UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECŢIEI SOCIALE ŞI
PERSOANELOR VÂRSTNICE
AMPOSDRU

Fondul Social European
POSDRU 2007-2013

Instrumente Structurale
2007-2013

MINISTERUL
EDUCAȚIEI
NAȚIONALE
OIPOSDRU

Universitatea
POLITEHNICA
din București

FONDUL SOCIAL EUROPEAN
Investeşte în oameni!
Programul Operaţional Sectorial pentru Dezvoltarea Resurselor Umane 2007 – 2013
Proiect POSDRU/107/1.5/S/76909 – *Valorificarea capitalului uman din cercetare prin burse doctorale (ValueDoc)*

UNIVERSITATEA **POLITEHNICA** DIN BUCUREŞTI

Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei
Departamentul Dispozitive, Circuite şi Aparate Electronice

Nr. Decizie Senat 225 din 27.09.2013

TEZĂ DE DOCTORAT

Selecția semnalelor și compensarea automată a întârzierii în mediul VoIP

Best Signal Selection with Automatic Delay Compensation in VoIP Environment

Autor: Ing. Radu-Sebastian MARINESCU

COMISIA DE DOCTORAT

Preşedinte	Prof. dr. ing. Gheorghe BREZEANU	de la	Universitatea Politehnica din București
Conducător de doctorat	Prof. dr. ing. Corneliu BURILEANU	de la	Universitatea Politehnica din București
Referent	Prof. dr. ing. Corneliu RUSU	de la	Universitatea Tehnică din Cluj Napoca
Referent	Prof. dr. ing. Ioan NICOLAESCU	de la	Academia Tehnică Militară București
Referent	Prof. dr. ing. Cristian NEGRESCU	de la	Universitatea Politehnica din București

Bucureşti - 2013

Acknowledgments

I am deeply thankful to my PhD coordinator, prof.dr.eng. Corneliu Burileanu, for giving me the opportunity to work on this thesis. I was impressed by his positive attitude, constantly encouraging me, supporting my ideas and believing in my work. I am also grateful for the fact that I was given the opportunity to work in his *Speech and Dialogue Research Laboratory* team, as I was able to enlarge my knowledge and experience in the field of digital signal processing.

I would also like to thank prof.dr.eng. Corneliu Rusu, prof.dr.eng Ioan Nicolaescu and prof.dr.eng Cristian Negrescu. As members of the assessment committee, they provided useful comments and remarks regarding my thesis, which helped me improve it.

I am grateful to my colleagues, dr.eng. Andi Buzo and dr.eng. Horia Cucu, from the *Speech and Dialogue Research Laboratory*, for the fruitful cooperation, the interesting ideas and recommendations provided throughout the drafting of my PhD thesis.

An essential role in the development of this thesis was played by my other colleagues from *Rohde & Schwarz Topex.* I would like to thank them for suggesting me the topic of this thesis, for the high-tech equipment and support, and for the amazing mood with which they surrounded me.

I could not forget my friends who were extremely considerate. Thus, I want to thank them for their forbearance.

Much gratitude goes to my whole family, for their support and faith in my work. Thanks to them the presentation given today is now possible. Last but not least, a special thanks goes to my wife Ana-Silvia, for her patience, support and understanding.

Contents

Figure List

Table List

Abbreviations

3SFM – Smoothed Sub-band Spectral Flatness Measure

AED – Adaptive Eigenvalue Decomposition

ALSFM – Accumulated Long-Term Spectral Flatness Measure

AMDF – Average Magnitude Difference Function

ASAE – Accumulated Speech Activity Envelope

ASDF – Average Square Difference Function

ATC – Air traffic control

ATM – Air traffic management

BSS – Best signal selection

CC – Cross-Correlation

DFT – Discrete Fourier Transform

DWT – Discrete Wavelet Transform

eBSS – Enhanced Best Signal Selection

ETSI – European Telecommunications Standards Institute

FFT – Fast Fourier Transform

GCC – General Cross-Correlation

IFFT – Inverse Fast Fourier Transform

ICT – Information and Communications Technologies

ITU – International Telecommunication Union

ITU-T – ITU Telecommunication Standardization Sector

LMS – Least Mean Square

LPC – Linear Prediction Coefficients

LSFM – Long-Term Spectral Flatness Measure

ML – Maximum Likelihood

MMSE – Minimum Mean Square Error

MOS – Mean Opinion Score

MS – Minimum Statistics

MVSS – Maximum Values of Sub-band SNR

PESQ – Perceptual Evaluation of Speech Quality

PHAT – Phase Transform

POLQA – Perceptual Objective Listening Quality Assessment

QoS – Quality of Service

RLS – Recursive Least Square

SAE – Speech Activity Envelope

SNR – Signal-to-Noise Ratio

TEO – Teager Energy Operator

TDE – Time Delay Estimation

TDM – Time-Division Multiplexing

VAD – Voice Activity Detection

VCS – Voice Communication System

VoIP – Voice over Internet Protocol

1 INTRODUCTION

1.1 BEST SIGNAL SELECTION OVERVIEW

In the last decades, air traffic spread more and more in the world, connecting more and more places. At the same time, the need to manage all the flights correctly and securely increased. Air traffic authorities imposed and updated several standards for the air traffic management (ATM) system, keeping in pace with the growing traffic flow. To achieve this, special voice communication systems (VCS) were developed. They ensure the communication between the pilots and the operators from the ground control centers. When a communication is initiated between the aircraft's pilot and the ground air traffic control operator, various systems are used. The pilot speaks through the aircraft's radio station and the signal is received by several ground radio stations. Then, the signal from each ground radio station arrives on different paths to the control center. Here one of the received signals is played to the operator. Ideally, this should be the one which is the clearest and offers the highest intelligibility. This is the equivalent of the *Best Signal Selection* (BSS) for the received signals, which now is achieved through special signal processing algorithms.

The BSS has some issues. First, the decision has to be made very fast, in tenths of seconds. This is imposed by the ED-136 standard, which states that the maximum delay in a

ATM-VCS between the speech initiation and speech play should not be more than 300ms. But this represents the overall system time, which includes delays for radio stations activations, signal transportation and VCS processing, leaving an even shorter time for signal processing.

Secondly, the number of radio stations is important. More receiving stations on the ground means more input signals in the VCS. Their number, as stated in [ED-136], is between 2 and 7. Because of security reasons, redundancy should be provided, thus it cannot be only one ground radio station.

Another issue regarding the receiving quality of the ground stations is represented by their location. Because they have to cover a large area, they are placed in different regions, usually at distances of at least 200 km between each other. Because of this spread, they can easily be affected by different weather conditions. Bad weather affects the amplitude modulated radio signal, which is used in air traffic communications. Therefore, for a period of time, a ground station can receive noisy signals in stormy weather, while on sunny days provides clean signals. Thus, the alternation of the best signal selection is influenced by the weather conditions.

Aircraft movement affects also the quality of the received radio signal. When an aircraft approaches a ground radio station, this station, generally, receives the best signal. But, after a short time, the position of the aircraft will change and another ground radio station will be closer. By now, the new closer ground station will receive the best signal in most cases, leading to a new change in best signal selection.

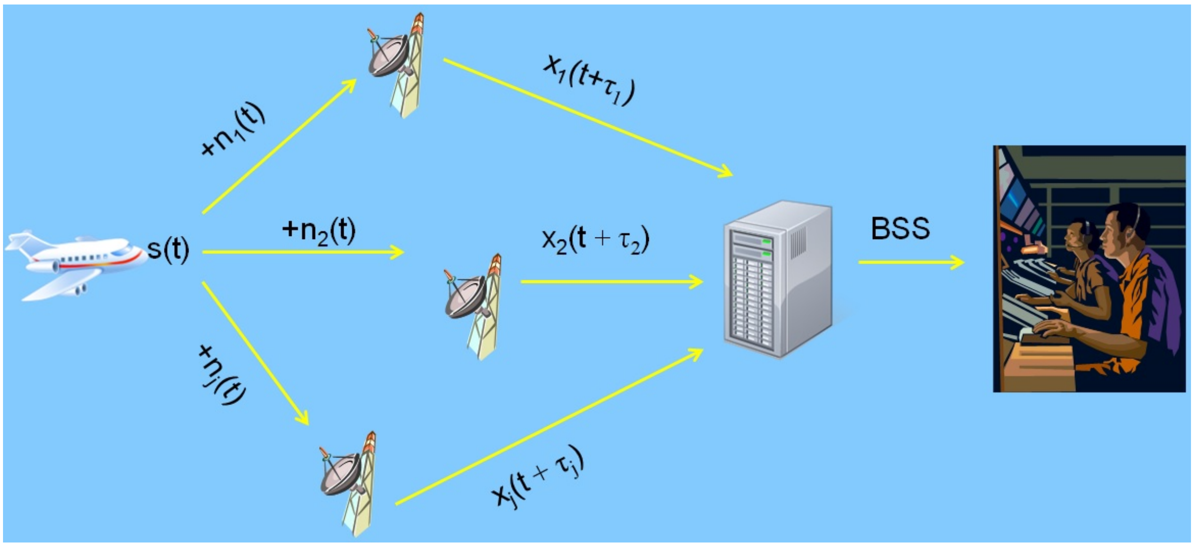


Figure 1.1 – Best Signal Selection Overview

A solution to the above issue could be to predict the position of the aircraft and then the station which will receive the best signal. But, this idea does not work for several aircrafts at the same time, because, for security reasons, any aircraft can initiate a communication anytime. Due to this, the communications' order cannot be guaranteed and the previous selections cannot be used efficiently for a proper prediction of the next best signal selection.

All the above presented issues describe largely the problem of best signal selection in air traffic control and management systems. In the next sub-sections the readers will be introduced to a deeper level of some solutions.

1.2 TIME DELAY ESTIMATION

The majority VCS use the time-domain multiplexing (TDM) technique to communicate between the command center and ground radio stations. This solution is in use for at least 20 years and during this time it proved to have many advantages. The technology is well known and easy to be implemented by the telecommunication operators, especially in locations that already have communication copper lines installed. Another important advantage of the TDM technique is the fixed and relatively small delay introduced by a switching level. In ATM systems the switching level delay is usually 125µs, and the overall delay is usually less than 1 ms. It is considered to be relatively small because it is compared with a syllable period. Also, because this delay is stable, the TDM technique practically does not affect the speech intelligibility.

On the other hand, a recent solution uses the VoIP protocol to connect ground radio stations with the command center. It is expanding because it is cheaper than the TDM technique and can also provide new multimedia capabilities. But, communication over the IP network is considered less reliable in contrast to the circuit-switched public telephone network, as it does not provide a network-based mechanism to ensure that data packets are not lost, and are delivered in sequential order [Sanchez, 2013]. Therefore, VoIP implementations may face problems mitigating latency, packet loss and jitter [Prabhakar, 2005]. Thus, in ATM-VCS based on VoIP, the delays introduced by the network are not stable and in some cases could be much longer than those from TDM networks.

Because of these, despite the fact that more ground radio stations receive the signal coming from the pilot at the same time, at the command center they can arrive with different delays, some times larger than 30ms. In this situation, for unaligned signals, the best signal selection algorithm cannot offer reliable results. Thus, prior to this, the signals received through VoIP environments must be realigned. As it can be anticipated, this represents an important issue which will be studied and analyzed in this work.

1.3 Voice Activity Detection

Voice activity detection (VAD) algorithms try to discriminate speech segments from noisy input signals. They are widely used in speech and communications applications, including ATM-VCS. In this kind of systems, besides speech/non-speech decision, their role increases in VoIP environments, being able to save important bandwidth, as non-speech segments are stopped from transmission in the network. For this, they usually look for speech features into the signal and then assign a VAD score. After this step a speech/non-speech decision is taken, based on the specific techniques of each method.

Generally, speech signals with better intelligibility lead to better VAD scores. This suggests including VAD in the best signal selection analysis. Thus, it is important to characterize the correlation between the VAD score and speech intelligibility. Therefore, it is necessary to perform a study of this topic, in order to obtain an optimal BSS solution for ATM-VCS.

1.4 Speech Enhancement and Signal-to-Noise Ratio Estimation

Speech enhancement algorithms try to improve speech signals affected by noise. Depending on the applications, speech enhancement can use one, two or multiple channels. While the single channel methods are based only on information from one channel, the multichannel methods can speculate the distribution of the microphones/sensors in order to obtain more information regarding the background noise.

The SNR estimation is the first step in various single or multi-channel speech enhancement. Its role is to estimate the background noise. Then, the additive background noise is subtracted from the noisy signal by different techniques. Depending on the applications, the processed signal will have a higher SNR and/or better speech intelligibility.

The SNR estimation and speech enhancement are important for this thesis because they support the idea of an enhanced BSS (eBSS). Using a multi-channel approach we can reduce the additive transport noise from the received signals. Thus, by subtracting the noise from the BSS, the processed eBSS should lead to better speech intelligibility.

1.5 Thesis Objectives and Outline

After a short introduction regarding the ATM-VCS I will now describe the main objective of this thesis – the development of a solution for best signal selection in a VoIP environment. Thus, the system has to integrate the new work in order to provide highly accurate selections for real time conditions. Besides this main goal, another aim was to provide an enhanced best signal selection (eBSS), which should make use of all the input signals, to offer a more intelligible output than the BSS. To achieve this, several specific objectives were addressed:

a)Develop a fast and accurate time delay estimation method, based on generalized cross-correlation. This is needed to align incoming signals before analyzing their intelligibility. Also, TDE is the first step in further speech enhancement processing for eBSS;

b)Implement an optimum VAD algorithm as an important part of the best signal selection process;

c)Search for an optimum signal-to-noise estimators which could be used for multichannel speech enhancement;

The thesis is organized around seven chapters, as follows:

Chapter 1 starts with an introduction to the *best signal selection* problem of the air traffic management voice communication systems. Then it summarizes the main tasks needed for obtaining an effective BSS and eBSS, and highlights the difficulties encountered by the system designer. Finally, chapter 1 describes the main objectives and outlines this thesis.

Chapter 2 presents the time delay estimation problem and different state-of-the art approaches to solve it. Firstly the general *adaptive filtering* solution is presented, with two basic variants, *least mean square* (LMS) and *recursive least square* (RLS) algorithms. Then, a little bit like in previous approaches, the *adaptive eigenvalue decomposition* (AED) is presented. This is followed by an introduction to the *difference functions*. Continuing the presentation of TDE solutions, the *generalized cross-correlation* (GCC) method is introduced, with all its traditional approaches. The end of this chapter describes the *wavelet based TDE*, including specific methods.

Chapter 3 proposes the accumulated GCC TDE methods for multi-frame analysis. Firstly the accumulated cross-power spectrum is presented, as well as ways extend it to all well-known GCC methods. Then the database and the metrics used in the following experiments are described. Further, based on several metrics (accuracy and error rate, relative error, standard deviation of relative error, computing time) all traditional GCC methods implemented with conventional and proposed approaches are analyzed.

Chapter 4 is dedicated to the description of several VAD algorithms. The standard G729, ETSI-AMR1 and ETSI-AMR2 VADs are presented at the beginning, as important reference points of this field. They are followed by two recently proposed VAD algorithms which were integrated into the BSS solution.

Chapter 5 describes the VAD algorithms analysis for the BSS solution. This is supported by detailed aspects regarding the correlation between the VAD scores and speech intelligibility. Beside the BSS solutions based on the two VAD algorithms presented in chapter 4, here a new BSS solution is introduced, called Smoothed Sub-band Spectral Flatness Measure (3SFM). Moreover, it is shown that this solution could also be used as a VAD in proper configuration.

Chapter 6 is dedicated to the enhanced best signal selection. It describe the state of the art unbiased estimator based on speech presence probability and distributed multi-channel speech enhancement. Further it is proposed the eBSS solution based on the SNR estimation. The simulated results are analyzed based on the Perceptual Objective Listening Quality Assessment (POLQA).

Chapter 7 summarizes the conclusion of this thesis regarding time delay estimation, voice activity detection, best signal selection and enhanced best signal selection issues. Then it includes references to the personal contribution and describes the further directions to be followed.

2 TIME DELAY ESTIMATION

Despite the various techniques developed over the years, this topic continues to be interesting. As technology evolved, more and more applications demanded a real time solution for time delay estimation. For echo canceling, acoustics, radar and sonar localization, seismic and medical processing, pattern detection and speech enhancement, scientists are still looking to improve the existent solutions. However, the variety of TDE applications, implementation aspects and proper constraints, inhibit the design of a unique solution. Instead, various approaches have been developed based on application specific aspects [Marinescu, 2013d].

The main used approaches for TDE can be presented in more categories: a) generalized cross-correlation (GCC), b) adaptive filtering and c) adaptive eigenvalue decomposition (AED). Beside these main categories other techniques like average square difference function (ASDF), average mean difference function (AMDF) and wavelet based TDE were proposed over time, but they did not catch on in digital signal applications. In the next subsections all these methods will receive some attention for their description.

2.1 ADAPTIVE FILTERING METHODS

The adaptive filters appeared as a solution for non-stationary environments and signals, or for applications where a low processing delay or a sample-by-sample adaptation process was required. Over the years several adaptation algorithms were proposed and were based in general on the well-known least mean square (LMS) and recursive least square (RLS) methods. The development of adaptive algorithms aimed to improve one or more factors as the speed of convergence to optimal operating condition, the computational complexity, the numerical stability, the minimum error at convergence or the robustness of the algorithm to initial parameter states. Some notable versions and comparison of several adaptive algorithms were presented in [Widrow, 1985][Dohnal, 1995][Lin, 1998][So, 2001][Zetterberg, 2005][Benesty, 2006][Khong, 2006][Dyba, 2008][Emadzadeh, 2008] [Hongyang, 2008][Jinhong, 2008][Iqbal, 2008][Hongyang, 2009][Paleologu, 2010][Sakhnov, 2011a] [Sakhnov, 2011c].

2.1.1 LEAST MEAN SQUARE ALGORITHM

From the time when B.Widrow proposed an adaptive filtering technique based on Least Mean Squares (LMS) [Widrow, 1959, 1960ab, 1961], an adaptive theory also found an application to delay estimation. An adaptive implementation of the time delay estimation via Widrow's LMS algorithm is usually referred to as TDLMS or LMSTDE. The adaptive filtering algorithms determine the time delay in an iterative manner.

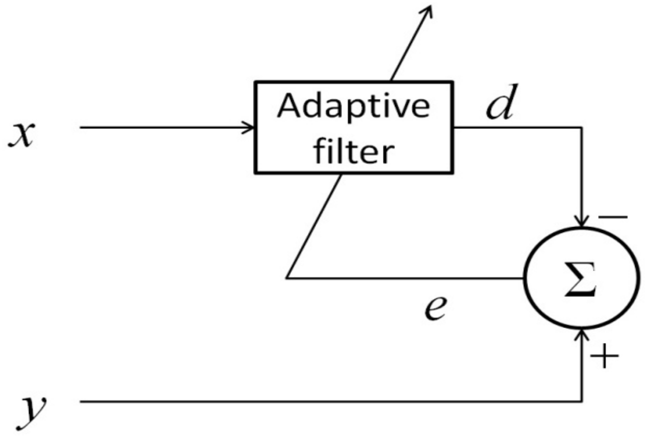


Figure 2.1 – System block diagram for LSM – TDE

The system block diagram of the LMS-TDE is shown in Figure 1. The mathematical model of the digital input signals is given by:

$$\begin{aligned} x(t) &= s(t) + n_1(t) \\ y(t) &= s(t-D) + n_2(t) \end{aligned} \qquad (2.1)$$

where *s(t)* is the unknown source signal, *$n_1(t)$* and *$n_2(t)$* are the additive noises and *D* the delay between received *x(t)* and *y(t)*. The basic idea is to model the time difference by an adaptive non causal FIR filter. If *W(t)* is the *L*-vector off filter weights at instant *t*, then the filter's output is expressed as follows:

$$d(t) = w^T(t) \cdot X(t) \tag{2.2}$$

where *T* denotes the transpose and $X(t)$=[$x(t)$, $x(t-1)$, …$x(t-L+1)$]^T the state of the filter, represented by the last *L* samples of the reference input signal, *x*. Then, the output error *e*(*t*) can be expressed as:

$$e(t) = y(t) - W^T(t) \cdot X(t) \tag{2.3}$$

In order to minimize the error between *y* and *d* the filter's weights are updated on each new sample based on the following relation:

$$W(t+1) = W(n) + \mu \cdot e \cdot X^*(t) \tag{2.4}$$

where μ is the feedback coefficient which influences the convergence speed and * indicates the complex conjugate. If μ is increasing then the algorithm could converge much quicker, but on the other hand it could face stability issues. For the proper μ, after the adaptation phase has been finished, the filter's coefficients insert an equal delay but opposite to the initial existing delay between the input signals *x* and *y*. In an ideal scenario, with no additive noise, the adapted coefficients can estimate perfectly the delay. In this situation all the coefficients will be zero, with the exception of the coefficient which corresponds to the true delay, which will be one. But, in real applications the additive noise is always present and some errors appear. These affect the filter's weightings which will be in the interval (0, 1). Now, the estimated delay corresponds to the maximum coefficient.

2.1.2 Recursive Least Square Algorithm

The RLS [Gauss, 1821] adaptive filter represents a time-update, additive-sample, version of the Wiener filter. While for non-stationary signals the RLS algorithm follows the time variations of the process, for stationary signals it yields the same optimal filter solution as the Wiener filter at the end of its convergence stage. Comparing it to previous LMS, it has a higher convergence speed and also a higher computation complexity.

The RLS algorithm uses an initial state when it begins the adaptation stage. Then it uses every input sample to adapt the filter coefficients. Considering the input signals *x(t)* and *y(t)* defined as in (2.1) from the above context with the filter coefficient vector expressed by *w*(*t*) = [$w_0(t)$, $w_1(t)$,…, $w_{L-1}(t)$] we have the following formula for the filter's output:

$$\hat{x}(t) = w^{\mathrm{T}}(t)x(t) \tag{2.5}$$

where $\hat{x}(t)$ represents an estimate of the delayed signal *y(t).*Similar to (2.3) the difference between the filter output and the target signal yields the filter error *e*(*t*).

For stationary signals it was shown [Vaseghi, 2006 – chapters 6 and 7] that the Wiener filter resulted by minimizing the above mean square error as:

$$\boldsymbol{w} = \boldsymbol{R}_{xx}^{-1}\boldsymbol{r}_{xy} \tag{2.6}$$

Now we have to express (2.6) by a time-update, recursive, adaptive form. As shown in [Vaseghi, 2006 – chapter 6], in the case of *N* sample vectors, the cross-correlation matrix is expressed as:

$$\boldsymbol{R}_{xx} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = \sum_{t=0}^{N-1}\boldsymbol{x}(t)\,\boldsymbol{x}^{\mathrm{T}}(t) \tag{2.7}$$

where $\boldsymbol{x}(t) = [x(t),\ldots, x(t - N - 1\ )]^{\mathrm{T}}$. It is now possible to express the above vector product in a recursive form as

$$\boldsymbol{R}_{xx}(t) = \boldsymbol{R}_{xx}(t-1) + \boldsymbol{x}(t)\,\boldsymbol{x}^{\mathrm{T}}(t) \tag{2.8}$$

To adapt to the time variations of the signal statistics, the autocorrelation estimate from (2.7) is windowed with an exponentially decreasing window:

$$\boldsymbol{R}_{xx}(t) = \lambda\boldsymbol{R}_{xx}(t-1) + \boldsymbol{x}(t)\,\boldsymbol{x}^{\mathrm{T}}(t) \tag{2.9}$$

where λ is known as the *forgetting* factor, or *adaptation*, and $0 < \lambda < 1$. In the same way we can write the cross-correlation vector:

$$\boldsymbol{r}_{xy} = \sum_{i=0}^{N-1}\boldsymbol{x}(t)\,\boldsymbol{y}(t) \tag{2.10}$$

Applying now the recursive form for the above equation we obtain

$$\boldsymbol{r}_{xy}(t) = \boldsymbol{r}_{xy}(t-1) + \boldsymbol{x}(t)\,\boldsymbol{y}(t) \tag{2.11}$$

Moreover, introducing here the same exponentially decreasing forgetting factor λ leads to an adaptation form

$$\boldsymbol{r}_{xy}(t) = \lambda\boldsymbol{r}_{xy}(t-1) + \boldsymbol{x}(t)\,\boldsymbol{y}(t) \tag{2.12}$$

To achieve a recursive solution of the least square error equation (2.12), we have to find a recursive time update formula for the inverse matrix in the form

$$\boldsymbol{R}_{xx}^{-1}(t) = \boldsymbol{R}_{xx}^{-1}(t-1) + \text{Update(t)} \tag{2.13}$$

This can be achieved by the following *matrix inversion lemma.*

The matrix inversion lemma

Let ***A*** and ***B*** be two positive-definite *L*× *L* matrices associated by

$$\boldsymbol{A} = \boldsymbol{B}^{-1} + \boldsymbol{C}\boldsymbol{D}^{-1}\boldsymbol{C}^{\mathrm{T}} \tag{2.14}$$

where ***C*** is a *L* × *N* matrix and ***D*** is a positive-definite *N* × *N* matrix. The matrix inversion lemma states that the inverse if the matrix ***A*** can be calculated as

$$\boldsymbol{A}^{-1} = \boldsymbol{B} - \boldsymbol{B}\boldsymbol{C}\left(\boldsymbol{D} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{B} \tag{2.15}$$

Proving this lemma consists of multiplying (2.14) and (2.15). Then the identity matrix results in the left- and right-hand sides of the equation.

To use the matrix inversion lemma for the recursive inverse correlation matrix we identify matrices ***A**, **B**, **C*** and ***D*** as:

$$\boldsymbol{A} = \boldsymbol{R}_{xx}(t) \tag{2.16}$$

$$\boldsymbol{B} = \lambda^{-1}\boldsymbol{R}_{xx}^{-1}(t-1) \tag{2.17}$$

$$\boldsymbol{C} = x(t) \tag{2.18}$$

$$\boldsymbol{D} = \text{identity matrix} \tag{2.19}$$

Now, replacing equations (2.9) and (2.10) in (2.8) yields:

$$\boldsymbol{R}_{xx}^{-1}(t) = \lambda^{-1}\boldsymbol{R}_{xx}^{-1}(t-1) - \frac{\lambda^{-2}\boldsymbol{R}_{xx}^{-1}(t-1)\boldsymbol{x}(t)\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{R}_{xx}^{-1}(t-1)}{1+\lambda^{-1}\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{R}_{xx}^{-1}(t-1)\boldsymbol{x}(t)} \tag{2.20}$$

Now let the variables $\phi(t)$ and $k(t)$ be

$$\phi(t) = \boldsymbol{R}_{xx}^{-1}(t) \tag{2.21}$$

and

$$k(t) = \frac{\lambda^{-1}\boldsymbol{R}_{xx}^{-1}(t-1)\boldsymbol{x}(t)}{1+\lambda^{-1}\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{R}_{xx}^{-1}(t-1)\boldsymbol{x}(t)} \tag{2.22}$$

or

$$k(t) = \frac{\lambda^{-1}\phi(t-1)\boldsymbol{x}(t)}{1+\lambda^{-1}\boldsymbol{x}^{\mathrm{T}}(t)\phi(t-1)\boldsymbol{x}(t)} \tag{2.23}$$

Based on the new equations (2.21) and (2.22) we can rewrite the recursive equation (2.13) for the inverse matrix computation:

$$\phi(t) = \lambda^{-1}\phi(t-1) - \lambda^{-1}k(t)\boldsymbol{x}^{\mathrm{T}}(t)\phi(t-1) \tag{2.24}$$

Using (2.23) and (2.24) we obtain

$$\begin{aligned} k(t) &= \left[\lambda^{-1}\phi(t-1) - \lambda^{-1}k(t)\boldsymbol{x}^{\mathrm{T}}(t)\phi(t-1)\right]\boldsymbol{x}(t) \\ &= \phi(t)\boldsymbol{x}(t) \end{aligned} \tag{2.25}$$

With the (2.24) and (2.25) equations we will further determine the recursive least square adaptation algorithm.

Recursive time-update of filter coefficients

Considering (2.6) and (2.21) the filter coefficients can be expressed as

$$\begin{aligned} \boldsymbol{w}(t) &= \boldsymbol{R}_{\boldsymbol{xx}}^{-1}(t)\boldsymbol{r}_{\boldsymbol{xy}}(t) \\ &= \phi(t)\boldsymbol{r}_{\boldsymbol{xy}}(t) \end{aligned} \tag{2.26}$$

Replacing the recursive form of the correlation vector in the above equation leads to

$$\begin{aligned} \boldsymbol{w}(t) &= \phi(t)\left[\lambda\boldsymbol{r}_{\boldsymbol{xy}}(t-1) + x(t)y(t)\right] \\ &= \lambda\phi(t)\boldsymbol{r}_{\boldsymbol{xy}}(t-1) + \phi(t)x(t)y(t) \end{aligned} \tag{2.27}$$

Substituting in the right hand of (2.27) the recursive form of the matrix $\phi(t)$ from equations (2.24) and (2.25) leads to

$$\boldsymbol{w}(t) = \left[\lambda^{-1}\phi(t-1) - \lambda^{-1}k(t)\boldsymbol{x}^{\mathrm{T}}(t)\phi(t-1)\right]\lambda\boldsymbol{r}_{\boldsymbol{xy}}(t-1) + k(t)y(t) \tag{2.28}$$

or

$$\boldsymbol{w}(t) = \phi(t-1)\boldsymbol{r}_{\boldsymbol{xy}}(t-1) - k(t)\boldsymbol{x}^{\mathrm{T}}(t)\phi(t-1)\boldsymbol{r}_{\boldsymbol{xy}}(t-1) + k(t)\boldsymbol{y}(t) \tag{2.29}$$

Replacing $\boldsymbol{w}(t-1) = \phi(t-1)\boldsymbol{r}_{\boldsymbol{xy}}(t-1)$ in equation (2.29) leads to

$$\boldsymbol{w}(t) = \boldsymbol{w}(t-1) + k(t)\left[\boldsymbol{y}(t) - \boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{w}(t-1)\right] \tag{2.30}$$

Finally, this equation is equivalent to the next form

$$\boldsymbol{w}(t) = \boldsymbol{w}(t-1) + k(t)e(i) \tag{2.31}$$

which represents the implementation of the recursive time-update least square error Wiener filter.

Now, for the input signals $x(t)$ and $y(t)$ the recursive least square adaptation algorithm with the initial values $\phi(t)=\delta\mathbf{I}$ and $\boldsymbol{w}(0)=\boldsymbol{w}_{\mathbf{I}}$ can be described in the following steps:

1)Update the filter gain vector with the following equation

$$k(t)=\frac{\lambda^{-1}\phi(t-1)\boldsymbol{x}(t)}{1+\lambda^{-1}\boldsymbol{x}^{\mathrm{T}}(t)\phi(t-1)\boldsymbol{x}(t)} \tag{2.32}$$

2)Compute the error signals with the next equation:

$$e(t)=y(t)-\boldsymbol{w}^{\mathrm{T}}(t-1)x(t) \tag{2.33}$$

3)Update the filter coefficients

$$\boldsymbol{w}(t)=\boldsymbol{w}(t-1)+k(t)e(i) \tag{2.34}$$

4)Update the inverse correlation matrix

$$\phi(t)=\lambda^{-1}\phi(t-1)-\lambda^{-1}k(t)\boldsymbol{x}^{\mathrm{T}}(t)\phi(t-1) \tag{2.35}$$

5)Jump to the first step for the next sample.

2.2 Adaptive Eigen Value Decomposition

The methods for time delay estimation differ from one application to another. A special case is represented by the indoor applications for speech, communication and localization. Beside the additive noise and non-stationarity of voice, signals are affected also by reverberations. In [Bedard, 1994] and later in [Champagne, 1996], it was shown that make achieving good performance for indoor applications more difficult.

A new method was proposed in [Benesty, 2000], including reverberations in a real signal model based on eigenvalue decomposition. In this sense the equations regarding the input signals in case of multiple microphones (i=1,2,…) are expressed bellow:

$$x_i(t)=\alpha_i s(t-\tau_i)+n_i(t), \tag{2.36}$$

where α_{i} is the attenuation factor caused by the propagation effects, $n_{\mathrm{i}}(t)$ represents the additive noise signal which affects the i-th microphone and τ_{i} is the time needed by the signal to propagate the source $s(t)$ to the microphone.

If we further consider that $s(t)$, $n_1(t)$ and $n_2(t)$ are stationary, uncorrelated, zero-mean Gaussian random processes then τ_{12} defines the relative delay between the two microphone signals 1 and 2 as follows:

$$\tau_{12} = \tau_1 - \tau_2\,, \tag{2.37}$$

But the above model is an ideal one because determining τ_{12} has a clear solution. In a real model it has to be more complete, including the room's reverberations from the acoustic environment, which were not taken into account in the ideal model. In this case the real model for multiple microphones is expressed as follows:

$$x_i(t) = g_i * s(t) + n_i(t) \tag{2.38}$$

where g_{i} represents the acoustic impulse response between the source $s(t)$ and the i-th microphone and * denotes the convolution operation. Moreover, the noisy signals $n_1(t)$ and $n_2(t)$ could be correlated. This is not a rare situation, because there are frequent cases with directional indoor noise, like those generated by an overhead projector or from a ceiling fan.

In this case, the approach proposed in [Benesty, 2000] analyzes the impulse responses between the source and the microphones to estimate the time-delay. The idea of this method supposes that the system (in generally the room) is linear and time invariant, thus results the following equation:

$$\mathrm{x}_1^T(t) \cdot g_2 = \mathrm{x}_2^T(t) \cdot g_1\ , \tag{2.39}$$

where $\mathrm{x}(t)$=[$x_{\mathrm{i}}(t)$, $x_{\mathrm{i}}(t - 1)$, … $x_{\mathrm{i}}(t - L$+1)]^T for i=1,2 are vectors of signal samples at the microphone outputs, T represents the transpose of a vector or a matrix, and the impulse response vectors of length L are defined as

$$\mathbf{g}_i = [\, g_{i,0} \;\; g_{i,1} \ldots \; g_{i,L\text{-}1}]^T, \quad i=1,2. \tag{2.40}$$

The linearity property of the above relation results from the fact that x_i=s*g_i, i=1,2, thus x_1*g_2=s*g_1*g_2=x_2*g_1.

For two microphone signals we have the following covariance matrix

$$R = \begin{bmatrix} R_{x_1x_1} & R_{x_1x_2} \\ R_{x_2x_1} & R_{x_2x_2} \end{bmatrix}, \tag{2.41}$$

where

$$R_{x_ix_j} = E\{\mathrm{x}_i(t) \cdot \mathrm{x}_j^T(t)\}, \quad i,j \tag{2.42}$$

and $E\{\cdot\}$ indicates mathematical expectation.

Consider the $2L\times1$ vector

$$\mathbf{u} = \begin{bmatrix} \mathbf{g}_2 \\ -\mathbf{g}_1 \end{bmatrix}. \tag{2.43}$$

Using (2.8), (2.10) and (2.12) we can easily obtain **R**·**u** = 0. This confirms that the vector **u**, which contains the two impulse responses, represents the eigenvector of the covariance matrix **R** corresponding to the eigenvalue 0. Furthermore, if the two impulse responses g_1 and g_2 have no common zeros and the autocorrelation matrix of the source signal $s(t)$ is full rank, which is assumed here, the covariance matrix **R** can have one and only one eigenvalue equal to 0 [Tong, 1993].

However, in real applications it is not simple to accurately estimate the vector **u**, because of the non-stationarity nature of speech, the length of the impulse responses, the additive background noise, etc. But, a trivial solution is represented by an iterative estimation algorithm of the eigenvector which corresponds to the maximum (or minimum) eigenvalue of **R**. The Frost algorithm can be used [Frost, 1972], which is a simple restrained LMS, or the algorithm presented in [Owsley, 1978]. Next, achieving the solution to this problem by using these techniques is presented. In order to obtain the optimum filter weights $\mathbf{u}_{\text{opt}}$ we have to minimize the quantity $\mathbf{u}^T\mathbf{R}\mathbf{u}$ with respect to **u** and subject to $||\mathbf{u}||^2 = \mathbf{u}^T\mathbf{u} = 1$.

Consider the error signal:

$$e(t) = \frac{\mathbf{u}^T(t)\mathrm{x}(t)}{\|\mathbf{u}(t)\|}, \tag{2.44}$$

where $\mathbf{x}(t) = [\ \mathrm{x}_1^T(t) \quad \mathrm{x}_2^T(t)]^T$. And now, the solution to the above eigenvalue problem is equivalent with the minimization of the mean square value of $e(t)$. This can be achieved by computing the gradient of $e(t)$ with respect to $\mathbf{u}(t)$

$$\nabla e(t) = \frac{1}{\|\mathbf{u}(t)\|}\left[\mathrm{x}(t) - e(t)\frac{\mathbf{u}(t)}{\|\mathbf{u}(t)\|}\right], \tag{2.45}$$

and we obtain the gradient-descent constrained LMS algorithm:

$$\mathbf{u}(t+1) = \mathbf{u}(t) - \mu e(t)\nabla e(t), \tag{2.46}$$

where μ is the adaptation step as a positive constant.

Substituting (2.44) and (2.45) into (2.46) yields

$$\mathbf{u}(t+1) = \mathbf{u}(t) - \frac{\mu}{\|\mathbf{u}(t)\|}\left[\mathrm{x}(t)\mathrm{x}^T(t)\frac{\mathbf{u}(t)}{\|\mathbf{u}(t)\|} - e^2(t)\frac{\mathbf{u}(t)}{\|\mathbf{u}(t)\|}\right], \tag{2.47}$$

and continuing with the mathematical expectation after convergence, we obtain

$$\mathbf{R}\frac{\mathbf{u}(\infty)}{\|\mathbf{u}(\infty)\|} = E\{e^2(t)\}\frac{\mathbf{u}(\infty)}{\|\mathbf{u}(\infty)\|}, \tag{2.48}$$

which is what is our solution: the eigenvector $\mathbf{u}(\infty)$ which corresponds to the smallest eigenvalue $E\{e^2(t)\}$ of the covariance matrix **R**.

In real applications we can avoid the round off error propagation by using the following adaptation formula, proposed by [Bellanger, 1989]:

$$\mathbf{u}(t+1) = \frac{\mathbf{u}(t) - \mu \cdot e(t) \cdot \nabla e(t)}{\|\mathbf{u}(t) - \mu \cdot e(t) \cdot \nabla e(t)\|}. \tag{2.49}$$

If this advantageous approach is used, then we can remove $\|\mathbf{u}(t)\|$ (which appears in $e(t)$ and $\nabla e(t)$) because it will always give us $\|\mathbf{u}(t)\| = 1$.

Note that the last equation can be used as a general way on any matrix **R** to obtain the eigenvector which corresponds to its smallest eigenvalue. However, making use of the fact that in our case the smallest eigenvalue is zero we can simplify even more the algorithm as follows:

$$e(t) = \mathbf{u}^T(t)\mathrm{x}(t) \tag{2.50}$$

and

$$\mathbf{u}(t+1) = \frac{\mathbf{u}(t) - \mu \cdot e(t) \cdot \mathrm{x}(t)}{\|\mathbf{u}(t) - \mu \cdot e(t) \cdot \mathrm{x}(t)\|}. \tag{2.51}$$

Finally, we can conclude that the last algorithm from (2.51) may be considered as an approximation of the previous one by neglecting the terms in $e^2(t)$ from (2.49). For indoor applications, these two algorithms should offer the same performance after they converged even for a low SNR. Also, they appear to be more efficient than any other methods for reverberant environment applications [Benesty, 2000].

2.3 DIFFERENCE FUNCTIONS

Difference functions were used from the beginning of the digital signal processing because they were based on low-cost computing operations. Moreover, Average Magnitude Difference Function (AMDF) only uses additions. For the other operator, Average Square Difference Function (ASDF), it was shown in [Jacovitti, 1993] that it offers better accuracy results, thanks to the inclusion of multiply operations. But, because direct generalized cross correlation benefits from the reduced complexity order of the FFT approach, average difference functions now have limited applications in our days.

2.3.1 Average Magnitude Difference Function

Thanks to its simplicity, this operator was widely used in audio applications, especially to determine the pitch period of voiced speech sound after the work from [Ross, 1974]. In [Jacovitti, 1987] a new faster correlation estimator was proposed, with applications for time delay estimation.

Considering *N* samples of two signals, $x_1(t)$ and $x_2(t)$, acquired at a sampling interval *T*, then the average magnitude difference function estimator is expressed as

$$\widehat{D}_{AMDF} = \arg\min_{\tau} \widehat{R}_{AMDF} \tag{2.52}$$

where

$$\widehat{R}_{AMDF}(\tau) = \frac{1}{N}\sum_{k=1}^{N} \left| x_1(kT) - x_2(kT+\tau) \right| \tag{2.53}$$

Compared to the direct cross correlation estimator which searches for the maximum value of the cross-correlation, the AMDF estimator searches for the minimum value of the average magnitude difference functions.

2.3.2 Magnitude Difference Function

Considering the above AMDF it is worthwhile to notice that the last averaging operation is not used in some applications. This eliminates the expensive time processing division operation leading to the magnitude difference estimator:

$$\widehat{D}_{MDF} = \arg\min_{\tau} \widehat{R}_{MDF} \tag{2.54}$$

where

$$\widehat{R}_{MDF}(\tau) = \sum_{k=1}^{N} \left| x_1(kT) - x_2(kT+\tau) \right| \tag{2.55}$$

2.3.3 Average Square Difference Function

The average square difference function tries to improve the estimation accuracy by including multiply operations. This increases the computing time, but for some applications it is a good trade between the accuracy result and processing speed. The ASDF estimator is similar to the AMDF estimator, searching also the minimum of the ASDF as in:

$$\widehat{D}_{ASDF} = \arg\min_{\tau} \widehat{R}_{ASDF} \tag{2.56}$$

where

$$\widehat{R}_{ASDF}(\tau)=\frac{1}{N}\sum_{k=1}^{N}\left[x_1(kT)-x_2(kT+\tau)\right]^2 \tag{2.57}$$

In [Jacovitti, 1993] it was shown that for a high SNR the ASDF estimator achieves higher accuracy results than the direct cross-correlation estimator. The better accuracy results were explained by the fact that for clear signals, the ASDF estimator yields correct estimation, while the direct cross-correlation does not.

2.3.4 Square Difference Function

Similar to the MDF estimator case, the ASDF could be simplified by removing the final average part. In this way results the Square Difference Function based estimator:

$$\widehat{D}_{SDF}=\arg\min_{\tau}\widehat{R}_{SDF} \tag{2.58}$$

where

$$\widehat{R}_{SDF}(\tau)=\sum_{k=1}^{N}\left[x_1(kT)-x_2(kT+\tau)\right]^2 \tag{2.59}$$

2.4 Generalized Cross-Correlation Methods

2.4.1 Cross-Correlation for Time Delay Estimation

Adaptive filtering methods lead to very high accuracy results when they are used to estimate the time delay, but only after the convergence period has finished. But, in several applications there is no time left to wait for the filter to adapt. The solution in these cases is represented by the generalized cross-correlation methods, based on the simple cross-correlation between two signals. This approach offers reliable accuracy results and it is much quicker than adaptive filtering methods because it does not need any adaptation time.

Assuming two noisy signals, $x_1(t)$ and $x_2(t)$ defined as in (2.36) which come from the same source $s(t)$, delayed with τ_{12}, computed as in (2.37). Then, the cross-correlation between the two signals can be expressed as

$$\begin{aligned} r_{x_1x_2}(k) &= E\left[x_1(t)x_2(t+k)\right] \\ &= E\left\{\left[\alpha_1 s(t)+n_1(t)\right]\left[\alpha_2 s(t-\tau_{12}+k)+n_2(t+k)\right]\right\} \\ &= \alpha_1\alpha_2 r_{ss}(k-\tau_{12})+\alpha_1 r_{sn_2}(k)+\alpha_2 r_{sn_1}(k-\tau_{12})+r_{n_1n_2}(k) \end{aligned} \tag{2.60}$$

where *E* denotes expectations.

Considering that the signal and noise are uncorrelated, then

$$r_{sn_2}(k) = r_{sn_1}(k - \tau_{12}) = r_{n_1 n_2}(k) = 0\,, \tag{2.61}$$

which implies

$$r_{x_1 x_2}(k) = \alpha_1 \cdot \alpha_2 \cdot r_{ss}(k - \tau_{12}) \tag{2.62}$$

Because the autocorrelation r_{ss} has its maximum value for $r_{ss}(0)$, than the time delay τ_{12} is obtained as the argument which maximizes $r_{x_1 x_2}$, which can also be seen in Figure 2.2 [Vaseghi, 2006].

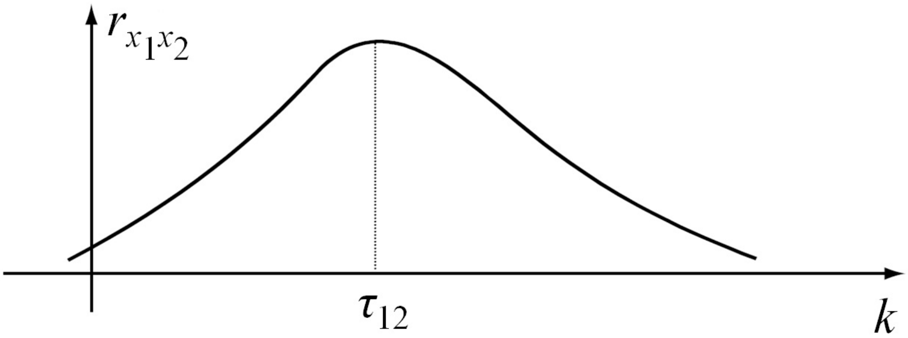


Figure 2.2 – The cross-correlation for two delayed signals is maximized when the argument value equals the delay

It can be shown that the cross-correlation between $x_1(t)$ and $x_2(t)$ is related to the cross-power spectral density function G_{x1x2} by the well-known Fourier transform relationship:

$$\begin{aligned} r_{x_1 x_2}(t) &= \mathcal{F}^{-1}\{G_{x_1 x_2}(f)\} \\ &= \int_{-\infty}^{\infty} G_{x_1 x_2}(f) \cdot e^{2\pi j f t} df \end{aligned} \tag{2.63}$$

where

$$G_{x_1 x_2}(f) = E\left[X_1(f) X_2^*(f)\right], \tag{2.64}$$

and

$$\begin{aligned} X_i(f) &= \mathcal{F}\{x_i(t)\} \\ &= \int_{-\infty}^{\infty} x_i(t) \cdot e^{-2\pi j f t} dt \end{aligned} \tag{2.65}$$

2.4.2 TRADITIONAL GENERALIZED CROSS-CORRELATION METHODS

The term of *generalized cross-correlation* was introduced by Knapp and Carter in [Knapp, 1976] when they pointed out that a common method of determining the time delay is to compute the cross-correlation function. To improve the accuracy of delay estimation, a pre-filtering of the inputs is necessary before calculating the cross correlation. This is equivalent to the addition of a filtering block. If then we consider that the signals $x_1(t)$ and $x_2(t)$ have been filtered with filters having transfer functions $H_1(f)$ and $H_2(f)$, the cross power spectrum between the filter outputs is given by:

$$G_{x_1x_2}^{g}(f)=H_1(f)\cdot H_2^*(f)\cdot G_{x_1x_2}(f) \tag{2.66}$$

Therefore, the generalized cross-correlation between $x_1(t)$ and $x_2(t)$ is:

$$R_{x_1x_2}^{g}(t)=\int_{-\infty}^{\infty}\Psi(f)\cdot G_{x_1x_2}(f)\cdot e^{j2\pi ft}df \tag{2.67}$$

where

$$\Psi(f)=H_1(f)\cdot H_2^*(f) \tag{2.68}$$

denotes the general frequency weighting [Knapp, 1976].

Over the years, different weighting functions were proposed to improve the estimation process of the basic cross-correlation. The well-known weighting functions, which are also analyzed in this thesis, are presented below. G_{x1x1} and G_{x2x2} represent the auto power spectrum of the noisy signals and $\gamma_{x_1x_2}^2(f)$ is the signal's coherence function.

$$G_{x_1x_1}(f)=E\left[X_1(f)X_1^*(f)\right] \tag{2.69}$$

$$G_{x_2x_2}(f)=E\left[X_2(f)X_2^*(f)\right] \tag{2.70}$$

$$\gamma_{x_1x_2}^2(f)=\frac{\left|G_{x_1x_2}(f)\right|^2}{G_{x_1x_1}(f)G_{x_2x_2}(f)} \tag{2.71}$$

For the normal Cross-Correlation (CC) the weighting function $\Psi(f)$ is 1. This is the basic and the fastest computing GCC, because it has no weighting operations.

The Eckart filter derives its name from work done in this area and published in [Eckart, 1951]. It maximizes the deflection criterion, i.e., the ratio of the change in mean correlator output due to signal present to the standard deviation of the correlator output due to noise alone [Knapp, 1976]. Its weighting function can be expressed as

$$\Psi_{Eckart}(f) = \frac{\left|G_{x_1x_2}(f)\right|}{\left[G_{x_1x_1}(f) - \left|G_{x_1x_2}(f)\right|\right] \cdot \left[G_{x_2x_2}(f) - \left|G_{x_1x_2}(f)\right|\right]} \tag{2.72}$$

Twenty years later, in 1971, Roth proposed a new processor in [Roth, 1971]. It has the desirable effect of suppressing those frequency regions where G_{x1x1} is large and the estimate of G_{x1x2} is more likely to be in error [Knapp, 1976].

$$\Psi_{Roth}(f) = \frac{1}{G_{x_1x_1}(f)} \tag{2.73}$$

The same year, another weighting function was proposed, the HT processor, by Hannan and Thomson. This assigns greater weight in regions of frequency domain where the coherence is large [Hannan, 1971]. In [Knapp, 1971], it was shown that HT processor is a maximum likelihood (ML) estimator for time delay under usual conditions. Under a low signal-to-noise ratio restriction, the HT processor is equivalent to Eckart pre-filtering and cross-correlation.

$$\Psi_{HT}(f) = \frac{\left|\gamma_{x_1x_2}(f)\right|^2}{\left|G_{x_1x_2}(f)\right| \cdot \left[1 - \left|\gamma_{x_1x_2}(f)\right|^2\right]} \tag{2.74}$$

The SCOT (Smoothed Coherence Transform) was introduced by Carter, Nuttall and Cable in [Carter, 1973], to reduce the influence of a strong tone. However, for smoothed signal and noise spectra, Hassab and Boucher have noted that the additional SCOT weighting function has weakened the performance of the basic cross correlator, while other functions have improved it [Hassab, 1980][Hassab, 1981].

$$\Psi_{SCOT}(f) = \frac{1}{\sqrt{G_{x_1x_1}(f) \cdot G_{x_2x_2}(f)}} \tag{2.75}$$

Phase Transform (PHAT) or Cross-power Spectrum Phase (CSP) was developed purely as an ad-hoc technique to avoid spreading of the above two presented operators. Ideally, PHAT does not suffer the spreading that other processors do. Also, because it weights G_{x1x2} as the inverse of $|G_{x1x2}|$, the errors are accentuated where the signal power is smallest [Knapp, 1976].

$$\Psi_{CSP}(f) = \frac{1}{\left|G_{x_1x_2}(f)\right|} \tag{2.76}$$

In practice a modified SCOT weighting function, called *Cross-Power Spectrum–m* (CPS-m), is also used, as expressed below

$$\Psi_{CPS-m}(f) = \frac{1}{\sqrt[m]{G_{x_1x_1}(f)\cdot G_{x_2x_2}(f)}} \tag{2.77}$$

where m is usually chosen as $1 < m < 2$.

In 1979, the HB processor was presented by Hassab and Boucher. It is similar to SCOT in that, for highly dynamic spectra, in addition to suppressing the cross-spectral estimate in frequency regions of low signal-to-noise ratio, high signal-to-noise ratio regions are also suppressed in attempt to reject strong tones in the observations [Hassab, 1979].

$$\Psi_{HB}(f) = \frac{\left|G_{x_1x_2}(f)\right|}{G_{x_1x_1}(f)\cdot G_{x_2x_2}(f)} \tag{2.78}$$

The Wiener processor was proposed in 1985 by Hero and Schwartz. Based on the channel's linearity it tries to estimate the original signal from the observation $x_1(t)$ and channel output signal from $x_2(t)$, by minimizing the mean-square errors. Thus, given the channel characteristics, the solution results in Wiener filters, which yield the Wiener weighting function [Hero, 1985].

$$\Psi_{Wiener}(f) = \left|\gamma_{x_1x_2}(f)\right|^2 \tag{2.79}$$

In 1996 a new weighting function, for acoustic localization, was presented, by Rabinkin et al., the ρ-Cross-power Spectrum Phase (ρ-CSP). It adds to the normal CSP the tuning parameter ρ (with values between 0 and 1) as a whitening parameter, which discards the non-speech portion (below 200Hz) of the CSP [Rabinkin, 1996].

$$\Psi_{\rho-CSP}(f) = \frac{1}{\left|G_{x_1x_2}(f)\right|^{\rho}} \tag{2.80}$$

Relatively recently, in 2009, in addition to the above work ρ-Cross-power Spectrum Phase with Coherence (ρ-CSPC) was proposed by Shean and Liu. The presence of the minimum of the coherence function in the weighting function helps to reduce errors for relatively small energy signals [Shean, 2009].

$$\Psi_{\rho-CSPC}(f) = \frac{1}{\left|G_{x_1x_2}(f)\right|^{\rho} + \min\left[\gamma_{x_1x_2}(f)^2\right]} \tag{2.81}$$

For the above presented GCC methods, an implementation block diagram is presented in Figure 2.3. First, the analysis frames of input signals are converted into frequency domain using the Fast Fourier Transform (FFT) block. Then, the cross-power spectrum is computed by multiplications of resulted spectra and weighting function. Going further, the generalized cross-correlation is obtained through an Inverse Fast Fourier Transform (IFFT). The final step consists in finding the argument which maximizes GCC and estimating the delay. This is the basic way to obtain an estimation of delay.

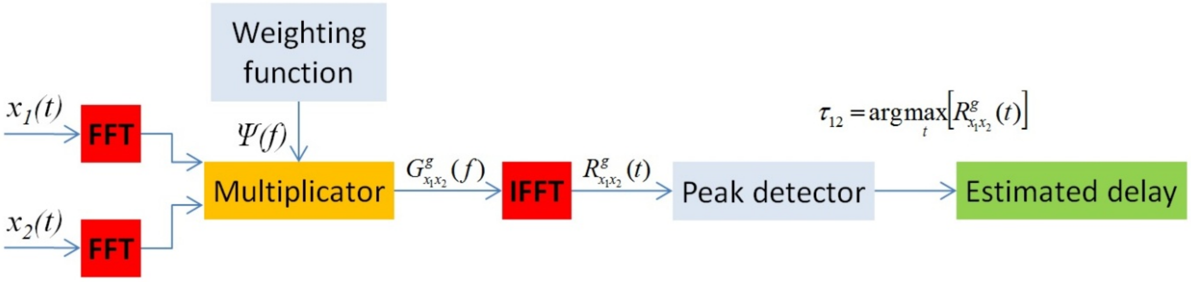


Figure 2.3 – Block diagram for a single frame GCC implementation

For a large window with L samples, FFT's complexity order is $O(L \cdot logL)$, with L a power of 2. Because these consume important processing time, it is natural to search for solutions which increase the computing speed. A way to achieve this is to divide the larger analysis window into smaller frames, as it is shown in Figure 2.4. Thus, the larger analysis window of L samples is divided in K smaller frames, of n samples each. If the length of the frame l is also a power of 2, then the new complexity order is $O(K \cdot l \cdot logl) = O(L \cdot log(L/K))$, which needs a smaller processing time. For each smaller analysis frame, the partial estimated delay is obtained similarly as in Figure 2.3. Then, the final estimated delay yields as the average of all partial estimated delays. In this way, it is also easier to estimate a variable delay. This approach is recommended especially when the estimated delay is expected to be considerably less than the length of the larger window.

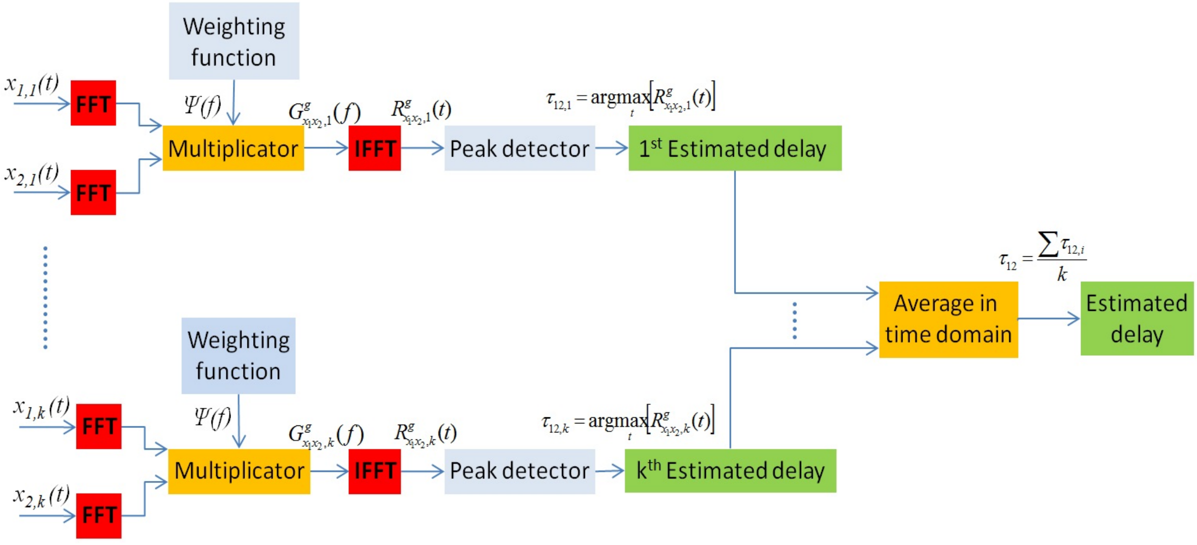


Figure 2.4 – Block diagram for multiple frames GCC with time domain average estimation

2.4.3 Accumulated Cross-Power Spectrum Phase Method

An alternative to the above multi-frame approach is the accumulated Cross-power Spectrum Phase (acc-CSP), proposed by Matassoni and Svaizer in [Matassoni, 2006]. It accumulates the cross-power spectrum over multiple frames in frequency domain, as shown in Figure 2.5. This scheme leads to a new computing time decrease, because the number of IFFT and peak detector is reduced to 1. In frequency domain it can be expressed as follows:

$$G_{acc-CSP}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)}{\left|G_{x_1x_2,k}\left(f\right)\right|}, \tag{2.82}$$

where *K* represents the number of accumulated frames. Besides the reduced computational complexity, the *acc-CSP* method enhances the estimation by intrinsic integration for fixed delay during the analysis window [Matassoni, 2006].

The acc-CSP method proposes the accumulation scheme of cross power spectrum in frequency domain, increasing the computation speed. Methods based on the approach presented in Figure 2.4 compute the TDE as the average of all partial estimated delays of each frame from the analysis window. Therefore, for *K* frames, the number of total FFT operations is equal to *3*x*K*, because two FFT are used to transform the signals from time to frequency domain, and then one IFFT is used on the cross power spectrum to return in the time domain, for each frame. Instead, the accumulation scheme from Figure 2.5 is faster because it does not calculate any partial TDEs. Because the cross-power spectrum averaging is computed in frequency domain, only one estimate will result, for any number of *K* frames. Thus, only one IFFT is needed for the final estimation and *2*x*K* FFTs for time to frequency transformations. This leads to a total number of *2*x*K + 1* FFT for the accumulating scheme, which is less than the *3*x*K* FFT needed by previous methods [Marinescu, 2013c]. Also, a small increase in the computation speed is due to the reduction to only one peak detector call.

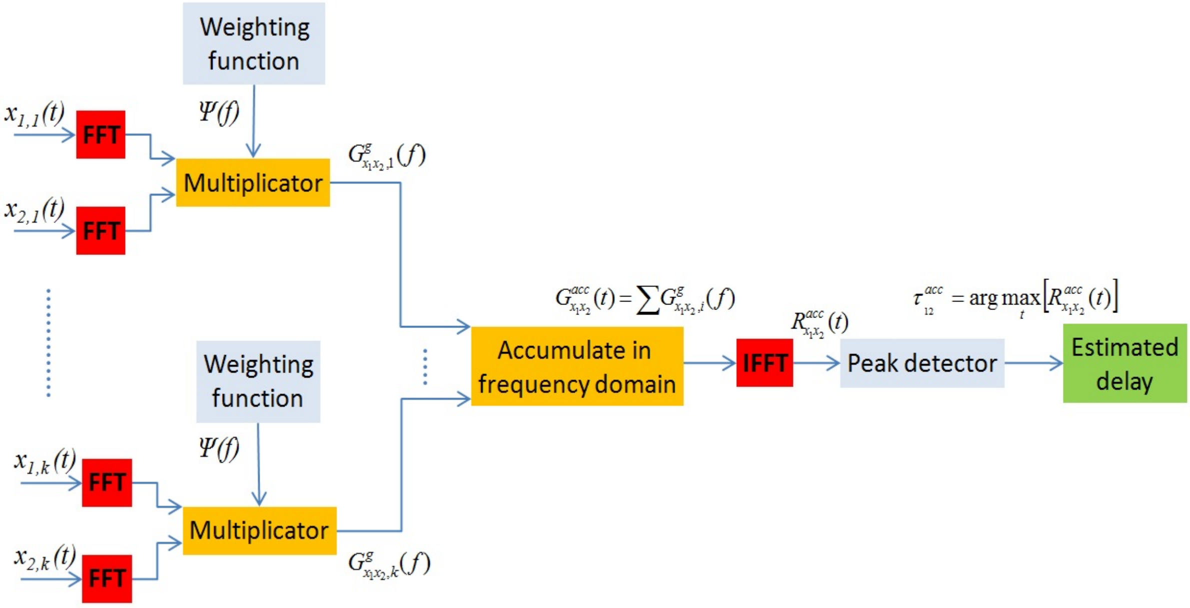


Figure 2.5 – Block diagram for accumulating multiple frames GCC in frequency domain

2.5 Wavelet Based Time Delay Estimation

In special acoustic tracking target applications the time delay estimation problem encounters some difficulties. The applications have to determine and predict the track of the target based on extremely short acoustic signals (transient) received by many sensors. Because of the diversity, short duration and non-stationary nature of these signals the previous TDE methods cannot offer reliable results. However, wavelet transform represents a powerful analysis tool for processing and decomposing these types of signals. The properties of the wavelet transform can provide additional tracking information to improve the previous TDE approaches. Based on these observations different combined methods for TDE were proposed, taking advantage of both wavelet transform and GCC.

2.5.1 Wavelet Pre-filtering Generalized Cross Correlation

The wavelet decomposition produces correlation coefficients between the input and the basic wavelet function. If the basic function is chosen properly so as to match the signal characteristics, then the higher the coefficients are, the higher the correlation is between the input signal and the basic function. Thus, the lower coefficients indicate a lower correlation between the input signal and the basic function, which indicate that the input signal does not contain the desired information, affected by noise. Based on the wavelet denoising, which eliminates the correlation coefficients which are smaller than a cutoff threshold, Wu et al. proposed in [Wu, 1997] the wavelet pre-filtering generalized cross-correlation method for time delay estimation.

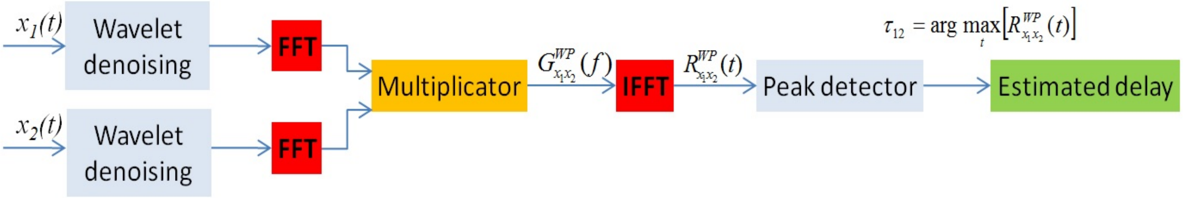


Figure 2.6 – Block Diagram for TDE Wavelet Pre-filtering Generalized Cross Correlation (WP-GCC)

The wavelet pre-filtering is seen as a new weighting function which modifies the cross-correlation input and it is done in a few steps:

1) using the discrete wavelet transform (DWT) the input signals are converted into the wavelet domain;

2) remove the noise by eliminating the coefficients which are smaller than a non-linear threshold;

3) apply the inverse discrete wavelet transform to obtain the filtered signals.

After the input signals were filtered by the wavelet denoising, the cross-correlation and maximum search is performed in order to estimate the delay, as shown in Figure 2.6.

2.5.2 *Wavelet Domain Inner Product*

The second TDE method, based on wavelet transform, was proposed in [Barsanti, 2003], exploiting the time variant property of the DWT. Despite the fact that the DWT filtering operations are time invariant and linear, combining them with the decimation operation (which is implied by the DWT) results in a time-variant system.

Based on the fact that for two delayed signals their DWT coefficients will correspond when the delay is zero, a comparison of the two sets of DWT coefficients over the entire possible delays was proposed. Then the estimated delay is selected as the one who produces the best correspondence between the two sets of DWT coefficients.

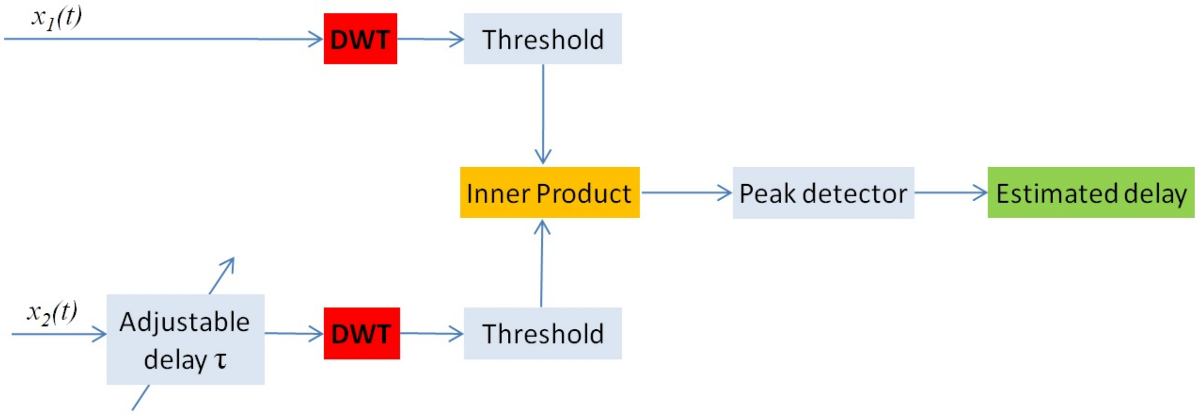


Figure 2.7 – Block Diagram for TDE Wavelet Domain Inner Product (WD-IP)

The processes presented in Figure 2.7 follows the next steps:

1) compute the DWT coefficients of the input signals $x_1(t)$ and $x_2(t-\tau)$;

2) apply the threshold for each DWT coefficients set in order to remove noise and to obtain the sparse representation;

3) with the resulting coefficients $C^1_{j,k}$ and $C^{2\tau}_{j,k}$ the similarity index *I(τ)* between the two sets of DWT coefficients is computed, using the inner product (vector dot product) calculated as in (2.83);

4) shift the input signals $x_2(t-\tau)$ by one sample and repeat the operational steps from 1 to 4 until the similarity index for all shifts is computed;

5) finally, estimate the delay as the automated introduced shift delay which maximizes the similarity index.

$$I(t) = \sum_j \sum_k C^1(j,k)\, C^{2t}(j,k) \tag{2.83}$$

In [Barsanti, 2003] it was shown that this method outperforms other approaches for TDE of extremely low short signals even at a lower SNR. However, this method comes with a major disadvantage with regard to the computing time. When the inner product and the shift operations of the correlator are separated no fast FFT-based correlation algorithm can be used. Hence the operations for similarity index and time shift have to be executed sequentially, by “brute force” [Barsanti, 2001].

3 ACCUMULATED GCC

After the presentation of TDE methods from the previous chapter, in order to choose the best TDE method, we should recall the specific demands and implementation issues of the BSS problem. As it was presented in the introduction part, before starting a BSS algorithm, we have to ensure that the input signals are aligned. Also, the introduced processing delay should not be more than 300ms, while the aircraft's pilot's conversation lasts just a few seconds in average. In these circumstances it is easy to observe that, despite the high accuracy results, the adaptive filtering and adaptive eigenvalue decomposition cannot be used in the BSS integration because of their long adaption time, which is usually more than a few seconds. In [Jacovitti, 1993] it was shown that difference functions offer good results only for high SNR and they could not yield reliable results for low SNR. Also, the wavelet based TDE methods are suited especially for transient signals and their implementation could lead to a long processing time. Thus, the only suitable method for the BSS integration remains the generalized cross-correlation. It offers good accuracy results, does not need any adaptation time, and does not have higher complexity than other methods.

Starting from these premises I developed two new GCC methods in [Marinescu, 2013b]. To extend the research on this topic, in this thesis the accumulating scheme will be applied to all well-known GCC functions. To the best of my knowledge, this technique was not presented by anyone in any other previous study.

3.1 PROPOSED ACCUMULATED GCC

In chapter 3 traditional GCC methods and the accumulated version of CSP in frequency domain (2.82) for a multi-frame analysis were presented. In order to improve the TDE accuracy from the ATM-VCS, the ρCSP (2.80) and ρCSPC (2.81) methods in [Marinescu, 2013b] were extended through the accumulation principle. These two new methods were called *accumulated ρ-Cross Power Spectrum Phase with Coherence* (*acc-ρCSPC*) and *accumulated ρ-Cross Power Spectrum Phase* (*acc-ρCSP*). In frequency domain their cross-power spectrum is

$$G_{acc-\rho CSPC}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)}{\left|G_{x_1x_2,k}\left(f\right)\right|^{\rho}+\min\left[\gamma_{x_1x_2,k}^{2}\left(f\right)\right]} \tag{3.1}$$

and

$$G_{acc-\rho CSP}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)}{\left|G_{x_1x_2,k}\left(f\right)\right|^{\rho}} \tag{3.2}$$

where K indicates the numbers of frames used in the analysis.

Thus, it is possible to take advantage of both ideas: accumulation scheme (Figure 2.5) and enhanced accuracy of ρCSPC and ρCSP. The new approach, summarized by (3.1) leads to faster computations compared to the previous methods, because it uses the accumulating scheme. It can also provide better results in unfavorable conditions for smaller frame sizes. Beside this, emphasis on speech regions from the spectrum is achieved by the whitening parameter (ρ), which reduces, at the same time, the impact of noise outside the speech region. For parts of the signal with small energy, the addition of the minimum coherence function limits the effect of a very small denominator [Rabinkin, 1996] [Shean, 2009].

The second method, *acc-ρCSP* was proposed as a faster variant of *acc-ρCSPC* for applications where relatively small energy signals are not encountered. In such circumstances the minimum coherence function can be omitted from (3.1) because in these cases there is no need to compute the coherence function and to search for its minimum. Hence, the computing complexity is reduced.

Over the years, several other studies discussed the details of TDE-GCC, like in [Youn, 1983] [Omologo, 1994, 1997] [Tianshuang, 1996] [Zetteberg, 2005] [Wilson, 2006] [Sun, 2010] [Sakhnov, 2011b, 2011c]. In this thesis the multi-frame analysis is extended with the accumulating cross-power spectrum scheme (Figure 2.5), not only for the above proposed methods, but also for the other well known GCC functions. Below the new accumulated cross power spectrum based formulae are presented:

$$G_{acc-CC}\left(f\right)=\sum_{k=1}^{K}G_{x_1x_2,k}\left(f\right) \tag{3.3}$$

$$G_{acc-Eckart}\left(f\right)=\sum_{k=1}^{K}\frac{\left|G_{x_1x_2,k}\left(f\right)\right|G_{x_1x_2,k}\left(f\right)}{\left[G_{x_1x_1,k}\left(f\right)-G_{x_1x_2,k}\left(f\right)\right]\left[G_{x_2x_2,k}\left(f\right)-G_{x_1x_2,k}\left(f\right)\right]} \tag{3.4}$$

$$G_{acc-Roth}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)}{G_{x_1x_1,k}\left(f\right)} \tag{3.5}$$

$$G_{acc-HT}\left(f\right)=\sum_{k=1}^{K}\frac{\gamma_{x_1x_2}^{2}\left(f\right)G_{x_1x_2,k}\left(f\right)}{\left|G_{x_1x_2,k}\left(f\right)\right|\left[1-\left|\gamma_{x_1x_2}\left(f\right)\right|^2\right]} \tag{3.6}$$

$$G_{acc-SCOT}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)}{\sqrt{G_{x_1x_1,k}\left(f\right)G_{x_2x_2,k}\left(f\right)}} \tag{3.7}$$

$$G_{acc-CPSm}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)}{\sqrt[m]{G_{x_1x_1,k}\left(f\right)G_{x_2x_2,k}\left(f\right)}} \tag{3.8}$$

$$G_{acc-HB}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)\left|G_{x_1x_2,k}\left(f\right)\right|}{G_{x_1x_1,k}\left(f\right)G_{x_2x_2,k}\left(f\right)} \tag{3.9}$$

$$G_{acc-Wiener}\left(f\right)=\sum_{k=1}^{K}\frac{G_{x_1x_2,k}\left(f\right)}{\left|\gamma_{x_1x_2}\left(f\right)\right|^2} \tag{3.10}$$

The following subchapters include a thorough analysis of the traditional GCC methods in different implementation schemes.

3.2 EXPERIMENTAL SETUP

In this section the preparations needed for a proper evaluation of the GCC methods are presented. Because the choices made at this stage could affect the evaluation results it is necessary to describe these setups. This will help the reader to understand the evaluation results and to adapt his further implementations for different situations.

3.2.1 EXPERIMENTAL DATABASE

In order to achieve reproducible evaluation results using accessible and well-known databases is recommended. Thus, the international noisy speech corpus Noizeus was used as the main database [Noizeus].

It contains 30 sentences produced by a group of 6 speakers: 3 males and 3 females. The sentences last a few seconds and were recorded in English at a sampling frequency of 8 kHz. It's worth mentioning that other well-known databases recorded at higher sampling frequency exist in the signal processing community. But, because the ATM-VCSs use a sampling rate of 8 kHz our experiments will also be performed at this sampling frequency.

Beside the clear version of the 30 sentences the Noizeus database contains also multiple versions corrupted with 8 different real-world noises, from Aurora database [Aurora] at 4 SNR levels (0, 5, 10, and 15 dB). The different noises are labeled as airport, babble, car, exhibition hall, restaurant, street, suburban train and train noise.

If we consider the 8 noisy variations of one clear sentence then we have $C_8^2 = 28$ pairs for 2 different noisy variations which could be seen as the input signals for our system. Furthermore, the sentences from Noizeus database can be combined at the same noise level for all 30 various sentences yielding a total of $C_8^2 \cdot 4 \cdot 30 = 3360$ signal pairs.

In further steps the database was split in two parts. The first half of the 15 sentences was used as the *development* database and the other half was used as the *evaluation* database.

3.2.2 EVALUATION METRICS

In this thesis, several methods were used for an extensive evaluation of the GCC methods. Because a perfect signals alignment is required in several further multichannel speech enhancement algorithms two accuracy metrics were used, the *accuracy rate* and the *error rate*. If we agree that a correct estimation is one where the estimated delay is equal to the real delay then the accuracy rate represents the ratio between the number of correct estimated delays and total number of estimations performed:

$$Accuracy_{rate} = \frac{No.correct\, estimations}{No.performed\, estimations} \tag{3.11}$$

Complementary to the above definition, the error rate denotes the ratio between the number of incorrectly estimated delays and the total number of performed estimations, as it is expressed bellow:

$$Error_{rate} = \frac{No.incorrect\, estimations}{No.performed\, estimations} \tag{3.12}$$

Another important statistical metric is the *relative error*, defined as

$$\delta_i = \frac{\tau_i - \tau}{\tau} \tag{3.13}$$

where δ_i represents the relative estimation error for the delay τ by the measured τ_i.

To characterize the variation from the average (mean) of the estimated delay, the fourth metric used is the *standard deviation of the relative error*, computed for the unbiased form as in the following formula:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\delta_i - \overline{\delta}\right)^2} \tag{3.14}$$

where $\overline{\delta}$ is the mean of the relative estimation error expressed in the next formula

$$\overline{\delta} = \frac{1}{N}\sum_{i=1}^{N}\delta_i \tag{3.15}$$

and *N* is the number of performed estimations.

For the above metrics the GCC methods were implemented and tested in Matlab. For the fifth metric, represented by the *computing time*, all methods were implemented in C language. In this way it was possible to obtain useful processing time results for any GCC approach. This metric is also important in the real-time implementation of the TDE GCC based solutions.

3.2.3 Calibrating Acc-ρCSPC and Acc-ρCSP

Recalling the definition of ρCSPC from (2.80) and ρCSP from (2.81) we observe easily that the whitening factor ρ is not defined. This is also true for acc-ρCSPC from (3.1) and for acc-ρCSP from (3.2). In [Rabinkin, 1996 $\rho = 0.75$ was chosen as an optimum value, while in [Shean, 2009] it varied from 0.78 to 0.9 depending on the SNR level. Hence, in these conditions it is naturally to search for a proper value for ρ to calibrate the above methods. To achieve this we have to run several TDE tests for different values of ρ. Then, the optimum value for ρ is chosen as the one which yields the maximum accuracy rate.

In this sense the development database was used to form the signals pairs. Then, for each signals pair 5 delay values (5, 10, 25, 50 and 100ms) was artificially introduced to be estimated by the acc-ρCSPC and acc-ρCSP methods. Considering these 5 delay values, the 8 noise types and half of the sentences we obtain a total number of $C_8^2 \cdot 4 \cdot 15 \cdot 5 = 8400$ tested pairs.

But ρ is not the only parameter to be set for the above TDE GCC methods. We have also to set the other parameters, which are common in all TDE GCC approaches. Depending on the applications' demands we have to choose adequate values for the *number of frames*, *frame size* and *overlap factor*. These three parameters influence all evaluation metrics and in real-time implementations they lead to trade-off between computing time, accuracy and fast response of the system. In this case, for calibrating acc-ρCSPC and acc-ρCSP by finding the optimum value for ρ, 4 averaging frames of 1024 samples each and an overlap factor of 25% were used.

The results of the simulation presented above are shown in Figure 3.1, presenting the accuracy rate characteristics of acc-ρCSPC. For different SNR the acc-ρCSPC has different accuracy rate characteristics with different optimum ρ. If for the SNR level of 0dB an optimum ρ is around 0.44, for 15 dB the optimum value is found around 0.74. This proves that the optimum ρ is highly influenced by the SNR and it is also highlighted the importance of the calibration stage. In some cases we may have a priori knowledge about the expected SNR level and we can then choose the optimum ρ. However, there are many cases when we do not know a priori the SNR level. In these situations the optimum value for ρ is the one which maximizes the accuracy rate for the average of the SNR levels. Because the evaluation experiments include general conditions the optimum value for ρ was chosen from the average SNR characteristic of the accuracy rate as $\rho = 0.73$.

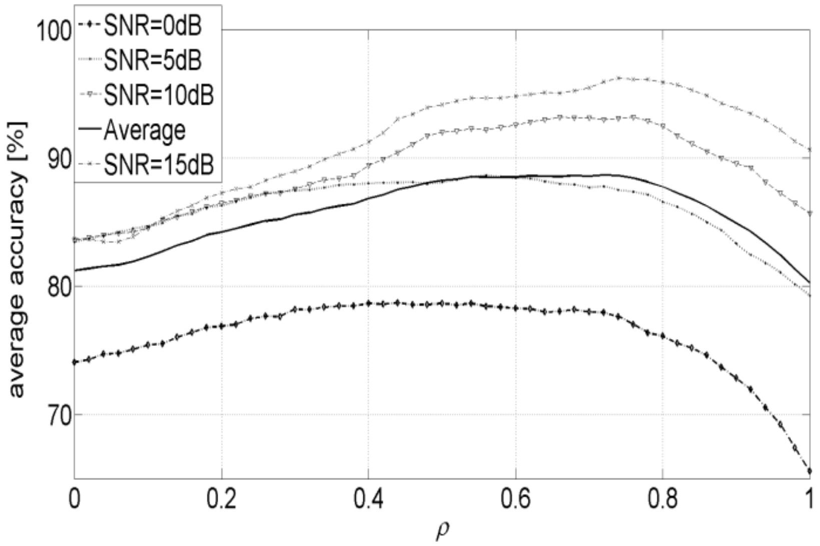


Figure 3.1 – The influence of SNR and ρ over the *acc-ρCSPC* accuracy

In situations in which the development database is not available or it is very limited, Figure 3.1 can be used to choose the optimum ρ, for a general or a specific narrow SNR domain.

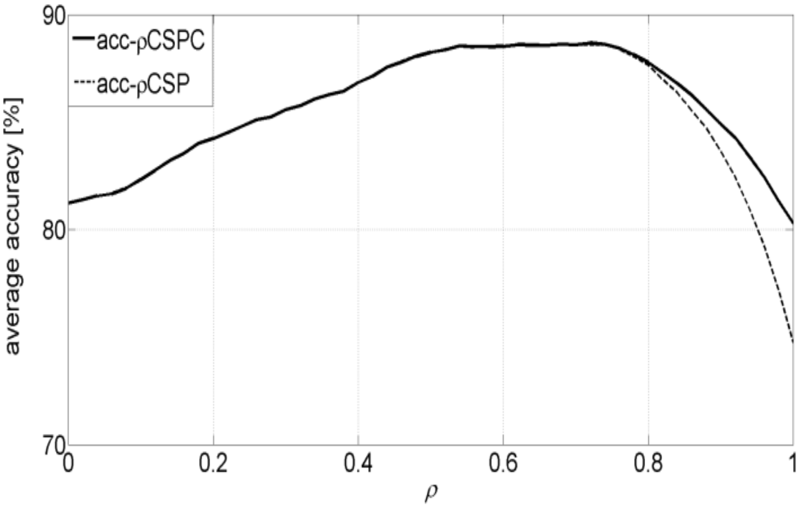


Figure 3.2 – The dependence of *acc-ρCSP* and *acc-ρCSPC* on *ρ*

In Figure 3.2 we can see the difference between the average SNR accuracy characteristics of the acc-ρCSPC and acc-ρCSP. It can be noted that for $\rho \in [0, 0.77]$ the accuracy of the two methods is practically equal. For $\rho > 0.77$ the acc-ρCSPC outperforms acc-CSP thanks to the introduction of the minimum coherence term in (2.81), implying this also for (3.1). Even so, since we have the same optimum ρ = 0.73 for both methods, which is less than 0.77, then we will have practically the same accuracy rate for acc-ρCSPC and acc-ρCSP, but a lower computing time for the second method because the coherence term was neglected. Nevertheless, more precise computing time results will follow in this chapter.

As a special note, the CPS-m method also has a variable parameter, m, which is usually set between 1 and 2. Even if a special calibration was not performed for this parameter, after performing some accuracy tests it was discovered that m = 4 yields a higher accuracy rate than other values.

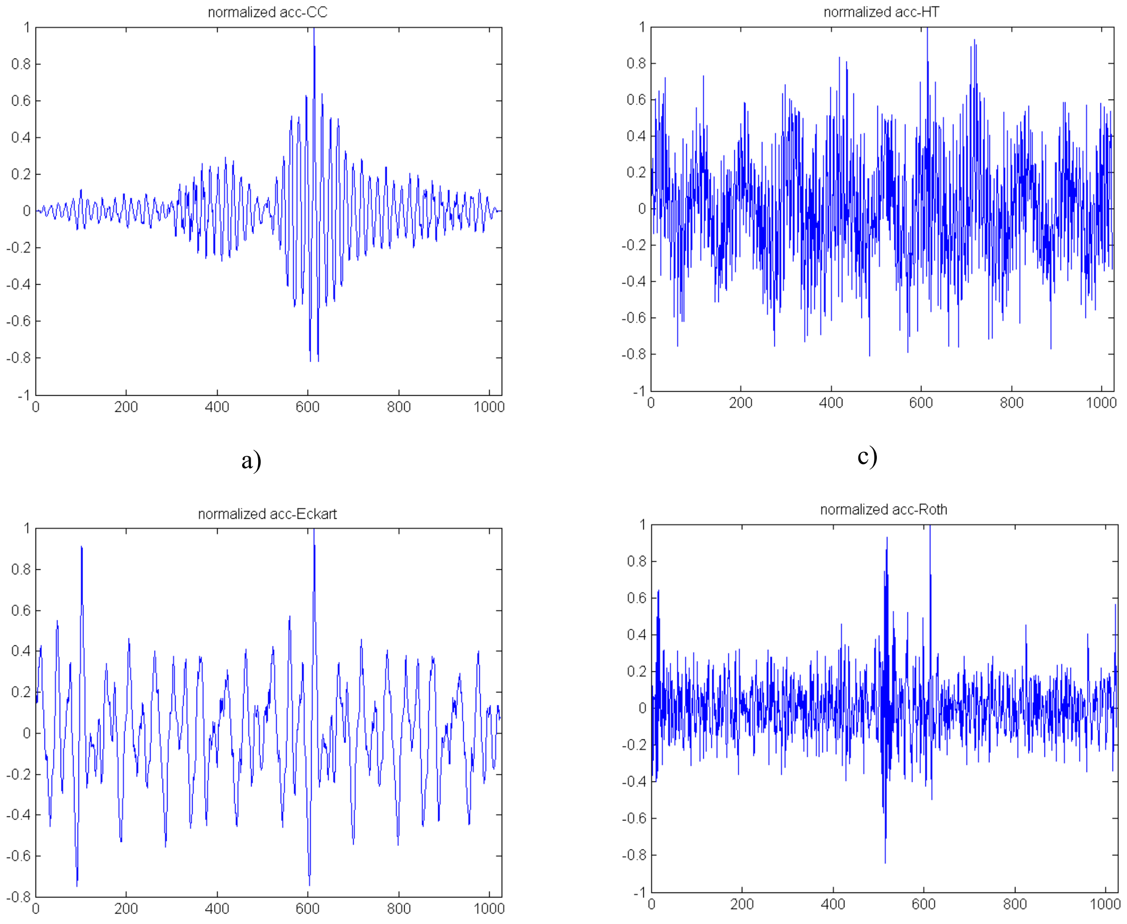
3.3 Experimental Results

After the calibration stage of *acc-ρCSPC* and *acc-ρCSP* we will analyze the proposed and previous TDE-GCC methods, based on the simulated results obtained for the presented metrics (accuracy and error rate, standard deviation of relative error, and computing time). First, we will evaluate the proposed acc-GCC methods by their statistical simulated results. Then we will include the results for the traditional GCC methods which use the basic TDE-GCC approach, like in Figure 2.3. The computing time of all GCC methods in several implementation schemes will end this evaluation section.

3.3.1 Evaluating Proposed Methods

As presented in section 0, the accumulated cross-power spectrum methods divide a large analysis window into several smaller frames. Next, they compute their specific cross-power spectrum for every frame adding the results to the previous calculations. After these operations are done for the last frame, the accumulated cross-power spectrum is obtained. Next, this is used to return in the time domain with the IFFT and then to find the estimated delay.

Before presenting any numerical result it is useful to look over the normalized accumulated cross-power spectra of all GCC methods. These are presented in Figure 3.3 and were computed for a large analysis window of 2048 samples, split in 4 smaller frames of 512 samples. In these cases the frames were not overlapped. The artificially introduced delay was 10 ms (equivalent to 80 samples for a sampling rate of 8 kHz).

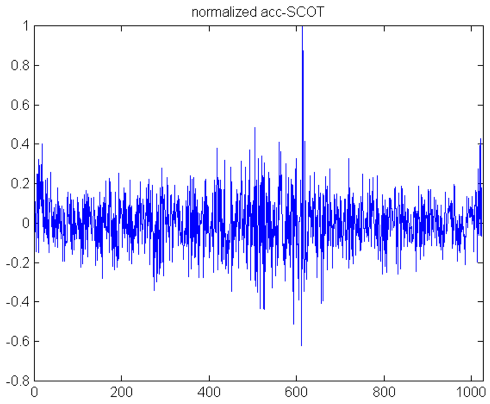


a)

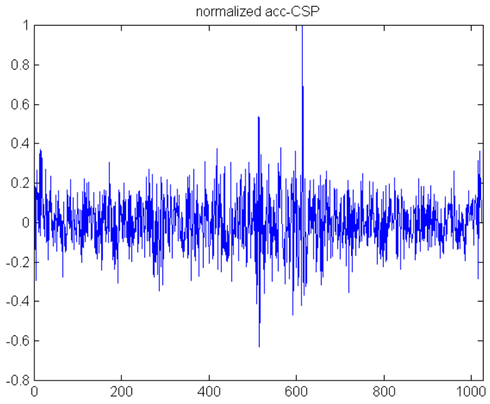
c)

b)

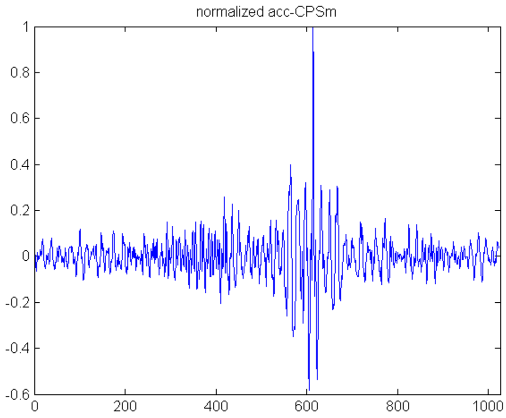
d)


normalized acc-SCOT
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
200
400
600
800
1000

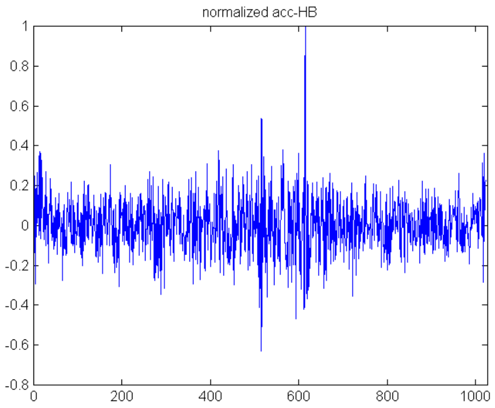
e)


normalized acc-CSP
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
200
400
600
800
1000

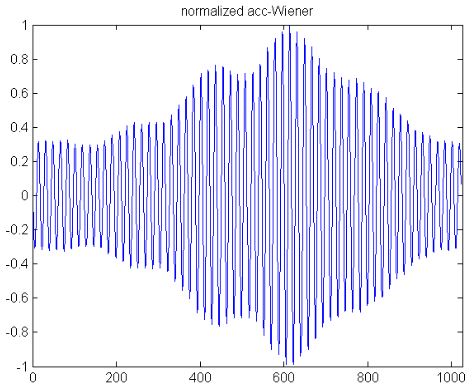
f)


normalized acc-CPSm
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
200
400
600
800
1000

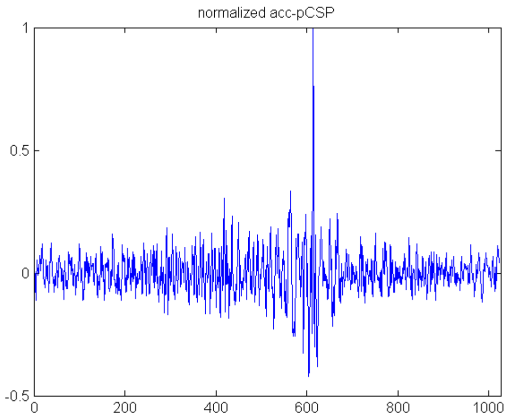
g)


normalized acc-HB
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
200
400
600
800
1000

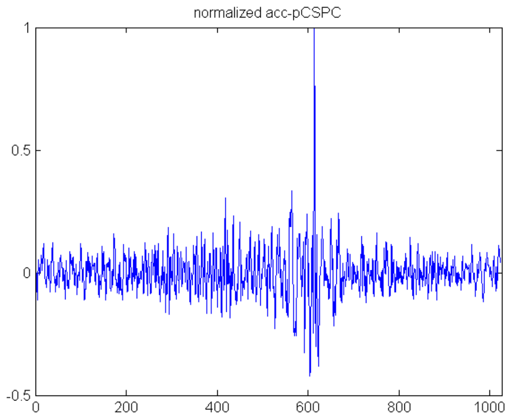
h)


normalized acc-Wiener
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
200
400
600
800
1000

i)


normalized acc-pCSP
1
0.5
0
-0.5
0
200
400
600
800
1000

j)



k)

Figure 3.3 – Accumulated Cross Power Spectrum for: a) cross-correlation, b) Eckart, c) HT, d) Roth, e) SCOT, f) CSP, g) CPS-*m*, h) HB, i) Wiener, j) *ρ*CSP, k) *ρ*CSPC

It can be seen than not all GCC methods improve the cross-power spectrum characteristic of the normal cross-correlation in order to assure an accurate determination of the maximum peak.

The analysis of proposed methods continues with several experiments intended to reveal the difference between all accumulated GCC approaches. In further experiments the same setup was kept: analysis window of 2048 sample, split in 4 frames of 512 samples without overlapping. At a sampling frequency of 8 kHz a single frame represents 64 ms of signal. Thus, the introduced delay varied from 5 to 50 ms. In order to achieve a clear analysis these delays were separated in two categories: small delays (from 5 to 30 ms) and large delays (from 35 to 50 ms). This division was made in order to obtain results for delays shorter than half of the frame size and separated results for delays longer than half of the frame.

The error rate, relative error and standard deviation relative error for all accumulated GCC approaches will be presented and commented in next subsections.

3.3.1.1 Error Rate for Accumulated GCC Methods

In the next 4 tables, from Table 3.1 to Table 3.4 we can see the error rates at different SNR and different delays for all accumulated GCC methods. The results, computed with the (3.12) formula, prove that for higher SNR the accumulated GCC methods yield lower error rates.

We can also notice that not all the proposed methods perform better than the basic cross-correlation, which was already shown in several studies like [Hassab, 1980][Hassab, 1981][Sakhnov, 2011b]. This could be explained by the fact that the papers in which some methods were proposed contain only mathematical presentation and no simulated results.

TABLE 3.1 – ERROR RATE FOR ACCUMULATED GCC METHODS, AT SNR = 0DB

	Error rate (%)									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.48**	**0.81**	1.19	1.57	6.26	14.24	23.48	34.48	53.83	72.43
Eckart	70.21	79.79	83.69	88.50	91.76	95.45	97.29	96.83	98.48	99.69
Roth	57.00	63.62	73.79	77.36	83.90	87.36	92.60	95.31	97.31	99.14
HT	90.26	91.00	94.24	96.17	96.33	98.02	98.45	99.26	99.52	99.60
SCOT	59.88	60.10	58.98	63.93	71.57	77.38	82.38	88.19	93.19	96.67
CSP	13.48	13.67	19.71	24.67	33.17	41.50	60.05	70.02	76.60	88.98
CPS-m	2.38	0.98	**0.95**	**1.02**	2.74	**4.50**	10.45	18.40	34.10	61.67
HB	13.48	13.67	19.71	24.67	33.17	41.50	60.05	70.02	76.60	88.98
Wiener	25.81	39.40	49.10	59.95	65.74	74.50	82.07	89.67	91.33	98.26
ρCSP	3.67	**0.81**	1.48	1.52	**2.71**	**4.50**	**9.90**	**18.31**	**31.71**	**57.33**
ρCSPC	3.88	0.98	1.60	2.69	3.50	5.26	10.62	21.17	34.95	61.86

TABLE 3.2 – ERROR RATE FOR ACCUMULATED GCC METHODS, AT SNR = 5DB

	Error rate (%)									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	3.36	7.67	17.48	28.12	50.26	68.71
Eckart	65.83	68.36	75.12	81.81	86.10	90.38	93.24	96.10	98.12	98.81
Roth	57.00	63.62	73.79	77.36	83.90	87.36	92.60	95.31	97.31	99.14
HT	83.40	87.93	89.29	91.33	93.88	95.81	97.50	98.12	99.33	99.40
SCOT	58.17	52.45	51.07	60.95	61.40	70.19	71.14	79.33	86.83	90.69
CSP	7.02	7.43	8.29	16.38	15.76	24.26	42.17	53.14	63.64	73.33
CPS-m	0.71	0.10	0.10	0.21	**0.12**	1.19	3.10	8.62	21.10	44.67
HB	7.02	7.43	8.29	16.38	15.76	24.26	42.33	53.14	63.64	73.33
Wiener	12.83	25.98	34.79	46.14	56.98	67.93	79.05	86.57	89.31	96.62
ρCSP	1.57	**0.00**	**0.00**	0.26	0.33	**1.12**	**2.50**	6.95	**15.55**	**36.05**
ρCSPC	1.57	**0.00**	**0.00**	1.33	2.07	1.26	3.48	**6.86**	17.21	40.86

TABLE 3.3 – ERROR RATE FOR ACCUMULATED GCC METHODS, AT SNR = 10DB

	Error Rate									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	2.52	7.57	16.81	25.71	47.64	67.26
Eckart	63.00	60.10	69.17	75.07	79.50	83.90	91.57	95.36	95.57	98.95
Roth	22.50	35.95	42.12	51.29	55.98	68.55	82.55	87.50	93.19	97.10
HT	76.74	79.83	84.71	88.38	91.55	92.76	95.02	96.55	98.43	98.50
SCOT	56.10	51.02	51.45	54.88	57.69	65.93	59.05	72.26	82.55	88.02
CSP	4.76	6.48	6.33	10.62	10.38	15.00	30.33	43.71	47.19	66.50
CPS-m	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.81	4.90	15.57	38.52
HB	4.76	6.48	6.33	10.62	10.38	15.00	30.33	43.71	47.19	66.50
Wiener	10.50	28.43	35.43	48.14	56.43	67.43	73.24	86.83	88.43	97.14
ρCSP	0.43	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.67**	**2.83**	**8.36**	**25.98**
ρCSPC	0.43	**0.00**	**0.00**	0.52	1.86	0.19	**0.67**	**2.83**	8.93	28.19

TABLE 3.4 – ERROR RATE FOR ACCUMULATED GCC METHODS, AT SNR = 15DB

	Error Rate (%)									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	2.50	7.50	16.86	25.14	49.90	66.33
Eckart	60.33	58.50	65.88	74.62	74.90	80.45	87.76	92.33	94.98	96.05
Roth	13.17	24.05	33.24	39.98	47.21	57.74	77.24	82.86	87.98	93.57
HT	69.69	75.69	78.19	86.12	88.69	91.76	93.62	95.71	97.00	97.74
SCOT	52.93	53.19	51.71	51.67	57.90	57.26	52.71	64.83	73.83	84.21
CSP	2.21	4.38	4.90	7.40	8.40	5.71	20.98	38.67	38.98	55.07
CPS-m	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	4.33	14.74	36.19
HB	2.21	4.38	4.90	7.40	8.40	5.71	20.98	38.67	38.98	55.24
Wiener	11.24	36.38	43.29	55.64	50.10	67.45	75.10	83.90	88.50	96.17
ρCSP	0.10	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.21	2.19	**5.00**	**23.38**
ρCSPC	0.26	**0.00**	**0.00**	0.98	3.14	**0.00**	0.21	**1.88**	6.33	28.81

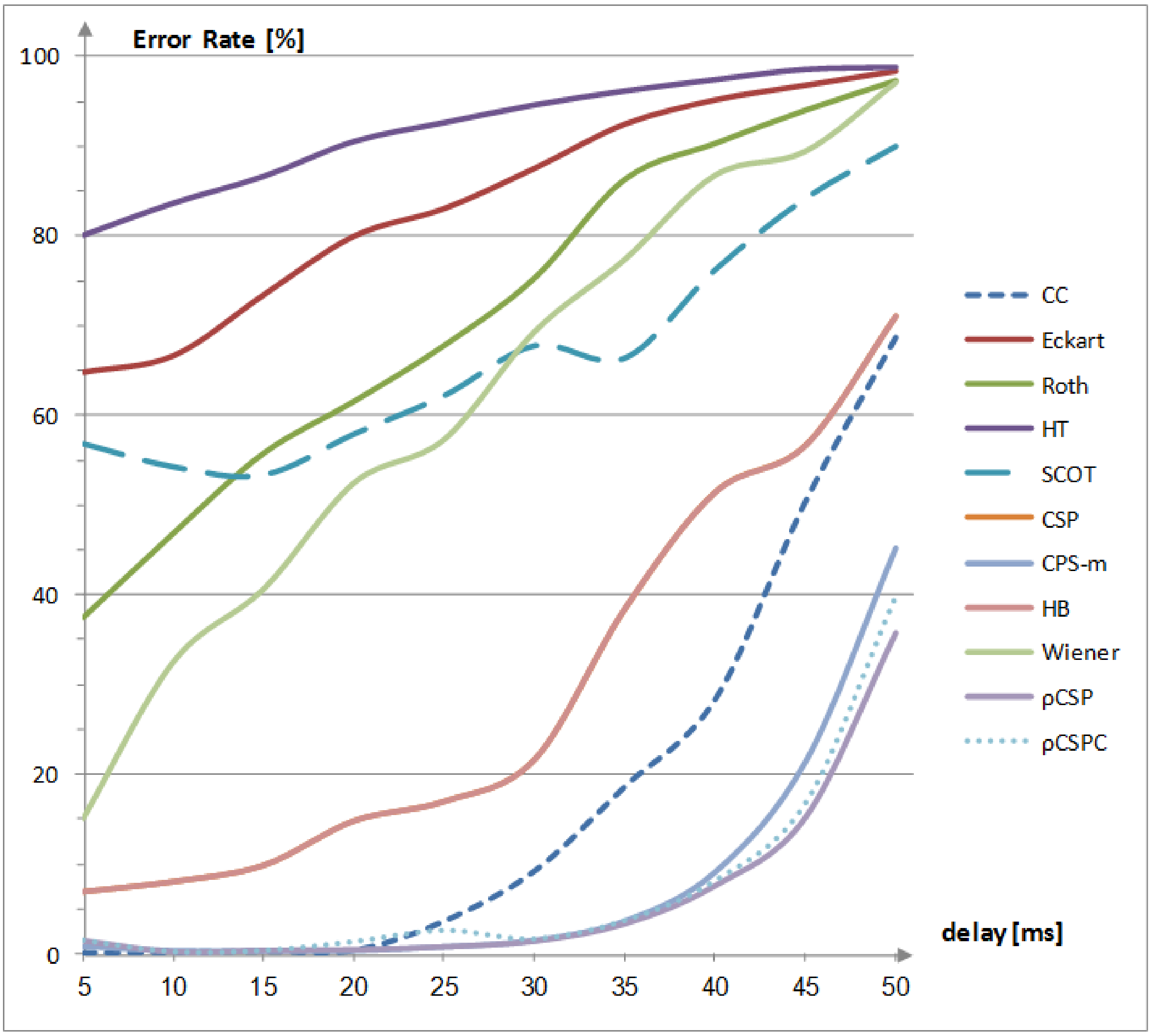


Figure 3.4 – Average error rates for accumulated GCC methods

The average error rate for all accumulated GCC methods is shown in Figure 3.4. It was computed as the average of error rates for all 4 SNR levels (0, 5, 10, and 15 dB).

From Table 3.1 to Table 3.4, and Figure 3.4 we can see that accumulated approaches for Eckart, Roth, HT, SCOT, and Wiener method are not suitable for accurate TDE, because they lead to high error rates even for relatively small delays. The accumulated HT and CSP approaches have almost the same error rates for all SNR levels. This could be explained by the similarity between weighting functions (2.76) and (2.78).

The CC, CPS-m, ρCSP, and ρCSPC offer smaller error rates (<5%) for small delays, with a small exception for CC. Moreover, while the CC approach outperforms all other methods with respect to the error rate for very small delays, for small and larger delays it cannot yield error rates as other approaches do. For relatively small delays these last 4 methods could be used with great confidence. For larger delays the error rates increase for all previous 4 methods, with a higher increasing rate for CC. The lowest error rate for large delays is achieved by acc-ρCSP and acc-ρCSPC, which were proposed and analyzed recently in [Marinescu, 2013a, 2013b].

The influence of the number of frames on the accumulated GCC error rate is highlighted in Table 3.5. The GCC error rates were computed for 1, 4 and 8 frames of 512 samples, at 15 dB SNR. It is shown that the error rate decreases when more frames are used. This is due to the fact that the accumulated cross power spectrum domain keeps the spectral information over multiple frames. Therefore, the correlation between the frames is maintained.

TABLE 3.5 – ACCUMULATED GCC ERROR RATE DEPENDENCY BY THE NUMBER OF FRAMES

	Number of frames		
GCC	**1**	**4**	**8**
CC	37.17	4.79	2.21
Eckart	63.67	66.95	59.69
ROTH	53.8	29.69	21.22
SCOT	64.92	56.68	45.96
CSP	26.35	5.86	1.66
CSP-m	24.94	1.72	**0.05**
HT	76.97	80.36	80.03
HB	26.35	5.86	1.66
Wiener	68.42	46.22	26.56
ρCSP	**18.2**	**0.6**	0.56
ρCSPC	19.3	**0.6**	0.56

Because *acc-ρCSP* outperforms all other accumulated GCC approaches and *acc*-CC should be the fastest with respect to computing time for these two methods the dependency between the accuracy rate and the estimated delay was further investigated. In order to obtain the results presented in Figure 3.5, 4 frames of 512 samples each were used. Then the average accuracy was computed from 0, 5, 10 and 15 dB SNR for *acc-ρCSP* and *acc*-CC.

For delays of 50 ms, which represent 78% of the frame size length (64 ms), the accuracy rate for *acc-ρCSP* is around 75%, while for *acc*-CC is less than 35%. This confirms that while most of GCC methods are able to estimate accurate delays smaller than half of the frame size, *acc-ρCSP* outperforms them and continues to provide reasonable accuracy for larger delays.

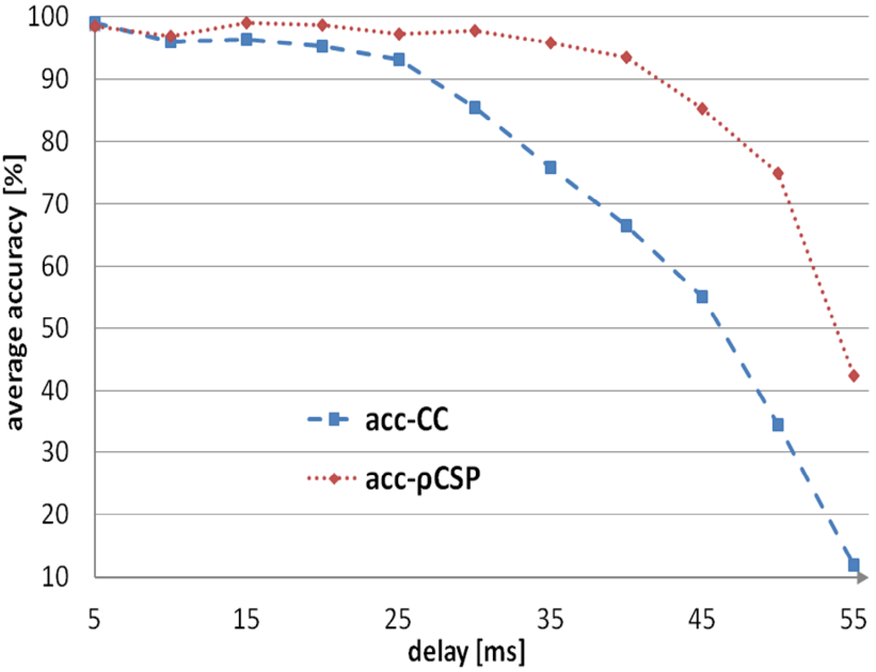


Figure 3.5 – Accuracy Rate Comparison between acc-CC and acc-ρCSP

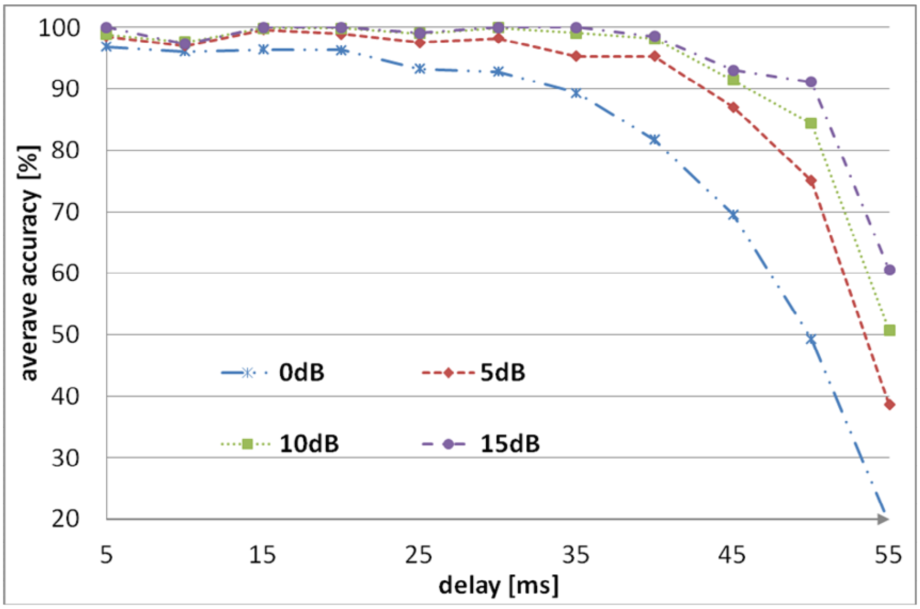


Figure 3.6 – The influence of SNR and delay over the *acc-ρCSP* accuracy

For a better characterization of the proposed *acc-ρCSP* the accuracy rate's variation with the SNR and delay is presented in Figure 3.6. It is shown that the higher the SNR is, the higher the accuracy gets. For delays up to 50% of the frame size, the difference between accuracies on various levels of SNR remains almost the same. Once the delay increases over 50% of the frame size, the accuracy decreases much faster for lower SNR.

3.3.1.2 Relative Error for Accumulated GCC Methods

The relative error for different SNR and different delays for all accumulated GCC methods is presented in the following 4 tables, from Table 3.6 to Table 3.9. The results for this metric were computed with formula (3.13). Similar to the previous metric results (error rate results) the higher SNR is, the lower the accumulated GCC relative error is.

The average of the relative error for all 4 SNR levels for all accumulated GCC methods is shown in Figure 3.7. It synthesizes the results from Table 3.6 to Table 3.9 offering a visual intuitive report.

TABLE 3.6 – RELATIVE ERROR FOR ACCUMULATED GCC METHODS, AT SNR = 0DB

	Relative Error [%]									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.06**	0.25	0.38	0.49	2.85	5.05	7.16	10.00	18.65	30.21
Eckart	59.59	74.43	82.65	89.27	89.33	95.25	102.24	94.98	101.62	101.98
Roth	110.50	74.49	84.26	83.11	83.13	92.34	92.18	90.93	100.07	97.05
HT	63.38	97.29	93.50	101.71	94.97	91.52	101.59	96.07	98.68	102.97
SCOT	21.54	28.13	25.13	35.52	46.81	49.25	70.59	79.90	84.00	91.60
CSP	11.96	24.02	17.27	26.40	34.01	44.97	63.28	74.55	77.99	90.78
CPS-m	1.77	0.19	**0.32**	**0.18**	**0.99**	**1.18**	**3.62**	**8.19**	**15.59**	**29.94**
HB	24.13	24.02	27.42	34.01	43.14	53.09	71.54	83.68	84.42	102.35
Wiener	17.01	17.48	16.58	20.91	23.61	32.35	42.31	45.07	47.98	56.59
ρCSP	2.87	0.19	0.69	0.60	1.75	3.66	8.06	14.51	24.85	43.34
ρCSPC	2.72	**0.16**	0.94	2.06	2.60	4.32	9.03	17.54	28.63	48.77

TABLE 3.7 – RELATIVE ERROR FOR ACCUMULATED GCC METHODS, AT SNR = 5DB

	Relative Error [%]									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	1.58	3.19	5.64	7.44	14.19	23.26
Eckart	58.45	66.94	80.84	76.85	85.60	89.58	95.55	92.66	100.36	98.04
Roth	51.40	34.13	55.68	69.39	71.25	89.97	85.55	99.58	100.63	100.23
HT	41.92	107.75	85.00	97.03	97.37	88.28	99.75	98.22	101.19	102.20
SCOT	18.70	7.17	15.23	27.52	31.91	42.04	55.72	71.85	74.96	85.01
CSP	7.26	6.38	4.96	18.01	15.17	29.81	48.56	51.32	66.57	80.15
CPS-m	0.27	**0.00**	**0.00**	**0.12**	**0.00**	**0.17**	**0.83**	**1.79**	**6.79**	**15.81**
HB	7.26	6.38	12.06	18.77	16.39	33.36	54.99	60.83	77.06	89.29
Wiener	12.35	10.85	10.54	13.95	17.42	23.88	34.49	39.08	44.57	51.69
ρCSP	1.11	**0.00**	**0.00**	0.26	0.02	0.39	1.37	4.45	9.48	25.22
ρCSPC	1.11	**0.00**	**0.00**	1.45	1.57	0.46	2.38	4.69	11.51	32.99

Table 3.8 – RELATIVE Error for Accumulated GCC Methods, at SNR = 10dB

	Relative Error [%]									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	1.26	3.35	5.94	7.50	13.54	20.06
Eckart	53.62	49.50	77.84	78.66	84.30	78.47	90.01	90.99	97.61	101.96
Roth	17.69	13.52	62.25	54.26	66.99	91.83	87.86	96.32	100.34	103.07
HT	76.50	87.02	91.79	93.77	89.96	92.67	96.24	98.50	99.66	99.02
SCOT	14.80	7.45	14.03	23.40	24.66	35.20	40.72	59.99	62.03	83.63
CSP	4.65	7.82	3.68	10.12	8.44	19.55	30.29	45.89	52.94	79.25
CPS-m	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.15**	**0.69**	**4.57**	**11.69**
HB	4.65	12.39	9.77	11.64	10.88	20.06	38.56	53.50	61.74	85.64
Wiener	5.07	10.19	12.25	13.90	18.57	22.49	31.43	40.77	43.38	51.39
ρCSP	0.56	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.60	1.99	5.62	20.53
ρCSPC	0.56	**0.00**	**0.00**	0.71	1.67	0.24	0.60	2.09	6.38	23.27

Table 3.9 – RELATIVE Error for Accumulated GCC Methods, at SNR = 15dB

	Relative Error [%]									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	1.55	3.51	6.26	7.66	13.93	20.88
Eckart	68.56	32.62	75.87	80.77	76.09	81.45	75.54	90.65	104.32	100.70
Roth	-1.64	8.18	38.92	53.74	42.92	63.78	75.91	92.28	93.88	97.57
HT	62.35	87.20	89.13	89.08	95.36	100.86	92.99	96.74	99.55	97.49
SCOT	16.83	12.57	13.24	21.31	21.79	24.80	33.08	53.11	54.06	80.53
CSP	2.26	4.52	2.24	8.29	7.89	8.85	21.86	39.16	46.20	63.43
CPS-m	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.72**	5.07	**10.39**
HB	2.26	4.52	8.33	8.29	9.11	9.36	25.34	51.72	47.56	66.44
Wiener	3.96	11.74	17.11	17.80	16.67	22.51	32.53	36.75	43.47	49.84
ρCSP	0.10	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.36	2.02	**4.75**	20.30
ρCSPC	0.17	**0.00**	**0.00**	0.95	3.10	**0.00**	0.36	1.79	6.22	26.33

Looking over Table 3.6 to Table 3.9 and Figure 3.7 we notice that the trend of the results of this metric is similar with the one of the error rate metrics. HT, Eckart and Roth acc-GCC approaches yield a very high error rate even at high SNR levels and for small delays. SCOT, Wiener, CSP, and HB perform much better and they can offer reliable results for high SNR and small delays. While for the previous metric CSP and HB offered almost the same results, now, CSP yields a smaller error rate than HB. This means that when these methods do not estimate the delay correctly, the estimated delay yielded by the CSP method is closer to the real delay than the estimated delay achieved through the HB approach is.

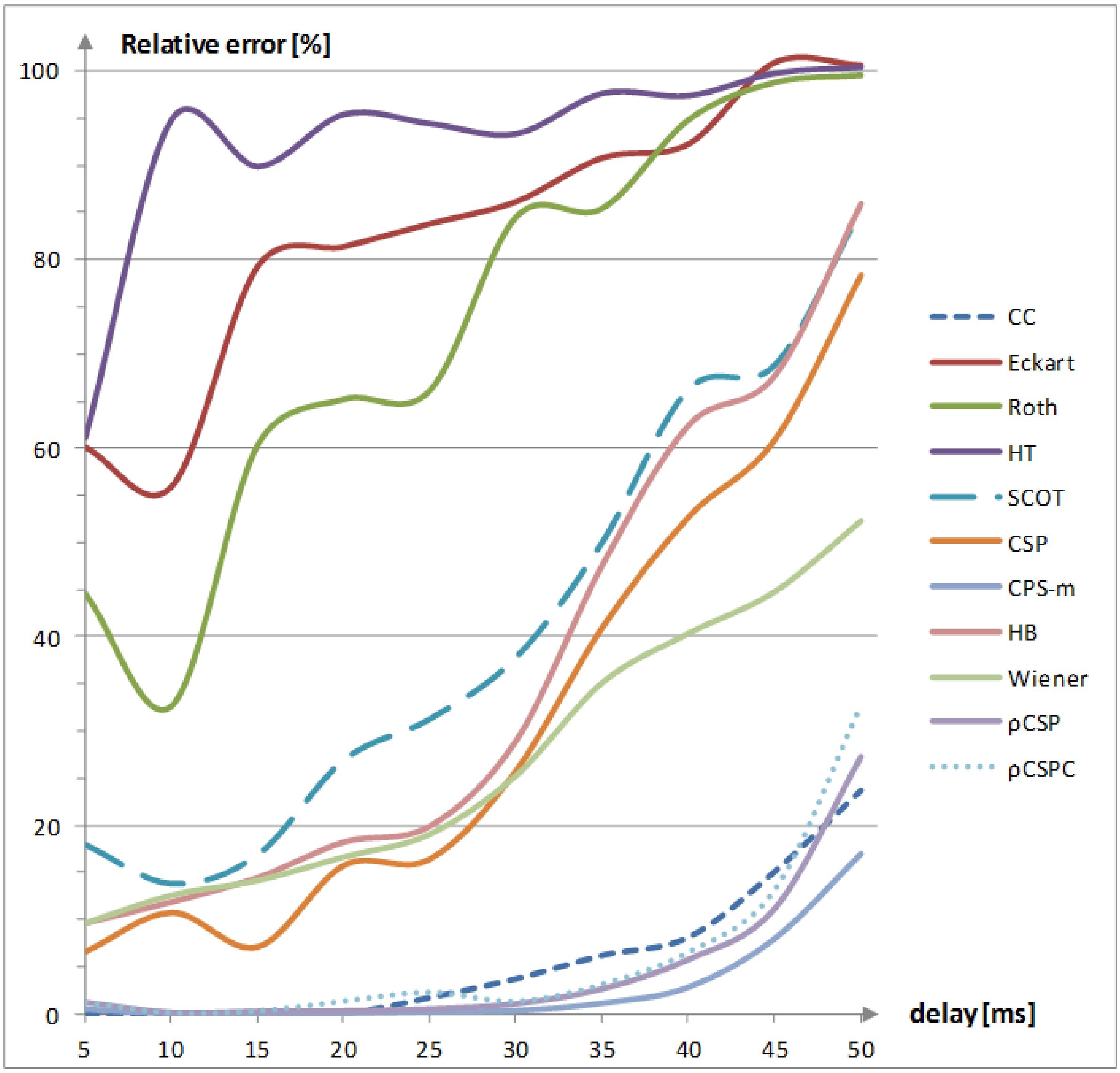


Figure 3.7 - Average relative error for accumulated GCC methods

The group of the last 4 methods, CC, CPS-m, ρCSP, and ρCSPC, estimates the delay with a small relative error. For small delays the relative error is usually < 5%. For large delays it starts to increase for all 4 approaches, but CPS-m outperforms all other methods.

In the tables we could find also several results with the relative error equal to 0. These are expected results, because the error rate was also 0 for the same configuration. These cases represent the correct time delay estimations.

3.3.1.3 Standard Deviation of Relative Error for Accumulated GCC Methods

The statistical analysis of the accumulated GCC methods for TDE would not be complete if the standard deviation of relative error would not be used. Using this metric we can characterize every method by their variation around the average estimation. Thus, the simulated results of this metric for all accumulated GCC methods for 4 SNR levels are presented in Table 3.10 to Table 3.13. Figure 3.8 points out the differences between the average standard deviations of relative error of all accumulated GCC methods. This average also represents the simulated results over all 4 SNR levels.

Analyzing results from Table 3.10 to Table 3.13 and from Figure 3.8 we obtain important details about the acc-GCC methods. For CC, CPS-m, ρCSP, and ρCSPC, the standard deviation of relative error increases with the delay. This is a normal trend because the higher the delay is, the higher the probability of wrong estimations gets and the higher the variance of the results is. However, for HT, Roth, Eckart and Wiener this is not true, moreover the standard deviation of relative error gets lower for larger delays. Also SCOT, HB and CSP show only a slowly increasing trend for this metric with the increased delay.

Table 3.10 – Standard Deviation of Relative Error for Accumulated GCC Methods, at SNR = 0dB

	STD									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	12.03	7.57	6.26	4.91	12.71	14.96	**16.34**	**19.43**	**27.23**	**33.31**
Eckart	392.71	241.59	196.12	143.68	120.04	111.08	92.22	81.88	73.90	65.67
Roth	478.61	265.64	198.94	155.04	129.08	113.93	94.88	82.11	72.08	62.64
HT	659.28	347.58	232.55	173.85	144.32	120.65	105.05	89.05	81.30	73.14
SCOT	129.70	121.14	67.45	79.71	80.02	82.97	71.21	70.24	62.08	58.71
CSP	129.73	112.18	79.77	81.10	76.67	89.77	78.59	79.54	70.27	67.56
CPS-m	**11.72**	2.80	**4.86**	**3.49**	**8.89**	**8.50**	16.76	25.00	31.95	38.88
HB	137.33	112.18	89.16	87.44	83.70	94.29	81.96	81.92	71.99	68.17
Wiener	122.78	66.96	49.83	46.98	44.52	45.91	52.03	46.16	40.44	37.68
ρCSP	15.51	**2.76**	8.15	6.67	13.04	19.31	27.97	35.95	44.22	49.42
ρCSPC	14.77	2.81	9.61	13.78	16.23	20.51	29.29	38.57	45.70	48.77

Table 3.11 – Standard Deviation of Relative Error for Accumulated GCC Methods, at SNR = 5dB

	STD									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	9.72	12.36	14.04	14.47	21.06	**26.62**
Eckart	346.36	232.92	188.64	128.27	120.25	105.37	93.97	84.36	75.31	67.76
Roth	431.47	235.62	165.86	128.38	115.97	111.02	93.53	81.55	70.75	70.12
HT	618.14	334.90	233.32	176.62	139.61	122.33	103.55	91.40	82.34	72.17
SCOT	37.83	47.59	38.19	52.14	59.46	76.54	69.65	71.32	63.89	66.61
CSP	25.79	37.34	45.43	46.90	50.17	77.54	75.31	69.79	73.62	76.05
CPS-m	4.63	0.04	0.03	3.41	**0.02**	**2.05**	**10.19**	**8.32**	**19.04**	28.59
HB	25.79	37.34	58.06	48.21	52.18	80.63	79.51	75.57	77.55	77.88
Wiener	107.15	48.52	34.95	32.12	31.24	37.63	39.87	38.16	37.97	35.90
ρCSP	9.30	**0.00**	**0.00**	4.90	0.54	5.45	11.16	20.01	28.43	41.81
ρCSPC	9.30	**0.00**	**0.00**	11.88	12.37	5.82	15.15	20.48	30.47	46.63

Table 3.12 – STANDARD Deviation of Relative Error for Accumulated GCC Methods, at SNR = 10dB

	STD									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	8.55	12.65	14.63	14.90	21.07	**21.81**
Eckart	299.92	192.77	173.04	123.75	122.46	106.09	101.46	83.90	80.95	73.71
Roth	250.12	185.00	174.33	133.70	111.44	114.57	96.07	86.25	78.47	74.59
HT	605.18	321.28	224.67	167.09	137.67	114.09	105.49	92.97	83.57	72.62
SCOT	34.62	44.49	40.58	46.26	54.83	69.36	57.43	71.36	66.67	70.00
CSP	20.21	69.02	36.79	35.18	34.86	69.20	61.27	70.75	73.18	20.21
CPS-m	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**1.70**	**3.25**	**15.44**	25.05
HB	20.21	74.77	49.96	38.87	40.71	69.78	69.59	75.99	78.16	80.26
Wiener	27.56	22.59	24.21	26.52	28.23	29.58	35.88	40.22	36.84	35.73
*ρ*CSP	6.60	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	7.75	13.74	22.55	40.42
*ρ*CSPC	6.60	**0.00**	**0.00**	8.43	12.81	4.88	7.75	14.11	24.00	42.59

Table 3.13 – STANDARD Deviation of Relative Error for Accumulated GCC Methods, at SNR = 15dB

	STD									
acc-GCC	Small delays (ms)						Large delays (ms)			
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
CC	**0.00**	**0.00**	**0.00**	**0.00**	9.43	12.97	15.20	15.27	21.19	**23.47**
Eckart	279.51	218.84	147.93	122.69	112.77	110.30	99.79	86.73	81.46	75.72
Roth	194.78	173.25	134.91	125.42	98.39	105.73	96.45	80.19	79.02	66.42
HT	537.09	303.44	219.39	171.15	137.94	121.08	101.96	91.36	81.73	72.79
SCOT	35.61	33.82	33.91	50.03	46.11	51.61	57.01	66.94	67.93	73.36
CSP	14.88	20.77	34.87	30.45	28.75	41.73	50.62	68.36	68.07	76.52
CPS-m	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**3.47**	**16.91**	24.25
HB	14.88	20.77	48.74	30.45	32.41	42.82	55.64	77.77	69.12	77.94
Wiener	18.86	32.58	51.61	26.03	25.86	32.32	40.75	37.01	40.48	32.63
*ρ*CSP	2.85	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	5.99	14.09	21.26	39.68
*ρ*CSPC	3.52	**0.00**	**0.00**	9.72	17.33	**0.00**	5.99	13.25	24.96	43.94

The above statement could seem bizarre at first glance, but when we corroborate these results with the previous ones from the error rate we find the explanation. For CC, CPS-m, ρCSP and ρCSPC there is no issue for larger delays, whereas for almost all other methods we notice that the error rate is very high (over 60%) in these cases. At these error rates these methods do not offer reliable results therefore it is clear that the decrease of standard deviation of relative error is due to the fact that almost all delays were much frequently estimated incorrectly. Moreover, the incorrect estimation delays are much frequently estimated with the same error, resulting in smaller variation of the relative error and then a smaller standard deviation.

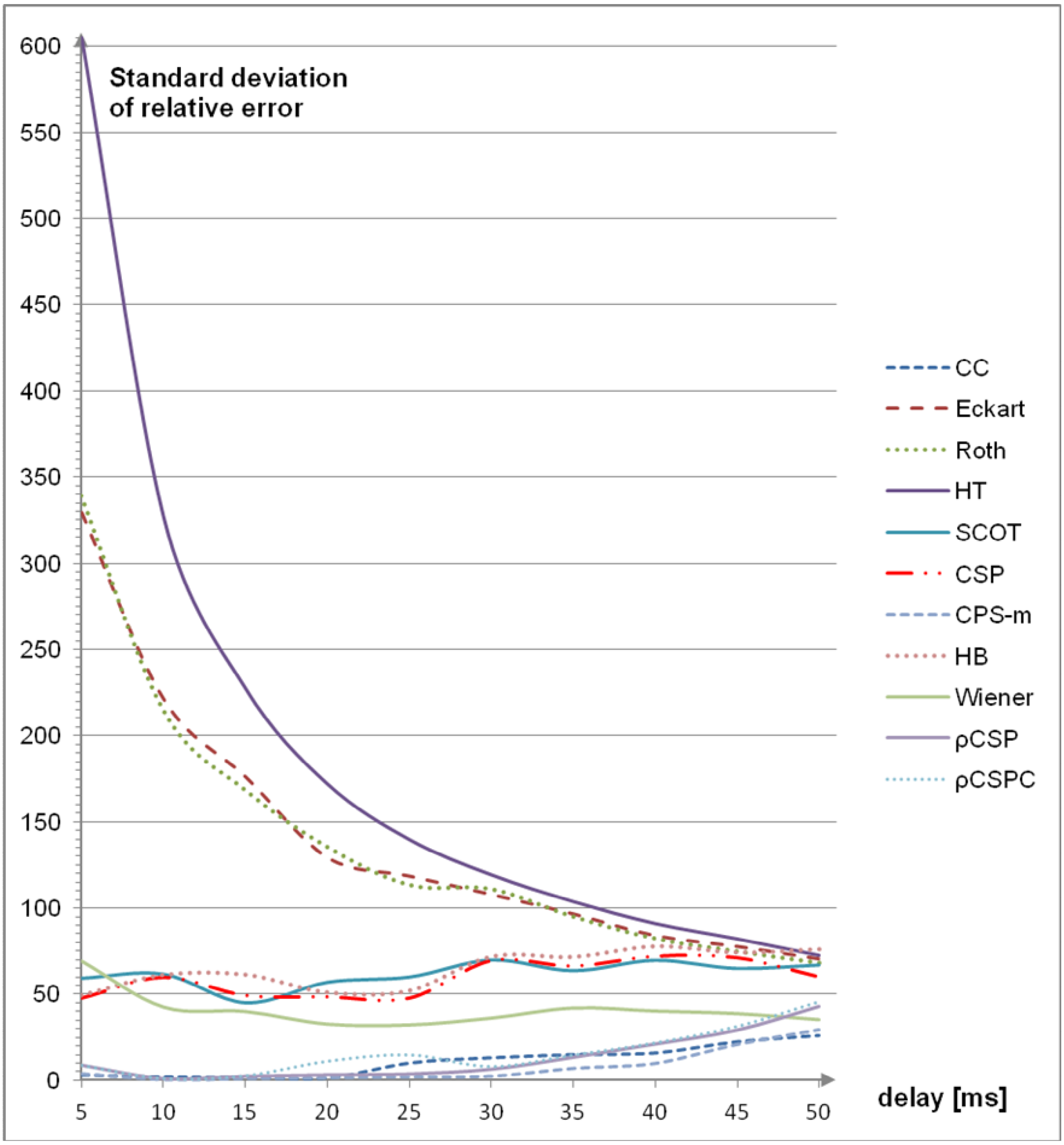


Figure 3.8 – Average standard deviation of relative error for accumulated GCC methods

Another conclusion drawn from these simulated results is represented by big difference between standard deviations for various approaches. This is also explained by the high error rate of the implied methods which leads to incorrectly estimated results with high variation.

The CC, CPS-m, ρCSP, and ρCSPC approaches offer reliable results regarding standard deviation, CC outperforming all methods for very small relative delays and CPS-m being the best for small and large delays.

Recalling previous simulated results we find that ρCSP outperforms all other methods as regards the error rate, while for the relative error and standard deviation CPS-m does. This difference is explained by the fact that when both methods estimate incorrectly, ρCSP introduces a higher error than CPS-m.

3.3.2 EVALUATING GCC METHODS

In this subsection all presented GCC methods are analyzed with regard to three implementation schemes. Recalling their descriptions from section 0 we remember that the implementation of scheme 1 (Figure 2.3) uses a large analysis window to perform TDE. On the other hand, the implementation of schemes 2 (Figure 2.4) and 3 (Figure 2.5) divides the large analysis window into several smaller frames. Then, scheme 2 proposes to compute the final TDE as an average of the smaller frames estimates, while scheme 3 proposes to accumulate the cross power spectrum of all frames in frequency domain and then compute a single TDE.

In the experiments simulated next, the same configuration as in the previous section was used. For scheme 2 and 3 the number of frames was 4, the frame size was fixed to 512 samples and the overlapping technique was not used. The length of all 4 smaller frames is equivalent to a large analysis frame of 2048 samples. This large frame was used to test the methods implemented with scheme 1.

3.3.2.1 Statistical Metrics Results for GCC methods

The results were again grouped in 2 categories, for small and large delays. From Table 3.14 to Table 3.17 the simulated results confirm that the all GCC methods, in any implementation scheme perform better for higher SNR. In these favorable conditions they yield a smaller error rate and a smaller standard deviation of relative error. Also, it is confirmed that a smaller delay has more chances to be estimated correctly than a larger one.

All GCC methods lead to the smallest error rate, in all four combinations (low/high SNR and small/large delays) when they are implemented with scheme 1. This implementation analyzes a larger frame and has more signal data to estimate the delay. As opposed to schemes 2 and 3, scheme 1 uses an analysis window of 256ms (2048 samples at a sampling frequency of 8 kHz) to estimate delays between 5 to 50 ms. Thus, these delays represent 2% to 20% of the analysis window, being categorized as relatively small delays for this scheme. On the other hand, for schemes 2 and 3 the delays between 5 to 50 ms correspond to almost 8% to 80% of the their frame size, including small and large delays. For these implementations a delay of 50 ms is categorized as a relatively large one and the probability of correct estimation decreases significantly.

Table 3.14 – GCC Methods Evaluation, with Low SNR (0dB), and Small Delays (5..30ms)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	0.53	94.52	8.42	5.13	35.69	**13.4**
Eckart	40.46	99.72	82.71	564.68	109.59	196.95
ROTH	34.99	98.97	65.05	758.9	117.82	220.88
SCOT	51.75	99.87	64.65	201.27	78.29	78.94
CSP	4.46	95.74	19.54	228.85	99.69	94.07
CSP-m	0.48	95.29	6.93	**1.29**	33.6	17.9
HT (ML)	57.82	99.81	92.81	786.93	122.57	272.72
HB	4.46	95.74	19.54	230.8	102.75	101.65
Wiener	31.8	99.01	55.68	116.64	44.2	66.52
ρCSP	**0.44**	93.27	**4.69**	6.35	34.71	22.09
ρCSPC	0.56	93.94	5.53	3.48	**33.53**	20.37

Table 3.15 – GCC Methods Evaluation, with Low SNR (0dB), and Large Delays (35..50ms)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	3.77	99.91	47.22	7.94	**21.54**	**26.47**
Eckart	50.6	100	97.64	223.31	40.5	79.11
ROTH	45.14	100	93.44	272.55	39.37	83.94
SCOT	55.6	100	83.16	113.09	36.73	70.90
CSP	8.07	99.96	62.42	106.65	44.31	78.89
CSP-m	1.81	99.94	31.55	**5.23**	22.34	29.99
HT (ML)	66.46	100	99.15	261.3	40.44	86.63
HB	8.07	99.96	62.42	111.73	44.84	83.10
Wiener	45.32	100	86.99	45.17	21.64	42.92
ρCSP	**1.25**	99.52	**27.52**	7.12	22.8	39.05
ρCSPC	1.35	99.57	30.87	10.31	22.91	41.78

Table 3.16 – GCC Methods Evaluation, with High SNR (15dB), and Small Delays (5..30ms)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	0	86.93	4.79	0	**27.41**	7.37
Eckart	13.63	99.69	66.95	355.88	89.89	161.42
ROTH	15.83	95.84	29.69	461.29	93.21	135.05
SCOT	49.6	99.75	56.68	0.51	55.79	39.92
CSP	0	88.82	5.86	0	61.69	38.25
CSP-m	0	84.98	1.72	0	29.89	5.76
HT (ML)	35.24	98.88	80.36	626.69	112.74	246.01
HB	0	88.82	5.86	0	67.21	45.55
Wiener	7.44	97.83	46.21	20.27	31.99	35.49
ρCSP	0	81.88	**0.6**	0	32.55	**3.86**
ρCSPC	0	84.13	**0.6**	0	35.11	4.37

TABLE 3.17 – GCC METHODS EVALUATION, WITH HIGH SNR (15DB), AND LARGE DELAYS (35..50MS)

GCC name	Error rate (%)			Standard Deviation of Relative Error		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
CC	0.08	100	39.62	0.51	**22.21**	17.93
Eckart	16.56	100	92.19	152.04	44.35	88.87
ROTH	34.79	100	77.77	233.78	40.71	82.75
SCOT	52.53	100	65.52	2.97	38.41	75.74
CSP	1.58	99.3	28.83	8.66	43.58	68.82
CSP-m	0	100	12.8	0	22.73	**8.33**
HT (ML)	41.83	100	96.21	213.36	42.89	86.43
HB	1.58	99.3	28.83	8.66	47.53	75.62
Wiener	28.89	100	85.85	14.04	20.58	37.10
ρCSP	0	99.4	**4.29**	0	26.99	18.18
ρCSPC	0	99.43	4.83	0	31.17	20.05

Implementation of scheme 2 leads to unusable results because the error rate is very high, more than 80%, for all GCC methods. Scheme 3 is the second best scheme with regard to error rate and for some weighting functions it has acceptable performances. It leads to faster computing speed than scheme 1 and represents a tradeoff between the computing time and the error rate. Because of these, scheme 3 could be used instead of scheme 1 for relatively very small delays, implemented by *acc-ρCSPC*, *acc-ρCSP*, acc-CC or acc-CPS-m approaches.

Regarding the standard deviation of relative error we have to notice a few aspects. First, for several GCC approaches implemented with scheme 2 and 3 the standard deviation of relative error decreases with the SNR. Normally, the standard deviation of error rate should increase with the decrease of the SNR level. However, this case is similar the above analyzed situation encountered when we discussed the results of this metric for accumulated GCC methods. The explanation stands in the fact that those implemented methods provide in these situations high error rates (over 60%). In these cases for the majority of the estimations they provide incorrect results much more frequently. Because of this the variation error decreases and this leads to a smaller standard deviation even for higher error rates.

Secondly, in spite of a smaller error rate from the GCC method implemented with scheme 1, usually the standard deviation of the relative error is higher for this scheme than for scheme 2 and 3. This is explained by the fact that scheme 1 uses a four times larger frame size and any incorrectly estimated delay varies in a larger domain. Thus, the variations of relative error are larger and this leads to higher standard deviation values.

3.3.2.2 Processing Time Evaluation for GCC Methods

Beside the statistical results the computing time is also important when we analyze the GCC method. This metric offers useful data for any real time system designer. First we compare the computed time for all methods implemented with all three schemes and then we further evaluate the accumulated approach. To perform the processing time tests the methods were implemented in C and were compiled with gcc-4.7.3. Then the tests were ran on a machine with Ubuntu 13.04 operating system, Intel Core i5 processor and 4 GB RAM.

Table 3.18 provides detailed data for the presented GCC methods, in all three implementation schemes. A large analysis frame of 2048 samples was used for scheme 1, whereas, for schemes 2 and 3, 4 smaller frames of 512 samples were used. As we expected, the results confirm that the implementation scheme 1, which uses a large analysis frame, is slower than those which divided the large analysis frame in several smaller frames, like scheme 2 and 3. Moreover, the computing time for the accumulating methods is reduced even more, thanks to the benefit offered by scheme 3 (which reduces the total FFT number, from 3*K* to 2*K*+1, as was presented in section 0).

TABLE 3.18 – COMPUTING TIME EVALUATION FOR GCC METHODS

GCC	**Computing time (µs)**		
	Scheme 1	Scheme 2	Scheme 3
CC	1.51	1.28	1.00
Eckart	3.72	3.89	3.61
ROTH	1.87	1.64	1.38
SCOT	2.41	2.20	1.92
CSP	2.28	2.08	1.82
CSP-m	5.30	5.03	4.72
HT	4.39	3.70	3.43
HB	2.87	2.67	2.39
Wiener	3.26	3.08	2.80
ρCSP	4.85	4.67	4.39
ρCSPC	6.66	6.54	6.26

For all schemes, the normal cross-correlation has the fastest processing time. This is because this CC has not any weighting operations. On the other hand side we find ρCSPC as the slowest method. Besides the ordinary FFTs, it has to spend additional time in computing its complex weighting function.

Depending on the size of the large analysis window and the number of smaller frames which divide it, the computing time may suffer significant variations. This is presented in Table 3.19, for the acc-ρCSPC method, where different window sizes were chosen for evaluation. The computing time for all possible and practical values of the frame size was measured for each window size. For example, if the window size is 1024 samples, the possible sizes for the frame are 512 samples and 1024 samples. The N/A cells indicate the situations in which the frame size is larger than the window size. It must be noted that the frame size cannot be too small because in that case the expected delay to be measured may be larger than the new smaller frame. The number of samples per frame was chosen to be a power of 2 in order to facilitate the fast computation of the FFT.

Table 3.19 – Acc-pCSP Processing Time Dependence on Window/Frame Size

Frame size	**Computing time in µs for windows of**			
	1024 samples	2048 samples	4096 samples	8192 samples
512	151	289	565	1058
1024	*163*	302	582	1081
2048	N/A	*330*	617	1124
4096	N/A	N/A	*670*	1193
8192	N/A	N/A	N/A	*1305*

The results show that by increasing the number of frames within the analysis window the processing time is reduced. For example, for a long window size of 8192 samples which is divided in 2 smaller frames of 4096 samples the processing time decreased by almost 8.6%, from 1305 µs to 1193 µs. If we divide the window in 4 frames of 2048 samples the processing time decreases even more, by almost 14%, from 1305 µs to 1124 µs. We stopped with iterations at 16 frames of 512 samples for which the processing time decreased by 19%, from 1305 µs to 1058 µs. This computing time reduction was demonstrated in section 0 and now it was confirmed by these tests.

The results also suggest that longer windows require longer processing time. On the other hand, longer windows provide better accuracy. Depending on the time and accuracy constraints that each application imposes the appropriate configuration may be chosen.

4 Voice Activity Detection

Identifying speech from non-speech segments (regions) in an audio signal is known as voice activity detection (VAD) or speech detection. This topic is still a very hot one for the telecommunications and audio processing community, because almost all speech applications need a special block for speech detection. Various VAD methods are used in applications such as voice communications, speaker recognition, text-to-speech, noise reduction for hearing aids, speech enhancement, voice coding, and compression.

The above mentioned speech applications generate various VAD algorithms. Usually, they look for speech features of the signal and then assign a VAD score. After this step a speech/non-speech decision is taken, based on the procedure of the method in question. In time one or more different features of voice signals and techniques were included in VAD algorithms, such as power envelope [Mark, 2002], energy or/and subband energy [ITU-T G729b] [Cornu, 2003] [Evangelopoulos, 2005][Soleimani, 2008] [Moattar, 2009] [Jiang, 2010], entropy [Ouzounov, 2004], linear prediction coefficients [Rabiner, 1977], SNR and channel SNR [ETSI-AMR], zero-crossing [Kotnik, 2001], voice pitch, cepstral distant measure [Haigh, 1993], correlation coefficients [Craciun, 2004] [Shuyin, 2009], wavelet transform [Wu, 2006][Lee, 2006], Walsh basic function representation [Pwint, 2005], longterm speech information [Ramirez, 2004] [Ghosh, 2011] [Ma, 2013], periodic to aperiodic component ratio [Seo, 2007] [Ishizuka, 2010]. Also, in [Chatlani, 2010] it is shown that the image processing technique called “local binary pattern” could be used for VAD algorithms. Besides these, some statistical methods train voice of noise mixtures in labeling them in supervised or semi-supervised mode like [Davis, 2006] [Farsi, 2008] [Fujimoto, 2008][Ying, 2011] [Harding, 2012] [Omar, 2012] [Germain 2013].

Generally, speech signals with higher SNR and/or better intelligibility yields higher VAD scores. But this observation suggests that finding an adequate VAD algorithm to fit demands from ATM-VCS is a key point in the best signal selection issue. Thus, VAD algorithms were implemented and tested in this work, as a solution for the BSS issue.

In time numerous VAD algorithms were developed in order to satisfy the specific demands for various applications. Thus, a unique solution, to fit all applications, could not be implemented. However, the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) and the European Telecommunications Standards Institute (ETSI) proposed globally-applicable and recommended standards for Information and Communications Technologies (ICT) to ensure interoperability between different networks and technologies (fixed, mobile, radio, converged, broadcast and Internet).

From among these standard VAD algorithms, ITU-T G729B and ETSI-AMR1&2[1] were important starting points for several other VAD algorithms. Because of this, they continue to be of high interest to the speech processing community; thus, it is only natural to start the VAD algorithm description with them. They are followed by two relatively new VAD algorithms which were further integrated into the BSS solution from ATM-VCS.

4.1 ITU-T G729.B

General description of ITU-T G729B VAD

The VAD ITU-T G729.B algorithm was introduced in 1996 in order to optimize the transmission rate during silent periods of speech. It provides a VAD decision for each frame at every 10 ms, based on several parametric features. Its flow chart is presented in Figure 4.1.

For the first N_i frames, the algorithm extracts the next four parameters: low and full-band frame energies, the set of Line Spectral Frequencies (LSF) and the zero crossing rate of the frame. Before the N_i-th frame a long-term average initialization stage is performed. Meanwhile, the frame energy obtained with LPC analysis is compared with a threshold of 15dB. If the frame energy is less than the threshold, then the voice activity detection is set to 0. Otherwise, the voice activity detection is set to 1. When the frame number reaches N_i it starts the initialization process for the features of the background noise.

[1] © European Telecommunications Standards Institute 2012. Further use, modification, copy, and/or distribution are strictly prohibited. ETSI standards are available from http://pda.etsi.org/pda”.

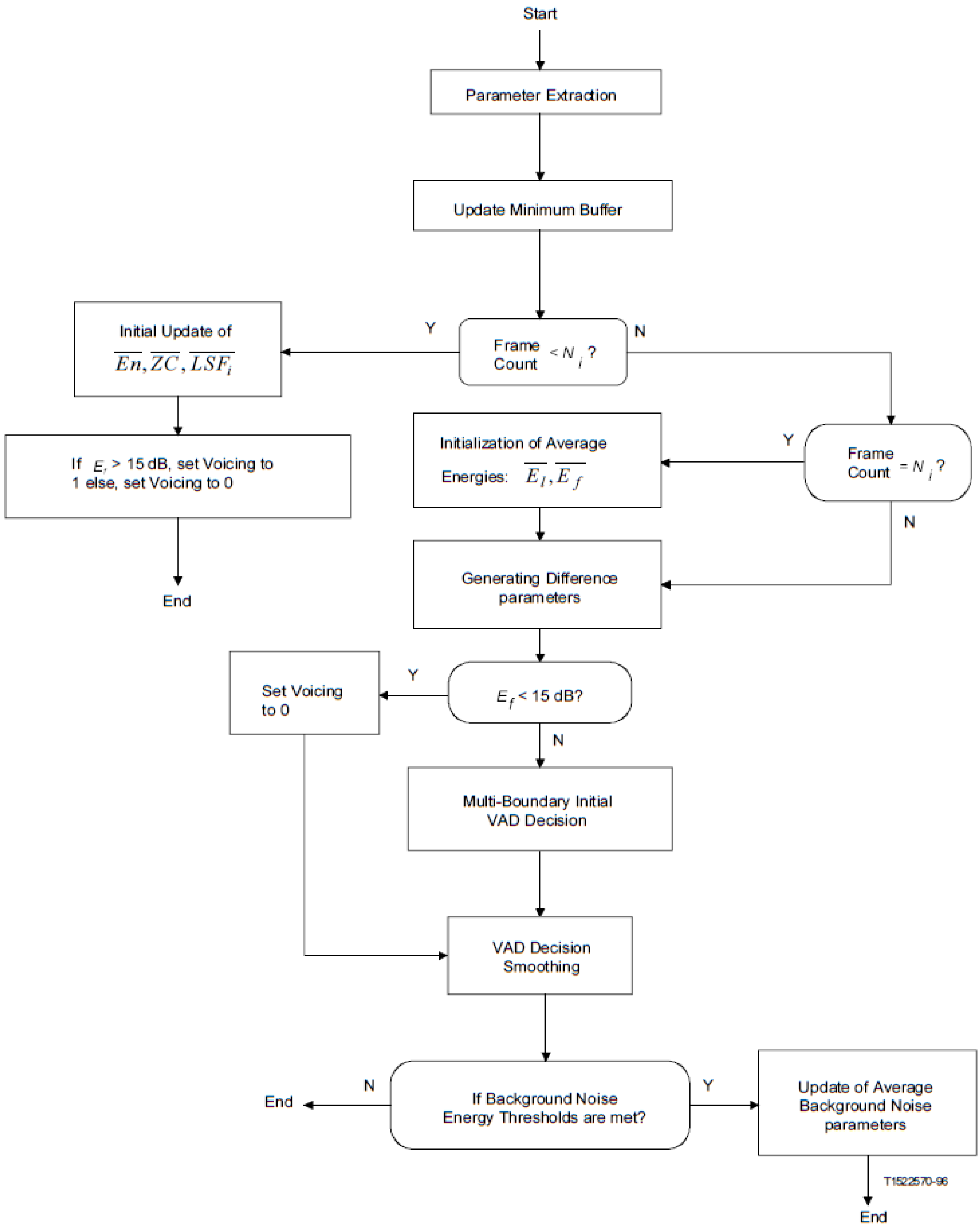

Start
Parameter Extraction
Update Minimum Buffer
Y
Frame Count $< N_i$?
N
Initial Update of $\overline{En}, \overline{ZC}, \overline{LSF_i}$
If E_f > 15 dB, set Voicing to 1 else, set Voicing to 0
End
Initialization of Average Energies: $\overline{E_l}, \overline{E_f}$
Y
Frame Count $= N_i$?
N
Generating Difference parameters
Set Voicing to 0
Y
E_f < 15 dB?
N
Multi-Boundary Initial VAD Decision
VAD Decision Smoothing
End
N
If Background Noise Energy Thresholds are met?
Y
Update of Average Background Noise parameters
T1522570-96
End

Figure 4.1 – G729B VAD flowchart[2]

[2] Figure from [ITU-T G729B]

Starting with the frame number N_{i+1} a set of difference parameters is computed. It represents the variation between current frame parameters and the average values of the background noise features. The difference parameters are:

- an energy difference;
- a spectral distortion;
- a zero-crossing rate;
- a low-band energy difference.

At this stage, the voice activity decision is based on a multi-boundary decision region space formed by the above four difference measures. Thus, the union of these regions corresponds to a voice activity and the complementary space of this union corresponds to a non-active voice. Further, a smoothing decision results based on energy and the past decisions of previous frames.

A final step in this algorithm is the update of running averages. This takes place only in the absence of speech, on a background noise and after a true adaptive threshold test.

Parameter extraction

For each 10ms frame a set of parameters is extracted from the input signal. The basic set is represented by the autocorrelation coefficients $\left\{R(i)\right\}_{i=0}^{q}$, where $q=12$. From this set the set of the linear prediction coefficients (LPC) will be derived and further the set of the *line spectral frequencies* $\left\{LSF_i\right\}_{i=1}^{p}$ will be derive, where $p=10$.

Then, the *low band energy* E_l, from 0 to the cutoff frequency F_l of an FIR filter is calculated by means of the following formula:

$$E_l = 10\lg\left[\frac{1}{N}\mathbf{h}^{\mathrm{T}}\mathbf{R}\mathbf{h}\right] \tag{4.1}$$

where $\mathbf{h}$ is the impulse response of the FIR filter and $\mathbf{R}$ is the Toeplitz autocorrelation matrix which has the autocorrelation coefficients on each diagonal.

Afterwards, the *full band energy* E_f is obtained from the first autocorrelation term $R(0)$ by:

$$E_f = 10\lg\left[\frac{1}{N}R(0)\right] \tag{4.2}$$

where $N=240$ is the length of the linear prediction coefficients (LPC) analysis speech window.

The zero-crossing rate (ZC) is computed in the normalized form for each frame as:

$$ZC = \frac{1}{2M}\sum_{i=0}^{M-1}\left|\operatorname{sgn}\left[x(i)\right]-\operatorname{sgn}\left[x(i-1)\right]\right| \tag{4.3}$$

where $x(i)$ is the preprocessed input signal and $\mathrm{M} = 80$, as defined in G.729 for codec description.

Initializing running averages

The first N_i frames, which have the energy E greater than 15 dB, are used to compute the next initializing parameters:

- the *spectral parameters of the background noise* $\{\overline{LSF_i}\}_{i=1}^{p}$ is initialized as the average of the $\{LSF_i\}_{i=1}^{p}$ of the actual frames;
- the *average of the background noise zero-crossing* $\overline{ZC}$ is initialized as the average of the ZC of the actual frames;
- the *average of the frame energy* $\overline{E}_n$ initialized as the average of frame energy E_f over the first N_i frames.

Further, the running averages of the background noise low-band energy $\overline{E}_l$ and the background noise energy $\overline{E}_f$ are computed based on the following rules:

```
if $\overline{E}_n \le T_1$ then
    $\overline{E}_l = \overline{E}_n + K_1$
    $\overline{E}_f = \overline{E}_n + K_0$
else if $T_1 < \overline{E}_n < T_2$
        $\overline{E}_l = \overline{E}_n + K_3$
        $\overline{E}_f = \overline{E}_n + K_2$
    else
        $\overline{E}_l = \overline{E}_n + K_5$
        $\overline{E}_f = \overline{E}_n + K_4$
```

where T_1, T_2, K_0, K_1, K_2, K_3, K_4, and K_5 are constant values defined in G729 VAD constant table.

Difference parameters

After the initialization stage completed, the G729B VAD algorithm computes four difference parameters for each frame. They are called difference measures because they are obtained as the difference between the current frame extraction parameters and the average running parameters of the background noise, as shown below:

- The spectral distortion ΔS

$$\Delta S = \sum_{i=1}^{p} (LSF_i - \overline{LSF_i})^2 \tag{4.4}$$

- The full-band energy difference ΔE_f

$$\Delta E_f = \overline{E}_f - E_f \tag{4.5}$$

- The low-band energy difference ΔE_l

$$\Delta E_l = \overline{E}_l - E_l \tag{4.6}$$

- The zero-crossing difference ΔZC

$$\Delta ZC = \overline{ZC} - ZC \tag{4.7}$$

Multi-boundary initial voice activity decision

An initial decision regarding the presence of voice, noted as I_{VD}, depends on the vector of difference measures. If the vector lies in the union of fourteen boundary decision regions, then it is considered voice activity and $I_{VD} = 1$, otherwise $I_{VD} = 0$ when no voice activity is detected. The union region is obtained in a four-dimensional space by fourteen boundary decisions expressed bellow:

1) $\Delta S > a_1 \cdot \Delta ZC + b_1$

2) $\Delta S > a_2 \cdot \Delta ZC + b_2$

3) $\Delta E_f < a_3 \cdot \Delta ZC + b_3$

4) $\Delta E_f < a_4 \cdot \Delta ZC + b_4$

5) $\Delta E_f < b_5$

6) $\Delta E_f < a_6 \cdot \Delta S + b_6$

7) $\Delta S > b_7$

8) $\Delta E_l < a_8 \cdot \Delta ZC + b_8$

9) $\Delta E_l < a_9 \cdot \Delta ZC + b_9$

10) $\Delta E_l < b_{10}$

11) $\Delta E_l < a_{11} \cdot \Delta S + b_{11}$

12) $\Delta E_l > a_{12} \cdot \Delta E_f + b_{12}$

13) $\Delta E_l < a_{13} \cdot \Delta E_f + b_{13}$

14) $\Delta E_l < a_{14} \cdot \Delta E_f + b_{14}$

where a_i and b_i are constant values defined in the G.729 standard. If at least one condition is “FALSE”, then $I_{VD} = 0$.

Smoothing the voice activity decision

The final voice activity decision for each frame is obtained after a smoothing procedure of four steps:

a) firstly, several conditions are evaluated to determine if the smoothing procedure is applicable for the current frame. Besides the I_{VD} verification, previous smoothed voice activity decision and energy parameters are also tested;

b) in the second stage the result of the first step is tested combined with the full band energy of the current frame, energy of the previous frame, smoothed voice activity of the previous 2 frames, a Boolean parameter, a smoothing counter, and a G729 constant value;

c) the third smoothing stage has four tested conditions for the current smoothed voice activity decision, full-band energy of the current frame, energy of the previous frame, a noisy continuity counter and a G729 constant value;

d) the final smoothing stage decides no speech if three combined conditions are “TRUE”; the combined conditions tests full-band energy of the current frame, the average full-band energy, frame count, two G729 VAD constant values and the smoothing activity for the current frame.

Updating the running averages

The final stage of the VAD module is represented by the running averages update. The update sequence is performed only if the next condition is “TRUE”:

$$E_f < \overline{E}_f + T_6$$

The update of the running averages of the background noise uses the first order autoregressive model, as expressed below in the general form for all 4 updates:

$$\overline{A}_i = \beta_i \overline{A}_i + \left(1 - \beta_i\right) A_i \tag{4.8}$$

where $\overline{A}_i$ represents the running averages of the full-band energy ($\overline{E}_f$), low-band energy ($\overline{E}_l$), zero-crossing ($\overline{ZC}$) and line spectral frequencies ($\overline{LSF}$), and β_i stands for the update coefficient set which depends on the current frame number.

4.2 ETSI-AMR 1

The first option for a VAD algorithm for the Adaptive Multi-Rate audio codec proposed by ETSI is known as the ETSI-AMR 1 VAD. Compared to G.729, this algorithm uses 20 ms frames to indicate any voice activity. More importantly, this algorithm does not look only for speech features in the signal, but it also searches for music or information tone details. Thus, further in this VAD algorithm, *speech detection* will refer not only to speech, but also to music and information tones.

Figure 4.2 shows that the ETSI-AMR1 VAD contains five important blocks. Based on the output of the following four blocks – filter bank and sub-band levels, pitch detection, tone detection and complex signal analysis – the VAD decision block yields vad_flag = 1 for *speech detection*, otherwise vad_flag = 0.

Filter block and sub-band levels calculation

This block is used to compute the sub-band levels of the input signal. First, a filter block divides the input signal into nine sub-band frequencies, as shown in Table 4.1. This is accomplished by the 3rd and 5th order filter blocks which split the input into low-pass and high-pass components, followed by sampling frequency decimation by 2.

Then, for each sub-band output the signal level is computed by means of the following rule:

$$l\left(n\right) = \sum_{i=FIRST_n}^{LAST_n} \left|x_n\left(i\right)\right| \tag{4.9}$$

where n is the frequency band, $x_n\left(i\right)$ is the i-th sample of sub-band n,

$$FIRST_n = \begin{cases} -2, & n \le 4 \\ -4, & 5 \le n \le 8 \\ -8, & n = 9 \end{cases}$$

these negative values meaning previous frame, and

$$LAST_n = \begin{cases} 9, & n \le 4 \\ 19, & 5 \le n \le 8 \\ 39, & n = 9 \end{cases}$$

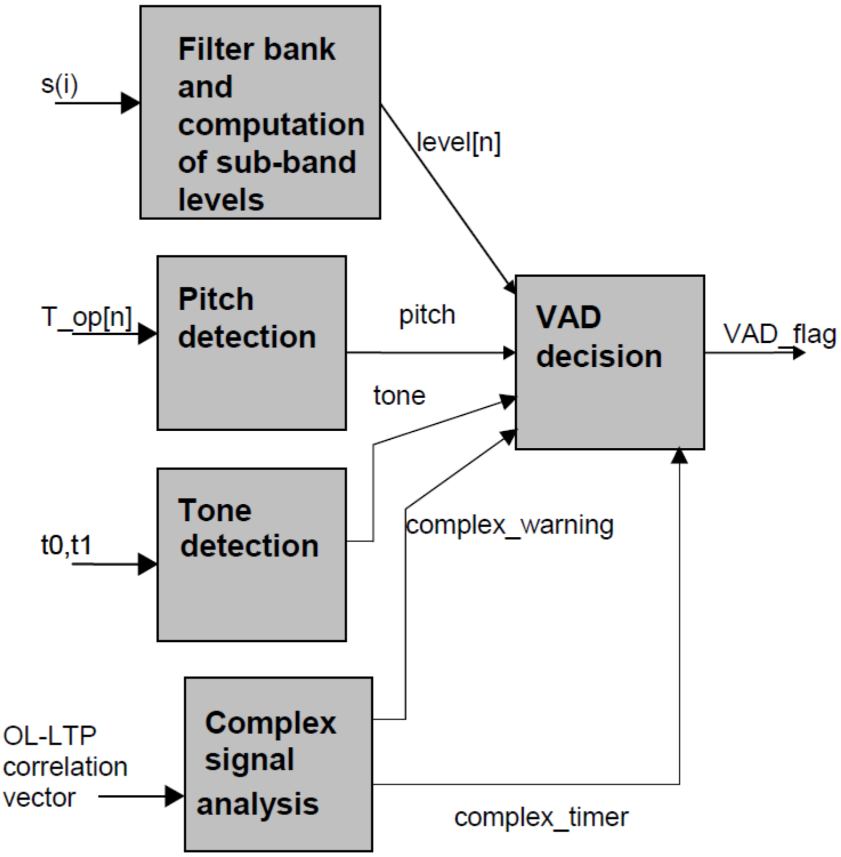


Figure 4.2 – Block Diagram for VAD ETSI-AMR1[3]

[3] Figure from [ETSI-AMR]

TABLE 4.1 – ETSI-AMR1 SUB-BAND DISTRIBUTION

Band number	Frequencies (Hz)
1	0 - 250
2	250 - 500
3	500 - 750
4	750 - 100
5	1000 - 1500
6	1500 - 2000
7	2000 - 2500
8	2500 - 3000
9	3000 - 4000

Pitch detection

This block is intended to detect any vowel sounds or other periodic signals. Based on open-loop pitch lags, represented by the values which maximize the autocorrelation function of the speech signal, the pitch detection procedure is described as follows. First the difference of two consecutive open-loop lags is compared. If this difference is smaller than a threshold, a lagcounter is incremented. Then, the pitch is detected if the sum of lagcounters from two consecutive frames is greater than a threshold.

Tone detection

In some cases it is possible that information tones are spotted by pitch detection because of their periodicity. But for these tones, a better detection is provided by the tone detection block. The decision is taken by comparing the autocorrelation maxima, computed by the open-loop pitch analysis, to the signal power related to the autocorrelation maxima multiplied by a tone threshold.

Complex Signal Analysis

The role of the correlated signal analysis block is to spot the correlated signal from the higher bands which were not detected by the pitch and tone detector blocks. The decision is taken based on statistics of the maximum correlation value computed from the output of the high-pass filter.

VAD Decision

As anticipated, this final block has to decide if the current frame contains speech or not. For each frame, it computes the estimated background noise and levels of each sub-band, the average noise level, the power of the input signal and an adaptive threshold. If the power of the input frame is higher than the adaptive threshold, then an intermediate speech activity decision is taken. Before a final decision for the current frame, the intermediate decision follows a smoothing procedure.

4.3 ETSI-AMR 2

The second option for VAD algorithm in AMR encoders returns a decision of voice activity for each 20 ms frame. Similar to ETSI-AMR 1, this algorithm verifies if the input signal contains voice, music or information tones, but uses a different approach. Its block diagram is shown in Figure 4.3.

Frequency Domain Conversion

First, a high-pass filter prepares the input signal for the frequency domain conversion. Then the filter output $s_{hp}(n)$ is pre-emphasised for the speech reconstruction in order to reduce the negative effects of noise, by means of the following formula:

$$d(n) = s_{hp}(n) + a_1 s_{hp}(n-1) \tag{4.10}$$

where a_1 is known as the pre-emphasis factor which is typically chosen $0.96 \le a_1 \le 0.99$. After the pre-emphasis a rectangular window is applied to the signal which is then padded with zeros to prepare the sequence for the FFT.

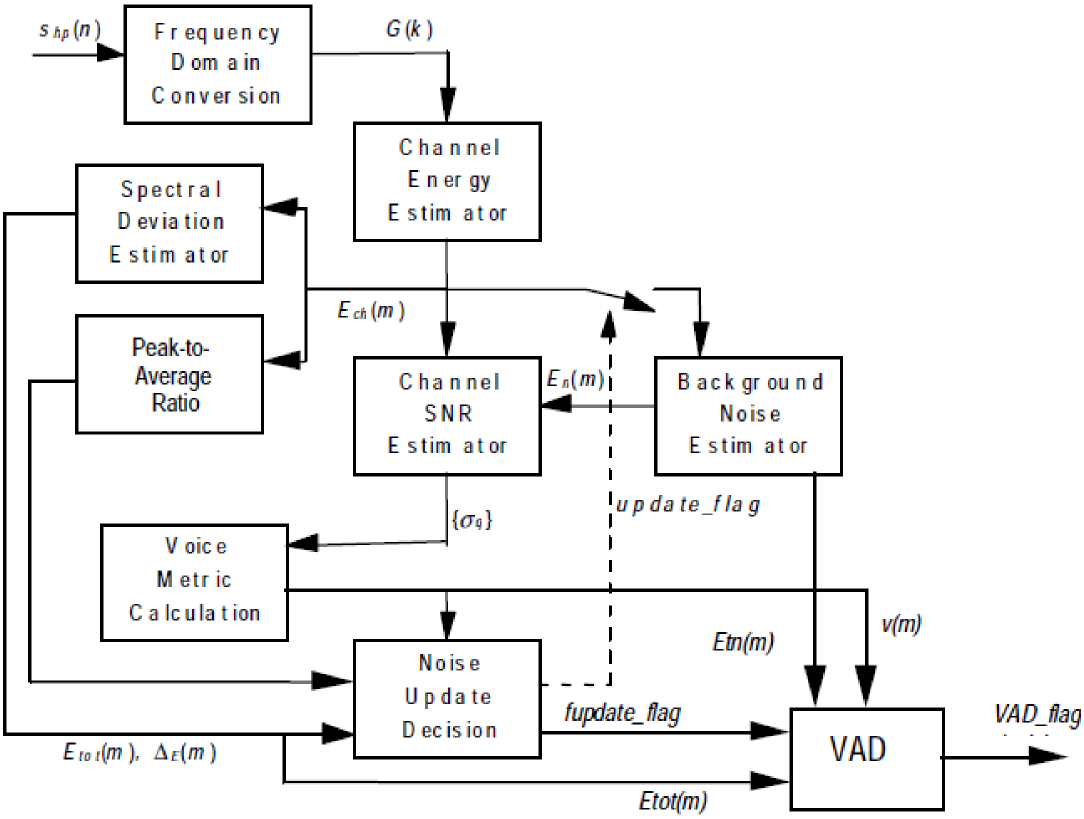


Figure 4.3 – Block Diagram for VAD ETSI-AMR2[4]

[4] Figure from [ETSI-AMR]

Channel Energy Estimator

The energy of the current frame is estimated by this block using several parameters such as the minimum allowable energy, channel smoothing factor, combination table for low and high channel, and the signal's spectrum resulting from the previous block. The estimated channel energy is used in further blocks which compute and estimate spectral deviation, peak-to-average ratio, channel SNR, and background noise.

Channel SNR Estimator and Voice Metric Calculation

These blocks estimate the channel SNR for the voice metric calculation, which is further needed in the background noise update and voice activity decisions. The computing flow follows these steps:

a) calculate an estimate for channel SNR based on the current channel noise energy and channel energy;

b) transform the previous result in 3/8 dB steps, from 0 to 89;

c) compute the voice metric as the sum of voice metric table values, indicated by the previous results.

Spectral Deviation Estimator

Based on the estimated channel energy the spectral deviation estimator is derived. Its role is to prevent false updates of the background noise when spectral deviation is very high. For this, the following actions are performed:

a) compute the log power spectrum from estimated channel energy;

b) estimate the spectral deviation based on the current log power spectrum and the average long-term log power spectrum;

c) compute the new exponential window factor based on instantaneous SNR and long-term peak SNR;

d) update the average long-term log power spectrum based on the current log power spectrum and the previously obtained exponential window factor.

Peak-to-Average Ratio

For the current frame it computes the log ratio between the maximum value of the estimated channel energy and its average. If the maximum estimated channel energy value is 10 times greater than the average of the estimated channel energy, then the background noise energy update could be inhibited by the noise update decision block.

Background Noise Update

Because of the non-stationary nature of speech, the algorithm has to update the background noise. In normal cases, the decision to start the update of the background noise is conditioned by the comparison between the previously calculated voice metric and an update threshold. If the condition is met, then the channel noise is estimated based on the minimum allowable energy, current channel energy and channel noise smoothing factor.

If the above condition is not met, a new set of conditions is evaluated. The new verified parameters are total channel energy, estimated spectral deviation, long-term prediction flag and sine wave flag. The last two flags are obtained from the open-loop pitch predictor of the speech encoder and from the peak-to-average ratio.

VAD Decision

This final block of the ETSI-AMR2 VAD algorithm block decides on the voice activity based on each 20 ms frame. However, it is important to mention that a VAD decision needs to analyze two 10 ms sub-frames.

The decision algorithm also includes a smoothed VAD decision procedure. This procedure involves different test conditions which are based on the next computed or defined parameters: quantized SNR, hangover count, burst count threshold and voice metric threshold.

4.4 Wavelet Based VAD

General Description

This subsection presents a VAD algorithm, proposed in [Wu, 2006], which includes the Teager energy operator (TEO), auto-correlation function and wavelet transform. Its final feature was called speech activity envelope (SAE) and it is used for the voice activity decision. As I noticed that the SAE could be used to extract information regarding the quality and intelligibility of speech signals this algorithm was implemented, tested and used as a solution to the BSS problem.

Compare to the previous standard VAD algorithms, which were presented earlier, this approach needs larger frames (32 to 128 ms) for the analysis. The input signal from each frame is decomposed by the Discrete Wavelet Transform (DWT) into four sub-bands in order to exploit more precisely the periodicity property of speech. Then, for each sub-band the TEO is applied to reduce the noise effects over the wavelet coefficients. Afterwards the auto-correlation function for each TEO sub-band is calculated, resulting in the Sub-band Signal

Autocorrelation Function (SSACF). To extract the periodic intensity of the signal, a Mean-Delta (MD) method is applied for each SSACF envelope, resulting in the MDSSACF. The first stage of this method uses Delta SSACF metric to characterize the local variation of each SSACF. Then, the average DSSACF is computed in order to extract the amount of periodicity from each subband. Consequently the SAE is obtained as the sum of all four MDSSACF. Depending on the application, the final voice activity decision compares the SAE with an adaptive or non-adaptive threshold. The block diagram for the wavelet based VAD is shown below, in Figure 4.4.

Wavelet Decomposition

The wavelet decomposition is performed by the wavelet transform (WT) which is based on time-frequency analysis. An important advantage of the wavelet analysis is the multi-resolution analysis (MRA) capability. This way larger time intervals can be used for a precise characterization of lower frequencies and shorter intervals for detailed information about higher frequencies. To implement the DWT efficiently, the filter bank method proposed in [Mallat, 1989] is used.

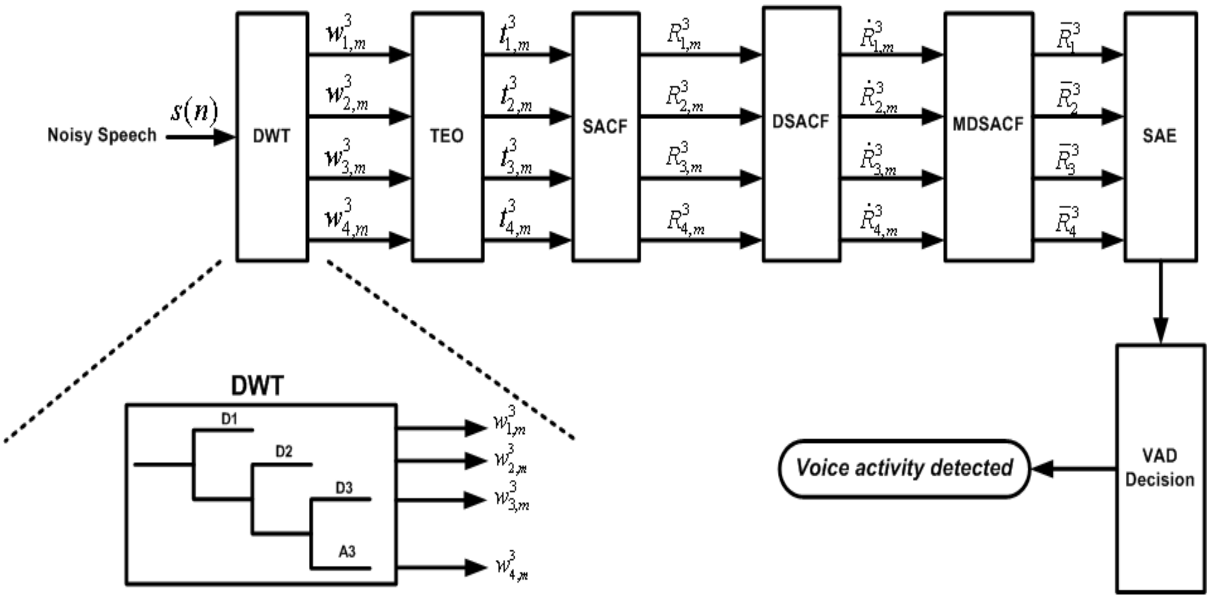


Figure 4.4 – Block diagram for the wavelet based VAD[5]

[5] Figure from [Wu, 2006]

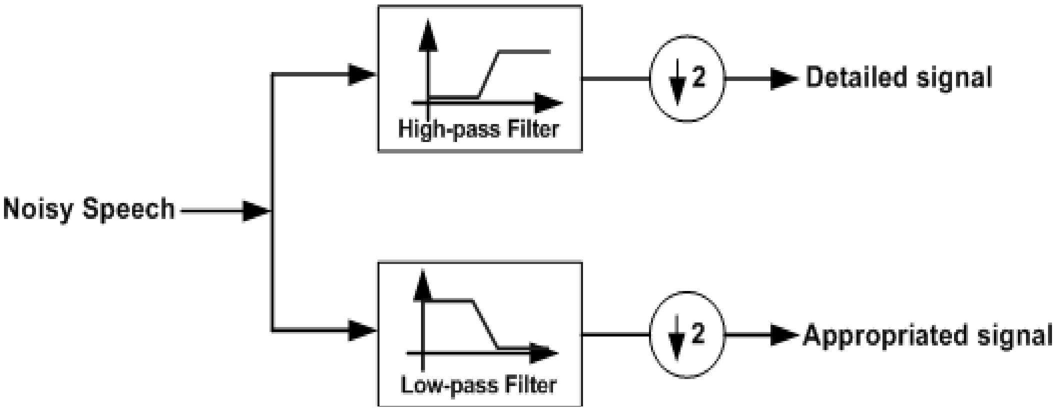


Figure 4.5 – DWT with filter bank[6]

The result of DWT decomposition, obtained by using quadrature mirror filters (QMF), is made of by two sub-band signals *A* and *D*, the approximation and detailed parts. The approximated part results after the low-pass filter and the detailed part after the high-pass filter. This process is illustrated in Figure 4.5, where ↓2 denotes the downsampling operator by 2.

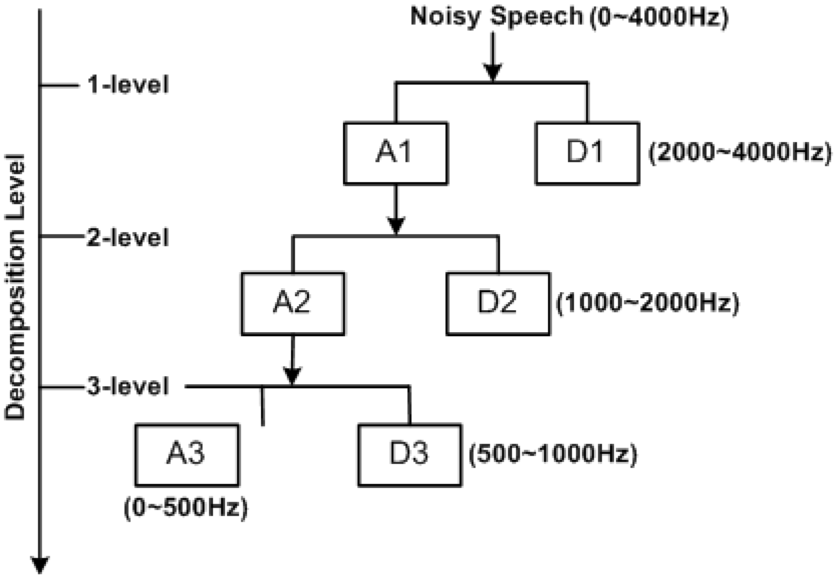


Figure 4.6 – The proposed three level wavelet decomposition structure[7]

This algorithm proposed implementing the QMF with the Daubechies family wavelet. Moreover, because the periodicity of the voice is found mainly in the low frequency bands, the algorithm grants higher resolution for the lower bands from a three level decomposition. This is also shown in Figure 4.6 where for decomposition order j we obtain $j+1$ subband coefficients. As the algorithm proposed third order decomposition, we can express the result as

[6] Figure from [Wu, 2006]
[7] Figure from [Wu, 2006]

$$w_{k,m}^{3} = DWT\{s(n), 3\}, \qquad n = \overline{1, N}, \qquad k = \overline{1,4} \tag{4.11}$$

where $s(n)$ is the input signal, N is the frame size, k is the sub-band number and $w_{k,m}^{3}$ is the m^{th} coefficient of k^{th} sub-band. Also, because of the downsampling operation the length of each subband will be $N/2^{k}$.

Teager Energy Operator

It was shown that TEO can increase the differentiation between speech and noise; moreover, it reduces the noise components which affect the noisy speech signals [Jabloun, 1999]. When compared to the popular noise reduction methods which operate in frequency domain, the TEO performs in time domain conducting to a faster computing speed.

For continuous-time TEO is described by the following formula

$$\psi_{c}\left[s(t)\right] = \left[\dot{s}(t)\right]^{2} - s(t)\ddot{s}(t) \tag{4.12}$$

where $s(t)$ is the continuous-time signal,

$$\dot{s}(t) = \frac{\delta s}{\delta t} \tag{4.13}$$

as the first derivative of $s(t)$ and

$$\ddot{s}(t) = \frac{\delta^{2} s}{\delta^{2} t} = \frac{\delta \dot{s}}{\delta t} \tag{4.14}$$

as the second order derivative of the input signal.

For the discrete-time implementation, the TEO is computed by

$$\psi_{d}\left[s(n)\right] = s(n)^{2} - s(n+1)s(n-1) \tag{4.15}$$

where $s(n)$ denotes the discrete-time input signal. Thus, regarding Figure 4.4 we can express this as follows:

$$t_{k,m}^{3} = \psi_{d}[w_{k,m}^{3}], \qquad k = \overline{1,4} \tag{4.16}$$

Sub-band Signal Auto-Correlation Function

The self-periodic intensity of a signal can be characterized well by the Auto-Correlation Function (ACF), defined as

$$R(k) = \sum_{n=0}^{p-k} s(n)s(n+k), \qquad k = \overline{0,p} \tag{4.17}$$

where k denotes the sample shift and p represents the length of ACF, or it can be written as

$$R_{k,m}^{3} = R\left[t_{k,m}^{3}\right], \tag{4.18}$$

where $R[\cdot]$ represents the auto-correlation operator.

In order to obtain a better characterization of the periodicity of the input signal, the ACF function is applied to all sub-band decomposition resulting from the TEO processing. This new feature is called the subband signal auto-correlation function.

Mean Delta SSACF

The Mean Delta method applied to each SSACF was proposed in [Ouzounov, 2004]. Its role here is to describe the periodic intensity of each SSACF. Thus, the Delta SSACF function is computed firstly by means of the following formula:

$$\dot{R}_M(k) = \frac{\sum_{m=-M}^{M} mR(k+m)}{\sum_{m=-M}^{M} m^2} \tag{4.19}$$

where $\dot{R}_M$ is the Delta SSACF over M sample neighborhood. The compact form found in Figure 4.4 is

$$\dot{R}_{k,m}^{3} = \Delta\left[R_{k,m}^{3}\right] \tag{4.20}$$

where Δ is the Delta operator.

The next step is to obtain an average of the absolute values of DSSACF over the M sample neighborhood, thus resulting in MDSSACF, expressed as follows:

$$\overline{R}_M = \frac{1}{N_b} \sum_{k=0}^{N_b-1} \left|\dot{R}_M(k)\right| \tag{4.21}$$

where N_b stands for the number of samples of the sub-band signal. Again, the form presented in Figure 4.4 is written as

$$\overline{R}_{k,m}^{3} = E\left[\dot{R}_{k,m}^{3}\right] \tag{4.22}$$

where E stands for the mean operator.

Speech Activity Envelope

The final Speech Activity Envelope is computed as the sum of all MDSSACF which describe the periodic intensity from each sub-band, as is shown below

$$SAE = \sum_{k=1}^{4} \overline{R}_k \tag{4.23}$$

This finishes the feature extraction process, which is now followed by the speech activity decision.

VAD decision and Adaptive Threshold

The decision part for this algorithm is based on an adaptive threshold technique, which was described in [Gerven, 1997]. In fact, this method computes two adaptive thresholds, one for the speech and one for the noise. They use statistical data from SAE feature from the noisy frames. In order to initialize the noise mean and variance, it is assumed that the first five frames do not include any speech activity. Further the speech and noise thresholds denoted by T_s and T_n are computed with the following formulae:

$$T_s = \mu_n + \alpha_s \sigma_n \tag{4.24}$$

$$T_n = \mu_n + \beta_n \sigma_n \tag{4.25}$$

where μ_n and σ_n are computed as the mean and variance of the SAE feature. The α_s and β_n parameters control the difference between the speech and noise thresholds. In this algorithm they are used as $\alpha_s = 5$ and $\beta_n = -1$.

Next, the VAD decision procedure is described by the following pseudo-code sequence:

```
$if\left(SAE(t) > T_s\right)$
    $then\, VAD(t) = 1$
$elseif\left(SAE(t) < T_n\right)$
    $then\, VAD(t) = 0$
$else\, VAD(t) = VAD(t-1)$
```

Figure 4.7 shows the *SAE* for a speech signal, with the two noise and speech thresholds. Based on the voice activity decision, the algorithm selects voice frames as those consecutive frames which have *SAE* greater then T_s, and non-speech frame those consecutive frames which have *SAE* lower than T_n. The switch decision of the VAD is suggested in Figure 4.7 by the *start-point* and *end-point.*

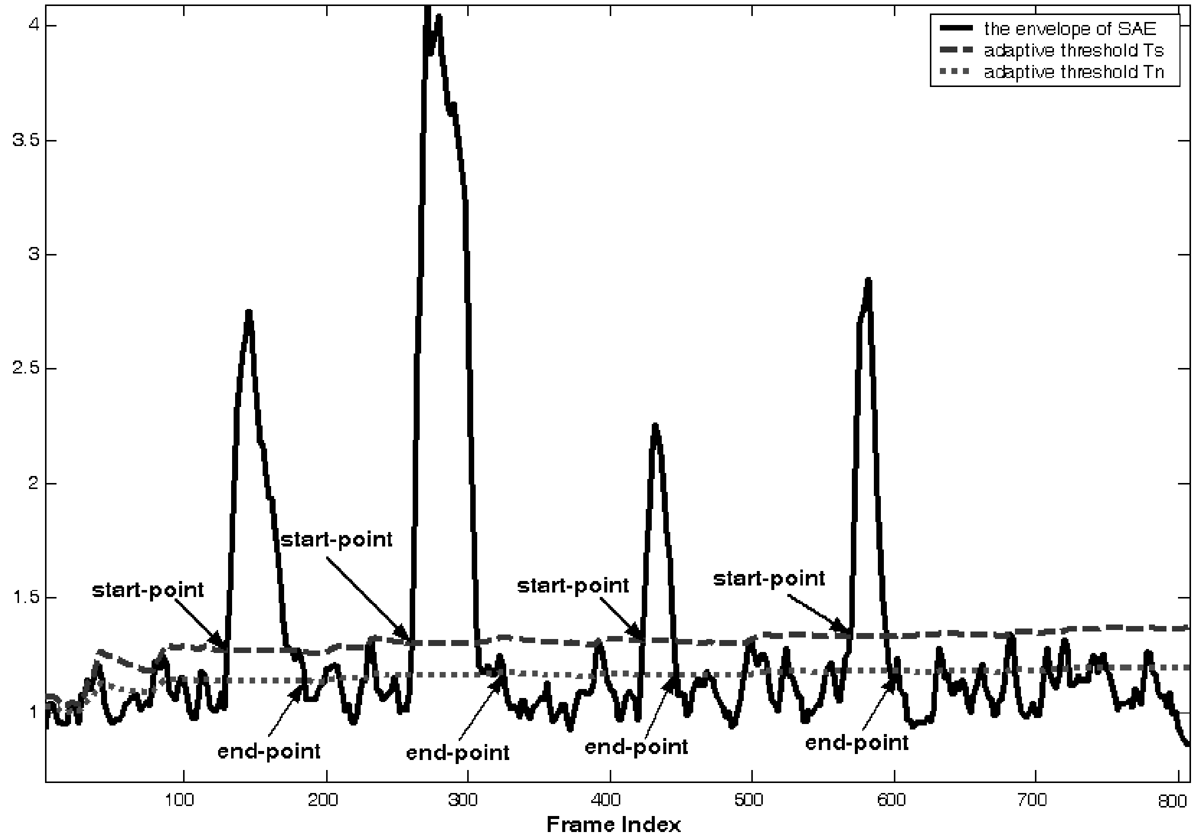


Figure 4.7 – Adaptive thresholding technique for voice activity decision[8]

In order to adapt the thresholds, the mean and the variance of the SAE are updated only in the noise periods as shown below:

$$\mu_n(t) = \gamma \mu_n(t-1) + (1-\gamma) SAE(t) \tag{4.26}$$

$$\sigma_n(t) = \sqrt{\left[SAE_{buffer}{}^2\right]_{mean}(t) - \left[\mu_n(t)\right]^2} \tag{4.27}$$

$$\left[SAE_{buffer}{}^2\right]_{mean}(t) = \gamma\left[SAE_{buffer}{}^2\right]_{mean}(t-1) + (1-\gamma) SAE(t)^2 \tag{4.28}$$

where $\left[SAE_{buffer}{}^2\right]_{mean}(t-1)$ is the mean of the previous *SAE* values from the noise only frames and $\gamma = 0.95$ is the update factor.

[8] Figure from [Wu, 2006]

4.5 VAD BASED ON LONG-TERM SPECTRAL FLATNESS MEASURE

General description

The VAD algorithm which was recently presented in [Ma, 2013], proposes using a new feature for speech detection, called long-term spectral flatness measure (LSFM). The input signal is divided in 20 ms frames with overlap of 10 ms. Each frame is then multiplied by a Hann window before estimating its spectrum by means of Welch-Bartlett method. Further, based on the spectrum from previous frames the LSFM is computed, which finishes the feature extraction procedure of the current frame. The next steps compose the voice activity decision which includes an adaptive threshold and a voting scenario proposed by [Ghosh, 2011].

Long-Term Spectral Flatness Measure

The Spectral flatness measure is used to characterize the uniformity of a sequence, expressed as the logarithmic ratio between the geometric and arithmetic means of the sequence. Thus, for the power spectrum, a low value of spectral flatness suggests a less uniform power frequency distribution, while a high value implies a more uniform power distribution. Further it was noticed that a voice sequence is described by a lower spectral flatness measure while the noisy one is described by high spectral flatness.

In speech signals the background noise is assumed to be stationary for a long period, while the speech is known as highly non-stationary. Based on these remarks, the LFSM VAD algorithm proposes using the spectral flatness measure over larger signal intervals. This way the long-term evolution of the spectral flatness measure is analyzed, to improve decision regarding speech and non-speech separation. In this approach the LSFM for the m^{th} frame and over all chosen frequencies $L_x(m)$ is calculated using the spectra of the last R frames of the input signal $x(n)$, as expressed below:

$$L_x(m) = \sum_k \log_{10} \frac{\mathrm{GM}(m,\omega_k)}{\mathrm{AM}(m,\omega_k)} \tag{4.29}$$

where $\mathrm{GM}(m,\omega_k)$ and $\mathrm{AM}(m,\omega_k)$ represent the geometric and arithmetic mean of the power spectrum, computed as:

$$\mathrm{GM}(m,\omega_k) = \sqrt[R]{\prod_{n=m-R+1}^{m} \mathrm{S}(n,\omega_k)}, \tag{4.30}$$

$$\mathrm{AM}\left(m,\omega_k\right)=\frac{1}{R}\sum_{n=m-R+1}^{m}\mathrm{S}\left(n,\omega_k\right) \tag{4.31}$$

In the above equations, $\mathrm{S}\left(n,\omega_k\right)$ is the short-time spectrum estimated here with the Welch-Bartlett method, as the average of previous *M* spectra. This method leads to a good trade-off between the spectral resolution reduction and variance reduction, as shown in [Davis, 2006] and is expressed by:

$$\mathrm{S}\left(n,\omega_k\right)=\frac{1}{M}\sum_{p=n-M+1}^{n}\left|X\left(p,\omega_k\right)\right|^2 \tag{4.32}$$

$$X\left(p,\omega_k\right)=\sum_{l=(p-1)N_{sh}+1}^{N_w+(p-1)N_{sh}}w\left(l-\left(p-1\right)N_{sh}-1\right)x\left(l\right)e^{-j\omega_k l} \tag{4.33}$$

where $X\left(p,\omega_k\right)$ represents the short-time Fourier transform (STFT) coefficient at frequency ω_k of the p th frame, $w\left(i\right)$ is the Hann window of N_w samples and N_{sh} represents the sample shift number.

The experiments of the VAD algorithm from [Ma, 2013] show that $R=30$ and $M=10$ are the optimum values for these parameters. Also, ω_k was chosen to be uniformly distributed between 500 Hz and 4 kHz. This was due to the fact that this frequency band is essential for speech intelligibility [Bies, 2003].

VAD decision

The initial decision, denoted by V_{INL} and regarding the voice activity is conditioned by the comparison between the LSFM of the last *R* frames and an adaptive threshold. If in the previous *R* frames, which end at the m^{th} frame, a speech frame is found, then $V_{INL}\left(m\right)=1$; otherwise $V_{INL}\left(m\right)=0$.

From this point further on, the final VAD decision is based on the next voting scheme. For the m^{th} frame, all previous *R* frames are analyzed by their initial decision and counted frames which were indicated as containing speech are counted. If 80% or more frames indicate speech activity, then the current frame is marked as speech activity; otherwise it is marked as non-speech.

The 80% was found experimentally in [Ma, 2013] yielding the maximum VAD accuracy in most noise tests.

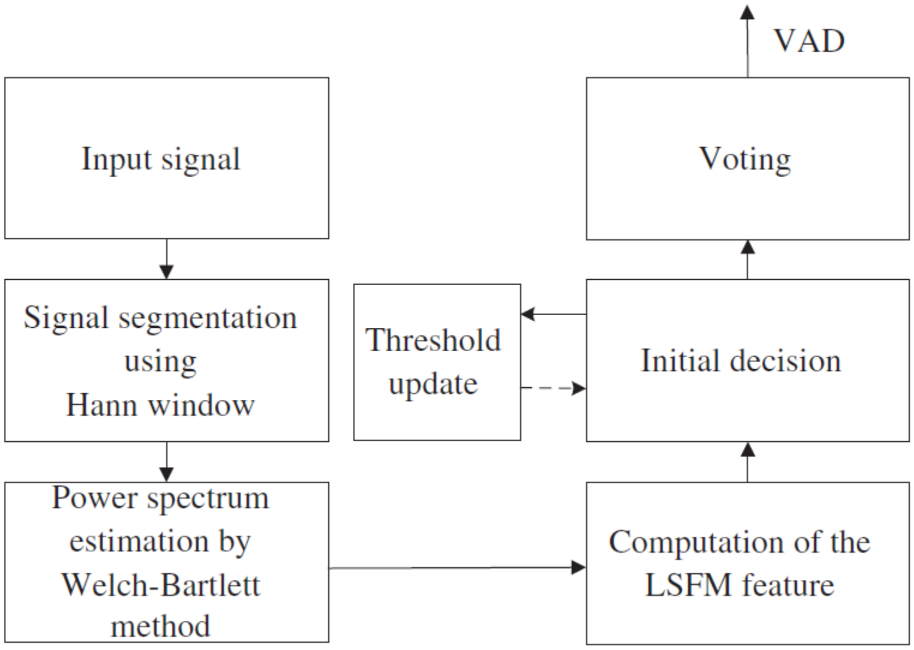

VAD
Input signal
Voting
Signal segmentation using Hann window
Threshold update
Initial decision
Power spectrum estimation by Welch-Bartlett method
Computation of the LSFM feature

Figure 4.8 – Block Diagram for LSFM-VAD[9]

Adaptive Threshold

An accurate VAD algorithm must adapt its decision by the varying acoustic environment. The algorithm which is described here also uses an adaptive threshold to increase the accuracy of VAD. Considering that the first N frames do not contain speech signals, then we can fill a noise-only LSFM buffer ψ_{INL} with the LSFM values of the frames.

Now, the initial threshold will be set to:

$$\mathrm{THR}_{\mathrm{INL}} = \min\left(\psi_{\mathrm{INL}}\right) \tag{4.34}$$

After initialization, we have to update the threshold for each new frame. For this two buffers are used: $\psi_{\mathrm{S+N}}$ and ψ_{N}. The first one will hold the last N LSFM values of the frames which were marked as containing speech, while the second will store the last N LSFM values of the frames which were marked as noise-only. Then, the adaptive threshold for the current frame is computed as:

$$\mathrm{THR}\left(m\right) = \lambda \cdot \max\left(\psi_{\mathrm{S+N}}\right) + \left(1-\lambda\right) \cdot \min\left(\psi_{\mathrm{N}}\right) \tag{4.35}$$

where λ is the combination parameter and was experimentally found to be optimum for performed experiments as $\lambda = 0.55$. Also, N = 100 in the description of the VAD algorithm form [Ma, 2013].

[9] Figure from [Ma, 2013]

5 BEST SIGNAL SELECTION

In this section, the proposed solution for the BSS problem and its specific issues is presented and commented. Briefly recalling the BSS problem description in the introduction chapter, the goal of this selection is to choose from among several noise signals received by different channels the signal with the best intelligibility and speech quality. Moreover, this decision has to take into account various problematic aspects such as:

- the speech sequence lasts in average, for less than a few seconds;
- the selection has to be done relatively quick, in less than 300ms;
- the analysis is done on relatively small parts of the signal, usually for less than 300ms;
- in the first part of 300ms of the received signal, the noise level undergoes significant variations due to the automatic gain control activation;
- for each reception the signals could be received with different delays;
- on the same channel, two consecutively received signals may be significantly different in terms of speech intelligibility.

Typical waveforms of multi-channel reception of the aircraft transmission in the ATM-VCS are shown in Figure 5.1. In this example, the speech lasts for less than 2 seconds. For channel 1, the voice signal begins around 0.5 s and ends at approximately 2.2 s. The varying signals around 0.3 and 2.45 seconds are caused by pushing and releasing the push-to-talk (PTT) button in order to start and end the communication. For the other 2 channels these limits vary, because the signals are received with different delays.

The first speech utterance is colored with red and lasts for less than 200ms. The green area represents the first part of the received signals, where there is noise and no speech utterance. An accurate BSS could not use only the green areas because they do not contain voice and thus it would be difficult to predict the speech quality. However, they could be used to compute the channel's background noise, but this is also a tricky task. Thus, the BSS has to decide about one channel based on the analysis of these parts of the signals.

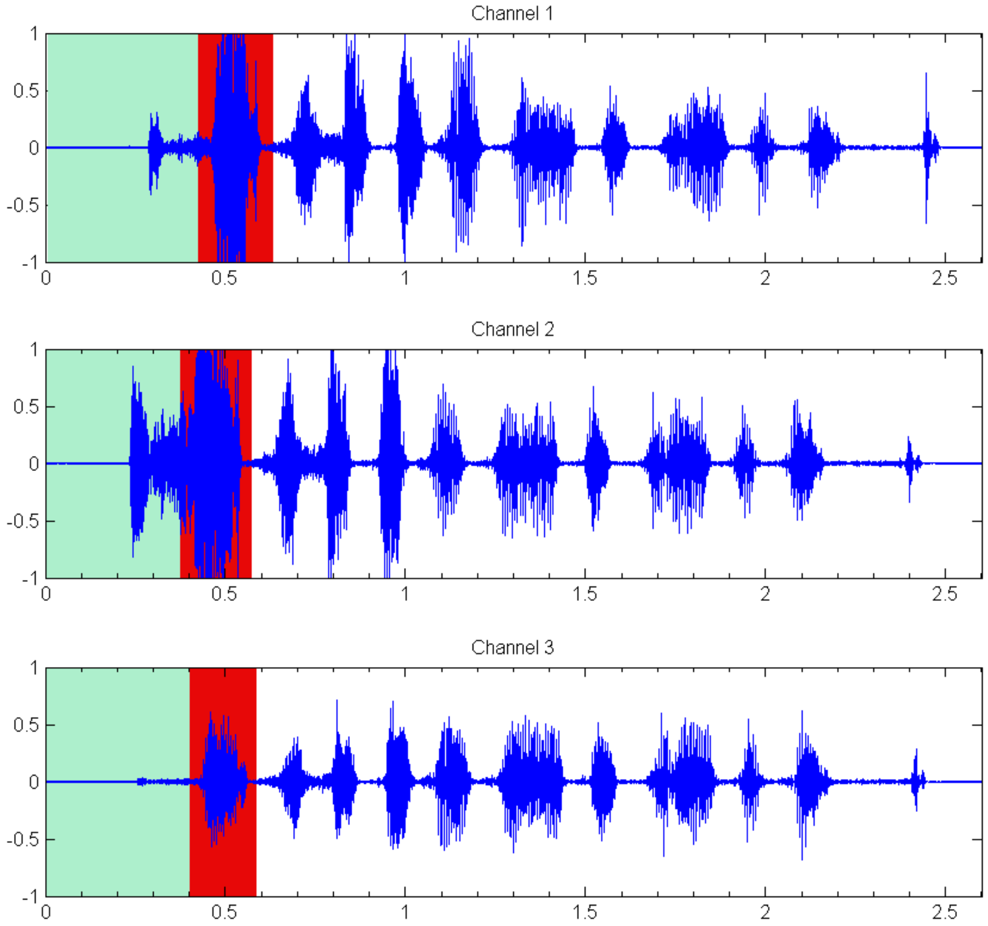


Figure 5.1 – Typical multi-channel reception in ATM-VCS

The issue here is represented by the different features of radio receivers. In this example we can conclude that dynamic compensation block of the 3rd receiver is not enabled, while it is quite clear that the 2nd receiver uses an automatic gain control (AGC) circuit. For the 2nd channel this is suggested by the recorded waveform. The noise is amplified to a specific limit when the PPT button is pressed at the beginning of the transmission. Later, the emergence of voice in the signal increases its power and the output is saturated. Then, the AGC reduces its gain; therefore, the noise level during the following non-speech regions is much lower. Thus, in this example we can see that, after the first utterance, the noise level is similar in all channels.

At this point, a BSS analysis should provide accurate results if computed for the second speech utterances. But the second utterances are situated relatively late in the signals and the results would be obtained after the imposed 300 ms limit. Hence the difficulty of the BSS analysis is increased by the fact that it should be performed on delayed signals, for their first utterances when the noise level may still vary.

A detailed analysis could prove that the channel with the best speech intelligibility could change during a conversation. In this case, at a first glance the BSS should change its first selection in order to adapt to the new situation. However, despite the fact that the first selected channel does not currently provide the best speech quality it is not indicated to change the BSS. This is due to the fact that the operator has already accommodated to the background noise of the first BSS and changing the channel in the middle of the conversation would result in other background noise. But this would cause high discomfort to the operator who would try to adapt to the new background noise. Moreover, changing the channel during the conversation would be more annoying, if conversations are very short which usually happens in ATM.

As various speech features from VAD algorithms are correlated more or less with the speech quality, a solution for the BSS problem could be constructed starting from this observation. Further in this section two solutions will be presented, which were tested and which offered good results.

5.1 WAVELET-BASED BSS

The first solution proposed for the BSS problem is based on the VAD algorithm from [Wu, 2006] which was also presented earlier. It uses wavelet decomposition, Teager energy operator, auto-correlation function and mean delta method to obtain the speech activity envelope (SAE). It has acceptable computing complexity and offers good results.

The performed experiments and analysis over the SAE indicated that this feature is closely correlated with speech quality. This can also be noted in Figure 5.2. The speech signal was corrupted with additive Gaussian white noise at various SNR levels, as presented in Figure 5.2a. When corrupted with SNR = 5 dB the signal is hardly intelligible, while at SNR = 25 dB it is quite easy to understand the speech. Their SAEs are presented in Figure 5.2b and were computed with the wavelet-based VAD algorithm from [Wu, 2006], which was presented earlier in this chapter. The parameters from this experiment were set as follows:

- the frame size was set to 1024 samples, equivalent to 128 ms for a sampling frequency of 8 kHz;
- the frames were overlapped with a factor of 50%;
- the wavelet function was set to 4th order Daubechies (db4);
- the length of the auto-correlation function $p = 40$;
- the Mean-Delta parameter $M = 2$.

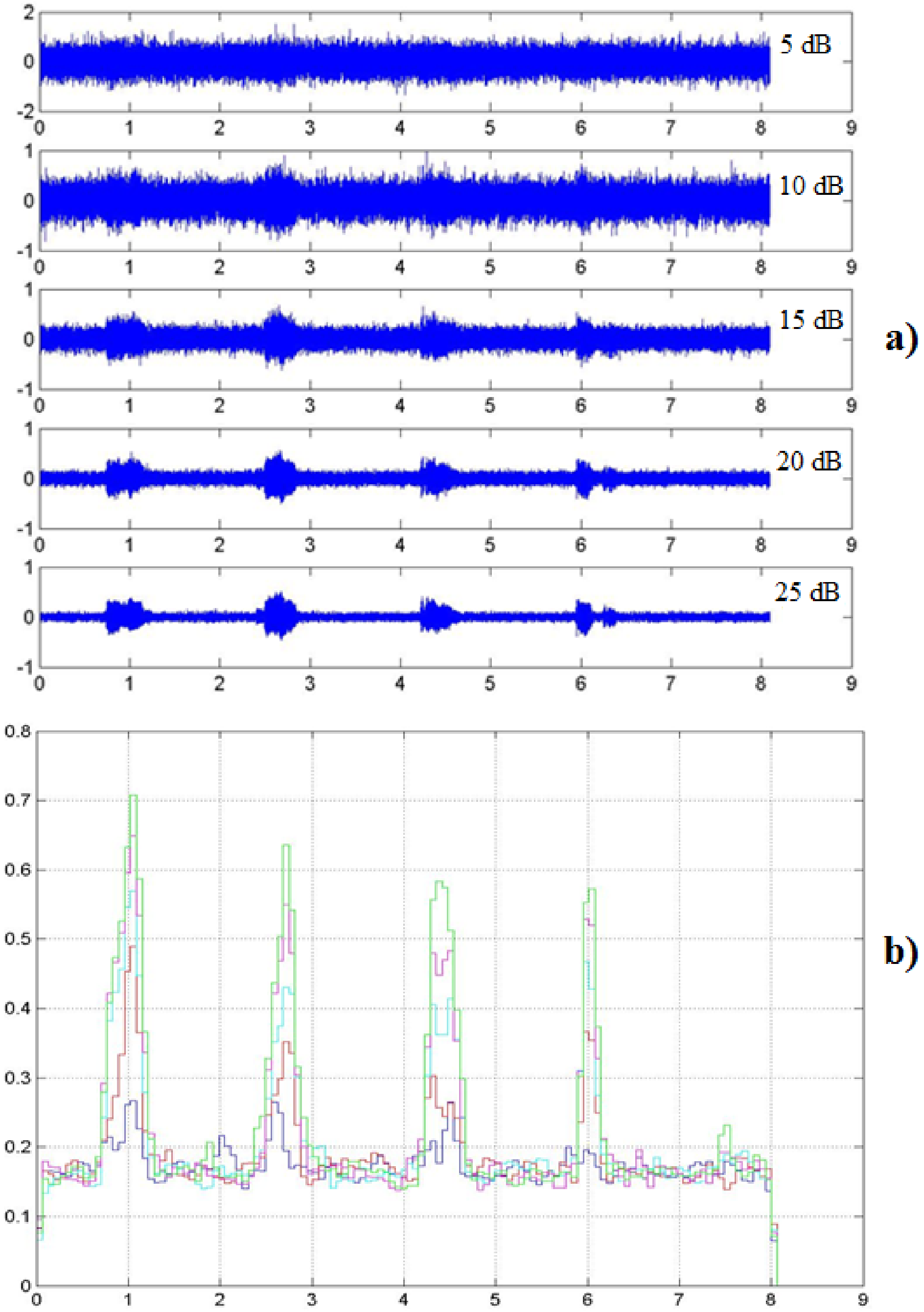


Figure 5.2 – SAE and speech quality: a) noisy signal waveforms for different SNR; b) SAE for different SNR: green – 25 dB, magenta – 20 dB, cyan – 15 dB, red – 10 dB, blue – 5dB

In this case the adaptive threshold was not computed since we needed only the speech activity envelope.

In Figure 5.2b we notice that signals corrupted with higher SNR have higher SAEs. Thus, the higher the SAE, the higher speech intelligibility is assumed. Moreover, during the non-speech intervals, the SAE has similar values for all SNRs. Hence, it can be considered a robust speech and intelligibility feature.

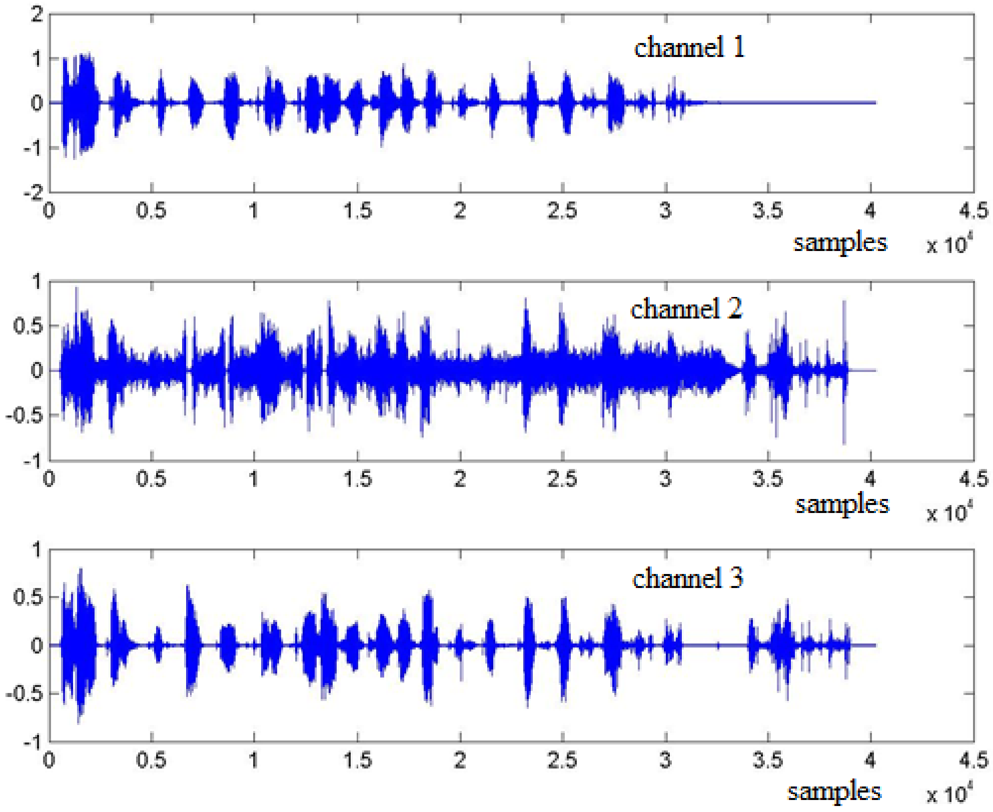


Figure 5.3 – Received signals in ATM and VoIP environment

Based on the previous observations, it appears natural to use the SAE features of the input signals to obtain the solution of the BSS problem. Hence, the first proposal for the BSS solution is described as follows. First, a fixed threshold is set to detect the beginning of the first utterance. In this example, it could be set to around 0.19. Then, for each frame of each channel the SAE is computed. If the SAE of *k* consecutive frames of a channel is higher than the fixed threshold, then we assume that the first utterance appeared on the respective channel. Depending on the length of the frame size, the overlap factor and the maximum response time imposed for the BSS, the *k* parameter may vary. However, in this experiment *k* = 3. Furthermore, if voice activity was spotted on channel *j*, then the accumulated SAE of the j^{th} channel is computed as:

$$ASAE_j = \sum_{i=0}^{i=k-1} SAE_j(m-i) \qquad (5.1)$$

where $SAE_j(m)$ is the speech activity envelope of the m^{th} frame for channel j.

Finally, after a simple comparison, the solution selected for the BSS is the one with the maximum ASAE. In case that voice activity was not detected on a channel, its ASAE will not be compared.

The purpose of Figure 5.2 was to highlight the existing correlation between the speech quality and the speech activity envelope. Therefore, a signal similar to those used in this simulation is rarely found the ATM-VCS. The noise corrupted signals from Figure 5.2 have a constant noise level from the start to the end of the communication. This seems like an ideal situation because the beginning of first utterance is detected much easier.

However, in the real-life conditions described above and presented in Figure 5.1, the beginning of the communication may have variable noise. Also, the voice could appear in less than 50ms from the communication initiated by the PTT. In these cases it is more difficult to distinguish the voice from the largely varying noise. A typical example is shown in Figure 5.3, where the voice starts immediately after the communication was initiated by pressing the PTT button. Their SAE is presented in Figure 5.4.

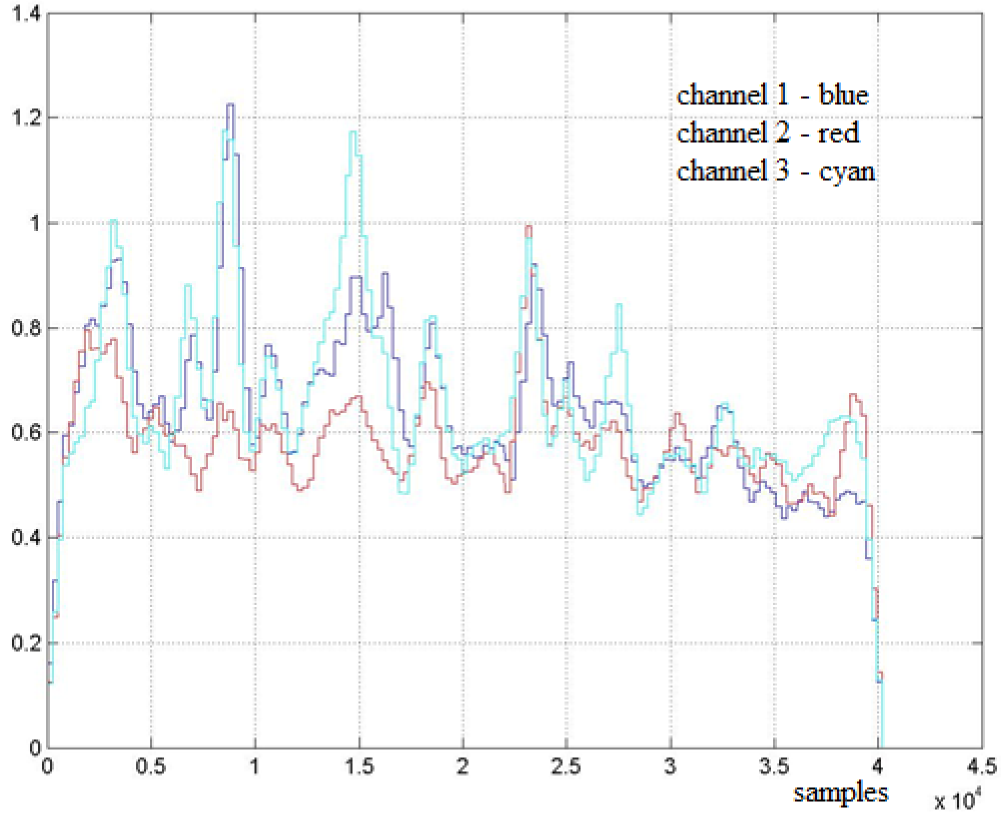


Figure 5.4 – SAE of the signals presented in Figure 5.3

Because another configuration with an overlap factor of 75%, p =24 and M = 8 was used for this measurements, the fixed threshold will have to be modified. Thus, we notice that a fixed threshold around 0.6 leads to good discrimination of speech and non-speech segments. However, for this value, the wavelet based BSS solution which computes the ASAE for each signal will decide the best quality speech quality for channel 2 (red). But, if we look again at Figure 5.3 we see that channel 2 is the noisiest one, thus having the lowest speech quality. To avoid this wrong decision, the fixed threshold should be set higher. Hence, a new fixed threshold around 0.8 will lead to a BSS between channel 1 and 3, which clearly offer better speech quality than the 2nd channel.

We saw above that setting a higher threshold would reduce the probability of a wrong decision with respect to the BSS. However, a higher threshold could also lead to undetected speech utterances or late decisions in low SNR and heavy variable noise level. Thus, when setting the level of the threshold we have to face the compromise between the reduced rate of a wrong decision and late or undetected voice activity.

5.2 LSFM-BASED BSS

The second proposal for solving the BSS problem is based on the long-term spectral flatness measure (LSFM) presented in the previous chapter where it was used in voice activity detection. Compared to the previous BSS solution based on wavelet decomposition, this second option has a better time resolution. In [Ma, 2013] a frame size of 20 ms is proposed with an overlap factor of 50%. In this settings, the LSFM is computed every 10 ms. This is an important advantage against the wavelet based solution, which usually uses 128 ms frames and an overlap factor of 50% or 75%, resulting new SAE every 32 or 64 ms. However, the wavelet based method needs fewer memory resources and smaller computing time.

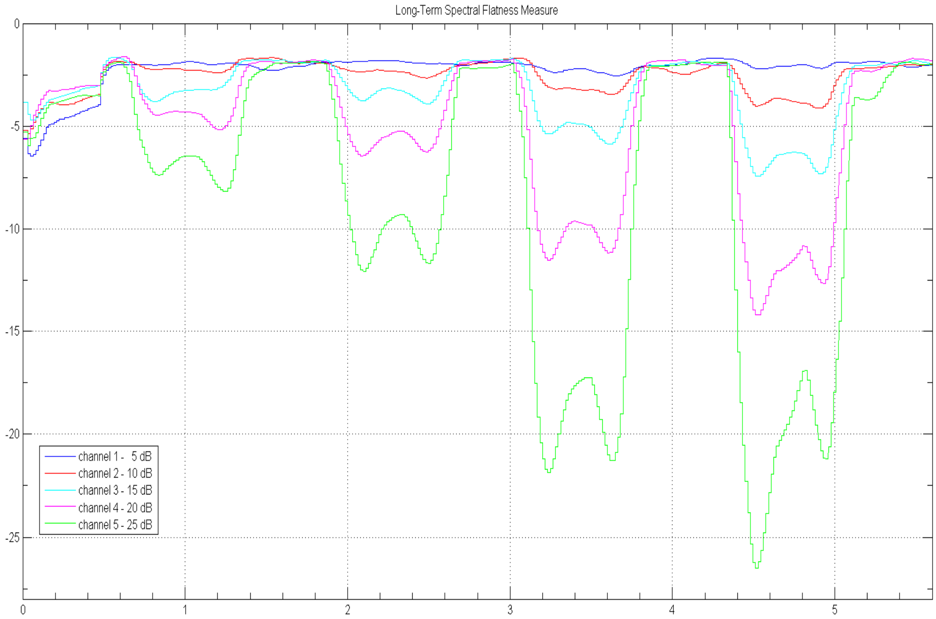


Figure 5.5 – Long-Term Spectral Flatness Measure for BSS

To highlight that the LSFM feature is able to characterize the speech quality, in the next experiment the same signals as in wavelet-based BSS were used. Therefore, the LSFM features of the five corrupted signals in Figure 5.2a are presented in Figure 5.5. For this, a configuration similar to the proposal from [Ma, 2013] was used:

- frame size of 20 ms, resulting in 160 samples at a sampling frequency of 8 kHz;
- overlap factor of 50%;
- the FFT order was decreased to 256 because the sampling frequency is also lower in our case;
- M = 10 previous spectra were used to estimate the current short-time spectrum with the Welch-Bartlett method;
- the LSFM feature was computed over the last R = 30 spectra;
- the LSFM was computed for the frequencies in the range 500 Hz – 4 kHz.

We can notice that, after the initialization process is finished, the LSFM features of each corrupted signal have similar values during the non-speech segments. On the speech segments the higher the SNR, the lower the LSFM. Thus, for the 5th corrupted signals, with the highest SNR (25 dB), the green characteristic has the smallest values during the speech section, while for the first signal, which is the heaviest noise corrupted signal (5 dB), the blue characteristic does not decrease significantly. Therefore the LSFM feature could be associated with speech quality.

Similar to the previous BSS solution, for the second BSS option we have to set a fixed threshold to detect voice activity. When the LSFM feature of channel j has k consecutive values below the threshold, we assume that the voice activity has started and we compute the accumulated long-term spectral flatness measure (ALSFM) of this channel by means of the following formula:

$$ALSFM_j = \sum_{i=0}^{i=k-1} LSFM_j\left(m-i\right) \tag{5.2}$$

where $LSFM_j\left(m\right)$ denotes the long-term spectral flatness measure of the m^{th} frame for channel j. The value of k will now depend on the frame size, the overlap factor and the maximum response time.

If $R=30$ then we have to wait $R*10ms=300ms$ initialization before we can obtain the first LSFM. Considering the demands from the ATM systems and the imposed response time of maximum 300 ms, the above configuration could not be implemented as a BSS solution. Therefore we have to adapt the configuration on the ATM system demands and the following parameters were used:

- frame size of 20 ms, representing 160 samples at a sampling frequency of 8 kHz;

- overlap factor of 50%;
- the FFT order was set to 256;
- $M = 8$ previous spectra were used to estimate the current short-time spectrum with the Welch-Bartlett method;
- the LSFM feature was computed over the last $R = 15$ spectra in order to decrease the initialization time;
- the LSFM was computed for the frequencies in the range 500 Hz – 4 kHz.

With this configuration adapted for air traffic management system demands, the signals in Figure 5.3 were tested again, but this time with LSFM-based BSS. The results are presented in Figure 5.6.

By comparing the LSFM features in Figure 5.6 to the SAE features in Figure 5.4 we notice that the use of LSFM features highlights the differences between the signals much clear. Now it is much easier to conclude that the 2nd channel has the worst speech quality, while the 3rd channel provides the best speech quality. Moreover, with the LSFM feature we can also notice a time difference between the 1st and 3rd channels much easier. This is due to the fact that LSFM has a higher time resolution, in this case 10 ms, while for the SAE feature it was 32 ms.

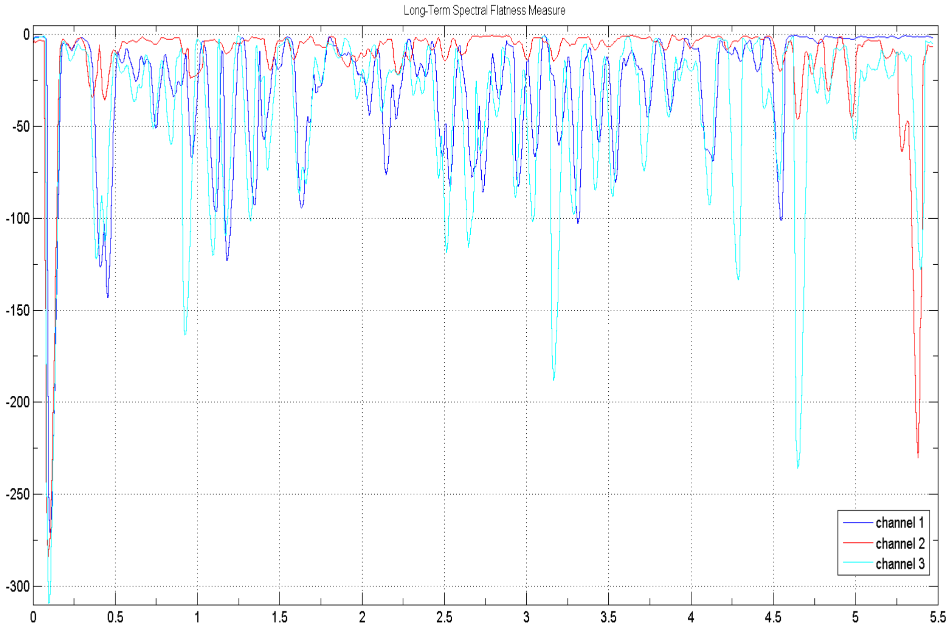


Figure 5.6 – LSFM of the signals presented in Figure 5.3

However, despite the better analysis over the entire signals offered by the LSFM, there are still some issues concerning the BSS. These refer to the fact that the BSS cannot wait to analyze the entire signals and it has to decide during the first 300 ms maximum. In this situation, for the first utterance there is a small difference between the LSFM feature of all channels. If we compute the ALSFM for k = 10 we will find that ALSFM3 < ALSFM_2 < ALSFM_1, meaning that the 1st channel has the worst speech quality and the 3rd channel has the best speech quality. From the BSS point of view the final decision is correct, the best speech quality is found on the 3rd channel, but we have to minimize the possible wrong decisions.

In order to minimize the number of wrong *best signal selections* it is important to analyze the sources which lead to these situations. As the reader could already guess, one major cause is represented by the varying noise level between the start of the received signal and the first utterance. However, we have already discussed about this and in ATM this is difficult to predict and to control if we do not have any information about the configuration of the radio stations with respect to the automatic gain control circuits.

Another important cause in the VoIP environment is related to time delays. As could be seen in Figure 5.6 the delay between the channels could influence the ALFSM feature. A detailed analysis is provided in the next section.

5.3 TIME DELAY AND BEST SIGNAL SELECTION

One of the main causes for wrong decisions with respect to the BSS appears to be the delayed signals. In the ATM-VCS based on TDM communications there was no delay problem because in these networks the delay is known and fixed. However, in the new environment based on VoIP the delays between the received signals through the network are un-known and variable. In these conditions it is important to know how much this can affect the BSS decisions.

The following simulated experiments will measure the exchange rate of the BSS at different delays. We defined the exchange BSS rate as the ratio between the numbers with different BSS decisions after we automatically delayed the signals and the total number of BSS decisions. In other words, for an aligned signals pair we choose the BSS with the first proposal option, based on wavelet decomposition. Then we delay the signals two times, backward and forward, and we reapply the wavelet-based BSS. If the new results indicate the same BSS, then the exchange rate is equal to zero. If the new results indicate 1 or 2 (from backward and from forward introduced delay) then the exchange BSS ratio will be 50%, respectively 100%, for this pair at the tested delay.

The signals for this test were selected from the [Noizeus] database and also from ATM-VCS communications. The wavelet-based BSS was configured as proposed in section 5.1 and the BSS decision was chosen based on ASAE features for k = 3 consecutive frames with voice activity. The introduced forward and backward delay was automatically varied between 2 ms and 120 ms, with a step of 2 ms.

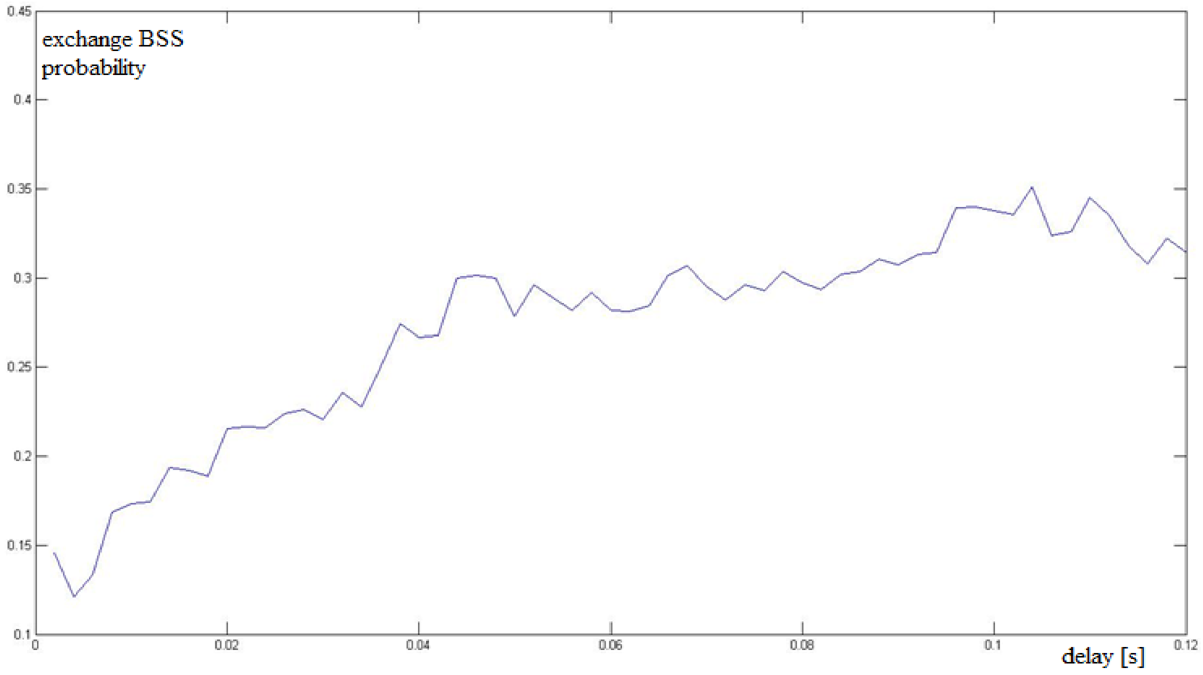


Figure 5.7 – Wavelet-based BSS dependence by the delay between the signals

Generally, the maximum delay in the VoIP environment for ATM-VCS is set to 40 ms through QoS demands. However, because of various causes in some ATM-VCS the QoS cannot be guaranteed. Therefore, Figure 5.7 shows the exchange BSS rate obtained by means of the above mentioned procedure, for a larger delay. We notice that the higher the delay, the higher the exchange BSS rate.

For a delay of 40 ms, the exchange BSS ratio is around 27%, while for delays over 100ms, it is almost 35%. Therefore, we conclude that the delay from VoIP environment from the ATM-VCS considerably affects the decision of BSS. Thus, in order to reduce the wrong BSS decisions special measures must be taken regarding this delay. Generally, it is not possible to obtain a high control of the delay in VoIP environments, hence a time delay estimation method should be used to align the signals before starting the BSS algorithm.

As shown earlier, Chapter 2 presents several TDE methods and Chapter 3 proposes and analyzes TDE methods which fit in ATM-VCS to decrease the probability of wrong BSS decisions.

5.4 OTHER BEST SIGNAL SELECTION ATTEMPTS

5.4.1 MVSS-BASED BSS

In [Jiang, 2010] a new VAD algorithm was proposed, based on maximum values of sub-band SNR (MVSS). It tries to take advantage of the fact that voice activity can be indicated by the high SNR points in sub-bands. Starting from this idea, the voice activity feature based on the MVSS method is computed. This procedure is described in the following steps:

a) the input signal is divided in 32 ms frames, with an overlap factor of 75%;

b) for each frame the power spectrum is computed;

c) the power spectrum is divided in nine sub-bands, similar to ETSI-AMR1, presented in Table 4.1;

d) the sub-band SNRs are estimated based on posteriori SNR of each component;

e) for each sub-band we extract the *M* largest values;

f) the MVSS for the i^{th} sub-band set $G(k)_{(i)}$ of the current frame *k* is computed as:

$$G_{max}(k)_i = \frac{1}{M}\sum_{r=1}^{M} P_{(r)}(k)_{(i)} \tag{5.3}$$

where $P_{(r)}(k)_{(i)}$ represents the r^{th} maximum value from the set $G(k)_{(i)}$ and $M = 6$;

g) the final distant feature gain $D(k)$ is computed as:

$$D(k) = \sum_{i=1}^{9} G_{\max(k)_i} + \sum_{i=1}^{9} [G_{\max(k)_i} - \overline{G_{\max(k)}}]^2 \tag{5.4}$$

where $\overline{G_{\max(k)}}$ is the average of all MVSS sets.

$$\overline{G_{\max(k)}} = \frac{1}{9}\sum_{i=1}^{9} G_{\max(k)_i} \tag{5.5}$$

In order to verify the correlation between this new voice feature and speech quality, the MVSS based feature was tested for the noise corrupted signals in Figure 5.2a. The results are presented in Figure 5.8. As we can see, despite a larger variation of this feature during non-speech sectors, above a fixed threshold around 3 it is correlated with speech quality. Also, higher the distance gain is, higher the SNR and speech quality are.

By comparing this feature to the previous ones we are able to notice a higher time resolution for the MVSS-based feature. This is due to the fact that it uses a frame size of 32 ms and on overlap factor of 75%. Hence, a value of this feature is provided every 8 ms, which is twice faster as for LSFM-based feature. However, these update rates can be modified for each method and a compromise must be made between time resolution and computing time.

Because the results of the above simulation indicated that the MVSS-based feature could be used to characterize speech quality, the feature was further tested with real signals from

ATM-VCS. Keeping the same configuration as in the previous simulation, the obtained gain difference features based on MVSS are presented in Figure 5.9.

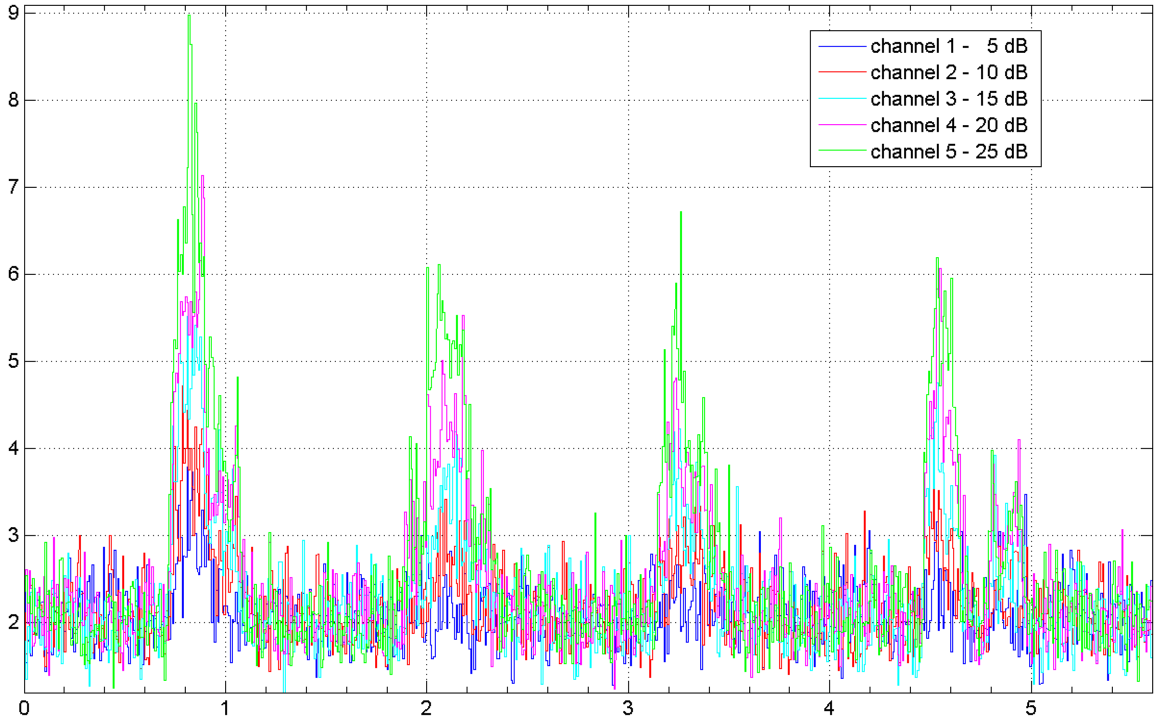


Figure 5.8 – MVSS-based distance gain feature

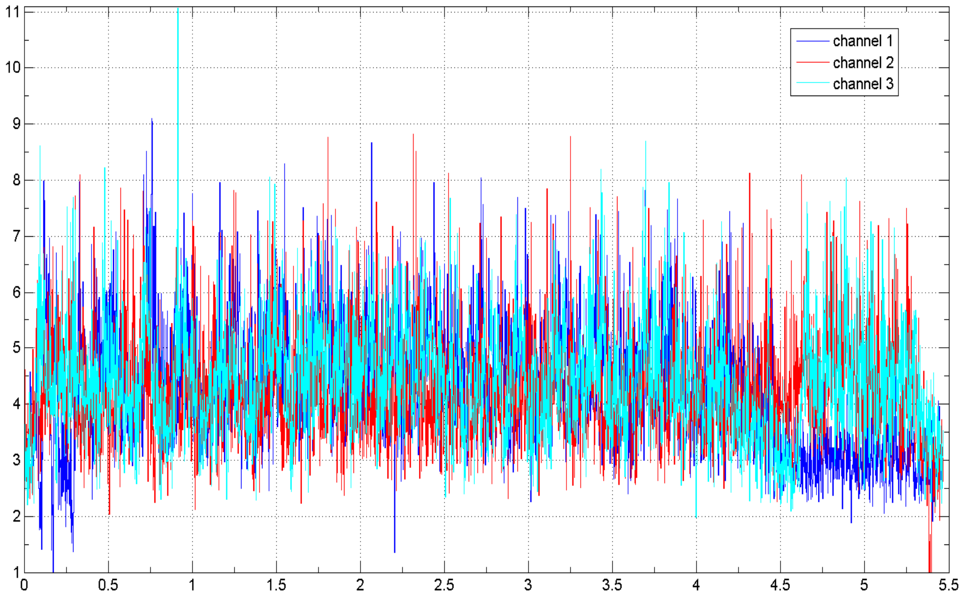


Figure 5.9 – MVSS-based distance gain features LSFM of the signals presented in Figure 5.3

We notice that the distance gain feature of the 2nd channel is generally the lowest one, and that the average distance gains are higher in channel 1 and 2. However, due to the high variations of this feature, it is very difficult to decide on the start of the first utterance. This could not be solved by a higher threshold, which worked for wavelet-based ASAE, because the large variation of this feature will often change its position with respect to the fixed threshold.

Compare to the previous two features, the distance gain leads to a higher incorrect BSS rate. However, because the results in Figure 5.8 suggest that there is a correlation between this feature and speech quality, further work is possible to improve the BSS rate for this feature. In this sense, the important direction to be followed relates to a better discrimination of the first utterance. For this, a possible solution could be a smoothed distance gain in order to minimize its large variations. Another solution could implement a hysteresis mechanism to invalidate sporadic values below the threshold when the first utterance is detected.

5.4.2 NEW 3SFM-BASED BSS

Based on the distance gain from the MVSS method and on the spectral flatness measure (SFM) this thesis proposes a new voice activity feature, called smoothed sub-band spectral flatness measure (3SFM). The idea behind this is to benefit from the strong points of the SFM, but with less memory and time processing than LSFM. Therefore, this new voice feature, which is also based on the MVSS method, can be computed by means of the following similar steps:

a) we divide the signal in 32 ms frames which overlap by 24ms;

b) for each *k* frame the power spectrum $S(k)$ is computed with the DFT;

c) the power spectrum is then divided in nine sub-bands, similar to the ETSI-AMR1 and Table 4.1;

d) for each sub-band j the SFM is computed with the following formula:

$$SFM_j(k) = \log_{10} \frac{GM(j,k)}{AM(j,k)} \tag{5.6}$$

where $GM(j,k)$ and $AM(j,k)$ represent the geometric and arithmetic means for the j^{th}. The above means are computed as shown below:

$$GM(j,k) = \sqrt[L_j]{\prod_{n=1}^{L_j} S_{\mathrm{j}}(k,\omega_n)} \tag{5.7}$$

$$AM\left(j,k\right)=\frac{1}{L_j}\sum_{n=1}^{L_j}S_{\mathrm{j}}\left(k,\omega_n\right) \tag{5.8}$$

where $S_{\mathrm{j}}\left(k\right)$ and L_j stand for the power spectrum and for the number of frequency bins of the j^{th} sub-band, respectively;

e) next it is computed the distance gain for the current frame as:

$$DSFM(k)=\sum_{i=1}^{9}SFM_i(k)+\sum_{i=1}^{9}[SFM_i(k)-\overline{SFM_i(k)}]^2 \tag{5.9}$$

where $\overline{SFM_i(k)}$ represents the average of all sub-bands spectral flatness measure:

$$\overline{SFM_j(k)}=\frac{1}{9}\sum_{i=1}^{9}SFM_j(k) \tag{5.10}$$

f) finally, the smoothed sub-band spectral flatness measure is computed as

$$3SFM\left(k\right)=\alpha\cdot DSFM\left(k\right)+\left(1-\alpha\right)\cdot 3SFM\left(k-1\right) \tag{5.11}$$

where α is the exponential smoothing factor.

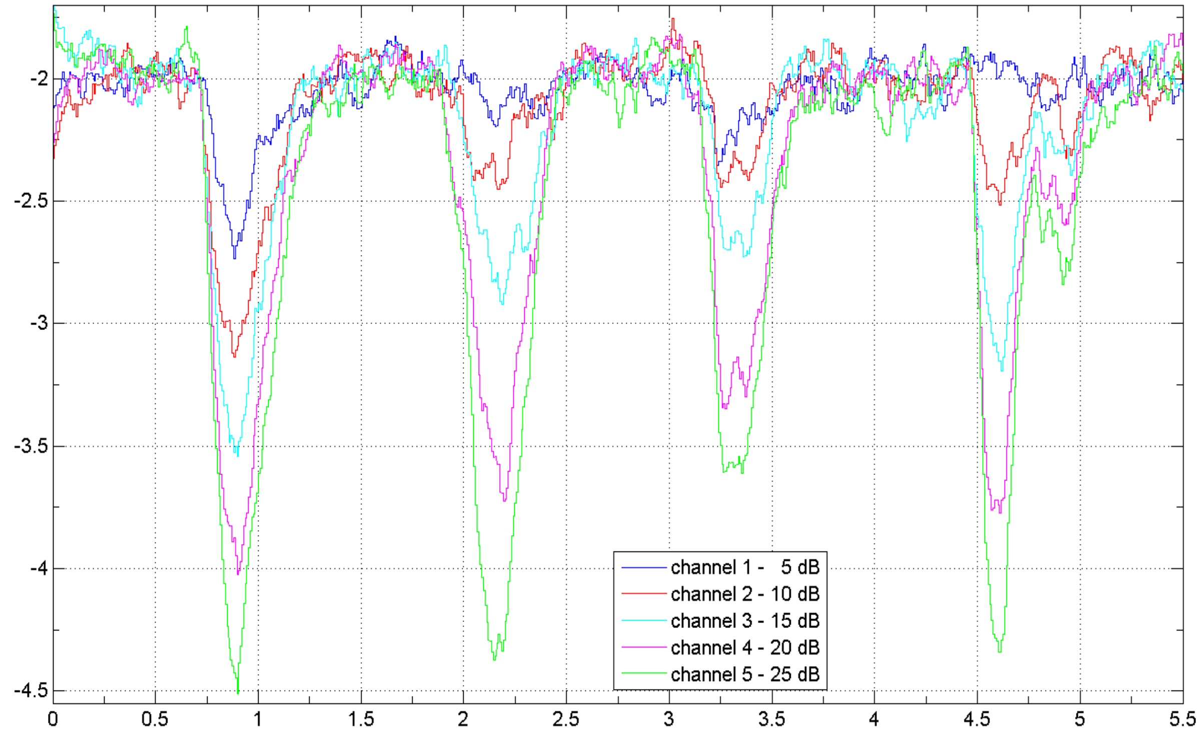


Figure 5.10 – 3SFM features for $\alpha = 0.9$

Depending on the value of α the 3SFM could be used for voice activity detection or as an option for the BSS algorithm. Thus, the following experiments will address both approaches. First, we analyze the correlation between 3SFM and speech quality. For this, we use the same configuration as the previous MVSS-based BSS and the smoothing factor $\alpha = 0.9$. The 3SFM features of the signals in Figure 5.2a are presented in Figure 5.10. For this configuration, we notice that the 3SFM feature is able to indicate the channels with the highest and lowest SNR, hence the feature can be used to decide the BSS. Moreover, we can easily differentiate between speech and non-speech segments using a threshold with a value of about –2.2. As the 3SFM levels have similar values during non-speech periods, the feature is robust to noise and can be used in VAD algorithms.

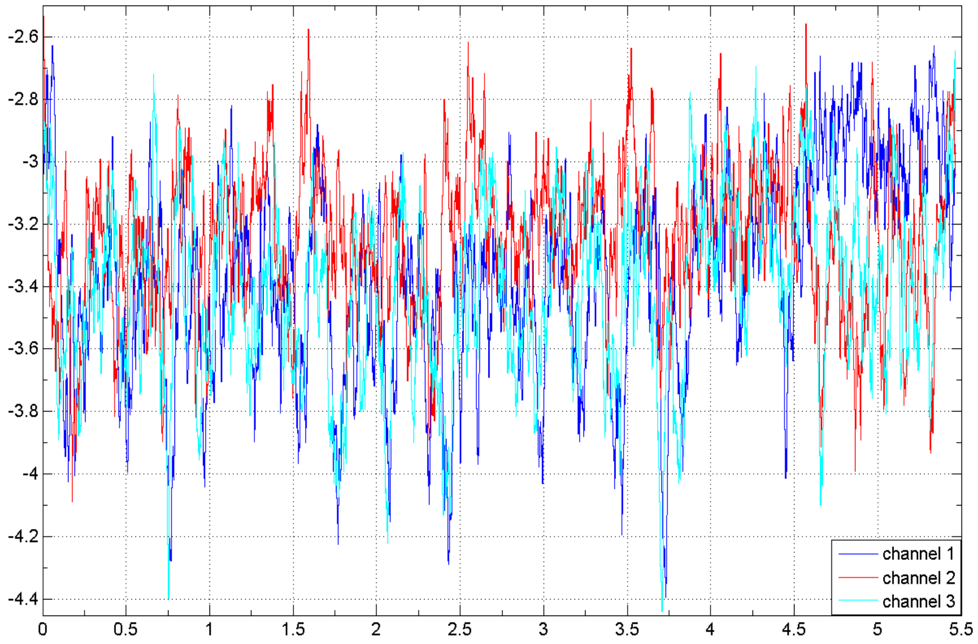


Figure 5.11 – 3SFM feature of the signals in Figure 5.3, $\alpha = 0.9$

In order to obtain more information regarding the 3SFM feature, we apply it to the real signals from the air traffic management system. The result is shown in Figure 5.11, where, for an easier comparison to previous approaches, the input signals were the same as those presented in Figure 5.3.

We can see that the 3SFM features for real air traffic control communication signals have high variations, which are also smaller than those obtained by means of the MVSS method. The smaller variation is explained by the smoothing operation. However, for $\alpha = 0.9$, the variations are high enough to perturb the detection of the first utterance and the final BSS decision.

A solution to this problem is an accentuated smoothing, obtained through a higher smoothing coefficient. Thus, the simulation was repeated for the same signals, but with $\alpha = 0.99$. The results are presented in Figure 5.12, where we can notice that the 3SFM features are more distinctive. Now it is clear that the 2nd channel leads to the worst speech quality, even from the first utterance. Based on the analysis of the features of all channels, for the first utterance the 1st channel offers the best speech quality. However, regarding the BSS decision by analyzing the entire features of the signals it is difficult to decide between 1st and 3rd channel. But this is not due to the proposed feature, but moreover to the similar speech quality of both waveforms, as seen in Figure 5.3.

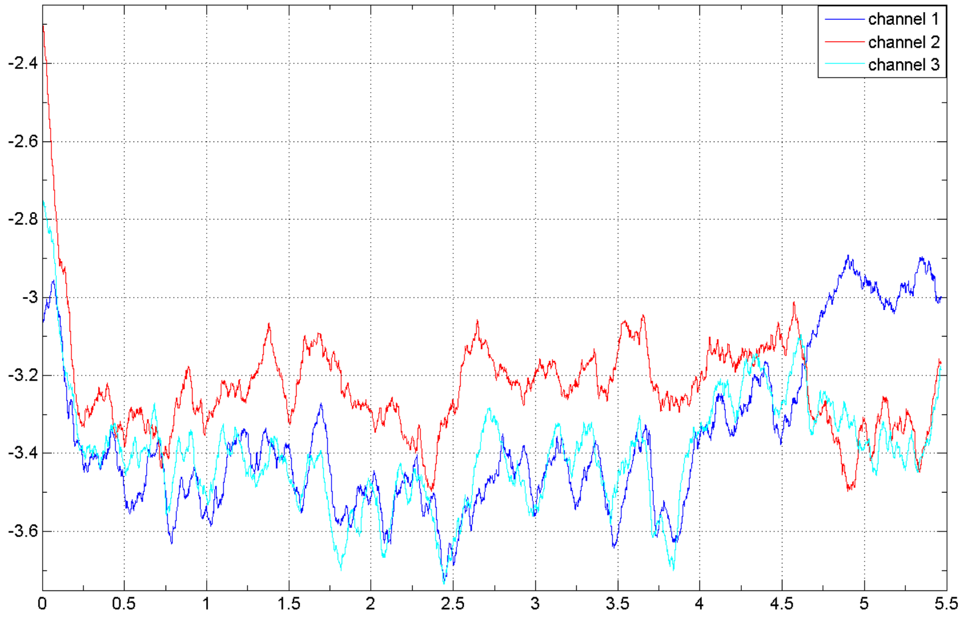


Figure 5.12 – 3SFM features of the signals from Figure 5.3, $\alpha = 0.99$

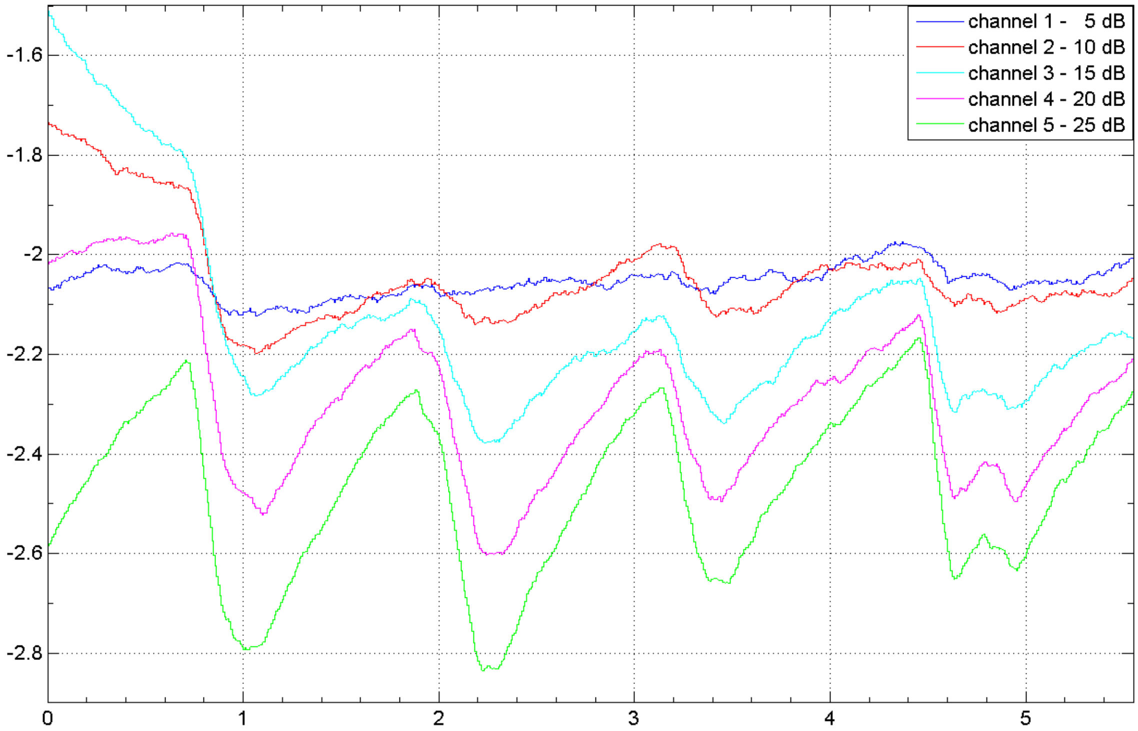


Figure 5.13 – 3SFM features for $\alpha = 0.9$

The over-smoothing with $\alpha = 0.99$ has positive effects for the BSS reducing the wrong decision rate. On the other hand, a higher α extends the exponential smoothing period. This is shown in Figure 5.13 where the 3SFM feature of the corrupted signals in Figure 5.2a was computed. We now notice that during the non-speech periods the 3SFM features do not have similar values. This is explained by the fact that the smoothing period was highly increased and the feature will need more time to update its value for the non-speech periods. Moreover, the feature now cannot be considered a robust one due to these effects and cannot lead to an accurate VAD.

As it was show in previous experiments the new proposed 3SFM feature could be used to implement VAD and BSS algorithms. Due to different characteristics caused by the smoothing parameter α this feature could not be used in the same time as a VAD and BSS component. However, because the smoothing step is the last one when computing this feature, for the cases when VAD and BSS need to work together a solution would be to compute two 3SFM features with two different smoothing factors.

6 ENHANCED BEST SIGNAL SELECTION

The Best Signal Selection must decide which one of several signals is the one that offers the best intelligibility. For this, several algorithms were proposed in the previous chapter. They compute scores for each received signal. Then, depending on the implemented BSS method, the final signal is selected as the one who yields the highest or lowest score. Thus, the BSS treats each signal individually and does not use common information between the received signals.

The above remark led to the idea to use all the information from all the signals to obtain an *enhanced best signal selection* (eBSS). In other words we can use all signals for a multichannel speech enhancement to calculate an eBSS. Therefore, the eBSS should offer higher speech intelligibility than the BSS.

Before discussing how to apply multi-channel speech enhancement methods for eBSS it is important to recall the main characteristics of the signals from ATM-VCS, described below:

- they usually last no more than a few seconds;
- in a VoIP environment they come with a different delay for each communication;

- for the SNR level of each channel we could not assume that it maintains its level from one communication to the next one;
- some receivers use an automatic gain control circuit to maintain a fixed speech level, but this leads to a difficult analysis of the signals.

The aim of the speech enhancement is to improve the signal that was corrupted by noise. For this, various algorithms work in frequency domain trying to estimate the background noise of each frequency bin. Going further, based on various subtraction techniques the enhanced signal is obtained, with a better SNR. However, this usually does not imply a higher intelligibility because of the additional musical noise and other processing artifacts. These were analyzed in several papers and the results from [Chen, 2012] showed the impact of SNR over- and under-estimation. Moreover, Chen and Loizou conclude in [Chen, 2012] that to increase the speech intelligibility better methods are needed to estimate the spectral SNR.

Because in the air traffic control systems the speech intelligibility is very important eBSS should be obtained through an increase of speech quality without any constraints regarding SNR. In our case, the additional noise from each channel depends on the transmission conditions between the aircraft and the radio receiver. Based on the long distance between the radio receivers (hundreds of kilometers) we can assume that the additional channel noises have a very low correlation. Considering that the signals were perfectly aligned we can obtain the eBSS by weighting and summing the input signals. Hence, with correct weightings the eBSS should offer higher speech intelligibility. However, if the weightings are not estimated correctly the eBSS could have a lower speech quality.

In order to obtain the correct weightings we have to compute the power of the received signals and to estimate the power of the background noise of each channel. Then, the lower the estimated SNR errors are, the better eBSS solution is achieved.

6.1 SNR ESTIMATION

The SNR estimation represents an important topic for signal processing and is a prerequisite step in noise cancelation. For speech signals various methods estimate the noise power spectrum by exploiting the non-speech periods of the signal. However, the accuracy of these methods is influenced by how accurate the separation between speech and non-speech segments is. The higher the non-stationary of the noise is, the more difficult to estimate the speech/non-speech segments gets and then the noise power spectrum is less accurately estimated.

In the last years several methods improved the estimation of the spectral noise power. From these ones, the minimum statistics (MS) methods [Martin, 2001][Cohen, 2003][Rangachari, 2006] are well-known. Usually the MS methods assume that even a small fraction of non-speech appears in the total analyzed time-span. Then the spectral noise power is estimated as the minimum value from an analyzed interval. For non-stationary noises when the noise power changes rapidly the power of noise could be over- or under-estimated, or could be tracked with some delay. The shorter the time-span used is, the shorter the tracking delay gets. However, the probability of having only speech segments in the observed time-span increases with the decrease of the time-span. In these cases the noise power will be overestimated leading to annoying artifacts, as residual noise and musical noise, after the speech enhancement stage.

To overcome this issue some methods [Sohn, 1998][Cohen, 2003][Rangachari, 2006] recursively average the noisy power spectrum based on speech presence probability (SPP). The introduction of the soft-decision SPP instead the hard-Boolean-decision from VAD improves the estimation of the noise power, especially in low SNR conditions. However, the tracking delay could not be eliminated. Thus, recent methods were proposed in order to reduce the tracking delay of the estimated spectral noise power for non-stationary noise environments. In [Hendriks, 2008] the subspace DFT approach brings some improvements in non-stationary conditions, with the cost of complex computations [Taghia, 2011], when comparing it to the MS methods. The minimum mean square error (MMSE)-based methods [Yu, 2009] and [Hendriks, 2010] represent an alternative solution, especially the last one which is robust to the variations of the noise level and less computationally demanding [Taghia, 2011].

In [Narayanan, 2012] a new SNR estimation method based on computational auditory scene analysis (CASA) is proposed. Despite its superior estimation when compared to other algorithms it cannot be used for eBSS because of a long processing delay.

6.1.1 Signal Model

The general approach to noise estimation processes the signals frame-by-frame. The DFT is used to transform the time-domain frames into frequency domain. If we consider the speech and noise to be additive then the noisy observation in the frequency domain, *Y*, is expressed as:

$$Y_k(l) = S_k(l) + N_k(l) \tag{6.1}$$

where $S_k(l)$ and $N_k(l)$represent the frequency spectrum of the speech and noise of the *l* frame for the k^{th} frequency bin. In this section, from now on, we will use the short form, without *k* and *l*, with the exception of the cases where the clarification is mandatory. Moreover we will use capital letters for the random variables, lower case letter for their realizations and hat symbol for the estimated values.

Going further, if we consider that the speech and noise are independent and with zero mean we can write the following expression:

$$\mathrm{E}\left(|Y|^2\right) = \mathrm{E}\left(|S|^2\right) + \mathrm{E}\left(|N|^2\right) \tag{6.2}$$

where E(·) denotes the statistical expectation operator. Based on this the spectral power of speech and noise, σ_S^2 and σ_N^2 respectively, are:

$$\sigma_S^2 = \mathrm{E}\left(|S|^2\right) \tag{6.3}$$

$$\sigma_N^2 = \mathrm{E}\left(|N|^2\right) \tag{6.4}$$

The *a priori* and *a posteriori* SNR, ξ and γ respectively, are:

$$\xi = \frac{\sigma_S^2}{\sigma_N^2} \tag{6.5}$$

$$\gamma = \frac{|y|^2}{\sigma_N^2} \tag{6.6}$$

6.1.2 Minimum Mean Square Error-Based Noise Power Estimation

In the MMSE approach for noise power estimation the distributions of speech and noise spectral coefficients is assumed to be Gaussian, as expressed below:

$$p_S(s) = \frac{1}{\sigma_S^2 \pi} \exp\left(-\frac{|s|^2}{\sigma_S^2} \right) \tag{6.7}$$

$$p_N(n) = \frac{1}{\sigma_N^2 \pi} \exp\left(-\frac{|n|^2}{\sigma_N^2} \right) \tag{6.8}$$

Combining these, the distribution of spectral noisy signal coefficients is obtained as:

$$p_Y(n) = \frac{1}{\sigma_N^2 (1+\xi)\pi} \exp\left(-\frac{|y|^2}{\sigma_N^2 (1+\xi)} \right) \tag{6.9}$$

For an easier handling we use the polar notation, the complex spectral noise and noisy speech coefficients can be expressed as:

$$N = De^{j\Delta} \tag{6.10}$$

$$Y = Re^{j\theta} \tag{6.11}$$

Now, the equation (6.8) is transformed into the polar coordinates based on the above notations, resulting:

$$p_{D,\Delta}\left(d,\delta\right)=\frac{d}{\pi\sigma_N^2}\exp\left(-\frac{d^2}{\sigma_N^2}\right) \tag{6.12}$$

Next, based on the combination of (6.7) and the independence and additivity assumptions of the noise and speech the distribution of $p_{Y|D,\Delta}\left(y\,|\,d,\delta\right)$ can be expressed as follows:

$$p_{Y|D,\Delta}\left(y\,|\,d,\delta\right)=\frac{1}{\sigma_S^2\pi}\exp\left[\frac{2dr\cos\left(\delta-\theta\right)-r^2-d^2}{\sigma_S^2}\right] \tag{6.13}$$

The MMSE estimate of the noise power is obtained from the conditional expectation $\mathrm{E}\left(\left|N\right|^2\,|\,y\right)$. Applying Bayes' rule we find

$$\mathrm{E}\left(\left|N\right|^2\,|\,y\right)=\frac{\int_0^\infty\int_0^{2\pi}d^2p_{Y|D,\Delta}\left(y\,|\,d,\delta\right)p_{D,\Delta}\left(d,\delta\right)\mathrm{d}\delta\mathrm{d}d}{\int_0^\infty\int_0^{2\pi}p_{Y|D,\Delta}\left(y\,|\,d,\delta\right)p_{D,\Delta}\left(d,\delta\right)\mathrm{d}\delta\mathrm{d}d} \tag{6.14}$$

Replacing (6.13) and (6.12) into (6.14) and using [Gradshteyn, 2007 – Eqs. 6.643.2 and 8.431.5] we can write:

$$\mathrm{E}\left(\left|N\right|^2\,|\,y\right)=\left(\frac{1}{1+\xi}\right)^2\left|y\right|^2+\frac{\hat{\xi}}{1+\hat{\xi}}\hat{\sigma}_N^2 \tag{6.15}$$

depending on the estimate $\hat{\sigma}_N^2$ and $\hat{\xi}$ of spectral noise power and *a priori* SNR, respectively. This form shows that these values have to be estimated in practice, where in general the speech signal is less stationary than the noise signal [Martin, 2001]. Assuming that the noise power from consecutive frames is correlated then we can use the previous noise power spectrum estimation for the current frame in (6.15), as below:

$$\hat{\sigma}_N^2\left(l\right)=\hat{\sigma}_N^2\left(l-1\right) \tag{6.16}$$

On the other hand, the *a priori* SNR is more difficult to estimate because the speech signal may vary significantly from one frame to another. Thus, in [Hendriks, 2010] a limited ML estimate for the *a priori* SNR in (6.15) is proposed and then a bias compensation, presented in next subsections.

The final noise power spectral density is computed based on its previous value, the current noise periodogram via (6.15) and the smoothing factor α, which in [Hendriks, 2010] is set to 0.8.

$$\hat{\sigma}_N^2(l) = \alpha \cdot \hat{\sigma}_N^2(l-1) + (1-\alpha) \cdot \mathrm{E}\left(|N|^2 \mid y(l)\right) \tag{6.17}$$

Further on we see that the MMSE is unbiased only if the estimated spectral power of noise and speech are the same with their real values. For this we express the expected value from (6.15) with respect to *Y,* $\hat{\sigma}_N^2$ and $\hat{\sigma}_S^2$ which leads to:

$$\mathrm{E}_Y\left(\mathrm{E}\left(|N|^2 \mid Y, \hat{\sigma}_N^2, \hat{\sigma}_S^2\right)\right) = \left(\frac{\hat{\sigma}_N^2}{\hat{\sigma}_S^2 + \hat{\sigma}_N^2}\right)^2 \left(\sigma_S^2 + \sigma_N^2\right) + \frac{\hat{\sigma}_S^2}{\hat{\sigma}_S^2 + \hat{\sigma}_N^2}\hat{\sigma}_N^2 \tag{6.18}$$

if we assume that $\hat{\sigma}_N^2$ and $\hat{\sigma}_S^2$ do not depend on *Y,* and apply [Gradshteyn, 2007 – Eq. 3.381.4]. If $\hat{\sigma}_S^2 = \sigma_S^2$ and $\hat{\sigma}_N^2 = \sigma_N^2$, then from (6.18) results

$$\mathrm{E}_Y\left(\mathrm{E}\left(|N|^2 \mid Y, \hat{\sigma}_N^2, \hat{\sigma}_S^2\right)\right) = \sigma_N^2 \tag{6.19}$$

which proves that the MMSE in (6.15) is unbiased. However, in practice $\hat{\sigma}_S^2 \neq \sigma_S^2$ and $\hat{\sigma}_N^2 \neq \sigma_N^2$, which imply the estimator to be biased because $\mathrm{E}_Y\left(\mathrm{E}\left(|N|^2 \mid Y, \hat{\sigma}_N^2, \hat{\sigma}_S^2\right)\right) \neq \sigma_N^2$ [Yu, 2009][Hendriks, 2010].

MMSE as a VAD

The MMSE can be seen as a VAD-based noise tracer in the cases when the *a priori* SNR is estimated by means of the limited ML estimate.

The MMSE solution from (6.15) combines the noisy observation and the previous estimated spectral noise, where both of them are weighted. The weightings depend on the *a priori* SNR $\hat{\xi}$ and vary between 0 and 1. Hence a soft decision is implied to the solution between σ_N^2 and $|y|^2$. However, if a limited ML estimate is used for the *a priori* SNR, as in [Hendriks, 2010], the soft decision will become a hard decision, which will be shown further.

The limited ML estimate of the *a priori* SNR is expressed as:

$$\hat{\xi}(l) = \max\left(0, \hat{\xi}^{ml}(l)\right) = \max\left(0, \hat{\gamma}(l) - 1\right) \tag{6.20}$$

where

$$\hat{\gamma}(l) = \frac{|y|^2}{\sigma_N^2(l-1)} \tag{6.21}$$

One argument for the limited ML estimate is that it leads to the computation of the bias in an analytic form as in (6.18). Replacing (6.20) into (6.15) results in the following MMSE estimator:

$$\mathrm{E}\left(|N|^2 \mid y\right) = \begin{cases} \sigma_N^2 (l-1) & ,|y|^2 \geq \sigma_N^2 (l-1) \\ |y|^2 & ,|y|^2 < \sigma_N^2 (l-1). \end{cases} \tag{6.22}$$

The above solution can be interpreted as a VAD-based detector. Now there are no weightings and the hard decision is taken by direct comparison between the previous estimate of the noise power and the noisy observation.

Bias compensation

As was shown before, the MMSE estimate from (6.15) is unbiased only when the estimated values match the true values. However, in practice the estimated values differ from the true values and the MMSE estimate is biased. Thus, in order to compensate this we compute the bias using [Gradshteyn, 2007 – Eq.3.381.1] and we obtain:

$$\begin{aligned} B^{-1} &= \frac{\mathrm{E}_Y\left(\mathrm{E}\left(|N|^2 \mid Y, \hat{\sigma}_N^2, \hat{\xi}\right)\right)}{\sigma_N^2} \\ &= \frac{\sigma_S^2 + \sigma_N^2}{\sigma_N^2}\, \Gamma\left(2, \frac{\hat{\sigma}_N^2}{\sigma_S^2 + \sigma_N^2}\right) + \exp\left(-\frac{\hat{\sigma}_N^2}{\sigma_S^2 + \sigma_N^2}\right)\frac{\hat{\sigma}_N^2}{\sigma_N^2} \end{aligned} \tag{6.23}$$

where

$$\Gamma(2,x) = \int_0^x e^{-t} t \,\mathrm{d}t \tag{6.24}$$

is the incomplete gamma function from [Gradshteyn, 2007 – Eq. 8.350.1]. Assuming the spectral noise power is known, $\hat{\sigma}_N^2 = \sigma_N^2$, then the expectation resulted from (6.22) is smaller than the true noise variance; when σ_S^2 is small with respect to σ_N^2, σ_N^2 is underestimated with $B > 1$, [Hendriks, 2010]. Therefore, the estimate spectral noise power is compensated by the next formula:

$$\hat{\sigma}_N^2 = \mathrm{E}\left(|N|^2 \mid y\right) B. \tag{6.25}$$

Safety net

In high non-stationary noises the spectral noise power could be significantly different from one frame to another. If the noise level suffers an abrupt increase in the next frame, it is possible for the spectral noise power tracker to consider this as a speech occurrence. Hence, the estimate spectrum noise power will not be updated correctly. To overcome this situation, a *safety-net* buffer is used [Hendriks, 2010]. The buffer is filled with the last 0.8 seconds of noisy speech periodograms, $|y|^2$. The final value which estimates the spectral noise power is chosen as the maximum value between the current estimation and the minimum value from the safety-net buffer, as expressed below:

$$\hat{\sigma}_N^2(l) \leftarrow \max\left[\hat{\sigma}_N^2(l), \min\left(|y(l-M+1)|^2, \ldots, |y(l)|^2\right)\right] \tag{6.26}$$

where *M* is the number of frames from the safety-net buffer.

This solution assures that the estimate noise power would not stagnate in case of rapidly changing noise levels. However, the tracking delay in these cases could not be eliminated.

6.1.3 Unbiased Estimator Based on Speech Presence Probability

In the previous section we noticed that the ML estimate can be used for the *a priori* SNR. In this case, the MMSE estimator in (6.15) becomes (6.22) and can be seen as a VAD-based spectral noise power estimator because it updates the estimated noise only when $|y(l)|^2 < \hat{\sigma}_N^2(l-1)$. Thus, the resulting estimator is biased. In order to solve this issue, the biased compensation in (6.23) is used.

Introducing a speech presence probability (SPP) instead of the hard-decision implies a soft-decision with an unbiased estimator; thus, the bias compensation will not be used. Therefore, the MMSE estimator based on the SPP is expressed by:

$$\mathrm{E}\left(|N|^2 \mid y\right) = P(\mathcal{H}_0 \mid y)\mathrm{E}\left(|N|^2 \mid y, \mathcal{H}_0\right) + P(\mathcal{H}_1 \mid y)\mathrm{E}\left(|N|^2 \mid y, \mathcal{H}_1\right) \tag{6.27}$$

where non-speech is assumed by $\mathcal{H}_0$ and speech presence by $\mathcal{H}_1$.

Estimation of the SPP

Under the same assumption as in the previous subsection, which stated that the noise and speech coefficients have Gaussian distribution, we express the *a posteriori SPP* with Bayes' theorem as:

$$P(\mathcal{H}_1 \mid y) = \frac{P(\mathcal{H}_1) p_{Y|\mathcal{H}_1}(y)}{P(\mathcal{H}_0) p_{Y|\mathcal{H}_0}(y) + P(\mathcal{H}_1) p_{Y|\mathcal{H}_1}(y)} \tag{6.28}$$

The *a posteriori* SPP cannot be computed without any knowledge about the *a priori* probability models and the likelihood functions of speech absence $p_{Y|\mathcal{H}_0}(y)$ and speech presence $p_{Y|\mathcal{H}_1}(y)$. In order to solve this issue, we use the following equation:

$$P(\mathcal{H}_1) = 1 - P(\mathcal{H}_0) \tag{6.29}$$

Moreover, when no observation is available we can assume an equal probability for presence or absence of speech. Hence, we leave them as $P(\mathcal{H}_1) = P(\mathcal{H}_0) = 0.5$. These values are considered a worst case assumption [McAulay, 1980]. Contrary to [Cohen, 2003], the fixed starting values are not based on any observation.

An indication about how well the observation y suits the modeling parameters is offered by the likelihood functions $\mathrm{p}_{\mathrm{Y}|\mathcal{H}_0}(y)$ and $\mathrm{p}_{\mathrm{Y}|\mathcal{H}_1}(\mathrm{y})$ in (6.28) for non-speech and speech presence. Using the assumption that *y* is complex Gaussian distributed we express the likelihood under the speech presence as:

$$p_{Y|\mathcal{H}_1}(y) = \frac{1}{\hat{\sigma}_N^2\left(1+\xi_{\mathcal{H}_1}\right)\pi}\exp\left(-\frac{|y|^2}{\hat{\sigma}_N^2\left(1+\xi_{\mathcal{H}_1}\right)}\right) \tag{6.30}$$

and the likelihood under speech absence as:

$$p_{Y|\mathcal{H}_0}(y) = \frac{1}{\hat{\sigma}_N^2\pi}\exp\left(-\frac{|y|^2}{\hat{\sigma}_N^2}\right) \tag{6.31}$$

We can notice that there is a difference between the distribution p_Y in (6.9) and the distribution $p_{Y|\mathcal{H}_0}$ in (6.30). In (6.9) ξ represents the true local SNR, while in (6.30) the *a priori* SNR $\xi_{\mathcal{H}_1}$ represents a model parameter for speech presence.

The *a posteriori* SPP can be obtained by replacing (6.30) and (6.31) into (6.28) [Cohen, 2001] as:

$$P(\mathcal{H}_1|y) = \left[1+\frac{P(\mathcal{H}_0)}{P(\mathcal{H}_1)}\left(1+\xi_{\mathcal{H}_1}\right)\exp\left(-\frac{|y|^2}{\hat{\sigma}_N^2}\frac{\xi_{\mathcal{H}_1}}{1+\xi_{\mathcal{H}_1}}\right)\right]^{-1} \tag{6.32}$$

where in [Gerkmann, 2012] it is assumed that $P(\mathcal{H}_0) = P(\mathcal{H}_1)$. Similarly to (6.15), the spectral noise power estimate used in (6.28), (6.30) and (6.31) is employed by its previous value, as presented in (6.16).

Derivation of $\mathbf{E}\left(|N|^2 \mid \boldsymbol{y}, \boldsymbol{\mathcal{H}}_0\right)$ and $\mathbf{E}\left(|N|^2 \mid \boldsymbol{y}, \boldsymbol{\mathcal{H}}_1\right)$

If we consider the *a posteriori* SNR expressed by:

$$\hat{\gamma} = \frac{|y|^2}{\hat{\sigma}_N^2} \tag{6.33}$$

then equation (6.32) can be solved and the solution will depend on $\xi_{\mathcal{H}_1}$ and $P(\mathcal{H}_1 \mid y)$ as follows:

$$\hat{\gamma} = \log\left(\frac{1+\xi_{\mathcal{H}_1}}{\frac{1}{P(\mathcal{H}_1 \mid y)} - 1}\right)\frac{1+\xi_{\mathcal{H}_1}}{\xi_{\mathcal{H}_1}} \tag{6.34}$$

If we choose the optimal *a posteriori* SNR, proposed in [Gerkmann, 2012] to satisfy

$$10\log_{10}\left(\xi_{\mathcal{H}_1}\right) = 15 \text{ dB} \tag{6.35}$$

then the *a posteriori* SNR satisfies $\hat{\gamma} > 1$ for already $P(\mathcal{H}_1 \mid y)$>0.075. Therefore, the ML estimate of the *a priori* SNR in (6.20) can be rewritten as:

$$\hat{\xi}^{ml} = \hat{\gamma} - 1 \tag{6.36}$$

when $P(\mathcal{H}_1 \mid y)$ is large enough due to voice activity. Then we can calculate the optimal estimator under speech presence as:

$$\mathrm{E}\left(|N|^2 \mid y, \hat{\xi}, \mathcal{H}_1\right) = \mathrm{E}\left(|N|^2 \mid y, \hat{\xi}^{ml} = \hat{\gamma} - 1\right) = \hat{\sigma}_N^2 \tag{6.37}$$

For the speech absence $Y = N$ resulting:

$$\mathrm{E}\left(|N|^2 \mid y, \mathcal{H}_0\right) = \mathrm{E}\left(|N|^2 \mid n\right) = |n|^2 = |y|^2 . \tag{6.38}$$

Now we can use (6.37) and (6.38) to rewrite (6.28) as follows:

$$\mathrm{E}\left(|N|^2 \mid y\right) = P(\mathcal{H}_1 \mid y) \cdot |n|^2 + P(\mathcal{H}_0 \mid y) \cdot |y|^2 \tag{6.39}$$

We notice now that the MMSE estimator based on SPP has a soft weighting. The final spectral noise power is computed using the smoothing formula (6.17).

Avoiding stagnation

We noticed earlier that the *VAD-based* MMSE estimator uses a safety-net buffer to keep the updating process of the spectral noise power alive when the noise level faces large variations. For the unbiased MMSE estimator based on the SPP similar situations may occur. When the $\hat{\sigma}_N^2$ is underestimated, the *a posteriori* speech presence probability $P\left(\mathcal{H}_1 \,|\, y\right)$ in (6.32) will be overestimated. This leads to a underestimation of $P\left(\mathcal{H}_0 \,|\, y\right)$ Under these circumstances, in (6.39) we can notice that a delay will appear in the noise power estimation. Moreover, if $\hat{\sigma}_N^2$ is seriously underestimated, then $P\left(\mathcal{H}_1 \,|\, y\right)$ will tend to 1 and $P\left(\mathcal{H}_0 \,|\, y\right)$ to 0. In this case, the noise power would not be updated.

To prevent this stagnation, [Gerkmann, 2012] proposes limiting $P\left(\mathcal{H}_1 \,|\, y\right)$. First we compute the smoothed $P\left(\mathcal{H}_1 \,|\, y\right)$ as:

$$\bar{\mathcal{P}}(l) = 0.9 \cdot \bar{\mathcal{P}}(l-1) + 0.1 \cdot P\left(\mathcal{H}_1 \,|\, y(l)\right) \tag{6.40}$$

If $P\left(\mathcal{H}_1 \,|\, y\right)$ is greater than 0.99, than we assume that the estimator stopped the update process, thus the $P\left(\mathcal{H}_1 \,|\, y\right)$ will be set to maximum 0.99, as follows:

$$P\left(\mathcal{H}_1 \,|\, y(l)\right) \leftarrow \begin{cases} min\left(0.99, P\left(\mathcal{H}_1 \,|\, y(l)\right)\right), & \bar{\mathcal{P}}(l) > 0.99 \\ P\left(\mathcal{H}_1 \,|\, y(l)\right), & \bar{\mathcal{P}}(l) \le 0.99 \end{cases} \tag{6.41}$$

With this implementation, the memory is reduced when comparing it with the previous safety-net approach.

6.2 Distributed Multi-channel Speech Enhancement

Over time, the speech enhancement topic caught the attention of numerous researchers. The foundation of this field is represented by the single channel speech enhancement techniques. They usually aim to increase speech quality by reducing the background noise and distortions. In this sense, various techniques based on spectral subtraction [Boll, 1979], Kalman filter [Tanabe, 2008], Bayesian estimation [Hao, 2009] [Udrea 2010][Ikuta, 2011], wavelet [Lun, 2010] were proposed.

The development of electronics technology influenced the speech domain and more and more applications increased the number from one to two, or several microphones. When two microphones are used, one of them needs to capture a reference noise. In cases of multiple microphones we can discriminate between microphone arrays and distributed configurations. While for the microphone arrays the known geometric configuration plays a major role, the distributed configurations, where the microphones are placed throughout a large area with unknown geometric or spacing information, cannot use such *a priori* knowledge. Thus, the microphone array speech enhancement methods such as [Ven, 1988] and [McCowan, 2001] would not work for distributed configurations. However, general models were developed for these configurations as well, such as [Lotter, 2003] [Trawicki, 2012], as an extension from the single channel, dual channels and microphone arrays.

6.2.1 System and models

The configuration is called distributed microphones because the microphones are spread over large areas like airports, train stations, restaurants, offices etc. In these places, the noise level is generally the same and it propagates in all directions. This is called diffuse noise and has low correlation between microphones [McCowan, 2001]. Thus, for *M* distributed microphone configuration, each microphone *i* would receive the corrupted with additive noise, attenuated and delayed clean signal. Considering that there is a time delay estimation block which will perfectly align the signals, the following equations model results:

$$y_i(t) = c_i s(t) + n_i(t) \tag{6.42}$$

where $c_i \in [0,1]$ is the attenuation factor and $n_i(t)$ the additive noise, which affect the clean signal $s(t)$. From a frame by frame processing in frequency domain, this is expressed as:

$$\begin{gathered} Y_i(l,k) = c_i S(l,k) + N_i(l,k) \\ R_i(l,k) e^{j\vartheta_i(l,k)} = c_i A(l,k) e^{j\alpha(l,k)} + N_i(l,k) \end{gathered} \tag{6.43}$$

where *l* is the frame number and *k* is the frequency bin. Further in this subsection we will use the simplified notation form as:

$$R_i e^{j\vartheta_i} = c_i A e^{j\alpha} + N_i \tag{6.44}$$

where the clean and noisy amplitude are represented by A > 0 and R_i, N_i is the spectral noise, while α and ϑ_i stand for the clean and noisy spectral phases. The aim of the distributive multi-channel speech enhancement is to estimate the clean signal $s(t)$ as accurate as possible, denoted by the clean amplitude *A* and spectral phase α.

[Lotter, 2003] and [Trawicki, 2012] assume Gaussian and Rayleigh distributions for noise and speech likelihood, expressed by:

$$p(Y_i|A,\alpha)=\frac{1}{\pi\sigma_{N_i}^2}\exp\left(-\frac{\left|Y_i-c_iAe^{j\alpha}\right|^2}{\sigma_{N_i}^2}\right) \tag{6.45}$$

$$p\left(A,\alpha\right)=\frac{1}{\pi\sigma_S^2}\exp\left(-\frac{A^2}{\sigma_S^2}\right). \tag{6.46}$$

where $\sigma_{N_i}^2$ and σ_S^2 represent the noise and speech spectral variances. Further we will use the diffuse noise assumption, which means that the noise from each microphone is uncorrelated. Therefore, the conditional joint distribution of the noisy spectral observation $\left\{Y_1,\ldots,Y_M\right\}$ is computed by multiplying the independent noisy spectral observation, as expressed below:

$$\begin{aligned} p\left(Y_1,\ldots,Y_M\,|A,\alpha\right)&=\prod_{i=1}^{M}p(Y_i|A,\alpha)\\ &=\prod_{i=1}^{M}\frac{1}{\pi\sigma_{N_i}^2}\exp\left(-\sum_{\mathrm{i=1}}^{\mathrm{M}}\frac{\left|Y_i-c_iAe^{j\alpha}\right|^2}{\sigma_{N_i}^2}\right) \end{aligned} \tag{6.47}$$

6.2.2 *Spectral amplitude estimation*

Based on the previous models the MMSE estimate of the short-time spectral amplitude (STSA) is obtained as:

$$\hat{A}_{STSA}=E\left[A\,|Y_1,\ldots,Y_M\right]=\frac{\int_0^\infty\int_0^{2\pi}Ap\left(Y_1,\ldots,Y_M\,|A,\alpha\right)p\left(A,\alpha\right)\mathrm{d}\alpha\mathrm{d}A}{\int_0^\infty\int_0^{2\pi}p\left(Y_1,\ldots,Y_M\,|A,\alpha\right)p\left(A,\alpha\right)\mathrm{d}\alpha\mathrm{d}A}. \tag{6.48}$$

Introducing in (6.48) the statistical models from (6.46) and (6.47), we obtain the following solution for $\hat{A}_{STSA}$:

$$\hat{A}_{STSA}=\Gamma\left(1.5\right)\left(\frac{\sigma_S^2}{1+\sum_{i=1}^{M}\xi_i}\right)^{\frac{1}{2}}\exp\left(-\frac{v}{2}\right)\left[\left(1+v\right)I_0\left(\frac{v}{2}\right)+vI_1\left(\frac{v}{2}\right)\right], \tag{6.49}$$

with

$$v = \frac{\left| \sum_{i=1}^{M} \left(\frac{\sqrt{\xi_i}}{\sigma_{N_i}} \right) Y_i \right|^2}{1 + \sum_{i=1}^{M} \xi_i} , \tag{6.50}$$

where ξ_i represents the *a priori* SNR, and $I_0(\cdot)$ and $I_1(\cdot)$ are the first kind modified Bessel function of 0th and 1st order. As we can see in (6.50), v is computed as the weighting SNR sum of the noisy observations, which is then normalized by the *a priori* SNR sum. For the particular case of $M = 1$, we obtain the single channel STSA estimator in [Ephraim, 1984].

6.2.3 Log-Spectral Amplitude Estimation

Due to the perception characteristic of the ear a log-MMSE estimate was proposed in [Ephraim, 1985]. If we apply the previous STSA estimate to it, we obtain the log spectrum amplitude (LSA) estimate:

$$\hat{A}_{LSA} = \exp\left(E\left[\ln(\mathrm{A}) \mid Y_1,\ldots,Y_M \right] \right) = \exp\left(E\left[Z \mid Y_1,\ldots,Y_M \right] \right), \tag{6.51}$$

where

$$E\left[Z \mid Y_1,\ldots,Y_M \right] = \frac{\mathrm{d}}{\mathrm{d}\mu} \left[\phi_{Z\mid Y_1,\ldots,Y_M}(\mu) \right] \Big|_{\mu=0} , \tag{6.52}$$

and $\phi_{Z\mid Y_1,\ldots,Y_M}(\mu) = E\left[A^{\mu} \mid Y_1,\ldots,Y_M \right]$ represents the moment generation function:

$$\phi_{Z\mid Y_1,\ldots,Y_M}(\mu) = \frac{\int_0^{\infty} \int_0^{2\pi} A^{\mu} p\left(Y_1,\ldots,Y_M \mid A,\alpha \right) p\left(A,\alpha \right) \mathrm{d}\alpha \mathrm{d}A}{\int_0^{\infty} \int_0^{2\pi} p\left(Y_1,\ldots,Y_M \mid A,\alpha \right) p\left(A,\alpha \right) \mathrm{d}\alpha \mathrm{d}A} . \tag{6.53}$$

Introducing in (6.53) the statistical models from (6.46) and (6.47), we obtain the following solution for $\phi_{Z\mid Y_1,\ldots,Y_M}(\mu)$:

$$\phi_{Z\mid Y_1,\ldots,Y_M}(\mu) = \frac{\Gamma\left(\frac{\mu}{2} + 1 \right)}{\left(1/\lambda \right)^{\mu/2}} \cdot {}_1F_1\left(-\frac{\mu}{2}; 1; -v \right), \tag{6.54}$$

where

$$\frac{1}{\lambda} = \frac{1}{\sigma_S^2} + \sum_{i=1}^{M} \frac{c_i^2}{\sigma_{N_i}^2} , \tag{6.55}$$

and $\bullet_1 F_1(\bullet;\bullet;\bullet)$ represents the confluent hypergeometric function equation [Gradshteyn, 2007 - Eq. 9.210].

Finally, the multichannel LSA estimator is obtained after differentiation and followed by exponentiation of (6.54) and the $\hat{A}_{LSA}$ solution is:

$$\hat{A}_{LSA} = \left(\frac{\sum_{i=1}^{M} \xi_i / \gamma_i}{\sum_{i=1}^{M} c_i^2 / R_i^2} \right)^{1/2} \left(\frac{\left| \sum_{i=1}^{M} \left(\sqrt{\xi_i} / \sigma_{N_i} \right) Y_i \right|}{1 + \sum_{i=1}^{M} \xi_i} \right) \exp\left(\frac{1}{2} \int_{v}^{\infty} \frac{e^{-t}}{t} \mathrm{d}t \right), \tag{6.56}$$

where v is the same as in (6.50). For the particular case of $M = 1$, we obtain the single channel LSA estimator in [Ephraim, 1985].

6.2.4 Spectral Phase Estimation

The estimation of the spectral phase plays an essential role in the final true source reconstruction. Less accuracy for the estimated spectral phase leads to more distortions of the optimal STSA and LSA estimators. Thus, to avoid having the estimated spectral phase alter the optimal STSA and LSA estimator, we use the constrained optimization formulation, regarding to the minimum:

$$\min_{e^{j\hat{\alpha}}} E\left[\left| e^{j\alpha} - e^{j\hat{\alpha}} \right|^2 \right], \text{ subject to} \left| e^{j\hat{\alpha}} \right| = 1. \tag{6.57}$$

Using the Lagrange multiplier optimization method, (6.57) is rewritten as follows:

$$\min_{g,\rho} E\left[\left| e^{j\alpha} - g \right|^2 \left| Y_1, \ldots, Y_M \right. \right] + \rho\left(|g| - 1 \right) \text{subject to} \, |g| = 1, \tag{6.58}$$

with

$$g = e^{j\hat{\alpha}} = g_R + j g_I, \tag{6.59}$$

where ρ represents the Lagrange multiplier.

Based on (6.58) we find the solution of the constrained MMSE spectral phase as:

$$\hat{\alpha} = \tan^{-1}\left(\frac{g_I}{g_R} \right). \tag{6.60}$$

where we make use of the following relationship:

$$\frac{g_I}{g_R} = \frac{E\left[\sin\alpha \mid Y_1,\ldots,Y_M\right]}{E\left[\cos\alpha \mid Y_1,\ldots,Y_M\right]}. \tag{6.61}$$

Substituting in (6.61) the attenuated spectral amplitude and variance as $A_i = c_i A$ and $\sigma_{S_i}^2 = c_i \sigma_S^2$, respectively, we obtain the solution for $\hat{\alpha}$ as:

$$\hat{\alpha} = \tan^{-1}\left(\frac{\sum_{i=1}^{M} \left(\sqrt{\xi_i} / \sigma_{N_i} \right) \operatorname{Im}(Y_i)}{\sum_{i=1}^{M} \left(\sqrt{\xi_i} / \sigma_{N_i} \right) \operatorname{Re}(Y_i)} \right). \tag{6.62}$$

For the particular case of $M = 1$, with the multichannel STSA and LSA estimators in (6.49) and (6.56) respectively, we obtain the popular single channel spectral phase estimator from [Ephraim, 1984].

6.3 IMPLEMENTED EBSS

6.3.1 PROPOSED EBSS DESCRIPTION

Based on the state-of the art [Gerkmann, 2012] unbiased MMSE using speech presence probability SNR estimation, an eBSS solution was developed. Assuming that the signals are perfectly aligned, the idea of eBSS is to compute the power of each noisy signal and to estimate their SNR. Then, the enhanced eBSS solution would be computed as a weighted sum of all the input signals in time domain. Thus, the speech quality improvements of this approach depend on the SNR estimation accuracy and on the chosen weightings.

Computing the power of the signals is necessary to adjust all the signals to the same level. Hence this is the first weighting operation and each l frame of j^{th} signal will be adjusted by the following weight:

$$w_{1,j}(l) = \frac{1}{P_j(l)} \frac{\sum_{i=1}^{M} P_i(l)}{M}, \tag{6.63}$$

where M is the number of channels and $P_i(l)$ is the computed power of the l^{th} frame of channel *i*.

After estimating the SNR of each first-order weighted channel, a second weighting operation must be performed. The higher the SNR is estimated on a channel, the higher the second weighting gets. Thus, the second weightings are expressed as:

$$w_{2,j}\left(l\right)=1-\frac{\sigma_{N_j}^2\left(l\right)}{\sum_{i=1}^{M}\sigma_{N_i}^2\left(l\right)}, \tag{6.64}$$

where $\sigma_{N_i}^2\left(l\right)$ represents the estimated noise power spectrum of i^{th} channel of frame *l,* computed as in [Gerkmann, 2012]. Using (6.63) and (6.64) we obtain the final weighting of the l^{th} frame:

$$w_j\left(l\right)=w_{1,j}\left(l\right)w_{2,j}\left(l\right) \tag{6.65}$$

Usually, the processing frames are overlapped, thus for two consecutive frames the overlap samples have to be multiplied by one of the two frames weightings. The effect of the two consecutive highly different weightings is reduced if the linear smoother combination is applied, as expressed below:

$$u_j(k,l)=\begin{cases} w_j(l)y_j(k) & ,0 < k < N_{OVL} \\ \left[w_j(l)+\dfrac{w_j(l+1)-w_j(l)}{N_{OVL}}(k-N_{OVL})\right]y_j(k) & ,N_{OVL}\leq k\leq N_{FSize} \end{cases} \tag{6.66}$$

where N_{FSize} is the frame size, N_{OVL} represents the number of overlapping samples for the frame-by-frame processing of the noisy signal $y\left(k\right)$ of the j^{th} channel. Then, the final eBSS signal is computed as the average of all weighted signals:

$$u_{eBSS}\left(k\right)=\frac{\sum_{i=1}^{M}u_i\left(k\right)}{M}. \tag{6.67}$$

6.3.2 *Experimental results*

Several simulated experiments were performed based on the above eBSS description. In order to evaluate the speech quality improvements, the Perceptual Objective Listening Quality Assessment (POLQA) standard was used with the Mean Opinion Score (MOS). Therefore, the POLQA-MOS was calculated for separated noisy signals and for the multi-channel speech enhancement eBSS.

The signals used in this experiment were selected from the [Noizeus] database, which was already used in this thesis, for TDE experiments in chapter 3. This speech corpus contains 8 different noisy variants for each clear signal, this corresponding to M = 8 channels. Further, the eBSS was computed based on a frame-by-frame processing, with an overlap factor of 50% and 256 samples per frame configuration.

The POLQA-MOS for the eBSS and for the 8 noise corrupted instances at SNR = 0 dB of the clean signal are shown in Table 6.1. Another multi-channel speech enhancement was also evaluated, which used equal weightings. In fact, this is obtained as the time-domain average of all noisy instances. We can notice that the eBSS increases the POLQA-MOS when comparing it with each noisy instance. However, the *average signal* leads to a better intelligibility score.

TABLE 6.1 – POLQA-MOS EVALUATION

Signal	**POLQA-MOS**
airport	1.61
babble	1.59
car	1.66
exhibition	1.67
restaurant	1.65
station	1.61
street	1.72
train	1.61
eBSS	2.46
equal weightings	2.62

We were expecting the eBSS to perform better than the *average signal*, because eBSS does not use fixed weightings and because it adapts them according to the variations of the background noise. The fact that the *average signal* yields a higher score than eBSS means that the eBSS is not accurate enough. This can be explained by the noise power spectrum estimation errors, the first and second improper weightings and the linear smoothing.

As further steps, modifying the weightings calculation could represent a solution to improve eBSS. Thus, the first and second weightings in (6.63) and (6.64) could be adapted as follows:

$$w_{1,j}(l) = \frac{1}{\sqrt{P_j(l)}} \frac{\sum_{i=1}^{M} \sqrt{P_i(l)}}{M}, \tag{6.68}$$

and

$$w_{2,j}(l) = 1 - \frac{\sigma_{N_i}(l)}{\sum_{i=1}^{M} \sigma_{N_i}(l)}, \tag{6.69}$$

respectively. This way, the transition from power spectrum to amplitude operations should be corrected.

Another alternative is represented by the SNR estimator. In this experiment the unbiased MMSE estimator was used, based on speech presence probability [Gerkmann, 2012]. While it assumes that the spectral power distribution is Gaussian for voice, other papers such as [Lotter, 2003] and [Trawicki, 2012] assume the voice spectral power distribution to be Rayleigh.

An unexploited option of the thesis experiments so far is represented by the implementation of the recently proposed distributed multi-channel speech enhancement [Trawicki, 2012].

7 CONCLUSIONS

7.1 GENERAL CONCLUSIONS

The main objective of this thesis was to develop a Best Signal Selection (BSS) method for air traffic control systems a in VoIP environment. This main target can be split into the following tasks: time delay estimation (TDE) and voice activity detection (VAD). For the VoIP environment, the TDE is an important issue due to the fact that the received signals arrive with variable delay for each new communication. Thus, the delays have to be estimated as accurately as possible in order to align the signal for further processing. The second task, VAD, has been correlated in this thesis with speech intelligibility in order to derive BSS solutions.

Regarding the TDE for BSS topic from the air traffic management and control systems it has shown that only the generalized cross-correlation (GCC) TDE method can provide a useful real-time solution. Other methods, such as adaptive filtering, adaptive eigenvalue decomposition, difference functions or wavelet-based TDE methods are not appropriate for this kind of applications.

Therefore, in chapter 3 of this thesis, the traditional GCC-TDE methods were evaluated. Moreover, the evaluation was extended by applying the accumulated cross-power spectrum technique for all well-known GCC functions in case of a multi-frame processing scheme. The experiments were performed using the standard Noizeus database.

The analysis showed that, after proper calibration, the ρCSP and the ρCSPC methods provide the lowest error rate, with a little advantage for the first one. CPS-m, CC, CSP and HB have reasonable error rate results. Under the same circumstances, the other methods like Eckart, ROTH, SCOT, HT (ML) and Wiener do not provide acceptable error rate results. Regarding the processing time, the normal cross correlation is the faster method, as it does not compute any weighting. On the opposite side, ρCSPC has to perform many time-consuming operations to calculate the weighting function, which results as the slowest one. Among the three schemes presented in chapter 2 and 3, the first one, which analyzes the signals using one large frame, is the slowest, but it offers the highest accuracy. The second scheme, which works over multiple frames of fewer samples averaging the final estimate in time domain, is a little faster, but does not provide any usable accuracy results. The third scheme works on smaller frames, similar to scheme 2, but it accumulates the cross-power spectrum in frequency domain. This leads to a good accuracy and it is also the fastest processing scheme.

The results in chapter 3 could be used for a better decision regarding the implementation of GCC methods, based on the application demands. It was also proved that *acc-ρCSP* could be successfully used for the TDE issue, to realign the received signals in the VoIP environments of the ATM-VCS.

Moreover, for expected delays which are comparable with the available analysis window, it is recommended to use a single large frame implementation. But if the expected delays are much smaller than the available analysis window, a faster solution is the accumulation scheme of the cross-power spectrum in frequency domain. Each of these schemes can be efficiently implemented to provide solutions for realigning noisy signals in applications such as speech enhancement, echo canceling, seismic and medical processing, radar and sonar localization, and pattern detection.

Regarding the BSS solutions, the correlation between speech intelligibility and VAD scores of different algorithms was analyzed in chapter 5. The experimental results showed that wavelet-based VAD [Wu, 2006] and long-term spectral flatness measure (LSFM) VAD [Ma, 2013] could be used to obtain new accumulated speech activity measures. These new measures are proposed in this thesis as solutions for the BSS problem.

In chapter 5 a new proposal for the VAD and BSS solution was also analyzed, the smoother sub-band spectral flatness measure (3SFM). This new feature, which combines LSFM and maximum values sub-band SNR (MVSS) from [Jiang, 2010], can be tuned for VAD or for accumulated speech activity measure. Besides the previous two proposals for the BSS problem, the 3SFM also provides a good correlation between speech intelligibility and VAD score; therefore, it could be used in ATM-VCS.

Besides the main objective, a secondary one attracted my interest. This is the *enhanced BSS* (eBSS), which should be obtained from a multi-channel speech enhancement approach.

It should offer a more intelligible signal than the normal BSS, based on all received signals. Considering that the input signals are aligned, a solution for eBSS was proposed in chapter 6, based on the state of the art SNR estimation [Gerkmann, 2012], followed by two weighting and smoothing operations. Despite the fact that the proposed eBSS solution increased speech intelligibility, measured by POLQA-MOS, the increase was not as high expected. Thus, this task remains open for improved solutions.

7.2 Personal Contributions

My personal contributions can be found in chapters 3, 5 and 6 of this thesis and are summarized as follows:

a) Two new GCC-TDE methods for multi-frame analysis were proposed, *acc-ρCSPC* and *acc-ρCSP*. They are based on cross power-spectrum phase (CSP) combining previous efficient methods that use accumulating cross power spectrum, whitening and coherence;

b) The accumulation principle for multi-frame analysis TDE was extended for the rest of 8 well-known GCC functions, such as normal correlation, Eckart, Roth, SCOT, HB, CPS-*m*, HT, and Wiener;

c) The GCC-TDE methods implemented in various schemes were characterized in detail, with regards to the accuracy and error rate, relative error and standard deviation of relative error;

d) The processing time results and implementation aspects in different schemes of all 11 well-known GCC-TDE methods were presented, compared and discussed;

e) A solution to realign signals from air traffic management and control systems in VoiP environment was developed, for further BSS or eBSS, based on the *acc-ρCSP*;

f) The correlation between VAD scores and speech intelligibility was analyzed;

g) Three BSS solutions were proposed the accumulated speech activity envelope, the accumulated long-term spectral flatness measure and the accumulated smoothed sub-band spectral flatness measure. These measures can be used to decide between several signals, which offers the best speech intelligibility;

h) A new VAD algorithm was proposed, smoothed sub-band spectral flatness measure, based on maximum values of sub-band SNR and long-term spectral flatness measure. From this VAD the third BSS solution was derived;

i) An eBSS solution for air traffic management and control systems in VoIP environment was proposed, based on *acc-ρCSP* TDE, unbiased SNR estimation based on speech presence probability with fixed priors, followed by two weighting and smoothing operations.

7.3 FUTURE WORK

The author of this thesis is interested in continued the work which was not concluded. Thus, regarding the TDE issue, the future work will involve analysis of *CPS-m* and *acc-CPS-m*. Better accuracy results are expected after a proper calibration for *m*; therefore, a study around this value is needed.

The VAD problem remains open for further developments, which can be used in air traffic management and control systems from VoIP environment, and also for other applications. The research and development work in this area could also provide a better BSS solution.

Regarding the multi-channel speech enhancement, the proposed eBSS solution in chapter 6 did not lead to the anticipated results. Therefore, it is expected that the current eBSS solution will be improved by adjusted weighting and smoothing methods. The distributive multi-channel speech enhancement approach was not experimentally exploited until now, so it naturally demands further attention.

Finally, I have to admit that work on this thesis triggered my interest for research in the field of digital signal processing. Thus, besides the above concrete future tasks, I will happily embrace further research in this field.

Publications List

1.[Marinescu, 2013a] **Marinescu, R.S.**, Buzo, A., Cucu, H., Burileanu, C,. "New Considerations for Accumulated ρ-Cross Power Spectrum Phase with Coherence Time Delay Estimation", *In Procedings of ICDT 2013, The Eight International Conference of Digital Telecommunications*, Venice-Italy, pp. 55-59, Apr. 2013 – **invited paper for IARIA journals.**

Time delay estimation (TDE) remains an important research issue because of its several approaches and large field of digital signal applications. As a solution for this topic, in this paper, we continue the evaluation of the recently proposed *accumulated ρ-cross power spectrum with coherence* TDE method. The experimental results confirm that the method is faster and more accurate than the previous separated variants. Another key finding is that the TDE based on accumulation of cross-power spectrum is at least twice as accurate as the TDE based on time domain averaging.

2.[Marinescu, 2013b] **Marinescu, R.S.**, Buzo, A., Cucu, H., Burileanu, C,. "Fast Accurate Time Delay Estimation Based on Enhanced Accumulated Cross-Power Spectrum Phase", *21st European Signal Processing Conference (EUSIPCO 2013)*, Marrakesh-Morocco, Sep. 2013.

The problem of time delay estimation (TDE) has many approaches and a large field of applications, making it an important research issue. For specific air traffic control systems, time delay estimation is a primary step for speech enhancement. In this paper we introduce two new TDE methods, based on cross power-spectrum phase (CSP), combining previous efficient methods that use accumulating cross power spectrum, whitening and coherence. We show that the proposed methods bring an accuracy improvement of more than 5%, while being 5% to 20% faster.

3.[Marinescu, 2013c] **Marinescu, R.S.**, Buzo, A., Cucu, H., Burileanu, C., "Extensive Evaluation Experiments for the Accumulated Cross-Power Spectrum Methods for Time Delay Estimation", *7th International Conference on Speech Technology and Human-Computer Dialogue (SpeD 2013)*, Cluj-Napoca – Romania, Oct. 2013.

In numerous real time signal processing applications, time delay estimation (TDE) is still a hot topic. A recently proposed solution, based on Generalized Cross Correlation (GCC) method, is evaluated further in this paper. Experimental results show that among traditional variants of GCC methods the *accumulated ρ-cross power spectrum phase* proposed by us is the most accurate. Another important aspect is that the computing time of this method is comparable with those of other popular GCC methods.

4.[Marinescu, 2013d] **Marinescu, R.S**., Buzo, A., Cucu, H., Burileanu, C., "Applying the Accumulation of Cross-Power Spectrum Technique for Traditional Generalized Cross-Correlation Time Delay Estimation", *International Journal On Advances in Telecommunications – IARIA*, accepted invited paper, Sep. 2013.

In many real time applications, time delay estimation requires a special solution. Despite the various approaches which were proposed over the years, the topic remains hot for digital signal processing because of its large field of applications and implementation forms. Among different classes of methods for this issue, general cross-correlation method is wildly used. It offers good results and does not need an adaptation time, like those based on adaptive filtering. In this paper, we make a survey and compare the most popular generalized cross-correlation methods. We extend the analysis, by applying the accumulation of cross-power spectrum technique, for all well known generalized cross-correlation methods. The comparisons are provided by detailed numerical and simulation analysis, using several metrics. Based on the accuracy rate, error rate, standard deviation of relative error and computing time we provide new considerations for traditional generalized cross-correlation methods.

5.**Marinescu, R.S**., Burileanu, C., "Voice Activity Detection for Best Signal Selection in Air Traffic Management And Control Systems", *ICASSP 2014*, submitted paper

The Best Signal Selection (BSS) in air traffic management and control systems has to decide among several signal instances of the same source which one offers the highest speech intelligibility. In these systems, the source signal is not available, thus, objective speech quality tests could not be used. However, information with regards to the speech quality could be obtained from the score of voice activity detection (VAD) algorithms. In this paper the correlation between speech quality and the score of VAD algorithms is analyzed. The results showed that the VAD score-based methods do not saturate for higher SNR, as the Perceptual Evaluation of Speech Quality (PESQ) does. A new VAD algorithm as a solution for the best signal selection problem is also proposed.

References

[Aurora] Aurora database, http://www.elda.org/article52.html [retrieved: Aug, 2013]

[Barsanti, 2001] Barsanti, R.J., "Passive Target Tracking with Uncertain Sensor Positions using Wavelet-Based Transient Signal Processing", Ph.D. Dissertation, Naval Postgraduate School, Monterey, California, Jun. 2001.

[Barsanti, 2003] Barsanti, R.J., Tummala, M., "Wavelet-based time delay estimates for transient signals", Conference Record of the 37th Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 1173 - 1177, Nov. 2003.

[Bedard, 1994] Bedard, S., Champagne, B., Stephenne, A., "Effects of room reverberation on time-delay estimation performance", In Proceedings of the IEEE ICASSP, Adelaide, Australia, pp. II-261–II-264, 1994.

[Bellanger, 1989] Bellanger, M., "Analyse des signaux et filtrage numerique adaptatif", Masson et CNET-ENST, Paris, 1989.

[Benesty, 2000], Benesty, J., "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization", J. Acoust. Soc. Am. Volume 107, Issue 1, pp. 384-391, Jan. 2000.

[Benesty, 2006] Benesty, J., Rey, H., Rey Vega, L., Tressens, S.,"A non-parametric VSS NLMS algorithm," IEEE Signal Processing Lett., vol. 13, pp. 581–584, Oct. 2006.

[Beritelli, 1998] Beritelli, F., Casale, S., Cavallaro, A., "A robust voice activity detector for wireless communications using soft computing," IEEE Journal on Selected areas in Communications (JSAC), vol. 16, no. 9, pp. 1818–1829, Dec 1998.

[Bies, 2003] Bies, D., "Engineering Noise Control: Theory and Practice", Taylor & Francis, New York – USA, 2003.

[Boll, 1979] Boll, S.F., "Suppression of acoustic noise in speech using spectral subtraction", IEEE Trans. Acoustics, Speech, and Signal Process., vol.27, pp.113-120, Feb. 1979.

[Carter, 1973] Carter, G.C., Nutall, A.H., Cable, P.G., "The Smoothed Coherence Transform", Proc. IEEE (Lett.), vol. 61, pp 1497-1498, Oct 1973.

[Champagne, 1996] Champagne, B., Bedard, S., Stephenne, A., "Performance of time-delay estimation in the presence of room reverberation", IEEE Trans. Speech Audio Process. 4, pp. 148–152, 1996.

[Chatlani, 2010] Chatlani, N., Soraghan, J.J., "LOCAL BINARY PATTERNS FOR 1-D SIG AL PROCESSING", 18th European Signal Processing Conference (EUSIPCO-2010), Denmark, pp. 95-99, Aug. 2010.

[Chen, 2012] Chen, F., Loizou, P.C., "Impact of SNR and gain-function over- and under-estimation on speech intelligibility", Elsevier Speech Communication, no.54, pp. 272–281, 2012.

[Cohen, 2001] Cohen, I., Berdugo, B., “Speech enhancement for non-stationary noise environments”, ELSEVIER Signal Process., vol. 81, no. 11, pp. 2403–2418, Nov. 2001.

[Cohen, 2003] Cohen, I., “Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging”, IEEE Trans. Speech Audio Process., vol. 11, no. 5, pp. 466–475, Sep. 2003.

[Cornu, 2003] Cornu, E., Sheikhzadeh, H., Brennan, R.L., Abutalebi, H.R., “ETSI AMR-2 VAD: evaluation and ultra low-resource implementation”, Proceedings of International Conference on Multimedia and Expo 2003, ICME '03, vol. 2, pp. II - 841-4, 2003.

[Craciun, 2004] Craciun, A., Gabrea, M., “Correlation coefficient-based voice activity detector algorithm”, Canadian Conference on Electrical and Computer Engineering, vol. 3, pp. 1789-1792, May 2004.

[Davis, 2006] Davis, A., Nordholm, S., Togneri, R., “Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold”, IEEE Trans. on Audio, Speech and Language Proc., vol. 14, no. 2, pp 412-424, Mar 2006.

[Dohnal, 1995] Dohnal, F., “Generalized Frequency Domain LMS Adaptive Filter”, In Proc. of Radioengineering, vol.4, issue 2, 1995.

[Dyba, 2008] Dyba, R.A., “Parallel Structures for Fast Estimation of Echo Path Pure Delay and Their Applications to Sparse Echo Cancellers”, In Proceedings of CISS 2008, 42nd Annual Conference on Information Sciences and Systems, article ID 10.1109/CISS.2008.4558529, pp. 241-245, Mar. 2008.

[Eckart, 1952] Eckart, C., “Optimal Rectifier System for Detection of Steady Signals”, Univ. California, Scripps Inst. Oceanography, Marine Physical Lab. Rep SIO 12692, SIO Ref 52-11, 1952.

[ED-136] European Organization for Civil Aviation Equipment, “Voice over Internet Protocol (VoIP) Air Traffic Management (ATM) System Operational and Technical Requirements”, Mar 2009.

[ED-137B] European Organization for Civil Aviation Equipment, “Interoperability Standards for VoIP ATM Components (Part 1: Radio - Part 2: Telephone - Part 3: European Legacy Telephone Interworking - Part 4: Recording - Part 5: Supervision)”, Jan 2012.

[ED-138] European Organization for Civil Aviation Equipment, “Network Requirements and Performances for Voice over Internet Protocol (VoIP) Air Traffic Management (ATM) Systems (Part 1: Network Specification – Part 2: Network Design Guideline)”, Feb 2009.

[Emadzadeh, 2008] Emadzadeh, A.A., Lopes, C.G., Speyer, J.L., “Online time delay estimation of pulsar signals for relative navigation using adaptive filters”, In Proc. of 2008 IEEE/ION, Position, Location and Navigation Symposium, article ID 10.1109/PLANS.2008.4570029, pp. 714–719, 2008.

[Ephraim, 1984] Ephraim, Y., Malah, D., “Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator”, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-32, pp. 1109–1121, 1984.

[Ephraim, 1985] Ephraim, Y., Malah, D., “peech enhancement using a minimum mean-square error log-spectral amplitude estimator”, IEEE Transactions on Acoustics, Speech and Signal Processing, no. 33, pp. 443–445, 1985.

[ETSI-AMR] European Telecommunications Standards Institute, “Adaptive Multi-Rate (AMR) speech codec; Voice Activity Detector (VAD)”, 3GPP TS 26.094 version 10.0.0 Release 10, 2011.

[Evangelopoulos, 2005] Evangelopoulos, G., Maragos, P., “Speech event detection using multiband modulation energy”, Proc. Interspeech 2005, Portugal, pp 685-688, Sep. 2005.

[Farsi, 2008] Farsi, H., Mozaffarian, M.A., Rahmani, H., “A Novel Method to Modify VAD used in ITU-T G.729B for Low SNRs”, International Journal of Computers and Communications, vol. 2, pp. 20-29, 2008.

[Frost, 1972] Frost III, O. L., “An algorithm for linearly constrained adaptive array processing”, Proc. IEEE vol. 60, pp. 926–935, 1972.

[Fujimoto, 2008] Fujimoto, M., Ishizuka, K., Nakatani, T., “A voice activity detection based on the adaptive integration of multiple speech features and a signal decision scheme”, IEEE International Conference on Acoustics, Speech and Signal Processing 2008, pp. 4441-4444, USA, Mar. 2008.

[Gauss, 1821] Gauss, F.C.J., “Theoria combinationis observationum erroribus minimis obnoxiae: Pars prior”, Gottingische gelehrte Anzeigen, 33:312:327, 1821.

[Germain, 2013] Germain, F.G., Sun, D.L., Mysore, G.J., “Speaker and Noise Independent Voice Activity Detection”, Interspeech, France, Aug. 2013.

[Gerkmann, 2012] Gerkmann, T., Hendriks, R.C., “Unbiased MMSE-based Noise Power Estimation with Low Complexity and Low Tracking Delay”, Journal Trans. Audio, Speech and Language Processing, vol. 20, no. 4, pp. 1383 – 1393, Dec 2012.

[Gerven, 1997] Gerven, S.V., Xie, F., “A comparative study of speech detection methods,” In Proceedings of Eurospeech, pp. 1095-1098, 1997.

[Ghosh, 2011] Ghosh, P.K., Tsiartas, A., Narayanan, S., “Robust Voice Activity Detection Using Long-Term Signal Variability”, IEEE Transactions on Audio, Speech, and Language Processing, ISSN :1558-7916, pp. 600-613, Mar. 2011.

[Gradshteyn, 2007] Gradshteyn, I.S., Ryzhik, I.M., “Table of Integrals Series and Products”, 7th ed. San Diego, CA, USA Elsevier Academic Press, 2007.

[Haigh, 1993] Haigh, J., Mason, J.S., “A voice activity detector based on cepstral analysis”, Proc. 3rd EUROSPEECH, Berlin Germany, pp 1103-1106, Sep. 1993.

[Hannan, 1971] Hannan, E.J., Thomson, P.J., “The Estimation of Coherence and Group Delay”, Biometrika, vol. 58, pp. 469-481, Dec 1971.

[Hao, 2009] J. Hao, H. Attias, S. Nagarajan, and T.W. Lee, “Speech enhancement, gain, and noise spectrum adaptation using approximate Bayesian estimation,” IEEE Trans. Audio Speech Lang. Process., vol.17, pp.24-37, Jan. 2009.

[Harding, 2012] Harding, P., Milner, B., “On the use of Machine Learning Methods for Speech and Voicing Classification”, In Proceedings of Interspeech, 2012.

[Hassab, 1979] Hassab, J.C., Boucher, R.E., “Optimum Estimation of Time Delay by Generalized Correlator”, IEEE Transaction on Acoustics, Speech, and Signal Processing, vol. assp-27, no. 3, pp. 373-380, Aug. 1979.

[Hassab, 1980] Hassab, J.C., Boucher, R.E., “A Cuantitative Study of Optimum and Suboptimum filters in Generalized Correlator”, IEEE 1979 Int. Conf. Acoust., Speech, Signal Processing Conf. Rec. 79CH 1379-7 ASSP, pp. 124-127, 1980.

[Hassab, 1981] Hassab, J.C., Boucher, R.E., “Performance of the Generalized Cross Correlator in the Presence of a Strong Spectral Peak in the Signal”, IEEE Transaction on Acoustics, Speech, and Signal Processing, vol. assp-29, no. 3, pp. 549-555, Jun. 1981.

[Hendriks, 2008] Hendriks, R.C., Jensen, J., Heusdens, R., “Noise tracking using DFT domain subspace decompositions”, IEEE Trans. Audio, Speech, Language Process., vol. 16, no. 3, pp. 541–553, Mar. 2008.

[Hendriks, 2010] Hendriks, R.C., Jensen, J., Heusdens, R., “MMSE based noise PSD tracking with low complexity”, in IEEE Int. Conf. Acoust., Speech, Signal Process., Dallas, TX, USA, pp. 4266–4269, Mar. 2010.

[Hero, 1985] Hero, A.O., Schwartz, S.C., “A New Generalized Cross Correlator”, IEEE Transactions, Acoustics, Speech and Signal Processing, vol. 33, pp. 38-45, Feb 1985.

[Hongyang, 2008] Hongyang, D., Dyba, R.A., “Efficient Partial Update Algorithm Based on Coefficient Block for Sparse Impulse Response Identification”, In Proceedings of CISS 2008, 42nd Annual Conference on Information Sciences and Systems, article ID 10.1109/CISS.2008.4558527, pp. 233-236, Mar. 2008.

[Hongyang, 2009] Hongyang, D., Dyba, R.A., , "Partial Update PNLMS Algorithm for Network Echo Cancellation", In Proceedings of ICASSP 2009, IEEE International Conference on Acoustics, Speech and Signal Processing, article ID 10.1109/ICASSP.2009.4959837, pp. 1329-1332, Apr. 2009.

[Ikuta, 2011] Ikuta,A., Orimoto, H., Xiao, O., "A Bayesian Approach for Noise Suppression of Speech Signal in Real Environment", 19th European Signal Processing Conference (EUSIPCO 2011), Spain, pp. 225 – 229, Sep. 2011.

[Iqbal, 2008] Iqbal, M. A., Grant, S. L., "Novel variable step size NLMS algorithm for echo cancellation", In Proceedings of ICASSP 2008, IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 241–244, 2008.

[Ishizuka, 2010] Ishizuka, K., Nakatani, T., Fujimoto, M., Miyazaki, N., "Noise robust voice activity detection based on periodic to aperiodic component ratio", Journal of Speech Communication, vol. 52, pp. 41-60, Jan. 2010.

[ITU-T G729b], ITU-T "A silence compression scheme for G.729 optimized for terminals conforming to recommendation V.70," ITU-T Recommendation G.729-Annex B, 1996.

[Jabloun, 1999] Jabloun, F., Cetin, A.E., Erzin, E., "Teager energy based feature parameters for speech recognition in car noise," IEEE Signal Processing Letters, 6(10), pp.259-261, 1999.

[Jacovitti, 1987] Jacovitti, G., Cusani, R., "An Efficient Technique for High Correlation Estimation", IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. ASSP-35, no. 5, pp. 654-660, May 1987.

[Jacovitti, 1993] Jacovitti, G., Scarano, G., "Discrete time techniques for Time delay estimation", IEEE Transactions on Signal Processing, Vol. 41, Issue: 2, pp. 525 – 533, Feb. 1993.

[Jiang, 2010] Jiang, W., Lo, W. K., Meng, H., "A new voice activity detection method using maximized Sub-band SNR", International Conference on Audio Language and Image Processing, pp. 80–84, 2010.

[Jinhong, 2008] Jinhong, W., Doroslovacki, M. "Partial Update NLMS Algorithm for Sparse System Identification with Switching Between Coefficient-based and Input-based Selection", In Proc. of CISS 2008, 42nd Annual Conference on Information Sciences and Systems, article ID 10.1109/CISS.2008.4558528, pp. 237–240, 2008.

[Khong, 2006] Khong, A.W.H., Naylor, P.A., "Efficient Use Of Sparse Adaptive Filters", In Proceedings of ACSSC '06, Fortieth Asilomar Conference on Signals, Systems and Computers, article ID 10.1109/ACSSC.2006.354982, pp. 1375-1379, Nov. 2006.

[Knapp, 1976] Knapp, C., Carter, G.C., "The Generalized Correlation Method for Estimation of Time Delay", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 24, issue 4, pp. 320-327, Aug. 1976.

[Kotnik, 2001] Kotnik, B., Kacic, Z., Horvat, B., "A multiconditional robust front-end feature extraction with a noise reduction procedure based on improved spectral subtraction algorithm", Proc. 7th EUROSPEECH, Denmark, pp. 197-200, Sep 2001.

[Lee, 2006] Lee, Y.C., Ahn, S.S., "Statistical model-based VAD algorithm with wavelet transform", IEICE Trans. Fundamentals, vol. E89-A, no. 6, pp 1594-1600, Jun 2006.

[Lin, 1998] Lin, S.N., Chern, S.J., "A new adaptive constrained LMS time delay estimation algorithm", Signal Processing, Volume 71, Issue 1, pp. 29-44, Nov. 1998.

[Lotter, 2003] Lotter, T., Benien, C., Vary, Peter., "Multichannel Direction-Independent Speech Enhancement Using Spectral Amplitude Estimation", EURASIP Journal on Applied Signal Processing 2003, pp. 1147–1156, 2003.

[Lun, 2010] Lun, D.P.K., Hsung, T.C. "Improved Wavelet Based A-priori SNR Estimation for Speech Enhancement", Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 2382-2385, 2010.

[Ma, 2013] Ma, Y., Nishihara, A., "Efficient voice activity detection algorithm using long-term spectral flatness measure", EURASIP Journal on Audio, Speech, and Music Processing, ISSN 1687-4722, Jan. 2013.

[Mallat, 1989] Mallat, S., "A theory for multiresolution signal decomposition: the wavelet representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7), pp. 674-693, 1989.

[Marinescu, 2013a] Marinescu, R.S., Buzo, A., Cucu, H., Burileanu, C,. "New Considerations for Accumulated ρ-Cross Power Spectrum Phase with Coherence Time Delay Estimation", In Procedings of ICDT 2013, The Eight International Conference of Digital Telecommunications, Venice-Italy, pp. 55-59, Apr. 2013.

[Marinescu, 2013b] Marinescu, R.S., Buzo, A., Cucu, H., Burileanu, C,. "Fast Accurate Time Delay Estimation Based on Enhanced Accumulated Cross-Power Spectrum Phase", 21st European Signal Processing Conference, Marrakesh-Morocco, Sep. 2013.

[Marinescu, 2013c] Marinescu, R.S., Buzo, A., Cucu, H., Burileanu, C., "Extensive Evaluation Experiments for the Accumulated Cross-Power Spectrum Methods for Time Delay Estimation", 7th International Conference on Speech Technology and Human-Computer Dialogue (SpeD 2013), Romania, Oct. 2013.

[Marinescu, 2013d] Marinescu, R.S., Buzo, A., Cucu, H., Burileanu, C., "Applying the Accumulation of Cross-Power Spectrum Technique for Traditional Generalized Cross-Correlation Time Delay Estimation", International Journal On Advances in Telecommunications – IARIA, submitted invited paper, sep. 2013.

[Mark, 2002] Mark, M., Birger, K., "Speech pause detection for noise spectrum estimation by trackingpower envelope dynamics," IEEE Trans. Speech Audio Proc., vol.10, pp.109-118, 2002.

[Martin, 2001] Martin, R., "Noise power spectral density estimation based on optimal smoothing and minimum statistics", IEEE Trans. Speech Audio Process., vol. 9, no. 5, pp. 504–512, Jul. 2001.

[Matassoni, 2006] Matassoni, M., Svaizer, P., "Efficient Time Delay Estimation Based on Cross-Power Spectrum Phase", European Signal Processing Conference (EUSIPCO), Florence - Italy, Sep 2006.

[McAulay, 1980] McAulay, R.J., Malpass, M.L., "Speech enhancement using a soft-decision noise suppression filter", IEEE Trans. Acoust., Speech, Signal Process., vol. 28, no. 2, pp. 137–145, Apr. 1980.

[McCowan, 2001] McCowan, I.A., "Robust Speech Recognition using Microphone Arrays", Queensland University of Technology, 2001.

[Moattar, 2009] Moattar, M.H., Homayounpour, M.M., "A Simple But Efficient Real-Time Voice Activity Detection Algorithm", 17th European Signal Processing Conference (EUSIPCO 2009), Scotland, pp. 2549-2553, Aug. 2009.

[Narayanan, 2012] Narayanan, A., Wang, D., "A CASA-Based System for Long-Term SNR Estimation", IEEE Transactions On Audio, Speech, and Language Processing, vol. 20, no. 9, pp. 2518-2527, Nov. 2012.

[Noizeus]: A noisy speech corpus http://www.utdallas.edu/~loizou/speech/noizeus/ (retrieved: Aug, 2013).

[Omar, 2012] Omar, M.K., "Speech Activity Detection for Noisy Data using Adaptation Techniques", In Proceedings of Interspeech, 2012.

[Omologo, 1994] Omologo, M., Svaizer, P., "Acoustic event localization using a cross-power spectrum phase based technique", Proceedings of ICASSP, Australia, pp. 273-276, Apr. 1994.

[Omologo, 1997] Omologo, M., Svaizer, P., "Use of the crosspower-spectrum phase in acoustic event location", IEEE Transactions on Speech Audio Process, pp. 288-292, May 1997.

[Ouzounov, 2004] Ouzounov, A., “Robust Feature for Speech Detection”, Cybernetics and Information Technologies, vol.4, No.2, pp.3-14, 2004.

[Owsley, 1978] Owsley, N. L., “Adaptive data orthogonalization”, in Proceedings of ICASSP 1978, IEEE International Conference on Acoustics Speech and Signal Processing , pp. 109–112, 1978.

[Paleologu, 2010] Paleologu, C., Benesty, J., Ciochina, S., “An Improved Proportionate NLMS Algorithm Based on the l0 Norm”, In Proc. Of ICASSP ‘10, 2010 IEEE International Conference on Acoustics Speech and Signal Processing, article ID10.1109/ICASSP.2010 5495903, pp. 309–312, 2010.

[Prabhakar, 2005] Prabhakar, G., Rastogi, R., Thotton, M., "OSS Architecture & Requirements for VoIP Networks", Bell Labs Technical Journal 10 (1), pp. 31–45, 2005.

[Pwint, 2005] Pwint, M., Sattar, F., “A new speech/non-speech classification method using minimal Walsh basis functions”, IEEE International Symposium on Circuits and Systems, vol. 3, pp. 2863-2866, May 2005.

[Rabiner, 1977] Rabiner, L.R., Sambur, M.R., “Application of an LPC distance measure to the voiced-unvoiced-silence detection problem,” IEEE Trans. Acous., Speech, Signal Proc., vol. 25- issue 4, pp. 338 – 343, 1977.

[Rabinkin, 1996] Rabinkin, D.V., Renomeron, R.J., Dahl, A., French, J.C., Flanagan J.L., Bianchi, M.H., “A DSP Implementation of Source Location Using Microphone Arrays”, The Journal of the Acoustical Society of America, Volume 99, Issue 4, pp. 2510-2527, Apr 1996.

[Ramirez, 2004] Ramirez, J., Segura, J.C., Benitez, C., Torre, A., Rubio, A., “Efficient voice activity detection algorithms using long-term speech information”, Speech Communication, vol. 42, issues 3-4, pp 271-287, 2004.

[Rangachari, 2006] Rangachari, S., Loizou, P.C., “A noise-estimation algorithm for highly non-stationary environments”, ELSEVIER Speech Commun., vol. 48, no. 2, pp. 220–231, 2006.

[Ross, 1974] Ross, M., Shaffer, H., Cohen, A., Freudberg, R., Manley, H., “Average Magnitude Difference Function Pitch extractor”, IEEE Transactions on Acoustics, Speech and Signal Processing, Volume: 22 , Issue: 5, pp. 353 – 362, 1974.

[Roth, 1971] Roth, P.R., “Effective measurements using digital signal analysis”, IEEE Spectrum, vol. 8, pp. 62-70, Apr 1971.

[Sakhnov, 2011a] Sakhnov, K., Verteletskaya, E., Simak, B., “Partial Update Algorithms and Echo Delay Estimation”, Communications – Scientific Journal of the University of Zilina, Zilina – Slovakia, vol. 13, no. 2, pp. 14-19, Apr. 2011.

[Sakhnov, 2011b] Sakhnov, K., Verteletskaya, E., Simak, B., “Echo Delay Estimation Using Algorithms Based on Cross-correlation”, Journal of Convergence Information Technology, Volume 6, Number 4, pp. 1 – 11, Apr. 2011.

[Sakhnov, 2011c] Sakhnov, K., Verteletskaya, E., Simak, B., “Adaptive Filtering Applications – Ch.4 Perceptual Echo Control and Delay Estimation”, ISBN 978-953-307-306-4, July 2011.

[Sanchez, 2013] Sanchez, B.P., “On-Demand Secure Teleconferencing on Public Cloud Infrastructures”, Universidad Complutense de Madrid, Master Thesys, http://eprints.ucm.es/22649/1/OnDemand_Secure_Teleconferencing_on_Public_Cloud_Architectures_Bernardo_Pericacho_S%C3%A1nchez.pdf, 2013.

[Seo, 2007] Seo, N., “Individual Voice Activity Detection Using Periodic to Aperiodic Component Ratio Based Activity Detection (Parade) and Gaussian Mixture Speaker Models”, University of Maryland, Final Project, 2007.

[Shean, 2009] Shean, M., Liu H., “A Modified Cross Power-Spectrum Phase Method Based on Microphone Array for Acoustic Source Localization,” IEEE International Conference on System, Man and Cybernetics, San Antonio, TX, USA, pp. 1286 – 1291, Oct. 2009.

[Shuyin, 2009] Shuyin, Z., Ying, G., Buhong, W., "Auto-correlation property of speech and its application in voice activity detection", First International Workshop on Education Technology and Computer Science (ETCS '09), vol. 3 (IEEE, Piscataway), pp. 265–268, 2009.

[So, 2001] So, H.C., Ching, P.C., "Comparative study of five LMS-based adaptive time delay estimators", IEE Proceedings - Radar, Sonar and Navigation, Volume 148, Issue 1, pp. 9 – 15, February 2001.

[Sohn, 1998] Sohn, J., Sung, W., "A voice activity detector employing soft decision based noise spectrum adaptation", in IEEE Int. Conf. Acoust., Speech, Signal Process., vol. 1, Seattle, WA, USA, pp. 365 – 368, May 1998.

[Soleimani, 2008] Soleimani, S.A., Ahadi, S.M., "Voice Activity Detection based on Combination of Multiple Features using Linear/Kernel Discriminant Analyses", 3rd International Conference on Information and Communication Technologies: From Theory to Applications, pp. 1-5, Apr 2008.

[Sun, 2010] Sun, Y., Qiu, T., "The SCOT Weighted Adaptive Time Delay Estimation Algorithm Based on Minimum Dispersion Criterion", In Proceedings of the ICICIP Conference on Intelligent Control and Information Processing, pp. 35-38, Aug. 2010.

[Taghia, 2011] Taghia, J., Taghia J., Mohammadiha, N., Sang, J., Bouse, V., Martin, R., "An evaluation of noise power spectral density estimation algorithms in adverse acoustic environments", in IEEE Int. Conf. Acoust., Speech, Signal Process., Dallas, TX, USA, pp. 4640-4643, May 2011.

[Tanabe, 2008] Tanabe, N., Furukawa, T., Tsuji, S., "Robust noise suppression algorithm with the Kalman filter theory for white and colored disturbance", IEICE Trans. Fundamentals, vol.E91-A, pp.818-829, Mar. 2008.

[Tianshuang, 1996] Tianshuang, Q., Hongyu, W., "An Eckart-weighted adaptive time delay estimation method", IEEE Transactions on Signal Processing, vol. 44, issue 9, pp. 2332-2335, Sep. 1996.

[Tong, 1993] Tong, L., Xu, G., Kailath, T., "Fast blind equalization via antenna arrays", In Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP1993), USA, Vol.IV, pp. 272–275, 1993.

[Trawicki, 2012] Trawicki, M.B., Johnson, M.T., "Distributed multichannel speech enhancement with minimum mean-square error short-time spectral amplitude, log-spectral amplitude, and spectral phase estimation", Elsevier Signal Processing, no. 92, pp 345-356, 2012.

[Udrea, 2010] Udrea, R.M., Vizireanu, D.N., Oprea, C.C., Pirnog, I., "A Time-Recursive Adaptive Algorithm for Colored Noise Reduction in Speech Enhancement", Sixth Advanced International Conference on Telecommunications, pp. 187 – 190, 2010.

[Vaseghi, 2006] Vaseghi, S.V., "Advanced Digital Signal Processing and Noise Reduction -Third Edition", John Wiley and Sons, England, ISBN: 978-0-470-09495-2, 2006.

[Widrow, 1959] Widrow, B., "Adaptive Sampled-Data Systems --- A Statistical Theory of Adaptation", IRE WESCON Convention Record, 4:74-85, 1959.

[Widrow, 1960a] Widrow, B., "Adaptive Sampled-data Systems", Proceedings of the First International Congress of the International Federation of Automatic Control, pp.406-411, 1960.

[Widrow, 1960b] Widrow, B., Hoff, M.E.Jr., "Adaptive Switching Circuits" IRE WESCON Convention Record, 4:96-104, August 1960.

[Widrow, 1961] Widrow, B., "The Speed of Adaption in Adaptive Control Systems", American Rocket Society (ARS) Guidance, Control and Navigation Conference Proceedings, 1933-61, August 1961.

[Widrow, 1985] Widrow, B., Stearns, S.D., "Adaptive signal processing", Penitence-Hall, ISBN 0130040290, USA, 1985.

[Wilson, 2006] Wilson, K.W., Darrell, T., “Learning a Precedence Effect-Like Weighting Function for the Generalized Cross-Correlation Framework”, IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, issue 6, pp. 2156-2164, Nov. 2006.

[Wu, 1997] Wu, S., So, H., Ching, P., “Improvement of TDOA Measurements Using Wavelet Denoising with a Novel Thresholding Technique”, IEEE Transactions on Acoustics, Speech, and Signal Processing, pp. 539 – 542, Apr. 1997.

[Wu, 2006] Wu, B.F., Wang, K.C., “Voice Activity Detection Based on Auto-Correlation Function Using Wavelet Transform and Teager Energy Operator”, Computational Linguistics and Chinese Language Processing, vol. 11, No. 1, pp. 87-100, Mar 2006.

[Ying, 2011] Ying, D., Yan, Y., Dang, J., Soong, F.K., “Voice Activity Detection Based on an Unsupervised Learning Framework” IEEE Transactions on Audio, Speech, and Language Processing 19(8), 2011.

[Youn, 1983] Youn, D.H., Ahmed, N., Carter, G.C., “On the Roth and SCOTH Algorithms: Time-Domain Implementations”, In Proceedings of the IEEE, vol. 71, issue 4, pp. 536-538, 1983.

[Zetterberg, 2005] Zetterberg, V., Pettersson, M.I., Claesson, I., “Comparison Between Whitened Generalized Crosscorrelation and Adaptive Filter for Time Delay Estimation”, In Proceedings of TS/IEEE, OCEANS, vol. 3, article ID 10.1109/OCEANS.2005.1640117, pp. 2356 – 2361, Sep. 2005.