Energy-Efficient Compressed Sensing Frameworks for the Compression of Electroencephalogram Signals

by

Simon Fauvel

B.Eng., McGill University, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

October 2013

Abstract

The use of wireless body sensor networks (WBSNs) is gaining popularity in monitoring and communicating information about a person's health. In such applications, the amount of data transmitted by the sensor node should be minimized. This is because the energy available in these battery-powered sensors is limited. In this thesis, we study the wireless transmission of electroencephalogram (EEG) signals. We propose novel, energy-efficient compressed sensing (CS) frameworks that take advantage of the inherent structure present in EEG signals (both temporal and spatial correlations) to efficiently compress these signals at the sensor node in WBSNs.

We first present a simple CS-based framework that is adapted to the EEG WBSN setting. We optimize the sparsifying dictionary and demonstrate that using a fixed sparse binary sensing matrix offers similar performances to optimal matrices while requiring far fewer computations.

We then add an energy-efficient Independent Component Analysis (ICA) preprocessing block to the simple CS framework to exploit the spatial correlations among EEG channels. We show that the proposed framework provides significant energy savings as compared to the state-of-the-art method. As well, for a fixed compression ratio, our system achieves similar normalized mean square error performance as the state-of-the-art method, which is better than that achieved by the simple CS framework.

We further improve on the energy performance of the framework by replacing the ICA preprocessing block by a simpler, correlations-based interchannel redundancy module and by using entropy coding. On the energy front, our proposed CS framework is up to 8 times more energy-efficient than the typical wavelet compression method. We also show that our method achieves a better reconstruction quality than the state-of-the art BSBL method. We further demonstrate that our method is robust to measurement noise and to packet loss, and that it is applicable to a wide range of EEG signal types.

We finally apply our CS framework to compress EEG signals in the context of a brain computer interface application and evaluate its impact on the performance of the system. We show that interesting energy savings can be realized at the cost of a mild decrease in classification accuracy.

Preface

This thesis presents the research conducted by Simon Fauvel, in collaboration with Dr. Rabab K. Ward. I hereby declare that I am the first author of this thesis. Chapters 3 to 5 are based on work that has been published or submitted for publication. Below are the publications related to this thesis.

C1 S. Fauvel, A. Agarwal, and R. Ward. Compressed Sensing and Energy-Aware Independent Component Analysis for Compression of EEG Signals. In *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, 2013.

J1 S. Fauvel and R. Ward. An Energy-Efficient Compressed Sensing Framework for the Compression of Electroencephalogram Signals. 2013 (submitted).

For both publications, I carried out the literature review, developed the frameworks, implemented them in software, carried out the simulations, analyzed the results, and wrote the manuscripts. In **C1**, A. Agarwal helped with the Matlab implementation of the proposed ICA algorithm.

Dr. Ward helped in formulating the research problem, supervised the direction of the research, and provided significant editorial comments and important suggestions for the organization of each manuscript.

Table of Contents

List of Tables

List of Figures

List of Acronyms

BCI	Brain Computer Interface
BSBL	Block-Sparse Bayesian Learning
BSBL-BO	Block-Sparse Bayesian Learning - Bounded Optimization
CDF	Cohen-Daubechies-Feauveau
CR	Compression Ratio
CS	Compressed Sensing
CSP	Common Spatial Patterns
DCT	Discrete Cosine Transform
EEG	Electroencephalogram
FICA	FastICA
FLOPS	Floating-Point Operations
GDP	Gross Domestic Product
IC	Independent Component
ICA	Independent Component Analysis
IID	Independent and Identically Distributed
IOC	Iteration of Convergence
LDA	Linear Discriminant Analysis
MMV	Multiple Measurements Vectors
NMSE	Normalized Mean Square Error
nD	n-Dimensional
PDF	Probability Density Function
RIP	Restricted Isometry Property
SNR	Signal-to-Noise Ratio
SOBI	Second-Order Blind Identification
SPIHT	Set Partitioning in Hierarchical Trees
WBSN	Wireless Body Sensor Network
xFICA	Cross-Product-Based FastICA

Acknowledgements

I would first like to thank my supervisor, Dr. Rabab Ward, for her continued guidance and support throughout my time at UBC, and for her patience as I was learning the ropes and figuring out this new environment. This whole journey was far from being smooth on many levels, and Dr. Ward's support was essential in bringing it to fruition. This research and thesis would not have been possible without her.

Thank you to my labmates for contributing to making my time at UBC pleasant. A special thanks to Tanaya and Mani, two wonderful colleagues, but more importantly two amazing friends. I will always remember and cherish the great times we had together.

Thank you to my friends back home, who I wish I could see more often. Thank you to my EWB family for providing constant inspiration and motivation, for helping me grow as a person, and for helping me maintain balance in my life (sort of).

And last but not least, a big thanks to my family for their never ending love and support. Given what we had to go through in the last few months, it's clear that I would not be here without you. Thank you for your continued support in my different ventures, even the more foolish ones. Knowing that you are always behind me is essential and is what has allowed me to persevere when motivation was low and when I was questioning myself.

This research was supported by the Qatar National Research Fund (QNRF No. NPRP 09-310-1-058), the Natural Sciences and Engineering Research Council of Canada (NSERC) through an Alexander Graham Bell Canada Graduate Scholarship (CGS-M), and the Fonds de recherche Nature et technologies (FQRNT) through a masters research scholarship (B1).

Chapter 1

Introduction

1.1 Telemedicine and Wireless Body Sensor Networks

Healthcare consumes a large part of the gross domestic product (GDP) of most countries, and the trend is going upward. In the last two decades, health expenditures as a percentage of GDP have steadily increased from 8.8% to 10.1% worldwide. The trend is more obvious in high income countries, where the same metric has gone from 9.5% to 12.0% during that period [6].

Most countries are also facing important increases in elderly population (defined as the total population above 65 years old) and in chronic disease patients. Between 1960 and 2012, the percentage of elderly population has gone from 5.1% to 7.8%, and the trend is predicted to accelerate in the decades to come [7, 45]. Chronic diseases are on the rise everywhere around the world and take up a significant portion of healthcare budgets [46]. Their impact can also be felt on the global economy (mainly due to productivity losses). In Canada, 58% of all annual healthcare spending went to treat chronic disease patients in 2010. It is estimated that Canada lost about 190 billion dollars in 2010 due to chronic diseases ($68 billion in direct health costs, and $122 billion in indirect costs) [44]. The financial impact of these increases is predicted to accelerate in the years and decades to come.

On top of the financial burden created by the increasing number of elderly and chronic disease patients, traditional healthcare is suffering from scalability issues. By definition, chronic disease patients require close monitoring of their condition over time, which puts significant pressure on health infrastructure. Traditional healthcare cannot provide the scalability required, as it relies on a physical, one-to-one relationship between the caregiver and the patient.

Cost-effective solutions are thus needed to mitigate these issues. One possibility is to enable patients to participate in their own treatment by giving them the technological tools necessary to monitor and remotely communicate their situation to caregivers. With recent advances in signal processing and very-low-power wireless communications, wireless body sensor networks (WBSNs) have gained popularity as a potential solution. The use of various sensors located on the patient's body allows WBSNs to measure and communicate different physiological signals (e.g. heart and brain activity) [40].

Such a setup has the potential to be cost-effective both for the patients and for the healthcare system as a whole. By giving the patients more control over their treatment, the burden on medical staff and infrastructures is reduced. The patients can also carry out most of their activities normally and avoid trips to a physical health infrastructure. Moreover, WBSNs give patients greater autonomy and ownership when it comes to taking care of their own health, which can lead to a better quality of life.

1.2 Electroencephalogram Signals

An important component of WBSNs is the study of the electrical brain activity. This is achieved via recording and analyzing the electroencephalogram (EEG) signals, using a collection of non-invasive wireless sensors located on a patient's scalp. On top of being non-invasive, EEG signals provide high temporal resolution, a desirable characteristic in many situations. This high temporal resolution comes at the expense of a reduced spatial resolution (determined by the number of sensors used).

EEG signals can be used to detect different medical conditions, such as epileptic seizures [41]. The detection of seizures through the use of a WBSN offers significant advantages. Because it is a relatively rare occurrence, seizure detection requires constant monitoring for an extended period of time, which is resource-intensive when carried in a health institution. Using an EEG WBSN can circumvent this by providing the patients a way to do the monitoring themselves and then consult a physician once the relevant data has been gathered.

Another important application of EEG signals in WBSNs is the use of a brain computer interface (BCI) that can detect the EEG patterns associated with a mental task performed by a patient [13]. The patient could use a mental task (such as attempting to move a finger or some arithmetic task) to operate a wheel chair, switch a light off or communicate with a caregiver. As the signals associated with the mental task are embedded in the patient's EEG, if their presence is detected in the EEG signals, then this information could be used to control the BCI. As will be seen later, one of the main components required in the successful use of BCIs in WBSNs is the development of advanced compression techniques that preserve the relevant information (or features) in the EEG signals [9].

Other common uses of EEG signals include sleep pattern studies, and the diagnosis and treatment of strokes, infections (e.g. encephalitis, cerebral abscess), cerebral tumors and diseases (e.g. Alzheimer's) (see e.g. [21, 32, 33, 59]).

In the above applications, it is preferable to have a system that is minimally obtrusive and allows the patient to move and walk freely, hence why WBSNs can be valuable.

1.3 EEG-Based Wireless Body Sensor Network

The setup for an EEG-based WBSN is as follows. EEG sensors are placed on a person's head, usually following an international convention (e.g. the international 10-20 system). An EEG sensor is also referred to as an EEG channel. The number of sensors (electrodes) depends on the application - some systems require few sensors while others require a few hundreds. Every sensor is wired to a single central microprocessor unit that would normally have three main components: a buffer (to store the EEG data stream coming from the different EEG channels; this buffer can be seen as memory), the microprocessor itself (to carry out any computations needed before transmission), and a low-power radio (to wirelessly transmit the data). The com-

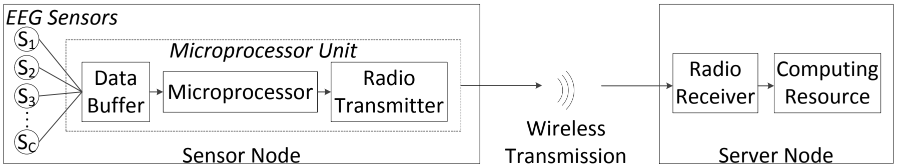


Figure 1.1: General block diagram for the EEG WBSN system

Table 1.1: EEG-Based WBSN Commercial Products

Device Name	**Release Date**	**Nb of EEG Channels**	**Battery Life**	**Reference**
Emotiv EPOC	December 2009	14	12h	[20]
Neurosky Mindwave	March 2011	1	8h	[43]
Interaxon Muse	December 2013	4	5h	[31]
Imec	Not announced yet	8	22h	[30]

bination of the EEG sensors and the microprocessor unit is what we refer to as the sensor node. This sensor node is battery-powered. The sensor node wirelessly transmits the EEG data to the server node placed in an adjacent room or somewhere in the vicinity of the sensor node. The server node is comprised of two main blocks: a low-power radio receiver (to receive the transmitted EEG data) and a computing resource (to carry out any post-transmission computations, storage, and any other desired operations). We assume there is no constraint on the energy supply or the computational power available from this server node. The complete system setup is shown in Fig. 1.1.

EEG-based WBSN products have started to appear on the market in the last decade or so. The best available commercial products are summarized in Table 1.1.

However, these products suffer from two major drawbacks. First, their battery life is limited. With a battery life of at most 22 hours for the least energy-hungry device, extended monitoring is not possible unless the patient were to change the batteries every day. This is clearly inconvenient. Second, the number of EEG channels is still small. With more EEG channels, greater spatial resolution could be achieved, which could improve current applications or even unlock new ones. Of course, the number of channels is intrinsically linked to the battery situation - using more channels would decrease the operating life time of the battery because of the additional sensing and wireless transmission required. There is also a limit to the number of data packets that the radio can transmit per unit of time, which may restrict the maximum number of channels that can be used.

Therefore, solutions that can address these two drawbacks are desirable and could lead to better, more practical commercial products available to customers and patients.

1.4 Challenges and Constraints of EEG-Based Wireless Body Sensor Networks

The energy available in the battery-powered sensor node in WBSNs is limited. This energy is needed for acquiring and digitizing the EEG samples, for carrying out the computations at the sensor node, and for wirelessly transmitting the data. Under current sensors technology, there is little that can be done to minimize the sensing energy - that is, the raw data must all be acquired and digitized. For computations carried out in the sensor node, energy savings should therefore be realized by using algorithms that are energy-efficient, i.e. have low computational complexity. To minimize the amount of data to be transmitted by the sensor node, the acquired signals should be compressed before their transmission. A higher compression ratio will minimize the energy required for transmission. In other words, reducing the amount of computations and the number of bits transmitted is therefore crucial if one is to minimize the overall energy consumption of the system. There is thus a need to find compression solutions for EEG signals that do not require much energy.

Traditionally, measurements are collected by the sensors at the Nyquist rate. Then, compression algorithms are directly applied to the acquired data, at the sensor node, prior to wirelessly transmitting the data to the server node. The computational demand (and thus energy consumption) of traditional compression algorithms is high at the sensor node. This is highly undesirable for WBSNs.

1.5 Aim of the Thesis

The aim of this thesis is to find a compression framework suitable for an EEG-based WBSN in the context of telemedicine. Based on the discussion presented in previous sections, our goal is to find a solution that is energy-efficient so that the system has a longer battery life.

1.6 Existing EEG Compression Algorithms

There exists an important body of work related to the compression of EEG signals. Broadly speaking, compression algorithms can be split into two categories: lossless and lossy. Lossless algorithms give the ability to exactly recover the original signal, at the expense of greater computational complexity and lower compression ratios. Lossy algorithms, on the other hand, do not allow for perfect recovery of the original signals. However, these algorithms provide higher compression ratios and tend to be simpler. Given the constraints mentioned in the previous section, in this work we focus on lossy algorithms, as lossless algorithms are too complex in the WBSN setting. We also require high compression ratios, which is only possible through the use of lossy algorithms.

For the sake of completeness, let us still briefly go over the body of literature regarding lossless algorithms. To achieve losslessness, the algorithms usually employ a lossy algorithm but also model the error so that perfect recovery can be achieved. Since the 1990's, many lossless EEG compression algorithms have been proposed, with varying complexity and compression performance (see for instance [4, 17, 38, 52–56, 61]).

In many practical cases, exact recovery is not necessary, and a small reconstruction error is tolerable (the magnitude of this error depends on the particular application). Lossy algorithms are an interesting solution in these instances. Table 1.2 gives an overview of the main lossy EEG compression algorithms in the literature.

Table 1.2: EEG Lossy Compression Algorithms

Reference	Year	Brief Description
[16]	2004	Thresholding of Daubechies-8 wavelet packets coefficients, followed by uniform quantization and run length encoding
[26]	2009	Classified signature and envelope vector sets approach
[27]	2010	Thresholding of CDF 9/7 wavelet coefficients, followed by uniform quantization and arithmetic encoding
[28]	2011	Thresholding of CDF 9/7 wavelet coefficients, followed by uniform quantization and set partitioning in hierarchical trees (SPIHT) encoding
[10]	2011	Thresholding of nearly-perfect reconstruction cosine-modulated filter banks coefficients, followed by dynamic uniform quantization and run length encoding
[18]	2011	Clustering of EEG channels, followed by regular and differential embedded zero-tree wavelet encoding

1.7 Compressed Sensing

Recent research has demonstrated the use of compressed sensing (CS) as an alternative compression scheme for physiological signals in the context of WBSNs [36]. CS is a novel paradigm which allows sampling of the signals at a sub-Nyquist rate. After acquiring the raw data, CS obtains a much smaller number of samples by taking linear projections of the raw data. This is a simple operation which can be done at a low energy cost. The reconstruction of the data is computationally complex and is done at the server node [15].

In a nutshell, if a signal can be sparsely represented by some dictionary (i.e., if it can be well represented by a small number of transform coefficients), then a small number of random linear projections (roughly proportional to the information rate of the signal) is sufficient to recover the signal exactly. To reconstruct the signal, we use greedy algorithms or an optimization-based approach. The CS theory also extends to compressible signals (where the signal has many very small coefficients in some dictionary), although in this case the reconstruction is not exact [14]. Chapter 2 contains more details about CS and its underlying theory.

1.8 Motivation

CS is an interesting paradigm in the case of WBSN since it requires very simple computations at the sensor node (non-adaptive random projections), i.e. the node where the signals are acquired and compressed. The reconstruction, which is computationally intensive, is shifted to the server node. In WBSNs, we generally place no limitation on the energy consumption (and computational power) of the server, whereas the sensor node is heavily constrained both in terms of energy available and computational power. As such, CS fully exploits the strengths of WBSNs.

1.9 Literature Review

The first study that applied CS to EEG compression used the multiple measurements vectors (MMV) approach to jointly compress and reconstruct the signals of the different EEG channels (i.e. all channels are reconstructed simultaneously) [5]. The obtained results were good (high compression ratio for reasonable reconstruction error) but the experimental setup was such that the EEG signals used were coming from repeated trials (asking the patient to repeat the same task many times and recording one EEG channel each time). This setup increases the coherence in the signals (asking someone to carry out the same task is bound to result in EEG signals that are highly coherent). This setting is of limited interest in telemedicine applications, since in these applications the patient is usually not prompted to act in a certain way or to repeat the same task multiple times.

Other works have looked at CS approaches to compress EEG signals. In [50], the authors developed a scheme using Slepian functions. However, it is necessary to know the Slepian coefficients prior to compression, which is undesirable from a computational point of view. Another work used a finite rate of innovation approach to compressively sample EEG signals [48]. The major drawback of this method is the fact that the innovation rate must be known, a task which can be difficult and impractical.

For telemedicine applications, the first study that addressed the potential of CS in EEG signal compression is found in [1] and [2]. This work focused on surveying existing sparsifying dictionaries and reconstruction algorithms, and testing different combinations of these elements to determine which yielded the best results. The conclusion was that the applicability of single-channel CS for EEG signals depends on the final application and the tolerable reconstruction error. This work therefore did not look into novel CS frameworks for WBSNs.

More recently, Independent Component Analysis (ICA) was appplied as a preprocessing step before using CS for the compression of EEG signals of newborn babies [39]. The compression results obtained were superior to other state-of-the-art methods that do not employ ICA preprocessing. This system however consumes much energy at the sensor node and would not be suitable for telemedicine applications. This is because the ICA algorithm used is computationally intensive, and such an operation must be carried at the sensor node.

1.10 Contributions of this Research

The above studies have resulted in some important questions: (i) What energy savings can be realized through the use of CS for EEG WBSN applications? (ii) Is it possible to exploit both the temporal correlations (intra-correlations) and the spatial correlations (inter-correlations between channels) to increase the compression performance of CS? (iii) How does CS compare with other state-of-the-art compression algorithms for EEG compression in WBSNs, for different types of EEG signals? (iv) What is the impact of CS compression on the performance of a practical application that uses EEG signals?

The main contributions of this thesis are:

- To propose novel CS frameworks that fully take advantage of the inherent structure present in EEG signals (both temporal and spatial correlations) to improve the compression performance.
- To compare CS frameworks with other state-of-the-art compression frameworks for EEG compression in WBSNs. It is also the first study

where different types of EEG signals representing a variety of applications are used to test the performance of the proposed and existing frameworks, thus providing a more robust answer to the usefulness and validity of the systems.

- To apply a CS framework to compress signals in the context of a BCI application and to evaluate its impact on the performance of the system.

1.11 Notation

Before getting started, let us first briefly introduce the notation used throughout this thesis:

- Regular letters (either lowercase or uppercase) represent scalar numbers and variables;
- Bold, lowercase letters represent column vectors;
- Bold, uppercase letters represent matrices;
- $\| \cdot \|_p$ corresponds to the ℓ_p norm of a vector, computed as follows:

  $$\|\boldsymbol{x}\|_p = \sum_{i=0}^{N} |\boldsymbol{x}_i|^p$$

  where $\boldsymbol{x}$ is a vector of length N.
- $(\cdot)^T$ corresponds to the transpose operation.

1.12 Organization of the Thesis

This thesis is organized as follows. Chapter 2 gives an overview of the theory underlying CS. Chapters 3-5 each present a different version of a CS framework for EEG compression in the context of WBSNs. Chapter 3 introduces a simple CS framework that provides the basis we will build upon and improve in the next 2 chapters. Chapter 4 improves the reconstruction performance of the simple framework by using an energy-efficient ICA preprocessing algorithm to exploit the interchannel correlations. Chapter 5 proposes the first complete, practical framework for an EEG-based WBSN system. Chapter 6 applies the framework of Chapter 5 to a simple BCI system to test the classification performance of such a system when EEG signals are compressed. Finally, Chapter 7 concludes this thesis by summarizing the contributions of this research and by offering possible extensions of our work and paths to explore.

Chapter 2

Compressed Sensing

This chapter briefly discusses the key theoretical concepts behind compressed sensing (CS). There is a very large body of literature around the theory of compressed sensing, and it is not our aim to cover it entirely. Given that this thesis is focused on applying compressed sensing to a practical problem, we focus on the basics that will allow the reader to quickly grasp how CS works. This chapter is organized as follows. Section 2.1 discusses what is meant by the sparsity of a signal. Section 2.2 explains the acquisition and reconstruction framework for CS. Section 2.3 discusses the necessary conditions on the measurement matrix so that reconstruction is possible, and Section 2.4 extends the CS theory to compressible signals. Finally, Section 2.5 introduces an alternative formulation for the CS problem and Section 2.6 clears up a few points about the two CS formulations.

2.1 Signal Sparsity

CS exploits the fact that most signals have a sparse representation in some dictionary. Denoting this dictionary by the matrix $\boldsymbol{\Psi}_{N\times K} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_K]$, we can write an original one-dimensional signal $\boldsymbol{f}$ of length N as

$$\boldsymbol{f} = \boldsymbol{\Psi c} = \sum_{i=1}^{K} c_i \boldsymbol{\psi}_i. \tag{2.1}$$

When the $K \times 1$ vector $\boldsymbol{c}$ has a large number of zero (or small, insignificant) coefficients, $\boldsymbol{f}$ can be obtained from $\boldsymbol{c}$ using few dictionary vectors $\boldsymbol{\psi}_i$. The number of nonzero elements of $\boldsymbol{c}$ is called the sparsity of $\boldsymbol{f}$. If there are S such elements, it is said that $\boldsymbol{c}$ is the S-sparse representation of $\boldsymbol{f}$ in dictionary $\boldsymbol{\Psi}$. $\boldsymbol{\Psi}$ is also called the synthesis operator.

2.2 Signal Acquisition and Reconstruction

The compressed sensing theory implies that instead of acquiring the N samples of the signal $\boldsymbol{f}$ and then compressing it, it is possible to only acquire M samples, where M is slightly larger than S but is still much smaller than N. As mentioned above, S is the smallest number of vector elements in the dictionary $\boldsymbol{\Psi}$ that represent the signal $\boldsymbol{f}$, so that all information in $\boldsymbol{f}$ is captured.

This sampling paradigm - referred to as ‘analog CS’ - is the ultimate goal of CS. However, it cannot yet be attained by present day sampling technologies. At present, to represent $\boldsymbol{f}$ using M samples, all the N samples are collected, discretized and then subsampled. The subsampling is carried out using M linear projections of $\boldsymbol{f}$. That is, CS subsamples $\boldsymbol{f}$ using an $M \times N$ measurement matrix[1] $\boldsymbol{\Phi}$, with $M \ll N$, thus creating a measurement vector $\boldsymbol{y}$ of length M:

[1] also referred to as the sensing matrix

$$\boldsymbol{y}_{M\times 1} = \boldsymbol{\Phi f} = \boldsymbol{\Phi \Psi c}. \tag{2.2}$$

This system of equations is largely underdetermined. That is, given a $\boldsymbol{y}$ vector, there is an infinite number of solutions for $\boldsymbol{f}$ (or equivalently, $\boldsymbol{c}$). However, since the signal we wish to recover ($\boldsymbol{f}$) is sparse, the correct solution is often the sparsest solution. This corresponds to solving the following l_0 optimization problem:

$$\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_0 \text{ subject to } \boldsymbol{y} = \boldsymbol{\Phi \Psi c}. \tag{2.3}$$

where the ℓ_0 pseudonorm $\|\cdot\|_0$ is the number of non-zero elements in a given vector. Unfortunately, this problem is NP hard, and as such it is not tractable. Indeed, solving this problem requires an exhaustive search over all subsets of columns of $\boldsymbol{\Phi}$, which is a combinatorial problem [14].

Fortunately, an ℓ_1 optimization problem, a more practical problem due to its convexity, was shown to be equivalent under some conditions. It can be rewritten as follows:

$$\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 \text{ subject to } \boldsymbol{y} = \boldsymbol{\Phi \Psi c}. \tag{2.4}$$

Equation 2.4 is called the synthesis prior formulation for CS. This problem can be recast as a linear programming one, for which many practical solvers exist. It has also been shown that perfect recovery can be achieved, even when a small number of measurements (i.e. $M \ll N$) is used [19]. Perfect recovery is possible if the restricted isometry property (RIP) of $\boldsymbol{\Phi \Psi}$

is satisfied. The RIP$_p$ is defined as follows:

$$(1-\delta_S)\|\boldsymbol{x}\|_p^2 \leq \|\boldsymbol{\Phi\Psi x}\|_p^2 \leq (1+\delta_S)\|\boldsymbol{x}\|_p^2 \tag{2.5}$$

where $\delta_S < 1$, $\boldsymbol{x}$ is any S-sparse vector and p is the chosen norm (typically 1 or 2). When $p = 2$, there is an intuitive geometric explanation of the RIP. Indeed, the RIP means that S-sparse vectors are not in the null space of $\boldsymbol{\Phi\Psi}$ and that S-sparse linear combinations of columns of $\boldsymbol{\Phi\Psi}$ behave close to an orthonormal system [15].

2.3 Incoherent Measurements

The minimum acceptable value for M that allows the perfect reconstruction of the signal $\boldsymbol{f}$ is not only linked to the degree of sparsity S of $\boldsymbol{f}$ in dictionary Ψ but also to μ, the degree of coherence between $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$. This coherence measures the largest correlation between any element of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ and is measured by

$$\mu(\boldsymbol{\Phi},\boldsymbol{\Psi}) = \sqrt{N} \cdot \max_{1\leq l,j\leq N} |\langle \boldsymbol{\phi}_l, \boldsymbol{\psi}_j \rangle|. \tag{2.6}$$

The number of measurements M is given by

$$M \geq C \cdot \mu^2(\boldsymbol{\Phi},\boldsymbol{\Psi}) \cdot S \cdot \log N \tag{2.7}$$

where C is a positive constant [15]. Thus, the smaller the coherence, the smaller the value of M can be. As such, it is important to select $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ so that they are maximally incoherent.

Ideally, one should not need to know the sparsifying dictionary $\boldsymbol{\Psi}$ in order to pick a measurement matrix $\boldsymbol{\Phi}$. Fortunately, some measurement matrices can be shown to be maximally incoherent with any sparsifying dictionary with overwhelming probability. Random matrices have this property. Indeed, a matrix $\boldsymbol{\Phi}$ generated by independent and identically distributed (IID) Gaussian random variables or by IID Bernoulli random variables would display this property [14]. This means that a random measurement matrix that is properly constructed can allow perfect reconstruction without having any knowledge about the original signal.

2.4 Extension to Compressible Signals

CS can further be extended to compressible signals (signals that are not purely sparse in a given dictionary but whose coefficients $\boldsymbol{c}$ decay with a power law when arranged in descending order). This setting is more realistic in practice, as real signals are rarely purely sparse but are often compressible. This is indeed the case for EEG signals.

Suppose that we are interested in recovering the P largest coefficients of the compressible signal $\boldsymbol{f}$ (where $P \ll N$). That is, we want to recover the indices as well as the magnitudes of the P largest values of $\boldsymbol{c}$. Suppose that the number of collected measurements (M) is equal to P, i.e. only $P \ll N$ random projections of the signal are collected. If the indices (i.e. the locations) of these P largest coefficients of the vector $\boldsymbol{c}$ are known, then optimal performance could be achieved, i.e. it is possible to exactly recover the magnitude of each of these P largest coefficients. This means that it is possible to reconstruct $\boldsymbol{c}$ (or equivalently $\boldsymbol{f}$) with an accuracy corresponding to its P largest coefficients. Now, it can be shown that CS can asymptotically obtain the same accuracy as this optimal solution, as long as the number of random projections is increased by a factor of $\mathcal{O}\Big(\log\left(\frac{N}{P}\right)\Big)$ [14]. This result is spectacular, as it means that compressed sensing's non-adaptive sampling scheme performs as well as a purely adaptive scheme that knows the locations of the P largest coefficients. In other words, CS is able to find the P largest coefficients without any knowledge of the signal. The only cost is a mild oversampling factor.

2.5 Analysis Prior Formulation

In section 2.2, we introduced the synthesis prior formulation for CS reconstruction:

$$\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 \text{ subject to } \boldsymbol{y} = \boldsymbol{\Phi\Psi c}.$$

where $\boldsymbol{f} = \boldsymbol{\Psi c}$. In this formulation, the objective was to solve for the sparse coefficients $\boldsymbol{c}$. This formulation is the most popular in the CS field and has been studied extensively.

There is an alternative formulation, where the objective is to solve for $\boldsymbol{f}$ such that $\boldsymbol{\Gamma f}$ is sparse. This formulation is referred to as the analysis prior formulation:

$$\min_{\boldsymbol{f}} \|\boldsymbol{\Gamma f}\|_1 \text{ subject to } \boldsymbol{y} = \boldsymbol{\Phi f}. \tag{2.8}$$

where the dictionary $\boldsymbol{\Gamma}$ is called the analysis operator. This formulation has only started to be studied recently and its surrounding body of literature is smaller than that of its synthesis counterpart.

The analysis and synthesis prior formulations are equivalent only when the sparsifying dictionaries are orthogonal (i.e. $\Psi = \Gamma^{-1}$). When the dictionaries are tight frames, then both the synthesis and analysis problems can be solved, although they do not yield the same result. Two frames are said to be tight if they satisfy a generalized version of Parseval's theorem:

$$\sum_{i=1}^{K} (\boldsymbol{g} \cdot \boldsymbol{\psi}_i)^2 = A \|\boldsymbol{g}\|_2^2$$

for all vectors $\boldsymbol{g}$ of length N, with $0 < A < \infty$. In this case, $\Psi = \Gamma^T$.

In our work, we use the synthesis prior formulation. This is primarily because the underlying theory is better known, many practical algorithms can solve this problem and previous work has focused on finding good dictionaries $\boldsymbol{\Psi}$ for EEG signals. On the other hand, there are few algorithms that can solve the analysis problem (Eq. 2.8) and no previous work has attempted to find a good analysis dictionary $\boldsymbol{\Gamma}$ for EEG signals. We note that the analysis prior formulation may yield better results than the synthesis prior formulation.

2.6 A Cautionary Note on the Analysis and Synthesis Operators

In sections 2.2 and 2.5, we introduced the notion of synthesis and analysis operators, which were used in solving the CS problem:

$$\begin{aligned} \text{Synthesis:} \quad & \boldsymbol{f} = \boldsymbol{\Psi c} \\ \text{Analysis:} \quad & \hat{\boldsymbol{c}} = \boldsymbol{\Gamma f} \end{aligned} \tag{2.9}$$

where $\boldsymbol{f}$ is the original signal, $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ are the coefficients in the transform domain, $\boldsymbol{\Psi}$ is the synthesis operator and $\boldsymbol{\Gamma}$ is the analysis operator. Knowledge of the synthesis and analysis operators allow us to solve the synthesis and analysis prior problems, respectively.

When $\boldsymbol{\Psi}$ and $\boldsymbol{\Gamma}$ are tight frames (i.e. $\Psi = \Gamma^T$), Eq. 2.9 becomes:

$$\begin{aligned} \text{Synthesis:} \quad & \boldsymbol{f} = \boldsymbol{\Psi c} \\ \text{Analysis:} \quad & \boldsymbol{c} = \boldsymbol{\Psi^T f} \end{aligned} \tag{2.10}$$

That is, the knowledge of either $\boldsymbol{\Psi}$ or $\boldsymbol{\Gamma}$ is enough to solve both the synthesis and the analysis problems. However, if we do not have tight frames, we must explicitly know $\boldsymbol{\Psi}$ and $\boldsymbol{\Gamma}$ to solve the respective problem.

In most practical instances, both the analysis and synthesis operators are known. When only the analysis operator $\boldsymbol{\Gamma}$ is known, the synthesis operator can be easily found if $\boldsymbol{\Gamma}$ is orthogonal or a tight frame. The case for when only the synthesis operator is known is analogous.

If only the analysis operator $\mathbf{\Gamma}$ is known and it is not orthogonal or a tight frame, then we cannot solve the synthesis prior problem since the synthesis operator $\mathbf{\Psi}$ is not known. However, if we only know the synthesis operator $\mathbf{\Psi}$, then the synthesis problem can be solved provided the specific algorithm used in solving it does not require the knowledge of $\mathbf{\Gamma}$.

Chapter 3

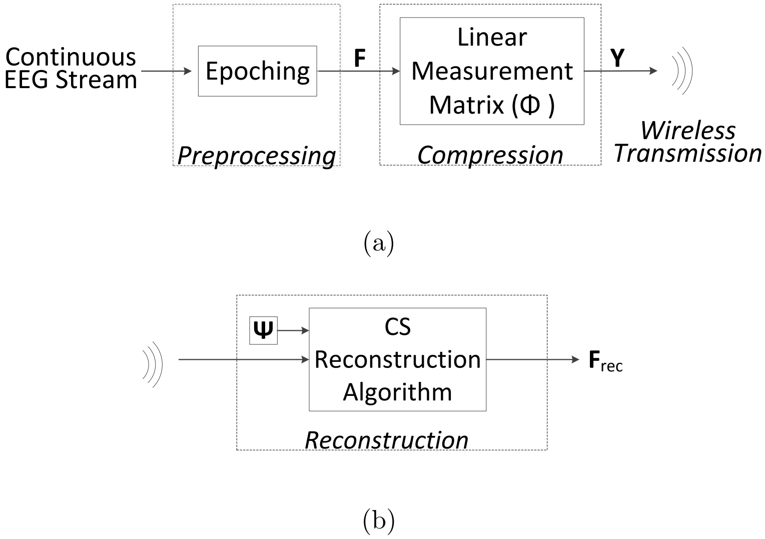
A Simple Compressed Sensing Framework for the Compression of EEG Signals

3.1 Problem Description

As mentioned in the introduction, compressed sensing (CS) is an interesting paradigm in the case of EEG-based WBSNs since the computational load at the sensor node is very light - the only operation that is carried out is to take non-adaptive random projections of the EEG signals. As such, CS has the potential to offer an energy-efficient compression scheme in this context.

While a few studies have looked into CS for EEG-based systems (refer to Section 1.9 in the introduction), most of them haven't considered it from a telemedicine perspective. Similarly, there has been little focus to optimize the different CS building blocks for an EEG telemedicine application.

In this chapter, we propose a simple CS framework that is suitable for a low-energy, power-efficient telemedicine application. To do so, we adapt the typical CS scheme to reduce the load at the sensor node and use a



(a)

(b)

Figure 3.1: Block diagrams of (a) the sensor node and (b) the server node for the simple compressed sensing framework

state-of-the-art reconstruction strategy.

This chapter is organized as follows. Section 3.2 presents the different building blocks of the framework. Section 3.3 shows the performance of the system through different experiments. Section 3.4 discusses the obtained results.

3.2 Methods

Below we present the different blocks that make up the system. We will discuss the preprocessing, the compression, the wireless transmission and the reconstruction. A block diagram of the proposed system is shown in Fig. 3.1.

3.2.1 Preprocessing

The EEG data is formed of C signals, where C represents the number of channels (sensors). The signal from each channel is first divided into non-overlapping line segments (epochs) of length N. In our experiments, N corresponded to 512 samples for each channel. Note that our framework operates on data from one epoch at a time. For C channels of EEG data, after dividing the data into epochs a total of C sequences of $N = 512$ data points each are obtained for each segment of time. These are represented by $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_C$. This forms a matrix $\boldsymbol{F}$. Each column of $\boldsymbol{F}$ contains one of the channels: $\boldsymbol{F}_{N \times C} = [\boldsymbol{f}_1 | \boldsymbol{f}_2 | \ldots | \boldsymbol{f}_C]$.

3.2.2 Compression

To compress the EEG signals contained in one epoch, we take their linear random projections. As mentioned in Section 2.3, the chosen measurement matrix $\boldsymbol{\Phi}$ must be maximally incoherent with the sparsifying dictionary. With overwhelming probability, random matrices satisfy this requirement irrespective of the dictionary used. The most often used matrices are random matrices with IID entries formed by sampling a Gaussian distribution ($\mathcal{N}(0, 1/N)$) or a Bernouilli distribution (with $P(\boldsymbol{\Phi}_{i,j} = +1/\sqrt{N}) = 1/2, P(\boldsymbol{\Phi}_{i,j} = -1/\sqrt{N}) = 1/2$). In fact, it can be proven that these two types of matrices (Gaussian and Bernouilli) are optimal. Unfortunately, these matrices are not suitable for WBSN applications. This is because generating a Gaussian random matrix requires a Gaussian random generator at the sensor node (which cannot be efficiently implemented on simple sensor hardware). Moreover, using such a matrix would lead to C large matrix-vector multiplications for each epoch (one for each of the C channels). As these multiplication operations are energy-intensive, they should be avoided in WBSN applications. They are also time consuming, and would therefore prevent the system from operating in near realtime mode.

The use of a full Bernouilli matrix would reduce the challenges mentioned above (it is easier to generate its random entries, and it also has simpler multiplication operations) but this unfortunately would still require a high number of computations.

Instead, we use what is known as sparse binary sensing. This was first proposed in [23] and has since been applied to WBSNs ([36, 63]). These matrices only contain d nonzero entries in each column, and the value of these nonzero entries is 1. The optimal value of d (that is, the smallest value for which reconstruction would be stable) is obtained experimentally and is much smaller than M. While the theoretical guarantees are not as strong as those for full random matrices, it has been shown that these matrices perform well in many practical applications. There are significant advantages to using such a matrix in WBSNs. The most important one is that the matrix multiplication operation is very simple: in fact, it consists of $(Nd - M)$ simple additions for each channel.

We propose the use of the same sparse binary sensing matrix for each channel (sensor) so that we can more easily generate and/or store the matrix in memory at the sensor node. This setup also preserves the interchannel correlations, a feature which will be exploited in later chapters. We will test 2 different settings: the first uses a fixed sensing matrix (stored in the sensor node memory) for all epochs, and the second generates a new matrix for each epoch.

We apply the $M \times N$ measurement matrix $\mathbf{\Phi}$ to each EEG channel in order to obtain the compressed measurements: $\boldsymbol{Y}_{M \times C} = [\boldsymbol{y}_1|\boldsymbol{y}_2|\ldots|\boldsymbol{y}_C] = [\mathbf{\Phi}\boldsymbol{f}_1|\mathbf{\Phi}\boldsymbol{f}_2|\ldots|\mathbf{\Phi}\boldsymbol{f}_C]$

3.2.3 Wireless Transmission

The compressed measurements $\boldsymbol{Y}$ are then wirelessly transmitted from the sensor node to the server node. Note that we make no attempt to model the wireless channel and instead treat it as a black box.

3.2.4 Reconstruction

The final step is to reconstruct the original EEG signals from the compressed measurements $\boldsymbol{Y}$. The vast majority of reconstruction algorithms use a dictionary in which the signals are sparse (or at least compressible).

Sparsifying Dictionary ($\mathbf{\Psi}$)

As discussed previously, one of the main elements of compressed sensing is the selection of a proper sparsifying dictionary $\mathbf{\Psi}$. Different sparsifying dictionaries were tested in [2] but there was no aim at optimizing the chosen dictionaries to obtain the best performance possible (that is, to obtain the dictionary in which EEG signals are the most sparse). Previous work has shown that EEG signals are sparse in the Gabor domain [5].

The Gabor dictionary is a redundant dictionary which provides optimal joint time-frequency resolution [22]. Gabor functions are sinusoidallymodulated Gaussian functions. The atoms in this dictionary are given by

$$g_i(n; n_0, f_0, s) = K(n_0, f_0, s) \ \cdot \ \exp\left(-\frac{(n-n_0)^2}{2s^2}\right) \cdot \ \sin\left(2\pi \cdot f_0(n-n_0)\right)$$

where n_0 and f_0 are the centers of the Gabor atom, s is the spread of the atom, and $K(n_0, f_0, s)$ is a normalizing constant.

We now require a discretization over the n_0, f_0 and s parameters - that is, we need to determine how to increment these parameters. For s, the chosen discretization is a dyadic scale (base 2). The time increment n_0 is proportional to the spread, and the frequency increment f_0 is inversely proportional to the spread. The size of the dictionary depends on the length of the EEG epoch considered in the time domain. To obtain the frequency step and the time step, we rely on the following equations:

$$\Delta f_0 = \frac{\sqrt{8\pi\alpha}}{sN}$$
$$\Delta n_0 = sN \times \sqrt{\frac{2\alpha}{\pi}}$$

where N is the epoch length and $\alpha = 0.5\text{ln}(0.5(B + 1/B))$. B is the base used (2 in our case).

These equations are based on the distance metric for Gabor atoms proposed in [8]. We can show that selecting n_0 and f_0 in this manner provides a dictionary with optimal distance between the atoms.

Reconstruction Algorithm

There exists a multitude of reconstruction algorithms for reconstructing the original signals. It is possible to use convex optimization to solve (2.4). It is also possible to use a greedy algorithm, which looks at finding a suboptimal solution to (2.3) by iteratively selecting the locally optimal solution, in the hope of getting closer to the globally optimal solution. Such algorithms converge to an acceptable solution faster than convex optimization algorithms. This is however at the expense of requiring slightly more measurements.

In our framework, we use a convex optimization algorithm, the Basis Pursuit Denoise algorithm implemented in the SPGL1 package [60]. Based on our tests, this algorithm required fewer measurements (i.e. a smaller value for M) than greedy algorithms to achieve an equivalent reconstruction accuracy; this results in a higher compression as a smaller M can be used. After performing the reconstruction on each channel data (i.e. on each column of $\boldsymbol{Y}$), we obtain the $N \times C$ matrix of reconstructed signals: $\boldsymbol{F}_{\text{rec}} = [\boldsymbol{f}_1|\boldsymbol{f}_2|\ldots|\boldsymbol{f}_C]$.

3.3 Results

3.3.1 Dataset

In order to assess the performance of the simple framework, we used the data from Dataset 1 of the BCI Competition IV [12]. This dataset was recorded from healthy subjects or generated artificially and contains the data of 7 patients. The recording was made using 59 EEG channels per subject at an initial sampling rate of 1000Hz. We resampled the data to 128Hz so as to provide a realistic sampling frequency in the context of telemedicine. Non-overlapping windows of length $N = 512$ were used in our experiments. The experiments were carried using 100 randomly extracted windows.

3.3.2 Performance Metrics

To quantify the compression performance, we used the compression ratio (CR), defined as follows:

$$\mathrm{CR} = \frac{N}{M} \tag{3.1}$$

To test the reconstruction quality, we used the normalized mean square error (NMSE), defined as follows:

$$\mathrm{NMSE}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\|\boldsymbol{x} - \mu_x\|^2} \tag{3.2}$$

where $\boldsymbol{x}$ is the original vector, $\boldsymbol{y}$ is the reconstructed vector and μ_x is the mean of $\boldsymbol{x}$.

The NMSE measures the distance between 2 vectors. Of course, the lower the NMSE, the better the reconstruction. Note that in our formula, we normalize by the de-meaned original signal so that differences in means between datasets do not bias the results.

3.3.3 Choice of the CS Measurement Matrix

As mentioned in section 3.2.2, we employ sparse binary sensing where the measurement matrix $\mathbf{\Phi}$ contains d nonzero entries in each column. There

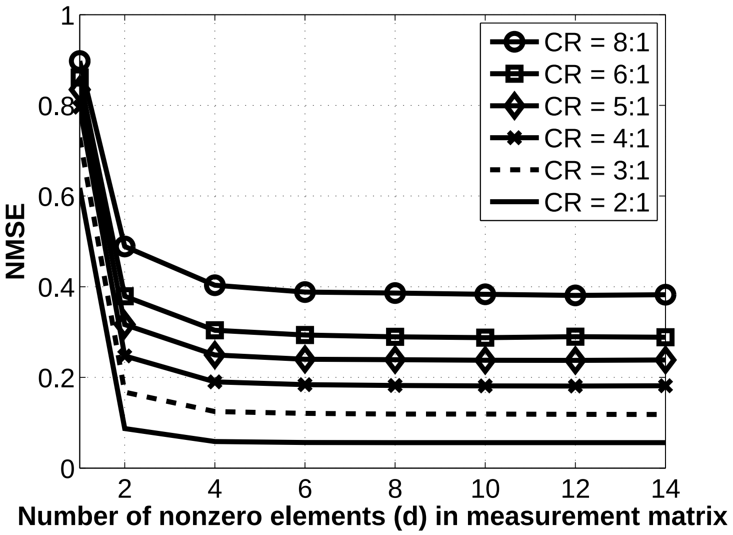


Figure 3.2: NMSE vs d for different compression ratios

are no theoretical guidelines for choosing the optimal value of d; we thus determine it experimentally. To do so, we vary the value of d and carry out the reconstruction for various compression ratios. The results are shown in Fig. 3.2.

As seen from this figure, the NMSE saturates relatively fast, which is a desirable property. Indeed, once the number of nonzero entries in each column reaches 8, increasing its value does not yield better reconstruction. As such, we select $d = 8$ in our implementation. Of course, in terms of energy consumption, it is desirable to have the lowest value for d, since this results in fewer random linear projections (and thus fewer computations) at the sensor node. Looking at Fig. 3.2, we can see that we could use as few as 4 nonzero entries per column for the measurement matrix. However, because improvements (albeit small) are still possible for $d > 4$, we decided

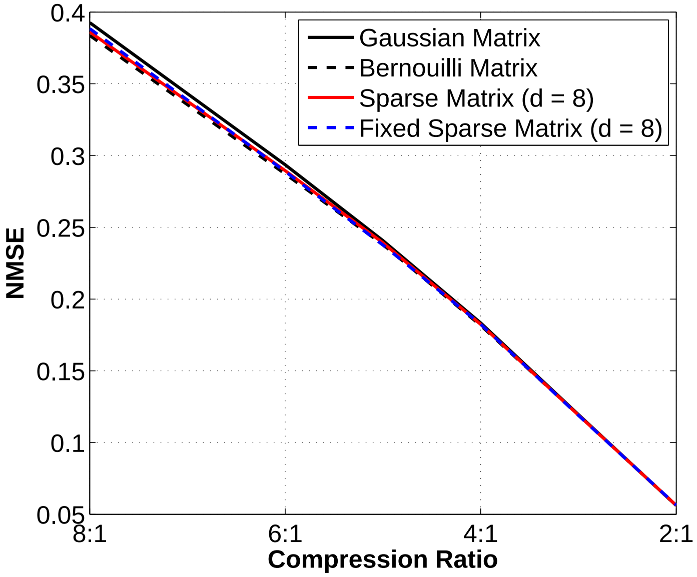


Figure 3.3: NMSE vs CR for different measurement matrices

to use $d = 8$ instead of 4 for our experiments.

We then set out to verify the performance of this sub-optimal measurement matrix by comparing the reconstruction performance with that obtained using an optimal random matrix (Gaussian or Bernouilli). We study the reconstruction error for different compression ratios using 4 different matrices: (1) an optimal Gaussian random matrix, in which each entry is formed by sampling an IID Gaussian random variable, (2) an optimal Bernouilli random matrix, in which each entry is formed by sampling an IID Bernouilli random variable, (3) a sparse binary sensing matrix (with $d = 8$) that is different for every epoch analyzed, and (4) a fixed sparse binary sensing matrix with ($d = 8$) i.e. the same matrix is used for all epochs. The results are shown in Fig. 3.3.

As seen from Fig. 3.3, although the theoretical guarantees for the sub-optimal sparse binary sensing matrices are weaker than for the optimal Gaussian and Bernouilli matrices, in practice their performance is statistically almost the same. We can thus safely assume that using a sub-optimal measurement matrix yields near-optimal reconstruction results. Of equal interest is that the fixed sparse binary sensing matrix does not result in degradation in the reconstruction quality. Because the proofs rely on the stochasticity of the matrices used, it is not possible to prove this result theoretically. The use of fixed sparse binary measurement matrices has significant advantages in the context of WBSNs since we can generate such a matrix offline and then store it in memory at the sensor node. The 2 other alternatives are 1) generating a new sparse binary sensing matrix at the sensor node for each epoch, or 2) generating the matrix at the server node and then wirelessly transmitting the positions of the nonzero entries to the sensor node. Both alternatives are more energy-hungry and are thus less desirable.

3.3.4 Reconstruction Performance

We then dive deeper into the reconstruction performance of the simple framework when using a fixed binary sparse sensing matrix with $d = 8$. The reconstruction performance for different compression ratios is shown in Table 3.1. Fig. 3.4 shows an example of the original and reconstructed EEG signals at different compression ratios so that one can empirically visualize the impact of compression on the EEG signals.

Table 3.1: Reconstruction Performance of the Simple Framework under Different Compression Ratios

CR	**Mean NMSE ± STD**
8:1	0.386 ± 0.187
6:1	0.290 ± 0.162
5:1	0.239 ± 0.145
4:1	0.182 ± 0.117
3.5:1	0.152 ± 0.101
3:1	0.119 ± 0.083
2.5:1	0.090 ± 0.065
2:1	0.056 ± 0.042

3.4 Discussion

In this chapter, we adapted the standard, simple CS framework to reduce the load at the sensor node. This was done by using a suitable measurement matrix that requires a small amount of computations. We showed that using such a matrix did not have an impact on the performance of the system as compared to using theoretically optimal measurement matrices. We also developed a sparsifying dictionary specific to EEG signals.

While this simple framework provides an interesting solution to the EEG-based WBSN application, it suffers from one major drawback: it does not exploit the spatial correlations (i.e. the correlations between EEG channels). It is quite likely that exploiting such correlations could improve the reconstruction performance of the CS framework. This will be investigated in the next 2 chapters.

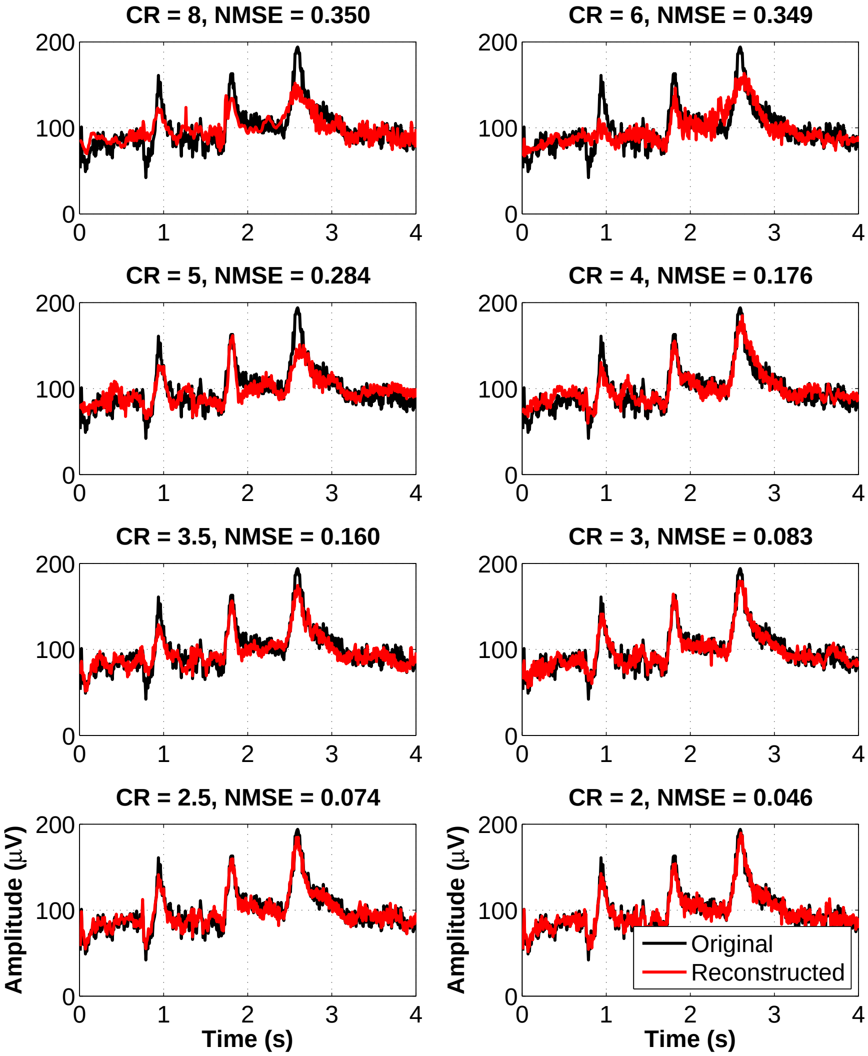


Figure 3.4: Original and reconstructed EEG signals (one channel) at different compression ratios

Chapter 4

Compressed Sensing and Energy-Efficient Independent Component Analysis Preprocessing for the Compression of EEG Signals

4.1 Problem Description

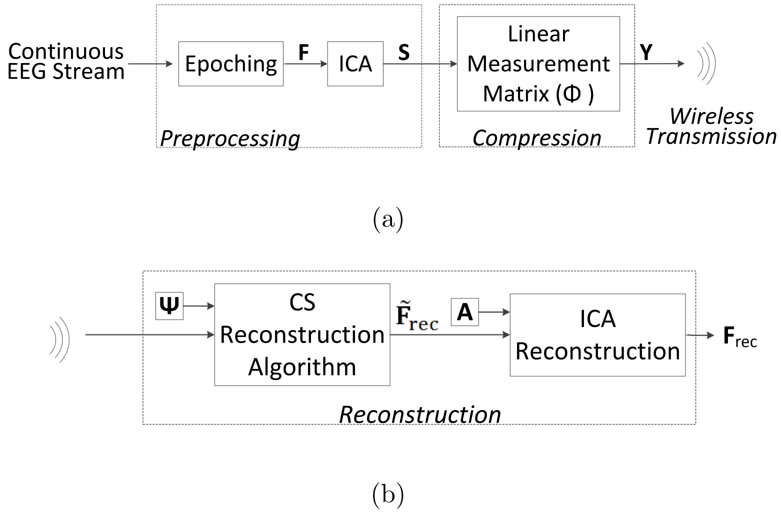
In this chapter, we improve the framework presented in the previous chapter by adding a preprocessing step that exploits the interchannel correlations. This is achieved through the use of Independent Component Analysis (ICA). Therefore, instead of compressing the original EEG signals, the independent components obtained after applying ICA to the original signals are compressed and wirelessly transmitted. This chapter discusses the development of an efficient ICA algorithm that requires low computational energy. By using the CS scheme presented in the previous chapter along with the proposed ICA algorithm, it is shown that the proposed framework achieves similar EEG compression results as those achieved by the state-of-the-art method proposed in [39] but is more energy-efficient at the sensor node.

This chapter is organized as follows. Section 4.2 presents the different building blocks of the proposed framework while Section 4.3 briefly introduces the state-of-the-art framework. Section 4.4 compares the performance of both frameworks through different experiments. Section 4.5 discusses the obtained results.

4.2 Methods

The proposed framework for compressing EEG signals is similar to the one presented in Chapter 3, with one addition: there is an ICA preprocessing step applied prior to compression. Compressed sensing is thus applied to the derived independent components directly. Instead of compressing the EEG signals $\boldsymbol{F}$ as in the previous chapter, we compress $\boldsymbol{S}$, the independent components: $\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{S}$. After CS reconstruction, we reconstruct the EEG signals from the independent components by using the ICA mixing matrix. A block diagram of the proposed system is shown in Fig. 4.1.

Since most of the framework is the same as for the one presented in the previous chapter, we will not go into the details of the framework again. We refer the reader to Section 3.2 for a detailed explanation. Instead, we focus on the energy-efficient ICA preprocessing step.



(a)

(b)

Figure 4.1: Block diagrams of (a) the sensor node and (b) the server node for the compressed sensing framework with ICA preprocessing

4.2.1 Energy-Efficient ICA

ICA has been around for close to two decades now and was proposed as a solution to the blind source separation problem. The solution to the problem is found by enforcing statistical independence of the source signals, commonly through maximizing the non-Gaussianity of the signals or through minimizing the mutual information of the signals [29]. ICA has also been successfully applied to EEG signals (see, for example, the study conducted by Makeig et al. [35]). The underlying basis as to why ICA works in the EEG case is that the electrical scalp potential measured by an EEG electrode can be seen as a mixture of a smaller number of ‘sources’ located in the brain that give rise to these potentials.

The ICA problem can be formulated as $\boldsymbol{F} = \boldsymbol{AS}$, where $\boldsymbol{F}$ is a matrix containing the measured mixed signals (each column containing one mixed signal), $\boldsymbol{A}$ is the mixing matrix, and $\boldsymbol{S}$ is a matrix containing the independent components (one source per column). The task is to find $\boldsymbol{A}$ and $\boldsymbol{S}$ from the observable measurements $\boldsymbol{F}$. Note that in our case, the mixed signals $\boldsymbol{F}$ correspond to the raw EEG signals.

FastICA (FICA) is a popular algorithm that can solve this problem [35]. Summarizing the FICA algorithm:

1. *Preprocessing*: Whiten (decorrelate) the mixed signals.

2. *Iteration*: In a deflationary manner (i.e. one at a time), estimate each independent component (IC) by maximizing its non-Gaussianity through a contrast function. Using Gram-Schmidt, orthogonalize the found IC with respect to the previously found ICs, and normalize it. Repeat this stage until the component converges.

However, this algorithm is computationally intensive and thus not suitable for WBSN applications. Acharyya et al. proposed an algorithm and an energy-efficient architecture to calculate n-dimensional (nD) cross-products, and mentioned that one potential application is FICA [3]. After identifying $n-1$ components with FICA, the n^{th} component can simply be identified by taking the cross-product of the first $n-1$ components since at that point, the direction for maximal independence has already been determined. We call this method xFICA. This allows the saving of one full iteration cycle, which is not negligible.

4.3 State-of-the-Art System

Besides comparing the proposed ICA-augmented framework with the simple framework of Chapter 3, we also compare it with the state-of-the-art system from the literature proposed in [39]. The block diagram of this system is the same as for ours, although different components and algorithms are used. Second Order Blind Identification (SOBI) is used as the ICA algorithm. The measurement matrix $\mathbf{\Phi}$ is the typical Gaussian random matrix. It is mentioned that the sparsifying dictionary is a Gabor dictionary, and the reconstruction algorithm used is iterative hard thresholding. However, due to a lack of details about the parameters used for these last two elements, we used our own Gabor dictionary in combination with Basis Pursuit.

4.4 Results

4.4.1 Data Used

We randomly selected 100 epochs of $N = 512$ samples each from dataset # 1 of BCI Competition IV [12]. We resampled the data to 128Hz and selected 12 channels in the sensorimotor area of the cortex out of the 59 channels available. The selected channels were F1, FZ, F2, FC3, FC1, FCZ, FC2, FC4, C3, C1, CZ and C2.

4.4.2 Reconstruction Performance

In this experiment, we wish to test the compression and reconstruction performance of 1) the simple framework of Chapter 3, 2) the proposed method and 3) the method presented in [39]. We thus vary the number of independent components retained as well as the number of measurements M and we compute the reconstruction error, in terms of the NMSE, against the compression ratio (again defined as $\mathrm{CR} = N/M$). To keep the comparison fair, we must include the mixing matrix entries in the total number of measurements when ICA is used, and we must ensure that the same total number of measurements is used for all methods (thus ensuring a constant compression ratio). To select I independent components, we first apply Principal Component Analysis to the EEG data so as to only keep the I components that account for the most variance in the original data.

The results are shown in Fig. 4.2. In Fig. 4.3, we show a slice from Fig. 4.2 by selecting a compression ratio of 3:1 and showing the reconstruction error for each block when we vary the number of independent components.

As can be seen from Fig. 4.2, adding an ICA preprocessing step decreases the NMSE for a fixed compression ratio (or, alternatively, it allows for an increase in the compression ratio for a fixed NMSE), with our method and the state-of-the-art yielding similar results. In Fig. 4.3, we can see that except for a few epochs for 8 ICs, using an ICA preprocessing step systematically yields better results than using CS alone. Of course, selecting fewer independent components has advantages both in terms of compression ratio and speed (since fewer sources need to be reconstructed). Our experiments also suggest that even if we use a small number of ICs (e.g. 3 or 4), we are still able to accurately represent the original signal. However, one must be careful to keep enough ICs so that the variance of the data is preserved. Indeed, in our experiments we observed that keeping less than 3 ICs made

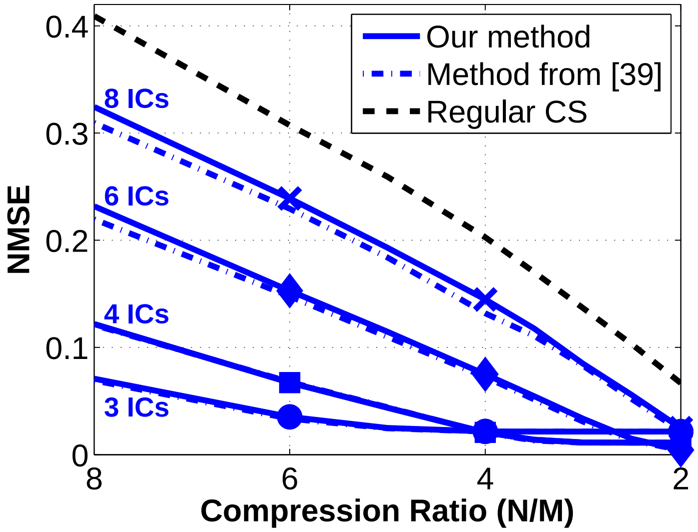


Figure 4.2: NMSE as a function of the compression ratio for regular CS, state-of-the-art from [39], and the proposed method.

it impossible to reconstruct the original signal faithfully.

4.4.3 Energy Analysis

We now wish to compare the energy performance at the sensor node of the state-of-the-art and proposed frameworks. We use the number of floating point operations (FLOPS) as a measure analogous to energy consumption. The number of FLOPS required is a measure of the dynamic power consumption. Reducing that value has the effect of reducing the overall power consumption. We look at the measurement matrices and the ICA algorithms used in both methods.

Measurement Matrix We first compare the random Gaussian matrix used in [39] and our binary sparse sensing matrix. The number of FLOPS

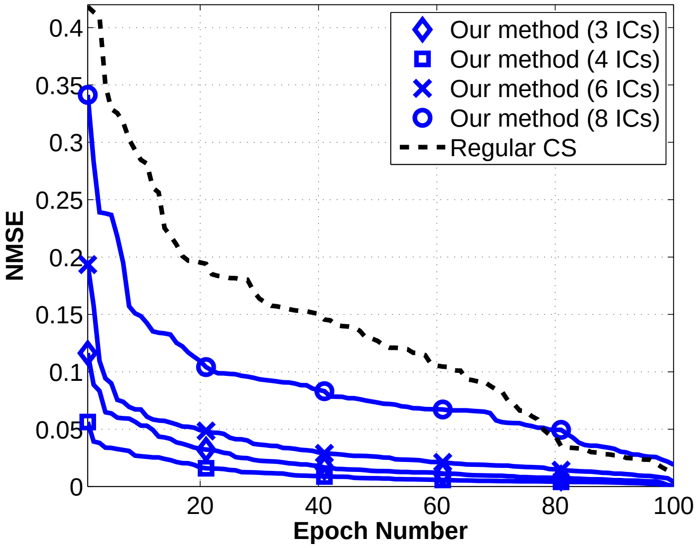


Figure 4.3: NMSE for the 100 epochs (in decreasing order of magnitude) when the compression ratio is 3:1.

(F) for both matrices is given by the following equations:

$$F_{\text{GM}} = (2MN - M) \cdot I$$

$$F_{\text{SBM}} = (Nd - M) \cdot I$$

where F_{GM} is the FLOPS count for the Gaussian matrix, F_{SBM} the FLOPS count for the sparse binary matrix, N the epoch length (512), M the number of measurements per IC, I the number of independent components retained and d the number of nonzero elements in each column of $\mathbf{\Phi}$.

We compare them for different compression ratios and numbers of ICs. The results are shown in Table 4.1. As can be seen, using a sparse binary

Table 4.1: FLOPS Comparison for Measurement Matrices

CR	**Number of ICs**	**3**	**4**	**6**	**8**
2	**FLOPS (Gaussian Matrix)**	$3.11 \cdot 10^6$	$3.09 \cdot 10^6$	$3.07 \cdot 10^6$	$3.04 \cdot 10^6$
	FLOPS (Sparse Binary Matrix)	$9.25 \cdot 10^3$	$1.34 \cdot 10^4$	$2.16 \cdot 10^4$	$2.98 \cdot 10^4$
3	**FLOPS (Gaussian Matrix)**	$2.06 \cdot 10^6$	$2.05 \cdot 10^6$	$2.02 \cdot 10^6$	$2.00 \cdot 10^6$
	FLOPS (Sparse Binary Matrix)	$1.03 \cdot 10^4$	$1.44 \cdot 10^4$	$2.26 \cdot 10^4$	$3.08 \cdot 10^4$
4	**FLOPS (Gaussian Matrix)**	$1.53 \cdot 10^6$	$1.52 \cdot 10^6$	$1.50 \cdot 10^6$	$1.47 \cdot 10^6$
	FLOPS (Sparse Binary Matrix)	$1.08 \cdot 10^4$	$1.49 \cdot 10^4$	$2.31 \cdot 10^4$	$3.13 \cdot 10^4$
6	**FLOPS (Gaussian Matrix)**	$1.01 \cdot 10^6$	$9.98 \cdot 10^5$	$9.70 \cdot 10^5$	$9.49 \cdot 10^5$
	FLOPS (Sparse Binary Matrix)	$1.13 \cdot 10^4$	$1.54 \cdot 10^4$	$2.36 \cdot 10^4$	$3.18 \cdot 10^4$
8	**FLOPS (Gaussian Matrix)**	$7.49 \cdot 10^5$	$7.37 \cdot 10^5$	$7.12 \cdot 10^5$	$6.87 \cdot 10^5$
	FLOPS (Sparse Binary Matrix)	$1.16 \cdot 10^4$	$1.57 \cdot 10^4$	$2.39 \cdot 10^4$	$3.21 \cdot 10^4$

matrix offers significant advantages, as it is between 1 and 2 order of magnitude more efficient than a random Gaussian matrix in terms of FLOPS.

ICA Algorithm We then compare the state-of-the-art ICA used in [39] and xFICA. To do so, we modify the numerical complexity models of [34] and then calculate the number of FLOPS (F) required for both algorithms:

$$\begin{aligned}
\mathrm{F}_{\mathrm{SOBI}} &= DNC^2/2 + 4C^3/3 + (D-1)C^3/2 \\
&\qquad + \mathrm{IOC}_{\mathrm{SOBI}} \cdot I^2[4I(D-1) + 17(D-1) + 4I + 75]/2 \\
\mathrm{F}_{\mathrm{xFICA}} &= NC^2/2 + 4C^3/3 + ICN \\
&\qquad + \mathrm{IOC}_{\mathrm{xFICA}} \cdot [2(I-1)(I-1+N) + 5N(I-1)^2/2] \\
&\qquad\quad + I + I(I-1)^3
\end{aligned}$$

where C is the number of channels (12), D the number of delay lags used for SOBI (we used $D = 100$, as recommended by the authors of [11]), and $\mathrm{IOC}_{\{\mathrm{SOBI,\ xFICA}\}}$ are the number of iterations for convergence (IOC) of each algorithm, respectively. To find the number of (IOC) for both algorithms,

Table 4.2: FLOPS Comparison Between xFICA and SOBI

Number of ICs		**3**	**4**	**6**	**8**
[39]	**IOC (SOBI)**	3.56	4.62	6.42	7.63
	FLOPS (SOBI)	$3.82 \cdot 10^6$	$3.90 \cdot 10^6$	$4.25 \cdot 10^6$	$4.98 \cdot 10^6$
Proposed	**IOC (xFICA)**	22.36	38.26	77.74	118.68
	FLOPS (xFICA)	$0.218 \cdot 10^6$	$0.623 \cdot 10^6$	$2.97 \cdot 10^6$	$8.40 \cdot 10^6$
% FLOPS Saved		94.29%	84.02%	30.28%	-68.45%

we took the mean number of iterations over the 100 signals. The results are shown in Table 4.2.

Table 4.2 demonstrates that when the number of ICs is six or less, our proposed xFICA requires significantly less FLOPS than the ICA used in [39]. As the number of ICs decreases, our method offers larger and larger energy savings.

4.5 Discussion

This chapter addresses the problem of efficiently compressing EEG signals in wireless body sensor network applications, where efficiency is measured in terms of compression ratio, reconstruction accuracy, and energy consumption. It proposes the use of compressed sensing after preprocessing the raw data with an energy-efficient independent component analysis method. ICA improves the reconstruction accuracy by exploiting the interchannel correlations. It was demonstrated that our system provides significant energy savings as compared to the state-of-the-art method, which also uses an ICA preprocessing block. As well, for a fixed compression ratio, our system achieves similar NMSE performance as the state-of-the-art method, which is much better than that achieved by CS only.

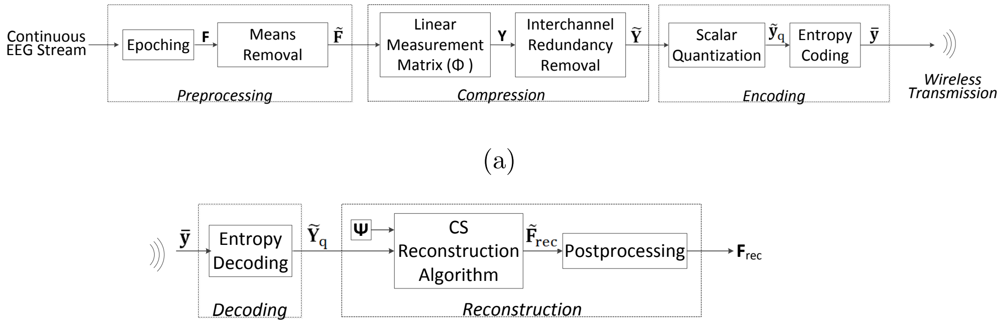
However, the approach suggested still suffers from two important drawbacks. The first one is that the proposed approach only works for a small number of EEG channels. Also, despite the fact that xFICA is more energy-efficient than other ICA algorithms, the computational complexity incurred at the sensor node is still too high for practical systems. In general, it is questionable as to whether ICA is a practical strategy in the context of WBSNs. As such, in the next chapter we investigate another strategy to exploit the interchannel correlations.

Chapter 5

An Energy-Efficient, Complete Compressed Sensing Framework for the Compression of EEG Signals

5.1 Problem Description

In this chapter, we build on the work of previous chapters and propose a novel CS framework that takes advantage of the inherent structure present in EEG signals (both temporal and spatial correlations) to improve the compression performance. To the best of our knowledge, this is also the first time that CS frameworks are compared with other state-of-the-art compression frameworks for EEG compression in WBSNs. It is also the first study where different types of EEG signals representing a variety of applications are used to test the performance of the proposed and existing frameworks, thus providing a more robust answer to the usefulness and validity of the systems.



(a)

(b)

Figure 5.1: Block diagrams of (a) the sensor node and (b) the server node for the proposed CS-based framework

This chapter is organized as follows. Sections 5.2 and 5.3 describe our framework and briefly introduce the benchmarking frameworks, respectively. Section 5.4 presents our results. Finally, section 5.5 provides a short analysis.

5.2 Methods

This section introduces the framework we developed to efficiently compress EEG signals using low energy consumption. Below we present the different blocks and algorithms that make up our proposed system. We will discuss the preprocessing, the compression, the encoding, the wireless transmission, the decoding and the reconstruction. A block diagram of the proposed system is shown in Fig. 5.1.

5.2.1 Preprocessing

As in the case of the simple framework of Chapter 3, the data is first epoched, with $N = 512$. For every epoch, the data is written in matrix form as $\boldsymbol{F}_{N\times C} = [\boldsymbol{f}_1|\boldsymbol{f}_2|\dots|\boldsymbol{f}_C]$, where C is the number of channels and $\boldsymbol{f}_i$ is the EEG data from one channel.

The mean of each channel is then removed. The resulting matrix is $\tilde{\boldsymbol{F}} = [\tilde{\boldsymbol{f}}_1|\tilde{\boldsymbol{f}}_2|\dots|\tilde{\boldsymbol{f}}_C]$. The means will be added back in the reconstruction phase. Removing the means leads to a higher compression ratio because the interchannel redundancy removal module (as will be discussed later) performs better on de-meaned EEG signals. It also reduces the total range of the signals, making it easier to quantize and encode them.

5.2.2 Compression

To compress the de-meaned EEG signals contained in one epoch, we first take their linear random projections and then apply an interchannel redundancy removal module.

Measurement Matrix ($\boldsymbol{\Phi}$)

As for the simple framework of Chapter 3, we use a fixed sparse binary sensing matrix $\boldsymbol{\Phi}$ with $d = 8$ nonzero entries in each column. The same matrix $\boldsymbol{\Phi}$ is used for each channel. After applying the measurement matrix to the de-meaned EEG signals, we obtain the compressed measurements: $\boldsymbol{Y}_{M\times C} = [\boldsymbol{y}_1|\boldsymbol{y}_2|\dots|\boldsymbol{y}_C] = [\boldsymbol{\Phi}\tilde{\boldsymbol{f}}_1|\boldsymbol{\Phi}\tilde{\boldsymbol{f}}_2|\dots|\boldsymbol{\Phi}\tilde{\boldsymbol{f}}_C]$

Interchannel redundancy removal

Because all EEG channels collect information related to the same physiological signal, there exist large interchannel correlations amongst them. Indeed, channels that are spatially close to one another tend to have signals that are highly correlated. Because we use the same measurement matrix for each channel, the linear projections of the signals of these channels are also correlated. To remove the redundancies inherent in the multi-channel EEG acquisition, we compute the differences between the compressed values of channel pairs (i.e. between $\boldsymbol{y}_i, \boldsymbol{y}_j$ for some index pairs (i, j) that we carefully select). To select the best channel pairs (i.e. those that will lead to the smallest differences), we run an algorithm that finds these pairs at the server node. The server node transmits the channel pairs to the sensor node. These pairs are then used to compute the differences for the next epoch. The redundancy removal algorithm is summarized in Fig. 5.2.

The threshold T controls the minimum acceptable correlation between channel pairs. That is, any channel pair whose correlation falls below T will not be selected. If, for a certain value of T, C or more pairs have correlations above T and these channel pairs are non-redundant (i.e. they are all linearly independent), T is in fact meaningless. However, if this is not the case, T determines the number of channel pairs i.e. the number of differences to be taken. We selected $T = 0.6$, as this was the value that gave us the best results experimentally. Also note that the selected channel pairs are wirelessly sent to the sensor node in a ‘predictive’ manner - of course, the server node does not know what the compressed values of the current

1. Compute $\boldsymbol{R}$, the matrix of correlation coefficients of the current epoch. Each entry of $\boldsymbol{R}$ is calculated as follows:

   $$\boldsymbol{R}_{i,j} = \frac{\mathrm{cov}(\boldsymbol{y}_i, \boldsymbol{y}_j)}{\sigma(\boldsymbol{y}_i)\sigma(\boldsymbol{y}_j)}$$

   where $\mathrm{cov}(\boldsymbol{a}, \boldsymbol{b})$ is the covariance between $\boldsymbol{a}$ and $\boldsymbol{b}$, and $\sigma(\boldsymbol{a})$ is the standard deviation of $\boldsymbol{a}$.

2. Subtract the identity matrix $\boldsymbol{I}$ from $\boldsymbol{R}$ (because the autocorrelations are not meaningful).

3. Define a loop index k that will be used to track the number of differences taken. Let $k = 1$.

4. Find $R_{\max}$, the maximum absolute value in matrix $\boldsymbol{R} - \boldsymbol{I}$. The row and column indices (i_k, j_k) corresponding to $R_{\max}$ correspond to the 2 channels used for the difference. Remove the channel pair (i_k, j_k) from the correlation pool.

5. Check that the selected channel pair is not redundant (i.e. it is linearly independent from the previously selected pairs when we look at it as a system of equations). If it is redundant, discard the current pair. If it is not, keep the current pair and increment k by 1.

6. Repeat steps 4 and 5 while $k < C$ or while $R_{\max} < T$, where C is the number of channels and T is a user-defined threshold.

7. If $k < C$, we have run out of good channel pairs and will thus need to pick individual channels (not a difference) to complete our system of equations. Choose a 'pair' $(i_k, 0)$ such that channel i_k is the channel with the smallest variance that is linearly independent from the previously selected pairs. Increment k by 1. Repeat until $k = C$.

8. Wirelessly transmit the selected channel pairs (i_k, j_k) , $k = 1$ to C to the sensor node. These pairs will be used to compute the differences in the next epoch.

Figure 5.2: The interchannel redundancy removal algorithm.

epoch are before they are transmitted, and so the channel pairs are based on the compressed values from previous epochs.

We observed experimentally that for a given dataset, the best channel pairs are mostly stable over time (i.e. the best pairs from one epoch to another are fairly constant). As such, there is no need to repeatedly transmit the best channel pairs. In practice, the best approach might be to periodically update the best channel pairs, in case there has been a shift in the signals.

The sensor node is only responsible for calculating the differences, which is computationally inexpensive. They are calculated for $k = 1$ to C as follows:

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{y}_{i_k} - \text{sign}(R_{i_k,j_k})\boldsymbol{y}_{j_k}.$$

After this stage, we obtain a matrix $\tilde{\boldsymbol{Y}}_{M\times C} = [\tilde{\boldsymbol{y}}_1|\tilde{\boldsymbol{y}}_2|\ldots|\tilde{\boldsymbol{y}}_C]$ whose columns are the difference signals.

Fig. 5.3 shows the probability density functions (PDFs) of the original measurement signals and of the difference signals, taken over 1000 random windows of length 512 of the datasets described at the start of Section 5.4. As can be seen, the pdf of the difference signals has the desirable property of being narrower than that of the original measurements. This makes it easier to encode the values (it requires fewer bits to do so), which results in a higher compression ratio.

5.2.3 Encoding

After removing the interchannel redundancies, the signals are quantized and an entropy coding scheme is applied. The output of the encoding module is a stream of bits to be transmitted wirelessly.

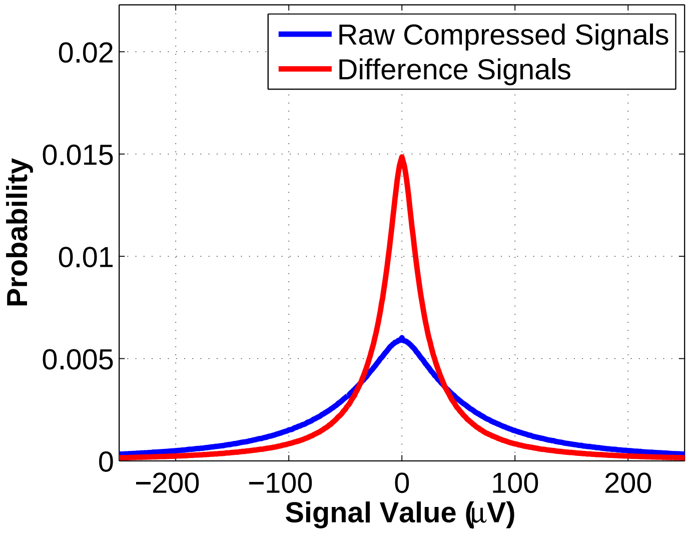


Figure 5.3: pdf of the raw CS measurements and of the difference signals

For quantization, the columns of $\tilde{\boldsymbol{Y}}^T$ are first stacked as a vector:

$$\tilde{\boldsymbol{y}}_{MC\times 1} = \mathrm{vec}(\tilde{\boldsymbol{Y}}^T) =$$

$$[\tilde{\boldsymbol{y}}_1(1)\tilde{\boldsymbol{y}}_2(1)\ldots\tilde{\boldsymbol{y}}_C(1)\tilde{\boldsymbol{y}}_1(2)\tilde{\boldsymbol{y}}_2(2)\ldots\tilde{\boldsymbol{y}}_C(2)\ldots\ldots\tilde{\boldsymbol{y}}_1(M)\tilde{\boldsymbol{y}}_2(M)\ldots\tilde{\boldsymbol{y}}_C(M)]^T.$$

This type of vectorization ensures that the compressed samples of a single channel are interleaved i.e. that they do not all come in one burst. After quantization, the resulting vector is $\tilde{\boldsymbol{y}}_q = Q(\tilde{\boldsymbol{y}})$ where Q is the scalar quantization operator. Assuming that the original signals are represented using 12 bits, we use a 15-bit quantization (to account for the increase in signal range caused by taking the linear projections).

Entropy coding is then applied on $\tilde{\boldsymbol{y}}_q$ as this further reduces the number of bits required to represent the signals. As seen in Fig. 5.3, the distribution of the difference signals $\tilde{\boldsymbol{y}}$ is definitely not uniform; in fact, it approximately has the shape of a Gaussian. We exploit this fact to achieve greater compression. Huffman encoding is known to perform well in such cases, so it is chosen as the entropy coding strategy. The codebook is generated offline using the information from Fig. 5.3 and is then stored in memory at the sensor node. We thus obtain a vector $\bar{\boldsymbol{y}} = H(\tilde{\boldsymbol{y}}_q)$ where H is the Huffman encoding operator. Note that for a fixed value for M, the length of $\bar{\boldsymbol{y}}$ varies slightly from one epoch to another, depending on the encoding.

5.2.4 Wireless Transmission

After compression and encoding, the EEG signals are wirelessly transmitted from the sensor node to the server node. As in [36], we assume that each data packet has a size of 127 bytes, of which 13 bytes are the MAC overhead. We simply form the data packets by splitting the vector $\bar{\boldsymbol{y}}$ in consecutive segments of 114 bytes. Note that we make no further attempt to model the wireless channel and that we do not study different modulation schemes for transmission.

5.2.5 Decoding

Upon receiving the encoded compressed EEG signals $\bar{\boldsymbol{y}}$ (assuming that we are dealing with a perfect channel), the server would first decode them. This is a straightforward decoding operation: $\tilde{\boldsymbol{y}}_q = H^{-1}(\bar{\boldsymbol{y}})$ where H^{-1} is the Huffman decoding operator. We then form an $M \times C$ matrix: $\tilde{\boldsymbol{Y}}_q = [\tilde{\boldsymbol{y}}_{q1}|\tilde{\boldsymbol{y}}_{q2}|\ldots|\tilde{\boldsymbol{y}}_{qC}]$.

5.2.6 Reconstruction

The final step is to reconstruct the original EEG signals from the decoded measurements $\tilde{\boldsymbol{Y}}_q$. Before applying the CS reconstruction algorithm, we first reverse the effect of the interchannel redundancy module to obtain $\boldsymbol{Y}_q$, the original compressed measurements (before their differences were taken). Once this is done, we can do CS reconstruction. As in Chapter 3, the sparsifying dictionary $\boldsymbol{\Psi}$ is a Gabor dictionary and the chosen reconstruction algorithm is the Basis Pursuit Denoise algorithm implemented in the SPGL1 package.

Postprocessing

The final step is to add back the means of each EEG channel.

5.3 State-of-the-Art Systems

A brief overview of the state-of-the-art systems that will be used to compare our results with is also given.

Given that the problem under investigation (EEG compression in a WBSN setting) has only started to be studied recently, there is not a large body of existing literature around it, and identifying a proper benchmark is challenging. EEG compression has been studied quite extensively in the last two decades but algorithms have not generally been designed for low energy consumption or for implementation on simple hardware. As such, some frameworks can achieve high compression ratios, but they require too many computations or some operations that are too complex given our target sensor hardware, making these frameworks prohibitive in WBSNs. Similarly, WBSNs started to gain momentum in the last decade but few of them have addressed their applications to EEG signals. As a result, there only exist very few papers that have explicitly studied the problem of EEG compression for WBSNs. In order to identify state-of-the-art systems to compare the performance of our proposed framework with, it is therefore necessary to extrapolate the results of this previous research in order to identify schemes that can offer good performance in the context we are interested in.

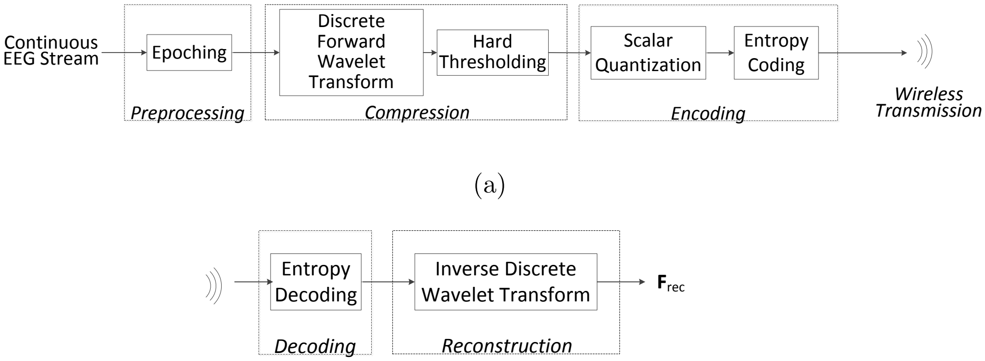
We use the following requirements for selecting state-of-the-art systems to compare our system with:

- low energy consumption: the system must not have a high computational requirement at the sensor node in order to conserve energy;
- simple sensor hardware: the system must be simple enough to operate using inexpensive hardware;
- high compression ratio: the achievable compression ratio must be high enough to justify the extra computations carried at the sensor node (as compared to wirelessly sending raw, uncompressed data).

Based on these requirements, we have selected 2 state-of-the-art compression methods to compare our proposed framework to. These are described in some details below.

5.3.1 JPEG2000-Based Compression

The JPEG2000-based EEG compression scheme was proposed in [27]. Amongst the non-CS methods, it is one of the simplest yet most effective compres-



(a)

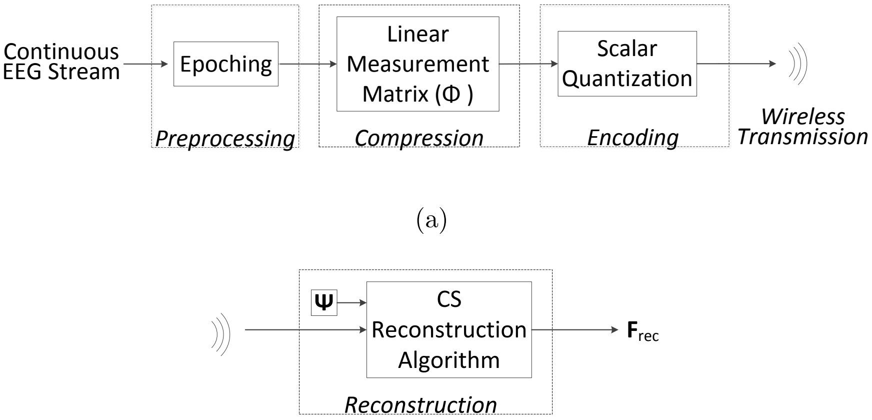
(b)

Figure 5.4: Block diagrams of (a) the sensor node and (b) the server node for the JPEG2000-based framework

sion schemes. It is a simple wavelet compression algorithm adapted from the JPEG2000 algorithm used for image compression. Its block diagram is shown in Fig. 5.4.

The wavelet coefficients of the raw EEG signals are first computed using the Cohen-Daubechies-Feauveau (CDF) 9/7 discrete wavelet transform. Depending on the desired compression ratio, a hard threshold is then applied so that only the largest wavelet coefficients of the signal are kept. These large coefficients are then encoded using arithmetic coding. At the server side, the received signals are decoded and then reconstructed by taking the inverse wavelet transform. In our implementation, we used a decomposition level of 7 for the wavelet transform. We determined this level experimentally, as it gave the lowest reconstruction error; furthermore, using more levels did not provide an improvement in reconstruction accuracy.

In contrast to CS schemes, this type of algorithm is adaptive in the



(a)

(b)

Figure 5.5: Block diagrams of (a) the sensor node and (b) the server node for the BSBL framework

sense that it relies on the exact knowledge of the signal to find its largest coefficients. Also, the bulk of the computations is done at the sensor node. We will discuss the implications of these later on.

5.3.2 BSBL CS Compression

Block-Sparse Bayesian Learning (BSBL) is a reconstruction method that was proposed in [64] and then applied to EEG signals in [63]. Its block diagram is shown in Fig. 5.5.

The main difference between the BSBL framework and our framework is in how the reconstruction is carried. Whereas we assume that EEG signals are sparse in a transform domain (Gabor frames in our case) and use that information to reconstruct the original signals, the BSBL framework does not rely on this assumption. BSBL was first proposed to reconstruct signals that have a block structure - that is, signals that have few blocks containing nonzero entries, and the remainder of the blocks containing only zeros. It was then experimentally shown that the BSBL scheme was effective in reconstructing signals even if their block partition boundaries are unknown or if they do not have any clear block structure. Raw EEG signals fall in this last category.

To carry out the reconstruction, the BSBL framework uses a "sparsifying" matrix $\mathbf{\Psi}$ which is an inverse discrete cosine transform (DCT) operator (EEG signals are not sparse in the DCT basis but as mentioned previously, BSBL does not rely on sparsity). We used the Bounded-Optimization variant of the BSBL family (BSBL-BO) with a block partition of 24. The experiments we carried found this value to be the optimal partition size.

5.4 Results

We now introduce the experimental results. We start by presenting the datasets used as well as defining the performance metrics selected. We then carry out experiments to evaluate the energy consumption and reconstruction accuracy.

5.4.1 Datasets

In order to assess the performance of the different algorithms on a wide range of EEG cases, we selected 3 different databases.

The first one is Dataset 1 of the BCI Competition IV [12]. This dataset was recorded from healthy subjects or generated artificially and contains the data of 7 patients. The recording was made using 59 EEG channels per subject at an initial sampling rate of 1000Hz.

The 2 other databases are from Physionet [24]. The first consists of seizure data of 22 pediatric subjects. The recordings contain 21 EEG channels at a sampling rate of 256Hz [51]. The second database is a collection of 108 polysomnographic recordings of different sleep disorders monitoring. Each recording contains between 5 and 13 EEG channels sampled at 256Hz [57].

For each dataset, we used a resolution of 12 bits for the original EEG signals. Note that some files in these datasets contain dead channels (i.e. channels where the output is always 0). We removed these channels from our analyis, as they provide no useful information. We also resampled the data to 128Hz so as to provide a realistic sampling frequency in the context of telemedicine as well as ensure uniformity between the datasets. Non-overlapping windows of length $N = 512$ were used in our experiments. The experiments were carried using 100 randomly extracted windows from each dataset.

5.4.2 Performance Metrics

To quantify the compression performance, we used the compression ratio (CR), which is redefined here as follows:

$$\mathrm{CR} = \frac{b}{\hat{b}} \tag{5.1}$$

where b is the total number of bits in the original (uncompressed signal) and $\hat{b}$ is the total number of bits in the compressed signal.

To test the reconstruction quality, we used the NMSE metric, as defined in Eq. 3.2.

5.4.3 Energy

We study the energy performance at the sensor node of the two CS schemes (BSBL and the proposed framework) and the JPEG2000 scheme. As mentioned previously, energy consumption is paramount in WBSNs due to the limited battery life at the sensor node. We also note that energy consumption is only important at the sensor node. For the server node, we assume an unlimited energy supply. We study the three broad classes of energy at the sensor node: 1) sensing energy (energy required to acquire and digitize the EEG samples), 2) computing energy (energy required to carry out the computations at the sensors) and 3) transmission energy (energy required to wirelessly transmit the data to the server).

After being acquired and digitized, the signals are compressed, encoded and transmitted. The sensing energy is constant for all schemes since all samples must first be acquired. It was also shown in [62] that the sensing energy is small (about an order of magnitude smaller) compared to the other two classes when using ultra-low-power sensors. As such, we can safely omit it in our analysis. For a fixed compression ratio, the transmission energy is the same for all schemes because the number of bits to be transmitted would be the same. We thus focus our efforts on the computation energy, which is mainly the energy required for compression and encoding.

We implemented the code in Embedded C and simulated it on a MICAZ target platform using the AVRORA simulator ([58]). We evaluated the performance based on the total cycle count, the run time and the energy consumption. To estimate the transmission energy, we rely on the experimental work done in [36], in which it was calculated that the transmission of each packet requires 524.72 μJ. The results are presented in the top part of Table 5.1. They refer to one epoch for one EEG channel so that the difference in the number of channels in the different datasets does not impact the presented results.

Table 5.1: Energy Consumption and Reconstruction Accuracy for all Frameworks

CR		**8:1**	**6:1**	**5:1**	**4:1**	**3.5:1**	**3:1**	**2.5:1**	**2:1**
Energy Consumption									
Cycle Count	**Proposed**	502,200	507,690	512,448	519,402	524,526	531,480	542,757	557,477
	BSBL	429,486	429,486	429,486	429,486	429,486	429,486	429,486	429,486
	JPEG2000	4,908,731	4,908,367	4,908,055	4,907,535	4,907,171	4,906,703	4,905,975	4,904,831
Run Time	**Proposed**	68.11 ms	68.86 ms	69.50 ms	70.45 ms	71.14 ms	72.08 ms	73.61 ms	75.61 ms
	BSBL	58.25 ms	58.25 ms	58.25 ms	58.25 ms	58.25 ms	58.25 ms	58.25 ms	58.25 ms
	JPEG2000	665.8 ms	665.7 ms	665.7 ms	665.6 ms	665.6 ms	665.5 ms	665.4 ms	665.3 ms
Computation Energy	**Proposed**	1.55 mJ	1.56 mJ	1.58 mJ	1.60 mJ	1.61 mJ	1.64 mJ	1.67 mJ	1.72 mJ
	BSBL	1.32 mJ	1.32 mJ	1.32 mJ	1.32 mJ	1.32 mJ	1.32 mJ	1.32 mJ	1.32 mJ
	JPEG2000	15.11 mJ	15.11 mJ	15.11 mJ	15.11 mJ	15.11 mJ	15.11 mJ	15.10 mJ	15.10 mJ
Computation + Transmission Energy	**Proposed**	1.99 mJ	2.15 mJ	2.29 mJ	2.48 mJ	2.62 mJ	2.82 mJ	3.08 mJ	3.49 mJ
	BSBL	1.76 mJ	1.91 mJ	2.03 mJ	2.20 mJ	2.33 mJ	2.50 mJ	2.73 mJ	3.09 mJ
	JPEG2000	15.55 mJ	15.70 mJ	15.82 mJ	15.99 mJ	16.12 mJ	16.29 mJ	16.51 mJ	16.87 mJ
Reconstruction Accuracy									
Mean NMSE (BCI Dataset)	**Proposed**	0.227 ± 0.147	0.145 ± 0.103	0.105 ± 0.077	0.065 ± 0.050	0.044 ± 0.036	0.025 ± 0.021	0.009 ± 0.009	0.001 ± 0.003
	BSBL	0.455 ± 0.205	0.312 ± 0.191	0.239 ± 0.159	0.180 ± 0.120	0.148 ± 0.103	0.114 ± 0.083	0.086 ± 0.063	0.059 ± 0.048
	JPEG2000	0.101 ± 0.079	0.079 ± 0.064	0.066 ± 0.055	0.053 ± 0.045	0.045 ± 0.040	0.037 ± 0.034	0.028 ± 0.026	0.018 ± 0.018
Mean NMSE (Seizure Dataset)	**Proposed**	0.552 ± 0.160	0.388 ± 0.148	0.298 ± 0.130	0.202 ± 0.107	0.149 ± 0.084	0.101 ± 0.067	0.056 ± 0.047	0.021 ± 0.025
	BSBL	0.607 ± 0.136	0.452 ± 0.144	0.371 ± 0.142	0.275 ± 0.127	0.234 ± 0.122	0.181 ± 0.106	0.140 ± 0.091	0.095 ± 0.070
	JPEG2000	0.159 ± 0.108	0.119 ± 0.098	0.097 ± 0.090	0.074 ± 0.080	0.062 ± 0.073	0.050 ± 0.065	0.038 ± 0.055	0.025 ± 0.042
Mean NMSE (Sleep Dataset)	**Proposed**	0.340 ± 0.148	0.192 ± 0.124	0.122 ± 0.101	0.062 ± 0.073	0.039 ± 0.057	0.022 ± 0.040	0.009 ± 0.020	0.003 ± 0.008
	BSBL	0.472 ± 0.158	0.300 ± 0.154	0.211 ± 0.125	0.134 ± 0.109	0.096 ± 0.092	0.066 ± 0.082	0.040 ± 0.059	0.020 ± 0.040
	JPEG2000	0.071 ± 0.061	0.043 ± 0.047	0.030 ± 0.040	0.018 ± 0.030	0.013 ± 0.025	0.009 ± 0.020	0.006 ± 0.014	0.003 ± 0.009

As expected, the CS-based schemes significantly outperform JPEG2000 in every category. In fact, even if we use the highest compression ratio for JPEG2000, it will never qualify from an energy perspective when compared with CS. JPEG2000, being an adaptive scheme, relies on a high amount of computations in order to efficiently compress the data while the random projections of CS require a significantly smaller lower number of computations. It is also interesting to note that small gains (decrease in the number of cycles, run time and energy consumption) are obtained by the proposed framework when the compression ratio increases. This is due to a reduction in redundancy removal and encoding computations. We note that BSBL is slightly more energy-efficient than the proposed framework. However, as we will see shortly that comes at the expense of a worse reconstruction accuracy. We will argue that the small increase in energy consumption is well worth it in this case.

Another interesting consideration would be to look at the comparison between sending compressed and uncompressed data. Sending uncompressed data requires 3.53 mJ of energy per channel (which is the energy required for wireless transmission, as no computations are carried out prior to transmission). We thus note that CS is more advantageous at any compression ratio, whereas JPEG2000 always consumes more energy than even sending uncompressed data.

5.4.4 Reconstruction Performance

We now look at the reconstruction performance of the three frameworks over the three datasets. We look at both noiseless and noisy cases.

Noiseless Case

For this experiment, we assume that the measurements are noise-free (that is, the output of the entropy decoding block is the same as the output of the interchannel redundancy removal module), and only look for the algorithms' ability to accurately reconstruct the original signals at different compression ratios. The reconstruction performance of the different frameworks over the three datasets is shown at the bottom of Table 5.1.

In terms of the NMSE, the proposed framework systematically outperforms BSBL. As expected, the reconstruction accuracy of the JPEG2000 framework is generally better than that of the CS-based frameworks, especially at high compression ratios. This is not a surprise because the JPEG2000 framework adaptively encodes the EEG signals, using full knowledge about the structure of the signal when finding its largest coefficients. In contrast, at the sensor nodes, the CS frameworks do not make any attempt at comprehending the signal at play; instead, they nonadaptively take a subset of linear random projections, and only rely on the signal structure on the server side when carrying out the reconstruction. As such, it is obvious that an adaptive algorithm will perform better than a nonadaptive one. However, it is interesting to note that as the compression ratio decreases, the gap in the reconstruction error quickly shrinks. At lower compression ratios, the proposed framework can even outperform the adaptive JPEG2000 scheme.

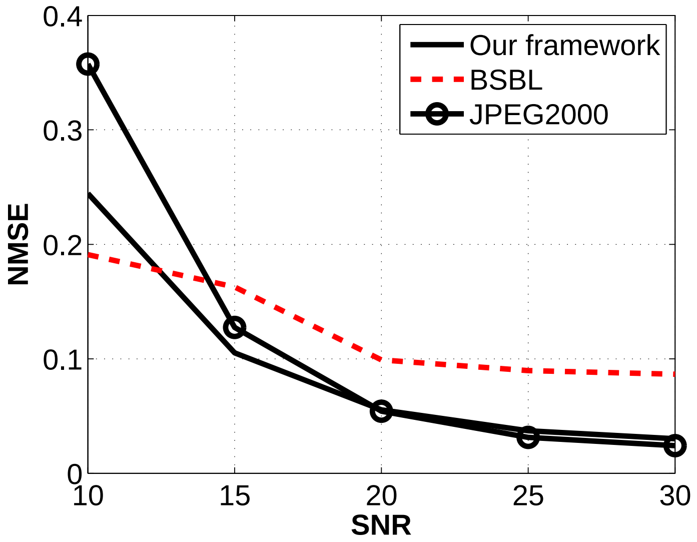


Figure 5.6: Reconstruction performance under Gaussian noise for a fixed CR (2.5:1), with varying SNR

Noisy Case

We then show that CS-based schemes are more robust than JPEG2000 to Gaussian noise and packet loss. Such studies have so far been omitted in EEG-based WBSN studies.

Gaussian Noise For this experiment, we arbitrarily fix the compression ratio to 2.5:1, and vary the signal-to-noise ratio (SNR) by adding Gaussian noise with varying standard deviations. The results are shown in Fig. 5.6.

The JPEG2000 framework is the most affected by Gaussian noise, especially when the SNR is low. Comparing our framework with BSBL, we notice that BSBL performs better at low SNRs, whereas our framework performs better for SNRs higher than 13dB.

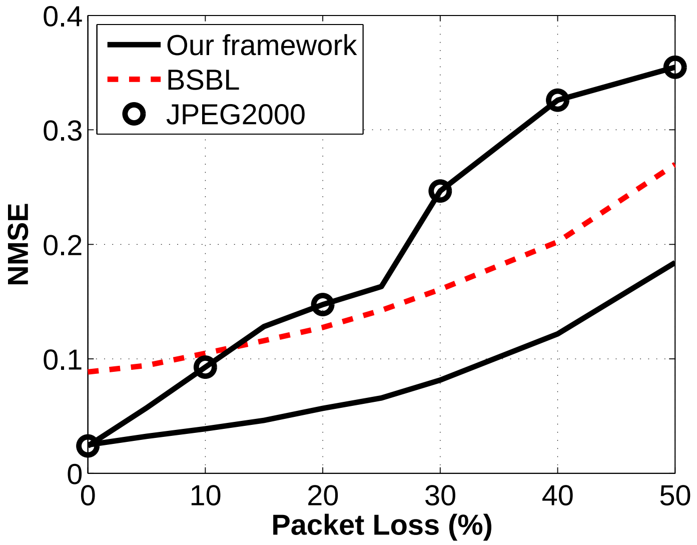


Figure 5.7: Reconstruction performance under packet loss for a fixed CR (2.5:1)

Packet Loss We then test the impact of packet loss. Again, we select a compression ratio of 2.5:1 and vary the percentage of packets lost through a noisy channel. This is done by randomly choosing the lost packets which then cannot be used to reconstruct the original signal. The results are shown in Fig. 5.7.

The important thing to note about this figure is the relationship between reconstruction accuracy and packet loss. What we notice is that for both CS-based methods, the slope of the line is much less steep than for the JPEG2000 framework. This property is desirable since it leads to a more graceful degradation in signal quality. In fact, we can see that as soon as packet loss happens, JPEG2000 becomes worse than the proposed framework. Similarly, when the percentage of packets lost go above 12%, BSBL performs better than JPEG2000. Knowing that in WBSN the packet loss rate can be high, we can see that CS becomes an attractive solution even from the perspective of reconstruction accuracy. It is also important to understand the possible implications of packet loss for the different frameworks. Because JPEG2000 is adaptive and only sends the largest coefficients, we run the risk of losing the largest coefficients, and we have no control over that - it is simply a matter of luck as to which coefficients are affected. In contrast, because the CS measurement operator takes nonadaptive random projections, no resulting measurement bears more weight than another one. In a way, this provides a safety net: even if we have a noisy channel with a high packet loss rate, we know that in all cases the degradation in signal quality will be proportional to the noise in the channel, and won't depend on the timing or location of these losses.

5.5 Discussion

In this chapter, we proposed a novel CS framework that exploits both the temporal correlations and the spatial correlations to efficiently compress EEG signals in WBSNs. On the energy front, our proposed CS framework is between 5 and 8 times more energy-efficient than the JPEG2000 framework in terms of sensor computations and wireless transmission. We also show that our method achieves a better reconstruction quality than the state-of-the art BSBL method which also uses CS. This was also the first study in which a wide range of EEG signals are used to validate the performance of different frameworks.

We wish to reiterate the advantages of using a CS framework in the context of WBSNs. Recall that the core purpose of WBSNs is to provide a *simple and practical tool* for people to participate in their own treatment. As such, energy consumption is paramount, and we also desire to have the simplest sensors possible, for affordability. What this means is that operations carried out at the sensor node must be as basic as possible. This is exactly what CS does: the main operations at the sensor node consist of computing random projections, which happen to be simple additions when using a sparse binary sensing matrix. The computational complexity is shifted to the server node, which has a lot more computing power than the sensor node. In that sense, a CS framework is an ideal solution that fully takes advantage of the strengths and limitations of WBSNs.

Furthermore, CS provides a simple and universal encoding scheme since the random measurement matrix $\mathbf{\Phi}$ is incoherent with any sparsifying basis $\mathbf{\Psi}$ with overwhelming probability. What this means is that if for some reason we wish to change $\mathbf{\Psi}$ along the way, it does not require us to also change $\mathbf{\Phi}$. Finally, CS measurements are robust to noise. The use of a larger number of CS measurements will lead to a progressively better reconstruction. Similarly, losing a few measurements (e.g. due to wireless channel interference) leads to a progressive reconstruction degradation. This is in contrast with a wavelet compression scheme, where losing one of the large coefficients can have an important impact on the quality of the reconstruction.

Chapter 6

A Compressed Sensing Framework for BCI Systems in WBSNs

6.1 Problem Description

Brain-Computer Interface (BCI) systems have received significant attention in the last few decades. BCIs provide a direct pathway between the brain and an external interface such as a computer or a prosthesis. Such devices can prove to be life-changing for disabled people by allowing them a more natural interaction with their environment despite their physical limitations.

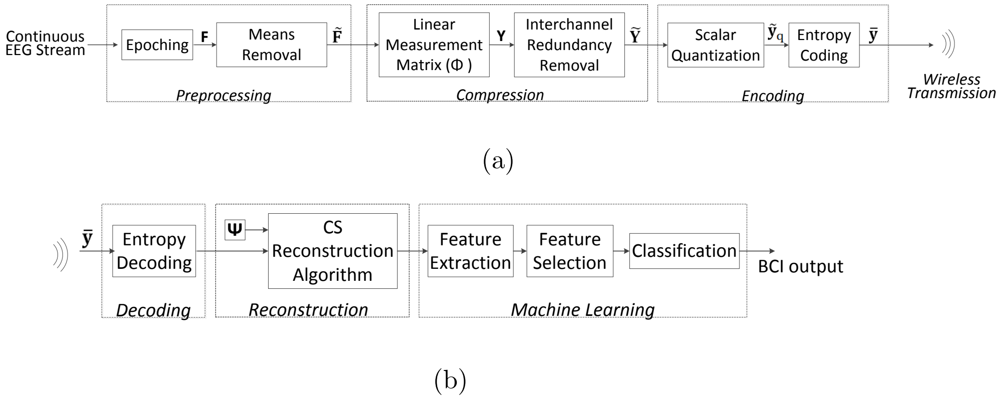
One widely used brain signal in BCIs is the EEG. These signals are popular because they provide high time resolution, a necessary feature to build a practical BCI [37]. A BCI would use the EEG signals to detect patterns associated with a mental task performed by a person (such as attempting to move a finger or visualizing some arithmetic task). The person would use such a simple mental task to operate a wheel chair, switch a light off or communicate with a caregiver, for instance. One popular mental task used in BCIs is what is called motor imagery, in which the user performs an imagined motion (e.g. flexing the left or right hand). It has been shown that motor imagery can be reliably identified when sensors (electrodes) are located over the sensorimotor area of the cortex [47].

There are two main categories of motor imagery BCI systems: synchronous and asynchronous (also called 'self-paced') systems. A synchronous BCI provides a time window during which the BCI can be operated (i.e. the system gives a cue to users to let them know when they can control it). The user's mental commands must be sent during that time interval specified by the system, and the brain output is ignored at all other times. In contrast, a self-paced BCI allows a person to control a device or an object whenever they wish. The system must figure out whether the user is trying to control the interface (called 'intentional control') or not (called 'idle' or 'no control'). This additional flexibility is of course desirable from a user experience perspective but it comes at the cost of increased complexity. In this work, we focus on synchronous BCIs.

In many cases, it is important to provide a BCI system that is minimally obtrusive to users and that allows them to carry on their daily tasks as normally as possible. This is when the concept of WBSN becomes interesting. We have shown in the past chapters that a CS framework provides an interesting solution in this setting.

In this chapter, we study a practical application of the framework developed in Chapter 5. We apply our framework to compress EEG signals that are then (after decompression) used to control a BCI.

This chapter is organized as follows. Section 6.2 presents the different



(a)

(b)

Figure 6.1: Block diagrams of (a) the sensor node and (b) the server node for the proposed CS-based BCI framework

building blocks of the framework. Section 6.3 shows the performance of the system through different experiments, and Section 6.4 discusses the obtained results.

6.2 Methods

We now present our proposed framework, shown in Fig. 6.1. The operations at the sensor node and the reconstruction part are exactly the same as for those of Chapter 5. As such, we refer the reader to Section 5.2 for further details. The only difference is that here we use an epoch length of $N = 128$. This is done to ensure that the BCI system can make classification decisions at a practical rate. We could also afford to reduce the value of d from 8 to 4.

After the reconstruction stage, we carry out machine learning steps to build the BCI. The features are extracted based on the synchronous cues given by the system. For each cue, we extract the window that starts 0.5s after the cue and of length 2s. This window has been shown to work well when working with motor imagery EEG data. We then use Common Spatial Patterns (CSP), after first having bandpass filtered the selected data between 8Hz and 35Hz. CSP is a well-known algorithm which looks for features that maximize the variance with respect to one class while minimizing it with respect to the other one [49]. We extract the log-variance of the CSPs to use as our features. The feature selection method we selected is simple: we select the $2m$ CSPs that best discriminate between classes (based on their eigenvalues). We used $m = 3$. The selected features are then fed to a Linear Discriminant Analysis (LDA) classifier. LDA is a simple classifier that is fast and that works well for BCIs [9].

6.3 Results

6.3.1 Data Used

We used 2 publicly available motor imagery datasets to test the performance of our proposed method:

- BCI Competition IV, dataset # 1 [12]: This data was recorded from healthy subjects (subjects a, b, f and g) or generated artificially (subjects c, d and e). The initial recording was made using 59 EEG channels per subject at a sampling rate of 1000Hz. The subjects performed a motor imagery task (imagining a left hand movement, a right hand movement or a foot movement). Only 2 of the 3 classes were selected for each subject. For each subject, there are 200 cues (100 for each

Table 6.1: Classification Performance for BCI Competition IV, dataset # 1

CR	**1:1**	**2:1**	**3:1**	**4:1**	**6:1**	**8:1**
Subject ‘a’	70.0	63.2	68.4	61.9	55.1	56.3
Subject ‘b’	56.2	57.9	54.4	51.0	50.1	56.1
Subject ‘c’	64.6	52.6	53.2	56.5	56.5	52.1
Subject ‘d’	81.0	75.1	70.6	58.9	55.1	57.4
Subject ‘e’	87.3	87.5	86.0	77.9	72.1	64.6
Subject ‘f’	72.5	72.1	69.2	62.2	63.6	55.4
Subject ‘g’	82.3	84.0	82.7	80.2	73.4	73.7
Mean ± STD	**73.4 ± 10.9**	**70.3 ± 13.1**	**69.2 ± 12.5**	**64.1 ± 10.9**	**60.8 ± 9.0**	**59.3 ± 7.4**

class). Out of the original 59 channels we selected 12 channels in the sensorimotor area of the cortex: F1, FZ, F2, FC3, FC1, FCZ, FC2, FC4, C3, C1, CZ, C2. We also resampled the data to 256Hz.

- BCI Competition IV, dataset # 2a [42]: This data was recorded from 9 subjects performing a motor imagery task (imagining one of 4 movements: left hand, right hand, both feet, tongue). We only look at left and right hands (2 classes). For each subject, there are 72 instances of each class, for a total of 144 cues. There is a total of 22 EEG channels, and the sampling rate is 250Hz.

6.3.2 Compression and Classification Performance

We first look at the classification performance when the system is subjected to different compression ratios. We tested 6 different compression ratios: 1:1 (uncompressed - this is the baseline), 2:1 (50% compression), 3:1 (67% compression), 4:1 (75% compression), 6:1 (83% compression) and 8:1 (87.5% compression). To obtain the classification performance, we used 10-fold cross-validation. The results are shown in Tables 6.1 and 6.2.

Table 6.2: Classification Performance for BCI Competition IV, dataset # 2a

CR	**1:1**	**2:1**	**3:1**	**4:1**	**6:1**	**8:1**
Subject 'A01'	83.0	80.2	75.9	72.3	73.2	66.2
Subject 'A02'	59.4	56.2	50.6	55.0	53.9	51.2
Subject 'A03'	94.6	93.4	92.0	90.5	87.8	91.5
Subject 'A04'	67.3	69.2	56.2	67.5	65.5	61.9
Subject 'A05'	62.7	59.2	62.9	50.8	52.9	51.5
Subject 'A06'	64.6	64.0	57.6	59.2	58.4	55.6
Subject 'A07'	73.3	75.4	67.3	59.7	54.3	53.9
Subject 'A08'	96.2	95.8	95.4	91.2	90.6	92.6
Subject 'A09'	74.2	73.8	72.7	68.2	66.5	69.7
Mean ± STD	**75.0 ± 13.5**	**74.1 ± 13.9**	**70.1 ± 15.6**	**68.3 ± 14.4**	**67.0 ± 14.3**	**66.0 ± 16.1**

As expected, there is a degradation in performance when the CR is greater than 1:1, i.e. when the data is compressed. However, we would argue that this degradation in performance is not significant.

6.3.3 Energy Analysis

We now turn our attention to the energy consumption of the proposed framework, as compared to that of sending uncompressed data. As mentioned previously, it is very important to use as little energy as possible at the sensor node since it is battery-powered. We place no constraint on the energy supply of the server node, as we assume it is wired.

The energy analysis is carried out in a manner analogous to that of Section 5.4.3. The results are presented in Table 6.3. They are normalized over one epoch for one EEG channel so that the difference in the number of channels in each dataset does not impact the presented results.

As can be seen from the results, interesting energy savings can be realized when using our proposed framework. This is especially true when the

Table 6.3: Energy Consumption Under Different Compression Ratios

CR	**Computation Energy**	**Transmission Energy**	**Total Energy**	**Energy Savings**
1:1	0 mJ	0.884 mJ	0.884 mJ	-
2:1	0.296 mJ	0.442 mJ	0.738 mJ	16.5 %
3:1	0.259 mJ	0.295 mJ	0.553 mJ	37.4 %
4:1	0.242 mJ	0.221 mJ	0.463 mJ	47.6 %
6:1	0.227 mJ	0.147 mJ	0.374 mJ	57.7 %
8:1	0.222 mJ	0.111 mJ	0.333 mJ	62.3 %

number of channels increases and when the sampling rate is high since the total absolute energy consumption of the system is compounded by these 2 factors.

6.4 Discussion

In this chapter, we applied our CS framework to efficiently compress EEG signals in WBSNs, in the context of a BCI application. By providing a simple, nonadaptive compression scheme at the sensor node, CS offers an energy-efficient solution to compress EEG signals in WBSNs. We verified the performance of the proposed compression framework by testing the classification performance of the BCI systems under different compression levels and by looking at the overall energy consumption of the sensor node. Our results showed that a CS framework leads to important energy savings at the cost of a lower classification accuracy.

Using a compressed sensing framework for WBSN-based BCI systems involves some relatively obvious tradeoffs. On the one hand, as the compression ratio increases, the classification accuracy decreases. However, at the same time the total energy consumption also decreases. It is thus a design choice to determine the appropriate compression ratio. We note that in WBSNs, energy consumption is an important consideration, which prompts us to say that in some situations, compression may be necessary.

We also note that using more sophisticated machine learning techniques could have improved the overall accuracy of our BCI system. Similarly, we could have optimized the parameters for each subject to obtain better classification performance. However, our aim was to obtain a simple system that worked well on the average case. In fact, we were only interested in the comparative performance of the BCI system under different compression ratios.

Chapter 7

Summary and Conclusions

7.1 Main Results

In this thesis, we proposed novel, energy-efficient CS frameworks that take advantage of the inherent structure present in EEG signals (both temporal and spatial correlations) to efficiently compress these signals in WBSNs.

In Chapter 3, we presented a simple CS-based framework that would become the basis for the bulk of the work in this thesis. We optimized the Gabor sparsifying dictionary and demonstrated that using a fixed sparse binary sensing matrix offered similar performances to optimal matrices while requiring far fewer computations. This framework has the advantage of being simple but does not exploit the spatial correlations between EEG channels.

In Chapter 4, we added an energy-efficient ICA preprocessing block to the simple CS framework to exploit the spatial correlations among EEG channels. We showed that the proposed framework provides significant energy savings as compared to the state-of-the-art method in [39]. As well, for a fixed compression ratio, our system achieves similar NMSE performance as the state-of-the-art method, which is much better than that achieved by the simple CS framework.

In Chapter 5, we proposed a complete, novel CS framework that exploits both the temporal correlations and the spatial correlations. On the energy front, our proposed CS framework is up to 8 times more energy-efficient than the JPEG2000 framework in terms of sensor computations and wireless transmission. The reconstruction accuracy of the JPEG2000 framework is generally better than that of the proposed framework but as the compression ratio decreases, the gap in the reconstruction error quickly shrinks. At lower compression ratios, the proposed framework can even outperform the JPEG2000 scheme. We also showed that our method achieves a better reconstruction quality than the state-of-the art BSBL method. This was the first study that compared CS frameworks with other state-of-the-art compression frameworks for EEG compression in WBSNs. It was also the first study where different types of EEG signals representing a variety of applications were used to test the performance of the proposed and existing frameworks, thus providing a more robust answer to the usefulness and validity of the systems.

Finally, in Chapter 6, we applied the framework of Chapter 5 to compress EEG signals in the context of a BCI application and evaluated its impact on the performance of the system.

7.2 Future Work

There are many possible directions for extending the work presented in this thesis. It would be interesting to implement the CS framework in hardware and test its reconstruction and energy performances in practice, under real-life conditions (instead of relying on simulations and assumptions).

It would also be interesting to look at CS reconstruction algorithms that can directly exploit the spatial correlations by jointly reconstructing the channels. Along the same lines, it may be useful to look at the analysis prior formulation for the CS reconstruction problem, as it could yield an improvement in reconstruction accuracy. These two approaches have been gaining momentum recently and have the potential to improve our CS framework.

Another avenue would be to use dictionary learning techniques to find adaptive dictionaries in which EEG signals are sparser than in our fixed optimized Gabor dictionary. We also see potential in combining a dictionary learning approach with an improved reconstruction algorithm to build a structured sparsifying dictionary, whose structure can then be exploited by the reconstruction algorithm to achieve reconstruction using fewer measurements.

It might also be worth looking for an optimal quantization and encoding strategy for CS measurements under a sparse binary sensing matrix. While this problem has been investigated for more traditional CS matrices (see e.g. [25]), it has not been explored for sparse binary sensing matrices.

Finally, on top of BCIs, other end-user systems (e.g. a seizure detection system) that use the developed compression framework could be developed and investigated.

Bibliography

[1] A. M. Abdulghani, A. J. Casson, and E. Rodriguez-Villegas. Quantifying the performance of compressive sensing on scalp EEG signals. In *3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, pages 1 –5, 2010.

[2] A. M. Abdulghani, A. J. Casson, and E. Rodriguez-Villegas. Compressive sensing scalp EEG signals: Implementations and practical performance. *Medical and Biological Engineering and Computing*, 50(11):1137–1145, 2012.

[3] A. Acharyya, K. Maharatna, and B. M. Al-Hashimi. Algorithm and architecture for n-D vector cross-product computation. *IEEE Transactions on Signal Processing*, 59(2):812–826, 2011.

[4] G. Antoniol and P. Tonella. EEG data compression techniques. *IEEE Transactions on Biomedical Engineering*, 44(2):105–114, 1997.

[5] S. Aviyente. Compressed sensing framework for EEG compression. In *IEEE/SP 14th Workshop on Statistical Signal Processing (SSP '07)*, pages 181–184, 2007.

[6] World Bank. Health expenditure, total (% of GDP). `http://data.worldbank.org/indicator/SH.XPD.TOTL.ZS`. Accessed: 23/08/2013.

[7] World Bank. Population ages 65 and above (% of total). `http://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS/countries`. Accessed: 23/08/2013.

[8] M. Barwiński. Product-based metric for Gabor functions and its implications for the matching pursuit analysis. Master's thesis, University of Warsaw, Warsaw, Poland, 2004.

[9] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch. A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural Engineering*, 4(2):R32, 2007.

[10] C. Bazan-Prieto, J. Cardenas-Barrera, M. Blanco-Velasco, and F. Cruz-Roldan. Electroencephalographic compression based on modulated filter banks and wavelet transform. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '11)*, pages 7067–7070, 2011.

[11] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines. Second-order blind separation of temporally correlated sources. In *Proceedings of the International Conference on Digital Signal Processing*, pages 346–351, 1993.

[12] B. Blankertz, G. Dornhege, M. Krauledat, K. Muller, and G. Curio. The non-invasive Berlin braincomputer interface: Fast acquisition of

effective performance in untrained subjects. *NeuroImage*, 37(2):539 – 550, 2007.

[13] B. Blankertz, G. Dornhege, M. Krauledat, K. R. Muller, V. Kunzmann, F. Losch, and G. Curio. The Berlin brain-computer interface: EEG-based communication without subject training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):147–152, 2006.

[14] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[15] E. J. Candes and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[16] J. L. Cárdenas-Barrera, J. V. Lorenzo-Ginori, and E. Rodríguez-Valdivia. A wavelet-packets based algorithm for EEG signal compression. *Informatics for Health and Social Care*, 29(1):15–27, 2004.

[17] J. Dauwels, K. Srinivasan, M.R. Reddy, and A. Cichocki. Near-lossless multichannel EEG compression based on matrix and tensor decompositions. *IEEE Journal of Biomedical and Health Informatics*, 17(3):708–714, 2013.

[18] V.R. Dehkordi, H. Daou, and F. Labeau. A channel differential EZW coding scheme for EEG data compression. *IEEE Transactions on Information Technology in Biomedicine*, 15(6):831–838, 2011.

[19] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289 –1306, 2006.

[20] Emotiv. EPOC specifications. `http://www.emotiv.com/epoc/download_specs.php`, 2012. Accessed: 29/08/2013.

[21] I. Feinberg, R. L. Koresko, and N. Heller. EEG sleep patterns as a function of normal and pathological aging in man. *Journal of Psychiatric Research*, 5(2):107–144, 1967.

[22] D. Gabor. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93(26):429–441, 1946.

[23] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.

[24] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B Moody, C.-K. Peng, and H. E. Stanley. Physiobank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[25] C.S. Gunturk, M. Lammers, A. Powell, R. Saab, and O. Yilmaz. Sigma delta quantization for compressed sensing. In *44th Annual Conference on Information Sciences and Systems (CISS '10)*, pages 1–6, 2010.

[26] H. Gürkan, U. Guz, and B. S. Yarman. EEG signal compression based on classified signature and envelope vector sets. *International Journal of Circuit Theory and Applications*, 37(2):351–363, 2009.

[27] G. Higgins, B. McGinley, M. Glavin, and E. Jones. Low power compres-

sion of EEG signals using JPEG2000. In *4th International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–4, 2010.

[28] G. Higgins, B. McGinley, N. Walsh, M. Glavin, and E. Jones. Lossy compression of EEG signals using SPIHT. *Electronics Letters*, 47(18):1017–1018, 2011.

[29] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[30] Imec. Imec, Holst Centre and Panasonic present wireless low-power active-electrode EEG headset. `http://www2.imec.be/be_en/press/imec-news/imeceeg2012.html`, 2012. Accessed: 29/08/2013.

[31] Interaxon. Muse: The brainwave sensing headband tech spec sheet. `http://www.interaxon.ca/muse/Muse_Tech_Spec_Sheet_2013.pdf`, 2013. Accessed: 29/08/2013.

[32] J. Jeong. EEG dynamics in patients with Alzheimer's disease. *Clinical Neurophysiology*, 115(7):1490–1505, 2004.

[33] K. G. Jordan. Emergency EEG and continuous EEG monitoring in acute ischemic stroke. *Journal of Clinical Neurophysiology*, 21(5):341–352, 2004.

[34] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon. ICA: A potential tool for BCI systems. *IEEE Signal Processing Magazine*, 25(1):57 –68, 2008.

[35] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski. Independent component analysis of electroencephalographic data. *Advances in Neural Information Processing Systems*, pages 145–151, 1996.

[36] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes. *IEEE Transactions on Biomedical Engineering*, 58(9):2456 –2466, 2011.

[37] D. J. McFarland and J. R. Wolpaw. Brain-computer interfaces for communication and control. *Communications of the ACM*, 54(5):60–66, 2011.

[38] N. Memon, X. Kong, and J. Cinkler. Context-based lossless and near-lossless compression of EEG signals. *IEEE Transactions on Information Technology in Biomedicine*, 3(3):231–238, 1999.

[39] B. Mijovic, V. Matic, M. De Vos, and S. Van Huffel. Independent component analysis as a preprocessing step for data compression of neonatal EEG. In *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '11)*, pages 7316–7319, 2011.

[40] A. Milenković, C. Otto, and E. Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications*, 29(13-14):2521–2533, 2006.

[41] H. R. Mohseni, A. Maghsoudi, and M. B. Shamsollahi. Seizure detection in EEG signals: A comparison of different approaches. In *28th*

Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '06), pages 6724–6727, 2006.

[42] M. Naeem, C. Brunner, R. Leeb, B. Graimann, and G. Pfurtscheller. Separability of four-class motor imagery data using independent components analysis. *Journal of Neural Engineering*, 3(3):208–216, 2006.

[43] Neurosky. What is Mindwave Mobile? `http://www.neurosky.com/products/mindwavemobile.aspx`, 2011. Accessed: 29/08/2013.

[44] Public Health Agency of Canada. Chronic diseases – most significant cause of death globally. `http://www.phac-aspc.gc.ca/media/nr-rp/2011/2011_0919-bg-di-eng.php`. Accessed: 23/08/2013.

[45] World Health Organization. Ageing and life course. `http://www.who.int/ageing/en/`. Accessed: 23/08/2013.

[46] World Health Organization. Global status report on noncommunicable diseases 2010. `http://www.who.int/chp/ncd_global_status_report/en/`. Accessed: 23/08/2013.

[47] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer. EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and Clinical Neurophysiology*, 103(6):642 – 651, 1997.

[48] K.-K. Poh and P. Marziliano. Compressive sampling of EEG signals with finite rate of innovation. *EURASIP Journal on Advances in Signal Processing*, 2010(1):183105, 2010.

[49] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, 8(4):441–446, 2000.

[50] S. Senay, L. F. Chaparro, M. Sun, and R. J. Sclabassi. Compressive sensing and random filtering of EEG signals using Slepian basis. *Proceedings of the EURASIP EUSIPCO*, 8, 2008.

[51] A. H. Shoeb. *Application of Machine Learning to Epileptic Seizure Onset Detection and Treatment*. PhD thesis, Massachusetts Institute of Technology, 2009.

[52] K. Srinivasan, J. Dauwels, and M. R. Reddy. A two-dimensional approach for lossless EEG compression. *Biomedical Signal Processing and Control*, 6(4):387 – 394, 2011.

[53] K. Srinivasan, J. Dauwels, and M. R. Reddy. Multichannel EEG compression: Wavelet-based image and volumetric coding approach. *IEEE Journal of Biomedical and Health Informatics*, 17(1):113–120, 2013.

[54] N. Sriraam. Neural network based near-lossless compression of EEG signals with non uniform quantization. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '07)*, pages 3236–3240, 2007.

[55] N. Sriraam. A high-performance lossless compression scheme for EEG signals using wavelet transform and neural network predictors. *International Journal of Telemedicine and Applications*, 2012:1–8, 2012.

[56] N. Sriraam and C. Eswaran. An adaptive error modeling scheme for the lossless compression of EEG signals. *IEEE Transactions on Information Technology in Biomedicine*, 12(5):587–594, 2008.

[57] M. G. Terzano, L. Parrino, A. Sherieri, R. Chervin, S. Chokroverty, C. Guilleminault, M. Hirshkowitz, M. Mahowald, H. Moldofsky, A. Rosa, et al. Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. *Sleep Medicine*, 2(6):537–553, 2001.

[58] B. L. Titzer, D. K. Lee, and J. Palsberg. AVRORA: Scalable sensor network simulation with precise timing. In *Fourth International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pages 477–482, 2005.

[59] A. Upton and J. Gumpert. Electroencephalography in diagnosis of herpes-simplex encephalitis. *The Lancet*, 295(7648):650 – 652, 1970.

[60] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2009.

[61] Y. Wongsawat, S. Oraintara, T. Tanaka, and K.R. Rao. Lossless multichannel EEG compression. In *IEEE International Symposium on Circuits and Systems (ISCAS '06)*, pages 1614–1617, 2006.

[62] R.F. Yazicioglu, T. Torfs, P. Merken, J. Penders, V. Leonov, V. Puers, B. Gyselinckx, and C. Van Hoof. Ultra-low-power biopotential inter-

faces and their applications in wearable and implantable systems. *Microelectronics Journal*, 40(9):1313 – 1321, 2009.

[63] Z. Zhang, T. Jung, S. Makeig, and B. Rao. Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware. *IEEE Transactions on Biomedical Engineering*, 60(1):221–224, 2013.

[64] Z. Zhang and B. D. Rao. Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation. *IEEE Transactions on Signal Processing*, 61(8):2009–2015, 2013.