NICOLA TESTONI

ADAPTIVE MULTISCALE BIOLOGICAL SIGNAL PROCESSING

ADAPTIVE MULTISCALE BIOLOGICAL SIGNAL PROCESSING

NICOLA TESTONI

Dissertation for the degree of Doctor of Philosophy

Advanced Research Center on Electronic Systems
University of Bologna

March 2008

Nicola Testoni: *Adaptive Multiscale Biological Signal Processing*, Dissertation for the degree of Doctor of Philosophy in Information Technology, © March 2008

SUPERVISORS:
Guido Masetti
Sergio Graffi
Nicolò Speciale

To my parents, to Silvia and to my friends
for having accompanied me in this wonderful journey.

Ai miei genitori, a Silvia ed ai miei amici
per avermi accompagnato in questo magnifico viaggio.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADID Adjustable Delay Iterative Deconvolution

AP Action Potentials

ARMA Auto-Regressive Moving-Average

AR Auto-Regressive

AWC	Adaptive Weight Control
AWGN	Additive White Gaussian Noise
AdWLSD	Adaptive delay Wavelet Least Squares Deconvolution
CWT	Continuous Wavelet Transform
DFT	Discrete Fourier Transform
DT	Deterministic Trends
DWT	Discrete Wavelet Transform
EEG	electroencephalography
EMG	electromyography
EN	Elastic Net
ER	Extracellular Recordings
FDR	Fisher Discriminant Ratio
FIR	Finite Impulse Response
FLD	Fisher Linear Discriminant
FWRD	Fourier Wavelet Regularized Deconvolution
HOS	Higer Order Statistics
IIR	Intinite Impulse Response
ISI	Intersymbol Interference
LFP	Local Field Potential
LMF	Least Mean Fourth
LMMSE	Linear Minimum Mean Square Error
LMS	Least Mean Square
LP	Linear Programming
LTI	Linear Time Invariant
MA	Moving-Average

MCMC	Monte-Carlo Markov Chain
MER	Multielectrode Extracellular Recordings
MSE	Mean Square Error
NEM	Non-linear Even Moments
PCA	Principal Component Analysis
PSD	Power Spectral Density
PSF	Point Spread Function
QCWT	Quasi-Continuous Wavelet Transform
RF	Radiofrequency
RLS	Regularized Least Square
SAF	Sub-band Adaptive Filter
SAR	Synthetic Aperture Radar
SIS	Shift Invariant Spaces
SNR	Signal to Noise Ratio
SOS	Second Order Statistics
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SWT	Stationary Wavelet Transform
ROC	Receiver Operating Characteristic
ROI	Region of Interest
TGC	Time-Gain Compensation
TRUS	Trans-Rectal Ultrasound
US	Ultrasound
WARES	Wavelet Adaptive Rescaling
WA	Wavelet Analysis

WLSD Wavelet Least Squares Deconvolution

WPT Wavelet Packet Transform

WT Wavelet Transform

ABSTRACT

Biological processes are very complex mechanisms, most of them being accompanied by or manifested as signals that reflect their essential characteristics and qualities. The development of diagnostic techniques based on signal and image acquisition from the human body is commonly retained as one of the propelling factors in the advancements in medicine and biosciences recorded in the recent past.

It is a fact that the instruments used for biological signal and image recording, like any other acquisition system, are affected by non-idealities which, by different degrees, negatively impact on the accuracy of the recording. This work discusses how it is possible to attenuate, and ideally to remove, these effects, with a particular attention toward ultrasound imaging and extracellular recordings.

Original algorithms developed during the Ph.D. research activity will be examined and compared to ones in literature tackling the same problems; results will be drawn on the base of comparative tests on both synthetic and *in-vivo* acquisitions, evaluating standard metrics in the respective field of application. All the developed algorithms share an adaptive approach to signal analysis, meaning that their behavior is not dependent only on designer choices, but driven by input signal characteristics too.

Performance comparisons following the state of the art concerning image quality assessment, contrast gain estimation and resolution gain quantification as well as visual inspection highlighted very good results featured by the proposed ultrasound image deconvolution and restoring algorithms: axial resolution up to 5 times better than algorithms in literature are possible. Concerning extracellular recordings, the results of the proposed denoising technique compared to other signal processing algorithms pointed out an improvement of the state of the art of almost 4 dB.

SOMMARIO

I processi biologici sono meccanismi complessi, la maggior parte dei quali accompagnati da segnali che ne riflettono le caratteristiche e le qualità essenziali. Lo sviluppo di tecniche diagnostiche basate sull'acquisizione di segnali ed immagini provenienti dal corpo umano è comunemente ritenuto come uno dei fattori chiave dello sviluppo delle scienze mediche e biologiche registrato nel recente passato.

È un dato di fatto che gli strumenti utilizzati per la registrazione di segnali biomedici, come ogni altro sistema di acquisizione, siano affetti da non-idealità le quali, in diversa maniera, influenzano negativamente l'accuratezza dei dati registrati. Questo lavoro discute come sia possibile attenuare, ed idealmente rimuovere, questi effetti, con una particolare attenzione per l'elaborazione di immagini ecografiche e segnali extracellulari.

Durante la trattazione verrano esaminiati gli algoritmi sviluppati durante l'attività di dottorato, comparandoli con altri presenti in letteratura e progettati per la soluzione degli stessi problemi; ogni conclusione verrà tratta in seguito a test comparativi su acquisizioni di segnali sintetici ed *in-vivo* ed in base a metriche standard nel rispettivo campo di applicazione. Tutti gli algoritmi sviluppati condividono un approccio adattativo all'analisi di segnale, ovvero l'elaborazione non dipende soltanto dalle scelte del progettista ma è guidata anche dalle caratteristiche del segnale di ingresso.

Il confronto delle prestazioni misurate sia in base allo stato dell'arte in fatto di stima della qualità delle immagini, guadagno di contrasto e di risoluzione, sia tramite ispezione visuale, ha evidenziato ottimi risultati per quanto riguarda gli algoritmi proposti per la deconvoluzione ed elaborazione delle immagini ecografiche, riportando risoluzioni assiali fino a 5 volte superiori agli algoritmi presenti in letteratura. Nel campo dei segnali extracellulari, il confronto dei risultati forniti dalla tecnica di rimozione del rumore qui proposta con altri algoritmi di elaborazione del segnali ha mostrato che è possibile migliorare lo stato dell'arte di circa 4 dB.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor and my tutor, Prof. Guido Masetti and Dr. Nicolò Speciale, whose expertise, understanding, and patience, added considerably to my graduate experience. I really appreciate their vast knowledge and skill in many areas (e.g. signal processing, multiresolution analysis, adaptive filtering) and their assistance in writing reports (i.e., grant proposals, scholarship applications and this thesis).

I would like to thank all my colleagues in my research group, Emanuele Baravelli, Luca De Marchi, Marco Messina, Simona Maggio, Alessandro Palladini, Matteo Montani, Martino Alessandrini, Salvatore Caporale and Francesco Franzè for the assistance they provided at all the levels of my research work.

I must also acknowledge Prof. Martin Vetterli, Dr. Andrea Ridolfi, Mrs. Jocelyne Plantefol, Mrs. Erika Raetz and all the peolple at the EPFL-LCAV laboratory for providing me the most conforatble, enjoyable and fruitful working environment ever.

Finally, I wish to acknowledge Dr. Christophe Pouzat and all the staff of the *Laboratoire de Physiologie Cérébrale* of the *UFR biomédicale de l'Université René Descartes* (Paris V) and Prof. Leonardo Masotti, Prof. Elena Biagi and all the staff of the *Laboratorio Ultrasuoni e Controlli non Distruttivi* of the *Facoltà di Ingegneria dell'Università degli Studi di Firenze* for the data they provided me and for many interesting discussion about my research topics.

Last but not least, I would also like to thank my family for the support they provided me throughout my entire life and in particular, I must acknowledge my sweetheart, Silvia, who started writing with me a much longer and much more attractive book.

INTRODUCTION

Biological processes are very complex mechanisms, encompassing both neural and hormonal stimuli and responses, inputs and outputs in the most different forms, including physical material or information, and actions that could as well be mechanical, electrical, or biochemical. Most of these processes are accompanied by or manifest themselves as signals that reflect their essential characteristics and qualities. Such signals may be very different in nature, for example biochemical, in the form of hormones and neurotransmitters; electrical, such as potentials and currents; physical, like pressure and temperature.

The development of diagnostic techniques based on signal and image acquisition from the human body is commonly retained as one of the propelling factors of the advancements in medicine and biosciences recorded in the recent past.

In fact, diseases or defects in biological systems almost always cause alterations in normal functions, giving birth to pathological processes that negatively impact on the performance and behavior of the systems themselves. A pathological process is typically associated with signals pat-

terns and anatomical features that are somewhat different from the ones of the corresponding healthy state. If a good understanding of the system of interest is retained, it is possible, after the investigation of the signals and features originated by the system, to assess its state, discriminating between normal and abnormal responses.

However, most physicians, like radiologist and neuroscientists, have to deal with additional problems when faced to the problem of diagnosing the health state of a biological system from its recorded signals. Like any acquisition system, the instruments used for biological signal and image recording are affected by non-idealities which, by different degrees, negatively impact on the accuracy of the recording. Unwanted spectral deformations, artifacts, and additional signal-dependent noise components are just a few examples of undesired effects introduced by the acquisition equipment.

Furthermore, both the undeformed signal and a significant part of these effects are characterized by behaviors which change over time, the respective degree of non-stationarity being the most varying, ranging from inter-acquisition to intra-acquisition. Being capable of tailoring their own processing to the time-varying characteristics of the input signals, adaptive algorithms are the most suited to this environment. A multiscale approach should also be employed in order to best exploit the information concerning the different time scales: separately tracking short- and long-term variations in a system provides better control over its behavior.

This work discusses how it is possible to attenuate, and ideally to remove, these effects, with a particular attention to ultrasound imaging and extracellular recordings. Algorithms in literature tackling this problem will be examined and compared to original ones developed during the Ph.D. research activity; results will be drawn on the base of comparative tests on both synthetic and *in-vivo* acquisitions, evaluating standard metrics for the respective field of application.

Some background material will be presented in Part i, discussing about the two chosen target environments of ultrasound imaging (Chapter 1) and extracellular recording

(Chapter 2) and about Wavelet Transform (Chapter 3).

In particular, Chapter 1 opens up with an overview about the history and the problems connected to imaging techniques based on ultrasound generation and acquisition. This brief discussion will be followed by a section illustrating how signal is acquired and the most typical parameters and quantities involved in this process. Finally, a comprehensive model in literature will be illustrated, discussing its structure and explaining how it is possible to move from this to other models describing different features of the investigated tissue.

A very similar structure is featured by Chapter 2, where the developments of extracellular recordings over the last one and a half century and the difficulties arising from their recording and processing will be briefly discussed. Modern neural signal acquisition technique will be described subsequently, outlining a typical experimental setup, its parameters and the acquisition equipment employed. Next, the classical neural signal model will be presented and compared to an original one, developed during the Ph.D research activity and discussed in Section 2.4, encompassing also a signal dependent noise component which allows for a more accurate representation of the received signal.

Chapter 3 introduces the Wavelet Transform as a tool for non-stationary signal representation. Four different flavors of this transform will be presented and discussed, explaining how each one deals with time-scale sampling and transform-domain representation redundancy; efficient implementations of each transform, based on perfect reconstruction filter banks, will also be provided. Finally, an algorithm for scaling function iterative generation will be illustrated, allowing for the computation of Wavelet filters coefficients.

The main contributes will be presented in Part ii, discussing tools for automatic impulse response estimation (Chapter 4), signal deconvolution (Chapter 5), adaptive filtering (Chapter 6) and denoising (Chapter 7).

More in details, Chapter 4 will introduce two algorithms already in literature, one of them based on the properties of the Wavelet Transform, both meant for the estimation of the acquisition system impulse response and based on the assumption of minimum phase. An original estimation

algorithm, which combines the benefits of the two former approaches, will then be presented in Section 4.3. After a brief discussion about the experimental setup usually adopted in this context, chapter will conclude with a comparative analysis of the performance of the three algorithms in terms of estimation accuracy and output response stability.

Chapter 5 will introduce the problem of signal deconvolution, discussing how it is possible to jointly estimate the unblurred signal and refine the first guess impulse response. Two algorithms already in literature will be illustrated and compared with two original techniques discussed in Section 5.2.1 and Section 5.3. While the first one is actually an evolution of a known algorithm, the second constitutes a remarkable perspective change, as it tackles the problem of deconvolving an echo signal in which reflectors are not time-aligned with the sampling grid. Performance comparisons at the end of the chapter will highlight how much positively this new algorithm compares with the state of the art.

Next, in Chapter 6 the wide class of Least Mean Square adaptive filtering algorithm will be presented. Four different filtering structures will be examined, ranging from the plain finite impulse response filter to the sub-band filtering scheme; for each of them, an adaptive gain version will also be depicted, with more details and original contributes given in Appendix A. Again, performance comparisons based on widely accepted image quality assessment metrics will be used to study weaknesses and strong points of each algorithm when used in the context of image deblurring.

Finally, Chapter 7 will make use of the model depicted in Section 2.4 to develop an original denoising algorithm capable of removing a wide spectrum of signal dependent noises from an incoming input signal. Algorithm derivation in the time domain presented in Section 7.1 will be followed by the discussion of the hypotheses made to find a tractable form and by the development of a similar algorithm in the Wavelet domain, as described in Section 7.2. A discussion concerning real-time estimation of algorithm parameters concluded by performance comparisons with other Wavelet-based denoising algorithm will end this chapter.

As a conclusion of this work, two sample applications

will be discussed in Part iii, one concerning the automatic classification of biological tissues by means of received ultrasonic echoes (Chapter 8), the other dealing with the problem of neural spike sorting (Chapter 9).

First, in Chapter 8, two sections will propose an original two-step biological tissue classifier. In the first section, discussion will be centered around signal preprocessing: the adaptive signal processing algorithms presented so far will be combined in order to remove systematic trends, attenuate the spectral deformation due to the ultrasound transducer and separate the true echo signal from the speckle component. The second section will discuss the problem of tissue classification, involving features extraction and statistical parameters evaluation on a given set of zones and regions of interests.

On the other hand, in Chapter 9, a classical spike sorting algorithm will be examined and then compared to an original approach based on pulse estimation and signal deconvolution. Insights about how to improve this method will be given, introducing an approach based on Shift Invariant Space and the exploitation of the intrinsic sparsity of the neural signal.

Two appendixes covering the topics of Least Mean Square filtering and Wavelet adaptive denoising will close this work.

Part I

BACKGROUND

1 ULTRASOUND IMAGING

Ultrasound (US) imaging, is a relatively inexpensive, fast and radiation-free imaging modality. It is excellent for a non-invasive imaging and diagnose of a number of organs and conditions, without x-ray radiation; it is however often difficult to interpret: as a matter of fact, results of diagnostics using conventional US images are highly dependent on the physician's skills.

Modern obstetric medicine relies heavily on ultrasounds to provide detailed images of the fetus and uterus both in 2D and, more recently, 3D. Ultrasound is also extensively used for evaluating the kidneys, liver, pancreas, heart, and blood vessels of the neck and abdomen. Last but not least, ultrasound can also be used to guide fine needle, tissue biopsy, and to facilitate sampling cells from an organ for lab testing, for example, to test for cancerous tissue.

US imaging is finding a greater role in the detection, diagnosis and treatment of heart disease, heart attack, acute stroke and vascular disease which can lead to stroke. Moreover, due to the recent developments in US signal processing, ultrasound is also being used more and more to image

the breasts and the prostate in order to allow for an early diagnosis of cancer in both these glands.

1.1 ULTRASOUND IMAGING: AN OVERVIEW

The development of ultrasound applications started in 1826, with a European experiment measuring distance under water using sound waves: Jean-Daniel Colladen, a Swiss physicist/engineer and Charles-Francois Sturm, a mathematician, used an underwater bell in an attempt to calculate the speed of sound in the waters of Lake Geneva. Despite their crude instruments, they managed to determine that the speed of sound under water was 1435 m/s, a result not too different from what it is known today.

Many advancements in this field were carried out, mainly in England during the XVIIth century, but the real breakthrough in the evolution of high frequency echo-sounding techniques came when the piezoelectric effect in certain crystals was discovered in 1880 by Pierre Curie and his brother Jacques Curie in Paris [1]. Further research and development in piezoelectricity soon followed, while the turn of the century saw the invention of the diode and the triode, allowing powerful electronic amplifications necessary for the developments of ultrasonic instruments.

The first tentative ultrasonic instruments for medical diagnosis date back to 1942 [2]: the images they produced were very rudimentary mosaics of light intensity points photographically recorded on heat-sensitive paper.

Systematic investigations of ultrasound techniques as a diagnostic tool finally took off in the United States in the late 1940s with the work of George Ludwig, who also determined the actual mean value of the velocity of sound in animal soft tissues [3]. 1952 saw the invention of the first linear hand-held B-mode instrument by John Reid and John Wild [4]: the instrument operated at a frequency of 15 MHz and allowed to visualize tumors by sweeping from side to side through breast lumps.

The increase in the research and application of ultrasound to medicine appeared to boom from 1966 onwards when there was an upsurge of centers and people in Europe, the United States and Japan begun to embark on studies about

the application of ultrasound to medical diagnosis.

Nowadays, modern ultrasound imaging systems are used to obtain images from almost any kind of soft tissue structures present in the human body: although anatomy can still be studied through the same 2D images made up of bright dots of the 1942 Dussik's experiment, although with a much higher resolution, 3D ultrasound technology is now becoming a viable commercial proposition due to the improvements in computer technology.

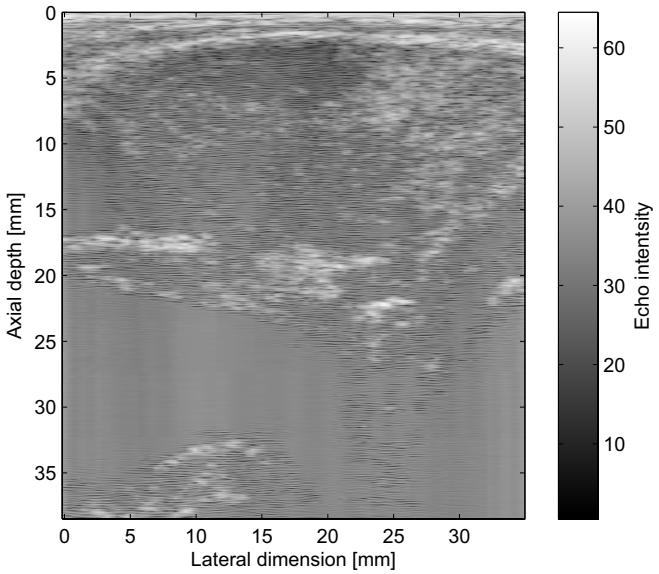
Figure 1 shows an example of echographic image along with some basic elaboration: this *in-vivo* scan of a human prostatic gland portraits one vertical section of the gland, containing a malignant tumor called *carcinoma*. This lesion is characterized by a lower echogenicity with respect to the healthy prostatic tissue: this corresponds to a lower intensity of the reflected echo signal. In fact, by carefully inspecting figure 1a near to the top border, it is possible to identify a slightly darker region. Skilled physicians use this kind of images to diagnose diseases embedded in many different tissues.

The elaborated image in figure 1b highlights a low-pass spatial effect introduced by the transducing equipment which may be due to the non homogeneous thickness of the gel coating used to bridge the gap between the transducer and the skin or to non-idealities in the transducer itself. This is only one example of the undesired effects the transducer has on the recorded echo signal.

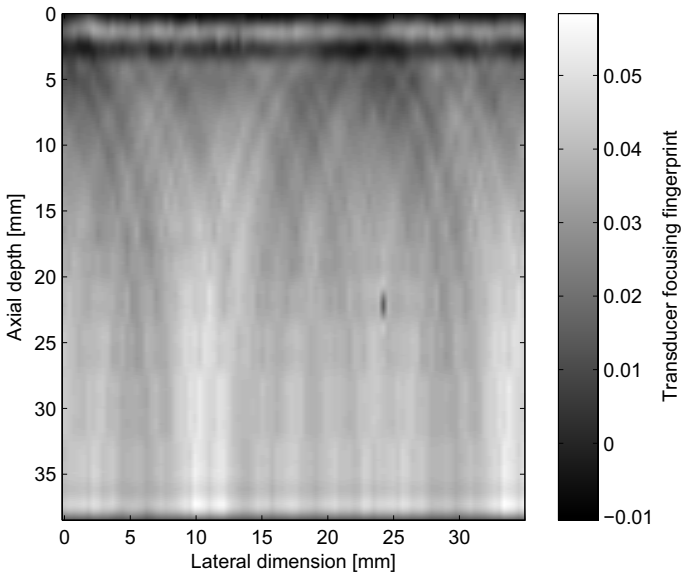
Moreover, due to its finite bandwidth, the transducer produce a deformation in the spectrum of both the transmitted pulse and the received echo. While the first deformation can be regarded as part of the transmitted pulse, the second deformation may alter significantly the received echo, reducing the resolution of the echographic image. It is understandable how this phenomenon is strictly dependent on the transducer: different probes would cause different distortions while recovering the same incoming echo signal.

Because of this, the first and most important task in tissue classification is the removal of the signal deformation introduced by the echographic probe.

This is not a trivial task: first of all, the probe impulse response changes, in practice, from column to column as the piezoelectric transducers configuration is modified to



(a) B-mode ultrasound image



(b) Central frequency estimation

Figure 1: A B-mode ultrasound scan of an *in-vivo* human prostatic gland affected by a carcinoma (a) and the corresponding transducer fingerprint extracted from the image (b); both image quality and tissue classification performance can be improved by removing transducer effects.

achieve focus. Next, even if the impulse response could be perfectly recovered, the presence of noise would give way to an ill-posed inversion problem [5, 6, 7, 8]. Finally, as the same transducer is used for both transmitting and receiving the ultrasound signal, it is not possible to recover the real pulse the tissue sees as a stimulus.

In this context, adaptive signal processing techniques are mandatory in order to track and remove the time-varying deformations the incoming ultrasound signal undergoes during the acquisition phase. Fast convergence to the transducer impulse response and real time data processing capabilities are both requirements that a system addressing this problem must satisfy in order to attenuate the impact the probe has on the acquired data.

1.2 SIGNAL ACQUISITION

The main units of a modern echographic equipment [9] are depicted in Figure 2. The echographic transducer is designed for both transmitting and receiving ultrasound signals: as the transmitted pulse features much more power than the received echo, in order to avoid saturation in the input stage an efficient insulation device must be employed to decouple it from the US transmitter during pulse emission.

The central frequency of the transmitted pulse is in between 2 and 15 MHz basing on the different tissues that are going to be investigated; some special equipments for dental or skin ultrasonography have a central frequency exceeding 50 MHz. Relative bandwidth at -6 dB is typically in the order of $60 \sim 70\%$ but there are cases in which relative bandwidth over 100% may be required. Because of this, the piezoelectric transducer is almost always built from advanced composite materials.

Sound speed c in the investigated tissues varies from the 1450 m/s of the body fat to the 1585 m/s of the muscles, with the two significant exceptions of lungs and bones, as it is shown in Table 1. Taking a mean value of 1540 m/s and considering resolvable an object with a diameter of 5 wavelength, the typical axial resolution of a 7.5 MHz transducer is approximatively 1 mm.

To improve axial resolution, higher frequencies must be

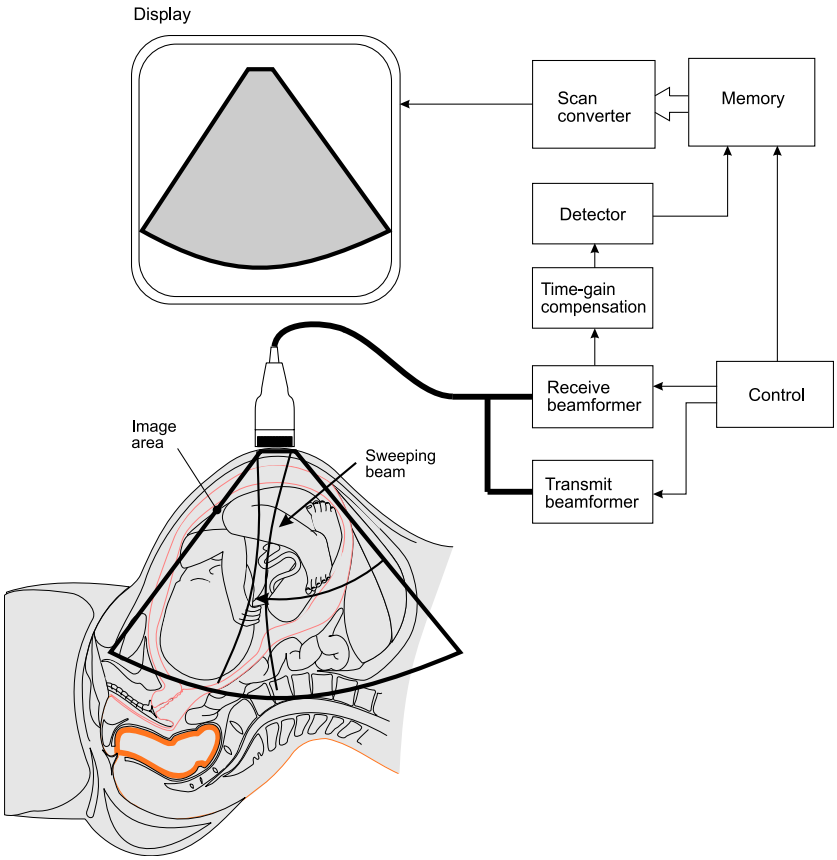


Figure 2: Schematic design of an ultrasound imaging equipment highlighting signal flow through the different units

	$c\left[\frac{m}{s}\right]$	$\alpha\left[\frac{dB}{MHz \cdot cm}\right]$
Air	330	$\geqslant 10$
Blood	1575	0.18
Bone	4080	8
Fat	1460	0.6
Lungs	900	30
Muscles	1580	1.5
Water	1480	0.002

Table 1: Sound speed and attenuation in various kinds of tissues [3].

employed: however, sound attenuation α must be taken into consideration too. This is another tissue dependent parameter: differently from the sound speed, there are various order of magnitude between the least and the most attenuating tissues, at it is shown in Table 1. Using a mean value of 0.7 dB/MHz · cm, the total attenuation from an echo source at 6 cm depth at 7.5 MHz is 31.5 dB and increases to 52.5 dB at a depth of 10 cm.

In order to compensate this depth dependent attenuation a post processing stage, known as Time-Gain Compensation (TGC), must be performed. This elaboration is based on a manual estimate of the attenuation, in the form of a TGC curve: a set of gain sliders corresponding to different tissue depth are moved until a calibration image looks homogeneous.

The physical system is usually made up of specialized integrated low noise logarithmic amplifiers, as they have to process signals with more than 60 dB of dynamic. Most state of the art echographic units allow to extract the Radiofrequency (RF) signal from the system at this point: only a few of them provide an RF output before the TGC.

After time-gain compensation took place, in order to recover the local intensity of the received echo, the signal is envelope detected by means of an Hilbert Transform. Signal is then fed into the display memory, in order to be shown or stored. If the ultrasonic probe is built on a convex piezoelectric array, like in Figure 2, the image is shaped to an annulus sector by means of the *scan converter*, otherwise is kept rectangular.

1.3 SIGNAL MODEL

Echographic signals result from the interaction between the pressure wave generated by the transducer and the tissue structure. A comprehensive model for the received signal $y(t)$, shown in Figure 3, is discussed in [10]:

$$y(t) = u(t) * T_f(t) * P_f(t) * x_0(t) * P_b(t) * T_b(t) \qquad (1.1)$$

where $u(t)$ is the electrical impulse driving the transducer, $T_f(t)$ and $T_b(t)$ are the transmission and reception transducer impulse response respectively, $P_f(t)$ and $P_b(t)$ are the

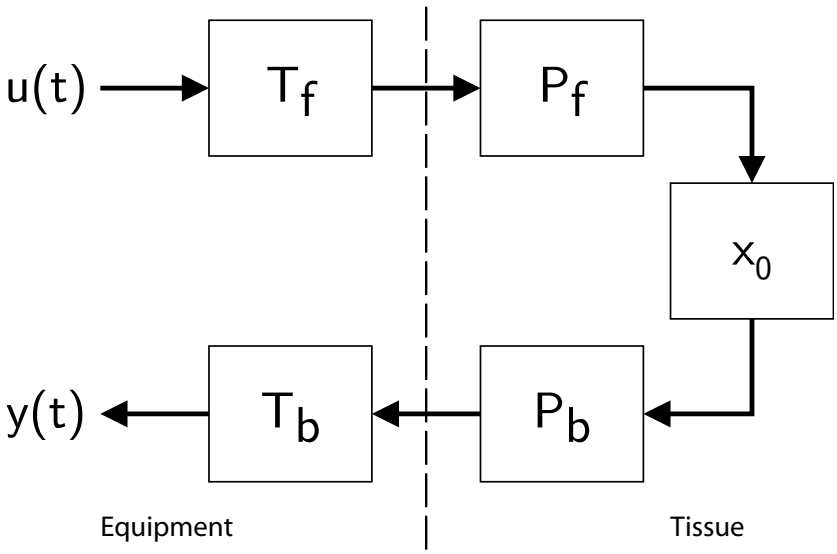


Figure 3: Time-domain convolutive model for the received echographic signal.

forward and backward propagation path impulse response respectively, and $x_0(t)$ is the target tissue impulse response.

Let $x(t) = P_f(t) * h_0(t) * P_b(t)$ be the tissue response and $h(t) = U(t) * T_f(t) * T_b(t)$ be the imaging system effective pulse; under the assumptions of weak scattering, narrow ultrasound beam and linear propagation, the echo signal $y(t)$ can be expressed [11] as

$$y(t) = h(t) * x(t) = c(t) + d(t) \tag{1.2}$$

with

$$\begin{aligned} c(t) &= \textstyle\sum_{k=1}^{N_c} h(t) * c_k(t - \theta_k) \\ d(t) &= \textstyle\sum_{k=1}^{N_d} h(t) * d_k(t - \tau_k) \end{aligned} \tag{1.3}$$

where $c(t)$ and $d(t)$ derive from the interaction of the pulse with the resolvable reflectors and the randomly located diffuse scatterers respectively, while N_c and N_d are the number of coherent reflectors and diffuse scatterers, θ_k and τ_k their time delays to the receiver, and c_k and d_k their relative strengths.

While coherent and diffuse component superposition is not altered by sampling $y(t)$, a noise component γ is introduced:

$$y(n) = c(n) + d(n) + \gamma(n) \tag{1.4}$$

Wold-decomposition theorem allows to split $y(n)$ into its components [12]. The coherent component $c(n)$ can be approximated by a summation of Gaussian modulated sinusoids: the resolvable reflector structure can be viewed as a summation of delta functions of random strength located at the resolvable scatterers' location, while the echo pulse can be thought a Gaussian modulated sinusoid. The diffuse component $d(n)$ is well modeled by an Auto-Regressive (AR) stochastic process. The noise component $\gamma(n)$ is usually considered white, although a more careful modeling may use a signal dependent model to improve the description of the acquisition system.

Due to the focusing equipment used in the modern ultrasonographers, the reception transducer impulse response can be modeled by a time varying transfer function: as a consequence, the whole imaging system effective pulse $h(t)$ becomes a time dependent function. On the other hand, due to the frequency dependent nature of the attenuation, both the forward and backward propagation path impulse response should be considered intrinsically time dependent.

This means that the Power Spectral Density (PSD) of the backscattered signal received by the ultrasound probe a held over a particular point is a function of both location (m, n) and frequency f [13]:

$$E[|Y(m,n,f)|^2] = |H(m,n,f)|^2|X(m,n,f)|^2 + \sigma_\gamma^2(m,n) \quad (1.5)$$

where the noise $\gamma(t)$ was supposed zero-mean Gaussian with standard deviation σ_γ and $E[\bullet]$ is the expectation operator.

Exploiting the Born approximation, the intensity of the backscattered signal can be represented by a combination of two terms:

$$|X(m,n,f)| = A(m,n,f)B(m,n,f) \quad (1.6)$$

where $A(m,n,f)$ describes the cumulative attenuation of the sample at location (m,n) and $B(m,n,f)$ is the backscatter intensity of the sample. Several models exist for the last [11], while the former, expressed in Neper, is usually assumed to be linearly proportional to the frequency [14].

2 EXTRACELLULAR SIGNAL RECORDINGS

The simultaneous and coordinated activity of neurons and the physical networks they create through their interconnections allow human brain to process and store informations. It is commonly believed that both spatial and temporal information are involved in these processes, generating patterns which are currently investigated by neurophysiologist.

Extracellular electrophysiology is currently the best tool for investigating the high-resolution recordings from awake animal neural tissue necessary to study these patterns. This tool is capable of delivering information on both spiking and synaptic activity by recording the electrical potentials which are generated in the neighborhood of neural cells.

The retrieved information must be carefully examined and interpreted, since it is extremely redundant as spikes are almost always superimposed: interpretation requires time, patience and attention. However, thanks to the steady developments in the automatic interpretation of Extracellular Recordings (ER) data, more and more tools are given to the neurophysiology community to ease this process.

2.1 EXTRACELLULAR RECORDINGS: AN OVERVIEW

ER measure changes in potential in the neighboring of a neuron or an axon. The first breaking in this field date back to 1875, when Richard Caton discovered that currents could be recorded from deep inside the brain [15]. ER have progressed significantly over the past 130 years, allowing scientist and physician to describe neurons and their interconnections in details, and are actually one the most common techniques for monitoring the activity of small neurons populations.

In 1952, two scientists, Hodgkin and Huxley, published a set of articles [16, 17] describing the way neural Action Potentials (AP) originate in neurons, winning the Nobel Prize in Physiology or Medicine 1963. Using a set of differential equations, they described the behavior of the cell membrane when excited by an external electrical stimulus, giving way to a clear interpretation of the mechanisms involved in excitation and inhibition in the peripheral and central portions of the nerve cell membrane.

Based on their observation, many models for both the neural cells behavior and their interconnections have been derived: ER and, more recently, Multielectrode Extracellular Recordings (MER) have been extensively used to validate these models, since they provide information both about spiking and synaptic activity.

However, both ER and MER suffers of a common drawback: each channel collects and records the AP of many neurons and/or axons at the same time, as well as a certain amount of background noise. This fact reflects into the necessity of a post-processing step known as *spike sorting*. As well as ER spike sorting, MER spike sorting is built on the identification and separation of the different spike waveforms –*templates*– throughout the recorded channels, but instead of looking at one channel at a time, each spike is detected using the information brought by all the channels at the same time.

Figure 4 shows an example of MER along with some basic elaboration: the data collected in this figure contains a train of four AP, whose presence can be easily detected by means of both visual inspection and data processing. The top row of the figure contains an estimation of the entropy of the

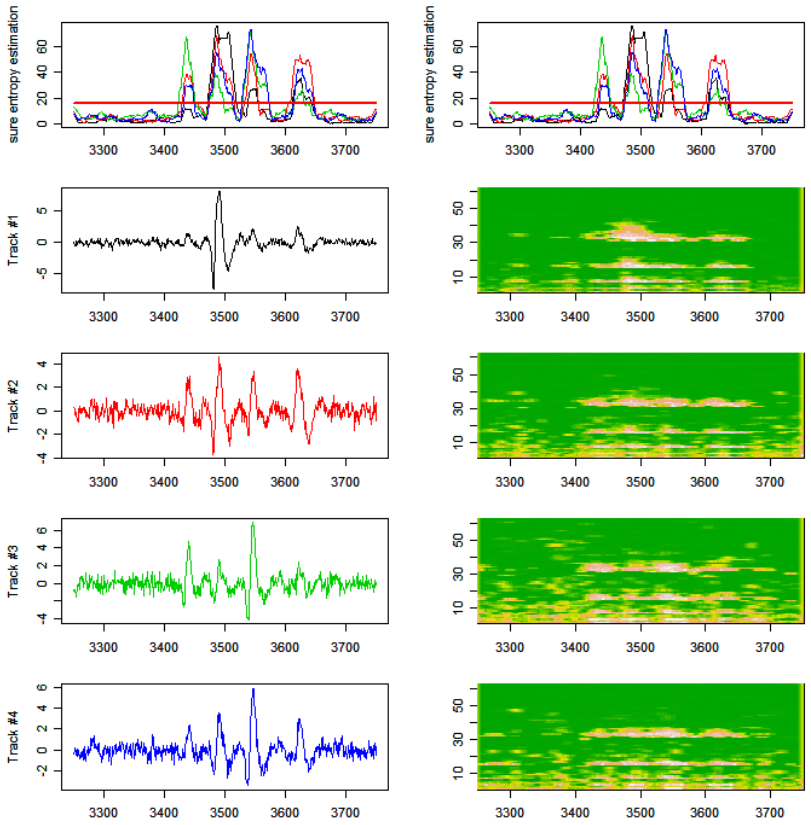


Figure 4: Extracellular waveform recorded[a] from the locust (*Schistocerca americana*) antennal lobe filtered between 300 Hz and 5 kHz; on the top row, entropy estimation is performed to reveal the four neural spikes recorded by the tetrode; on the bottom rows, the raw signal is associated to its multiscale analysis to enhance the differences between the recordings of each channel.

a The author gratefully acknowledge C. Pouzat and the Laboratoire de Physiologie Cérébrale, UFR biomédicale de l'Université René Descartes (Paris V) for the data provided.

signal, each color referring to a different electrode: four distinct peaks are present, corresponding to each AP. Each of the following rows shows on the left size the electrical potential recorded by the corresponding electrode, while on the right side a multiband signal processing technique is employed to show the Wavelet Packet Tree power spectrum.

AP clearly overlaps both in time and frequency, making it harder to separate each other; at the same time, each electrode records a different mixture for each train of action potential; this is due to the attenuation each AP undergoes during the travel from the spiking neuron to the recording electrode.

The first and most important task in spike sorting is the detection of each spike. This is typically done using detectors which are triggered on any times the input signal becomes greater than a fixed threshold. It is easy to understand how noise can interfere with such a detector: if the noise power is sufficient to make the signal greater than the detector threshold even in absence of an AP, a false AP is recorded.

At the same time, noise can alter the feature of an AP in different ways, simply by delaying, anticipating, lengthening or shortening the duration of an AP. These problems becomes more evident in MER spike sorting, where each channel contributes to the localization of the AP. In this context, noise is considered as the principal cause of the statistical spreading of model parameters, thus impinging negatively on the classification capability of algorithms based on clustering techniques to separate the different spikes.

In most ER and MER experimental setups and equipments, signal is filtered before the detection stage takes place [18]. However, this filtering stage is performed for reasons that differ from denoising: first of all, it is used to remove the Local Field Potential (LFP), that is, the part of the signal that depends on the synchronous flow of current into a parallel set of dendrites [18], this fact being commonly observed in electromyography (EMG) [19, 20] and electroencephalography (EEG) [21].

Second, it is necessary in order to guarantee that the subsequent analog to digital conversion does not introduce aliasing into the sampled signal, that is the production of

artifacts as a results of sampling at intervals too large to permit a faithful reconstruction of the original signal. As a byproduct of this step, noise is filtered out outside the filter bandwidth, but more intensive calculations must be performed in order to eliminate or at least reduce the noise within the filter bandwidth.

2.2 SIGNAL ACQUISITION

Acquiring neural signals using the MER technique require a proper experimental setup, as it is shown in Figure 5. In the case depicted, the MER probe is going to be inserted into the antennal lobe of a common insect (bug, cricket, locust, ...) in order to record the AP evoked by different odors delivered to the receptors on the antennas.

Before starting the experiment, the insect is stripped of part of the head exoskeleton; mandibles and alimentary canal are removed too, in order to prevent the insect to swallow. Insect is then firmly held by a constraining device, connected to the micromanipulator. This device is used to position the MER probe tip, shown in Figure 6, near the antennal lobe, in the insect brain. Microphotographies like Figure 7 allow physicians to asses the correct positioning of the probe within the brain.

Antennas are then connected to the odor delivery systems by means of small pipes: a computer controlled pump allows vaporized micro drops of essence to be delivered from a cartridge to the antennas. Insect life is sustained during all the experiment by means of a saline solution.

A Faraday cage is usually employed in order to insulate both the experiment and the recording equipment from electromagnetic interference, since the MER probe acts like an antenna. Vibrations that could alter the position of the probe tip in the insect brain, thus modifying the AP waveforms, are dampened by means of a floating table.

The recording system, not shown in Figure 5, amplifies and samples the electric potentials measured at each electrode in the probe tip. Low noise amplification with gains in the order of tenth of thousands are quite common in this field as the recorded potentials are in the order of the hundred of microvolts, as discussed in [22, 23].

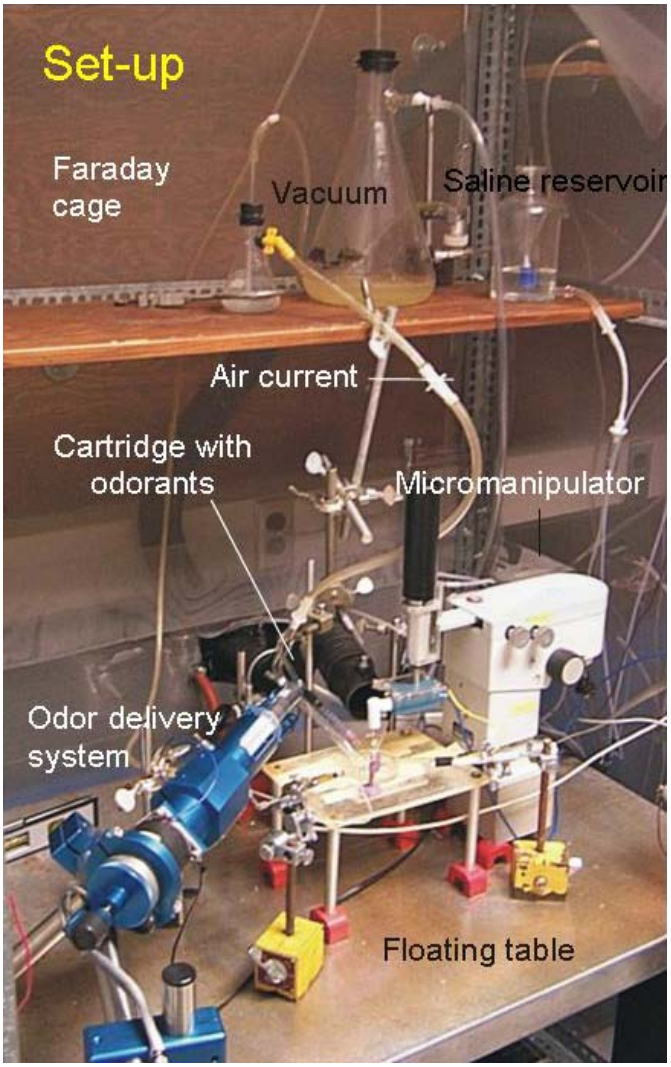


Figure 5: Extracellular signal acquisition equipments; the system is insulated from external vibrations and electromagnetic noises.

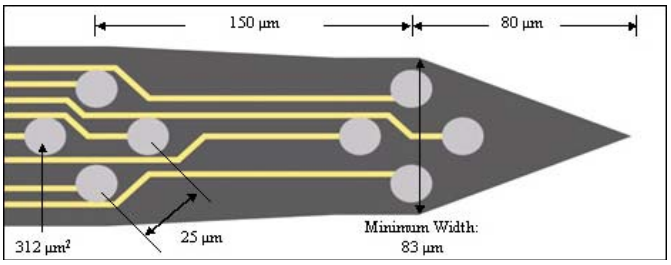


Figure 6: Schematic design of one tip of the MER acquisition probe; each tip carries 8 electrodes grouped in two tetrodes.

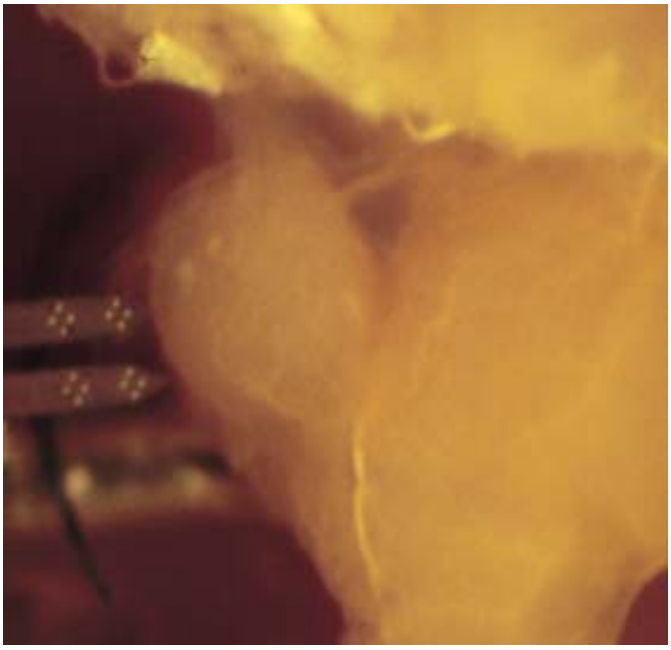


Figure 7: Microphotography of the MER probe inserted near a Locust antennal lobe; both the probe tips are visible.

As discussed in the previous section, signal is then filtered: a four pole elliptic low-pass filter with a cutoff frequency of 300 Hz is usually employed to extract the LFP, while band-pass filtering between 300 Hz and 5 kHz is used to capture the AP. The signal is then sampled with frequencies ranging from 10 kHz to 25 kHz, depending on the hardware setup and the application.

2.3 CLASSICAL SIGNAL MODEL

The classical model for ER signals, shown in Figure 8, consists of three main components [18]: the pure superposition of the AP, the LFP and a purely additive white noise. Under the assumptions of low-pass bandwidth for the LFP and Gaussian nature for the noise, a simple elliptic based filtering stage is often employed [24, 25, 26, 27] to remove both the LFP and part of the noise. As visible in Figure 9, the recorded signal still contains background noise, as predicted by the model, with spikes embedded in it.

It may happen that some spikes have amplitude similar to the noise: this is the case of the one at about 40 msec

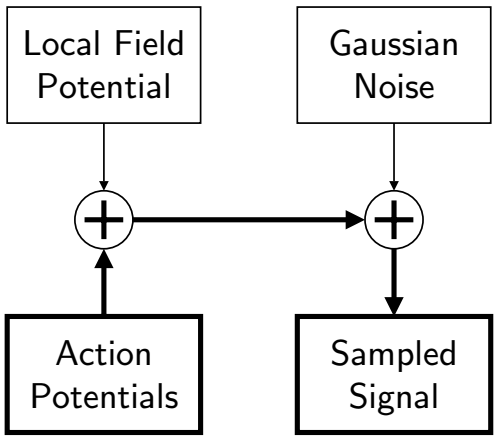


Figure 8: Classical extracellular recordings signal model; signal path from neurons to the electrodes has been highlighted.

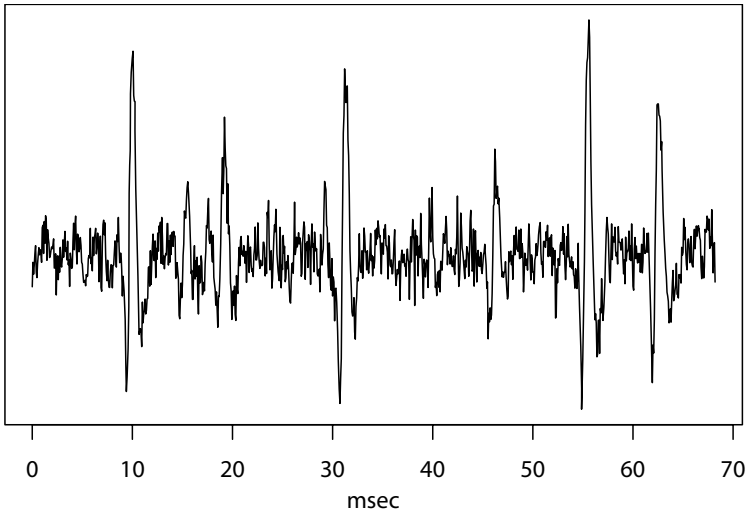


Figure 9: Extracellular waveform recorded from the locust (*Schistocerca americana*) antennal lobe filtered between 300 Hz and 5 kHz.

in Figure 9. Threshold-based detector will usually fail in detecting spikes like this one, unless a certain number of false spikes is allowed to leak in the subsequent elaboration stages. If a feasible denoising technique is applied to the ER signal before the detection stage, performance will improve, allowing to recover fainter spikes.

Basing on a model rather similar to the one depicted in Figure 8, many algorithms were designed in the field of Signal Processing [28] in order to recover a signal embedded in stationary white noise. As this model does not consider the LFP as a source of noise, algorithms that are build on it do not separate LFP from the AP sequence. This is not an issue in fields like EEG spike detection [27, 29], where researchers are interested in reconstructing both high frequency (Gamma, Beta, SRM) and low frequency waves (Alpha, Theta and Delta). On the contrary, these algorithms are not so effective when neurobiologists are interested in getting rid of LFP or separate them form the APs sequence.

Finally, it has been recently observed [30, 29, 31, 32, 33] that the noise in which the APs are embedded can be much more accurately modeled using non-stationary or signal dependent models.

Following these consideration, a flexible signal model has been developed, whose parameters allow to take into consideration different kinds of noise, as additive Gaussian, jitter and speckle.

2.4 EXTENDED SIGNAL MODEL

This model, depicted in Figure 10, simply extends the classical model, while preserving the additive structure and the basic components of the sampled signal. Terms have been grouped together in order to describe their origin: both LFP and AP originate directly from neurons and the extracellular environment and, as they are the primary signal source, they are grouped together in the *Neural Stage*. On the other hand, the *Noise Stage* encompasses both the white noise, whose presence is commonly accepted for many reasons, for example the thermal induced effects onto the recording electrodes, and the signal dependent component.

In fact, as the sampled signal cannot be perfectly aligned

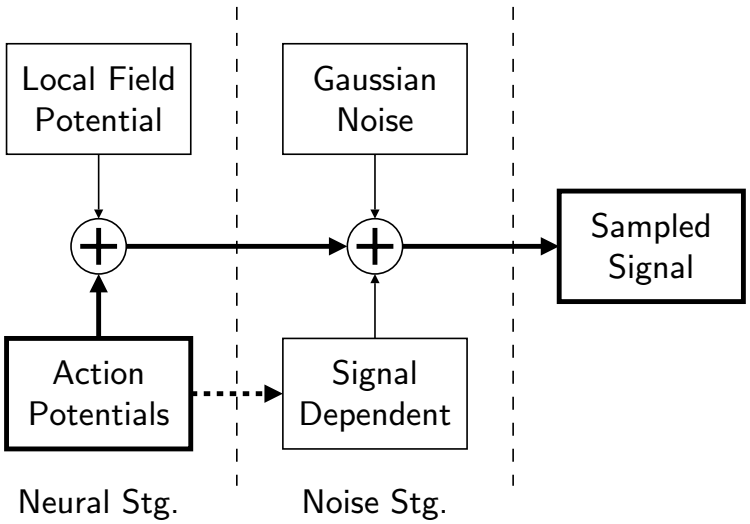


Figure 10: Extended signal model with signal dependent noise; contributes to the sampled signal are separated in two different stages, depending on their origin.

to each maximum of the AP sequence, AP generated by the same neuron could differ one from each other in the recorded signal: this effect can be well modeled by a jitter-like noise term, which is intrinsically signal dependent and should be considered beside the Gaussian noise term. Moreover, due to the interaction with the neighboring environment, most of the noise concentrates around the AP, showing a behavior much similar to a speckle-like noise term, which again is signal-dependent. Finally, since sampled signals are processed, quantization noise should be considered: this noise originates from the rounding error between the analog input voltage of the A/D converter and the output digitized value, it is strongly related to the number of quantization bits [34] and again it is non linear and signal dependent.

These considerations lead to the following signal model:

$$y[k] = x[k] + (x * f)[k]u[k] + v[k] \tag{2.1}$$

where the term the $v[k]$ represents the white noise while the signal dependent component $(x * f)[k]u[k]$ is modeled with a direct product between a stationary, uncorrelated noise u with given mean and variance and a filtered version of the clean signal.

The flexibility in the model is given by the filter f: in fact, it is possible to specialize (2.1) for different kinds of noise simply by adjusting the filter coefficients.

In particular, if f is chosen to be a first order derivative filter, (2.1) can be thought as the first order truncated Taylor series of

$$y[k] = x(kT_s + u[k]) + v[k] \tag{2.2}$$

centered on kT_s where $z(kT_s)$ has been substituted with the convolution of the clean signal with the filter.

Moreover, if f is set to be a Kronecker delta sequence, the convolution operation has no effect on x, leading to a model where the signal is in direct product with the stationary uncorrelated noise u. This is rather similar to a fully developed speckle model, this fact hinting that standard despeckling techniques too may achieve good denoising results in this case. Finally, if f is set to zero, the signal dependent noise term disappears, giving way to a standard additive Gaussian noise model.

Concerning AP signals, they are well localized both in time and frequency: they last a finite amount of time, a few milliseconds typically, and most of their energy is concentrated within finite bandwidth, usually no more than 1 kHz [18, 35]. Because of this, it is natural to search for mathematical operators that behaves well in both these domains in order to design proper denoising algorithms. From this standpoint, Wavelet Analysis (WA) is among the best techniques that can be used to elaborate signals of this kind, as it cuts up data into their different frequency components and then studies each component with a resolution matched to its scale.

3

WAVELETS

Any functional equation of the form

$$f(t) = \sum_{k=0}^{N} c_k f(2t - k) \tag{3.1}$$

is said *dilation equation* or *two-scale difference equation* and any non-zero solution $f(t)$ is called *scaling function* [36, 37]; coefficients $c_0, \ldots, c_N$ may be real or complex: if they are real then the scaling function $f(t)$ will be real-valued. It can be shown that there exists a square-integrable scaling function $f(t)$ if the coefficients $c_0, \ldots, c_N$ satisfy the following conditions:

$$\sum_k c_{2k} = \sum_k c_{2k+1} = 1 \tag{3.2}$$

and

$$\sum_k c_k \overline{c}_{k+2j} = \begin{cases} 2, & \text{if } j = 0 \\ 0, & \text{if } j \neq 0 \end{cases} \tag{3.3}$$

where it is assumed $c_k = 0$ if $k < 0$ or $k > N$ [38, 39].

Under these conditions it is defined *wavelet function* [40] or simply *wavelet* the following function:

$$g(t) = \sum_{k=0}^{N} (-1)^k c_{N-k} f(2t-k) \tag{3.4}$$

while the functions

$$g_{a,b}(t) = \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right) \tag{3.5}$$

are obtained by dilation and translation from $g(t)$.

In case $f(t)$ satisfies

$$\int_{-\infty}^{+\infty} f(t)\bar{f}(t-k)dt = 0, \quad \forall k \neq 0 \tag{3.6}$$

then the scaling function $f(t)$ is orthogonal to its integer translate, and the functions $g_{2^j,2^j k}(t)$ form an orthonormal base for $L_2(\mathbb{R})$ after a proper normalization of $g(t)$; this property is central in the process of signal reconstruction [41, 42].

3.1 WAVELET TRANSFORM DEFINITION

Given a signal $x(t)$ with time-variant spectrum the Continuous Wavelet Transform (CWT) is defined as the orthogonal projection of $x(t)$ on the functional space defined by the functions $g_{a,b}(t)$ [41]:

$$CWT(x, g; a, b) = \langle x | g_{a,b} \rangle = \int_{-\infty}^{+\infty} x(t)\overline{g}_{a,b}(t)dt \tag{3.7}$$

where $a \in \mathbb{R}_+ \setminus \{0\}$ is the scale or dilation parameter and $b \in \mathbb{R}$ is the time or translation parameter.

For computational reasons, different sampling methods have been applied to the domain of the dilation and translation parameters: each method leads to a different wavelet transformation technique, to improve time-frequency resolution or computational efficiency; examples of these sampling methods are the Discrete Wavelet Transform (DWT) [41], the Stationary Wavelet Transform (SWT) [43] and the Quasi-Continuous Wavelet Transform (QCWT) [42].

In the DWT the scale and time parameters are defined respectively as $a = 2^j$ and $b = 2^j k$ where $j, k \in \mathbb{Z}$: the result is a very fast implementation [41, 43] that does not introduce any form of redundancy but has a very poor time-scale resolution at the higher scales. This fact introduces an upper limit in the scale domain: it can be shown that a strong relationship exists between the number of samples of the signal $x(t)$ and the scale bound.

The SWT tries to avoid the poor time-scale resolution keeping a low computational weight [43]: this is achieved by defining the scale and time parameters respectively as $a = 2^j$ and $b = k$ where $j, k \in \mathbb{Z}$; an efficient implementation of this technique consists in a filter bank where each filter is the dyadic up-sampled version of the previous one.

The QCWT is an even better solution to the poor time-scale resolution of the DWT but it is computationally heavier than the SWT [42]. Scale and time parameters are defined as $a \in \mathbb{R}_+ \setminus \{0\}$ and $b \in \mathbb{Z}$: this means that only the time parameter has been sampled, while the scale parameter can be set at designer's will. Like the SWT, an efficient implementation of the QCWT uses a filter bank but direct relations between filters cannot be always shown.

3.2 DISCRETE WAVELET TRANSFORM

The DWT [41, 43] is the most efficient method to compute a Wavelet Transform (WT) but suffers of very low time-scale resolution problems; anyway, whenever very high speed and low resource usage is needed, this method is to be taken in consideration. The DWT defines the scale and time parameter as $a = 2^j$ and $b = 2^j k$ where $j, k \in \mathbb{Z}$ so that $g_{a,b}(t)$ can be written in the form

$$g_{j,k}(t) = 2^{-j/2} g(2^{-j} t - k) \tag{3.8}$$

An auxiliary function

$$f_{j,k}(t) = 2^{-j/2} f(2^{-j} t - k) \tag{3.9}$$

is also defined. Defining a new set of coefficients $d_k = (-1)^k c_{N-k}$, it is possible to write both the expressions of

$f(t)$ and $g(t)$ in a similar form:

$$f(2^{-1}t) = \sum_{k=0}^{N} c_k f(t-k) \tag{3.10}$$

and

$$g(2^{-1}t) = \sum_{k=0}^{N} d_k f(t-k) \tag{3.11}$$

Applying the induction principle to (3.8) and (3.9) the following is obtained:

$$\begin{aligned} g_{j,k}(t) &= 2^{-1/2} g_{j-1,k}(2^{-1}t) \\ &= \textstyle\sum_n 2^{-1/2} d_{n-2k} f_{j-1,n}(t) \end{aligned} \tag{3.12}$$

and symmetrically

$$\begin{aligned} f_{j,k}(t) &= 2^{-1/2} f_{j-1,k}(2^{-1}t) \\ &= \textstyle\sum_n 2^{-1/2} c_{n-2k} f_{j-1,n}(t) \end{aligned} \tag{3.13}$$

The results of the projection of the signal $x(t)$ on the functional space generated by the family of functions $f_{j,k}(t)$ are called *approximation coefficients*:

$$A_{j,k} = \int_{-\infty}^{+\infty} x(t) f_{j,k}(t) dt \tag{3.14}$$

while the *detail coefficients* are the result of the projection of the signal $x(t)$ on the functional space generated by the family of functions $g_{j,k}(t)$:

$$D_{j,k} = \int_{-\infty}^{+\infty} x(t) g_{j,k}(t) dt \tag{3.15}$$

To avoid nomenclature issues, from here on the wavelet coefficients will be defined as follow:

$$L_k = 2^{-1/2} c_k \quad H_k = 2^{-1/2} d_k \tag{3.16}$$

where L stands for low-pass and H stands for high-pass.

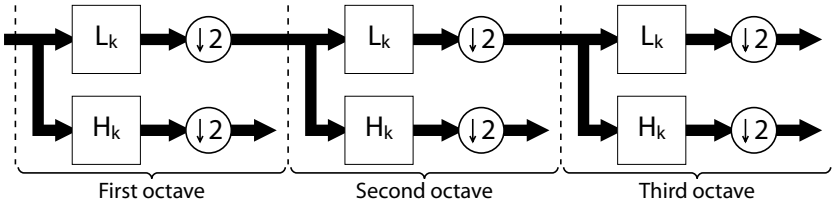


Figure 11: Block diagram of the DWT algorithm

This straightly leads to the definition of the algorithm used for the calculus of the DWT:

$$\begin{aligned} A_{0,k} &= \int_{-\infty}^{+\infty} x(t) f(t-k)\,dt \\ A_{j,k} &= \sum_n L_{n-2k} A_{j-1,n} \\ D_{j,k} &= \sum_n H_{n-2k} A_{j-1,n} \end{aligned} \tag{3.17}$$

The convolutions at each stage have a step equal to 2: this can be easily accomplished in hardware by applying a standard convolution with step equal to 1 and discarding the even numbered elements in each $A_{j,k}$ sequence.

The scale parameter j defines the octave at which the transform is computed; each hardware stage implements an octave and is composed by two FIR filters with coefficients L_k and H_k respectively for the approximation and the detail channel and a dyadic down-sampler that operates only on the approximation channel; each stage takes its inputs from the previous stage approximation output channel. The resulting structure is shown in Figure 11.

3.3 STATIONARY WAVELET TRANSFORM

The SWT method basically tries to augment the time-scale resolution keeping a low computational weight [43]: it is a very good compromise solution that should be taken in consideration whenever high frequency sampling is combined with low frequency signal analysis. Despite it resembles the DWT, its complexity grows exponentially with the number of octaves.

The SWT defines the scale and time parameter as $a = 2^j$ and $b = k$ where $j, k \in \mathbb{Z}$: this allow to write $g_{a,b}(t)$ in the form

$$g_{j,k}(t) = 2^{-j/2} g(2^{-j}(t-k)) \tag{3.18}$$

As well as the DWT, the following auxiliary function is defined:

$$f_{j,k}(t) = 2^{-j/2} f(2^{-j}(t-k)) \tag{3.19}$$

Expressions (3.10) and (3.11) hold also in the case of the SWT; applying the induction principle to (3.18) and (3.19) the following holds:

$$\begin{aligned} g_{j,k}(t) &= 2^{-1/2} g_{j-1,k/2}(t/2) \\ &= \textstyle\sum_n 2^{-1/2} d_{2^{-(j-1)}(n-k)} f_{j-1,n}(t) \end{aligned} \tag{3.20}$$

and symmetrically

$$\begin{aligned} f_{j,k}(t) &= 2^{-1/2} f_{j-1,k/2}(t/2) \\ &= \textstyle\sum_n 2^{-1/2} c_{2^{-(j-1)}(n-k)} f_{j-1,n}(t) \end{aligned} \tag{3.21}$$

Using the expressions (3.14), (3.15) and (3.16) it is possible to define the algorithm used for the evaluation of the SWT:

$$\begin{aligned} A_{0,k} &= \textstyle\int_{-\infty}^{+\infty} x(t) f(t-k)\,dt \\ A_{j,k} &= \textstyle\sum_n L_{2^{-(j-1)}(n-k)} A_{j-1,n} \\ D_{j,k} &= \textstyle\sum_n H_{2^{-(j-1)}(n-k)} A_{j-1,n} \end{aligned} \tag{3.22}$$

In this case too non-standard convolutions are present, whose structure is:

$$\sum_n a\left[\frac{n-k}{2^\alpha}\right] \cdot b[n] = \sum_m a[m] \cdot b[2^\alpha m + k] \tag{3.23}$$

This kind of convolution can be implemented in hardware by inserting $2^\alpha - 1$ zeros between each element of $a[m]$ and applying a standard convolution with step equal to 1.

The SWT has the same structure of the DWT [41, 43] as each hardware stage implements an octave and is composed by two Finite Impulse Response (FIR) filters respectively for the approximation and the detail channel; in this case however the filters coefficients are not constant throughout the whole system: in fact at each stage an exponentially growing number of zeros are inserted between the elements of L_k and H_k. The resulting structure is shown in Figure 12.

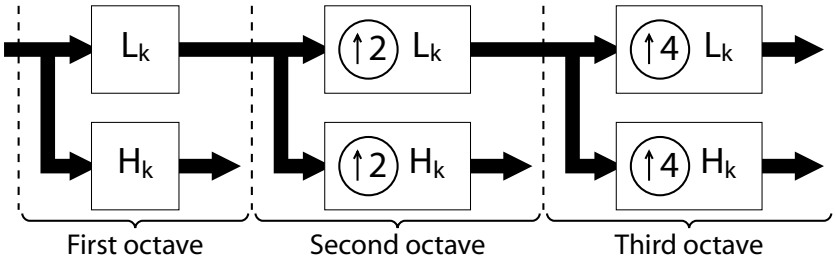


Figure 12: Block diagram of the SWT algorithm

3.4 QUASI-CONTINUOUS WAVELET TRANSFORM

The QCWT uses a totally different approach to overcome the poor time-scale resolution of the DWT [42]: first of all it uses a filter bank where filters are each other independent and each one is used to compute a single scale; second, there are no approximation channels as each stage produces directly the final result.

As it was discussed in the previous sections, scale and time parameters are defined as $a \in \mathbb{R}_+ \setminus \{0\}$ and $b \in \mathbb{Z}$: this shows that only the time parameter is sampled – a fact very useful in digital systems, where the data channel does not bear a time-continuous information but is a sampled version of a time-continuous signal – while the scale parameter retains his domain.

Using the relations that bounds the scale function to the wavelet function it can be shown that, for a proper set of coefficients $q_a[n]$, the following equality holds:

$$a^{-1/2} g(a^{-1}t) = \sum_n q_a[n] f(t-n) \tag{3.24}$$

Whenever $f(t)$ is orthogonal to its integer translates it is possible to calculate the coefficients $q_a[n]$:

$$q_a[n] = \frac{\langle a^{-1/2} g(a^{-1}t) | f(t-n) \rangle}{\langle f(t) | f(t) \rangle} \tag{3.25}$$

All the elements to define the QCWT algorithm being known, the resulting structure is given by:

$$\begin{aligned} D_{a,b} &= \textstyle\sum_n A_{0,n} \cdot q_a[n-b] \\ A_{0,n} &= \textstyle\int_{-\infty}^{+\infty} x(t) f(t-b)\,dt \end{aligned} \tag{3.26}$$

It is therefore quite simple to devise a system based on the QCWT algorithm: in fact the whole system consists of a bank of filters whose coefficients can be precalculated with the use of (3.25) providing that $f(t)$ is orthogonal to its integer translates [38]; in this case the orthogonality of $f(t)$ is fundamental for both analysis and reconstruction.

3.5 SCALING FUNCTION GENERATION

According to what it is said in [40], if the value of the scaling function at the integers are known, then the dilation equation determines the values of the scaling function at the half-integers and, by recursion, at every *dyadic point* $x = 2^{-n}k$ where $n \in \mathbb{N}$ and $k \in \mathbb{Z}$.

This observation is the key of the *dyadic interpolation method*: Daubechies and Lagarias [44] and Micchelli and Prautzsch [45] independently developed this recursion using products of two $N \times N$ matrices built with the coefficients $c_0, \ldots, c_N$ to calculate the scaling function at every dyadic point.

Given an orthogonal set of coefficients $c_0, \ldots, c_N$ two $N \times N$ matrices $(T_0)_{i,j} = c_{2i-j-1}$ and $(T_1)_{i,j} = c_{2i-j}$ are build as follow:

$$T_0 = \begin{pmatrix} c_0 & 0 & 0 & \cdots & 0 & 0 \\ c_2 & c_1 & c_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & c_N & c_{N-1} \end{pmatrix} \tag{3.27}$$

and

$$T_1 = \begin{pmatrix} c_1 & c_0 & 0 & \cdots & 0 & 0 \\ c_3 & c_2 & c_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & c_N \end{pmatrix} \tag{3.28}$$

Each of them contains the $(N-1) \times (N-1)$ sub-matrix

$M_{i,j} = c_{2i-j}$:

$$M = \begin{pmatrix} c_1 & c_0 & 0 & \cdots & 0 & 0 \\ c_3 & c_2 & c_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & c_N & c_{N-1} \end{pmatrix} \tag{3.29}$$

By letting $\mathrm{supp}(f(t)) \subset [0, N]$ the function $v\colon [0,1] \to \mathbb{R}^N$ is defined as

$$v(x) = \begin{pmatrix} f(x) \\ f(x+1) \\ \vdots \\ f(x+N-1) \end{pmatrix} \tag{3.30}$$

If $f(x)$ is continuous and compact supported it shall be $f(0) = f(N) = 0$: under these condition it is possible to demonstrate that $v(x)$ has the following properties [40]:

$$\begin{aligned} v(0) &= T_0 v(0) \\ v(1) &= T_1 v(1) \\ v(1/2) &= T_1 v(0) = T_0 v(1) \end{aligned} \tag{3.31}$$

For every dyadic x in the interval $(0,1)$, its *upper binary expansion* is defined as

$$x = \sum_{i=1}^{m} d_i \cdot 2^{-i} \tag{3.32}$$

where $d_i \in \{0,1\}$ and $m < K \in \mathbb{N}$.

If all of the above conditions hold, it can be shown [40] that

$$v(x) = T_{d_1} \cdots T_{d_m} v(0) \tag{3.33}$$

where $v(0) = (0, a_1, \ldots, a_{N-1})^T$, with $a = (a_1, \ldots, a_{N-1})^T$ being the right eigenvector for M for the eigenvalue 1. Because of the orthogonality of the coefficient set, M necessarily has $(1, \ldots, 1)$ as left eigenvector for the eigenvalue 1: this means that at least one possible choice for a for every orthogonal coefficient set is present.

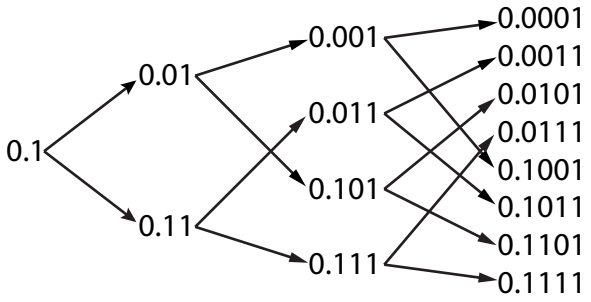


Figure 13: Binary tree used to cover dyadic points $x \in (0, 1)$

Once the starting vector has been defined, it is possible to start the iterative method needed to define the scaling function at every dyadic point; as shown in Figure 13, it is possible to build a binary tree that contains every dyadic point $x \in (0, 1)$: the naming convention of each node within the tree along with its upper binary expansion is shown in the same figure.

Following the arrows it is possible to see that each element ends with 1 and is obtained from the previous one by simply introducing a 0 or a 1 after the comma; furthermore, adding a 0 always brings to the upper half of the tree, while a 1 takes to the lower half; finally, by using all the nodes in the tree, the maximum coverage of the considered interval is obtained.

Using this tree, the recursion is intuitively defined: the vector $T_1 v(0)$ is computed, corresponding to dyadic point 0.1, then the matrices T_0 and T_1 are used to obtain the other nodes in the tree; in this way, every step in the loop produce a useful result.

3.6 GENERATION OF WAVELET FILTERS COEFFICIENTS

Given an orthogonal set of coefficients $c_0, \ldots, c_N$, the generation of the coefficient set for the DWT and SWT FIR filters is quite straightforward, while the QCWT requires some consideration of algebraic nature.

For the DWT and SWT the filters were defined as

$$\begin{aligned} L_k &= 2^{-1/2} c_k \\ H_k &= 2^{-1/2} d_k \end{aligned} \tag{3.34}$$

where $d_k = (-1)^k c_{N-k}$: L_k and H_k form a pair of *quadrature mirror filters* and are fully defined as soon as the wavelet coefficients are defined.

The QCWT uses instead the following filter definition:

$$q_a[n] = \frac{\langle a^{-1/2} g(a^{-1}t) | f(t-n) \rangle}{\langle f(t) | f(t) \rangle} \tag{3.35}$$

This means that both the scaling and the wavelet function have to be computed in order to determine the set of coefficient of the FIR filter.

If $f(t)$ has been already normalized, it is possible to rewrite (3.35) as follows

$$q_a[n] = \langle a^{-1/2} g(a^{-1}t) | f(t-n) \rangle \tag{3.36}$$

that is

$$q_a[n] = \int_{-\infty}^{+\infty} a^{-1/2} g(a^{-1}t) \bar{f}(t-n) dt \tag{3.37}$$

Since both $f(t)$ and $g(t)$ are compactly supported, it is possible to restrict the integration to supports intersection but, as a grows, the support of $g(t)$ grows too: this is disadvantageous from the computational point of view if both $f(t)$ and $g(t)$ had been generated by recursion. In fact, in this case, $g(t)$ should be interpolated in order to compute $q_a[n]$ with the right precision.

However it is possible to elaborate (3.37) and swap the dilation for the translation:

$$q_a[n] = \int_{-\infty}^{+\infty} a^{1/2} g(t + a^{-1}n) \bar{f}(an) dt \tag{3.38}$$

In this way, as a grows, the support of $f(t)$ shrinks and there is no more need of further elaboration on either $f(t)$ or $g(t)$; however there still is an upper limit for a if both $f(t)$ and $g(t)$ had been generated by recursion since it can be shown that, in this case, for any given number of iterations it exists a value for a such that the error on $q_a[n]$ is greater than the computed value for $q_a[n]$.

From now on it is assumed that both $f(t)$ and $g(t)$ were generated by recursion: $f(t)$ and $g(t)$ are said to have been calculated with *precision* α when the interval $[0, 1]$ has been

divided into 2^α parts: in this case it is possible to rewrite (3.38) as follows

$$q_a[n] = \sum_{i=-\infty}^{+\infty} a^{1/2} g_d[2^\alpha(i\Delta t + a^{-1}n)] \cdot \bar{f}_d[2^\alpha(ai\Delta t)]\Delta t \quad (3.39)$$

where $g_d[n] = g(2^{-\alpha}n)$ and $f_d[n] = f(2^{-\alpha}n)$.

Simplification of (3.39) is feasible if $\Delta t = 2^{-\alpha}$ is chosen: because of this it follows that

$$q_a[n] = 2^{-\alpha} a^{1/2} \sum_{i=-\infty}^{+\infty} g_d[i + n \cdot 2^\alpha/a)] \cdot \bar{f}_d[a \cdot i] \quad (3.40)$$

but $supp(f(t)) \subset [0, N]$ so the sum extents can be adjusted in this way

$$q_a[n] = 2^{-\alpha} a^{1/2} \sum_{i=0}^{N\cdot 2^\alpha/a} g_d[i + n \cdot 2^\alpha/a)] \cdot \bar{f}_d[a \cdot i] \quad (3.41)$$

that originates a filter with $N \cdot a$ coefficients.

However it has to be noted that if $f(t)$ and $g(t)$ are known in closed form, any set of coefficient can be computed exactly; for example, by considering the Haar function, it is possible to write

$$q_a[k] = \begin{cases} a^{-1/2} & :k \in [0, \lfloor a/2 \rfloor) \\ a^{-1/2}(a - 2\lfloor a/2 \rfloor - 1) & :k = \lfloor a/2 \rfloor \\ -a^{-1/2} & :k \in (\lfloor a/2 \rfloor, \lfloor a \rfloor) \\ -a^{-1/2}(a - \lfloor a/2 \rfloor) & :k = \lfloor a \rfloor \\ 0 & :k \notin [0, \lfloor a \rfloor] \end{cases} \quad (3.42)$$

for the QCWT, while for the SWT and the DWT it simply is

$$L_k = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\} \quad H_k = \left\{ \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right\} \quad (3.43)$$

Part II

TOOLS

4

IMPULSE RESPONSE ESTIMATION

As it has been discussed before, it is of crucial importance to remove the deformation introduced by the transducer on the input signal. While this conversion can be skipped in neural signal processing as electrical signals are already involved, it is mandatory when treating echographic signals.

In order to remove this deformation, the need to first estimate the acquisition system impulse response is present. This chapter focuses on how to obtain this estimate from the received signal itself, without the need of an experimental setup. In fact, this information is usually obtained acquiring the incoming signal from a test sample whose response to the acquisition system is well known thanks to theoretical considerations.

A common example of this measuring technique is the insonification of a metal wire sank in a water filled tank for the estimation of an echographic system impulse response. The water does not significantly alter the traveling echo pulse due to its low attenuation; at the same time, the metal wire behaves almost as an ideal acoustic reflector, mimicking a Dirac delta behavior.

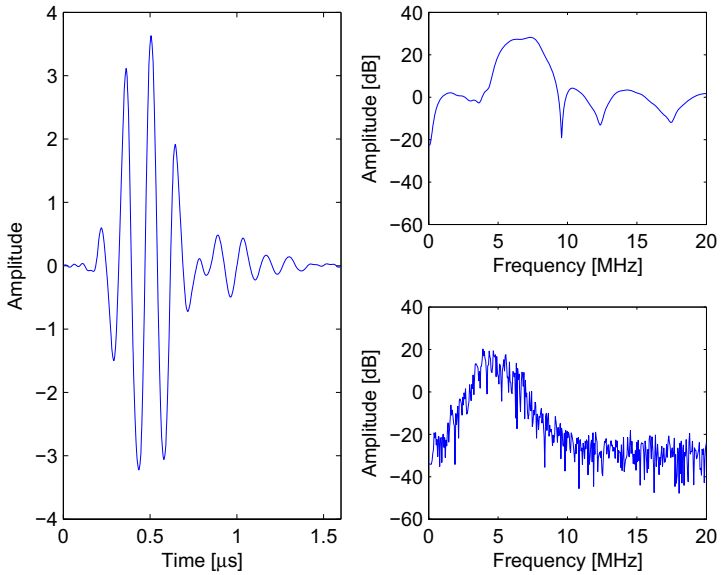


Figure 14: Acquisition system impulse response (left) and spectrum (top-right) recorded from a water tank experiment. Impulse spectrum is much smoother than a typical incoming signal spectrum (bottom-right) recorded from *in-vivo* tissues

With reference to the echo signal model (1.2), this means that $x(t) = \delta(t)$, with a good degree of approximation. This also means that the sampled incoming echo signal $y(n)$ is equal to the acquisition system impulse response $h(n)$, except for some unavoidable noise $v(n)$. This setup is easily reproducible and affordable, however it does not consider any time dependent behavior of the acquisition system; anyway it gives qualitative informations about the impulse response shape and its spectrum.

As shown in Figure 14, the echo pulse amplitude spectrum is quite smooth and assumes its maximum approximatively in correspondence to the transducer central frequency. On the contrary, the incoming signal spectrum is quite peaky and its maximum shifts toward lower frequencies due to the frequency dependent attenuation.

The next few sections will discuss how it is possible to exploit these differences to estimate what it is called the *local impulse response* directly from the incoming tissue echo, without requiring a specifically designed experimen-

tal setup.

4.1 HOMOMORPHIC DECONVOLUTION

With reference to the signal model (1.2), if the noise term is neglected, the incoming signal $y(t)$ can be thought as the convolution of the tissue impulse response $x(t)$ and the imaging system impulse response $h(t)$.

By taking the Fourier Transform of $y(t)$ the usual product in the frequency domain is obtained:

$$Y(\omega) = H(\omega)X(\omega) \tag{4.1}$$

This domain change allows to better exploit the evident differences in the spectra shown in Figure 14 in order to estimate the imaging system impulse response from the incoming signal.

If $H(\omega)$ is supposed to be minimum phase [6, 46, 47, 48], all the informations regarding the shape of the imaging system impulse response are contained in the amplitude spectra. By taking the logarithm of the amplitude of (4.1), the next equality holds:

$$\log\{|Y(\omega)|\} = \log\{|H(\omega)|\} + \log\{|X(\omega)|\} \tag{4.2}$$

where two additive components are clearly highlighted. This is the so-called *log-spectrum* of $y(t)$.

Recalling the fact that $H(\omega)$ is observed to be much smoother than $X(\omega)$, it is possible to obtain a non-minimum phase approximation of $h(t)$ by considering $\log\{|Y(\omega)|\}$ as a signal and extracting its low-pass component. This procedure is equivalent to selecting the low-quefrency N_c samples of the cepstrum [49] of $y(t)$, where N_c is the so-called *cut-off parameter*.

The minimum phase version of $h(t)$ can be easily estimated in the cepstral domain [50, 51, 52]: in fact, calling C_x the cepstrum of a real valued N-samples input signal x, its minimum phase version $\hat{C}_x$ is given by

$$\hat{C}_x(n) = \begin{cases} C_x(n) & n = 0 \\ 2\mathfrak{R}\{C_x(n)\} & 1 \leqslant n \leqslant \frac{N}{2} \\ 0 & \frac{N}{2} < n \leqslant N-1 \end{cases} \tag{4.3}$$

where $\Re\{x\}$ stands for the real part of x.

It is now straightforward to layout the homomorphic deconvolution estimate of the imaging system impulse response. First, the cepstrum C_y of the received ultrasonic signal y is taken

$$C_y = \mathcal{F}^{-1}\{\log|\mathcal{F}\{y \cdot w\}|\} \tag{4.4}$$

where w is a proper windowing function necessary to remove the undesired border effects in the inner Fourier transform: typically an Hamming function is used.

Then, by plugging the cut-off parameter into (4.3), the following is obtained

$$\hat{C}_h(n) = \begin{cases} C_y(n) & n = 0 \\ 2\Re\{C_y(n)\} & 1 \leqslant n \leqslant N_c \\ 0 & N_c + 1 \leqslant n \leqslant N - 1 \end{cases} \tag{4.5}$$

from which the time domain minimum phase estimate $\hat{h}$ of the impulse response can be obtained by inverting the cepstrum

$$\hat{h} = \mathcal{F}^{-1}\{\exp(\mathcal{F}\{\hat{C}_h\})\} \tag{4.6}$$

It should be remarked that it is not feasible to recover the tissue impulse response $x(t)$ using this method: the minimum phase assumption used for $h(t)$ does not hold for $x(t)$. More advanced deconvolution techniques must be exploited in order to recover $x(t)$, some of which will be discussed in the next chapter.

Figure 15 features a comparison between three different estimations of $\hat{h}$ for different values of the cut-off parameter. All of them are smooth and feature a central frequency around 5 MHz; their spectra are quite similar, with the curve corresponding to $N_c = 10$ being almost identical to the one related to $N_c = 20$. In the time domain, the estimation corresponding $N_c = 10$ has a shorter tail: however, all the curves behave almost in the same way for the first 0.4 μs.

It is usually up to the user experience to set the cut-off parameter to a proper value, to balance between spectral estimation accuracy and smoothness. In the field of ultrasonic processing, $N_c = 20$ is usually accepted as a typycal value for impulse response cepstral estimation.

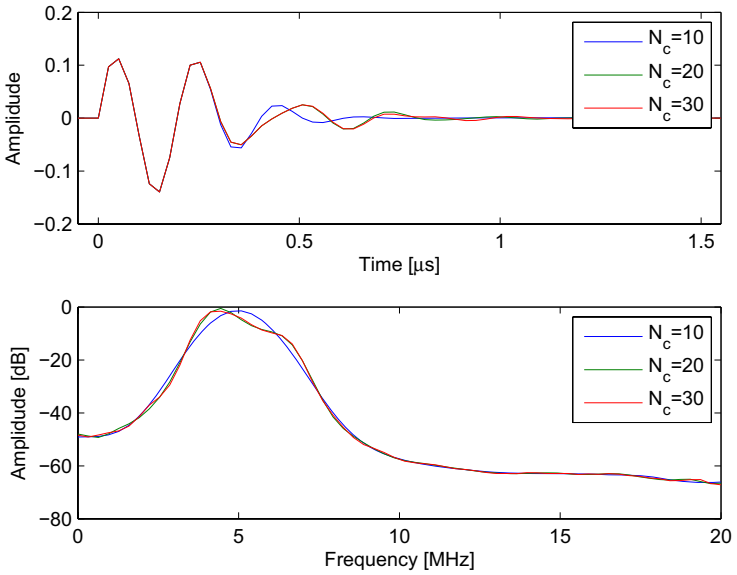


Figure 15: Comparison of cepstrum minimum phase estimations for different values of N_c; time domain impulse responses are shown on top, while the corresponding amplitude spectra ar shown on bottom.

4.2 WAVELET-BASED BLIND DECONVOLUTION

Some of the reconstruction properties of the Wavelet Transform can be exploited in order to further the estimation of the acquisition system impulse response. In particular, polynomial reconstruction is quite useful when trying to estimate a smooth polynomial function from its noisy samples.

In fact, given an input signal $s(t)$ made up only of a K order polynomial component $q(t)$, if a Wavelet $\phi(t)$ with $K+1$ vanishing moments is used to decompose it, identically zero detail coefficients will be obtained. Due to the superposition principle, if an additive white noise term $\nu(t)$ is added to $s(t)$, the detail coefficients at each scale of $s(t)$ will correspond to the ones of $\nu(t)$. Moreover, as the effective bandwidth at each analysis scale reduces, so does the noise power in the approximated signals.

This means that, given a proper Wavelet with the right number of vanishing moments, it is possible to recover an almost perfect estimation of the polynomial component in

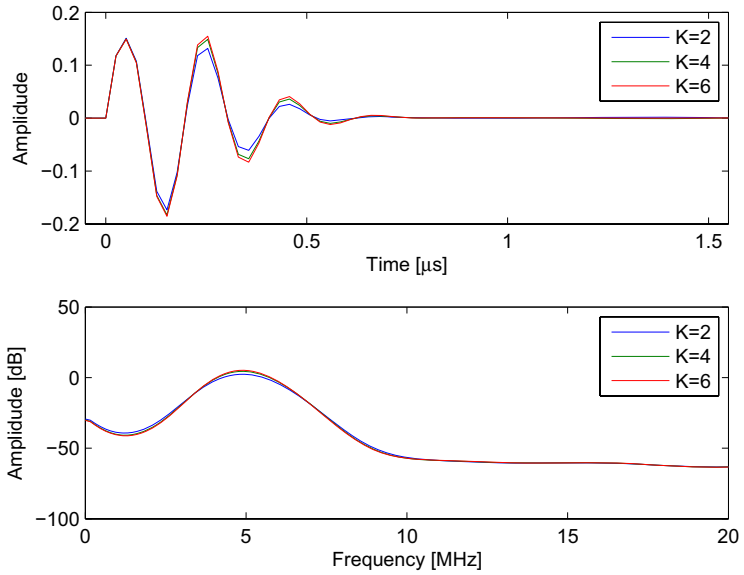


Figure 16: Comparison of Wavelet minimum phase estimations for different values of K. Both time and frequency domain are quite insensitive to this parameter.

$s(t)$ simply by filtering it through a proper number M of octaves.

Here, instead of having only one parameter as in the previous method, two parameters can be used to tune the impulse response estimation algorithm: K, the number of vanishing moments of $\phi(t)$, and M, the number of octaves in the Wavelet analysis.

The resultant algorithm is quite simple: after estimating the log-spectrum of $y(t)$, a non-minimum phase version of the log-spectrum of $h(t)$ is obtained calculating an M scale Wavelet Transform and dropping any detail coefficient before inverting the transform. Finally the minimum phase version is estimated by making use of (4.3).

Figure 16 compares three different minimum phase estimations of $h(t)$ obtained using this method while varying the number of vanishing moments of the analysis Wavelet. Symlets [53] with a 2, 4 and 6 vanishing moments were used in this comparison. It appears that this parameter does not alter significantly the impulse response estimated shape both in the time and frequency domain.

On the other hand, Figure 17 highlights a problem known

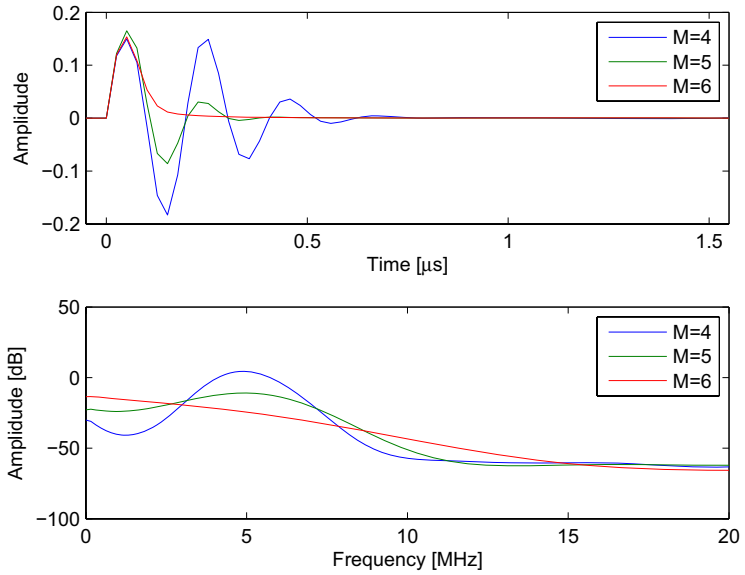


Figure 17: Comparison of Wavelet minimum phase estimations for different values of M. The number of octaves must be carefully chosen in order to avoid oversmoothing.

as *oversmoothing* [6]: using too many octaves in the Wavelet analysis stage removes too much details from the impulse response. The net result is a curve which is way too smooth (hence *oversmoothed*) than the desired response, both in the time and frequency domain. The opposite problem (*undersmoothing*) is caused by a too coarse Wavelet analysis, where spectrum noise leaks into the impulse response.

4.3 COMBINED WAVELET/CEPSTRUM BLIND DECONVOLUTION

These two techniques can be successfully combined in a mixed Wavelet/cepstrum estimation. Both of them can be viewed as a low pass filtering of the log-spectrum signal: while the first operates as an ideal box filter in the cepstrum domain, the second uses a set of weights to attenuate the high frequency noise component.

This smoothing effect is quite useful, as it allows the user to avoid discontinuities once the impulse response is brought back to the time domain. However, as Wavelet basis

functions have a compact time-domain support, they have an infinite frequency domain support. This means than a certain amount of high-frequency noise always leaks into the impulse response estimate.

Even if negligible at visual inspection, this component reduces the minimum distance of the zeros of the Z-transform $H(z)$ of $h(t)$ from the unitary circle, making these estimates less apt for signal deconvolution.

One possible solution to this problem is to approximate the weight function originating from the Wavelet analysis with a function which is non-zero for the first N_c samples, and zero elsewhere. Here, once K and M have been set, N_c should be chosen in order to get a not too coarse approximation of the Wavelet weight function.

The resultant algorithm is quite similar to what discussed in Section 4.1: first the cepstrum C_y of the received ultrasonic signal y is calculated using (4.4); then the cepstrum of the acquisition system impulse response is estimated using

$$\hat{C}_h(n) = \begin{cases} C_y(n)W(n) & n = 0 \\ 2\Re\{C_y(n)W(n)\} & 1 \leqslant n \leqslant N_c \\ 0 & N_c + 1 < n \leqslant N - 1 \end{cases} \tag{4.7}$$

where $W(n)$ are the weights obtained from the Inverse Fourier Transform of the equivalent filter corresponding to the Wavelet analysis with parameters K and M. Finally the time domain minimum phase estimate $\hat{h}$ is calculated inverting the cepstrum $\hat{C}_h(n)$ with (4.6).

As shown in Figure 18, the cut-off parameter contributes to the smoothing of the impulse spectrum too. However, if its value is set too small, undesired ringings appear in the frequency domain and the time domain response decreases both in amplitude and duration.

In this example a second order Symlet is selected as analysis Wavelet ($K = 2$) and the analysis is carried up to the forth octave ($M = 4$): comparing Figure 16 with Figure 18, $N_c = 15$ appears to be a good setting for the cut-off parameter.

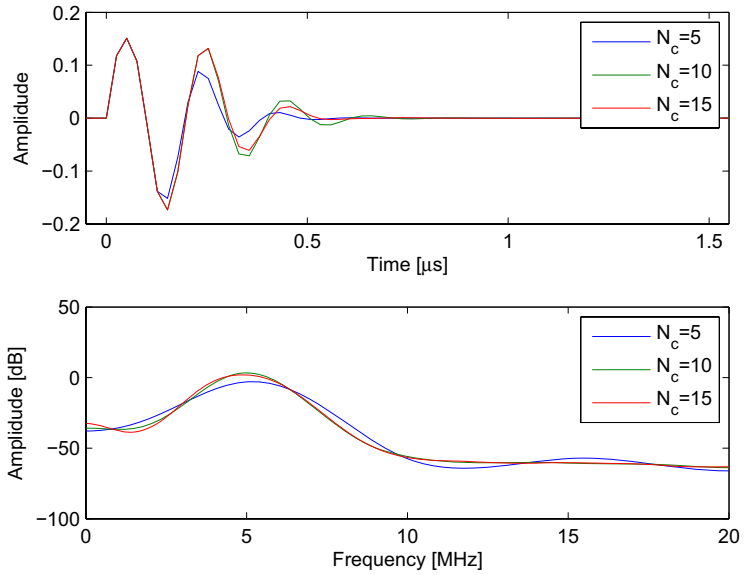


Figure 18: Comparison of combined minimum phase estimations for different values of N_c; too small values of the cut-off parameter generate unwanted spectral oscillations.

4.4 PERFORMANCE COMPARISON

Estimation algorithms are usually evaluated in terms of accuracy, comparing the estimated signal with the true one. However, it is not possible to perform this check for pulse estimation algorithms: the true acquisition system impulse response cannot be acquired, even in the water-filled tank experiment.

In fact, the metal wire has a non negligible diameter, which means that the probing signal encounters two – and not one – interfaces: the first between water and metal, the second between metal and water. As a consequence, two usually overlapped pulses will be recorded by the imaging system. Moreover, as each interface may occur at a depth which does not correspond to a point belonging to the sampling grid, the hypothesis that both these pulses have a non-minimum phase must be considered.

Figure 19 shows an example of deconvolution of the incoming signal: the pulse used in this case is extracted from the original signal using the combined Wavelet/cepstrum estimation technique. This pulse can be fruitfully used to

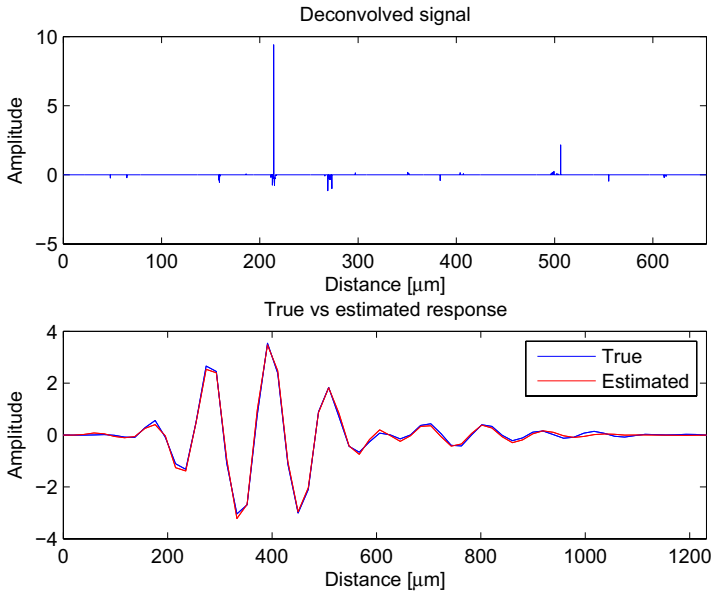


Figure 19: Top: rater-tank experiment deconvolved pulse; the two interfaces are clearly visible at 200µm and 500µm. Bottom: reconstruction of the incoming signal using minimum-phase estimates.

detect the presence of the two interfaces, respectively at 200µm and 500µm, by means of a proper iterative deconvolution technique, described in Chapter 5. At the same time, it allows for the reconstruction of the incoming signal, with a good accuracy.

Basing on this evidence, the performance of the deconvolution techniques described in this chapter are evaluated using the following metrics:

- the MSE between the water-tank recorded pulse and the one estimated after two deconvolution step;
- the minimum distance from the unitary circle the of zeros of the filter corresponding to the estimated impulse;
- the number of coefficient which retain the 99.9% of the corresponding estimated impulse response energy.

The first one estimates how good is the estimated impulse response at catching the shape of the water-tank recorded

Method	Parameters	MSE [dB]	Distance	Length
CEP	$N_c = 30$	−18.1	0.0126	41
	$N_c = 2$	−5.1	0.5784	3
WAV	$K = 8, M = 1$	−18.4	0.0035	38
	$K = 8, M = 8$	−5.0	0.9154	2
	$K = 1, M = 8$	−5.0	0.8785	2
MIX	$K = 4, M = 2, N_c = 32$	−18.4	0.0132	42
	$K = 8, M = 8, N_c = 3$	−5.0	0.9154	2
	$K = 1, M = 6, N_c = 2$	−5.1	0.4716	2

Table 2: A performance comparison of some significant setups for the different impulsive response estimation algorithms: cepstrum based (CEP), Wavelet based (WAV), combined Wavelet/cepstrum (MIX)

pulse from it's deconvolution: by doing only two processing steps, the deconvolution system is forced to consider only the two most prominent estimated interfaces.

The second one helps in evaluating how stable a deconvolution technique based on the corresponding impulse response is going to be: as filter inversion is typically used in a deconvolution process, the higher the distance, the higher the filtering stability.

The third one is connected to how many computational resources should be allocated to perform the corresponding deconvolution: the fewer this number, the less multiplier units can be used in an hardware implementation.

Table 2 features the best results in each category for each algorithm: the optimization space comprises, where applicable, N_c values ranging from 1 to 32 for the cut-off parameter, K values ranging from 1 to 8 for the number of vanishing moments, and M values ranging from 1 to 8 for the depth of the Wavelet analysis.

Comparing the results shown in Table 2, the best results for what concerns MSE are achieved by the combined Wavelet/cepstrum method as it allows for the same MSE but with a distance three times higher than the plain Wavelet based method. As for the distance of the impulsive response zeros from the unitary circle, the best results are again achieved by the same method: careful attention must

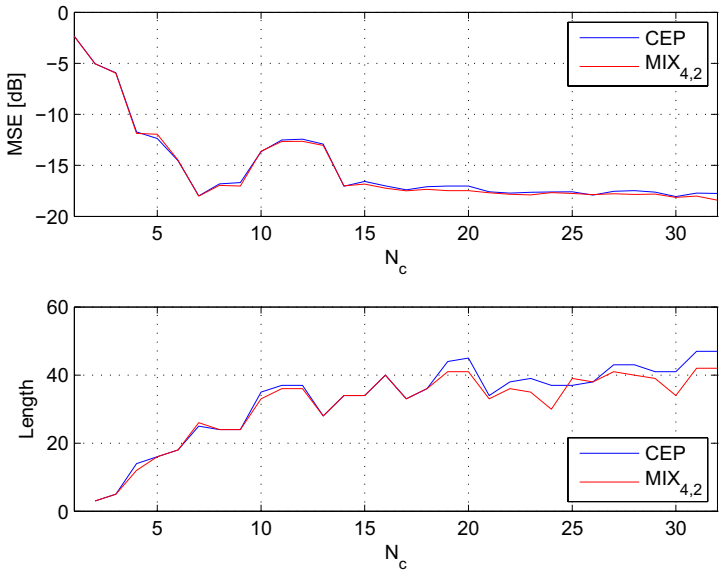


Figure 20: MSE (top) and impulse response length (bottom) as a function of the cut-off parameter in the cepstrum (CEP) and in the best combined Wavelet/cepstrum ($MIX_{4,2}$) estimations

be paid using these estimates as the corresponding MSE is not low enough when compared to the previous results. Finally, almost the same considerations can be done for the minimum filter length: using a filter so short may lead to strongly biased estimations of the un-blurred input signal.

As a conclusion, it should be remarked that shorter impulse responses can be obtained trading-off MSE for impulse response length, as shown in Figure 20. In fact, by choosing $N_c = 7$ for both the estimation techniques, corresponding to the leftmost local minimum of the curves, one gets MSE$= -18$ dB for each of them while the corresponding impulse response lengths are 25 and 26 coefficients respectively.

5 DECONVOLUTION TECHNIQUES

Deconvolution techniques are used in signal and image processing to remove the distortion caused by the transmission and reception systems while reducing the background noise. Synthetic Aperture Radar (SAR) image quality typically suffers from reduced resolution due to both the effect of limited effective aperture size and noise. Iterative and Wavelet-based deconvolution algorithms can be applied to SAR images to enhance contrast and visual quality, easing the post-processing phase.

Ultrasound images obtained from echographic units can be thought as a particular kind of SAR image. In this field, as the ultrasonic acquisition system introduces an unwanted spectral shaping of the backscattered echo signal, deconvolution is used to eliminate this effect and to obtain the pure tissue response. As discussed in Chapter 1, a convenient model to represent RF echo signal $y(n)$ at the A/D converter output involves the convolution between the tissue reflectivity function $x(n)$ and the acquisition system impulse response $h(n)$:

$$y(n) = h(n) * x(n) + \gamma(n) \tag{5.1}$$

where $\gamma(n)$ is a zero-mean Additive White Gaussian Noise (AWGN) term with variance σ_γ^2 [11] and $*$ denotes the convolution operation. Given $y(n)$ and $h(n)$, deconvolution algorithms seek to estimate $x(n)$.

A naïve deconvolution estimate $\hat{x}(n)$ can be obtained by simply convolving $y(n)$ with an approximation of the inverse acquisition system impulse response $\hat{w}(n)$ so that

$$\begin{aligned} h(n) * \hat{w}(n) &= \delta(n - n_0) \\ \hat{x}(n) &= x(n - n_0) + \gamma(n) * \hat{w}(n) \end{aligned} \tag{5.2}$$

where $\delta(n)$ is the Kronecker delta function and n_0 is an appropriate time-delay useful to relax the minimum-phase requirement on $h(n)$ needed to guarantee a stable inversion process.

Unfortunately, both the variance of the colored noise $\nu(n) * \hat{w}(n)$ in $\hat{x}(n)$ and the MSE between $\hat{x}(n)$ and $x(n)$ are large when the inversion process involving $H(n)$ is ill conditioned, making $\hat{x}(n)$ an unsatisfactory deconvolution estimate. In general, deconvolution algorithms can be interpreted as estimators of $x(n)$ from the noisy signal $\hat{x}(n)$ in (5.2).

Since deconvolution is a recurring theme in a wide variety of signal and image processing applications, many algorithms have been proposed to address this problem [5, 54, 55]. Two of the presented deconvolution algorithms exploit the fact that the tissue response $x(n)$ can be *economically* represented in the Wavelet domain, which means that fewer transform-domain coefficients are needed to capture signal features. From another standpoint, due to the complex structure of the echo scatterers in the tissue [56, 57, 58, 54], the tissue response $x(n)$ can be modeled as a $1/f$ type process; the generalized power spectrum $S_x(\omega)$ of such a process obeys to the following power law [59, 60]

$$S_x(\omega) = \sigma_x^2 |\omega|^{\beta - 1} \tag{5.3}$$

where σ_x^2 is the variance of the signal $x(n)$ and β is a scaling parameter in the range $-1 \leqslant \beta \leqslant 1$. Using this model, the variance of the Wavelet signal coefficients $x_j(n)$ can be shown to be

$$\mathrm{var}\{x_j(n)\} = \sigma_w^2 2^{-j\beta} \tag{5.4}$$

where j is the scale index and σ_w^2 a constant related to both the variance of the signal σ_x^2 and the Wavelet function used in the Wavelet Decomposition process [56]. A linear relationship between the Wavelet scale and the logarithm of the variance of input signal Wavelet coefficients exists, and can be exploited to improve the estimate of the tissue response $x(n)$.

5.1 FOURIER-WAVELET REGULARIZED DECONVOLUTION

The FWRD algorithm [5] consider the RF echo acquisition system as a Linear Time Invariant (LTI) system $\mathcal{H}$ whose impulse response, accordingly to (5.1), is $h(n)$. As shown in Figure 21, it uses scalar shrinkage in both Fourier and Wavelet domains to obtain a good estimation of $x(n)$ from $\hat{x}(n)$. This double-domain technique is used in order to find an effective representation of both the input signal $x(n)$ and the contaminating noise $\gamma(n)$, as $x(n)$ is better represented in the Wavelet domain while $\gamma(n)$ in the Fourier domain.

Given an orthonormal basis $\{b_k\}_{k=0}^{N-1}$ for $\mathbb{R}^N$, be it the Fourier or the Wavelet one, $\hat{x}$ from (5.2) can be expressed as

$$\hat{x} = \sum_{k=0}^{N-1} \big(\langle x, b_k\rangle + \langle \hat{w} * \gamma, b_k\rangle\big) b_k \tag{5.5}$$

where $\hat{w}$ is again the inverse acquisition system impulse response. An improved estimate $\hat{x}_\lambda$ can be obtained by simply shrinking the k-th component in (5.5) with an appropriate scalar λ_k with $0 \leqslant \lambda_k \leqslant 1$:

$$\hat{x}_\lambda = \sum_{k=0}^{N-1} \big(\langle x, b_k\rangle + \langle \hat{w} * \gamma, b_k\rangle\big) \lambda_k b_k \tag{5.6}$$

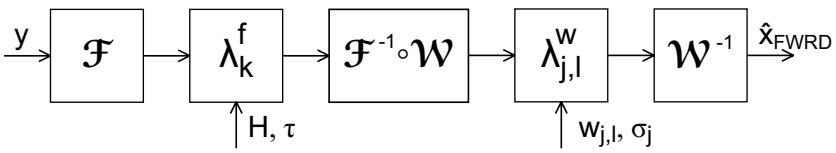


Figure 21: Block diagram of the FWRD algorithm: Fourier domain shrinking is performed in the first row, Wavelet shrinking in the second one.

The parameter λ_k should be taken close to 1 whenever noise energy is negligible when compared to signal energy, and close to 0 in the reciprocal case. The shrinkage by λ_k can also be interpreted as a form of regularization for the deconvolution inverse problem [61].

In the Fourier domain, model (5.1) can be written as

$$Y(f_k) = H(f_k)X(f_k) + \Gamma(f_k) \tag{5.7}$$

where Y, H, X and Γ are respectively the N-length Discrete Fourier Transform (DFT) of y, h, x and ν, while $f_k = \pi k/N$, $k = 1, \ldots, N$ are the normalized DFT frequencies. Rewriting (5.2) in the Fourier domain the following is obtained

$$\hat{X}(f_k) = \frac{Y(f_k)}{H(f_k)} = \begin{cases} X(f_k) + \frac{\Gamma(f_k)}{H(f_k)} & |H(f_k)| > 0 \\ 0 & |H(f_k)| = 0 \end{cases} \tag{5.8}$$

and taking the shrinking parameter λ_k^f as

$$\lambda_k^f = \frac{|H(f_k)|^2}{|H(f_k)|^2 + \tau} \tag{5.9}$$

the DFT $\hat{X}_{\lambda^f}(f_k)$ of the Fourier shrunk estimate can be written as

$$\hat{X}_{\lambda^f}(f_k) = \frac{Y(f_k)H^*(f_k)}{|H(f_k)|^2 + \tau} \tag{5.10}$$

where $\tau > 0$ is a regularization parameter which can be optimized in order to minimize the MSE between the tissue reflectivity function $x(n)$ and its final estimate $\hat{x}(n)$ after both Fourier and Wavelet shrinking have been applied [62].

As in real cases $x(n)$ is not available, τ is usually set to minimize the weighted MSE between $y(n)$ and its estimate $\hat{y}(n)$ generated from $\hat{x}(n)$ [63]:

$$\sum_{k=-(N/2)+1}^{N/2} \frac{|H(f_k)|^2}{|H(f_k)|^2 + \eta} \frac{|H(f_k)\hat{X}(f_k) - Y(f_k)|^2}{|H(f_k)|} \tag{5.11}$$

where

$$\eta = \frac{N\sigma_\gamma^2 \|h\|_2^2}{\|y - \bar{y}\|_2^2} \tag{5.12}$$

with $\bar{y}$ being the estimated mean value of y.

After Fourier shrinking, Wavelet shrinking is performed; this process involves the DWT of the signal $\hat{x}_{\lambda^f}(n)$ obtained from the inverse DFT of $\hat{X}_{\lambda^f}(f_k)$.

Calling ϕ the low-pass scaling function and ψ the prototype bandpass Wavelet function [64], for proper choices of these functions, $\phi_{j,l}(t) = 2^{-j/2}\phi(2^{-j}t - l)$ and $\psi_{j,l}(t) = 2^{-j/2}\psi(2^{-j}t - l)$ form an orthonormal basis for $\mathbb{L}^2(\mathbb{R})$ with $j \in \mathbb{N}$ and $l \in \mathbb{Z}$ being respectively the scale parameter and the translation parameter. Accordingly, Wavelet coefficients were defined as $w_{j,l} = \langle x, \psi_{j,l} \rangle$ while scaling coefficient as $s_{j,l} = \langle x, \phi_{j,l} \rangle$ so that a finite resolution approximation x^J of x is given by

$$x^J(t) = \sum_{l=0}^{N_{J-1}-1} s_{J-1,l}\phi_{J-1,l}(t) + \sum_{j=0}^{J-1}\sum_{l=0}^{N_j-1} w_{j,l}\psi_{j,l}(t)$$

where the parameter J controls the resolution of the Wavelet reconstruction x^J of x.

Wavelet coefficients are shrank using $\lambda^w_{j,l}$:

$$\lambda^w_{j,l} = \frac{|w_{j,l}|^2}{|w_{j,l}|^2 + \sigma_j^2} \tag{5.13}$$

while scaling coefficients are shrank using $\lambda^s_{j,l}$:

$$\lambda^s_{j,l} = \frac{|s_{j,l}|^2}{|s_{j,l}|^2 + \xi_j^2} \tag{5.14}$$

where σ_j^2 and ξ_j^2 are the noise variances at the j-th scale, and $w_{j,l}$ and $s_{J-1,l}$ are respectively the Wavelet and scaling coefficients of $x(n)$.

As $x(n)$ is not known, $w_{j,l}$ and $s_{J-1,l}$ must be estimated too. A feasible solution is performing an hard thresholding upon the Wavelet and scaling coefficients of $\hat{x}_{\lambda^f}(n)$ with threshold $\sigma_j\sqrt{2\log N}$ [65].

Reliable estimations of σ_j^2 and xi_j^2 can be obtained respectively from the Wavelet and scaling coefficients of $y(n)$ as

$$\sigma_j^2 = \sum_{k=-(N/2)+1}^{N/2} \frac{|\Psi_{j,0}(f_k)|^2}{|H(f_k)|^2}\sigma_\gamma^2 \tag{5.15}$$

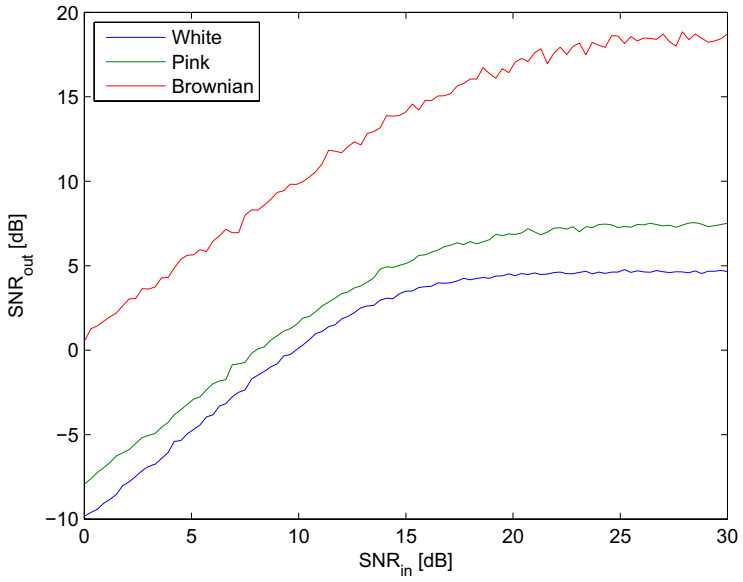


Figure 22: FWRD deconvolution comparison on synthetic signals based on different noise models: the input/output SNRs curves all display an upper saturation effect.

and

$$\xi_j^2 = \sum_{k=-(N/2)+1}^{N/2} \frac{|\Phi_{j,0}(f_k)|^2}{|H(f_k)|^2} \sigma_\gamma^2 \tag{5.16}$$

where $\Psi_{j,0}(f_k)$ and $\Phi_{j,0}(f_k)$ are the Fourier transform of $\psi_{j,0}(n)$ and $\phi_{j,0}(n)$ respectively.

Finally, if not otherwise prescribed, the noise standard deviation σ_γ can be estimated too using the first scale Wavelet coefficients of y [65]:

$$\sigma_\gamma = \frac{\text{MAD}\{w_{0,l}\}}{0.6745} \tag{5.17}$$

The final estimate $\hat{x}(n)$ of $x(n)$ is obtained by applying the Inverse DWT to the shrank Wavelet and scaling coefficients.

Figure 22 features a comparison of the deconvolution performance of the FWRD algorithm when applied to three kinds of synthetic echo signals convolved with a real world acquisition system impulse response and corrupted by different levels of AWGN. Knowing exactly both the echo signal

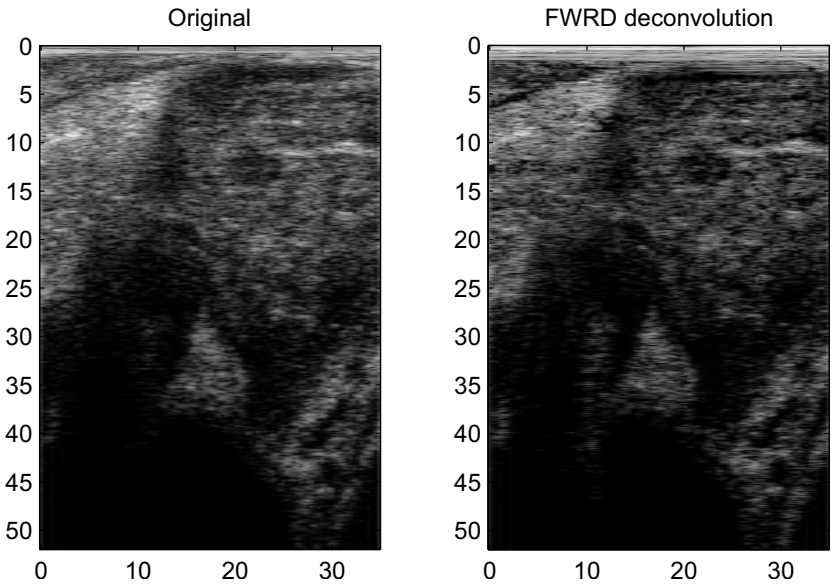


Figure 23: Example of FWRD deconvolution method applied to echographic signals: improvements in both resolution and contrast are visible.

and the impulse response, the mean SNR on 100 deconvolution trials was measured sweeping the input SNR from 0 to 30 dB with a step of 0.3 dB.

As already discussed, the choice of modeling the echo signal as a white, pink or brownian noise is used in literature due to the complex structure of the scatterers in the tissue. Recalling equation (5.3), it is possible to switch between the different noise models by tuning the scaling parameter β: in case of $\beta = 1$, x is purely white noise, while $\beta = -1$ corresponds to Brownian noise.

Figure 22 shows that FWRD is best at deconvolving Brownian noise, whereas white and pink noise have almost the same behavior. In all cases, due to the regularization step, an upper performance saturation effect is present, limiting the maximum output SNR to 18 dB.

Figure 23 shows an example of *in-vivo* echographic image deconvolution: the overall contrast improvement is evident. Moreover, the boundaries of the different tissues got sharper, corresponding to an image resolution improvement.

5.2 WAVELET LEAST SQUARES DECONVOLUTION

The Wavelet Least Squares Deconvolution (WLSD) algorithm uses equation (5.4) to drive a gradient based optimization technique, which minimize an error energy comprising two terms. The first one measures the non-linearity of the logarithmic variance progression at each scale of the WT. The second one evaluates how similar the convolution between the estimated inverse filter and the acquisition system impulse response is to an impulsive function.

Taking steps from the Wavelet series $X_j(k)$ of the discrete-time signal $x(n)$, the Wavelet coefficients at each scale j can be written in the matrix and vector notation as

$$\mathbf{X}_j = \mathbf{H}_j\mathbf{x}, \qquad j = 1, \ldots, J \tag{5.18}$$

where $\mathbf{H}_j$ is a convolution type matrix whose elements corresponds to the equivalent filter coefficients for the j-th scale, $\mathbf{x}$ is the signal vector and J is the total number of scales.

Calling $\mathbf{Y}$ the convolution matrix which corresponds to the RF echo signal and $\hat{\mathbf{w}}$ the approximation of the inverse filter vector, in absence of noise equation (5.2) can be rewritten in matrix form as

$$\hat{\mathbf{x}} = \mathbf{Y}\hat{\mathbf{w}} \tag{5.19}$$

where $\hat{\mathbf{x}}$ is the estimated tissue response vector. Using (5.18) and (5.19), the Wavelet coefficients $\hat{\mathbf{X}}_j$ of the estimated tissue response can be obtained as

$$\hat{\mathbf{X}}_j = \mathbf{H}_j\mathbf{Y}\hat{\mathbf{w}} = \mathbf{D}_j\hat{\mathbf{w}}, \qquad j = 1, \ldots, J \tag{5.20}$$

This relation can be used to estimate the variance V_j of the Wavelet coefficients at the j-th scale

$$V_j = \frac{\hat{\mathbf{w}}^T\mathbf{D}_j^T\mathbf{D}_j\hat{\mathbf{w}}}{N_j - 1} = \hat{\mathbf{w}}^T\mathbf{C}_j\hat{\mathbf{w}} \tag{5.21}$$

where N_j is the number of Wavelet coefficients at the j-th scale. The first part of the energy term is thus defined taking the logarithm of (5.4) and (5.21)

$$e_x = \sum_{j=1}^{J} \Big(\log_2(V_j) - \log_2(\mathrm{var}\{x_j(n)\}) \Big)^2 \tag{5.22}$$

which can be rewritten as

$$e_x = \sum_{j=1}^{J} \left(\log_2(\hat{\mathbf{w}}^T \mathbf{C}_j \hat{\mathbf{w}}) - (\hat{\alpha} - j\hat{\beta}) \right)^2 \tag{5.23}$$

where $\hat{\beta}$ is the estimate of the scaling parameter and $\hat{\alpha}$ is the estimate of the energy parameter $\alpha = \log_2(\sigma_w^2)$.

The second part of the energy term is defined using the matrix form of (5.2)

$$e_w = (\mathbf{G}\hat{\mathbf{w}} - \boldsymbol{\delta})^T (\mathbf{G}\hat{\mathbf{w}} - \boldsymbol{\delta}) \tag{5.24}$$

where $\mathbf{G}$ is the convolution matrix for the transducer impulse response and $\boldsymbol{\delta} = [0 \dots 0\, 1\, 0 \dots 0]^T$. The total energy is defined as the sum of these partial energy terms

$$\begin{aligned} e(\hat{\mathbf{w}}, \hat{\alpha}, \hat{\beta}) = \textstyle\sum_{j=1}^{J} \left(\log_2(\hat{\mathbf{w}}^T \mathbf{C}_j \hat{\mathbf{w}}) - (\hat{\alpha} - j\hat{\beta}) \right)^2 + \\ + (\mathbf{G}\hat{\mathbf{w}} - \boldsymbol{\delta})^T (\mathbf{G}\hat{\mathbf{w}} - \boldsymbol{\delta}) \end{aligned} \tag{5.25}$$

and is used to drive an iterative minimization technique based on conjugate gradient. In order to speed-up algorithm convergence, the gradient and the Hessian matrix of the error energy $e(\hat{\mathbf{w}}, \hat{\alpha}, \hat{\beta})$ should be obtained algebraically; additionally, ad-hoc preconditioners can be used to further reduce the estimation time. Good guess for the first iteration are $\hat{\mathbf{w}} = \mathbf{G}^{\dagger}\boldsymbol{\delta}$, $\hat{\alpha} = 0$ and $\hat{\beta} = 0$.

Figure 24 features a comparison of the deconvolution performance of the WLSD algorithm applied to the same synthetic echo signals used for the FWRD algorithm. As the FWRD algorithm, WLSD performs best on Brownian noise, whereas white and pink noise are quite similar. Performance saturation happens at a higher input SNR, making the WLSD best suitable to process signals with an noise level at most 15 dB lower than signal level.

Figure 25 shows an example of *in-vivo* echographic image deconvolution: again, the overall contrast improvement is evident; however when compared to FWRD, WLSD result displays less sharper tissue boundaries and an higher background noise.

5.2.1 *Adaptive delay WLSD*

As already stated, equations (5.2) allow for input/output delays different from zero: this additional degree of freedom

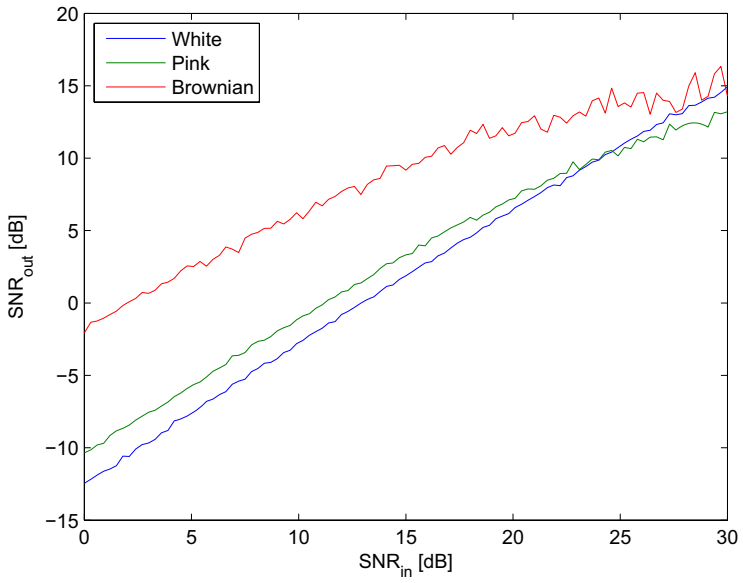


Figure 24: FWRD deconvolution comparison on synthetic signals using the same parameters of Figure 22.

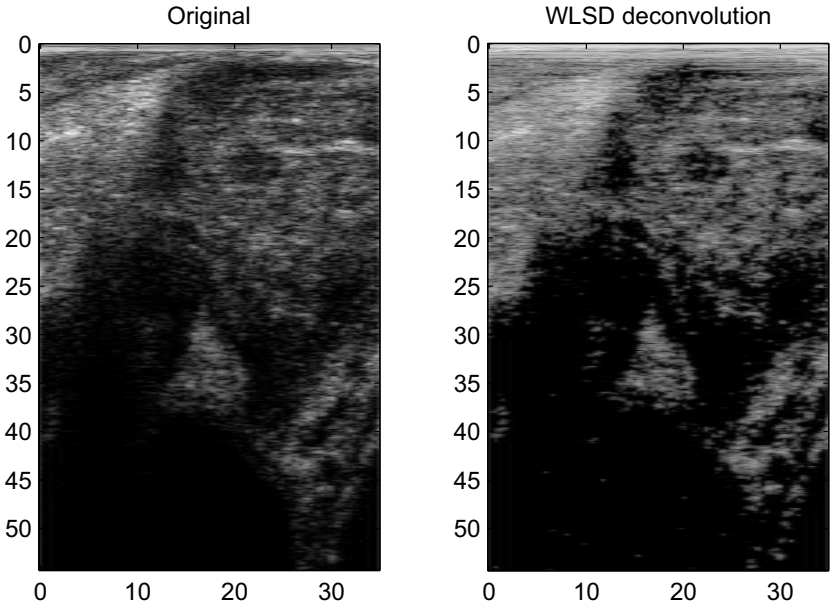


Figure 25: WLSD deconvolution applied to the same echographic image of Figure 23.

can be used deal with non-minimum phase acquisition system impulse responses. From the point of view of the WLSD algorithm, exploiting this additional flexibility could lead to a further minimization of the error energy, hence to improve the deconvolution performances.

Recalling equation (5.21) and plugging into it the initial guess that was used in previous section to initialize the WLSD algorithm, the next relationships are obtain

$$V_j = \boldsymbol{\delta}^T (\mathbf{G}^\dagger)^T \mathbf{C}_j \mathbf{G}^\dagger \boldsymbol{\delta} = \boldsymbol{\delta}^T \mathbf{M}_j \boldsymbol{\delta} \tag{5.26}$$

Equation (5.26) simply states that, when the 1 in δ is moved to the i-th element, the corresponding variance is given by i-th element of the diagonal of $\mathbf{M}_j$. In practice V_j can be substituted with

$$V_j(n) = [\text{diag}\{\mathbf{M}_j\}]_n \tag{5.27}$$

where diag is the operator that given an $N \times N$ square matrix produces the N element vector corresponding to the main diagonal and n is the position of the 1 in δ.

Since it was chose to set $\hat{\mathbf{w}} = \mathbf{G}^\dagger \delta$, the energy term (5.24) is usually negligible when compared to (5.22). In order to find the best values of $\hat{\alpha}$, $\hat{\beta}$ and n, one can minimize the error term

$$E(n, \hat{\alpha}, \hat{\beta}) = \sum_{j=1}^{J} \left(\log_2(V_j(n)) - (\hat{\alpha} - j\hat{\beta}) \right)^2 \tag{5.28}$$

A simple procedure to achieve this result is to optimize separately the discrete parameters (i.e. n) from the continuous ones ($\hat{\alpha}$ and $\hat{\beta}$). Starting with $\hat{\alpha} = 0$ and $\hat{\beta} = 0$, one finds the value n_0 of n which minimizes (5.28).

Looking for the best values of $\hat{\alpha}$ and $\hat{\beta}$, (5.28) is expanded choosing $n = n_0$

$$\begin{cases} \frac{\partial E}{\partial \hat{\alpha}} = -2 \sum_{j=1}^{J} (\log_2(V_j(n_0)) - (\hat{\alpha} - j\hat{\beta})) = 0 \\ \frac{\partial E}{\partial \hat{\beta}} = 2 \sum_{j=1}^{J} j(\log_2(V_j(n_0)) - (\hat{\alpha} - j\hat{\beta})) = 0 \end{cases} \tag{5.29}$$

which can be rewritten as

$$\begin{cases} \sigma_0 = J\hat{\alpha} - \frac{J(J+1)}{2}\hat{\beta} \\ \sigma_1 = \frac{J(J+1)}{2}\hat{\alpha} - \frac{J(J+1)(2J+1)}{6}\hat{\beta} \end{cases} \tag{5.30}$$

where $\sigma_0 = \sum_{j=1}^{J} \log_2(V_j(n_0))$ and $\sigma_1 = \sum_{j=1}^{J} j \log_2(V_j(n_0))$. System (5.30) always has a unique solution if $J \neq 1$, i.e. if the Wavelet analysis has more than one octave:

$$\begin{cases} \hat{\alpha}_0 = \frac{2\sigma_0(2J+1)+6\sigma_1}{J(J-1)} \\ \hat{\beta}_0 = \frac{6\sigma_0(J+1)-12\sigma_1}{J(J^2-1)} \end{cases} \tag{5.31}$$

One now has the choice to plug $\hat{\alpha}_0$ and $\hat{\beta}_0$ into (5.28) and obtain an updated value n_1 for the delay parameter, which in times can be used in (5.30) to estimate $\hat{\alpha}_1$ and $\hat{\beta}_1$. No more than three iteration are usually required for this algorithm to achieve convergence. The final estimates of $\hat{\mathbf{w}}$ and $\hat{\mathbf{x}}$ are obtained in the following way:

$$\begin{aligned} \hat{\mathbf{w}} &= \mathbf{G}^{\dagger}\boldsymbol{\delta}(n_{opt}) \\ \hat{\mathbf{x}} &= \mathbf{Y}\hat{\mathbf{w}} \end{aligned} \tag{5.32}$$

As a consequence of the property to autonomously adapt its input/output delay to best suit the acquisition system impulse response, this algorithm is called Adaptive delay Wavelet Least Squares Deconvolution (AdWLSD).

Figure 26 features a comparison of the deconvolution performance of the AdWLSD algorithm to the same synthetic echo signals used for the FWRD algorithm. Again, the best performance are recored for Brownian noise, whereas white and pink noise behaviors differ for no more than 2 dB. However, as no regularization step is necessary to perform AdWLSD, this algorithm does not feature the performance saturation effect characteristic of FWRD.

Figure 27 shows an example of *in-vivo* echographic image deconvolution: contrast improvement is on the same levels of FWRD while tissue boundaries sharpening is comparable to WLSD. Finally, background noise is slightly higher than FWRD, but less than WLSD.

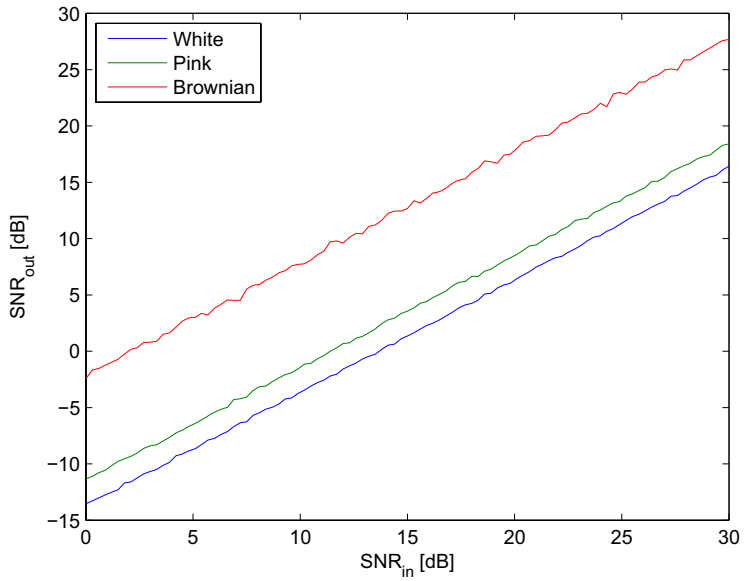


Figure 26: AdWLSD deconvolution comparison on syntetic signals using the same parameters of Figure 22.

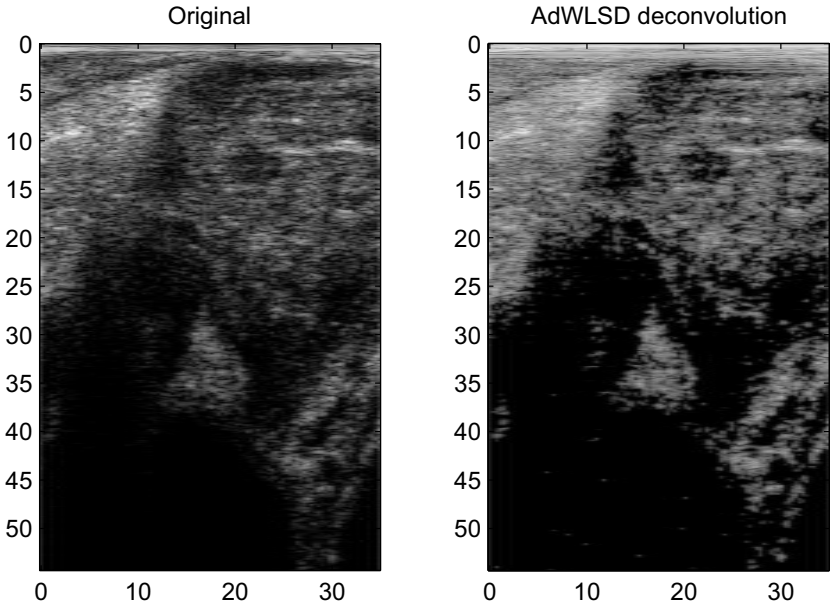


Figure 27: AdWLSD deconvolution applied to the same echographic image of Figure 23.

5.3 ADJUSTABLE DELAY ITERATIVE DECONVOLUTION

Many commonly used deconvolution algorithms, FWRD and WLSD among them, require the acquisition system impulse response to be minimum-phase to ensue reconstruction stability. At the same time, impulse response estimation algorithms usually discard phase information and perform their task in the cepstrum domain assuming the response to be minimum-phase. Combining these algorithms it is possible to reconstruct an incoming blurred signal, although with a certain degree of approximation.

This is due to the fact that each time an echo is generated by an interface which is not aligned to the sampling grid implicitly defined by the time domain sampling, that echo has a different phase content with respect to the ones generated by interfaces aligned to the grid.

By considering the simplest possible tissue response in the continuous time domain – a Dirac delta at time θ_0 – and calling $h(t)$ the unsampled acquisition system impulse response and T_0 the sampling period, the following is obtained

$$y(nT_0) = \int_{-\infty}^{+\infty} \delta(\tau - \theta_0) h(nT_0 - \tau) d\tau = h(nT_0 - \theta_0) \quad (5.33)$$

Moving to the Z-domain and calling m the integer part of the ratio θ_0/T_0, the following equality holds:

$$Y[z] = z^{-m} z^{-(\frac{\theta_0}{T_0} - m)} H[z] \quad (5.34)$$

Working in a discrete environment, the non-integer delay term is usually neglected, giving way to approximate estimation of the received echo and increasing the output noise level. In order to improve deconvolution performance, it is possible to jointly estimate what is called the *equivalent impulse response* $H_{eq}[z]$

$$H_{eq}[z] = z^{-(\frac{\theta_0}{T_0} - m)} H[z] \quad (5.35)$$

The founding idea which the ADID algorithm is built on is the possibility to decompose any causal filter with no poles or zeros on the unit circle in a cascade of a minimum-phase and an all-pass filter [66]. The non-integer delay effect

is modeled using the all-pass filter, while the amplitude information descends from the minimum-phase estimate of $H[z]$.

Desiring to accurately perform the filtering stage, the properties of the following parametric all-pass filter are analytically evaluated, obtaining:

$$P(z) = \frac{-a + z^{-1}}{1 - az^{-1}} \tag{5.36}$$

which, evaluated on the unit circle, becomes:

$$P(e^{j\omega}) = \frac{[(a^2+1)\cos(\omega) - 2a] + \jmath[(a^2-1)\sin(\omega)]}{[(a^2+1) - 2a\cos(\omega)]^2} \tag{5.37}$$

To ensue filter stability the tuning parameter a must satisfy $|a| = 1$.

While the modulus of $P(e^{j\omega})$ is always 1 independently of both a and ω, the analytical phase $\phi_P(\omega)$ of P, given by argument of $P(e^{j\omega})$, depends on both these parameters:

$$\phi_P(\omega, a) = \tan^{-1}\left(\frac{(a^2-1)\sin(\omega)}{(a^2+1)\cos(\omega) - 2a}\right) \tag{5.38}$$

So, the group delay $\tau = \tau_g(\omega, a)$, defined as the inverse of the derivative of the phase [66], becomes

$$\tau_g(\omega, a) = \frac{1 - a^2}{1 - 2a\cos(\omega) + a^2} \tag{5.39}$$

For a given value of the filter parameter a, it is possible to evaluate the effective group delay of the whole echographic pulse exploiting the following weighted sum:

$$\tau_{g,eff}(a) = \frac{\int_0^\pi \tau_g(\omega, a)|H(e^{\jmath\omega})|d\omega}{\int_0^\pi |H(e^{\jmath\omega}|d\omega} \tag{5.40}$$

In order to approximate equation (5.40) it is possible to estimate the group delay around the pulsation corresponding to the highest energy spectral density, getting

$$\tau_{g,eff}(a) \simeq \tau_g(\omega_0, a)|\omega_0 = \frac{\int_0^\pi \omega|H(e^{\jmath\omega})|d\omega}{\int_0^\pi |H(e^{\jmath\omega}|d\omega} \tag{5.41}$$

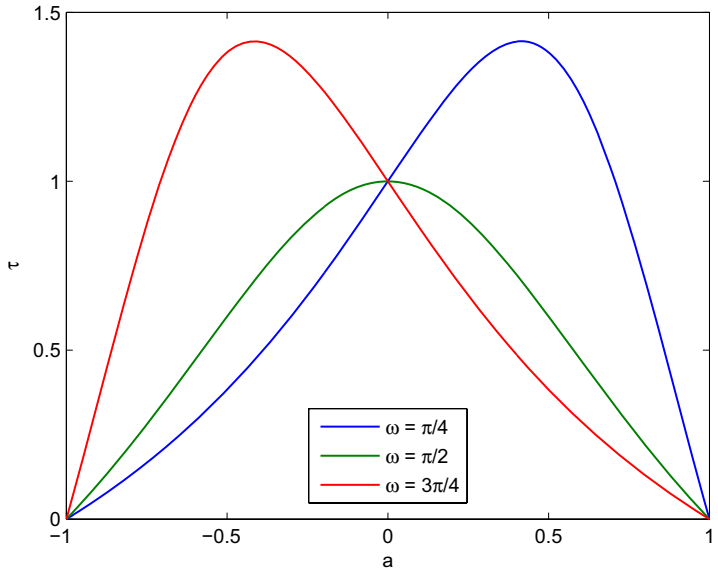


Figure 28: Group delay as a function of the all-pass filter parameter a for different values of the pulsation ω.

As shown in Figure 28, the group delay is a smooth function of the filter parameter a in the definition domain; function shape depends on the pulsation ω and is symmetrical around $\omega = \pi/2$. Solving a from (5.39) as function of both ω and τ, two values of a for each given ω and τ pair are obtained

$$a_{1,2}(\omega, \tau) = \frac{\tau \cos(\omega) \pm \sqrt{1 - \tau^2 \sin^2(\omega)}}{\tau + 1} \tag{5.42}$$

where a_1 corresponds to the minus sign and a_2 to the plus one.

Recalling than m is the integer part of the ratio θ_0/T_0, the difference $q = \theta_0/T_0 - m$ is always in the range $[0, 1]$. As the all-pass filter $P(z)$ is meant for mimicking this delay, τ is to be kept within the same range. These values always allow to find acceptable values for the filter parameter a independently from ω: in fact, in order to ensue $a \in \mathbb{R}$, τ must not be greater than $1/\sin(\omega)$ with $\omega \in [0, \pi]$. Thus the highest acceptable value for τ is 1, corresponding to $\omega = \pi/2$. This is hinted by Figure 28 where, for $\omega = \pi/2$, the highest group delay is 1, corresponding to $a = 0$.

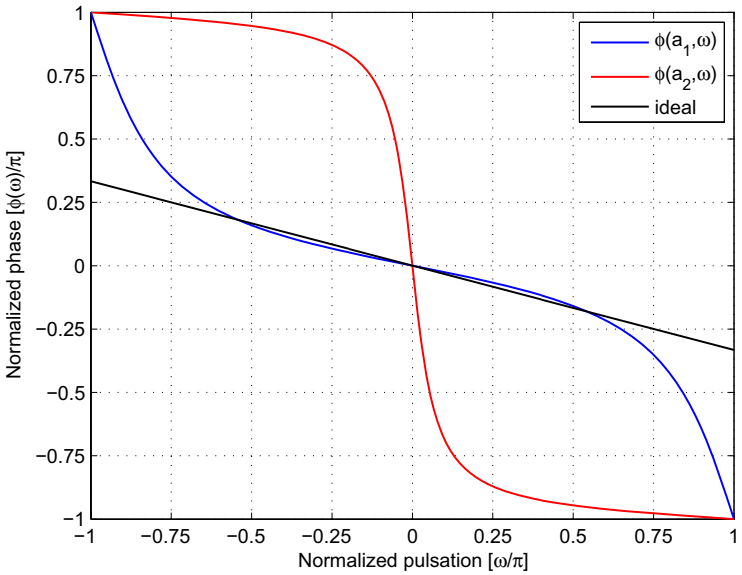


Figure 29: Phase behavior of the all-pass filter compared to the ideal one: a_1 allows to best reconstruct the ideal trend.

For different values of τ and ω, equation (5.42) always has two solutions. The corresponding values of a reflect into different behaviors of $P(z)$: it is particularly interesting to estimate how similar the unwrapped phase of the all-pass filter is to the phase of the non-integer delay term z^{-q}. Exploiting the definition of phase, for this term the next equality holds

$$\phi_t(\omega, q) = \arg(e^{-j\omega q}) = -\omega q \tag{5.43}$$

Using the same value for q and τ, Figure 29 shows that for $\omega_0 = \pi/3$ and $\tau = 1/3$, a_1 clearly allows to best reproduce the ideal behavior.

In order to evaluate which of the two possible values for the filter parameter is best for mimicking the non-integer delay term effect over all the possible values of ω_0 and τ, the MSE between $\phi_t(\omega, \tau)$ and $\phi_P(\omega, a)$ was tabulated, choosing $a = a_{1,2}(\omega_0, \tau)$. Figure 30 displays a graphical representations of this comparison: a_1 results best over all the acceptable values of ω_0 and τ. The only case for which a_1 and a_2 feature the same performance is when $\tau = 1$ and $\omega_0 = \pi/2$. Otherwise a_1 always corresponds to an MSE lower than a_2.

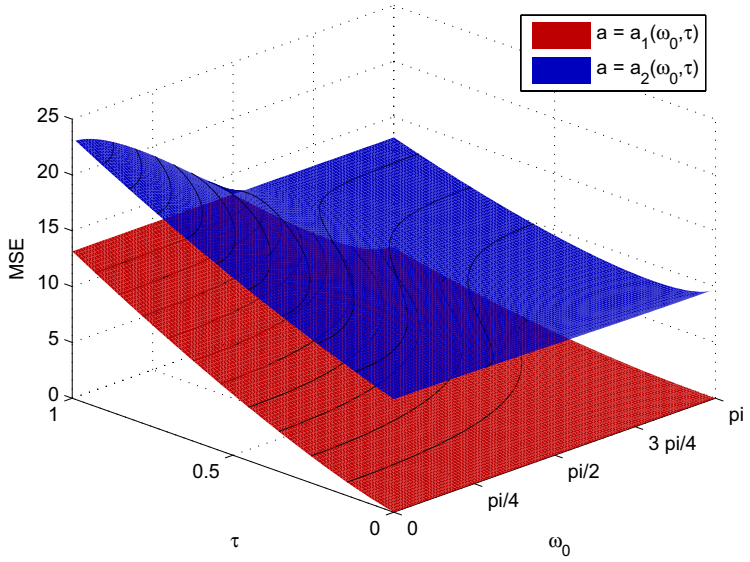


Figure 30: Evaluation of the MSE between the phase of the all-pass filter and the ideal delay term for different values of τ and ω_0

Having shown that an all-pass filter with an appropriate setting of its parameter can adequately emulate the behavior of a non-integer delay term, a modified version of the CLEAN algorithm [67] is used to perform signal deconvolution. Given the dirty signal y and the estimated acquisition system impulse response h, the original algorithm steps can be summarized in this way:

1. Find the position p_k for which h has the best correlation with y.
2. Optimize the amplitude g_k of h so that $g_k \cdot h$ becomes a good fit for y and subtract the final result from y.
3. Repeat steps 1 and 2, each time replacing y with the result of step 2 until its energy becomes lower than a fixed threshold or the maximum number of iterations is reached.
4. Return all the combinations of positions p_k and amplitudes g_k found evaluating steps 1 and 2.

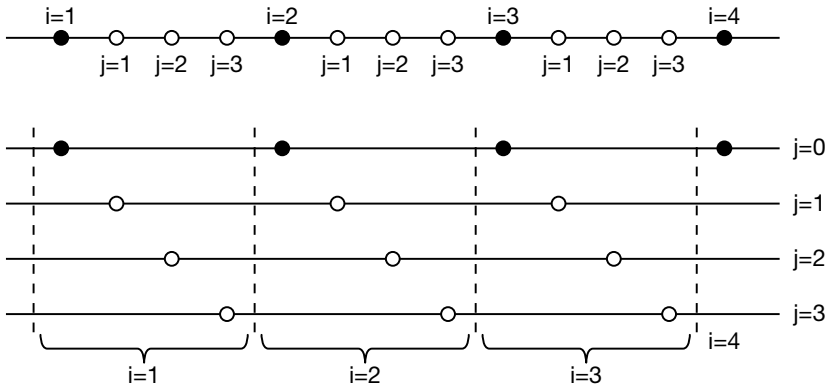


Figure 31: ADID computational grid scheme: original grid points are filled in black, while refined points are filled in white. The grid scheme used in the naïve implementation is depicted in the top row, while the one exploited by the fast method is shown in the bottom row.

To the purposes of the ADID algorithm, the first step in this sequence is substituted with one capable to take into account non-integer delays. The values which can be use for p_k are usually taken from an integer grid: in practice, this grid is refined adding $N-1$ equally spaced points between each point. Within each interval, each of these new points is associated with a value of the parameter τ used to estimate the all-pass filter parameter a: in particular, for the j-th point $\tau = j/N$ is set.

A naïve implementation of the ADID algorithm uses the grid shown in the top row of Figure 31: at each white point of the grid an equivalent acquisition system impulse response h_e is synthesized by convolving h with the corresponding all-pass filter, while at black points the original h is used. Correlation with the original signal is then computed and a vector is generated: the position p_k within this vector, corresponding to the highest correlation, is then used along with the matching pulse.

Although straightforward, this procedure is quite suboptimal and slow, since there is the need to recompute h_e at each point of the grid. An optimization of this policy is shown on the bottom row of Figure 31: instead of evaluating h_e at each point, all the $N-1$ versions of h are first precalculated then used to calculate the correlations between each

of them and y on the original grid. In fact, the white points corresponding to the same j are separated by the very same interval of the black points. Thinking about the black dots as corresponding to $j = 0$, the position p_k is determined by the coordinates of the interval which achieves the best correlation with y.

So, given the dirty signal y and the estimated acquisition system impulse response h, the ADID algorithm steps can be summarized in this way:

1. Set the number N of subdivision of the base interval.
2. Generate the $N - 1$ non-integer shifted versions h_e of h by filtering h with a properly set all-pass filter $P(z)$.
3. Call $\mathcal{H}$ the set of h and h_e
4. Find the position p_k for which there is the best correlation between y and any of the elements of $\mathcal{H}$.
5. Optimize the amplitude g_k of the corresponding element h_k of $\mathcal{H}$ so that $g_k \cdot h_k$ becomes a good fit for y and subtract the final result from y[1].
6. Repeat steps 4 and 5, each time replacing y with the result of step 5 until its energy becomes lower than a fixed threshold or the maximum number of iterations is reached.
7. Return all the combinations of positions p_k and amplitudes g_k found evaluating steps 1 and 2.

Processing the signal incoming from a water-tank experiment, varying the number of subdivisions does not significantly affect the MSE measured between the input signal and the one reconstructed after the deconvolution, as shown in Figure 32. However, measuring the energy fraction, i.e. the ratio between the cumulative energy of the 10

1 Supposing h_k and y to be vector of the same size, an estimate of g_k minimizing the MSE between them is given by

$$g_k = \frac{\langle y | h_k \rangle}{\langle h_k | h_k \rangle}$$

where $\langle \bullet | \bullet \rangle$ denotes the dot product.

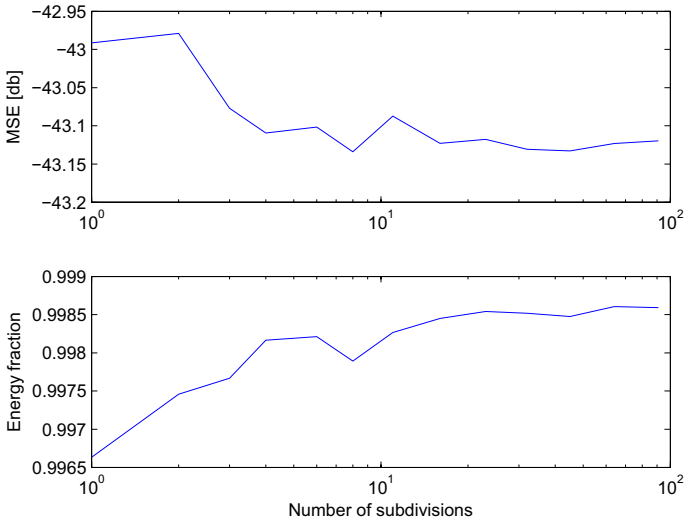


Figure 32: ADID performance evaluation for a constant number of iterations as a function of the number of subdivisions. The MSE on the top row is less affected than the energy fraction, as shown in the bottom row.

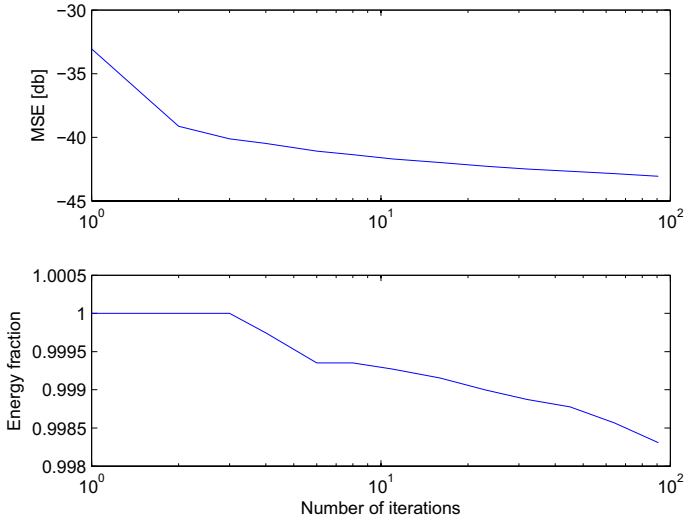


Figure 33: ADID performance evaluation for a constant number of subdivision as a function of the number of iterations. The MSE on the top row is positively affected while the energy fraction shown in the bottom row is slightly reduced.

most prominent deconvolved pulses and the energy of the whole restored signal, a performance improvement is observable. In fact, as the number of subdivisions grows, the energy fraction increases, meaning that a lower number of signal samples are necessary to capture signal information, i.e. the two interfaces are more easily located.

Figure 33 shows the opposite behavior for what concerns the dependencies from the number of iterations. As this number grows, the MSE decreases, meaning than the incoming signal is better and better reconstructed. However, at the same time, the energy fraction slowly decreases as more pulses are added to the deconvolved signal.

5.4 PERFORMANCE COMPARISON

As already stated, image restoration techniques, such as deconvolution, can be employed to improve the resolution of US images and their diagnostic significance. Two approaches are the most common when dealing with US image deconvolution. The first incorporates the Point Spread Function (PSF) estimation procedure within the deconvolution algorithm. In the second approach, PSF and true image estimation are two disjoint procedures.

Following this second processing flow and using the pulses estimated using the methods presented in Chapter 4, the algorithms presented in this chapter were evaluated as de-blurring techniques for biomedical images to verify and compare their effectiveness. They were tested on a US RF-signal database which comprises both synthetic phantom (CIRS Model 047) and in-vivo Trans-Rectal Ultrasound (TRUS) acquisitions of prostatic glands (264 frames), all obtained with a commercial US equipment (MYLAB90 Esaote S.p.a.). After the processing of the signal envelope, de-blurring performances were evaluated in terms of enhancement in both image resolution and quality.

To quantify resolution improvement, axial Resolution Gain at -6 dB (Gax) [68] was measured; conversely, Peak Signal to Noise Ratio (PSNR) and Quality Index (QI) were used to compute the dissimilarity between the original and processed image, in terms of loss of correlation, luminance and contrast distortion [69]; finally image contrast enhance-

Phantoms	Gax	PSNR [dB]	QI	CG
FWRD	0.84 ± 0.17	22.34 ± 2.09	0.79 ± 0.09	1.15 ± 0.07
WLSD	3.83 ± 0.95	22.56 ± 1.92	0.83 ± 0.07	2.56 ± 0.22
AdWLSD	1.37 ± 0.24	21.96 ± 1.76	0.80 ± 0.07	2.35 ± 0.24
ADID	6.67 ± 0.42	21.91 ± 1.68	0.80 ± 0.06	3.33 ± 0.56
In-vivo	**Gax**	**PSNR [dB]**	**QI**	**CG**
FWRD	1.64 ± 0.22	26.54 ± 2.38	0.93 ± 0.04	0.83 ± 0.11
WLSD	3.38 ± 1.13	26.63 ± 1.96	0.93 ± 0.03	3.22 ± 0.28
AdWLSD	3.79 ± 0.97	26.19 ± 1.85	0.93 ± 0.03	3.15 ± 0.26
ADID	8.73 ± 0.54	25.45 ± 2.06	0.91 ± 0.03	4.43 ± 0.86

Table 3: A performance comparison of the discussed deconvolution algorithms applied on echographic images: on top, results related to the phantoms are shown, while results on *in-vivo* images are shown on the bottom.

ment on phantoms was measured by means of Contrast Gain (CG) [70].

All the images from the dataset were processed with the proposed deconvolution algorithms driven by the pulse estimated by the two following procedures: in the first one, the pulse was estimated using the returning echo of a water-tank experiment, while in the second one the estimation was conducted directly on the *in-vivo* image. It was found that image processing based on the second method provides better performance with respect to the considered evaluation metrics. This is due to the system response aberration caused by tissue intrinsic sound speed propagation constant inhomogeneities, which can be accounted only by estimating the pulse form the *in-vivo* frames.

Table 3 reports the mean values and the standard deviations of the results obtained processing both phantom and *in-vivo* images. All the discussed algorithms provide a good resolution increase in the axial direction for both *in-vivo* and phantom acquisitions, with the only exception of FWRD applied to phantoms; better performance were recorded processing the *in-vivo* frames. ADID features the best results on both phantoms and *in-vivo* frames with a very low deviation from the mean value. The standard WLSD scores second

on phantom while its adaptive delay version is second best on *in-vivo* images, however with a deviation higher than ADID.

Peak SNR is almost the same for all the algorithms, with fluctuations less than 2 dB, both on phantoms and *in-vivo* frames. These fluctuations are comparable to the relative deviations from the mean values, thus negligible. The same happens regarding image quality, the only noticeable fact being that *in-vivo* images are better processed comparing to the phantoms.

Finally, contrast gain estimations again award ADID for what concerns the mean values. However, while these results are quite good, the same cannot be told concerning the standard deviation: in this case AdWLSD is the best algorithm on *in-vivo* frames, while FWRD gets the best results on phantoms.

Figure 34 features a visual comparison of the different deconvolution algorithms processing output. At a visual inspection, the lumen at coordinates $[5, -25]$ is best rendered by the FWRD algorithm, while the best overall improvement in image resolution is achieved by ADID.

Although the best image background noise rejection is again obtained by FWRD, several anatomical structures were visible after ADID processing that could not be seen in the original image. WLSD and AdWLSD are both good compromise solutions, featuring the fastest computation time among all the discussed algorithms.

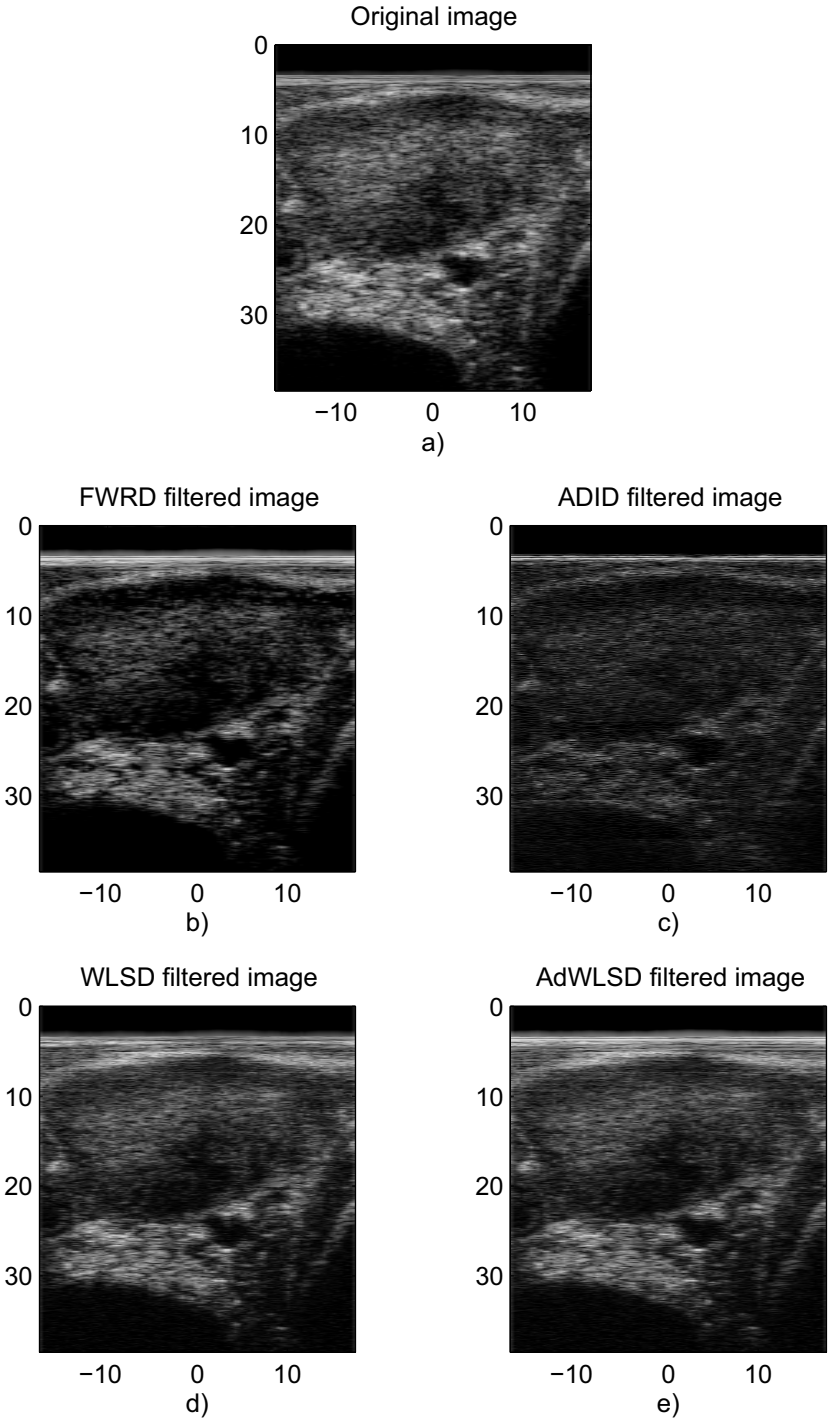


Figure 34: Deconvolution algorithms visual comparison on echographic images; from top to bottom: a) Original image, b) FWRD output, c) ADID output, d) WLSD output, e) AdWLSD output.

6

ADAPTIVE FILTERING

A very important part of statistical signal processing is constituted by the subject of adaptive filtering. The use of algorithms originating from adaptive filter theory is usually advantageous whenever it is necessary to elaborate signals which are non-stationary or are generated by an environment of unknown statistics.

However at the price of an higher computational cost, adaptive filtering often provides new signal-processing capabilities that would not be otherwise possible using fixed filters designed by conventional methods. Because of their intrinsic nature to adapt themselves to the different signal characteristics, adaptive filters have been successfully applied to fields like communications, radar, seismology and biomedical engineering.

In each one of these environments, a signal is generated and transmitted into a medium of some kind then received and used for a generic task: this give way to a model where a cascade of a transmitter, a channel and a receiver is present. The function of the transmitter is to convert the input signal into a waveform suitable for trans-

mission over the channel. Typically the channel suffers from two major kinds of non-idealities: noise and Intersymbol Interference (ISI).

Despite the fact that noise is always present at the output of every signal transmission/reception system, its cause are the most variables. In general, it can be due the system itself or to the environment in which the system operates: an example of the former case is the thermal noise present in every electronic equipment, while in the latter interfering signals may degrade what is meant to be received.

ISI is quite a different mechanism: in practice, for any physical channel it is impossible to have a time domain impulse response $h(t)$ of this kind

$$h(t) = A\delta(t - \tau) \tag{6.1}$$

or equivalently in the frequency domain

$$H(\jmath\omega) = A\exp(-\jmath\omega\tau) \tag{6.2}$$

where A is a scaling factor, δ is the Dirac function and τ is a proper time delay.

At most, it is possible that a physical system approximates (6.2) over a finite interval of frequency. Because of this the channel is called *dispersive* and each transmitted pulse blends with its neighbors according to a pattern called ISI [71].

In order to recover the transmitted signal, the removal of both noise and ISI is mandatory: this task is performed by properly filtering the received signal. Moreover, if the scaling factor varies along time, adaptive filtering techniques are necessary to track these variations and improve the estimate of the true signal.

6.1 LEAST MEAN SQUARE ADAPTIVE FILTER

As already discussed in Chapter 5, an ultrasound imaging acquisition system introduces an unwanted spectral shaping of the backscattered echo signal. Usually this deformation is quite tricky to characterize and slowly varies along time. Similar behaviors are observed in other acquisition systems too.

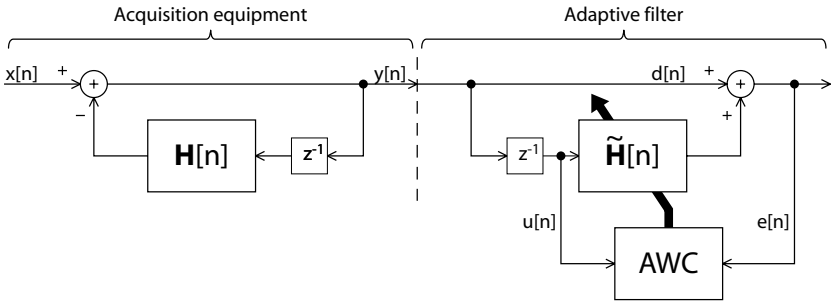


Figure 35: LMS adaptive filter as a mean for deconvolving signals: in this configuration the adaptive filter, once tuned, can remove the spectral deformation due to the acquisition equipment

If a proper order AR model [71] is used to describe the spectral deformation of the input signal, assumed having uncorrelated samples, a Least Mean Square (LMS) adaptive filter can be used to remove this unwanted effect in a very simple though effective way. While LMS algorithms are not the only solution to this filtering problem, they are the standard against which other linear adaptive filters are benchmarked because of their simplicity.

As shown on the right side of Figure 35, an LMS algorithm is built around two basic structures: a filtering process, denoted by $\tilde{\mathbf{H}}[n]$, which generates an estimated signal $e[n]$ combining its output with the desired response $d[n]$ and an Adaptive Weight Control (AWC) process, which adapts the filter weight according to the input signal $u[n]$ and the estimated signal. If $\tilde{\mathbf{H}}[n]$ is an exact copy of the autoregressive coefficient sequence $\mathbf{H}[n]$, the output of the filtering process effectively reconstructs the feedback term of the AR model: as a consequence, $e[n]$ becomes an exact copy of the input signal $x[n]$.

If $\mathbf{H}$ is supposed to be time-invariant, the whole system can be modeled in the time domain as

$$\begin{aligned} y[n] &= x[n] - \textstyle\sum_{i=1}^{N} y[n-i]H_{i-1} \\ e[n] &= d[n] + \textstyle\sum_{i=0}^{N-1} u[n-i]\tilde{H}_i[n] \\ \tilde{\mathbf{H}}[n+1] &= \mathrm{AWC}(\tilde{\mathbf{H}}[n], u[n], e[n]) \end{aligned} \tag{6.3}$$

where $\tilde{H}_i[n]$ is the i-th element of the filter $\tilde{\mathbf{H}}$ at time n.

If the AWC rule is properly designed, after a brief period of time, eventually $\tilde{\mathbf{H}}$ will converge to $\mathbf{H}$ and the AWC

should stop updating $\tilde{\mathbf{H}}$. From this time on, system (6.3) can be rewritten as

$$\begin{aligned} y[n] &= x[n] - \sum_{i=1}^{N} y[n-i]H_{i-1} \\ e[n] &= d[n] + \sum_{i=0}^{N-1} u[n-i]H_i \end{aligned} \tag{6.4}$$

Recognizing that $d[n] = y[n]$ and $u[n] = y[n-1]$, the following relationships are obtained

$$\begin{aligned} y[n] &= x[n] - \sum_{i=1}^{N} y[n-i]H_{i-1} \\ e[n] &= y[n] + \sum_{i=0}^{N-1} y[n-i-1]H_i \end{aligned} \tag{6.5}$$

where both the second members of the left hand side terms of (6.5) can be made identical with simple algebraic substitutions, making $e[n]$ an exact copy of $x[n]$.

The simplest AWC rule minimize $J[n] = e[n]^2/2$ using an iterative steepest-descent algorithm:

$$\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu \frac{\partial J[n]}{\partial \tilde{H}_i[n]} \tag{6.6}$$

where $\mu > 0$ is the so-called *step-size parameter*. It can be shown [71] that, as long as $x[n]$ is assumed uncorrelated, if μ is properly chosen $\tilde{\mathbf{H}}$ will indeed converge to $\mathbf{H}$.

Exploiting (6.3) it is possible to explicit the update term in (6.6) as

$$\frac{\partial J[n]}{\partial \tilde{H}_i[n]} = \frac{\partial J[n]}{\partial e[n]} \cdot \frac{\partial e[n]}{\partial \tilde{H}_i[n]} = e[n]u[n-i] \tag{6.7}$$

so (6.6) finally becomes

$$\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu e[n]u[n-i] \tag{6.8}$$

As a consequence, at each time step the LMS algorithm performs the following operations

1. $f[n] = \sum_{i=1}^{N} u(n-i)H_i[n]$
2. $e[n] = d[n] + f[n]$
3. $\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu e[n]u[n-i]$

Having assumed $\mathbf{H}$ to be time independent, the mean time necessary for $\tilde{\mathbf{H}}$ to converge to $\mathbf{H}$ depends on two factors: the initial value of $\tilde{\mathbf{H}}$ and the step-size parameter

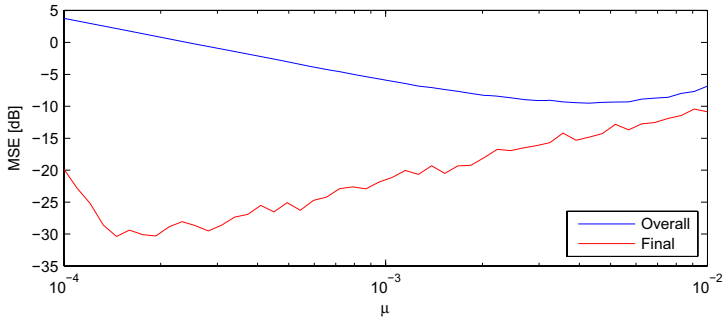


Figure 36: LMS algorithm overall and final MSE dependency on the step-size parameter: overall MSE takes into consideration the initial transient while final MSE describes only the last 100 time steps.

μ. The closer $\tilde{\mathbf{H}}$ is to $\mathbf{H}$, the less time is required for convergence. The larger the step-size parameter μ, the faster the convergence rate.

Large values of μ may however lead to instability due to the implicit feedback loop on $e[n]$: the highest value of μ which guarantee stability is called *critical value* μ_{crit} and is rather difficult to identify. It is nevertheless the best value at which μ can be set in order to get the maximum convergence rate.

To find the optimal value of μ a sequence of values is generated; then each value in the sequence, 100 different realizations of the same first order AR model were simulated, measuring the mean overall MSE between the whole true and estimated input sequence and the mean final MSE between the last 100 samples of the same sequences. Figure 36 shows the behavior of both the overall and final MSE as a function of μ.

Both the curves feature a minimum, but they hare located at very different positions along the μ axis. The minimum of the overall MSE corresponds to higher values of μ as the duration of the initial transient is lessened in this case. On the other hand, the final MSE curves reach its minimum for low values of μ as the prediction of the AR model parameters is much more stable and reliable.

Figure 37 displays a comparison of the different LMS algorithm behaviors for some values of the step-size parameter μ. For each value of μ, 100 different realizations of the same

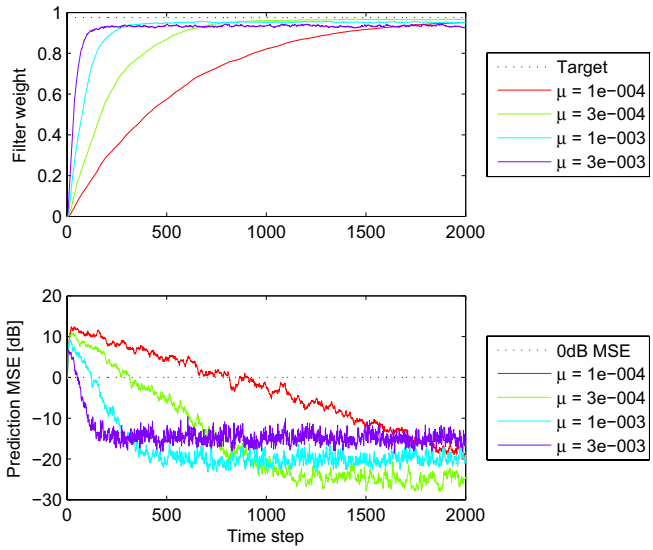


Figure 37: LMS algorithm convergence behavior for different values of μ and constant $\mathbf{H}$: larger values of the step-size parameter corresponds to higher convergence rates.

first order AR model were simulated, recording both the mean estimated AR coefficient and measuring the mean MSE between the true and estimated input sequence.

Larger values of the step-size parameter corresponds to higher convergence rates, meaning that the estimated value of the filter coefficient more rapidly converges to the target value. However, the same values of μ corresponds to larger estimation errors, which should be avoided in order to correctly estimate the input signal $x[n]$.

In the more general case of $\mathbf{H}$ dependent on time, it is still possible to have convergence of $\tilde{\mathbf{H}}$ to $\mathbf{H}$: if $\mathbf{H}$ changes slowly in time, the AWC may have enough time to update $\tilde{\mathbf{H}}$, as shown in Figure 38. However, if an abrupt change happens only high value of μ allow to track the AR model.

Here, for each value of μ, 100 different realizations of the same time-varying first order AR model were simulated, measuring the mean overall MSE between the whole true and estimated input sequence and the mean finale MSE between the last 100 samples of the same sequences. It is evident that the higher the value of μ, the more reliable the tracking of $\mathbf{H}$ is.

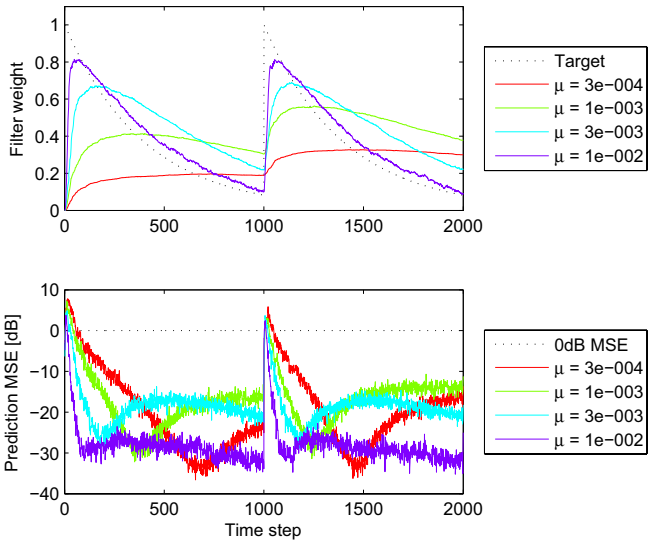


Figure 38: LMS algorithm convergence behavior for different values of μ and time-varying $\mathbf{H}$: only the largest value of the step-size parameter allows for a correct tracking.

Comparing Figure 37 with Figure 38, it seems that finding the best solution for tracking both time-dependent and time-independent AR models requires a way to adaptively change the time-step parameter μ. In this case, in fact, it could be possible to smoothly move from one performance curve to another, exploiting both the fast convergence rate corresponding to high values of μ, and the smooth and reliable estimation of $\mathbf{H}$ due to lower values of μ.

6.1.1 *Adaptive Gain LMS*

An improvement of (6.6) which allows for the updating of the step-size parameters involves a steepest-descent iterative minimization on μ.

$$\begin{aligned} \tilde{H}_i[n+1] &= \tilde{H}_i[n] - \mu[n]\frac{\partial J[n]}{\partial \tilde{H}_i[n]} \\ \mu[n+1] &= \left[\mu[n] - \alpha\frac{\partial J[n]}{\partial \mu[n]}\right]_{\mu-}^{\mu+} \end{aligned} \tag{6.9}$$

where $\alpha > 0$ is called *learning-rate parameter*, while $\mu-$ and $\mu+$ are respectively the lower and upper saturating threshold on μ, used to keep the estimated value of μ within

acceptable bounds.

By expanding the first equation in (6.9), a relationship identical to (6.8) is found, with the only notable exception that here μ depends on the time step n:

$$\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu[n]e[n]u[n-i] \tag{6.10}$$

Doing the same with the second equation of (6.9), an auxiliary set of variables $\Phi_i[n]$ must be introduced into the computation:

$$\begin{aligned} \Psi_i[n] &= \frac{\partial \tilde{H}_i[n]}{\partial \mu[n]} \\ \mu[n+1] &= \left[\mu[n] - \alpha e[n] \sum_{i=1}^{N} u[n-i]\Psi_i[n]\right]_{\mu-}^{\mu+} \end{aligned} \tag{6.11}$$

Here $\Psi_i[n]$ represents the gradient of the LMS adaptive filter coefficients with respect to the step-size parameter. The update rule for $\Psi_i[n]$ is found differentiating (6.10) with respect to $\mu[n]$:

$$\begin{aligned} \eta[n] &= e[n] + \mu[n] \sum_{i=1}^{N} u[n-i]\Psi_i[n] \\ \Psi_i[n+1] &= \Phi_i[n] - u[n-i]\eta[n] \end{aligned} \tag{6.12}$$

where $\eta[n]$ is another auxiliary variable representing the input signal second order estimation, as it appears more evident from the following relationship

$$\eta[n] = d[n] + \sum_{i=1}^{N} u[n-i](H_i[n] + \mu[n]\Psi_i[n]) \tag{6.13}$$

obtained from (6.12) by expanding $e[n]$ and collecting $u[n-i]$.

In conclusion, at each time step the Adaptive Gain LMS algorithm performs the following operations

1. $f[n] = \sum_{i=1}^{N} u(n-i)H_i[n]$
2. $\phi[n] = \sum_{i=1}^{N} u(n-i)\Psi_i[n]$
3. $e[n] = d[n] + f[n]$
4. $\eta[n] = e[n] + \mu[n]\phi[n]$
5. $\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu[n]e[n]u[n-i]$
6. $\Psi_i[n+1] = \Psi_i[n] - \eta[n]u[n-i]$
7. $\mu[n+1] = [\mu[n] - \alpha e[n]\phi[n]]_{\mu-}^{\mu+}$

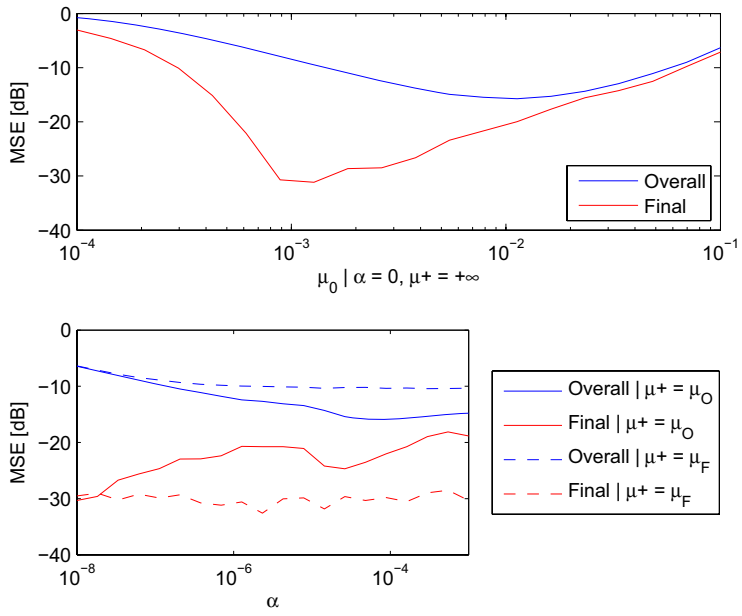


Figure 39: Adaptive Gain LMS algorithm performance dependency on the step-size parameter: two different optimization criteria can be chosen, leading to very different algorithm behaviors.

The lower threshold $\mu-$ is usually set to 0, while $\mu+$ and α are left free for optimization. A typical procedure used to setup these parameters is made up of two steps: first, with $\alpha = 0$, the best value of μ is found optimizing the overall or final MSE in a fashion like to what was discussed in the previous section; $\mu+$ is then set to two times this value. Finally, α is optimized, looking again for that particular value which minimizes the overall or final MSE.

Figure 39 displays the estimated dependency of the two discussed quantities as a function of $\mu+$ and α. Also in this case 100 different realizations of the same time-independent first order AR model were simulated, measuring each time the mean overall MSE between the whole true and estimated input sequence and the mean finale MSE between the last 100 samples of the same sequences.

When $\mu+$ is chosen so to optimize the final MSE ($\mu+ = \mu_F \simeq 2 \cdot 8.8\ 10^{-4}$), the parameter α does not have a significant influence on the final MSE; the overall MSE however gets lower for larger values of α, so α around 10^{-4} should

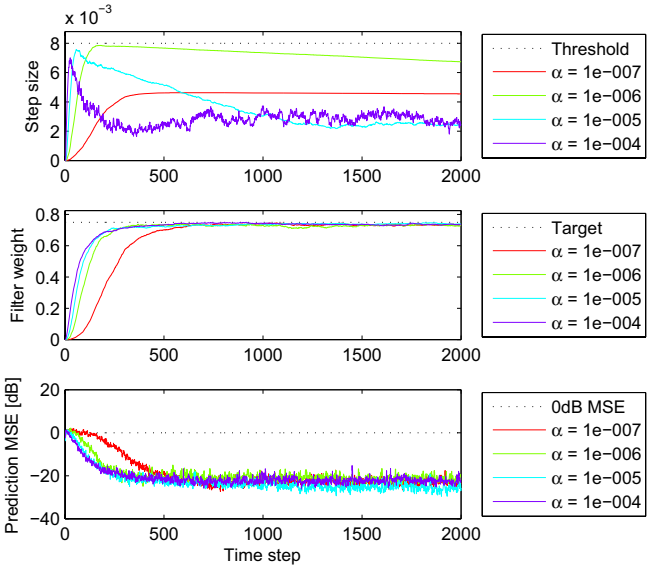


Figure 40: Adaptive Gain LMS algorithm performance on time-invariant AR model: once a proper $\mu+$ value is chosen, good performance are obtained for a wide range of learning-rate parameters.

be chosen. In the case of $\mu+$ optimized for overall MSE ($\mu+ = \mu_O \simeq 2 \cdot 10^{-2}$), the parameter α influences both the final and overall MSE: high values of α can be used to further reduce the overall MSE, while optimization of the final MSE requires very low value of α.

Figure 40 displays a comparison of the different LMS algorithm behaviors for some values of the learning-rate parameter α for a fixed valued of the upper saturating threshold $\mu+$ in the case of a time-invariant AR model. For each value of α, 100 different realizations of the same first order AR model were simulated, recording the mean estimated optimal step-size parameter, the mean estimated AR coefficient and measuring the mean MSE between the true and estimated input sequence.

All the values of α in the simulated range allows for both a good convergence rate and a low final MSE. Higher values of α corresponds to a faster convergence of the estimated AR model coefficient to its true value.

Figure 41 displays a comparison of the different LMS algorithm behaviors for some values of the learning-rate

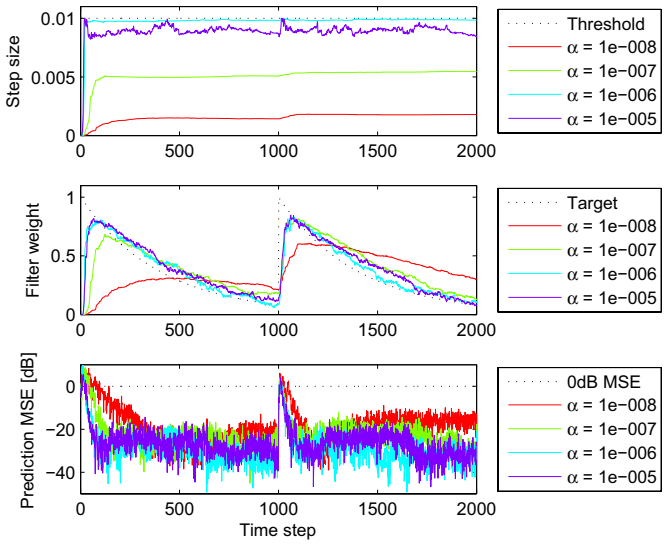


Figure 41: Adaptive Gain LMS algorithm performance on time-variant AR model: good tracking performance strongly depends on the value of the learning-rate parameter.

parameter α for a fixed valued of the upper saturating threshold $\mu+$ in the case of a time-variant AR model, featuring both slow and abrupt change in model parameters.

Even in this case it is possible to have $\tilde{\mathbf{H}}$ converge to $\mathbf{H}$: higher values of the learning-rate parameter can be used to improve tracking performance in presence of discontinuities in the AR coefficients.

6.2 NON-LINEAR EVEN MOMENTS

The LMS algorithm belongs to a class of adaptive methods designated as Second Order Statistics (SOS): this family of techniques deals only with signal power, thus requiring only the first and second moment of the input signal to be estimated. These methods are suitable when the signal to be filtered is Gaussian as these two quantities completely characterize the signal. As already discussed, very interesting simplifications leads to algorithms which computationally slight intensive, involving only linear operations.

A more versatile family is constituted by the Higer Order Statistics (HOS) adaptive algorithms, which make use of a

larger set of signal moments, making the techniques belonging to this class capable of dealing with non-Gaussian input signals. Although these algorithms require more intensive computations since they usually involve non linear operations, they yield better approximations of the true distribution of the elaborated signals [72].

The simplest HOS extension of the LMS algorithm is the Least Mean Fourth (LMF) adaptive filter [73]. In this algorithm, the function the AWC rule aims to minimize is substituted by the following fourth order function:

$$J[n] = \frac{1}{4} e[n]^4 \tag{6.14}$$

Higher order function can be used as well: suitable function are typically in the form

$$J[n] = \frac{1}{2K} e[n]^{2K} \tag{6.15}$$

where K is a suitable integer used to chose the model order.

Applying (6.6) to this function choice, the following AWC rule is obtained:

$$\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu e[n]^{2K-1} u[n-i] \tag{6.16}$$

which is identical to (6.8) with the only exception of the non linear power operator applied to the estimated signal $e[n]$.

A further optimization can be made observing that any positive weight combination of even powers is a convex function: this means that the resulting function has no local minima and only one global minimum. It is thus suitable for steepest-descent optimization.

In this case AWC rule minimizes

$$J[n] = \sum_{i=1}^{M} \frac{a_i}{2i} e[n]^{2i} \tag{6.17}$$

where M set the maximum model order and a_i is a set of positive weights defining the convex minimization surface.

Applying (6.6) in this case too the AWC rule becomes:

$$\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu \left(\sum_{i=1}^{M} a_i e[n]^{2i-1} \right) u[n-i] \tag{6.18}$$

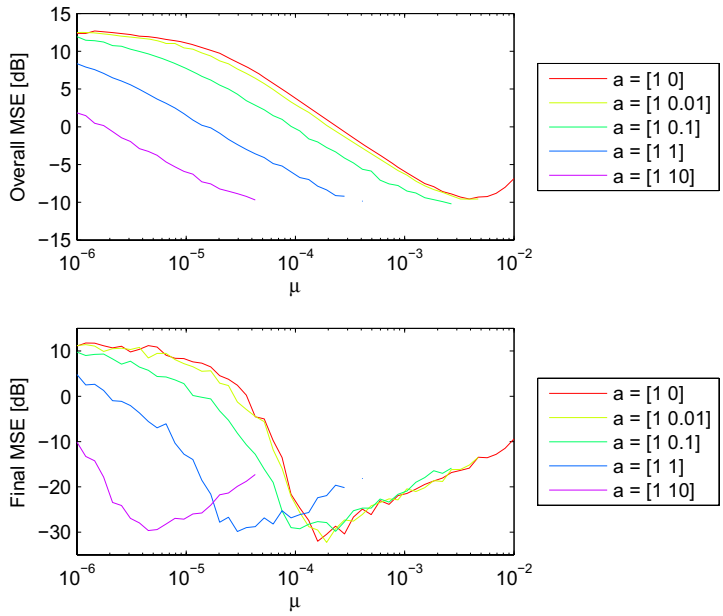


Figure 42: NEM algorithm performance dependency on step-size and function weights: overall MSE is greatly diminished for larger values of a_2, however μ must be kept low to ensure convergence.

The algorithm originating from this choice is called Nonlinear Even Moments (NEM) [72].

At each time step the NEM algorithm compute the following operations

$$
\begin{aligned}
&1. \quad f[n] = \sum_{i=1}^{N} u(n-i)H_i[n] \\
&2. \quad e[n] = d[n] + f[n] \\
&4. \quad \epsilon[n] = \sum_{i=1}^{M} a_i e[n]^{2i-1} \\
&3. \tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu\epsilon[n]u[n-i]
\end{aligned}
$$

Optimization of this algorithms involves the choice of both μ and the minimization function weights. While for the latter no optimization policy is known, the former can be again chosen to minimize the overall or final MSE between the true and the estimated input signal.

Setting different values of a_2 and performing the same tests done for the previous algorithms, Figure 42 was obtained: this graph shows that as a_2 becomes larger, the overall MSE reduces with respect to the same value of μ.

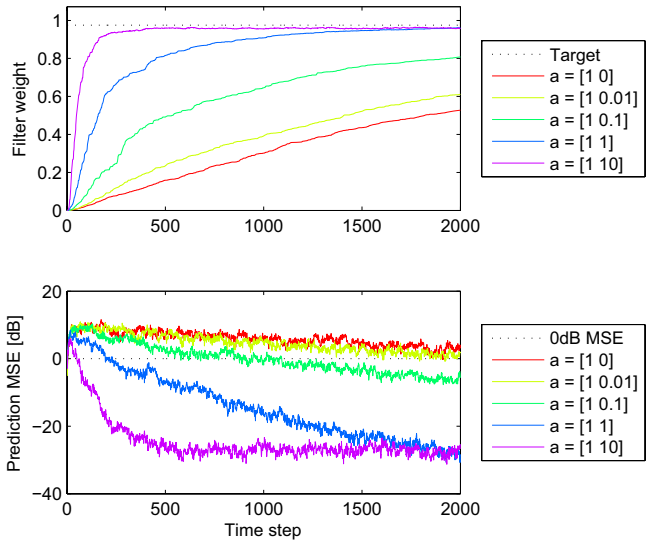


Figure 43: NEM algorithm convergence study on a time-invariant AR model for a varying a_2 and constant μ: high values of a_2 allow for a faster convergence.

This means that higher convergence rate can be obtained at a cost of lower values of μ. Moreover, the minima of the two curves gets closer when a_2 gets larger, meaning that it becomes easier to find a value of μ which minimizes both the overall and final MSE.

Figure 43 is a study of the convergence behavior of the NEM algorithm performed on 100 different realizations of the same first order time-invariant AR model, for the same value of the step-size parameter μ. The weight a_1 was kept constant at the value of 1, while a_2 was set to different values; the mean estimated AR coefficient the mean MSE between the true and estimated input sequence were recored. All the values of a_2 allowed NEM convergence to the target coefficient value: as anticipate in Figure 42, higher values of a_2 corresponds to faster convergence.

Figure 43 shows a comparisons of the different convergence behaviors of the NEM algorithm estimated averaging 100 different realizations of the same first order time-variant AR model, for the same value of the step-size parameter μ. The same policy of Figure 43 was used for the function weights.

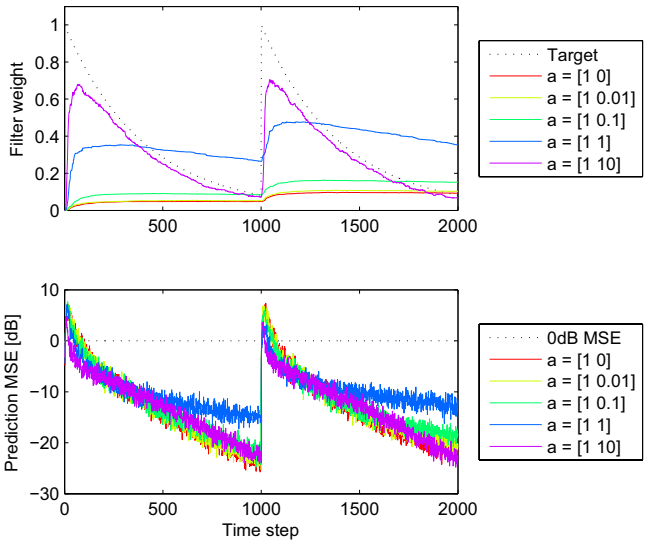


Figure 44: NEM algorithm convergence study on a time-variant AR model for a varying a_2 and constant μ: high values of a_2 allow for a faster convergence.

The tracking of time-variant AR models is not as successful as for the case of the adaptive gain LMS algorithm, nevertheless good agreement between the estimated filter weight and its true value can be obtained for high values of a_2.

An adaptive gain version of the NEM algorithm can be devised in a manner very similar to the one used for the LMS adaptive filter, as discussed in Section A.1. The operation scheduled at each time step for this algorithm are the following:

1. $f[n] = \sum_{i=1}^{N} u[n-i]\tilde{H}_i[n]$
2. $\phi[n] = \sum_{i=1}^{N} u[n-i]\Psi_i[n]$
3. $e[n] = d[n] + f[n]$
4. $\epsilon_0[n] = \sum_{i=1}^{M} a_i e[n]^{2i-1}$
5. $\epsilon_1[n] = \sum_{i=1}^{M} (2i-1) a_i e[n]^{2i-2}$
6. $\eta[n] = \epsilon_0[n] + \mu[n]\epsilon_1[n]\phi[n]$
7. $\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu[n]\epsilon_0[n]u[n-i]$
8. $\Psi_i[n+1] = \Psi_i[n] - \eta[n]u[n-i]$
9. $\mu[n+1] = [\mu[n] - \alpha\epsilon_0[n]\phi[n]]_{\mu-}^{\mu+}$

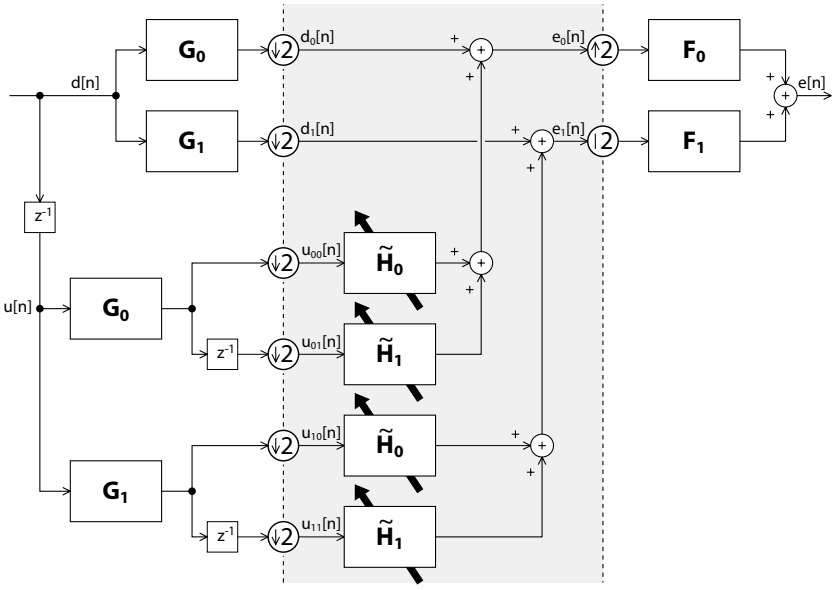


Figure 45: Schematic of the sub-band adaptive filter: this design is equivalent to the left side of figure 35; within the gray shaded region, processing happens at a different rate.

6.3 SUB-BAND ADAPTIVE FILTER

Another way to improve the convergence rate of the LMS algorithm is to use a sub-band decomposition scheme for the filtering process. In fact it has been observed [74] that convergence performance depends on the length of the adaptive filter: the longer the filter, the slower the convergence.

Moreover, in ultrasound processing applications the signal elaborated by the adaptive filter is correlated: the standard LMS algorithm guarantee stability for a proper value of μ only for uncorrelated signals. Adopting a sub-band decomposition scheme may be helpful, as correlation length in each band can be diminished exploiting properly designed filter banks.

These advantages come at the cost of the introduction of an additional analysis and a synthesis section, enclosing a region where processing happens at rate different from the external environment: the Sub-band Adaptive Filter (SAF) algorithm in fact belongs to the so called *multirate* filter class.

In order to perform sub-band decomposition, a pair of perfect reconstruction analysis and synthesis filters are needed, respectively denoted in Figure 45 by $\mathbf{G}_0(z)$, $\mathbf{G}_1(z)$, $\mathbf{F}_0(z)$ and $\mathbf{F}_1(z)$. Orthogonal Wavelet filters were chosen for this application.

Next, polyphase decomposition must be applied to the FIR filter $\tilde{\mathbf{H}}$:

$$\tilde{\mathbf{H}}(z) = \tilde{\mathbf{H}}_0(z^2) + z^{-1}\tilde{\mathbf{H}}_1(z^2) \tag{6.19}$$

where $\tilde{\mathbf{H}}_0$ and $\tilde{\mathbf{H}}_1$ are both FIR filters accounting respectively for the even and odd coefficients of $\tilde{\mathbf{H}}$.

Finally, exploiting the linearity of the filtering and down-sample operation it is possible to highlight two similar signal paths: one originating from $u_0[n]$ and ending into $e_0[n]$, the other from $u_1[n]$ to $e_1[n]$. Each path corresponds to a different signal band: because of the Wavelet choice, filter $\mathbf{G}_0$ is associated to the lower half-band, while $\mathbf{G}_1$ to the upper half.

In this case, the function cost to be minimized is chosen as:

$$J[n] = \frac{1}{2}(a_0 e_0[n]^2 + a_1 e_1[n]^2) \tag{6.20}$$

where a_0 and a_1 are two positive real constants.

Applying (6.6) to this function choice, it is necessary to distinguish between the filter coefficients belonging respectively to $\tilde{\mathbf{H}}_0$ and $\tilde{\mathbf{H}}_1$. Because of this, the AWC rule is effectively composed by two rules

$$\begin{aligned} \tilde{H}_{0,i}[n+1] &= \tilde{H}_{0,i}[n] - \mu\frac{\partial J[n]}{\partial \tilde{H}_{0,i}[n]} \\ \tilde{H}_{1,i}[n+1] &= \tilde{H}_{1,i}[n] - \mu\frac{\partial J[n]}{\partial \tilde{H}_{1,i}[n]} \end{aligned} \tag{6.21}$$

where $\tilde{H}_{0,i}[n]$ and $\tilde{H}_{1,i}[n]$ are respectively the i-th coefficients of $\tilde{\mathbf{H}}_0$ and $\tilde{\mathbf{H}}_1$.

Expanding the partial derivatives in (6.21) the following relationship are obtained:

$$\begin{aligned} \frac{\partial J[n]}{\partial \tilde{H}_{0,i}[n]} &= a_0 e_0[n]\frac{\partial e_0[n]}{\partial \tilde{H}_{0,i}[n]} + a_1 e_1[n]\frac{\partial e_1[n]}{\partial \tilde{H}_{0,i}[n]} \\ \frac{\partial J[n]}{\partial \tilde{H}_{1,i}[n]} &= a_0 e_0[n]\frac{\partial e_0[n]}{\partial \tilde{H}_{1,i}[n]} + a_1 e_1[n]\frac{\partial e_1[n]}{\partial \tilde{H}_{1,i}[n]} \end{aligned} \tag{6.22}$$

Then, from Figure 45, it is possible to write down both $e_0[n]$ and $e_1[n]$ respectively as

$$\begin{aligned} e_0[n] = d_0[n] &+ \sum_{i=1}^{N/2} u_{00}[n-i]\tilde{H}_{0,i}[n] \\ &+ \sum_{i=1}^{N/2} u_{01}[n-i]\tilde{H}_{1,i}[n] \\ e_1[n] = d_1[n] &+ \sum_{i=1}^{N/2} u_{10}[n-i]\tilde{H}_{0,i}[n] \\ &+ \sum_{i=1}^{N/2} u_{11}[n-i]\tilde{H}_{1,i}[n] \end{aligned} \tag{6.23}$$

Thus, the generic term $\partial e_\alpha[n]/\partial\tilde{H}_{\beta,i}[n]$ becomes

$$\frac{\partial e_\alpha[n]}{\partial\tilde{H}_{\beta,i}[n]} = u_{\alpha\beta}[n-i] \tag{6.24}$$

Substituting (6.24) into (6.22), (6.21) becomes

$$\begin{aligned} \tilde{H}_{0,i}[n+1] &= \tilde{H}_{0,i}[n] - \mu a_0 e_0[n]u_{00}[n-i] \\ &\quad -\mu a_0 e_0[n]u_{01}[n-i] \\ \tilde{H}_{1,i}[n+1] &= \tilde{H}_{1,i}[n] - \mu a_0 e_0[n]u_{10}[n-i] \\ &\quad -\mu a_0 e_0[n]u_{11}[n-i] \end{aligned} \tag{6.25}$$

It is in general possible to extend the SAF algorithm to an arbitrary number of sub-bands exploiting, for example, the structure of the Wavelet Packet Transform (WPT): in this case, the filters $\mathbf{G}_0$, $\mathbf{G}_1$ are substituted by the orthogonal decomposition filter bank $\mathcal{G}$, as well as $\mathbf{F}_0$, $\mathbf{F}_1$ by the reconstruction bank $\mathcal{F}$. At the same time, the coefficients of the filter $\tilde{\mathbf{H}}$ are subdivided into smaller groups, according to the remainder of the division of their index by the total number of bands: if K bands are used, coefficients $\tilde{H}_0$, $\tilde{H}_K$, $\tilde{H}_{2K}$, ... will be grouped together; the same will happen for coefficients $\tilde{H}_1$, $\tilde{H}_{K+1}$, $\tilde{H}_{2K+1}$, ... and so on.

The AWC rule thus become

$$\tilde{H}_{\beta,i}[n+1] = \tilde{H}_{\beta,i}[n] - \mu \sum_{\alpha=0}^{K-1} a_\alpha e_\alpha[n]u_{\alpha\beta}[n-i] \tag{6.26}$$

where α is an integer index running from 0 to $K-1$.

In conclusion, at each time step within the gray shaded region of Figure 45, for $\alpha, \beta \in [0, K-1]$, the SAF algorithm performs the following operations:

1. $f_\alpha[n] = \sum_{\beta=0}^{K-1} \sum_{i=1}^{N/K} u_{\alpha\beta}[n-i]H_{\beta,i}[n]$
2. $e_\alpha[n] = d_\alpha[n] + f_\alpha[n]$
3. $\tilde{H}_{\beta,i}[n+1] = \tilde{H}_{\beta,i}[n] - \mu \sum_{\alpha=0}^{K-1} a_\alpha e_\alpha[n]u_{\alpha\beta}[n-i]$

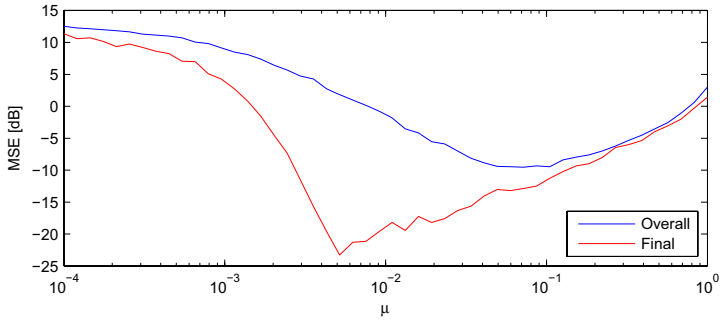


Figure 46: SAF algorithm overall and final MSE dependency on the step-size parameter: the minima of the two curves are closer each other than the standard LMS algorithm.

Processing before entering this region involves filtering with the analysis filter bank and downsampling; after exiting the same region, upsampling, filtering with the synthesis filter bank and summation is performed.

To optimize the SAF behavior, the set of weight a and the step-size parameter can be used. In particular, it is useful to choose each a_α as the reciprocal of the power associated to the corresponding signal band α. This choice allows for faster convergence rates of the SAF algorithm [74].

As it is usually not possible to know in advance these values, or when the signal to be elaborated is non-stationary, power estimations after the processing of the input signal $u[n]$ with the corresponding analysis filter $\mathbf{G}_\alpha$ can be performed. The reciprocal of these estimations are used in place of the constant weighting coefficients. This technique is useful to track time variations of the power distribution among signal bands.

To find the optimal value of μ, a technique identical to the one used for the standard LMS algorithm may be employed. Figure 46 shows how final and overall MSE change as function of μ, averaging the results of 100 simulations of the same second order AR model. The SAF algorithm was setup for estimating a second order filter using a two band scheme designed on the Haar wavelet.

Both curves feature a minimum, located each other closer than the minima of the basic LMS algorithms. As already discussed for the NEM algorithm in the previous section, this means that it is easier to find a value of μ capable to

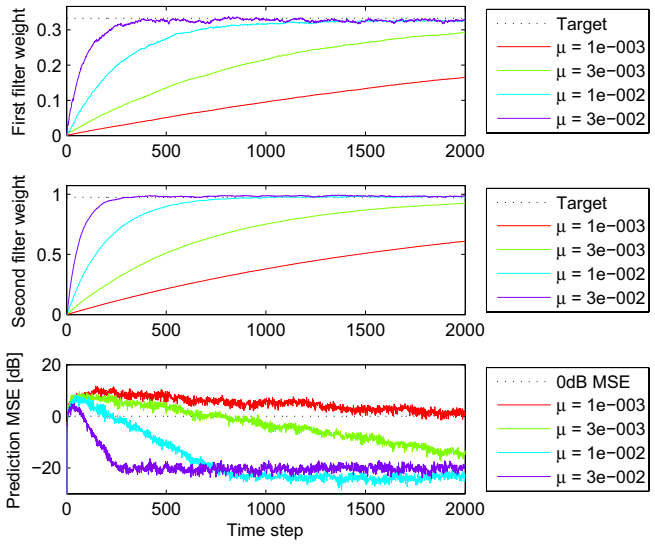


Figure 47: SAF algorithm convergence behavior for different values of μ and time-invariant AR model: both coefficients can be estimated quite accurately with fast convergence.

minimize both the overall and final MSE.

Figure 47 displays a comparison of the different SAF algorithm behaviors for some values of the step-size parameter μ. For each value of μ, 100 different realizations of the same second order time-invariant AR model were simulated, recording both the mean estimated AR coefficients and measuring the mean MSE between the true and estimated input sequence.

The results obtained are in accordance to Figure 46: in fact, $\mu = 3e-2$ allows for faster convergence, a lower overall MSE but a higher final one; on the other side, when $\mu = 1e-2$, a slower convergence and an higher overall MSE are found, while the final MSE improves of almost 3 dB.

In the case of a second order time-variant AR model, tracking performance are globally worse than the adaptive gain LMS algorithm, with a prediction MSE of at most -20 dB when convergence is reached. As expected, higher values of μ corresponds to faster tracking.

Again, an adaptive gain version of the NEM algorithm can be derived to improve the time-tracking capabilities,

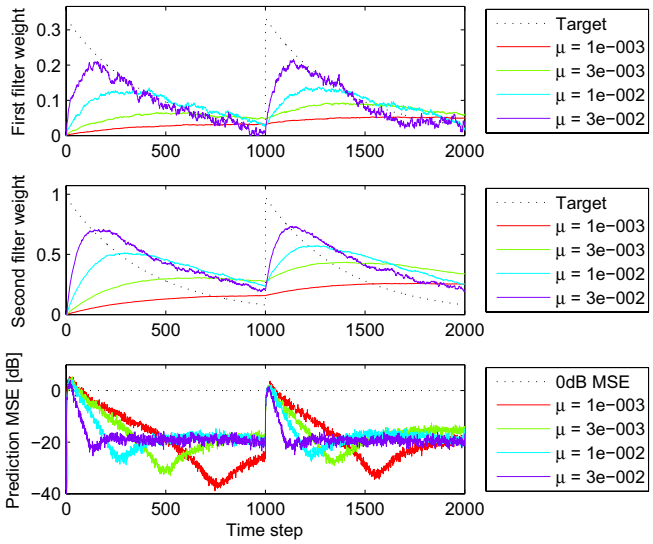


Figure 48: SAF algorithm convergence behavior for different values of μ and time-variant AR model: tracking performance are much worse than the standard LMS algorithm.

as discussed in Section A.2. For this adaptive algorithm, the following operations will be executed within the gray shaded region for $\alpha, \beta \in [0, K-1]$ instead of the already listed ones:

1. $f_\alpha[n] = \sum_{\beta=0}^{K-1} \sum_{i=1}^{N/K} u_{\alpha\beta}[n-i]\tilde{H}_{\beta,i}[n]$
2. $\phi_\alpha[n] = \sum_{\beta=0}^{K-1} \sum_{i=1}^{N/K} u_{\alpha\beta}[n-i]\Psi_{\beta,i}[n]$
3. $e_\alpha[n] = d_\alpha[n] + f_\alpha[n]$
4. $\eta_\alpha[n] = e_\alpha[n] + \mu n \phi_\alpha[n]$
5. $\tilde{H}_{\beta,i}[n+1] = \tilde{H}_{\beta,i}[n] - \mu(n) \sum_{\alpha=0}^{K-1} a_\alpha e_\alpha[n] u_{\alpha\beta}[n-i]$
6. $\Psi_{\beta,i}[n+1] = \Psi_{\beta,i}[n] - \sum_{\alpha=0}^{K-1} a_\alpha \eta_\alpha[n] u_{\alpha\beta}[n-i]$
7. $\mu[n+1] = \left[\mu[n] - \xi \sum_{\alpha=0}^{K-1} a_\alpha e_\alpha[n] \phi_\alpha[n]\right]_{\mu-}^{\mu+}$

Processing outside this region goes on as in the fixed gain algorithm: the same structure of filter banks, downsamplers and upsamplers may be used.

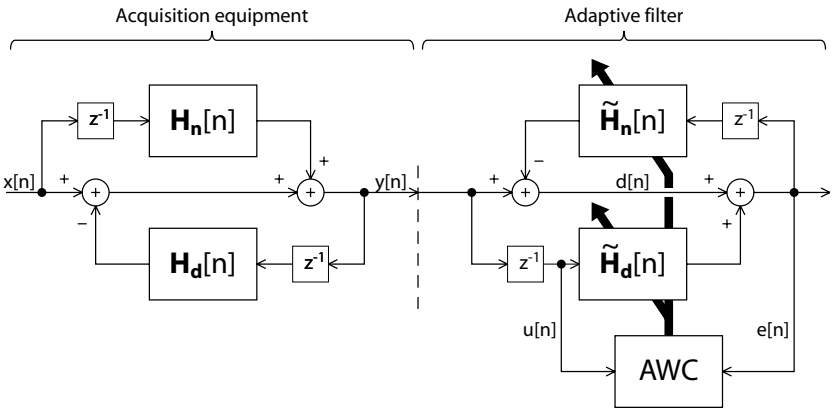


Figure 49: IIR configuration LMS adaptive filter for signal deconvolutions: this structure can be useful for restoring either very long finite or infinite deformations with high convergence rates.

6.4 INFINITE IMPULSE RESPONSE LMS

Switching to an Intinite Impulse Response (IIR) filter model is another way to improve the convergence rate of the LMS algorithm when dealing with very high order AR model. This is moreover the most straightforward solution to recover the input sequence of an Auto-Regressive Moving-Average (ARMA) model [71] as shown in Figure 49. In fact, the adaptive filter structure is the mirror inverse version of the one generating the spectrally deformed signal $y[n]$.

In this case, two adaptive filters are used, one recovering AR part of the model ($\tilde{\mathbf{H}}_\mathbf{d}$), the other estimating the Moving-Average (MA) part ($\tilde{\mathbf{H}}_\mathbf{n}$). The coefficients of these two filters will become estimates of the respective elements of $\mathbf{H_d}$ and $\mathbf{H_n}$.

Some consideration must be made when the same filtering structure is used to estimate the coefficients of an AR model. In fact, only the filter $\mathbf{H_d}$ is present in this case. If a number of coefficients equal or greater than the AR model order is used for the $\tilde{\mathbf{H}}_\mathbf{d}$, filters $\tilde{\mathbf{H}}_\mathbf{n}$ and $\tilde{\mathbf{H}}_\mathbf{d}$ will eventually converge respectively to 0 and to $\mathbf{H_d}$ in the same fashion of the standard LMS algorithm.

However, if the number of coefficients is lower than the AR model order, it is not possible for the two filters to converge to the correct values. Nevertheless, the adaptive algorithm

is forced by its AWC rule to recover the original signal $x[n]$ at its best, using all its degree of freedom. Exploiting the Z-transform of $\tilde{\mathbf{H}}_\mathbf{n}$ and $\tilde{\mathbf{H}}_\mathbf{d}$ it is possible to obtain an estimate of $\mathbf{H}_\mathbf{d}$.

In fact, with reference to Figure 49, in the Z-transform domain the following relationships does hold

$$\begin{aligned} Y(z) &= \frac{1}{1+z^{-1}H_d(z)}X(z) \\ E(z) &= \frac{1+z^{-1}\tilde{H}_d(z)}{1+z^{-1}\tilde{H}_n(z)}Y(z) \end{aligned} \tag{6.27}$$

where $H_d(z)$, $\tilde{H}_d(z)$ and $\tilde{H}_n(z)$ are respectively the Z-transform of the filters $\mathbf{H}_\mathbf{d}$, $\tilde{\mathbf{H}}_\mathbf{d}$ and $\tilde{\mathbf{H}}_\mathbf{n}$.

If convergence is reached $E(z) = X(z)$, so it must be

$$1 + z^{-1}H_d(z) = \frac{1 + z^{-1}\tilde{H}_d(z)}{1 + z^{-1}\tilde{H}_n(z)} \tag{6.28}$$

which leads to the following relationship between $H_d(z)$ and the two adaptive filters

$$H_d(z) = \frac{\tilde{H}_d(z) - \tilde{H}_n(z)}{1 + z^{-1}\tilde{H}_n(z)} \tag{6.29}$$

This is an effective mean for estimating very long $\mathbf{H}_\mathbf{d}$ filters using short adaptive filters. In fact (6.29) can potentially account for infinite length $\mathbf{H}_\mathbf{d}$ filters due to its structure, identical to the one of an IIR filter. A similar result can be obtained to estimate the parameters of a MA model, simply by switching $\tilde{\mathbf{H}}_\mathbf{d}$ with $\tilde{\mathbf{H}}_\mathbf{n}$ and vice versa:

$$H_n(z) = \frac{\tilde{H}_n(z) - \tilde{H}_d(z)}{1 + z^{-1}\tilde{H}_d(z)} \tag{6.30}$$

In order to derive the sequence of operations defining the AWC rule, the relationship linking the inputs $u[n]$ and $d[n]$ to the estimated signal must be drawn:

$$e[n] = d[n] + \sum_{i=1}^{M} H_{d,i}[n]u[n-i] - \sum_{i=1}^{N} H_{n,i}[n]e[n-i] \tag{6.31}$$

where M and N are respectively the order of the filters $\tilde{\mathbf{H}}_\mathbf{d}$ and $\tilde{\mathbf{H}}_\mathbf{n}$, and $H_{d,i}[n]$ and $H_{n,i}[n]$ their i-th coefficients at time-step n.

The AWC rule minimizes the same instantaneous cost function of the FIR version, so $J[n] = e[n]^2/2$. If a fixed gain steepest-descent algorithm is applied to optimize both the filters, the following rules are obtained:

$$\begin{aligned} \tilde{H}_{d,i}[n+1] &= \tilde{H}_{d,i}[n] - \mu \frac{\partial J[n]}{\partial \tilde{H}_{d,i}[n]} \\ \tilde{H}_{n,i}[n+1] &= \tilde{H}_{n,i}[n] - \mu \frac{\partial J[n]}{\partial \tilde{H}_{n,i}[n]} \end{aligned} \tag{6.32}$$

Expanding the two partial derivatives, the dependency of the current estimated value $e[n]$ from its past samples must be considered:

$$\begin{aligned} \frac{\partial J[n]}{\partial \tilde{H}_{d,i}[n]} &= e[n] \frac{\partial e[n]}{\partial \tilde{H}_{d,i}[n]} \\ &= e[n] \left(u[n-i] - \textstyle\sum_{j=1}^{N} \tilde{H}_{n,j}[n] \frac{\partial e[n-j]}{\partial \tilde{H}_{d,i}[n]} \right) \\ \frac{\partial J[n]}{\partial \tilde{H}_{n,i}[n]} &= e[n] \frac{\partial e[n]}{\partial \tilde{H}_{n,i}[n]} \\ &= e[n] \left(-e[n-i] - \textstyle\sum_{j=1}^{N} \tilde{H}_{n,j}[n] \frac{\partial e[n-j]}{\partial \tilde{H}_{n,i}[n]} \right) \end{aligned} \tag{6.33}$$

If the two filters adapt slowly enough, the following approximations hold [71]:

$$\begin{aligned} \frac{\partial e[n-j]}{\partial \tilde{H}_{d,i}[n]} &\simeq \frac{\partial e[n-j]}{\partial \tilde{H}_{d,i}[n-j]} = \alpha_i[n-j] \\ \frac{\partial e[n-j]}{\partial \tilde{H}_{n,i}[n]} &\simeq \frac{\partial e[n-j]}{\partial \tilde{H}_{n,i}[n-j]} = \beta_i[n-j] \end{aligned} \tag{6.34}$$

Combining (6.34) with (6.33) highlights another couple of recursive relationships:

$$\begin{aligned} \alpha_i[n] &= u[n-i] - \textstyle\sum_{j=1}^{N} \tilde{H}_{n,j}[n] \alpha_i[n-j] \\ \beta_i[n] &= -e[n-i] - \textstyle\sum_{j=1}^{N} \tilde{H}_{n,j}[n] \beta_i[n-j] \end{aligned} \tag{6.35}$$

Finally, it is possible to plug $\alpha_i[n]$ and $\beta_i[n]$ into (6.32) obtaining

$$\begin{aligned} \tilde{H}_{d,i}[n+1] &= \tilde{H}_{d,i}[n] - \mu e[n] \alpha_i[n] \\ \tilde{H}_{n,i}[n+1] &= \tilde{H}_{n,i}[n] - \mu e[n] \beta_i[n] \end{aligned} \tag{6.36}$$

In conclusion, at each time-step, the IIR LMS algorithm

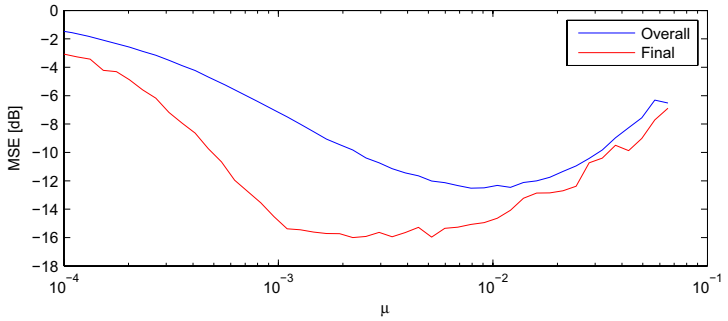


Figure 50: LMS IIR algorithm overall and final MSE dependency on the step-size parameter: overall results are slightly better than LMS FIR algorithm.

performs the following operations

1. $f_1[n] = \sum_{i=1}^{M} \tilde{H}_{d,i}[n]u[n-i]$
2. $f_2[n] = \sum_{i=1}^{N} \tilde{H}_{n,i}[n]e[n-i]$
3. $g_{1,i}[n] = \sum_{j=1}^{N} \tilde{H}_{n,j}[n]$
4. $g_{2,i}[n] = \sum_{j=1}^{N} \tilde{H}_{n,j}[n]$
5. $e[n] = d[n] + f_1[n] - f_2[n]$
6. $\alpha_i[n] = u[n-i] - g_{1,i}[n]$
7. $\alpha_i[n] = -e[n-i] - g_{2,i}[n]$
8. $\tilde{H}_{d,i}[n+1] = \tilde{H}_{d,i}[n] - \mu e[n]\alpha_i[n]$
9. $\tilde{H}_{n,i}[n+1] = \tilde{H}_{n,i}[n] - \mu e[n]\beta_i[n]$

Despite its complexity, only one parameter is needed to optimize the behavior of this algorithm: like the FIR LMS algorithm, μ is left free for optimization.

Adopting the same strategy employed for FIR LMS algorithm optimization, Figure 50 was obtained. The best overall MSE obtained in the case of a time-invariant first order ARMA model is better than the result reported for the FIR version on an AR model; however, a far worse result is found for the final MSE.

This may be due to the fact that the optimization surface has may local minima [71], in addition to global minima, meaning that, during optimization, LMS IIR algorithm can get stuck far away from the optimal solution.

This is clearly observable in Figure 51, where the LMS IIR algorithm is tested against a first order time-invariant ARMA

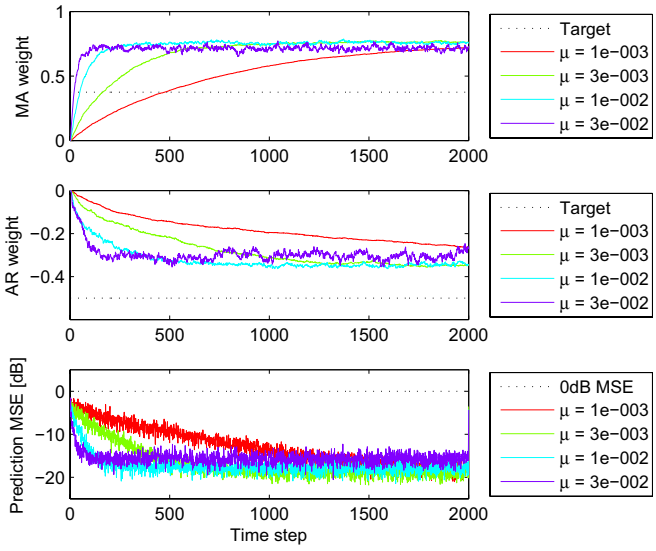


Figure 51: LMS IIR algorithm convergence behavior for different values of μ on a time-invariant ARMA model: however model weights are not correctly estimated, the resulting prediction MSE is acceptable.

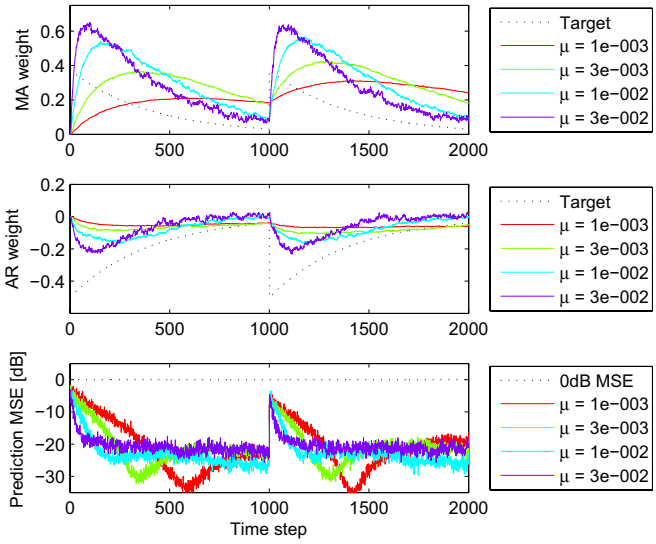


Figure 52: LMS IIR algorithm convergence comparisons for different values of μ on a time-variant ARMA model: a behavior very like to the time-invariant model is observed.

model. While the prediction MSE is acceptable and the convergence rate is still quite good, the estimated values for the AR and MA weights are not matching their corresponding true values. As in the case of the FIR version, higher values of the step-size parameter ar useful in order to obtain fast convergence rate, but almost always corresponds to higher prediction MSE, and vice versa.

The same happens in the case of a time-variant ARMA model: the algorithm is still capable to track both smooth and abrupt variations in the model parameters, however it converges to wrong values. In this case it is even more evident the fact that a time-based optimization of the step-size parameter can greatly help in achieving better results in the final prediction MSE while keeping an high convergence rate.

An adaptive gain version of the IIR LMS algorithm can be obtained following the procedure illustrated in the case of the standard LMS, as discussed in Section A.3. It has however a very high computational cost due to the presence of many filters linked to the feedback loops. At each time step, this algorithms computes the following filtering operations:

1. $f_1[n] = \sum_{i=1}^{M} u[n-i]\tilde{H}_{d,i}[n]$
2. $f_2[n] = \sum_{i=1}^{N} e[n-i]\tilde{H}_{n,i}[n]$
3. $f_3[n] = \sum_{i=1}^{N} \phi[n-i]\tilde{H}_{n,i}[n]$
4. $g_{1,i}[n] = \sum_{j=1}^{N} a_i[n-j]\tilde{H}_{n,j}[n]$
5. $g_{2,i}[n] = \sum_{j=1}^{N} b_i[n-j]\tilde{H}_{n,j}[n]$
6. $g_{3,i}[n] = \sum_{j=1}^{N} \alpha_i[n-j]\tilde{H}_{n,j}[n]$
7. $g_{4,i}[n] = \sum_{j=1}^{N} \beta_i[n-j]\tilde{H}_{n,j}[n]$
8. $\Gamma_1[n] = \sum_{i=1}^{M} u[n-i]\Psi_{d,i}[n]$
9. $\Gamma_2[n] = \sum_{i=1}^{N} e[n-i]\Psi_{n,i}[n]$
10. $\Gamma_{3,i}[n] = \sum_{j=1}^{N} a_i[n-j]\Psi_{n,j}[n]$
11. $\Gamma_{4,i}[n] = \sum_{j=1}^{N} b_i[n-j]\Psi_{n,j}[n]$

The remaining operations are linked to the update of the quantities that are input to the filters and give origin to the

final estimate $e[n]$:

$$
\begin{aligned}
12. \quad & e[n] = d[n] + f_1[n] - f_2[n] \\
13. \quad & a_i[n] = u[n-i] - g_{1,i}[n] \\
14. \quad & b_i[n] = -e[n-i] - g_{2,i}[n] \\
15. \quad & \phi[n] = \Gamma_1[n] - \Gamma_2[n] - f_3[n] \\
16. \quad & \alpha_i[n] = -\Gamma_{3,i}[n] - g_{3,i}[n] \\
17. \quad & \beta_i[n] = -\phi[n-i] - \Gamma_{4,1} - g_{4,i} \\
18. \quad & \eta[n] = e[n] + \mu[n]\phi[n] \\
19. & \tilde{H}_{d,i}[n+1] = \tilde{H}_{d,i}[n] - \mu[n]e[n]a_i[n] \\
20. & \tilde{H}_{n,i}[n+1] = \tilde{H}_{n,i}[n] - \mu[n]e[n]b_i[n] \\
21. & \Psi_{d_i}[n+1] = \Psi_{d_i}[n+1] - \eta[n]a_i[n] - \mu[n]e[n]\alpha_i[n] \\
22. & \Psi_{n_i}[n+1] = \Psi_{n_i}[n+1] - \eta[n]b_i[n] - \mu[n]e[n]\beta_i[n] \\
23. \quad & \mu[n+1] = [\mu[n] - Ke[n]\phi[n]]_{\mu-}^{\mu+}
\end{aligned}
$$

6.5 PERFORMANCE COMPARISON

As discussed in the introduction of this chapter, adaptive filtering techniques can be fruitfully employed to remove the time-varying signal deformations present in environments such as US imaging. Because of this, algorithm configurations discussed in this chapter were designed in order to perform signal deconvolution, so to estimate the input signal $x[n]$. This signal can be thought as the undeformed incoming echo before RF signal acquisition: the various AR models reflect different way to describe the effect the US transducer has on this signal.

In this section, the algorithms presented in this chapter are testbenched as de-blurring techniques for biomedical images to verify and compare their effectiveness. Using the same databases and metrics discussed in Section 5.4, images were processed with the proposed deconvolution algorithms initialized with null filters and the parameters listed in Table 4.

Two parameters category were identified and optimized: filter related parameters which include, where applicable, order, optimization function weights (NEM) and number of bands (SAF); on the other hand, step-size related parameters include, where applicable, the step-size and its upper threshold and the learning-rate (Adaptive Gain LMS).

Algorithm	Filter	Step-size	Power [dB]
LMS	$N = 2$	$\mu = 1.29$	-27.7
LMS-AG	$N = 3$	$\mu+ = 0.93, \alpha = 7.28$	-27.5
LMS-NEM	$N = 3, a_2 = 1.36$	$\mu = 0.15^{(*)}$	-25.2
LMS-SAF	$N = 4, K = 4$	$\mu = 0.23$	-29.3
LMS-IIR	$M = 3, N = 8$	$\mu = 0.13$	-29.3

Table 4: Optimized LMS parameters for US image deconvolution: the same parameters where used on *phantoms* and *in-vivo* images. (*) For *in-vivo* images $\mu = 0.053$ was found.

The values listed in Table 4 were obtained minimizing the overall estimated signal power $e[n]$ over a subset of the whole database. This procedure is equivalent to a minimization of the variance of estimated signal $e[n]$ and is one of the most widely used parameter optimization technique when it is impossible to know the true signal $x[n]$,

The main alternative to this optimization technique is based on the minimization of the correlation between the output sequence samples. This procedure, assuming of the true input signal to be white, did not prove as effective as the variance minimization: this is most probably due to the fact the true input signal is not perfectly white, as reported by some authors [56, 57, 58, 54].

Table 5 reports the mean values and the standard deviations of the results obtained processing both phantom and *in-vivo* images. All the discussed adaptive filters features a positive increase in axial resolution both on *in-vivo* and phantom images, with very similar performance between the two kind of images. Best results in terms of mean axial resolution gain is achieved by the standard LMS algorithm, however the SAF algorithm features the highest ratio between mean and standard deviation. Second in line on the base of both these criteria is the adaptive gain variant of the standard LMS algorithm.

Peak SNR features some significant variants on the *in-vivo* images, ranging from 24 to the almost 30 dB of the NEM algorithm. While this same algorithm achieves the best results on the phantom images too, its fluctuations around the mean value are the biggest among all the tested algorithms.

Phantoms	Gax	PSNR [dB]	QI	CG
LMS	1.81 ± 0.16	21.06 ± 1.47	0.71 ± 0.11	1.76 ± 0.20
LMS-AG	1.39 ± 0.23	21.37 ± 1.30	0.73 ± 0.10	1.65 ± 0.11
LMS-NEM	1.30 ± 0.17	24.30 ± 1.47	0.90 ± 0.03	1.32 ± 0.05
LMS-SAF	1.41 ± 0.21	22.38 ± 1.55	0.79 ± 0.10	1.39 ± 0.06
MS-IIR	1.60 ± 0.30	23.47 ± 1.26	0.87 ± 0.03	1.45 ± 0.10
In-vivo	Gax	PSNR [dB]	QI	CG
LMS	1.93 ± 0.39	24.04 ± 2.00	0.85 ± 0.06	2.84 ± 0.31
LMS-AG	1.81 ± 0.22	24.30 ± 1.93	0.86 ± 0.06	2.04 ± 0.17
LMS-NEM	1.14 ± 0.15	29.64 ± 3.10	0.97 ± 0.02	1.25 ± 0.09
LMS-SAF	1.45 ± 0.17	25.30 ± 2.24	0.88 ± 0.06	1.58 ± 0.13
LMS-IIR	1.46 ± 0.24	27.62 ± 2.11	0.95 ± 0.02	1.65 ± 0.22

Table 5: A performance comparison of the discussed adaptive filters applied on US images: results on phantom images are shown on top, on *in-vivo* on the bottom.

Taking again into consideration the ratio between mean and standard deviation, the best algorithm on both categories is the IIR LMS. Almost the same considerations can be drawn for image quality.

Finally, concerning contrast gain estimations, the best overall results are obtained by the standard LMS algorithm followed by its adaptive gain version. However, these two algorithms feature a rather high deviation from their mean values: the two best algorithms following this criteria are again the NEM and the SAF.

Figure 53 shows a visual comparison of the estimations performed by each adaptive filter on a single US image. From a visual inspection, the tissue structure around coordinates $[10, 20]$ is most clearly reconstructed by the Adaptive Gain LMS and IIR adaptive filters. On the other hand, in terms of noise rejection and border sharpening, the best algorithms is the SAF, most probably due to the adaptive weighting applied to the different bands.

Finally, although featuring the best resolution improvement, the estimation yielded by the standard LMS algorithms is the noisiest and the less contrasted out of the five, justifying the adoption of advanced filtering techniques.

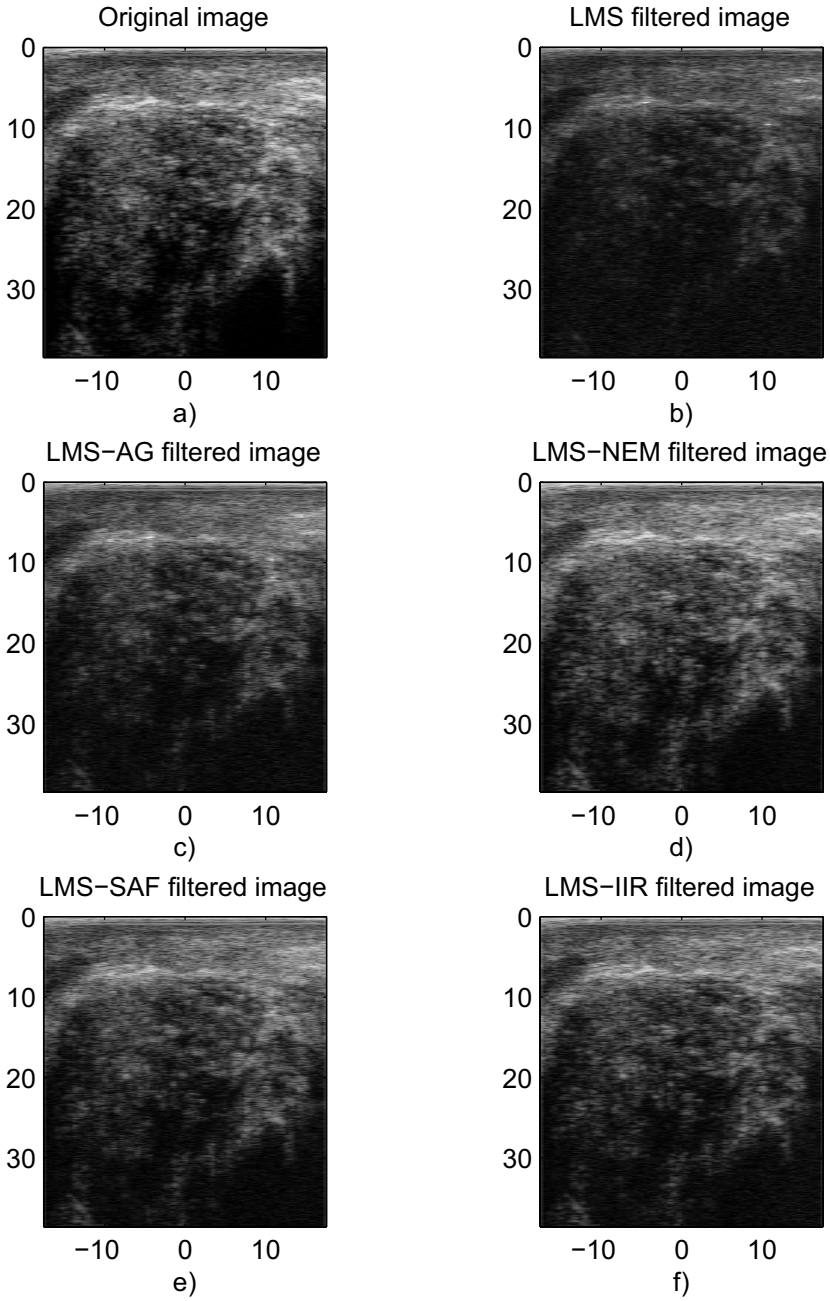


Figure 53: LMS algorithms visual comparison on echographic images deconvolution; from top to bottom: a) Original image, b) LMS output, c) Adaptive Gain LMS output, d) NEM output, e) SAF output, f) IIR LMS output.

7 ADAPTIVE DENOISING

In Section 2.4 the following model for the acquired neural signal was introduced:

$$y[k] = x[k] + (x * f)[k]u[k] + v[k] \tag{7.1}$$

where the term $v[k]$ represents the white noise component and the signal dependent one, $(x * f)[k]u[k]$, is modeled with a direct product between a stationary, uncorrelated noise u with given mean and variance and a filtered version of the clean signal.

This model can be specialized for different kinds of noise simply by adjusting the coefficients of the filter f. In order to fully exploit the inherent model flexibility, an *ad-hoc* denoising algorithm called Wavelet Adaptive Rescaling (WARES) was designed, which depends directly on the filter coefficients, rather than focusing only on some of the many noise models.

The next sections discuss the WARES algorithm starting from consideration based on a time domain approach to the problem; some limitations will then be highlighted, leading to a Wavelet domain approach. Some details are given

about the adaptive rescaling parameter; the discussion is concluded by signal denoising performance comparisons both on synthetic and real-world data.

7.1 EXTRACELLULAR RECORDINGS DENOISING

In order to estimate $x[k]$ from its noisy samples $y[k]$, following [75] a Linear Minimum Mean Square Error (LMMSE) approach can be adopted, resulting in:

$$\hat{\mathbf{x}} = E[\mathbf{x}] + \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1} \cdot (\mathbf{y} - E[\mathbf{y}]) \tag{7.2}$$

with $\mathbf{x} = (x[0], x[1], \ldots, x[k])$ and $\mathbf{y} = (y[0], y[1], \ldots, y[k])$ being the vectors corresponding respectively to the signal to recover and the recorded signal and $\mathbf{C}_{xy}$ and $\mathbf{C}_{yy}$ covariance matrices between the subscripted signals. This means that to be able to denoise the recorded signal, only the expectation of the AP sequence and noise statistics up to the second order must be given.

Expanding the sample indexes in equation (7.2) one obtains

$$\hat{x}[k] = E\big[x[k]\big] + \sum_{pl} \big[\mathbf{C}_{xy}\big]_{kp} \big[\mathbf{C}_{yy}^{-1}\big]_{pl} (y[l] - E\big[y[l]\big]) \tag{7.3}$$

where the subscript ij has been used to indicate the element on the i-th row and j-th column of the corresponding matrix.

If u and v are supposed to be both stationary with zero mean and u, v and x each other independent, it can be shown that

$$E\big[x[k]\big] = E\big[y[k]\big] \tag{7.4}$$

$$\mathbf{C}_{xy}\big] = \mathbf{C}_{xx} \tag{7.5}$$

$$\mathbf{C}_{yy} = \mathbf{C}_{xx} + \mathbf{C}_{zz} \bullet \mathbf{C}_{uu} + \mathbf{C}_{vv} \tag{7.6}$$

where z represent the signal originating from the convolution of x with the filter f and $\bullet$ represents the Hadamard or entrywise matrix product.

These relationships are derived in Section B.1, and link quantities measured on the received noisy signal y with quantities which could be measured on the clean signal x.

Moreover, if u and v are suppose to be both uncorrelated, the matrices $\mathbf{C}_{uu}$ and $\mathbf{C}_{vv}$ are both diagonal. Now, if $\mathbf{C}_{xx}$

is supposed to be diagonal too, $\mathbf{C}_{xy}$ and $\mathbf{C}_{yy}$ both become diagonal, making the product $\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}$ of straightforward calculation. However, this means that the signal itself is supposed to be uncorrelated: it is common opinion that this hypothesis is unrealistic.

7.2 DENOISING IN THE WAVELET DOMAIN

By approaching in the Wavelet domain the problem discussed in the previous section, the uncorrelation hypothesis on the signal can be translated into a less stringent one, involving only its Wavelet coefficients.

Recalling the definition of SWT as

$$\mathrm{SWT}_{m,n}(x) = \sum_{k\in\mathbb{Z}} x[k]\, h_m[k-n] \quad m \in \mathbb{N},\, n \in \mathbb{Z} \tag{7.7}$$

it is possible to substitute the estimation of $x[k]$ in (7.1) with the estimation of its SWT coefficients starting from the SWT coefficients of $y[k]$. Calling m the octave index of the SWT, inspired by [75], the following LMMSE in the Wavelet domain for the estimation of $x_m[k]$ is proposed:

$$\hat{\mathbf{x}}_m = \mathrm{E}[\mathbf{x}_m] + \mathbf{C}_{x_m y_m}\mathbf{C}_{y_m y_m}^{-1} \cdot (\mathbf{y}_m - \mathrm{E}[\mathbf{y}_m]) \tag{7.8}$$

with $\mathbf{x}_m = (x_m[0], x_m[1], \ldots, x_m[k])$ being the vector corresponding to the m-th octave SWT coefficients of the signal to recover, $\mathbf{y}_m = (y_m[0], y_m[1], \ldots, y_m[k])$ the one referring to the recorded signal and $\mathbf{C}_{x_m y_m}$ and $\mathbf{C}_{y_m y_m}$ covariance matrices between the subscripted Wavelet coefficients. In order to denoise the recorded signal, the statistics of both the AP sequence and noise must be known up to the second order in the Wavelet domain.

As well as it was done for the time domain, the sample indexes in equation (7.8) can be expanded, obtaining

$$\hat{x}_m[k] = \mathrm{E}\big[x_m[k]\big] + \sum_{pl} \big[\mathbf{C}_{x_m y_m}\big]_{kp} \cdot \big[\mathbf{C}_{y_m y_m}^{-1}\big]_{pl} (y_m[l] - \mathrm{E}\big[y_m[l]\big]) \tag{7.9}$$

where the symbology is the same of the time-domain case.

Again, if u and v are supposed to be both stationary with zero mean and u, v and x each other independent, it can be shown that

$$\mathrm{E}\left[x_m[k]\right] = \mathrm{E}\left[y_m[k]\right] \tag{7.10}$$

$$\mathbf{C}_{x_m y_m} = \mathbf{C}_{x_m x_m} \tag{7.11}$$

$$\mathbf{C}_{y_m y_m} = \mathbf{C}_{x_m x_m} + \mathcal{C}_m^{(1)} + \mathcal{C}_m^{(2)} \tag{7.12}$$

with

$$\left[\mathcal{C}_m^{(1)}\right]_{pq} = \sum_{ij} H_m[p, q, i, j] \left[\mathbf{C}_{vv}\right]_{ij}$$

$$\left[\mathcal{C}_m^{(2)}\right]_{pq} = \sum_{ijrs} A_m[p, q, i, j, r, s] \left[\mathbf{R}_{xx}\right]_{rs} \left[\mathbf{C}_{uu}\right]_{ij}$$

where $\mathbf{R}_{xx}$ is the autocorrelation matrix of $x[n]$ and

$$H_m[p, q, i, j] = h_m[p - i]\, h_m[q - j]$$
$$A_m[p, q, i, j, r, s] = h_m[p - i]\, h_m[q - j]\, f[i - r]\, f[j - s]$$

These relationships are derived in Section B.2.

Comparing equation (7.12) with (7.6) it is possible to see that in the Wavelet domain it is not so straightforward to understand under which conditions $\mathbf{C}_{y_m y_m}$ becomes diagonal. Exploiting all the previous assumptions and supposing u and v to be also both uncorrelated with standard deviations σ_u and σ_v, $\mathbf{C}_{uu}$ and $\mathbf{C}_{vv}$ can be written as:

$$\left[\mathbf{C}_{uu}\right]_{ij} = \begin{cases} \sigma_u^2, & i = j \\ 0, & i \neq j \end{cases} \qquad \left[\mathbf{C}_{vv}\right]_{ij} = \begin{cases} \sigma_v^2, & i = j \\ 0, & i \neq j \end{cases} \tag{7.13}$$

Using (7.13) and assuming x to be stationary over both the supports of f and h_m, the last two terms of (7.12) can be simplified, highlighting their underlaying structure:

$$\left[\mathcal{C}_m^{(1)}\right]_{pq} = \sigma_v^2\, R_{h_m h_m}[p - q] \tag{7.14}$$

$$\left[\mathcal{C}_m^{(2)}\right]_{pq} = \sigma_u^2\, R_{h_m h_m}[p - q] \sum_d R_{ff}[d]\, R_{xx}^{(q)}[d] \tag{7.15}$$

where $R_{h_m h_m}[\,\cdot\,]$ is the value assumed by the autocorrelation function of the filter h_m at the corresponding lag, $R_{ff}[\,\cdot\,]$ is its equivalent for the filter f and $R_{xx}^{(q)}[\,\cdot\,]$ the same

for the signal x but this time centered on the sample q. More details are given in Section B.2.

If the octave index m is not too large, the support $S_m^{(h)}$ of h_m is sufficiently small so that the autocorrelation function of h_m can be well approximated by a delta function, x can be supposed stationary over $S_m^{(h)}$ and the matrix $\mathbf{C}_{x_m x_m}$ is almost diagonal.

This is commonly retained to be much more realistic than supposing x uncorrelated. Moreover, it can be read as a requirement for x to carry its most significant frequency components within a sufficiently narrow support in the frequency domain, which is verified by the AP waveform.

Thanks to these hypothesis, $R_{h_m h_m}[p-q] = \gamma_m \delta[p-q]$ holds, thus both $\mathbf{C}_{x_m y_m}$ and $\mathbf{C}_{y_m y_m}$ become diagonal, leading to

$$\hat{x}_m[k] = E\big[y_m[k]\big] + \frac{\big[\mathbf{C}_{x_m y_m}\big]_{kk}}{\big[\mathbf{C}_{y_m y_m}\big]_{kk}} \left(y_m[k] - E\big[y_m[k]\big]\right) \quad (7.16)$$

and, since the filters h_m are never low-pass, the continuous component of y gets annihilated, resulting into the following equality

$$\hat{x}_m[k] = \frac{\big[\mathbf{C}_{x_m y_m}\big]_{kk}}{\big[\mathbf{C}_{y_m y_m}\big]_{kk}} y_m[k] = \lambda_m[k]\, y_m[k] \quad (7.17)$$

which clearly represents a rescaling approach to the denoising problem in the Wavelet domain with rescaling factor $\lambda_m[k]$.

7.3 EXPANDING THE RESCALING PARAMETER

The resulting denoising algorithm, called WARES, is made adaptive by designing the estimation of the rescaling parameter $\lambda_m[k]$ for each sample k and each octave m to be driven by the input signal y. Moreover, as this involves statistics taken from all the octaves in the SWT, performance are intrinsically tied to the Wavelet Transform and influenced by the choice of the Wavelet base. Finally, it is possible to calculate both upper and lower constraints for the value of the rescaling factor, leading to an always stable, non-linear denoising algorithm.

Plugging (7.10–7.12) and (7.14–7.15) in (7.17), the following relationship for the rescaling factor are obtained

$$\lambda_m[k] = 1 - \frac{R_{h_m h_m}[0]}{\left[\mathbf{C}_{y_m y_m}\right]_{kk}} \left(\sigma_v^2 + \sigma_u^2 \sum_d R_{ff}[d]\, R_{xx}^{(k)}[d]\right) \quad (7.18)$$

where only the quantity $R_{xx}^{(k)}[d]$ is unknown. To estimate this term, the following relationship can be exploited:

$$\begin{aligned}\left[\mathbf{C}_{x_m x_m}\right]_{kk} &= E\left[(x_m[k] - E[x_m[k]])^2\right] \\ &= E[x_m[k]^2] - E[x_m[k]]^2 \end{aligned} \quad (7.19)$$

where, thanks to the fact that filters h_m are never low-pass, the last term is zero, thus leaving only $E[x_m[k]^2]$. Now, by expanding the SWT filters and exploiting the linearity of the expectation operator, the following holds

$$\left[\mathbf{C}_{x_m x_m}\right]_{kk} = \sum_{ij} h_m[i]\, h_m[j]\, E[x[k-i]\, x[k-j]] \quad (7.20)$$

where the stationarity of x over the Wavelet filters' supports can be exploited to adjust the summation indexes as follows

$$\left[\mathbf{C}_{x_m x_m}\right]_{kk} = \sum_d E[x[k]\, x[k+d]] \sum_j h_m[j+d]\, h_m[j] \quad (7.21)$$

which results in

$$\left[\mathbf{C}_{x_m x_m}\right]_{kk} = \sum_d R_{xx}^{(k)}[d]\, R_{h_m h_m}[d] \quad (7.22)$$

Equipped with this relationship, (7.12) can be used to estimate $R_{xx}^{(k)}[d]$: in fact, rearranging its terms it is possible to write

$$\begin{aligned}\left[\mathbf{C}_{y_m y_m}\right]_{kk} - \sigma_v^2\, R_{h_m h_m}[0] &= \sum_d R_{xx}^{(k)}[d] \\ &\cdot \left(R_{h_m h_m}[d] + \sigma_u^2\, R_{h_m h_m}[0]\, R_{ff}[d]\right)\end{aligned} \quad (7.23)$$

where one can substitute $\left[\mathbf{C}_{y_m y_m}\right]_{kk}$ with $\sigma_{y_m}^2[k]$ and $R_{h_m h_m}[0]$ with 2^{-m} because of the properties of the Wavelet filters [38].

Note that the term between parenthesis does not depend on the sample index k.

By letting

$$s_m[k] = \left[\mathbf{C}_{y_m y_m}\right]_{kk} - \sigma_v^2 R_{h_m h_m}[0] \tag{7.24}$$

$$r_d[k] = R_{xx}^{(k)}[d] \tag{7.25}$$

$$\left[\mathbf{A}\right]_{md} = R_{h_m h_m}[d] + \sigma_u^2 R_{h_m h_m}[0] R_{ff}[d] \tag{7.26}$$

equation (7.23) can be rewritten as

$$s_m[k] = \sum_d \left[\mathbf{A}\right]_{md} r_d[k] \tag{7.27}$$

Then, $R_{xx}^{(k)}[d]$ is estimated by using the pseudoinverse of $\mathbf{A}$ as this matrix is rectangular in most cases

$$R_{xx}^{(k)}[d] \simeq \sum_m \left[\mathbf{A}^\dagger\right]_{dm} \left(\sigma_{y_m}^2[k] - 2^{-m}\,\sigma_v^2\right) \tag{7.28}$$

Equality (7.18) can be split into two terms as follows

$$\lambda_m[k] = \frac{\sigma_{y_m}^2[k] - 2^{-m}\,\sigma_v^2}{\sigma_{y_m}^2[k]} - \sum_{dp} \left(2^{-m}\,\sigma_u^2\,R_{ff}[d]\right) \cdot \left[\mathbf{A}^\dagger\right]_{d\mu} \frac{\sigma_{y_\mu}^2[m] - 2^{-\mu}\,\sigma_v^2}{\sigma_{y_m}^2[k]} \tag{7.29}$$

As one can notice, the first and the last term in (7.29) have the same structure; moreover, the double summation can be thought as a matrix product between three terms. By doing the following positions

$$\left[\mathbf{U}\right]_{md} = 2^{-m}\,\sigma_u^2\,R_{ff}[d] \tag{7.30}$$

$$\left[\mathbf{V}[k]\right]_{\mu m} = \frac{\sigma_{y_\mu}^2[m] - 2^{-\mu}\,\sigma_v^2}{\sigma_{y_m}^2[k]} \tag{7.31}$$

equation (7.29) simply becomes

$$\lambda_m[k] = \left[\mathbf{V}[k]\right]_{mm} - \sum_{d\mu} \left[\mathbf{U}\right]_{md} \left[\mathbf{A}^\dagger\right]_{d\mu} \left[\mathbf{V}[k]\right]_{\mu m} \tag{7.32}$$

where the right hand side term clearly is the diagonal of $\mathbf{V}[k] - \mathbf{U}\,\mathbf{A}^\dagger\,\mathbf{V}[k]$. Calling *diag* the operator which extracts

the main diagonal from a matrix, the next relationship holds:

$$\lambda_m[k] = \left[\mathrm{diag}((\mathbf{I} - \mathbf{U}\,\mathbf{A}^\dagger)\,\mathbf{V}[k])\right]_m \tag{7.33}$$

where $\mathbf{V}$, the only term that depends on the sample index m, was separated from the other terms which do not depend on it and thus can be precalculated in order to reduce the computational cost. The term $\mathbf{\Lambda} = \mathbf{I} - \mathbf{U}\,\mathbf{A}^\dagger$ is called *rescaling matrix* .

Finally, it is possible to show that $\lambda_m[k]$ is both upper and lower bounded:

$$0 \leqslant \lambda_m[k] \leqslant 1 \tag{7.34}$$

Concerning the lower bound, $\lambda_m[k]$ represents a ratio between positive quantities, respectively the local variance of signals x and y, so it must always be positive. Moreover, under the hypotheses made, it's easy to verify that the variance of x cannot be greater than the variance of y. Following the definition, the next relationship holds

$$\begin{aligned}\sigma_y^2 &= E[(y - E[y])^2] \\ &= E[(x + (x * f)\,u + v - E[x])^2] \\ &= \sigma_x^2 + 2\,E[(x - E[x])((x * f)\,u + v)] + E[((x * f)\,u + v)^2]\end{aligned}$$

where the second term in the last equation is zero since u and v are supposed to be stationary with zero mean and independent of x. So the following is obtained

$$\sigma_x^2 = \sigma_y^2 - E[((x * f)\,u + v)^2]$$

showing that σ_x^2 must be lower than σ_y^2 since the right hand term is the expectation of positive quantities.

Since $\lambda_m[k]$ was estimated through a MSE minimization, it is possible to obtain values outside the theoretical bounds. Whenever this happens, the corresponding element of $\lambda_m[k]$ should be trimmed to the nearest boundary of the interval of feasible values.

The estimation of the rescaling factor can be now summarized into these steps:

1. Estimation of the variance of the Wavelet coefficients of y around sample m;

2. Generation of the variance ratios matrix $\mathbf{V}$;
3. Matrix product between the rescaling matrix $\boldsymbol{\Lambda}$ and $\mathbf{V}$ and extraction of the principal diagonal of the result;
4. Trimming of the resulting vector to the interval $[0, 1]$

As anticipated at the beginning of this section, this procedure is clearly both data driven and non-linear. Moreover, since a matrix product is involved, the value of the rescaling factor of each octave depends on the whole set of Wavelet coefficient and not on a subset nor a single coefficient.

Finally, it is reasonable remark that the computational complexity of this approach is fairy small, as Wavelet coefficients variance can be easily computed on-line using an appropriate non linear filter, and matrix multiplication followed by diagonal extraction can be substituted by a series of vector inner product. In numerical terms, if N is the number of octaves in the Wavelet Transform, only N variances, N^2 divisions, N^2 multiplications and $O(N^2)$ linear operations must be calculated at any given time-step.

7.4 SYNTHETIC SIGNAL DENOISING

In order to quantitatively evaluate denoising performance of the WARES algorithm when applied to ER signals, tests on synthetically generated signals with decreasing SNR where preformed, comparing WARES to other Wavelet-based algorithms [28].

Each spike is described as the superposition of 2 or more functions in the form

$$f(t, \mathbf{p}) = \begin{cases} p_0 \left(\frac{t}{p_1} \exp\left(1 - \frac{t}{p_1}\right) \right)^{p_2}, & t > 0 \\ 0, & t \leqslant 0 \end{cases} \tag{7.35}$$

where $\mathbf{p} = (p_o, p_1, p_2)$ is the set of parameters defining the shape of the function: p_0 sets the maximum amplitude, which is reached at $x = p_1$, while p_2 controls how fast the function vanishes in 0 and toward $+\infty$. As a result, an AP is defined as follows

$$s(t) = \sum_{i=0}^{N_c} f(t - \theta_i, \mathbf{p}_i) \tag{7.36}$$

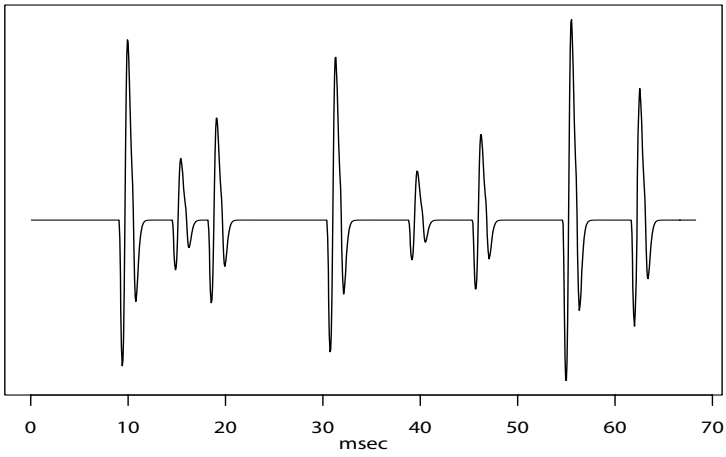


Figure 54: Example of synthetic spike train generated using formulas (7.35-7.37).

where θ_i is a set of time displacements and $\mathbf{p}_i$ is a set of valid parameters for the function (7.35).

Finally, to generate a non-overlapping spike train, a sequence of amplitude scaled and translated versions of function (7.36) is computed

$$x(t) = \sum_{i=0}^{N_s} A_i \, s(t - \tau_i) \tag{7.37}$$

where A_i is the set of scalar amplitudes and τ_i the set of time displacements.

An example of spike train generated in this way is showed in Figure 54: by adding a proper amount of noise, very realistic test sequences can be obtained, as it can be seen in Figure 55.

The synthetic signal in Figure 54 was used with different levels of noise to assess WARES performance against the seven best Wavelet denoising algorithms chosen from [28]. From now on, these algorithms will be called *competitors* for the sake of briefness.

The SNR for the additive component was swept over the range $[-10, 30]$ dB with a step of 5 dB, at different SNR for the multiplicative component. For each value of the input SNR, 1000 noisy sequence were generated and processed using each of the chosen denoising algorithms; the output SNR was calculated comparing the denoised signal to the

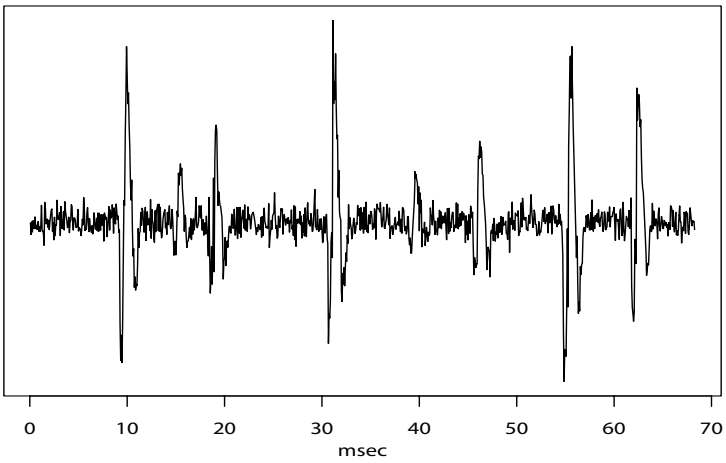


Figure 55: A noisy version of the synthetic spike train shown in Figure 54.

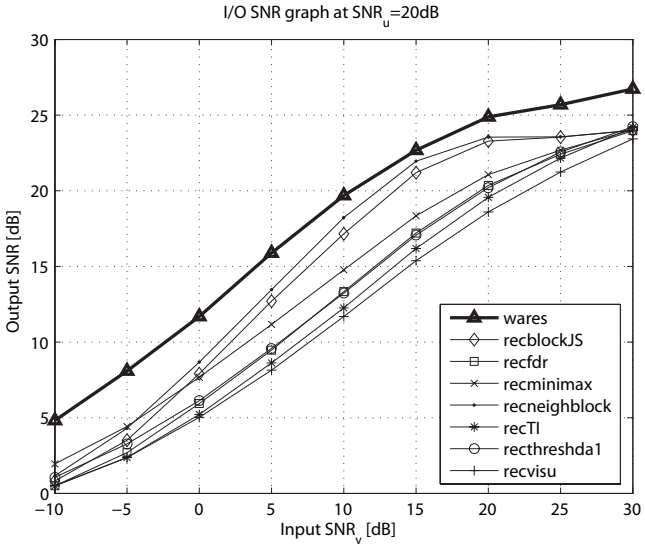


Figure 56: Input/Output SNR performance chart for different Wavelet-based denoising techniques in presence of low multiplicative noise.

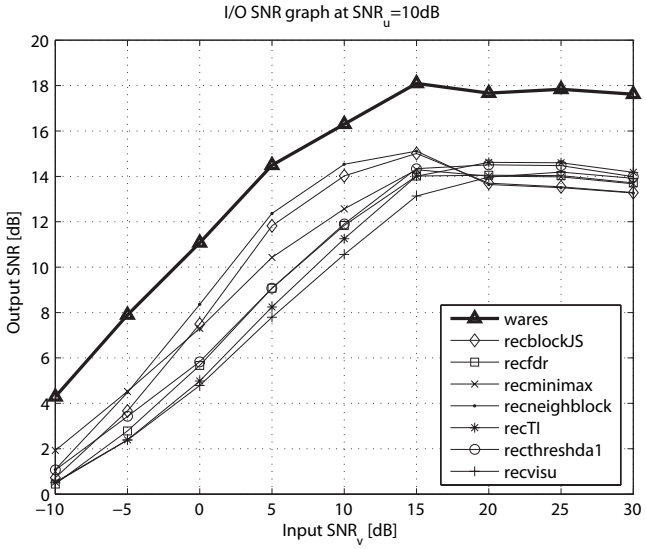


Figure 57: Input/Output SNR performance chart for different Wavelet-based denoising techniques in presence of medium multiplicative noise.

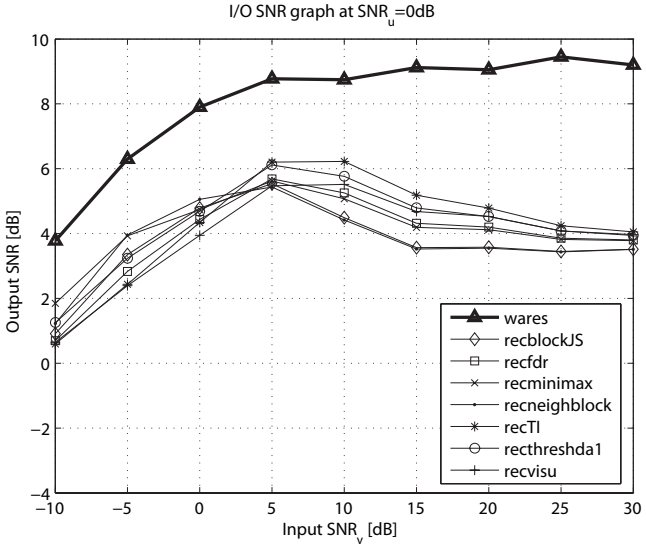


Figure 58: Input/Output SNR performance chart for different Wavelet-based denoising techniques in presence of high multiplicative noise.

original clean signal. The curves shown in figures 56-58 display the output SNR as a function of the input SNR for WARES and its competitors.

The results show that for every test point at least 1 dB is gained in terms of output SNR, this gain being more evident in presence of high multiplicative noise (Figure 58), where WARES performs almost 5 dB than its competitors.

In fact, in presence of low multiplicative noise (Figure 56) the difference between the model (7.1) and a standard additive white noise model is not enough to produce a significant separation between WARES and its competitors.

However, starting from intermediate levels of multiplicative noise (Figure 57) an asymptotical behavior in the performance of the tested algorithms appears: under these conditions, WARES performs 2 dB better than any competitor, gaining almost 4 dB in the high SNR region where signal model (7.1) is more accurate than the pure additive model.

Finally, under high levels of multiplicative noise, WARES is much more effective than any other Wavelet based algorithms, providing an evident improvement in performance over all the SNR tested: here the multiplicative component is strong enough to make the additive noise model to simple for the simulated data.

7.5 REAL DATA DENOISING

It is believed that performance evaluation of denoising algorithms on real data requires more than a numerical assessment of the quality of the denoised data in terms of SNR, a visual inspection of them being much more preferable.

WARES denoising was performed on three sets of data, all of them courtesy of Dr. Christophe Pouzat, Laboratoire de Physiologie Cérébrale, UFR biomédicale de l'Université René Descartes (Paris V). The first one, called *locust*, is a tetrode recording of 20 s from the locust (*Schistocerca americana*) antennal lobe, sampled at 15 kHz and digitally filtered between 300 Hz and 5 kHz, in order to eliminate LFP and avoid aliasing. The second and the third ones, called *rampe15s* and *rampe10s*, are both electromyographic recording of 15 s and 10 s respectively.

For each of these three datasets a comparison is presented

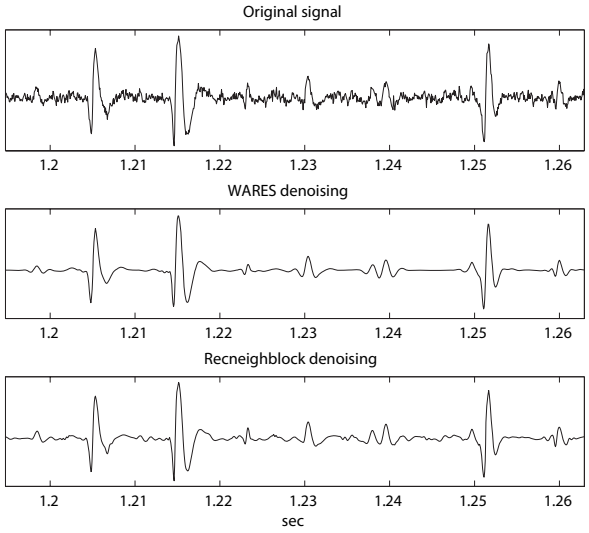


Figure 59: Comparison of the original signal and its denoised versions for a sample of the *locust* dataset containing spikes with different amplitude.

between 70 ms of the original signal (top track in each of Figures 59-61) and its denoised version (center and bottom tracks) obtained respectively by applying the WARES and the *recneighblock* algorithms to the original signals. The *recneighblock* algorithm was chosen because all three datasets have an intermediate level of multiplicative noise and this is the second best algorithm out of WARES in processing this kind of data, according to Figure 57.

The *locust* dataset (Figure 59) is the noisiest of the three, but spikes with very different amplitudes can still be easily identified; the presence of noise motivated the choice to apply a denoising method. The overall result appears quite good in terms of both smoothness and noise rejection, but some slight alterations in spikes amplitude can be highlighted. The region between two adjacent spikes tends to be much less noisier when compared to the signal denoised by the competitor algorithm. The absence of LFP is evident in both the original and the denoised signals.

The *rampe15s* dataset (Figure 60) has very little noise, but the presence of a small amount of LFP alters the data introducing a low frequency component. Since WARES makes use only of the WT detail coefficients, it can remove the

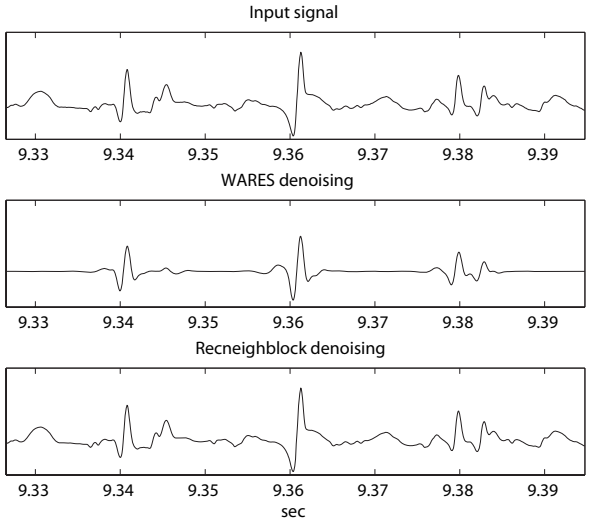


Figure 60: Comparison of the original signal and its denoised versions for a sample of the *rampe15s* dataset containing both spikes and LFP.

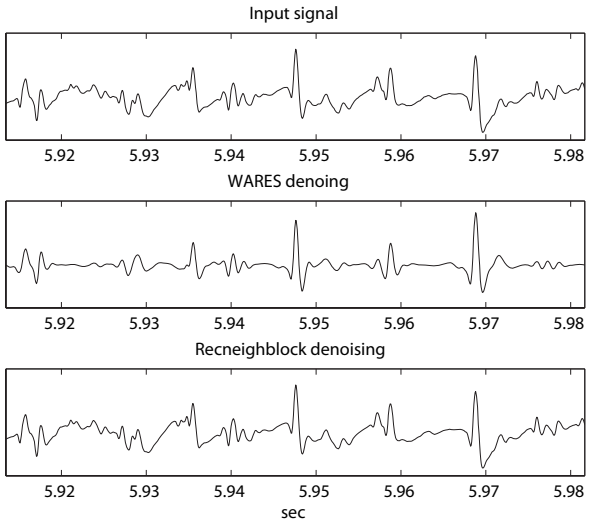


Figure 61: Comparison of the original signal and its denoised versions for a sample of the *rampe10s* dataset containing both spikes and LFP.

LFP much better than its competitors, simply by changing the number of scales the WT is performed on. This is quite evident in the results shown, where the denoised spikes resemble much more closely to the well-known waveform than the noisy ones.

Finally, the *rampe10s* dataset (Figure 61) has again very little noise, but the LFP presence is much more evident and alters the data much more seriously than the *rampe10s* dataset. Again, after a good tuning of the number of WT scales, WARES can remove the LFP much better than its competitors and efficiently recovers the spikes waveforms. However, since the LFP component is much stronger than the previous case, some leakage of this low-pass component into the denoised signal is present.

Part III

APPLICATIONS

8

BIOLOGICAL TISSUE CLASSIFICATION

As already discussed in Chapter 1, ultrasounds are a cost effective, mobile, noninvasive, harmless, and sufficiently accurate imaging technique. In Section 1.1 the problems involved into the acquisition and processing of ultrasound signals were introduced: the main drawback is that signal resolution is limited by finite bandwidth of the transducer and by non-negligible width of the transmitted acoustic beams.

Image restoration techniques, such as deconvolution techniques, can be employed to improve US images resolution and their diagnostic significance. Generally speaking, as the PSF is a band-limited function and due to the presence of noise, true image estimation is an ill-posed problem.

To obtain a stable algorithm which provides a unique solution, additional constraints must be imposed. Designing a method which exhibits the most appropriate compromise among computational complexity, reliability and portability for biomedical real-time imaging applications is still an open challenge.

Moreover, signal processing after image restorations must

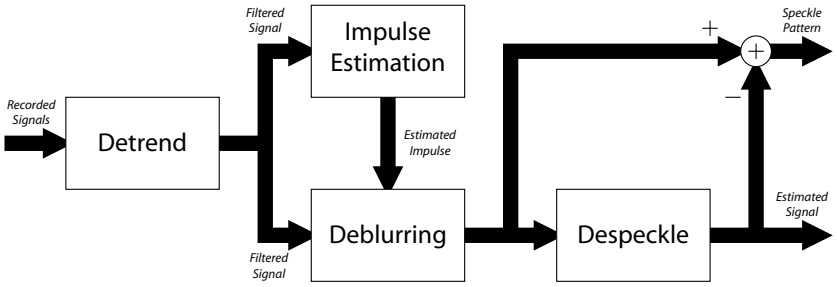


Figure 62: Schematic of the proposed US signal processing algorithm: pulse estimation and deconvolution may be jointly substituted by adaptive filtering.

deal with a noise peculiar of this kind imaging systems: speckle noise. This image degradation effect appears as a granular texture in different shades of gray and increases the difficulty of human interpretation in discriminating fine details of images.

The source of this noise is attributed to random interference between the returns issued from numerous subresolution scatterers present on a surface, so it is the result of tissue ultrasound interaction, and not of noise originating at some external source. Thus, even if speckle degrades the visual quality of US images, on the other hand it can provide useful information on many characteristics of the scatterer structure: for this reason, de-speckling procedures are important for both enhancing US images and extracting speckle patterns for automatic classification of the investigated tissues.

8.1 ECHOGRAPHIC SIGNAL PROCESSING

Echographic signal processing is usually designed as a multistep process, as shown in Figure 62. As introduced in Section 1.1, the first step is the removal of the Deterministic Trends (DT) in the US received echo.

These trends are present both in phantom and in-vivo images, depending on the US recording equipment and have a piecewise polynomial behavior, as shown in Figure 63. Experimental measurements showed that negligible differences are present between DT belonging to adjacent vertical scan lines and that no information related to the tissue is

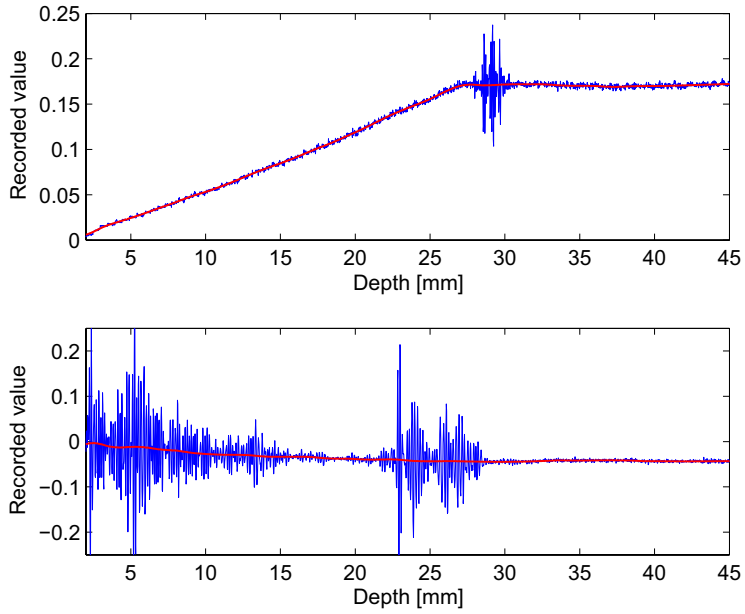


Figure 63: Piecewise polynomial trends in the received ultrasound signals: their presence is relevant both in phantom (top) and in-vivo (bottom) images.

present.

Removal of DT is took to be mandatory for image restoration and quality improvement as they may alter the US echo in significant ways. As shown in Figure 64, the effect of DT removal is much more evident in the lower portion of the US image, where cumulative frequency dependent attenuation is responsible for contrast reduction and resolution degradation.

DT removal can be performed by means of proper order MA filtering using, for example, a specifically tuned adaptive filter from 6. As an alternative, jointly DT and noise removal can be performed using the adaptive denoising algorithm discussed in Chapter 7: in fact, DT can be seen as the equivalent of LFP in the neural environment.

Once DT removal has been performed, signal deconvolution can be carried on. Two approaches are followed above all to face with US image deconvolution.

The first incorporates the PSF identification procedure

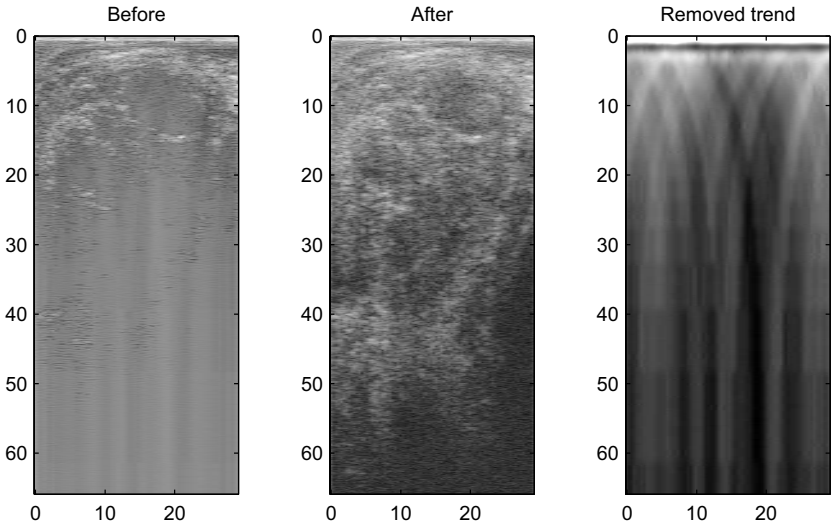


Figure 64: A comparison of the incoming US image with and without trend removal: the lower part of the elaborated image feature much more contrasted details.

within the deconvolution algorithm. This usually lead to the development of computationally heavy algorithms, commonly retained as far from satisfying the real-time signal processing constraints required for US biomedical investigation. Adaptive algorithms discussed in Chapter 6 are some notable exceptions within this category.

In the second approach, PSF and true image estimation are two disjoint procedures. Within this context, these procedures can be implemented by relatively simple algorithms, often suitable for real-time implementation, as shown in Chapter 4 and in Chapter 5.

Figure 62 feature a block structure which follows this last policy: in fact, after DT removal, US signal is fed in parallel to both an impulse response estimator and to a deconvolution algorithm. As the former updates the estimate of the US acquisition equipment impulse response, the latter is responsible for the true deconvolution step. The link between these units allows the system to adapt continuously to the different equipment working conditions.

Finally, more advanced US deconvolution technique can be used: Viterbi algorithm decoding [76, 77] can be supported by adaptive filtering and dynamic alphabet estimations in order to improve its time tracking capabilities. The

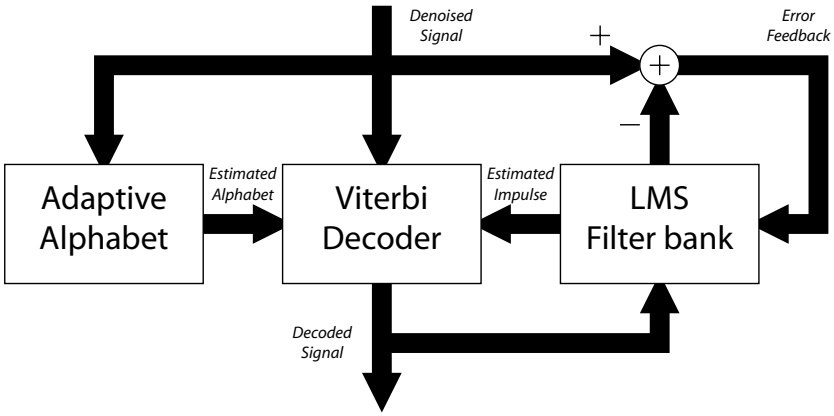


Figure 65: Advanced deconvolution algorithm based on Viterbi decoding and adaptive filtering: Viterbi time tracking capabilities are enhanced by means of adaptive techniques.

configuration depicted in Figure 65 originate from the parallel that can be made between the US acquisition system and a generic transmission channel: from this standpoint, the transducer PSF represents the ISI introduced by the channel.

Here the LMS adaptive filter bank is used to feed the Viterbi decoder with the estimated transducer PSF rather than to perform signal deconvolution. Assuming that the filters in this section are capable of efficiently tracking the time variations of the PSF, Viterbi algorithm can recover the true US signal very efficiently.

However, due to its computational complexity, a simplification in the trellis diagram is mandatory in order to achieve real-time signal processing. Besides this structural optimization, dynamic adaptation of the decoding alphabet is deemed fundamental to aid in shrinking the research space required for the estimation of the optimal output sequence.

Table 6 shows how deconvolution based on Viterbi decoding positively confronts with the other deconvolution techniques presented in Chapter 5. Although ADID is the best at enhancing image contrast, the best results with respect to axial resolution gain, peak SNR and quality index are obtained applying Viterbi decoding.

	Gax	PSNR [dB]	QI	CG
Viterbi	11.00	30.50	0.94	3.96
ADID	8.73	25.45	0.91	4.43
WLSD	3.38	26.63	0.93	3.22
FWRD	1.64	26.54	0.93	0.83

Table 6: Viterbi algorithm compared to the deconvolution algorithms of Chapter 5: positive results are obtained on *in-vivo* images.

Finally, the last preprocessing step suggested by Figure 62 is the separation of the speckle component. The point scatterer model for the RF echo introduced in Chapter 1 is adequate when weak scattering, narrow ultrasound beam and linear propagation hypothesis are fulfilled; instead, for real US images, more sophisticated models were proposed [78].

Since signals of interest are not stationary, the speckle noise term must be assumed non-stationary as well and a time-frequency projection is supposed to be suitable for the analysis. Additionally, as the envelope-detected signal can be modeled by a Rayleigh distribution with mean proportional to the standard deviation, speckle could be modeled as multiplicative noise [79].

Because of this, it is possible to separate this component by means of the adaptive algorithms discussed on Chapter 7: in fact, substituting the filter f in (7.1) with a Kroneker delta sequence, a multiplicative noise model is obtained. Being both capable of dealing with non-stationary signals and multiplicative noise, the WARES algorithm is among the best suited algorithms to perform speckle detection and separation from the true RF echo within the incoming US signal.

An example of the results of these preprocessing steps is shown in Figure 66: here, the received echo from a prostatic gland affected by carcinoma is elaborated following the scheme depicted in Figure 62. From the raw image on the top, to the representations of coherent and speckle components on the bottom, it is possible to evaluate how each step in the preprocessing scheme operates on the

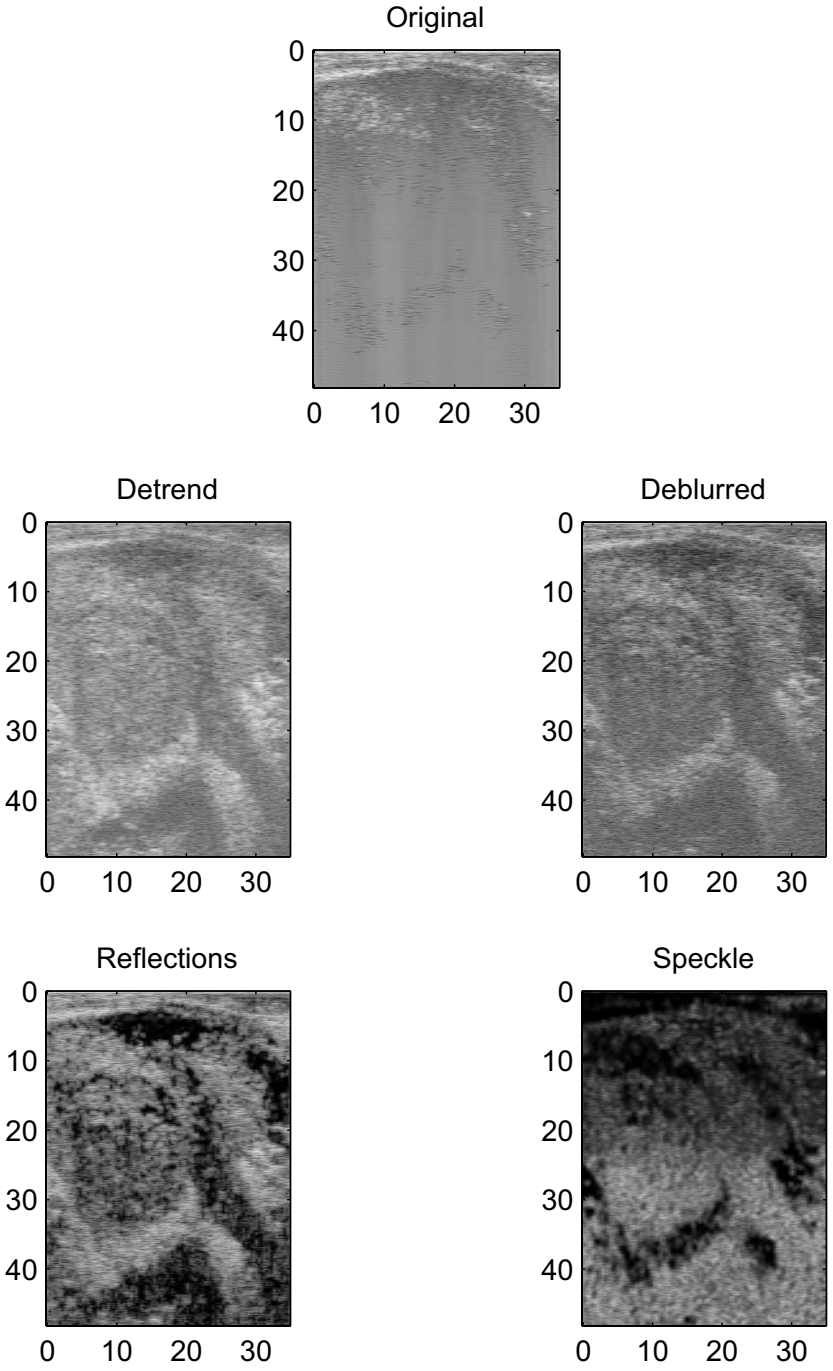


Figure 66: Visual comparison of several echo preprocessing steps: from top to bottom, the raw RF image of an *in-vivo* prostatic gland, the same for detrending and deblurring, and finally two representations of coherent and speckle components.

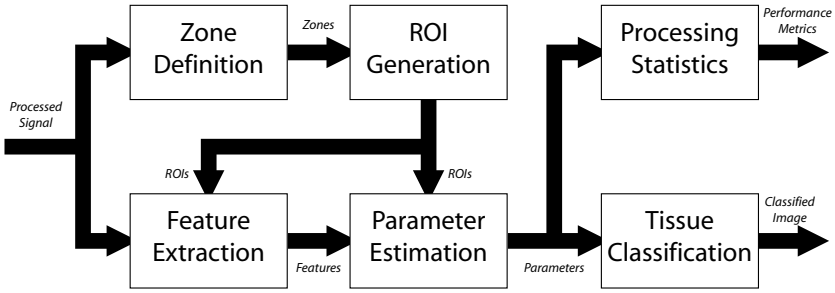


Figure 67: Schematic of the proposed US tissue classification algorithm: upper blocks are used only during the training phase to test for the effectiveness of the chosen features and parameters.

data, enhancing contrast, improving axial resolution and providing a better separation between tissue kinds.

8.2 ULTRASOUND IMAGE CLASSIFICATION

After being preprocessed, signal is ready for being classified. For this purpose, a modular algorithm has been designed, separating conceptually different procedures in six blocks, as shown in Figure 67. The three upper blocks are used only during algorithm training, while the remaining three constitutes the actual classification path.

Within the *zone definition* block, large and usually unstructured regions are identified within the RF received echo image, representing homogeneous portion of the image and corresponding to the different kinds of tissue the system is going to be trained to recognize and separate. This identification process is usually performed by some skilled user, capable of identifying the target tissues by visually inspecting the US image, as shown on the left side of Figure 68. Up to a certain level, this task can be automatized employing segmentation algorithms initialized by expectation maximization procedures.

This step is meant for generating the groundtruth to be compared with the classification results in order to estimate the Receiver Operating Characteristic (ROC) curve of the current algorithm setup. It is useful for estimating other performance metrics too: knowing the response of the "oracle"

about a given region of tissue, it is possible to estimate how good at discriminating between tissues a given combination of feature and parameter is. Features and parameter will be discussed later on in this section.

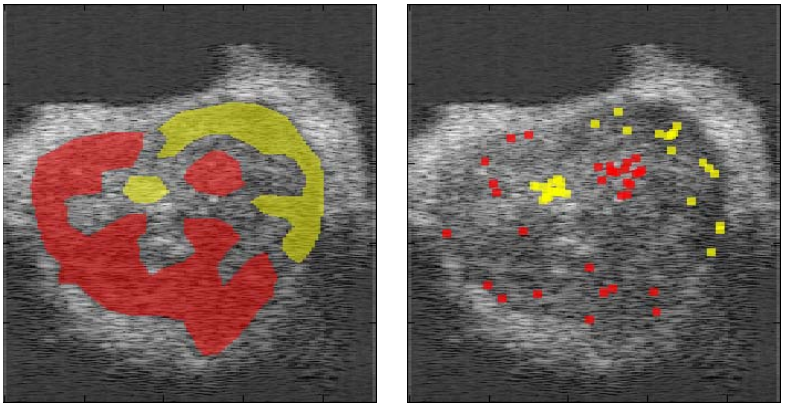
During the training phase, a list of Region of Interest (ROI) is generated too. These regions are small patch of US image, usually each other overlapped, which constitutes the smallest portion of image the classifier is going to work on. Although the classifier is capable to work on elements smaller than a ROI, these patches were introduced to add robustness to tissue analysis.

In fact, processing the data originating from a clinical database comprising of more than 250 malignant and 150 benignant US images highlighted the fact that information relevant for tissue classification can be extracted not only from the punctual value of the estimated features but from the moments of the local value distribution too. Because of this, within each previously identified zone, ROI representing square portions of the tissue under examination are generated and then fed to both feature extraction and parameter estimation blocks.

In order to increase the number of points on which parameter computation is going to be performed, larger and larger ROI can be generated, with the upper limit of one ROI coinciding to its parent zone. On the other hand, to keep the balance between the ROI population within the different zones, random generation of ROI is supported: with this choice, the same number of ROI is instantiated at random locations within each region, as shown on the right side of Figure 68.

Concerning the *feature extraction* block, it is dedicated to performing topology preserving operations, allowing the algorithm to access local information regarding the currently examined tissue in domains different from the starting one. Estimations of Fourier or Wavelet signal components are usually performed in this block, as well as AR, MA or ARMA parameters.

The distinction between coherent and diffuse components, as well as the separation between speckle and signal components discussed in the previous section, can be seen



(a) Selected zones (b) Random generated ROIs

Figure 68: Selected zones and the corresponding set of ROI for a US image of a prostate affected by carcinoma: different colors highlight different tissues.

as particular kinds of feature extractions, and are usually integrated within this block. This is due to the fact that, as discussed in the introduction of this chapter, the diffuse and the speckle components usually contain meaningful information regarding tissue structure and are thus retained significant from the standpoint of tissue classification.

Features are usually classified into two different categories: *textural* features and *parametric* features. The first group is composed by features that examine and numerically evaluate aspects related to image topology and geometrical structures, like repeating patterns, reflector clusters and macroscopic edges. The second group collects features which are much more strictly related to the incoming signal model, locally estimating its parameters, like its central frequency, attenuation coefficient, power within a given set of sub-bands and so on.

The former feature set is intrinsically connected to bidimensional information and investigate the signal at a larger scale, examining somewhat larger chunks of image than the latter features, resulting into coarser resolution estimates. Vice versa, parametric features describe the tissue at a lower scale, focusing on details which are observed at a lower scale and directly on the received signal, during its acquisition.

It is usually preferable to first extract the information con-

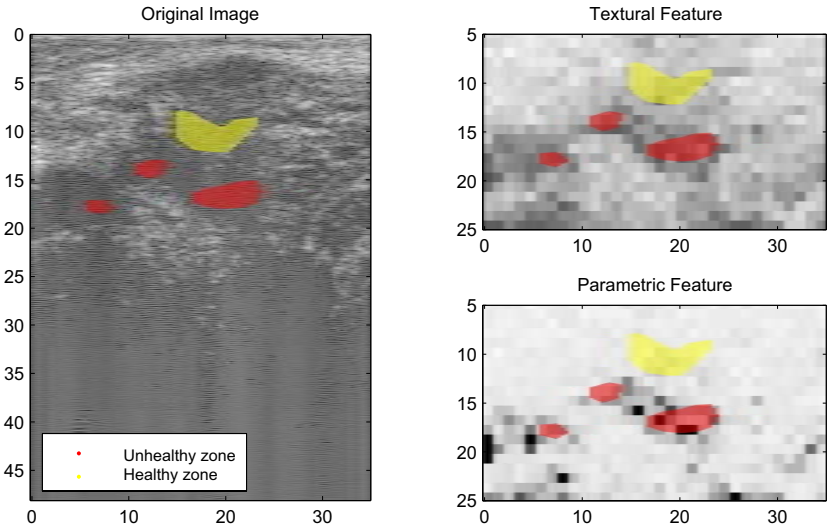


Figure 69: Visual comparison of textural and parametric feature: on the left, a prostatic gland affected by carcinoma (red spots); on the right, two different visualization of the same image, one obtained from a textural analysis (top), the second from a parametric one (bottom).

tained within the raw RF signal by computing parametric features, then convert the received US echo into an image and estimate the chosen textural features on it, rather than to pick one feature kind. Although this increases the computational cost of classification algorithms, the return in terms of augmented accuracy pays off in full. In fact, while the first feature set somehow mimics the diagnostic skill of an expert physician, the second allow the classification algorithm to examine and exploit information which is not perceived by a visual inspection.

Figure 69 shows an example of low-resolution textural and parametric features compared to the original US image of a prostatic gland affected by carcinoma. Groundtruth originating from subsequent *in-vivo* gland biopsy is highlighted by red and yellow spots, both on the original image and on the derived features. It is possible to observe how both the features allow to separate different kind of tissues: by combining the information from each feature, it is possible to achieve tissue classification with a satisfying accuracy.

After features have been generated, parameters are estimated over each ROI. As discussed in the previous paragraphs, estimations about the statistical distribution of feature values over each ROI are performed, allowing the classification engine to access local information concerning the second and higher moments.

As a results, for each ROI, a set of parameter is generated. Concatenating the parameter sets resulting from each feature, a feature vector $\bar{x}$ with cardinality equal to the product between the number of selected features and parameters is generated. Exploiting the output of the zone definition block, each ROI, thus each feature vector, can be tagged as belonging to a given zone.

During the training phase, this information is used by the final blocks for two different purposes: while the tissue classification block trains itself to discriminate between feature vectors belonging to the different classes, the processing statistics block numerically quantify the goodness of a given feature-parameter combination $\bar{x}_k$ at discriminating between ROI belonging a given pair of zones z_i and z_j.

Two different parameters are estimated by this final block:

1. the Fisher Discriminant Ratio (FDR) [80], defined as the ratio
$$d_{(i,j,k)} = \frac{(\mu_{i,k} - \mu_{j,k})^2}{\sigma_{i,k}^2 + \sigma_{j,k}^2}$$
where values $\mu_{i,k}$ and $\mu_{j,k}$ are the mean value of the $\bar{x}_k$ estimated respectively over zone z_i and z_j, and $\sigma_{i,k}$ and $\sigma_{j,k}$ are the corresponding standard deviations;

2. the histogram neg-superposition or *divergence* [80], defined as
$$\bar{s}_{i,j,k} = \int_{-\infty}^{+\infty} (p(\bar{x}_k|z_i) - p(\bar{x}_k|z_j) \ln \frac{p(\bar{x}_k|z_i)}{p(\bar{x}_k|z_j)} d\bar{x}_k$$
where $p(\bar{x}_k|z_i)$ is the probability of the value $\bar{x}_k$ to be present within the i-th zone.

As the cardinality of the space $\bar{x}$ belongs to may be very high, non-linear, kernel based, feature extraction [81] is performed on data: feature-parameter combinations ranking

	Training [ms]	Testing [ms]	ROC area	FoM $\left[\frac{\text{ROC area}}{\text{Test time}}\right]$
FLD	0.348	1.93	0.998	517
RLS	52.1	59.2	0.989	16.7
SVM	2750	7.46	0.998	134
EN	0.255	0.988	0.986	998

Table 7: Performance comparison of the chosen classification algorithms: EN and FLD yield the best performance in terms Figure of Merit (FoM).

by means of classification accuracy is a very time consuming process, which however yield very important results in terms of classification computational complexity reduction. The compound of ranking, FDR and *divergence* is used to identify which ones feature-parameter combinations are the the best suited for tissue identification.

Finally, the performance yielded by the classifier on a testing set disjoint from the training/validation set are retained as a valid index of the currently extracted feature-parameter set accuracy at discriminating between the different zones. In particular, choosing a zone z_i and estimating the fractions of true positives and false positives of the test "$\bar{x}$ belongs to zone z_i" [81] as the detection threshold is varied, the ROC curve can be plotted, as shown in Figure 70. The area under this curve can vary between 0 and 1, being 1/2 the worst value.

This parameter is related in a direct and natural way to cost/benefit analysis of diagnostic decision making. It was widely used by physicians in medicine, radiology, psychology and other areas for many decades since now, and it has been introduced relatively recently in other areas like machine learning and data mining.

Figure 70 represents a comparison between the ROC curves obtained from four different classifiers processing a dataset of US images of prostatic glands affected by carcinoma. More than 9000 ROI where automatically generated, their feature vector estimated, then classified with four different algorithms: Fisher Linear Discriminant (FLD) [80], Regularized Least Square (RLS) [82], Support Vector Machine (SVM) [83], and Elastic Net (EN) [84].

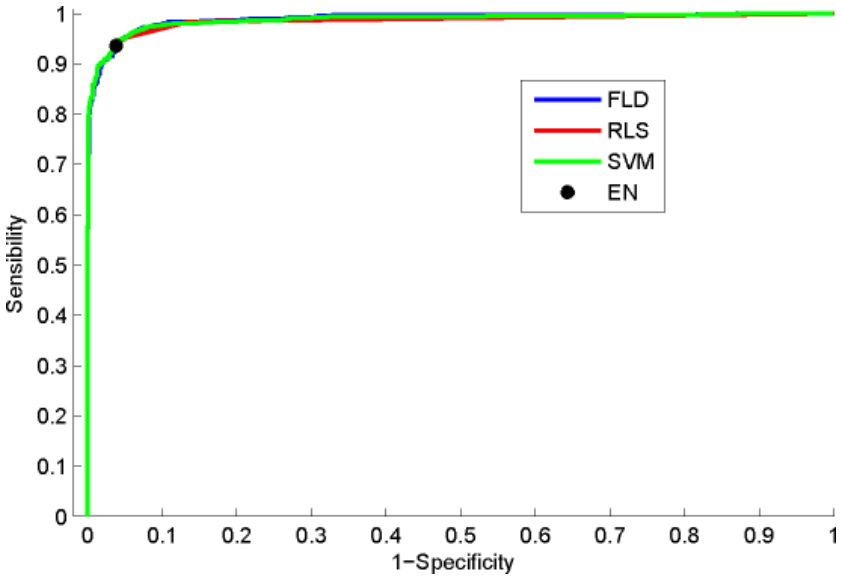


Figure 70: A comparison of the ROC curves obtained from different classification algorithms on the same data-set processed with the proposed method of feature/parameter generation.

As shown in Table 7, all the tested classifier gave very good results in terms of area below their ROC curves, with EN being the fastest algorithm in terms of combined training and testing time. The fact that a simple classifier like FLD is capable of yielding almost ideal ROC hints that kernel-based feature extraction techniques generate components very representative of the different tissue kinds, making sufficiently fast and accurate tissue classification feasible.

9 NEURAL SPIKE SORTING

In Section 2.1 a technique known as multi-electrode extracellular recordings or MER was introduced: this is one of the main techniques used by neurophysiologist to study the central nervous system and by neurologist to diagnose neuro-muscular diseases. These recordings are made of the superposition of neural action potentials AP, which must be separated and assigned to individual neurons: this problem is called spike sorting.

To improve existing sorting algorithms [85], a signal model was presented in Section 2.4 which allowed to develop an ad-hoc Wavelet based denoising algorithm which can account for different kinds of noise, as discussed in Chapter 7. The application of this algorithm to MER proved quite useful to ease the sorting of overlapping APs using an efficient L_1 norm minimization approach with a reduced implementation complexity.

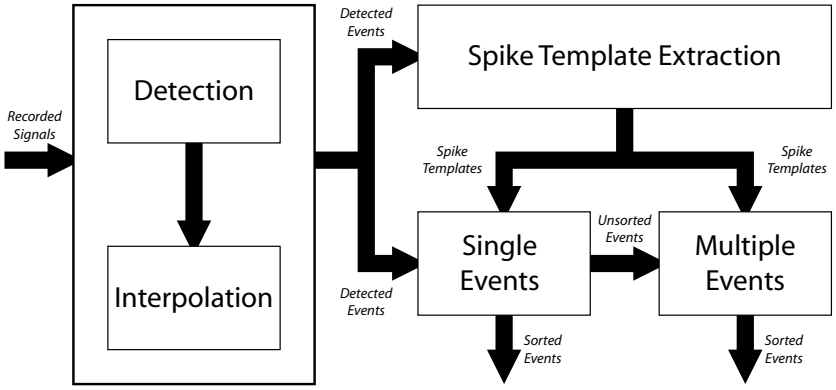


Figure 71: Schematic of a classical spike sorting algorithm: template matching is used to sort single and multiple events, originating an exponential dependency of computational cost from the number of superimpose spikes.

9.1 CLASSICAL SPIKE SORTING ALGORITHMS

A spike sorting algorithm [85] is typically a multistep algorithm based on a five steps, as shown in Figure 71. Spike detection is performed by amplitude thresholding each of the signals received from the MER acquisition probe: an AP is recorded whenever the neural signal exceeds a fixed amplitude threshold.

After this step, the recorded APs are oversampled by means of interpolation algorithms and aligned to their respective peaks: this procedure has a poor efficiency as no information concerning the noise embedding the signal is exploited. Moreover its accuracy is fundamental for the subsequent step, where a spike templates library is build by means of Principal Component Analysis (PCA). This step is responsible both for the removal of the noise in which the spikes are embedded and for the generation of the set of templates used to distinguish between different cells activation.

The last step, which constitutes the real sorting phase, is performed following two different policies: in case of single events, where only one cell fires a spike at a given time, template matching is performed, selecting the most feasible template from the generated library. All the detected activations undergo this phase: if a sufficient accuracy is reached,

the activation is considered sorted and is excluded from the phase of multiple activation template matching, otherwise it is marked for further processing.

Only the unsorted activations are processed with multiple templates: this is due to the fact that this procedure has a very high computational cost, thus requiring a very long time to be carried on. Naïve multiple template matching algorithms try every possible combination of template and relative delays, resulting into a complexity which is exponential in the number of superimposed AP to be sorted and in the maximum lag separating two adjacent spikes.

To reduce this cost to a more acceptable one, a Monte-Carlo Markov Chain (MCMC) approach can be followed [85], embedding into the Markov Chain model informations about refractory times, inter-spike interval and any other meaningful parameter related to neuronal activation. At the same time, thanks to the Monte Carlo approach, only a fraction of all the candidate activation sequences is testes and matched with the current activation. Nevertheless, triple and more complex activations are still too complex to be sorted, resulting into a number of unsorted events depending on the recorded neural activity.

9.2 IMPROVED SPIKE SORTING ALGORITHMS

Tackling the inefficiency of standard spike sorting algorithms, improvements can be worked out exploiting the information embedded into the signal. Three different areas had been identified where advancements could take place: signal preprocessing, template extraction and spike sorting. Different techniques can be exploited in each area to optimize the sequence of operations and reduce the computational complexity.

First, it proved quite successful to introduce a denoising stage before spike detection take place, as shown in Figure 72. Adopting model (7.1) for the neural signal, it is natural to remove noise with the WARES algorithm, as discussed in Chapter 7. This adaptive denoising algorithm can account for a large number of noise models, while preserving the shape of the neural spikes, as shown in Figure 73.

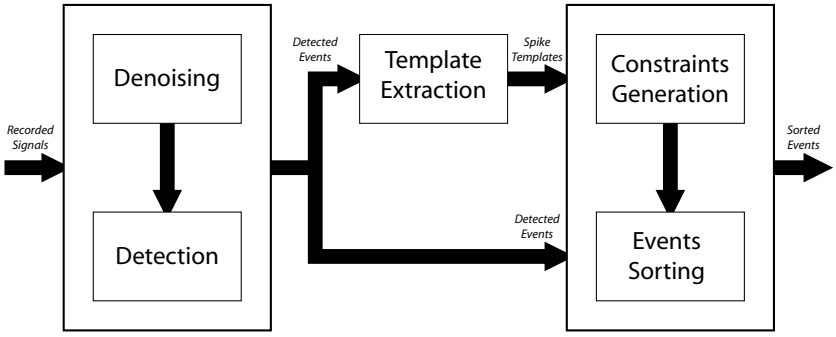


Figure 72: Schematic of the proposed improved spike sorting algorithm: different techniques can be exploited in each area to optimize the sequence of operations and reduce the computational complexity.

Comparing the results obtained from this algorithm with the output of the adaptive block denoising technique *rec-neighblock* [28], some differences may be highlighted: first of all, WARES features a better noise rejection than the block technique, as shown in figures 56-58. Secondly, in presence of adjacent spikes WARES allows for a better reconstruction of low amplitude APs. Last but not least, LFP low-frequency oscillations can be removed much more efficiently, without requiring and additional filtering stage.

In conclusion, as the global noise level gets reduced, AP detection thresholding can be lowered, improving the efficiently in discerning faint spikes; at the same time, since the WARES algorithm accounts for jitter noise removal AP interpolation is not deemed important anymore.

Moving on, thinking about spike sorting as a complex deconvolution process, template extraction can be substituted by one of the impulse response estimation algorithms discussed in Chapter 4. In fact, although the fired AP may change over the entire lifetime of a neural cell, it commonly assumed [18, 32] to do not change over a short amount of time.

From this standpoint, the series of events represented by a neuron firing its AP at given instants in time can be thought as the convolution of the neuron AP with a train of properly spaced Dirac delta functions. Because of this,

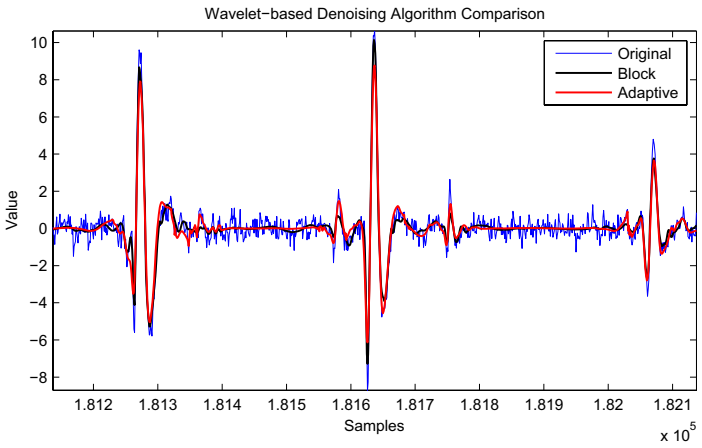


Figure 73: Received neural signal compared to two different denoised versions: one resulting from a block algorithm (*recneighblock* [28]), the other from the WARES adaptive algorithm.

if it is assumed that all neurons involved in a given MER experiment fire the same AP, cepstral techniques can be used, after a proper signal denoising step, to estimate the AP.

The biggest difficulty in this task is that of the AP being not guaranteed to be minimum-phase: this fact implies that algorithms discussed in Chapter 4 will fail in reconstructing the phase of the AP, as it can be clearly seen in Figure 74. Here, after performing signal denoising with the WARES algorithm, a mixed domain cepstral processing was undertaken to extract the neural AP, whose estimation is shown on the right side of Figure 74.

The shape of the estimated AP resembles the true one, although not as closely as one could desire. In fact, a phase inversion is present, as each of the recorded AP shown on top of Figure 74 first display a pronounced undershoot, followed by two oscillations; on the contrary, the estimated AP features first an overshoot followed by an undershoot.

Besides, the effects of this phenomenon are visible in the results of the deconvolution processing; processing the denoised signal by means of the ADID algorithms –on bottom left of Figure 74–, all the estimated spikes amplitudes are negative. Nevertheless, the results of the deconvolution are quite good at locating the AP in time, while preserving the

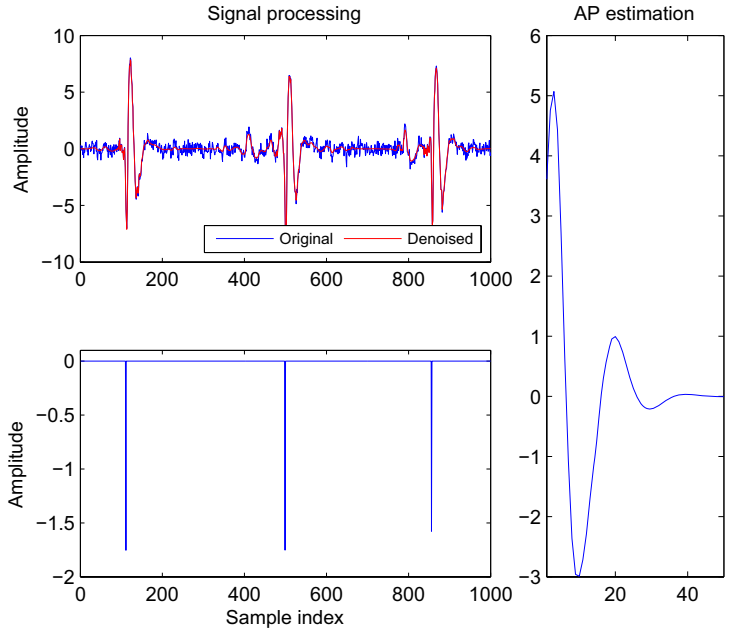


Figure 74: Example of spike sorting combining pulse estimation and deconvolution algorithms: mixed domain cepstral estimation and ADID were used respectively.

relative spacing.

An alternative solution can be envisioned by means of Shift Invariant Spaces (SIS) [86, 87]: in this context a signal in L^2 is assumed to belong to a space in the form

$$\mathcal{S}(\Phi) := \text{closure}_{L_2}\text{span}\{\phi_i(x-k) : i = 1,\dots,n, k \in \mathbb{Z}^d\} \tag{9.1}$$

where $\Phi = \{\phi_1,\dots,\phi_n\}$ is a set of function in $L^2(\mathbb{R}^d)$.

Given a space $\mathcal{S}$ and a set of functions Φ, if (9.1) holds for every element in the space, then Φ is called a *set of generators* for $\mathcal{S}$. If so, a generic element in $\mathcal{S}$ can be expressed as

$$x(n) = \sum_{i=1}^{N} \sum_{j\in\mathbb{Z}^d} \alpha_i(j)\phi_i(n+j) \tag{9.2}$$

where $\alpha_i \in l^2(\mathbb{Z}^d)$ is a proper set of weights.

Thinking about f in (9.2) as the received neural signal, it is possible to associate the set of AP templates fired by the cells

to the set of generators ϕ_i and the amplitude of each AP to the set of weights α_i. From this point of view, the problem of estimating the set of AP templates given the signal f is the same of identifying the generator set Φ which give birth to the space $\mathcal{S}$ which f belongs to. According to [86, 87], this problem can be efficiently solved in the frequency domain by means of Singular Value Decomposition (SVD).

Depending on the assumptions made on the AP waveform set, the pulse estimation or the SIS method are used as template extraction procedure in the schematic of shown in Figure 72.

Once the set of AP templates has been estimated, it is possible to exploit the intrinsic signal sparsity and adopt a L_1 norm minimization technique to sort the spikes with a reduced complexity. In fact, due to the presence of the refractory period, neural cells can fire AP only once or twice every 5 ms: at a sampling rate of 25 KHz, this means that if it could be possible to look to the true activation sequence, more or less only one over 100 of its samples are different from zero. The information resulting from spike soring is thus very sparse.

In mathematical terms, looking at the received signal $x[n]$ as to a superposition of convolutions between the activation sequence of each neural cell $a_i[n]$ and the corresponding action potential ϕ_i, it is possible to express $x[n]$ as

$$x[n] = \sum_{i=1}^{N} \sum_{m \in \mathbb{Z}} a_i[m]\phi_i[n-m] \tag{9.3}$$

where N is the number of concurrently spiking neurons.

Considering only an interval in time, (9.3) can be rewritten in matrix form as

$$\mathbf{x} = \Phi\mathbf{a} \tag{9.4}$$

where Φ is the column-wise tiling of the convolution matrices $\Phi_0, \Phi_1, \ldots, \Phi_N$ and $\mathbf{a}$ is the row-wise tiling of the activation vectors $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N$.

In theory, to recover the activation vector $\mathbf{a}$, the following constrained minimization of the L_0 norm of $\mathbf{a}$ should be used.

$$\mathbf{a}_e = \min_{\mathbf{a}} \|\mathbf{a}\|_0 \quad \text{s.t.} \quad \Phi\mathbf{a} = \mathbf{x} \tag{9.5}$$

In presence of noise, a vector $\mathbf{n}$ is added to $\mathbf{x}$; if this noise is considered to have zero mean and variance σ_n^2, the minimization problem becomes

$$\mathbf{a}_e = \min_{\mathbf{a}} \|\mathbf{a}\|_0 \quad \text{s.t.} \quad \|\Phi\mathbf{a} - \mathbf{x}\|_2 \leqslant \sigma_n^2 \tag{9.6}$$

Both these minimizations are NP complex problems due to the presence of the L_0 norm: according to [88, 89], under some constraints on the sparsity of $\mathbf{a}$ and the structure of Φ, the L_0 norm can be substituted by a L_1 norm. The resulting problem is thus much easier to solve.

Moreover, additional constraints can be introduced to further simplify the minimization task. First of all, if the set of AP waveforms was correctly estimated, there is no need to consider negative values of the elements of the activation vector: AP may have different amplitudes on different electrodes due to distance attenuation but no phase inversion is possible. Next, neurons shall activate only once in a given time interval if such is chosen short enough.

These considerations can be expressed in mathematical terms as

$$\begin{aligned} \mathbf{a} &\succeq 0 \\ \mathbf{a}^T\Sigma\mathbf{a} &= 0 \end{aligned} \tag{9.7}$$

where Σ is the so called *selection matrix*.

In case of single events sorting Σ elements are all ones except for the main diagonal, which is filled with zeros. When multiple activation is considered, Σ features a block structure: on the main diagonal, blocks identical to the previous case are present; outside the main diagonal, blocks are filled with ones if the corresponding neurons cannot activate simultaneously, with zeros otherwise. Finally, in case a given neuron cannot activate at all in the considered time interval, the corresponding diagonal block is filled with ones on the main diagonal too.

Thanks to the first constraint in (9.7) the L_1 norm of $\mathbf{a}$ can be written as the plain sum of the activation vector elements; the minimization problem thus becomes

$$\begin{aligned} \mathbf{a}_e = \min_{\mathbf{a}} \mathbf{1}^T\mathbf{a} \quad \text{s.t.} \quad & \mathbf{a} \succeq 0 \\ & \mathbf{a}^T\Sigma\mathbf{a} = 0 \\ & \|\Phi\mathbf{a} - \mathbf{x}\|_2 \leqslant \sigma_n^2 \end{aligned} \tag{9.8}$$

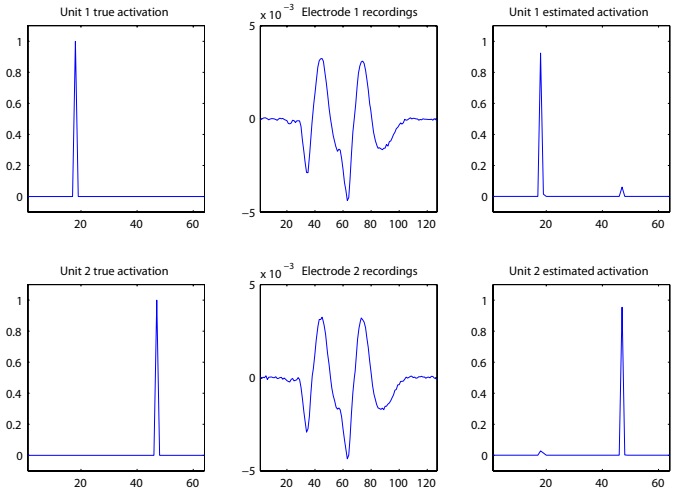


Figure 75: L_1 norm minimization based spike sorting example: a test on two very similar superimposed spikes shows that sorting is feasible.

which is a much simpler Linear Programming (LP) task [90].

As hinted in Figure 72, by switching the selection matrix, thus the LP task constraints, the same technique can be used to sort both single and multiple activation, originating a spike sorting algorithm with a complexity much lower than the classic one.

Figure 75 shows a test of the L_1 norm minimization based spike sorting technique performed on two very similar superimposed spikes in the case of a two channel simulated MER. The estimated activation vectors for the involved neurons are very similar to the true one, although a complete separation was not possible.

CONCLUSION

Adaptive processing of biological signals in the recent past years steadily increased in importance due to the improvements in the acquisition equipment. In the US field it is possible nowadays to record up to 50 high resolution frames per seconds made of more than 4000 rows and 128 columns. At the same time, in the MER research environment, 64 or more ER traces can be recorded at the same time in order to track the activity of neural systems as complex as Primates brain.

Signals incoming from these sources feature complex models and are typically subject to non-linear, time-varying degrading effects due to signal dependent noise and to acquisition equipment. In this work, many advanced signal processing algorithms, most of them original creations developed during Ph.D research activity, were presented as a mean to adaptively recover the undistorted incoming signal. These algorithms share an adaptive approach to signal analysis, meaning that their behavior is not dependent only on designer choices, but driven by input signal characteristics too.

In Chapters 1 and 2 of this work, after a swift recalling of

US imaging and MER technique history, a wide perspective about how signals are acquired was given. Respective signal models were then presented, highlighting the differences between the current state of the art and what was developed during the research activity. In particular, an innovative flexible signal model was introduced for the purpose of ER signal processing, introducing a signal dependent noise component to account for misalignment between neural spikes and recording clock.

Chapter 3 discussed about some flavors of Wavelet Transform, namely CWT, DWT, SWT and QCWT, presenting the differences between each of them, their strong points and the algorithms meant for calculating the respective transform domain coefficients. The last sections of this chapter discussed respectively how to iteratively estimate the Wavelet scaling function for the purpose of computing the QCWT and how to calculate the coefficients of the filters employed in SW/HW implementation of the different transforms.

Detailed descriptions of the tools employed in this work to perform signal processing were given in the central four chapters. Impulse response estimation tools were introduced first, in Chapter 4: starting from two known techniques, an original algorithm combining the advantages of the former two was proposed. The three resulting methods were then compared in terms of estimation error, impulse stability and coefficient number: the flexibility of the proposed algorithm allowed it to better the other methods on two metrics out of three.

In Chapter 5 the discussion moved to deconvolution techniques: two algorithms already in literature are compared to an original one, called ADID, capable of dealing with both non-minimum phase estimated impulses and non-synchronized sampling. A non-iterative, adaptive version of the WLSD algorithm was also presented in this chapter: these techniques were both developed during the current research activity. Performance comparisons made use of the state of the art concerning image quality assessment, contrast gain estimation and resolution gain quantification; visual inspection was also used in the concluding part of the chapter, highlighting the good results featured by ADID.

Chapter 6 examined several adaptive filtering techniques based on the LMS filter update policy in a configuration

meant for signal deconvolution. Starting from the basic adaptive FIR filter, three advanced algorithms were presented, based respectively on higher order statistics, polyphase decomposition and IIR filtering. Original contributes, constituted by the adaptive gain versions of the NEM, SAF and IIR LMS algorithms, were introduced in this chapter by a discussion on the adaptive gain FIR LMS filter and are developed in details in the following appendixes. Performance comparison followed the same criteria of Chapter 5.

Finally, Chapter 7 presented an innovative denoising algorithm, called WARES, based on the signal model introduced in Chapter 2; this technique was developed during the research activity for the purpose of signal dependent noise removal. The method is based on LMMSE minimization: Wavelet domain was used in order to overcome some restrictive hypotheses that were introduced by the development of a similar technique in the time domain. Performance comparison was performed both on terms of synthetic and real signals denoising, paralleling the results of WARES with other Wavelet based signal processing algorithms. A detailed documentation concerning the derivation of both the time and Wavelet domain implementations is given in the following appendixes.

The final chapters examined two possible applications of adaptive multiscale signal processing algorithms. An innovative preprocessing section meant for biological tissue classification by means of US imaging was presented in Chapter 8: thanks to the flexibility of the model adopted in WARES, this algorithm was used both to perform preliminary denoising and to separate true echoes from speckle noise. Blind deconvolution was accomplished first by estimating the acquisition system PSF with the algorithms presented in Chapter 4 then applying the deconvolution techniques discussed in Chapter 5. A classification protocol exploiting the results produced by this section was also detailed, showing that accurate tissue classification based on US imaging is feasible.

A second application concerning neural spike sorting was depicted in Chapter 9: again WARES algorithm was employed to extract the desired signal from the embedding noise. Spike template library was built by applying pulse estimation on the denoised signal. Finally, ADID deconvo-

lution allowed for position and amplitude estimation of each neural spike within the selected time interval. Some more advanced techniques were also presented, offering the possibility to reduce the overall system complexity and to improve accuracy in presence of simultaneous firing neurons.

Parts of the research results presented in this thesis have already been published in [P-1, P-2, P-3, P-4, P-5, P-6, P-7, P-8, P-9, P-10]. The other results will be the subject of future papers.

Part IV

APPENDIX

A

LEAST MEAN SQUARE FILTERING

A.1 ADAPTIVE GAIN NEM DERIVATION

The derivation of the adaptive gain version of the NEM algorithm starts from (6.17). This relationship defines the instantaneous cost function $J(n)$ as

$$J[n] = \sum_{i=1}^{M} \frac{a_i}{2i} e[n]^{2i} \tag{A.1}$$

where, with reference to Figure 35,

$$e[n] = d[n] + \sum_{i=1}^{N} u[n-i]\tilde{H}_i[n] \tag{A.2}$$

is the estimate of the input signal $x[n]$ at time step n and $\tilde{H}_i[n]$ is the i-th coefficient of the filter $\tilde{\mathbf{H}}$ at the same step.

As for the Adaptive Gain LMS filter, the AWC rule for this algorithm involves the following steepest-descent update policy for each filter coefficient $\tilde{H}_i[n]$ and for the step-size

parameter $\mu[n]$

$$\begin{aligned} \tilde{H}_i[n+1] &= \tilde{H}_i[n] - \mu[n]\frac{\partial J[n]}{\partial \tilde{H}_i[n]} \\ \mu[n+1] &= \left[\mu[n] - \alpha\frac{\partial J[n]}{\partial \mu[n]}\right]_{\mu-}^{\mu+} \end{aligned} \tag{A.3}$$

where α is the *learning-rate* parameter for this algorithm.

Expanding the partial derivatives in (A.3), these relationships are obtained:

$$\begin{aligned} \frac{\partial J[n]}{\partial \tilde{H}_i[n]} &= \frac{\partial J[n]}{\partial e[n]}\frac{\partial e[n]}{\partial \tilde{H}_i[n]} \\ &= \epsilon_0[n]u[n-i] \\ \frac{\partial J[n]}{\partial \mu_[n]} &= \frac{\partial \mu_[n]}{\partial e[n]}\frac{\partial e[n]}{\partial \tilde{H}_i[n]} \\ &= \epsilon_0[n]\sum_{i=1}^{N} u[n-i]\frac{\partial \tilde{H}_i[n]}{\partial \mu[n]} \end{aligned} \tag{A.4}$$

where the partial derivative of $J[n]$ with respect to $e[n]$ was conveniently named as $\epsilon_0[n]$. As a consequence, from the definition of $J[n]$ it follows that

$$\epsilon_0[n] = \sum_{i=1}^{M} a_i e[n]^{2i-1} \tag{A.5}$$

The only yet unknown quantity in this algorithm is the partial derivative of $\tilde{H}_i[n]$ with respect to $\mu[n]$: calling it $\Psi_i[n]$, a recursive definition of this quantity is obtained differentiating the first equation of (A.3):

$$\Psi_i[n+1] = \Psi_i[n] - \left[\frac{\partial J[n]}{\partial \tilde{H}_i[n]} + \mu[n]\frac{\partial}{\partial \mu[n]}\frac{\partial J[n]}{\partial \tilde{H}_i[n]}\right] \tag{A.6}$$

Substituting the first equation of (A.4) in (A.6) and expanding the second partial derivative, the following relationship is found

$$\Psi_i[n+1] = \Psi_i[n] - u[n-i]\left[\epsilon_0[n] + \mu[n]\frac{\partial \epsilon_0[n]}{\partial e[n]}\frac{\partial e[n]}{\partial \mu[n]}\right] \tag{A.7}$$

where the partial derivative of $\epsilon_0[n]$ is named $\epsilon_1[n]$ and the quantity between square brackets is named $\eta[n]$: from the definition of $\epsilon_0[n]$ it is found that

$$\epsilon_1[n] = \sum_{i=1}^{M}(2i-1)a_i e[n]^{2i-2} \tag{A.8}$$

Calling $\phi[n]$ the partial derivative of $e[n]$ with respect to $\mu[n]$, the following relationship for $\eta[n]$ is obtained

$$\eta[n] = \epsilon_0[n] + \mu[n]\epsilon_1[n]\phi[n] \tag{A.9}$$

which concludes the derivation of the AWC rule equations for the Adaptive Gain NEM algorithms

It is now possible to write down the sequence of operations required at each step to compute the input signal estimation $e[n]$ and the updates of the internal variables:

1. $f[n] = \sum_{i=1}^{N} u[n-i]\tilde{H}_i[n]$
2. $\phi[n] = \sum_{i=1}^{N} u[n-i]\Psi_i[n]$
3. $e[n] = d[n] + f[n]$
4. $\epsilon_0[n] = \sum_{i=1}^{M} a_i e[n]^{2i-1}$
5. $\epsilon_1[n] = \sum_{i=1}^{M} (2i-1)a_i e[n]^{2i-2}$
6. $\eta[n] = \epsilon_0[n] + \mu[n]\epsilon_1[n]\phi[n]$
7. $\tilde{H}_i[n+1] = \tilde{H}_i[n] - \mu[n]\epsilon_0[n]u[n-i]$
8. $\Psi_i[n+1] = \Psi_i[n] - \eta[n]u[n-i]$
9. $\mu[n+1] = [\mu[n] - \alpha\epsilon_0[n]\phi[n]]_{\mu-}^{\mu+}$

As a conclusion, it is possible to show that $\eta[n]$ is weighted sum of second estimations of the input signal $x[n]$. Expanding $\epsilon_0[n]$ and $\epsilon_1[n]$ in (A.9) the following relationship is found

$$\eta[n] = \sum_{i=1}^{M} a_i e[n]^{2i-1} + \mu[n]\phi[n] \sum_{i=1}^{M} (2i-1)a_i e[n]^{2i-2} \tag{A.10}$$

where it is possible to collect a_i and $e[n]$, yielding

$$\eta[n] = \sum_{i=1}^{M} a_i e[n]^{2i-2} \left[e[n] + (2i-1)\mu[n]\phi[n]\right] \tag{A.11}$$

Calling $\xi_i[n]$ the terms between square brackets and substituting (A.2) and (A.4) into (A.11), by collecting $u[n-j]$ the following definition for $\xi_i[n]$ is obtained

$$\xi_i[n] = d[n] + \sum_{j=1}^{N} u[n-j]\left[\tilde{H}_j[n] + (2i-1)\mu[n]\Psi_j[n]\right]$$

(A.12)

Recalling that $\Psi_j[n]$ is the partial derivative of $\tilde{H}_i[n]$ with respect to $\mu[n]$, (A.12) can be thought as a second order estimate of $x[n]$. Substituting (A.12) into (A.11), it is possible to define $\eta[n]$ as the following time-varying weighted sum of second order estimations:

$$\eta[n] = \sum_{i=1}^{M} a_i e[n]^{2i-2} \xi_i[n] \tag{A.13}$$

It is worth noting that (A.13) reduces to (6.13) when $M = 1$.

A.2 ADAPTIVE GAIN SAF DERIVATION

The derivation of the adaptive gain version of the SAF algorithm is based on the following instantaneous cost function $J(n)$

$$J[n] = \frac{1}{2} \sum_{\alpha=0}^{K-1} \frac{a_\alpha}{e}_\alpha [n]^2 \tag{A.14}$$

where, with reference to Figure 45,

$$e_\alpha[n] = d_\alpha[n] + \sum_{\beta=0}^{K-1} \sum_{i=1}^{N/K} u_{\alpha\beta}[n-i] \tilde{H}_{\beta,i}[n] \tag{A.15}$$

is the estimation of the coefficients of the α-th band of the input signal $x[n]$ at time-step n and $\tilde{H}_{\beta,i}[n]$ the i-th coefficients of the β-th polyphase component of the filter $\tilde{\mathbf{H}}$. The following relationship between $\tilde{H}_{\beta,i}[n]$ and $\tilde{H}_i[n]$ holds due to the polyphase decomposition properties:

$$\tilde{H}_{\beta,i}[n] = \tilde{H}_{\beta+iK}[n] \tag{A.16}$$

The AWC rule for this algorithm involves the following steepest-descent update policy for each filter coefficient $\tilde{H}_{\beta,i}[n]$ and for the step-size parameter $\mu[n]$

$$\begin{aligned} \tilde{H}_{\beta,i}[n+1] &= \tilde{H}_{\beta,i}[n] - \mu[n] \frac{\partial J[n]}{\partial \tilde{H}_{\beta,i}[n]} \\ \mu[n+1] &= \mu[n] - \xi \frac{\partial J[n]}{\partial \mu[n]} \end{aligned} \tag{A.17}$$

where ξ is the *learning-rate* parameter for this algorithm.

Expanding the partial derivatives in (A.17), these relationships are obtained:

$$\begin{aligned} \frac{\partial J[n]}{\partial \tilde{H}_{\beta,i}[n]} &= \sum_{\alpha=0}^{K-1} \frac{\partial J[n]}{\partial e_\alpha[n]} \frac{\partial e_\alpha[n]}{\partial \tilde{H}_{\beta,i}[n]} \\ \frac{\partial J[n]}{\partial \mu[n]} &= \sum_{\alpha=0}^{K-1} \frac{\partial J[n]}{\partial e_\alpha[n]} \frac{\partial e_\alpha[n]}{\partial \mu[n]} \end{aligned} \tag{A.18}$$

From (A.14) and (A.15) each of the partial derivatives on the left hand side of (A.18) can be expressed as follows:

$$\begin{aligned} \frac{\partial J[n]}{\partial e_\alpha[n]} &= a_\alpha e_\alpha[n] \\ \frac{\partial e_\alpha[n]}{\partial \tilde{H}_{\beta,i}[n]} &= u_{\alpha\beta}[n-i] \\ \frac{\partial e_\alpha[n]}{\partial \mu[n]} &= \sum_{\beta=0}^{K-1} \sum_{i=1}^{N/K} u_{\alpha\beta}[n-i] \frac{\partial \tilde{H}_{\beta,i}[n]}{\partial \mu[n]} \end{aligned} \tag{A.19}$$

Again, the only yet undefined quantity is the partial derivative of $\partial \tilde{H}_{\beta,i}[n]$ with respect to $\mu[n]$. Calling $\Psi_{\beta,i}[n]$ this quantity, a suitable recursive relationship for its evaluation can be obtained differentiating the first equation in (A.17), yielding

$$\Psi_{\beta,i}[n+1] = \Psi_{\beta,i}[n] - \left[\frac{\partial J[n]}{\partial \tilde{H}_{\beta,i}[n]} + \mu[n] \frac{\partial}{\partial \mu[n]} \frac{\partial J[n]}{\partial \tilde{H}_{\beta,i}[n]} \right] \tag{A.20}$$

where the second order derivative of $J[n]$ can be obtained expanding and differentiating the first equation in (A.18)

$$\frac{\partial}{\partial \mu[n]} \frac{\partial J[n]}{\partial \tilde{H}_{\beta,i}[n]} = \sum_{\alpha=0}^{K-1} a_\alpha u_{\alpha\beta}[n-i] \frac{\partial e_\alpha[n]}{\partial \mu[n]} \tag{A.21}$$

By substituting (A.18) and (A.21) into (A.20) and collecting the appropriate terms, the following recursive expression for $\Psi_i[n]$ is obtained

$$\Psi_{\beta,i}[n+1] = \Psi_{\beta,i}[n] - \sum_{\alpha=0}^{K-1} a_\alpha u_{\alpha\beta}[n-i] \left[e_\alpha[n] + \mu[n] \frac{\partial e_\alpha[n]}{\partial \mu[n]} \right] \tag{A.22}$$

where the term within square brackets is called $\eta[n]$.

Calling $\phi_\alpha[n]$ the partial derivative of $e_\alpha[n]$ with respect to $\mu[n]$, $\eta[n]$ can be expressed in the following way

$$\eta[n] = e_\alpha[n] + \mu[n]\phi_\alpha[n] \tag{A.23}$$

which, very likely to what was discussed in the previous section, can be thought as a second order estimate of the α-th band input signal coefficients.

It is now possible to write down the sequence of operations required at each step to compute the α-th band input signal coefficient estimations $e_\alpha[n]$ and the updates of the internal variables:

$$\begin{aligned}
&1. \quad f_\alpha[n] = \textstyle\sum_{\beta=0}^{K-1}\sum_{i=1}^{N/K} u_{\alpha\beta}[n-i]\tilde{H}_{\beta,i}[n] \\
&2. \quad \phi_\alpha[n] = \textstyle\sum_{\beta=0}^{K-1}\sum_{i=1}^{N/K} u_{\alpha\beta}[n-i]\Psi_{\beta,i}[n] \\
&3. \quad e_\alpha[n] = d_\alpha[n] + f_\alpha[n] \\
&4. \quad \eta_\alpha[n] = e_\alpha[n] + \mu n\phi_\alpha[n] \\
&5. \tilde{H}_{\beta,i}[n+1] = \tilde{H}_{\beta,i}[n] - \mu(n)\textstyle\sum_{\alpha=0}^{K-1} a_\alpha e_\alpha[n]u_{\alpha\beta}[n-i] \\
&6. \Psi_{\beta,i}[n+1] = \Psi_{\beta,i}[n] - \textstyle\sum_{\alpha=0}^{K-1} a_\alpha \eta_\alpha[n]u_{\alpha\beta}[n-i] \\
&7. \quad \mu[n+1] = \left[\mu[n] - \xi\textstyle\sum_{\alpha=0}^{K-1} a_\alpha e_\alpha[n]\phi_\alpha[n]\right]_{\mu-}^{\mu+}
\end{aligned}$$

A.3 ADAPTIVE GAIN IIR LMS DERIVATION

The derivation of the adaptive gain version of the IIR LMS algorithm is based on the same instantaneous cost function $J(n)$ of the standard LMS

$$J[n] = \frac{1}{2}e_\alpha[n]^2 \tag{A.24}$$

where, with reference to Figure 49,

$$e[n] = d[n] + \sum_{i=1}^{M} H_{d,i}[n]u[n-i] - \sum_{i=1}^{N} H_{n,i}[n]e[n-i] \tag{A.25}$$

Applying the steepest-descent algorithm to both filters and to the step-size parameter, the following iterative relationships define the AWC rule set for this adaptive filtering algorithm

$$\begin{aligned}
\tilde{H}_{d,i}[n+1] &= \tilde{H}_{d,i}[n] - \mu[n]\frac{\partial J[n]}{\partial \tilde{H}_{d,i}[n]} \\
\tilde{H}_{n,i}[n+1] &= \tilde{H}_{n,i}[n] - \mu[n]\frac{\partial J[n]}{\partial \tilde{H}_{n,i}[n]} \\
\mu[n+1] &= \mu[n] - K\frac{\partial J[n]}{\partial \mu[n]}
\end{aligned} \tag{A.26}$$

where K is the *learning-rate* parameter for the IIR LMS algorithm.

Expanding the partial derivatives in (A.26), these relationships are found:

$$\begin{aligned}
\frac{\partial J[n]}{\partial \tilde{H}_{d,i}[n]} &= e[n]\frac{\partial e[n]}{\partial \tilde{H}_{d,i}[n]} = e[n]a_i[n] \\
\frac{\partial J[n]}{\partial \tilde{H}_{n,i}[n]} &= e[n]\frac{\partial e[n]}{\partial \tilde{H}_{n,i}[n]} = e[n]b_i[n] \\
\frac{\partial J[n]}{\partial \mu[n]} &= e[n]\frac{\partial e[n]}{\partial \mu[n]} = e[n]\phi[n]
\end{aligned} \tag{A.27}$$

where the three additional variables $a_i[n]$, $b_i[n]$ and $\phi[n]$ are used in place of the partial derivatives of $e[n]$.

These quantities can be defined in a recursive way by carrying on the differentiations in (A.27)

$$\begin{aligned}
\frac{\partial e[n]}{\partial \tilde{H}_{d,i}[n]} &= u[n-i] - \textstyle\sum_{j=1}^{N} H_{n,j}[n]\frac{\partial e[n-j]}{\partial \tilde{H}_{d,i}[n]} \\
\frac{\partial e[n]}{\partial \tilde{H}_{n,i}[n]} &= -e[n-i] - \textstyle\sum_{j=1}^{N} H_{d,j}[n]\frac{\partial e[n-j]}{\partial \tilde{H}_{d,i}[n]} \\
\frac{\partial e[n]}{\partial \mu[n]} &= \textstyle\sum_{i=1}^{M} \frac{\partial \tilde{H}_{d,i}[n]}{\mu[n]} u[n-i] \\
&\quad - \textstyle\sum_{i=1}^{N} \frac{\partial \tilde{H}_{n,i}[n]}{\mu[n]} e[n-i] \\
&\quad - \textstyle\sum_{i=1}^{N} \tilde{H}_{n,i}[n]\frac{e[n-i]}{\mu[n]}
\end{aligned} \tag{A.28}$$

where, if the filter adapts at a slow enough rate, the following approximation are justified

$$\begin{aligned}
\frac{\partial e[n-j]}{\partial \tilde{H}_{d,i}[n]} &\simeq \frac{\partial e[n-j]}{\partial \tilde{H}_{d,i}[n-j]} = a_i[n-j] \\
\frac{\partial e[n-j]}{\partial \tilde{H}_{n,i}[n]} &\simeq \frac{\partial e[n-j]}{\partial \tilde{H}_{n,i}[n-j]} = b_i[n-j] \\
\frac{\partial e[n-i]}{\partial \mu[n]} &\simeq \frac{\partial e[n-i]}{\partial \mu[n]} = \phi[n-i]
\end{aligned} \tag{A.29}$$

which can be used to close the recursive relationships involving $a_i[n]$, $b_i[n]$ and $\phi[n]$. Calling the first two partial derivatives in the last equation of (A.27) respectively $\Psi_{d,i}[n]$ and $\Psi_{n,i}[n]$, the following relationships are obtained

$$\begin{aligned}
a_i[n] &= u[n-i] - \textstyle\sum_{j=1}^{N} H_{n,j}[n]a_i[n-j] \\
b_i[n] &= -e[n-i] - \textstyle\sum_{j=1}^{N} H_{d,j}[n]b_i[n-j] \\
\phi[n] &= \textstyle\sum_{i=1}^{M} \Psi_{d,i}[n]u[n-i] \\
&\quad - \textstyle\sum_{i=1}^{N} \Psi_{n,i}[n]e[n-i] \\
&\quad - \textstyle\sum_{i=1}^{N} \tilde{H}_{n,i}[n]\phi[n-i]
\end{aligned} \tag{A.30}$$

The quantities $\Psi_{d,i}[n]$ and $\Psi_{n,i}[n]$ can be evaluated by means of recursive relationships too: differentiating the first two equations of (A.26) and starting from $\Psi_{d,i}[n]$, the following is obtained

$$\Psi_{d,i}[n+1] = \Psi_{d,i}[n] - \left(\frac{\partial J[n]}{\partial \tilde{H}_{d,i}[n]} + \mu[n] \frac{\partial}{\partial \mu[n]} \frac{\partial J[n]}{\partial \tilde{H}_{d,i}[n]} \right) \tag{A.31}$$

Exploiting first the equation in (A.27), the second order derivative of $J[n]$ which appears in (A.31) can be expressed as

$$\frac{\partial}{\partial \mu[n]} \frac{\partial J[n]}{\partial \tilde{H}_{d,i}[n]} = \phi[n] a_i[n] + e[n] \frac{\partial a_i[n]}{\partial \mu[n]} \tag{A.32}$$

Plugging (A.32) into (A.31) and doing the following substitutions

$$\begin{aligned} \eta[n] &= e[n] + \mu[n]\phi[n] \\ \alpha_i[n] &= \tfrac{\partial a_i[n]}{\partial \mu[n]} \end{aligned} \tag{A.33}$$

equation (A.31) can be expressed as

$$\Psi_{d,i}[n+1] = \Psi_{d,i}[n] - \eta[n] a_i[n] - \mu[n] e[n] \alpha_i[n] \tag{A.34}$$

Calling $\beta_i[n]$ the partial derivative of $b_i[n]$ with respect to $\mu[n]$ and following the same steps which lead from (A.31) to (A.34), $\Psi_{n,i}[n]$ can be expressed as

$$\Psi_{n,i}[n+1] = \Psi_{n,i}[n] - \eta[n] b_i[n] - \mu[n] e[n] \beta_i[n] \tag{A.35}$$

Finally, it is possible to derive recursive relationships for $\alpha_i[n]$ and $\beta_i[n]$ too, differentiating the first two equations of (A.30) with respect to $\mu[n]$ and obtaining

$$\begin{aligned} \alpha_i[n] &= -\textstyle\sum_{j=1}^{N} \Psi_{n,j}[n] a_i[n-j] - \sum_{j=1}^{N} \tilde{H}_{n,j}[n] \alpha_i[n-j] \\ \beta_i[n] &= -\phi[n-i] - \textstyle\sum_{j=1}^{N} \Psi_{n,j}[n] b_i[n-j] \\ &\quad - \textstyle\sum_{j=1}^{N} \tilde{H}_{n,j}[n] \beta_i[n-j] \end{aligned} \tag{A.36}$$

It is now possible to write down the sequence of operations required at each step to compute the input signal

estimation $e[n]$ and the updates of the internal variables:

$$
\begin{aligned}
1.\quad & f_1[n] = \sum_{i=1}^{M} u[n-i]\tilde{H}_{d,i}[n] \\
2.\quad & f_2[n] = \sum_{i=1}^{N} e[n-i]\tilde{H}_{n,i}[n] \\
3.\quad & f_3[n] = \sum_{i=1}^{N} \phi[n-i]\tilde{H}_{n,i}[n] \\
4.\quad & g_{1,i}[n] = \sum_{j=1}^{N} a_i[n-j]\tilde{H}_{n,j}[n] \\
5.\quad & g_{2,i}[n] = \sum_{j=1}^{N} b_i[n-j]\tilde{H}_{n,j}[n] \\
6.\quad & g_{3,i}[n] = \sum_{j=1}^{N} \alpha_i[n-j]\tilde{H}_{n,j}[n] \\
7.\quad & g_{4,i}[n] = \sum_{j=1}^{N} \beta_i[n-j]\tilde{H}_{n,j}[n] \\
8.\quad & \Gamma_1[n] = \sum_{i=1}^{M} u[n-i]\Psi_{d,i}[n] \\
9.\quad & \Gamma_2[n] = \sum_{i=1}^{N} e[n-i]\Psi_{n,i}[n] \\
10.\quad & \Gamma_{3,i}[n] = \sum_{j=1}^{N} a_i[n-j]\Psi_{n,j}[n] \\
11.\quad & \Gamma_{4,i}[n] = \sum_{j=1}^{N} b_i[n-j]\Psi_{n,j}[n] \\
12.\quad & e[n] = d[n] + f_1[n] - f_2[n] \\
13.\quad & a_i[n] = u[n-i] - g_{1,i}[n] \\
14.\quad & b_i[n] = -e[n-i] - g_{2,i}[n] \\
15.\quad & \phi[n] = \Gamma_1[n] - \Gamma_2[n] - f_3[n] \\
16.\quad & \alpha_i[n] = -\Gamma_{3,i}[n] - g_{3,i}[n] \\
17.\quad & \beta_i[n] = -\phi[n-i] - \Gamma_{4,1} - g_{4,i} \\
18.\quad & \eta[n] = e[n] + \mu[n]\phi[n] \\
19.\quad & \tilde{H}_{d,i}[n+1] = \tilde{H}_{d,i}[n] - \mu[n]e[n]a_i[n] \\
20.\quad & \tilde{H}_{n,i}[n+1] = \tilde{H}_{n,i}[n] - \mu[n]e[n]b_i[n] \\
21.\quad & \Psi_{d_i}[n+1] = \Psi_{d_i}[n+1] - \eta[n]a_i[n] - \mu[n]e[n]\alpha_i[n] \\
22.\quad & \Psi_{n_i}[n+1] = \Psi_{n_i}[n+1] - \eta[n]b_i[n] - \mu[n]e[n]\beta_i[n] \\
23.\quad & \mu[n+1] = [\mu[n] - \kappa e[n]\phi[n]]_{\mu-}^{\mu+}
\end{aligned}
$$

B

WAVELET ADAPTIVE RESCALING DENOISING

B.1 TIME-DOMAIN RELATIONSHIPS DERIVATION

In order to estimate the clean signal $x[k]$ from the noisy signal model (7.1) described in Chapter 7 some relationships must be established between quantities dependent on $x[k]$ and on $y[k]$. Hypothesis made on both noises u and v as well as on x during the derivation of such relationships will be summarized at the end of this section.

Equation (7.2) involves both quantities which are directly measurable on the input signal $y[k]$ ($E[y[k]]$ and $\mathbf{C}_{yy}$) and quantities which depends on the unknown clean signal $x[k]$ ($E[x[k]]$ and $\mathbf{C}_{xy}$). The goal is to highlight the dependency between them, in order to infer useful properties.

Starting from $E[y[k]]$, due to the linearity of the expectation operator, is it possible to write

$$E[y[k]] = E[x[k]] + E[z[k]u[k]] + E[v[k]] \tag{B.1}$$

where z represent the convolution of the input signal x with the filter f as defined by the model (2.1).

By assuming v stationary with zero mean, the last term in (B.1) can be removed. Now, thanks to the definition of z and by assuming u independent of x, the following is obtained:

$$E[y[k]] = E[x[k]] + \sum_i f[i]E[x[k-i]]E[u[k]] \tag{B.2}$$

where the linearity of the expectation estimator was used. Finally, assuming u stationary with zero mean it is possible to write

$$E[y[k]] = E[x[k]] \tag{B.3}$$

which allow to substitute any expectation on $x[n]$ with an expectation on $y[n]$ and vice-versa.

Equipped with this relationship, $\mathbf{C}_{xy}$ can be considered: by definition

$$\left[\mathbf{C}_{xy}\right]_{pq} = E[(x[p] - E[x[p]])(y[q] - E[y[q]]) \tag{B.4}$$

Expanding the term $y[m]$ and plugging in (B.3) the following holds

$$\left[\mathbf{C}_{xy}\right]_{pq} = E\left[(x[p] - E[x[p]])(x[q] + z[q]u[q] + v[q] - E[x[q]])\right] \tag{B.5}$$

Next, exploiting again the linearity of the expectation operator, (B.5) is split up into

$$\begin{aligned}\left[\mathbf{C}_{xy}\right]_{pq} = E\left[(x[p] - E[x[p]])(x[q] - E[x[q]])\right] \\ + E[x[p]z[q]u[q]] + E[x[p]v[q]] \\ - E[x[p]]E[z[q]u[q] - v[q]]\end{aligned} \tag{B.6}$$

As $z[q]u[q] + v[q] = y[q] - x[q]$, assuming again u and v stationary with zero mean and u independent of x, (B.3) can be used to eliminate the last term from (B.6). Moreover, if v is assumed independent of x as well, the third term of (B.6) is zero too. Finally, remembering that z is defined as the convolution of x with the filter f, thanks to the linearity of the expectation, even the second term of (B.6) is zero, leaving only

$$\left[\mathbf{C}_{xy}\right]_{pq} = \left[\mathbf{C}_{xx}\right]_{pq} \tag{B.7}$$

where the right hand side does not depend on y.

Finally, $\mathbf{C}_{yy}$ can be examined: by definition

$$\left[\mathbf{C}_{yy}\right]_{pq} = E[(y[p] - E[y[p]])(y[q] - E[y[q]])] \tag{B.8}$$

holds, from which, expanding the term $y[p]$ and plugging in (B.3), the following is obtained

$$\begin{aligned}\left[\mathbf{C}_{yy}\right]_{pq} = E\big[&(x[p] + z[p]\,u[p] + v[p] - E[x[p]]) \\ &\cdot (x[q] + z[q]\,u[q] + v[q] - E[x[q]])\big]\end{aligned} \tag{B.9}$$

Next, exploiting again the linearity of the expectation operator, (B.9) is split up into four terms

$$\begin{aligned}\left[\mathbf{C}_{yy}\right]_{pq} = &\, E\big[(x[p] + E[x[p]])(x[q] + E[x[q]])\big] \\ &+ E\big[(x[p] + E[x[p]])(z[q]\,u[q] + v[q])\big] \\ &+ E\big[(z[p]\,u[p] + v[p])(x[q] + E[x[q]])\big] \\ &+ E\big[(z[p]\,u[p] + v[p])(z[q]\,u[q] + v[q])\big]\end{aligned} \tag{B.10}$$

By assuming again u and v to be both stationary with zero-mean and independent of x, the next relationship holds

$$\begin{aligned}\left[\mathbf{C}_{yy}\right]_{pq} = &\left[\mathbf{C}_{xx}\right]_{pq} + E\big[z[p]z[q]\big]\,E\big[u[p]u[q]\big] \\ &+ E[v[p]v[q]]E\big[z[p]\big]\,E\big[u[p]v[q]\big] \\ &+ E\big[v[p]u[q]\big]\,E\big[z[q]\big]\end{aligned} \tag{B.11}$$

so that, if u and v are each other independent, the final outcome is

$$\left[\mathbf{C}_{yy}\right]_{pq} = \left[\mathbf{C}_{xx}\right]_{pq} + \left[\mathbf{C}_{zz}\right]_{pq}\left[\mathbf{C}_{uu}\right]_{pq} + \left[\mathbf{C}_{vv}\right]_{pq} \tag{B.12}$$

where, again, the right hand side does not depend on y, but only on x, u and v and each term depends only on one of these signal.

In order to obtain these results the following hypothesis were made on each contributor to the input signal y:

- u is supposed to be stationary with zero mean and independent of v and x;
- v is supposed to be stationary with zero mean and independent of u and x;
- x is supposed to be independent of both u and v, and its first derivative z is supposed to be well approximated by the convolution of x itself with an appropriate order derivative FIR filter k.

B.2 WAVELET-DOMAIN RELATIONSHIPS DERIVATION

As it was done for the time-domain in Section B.1, similar relationship can be derived in the Wavelet domain. Moreover, since Wavelet filters are involved and their coefficients are known quantities, they should be kept separate them from any other term whenever useful. Hypothesis made on both noises u and v as well as on x during the derivation of such relationships will be summarized at the end of this section.

Before starting, it's worth remembering that the SWT coefficients of the signal x for the m-th octave are obtained in the following way:

$$x_m[k] = \sum_i h_m[i]\, x[k-i] \tag{B.13}$$

where h_m is the equivalent Wavelet filter for the m-th octave.

Now, considering $E[y_m[k]]$, due to the linearity of the convolution operator embedded in the SWT, it is possible to rewrite it as

$$E[y_m[k]] = E\left[\sum_i h_m[i]\, y[k-i]\right] = \sum_i h_m[i]\, E[y[k-i]] \tag{B.14}$$

where the same hypotheses that led to (B.3) are assumed in order to substitute the expectation on $y[n]$ with an expectation on $x[n]$, thus obtaining

$$E[y_m[k]] = \sum_i h_m[i]\, E[x[k-i]] = E[x_m[k]] \tag{B.15}$$

which in turn allow to substitute any expectation on $x_m[k]$ with an expectation on $y_m[k]$ and vice-versa.

The linearity of the SWT can be used as well to decompose $y_m[k]$ as

$$\begin{aligned} y_m[k] &= \sum_i h_m[i]\,(x[k-i] + z[k-i]u[k-i] + v[k-i]) \\ &= x_m[q] + (zu)_m[k] + v_m[q] \end{aligned} \tag{B.16}$$

where

$$(zu)_m[k] = \sum_i h_m[i] z[k-i] u[k-i] \tag{B.17}$$

Equipped with these relationships, $\mathbf{C}_{x_m y_m}$ is now considered: by definition

$$\left[\mathbf{C}_{x_m y_m}\right]_{pq} = E[(x_m[p] - E[x_m[p]])(y_m[q] - E[y_m[q]])] \tag{B.18}$$

By using both (B.15) and (B.16) the following is obtained

$$\begin{aligned}\left[\mathbf{C}_{x_m y_m}\right]_{pq} = E\big[&(x_m[p] - E[x_m[p]]) \\ &\cdot (x_m[q] + (zu)_m[q] + v_m[q] - E[x_m[q]])\big]\end{aligned} \tag{B.19}$$

which in turn can be written as

$$\begin{aligned}\left[\mathbf{C}_{x_m y_m}\right]_{pq} = \left[\mathbf{C}_{x_m x_m}\right]_{pq} &+ E\big[x_m[p](zu)_m[q]\big] \\ &+ E\big[x_m[p]\big]\, E\big[(zu)_m[q] + v_m[q]\big] \\ &+ E\big[x_m[p] v_m[q]\big]\end{aligned} \tag{B.20}$$

As $(zu)_m[q] + v_m[q] = y_m[q] - x_m[q]$, making use of (B.15) the last term in (B.20) can be eliminated, leaving only two terms: by expanding the SWT transform and assuming v to be independent of x, $E[x_m[p] v_m[q]]$ becomes

$$E[x_m[p] v_m[q]] = \sum_{ij} h_m[i]\, h_m[j] E[x[p-i]] E[v[q-j]] \tag{B.21}$$

which is zero if v is assumed to be stationary with zero mean. Expanding again the SWT but assuming this time u independent of x and recalling the definition of z, $E[x_m[p](zu)_m[q]]$ becomes

$$\begin{aligned}E[x_m[p](zu)_m[q]] = \sum_{ijr} &h_m[i]\, h_m[j]\, f[r] \\ &\cdot E[x[p-i] x[q-j-r]]\, E[u[q-j]]\end{aligned} \tag{B.22}$$

which in turn is zero if u is assumed to be stationary with zero mean. Much similarly to (B.7), only remaining terms are

$$\left[\mathbf{C}_{x_m y_m}\right]_{pq} = \left[\mathbf{C}_{x_m x_m}\right]_{pq} \tag{B.23}$$

where the right hand side does not depend on y.

Finally, examining $\mathbf{C}_{y_m y_m}$, by definition the following holds

$$\left[\mathbf{C}_{y_m y_m}\right]_{pq} = E[(y_m[p] - E[y_m[p]])(y_m[q] - E[y_m[q]])] \tag{B.24}$$

from which, expanding the term $y_m[p]$ and plugging in (B.15), the next equation is obtained

$$\begin{aligned}\left[\mathbf{C}_{y_m y_m}\right]_{pq} = E\big[&(x_m[p] + (zu)_m[p] + v_m[p] - E[x_m[p]]) \\ &\cdot (x_m[q] + (zu)_m[q] + v_m[q] - E[x_m[q]])\big]\end{aligned} \tag{B.25}$$

Next, exploiting again the linearity of the expectation operator and splitting up (B.25) into

$$\begin{aligned}&\left[\mathbf{C}_{y_m y_m}\right]_{pq} = \\ &\quad = E\big[(x_m[p] - E[x_m[p]])(x_m[q] - E[x_m[q]])\big] \\ &\quad + E\big[(x_m[p] - E[x_m[p]])((zu)_m[q] + v_m[q])\big] \\ &\quad + E\big[((zu)_m[p] + v_m[p])(x_m[q] - E[x_m[q]])\big] \\ &\quad + E\big[((zu)_m[p] + v_m[p])((zu)_m[q] + v_m[q])\big]\end{aligned} \tag{B.26}$$

By assuming again u and v to be both stationary with zero mean and independent of x, the next relationship holds

$$\mathbf{C}_{y_m y_m} = \mathbf{C}_{x_m x_m} + \mathcal{C}_m^{(1)} + \mathcal{C}_m^{(2)} + \mathcal{C}_m^{(3)} + \left(\mathcal{C}_m^{(3)}\right)^T \tag{B.27}$$

where

$$\begin{aligned}\left[\mathcal{C}_m^{(1)}\right]_{pq} &= E[v_m[p]\, v_m[q]] \\ \left[\mathcal{C}_m^{(2)}\right]_{pq} &= E[(zu)_m[p]\, (zu)_m[q]] \\ \left[\mathcal{C}_m^{(3)}\right]_{pq} &= E[(zu)_m[p]\, v_m[q]]\end{aligned}$$

Expanding the SWT and recalling the definition of z, if u is assumed to be independent of v, the last two terms of (B.27) can be eliminated as $E[(zu)_m[p]\, v_m[q]]$ becomes

$$\begin{aligned}\left[\mathcal{C}_m^{(3)}\right]_{pq} = \sum_{ijr} &h_m[i]\, h_m[j]\, f[r]\, E[x[p - i - r]] \\ &\cdot E[u[p - i]]\, E[v[q - j]]\end{aligned} \tag{B.28}$$

which is zero since u and v were supposed to be both stationary with zero-mean.

In turn, expanding the SWT in the third term of (B.27) and adjusting the summation indexes this expression is obtained

$$\left[\mathcal{C}_m^{(1)}\right]_{pq} = \sum_{ij} h_m[p-i]\, h_m[q-j]\, E[v[i]\, v[j]] \tag{B.29}$$

which, since v is supposed to be stationary with zero mean, can be written as

$$\left[\mathcal{C}_m^{(1)}\right]_{pq} = \sum_{ij} H_m[p,q,i,j]\, \left[\mathbf{C}_{vv}\right]_{ij} \tag{B.30}$$

where $\left[\mathbf{C}_{vv}\right]$ is the autocovariance matrix of v and $H_m[p,q,i,j] = h_m[p-i]\, h_m[q-j]$.

Operating in the same way on the second term of (B.27), it is first obtained

$$\left[\mathcal{C}_m^{(2)}\right]_{pq} = \sum_{ijrs} h_m[p-i]\, h_m[q-j] f[i-r]\, f[j-s] \cdot E[x[r]\, x[s]]\, E[u[i]\, u[j]] \tag{B.31}$$

then, since u is supposed to be stationary with zero mean, the last term can be rewritten as

$$\left[\mathcal{C}_m^{(2)}\right]_{pq} = \sum_{ij} A_m[p,q,i,j,r,s]\, \left[\mathbf{R}_{xx}\right]_{rs}\, \left[\mathbf{C}_{uu}\right]_{ij} \tag{B.32}$$

where $\left[\mathbf{C}_{uu}\right]$ is the autocovariance matrix of u, $\left[\mathbf{R}_{xx}\right]$ is the autocorrelation matrix of x, and $A_m[p,q,i,j,r,s] = h_m[p-i]\, h_m[q-j]\, k[i-r]\, k[j-s]$.

Finally, (B.27) can be rewritten as

$$\mathbf{C}_{y_m y_m} = \mathbf{C}_{x_m x_m} + \mathcal{C}_m^{(1)} + \mathcal{C}_m^{(2)} \tag{B.33}$$

where, thanks to (B.30) and (B.32), the right hand side does not depend on y, but only on x, u and v and each term depends only on one of these signal, just as (B.12). Moreover, with the only exception of $\mathbf{C}_{x_m x_m}$, any information regarding the Wavelet filters h_m and the derivative filter k_m is contained in the terms A_m and H_m.

It's worth noting that so far nothing more than the same hypothesis made in the time domain were used. By making some further assumptions, it is possible to derive two other very interesting relationship about the second and third term of (B.33).

In fact, if both u and v are assumed to be uncorrelated other than stationary, their covariance matrices become diagonal, with identical terms on the main diagonal. So, calling σ_v the standard deviation of v, the next relationship is obtained

$$\left[\mathcal{C}_m^{(1)}\right]_{pq} = \sigma_v^2 \sum_i h_m[p-i]\, h_m[q-i] \tag{B.34}$$

where the summation in j has been removed due to the diagonal nature of $\mathbf{C}_{vv}$.

Now, by adjusting the only left summation index, this equality holds:

$$\begin{aligned}\left[\mathcal{C}_m^{(1)}\right]_{pq} &= \sigma_v^2 \sum_i h_m[i]\, h_m[i+p-q] \\ &= \sigma_v^2 R_{h_m h_m}[p-q] \end{aligned} \tag{B.35}$$

where $R_{h_m h_m}[i]$ is the value assumed by autocorrelation function of the filter h_m at the i-th lag. Calling σ_u the standard deviation of u and following the same steps as before, the following is obtained

$$\begin{aligned}\left[\mathcal{C}_m^{(2)}\right]_{pq} = \sigma_u^2 \sum_{irs} & h_m[i]\, h_m[i+p-q]\, f[r]\, f[s] \\ & \cdot E[x[q-i-r]x[q-i-s]] \end{aligned} \tag{B.36}$$

If x is supposed to be stationary on both the support of h_m and f, the last relationship can be rewritten as

$$\begin{aligned}\left[\mathcal{C}_m^{(2)}\right]_{pq} = \sigma_u^2 \sum_{irs} & h_m[i]\, h_m[i+p-q]\, f[r]\, f[s] \\ & \cdot E[x[q]x[q+r-s]] \end{aligned} \tag{B.37}$$

where the expectation term is nothing more than the autocorrelation function of x centered on the sample n for

the lag $r - s$. Adjusting some summation indexes, the next equality is obtained

$$\begin{aligned}\left[\mathcal{C}_m^{(2)}\right]_{pq} &= \sigma_u^2 \left(\sum_i h_m[i]\, h_m[i+p-q] \right) \\ &\quad \cdot \sum_d \left(\sum_s k[s]\, k[s+d] \right) R_{xx}^{(n)}[d] \\ &= \sigma_u^2 \, R_{h_m h_m}[p-q] \sum_d R_{ff}[d]\, R_{xx}^{(q)}[d] \end{aligned} \quad \text{(B.38)}$$

where $R_{h_m h_m}[i]$ is the value assumed by autocorrelation function of the filter h_m at the i-th lag and $R_{ff}[i]$ is its equivalent for the filter f. Examining equation (B.35) and (B.38), it's even more evident how the contributes of each single component of the input signal y can be separated.

In conclusion, the hypothesis made to obtain these relationships can be summarized as follows:

- u is supposed to be stationary with zero mean and independent of v and x;
- v is supposed to be stationary with zero mean and independent of u and x;
- x is supposed to be independent of both u and v and stationary over the support of the equivalent wavelet filters h_m and the model filter f.

PUBLICATIONS

[P-1] A. Marcianesi, R. Padovani, N. Speciale, N. Testoni, and G. Masetti, "Fpga implementation of QCWT based algorithm for filtering low SNR signals," in *Proceedings of the 2003 IEEE/EURASIP Workshop on Nonlinear Signal and Image Processing.*, Grado, Italy, June 2003.

[P-2] N. Testoni, L. D. Marchi, N. Speciale, and G. Masetti, "Real-time classification methods for biological tissues," in *Proceedings of the 2004 International Symposium on Nonlinear Theory and Applications (NOLTA 2004)*, Fukuoka, Japan, December 2004, pp. 537–540.

[P-3] S. Maggio, N. Testoni, L. D. Marchi, N. Speciale, and G. Masetti, "Wavelet-based deconvolution algorithms applied to ultrasound images," in *Proceedings of the 5th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, St Paul's Bay, Malta, September 2005, pp. 77–81.

[P-4] ——, "Ultrasound images enhancement by means of deconvolution algorithms in the wavelet domain," *WSEAS Transactions on Systems*, vol. 11, no. 4, pp. 1958–1965, November 2005.

[P-5] N. Testoni, M. Cisterni, and E. Franchi, "A regular modular architecture for pipelined binary tree multipliers based on a SOG structure," in *Proceedings of the 2005 IEEE Ph. D. Research in Microelectronics and Electronics (PRIME2005)*, Lausanne, Switzerland, July 2005, pp. 44–47.

[P-6] L. D. Marchi, N. Testoni, and N. Speciale, "Prostate tissue characterization via ultrasound speckle statistics," in *Proceedings of the 6th IEEE International Symposium on Signal Processing & Information Technology (ISSPIT 2006)*, Vancouver, Canada, August 2006, pp. 208–211.

[P-7] L. D. Marchi, A. Palladini, N. Testoni, and N.Speciale, "Blurred ultrasonic images as ISI-affected signals: Joint tissue response, estimation and channel tracking in the proposed paradigm," in *Proceedings of the 2007 IEEE International Ultrasonic Symposium*, New York, USA, October 2007.

[P-8] A. Palladini, N. Testoni, L. D. Marchi, and N. Speciale, "A reduced complexity estimation algorithm for ultrasound images de-blurring," in *Proceedings of the 20th IEEE International Symposium on Compupter Based Medical Systems (CBMS07)*, Maribor, Slovenia, June 2007, pp. 275–280.

[P-9] ——, "ML estimation for acoustical image deblurring," in *Proceedings of the 29th International Symposium on Acoustical Imaging (AI29)*, Kanagawa, Japan, 2007 2007.

[P-10] N. Testoni, N. Speciale, A. Ridolfi, and C. Pouzat, "Adaptive wavelet-based signal dejittering," in *Proceedings of the 2007 IEEE Ph. D. Research in Microelectronics and Electronics (PRIME2007)*, Bordeaux, France, June 2007, pp. 257–260.

BIBLIOGRAPHY

[1] J. Curie, "Développement par pression de l'électricite polaire dans les cristaux hémièdres à faces inclinées," *C.R. Acad. Sci. (Paris)*, vol. 91, p. 294, 1880. (Cited on page 10.)

[2] K. Dussik, "On the possibility of using ultrasound waves as a diagnostic aid," *Neurol. Psychiat.*, vol. 174, pp. 153–168, 1942. (Cited on page 10.)

[3] G. D. Ludwig, "The velocity of sound through tissues and the acoustic impedance of tissues," *Journal of the Acoustical Society of America*, vol. 22, no. 6, pp. 862–866, November 1950. (Cited on pages 10 and 14.)

[4] J. J. Wild and J. M. Reid, "Application of echo-ranging techniques to the determination of structure of biological tissues," *Science*, vol. 115, pp. 226–230, 1952. (Cited on page 10.)

[5] R. Neelamani, C. Hyeokho, and R. Baraniuk, "Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems," *IEEE Trans. Signal Processing*, vol. 52, no. 2, pp. 418–433, February 2004. (Cited on pages 13, 58, and 59.)

[6] D. Adam and O. Michailovich, "Blind deconvolution of ultrasound sequences using nonparametric local polynomial estimates of the pulse," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 2, pp. 118–131, February 2002. (Cited on pages 13, 47, and 51.)

[7] T. Taxt and J. Strand, "Two-dimensional noise-robust blind deconvolution of ultrasound images," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 48, no. 4, pp. 861–866, July 2001. (Cited on page 13.)

[8] T. Taxt, "Three-dimensional blind deconvolution of ultrasound images," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 48, no. 4, pp. 867–871, July 2001. (Cited on page 13.)

[9] J. Jensen, "Ultrasound imaging and its modeling," in Imaging of Complex Media with Acoustic and Seismic Waves, Topics in Applied Physics. Springer, 2000. (Cited on page 13.)

[10] C. Chen, W.-L. Hsu, and S.-K. Sin, "A comparison of wavelet deconvolution techniques for ultrasonic NDT," in *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '88)*, vol. 2, April 1988, pp. 867–870. (Cited on page 15.)

[11] M. F. Insana, R. F. Wagner, D. G. Brown, and T. J. Hall, "Describing small-scale structure in random media using pulse-echo ultrasound," *The Journal of the Acoustical Society of America*, vol. 87, no. 1, pp. 179–192, January 1990. (Cited on pages 16, 17, and 58.)

[12] F. Cohen, G. Georgiou, and E. Halpern, "Wold decomposition of the backscatter echo in ultrasound images of soft tissue organs," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 44, no. 2, pp. 460–472, March 1997. (Cited on page 17.)

[13] G. Treece, R. Prager, and A. Gee, "Ultrasound attenuation measurement in the presence of scatterer variation for reduction of shadowing and enhancement," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 52, no. 12, pp. 2346–2360, Dec. 2005. (Cited on page 17.)

[14] S. Flax, N. J. Pelc, G. Glover, F. Gutmann, and M. McLachlan, "Spectral characterization and attenuation measurements in ultrasound," *Ultrasonic Imaging*, vol. 5, no. 2, pp. 95–116, April 1983. (Cited on page 17.)

[15] R. Caton, "The electric currents of the brain," *British Medicine Journal*, vol. 2, p. 278, 1875. (Cited on page 20.)

[16] A. Hodgkin, A. Huxley, and B. Katz, "Measurement of current-voltage relations in the membrane of the giant axon of loligo," *Journal of Physiology*, vol. 116, pp. 424–448, 1952. (Cited on page 20.)

[17] A. Hodgkin and A. Huxley, "A quantitative description of membrane current and its application to conduction

and excitation in nerve," *Journal of Physiology*, vol. 117, no. 4, pp. 500–544, 1952. (Cited on page 20.)

[18] M. S. Lewickiy, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, pp. 53–78, 1998. (Cited on pages 22, 25, 29, and 150.)

[19] C. M. Gray, "Synchronous oscillations in neuronal systems: Mechanisms and functions," *Journal of Computational Neuroscience*, vol. 1, no. 1-2, pp. 11–38, June 1994. (Cited on page 22.)

[20] C. Bédard, H. Kröger, and A. Destexhey, "Modeling extracellular field potentials and the frequency-filtering properties of extracellular space," *Biophysical Journal*, vol. 86, pp. 1829–1842, March 2004. (Cited on page 22.)

[21] S. N. Baker, J. M. Kilner, E. M. Pinches, and R. N. Lemon, "The role of synchrony and oscillations in the motor output," *Experimental Brain Research*, vol. 128, no. 1-2, pp. 109–117, September 1999. (Cited on page 22.)

[22] N. Klauke, G. L. Smith, and J. Cooper, "Extracellular recordings of field potentials from single cardiomyocytes," *Biophysical Journal*, vol. 91, no. 7, pp. 2543–2551, October 2006. (Cited on page 23.)

[23] G. T. A. Kovacs, "Electronic sensors with living cellular components," *Proc. IEEE*, vol. 91, no. 6, pp. 915–929, June 2003. (Cited on page 23.)

[24] J. McNames, "Overview of spike discrimination and spike sorting," Biomedical Signal Processing Laboratory, E&C Engineering, Portland State University, Seminar Talk, 2001. (Cited on page 25.)

[25] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Computation*, vol. 16, pp. 1661–1687, 2004. (Cited on page 25.)

[26] M. I. Chelaru and M. S. Jog, "Spike source localization with tetrodes," *Journal of Neuroscience Methods*, vol. 142, pp. 305–315, 2005. (Cited on page 25.)

[27] M. Z. Othman, M. M. Shaker, and M. F. Abdullah, "Eeg spikes detection, sorting, and localization," *Transactions On Engineering, Computing And Technology*, vol. 9, pp. 205–208, November 2005. (Cited on pages 25 and 27.)

[28] A. Antoniadis, J. Bigot, and T. Sapatinas, "Wavelet estimators in nonparametric regression: A comparative simulation study," *Journal of Statistical Software*, vol. 6, no. 6, pp. 1–83, 2001. (Cited on pages 27, 123, 124, 150, and 151.)

[29] R. Q. Quiroga, O. Sakowitz, Basar, and M. Schürmannd, "Wavelet transform in the analysis of the frequency composition of evoked potentials," *Brain Research Protocols*, vol. 8, pp. 16–24, 2001. (Cited on page 27.)

[30] K. H. Kim and S. J. Kim, "Neural spike sorting under nearly 0-db signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 10, pp. 1406–1411, October 2000. (Cited on page 27.)

[31] J. C. Letelier and P. P. Weber, "Spike sorting based on discrete wavelet transform coefficients," *Journal of Neuroscience Methods*, vol. 101, pp. 93–106, 2000. (Cited on page 27.)

[32] R. Vollgraf and K. Obermayer, "Improved optimal linear filters for the discrimination of multichannel waveform templates for spike-sorting applications," *IEEE Signal Processing Lett.*, vol. 13, no. 3, pp. 121–124, March 2006. (Cited on pages 27 and 150.)

[33] K. G. Oweiss and D. J. Anderson, "Spike sorting:a novel shift and amplitude invariant technique," *Neurocomputing*, vol. 44-46, pp. 1133–1139, 2002. (Cited on page 27.)

[34] W. Kester, Ed., *The Data Conversion Handbook*. Elsevier/Newnes, 2005. (Cited on page 28.)

[35] C. S. Pattichis and M. S. Pattichis, "Time-scale analysis of motor unit action potentials," *IEEE Trans. Biomed.*

Eng., vol. 46, no. 11, pp. 1320–1329, November 1999. (Cited on page 29.)

[36] Y. Meyer, *Ondelettes et opérateurs – I : Ondelettes*. Hermann, Editeurs des Sciences et des Arts, 1989. (Cited on page 31.)

[37] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inform. Theory*, vol. 36, no. 5, pp. 961–1005, September 1990. (Cited on page 31.)

[38] ——, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. XLI, pp. 909–996, 1988. (Cited on pages 31, 38, and 120.)

[39] S. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 2091–2110, December 1989. (Cited on page 31.)

[40] D. Colella and C. Heil, "Characterizations of scaling functions: Continuous solutions," *SIAM J. Matrix Anal. Appl.*, vol. 15, no. 2, pp. 496–518, April 1994. (Cited on pages 32, 38, and 39.)

[41] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Trans. Inform. Theory*, vol. 38, pp. 569–586, Mar. 1992. (Cited on pages 32, 33, and 36.)

[42] S. H. Maes, "Fast quasi-continuous wavelet algorithms for analysis and synthesis of one-dimensional signals," *SIAM Journal on Applied Mathematics*, vol. 57, no. 6, pp. 1763–1801, 1997. (Cited on pages 32, 33, and 37.)

[43] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms i." *Comm. Pure Appl. Math.*, vol. 44, no. 2, pp. 141–183, 1991. (Cited on pages 32, 33, 35, and 36.)

[44] I. Daubechies and J. C. Lagarias, "Two-scale difference equations II. local regularity, infinite products of matrices and fractals," *SIAM Journal on Mathematical*

Analysis, vol. 23, no. 4, pp. 1031–1079, 1992. (Cited on page 38.)

[45] A. Micchelli and H. Prautzsch, "Refinement and subdivision for spaces of integer translates of a compactly supported function," in *Pitman Research Notes In Mathematics Series*. Longman Publishing Group, White Plains, NY, USA, 1987, pp. 192–222. (Cited on page 38.)

[46] J. Jensen and S. Leeman, "Nonparametric estimation of ultrasound pulses," *IEEE Trans. Biomed. Eng.*, vol. 41, no. 10, pp. 929–936, October 1994. (Cited on page 47.)

[47] K. Kaaresen and E. Bolviken, "Blind deconvolution of ultrasonic traces accounting for pulse variance," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 46, no. 3, pp. 564–573, May 1999. (Cited on page 47.)

[48] T. Taxt and R. Jirik, "Superresolution of ultrasound images using the first and second harmonic signal," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 51, no. 2, pp. 163–175, February 2004. (Cited on page 47.)

[49] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," *Proc. IEEE*, vol. 65, no. 10, pp. 1428–1443, Oct. 1977. (Cited on page 47.)

[50] S.-C. Pei and S.-T. Lu, "Design of minimum-phase FIR digital filters by differential cepstrum," *IEEE Trans. Circuits Syst.*, vol. 33, no. 5, pp. 570–576, May 1986. (Cited on page 47.)

[51] S.-C. Pei and S.-B. Jaw, "Improved design of minimum-phase FIR digital filters by cepstrum and fast hartley transform," *Electronics Letters*, vol. 26, no. 14, pp. 1063–1064, July 1990. (Cited on page 47.)

[52] N. Damera-Venkata, B. Evans, and S. McCaslin, "Design of optimal minimum-phase digital FIR filters using discrete hilbert transforms," *IEEE Trans. Signal Processing*, vol. 48, pp. 1491–1495, 2000. (Cited on page 47.)

[53] I. Daubechies, *Ten Lectures on Wavelets*. SIAM, 1992. (Cited on page 50.)

[54] M. Izzetoglu, B. Onaral, and N. Bilgutay, "Wavelet domain least squares deconvolution for ultrasonic backscattered signals," in *Proceedings of the 22nd IEEE/EMBS International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, July 2000, pp. 321–324. (Cited on pages 58 and 111.)

[55] S. Wan, B. Raju, and M. Srinivasan, "Robust deconvolution of high-frequency ultrasound images using higher-order spectral analysis and wavelets," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 50, no. 10, pp. 1286–1295, October 2003. (Cited on page 58.)

[56] G. Wornell, *Signal Processing with Fractals: A Wavelet-Based Approach*. Prentice Hall, 1996. (Cited on pages 58, 59, and 111.)

[57] G. Georgiou and F. Cohen, "Statistical characterization of diffuse scattering in ultrasoundimages," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 45, no. 1, pp. 57–64, January 1998. (Cited on pages 58 and 111.)

[58] J. M. Girault, F. Ossant, A. Ouahabi, D. Kouame, and F. Patat, "Time-varying autoregressive spectral estimation for ultrasound attenuation in tissue characterization," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 45, no. 3, pp. 650–659, May 1998. (Cited on pages 58 and 111.)

[59] P. Abry, P. Goncalves, and P. Flandrin, "Wavelet-based spectral analysis of 1/f processes," in *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '93)*, vol. 3, April 1993, pp. 237–240. (Cited on page 58.)

[60] B. Pellegrini, "A general model of 1/f noise," *Microelectronics Reliability*, vol. 40, no. 11, pp. 1775–1780, November 2000. (Cited on page 58.)

[61] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Mag.*, vol. 14, no. 2, pp. 24–41, Mar. 1997. (Cited on page 60.)

[62] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*. John Wiley, New York, 1977. (Cited on page 60.)

[63] A. D. Hillery and R. T. Chin, "Iterative wiener filters for image restoration," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 39, no. 8, pp. 1892–1899, Aug. 1991. (Cited on page 60.)

[64] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1998. (Cited on page 61.)

[65] D. Donoho and I. Johnstone, "Threshold selection for wavelet shrinkage of noisy data," in *Proceedings of the 16th IEEE/EMBS International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, November 1994, pp. A24–A25. (Cited on pages 61 and 62.)

[66] D. Manolakis, *Statistical and Adaptive Signal Processing - Spectral Estimation*. Artech House, 2005. (Cited on pages 70 and 71.)

[67] J. A. Högbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomical and Astrphisical Supplement*, vol. 15, pp. 417–426, 1974. (Cited on page 74.)

[68] U. Abeyratne, A. Petropulu, J. Reid, T. Golas, E. Conant, and F. Forsberg, "Higher order versus second order statistics in ultrasound image deconvolution," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 44, no. 6, pp. 1409–1416, November 1997. (Cited on page 78.)

[69] C. Loizou, C. Pattichis, C. Christodoulou, R. Istepanian, M. Pantziaris, and A. Nicolaides, "Comparative evaluation of despeckle filtering in ultrasound imaging of the carotid artery," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 52, no. 10, pp. 1653–1669, October 2005. (Cited on page 78.)

[70] J. Tang, E. Peli, and S. Acton, "Image enhancement using a contrast measure in the compressed domain,"

IEEE Signal Processing Lett., vol. 10, no. 10, pp. 289–292, Oct. 2003. (Cited on page 79.)

[71] S. Haykin, *Adaptive Filter Theory, Fourth Edition*. Prentice Hall, 2002. (Cited on pages 84, 85, 86, 104, 106, and 107.)

[72] A. K. Barros, J. Principe, Y. Takeuchi, and N. Ohnishi, "Using non-linear even functions for error minimization in adaptive filters," *Neurocomputing*, vol. 70, no. 1-3, pp. 9–13, December 2006. (Cited on pages 94 and 95.)

[73] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family," *IEEE Trans. Inform. Theory*, vol. 30, no. 2, pp. 275–283, Mar. 1984. (Cited on page 94.)

[74] S. Sandeep Pradham and V. U. Reddy, "A new approach to subband adaptive filtering," *IEEE Trans. Signal Processing*, vol. 47, no. 3, pp. 655–664, Mar. 1999. (Cited on pages 98 and 101.)

[75] S. Kay, *Fundamentals of Statistical Processing, Volume I: Estimation Theory*. Prentice Hall, Englewood Cliffs, NJ, 1993. (Cited on pages 116 and 117.)

[76] J. Forney, G.D., "The viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, March 1973. (Cited on page 136.)

[77] C. Miller, B. R. Hunt, M. W. Marcellin, and M. A. Neifeld, "Image restoration with the viterbi algorithm," *Journal of the Optical Society of America A*, vol. 17, no. 2, pp. 265–275, February 2000. (Cited on page 136.)

[78] F. Argenti, G. Torricelli, and L. Alparone, "Signal-dependent noise removal in the undecimated wavelet domain," in *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, vol. 4, May 2002. (Cited on page 138.)

[79] A. Achim, A. Bezerianos, and P. Tsakalides, "Novel bayesian multiscale method for speckle removal in medical ultrasound images," *IEEE Trans. Med. Imag.*,

vol. 20, no. 8, pp. 772–783, August 2001. (Cited on page 138.)

[80] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. Elsevier, Academic Press, 2003. (Cited on pages 144 and 145.)

[81] A. R. Webb, *Statistical Pattern Recognition, Second Edition*. Wiley, 2002. (Cited on pages 144 and 145.)

[82] T. Poggio and S. Smale, "The mathematics of learning: Dealing with data," *Notices of the AMS*, vol. 50, no. 5, pp. 537–544, May 2003. (Cited on page 145.)

[83] V. N. Vapnik, *Statistical Learning Theory*. John Wiley and Sons, New York, 1998. (Cited on page 145.)

[84] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, April 2005. (Cited on page 145.)

[85] C. Pouzat, *Methods and Models in Neurophysics, Les Houches 2003 Summer School*. Elsevier, 2005, ch. Technique(s) for Spike-Sorting, pp. 729–786. (Cited on pages 147, 148, and 149.)

[86] A. Aldroubi, C. Cabrelli, D. Hardin, and U. Molter, "Optimal shift invariant spaces and their parseval generators," in *Proceedings of the 2005 International Conference "Modern Methods of Time-Frequency Analysis"*, 2005. (Cited on pages 152 and 153.)

[87] A. Aldroubi, C. Cabrelli, and U. Molter, *Harmonic Analysis and Applications*, ser. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2006, ch. Learning the Right Model from the Data, pp. 325–333. (Cited on pages 152 and 153.)

[88] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications On Pure & Applied Mathematics*, vol. 59, pp. 1207–1223, 2005. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:math/0503066 (Cited on page 154.)

[89] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n," May 2006, to appear in Annals of Statistics. (Cited on page 154.)

[90] S. Boyd and L. Vandernberghe, *Convex Optimization*. Cambridge University Press, 2006. (Cited on page 155.)