detection and CS compression to be associated with intra-cortically implanted MEAs within BMI systems. The state-of-the-art neural data processing techniques were investigated as to define a spike sorting methodology adapted to the compression approach. Thereafter, two circuits were designed as to meet the constraints imposed by the related physiological context. The objective of minimizing the overall implanted circuit power through the reduction of wireless data bandwidth was shown feasible and efficient via the suggested means.

The analysis of previous works concerning neural data processing allowed the identification of the current most accurate spike sorting methodologies. Thereupon, further evaluation through mathematical simulation served to define and optimize parameters ensuring satisfactory spike sorting performance after the proposed data compression. Besides others contributions, this procedure permitted the minimization of the quantization bit width which has directly affected the final circuits specifications.

After that, a low-area and low-power driven design was performed for the circuits design. Their architecture, their logic synthesis, and their physical layout were carefully designed on this sense. On that occasion, techniques such as the use of high threshold voltage and clock gating were employed. Finally, the results showed to satisfy the requirements for real implementation.

Lastly, the herein proposed solution was compared with related work. The designed circuits were shown the most optimized regarding area and power among the analyzed implementations. Furthermore, the compression methodology was the one that provided the greatest data rate reduction.

7.1 Future Work

Despite all the advantages cited above, the suggested approach still requires supplementary study. For instance, a detailed analysis of its performance within a more realistic BMI environment is necessary. In addition, it could be improved with some architectural modifications.

Even though the chosen spike sorting methodology was verified for different noise levels, this processing must be validated with real recording data in the context of a complete BMI system. By doing so, it would be possible to estimate the precision provided by this method for the control of actuators. As a result, a solution for the restoration of sensorimotor functions could be proposed.

From the hardware perspective, a single architecture for the processing of multiple recording channels can potentially increase the overall system efficiency by using shared resources. This circuit would benefit from less stringent area limitations as a larger surface would be available for its implementation attached to the MEA. Under those circumstances, a strategy for dealing with overlapping spikes could be applied improving spike sorting results.

In conclusion, future work may include the design of a multiple-channel spike detection and compression circuit to be implemented within a BMI system controlling actuators. Such system may be used to perform sensorimotor functions for patients suffering from physical limitations or neurological disorders.

REFERENCES

[1] N. A. Carlson. *Foundations of Physiological Psychology, 2nd Edition*. Allyn and Bacon, 1992.

[2] S. Freeman. *Biological Science, 2nd Edition*. Prentice Hall, 2005.

[3] J. Echauz, G. Georgoulas, O. Smart, A. Gardner, H. Firpi, B. Litt, and G. J. Vachtsevanos. Monitoring, signal analysis, and control of epileptic seizures: a paradigm in brain research. In *Proceedings Mediterranean Conference on Control & Automation*, pages 1–6, Athens, Greece, June 2007.

[4] F. Galán, M. Nuttin, E. Lew, P. W. Ferrez, G. Vanacker, J. Philips, and J. R. Millán. A brain-actuated wheelchair: asynchronous and non-invasive brain computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119:2159–2169, September 2008.

[5] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz. Cortical control of a prosthetic arm for self-feeding. *Nature*, 453:1098–1101, June 2008.

[6] J. Ph. Lachaux, D. Rudrauf, and P. Kahane. Intracranial EEG and human brain mapping. *Journal of Physiology, Paris*, 97:613–628, July 2003.

[7] E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7:456–461, April 2004.

[8] T. M. Seese, G. M. Saidel, and C. R. Davies. Characterization of tissue morphology, angiogenesis and temperature in the adaptative response of muscle tissue to chronic heating. *Laboratory Investigation*, 78:1553–1562, December 1998.

[9] F. Chen, A. P. Chandrakasan, and Stojanović. Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors. *IEEE Journal of Solid-State Circuits*, 47:744–756, March 2012.

[10] C. R. Noback, N. L. Strominger, R. J. Demarest, and D. A. Ruggiero. *The Human Nervous System, 6th Edition*. Humana Press Inc., 2005.

[11] R. H. Olsson III and K. D. Wise. A three-dimensional neural recording microsystem with implantable data compression circuitry. *IEEE Journal of Solid-State Circuits*, 40:2796–2804, December 2005.

[12] V. Karkare, S. Gibson, and D. Marković. A 130-μW, 64-channel neural spike-sorting DSP chip. *IEEE Journal of Solid-State Circuits*, 46:1214–1222, May 2011.

[13] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, April 2006.

[14] B. Coppa, R. Héliot, O. Michel, and D. David. Low-cost intracortical spiking recordings compression with classification abilities for implanted BMI devices. In *34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2623–2626, San Diego, CA, United States, August 2012.

[15] R. H. Hall. Neural communication. 1998.

[16] W. Wattanapanitch, M. Fee, and R. Sarpeshkar. An energy-efficient micropower neural recording amplifier. *IEEE Transactions on Biomedical Circuits and Systems*, 1:136–147, June 2007.

[17] R. Q. Quiroga. Spike sorting. *Scholarpedia Journal*, 2:3583, December 2007.

[18] M. S. Fee, P. M. Mitra, and D. Kleinfeld. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability. *Journal of Neuroscience Methods*, 69:175–188, November 1996.

[19] P. P. Mitra and H. Bokil. *Observed Brain Dynamics*. Oxford University Press, 2007.

[20] S. Gibson, V. Wang, and D. Marković. Effects of quantization on neural spike sorting. In *IEEE International Symposium on Circuits and Systems*, pages 2099–2102, Rio de Janeiro, Brazil, May 2011.

[21] S. Gibson, J. W. Judy, and D. Marković. Comparison of spike-sorting algorithms for future hardware implementation. In *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5015–5020, Vancouver, Canada, August 2008.

[22] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, 16:1661–1687, August 2004.

[23] J. F. Kaiser. On a simple algorithm to calculate the 'energy' of a signal. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 381–384, Albuquerque, NM, United States, April 1990.

[24] Z. Nadasdy, R. Q. Quiroga, Y. Ben-Shaul, B. Pesaran, D. A. Wagenaar, and R. Andersen. Comparison of unsupervised algorithms for on-line and off-line spike sorting. *Abstract Viewer/Itinerary Planner*, 2002.

[25] J. A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[26] M. Blatt, S. Wiseman, and E. Domany. Superparamagnetic clustering of data. *Physical Review Letters*, 76:3251–3254, April 1996.

[27] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transaction on Information Theory*, 51:4203–4215, December 2005.

[28] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346:589–592, May 2008.

[29] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transaction on Information Theory*, 53:4655–4666, December 2007.

[30] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27:265–274, November 2009.

[31] R. A. Normann. Microfabricated electrode arrays for restoring lost sensory and motor functions. In *12th International Conference on Transducers, Solid-State Sensors, Actuators and Microsystems*, volume 2, pages 959–962, Boston, MA, United States, June 2003.

[32] J. L. Bohorquez, A. P. Chandrakasan, and J. L. Dawson. A 350 μW CMOS MSK transmitter and 400 μW OOK super-regenerative receiver for medical implant communications. *IEEE Journal of Solid-State Circuits*, 44:1248–1259, April 2009.

APPENDIX A MATLAB MODELING EXAMPLES

A.1 Gaussian Quantization

```
function quantized_data = gaussian_quantization(data, bit_resolution, vfs)
%% Performs gaussian quantization
%
%  INPUT
%        data           input data
%        bit_resolution quantization bit resolution
%        vfs            full-scale voltage
%
%  OUTPUT
%        quantized_data quantized data

x = -vfs/2:0.001:vfs/2;
gaussian_dist = gaussmf(x, [1 0]);

acc_gaussian_dist = zeros(1, length(gaussian_dist));
acc_gaussian_dist(1) = gaussian_dist(1);
for i = 2:length(gaussian_dist)
    acc_gaussian_dist(i) = acc_gaussian_dist(i-1) + gaussian_dist(i);
end
acc_gaussian_dist = (2^bit_resolution-1)*acc_gaussian_dist/max(acc_gaussian_dist);

quantized_data = zeros(1, length(data));
for i = 1:length(data)
    if data(i) > max(x)
        quantized_data(i) = 2^(bit_resolution-1)-1;
    else
        quantized_data(i) = round(acc_gaussian_dist(find(data(i) <= x, 1)))
                            - 2^(bit_resolution-1);
    end
end
```

A.2 AV Spike Detection

```
function [spikes spike_indexes] = AV_spike_detection(data, C, N)
%% Performs AV spike detection using 32 samples as spike length and aligning the
%  maximum amplitude to the twelfth sample
%
%  INPUT
%        data          input data
%        C             threshold multiplication factor
%        N             number of samples to consider for the threshold calculation
%
%  OUTPUT
%        spikes        detected spikes
%        spike_indexes detected spikes maximum amplitude indexes within ’data’

std_estimate = median(abs(data(1:N))/0.6745);
threshold = C*std_estimate;

spikes = [];
spike_indexes = [];
detection_indexes = find(data > threshold);
last_extracted_index = 0;
for i = 1:length(detection_indexes)
    if detection_indexes(i) > last_extracted_index &&
       detection_indexes(i)+19 < length(data)
        [~, max_idx] = max(data(detection_indexes(i):detection_indexes(i)+19));
        max_idx = max_idx + detection_indexes(i)-1;
        if max_idx-11 > 0 && max_idx+20 < length(data)
            spikes = [spikes; data(max_idx-11:max_idx+20)];
            spike_indexes = [spike_indexes max_idx];
            last_extracted_index = detection_indexes(i)+31;
        end
    end
end
spikes = spikes’;
```

A.3 NEO Spike Detection

```
function [spikes spike_indexes] = NEO_spike_detection(data, C, N)
%% Performs NEO spike detection using 32 samples as spike length and aligning the
%  maximum amplitude to the twelfth sample
%
%  INPUT
%        data          input data
%        C             threshold multiplication factor
%        N             number of samples to consider for the threshold calculation
%
%  OUTPUT
%        spikes        detected spikes
%        spike_indexes detected spikes maximum amplitude indexes within 'data'

threshold = 0;
for i = 2:N-1
    threshold = threshold + data(i)^2 - data(i+1)*data(i-1);
end
threshold = C*threshold/(N-2);

neo = zeros(1, length(data)-2);
for i = 2:length(data)-2
    neo(i-1) = data(i)^2 - data(i+1)*data(i-1);
end

spikes = [];
spike_indexes = [];
detection_indexes = find(neo > threshold) + 1;
last_extracted_index = 0;
for i = 1:length(detection_indexes)
    if detection_indexes(i) > last_extracted_index &&
       detection_indexes(i)+19 < length(data)
        [~, max_idx] = max(data(detection_indexes(i):detection_indexes(i)+19));
        max_idx = max_idx + detection_indexes(i)-1;
        if max_idx-11 > 0 && max_idx+20 < length(data)
            spikes = [spikes; data(max_idx-11:max_idx+20)];
            spike_indexes = [spike_indexes max_idx];
            last_extracted_index = detection_indexes(i)+31;
        end
    end
end
spikes = spikes';
```

A.4 Spike Clustering

```
function spike_classes = spike_clustering(spikes, ref_classes, feature_extraction)
%% Performs spike clustering constructing a MST to initialize k-means clustering
%
%   INPUT
%         spikes              spikes
%         ref_classes         real spike classes, 0 for for inexistent spikes
%         feature_extraction selects the feature extraction approach (default PCA)
%
%   OUTPUT
%         spike_classes       assigned spike classes

if isempty(spikes) == 1
    spike_classes = [];
    return;
end

% Feature extraction
if strcmp(feature_extraction, 'none') == 1
    spike_features = spikes;
elseif strcmp(feature_extraction, 'WT') == 1
    spike_features = zeros(size(spikes, 1), size(spikes, 2));
    for i = 1:size(spikes, 2)
        if size(spikes, 1) == 6
            spike_features(:, i) = pm_haar(spikes(:, i), 1);
        else
            spike_features(:, i) = pm_haar(spikes(:, i), 4);
        end
    end
else
    PC = pca(spikes);
    spike_features = (spikes'*PC)';
end

% Calculates the distances among all points
distances = zeros(size(spike_features, 2), size(spike_features, 2));
for i = 1:size(spike_features, 2)
    for j = 1:i
        if i == j
            distances(i, j) = Inf;
        else
            distances(i, j) = norm(spike_features(:, i) - spike_features(:, j));
            distances(j, i) = distances(i, j);
        end
    end
end

% MST construction
mst = zeros(1, size(spike_features, 2));
mst(1) = randi([1 size(spike_features, 2)]);
mst_distances = zeros(1, size(spike_features, 2));
for i = 2:length(mst)
    min_distance = Inf;
    closest_spike = 0;
    for j = 1:i-1
        [min_value min_idx] = min(distances(mst(j), :));
        if min_value < min_distance
```

```
            closest_spike = min_idx;
            min_distance = min_value;
        end
    end
    mst(i) = closest_spike;
    mst_distances(i) = min_distance;
    for j = 1:i-1
        distances(mst(j), closest_spike) = Inf;
        distances(closest_spike, mst(j)) = Inf;
    end
end

% Calculates cluster centers
threshold = mean(mst_distances) + std(mst_distances);
cluster = mst(1);
cluster_centers = [];
for i = 2:length(mst_distances)
    if mst_distances(i) < threshold
        cluster = [cluster mst(i)];
    elseif length(cluster) > size(spike_features, 2)/6
        cluster_centers = [cluster_centers mean(spike_features(:, cluster), 2)];
        cluster = [];
    end
end

% Performs k-means clustering
if isempty(cluster_centers)
    spike_classes = zeros(1, size(spikes, 2));
    fprintf(’Could␣not␣initialize␣any␣class␣center!\n’);
else
    spike_classes = (litekmeans(spike_features’, [], ’start’, cluster_centers’))’;

    % Class Mapping
    intersectMatrix = zeros(max(ref_classes), max(spike_classes));
    for i = 1:size(spikes, 2)
        if ref_classes(i) ~= 0
            intersectMatrix(ref_classes(i), spike_classes(i)) =
                           intersectMatrix(ref_classes(i), spike_classes(i)) + 1;
        end
    end

    permut = max(ref_classes)+1:max(ref_classes)+max(spike_classes);
    for i = 1:min([max(ref_classes) max(spike_classes)])
        [~, idx] = max(intersectMatrix(:));
        [r c] = ind2sub(size(intersectMatrix), idx);
        permut(c) = r;
        intersectMatrix(r, :) = -Inf*ones(1, size(intersectMatrix, 2));
        intersectMatrix(:, c) = -Inf*ones(size(intersectMatrix, 1), 1);
    end
    spike_classes = permut(spike_classes);
end
```

APPENDIX B LOGIC SYNTHESIS SCRIPTS

B.1 Timing Constraints

```
# Define clock frequencies
create_clock -period 41667 -name sampling_clk [get_port "samplingClk"]
create_clock -period 15625 -name compression_clk [get_port "compressionClk"]

# Define input and output delays
set samplingClk_driven_inputs [get_port "din"]
set compressionClk_driven_inputs [get_port "rst"]
set_input_delay -clock sampling_clk 2 $samplingClk_driven_inputs
set_input_delay -clock compression_clk 2  $compressionClk_driven_inputs
set_output_delay -clock compression_clk 2 [all_outputs]

# Define clock setup and hold uncertainty
set_clock_uncertainty 0.5 -setup [all_clocks]
set_clock_uncertainty 0.5 -hold [all_clocks]

# Define ideal networks
set_ideal_network [get_ports samplingClk]
set_ideal_network [get_ports compressionClk]
set_dont_touch_network [all_clocks]
set_input_transition -max 0.5 [all_inputs]
set_drive 0 {rst samplingClk compressionClk}

# Define operating conditions
set_operating_conditions -max slow_1_62V_70C -max_library D_CELLSL_MOSLP_slow_1_62V_
set_wire_load_mode enclosed
```

B.2 Logic Synthesis Script

```
# Include file with auxiliary RTL Compiler commands and variables
include load_etc.tcl

# Include setup script
include ./scripts/av_setup.tcl

# General script constants
set SYN_EFF high
set MAP_EFF high
set DATE [clock format [clock seconds] -format "\%b\%d-\%T"]
set VCD_FILE ./CMP_AV.vcd

# Configurate standard-cells naming
set map_fancy_names 1
```

```
# Print use statistics during optimization
set iopt_stats 1

# Synthesis results directories
set _OUTPUTS_PATH outputs_${DATE}
set _LOG_PATH logs_${DATE}
set _REPORTS_PATH reports_${DATE}

# Define standard-cells library and HDL/RTL paths
set_attribute lib_search_path ${LIBRARY_PATH}/
set_attribute script_search_path ${SCRIPT_PATH}/
set_attribute hdl_search_path ${HDL_PATH}/
set_attribute information_level 9 /

# Specify the standard-cells library
set_attribute library $LIBRARY /

# Specify .lef files for physical layout estimation
set_attribute lef_library $LEFS /
set_attribute interconnect_mode ple /

# Activate incremental optimization for total negative slack
set_attribute tns_opto true /

# Activate clock gating technique
set_attribute lp_insert_clock_gating true /
# Increase power analysis effort
set_attribute lp_power_analysis_effort high /
set_attribute hdl_track_filename_row_col true /

# Load HDL/RTL code
read_hdl -vhdl $RTL

# Analyze and elaborate analyze RTL data structure
elaborate $DESIGN
# Show problems, if any
check_design -unresolved -all

# Load project constraints file
read_sdc -stop_on_errors $SDC_CONSTRAINTS_FILE

# Create synthesis results directories
if {![file exists ${_OUTPUTS_PATH}]} {
        file mkdir ${_OUTPUTS_PATH}
}
if {![file exists ${_LOG_PATH}]} {
        file mkdir ${_LOG_PATH}
}
if {![file exists ${_REPORTS_PATH}]} {
        file mkdir ${_REPORTS_PATH}
}

# Show preliminary design timing analysis
report timing -lint

# Define cost groups
rm [find /designs/* -cost_group *]
```

```
if {[llength [all::all_seqs]] > 0} {
        define_cost_group -name I2C -design $DESIGN
        define_cost_group -name C2O -design $DESIGN
        define_cost_group -name C2C -design $DESIGN
        path_group -from [all::all_seqs] -to [all::all_seqs] -group C2C -name C2C
        path_group -from [all::all_seqs] -to [all::all_outs] -group C2O -name C2O
        path_group -from [all::all_inps] -to [all::all_seqs] -group I2C -name I2C
}

foreach cg [find / -cost_group *] {
        report timing -cost_group [list $cg] > \
                                              $_REPORTS_PATH/${DESIGN}_pretim.rpt
}

# Activate power optimizations
set_attribute max_leakage_power 0.0 "/designs/$DESIGN"
set_attribute max_dynamic_power 0.0 "/designs/$DESIGN"
set_attribute lp_power_optimization_weight 1 "/designs/$DESIGN"

# Synthesize RTL to generic logic gates
synthesize -to_generic -eff $SYN_EFF
# Synthesize generic netlist to netlist mapped to the standard-cells library
synthesize -to_mapped -eff $MAP_EFF -no_incr

# Perform incremental synthesis
synthesize -to_mapped -eff $MAP_EFF -incr

# Load switching activity
read_vcd -static -vcd_module uut $VCD_FILE

# Generate timing analysis for each cost group
foreach cg [find / -cost_group -null_ok *] {
        report timing -cost_group [list $cg] > \
                           $_REPORTS_PATH/${DESIGN}_[basename $cg]_post_incr.rpt
}

# Generate clock gating, power, area, and gates reports
report clock_gating > $_REPORTS_PATH/${DESIGN}_clockgating.rpt
report power -depth 0 > $_REPORTS_PATH/${DESIGN}_power.rpt
report area > $_REPORTS_PATH/${DESIGN}_area.rpt
report gates -power > $_REPORTS_PATH/${DESIGN}_gates_power.rpt

# Write the final netlist, timing informationad, and updated constraints file
write_hdl > ${_OUTPUTS_PATH}/${DESIGN}_m.hvsyn
write_sdf -prec 3 -edges check_edge > ${_OUTPUTS_PATH}/${DESIGN}.sdf
write_sdc > ${_OUTPUTS_PATH}/${DESIGN}_m.sdc

# Write necessary information for the physical layout
write_encounter design $DESIGN -basename ../fe/${DESIGN}

# End of synthesis
file copy [get_attr stdout_log /] ${_LOG_PATH}/.
exit
```

APPENDIX C ORIGINAL WORK PROPOSAL

Circuito de Baixa Potência para a Detecção e Compressão de Impulsos Nervosos em ICMs

Ângelo C. Lapolli[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

aclapolli@inf.ufrgs.br

Abstract. *Within brain-machine interface systems, nodes from cortically implanted wireless sensor networks for capturing neural signals have a low power budget due their proximity from the biological tissue. Aiming the reduction of energy dissipation, data compression methods are used in order to decrease nodal transmission rate; still, this compression must not preclude signal decoding. In this context, this paper proposes the basis of a low-power circuit for neural spike detection and compression using compressed sensing method.*

Resumo. *Em sistemas de interface cérebro-máquina, os nodos de redes de sensores sem fio implantados intracorticalmente para a captura dos sinais nervosos possuem fortes restrições de potência devido a sua proximidade ao tecido biológico. Com o objetivo de reduzir a energia dissipada, aplicam-se métodos de compressão de dados visando diminuir a taxa de transmissão desses nodos; no entanto, tal compressão não deve impossibilitar a decodifiação dos sinais. Nesse contexto, este artigo propõe as bases de um circuito de baixa potência para a detecção e a compressão de impulsos nervosos utilizando o método de amostragem compressiva.*

1. Introdução

Os avanços na área de microeletrônica no sentido da diminuição do consumo de potência e do aumento da capacidade de integração lógica dos circuitos vêm incentivando o desenvolvimento de Redes de Sensores Sem Fio (RSSFs), essas exploram informações capturadas de pontos distribuídos no espaço para diferentes aplicações. Assim como muitas das tecnologias atuais, as RSSFs surgiram inicialmente no meio militar e foram sendo gradualmente adotadas para fins civis. Atualmente, elas podem servir, por exemplo, para o monitoramento de condições climáticas, para o controle de tráfego rodoviário e para a automação industrial e doméstica [Stankovic 2011]. Mais recentemente, na área médica, é objeto de intenso estudo a sua aplicação para o acompanhamento de sinais vitais de pacientes que sofrem de distúrbios cardíacos ou neurológicos [Abdaoui 2013] e para a restauração de funções neuromotoras no âmbito de interfaces cérebro-máquina (ICMs) [Sevillano 2009].

O estudo realizado aqui insere-se nesse último contexto, onde as ICMs detectam e decodificam sinais nervosos visando o controle de um dispositivo externo, geralmente próteses robóticas tais como cadeiras de rodas ou membros mecânicos. A captura dos sinais nervosos é realizada pelo implante intracortical de matrizes de microeletrodos associadas a um sistema de processamento e transmissão de dados que compõem uma RSSF; dessa forma, evita-se a passagem de cabos através do crânio proporcionando melhor usabilidade ao usuário. A decodificação desses sinais consiste na realização da classificação de impulsos nervosos, isto é, a sua detecção e a identificação de seus neurônios de origem.

Os nodos dessa RSSF possuem fortes restrições de potência, já que a sua área é limitada pela base dos microeletrodos e a sua densidade de potência é restringida de forma a não danificar o tecido biológico [Seese 1998]. Assim como afirmado em [Chen 2012], o custo energético da transmissão sem fio dos dados coletados é muito superior ao de qualquer outra das funções nodais (conversão analógico-digital, detecção de impulsos, etc). Sendo assim, a redução da taxa de transmissão por métodos de compressão de dados é recentemente objeto de intenso estudo buscando o respeito às restrições de potência.

Sob essa perspectiva, a simplicidade do método de amostragem compressiva ou *Compressed Sensing* (CS) [Donoho 2006] motiva o estudo da sua viabilidade em ICMs. Em [Chen 2012], o uso de CS é proposto sobre sinais biológicos genéricos esparsamente representáveis em um determinado domínio, é então proposta a compressão do sinal capturado em sua totalidade. Especificamente para o caso dos sinais nervosos, foi demonstrado em [Coppa 2012] que CS pode ser utilizado para a compressão individual de cada impulso nervoso detectado mantendo-se a capacidade de classificá-los. Com esta abordagem, a taxa de compressão aumenta resultando em uma maior redução do consumo total de energia do sistema.

Baseado sobre o estudo realizado em [Coppa 2012], este trabalho tem como objetivo analisar técnicas de fabricação de circuitos de baixa potência através da concepção de uma solução em hardware para a detecção, a extração e a compressão de impulsos nervosos por amostragem compressiva. Na seção seguinte, são descritos os fatores fisiológicos que definem as condições de contorno do sistema e inspiram as motivações para a sua concepção. Em seguida, apresentam-se trabalhos relacionados a esse campo de estudo a partir dos quais são explicadas técnicas dedicadas a realização das diferentes etapas de classificação de impulsos nervosos e é apresentado o método de amostragem compressiva. Sob esse suporte, em um terceiro momento, é especificado o circuito a ser projetado, e, na seção subsequente, são apresentados resultados preliminares obtidos por simulação. Finalmente, é apresentado um cronograma para a continuação do estudo e são dadas considerações finais.

2. Contexto e Motivação

2.1. Sinais nervosos: características e método de captura

O sistema nervoso humano é o responsável pela comunicação intercelular através da transmissão de sinais elétricos entre as diferentes partes do corpo. Ele possui como unidade fundamental células denominadas neurônios, que, com o suporte de células auxiliares, formam uma rede complexa de caminhos nervosos que interliga as diferentes estruturas fisiológicas do organismo. Através dessa rede, propagam-se sinais relativos a informações motoras e sensoriais a serem convertidas em reações mecânicas e bioquímicas necessárias à manutenção da vida e à interação harmônica com o ambiente externo.

A fim de compreender a natureza desses sinais, é preciso analisar o processo interno aos neurônios que permite a sua geração [Hall 1998]. De forma simplificada, esses sinais são compostos por impulsos elétricos resultantes da transferência de íons através da membrana celular neuronal cuja permeabilidade é alterada em função do recebimento de neurotransmissores, isto é, substâncias emitidas pelas células nervosas adjacentes. Quando em repouso, um neurônio apresenta uma diferença de potencial através da sua membrana de aproximadamente -70mV; se, por influência da ação dos neurotransmissores, esse valor atinge uma tensão limite da ordem de -65mV, o neurônio dispara um impulso elétrico. Esse impulso é determinado pela mudança abrupta do potencial, o qual atinge picos de aproximadamente 40mV e retorna rapidamente ao estado de repouso (Figura 1). Devido às diferenças na forma de cada neurônio, esses impulsos assumem curvas características que permitem distingui-los quanto à sua célula de origem, etapa fundamental para a decodificação realizada pelas ICMs.

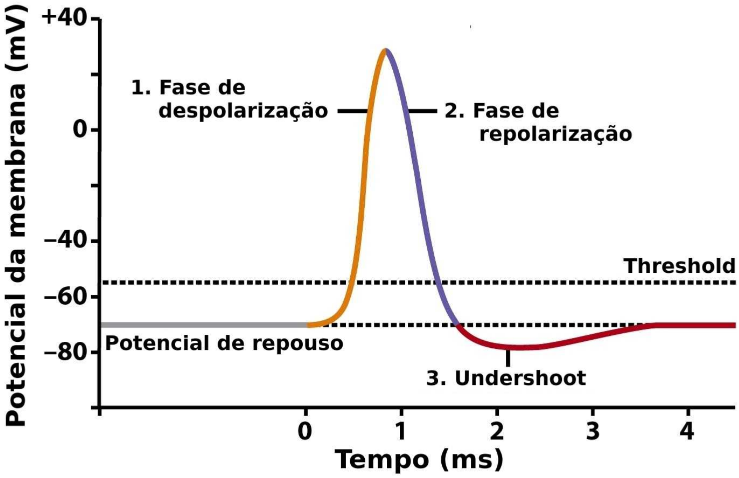


Figura 1. Adaptada de [Freeman 2005] - Fases de um Impulso Neuronal

O procedimento médico padrão de captura desses sinais, a eletroencefalografia a partir de eletrodos aplicados sobre o couro cabeludo, apresenta importantes limitações relativas à sua resolução espacial [Friedman 2009]. Por outro lado, considerando que esses sinais são atenuados pelas estruturas de proteção cerebral, obtêm-se dados mais precisos pelo implante intracortical de microeletrodos. Nesse caso, entretanto, a proximidade desses sensores elétricos ao tecido biológico interno lhes impõe fortes restrições de dissipação de potência. Conclui-se, a partir de [Seese 1998], que a densidade de potência desses dispositivos deve ser inferior a 800 µW/mm^2 para que as células não sejam danificadas. Além disso, circuitos associados aos microeletrodos têm sua área restringida à área da base dos sensores que é da ordem de 0.01 mm^2 [Normann 2003].

2.2. Motivação

A coordenação das atividades funcionais do organismo humano, as informações referentes aos órgãos sensoriais e as ordens para movimentação dos músculos são realizadas através da transmissão dos impulsos nervosos. Sendo assim, a detecção e a interpretação desses sinais aliadas a sistemas de processamento inteligente pode ser de extrema utilidade não somente para a restauração de funções sensoriais e motoras, como também para o estudo e a monitoração de distúrbios neurológicos.

O acesso aos sinais nervosos com maior relação de sinal-ruído proporcionado pela captura a partir de microeletrodos implantados intracorticalmente facilita o monitoramento das condições neurológicas de pacientes que sofrem de distúrbios desse tipo [Friedman 2009]. No caso da epilepsia, por exemplo, convulsões podem ser previstas para alertar o paciente e seus acompanhantes a tomarem precauções de segurança. Outra possibilidade é a de servir como guia para a administração de medicamentos no caso de coma induzido. Além desse tipo de monitoramento, tal técnica pode servir a estudar doenças cujas causas e tratamento são até então desconhecidas.

No âmbito da restauração de funções sensoriais e motoras, um sistema de interface cérebro-máquina pode permitir a comunicação da rede nervosa com próteses robóticas realizando funções de membros perdidos. Exemplos desse tipo de aplicação incluem o controle de cadeiras de rodas, de braços mecânicos e de computadores pessoais.

Assim como exemplificado acima, existe uma gama extensa de aplicações possíveis a partir de um sistema de comunicação cérebro-máquina utilizando o implante intracortical de microeletrodos. Todas elas envolvem o processamento dos sinais capturados e, por consequência, técnicas de classificação de impulsos nervosos. Carece observar, no entanto, que a implementação de tais sistemas exige um procedimento cirúrgico invasivo; sendo, portanto, aconselhadas a pacientes cuja condição limitante possa ser consideravelmente atenuada.

3. Trabalhos Relacionados

A seguir, serão analisados algoritmos adaptados à implementação em hardware que realizem cada uma das etapas de classificação de impulsos nervosos: detecção, extração, compressão e agrupamento. Observe que esses algoritmos são discutidos sob o pressuposto de um sinal de entrada digital por esse ser a forma de implementação escolhida para a realização do sistema. Ao final desta seção, são apresentadas soluções similares ao sistema a ser proposto aqui.

3.1. Detecção e Extração de Impulsos Nervosos

O primeiro passo do procedimento de classificação consiste no isolamento dos impulsos nervosos a partir do sinal contínuo recebido dos microeletrodos, no qual deve-se primeiramente detectar os disparos distinguindo-os do ruído de fundo e, em seguida, extrair as amostras que os representem. Os algoritmos para a realização de tal detecção valem-se geralmente da análise da amplitude, da energia e do espectro de frequência do sinal original amplificado e filtrado por um filtro passa-banda. Quando da extração, procura-se alinhar os impulsos em função da sua amplitude máxima em uma janela de tempo de duração pré-determinada.

Em [Gibson 2008], analisa-se a eficiência de três algoritmos de detecção de impulsos nervosos sob a perspectiva de uma implementação em hardware. Os algoritmos estudados então são: *Absolute Value* (AV), *Nonlinear Energy Operator* (NEO) e *Stationary Wavelet Transform Product* (SWTP). Desses, os dois primeiros são aqui considerados por apresentarem baixa complexidade de cálculo, estando assim em conformidade com o objetivo de minimizar a potência dissipada.

A técnica de detecção conhecida por *Absolute Value* (AV) baseia-se na determinação de um valor de amplitude limite Thr dado por:

$$Thr=4\cdot\sigma_N \text{ com } \sigma_N=median\left\{\frac{|x(n)|}{0.6745}\right\} ,$$

onde $x(n)$ é uma amostra do sinal no tempo n e σ_N é uma estimativa do desvio padrão do ruído. Dessa forma, um impulso nervoso é considerado presente sempre que uma amostra é superior a esse valor limite. O fator multiplicativo 4 pode ser alterado visando otimizar a proporção de impulsos não detectados e erros de detecção. É concluído em [Gibson 2008] que este método, apesar de extremamente simples, apresenta uma degradação considerável da sua capacidade de detecção em função da diminuição da relação sinal-ruído.

A técnica NEO, por sua vez, mostra-se mais adaptada à detecção de impulsos nervosos em sinais com relação de sinal-ruído inferior. Essa consiste na definição de um valor NEO ψ dado, em tempo discreto, por:

$$\psi[x(n)]=x^2(n)-x(n+1)\cdot x(n-1) .$$

O NEO assume valores elevados sempre que a potência instantânea ou a frequência local do sinal é alta, o que caracteriza a presença de um impulso nervoso. Analogamente à técnica AV, esse valor é constantemente calculado e comparado a um limite dado pela sua média escalada a partir de N amostras do sinal na seguinte forma:

$$Thr=C\frac{1}{N}\sum_{n=1}^{N}\psi[x(n)] ,$$

onde a escala C pode ser escolhida experimentalmente de forma a otimizar a detecção.

Uma vez que um impulso nervoso é detectado, segue a extração das amostras relevantes a representá-lo. Inicialmente define-se a duração da janela de tempo (número de amostras) a ser extraída; essa deve ser minimizada procurando tanto a compressão de dados, quanto a diminuição do custo computacional de seu processamento, mas sem evidentemente prejudicar o processo de classificação. Em seguida, alinha-se as amostras do suposto impulso nervoso tomando a sua máxima amplitude como referência, a qual pode ser encontrada após um processo de interpolação buscando evitar erros devido a uma baixa taxa de amostragem [Quiroga 2004].

3.2. Compressão

A partir dos impulsos nervosos detectados, o procedimento padrão de classificação segue com a extração das suas informações que permitirão distiguir os seus neurônios de origem. Esta etapa geralmente diminui a quantidade de dados a ser armazenada por impulso e, portanto, confunde-se com uma compressão. Assim como previamente mencionado, essa é uma alternativa para reduzir a taxa de dados a ser transmitida e, consequentemente, reduzir a potência total do sistema. Nesse contexto, é adotado aqui o método de amostragem compressiva cujo desempenho relativo à classificação é avaliado em [Coppa 2012]. Outras técnicas utilizadas nesta etapa incluem: transformada discreta *wavelets* [Quiroga 2004], análise de componentes principais e diferenciação discreta [Karkare 2011].

A amostragem compressiva foi concebida com a finalidade de permitir a reconstrução de sinais esparsos a partir de representações com baixas taxas de amostragem. Este método consiste em representar o sinal esparso original $x \in \mathbb{R}^N$ por uma projeção em um subespaço dada por:

$$y = \Phi \cdot x \quad \text{com} \quad \Phi \in \mathbb{R}^{m \times N} \quad \text{e} \quad m > N \quad ,$$

onde Φ é chamado de matriz de amostragem e $y \in \mathbb{R}^m$ é o sinal comprimido, resultando N/m como taxa de compressão. Apesar de apresentar baixa complexidade de compressão, uma reconstrução eficiente do sinal original envolve um problema complexo de minimização. Não obstante, [Coppa 2012] mostra ser possível recuperar a informação necessária às aplicações de ICMs diretamente a partir dos impulsos nervosos individualmente comprimidos. Usando-se uma matriz de amostragem binária e aleatória cuja a probabilidade de cada estado (± 1) vale 0.5 , a classificação mostra-se viável com uma taxa de compressão de 5.33 considerando $N = 32$ (número de amostras a $24\,kHz$ representando um único impulso nervoso) e $m = 6$.

3.3. Agrupamento

Algoritmos de agrupamento servem a caracterizar dados segundo certo certo critério de semelhança. Para fins de classificação de impulsos nervosos, os algoritmos que vêm sendo recentemente utilizados fundamentam-se sobre os métodos que envolvem *nearest neighbor search*, isto é, o agrupamento a partir de representações em um domínio multidimensional Euclideano, onde a distância entre pontos serve como critério de semelhança. Exemplos dessa abordagem incluem os algoritmos de *super-paramagnetic clustering* utilizado sobre impulsos nervosos em [Quiroga 2004] e de *k-means clustering*, o qual é aplicado em [Coppa 2012] da maneira a ser descrita a seguir. Apesar dessa etapa da classificação não estar prevista para implementação no circuito a ser proposto, ela é importante para que se possa analisar o impacto da compressão dos impulsos sobre o desempenho de classificação do sistema como um todo.

K-means clustering é um algoritmo iterativo que atribui cada ponto do conjunto de dados ao grupo de centro mais próximo em distância Euclideana; em seguida, os centros dos grupos são atualizados pela média de todos pontos pertencentes a ele. Esse procedimento é repetido até que não ocorra mais alterações no agrupamento. O procedimento aplicado em [Coppa 2012] envolve a inicialização automática dos centros dos grupos pela construção de uma árvore de extensão mínima, essa é inicializada com um ponto escolhido randomicamente do conjunto de dados e é então construída incrementalmente pela inserção do ponto mais próximo de qualquer já pertencente à árvore. A sequência das distâncias entre cada ponto inserido é armazenada em memória e os grupos são formados pela definição de um valor limite dessas distâncias; os centros de cada grupo resultante são então utilizados para a iniciar o algoritmo de *k-means* *clustering*.

3.4. Soluções Similares

Com o objetivo de estabelecer parâmetros de comparação ao circuito a ser projetado, tomar-se-ão como referência três soluções da literatura de mesma finalidade. [Chen 2012] propõe um circuito que aplica amostragem compressiva em RSSFs detectando sinais biológicos genéricos, [Olsson III 2005], por sua vez, apresenta um método de detecção de impulsos como única estratégia de redução da taxa de transmissão, e [Karkare 2011] propõe um circuito de detecção e compressão de impulsos.

Em [Chen 2012], é apresentado uma arquitetura para a realização de amostragem compressiva em tempo real sobre dados capturados por microeletrodos. Por considerar sinais biológicos genéricos, a etapa de detecção de impulsos não é realizada. Assume-se então uma matriz de amostragem binária pseudo-aleatória. Obtém-se como parâmetros do circuito uma potência de 1.9 µW e uma área de 0.09 mm² sob um processo CMOS 90 nm. A taxa de compressão resultante vale 90 %.

Outra solução em [Olsson III 2005] propõe um circuito para a detecção de impulsos nervosos em 32 canais pela técnica de *absolute value*. Este, fabricado em um processo CMOS 500 nm, ocupa um área média de 0.11 mm² por canal e dissipa 75 µW. Segundo os autores, atinge-se uma taxa de compressão de 92 %.

Por fim, [Karkare 2011] sugere um circuito que realiza a detecção de impulsos pelo método de *nonlinear energy operator* seguido por diferenciação discreta para a extração de suas características. O circuito, capaz de suportar até 64 canais, apresenta, considerando um único canal, uma área de 0.06 mm² e dissipa 2.03 µW. Nesse caso, a taxa de compressão estimada é de 91.25 %.

4. Proposta

Tendo analisado as bases teóricas e as soluções existentes para o problema da realização de classificação de impulsos nervosos no caso de implante intracortical de matrizes de microeletrodos, especificar-se-á, a seguir, a estrutura geral do circuito proposto com o objetivo de abordar essa questão. Ao final, apresentar-se-á a metodologia de projeto a ser utilizada.

4.1. Especificação do Circuito

Sob uma perspectiva global, o sistema de interface cérebro-máquina assumido baseia-se sobre uma rede de microeletrodos e circuito associado para capturar sinais nervosos, detectar impulsos, os comprimir e os transmitir a uma central encarregada de os decodificar para o controle de um dispositivo externo. O esquemático geral desse sistema é apresentado pela Figura 2, onde o circuito a ser projetado insere-se no bloco de detecção e compressão de impulsos nervosos.

De maneira mais específica, no contexto do circuito a ser proposto, considerar-se-á um sinal de entrada previamente filtrado por um filtro elíptico passa-banda (300 Hz – 3 kHz) e digitalizado por um conversor a uma taxa de amostragem de 24 kHz e quantização uniforme. O sinal de saída, por sua vez, deverá ser os impulsos nervosos detectados e individualmente comprimidos por amostragem compressiva. A Figura 3 ilustra a estrutura modular geral escolhida.

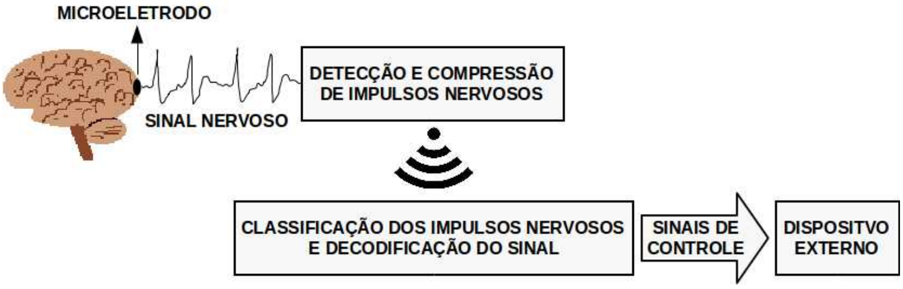

MICROELETRODO
SINAL NERVOSO
DETECÇÃO E COMPRESSÃO DE IMPULSOS NERVOSOS
CLASSIFICAÇÃO DOS IMPULSOS NERVOSOS E DECODIFICAÇÃO DO SINAL
SINAIS DE CONTROLE
DISPOSITVO EXTERNO

Figura 2. Esquema Geral do Sistema de Interface Cérebro-Máquina

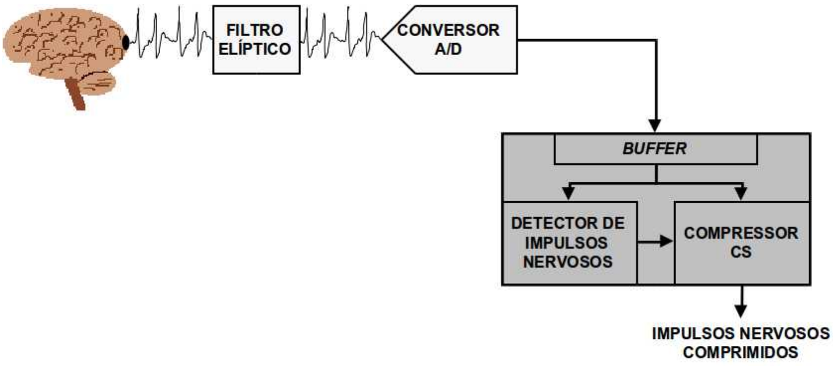


Figura 3. Esquema modular do circuito proposto, este representado em cinza escuro.

O sinal de entrada é primeiramente armazenado em um *buffer*, o qual é acessível por dois outros módulos: um responsável pela detecção dos impulsos e outro encarregado da compressão; sempre que um impulso for detectado, o primeiro deverá notificar o segundo para que este entre em operação. Além dessa notificação, o módulo de detecção poderá também enviar ao compressor dados relativos ao alinhamento das amostras dos impulsos nervosos, as quais serão lidas diretamente do *buffer*.

A partir de análise prévia considerando desempenho e complexidade de implementação, deverá ser escolhido um dos método de detecção de impulsos nervosos dentre *absolute value* e *nonlinear energy operator*. Independentemente do método, o detector deverá ser autoconfigurável em função do sinal recebido tendo cada impulso será representado por 32 amostras.

Quanto ao módulo de compressão CS, utilizar-se-á uma matriz de amostragem com 6 linhas e 32 colunas cujos elementos serão binários (± 1) e randomicamente distribuídos. O tempo da realização de uma compressão deverá se limitar ao intervalo até a potencial chegada do próximo impulso. O sinal de saída deve ser inicialmente zerado e atualizado ao fim de cada compressão.

Por fim, a fim de evitar danos ao tecido biológico, a densidade de potência do circuito deverá ser inferior à 800 μW/mm² e a sua área da ordem de 0.01 mm² de forma a poder ocupar junto com o filtro e o conversor analógico-digital a base de um único microeletrodo [Normann 2003].

4.2. Metodologia de Projeto

O projeto do circuito acima descrito será dividido nas seguintes fases: modelagem matemática, descrição ao nível RTL (*Register Transfer Level*), síntese lógica e síntese física.

A modelagem matemática do sistema, cuja metodologia e resultados preliminares são apresentados na próxima seção, servirá a avaliar o desempenho do procedimento de classificação de impulsos nervosos escolhido. Pela análise dos seus resultados, dever-se-á escolher os parâmetros dos algoritmos de detecção e estudar a influência das matrizes de amostragem buscando sempre uma implementação otimizada em hardware. Ao fim, os impulsos comprimidos obtidos nessa fase servirão como resultados de comparação para a validação do circuito.

Em seguida, tendo definido os parâmetros matemáticos que envolvem a solução proposta, será concebida a arquitetura do circuito a qual será descrita textualmente em VHDL (*VHSIC Hardware Description Language*) ao nível RTL. A validação de tal descrição dar-se-á com o auxílio de uma ferramenta de simulação.

As sínteses lógica e física serão realizadas com a utilização de técnicas de fabricação de cricuitos integrados de baixa potência. Estas técnicas serão analisadas e avalidas a partir dos resultados obtidos. Finalmente, estimar-se-ão a área e a potência do circuito final.

5. Resultados Preliminares

Visando avaliar o desempenho potencial do circuito no que diz respeito ao seu desempenho na realização da classificação, modelou-se matematicamente a sua operação. Avaliaram-se primeiramente o desempenho dos métodos de detecção de impulsos escolhidos e, em seguida, analisou-se a perda de capacidade de classificação causada pela compressão do sinal original. Em todos os casos, utilizaram-se os sinais nervosos simulados publicados por [Quiroga 2004] nos quais o nível de ruído, determinado a partir do seu desvio padrão, foi atribuído por normalização a 0.05, 0.1, 0.15 e 0.2 relativo a amplitude máxima dos impulsos nervosos. Sobre estes sinais, assumiu-se a aplicação de um filtro elíptico e de um conversor analógico-digital na forma determinada pela especificação do circuito.

5.1. Avaliação dos Métodos de Detecção de Impulsos

Os métodos AV e NEO foram aplicados sobre os sinais nervosos simulados descritos acima; no caso do NEO, utilizou-se $C=8$ e N tal a considerar o sinal completo. Em seguida, os impulsos detectados, cada um contendo 32 amostras, foram alinhados de forma a ter seu máximo valor de amplitude na sua décima segunda amostra.

A Figura 4 ilustra a forma dos impulsos nervosos originais e dos detectados junto com as suas representações em termos das suas duas primeiras componentes principais. Tais resultados foram obtidos a partir de um sinal contendo 500 *spikes* e com nível de ruído de 0.1. Nessas representações, as classes de cada impulso são identificadas por cores diferentes.

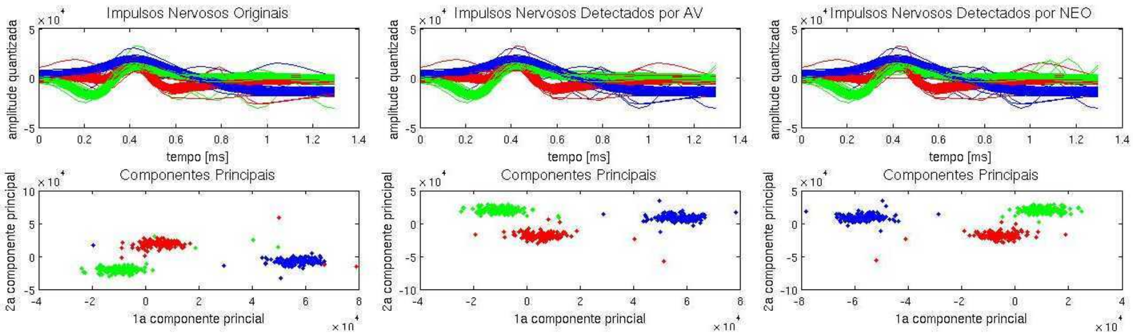
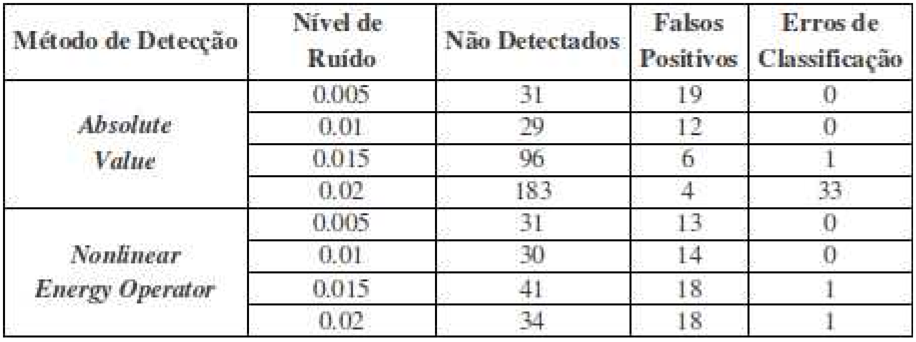


Figura 4. Impulsos Nervosos Extraídos de um Sinal com Nível de Ruído de 0.1

A Tabela 1 demonstra resultados relativos ao desempenho da detecção e a seu impacto na classificação. Constata-se que para baixos níveis de ruído, ambos métodos de detecção avaliados apresentam bons resultados; com o aumento da intensidade de ruído, no entanto, o método NEO demonstra-se melhor adaptado conforme já observado por [Gibson 2008].

Tabela 1. Desempenho de Detecção de Impulsos Nervosos para um Total de 500 Impulsos Originais



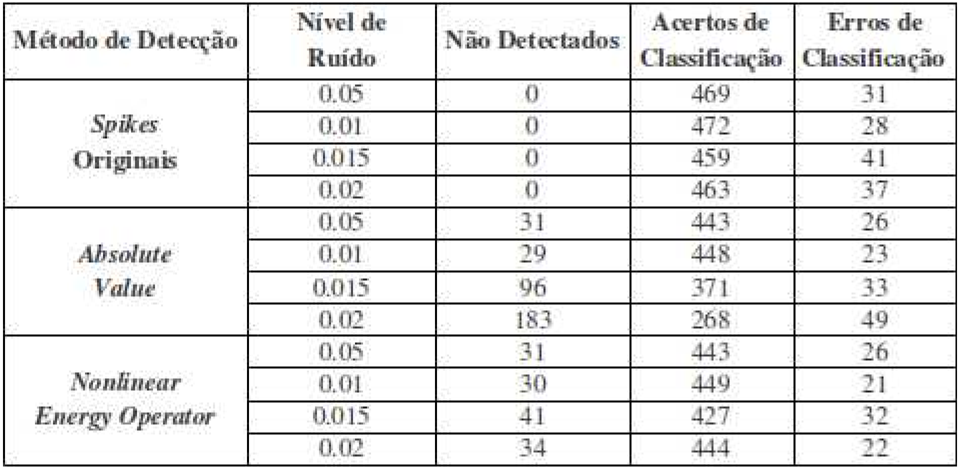
Método de Detecção	Nível de Ruído	Não Detectados	Falsos Positivos	Erros de Classificação
Absolute Value	0.005	31	19	0
	0.01	29	12	0
	0.015	96	6	1
	0.02	183	4	33
Nonlinear Energy Operator	0.005	31	13	0
	0.01	30	14	0
	0.015	41	18	1
	0.02	34	18	1

5.2. Avaliação do Método de Amostragem Compressiva

O procedimento de avaliação do impacto da etapa de compressão sobre o desempenho de classificação iniciou pela determinação da matriz de amostragem a ser utilizada pelo método de amostragem compressiva. Tomaram-se como critérios para a definição dessa matriz a distância entre os centros de cada grupo de impulsos e a distância média entre cada impulso e o centro de seu grupo, tudo isso na representação em componentes principais dos impulsos comprimidos. Em seguida, aplicou-se a compressão sobre os impulsos nervosos originais e sobre os detectados pelo método AV e pelo método NEO utilizando sinais com diferentes níveis de ruído. Posteriormente, esses foram classificados no espaço comprimido e compararam-se as classes obtidas às originais.

A Tabela 2 apresenta o número de acertos e erros de classificação a partir dos impulsos comprimidos. O desempenho da classificação sobre os impulsos detectados pelo método AV mostra-se prejudicado com o aumento do nível de ruído. No entanto, ao avaliarmos somente o impacto da compressão, a taxa média de erros de classificação relativa ao número total de impulsos originais (500) demonstra-se de apenas 6.15 %.

Tabela 2. Desempenho da Classificação de Impulsos Nervosos no Espaço Comprimido para um Total de 500 Impulsos Originais

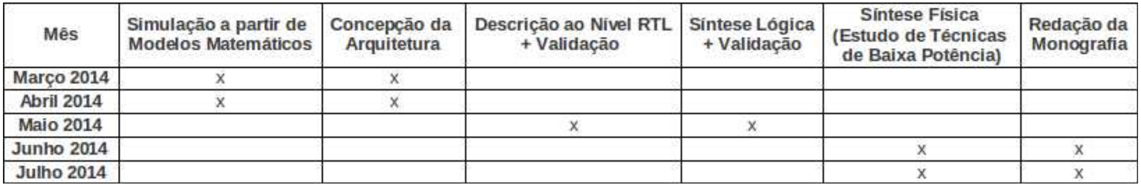


Método de Detecção	Nível de Ruído	Não Detectados	Acertos de Classificação	Erros de Classificação
Spikes Originais	0.05	0	469	31
	0.01	0	472	28
	0.015	0	459	41
	0.02	0	463	37
Absolute Value	0.05	31	443	26
	0.01	29	448	23
	0.015	96	371	33
	0.02	183	268	49
Nonlinear Energy Operator	0.05	31	443	26
	0.01	30	449	21
	0.015	41	427	32
	0.02	34	444	22

6. Cronograma

Fundamentados sobre esse artigo, a concepção do circuito e um estudo mais aprofundado da sua viabilidade de implementação serão realizados conforme o cronograma da Tabela 3. Por fim, as bases teóricas e a descrição do trabalho realizado serão compiladas na forma de uma monografia juntamente com os resultados e conclusões obtidas.

Tabela 3. Cronograma



Mês	Simulação a partir de Modelos Matemáticos	Concepção da Arquitetura	Descrição ao Nível RTL + Validação	Síntese Lógica + Validação	Síntese Física (Estudo de Técnicas de Baixa Potência)	Redação da Monografia
Março 2014	x	x				
Abril 2014	x	x				
Maio 2014			x	x		
Junho 2014					x	x
Julho 2014					x	x

7. Conclusão

No contexto de sistemas de interfaces cérebro-máquina, este artigo apresentou as bases de um circuito de baixa potência para a detecção e a compressão intracortical de impulsos nervosos. Esta solução procura permitir a classificação desses impulsos minizando erros e respeitando as restrições de potência impostas pela aplicação. Nessa perspectiva, foram apresentados os métodos conhecidos de resolução desse problema; dentre eles, alguns foram selecionados e tiveram o seu desempenho brevemente analisado. Os trabalhos subsequentes consistirão na concepção do circuito aqui especificado com o objetivo de estudar técnicas de fabricação de circuitos integrados de baixa potência.

Referências

J. A. Stankovic, A. D. Wood e T. He (2011), “Realistic Applications for Wireless Sensor Networks”, Theoretical Aspects of Distributed Computing in Sensor

Networks, pp. 835-863.

R. Abdaoui, M. Villegas e G. Baudoin (2013), “Potentiality of a low power 60 GHz transceiver approach for EEG/ECG WSN medical applications”, IEEE Faible Tension Consommation, Paris, França.

J. L. Sevillano, D. Cascado, D. Cagigas, S. Vicente, C. D. Luján e F. Díaz del Rio (2009), “A Real-Time Wireless Sensor Network for Wheelchair Navigation”, IEEE/ACS International Conference on Computer Systems and Applications, Rabat, Marrocos.

T. M. Seese, G. M. Saidel, C. R. Davies (1998), “Characterization of tissue morphology, angiogenesis and temperature in the adaptative response of muscle tissue to chronic heating”, Lab Invesigation, vol. 78, no. 12.

F. Chen, A. P. Chandrakasan e V. M. Stojanović (2012), “Design and Analysis of a Hardware-Efficient Compressed Sensing Architecture for Data Compression in Wireless Sensors, IEEE Journal of Solid-State Circuits, vol. 47, no. 3, pp. 744-756.

D. L. Donoho (2006), “Compressed Sensing”, IEEE Transactions on Information Theory, vol. 52, pp. 1289-1306.

B. Coppa, R. Héliot, O. Michel e D. David (2012), “Low-cost intracortical spiking recordings compression with classification abilities for implanted BMI devices”, 34th Annual International IEEE EMBS Conference, San Diego, Estados Unidos.

R. H. Hall (1998), “Neural Communication”, Neuroscience Modules, http://web.mst.edu/~rhall/neuroscience/01_fundamentals/neural_communication.pdf

S. Freeman (2005), “Biological Science, 2nd”, Pearson Prentice Hall.

D. Friedman, J. Claassen e L. J. Hirsch (2009), “Continuous Electroencephalogram Monitoring in the Intensive Care Unit”, International Anesthesia Research Society, vol. 109, no. 2, pp. 506-523.

R. A. Normann (2003), “Microfabricated electrode arrays for restoring lost sensory and motor functions”, 12th International Conference on Solid-State Sensors, Actuators and Microsystems, Boston, Estados Unidos.

S. Gibson, J. W. Judy e D. Markovi ć (2008), “Comparison of Spike-Sorting Algorithms for Future Hardware Implementation”, 30th Annual International IEEE EMBS Conference, Vancouver, Canada.

R. Q. Quiroga, Z. Nadasdy e Y. Ben-Shaul (2004), “Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering”, Neural Computation, no. 16.8, pp. 1661-1667.

R. H. Olsson III e K. D. Wise (2005), “A Three-Dimensional Neural Recording Microsystem With Implantable Data Compression Circuitry”, IEEE Journal of Solid-State Circuits, vol. 40, no. 12, pp. 2796-2804.

V. Karkare, S. Gibson e D. Marković (2011), “A 130-μW, 64-Channel Neural Spike-Sorting DSP Chip, IEEE Journal of Solid-State Circuits, vol. 46, no. 5, pp. 1214-1222.