UNIVERSIDAD DE JAÉN

ESCUELA POLITÉCNICA SUPERIOR DE LINARES
DEPARTAMENTO DE INGENIERÍA DE TELECOMUNICACIÓN

TESIS DOCTORAL

CLASSIFICATION AND SEPARATION TECHNIQUES BASED ON FUNDAMENTAL FREQUENCY FOR SPEECH ENHANCEMENT

PRESENTADA POR:
PABLO ANTONIO CABAÑAS MOLERO

DIRIGIDA POR:
DR. D. PEDRO VERA CANDEAS
DR. D. NICOLÁS RUIZ REYES

JAÉN, 11 DE ENERO DE 2016

ISBN 978-84-16819-29-4

Classification and Separation Techniques based on Fundamental Frequency for Speech Enhancement

Ph.D. Thesis

PABLO ANTONIO CABAÑAS MOLERO

January 2016

Dept. of Telecommunication Engineering
University of Jaén
Linares, Jaén, Spain

This work was carried out under the supervision of

Dr. Pedro Vera Candeas
Department of Telecommunication Engineering
University of Jaén
Linares, Jaén, Spain

Dr. Nicolás Ruiz Reyes
Department of Telecommunication Engineering
University of Jaén
Linares, Jaén, Spain

Abstract

In the last decades, the fast development of digital signal processing technology has generated a growing interest on applications based on audio signals. Speech classification and separation algorithms are essential in this context because, in practical situations, the speech signal is degraded by background noise and other interferences, which is a serious obstacle for certain applications. In this thesis, we focus on the development of new classification and speech enhancement algorithms based, explicitly or implicitly, on the fundamental frequency (F0). The F0 of speech has a number of properties that enable speech discrimination from the remaining signals in the acoustic scene, either by defining F0-based signal features (for classification) or F0-based signal models (for separation).

In this thesis, we work in two different application scenarios. In the first one, it is assumed that the algorithms must be implemented on digital hearing aids which impose strong constrains in terms of computational capacity. We develop an acoustic environment classification algorithm based on F0 to classify the input signal into speech and nonspeech classes. The proposed pitch estimator is able to determine the F0 of the input signal with low computational cost, while maintaining a high speech/nonspeech classification accuracy. In addition, we also focus on the problem of classifying the input signal on a frame-by-frame basis for voiced speech detection, with the aim of developing a low-complexity speech enhancement algorithm based on F0.

In a second scenario, we assume that the aforementioned limitations no longer apply. In this case, we address the problem of speech and noise separation using compositional models and source-specific mathematical constrains. In recent works, compositional modeling of audio with matrix decomposition algorithms, most of them derived from *nonnegative matrix factorization* (NMF), has obtained very good results in audio source separation, specially in music signals. Although its application to speech has not been as explored as in music, the great potential of these techniques is very promising. The proposed signal model, in conjunction with the developed regularized NMF, obtains better separation measures for speech and noise separation than other NMF-based methods without the proposed restrictions.

Resumen

Durante las últimas décadas, el desarrollo de las tecnologías de procesado digital de señal ha despertado un interés cada vez mayor por las aplicaciones basadas en señales de audio. Los algoritmos de clasificación y separación de voz resultan esenciales en este contexto, puesto que, en situaciones prácticas, la señal vocal se encuentra afectada por ruido de fondo u otras interferencias, lo cual dificulta o impide ciertas aplicaciones. Con esta tesis se pretende dar lugar a nuevos algoritmos de clasificación y mejora de voz basados, explícita o implícitamente, en la frecuencia fundamental (F0). La F0 de la voz posee unas propiedades que permiten la discriminación de esta fuente respecto al resto de señales de la escena acústica, ya sea mediante la definición de características (para clasificación) o la definición de modelos de señal (para separación).

En esta tesis se abordan dos escenarios de trabajo. En el primero, se supone que los algoritmos deben implementarse en audífonos digitales, con las enormes restricciones en capacidad computacional que ello conlleva. Se desarrolla un algoritmo de clasificación de entorno acústico basado en F0 capaz de clasificar la señal en las clases voz y no-voz. El algoritmo propuesto consigue estimar la F0 de la señal utilizando un mínimo coste de recursos, sin alterar gravemente los resultados de la clasificación. En segundo lugar, se aborda también el problema de clasificar la señal de entrada trama a trama para la detección de voz sonora, con objeto de desarrollar un algoritmo de mejora de voz de bajo coste basado en F0.

En un segundo escenario, se supone que no existen restricciones graves de capacidad computacional. En este caso, se aborda el problema de la separación de voz y ruido mediante el uso de modelos basados en patrones, con restricciones específicas para voz y ruido. En la literatura reciente, modelos basados en patrones junto con algoritmos de descomposición de matrices, casi todos basados en *nonnegative matrix factorization* (NMF), han obtenido muy buenos resultados en separación de fuentes sonoras, sobre todo en música. Aunque su aplicación a señales de voz ha sido algo menos explorada, las posibilidades de estas técnicas resultan prometedoras. El modelo propuesto, junto con un algoritmo de descomposición NMF con restricciones, permite separar voz y ruido obteniendo mejores resultados que otros modelos basados en patrones sin las restricciones propuestas.

Preface

This PhD thesis has been carried out at the Department of Telecommunication Engineering, University of Jaén, and supported by the Andalusian Business, Science and Innovation Council under project 2010-TIC6762.

First, I wish to express my gratitude to Dr. Nicolás Ruiz Reyes, the person who encouraged me to start my research career on signal processing, and whose support, advice and expertise have been very valuable for me throughout these years. I also want to thank Dr. Pedro Vera Candeas for his excellent guidance, research attitude and kind cooperation during the course of my research work. Without his wide technical knowledge and ideas this thesis would not have been possible.

I am grateful to the members of the research group TIC-188, especially to Dr. Damián Martínez Muñoz and Dr. Francisco Jesús Cañadas Quesada for their numerous collaborations, and for giving me help and support when required. Also, I owe thanks to the people of the Multispeech group at INRIA Nancy, in particular Dr. Juan A. Morales Cordovilla, for the warm welcome and for giving me the opportunity to live an enriching experience.

Of course, I want to thank all the guys from the lab who have made my working environment so friendly and enjoyable, including Julio, Antonio, Rocío, Francisco, David, Juan, Amparo, Pedro, José Guadalupe, Salah, Piedad, Diego, Casto, Violeta and Unai. A special mention belongs to Dr. Julio Carabias Orti and Dr. Francisco J. Rodríguez Serrano, whose simultaneous work has been very helpful, and who have been always around to share knowledge and code.

Finally, my deepest gratitude goes to my family, for their love and unconditional support.

Pablo Antonio Cabañas Molero
January 2016

Contents

List of Included Publications

This thesis is a compound thesis consisting of the following 3 publications, which are preceded by an introductory overview of the research field and a summary of the contributions. The publications are referred in the text with [P1], [P2] and [P3].

P1 **P. Cabañas-Molero, N. Ruiz-Reyes, P. Vera-Candeas, and S. Maldonado-Bascon**, "Low-complexity F0-based speech/nonspeech discrimination approach for digital hearing aids", in *Multimedia Tools and Applications*, Volume 54, Issue 2, August 2011, pp. 291–319.

P2 **P. Cabañas-Molero, D. Martínez-Muñoz, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Rodríguez-Serrano**, "Voicing detection based on adaptive aperiodicity thresholding for speech enhancement in non-stationary noise", in *IET Signal Processing*, Volume 8, Issue 2, April 2014, pp. 119–130.

P3 **P. Cabañas-Molero, D. Martínez-Muñoz, P. Vera-Candeas, F. J. Cañadas-Quesada, and N. Ruiz-Reyes**, "Compositional model for speech denoising based on source/filter speech representation and smoothness/sparseness noise constraints", accepted for publication in *Speech Communication*, Special Issue on Advances in Sparse Modeling and Low-rank Modeling for Speech Processing, 2015.

Chapter 1

Introduction

The design of machines that are able to "listen" has been one of the most attractive lines of research in the last years. Although human listeners are extraordinarily skilled in identifying and focusing on speech sounds, even in adverse acoustic conditions, these tasks are extremely challenging for machines. In real scenarios, the speech waveform observed by a listener is altered by multiple factors, including competing sources, environmental noises or degradations introduced by the transmission channel. In these conditions, most of the speech processing knowledge acquired for clean speech is not enough for practical applications, and new techniques are required for handling degraded speech.

In this thesis, we are interested in developing classification and separation techniques that are useful for speech enhancement. We can define speech enhancement as *the set of operations aiming to improve the quality and intelligibility of the desired speech source, introducing some kind of technology between the (possibly noisy) speech signal and the human listener*. Nowadays, thanks to the fast growth of digital systems, this technology for processing digital information is ubiquitous, but the solutions in this field still require intensive research.

There are a great number of applications where speech enhancement and classification plays an important role. For example, in digital hearing aids, it is very useful to automatically classify the acoustic environment surrounding the user. When the presence of a speaker is detected, the device can activate a hearing program to improve the perception of speech. On the other hand, if the input signal is just noise, the device can select a more comfortable program, in order to avoid the amplification of annoying sounds. In hands-free communication systems, the signal reaching the microphone may suffer from a severe degradation, being not intelligible or annoying for the listener at the other side of the communication channel. Speech separation is also useful for restoring old or degraded recordings, for example, in surveillance systems.

1.1 Speech and Fundamental Frequency

Although humans can produce a great variety of speech sounds, the shape of the vocal tract and its mode of excitation are restricted, which enables to find common properties to describe how speech is. Without necessarily knowing much about the speech production process, we can observe several basic characteristics of the speech signal by simply looking at a typical speech waveform, such as the one depicted in Figure 1.1.

We observe that the speech signal

- is *time variant*
- is *quasi-periodic* in some segments (at voiced regions), has a *stochastic spectral character* in other segments (at unvoiced regions) or is *paused*
- is *quasi-stationary* in time intervals of 5-25 ms, which implies that the vocal tract shape (and, consequently, its transfer function) remain nearly unchanged within this interval
- changes *continuously and gradually*, not abruptly.

These characteristics of the speech signal are determined by its generation process, which originates in the lungs as the speaker exhales. Speech consists of pressure waves that are created by the airflow passing through the vocal tract. The vocal folds in the larynx can open and close quasi-periodically to interrupt this airflow, resulting in *voiced speech*. Vowels are the most prominent examples of this type of sound. Voiced speech is characterized by its periodicity, where the frequency of the excitation provided by the vocal chords is known as the *fundamental frequency*. The periodicity of voiced speech gives rise to a spectrum

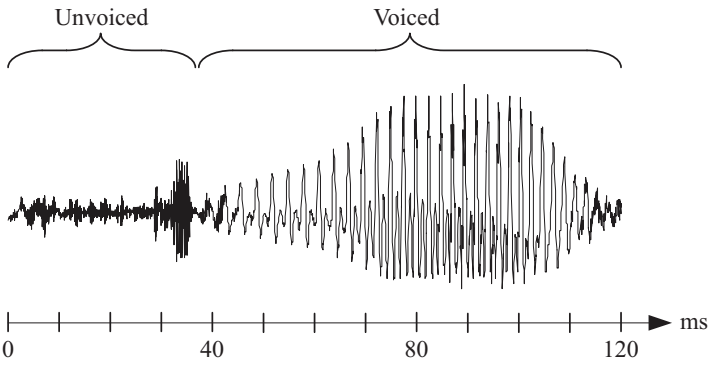

Figure 1.1: A speech waveform of unvoiced and voiced speech.

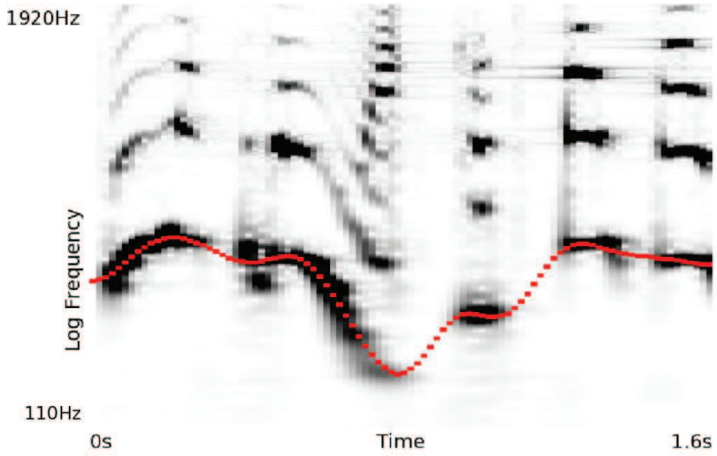

Figure 1.2: A speech spectrogram and F_0 contour.

containing harmonics at approximately multiples of the fundamental frequency f_0, placed at frequencies lf_0, for integers $l \geq 1$. These harmonics are known as *partials*. It is worth noting that voiced segments are not perfectly periodic, but locally quasi-periodic, which causes that the resulting spectrum is not purely harmonic. For *unvoiced speech*, the vocal chords do not vibrate. Instead, the excitation is provided by a turbulent airflow passing through a constriction in the vocal tract, which gives to unvoiced phonemes a certain noisy characteristic. The positions of the other articulators in the vocal tract serve to filter the noisy excitation, amplifying certain frequencies while attenuating others. The spectra for unvoiced speech usually have a more or less flat shape, with prominence of high frequency components.

In Figure 1.2, we can observe more properties of the speech spectrogram and the fundamental frequency. As seen, the pitch produced by a speaker varies slowly across time, according to the speaker's intonation. In addition, as a consequence of the alternation between voiced and unvoiced speech, the spectrum alternates harmonic regions with not periodic regions which do not produce harmonics. When the speech signal is corrupted by background noise, the speech spectrum still has spectral spikes that are strongly marked, while the spectrogram of the background noise has a tendency to be more flat, without significant spiky regions. Consequently, detecting the harmonic parts of the spectrogram in a noisy background can be viewed as searching for thin and harmonically distributed "islands" which rise out of a "sea" of noise [110].

Obviously, the harmonic structure of the spectrum is not exclusive to speech signals. Many other sounds, such as musical instruments, produce harmonics at multiples of a fundamental frequency. However, the evolution across time of

the pitch sequence is characteristic of speech, and presumably is a good cue for discriminating (and separating) speech from other sounds. Some psychoacoustic experiments demonstrate that the human auditory system employs the perceived pitch not only for discriminating the target source from inharmonic noises, but also from other harmonic interferences [25].

Consequently, when designing an algorithm for speech classification or separation, we can take the following points into account:

- The pitch contour and the voiced/unvoiced behavior of speech can provide useful features for speech classification.
- For speech separation, the employed speech model should have the typical harmonic structure of speech, and the background model should have a non-harmonic spectrogram or be prevented from having similar pitch evolution.

1.2 Scope of the Thesis

This thesis is focused on developing classification and separation algorithms for speech signals affected by background noise. We have employed the pitch properties of speech signals to derive these algorithms. In this thesis, we intent to explore two application scenarios:

- Ultra-low power devices, such as hearing aids, with restricted computational capacity.
- Usual scenario without computational restrictions.

For both scenarios, we suppose that the background noise only contains environmental sounds, and not a competing speech signal. No further assumptions are made about the background noise or the speaker. The algorithms must work with one channel signals. We can formulate the following objectives:

- Develop a sound classification algorithm for hearing aids able to, at least, discriminate between speech and nonspeech classes, based on features extracted from F0.
- Develop a frame-by-frame voicing detection algorithm to separate speech from background noise based on pitch and voicing decisions, and explore its implementation on hearing aids.
- Explore decomposition algorithms based on compositional models to create signal models for speech and background noise, with the aim of separating both signals.

1.3 Scientific Contributions

Main scientific contributions of the thesis comprise:

- Formulation of the decimated difference function for estimating F0 in ultra-low power devices, where the computational resources are very limited [P1].
- Definition of a dynamic threshold for the aperiodicity measure derived from the difference function, enabling to detect voiced segments with this feature in the presence of non-stationary noise [P2].
- Formulation of a compositional model based on mathematical constraints that represent the properties of background noise in a generic way, and are highly discriminative with respect to the typical shape and pitch evolution of speech [P3].

Three publications are included in this thesis, summarized in the following list. Chronological order of publishing is used.

[P1] Low-complexity F0-based Speech/Nonspeech Discrimination Approach for Digital Hearing Aids

As mentioned earlier, digital hearing aids impose strong complexity and memory constraints on the development of signal processing algorithms, avoiding the application of conventional solutions. This paper proposes a low-complexity approach for automatic speech/nonspeech classification in digital hearing aids, based on an efficient estimation of the fundamental frequency. The proposed scheme consists of two stages: analysis and classification. In the analysis stage, a set of signal features derived from F0 are computed. Here, F0 is estimated using a decimated version of the difference function, which considerably reduces the required number of operation per second with respect to the conventional difference function. For the classification stage, two low-complexity classifiers are evaluated: the C4.5 decision tree and a Multi-layer Perceptron (MLP), the MLP being finally chosen because it provides the best classification accuracy rates and fits to the typical computational and memory constraints of hearing devices. Finally, a Hidden Markov Model (HMM) is used to provide some temporal context to the decision sequence. To demonstrate the feasibility of its implementation in a realistic hearing aid, the number of operations and memory requirements of the algorithm are analyzed, and compared to the computational capacity of a realistic processor for hearing aids. For the experiments, an audio database including clean speech, noisy speech, music and noise signals has been used.

[P2] Voicing Detection based on Adaptive Aperiodicity Thresholding for Speech Enhancement in Non-stationary Noise

In this study, we present a novel voicing detection algorithm that employs the well-known aperiodicity measure to detect voiced speech in signals contaminated with non-stationary noise. The method computes a signal-adaptive decision threshold which takes into account the current noise level, enabling voicing detection by direct comparison with the extracted aperiodicity. This adaptive threshold is updated at each frame by making a simple estimate of the current noise power, being thus adapted to fluctuating noise conditions. Once the aperiodicity is computed, the method only requires a small number of operations, and enables its implementation in challenging devices (such as hearing aids) if the difference function is computed as proposed in [P1]. Evaluation over a database of speech sentences degraded by several types of noise reveals that the proposed voicing classifier is robust against different noises and signal-to-noise ratios. Additionally, to evaluate the applicability of the method for speech enhancement, a simple F0-based speech enhancement algorithm integrating the proposed classifier is implemented. The system is shown to achieve competitive results, in terms of objective measures, when compared with other well-known speech enhancement approaches.

[P3] Compositional Model for Speech Denoising based on Source/Filter Speech Representation and Smoothness/Sparseness Noise Constraints

This work presents a speech denoising algorithm based on a regularized non-negative matrix factorization (NMF), in which several constraints are defined to describe the background noise in a generic way. The observed spectrogram is decomposed into four signal contributions: the voiced speech source and three generic types of noise. The speech signal is represented by a source/filter model which captures only voiced speech, where the filter bases are trained on a database of individual phonemes (resulting in a small dictionary of phoneme envelopes) and the source bases are pitch-related excitation patterns. The three remaining terms represent the background noise as a sum of three different types of noise (smooth noise, impulsive noise and pitched noise), where each type of noise is characterized individually by imposing specific spectro-temporal constraints, based on sparseness and smoothness restrictions. The method was evaluated on the CHiME-3 development dataset and compared with conventional semi-supervised NMF with sparse activations. Our experiments show that, with a similar number of bases, source/filter modeling of speech in conjunction with the proposed noise constraints produces better separation results than sparse training of speech bases.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 introduces some fundamental concepts about audio classification, with special attention to classification problems involving speech signals for enhancement. Techniques for fundamental frequency estimation are also reviewed. Chapter 3 focuses on speech enhancement techniques, including algorithms based on matrix decomposition and compositional models. Chapter 4 presents the conclusions of the work. Finally, the articles published during the development of this thesis are included.

Chapter 2

Speech Classification for Enhancement Applications

2.1 Introduction

The general structure of an automatic sound classification system can be described with the block diagram depicted in Figure 2.1. The basic idea is to categorize the input signal into one of a set of possible output classes, according to a predefined taxonomy. From the sound data, a number of relevant features are extracted which are then classified by some sort of classification algorithm. Possible classification errors may be corrected by an optional post-processing step, which also controls the transient behavior of the algorithm. As an output, a label describing the class of the signal is returned.

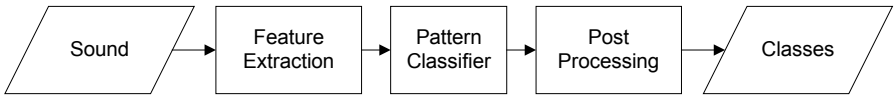

Figure 2.1: Block diagram of an audio classification system.

The criterion upon which the signals are classified depends on the target application, which also determines other properties of the system, such as decision delay or computational complexity. For example, in a music database manager, it is useful to organize audio collections by music genre. For an audio coder, the optimal coding scheme can be selected if the system is able to identify the audio at the input as speech or music. In the context of speech signals, automatic speech recognition (ASR) can be viewed as a classification problem, in which the input waveform is converted into a sequence of lexical units. In this thesis, we are interested in classification problems that are, in some way, useful for speech

enhancement purposes. If the ultimate goal is to improve the speech signal perceived by a listener, a classification algorithm can help in the process by selecting the best enhancing approach at each moment or by setting certain parameters of the enhancer. The following problems show why sound classification is useful for speech enhancement.

Acoustic environment classification is a topic of great importance in digital hearing aids [2, 12, 99]. According to the classifier decision, the device can select the most appropriate amplification program to the detected acoustic environment, thus increasing the comfort level. For instance, suppose that the user is listening to a speaker in the presence or not of certain background noise, such as in a conference or when watching television. In this situation, the hearing aid should decide that it is worth amplifying the signal (for example, by selecting a "speech amplification program") in order to help the user understand the message. On the contrary, if the user is embedded in a noisy place, such as a traffic jam, the device should switch off the amplification program, thus avoiding to amplify unpleasant noises and saving battery life. Detecting more specific acoustic environments, such as "music", "speech in noise", "speech in quiet" or "speech in music", is useful when the device implements hearing programs targeted to those situations.

Voice activity detection (VAD) stands for locating speech segments from a noisy input. Clearly, for enhancement applications, VAD can be used for updating noise models (from detected silence) and selecting speech segments for further processing (possibly taking advantage of the acquired noise models). Since other sources may be active at amplitudes comparable to target speech, just observing the input signal activity does not suffice for robust VAD. Consequently, VAD remains as a complex classification problem [142], specially if the noise is nonstationary or speech-like. VAD can be viewed as a particular case of acoustic environment classification, in which "speech" and "nonspeech" are the only considered options. One of the contributions of this thesis is the design of a speech/nonspeech discrimination system for hearing aids, based on features derived from fundamental frequency [P1].

Voicing detection is the process of determining speech segments produced by vibration of the vocal chords [1]. Unlike VAD, whose purpose is to determine the presence of speech, either voiced or unvoiced, voicing detection focuses only on voiced parts. Since voiced segments are more or less periodic, this problem is often associated with fundamental frequency estimation, although this is not always the case. Voiced detection is very useful for certain enhancement approaches, specially for those involving binary masking

[65], comb filtering [20], harmonic tunneling [34] or sinusoidal synthesis [68]. In this thesis, we propose a voicing detection algorithm based on a single feature, the aperiodicity, whose computation is made robust against nonstationary background noise [P2].

Classification and enhancement of speech signals can be viewed as closely related problems, in the sense that the solution of the former facilitates resolving the later, and vice versa. A perfect speech separation reduces the classification problem to characterizing separated sources, whereas a perfect classification enables to accurately approximate separation parameters and source signal models.

2.2 Feature Extraction

2.2.1 Signal Analysis

Digital audio signals are generally recorded as time-domain pulse-code modulation (PCM) waveforms. In order to analyze the signal, the input waveform is broken into small (possibly overlapping) short-term *frames*, also called *analysis windows*. The temporal resolution of the system can be characterized by two parameters, *frame length* and *frame shift*. The former defines the duration of each frame. It is set to a value where the input can be assumed mostly stationary, which for speech may stand for 20–64 milliseconds. For classification, short frame lengths (around 20–25 ms) are preferred for capturing rapid dynamics of speech, whereas longer frames (around 64 ms) are often used in signal enhancement, where slower transitions reduce audible artifacts from estimation errors. Frame shift is the amount of input time advanced before extracting a new frame, and is usually set to 50% or less of frame length. The difference of these two values is *frame overlap*.

The input signal is usually transformed into a spectral domain, commonly by Fourier analysis of the short-term frames. Before computing the short-time Fourier transform (STFT), input frames are conventionally multiplied by a window function (e.g. Hann or Hamming window), in order to increase the sensitivity to weaker spectral components. The sequence of short-term spectra over time is called the *spectrogram*, denoted by $X(t, f)$, where f is the frequency bin index and t is the time-frame index. Other usual frequency-time representations are obtained by filtering the time-domain signal with a filterbank of bandpass filters, such as in the weighted overlapp-add (WOLA) analysis [24], common in hearing aids, or the *cochleagram* [94], which is used for extracting auditory-based features.

In the previous introductory discussion, we have assumed that a set of features, arranged as a feature vector $\mathbf{c}$, represents the object to be classified as a whole. Depending on the portion of time represented by a feature vector, we can distinguish two different approaches: *frame-based feature vectors* and *texture-based feature vectors*. In the frame-based approach, a new feature vector is computed for each frame or analysis window, hence representing the properties of the signal in portions of 20–64 ms, as mentioned above. The frame-based approach is useful when real-time classification is desired. However its major drawback is that it does not allow to take into account other long-term characteristics of the signal that often provide a better description of the different sound classes. While two signals, corresponding to different classes, may appear similar in a single frame, observing their long-term behavior is more likely to bring out the characteristic patterns, enabling a robust discrimination. As a result, the concept of *texture window* was introduced. A texture window is a long-term segment (in the range of seconds) containing a number of analysis windows. In the texture-based approach, only one feature vector for each texture window is generated. This feature vector is not just the concatenation of the vectors obtained in each frame, but often statistical measures of them, such as the mean or standard deviation. Also, certain features only have sense for long-term signal segments, and are often defined from features measured at each frame. For example, a feature describing properties of the pitch contour of a signal can only be computed for an extended observation period, based on consecutive frame-by-frame pitch estimations. When working with texture windows, the temporal resolution of the classifier is defined by the *texture window length* and the *texture window shift* parameters, both measured in number of frames.

The texture-based approach is preferred in applications requiring an accurate classification, and where the decision can be returned with a certain delay. For instance, in the case of hearing aids, it is crucial to provide a robust and stable decision, even if the system takes a few seconds in reacting to environment changes [13, 99].

2.2.2 Signal Features

In order to classify an incoming signal, some measures or *features* are extracted from it. A set of D features extracted from an analysis or texture window is represented as a D-dimensional vector $\mathbf{c} = [c_1, c_2, \ldots, c_D]^{\mathrm{T}}$ called *feature vector*. The key point is that the chosen features must contain valuable information that allows to properly distinguish among the considered classes. In other words, the features should measure signal properties that tend to present distinguishable values among the different audio classes.

In classification of signals affected by noise, there are two approaches for extracting signal features. The first one consists in formulating features (or combinations of features) that are robust to background noise, such that the signals pertaining to a certain class exhibit characteristic values for those features (or for their combination) in both clean and noisy conditions. The second one is based on performing some kind of preprocessing to estimate the effect of the noise before extracting the features, which are then redefined to take into account this effect. This preprocessing may be accomplished with noise estimation techniques (see Section 3.2.5). There is an obvious third approach consisting in enhancing the signal before extracting the features. However, in our study we are analyzing the classification problem as a tool for resolving signal enhancement and not the other way around.

Many signal features have been proposed in the literature for resolving the problems outlined in Section 2.1. In the following, we overview some of the most common of these features.

Timbral Features

These features provide numerical quantities measuring the spectral shape of the signal. Probably, the most famous is the *spectral centroid*, defined in [115]. The spectral centroid of a short-term spectrum $X(t, f)$ is a measure of the center of gravity of its energy distribution, and thus, it outlines if the spectrum contains a majority of high or low frequencies. Higher centroid values correspond to spectra skewed to the range of high frequencies. Due to its effectiveness to describe the spectral shape, the centroid has often been used in audio classification tasks including speech, noise and music classes [84, 129]. A similar feature is the *spectral rolloff*, defined as the frequency below which 85% of the accumulated magnitudes of the spectrum is concentrated. This measured was first proposed as a feature to distinguish between voiced and unvoiced speech [115], since voiced frames tend to have a lower rolloff. It has also been found useful for discriminating speech from other sources, such as music [84]. The *spectral flux* is the average difference between the magnitude spectra corresponding to successive frames of the STFT. This feature is related to the amount of spectral local changes, being generally higher for speech than for noise or music [87]. The *voice2white* parameter, proposed in [54], is a measure of the energy inside the typical speech band (300–3600 Hz) with respect to the whole energy of the frame. Consequently, this feature is useful for discriminating between speech and nonspeech signals. Features such as the *spectral flatness measure* [66], the *Renyi entropy* [64] or the *Shannon entropy* [106] measure the degree of randomness in the signal, and hence are also adequate for speech and noise discrimination. In the time domain, the *zero cross-*

ing rate counts the number of times that the signal amplitude changes sign during the analysis window [129]. As with the previous features, it is also a measure of noisiness.

Auditory-based Features

The overwhelming majority of speech recognition systems today, as well as many classification algorithms, make use of features that are based on either *Mel Frequency Cepstral Coefficients* (MFCCs) [26] or features based on *perceptual linear predictive* (PLP) analysis of speech [59]. MFCCs are a compact representation of the spectrum of an audio signal that takes into account the nonlinear human perception of pitch. For the extraction of MFCCs, the FFT bins are combined according to a set of triangular weighting functions that approximate the human pitch perception as described by the Mel scale. This can be viewed as filtering the spectrum with a filterbank of triangular bandpass filters, and then integrating the output of each filter over the frequency. The filterbank usually consists of 40 filters, such that the 13 first filters (low frequencies, below 1 kHz) have linearly spaced center frequencies, and the 27 last filters (high frequencies, above 1 kHz) have logarithmically spaced center frequencies. The 40 filterbank output coefficients are log compressed, an a Discrete Cosine Transform (DCT) is applied to decorrelate the coefficients, providing the so-called MFCCs. Usually, for classification tasks, only the first coefficients (between 5 and 20 MFCCs) are useful for obtaining a good performance. The cepstral computation can also be thought of as a means to separate the effects of the excitation and frequency-shaping components of the source-filter model of speech production.

The computation of the PLP coefficients is based on a somewhat different implementation of similar principles. As in MFCC processing, the input spectrum is weighted and integrated using a set of asymmetrical functions based of auditory perception. In this case, these functions are spaced according to the Bark scale, and are based on the auditory masking curves of [119]. The filter-bank output values are weighted by a preemphasis step to simulate the sensitivity of hearing (according to the equal-loudness curve), and the equalized values are raised to the power of 0.33. The resulting spectrum is processed by linear prediction to obtain a smoothed approximation based on all-pole modeling. Finally, cepstral coefficients are obtained from the predictor coefficients by a recursion that is equivalent to the logarithm of the all-pole model spectrum followed by an inverse Fourier transform. PLP processing is also frequently used in conjunction with the RASTA (relative spectral analysis) algorithm [60]. RASTA processing inserts a bandpass filter after the compressed values that emerge from the preemphasis step, in or-

der to model the tendency of the auditory periphery to emphasize the transient portions of incoming signals.

The *amplitude modulation spectrograms* (AMS) are motivated by neurophysiological experiments on periodicity coding in the auditory cortex of mammals [127]. The AMS representation is a two-dimensional feature which contains information about the prominent modulation frequencies for each center frequency. Each complex coefficient of the FFT is considered as a function of time across consecutive frames, i.e., as a band pass filtered complex time signal. The number of bands is reduced to a few channels (between 3 and 15) by adding the FFT coefficients of neighboring bands, grouped according to a Bark scale. The signal in each band is analyzed again by computing the Fourier transform, producing a modulation spectrum for each channel. The modulation spectrum at each band is discretized into a few coefficients following a logarithmic scale. For voiced speech, the AMS feature matrix exhibits vertical bars at the fundamental frequency and its multiples, being useful for speech and noise discrimination.

Other features inspired on Auditory Scene Analysis measure properties related to the onset/offset of sounds, frequency modulation, pitch or voicing [12]. Fundamental frequency has a great potential for characterizing speech signals, because speech has a characteristic pitch behavior. Techniques for fundamental frequency estimation are reviewed in Section 2.4, along with the pitch-based features employed in this thesis [P1] for speech/nonspeech discrimination in hearing aids.

Other Features

Certain features describe the signal regarding its dynamic energy properties, its statistical behavior or its predictability [14]. Although the energy level of the signal in a single frame is irrelevant for classification, its long-term variation can provide useful information in distinguishing audio types. Several features have been proposed to describe the smoothed trajectory of the signal level, often extracted from texture windows. Some examples are the *low energy rate*, the *evelope* or the *loudness*. The statistical behavior of the signal can be described mathematically by the central moments of its time-domain waveform, in features such as the *sample skewness* or the *sample kurtosis* [14].

Another common feature in audio classification is derived from linear prediction analysis. A P-order linear prediction of a sample $x(n)$ is a prediction of its amplitude value as a linear combination of its past P samples, in the form $a_1x(n-1) + a_2x(n-2) + \ldots + a_Px(n-P)$. The a_p coefficients are called the *Linear Prediction Coefficients* (LPC), and can be obtained by one of several algorithms proposed in the literature, which aim at obtaining a prediction error as lowest as possible. One of these algorithms is the so-called autocorrelation

method of autoregressive modeling [67]. The set of coefficients a_p can be used as a feature for classification, as well as the prediction error. Signals with sudden amplitude changes and high noise components are more likely to yield higher values for the prediction error and vice versa.

Noise-adapted Features

A common approach in classification of signals degraded by noise is to redefine classic features taking into account the estimated noise level or the long-term behavior of the features. Usually, the long-term behavior of a feature is a good indication of its expected value in absence of speech. An example of this approach is used in the G.729B standard [5], which conducts a VAD decision on every frame using four different parameters: a full-band energy difference, a low-band energy difference, a differential spectral flux measure and a zero-crossing rate difference. Essentially, these parameters are noise-adapted versions of classic features (full-band energy, low-band energy, spectral flux and zero-crossing rate) which are formulated as the difference between the parameter itself extracted in the current frame and its long-term average. The long-term averages of the parameters are supposed to follow the changing nature of the background noise, and are updated based on a first order autoregressive scheme only if the full-band energy difference is less than a certain threshold.

A similar approach is applied in the ETSI advanced front-end (AFE) standard [38], but using the logarithmic energy (and its long-term estimated mean) as a unique feature. In this case, the VAD decision is based on a SNR threshold and a hangover mechanism that updates the mean logarithmic energy. A more elaborate process is employed in the ETSI extended front-end standard [37]. Here, the algorithm maintains an estimate of the noise energy spectrum (defined on a mel frequency scale), and both a smoothed and long-term average version of the signal espectrum. At each frame, the algorithm computes the deviation between the smoothed signal spectrum and its long-term average, and the peak-to-average ratio in the smoothed signal spectrum. Whenever these quantities are below a threshold (which is an evidence of nonspeech), the noise spectrum is updated using a smoothing operator. The final feature for taking the VAD decision is the estimated SNR. The algorithm also provides a voicing decision based on a pitch estimator, in which the presence of pitch is determined from a correlation score for each generated pitch candidate.

2.3 Classification Algorithms

After the feature extraction process, a decision on the class to which the input signal belongs to must be made based on the extracted features. This process is performed by the classifying algorithm. The extracted feature vector $\mathbf{c}$ forms the input to the classifier, and the output is the assignment of the input signal to one of the C considered audio classes, denoted as w_k, with $k = 1, \ldots, C$.

From a graphical point of view, classifying means to find in which decision region falls a given feature vector, and assigning to this vector the class w_k corresponding to the estimated region (see Figure 2.2). The goal of pattern recognition is to use a set of available training samples to find decision boundaries that separate the classes in an optimal way. In other words, in a *training stage*, the borders between classes that provide the best discrimination for a given set of training vectors (whose class is known) are found. Then, in the *test stage*, the trained classifier can classify new unknown vectors according to the computed boundaries. Note that the training stage is performed *off-line*, in a design phase, and not in the final application.

This intuitive idea of classification can be expressed more formally as follows: given a classification problem with C classes, a set of C discriminant functions $g_k(\mathbf{c})$ is defined. Classification of a feature vector $\mathbf{c}$ consists of performing the following operation:

$$\text{Decide } w_i \text{ if } \; g_i(\mathbf{c}) > g_j(\mathbf{c}) \quad \text{for all } j \neq i. \tag{2.1}$$

Thus, a classifier can be viewed as an algorithm that evaluates all discriminant functions for a given feature vector and assigns the class corresponding to the largest discriminant. The goal of the training phase is to derive this set of discriminant functions.

Depending on the chosen approach for finding these functions, classifiers can be grouped into *generative classifiers* and *machine learning* classifiers. The first ones model the probability density function (pdf) of the feature set for each audio class. In this case, classification can be interpreted as determining the probability of each class for a given input vector, and selecting the class that produces the highest probability. This type of classification algorithms include, among others classifiers, Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs) and, in general, all algorithms based on Bayesian classification. The advantage of these classifiers is that they are relatively easy to train because the pdfs of the feature set can be determined separately for each class. This training process involves estimating the parameters of the pdfs, which usually follow well-known and tractable mathematical expressions. For some algorithms, the pdf for certain features are completely known (including their parameters), and no training

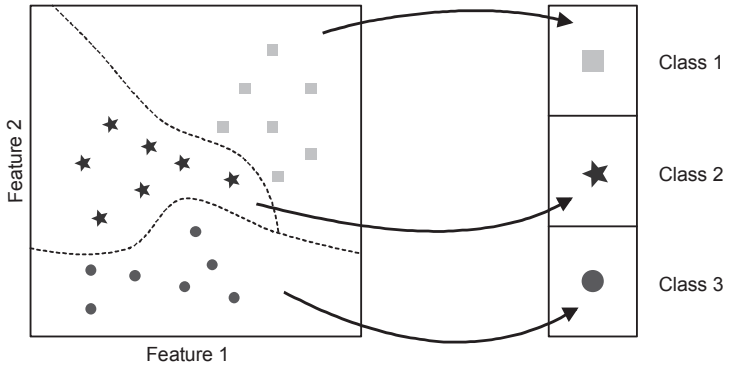

Figure 2.2: Pattern classification viewed as establishing a mapping from a feature space to a decision space.

is required. The major drawback of generative classifiers is, however, that they demand to know the probability distributions for the feature set a priori.

Machine learning classifiers, on the other hand, determine the boundaries between classes without modeling the probabilities directly. Instead, they approximate the discriminant functions by connecting a set of operations in series, where the parameters of these operations can be learned to achieve optimal class boundaries. Examples of these classifiers include Artificial Neural Networks, Support Vector Machines or decision trees. Training this kind of algorithms can be a difficult task because it involves considering all audio classes at the same time, and often demands the use of iterative algorithms which are not guaranteed to converge to a good result. On the other hand, discriminative classifiers can be very adequate when there is a large training set or a high number of features, and they are the only option when the underlying pdfs are unknown.

2.3.1 Generative Classifiers

Bayesian Classification

As mentioned above, the central problem in statistical pattern recognition is finding the set of discriminant functions for a classifier. The *Bayes Decision Theory* describes the classification problem when the pdfs of the classes are known. The pdf describing each class $p(\mathbf{c}|w_k)$ is the conditional pdf of $\mathbf{c}$ given the class w_k. This quantity is also called the *likelihood* of the observed vector when hypothesizing class w_k. Besides, it should be noted that, in general, some classes in a classification problem are more probable than others. Therefore, each class is also associated with its *a priori probability* $p(w_k)$.

In order to make a classification, we want to know how probable a class w_k is, given an observation $\mathbf{c}$. This is the so-called *a posteriori probability* $p(w_k|\mathbf{c})$, and its relationship to the class likelihood is provided by the *Bayes Rule*:

$$p(w_k|\mathbf{c}) = \frac{p(\mathbf{c}|w_k)p(w_k)}{p(\mathbf{c})}, \tag{2.2}$$

where $p(\mathbf{c}) = \sum_{k=1}^{C} p(\mathbf{c}|w_k)p(w_k)$. Consequently, for a given observation $\mathbf{c}$, the classifier will decide the class w_k for which the posterior probability is highest, thus obtaining the following decision rule:

$$\text{Decide } w_i \text{ if } \; p(w_i|\mathbf{c}) > p(w_j|\mathbf{c}) \quad \text{for all } j \neq i. \tag{2.3}$$

This decision rule is called the *Maximum A Posteriori* (MAP) criterion. Observe that the term $p(\mathbf{c})$ is only a scale factor and does not affect the decision. Then, we can conclude that a MAP classifier is the one whose discriminant functions are given by $g_k(\mathbf{c}) = p(w_k|\mathbf{c})$.

In many problems, all classes are equally probable a priori. In this case, the $p(w_k)$ term is constant for all k and therefore it does not affect the decision. Thus, with equal priors, maximizing the posterior probabilities is the same as maximizing the likelihoods, obtaining the following rule:

$$\text{Decide } w_i \text{ if } \; p(\mathbf{c}|w_i) > p(\mathbf{c}|w_j) \quad \text{for all } j \neq i, \tag{2.4}$$

which is called the *Maximum Likelihood* (ML) criterion. To decide a class for a given feature vector, we evaluate each conditional pdf and select the one that provides a higher value. The discriminant functions of a ML classifier are $g_k(\mathbf{c}) = p(\mathbf{c}|w_k)$.

When the density forms of the classes are known, but their parameters are not, it is necessary to use a *parameter estimation* technique in a training stage, using a set of training feature vectors.

Gaussian Mixture Models

A Gaussian Mixture Model (GMM) represents the distribution of each class as a weighted sum of gaussian densities. Hence, each class w_k can be modeled as a mixture model, obtaining class likelihoods of the following form:

$$p(\mathbf{c}|w_k) = \sum_{m=1}^{M} a_{km} p_{km}(\mathbf{c}), \tag{2.5}$$

where M is the number of gaussians and a_{km} are the weights of each gaussian. The individual gaussian densities $p_{km}(\mathbf{c})$ are called the *components* of the mixture,

such that $p_{km}(\mathbf{c}) \sim N(\boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km})$, where each component within each class has its own mean vector $\boldsymbol{\mu}_{km}$ and covariance matrix $\boldsymbol{\Sigma}_{km}$.

Training a GMM is done by *maximum likelihood* (ML) parameter estimation, in which the set of parameters of the gaussians $\boldsymbol{\theta}_k = \{a_{km}, \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}\}$ is estimated by maximizing the likelihood of a giving training set. Suppose that a training set consisting of J training samples $\mathbf{c}_j$ is available for a given class w_k. ML estimation of $\boldsymbol{\theta}_k$ consist of resolving the following problem:

$$\hat{\boldsymbol{\theta}}_k = \arg\max_{\boldsymbol{\theta}_k} \prod_{j=1}^{J} p(\mathbf{c}_j | w_k, \boldsymbol{\theta}_k), \tag{2.6}$$

where now we have written the class density as $p(\mathbf{c}|w_k, \boldsymbol{\theta}_k)$ to denote its dependency of the set of parameters. Usually, this problem is accomplished by making use of a so-called *expectation-maximization* (EM) algorithm [28]. This consists of a set of iterations in which the parameters are updated in such a way that the the criterion in (2.6) increases monotonically until a certain threshold is reached.

Hidden Markov Models

So far we have assumed that, given a single feature vector $\mathbf{c}$, the system takes an immediate decision on the signal class. However, in many classification problems, it is necessary to observe a sequence of measurements through time, $\mathbf{c}_0, \mathbf{c}_1, \ldots$, in order to make a reliable decision. A clear example is seen in speech recognition, in which the decision about the word or phoneme pronounced by a speaker must be made after observing a sequence of vectors. In these cases, an audio class is not only defined by characteristic values of the feature vectors, but also by their progression through time. The most effective classifying tool for these situations is based on a structure referred to as Hidden Markov Models (HMM) [104].

HMMs are statistical models of time-series data. An HMM models a time series as having been generated by a process that goes through a series of *states*. Depending on the problem, each state may correspond to a different sound class or to a different phase within a certain sound class, in which case a different HMM is defined for each possible class. Suppose that the model has Q different states, denoted as q_i, with $i = 1, \ldots, Q$. Given a feature vector $\mathbf{c}_t$, the likelihood of this vector supposing that the model is in the state q_i at instant t is $p(\mathbf{c}_t|q_i, \mathbf{b}_i)$, where $\mathbf{b}_i$ is the set of parameters of the density function, and $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_Q\}$ is the global set of density parameters. The model is named hidden because the sequence of states is not observable, but only the visible feature vectors, which are supposed to be generated from the true sequence of states according to their corresponding density functions. When in any state, the next state that the process

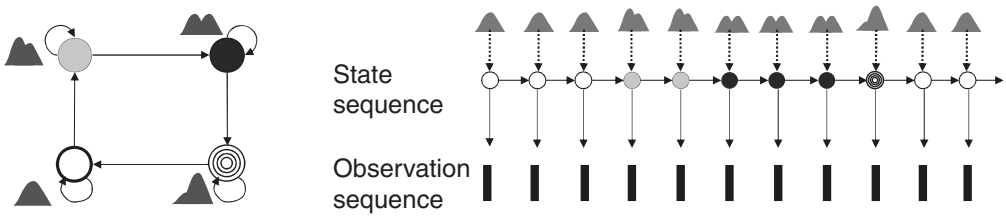

Figure 2.3: Schematic illustration of a HMM. The four circles represent the states of the HMM and the arrows represent allowed transitions. Each HMM state is associated with a state output distribution as shown. The process progresses thorough a sequence of states. At each visited state, it generates an observation by drawing from the corresponding state output distribution.

will visit in the next time instant is determined stochastically, and is only dependent on the current state. The transition probability can be defined as the matrix $\mathbf{A} = \{a_{ij}\}_{ij}$, where a_{ij} is the probability of a transition from state q_i to state q_j. Also, an initial state distribution $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_Q\}$ is defined, where π_i is the probability that state q_i is the first state in the state sequence. Graphically, the progression of a HMM is represented in Figure 2.3.

The compact notation $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ is used to represent a HMM. The design of a HMM for an audio class includes choosing the number of states Q as well as the density forms of the pdfs (e.g. GMMs), and estimate all the parameters using a training set. The most common parameter estimation procedure is the Baum-Welch algorithm [71], based on expectation-maximization. In the test stage, given a sequence of vectors $\mathbf{c}_0, \mathbf{c}_1, \ldots$, all HMMs are evaluated to determine their sequence of states, and the HMM (i.e. audio class) that produces the state sequence with higher probability is chosen.

HMMs can also be useful as a post-processing stage, in order to incorporate temporal information to the decision process. For example, each audio class can be represented by a different state, such that the resulting state sequence can be viewed as a smoothed decision sequence. In this case, the transition probabilities must be chosen carefully to determine the steadiness of the system.

2.3.2 Machine Learning Classifiers

k-Nearest Neighbor Classifier

The k-nearest neighbor classifier, or k-NN, is essentially a distance-based classifier [32]. To obtain the class corresponding to a new vector $\mathbf{c}$, the algorithm

simply looks for its k nearest vectors (*neighbors*) in the training set, and weigh, usually applying a majority rule, the class number they belong to.

For expressing this idea in a more formal way, let us consider a set of J training samples $\mathbf{c}_j$ organized into C different classes. The algorithm computes a distance measure between $\mathbf{c}$ and each vector $\mathbf{c}_j$ in the training set, and selects the k nearest $\mathbf{c}_j$. This distance criterion is often based on the Euclidean distance, so the algorithm computes J distances as $d_j = ||\mathbf{c} - \mathbf{c}_j||$. The classifier assigns the label which is most frequent among the k nearest samples (according to distances d_j). Usually, choosing moderate values for k improves performance in comparison with choosing simply the class of the nearest vector, because it yields smoother decision boundaries and provides more probabilistic information. However, large values for k can be detrimental, not only because of the increased computation complexity, but because it destroys the locality of the estimation by considering samples that are too far away. In addition, from a computational point of view, a k-NN classifier requires to store all feature vectors of the training database in order to compare the input vector with each training instance. Consequently, if the classifier is implemented in a low-power device, such as a hearing aid, the number of training samples J must be very limited, as well as the value for k.

Artificial Neural Networks

An artificial neural network [62] is a parallel, distributed information processing structure consisting of a set of processing units, called *neurons*, interconnected via unidirectional links called *connections*. Each neuron has one or more inputs values and produces a single output which branches into one or more connections to feed other neurons. The mathematical operation performed within each neuron can be defined arbitrarily, with the restriction that it must be completely local; that is, it must depend only on the current input values arriving at the neuron and on values stored in the neuron's local memory.

Among the multiple variations of neural networks, the most common is probably the multilayer perceptron (MLP). The basic architecture of a multilayer perceptron consists of three layers of neurons (input, hidden and output layers) in which each neuron in the hidden and output layers is interconnected with all the neurons in the previous layer by links with adjustable weights. This type of neural networks is commonly known as "feed-forward neural networks", and is probably the most popular and widely-used network in many practical implementations. It has the advantage that there are good training algorithms to determine the parameters of the network, and the computational cost for classifying an input vector is moderate and deterministic. It is worth mentioning that multilayer perceptrons may have more than one hidden layer, but it has been shown that a single hidden

layer is sufficient enough to approximate any function to arbitrary accuracy, given a sufficient and finite number of neurons.

For a classification problem with C classes, in which the input feature vector $\mathbf{c} = [c_1, c_2, \ldots, c_D]^{\mathrm{T}}$ is composed of D features, the number of neurons in the input layer is usually set to D, and the number of neurons in the output layer is set to C. With this configuration, each neuron in the input layer is fed by a single feature of the vector, and each neuron in the output layer produces the probability value for a single class. The number of neurons in the hidden layer M determines the complexity of the network, and must be designed carefully. If too many hidden neurons are used, the capability to generalize will be poor; on the contrary, if too few hidden neurons are considered, the training data cannot be learned satisfactorily.

As mentioned before, each neuron in the input layer is connected to all the neurons in the hidden layer, where each connection has an associated weight, denoted as a_{dm}, with $d = 1, \ldots, D$ and $m = 1, \ldots, M$. The output value y_m produced by the mth hidden neuron can be expressed as follows:

$$y_m = f\left(\sum_{d=1}^{D} c_d a_{dm} + b_m\right), \tag{2.7}$$

where b_m is a parameter of the neuron and $f(\cdot)$ is the transfer function executed in the neuron. This transfer function can take a variety of mathematical expressions, but the most common in MLPs is the logarithmic sigmoid, with the form $f(x) = 1/(1+e^{-x})$. The output neurons perform the same processing that the one in (2.7), but they are fed by the values y_m produced by the hidden neurons, and connected to them by links with their corresponding weights.

In the training process, the weights of the network and the parameters of each neuron are adjusted to approximate the desired function (i.e., to minimize the classification error for the given training set). A variety of algorithms has been proposed in the literature aiming at training multilayer perceptrons, including the gradient descent, Gauss-Newton or Levenberg-Marquardt [8].

In the last years, *deep neural networks* (DNNs) have rapidly gained popularity for resolving complex classification problems [63]. Proposed in 2006, DNNs enable to approximate discriminative functions with a large number of features and training instances. Compared to the training methods of traditional deep models with a high number of hidden layers, DNNs can prevent over-fitting to the training set via a special unsupervised pre-training procedure. Also, they can express highly variant functions, discover the underlying regularity of multiple features, and have strong generalization abilities. Recently, DNNs have received much attention in the speech processing community, with successful applications in speech recognition, natural language processing and classification [142].

2.4 Fundamental Frequency Estimation

A key property of many sounds, including speech and music, is the *pitch*. In the context of music, the American Standard Association defines the term pitch as *that attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from low to high* [3]. As such, the pitch of a sound is strictly speaking a perceptual phenomenon, although it is caused by physical stimuli that exhibit a certain behavior. Signals that cause the sensation of pitch are, broadly speaking, that kind of signals that are well-described by a set of harmonically related sinusoids, meaning that their frequencies are approximately integer multiples of a *fundamental frequency*. In fact, we can say that pitch is the perceptual correlate of the fundamental frequency of a signal, and is often described as "the perceived fundamental frequency of a sound". In the literature, however, it is common to use the terms pitch and fundamental frequency indistinctly, and we will do so throughout the text.

Signals that have frequencies that are integer multiples of a fundamental frequency can be represented using the following model for $n = 0, \ldots, N-1$:

$$x(n) = \sum_{l=1}^{L} A_l \cos(\omega_0 l n + \phi_l). \tag{2.8}$$

This signal model is often known as the harmonic model. The quantity ω_0 is the fundamental frequency and L is the number of harmonics, where the term harmonic refers to each sinusoid in the sum of (2.8). $A_l > 0$ and $\phi_l \in (-\pi, \pi)$ are the amplitude and the phase of the lth harmonic, respectively. The amplitude determines how dominant (or loud) a given harmonic is, while the phase can be thought of as representing a time-shift of the harmonic, as we can express the argument of the cosine function as $\omega_0 l n + \phi_l = \omega_0 l (n - n_l)$, with $n_l = \frac{\phi_l}{\omega_0 l}$. The number of harmonics L can be any integer between 1 and π/ω_0, although it is generally not possible to say in advance how many harmonics are going to be present (in practice, however, it is assumed to be a known parameter). For mathematical convenience, it is more usual to formulate the harmonic model in terms of complex exponentials with the form $e^{jw_0 l n}$. If the signal samples are arranged into a vector $\mathbf{x} = [x(0), \ldots, x(N-1)]^{\mathrm{T}}$, and the complex amplitudes of the harmonics $A_l e^{j\phi_l}$ are grouped in vector $\mathbf{a} = [A_1 e^{j\phi_1}, \ldots, A_L e^{j\phi_L}]^{\mathrm{T}}$, we can write the harmonic model as

$$\mathbf{x} = \mathbf{Z}\mathbf{a}, \tag{2.9}$$

where $\mathbf{Z}$ is a matrix having a Vandermonde structure, being constructed from L complex sinusoidal vectors as

$$\mathbf{Z} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ e^{jw_0} & e^{jw_0 2} & \cdots & e^{jw_L} \\ \vdots & \vdots & \ddots & \vdots \\ e^{jw_0(N-1)} & e^{jw_0 2(N-1)} & \cdots & e^{jw_0 L(N-1)} \end{pmatrix}. \tag{2.10}$$

We recall that for signals that can be expressed using (2.8) or (2.9), the pitch, i.e., the perceptual phenomenon, and the fundamental frequency are the same. It is interesting to note that, while an harmonic signal is comprised as a sum of a number of individual components, these are perceived as being one object by the human auditory system.

Functions that perfectly obey the harmonic model are *periodic*. In fact, any periodic signal can be decomposed using the model in (2.8) or (2.9). Periodic signals have the following property:

$$x(n) = x(n - \tau), \tag{2.11}$$

or, equivalently, $x(n) = x(n + \tau)$, where τ is the so-called *pitch period*, i.e., the smallest time interval over which the signal $x(n)$ repeats itself, measured in samples. It should be stressed that, while $x(n)$ is defined for integers n, τ is not generally an integer. In fact, since pitch is a continuous phenomenon, it is not accurate to restrict τ to only integer values, although this is often done. The pitch period (in samples) and the pitch ω_0 are each others' reciprocal, i.e., $\omega_0 = 2\pi/\tau$. To express the fundamental frequency in Hz, denoted by f_0, one must use the relation $\omega_0 = 2\pi f_0/f_s$, where f_s is the sampling frequency.

The signals generated by real-world sound sources are not strictly periodic; instead, their cycles are slightly different from each other, and hence we can say that practical signals are indeed *pseudo-periodic* signals. Additionally, in real-world sounds the harmonics do not perfectly match their theoretical values at integer multiples of ω_0; instead, they depart somewhat from their ideal frequencies, a phenomenon designated as *inharmonicity*. Also, in practical situations, the signal is affected by background noise, and hence the model must take into account the presence of a stochastic, non predictable signal term. All these factors will affect our ability to always estimate the fundamental frequency correctly, and so the main challenge of a pitch estimator is to deal with these phenomena in a robust way.

Pitch estimation is then the art of finding ω_0 from an observed signal whose characteristics are not known in detail, and where the signal may depart from periodicity (or harmonicity) in several ways. Many pitch estimation algorithms have

been proposed in the literature. These may be divided into *parametric* and *non-parametric* algorithms. While parametric algorithms assume an explicit model for the noisy signal, for instance, the model in (2.8) for the source part, non-parametric methods do not make such assumptions. At this point, it is worth mentioning that the techniques reviewed here are limited to the single pitch case. In many situations, like in most music or multi-speaker recordings, the signal consists of many periodic sounds, in which case the signal is referred to as *multi-pitch signal*. In this thesis, we do not address problems involving the estimation of multiple fundamental frequencies, even though the model proposed in [P3] is potentially able to represent multiple pitched sources. Anyway, note that some methods reviewed here are extensible to multi-pitch estimation [23].

2.4.1 Non-Parametric Pitch Estimation

Non-parametric algorithms avoid using explicit signal models and identify the pitch of a signal either from its periodicity in the time domain, its harmonic structure in the frequency domain, or from the periodicity of individual frequency bins in the time-frequency domain.

Comb Filtering Method

An intuitive approach for finding ω_0, or, equivalently, τ, is to use the relation in (2.11) directly. To obtain an estimate of τ, we can simply subtract the right-hand side of Equation (2.11) from the left-hand side, i.e., $x(n) - x(n-\tau) = 0$ and then choose the lowest τ for which this expression approximately holds. As stated before, the signal may not be perfectly periodic but may be changing slowly, and there will always be background noise present when dealing with real-life signals. Consequently, the relation in (2.11) is only approximate, i.e., $x(n) \approx x(n-\tau)$, so we can instead measure the difference $e(n)$ as $x(n) - x(n-\tau)$, which we can call the modeling error. To allow for some variation of the amplitude of the signal, we can also include a positive scale factor α close to 1 to account for this variation, so that $x(n) \approx \alpha x(n-\tau)$, and define the modeling error in a more generic way as

$$e(n) = x(n) - \alpha x(n-\tau). \tag{2.12}$$

Taking the z-transform of this expression we get

$$E(z) = X(z) - \alpha X(z) z^{-\tau} = X(z)(1 - \alpha z^{-\tau}). \tag{2.13}$$

From this, we see that the process of matching a signal with a delayed version of itself can be seen as a filtering operation on $x(n)$, where the output of the filter is

the modeling error signal $e(n)$, and the transfer function of the filter is

$$H(z) = \frac{E(z)}{X(z)} = 1 - \alpha z^{-\tau}. \tag{2.14}$$

This mathematical structure is a well-known filter known as the inverse comb filter. Analyzing the filter as a polynomial, we see that it has zeros located at a distance of α from the origin at angles $2\pi k/\tau$, for $k = 1, 2, \ldots, \tau$. Essentially, this filter suppresses signal components at frequencies $2\pi k/\tau$, which correspond to the harmonic positions of a signal with period τ. Consequently, if the filter is tuned to the fundamental frequency of the input signal, the output will be heavily attenuated. In order to use the inverse comb filter to find an estimate of the pitch period, we must apply this filter for several candidate τ values to our observed signal $x(n)$, and then somehow measure how large the output modeling error $e(n)$ is. An usual way of measuring the size of the error is using the mean squared error (MSE), i.e.,

$$J(\tau) = \frac{1}{N-\tau} \sum_{n=\tau}^{N-1} e^2(n). \tag{2.15}$$

This metric is a function of τ, since we will get different errors for different τ values. We then pick as our estimate the τ for which the cost function $J(\tau)$ is the minimum, i.e.,

$$\hat{\tau} = \arg\min_{\tau} \frac{1}{N-\tau} \sum_{n=\tau}^{N-1} \left(x(n) - \alpha x(n-\tau)\right)^2. \tag{2.16}$$

Note that a suitable range over which to compute $J(\tau)$ must be chosen. For speech, this would be τ values corresponding to fundamental frequencies from 60 to 440 Hz (τ from 18 to 133 samples for $f_s = 8$ kHz). It is actually possible to find an optimal $\alpha > 0$ in the sense of the MSE, but it is not that critical, and one can simply choose α to be close to 1 or even 1. The comb filtering approach has a rich history for fundamental frequency estimation and related problems, such as enhancement [96, 85]. In fact, the comb filtering analysis is the theoretical foundation of all methods based on examining periodicity in the time domain (for example, the autocorrelation), which are essentially particularizations or improvements of the estimator in (2.16), as we will see.

Autocorrelation Method

Perhaps the most universally applied principle for pitch estimation is the so-called autocorrelation method, which can be derived based on the comb filtering approach. Suppose that we are inspecting if the signal is perfectly periodic, so that

$\alpha = 1$ in (2.12). In that case, the modeling error for a given τ can be written as

$$e(n) = x(n) - x(n-\tau). \tag{2.17}$$

Inserting this expression into the definition of the MSE in (2.15), we obtain

$$\begin{aligned} J(\tau) &= \frac{1}{N-\tau}\sum_{n=\tau}^{N-1}\left(x(n)-x(n-\tau)\right)^2 \\ &= \frac{1}{N-\tau}\left(\sum_{n=\tau}^{N-1}x^2(n)+\sum_{n=\tau}^{N-1}x^2(n-\tau)-2\sum_{n=\tau}^{N-1}x(n)x(n-\tau)\right). \end{aligned} \tag{2.18}$$

From this, we can make a number of observations. The first term, $\sum_{n=\tau}^{N-1}x^2(n)$, is the power (or variance) of the signal $x(n)$ and does not depend on τ. The second term, $\sum_{n=\tau}^{N-1}x^2(n-\tau)$, which is the power of the signal $x(n-\tau)$, is essentially equivalent to the first term, since we are assuming that the signal is stationary within the segment under analysis, and hence the term can be supposed to be constant with respect τ. The only part that actually changes with τ is:

$$R(\tau) = \frac{1}{N-\tau}\sum_{n=\tau}^{N-1}x(n)x(n-\tau). \tag{2.19}$$

This quantity is known in the signal processing field as the *autocorrelation function* (ACF). It is a function of τ, which is commonly referred to as the *lag* in this context. For $\tau = 0$, we get that $R(0) = \frac{1}{N-\tau}\sum_{n=\tau}^{N-1}x^2(n)$, which is the power of the signal. If the signal is perfectly periodic with period τ, then $R(\tau) = R(0)$. Moreover, it can easily be shown that $R(\tau) \leq R(0)$ for all τ, i.e., the highest possible value of $R(\tau)$ that we can hope to obtain is the same as $R(0)$, which is reached for $\tau \neq 0$ only if the signal is perfectly periodic. Consequently, the autocorrelation function can be seen as a mean to measure the extent to which $x(n)$ and $x(n-\tau)$ are similar, leading to the following estimator:

$$\hat{\tau} = \arg\max_{\tau}\frac{1}{N-\tau}\sum_{n=\tau}^{N-1}x(n)x(n-\tau). \tag{2.20}$$

This estimator is known as the autocorrelation method [95, 48]. It is the most commonly used principle for pitch estimation [126], and many variations of this method has been introduced throughout the years, many of which basically boil down to modifying (2.20) to different measures of the goodness of the fit, which more generally can be written as

$$J(\tau) = \left(\frac{1}{N-\tau}\sum_{n=\tau}^{N-1}\left(x(n)-x(n-\tau)\right)^p\right)^{1/p}. \tag{2.21}$$

For example, the so-called average magnitude difference function (AMDF) can be obtained from this by setting $p = 1$ [108]. Similarly, the YIN algorithm is based on the difference function with a number of modifications to decrease estimation errors [27]. Other methods employ similar principles, even though they do not use the ACF function explicitly. In [31, 93], instead of the ACF, the cross correlation of two adjacent single-period waveform segments is used, giving better time resolution at high pitch frequencies. Autocorrelation-based pitch detectors perform well with a certain level of noise, since the ACF of an aperiodic noise source typically falls off rapidly with lag [121], and the noise can be assumed to be uncorrelated with the source signal. This robustness can be increased by employing subsequent temporal continuity constrains or principles based on auditory analysis. For example, in [97] the ACFs computed on a frame-by-frame basis are stacked into a matrix in which high energy regions are detected. For each region, the system estimates a smooth pitch contour using Dynamic Time Warping (DTW), and the correct contours are selected based on auditory principles.

The ACF function has also been used for measuring periodicity in approaches employing an auditory-based front-end. Instead of taking the ACF of the full-band signal, [109] uses an auditory filterbank to divide the signal into subbands. In each low frequency band the ACF is calculated directly, while in the high frequency bands, which normally include multiple harmonics, the ACF is taken from the signal envelope. The advantage of this multiband approach is that subbands that are dominated by noise (or lack a reliable ACF peak) can be deleted before the subband ACFs are combined to give an overall pitch estimate. This idea has been extended in [140] and later in [69], where multiple pitch candidates are obtained from each frame, and a tracking algorithm based on a HMM is used to find the optimal sequence of zero, one or two sources, thereby implicitly performing voiced/voiceless discrimination.

The YIN Algorithm

The YIN algorithm proposed in [27] describes an extension of the difference function which provides two important advantages: first, a significant improvement in pitch accuracy estimation, and second, a periodicity measure that enables to perform robust voicing detection in clean signals.

The difference function, which can be written as

$$d(\tau) = \sum_{n=1}^{W} (x(n) - x(n+\tau))^2 \tag{2.22}$$

for a window length W, is zero at $\tau = 0$ and often nonzero at the period because of imperfect periodicity. Consequently, unless a lower limit is set on the search

range, the algorithm will consistently choose the zero-lag dip instead of the period dip, thus failing. In general, setting an upper frequency limit to reject erroneous estimations near zero lag is a common problem in all ACF-based methods, and it is difficult to find a successful value for all situations. For example, in speech signals, a strong resonance at the first formant of speech might produce a series of secondary dips, one of which might be deeper than the period dip. A lower limit on the search range is not a satisfactory way of avoiding this problem because the ranges of the formant and F0 are known to overlap. Moreover, as with other methods based on temporal analysis, the difference function is also prone to octave errors, because the integer multiples of the period (and doublings in particular, i.e., half the F0) have sometimes higher influence in the function. The solution proposed by YIN alleviates these problems by following these steps:

1. Replace the difference function by the *cumulative mean normalized difference function*, defined as

$$d'(\tau) = \begin{cases} 1, & \text{if } \tau = 0 \\ d(\tau) \Big/ \left[(1/\tau) \sum_{j=1}^{\tau} d(j) \right], & \text{if } \tau > 0. \end{cases} \tag{2.23}$$

   This new function is obtained by dividing each value of the old function by its average over shorter-lag values. It differs from $d(\tau)$ in that it starts at 1 rather than 0, tends to remain large at low lags, and drops below 1 only where $d(\tau)$ falls below average. The main benefit of this formulation is that avoiding the zero-lag dip is no longer needed, because the function is only zero at the period (and integer multiples) when the signal is perfectly periodic. A second benefit is that the function is normalized, providing an absolute measure of periodicity regardless of the power of the signal. In fact, for a pseudo-periodic signal with period τ, $d'(\tau)$ can be interpreted as the proportion of "aperiodic power" within the total power of the signal. An example illustrating the differences between $d(\tau)$ and $d'(\tau)$ is depicted in Figure 2.4.

2. Set an absolute threshold for selecting the pitch period. As with the ACF function, a problem of $d'(\tau)$ is that higher-order dips may potentially be deeper than the period dip. If a higher-order dip falls within the search range, the result is an octave error, sometimes called subharmonic error in this case. Since the function is normalized, the solution applied by YIN is to define an absolute threshold and choose the smallest value of τ (i.e., the first dip) that gives a minimum of $d'(\tau)$ deeper than this threshold. If none is found, the global minimum is chosen instead. As reported in [27], an

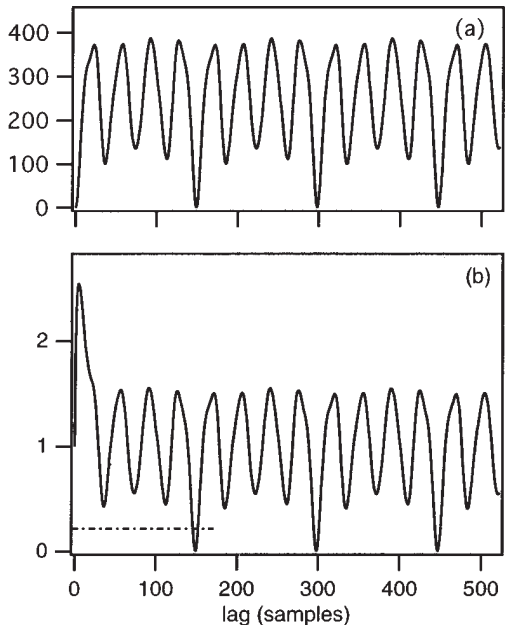

Figure 2.4: (a) Difference function calculated for a speech signal. (b) Cumulative mean normalized difference function. The function starts at 1 rather than at 0, and remains high until the dip at the period.

absolute threshold around 0.1–0.2 reduces considerably the error rate. In addition, the value of the function at the estimated period $d'(\hat{\tau})$, which is called *aperiodicity*, can be interpreted as a robust measure of voicing, so the threshold implicitly provides voiced/unvoiced discrimination. In other words, if the aperiodicity is below 0.1–0.2, the frame can be considered voiced, and unvoiced otherwise. The problem of this threshold to perform voicing detection, however, is that it is extremely sensitive to background noise, and consequently is not reliable in noisy signals. One of the contributions of this thesis [P2] is to compute a robust threshold for the aperiodicity measure that enables to perform voicing detection in noisy conditions.

3. Parabolic interpolation for choosing not integer τ values. To overcome the limitation of the estimator to integer periods, each local minimum of $d'(\tau)$ and its immediate neighbors are fit by a parabola, such that the interpolated minimum is used to select the period dip in the previous step, instead of the original $d'(\tau)$ value. Once the period dip is chosen, the estimated period is then the τ value corresponding to the minimum of the interpolated curve, which now is not necessarily integer.

4. Best local estimate across neighboring frames. In order to ensure that estimates are stable and do not fluctuate, the algorithm choses the best estimate within a small interval around the current frame, where the length of the interval is equal to the largest expected period (for instance, 25 ms for a minimum pitch of 40 Hz). By best estimate, we mean the period across the interval that produces the lower aperiodicity. Based on this initial estimate, the estimation algorithm is applied again with a restricted search range of $\pm 20\%$ of the initial estimate to obtain the final pitch estimation at each frame within the interval.

Algorithms in the Frequency Domain

Non-parametric algorithms operating in the frequency domain typically identify harmonic peaks in the short-time amplitude, log-amplitude or power spectrum. The first step usually consists of detecting sinusoids in the spectrum using a measure of closeness between each local spectral peak's shape and the ideal sinusoidal peak. This detection is often based on the mean square difference between the observed peak and the window main lobe [53], because the width of each peak depends on the window used in the spectral analysis (other factors also intervene, such as the rate of change of pitch). Many methods simply perform peak picking using a fixed amplitude threshold [92], an amplitude-envelope based threshold [39], a psychoacoustical masking threshold [130] or sinusoidal descriptors [143, 17]. The fundamental idea is to discard spectral peaks produced by noise or sidelobes, and select those originated by sinusoids.

In a second step, the identified spectral peaks are compared to the predicted harmonics for each F0 candidate, from which a fitting measure of harmonicity is computed. For instance, the Schroeder's technique [118] measures harmonicity by entering all integer submultiples of the peaks in a histogram. Since the F0 is the integer submultiple of all the harmonics, in an ideal case, the entry with the highest weight in the histogram is the correct F0. In [88] a fitness measure designated as "two-way mismatch" is described, where, for each F0 candidate, mismatches between the theoretical and the detected harmonic frequencies are averaged over a fixed subset of the available partials. The approach in [30] lies in a pair-wise evaluation, in which partials with successive harmonic numbers are identified. The identified pairs are then rated according to harmonicity, timbral smoothness, appearance of intermediate spectral peaks, and harmonic number. The algorithm in [52] convolves the power spectrum in the log-frequency domain with a filter that sums the energy of the F0 harmonics while rejecting the smoother additive noise. Before this filter, a normalization is applied based on a fixed average speech spectrum to remove dependency on the singularities of the speech signal.

Essentially, the common objective of these methods is to define a pitch salience measure for selecting pitch candidates in the frequency domain, often associating a measure of confidence or likelihood with each candidate. This initial frame-by-frame analysis is often followed by post-processing to reduce errors and obtain a smooth contour, for example using hidden Markov models (HMM) or dynamic programming (DP) techniques, such as dynamic time warping (DTW).

2.4.2 Parametric Pitch Estimation

Parametric algorithms for pitch estimation define a parametric stochastic model for the noisy signal in which the pitch, or equivalent, is one of the parameters of the model. The pitch is then estimated by calculating the Minimum Mean Squared Error (MMSE) or Maximum Likelihood (ML) estimate of the model parameters from the observed signal, which involves minimizing (or maximizing) some kind of objective function of the unknown parameters.

As an illustrative example of ML estimation, let us consider the harmonic model expressed in Equation 2.9. The unknown parameters of the model are the fundamental frequency ω_0, the amplitudes of the sinusoids A_l and the phases of the sinusoids ϕ_l, which can be denoted together by $\boldsymbol{\theta} = \{\omega_0, A_1, \phi_1, \ldots, A_L, \phi_L\}$. Assuming that the modeling error $\mathbf{e}$, that is, the difference between the observed signal $\mathbf{x}$ and the model $\mathbf{Za}$, is a colored Gaussian noise, the likelihood function of the observed signal can be written as:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{\pi^N \det(\mathbf{Q})} e^{-\mathbf{e}^{\mathrm{H}}\mathbf{Q}^{-1}\mathbf{e}}, \tag{2.24}$$

where $\mathbf{Q}$ is the covariance matrix of $\mathbf{e} = \mathbf{x} - \mathbf{Za}$ (in principle, this matrix is unknown, so it is another parameter of the model), $\det(\cdot)$ denotes the matrix determinant and $(\cdot)^{\mathrm{H}}$ is the conjugate transpose. Taking the logarithm of this expression, we get the so-called log-likelihood function, i.e.,

$$\zeta(\boldsymbol{\theta}) = \ln p(\mathbf{x}|\theta) = -N \ln \pi - \ln \det(\mathbf{Q}) - \mathbf{e}^{\mathrm{H}}\mathbf{Q}^{-1}\mathbf{e}. \tag{2.25}$$

The maximum likelihood estimates of the parameters are then

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \zeta(\boldsymbol{\theta}). \tag{2.26}$$

The solution to this problem provides the parameter values that are most likely to explain the observed signal, including ω_0. Similarly, by incorporating prior distributions on the parameters, Bayes theorem can be used to obtain a Maximum a Posteriori (MAP) estimate, maximizing the probability $p(\boldsymbol{\theta}|\mathbf{x})$.

The principle of ML estimation is one of the most commonly used in pitch estimation, and estimators based on it are known to have excellent performance for a large number of situations. In general, all parametric estimators assume an inherent statistical behavior, even if this behavior is not explicitly formulated, because the model cannot construct exactly the input signal, and the minimization of any error measure is, in a certain way, a ML estimation. The success of a parametric algorithm will depend on how well the formulated model (with all its practical assumptions) approximates the signal statistically. Parametric approaches have the advantages that the assumptions about the signal are explicit, the limitations of an algorithm are often predictable and the performance can be optimal in a well defined sense. The disadvantage is that the performance may degrade when the, often quite strong, modeling assumptions are not satisfied. A good description of several parametric methods is provided in [23].

Harmonic Summation

The harmonic summation approach to pitch estimation can be derived from the Fourier transform of the model in (2.8). The Fourier transform of a signal $x(n)$ over $n = 0, 1, \ldots, N-1$ is denoted here as $X(\omega)$, for $0 \leq \omega \leq 2\pi$. Let us assume that the observed signal $x(n)$ is not perfectly periodic. In that case, we might think of fitting the model in (2.8) to $x(n)$ and choosing the fundamental frequency ω_0 that best fits. In that regard, we can apply the MSE, as defined in (2.15), to measure the goodness of our model:

$$J(\omega_0) = \sum_{n=0}^{N-1} \left(x(n) - \sum_{l=1}^{L} A_l \cos(\omega_0 l n + \phi_l) \right)^2. \tag{2.27}$$

Equivalently, this MSE can be formulated in the frequency domain by taking the Fourier transform of this expression. It can be shown that, for large N and removing some terms independent from ω_0, this MSE is given by

$$J(\omega_0) = \frac{1}{2\pi} \int_0^{2\pi} |X(\omega)|^2 d\omega - \frac{2}{N} \sum_{l=1}^{L} |X(\omega_0 l)|^2. \tag{2.28}$$

The first term, $\frac{1}{2\pi} \int_0^{2\pi} |X(\omega)|^2 d\omega$, is just the power of the signal $x(n)$ computed in the frequency domain, and is constant with respect to ω_0. The second term, $\frac{2}{N} \sum_{l=1}^{L} |X(\omega_0 l)|^2$, is a summation of the power spectrum evaluated at the harmonic frequencies of a given ω_0. To minimize the MSE, it can be observed that we must maximize this second term, as it is subtracted from the first term. Hence, by measuring $\sum_{l=1}^{L} |X(\omega_0 l)|^2$ for different ω_0, one can obtain an estimate of the

fundamental frequency by picking the value for which the sum is the maximum, i.e.,

$$\hat{\omega_0} = \arg\max_{\omega_0} \sum_{l=1}^{L} |X(\omega_0 l)|^2, \tag{2.29}$$

where the search is often conducted over the audible range of ω_0, such that $\omega_0 \leq \pi/L$. This estimator is the harmonic summation method [98], and indeed it provides the ω_0 that fits best the harmonic model in the MSE sense. An advantage of this approach is that it is robust against many forms of background noise, because the harmonic peaks, which concentrate most of the energy of the signal, usually remain detectable even at poor SNRs. In fact, it can be demonstrated that the harmonic summation method is equivalent to the ML estimator in (2.26) under the assumption of white Gaussian noise. The reason is that, for the white noise case, the covariance matrix of $\mathbf{e}$ reduces to a scaled diagonal matrix $\mathbf{Q} = \sigma^2\mathbf{I}$, where σ is the variance of the error $\mathbf{e}$ and $\mathbf{I}$ is the identity matrix. Substituting into (2.25), the log-likelihood function can now be written as

$$\zeta(\boldsymbol{\theta}) = -N \ln \pi - N \ln \sigma^2 - \frac{1}{\sigma^2} \parallel \mathbf{e} \parallel_2^2, \tag{2.30}$$

where it can be seen that the ML estimator is simply the minimizer of the 2-norm of the modeling error $\mathbf{e}$. As a result, we can formulate the estimator in this way

$$\hat{\omega_0} = \arg\min_{\mathbf{a},\omega_0} \parallel \mathbf{x} - \mathbf{Z}\mathbf{a} \parallel_2^2, \tag{2.31}$$

which essentially is equivalent to minimizing the cost function in (2.27), leading to the same conclusion. The fundamental frequency estimator based on this principle is called the non-linear least-squares (NLS) method, because the fundamental frequency is a nonlinear parameter of the cost function. The harmonic summation is simply an implementation of this theoretical analysis.

It is possible to obtain another method similar to this. Since $\frac{1}{N}|X(\omega)|^2$, according to the harmonic model, is ideally non-zero only for frequencies equal to those of the harmonics, a multiplication of the spectrum evaluated for a set of candidate fundamental frequencies is only non-zero for the true fundamental frequency. This principle can be stated as

$$\hat{\omega_0} = \arg\max_{\omega_0} \prod_{l=1}^{L} |X(\omega_0 l)|^2, \tag{2.32}$$

and we refer to this as the harmonic product method [98].

Many variations of these principles have been proposed in the literature. A harmonic summation method in the log-frequency domain is proposed in [61], in

which the spectrum is shifted along the log-frequency axis, weighted and summed. Following the pitch estimation, frames are classified as voiced or unvoiced based on the correlation coefficient between adjacent pitch periods. In a similar approach, [10] convolves the spectrum in the log-frequency domain with a train of harmonically spaced delta functions and selects the highest peak. Three harmonic summing algorithms for multipitch estimation were described in [78] for music signals; these were later extended in [79] to use an auditory front end which gave a small improvement in some cases.

Other Methods

Many statistical approaches for pitch estimation, specially those based on more elaborate signal models, employ the Expectation Maximization (EM) algorithm. The EM algorithm is an iterative method for ML estimation involving several nonlinear parameters, which guaranties convergence at least to a local maximum. Perhaps, the most illustrative example involving the harmonic model can be found in [22], which is a solution to the ML estimator in (2.26) based on EM. The capabilities of the EM method, however, are preferably exploited for the the resolution of much more complex and accurate models. In [110], for instance, a parametric model for the whole time-frequency power spectrogram of voiced speech is proposed. The power spectrum of each harmonic is modeled as a Gaussian function, while the time evolution of the amplitude of each harmonic is represented as a sum of overlapping Gaussians, the pitch contour being represented as a cubic spline. The noise spectrum is similarly modeled as a sum of overlapping Gaussians on a uniform grid. The EM algorithm determines the ML model parameters, enabling a complete joint parametrization of speech and noise spectrograms. In [49], the instantaneous frequency of each STFT bin is extracted and a statistical model for each harmonic of a source is defined. The EM is used to find the ML estimate of the pitches present in each frame and a multiple agent approach is then used to track the pitch of multiple sources. A closely related method to the EM algorithm, very popular among approaches based on atomic decompositions, is the Harmonic Matching Pursuit algorithm [51]. The algorithm progressively reduces the residual by selecting, at each iteration, the best atom from a dictionary containing versions of a parametric harmonic function. The method is not exactly a ML estimator, but it resembles the iterative structure of EM for optimizing an objective function. Similarly, the nonnegative matrix factorization algorithm can be viewed as a pitch estimator, assuming that the bases represent pitch-related patterns. This aspect will be addressed in Section 3.3.7.

Other statistical methods estimate the pitch contour by exploiting the ability of hidden Markov models to represent the temporal dynamics of the signal. In

this context, a parametric model is used to characterize the shape of the signal at each possible state. An example is found in [139], where the pitch is quantized into discrete values (including an unvoiced state) and a separate GMM is trained to represent the log power spectrum for each pitch possibility. This is then used in a factorial HMM to track the pitch of one or more sources. It was found that the use of speaker-dependent or gender-dependent models improved the tracking performance of multiple speakers significantly.

Another family of parametric methods rely on the principles of subspace orthogonality. In the HMUSIC algorithm [21], the harmonic model is combined with a white noise model and the algorithm simultaneously estimates both the pitch and the number of harmonics present in the signal, based on subspaces decomposition. Despite providing high-resolution pitch estimates, the method requires white noise and is computationally complex.

Methods based on filtering can also be addressed from a parametric perspective. The idea is similar to that of the comb filter described before, but here the constraints of the filter are based on the signal model. The most illustrative technique is the optimal filterbank design [23], where the goal is to find a set of filters that pass power undistorted at harmonic frequencies, while minimizing the power at all other frequencies. This technique is inspired on the signal adaptive filters used in the field of beamforming and direction of arrival estimation.

2.4.3 F0-based Features for Classification

In [112], a set of features derived from F0 estimation were defined for the problem of speech/music discrimination. In particular, the set is composed of seven features, all of them having musical meaning. The YIN algorithm was used for estimating F0, providing three values at each short-time frame: the estimated pitch $F0$, the aperiodicity measure $Ap0$, and a normalized aperiodicity value Ap in the range between 0 and 1.

When dealing with speech signals, the estimated F0 fits to a characteristic pattern for most of the analyzed signals. Speech signals contain voiced frames (near-periodic) and unvoiced frames (aperiodic) which are alternated in short time intervals. In most of languages, words are composed of voiced and unvoiced phonemes, which results in several voiced-unvoiced boundaries within a word. Good estimates of F0 are accomplished for voiced frames, while it does not make sense to estimate F0 for unvoiced frames. Moreover, voiced speech frames have a time-varying F0, because the pitch changes when voiced phonemes are pronounced.

The set of features derived from F0 estimation, defined for a texture window comprising several consecutive short-time frames, is:

1. *Dynamic range of aperiodicity* (D_{Ap}). It is defined as the difference between the maximum and minimum values of the normalized aperiodicity (Ap) within a texture window. Feature D_{Ap} is expressed as follows:

$$D_{Ap} = \max(\mathbf{Ap}) - \min(\mathbf{Ap}), \tag{2.33}$$

$\mathbf{Ap} = [Ap_1, Ap_2, ..., Ap_t, ..., Ap_T]$ being the vector containing the values of the normalized aperiodicity computed for a given texture window, and T the number of analysis windows in the computation interval (texture window).

This feature is intended to discriminate between speech and music when the music signal is either noisy (unvoiced) or voiced during the whole texture window. Speech signals typically alternate voiced frames (low aperiodicity) and unvoiced frames (high aperiodicity) during a texture window. Typically, speech signals show high dynamic range of aperiodicity in the computation interval, while music signals tend to provide lower values.

2. *Average of the estimated F0* ($F0_{av}$). It is defined as the mean value of the F0 estimated at the current texture window.

Before computing this feature, the estimated F0 (in Hz) is converted to octaves. In this way, the logarithmic behavior of the ear is taken into account. It is assumed that octave 0 coincides with 440 Hz (note A in 4th scale), and the octave at the t-th analysis window is computed as follows:

$$O_t = log_2(F0_t) - \log_2(440). \tag{2.34}$$

Here, 27.5 Hz and 7040 Hz are the minimum and maximum values that can estimated for the fundamental frequency. These frequencies correspond to octaves -4 and 4, respectively. Therefore, computation of feature $F0_{av}$ is performed as follows:

$$F0_{av} = \frac{\sum_{t=1}^{T} O_t}{T}, \tag{2.35}$$

where T is the number of analysis windows at each texture window. Speech signals have a typical pitch range which goes from -2.5 to -1 octaves.

3. *Dynamic range of estimated F0* (D_{F0}). It is defined as the difference between the maximum and minimum values of the estimated F0 within the current texture window. Feature D_{F0} is expressed as follows:

$$D_{F0} = \max(\mathbf{O}) - \min(\mathbf{O}), \tag{2.36}$$

$\mathbf{O} = [O_1, O_2, ..., O_t, ..., O_T]$ being the vector containing the values of the fundamental frequency (expressed in octaves) computed for a given texture window.

In speech signals, speaker's intonation makes the estimated F0 varies in a typical range (within an octave). Further, noisy speech frames are sometimes labeled as voiced, the estimated F0 being very high. In these cases, feature D_{F0} is very high in the current texture window.

4. *Maximum note duration* (ND_{max}). It is defined from the number of consecutive analysis windows comprising the longest musical note within the observation interval (the current texture window). Therefore, computation of the musical note corresponding to the each analysis window from the estimated F0 must be first addressed. The musical note at the t-th analysis window is here computed as follows:

$$Note_t = \lfloor 12 \cdot (O_t + 4) + 0.5 \rfloor + 1. \tag{2.37}$$

   In this way, since octaves range from -4 to 4, musical notes are ordered from 1 to 96. To understand equation (2.37), note that 12 consecutive semitones represent an octave. Once all musical notes in the current texture window have been computed, feature ND_{max} is obtained from the longest time interval containing the same musical note.

5. *Number of notes* (N_{note}). This parameter is defined as the number of different notes contained within the observation interval (the current texture window). From the fundamental frequencies estimated in the observation interval, we compute how many different notes are detected. For speech signals, it is common to obtain high values of parameter N_{note} (around 6 notes), because the estimated F0 slowly changes with the speaker's intonation. On the other hand, lower values of parameter N_{note} (around 2 notes) are usually obtained for music signals, because the estimated F0 remains steady in variable duration intervals.

6. *Voiced ratio* (VR). It is defined as the ratio between the number of voiced frames and the total number of frames within the observation interval. This parameter informs us about the percentage of frames in which F0 is properly estimated at each observation interval. It can be expressed as follows:

$$VR = \frac{N_{voiced}}{T} \tag{2.38}$$

   where N_{voiced} is the number of frames that fulfil the following condition: $Ap0_t \leq 0.2$. $Ap0_t$ is the aperiodicity at the t-th analysis window of the current texture window.

   Generally, speech signals have a balanced ratio of voiced frames (parameter VR usually ranges from 0.3 to 0.6). However, the ratio of voiced frames

tends to be small for music signals (parameter VR is very often below 0.2), because polyphonic frames are usually labeled as unvoiced.

7. *Average value of the aperiodicity* (Ap_{av}). Mean value of the normalized aperiodicity at the current texture window. It is only defined for voiced frames, and can be expressed as follows:

$$Ap0_{av} = \frac{\sum_{t \in V} Ap_t}{N_{voiced}}, \tag{2.39}$$

where set V is composed of those frames that fulfil the following condition: $Ap_t \leq 0.2$.

Parameter Ap_{av} usually ranges from 0.08 to 0.12 for speech signals. In voiced speech frames, vocal folds are most of the time vibrating, and the aperiodicity is typically around 0.1. Feature Ap_{av} has good discrimination capability due to its different behavior for speech and music.

Chapter 3

Speech Enhancement

3.1 Introduction

Numerous approaches for single-channel speech enhancement have been developed over the last decades. A number of speech enhancement algorithms operate in the time domain and typically use adaptive filters or Kalman filters [47]. The majority of algorithms, however, perform the enhancement in the frequency domain, in which the speech signal is sparse and therefore is more easily separable from background noise.

The speech enhancement problem can be formulated as an audio source separation problem, in which, given the noisy speech signal

$$x(n) = s(n) + b(n), \tag{3.1}$$

the clean speech signal $s(n)$ must be isolated from the background noise $b(n)$. This is generally an ill-posed problem, and its success rate heavily depends on the appropriate characterization of one or both source signals. In the frequency domain, the mixture can be expressed as $X(t, f) = S(t, f) + B(t, f)$, where f is the frequency bin index and t is the time frame index.

Two main approaches can be distinguished for speech enhancement: *filter-based methods* retrieve the clean speech spectrum based on a previous estimation of the noise spectral power (or, equivalently, of the SNR in each time-frequency point). That is, given an estimate of $|B(t, f)|$, they try to derive a filter to retrieve $|S(t, f)|$ according to a certain criterion. On the other hand *model-based methods* formulate parametric models for speech and possibly for background noise, and try to estimate their parameters jointly. Then, the estimated speech model is the resulting enhanced speech. In this thesis, we will work with compositional models and matrix decomposition for speech and noise separation.

3.2 Filter-based Enhancement Algorithms

These methods perform the enhancement of speech segments using an adaptive filter, which, in most cases, is based on noise power and SNR estimates. The most prominent techniques are briefly discussed below. All of these methods can be cast in a spectral modification framework, which achieves noise reduction through the application of a spectral gain function. In other words, the objective is to calculate a filter $G(t, f)$, such that

$$|S(t, f)| \approx |X(t, f)|G(t, f), \tag{3.2}$$

where an estimate of $|B(t, f)|$ (or the SNR) is given.

3.2.1 Spectral Subtraction

The earliest approach for enhancing speech degraded by noise is the power spectral subtraction (PSS) method introduced in [9] and [6], whose principle is to subtract the short-time power spectral magnitude of noise from the noisy-speech power magnitude, assuming uncorrelated and additive noise. The noise spectrum is usually estimated during speech pauses, detected by a VAD, using first-order recursive noise power and signal power estimates:

$$P_X(t, f) = \alpha_X P_X(t-1, f) + (1-\alpha_X)|X(t, f)|^2 \tag{3.3}$$

$$P_B(t, f) = \alpha_B P_B(t-1, f) + (1-\alpha_B)|B(t, f)|^2, \tag{3.4}$$

where the smoothing parameters α_X and α_B are typically in the range $0 \leq \alpha_X \leq 0.5$ and $0.5 \leq \alpha_B < 1$. $P_X(t, f)$ and $P_B(t, f)$ denote the estimates of the power of signal and noise, respectively. Since these short-time estimates are subject to random fluctuations, a simple subtraction of estimated powers may yield negative results. Thus, a limitation is necessary, and an estimate of the clean speech power may be obtained via

$$\begin{aligned} |\hat{S}(t, f)|^2 &= \max(P_X(t, f) - P_B(t, f), 0) \\ &= P_X(t, f) \max\left(1 - \frac{P_B(t, f)}{P_X(t, f)}, 0\right) = P_X(t, f)|G_{\text{SS}}(t, f)|^2, \end{aligned} \tag{3.5}$$

where the $\max(\cdot)$ function guarantees nonnegative results. The spectral subtraction method can be interpreted as a time-variant linear filter with magnitude response

$$G_{\text{SS}} = \sqrt{\max\left(1 - \frac{P_B(t, f)}{P_X(t, f)}, 0\right)}. \tag{3.6}$$

Since we subtract in the power spectral density domain, this approach is called *power subtraction*. Many variations of this basic principle have been proposed, such as the *magnitude subtraction*

$$|\hat{S}(t,f)| = \max(\sqrt{P_X(t,f)} - \sqrt{P_B(t,f)}, 0)$$
$$= \sqrt{P_X(t,f)} \max\left(1 - \frac{\sqrt{P_B(t,f)}}{\sqrt{P_X(t,f)}}, 0\right) = \sqrt{P_X(t,f)}|G_{\text{SS}}(t,f)|. \quad (3.7)$$

Spectral subtraction techniques typically achieve a fairly good speech quality. However, the residual noise after processing is characterized by many spectral outliers. These outliers appear randomly in all spectral bins and generate short sinusoidal tones when synthesizing the output signal in the time domain. In listening experiments, these random fluctuations are perceived as rapid fluctuations, also known as *musical noise*. As a consequence, the processed signal may not have enough quality for certain applications. Nevertheless, despite this problem, PSS is perhaps the most popular algorithm for speech enhancement used today, thanks to its low complexity and high efficiency.

The multiple improvements proposed in the literature basically attempt to reduce the output musical noise or, at least, its subjective perception. In [132], the PSS method is modified including a human hearing model based on the masking phenomenon commonly used in audio coding. The subtraction parameters are continuously adapted according to the noise masking threshold, obtaining a significant reduction of the perceived noise. Other improvements are based on the observation that the spectrum of real-world noise is not flat, which implies that the noise does not affect the speech signal uniformly over the whole spectrum. Based on this fact, several implementations propose a non-linear spectral subtraction, with different subtraction parameters for different frequency bands. An example is found in [72], where a multi-band spectral subtraction technique for colored noise is proposed. The authors propose to split the frequency spectrum linearly into a number of non-overlapping bands. A traditional spectral subtractor with a different over-subtraction factor is applied to each band. They found that four is the optimal number of bands in terms of speech quality. The algorithm notably outperforms the original algorithm for different SNRs.

3.2.2 Wiener Filtering

Another common approach for single-channel noise reduction is the application of the Wiener filter [137], which is an optimal estimator of the desired signal in the minimum mean-square error (MMSE) sense. The Wiener filter was originally formulated in the time domain and assumes wide-sense stationary input signals.

The coefficients $h(k)$ of the optimal Wiener filter are the solution to this problem

$$h(k) = \arg\min_{h(k)} \mathrm{E}\left\{ \left(s(n) - \sum_{k=-\infty}^{\infty} h(k)x(n-k) \right)^2 \right\}, \tag{3.8}$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator. This solution is obtained in [131] and, as seen, is an infinite-impulse response (IIR) filter requiring infinitely long stationary signals, therefore not realizable. A practical "Wiener" gain function inspired by the Fourier transform of the IIR Wiener filter may be computed to enable a frame-by-frame DFT processing. This corresponding Wiener filter is expressed as

$$G_{\mathrm{W}}(t,f) = \frac{\sigma_S^2(t,f)}{\sigma_S^2(t,f) + \sigma_B^2(t,f)} = \frac{\xi(t,f)}{1+\xi(t,f)}, \tag{3.9}$$

where $\sigma_S^2(t,f)$ is the speech power, $\sigma_B^2(t,f)$ is the noise power and $\xi(t,f)$ is the a priori SNR. The Wiener gain depends only on $\xi(t,f)$, which must be estimated using SNR estimation techniques.

Various works in the literature propose enhancers based on Wiener gain. The work in [114] presents a least mean-square adaptive filtering approach that exploits the quasi-periodic nature of the speech waveform to supply a reference signal to the adaptive filter. The method has the advantage of not requiring a priori knowledge of the properties of the noise signal. The Wiener filter can also be estimated iteratively by assuming an all-pole model for speech production. The iterative Wiener filter was originally formulated in [86]. In this technique, the speech signal is modeled as the response of an all-pole system, and the approach solves the maximum a posteriori estimate of the speech signal given the noisy signal. Unfortunately, the convergence criteria is not specified.

Although the Wiener filter generally achieves satisfactory noise reduction, it also introduces distortions that can be perceptually unacceptable for very low SNRs. In [18], the relationship between noise reduction and speech distortion with the single-channel Wiener filter is formally studied. The authors demonstrate that the level of noise attenuation is proportional to the level of speech degradation, and a trade-off should be adopted depending on the application.

3.2.3 Nonlinear MMSE Estimation

Statistical model-based algorithms rely on the MMSE estimation of the short-time spectral amplitudes $|S(t,f)|$, commonly by assuming that speech and noise amplitudes are independent Gaussian random variables. The approach, however, is flexible enough to propose solutions assuming non-Gaussian densities, or to define functions $c(|S(t,f)|)$ of the DFT amplitudes as the MMSE estimation target.

The use of functions enables to achieve a better fit to the observed probability distributions, or introduce perceptually more meaningful error measures.

In [35] the authors derive the optimal MMSE short-time spectral amplitude estimator (i.e., the one that minimizes $\mathrm{E}\{(|S(t,f)| - |\hat{S}(t,f)|)^2\}$) for the Gaussian case. Its performance is compared with the Wiener filter, resulting in a significant reduction of the noise and providing enhanced speech with colorless residual. Also, they observe that the Wiener filter is optimal in the sense of MMSE signal spectral estimation, but is not an optimal spectral magnitude estimator under the Gaussian assumption. The same authors further extend their algorithm in [36], where they minimize the MSE of the log-spectral amplitude, that is, $\mathrm{E}\{(\log(|S(t,f)|) - \log(|\hat{S}(t,f)|))^2\}$. The resulting gain function is given by

$$G_{\text{MMSE}}(t,f) = \frac{\xi(t,f)}{1+\xi(t,f)} \exp\left(\frac{1}{2}\int_{v(t,f)}^{\infty} \frac{e^{-z}}{z}\,\mathrm{d}z\right), \tag{3.10}$$

where $\xi(t,f)$ is the a priori SNR and $v(t,f)$ is defined as

$$v(t,f) = \frac{\xi(t,f)}{1+\xi(t,f)} \cdot \frac{\sigma_X^2(t,f)}{\sigma_B^2(t,f)}. \tag{3.11}$$

This gain function improves the estimation of small amplitudes, because the error measure, based on the logarithmic operation, places more emphasis on small values. Small speech amplitudes are very important for speech intelligibility, and indeed this estimator is reported to provide improved perceived quality.

In many situations, it turns out that the assumption of Gaussian probability is inappropriate. In [91], the probability density function of speech coefficients is modeled by a complex Laplacian or by a complex bilateral Gamma, and the probability of noise coefficients is either modeled by a complex Gaussian or complex Laplacian. This estimator obtains higher noise reduction than the traditional MMSE estimator, and the residual noise is lower when the input noise follows a Laplacian density.

MMSE estimators have been sometimes preferred over spectral subtraction, partly because they have shown to be successful in eliminating musical noise even with poorly stationary noise [16]. The reason of this reduction is the low variance estimate of the obtained spectra. In general, although they significantly reduce the noise level, still have the disadvantage of requiring an estimate of the a priori SNR.

3.2.4 Binary Masks

The time-frequency gain function applied by the above methods contains continuous values, being often referred to as a *soft mask* in the context of source

separation. A *binary mask*, in contrast, is a gain function that takes one of two values, 1 and 0. Then, the approach is not to construct the closest possible version of the original speech, but simply to select the correct time-frequency bins. The most widely used goal is to estimate the so-called Ideal Binary Mask (IBM) [107], which is defined as the mask in which the time-frequency bins dominated by the target signal are set to 1.

There are several reasons for using binary masks. First, it is not strictly necessary to make an estimation of the noise and/or speech. In fact, the enhancement problem can be addressed as a classification problem, in which the goal is to decide whether to retain a particular bin. Second, it has been shown experimentally that the IBM provides perfect intelligibility [77]. Further, if the mask selects the appropriate speech bins, the result is fully understandable independently of the background noise. Third, within selected time-frequency regions, the mask does not introduce artifacts, making the results very attractive for applications such as speech recognition.

In practice, it is only possible to obtain a binary mask that is just an approximation of the IBM. Two strategies have been followed for the estimation of the IBM. The first one is based on computational auditory scene analysis (CASA), which comprises all those processing techniques aiming to mimic the behavior of the human auditory system. The majority of the CASA models [11] are based on a time-frequency representation of the signal with a cochleagram, and perform the separation following three steps: segmentation, grouping and masking. In the segmentation step, the system identifies zones in the cochleagram whose time-frequency points are likely to have a common origin. In the grouping step, the identified zones are grouped into actual sound sources. Finally, a binary mask is applied to segregate the sources. For segmentation and grouping, CASA-based algorithms employ features such as periodicity across frequency, common onsets and offsets, pitch or common amplitude and frequency modulations. The main problem of this approach is the correct estimation of these features in noise, for instance, the pitch contour and voicing state. One of the earliest approaches for voiced speech segregation was proposed in [102], based on pitch estimation and harmonic selection. A simple harmonic binary mask is sufficient to obtain improved intelligibility measures, as we corroborate in [P2], assuming a way to perform robust pitch and voicing estimation. More elaborate segmentation and grouping procedures have been proposed in [65].

A second strategy to estimate the IBM is the use of classification techniques to identify points as either speech-dominated or noise-dominated. A speech enhancer using binary masks and classifiers was introduced in [74, 73]. The classification of each time-frequency cell was on the basis of the likelihood ratio of two GMMs trained respectively on training data cells whose local SNR was above

and below a threshold. For each frequency channel, a feature vector comprising modulation spectrum measures for that channel was extracted. Binary masks for enhancement have also been estimated using Support Vector Machines [56], deep belief networks [135] and sparse coding techniques [81].

3.2.5 Noise Power Spectrum Estimation

Most noise-reduction algorithms require an estimate of the background noise power spectrum or, equivalently, the SNR at each time-frequency bin. The accuracy of the noise estimation has a major impact on both the quality and intelligibility performance of the processed speech. When approaching the problem of noise estimation, the following assumptions are often made:

- The noise signal is more stationary than the speech signal.
- Speech and noise are statistically independent.
- Voiced speech is harmonic and the noise spectrum is relatively flat.

The first noise estimation approaches used Voice Activity Detector (VAD) estimators to identify noise-only intervals. The noise could be then calculated by a temporal average during the speech absences using an averaging time-constant that depends on the assumed stationarity of the noise.

A minimum statistics approach was introduced to estimate the noise in [90, 89]. The basis of this approach is that over a given time-interval there will be pauses in the speech in every frequency band and consequently the minimum value of the noisy speech spectrum within a frequency band will correspond to the noise power.

The noise power spectrum can also be calculated by using a Minimum Mean Squared Error (MMSE) estimator. In [58], an MMSE estimator was used to minimise the power of the difference function between the estimated and the true noise power spectrum. This algorithm was found to perform best in a comparative evaluation of several noise estimation algorithms in [125]. The work in [58] has been further extended in [46], where a soft decision Speech Presence Probability (SPP) was used to update the noise adequately. While decreasing the computational complexity of the original algorithm, the estimation accuracy was maintained.

The harmonic tunneling technique makes use of the harmonic structure of the voiced speech spectrum [34]. This estimate of the noise is obtained by sampling the noise spectrum in the gaps (or "tunnels") between the harmonic spectral peaks, requiring accurate pitch and voicing estimation.

3.3 Compositional Models for Speech Enhancement

3.3.1 Introduction

Many types of data can be represented as constructive combinations of parts, that is, as combinations that are strictly additive, where none of the parts produces subtraction. These data are often referred to as *compositional data*, and the mathematical models used to represent them are called *compositional models*. These models take the form of nonnegative linear combinations of parts which are also nonnegative, ensuring that the combination is purely constructive.

The motivation for applying compositional models to audio processing is that sound can be viewed as compositional data as well. Although time domain signals are not nonnegative and may occasionally cancelate each other, concurrent sources are approximately additive in the spectral domain. Furthermore, even the sound produced by a single source is often the combination of more elementary sounds. For example, the sound produced by a piano is composed of its individual musical notes. Similarly, the noise generated by a machine can be viewed as the contribution of all of its sounding mechanisms. The assumption of additivity is specially true for sparse signals in the time-frequency domain, where the energy of each component is concentrated on a limited amount of bins.

Another essential aspect of compositional models is that the number of parts of the model is limited, and much lower than the number of possible instances of the data. These parts can be viewed as building blocks of the data which are able to construct any observation. In the case of audio, this means that each individual source can be modeled with a limited number of patterns which are assumed sufficient for composing any possible sound of the source. A clear example of this can be viewed in music signals. A musical performance is a compositional mixture, in which the basic patterns are the notes from various instruments. For speech and noise signals, this idea is not as intuitive, because these sources cannot be apparently constructed from basic patterns. However, as we will see, compositional models can be used to perform *sparse approximations*, where a slightly differing instance of a sound is represented as a sparse combination of its nearest patterns.

A key motivation for using this approach in audio processing is the existence of mathematical tools for decomposing an input signal into useful constructive parts. Two techniques exist for this purpose: *non-negative matrix factorization* (NMF) and *probabilistic latent component analysis* (PLCA). NMF models [133, 40] treat non-negative time-frequency representations of the signal as matrices, which are decomposed into products of non-negative component matrices. Some of these matrices represent the spectral patterns, and others their respective activation in the signal over time. The PLCA models treat the non-negative

time-frequency representations as the result of an stochastic process, in which the atomic units of the signal are represented as probability density functions [120]. In practice, the two approaches are almost equivalent, and in fact arithmetically identical under some circumstances [29]. In this thesis, we focus on algorithms based on NMF. Other approaches derived from NMF, such as *nonnegative tensor factorization*, for multichannel signals [42], or *nonnegative matrix deconvolution*, based on convolutive patterns [123, 116], are not treated in this thesis either.

The decomposition of signals using these techniques has given rise to new solutions for several audio processing problems, specially for source separation and signal enhancement [123, 133]. In music applications, the techniques are usually employed for extracting or suppressing specific instruments from mixed tracks [55, 76]. In speech processing, the objective is usually segregate the target speech from other sources, generally noise, with the aim of improving objective quality or intelligibility [4, 105, 124, 138, 70]. Another notable application of NMF is related to classification purposes [19]. In a sense, the separation performed by compositional models functions like a classifier, because they select components belonging to single-source sets which indeed give information about the nature of the signal. In music applications, these techniques have been used for instrument recognition [111], genre classification [101], automatic transcription [15, 83, 122, 57, 7] and coding [100, 103]. In speech processing, NMF analysis has been found useful for speech recognition [44, 43] and speaker identification [128]. In the case of speech, these approaches can be used also for pitch estimation, because the patterns used for decomposing the signal can represent excitation signals corresponding to different pitch candidates [33]. Finally, NMF can also perform model learning, that is, given a certain compositional model, NMF can learn the patterns of the model from a set of training data. The topic of model learning is essential for NMF-based analysis algorithms, because the learned patterns can be used to decompose a new observation in a useful way.

The application of compositional analysis to audio processing involves the two following important tasks:

- Formulating an appropriate compositional model that represents the generation of the observed mixture.
- Designing a decomposition algorithm for estimating the parameters of the model.

In this thesis, we focus on the formulation of signal models for speech and noise signals, in conjunction with decomposition algorithms that enable to find meaningful patterns for the model [P3].

3.3.2 Signal Representation

As mentioned above, audio signals can be considered as constructive data in the frequency domain. This affirmation has a theoretical basis: the power spectum of the sum of uncorrelated signals is the sum of the power spectra of the individual signals. In addition, when applying NMF for source separation, it is necessary to point out some important aspects.

In our problem, the observed noisy signal $x(n)$ is the instantaneous sum of two contributions: an utterance produced by a single speaker $s(n)$ and the interfering noise $b(n)$. For the application addressed in this thesis, the background signal $b(n)$ can be composed by any combination of noises commonly found in daily life. Other types of interference, such as music accompaniment or concurrent speakers, are not considered, unless they appear as noise (for instance, babble noise or music present in the noisy environment). Consequently, the problem of separating $s(n)$ and $b(n)$ can be viewed as a typical speech enhancement problem, where little assumptions are made about the noise.

In NMF-based separation methods, sound sources are represented through their spectro-temporal distribution $[\mathbf{S}]_{ft} = s_{ft}$ and $[\mathbf{B}]_{ft} = b_{ft}$, where s_{ft} and b_{ft} are, respectively, the f^{th} frequency element of speech and noise in frame t. Although different time-frequency representations can be employed, the chosen representation must fulfill two requirements:

- each element must be non-negative,
- it must be invertible, in order to retrieve the separated signals in the time-domain.

The magnitude spectrogram or the power spectrogram, computed directly through the STFT, are the most common representations, although spectral distributions based on a logarithmic frequency scale are also usually employed.

The observed mixture $\mathbf{X}$ can be expressed as

$$\mathbf{X} = \mathbf{S} + \mathbf{B}. \tag{3.12}$$

Note that this expression assumes that the sources combine additively in the chosen representation (as mentioned, in the case of magnitude and power spectrograms, this is true if speech and noise are statistically independent). Since this framework neglects any phase information, it is not possible to estimate the phase of individual sources. Typically, once $\mathbf{S}$ and $\mathbf{B}$ have been estimated, each source is synthesized with the same phase as the mixture. This approach produces good results for separation, since the auditory system is quite insensitive to phase.

3.3.3 Basic NMF Model

The most simple compositional model represents the observed spectrogram as a non-negative linear combination of atomic units (which we will simply refer to as *bases*). In its simplest form, these bases are spectral vectors, representing steady-state sounds, such that any spectral vector in the input spectrogram can be decomposed into a non-negative linear combination of these bases.

Let $\mathbf{w}_k$ represent the set of basis vectors, indexed by $k = 1, \ldots, K$, where K is the total number of bases. Each spectral vector $\mathbf{x}_t$ at time instant t can be expressed as a linear combination of the elementary bases, in the form

$$\mathbf{x}_t = \sum_{k=1}^{K} \mathbf{w}_k h_{kt}, \tag{3.13}$$

where h_{kt} is the non-negative *activation* of the kth basis in frame t. Consequently, the given spectrogram is modeled as a time-varying combination of a certain set of bases $\mathbf{w}_k$. Observe that the set of bases $\mathbf{w}_k$ can be viewed as a dictionary of building blocks, from which any instance of the spectrogram can be constructed. The time-varying activations weights h_{kt} determine how the bases must be combined to approximate the observed data at each time instant

We can arrange all of the bases $\mathbf{w}_k$ as columns of a matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$. Similarly, we can arrange the activation coefficients h_{kt} as elements of the matrix $[\mathbf{H}]_{kt} = h_{kt}$ The composition of $\mathbf{X}$ in terms of the basis vectors and their activations can now be written as

$$\mathbf{X} \approx \mathbf{WH}, \tag{3.14}$$

where all entries are strictly non-negative.

In order to decompose the signal into the product $\mathbf{WH}$, we must determine the $\mathbf{W}$ and $\mathbf{H}$ that together achieve the best approximation to $\mathbf{X}$. To do so, we define a scalar-valued divergence $D(\mathbf{X}||\mathbf{WH})$ between the observed spectrogram $\mathbf{X}$ and the reconstruction $\mathbf{WH}$, which measures the error between the two. It is assumed that the minimum value of the divergence is zero, which is only reached for perfect reconstruction, i.e., $\mathbf{X} = \mathbf{WH}$. Typically, the divergence is calculated entry-wise, i.e.,

$$D(\mathbf{X}||\mathbf{Y}) = \sum_{f,t} d(x_{ft}, y_{ft}), \tag{3.15}$$

where $d(\cdot)$ is a divergence measure between two scalars.

The optimal values $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$ of $\mathbf{W}$ and $\mathbf{H}$ are obtained by minimizing this divergence:

$$\hat{\mathbf{W}}, \hat{\mathbf{H}} = \arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{X}||\mathbf{WH}) \quad \mathbf{W} \succeq 0, \mathbf{H} \succeq 0. \tag{3.16}$$

Here we have assumed that both the basis vectors $\hat{\mathbf{W}}$ and their activations $\hat{\mathbf{H}}$ are unknown, and must be estimated from the decomposition. If $K < T$, where T is the number of frames (i.e., the number of observed vectors), the problems becomes overdetermined, and solving the decomposition leads to find a compressed approximation of $\mathbf{X}$ with a reduced number of components, potentially revealing its underlying structure. In fact, if $\mathbf{X}$ is naturally composed as a combination of repetitive and distinguishable patterns, this decomposition will tend to find these patterns in $\mathbf{W}$, where $\mathbf{H}$ will find how these patterns have combined to produce $\mathbf{X}$.

Commonly, the dictionary of basis vectors $\mathbf{W}$ is known a priori (for instance, when it has been learned in a training phase), such that the decomposition only requires to estimate the activations:

$$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} D(\mathbf{X}||\mathbf{W}\mathbf{H}) \quad \mathbf{H} \succeq 0. \tag{3.17}$$

A similar solution may also be defined when $\mathbf{H}$ is known and $\hat{\mathbf{W}}$ must be obtained.

The most common divergence $d(\cdot)$ in matrix decomposition problems is the (squared) Euclidean distance (EUC), expressed as

$$d_{\text{EUC}}(x, y) = (x - y)^2, \tag{3.18}$$

However, in the context of audio modeling, other divergence measures have been found more appropriate [133, 15, 136]. The Euclidean distance emphasizes errors in high-energy components, which leads to solutions where only high-energy bins are accurately represented. This supposes an important problem for audio analysis, because audio signals typically have a large dynamic range, and some low-energy components (often in higher frequencies) are perceptually as important as high energy components.

Divergence measures that assign greater emphasis to low-energy components are required for audio. Two common alternatives are the generalized Kullback-Leibler (KL) divergence

$$d_{\text{KL}}(x, y) = x \log(x/y) - x + y, \tag{3.19}$$

and the Itakura-Saito (IS) divergence

$$d_{\text{IS}}(x, y) = x/\hat{x} - \log(x/\hat{x}) - 1. \tag{3.20}$$

Unlike the EUC divergence, the IS divergence assigns the same importance to high and low energy components, because it is scale invariant. The KL divergence provides a good compromise between the two [133, 15, 136]. A generalization of

the above divergences is the beta-divergence [41], which is defined as a function of a parameter β:

$$d_\beta(a,b) = \begin{cases} \frac{1}{\beta(\beta-1)}(a^\beta + (\beta-1)b^\beta - \beta a b^{\beta-1}), & \beta \in \Re^+ \setminus \{0,1\} \\ a\log\frac{a}{b} - a + b, & \beta = 1 \\ \frac{a}{b} - \log\frac{a}{b} - 1, & \beta = 0. \end{cases} \tag{3.21}$$

As can be seen, the EUC distance ($\beta = 2$), the KL divergence ($\beta = 1$) and the IS divergence ($\beta = 0$) are particular cases of this measure.

The EUC and KL divergences, supposing a fixed x, are convex as a function of y. Similarly, for these divergences, the function $D(\mathbf{X}||\mathbf{Y})$ is also convex in $\mathbf{Y}$, supposing a constant $\mathbf{X}$. In this case, the optimization problem expressed in (3.17), in which $\mathbf{H}$ must be estimated and $\mathbf{W}$ is fixed, consist of minimizing a convex function, and consequently, it can be solved by any convex optimization technique. However, when both $\mathbf{W}$ and $\mathbf{H}$ have to be estimated, as expressed in the problem in (3.16), the function $D(\mathbf{X}||\mathbf{WH})$ becomes biconvex in $\mathbf{W}$ and $\mathbf{H}$. This means that it is not jointly convex in both of these variables, but only convex in each one individually, assuming the other fixed. Therefore, the problem in equation (3.16) cannot directly be solved through convex optimization methods. Nevertheless, convex optimization methods may still be employed by alternately estimating $\mathbf{H}$ given $\mathbf{W}$, and then estimating $\mathbf{W}$ given $\mathbf{H}$, repeating the process until convergence is reached. Other divergences, such as the IS divergence, are not convex, and minimizing the objective function requires more carefully designed optimization algorithms than the convex divergences [40].

The most famous technique for non-negative decompositions is based on the so called *multiplicative update rules*, initially proposed by Lee and Seung [82]. The parameters to be estimated are first initialized to random positive values, and then iteratively updated by applying a multiplicative gradient. In contrast to other gradient-based methods, the multiplicative rules have the advantage that the step size is chosen automatically at each iteration, and the nonnegativity of the parameters is preserved, because both the gradient and the parameters are nonnegative. The multiplicative updates that decrease the beta-divergence are given as

$$\mathbf{H} \leftarrow \mathbf{H} \bullet \frac{\mathbf{W}^{\mathrm{T}}\left(\mathbf{X} \bullet \mathbf{Y}^{\beta-2}\right)}{\mathbf{W}^{\mathrm{T}}\mathbf{Y}^{\beta-1}} \tag{3.22}$$

and

$$\mathbf{W} \leftarrow \mathbf{W} \bullet \frac{\left(\mathbf{Y}^{\beta-2} \bullet \mathbf{X}\right)\mathbf{H}^{\mathrm{T}}}{\mathbf{Y}^{\beta-1}\mathbf{H}^{\mathrm{T}}} \tag{3.23}$$

where $\mathbf{Y} = \mathbf{WH}$, the symbol $\bullet$ denotes element-wise matrix product, and all the divisions are element-wise. It can be easily seen that if $\mathbf{W}$ and $\mathbf{H}$ are non-negative,

the terms that are used to update them are also non-negative. This optimization scheme can be shown to be non-decreasing with respect to the beta-divergence, and able to reach a local minimum after a few iterations.

In addition to multiplicative updates, a variety of alternative methods have been proposed, based on e.g. second-order methods [141], projected gradient, etc. The methods can also be accelerated by active-set methods [75, 134]. There also exist divergences that aim at optimizing the perceptual quality of the representation [100], which is useful in audio coding applications. In most of the other applications of compositional models such as source separation and signal analysis, however, the quality of the representation is affected more by its ability to isolate latent compositional units from a mixture signal, not the ability to represent accurately the observations. Therefore, simple divergences such as the KL or IS are the most commonly used even in the applications where a mixture is separated into parts for listening purposes.

3.3.4 Source Separation

Separation of audio signals into their individual sound sources is probably the most immediate application of compositional modeling. Essentially, it is assumed that any sound source in the mixture has its characteristic basis functions. The observed mixture is then composed of atoms from the individual sources, such that the separation of any particular source only requires the segregation of the contribution of its bases from the mixture. It is worth noting that, in this context, a source does not necessarily refer to a physical sound source, but also to any group of acoustic events that should be jointly modeled, such as background noise.

Source separation with NMF can be explained mathematically as follows. Suppose that $\mathbf{W}^s$ represent the set of bases of the speech source, and $\mathbf{W}^b$ is the set of bases of the background noise. Then, any spectrogram of speech $\mathbf{S}$ can be modeled as $\mathbf{S} = \mathbf{W}^s\mathbf{H}^s$, and any spectrogram of noise can be modeled as $\mathbf{B} = \mathbf{W}^b\mathbf{H}^b$, where $\mathbf{H}^s$ and $\mathbf{H}^b$ are the activations of speech and noise patterns. An observed mixture $\mathbf{X}$ combining speech and noise can be expressed as:

$$\mathbf{X} = \mathbf{W}^s\mathbf{H}^s + \mathbf{W}^b\mathbf{H}^b. \tag{3.24}$$

This equation can be written more compactly by stacking both dictionaries in a single matrix $\mathbf{W} = [\mathbf{W}^s, \mathbf{W}^b]$, as well as the activations $\mathbf{H} = [\mathbf{H}^s; \mathbf{H}^b]$. The compact form of the model

$$\mathbf{X} = \mathbf{W}\mathbf{H} \tag{3.25}$$

is again the basic NMF model explained in the previous section. In *unsupervised* source separation, the matrices $\mathbf{W}$ and $\mathbf{H}$, which are unknown, are estimated from

the observation $\mathbf{X}$ by resolving the problem in (3.16), followed by a process that identifies the source each basis is predominantly associated with. However, this simple approach has a number of problems. As mentioned earlier, NMF is able to find the patterns that compose a given mixture if the mixture is very simple, and the patterns are repetitive and very different. However, for realistic acoustic scenes, NMF cannot split the given spectrogram into useful patterns, i.e., it is not guaranteed that each estimated pattern contains information from a single source. Consequently, practical audio separation requires to incorporate certain previous knowledge about the sources. There are three main approaches for doing this, which are not mutually exclusive:

- Learning the dictionaries $\mathbf{W}^s$ and $\mathbf{W}^b$ in advance from training material in which speech and noise are isolated (i.e. $\mathbf{W}^s$ is learned from clean speech and $\mathbf{W}^b$ is learned from isolated noise). During separation, the dictionaries are kept fixed and only their activations $\mathbf{H}^s$ and $\mathbf{H}^b$ are estimated. This approach is discussed in the next Section.

- Imposing different mathematical restrictions to speech and noise coefficients. In basic NMF, the only criterion for estimating the parameters is to minimize the reconstruction error, under the condition that the parameters are nonnegative. However, further restrictions can be imposed on the parameters to obtain solutions with certain properties. This approach leads to regularized decompositions in which the objective function is a weighed combination of the reconstruction error and the defined constraints. We will see this approach en Section 3.3.6.

- Defining source-specific generative models, in which the way the basis vectors are combined is more elaborated (and different for each source). The basic NMF model assumes that each source is generated as a simple linear combination of its bases. However, the constructive combination can be more complex, and inspired, for example, in the production principles of the source. An example is the source/filter model explained in Section 3.3.7.

Independently of the chosen approach, once the estimated speech source $\hat{\mathbf{S}}$ and the estimated noise source $\hat{\mathbf{B}}$ are computed, the resulting components must be synthesized in the time domain. In practice, the decomposition will not be exact and we will only achieve approximate decomposition, i.e., $\mathbf{X} \approx \hat{\mathbf{S}} + \hat{\mathbf{B}}$, and as a consequence, $\mathbf{X}$ is not fully explained by the decomposition. To be able to account for all the energy in the input signal, we can use an alternative method to extract the contributions of the individual sources. Although the separated signals do not completely explain the mixed signal, we assume that they characterize

the relative proportions of the individual signals in the mixture. This leads, for example, to the following estimate for the separated speech:

$$\hat{\mathbf{S}} = \mathbf{X} \bullet \frac{\hat{\mathbf{W}}^s \hat{\mathbf{H}}^s}{\hat{\mathbf{W}}^s \hat{\mathbf{H}}^s + \hat{\mathbf{W}}^b \hat{\mathbf{H}}^b}. \tag{3.26}$$

This filter response is used by the well-known Wiener filter, and the reconstruction is often referred to as the Wiener reconstruction. Finally, to convert the separated spectrograms back to the time domain, the phase of the original mixture is used.

3.3.5 Learning Basis Vectors

To perform *supervised* separation, source-specific dictionaries are obtained in a training stage from a source-specific data set, and finally combined to form the whole dictionary of basis vectors. The dictionary is then kept fixed, and only the activations are estimated, as expressed in (3.17).

There are two main approaches for dictionary learning: the first approach, the *decomposition based learning*, attempts to learn dictionary bases by factorizing training data, in which the sources are isolated, whereas the second approach, the *examplar-based approach*, uses samples from the training data itself as its dictionary atoms, without performing any training. Each method have their strengths and weaknesses. For example, dictionaries learned by decomposition generalize better to unseen data, and consequently can be smaller than exemplar dictionaries. Exemplar dictionaries have the advantage that are easy to generate, because they do not require training, and are almost as discriminative as learned dictionaries [44, 113].

So far, we have assumed that each basis represents a single-frame pattern. Since sources often have similar characteristics in the short-term observations (such as unvoiced phonemes and broadband noise, or voiced phonemes and music), it seems beneficial to use information from multiple time frames. In multi-frame approaches, each basis vector models the spectra of a certain number of consecutive frames. In this case, the observed spectrogram is processed following a sliding window approach, where each input vector is formed by stacking a sequence of consecutive frames [44, 50]. This approach is often used in conjunction with exemplar-based learning.

Learning Speech Bases

It is possible to learn speech models by applying unsupervised NMF to a corpus of training speech. In that case, low-rank factorization is used to capture a compressed model of speech spectra. Alternatively, the speech bases can be chosen

without training, as exemplars. However, learning speech bases involves a series of problems. The complex and non-stationary nature of human speech makes the task quite challenging compared to e.g. modeling musical instruments, whose spectro-temporal trajectories are more consistent. The large variation in casual pronunciation makes it unlikely to find a perfect match to observed speech, and for that reason, joint approximation with multiple candidates is well motivated in speech modelling. Another consequence is that a high number of bases is required to provide a good approximation. In compositional modeling of speech is common to use thousands of bases [44, 45].

For these reasons, a higher level of supervision in learning is recommendable for sparse representation or noise robust speech processing to ensure modelling of characteristic large-scale patterns. Already in [117] some supervision was brought into NMF-based algorithms by segmenting the training corpus into individual phonemes and learning a separate basis for each. Phoneme-dependent bases were also used in [105].

3.3.6 Regularized NMF and Constraints

In standard NMF the only constraint is the element-wise non-negativity of all matrices, and the only objective is the minimization of the reconstruction error. For realistic data, as commented before, this decomposition is not sufficient to achieve a parts-based representation. This happens particularly when the number of bases K in $\mathbf{W}$ is greater than the dimension of the vectors, as there are infinitely many different factorizations that can approximate the input matrix. Learning a dictionary of bases $\mathbf{W}$ at training time, as explained previously, is helpful to provide meaningful and interpretable decompositions at test time, as long as both the test and training data have similar properties. However, the problem of how to learn useful bases for each source during training still remains, because there is not guarantee that the resulting bases are representative of the source. Furthermore, even if a good dictionary $\mathbf{W}$ is specified, there may be multiple activation matrices $\mathbf{H}$ that produce a minimum-divergence solution for the same input data $\mathbf{X}$. Particularly, if the application requires a large number of bases, the resulting activations may not chose the most appropriate ones.

To obtain useful solutions, it is customary to define additional constraints to the factorization problem, in order to enhance and control desired properties. The way to introduce constraints to the model is by using a "penalty term approach". That is, rather than minimizing only the reconstruction error (EUC, KL or IS typically), the objective cost function includes one or more terms that quantify the desired properties on the matrices. The constrained NMF problem is then

expressed as

$$\hat{\mathbf{W}}, \hat{\mathbf{H}} = \arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{X}||\mathbf{WH}) + \lambda C_W(\mathbf{W}) + \alpha C_H(\mathbf{H}), \tag{3.27}$$

where the functions $C_W(\cdot)$ and $C_H(\cdot)$ measure desired properties of the matrices, and λ and α are weight parameters that can be adjusted to increase or decrease the influence of the constraints over the NMF minimization procedure. The constraints can also be defined individually for a certain subset of bases or activations. For instance, if the basis matrix contains bases of two sources, in the form $\mathbf{W} = [\mathbf{W}^s, \mathbf{W}^b]$, a different constraint can be applied to each subset. Similarly, a different constraint can be defined for each subset of activations ($\mathbf{H}^s$ and $\mathbf{H}^b$).

To solve regularized NMF problems such as (3.27), many authors propose update rules adapted only to the proposed constrains. A generic solution is proposed by Virtanen in [133], based on a heuristic approach to derive multiplicative update equations similar to those presented by Lee and Seung [82]. This approach relies on computing the gradients of the objective function $C(\cdot)$ (containing one or more constraints) with respect to the parameters, $\nabla_{\mathbf{W}} C(\cdot)$ and $\nabla_{\mathbf{H}} C(\cdot)$, and then splitting them into the difference of two non-negative terms, in the form $\nabla_{\mathbf{W}} C(\cdot) = \nabla^+_{\mathbf{W}} C(\cdot) - \nabla^-_{\mathbf{W}} C(\cdot)$ and $\nabla_{\mathbf{H}} C(\cdot) = \nabla^+_{\mathbf{H}} C(\cdot) - \nabla^-_{\mathbf{H}} C(\cdot)$. The update rules are finally expressed as

$$\mathbf{H} \leftarrow \mathbf{H} \bullet \frac{\nabla^-_{\mathbf{H}} C(\cdot)}{\nabla^+_{\mathbf{H}} C(\cdot)} \tag{3.28}$$

and

$$\mathbf{W} \leftarrow \mathbf{W} \bullet \frac{\nabla^-_{\mathbf{W}} C(\cdot)}{\nabla^+_{\mathbf{W}} C(\cdot)}. \tag{3.29}$$

Observe that the rules in (3.23) and (3.22) are a particular case of these expressions, in which no constraints are applied. Consequently any regularized NMF problem can be solved by finding the gradients of the objective function with respect to each parameter.

Some of the most important constraints are sparsity or smoothness [133]. More details are given in [P3] and references therein.

3.3.7 Excitation/Filter Model for Speech

In [33], a compositional source/filter model is proposed to represent the signal of interest, and to allow its discrimination from the remaining sources. This representation is specially interesting for vocal sounds, since it approximates their underlaying production characteristics. According to the model, each speech spectral vector $\mathbf{s}_t$ is decomposed into an excitation part $\mathbf{s}_t^{F_0}$ multiplied by a filter part

$\mathbf{s}_t^{\Phi}$, which are respectively composed by a linear combination of P elementary excitation bases $\mathbf{w}_p^{F_0}$ and E elementary filter bases $\mathbf{w}_e^{\Phi}$, as follows:

$$\mathbf{s}_t = \mathbf{s}_t^{\Phi} \bullet \mathbf{s}_t^{F_0} = \left(\sum_{e=1}^{E} h_{et}^{\Phi} \mathbf{w}_e^{\Phi} \right) \bullet \left(\sum_{p=1}^{P} h_{pt}^{F_0} \mathbf{w}_p^{F_0} \right), \tag{3.30}$$

where h_{et}^{Φ} and $h_{pt}^{F_0}$ are non-negative gains, and $\bullet$ denotes the Hadamard product. The excitation bases $\mathbf{w}_p^{F_0}$ represent the discrete collection of sounds from which the signal can be constructed, and which are further modulated by a combination of bases $\mathbf{w}_e^{\Phi}$. Since the model is designed to represent vocals, it is convenient that each vector $\mathbf{w}_p^{F_0}$ represents the glottal signal corresponding to an individual fundamental frequency or pitch. In [33], the bases $\mathbf{w}_p^{F_0}$ are generated using the glottal source model KLGLOTT88 [80], resulting in a fixed dictionary of pitch-related excitations. If a sufficient number of excitations P is used, it is possible to have a fine grid of pitch values, thus covering the whole pitch range of the speaker with enough resolution. On the other hand, the filter bases $\mathbf{w}_e^{\Phi}$ must be able to represent the smooth envelop of the signal. In [33], these bases are generated from a family of smooth functions, resulting in a fixed dictionary of smooth components that can combine to represent any arbitrary smooth envelope.

The P excitation bases $\mathbf{w}_p^{F_0}$ can be grouped into a matrix $\mathbf{W}^{F_0} = [\mathbf{w}_1^{F_0}, \ldots, \mathbf{w}_P^{F_0}]$, and the E filter bases $\mathbf{w}_e^{\Phi}$ into a matrix $\mathbf{W}^{\Phi} = [\mathbf{w}_1^{\Phi}, \ldots, \mathbf{w}_E^{\Phi}]$. Following this notation, the model in (3.30) can then be written in matrix form as

$$\mathbf{S} = (\mathbf{W}^{\Phi}\mathbf{H}^{\Phi}) \bullet (\mathbf{W}^{F_0}\mathbf{H}^{F_0}). \tag{3.31}$$

Here, the gain matrices $[\mathbf{H}^{\Phi}]_{et} = h_{et}^{\Phi}$ and $[\mathbf{H}^{F_0}]_{pt} = h_{pt}^{F_0}$ give the decomposition of the speech source into the dictionaries $\mathbf{W}^{\Phi}$ and $\mathbf{W}^{F_0}$. These amplitudes are estimated from the input signal, while the dictionaries are kept fixed.

This source/filter model has two interesting advantages. First, it describes a generative model that is characteristic of speech, and significantly discriminative from typical noise sounds, which usually do not fit this structure. And second, since the pitch and timbre information is individually represented, it provides a more structured description of speech, allowing the use of a reasonable number of bases. Although the model can be extended to deal also with unvoiced speech, this extension makes the model more sensitive to capture interferences. In this thesis, we focus on separating voiced speech, and use this model as a base of our method [P3].

Chapter 4

Results and Conclusions

In this thesis, we have proposed classification and separation algorithms for applications dealing with speech signals degraded by noise. Particularly, we have developed techniques that are useful for two signal processing fields: hearing aids and speech enhancement. In both cases, the ultimate goal is to deliver an improved speech signal to a human listener. In a hearing aid, the device applies an amplification to the input speech for compensating the patient's hearing losses. In a generic speech enhancer, the objective is to remove the background noise to produce a signal with better objective quality or intelligibility.

The basic concept in the presented work is that fundamental frequency of speech can serve as a robust cue for distinguishing and separating speech from other sounds. In the case of classification, one can define one or more features derived from fundamental frequency (or periodicity) to characterize speech. Fortunately, there are many pitch estimators that are able to work under realistic noise conditions. Specifically, estimators based on autocorrelation principles have been proven to perform relatively well in noise [121, 27]. However, when designing such classifiers, it is necessary to take into account the following problems:

- If the features for classification are sensitive to noise, it is necessary to make a robust estimation of these features. This involves to study the influence of the noise on the chosen features, and its robust computation based on the noise level, often estimated by using noise power estimation techniques
- If the algorithm is implemented on a device with very low computational capacity, such as a hearing aid, the system must estimate the fundamental frequency with the minimum number of operations. This implies to adapt conventional pitch estimators to enable its implementation in hearing aids.

In the case of separation from background noise, it is possible to define a parametric speech model based on the harmonic structure of speech which takes

implicitly into account the F0 of the signal. The model of the background signal can be restricted to be non-harmonic or to have a pitch sequence that is not characteristic of speech. Separation techniques based on compositional models and matrix factorization can be employed to define such models. In that case, the following problems must be solved:

- Formulate an appropriate composite model for speech. This model can be composed of harmonic excitation patterns, where each pattern corresponds to a different possible F0.
- Formulate appropriate mathematical constraints in the decomposition algorithm to define the properties of the noise. These constraints should make the noise to adopt a non-harmonic shape or to have a pitch contours different from typical speech.

In the follow, we summarize the results and conclusions of the works presented in this thesis.

4.1 Speech/Nonspeech Classification for Hearing Aids

In [P1] a low-complexity speech/nonspeech discrimination approach for digital hearing aids is proposed. The proposed approach mainly relies on a low-complexity method for F0 estimation, which consists on computing a decimated version of the cumulative mean difference function. This function is parametrized with a parameter S, which determines the computational complexity of the estimator. The proposed speech/nonspeech discrimination scheme is completed with a feature extraction stage (using F0-based features), a low-complexity classifier and a HMM postprocessing stage. The complexity of the system is mainly due to the F0 estimation stage. The remaining stages do not almost increase system complexity.

Classification accuracy rates are analyzed together with the complexity requirements in order to select the more appropriate classifier and an optimum value of parameter S. In such sense, a Multi-layer Perceptron classifier is selected, and $S = 30$ is a good trade-off value between accuracy and complexity. The proposed speech/nonspeech discrimination scheme is feasible to be implemented in ultra low-power DSP-based digital hearing aids by choosing the suitable configuration setup. Parameter S must be below 20 when the system is intended to operate at 1.28 MIPS, in order to extend the battery operation time (ultra low-power consumption).

For the chosen configuration (MLP and $S = 30$), the system achieves a global accuracy rate equal to 88.76%, evaluated over a database containing clean speech,

noisy speech, music and noise. These results are similar to those obtained by recent methods in the literature. The classification accuracy loss between the decimated difference function (proposed F0 estimation method) and the YIN algorithm is reduced to only 1% in our configuration setup (MLP and $S = 30$). This result evidences the good performance of the proposed F0 estimation method when combined with a MLP-based classifier for speech/nonspeech discrimination in digital hearing aids. The global accuracy rate is increased about 1% when HMM postprocessing is incorporated into the speech/nonspeech discrimination scheme. Higher accuracy rates can be achieved if a certain decision delay is allowed. From experimental results, a 1-s delay is chosen as an optimum value, the classification accuracy rate being about 95%.

Fundamental frequency estimation has a wide range of potential applications in digital hearing aids. Speech intelligibility improvement in digital hearing aids from F0 estimation will be explored in the next future.

4.2 Voicing Detection in Non-stationary Noise

In [P2], we have presented an algorithm for voicing detection intended to work in acoustic environments where the noise is non-stationary. The algorithm computes a signal-adaptive threshold that is compared to the aperiodicity value provided by the difference function, which is a well-known time domain measure of voicing. In clean speech, a fixed threshold is enough to achieve an accurate voicing detection. However, under non-stationary noise, this threshold must be made adaptive and dependent on the current SNR. We have derived an equation to compute this signal-adaptive threshold by assuming that the interfering noise is additive and uncorrelated, and proposed a simple algorithm to estimate the background noise power by assuming local stationarity. Provided an efficient approximation of the difference function, the method is also good enough to be implemented in hearing aids, introducing only a moderate degradation in the system performance.

Experimental results over the NOIZEUS database revealed that the proposed voicing detector is robust enough whenever the assumptions made for the noise hold. The fixed YIN threshold was outperformed for all types of noise, and the method obtained better voicing detection results than the state-of-the-art ETSI ES 202 211 classifier under white-like background noises (such as car, street or train). A simple F_0-based speech enhancement scheme integrating the proposed classifier was implemented to demonstrate the applicability of the method for denoising. The implemented speech enhancement scheme obtained similar quality results, in terms of objective measures (PESQ and LLR), when compared with several well-

known approaches for speech enhancement, such as spectral subtraction, MMSE or the Wiener filter in the ETSI ES 202 050 standard.

Currently, we are working on extending the method to perform speech enhancement on hearing aids. Although the voicing detector is found to produce good classification results when implemented with the decimated difference function proposed in [P1] (and hence it is possible to implement the voicing detector in a hearing aid), the enhancement stage does not make use of the signal representation used in hearing aids. Typically, hearing aids analyze the input signal using a bank of band-pass filters, such that the signal at each output is weighted according to the patient's hearing prescription. It is possible to improve the perceived signal if the system takes into account the F0 and the provided voicing decision.

4.3 Compositional Model for Speech/Noise Separation

In [P3], we proposed a NMF-based algorithm for voiced speech and noise separation, in which the noise components are constrained to obey certain mathematical properties that are characteristic of many background noises. These properties are not defined for specific noises, but in a generic way, enabling to apply the algorithm to a wide range of environments without requiring prior training or a large number of bases. The speech source is represented through an excitation/filter model previously proposed in the literature, with the incorporation of a dictionary of filter bases learned in a training stage from a database of isolated phonemes. The excitation patterns comprise the set of F0s that can be used to approximate the observed speech. This speech model allows to represent the speech signal with a reasonable number of bases, and consequently is computationally more efficient than other speech representations based on compositional models, such as those based on sparse coding.

The method was evaluated on the simulated mixtures of the 3rd CHiME development set. The experiments demonstrate that, in general, the proposed restrictions are adequate for real-word background environments, and improve significantly the results obtained by the model without restrictions. In comparison with other constrained semi-supervised decompositions, such as sparse NMF with speech bases, our method obtains better separation results in terms of SDR and SIR. Specifically, we obtained an average SDR gain of 4.56 dB, which supposed an improvement of 1.65 dB over sparse NMF, and 0.67 over the OM-LSA algorithm. The method also obtained promising results at the SiSEC 2013 international campaign, although the proposed restrictions were not able to characterize appropriately one of the tested environments. The current results of the algorithm,

although promising, may however not be usable for most practical applications, due to its limitation to voiced speech.

Future improvements of the algorithm will be focused on characterizing certain properties of the noise more accurately. It was observed that, although the smooth noise type is able to approximate many instances of noise, it often imposes a strict representation in a sense that the bases and amplitudes must be smooth. Better results could be obtained if the bases and gains are not restricted to be smooth, but their combination is (at least, in comparison with speech). In addition, we intend to explore the incorporation of unvoiced parts to the speech model. Restricting the activations of unvoiced phonemes may help in avoiding the problem of capturing noise components with them. It is also interesting to explore a real-time implementation of the system, or its application in conjunction with spatial cues for multichannel signals.

Bibliography

[1] S. Ahmadi and A. Spanias, "Cepstrum-based pitch detection using a new statistical V/UV classification algorithm", *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 333–338, 1999.

[2] E. Alexandre, L. Cuadra, M. Rosa, and F. López-Ferreras, "Feature Selection for Sound Classification in Hearing Aids Through Restricted Search Driven by Genetic Algorithms", *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2249–2256, Nov. 2007.

[3] ANSI, *ANSI S1.1-1994*, American National Standard Acoustical Terminology, Acoustical Society of America, 1994.

[4] D. Bansal, B. Raj, and P. Smaragdis, "Bandwidth expansion of narrowband speech using non-negative matrix factorization", in *9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, 2003.

[5] A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, "ITU-T recommendation G. 729 Annex B: A silence compression scheme for use with G. 729 optimized for V. 70 digital simultaneous voice and data applications", *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 64–73, Sep. 1997.

[6] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise", in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 208–211, 1979.

[7] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 538–549, 2010

[8] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[9] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol.

27, no. 2, pp. 113–120, 1979.

[10] J. C. Brown, “Musical fundamental frequency tracking using a pattern recognition method”, *J. Acoust. Soc. Am.*, vol. 92, no. 3, pp. 1394–1402, Sep. 1992.

[11] G. J. Brown and M. Cooke, “Computational auditory scene analysis”, *Computer speech and language*, vol. 8, no. 4, pp. 297–336, 1994.

[12] M. Büchler, “Algorithms for sound classification in hearing instruments”, PhD Thesis, Swiss Federal Institute of Technology, Zurich, 2002.

[13] M. Büchler, S. Allegro, S. Launer, and N. Dillier, “Sound classification in hearing aids inspired by auditory scene analysis”, *EURASIP Journal on Applied Signal Processing*, vol. 18, pp. 2991–3002, 2005.

[14] J. J. Burred and A. Lerch, “A Hierarchical Approach to Automatic Musical Genre Classification”, in *Proc. of the 6th International Conference on Digital Audio Effects (DAFX)*, London, UK, September 2003.

[15] J. Carabias-Orti, F. Rodriguez-Serrano, P. Vera-Candeas, F. Canadas-Quesada, and N. Ruiz-Reyes, “Constrained non-negative sparse coding using learnt instrument templates for realtime music transcription”, *Engineering Applications of Artificial Intelligence*, pp. 1671–1680, 2013.

[16] O. Cappe, “Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor”, *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 345–349, 1994.

[17] J. Charpentier, “Pitch detection using the short-term phase spectrum”, in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP’86)*, Tokyo, Japan, 1986, pp. 113–116.

[18] J. Chen, J. Benesty, Y. Huang, and S. Doclo, “New insights into the noise reduction Wiener filter”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1218–1234, 2006.

[19] Y.-C. Cho and S. Choi, “Nonnegative features of spectro-temporal sounds for classification”, *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1327–1336, 2005.

[20] E. Cho, J. O. Smith, and B. Widrow, “Exploiting the harmonic structure for speech enhancement”, in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, March 2012, pp 4569–4572.

[21] M. G. Christensen, A. Jakobsson, and S. H. Jensen, “Joint high-resolution fundamental frequency and order estimation”, *IEEE Trans. Audio, Speech,*

Lang. Process., vol. 15, no. 5, pp. 1635–1644, 2007.

[22] M. G. Christensen, P. Stoica, A. Jakobsson, and S. H. Jensen, "Multi-pitch estimation", *Signal Processing*, vol. 88, no. 4, pp. 972–983, 2008.

[23] M. G. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, Synthesis Lectures on Speech and Audio Processing, Morgan & Claypool, 2009.

[24] R. Crochiere, "A weighted overlap-add method of short-time Fourier analysis/synthesis", *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 28, pp. 99–102, Feb. 1980.

[25] C. J. Darwin, "Perceptual grouping of speech components differing in fundamental frequency and onset-time", *The Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology*, vol. 33, no. 2, pp. 185–207, 1981.

[26] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 357–366, 1980.

[27] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music", *Journal of the Acoustic Society of America (JASA)*, vol. 111, no. 4, pp. 1917–1930, April 2002.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal Royal Statistical Society*, Series B, vol. 39, no. 1, pp. 1–38, 1977.

[29] C. Ding, T. Li, and W. Ping, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing", *Computational Statistics & Data Analysis*, vol. 52, no. 8, pp. 3913–3927, 2008.

[30] K. Dressler, "Pitch estimation by the pair-wise evaluation of spectral peaks", in *AES 42nd International Conference*, Ilmenau, Germany, 2011.

[31] J. Droppo and A. Acero, "Maximum a posteriori pitch tracking", in *Proc. Intl. Conf. on Spoken Lang. Processing (ICSLP)*, 1998.

[32] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, New York Wiley, 1973.

[33] J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation", *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1180–1191, 2011.

[34] D. Ealey, H. Kelleher, and D. Pearce, “Harmonic tunneling: tracking non-stationary noises during speech”, in *Proc. of the 7th European Conference on Speech Communication and Technology*, Aalborg, Denmark, Sep 2001, pp. 437–440.

[35] Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[36] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator”, *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 443–445, 1985.

[37] ETSI ES 202 211 V1.1.1, “Speech processing, Transmission and quality aspects (STQ); Distributed speech recognition; Extended front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm”, 2003.

[38] ETSI ES 202 050 V1.1.5, “Speech processing, Transmission and quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms”, 2007.

[39] M. R. Every, “Separation of musical sources and structure from single-channel polyphonic recordings”, PhD dissertation, Dept. Electron., Univ. York, York, U.K., 2006.

[40] C. Fevotte, N. Bertin, and J.-L. Durrieu, “Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis”, *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[41] C. Fevotte and J. Idier, “Algorithms for nonnegative matrix factorization with the beta-divergence”, *Neural Computation*, vol. 23, pp. 2421–2456, 2011.

[42] D. FitzGerald, M. Cranitch and E. Coyle, “Extended nonnegative tensor factorisation models for musical source separation”, *Computat. Intell. Neurosci.*, vol. 2008, 2008.

[43] J. T. Geiger, F. Weninger, A. Hurmalainen, J. F. Gemmeke, M. Wollmer, B. Schuller, G. Rigoll, and T. Virtanen, “The TUM+TUT+KUL approach to the CHiME Challenge 2013: Multi-stream ASR exploiting BLSTM networks and sparse NMF”, in *Proc. of 2nd CHiME Workshop held in conjunction with ICASSP’13*, 2013, pp. 25–30.

[44] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, “Exemplar-based sparse representations for noise robust automatic speech recognition”, *IEEE*

Transactions on Audio, Speech, and Language Processing, vol. 19, no. 7, pp. 2067–2080, 2011.

[45] J. F. Gemmeke and H. Van hamme, "Advances in Noise Robust Digit Recognition using Hybrid Exemplar-Based Techniques", in *Proc. of the 13th Annual Conference of International Speech Communication Association (INTERSPEECH)*, Portland, OR, USA, 2012, pp. 2134–2137.

[46] T. Gerkmann and R. C. Hendriks, "Unbiased MMSE-based noise power estimation with low complexity and low tracking delay", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 4, pp. 1383 –1393, May 2012.

[47] J. D. Gibson, B. Koo, and S. D. Gray, "Filtering of colored noise for speech enhancement and coding", *IEEE Trans. Signal Process.*, vol. 39, no. 8, pp. 1732–1742, 1991.

[48] J. S. Gill, "Automatic extraction of the excitation function of speech with particular reference to the use of correlation methods", in *3rd I.C.A.*, Stuttgart, Germany, 1959.

[49] M. Goto, "A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals", *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[50] E. M. Grais and H. Erdogan, "Single channel speech music separation using nonnegative matrix factorization with sliding windows and spectral masks", in *Proc. of the 12th Annual Conference of International Speech Communication Association (INTERSPEECH)*, Florence, Italy, 2011, pp. 1773–1776.

[51] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit", *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 101–111, 2003.

[52] S. Gonzalez and M. Brookes, "PEFAC - A pitch estimation algorithm robust to high levels of noise", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 518–530, 2014.

[53] D. Griffin and J. Lim, "Multiband Excitation Vocoder", *IEEE Trans. on Acoust., Speech and Sig. Process.*, vol. 36, no. 8, pp. 1223–1235, 1988.

[54] E. Guaus and E. Batlle, "A non-linear rhythm-based style classification for broadcast speech-music discrimination", in *AES 116th Convention*, 2004.

[55] M. Helén and T. Virtanen, "Separation of Drums from Polyphonic Music Using Non-negative Matrix Factorization and Support Vector Machine", in *Proc. of the 13th European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, 2005.

[56] K. Han and D. L. Wang, "An SVM based classification approach to speech separation", in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4632–4635.

[57] T. Heittola, A. Klapuri, and T. Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation", in *Proc. of International Conference on Music Information Retrieval*, Kobe, Japan, 2009.

[58] R. Hendriks, R. Heusdens, and J. Jensen, "MMSE based noise PSD tracking with low complexity", in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2010, pp. 4266–4269.

[59] H. Hermansky, "Perceptual linear predictive (PLP) anlysis of speech", *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, 1990.

[60] H. Hermansky and N. Morgan, "RASTA processing of speech", *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 578–589, 1994.

[61] D. J. Hermes, "Measurement of pitch by subharmonic summation", *J. Acoust. Soc. Am.*, vol. 83, no. 1, pp. 257–264, Jan. 1988.

[62] R. Hetch-Nielsen, *Neurocomputing*, Addison-Wesley, New York, 1990.

[63] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets", *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[64] D. Hosseinzadeh and S. Krishnan, "On the use of complementary spectral features for speaker recognition", *EURASIP Journal on Advances in Signal Processing*, 2008.

[65] G. Hu and D. L. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation", *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1135–1150, 2004.

[66] O. Izmirli, "Using spectral flatness based feature for audio segmentation and retrieval", Technical report, Center for Arts and Technology, Department of Mathematics and Computer Science, Connecticut College, 1999.

[67] L. B. Jackson, *Digital Filters and Signal Processing*, Kluwer Academic Publishers, 1989.

[68] J. Jensen and J. H. L. Hansen, "Speech enhancement using a constrained iterative sinusoidal model", *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 7, pp. 731–740, Oct. 2001.

[69] Z. Jin and D. L. Wang, "HMM-based multipitch tracking for noisy and reverberant speech", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 5, pp. 1091–1102, Jul. 2011.

[70] C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, "Real-time speech separation by semi-supervised nonnegative matrix factorization," in *Proc. of LVA/ICA*, Tel Aviv, Israel, March 2012, pp. 322–329.

[71] B. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate observations of Markov chains", *IEEE Transactions on Information Theory*, vol. 32, no. 2, 1986.

[72] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise", in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, FL, USA, May 2002, pp. 4160–4164.

[73] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners", *J. Acoust. Soc. Am.*, vol. 126, no. 3, pp. 1486–1494, Sep. 2009.

[74] G. Kim and P. Loizou, "Improving speech intelligibility in noise using environment-optimized algorithms", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2080–2090, Nov. 2010.

[75] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons", *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[76] M. Kim, J. Yoo, K. Kang, and S. Choi. "Nonnegative matrix partial cofactorization for spectral and temporal drum source separation", *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1192–1204, 2011.

[77] U. Kjems, M. S. Pedersen, J. B. Boldt, T. Lunner, and D. L. Wang, "Speech intelligibility of ideal binary masked mixtures", in *Proc. European Signal Processing Conf. (EUSIPCO)*, Aalborg, Denmark, Aug. 2010, pp. 1909–1913.

[78] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes", in *Proc Intl Conf Music Inf. Retrieval*, vol. 6, 2006, pp. 216–221.

[79] A. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 255–266, Feb. 2008.

[80] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers", *The Journal of the Acoustical Society of America*, vol. 87, no. 2, pp. 820–857, 1990.

[81] A. A. Kressner, D. V. Anderson, and C. J. Rozell, "A novel binary mask estimator based on sparse approximation", in *Proc. IEEE Intl. Conf. on*

Acoustics, Speech and Signal Processing (ICASSP), 2013.

[82] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization", in *Proc. of Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[83] C.-T. Lee, Y.-H. Yang, and H.-H. Chen, "Multipitch Estimation of Piano Music by Exemplar-Based Sparse Representation", *IEEE Transcations on Multimedia*, vol. 14, no. 3, pp. 608–618, 2012.

[84] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval", *Pattern recognition letters*, pp. 533–544, 2001.

[85] J. S. Lim, A. V. Oppenheim, and L. Braida, "Evaluation of an adaptive comb filtering method for enhancing speech degraded by white noise addition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 4, pp. 354–358, Aug. 1978.

[86] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech", in *Proceedings of the IEEE*, vol. 67, no. 12, 1979, pp. 1586–1604.

[87] L. Lu, H. J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation", *IEEE Transactions on speech and audio processing*, vol. 10, no. 7, pp. 504–516, October 2002.

[88] R. C. Maher and J. W. Beauchamp, "Fundamental Frequency Estimation of Musical Signals Using a Two-Way Mismatch Procedure", *Journal of the Acoustical Society of America*, vol. 95, no. 4, pp. 2254–2263, 1993.

[89] R. Martin, "Spectral subtraction based on minimum statistics", in *Proc. European Signal Processing Conf.*, 1994, pp. 1182–1185.

[90] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics", *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[91] R. Martin, "Speech enhancement based on minimum mean-square error estimation and supergaussian priors", *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 845–856, 2005.

[92] J. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Trans. Acoust., Speech, Sig. Process.*, vol. 34, no. 4, pp. 744–754, 1986.

[93] Y. Medan, E. Yair, and D. Chazan, "Super resolution pitch determination of speech signals", *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 40–48,

1991.

[94] R. Meddis, “Simulation of mechanical to neural transduction in the auditory receptor”, *J Acoust Soc Am.*, vol. 79, no. 3, pp. 702–11, 1986.

[95] R. L. Miller and E. S. Weibel, “Measurement of the fundamental period of speech using a delay line”, in *51st Meeting of the Acoustical Society of America*, 1956.

[96] J. Moorer, “The optimum comb method of pitch period analysis of continuous digitized speech”, *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, no. 5, pp. 330–338, Oct. 1974.

[97] J. A. Morales-Cordovilla, P. Cabañas-Molero, A. M. Peinado and V. Sánchez, “A robust pitch extractor based on DTW lines and CASA with application in noisy speech recognition”, in *Iberspeech*, Communications in Computer and Information Science (Springer), Madrid, Spain, November 2012, pp. 197–206.

[98] M. Noll, “Pitch determination of human speech by harmonic product spectrum, the harmonic sum, and a maximum likelihood estimate”, in *Proc. Symposium on Computer Processing Communications*, 1969, pp. 779-797.

[99] P. Nordqvist and A. Leijon, “An efficient robust sound classification algorithm for hearing aids”, *J. Acoust. Soc. Amer.*, vol. 115, no. 6, pp. 3033–3041, 2004.

[100] J. Nikunen and T. Virtanen, “Object-based audio coding using non-negative matrix factorization for the spectrogram representation”, in *Proc. of the 128th Audio Engineering Society Convention*, London, UK, 2010.

[101] Y. Panagakis, C. Kotropoulos, and G. R. Arce, “Music Genre Classification via Sparse Representations of Auditory Temporal Modulations”, in *Proc. of the 17th European Signal Processing Conference (EUSIPCO)*, Glasgow, Scotland, UK, 2009, pp. 1–5.

[102] T. W. Parsons, “Separation of speech from interfering speech by means of harmonic selection”, *J. Acoust. Soc. Am.*, vol. 60, no. 4, pp. 911–918, Oct. 1976.

[103] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, “Sparse representations in audio & music: from coding to source separation”, in *Proc. of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2009.

[104] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, in *Proc. of the IEEE*, February 1989, vol. 77, no. 2, pp. 257–286.

[105] B. Raj, R. Singh, and T. Virtanen, "Phoneme-dependent NMF for speech enhancement in monaural mixtures", in *Proc. of International Conference on Spoken Language Processing (INTERSPEECH)*, 2011, pp. 1217–1220.

[106] A. Ramalingam and S. Krishnan, "Gaussian mixture modeling of short-time fourier transform features for audio fingerprinting", *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, pp. 457–463, Dec. 2006.

[107] N. Roman, D. Wang, and G. J. Brown, "Speech segregation based on sound localization", *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 2236–2252, Oct. 2003.

[108] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. J. Manley, "Average magnitude difference function pitch extractor", *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, pp. 353–362, 1974.

[109] J. Rouat, Y. C. Liu, and D. Morissette, "A pitch determination and voiced/unvoiced decision algorithm for noisy speech", *Speech Communication*, vol. 21, no. 3, pp. 191–207, 1997.

[110] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama, "Single and multiple F0 contour estimation through parametric spectrogram modeling of speech in noisy environments", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1135–1145, 2007.

[111] R. Rui and C.-C. Bao, "Projective Non-negative Matrix Factorization with Bregman Divergence for Musical Instrument Classification", in *Proc. of IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*, Hong Kong, 2012, pp. 415–418.

[112] N. Ruiz-Reyes, P. Vera-Candeas, J. E. Muñoz, S. Garcia-Galán, and F. J. Cañadas, "New speech/music discrimination approach based on fundamental frequency estimation", *Multimedia Tools and Applications*, vol. 41, pp. 253–286, January 2009.

[113] T. N. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, D. Van Compernolle, K. Demuynck, J. F. Gemmeke, J. R. Bellegarda, and S. Sundaram, "Exemplar-Based Processing for Speech Recognition: An Overview", *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 98–113, 2012.

[114] M. Sambur, "Adaptive noise canceling for speech signals", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 5, pp. 419–423, 1978.

[115] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator", in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1997.

[116] M. N. Schmidt and M. Morup, "Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation", in *Proc. of the 6th International Conference on Independent Component Analysis and Blind Source Separation (ICA)*, Charleston, SC, USA, 2006, pp. 700–707.

[117] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization", in *Proc. of International Conference on Spoken Language Processing (INTERSPEECH)*, 2006.

[118] M. R. Schroeder, "Period histogram and product spectrum: New methods for fundamental-frequency measurement", *J. Acoust. Soc. Am.*, vol. 43, no. 4, pp. 829–834, Apr. 1968.

[119] M. R. Schroeder, "Recognition of complex acoustic signals", Life Sciences Research Report 5, T.H. Bullock, Ed. Berlin: Abakon Verlag, 1977.

[120] M. Shashanka, B. Raj, and P. Smaragdis, "Sparse overcomplete latent variable decomposition of counts data", in *Proc. of Neural Information Processing Systems*, Vancouver, Canada, 2007.

[121] T. Shimamura and H. Kobayashi, "Weighted autocorrelation for pitch extraction of noisy speech", *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 7, pp. 727–730, Oct. 2001.

[122] P. Smaragdis and J. C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription", in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2003, pp. 177–180.

[123] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–14, 2007.

[124] P. Sprechmann, A. M. Bronstein, M. M. Bronstein, and G. Sapiro, "Learnable Low Rank Sparse Models for Speech Denoising", in *Proc. of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 136–140.

[125] J. Taghia, N. Mohammadiha, J. Sang, V. Bouse, and R. Martin, "An evaluation of noise power spectral density estimation algorithms in adverse acoustic environments", in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4640 –4643.

[126] D. Talkin, “A robust algorithm for pitch tracking (RAPT)”, *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds. Amsterdam: Elsevier, pp. 495–518, 1995.

[127] J. Tchorz and B. Kollmeier, “SNR estimation based on amplitude modulation analysis with applications to noise suppression”, *IEEE Trans. Speech, Audio Process.*, vol. 11, no. 3, pp. 184–192, May 2003.

[128] C. Tzagkarakis and A. Mouchtaris, “Sparsity Based Noise Robust Speaker Identification Using a Discriminative Dictionary Learning Approach”, in *Proc. of the 21st European Signal Processing Conference (EUSIPCO)*, Marrakech, Morocco, 2013.

[129] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals”, *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, July 2002.

[130] S. van de Par, A. Kohlrausch, R. Heusdens, J. Jensen and S. H. Jensen, “A Perceptual Model for Sinusoidal Audio Coding Based on Spectral Integration”, *EURASIP Journal on Applied Signal Processing*, vol. 9, pp. 1292–1304, 2005.

[131] H. L. Van Trees, *Detection, Estimation and Modulation Theory, PartI*, pp. 198–206, Wiley, 1968.

[132] N. Virag, “Single channel speech enhancement based on masking properties of the human auditory system”, *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 126–137, Mar. 1999.

[133] T. Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[134] T. Virtanen, J. Gemmeke, and B. Raj, “Active-set Newton algorithm for overcomplete non-negative representations of audio”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no.11, 2013.

[135] Y. Wang and D. L.Wang, “Boosting classification based speech separation using temporal dynamics”, in *Proc. Interspeech Conf.*, 2012.

[136] F. Weninger and B. Schuller, “Optimization and parallelization of monaural source separation algorithms in the openBliSSART toolkit”, *Journal of Signal Processing Systems*, vol. 69, no. 3, pp. 267 - 277, 2012.

[137] N. Wiener, *The interpolation, extrapolation and smoothing of stationary time series*, Wiley, 1949.

[138] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors", in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 4029–4032.

[139] M. Wohlmayr, M. Stark, and F. Pernkopf, "A probabilistic interaction model for multipitch tracking with factorial hidden Markov models", *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 799–810, 2011.

[140] M. Wu, D. L. Wang, and G. J. Brown, "A multipitch tracking algorithm for noisy speech", *IEEE Trans. Speech Audio Process.*, vol. 11, no. 3, pp. 229–241, May 2003.

[141] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with constrained second-order optimization", *Signal Processing*, vol. 87, no. 8, pp. 1904–1916, 2007.

[142] X.-L. Zhang and J. Wu, "Deep Belief Networks Based Voice Activity Detection", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, April 2013.

[143] M. Zivanovic, A. Röbel, and X. Rodet, "Adaptive Threshold Determination for Spectral Peak Classification", in *Proc. of the 10th Int. Conf. on Digital Audio Effects (DAFx-07)*, Bordeaux, France, September, 2007, pp. 47–54.

Paper A

Low-complexity F0-based Speech/Nonspeech Discrimination Approach for Digital Hearing Aids

P. Cabañas-Molero, N. Ruiz-Reyes, P. Vera-Candeas, and S. Maldonado-Bascon, "Low-complexity F0-based speech/nonspeech discrimination approach for digital hearing aids", in *Multimedia Tools and Applications*, Volume 54, Issue 2, August 2011, pp. 291–319.

Abstract

Digital hearing aids impose strong complexity and memory constraints on digital signal processing algorithms that implement different applications. This paper proposes a low complexity approach for automatic sound classification in digital hearing aids. The proposed scheme, which operates on a frame-by-frame basis, consists of two stages: analysis stage and classification stage. The analysis stage provides a set of low-complexity signal features derived from fundamental frequency (F0) estimation. Here, F0 estimation is performed by a decimated difference function, which results in a reduced-complexity analysis stage. The classification stage has been designed with the aim of reducing the complexity while maintaining high accuracy rates. Two low-complexity classifiers have been evaluated, a tree-based C4.5 and a Multi-layer Perceptron (MLP), the MLP being chosen because it provides the best accuracy rates and fits to the computational and memory constraints of ultra low-power DSP-based hearing aids. The classification stage is composed of a MLP classifier followed by a Hidden Markov Model (HMM), providing a good trade-off solution between complexity and classification accuracy rate. The goal of the proposed approach is to perform a robust discrimination among speech/nonspeech parts of audio signals in commercial digital hearing aids, the computational cost being a critical issue. For the experiments, an audio database including speech, music and noise signals has been used.

1 Introduction

Hearing losses affect about 13% of the population in most developed countries. Approximately 90% of those with hearing impairments could benefit from modern hearing aids [1]. However, about 25% of those who own hearing aids do not wear them because of irritating and unpleasant whistles and/or other amplified noises caused by the surrounding background noise they encounter in their everyday life. This astonishing irregular use of hearing aids arises from a variety of reasons [2]. The problem becomes more accentuated because understanding speech with background noise is much more difficult for hearing-impaired people than for healthy listeners [3].

Research in digital hearing aids can improve the quality of life of many people. Approaches in signal processing research on digital hearing aids fall into four areas, which cover signal acquisition, amplification, transmission, measurement, filtering, parameter estimation, separation, detection, enhancement, modeling, and classification. The first area uses advanced signal processing techniques to characterize and compensate for various hearing impairments, such as loudness

and frequency selectivity loss. The second area consists of effective target signal enhancement and noise reduction, which includes adaptive microphone array technologies, spectral subtraction algorithms, blind source separation and sound classification. The third area focuses on the real-world use of hearing aids and addresses issues such as flexibility, convenience, feedback cancellation, and artifact reduction. The fourth area is devoted to expanding hearing aid technology into devices that are also able to perform other functions, such as mobile phones and music players. In this area, issues such as echo cancellation, bone-conductive microphones, and wireless voice link are of interest [4]. The signal processing research in this work falls into the second area.

Hearing aid users can improve the perception of signals at different listening conditions if a variety of amplification schemes are available in digital hearing aids [5, 6]. Some modern digital hearing aids allow the user to manually select among different programs depending on the acoustic environment the user is in. The problem here is that the user has to recognize the acoustic environment and selects the best-suited program using a switch on the hearing device or with some kind of remote control. This approach commonly exceeds the abilities of most hearing aid users (especially the elderly), in particular for the smallest In-The-Canal (ITC) or Completely-In-the-Canal (CIC) hearing aids.

The previous paragraph shows the need for hearing aids that are able to automatically classify the acoustic environment the user is in. Hearing-impaired people are willing to use hearing aids that allow to automatically classify different acoustic environments. However, few hearing aids on the market can perform classification and adaptation tasks. These advanced functionalities, when incorporated to hearing aids, can improve speech intelligibility, which increases the comfort level of hearing-impaired people, allowing them to lead a normal life. Furthermore, recent studies [7] suggest that automatic switching is deemed useful by most of hearing-impaired people, even if its performance is not completely perfect.

Because of the limitations imposed by the hardware requirements (computational speed, memory need and power consumption) and other practical factors, the development and implementation of signal processing techniques for digital hearing aids has been a challenging and active research area over the last decade. In particular, developing an automatic sound classification system for digital hearing aids is a really complex and challenging goal mainly due to the just mentioned limitations. Digital hearing aids must work at very low clock frequencies in order to reduce power consumption and thus increase battery life. From this constraint, an upper bound in the number of operations per second is derived. Therefore, signal processing techniques and algorithms must be tailored for properly classifying audio signals while using the minimum possible number of operations.

In last years, some contributions have been made regarding the problem of sound classification in digital hearing aids. Nordqvist and Leijon propose in [8] a Hidden Markov Model (HMM) based sound classification algorithm for hearing aids. The algorithm only uses modulation characteristics of the signal, being implemented in a digital signal processor (DSP) based hearing device. Three listening environment categories are considered for testing: speech in traffic noise, speech in babble, and clean speech. In [1], a sound classification system for acoustic environment recognition in hearing aids is proposed. The system distinguishes four sound classes (clean speech, speech in noise, noise and music) using a set of features inspired by auditory scene analysis. The work in [9] is centered on exploring proper training algorithms for Multi-layer Perceptrons (MLPs) to be used within digital hearing aids. The training methods explored in [9] are Gradient Descent, Levenberg-Marquardt and Levenberg-Marquardt with Bayesian Regularization. The work in [2] deals with feature selection for improving sound classification in hearing aids. A genetic algorithm with restricted search is evaluated for feature selection, showing promising results. The approaches proposed in [10] and [11] discriminate among speech and nonspeech classes using neural network (NN) classifiers specifically tailored to be implemented in hearing aid devices. In the former approach, the NN is tailored by properly reducing the numbers of neurons without degrading the classification performance. In the latter, the activation function of each neuron is severely simplified, and the effects of the finite-precision of the DSP are taken into account to optimally quantize the parameters of the network.

The short-term goal of this work is the design of an efficient automatic speech-nonspeech discrimination system that can be programmed in a low-power DSP-based hearing aid. Although many other variations, apart from speech and nonspeech, exist in the auditory environment, discrimination between speech and nonspeech is a crucial task in hearing aids. As explained in [10], intelligibility of speech (in presence or not of background noise) and its discrimination from nonspeech sounds (whose amplification is unpleasant and irritating) are the two most important aspects for hearing aid users. An automatic speech/nonspeech discrimination system can clearly assist the hearing device in satisfying both needs, by automatically selecting an amplification program on speech (improving intelligibility) and an attenuating program on nonspeech (improving comfort). It is clear, however, that the identification of more specific acoustic environments, such as "speech in noise" or "music", is an important and desirable feature, since it enables the automatic selection of amplification schemes specifically fitted to those listening conditions. Nevertheless, an efficient speech/nonspeech discrimination approach is also very appreciated by hearing aid patients, specially if it provides a robust performance. For this reason, the design of algorithms for discriminating

between speech and other sounds in hearing aids is still the subject of recent and active research [10, 11, 12].

In response of such need, the goal of the proposed approach is to perform a robust discrimination among speech/nonspeech parts of audio signals in commercial digital hearing aids. The system, that operates on a frame-by-frame basis, is basically composed of a feature extraction stage (analysis stage) and a classification stage.

Signal feature extraction is typically performed by Fourier transform computation of windowed audio frames [13, 14, 15]. Recently, signal features have been extracted from the output of a weighted overlap-add (WOLA) filterbank in DSP-based hearing aids [2, 9]. In this work, signal features are extracted from fundamental frequency (F0) estimates provided by a decimated difference function, which results in a reduced-complexity analysis stage. Note that F0 estimation can also be employed for other concurrent applications in digital hearing aids, such as adaptive filtering, noise canceling, speech enhancement and speech separation [16, 17].

In order to approach the global long-term goal of improving speech intelligibility, it is important to select a suitable classifier. The ideal candidate in hearing aid applications should require as low complexity as possible while maintaining a high enough classification accuracy rate. As shown in Section 4, the classification stage is composed of a MLP classifier followed by a HMM [18, 19]. A feasible alternative to the MLP classifier is the tree-based C4.5 classifier [20, 21], which also fits to the complexity and memory constraints, but with lower accuracy rates. The HMM postprocessing step incorporates memory into the system, avoiding occurrence of isolate errors. Combination of MLP and HMM provides a good trade-off solution between complexity and classification accuracy rate.

The main contribution of this work is the proposed low-complexity and high-accuracy approach for speech/nonspeech discrimination in a DSP-based hearing aid. The main novelties of the paper are: 1) the decimated difference function for F0 estimation, and 2) the two-stage cascaded classification scheme (MLP classifier + HMM) for speech/nonspeech discrimination in a DSP-based hearing aid.

The paper is structured as follows. Section 1 outlines the problem of automatic sound classification in digital hearing aids, briefly describes some recent approaches for the problem and states the main contribution and novelties of the paper. Problem statement, including design constraints, data structure and windowing scheme, is described in Section 2. Section 3 describes in detail the main components of the proposed approach. In Section 4, the experimental setup is explained and different results are shown. Finally, Section 5 is devoted to summarize the main conclusions of the work. Future works are also pointed out.

2 Problem Statement

2.1 Design Constraints

As mentioned, DSP-based hearing aids generally have strong constraints in terms of computational capacity and memory. These constraints mainly arise from the small size of digital hearing aids, specially for the smallest ITC or CIC models. The smallest the hearing aid is, the strongest the constraints are. Note that the DSP in a digital hearing aid usually has to integrate not only the CPU core but also A/D and D/A converters, a filterbank, RAM, ROM and EPROM memories and input/output ports. DSP-based hearing aids contain a small battery for supplying energy to the DSP, which also influences in the aforementioned constraints. The hearing aid has to work at very low clock frequencies in order to reduce power consumption and thus extend battery operating time.

There are on the market hearing aids with less restrictions, such as Behind-The-Ear (BTE) and In-The-Ear (ITE) hearing aids. They allow the implementation of more powerful signal processing algorithms at the expense of a higher size. Anyway, the computational capabilities of ultra low-power DSPs have increased in the last few years, allowing the implementation of new signal processing algorithms in digital hearing aids [9, 22].

In this work, we propose a low-complexity speech/nonspeech discrimination approach specifically tailored to be implemented in a low-power DSP-based hearing aid. In order to explore the feasibility of the proposed approach to be used in a realistic hearing device, a commercial DSP for hearing aids has been considered in our study. Here, Toccata Plus™ flexible DSP system for hearing aids from ON Semiconductor has been chosen as a reference [23, 24]. This platform is employed by several manufacturers as the core part of their hearing aid devices, being considered as representative of state-of-the-art low-power signal processors. Furthermore, as mentioned in [11], most of the hearing aids currently available on the market integrate a processor with similar computational speed (up to 2.56 MIPS). Only the latest hearing instruments are based on more advanced DSP platforms (such as the Orela© 4500 series or the Ezairo™ 5900, both from ON Semiconductor), offering a computational capacity that does not use to exceed 5 MIPS [24]. For this reason, in recent works, the Toccata Plus DSP system (or similar) has been considered as a reference to design sound classification algorithms for digital hearing aids [11, 25, 26, 27].

Toccata Plus block diagram is shown in Figure 1. The processing elements of the entire system are: a) the RCore, a fully programmable DSP core, and b) the WOLA filterbank coprocessor, a dedicated configurable processor that transforms the audio signal to the time-frequency domain. The system also integrates other

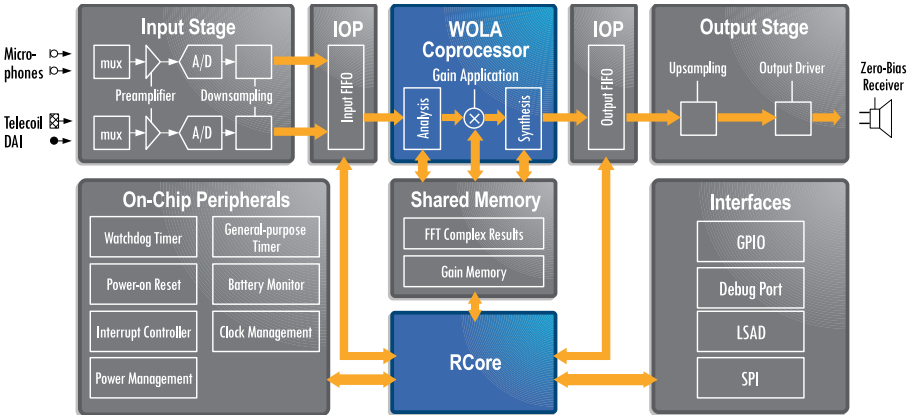

Figure 1: Toccata Plus block diagram.

components, such as A/D and D/A converters, RAM memories (8-Kwords for data and 12-Kwords for program instructions) and several input/output interfaces.

The RCore processor is the main element of the system. This processor executes all algorithms implemented on the device, including the signal processing stages that compensate the hearing losses. The RCore processor can operate at three configurable clock frequencies in the Toccata Plus platform: 1.28 MHz, 1.92 MHz and 2.56 MHz. Since the processor is able to execute one instruction per clock cycle, a configurable computational power of 1.28, 1.92 or 2.56 MIPS is provided by the DSP.

In Section 4, it will be shown how the computational cost of the proposed low-complexity speech/nonspeech discrimination approach matches to the computational constraints of the chosen DSP-based hearing aid.

2.2 Data Structure and Windowing Scheme

In this work, an *analysis window* of 20 ms ($W = 320$ samples at $f_s = 16,000$ Hz sampling rate) is defined. This value will be justified later (see subsection 3.1). A *texture window* of approximately 1 s (50 analysis windows) is also defined. Overlapping with a hop size of 160 samples (half-window overlapping) is performed, which results in 99 short-time frames for the 1 s-length texture window.

Since F0 estimation is performed frame-by-frame using the 20 ms-length analysis window with half-window overlapping, a 99-length low-level feature vector $\mathbf{L}$ is defined for the 1 s-length texture window. From vector $\mathbf{L}$, containing F0 estimates, a high-level feature vector $\mathbf{H}$ is also defined for the 1 s-length texture window. Features in vector $\mathbf{H}$, providing valuable information about the temporal evolution of F0 estimates within the texture window, are applied to the classifica-

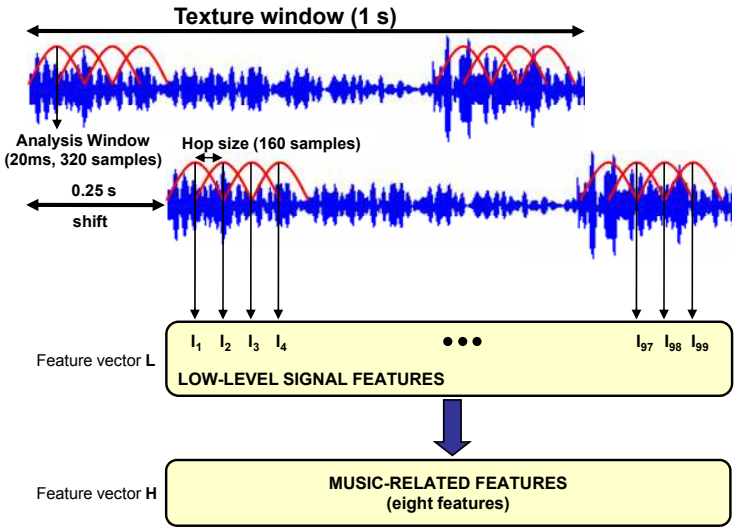

Figure 2: Example illustrating how to compute the input values to the classification stage.

tion stage in order the system to decide whether the analyzed 1 s-length window belongs to the speech class or the nonspeech class.

Instead of using typical statistical features (mean, standard deviation, skewness, etc.), eight features with musical meaning are here considered [28]. They are briefly described in subsection 3.2. Therefore, the 99-length low-level feature vector $\mathbf{L}$, containing F0 estimates, is transformed into a lower dimensional high-level feature vector $\mathbf{H}$, containing eight music-related features to be applied to the classification stage.

The texture window is shifted by 250 ms, which entails updating feature vector $\mathbf{H}$ every 250 ms. Hence, decisions about the class the current segment belongs to are taken every 250 ms. Lower values of the texture window shift allow to reduce the time during which the system stands at a erroneous status at the expense of increasing the computational cost. Figure 2 shows the windowing scheme from which the input values for the classification stage are computed.

To complete this section, it is worth mentioning that the complete implementation of the proposed speech/nonspeech discrimination approach into the hearing aid itself is out of the scope of this paper. Performance results, shown in Section 4, were obtained by computer simulations. Moreover, the complexity values of the proposed approach (expressed in MIPS) are also shown in Section 4 not only for the overall system but also for each constituent stage.

3 System Description

A block diagram of the proposed low-complexity and high-accuracy approach for speech/nonspeech discrimination in DSP-based hearing aids is depicted in Figure 3. The input audio signal is analyzed using the analysis window and half-window overlapping scheme for F0 estimation. At each texture window, F0 estimates are processed to compute high-level music-related features, which are then applied to the classification stage. The classification scheme consists of two constitutive elements that operate in series. First, a low-complexity classifier evaluates the high-level music-related features and computes the probability the current audio segment to be speech or nonspeech. The HMM postprocessing step is included to provide valuable information from past audio frames to the classification stage. Therefore, the proposed classification scheme, composed of a low-complexity classifier followed by HMM postprocessing, incorporates memory to the speech/nonspeech discriminator, which allows to increase the classification accuracy rate, as shown in Section 4.

Next, the main blocks of the proposed approach for speech/nonspeech discrimination in DSP-based hearing aids are described.

3.1 Decimated Difference Function for F0 Estimation

F0 can be estimated in both time and frequency domains. The autocorrelation function (ACF) and its modifications are the generalized way of computing the fundamental period in the time domain [29]. ACF-based algorithms tend to estimate an integer multiple of the fundamental period, because the analyzed signal is also periodic for all integer multiples of the fundamental period. However, the main inconvenient of time-domain algorithms is their inability of handling multiple F0 estimation (a very common situation, for example, in western music). In general, multipitch signals are not periodic enough, while time-domain algorithms are just based on periodicity.

Frequency-domain algorithms are more robust for multipitch estimation [30]. Frequency-based techniques perform pitch estimation from the Fourier transform of the audio signal, which is composed of a train of delta functions for real-world periodic sounds. Frequency-based algorithms search for delta functions equidistant in frequency to estimate F0. Although some multipitch estimators are able to detect more than one F0 present at the same time, multipitch estimation is still an open research field [31, 32], and the solutions are often computationally very complex.

Taking complexity constraints into account, F0 estimation in digital hearing aids should be performed at the lowest possible computational cost. A very sim-

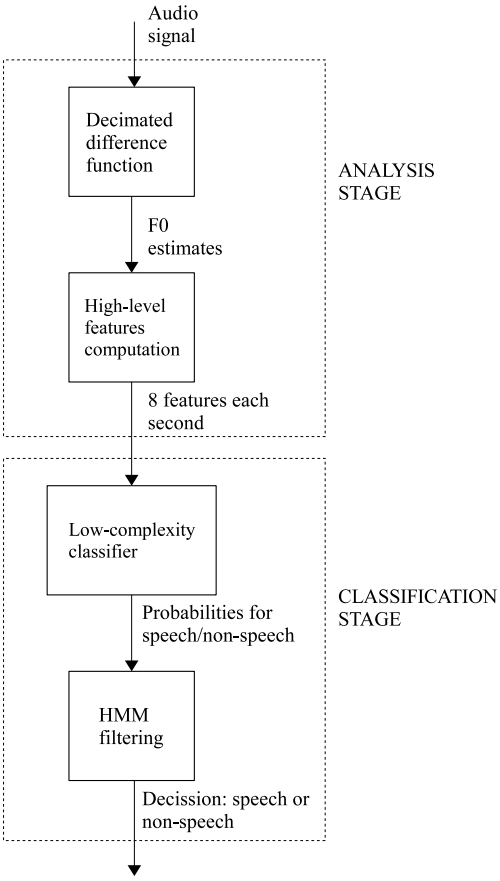

Figure 3: Block diagram of the proposed approach for speech/nonspeech discrimination in hearing aids.

ple method to estimate the fundamental frequency of a signal $x(n)$ relies on the following property: a periodic signal fulfills $x(n) = x(n + \tau_0)$, where τ_0 is the period. However, the same property is also fulfilled for all integer multiples of the fundamental period. This property is exploited by the difference function $df(n, \tau)$, which can be defined as follows [29]:

$$df(n,\tau) = \sum_{l=0}^{W-1} \left(x(n+l) - x(n+l+\tau)\right)^2, \tag{1}$$

where n is the time index, τ the delay of the difference function and W the length of the analysis window. The minimum value of the difference function $df(n, \tau)$ arises at the fundamental period (and its integer multiples) of $x(n)$. The difference

function is directly related with the ACF function $r(n,\tau)$ in the following way:

$$df(n,\tau) = r(n,0) + r(n+\tau,0) - 2r(n,\tau). \tag{2}$$

The complexity of the difference function is proportional to the analysis window length W and the sampling frequency f_s of the input audio signal. As stated in [29], the parameter W should be chosen at least as high as the maximum fundamental period to be estimated. This requirement implies a high complexity in the computation of the difference function $df(n,\tau)$, specially when dealing with low pitched signals, such as speech. In our implementation, the analysis window length has been fixed to 20 ms, which allows to estimate fundamental frequencies above 50 Hz.

In [33], a low-complexity method for F0 estimation in digital hearing aids is proposed. The proposed method in [33] computes the difference function at some outputs of the filterbank incorporated into the DSP-based hearing aid, showing promising but not good enough results. Complexity reduction for difference function computation can also be achieved by decimating the input audio signal. In this way, the number of sums and multiplications for each index value τ is reduced according to the decimation factor.

In this work, we intend to reduce the complexity of the difference function as much as possible, while maintaining the estimation accuracy. The solution proposed here is to redefine the difference function by applying a decimation factor, which results in the so-called *decimated difference function*:

$$ddf(n,\tau) = \sum_{l=0}^{S-1} \big(x(n + d(W,\tau)l) - x(n + d(W,\tau)l + \tau)\big)^2, \tag{3}$$

where S is the number of samples used to compute the decimated difference function and $d(W,\tau)$ is the applied decimation factor. In order to use the same window length W when computing $ddf(n,\tau)$ for all delays, the decimation factor must be a function of the window length W and the delay τ, as expressed in Equation (4):

$$d(W,\tau) = \left\lfloor \frac{W - \tau - 1}{S - 1} \right\rfloor. \tag{4}$$

Note that Equation (4) is derived by supposing that the highest delay for $ddf(n,\tau)$ must be less or equal than $W - S$. Moreover, the decimation factor, as defined in Equation (4), avoids sample selection out of the current audio frame. Figure 4 illustrates the influence of the decimation factor when evaluating the decimated difference function for any index value τ.

The decimated difference function $ddf(n,\tau)$ requires S subtractions, S multiplications and $S-1$ additions for each output value. Therefore, the computation

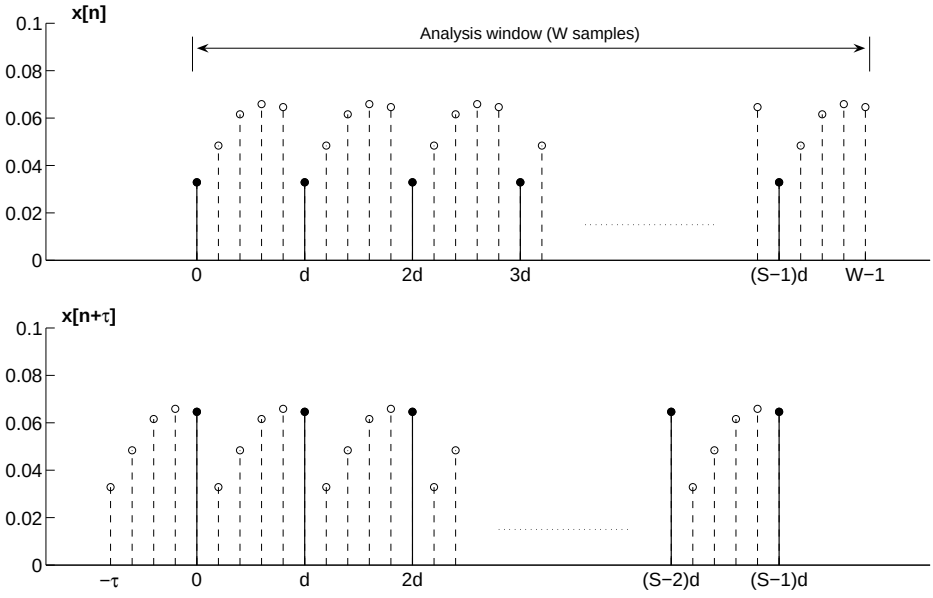

Figure 4: Example illustrating how the decimated difference function is computed for a given index value τ. Samples considered for computation are depicted in black. As illustrated, the decimation factor allows to select S equally spaced samples within the analyzed audio frame.

of each output value is now directly proportional to parameter S. A trade-off solution between complexity and estimation reliability can be achieved by properly selecting parameter S, as seen in the results.

A modification of the difference function $df(n, \tau)$ is proposed in [29], aiming to avoid typical errors in estimating the fundamental period. For this reason, the cumulative mean normalized difference function $cmdf(n, \tau)$ is defined as follows:

$$cmdf(n, \tau) = \frac{\tau \cdot df(n, \tau)}{\sum_{j=1}^{\tau} df(n, j)}. \tag{5}$$

The cumulative mean normalized difference function is introduced in order to better discriminate the fundamental period from its integer multiples [29]. Note that all these periods lead to local minima of the difference function $df(n, \tau)$. However, the cumulative mean normalized difference function reinforces the local minimum due to the fundamental period in relation to its integer multiples. As a consequence, in our approach, the *cumulative mean normalized decimated*

difference function, denoted by $cmddf(n,\tau)$, has been considered:

$$cmddf(n,\tau) = \frac{\tau \cdot ddf(n,\tau)}{\sum_{j=1}^{\tau} ddf(n,j)}. \quad (6)$$

The computation of the cumulative mean normalized decimated difference function $cmddf(n,\tau)$ does not almost increase implementation complexity. In addition to the operations of function $ddf(n,\tau)$, it requires a multiplication, a division and a summation to compute each output value. Summarizing, S subtractions, $S+1$ multiplications, S additions and one division are required to calculate each output value of function $cmddf(n,\tau)$.

The complexity of evaluating the function $cmddf(n,\tau)$ for all time indexes n and delay values τ is too high to be implemented in ultra low-power digital hearing aids, and should be further reduced. It is expected that decimation in both dimensions has a great impact on the final complexity. Nevertheless, function $cmddf(n,\tau)$ should be decimated according to the application we are dealing with. Logically, all n values at a high enough sampling rate ($f_s = 16,000$ Hz) are not required to obtain good speech/nonspeech discrimination results. With regard to the delay, it initially ranges from $\tau = 4$ to $\tau = W - S$, where W is the length of the analysis window. Delays from $\tau = 1$ to $\tau = 3$ lead to inaccurate estimations for low frequency signals, because the shifted signal is very similar to the original one [29]. As a consequence, the upper bound for F0 estimation is fixed to $f_s/4$, with f_s being the sampling frequency of the input signal.

In our implementation, the 20 ms-length analysis window contains $W = 320$ samples at $f_s = 16,000$ Hz sampling rate. Further, half-window overlapping is performed, which involves evaluating function $cmddf(n,\tau)$ each $T_h = 10$ ms. Taking into account that $(W - S - 3)/T_h$ delays (or samples) per second are evaluated by function $cmddf(n,\tau)$, we can calculate the number of sums, multiplications and divisions per second required by our F0 estimation method. Summarizing, the complexity requirements of function $cmddf(n,\tau)$ are the following:

- $2S(W - S - 3)/T_h$ sums (additions and subtractions) per second.
- $(S + 1)(W - S - 3)/T_h$ multiplications per second.
- $(W - S - 3)/T_h$ divisions per second.

The overall complexity of function $cmddf(n,\tau)$, taking multiply-accumulate (MAC) operations into account, is equal to $(2S + 4)(W - S - 3)/T_h$ MIPS.

In Section 4, the system performance has been assessed with different values of parameter S in order to find a good balance between complexity and accuracy rate in the speech/nonspeech discrimination task.

Next, the method for estimating the fundamental frequency at a given time n_0 from function $cmddf(n, \tau)$ is described:

1. For all possible values of index τ, the minimum of function $cmddf(n_0, \tau)$ is calculated.

2. This minimum value is compared with a threshold in order to discriminate between voiced and unvoiced frames [29]. When it is far from zero, the signal frame is not periodic and the fundamental period cannot be estimated. Otherwise, the signal frame is labeled as voiced and the fundamental period is estimated. The threshold is here fixed to 0.15 [29].

3. When the current audio frame is labeled as voiced, the estimated fundamental period is the delay τ_0 for which function $cmddf(n_0, \tau)$ takes the minimum value.

Finally, the method provides three values for the current audio frame:

- The estimated F0. It is the inverse of the estimated fundamental period ($\mathrm{F0} = 1/\tau_0$).

- Aperiodicity, denoted by $Ap0$, which is defined as the value of function $cmddf(n_0, \tau)$ at the estimated fundamental period, $Ap0 = cmddf(n_0, \tau_0)$. As stated before, F0 estimation is not valid when this parameter is above 0.15, the analyzed signal being considered to be unvoiced. A normalized measure of aperiodicity Ap ranging from 0 to 1 can also be used.

- Power of the windowed discrete-time signal, denoted by P. When this parameter is below a threshold (2^{-15} for normalized signals), the signal is considered to be a silence.

3.2 Music-Related Features Computed from F0 Estimation

When dealing with speech signals, F0 estimates match to a characteristic pattern for most of the analyzed signals. Speech signals contain voiced frames (near-periodic) and unvoiced frames (aperiodic), which alternate in time. In most of languages, words are composed of voiced and unvoiced phonemes, which results in several voiced-unvoiced boundaries within a word. Good F0 estimates can be accomplished for voiced frames, while it makes no sense to estimate F0 for unvoiced frames. Moreover, voiced speech frames have a time-varying F0, because the pitch changes when voiced phonemes are pronounced. Instead, music and noise (nonspeech) signals show a quite changing behavior. There is not a generic

pattern for such signals. Their properties depend on several factors, such as the music genre, polyphony, instruments involved, type of noise, etc [28]. However, two specific patterns can be identified in music signals: 1) F0 does not almost change when only one musical note is played at any time (near-steady state within the same note), and 2) step-wise changes often happen when passing from a musical note to another. Noisy environments, in turn, tend to remain unvoiced most of the time, although they often exhibit short pitched parts, or long pitched parts with near-steady F0 (for instance, in pitched stationary noises).

In general, although there is not a generic pattern for nonspeech sounds, they differ from speech signals in two main aspects: 1) the absence of intonation (i.e. the time-varying F0 that arises from the pronunciation of voiced phonemes), and 2) the absence of the typical alternation of voiced and unvoiced frames that results from the articulation of words (composed of voiced and unvoiced phonemes). In order to better illustrate these differences, Figures 5 and 6 are included. In Figure 5 an example of F0 estimation for a representative speech signal is shown. As can be seen, F0 estimation (in voiced frames) slowly varies in terms of the speaker's intonation, and the $Ap0$ sequence exhibits large variations between voiced and unvoiced frames. Figure 6 shows the F0 estimated for a music signal. As can be seen, F0 estimation is nearly-steady in voiced frames, and the $Ap0$ sequence exhibits smaller variations than those observed in speech signals.

In order to capture these main differences, a set of musically-inspired features derived from F0 were proposed in [28] for speech/music discrimination. Here, we employ these features for speech/nonspeech classification in hearing aids.

Next, the set of F0-based features, proposed by Ruiz-Reyes et al. in [28], is briefly defined:

1. *Dynamic range of aperiodicity* (D_{Ap}). It is defined as the difference between the maximum and minimum values of the normalized aperiodicity Ap within the current texture window.

2. *Average of F0 estimates* ($F0_{av}$). It is defined as the mean value of F0 estimates at the current texture window.

3. *Dynamic range of F0 estimates* (D_{F0}). It is defined as the difference between the maximum and minimum values of F0 estimates within the current texture window.

4. *Maximum note duration* (ND_{max}). It is defined from the number of consecutive analysis windows comprising the longest musical note within the observation interval (the current texture window).

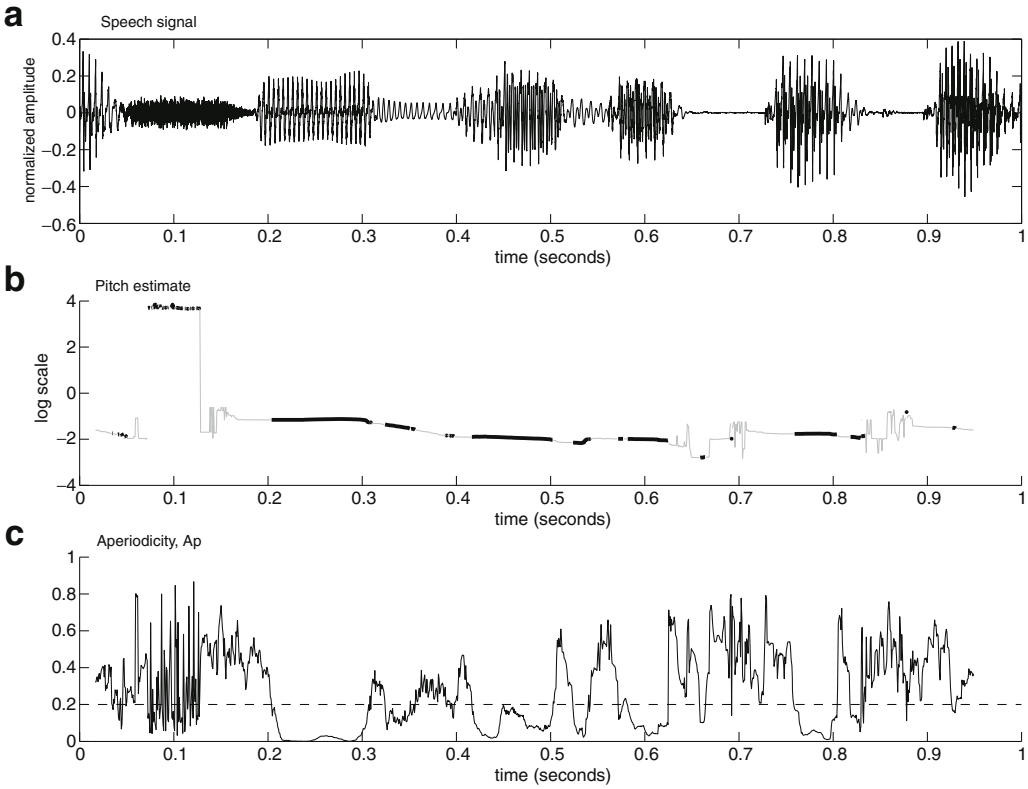

Figure 5: F0 estimate for a representative speech signal of 1 s. (a) Normalized waveform. (b) Estimated F0. The thick line corresponds to the segments that are "classified" as voiced. (c) Aperiodicity. The dashed line represents the boundary to "classify" the signal frames as voiced or unvoiced.

5. *Number of notes* (N_{note}). It is defined as the number of different notes contained within the observation interval (the current texture window). From F0 estimates in the observation interval, we compute how many different notes are detected.

6. *Voiced ratio* (VR). It is defined as the ratio between the number of voiced frames and the total number of frames within the observation interval. This parameter informs us about the percentage of frames in which F0 is properly estimated at each observation interval.

7. *Average value of the aperiodicity* ($Ap0_{av}$). Mean value of the normalized aperiodicity at the current texture window. It is only defined for voiced frames, and informs us about the periodicity of voiced frames.

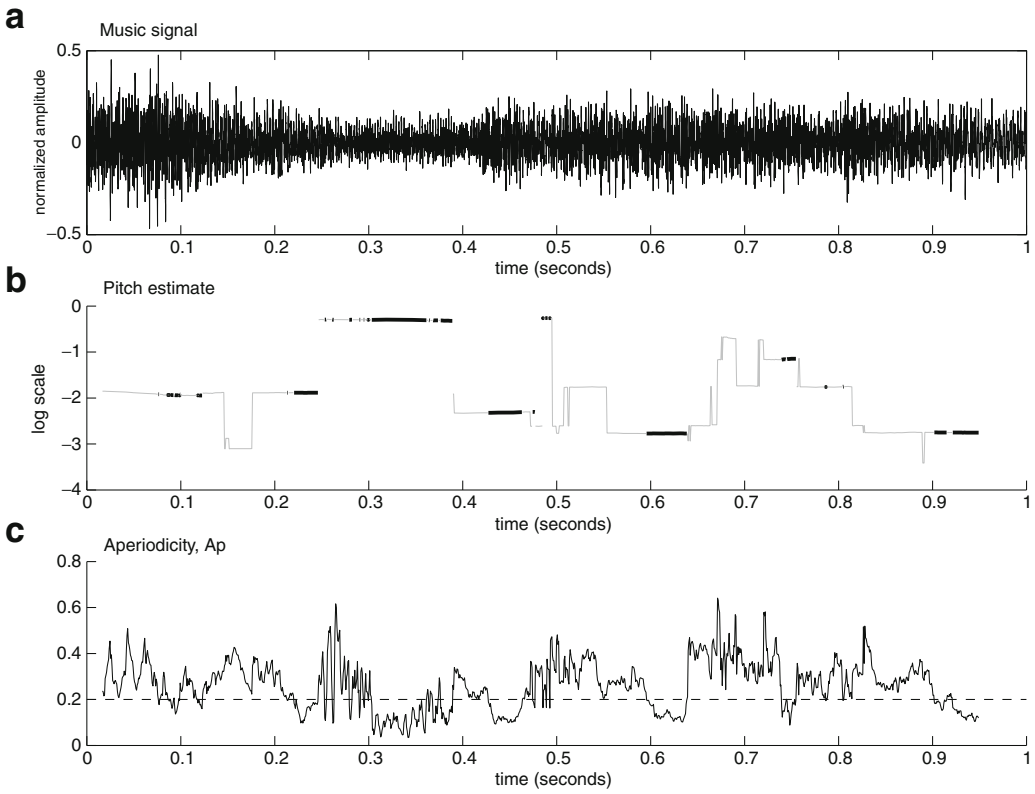

Figure 6: F0 estimate for a music signal of 1 s. (a) Normalized waveform. (b) Estimated F0. The thick line corresponds to the segments that are "classified" as voiced. (c) Aperiodicity. The dashed line represents the boundary to "classify" the signal frames as voiced or unvoiced.

8. *Aperiodic power* (Apw). It is defined as the ratio between the power of unvoiced frames and the total power at the current texture window. It informs us about the percentage of power due to unvoiced frames.

To illustrate the discrimination ability of some of these features, Figure 7 is included. The normalized histograms in Figure 7 correspond to features N_{note} and Apw when they are computed using both the proposed F0 estimation method and the YIN algorithm. In subplots (a) and (c), high values of parameter N_{note} are obtained for speech signals, because F0 estimates change with the speaker's intonation. Rather, lower values of parameter N_{note} are usually obtained for non-speech signals, because F0 estimates remain steady in variable-length intervals. As shown in subplots (b) and (d), the parameter Apw (aperiodic power) is usually low for speech signals, because unvoiced frames have typically less power than voiced frames. This situation does not happen for noise or music signals.

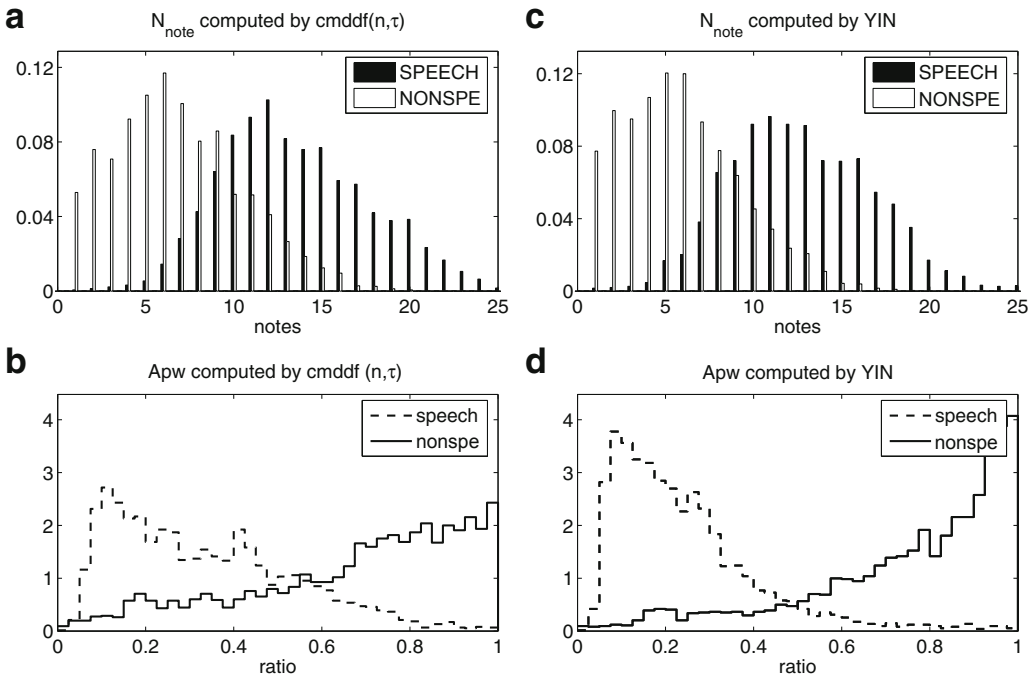

Figure 7: Normalized histograms of features N_{note} and Apw for both speech and non-speech in the following cases: (a) feature N_{note} is computed using the proposed method for F0 estimation, (b) feature Apw is computed using the proposed method for F0 estimation, (c) feature N_{note} is computed using the YIN algorithm, (d) feature Apw is computed using the YIN algorithm.

Further details about the motivation of the F0-based features and their typical behavior when classifying speech/nonspeech can be obtained in [28].

When all analysis windows in a texture window are labeled as either unvoiced or silence, the previously described features have no sense, and a boolean flag is activated to inform the classification stage about it.

The required complexity for computing the just-described features is much lower than the computational cost of the proposed method for F0 estimation. The main reason is that the music-related features are computed each 250 ms, while F0 estimation is performed each 10 ms. The complexity values of these music-related features are shown in Section 4 (Table 3), being derived from their definition in [28].

3.3 Low-Complexity Classifier

To achieve high accuracy rates, the classifier parameters are previously found by performing a training process with a suitable sound database. After training, the

classifier is able to classify new input patterns with satisfactory results. There exist a high number of approaches for pattern recognition, each one with different characteristics in terms of performance and complexity. Among them, several low-complexity classifiers have been proposed in the literature for automatic sound classification in digital hearing aids [1, 9]. For such application, it is crucial not only to achieve a high accuracy rate, but also to keep the computational and memory requirements as low as possible.

In this work, two different classifiers have been considered for evaluation: MLP and the tree-based C4.5 classifier. The choice of the MLP classifier, among a variety of algorithms proposed in the literature, is motivated by the fact that neural networks (NNs) have proven to exhibit a proper learning behaviour for sound classification problems. As pointed out in [7], [9] and [10], NNs are able to achieve very good results in terms of classification accuracy rate when compared to other widely used algorithms, such as rule-based classifiers, the Fisher linear discriminant, the k-Nearest Neighbor algorithm or Bayessian classifiers. Furthermore, in [28], it was proved that NNs, evaluated over the set of music-related features employed in this work, provided better accuracy rates than other recent and powerful classification schemes, such as Support Vector Machines.

A feasible alternative to the MLP classifier, mainly when accounting for the computational cost, is the tree-based C4.5 classifier. Although well-known, this classifier has not been yet investigated for sound classification tasks, and thus no performance results have been reported in the literature on this particular issue. For this reason, and attracted by its low computational requirements, we have evaluated the C4.5 classifier along with the MLP, with the aim of choosing the optimal solution for its application in hearing aids. In addition, both classifiers have also been compared with the classic k-Nearest Neighbor (k-NN) algorithm, which is commonly used in the literature for comparison purposes [1, 9]. Here, given its high computational and memory requirements, the k-NN classifier is only considered as an "anchor", and not as a feasible option to be implemented in a hearing aid device.

Next, the three considered classifiers are briefly described:

- *k-Nearest Neighbor classifier (k-NN).* The k-NN classifier needs to store in memory the whole training set to compute all possible distances, thereby requiring high computation time and memory. Although the k-NN classifier involves high computational cost, it is typically used as an "anchor" for comparison with feasible classifiers [10]. In this work, a 1-NN classifier is considered as an "anchor" for comparison purposes

- *Multi-layer Perceptron* (MLP). In this work, a three-layer MLP with 8 input neurons, 8 hidden neurons and 2 output neurons is considered for evalua-

tion. Such a network requires 80 multiplications and additions to provide an output value. Moreover, it requires 90 memory-words to store the MLP parameters (the weights of the links and the neuron bias values). An additional number of operations equal to 200 is also required to compute the sigmoid activation function at hidden and output layer neurons (20 operations per neuron) [26]. MLPs have been previously used for automatic sound classification in digital hearing aids [9].

- *Tree-based C4.5 classifier*. It builds decision trees from the training set using the concept of information entropy [20]. Given a set T of training vectors, the attribute and splitting value that provide the highest normalized information gain (difference in entropy) are used to divide set T into two subsets, T_L and T_R. The same process is recursively applied on each subset until a stopping criterion is satisfied. As a result, a binary tree with n nodes and l leaves is obtained. For each node, two values are memory-stored: an index identifying the attribute and the splitting value for that attribute. For each leaf, two values are also stored: the class (or decision) associated to the leaf, and a probability value of right classification. Therefore, $2n + 2l$ memory-words are required to store the whole tree in the device. Once the tree is built, the complexity of the classifier is proportional to the tree depth (a tree depth equals to D involves D comparisons to classify a new instance).

Complexity information regarding the three considered classifiers is reported in Section 4 (Table 4). The complexity values in Table 4 arise from the theoretical definition of each considered classifier.

3.4 HMM Postprocessing Stage

As defined in [18], a Hidden Markov Model (HMM) is a double stochastic process with an underlying stochastic process that is not observable, but can only be observed through another set of stochastic processes that produce the sequence of observations. In other words, a HMM is a mathematical description of a system which may be described at any time as being in one of a set of J distinct states, and which changes its state at discrete times according to certain probabilities. The model is named "hidden" because the state of the model, $q(t)$, at a given time instant t, is not directly observable. Instead, only an indirect output value (or set of values), $o(t)$, which is a probabilistic function of the actual state, can be observed. HMMs constitute a useful tool for modeling time-varying processes, being widely used in applications such as speak recognition or speaker verification [19].

Although the probabilities returned by our classifier can be directly used to take a final decision, usually the decision sequence contains isolated errors. These errors are specially undesirable, because they make the device to change its hearing program for short time intervals, severely reducing the comfort level experienced by the user. Typically, the environment surrounding the user exhibits a rather steady behavior; a given state (lets say "speech") is held for a certain period of time, and then, at a certain instance, the environment changes to another state (lets say "nonspeech"), which in turn remains steady during another time interval. In order to incorporate this inherent temporal information into the speech/nonspeech discrimination problem, we model the environment with an HMM, which post-processes the probabilities returned by the classifier.

In the proposed approach, a two-states HMM is employed, where each state corresponds to a different sound class (state 1 for "speech" and state 2 for "nonspeech"). Therefore, there are only four transition probabilities, which correspond to the following state transitions: speech/speech, speech/nonspeech, nonspeech/speech and nonspeech/nonspeech. The model receives as input data (observation) the sequence of probabilities computed by the classifier, $O = [p_j(1), p_j(2), p_j(3), ..., p_j(T)]$, with $j \in \{1, 2\}$, and provides the optimal state sequence $Q = [q(1), q(2), q(3), ..., q(T)]$ associated to the received observation. Estimation of the optimal state sequence (optimal path) is accomplished by the Viterbi algorithm [19].

Once the classifier provides the probabilities $p_j(T)$ for the current texture window, the HMM post-processing stage has immediately to make the final decision. No backtracking has to be performed for retrieving previous states, $[q(1), q(2), ..., q(T-1)]$, since they were already estimated at previous texture windows.

Although the Viterbi algorithm can work accounting for information only from past signal segments, more accurate results can be achieved by incorporating some future information to the model. However, this accuracy gain is achieved at the expense of increasing the system delay. In Section 4, the proposed low-complexity speech/nonspeech discrimination approach is evaluated when HMM postprocessing is performed for different delay values. Moreover, influence of the delay on the considered application (automatic sound classification in digital hearing aids) is further discussed.

It can be demonstrated that the HMM post-processing stage involves low computational and memory requirements, being negligible with respect to those of the F0 estimation stage. Complexity values of HMM post-processing are included in Section 4 (Table 4). However, in order to make the paper as concise and clear as possible, justifying complexity values is out of the scope of this work.

4 Experimental Setup and Results

4.1 Experimental Setup

The sound database used for the experiments consists of 2936 files, with a length of 2.5 seconds each one. The coding parameters of the audio files are the following: 16000 Hz sampling frequency and 16 bits per sample. The audio files belong to the following categories: 1964 speech files, 366 noise files and 606 music files. The first category is subdivided into two classes: speech in quiet and speech in noise. The speech in noise files have different SNR values, ranging from 0 to 20 dB. Noise files include the following classes: aircraft, bus, cafe, car, kindergarden, living room, nature, school, shop, sports, traffic, train and train station. Finally, music files include two classes: vocal music and instrumental music. This database has been previously used for automatic sound classification in digital hearing aids [9, 2].

The classification results are calculated using a ten-fold cross-validation evaluation, where the dataset to be evaluated is randomly partitioned so that 10% is used for testing and 90% is used for training. The process is iterated with different random partitions and the results are averaged. The results presented in this section are obtained with 50 iterations. This ensures that the calculated accuracy will not be biased because of a particular partitioning of the whole dataset for training and testing.

In the experimental setup, system parameters are configured as follows:

- F0 estimation:
  - 20 ms-length analysis window.
  - Half-window overlapping. It implies that F0 estimates are obtained each 10 ms.
  - F0 estimation ranges from 50 Hz to $f_s/4 = 4$ kHz.
- Music-related features computation:
  - 1 s-length texture window. The texture window comprises $L = 99$ analysis windows.
  - The texture window is shifted each 250 ms.
- Low-complexity classifier:
  - The simplest possible configuration is used for the k-NN classifier. Therefore, the considered classifier is 1-NN (only one nearest neighbor).

- A three-layers configuration with 8-8-2 neurons is considered for the MLP-based classifier. A sigmoid activation function is applied to neurons at hidden and output layers.
- The C4.5 algorithm is executed with the following configuration values: *minimum number of instances per leaf* equal to 2 and *confidence factor* equal to 0.25. These values are recommended in [20].

• HMM filtering:

- Prior probabilities are fixed to 0.5 (speech and nonspeech probabilities are supposed to be the same).
- It is supposed that state transitions happen each 120 seconds on average. From this value, the transition matrix is derived.

The last block is disabled when testing on the above described database, because the audio files are too short to get advantage of HMM postprocessing. However, a signal of about one hour has been recorded from a radio broadcasting program (with speech and nonspeech parts) to demonstrate the effectiveness of HMM filtering.

4.2 Accuracy Results

First, we are going to assess the proposed F0 estimation method for different values of the parameter S, with the aim of obtaining an optimum value. Table 1 shows the mean accuracy rates provided by the proposed speech/nonspeech discrimination approach (excluding HMM postprocessing) for different values of the parameter S and different classifiers. Results in Table 1 are particularized for each sound class, so that the columns "Speech" and "Non-speech" express, respectively, the percentage of speech and nonspeech texture windows correctly classified. The column "Global" expresses the global accuracy rate, i.e. the percentage of texture windows (either speech or nonspeech) correctly classified. Table 1 also shows the performance of the proposed approach when F0 is estimated by the YIN algorithm [29]. Comparison with the YIN algorithm aims to evaluate the accuracy loss due to the decimated difference function (proposed F0 estimation method).

From Table 1, it can be stated that nonspeech frames are more frequently misclassified than speech frames when using low values of parameter S. Nonspeech frames have a more heterogeneous nature than speech ones (different levels of polyphony and different pitched and non-pitched sources). This property makes nonspeech frames more sensitive to F0 estimation errors due to low values of parameter S. However, the speech class exhibits a more steady performance with

Table 1: Assessing the proposed F0 estimation method for different values of the parameter S and different classifiers. Comparison with the YIN algorithm.

F0 estim. method	Accuracy rate with MLP (%)			Accuracy rate with C4.5 (%)			Accuracy rate with 1-NN (%)		
	Global	Speech	Non-speech	Global	Speech	Non-speech	Global	Speech	Non-speech
$S = 10$	87.54	89.35	83.88	85.73	89.05	79.01	81.28	86.99	69.73
$S = 20$	88.54	89.85	85.88	86.58	89.03	81.63	83.75	88.29	74.57
$S = 30$	88.76	89.59	87.07	87.46	89.24	83.86	85.14	89.11	77.12
$S = 40$	89.17	89.86	87.77	87.73	89.53	84.09	86.12	89.85	78.59
$S = 50$	89.42	89.63	88.98	87.99	88.93	86.09	86.01	88.84	80.29
$S = 100$	89.01	89.06	88.89	88.25	89.03	86.67	87.33	89.04	83.88
YIN algorithm	90.13	90.03	90.35	88.95	89.64	87.56	88.41	90.01	85.18

respect to the parameter S. With higher values of S, the global accuracy rate is increased and classification errors are more fairly distributed among speech and nonspeech classes.

The global accuracy rates in Table 1 are also shown in Figure 8 to better understand the performance of the function $cmddf(n, \tau)$ with respect to the parameter S. As shown in Figure 8, global accuracy rates tend to grow as the parameter S is increased, regardless of the considered classifier.

With regard to the classifier, the following comments arise from Figure 8. MLP achieves the best results for all values of the parameter S, followed by the tree-based C4.5 algorithm and the 1-NN classifier. These results are in line with other related previous works, where neural networks have demonstrated to achieve higher accuracy rates than other classifiers [1, 9].

Another meaningful result that arises from Figure 8 is the following: the accuracy loss between the decimated difference function (proposed F0 estimation method) and the YIN algorithm is reduced to 0.5% in the best case (MLP and $S = 55$). This result evinces the good performance of the proposed F0 estimation method when combined with MLP for speech/nonspeech discrimination in digital hearing aids.

4.3 Complexity Evaluation

In order to choose an optimum value of parameter S and the more suitable classifier for the application we are dealing with, system complexity requirements must be taken into account. In such sense, Table 2 shows the number of instructions required to implement the proposed F0 estimation method for different values of the

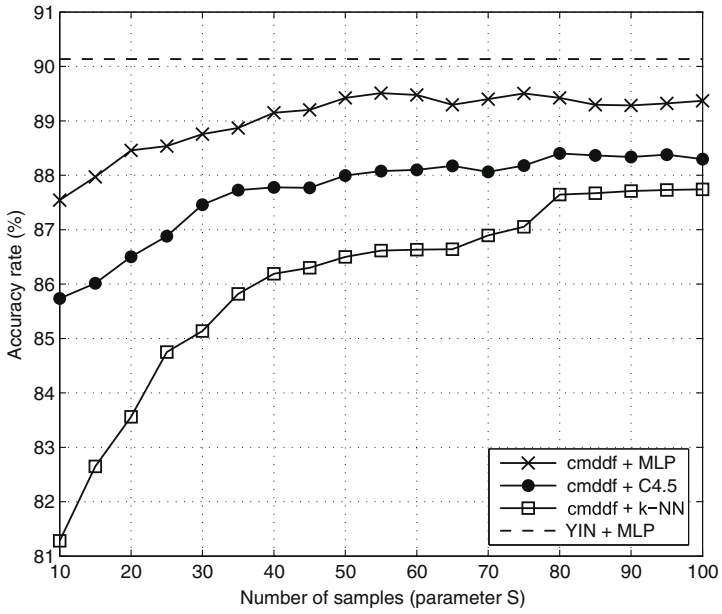

Figure 8: Performance of the function $cmddf(n, \tau)$: classification accuracy rate vs parameter S.

parameter S in the *Toccata Plus* DSP system from On Semiconductor. In Table 2, only the complexity of the decimated difference function (proposed F0 estimation method) is considered. Complexity details of such function are included in Section 3.1. The results in Table 2 are obtained by supposing that the *Toccata Plus* DSP is able to perform a simple operation (summation, MAC, multiplication, division) in one single instruction.

Complexity requirements for the remaining stages of the proposed low complexity speech/nonspeech discrimination approach, namely music-related features computation and classification, are summarized in Tables 3 and 4.

Table 3 shows the number of operations per texture window required to compute each one of the music-related features. In Table 3, complexity values are expressed as a function of the parameter L, which denotes the number of F0 estimates within a single texture window. According to our experimental setup, $L = 99$. The number of instructions per second, as stated in the last column, is obtained by supposing that the music-related features are computed 4 times per second (the texture window is shifted by 250 ms), and that the logarithm operation takes 16 instructions in the processor.

Table 2: Complexity of the decimated difference function (proposed F0 estimation method) for different values of the parameter S.

F0 estimation method	Instructions per second
Proposed one, S = 10	736,800 (0.73 MIPS)
Proposed one, S = 20	1,306,800 (1.30 MIPS)
Proposed one, S = 30	1,836,800 (1.83 MIPS)
Proposed one, S = 40	2,326,800 (2.32 MIPS)
Proposed one, S = 50	2,776,800 (2.77 MIPS)
Proposed one, S = 100	4,426,800 (4.42 MIPS)
YIN algorithm	20,415,200 (20.41 MIPS)

Table 3: Complexity requirements of music-related features.

Feature	Sums per texture window	Multiplications or divisions per texture window	Comparisons per texture window	Logarithms per texture window	Instructions per second ($L = 99$)
D_{Ap}	1	0	$2L - 1$	0	792
$F0_{av}$	$L - 1$	1	0	0	396
D_{F0}	1	0	$2L - 1$	0	792
ND_{max}	L	$2L$	L	L	7,920
N_{note}	0	0	$\frac{L(L-1)}{2}$	0	19,404
VR	$L - 1$	1	L	0	792
$Ap0_{av}$	$L - 1$	1	0	0	396
Apw	$L - 1$	1	0	0	396
All	$5L - 2$	$2L + 4$	$\frac{L^2}{2} + \frac{11L}{2} - 2$	L	30,888

Table 4 shows the complexity requirements of each considered classifier. It also shows the complexity of HMM post-processing. The following assumptions were made for the three considered classifiers:

- Complexity of the MLP is obtained by supposing that the activation function at each neuron takes 20 instructions [2].
- It is supposed that the decision tree depth for the C4.5 classifier is $D = 12$. Actually, the tree depth depends on the training process, and cannot be configured a priori. However, during the training process, 12-depth decision trees were very often obtained. That is the reason for choosing $D = 12$ as a suitable value for the tree depth.

Table 4: Complexity requirements of the classification stage. Comparison between the three considered classifiers.

Classifier	Sums per decision	Multiplic. per decision	MAC per decision	Comparisons per decision	Instruct. per second
MLP	0	0	80	0	1,120
C4.5	0	0	0	D	48
1-NN	$8T$	0	$8T$	$T-1$	1,257,996
HMM	0	6	0	3	36

- Complexity of the 1-NN classifier is proportional to the size of the training set, which consists of $T = 18500$ feature vectors for the considered database.

From Table 4, it results that MLP and C4.5 classifiers lead to a low computational cost in comparison to the 1-NN classifier. Although the complexity of the C4.5 classifier is much lower than that of MLP, the complexity values in both cases are negligible compared to those of the F0 estimation stage. Therefore, the classification stage of the proposed approach can be regarded as a low-complexity stage, even when implemented with a neural network. Taking into account that MLP provides higher accuracy rates than C4.5 with lower complexity, we have chosen MLP as the optimum classifier for the application we are dealing with.

The main conclusion from Tables 2, 3 and 4 is the following: the complexity of the proposed speech/nonspeech discrimination approach for hearing aids is mainly due to the F0 estimation stage. The remaining stages do not almost increase the system complexity, as shown in Tables 3 and 4. Note that all stages are executed each 250 ms, except the F0 estimation stage, which is executed each 10 ms. Therefore, the value of the parameter S has a great impact on the overall complexity of the proposed approach and must be properly chosen to match the overall complexity to the constraints of the *Toccata Plus* DSP.

Figure 9 shows the overall complexity of the proposed approach for speech-nonspeech discrimination in digital hearing aids as a function of the parameter S. The overall complexity results in Figure 9 are obtained under the assumption that classification is performed by a MLP-based classifier.

As shown in Figure 9, complexity values higher than 45 avoid the algorithm to be implemented on the chosen device (*Toccata Plus* DSP system for hearing aids), because the algorithm complexity outperforms the maximum computational capacity of the DSP system. Moreover, the parameter S must be below 20 when the system is operating at 1.28 MIPS, which is a configuration intended for ultra low-power consumption in order to extend the battery operation time.

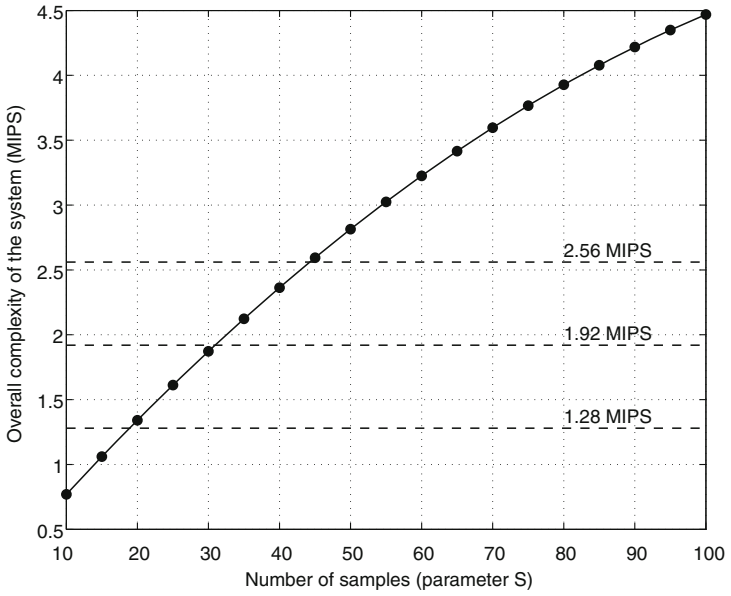

Figure 9: Complexity requirements of the proposed approach as a function of parameter S.

An adequate selection for the parameter S can be made by analyzing Figures 8 and 9. From these figures, it results that a good choice for S is around 30 samples. Higher values ($S > 30$) do not almost improve the accuracy rate, while complexity is meaningfully increased.

4.4 Evaluation of HMM Postprocessing

For assessing the benefits of HMM postprocessing, a one hour-length radio broadcasting program that alternates speech and nonspeech intervals was downloaded (www.rtve.es/resources/mp3/2/0/1222050144702.mp3). The input file is first classified by the MLP-based classifier. The result is then filtered by the HMM stage with different delay values. Table 5 shows the improvement in the classification accuracy rate when HMM filtering is included in the proposed scheme. Moreover, Table 5 also shows how the delay influences the classification accuracy rate. Delay values ranging from 0 to 2.5 seconds are here considered.

The global accuracy rate is increased about 1% when HMM postprocessing is incorporated into the speech/nonspeech discrimination scheme. Higher accuracy rates can be achieved if a certain decision delay is allowed. However, the accuracy rate only increases with the decision delay until an upper bound is reached

Table 5: Performance evaluation of the HMM postprocessing stage. Comparison for different delay values.

Classification scheme	Accuracy rate	Decision delay (s)
MLP	92.86 %	0
MLP + HMM (no delay)	93.87 %	0
MLP + HMM (1 window shift)	94.56 %	0.25
MLP + HMM (2 windows shift)	94.93 %	0.5
MLP + HMM (3 windows shift)	95.09 %	0.75
MLP + HMM (4 windows shift)	95.17 %	1
MLP + HMM (5 windows shift)	95.20 %	1.25
MLP + HMM (6 windows shift)	95.21 %	1.5
MLP + HMM (7 windows shift)	95.21 %	1.75
MLP + HMM (8 windows shift)	95.21 %	2
MLP + HMM (9 windows shift)	95.22 %	2.25
MLP + HMM (10 windows shift)	95.23 %	2.5
MLP + HMM (infinite delay)	95.23 %	∞

(95.23% for the selected one hour-length file). As shown in Table 5, a decision delay of about 1 s can be chosen as an optimum value.

The results in Table 5 are obtained when $S = 30$ is chosen. Therefore, the accuracy rate is somewhat higher for the one hour-length file (92.86% with MLP) than for the audio database containing 2.5 second-length files (88.76% with MLP). The results in Table 5 highlight how critical the considered sound database is for speech/nonspeech discrimination.

As explained in Section 3.4, HMM postprocessing avoids that speech/nonspeech decision bounces in consecutive texture windows. In order to illustrate the benefits of HMM postprocessing, Figure 10 is included.

As seen in Figure 10, HMM postprocessing performs time-filtering on decisions taken by the classifier, thus providing a more stable output. This property makes HMM postprocessing very useful in digital hearing aid applications, because switching between speech/nonspeech decisions in short intervals causes annoying effects to hearing aids users. Time-filtering is more effective when the model operates with a certain delay, since the output is computed considering some future decisions. In the example of Figure 10, it is shown how HMM postprocessing with one second-delay eliminates all isolated errors at the expense of increasing the system latency in responding to environment transitions.

For the application we address in this paper (speech/nonspeech discrimination in hearing aids), a few seconds delay is often considered as an acceptable latency. For instance, the approach proposed by Nordqvist and Leijon [8] takes about 2-10

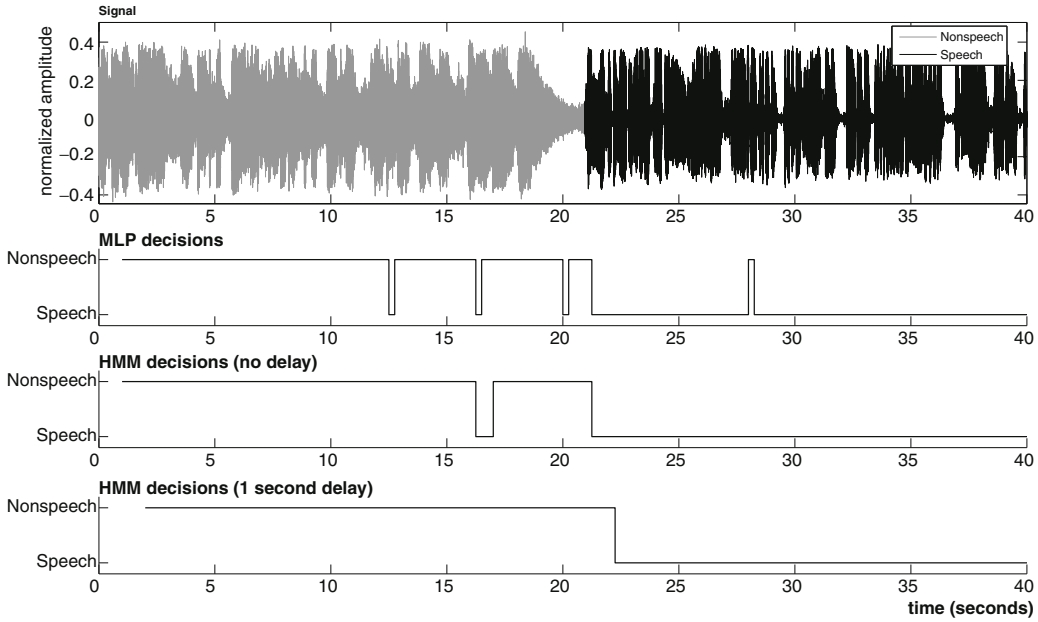

Figure 10: An example of speech/nonspeech discrimination before and after HMM processing. In the upper subplot, a transition between nonspeech (grey line) and speech (black line) is depicted. The second subplot shows the decisions taken by the MLP-based classifier. The third subplot shows the decisions after HMM postprocessing (with no delay). Finally, in the bottom subplot, the decisions taken by the HMM model operating with one s-delay are depicted.

seconds to change its output after a transition from one listening environment to another occurs. In the approach by Büchler et al. [1], the output of the classifier is observed over a certain time (typically, 10 seconds), and the class that more often appears in that time interval is taken as a result.

4.5 Comparison and Discussion

Several approaches dealing with the problem of speech/nonpeech discrimination in hearing aids have been proposed in the literature [2, 10, 26]. The approach in [2] employs a pattern classifier with two layers, where the first layer classifies the audio signal into speech or nonspeech using a set of features selected by a genetic algorithm. For this approach, an accuracy rate of about 93% is reported. In [26], speech/nonspeech discrimination is performed using a set of several spectral features, obtaining an accuracy rate equal to 86.7%. The speech/nonspeech discriminator described in [10] makes use of a tailored NN to perform the sound

classification, reporting an accuracy rate of about 90% with a NN of moderated complexity. These results evidence that the proposed approach is competitive in comparison with recently published algorithms dealing with the same problem. In addition, apart from these quantitative comparisons, several qualitative advantages can be appreciated in our approach.

In our approach, we have focused on the efficient computation of a "general purpose" feature, namely fundamental frequency, and on its application for sound classification in hearing aids. It is widely known that F0 has a broad range of applications in digital processing of audio signals [29]. In the context of hearing aids, several concurrent applications can benefit from an efficient estimation of F0, such as speech enhancement, noise canceling or adaptive filtering, which are topics of great interest for improving the comfort level of hearing aids users. Supposing that all these applications were based on F0, the computational effort due to F0 estimation would be shared by all of them, and the additional cost due to sound classification can be considered negligible with respect to the global computational cost. Further, in our approach, sound classification is conducted over long-term F0-based features, which leads to a very low complexity.

Another advantage of the presented algorithm relies on its flexibility. In the proposed approach, F0 estimates are provided by a decimated difference function, which depends on parameter S to select different levels of complexity. With more powerful computational resources, higher values of S can be selected, thus obtaining better results. Moreover, the proposed approach is not at all dependent on the decimated difference function, and other methods for F0 estimation (for instance, YIN) might be used if strong hardware constraints were not imposed. In the future, as the memory and computational power increases in low-power DSPs, more accurate F0 estimation methods will be available to be implemented in digital hearing aids.

In summary, we may point out the following advantages of the proposed approach: 1) it achieves accuracy results that are in line with previous published works, 2) it is based on an efficient F0 estimation, which is a desirable feature for other concurrent applications in hearing aids, and 3) it is a feasible approach to be implemented into current hearing devices, and flexible enough to provide a better performance if higher computational resources are available.

5 Conclusions and Future Work

In this paper a low-complexity speech/nonspeech discrimination approach for digital hearing aids is proposed. The proposed approach mainly relies on a low-complexity method for F0 estimation, which consists on computing a decimated

difference function. The proposed speech/nonspeech discrimination scheme is completed with a feature extraction stage (music-related features), a low-complexity classifier and a HMM postprocessing. The complexity of the proposed discrimination approach is mainly due to the F0 estimation stage. The remaining stages do not almost increase system complexity.

Classification accuracy rates are analyzed together with the complexity requirements in order to select the more appropriate classifier and an optimum value of parameter S. In such sense, a MLP-based classifier is selected, and $S = 30$ is a good trade-off value between accuracy and complexity. The proposed speech/nonspeech discrimination scheme is feasible to be implemented in ultra low-power DSP-based digital hearing aids by choosing the suitable configuration setup. Parameter S must be below 20 when the system is intended to operate at 1.28 MIPS, in order to extend the battery operation time (ultra low-power consumption).

The accuracy loss between the decimated difference function (proposed F0 estimation method) and the YIN algorithm is reduced to only 1% in our configuration setup (MLP and $S = 30$). This result evinces the good performance of the proposed F0 estimation method when combined with a MLP-based classifier for speech/nonspeech discrimination in digital hearing aids. The global accuracy rate is increased about 1% when HMM postprocessing is incorporated into the speech/nonspeech discrimination scheme. Higher accuracy rates can be achieved if a certain decision delay is allowed. From experimental results, a 1-s delay is chosen as an optimum value, the classification accuracy rate being about 95%.

Fundamental frequency estimation has a wide range of potential applications in digital hearing aids. Speech intelligibility improvement in digital hearing aids from F0 estimation will be explored in the next future.

References

[1] M. Büchler, S. Allegro, S. Launer, and N. Dillier, "Sound classification in hearing aids inspired by auditory scene analysis", *EURASIP Journal on Applied Signal Processing*, vol. 18, pp. 2991–3002, 2005.

[2] E. Alexandre, L. Cuadra, M. Rosa, and F. López-Ferreras, "Feature Selection for Sound Classification in Hearing Aids Through Restricted Search Driven by Genetic Algorithms", *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2249–2256, Nov. 2007.

[3] B. C. Moore, *An Introduction to the Psychology of Hearing*, 3rd ed. New York: Academic, 1989.

[4] F. Luo and A. Nehorai, "Recent Developments in Signal Processing for Digital Hearing Aids", *IEEE Signal Processing Magazine*, vol. 23, no. 5, pp. 103–106, Sept. 2006.

[5] G. Keidser, "The relationships between listening conditions and alterative amplification schemes for multiple memory hearing aids", *Ear Hear*, vol. 16, pp. 575–586, 1995.

[6] G. Keidser, "Selecting different amplification for different listening conditions", *J. of the American Academy of Audiology*, vol. 7, pp. 92–104, 1996.

[7] M. Büchler, "Algorithms for sound classification in hearing instruments", PhD Thesis, Swiss Federal Institute of Technology, Zurich, 2002.

[8] P. Nordqvist and A. Leijon, "An efficient robust sound classification algorithm for hearing aids", *J. Acoust. Soc. Amer.*, vol. 115, no. 6, pp. 3033–3041, 2004.

[9] E. Alexandre, L. Cuadra, L. Alvarez, and M. Rosa-Zurera, "NN-based automatic sound classifier for digital hearing aids", in *Proc. of IEEE Int. Symposium on Intelligent Signal Processing*, Alcalá de Henares, Spain, October 2007.

[10] E. Alexandre, L. Cuadra, M. Rosa, and F. López-Ferreras, "Speech/nonspeech classification in hearing aids driven by tailored neural networks", *Speech, Audio, Image and Biomedical Signal Processing Using Neural Networks*, B. Prasad and S. M. Prassana Eds., pp. 145–167, Springer, Berlin, Germany, 2008.

[11] R. Gil-Pita, E. Alexandre, L. Cuadra, R. Vicen, and M. Rosa-Zurera, "Analysis of the Effects of Finite Precision in Neural Network-Based Sound Classifiers for Digital Hearing Aids", *EURASIP Journal on Advances in Signal Processing*, doi:10.1155/2009/456945, 2009.

[12] E. Alexandre, M. Rosa-Zurera, L. Cuadra, and R. Gil-Pita, "Application of fisher linear discriminant analysis to speech/music classification", in *Proc. of the 120th Audio Engineering Society Convention*, Paris, France, 2006, vol. 2, pp. 1666–1669.

[13] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator", in *Proc. IEEE ICASSP'97*, Munich, Germany, 1997, pp. 1331–1334.

[14] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia applications", in *Proc. IEEE ICASSP'2000*, 2000, vol. 6, pp. 2445–2448.

[15] H. Harb and L. Chen, "Robust speech music discrimination using spectrum's first order statistics and neural networks ", in *Proc. IEEE Int. Symp. on Signal Processing and Its Applications*, 2003, vol. 2, pp. 125–128.

[16] J. Benesty, S. Makino, and J. Chen, *Speech enhancement*, Springer, ISBN 354024039X, 2005.

[17] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama, "Single and Multiple F_0 Contour Estimation Through Parametric Spectrogram Modeling of Speech in Noisy Environments", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1135–1145, May 2007.

[18] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models", *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 3, no. 1, pp. 4–16, January 1986.

[19] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", in *Proc. of the IEEE*, February 1989, vol. 77, no. 2, pp. 257–286.

[20] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[21] J. R. Quinlan, "Improved use of continuous attributes in C4.5", *Journal of Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.

[22] T. Rohdenburg, V. Hohmann, and B. Kollmeier, "Robustness analysis of binaural hearing aid beamformer algorithms by means of objective perceptual quality measures", in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, October 2007.

[23] On Semiconductor, "Toccata Plus Flexible DSP System for Hearing Aids", http://www.amis.com/products/dsp/toccata_plus.html.

[24] On Semiconductor, http://www.onsemi.com.

[25] L. Cuadra, E. Alexandre, R. Gil-Pita, R. Vicen, and L. Álvarez, "Influence of Acoustic Feedback on the Learning Strategies of Neural Network-Based Sound Classifiers in Digital Hearing Aids", *EURASIP Journal on Advances in Signal Processing*, doi:10.1155/2009/465189, 2009.

[26] E. Alexandre, L. Cuadra, L. Álvarez, and M. Utrilla, "Exploring the feasibility of a two-layer NN-based sound classifier for hearing aids", in *Proc. EUSIPCO 2007*, Poznań, Poland, September 2007.

[27] R. Dong, D. Hermann, E. Cornu and E. Chau, "Low-power implementation of an HMM-based sound environment classification algorithm for hearing aid application", in *Proc. EUSIPCO 2007*, Poznań, Poland, September 2007.

[28] N. Ruiz-Reyes, P. Vera-Candeas, J. E. Muñoz, S. Garcia-Galán, and F. J. Cañadas, "New speech/music discrimination approach based on fundamental frequency estimation", *Multimedia Tools and Applications*, vol. 41, pp. 253–286, January 2009.

[29] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music", *Journal of the Acoustic Society of America (JASA)*, vol. 111, no. 4, pp. 1917–1930, April 2002.

[30] A. Klapuri, "Multiple fundamental frequency estimation by harmonicity and spectral smoothness", *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

[31] A. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model", *IEEE Trans. Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 255–266, Feb. 2008.

[32] W. C. Chang, W. Y. Alvin Su, Y. Chunghsin, A. Röbel, and X. Rodet, "Multiple-F0 tracking based on a high-order HMM model", in *Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, September 2008.

[33] P. Vera-Candeas, F. J. Cañadas-Quesada, E. Alexandre, M. Rosa, and N. Ruiz-Reyes, "Musical-inspired features for automatic sound classification in digital hearing aids", in *Proc. of 124th Audio Engineering Society Convention*, Amsterdam, The Netherlands, May 2008.

Paper B

Voicing Detection based on Adaptive Aperiodicity Thresholding for Speech Enhancement in Non-stationary Noise

P. Cabañas-Molero, D. Martínez-Muñoz, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Rodríguez-Serrano, "Voicing detection based on adaptive aperiodicity thresholding for speech enhancement in non-stationary noise", in *IET Signal Processing*, Volume 8, Issue 2, April 2014, pp. 119–130.

Abstract

In this study, the authors present a novel voicing detection algorithm that employs the well-known aperiodicity measure to detect voiced speech in signals contaminated with non-stationary noise. The method computes a signal-adaptive decision threshold which takes into account the current noise level, enabling voicing detection by direct comparison with the extracted aperiodicity. This adaptive threshold is updated at each frame by making a simple estimate of the current noise power, and thus is adapted to fluctuating noise conditions. Once the aperiodicity is computed, the method only requires a small number of operations, and enables its implementation in challenging devices (such as hearing aids) if an efficient approximation of the difference function is employed to extract the aperiodicity. Evaluation over a database of speech sentences degraded by several types of noise reveals that the proposed voicing classifier is robust against different noises and signal-to-noise ratios. Additionally, to evaluate the applicability of the method for speech enhancement, a simple F_0-based speech enhancement algorithm integrating the proposed classifier is implemented. The system is shown to achieve competitive results, in terms of objective measures, when compared with other well-known speech enhancement approaches.

1 Introduction

Voicing detection (also referred to as voiced/unvoiced classification) is the process of determining whether a short-time speech segment is produced by a significant vibration of the vocal cords [1]. Voiced speech sounds are usually more or less periodic (e.g., when pronouncing a vowel or a semi-vowel), whereas unvoiced segments are typically noise-like, and include both speech pauses (possibly with background noise) and unvoiced phonemes. Classification of speech into voiced and unvoiced is of great interest in many speech processing applications, such as speech coding, speech analysis/synthesis, automatic speech recognition or speech enhancement.

In the context of speech enhancement, voicing detection becomes essential in all those approaches based on fundamental frequency F_0 estimation. Essentially, these approaches attempt to estimate the F_0 and voicing state of the speech signal from its noisy observation, and exploit the harmonic structure of voiced speech to enhance the quality of the signal. This harmonicity-based enhancement is usually performed by extracting frequency components at integer multiples of F_0, for example, with adaptive comb filtering [2] or sinusoidal synthesis [3]. In this class of algorithms, robust and accurate F_0 estimation and voicing detection

are essential issues, because any error in these tasks causes severe deterioration in the achieved speech quality. For example, errors in F_0 estimation may distort the target speech signal, while inaccuracies in voicing detection may cause losing voiced segments. Voicing detection must not be confused with voice activity detection (VAD), whose purpose is to determine the presence of speech (including voiced and unvoiced) and which is usually implemented in conjunction with traditional speech enhancement methods (which do not require voicing classification), such as spectral subtraction [4], statistical model-based algorithms [5] or subspace approaches [6].

Although pitch estimation and voicing detection are practically considered as a closed problem in clean speech, these tasks are still challenging under adverse noise conditions. Concerning voicing detection, several methods have been proposed during the last three decades to address the problem. Typical approaches focus on extracting acoustic parameters that reflect the characteristics of voiced speech, such as zero crossing rate, short-time energy, autocorrelation peaks or cepstral coefficients [7, 1]. Some methods make the voicing decision at the time that F_0 is estimated [8], for example, measuring properties associated with periodicity [9, 10]. In all these typical methods, voicing decisions are taken by setting thresholds on parameters values, or by means of pattern recognizers trained on these features. The main problem here is that the optimal thresholds (or training) depend critically on the signal-to-noise ratio (SNR), and the performance degrades with different noise levels or with non-stationary noises. For this reason, several adaptive methods have been proposed for making voicing detection under varying noise conditions. The method in [11] estimates adaptively the probability density function of correlation peak values, and derives the optimal threshold for voicing detection in non-stationary noise. In [12] a voicing decision algorithm is proposed for the ETSI speech coding standard ES 202 211, which employs an adaptive VAD and several signal features for classifying each frame into nonspeech, unvoiced, mixed voiced or fully voiced. In [13] a novel measure of voicing is defined based on the computation of a robust dominance spectrum. In [14] an adaptive system based on noise classification is presented, which uses a VAD and a neural network to identify the type of noise and the SNR before taking a voicing decision.

Among the traditional estimation methods, the YIN algorithm proposed in [10] is one of the most accurate F_0 and voicing estimators in absence of background noise. Although not intended to work under noise, the YIN algorithm has been proven to perform relatively well for pitch estimation on voiced speech under a wide range of noisy environments, such as white noise and some real acoustic interferences [15] (it does not happen for noisy environments containing periodic components). This good performance is due to the considerable robustness of the normalized difference function, in which the location of the local minima remains

approximately unchanged if the interfering noise has a broadband and relatively flat spectrum. However, despite its robustness in F_0 estimation accuracy, YIN exhibits severe weaknesses when applied to voicing detection in noise. Typically, the aperiodicity value provided by the algorithm is a good measure for voicing detection in clean signals, providing an accurate voicing detection by a fixed threshold. However, as just mentioned, when the signal is corrupted by noise, this threshold must be changed depending on the noise level, which is time-varying and not always known in advance. Therefore, if an inappropriate threshold is selected, YIN will consider a noisy voiced frame as unvoiced, even if the correct pitch period (i.e. local minimum) has been estimated.

In this paper, we propose an algorithm for voicing detection based on aperiodicity thresholding which can work in low SNR environments and in the presence of different types of non-stationary noises. The proposed method is also based on the computation of the cumulative mean normalized difference function, and hence the aperiodicity measure is extracted in the same way as YIN. However, unlike the YIN algorithm, where a fixed threshold is employed, the proposed method updates the so-called "aperiodicity threshold" at each frame, making use of a rough estimation of the underlying noise power. This noise power is continuously estimated from previously detected silence or voiced parts, under the assumption that the noise remains stationary within short intervals. The performance of the method is evaluated on a database of speech sentences degraded with several real acoustic noises at different SNRs. Comparative results with the original YIN algorithm demonstrate that the proposed method obtains a better performance for voicing detection, and show that it is competitive (and more accurate in certain cases) compared to the voicing detector of the ETSI 202 211 standard. Also, to evaluate the applicability of the method for speech enhancement, we implement a simple F_0-based speech enhancement algorithm which relies on the proposed voicing detector. The implemented scheme, which is constructed with simple and well-known signal processing techniques, is evaluated in comparison with several existing speech enhancement approaches, showing competitive results in terms of different objective measures.

An important feature of the proposed method is that, except for the conventional difference function, it is efficient enough to be implemented in ultra-low power devices, such as digital hearing aids. Generally, this kind of devices imposes severe constraints in terms of computational capacity, which avoid the implementation of common signal processing algorithms, including the conventional difference function. In a previous work [16], we proposed an algorithm to compute an accurate approximation of the difference function which reduces drastically the number of required operations. This approximation was shown to achieve good performance when applied to sound classification in hearing aids.

In this work, we extend the experimental evaluation given in [16] by implementing this efficient difference function within the proposed voicing detector. The results demonstrate that this efficient implementation remains effective for voicing detection and speech enhancement, showing only a moderate degradation in comparison to the standard difference function.

The novelty of this work lies in the computation of a signal-adaptive threshold linked to the aperiodicity measure for voicing detection in non-stationary noise. Additionally, we show experimentally that, when the efficient difference function proposed in [16] is employed, the method maintains effectiveness, and allows its implementation in digital hearing aids. The paper is organized as follows. In Section 2, the proposed voicing classifier is explained in detail. In Section 3, a simple F_0-based speech enhancement algorithm is designed to illustrate the applicability of the method for speech denoising. In Section 4, the accuracy of the proposed voicing classifier is evaluated, and the results are compared with the voicing decision system in the ETSI 202 211 standard. Comparative evaluation of the method for the speech enhancement task is also addressed. In both cases, the version of the method incorporating the efficient difference function is also evaluated. Finally, Section 5 contains the main conclusions of the paper and directions for future research.

2 Voiced-Unvoiced Classification

2.1 Signal-adaptive Aperiodicity Threshold

In order to classify each frame as voiced or unvoiced, a robust decision method against different noise levels and types of noise is here applied. As with the YIN algorithm, the proposed method is also based on the computation of the cumulative mean normalized difference function, but here the aperiodicity value is compared to a dynamic threshold that takes the presence of background noise into account for voicing decision.

The classic difference function for a time-domain signal x is defined as

$$d_x(t,\tau) = \sum_{m=1}^{W} (x[t+m] - x[t+\tau+m])^2, \tag{1}$$

where τ is the delay, t the time reference index and W the window size. The relation between the autocorrelation and the difference function is given by $d_x(t,\tau) = r_x(t,0) + r_x(t+\tau,0) - 2r_x(t,\tau)$. As a consequence, the minimum value of the difference function is zero and is obtained for $\tau = 0$. For other values of parameter τ, the minimum value of $d_x(t,\tau)$ occurs when the delayed signal is the most

similar to the original one. In the case of periodic signals, the difference function has several zeros at delays equal to the period and its integer multiples.

An interesting property of the difference function can be derived from the identity $2(x^2[t] + x^2[t+\tau]) = (x[t] + x[t+\tau])^2 + (x[t] - x[t+\tau])^2$. Averaging over a window, we can obtain the following relation [10]:

$$2r_x(t,0) + 2r_x(t+\tau,0) = s_x(t,\tau) + d_x(t,\tau), \tag{2}$$

where $s_x(t,\tau)$ and $d_x(t,\tau)$ are the summation and difference functions, which estimate the periodic and aperiodic powers of the signal, respectively.

The cumulative mean normalized difference (CMND) function is defined from the difference function as [10]

$$cmnd_x(t,\tau) = \begin{cases} 1, & \text{if } \tau = 0 \\ d_x(t,\tau) \Big/ \Big[(1/\tau) \sum_{j=1}^{\tau} d_x(t,j)\Big], & \text{if } \tau > 0 \end{cases} \tag{3}$$

The numerator of (3) is proportional to the aperiodic power, whereas the denominator is approximately twice the signal power [10]. Thus, $cmnd_x(t,\tau)$ is proportional to the aperiodic/total power ratio.

According to [10], the estimated period is obtained as the delay of the first local minimum in the CMND function below a fixed threshold (if none is found, the global minimum is used). This threshold is also used to discriminate between voiced and unvoiced frames. For voiced frames, the signal is quasi-periodic and the CMND function has local minima close to zero at integer multiples of the period. On the contrary, for unvoiced frames the signal is usually not similar to its delayed copies and therefore the local minima of the CMND function are far from zero. The value of the CMND function for the estimated period is called *aperiodicity*. When the aperiodicity value is below the absolute threshold, the frame is labeled as voiced. Otherwise, the frame is labeled as unvoiced.

Let us explain the selection of this threshold in more detail. For quasi-periodic signals of period T, the periodic power is going to be much higher than the aperiodic power at $\tau = T$. In terms of the CMND function, its value at the period T can be approximated by

$$cmnd_x(t,T) \approx \frac{d_x(t,T)}{2r_x(t,0)}. \tag{4}$$

Using the power decomposition between periodic and aperiodic terms given in (2), we can rewrite (4) as follows:

$$cmnd_x(t,T) \approx \frac{2d_x(t,T)}{s_x(t,T) + d_x(t,T)}. \tag{5}$$

Since x is quasi-periodic, we can suppose that $s_x(t,T) > Sd_x(t,T)$, where S is the number of times that, at least, $s_x(t,T)$ must be greater than $d_x(t,T)$ for considering a given signal as quasi-periodic. By substitution, a maximum value for the CMND function can be obtained as a condition of periodicity:

$$cmnd_x(t,T) < \frac{2}{S+1}. \tag{6}$$

A threshold ranging from 0.1 to 0.2 (from $S = 19$ to $S = 9$) is recommended in [10] for accurate F_0 estimation and voicing detection. In Figures 1a and 1b, examples of the CMND function for voiced and unvoiced frames are depicted. We can see the local minima due to the fundamental period (and its integer multiples) for the voiced frame. For the unvoiced frame, the global minimum is above the threshold, which is here fixed to 0.2.

When a speech signal $s[t]$ is corrupted by additive noise, $x[t] = s[t] + n[t]$, the CMND function does not always exhibit local minima close to zero for voiced frames. Supposing that the noise is uncorrelated with the voiced speech, the autocorrelation function of the noisy signal is obtained by summing the autocorrelation functions of the speech and noise components. In terms of autocorrelation, the initial value $r_x(t,0)$ is the energy of the speech plus the noise, $r_x(t,0) = r_s(t,0) + r_n(t,0)$. Besides, when signal $s[t]$ is periodic with period T, the difference function at $\tau = T$ of the noisy signal is $d_x(t,T) = d_n(t,T)$, because $d_s(t,T) = 0$, and the value of the CMND function at $\tau = T$ is given as

$$cmnd_x(t,T) \approx \frac{d_n(t,T)}{2(r_s(t,0) + r_n(t,0))}. \tag{7}$$

We also suppose that the value of the autocorrelation function of the noise signal $n[t]$ at delay $\tau = T$ is much lower that the value at the origin, $r_n(t,T) << r_n(t,0)$, and that the interfering noise does not contain periodic components. In such case, taking into account the relation between difference function and autocorrelation, the difference function of the noise component at $\tau = T$ can be approximated by $d_n(t,T) \approx 2r_n(t,0)$, which is valid for most types of real noises. Therefore, (7) is rewritten as follows:

$$cmnd_x(t,T) \approx \frac{r_n(t,0)}{r_s(t,0) + r_n(t,0)}. \tag{8}$$

As stated above, (8) is only valid for periodic signals. For the case of quasi-periodic signals, the CMND function can be approximated as

$$cmnd_x(t,T) \approx \frac{d_s(t,T)/2 + r_n(t,0)}{r_s(t,0) + r_n(t,0)}. \tag{9}$$

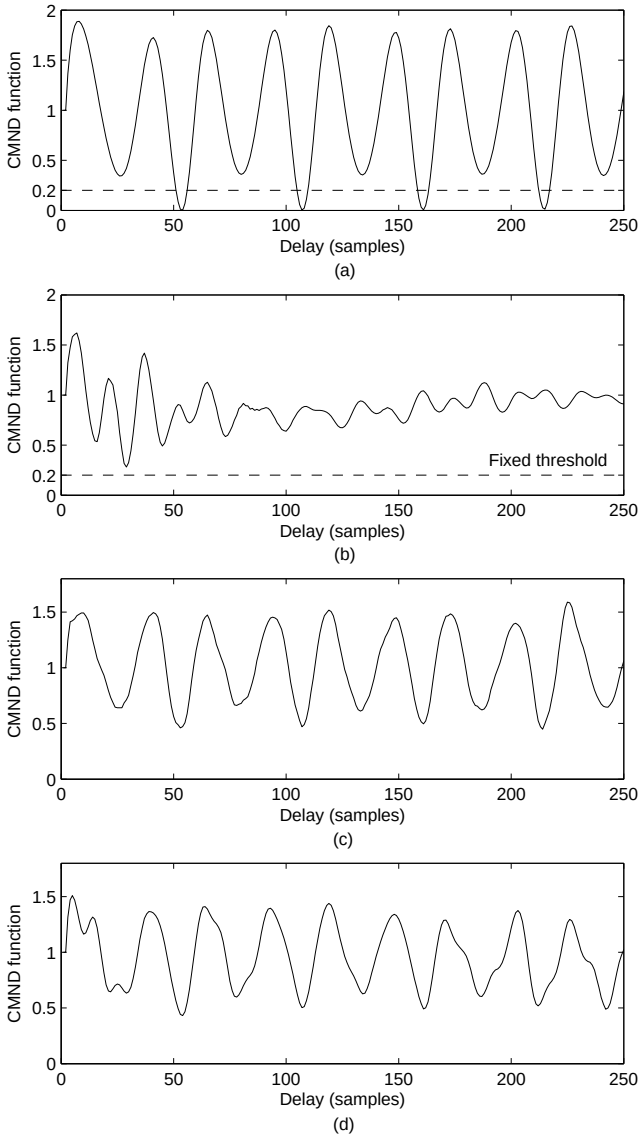

Figure 1: Representation of the CMND function for a speech signal: (a) clean voiced frame, (b) clean unvoiced frame, (c) voiced frame + white noise with the same energy as speech, (d) voiced frame + car noise with the same energy as speech.

Finally, since the relation between periodic and aperiodic power in (2) can be expressed as $4r_s(t,0) \approx s_s(t,T)+d_s(t,T)$, and considering the aforementioned condition of periodicity, $s_s(t,T) > Sd_s(t,T)$, we can write $d_s(t,T) < \frac{4}{S+1} r_s(t,0)$. Hence, for quasi-periodic signals affected by noise, the CMND function for $\tau = T$

should be below the following threshold:

$$cmnd_x(t,T) < \frac{\frac{2}{S+1} r_s(t,0) + r_n(t,0)}{r_s(t,0) + r_n(t,0)} = \alpha(t). \tag{10}$$

As can be seen, this threshold depends on the powers of signal and noise in the current frame. Threshold $\alpha(t)$ is a signal-adaptive voiced/unvoiced threshold, unlike the fixed one proposed in [10]. In order to compute this threshold, an estimation of the noise and speech powers in the current frame is needed. In the proposed approach, the noise power is dynamically estimated from the noisy signal by using the simple algorithm described in Section 2.2. The speech power, assuming that speech and noise are uncorrelated, can be obtained as $r_s(t,0) = r_x(t,0) - r_n(t,0)$.

A remarkable aspect of (10) is that, when the power of the speech signal is considerably lower than the noise power, the computed threshold tends to 1. In practice, real acoustic noises produce aperiodicity values lower than 1 and, in consequence, noise-only frames may potentially be regarded as voiced according to (10). To overcome this problem, we establish a minimum SNR value, SNR_{min}, as a condition to compute $\alpha(t)$. If the estimated SNR in the current frame is below SNR_{min}, the frame is labeled directly as unvoiced, without the need for computing $\alpha(t)$. Otherwise, threshold $\alpha(t)$ is employed as usual to perform voicing detection. In our configuration, the minimum allowed SNR value is set to $SNR_{min} = -6$ dB, since it was found that this value obtains the best results in our experiments across several types of noise and SNRs.

Figure 1c illustrates the effect of white additive noise over the CMND function in a voiced frame. This speech frame is the same as in Figure 1a. As can be seen, the local minima are now far from zero. According to (8), since speech and noise have the same energy here, the local minima should have values close to 0.5. Similarly, Figure 1d depicts CMND function when noise from a car is added to the speech signal. This example illustrates the effect on the CMND function when the signal is corrupted by a real acoustic noise.

2.2 Noise Power Estimation

In order to evaluate (10), it is necessary to know the noise power $r_n(t,0)$ in the current frame. To determine this noise power, we propose an estimation algorithm that is based on similar principles as those applied in the so-called minimum statistics approaches [17]. Basically, the proposed procedure is based on two main assumptions. First, it is assumed that the background noise is rather stationary within a finite time interval (or window), and thus, its properties in the current frame can be considered quite similar to those in immediately preceding

frames. Secondly, it is assumed that, within this time interval, the speech utterance produces (among other sounds) almost-periodic frames or almost-silence frames. This assumption rests on the observation that, during speech activity, either brief speech pauses (inserted between the words and syllables) or highly periodic voiced speech frames (which approximate quite well to a perfectly periodic signal) can be found in short periods of time, accompanied or not by other speech sounds (such as fricatives or plosives).

Taking these assumptions into account, we can derive a simple algorithm to compute a rough estimate of $r_n(t,0)$. First, we define a sliding window containing the last W_T processed frames. To allow an effective tracking of the noise power, the length of this window has to be relatively short, but long enough to ensure the presence of voiced or silence frames. In our simulations, we found good performance using a sliding window of length a few tenths of a second. Within this window, the global minimum of the CMND function is computed as follows:

$$ap_{min}(t) = \min_{\substack{m\in[t-W_T+1,t] \\ \tau\in[T_{min},T_{max}]}} cmnd_x(m,\tau), \tag{11}$$

where T_{min} and T_{max} are the minimum and maximum expected values for the pitch period of speech (we set the pitch range between 30 and 500 Hz). Let t_{min} be the time-frame index corresponding to this global minimum. If the frame t_{min} was labeled as voiced, we can suppose that this global minimum is the result of a highly periodic signal, and hence, according to (8), the noise power estimate $\hat{r}_n(t,0)$ can be computed as

$$\hat{r}_n(t,0) = ap_{min}(t) r_x(t_{min},0). \tag{12}$$

Since the current window may not contain any voiced frame, or all voiced frames may be far from periodicity (which leads to an overestimate of the noise power), we also compute the minimum $P_{min}(t)$ of the total power $r_x(t,0)$ across the window, under the assumption that this value may correspond to a silence frame. From both estimates, $P_{min}(t)$ and $\hat{r}_n(t,0)$, the final noise power estimate is given by

$$\hat{r}'_n(t,0) = \min\{\hat{r}_n(t,0), P_{min}(t)\}. \tag{13}$$

This estimate is replaced in (10) to compute the signal-adaptive voicing threshold for the current frame.

2.3 Hidden Markov Model Post-processing

Although the thresholds $\alpha(t)$ and SNR_{min} can be directly used to take a final decision, usually frame-by-frame decisions lead to isolated errors, which have a great

effect on the output quality. Typically, the speech signal is voiced or unvoiced in the same phoneme and, consequently, a given voicing state is held for a certain number of frames. In order to incorporate this inherent temporal dependence, we model the behavior of the speech signal with a hidden Markov model (HMM).

In the proposed approach, a three-state HMM is employed, where each state corresponds to a different sound class, namely silence, voiced and unvoiced. The probabilities corresponding to each state (at time instant t) are defined respectively as follows:

$$p_{silence}(t) = 1 - 0.5^{(r_n(t,0)/r_s(t,0))SNR_{min}} \tag{14}$$

$$p_{voiced}(t) = \big(1 - p_{silence}(t)\big)0.5^{ap_0(t)/\alpha(t)} \tag{15}$$

$$p_{unvoiced}(t) = 1 - p_{silence}(t) - p_{voiced}(t), \tag{16}$$

where $ap_0(t)$ is the current aperiodicity. Note that the silence state is introduced to cope with those situations in which the speech power is much lower than the noise power, in which case $\alpha(t)$ may be not reliable for voicing detection, as discussed earlier. In (14), a probability of silence $p_{silence}(t)$ greater than 0.5 is obtained whenever the estimated SNR is below SNR_{min}. The remaining probability is distributed among voiced and unvoiced states depending on $\alpha(t)$ and $ap_0(t)$. These state probability equations were chosen experimentally based on the observation of experimental data. Observe that, rather than estimating the likelihoods of the states using Gaussian Mixture Models (which require to train the parameters of the gaussians), likelihoods are here directly computed by equations (14)-(16) from the values of $ap_0(t)$, $\alpha(t)$, $r_n(t,0)$ and $r_s(t,0)$. No other features are extracted and no training is performed.

The model receives as input data (observation) the sequence of probabilities in (14)-(16), $\mathbf{O} = [\mathbf{p}(1), \mathbf{p}(2), \mathbf{p}(3), \ldots, \mathbf{p}(t)]$, where t is the current frame index and $\mathbf{p}(t) = [p_{silence}(t), p_{voiced}(t), p_{unvoiced}(t)]^{\mathrm{T}}$. As output, it provides the optimal state sequence associated with the received observation, $\mathbf{Q} = [q(1), q(2), q(3), \ldots, q(t)]$, with $q(t) \in \{1, 2, 3\}$. Estimation of the optimal state sequence (optimal path) is accomplished by the Viterbi algorithm [18]. Observe that, since real-time is a requirement, the Viterbi algorithm is here applied in a causal fashion, i.e., the optimal state at each time instant is derived based only on past information. State transition probabilities p_{ij} are estimated from the ground truth annotation of the NOIZEUS database [19] by making the simple calculation $p_{ij} =$ (number of transitions from state i to state j)/(number of transitions from state i). Similarly, initial state probabilities π_i are estimated as $\pi_i =$ (number of sentences starting at state i)/(number of sentences).

2.4 Algorithm Overview and Examples

In this section, we give a summary of the processing steps performed by the proposed voicing detection algorithm. All these steps are repeated for each incoming time frame, thus enabling voicing detection in real time. The algorithm also returns a set of values which are employed by the system described in Section 3 to perform the speech enhancement operation. Note that each step, except the difference function, involves only a small number of operations.

1. Compute the total power $r_x(t,0)$ of the input noisy frame.
2. Execute the YIN algorithm to compute $cmnd_x(t,\tau)$ and estimate $F_0(t)$ and $ap_0(t)$. Alternatively, $cmnd_x(t,\tau)$ can be computed from the approximate difference function proposed in [16].
3. Compute $\hat{r}'_n(t,0)$ from the current window using (11)-(13).
4. Determine $\alpha(t)$ using (10).
5. Compute the probabilities in (14)-(16) and decide on the current voicing state using the Viterbi algorithm.
6. Return $r_x(t,0)$, $\hat{r}'_n(t,0)$, $F_0(t)$ and the voicing decision.

To illustrate the voicing detection scheme, Figure 2 shows an example where a male sentence is degraded by stationary white noise at an SNR of 0 dB. In subplot (b), the corresponding aperiodicity sequence $ap_0(t)$ is depicted together with the proposed adaptive threshold $\alpha(t)$ and the fixed threshold of the YIN algorithm (here set to 0.2). For clarity, values of $\alpha(t)$ in frames with SNR below SNR_{min} have been omitted, since $\alpha(t)$ is not reliable in such a case. In subplot (c), the true voicing state is compared with the decisions taken by both the proposed method and the YIN algorithm. As seen in the figure, the proposed method performs a more accurate voicing detection than YIN, since it is able to adapt to the instantaneous SNR. YIN clearly fails in detecting voiced frames with low SNR.

Figure 3 shows a second example where a female sentence is degraded by non-stationary street noise at an overall SNR of 0 dB. In this example, a burst of noise originating from a passing car is produced during the time-interval from 0.25 to 1.75 s. As seen, this burst of noise makes the fixed YIN threshold to misclassify most of the voiced frames during this time interval. The proposed method, on the other hand, is able to detect many of these frames, because it adapts to the current SNR by estimating the background noise power. To illustrate this noise power estimation, Figure 3d shows a comparison between the true noise power and the noise power estimated by our method in this example. Here, a sliding window

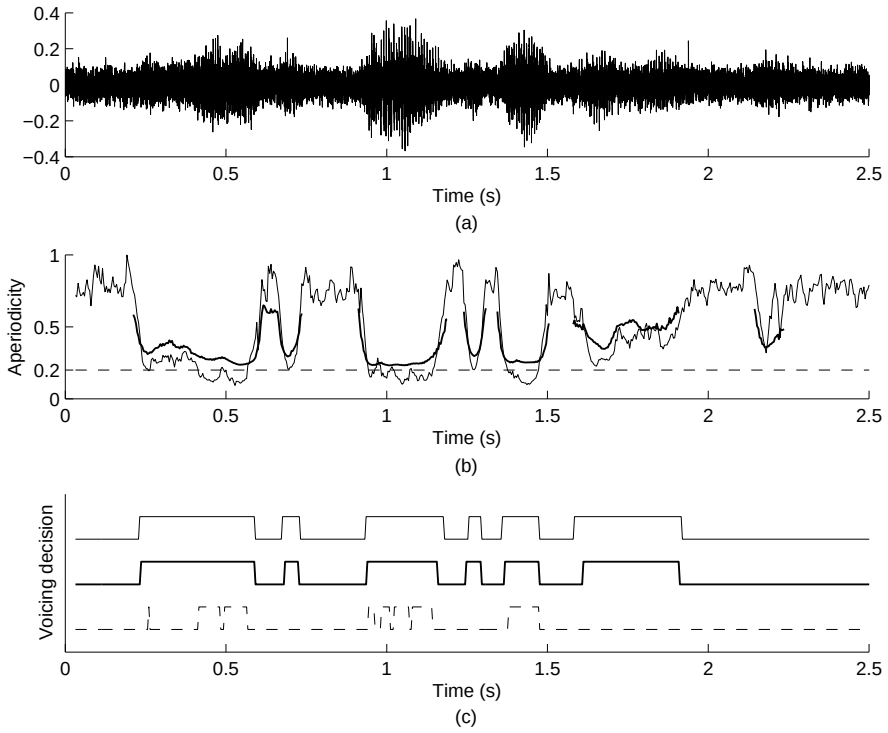

Figure 2: (a) Speech signal degraded by white noise at an overall SNR of 0 dB. (b) Aperiodicity sequence (solid line) and comparison between the proposed adaptive V/U threshold (bold line) and the fixed YIN threshold (dashed line). (c) Ground truth voicing state (solid line) and comparison between the proposed decision (bold line) and the YIN decision (dashed line).

with a length of 0.3 s has been used. As shown, the proposed estimator is able to approximately track the changing noise power, being particularly accurate for decreasing levels. For increasing noise levels, however, the proposed estimator follows the noise power with a certain delay. This behavior can be explained because the method always chooses the lowest possible estimate within the window, as seen in (13). When the noise is decreasing, this estimate is usually taken from the last frames of the window, and thus is more accurate. By contrast, when the noise is increasing, the estimate is usually taken from the beginning of the interval, introducing thus a certain inaccuracy. Despite this, the estimated noise power is good enough to achieve an accurate voicing detection.

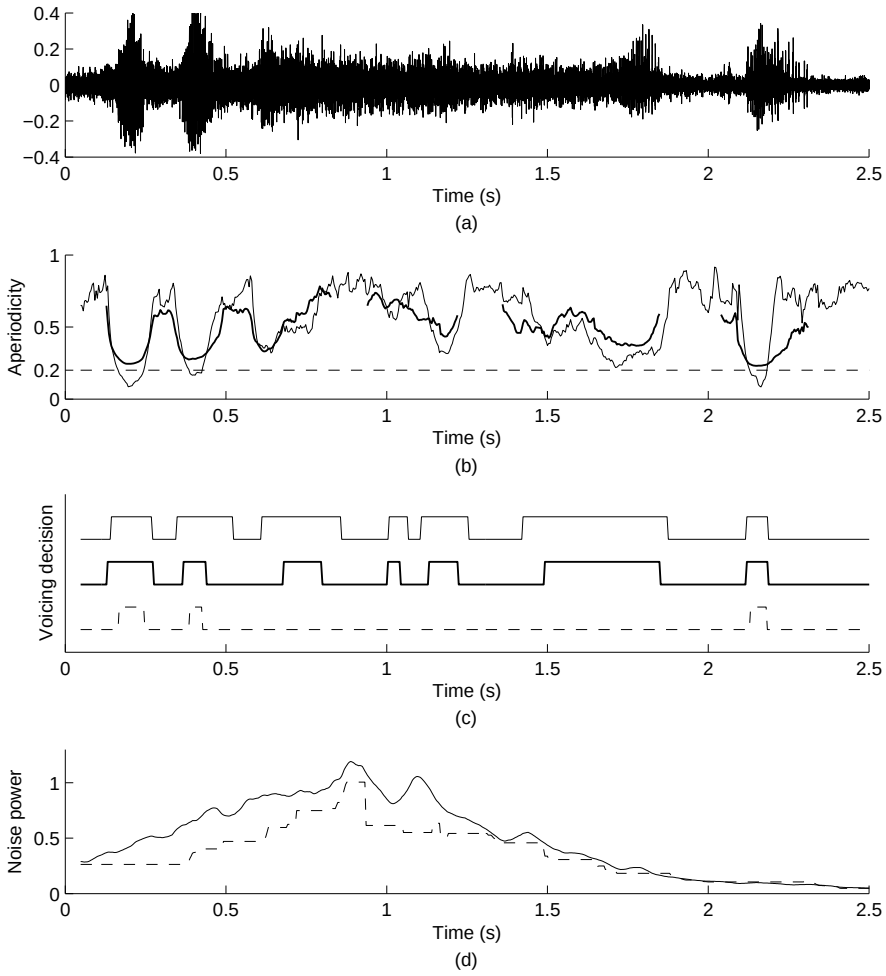

Figure 3: (a) Speech signal degraded by non-stationary street noise at an overall SNR of 0 dB. (b) Aperiodicity sequence (solid line) and comparison between the proposed adaptive V/U threshold (bold line) and the fixed YIN threshold (dashed line). (c) Ground truth voicing state (solid line) and comparison between the proposed decision (bold line) and the YIN decision (dashed line). (d) Comparison between estimated noise power (dashed line) and the true noise power (solid line).

3 Application to Speech Enhancement

Figure 4 shows the block diagram of the implemented F_0-based speech enhancement system which integrates the proposed voicing detector. As shown, the core of the system is the voicing detection algorithm, which determines the processing strategy applied to each incoming frame. The remaining blocks of the system perform the enhancement task employing simple signal processing operations.

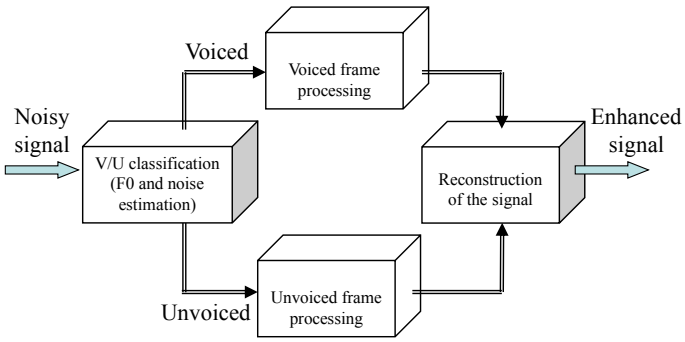

Figure 4: Block diagram of the proposed system for speech enhancement.

This simple configuration allows us to evaluate the proposed voicing classifier in terms of speech enhancement performance, and to compare it with other speech enhancement algorithms.

The system operates in the following way: first, a frame-by-frame analysis of the noisy speech signal is performed in order to classify each frame as voiced or unvoiced. The voicing classifier also provides the estimated F_0 and an estimation of the noise level. According to the voiced-unvoiced classification, the signal frame is processed using different blocks. This approach is justified by the nature of voiced frames, in which the frequency spectrum of the signal has a harmonic nature (i.e., the spectral peaks belonging to speech are located in frequencies multiple of F_0). Thus, a comb filter can be designed and applied when the F_0 is well estimated. However, when unvoiced frames are detected, the structure of the speech spectrum is not known in advance. In such a case, we employ a spectral subtraction technique. Finally, the last block of Figure 4 reconstructs the enhanced signal from the processed frames making use of an overlap-add procedure.

3.1 Voiced Signal Enhancement

When the current frame is labeled as voiced, the signal is filtered by a comb filter tuned to the estimated fundamental frequency $F_0(t)$ and its integer multiples. The use of this comb filter to enhance voiced speech involves certain implementation issues that must be taken into account to handle the inharmonicity of speech. It is known that the quasi-periodic signal generated by the vibration of the vocal chords is not purely harmonic. As a consequence, the spectral peaks are not exactly located at integer multiples of F_0, but slightly shifted from their harmonic position. Furthermore, spectral peaks belonging to higher harmonics are more shifted in frequency than those corresponding to lower harmonics. This inharmonicity can

produce high-order harmonics that may or may not be completely removed when applying a purely harmonic comb filter. To alleviate this problem, we implement a peak picking procedure for locating the harmonics and constructing the filter.

Before applying the filter, we first take into account the estimated SNR in the current frame. If this SNR exceeds a certain threshold, SNR_{max}, the frame is not processed. In this case, the signal quality is considered good enough, and the use of the comb filter, which may suffer from inharmonicity and order mis-estimation, is avoided. When the estimated SNR is below this threshold, the following steps are applied:

- We apply a Hanning window $w_H[m]$ to the signal frame t, and the discrete Fourier transform (DFT) is calculated in the form:

$$X(k,t) = \sum_{m=0}^{M-1} x[t+m] w_H[m] e^{-j(2\pi/M)km}. \tag{17}$$

- A comb filter adapted to F_0 and its harmonics is constructed. To maintain a certain spectral peak, all the frequency bins belonging to its main lobe are located and conserved, as it is usually done when applying comb filtering to speech extraction [20]. In order to minimize the effect of inharmonicity, the maximum of the magnitude spectrum at each multiple of F_0 is searched for over a range of frequency bins, as follows:

$$k^t_{max}(i) = \arg\max_k \left(X(k,t) \Pi(k - i \cdot k^t_{F_0}) \right). \tag{18}$$

  Here, $k^t_{max}(i)$ is the frequency bin corresponding to the maximum value of $X(k,t)$ around the ith multiple of F_0, $\Pi(k)$ is a rectangular function with a width equal to the main lobe width of the analysis window, and $k^t_{F_0}$ is the bin corresponding to the fundamental frequency. The comb filter is then constructed as

$$W(k,t) = \sum_{i=1}^{I} \Pi\left(k - k^t_{max}(i)\right), \tag{19}$$

  where I is the number of considered harmonics.

- Finally, the input frame is filtered by the comb filter, and the inverse DFT is applied to obtain the enhanced voiced frame, that is,

$$x_f[t+m] = \frac{1}{M} \sum_{k=0}^{M-1} X(k,t) W(k,t) e^{j(2\pi/M)mk}. \tag{20}$$

In our experiments, we implemented the algorithm with $SNR_{max} = 20$ dB and $I = 9$.

3.2 Unvoiced Signal Enhancement

In the case of signal frames labeled as unvoiced, which correspond to unvoiced phonemes or speech silence, another enhancement technique has to be applied. Unlike voiced parts, unvoiced phonemes cannot be modeled as a sum of basic harmonic components and, in consequence, they cannot be enhanced by using the aforementioned harmonic comb filter. The solution adopted in the proposed approach is to enhance these frames by using the classical spectral subtraction method. Among the multiple variations of this technique proposed in the literature, we have employed the multiband spectral subtraction algorithm described in [21], which is robust under colored interferences and produces low residual noise.

In order to achieve an appropriate enhancement, an estimate of the underlying noise spectrum has to be supplied to the spectral subtraction algorithm. Furthermore, since realistic noisy environments are typically non-stationary, this estimate must be updated as often as possible. To do so, we estimate the noise spectrum for each unvoiced frame exploiting the information provided by the noise power estimator described in Section 2.2. That is, assuming that the background noise remains stationary within the current window, we can estimate the noise spectrum either from the last almost-silence frame or from the last almost-periodic frame, in the same way as (13). Hence, depending on the chosen estimator in (13), the noise spectrum is estimated by using one of the two following procedures:

- If $P_{min}(t)$ is the current noise power estimator, we can assume that its corresponding time-frame is almost-silence, and thus, its spectrum can be considered as a good estimator of the current noise spectral pattern.

- On the other hand, if $\hat{r}_n(t,0)$ is the current noise power estimator, its corresponding time-frame can be considered highly periodic, and the underlying noise spectrum can be estimated by performing the harmonic tunneling technique [22]. Essentially, the harmonic tunneling algorithm estimates the noise spectrum by exploiting the gaps between harmonics, under the assumption that these gaps contain only noise energy. To further estimate the noise spectrum in those DFT bins occupied by the harmonics, a simple interpolation is performed.

In practice, to obtain a more robust noise pattern, the noise spectrum estimation is also performed on the four neighboring frames around the last almost-silence frame (in the first case) or the last almost-periodic frame (in the second case). The five spectral patterns are then averaged to obtain a final noise spectrum estimate.

4 Experimental Results

To assess the performance of the proposed voicing detection algorithm, we have taken the NOIZEUS database developed in [19]. This database contains 30 speech sentences pronounced by three male and three female speakers, and affected by different types of real-world noises at different SNRs. These noisy environments consist of several recordings taken from different places, including babble, car, restaurant, street, airport or train, where several non-stationary noise sources can be perceived. All sentences in the database are sampled at 8 kHz and have a length between 2.1 and 3.5 s. Evaluation over this database is conducted to assess both voicing detection accuracy and speech enhancement performance.

For the experiments, the length of the signal frames is set to 512 samples (64 ms at 8 kHz). This frame size is considered large enough to assume a perfect uncorrelation between the speech signal and the interfering noise, and hence, to consider (10) and (12) as completely valid. The hop size between frames is set to 32 samples (4 ms). To estimate the noise power, a sliding window with a length of 0.5 s is employed, which is a good trade-off between a moderate noise tracking delay and an accurate noise power estimation. In the blocks devoted to enhance voiced and unvoiced frames, windowing with a Hanning window is applied, and the order of the DFT is set to 8192 frequency bins. This DFT size provides a high enough resolution for the proper enhancement of low frequencies, and was chosen empirically as a trade-off between achieved quality and complexity. The proposed voicing detector is also evaluated when it is implemented with the approximate difference function proposed in [16]. This implementation reduces considerably the computational cost (the approximate difference function involves less than 3 million instructions per second), and enables the integration of the algorithm in hearing aids. A deeper explanation can be found in [16].

4.1 Voicing Detection Accuracy

In order to assess the robustness of the proposed voicing detection algorithm, we have evaluated its performance over the database by computing three values: the accuracy rate (ACC), the hit rate (HR) and the false-alarm rate (FA). These measures are given by

$$\text{ACC} = \frac{N_V^{\text{corr}} + N_U^{\text{corr}}}{N_V + N_U} \times 100 \tag{21}$$

$$\text{HR} = \frac{N_V^{\text{corr}}}{N_V} \times 100 \tag{22}$$

$$\text{FA} = \frac{N_U^{\text{err}}}{N_U} \times 100, \tag{23}$$

where N_V and N_U are, respectively, the total number of voiced and unvoiced frames, N_V^{corr} and N_U^{corr} are the number of voiced and unvoiced frames correctly classified, and N_U^{err} is the number of unvoiced frames erroneously detected as voiced. Essentially, ACC represents the percentage of correctly classified frames, HR is the percentage of voiced frames correctly detected, and FA is the percentage of unvoiced frames erroneously classified as voiced. For computing these measures, the true voicing state of each sentence was determined by labeling manually the original excerpts in absence of noise.

For comparison purposes, we have used two systems as a reference: the voicing detection performed by the YIN algorithm and the state-of-the-art voicing classifier included in the ETSI ES 202 211 standard. The YIN algorithm was configured with the same settings (frame size and hop size) as the proposed method, and the aperiodicity threshold for voicing classification was set to 0.2. This threshold offers a good voicing detection accuracy on clean speech, and it has been proven to perform well with related applications [23]. On the other hand, the ETSI front-end algorithm was executed employing the available implementation provided by ETSI. Since the ETSI front-end distinguishes four types of sounds ("non-speech", "unvoiced", "mixed voiced" and "fully voiced"), these classes were grouped in order to compare with our method. Specifically, "non-speech" and "unvoiced" frames were simply tagged as unvoiced, while "mixed voiced" and "fully voiced" frames were considered as voiced.

The obtained results for different real-world noises included in the NOIZEUS database are shown in Table 1. Here, we have also considered white noise in order to assess the proposed classifier with stationary noise. In addition, the results obtained by our method when implemented with the approximate difference function in [16] are also included under the tag *Proposed (efficient)*. As seen in the table, in comparison with YIN, the proposed method yields a clearly higher ACC and HR in each of the seven types of noise for all SNRs, although the FA is notably increased. For low SNRs (0 and 5 dB), the proposed system becomes clearly advantageous over YIN, since the gain obtained in ACC and HR compensates for the increase of the FA. For high SNRs, however, the differences in the ACC are less significant, and the performance achieved by the method is even slightly lower than that obtained by YIN in the case of clean speech. Compared to the ETSI classifier, the proposed system is more robust against white noise (HR is 70.9% for SNR=0 dB) and real-world noises with similar properties (car, train or street) than against the remaining kinds of noise. As shown, the proposed detection scheme clearly outperforms the ETSI algorithm for white, car, train and street noises, especially at low SNR levels. This behavior is not surprising since these types of noise fulfill the approximations made in the proposed voicing decision algorithm in a better manner (i.e., relatively flat and broadband spectrum, and absence

Table 1: ACC, HR and FA rates of the proposed voiced-unvoiced classifier with different noise types and SNRs. Comparison with YIN and ETSI 202 211 (%).

Noise type	SNR (dB)	Proposed	Proposed (efficient)	YIN	ETSI 202 211
		ACC (HR, FA)	ACC (HR, FA)	ACC (HR, FA)	ACC (HR, FA)
Airport	0	66.53 (38.08, 10.04)	64.31 (34.79, 11.38)	59.08 (14.76, 4.41)	68.03 (37.76, 7.04)
	5	76.04 (56.64, 07.98)	74.14 (54.81, 09.93)	69.31 (35.33, 2.70)	77.63 (59.84, 7.71)
	10	83.42 (75.79, 10.30)	81.69 (74.45, 12.35)	80.58 (61.96, 3.86)	85.50 (77.43, 7.86)
	15	90.33 (89.68, 09.14)	88.33 (87.95, 11.35)	90.14 (81.22, 2.51)	89.03 (85.39, 7.98)
Babble	0	66.06 (35.80, 8.84)	65.12 (34.85, 09.95)	59.12 (09.58, 0.07)	66.35 (29.03, 2.92)
	5	73.78 (51.86, 8.17)	72.37 (50.65, 09.74)	69.58 (33.33, 0.56)	77.09 (54.38, 4.20)
	10	85.45 (75.84, 6.64)	82.57 (73.06, 09.60)	82.53 (61.52, 0.17)	87.33 (75.66, 3.06)
	15	91.72 (90.30, 7.11)	89.40 (88.75, 10.06)	91.06 (80.89, 0.56)	90.28 (84.83, 5.23)
Car	0	74.60 (50.92, 5.90)	72.11 (48.20, 8.20)	58.16 (07.44, 0.06)	64.86 (22.82, 0.52)
	5	86.26 (73.57, 4.94)	82.83 (71.50, 7.83)	69.50 (32.46, 0.00)	76.41 (49.65, 1.54)
	10	90.44 (84.61, 4.76)	88.49 (83.13, 7.10)	82.44 (61.11, 0.00)	85.92 (71.55, 2.25)
	15	93.95 (91.71, 4.20)	92.18 (90.95, 6.81)	91.43 (81.24, 0.18)	90.83 (83.27, 2.95)
Restaurant	0	65.04 (37.50, 12.27)	62.99 (33.00, 12.31)	60.33 (15.63, 2.85)	68.68 (40.41, 8.03)
	5	74.13 (56.24, 11.13)	73.89 (56.40, 11.70)	71.57 (38.92, 1.53)	79.30 (63.42, 7.92)
	10	83.24 (77.74, 12.23)	82.52 (77.99, 13.75)	83.08 (65.13, 2.13)	85.31 (78.89, 9.41)
	15	88.04 (87.84, 11.79)	87.21 (88.01, 13.45)	91.33 (81.73, 0.77)	89.50 (86.87, 8.33)
Street	0	74.46 (51.65, 6.76)	71.52 (49.06, 9.98)	63.07 (18.37, 0.12)	69.10 (33.63, 1.69)
	5	83.57 (71.24, 6.27)	80.50 (68.21, 9.38)	74.21 (43.04, 0.11)	79.14 (55.71, 1.57)
	10	89.60 (84.41, 6.12)	87.56 (83.20, 8.85)	86.69 (71.53, 0.83)	86.85 (75.13, 3.50
	15	93.17 (92.59, 6.36)	91.93 (90.92, 7.23)	93.37 (85.52, 0.16)	91.50 (85.01, 3.15)
Train	0	73.46 (46.19, 4.07)	70.15 (44.36, 8.60)	63.98 (20.24, 0.00)	67.47 (28.92, 0.78)
	5	82.81 (67.41, 4.51)	79.30 (64.08, 8.17)	76.97 (49.01, 0.00)	80.22 (58.71, 2.07)
	10	90.13 (83.50, 4.41)	87.31 (80.49, 7.07)	87.69 (72.88, 0.12)	88.16 (76.90, 2.57)
	15	94.64 (91.86, 2.52)	92.02 (90.14, 6.44)	94.07 (87.22, 0.29)	92.14 (86.71, 3.39)
White	0	85.60 (70.95, 2.33)	82.37 (68.20, 5.95)	66.31 (25.47, 0.06)	65.36 (24.15, 0.70)
	5	91.54 (84.14, 2.36)	89.52 (82.92, 5.04)	79.09 (53.73, 0.02)	78.01 (53.32, 1.66)
	10	95.29 (92.70, 2.58)	93.62 (92.11, 5.13)	89.91 (77.71, 0.05)	86.91 (73.61, 2.13)
	15	96.70 (96.10, 2.80)	94.62 (94.04, 4.91)	94.83 (88.75, 0.17)	91.72 (84.95, 2.71)
average (noise)		83.57 (71.67, 6.66)	81.44 (69.86, 9.01)	77.83 (52.00, 0.86)	80.66 (62.07, 4.03)
clean		96.66 (96.11, 2.89)	94.93 (94.89, 5.03)	97.27 (96.05, 1.72)	95.21 (92.95, 2.93)

of periodic components). On the contrary, under airport, babble and restaurant noises, our algorithm obtains worse results than the ETSI method, just because these noises do not fulfill the considered assumptions in some cases. Specifically, these noisy environments contain sound sources which are actually pitched, and hence are erroneously detected as voiced by the method, leading to a relatively high FA rate. Furthermore, these noise sources usually have a spectrum that is concentrated around the main components of speech, which makes it more difficult to precisely detect voiced frames and results in a lower HR. Generally, in comparison with the ETSI classifier, our algorithm tends to increase the HR for all types of noise, at the price of increasing also the FA. As can be seen in Table 1, the

efficient implementation of our method using the approximate difference function produces only a moderate degradation in the results, affecting particularly the FA. The degradation is more noticeable for car, street, train and white noises at low SNRs. Anyway, *Proposed (efficient)* still performs better than ETSI for car, street and white noises.

4.2 Speech Enhancement Evaluation

In order to evaluate the implemented speech enhancement system, we have used two objective measures: PESQ and the log-likelihood ratio (LLR). PESQ is a standard measure recommended by ITU for speech quality assessment of telephony and narrowband speech coders [24], showing high correlations with subjective listening tests [25]. The PESQ values are between 1.0 and 4.5, with higher values indicating higher subjective quality. LLR, on the other hand, is a classic spectral distance measure based on speech production principles, which evaluates the dissimilarity between all-pole models of the clean and enhanced speech signals [26]. The LLR measure is defined as

$$d_{\mathrm{LLR}}(\mathbf{a}_e, \mathbf{a}_c) = \log \frac{\mathbf{a}_e \mathbf{R}_c \mathbf{a}_e^{\mathrm{T}}}{\mathbf{a}_c \mathbf{R}_c \mathbf{a}_c^{\mathrm{T}}}, \tag{24}$$

where $\mathbf{a}_c$ are the linear predictive coding (LPC) coefficients of the clean speech frame, $\mathbf{a}_e$ are the coefficients of the enhanced frame and $\mathbf{R}_c$ is the autocorrelation matrix of the clean frame. LLR values are limited in the range between 0 and 2 (as in [25]), where high values indicate poor performance.

To compare the proposed system with other approaches in terms of PESQ, we have taken the PESQ results published in [27] for several state-of-the-art speech enhancement methods. These results were obtained over the NOIZEUS database considering four types of noise: babble, car, street and white, with SNRs of 0, 5 and 10 dB. The following algorithms were included in the comparison: the classical spectral subtraction algorithm (SS), the statistical minimum mean-square error (MMSE) algorithm [5] and a more recent variation of the SS algorithm based on a geometric approach (GA), originally proposed in [27]. In addition, we processed the database with the Wiener filtering scheme implemented in the ETSI advanced front-end, which is part of the ETSI ES 202 050 standard [28]. As with the proposed system, the ETSI advanced front-end is also designed to work in non-stationary noise. This method estimates a noise model online using a VAD and taking assumptions about the stationarity of the noise, and employs the estimated model to implement the Wiener filtering.

In Table 2, the PESQ results obtained with the proposed system, for both the conventional and efficient implementation of the voicing detector, are presented in

Table 2: PESQ values obtained by the proposed speech enhancement system and comparison against the geometric approach algorithm (GA), spectral subtraction (SS), MMSE algorithm and ETSI Wiener filtering. Proposed system with ideal voicing and F_0 detection is also evaluated.

Noise type	SNR (dB)	Proposed	Proposed (efficient)	Proposed (ideal)	GA	SS	MMSE	ETSI Wiener
babble	0	1.80	1.80	2.07	1.81	1.73	1.76	1.80
	5	2.17	2.12	2.33	2.16	2.04	2.12	2.15
	10	2.51	2.43	2.60	2.50	2.37	2.51	2.52
car	0	1.84	1.81	2.01	1.84	1.69	1.93	1.83
	5	2.17	2.12	2.27	2.19	1.98	2.28	2.17
	10	2.48	2.45	2.54	2.51	2.31	2.66	2.54
street	0	1.89	1.81	1.97	1.76	1.70	1.80	1.82
	5	2.25	2.15	2.28	2.16	2.00	2.20	2.18
	10	2.48	2.45	2.56	2.50	2.36	2.58	2.53
white	0	1.83	1.79	1.86	1.81	1.66	2.00	1.92
	5	2.12	2.10	2.16	2.20	1.95	2.39	2.29
	10	2.41	2.41	2.45	2.53	2.29	2.74	2.67
average		2.16	2.12	2.26	2.16	2.01	2.25	2.20

comparison with the remaining approaches. Also, results of the proposed system assuming a perfect voicing and F_0 estimation are included as a reference under the tag *Proposed (ideal)*. These values represent the upper bound of the implemented enhancing scheme and indicate the maximum performance the voicing detector can reach in combination with the implemented enhancement blocks.

As shown, the proposed algorithm performs best in situations where the noise is not very stationary (babble and street) and the SNR is 0 dB. By contrast, when the noise is rather stationary (white, car), MMSE obtains better results than the proposed approach. This result can be justified knowing that, under rather stationary conditions, an accurate description of the noise can be provided to the MMSE estimator and, in such cases, its estimation of the speech coefficients is more effective than the proposed harmonic filtering, even with perfect voicing detection. In contrast, when the noise is non-stationary (its properties change over time), the employed harmonic enhancement is usually favored over the MMSE estimation of speech components for low SNRs (0 and 5 dB). As seen in the table, the improvement achieved by *Proposed (ideal)* at 0 dB with respect to the remaining methods is higher for babble and street noises. In particular, for street noise, where the proposed classifier obtained relatively good accuracy, our method yields the best results at lower SNRs, outperforming also the ETSI advanced front-end. In the presence of babble noise, where the classifier obtained worse results, our method is competitive with respect to GA, which is a method particularly robust against correlated noises. In babble and car noises, the proposed and the ETSI method

yield similar performances at low SNRs, and for white, the proposed scheme is outperformed by ETSI. The *Proposed (efficient)* version of the method involves a slightly worse performance of the system for all conditions. As before, the system is more affected by car, street and white noises, whereas for babble at 0 dB the system obtains the same results. Generally, *Proposed (efficient)* remains competitive for real-world noises at 0 dB, and only for white noise the degradation is more significant.

Here, we have only analyzed the influence of the voicing detector over the enhancement process. In order to illustrate the effect of F_0 estimation in the system, we measure the Gross Pitch Error (GPE) of the YIN estimation when applied in combination with our voicing detector. The GPE is defined as the percentage of correctly classified voiced frames which have an incorrect pitch (a pitch value is considered incorrect if the distance from the true pitch is greater than 20%). The GPE for each situation of Table 2 is summarized in Table 3, where the efficient results are included in brackets. As seen, the most problematic noise is again babble interference, with a GPE of 32% at 0 dB. For the remaining situations, the GPE is maintained within reasonable levels, demonstrating that the YIN estimator is adequate for the proposed method. When Table 3 is analyzed in conjunction with Table 1, it can be seen that the proposed voicing estimator is able to recover many frames with correct pitch that were considered unvoiced by YIN (the percentage of voiced frames that are correctly detected, in both F_0 and voicing, is clearly higher than the HR obtained by YIN itself). Our efficient implementation of the difference function involves also a certain degradation in pitch accuracy, but it is not excessive.

To further evaluate the performance of the method, we have also considered the well-known LLR measure, which is based on more objective criteria (and not in subjective principles such as PESQ). As before, the LLR results for the MMSE, GA and SS algorithms were taken from [27], whereas the results for the Wiener filter were measured from the output signals obtained with the ETSI code. All values are shown in Table 4. As can be seen, our algorithm provides similar results for all real-world noises (babble, car and street) and, in comparison with the ETSI approach, the method performs better for babble and street, and performs worse (or similar) for car and white. *Proposed (efficient)* remains also competitive in all conditions in terms of this measure. Generally, compared with the remaining approaches, *Proposed* and ETSI obtained the best average results, presumably because the chosen criterion, which penalizes differences in the LPC-modeled spectrum, is favorable to both the harmonic and Wiener strategies.

Table 3: GPE (%) of the YIN F_0 estimator in conjunction with the proposed voicing detector for different noise types and SNRs. Results when using the approximate difference function are included in brackets.

Noise type	SNR = 0 dB	SNR = 5 dB	SNR = 10 dB
babble	32.18 (33.04)	18.85 (19.72)	8.06 (11.03)
car	22.03 (25.88)	14.00 (16.63)	5.28 (8.63)
street	18.60 (23.21)	9.22 (14.70)	3.53 (7.53)
white	9.03 (14.56)	4.37 (8.68)	2.51 (4.59)

Table 4: LLR values obtained by the proposed speech enhancement system and comparison against the geometric approach algorithm (GA), spectral subtraction (SS), the MMSE algorithm and the ETSI Wiener filter. Proposed system with ideal voicing and F_0 detection is also evaluated.

Noise type	SNR (dB)	Proposed	Proposed (efficient)	Proposed (ideal)	GA	SS	MMSE	ETSI Wiener
babble	0	0.91	0.93	0.83	1.06	0.94	1.15	0.94
	5	0.72	0.73	0.68	0.86	0.75	0.90	0.75
	10	0.55	0.55	0.52	0.69	0.55	0.67	0.57
car	0	0.91	0.92	0.86	0.98	1.00	1.01	0.90
	5	0.71	0.72	0.69	0.80	0.78	0.79	0.69
	10	0.55	0.55	0.53	0.66	0.59	0.63	0.54
street	0	0.91	0.92	0.86	1.04	1.01	1.12	0.93
	5	0.71	0.73	0.69	0.84	0.81	0.88	0.75
	10	0.57	0.57	0.55	0.70	0.62	0.68	0.58
white	0	1.31	1.35	1.30	1.55	1.74	1.54	1.31
	5	1.14	1.21	1.13	1.28	1.47	1.25	1.10
	10	0.95	1.05	0.95	1.09	1.22	1.05	0.91
average		0.83	0.85	0.80	0.96	0.96	0.97	0.83

5 Summary and Conclusions

In this work, we have presented an algorithm for voicing detection intended to work in acoustic environments where the noise is non-stationary. The algorithm computes a signal-adaptive threshold that is compared to the aperiodicity value provided by the difference function, which is a well-known time domain measure of voicing. In clean speech, a fixed threshold is enough to achieve an accurate voicing detection. However, under non-stationary noise, this threshold must be made adaptive and dependent on the current SNR. We have derived an equation to compute this signal-adaptive threshold by assuming that the interfering noise is additive and uncorrelated, and proposed a simple algorithm to estimate the background noise power by assuming local stationarity. Provided an efficient

approximation of the difference function, the method is also good enough to be implemented in hearing aids, introducing only a moderate degradation in the system performance. Experimental results over the NOIZEUS database revealed that the proposed voicing detector is robust enough whenever the assumptions made for the noise hold. The fixed YIN threshold was outperformed in all cases, and the method obtained better results than the state-of-the-art ETSI ES 202 211 classifier under white-like background noises (such as car, street or train). A simple F_0-based speech enhancement scheme integrating the proposed classifier was implemented to demonstrate the applicability of the method for denoising. The implemented speech enhancement scheme obtained similar quality results, in terms of objective measures, when compared with several well-known approaches for speech enhancement.

Currently, we are working on extending the method to perform speech enhancement on complete utterances, without the need of providing a frame-by-frame output. In such case, one can exploit temporal continuity to refine initial frame-basis decisions. For the same purpose, we also intend to apply the method in combination with sound source separation techniques based on noise and speech source modeling.

References

[1] S. Ahmadi and A. Spanias, "Cepstrum-based pitch detection using a new statistical V/UV classification algorithm", *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 333–338, 1999.

[2] A. Nehorai and B. Porat, "Adaptive comb filtering for harmonic signal enhancement", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 5, pp. 1124–1138, 1986.

[3] E. George and M. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model", *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 5, pp. 389–406, 1997.

[4] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[6] Y. Ephraim and H. Van Trees, “A signal subspace approach for speech enhancement”, *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 251–266, 1995.

[7] Y. Qi and B. Hunt, “Voiced-unvoiced-silence classifications of speech using hybrid features and a network classifier”, *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 2, pp. 250–255, 1993.

[8] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, “Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds”, *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[9] P. Boersma, “Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound”, in *Proc. of Institute of Phonetic Sciences*, Amsterdam, The Netherlands, 1993, vol. 17, pp. 97–110.

[10] A. de Cheveigné and H. Kawahara, “Yin, a fundamental frequency estimator for speech and music”, *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[11] H. Kobatake, “Optimization of voiced/unvoiced decisions in nonstationary noise environments”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 1, pp. 9–18, 1987.

[12] ETSI ES 202 211 V1.1.1, “Speech processing, Transmission and quality aspects (STQ); Distributed speech recognition; Extended front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm”, 2003.

[13] T. Nakatani, S. Amano, T. Irino, K. Ishizuka, and T. Kondo, “A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments”, *Speech Communication*, vol. 50, no. 3, pp. 203–214, 2008.

[14] F. Beritelli, S. Casale, S. Russo, and S. Serrano, “Adaptive V/UV speech detection based on characterization of background noise”, *EURASIP Journal on Audio, Speech, and Music Processing*, 2009.

[15] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama, “Single and multiple F0 contour estimation through parametric spectrogram modeling of speech in noisy environments”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1135–1145, 2007.

[16] P. Cabañas-Molero, N. Ruiz-Reyes, P. Vera-Candeas, and S. Maldonado-Bascon, “Low-complexity F0-based speech/nonspeech discrimination ap-

proach for digital hearing aids", *Multimedia Tools and Applications*, vol. 54, no. 2, pp. 291–319, 2011.

[17] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics", *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[18] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", in *Proc. of the IEEE*, Feb 1989, vol. 77, no. 2, pp. 257–286.

[19] Y. Hu and P. Loizou, "Subjective comparison of speech enhancement algorithms", in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 2006, pp. 153–156.

[20] E. Cho, J. O. Smith, and B. Widrow, "Exploiting the harmonic structure for speech enhancement", in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, March 2012, pp 4569–4572.

[21] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise", in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, FL, USA, May 2002, pp. 4160–4164.

[22] D. Ealey, H. Kelleher, and D. Pearce, "Harmonic tunneling: tracking non-stationary noises during speech", in *Proc. of the 7th European Conference on Speech Communication and Technology*, Aalborg, Denmark, Sep 2001, pp. 437–440.

[23] N. Ruiz-Reyes, P. Vera-Candeas, J. Muñoz, S. García-Galán, and F. Cañadas, "New speech/music discrimination approach based on fundamental frequency estimation", *Multimedia Tools and Applications*, vol. 41, no. 2, pp. 253–286, 2009.

[24] ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ), and objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs", 2000.

[25] Y. Hu and P. Loizou, "Evaluation of objective quality measures for speech enhancement", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, 2008.

[26] P. Loizou, "Speech quality assessment", *Multimedia Analysis, Processing and Communications*, L. Weisi et al. Eds., pp. 623–654, Springer Verlag, 2011.

[27] Y. Lu and P. Loizou, “A geometric approach to spectral subtraction”, *Speech Communication*, vol. 50, no. 6, pp. 453–466, 2008.

[28] ETSI ES 202 050 V1.1.5, “Speech processing, Transmission and quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms”, 2007.

Paper C

Compositional Model for Speech Denoising based on Source/Filter Speech Representation and Smoothness/Sparseness Noise Constraints

P. Cabañas-Molero, D. Martínez-Muñoz, P. Vera-Candeas, F. J. Cañadas-Quesada, and N. Ruiz-Reyes, "Compositional model for speech denoising based on source/filter speech representation and smoothness/sparseness noise constraints", accepted for publication in *Speech Communication*, Special Issue on Advances in Sparse Modeling and Low-rank Modeling for Speech Processing, 2015.

Abstract

We present a speech denoising algorithm based on a regularized non-negative matrix factorization (NMF), in which several constraints are defined to describe the background noise in a generic way. The observed spectrogram is decomposed into four signal contributions: the voiced speech source and three generic types of noise. The speech signal is represented by a source/filter model which captures only voiced speech, and where the filter bases are trained on a database of individual phonemes, resulting in a small dictionary of phoneme envelopes. The three remaining terms represent the background noise as a sum of three different types of noise (smooth noise, impulsive noise and pitched noise), where each type of noise is characterized individually by imposing specific spectro-temporal constraints, based on sparseness and smoothness restrictions. The method was evaluated on the 3rd CHiME Speech Separation and Recognition Challenge development dataset and compared with conventional semi-supervised NMF with sparse activations. Our experiments show that, with a similar number of bases, source/filter modeling of speech in conjunction with the proposed noise constraints produces better separation results than sparse training of speech bases, even though the system is only designed for voiced speech and the results may still not be practical for many applications.

1 Introduction

Speech separation from background noise and other acoustical interferences (a problem often referred to as *speech enhancement*) is one of the most popular lines of research in signal processing. Applications include hands-free communications systems, automatic speech recognition, hearing aids and, in general, every situation where a contaminated speech signal must be restored to its original form. The problem is specially difficult for one-channel mixtures, where spatial information is unavailable as a cue for separating sound sources. Traditionally, speech enhancement has been accomplished by using filter-based algorithms, in which the clean speech spectrum is retrieved based on the estimation of the power spectral density (PSD) of the undesired sound [2, 10, 36, 7]. More recently, algorithms based on computational auditory scene analysis (CASA) [3] have been proposed to separate speech without requiring prior knowledge about the interfering sources [16].

A solution that has gained considerable attention in the last years is the use of model-driven methods, in which speech and noise components are modeled through parametric descriptions that characterize the behavior of each component

[19, 32]. The separation process consists then in estimating the parameters of these models, usually by resolving a minimization problem (an example can be found in [19]). Among all the model-driven methods, probably the most popular are those based on *compositional* models, specially due to their easy formulation and fast computation [32]. In compositional models, the spectrogram of each source signal is modeled by a combination of spectral bases, which represent spectral patterns (which may be unknown) from which that source can be constructed. The observed mixed signal can then be expressed as a constructive combination of the different basis spectra corresponding to the underlying sources, and the separation is accomplished by decomposing the input spectrogram into these bases and their corresponding gains in each time instant. The success of this model relies on the fact that many common sounds can be approximated as a time-varying combination of repetitive fixed patterns. For this reason, compositional models have been widely applied to music signals, which are typically constructed from repetitive structures (notes, chords) that combine along time with different degrees of intensity. Another reason for the success of these models is the existence of mathematical tools that enable to estimate their parameters with fast converging iterative algorithms, most of them derived from the field of non-negative matrix factorization (NMF) [21]. During the last years, powerful NMF-based algorithms for music analysis or separation have been developed, based either on formulating appropriate signal models [33, 6] or imposing constraints to the decomposition method [30, 5].

Recently, some efforts have been made to extend the applicability of NMF-based methods for the analysis and separation of speech signals. Since speech is not as intrinsically repetitive as music, mainly due to the high number of possible pronunciations and intonations, the majority of the methods in the literature need to use large dictionaries of speech and noise patterns, which may be composed by thousands of bases without any particular high level meaning. These dictionaries are usually learned from training material imposing sparsity on the activations, such that at test time, the mixture is factorized keeping the bases fixed and optimizing the activations, also enforcing sparsity or any other appropriate constraint. For instance, in [37] a regularized NMF is proposed for speech denoising, where the activations are imposed to preserve the same statistics found during training. In [26], a sparse NMF decomposition is used to separate concurrent speakers from a given mixture, based on speaker-dependent dictionaries which are also learned enforcing sparsity. In [35], a discriminative training approach with separate bases for analysis and reconstruction is proposed, where the reconstruction bases (trained with material including mixed sources) are optimized to recover the sources with Wiener filtering. Recently, a strategy that has become popular for acquiring basis functions is to use exemplar-based approaches, in which the bases are randomly

selected as a subset of the training data, without performing any training. This approach is reported to produce good results for speech separation and recognition [14, 13], specially when the exemplars cover several time frames. Other methods try to exploit the structure of speech to construct speech bases with a certain high level meaning. For example, a method is proposed in [25] that employs separate bases for each phoneme, learned from a corpus of individual phonemes. Although the bases trained in this way provide a good separation, the system requires prior knowledge about the location of the phonemes in the recording. In [17], speech is modeled using trained spectro-temporal template atoms, such that an atom is trained for each state label of a recognizer. The method described in [31] relies on a considerably different approach. Instead of learning basis vectors for each source, the method trains parameters of prior distributions defined for these basic vectors, following a Bayesian perspective. During separation, the basis vectors can be updated to better approximate the input signal, as long as they fit the learned distributions. Most of the algorithms for speech and noise separation are *supervised*, meaning that both speech and noise bases have to be trained. Recently, there have been an interest to develop robust *semi-supervised* algorithms, where the noise model can be learned online. One example is the work by [22], where the priors for noise bases are updated from the data to separate speech and noise with a Bayesian NMF.

The first semi-supervised method in the literature designed to decompose vocal sounds into bases with an explicit higher level meaning is described in [9]. In [9], a source/filter signal model is proposed to represent the source of interest, such that, at each frame, the source is assumed to have an excitation part, approximated by combining a dictionary of excitation bases, and a filter part, approximated by combining a dictionary of filter bases. This representation is assumed discriminative enough to allow the separation of the target source from the remaining content, without requiring any training or further constraints. Although the model proposed in [9] is generic and potentially applicable to a wide range of music applications, it is interesting to explore which modifications would be useful for speech and noise separation. Specifically, three important aspects can be observed. First, in speech utterances the speaker produces a higher number of pitches than in music, due to the natural intonation present in common speech. Second, since the number of phonemes is limited, it is possible to define a specific set of spectral filters for each phoneme. Instead of using generic smooth functions as in [9], these filters can be learned from actual phonemes in a previous training stage. Since the filter and source contributions are decoupled in the model, it is possible to characterize each phoneme with a small number of filters. And third, background noise can be characterized imposing certain mathematical restrictions to its bases and gains. For instance, it is known that most real noises exhibit a rel-

atively smooth spectrogram in comparison with the target speech. In this case, if the noise matrix is constrained to be smooth, it will capture the background sound more effectively, thus avoiding the inclusion of speech components. Following this strategy, different types of noises (or even other interferences, such as music) can be jointly captured provided a mathematical restriction describing their behavior, as long as these restrictions are distinguishable from speech. The idea of incorporating constraints to the parameters in addition to source/filter modeling is not new in the context of NMF, and has been applied before for semi-supervised speech/noise separation. In [27], a similar probabilistic non-negative source/filter model is proposed for separating speech from noise, in which the constraints are focused on characterizing the dynamics of speech. The generic framework by [24] also enables implicitly to define sources under a source/filter representation, and to incorporate constraints to the parameters of the model.

In this paper, we propose an extension of the signal model described in [9] focused on defining constraints for the background noise, and applied for the problem of noise and voiced speech separation. The spectrogram of the input signal is modeled as a sum of 4 matrices, each representing a different contribution to the mixture. The first component represents the voiced speech source by means of a source/filter model (similar to [9]), in which the speech spectrogram is factorized into a combination of fixed spectral excitation functions modulated by a combination of fixed phoneme filters. As in [9], each excitation basis corresponds to the glottal spectrum for a particular pitch, such that all the bases cover a discretized range of pitch candidates. The filter bases are trained on the TIMIT database [12] for each individual phoneme (using the same source/filter model), such that a few basis filters are learned from each phoneme. The remaining three matrices are used to represent different properties of the background noise. For each noise matrix, a specific spectro-temporal behavior is imposed to its basis and amplitude vectors during the decomposition process, allowing to capture a particular type of noise. Three types of noises are considered: pseudo-stationary broadband noises (which are characterized by smoothness in both time and frequency directions), impulsive noises (characterized by smoothness in frequency and sparseness in time) and "pitched" interferences (characterized by sparseness in frequency and smoothness in time). The parameters of the model (noise bases and gains, and amplitude gains for speech excitations/filters) are estimated by minimizing a global objective function, which depends on the reconstruction error and the class-specific cost measures imposed to the noise bases and gains. The estimation algorithm is based on the multiplicative update rules employed in NMF. The proposed approach has two important advantages. First, it does not require to train the background noise beforehand, because it is learned from the observed signal by imposing generic noise properties. And second, the source/filter model

used for the speech source allows for a representation of speech with a relatively small number of bases, contrary to other representations (such as sparse decompositions) in which a large dictionary is required. We also demonstrate experimentally that, with a similar number of bases, our method outperforms conventional sparse NMF with trained speech bases.

The remaining of this paper is organized as follows. In Section 2, the basic principles behind source separation with compositional models and NMF decomposition are briefly described. The source/filter model inherited from [9] to model speech is described as a starting point of our proposal. In Section 3 the proposed signal model is described in detail along with its corresponding regularized NMF decomposition. The mathematical restrictions proposed to model the background noise are formulated. In Section 4, details are given about the training stage employed to learn the phoneme filters. Finally, in Section 5, the algorithm is evaluated on the set of noisy speech signals proposed in the 3rd CHiME development set, providing separation results and comparison with state-of-the-art approaches. The last section states the main conclusions and lines for future work.

2 Basic Principles of Source Separation with NMF

2.1 NMF for Source Separation

Separation algorithms employing NMF are based on compositional signal models in which the spectrogram of each source is modeled as a time-varying combination of source-specific basis functions, called *components* or bases, and where all elements of the model are non-negative. Let $\mathbf{s}_t$ be the spectral vector of the speech source in frame t. In its simplest form, the model of $\mathbf{s}_t$ is a linear combination of basis functions $\mathbf{w}_j^s$ as

$$\mathbf{s}_t = \sum_{j=1}^{J} h_{jt}^s \mathbf{w}_j^s, \tag{1}$$

where J is the number of speech bases, and h_{jt}^s is the weight or *gain* of the j^{th} basis function in frame t. Similarly, each noise spectral vector $\mathbf{b}_t$ can be expressed as a linear combination of K noise basis functions $\mathbf{w}_k^b$ with gains h_{kt}^b. According to this model, the input spectrogram $\mathbf{X}$ can be represented as a combination of speech and noise bases, which can be written using matrix notation as

$$\mathbf{X} = \mathbf{W}^s\mathbf{H}^s + \mathbf{W}^b\mathbf{H}^b = \mathbf{W}\mathbf{H}, \tag{2}$$

where $\mathbf{W}^s = [\mathbf{w}_1^s, \ldots, \mathbf{w}_J^s]$, $\mathbf{W}^b = [\mathbf{w}_1^b, \ldots, \mathbf{w}_K^b]$, $[\mathbf{H}^s]_{jt} = h_{jt}^s$ and $[\mathbf{H}^b]_{kt} = h_{kt}^b$. Observe that the matrices $\mathbf{W}^s$ and $\mathbf{W}^b$ can be viewed as source-specific dictionaries, containing the building patterns from which each source can be constructed in

any frame. The gain matrices $\mathbf{H}^s$ and $\mathbf{H}^b$ determine how these patterns must combine in each frame to compose the actual realization of the sources in the observed mixture. The complete set of bases $\mathbf{W} = [\mathbf{W}^s, \mathbf{W}^b]$ and their corresponding gains $\mathbf{H} = [\mathbf{H}^s; \mathbf{H}^b]$ allows us to construct entirely the observed signal.

The NMF algorithm proposed by [21] is able to decompose an input non-negative matrix $\mathbf{X}$ into a product of two non-negative matrices $\mathbf{WH}$. More specifically, given a matrix $\mathbf{X}$, NMF finds the matrices $\mathbf{W}$ and $\mathbf{H}$ that minimize the reconstruction error between $\mathbf{X}$ and $\mathbf{WH}$. In NMF, the reconstruction error is measured by a function of the form $D(\mathbf{X}, \mathbf{WH}) = \sum_{ft} d([\mathbf{X}]_{ft}, [\mathbf{WH}]_{ft})$, where $d(a, b)$ gives a measure of divergence between two scalars. The most common forms of divergence d used in NMF can be generalized under the so-called β-divergence

$$d_\beta(a,b) = \begin{cases} \frac{1}{\beta(\beta-1)}(a^\beta + (\beta-1)b^\beta - \beta a b^{\beta-1}), & \beta \in \Re^+ \setminus \{0,1\} \\ a \log \frac{a}{b} - a + b, & \beta = 1 \\ \frac{a}{b} - \log \frac{a}{b} - 1, & \beta = 0. \end{cases} \quad (3)$$

The Euclidean distance ($\beta = 2$), the Kullback-Leibler (KL) divergence ($\beta = 1$) and the Itakura-Saito (IS) divergence ($\beta = 0$) are typical measures employed in NMF.

The attractiveness of NMF for source separation is due to its ability for finding patterns in data. If the analyzed data is naturally composed by a combination of patterns, NMF tends, to a certain extent, to decompose the data into its underlying components. This ability, however, is limited by the complexity of the data. In general, when applied to realistic audio signals, NMF is unable to find adequate bases for separation, i.e., there is no guarantee that each extracted basis contains information from a single source, or even if it does, it is extremely difficult to identify the source of origin. For this reason, practical audio separation algorithms introduce certain prior knowledge about the sources in the decomposition process. A common practice is to learn a set of bases (or dictionary) for each individual source in advance, usually by decomposing training material where each source is available in isolation. For example, in the case of speech and noise, one could learn bases $\mathbf{W}^s$ from a clean speech database, and bases $\mathbf{W}^b$ from a noise database. During separation, the learned basis vectors $\mathbf{W} = [\mathbf{W}^s, \mathbf{W}^b]$ are kept fixed, and only their gains $\mathbf{H} = [\mathbf{H}^s; \mathbf{H}^b]$ are estimated. This *supervised* approach obtains good results when the sources in the mixture maintain the properties seen in the training stage, but any mismatch decreases the quality. In addition, if a source has a great variability, a high number of bases is needed to capture all possible realizations of the source, making the method computationally more complex. Another remedy is to impose certain mathematical restrictions to

the bases and gains. In standard NMF, the only criterion for estimating $\mathbf{W}$ and $\mathbf{H}$ is to minimize the reconstruction error $D(\mathbf{X}, \mathbf{WH})$, and no additional constrains are imposed for $\mathbf{W}$ and $\mathbf{H}$ (except for non-negativity). If these constraints encode, in some way, the behavior of the sources, it is possible to obtain bases and gains with relation to sources, thus enabling separation. Finally, a third solution is to define source-specific generative models, in which the basis vectors or the way they combine have a different mathematical expression for each source. The model described so far, expressed in (1), assumes that the source is generated by a simple linear combination of its basis vectors. However, more complex models can be defined for certain sources, for example, inspired in the physical principles involved in their production. The source/filter model explained in the next subsection is an example of this strategy.

2.2 Source/Filter Model for Speech

In [9], a compositional source/filter model is proposed to represent the signal of interest (usually a lead music instrument or singing voice), and to allow its discrimination from the remaining sources. This representation is specially interesting for vocal sounds, since it approximates their underlaying production characteristics. According to the model, each speech spectral vector $\mathbf{s}_t$ is decomposed into an excitation part $\mathbf{s}_t^{F_0}$ multiplied by a filter part $\mathbf{s}_t^{\Phi}$, which are respectively composed by a linear combination of P elementary excitation bases $\mathbf{w}_p^{F_0}$ and E elementary filter bases $\mathbf{w}_e^{\Phi}$, as follows:

$$\mathbf{s}_t = \mathbf{s}_t^{\Phi} \bullet \mathbf{s}_t^{F_0} = \left(\sum_{e=1}^{E} h_{et}^{\Phi} \mathbf{w}_e^{\Phi} \right) \bullet \left(\sum_{p=1}^{P} h_{pt}^{F_0} \mathbf{w}_p^{F_0} \right), \tag{4}$$

where h_{et}^{Φ} and $h_{pt}^{F_0}$ are non-negative gains, and $\bullet$ denotes the Hadamard product. The excitation bases $\mathbf{w}_p^{F_0}$ represent the discrete collection of sounds from which the signal can be constructed, and which are further modulated by a combination of bases $\mathbf{w}_e^{\Phi}$. Since the model is designed to represent vocals, it is convenient that each vector $\mathbf{w}_p^{F_0}$ represents the glottal signal corresponding to an individual fundamental frequency or pitch. In [9], the bases $\mathbf{w}_p^{F_0}$ are generated using the glottal source model KLGLOTT88 [18], resulting in a fixed dictionary of pitch-related excitations. If a sufficient number of excitations P is used, it is possible to have a fine grid of pitch values, thus covering the whole pitch range of the speaker with enough resolution. On the other hand, the filter bases $\mathbf{w}_e^{\Phi}$ must be able to represent the smooth envelop of the signal. In [9], these bases are generated from a family of smooth functions, resulting in a fixed dictionary of smooth components that can combine to represent any arbitrary smooth envelope.

The P excitation bases $\mathbf{w}_p^{F_0}$ can be grouped into a matrix $\mathbf{W}^{F_0} = [\mathbf{w}_1^{F_0}, \ldots, \mathbf{w}_P^{F_0}]$, and the E filter bases $\mathbf{w}_e^{\Phi}$ into a matrix $\mathbf{W}^{\Phi} = [\mathbf{w}_1^{\Phi}, \ldots, \mathbf{w}_E^{\Phi}]$. Following this notation, the model in (4) can then be written in matrix form as

$$\mathbf{S} = (\mathbf{W}^{\Phi}\mathbf{H}^{\Phi}) \bullet (\mathbf{W}^{F_0}\mathbf{H}^{F_0}). \tag{5}$$

Here, the gain matrices $[\mathbf{H}^{\Phi}]_{et} = h_{et}^{\Phi}$ and $[\mathbf{H}^{F_0}]_{pt} = h_{pt}^{F_0}$ give the decomposition of the speech source into the dictionaries $\mathbf{W}^{\Phi}$ and $\mathbf{W}^{F_0}$. These amplitudes are estimated from the input signal, while the dictionaries are kept fixed.

This source/filter model has two interesting advantages. First, it describes a generative model that is characteristic of speech, and significantly discriminative from typical noise sounds, which usually do not fit this structure. And second, since the pitch and timbre information is individually represented, it provides a more structured description of speech, allowing the use of a reasonable number of bases. Although the model can be extended to deal also with unvoiced speech, this extension makes the model more sensitive to capture interferences. In this paper, we focus on separating voiced speech, and use this model as a base of our method.

3 Proposed Regularized Decomposition for Speech and Noise Separation

The method described here overcomes the above-mentioned limitations of standard NMF when applied to source separation, with particular emphasis on speech and noise sources. Our method relies on two main aspects: first, a signal model describing speech and noise with different mathematical expressions, and second, a decomposition algorithm that further improves speech and noise isolation by imposing specific constraints to the background noise, inspired by generic properties of common noises.

In our algorithm, we use a time-frequency representation with logarithmic scale, which is computed as follows. First, the time-domain input signal is divided into frames with a length of 64 ms and 50% frame overlap, windowed by a Hamming window. Then the STFT is computed and discretized to a resolution of $1/4$ semitones by integrating the magnitude values within each defined band, with $1/4$ semitones width. Specifically, if $\chi(\Omega, t)$ is the STFT of frame t and Ω is frequency in Hz, the f^{th} frequency element of the spectrogram is computed as

$$x_{ft}^{o} = \sum_{\mathcal{F}(f)}^{\mathcal{F}(f+1)} |\chi(\Omega, t)| \, \mathrm{d}\Omega, \tag{6}$$

where $\mathcal{F}(f) = 32.7 \cdot 2^{(f-1)/48}$ Hz, with $f = 1, \ldots, F$. We use $F = 380$ frequency points, thus representing spectral information from 32.7 Hz to approximately 8000 Hz.

In order to make the algorithm independent from the norm and length of the input signal, the spectrogram is normalized taking into account the parameter β that will be used in the β-divergence and the size of the spectrogram, as follows:

$$x_{ft} = \frac{x^o_{ft}}{\left(\frac{1}{FT}\sum_{f=1}^{F}\sum_{t=1}^{T} {x^o_{ft}}^{\beta}\right)^{1/\beta}}, \tag{7}$$

where F is the number of frequency elements and T is the number of frames.

3.1 Signal Model

The proposed signal model can be viewed as an extension of the generic framework described in [9], with appropriate modifications for the problem of speech/noise separation. The input spectrogram, represented by the $F \times T$ dimensional matrix $\mathbf{X}$, is modeled as an instantaneous sum of four contributions, in the form

$$\mathbf{X} \approx \widehat{\mathbf{X}} = \underbrace{(\mathbf{W}^{\Phi}\mathbf{Z}^{\Phi}\mathbf{H}^{\Phi}) \bullet (\mathbf{W}^{F_0}\mathbf{Z}^{F_0}\mathbf{H}^{F_0})}_{\text{speech}} + \underbrace{\mathbf{W}^{N}\mathbf{Z}^{N}\mathbf{H}^{N} + \mathbf{W}^{I}\mathbf{Z}^{I}\mathbf{H}^{I} + \mathbf{W}^{H}\mathbf{Z}^{H}\mathbf{H}^{H}}_{\text{noise}}. \tag{8}$$

The first term represents the target speech by means of the source/filter model described above. The remaining three terms represent the background noise as a sum of three different types of noises, each expressed by a different matrix factorized into a set of basis functions and activations. The signal model is denoted by $\widehat{\mathbf{X}}$. For mathematical convenience, all basis vectors (columns in basis matrices) and their amplitudes across time (rows in gain matrices) are assumed to have unit L_2 norm, e.g., $\|\mathbf{w}^{\Phi}_e\| = 1$ and $\|\mathbf{h}^{\Phi}_e\| = 1$, where $\mathbf{h}^{\Phi}_e = \left[h^{\Phi}_{e1}, \ldots, h^{\Phi}_{eT}\right]$ and $\|\cdot\|$ denotes the L_2 norm. To enable the representation of arbitrary signal levels, the amplitude information is stored in diagonal matrices $\mathbf{Z}^{\Phi}$, $\mathbf{Z}^{F_0}$, $\mathbf{Z}^{N}$, $\mathbf{Z}^{I}$ and $\mathbf{Z}^{H}$, such that, for example, $\mathbf{Z}^{\Phi} = \mathrm{diag}(\mathbf{z}^{\Phi})$, where $\mathbf{z}^{\Phi}$ is a vector of weighs with E elements, one per basis function. For the unknown matrices, which are estimated iteratively, this normalization is maintained in the inference algorithm by rescaling the vectors after each iteration. Working with normalized vectors has the advantage that all restrictions can be defined with independence of the scale of the input signal and the algorithm parameters. Also, since all bases and amplitude sequences have the same importance, any possible bias towards choosing certain patterns is avoided, thus providing a more meaningful decomposition.

Each considered noise class is characterized by a specific spectro-temporal behavior and, consequently, their respective bases and gains are characterized

individually, by imposing specific mathematical constrains. In our method, the following types of noise are defined:

- The first type, referred to as "smooth noise", is characterized by a spectrogram with small variations between adjacent values. This noise can be modeled as a combination of bases with a smooth spectral shape (smoothness in frequency), which are activated in time with smooth amplitude variations (smoothness in time). This noise is expressed in the term $\mathbf{W}^N\mathbf{Z}^N\mathbf{H}^N$, where the matrix $\mathbf{W}^N$ is composed of R_n basis vectors.

- The second type, referred to as "impulsive noise", describes those broadband noises consisting in bursts of energy that concentrate in short time intervals. This noise can be modeled by smooth spectral patterns (smoothness in frequency) that activate in isolated time instants (sparseness in time). It is captured by the term $\mathbf{W}^I\mathbf{Z}^I\mathbf{H}^I$, where $\mathbf{W}^I$ has R_i bases.

- The last type, referred to as "pitched noise", contains narrowband interferences that are usually pitched, and that remain active in consecutive frames. Consequently, this noise can be modeled by means of sparse spectral vectors (sparseness in frequency), whose gains demonstrate slow variations across time (smoothness in frequency). This noise is captured in $\mathbf{W}^H\mathbf{Z}^H\mathbf{H}^H$, where the matrix $\mathbf{W}^H$ contains R_h basis vectors.

The reason for decomposing the noise into three individual noise classes is to provide a certain flexibility to the model, giving it the ability to capture different types of possible interferences. These noise classes are quite generic, and enable to describe a wide range of acoustic scenes. Contrary to other methods, in which the noise patterns are trained, our algorithm learns the noise basis and their gains directly from the input data. Since each noise is modeled by combining several bases, it is possible to catch noises with certain spectral variability (i.e., non-stationary).

The source/filter model used to represent the speech signal is conceptually identical to the one described in the previous section. However, two relevant observations are made. First, in order to capture the natural intonation of speech, the excitation dictionary $\mathbf{W}^{F_0}$ must cover the complete pitch range with enough resolution. In our method, the dictionary $\mathbf{W}^{F_0}$ is composed of $P = 612$ excitations, such that they cover the pitch range from 60 Hz to 350 Hz with logarithmically spaced values. Specifically, each glottal excitation $\mathbf{w}_p^{F_0}$ is generated with fundamental frequency

$$\mathrm{F0}(p) = 2^{(p-1)/(12\cdot 20)}60, \quad p = 1, \ldots, 612, \tag{9}$$

obtaining a resolution of 20 pitches per semitone. This resolution is assumed to be high enough to approximate any arbitrary speech contour. Regarding the filter dictionary $\mathbf{W}^{\Phi}$, a significant modification is introduced in our model. Instead of using a family of elementary smooth functions as in [9], the filter bases are trained beforehand on a database of isolated speech phonemes, such that each phoneme envelope is represented by a certain number of filter bases. Since the number of phonemes is limited and each phoneme has homogeneous properties, it is possible to represent the filter information of speech with a reasonable number of basis vectors. In the next section, the procedure employed to construct $\mathbf{W}^{\Phi}$ is detailed, along with important insights about the chosen number of filters.

3.2 Decomposition Algorithm

The set of model parameters $\mathbf{\Theta} = \{\mathbf{H}^{\Phi}, \mathbf{H}^{F_0}, \mathbf{W}^{N}, \mathbf{H}^{N}, \mathbf{W}^{I}, \mathbf{H}^{I}, \mathbf{W}^{H}, \mathbf{H}^{H}\}$ is estimated by minimizing a cost function $C(\mathbf{\Theta})$ that is a sum of two kind of terms: a reconstruction error term $D(\mathbf{X}, \widehat{\mathbf{X}})$ and several noise penalty terms, which enforce specific properties for each type of noise. The cost function is given by

$$C(\mathbf{\Theta}) = D(\mathbf{X}, \widehat{\mathbf{X}}) + C_N(\mathbf{W}^N, \mathbf{H}^N) + C_I(\mathbf{W}^I, \mathbf{H}^I) + C_H(\mathbf{W}^H, \mathbf{H}^H). \quad (10)$$

Here, the reconstruction error $D(\mathbf{X}, \widehat{\mathbf{X}})$ is measured by the β-divergence, as explained in Section 2.2. The class-specific noise cost functions $C_N(\cdot)$, $C_I(\cdot)$ and $C_H(\cdot)$ enforce the algorithm to obtain noise bases and gains that satisfy certain noise-like properties, thus reducing the likelihood of capturing speech components.

Restrictions for Smooth Noise

As mentioned above, the parameters $\mathbf{W}^N$ and $\mathbf{H}^N$ in (10) are intended to capture all those types of noise characterized by a smooth spectrogram, in which adjacent values demonstrate small energy variations. This generic definition is able to describe a wide variety of interferences such as white, pink or babble noises, which have been handled by considering similar assumptions in previous works [19]. The interest of defining this class of noise is based on the fact that speech and many noises have clearly different spectro-temporal appearances. While speech is composed of strongly marked harmonic combs, most noises tend to have a relatively flat spectrum, without significant peaks. Consequently, if this smooth behavior is imposed to the noise, a highly discriminative decomposition can be achieved, thus helping the speech source/filter model. This smooth noise does not have to be necessarily stationary, since the noise model, that employs several bases, allows certain variability.

Given this behavior, it is reasonable to model this noise by a combination of smooth spectral shapes, whose gains are also smooth across time. In other words, each column of $\mathbf{W}^N$ can be forced to be smooth, and each row of $\mathbf{H}^N$ can also be constrained to be smooth. Consequently, a high cost must be assigned to large changes between consecutive frequencies in $\mathbf{W}^N$, and to large changes between consecutive frames in $\mathbf{H}^N$. In our method, the spectral smoothness of matrix $\mathbf{W}^N$ is measured by a function $SSM(\cdot)$, expressed as the sum of the squared differences between consecutive frequencies

$$SSM(\mathbf{W}^N) = \sum_{r=1}^{R_n} \sum_{f=2}^{F} (w_{fr}^N - w_{f-1,r}^N)^2, \tag{11}$$

where $[\mathbf{W}^N]_{fr} = w_{fr}^N$. Similarly, the temporal smoothness of matrix $\mathbf{H}^N$ is measured by a function $TSM(\cdot)$, which penalizes large variations in the temporal direction

$$TSM(\mathbf{H}^N) = \sum_{r=1}^{R_n} \sum_{t=2}^{T} (h_{rt}^N - h_{r,t-1}^N)^2, \tag{12}$$

with $[\mathbf{H}^N]_{rt} = h_{rt}^N$. This penalty function is identical to the temporal continuity criterion proposed in [30]. Based on these restrictions, the cost function for the bases and gains of the smooth noise can then be written as

$$C_N(\mathbf{W}^N, \mathbf{H}^N) = \alpha \gamma_W^N SSM(\mathbf{W}^N) + \alpha \gamma_H^N TSM(\mathbf{H}^N), \tag{13}$$

where α is a positive coefficient that weighs the importance of the smoothness restrictions in the global cost function, and γ_W^N and γ_H^N are normalization constants. Assuming that the columns of $\mathbf{W}^N$ are normalized to have L_2 norm equal to 1, the maximum value of the function $SSM(\mathbf{W}^N)$ is $2R_n(F-1)/F$, which is reached when $\mathbf{W}^N$ is extremely non-smooth (i.e., when its columns alternate zeros and maximum activations). Consequently, the constant $\gamma_W^N = F/(2R_n(F-1))$ normalizes the smoothness measure $SSM(\mathbf{W}^N)$ to a maximum value of 1, making it independent from the number of frequencies and bases. Similarly, assuming that the rows in $\mathbf{H}^N$ have normalized L_2 norm, the constant $\gamma_H^N = T/(2R_n(T-1))$ normalizes the temporal smoothness measure of $\mathbf{H}^N$, making it independent from the number of frames and bases.

Restrictions for Impulsive Noise

Other types of noise appear in the mixture spectrogram as sudden bursts of energy, often covering a broad frequency range for a reduced time. These sounds can be viewed as transients, with most of their energy concentrated in short time intervals, and with the possibility of being repetitive. Certain types of machine noise,

gunfire or microphone popping are examples of these category. Unlike the above noise class, these noises cannot be considered smooth across time, and therefore require a separate characterization.

Since these interferences have a relatively smooth short-time spectrum (at least, in comparison with speech), it is natural to enforce the spectral patterns in $\mathbf{W}^I$ to be smooth, as in the previous case. By contrast, their corresponding gains in $\mathbf{H}^I$ must be sparse across time, meaning that they only activate in isolated time instants. There are several ways for measuring sparsity properties in a matrix. In our study, we have chosen the method employed in [30], in which the sparsity cost function $SP(\cdot)$ is formulated as a L_1 norm

$$SP(\mathbf{H}^I) = \sum_{r=1}^{R_i} \sum_{t=1}^{T} h_{rt}^I, \tag{14}$$

where $[\mathbf{H}^I]_{rt} = h_{rt}^I$. This function penalizes non-zero entries in matrix $\mathbf{H}^I$, leading to a solution where a few elements are active. The smoothness of the basis vectors in $\mathbf{W}^I$ is controlled by the function $SSM(\cdot)$, as in (11). From both restrictions, the term constraining the bases and gains of the impulsive noise can be expressed as

$$C_I(\mathbf{W}^I, \mathbf{H}^I) = \alpha\gamma_W^I SSM(\mathbf{W}^I) + \lambda\gamma_H^I SP(\mathbf{H}^I), \tag{15}$$

where λ is a positive weight for equilibrating the importance of sparseness measures in the global cost function, α is the aforementioned weight for smoothness terms, and γ_W^I and γ_H^I are normalization constants. As before, it is assumed that the L_2 norm of each row in $\mathbf{H}^I$ equals unity. In that case, the maximum value of the function $SP(\mathbf{H}^I)$ is $R_i\sqrt{T}$, which is produced when all elements of the matrix are active with maximum energy. Hence, the sparseness measure can be normalized with the constant $\gamma_H^I = 1/(R_i\sqrt{T})$. As before, the smoothness function is normalized by $\gamma_W^I = F/(2R_i(F-1))$.

Restrictions for Pitched Noise

Certain types of interferences are composed by sounds which are nearly periodic, and whose spectrum demonstrates one or more significant spectral peaks. For example, the sound produced by common sources such as vehicle horns, sirens or certain machines follows this description. Because of this energy distribution, the short-time spectrum of these sounds does not fit the spectral smoothness restriction imposed to the above classes, and therefore will tend to be captured by the speech part of the model. In order to avoid this, a specific characterization is required, which must be also sufficiently discriminative from speech. In this

sense, two important properties are observed. First, except for certain cases, the spectrum of theses noises is generally not harmonic, but consisting in isolated peaks that concentrate most of the energy in a few coefficients. And second, unlike speech intonation, the pitch produced by these sounds usually remains stable during a number of frames, often with slow energy variations. Consequently, we should expect the spectral patterns in $\mathbf{W}^H$ to be highly sparse, with only a few active elements, while their corresponding gains in $\mathbf{H}^H$ should be relatively smooth across time.

In our algorithm, this behavior is imposed to the parameters $\mathbf{W}^H$ and $\mathbf{H}^H$ by applying the cost function $SP(\cdot)$, described above, to $\mathbf{W}^H$, thus enforcing spectral sparseness, and by applying the function $TSM(\cdot)$, expressed in (12), to the gains $\mathbf{H}^H$, thus enforcing temporal stability. The overall penalty term for the pitched noise can be expressed as

$$C_H(\mathbf{W}^H, \mathbf{H}^H) = \lambda \gamma_W^H SP(\mathbf{W}^H) + \alpha \gamma_H^H TSM(\mathbf{H}^H), \tag{16}$$

where λ and α are the specific sparseness and smoothness weighting coefficients, and where the constants $\gamma_W^H = 1/(R_h\sqrt{F})$ and $\gamma_H^H = T/(2R_h(T-1))$ respectively normalize the sparseness and smoothness measures.

Estimation of the Parameters

The estimation algorithm is based on the multiplicative gradient strategy proposed in [21]. The unknown matrices of the model are first initialized to random positive numbers, and then they are alternatively updated at each iteration with multiplicative update rules. These update rules are appropriately derived to decrease the criterion given in (10), under the model approximation $\widehat{\mathbf{X}}$ formulated in (8). In each iteration, after a certain matrix has been updated with its corresponding rule, the model $\widehat{\mathbf{X}}$ is recomputed before updating the subsequent matrices, repeating the process until a certain number of iterations is reached.

In the objective function in (10), the gain matrices $\mathbf{H}^{F_0}$ and $\mathbf{H}^{\Phi}$ describing the decomposition of the speech part are only restricted by the reconstruction error. Consequently, the problem of estimating $\mathbf{H}^{F_0}$ and $\mathbf{H}^{\Phi}$ is the same as in [9], i.e. reducing the β-divergence between $\mathbf{X}$ and the model $\widehat{\mathbf{X}}$, which is accomplished

by the following update rules:

$$\mathbf{H}^{F_0} \leftarrow \mathbf{H}^{F_0} \bullet \frac{(\mathbf{W}^{F_0}\mathbf{Z}^{F_0})^{\mathrm{T}}\left((\mathbf{W}^{\Phi}\mathbf{Z}^{\Phi}\mathbf{H}^{\Phi}) \bullet \widehat{\mathbf{X}}^{\beta-2} \bullet \mathbf{X}\right)}{(\mathbf{W}^{F_0}\mathbf{Z}^{F_0})^{\mathrm{T}}\left((\mathbf{W}^{\Phi}\mathbf{Z}^{\Phi}\mathbf{H}^{\Phi}) \bullet \widehat{\mathbf{X}}^{\beta-1}\right)}$$
$$\mathbf{H}^{\Phi} \leftarrow \mathbf{H}^{\Phi} \bullet \frac{(\mathbf{W}^{\Phi}\mathbf{Z}^{\Phi})^{\mathrm{T}}\left((\mathbf{W}^{F_0}\mathbf{Z}^{F_0}\mathbf{H}^{F_0}) \bullet \widehat{\mathbf{X}}^{\beta-2} \bullet \mathbf{X}\right)}{(\mathbf{W}^{\Phi}\mathbf{Z}^{\Phi})^{\mathrm{T}}\left((\mathbf{W}^{F_0}\mathbf{Z}^{F_0}\mathbf{H}^{F_0}) \bullet \widehat{\mathbf{X}}^{\beta-1}\right)} \tag{17}$$

where $\bullet$ is the Hadamard product, $(\cdot)^{\mathrm{T}}$ is the transpose operator and all fractions and exponentials are applied element by element.

The noise matrices, on the other hand, are further restricted by their corresponding regularization terms. The update rules for these matrices can be derived knowing that, for a NMF-based decomposition problem, the multiplicative updating rule for a generic matrix $\mathbf{A}$ can be expressed as [30]

$$\mathbf{A} \leftarrow \mathbf{A} \bullet \frac{\nabla^{-}_{\mathbf{A}} C(\mathbf{\Theta})}{\nabla^{+}_{\mathbf{A}} C(\mathbf{\Theta})}, \tag{18}$$

where $\nabla_{\mathbf{A}} C(\mathbf{\Theta})$ is the partial derivative of the objective function $C(\mathbf{\Theta})$ with respect to $\mathbf{A}$, such that it is expressed as a difference of two entry-wise positive terms $\nabla_{\mathbf{A}} C(\mathbf{\Theta}) = \nabla^{+}_{\mathbf{A}} C(\mathbf{\Theta}) - \nabla^{-}_{\mathbf{A}} C(\mathbf{\Theta})$. Observe that, since these two terms are both non-negative, the gradient applied to $\mathbf{A}$ is also non-negative.

Our particular objective function in (10) can be written as a weighted sum of its constituting functions $D(\cdot)$, $SSM(\cdot)$, $TSM(\cdot)$ and $SP(\cdot)$. Consequently, the update rule corresponding to a certain noise matrix can be expressed in terms of the partial derivatives of the functions affected by the considered matrix. The derivatives of these functions, evaluated for a generic noise component $\mathbf{WZH}$, are expressed as

$$\begin{aligned}
\nabla_{\mathbf{W}} D(\mathbf{X}, \widehat{\mathbf{X}}) &= \widehat{\mathbf{X}}^{\beta-1}(\mathbf{ZH})^{\mathrm{T}} - (\widehat{\mathbf{X}}^{\beta-2} \bullet \mathbf{X})(\mathbf{ZH})^{\mathrm{T}} \\
\nabla_{\mathbf{H}} D(\mathbf{X}, \widehat{\mathbf{X}}) &= (\mathbf{WZ})^{\mathrm{T}}\widehat{\mathbf{X}}^{\beta-1} - (\mathbf{WZ})^{\mathrm{T}}(\widehat{\mathbf{X}}^{\beta-2} \bullet \mathbf{X}) \\
[\nabla_{\mathbf{W}} SSM(\mathbf{W})]_{fr} &= 4[\mathbf{W}]_{fr} - 2\left([\mathbf{W}]_{f-1r} + [\mathbf{W}]_{f+1r}\right) \\
[\nabla_{\mathbf{H}} TSM(\mathbf{H})]_{rt} &= 4[\mathbf{H}]_{rt} - 2\left([\mathbf{H}]_{rt-1} + [\mathbf{H}]_{rt+1}\right) \\
\nabla_{\mathbf{H}} SP(\mathbf{H}) &= \mathbf{1} - \mathbf{0}
\end{aligned} \tag{19}$$

where $\mathbf{1}$ is an all-ones matrix and $\mathbf{0}$ a all-zeros matrix. Observe that each derivative is here expressed as a subtraction of two positive parts, thus enabling the formulation of any gradient.

We can obtain the complete set of update rules by particularizing the expressions in (18) and (19) for each noise matrix of the model. For example, the multiplicative update rules for the noise components $\mathbf{W}^I$ and $\mathbf{H}^I$, constrained respectively by the functions $SSM(\cdot)$ and $SP(\cdot)$, can be written as

$$\begin{aligned}\mathbf{W}^I &\leftarrow \mathbf{W}^I \bullet \frac{\nabla^-_{\mathbf{W}^I} D(\mathbf{X}, \widehat{\mathbf{X}}) + \alpha\gamma^I_W \nabla^-_{\mathbf{W}^I} SSM(\mathbf{W}^I)}{\nabla^+_{\mathbf{W}^I} D(\mathbf{X}, \widehat{\mathbf{X}}) + \alpha\gamma^I_W \nabla^+_{\mathbf{W}^I} SSM(\mathbf{W}^I)} \\ \mathbf{H}^I &\leftarrow \mathbf{H}^I \bullet \frac{\nabla^-_{\mathbf{H}^I} D(\mathbf{X}, \widehat{\mathbf{X}}) + \lambda\gamma^I_H \nabla^-_{\mathbf{H}^I} SP(\mathbf{H}^I)}{\nabla^+_{\mathbf{H}^I} D(\mathbf{X}, \widehat{\mathbf{X}}) + \lambda\gamma^I_H \nabla^+_{\mathbf{H}^I} SP(\mathbf{H}^I)}\end{aligned} \tag{20}$$

In our implementation, at the end of each iteration, all basis matrices are normalized column-wise and their activations are normalized row-wise. In order to keep the amplitude information, the weighting vectors are rescaled appropriately. For example, in the case of $\mathbf{W}^I$ and $\mathbf{H}^I$, the matrix $\mathbf{Z}^I$ is rescaled as $\mathbf{Z}^I \leftarrow \mathrm{diag}\left(\left[\|\mathbf{w}^I_1\|\|\mathbf{h}^I_1\| z^I_1, \ldots, \|\mathbf{w}^I_{R_i}\|\|\mathbf{h}^I_{R_i}\| z^I_{R_i}\right]\right)$, and the normalization is carried out as $\mathbf{W}^I \leftarrow \left[\frac{\mathbf{w}^I_1}{\|\mathbf{w}^I_1\|}, \ldots, \frac{\mathbf{w}^I_{R_i}}{\|\mathbf{w}^I_{R_i}\|}\right]$ and $\mathbf{H}^I \leftarrow \left[\frac{\mathbf{h}^I_1}{\|\mathbf{h}^I_1\|}; \ldots; \frac{\mathbf{h}^I_{R_i}}{\|\mathbf{h}^I_{R_i}\|}\right]$. This approach is remotely related to the Probabilistic Latent Component Analysis (PLCA) framework [28], in which the bases and their gains are viewed as probability distributions (normalized to sum 1), and a mixing weight is defined for each basis. Although the convergence of the algorithm is not guaranteed, due to the normalization steps, in practice it produces solutions that decrease the reconstruction error and the constraints.

3.3 Signal Synthesis

Once the parameters have been estimated, each component is synthesized in the time-domain. In our method, this separation is performed by applying Wiener filters to the mixture STFT, where the filter for each source is computed from the estimated spectrograms. For example, if $\widehat{\mathbf{S}} = (\mathbf{W}^\Phi \mathbf{Z}^\Phi \mathbf{H}^\Phi) \bullet (\mathbf{W}^{F_0} \mathbf{Z}^{F_0} \mathbf{H}^{F_0})$ is the estimated spectrogram of the speech signal, its Wiener spectro-temporal mask $\mathbf{M}^S$ is computed as

$$\mathbf{M}^S = \frac{\widehat{\mathbf{S}}^2}{\widehat{\mathbf{S}}^2 + (\mathbf{W}^N \mathbf{Z}^N \mathbf{H}^N)^2 + (\mathbf{W}^I \mathbf{Z}^I \mathbf{H}^I)^2 + (\mathbf{W}^H \mathbf{Z}^H \mathbf{H}^H)^2}. \tag{21}$$

Since the spectrograms are defined on a logarithmic frequency scale, it is necessary to convert this mask to linear scale before filtering the input STFT. In practice, the filtering is done by multiplying all frequency bins within the same logarithmic band by the same corresponding masking coefficient. The filtered spectrum is then transformed to the time-domain by the inverse STFT.

4 Learning of Phoneme Filters

The dictionary of filter bases $\mathbf{W}^{\Phi}$ is learned from a database of isolated speech phonemes. For this purpose, we used the TIMIT database [12], which is composed of thousands of utterances pronounced by 630 speakers, where the phonemes are annotated. Since we are focused on separating voiced speech, we learned basis filters only for pitched phonemes. Specifically, we selected the following sets of phonemes: vowels (iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy, ow, uh, uw, ux, er, ax, ix, axr), nasals (m, n, ng, em, en, eng, nx), voiced stops (b, d, g), voiced affricates (jh), voiced fricatives (z, zh, v, dh), semivowels and glides (l, r, w, y, hh, hv, el). In order to learn filters for a certain phoneme, all instances of that phoneme were grouped into a single $F \times T_v$ spectrogram $\mathbf{X}_v$, which was then decomposed into the following source/filter model:

$$\mathbf{X}_v \approx \widehat{\mathbf{X}}_v = (\mathbf{W}_v^{\Phi}\mathbf{Z}_v^{\Phi}\mathbf{H}_v^{\Phi}) \bullet \mathbf{S}_v^{F_0}, \tag{22}$$

where $\mathbf{W}_v^{\Phi}$ is the set of filters we want to learn (composed of E_v vectors), $\mathbf{H}_v^{\Phi}$ are their corresponding unknown amplitudes, $\mathbf{Z}_v^{\Phi} = \mathrm{diag}(\mathbf{z}_v)$ is a diagonal matrix of weights and $\mathbf{S}_v^{F_0}$ is a $F \times T_v$ excitation matrix, in which each column contains the excitation signal corresponding to the F_0 of frame t, generated with the KLGLOTT88 model.

Since the TIMIT database does not provide annotated pitch information, we annotated each utterance automatically. For doing this, we used two different pitch estimators: the PEFAC algorithm [15] and a modified version of the YIN method [8], in which the results were post-processed to obtain certain time continuity. All instances of a phoneme for which both algorithms provided the same pitch sequence were considered well annotated, and incorporated to the spectrogram $\mathbf{X}_v$. Since these estimators are based on completely different principles, it is reasonable to consider similar results as evidence of a correct estimation. The matrix $\mathbf{S}_v^{F_0}$ was then generated from the estimated pitches, by accordingly grouping the excitation signals of all instances of the phoneme.

The decomposition in (22), however, does not provide useful bases by itself. Without further restrictions, there is the risk of learning bases that do not represent a complete envelope, but only parts of it, thus losing any phoneme characterization. In addition, since the envelope of speech signals is smooth in nature, each basis in $\mathbf{W}_v^{\Phi}$ must be consequently smooth, and therefore accordingly restricted. Also, since only a single vocal tract is articulated in each frame, the decomposition must be as sparse as possible, meaning that only a few filter bases (ideally, only one) must be active at each time. Enforcing sparsity to $\mathbf{H}_v^{\Phi}$ also enables to obtain bases that represent the complete envelope, because the few active filters in each frame must cover all the spectral content . In order to impose these proper-

ties, the spectrogram is decomposed by minimizing the following weighted cost function:

$$C_v(\mathbf{W}_v^\Phi, \mathbf{H}_v^\Phi) = D(\mathbf{X}_v, \widehat{\mathbf{X}}_v) + \alpha_v \gamma_W^\Phi SSM(\mathbf{W}_v^\Phi) + \lambda_v \gamma_H^\Phi SP(\mathbf{H}_v^\Phi), \qquad (23)$$

where $SSM(\cdot)$ imposes spectral smoothness to $\mathbf{W}_v^\Phi$, and $SP(\cdot)$ imposes sparseness to $\mathbf{H}_v^\Phi$. The constants $\gamma_W^\Phi = F/(2E_v(F-1))$ and $\gamma_H^\Phi = 1/(E_v\sqrt{T})$ normalize these measures, and the weights α_v and λ_v equilibrate the importance of each term. In our experiments, we tested different values for α_v and λ_v. Although it is difficult to determine optimal values, mainly because the quality of the obtained bases is not measurable, any values around $\alpha_v = 1$ and $\lambda_v = 0.1$ were found to produce satisfactory decompositions. The matrices are found iteratively by applying the following update rules:

$$\mathbf{W}_v^\Phi \leftarrow \mathbf{W}_v^\Phi \bullet \frac{\left(\mathbf{S}_v^{F_0} \bullet \widehat{\mathbf{X}}_v^{\beta-2} \bullet \mathbf{X}_v\right)(\mathbf{Z}_v^\Phi \mathbf{H}_v^\Phi)^{\mathrm{T}} + \alpha_v \gamma_W^\Phi \nabla^-_{\mathbf{W}_v^\Phi} SSM(\mathbf{W}_v^\Phi)}{\left(\mathbf{S}_v^{F_0} \bullet \widehat{\mathbf{X}}_v^{\beta-1}\right)(\mathbf{Z}_v^\Phi \mathbf{H}_v^\Phi)^{\mathrm{T}} + \alpha_v \gamma_W^\Phi \nabla^+_{\mathbf{W}_v^\Phi} SSM(\mathbf{W}_v^\Phi)}$$

$$\mathbf{H}_v^\Phi \leftarrow \mathbf{H}_v^\Phi \bullet \frac{(\mathbf{W}_v^\Phi \mathbf{Z}_v^\Phi)^{\mathrm{T}}\left(\mathbf{S}_v^{F_0} \bullet \widehat{\mathbf{X}}_v^{\beta-2} \bullet \mathbf{X}_v\right) + \lambda_v \gamma_H^\Phi \nabla^-_{\mathbf{H}_v^\Phi} SP(\mathbf{H}_v^\Phi)}{(\mathbf{W}_v^\Phi \mathbf{Z}_v^\Phi)^{\mathrm{T}}\left(\mathbf{S}_v^{F_0} \bullet \widehat{\mathbf{X}}_v^{\beta-1}\right) + \lambda_v \gamma_H^\Phi \nabla^+_{\mathbf{H}_v^\Phi} SP(\mathbf{H}_v^\Phi)} \qquad (24)$$

As before, each basis vector in the model and each amplitude row is normalized. A vector $\mathbf{z}_v$ with size E_v is used to store the amplitude information for each filter basis, which is maintained in a similar way as explained before. In this case, however, these variables provide useful information about the importance of each learned basis for approximating the input spectrogram. That is, by inspecting $\mathbf{z}_v$, it is possible to infer how many bases are necessary to account for most of the envelop information of a certain phoneme. The filter bases with significantly higher weights can be selected as representative patterns of the studied phoneme, discarding the remaining vectors. In our experiments, we decomposed each phoneme into a model with $E_v = 100$ filter bases, and then manually selected the most significant filters by examining their respective weights $\mathbf{z}_v$. Between 3 and 4 bases were selected per phoneme, resulting in a dictionary $\mathbf{W}^\Phi$ with only 129 basis vectors. An example result for the phoneme ae is given in Fig. 1, depicting the selected filters and their respective weights.

It is worth noting that the obtained filters are really averaged shapes of each phoneme envelope. They are not building atoms, neither actual instances of phoneme filters, as occurs in the overcomplete dictionaries used in other speech representations. Therefore, the model may not be able to capture all spectral details characteristic of a particular speaker or pronunciation, but only to provide a sufficient approximation. Theoretically, as derived from the learning process, the

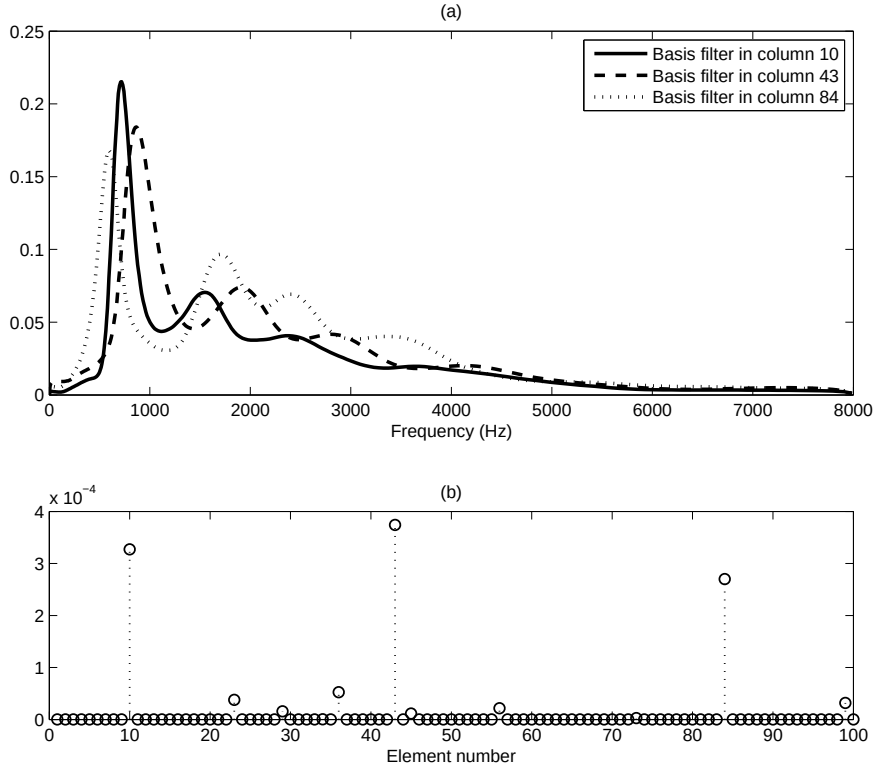

Figure 1: Learning of basis filters for phoneme ae. (a) Selected basis filters from $\mathbf{W}_v^{\Phi}$, corresponding to the columns 10, 43 and 84. (b) Obtained weight vector $\mathbf{z}_v$, which indicates that the most significant filters are in the columns 10, 43 and 84 .

selected bases for each phoneme can combine to represent most instances of the phoneme with a small reconstruction error, but some details will be lost. Also, since these filters impose a strict representation of speech, the model should be less sensitive to the capture of noise components. In other words, the use of such a limited number of filters should make the model less flexible, in exchange of increasing its discrimination capability.

5 Experimental Results

We have tested our algorithm on the development dataset of the 3rd CHiME Speech Separation and Recognition Challenge [1], which features talkers speaking in real-world noisy environments recorded using a six-channel device. Specifically, we have employed the simulated subset of the corpus, in which the mixtures

are generated by artificially mixing clean speech data (recorded in a sound proof booth) with noisy backgrounds. The background noise signals were recorded in four different public environments: bus (BUS), cafeteria (CAF), pedestrian area (PED) and street junction (STR). The dataset consists of 410 utterances in each of these environments, giving a total of 1640 utterances, which are produced by 4 speakers by reading sentences from the Wall Street Journal (WSJ-0) corpus.

Three metrics are used to assess the performance of the developed system: the source-to-distortion ratio (SDR), which provides information on the overall quality of the separated speech, the source-to-interference ratio (SIR), which is a measure of the presence of noise components in the output speech signal, and the source-to-artifacts ratio (SAR), which measures the importance of the artificial components introduced by the method [29, 11].

In addition, preliminary results of our method (with a not optimal setting of the configuration parameters) were submitted to the SiSEC 2013 evaluation campaign for speech and real-world background noise separation [23]. The SiSEC development and test datasets are composed of two-channel artificial mixtures of speech and real-world noise captured in six different environments: subway car moving (Su1), subway car standing at station (Su2), two different cafeterias (Ca1 and Ca2), and two different squares (Sq1 and Sq2). The development dataset is composed of 9 mixtures including only the environments Su1, Ca1 and Sq1. The test dataset consists of 20 audio files, and includes the six types of noise.

In our experiments, the CHiME dataset is used to make a large-scale and thorough evaluation of the method, investigating different settings of its parameters and comparing its performance with other semi-supervised NMF approaches. On the other hand, the results on the SiSEC dataset are provided here to show the performance of the system on a more noisy scenario, and in comparison with state-of-the-art methods based on different techniques. Sound samples of our experiments on the CHiME dataset can be found at http://www4.ujaen.es/∼damian/speech.html. For the SiSEC evaluation, the obtained separation measures, along with the output speech waveform for each file, can be found at http://sisec.wiki.irisa.fr/tiki-index.php?page=Two-channel+mixtures+of+speech+and+real-world+background+noise.

5.1 Results on the CHiME Development Set

We evaluate the method for different combinations of the following parameters: α (weight of the smoothness constrains), λ (weight of the sparseness constrains), R_n (number of bases for the smooth noise), R_i (number of bases for the impulsive noise), R_h (number of bases for the pitched noise) and β.

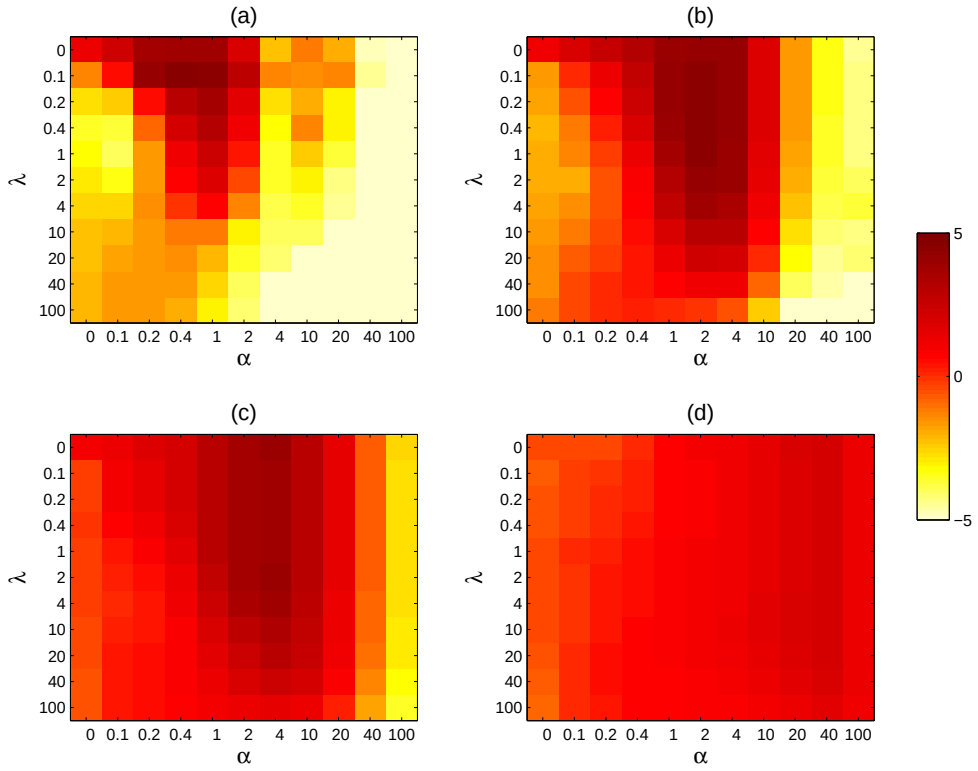

Figure 2: Average SDR gain obtained on the CHiME development dataset as a function of the parameters α, λ and β. (a) $\beta = 0$, (b) $\beta = 0.5$, (c) $\beta = 1$ and (d) $\beta = 2$.

Fig. 2 illustrates the optimization of the parameters α and λ for different values of β. Here, we employ the average SDR gain (that is, the difference between the original SDR, measured without performing separation, and the achieved SDR) obtained on the CHiME development set as a criterion to decide the best combination. Observe that the values $\alpha = 0$ and $\lambda = 0$ corresponds to the case where no restrictions are applied to the background noise, and therefore the algorithm tries only to minimize the β-divergence. In this case, the method obtains a SDR gain approximately equal to 1.39 dB for $\beta = 0$, which suggests that the source/filter model is effective in separating the target speech without further restrictions. It can be seen that applying the proposed restrictions on the background noise increases significantly the separation quality. The best results (4.56 dB) are achieved for $\alpha = 0.4$, $\lambda = 0.1$ and $\beta = 0$, obtaining an improvement of 3.17 dB over the best results of the method without restrictions. For $\beta = 0.5$, the best SDR gain is equal to 4.5 dB, and is reached when $\alpha = 2$ and $\lambda = 0.4$. For $\beta = 1$, the best per-

formance is obtained with $\alpha = 4$ and $\lambda = 2$, with an SDR gain of 3.97 dB. In this experiment, the number of noise bases is set to $R_n = R_i = R_h = 50$. Preliminary experiments showed that the optimization of α and λ does not depend critically on the number of noise bases, as long as a sufficient number of bases is defined. In all cases, the number of iterations was set to 50, which was found sufficient to reach convergence at a reasonable computational cost.

The results in Fig. 2 illustrate that the smoothness constraints (weighted by parameter α) have a greater impact on the performance, which means that the background noise in the dataset is represented more accurately by smooth components. We also observe that, assuming the best combination for α and λ, the IS divergence ($\beta = 0$) provides better results than other values of β. The reason for this is that the IS divergence is more sensitive to differences in low energy elements, which comprise a significant portion of the speech spectrum. Since the employed source/filter model explicitly incorporates a sparse representation of speech, by means of a dictionary of excitation bases, our method can potentially obtain a closer reconstruction under the IS divergence. Higher values of β allow slightly higher levels of noise in the separated speech for the same value of divergence. For this reason, as the value of β grows, the optimal values of α and λ must also be higher, meaning that heavier noise constraints are required to obtain similar performance.

Fig. 3 provides a more detailed study of the effect of the number of noise bases over the system performance. Here the parameters $\alpha = 0.4$, $\lambda = 0.1$ and $\beta = 0$ are kept fixed, and only the noise dictionary size is varied. In general, the system performs better as the number of bases is increased, although beyond a certain point the quality does not significantly improve. The method is specially sensitive to the number of bases for the smooth noise, R_n. In fact, when an appropriate number of smooth noise bases is set ($R_n \geq 50$), the remaining vectors do not produce significant variations, and small values of R_i and R_h can be used for higher computational efficiency. These results may be explained by the noise content of the database, which seems to be composed mainly of smooth noises. However, the results always show a certain improvement when the three types of bases are used. This figure demonstrates that the method is not specially dependent on the number of noise bases, and consequently any configuration with a sufficient number of bases produces satisfying results.

5.2 Comparison to other Methods

We consider three different approaches to compare the performance of our method: the optimally modified log spectral amplitude estimator (OM-LSA) [7], semi-supervised sparse NMF with trained speech bases (SNMF) and semi-supervised

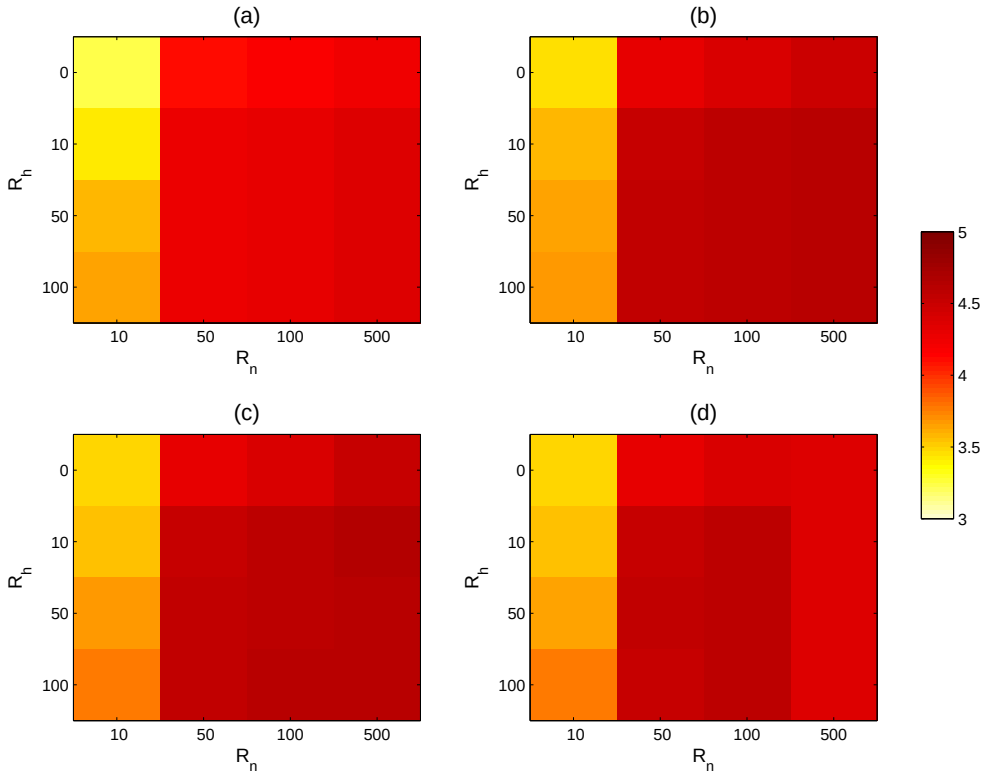

Figure 3: Average SDR gain obtained on the CHiME development dataset as a function of R_n, R_i and R_h. (a) $R_i = 0$, (b) $R_i = 10$, (c) $R_i = 50$ and (d) $R_i = 100$.

sparse NMF with randomly selected speech exemplars (ENMF). For SNMF and ENMF, we use 1000 speech bases and 150 noise bases, such that the noise bases are learned also at test time. Since this number of bases is similar (but slightly larger) to that used in our method, we can get a fair conclusion about what method provides the best separation. For SNMF, speech bases are learned by factorizing the training subset of the TIMIT database, while in ENMF the speech bases are randomly selected from this same subset. For both cases, we use the same data representation as in our method, described in Section 3, and the same number of iterations. For OM-LSA, we executed the code published by the authors. For SNMF and ENMF, we executed the "well done" version developed by [20], in which the objective function is defined in terms of normalized bases.

In order to make a thorough comparison, we also investigate different settings of β and the sparsity weight μ for SNMF and ENMF. Specifically, we evaluate $\beta \in \{0, 0.5, 1, 2\}$ and $\mu \in \{0, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$. Fig. 4 shows the

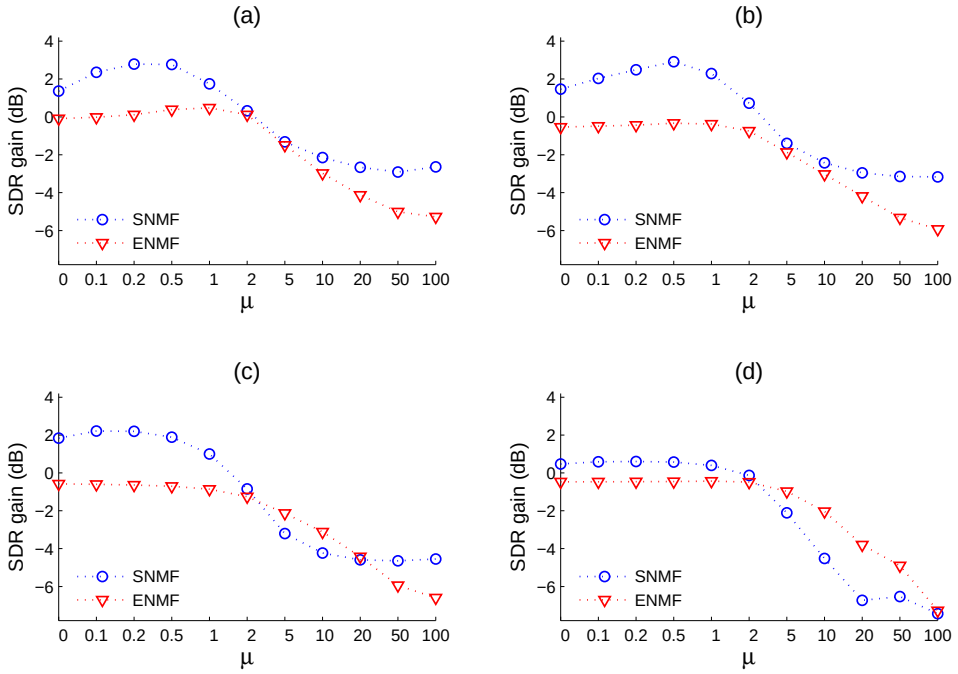

Figure 4: Average SDR gain obtained by SNMF and ENMF on the CHiME development set as a function of the sparsity weight μ. (a) $\beta = 0$, (b) $\beta = 0.5$, (c) $\beta = 1$ and (d) $\beta = 2$.

average SDR gain obtained on the CHiME development set for all of the evaluated configurations. As seen, the best performance for SNMF is obtained with $\beta = 0.5$ and $\mu = 0.5$, giving a SDR gain of 2.91 dB. For ENMF, the best configuration is $\beta = 0$ and $\mu = 1$, with a SDR gain of 0.47 dB. Observe that these results are lower than the best measure reached by our algorithm.

Figure 5 compares the separation measures obtained on the CHiME dataset by the best configuration of each method. The line in the middle of each box represents the mean value over the dataset, while the lower and upper lines of each box show the 25th and 75th percentiles of the measures. The lines extending above and below each box show the extent of the rest of the values, excluding outliers (extreme values in the data). As seen, our method outperforms the other approaches, obtaining an average improvement in SDR of 1.65 dB over SNMF, and 0.67 dB over OM-LSA. It is interesting to note that our system is considerably more robust to interferences than the remaining approaches, as demonstrated by the SIR measures, mainly due to the strict representation of the speech signal imposed by our model. However, probably this is also the reason why our

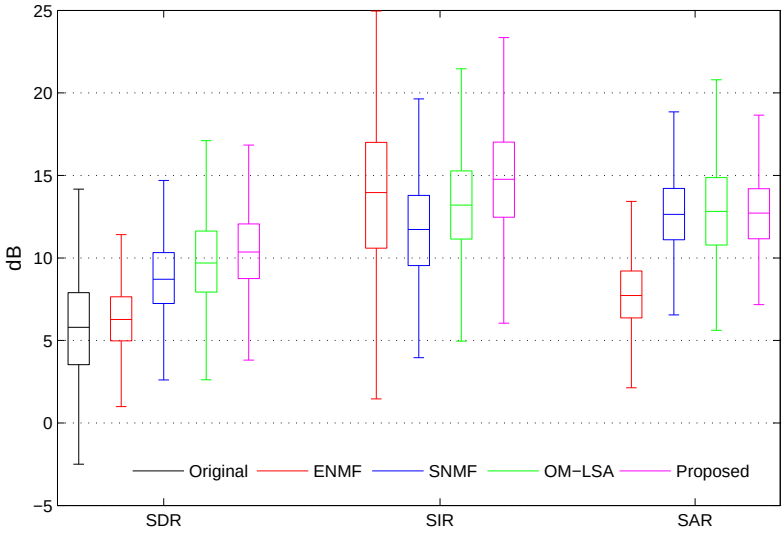

Figure 5: SDR, SIR and SAR measures obtained on the CHiME development dataset by the best setting of the methods. The original SDR of the noisy dataset is also depicted.

method produces similar or worse levels of artifacts than SNMF and OM-LSA. The poor results obtained by ENMF can be explained by the fact that the method is running here in a semi-supervised fashion, with exemplars covering only one time-frame. Such is not the usual setting for ENMF, which often involves the use of multi-frame exemplars (thus increasing the computational cost) and a noise dictionary. Similarly, the results for SNMF could be significantly improved by training noise bases. However, our intent here is not to explore the full potential of these methods, but to offer a comparison in the same conditions, with similar prior information. These results demonstrate that, in a semi-supervised scenario, the proposed restrictions in conjunction with source/filter modeling of speech provides a better separation than sparse NMF, even though our system is limited to voiced speech.

Figure 6 offers the same comparison for each individual environment in the CHiME dataset. It can be seen that the proposed approach consistently outperforms the other algorithms for all noise conditions, which suggests that the proposed constraints and their setting are robust against multiple acoustic scenarios.

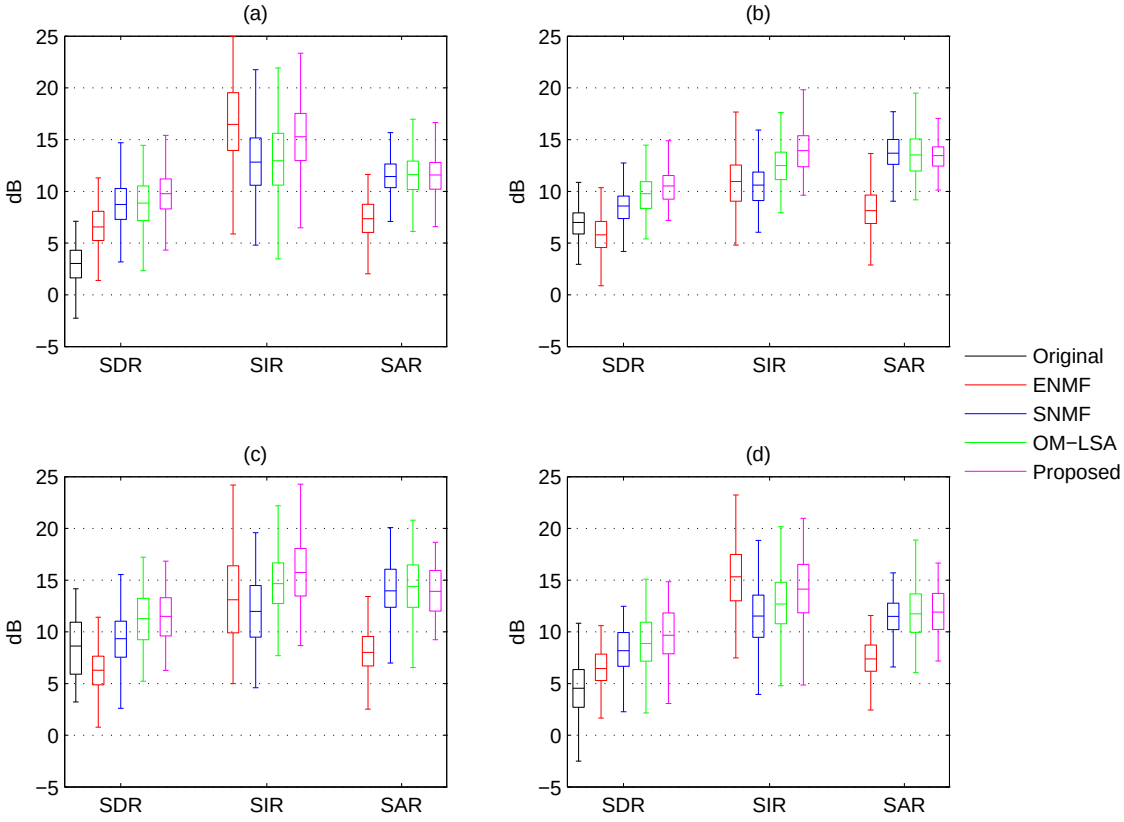

Figure 6: SDR, SIR and SAR measures obtained by the best setting of the methods on the four environments of the CHiME development dataset. (a) BUS, (b) CAF, (c) PED and (d) STR.

5.3 Performance Evaluation at SiSEC 2013

Tables 1 and 2 provide the resulting average separation measures for the environments included in the development and test datasets of SiSEC 2013. Here, the algorithm is running with parameters $\beta = 1$, $\alpha = 4$ and $\lambda = 2$. Although this setting is not optimal, it should be good enough for evaluating the performance of the algorithm. These results were measured by the organizers of the campaign over the single-channel submitted speech signals. In addition, the tables illustrate the results obtained by other algorithms submitted to the campaign providing also a single-channel output, and whose methodologies are described in [34] and [4]. The method by [34] is based on classic overdetermined blind source separation

Table 1: Average SDR, SIR and SAR results for each environment in the development dataset of SiSEC 2013.

	Original	Proposed			Wang			Bryan		
	SDR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
Ca1	1.9	5.4	15.4	6.1	5.8	20.9	6.0	5.6	18.4	5.9
Sq1	-3.9	9.6	17.3	10.7	11.6	19.3	12.5	10.2	15.6	12.1
Su1	-11.7	1.5	5.8	5.8	3.8	18.1	4.1	4.2	13.6	4.9
All	-3.7	6.4	14.1	8.1	7.9	19.6	8.4	7.3	16.1	8.4

Table 2: Average SDR, SIR and SAR results for each environment in the test dataset of SiSEC 2013.

	Proposed			Wang			Bryan		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
Ca1	3.4	14.6	4.1	4.2	19.6	4.4	3.7	13.9	4.5
Ca2	3.7	17.1	4.0	4.6	20.9	4.7	3.8	16.5	4.2
Sq1	8.9	18.6	9.9	11.6	22.7	12.0	13.1	21.8	13.7
Sq2	10.9	20.5	11.5	13.8	21.2	14.8	12.9	18.2	14.6
Su1	5.0	23.2	5.2	5.2	22.1	5.3	5.6	21.4	5.7
Su2	2.2	5.9	6.0	6.1	25.2	6.2	5.6	23.0	5.7
All	6.1	17.1	7.0	8.0	21.6	8.3	7.8	18.5	8.5

techniques, in which the permutation problem is solved following an original approach, and the resulting speech is further enhanced by a post-filter. The method by [4] consists in a regularized PLCA algorithm, in which the regularizations are based on graphical annotations made by the user over the spectrogram. This algorithm can be viewed as a constrained NMF with KL divergence, in which penalty masks are defined interactively for each source in the spectrogram. Although both algorithms make use of certain prior information (multichannel information in the first case, and user interaction in the second case), they enable to compare our method against state-of-the-art results, obtained over the same material.

As observed in the tables, although our algorithm performs worse than the remaining approaches, it achieves competitive results for the three considered measures in most of the environments. The method only fails to separate the speech source in certain mixtures with subway noise, particularly for Su1 in the development set and Su2 in the test set. In the first case, this can be explained by the high level and the specific properties of the background noise, consisting in a moving car that produces a pitched noise with varying fundamental frequency.

The proposed constraints are unable to represent this type of noise (because it is not smooth in the time direction), which is then captured by the speech model. In the second case, the background noise contains an interfering tone and a burst of noise produced by the doors of the car. Although these noises can be theoretically modeled by the system, it seems that the parameters α and λ are not optimal in this case. A higher value of λ could improve the separation performance in this situation.

6 Conclusion

We proposed a NMF-based algorithm for voiced speech and noise separation, in which the noise components are constrained to obey certain mathematical properties that are characteristic of many background noises. These properties are not defined for specific noises, but in a generic way, enabling to apply the algorithm to a wide range of environments without requiring prior training or a large number of bases. The speech source is represented through a source/filter model previously proposed in the literature, with the incorporation of a dictionary of filter bases learned in a training stage from a database of isolated phonemes. This speech model allows to represent the speech signal with a reasonable number of bases, and consequently is computationally more efficient than other speech representations based on compositional models, such as those based on sparse coding.

The method was evaluated on the simulated mixtures of the 3rd CHiME development set. The experiments demonstrate that, in general, the proposed restrictions are adequate for real-word background environments, and improve significantly the results obtained by the model without restrictions. In comparison with other constrained semi-supervised decompositions, such as sparse NMF with speech bases, our method obtains better separation results. The method also obtained promising results at the SiSEC 2013 international campaign, although the proposed restrictions were not able to characterize appropriately one of the tested environments. The current results of the algorithm, although promising, may however not be usable for most practical applications, due to its limitation to voiced speech.

Future improvements of the algorithm will be focused on characterizing certain properties of the noise more accurately. It was observed that, although the smooth noise type is able to approximate many instances of noise, it often imposes a strict representation in a sense that the bases and amplitudes must be smooth. Better results could be obtained if the bases and gains are not restricted to be smooth, but their combination is (at least, in comparison with speech). In addition, we intend to explore the incorporation of unvoiced parts to the speech

model. Restricting the activations of unvoiced phonemes may help in avoiding the problem of capturing noise components with them. It is also interesting to explore a real-time implementation of the system, or its application in conjunction with spatial cues for multichannel signals.

References

[1] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' Speech Separation and Recognition Challenge: Dataset, task and baselines", in *Proc. of IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

[2] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction", in *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[3] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*, MIT Press, Cambridge, MA, 1994.

[4] N. Bryan and G. Mysore, "An efficient posterior regularized latent variable model for interactive sound source separation", in *Proc. of the 30th International Conference on Machine Learning (ICML)*, May 2013, vol. 28. pp. 208–216.

[5] F. J. Cañadas-Quesada, P. Vera-Candeas, N. Ruiz-Reyes, J. Carabias-Orti, and P. Cabañas-Molero, "Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints", *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 26, pp 1–17, 2014.

[6] J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. Cañadas-Quesada, "Musical instrument sound multi-excitation model for non-negative spectrogram factorization", *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1144–1158, 2011.

[7] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments", *Signal Processing*, vol. 81, no. 11, pp. 2403–2418, 2001.

[8] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music", *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[9] J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation",

IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 6, pp. 1180–1191, 2011.

[10] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[11] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL Toolbox User Guide – Revision 2.0", Technical Report 1706, IRISA, 2005.

[12] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "TIMIT Acoustic-Phonetic Continuous Speech Corpus", Linguistic Data Consortium, Philadelphia, 1993.

[13] J. T. Geiger, F. Weninger, A. Hurmalainen, J. F. Gemmeke, M. Wollmer, B. Schuller, G. Rigoll, and T. Virtanen, "The TUM+TUT+KUL approach to the CHiME Challenge 2013: Multi-stream ASR exploiting BLSTM networks and sparse NMF", in *Proc. of 2nd CHiME Workshop held in conjunction with ICASSP'13*, 2013, pp. 25–30.

[14] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2067–2080, 2011.

[15] S. Gonzalez and M. Brookes, "PEFAC - A pitch estimation algorithm robust to high levels of noise", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 518–530, 2014.

[16] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2067–2079, 2010.

[17] A. Hurmalainen, J. F. Gemmeke, and T. Virtanen, "Modelling non-stationary noise with spectral factorisation in automatic speech recognition", *Computer Speech and Language*, vol. 27, no. 3, pp. 763–779, 2013.

[18] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers", *The Journal of the Acoustical Society of America*, vol. 87, no. 2, pp. 820–857, 1990.

[19] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama, "Single and multiple F0 contour estimation through parametric spectrogram modeling of speech in noisy environments", *IEEE Transactions on Audio, Speech, and Language Processing*, vol.15, no. 4, pp. 1135–1145, 2007.

[20] J. Le Roux, J. R. Hershey, and F. Weninger, “Sparse NMF – half-baked or well done?”, MERL Technical Report, TR2015-023, 2015.

[21] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization”, in *Proc. of Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[22] N. Mohammadiha, P. Smaragdis, and A. Leijon, “Supervised and Unsupervised Speech Enhancement Using Nonnegative Matrix Factorization”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, 2013.

[23] N. Ono, Z. Koldovsky, S. Miyabe, and N. Ito, “The 2013 signal separation evaluation campaign”, in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013, pp. 1–6.

[24] A. Ozerov, E. Vincent, and F. Bimbot, “A general flexible framework for the handling of prior information in audio source separation”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no.4, pp. 1118–1133, 2012.

[25] B. Raj, R. Singh, and T. Virtanen, “Phoneme-dependent NMF for speech enhancement in monaural mixtures”, in *Proc. of International Conference on Spoken Language Processing (INTERSPEECH)*, 2011, pp. 1217–1220.

[26] M. N. Schmidt and R. K. Olsson, “Single-channel speech separation using sparse non-negative matrix factorization”, in *Proc. of International Conference on Spoken Language Processing (INTERSPEECH)*, 2006, pp. 2614–2617.

[27] U. Simsekli, J. Le Roux, and J. R. Hershey, “Non-negative source-filter dynamical system for speech enhancement”, in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6206–6210.

[28] P. Smaragdis, B. Raj, and M. Shashanka, “Sparse and shift-invariant feature extraction from non-negative data”, in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2008, pp. 2069–2072.

[29] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[30] T. Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–

1074, 2007.

[31] T. Virtanen and A. Cemgil, "Mixtures of gamma priors for non-negative matrix factorization based speech separation", in *Proc. of Independent Component Analysis and Signal Separation (ICA)*, 2009, pp. 646–653.

[32] T. Virtanen, J. F. Gemmeke, B. Raj, and P. Smaragdis, "Compositional models for audio processing", *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 125–144, 2015.

[33] T. Virtanen and A. Klapuri, "Analysis of polyphonic audio using source-filter model and non-negative matrix factorization", in *Advances in Models for Acoustic Processing, Neural Information Processing Systems Workshop*, 2006.

[34] L. Wang, H. Ding, and F. Yin, "A region-growing permutation alignment approach in frequency-domain blind source separation of speech mixtures", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 549–557, 2011.

[35] F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation", in *Proc. of International Speech Communication Association (INTERSPEECH)*, 2014, pp. 865–869.

[36] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, The MIT Press, 1964.

[37] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors", in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 4029–4032.