University of Alberta

Signal Processing for Sparse Discrete Time Systems

by

Omid Taheri

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

in

Signal and Image Processing

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 2013

Abstract

In recent years compressive sampling (CS) has appeared in the signal processing literature as a legitimate contender for processing of sparse signals. Natural signals such as speech, image and video are compressible. In most signal processing systems dealing with these signals the signal is first sampled and later on compressed. The philosophy of CS however is to sample and compress the signal at the same time. CS is finding applications in a wide variety of areas including medical imaging, seismology, cognitive radio, and channel estimation among others.

Although CS has been given a great deal of attention in the past few years the theory is still naive and its fullest potential is still to be proven. The research in CS covers a wide span from theory of sampling and recovery algorithms to sampling device design to sparse CS-based signal processing applications. The contributions of this thesis are as follows; (i) The analog-to-information converter (AIC) is the device that is designed to collect compressed samples. It is a replacement for the analog-to-digital converter in a traditional signal processing system. We propose a modified structure for the AIC which leads to reducing the complexity of the current design without sacrificing the recovery performance. (ii) Traditional parameter estimation algorithms such as least mean square (LMS) do not assume any structural information about the system. Motivated by the ideas from CS we introduce a number of modified LMS algorithms for the sparse channel estimation problem. Decimated LMS algorithms for the special case of frequency sparse channels are also given. (iii) At last we consider the problem of CS of two dimensional signals. The most straightforward approach is to first find the vector form of a two dimensional signal and then use traditional CS methods to collect the compressed samples. However, our approach samples all the columns of a two dimensional signal with the same measurement matrix. This leads to simplification of the sampling process and also enables us to perform parallel signal recovery.

Table of Contents

1 Introduction **1**
1.1 Proposed research problems 2
1.2 Organization of the thesis 4

2 Preliminaries **5**
2.1 Sampling 5
2.2 Adaptive filtering and system identification 12
2.2.1 Least mean square algorithm 16
2.3 Compressive sampling review 18
2.3.1 Undersampling 18
2.3.2 Compressive sampling basics 19

3 Segmented Analog to Information Conversion in Compressive Sampling **22**
3.1 Setup and background for AIC 23
3.2 Segmented compressive sampling method 27
3.2.1 The basic idea and the model 28
3.2.2 Implementation issues and discussion 33
3.3 RIP for the segmented compressive sampling method 35
3.4 Performance analysis of the recovery 38
3.5 Simulation results 47
3.5.1 Simulation example 1: l_1-norm minimization-based recovery for a time sparse signal 48
3.5.2 Simulation example 2: Time sparse signal with empirical risk minimization-based recovery 50
3.5.3 Simulation example 3: l_1-norm minimization-based recovery for an OFDM signal 52
3.5.4 Simulation example 4: The number of BMIs in the conventional AIC is insufficient for successful recovery 54
3.6 Chapter summary 56

4 Sparse Channel Estimation **57**
4.1 Standard LMS algorithm 58
4.2 Sparsity aware LMS algorithms 62
4.3 Reweighted l_1-norm penalized LMS 63
4.4 Convergence study of the reweighted l_1-norm penalized LMS algorithm 64
4.4.1 Mean convergence 64
4.4.2 Excess MSE 66
4.5 l_p-pseudo-norm penalized LMS algorithm 71
4.6 Penalized LMS algorithms for an arbitrary sparsity basis 72
4.7 Decimated LMS algorithms for frequency sparse channels 73
4.7.1 Decimated LMS 74
4.7.2 Zero attracting decimated LMS 75
4.8 Simulation results 75
4.8.1 Simulation example 1: Time sparse channel estimation 76
4.8.2 Simulation example 2: Arbitrary sparsity basis 77

4.8.3 Simulation example 3: Effect of sparsity level on the performance of the reweighted l_1-norm penalized LMS algorithm . 79
4.8.4 Simulation example 4: Performance comparison of decimated LMS algorithms . . . 81
4.8.5 Simulation example 5: Effect of sparsity level on the decimated LMS algorithm . . . 81
4.9 Chapter summary . . . 82

5 Two Dimensional Compressive Sampling **84**
5.1 Parallel CS of 2D signals . . . 84
5.2 Video compression application . . . 86
5.2.1 CS-based video compression . . . 88
5.2.2 Motion estimation enabled CS-based video compression . . . 92
5.2.3 Simulation results . . . 94
5.3 Chapter summary . . . 98

6 Conclusion **102**
6.1 Summary of contributions . . . 102
6.1.1 AIC design . . . 102
6.1.2 Sparse channel estimation . . . 103
6.1.3 Two dimensional CS . . . 103
6.2 Probable future research . . . 104
6.2.1 Theoretical analysis of segmented AIC for l_1-norm minimization based recovery . . . 104
6.2.2 Quantized segmented AIC . . . 104
6.2.3 Sparse RLS and Kalman filter algorithms . . . 104
6.2.4 Direct two dimensional CS for an arbitrary sparsity basis . . 105

Bibliography **106**

A Proof of Theorem 3.1 **112**

B Proof of Lemma 3.1 **114**

C Proof of Theorem 3.2 **116**

D Proof of Lemma 3.2 **117**

E Proof of Theorem 3.3 **118**

List of Tables

3.1 Percentage that the positions of the nonzero signal values are correctly identified. 55

4.1 Value of α' for different sparsity levels. 71

5.1 MSE values for different scenarios. 86
5.2 PSNR values for different sampling scenarios. 97
5.3 Recovery PSNR comparison of the proposed method with the one in [87]. 99

List of Figures

1.1 Block diagram of the sparse signal processing problem. 2

2.1 The mathematical structure of a C/D converter. 6
2.2 A continuous-time signal $x_c(t)$ being sampled with two different sampling periods. 7
2.3 A band-limited signal. 8
2.4 The spectrum $X_s(j\omega)$ in the case that $\omega_s \geq 2\omega_c$. 8
2.5 The spectrum $X_s(j\omega)$ in the case that $\omega_s < 2\omega_c$. 8
2.6 The frequency and impulse response of the ideal low pass filter. . . . 10
2.7 Block diagram representation of the reconstruction system. 11
2.8 Discrete-time processing of continuous-time signals. 11
2.9 Physical block diagram of A/D conversion. 12
2.10 Sample and hold with $x_c(t)$ as the input. 12
2.11 Basic communication system. 13
2.12 The four classes of adaptive filter applications. 15
2.13 Block diagram of a communication system. 17

3.1 The structure of the AIC based on RMPI. 26
3.2 Sub-sample selection principle for building additional samples in Example 3.2. 33
3.3 Example 3.5: Coefficients C_{1e} and $2C_1$ versus SNR. Since $C_{1e} < 2C_1$ for all values of SNR, one can conclude that the MSE corresponding to the empirical risk minimization-based recovery for the proposed segmented CS-based AIC must be lower than that for the conventional AIC. 45
3.4 Recovery of the time sparse signal based on the l_1-norm minimization algorithm: MSEs versus K_a/K. 49
3.5 Recovery of the time sparse signal based on the empirical risk minimization method: MSEs versus K_a/K. 51
3.6 Recovery of the sparse OFDM signal based on the l_1-norm minimization algorithm: MSEs versus K_a/K. 53

4.1 Simulation example 1, Case 1: MSEs of different estimation algorithms vs number of iterations ($S = 1$). 77
4.2 Simulation example 1, Case 2: MSEs of different estimation algorithms vs number of iterations ($S = 4$). 78
4.3 Simulation example 2: MSEs of estimation algorithms vs number of iterations for a DCT sparse channel ($S = 2$). 79
4.4 Simulation example 3: Excess MSE versus number of iterations. . . . 80
4.5 Simulation example 4: Performance comparsion of different decimated estimation algorithms. 82
4.6 Simulation example 5: Performance of the decimated LMS and the standard LMS for different sparsity levels. 83

5.1 Reconstructed *Akiyo* frames. 98
5.2 Reconstructed *Coastguard* frames. 100

5.3 Reconstructed *Coastguard* frames from the motion estimation enabled method. 100
5.4 PSNR vs compression ratio. 101
5.5 Reconstructed *Stefan* frames from the motion estimation enabled method. 101

List of Symbols

Symbol	Description	First use
$x_c(t)$	Continuous-time signal being sampled	5
$x[n]$	Discrete-time signal of samples	5
f_s	Sampling frequency	6
$X(jw)$	Fourier transform of $x(t)$	7
$\mathcal{F}\{x(t)\}$	Fourier transform of $x(t)$	7
$\boldsymbol{w}$	The channel impulse response	17
$\boldsymbol{w}_k$	The estimate of channel impulse response at time-step k	17
$\boldsymbol{x}_k$	The channel's input data vector at time-step k	17
n_k	Additive noise component at time-step k	17
d_k	Desired response at time-step k	17
e_k	Error signal at time-step k	17
$(\cdot)^T$	Transposition operator	17
σ_n^2	Variance of the noise component n_k	17
μ	The LMS algorithm's step size	17
$\boldsymbol{R}$	Covariance matrix of $\boldsymbol{x}_k$	17
$\lambda_{\max}$	Maximum eigenvalue of $\boldsymbol{R}$	17
$E\{\cdot\}$	Mathematical expectation	17
$\boldsymbol{f}$	A sparse signal	18

N	Dimension of sparse signal $\boldsymbol{f}$, also length of channel impulse response $\boldsymbol{w}$	18
S	Sparsity level	18
$\boldsymbol{\Psi}$	Sparsity basis	18
$\boldsymbol{y}$	Vector of compressed samples	18
$\boldsymbol{\Phi}$	CS measurement matrix	18
K	Number of rows of measurement matrix $\boldsymbol{\Phi}$	18
$\boldsymbol{x}$	Representation of $\boldsymbol{f}$ in sparsity basis $\boldsymbol{\Psi}$	19
$\|\cdot\|_{l_1}$	l_1-norm of a vector	19
$\|\cdot\|_{l_2}$	Euclidean norm of a vector	20
$\boldsymbol{n}$	The noise vector of the sampling process	20
M	The number of segments of the segmented AIC	22
$[\cdot]_{k,n}$	The (k,n)th entry of a matrix	24
$\boldsymbol{Y}$	The sub-sample matrix	29
$\mathcal{P}^{(i)}$	Set of M permutations applied to the columns of $\boldsymbol{Y}$	29
$\boldsymbol{Y}^{\mathcal{P}^{(i)}}$	The permuted version of $\boldsymbol{Y}$	30
K_a	The number of additional samples	32
K_e	The extended number of samples	33
$\boldsymbol{\Phi}_e$	The extended measurement matrix	33
$\boldsymbol{v}_k$	Coefficient error vector at time-step k	59
$[\cdot]_i$	The ith entry of a vector	59
ξ_k	Excess mean square error at time-step k	60
ξ	Excess mean square error	61
$\mathrm{sgn}(\cdot)$	Component-wise sign function of a vector	62

$\lvert \cdot \rvert$	Component-wise absolute value of a vector	63
$[\cdot]_{r,:}$	The rth row of a matrix	84
$[\cdot]_{:,c}$	The cth column of a matrix	84

List of Abbreviations

Abbreviation	Description	First use
2D	Two dimensional	3
A/D	Analog-to-digital	2
AIC	Analog-to-information converter	2
AWGN	Additive white Gaussian noise	85
BPSK	Binary phase shift keying	73
BMI	Branch of mixers and integrators	3
CS	Compressive sampling	1
C/D	Continuous-to-discrete-time	5
CIR	Channel impulse response	17
D/A	Digital-to-analog	11
DCT	Discrete cosine transform	77
DSP	Digital signal processing	5
DWT	Discrete wavelet transform	87
EEG	Electroencephalogram	16
FFT	Fast Fourier transform	5
FIR	Finite impulse resonse	16
i.i.d.	Independent and identically distributed	17
ISI	Intersymbol interference	13

LASSO	Least absolute shrinkage and selection operator	26
LMS	Least mean square	3
LPC	Linear predictive coding	16
MAD	Mean absolute difference	93
MRI	magnetic resonance imaging	1
MSE	Mean square error	13
OFDM	Orthogonal frequency-division-multiplexing	52
PAM	Pulse amplitude modulation	15
PSNR	Peak signal to noise ratio	90
QPSK	Quadrature phase-shift keying	52
RZA – LMS	Reweighted zero attracting LMS	62
RIP	Restricted isometry property	20
RLS	Recursive least squares	3
RMPI	Random modulation preintegration	25
SNR	Signal-to-noise ratio	44
ZA – LMS	Zero attracting LMS	62
ZAD – LMS	Zero attracting decimated LMS	75

Chapter 1

Introduction

The sampling device is an indispensable part of a digital signal processor which is interacting with analog inputs. Sampling is costly, consumes energy and time and can be the bottleneck in a variety of practical applications. For example, in wireless sensor networks, sampling consumes a portion of energy from a finite energy source, which can lead to a significant decrease of the network lifetime. Another example is magnetic resonance imaging (MRI), where the sampling process is very lengthy and, thus, influences the patient's comfort and the reliability of the results, especially if many samples have to be collected.

Shannon sampling theorem states that the analog signal can be recovered by an appropriate low pass filter, only if it is sampled at a rate greater than or equal twice the highest frequency in the signal. Therefore, often a very large number of samples have to be collected to represent an analog signal in digital format if the signal has a rather wide bandwidth. Most of natural signals are, however, compressible and they can be represented with a lower information rate than the rate suggested by the Shannon sampling theorem. Therefore, robust and reliable methods which can reduce the number of samples for signal representation leading to saving energy, time and cost are extremely important. Recently the new theory of compressive sampling (CS) has emerged which defies the common wisdom in signal processing on how to sample an analog signal [1]. It provides us with means of sampling sparse signals with a much lower rate than what Shannon's theorem requires. Despite its great importance and already widely recognized potential, the theory of CS is on early stages of its development. The first papers in this field were published in late 2005. The CS theory is still naive in many aspects but is becoming a major branch in the signal processing world.

1.1 Proposed research problems

CS is a promising solution to the under-sampled signal recovery problem. This thesis builds on the ideas of CS and considers the problem of sparse signal processing for discrete-time systems. The problem setup is that a signal is first to be compressively sampled, transmitted through a communication channel next, and then recovered at the receiver side. This setup is shown as a block diagram in Fig. 1.1. We aim at exploiting the sparsity of the signal in order to either simplify the sampling and signal recovery procedure or improve the recovery performance compared to existing CS signal acquisition methods. After the collection of the compressed samples, these samples are transmitted through a channel. At the receiver side, the channel has to be estimated in order to find an estimate of the data that was transmitted over the channel. Our focus in this stage is to outperform standard channel estimation algorithms when the communication channel is sparse. For the purposes of signal

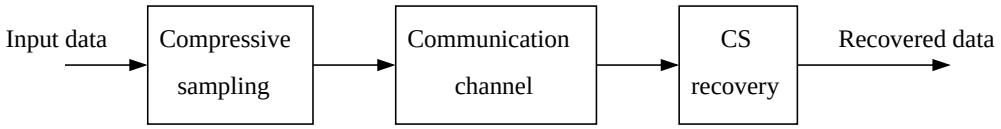


Figure 1.1: Block diagram of the sparse signal processing problem.

acquisition we propose an extension of the analog to information converter (AIC) design which leads to better recovery performance. The new AIC design, however, collects the compressed samples at a slightly higher rate than the traditional AIC. We also consider the problem of two dimensional CS and propose the parallel CS method which has the advantage of less complex sampling and parallel signal recovery. However, a larger number of samples are required for the parallel CS method compared to the traditional CS in order to achieve a certain recovery performance. For the channel estimation problem we propose several channel estimation algorithms with the goal of improving the channel estimate over the result of standard algorithms. A better channel estimate leads to better signal recovery. The research problems tackled during this PhD work are presented in the following with more detail.

Analog to information converter design: In CS we are interested in directly collecting the compressed information about sparse analog signals, rather than sampling the signals with a high speed analog-to-digital (A/D) converter and then com-

pressing the digital samples. This goal can be achieved through the AIC structure which includes a series of parallel branches of mixers and integrators (BMIs). We propose an extension to the AIC which allows reusing the measurements from each BMI in order to improve the recovered signal quality [2]. The new AIC structure achieves almost the same performance with only half the number of BMIs required by the current AIC designs. Therefore, our design leads to significant complexity reduction of the sampling device.

The AIC design introduced in [2] enables us to retrieve additional information from the already collected measurements and reduce the complexity of the sampling device. Since sampling devices are present in almost every digital signal processors, such as MRIs, digital communication systems, digital storage systems, etc., significant reduction of the sampling device complexity has a profound effect on simplifying the existing systems.

Sparse channel estimation: An estimate of the communication channel is required at the receiver side in order to correctly recover the transmitted data. Some of the classic methods often used for channel estimation include the least mean square (LMS), the recursive least squares (RLS), and the Kalman filter algorithms. These algorithms can be adjusted so that they perform better than their standard versions when estimating a sparse communication channel. We propose several modifications of the standard LMS algorithm for the purpose of sparse channel estimation.

Simpler LMS type channel estimation algorithms help reduce the complexity and consequently the cost of communication devices such as cell phones. These methods can also be deployed in sensor networks leading to an increase in the lifetime of the network by preserving the batteries. Sensor networks are used in applications when there is almost no infrastructure in place, such as wildlife tracking. CS-based sensor networks can help reduce the cost and size of the transmitter nodes in the network.

Two dimensional CS: The most common way in the CS literature to take the compressed samples of a two dimensional (2D) signal is to vectorize the signal. Once the signal is vectorized the compressed samples can be collected through the use of a measurement matrix. Depending on the size of the original signal, the measurement matrix may become large which leads to the use of a large number of BMIs in the corresponding AIC. In this dissertation we present another approach for CS of 2D signals. Video compression is looked at as a sample application of two dimensional CS.

1.2 Organization of the thesis

Preliminary and background material is presented in Chapter 2. Our extended AIC design, named segmented AIC, is introduced in Chapter 3. The proposed modifications of the standard LMS for sparse channel estimation are given in Chapter 4. The two dimensional CS algorithm and its application for video compression is introduced in Chapter 5. Chapter 6 concludes the thesis and gives ideas on future research that can be done in the areas pursued in this PhD work.

Chapter 2

Preliminaries

In this introductory chapter we first review the basics of sampling of analog signals. The principles of adaptive filtering and system identification are also presented here. A brief overview of CS concludes the chapter.

2.1 Sampling

The theory of signal processing has a wide range of applications in communications, entertainment, medicine, etc. The signal processing technology was mostly analog before the 1960s [3]. However, digital computers and microprocessors along with developments such as fast Fourier transform (FFT) caused a shift to digital technologies and digital signal processing (DSP). A great deal of the signals encountered in practice are continuous-time or analog. In order to process these signals in the discrete-time domain we first have to acquire a discrete-time representation of the continuous-time signal. Once a discrete-time representation of the analog signal is obtained a wide variety of DSP algorithms are at disposal in order to either change the signal in a desirable way or extract useful information out of it. Finding a discrete-time representation of the continuous-time signal is usually done through a procedure called sampling or continuous-to-discrete-time (C/D) conversion. It is obvious that the continuous-time signal should be recoverable from its discrete-time samples in order for the sampling scheme to be valid and applicable in practice.

The most common way of acquiring a discrete-time sequence from a continuous-time signal is through periodic sampling. Let $x_c(t)$ be the continuous-time signal being sampled. The sample sequence, $x[n]$, is obtained from $x_c(t)$ as

$$x[n] = x_c(nT), \qquad -\infty < n < \infty \tag{2.1}$$

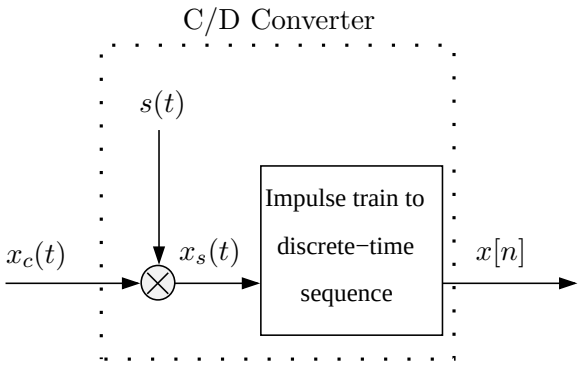


Figure 2.1: The mathematical structure of a C/D converter.

where T is the sampling period. The sampling frequency is $f_s = 1/T$ which denotes the number of samples collected per second. A mathematical representation of the sampling process using an impulse train modulator is shown in Fig. 2.1.

In Fig. 2.1, $s(t)$ is the following periodic impulse train which is being modulated by $x_c(t)$.

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT). \tag{2.2}$$

The modulated impulse train $x_s(t)$ can be expressed as

$$x_s(t) = x_c(t)s(t) = x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x_c(nT)\delta(t - nT). \tag{2.3}$$

The discrete-time signal $x[n]$ is constructed from the modulated impulse train $x_s(t)$. Fig. 2.2 shows a continuous-time signal $x_c(t)$ being sampled with two different sampling periods. The two diagrams to the left of Fig. 2.2 show the signals $x_s(t)$ and $x[n]$ for the case when the sampling period is $T = T_0$. The diagrams to the right of Fig. 2.2 correspond to the case when the sampling period has been doubled. It can be seen in Fig. 2.2 that the resulting discrete-time signals $x[n]$ are completely different for the two sampling periods considered. Increasing the sampling period and thus collecting samples at a lower rate is desirable in practice as it leads to a less complex C/D converter and a smaller number of samples collected per time unit. However, we may not be able to increase the sampling period endlessly since we should be able to recover the continuous-time signal, $x_c(t)$, from its discrete-time counterpart $x[n]$. In order to find the minimum sampling frequency while still being able to recover $x_c(t)$ from $x[n]$ the frequency domain representation of these signals through Fourier transform is typically considered.

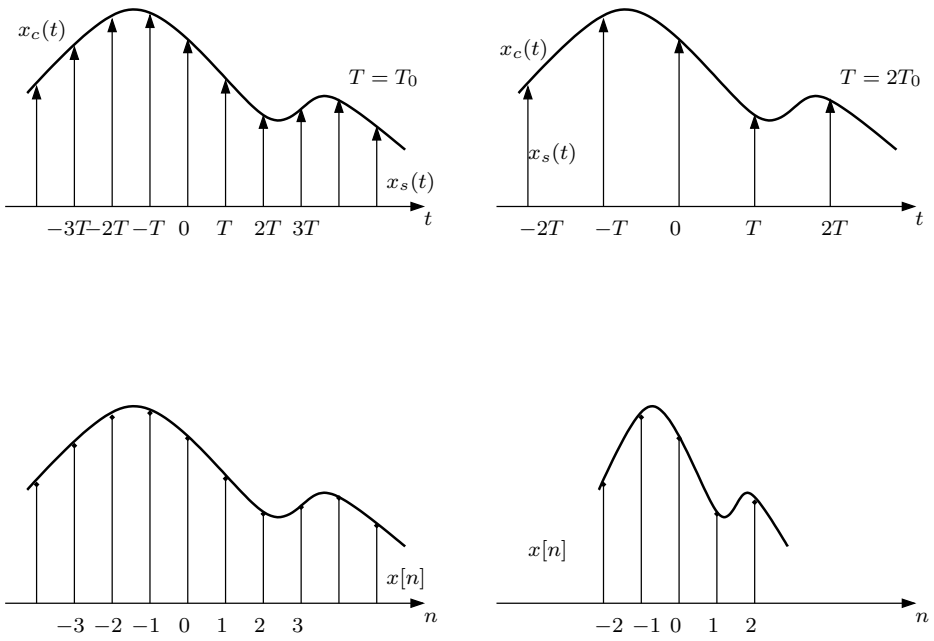


Figure 2.2: A continuous-time signal $x_c(t)$ being sampled with two different sampling periods.

Let the Fourier transform of a signal $x(t)$ denoted by $X(j\omega)$ be defined as

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt. \tag{2.4}$$

An alternative notation that we use to denote the operation of taking the Fourier transform of $x(t)$ is $\mathcal{F}\{x(t)\}$. From the above definition $S(j\omega)$, the Fourier transform of $s(t)$ in equation (2.2), can be derived as

$$S(j\omega) = \frac{2\pi}{T}\sum_{k=-\infty}^{\infty}\delta(\omega - k\omega_s) \tag{2.5}$$

where ω_s is the sampling frequency in radians per second and is equal to $2\pi f_s$ with $f_s = 1/T$. From the convolution property of the Fourier transform we have

$$X_s(j\omega) = \mathcal{F}\{x_s(t)\} = \frac{1}{2\pi}X_c(j\omega) \star S(j\omega). \tag{2.6}$$

Therefore, $X_s(j\omega)$ can be derived as

$$X_s(j\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(\omega - k\omega_s). \tag{2.7}$$

Equation (2.7) describes the relation between the spectrum of $x_s(t)$ and the original signal $x_c(t)$. $X_s(j\omega)$ is the collection of periodically repeated replicas of $X_c(j\omega)$. Let us consider the specific example of a band-limited signal $x_c(t)$ whose frequency

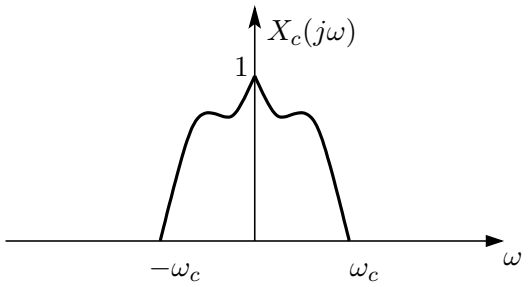


Figure 2.3: A band-limited signal.

components stretch out from $-\omega_c$ to ω_c as shown in Fig. 2.3. In the case that the sampling frequency is greater than twice the maximum frequency of $x_c(t)$, $\omega_s \geq 2\omega_c$, the spectrum of $x_s(t)$ will look like what is shown in Fig. 2.4. Since $\omega_s \geq 2\omega_c$,

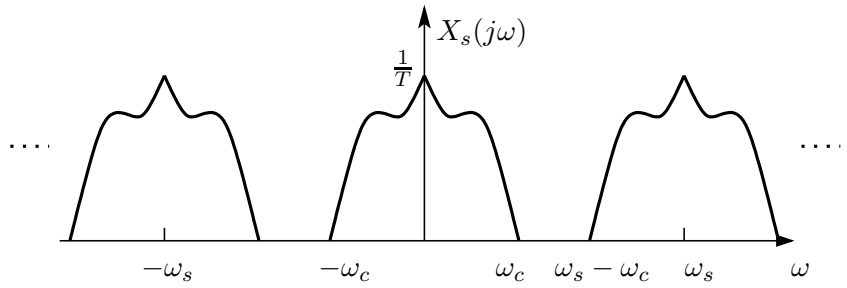


Figure 2.4: The spectrum $X_s(j\omega)$ in the case that $\omega_s \geq 2\omega_c$.

$\omega_s - \omega_c \geq \omega_c$ and therefore, the two neighboring replicas of $X_s(j\omega)$ corresponding to $k = 0$ and $k = 1$ in equation (2.7) do not interfere with each other. The same is true for any other two consecutive k values. However, in the case that $\omega_s < 2\omega_c$ the resulting $X_s(j\omega)$ will be as shown in Fig. 2.5. As it can be seen in Fig. 2.5 the

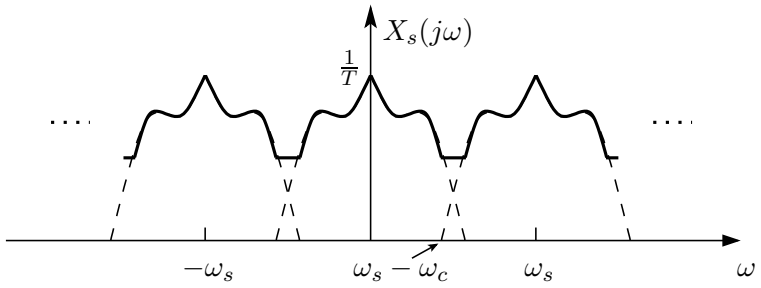


Figure 2.5: The spectrum $X_s(j\omega)$ in the case that $\omega_s < 2\omega_c$.

two neighboring copies of $X_c(j\omega)$ interfere with each other and it can be concluded that in this case the original spectrum $X_c(j\omega)$ can not be recovered from $X_s(j\omega)$. This type of distortion of the original continuous-time signal $x_c(t)$ evident in the

spectrum $X_s(j\omega)$ due to sampling with a low rate is called *aliasing*. The Shannon sampling theorem based on the above discussion is stated as follows [3].

Theorem 2.1. *Let $x_c(t)$ be a band-limited signal with*

$$X_c(j\omega) = 0 \quad for \quad |\omega| \geq \omega_c. \tag{2.8}$$

Then $x_c(t)$ is uniquely determined by its samples $x[n] = x_c(nT), n = 0, \pm 1, \pm 2, \cdots$, if

$$\omega_s = \frac{2\pi}{T} \geq 2\omega_c. \tag{2.9}$$

The maximum frequency of the signal $x_c(t)$, that is ω_c, is called the *Nyquist frequency* and the minimum frequency ω_s with which the signal has to be sampled to guarantee a successful recovery is referred to as the *Nyquist rate*.

Assuming that the signal is sampled with the *Nyquist rate* the spectrum $X_s(j\omega)$ looks like what is shown in Fig. 2.4 which contains periodic copies of the spectrum of the original continuous-time signal $x_c(t)$. Therefore, it is possible to recover the band-limited signal $x_c(t)$ from its discrete-time samples $x[n]$. The first step in the process is to generate a modulated impulse train $x_s(t)$ from $x[n]$ with the knowledge of the sampling period T as

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT). \tag{2.10}$$

Here, $x_s(t)$ would have a spectrum like the one in Fig. 2.4. The second step of the reconstruction process is to pass $x_s(t)$ given by equation (2.10) through an ideal low pass filter with impulse response $h_r(t)$ and frequency response $H_r(j\omega)$ shown in Fig. 2.6. The resulting signal $x_r(t)$ will have the same spectrum as the original signal $x_c(t)$ and can be written as

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t - nT). \tag{2.11}$$

The impulse response of the ideal low pass filter $h_r(t)$ can be derived as

$$h_r(t) = \frac{\sin(\pi t/T)}{\pi t/T}. \tag{2.12}$$

Therefore, $x_r(t)$ in (2.11) can be rewritten as

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]\frac{\sin(\pi(t - nT)/T)}{\pi(t - nT)/T}. \tag{2.13}$$

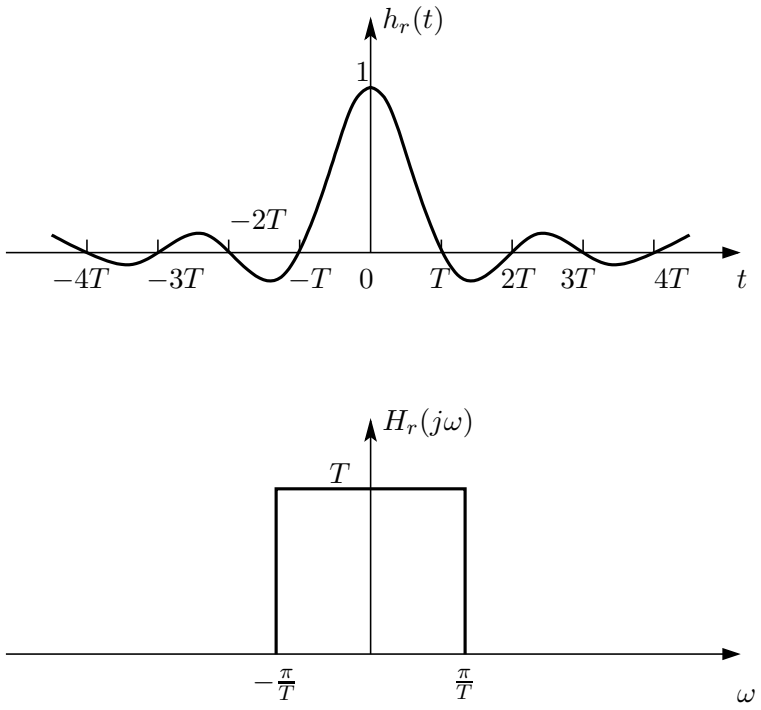


Figure 2.6: The frequency and impulse response of the ideal low pass filter.

A block diagram of the above described reconstruction is shown in Fig. 2.7. The reconstruction system in this figure is also called an ideal discrete-time-to-continuous-time (D/C) converter. For the purposes of discrete-time processing of continuous-time signals one can use the setup shown in Fig. 2.8. A discrete-time representation $x[n]$ of the continuous-time signal $x_c(t)$ is first acquired through sampling. The resulting discrete-time signal is processed with some DSP technique. At last the processed discrete-time signal $y[n]$ passes through a D/C converter to form a continuous-time signal $y_r(t)$.

The ideal C/D converter in Fig. 2.1 converts a continuous-time signal to the corresponding discrete-time signal. Every sample of the discrete-time signal $x[n]$ in equation (2.1) is of infinite precision and it may not be possible to store these samples in a finite length memory unit in a digital signal processing system. A general approach used in many real world applications is to quantize the infinite precision discrete-time samples and then use these quantized samples as the input of an A/D converter. An A/D converter is a physical device with a voltage level as its input which produces at the output the binary code representing the quantized version of the input. The conversion of the analog input voltage to its digital representation

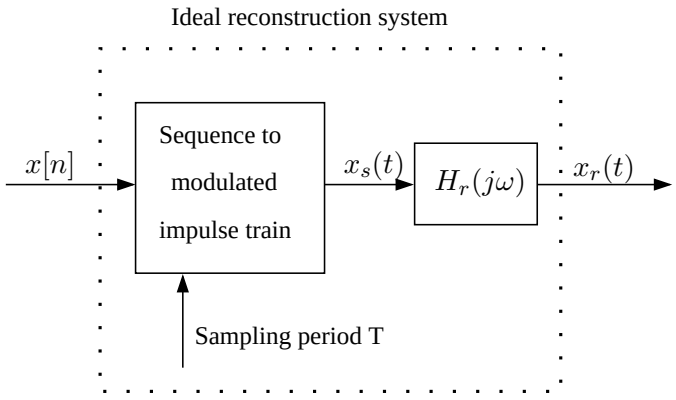


Figure 2.7: Block diagram representation of the reconstruction system.

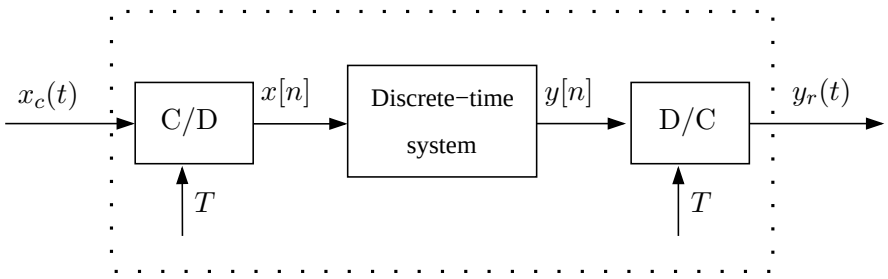


Figure 2.8: Discrete-time processing of continuous-time signals.

is not instantaneous. Therefore, a sample and hold device is usually included in the circuitry of an A/D converter. A configuration for A/D conversion of a continuous-time signal $x_c(t)$ is shown in Fig. 2.9. In this figure, $x_B[n]$ denotes the binary representation of the quantized samples taken from the continuous-time signal $x_c(t)$. The input-output relation of the sample and hold block in Fig. 2.9 is

$$x_0(t) = \sum_{n=-\infty}^{\infty} x_c(nT)h_0(t - nT). \tag{2.14}$$

$h_0(t)$ in (2.14) is defined as

$$h_0(t) = \begin{cases} 1, & 0 < t < T \\ 0, & \text{otherwise.} \end{cases} \tag{2.15}$$

Fig. 2.10 shows a hypothetical input to the sample and hold block and its corresponding output, the quantized samples taken at the end of every sampling period, and the output of a digital-to-analog (D/A) converter. The D/A converter takes the binary representation of a sample as its input and produces a constant voltage level at the output. Entries of the digital sequence $x_B[n]$ in Fig. 2.9 are of finite precision and have a finite length binary representation. The quantized samples are

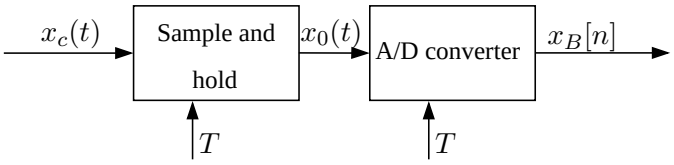


Figure 2.9: Physical block diagram of A/D conversion.

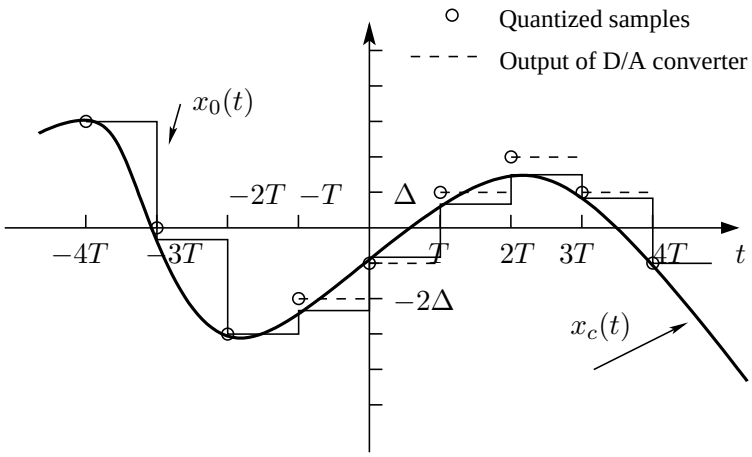


Figure 2.10: Sample and hold with $x_c(t)$ as the input.

not always equal to the unquantized samples and therefore, the quantization process introduces some error into the digital samples. Hence, the original unquantized samples can not be exactly recovered from the quantized samples at the output of the D/A converter as it can be seen in the specific example of Fig. 2.10.

2.2 Adaptive filtering and system identification

One of the topics considered in this dissertation is the problem of channel estimation for sparse communication channels. Channel estimation can be thought of as a sub category of the system identification problem. Estimation theory has a wide range of applications such as communications, radar, navigation, biomedical engineering, etc. [4–6]. Consider the basic digital communication system shown in Fig. 2.11. This system consists of a transmitter, channel, and a receiver. If we assume that there is no additive noise in the communication channel and the channel has an ideal impulse response given by

$$h[n] = A\delta[n - n_0], \tag{2.16}$$

Figure 2.11: Basic communication system.

then the received signal is a shifted and scaled version of the transmitted signal and the information sent over the channel can be easily recovered. In reality, however, the channel impulse response is more complicated than what is given in (2.16) which results in intersymbol interference (ISI). Also, there is an element of noise present in the received signal usually modeled as an additive component which is unrelated to the transmitted signal. These two sources of impairment, ISI and additive noise, forces the receiver to find an *estimate* of the original information sent by the transmitter. Due to the presence of non-deterministic noise the estimation theory is *statistical*. The three different types of estimation include filtering, smoothing, and prediction. In filtering, the measured data up to time t is used to extract information about the parameter of interest at time t. Smoothing exploits the data up to time t to estimate the parameter of interest at a past time $t' < t$. Finally, prediction uses the data up to time t to forecast the parameter of interest at a future time $t' > t$.

Filters in general can be categorized as *linear* or *nonlinear*. A linear filter's output is a linear combination of its inputs. The statistical approach to the design of a linear filter usually assumes the existence of some statistical information such as mean and correlation functions of the input signal as well as the additive noise. One approach is to minimize the mean square error (MSE) between the desired output and the filter output. For stationary inputs the solution is the *Wiener filter* [7]. If the signal is non-stationary, the Wiener filter in not optimum and a more successful algorithm is the *Kalman filter* [8]. If the statistical information about the system is not available, two different approaches can be taken. The *non-recursive* approach first observes the system for a period of time in order to estimate the necessary statistical parameters. Then it uses the derived statistical information to find the Wiener filter parameters. Another approach is to use *adaptive filters*. An adaptive filter adjusts itself through the use of a *recursive algorithm*. In a stationary setup, the adaptive filter converges to the Wiener solution in mathematical expectation after a number of iterations. The use of an adaptive filter in a non-stationary environment enables the filter to *track* the system. The collection of recursive algo-

rithms developed in the literature can be compared based on the merits of *rate of convergence*, *robustness to noise*, and *computational complexity*.

Since the parameters of adaptive filters may change over the course of the filtering process, adaptive filters are nonlinear. However, adaptive filters are deemed linear if the input-output relation is linear whenever the parameters are fixed. There are two different approaches to derive recursive algorithms for linear adaptive filters: stochastic gradient approach, and least-squares estimation.

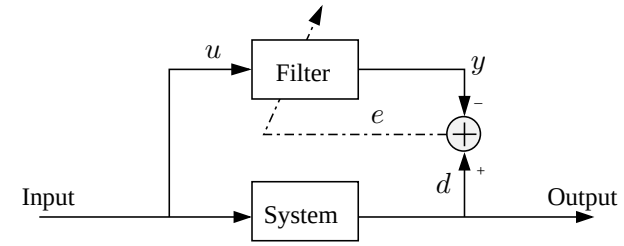
A transversal filter is used as the underlying structure of a linear adaptive filter in the stochastic gradient approach. A cost function defined as the mean square of the difference between the desired response and the filter's output is minimized. An iterative procedure based on the *method of steepest descent* is used to solve the *Wiener-Hopf equations*. The instantaneous values of the correlation matrix of the input and the cross-correlation between the desired response and the input is used to find an estimate of a *gradient* vector. This widely used algorithm is known as the *least mean square* method which is revisited in Section 2.2.1.

The other approach of designing recursive algorithms for adaptive filters is through the *method of least squares*. The cost function being minimized in this method is defined as the sum of weighted error squares. The error is again defined as the difference between a desired response and the filter's actual output.

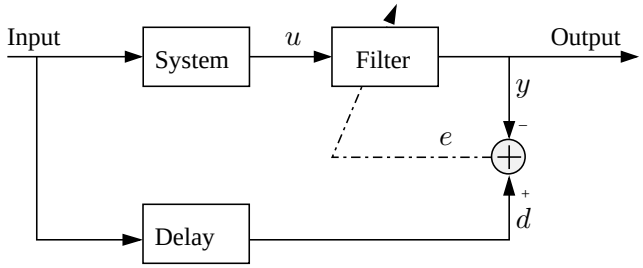
Adaptive filters have applications in a variety of different fields. All the different applications of adaptive filters use a desired response and an input vector to form an estimation error which is used to adjust the filter coefficients. Based on the nature of the desired response the application of adaptive filters can be classified into four categories: *identification*, *inverse modeling*, *prediction*, and *interference cancellation*. These applications are shown in Fig. 2.12 in which u, y, d, and e denote the input of the adaptive filter, its output, the desired response, and the error signal respectively.

Identification: An identification application shown in Fig. 2.12(a) uses an adaptive filter algorithm to find a mathematical model for the system. The input to the system and the filter is the same and the output of the system is used as the desired response. System identification is abundantly used in control theory [9, 10]. The problem of channel estimation in communication systems is another well known application [11, 12].

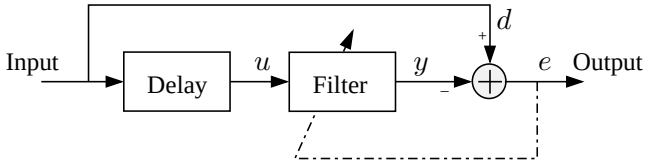
Inverse modeling: In the inverse modeling application shown in Fig. 2.12(b),



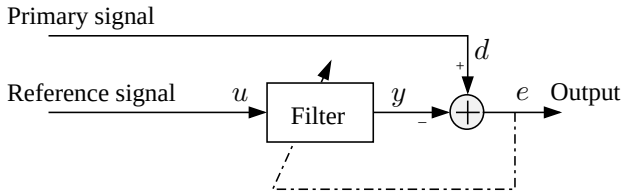
(a) Identification.



(b) Inverse modeling.



(c) Prediction.



(d) Interference cancellation.

Figure 2.12: The four classes of adaptive filter applications.

the adaptive filter provides the inverse of the system model. The best case scenario for a linear system is that the combination of system and adaptive filter makes for an ideal transmission medium. Adaptive equalization is widely used in communication systems and it is a good example of inverse modeling [13]. The purpose of an equalizer is to correct the signal's distortion due to ISI. Zero-forcing equalizers were among the early structures developed for adaptive equalization of pulse amplitude modulation (PAM) systems in 1960s [14, 15]. Research on nonlinear receiver structures under various optimality criteria followed [16]. The result was the maximum likelihood sequence estimator [17] which uses the Viterbi algorithm [18]. Adaptive versions of the maximum likelihood estimator were also developed [19]. Decision

feedback equalizers were also introduced as a simple suboptimal structure [20].

Prediction: In prediction applications as shown in Fig. 2.12(c), the adaptive filter predicts the current value of a random signal. The current value of the random signal is used as the desired response while its past values are used as the filter inputs. Adaptive filtering is utilized to develop a model of the signal of interest. Among the prediction applications of adaptive filters is, for example, predictive coding of speech [5,21]. In most applications such as linear predictive coding of speech (LPC), the error signal is usually stored or transferred since it has smaller variance than the original signal [6]. Another application of prediction is in processing and interpreting the electroencephalograms (EEGs) of the brain [22]. Adaptive filtering is used to detect whether certain resonances called rhythms are present in the EEG [23]. Many other predictive applications are developed as well.

Interference cancellation: The adaptive filter is used to remove an interference signal present in the primary signal in interference cancellation applications shown in Fig. 2.12(d) [24]. Here, a reference signal which is a highly disturbed version of the primary signal and mostly represents the interference is used as the adaptive filter input, and the primary signal is used as the desired response. Among the applications are echo cancellation in long distance transmission lines [25], adaptive line enhancement [26, 27], and adaptive beamforming [28–30].

2.2.1 Least mean square algorithm

The LMS algorithm is very well known in the field of adaptive signal processing [6, 31, 32]. As mentioned earlier, LMS belongs to the class of stochastic gradient algorithms. The LMS algorithm does not need specific stochastic knowledge of the channel and input data sequence unlike some other known parameter estimation methods such as RLS and the Kalman filter. LMS is being employed in a wide variety of applications in signal processing and communications including system identification, echo cancellation, channel estimation, adaptive line enhancement, etc. The application considered in this work is that of estimating a finite impulse response (FIR) channel. The choice of the channel estimation algorithm for use in a communication system comes down to the available information about the statistics of the system, the desired performance of the estimation algorithm, as well as the complexity of the estimation process.

The system model of the communication system used in this section and later

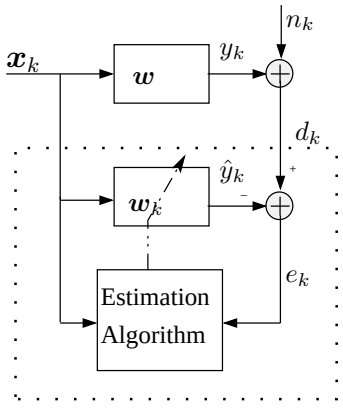


Figure 2.13: Block diagram of a communication system.

on in Chapter 4 is shown in Fig. 2.13. The LMS algorithm is used to estimate the channel impulse response (CIR) of the communication system. Let us first introduce the notations in this figure. The vector $\boldsymbol{w}$ is the actual CIR which is to be estimated, $\boldsymbol{w}_k$ is the estimated coefficient vector at time-step k, $\boldsymbol{x}_k$ is the system's input data vector, n_k is the additive noise, d_k is the desired response, and e_k is the error signal. The CIR is assumed to be of length N, and therefore, $\boldsymbol{w} = (w_1, w_2, \cdots, w_N)^T$, $\boldsymbol{w}_k = (w_{1,k}, w_{2,k}, \cdots, w_{N,k})^T$, and $\boldsymbol{x}_k = (x_k, x_{k-1}, \cdots, x_{k-N+1})^T$, where $(\cdot)^T$ stands for vector transposition. From Fig. 2.13 it can be seen that

$$\begin{aligned} d_k &= \boldsymbol{w}^T \boldsymbol{x}_k + n_k \\ e_k &= d_k - \boldsymbol{w}_k^T \boldsymbol{x}_k. \end{aligned} \tag{2.17}$$

The noise samples, n_k, are assumed to be independent and identically distributed (i.i.d.) with zero mean and variance of σ_n^2. Also, the input data sequence, $\boldsymbol{x}_k$, and the additive noise samples, n_k, are assumed to be uncorrelated.

In standard LMS, a cost function of the form $L_k = (1/2)e_k^2$ is minimized using the gradient descent algorithm [31]. The update equation of the standard LMS algorithm can be derived from the above mentioned cost function as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \mu \frac{\partial L_k}{\partial \boldsymbol{w}_k} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k \tag{2.18}$$

where μ is the step size of the iterative algorithm. In order to make sure that the LMS algorithm converges, μ is chosen such that $0 < \mu < \lambda_{\max}^{-1}$ with $\lambda_{\max}$ being the maximum eigenvalue of the covariance matrix of $\boldsymbol{x}_k$ which is $\boldsymbol{R} = E\left\{\boldsymbol{x}_k \boldsymbol{x}_k^T\right\}$ where $E\{\cdot\}$ denotes the mathematical expectation.

2.3 Compressive sampling review

According to Shannon's sampling theorem, an analog band-limited signal can be recovered from its discrete-time samples if the sampling rate is at least twice the bandwidth of the signal. However, most of the practical signals (such as natural images or human speech) are compressible. This suggests that the information rate of these signals is less than what suggested by their bandwidth. Recent theory of CS, aims at finding a way to recover these signals from a smaller number of samples [1, 33–35]. The discrete CS problem is to recover a signal $\boldsymbol{f} \in \mathbb{R}^N$ from a small (less than N) number of measurements. CS relies on two important features, one is *sparsity* which pertains to the signal being sampled and the other one, *incoherence*, relates to the sampling waveforms. A signal $\boldsymbol{f} \in \mathbb{R}^N$ is called sparse if it has a small number of nonzero coefficients when described in a proper basis $\boldsymbol{\Psi}$. As a result, an S-sparse signal in the $\boldsymbol{\Psi}$ domain is a signal with at most S nonzero coefficients in that domain. Incoherence feature of the sampling waveforms states that unlike the signal of interest the waveforms must have a dense representation in the $\boldsymbol{\Psi}$ domain. An interesting fact about CS is that one can design a universal sampling matrix (for example, a random projection matrix) that works for all S-sparse signals regardless of the nature of the signal. Therefore, CS is capable of sampling sparse signals in a universal manner. CS has already found a wide range of applications such as image acquisition [36], sensor networks [37], cognitive radios [38], communication channel estimation [39, 40], etc.

2.3.1 Undersampling

The CS reconstruction problem is that of recovering a sparse signal $\boldsymbol{f} \in \mathbb{R}^N$ from a set of linear measurements $\boldsymbol{y} = (y_1, y_2, \ldots, y_K)^T$ which are computed by correlating the signal of interest with a set of sampling or measurement waveforms. These measurements are of the form

$$y_k =< \boldsymbol{f}, \boldsymbol{\phi}_k >, \quad k = 1, ..., K \tag{2.19}$$

where $< \cdot, \cdot >$ denotes the inner product of two vectors. The measurement process can also be described by the matrix equation $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{f}$ where $\boldsymbol{\Phi}$ is the matrix built by stacking the measurement vectors $\boldsymbol{\phi}_k$ into its rows. In CS we are interested in the *underdetermined* case where the number of measurements is less than the dimension of the original signal, i.e., $K \ll N$. In general, this is a system of equations with

more unknowns than measurements which does not have a unique solution. However, knowing that the signal of interest is sparse, one can exactly recover the signal by linear programming provided that enough measurements are taken from the signal.

2.3.2 Compressive sampling basics

The basic mathematics supporting the CS theory is discussed in this section. The topics include the characterization of a sparse signal, design of measurement matrix, and mathematical results ensuring signal recovery from the compressed measurements.

Two main premises

There are two main facts about the signal of interest and the sampling mechanism that makes CS possible. These are *sparsity* and *incoherence*. Consider the following representation of a signal $\boldsymbol{f} \in \mathbb{R}^N$ in an orthonormal basis $\boldsymbol{\Psi}$ as

$$\boldsymbol{f} = \sum_{i=1}^{N} x_i \boldsymbol{\psi}_i \tag{2.20}$$

where $\boldsymbol{\psi}_i,\ i = 1, \ldots, N$ are the basis vectors. The signal $\boldsymbol{f}$ is sparse if the majority of its coefficients x_i in the sparsity basis $\boldsymbol{\Psi}$ are zero. Let $\boldsymbol{\Psi}$ be the $N \times N$ orthonormal matrix with $\boldsymbol{\psi}_i$ as its rows. Let the vector of signal coefficients be denoted as $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T$. Then equation (2.20) can be written in the matrix form as

$$\boldsymbol{f} = \boldsymbol{\Psi}^T \boldsymbol{x}. \tag{2.21}$$

Collection of compressed samples is done by multiplying a $K \times N$ measurement matrix $\boldsymbol{\Phi}$ with the signal as

$$\boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{f} = \boldsymbol{\Phi}^{'} \boldsymbol{x} \tag{2.22}$$

where $\boldsymbol{\Phi}^{'} = \boldsymbol{\Phi}\boldsymbol{\Psi}^T$. It has been shown in [1] that the signal can be recovered by solving the following l_1-norm minimization problem

$$\min \|\tilde{\boldsymbol{x}}\|_{l_1} \ \text{ subject to } \ \boldsymbol{\Phi}^{'} \tilde{\boldsymbol{x}} = \boldsymbol{y} \tag{2.23}$$

where $\|\cdot\|_{l_1}$ denotes the l_1-norm of a vector. The l_1-norm minimization problem of (2.23) is a convex optimization problem and it can be solved in the form of a linear program.

A key assumption about the sampling mechanism is that the sampling vectors $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_K$ which constitute the rows of measurement matrix $\boldsymbol{\Phi}$ should have a

dense representation in the sparsity basis $\boldsymbol{\Psi}$ unlike the signal of interest. This property ensures incoherence between the sampling waveforms and the sparsity basis.

Signal recovery

The restricted isometry property (RIP) has been introduced as a measure of whether or not the signal can be recovered from its compressed samples [1]. Let $\boldsymbol{\Phi}_{\mathcal{T}}$ be a submatrix of $\boldsymbol{\Phi}$ obtained by keeping the columns with their indices in the set $\mathcal{T} \subset \{1, \ldots, N\}$. The S-restricted isometry constant δ_S is the smallest number satisfying the following inequality

$$(1 - \delta_S)\|\boldsymbol{c}\|_{l_2}^2 \leq \|\boldsymbol{\Phi}_{\mathcal{T}}\boldsymbol{c}\|_{l_2}^2 \leq (1 + \delta_S)\|\boldsymbol{c}\|_{l_2}^2 \tag{2.24}$$

for all sets $\mathcal{T}$ of cardinality less than or equal to S and all vectors $\boldsymbol{c}$. In (2.24) $\|\cdot\|_{l_2}$ denotes the Euclidean norm of a vector. If the isometry constant δ_S is small enough, it means that the measurement matrix almost preserves the magnitude of the S-sparse vectors $\boldsymbol{c}$. The following result on the recovery performance of the l_1-norm minimization algorithm of (2.23) is of interest [1].

Theorem 2.2. *Assume that* $\delta_{2S} < \sqrt{2} - 1$. *Then the solution* $\boldsymbol{x}^*$ *to* (2.23) *obeys*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_{l_2} \leq C_0 \frac{\|\boldsymbol{x} - \boldsymbol{x}_S\|_{l_1}}{\sqrt{S}} \tag{2.25}$$

for some constant C_0.

In this theorem, $\boldsymbol{x}_S$ is the S-sparse vector derived from $\boldsymbol{x}$ by retaining only the S elements of $\boldsymbol{x}$ with the largest magnitude. Note that if $\boldsymbol{x}$ itself is S-sparse, then the recovery is exact.

Robustness of CS

In any practical setting, measurements are accompanied by observation noise. In order for CS to be acceptable as a sampling method it should be able to provide a good estimate of the signal from noisy samples. In other words, the sampling method should be robust to observation noise. The measurement noise can be incorporated into the sampling process as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{n}$ where $\boldsymbol{n}$ is the noise term with a bounded energy $\|\boldsymbol{n}\|_{l_2} \leq \gamma$. The l_1-norm minimization problem of (2.23) is then changed accordingly as follows [1]:

$$\min\|\tilde{\boldsymbol{x}}\|_{l_1} \text{ subject to } \|\boldsymbol{\Phi}\tilde{\boldsymbol{x}} - \boldsymbol{y}\|_{l_2} \leq \gamma. \tag{2.26}$$

The above problem is convex and has a unique solution. The following result ensures that the reconstruction error is bounded [1].

Theorem 2.3. *Assume that* $\delta_{2S} < \sqrt{2} - 1$. *Then the solution* $\boldsymbol{x}^*$ *to* (2.26) *satisfies*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_{l_2} \leq C_0 \frac{\|\boldsymbol{x} - \boldsymbol{x}_S\|_{l_1}}{\sqrt{S}} + C_1 \gamma \tag{2.27}$$

for some constants C_0 *and* C_1.

Candidates for the measurement matrix

The RIP has become a measure of whether a certain matrix can be used as a CS measurement matrix or not. Random matrices are suitable choices for a measurement matrix. Among the well known examples of measurement matrices are the following:

Gaussian distributed matrices: Let the entries of the $K \times N$ measurement matrix be taken independently from a normal distribution with zero mean and variance $1/K$. Such a matrix satisfies the RIP for $S \leq CK/\log(N/K)$ with a high probability.

Bernoulli distributed matrices: The $K \times N$ matrix with its entries independently sampled from a Bernoulli distribution ($P\{\boldsymbol{\phi}_{ki} = \pm 1/\sqrt{K}\} = 1/2$). This matrix is conjectured to satisfy the RIP with a high probability if $S \leq CK/\log(N/K)$.

Chapter 3

Segmented Analog to Information Conversion in Compressive Sampling

The sampling process often used in the CS literature consists of two steps. First, an analog signal is sampled at the *Nyquist rate* and then a measurement matrix is applied to the time domain samples in order to collect the compressed samples. This sampling approach, however, defeats one of the primary purposes of CS, which is avoiding high rate sampling. A more practical approach for *direct* sampling and compression of analog signals belonging to the class of signals in a union of subspaces is taken in [41] and the follow up work [42]. Another practical approach to CS, which avoids high rate sampling, has been presented in [1], [43], and the name AIC has been coined. The AIC device consists of several parallel BMIs in which the analog signal is measured against different random sampling waveforms. Therefore, for every collected compressed sample, there is a BMI that multiplies the signal by a sampling waveform and then integrates the result over the sampling period T.

In this chapter, we study our segmented CS method and the segmented CS-based AIC structure which is capable of collecting more samples than the number of BMIs. With more samples, the recovery performance can be improved as compared to the case when the AIC of [1] with the same number of BMIs is used for sampling. The specific contributions of this work are the following. (i) A new segmented CS-based AIC structure is developed. Some preliminary results have been reported in [44]. In this structure, the integration period T is divided into M equal segments such that the sampling rate of the so-obtained segmented AIC is M times higher than the sampling rate of the AIC of [1]. Then the sub-samples collected over M different

segments and K different BMIs are reused so that a larger number of samples (at most K^2 correlated samples) than the number of BMIs is collected. We show that our segmented CS-based AIC technique is equivalent to extending the measurement matrix, which consists of the BMI sampling waveforms, by adding new rows without actually increasing the number of BMIs.[1] (ii) We prove that the RIP, i.e., the sufficient condition for signal recovery based on compressed samples, is satisfied for the extended measurement matrix resulting from the segmented CS-based AIC structure with overwhelming probability if the original matrix of BMI sampling waveforms satisfies the RIP. Thus, our segmented AIC is a valid candidate for CS. (iii) We also prove that the signal recovery performance based on the empirical risk minimization approach can be improved if our segmented AIC is used for sampling instead of the AIC of [1] with the same number of BMIs. Some preliminary results on this topic have been reported in [47]. The mathematical challenge in such a proof is that the samples collected by our segmented AIC are correlated, while all results on performance analysis of the signal recovery available in the literature are obtained for the case of uncorrelated samples.

The rest of this chapter is organized as follows. The segmented CS, is explained in Section 3.2. We prove in Section 3.3 that the extended measurement matrix resulting from the proposed segmented CS satisfies the RIP and, therefore, the segmented CS is a legitimate CS method for AIC. The signal recovery performance analysis for our segmented CS is given in Section 3.4. Section 3.5 shows our simulation results and Section 3.6 concludes the chapter.

3.1 Setup and background for AIC

Setup for CS of analog signals: CS deals with a low rate representation of sparse or compressible signals, i.e., such signals which have few nonzero or, correspondingly, significantly different from zero projections on the vectors of an orthogonal basis (sparsity basis). It is assumed that the analog signal $f(t)$ can be represented or approximated as a linear combination of a finite number of N basis functions $\{\psi_n(t)\}_{n=1}^N$ defined over the time period $t \in [0, T]$. Hence, the signal $f(t)$ is also

[1] In this respect, the works [45] and [46] also need to be mentioned. In [45], Toeplitz-structured measurement matrices are considered, while measurement matrix that is built based on only one random vector with shifts of $D \geq 1$ in between the rows appear in radar imaging application considered in [46].

defined over the same time period and it can be mathematically expressed as

$$f(t) = \sum_{n=1}^{N} x_n \psi_n(t) = \boldsymbol{x}^T \boldsymbol{\Psi}(t) \tag{3.1}$$

where $\{x_n\}_{n=1}^N$ are some coefficients, $\boldsymbol{x} \triangleq (x_1, \ldots, x_N)^T$ is a vector of such coefficients, $\boldsymbol{\Psi}(t) \triangleq (\psi_1(t), \ldots, \psi_N(t))^T$. If $f(t)$ is sparse or compressible, i.e., the vector $\boldsymbol{x}$ has a small number of nonzero or significantly different from zero elements, the basis $\{\psi_n(t)\}_{n=1}^N$ is called a sparsity basis and $\boldsymbol{\Psi}(t)$ maps the discrete vector of coefficients $\boldsymbol{x}$ onto a continuous-time signal $f(t)$. It is known that a universal CS method can be designed to effectively sample and recover S-sparse signals regardless of the specific sparsity domain [1], [33].

The measurement operator $\boldsymbol{\Phi}(t)$ is the collection of $K < N$ sampling waveforms $\{\phi_k(t)\}_{k=1}^K$, i.e., $\boldsymbol{\Phi}(t) \triangleq (\phi_1(t), \ldots, \phi_K(t))^T$. One of the practical choices for the sampling waveforms is a pseudo random ± 1 chip sequence which alternates its value at a rate higher than, for example, the *Nyquist rate* for bandlimited signals [1,43] or just the rate N/T in the traditional CS setup with the finite number of N possible projections. Let the chip duration T_c be set to T/N_c where N_c is the number of chips per signal period T. The discrete measurement y_k can be expressed as

$$y_k = \int_0^T f(t)\phi_k(t)\, dt. \tag{3.2}$$

Then the relation between the $K \times 1$ vector of measurements $\boldsymbol{y} \triangleq (y_1, \ldots, y_K)^T$ and the sparse coefficient vector $\boldsymbol{x}$ can be explained in terms of the $K \times N$ matrix $\boldsymbol{\Phi}' = \boldsymbol{\Phi}(t)\boldsymbol{\Psi}(t)$ with its (k,n)th entry given as

$$[\boldsymbol{\Phi}']_{k,n} = \int_0^T \psi_n(t)\phi_k(t)\, dt. \tag{3.3}$$

Using the matrix $\boldsymbol{\Phi}'$, we can compactly represent the vector of discrete measurements as $\boldsymbol{y} = \boldsymbol{\Phi}'\boldsymbol{x}$. Then, the $K \times N_c$ measurement matrix $\boldsymbol{\Phi}$ and the $N \times N_c$ sparsity basis $\boldsymbol{\Psi}$ can be derived as the discrete equivalents of $\boldsymbol{\Phi}(t)$ and $\boldsymbol{\Psi}(t)$. Specifically, let the entries of $\boldsymbol{\Phi}$ be given as

$$[\boldsymbol{\Phi}]_{k,n} = \int_{(n-1)T_c}^{nT_c} \phi_k(t)\, dt \tag{3.4}$$

and the entries of $\boldsymbol{\Psi}$ as

$$[\boldsymbol{\Psi}]_{m,n} = \int_{(n-1)T_c}^{nT_c} \psi_m(t)\, dt. \tag{3.5}$$

Then it can be seen that $\boldsymbol{\Phi}' = \boldsymbol{\Phi}\boldsymbol{\Psi}^T$. Moreover, the discrete counterpart of the analog signal $f(t)$, denoted as vector $\boldsymbol{f}$, is given as $\boldsymbol{f} = \boldsymbol{\Psi}^T\boldsymbol{x}$. Using the measurement matrix $\boldsymbol{\Phi}$, the $K \times 1$ vector of compressed samples $\boldsymbol{y}$ can be equivalently written as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{f} = \boldsymbol{\Phi}'\boldsymbol{x}$. In the noisy case, the sampling process can be expressed as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{f} + \boldsymbol{n} = \boldsymbol{\Phi}'\boldsymbol{x} + \boldsymbol{n} \tag{3.6}$$

where $\boldsymbol{n}$ is a zero mean noise vector with i.i.d. entries of variance σ_n^2.

In the traditional CS setup for discrete signals, the sparsity basis matrix $\boldsymbol{\Psi}$ with entries given by (3.5) is considered to be $N \times N$ orthonormal matrix. This corresponds to the case when $N_c = N$ for the sampling waveforms. However, there exist applications where this condition is not satisfied and N_c is larger than N. The mathematical analysis and the proves given in this chapter consider the traditional CS setup where the matrix $\boldsymbol{\Psi}$ is square and orthonormal. However, we include some simulation results attesting to the fact that our segmented CS method also works when $N_c > N$.

Another important issue is the number of required compressed samples for successful signal recovery. Among various bounds on the sufficient number of collected compressed samples[2] K ($S < K < N$) required for recovering an S-sparse signal, the first and most popular one is given by the following inequality $S \leq CK/\log(N/K)$ where C is some constant [1]. This bound is derived based on the uniform uncertainty principle which states that $\boldsymbol{\Phi}$ must satisfy the following RIP [1], [49]. As shown in [33], [50], if the entries of $\boldsymbol{\Phi}$ are, for example, independent zero mean Gaussian random variables with variance $1/N$, then $\boldsymbol{\Phi}$ satisfies the RIP for $S \leq CK/\log(N/K)$ with high probability.[3] It is known that the same holds when the entries of $\boldsymbol{\Phi}$ are independent zero mean Bernoulli variables with variance $1/N$ [50]. Since the variance of the elements of $\boldsymbol{\Phi}$ is $1/N$, the following isometry constant inequality is used instead of (2.24) previously given in Chapter 2.

$$\frac{K}{N}(1-\delta_S)\|\boldsymbol{c}\|_{l_2}^2 \leq \|\boldsymbol{\Phi}_{\mathcal{T}}\boldsymbol{c}\|_{l_2}^2 \leq \frac{K}{N}(1+\delta_S)\|\boldsymbol{c}\|_{l_2}^2 \tag{3.7}$$

AIC: The random modulation preintegration (RMPI) structure is proposed for AIC in [1]. The RMPI multiplies the signal with the sampling waveforms in the

[2]See [48] for broader review.

[3]Note that in order to ensure consistency throughout the chapter, the variance of the elements in $\boldsymbol{\Phi}$ is assumed to be $1/N$ instead of $1/K$ as, for example, in [33]. Thus, the multiplier K/N is added in the left- and right-hand sides of (3.7).

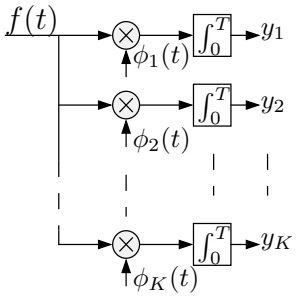


Figure 3.1: The structure of the AIC based on RMPI.

analog domain and then integrates the product over the time period T to produce samples. It implies that the sampling device has a number of parallel BMIs in order to process the analog signal in real-time. The RMPI structure is shown in Fig. 3.1, where the previously introduced notations are used.

Recovery methods: As discussed in Section 2.3.2, a sparse signal can be recovered from its noiseless sample vector $\boldsymbol{y}$ by solving the following problem [33], [51]

$$\min\|\tilde{\boldsymbol{x}}\|_{l_1} \quad \text{subject to} \quad \boldsymbol{\Phi}^{'}\tilde{\boldsymbol{x}} = \boldsymbol{y}. \tag{3.8}$$

In the noisy case, the recovery problem is modified as [52]

$$\min\|\tilde{\boldsymbol{x}}\|_{l_1} \quad \text{subject to} \quad \|\boldsymbol{\Phi}^{'}\tilde{\boldsymbol{x}} - \boldsymbol{y}\|_{l_2} \leq \gamma \tag{3.9}$$

where γ is the bound on the square root of the noise energy. In order to reconstruct an analog signal, i.e., obtain the estimate $\hat{f}(t)$ from the measurement vector $\boldsymbol{y}$, one should first solve for $\tilde{\boldsymbol{x}}$ using (3.8) or (3.9) and then calculate $\hat{f}(t)$ based on (3.1).

Another technique for sparse signal recovery from noisy samples (see [35]) uses the empirical risk minimization method that was first developed in statistical learning theory for approximating an unknown function based on noisy measurements [53]. Note that the empirical risk minimization-based recovery method is of a particular interest since under some approximations (see [35, p. 4041]) it reduces to the well known least absolute shrinkage and selection operator (LASSO) method [54].

In application to CS, the unknown function is the sparse signal and the collected data are the noisy compressed samples. Let the entries of the measurement matrix $\boldsymbol{\Phi}$ be selected with equal probability as $\pm 1/\sqrt{N}$, and the energy of the signal $\boldsymbol{f}$ be bounded so that $\|\boldsymbol{f}\|^2 \leq NB^2$. The risk $r(\hat{\boldsymbol{f}})$ of a candidate reconstruction $\hat{\boldsymbol{f}}$ and

the empirical risk $\hat{r}(\hat{\boldsymbol{f}})$ are defined as [53]

$$r(\hat{\boldsymbol{f}}) \triangleq \frac{\|\hat{\boldsymbol{f}}-\boldsymbol{f}\|^2}{N}+\sigma_n^2, \qquad \hat{r}(\hat{\boldsymbol{f}}) \triangleq \frac{1}{K}\sum_{j=1}^{K}\left(y_j-\boldsymbol{\phi}_j\hat{\boldsymbol{f}}\right)^2. \tag{3.10}$$

Then the candidate reconstruction $\hat{\boldsymbol{f}}_K$ obtained based on K samples can be found as [35]

$$\hat{\boldsymbol{f}}_K = \arg\min_{\hat{\boldsymbol{f}}\in\mathcal{F}(B)}\left\{\hat{r}(\hat{\boldsymbol{f}})+\frac{c(\hat{\boldsymbol{f}})\log 2}{\epsilon K}\right\} \tag{3.11}$$

where $\mathcal{F}(B) \triangleq \{\boldsymbol{f} : \|\boldsymbol{f}\|^2 \le NB^2\}$, $c(\hat{\boldsymbol{f}})$ is a non-negative number assigned to a candidate signal $\hat{\boldsymbol{f}}$, and $\epsilon = 1/\left(50(B+\sigma_n)^2\right)$. Moreover, $\hat{\boldsymbol{f}}_K$ given by (3.11) satisfies the following inequality [35]

$$E\left\{\frac{\|\hat{\boldsymbol{f}}_K-\boldsymbol{f}\|^2}{N}\right\} \le C_1 \min_{\hat{\boldsymbol{f}}\in\mathcal{F}(B)}\left\{\frac{\|\hat{\boldsymbol{f}}-\boldsymbol{f}\|^2}{N}+\frac{c(\hat{\boldsymbol{f}})\log 2+4}{\epsilon K}\right\} \tag{3.12}$$

where $C_1 = [(27-4e)(B/\sigma_n)^2+(50-4\sqrt{2})B/\sigma_n+26]/[(23-4e)(B/\sigma_n)^2+(50-4\sqrt{2})B/\sigma_n+24]$, $e = 2.7183\ldots$.

Let a compressible signal $\boldsymbol{f}$ be defined as a signal for which $\|\boldsymbol{f}^{(m)}-\boldsymbol{f}\|^2 \le NC_Am^{-2\alpha}$, where $\boldsymbol{f}^{(m)}$ is the best m-term approximation of $\boldsymbol{f}$ which is obtained by retaining the m most significant coefficients of vector $\boldsymbol{x}$ and $C_A > 0$ and $\alpha \ge 0$ are some constants. Let also $\mathcal{F}_c(B,\alpha,C_A) \triangleq \{\boldsymbol{f} : \|\boldsymbol{f}\|^2 \le NB^2, \|\boldsymbol{f}^{(m)}-\boldsymbol{f}\|^2 \le NC_Am^{-2\alpha}\}$ be the set of compressible signals. Then based on the weight assignment $c(\hat{\boldsymbol{f}}) = 2\log(N)N_{\hat{\boldsymbol{x}}}$, where $N_{\hat{\boldsymbol{x}}}$ is the actual number of nonzero projections of $\hat{\boldsymbol{f}}$ onto the sparsity basis, the following inequality holds [35]

$$\sup_{\boldsymbol{f}\in\mathcal{F}_c(B,\alpha,C_A)} E\left\{\frac{\|\hat{\boldsymbol{f}}_K-\boldsymbol{f}\|^2}{N}\right\} \le C_1C_2\left(\frac{K}{\log N}\right)^{-2\alpha/(2\alpha+1)} \tag{3.13}$$

where $C_2 = C_2(B,\sigma_n,C_A) > 0$ is a constant.

If signal $\boldsymbol{f}$ is indeed sparse and belongs to $\mathcal{F}_s(B,S) \triangleq \{\boldsymbol{f} : \|\boldsymbol{f}\|^2 \le NB^2, \|\boldsymbol{f}\|_{l_0} \le S\}$, then there exists a constant $C_2' = C_2'(B,\sigma_n) > 0$ such that [35]

$$\sup_{\boldsymbol{f}\in\mathcal{F}_s(B,S)} E\left\{\frac{\|\hat{\boldsymbol{f}}_K-\boldsymbol{f}\|^2}{N}\right\} \le C_1C_2'\left(\frac{K}{S\log N}\right)^{-1}. \tag{3.14}$$

3.2 Segmented compressive sampling method

A significant advantage of the AIC is that it removes the need for high speed sampling. The smaller the number of samples K being collected, the less number of

BMIs is required, thus, the less complex the AIC hardware is. The minimum number of samples required for successful signal recovery is given by the bound based on the RIP given in equation (3.7). The practical rule of thumb for the noiseless case is that four incoherent measurements are required for successful recovery of each nonzero coefficient [1]. However, in the event that the measurements are noisy a larger number of samples allows for a better signal recovery. Indeed, the MSE between the actual and recovered signals is bounded in the noisy case as given in [1, p. 27, Theorem 3] for compressible signals. Such a bound contains a coefficient which depends inversely on the number of available samples. Thus, the larger the number of samples, the better the recovery performance is in the noisy case. Moreover, in practice when the signal sparsity level may not be exactly known, the number of BMIs may be insufficient to guarantee successful signal recovery. Therefore, we may need to collect a larger number of samples to enable recovery. In order to collect a larger number of compressed samples using the AIC structure in Fig. 3.1, we need to increase the hardware complexity by adding more BMIs. The latter makes the AIC device more complex. Therefore, it is desirable to reduce the number of parallel BMIs in the AIC without sacrificing the signal recovery accuracy. This goal can be achieved by increasing the sampling rate of the AIC, but still keeping it significantly lower than the sampling rate required by the A/D converter. The higher sampling rate can be realized by splitting the integration period T in every BMI of the AIC in Fig. 3.1 into $M \leq K$ shorter subperiods (segments). Note that since the original integration period is divided into a number of smaller subperiods, the samples collected over all parallel BMIs during one subperiod do not have complete information about the signal. Therefore, they are called incomplete samples. Hereafter, the complete samples obtained over the whole period T are referred to as just samples, while the incomplete samples are referred to as sub-samples.

3.2.1 The basic idea and the model

The basic idea is to collect a number of sub-samples by splitting the integration period into a number of subperiods and then reuse such sub-samples in order to build additional samples. In this manner, a larger number of samples than the number of BMIs can be collected. It allows for a tradeoff between the AIC and A/D converter structures by allocating $M \leq K$ sub-samples per time-unit T to K BMIs. Indeed, the signal is measured at a low rate by correlating it to a number of sampling

waveforms just as in the AIC, while at the same time the integration period T is split into shorter sub-intervals, i.e., the sampling rate is slightly increased. However, such sampling rate is still significantly lower than that required by the A/D converter.

Let the integration period be split into M sub-intervals and the vectors of sub-samples collected against the sampling waveforms $\{\boldsymbol{\phi}_k\}_{k=1}^K$ being denoted as $\boldsymbol{y}_k = (y_{k,1},\ \ldots,\ y_{k,M})^T,\ k = 1, \ldots, K$. The sub-sample $y_{k,j}$ is given by

$$y_{k,j} = \int_{(j-1)T/M}^{jT/M} f(t)\phi_k(t)dt. \tag{3.15}$$

Then the total number of sub-samples collected by all BMIs over all subperiods is MK. These sub-samples can be gathered in the following $K \times M$ matrix

$$\boldsymbol{Y} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,M} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,M} \\ \vdots & \vdots & \vdots & \vdots \\ y_{K,1} & y_{K,2} & \cdots & y_{K,M} \end{pmatrix} \tag{3.16}$$

where the kth row contains the sub-samples obtained by correlating the measured signal with the waveform $\boldsymbol{\phi}_k$ over M subperiods each of length T/M.

The original K samples, i.e., the samples collected at BMIs over the whole time period T, can be obtained as

$$y_k = \sum_{m=1}^{M} [\boldsymbol{Y}]_{k,m} = \sum_{m=1}^{M} y_{k,m}, \qquad k = 1, \ldots, K. \tag{3.17}$$

In order to construct additional samples, we consider columnwise permuted versions of $\boldsymbol{Y}$. The following definitions are then in order.

The *permutation* π is a one-to-one mapping of the elements of some set $\mathcal{D}$ to itself by simply changing the order of the elements. Then $\pi(k)$ stands for the index of the kth element in the permuted set. For example, let $\mathcal{D}$ consist of the elements of a $K \times 1$ vector $\boldsymbol{z}$, and the order of the elements in $\mathcal{D}$ is the same as in $\boldsymbol{z}$. After applying the permutation function π to $\boldsymbol{z}$, the permuted vector is $\boldsymbol{z}^\pi = \left(z_{\pi(1)}, \ldots, z_{\pi(k)}, \ldots, z_{\pi(K)}\right)^T$. If vector $\boldsymbol{z}$ is itself the vector of indices, i.e., $\boldsymbol{z} = (1, \ldots, K)^T$, then obviously $z_{\pi(k)} = \pi(k)$.

Different permuted versions of the sub-sample matrix $\boldsymbol{Y}$ can be obtained by applying different permutations to different columns of $\boldsymbol{Y}$. Specifically, let $\mathcal{P}^{(i)} = \{\pi_1^{(i)}, \ldots, \pi_j^{(i)}, \ldots, \pi_M^{(i)}\}$ be the ith set of column permutations with $\pi_j^{(i)}$ being the permutation function applied to the jth column of $\boldsymbol{Y}$, and let I stand for the number of such permutation sets. Then according to the above notations, the matrix

resulting from applying the set of permutations $\mathcal{P}^{(i)}$ to the columns of $\boldsymbol{Y}$ can be expressed as $\boldsymbol{Y}^{\mathcal{P}^{(i)}} = \left(\boldsymbol{y}_1^{\pi_1^{(i)}}, \ldots, \boldsymbol{y}_j^{\pi_j^{(i)}}, \ldots, \boldsymbol{y}_M^{\pi_M^{(i)}}\right)$ where $\boldsymbol{y}_j$ is the jth column of $\boldsymbol{Y}$.

Permutation sets $\mathcal{P}^{(i)},\ i = 1, \ldots, I$ are chosen in such a way that all sub-samples in a specific row of $\boldsymbol{Y}^{\mathcal{P}^{(i)}}$ come from different rows of the original sub-sample matrix $\boldsymbol{Y}$ as well as from different rows of other permuted matrices $\boldsymbol{Y}^{\mathcal{P}^{(1)}}, \ldots, \boldsymbol{Y}^{\mathcal{P}^{(i-1)}}$. For example, all sub-samples in a specific row of $\boldsymbol{Y}^{\mathcal{P}^{(1)}}$ must come from different rows of the original matrix $\boldsymbol{Y}$ only, while the sub-samples in a specific row of $\boldsymbol{Y}^{\mathcal{P}^{(2)}}$ come from different rows of $\boldsymbol{Y}$ and $\boldsymbol{Y}^{\mathcal{P}^{(1)}}$ and so on. This requirement is forced to make sure that an additional sample is correlated to an original or another additional sample only over one segment. Then the additional $K\,I$ samples can be obtained based on the permuted matrices $\boldsymbol{Y}^{\mathcal{P}^{(i)}},\ i = 1, \ldots, I$ as

$$y_k^{\mathcal{P}^{(i)}} = \sum_{m=1}^{M} [\boldsymbol{Y}^{\mathcal{P}^{(i)}}]_{k,m}, \qquad k = 1, \ldots, K, \quad i = 1, \ldots, I. \tag{3.18}$$

It is worth noting that in terms of the hardware structure, the sub-samples used to generate additional samples must be chosen from different BMIs as well as different segments. This is equivalent to collecting additional samples by correlating the signal with additional sampling waveforms which are not present among the actual BMI sampling waveforms. Each of these additional sampling waveforms comprises the non-overlapping subperiods of M different original waveforms.

Now the question is how many permuted matrices, which satisfy the above conditions, can be generated based on $\boldsymbol{Y}$. Consider the following $K \times M$ matrix

$$\boldsymbol{Z} \triangleq \underbrace{(\boldsymbol{z}, \boldsymbol{z}, \ldots, \boldsymbol{z})}_{M \text{ times}} \tag{3.19}$$

where $\boldsymbol{z}$ is the vector of indices. Applying the column permutation set $\mathcal{P}^{(i)}$ to the columns of $\boldsymbol{Z}$, a permuted matrix $\boldsymbol{Z}^{\mathcal{P}^{(i)}} = \left(\boldsymbol{z}^{\pi_1^{(i)}}, \ldots, \boldsymbol{z}^{\pi_j^{(i)}}, \ldots, \boldsymbol{z}^{\pi_M^{(i)}}\right)$ is obtained. Then the set of all permuted versions of $\boldsymbol{Z}$ can be denoted as $\mathcal{S}_{\boldsymbol{Z}} = \{\boldsymbol{Z}^{\mathcal{P}^{(1)}}, \ldots, \boldsymbol{Z}^{\mathcal{P}^{(I)}}\}$. With these notations, the following theorem is in order.

Theorem 3.1. *The size of $\mathcal{S}_{\boldsymbol{Z}}$, i.e., the number I of permutation sets $\mathcal{P}^{(i)},\ i = 1, \ldots, I$ which satisfy the conditions*

$$[\boldsymbol{Z}^{\mathcal{P}^{(i)}}]_{k,j} \neq [\boldsymbol{Z}^{\mathcal{P}^{(i)}}]_{k,r},\ \forall \boldsymbol{Z}^{\mathcal{P}^{(i)}} \in \mathcal{S}_{\boldsymbol{Z}},\ \ j \neq r,\ k \in \{1, \ldots, K\},\ j, r \in \{1, \ldots, M\} \tag{3.20}$$

$$\exists! j \text{ or } \nexists j \text{ such that } [\boldsymbol{Z}^{\mathcal{P}^{(i)}}]_{k,j} = [\boldsymbol{Z}^{\mathcal{P}^{(l)}}]_{h,j}, \quad \forall \boldsymbol{Z}^{\mathcal{P}^{(i)}}, \boldsymbol{Z}^{\mathcal{P}^{(l)}} \in \mathcal{S}_{\boldsymbol{Z}},\ \boldsymbol{Z}^{\mathcal{P}^{(i)}} \neq \boldsymbol{Z}^{\mathcal{P}^{(l)}},$$
$$\forall j \in \{1, \ldots, M\} \forall k, h \in \{1, \ldots, K\} \tag{3.21}$$

is at most $K-1$.

Proof. See Appendix A. □

Remark 3.1. Using the property that $z_{\pi(k)} = \pi(k)$ for the vector of indices $\boldsymbol{z}$, the conditions (3.20) and (3.21) can also be expressed in terms of permutations as

$$\pi_j^{(i)}(k) \neq \pi_r^{(i)}(k) \quad \forall i \in \{1,\ldots,I\},\ j \neq r,\ k \in \{1,\ldots,K\},\ j,r \in \{1,\ldots,M\} \tag{3.22}$$

$$\exists! j \text{ or } \nexists j \text{ such that } \pi_j^{(i)}(k) = \pi_j^{(l)}(h)\ \forall i,l \in \{1,\ldots,I\},\ i \neq l,\ \forall j \in \{1,\ldots,M\},$$
$$\forall k,h \in \{1,\ldots,K\}. \tag{3.23}$$

Example 3.1. For this example let the specific choice of index permutations be $\pi_s(k) = ((s+k-2) \bmod K) + 1,\ s,k = 1,\ldots,K$ with π_1 being the identity permutation and 'mod' standing for the modulo operation. For this specific choice, $\pi_j^{(i)} = \pi_{[i(j-1) \bmod K]+1},\ i = 1,\ldots,K-1,\ j = 1,\ldots,M$. Consider the following matrix notation for the set $\mathcal{P}$ where the elements along the ith row are the permutations $\mathcal{P}^{(i)},\ i = 1,\ldots,I$

$$\mathcal{P} \triangleq \begin{pmatrix} \mathcal{P}^{(1)} \\ \mathcal{P}^{(2)} \\ \mathcal{P}^{(3)} \\ \vdots \\ \mathcal{P}^{(K-2)} \\ \mathcal{P}^{(K-1)} \end{pmatrix} = \begin{pmatrix} \pi_1^{(1)} & \pi_2^{(1)} & \pi_3^{(1)} & \ldots & \pi_M^{(1)} \\ \pi_1^{(2)} & \pi_2^{(2)} & \pi_3^{(2)} & \ldots & \pi_M^{(2)} \\ \pi_1^{(3)} & \pi_2^{(3)} & \pi_3^{(3)} & \ldots & \pi_M^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_1^{(K-2)} & \pi_2^{(K-2)} & \pi_3^{(K-2)} & \ldots & \pi_M^{(K-2)} \\ \pi_1^{(K-1)} & \pi_2^{(K-1)} & \pi_3^{(K-1)} & \ldots & \pi_M^{(K-1)} \end{pmatrix}$$

$$= \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & \ldots & \pi_M \\ \pi_1 & \pi_3 & \pi_5 & \ldots & \pi_{[2(M-1) \bmod K]+1} \\ \pi_1 & \pi_4 & \pi_7 & \ldots & \pi_{[3(M-1) \bmod K]+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_1 & \pi_{K-1} & \pi_{K-3} & \ldots & \pi_{[(K-2)(M-1) \bmod K]+1} \\ \pi_1 & \pi_K & \pi_{K-1} & \ldots & \pi_{[(K-1)(M-1) \bmod K]+1} \end{pmatrix}. \tag{3.24}$$

Note that not all permutations $\mathcal{P}^{(i)},\ i = 1,\ldots,I$ used in (3.24) may be permissible. In fact, the set of permutations $\mathcal{P}^{(i)}$ with $K/gcd(i,K) < M$ has at least one repeated permutation that contradicts the condition (3.22). Here $gcd(\cdot,\cdot)$ stands for the greatest common devisor of two numbers. For example, for $K = 8$ and $M = 4$, $K/gcd(4,K) = 2 < M$ and $\mathcal{P}^{(4)}$ is impermissible. Therefore, instead of $K-1 = 7$,

only the following 6 sets of permutations are allowed

$$\mathcal{P} = \begin{pmatrix} \pi_1^{(1)} & \pi_2^{(1)} & \pi_3^{(1)} & \pi_4^{(1)} \\ \pi_1^{(2)} & \pi_2^{(2)} & \pi_3^{(2)} & \pi_4^{(2)} \\ \pi_1^{(3)} & \pi_2^{(3)} & \pi_3^{(3)} & \pi_4^{(3)} \\ \pi_1^{(4)} & \pi_2^{(4)} & \pi_3^{(4)} & \pi_4^{(4)} \\ \pi_1^{(5)} & \pi_2^{(5)} & \pi_3^{(5)} & \pi_4^{(5)} \\ \pi_1^{(6)} & \pi_2^{(6)} & \pi_3^{(6)} & \pi_4^{(6)} \end{pmatrix} = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \\ \pi_1 & \pi_3 & \pi_5 & \pi_7 \\ \pi_1 & \pi_4 & \pi_7 & \pi_2 \\ \pi_1 & \pi_6 & \pi_3 & \pi_8 \\ \pi_1 & \pi_7 & \pi_5 & \pi_3 \\ \pi_1 & \pi_8 & \pi_7 & \pi_6 \end{pmatrix}. \tag{3.25}$$

Theorem 3.1 shows how many different permuted versions of the original sub-sample matrix $\boldsymbol{Y}$ can be obtained such that the correlation between the original and additional samples would be minimal. Indeed, since the set of sub-samples that are used to build additional samples is chosen in such a way that additional samples have at most one sub-sample in common with the previous samples, i.e., conditions (3.22) and (3.23) are satisfied, the set of permutations (3.24) is a valid candidate. The ith element of $\mathcal{P}$, i.e., the element $\mathcal{P}^{(i)} = \left(\pi_1^{(i)}, \ldots, \pi_M^{(i)}\right)$, is the set of permutations applied to $\boldsymbol{Y}$ to obtain $\boldsymbol{Y}^{\mathcal{P}^{(i)}}$. Adding up the entries along the rows of $\boldsymbol{Y}^{\mathcal{P}^{(i)}}$, a set of K additional samples can be obtained.

Example 3.2. Let the number of additional samples K_a be at most K. This means that all permutations are given by only $\mathcal{P}^{(1)}$ in (3.24). In this special case, the sub-sample selection method can be given as follows. For constructing the $(K+1)$st sample, M sub-samples on the main diagonal of $\boldsymbol{Y}$ are summed up together. Then the M sub-samples on the second diagonal are used to construct the $(K+2)$nd sample, and so on up to the K_ath sample. Mathematically, the so constructed additional samples can be expressed in terms of the elements of $\boldsymbol{Y}$ as

$$y_{K+k} = \sum_{m=1}^{M} y_{l,m}, \qquad k = 1 \ldots, K_a \tag{3.26}$$

where $l = [(k+m-2) \bmod K] + 1$ and $K_a \leq K$. Fig. 3.2 shows schematically how the sub-samples are selected in this example.

The proposed segmented sampling process can be equivalently expressed in terms of the measurement matrix. Let $\boldsymbol{\Phi}$ be the original $K \times N$ measurement matrix. Let the kth row of the matrix $\boldsymbol{\Phi}$ be $\boldsymbol{\phi}_k = \left(\boldsymbol{\phi}_{k,1}, \ldots, \boldsymbol{\phi}_{k,M}\right)$ where $\boldsymbol{\phi}_{k,j},\ j = 1, \ldots, M$ are some vectors. Let also for simplicity, the length of $\boldsymbol{\phi}_{k,j}$ be N/M where N/M is a positive integer number. The set of permutations applied to $\boldsymbol{Y}$ in order to obtain $\boldsymbol{Y}^{\mathcal{P}^{(i)}}$ is $\mathcal{P}^{(i)}$. Then the operation $\boldsymbol{\Phi}^{\mathcal{P}^{(i)}}$ can be expressed as follows. The first group

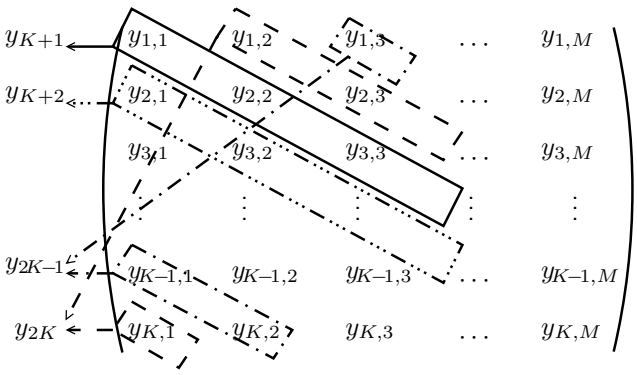


Figure 3.2: Sub-sample selection principle for building additional samples in Example 3.2.

of N/M columns of $\mathbf{\Phi}$, which are the vectors $\boldsymbol{\phi}_{k,1},\ k = 1, \ldots, K$, are permuted with $\pi_1^{(i)}$. The second group of N/M columns of $\mathbf{\Phi}$ are permuted with $\pi_2^{(i)}$ and so on until the last group of N/M columns of $\mathbf{\Phi}$ which are permuted with $\pi_M^{(i)}$. Then the $K_e \times N$ extended measurement matrix which combines all possible permutations $\mathcal{P}^{(i)},\ i = 1, \ldots, I$ can be expressed as

$$\mathbf{\Phi}_e = \left(\mathbf{\Phi}^T, (\mathbf{\Phi}^{\mathcal{P}^{(1)}})^T, \ldots, (\mathbf{\Phi}^{\mathcal{P}^{(I)}})^T\right)^T \tag{3.27}$$

where $K_e \triangleq K + K_a = K + KI$.

Example 3.3. Continuing with the set up used in Example 3.2, let $K_a \leq K$. Then the extended measurement matrix is

$$\mathbf{\Phi}_e = \begin{pmatrix} \mathbf{\Phi} \\ \mathbf{\Phi}_1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_{1,1} & \boldsymbol{\phi}_{1,2} & \cdots & \boldsymbol{\phi}_{1,M} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}_{K,1} & \boldsymbol{\phi}_{K,2} & \cdots & \boldsymbol{\phi}_{K,M} \\ \boldsymbol{\phi}_{1,1} & \boldsymbol{\phi}_{2,2} & \cdots & \boldsymbol{\phi}_{M,M} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}_{K_a,1} & \boldsymbol{\phi}_{\pi_2(K_a),M} & \cdots & \boldsymbol{\phi}_{\pi_M(K_a),M} \end{pmatrix} \tag{3.28}$$

where $\mathbf{\Phi}_1$ contains only K_a rows of $\mathbf{\Phi}^{\mathcal{P}^{(1)}}$ and $\mathbf{\Phi}_1 = \mathbf{\Phi}^{\mathcal{P}^{(1)}}$ if $K_a = K$.

3.2.2 Implementation issues and discussion

Due to the special structure of the extended measurement matrix $\mathbf{\Phi}_e$, the sampling hardware needs only K parallel BMIs for collecting KI samples. These BMIs are essentially the same as those in Fig. 3.1. The only difference is that the integration period T is divided into M equal subperiods. At the end of every subperiod, each integrator's output is sampled and the integrator is reset. Therefore, some factors,

which may influence the complexity of a hardware implementation of the proposed segmented AIC, are the following. Since the sampling rate of the segmented AIC is M times higher than that of the conventional AIC with the same number of BMIs, the complexity of the segmented AIC can slightly increase as compared to the conventional AIC with K BMIs. However, the rate increased in $M \leq K$ times is still by far less than, for example, the required *Nyquist rate* which depends on the signal bandwidth. As compared to the AIC with KI BMIs, the segmented AIC has only K BMIs, that makes the complexity of the segmented AIC for collecting KI samples significantly smaller than that of the conventional AIC with KI BMIs. In addition, a multiplexer which selects the sub-samples for constructing additional samples is needed in the proposed segmented AIC. It is worth noting, however, that partial sums can be kept for constructing the samples (original and additional), that is, the results of the integration are updated and accumulated for each sample iteratively after each subperiod. In this way, there is no need of designing the circuitry for memorizing the matrix of sub-samples $\boldsymbol{Y}$, but only the partial sums for each sample are memorized at any current subperiod. One more factor which may have an effect on the performance of the segmented AIC is the hardware sampling noise introduced at time instances $mT/M,\ m = 1, \ldots, M$ when the output of each BMI is sampled to collect a sub-sample. This sampling noise appears M times over a time period T for the segmented AIC while it appears once over T for the conventional AIC. However, the amount of the hardware sampling noise depends on the specific hardware implementation of the sampler and is out of the scope of this chapter.

Finally, it is worth noting that the possibility of improving the signal recovery performance due to increasing the sampling rate in each BMI of the proposed segmented AIC agrees with the convention that the recovery performance cannot be improved only due to the post processing. Moreover, note that since the original random sampling waveforms are linearly independent with high probability, the additional sampling waveforms of our segmented CS method are also linearly independent with overwhelming probability. However, a sufficient condition that guarantees that the extended measurement matrix of the proposed segmented CS-based AIC scheme is a valid choice is the RIP. Therefore, the RIP for the proposed segmented CS method is analyzed in the next section.

3.3 RIP for the segmented compressive sampling method

The purpose of this section is to show that the extended measurement matrix $\mathbf{\Phi}_e$ in (3.27) satisfies the RIP if the original measurement matrix $\mathbf{\Phi}$ satisfies it. The latter will also imply that $\mathbf{\Phi}_e$ can be used as a valid CS measurement matrix. In our setup it is only assumed that the elements of the original measurement matrix are i.i.d. zero mean Gaussian variables and the measurement matrix is extended by constructing its permuted versions as described in the previous section.

Let us first consider the special case of Example 3.3. In this case, $\mathbf{\Phi}$, $\mathbf{\Phi}_1$, and $\mathbf{\Phi}_e$ are the original measurement matrix, the matrix of additional sampling waveforms, and the extended measurement matrix given by (3.28), respectively. Let the matrix $\mathbf{\Phi}$ satisfy the RIP with sufficiently high probability. For example, let the elements of $\mathbf{\Phi}$ be i.i.d. zero mean Gaussian random variables with variance $1/N$. Let $\mathcal{T}$ be any subset of size S of the set $\{1, \ldots, N\}$. Then for any $0 < \delta_S < 1$, the matrix $\mathbf{\Phi}_{\mathcal{T}}$, which is a sub-matrix of $\mathbf{\Phi}$ which consists of only the columns with their indices in the set $\mathcal{T}$ satisfies (3.7) with the following probability [50]

$$\Pr\{\mathbf{\Phi}_{\mathcal{T}} \text{ satisfies (3.7)}\} \geq 1 - 2\left(12/\delta_S\right)^S e^{-C_0(\delta_S/2)K} \tag{3.29}$$

where $C_0\left(\delta_S/2\right) = \delta_S^2/16 - \delta_S^3/48$. Hereafter, the notation C_0 is used instead of $C_0\left(\delta_S/2\right)$ for brevity.

First, the following auxiliary result on the extended measurement matrix $\mathbf{\Phi}_e$ is of interest.

Lemma 3.1. *Let the elements of the measurement matrix $\mathbf{\Phi}$ be i.i.d. zero mean Gaussian variables with variance $1/N$, $\mathbf{\Phi}_e$ be formed as shown in (3.28), and $\mathcal{T} \subset \{1, \ldots, N\}$ of size S. If K_a is chosen so that $min\{K, K_a + M - 1\} \leq \lceil (K + K_a)/2 \rceil$, then for any $0 < \delta_S < 1$, the following inequality holds*

$$Pr\{(\mathbf{\Phi}_e)_{\mathcal{T}} \text{ satisfies (3.7)}\} \geq 1 - 4\left(12/\delta_S\right)^S e^{-C_0 \lfloor \frac{K+K_a}{2} \rfloor} \tag{3.30}$$

where $\lceil x \rceil$ and $\lfloor x \rfloor$ are the smallest integer larger than or equal to x and the largest integer smaller than or equal to x, respectively, and C_0 is a constant given after (3.29).

Proof. See Appendix B. □

Using the above lemma, the following main result, which states that the extended measurement matrix $\mathbf{\Phi}_e$ in (3.28) satisfies the RIP with overwhelming probability, is in order.

Theorem 3.2. *Let $\mathbf{\Phi}_e$ be formed as in* (3.28) *and let the elements of $\mathbf{\Phi}$ be i.i.d. zero mean Gaussian variables with variance $1/N$. If $\min\{K, K_a+M-1\} \leq \lceil (K+K_a)/2 \rceil$, then for any $0 < \delta_S < 1$, there exist constants C_3 and C_4, which depend only on δ_S, such that for $S \leq C_3 \lfloor (K+K_a)/2 \rfloor / \log(N/S)$ the inequality* (3.7) *holds for all S-sparse vectors with probability that satisfies the following inequality*

$$Pr\{\mathbf{\Phi}_e \text{ satisfies RIP}\} \geq 1 - 4e^{-C_4 \lfloor (K+K_a)/2 \rfloor} \tag{3.31}$$

where $C_4 = C_0 - C_3 \left[1 + (1 + \log(12/\delta_S)) / \log(N/S)\right]$ and C_3 is small enough that guarantees that C_4 is positive.

Proof. See Appendix C. □

Let us consider now the general case when the number of additional samples K_a is larger than the number of BMIs K, i.e., $K_a > K$ and $K_e > 2K$, and the extended measurement matrix is given by (3.27). Note that while proving Lemma 3.1 for the special case of Example 3.3, we were able to split the rows of $\mathbf{\Phi}_e$ into two sets each consisting of independent entries. In the general case, some of the entries of the original measurement matrix appear more than twice in the extended measurement matrix $\mathbf{\Phi}_e$, and it is no longer possible to split the rows of $\mathbf{\Phi}_e$ into only two sets with independent entries. Because of the way the additional samples are built, the samples $y_{lK+1}, y_{lK+2}, \ldots, y_{(l+1)K}$ obtained based on the permuted matrix $\boldsymbol{Y}^{\mathcal{P}^{(l)}}$ are uncorrelated with each other, but they are correlated with every other set of samples obtained based on the original matrix $\boldsymbol{Y}$ and the permuted matrices $\boldsymbol{Y}^{\mathcal{P}^{(i)}}, \forall i,\ i \neq l$. Thus, the following principle can be used for partitioning the rows of $\mathbf{\Phi}_e$ into the sets with independent entries. First, the rows corresponding to the original samples form a single set with independent entries, then the rows corresponding to the first set of additional samples based on the matrix $\boldsymbol{Y}^{\mathcal{P}^{(1)}}$ form another set and so on. Then the number of such sets is $n_p = \lceil K_e/K \rceil$, while the size of each set is

$$K_i = \begin{cases} K, & 1 \leq i < \lceil \frac{K_e}{K} \rceil - 1 \\ K_e - (\lceil \frac{K_e}{K} \rceil - 1)K, & i = \lceil \frac{K_e}{K} \rceil. \end{cases} \tag{3.32}$$

The extended measurement matrix in (3.27) can be rewritten as

$$\mathbf{\Phi}_e = \left((\mathbf{\Phi}_e)_1^T, (\mathbf{\Phi}_e)_2^T, \ldots, (\mathbf{\Phi}_e)_{n_p}^T \right)^T \tag{3.33}$$

where $(\mathbf{\Phi}_e)_i$ is the ith partition of $\mathbf{\Phi}_e$ of size given by (3.32). Then the general form of Lemma 3.1 is as follows.

Lemma 3.2. *Let the elements of the measurement matrix $\mathbf{\Phi}$ be i.i.d. zero mean Gaussian variables with variance $1/N$, $\mathbf{\Phi}_e$ be the extended measurement matrix (3.27), and $\mathcal{T} \subset \{1, \ldots, N\}$ be of size S. Let also $K_a > K$ and $n_p = \lceil K_e/K \rceil$. Then, for any $0 < \delta_S < 1$, the following inequality holds*

$$\begin{aligned} Pr\{(\mathbf{\Phi}_e)_{\mathcal{T}} \text{ satisfies (3.7)}\} \geq 1 - 2(n_p - 1)\left(12/\delta_S\right)^S \left(e^{-C_0 K}\right) \\ - 2\left(12/\delta_S\right)^S \left(e^{-C_0 K_{n_p}}\right) \end{aligned} \tag{3.34}$$

where $K_{n_p} = K_e - \left(\lceil \frac{K_e}{K} \rceil - 1\right) K$ and C_0 is a constant given after (3.29).

Proof. See Appendix D. □

Lemma 3.2 is needed to prove that the extended measurement matrix (3.33) satisfies the RIP with overwhelming probability. Therefore, the general version of Theorem 3.2 is as follows.

Theorem 3.3. *Let the elements of $\mathbf{\Phi}$ be i.i.d. zero mean Gaussian variables with variance $1/N$ and $\mathbf{\Phi}_e$ be formed as in (3.27). If $K_a > K$, then there exist constants C_3, C_4, and C_4' for any $0 < \delta_S < 1$, such that for $S \leq C_3 K_{n_p}/\log(N/S)$ the inequality (3.7) holds for all S-sparse vectors with probability that satisfies the following inequality*

$$Pr\{\mathbf{\Phi}_e \text{ satisfies RIP}\} \geq 1 - 2(n_p - 1)e^{-C_4' K} - 2e^{-C_4 K_{n_p}} \tag{3.35}$$

where $C_4' = C_0 - (C_3 K_{n_p}/K)\left[1 + (1 + \log(12/\delta_S))/\log(N/S)\right]$, C_4 is given after (3.31), and C_3 is small enough to guarantee that C_4 and C_4' are both positive.

Proof. See Appendix E. □

When splitting the rows of $\mathbf{\Phi}_e$ in a number of sets as described before Lemma 3.2, it may happen that the last subset $(\mathbf{\Phi}_e)_{n_p}$ has the smallest size K_{n_p}. As a result, the dominant term in (3.35) will likely be the term $2e^{-C_4 K_{n_p}}$. It may lead to a more stringent sparsity condition, that is, $S \leq C_3 K_{n_p}/\log(N/S)$. To improve the lower bound in (3.35), we can move some of the rows from $(\mathbf{\Phi}_e)_{n_p-1}$ to $(\mathbf{\Phi}_e)_{n_p}$ in order to make the last two partitions of almost the same size. Then the requirement on the sparsity level will become $S \leq C_3 K'/\log(N/S)$ where $K' = \lfloor (K + K_{n_p})/2 \rfloor$. Therefore, the lower bound on the probability calculated in (3.35) improves.

3.4 Performance analysis of the recovery

In this section, we aim at answering the question whether signal recovery also improves if the proposed segmented CS method, i.e., the extended measurement matrix $\boldsymbol{\Phi}_e$ in (3.27), is used instead of the original matrix $\boldsymbol{\Phi}$. The study is performed based on the empirical risk minimization method for signal recovery from noisy random projections [35]. As mentioned in Section 3.1, the LASSO method can be viewed as one of the possible implementations of the empirical risk minimization method.

We first consider the special case of Example 3.3 when the extended measurement matrix is given by (3.28). Let the entries of the measurement matrix $\boldsymbol{\Phi}$ be selected with equal probability as $\pm 1/\sqrt{N}$, i.e., be i.i.d. Bernoulli distributed with variance $1/N$. The Bernoulli case is used here in order to keep our derivations short by only emphasizing the differences caused by our construction of matrix $\boldsymbol{\Phi}_e$ with correlated rows as compared to the case analyzed in [35], where the measurement matrix consists of all i.i.d. entries. Moreover, the Bernoulli case is the one which is practically appealing. Note that our results can be easily applied to the case of Gaussian distributed entries of $\boldsymbol{\Phi}$ by only changing the moments of Bernoulli distribution to the moments of Gaussian distribution.

Let $r(\hat{\boldsymbol{f}}, \boldsymbol{f}) \triangleq r(\hat{\boldsymbol{f}}) - r(\boldsymbol{f})$ be the *excess risk* between the candidate reconstruction $\hat{\boldsymbol{f}}$ of the signal sampled using the extended measurement matrix $\boldsymbol{\Phi}_e$ and the actual signal $\boldsymbol{f}$, and $\hat{r}(\hat{\boldsymbol{f}}, \boldsymbol{f}) \triangleq \hat{r}(\hat{\boldsymbol{f}}) - \hat{r}(\boldsymbol{f})$ be the *empirical excess risk* between the candidate signal reconstruction and the actual signal. Then the difference between the *excess risk* and the *empirical excess risk* can be found as

$$r(\hat{\boldsymbol{f}}, \boldsymbol{f}) - \hat{r}(\hat{\boldsymbol{f}}, \boldsymbol{f}) = \frac{1}{K_e} \sum_{j=1}^{K_e} (U_j - E[U_j]) \tag{3.36}$$

where $U_j \triangleq (y_j - \boldsymbol{\phi}_j \boldsymbol{f})^2 - (y_j - \boldsymbol{\phi}_j \hat{\boldsymbol{f}})^2$.

The MSE between the candidate reconstruction and the actual signal can be expressed as [53]

$$\text{MSE} \triangleq E\left\{\|\boldsymbol{g}\|^2\right\} = N r(\hat{\boldsymbol{f}}, \boldsymbol{f}) \tag{3.37}$$

where $\boldsymbol{g} \triangleq \hat{\boldsymbol{f}} - \boldsymbol{f}$. Therefore, if we know an upper bound on the right-hand side of (3.36), denoted hereafter as U, we can immediately find an upper bound on the MSE in the form $\text{MSE} \leq N\hat{r}(\hat{\boldsymbol{f}}, \boldsymbol{f}) + NU$. In other words, to find the candidate reconstruction $\hat{\boldsymbol{f}}$, one can minimize $\hat{r}(\hat{\boldsymbol{f}}, \boldsymbol{f}) + U$. This minimization will result in a bound on the MSE as in (3.12).

The Craig-Bernstein inequality [35], [55] can be used in order to find an upper bound U on the right-hand side of (3.36). In our notations the Craig-Bernstein inequality states that the probability of the following event

$$\frac{1}{K_e}\sum_{j=1}^{K_e}(U_j - E\{U_j\}) \leq \frac{\log\left(\frac{1}{\delta}\right)}{K_e\epsilon} + \frac{\epsilon \operatorname{var}\left\{\sum_{j=1}^{K_e} U_j\right\}}{2K_e(1-\zeta)} \tag{3.38}$$

is greater than or equal to $1-\delta$ for $0 < \epsilon h \leq \zeta < 1$, if the random variables U_j satisfy the following moment condition for some $h > 0$ and for all $k \geq 2$

$$E\left\{|U_j - E\{U_j\}|^k\right\} \leq \frac{k!\operatorname{var}\{U_j\}\, h^{k-2}}{2}. \tag{3.39}$$

The second term on the right-hand side of (3.38) contains the variance $\operatorname{var}\left\{\sum_{j=1}^{K_e} U_j\right\}$, which we need to calculate or at least find an upper bound on it.

In the case of the extended measurement matrix, the random variables U_j, $j = 1, \ldots, K_e$ all satisfy the moment condition for the Craig-Bernstein inequality [55] with the same coefficient $h = 16B^2e + 8\sqrt{2}B\sigma_n$ where σ_n^2 is the variance of the Gaussian noise.[4] Moreover, it is easy to show that the following bound on the variance of U_j is valid for the extended measurement matrix

$$\operatorname{var}\{U_j\} \leq \left(2\frac{\|\boldsymbol{g}\|^2}{N} + 4\sigma_n^2\right)\frac{\|\boldsymbol{g}\|^2}{N} \leq \left(8B^2 + 4\sigma_n^2\right) r(\hat{\boldsymbol{f}}, \boldsymbol{f}). \tag{3.40}$$

However, unlike [35], in the case of the extended measurement matrix, the variables U_j are not independent from each other. Thus, we can not simply replace the term $\operatorname{var}\left\{\sum_{j=1}^{K_e} U_j\right\}$ with the sum of the variances for U_j, $j = 1, \ldots, K_e$. Using the definition of the variance, we can write that

$$\begin{aligned}
\operatorname{var}\left\{\sum_{j=1}^{K_e} U_j\right\} &\triangleq E\left\{\left(\sum_{j=1}^{K_e} U_j\right)^2\right\} - \left(E\left\{\sum_{j=1}^{K_e} U_j\right\}\right)^2 \\
&= \sum_{j=1}^{K_e} E\{U_j^2\} + 2\sum_{i=1}^{K_e-1}\sum_{j=i+1}^{K_e} E\{U_iU_j\} - K_e^2\left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2 \\
&= \sum_{j=1}^{K_e}\left(E\{U_j^2\} - \left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2\right) + 2\sum_{i=1}^{K_e-1}\sum_{j=i+1}^{K_e}\left(E\{U_iU_j\} - \left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2\right) \\
&= \sum_{j=1}^{K_e}\operatorname{var}\{U_j\} + 2\sum_{i=1}^{K_e-1}\sum_{j=i+1}^{K_e}\left(E\{U_iU_j\} - \left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2\right)
\end{aligned} \tag{3.41}$$

[4] The derivation of the coefficient h coincides with a similar derivation in [35], and therefore, is omitted.

where the upper bound on var$\{U_j\}$ is given by (3.40). Using the fact following form the noisy model (3.6) that the random noise components n_i and n_j are independent from $\boldsymbol{\phi}_i\boldsymbol{g}$ and $\boldsymbol{\phi}_j\boldsymbol{g}$, respectively, $E\{U_iU_j\}$ can be expressed as

$$\begin{aligned} E\{U_iU_j\} &= E\big\{[2n_i\boldsymbol{\phi}_i\boldsymbol{g}-(\boldsymbol{\phi}_i\boldsymbol{g})^2][2n_j\boldsymbol{\phi}_j\boldsymbol{g}-(\boldsymbol{\phi}_j\boldsymbol{g})^2]\big\} \\ &= 4E\big\{n_in_j\big\}E\big\{\boldsymbol{\phi}_i\boldsymbol{g}\boldsymbol{\phi}_j\boldsymbol{g}\big\} - 2E\big\{n_i\big\}E\big\{\boldsymbol{\phi}_i\boldsymbol{g}(\boldsymbol{\phi}_j\boldsymbol{g})^2\big\} \\ &\quad - 2E\big\{n_j\big\}E\big\{\boldsymbol{\phi}_j\boldsymbol{g}(\boldsymbol{\phi}_i\boldsymbol{g})^2\big\} + E\big\{(\boldsymbol{\phi}_i\boldsymbol{g})^2(\boldsymbol{\phi}_j\boldsymbol{g})^2\big\}. \end{aligned} \tag{3.42}$$

The latter expression can be further simplified using the fact that $E\{n_i\} = E\{n_j\} = 0$. Thus, we obtain that

$$E\{U_iU_j\} = 4E\big\{n_in_j\big\}E\big\{(\boldsymbol{\phi}_i\boldsymbol{g})(\boldsymbol{\phi}_j\boldsymbol{g})\big\} + E\big\{(\boldsymbol{\phi}_i\boldsymbol{g})^2(\boldsymbol{\phi}_j\boldsymbol{g})^2\big\}. \tag{3.43}$$

It is easy to verify that if $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ are independent from each other, then $E(U_iU_j) = E\left\{(\boldsymbol{\phi}_i\boldsymbol{g})^2\right\}E\left\{(\boldsymbol{\phi}_j\boldsymbol{g})^2\right\} = \left(\|\boldsymbol{g}\|^2/\ N\right)^2$ as in [35]. However, in our case, $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ may depend on each other. If they indeed depend on each other, they have $L = N/M$ common entries, while the rest of the entries are independent. In addition, the additive noise terms n_i and n_j are no longer independent random variables as well and, thus, $E\big\{n_in_j\big\} = \sigma_n^2/M$. Without loss of generality, let the first L entries of $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ be the same, that is,

$$\boldsymbol{\phi}_i\boldsymbol{g} = \overbrace{g_1a_1 + \ldots + g_La_L}^{A} + \overbrace{g_{L+1}\phi_{i,L+1} + \ldots + g_N\phi_{i,N}}^{P_i} \tag{3.44}$$

$$\boldsymbol{\phi}_j\boldsymbol{g} = \overbrace{g_1a_1 + \ldots + g_La_L}^{A} + \overbrace{g_{L+1}\phi_{j,L+1} + \ldots + g_N\phi_{j,N}}^{P_j} \tag{3.45}$$

with $a_1, ..., a_L$ being the common part between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$.

Let $\boldsymbol{g}_A$ be a sub-vector of $\boldsymbol{g}$ containing the L elements of $\boldsymbol{g}$ corresponding to the common part between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, and $\boldsymbol{g}_{A'}$ be the sub-vector comprising the rest of the elements. Then using the fact that A, P_i, and P_j are all zero mean independent random variables, we can express $E\{(\boldsymbol{\phi}_i\boldsymbol{g})(\boldsymbol{\phi}_j\boldsymbol{g})\}$ from the first term on the right-hand side of (3.43) as

$$\begin{aligned} E\{(\boldsymbol{\phi}_i\boldsymbol{g})(\boldsymbol{\phi}_j\boldsymbol{g})\} &= E\{(A+P_i)(A+P_j)\} = E\{A^2\} + E\{AP_i\} + E\{AP_j\} + E\{P_iP_j\} \\ &= E\{A^2\} = \frac{\left(\sum_{k=1}^{L} g_k^2\right)^2}{N} = \frac{\|\boldsymbol{g}_A\|^2}{N}. \end{aligned} \tag{3.46}$$

Similarly, the second term on the right-hand side of (3.43) can be expressed as

$$E\big\{(\boldsymbol{\phi}_i\boldsymbol{g})^2(\boldsymbol{\phi}_j\boldsymbol{g})^2\big\} = E\left\{(A^2 + P_i^2 + 2AP_i)(A^2 + P_j^2 + 2AP_j)\right\}. \tag{3.47}$$

Using the facts that $4E\{n_i n_j\} = 4\sigma_n^2/M$, $E\{A^2\} = \|\boldsymbol{g}_A\|^2/N$, and $E\{P_i^2\} = \|\boldsymbol{g}_{A'}\|^2/N$, the expression (3.47) can be further rewritten as

$$\begin{aligned} E\{(\boldsymbol{\phi}_i\boldsymbol{g})^2(\boldsymbol{\phi}_j\boldsymbol{g})^2\} &= E\left\{A^4 + A^2P_i^2 + A^2P_j^2 + P_i^2P_j^2\right\} \\ &= E\{A^4\} + 2\frac{\|\boldsymbol{g}_A\|^2}{N}\cdot\frac{\|\boldsymbol{g}_{A'}\|^2}{N} + \left(\frac{\|\boldsymbol{g}_{A'}\|^2}{N}\right)^2 \\ &= E\{A^4\} + \left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2 - \left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2. \end{aligned} \tag{3.48}$$

Substituting (3.46) and (3.48) into (3.43), we obtain that

$$E\{U_iU_j\} = \frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}_A\|^2}{N} + E\{A^4\} + \left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2 - \left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2. \tag{3.49}$$

Moreover, substituting (3.49) into (3.41), we find that

$$\operatorname{var}\left\{\sum_{j=1}^{K_e}U_j\right\} = \sum_{j=1}^{K_e}\operatorname{var}\{U_j\} + 2\sum_{\boldsymbol{\phi}_i,\boldsymbol{\phi}_j\,\text{dependent}}\left(E\{A^4\} - \left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2 + \frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}_A\|^2}{N}\right). \tag{3.50}$$

Since the extended measurement matrix is constructed so that the waveforms $\{\boldsymbol{\phi}_i\}_{i=K+1}^{K_e}$ are built upon M rows of the original matrix and using then the inequality[5] $E\{A^4\} - \left(\|\boldsymbol{g}_A\|^2/N\right)^2 \leq 2\left(\|\boldsymbol{g}_A\|^2/N\right)^2$ for all these M rows, we obtain for every additional $\boldsymbol{\phi}_i$ that

$$\sum_{k=1}^{M}\left(E\{A^4\} - \left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2 + \frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}_A\|^2}{N}\right) \leq \sum_{k=1}^{M}\left(2\left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2 + \frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}_A\|^2}{N}\right) \tag{3.51}$$

where $\boldsymbol{g}_A$ corresponds to the first L entries of $\boldsymbol{g}$ for $k=1$, to entries from $L+1$ to $2L$ for $k=2$ and so on. Applying also the triangle inequality, we find that

$$\sum_{k=1}^{M}\left(2\left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2 + \frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}_A\|^2}{N}\right) \leq 2\left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2 + \frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}\|^2}{N}. \tag{3.52}$$

Combining (3.51) and (3.52) and using the fact that there are K_a additional rows in the extended measurement matrix, we obtain that

$$\begin{aligned} 2\sum_{\boldsymbol{\phi}_i,\boldsymbol{\phi}_j\,\text{dependent}}\left(E\{A^4\} - \left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2 + \frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}_A\|^2}{N}\right) &\leq 4K_a\left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2 \\ &\quad + \frac{8\sigma_n^2K_a}{M}\cdot\frac{\|\boldsymbol{g}\|^2}{N}. \end{aligned} \tag{3.53}$$

[5] We skip the derivation of this inequality since it is relatively well known and can be found, for example, in [35, p. 4039].

Noticing that $\|\boldsymbol{g}\|^2/N = r(\hat{\boldsymbol{f}}, \boldsymbol{f})$ and $\|\boldsymbol{g}\|^2 \leq 4NB^2$, the right-hand side of the inequality (3.53) can be further upper bounded as

$$4K_a \left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2 + \frac{8\sigma_n^2 K_a}{M} \cdot \frac{\|\boldsymbol{g}\|^2}{N} \leq 16K_a B^2\, r(\hat{\boldsymbol{f}}, \boldsymbol{f}) + \frac{8\sigma_n^2 K_a}{M}\, r(\hat{\boldsymbol{f}}, \boldsymbol{f}). \tag{3.54}$$

Using the upper bound (3.54) for the second term in (3.50) and the upper bound (3.40) for the first term in (3.50), we finally find the upper bound for var $\left\{\sum_{j=1}^{K_e} U_j\right\}$ as

$$\text{var}\left\{\sum_{j=1}^{K_e} U_j\right\} \leq K_e \left(8B^2\left(1 + \frac{2K_a}{K_e}\right) + 4\sigma_n^2\left(1 + \frac{2K_a}{MK_e}\right)\right) r(\hat{\boldsymbol{f}}, \boldsymbol{f}). \tag{3.55}$$

Therefore, based on the Craig-Bernstein inequality, the probability that for a given candidate signal $\hat{\boldsymbol{f}}$ the following inequality holds

$$r(\hat{\boldsymbol{f}}, \boldsymbol{f}) - \hat{r}(\hat{\boldsymbol{f}}, \boldsymbol{f}) \leq \frac{\log(\frac{1}{\delta})}{K_e \epsilon} + \frac{\left(8B^2\left(1 + \frac{2K_a}{K_e}\right) + 4\sigma_n^2\left(1 + \frac{2K_a}{MK_e}\right)\right) r(\hat{\boldsymbol{f}}, \boldsymbol{f})\, \epsilon}{2(1-\zeta)} \tag{3.56}$$

is greater than or equal to $1 - \delta$.

Let $c(\hat{\boldsymbol{f}})$ be chosen such that the Kraft inequality $\sum_{\hat{\boldsymbol{f}} \in \mathcal{F}(B)} 2^{c(\hat{\boldsymbol{f}})} \leq 1$ is satisfied (see also [35]), and let $\delta(\hat{\boldsymbol{f}}) = 2^{-c(\hat{\boldsymbol{f}})}\, \delta$. Applying the union bound to (3.56), it can be shown that for all $\hat{\boldsymbol{f}} \in \mathcal{F}(B)$ and for all $\delta > 0$, the following inequality holds with probability of at least $1 - \delta$

$$\begin{aligned} r(\hat{\boldsymbol{f}}, \boldsymbol{f}) - \hat{r}(\hat{\boldsymbol{f}}, \boldsymbol{f}) \leq\ & \frac{c(\hat{\boldsymbol{f}}) \log 2 + \log(\frac{1}{\delta})}{K_e\, \epsilon} \\ & + \frac{\left(8B^2\left(1 + \frac{2K_a}{K_e}\right) + 4\sigma_n^2\left(1 + \frac{2K_a}{MK_e}\right)\right) r(\hat{\boldsymbol{f}}, \boldsymbol{f})\, \epsilon}{2(1-\zeta)}. \end{aligned} \tag{3.57}$$

Finally, setting $\zeta = \epsilon\, h$ and

$$a = \frac{\left(8B^2\left(1 + \frac{2K_a}{K_e}\right) + 4\sigma_n^2\left(1 + \frac{2K_a}{MK_e}\right)\right) \epsilon}{2(1-\zeta)} \tag{3.58}$$

$$\epsilon < \frac{1}{\left(4\left(1 + \frac{2K_a}{K_e}\right) + 16e\right) B^2 + 8\sqrt{B}\sigma_n + 2\sigma_n^2\left(1 + \frac{2K_a}{MK_e}\right)} \tag{3.59}$$

where $0 < \epsilon\, h \leq \zeta < 1$ as required by the Craig-Bernstein inequality, the following inequality holds with probability of at least $1 - \delta$ for all $\hat{\boldsymbol{f}} \in \mathcal{F}(B)$

$$(1-a) r(\hat{\boldsymbol{f}}, \boldsymbol{f}) \leq \hat{r}(\hat{\boldsymbol{f}}, \boldsymbol{f}) + \frac{c(\hat{\boldsymbol{f}}) \log 2 + \log(\frac{1}{\delta})}{K_e\, \epsilon}. \tag{3.60}$$

The following result on the recovery performance of the empirical risk minimization method is in order.

Theorem 3.4. *Let ϵ be chosen as*

$$\epsilon = \frac{1}{\left(60\,(B+\sigma_n)^2\right)} \tag{3.61}$$

which satisfies the inequality (3.59)*, then the signal reconstruction $\hat{\boldsymbol{f}}_{K_e}$ given by*

$$\hat{\boldsymbol{f}}_{K_e} = \arg \min_{\hat{\boldsymbol{f}} \in \mathcal{F}(B)} \left\{ \hat{r}(\hat{\boldsymbol{f}}) + \frac{c(\hat{\boldsymbol{f}}) \log 2}{\epsilon K_e} \right\} \tag{3.62}$$

satisfies the following inequality

$$E\left\{ \frac{\|\hat{\boldsymbol{f}}_{K_e} - \boldsymbol{f}\|^2}{N} \right\} \leq C_{1e} \min_{\hat{\boldsymbol{f}} \in \mathcal{F}(B)} \left\{ \frac{\|\hat{\boldsymbol{f}} - \boldsymbol{f}\|^2}{N} + \frac{c(\hat{\boldsymbol{f}}) \log 2 + 4}{\epsilon K_e} \right\} \tag{3.63}$$

where C_{1e} is the constant given as

$$C_{1e} = \frac{1+a}{1-a}, \quad a = \frac{2\left(1+\frac{2K_a}{K_e}\right)\left(\frac{B}{\sigma_n}\right)^2 + \left(1+\frac{2K_a}{MK_e}\right)}{(30-8e)\left(\frac{B}{\sigma_n}\right)^2 + \left(60-4\sqrt{2}\right)\left(\frac{B}{\sigma_n}\right) + 30} \tag{3.64}$$

with the coefficient a obtained from (3.58) *for the specific choice of ϵ in* (3.61)*.*

Proof. The proof follows the same steps as the proof of the related result for the uncorrelated case [35, p. 4039–4040] with the exception of using, in our correlated case, the above calculated values for ϵ (3.61) and a (3.64) instead of ϵ and a for the uncorrelated case. □

Example 3.4. Let one set of samples be obtained based on the measurement matrix $\boldsymbol{\Phi}_e$ with $K_a = K$, $K_e = 2K$, and $M = 8$, and let another set of samples be obtained using a $2K \times N$ measurement matrix with all i.i.d. (Bernoulli) elements. Let also ϵ be selected as given by (3.61). Then the MSE error bounds for these two cases differ from each other only by a constant factor given for the former case by C_{1e} in (3.64) and in the latter case by C_1 (see (3.12) and the row after). Considering the two limiting cases when $B/\sigma_n \to 0$ and $B/\sigma_n \to \infty$, the intervals of change for the corresponding coefficients can be obtained as $1.08 \leq C_{1e} \leq 2.88$ and $1.06 \leq C_1 \leq 1.63$, respectively.

The following result on the achievable recovery performance for a sparse or compressible signal sampled based on the extended measurement matrix $\boldsymbol{\Phi}_e$ is of importance.

Theorem 3.5. *For a sparse signal* $\boldsymbol{f} \in \mathcal{F}_s(B,S) \triangleq \{\boldsymbol{f} : \|\boldsymbol{f}\|^2 \leq NB^2, \|\boldsymbol{f}\|_{l_0} \leq S\}$ *and corresponding reconstructed signal* $\hat{\boldsymbol{f}}_{K_e}$ *obtained according to* (3.62)*, there exists a constant* $C'_{2e} = C'_{2e}(B,\sigma_n) > 0$*, such that*

$$\sup_{\boldsymbol{f}\in\mathcal{F}_s(B,S)} E\left\{\frac{\|\hat{\boldsymbol{f}}_{K_e} - \boldsymbol{f}\|^2}{N}\right\} \leq C_{1e}C'_{2e}\left(\frac{K_e}{S\log N}\right)^{-1}. \tag{3.65}$$

Similarly, for a compressible signal $\boldsymbol{f} \in \mathcal{F}_c(B,\alpha,C_A) \triangleq \{\boldsymbol{f} : \|\boldsymbol{f}\|^2 \leq NB^2, \|\boldsymbol{f}^{(m)} - \boldsymbol{f}\|^2 \leq NC_A m^{-2\alpha}\}$ *and corresponding reconstructed signal* $\hat{\boldsymbol{f}}_{K_e}$ *obtained according to* (3.62)*, there exists a constant* $C_{2e} = C_{2e}(B,\sigma_n,C_A) > 0$*, such that*

$$\sup_{\boldsymbol{f}\in\mathcal{F}_c(B,\alpha,C_A)} E\left\{\frac{\|\hat{\boldsymbol{f}}_{K_e} - \boldsymbol{f}\|^2}{N}\right\} \leq C_{1e}C_{2e}\left(\frac{K_e}{\log N}\right)^{-2\alpha/(2\alpha+1)}. \tag{3.66}$$

Proof. The proof follows the same steps as the proofs of the related results for the uncorrelated case [35, p. 4040–4041] with the exception of using, in our correlated case, the above calculated values for ϵ in (3.61) and a in (3.64) instead of ϵ and a for the uncorrelated case. □

Example 3.5. Let one set of samples be obtained based on the extended measurement matrix $\boldsymbol{\Phi}_e$ with $K_a = K$, $K_e = 2K$, and $M = 8$ and let another set of samples be obtained using the $K \times N$ measurement matrix with all i.i.d. (Bernoulli) elements. The error bounds corresponding to the case of K uncorrelated samples and our case of K_e correlated samples are (3.14) and (3.65), respectively. The comparison between these two error bounds boils down in this example to comparing $2C_1C'_2$ and $C_{1e}C'_{2e}$. Assuming the same ϵ as (3.61) for both methods, $C'_{2e} = C'_2$. Fig. 3.3 compares C_{1e} and $2C_1$ versus the signal-to-noise ratio (SNR) B^2/σ_n^2. Since $C_{1e} < 2C_1$ for all values of SNR, the quality of the signal recovery, i.e., the corresponding MSE, for the case of $2K \times N$ extended measurement matrix is expected to be better than the quality of the signal recovery for the case of $K \times N$ measurement matrix of all i.i.d. entries.

The above results can be easily generalized for the case when $K_a > K$. Indeed, we only need to recalculate $\text{var}\left\{\sum_{j=1}^{K_e} U_j\right\}$ for $K_a > 2K$. The only difference with the previous case of $K_a \leq K$ is the increased number of pairs of dependent rows in the extended measurement matrix $\boldsymbol{\Phi}_e$, which has a larger size now. The latter affects only the second term in (3.50). In particular, every row in $\boldsymbol{\Phi}^{\mathcal{P}^{(1)}}$ depends on M rows of the original measurement matrix $\boldsymbol{\Phi}$. Moreover, the term $\sum_{i=1}^{2K-1}\sum_{j=i+1}^{2K} E\{U_iU_j\}$

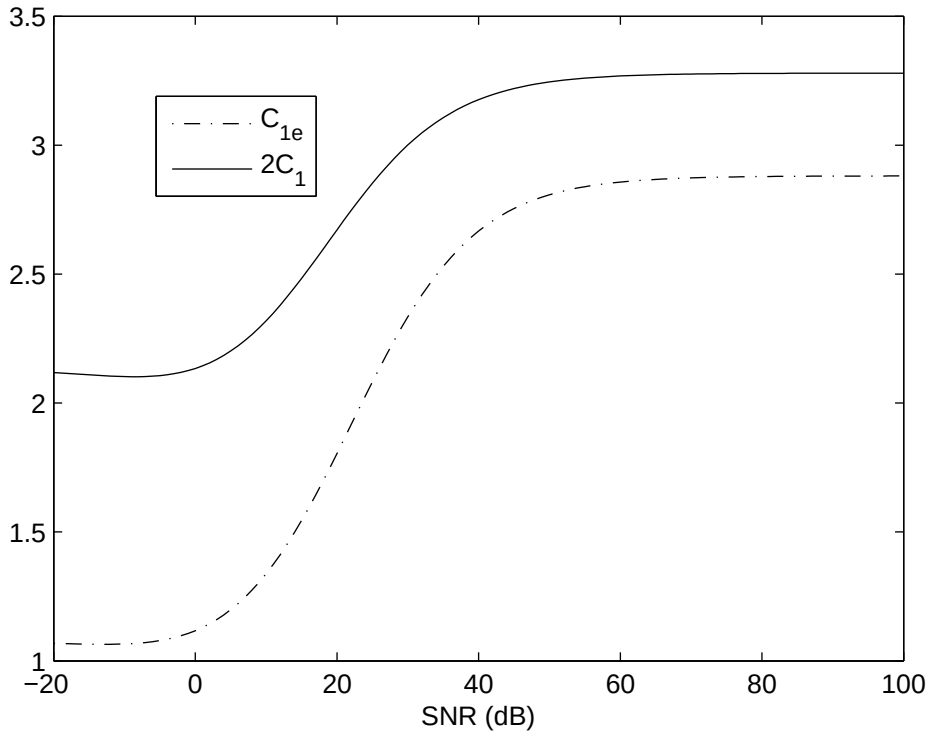


Figure 3.3: Example 3.5: Coefficients C_{1e} and $2C_1$ versus SNR. Since $C_{1e} < 2C_1$ for all values of SNR, one can conclude that the MSE corresponding to the empirical risk minimization-based recovery for the proposed segmented CS-based AIC must be lower than that for the conventional AIC.

over all these M rows is bounded as in (3.52). Then considering all KM pairs of dependent rows from $\mathbf{\Phi}$ and $\mathbf{\Phi}^{\mathcal{P}^{(1)}}$, we have

$$2 \sum_{\boldsymbol{\phi}_i, \boldsymbol{\phi}_j \text{dependent}} \left(E\{A^4\} - \left(\frac{\|\boldsymbol{g}_A\|^2}{N} \right)^2 + \frac{4\sigma_n^2}{M} \cdot \frac{\|\boldsymbol{g}_A\|^2}{N} \right) \leq 4K \left(\frac{\|\boldsymbol{g}\|^2}{N} \right)^2 + \frac{8\sigma_n^2 K}{M} \cdot \frac{\|\boldsymbol{g}\|^2}{N}. \quad (3.67)$$

Similarly, every row of $\mathbf{\Phi}^{\mathcal{P}^{(2)}}$ depends on M rows of $\mathbf{\Phi}^{\mathcal{P}^{(1)}}$ and M rows of $\mathbf{\Phi}$. Considering all these $2KM$ pairs of dependent rows, we have

$$2 \sum_{\boldsymbol{\phi}_i, \boldsymbol{\phi}_j \text{dependent}} \left(E\{A^4\} - \left(\frac{\|\boldsymbol{g}_A\|^2}{N} \right)^2 + \frac{4\sigma_n^2}{M} \cdot \frac{\|\boldsymbol{g}_A\|^2}{N} \right) \leq 4(2K) \left(\frac{\|\boldsymbol{g}\|^2}{N} \right)^2 + \frac{8\sigma_n^2 (2K)}{M} \cdot \frac{\|\boldsymbol{g}\|^2}{N}. \quad (3.68)$$

Finally, the number of rows in the last matrix $(\mathbf{\Phi}_e)_{n_p}$ is K_{n_p} (see (3.32) and (3.33)). Every row of $(\mathbf{\Phi}_e)_{n_p}$ depends on M rows of each of the previous $n_p - 1$ matrices $\mathbf{\Phi}^{\mathcal{P}^{(i)}}$, $i = 1, \ldots, n_p - 1$. Considering all $(n_p - 1)K_{n_p}M$ pairs of dependent

rows, we have

$$2\sum_{\boldsymbol{\phi}_i,\boldsymbol{\phi}_j\text{dependent}}\left(E\{A^4\}-\left(\frac{\|\boldsymbol{g}_A\|^2}{N}\right)^2+\frac{4\sigma_n^2}{M}\cdot\frac{\|\boldsymbol{g}_A\|^2}{N}\right)\leq 4(n_p-1)K_{n_p}\left(\frac{\|\boldsymbol{g}\|^2}{N}\right)^2 + \frac{8\sigma_n^2(n_p-1)K_{n_p}}{M}\cdot\frac{\|\boldsymbol{g}\|^2}{N}. \tag{3.69}$$

Using the equation (3.41) and the inequalities (3.67)–(3.69), we can find the following bound

$$\text{var}\left\{\sum_{j=1}^{K_e}U_j\right\}\leq K_e\left(8B^2\left(1+\frac{D}{K_e}\right)+4\sigma_n^2\left(1+\frac{D}{MK_e}\right)\right)r(\hat{\boldsymbol{f}},\boldsymbol{f}) \tag{3.70}$$

where $D=2K\sum_{i=1}^{n_p-2}i+2K_{n_p}(n_p-1)$. Note that in the case when $K_e=n_pK$, we have $D/K_e=n_p-1$.

Therefore, it can be shown for the general extended matrix (3.27) that the inequality (3.60) holds for the following values of a and ϵ:

$$a=\frac{\left(8B^2\left(1+\frac{D}{K_e}\right)+4\sigma_n^2\left(1+\frac{D}{MK_e}\right)\right)\epsilon}{2(1-\zeta)} \tag{3.71}$$

$$\epsilon<\frac{1}{\left(4\left(1+\frac{D}{K_e}\right)+16e\right)B^2+8\sqrt{B}\sigma_n+2\sigma_n^2\left(1+\frac{D}{MK_e}\right)}. \tag{3.72}$$

Moreover, the theorems similar to Theorems 3.4 and 3.5 follow straightforwardly with the corrections to a and ϵ which are given now by (3.71) and (3.72), respectively.

We finally make some remarks on *non-RIP* conditions for l_1-norm minimization-based recovery. Since the extended measurement matrix of the proposed segmented CS method satisfies the RIP, the results of [52] on recoverability and stability of the l_1-norm minimization straightforwardly apply. A different non-RIP-based approach for studying the recoverability and stability of the l_1-norm minimization, which uses some properties of the null space of the measurement matrix, is used in [56]. Then the non-RIP sufficient condition for recoverability of a sparse signal from its noiseless compressed samples with the algorithm (3.8) is [56]

$$\sqrt{S}<\min\left\{0.5\frac{\|\boldsymbol{v}\|_{l_1}}{\|\boldsymbol{v}\|_{l_2}}:\ \boldsymbol{v}\in\{\mathcal{N}(\boldsymbol{\Phi})\setminus\{0\}\}\right\} \tag{3.73}$$

where $\mathcal{N}(\boldsymbol{\Phi})$ denotes the null space of the measurement matrix $\boldsymbol{\Phi}$.

Let us show that the condition (3.73) is also satisfied for the extended measurement matrix $\boldsymbol{\Phi}_e$. Let $\boldsymbol{d}$ be any vector in the null space of $\boldsymbol{\Phi}_e$, i.e., $\boldsymbol{d}\in\mathcal{N}(\boldsymbol{\Phi}_e)$.

Therefore, $[\mathbf{\Phi}_e]_i \boldsymbol{d} = 0,\ i = 1, \ldots, K_e$ where $[\mathbf{\Phi}_e]_i$ is the ith $1 \times N$ row-vector of $\mathbf{\Phi}_e$. Since the first K rows of $\mathbf{\Phi}_e$ are exactly the same as the K rows of $\mathbf{\Phi}$, we have $[\mathbf{\Phi}]_i \boldsymbol{d} = 0,\ i = 1, \ldots, K$. Hence, $\boldsymbol{d} \in \mathcal{N}(\mathbf{\Phi})$ and we can conclude that $\mathcal{N}(\mathbf{\Phi}_e) \subset \mathcal{N}(\mathbf{\Phi})$. Due to this property, we have $\min\{0.5\|\boldsymbol{v}\|_{l_1}/\|\boldsymbol{v}\|_{l_2} :\ \boldsymbol{v} \in \mathcal{N}(\mathbf{\Phi})\} \leq \min\{0.5\|\boldsymbol{v}\|_{l_1}/\|\boldsymbol{v}\|_{l_2} :\ \boldsymbol{v} \in \mathcal{N}(\mathbf{\Phi}_e)\}$. Therefore, if the original measurement matrix $\mathbf{\Phi}$ satisfies (3.73), so does the extended measurement matrix $\mathbf{\Phi}_e$, and the signal is recoverable from the samples taken by $\mathbf{\Phi}_e$.

Moreover, the necessary and sufficient condition for all signals with $\|\boldsymbol{x}\|_{l_0} < S$ to be recoverable from noiseless compressed samples using the l_1-norm minimization (3.8) is that [56]

$$\|\boldsymbol{v}\|_{l_1} > 2\|\boldsymbol{v}_{\mathcal{T}}\|_{l_1},\ \forall \boldsymbol{v} \in \{\mathcal{N}(\mathbf{\Phi}) \setminus \{0\}\} \tag{3.74}$$

where $\mathcal{T}$ is the set of indices corresponding to the nonzero coefficients of $\boldsymbol{x}$. It is easy to see that since $\mathcal{N}(\mathbf{\Phi}_e) \subset \mathcal{N}(\mathbf{\Phi})$, the condition (3.74) also holds for the extended measurement matrix if the original measurement matrix satisfies it.

3.5 Simulation results

In our simulations, three different measurement matrices (sampling schemes) are used: (i) the $K \times N_c$ measurement matrix $\mathbf{\Phi}$ with i.i.d. entries referred to as the original measurement matrix; (ii) the extended $K_e \times N_c$ measurement matrix $\mathbf{\Phi}_e$ obtained using the proposed segmented CS method and referred to as the extended measurement matrix; and (iii) the $K_e \times N_c$ measurement matrix with all i.i.d entries referred to as the enlarged measurement matrix. This last measurement matrix corresponds to the sampling scheme with K_e independent BMIs in the AIC in Fig. 3.1. The number of segments in the proposed segmented CS method M is set to 8.

The noisy case corresponding to the model (3.6) is always considered. Then in order to make sure that the measurement noise for additional samples obtained based on the proposed extended measurement matrix is correlated with the measurement noise of the original samples, the $K \times M$ matrix of noisy sub-samples with the noise variance σ_n^2/M is first generated. Then the permutations are applied to this matrix and the sub-samples along each row of the original and permuted matrices are added up together to build noisy samples.

The recovery performance for three aforementioned sampling schemes is measured using the MSE between the recovered and original signals. In all examples,

MSE values are computed based on 1000 independent simulation runs for all sampling schemes tested.

3.5.1 Simulation example 1: l_1-norm minimization-based recovery for a time sparse signal

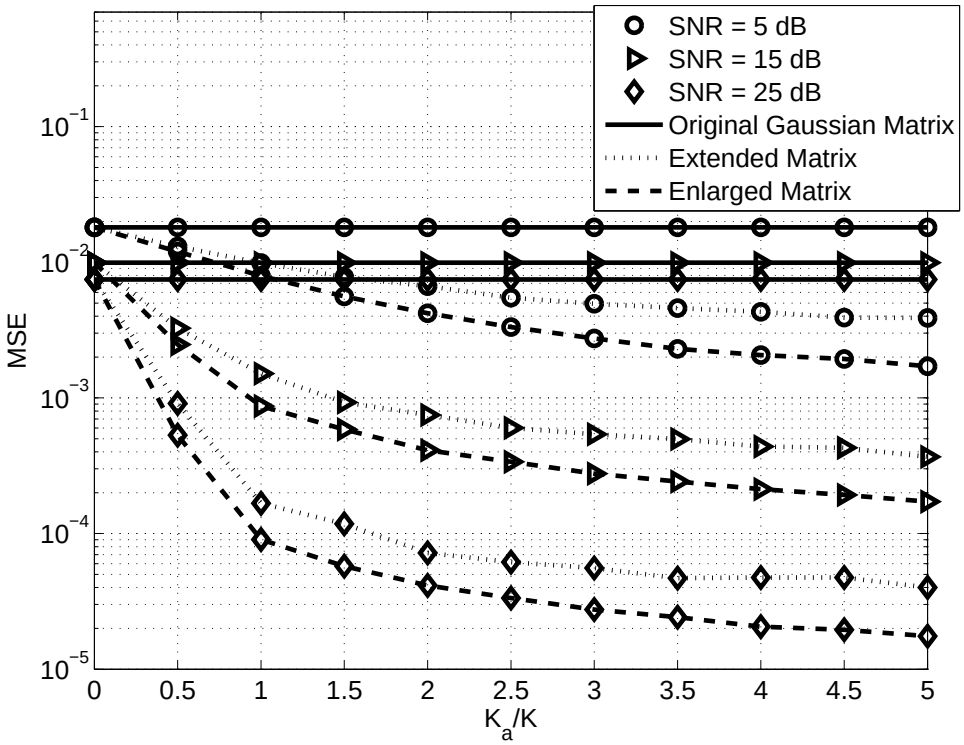
In our first example, the signal is assumed to be sparse in the time domain. Particularly, let $f(t)$ be the continuous-time signal as in (3.1) with $N = 128$ basis functions $\{\psi_n(t)\}_{n=1}^{N}$ of the type

$$\psi_n(t) = \begin{cases} \frac{N}{T}, & t \in [(n-1)T/N,\, nT/N] \\ 0, & otherwise \end{cases}, \quad n = 1, \ldots, N. \tag{3.75}$$

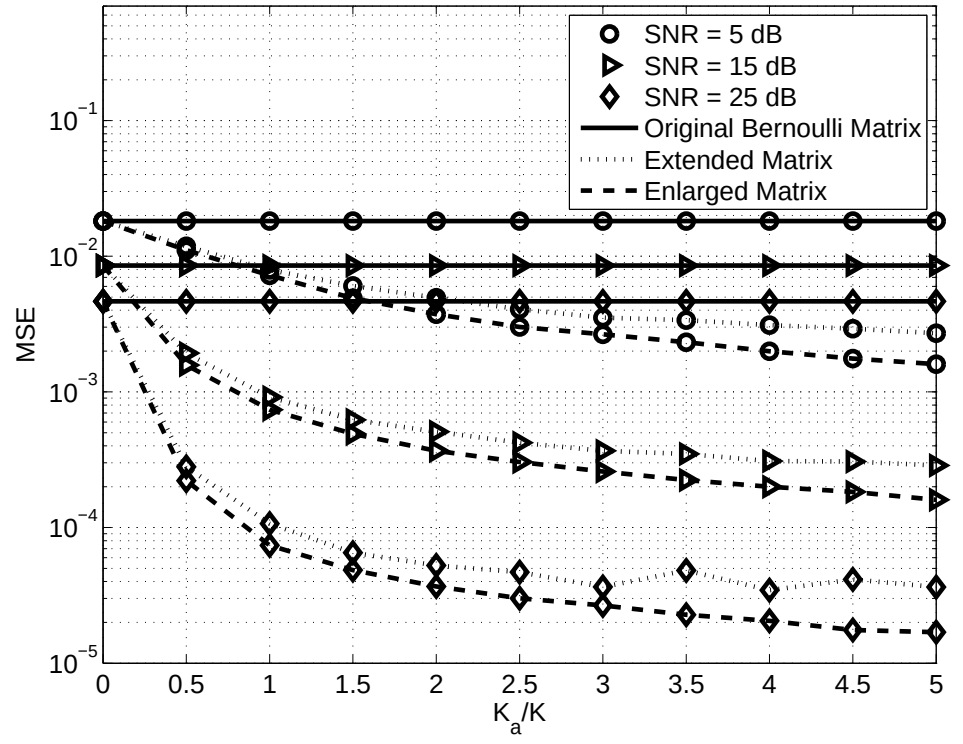
These basis functions form the sparsity basis $\boldsymbol{\Psi}(t)$. Choosing N_c to be equal to N, we obtain based on (3.5) that $\boldsymbol{\Psi} = \boldsymbol{I}$. Over one time period T only 3 projections of the signal onto the sparsity basis are nonzero and are set to $+1$ or -1 with equal probabilities.

The l_1-norm minimization algorithm (3.9) is used to recover the signal sampled using the three aforementioned sampling schemes. Since $\boldsymbol{\Psi} = \boldsymbol{I}$ for the considered time sparse signal, then $\boldsymbol{\Phi}' = \boldsymbol{\Phi}$ in (3.9). The number of BMIs in the sampling device is $K = 16$, while γ in (3.9), which is the bound on the root square of the noise energy, is set to $\sqrt{K'}\sigma_n$. Here $K' = K$ for the sampling scheme based on the original measurement matrix, while $K' = K_e$ in the other two schemes. The entries of the original and enlarged measurement matrices are generated as i.i.d. Gaussian or i.i.d. Bernoulli distributed random variables with zero mean and variance $1/N$. This corresponds to the case of sampling waveforms with chip duration T/N and i.i.d. Gaussian or i.i.d. Bernoulli distributed chip amplitudes, respectively. The SNR is defined as $\|\boldsymbol{\Phi f}\|_{l_2}^2/\|\boldsymbol{n}\|_{l_2}^2$. Approximating $\|\boldsymbol{\Phi f}\|_{l_2}^2$ by $(K'/N)\|\boldsymbol{f}\|_{l_2}^2$, which is valid because of (3.7), the corresponding noise variance σ_n^2 can be calculated when SNR is given and vice versa. For example, the approximate SNR in dBs can be calculated as $10\log_{10}(3/N\sigma_n^2)$.

Figs. 3.4(a) and 3.4(b) show the MSEs corresponding to all three aforementioned measurement matrices versus the ratio of the number of additional samples to the number of original samples K_a/K for the Gaussian and Bernoulli cases, respectively. The results are shown for three different SNR values of 5, 15 and 25 dB. It can be seen from the figures that better recovery quality is achieved by using the extended measurement matrix as compared to the original measurement matrix. The corresponding MSE curves in both Figs. 3.4(a) and 3.4(b) are similar to each other



(a) Measurement matrix with Gaussian distributed entries



(b) Measurement matrix with Bernoulli distributed entries

Figure 3.4: Recovery of the time sparse signal based on the l_1-norm minimization algorithm: MSEs versus K_a/K.

which confirms the fact that both Gaussian and Bernoulli measurement matrices are good candidates, although Bernoulli is practically preferable. As expected, the recovery performance in the case of the extended measurement matrix is slightly worse than that in the case of the enlarged measurement matrix. This difference, however, is small as compared to the performance improvement provided by the extended measurement matrix over the original measurement matrix. Note also that in the case of the enlarged measurement matrix, the AIC in Fig. 3.1 consists of K_e BMIs, while only K BMIs are required in the case of the extended measurement matrix. For example, the number of such BMIs halves for the proposed segmented AIC if $K_a/K = 1$. Additionally, it can be seen that the rate of MSE improvement decreases as the number of collected samples increases. The latter can be observed for both the extended and enlarged measurement matrices and for all three values of SNR.

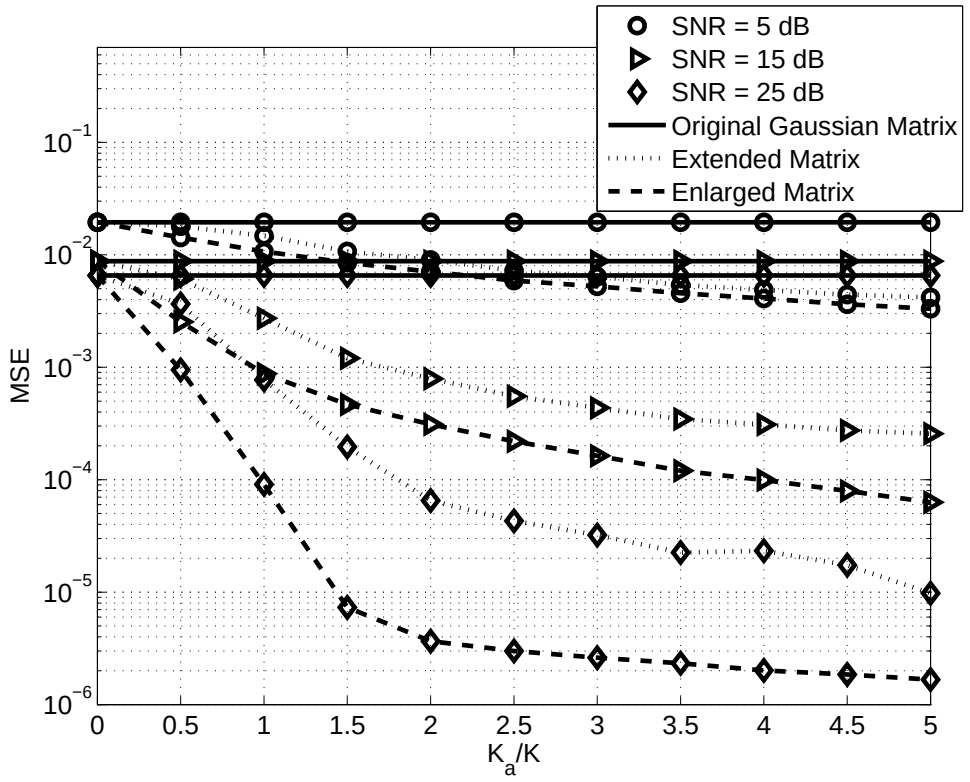
3.5.2 Simulation example 2: Time sparse signal with empirical risk minimization-based recovery

In our second simulation example, the empirical risk minimization method is used to recover the same time sparse signal as in our first simulation example. The signal is sampled using the three sampling schemes tested with $K = 24$. The minimization problem (3.11) is solved to obtain a candidate reconstruction $\hat{\boldsymbol{f}}_{K'}$ of the original sparse signal $\boldsymbol{f}$. Considering $\hat{\boldsymbol{f}}_{K'} = \boldsymbol{\Psi}^T \hat{\boldsymbol{x}}_{K'}$, the problem (3.11) can be rewritten in terms of $\hat{\boldsymbol{x}}_{K'}$ as

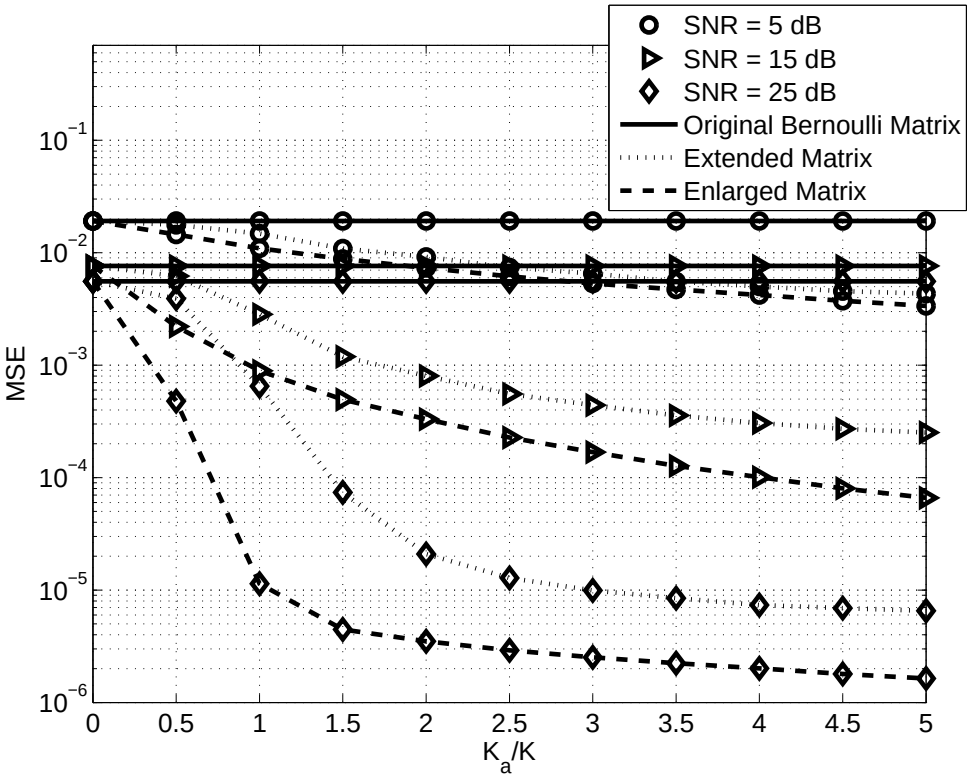
$$\begin{aligned} \hat{\boldsymbol{x}}_{K'} &= \arg\min_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \hat{r}(\boldsymbol{\Psi}^T \hat{\boldsymbol{x}}) + \frac{c(\hat{\boldsymbol{x}}) \log 2}{\epsilon K'} \right\} \\ &= \arg\min_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \|(y) - \boldsymbol{\Phi}\boldsymbol{\Psi}^T \hat{\boldsymbol{x}}\|_{l_2}^2 + \frac{2 \log 2 \log N}{\epsilon} \|\hat{\boldsymbol{x}}\|_{l_0} \right\} \end{aligned} \tag{3.76}$$

and solved using the iterative bound optimization procedure [35]. This procedure uses the threshold $\sqrt{2 \log 2 \log N / \lambda \epsilon}$ where λ is the largest eigenvalue of the matrix $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$. In our simulations, this threshold is set to 0.035 for the case of the extended measurement matrix and 0.05 for the cases of the original and enlarged measurement matrices. These threshold values are optimized as recommended in [35]. The stopping criterion for the iterative bound optimization procedure is $\|\hat{\boldsymbol{x}}^{(i+1)} - \hat{\boldsymbol{x}}^{(i)}\|_{l_\infty} \leq \theta$ where $\|.\|_{l_\infty}$ is the l_∞-norm and $\hat{\boldsymbol{x}}^{(i)}$ denotes the value of $\hat{\boldsymbol{x}}$ obtained in the ith iteration. The value $\theta = 0.001$ is selected.

Figs. 3.5(a) and 3.5(b) show the MSEs for all three measurement matrices tested



(a) Measurement matrix with Gaussian distributed entries



(b) Measurement matrix with Bernoulli distributed entries

Figure 3.5: Recovery of the time sparse signal based on the empirical risk minimization method: MSEs versus K_a/K.

versus the ratio K_a/K for the Gaussian and Bernoulli cases, respectively. The results are shown for three different SNR values of 5, 15 and 25 dB. The same conclusions as in the first example can be drawn in this example.

3.5.3 Simulation example 3: l_1-norm minimization-based recovery for an OFDM signal

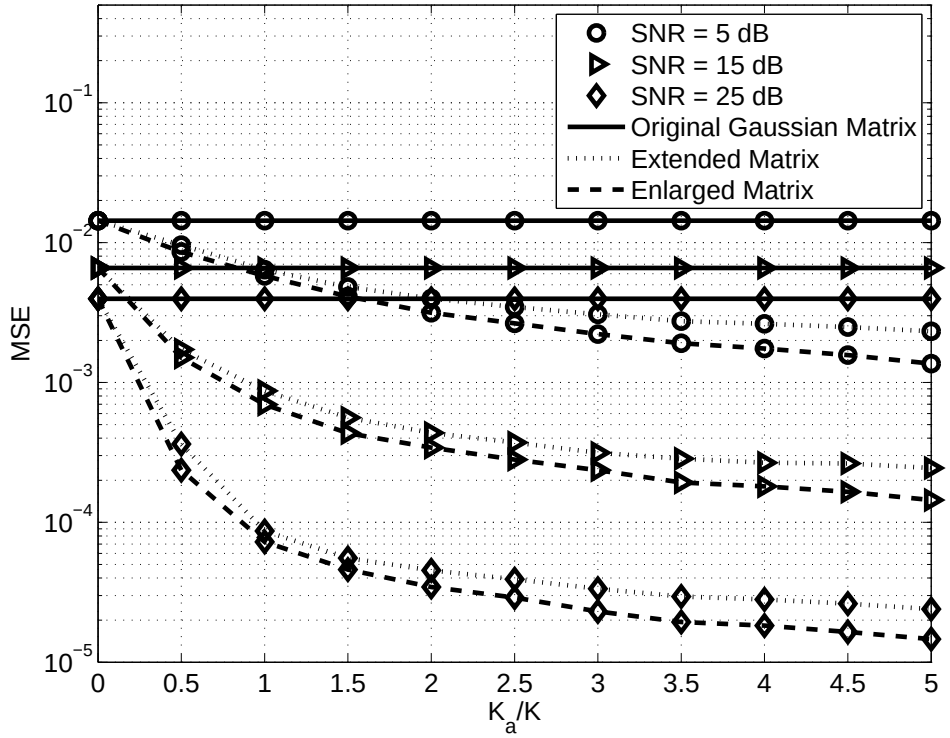
In our third example, we consider an orthogonal frequency-division-multiplexing (OFDM) signal with 3 nonzero subcarriers out of 128 available frequency bins. Nonzero subcarriers are modulated with quadrature phase-shift keying (QPSK) symbols. The $N = 128$ basis functions $\{\psi_n(t)\}_{n=1}^{N}$ are

$$\psi_n(t) = \cos\left((n-1)\frac{2\pi}{T}t\right) \quad t \in [0\ ,\ T], \quad n = 1, \ldots, N. \tag{3.77}$$

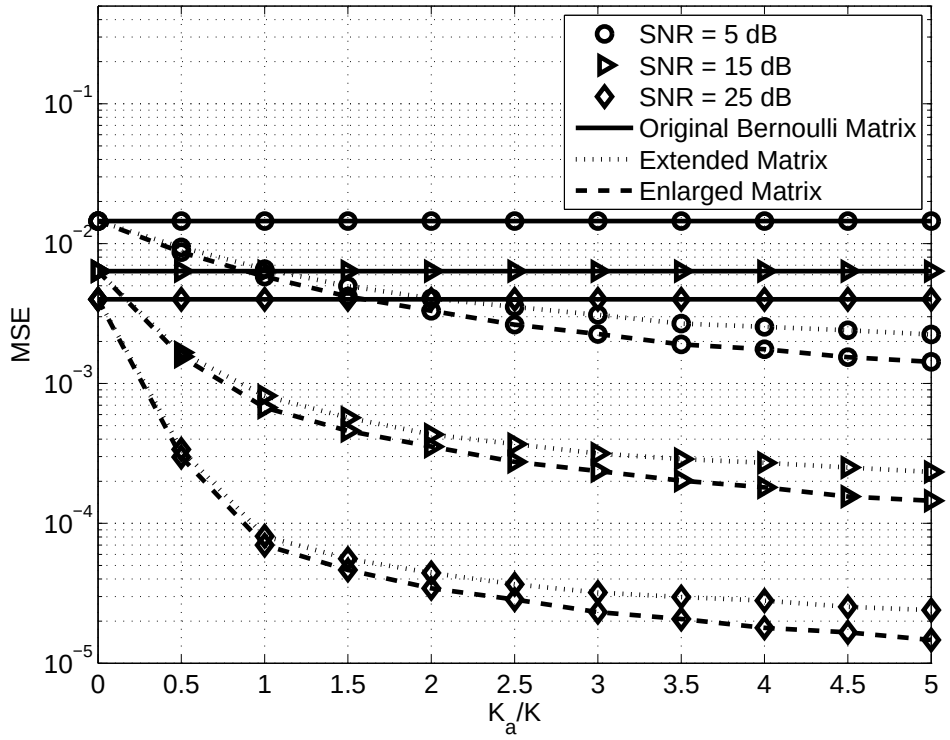
The number of chips per symbol duration in the sampling waveform is set to $N_c = 256$. It is because we need to ensure that the rows of the 128×256 sparsity matrix $\mathbf{\Psi}$, which is calculated according to (3.5), are approximately orthogonal. Then the SNR can be defined as $\|\mathbf{\Phi}'\boldsymbol{x}\|_{l_2}^2/\|\boldsymbol{n}\|_{l_2}^2$. Moreover, one can approximate $\|\mathbf{\Phi}'\boldsymbol{x}\|_{l_2}^2 = \|\mathbf{\Phi}\mathbf{\Psi}\boldsymbol{x}\|_{l_2}^2$ by $\psi_{ave}(K'/N)\|\boldsymbol{x}\|_{l_2}^2$, where ψ_{ave} is the average norm of the N rows of the sparsity matrix $\mathbf{\Psi}$ and $K' = K$ for the sampling scheme based on the original measurement matrix, while $K' = K_e$ for the other two schemes. Considering that the nonzero subcarriers are modulated with unit norm QPSK symbols, the approximate SNR in dBs can be calculated as $10\log_{10}(3\psi_{ave}/N\sigma_n^2)$. The l_1-norm minimization-based recovery method is used for signal recovery based on the compressed samples obtained using the three sampling schemes tested.

Figs. 3.6(a) and 3.6(b) show the MSEs for all three measurement matrices tested versus the ratio K_a/K for the Gaussian and Bernoulli cases, respectively. Comparing the results in Fig. 3.6 and Fig. 3.4, one can deduce the universality of the Gaussian and Bernoulli measurement matrices, which means that we are able to recover the signal using the measurements collected with these measurement matrices regardless of the sparsity basis. As in the previous two simulation examples, the proposed segmented CS scheme significantly outperforms the original sampling scheme with the same number of BMIs, while slightly deteriorates in performance compared to the sampling scheme with enlarged number of BMIs.

It is also worth mentioning that the MSE of the recovered signal depends on the ratio between the sparsity and the extended number of samples (see for example (3.63)). Moreover, the RIP sets a bound for the number of samples required for



(a) Measurement matrix with Gaussian distributed entries



(b) Measurement matrix with Bernoulli distributed entries

Figure 3.6: Recovery of the sparse OFDM signal based on the l_1-norm minimization algorithm: MSEs versus K_a/K.

successful recovery given the sparsity level of the signal. Thus, if the signal is not sparse enough and the number of collected samples is low, a recovery algorithm can fail to recover the signal from such small number of samples. If the number of samples is sufficient to ensure successful recovery, but the ratio between the sparsity and the number of samples is high, the MSE can be still high. By using the technique proposed in this chapter for extending the number of samples, this situation can be improved as we show in our next example.

3.5.4 Simulation example 4: The number of BMIs in the conventional AIC is insufficient for successful recovery

Our last simulation example considers the case when the number of original compressed samples, that is, the number of BMIs in the conventional AIC, is insufficient for successful recovery. The time sparse signal described in our first simulation example is assumed. The number of basis functions is $N = 128$, however, the number of nonzero projections, i.e., the signal sparsity level is $S = 5$. The number of BMIs in the conventional AIC is $K = 9$ and the number of segments in the proposed segmented AIC is $M = 8$. Since generally speaking four times as many samples are needed as the sparsity level of the signal to guarantee exact recovery in noiseless case [1], the number of samples that can be collected by the conventional AIC with $K = 9$ BMIs is insufficient for exact recovery even in the noise free case. Thus, the conventional AIC is not applicable and only the sampling schemes based on the extended and enlarged measurement matrices are compared to each other in terms of the percentage that the positions of the nonzero values of the time sparse signal are correctly identified. The number of samples can be increased to at most $K^2 = 81$ if the proposed segmented AIC is used. The MSEs averaged over all cases of successful recovery are also reported. Two different cases of (a) no measurement noise and (b) SNR=25 dB are considered. The simulation results are gathered in Table 3.1.

The results in Table 3.1 show that although the AIC with $K = 9$ (the column $K_a = 0$ in the table) branches cannot successfully recover the positions of nonzero entries of the signal, the segmented AIC is able to find those positions and the success rate increases as K_a increases. The success rate of the AIC with larger number of BMIs is higher as expected. For both schemes the lower success rates can be observed in the noisy case as compared to the noiseless case.

Table 3.1: Percentage that the positions of the nonzero signal values are correctly identified.

K_a			0	9	18	27	36	45
Noiseless case	AIC with K_e branches	Percentage	0	22.45	90.5	100	100	100
		MSE	N/A	0	0	0	0	0
	Segmented AIC	Percentage	0	10.2	58.35	83.10	89.7	90.5
		MSE	N/A	0	0	0	0	0
SNR=25 dB	AIC with K_e branches	Percentage	0	11	71.45	98	100	100
		MSE	N/A	5.48×10^{-4}	4.91×10^{-4}	2.78×10^{-4}	1.53×10^{-4}	1.04×10^{-4}
	Segmented AIC	Percentage	0	4.85	34.30	57.20	70.5	73.65
		MSE	N/A	6.06×10^{-4}	5.88×10^{-4}	5.78×10^{-4}	4.82×10^{-4}	3.98×10^{-4}

3.6 Chapter summary

A new segmented CS method for AIC has been proposed. According to this method, an analog signal measured by K parallel BMIs, each characterized by a specific random sampling waveform, is first segmented in time into M segments so that a $K \times M$ matrix of sub-samples is obtained. Then the sub-samples collected on different segments and different BMIs are reused so that a larger number of samples (at most K^2) than the number of BMIs is collected. Such samples are correlated to each other over at most one segment and the technique is shown to be equivalent to extending the measurement matrix consisting of the BMI sampling waveforms by adding new rows without actually increasing the number of BMIs. Such extended measurement matrix satisfies the RIP with overwhelming probability if the original measurement matrix of BMI sampling waveforms satisfies it. Due to the inherent structure of the proposed segmented CS method, the complexity of the sampling device is slightly increased, while the signal recovery performance is shown to be significantly improved. Specifically, we have proved that the performance of the signal recovery based on the empirical risk minimization improves when the segmented AIC is used for sampling instead of the conventional AIC with the same number of BMIs. Remarkably, if the number of BMIs is insufficient in the conventional AIC to guarantee successful recovery, the proposed segmented AIC supplies the recovery algorithm with additional samples so that successful recovery becomes possible. At the same time, the complexity increase is only due to the $M \leq K$ times higher sampling rate and the necessity to solve a larger size optimization problem at the recovery stage, while the number of BMIs remains the same at the sampling stage. The validity, effectiveness, and superiority of the proposed segmented AIC over the conventional AIC is also justified based on our simulation results.

Chapter 4

Sparse Channel Estimation

The standard recursive parameter estimation algorithms do not assume any information about the specific structure of the channel being estimated. However, being aware of the channel structure one can modify the standard algorithms in order to have a better estimate of the channel. In this chapter, we are concerned with a class of channels where the CIR is sparse.

Sparsity aware modifications of the LMS algorithm are abundant in the signal processing literature [57], [58]. The methods introduced in [57], [58] add a penalty term to the standard LMS error function which is designed in a way to force the solution to be sparse. A penalty in the form of the l_0-pseudo-norm of the CIR is used in [58], while [57] uses the l_1-norm. A performance analysis of the l_0-pseudo-norm constraint LMS algorithm of [58] is given in [59]. In [60], [61], variations of the algorithms in [57] are introduced. In [60], the filter coefficients are updated in a transform domain which leads to faster convergence for non-white inputs. The authors of [62] introduce a scheme that employs two sequential adaptive filters for communication line or network echo cancellers. The method exploits the sparseness of the CIR and uses two sequential LMS type structures which are both shorter than the largest delay of the channel. A family of the so called natural gradient estimation algorithms is also studied in [63]. It is shown that the class of sparse LMS algorithms presented has faster convergence rate.

Other channel estimation algorithms have also been modified to either better adapt to a sparse channel or achieve the same performance as the corresponding standard algorithms with lower complexity. Time and norm-weighted LASSO with weights obtained from RLS algorithm has been presented in [64]. A greedy RLS algorithm designed for finding sparse solutions to linear systems has been presented

in [65], and it has been demonstrated that it has better performance than the standard RLS algorithm for estimating sparse time-varying FIR channels. A CS-based Kalman filter has been developed in [66] for estimating signals with time varying sparsity pattern.

In this chapter, we first derive the reweighted l_1-norm penalized LMS algorithm which is based on modifying the LMS cost function by adding the l_1-norm penalty term and also introducing a reweighting of the CIR coefficients. Then the main contribution follows that is the in depth study of the convergence and excess MSE analysis of the reweighted l_1-norm penalized LMS algorithm. Our simulation results show that the proposed algorithm outperforms the standard LMS as well as the penalized sparsity-aware LMS algorithms of [57] and approve our theoretical studies.

The rest of the chapter is organized as follows. Sections 4.1 and 4.2 review the standard LMS algorithm and some sparsity-aware modifications of it, respectively. In Section 4.3, the reweighted l_1-norm penalized LMS algorithm is introduced. An analytical study of the convergence of the reweighted l_1-norm penalized LMS algorithm as well as its excess MSE is given in Section 4.4. The l_p-pseudo-norm penalized LMS algorithm is introduced in Section 4.5. Decimated LMS algorithms for estimation of a frequensy sparse CIR are introduced in Section 4.7. Simulation results comparing the performance of different sparsity-aware LMS algorithms are given in Section 4.8. Section 4.9 concludes the chapter.

4.1 Standard LMS algorithm

Let us reconsider the block diagram of the communication system shown in Fig. 2.13. The system equations in (2.17) derived form Fig. 2.13 are rewritten here.

$$\begin{aligned} d_k &= \boldsymbol{w}^T \boldsymbol{x}_k + n_k \\ e_k &= d_k - \boldsymbol{w}_k^T \boldsymbol{x}_k. \end{aligned} \tag{4.1}$$

Standard LMS minimizes the cost function $L_k = (1/2)e_k^2$ with a gradient descent algorithm and its update equation is derived as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \mu \frac{\partial L_k}{\partial \boldsymbol{w}_k} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k \tag{4.2}$$

with μ being the step size of the algorithm.

For the purpose of convergence analysis of LMS a coefficient error vector is

usually defined as

$$\boldsymbol{v}_k = \boldsymbol{w}_k - \boldsymbol{w}. \tag{4.3}$$

The data vector $\boldsymbol{x}_k$ is assumed to be independent of the coefficient error vector $\boldsymbol{v}_k$. Using (2.17), (2.18), and (4.3) we can write

$$\begin{aligned}\boldsymbol{v}_{k+1} &= \boldsymbol{v}_k + \mu\big((\boldsymbol{w}^T - \boldsymbol{w}_k^T)\boldsymbol{x}_k + n_k\big)\boldsymbol{x}_k \\ &= \boldsymbol{v}_k - \mu\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k + \mu n_k\boldsymbol{x}_k = \boldsymbol{v}_k - \mu\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k + \mu n_k\boldsymbol{x}_k.\end{aligned} \tag{4.4}$$

Taking expectation from the both sides of (4.4) we can derive the evolution equation for $E\{\boldsymbol{v}_k\}$. Note that due to the independence assumption of the data vector and the coefficient error vector, we have $E\left\{\mu\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\right\} = \mu E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\right\}E\{\boldsymbol{v}_k\}$. Also, $E\{\mu n_k\boldsymbol{x}_k\} = \mu E\{n_k\}E\{\boldsymbol{x}_k\} = 0$, since n_k is zero mean. We can write the evolution equation as

$$E\{\boldsymbol{v}_{k+1}\} = (\boldsymbol{I} - \mu\boldsymbol{R})E\{\boldsymbol{v}_k\} \tag{4.5}$$

where $\boldsymbol{R} = E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\right\}$, and $\boldsymbol{I}$ is the identity matrix. The following result establishes the mean convergence of standard LMS algorithm.

Theorem 4.1. *If the maximal eigenvalue of the matrix $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then the mean coefficient error vector of the standard LMS algorithm $E\{\boldsymbol{v}_k\}$ converges to zero as $k \to \infty$.*

Proof. Let $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ be the eigenvalue decomposition of matrix $\boldsymbol{R}$. Defining $\boldsymbol{c}_k \triangleq \boldsymbol{Q}^T\boldsymbol{v}_k$, equation (4.5) becomes

$$E\{\boldsymbol{c}_{k+1}\} = (\boldsymbol{I} - \mu\boldsymbol{\Lambda})E\{\boldsymbol{c}_k\}. \tag{4.6}$$

From the above equation it can be seen that

$$E\{\boldsymbol{c}_{k+M}\} = (\boldsymbol{I} - \mu\boldsymbol{\Lambda})^M E\{\boldsymbol{c}_k\}. \tag{4.7}$$

Since $\boldsymbol{I} - \mu\boldsymbol{\Lambda}$ is a diagonal matrix, the convergence behavior of every element of the vector $E\{\boldsymbol{c}_{k+M}\}$ can be studied separately. Let λ_i be the ith diagonal element of the matrix $\boldsymbol{\Lambda}$. From (4.7), we have

$$\big[E\{\boldsymbol{c}_{k+M}\}\big]_i = (1 - \mu\lambda_i)^M \big[E\{\boldsymbol{c}_k\}\big]_i \tag{4.8}$$

where $[\cdot]_i$ denotes the ith entry of a vector. Since the largest eigenvalue of $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then all the diagonal elements $1 - \mu\lambda_i$ are smaller than 1. Therefore every entry of $E\{\boldsymbol{c}_{k+M}\}$ tends to zero as $M \to \infty$. Since $\boldsymbol{v}_k = \boldsymbol{Q}\boldsymbol{c}_k$, the mean coefficient error vector $E\{\boldsymbol{v}_k\}$ tends to an all zero vector as $k \to \infty$. □

In order to find the excess MSE of the standard LMS algorithm we need to find an expression for $E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\}$. Using the expression in (4.4) for $\boldsymbol{v}_{k+1}$, the variable $\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T$ can be written as follows

$$\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T = \left(\boldsymbol{v}_k - \mu\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k + \mu n_k\boldsymbol{x}_k\right)\left(\boldsymbol{v}_k^T - \mu\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T + \mu n_k\boldsymbol{x}_k^T\right). \tag{4.9}$$

Expanding the right hand side of (4.9), and then taking expectation from the both sides of the equation we obtain

$$\begin{aligned} E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\} &= E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\} - \mu\left(E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} + E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right) \\ &\quad + \mu^2 E\left\{n_k^2\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} + \mu\left(E\left\{n_k\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} + E\left\{n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\right\}\right) \\ &\quad - \mu^2\left(E\left\{n_k\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} + E\left\{n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\}\right) \\ &\quad + \mu^2 E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\}. \end{aligned} \tag{4.10}$$

Note that due to the independence of the additive noise, n_k, and the data and coefficient error vectors as well as the fact that the additive noise is zero mean, we have

$$\begin{aligned} E\left\{n_k\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} &= E\left\{n_k\right\}E\left\{\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} = 0 \\ E\left\{n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\right\} &= E\left\{n_k\right\}E\left\{\boldsymbol{x}_k\boldsymbol{v}_k^T\right\} = 0 \\ E\left\{n_k\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} &= E\left\{n_k\right\}E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} = 0 \\ E\left\{n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} &= E\left\{n_k\right\}E\left\{\boldsymbol{x}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} = 0. \end{aligned} \tag{4.11}$$

Using the fact that $E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} = 2\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}\ \mathrm{tr}\left\{\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}$ for Gaussian input sequences (see, for example, equation (12) of [70] and the derivation of equation (35) in [71]) in (4.10), the following expression for $E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\}$ is derived

$$\begin{aligned} E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\} &= E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\} - \mu\left(E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right) + \mu^2\sigma_n^2\boldsymbol{R} \\ &\quad + \mu^2\left(2\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}\ \mathrm{tr}\left\{\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}\right). \end{aligned} \tag{4.12}$$

The excess MSE denoted as ξ_k is defined as $\xi_k = E\left\{\left(\boldsymbol{v}_k^T\boldsymbol{x}_k\right)^2\right\}$. Then the excess MSE, ξ_k, can be further expanded and simplified as shown below

$$\begin{aligned} \xi_k &= E\left\{\left(\boldsymbol{v}_k^T\boldsymbol{x}_k\right)^2\right\} = E\left\{\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\right\} \\ &\overset{(a)}{=} E\left\{\ \mathrm{tr}\left\{\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\right\}\right\} \overset{(b)}{=}\ \mathrm{tr}\left\{\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}. \end{aligned} \tag{4.13}$$

In (4.13), equality (a) holds since $\boldsymbol{v}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k$ is a scalar and therefore, it is equal to its trace. Also, (b) holds since $\text{tr}\left\{\boldsymbol{v}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k\right\} = \text{tr}\left\{\boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{v}_k \boldsymbol{v}_k^T\right\}$ and the two mathematical operators of matrix trace and expectation are interchangable. Let $\boldsymbol{R_v} = \lim_{k\to\infty} E\left\{\boldsymbol{v}_k \boldsymbol{v}_k^T\right\}$, and $\xi = \lim_{k\to\infty} \xi_k$. Then from (4.13) we have $\xi = \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}$. Therefore, to find ξ we need to find an expression for $\text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}$ from equation (4.12). Letting $k \to \infty$ in (4.12) we have

$$\boldsymbol{R_v} = \boldsymbol{R_v} - \mu\left(\boldsymbol{R_v}\boldsymbol{R} + \boldsymbol{R}\boldsymbol{R_v}\right) + \mu^2\sigma_n^2\boldsymbol{R} + \mu^2\left(2\boldsymbol{R}\boldsymbol{R_v}\boldsymbol{R} + \boldsymbol{R}\,\text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}\right). \tag{4.14}$$

Crossing out $\boldsymbol{R_v}$ from the both sides of the above equation and then dividing the resulting equation by μ we can write

$$\boldsymbol{R_v}\boldsymbol{R} + \boldsymbol{R}\boldsymbol{R_v} - 2\mu\boldsymbol{R}\boldsymbol{R_v}\boldsymbol{R} = \mu\boldsymbol{R}\left(\sigma_n^2 + \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}\right). \tag{4.15}$$

Breaking $2\mu\boldsymbol{R}\boldsymbol{R_v}\boldsymbol{R}$ into the sum of two identical terms and then factoring out $\boldsymbol{R}\boldsymbol{R_v}$ and $\boldsymbol{R_v}\boldsymbol{R}$ we obtain

$$\boldsymbol{R}\boldsymbol{R_v}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right) + \left(\boldsymbol{I} - \mu\boldsymbol{R}\right)\boldsymbol{R_v}\boldsymbol{R} = \mu\boldsymbol{R}\left(\sigma_n^2 + \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}\right). \tag{4.16}$$

Right multiplying the both sides of the above equation with $\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}$ and noting that $\sigma_n^2 + \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}$ is a scalar the following expression can be derived

$$\boldsymbol{R}\boldsymbol{R_v} + \left(\boldsymbol{I} - \mu\boldsymbol{R}\right)\boldsymbol{R_v}\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1} = \mu\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\left(\sigma_n^2 + \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}\right). \tag{4.17}$$

Taking the trace of both sides of the above equation we have

$$\begin{aligned}\text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\} + \text{tr}\left\{\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)\boldsymbol{R_v}\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\} \\ = \mu\left(\sigma_n^2 + \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}\right)\ \text{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\}.\end{aligned} \tag{4.18}$$

Since $\text{tr}\left\{\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)\boldsymbol{R_v}\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\}$ equals $\text{tr}\left\{\boldsymbol{R_v}\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\}$ which in turn is equal to $\text{tr}\left\{\boldsymbol{R_v}\boldsymbol{R}\right\}$, (4.18) can be simplified to the following equation

$$\text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\} + \text{tr}\left\{\boldsymbol{R_v}\boldsymbol{R}\right\} = \mu\left(\sigma_n^2 + \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}\right)\ \text{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\}. \tag{4.19}$$

Since $\text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\} = \text{tr}\left\{\boldsymbol{R_v}\boldsymbol{R}\right\}$, from (4.19) we can write

$$\text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}\left(2 - \mu\ \text{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\}\right) = \mu\sigma_n^2\ \text{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\}. \tag{4.20}$$

Having in mind that $\xi = \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\}$, an expression for ξ can be derived from (4.20). Thus, it can be shown that the excess MSE is given by

$$\xi = \text{tr}\left\{\boldsymbol{R}\boldsymbol{R_v}\right\} = \frac{\eta}{2-\eta}\sigma_n^2 \tag{4.21}$$

where $\eta \triangleq \mu\ \text{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I} - \mu\boldsymbol{R}\right)^{-1}\right\}$.

4.2 Sparsity aware LMS algorithms

The standard LMS algorithm assumes no structural information about the signal/system to be estimated. Taking any structural information into account, one should be able to modify the algorithm and benefit by lower estimation error, faster convergence, or lower algorithm complexity. In this chapter, we are interested in the case when the CIR is sparse. Several sparsity-aware modifications of the standard LMS have been introduced in the literature [57], [67].

The Zero Attracting LMS Algorithm: In order to penalize the non-sparse solutions, an additional term proportional to the l_1-norm of $\boldsymbol{w}_k$ is included in the cost function of the standard LMS algorithm in [57]. Then the new cost function becomes $L_k^{\mathrm{ZA}} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{ZA}}\|\boldsymbol{w}_k\|_{l_1}$, where γ_{ZA} is the weight associated with the penalty term. In [57], the CIR is assumed to be sparse in the time domain. Note that the cost function L_k^{ZA} is convex, and therefore, it is guaranteed that the gradient descent method converges under some condition. The corresponding algorithm is called the zero attracting LMS (ZA-LMS) and its update equation is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{ZA}}\mathrm{sgn}(\boldsymbol{w}_k) \tag{4.22}$$

where $\rho_{\mathrm{ZA}} \triangleq \mu\gamma_{\mathrm{ZA}}$ and $\mathrm{sgn}(\cdot)$ is the sign function which operates on every component of the vector separately and it is zero for $x = 0$, 1 for $x > 0$, and -1 for $x < 0$.

The Reweighted Zero Attracting LMS Algorithm: The exact measure of sparsity, that is, the l_0-pseudo-norm can also be added as the penalty term to the standard LMS's cost function. Since the complexity of using the l_0-pseudo-norm penalty is high, a logarithmic penalty that resembles the l_0-pseudo-norm is used as an alternative in [57]. The modified cost function is then

$$L_k^{\mathrm{RZA}} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{RZA}} \sum_{i=1}^{N} \log\left(1 + \frac{[\boldsymbol{w}_k]_i}{\epsilon'_{\mathrm{RZA}}}\right) \tag{4.23}$$

where γ_{RZA} and ϵ'_{RZA} are some positive numbers. Note that the same penalty term is also used, for example, in [68]. Since the logarithmic penalty in (4.23) resembles the l_0-pseudo-norm better than the l_1-norm penalty in ZA-LMS algorithm, one can expect that the corresponding algorithm, called in [57] as the reweighted zero attracting LMS (RZA-LMS) algorithm, has a better performance than the ZA-LMS algorithm. The update equation for the RZA-LMS is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{RZA}} \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{1 + \epsilon_{\mathrm{RZA}}|\boldsymbol{w}_k|} \tag{4.24}$$

where $\rho_{\text{RZA}} \triangleq \mu\gamma_{\text{RZA}}\epsilon_{\text{RZA}}$, $\epsilon_{\text{RZA}} \triangleq 1/\epsilon'_{\text{RZA}}$, and $|\cdot|$ is the component-wise absolute value operator. However, the cost function (4.23) is not convex and the convergence and consistency analysis is problematic for (4.24).

In [60], variations of the ZA-LMS and RZA-LMS algorithms, in which the filter coefficients are updated in a transform domain leading to faster convergence with non-white system inputs, have been proposed. A simplified version of the RZA-LMS algorithms obtained through the use of piece-wise approximation of the l_0-pseudo-norm has also been presented in [61].

4.3 Reweighted l_1-norm penalized LMS

The reweighted l_1-norm minimization for sparse signal recovery proposed in [69] has a better performance than the standard l_1-norm minimization that is usually employed in the CS literature. Therefore, one approach to enforce the sparsity of the solution for the sparsity-aware LMS-type algorithms is to introduce the reweighted l_1-norm penalty term in the cost function [67]. Our reweighted l_1-norm penalized LMS algorithm considers a penalty term proportional to the reweighted l_1-norm of the coefficient vector. The corresponding cost function can be written as

$$L_k^{\text{rl}_1} \triangleq (1/2)e_k^2 + \gamma_{\text{r}}\|\boldsymbol{s}_k\boldsymbol{w}_k\|_{l_1} \tag{4.25}$$

where γ_{r} is the weight associated with the penalty term and elements of $\boldsymbol{s}_k$ are set to

$$[\boldsymbol{s}_k]_i = \frac{1}{\epsilon_{\text{r}} + |[\boldsymbol{w}_{k-1}]_i|}, \quad i = 1, \ldots, N \tag{4.26}$$

with ϵ_{r} being some positive number. The update equation can be derived by differentiating (4.25) with respect to the vector of CIR coefficients and using the gradient descent principle shown in (2.18). The resulting update equation is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\text{r}}\frac{\text{sgn}(\boldsymbol{w}_k)}{\epsilon_{\text{r}} + |\boldsymbol{w}_{k-1}|} \tag{4.27}$$

where $\rho_{\text{r}} = \mu\gamma_{\text{r}}$. The absolute value operator as well as the $\text{sgn}(\cdot)$ and the division operator in the last term of (4.27) are all component-wise. Therefore, the variable $\text{sgn}(\boldsymbol{w}_k)/(\epsilon_{\text{r}} + |\boldsymbol{w}_{k-1}|)$ is a vector where its ith entry is $[\text{sgn}(\boldsymbol{w}_k)]_i/(\epsilon_{\text{r}} + |[\boldsymbol{w}_{k-1}]_i|)$. Note that although the weight vector $\boldsymbol{s}_k$ changes in every stage of this sparsity-aware LMS algorithm, it does not depend on $\boldsymbol{w}_k$, and the cost function $L_k^{\text{rl}_1}$ is convex unlike the cost function for the l_p-pseudo-norm penalized LMS and RZA-LMS algorithms. Therefore, the reweighted l_1-norm penalized LMS algorithm is

guaranteed to converge to the global minimum under some conditions. Thus, we study the convergence of the proposed algorithm in the next section.

4.4 Convergence study of the reweighted l_1-norm penalized LMS algorithm

The reweighted l_1-norm penalized LMS algorithm follows the logic that the penalty term resembling the l_0-pseudo-norm of the coefficient vector forces the solution of the modified LMS algorithm to be sparse. The cost function of the reweighted l_1-norm penalized LMS algorithm is given in (4.25), while the update equation is given in (4.27).

4.4.1 Mean convergence

We first study the mean convergence of the reweighted l_1-norm penalized LMS algorithm. The update equation for the coefficient error vector of the l_1-norm penalized LMS $\boldsymbol{v}_k$ can be written as

$$\begin{aligned}\boldsymbol{v}_{k+1} &= \boldsymbol{v}_k + \mu\big((\boldsymbol{w}^T - \boldsymbol{w}_k^T)\boldsymbol{x}_k + n_k\big)\boldsymbol{x}_k - \rho_{\mathrm{r}}\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \\ &= \boldsymbol{v}_k - \mu\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k + \mu n_k\boldsymbol{x}_k - \rho_{\mathrm{r}}\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}. \end{aligned} \tag{4.28}$$

Since $\boldsymbol{v}_k^T\boldsymbol{x}_k$ is a scalar which is equal to $\boldsymbol{x}_k^T\boldsymbol{v}_k$, (4.28) can be rewritten as

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \mu\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k + \mu n_k\boldsymbol{x}_k - \rho_{\mathrm{r}}\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}. \tag{4.29}$$

From (4.29) we can derive the evolution equation for $E\{\boldsymbol{v}_k\}$. Note that n_k and $\boldsymbol{x}_k$ are independent since n_k is assumed to have zero mean, i.e., $E\{\mu n_k\boldsymbol{x}_k\} = 0$. Then the evolution equation is

$$E\{\boldsymbol{v}_{k+1}\} = (\boldsymbol{I} - \mu\boldsymbol{R})\,E\{\boldsymbol{v}_k\} - \rho_{\mathrm{r}}E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\right\}. \tag{4.30}$$

It is easy to see that

$$\frac{-\mathbf{1}}{\epsilon_{\mathrm{r}}} \le \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \le \frac{\mathbf{1}}{\epsilon_{\mathrm{r}}} \tag{4.31}$$

where $\mathbf{1}$ is the vector with all of its entries set to one. Therefore, the vector $\rho_{\mathrm{r}}E\{\mathrm{sgn}(\boldsymbol{w}_k)/(\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|)\}$ is bounded between $(-\rho_{\mathrm{r}}/\epsilon_{\mathrm{r}})\mathbf{1}$ and $(\rho_{\mathrm{r}}/\epsilon_{\mathrm{r}})\mathbf{1}$. This bound on the second term on the right hand side of (4.31) is helpful for studying the mean convergence of the reweighted l_1-norm penalized LMS algorithm. The following theorem establishes our main result on the mean convergence of the reweighted l_1-norm penalized LMS algorithm.

Theorem 4.2. *If the maximal eigenvalue of the matrix $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then the mean coefficient error vector $E\{\boldsymbol{v}_k\}$ is bounded as $k \to \infty$.*

Proof. Let $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ be the eigenvalue decomposition of $\boldsymbol{R}$. Equation (4.30) can be rewritten as

$$E\{\boldsymbol{c}_{k+1}\} = (\boldsymbol{I} - \mu\boldsymbol{\Lambda})\, E\{\boldsymbol{c}_k\} - \boldsymbol{w}'_k \tag{4.32}$$

where

$$\begin{aligned} \boldsymbol{c}_k &\triangleq \boldsymbol{Q}^T\boldsymbol{v}_k \\ \boldsymbol{w}'_k &\triangleq \rho_{\mathrm{r}}\boldsymbol{Q}^T E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\right\}. \end{aligned} \tag{4.33}$$

Let also $\boldsymbol{q}$ be the vector whose ith entry is the sum of the absolute values of the elements in the ith row of the matrix $\boldsymbol{Q}^T$. The variable q_m is defined as the maximum element of the vector $\boldsymbol{q}$. The vector $\boldsymbol{Q}^T\mathrm{sgn}(\boldsymbol{w}_k)$ is thus bounded between $q_m\mathbf{1}$ and $-q_m\mathbf{1}$. Therefore, the variable $\boldsymbol{w}'_k$ in (4.33) is bounded between $(\rho_{\mathrm{r}}q_m/\epsilon_{\mathrm{r}})\mathbf{1}$ and $(-\rho_{\mathrm{r}}q_m/\epsilon_{\mathrm{r}})\mathbf{1}$.

It is easy to see from (4.32) that

$$E\{\boldsymbol{c}_{k+M}\} = (\boldsymbol{I} - \mu\boldsymbol{\Lambda})^M\, E\{\boldsymbol{c}_k\} - \sum_{m=0}^{M-1} (\boldsymbol{I} - \mu\boldsymbol{\Lambda})^m\, \boldsymbol{w}'_{k+M-m-1}. \tag{4.34}$$

Moreover, since $\boldsymbol{\Lambda}$ and correspondingly $\boldsymbol{I} - \mu\boldsymbol{\Lambda}$ are diagonal matrices, the convergence behavior of every element of the vector $E\{\boldsymbol{c}_{k+M}\}$ can be studied separately.

Let λ_i be the ith diagonal element of the matrix $\boldsymbol{\Lambda}$. From (4.34), we have

$$\left[E\{\boldsymbol{c}_{k+M}\}\right]_i = (1 - \mu\lambda_i)^M \left[E\{\boldsymbol{c}_k\}\right]_i - \sum_{m=0}^{M-1} (1 - \mu\lambda_i)^m \left[\boldsymbol{w}'_{k+M-m-1}\right]_i. \tag{4.35}$$

Since the largest eigenvalue of $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then all the diagonal elements $1 - \mu\lambda_i$ are smaller than 1. Also note that the ith entry of the vector $\boldsymbol{w}'_k$ is bounded between $\rho_{\mathrm{r}}q_m/\epsilon_{\mathrm{r}}$ and $-\rho_{\mathrm{r}}q_m/\epsilon_{\mathrm{r}}$. Therefore, by letting $M \to \infty$, the sum on the right hand side of (4.35) is a geometric series with a common ratio of $1 - \mu\lambda_i$ and is bounded between $\rho_{\mathrm{r}}q_m/(\mu\lambda_i\epsilon_{\mathrm{r}})$ and $-\rho_{\mathrm{r}}q_m/(\mu\lambda_i\epsilon_{\mathrm{r}})$. The other term on the right hand side of (4.35) approaches zero as $M \to \infty$. As a result, $\left[E\{\boldsymbol{c}_{k+M}\}\right]_i$ as well as the whole vector $E\{\boldsymbol{c}_{k+M}\}$ are bounded when $M \to \infty$. Since according to (4.33) $E\{\boldsymbol{c}_k\}$ is a rotated version of $E\{\boldsymbol{v}_k\}$, the coefficient error vector $\boldsymbol{v}_k$ is also bounded in mean. Therefore, if the largest eigenvalue of $\boldsymbol{I} - \mu\boldsymbol{R}$ is smaller than 1, then $E\{\boldsymbol{v}_k\}$ is bounded as $k \to \infty$. □

Note that the condition in Theorem 4.2 is the same as the mean convergence condition for the standard LMS algorithm given in Theorem 4.1.

4.4.2 Excess MSE

We now turn to the excess MSE calculation for the reweighted l_1-norm penalized LMS algorithm. Using the expression in (4.28) for $\boldsymbol{v}_{k+1}$, the variable $\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T$ can be written as

$$\begin{aligned}\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T = &\left(\boldsymbol{v}_k - \mu\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k + \mu n_k\boldsymbol{x}_k - \rho_{\mathrm{r}}\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\right)\\ &\times\left(\boldsymbol{v}_k^T - \mu\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T + \mu n_k\boldsymbol{x}_k^T - \rho_{\mathrm{r}}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|}\right).\end{aligned} \tag{4.36}$$

Expanding the right hand side of (4.36) and then taking expectation of the both sides results in the following equation.

$$\begin{aligned}E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\} = &E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\} - \mu\left(E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} + E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right)\\ &+ \mu^2 E\left\{n_k^2\boldsymbol{x}_k\boldsymbol{x}_k^T\right\} + \mu\left(E\left\{n_k\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} + E\left\{n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\right\}\right)\\ &- \mu^2\left(E\left\{n_k\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{x}_k^T\right\} + E\left\{n_k\boldsymbol{x}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\}\right)\\ &+ \mu^2 E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\}\\ &- \rho_{\mathrm{r}}\left(E\left\{\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|}\right\} + E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right\}\right)\\ &+ \mu\rho_{\mathrm{r}}\left(E\left\{\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|}\right\} + E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\right\}\right)\\ &- \mu\rho_{\mathrm{r}}\left(E\left\{n_k\boldsymbol{x}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|}\right\} + E\left\{n_k\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\boldsymbol{x}_k^T\right\}\right)\\ &+ \rho_{\mathrm{r}}^2\left(E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|}\right\}\right).\end{aligned} \tag{4.37}$$

Note that due to the independence of the additive noise n_k of the data and coefficient error vectors and due to the fact that the additive noise is zero mean, in addition to (4.11) we also have

$$E\left\{n_k\boldsymbol{x}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|}\right\} = E\left\{n_k\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}\boldsymbol{x}_k^T\right\} = 0.$$

Replacing $2\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\left\{\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}$ in (4.37), the expression for

$E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\}$ can be derived as

$$\begin{aligned}E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\} &= E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\} - \mu\left(E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right) + \mu^2\sigma_{\mathrm{n}}^2\boldsymbol{R} \\ &\quad + \mu^2\left(2\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\left\{\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}\right) \\ &\quad - \rho_{\mathrm{r}}\left((\boldsymbol{I}-\mu\boldsymbol{R})E\left\{\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}\right\} + E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right\}(\boldsymbol{I}-\mu\boldsymbol{R})\right) \\ &\quad + \rho_{\mathrm{r}}^2\left(E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}\right\}\right). \end{aligned} \tag{4.38}$$

Let $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ be defined as

$$\boldsymbol{A}_k \triangleq \rho_{\mathrm{r}}\left((\boldsymbol{I}-\mu\boldsymbol{R})E\left\{\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}\right\} + E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right\}(\boldsymbol{I}-\mu\boldsymbol{R})\right) \tag{4.39}$$

and

$$\boldsymbol{B}_k \triangleq \rho_{\mathrm{r}}^2\left(E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}\right\}\right). \tag{4.40}$$

Then, (4.38) can be rewritten as

$$\begin{aligned}E\left\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\right\} &= E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\} - \mu\left(E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right) + \mu^2\sigma_{\mathrm{n}}^2\boldsymbol{R} \\ &\quad + \mu^2\left(2\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\left\{\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}\right) - \boldsymbol{A}_k + \boldsymbol{B}_k. \end{aligned} \tag{4.41}$$

Letting $k \to \infty$ in (4.41), we obtain

$$\begin{aligned}\boldsymbol{R}_{\boldsymbol{v}} &= \boldsymbol{R}_{\boldsymbol{v}} - \mu\left(\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R} + \boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right) + \mu^2\sigma_{\mathrm{n}}^2\boldsymbol{R} \\ &\quad + \mu^2\left(2\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R} + \boldsymbol{R}\,\mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}\right) + \lim_{k\to\infty}\left(\boldsymbol{B}_k - \boldsymbol{A}_k\right). \end{aligned} \tag{4.42}$$

Crossing out $\boldsymbol{R}_{\boldsymbol{v}}$ from the both sides of (4.42) and then dividing the resulting equation by μ, we find that

$$\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R} + \boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}} - 2\mu\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R} = \mu\boldsymbol{R}\left(\sigma_{\mathrm{n}}^2 + \ \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}\right) + \frac{1}{\mu}\lim_{k\to\infty}\left(\boldsymbol{B}_k - \boldsymbol{A}_k\right). \tag{4.43}$$

Breaking $2\mu\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}$ into the sum of two identical terms and then factoring out $\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}$ and $\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}$, we also obtain

$$\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right) + \left(\boldsymbol{I}-\mu\boldsymbol{R}\right)\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R} = \mu\boldsymbol{R}\left(\sigma_{\mathrm{n}}^2 + \ \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}\right) + \frac{1}{\mu}\lim_{k\to\infty}\left(\boldsymbol{B}_k - \boldsymbol{A}_k\right). \tag{4.44}$$

Multiplying both sides of (4.44) by $\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}$ from right, the following can be derived

$$\begin{aligned}\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}} + \left(\boldsymbol{I}-\mu\boldsymbol{R}\right)\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1} &= \mu\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\left(\sigma_{\mathrm{n}}^2 + \ \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}\right) \\ &\quad + \frac{1}{\mu}\lim_{k\to\infty}\left(\boldsymbol{B}_k - \boldsymbol{A}_k\right)\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}. \end{aligned} \tag{4.45}$$

Note that $\sigma_{\rm n}^2 + \ \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}$ here is a scalar. Taking the trace of the two sides of (4.45), we have

$$\begin{aligned}\mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\} + \mathrm{tr}\left\{\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\} \\ = \mu\left(\sigma_{\rm n}^2 + \ \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}\right)\ \mathrm{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\} \\ + \frac{1}{\mu}\lim_{k\to\infty}\ \mathrm{tr}\left\{\left(\boldsymbol{B}_k-\boldsymbol{A}_k\right)\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}. \end{aligned} \tag{4.46}$$

Note that $\mathrm{tr}\left\{\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}$ equals $\mathrm{tr}\left\{\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}$ which in turn is equal to $\mathrm{tr}\left\{\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}\right\}$. Therefore, equation (4.46) can be simplified as follows

$$\begin{aligned}\mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\} + \mathrm{tr}\left\{\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}\right\} = \mu\left(\sigma_{\rm n}^2 + \ \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}\right)\ \mathrm{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\} \\ + \frac{1}{\mu}\lim_{k\to\infty}\ \mathrm{tr}\left\{\left(\boldsymbol{B}_k-\boldsymbol{A}_k\right)\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}. \end{aligned} \tag{4.47}$$

Since $\mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\} = \mathrm{tr}\left\{\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{R}\right\}$, we can further rewrite (4.47) as

$$\begin{aligned}\mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}\left(2-\mu\ \mathrm{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}\right) = \mu\sigma_{\rm n}^2\ \mathrm{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\} \\ + \frac{1}{\mu}\lim_{k\to\infty}\ \mathrm{tr}\left\{\left(\boldsymbol{B}_k-\boldsymbol{A}_k\right)\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}. \end{aligned} \tag{4.48}$$

Having in mind that the excess MSE ξ is given as $\xi = \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\}$, we obtain from (4.48) the following expression for ξ:

$$\xi = \ \mathrm{tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\boldsymbol{v}}\right\} = \frac{\eta}{2-\eta}\sigma_{\rm n}^2 + \frac{\beta-\alpha}{\mu(2-\eta)} \tag{4.49}$$

where $\eta \triangleq \mu\mathrm{tr}\left\{\boldsymbol{R}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}$, $\beta \triangleq \lim_{k\to\infty}\beta_k$, $\alpha \triangleq \lim_{k\to\infty}\alpha_k$, and the variables β_k and α_k are defined as $\beta_k \triangleq \mathrm{tr}\left\{\boldsymbol{B}_k\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}$, and $\alpha_k \triangleq \mathrm{tr}\left\{\boldsymbol{A}_k\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}$.

We now further examine the variables β_k and α_k. The matrix $\boldsymbol{B}_k\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}$ can be expressed as

$$\boldsymbol{B}_k\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1} \quad = \rho_{\rm r}^2\left(E\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\rm r}+|\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\rm r}+|\boldsymbol{w}_{k-1}^T|}\right\}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right). \tag{4.50}$$

Using (4.50), we obtain

$$\begin{aligned}\beta_k = \ \mathrm{tr}\left\{\boldsymbol{B}_k\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\} \\ = \rho_{\rm r}^2\left(E\left\{\ \mathrm{tr}\left\{\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\rm r}+|\boldsymbol{w}_{k-1}|}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\rm r}+|\boldsymbol{w}_{k-1}^T|}\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\}\right\}\right). \end{aligned} \tag{4.51}$$

Moreover, β_k in (4.51) can also be written as

$$\beta_k = \rho_{\mathrm{r}}^2 \left(E \left\{ \mathrm{tr} \left\{ \frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|} \left(\boldsymbol{I} - \mu \boldsymbol{R} \right)^{-1} \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \right\} \right\} \right). \tag{4.52}$$

The matrix $\boldsymbol{I} - \mu \boldsymbol{R}$ is symmetric, and its eigenvalue decomposition can be written as $\boldsymbol{I} - \mu \boldsymbol{R} = \boldsymbol{U} \boldsymbol{\Gamma} \boldsymbol{U}^T$ with $\boldsymbol{U}$ being an orthonormal matrix of eigenvectors and $\boldsymbol{\Gamma}$ being a diagonal matrix of eigenvalues. Therefore, $(\boldsymbol{I} - \mu \boldsymbol{R})^{-1} = \boldsymbol{U} \boldsymbol{\Gamma}^{-1} \boldsymbol{U}^T$ and β_k from equation (4.52) can be written as

$$\begin{aligned} \beta_k &= \rho_{\mathrm{r}}^2 \left(E \left\{ \mathrm{tr} \left\{ \frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|} \boldsymbol{U} \boldsymbol{\Gamma}^{-1} \boldsymbol{U}^T \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \right\} \right\} \right) \\ &= \rho_{\mathrm{r}}^2 \left(E \left\{ \mathrm{tr} \left\{ \boldsymbol{\Gamma}^{-1} \boldsymbol{U}^T \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|} \boldsymbol{U} \right\} \right\} \right). \end{aligned} \tag{4.53}$$

Let λ_{max} be the largest eigenvalue of the covariance matrix $\boldsymbol{R}$. Also, let μ be small enough such that $(1 - \mu \lambda_{max})^{-1}$ is positive. In (4.53), since $\boldsymbol{\Gamma}^{-1}$ is a diagonal matrix whose diagonal elements are all non-negative and less than or equal to $(1 - \mu \lambda_{max})^{-1}$, we have

$$\beta_k \leq \frac{\rho_{\mathrm{r}}^2}{1 - \mu \lambda_{max}} \times \left(E \left\{ \mathrm{tr} \left\{ \boldsymbol{U}^T \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|} \boldsymbol{U} \right\} \right\} \right). \tag{4.54}$$

Note that

$$\begin{aligned} \mathrm{tr} \left\{ \boldsymbol{U}^T \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|} \boldsymbol{U} \right\} &= \mathrm{tr} \left\{ \frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|} \boldsymbol{U} \boldsymbol{U}^T \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \right\} \\ &= \frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}^T|} \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|} \\ &\leq \frac{\mathrm{sgn}(\boldsymbol{w}_k^T) \mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}}^2} \leq \frac{N}{\epsilon_{\mathrm{r}}^2}. \end{aligned} \tag{4.55}$$

Substituting (4.55) in (4.54), the following bound on β_k can be finally obtained

$$\beta_k \leq \frac{N \rho_{\mathrm{r}}^2}{\epsilon_{\mathrm{r}}^2 \left(1 - \mu \lambda_{max} \right)}. \tag{4.56}$$

Moreover, β_k in (4.53) can also be written as

$$\beta_k = \rho_{\mathrm{r}}^2 \left(E \left\{ \mathrm{tr} \left\{ \boldsymbol{z}_k^T \boldsymbol{z}_k \right\} \right\} \right) = \rho_{\mathrm{r}}^2 \left(E \left\{ \| \boldsymbol{z}_k \|_{l_2}^2 \right\} \right) \tag{4.57}$$

where $\boldsymbol{z}_k$ is defined as

$$\boldsymbol{z}_k \triangleq \boldsymbol{\Gamma}^{-1/2} \boldsymbol{U}^T \frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}} + |\boldsymbol{w}_{k-1}|}. \tag{4.58}$$

Therefore, it can be seen from (4.57) that β_k is non-negative. Since, β_k is upper bounded and non-negative, so is β.

The variable α_k can be derived as

$$\begin{aligned}\alpha_k &= \mathrm{tr}\left\{\boldsymbol{A}_k\left(\boldsymbol{I}-\mu\boldsymbol{R}\right)^{-1}\right\} = \rho_{\mathrm{r}}\left(E\left\{\mathrm{tr}\left\{\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}+\frac{\mathrm{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\boldsymbol{v}_k^T\right\}\right\}\right)\\ &= 2\rho_{\mathrm{r}}\left(E\left\{\mathrm{tr}\left\{\boldsymbol{v}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}\right\}\right\}\right)\\ &= 2\rho_{\mathrm{r}}\left(E\left\{\mathrm{tr}\left\{\boldsymbol{w}_k\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}\right\}\right\}-E\left\{\mathrm{tr}\left\{\boldsymbol{w}\frac{\mathrm{sgn}(\boldsymbol{w}_k^T)}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}^T|}\right\}\right\}\right). \qquad (4.59)\end{aligned}$$

Assuming that $\lim_{k\to\infty} E\left\{\mathrm{sgn}(\boldsymbol{w}_k)\right\} = \mathrm{sgn}(\boldsymbol{w})$ which is a common assumption and it is, for example, the same as in [57], α_k in (4.59) can be written as

$$\alpha_k = 2\rho_{\mathrm{r}}\left(E\left\{\left\|\frac{\boldsymbol{w}_k}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\right\|_{l_1}\right\}-E\left\{\left\|\frac{\boldsymbol{w}}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\right\|_{l_1}\right\}\right). \qquad (4.60)$$

Defining $\beta' \triangleq \beta/\rho_{\mathrm{r}}^2$, and $\alpha' \triangleq \alpha/\rho_{\mathrm{r}}$, the excess MSE equation of (4.49) can be rewritten as

$$\xi = \frac{\eta}{2-\eta}\sigma_{\mathrm{n}}^2 + \frac{\beta'\rho_{\mathrm{r}}}{\mu(2-\eta)}\left(\rho_{\mathrm{r}}-\frac{\alpha'}{\beta'}\right) \qquad (4.61)$$

where β' is non-negative and upper bounded by $N/\epsilon_{\mathrm{r}}^2\left(1-\mu\lambda_{max}\right)$, and α' is given as

$$\alpha' = \lim_{k\to\infty} 2\left(E\left\{\left\|\frac{\boldsymbol{w}_k}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\right\|_{l_1}\right\}-E\left\{\left\|\frac{\boldsymbol{w}}{\epsilon_{\mathrm{r}}+|\boldsymbol{w}_{k-1}|}\right\|_{l_1}\right\}\right). \qquad (4.62)$$

It can be seen from (4.61) that if α' is positive, then choosing ρ_{r} in a way that $\rho_{\mathrm{r}} < \alpha'/\beta'$ can lead to the excess MSE of the reweighted l_1-norm penalized LMS algorithm being smaller than that of the standard LMS algorithm given in (4.21). The following example shows how the value of α' varies with respect to the sparsity level of the CIR that is being estimated.

Example 4.1. A time sparse CIR of length $N = 16$ whose sparsity level varies from $S = 1$ to $S = 16$ is considered in this example. The nonzero entries of the CIR take the values of 1 or -1 with equal probabilities each equal to half. In order to ensure a constant value for the term $\eta\sigma_{\mathrm{n}}^2/(2-\eta)$ in the excess MSE equation of (4.61) for different values of sparsity S, σ_{n}^2 is a constant set to 0.01. The step size μ is set to 0.05, while $\rho_{\mathrm{r}} = 5\times10^{-4}$ and $\epsilon_{\mathrm{r}} = 0.05$ in (4.27). Elements of the training sequence

Table 4.1: Value of α' for different sparsity levels.

S	1	2	3	4	5	6	7	8
α'	3.23	2.99	2.74	2.45	2.11	1.74	1.32	0.89
S	9	10	11	12	13	14	15	16
α'	0.39	-0.17	-0.79	-1.46	-2.23	-3.10	-4.07	-5.20

$\boldsymbol{x}_k$ are chosen with equal probability from the set $\{1, -1\}$. Table 4.1 shows the value of α' after 250 iterations of the reweighted l_1-norm penalized LMS algorithm for different sparsity levels.

The results in Table 4.1 show that as the CIR becomes less and less sparse, i.e., as S increases, α' becomes smaller to a point that it takes a negative value. Therefore, based on (4.61) we can expect a smaller excess MSE for the reweighted l_1-norm penalized LMS algorithm compared to that of the standard LMS algorithm providing that the sparsity level is small enough so that α' is positive.

4.5 l_p-pseudo-norm penalized LMS algorithm

Motivated by the better performance of RZA-LMS over ZA-LMS [57] due to the fact that the logarithmic penalty term of the RZA-LMS is closer to the l_0-pseudo-norm penalty, we consider another function of $\boldsymbol{w}_k$ that is more similar to the l_0-pseudo-norm. Such function is the l_p-pseudo-norm of $\boldsymbol{w}_k$ with $0 < p < 1$. The smaller the value of p is the more the l_p-pseudo-norm resembles the l_0-pseudo-norm. In this case, the cost function for the l_p-pseudo-norm penalized method becomes

$$L_k^{\mathrm{lp}} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{p}}\|\boldsymbol{w}_k\|_{l_p} \tag{4.63}$$

where $\|\cdot\|_{l_p}$ stands for the l_p-pseudo-norm of a vector and γ_{p} is the corresponding weight term. Similar to the cost function of the RZA-LMS, the cost function (4.63) is not convex and the analysis of the global convergence and consistency of the corresponding algorithm is problematic. However, as it will be seen in the next section, the method based on (4.63) shows better performance than the RZA-LMS which faces the same problems. Using gradient descent, the update equation based

on (4.63) can be derived as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{p}} \frac{\left(\|\boldsymbol{w}_k\|_{l_p}\right)^{1-p} \operatorname{sgn}(\boldsymbol{w}_k)}{|\boldsymbol{w}_k|^{(1-p)}} \tag{4.64}$$

where $\rho_{\mathrm{p}} = \mu\gamma_{\mathrm{p}}$. Practically, we need to impose an upper bound on the last term in (4.64) in the situation when an entry of $\boldsymbol{w}_k$ approaches zero, which is the case for a sparse CIR. Then the update equation (4.64) is modified as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{p}} \frac{\left(\|\boldsymbol{w}_k\|_{l_p}\right)^{1-p} \operatorname{sgn}(\boldsymbol{w}_k)}{\epsilon_{\mathrm{p}} + |\boldsymbol{w}_k|^{(1-p)}} \tag{4.65}$$

where ϵ_{p} is a value which bounds the last term in (4.64).

4.6 Penalized LMS algorithms for an arbitrary sparsity basis

The ZA-LMS and RZA-LMS algorithms in the form derived in [57] are only applied to the case when the channel is sparse in the time domain. However, these algorithms as well as the l_p-pseudo-norm penalized LMS and reweighted l_1-norm penalized LMS algorithms can be modified to accommodate the case of an arbitrary sparsity basis. Consider the ZA-LMS algorithm in the case when the CIR is sparse in another sparsity domain. Let $\boldsymbol{\Psi}$ be the $N \times N$ orthonormal matrix denoting a specific sparsity basis. The CIR $\boldsymbol{w}$ is sparse in the sparsity domain $\boldsymbol{\Psi}$ if its representation in $\boldsymbol{\Psi}$, that is, the vector $\boldsymbol{w}_\Psi = \boldsymbol{\Psi}\boldsymbol{w}$, has only a few nonzero components. The ZA-LMS cost function can be rewritten then as

$$L_k^{\mathrm{ZA}} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{ZA}} \|\boldsymbol{\Psi}\boldsymbol{w}_k\|_{l_1} \tag{4.66}$$

and the update equation becomes

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{ZA}} \operatorname{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)\boldsymbol{\Psi} \tag{4.67}$$

where $\operatorname{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)$ as well as $\operatorname{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)\boldsymbol{\Psi}$ are row vectors. For the reweighted l_1-norm penalized LMS algorithm the new cost function is

$$L_k^{\mathrm{rl}_1} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{r}} \|\boldsymbol{s}'_k \boldsymbol{\Psi}\boldsymbol{w}_k\|_{l_1} \tag{4.68}$$

with the entries of $\boldsymbol{s}'_k$ being

$$[\boldsymbol{s}'_k]_i = \frac{1}{\epsilon_{\mathrm{r}} + |[\boldsymbol{\Psi}\boldsymbol{w}_{k-1}]_i|}, \quad i = 1, \ldots, N. \tag{4.69}$$

From the above cost function, the update equation for the reweighted l_1-norm penalized LMS algorithm can be derived as

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{r}} \frac{\mathrm{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)\boldsymbol{\Psi}}{\epsilon_{\mathrm{r}} + |\boldsymbol{\Psi}\boldsymbol{w}_{k-1}|}. \tag{4.70}$$

Finally, the modified cost function of the l_p-pseudo-norm penalized LMS algorithm can be written as

$$L_k^{\mathrm{lp}} \triangleq (1/2)e_k^2 + \gamma_{\mathrm{p}} \|\boldsymbol{\Psi}\boldsymbol{w}_k\|_{l_p} \tag{4.71}$$

and the update equation for this algorithm becomes

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu e_k \boldsymbol{x}_k - \rho_{\mathrm{p}} \frac{\left(\|\boldsymbol{\Psi}\boldsymbol{w}_k\|_{l_p}\right)^{1-p} \mathrm{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)\boldsymbol{\Psi}}{\epsilon_{\mathrm{p}} + |\boldsymbol{\Psi}\boldsymbol{w}_k|^{(1-p)}}. \tag{4.72}$$

4.7 Decimated LMS algorithms for frequency sparse channels

The signals of interest here are those that are sparse in a domain such that they can be recovered from random time samples. We aim at exploiting this feature in order to design less complex or more accurate variations of the LMS algorithm. The basic idea of our decimated LMS is to estimate at even time-steps only the even taps of the CIR and do so for the odd taps at odd time-steps. To this end, the training sequence is chosen in a way that nothing is being sent into the channel at every other time-step. For example, a sample training sequence with binary phase shift keying (BPSK) symbols is $\boldsymbol{x} = 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, \cdots$. In this way, at odd time-steps only the odd taps of the channel contribute in the received symbol d_k and the even taps only affect d_k at even time-steps. This special training sequence can also be beneficial in a multiuser setup. Consider a case when two users are using a shared medium and their signals interfere with each other. Using an alternatively on and off training sequence one can use the algorithms introduced here in order to estimate the CIRs of the two users simultaneously. The even taps of the CIR of the first user can be updated during the even time-steps while the odd taps of the CIR of the second user is updated during the odd time-steps. Due to the special pattern of the training sequence the two users do not interfere with each other in the channel estimation phase.

We develop two variations of the LMS algorithm based on the aforementioned idea [72,73]. Note however that other channel estimation algorithms can be modified based on this idea and the LMS algorithm is selected only as a popular example.

According to the first algorithm, only even or odd taps have to be updated, and at the end of the training phase, an l_1-norm minimization problem is solved to estimate all channel taps. The other algorithm is an adaptation of the ZA-LMS which alternatively updates the even and odd channel taps at each time-step.

4.7.1 Decimated LMS

Let $\boldsymbol{w}_k^o$ be the set of odd indexed coefficients of the vector $\boldsymbol{w}_k$ and $\boldsymbol{w}_k^e$ denote the even indexed entries of $\boldsymbol{w}_k$. The training data sequence $\boldsymbol{x}$ is designed so that a BPSK symbol is being sent into the channel at odd time-steps and nothing is sent at even time-steps. Therefore, at odd time-steps when $n = 2i + 1$, we have $y_k = \boldsymbol{w}^T \boldsymbol{x}_k = (\boldsymbol{w}^o)^T \boldsymbol{x}_k^o$, since $\boldsymbol{x}_k^e$ will be an all zero vector. Also, at even time-steps when $n = 2i$, $y_k = (\boldsymbol{w}^e)^T \boldsymbol{x}_k^e$. In the decimated LMS algorithm, either the even or odd taps of the CIR are being estimated. Let us assume that an estimate of the odd taps is to be obtained. Then at every odd time-step the following LMS type update rule is used

$$\boldsymbol{w}_{k+2}^o = \boldsymbol{w}_k^o + \mu e_k \boldsymbol{x}_k^o \tag{4.73}$$

where $e_k = d_k - (\boldsymbol{w}_k^o)^T \boldsymbol{x}_k^o$. At the end of the training process, the LMS estimate of the CIR's odd taps $\boldsymbol{w}_{k_f}^o$ is available where k_f is the final training time-step. The estimate of the complete CIR at time-step k_f, $\boldsymbol{w}_{k_f}$, is obtained as $\boldsymbol{w}_{k_f} = \boldsymbol{\Psi}^{-1}(\boldsymbol{w}_\Psi)_{k_f}$ where $(\boldsymbol{w}_\Psi)_{k_f}$ is the solution to the following minimization problem

$$\min \|\tilde{\boldsymbol{w}}\|_{l_1} \quad \text{subject to} \quad \left\| \left(\boldsymbol{\Psi}^{-1}\tilde{\boldsymbol{w}}\right)^o - \boldsymbol{w}_{k_f}^o \right\|_{l_2} \leq \beta \tag{4.74}$$

when β is some positive number. The above equation is the l_1-norm minimization problem for the case of noisy compressed samples [33].

The decimated LMS deals with vectors of a smaller size than the standard LMS. In the case when the total number of CIR taps, i.e., the cardinality of $\boldsymbol{w}_k$ is even, the size of the vectors in the decimated LMS is exactly half the standard LMS. In addition, the decimated LMS is only executed at odd or even time-steps depending on whether it estimates $\boldsymbol{w}_k^o$ or $\boldsymbol{w}_k^e$. Therefore, the decimated LMS is run only half the times that the standard LMS is executed. Considering the size of the vectors involved as well as the number of iterations that each algorithm needs for estimating CIR, we can conclude that the complexity of the decimated LMS for estimating $\boldsymbol{w}_k^o$ or $\boldsymbol{w}_k^e$ is about a fourth of the standard LMS. However, decimated LMS has to solve

an l_1-norm minimization problem at the end of the training process to obtain the estimate of the complete CIR $\boldsymbol{w}_k$.

4.7.2 Zero attracting decimated LMS

Motivated by the fact that ZA-LMS has better performance than the standard LMS, we also present the zero attracting decimated LMS (ZAD-LMS) algorithm. In this method, both even and odd taps of the CIR are being estimated. In order to force sparsity of the CIR, a term similar to the one in the update equation for the ZA-LMS in (4.67) is present in ZAD-LMS. Let us first consider odd taps of the CIR. For ZAD-LMS, the terms $\boldsymbol{w}_{k+1}$ and $\boldsymbol{w}_k$ in (4.67) are replaced with $\boldsymbol{w}^o_{k+2}$ and $\boldsymbol{w}^o_k$, while $\mu e_k \boldsymbol{x}^o_k$ replaces $\mu e_k \boldsymbol{x}_k$. Now we only need to replace $\rho_{\text{ZA}}\text{sgn}(\boldsymbol{\Psi}\boldsymbol{w}_k)\boldsymbol{\Psi}$ with a similar term that results in a vector of the same size as $\boldsymbol{w}^o_k$. We choose $\rho_{\text{ZAD}}\text{sgn}(\boldsymbol{\Psi}\tilde{\boldsymbol{w}}_k)\boldsymbol{\Psi}^o$ where $\boldsymbol{\Psi}^o$ is a sub-matrix of $\boldsymbol{\Psi}$ with only odd indexed columns retained. Note that $\tilde{\boldsymbol{w}}_k$ is also used instead of the current estimate of CIR $\boldsymbol{w}_k$, where $\tilde{\boldsymbol{w}}_k$ is set to an all zero vector when the algorithm starts at $n = 1$ time-step and also for $n = 2$. For every $k = 2i + 1 > 2$, $\tilde{\boldsymbol{w}}_k$ has its odd indexed components set equal to $\boldsymbol{w}^o_{k-2}$ and its even indexed components are set equal to $\boldsymbol{w}^e_{k-1}$. When $k = 2i > 2$, $\boldsymbol{w}^o_{k-1}$ and $\boldsymbol{w}^e_{k-2}$ are used to form the vector $\tilde{\boldsymbol{w}}_k$. Therefore, the set of update equations of the ZAD-LMS can be written as

$$\boldsymbol{w}^o_{k+2} = \boldsymbol{w}^o_k + \mu e_k \boldsymbol{x}^o_k - \rho_{\text{ZAD}}\text{sgn}(\boldsymbol{\Psi}\tilde{\boldsymbol{w}}_k)\boldsymbol{\Psi}^o, \;\; k = 2i + 1 \tag{4.75a}$$

$$\boldsymbol{w}^e_{k+2} = \boldsymbol{w}^e_k + \mu e_k \boldsymbol{x}^e_k - \rho_{\text{ZAD}}\text{sgn}(\boldsymbol{\Psi}\tilde{\boldsymbol{w}}_k)\boldsymbol{\Psi}^e, \;\; k = 2i. \tag{4.75b}$$

At the end of training, i.e., at the k_fth time-step, the two vectors $\boldsymbol{w}^o_{k_f}$ and $\boldsymbol{w}^e_{k_f-1}$ or $\boldsymbol{w}^e_{k_f}$ and $\boldsymbol{w}^o_{k_f-1}$, depending on k_f being odd or even, produce the estimate of CIR $\boldsymbol{w}_{k_f}$. The ZAD-LMS method is summarized in Algorithm 4.1.

4.8 Simulation results

In this section we compare the performance of different channel estimation algorithms under several scenarios. The algorithms being considered here are ZA-LMS, and RZA-LMS of [57] as well as the reweighted l_1-norm penalized LMS, and the l_p-pseudo-norm penalized LMS of [67]. The performance of decimated type LMS algorithms is also studied. The standard LMS algorithm is also included as a reference point in all simulation examples.

Algorithm 4.1 The ZAD-LMS Algorithm

Input: Data sequence $\boldsymbol{x}_k$ and observations d_k, $k = 1, \cdots, k_f$.
Output: The estimated channel $\boldsymbol{w}_{k_f}$.
1. Initialize by equating $\tilde{\boldsymbol{w}}_1$ and $\tilde{\boldsymbol{w}}_2$ to all zero vector.
2. Perform k_f iterations of ZAD-LMS:
for $k = 1$ to k_f **do**
  if k is odd, **then**
    Update $\tilde{\boldsymbol{w}}_k$ with $\boldsymbol{w}_{k-2}^o$ and $\boldsymbol{w}_{k-1}^e$.
    Run the ZAD-LMS update equation (4.75a).
  else
    Update $\tilde{\boldsymbol{w}}_k$ with $\boldsymbol{w}_{k-1}^o$ and $\boldsymbol{w}_{k-2}^e$.
    Run the ZAD-LMS update equation (4.75b).
  end if
end for
3. Form $\boldsymbol{w}_{k_f}$ using $\boldsymbol{w}_{k_f}^o$ and $\boldsymbol{w}_{k_f-1}^e$ or $\boldsymbol{w}_{k_f}^e$ and $\boldsymbol{w}_{k_f-1}^o$.

4.8.1 Simulation example 1: Time sparse channel estimation

In this example, we consider the problem of estimating a CIR of length $N = 16$. The CIR is assumed to be sparse in the time domain. Two different sparsity levels of $S = 1$ and $S = 4$ are considered. The positions of the nonzero taps in the CIR are chosen randomly. The value of each nonzero tap is a zero mean Gaussian random variable with a variance of 1.

Two different SNR values of 10 dB and 20 dB are considered. For the l_p-pseudo-norm penalized LMS algorithm, p is chosen to be 1/2 with $\epsilon_{\mathrm{p}} = 0.05$ and $\rho_{\mathrm{p}} = 2 \times 10^{-4}$. The parameters of the reweighted l_1-norm penalized LMS algorithm are set to $\rho_{\mathrm{r}} = 2 \times 10^{-4}$ and $\epsilon_{\mathrm{r}} = 0.05$. For the ZA-LMS and the RZA-LMS algorithms, $\rho_{\mathrm{ZA}} = 5 \times 10^{-4}$, $\rho_{\mathrm{RZA}} = 4 \times 10^{-3}$, and $\epsilon_{\mathrm{RZA}} = 25$. Parameter values for the ZA-LMS and RZA-LMS algorithms are optimized through simulations. The step size is set to $\mu = 0.05$ for all algorithms. The measure of performance is the MSE between the actual and estimated CIR. Simulation results are averaged over 10000 simulation runs to smooth out the curves.

Fig. 4.1 shows the MSE results versus the number of iterations for different estimation algorithms for the case when the sparsity level is $S = 1$. It can be seen that for both SNR values tested the l_p-pseudo-norm penalized LMS algorithm has the best performance followed by the reweighted l_1-norm penalized LMS algorithm, and then by the RZA-LMS, ZA-LMS, and standard LMS algorithms. The MSEs of the RZA-LMS and reweighted l_1-norm penalized LMS algorithms are close to each

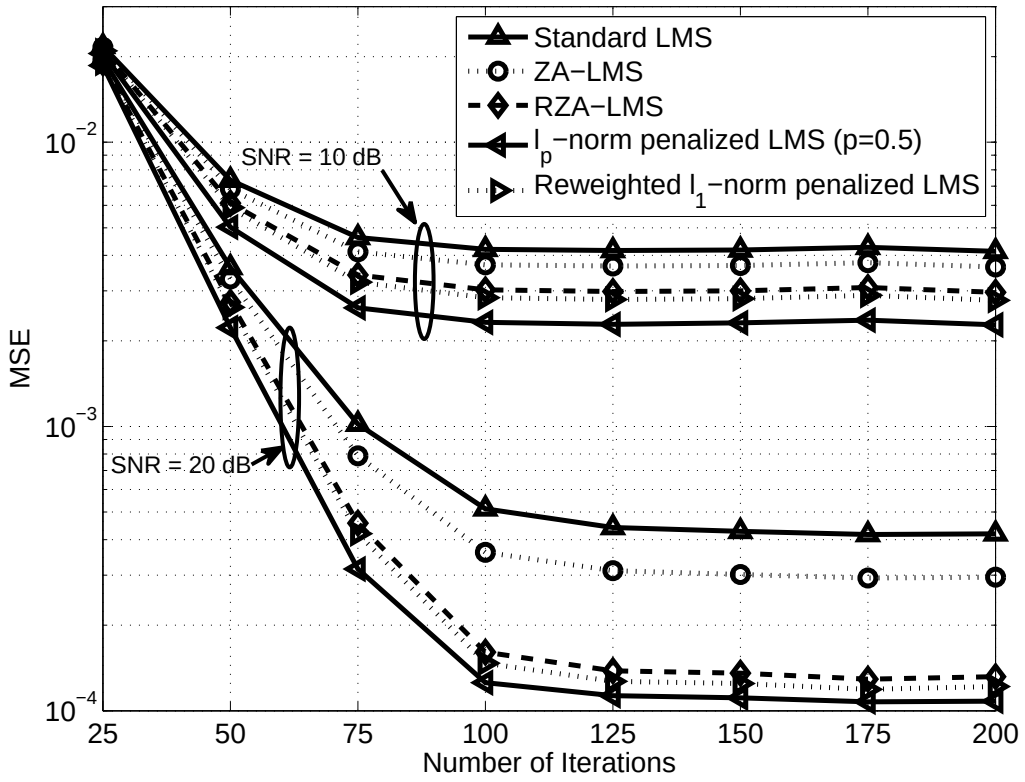


Figure 4.1: Simulation example 1, Case 1: MSEs of different estimation algorithms vs number of iterations ($S = 1$).

other. As the SNR increases, the performance of all the algorithms improves as expected. Also, it can be seen in Fig. 4.1 that the performance gap between the MSE of the standard LMS algorithm and the MSEs for the rest of the algorithms gets bigger. The l_p-pseudo-norm penalized LMS and reweighted l_1-norm penalized LMS algorithms have faster convergence rate compared to the standard LMS algorithm.

Fig. 4.2 shows the simulation results for the case when the sparsity level is set to $S = 4$. The parameter choices for all the algorithms are the same as in the previous case. Most of the observations from Fig. 4.1 also hold for this case of increased sparsity level. However, increasing the sparsity level of the CIR has lead to a decrease in the performance gap between the sparsity-aware LMS algorithms and standard LMS algorithm.

4.8.2 Simulation example 2: Arbitrary sparsity basis

In this simulation example, a CIR of length $N = 16$ with the sparsity level of $S = 2$ is being estimated which is sparse in the discrete cosine transform (DCT) domain. The positions of nonzero taps in the DCT domain are chosen randomly.

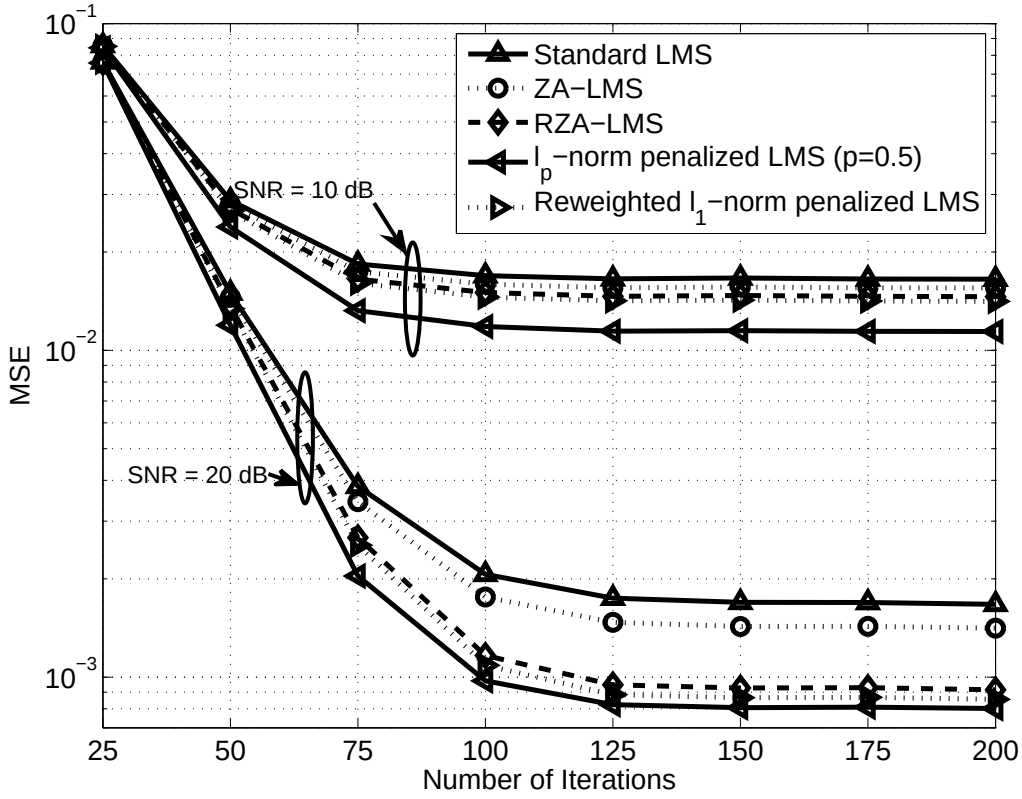


Figure 4.2: Simulation example 1, Case 2: MSEs of different estimation algorithms vs number of iterations ($S = 4$).

The value of the nonzero indices in the DCT domain are set to 1 or -1 with the same probabilities each equal to half. The algorithms being compared here are the ZA-LMS, RZA-LMS, l_p-pseudo-norm penalized LMS, reweighted l_1-norm penalized LMS, and standard LMS algorithms. As in the first simulation scenario, two different SNR values of 10 and 20 dBs are tested. Parameter choices for the 10 dB SNR case are as follows. For the l_p-pseudo-norm penalized LMS algorithm, $p = 1/2$, $\epsilon_{\rm p} = 0.05$, and $\rho_{\rm p} = 2 \times 10^{-4}$. Parameters of the reweighted l_1-norm penalized LMS algorithm are $\rho_{\rm r} = 2 \times 10^{-4}$ and $\epsilon_{\rm r} = 0.05$. For the ZA-LMS and the RZA-LMS algorithms, the values are $\rho_{\rm ZA} = 5 \times 10^{-4}$, $\rho_{\rm RZA} = 4 \times 10^{-3}$, and $\epsilon_{\rm RZA} = 25$. The step size μ is set to 0.05. For the 20 dB SNR case, $\rho_{\rm r}$, $\rho_{\rm p}$, and $\rho_{\rm RZA}$ are reduced by half.

The MSE curves in Fig. 4.3 are averaged over 10000 simulation runs. The same conclusions as in Simulation example 1 hold here as well. For the SNR of 10 dB SNR, the l_p-pseudo-norm penalized LMS algorithm outperforms all the other algorithms followed by the reweighted l_1-norm penalized LMS algorithm, and then by the RZA-LMS and ZA-LMS algorithms. However, when the SNR is set to 20 dB, the reweighted l_1-norm penalized LMS and RZA-LMS algorithms show a better

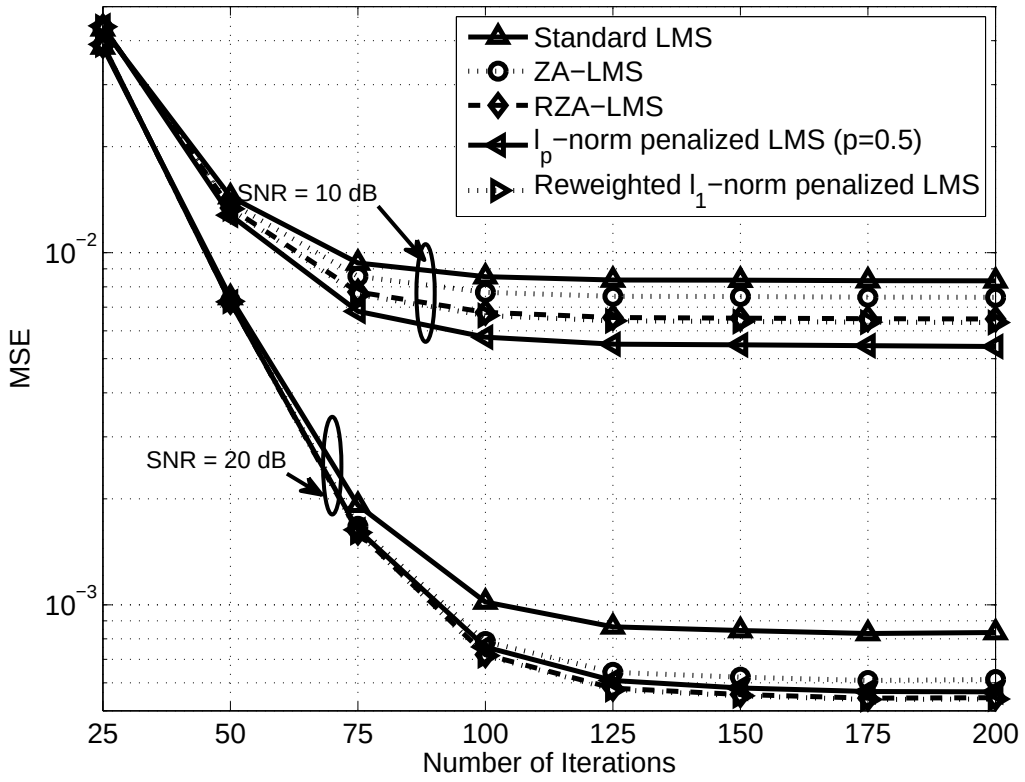


Figure 4.3: Simulation example 2: MSEs of estimation algorithms vs number of iterations for a DCT sparse channel ($S = 2$).

performance than the l_p-pseudo-norm penalized LMS algorithm.

4.8.3 Simulation example 3: Effect of sparsity level on the performance of the reweighted l_1-norm penalized LMS algorithm

In this example, we study the effect that the increasing sparsity level of CIR has on the performance of the reweighted l_1-norm penalized LMS algorithm. A CIR is assumed to be sparse in the time domain and it is of length $N = 16$. The sparsity level varies from 2 to 8. The positions of the nonzero taps of the CIR are chosen randomly and the values of nonzero taps are set to 1 or -1 with equal probability each equal to half. Parameters of the reweighted l_1-norm penalized LMS algorithm are $\rho_{\mathrm{r}} = 2 \times 10^{-4}$ and $\epsilon_{\mathrm{r}} = 0.05$. The step size μ is set to 0.05. Variance of the additive noise term n_k is $\sigma_{\mathrm{n}}^2 = 0.01$. Excess MSE is used as a performance measure in this example. We have chosen a constant variance σ_{n}^2 for the noise in order to make sure that the standard LMS algorithm has the same excess MSE regardless of the sparsity level of the channel. The excess MSE curves are averaged over 10000 simulation runs. According to (4.13), the excess MSE can

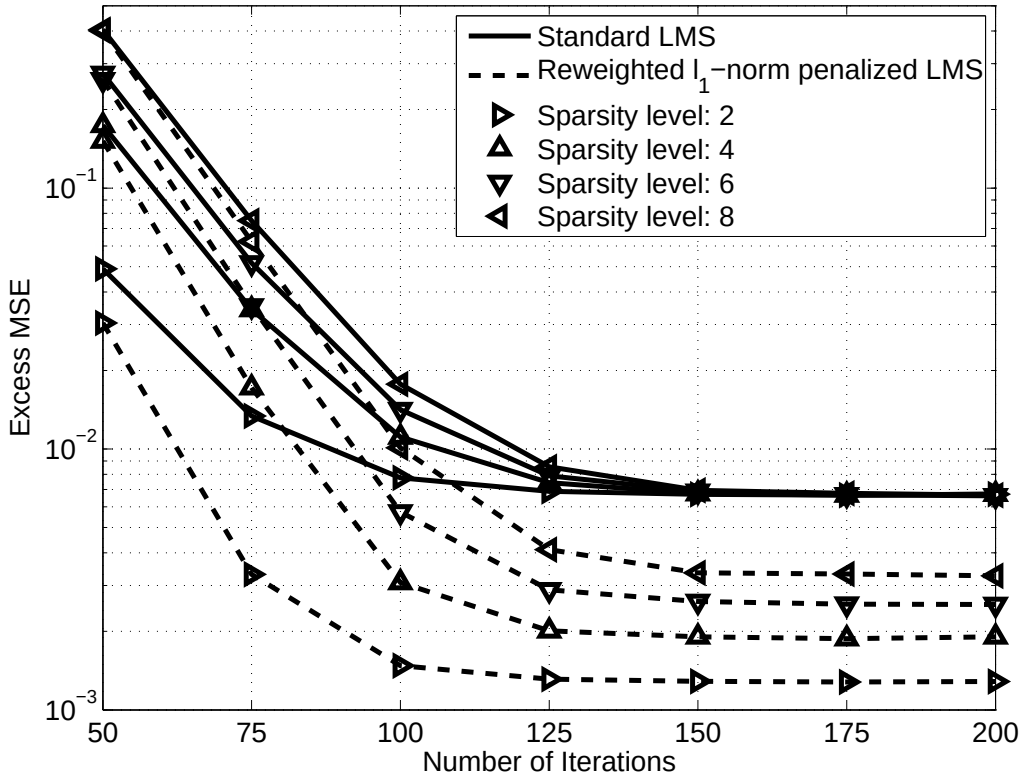


Figure 4.4: Simulation example 3: Excess MSE versus number of iterations.

be derived as $\xi_k = \mathrm{tr}\left\{\boldsymbol{R}E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}$. In this simulation example with $\boldsymbol{x}_k$ being an i.i.d. BPSK sequence, the covariance matrix $\boldsymbol{R}$ becomes identity, and therefore, ξ_k can be evaluated as $\mathrm{tr}\left\{E\left\{\boldsymbol{v}_k\boldsymbol{v}_k^T\right\}\right\}$.

Fig. 4.4 shows the excess MSE versus the number of iterations for the standard LMS and reweighted l_1-norm penalized LMS algorithms when the CIR sparsity level is varied from 2 to 8. It can be seen that the standard LMS algorithm results in the same excess MSE regardless of the sparsity level of the CIR. However, the excess MSE of the reweighted l_1-norm penalized LMS algorithm increases with increasing sparsity level which is due to the fact that the value of α' in equation (4.62) is decreasing. For example, α' is equal to 2.7, 2.3, 2.0, and 1.6 for sparsity levels of 2, 4, 6, and 8, respectively, after 150 iterations. It can be also seen that in all cases, the reweighted l_1-norm penalized LMS algorithm outperforms the standard LMS algorithm.

4.8.4 Simulation example 4: Performance comparison of decimated LMS algorithms

In this simulation example, the problem of estimating a CIR of length $N = 16$ is considered. The sparsity domain is the DCT domain. The sparsity level of the CIR is 1, i.e., only one of the coefficients of its representation in $\mathbf{\Psi}$ is nonzero. This nonzero coefficient takes the value of either 1 or -1 with the same probability.

Alongside the proposed algorithms, i.e., the decimated LMS and ZAD-LMS, the performance of the standard LMS and ZA-LMS methods are also measured as a point of reference. Parameter choices for these algorithms are $\rho_{ZA} = \rho_{ZAD} = 5 \times 10^{-4}$, $\mu = 0.05$. In the decimated LMS the parameter β in (4.74) is set to 0.1 when the SNR equals 10 dB, and it is set to 0.05 when SNR = 20 dB.

Fig. 4.5 shows the MSEs of different estimation methods versus the length of the training sequence, i.e., the number of iterations. It can be seen that after a certain number of iterations the decimated LMS catches up with the standard LMS and it then displays a better performance despite it being less complex than the standard LMS. For both SNRs of 10 and 20 dB, the ZA-LMS has the best performance while the proposed ZAD-LMS method shows a better performance than the standard LMS for all training sequence lengths unlike the decimated LMS whose performance is worse than that of the LMS for small training sequences.

4.8.5 Simulation example 5: Effect of sparsity level on the decimated LMS algorithm

This simulation example tests the effect of sparsity level on the performance of the decimated LMS. Since decimated LMS solves an l_1-norm minimization problem at the end of training to find an estimate of CIR, it is expected that its performance deteriorates with increasing the sparsity level of CIR. In this example, a CIR of length 64 is chosen. The sparsity basis is the DCT domain and the sparsity level of the CIR is varied from $S = 1$ to $S = 4$. The SNR is 10 dB, $\mu = 0.005$, and $\beta = 0.1$ in (4.74). Results are averaged over ten thousand simulation runs.

Fig. 4.6 depicts the MSE curves versus length of the training sequence. In this figure, standard LMS is chosen as the point of reference and its performance is compared with decimated LMS for different sparsity levels. In order to allow for a fare comparison, the CIR's energy is kept the same which results in the same MSE values for standard LMS regardless of a specific sparsity level. It can be seen from

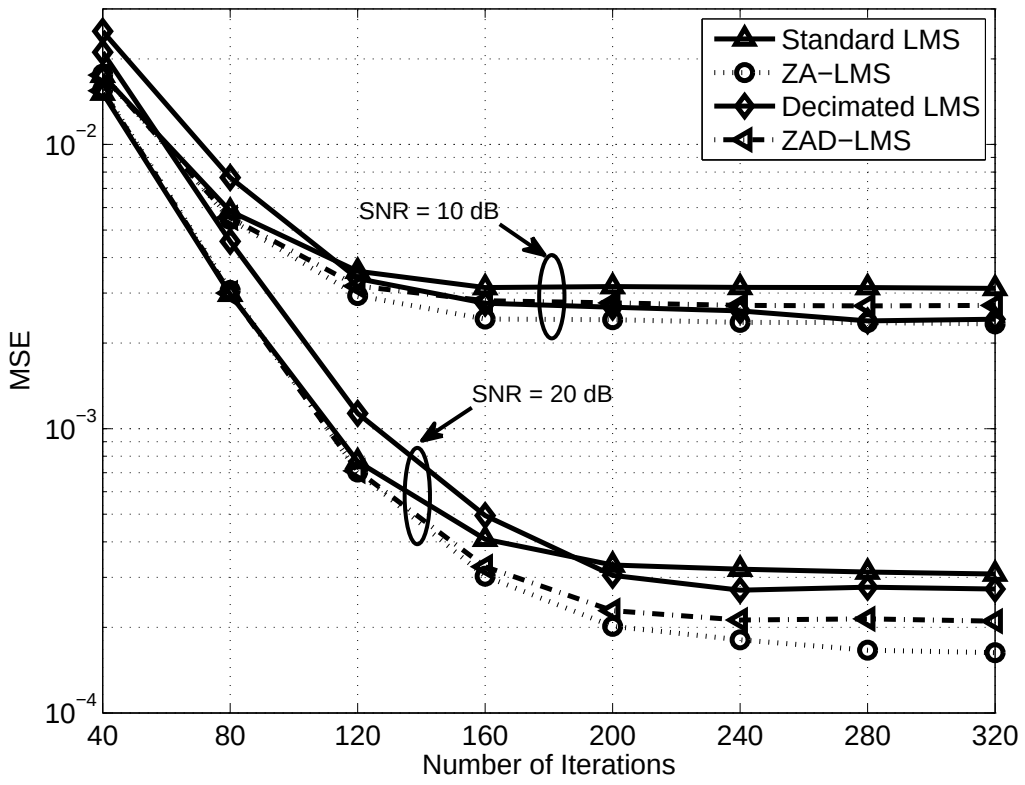


Figure 4.5: Simulation example 4: Performance comparsion of different decimated estimation algorithms.

Fig. 4.6 that the smaller the sparsity level of the signal the better is the performance of the decimated LMS as it is expected.

4.9 Chapter summary

Sparse channel estimation problem has been considered in this chapter and the reweighted l_1-norm penalized LMS algorithm has been introduced. Mathematical analysis of reweighted l_1-norm penalized LMS's mean convergence as well as the attainable excess MSE of the algorithm was presented. The excess MSE result shows that the reweighted l_1-norm penalized LMS can outperform the standard LMS algorithm. Update equations of the reweighted l_1-norm penalized LMS as well as the ZA-LMS, RZA-LMS, and the l_p-pseudo-norm penalized LMS algorithms were generalized to the case of an arbitrary sparsity basis. Simulation results for a DCT sparse channel are also given along the time sparse channel example. The performance of the reweighted l_1-norm penalized LMS has been compared to that of the standard LMS, ZA-LMS, RZA-LMS, and our l_p-pseudo-norm penalized LMS

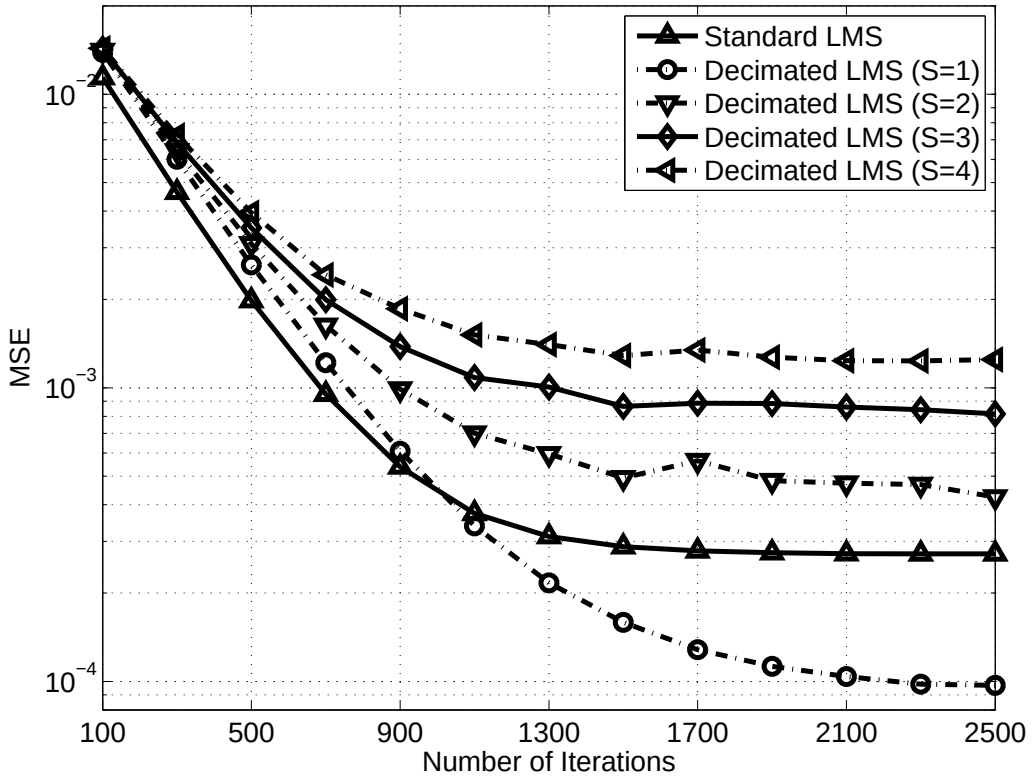


Figure 4.6: Simulation example 5: Performance of the decimated LMS and the standard LMS for different sparsity levels.

through computer simulations. These results show that the reweigthed l_1-norm penalized LMS outperforms the standard LMS, ZA-LMS, and the RZA-LMS.

Two sparsity aware modifications of the standard LMS algorithm, which are the decimated LMS and ZAD-LMS, for frequency sparse channel estimation have been introduced. They are motivated by the need of deriving channel estimation methods with lower complexity. The algorithms have been compared in terms of MSE to the standard LMS and the ZA-LMS. Simulations demonstrating the effectiveness of the proposed methods have been also shown.

Chapter 5

Two Dimensional Compressive Sampling

A straightforward solution to collecting compressed samples of a 2D signal is to vectorize the signal first. A measurement matrix can then be applied to collect the compressed samples. Let the 2D signal of size $N \times N$ be vectorized into a $N^2 \times 1$ vector. The measurement matrix being used to sample this signal is of size $K \times N^2$. Depending on the value of N^2 this measurement matrix becomes too large and the recovery problem becomes complex. In this chapter we propose another alternative by sampling the columns of the 2D signal with the same measurement matrix in order to reduce complexity and also be able to perform parallel recovery.

The rest of the chapter is organized as follows. The parallel CS method is introduced in Section 5.1. Section 5.2 describes the video signal model used in this chapter. The CS-based video compression algorithm as well as a motion estimation enabled version of it is also elaborated in this section. Section 5.3 concludes the chapter.

5.1 Parallel CS of 2D signals

Let us first introduce the notations used in this chapter to access a specific part of a vector or a matrix. $[\boldsymbol{v}]_{a:b}$ denotes a sub-vector of $\boldsymbol{v}$ including entries of the vector from index a to index b. For a matrix $\boldsymbol{M}$, $[\boldsymbol{M}]_{r,:}$ denotes the vector corresponding to the rth row of the matrix, while $[\boldsymbol{M}]_{:,c}$ denotes the cth column of it.

Let $\boldsymbol{X}$ be a sparse matrix of size $N \times N$. The vector $\boldsymbol{x}_v$ can be formed from $\boldsymbol{X}$ by putting all the columns of the matrix on top of each other. $\boldsymbol{x}_v$ is of size $1 \times N^2$

and is given as

$$[\boldsymbol{x}_v]_{(n-1)N+1:nN} = [\boldsymbol{X}]_{:,n}, \quad n = 1, 2, \cdots, N. \tag{5.1}$$

The measurement matrix $\boldsymbol{\Phi}$ of size $K \times N^2$ can be used to collect the vector of compressed samples $\boldsymbol{y}$ from $\boldsymbol{X}$ as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}_v$. In order to find an estimate of $\boldsymbol{x}_v$ from $\boldsymbol{y}$ a CS recovery algorithm such as the following is used.

$$\min \|\hat{\boldsymbol{x}}_v\|_{l_1}, \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi}\hat{\boldsymbol{x}}_v. \tag{5.2}$$

We pursue a different approach to two dimensional CS which samples every column of $\boldsymbol{X}$ with the same measurement matrix $\boldsymbol{\Phi}'$. The matrix $\boldsymbol{\Phi}'$ is of size $K' \times N$ and K' is chosen in a way so that the least sparse column of $\boldsymbol{X}$ can be recovered from K' compressed samples. In order to force the columns of $\boldsymbol{X}$ to have the same sparsity level we first apply random permutations on the rows of the matrix. The average sparsity level of a column after permutation is going to be smaller than the maximum sparsity level of the original matrix $\boldsymbol{X}$. Therefore, we could choose the number of rows of $\boldsymbol{\Phi}'$, K', to be smaller than the case of the non-permuted matrix.

The complexity of the l_1-norm minimization method used to recover an $N \times 1$ vector from its compressed samples is $O(N^3)$. Therefore, the first approach to two dimensional CS which vectorizes the matrix $\boldsymbol{X}$ has a complexity of $O(N^6)$. However, the last approach solves N parallel recovery problems each with a complexity of $O(N^3)$ and so in total the complexity is $N \cdot O(N^3)$. The advantages of the parallel CS method are lower complexity and ability to solve parallel recovery problems. The disadvantage of this method is that the total number of samples collected is more than the number of samples in the first approach where the matrix is first vectorized and then sampled. The following example illustrates the aforementioned.

Example 5.1. This example compares the performance of the parallel CS method and the vectorizing approach to two dimensional CS. Let $N = 20$, and so the sparse matrix $\boldsymbol{X}$ be of size 20×20. The measurements are corrupted with additive white Gaussian noise (AWGN) with a standard deviation of $\sigma = 0.05$. Only $S = 20$ out of 400 entries of $\boldsymbol{X}$ are set to ± 1 with a probability of $1/2$ and the rest of the entries are set to zero. For the parallel method we make sure that every column of $\boldsymbol{X}$ has the same number of nonzero entries, so the comparison with the vectorizing approach would be fair. Table 5.1 shows the simulation results for different values of K' and K. Average MSEs are provided in the table over 100 simulation runs. In this table,

Table 5.1: MSE values for different scenarios.

(K', K)	Total number of samples (approach 2, approach 1)	MSE (approach 2, approach 1)
$(8, 160)$	$(160, 160)$	$(0.0020, 0.0013)$
$(10, 160)$	$(200, 160)$	$(0.0016, 0.0012)$
$(12, 160)$	$(240, 160)$	$(0.0012, 0.0011)$
$(16, 160)$	$(320, 160)$	$(0.0011, 0.0011)$

approach 1 is the vectorizing approach and approach 2 is the parallel CS. As it can be seen in Table 5.1, in order for approach 2 to achieve the same MSE as approach 1 we need to collect twice as many samples as required for approach 1.

5.2 Video compression application

The amount of raw data in a video file is very large and therefore, transmitting such data consumes a large portion of system's resources such as bandwidth and energy. It should also be noted that there are several types of redundancies available in a video. These redundancies can be identified and taken advantage of to compress the video file with a lower bit rate than the bit rate of the raw video stream. The major types of redundancies in video are statistical, spatial and temporal redundancies [74]. Statistical redundancy addresses the fact that there are several pixels in a video frame with the same gray levels or color in case of a black and white picture or respectively a color video. The so-called spatial redundancy represents the correlation among neighboring pixels. In fact, in a typical video frame the gray levels change smoothly over a region. Finally the temporal redundancy stands for the inter-frame similarities between two frames of a video stream.

One can approach the video compression problem as a generalized version of image compression. In such solutions every frame of the video sequence is treated as an image and is compressed using an image compression algorithm. Methods such as run-length coding, Lempel-Ziv coding [75, 76], and Huffman coding [77] employ the statistical redundancy of an image by assigning fewer bits to frequently occurring

gray levels or color values [74]. The so-called spatial redundancy is used in predictive coding [78] and transform coding. Among the transform coding methods are block DCT-based coding and discrete wavelet transform (DWT)-based coding [79, 80]. Other image coding schemes including vector quantization [81] and fractal image compression [82] can also be used for the purpose of video compression.

The video sequence can be considered as a three dimensional signal as well. In a slow motion video most neighboring frames appear to be similar and two frames would be significantly different from each other, only if camera or objects suddenly move. Therefore, the temporal redundancy can be exploited to help predicting future frames from previous frames. One of the ways to benefit from this feature is through motion estimation methods. Motion estimation methods can be classified into block matching and pel-recursive techniques [83]. The method used in this chapter is the block matching algorithm.

The issue of CS-based video compression has been addressed in a number of articles [84–87]. In [84], every frame of the video signal is treated as a single image with the DWT being the sparsity basis. The sparsity basis in [85] is the motion compensated wavelet basis of [88]. The motion estimation algorithm used in this method is the block matching algorithm. The method in [86] benefits from distributed video coding. Even frames are coded with the intra-frame coder of [89]. Odd frames are divided into non-overlapping sub-blocks and each sub-block is sampled with a CS measurement matrix. The dictionary used at the decoder side is an adaptive and overcomplete dictionary unlike usual CS applications where a fixed orthogonal basis is used. The method proposed in [87] classifies the frames into two groups of reference and non-reference frames. Reference frames are fully sampled and divided into non-overlapping sub-blocks. Based on the distribution of the DCT coefficients in each sub-block, they are furthermore classified into two groups of regular and sparse. In the upcoming non-reference frames, the regular sub-blocks are always fully sampled and compressed samples are collected from sparse sub-blocks.

In the methods of [86] and [87], tiling of the video frames and then classification of the resulting sub-blocks is performed. Different classes of sub-blocks are processed differently. Therefore, one can expect different qualities for different sub-blocks in the recovered frame. This quality mismatch draws attention especially in the cases when a relatively high compression ratio is used for sparse sub-blocks. We are also required to provide the receiver with the extra information about which sub-blocks

are spare and which are regular. Extra communication is also necessary for the method in [86] because of the feedback channel from the decoder to encoder as well as the classification of the sub-blocks.

Note that all of these methods need to either fully sample a whole frame or parts of a frame. The one in [85] collects compressive measurements of the wavelet transformed and linear restricted version of the video frames. Therefore, it needs the fully sampled frames to calculate the wavelet transform. In [86], even frames are fully sampled, and in [87], the reference frames as well as the regular sub-blocks in non-reference frames are fully sampled.

A CS-based video compression method exploiting the temporal redundancy of a video is introduced in this section. In general, every other l frame of the video file is selected as a reference frame. The rest of the frames are non-reference frames, which will be recovered from the compressed data based on their correlation with the reference frames. The DWT domain or the DCT domain is considered as the sparsity basis for reference frames. For the non-reference frames the sparsity basis is the difference domain. As for the recovery algorithm, the l_1-norm minimization is used in a parallel manner which helps reduce the compression and decompression time and also simplifies the hardware design. Also a variation of the method exploiting block matching motion estimation algorithm is presented which has a better performance over the original method in the case of fast motion video. The drawback of the motion estimation enabled version of the algorithm is that the motion vectors should be transmitted to the receiver side as well. The original method, however, does not require any extra communication. The results of this section are partly presented in [90].

5.2.1 CS-based video compression

Modeling of the video signal

The video sequence is divided into disjoint blocks, where each block contains l consecutive video frames. These l frames are processed together. The first video frame of each block is considered as the reference frame. The sparsity basis for reference frames is the DCT or DWT basis. The rest of the frames (non-reference frames) are represented by their differences with the previous reference frames. Depending on the rate of change in video frames, one can consider every other two, three or even more video frames as reference frames and recover the rest of the frames using

their pixel-wise difference with the closest previous reference frame. It has been observed that the difference between two consecutive frames is sparser than the DCT or DWT of the non-reference frame in the case of slow motion videos. Therefore, the less frequent we choose a reference frame a higher compression rate can be achieved. Although this higher compression rate may come at the expense of image quality.

Let $\boldsymbol{F}_i$ denote the $N \times N$ image corresponding to the ith frame of a video sequence. Let the first frame of each block be the reference frame and

$$\boldsymbol{E}_i = \left(\boldsymbol{F}_{li}^T, \cdots, \boldsymbol{F}_{li+(l-1)}^T\right)^T \tag{5.3}$$

be the $lN \times N$ extended virtual frame which consists of l consecutive frames. The matrix $\boldsymbol{E}_i$ serves as the data matrix of the parallel CS method.

CS-based video compression algorithm

The first step in the algorithm is to find the sparse representation of $\boldsymbol{E}_i$, that is $\boldsymbol{E}_i^s$, in a suitable sparsity basis $\boldsymbol{\Psi}$. For the reference frame the DWT or DCT domain is used as the sparsity basis. The sparsity basis for the non-reference frames is their pixel-wise difference with the reference frame, $\boldsymbol{F}_j - \boldsymbol{F}_{li}, \quad j = li+1, \ldots, li+(l-1)$. Mathematically speaking we have

$$\boldsymbol{E}_i^s = \boldsymbol{\Psi}\boldsymbol{E}_i, \quad \boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{T} & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} & \ldots & \boldsymbol{0} \\ \vdots & & \vdots & & \vdots \\ -\boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \tag{5.4}$$

where $\boldsymbol{T}$, $\boldsymbol{0}$, and $\boldsymbol{I}$ are the $N \times N$ DCT or DWT matrix, the all zero matrix, and the identity matrix, respectively. The DCT matrix is,

$$\boldsymbol{T}_{DCT} = (t_{ij})_{N \times N} \tag{5.5}$$

with entries $t_{1j} = 1/\sqrt{N}, \quad j = 1, \ldots, N$ and $t_{ij} = \sqrt{\frac{2}{N}} \cos \frac{\pi(2i-1)(j-1)}{2N}, \quad i \neq 1, j = 1, \ldots, N$. Assuming that Haar wavelets are used, with an even N the DWT matrix

is,

$$\boldsymbol{T}_{DWT} = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & \dots & 0 & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0.5 & 0.5 \\ 0.5 & -0.5 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 & \dots & 0 & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0.5 & -0.5 \end{pmatrix}_{N \times N} . \tag{5.6}$$

Applying the matrix $\boldsymbol{T}_{DWT}$ to an $N \times 1$ data vector, the first half entries of the result are called averages or approximate coefficients. The other half of the elements are called detail coefficients.

To ensure that all columns of $\boldsymbol{E}_i^s$ have the same sparsity level, different random permutations are applied to the rows of $\boldsymbol{E}_i^s$ to build a new matrix $\boldsymbol{E}_i'$. Proof of the legitimacy of this assumption is given in Lemma 5.1. Finally, the $K' \times lN$ measurement matrix $\boldsymbol{\Phi}'$ can be applied to $\boldsymbol{E}_i'$ in order to obtain the $K' \times N$ matrix $\boldsymbol{Y}_i$ of compressed samples

$$\boldsymbol{Y}_i = \boldsymbol{\Phi}' \boldsymbol{E}_i'. \tag{5.7}$$

Lemma 5.1. *Applying different random permutations on the rows of $\boldsymbol{E}_i^s$ leads to the same average sparsity level for each column of $\boldsymbol{E}_i'$.*

Proof. Let $N_i, i = 1, \dots, lN$ be the number of nonzero coefficients in $\boldsymbol{v}_i$ which is the ith row of $\boldsymbol{E}_i^s$. Let $\boldsymbol{p}_i$ be the $1 \times N_i$ vector corresponding to the indices of the nonzero coefficients of $\boldsymbol{v}_i$. There is a total of $N!$ different permutations that can be applied to $\boldsymbol{v}_i$. With a random permutation from the $N!$ possibilities, the entry $\left[\boldsymbol{v}_i\right]_{[\boldsymbol{p}_i]_j}, j = 1, \dots, N_i$ is permuted to the columns $1, 2, \dots, N$ with the equal probability of $\frac{1}{N}$. Therefore, considering the permuted ith row, $\boldsymbol{v}_i'$, the average number of nonzero coefficients in every column is $\frac{N_i}{N}$. Assuming the permutations applied to different rows are independent from each other, the average number of nonzero coefficients in each column of $\boldsymbol{E}_i'$ is $\sum_{i=1}^{lN} \frac{N_i}{N}$. Therefore, in average, every column of $\boldsymbol{E}_i'$ has the same sparsity level. □

Note that the compression hardware can be built in a parallel manner such that the compressed samples of all columns of matrix $\boldsymbol{E}_i'$ can be obtained at the same time. To provide compression ratio versus peak signal to noise ratio (PSNR) results we also quantize the collected measurements. First, the dynamic range of the samples is calculated and then a uniform q-level quantization is performed. The

Algorithm 5.1 The Proposed Video Compression Algorithm

Input: The block of l consecutive frames $\boldsymbol{F}_{li}$ up to $\boldsymbol{F}_{li+(l-1)}$.
Output: The compressed extended virtual frame $\boldsymbol{Y}_i$.
1. Form the data matrix:
$\boldsymbol{E}_i \Leftarrow \left(\boldsymbol{F}_{li}^T, \boldsymbol{F}_{li+1}^T, \cdots, \boldsymbol{F}_{li+(l-1)}^T\right)^T$.
2. Find the sparse representation:
$\boldsymbol{E}_i^s \Leftarrow \boldsymbol{\Psi}\boldsymbol{E}_i$ where $\boldsymbol{\Psi}$ is defined as in (5.4).
3. Force the same sparsity level for all columns of $\boldsymbol{E}_i^s$:
for $n = 1$ to lN **do**
$\left[\boldsymbol{E}_i'\right]_{n,:} \Leftarrow \boldsymbol{P}_k\left[\boldsymbol{E}_i^s\right]_{n,:}$
where $\boldsymbol{P}_n$ is a random permutation matrix and $\left[\boldsymbol{E}_i'\right]_{n,:}$ denotes the nth row of matrix $\boldsymbol{E}_i'$.
end for
4. Obtain the compressed extended virtual frame:
$\boldsymbol{Y}_i = \boldsymbol{\Phi}\boldsymbol{E}_i'$.
5. Quantize $\boldsymbol{Y}_i$.

following lemma derives the compression ratio achieved in this method assuming that the gray levels are represented with 8 bits in the original frames.

Lemma 5.2. *The compression ratio of the aforementioned CS-based video compression method is* $1 - \frac{K' \log_2 q}{8lN}$.

Proof. Since a q-level quantizer is used, the number of bits required to represent every quantized symbol is $\log_2 q$. Here, $N_c = K' \log_2 q$ is the number of bits required to represent the compressed samples taken from each column. Since the gray levels are represented with 8 bits, the number of bits representing each column of the original frame is $N_o = 8lN$. Therefore, the compression ratio attained by this method is

$$CR = \frac{N_o - Nc}{N_o} = \frac{8lN - K' \times \log_2 q}{8lN} = 1 - \frac{K' \log_2 q}{8lN}. \tag{5.8}$$

□

Algorithm 5.1 summarizes the proposed video compression method.

CS-based video decompression algorithm

In the recovery stage, every column of $\boldsymbol{E}_i'$ is recovered by solving the following l_1-norm minimization problem

$$\min \|\left[\hat{\boldsymbol{E}}_i'\right]_{n,:}\|_{l_1}, \quad \text{subject to} \quad \left[\boldsymbol{Y}_i\right]_{n,:} = \boldsymbol{\Phi}\left[\hat{\boldsymbol{E}}_i'\right]_{n,:}. \tag{5.9}$$

Algorithm 5.2 The Proposed Video Decompression Algorithm

Input: The compressed extended virtual frame $\boldsymbol{Y}_i$.
Output: The estimated l-frame video sequence $\hat{\boldsymbol{F}}_{li}$ up to $\hat{\boldsymbol{F}}_{li+(l-1)}$.
1. Parallel recovery:
for $n = 1$ to N **do**
$\left[\hat{\boldsymbol{E}}_i'\right]_{:,n} \Leftarrow$ solve problem (5.9).
end for
2. Apply the inverse row permutations:
for $n = 1$ to lN **do**
$\left[\hat{\boldsymbol{E}}_i^s\right]_{n,:} \Leftarrow \boldsymbol{P}_n^{-1}\left[\hat{\boldsymbol{E}}_i'\right]_{n,:}$.
end for
3. Retrieve the estimated frames:
$\left(\hat{\boldsymbol{F}}_{li}^T, \hat{\boldsymbol{F}}_{li+1}^T, \cdots, \hat{\boldsymbol{F}}_{li+(l-1)}^T\right)^T \Leftarrow \hat{\boldsymbol{E}}_i$ where $\hat{\boldsymbol{E}}_i = \boldsymbol{\Psi}^{-1}\hat{\boldsymbol{E}}_i^s$.

Then the inverse row permutations are applied to $\hat{\boldsymbol{E}}_i'$ in order to obtain matrix $\hat{\boldsymbol{E}}_i^s$. Finally, the inverse DWT or DCT is applied to that part of the matrix $\hat{\boldsymbol{E}}_i^s$ corresponding to the reference frame $\boldsymbol{F}_{li}$. Let $\hat{\boldsymbol{F}}_{li}$ denote the reconstructed reference frame. Since the difference domain is chosen as the sparsity basis, the estimates $\hat{\boldsymbol{F}}_{li+1}, \ldots, \hat{\boldsymbol{F}}_{li+(l-1)}$ of the non-reference frames are calculated by a simple pixel by pixel addition of $\hat{\boldsymbol{F}}_{li}$ to the corresponding parts of $\hat{\boldsymbol{E}}_i^s$. Algorithm 5.2 summarizes the proposed video recovery method for the case of the difference domain being the sparsity basis. Note that the proposed recovery algorithm can also be implemented in parallel manner where all columns of $\boldsymbol{E}_i'$ can be recovered at the same time.

5.2.2 Motion estimation enabled CS-based video compression

Compression algorithm

Use of the difference domain as the sparsity basis in slow motion videos results in good quality outcomes, but the quality of the recovered video starts to deteriorate when the objects in the video start to move faster. One way to deal with this problem is to introduce a variable rate compression method. Such method would take a different number of samples based on how fast the objects move in consecutive video frames. The other way is to choose a sparsity basis other than the difference domain that can account for the movement of objects. We propose a variation of the compression algorithm introduced before by exploiting the motion compensated difference of the non-reference and reference frames.

The motion estimation method in use is the block matching algorithm. The non-reference video frames are first divided into non-overlapping sub-blocks. Then,

the algorithm searches in the p-pixel vicinity of a sub-block to find a sub-block of the same dimension in the previous reference frame that resembles it the most. The amount of offset in the position of the chosen sub-block in the reference frame is recorded as the motion vector calculated for that specific sub-block. The algorithm also records the amount of error between the original sub-block and the candidate sub-block of the reference frame. For the fast motion videos this error signal is much more sparse than the pixel by pixel difference of the frames. The only drawback of this method is that the motion vectors are also needed to be transmitted along with the compressed measurements. For a 256×256 video frame, the number of motion vectors are 256 considering a typical size 16×16 sub-block used in MPEG.

The extended virtual frame $\boldsymbol{E}_i$ is defined as before. The part corresponding to the reference frame is replaced by the DWT or the DCT of $\boldsymbol{F}_{li}$ and the rest of the frames are replaced by $\boldsymbol{F}_j(\boldsymbol{v}_{j,i}) - \boldsymbol{F}_{li}, \quad j = li+1, \ldots, li+(l-1)$, where $\boldsymbol{F}_j(\boldsymbol{v}_{j,i})$ is frame $\boldsymbol{F}_j$ with its sub-blocks displaced with the calculated motion vectors $\boldsymbol{v}_{j,i}$ with respect to the reference frame $\boldsymbol{F}_{li}$. We can rewrite equation (5.4) as

$$\boldsymbol{E}_i^s = \boldsymbol{\Psi}(\boldsymbol{v}_i)\boldsymbol{E}_i \tag{5.10}$$

with $\boldsymbol{v}_i$ being the set of all motion vectors for the block of l frames $\boldsymbol{F}_{li}, \ldots, \boldsymbol{F}_{li+(l-1)}$. The signal $\boldsymbol{E}_i^s$ would undergo row permutations and then compressed samples are drawn based on equation (5.7).

The common criteria for selecting a candidate motion vector for a specific sub-block are mean absolute difference (MAD) and MSE. The block matching algorithm chooses the motion vector that leads to either the minimum MAD or minimum MSE. The optimum criterion is the one that leads to the most sparse error signal. Therefore, the criterion that is used in the motion estimation enabled version of the compression algorithm is the l_0-pseudo-norm of the error signal. This way the block matching algorithm issues the set of motion vectors that lead to an error image with the minimum l_0-pseudo-norm. Algorithm 5.3 explains the motion estimation enabled compression algorithm.

Decompression algorithm

The primary steps of this algorithm are the same as the CS-based video decompression described earlier. An estimate $\hat{\boldsymbol{E}}_i'$ is obtained solving the problem in (5.9) for every column of $\boldsymbol{E}_i'$. The inverse row permutations are then used to obtain the sparse

Algorithm 5.3 The Motion Estimation Enabled Variant of the Proposed Video Compression Algorithm

Input: The block of l consecutive frames $\boldsymbol{F}_{li}$ up to $\boldsymbol{F}_{li+(l-1)}$.
Output: The compressed extended virtual frame $\boldsymbol{Y}_i$ and the set of motion vectors $\boldsymbol{v}_i$.
1. Form the data matrix:
$\boldsymbol{E}_i \Leftarrow \left(\boldsymbol{F}_{li}^T, \boldsymbol{F}_{li+1}^T, \cdots, \boldsymbol{F}_{li+(l-1)}^T\right)^T$.
2. Calculate the motion vectors $\boldsymbol{v}_i$ and find the sparse representation:
$\boldsymbol{E}_i^s \Leftarrow \boldsymbol{\Psi}(\boldsymbol{v}_i)\boldsymbol{E}_i$.
3. Force the same sparsity level for all columns of $\boldsymbol{E}_i^s$:
for $n = 1$ to lN **do**
$\left[\boldsymbol{E}_i'\right]_{n,:} \Leftarrow \boldsymbol{P}_n\left[\boldsymbol{E}_i^s\right]_{n,:}$
where $\boldsymbol{P}_n$ is a random permutation matrix and $\left[\boldsymbol{E}_i'\right]_{n,:}$ denotes the nth row of matrix $\boldsymbol{E}_i'$.
end for
4. Obtain the compressed extended virtual frame:
$\boldsymbol{Y}_i = \boldsymbol{\Phi}\boldsymbol{E}_i'$.
5. Quantize $\boldsymbol{Y}_i$.

matrix $\hat{\boldsymbol{E}}_i^s$. To recover an estimate $\hat{\boldsymbol{F}}_{li}$ of the reference frame, the inverse DWT or DCT is applied to the sub-matrix comprising of the first N rows of the matrix $\hat{\boldsymbol{E}}_i^s$. Since motion estimation is adopted for this method, the estimated reference frame $\hat{\boldsymbol{F}}_{li}$ is first displaced by the motion vectors $\boldsymbol{v}_i$, and then the non-reference frames are recovered using the corresponding error image and the displaced version of $\hat{\boldsymbol{F}}_{li}$. Algorithm 5.4 summarizes the explained decompression algorithm. Note that the proposed recovery algorithm can also be implemented in parallel where all columns of $\boldsymbol{E}_i'$ can be recovered at the same time.

5.2.3 Simulation results

The proposed video compression method is applied to the standard slow motion *Akiyo* and fast motion *Coastguard* videos. The results for different choices of sparsity bases, i.e., DWT or DCT are provided. PSNR is used as a quality measure for comparison between different sparsity bases or sampling methods.[1] The PSNR is defined as

$$PSNR = 20\ \log_{10}\left(\frac{max_{\boldsymbol{I}}}{\sqrt{MSE}}\right) \tag{5.11}$$

[1]Note that the PSNR is most commonly used as a quality measure of reconstruction methods, especially in image compression [74].

Algorithm 5.4 The Proposed Motion Estimation Enabled Video Decompression Algorithm

Input: The compressed extended virtual frame $\boldsymbol{Y}_i$ and the motion vectors $\boldsymbol{v}_i$.
Output: The estimated l-frame video sequence $\hat{\boldsymbol{F}}_{li}$ up to $\hat{\boldsymbol{F}}_{li+(l-1)}$.
1. Parallel recovery:
for $n = 1$ to N **do**
$\left[\hat{\boldsymbol{E}}_i^{'}\right]_{:,n} \Leftarrow$ solve problem (5.9).
end for
2. Apply the inverse row permutations:
for $n = 1$ to lN **do**
$\left[\hat{\boldsymbol{E}}_i^{s}\right]_{n,:} \Leftarrow \boldsymbol{P}_n^{-1}\left[\hat{\boldsymbol{E}}_i^{'}\right]_{n,:}$.
end for
3. Retrieve the estimated frames:
$\left(\hat{\boldsymbol{F}}_{li}^T, \hat{\boldsymbol{F}}_{li+1}^T, \cdots, \hat{\boldsymbol{F}}_{li+(l-1)}^T\right)^T \Leftarrow \hat{\boldsymbol{E}}_i$, where $\hat{\boldsymbol{E}}_i = \boldsymbol{\Psi}(\boldsymbol{v}_i)^{-1}\hat{\boldsymbol{E}}_i^s$.

where $max_{\boldsymbol{I}}$ is the maximum value of the image[2] $\boldsymbol{I}(k,j)$ and the MSE between the original $\boldsymbol{I}(k,j)$ and reconstructed $\boldsymbol{I}_R(k,j)$ images is defined as

$$MSE = (1/mn)\sum_{k=1}^{m}\sum_{j=1}^{n}\|\boldsymbol{I}(k,j) - \boldsymbol{I}_R(k,j)\|^2. \tag{5.12}$$

Based on the fact that images are more of a low pass nature, the wavelet approximation coefficients have more significant values than the vertical, horizontal, and diagonal detail coefficients. Therefore, it is expected that the second level wavelet decomposition used as a sparsity basis would result in reconstructed frames of better quality. Moreover, if the camera and the objects do not move suddenly, the consecutive frames must be similar to each other and the difference between these frames is likely to be sparser than the representation of one of them in the wavelet domain. Therefore, considering every other three or four frame as the reference frame may result in better PSNR values. In the case of fast motion videos, the pixel by pixel difference of the frames is not as sparse as the slow motion videos. In such cases, the error frame of the motion estimation methods are much sparser than the difference. Therefore, it is expected that using the motion estimation enabled version of the algorithm would result in a noticeable performance improvement while dealing with fast motion videos. The same is true for slow motion videos as well, but the improvement is not as much as the fast motion videos. The following simulation results confirm all of the aforementioned intuitions.

[2]Note that $max_{\boldsymbol{I}}$ is usually replaced by 255 in image processing when the image takes luminance values from 0 to 255.

Two or four frames of the *Akiyo* and *Coastguard* videos both starting from frame 65 are considered. Table 5.2 shows the PSNR values for different scenarios, where R_s is the ratio of the number of samples taken from each column of a single frame to the number of entries in each column. In this table "+ME" under "Method" means that the motion estimation enabled version of the algorithm is applied. Note that the PSNR values for f_{67} and f_{68} are not provided in the table for the case when every other frame is considered as the reference frame. Entries of the measurement matrix are i.i.d. samples from Bernoulli distribution with $\left(Pr\left(\mathbf{\Phi}_{i,j} = \pm 1/\sqrt{m}\right) = \frac{1}{2}\right)$. Taking a closer look at the PSNRs in Table 5.2 reveals that not only the motion estimation enabled version of the algorithm has better recovery quality, but also it would result in almost the same PSNR for the reference and non-reference frames. As one would expect the gain in the PSNR using the motion estimation enabled version of the algorithm would be more significant in case of the fast motion videos. Results in Table 5.2 confirm this expectation. Fig. 5.1 shows four reconstructed frames of *Akiyo* video where the two level wavelet decompositions are considered for the reference frame. Fig. 5.2 depicts two reconstructed frames of *Coastguard* video with the assumption of two level wavelet as the sparsity basis for the reference frame. The video frames shown in Fig. 5.3 are the recovered frames of *Coastguard* video using the motion estimation enabled version of the algorithm with the two level wavelet domain being the sparsity basis for the reference frame. Note that the results in Table 5.2, and Figs. 5.1, 5.2, and 5.3 are obtained from the unquantized compressed frames.

Fig. 5.4 shows the PSNR versus compression ratio for the two video sequences considered. The sparsity basis considered for the reference frames is the two-level DWT domain. In this case compressed frames are being quantized to have exact values for the compression ratio.

To further compare the algorithms proposed in this chapter with the one in [87], we apply these algorithms to the same experimental videos and we compare the results. The slow motion video used in [87] is *Akiyo* and for the fast motion video *Stefan* is used. Table 5.3 compares the recovery PSNRs for the proposed algorithms with the results in [87]. The results provided are for the 217th frame of *Akiyo* and the 100th frame of *Stefan*. In both cases the previous frame is considered as the reference frame. The highest PSNR values for each video is shown in bold face. It can be seen that with the proposed motion estimation enabled algorithm, we can

Table 5.2: PSNR values for different sampling scenarios.

	Method	R_s	PSNR (dB)			
			f_{65}	f_{66}	f_{67}	f_{68}
Akiyo	DCT	0.5	40.24 43.15	39.01 41.89	- 41.76	- 41.75
	DWT	0.5	43.20 47.86	40.99 45.32	- 45.02	- 44.97
	2-level DWT	0.5	45.58 47.33	43.64 45.45	- 45.14	- 45.00
	DCT+ME	0.5	40.59 44.09	40.59 44.09	- 44.1	- 44.09
	DWT+ME	0.5	43.44 48.74	43.44 48.74	- 48.74	- 48.75
	2-level DWT+ME	0.5	46.05 48.04	46.04 48.04	- 48.04	- 48.04
Coastguard	DCT	0.625	31.25 30.15	31.02 28.17	- 28.25	- 28.30
	2-level DWT	0.625	33.61 32.26	31.63 30.72	- 30.28	- 30.14
	DCT+ME	0.625	35.15 34.12	35.15 34.13	- 34.22	- 34.21
	2-level DWT+ME	0.625	35.60 34.42	35.60 34.42	- 34.44	- 34.50

reach higher PSNRs with a smaller sampling ratio, R_s.

Fig. 5.5 shows the frames corresponding to the last row of Table 5.3.

(a) Frame 65. (b) Frame 66.

(c) Frame 67. (d) Frame 68.

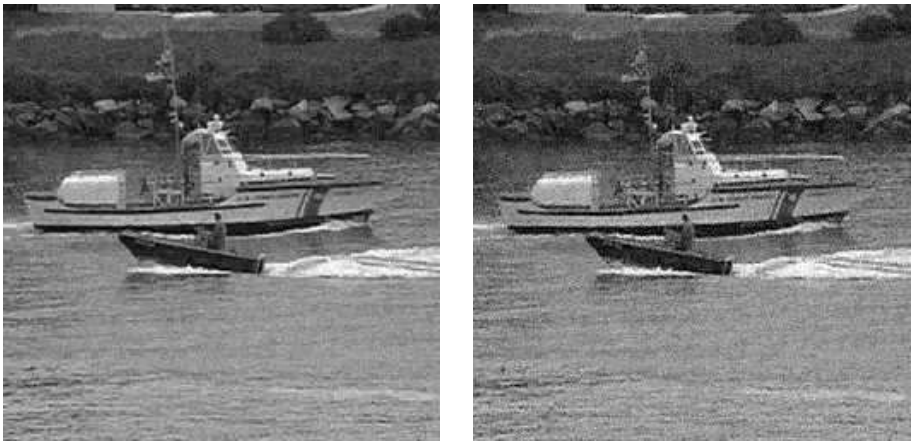
Figure 5.1: Reconstructed *Akiyo* frames.

5.3 Chapter summary

The problem of two dimensional CS was considered and a parallel CS method was developed here. As a sample application of two dimensional CS a video compression method that exploits the similarity between consecutive video frames was given. The method applies the same measurement matrix to every column of the permuted extended virtual frame, which enables us to develop parallel algorithms at both compression and decompression stages that simplifies the video coding hardware. In

Table 5.3: Recovery PSNR comparison of the proposed method with the one in [87].

	Method	R_s	PSNR (dB)	
			Reference frame	Frame in question
Akiyo	[87]	0.6	-	31.94
	[87]	0.89	-	41.42
	DCT	0.5	38.57	37.5
	DWT	0.5	42.21	39.92
	2-level DWT	0.5	44.91	43.35
	DCT+ME	0.5	38.48	38.48
	DWT+ME	0.5	39.57	39.57
	2-level DWT+ME	0.5	45.08	**45.08**
Stefan	[87]	0.78	-	26.06
	[87]	0.89	-	37.58
	2-level DWT	0.75	33.9	31.6
	2-level DWT+ME	0.75	34.55	34.55
	2-level DWT+ME	0.84	38.43	**38.45**



(a) Frame 65. (b) Frame 66.

Figure 5.2: Reconstructed *Coastguard* frames.

(a) Frame 65. (b) Frame 66.

Figure 5.3: Reconstructed *Coastguard* frames from the motion estimation enabled method.

order to further benefit from the aforementioned similarity of the consecutive video frames in a video sequence, a motion estimation enabled variant of the algorithm is also proposed. This version of the compression algorithm improves the quality of the recovered frames, especially when dealing with fast motion videos, at the expense of the extra communication required to transmit the motion vectors. Simulation results demonstrate the effectiveness of the proposed video compression method.

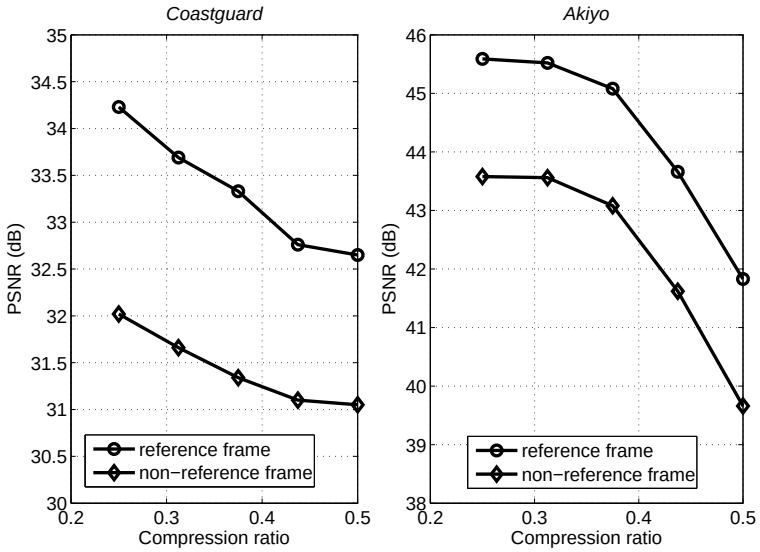
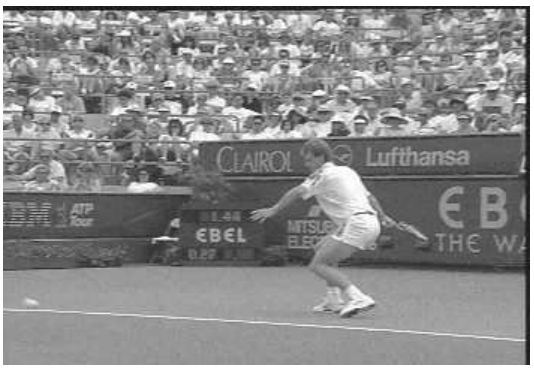
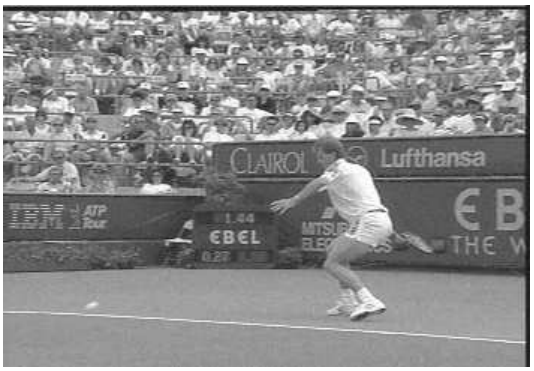


Figure 5.4: PSNR vs compression ratio.



(a) Frame 99.



(b) Frame 100.

Figure 5.5: Reconstructed *Stefan* frames from the motion estimation enabled method.

Chapter 6

Conclusion

A summary of the contributions of the thesis is given in this chapter. We also comment on the future work that can be done regarding different research problems considered.

6.1 Summary of contributions

CS shows great promise in under-sampled signal recovery and has given a lot of attention in the signal processing literature in the past few years. Though various aspects of it from compressive signal acquisition and sampling to sparse signal processing algorithms is still imature. The areas of CS studied in this dissertation include AIC design, sparse channel estimation, and two dimensional CS with application in video compression.

6.1.1 AIC design

We proposed a new segmented CS method for AIC. An analog signal measured by K parallel BMIs is first segmented in time into M segments so that a $K \times M$ matrix of sub-samples is obtained. The sub-samples collected on different segments and different BMIs are reused in a way to collect more samples than the number of BMIs. These samples are correlated to each other over at most one segment. We find the measurement matrix corresponding to this method of sampling and show that it satisfies the RIP if the original measurement matrix of BMI sampling waveforms satisfies it. The extra samples collected help improve the signal recovery performance. Specifically, we proved that the performance of the signal recovery based on the empirical risk minimization improves when the segmented AIC is used for sampling instead of the conventional AIC with the same number of BMIs. In

another setting, if the number of BMIs is insufficient in the conventional AIC to guarantee successful recovery, the proposed segmented AIC supplies the recovery algorithm with additional samples so that successful recovery becomes possible. The complexity increase of the segmented AIC is only due to the $M \leq K$ times higher sampling rate and the necessity to solve a larger size optimization problem at the recovery stage, while the number of BMIs remains the same at the sampling stage. We presented simulation results to justify the effectiveness and superiority of the segmented AIC over conventional AIC.

6.1.2 Sparse channel estimation

We considered the problem of sparse channel estimation. Building on the idea of the ZA-LMS which introduces a penalty term in the cost function of standard LMS, we proposed two algorithms: reweighted l_1-norm penalized and l_p-pseudo-norm penalized LMS. Mathematical analysis of reweighted l_1-norm penalized LMS's mean convergence as well as the attainable excess MSE of the algorithm was presented in the thesis as well. The excess MSE result shows that the reweighted l_1-norm penalized LMS can outperform the standard LMS algorithm. Modified versions of these algorithms were also given for the case of an arbitrary sparsity basis for the CIR. Simulation results for a DCT sparse channel were given along with the time sparse channel example. The performance of the reweighted l_1-norm penalized LMS has been compared to the standard LMS, ZA-LMS, RZA-LMS, and our l_p-pseudo-norm penalized LMS through computer simulations. These results show that reweigthed l_1-norm penalized LMS outperforms standard LMS, ZA-LMS and the RZA-LMS.

We also introduced decimated LMS and ZAD-LMS algorithms for estimation of frequency or DCT sparse CIRs. Decimated LMS and ZAD-LMS have the advantage of being able to accomodate multiple users while having smaller complexity than the standard LMS. However, the performance of decimated LMS degrades with increasing CIR sparsity level. The algorithms have been compared in terms of MSE to the standard LMS and the ZA-LMS. Simulations demonstrating the effectiveness of the proposed decimated LMS methods were also given.

6.1.3 Two dimensional CS

Two dimensional signals are usually vectorized and the a CS measurement matrix of a large size is used to collect the compressed samples. We proposed the idea of

sampling all the columns of a two dimensional signal with the same measurement matrix of a smaller size. This leads to decreased complexity due to the use of a smaller measurement matrix. As an application, we proposed a CS-based video compression method that exploits the similarity between consecutive video frames by assuming the pixel-wise difference of the non-refernce and reference frames as the sparsity basis. The method applies the same measurement matrix to every column of the permuted so-called extended virtual frame. We are able to develop parallel algorithms at both compression and decompression stages that significantly simplifies the video coding hardware. In order to further benefit from the aforementioned similarity of the consecutive video frames in a video sequence, a motion estimation enabled variant of the algorithm is also proposed. This variant of the compression algorithm improves the quality of the recovered frames, especially when dealing with fast motion videos, at the expense of extra communication required to transmit the motion vectors.

6.2 Probable future research

6.2.1 Theoretical analysis of segmented AIC for l_1-norm minimization based recovery

The mathimatical analysis of the segmented CS method with empirical risk minimization as the recovery algorithm is given in this thesis. For l_1-norm minimization based recovery, however, this is more challenging since the rows of the measurement matrix are correlated. The problem comes down to finding the extremal eigenvalues of a correlated random matrix.

6.2.2 Quantized segmented AIC

In every real world system, the measurements are stored with finite precision and this is the case for AIC as well. As an example, the idea of one bit CS was introduced in [91] where only the sign of the measurements are retained. This method recovers the signal on the unit sphere within a scaling factor. It is necessary to study the performance of our segmented AIC when the compressed samples are quantized.

6.2.3 Sparse RLS and Kalman filter algorithms

Parameter estimation algorithms such as RLS and Kalaman filter can also be modified when the signal of interest is sparse. Several modifications of these algorithms

have been presented in the literature. The ideas of reweighted l_1-norm penalized, l_p-pseudo-norm penalized LMS, decimated LMS and ZAD-LMS can also be applied to RLS and Kalman filter. A mathematical analysis of the performance improvement of these algorithms over the standard RLS and Kalman filter can also be done.

6.2.4 Direct two dimensional CS for an arbitrary sparsity basis

The two dimensional CS method presented in this thesis first transfers the signal to the sparsity basis and then a permutation is applied to make sure that the columns are of even sparsity level. This means that the signal is first sampled with the *Nyquist rate* and then compressed. The setup described in the thesis forces us to do so in order to be able to run a parallel recovery algorithm. It might be possible to change the recovery problem in a way that the need of finding the sparse representation of the signal is rectified. In this way, there will be no need to sample the signal with a high rate and compressed samples can be collected in one step which leads to simplification of the sampler.

Bibliography

[1] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[2] O. Taheri and S. A. Vorobyov, "Segmented compressed sampling for analog-to-information conversion: Method and performance analysis," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 554–572, Feb. 2011.

[3] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice Hall, New Jersey, 2010.

[4] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 146–181, Mar. 1974.

[5] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, Apr. 1975.

[6] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 2002.

[7] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications*, Technology Press and Wiley, New York, 1949.

[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 34–45, Mar. 1960.

[9] L. Ljung, *System Identification–Theory For the User*, Prentice Hall, New Jersey, 1999.

[10] J. Schoukens, T. Dobrowiecki, and R. Pintelon, "Parametric and nonparametric identification of linear systems in the presence of nonlinear distortions: Frequency domain approach," *IEEE Transactions on Automatic Control*, vol. 43, no. 2, pp. 176–190, Feb. 1998.

[11] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation for coded modulation of intersymbol interference channels," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 989–995, Aug. 1989.

[12] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Transactions on Broadcasting*, vol. 48, no. 3, pp. 223–229, Sept. 2002.

[13] S. U. H. Qureshi, "Adaptive equalization," *Proceedings of the IEEE*, vol. 73, no. 9, pp. 1349–1387, Sept. 1985.

[14] R. W. Lucky, "Automatic equalization for digital communication," *Bell System Technical Journal*, vol. 44, pp. 547–588, Apr. 1965.

[15] R. W. Lucky, "Techniques for adaptive equalization for digital communication," *Bell System Technical Journal*, vol. 45, pp. 255–286, Feb. 1966.

[16] R. W. Lucky, "A survey of the communication theory literature: 1968-1973," *IEEE Transactions on Information Theory*, vol. 19, no. 6, pp. 725–739, Nov. 1973.

[17] G. D. Forney, "Maximum liklihood sequence estiamtion of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 363–378, May 1972.

[18] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[19] F. R. Magee and J. G. Proakis, "Adaptive maximum likelihood sequence estimation for digital signaling in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 120–124, Jan. 1973.

[20] C. A. Belfiore and J. H. Park, "Decision feedback equalization," *Proceedings of the IEEE*, vol. 67, pp. 1143–1156, Aug. 1979.

[21] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *Journal of the Acoustical Society of America*, vol. 50, no. 2B, pp. 637–655, 1971.

[22] T. Bohlin, "Comparison of two methods of modeling stationary EEG singals," *IBM Journal of Research and Development*, vol. 17, no. 3, pp. 194–205, May 1973.

[23] W. Gersch, "Spectral analysis of EEGs by autoregressive decomposition of time series," *Mathematical Biosciences*, vol. 7, pp. 205–222, Feb. 1970.

[24] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hean, J. R. Zeidler, E. Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975.

[25] M. Sondhi, "An adaptive echo canceller," *Bell System Technical Journal*, vol. 46, pp. 497–511, Mar. 1967.

[26] H. M. Ibrahim, R. R. Gharieb, and M. M. Hassan, "A higher-order statistics-based adaptive algorithm for line enhancement," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 527–532, Feb. 1999.

[27] T. Kwan and K. Martin, "Adaptive detection and enhancement of multiple sinusoids using cascade IIR filter," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 7, pp. 937–947, Jul. 1989.

[28] J. Li and P. Stoica, *MIMO Radar Signal Processing*, Wiley-IEEE Press, Oct. 2008.

[29] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 1, pp. 27–34, Jan. 1982.

[30] S. A. Vorobyov, A. B. Gershman, and Z. Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 313–324, Feb. 2003.

[31] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, 1985.

[32] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.

[33] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[34] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[35] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, Sept. 2006.

[36] M. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in *Proc. IEEE International Conference on Image Processing, ICIP'09*, Atlanta, USA, Oct. 2006, pp. 1273–1276.

[37] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Joint source-channel communication for distributed estimation in sensor networks," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3629–3653, Oct. 2007.

[38] Z. Yu, S. Hoyos, and B. M. Sadler, "Mixed-signal parallel compressed sensing and reception for cognitive radio," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'08*, Las Vegas, USA, Apr. 2008, pp. 3861–3864.

[39] W. U. Bajwa, J. Haupt, G. Raz, and R. Nowak, "Compressed channel sensing," in *Proc. IEEE Conference on Information Sciences and Systems, CISS'08*, Princeton, USA, Mar. 2008, pp. 5–10.

[40] G. Taubock and F. Hlawatsch, "A compressed sensing technique for OFDM channel estimation in mobile environments: Exploiting channel sparsity for reducing pilots," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'08*, Las Vegas, USA, Apr. 2008, pp. 2885–2888.

[41] Y. M. Lu and M. N. Do, "A theory for sampling signals from a union of subspaces," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2334–2345, Jun. 2008.

[42] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[43] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *Proc. IEEE International Symposium on Circuits and Systems, ISCAS'07*, New Orleans, USA, May 2007, pp. 1959–1962.

[44] O. Taheri and S. A. Vorobyov, "Segmented compressed sampling for analog-to-information conversion," in *Proc. IEEE Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP'10*, Aruba, Dutch Antilles, Dec. 2010, pp. 113–116.

[45] W. Badjwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proc. IEEE Statistical Signal Processing Workshop, SSP'07*, Madison, USA, Aug. 2007, pp. 294–298.

[46] R. Baraniuk and P. Steeghs, "Compressive radar imaging," in *Proc. IEEE Radar Conference, RadarCon'07*, Waltham, USA, Apr. 2007, pp. 128–133.

[47] O. Taheri and S. A. Vorobyov, "Empirical risk minimization-based analysis of segmented compressed sampling," in *Proc. 44th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, USA, Nov. 2010, pp. 233–235

[48] D. L. Donoho and J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lowers dimension," *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 1–53, Jan. 2009.

[49] E. J. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[50] R. G. Baraniuk, M. Davenport, R. De Vore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Dec. 2008.

[51] D. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 797–829, Jun. 2006.

[52] E. J. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1223, Aug. 2006.

[53] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[54] D. Angelosante and G. B. Giannakis, "RLS-weighted LASSO for adaptive estimation of sparse signals," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'09*, Taipei, Taiwan, Apr. 2009, pp. 3245–3248.

[55] C. Craig, "On the Tchebycheff inequality of Bernstein," *The Annals of Mathematical Statistics*, vol. 4, no. 2, pp. 94–102, May 1933.

[56] Y. Zhang, "Theory of compressive sensing via l_1-minimization: A non-RIP analysis and extensions," *Rice University CAAM Technical Report TR08–11*, 2008. Available [Online]: http://www.caam.rice.edu/~yzhang/reports/tr0811.pdf

[57] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'09*, Taipei, Taiwan, Apr. 2009, pp. 3125–3128.

[58] Y. Gu, J. Jin, and S. Mei, "l_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, Sept. 2009.

[59] G. Su, J. Jin, Y. Gu, and J. Wang, "Performance analysis of l_0 norm constraint least mean square algorithm," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 2223–2235, May 2012.

[60] K. Shi and X. Ma, "Transform domain LMS for sparse system identification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'10*, Dallas, USA, Mar. 2010, pp. 3714–3717.

[61] J. Yang and G. E. Sobelman, "Sparse LMS with segment zero attractors for adaptive estimation of sparse signals," in *Proc. IEEE Asia Pacific Conference on Circuits and Systems, APCCAS'10*, Kuala Lumpur, Malaysia, Dec. 2010, pp. 422–425.

[62] N. J. Bershad and A. Bist, "Fast coupled adaptation for sparse impulse responses using a partial Haar transform," *IEEE Transactions on Signal Processing*, vol. 53, no. 3, pp. 966–976, Mar. 2005.

[63] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson, "Exploiting sparsity in adaptive filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1883–1894, Aug. 2002.

[64] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the l_1-norm," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3436–3447, July 2010.

[65] B. Dumitrescu, A. Onose, P. Helin, and I. Tabus, "Greedy sparse RLS," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2194–2207, May 2012.

[66] N. Vaswani, "Kalman filtered compressed sensing," in *Proc. IEEE International Conference on Image Processing, ICIP'08*, San Diego, USA, Oct. 2008, pp. 893–896.

[67] O. Taheri and S. A. Vorobyov, "Sparse channel estimation with l_p-norm and reweighted l_1-norm penalized least mean squares," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'11*, Prague, Czech Republic, May 2011, pp. 2864–2867.

[68] B. Liu and M. D. Sacchi, "Minimum weighted norm interpolation of seismic records," *Geophysics*, vol. 69, no. 6, pp. 1560–1568, Nov./Dec. 2004.

[69] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[70] S. C. Douglas and W. Pan, "Exact expectation analysis of the LMS adaptive filter," *IEEE Transactions on Signal Processing*, vol. 43, no. 12 , pp. 2863–2871, Dec. 1995.

[71] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. 29, no. 3, pp. 722–736, June 1981.

[72] O. Taheri and S. A. Vorobyov, "Decimated least mean squares for frequency sparse channel estimation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'12*, Kyoto, Japan, Mar. 2012, pp. 3181–3184.

[73] O. Taheri and S. A. Vorobyov, "Reweighted l_1-norm penalized LMS for sparse channel estimation and its analysis ," Submitted to *IEEE Transactions on Signal Processing*.

[74] M. K. Mandal, *Multimedia Signals and Systems,* Kulwer Academic Publishers, 2002.

[75] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[76] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, Sept. 1978.

[77] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proceedings of the Institute of Radio Engineers*, vol. 40, no. 9, pp. 1098–1101, Sept. 1952.

[78] J. Jiang, E. A. Edirisinghe, and H. Schroder, "A novel predictive coding algorithm for 3-D image compression," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 3, pp. 430–437, Aug. 1997.

[79] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3455–3462, Dec. 1993.

[80] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[81] R. M. Gray, "Vector quantization," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 1, no. 2, pp. 4–29, Apr. 1984.

[82] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 18–30, Jan. 1992.

[83] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 858–876, June 1995.

[84] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. of Picture Coding Symposium, PCS'06*, Beijing, China, Apr. 2006.

[85] J. Y. Park and M. B. Wakin, "A multiscale framework for compressive sensing of video," in *Proc. of Picture Coding Symposium, PCS'09*, Chicago, Illinois, USA, May 2009.

[86] J. Prades-Nebot, Y. Ma, and T. Huang, "Distributed video coding using compressive sampling," in *Proc. of Picture Coding Symposium, PCS'09*. Chicago, Illinois, USA, May 2009, pp. 1–4.

[87] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive video sampling," in *Proc. European Signal Processing Conference, EUSIPCO'08*, Lausanne, Switzerland, Aug. 2008.

[88] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1530–1542, Dec. 2003.

[89] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *IEEE Proceedings*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[90] H. Fang, S. A. Vorobyov, H. Jiang, and O. Taheri, "2D Signal Compression via Parallel Compressed Sensing with Permutations," accepted for publication in *Proc. 46th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, USA, Nov. 2012.

[91] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," *in Proc. IEEE Conference on Information Science and Systems, CISS'08*, Princeton, USA, Mar. 2008, pp. 16–21.

Appendix A

Proof of Theorem 3.1

The total number of possible permutations of $\boldsymbol{z}$ is $K!$. Let $\mathcal{A}$ be the set of permutations $\pi_s,\ s = 1, \ldots, |\mathcal{A}|$ that satisfy the following condition

$$\pi_s(k) \neq \pi_t(k), \quad s \neq t,\ \forall s, t \in \{1, \ldots, |\mathcal{A}|\},\ \forall k \in \{1, \ldots, K\}. \tag{A.1}$$

It is easy to see that the number of distinct permutations satisfying the condition (A.1) is K, so $|\mathcal{A}| = K$. It is also straightforward to see that the choice of such K distinct permutations is not unique. As a specific choice, let the elements of $\mathcal{A}$, i.e., the permutations $\pi_s,\ s = 1, \ldots, K$, be

$$\pi_s(k) = ((s + k - 2) \bmod K) + 1, \quad s, k = 1, \ldots, K \tag{A.2}$$

with π_1 being the identity permutation, i.e., the permutations that does not change $\boldsymbol{z}$.

Consider now the matrix $\boldsymbol{Z}$ which consists of M identical columns $\boldsymbol{z}$. The ith set of column permutations of matrix $\boldsymbol{Z}$ is $\mathcal{P}^{(i)} = \{\pi_1^{(i)}, \ldots, \pi_M^{(i)}\}$ and the corresponding permuted matrix is $\boldsymbol{Z}^{\mathcal{P}^{(i)}}$. Let $\{\pi_1^{(i)}, \ldots, \pi_M^{(i)}\}$ be any combination of the K permutations in (A.2). Then there are K^M possible choices for $\mathcal{P}^{(i)}$. However, not all of these possible choices are permissible by the conditions of the theorem.

Indeed, let the set $\mathcal{P}^{(1)}$ be a combination of permutations from $\mathcal{A}$ that satisfies (3.22). There are $I - 1$ other sets $\mathcal{P}^{(i)},\ i = 2, \ldots, I$ which satisfy both (3.22) and (3.23). Gathering all such sets in one set, we obtain the set $\mathcal{P} = \{\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(I)}\}$. Now let $\mathcal{P}^{(I+1)} = [\pi_1^{(I+1)}, \ldots, \pi_M^{(I+1)}]$ be one more set of permutations where $\exists \pi_m^{(I+1)},\ m = 1, \ldots, M$ such that $\pi_m^{(I+1)} \notin \mathcal{A}$. An arbitrary kth row of $\boldsymbol{Z}^{\mathcal{P}^{(I+1)}}$ is $\left([\boldsymbol{Z}^{\mathcal{P}^{(I+1)}}]_{k,1}, \ldots, [\boldsymbol{Z}^{\mathcal{P}^{(I+1)}}]_{k,M}\right)$ where $[\boldsymbol{Z}^{\mathcal{P}^{(I+1)}}]_{k,1}, \ldots, [\boldsymbol{Z}^{\mathcal{P}^{(I+1)}}]_{k,M} \in \{1, \ldots, K\}$. This exact same row can be found as the first row of one of the permuted matrices $\boldsymbol{Z}^{\mathcal{P}^{(i)}}$, $\mathcal{P}^{(i)} \in \mathcal{P}$. Specifically, this is the permuted matrix $\boldsymbol{Z}^{\mathcal{P}^{(i)}}$ that

is obtained by applying the permutations $\mathcal{P}^{(i)} = \left\{\pi_{[\boldsymbol{Z}^{\mathcal{P}^{(I+1)}}]_{k,1}}, \ldots, \pi_{[\boldsymbol{Z}^{\mathcal{P}^{(I+1)}}]_{k,M}}\right\}$. The permutations $\mathcal{P}^{(i)}$ either has to belong to $\mathcal{P}$ or being crossed out from $\mathcal{P}$ because of conflicting with another element $\mathcal{P}^{(l)} \in \mathcal{P}$, $l \neq i$. In both cases, $\mathcal{P}^{(I+1)}$ can not be added to $\mathcal{P}$ because it will contradict the conditions (3.22) and (3.23).

Therefore, the set $\mathcal{P}$ can be built using only the permutations from the set $\mathcal{A}$, i.e., the K permutations in (A.2). Rearranging the rows of $\boldsymbol{Z}^{\mathcal{P}^{(i)}}$ in a certain way, one can force the elements in the first column of $\boldsymbol{Z}^{\mathcal{P}^{(i)}}$ to appear in the original increasing order, i.e., enforce the first column to be equivalent to the vector of indices $\boldsymbol{z}$. It can be done by applying to each permutation in the set $\mathcal{P}^{(i)}$ the inverse permutation $\left(\pi_1^{(i)}\right)^{-1}$, which itself is one of the permutations in (A.2). Therefore, the set $\mathcal{P}^{(i)} = \{\pi_1^{(i)}, \ldots, \pi_M^{(i)}\}$ can be replaced by the equivalent set $\left\{\left(\pi_1^{(i)}\right)^{-1}\pi_1^{(i)}, \ldots, \left(\pi_1^{(i)}\right)^{-1}\pi_M^{(i)}\right\} = \left\{\pi_1, \ldots, \left(\pi_1^{(i)}\right)^{-1}\pi_M^{(i)}\right\}$ where $\left(\pi_1^{(i)}\right)^{-1}\pi_j^{(i)} \in \mathcal{A}$. Hence, the permutations of the form $\mathcal{P}^{(i)} = \{\pi_1, \ldots, \pi_j^{(i)}, \ldots, \pi_M^{(i)}\}$ can only be considered. Since the condition (3.22) requires that $\pi_2^{(i)}$ should be different from π_1, the only available options for the permutations on the second column of $\boldsymbol{Z}$ are the $K-1$ permutations $\pi_2, \ldots, \pi_K$ in (A.2). Therefore, I at most equals $K-1$. Note that I can be smaller than $K-1$ if for some $i \in \{1, \ldots, K-1\}$, $\quad K/gcd(i, K) < M$ (also see Example 3.1 after Theorem 3.1). Thus, in general $I \leq K-1$.

Appendix B

Proof of Lemma 3.1

Let all the rows of $(\mathbf{\Phi}_e)_{\mathcal{T}}$ be partitioned into two sets of sizes (cardinality) as close as possible to each other, where all elements in each set are guaranteed to be statistically independent. In particular, note that the elements of the new K_a rows of $\mathbf{\Phi}_e$ are chosen either from the first $K_a + M - 1$ rows of $\mathbf{\Phi}$ if $K_a + M - 1 < K$ or from the whole matrix $\mathbf{\Phi}$. Therefore, if $K_a + M - 1 < K$, the last $K - K_a - M + 1$ rows of $\mathbf{\Phi}$ play no role whatsoever in the process of extending the measurement matrix and they are independent on the rows of $\mathbf{\Phi}_1$ in (3.28). These rows are called unused rows. Thus, one can freely add any number of such unused rows to the set of rows in $\mathbf{\Phi}_1$ without disrupting its status of being formed by independent Gaussian variables. Since $\min\{K, K_a+M-1\} \leq \lceil (K + K_a)/2 \rceil$, there exist at least $\lfloor (K + K_a)/2 \rfloor - K_a$ unused rows which can be added to the set of rows in $\mathbf{\Phi}_1$. Such process describes how the rows of $(\mathbf{\Phi}_e)_{\mathcal{T}}$ are split into the desired sets $(\mathbf{\Phi}_e)^1_{\mathcal{T}}$ and $(\mathbf{\Phi}_e)^2_{\mathcal{T}}$ of statistically independent elements. As a result, the first matrix $(\mathbf{\Phi}_e)^1_{\mathcal{T}}$ includes the first $\lceil (K + K_a)/2 \rceil$ rows of $(\mathbf{\Phi}_e)_{\mathcal{T}}$, while the rest of the rows are included in $(\mathbf{\Phi}_e)^2_{\mathcal{T}}$.

Since the elements of $(\mathbf{\Phi}_e)^1_{\mathcal{T}}$ and $(\mathbf{\Phi}_e)^2_{\mathcal{T}}$ are i.i.d. Gaussian, these matrices will satisfy (3.7) with probabilities equal or larger than $1 - 2\,(12/\delta_S)^S\, e^{-C_0 \lceil K_e/2 \rceil}$ and $1 - 2\,(12/\delta_S)^S\, e^{-C_0 \lfloor K_e/2 \rfloor}$, respectively. Therefore, both matrices $(\mathbf{\Phi}_e)^1_{\mathcal{T}}$ and $(\mathbf{\Phi}_e)^2_{\mathcal{T}}$ satisfy (3.7) simultaneously with the common probability

$$\Pr\{(\mathbf{\Phi}_e)^i_{\mathcal{T}} \text{ satisfies (3.7)}\} \geq 1 - 2(12/\delta_S)^S e^{-C_0 \lfloor K_e/2 \rfloor}, \quad i = 1, 2. \tag{B.1}$$

Let $K'_1 \triangleq \lceil K_e/2 \rceil$ and $K'_2 \triangleq \lfloor K_e/2 \rfloor$. Consider the event when both $(\mathbf{\Phi}_e)^1_{\mathcal{T}}$ and $(\mathbf{\Phi}_e)^2_{\mathcal{T}}$ satisfy (3.7). Then the following inequality hold for any vector $\boldsymbol{c} \in \mathbb{R}^S$:

$$\sum_{i=1}^{2} \frac{K'_i}{N}(1 - \delta_S)\|\boldsymbol{c}\|^2_{l_2} \leq \sum_{i=1}^{2} \|(\mathbf{\Phi}_e)^i_{\mathcal{T}}\boldsymbol{c}\|^2_{l_2} \leq \sum_{i=1}^{2} \frac{K'_i}{N}(1 + \delta_S)\|\boldsymbol{c}\|^2_{l_2} \tag{B.2}$$

or, equivalently,

$$\frac{K_e}{N}(1-\delta_S)\|\boldsymbol{c}\|_{l_2}^2 \leq \|(\boldsymbol{\Phi}_e)_{\mathcal{T}}\boldsymbol{c}\|_{l_2}^2 \leq \frac{K_e}{N}(1+\delta_S)\|\boldsymbol{c}\|_{l_2}^2. \tag{B.3}$$

Therefore, if both matrices $(\boldsymbol{\Phi}_e)_{\mathcal{T}}^1$ and $(\boldsymbol{\Phi}_e)_{\mathcal{T}}^2$ satisfy (3.7), then the matrix $(\boldsymbol{\Phi}_e)_{\mathcal{T}}$ also satisfies (3.7). Moreover, the probability that $(\boldsymbol{\Phi}_e)_{\mathcal{T}}$ does not satisfy (3.7) can be found as

$$\begin{aligned}
\Pr\{(\boldsymbol{\Phi}_e)_{\mathcal{T}} \text{ does not satisfy (3.7)}\} &\leq \Pr\{(\boldsymbol{\Phi}_e)_{\mathcal{T}}^1 \text{ or } (\boldsymbol{\Phi}_e)_{\mathcal{T}}^2 \text{ does not satisfy (3.7)}\} \\
&\overset{(a)}{\leq} \sum_{i=1}^{2} \Pr\{(\boldsymbol{\Phi}_e)_{\mathcal{T}}^i \text{ does not satisfy (3.7)}\} \\
&\overset{(b)}{\leq} 4\,(12/\delta_S)^S\, e^{-C_0\lfloor K_e/2 \rfloor}
\end{aligned} \tag{B.4}$$

where the inequality (a) follows from the union bounding and the inequality (b) follows from (B.1). Thus, the inequality (3.30) holds.

Appendix C

Proof of Theorem 3.2

According to (3.30), the matrix $(\mathbf{\Phi}_e)_{\mathcal{T}}$ does not satisfy (3.7) with probability less than or equal to $4\,(12/\delta_S)^S\, e^{-C_0\lfloor K_e/2\rfloor}$ for any subset $\mathcal{T} \subset \{1,\ldots,N\}$ of cardinality S. Since there are $\binom{N}{S} \leq (Ne/S)^S$ different subsets $\mathcal{T}$ of cardinality S, $\mathbf{\Phi}_e$ does not satisfy the RIP with probability

$$\begin{aligned}
\Pr\{\mathbf{\Phi}_e \text{ does not satisfy RIP}\} &\leq 4\binom{N}{S}\,(12/\delta_S)^S\, e^{-C_0\lfloor K_e/2\rfloor} \\
&\leq 4\,(Ne/S)^S\,(12/\delta_S)^S\, e^{-C_0\lfloor K_e/2\rfloor} = 4e^{-(C_0\lfloor K_e/2\rfloor - S[\log(Ne/S)+\log(12/\delta_S)])} \\
&\leq 4e^{-\{C_0\lfloor K_e/2\rfloor - C_3[\log(Ne/S)+\log(12/\delta_S)]\lfloor K_e/2\rfloor/\log(N/S)\}} \\
&= 4e^{-\{C_0 - C_3[1+(1+\log(12/\delta_S))/\log(N/S)]\}\lfloor K_e/2\rfloor}. \qquad \text{(C.1)}
\end{aligned}$$

Setting $C_4 = C_0 - C_3\,[1 + (1 + \log(12/\delta_S))\,/\log(N/S)]$ and choosing C_3 small enough that guarantees that C_4 is positive, we obtain (3.31).

Appendix D

Proof of Lemma 3.2

The method of the proof is the same as the one used to prove Lemma 3.1 and is based on splitting the rows of $\mathbf{\Phi}_e$ into a number of sets with independent entries. Here, the splitting is carried out as shown in (3.33).

Let $(\mathbf{\Phi}_e)^i_{\mathcal{T}},\ i = 1, \ldots, n_p - 1$ be the matrix containing the $(i-1)K + 1$th to the iKth rows of $(\mathbf{\Phi}_e)_{\mathcal{T}}$. The last $K_e - (n_p - 1)K$ rows of $(\mathbf{\Phi}_e)_{\mathcal{T}}$ form the matrix $(\mathbf{\Phi}_e)^{n_p}_{\mathcal{T}}$. Since the matrices $(\mathbf{\Phi}_e)^i_{\mathcal{T}},\ i = 1, \ldots, n_p - 1$ consist of independent entries, they satisfy (3.7) each with probability of at least $1 - 2\,(12/\delta_S)^S\, e^{-C_0 K}$. For the same reason, the matrix $(\mathbf{\Phi}_e)^{n_p}_{\mathcal{T}}$ satisfies (3.7) with probability greater than or equal to $1 - 2\,(12/\delta_S)^S\, e^{-C_0 K_{n_p}}$. In the event that all the matrices $(\mathbf{\Phi}_e)^i_{\mathcal{T}},\ i = 1, .., n_p$ satisfy (3.7) simultaneously for $\boldsymbol{c} \in \mathbb{R}^S$ we have

$$\sum_{i=1}^{n_p} \frac{K_i}{N}(1-\delta_S)\|\boldsymbol{c}\|_{l_2}^2 \leq \sum_{i=1}^{n_p} \|(\mathbf{\Phi}_e)^i_{\mathcal{T}}\boldsymbol{c}\|_{l_2}^2 \leq \sum_{i=1}^{n_p} \frac{K_i}{N}(1+\delta_S)\|\boldsymbol{c}\|_{l_2}^2$$
$$\Rightarrow \frac{K_e}{N}(1-\delta_S)\|\boldsymbol{c}\|_{l_2}^2 \leq \|(\mathbf{\Phi}_e)_{\mathcal{T}}\boldsymbol{c}\|_{l_2}^2 \leq \frac{K_e}{N}(1+\delta_S)\|\boldsymbol{c}\|_{l_2}^2. \qquad \text{(D.1)}$$

Therefore, using the union bound and (D.1), we can conclude that

$$\begin{aligned}\Pr\{(\mathbf{\Phi}_e)_{\mathcal{T}} \text{ does not satisfy (3.7)}\} &\leq \sum_{i=1}^{n_p} \Pr\{(\mathbf{\Phi}_e)^i_{\mathcal{T}} \text{ does not satisfy (3.7)}\} \\ &\leq\ 2(n_p - 1)\,(12/\delta_S)^S \left(e^{-C_0 K}\right) + 2\,(12/\delta_S)^S \left(e^{-C_0 K_{n_p}}\right) \qquad \text{(D.2)}\end{aligned}$$

which proves the lemma.

Appendix E

Proof of Theorem 3.3

According to Lemma 3.2, for any subset $\mathcal{T} \subset \{1, \ldots, N\}$ of size S, the probability that $(\mathbf{\Phi}_e)_{\mathcal{T}}$ does not satisfy (3.7) is less than or equal to $2(n_p-1)\left(12/\delta_S\right)^S \left(e^{-C_0 K}\right) + 2\left(12/\delta_S\right)^S \left(e^{-C_0 K_{n_p}}\right)$. Using the fact that there are $\binom{N}{S} \leq (Ne/S)^S$ different subsets $\mathcal{T}$, the probability that the extended measurement matrix $\mathbf{\Phi}_e$ does not satisfy the RIP can be computed as

$$
\begin{aligned}
&\Pr\{\mathbf{\Phi}_e \text{ does not satisfy the RIP}\} \\
&\leq 2(n_p-1)\binom{N}{S}\left(12/\delta_S\right)^S e^{-C_0 K} + 2\binom{N}{S}\left(12/\delta_S\right)^S e^{-C_0 K_{n_p}} \\
&\leq 2(n_p-1)\left(Ne/S\right)^S \left(12/\delta_S\right)^S e^{-C_0 K} + 2\left(Ne/S\right)^S \left(12/\delta_S\right)^S e^{-C_0 K_{n_p}} \\
&= 2(n_p-1)e^{-(C_0 K - S[\log(Ne/S)+\log(12/\delta_S)])} + 2e^{-\left(C_0 K_{n_p} - S[\log(Ne/S)+\log(12/\delta_S)]\right)} \\
&\leq 2(n_p-1)e^{-\left\{C_0 K - \frac{C_3 K_{n_p}}{K}[\log(Ne/S)+\log(12/\delta_S)]K/\log(N/S)\right\}} \\
&\quad + 2e^{-\left\{C_0 K_{n_p} - C_3 K_{n_p}[\log(Ne/S)+\log(12/\delta_S)]K_{n_p}/\log(N/S)\right\}} \\
&= 2(n_p-1)e^{-\left\{C_0 - \frac{C_3 K_{n_p}}{K}[1+(1+\log(12/\delta_S))/\log(N/S)]\right\}K} \\
&\quad + 2e^{-\{C_0 - C_3[1+(1+\log(12/\delta_S))/\log(N/S)]\}K_{n_p}}. \qquad \text{(E.1)}
\end{aligned}
$$

Denoting the constant terms as $C_4 = C_0 - C_3\left[1 + (1 + \log\left(12/\delta_S\right))/\log\left(N/S\right)\right]$ and $C_4' = C_0 - \left(C_3 K_{n_p}/K\right)\left[1 + (1 + \log\left(12/\delta_S\right))/\log\left(N/S\right)\right]$ and choosing C_3 small enough in order to guarantee that C_4 and C_4' are positive, we obtain (3.35).