Fundamental frequency estimation techniques for multi-microphone speech input

Thesis · January 2005

CITATIONS
6

READS
2,048

2 authors:

Federico Flego
University of Cambridge
20 PUBLICATIONS **140** CITATIONS
SEE PROFILE

Maurizio Omologo
Fondazione Bruno Kessler
197 PUBLICATIONS **3,950** CITATIONS
SEE PROFILE

PhD Dissertation

International Doctorate School in
Information and Communication Technologies

DIT - University of Trento

Fundamental frequency estimation techniques for multi-microphone speech input

Federico Flego

Advisor:

Maurizio Omologo

ITC-irst Centro per la Ricerca Scientifica e Tecnologica (Trento)

March 2006

to Clara

Acknowledgments

My deepest gratitude goes to *Clara*, who I met just before starting the PhD and who has always supported me during the many difficult moments. I have no words to thank her enough, and her unconditional love is the most important lesson I have ever received.

I also thank my parents *Gianfranco* and *Laura* and my sister *Francesca*, who have always encouraged me and supported me, and not just morally! It makes me very proud to know that I can count on them, wherever I may be and at any time.

I wish to thank all my friends who, despite neglecting them for some time, never stopped their encouragement and enthusiasm. I send my deepest affection to *Sabrina* and *Stefano*, colleagues during the PhD and real friends. Both of them had a baby during their studies, and it is also for this reason that I admire them... to the point that I would like to emulate them soon! An affectionate thought goes also to *Leonardo*, *Emanuela*, *Italo* and *Valentina*, with whom I need just a few words and a glass of good wine to let our complicity emerge.

A particular thanks also goes to *Alfiero* and *Luca*, who "squeezed" into their office to make room for me when I moved to the ITC-irst. They helped me a lot, Alfiero always available for a technical and less technical chat, Luca with his witty and sharp jokes. Beside them, I would really like to thank all the guys of the *"goviniitc"*, who helped me in the difficult moments with their contagious free-and-easy spirit.

I would also like to thank *Arianna* and *Paolo* for everything they have done and continue doing respectively as representatives of the PhD students of the DIT and president of the ADI of Trento.

When I started the PhD I had professor *Alessandro Zorat* as a tutor, who always fulfilled my needs even when these were strictly in contrast

with his own. He deserves special thanks because he was never conditional upon my choices and he was always available and affectionate to me. I regret not having been able to understand sooner his incredible qualities, and I apologise for this.

The research work presented in this thesis was then carried out at the ITC-irst, and could not have been accomplished without the supervision of my advisor, Maurizio Omologo, who I thank for giving me the opportunity to join the SHINE research group.

During the years I spent in the ITC-irst I had the opportunity to interact with researchers who have been a very good example to me, both from a professional and "human" viewpoint: *Daniele Falavigna, Diego Giuliani, Edmondo Trentin, Fabio Brugnara, Gianni Lazzari, Marcello Federico, Marco Matassoni, Mario Zen, Mauro Cettolo, Michele Zanin, Nadia Mana, Oswald Lanz, Piergiorgio Svaizer, Roberto Gretter, Romeo Rizzi*. I have learned a lot from them, even from some simple chats, in particular I learned how important is to be humble and to respect each other.

All my gratitude goes also to *Fabrizio Granelli*, professor at the Engineering Faculty of the Trento University, for the trust he put in me several times by allowing me to be an assistant in the course he gave. Shy and clumsy the first time, experience allowed me to acquire confidence and to understand the difficulties and the responsibilities of the one who stays on the other side of the desk. I hope the students have been satisfied!

Moreover, I am deeply thankful to professors *Francesco de Natale, Gian Antonio Mian* and *Raffaele Parisi*, respectively of the University of Trento, Padova and of Roma, for their availability to be part of the external commission that participated in the final evaluation of the thesis.

During the last year of my PhD I had the possibility -thanks to my advisor- to spend some time in Japan, at the NTT Communication Science

Laboratories of Kyoto. They were six very intensive months and besides the excellent professional experience acquired, I had the opportunity to get closer to the Japanese culture thanks to some people of a rare sensitivity and intelligence. For this reason, besides thanking all the friends of the NTT, I sincerely, and with gratitude, thank *Masato Miyoshi, Sachiko Matsubara, Setsuko Kohaya, Shoji Makino* and *Shoko Araki*. I send a special embrace also to *Marc Delcroix*, without whom the Japanese experience would have not been as intensive and as intense.

Finally, thanks and a wish of serenity to all those people who apply themselves in what they are doing with seriousness and humbleness.

Abstract

In speech processing, the estimation of fundamental frequency (f_0) aims to measure the frequency with which the vocal folds vibrate during voiced speech. This task is generally performed exploiting signal processing techniques applied to the speech signal previously acquired by an acoustic sensor. f_0 represents a high-level speech feature which is exploited by many speech processing applications, such as speech recognition, speech coding and speech synthesis, to improve their performance. After decades of research and innovation, the performance of these pitch based speech applications has improved to the point that they are now robust for most practical applications. However, phenomena as noise and reverberation, characteristic of real-world acoustic scenarios, have still to be coped with. Currently, performance of f_0 estimation techniques, conceived to work on high-quality speech signals, drops dramatically whenever such adverse acoustic conditions are considered. To overcome these limitations, the proposed f_0 estimation algorithm exploits the information redundancy provided by a Distributed Microphone Network (DMN), which consists of a generic set of microphones localized in space without any specific geometry. The DMN outputs are parallelly processed in the frequency domain, and each channel reliability is evaluated to derive a common representation from which f_0 is finally obtained. Compared to state of the art f_0 estimation techniques, this approach demonstrated to be particularly robust. To show this fact, experimental results were obtained from tests conducted on international speech databases, acquired from real noisy and reverberant scenarios.

As a second example of f_0 based application in distant-talking contexts, a Blind Source Separation (BSS) system was addressed. To improve its separation performance a f_0 post-processing scheme, based on adaptive comb filters, was designed. Tests conducted on reverberant speech data confirmed the advantages of the proposed solution.

Keywords

Fundamental frequency, pitch, noise, reverberation, blind source separation

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Context

The context of this thesis is speech fundamental frequency (or pitch) estimation based on a multi-microphone speech input. Pitch estimation belongs to the Speech Processing area which, in turn, comprises many research disciplines such as electrical engineering (computer science, signal processing and acoustics), psychology (psychoacoustics and cognition) and linguistics (phonetics, phonology and syntax). The objective of pitch estimation is to measure the oscillation frequency (f_0) of the vocal folds in voiced speech. The estimated f_0 represents a useful source of information for many speech applications such as, among others, speech recognition, speech coding and speech synthesis. The particular acoustic scenario addressed in this thesis, considers speech signals acquired by a set of far field microphones, whose output results thus severely degraded by the environmental noise and reverberation effects. Pitch estimation based on such a microphone setup has not been largely addressed in the literature so far. However, it is likely to become a reference scenario considering the growing interest for pitch based speech applications designed to work in distant-talking contexts.

1.2 The Problem

During the speech production process the airflow produced by the lungs passes through the larynx, the pharyngeal, the oral and nasal cavity, finally radiating through lips and nose. The overall shape of the vocal tract actuates as a resonator which modulates the airflow to produce the desired sounds. When voiced speech is generated, as for example during vowel production, the vocal folds at the top of the larynx open and close in a quasi-periodic fashion for the air pressure which accumulates below them. The frequency of these oscillations is given the name of *speech fundamental frequency (f_0)* and is responsible for the perceived pitch of the produced sound. For this reason the f_0 estimation algorithms are also referred to as Pitch Detection Algorithms (PDAs), although what is actually measured is the vocal folds oscillating frequency, a physical measurement, not the consequent subjective perception.

To detect and estimate f_0 in voiced speech, modern approaches rely on digital signal processing techniques applied to the signals provided by one or more microphones, which are used to record the speaker. In these signals f_0 manifests itself as a periodic pattern in the time domain, or as a series of peaks in the frequency domain. In the first case the period with which the pattern repeats itself (T_0) coincides with the inverse of f_0 while, in the second case, f_0 determines the position of the first peak as well as the spacing between two adjacent peaks. Pitch estimation is thus generally carried out in one, or the other domain, trying to detect signal self-similarities or frequency peak positions, respectively.

The main difficulties encountered during the estimation procedure are mainly related with the inherent variability of the human voice on one side, and with the inevitable quality loss which occurs during speech signal acquisition, on the other side.

Speech variability principally accounts for intonation variation, magnitude dynamics and phone unit durations, which depend on the articulatory movements of the vocal tract or speech organs, occurring continuously during phonation, and on the varying air pressure produced by the lungs. This reflects in variations of the pitch period length and waveform shape in the time domain, or in changes of the peaks amplitude and position in the frequency domain.

The quality of the acquired signal instead, depends on several factors such as the clarity with which the speech is uttered, the noise and reverberation of the environment, and the distortion introduced by the acoustic sensors employed for the acquisition. Also the possible channel over which the signal is transmitted, contributes to the overall quality loss.

Current f_0 estimation techniques perform very well on speech signals that were clearly uttered and acquired by means of a close-talk microphone in a quiet environment. But when they have to deal with the above described detrimental factors, which represent real-world situations, performance drops dramatically. Since constraining the talker to be in a quiet environment and to use a close-talk microphone is not a feasible solution in practical applications, research in speech fundamental frequency estimation is currently focusing on more robust systems. These systems must be able to extract f_0 from real-world speech signals, that is, spontaneous speech acquired from one or more far field microphones, in a noisy and reverberant context. An example of such applications is given by the speech technologies applied to household appliances. In fact, in the domotics context, the end-user must have the maximum movement freedom and cannot be asked to continuosly wear a close-talk microphone.

Besides the problem of pitch estimation in a real-world context, the problem of speech enhancement based on f_0 information is also addressed.

Pitch information is exploited differently by many applications to improve the final outcome. Here, a Blind Source Separation (BSS) system is considered, which separates the different speech sources from a mixture of three different speakers talking simultaneously. The outputs of the considered system results often distorted because of the overlapping of the talkers signals in the time-frequency domain. Pitch information is thus used in this context to reduce the distortion effects thus improving the BSS system separation performance.

1.3 The Solution

The proposed solution for robust f_0 estimation in noisy and reverberant scenarios, exploits the information redundancy achievable when several acoustic sensors are employed to acquire a speech signal from different positions. The microphone setup employed is a Distributed Microphone Network (DMN). Originally proposed in [6], a DMN consists of a generic set of microphones localized in space without any specific geometry. Such a microphone setup, on the one hand, allows the talker to move freely in the space without being forced to wear a headset microphone or to keep a specific position as well as a specific head orientation. On the other hand, it provides speech acquisitions which result severely affected by the reverberation effect as well as by the noise generally present in real-world scenarios.

However, the speech signal quality loss can be compensated if the redundancy offered by the different DMN outputs is exploited. The proposed approach exploits such information redundancy processing all DMN channels in a parallel fashion, estimating blindly the reliability degree of each of them. The applied fusion method bases then on the most reliable channels to provide robust f_0 estimates.

Regarding the problem of enhancing the outputs of a blind source separation (BSS) system exploiting pitch information, a scheme based on f_0 driven adaptive comb filters is proposed. The scheme represents an extension of the binary mask based BSS system described in [10], and founds on the fact that the harmonic structure of voiced speech exhibits a regular pattern in the time-frequency domain. Whenever signals from simultaneous talkers overlap in this domain, such harmonic structure is deteriorated resulting in distorted separated signals. The f_0 driven adaptive filters objective then, is to restore the original harmonic structure of voiced speech segments basing on the pitch information previously extracted. The proposed scheme demonstrated to improve the original BSS system performance, while keeping the overall system complexity low.

1.4 Innovative Aspects

The main innovative aspect of this thesis is to perform robust f_0 estimation on speech signals acquired in a distant-talking scenario. Employing a Distributed Microphone Network (DMN) [6], the reverberant and noisy microphone outputs are parallelly processed in the frequency domain, to derive a common representation for all channels. The proposed fusion procedure takes into account the reliability of each contribute, which is estimated comparing each channel spectrum with a reference spectrum. The signal quality provided by each microphone depends on the talker position which is allowed, if the above microphone setup is used, to move freely in the considered scenario.

The proposed approach for robust pitch estimation, based on a set of far-field microphones, belongs to a new general scheme where real-world scenarios are considered for speech processing applications. The use of

pitch information as an additional descriptor of speech characteristics results advantageous in many practical contexts. To demonstrate this, a f_0 based adaptive comb filtering scheme was derived and reported in the second part of the thesis. The scheme was applied to a modified version of a binary mask based Blind Source Separation System [10], which separates the speech signals of three speakers uttering at the same time. The harmonic structure of each separated speech signal is enhanced by the f_0 driven adaptive comb filters, improving the BSS system performance, in particular when a reverberant scenario is addressed.

1.5 Structure of the Thesis

After a brief introduction of the speech *fundamental frequency* f_0 (or *pitch*) concept, **Chapter 2** describes the state of the art of speech fundamental frequency (or pitch) estimation algorithms. Several techniques are presented, and examples of their working principle applied to voiced speech segments are given. These algorithms either belong to the initial phase of pitch estimation research or are based on recent findings. The description of the former algorithms is included because they still constitute the basis for many of the modern proposed solutions.

Chapter 3 presents the human speech production mechanism, describing the role of the various organs involved, and their effect on the characteristics of the produced sound. The source-filter model is then introduced to approximate the physical process as a vocal tract filter driven by an excitation signal. Once the relation between the latter and f_0 is showed, the way that speech processing applications can employ pitch information to improve their performance, is described. Recently, these applications have become more and more robust to work in real-world noisy and reverberant

contexts. The rest of the chapter is thus devoted to the analysis of the noise and reverberation adverse effects on pitch estimation.

Chapter 4 shows the limitations of state of the art pitch extraction algorithms when tested on noisy and reverberant speech signals. These drawbacks are more evident as the acoustic sensors, employed for speech signal acquisition, are kept far from the end-user of pitch based systems. To overcome these limitations and to allow the talker to freely move in the space, independently from microphone position, the concept of Distributed Microphone Network (DMN) is introduced. This microphone setup is then exploited to derive a new pitch exctraction algorithm based on the Multi-microphone Periodicity Function (MPF). The performance of the proposed algorithm are then measured employing real world speech data and compared with those of other state of the art algorithms. Detailed results are presented in **Chapter 5**, where two different acoustic scenarios are addressed.

Chapter 6 provides an example of how pitch information can be exploited to enhance the quality of speech signals output by a Blind Source Separation (BSS) system. A specific reference BSS setup is considered, where the speech stream of each of three talkers speaking simultaneously is separated employing two microphones and binary time-frequency masks. A modification of the reference system is then proposed in two steps. First, continuous time-frequency masks are introduced to separate the signal contributes of each speaker. Then, a scheme based on a pitch extractor, and on f_0 driven adaptive comb filters, is integrated to further process each of the previously obtained outputs. Performance results and comparisons with the reference system are provided at the end of the chapter.

Chapter 7 contains a summary of results, conclusions and suggestions for future work.

Chapter 2

State of the Art

This chapter describes the state of the art of speech *fundamental frequency* (f_0) estimation algorithms. The term *pitch* is also used to indicate the fundamental frequency in this context, although pitch derives from psychoacoustic, where it refers more properly to the subjective perception produced by voiced speech. Both measures though, refer to the phenomenon occurring when voiced sounds are uttered, that is, to the periodic oscillation of the vocal folds. This oscillation is responsible for intonation and manifests itself in the sound pressure waveform, as a periodic pattern.

The aim of pitch estimation algorithms is to detect and measure the period length of the repeating pattern characteristic of voiced speech. This measure is referred to as *fundamental period* (T_0) and results to be the inverse of the fundamental frequency, that is, $T_0 = 1/f_0$.

In this chapter the description of several state of the art pitch estimation algorithms is given along with examples of their working principle applied to voiced speech segments. Their description is presented considering the classification given in [43]. The first section describes those algorithms that operate in the *time domain* while the second section gives an account of those based on *short-term analysis*.

The expression "time domain" refers to algorithms which directly an-

alyze the speech pressure waveform in order to detect specific temporal features, such as maxima and minima, that provide useful information to estimate f_0 on a period-by-period basis.

"Short-term analysis" instead, considers segments of the speech signal long enough to include several pitch periods. Different types of transformation are then applied to the data to obtain its representation in a different domain, more suitable for estimating f_0. The lag-domain and frequency domain are the most common operative domains for this class of algorithms, from which an averaged value for the considered speech segment is obtained.

What follows is a literature survey of some of the most famous and important techniques devised in the past 50 years, to perform speech fundamental frequency estimation. The reader interested in the innovative aspects of the proposed research work, can directly address Chapter 4.

2.1 Introduction

The algorithms which deal with the problem of estimating the fundamental frequency of a periodic or quasi-periodic signal are commonly referred to as pitch detection algorithm or PDA. What these techniques actually provide is the estimation of the fundamental frequency, reminding however the psychologically link between f_0 and *pitch*, which is defined by the American National Standards Institute (ANSI) as the *auditory attribute of sound according to which sounds can be ordered on a scale from low to high.*

Such definition is not world wide accepted though, and there is a lot of debate mostly related with the fact that it not possible to build a one-to-one relationship between pitch and frequency. Actually the term pitch is given two different meaning depending on the context in which it is used.

In the psychoacoustic field the term pitch is used to indicate an auditory sensation, that is, a subjective attribute. In the signal processing field instead, related with voice or music signals, pitch is used to indicate the oscillation frequency of the vocal folds or of a playing instrument, that is, the fundamental frequency. In this work, the term pitch will be used with the latter meaning.

Pitch detection algorithms have been investigated since early '60s and still there is a lot of research effort to improve PDAs performance measured on tasks becoming each time more and more difficult.

The common model for a generic PDA consists of three processing blocks [43, 61], as shown in Figure 2.1: the *preprocessor*, the *extractor* and the *post-processor*. The preprocessor block task is to mainly perform data reduction, or to apply linear or non linear transformation to the data before it is processed by the next block. Data reduction is often necessary since, depending on the extractor, there might be no need for certain time or frequency features of the signal to be analyzed. A common preprocessing step is to low-pass filter the signal to keep just the frequency content below a few kHz since, for most pitch extractors, the high frequency content of speech signal does not provide any additional information for fundamental frequency estimation.

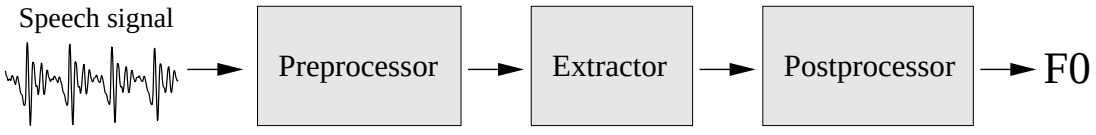

Figure 2.1: *Fundamental processing blocks of a PDA. Speech signal is first preprocessed in order to reduce data complexity or to apply linear or non linear transformations. The extractor block is responsible for estimating the signal fundamental frequency while the post-processing block can perform error detection or smoothing on the previously estimated* f_0 *values.*

A first distinction between PDAs is given in [43] and distinguishes be-

tween *time domain* and *short term analysis* based pitch extraction algorithms. The rule to label an algorithm as belonging to one or the other specific domain, is to consider the domain of the input to the extractor block. In case the input to the extractor is the signal itself, opportunely conditioned by the preprocessor, and pitch estimation is carried on a period basis, that is, the algorithm is capable of determining each individual period length, the algorithm belongs to the time domain category. When instead the preprocessor takes short-term intervals (frames) of the input signal, each including several periods, and provides the extractor with an alternate representation as, for example, the autocorrelation values (lag domain), or spectral values (frequency domain), the algorithm is said to belong to the short term analysis category. In this case each provided period length estimate can be considered an "average" of several contiguous period length values.

2.2 Time domain Pitch Determination

Time domain pitch determination is the oldest way to perform automatic pitch determination. The techniques presented in this section belong to the early phase of pitch estimation research and their performance have been substantially improved by modern approaches. However, they provide interesting and important insights on the subject constituting, in some cases, the basis of modern pitch estimation techniques. For this reason they are reported here and their simplified schematic grouping shown in Figure 2.2 will be used as a reference.

The principal characteristic of these algorithms is to perform pitch estimation on a period-by-period basis, that is, they are capable of determining the length of each individual period of the repeating pattern constituting the voiced speech signal.

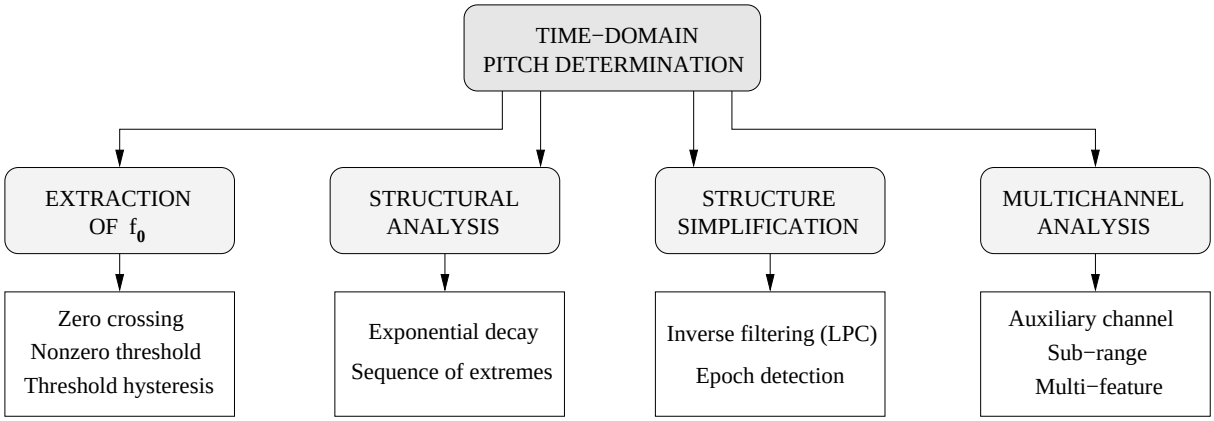

Figure 2.2: *Simplified diagram of time domain pitch determination algorithms [43].*

The Time domain based PDAs, can be further grouped into those which aim to extract the fundamental frequency, as the zero (or nonzero) threshold crossing based ones; those which perform structural analysis, basing on the periodic exponential decay characteristic of voicing sound; those performing waveform structure simplification in order to extract a sequence of extremes from which estimate the fundamental frequency and those computing parallel processing by means of multichannel[1] analysis.

2.2.1 Fundamental Frequency Extraction Algorithms

The simplest time domain based PDAs are the Zero-crossing Analysis Basic Extractor (ZXABE) and the Threshold Analysis Basic Extractor (TABE) [43], dating back to the 60s and developed on analog systems. That period also coincided with an advance of the computer in the domain of signal processing and these simple PDAs, and their derivations, were suitable for being implemented as computer programs. The basic principle of ZXABE, as shown in the left panel of Figure 2.3, is to produce a marker each time

[1]The term "multichannel" used here was inherited from [43]. It indicates that the data flow of a single input speech signal is duplicated to be processed by more than one processing unit, in a parallel fashion. It has not to be confused with the term "multi-microphone", which is used in this thesis to indicate several input speech signals, proceeding from different acoustic sensors.

the signal change polarity, that is, each time the signal amplitude changes from a negative to a positive value. As evident from the figure, doing this way a lot of markers are produced even within a single pitch period of the signal. The problem comes from the presence in the signal of the harmonics other than the fundamental frequency. This poses the necessity to preprocess the signal in order to provide the ZXABE, with a signal that has only two zero crossings per period. The latter is not easy to achieve since it requires to isolate the fundamental frequency or, at least, to enhance it while attenuating the other harmonics. Also phase of each harmonic should be taken in account, since on these values depends how much the waveform will reveal the presence of the fundamental.

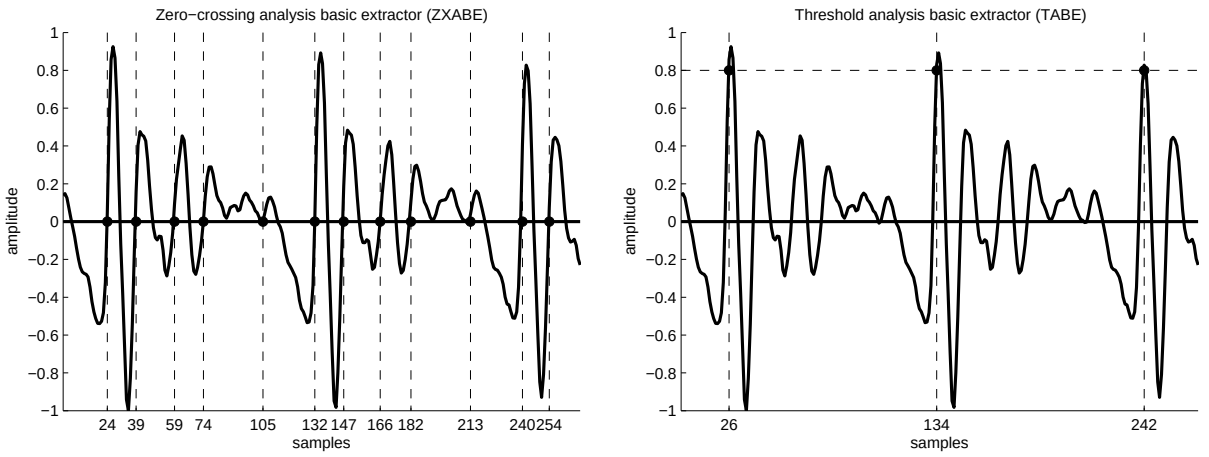

Figure 2.3: *Examples of PDA with zero-crossing (left) and threshold analysis (right) based extractor applied to a voiced segment of speech.*

An alternative is represented by the TABE which employs a higher threshold, in order to detect just the zero crossings relative to the fundamental frequency. In the example shown in the right panel of Figure 2.3, the threshold is set to 0.8 and the three markers positioned at samples 26, 134 and 242 respectively, determine exactly (with sample accuracy) the duration of the two periods shown. Even if more accurate than the ZXABE, also the TABE might miss some peaks necessary for correct marking

or detect others which are not. This comes from the difficulty of fixing a threshold which guarantees perfect periods length detection, since amplitude varies with time and there are cases where more than one high peak occur at each period. In the latter case, setting the threshold high enough to avoid false detection and low enough to avoid missing any target peak is not easy to accomplish.

Another solution is to set two positive threshold, a lower and higher one and to mark the beginning of a period only when the two are crossed successively. Signal values which cross just one threshold repeatedly will not generate a marker. This extractor is named "TABE with hysteresis" and improves the performance respect to the above described extractors. Still though, there is the need to fix the threshold values which may work for certain speech segment and not for others.

In the three cases a common preprocessing technique is to low-pass filter the speech signal in order to attenuate higher harmonics by about 6 to 12 dB per octave. The objective is to clear out higher harmonics so that threshold crossing is due just to the presence in the signal of the fundamental frequency. Though improving the performance, this rule of thumb is not suited for the voiced speech signal, which continuously varies the position and amplitude of the fundamental frequency and its harmonics. Additionally the fundamental frequency is not always present in voiced speech, even if it is perceptually perceived as dominant by the human ear. A lot of effort was put to design complex preprocessors implementing adaptive low-pass and non-linear filtering in order to enhance the frequency region were f_0 was expected to be and to cope with its variations during time.

Several non-linear filtering techniques were used in the preprocessors of these extractors, with the objective of flattening the voiced speech spectrum. The effect of formants is to enhance some harmonics while attenuating others. Designing a preprocessor which can provide the pitch extractor

an input independent from the particular sound uttered, that is, with a flat spectrum, was demonstrated to be particularly beneficial for estimating correctly f_0. To this aim half-wave or full-wave rectification, as well as squaring or peak clipping the signal were introduced in the preprocessor [103, 85].

Using thresholds, makes it necessary to normalize the signal amplitude but this cannot be accomplished in advance on the whole signal, since its dynamic varies with time. The solution in this case can be the use of a dynamic compressor to normalize over short-term portions of the signal.

The zero-crossing method and, more in general, those based on thresholding the signal in the time domain, provide good results on synthetic speech signals or on input recorded in quiet and non reverberant environment. When these conditions are not satisfied however, performance drop dramatically due to the fact that noise and, most of all, reverberation, affect the waveform severely in a way that those algorithms were not designed to cope with.

2.2.2 Structural Analysis

This approach considers the temporal structure of the speech waveform. The main idea behind this approach is the fact that the speech signal can be considered as the output of a pulse train coming from the glottis (source) filtered by the time-variant response of the vocal tract (filter). The latter actuates as a passive filter and its impulse response is, accordingly, a summation of exponentially dumped sinusoids whose envelope gradually decays away and whose maximum occurs at every excitation point, that is, at every source pulse. Observing the waveform of a voiced speech signal and considering the model behind it, it is possible for one to guess the periodicity out of the signal temporal structure. From this starting point

two algorithms were devised: the *envelope modeling* and the *sequence of extremes* based algorithms.

Envelope modeling

The envelope modeling approach bases on building a model of the signal temporal structure deriving a decaying envelope for thresholding the signal. The time instant at which the speech signal exceeds the modeled envelope, is assumed as the beginning of a new period. When this occurs, the envelope model is reset and used again as a threshold for determining the next source impulse instant. This algorithm bases very much on peak detection and on the discrimination between primary peaks (related with the source impulses) and secondary peaks, due to the oscillating behaviour of the decaying impulse response. The main difficult is thus to enhance primary peaks and to suppress other peaks, in order to find the appropriate time-constants which has to continuously fit the decaying behaviour of the speech signal. This value is critical for the correct detection of the beginning of each period: if it is too short, secondary peaks will wrongly trigger a new period estimation, while being it too long, primary peaks will be suppressed.

In the left panel of Figure 2.4 the algorithm applied to a segment of voiced speech with a time constant $\tau = 4\ ms$ is shown. In this case markers at samples values of 57, 165 and 272 are correctly positioned. In the right panel of Figure 2.4 instead, the wrong time constant $\tau = 2\ ms$ is set and a many period markers are wrongly detected.

This problem is most likely to happen when the speech signal rapidly changes its waveform envelope, due to transitions between uttered sounds. In this case the algorithm cannot adapt instantaneously to the change and might fail to provide correct results. This PDA was early implemented on analogue hardware and a lot of work was done to overcome the limitations

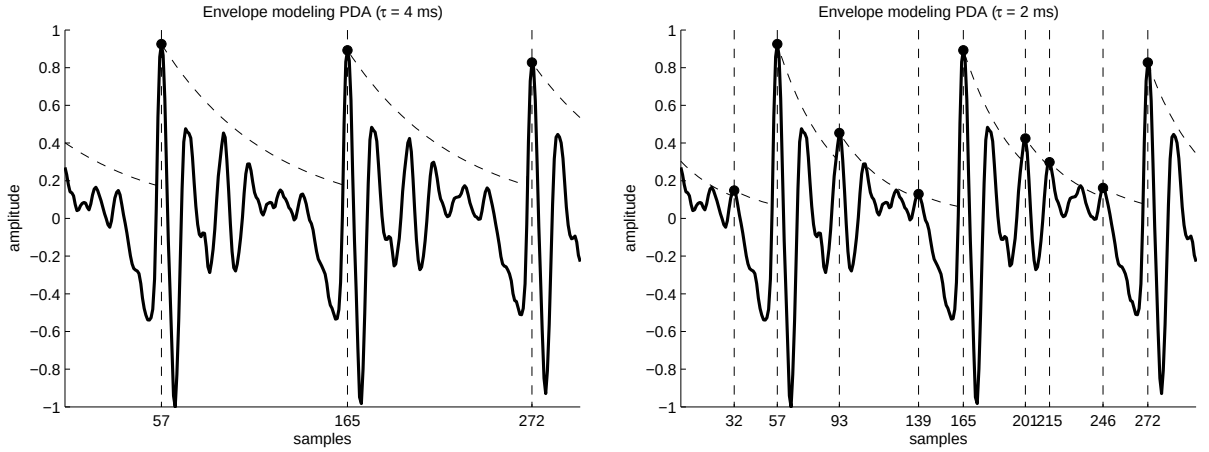

Figure 2.4: *Examples of PDAs with an envelope modeling based extractor applied to a voiced segment of speech. Determining correctly the time constant τ of the decaying envelope is crucial for the correct behaviour of the algorithm. At the left an example with a correctly estimated $\tau = 4$ ms shows period markers at sample instant 57, 165 and 272. At the right the same speech segment on which the algorithm run with $\tau = 2$ ms detecting many false period markers.*

imposed by the hardware capabilities of that time [20, 21, 3, 26, 27].

Among others, the *peak-picker* algorithm [44] was based on the envelope modeling approach and on earlier work by [41]. Being suitable for real-time applications, given its small input output delay, was also implemented as part of cochlear implant prosthesis for pattern processing hearing aids at the University College London [45].

The sequence of extremes PDA approach bases instead on an heuristic model of the temporal signal structure. Mainly, the idea is to perform data reduction at the extractor level in a iterative fashion, individuating at each iteration some anchor points from which pitch period markers are afterword derived. Maxima, minima and zero or nonzero threshold crossings are often used along with decisions and branching to design algorithm which principally base on the following steps

1. *data reduction*: eliminate samples from the incoming signal which does not belong to a chosen feature;

2. *selection*: choose (enhance) samples which provide useful information related to period delimiters and discard (attenuate) the others. One or more iterations;

3. *markers detection*: derive all possible delimiters and search among them for the sequence showing the most regular behaviour.

This class of algorithms is not always suitable for real time applications because, although they provide f_0 estimation on a period by period basis, they need to process (see step 3) segments of signal larger than one single period, in order to choose the correct delimiters sequence.

Peak detection and global correction

One of the most known algorithms belonging to this class is that based on *peak detection and global correction* [86, 87]. This algorithm was designed for a speech recognizer tool and made use of maxima and minima localization on a frame basis of 25 ms. Within each frame, first all maxima and minima where searched and tested applying a set of conditions to verify whether they could be candidates period delimiters. These conditions involved comparison with other maxima and minima, and with absolute maxima and minima, as well with other values obtained from those by linear interpolation. Also period length prediction is used to correct errors basing on the past estimated period lengths, in particular markers can be shifted, removed or inserted to adjust the final f_0 contour to be consistent. Nevertheless, as it happens to many algorithms which include pitch correction features based on the past estimated values, whenever too many errors are encountered, the global correction routine fails and can severely prejudice the correctness of future estimates [53].

Pitch chaining

Another algorithm which performs direct analysis of speech waveform is the *pitch chaining* algorithm [93]. This PDA searches, for each analyzed frame, for all possible combinations of three maxima and all minima which provides "period twins", that is, three markers defining two adjacent period values. Before next frame is processed, a tree is updated: each branch represents the concatenation of period twins computed from all previous frames which form a consistent pitch contour. Each period twins from the new combination is thus added to the proper chain, starting a new branch when coincidence in the period values is missing. The choice between the different chains is done when the longest one exceeds a preset length. At this time, the fundamental frequency for this chain is computed and output while all earlier chains are deleted. Although this algorithm works on a period by period basis, it provides averaged pitch estimation every 10ms, being thus not suitable for real-time processing.

An algorithm based on time structural analysis of the speech waveform, which does not rely on peaks position and amplitude but exploits *zero-crossing* and *excursion cycles* (EC) information, is described in [64, 65]. The peculiarity of this approach is in the way it applies data reduction to the input: only the excursion cycles are retained, which are defined as the sum of the signal amplitude values over two consecutive zero-crossing of its waveform. Considering that a period of voiced speech is dominated by the harmonic enhanced by the first formant, in the best case there will be just two EC detected per period, guaranteeing at least a number of ECs one order of magnitude less than the number of samples. Using ECs to represent the signal and applying further processing tasks, permits to reduce the computational effort significantly. To obtain the final f_0 estimate, structural knowledge of the voiced speech signal is exploited in

order to reduce the number of ECs and "isolate" the "significant" ones, which mark the beginning of each pitch period.

Mixed-feature

The time domain based algorithms basing on structural analysis described so far, all based on either regularity of the speech signal, either on its peakedness. Speech signal loses its regularity when rapid changes in fundamental frequency or sound quality occur. Also strong peaks can be missing or be not so prominent in case of nasals, back vowels or speech with falsetto excitation. In Figure 2.5 is shown the block diagram of one of the best known PDA [37] which exploits both regularity and peakedness in order to overcome the limitations above mentioned.

The algorithm proposed, set several rules to interpret sequences of (local) maxima and minima. In particular, after reducing the higher formants applying a 900Hz cutoff low-pass filter to the input signal, it generates at each processing step a series of six peaks, $M_i,\ i = 1, 2, 3, 4, 5, 6$.

As shown in Figure 2.6, the input signal is marked with label M_1 on its positive peak and with label M_4 on its following negative peak; Label M_5 registers the peak-to-valley distance between them and the same holds for label M_2 which register the valley-to-peak distance between M_4 and M_1. Labels M_3 and M_6 are relative to peak-to-previous-peak and valley-to-previous-valley measurements. Each label carries information about the time instant of the occurred event and the relative measured value. Each stream of pulses with the same label M_i is then used as input to one Primary Extractor (PE) based on the exponentially decaying technique, as explained earlier in this section. All PEs are identical and adapt the time constant of the decaying envelope basing on the previous period estimate.

Once an estimate is obtained from each of the six PEs, an evaluation procedure is performed: a 6x6 matrix is formed where each column corre-

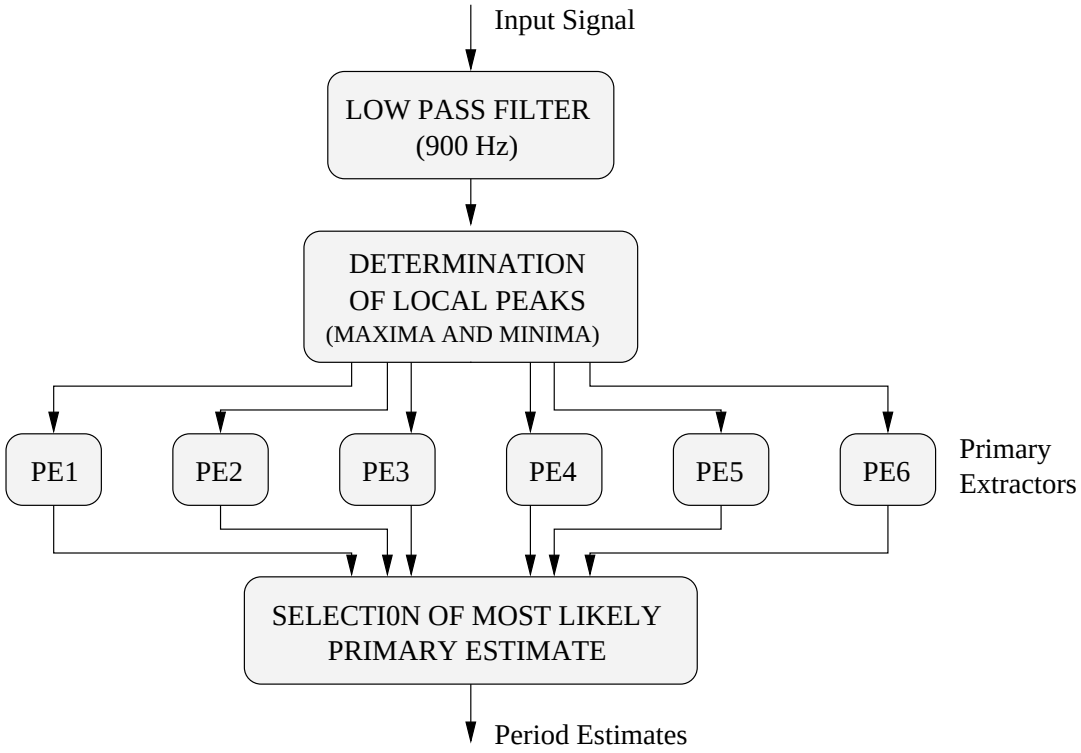

Figure 2.5: *Mixed-feature based PDA which exploits both regularity and peakedness of the incoming signal. Six Primary Extractors (PE) elaborate sequences of maxima and minima, as well as inter-peaks measurements, to produce six period values which will be selected to provide a final estimate.*

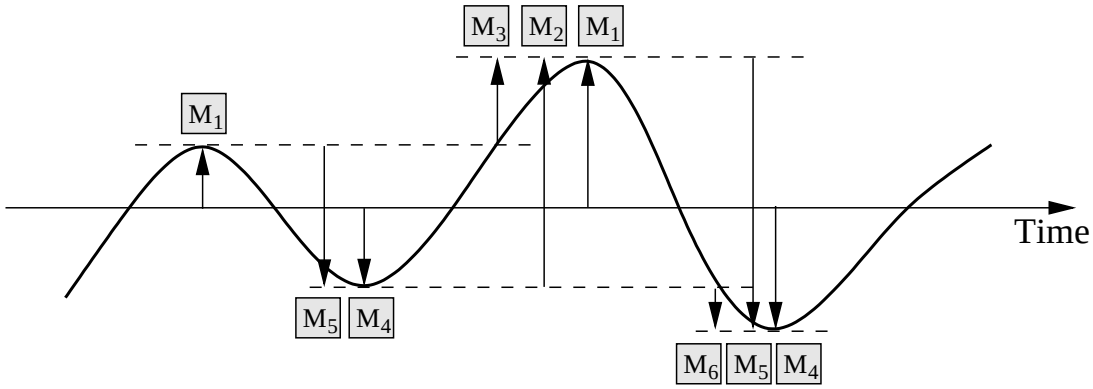

Figure 2.6: *Example of the six individual peak functions* $M_i,\ i = 1 \ldots 6$ *on a voiced speech segment. Each stream of measurements with the same label will feed a Primary Extractor (PE) which will in turn return a period estimate.*

sponds to an extractor while row one represents the direct estimates from the PEs, row two and three reports the estimates of the two previous periods, respectively; row four, five and six represent the sum of estimates from the first and second, the second and third and the first to the third rows respectively.

The final period estimate is computed as the one that has the highest degree of coincidence, evaluated using absolute difference between values, with the other values in the matrix.

The reason for including also rows from four to six, stems from the fact that all extractors are biased toward too high f_0 errors, that is, relative to the second and third harmonics still present in the signal. If this would be the case, these matrix rows will contribute to provide the correct result.

2.2.3 Structure Simplification

This category of PDAs comprises the algorithms which perform simplification of the temporal structure of the voiced speech signal. These methods can be considered as in between the fundamental frequency extraction algorithms (Section 2.2.1) and direct analysis of the temporal structure algorithms (Section 2.2.2).

The reason for this approach originated from the drawbacks of the methods reported above. The first issue is that the fundamental frequency is not always present and nonlinear filtering is not always able to reconstruct it. The second one is that, handling directly the signal temporal structure, inevitably induces the designer to introduce several heuristic-based solutions which imply loss of generality [43].

Among the PDAs based on structure simplification there are those that perform *inverse filtering* and those relying on *epoch detection*.

Inverse filtering - Linear Predictive Coding (LPC)

Inverse filtering is a technique which estimates the inverse response of the vocal tract in order to obtain the glottal excitation signal. This signal reflects the regular pulse-like variation of the air pressure generating in the larynx and results thus more suitable for f_0 estimation.

The approach bases on the source-filter model which will be described more in-depth in Chapter 3. According to this model, the voiced speech signal $x(n)$ can be seen as the result of the convolution of the glottal stream of pulses $s(n)$ with the vocal tract impulse response $h(n)$:

$$x(n) = s(n) * h(n), \qquad X(z) = S(z) \cdot H(z), \tag{2.1}$$

where the right-hand expression represents the source-filter model in the frequency domain[2]. According to this expression, if the vocal tract transfer function $H(z)$ is known, $S(z)$ is easily obtained after multiplication of both sides by $1/H(z)$.

As shown in [24], the vocal tract can be regarded as a lossless acoustic tube actuating on the excitation signal as a resonator. This is particularly true for voiced sounds for which the vocal tract transfer function $H(z)$ can be approximated as an all-pole filter. The consequence of this hypothesis is that the inverse $1/H(z)$ can be modeled as an all-zero filter by means of a non-recursive state equation as follows:

$$s(n) = d_0 x(n) + d_1 x(n-1) + \ldots + d_p x(n-p). \tag{2.2}$$

The inverse transfer function $H^{-1}(z) = 1/H(z)$ can thus be written as:

[2] $X(z)$, $S(z)$ and $H(z)$ are the z-transforms of the discrete signals $x(n)$, $s(n)$ and $h(n)$, respectively. If the complex variable z is set to $e^{j2\pi f}$, the frequency spectrum is obtained.

$$H^{-1}(z) = \sum_{i=0}^{p} d_i z^{-i}, \tag{2.3}$$

and its coefficients $d_i,\ i = 1, \ldots, p$, could be determined by a complete formant analysis, by means of methods like *peak picking* or *analysis by synthesis* proposed in [28]. However, these methods are not eligible for this task being too complex and requiring thus too much effort to be applied.

A common and very popular approach to estimate the coefficients in Equation 2.3 is Linear Predictive Coding (LPC). Linear prediction states that it is possible to predict the sample $x(n)$ from the values of the previous samples to within an additive error signal (or residual signal) $e(n)$:

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + \ldots + a_p x(n-p) + e(n), \tag{2.4}$$

where $x(n)$ represents the speech signal, a_i the filter coefficients and $e(n)$ the error signal. Equation 2.4 is a purely recursive digital filter, that is, an all-pole model of the vocal tract transfer function.

The vocal tract parameters vary during the speech process thus implying that coefficients a_i must be time variant as well, in order to follow their variations. This implies that parameter estimation must be performed on a short-term based analysis which is usually $10-30\ ms$ length.

A typical approach to determine the predictor coefficients is to minimize the energy of the error signal $e(n)$ within a given frame. Solving for $e(n)$ gives:

$$e(n) = x(n) - \hat{x}(n), \tag{2.5}$$

defining with

$$\hat{x}(n) = a_1 x(n-1) + a_2 x(n-2) + \ldots + a_p x(n-p), \tag{2.6}$$

the predicted sample. Equations 2.5 and 2.6 represent non-recursive digital filter and the first one has the same structure of Equation 2.2.

From linear filter theory, in case of a stationary signal, the predictor would be able to estimate perfectly and the error will be zero. Speech signal can be considered stationary just in between glottal pulse excitation instants, consisting of a sum of decaying sinusoids.

LPC analysis assumes the source excitation signal to be impulsive and provides a solution for coefficients a_i so that the error function $e(n)$ can be considered, within a certain approximation and considering the purpose of the analysis[3], the glottal pulse function $s(n)$.

To obtain this, the mean square of $e(n)$ is expressed as a function of the predictor coefficients and a set of linear equations is solved by exploiting autocorrelation and covariance functions [55, 57].

In Figure 2.7 an example of LPC analysis is shown. On the upper left panel three periods of a vowel sound are represented and the corresponding magnitude spectrum is plotted at the right. The latter shows clearly the harmonic structure of the signal as a sequence of narrow peaks. Applying the LPC analysis using 18 coefficients a_i, the inverse transfer function $H^{-1}(z)$ is estimated and its inverse, that is $H(z)$, is shown in the right middle panel, with formants label $F_i,\ i = 1, \ldots, 4$. Once $H^{-1}(z)$ is know, it is possible to obtain the glottal pulse transfer function $S(z) = X(z) \cdot H^{-1}(z)$ which is reported in the bottom panel at the right. The formant structure has been almost removed while the harmonic structure has been preserved. In the time domain, the residual signal $e(n)$ shows a series of peaks corresponding to the excitation signal $s(n)$, and is shown in the bottom panel at the left.

LPC analysis is not free from drawbacks: the minimization operation

[3]The actual excitation signal is not preserved by LPC analysis which cannot really distinguish between components of the vocal tract and those belonging to the glottal pulse and retains the latter just to an impulsive extent.

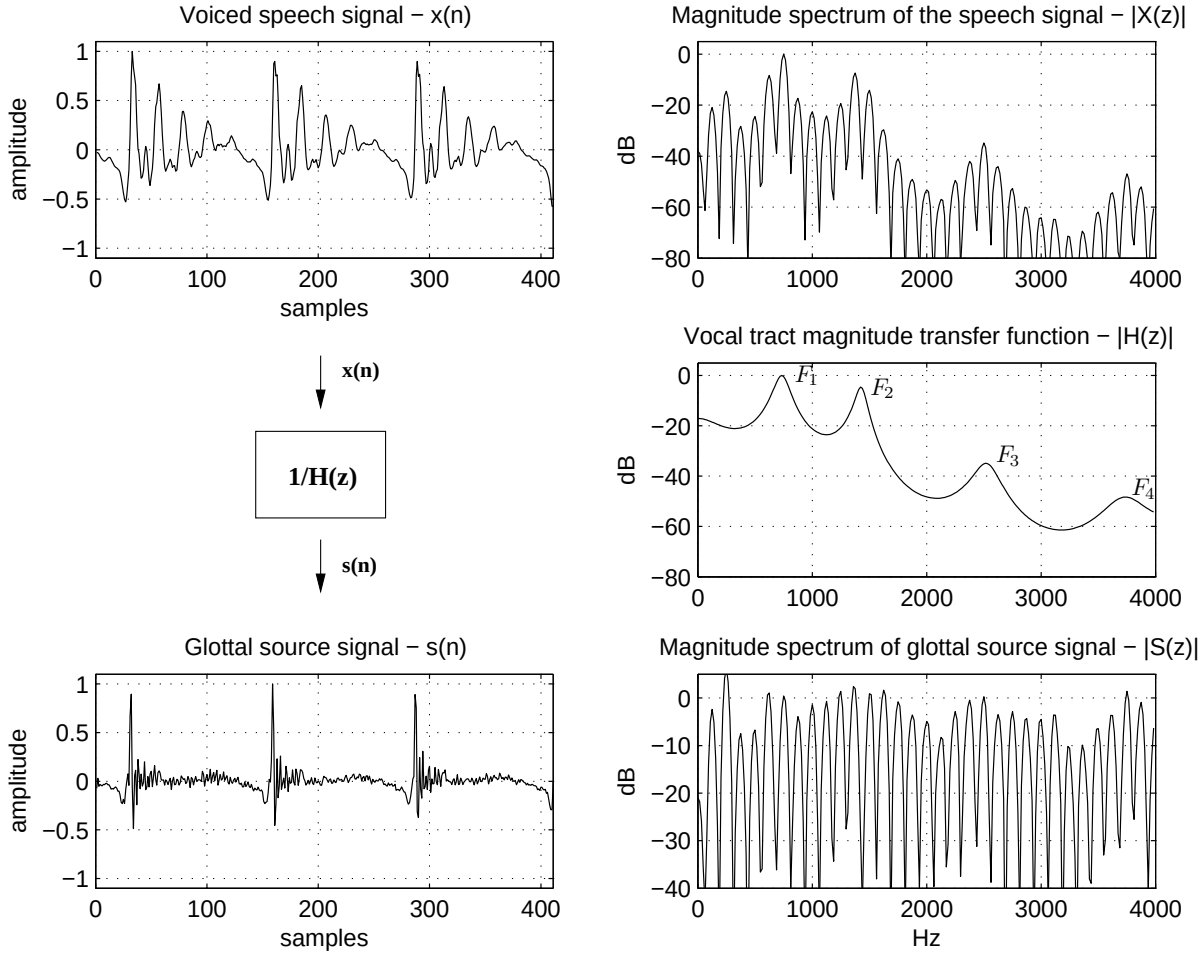

Figure 2.7: *Example of LPC analysis. Upper left panel: three periods of a vowel sound; upper right panel: corresponding magnitude spectrum showing the signal harmonic structure as a sequence of narrow peaks; middle right panel: vocal tract transfer function $H(z)$ estimated by LPC analysis (18 coefficients). Formants are labeled with F_i, $i = 1, \ldots, 4$; bottom right panel: glottal pulse transfer function obtained as $S(z) = X(z)/H(z)$. The formant structure has been almost removed while the harmonic structure has been preserved; bottom left panel: residual signal $e(n)$ in the time domain. The excitation signal $s(n)$ can be approximated with the series of peaks shown.*

applied to the error signal not always preserve the excitation signal [35], additionally, when the first formant frequency is the same of the fundamental frequency f_0, removing the formant effect tend to cancel the latter from the residual signal, frustrating any further attempt to detect the cor-

rect f_0. However, in case the residual signal is successfully, direct analysis of the its temporal (Section 2.2.2) structure can be applied, as done by several time domain algorithm [8, 106].

Epoch detection

Algorithms which base on epoch detection, aim to detect the events or "epochs" related with each glottal closure instant. Considering this instant as a generation of a pulse-like pressure wave, all frequencies (and all resonances) of the vocal tract transfer function can be considered to be excited at the same moment.

Phase coherence among frequencies is an important requisite among these class of algorithms, since they base on detecting the epoch instants, synchronously among the outputs of a filter bank.

One of the first algorithm [101, 102, 112] designed to perform epoch detection, was constituted by a bandpass filter bank, whose outputs were full-wave rectified and smoothed in order to perform amplitude demodulation. Each smoothed output resulted thus in an envelope-like function with period equal the fundamental frequency, synchronous with all other filter bank outputs and independent of the particular bandpass filter. Summing up all these signals resulted in a waveform suitable for direct analysis of its temporal (Section 2.2.2) structure.

In Figure 2.8 is shown an example of epoch detection on voiced speech segment. The top left graph shows the input speech signal, and each of the remaining signals on the left side labeled ch1, ch2, ..., is the output of a bandpass filter with center frequency shown beside the channel label. At the right side of the figure, are the rectified and smoothed bandpass filter outputs and the bottom right graph shows their sum. This class of algorithms can correctly detect the epoch even if the fundamental harmonic is absent, or even in the case its value coincides with one of the formant

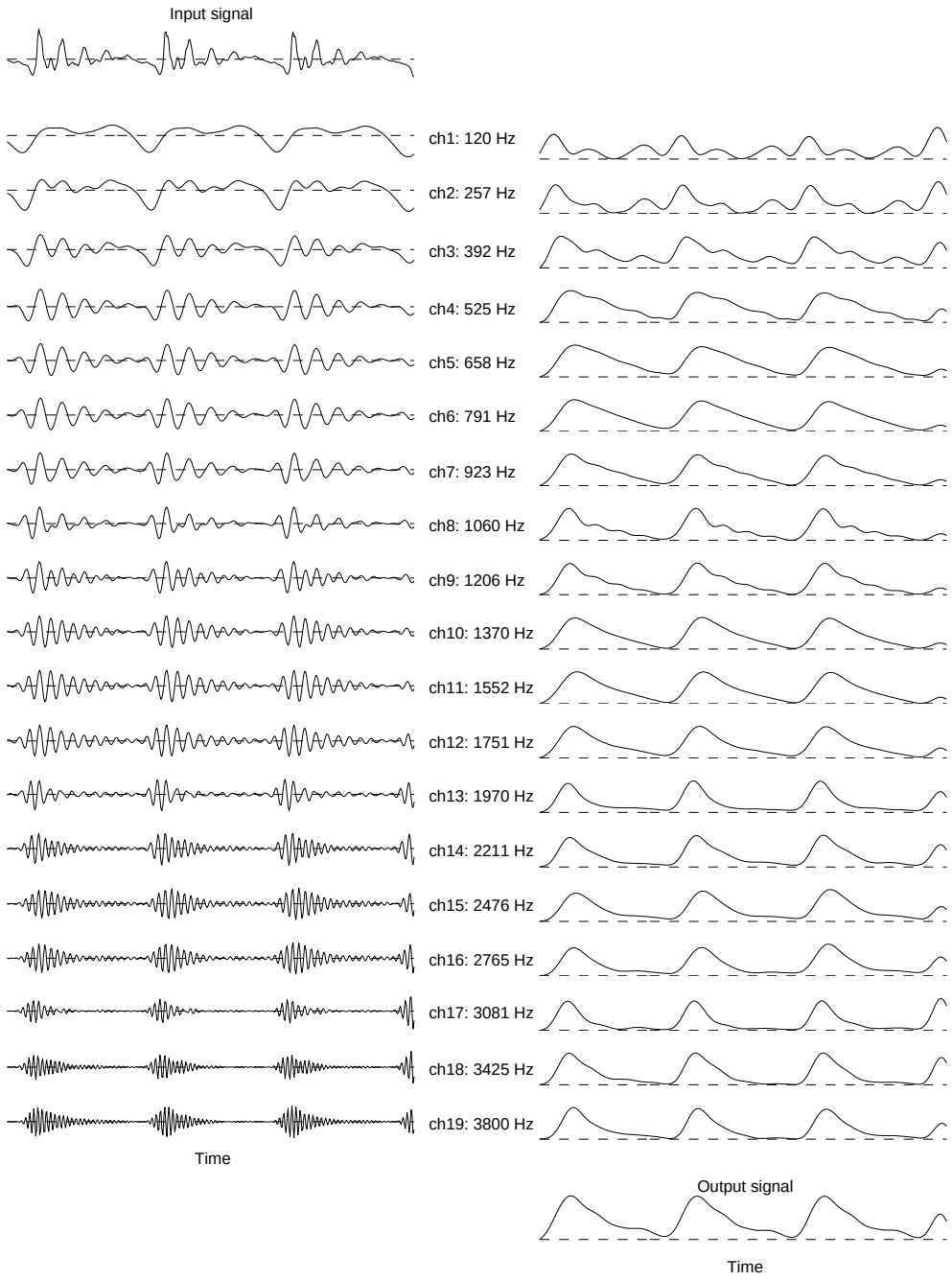

Figure 2.8: *Example of epoch detection on a voiced speech segment. Top left graph shows a voiced speech segment. Below it, the bandpass filter output for each channel ch1, ch2, ... and center frequency shown beside. At the right side of the figure, are the rectified and smoothed bandpass filter outputs while the bottom right graph shows their sum.*

frequencies. Instead, the limitations of this approach become visible whenever the excitation signal shows weak discontinuities, such as in the case of falsetto voice and voiced fricatives. Additionally, this approach is the least suitable for speech signals recorded in a reverberant environment. In fact, the hypothesis of phase coherence among frequencies of the excitation signal turns out to be false in this case.

2.2.4 Multichannel Analysis

The *Multi-channel analyzers* term refers to PDAs which base on different type of parallel processing of the incoming speech signal. Among these algorithm class, there are many which were previously described, or cited. Mainly, there are three types of multi-channel analyzers [43], each one based on one of the following principle:

- *Main channel and auxiliary channel principle*: this setup takes into account the use of an auxiliary channel with the main purpose of adapting the principal channel operation. An example of this procedure is represented by the open-loop tracking filter system, in which the auxiliary channel is used to derive some information from the input signal, that will be used to tune the period estimator;
- *The sub-range principle*: this type of PDAs base on several identical (or similar) preprocessors tuned to operate on different frequency subranges. Then one of their output is selected and used as input for the pitch extractor algorithm;
- *The multi-feature principle*: in this case there are several PDAs which perform parallel processing. Each PDA operates independently from the others and provides different signal features or, alternatively, all PDAs compute the same set of features obtained with different techniques. A common stage for all PDAs can be present for preprocessing.

In this setup a data fusion technique has to be used to select from all channel outputs or combine them to provide a single pitch estimate.

Whenever selection among channel results must be done, the way of how to detect the channel providing the correct estimate is not trivial. A basic decision rule can be to choose the channel providing the longest period estimate, or the one that showed the highest number of occurrences of a certain pitch value. Another issue, related with multichannel analysis is related with the phase of period markers. Different channels can provide period markers which might differ in phase, that is, glottal cycle begin and end may be detected differently by each channel. This is explained considering that different signal features are involved in each channel. In case the particular application does not require pitch phase information, an average can be computed among all phases.

2.3 Short Term Analysis Pitch Determination

The short term analysis based PDAs differ from the time domain based algorithms in that pitch estimation is performed on a short segment of the input speech signal. This implies that the estimated fundamental frequency does not refer any more to a specific time instant (or glottal cycle) but may include several pitch periods representing their average.

Being not a period-by-period processing technique, the short term analysis does not estimate glottal cycle phase information. In case this information is not needed, it represents an advantage since these algorithms are more robust to phase distortion, which can severely affect the performance of time domain techniques.

Also this class of algorithms turns out to be more robust to noise or signal corruptions. This because, for each estimate, a signal segment longer

than a single period is considered and, as long as signal corruptions does not affect the whole set of data, still the correct information is recoverable.

By a computational point of view, these algorithms need more processing time respect their time domain counterpart. This fact, that could have represented an issue in the past, nowadays is not a problem any more, considering the advances in modern digital computer technology.

Considering that the speech signal, is a non-stationary signal, its characteristics, as for example periodicity, change as a function of time. Using a short term based approach, instantaneous values of pitch epochs are not generally estimated and careful must be taken not to provide a period estimation which does not reflect important local pitch variations.

To control this process, windows are employed to select speech segments to be processed at each step. Given the sampled speech signal $x(n)$, its short-term segment $x_s(n,q)$ is obtained by multiplying it with a window function $w(n)$ as follows:

$$x_s(n,q) = x(n) \cdot w(n-q), \qquad w(n) = \begin{cases} \neq 0, \ 0 \leq n \leq N \\ = 0, \ \text{otherwise} \end{cases} \tag{2.7}$$

The windowed signal $x_s(n,q)$ in equation 2.7, is given by the original signal values $x(q), \ldots, x(q+N)$, weighted by the window values $w(n)$. The window function $w(n)$ can be any time limited function and its choice depends on the application and on the characteristics of signal $x_s(n,q)$, required for further processing. The most common used window functions are listed in Table 2.1 and each one of them represents a different trade-off between time and frequency resolution capabilities.

In fact, multiplying the input signal by a window function in the time domain, in the frequency domain turns out to be a convolution operation of the frequency response of the window with the signal spectrum.

Rectangular	Hanning	Hamming	
$\begin{cases} 1, \\ 0, \end{cases}$	$\begin{cases} 0.5-0.5\cos(2\pi n/N), \\ 0, \end{cases}$	$\begin{cases} 0.54-0.46\cos(2\pi n/N), \\ 0, \end{cases}$	$0 \leq n \leq N$ otherwise

Bartlett
$\begin{cases} 2n/N, & 0 \leq n \leq N/2, \\ 2-2n/N, & N/2 < n \leq N, \\ 0, & \text{otherwise} \end{cases}$

Blackman
$\begin{cases} 0.42-0.5\cos(2\pi n/N)+0.08\cos(4\pi n/N), & 0 \leq n \leq N \\ 0, & \text{otherwise} \end{cases}$

Table 2.1: Some commonly used windows of length $N+1$ samples (assuming N even) symmetric respect to sample $N/2$ [78].

In Figure 2.9 are shown the absolute values of the Fourier transforms expressed in decibels of the windows listed in Table 2.1. The main window characteristics in the frequency domain are the resolution capability, the peak-sidelobe level and side lobe role-off. Resolution refers to the capability to distinguish different tones and is inversely proportional to the main lobe width (plotted in red in figure). The peak-sidelobe level refers to the maximum response outside the main lobe and determines whether signals with small peaks in the frequency domain are hidden by nearby stronger ones. The side lobe roll-off is measured as the side lobe decay per decade[4] and is trade-off with the peak-sidelobe level [42, 78, 85].

Each set of samples involved at each processing step, is referred to as frame. Successive frames can overlap to a certain extent, so that the inter-

[4]A frequency decade is a 10-fold increase or decrease in frequency.

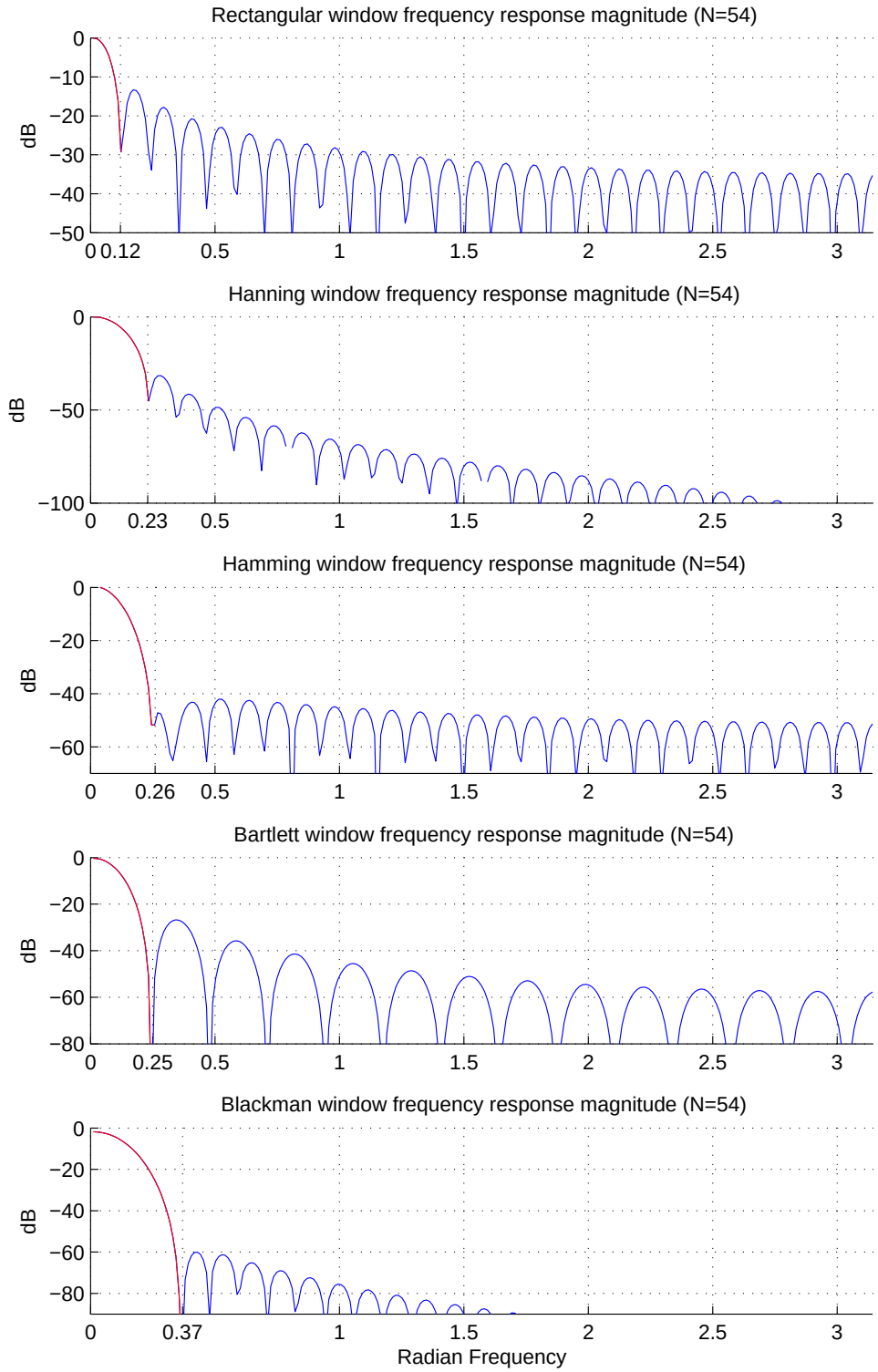

Figure 2.9: *Fourier transforms (log magnitude) of windows listed in Table 2.1.*

val between successive estimates can be set shorter than the frame length. The window parameter N is important for PDAs based on short-term analysis. It has to be large enough to include a sufficient number of signal samples for correct f_0 estimation, and small enough to capture fundamental frequency variations within short intervals. Usually a value between 20 ms and 50 ms is used depending on the application. In case a value of 50 Hz is set[5] for the minimum fundamental frequency allowed, this would imply that from one to two and a half periods respectively will be comprised within one frame. This concept is strongly related with the PDA performance in case of signal perturbations. In fact, when a local perturbation occurs in the speech signal, due to noise or other causes, the behaviour of the PDA depends on the extent of the irregularity duration over the considered frame. In case the analysis frame is too short so that it contains only or mainly perturbed signal, the estimate will be wrong. However, in case the frame length is such that the contribution of perturbed signal portions is small, the algorithm will still be capable of giving a correct estimate. It has to be recalled though, that the speech signal can only be regarded as quasi-stationary, implying that the pitch period values are not constant within a given frame. Consequently, the estimate will be an average of several consecutive signal periods, becoming less accurate as the analysis frame gets longer.

The different processing steps involved in short-term analysis are summarized in the scheme of Figure 2.10. As shown, the input signal can undergo a pre-processing step where low-pass filtering, centre clipping or inverse filtering can be applied at this stage to reduce the signal temporal complexity. After this step, frame division of the incoming signal takes place and the specified short-term transformation is applied to each frame. The output of this process is generally a signal with a peak(s) whose posi-

[5]Generally the fundamental frequency of speech in adult humans is in the range of about $50-500$ Hz.

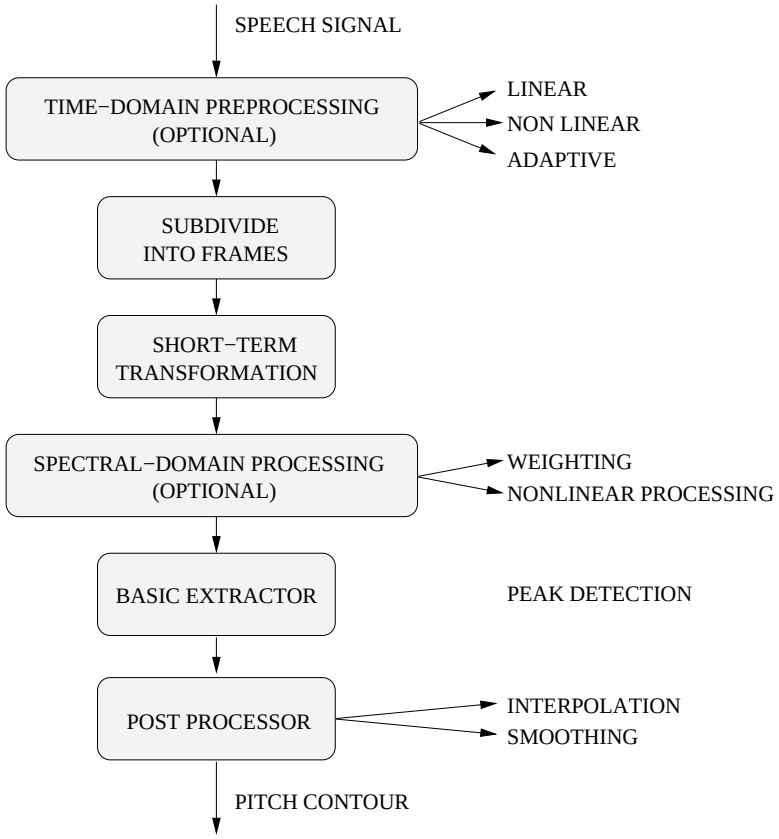

Figure 2.10: *Block diagram of a sample short-term analysis PDA.*

tion(s) and amplitude(s) is related with the fundamental frequency or its period and with the degree of periodicity respectively. The purpose of the next block is to optionally apply spectral-domain processing. Depending on the application, it may be necessary, for example, to apply weighting or to compute absolute of complex spectra values. The basic extractor block is almost always a peak detector. The peak amplitude is generally compared with a preset threshold to perform voiced/unvoiced decision and its position provides the fundamental frequency or period value. The final processing block is the post-processor which is responsible for improving the estimate resolution applying peak interpolation or for correcting errors applying smoothing techniques.

By a computational point of view, the major complexity of these PDA is

in the short-term transformation block. As shown in [43], the input/output relation of this block can be written as

$$\mathbf{X} = \mathbf{W} \cdot \mathbf{x} \tag{2.8}$$

where $\mathbf{x}$ is the vector containing all samples considered in the frame being processed, $\mathbf{W}$ is the transformation matrix and $\mathbf{X}$ is the output vector, or short-term spectrum. When this transformation is used by means of a direct implementation, the computation complexity will increase with the square of the length of vector $\mathbf{x}$. Therefore, to keep complexity low, it is important to limit the frame length. However, when spectral transformation are involved, the frequency resolution achieved increase proportionally to the number of signal samples used. To fulfill both requirements, a common devised solution is to reduce the computational complexity of transformation $\mathbf{W}$ and to perform interpolation on the output vector $\mathbf{X}$ values. For example, in case Fourier transform is applied, the Fast Fourier Transform (FFT) algorithm [15] can be used with a computation complexity proportional to the logarithm of the length of the input data. Another solution was represented by the Average Magnitude Difference Function (AMDF) algorithm [89], which based on summations instead of the more computationally expensive multiplications.

As shown in Figure 2.11, short-term analysis comprises *lag-domain* analysis, such as autocorrelation techniques, where the lag variable, also named "quefrency", refers to the pitch period length, expressed in samples; *frequency domain* analysis, which operates in the frequency domain after transformation of the input signal and *maximum-likelihood* analysis.

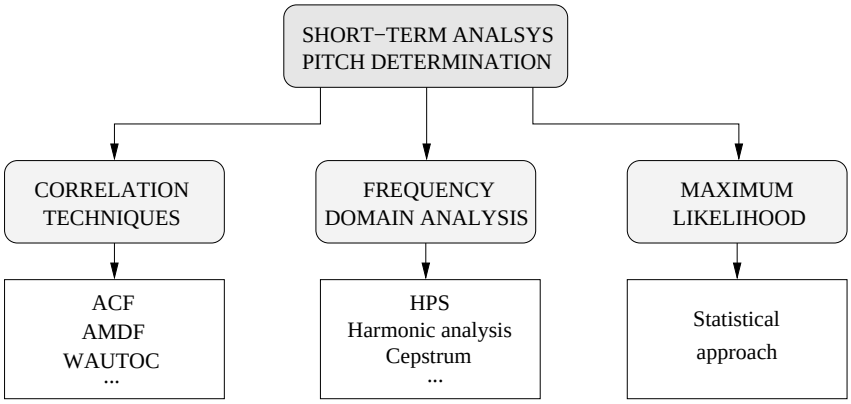

Figure 2.11: *Sub-classification of short-term fundamental frequency estimation algorithms [43].*

2.3.1 Lag-domain analysis

Autocorrelation Function (ACF)

Correlation is a measure of similarity between two signals and was one of the earliest technique for pitch estimation among PDAs based on short-term analysis. When the input signal is correlated with itself, that is, when autocorrelation is computed, possible signal self-similarities are pointed out.

The autocorrelation function (ACF) $r(\tau)$ applied to the discrete signal $x(n)$ is defined as:

$$r(\tau) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)x(n+\tau), \tag{2.9}$$

where the parameter N determines the number of samples of $x(n)$ involved in the operation and the factor $2N+1$ at the denominator normalize the result. The variable τ set the lag between the signal and a delayed version of itself. One important properties of Equation 2.9 is that when the signal $x(n)$ is periodic of period T_0 samples, the autocorrelation is also periodic with the same period:

$$x(n+kT_0) = x(n), \quad \forall k \in \mathbb{Z} \quad \Longrightarrow \quad r(\tau+kT_0) = r(\tau), \quad \forall k \in \mathbb{Z}. \tag{2.10}$$

Another characteristic of the autocorrelation function is that it is an even function, that is, $r(-\tau) = r(\tau)$ and that its maximum value is found at lag position $\tau = 0$ in which case $r(0)$ represents the signal power and holds:

$$r(\tau) \leq r(0), \quad \forall \tau \in \mathbb{Z}. \tag{2.11}$$

From Equations 2.10 and 2.11 it is possible to state that the autocorrelation function $r(\tau)$, applied to a periodic signal $x(n)$, will show a series of peaks at positions $\tau = kT_0$:

$$r(kT_0) = r(0), \quad \forall k \in \mathbb{Z}. \tag{2.12}$$

When $x(n)$ is not stationary but can be regarded just quasi-stationary over short segments, as is the case of voiced speech, Equation 2.9 must be modified so that it can be used for short-term processing. In addition, due to possible changes in the dynamic of the speech signal, relation 2.11 may not hold any more. Given this, a new definition of autocorrelation must be devised to take into account the speech signal characteristics. In [82] the following autocorrelation function is proposed:

$$r(\tau, q) = \frac{1}{N} \sum_{n=0}^{N-1} [x(q+n)\ w(n)] \cdot [x(q+n+\tau)\ w(n+\tau)], \tag{2.13}$$

where q is the starting sample and $w(n)$ is a window function which is null for values of n outside the interval $0 \leq n \leq N-1$.

Figure 2.12 shows the autocorrelation function applied to a segment of voiced speech. The largest non-zero offset peak is found at lag $\tau = 197$ samples which, considering the sampling frequency used, provides an estimated period $T_0 \approx 100\ Hz$.

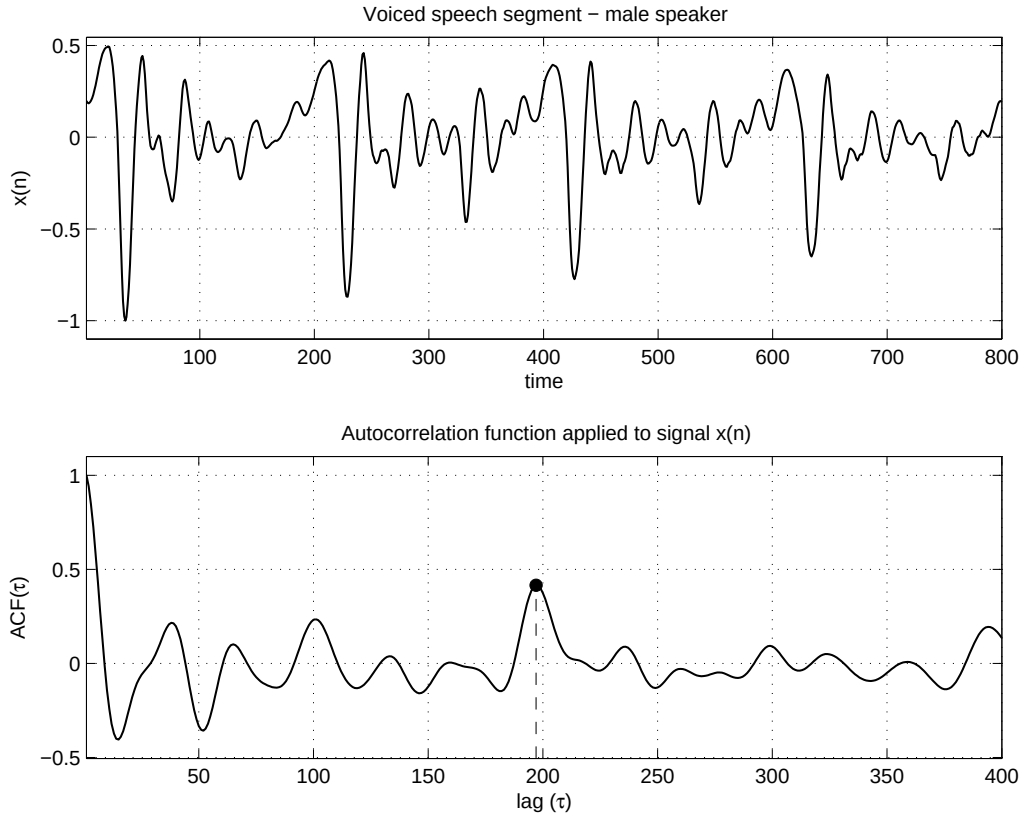

Figure 2.12: *Example of autocorrelation function (ACF) applied to a segment of voiced speech from a male speaker. The largest non-zero offset peak is found at lag* $\tau = 197$.

One of the major flaws of the autocorrelation function is that it is very sensitive to formant positions. As pointed out in [97], it is very likely to happen that the estimated period would be $T_0 \pm T_F$, where T_0 is the actual fundamental period and T_F is the period of a major formant. To overcome this limitation, spectral smoothing techniques were applied to the signal before it was processed by the ACF.

One way to reduce or suppress the formant effect, is to apply a "spectral

flattener" technique, such as those based on a instantaneous non-linear function [82, 103]. One of these non-linear function is the compressed *centre clipping* (*clc*) function whose input-output relation given by

$$y(n) = clc\,[x(n)] = \begin{cases} x(n) - C_L, & x(n) \geq C_L \\ 0, & |x(n)| < C_L \\ x(n) + C_L, & x(n) \leq -C_L \end{cases} \tag{2.14}$$

Equation 2.14 and its application to a segment of voiced speech is shown in Figure 2.13. The effect of preprocessing the signal in this way is to remove the prominent peaks in the signal spectrum due to formant resonances.

All the signal values whose absolute value falls below a pre-set threshold C_L are set to zero while the remaining signal values are compressed subtracting the constant C_L as shown in figure. The result is that many smaller signal peaks due to higher harmonics and formants are removed. The autocorrelation function applied to the compressed centre clipped signal is shown at the bottom of Figure 2.13. The voiced speech signal used is the same which was previously used in the example of Figure 2.12. This time the estimated lag resulted $\tau = 195$ samples and can be regarded as a more precise estimation, since it bases on a signal where just the peaks due glottal closure instants are preserved. In addition, comparing the autocorrelation functions of Figures 2.12 and 2.13, it can be noted that the latter contains fewer and more prominent peaks, most of them related with the pitch period. The scheme based on compressed centre clipping function followed by ACF, was shown [22, 84] to perform much better in speech pitch estimation. However, this approach need to estimate continuously the signal amplitude to adapt the threshold C_L.

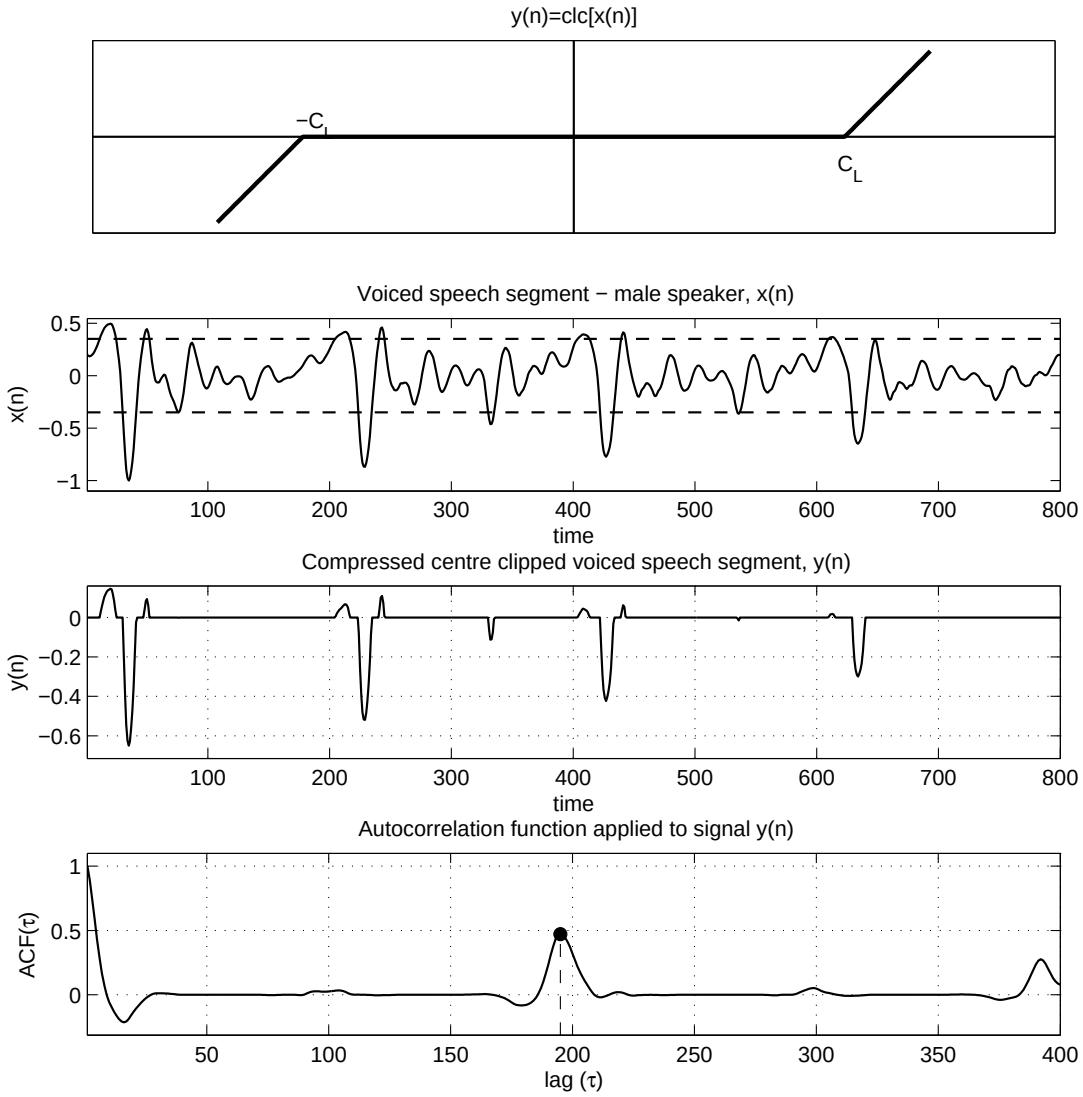

Figure 2.13: *Top panel: compressed centre clipping function; Second panel: voiced speech segment and centre clipping threshold set to* $C_L = 0.35$*; Third panel: output of the flattener function* $y(n) = clc[x(n)]$*; Bottom panel: example of autocorrelation function (ACF) computed on compressed centre clipping* $y(n)$ *function. The largest non-zero offset peak is found at lag* $\tau = 195$.

An alternative method for spectral flattening was proposed in [103] and bases on a set of bandpass filters. Each filter bandwidth was set to 100 Hz and its output was normalized by the short-term estimated signal envelope. All contributes were finally added together to provide a signal with

flat spectrum.

Another method which measures the similarity between two segments x and y of a speech signal to estimate their common periodicity, is that reported in [62]. The main difference with the autocorrelation method is that the two segments x and y to be compared, are chosen to be exactly adjacent and non overlapping. Each segment length τ is increased one sample at each step and a first gross pitch estimate is carried out finding the value for τ that maximizes the cross-correlation coefficient

$$\rho_\tau(x, y) = \frac{(x, y)_\tau}{|x|_\tau |y|_\tau}, \tag{2.15}$$

where $(x, y)_\tau$ is the inner product of the two segments x and y taken as if they were vectors of length τ, and the normalization factors $|x|_\tau$ and $|y|_\tau$, represent the energy of each segment. This method provides a first pitch period estimate T_0 which has a maximum resolution limited by the sampling frequency, that is, it is an exact multiple of the sampling period. To estimate the pitch period with "infinite resolution", as reported by the authors, linear interpolation is applied to the second segment y so that it perfectly matches the first segment x.

The algorithm, tested on synthetic as well as on real speech data, is reported to perform very well, also by the accuracy point of view. Still octave errors occur but principally during voiced/unvoiced transitions.

The Simplified Inverse Filter Transformation (SIFT) algorithm

As shown in the previous section, applying a spectral flattener to a speech signal before computing its autocorrelation, provides better results since pitch estimates are not biased by formants positions. The scheme presented in Figure 2.14, describes the Simplified Inverse Filter Transforma-

tion (SIFT) algorithm which base on inverse filtering in order to remove the formant effects from the speech signal [56].

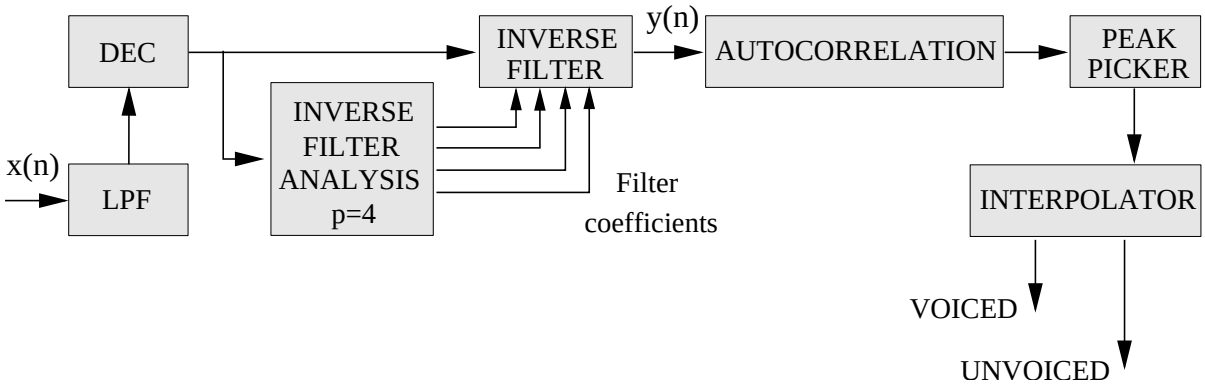

Figure 2.14: *Block diagram of the SIFT algorithm. The input signal is low-pass filtered and decimated and then processed applying the LPC analysis to obtain the excitation source signal. Autocorrelation is thus applied to obtain the pitch period estimation and interpolation is used to recover the original resolution.*

The first processing step is a low-pass filter which filters out all signal information with frequencies above 900 Hz. After this, it is possible to apply down-sampling to obtain a signal with sampling frequency of 2 kHz. This frequency range was proved to include all necessary information for pitch estimation and permitted to reduce the computation load. Next step applies the inverse filtering technique by means of LPC analysis as described in Section 2.2.3. The order of the LPC analysis is set to four, since the frequency range $0 \div 1$ kHz generally includes just two formants. The estimated coefficients are then used to drive a filter which approximates the inverse vocal tract transfer function and whose output $y(n)$ represents the glottal excitation source. Autocorrelation is then applied to $y(n)$ to estimate its periodicity. Since at this point the frequency resolution is low (2 kHz), interpolation around the autocorrelation peak found is necessary to provide a more precise estimate.

This scheme, compared to the autocorrelation approach, proved to be

more robust to formant effects and provided better results. Also it can estimate voiced/unvoiced activity, since the peaks of the autocorrelation function, when applied to the estimated source excitation signal, better reveal the degree of periodicity of the analyzed signal. The main limitations associated with this algorithm instead, are those associated with LPC analysis, such as the cancellation of the excitation signal whenever a formant position coincides with the fundamental frequency.

Average Magnitude Difference Function (AMDF)

The Average Magnitude Difference Function (AMDF), as the Autocorrelation function described earlier, is a function that measures the degree of similarity between two signals. In case a speech signal and its delayed version are used as inputs, the AMDF reveals the its possible periodicity.

$$\mathrm{AMDF}(\tau, q) = \frac{1}{N} \sum_{n=q}^{q+N-1} |x(n) - x(n+\tau)| \tag{2.16}$$

Equation 2.16 was originally presented in [63] and, a few years later, also in [89]. Similarly to the ACF, AMDF compares two segments of signal $x(n)$ which are delayed of τ samples respect each other. In case a voiced speech signal is analyzed and τ equals its fundamental period T_0, the function exhibits a minimum. The AMDF bases on summations for the computation and it is thus faster respect to the ACF by a computational point of view. Nevertheless, it is more sensitive to changes in signal amplitude [43], being thus more prone to pitch estimation error.

Figure 2.15 shows the AMDF applied to a segment of voiced speech from a male speaker. The minimum of the function is found at $\tau = 198$ samples, in accordance with the actual value of the signal pitch period.

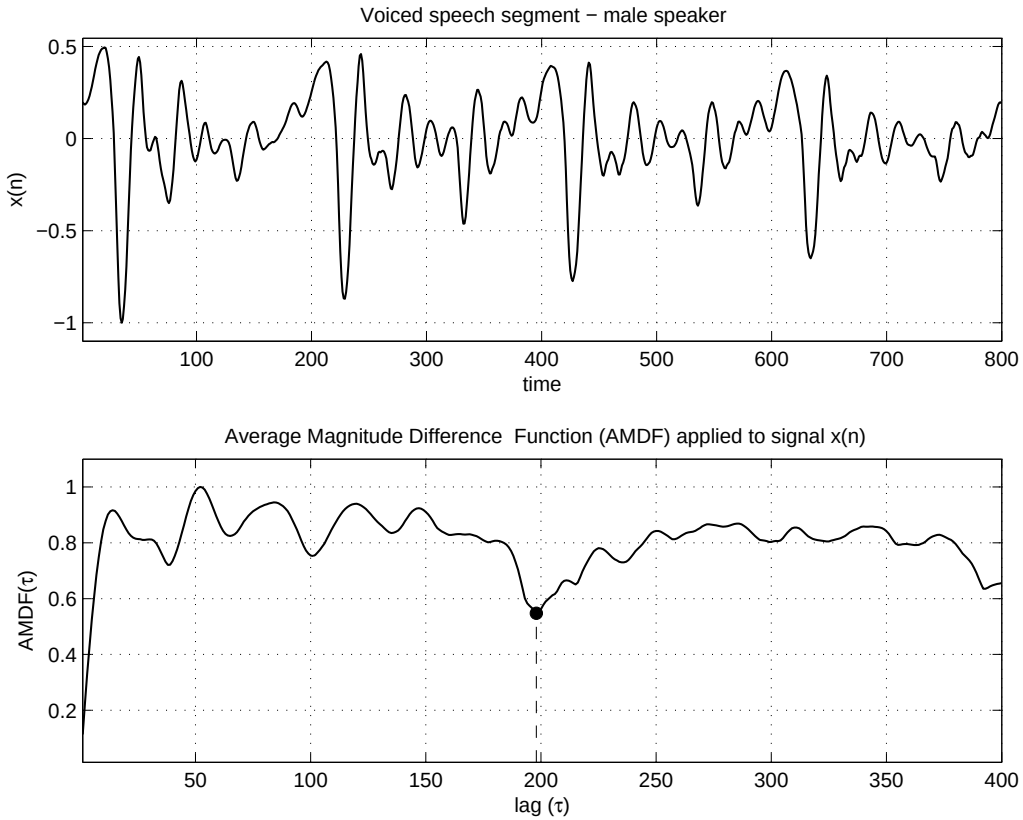

Figure 2.15: *Example of Average Magnitude Difference Function (AMDF) applied to a segment of voiced speech from a male speaker. The largest negative non-zero offset peak is found at lag* $\tau = 198$.

Another solution based on a distance function is presented in [70, 91, 92] in which the AMDF is a particular case of a more general distance function. Also in [69] a generalized distance function is used and dynamic programming is implemented for error correction and estimate refinement. In this case however, a whole set of estimates is needed before applying error correction, thus making this approach not suitable for real time processing.

Weighted autocorrelation (WAUTOC)

The ACF and AMDF exhibit similar characteristics: while the Autocorrelation Function produces a peak in correspondence of the pitch period

T_0, the Average Magnitude Difference Function produces a notch in correspondence to the same lag value.

To exploit the common behaviour of both functions, the Weighted Autocorrelation (WAUTOC) function [98] weights the ACF values by those provided by the AMDF for each value of the lag variable τ. The result is that the peak generated by the numerator is strengthened by the notch that the denominator (AMDF) produces at $\tau = T_0$.

$$\text{wautoc}(\tau, q) = \frac{\sum_{n=0}^{N-1} [x(q+n)\ w(n)] \cdot [x(q+n+\tau)\ w(n+\tau)]}{\epsilon + \sum_{n=q}^{q+N-1} |x(n) - x(n+\tau)|} \tag{2.17}$$

The Equation 2.17 is the ratio of Equations 2.13 and 2.16, where the parameters q and N indicate the starting point of signal $x(n)$ and the number of samples involved for the computation, respectively. The term ϵ in the denominator is necessary to avoid division by zero in case the summation of the AMDF resulted null.

An example of the WAUTOC function applied to a segment of voiced speech is shown in Figure 2.16. The estimated pitch period resulted $\tau = 198$ samples in accordance with the estimates provided by the ACF and AMDF individually and shown in Figures 2.12 and 2.15.

This method resulted more robust to noise conditions compared to the previous described methods. In fact, the signal components which belong to the noise source, produce a different effect in the numerator and denominator of Equation 2.17, while the periodic signal components, proceeding from the voice source, show a common behaviour, exploited by the wautoc function.

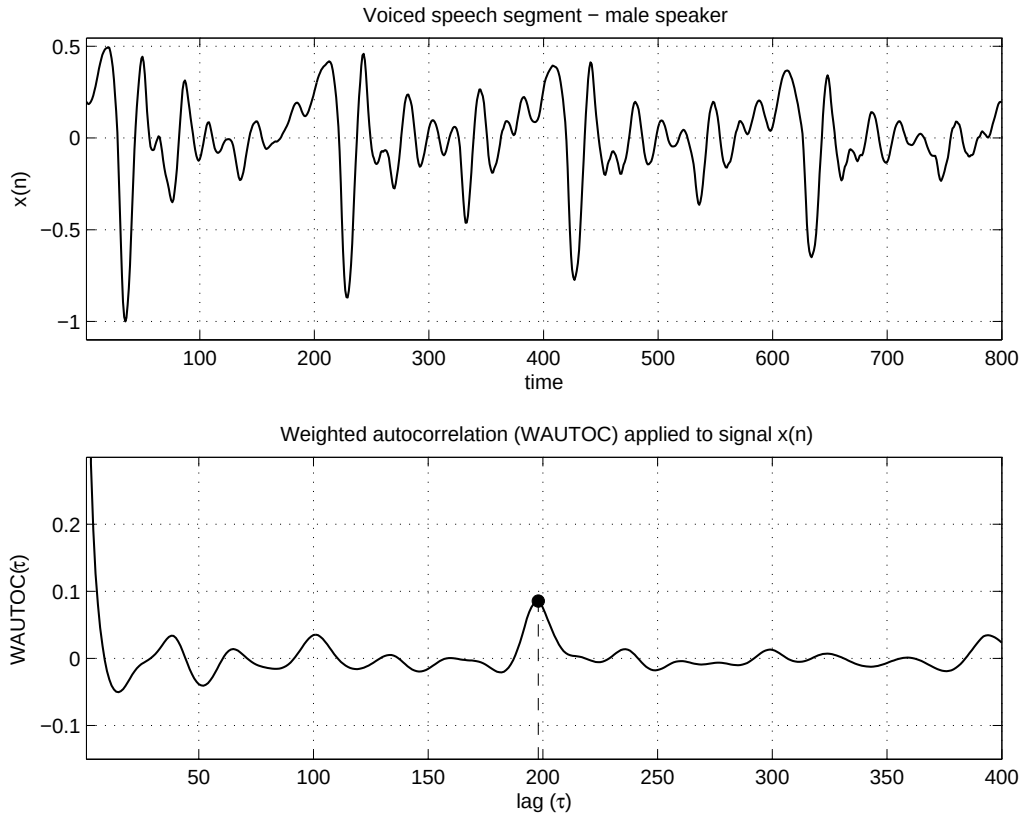

Figure 2.16: *Example of Weighted Autocorrelation (WAUTOC) function applied to a segment of voiced speech from a male speaker. The largest non-zero offset peak is found at lag* $\tau = 198$.

YIN - Cumulative Mean Normalized Difference Function (CMNDF)

The *YIN* algorithm is a time domain based algorithm derived from the autocorrelation function which represents one of the state of the art among the pitch detection algorithms [17].

The basic building block of this algorithm is the difference function:

$$d(\tau, q) = \sum_{n=0}^{N-1} [x(q+n) - x(q+n+\tau)]^2, \tag{2.18}$$

which, expanding the term inside the square brackets, can be expressed in term of the Autocorrelation Function $r(\tau, q)$ in Equation 2.13:

$$d(\tau, q) = N\left\{r(0, q) + r(0, q + \tau) - 2r(\tau, q)\right\}. \tag{2.19}$$

The first two terms at the right-hand side of Equation 2.19 are energy terms. Assuming them constant, the difference function $d(\tau, q)$ would express the opposite variations of the autocorrelation function $r(\tau, q)$. This is not always true, since the second term depends on the variable τ and may vary depending on the signal amplitude. Nevertheless, as reported by the author, Equation 2.18 proved to behave better than the Autocorrelation Function. It resulted less sensitive to changes in signal amplitudes, being thus less prone to "too low/too high" f_0 estimation errors.

The difference function, as the ACF and AMDF, has an absolute minimum for $\tau = 0$ and can produce additional dips at frequencies corresponding to a strong first formant F_1. The frequency region of F_1 and of the fundamental frequency f_0 overlap, thus making difficult to set a lower limit in the pitch period search range.

To overcome this limitations, the Cumulative Mean Normalized Difference Function was derived:

$$d'(\tau, q) = \begin{cases} 1, & \tau = 0, \\ \dfrac{d(\tau, q)}{(1/\tau)\sum_{j=1}^{\tau} d(j, q)}, & \text{otherwise.} \end{cases} \tag{2.20}$$

This function provides some advantages compare to the previous one: first of all there is no need to set a lower limit for the search range, since $d'(\tau, q)$ starts from 1 and remain large for low lags.

The direct consequence of this is that "too high" errors, that is, pitch period estimates smaller than the real one, are reduced. Another advantage is represented by the normalization which permits to apply a threshold to the above function to further reduce pitch estimation errors. The YIN algorithm also includes a post processing procedure which corrects each

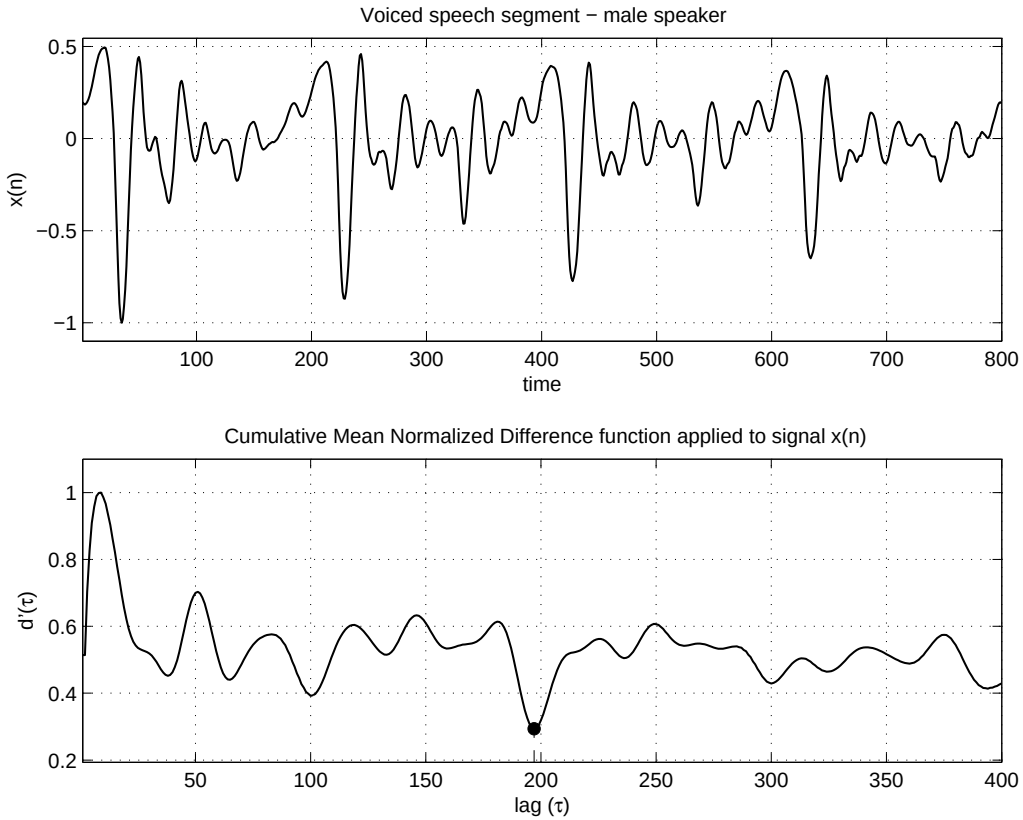

Figure 2.17: *Example of Cumulative Mean Normalized Difference function applied to a segment of voiced speech from a male speaker. The largest non-zero offset peak is found at lag* $\tau = 197$.

pitch estimate in case a large fluctuation between it and the surrounding estimates is found.

The YIN algorithm also includes a post processing procedure to correct the obtained pitch estimates. For each of them the CMNDF minimum value is considered, and is compared with the one relative to the surrounding estimates. In case a lower CMNDF value is found, the initial estimate is substituted with that associated with the lowest CMNDF provided by the search. This procedure results similar to median filtering or dynamic programming techniques [43], with the difference that the correction bases on estimate reliability rather than on continuity of successive values.

Figure 2.17 shows the Cumulative Mean Normalized Difference function

applied to a segment of voiced speech. The estimated pitch period resulted in $\tau = 197$ samples.

2.3.2 Frequency domain analysis

Frequency domain analysis bases on the spectral representation of the speech signal which is derived primarily by means of the Fourier transform. The main reason to switch to the frequency domain is that here the harmonic structure of the excitation signal is better revealed, as shown in Figure 3.5.

A common tool which provides the Fourier representation of a signal is the Discrete Fourier Transform (DFT, Equation 3.4) which is defined for sampled periodic signals. Since the speech signal is a quasi-stationary signal, the speech frequency representation provided by the DFT does not always represent a reliable source of information. This is one of the reason why the frequency domain PDAs gained the reputation, among a part of the research community, to be "clumsy and non-versatile approaches to pitch determination" [43].

However, several methods based on this domain have been devised and successfully tested. In any case, the frequency domain provides important clues on the harmonic structure of the signal which are not always evident in the time domain. This is particularly true in case the signal is distorted by noise or reverberation.

Another important characteristic of this approach is that, after the input signal has been down-sampled for computing the DFT, the resolution can be easily recovered in the frequency domain by means of interpolation.

To estimate f_0 in the frequency domain, a direct approach would be the localization of the first peak in the spectrum. This approach, however, cannot cope with a speech signal where the fundamental frequency is weak or absent. In addition, since the frequency resolution of the DFT is con-

stant with frequency, the relative resolution will get lower for decreasing values of the estimated f_0.

To overcome these limitations, other approaches measure the spacing within higher harmonics of the fundamental frequency and estimate f_0 computing their weighted average.

Harmonic Product Spectrum (HPS)

The Harmonic Product Spectrum (HPS) bases on the principle of spectral compression and exploits information provided by the spectrum of the speech signal [73, 96]. Given a speech signal segment $x(n)$, its logarithmic power spectrum is computed and compressed along the frequency axis by integer factors. The original spectrum and the compressed versions are then added together to provide the (logarithmic) HPS:

$$P(k) = \sum_{m=1}^{M} \log |X(mk)|^2 = 2 \log \prod_{m=1}^{M} |X(mk)|, \tag{2.21}$$

where M denotes the total number of spectra involved in the computation. $X(k)$ represents the discrete Fourier transform of $x(n)$, with the convention that the zero frequency bin has index $k = 0$. To obtain the Harmonic Product Spectrum, the antilogarithm of $P(k)$ must be taken.

When the compressed spectra are added together, the contributes of the harmonics present in the speech signal add constructively, since they are multiple of the fundamental frequency f_0. The frequency components of noise and of unvoiced speech instead, if present, do not exhibit the same relationship among each other, and will consequently be smeared out by the sum operation.

Figure 2.18 shows the working principle of the HPS algorithm. The top right panel shows the logarithmic power spectrum $\log |X(k)|^2$ of a segment of voiced speech from a male speaker (shown at the left). The peak due

to the fundamental frequency $f_0 \approx 100\ Hz$ is clearly visible as well as the harmonics $f_l = l f_0,\ l = 2, 3, \ldots$

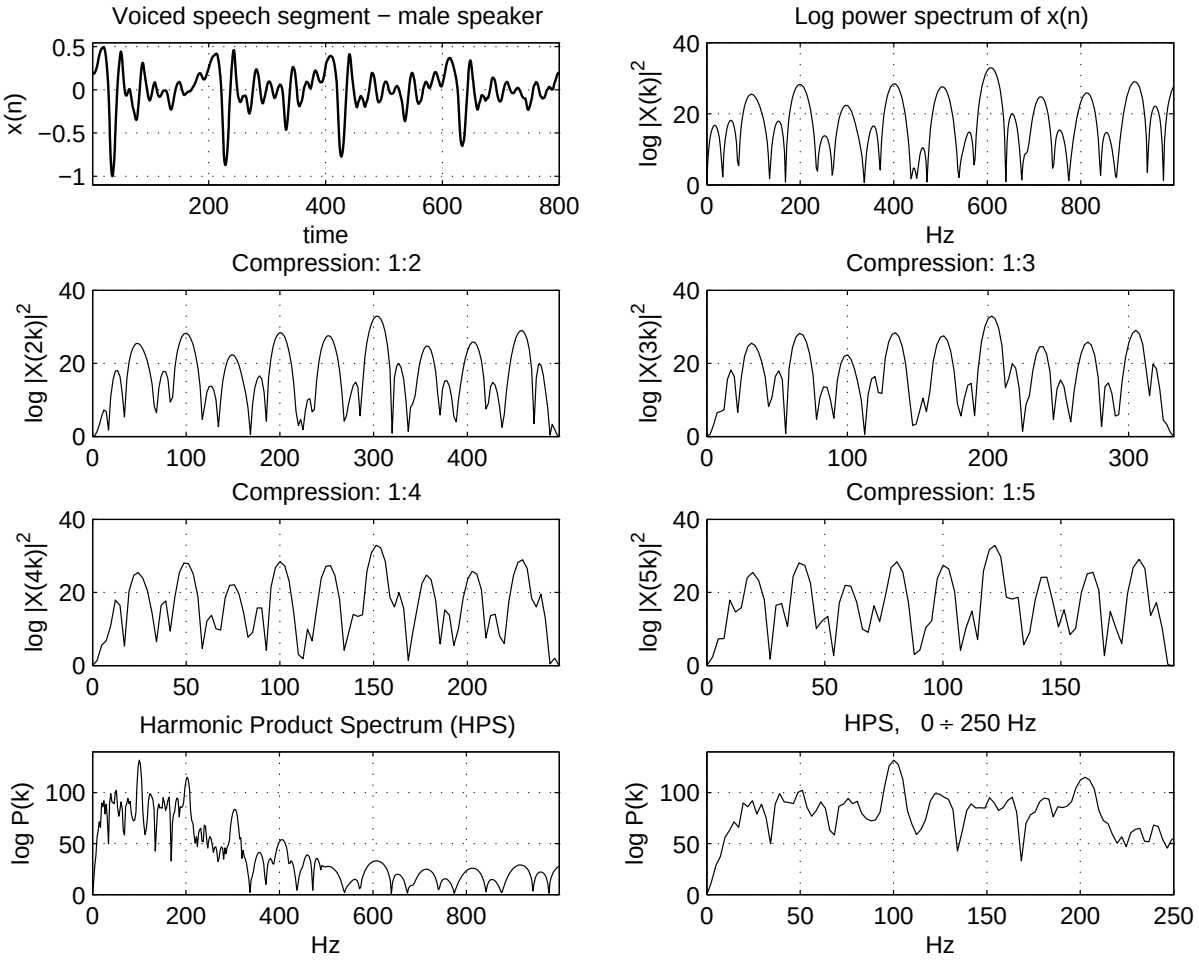

Figure 2.18: *Example of Harmonic Product Spectrum computed on a segment of voiced speech from a male speaker. Top left: voiced speech segment $x(n)$; Top right: logarithmic power spectrum of signal $x(n)$; Second and third row: compressed logarithmic power spectrum* $\log|X(mk)|^2$*, with* $m = 2, 3, 4, 5$ *respectively; Bottom left: Harmonic Product Spectrum of $x(n)$; Bottom right: magnification of HPS function from* $0 \div 250\ Hz$*. The largest peak found around* $100\ Hz$ *determines the estimated* f_0*.*

In the example, four compressed versions of $\log|X(k)|^2$ are calculated, that is $\log|X(mk)|^2,\ m = 2, \ldots, 5$ (four panels in the middle), and their sum, according to Equation 2.21, is computed and plotted in the bottom left panel of figure. The bottom right panel shows a magnification of

the result, where the largest peak represents the estimated fundamental frequency, which is $f_0 \approx 100Hz$.

This algorithm has two main advantages: it is particular robust to noise and does not need the fundamental frequency to be particularly strong to provide the correct estimate.

Frequency and Period histograms

The Harmonic Product Spectrum described previously is a generalization of the principle of spectral compression which was firstly introduced in [96], where the frequency and period histograms are proposed. The procedure to build up frequency histograms is similar to the way that HPS was derived. The difference is that instead of compressing the whole spectrum, just the peak frequency positions from each power or amplitude spectrum are used to update a histogram at each compressing step. The frequency position in the histogram relative to the largest number of occurrences will then determine the estimated f_0.

The same approach can be taken using the signal period values instead of frequencies: by means of a filter bank of narrow band-pass filters, the period of each filter output is estimated and its value used to update a histogram of period occurrences. As for the frequency histogram, the fundamental period estimate will be that with the highest number of occurrences.

Psychoacoustically-based harmonic pattern matching

The PDAs based on psychoacoustic analysis employ functional models of pitch perception applied to speech signals: the harmonic structure of the speech signal is analyzed in the frequency domain and used to increase the robustness of the fundamental frequency estimate.

Among others, two pitch extraction algorithms were firstly conceived, both based on harmonic pattern matching. The first, described in [59,

60], "maximizes the energy of the signal frequency components that pass through a spectral comb"; the second, reported in [79], "minimizes the difference between the input spectrum and reference spectra". Both the spectral comb and the reference spectra characteristics depend on the parameter p, which represents the trial fundamental frequency. The term "trial" [43] refers to the fact that p is varied within a given range of frequency values and, for each of them, the score provided by the matching procedure is evaluated. The frequency value that obtained the best score, will then be output as the estimated f_0.

The spectral comb based PDA defines a frequency impulse sequence based on a trial fundamental frequency[6] p:

$$C(m,p) = \begin{cases} l^{-1/s} & m = lp; \quad l \in \mathbb{Z}^+,\ s > 1, \\ 0 & \text{otherwise}, \end{cases} \tag{2.22}$$

where m is the spectrum bin index and s a positive integer. Given the discrete frequency spectrum $X(k)$ computed from the speech signal $x(n)$, its absolute value $X'(k) = |X(k)|$ is derived to compute the harmonic estimator function $X_C(p)$ as follows:

$$X_C(p) = \sum_{l=1}^{N/2p} X'(lp)C(lp,p), \tag{2.23}$$

where $N/2$ is less or equal the spectrum bin index corresponding to the Nyquist frequency. Equation 2.23 reaches its maximum when the spacing between the peaks of the spectral comb $C(m,p)$ match the harmonic structure of the voiced speech spectrum $X(k)$. The value of p, corresponding to this maximum, provides thus the fundamental frequency estimate.

[6] Actually, p represents here the the index position in the discrete spectrum relative to the trial frequency.

The reason for assigning a decreasing amplitude to the comb filter peaks in Equation 2.22, lays in the fact that a fundamental frequency harmonic that matches a certain peak of the comb $C(m,p)$, with p corresponding to the actual fundamental frequency, will also match a peak, with lower weight, of the comb filter $C(m,p')$, with p' sub-multiple of p.

In the latter case, the difference in weighting will guarantee that the value of $X_C(p')$ will be less than $X_C(p)$, thus avoiding that a sub-multiple of f_0 is provided as the final estimate.

Another approach, similar to the one just described, bases on the difference between the magnitude spectrum $X'(k)$ and a reference spectrum, which is defined as

$$R(m,p) = \begin{cases} |H(m)| & m = lp; \quad l \in \mathbb{Z}^+, \\ 0 & \text{otherwise}, \end{cases} \tag{2.24}$$

where the function $H(m)$ represents the vocal tract transfer function, estimated applying LPC analysis to the speech signal as reported in Section 2.2.3. The frequency comb filter resulting from Equation 2.24 is similar to that of Equation 2.22 with the difference that each peak weight is now related to the current vocal tract configuration.

To estimate the fundamental frequency, the spectral distance function is calculated, over L harmonics, as follows:

$$D(p) = \frac{1}{L} \sum_{l=1}^{L} |\log R(lp,p) - \log X'(lp)|, \tag{2.25}$$

and the value of p for which $D(p)$ reaches its minimum provides the estimated fundamental frequency. The main advantage of this approach, is that the formant positions and amplitudes do not affect the computation of the spectral distance. In fact, taking the difference of the logarithmic

spectra, as in Equation 2.25, actuates as a spectral flattener, that is, removes the effects of the vocal tract. In Figure 2.19 an example of the performance of this approach is given.

After limiting the spectral range of the signal spectrum $X(k)$ to $0 \div 2\ kHz$ (second panel), LPC analysis is applied and the vocal tract transfer function $H(m)$ is obtained (third panel). The spectral distance $D(p)$ is computed then for each value of p in the range $50 \div 500\ Hz$ and its minimum ($125\ Hz$ in the example) is taken as the estimated f_0 (bottom panel).

Another PDA which is based on a functional model of speech perception [38], but exploits the frequency spectrum in a way similar to that described in this section, is reported in [23]. In this PDA an harmonic sieve is used instead of a spectral comb, to retain the optimal harmonic structure. Spectrum peaks which passes through the sieve, are considered disregarding their amplitude value and the size of the sieve meshes is proportional to their center frequency, in accordance to the auditory model.

Cepstrum processing

Computing the cepstrum [71, 72] of a signal is equivalent to perform a homomorphic transformation [78]. The theory of homomorphic systems concerns systems where signals are combined together by means of convolution.

As it will be shown in Chapter 3 (see for example Figures 3.2 and 3.3), a speech signal $x(n)$ can be thought of as the result of a convolutional operation between the excitation signal $s(n)$ and the vocal tract transfer function $h(n)$, or, in the frequency domain, as the product between the respective discrete Fourier transforms:

$$x(n) = s(n) * h(n), \qquad X(m) = S(m) \cdot H(m) \tag{2.26}$$

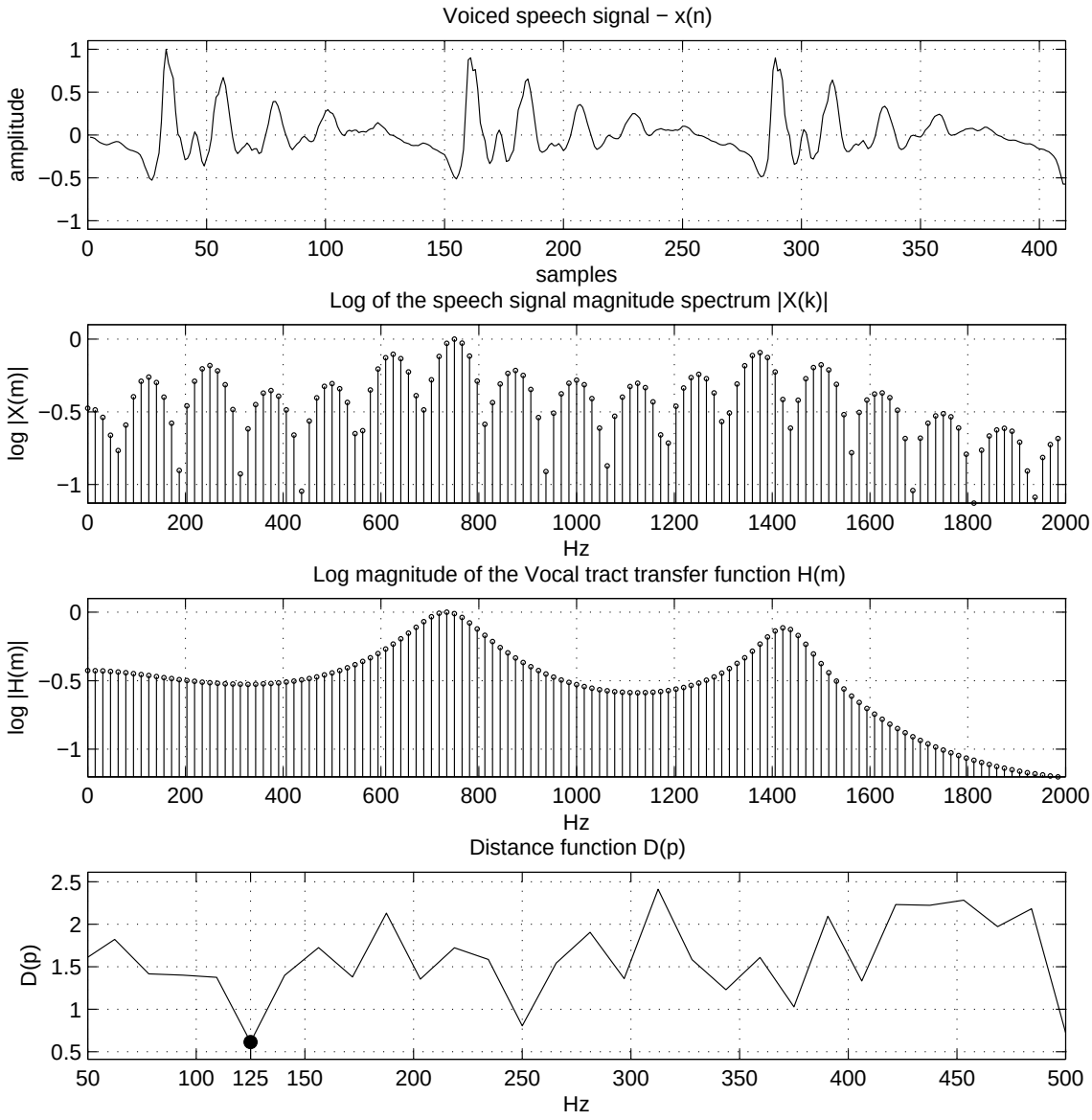

Figure 2.19: *Example of LPC based spectral distance function. Top panel: voiced speech signal $x(n)$; Second panel: logarithmic magnitude of the signal spectrum $X(k)$; Third panel: logarithmic of the vocal tract transfer function $H(m)$, estimated by means of LPC analysis; Bottom panel: distance function $D(p)$ computed as the log difference between $X(k)$ and $H(m)$. Its minimum determines the estimated fundamental frequency, $f_0 = 125\ Hz$.*

The objective of cepstrum processing is to separate the effect of $s(n)$ from that of $h(n)$, that is, to undo the convolution shown at left side of

Equation 2.26.

For this, the logarithmic of the signal power spectrum is computed so that the product turns into a sum:

$$\log |X(m)|^2 = \log |S(m)|^2 + \log |H(m)|^2, \tag{2.27}$$

and its inverse discrete Fourier transform is calculated, providing the power cepstrum[7], denoted with $x(d)$, where the variable d takes the name of "quefrency" and is a measure of time, as the lag variable τ in the autocorrelation function:

$$x(d) = s(d) + h(d) \tag{2.28}$$

As shown in the center panel of Figure 2.20, the log power spectrum of a voiced speech signal has the shape of a high frequency cosine-like ripple due to the harmonics, modulated by a low frequency ripple (plotted with dashes) due to the vocal tract effect. These two components are additive in the log domain. If they are thought of as time domain signals, their Fourier transform will ideally be a spectrum with a pulse in correspondence of the fundamental frequency, and a spectrum with energy just in the low frequency region, respectively. This is approximately the behaviour that the functions $s(d)$ and $h(d)$ show, respectively[8].

The sum of these functions, (Equation 2.28) is plot in the bottom panel of the figure which evidences the peak due to the high-frequency component at quefrency $d = 129$ samples, which will be the final pitch estimate.

Ideally, the contribute of the excitation source in the cepstrum domain, $s(d)$, shall be a train of impulses. Actually, due to the windowing operation

[7]Cepstrum and complex cepstrum are obtained when the power spectra in Equation 2.27 is substituted by spectra and amplitude spectra, respectively.

[8]Computing the inverse or direct discrete Fourier transform of even functions, as $\log |X(m)|^2$, returns the same result.

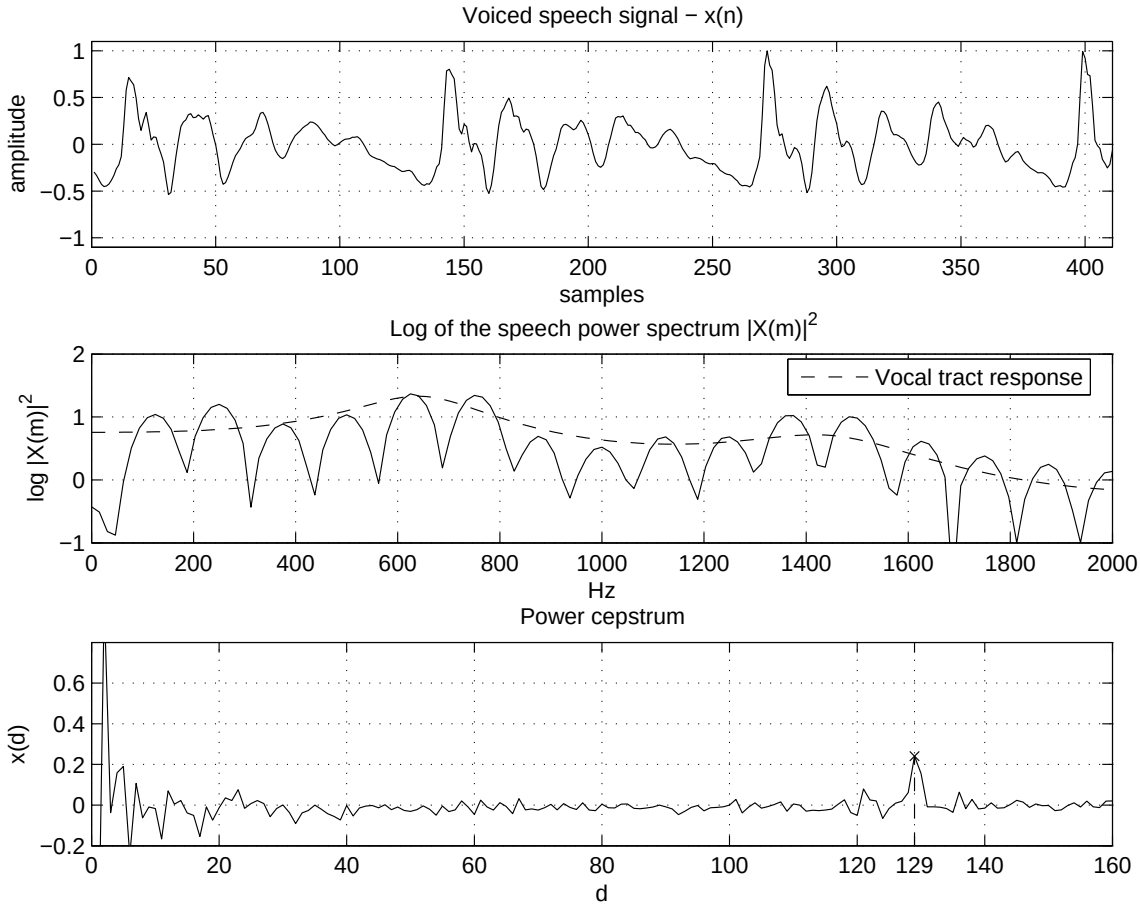

Figure 2.20: *Example of cepstrum processing. Top panel: voiced speech signal* $x(n)$*; Middle panel: logarithm of the power spectrum of signal* $x(n)$*. The high frequency cosine-like ripple is plotted with a continuous line, while the vocal tract contribute is plotted with a dashed line; Bottom panel: cepstrum function for signal* $x(n)$*. The peak at quefrency* $d = 129$ *represents the estimated signal fundamental period.*

applied on signal $x(n)$, prior to its discrete Fourier transform computation, the cepstrum peaks decrease in amplitude with increasing quefrency. To compensate for this effect, a weighting of the cepstrum function $x(d)$ is carried out before estimating the fundamental frequency, thus avoiding possible halving/doubling pitch errors.

Being the cepstrum technique able to separate the vocal tract from the excitation source effects, results quite insensitive to formant positions. On

the other hand, it needs several harmonics so that a peak in the cepstrum is produced, being thus not suitable to estimate the pitch of sinusoidal signals. Even so, after it was first published, cepstrum processing became a reference pitch estimation technique for many pitch detection which were consequently compared against it.

Dominance Spectrum based f_0 estimation

The dominance spectrum was firstly proposed in [67, 66], where a robust fundamental frequency estimation technique is presented. The method is regarded by the authors, as a frequency domain based method, since it exploits the Instantaneous Frequency (IF) defined as the phase derivative with respect to time of a sinusoidal component [1].

Defining with $\phi(f)$ the phase of the speech signal component output by a narrow band-pass filter with center frequency f, the IF $\dot{\phi}(f)$ is defined as its phase derivative with respect to time.

The degree of dominance $D_0(f_i)$ is thus defined as:

$$D_0(f_i) = \log \frac{1}{B(f_i)^2}, \qquad B(f_i)^2 = \frac{\sum_{k=i-K/2}^{i+K/2} [\dot{\phi}(f_k) - f_i]^2 \cdot X(f_k)^2}{\sum_{k=i-K/2}^{i+K/2} X(f_k)^2} \tag{2.29}$$

where $X(f_k)$ represents the value of the discrete Fourier transform of $x(n)$ at the frequency value relative to the k-th bin. Function $B(f_i)^2$ is derived as the weighted average of the squared difference between the center frequency f_i, and the IFs $\dot{\phi}(f_k)$, computed over a frequency range of $K+1$ frequency bins.

When a harmonic component of a voiced speech signal coincides with the bin center frequency f_i, the instantaneous frequency $\dot{\phi}(f_k)$ takes a value

close to f_i and $B(f_i)^2$ becomes minimum, producing a peak in the function $D_0(f_i)$.

The dominance spectrum is obtained computing the degree of dominance in Equation 2.29 for all values of f_i. Its peculiarity is that it is characterized by sharper peaks in correspondence of the fundamental frequency harmonics, compared to the power spectrum counterpart. In addition, when background noise is present, the value of $\dot{\phi}(f_k)$ increases proportionally to f_k, in frequency regions where speech harmonics are absent. For this reason the dominance spectrum is more suitable than the power spectrum for f_0 estimation in noisy conditions. Figure 2.21 shows an example of dominance spectrum (top) and power spectrum (bottom), computed on a voiced speech segment with $f_0 \approx 117\ Hz$. The dominance spectrum show sharper peaks at harmonic positions, compared to the power spectrum.

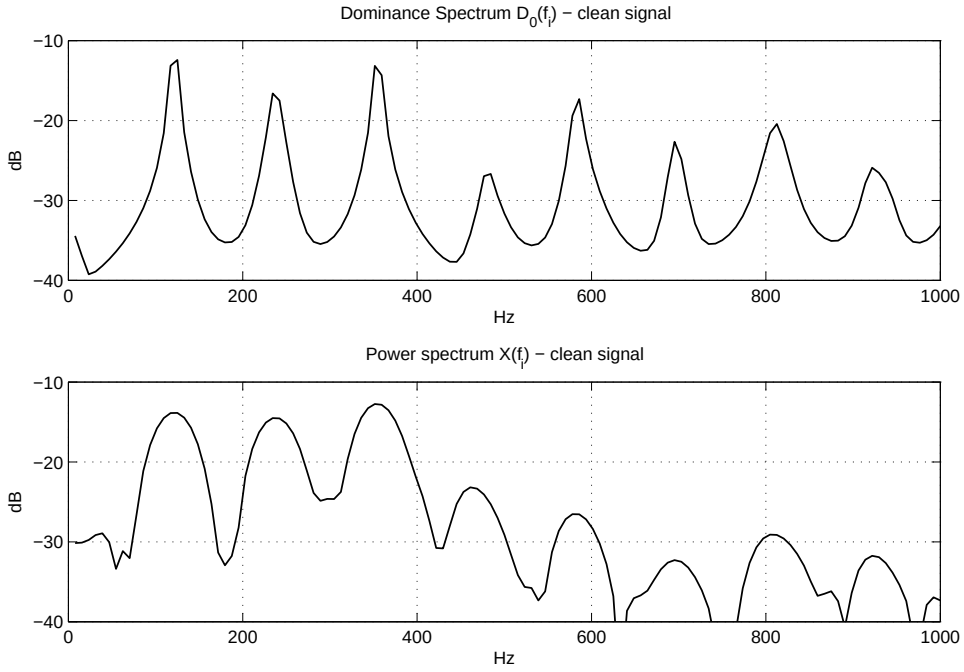

Figure 2.21: *Top: dominance spectrum of a voiced speech signal with* $f_0 \approx 117\ Hz$*; Bottom: power spectrum computed on the same segment of speech signal.*

To finally estimate the fundamental frequency, first the harmonic dom-

inance function is defined as:

$$D_{t_0}(f_i) = \sum_{l=1}^{L} \{D_0(lf_i) - E(D_0(f_i))\}, \tag{2.30}$$

where $E(D_0(f_i))$ is the average value of $D_0(f_i)$ over all frequency bins and L is the number of harmonics considered in the computation. Then the value of f_i for which Equation 2.30 takes a maximum, is chosen as the final fundamental frequency estimate:

$$f_0 = \arg\max_{f_i} \{D_{t_0}(f_i)\}. \tag{2.31}$$

Additional post-processing by means of dynamic programming is done on the estimated f_0 values to correct possible errors and to provide more precise estimates.

2.3.3 Maximum-likelihood pitch determination

Maximum-likelihood pitch determination uses a statistical approach to find the parameters which model a segment of speech signal. Given a signal $a(n)$ of length K samples, consisting of a Gaussian noise source $g_n(n) \in \mathrm{N}(0, \sigma^2)$, i.e. with zero mean and variance σ^2, and a voiced speech signal $x(n)$ with fundamental frequency f_0:

$$a(n) = x(n) + g_n(n), \quad 0 \leq n \leq K - 1, \tag{2.32}$$

it can be written in vector notation as

$$\boldsymbol{a} = \boldsymbol{x} + \boldsymbol{g_n}. \tag{2.33}$$

The objective is to find $\hat{f}_0$, $\hat{\sigma}^2$ and $\hat{\boldsymbol{x}}$ such that they are the most likely values, in the least-squares sense, for f_0, σ^2 and $\boldsymbol{x}$.

As shown in [33, 111], this can be achieved exploiting the Gaussian characteristics of the noise source and modeling the signal $a(n)$ as a stochastic process as follows:

$$g(\boldsymbol{a}|f_0, \boldsymbol{x}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{K/2}} \exp\left\{ \frac{1}{2\sigma^2} \sum_{n=0}^{K-1} \left[a(n) - x(n)\right]^2 \right\}. \tag{2.34}$$

Function g represents the probability density function that vector $\boldsymbol{a}$ is generated by the sum of vector $\boldsymbol{x}$, modeled as a periodic component with frequency f_0, and of vector $\boldsymbol{g_n}$ which has statistics $\mathrm{N}(0, \sigma^2)$. Finding the values of $\hat{\boldsymbol{x}}$, $\hat{f}_0$ and $\hat{\sigma}^2$ which maximizes Equation 2.34 provides the optimal solution in the least-squares sense. This is done computing the $\log g(\boldsymbol{a}|\hat{f}_0, \hat{\boldsymbol{x}}, \hat{\sigma}^2)$ and setting its partial derivatives to zero.

The final formulation provides a solution which is similar to the harmonic compression based PDAs: a comb filter which enhances the harmonic structures and optimally matches the signal in the time domain. These PDAs demonstrated to be resistant to noise conditions but somehow too sensitive to octave errors in case a strong first formant coincides with a fundamental frequency harmonic.

Chapter 3

From speech modeling to pitch based applications

The speech production mechanism has been studied since ancient times and, nowadays, the functions of the organs involved during speech production as well as their effects on the uttered sound characteristics are well known. When voiced sounds are produced, the vocal folds oscillates regularly under the air pressure which accumulates below them and this phenomenon is the main responsible of the pitch perceived by a listener. Pitch is thus a subjective perception which is strongly related with the speech fundamental frequency f_0, that measures the frequency of such oscillations. In the context of speech applications, and particularly when fundamental frequency estimators are concerned, the terms "pitch" and "fundamental frequency" are usually used with the same meaning.

To estimate f_0 a common approach consists in acquiring the speech signal by means of an acoustic sensor (microphone) and analyzing the provided waveform. The analysis is generally carried out relying on signal processing techniques and on the source-filter model, which approximates the speech production mechanism as a vocal tract filter driven by an excitation signal.

In case the processed speech signal is not degraded by the ambient

noise and reverberation effects, current proposed solutions provides very accurate f_0 estimates. For this reason, many speech processing applications designed to work in such good acoustic conditions, integrate a pitch extractor algorithm to exploit the provided f_0, thus improving their performance. However, when pitch extractor algorithms are tested on noisy and reverberant signals they lack of accuracy and robustness. Noise and reverberation are generally present in any real-world context and only posing particular constraints, as for example the use of close-talk microphones in a quiet room, it is possible to avoid them. To study these detrimental acoustic phenomena, a mathematical model was devised. This model permits to obtain, by means of computer simulations, speech signals as if they were recorded under some given acoustic conditions. The design and test of pitch extractor algorithms, able to cope with real-world acoustic scenarios, is thus made easier.

3.1 Speech Production

Acoustic speech output in humans results from a combination of a source of sound energy, the larynx, modulated by a transfer function determined by the shape of the supra-laryngeal vocal tract [36, 74].

Speech signal can be broadly classified into voiced and unvoiced speech, including sounds which can result from a simultaneous combination of both.

Voiced speech is produced by a repeating sequence of events driven by the airflow produced by the lungs. First the vocal cords are brought together (adduction), and the air pressure in the larynx increases until it gets greater than the resistance offered by the vocal folds themselves. At this point the vocal folds are forced to open and the airflow propagates through the oral, nasal, and pharyngeal cavities (see Figure 3.1), which actuate as

a resonator modulating the airflow that is finally radiated through lips and nose.

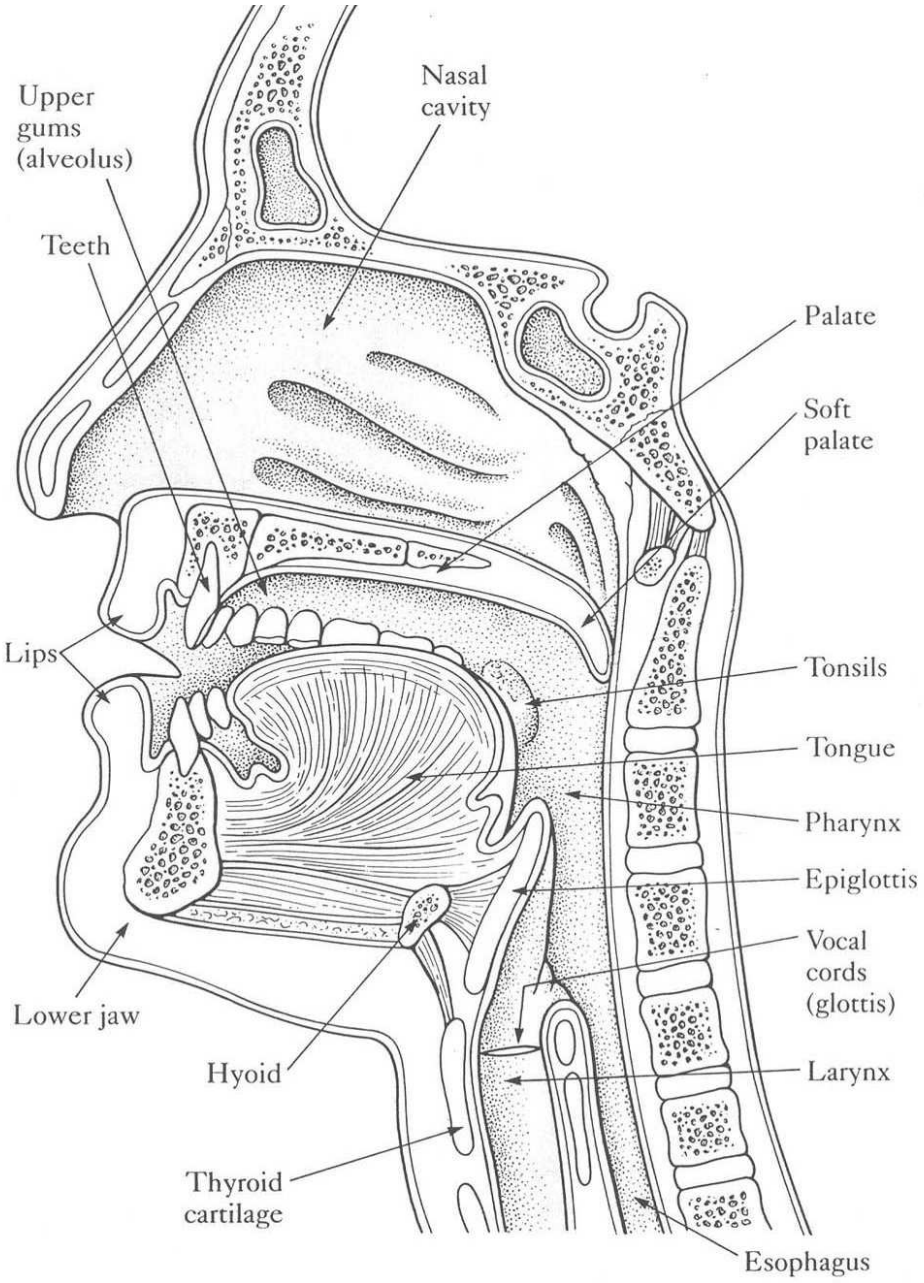

Figure 3.1: *Vocal tract configuration with raised soft palate for articulating non-nasal sounds [19].*

As soon as this happens, the airflow velocity increases and, for the Bernoulli effect, the air pressure in the larynx decreases. This cause the

vocal folds to close rapidly and the process repeats this way in a quasi-periodic fashion as long as a steady supply of pressurized air is generated by the lungs through the larynx.

The frequency with which the glottis vibrates during phonation, determines the fundamental frequency f_0 of the laryngeal source and largely depends on the tension of the laryngeal muscles and the air pressure generated by the lungs, contributing to the perceived pitch of the produced sound.

While frequency is a physical measurement of a vibration, pitch is related to the human perception and, the relationship between them has been studied in depth and involves complex psychoacoustic phenomena[1].

Although what the solutions proposed in this field actually do is f_0 estimation, often they are regarded as pitch detection algorithms. Since the psychological relationship between the f_0 of a given signal and the relative perceived pitch is well known[2], the above distinction is not so important given that, a true pitch detector, should take into account perceptual models in order to estimate pitch and give a result on a pitch scale.

Although, in most European languages, individual phonemes are recognizable regardless of the pitch, this is mostly responsible for intonation patterns associated with questions and statements and carries information about speaker emotional state. In tonal languages instead, pitch motion of an utterance contributes to the lexical information in a word.

The frequency spectrum of voiced speech reveals high energy in the frequency regions relative to the fundamental frequency and its harmonics, which falls off gradually for increasing values of frequency. The final spec-

[1]For example the note A above middle C is perceived to be of the same pitch as a pure tone of 440Hz, but does not necessarily contain that frequency.

[2]Pitch is loosely related with the base 2 logarithmic of the fundamental frequency, that is, for every doubling of f_0, the perceived pitch increases of about an octave. The relation is, however, biased by many factors such as the sound frequency, intensity, harmonic content, etc [16, 109].

trum shape however, is partly independent of f_0 and is determined by the way the vocal folds close and open and by the vocal tract shape. This actuates as a time-varying acoustic filter which suppresses certain frequencies while allowing and boosting other frequencies which form local maxima in the spectrum and are named *formants*. Frequency position and intensity of formants depends on the overall shape, length and volume of the vocal tract.

During *unvoiced* speech, vocal folds do not vibrate and a constriction is formed at some point along the vocal tract and air is forced through the constriction to produce turbulence. Given the aperiodicity and random behaviour of the turbulent flow produced, in the frequency domain unvoiced speech is characterized by a continuous frequency distribution, opposed to the discrete harmonic set of voiced spectrum.

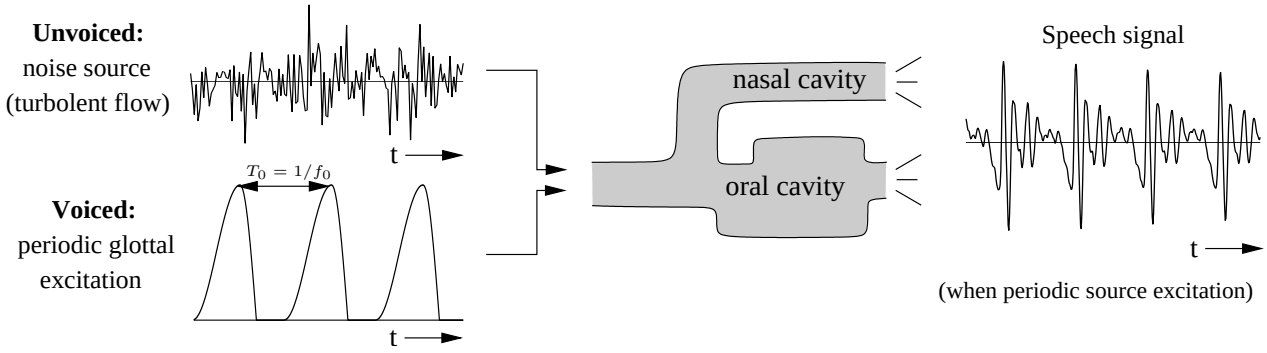

Figure 3.2: *source-filter model: time domain. At the left is the random varying waveform in the case of unvoiced speech (top) or glottal pulse shaped waveform representing voiced speech (bottom). One of this sources (or a mixture of both) is filtered by the vocal tract (centre) which is represented in gray as a nasal plus oral cavity. The output (right) is an example of voiced speech which is the result of filtering the periodic source signal with the impulse response of the vocal tract and considering the lips radiation effect.*

According to the source-filter model for speech production [25], the speech signal can be modeled as a convolution of the excitation signal with the vocal tract impulse response as follows

$$x(n) = s(n) * h(n), \tag{3.1}$$

where $x(n)$ is the sampled speech signal sample, $s(n)$ is the sampled version of the glottal pulse excitation signal (voiced speech) or a random discrete function (unvoiced speech), and $h(n)$ is the sampled vocal tract impulse response, which includes here the lips radiation effect.

In the z-transform domain (or discrete frequency domain) Equation 3.1 turns into

$$X(z) = S(z) \cdot H(z), \tag{3.2}$$

being $X(z)$, $S(z)$ and $H(z)$ the z-transformations of $x(n)$, $s(n)$ and $h(n)$, respectively. Equation 3.2 results very useful since in the z-domain, the convolution operator '$*$' turns into a multiplication and this makes it possible to obtain $S(z)$ by simple multiplication of $X(z)$ by the inverse filter $1/H(z)$.

Figure 3.2 shows the source-filter model in the time domain, a schematic model of the human speech production system where the source is a a combination of periodic pulses, generated by vocal cords vibrations at the glottis, and of an contribution from turbulent flow. When only the first contribution is present the output of the generation process is named *voiced speech* while, when only turbulence flows is generated, *unvoiced* speech is produced.

Figure 3.3 shows the same schematic model but in the frequency domain. The periodic source here is represented as a series of frequency lines, spaced f_0 Hz one from the other and falling off gradually.

The oral and nasal airways, as well as the lips radiation effect, are shown here as a time-varying acoustic filter which reflects the overall shape, length and volume of the vocal tract.

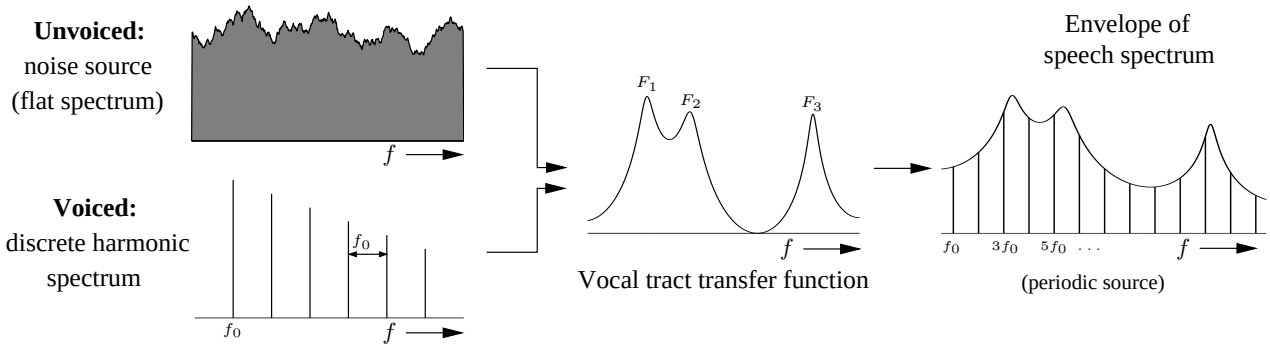

Figure 3.3: *source-filter model: frequency domain. At the left (top) is an almost flat spectrum of a random signal representing unvoiced speech source and the spectrum (bottom) of a periodic glottal source, characterized by equally spaced (f_0) spectral lines. One of these sources (or a mixture of both) is filtered by the vocal tract transfer function (centre). In case of voiced speech, the articulatory organs are positioned so that specific frequency regions, i.e. formants (F_1, F_2, ...), of the input source are amplified. The output (right) is an example of voiced speech whose spectrum is the product of the equally spaced spectral lines by the vocal tract transfer function.*

The effect of this filter is to attenuate the passage of certain frequencies while amplifying the other frequencies. The peaks of these frequency regions, or formants, where local energy maxima occur, are usually referred to with labels F_1, F_2, ..., and their position depends on the particular sound produced, not on its pitch.

The different sounds produced in human language are grouped in phonemes, which are mental abstractions of speech sounds and represent the basic theoretical unit that can be used to distinguish words. Different phonemes are identified by patterns of prominent frequency regions, in particular the vowels show strong stable formants and are generally classified basing on the first two of them, F_1 and F_2.

For adult speakers, formant $F1$ is approximately in the range $300 \div 1000\ Hz$, and the lower its value, the closer the tongue is to the roof of the mouth. The frequency of $F2$ instead, is proportional to the frontness or backness of the highest part of the tongue and ranges about $850 \div 2500\ Hz$. Voiced

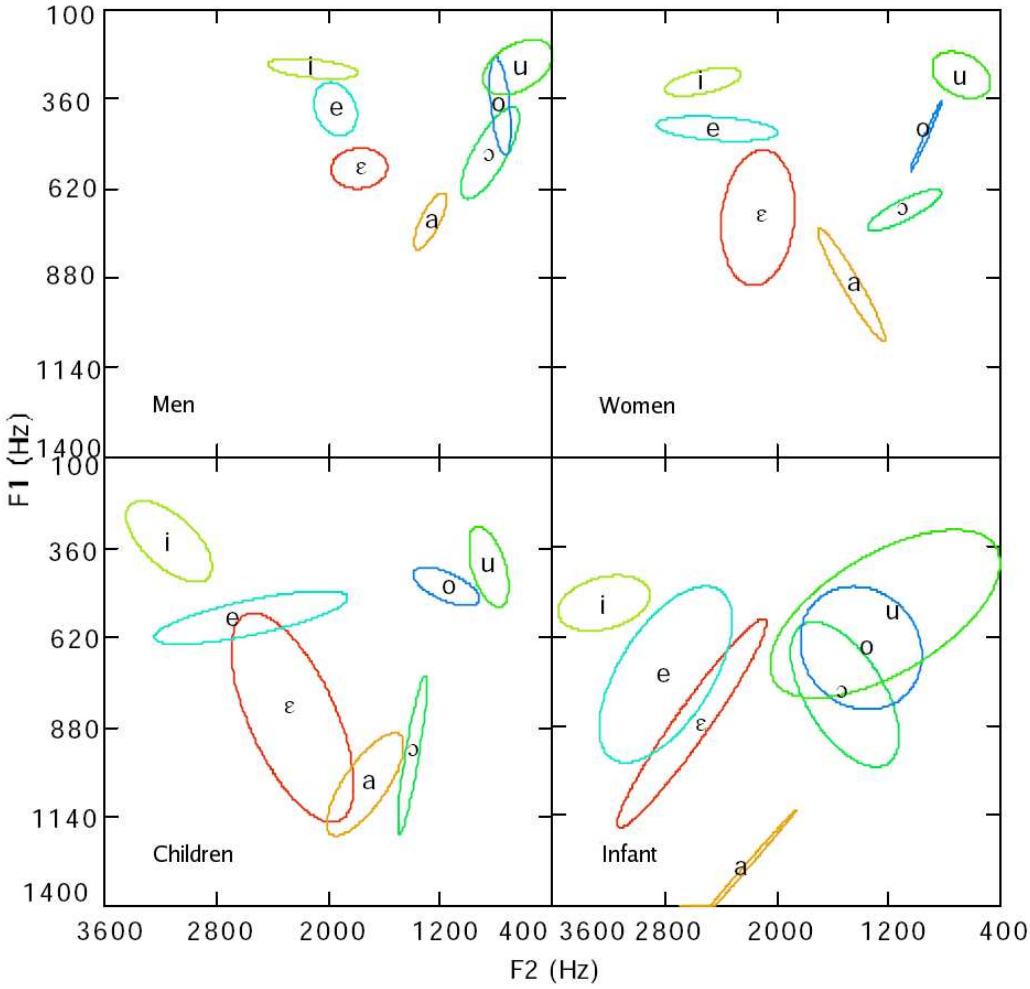

Figure 3.4: *$F1/F2$ chart of Italian vowels for males, females, children and infants determined from four groups consisting of four subjects each. Each ellipses show the area of existence for each vowel in the $F1-F2$ plane and is centered on the mean values of the estimated formants. The axis lengths and their orientation are determined by the standard deviation and covariance of $F1$ and $F2$ respectively [113].*

speech produced by female and male speakers have different formant frequency ranges, being these determined by the different size of their vocal tract. However the formant frequencies ratio keep consistent across males, females. This is depicted in Figure 3.4, where a $F1/F2$ chart of Italian vowels for males, females, children and infants is shown.

3.2 Basic of Fundamental Frequency Estimation

Time domain, frequency domain or a combination of both as well as other specific domains, allow each a particular representation of the studied signal. Most of the time, even if the analyzed signal source is common, different domain representations provide complementary information about the signal properties. The most common difference between time and frequency domain based analysis is that time domain provides, as the name suggests, better description of the time evolution of the waveform being considered. Frequency domain instead, shows how the signal energy is distributed among frequencies, something which is usually not readily observable by means of time domain analysis.

An important concept related with time and frequency based analysis is the *time-frequency uncertainty principle*, which states that there is a fundamental trade-off between the time resolution and frequency resolution achievable (See Appendix A).

3.2.1 The Discrete Fourier Transform (DFT)

Extracting the fundamental frequency (f_0) from a periodic sound signal, is related with detecting the lowest frequency component, or partial, among an equally spaced set of frequency components, which is characteristic of voiced sound.

The statement above can be made clearer considering the voiced speech signal as a continuous signal $x(t)$ with period T (left panel of Figure 3.5), and recalling the Fourier decomposition for such a signal

$$X_m = \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{T}{2}} x(t)\ e^{-j\frac{2\pi m}{T}t}\ dt \qquad\qquad x(t) = \sum_{m=-\infty}^{\infty} X_m\ e^{j\frac{2\pi m}{T}t}. \qquad (3.3)$$

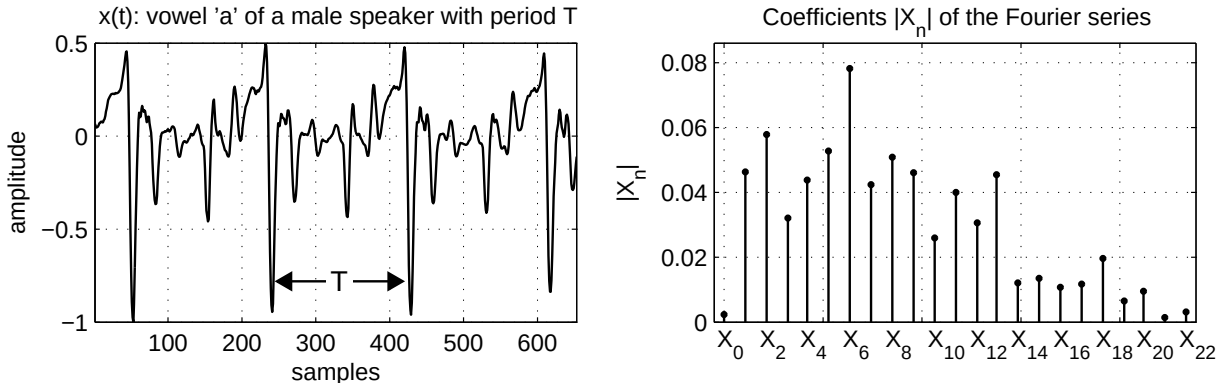

Figure 3.5: *Left: Example of voiced speech segment ('a' vowel) with period T; Right: subset of Fourier coefficients X_n relative to the voiced speech segment (labels are relative to even coefficients only).*

In particular, the expression at the right, shows that $x(t)$ can be written as a summation of infinite weighted complex exponentials, each with frequency multiple of the fundamental frequency, that is, $mf_0 = m/T$.

The complex weights X_m (right panel of Figure 3.5), are obtained from a period of the signal itself, as shown from the left equation of (3.3), and represent the harmonic contribute of the signal at that particular frequency mf_0. Those frequencies are harmonically related, meaning that the ratio between each of them and the lowest one[3], f_0, is a whole-number.

It is important noting that the integration limits of left equation of (3.3) can be changed to include any whole number of periods T, adjusting consequently the normalization factor $1/T$, and the result for the coefficients X_m will be the same.

In the practice, any handled signal is represented by a discrete set of values $x(nT_s)$, $n \in \mathbb{Z}$, obtained from sampling the continuous periodic signal $x(t)$ with sampling frequency $f_s = 1/T_s$. The correspondent version of the Fourier decomposition in the discrete time domain, is represented

[3] f_0 is considered here as the lowest frequency, since X_0 represents the mean of the signal in a period, a component not showing an oscillating behaviour and easily removable. Components X_{-i} for $i \geq 1$ are, as the Fourier theory shows for real $x(t)$, complex conjugate of X_i.

then by the Discrete Fourier Transform (DFT) and can be applied to any discrete periodic signal [78]:

$$X_m = \frac{1}{N}\sum_{n=0}^{N-1} x(nT_s)e^{-j2\pi\frac{nm}{N}} \qquad x(nT_s) = \sum_{m=0}^{N-1} X_m e^{j2\pi\frac{nm}{N}}. \tag{3.4}$$

It is still possible to exactly compute the coefficients X_m, as provided by Equation 3.3, applying the DFT (left equation of (3.4)), provided that:

- the sampling frequency $f_s = 1/T_s$ is set to a value greater than or equal to the Nyquist rate, that is, to twice the maximum spectral extension $F_{\max}$ of the considered signal, $f_s \geq 2F_{\max}$;
- the N values $x(nT_s)$, $n = 0, \ldots, N-1$, used in the summation of left equation of (3.4) must include exactly one (or more) period T of the signal itself, that is $T = NT_s$.

Note that the term $e^{-j2\pi\frac{nm}{N}}$ in left equation of (3.4) is invariant to translation of N samples, that is, $e^{-j2\pi\frac{nm}{N}} = e^{-j2\pi\frac{(n+lN)m}{N}}$, for any value of $l \in \mathbb{Z}$. Also, given that $x(nT)$ is considered periodic of N samples, the values of X_m will not change in case the summation is extended to comprise any whole number of periods, and the result divided by the same number in order to maintain the same dynamics.

In the practice, voiced speech signal is far from showing such a perfect periodic behaviour. During phonation, articulatory movements continuously take place to permit transitions between different phonemes thus changing formants position and amplitudes. Pitch also is not stationary: the glottis changes its fundamental frequency depending on intonation and emotional state. Both these phenomena entail that the output signal cannot be regarded as stationary. Therefore each instantaneous period, that

is, the signal segment between each pair of glottal closure instants, changes its duration and shape slowly over time.

In addition, when the DFT is computed on voiced segments, the period length is not known in advance, since if it was known, there would be no need of performing f_0 estimation. This makes it impossible to fulfill the second requirement listed above posing the need to introduce some approximations.

The common assumption that is made when voiced speech signal is considered, is to treat it as quasi-stationary, that is, a signal that can be regarded as stationary over a short segment. The actual length of the latter depends on too many variables to be determined uniquely, even though some work was done in this sense and vowels sound were estimated to be quasi-stationary for $40 - 80\ ms$ while, stops and plosives are time-limited by less than $20ms$ [81]. The variance of these figures are related with the speaker age and gender, as well as with the emotional state or the environmental noise.

This assumption, though not always verified, permits to compute DFT on a segment long enough to include at least one period of the smallest f_0 that has to be estimated.

Computing the DFT on a speech signal segment, is equivalent to multiply the latter for a window function, which is zero outside a desired interval. This procedure results in computing Equation 3.5, also referred to as the Short-Time Fourier Transform (STFT),

$$X_{n_0}(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n_0 + n) w(n) e^{-j2\pi \frac{mn}{N}}, \tag{3.5}$$

where $w(n)$ is a N samples length window function and n_0 is the starting

sample of the speech signal segment considered in the computation. The sampling period T_s has been omitted for simplicity.

Since the actual pitch period is not know, N is set so that the computation is carried out on a speech segment long enough to include more than one period. Doing this way, the resulting values $X_{n_0}(m)$ represent the complex weights X_m of Equation 3.4 computed on a discrete periodic signal with the period given by $x(n_0 + n)w(n)$, for $0 \leq n \leq N-1$. The spectrum values thus obtained are just an approximation of the values that would have been obtained computing Equation 3.4 on N samples of $x(n)$, with N set to match exactly the current pitch period value.

The simplest window function is the *rectangular* window, which has unitary amplitude within the target speech segment, and zero outside. However, this introduces high frequencies components in the DFT, associated with the abrupt edges of the window. This side effect can be mitigated using a smoother window, such as those reported in Table 2.1, but this will come at the main expense of the maximum frequency resolution achievable, which is maximum in case the rectangular window is used (see Section 2.3).

The number of complex multiplications and sums required to compute the DFT of a N samples length signal, is of the order of N^2, which means an algorithmic complexity of $O(N^2)$. In practical application, fast Fourier Transform (FFT) algorithms are employed to reduce the amount of computation time required. The most common known FFT algorithm is the *Cooley-Tukey algorithm* [15], which is a *divide and conquer* algorithm that recursively decomposes the original N points transform into two transforms, each of length $N/2$. To accomplish this, the value of N must be a power of two and the algorithm complexity reduces to $O(N \log N)$. Other solutions exist, which further reduces the overall complexity or the data storage space needed by means of in-place computation techniques [78].

3.2.2 The spectrogram

Time and frequency domain based analysis reveal different characteristics of a given speech signal. It is often the case that one or the other domain are not sufficient to describe completely the complexity of a speech signal. In fact, the time domain analysis completely disregard frequency information and the same holds for the frequency domain analysis. Not even when both descriptions are available separately, it is straightforward to derive the reciprocal relation of the time and frequency variables.

To gain a better insight into the speech signal characteristics, it is possible to create its spectrogram. A spectrogram is obtained as a succession of signal spectra, each computed on an adjacent time frame (successive frames can overlap) of the considered signal. Each spectrum is obtained using the Short Time Fourier Transform in Equation 3.5, with the parameter n_0 increasing at each step to span the whole signal length. Aligning each spectrum vertically, result in a three-dimensional plot of the energy of the frequency content of a signal as it changes over time. The three dimensions are the frequency, usually plotted along the vertical axis, the time, generally represented on the horizontal axis and the signal energy of each time-frequency point in the graph, plotted with different colors indicating the energy intensity[4].

Figure 3.6 shows an example of spectrograms computed on a speech signal from a female speaker. For both top and middle spectrograms, the frequency range is set from 0 to 3500 Hz and the time range spans one second of the signal shown in the bottom panel. On the right of each spectrogram, is placed a color-bar to map the colors in the graph to intensity values (dB in this example).

The top panel of the figure, is obtained computing the signal spectrum

[4]In case of black and white spectrograms, the energy intensity is displayed using grey levels.

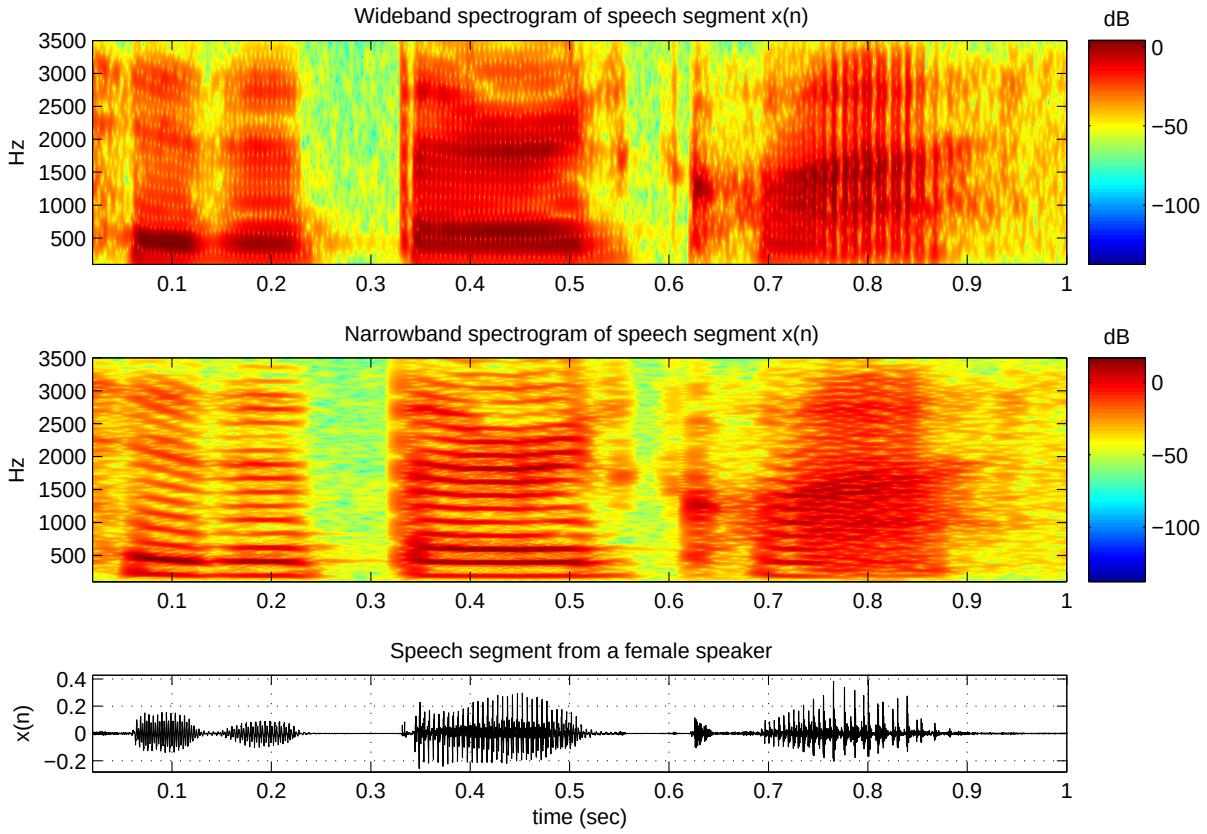

Figure 3.6: *A wide-band (top) and narrow-band (middle) spectrograms are shown along with the speech signal from a female speaker (bottom), used to derive them. The spoken sentence is: "*to the third class*".*

every 1 millisecond and using a frame length of 10 milliseconds. This settings provide poor resolution in the frequency dimension and good resolution in the time dimension. The final spectrogram is thus called wide-band spectrogram, and it is characterized by a vertically striated appearance. This is due to the fact that, as the short time analysis window slides along in time, it alternately covers high and low energy regions which occur within a pitch period in the waveform [78].

The middle panel instead, is obtained using the same analysis step but a longer frame length, that is, of 40 milliseconds. The resulting narrow-band spectrogram, provides thus higher resolution in the frequency dimension at

the expense of a poorer resolution in the time dimension. This time, the graph is characterized by horizontal striations, indicating that neighboring spectra vary slowly and smoothly with time. The reason for this is that the vocal tract movements can be considered slow compared to the length of the analysis window.

Wide-band spectrograms are particularly useful to track the variations of the vocal tract resonance frequencies, that is, the formants. In fact, a strong formant around 500 Hz for the first two voiced segments and two formants around 500 and 2000 Hz, respectively, are clearly visible in the spectrogram shown at the top of the figure. Narrow-band spectrograms, instead, better reveals the harmonic structure of the voiced portions of speech signals. The ridges of the fundamental frequency around 200 Hz and its harmonics are easily discernible for the same first two voiced segments, previously considered.

3.3 Applications of Fundamental Frequency Estimation Techniques

Fundamental frequency estimation plays an important role in many speech related applications such as, for example, speech coding, signal processing hearing aids, glottal-synchronous speech analysis, music transcription, speaker and speech recognition, blind source separation and dereverberation techniques. A short description of these techniques is provided in the following.

3.3.1 Speech coding

Nowadays, transmitting speech signals over a mobile communication channel, is one of most common and natural activity. In mobile communica-

tions, one of the prior concern when designing new speech related applications, is the available bandwidth, which heavily affects the final systems design, in terms of feasibility, performance and cost. The main consequence of this, was that speech coding algorithms started to be developed in order to reduce the bit rate of the speech data to be transmitted or stored. Many solutions were devised in this field, all exploiting the high speech signal correlation between adjacent time samples. In fact, as the source-filter model of Figures 3.2 and 3.3 shows, the speech signal production process can be decomposed as a chain of basic blocks, each one driven by its own parameters.

These parameters, that is, voiced/unvoiced information, fundamental frequency and formant positions, are sufficient to synthesize the original speech signal and need very low bandwidth to be transmitted. A well known speech coding method is the Code Excited Linear Prediction (CELP) algorithm, which bases on LPC analysis (see Section 2.2.3). This technique is capable to compress speech signal sampled at 8 kHz with 16-bit resolution, down to $2.4 \div 4.8$ kbit/sec [95].

3.3.2 Signal processing hearing aids

Even if meaningless alone, pitch information represents an important cue to the speech message and hearing-impaired persons can take important advantages from it. In particular, there are some patients, whose auditory system capabilities are limited to an extent, that can only process a very basic audio information stream.

Hearing aids capable of transmitting to this class of patients the whole speech signal, by means of amplification or cochlea stimulation, can saturate their auditory system and be counterproductive. A correct approach in these cases is to design devices that extract the fundamental frequency from the considered speech signal and provide it to the impaired listener.

This solution, though simple, revealed to be beneficial to some patients and demonstrates the importance of pitch information for semantic disambiguation. For the same reason the output of fundamental frequency estimators is also used in automatic speech recognition systems.

3.3.3 Glottal-synchronous speech analysis

As reported in Section 3.2, to obtain a precise frequency representation of a periodic signal by means of the discrete Fourier transform, the analysis frame length has to be set to an exact multiple of the signal period. Otherwise, just an approximated representation is produced.

Knowing the speech signal f_0, permits to carry out glottal-synchronous speech analysis, which continuously update the analysis frame length to a multiple of the pitch period. This permits to achieve precise frequency representation of the speech signal and proved very useful in correctly estimating the vocal tract transfer function.

3.3.4 Music transcription

Music transcription is the act of generating a symbolic notation which represents the musical piece taken into consideration [51].

This operation was, and still is, carried out by hand, since it represents a difficult task and musical education is required for it. For these reasons, and being manual music transcription very time-consuming, several algorithms for automatic music analysis, started to appear.

To transcribe a musical piece, it is necessary to annotate the notes, the timings, the instruments playings and the pitches. In particular, polyphonic music, that is, music generated by several voices and/or instruments, is characterized by the presence of multiple pitch lines.

Given that, the algorithms which estimate the pitch of a single source,

as those presented in Chapter 2, can not be used for signals were multiple pitch streams are present.

The current approaches for multiple pitch estimation, involve complex algorithms which exploit the findings in the speech pitch estimation field and base on auditory scene analysis, trying to mimic the human auditory system. Perceptual cues are also used, exploiting spatial proximity of the events in the time and frequency domain, harmonic relationships and signal changes as onsets, offsets, amplitude and frequency modulations. But, differently from the speech processing field, where the interest for improvements in speech pitch estimation is high and often driven by economic interests, in the musical field, relative little work has been done so far.

Therefore, nowadays, there exist very few completely automatic system able to transcribe real-world music performances and usually, many restrictions have to be set on the analyzed musical piece. These restrictions are relative to the type of instruments, musical genre and maximum polyphony allowed, as well as the presence of percussive sounds or other effects.

An interesting description of f_0 based music scene description systems can be found in [39, 40].

3.3.5 Speaker recognition

Humans have the natural ability to recognize persons, whom they are acquainted with, just hearing their voice. In fact, the speech signal carries many clues which are characteristic of each individual.

These clues, can be divided into high level features, as dialect, pronunciation preferences, melody, prosodic patterns, talking style, etc. and low level features, as pitch period, formant transitions, timbre, rhythm, tone, etc.

Automatic speaker recognition systems are designed to recognize who is speaking, by means of different approaches as dynamic time warping

(DTW) techniques, statistical methods based on Hidden Markov Models (HMMs) or Gaussian mixture models (GMMs), vector quantization (VQ) and neural networks (NN) [14]. These systems can perform Speaker Identification (SI) or Speaker Verification (SV). The target of the former system is to provide an identity, chosen from a set of known speaker identities, for the unknown person that is using the system. The latter instead, has to determine whether the speaker using the system is really who she/he claims to be.

A speaker recognition system usually works in two steps: first it has to create an internal database of speaker models. Each model represents a separate individual and is defined by a set of salient features or parameters derived from her/his speech. This phase is referred to as "enrollment" of the speaker to the system or "training" phase, during which the system *learns* the speaker voice patterns. The second step occurs when the system has to recognize an end-user. At this time, the features extracted from the speech signal of the current speaker, are matched with those describing the models previously stored in the internal database. Depending on the matching result, the system provides its response.

Among the different features, pitch information was shown to be an important descriptor, on which the majority of the speaker recognition systems relies on [7]. In particular, under noisy conditions, the pitch resulted useful to detect high signal to noise ratio regions, in the speech signal spectrum. In fact, during voiced speech frames, the frequency regions occupied by the fundamental frequency and its harmonics, show higher energy than the regions laying in between. The latter have a lower signal to noise ratio and are more likely to be filled by noise components.

The security field is the main responsible for the growing interest in speaker recognition systems. Possible target applications, among others,

are phone access to banking services, secure authentication in network environments, secure transactions management.

3.3.6 Automatic Speech Recognition (ASR)

Humans to humans interactions are mostly based on vocal communication, which represents one of the most natural and quickest way to exchange information. In recent years, as a side effect of the increasing capabilities of modern computers, the speech recognition research community has focused more and more on algorithms that make easier human to computer communication. The objective of the research is to extend the human speech communication abilities to the machines, so that the interaction could result easier.

Nowadays in fact, computers and other digital devices have become essentials in many field, for their capability of fast and precise computation and of handling large amount of data. They are now present in almost all scenarios, and have radically changed the way the information is exchanged. Speech recognition systems are already employed for dictation tasks, car navigation and language learning systems as well as in the medicine, manufacturing, process control, robotics, transportation and other fields [46].

In addition, speech recognition systems can improve the quality of life of disabled persons, such as physically or visually impaired people. Voice-operated devices, for example, can help people who are physically unable to operate a keyboard or to control the environment they live in.

The study of Automatic Speech Recognition (ASR) is dated back to year 1936, when the universities and the Defense Advanced Research Project Agency (DARPA) of United States, as well as the AT&T Bell Laboratories started investigating the topic. But it was not until the early 1980's

that the ASR based technology reached the commercial market. These earliest systems had a limited vocabulary of about a thousand words and could not work in real time, usually being three times slower than humans. These limitations were mostly imputable to the lack of fast computation capabilities, as those provided by modern computers.

However, thanks to the advances in computer technology, during the past decade there was a very significant progress in this field, and many ASR based products and services, started to appear on the market. As an example, modern ASR based dictation systems are capable of accuracy[5] levels of more than 95%, with a transcription speed of more than 160 words per minutes. Also, dictation systems that can work in real-time with a 100,000-word vocabulary are not far to be achieved.

The design of modern ASR systems involves many disciplines, as pattern recognition, algorithmics, phonetics, linguistics, signal processing, information systems, formal language theory and artificial intelligence. Their general working principle can be mainly divided into a feature extraction step, which takes place before the training or recognition phase is carried out.

A Front-End (see Figure 3.7) is responsible for converting the input speech waveform to a stream of feature vectors which better represent the speech acoustic characteristics in a given domain[6]. Features are usually extracted just from speech segments which are previously detected by a voice activity detector (VAD). Non-speech segments can be used to estimate the acoustic characteristics of the underlying scenario, as background

[5]The speech recognition accuracy is defined as the Word Error Rate (WER) which represents the average number of word errors. Errors comprise the number of *substitutions* (the reference word is replaced by another word), *insertions* (a word is hypothesized that was not in the reference) and *deletions* (a word in the reference transcription is missed).

[6]An example of commonly used speech features are the Mel Frequency Cepstrum Coefficients (MFCCs).

noise and reverberation, in order to remove their detrimental effects from the extracted features.

The training phase is necessary so that the ASR system "learns" the reference patterns representing the different speech sounds (phrases, words, phonemes) that will represent the linguistic domain of the application. Speech examples, along with their exact trascription, are provided to the system during this phase, so it can estimate and store the parameters of the acoustic models that will represent such reference patterns. Common employed acoustic models base on the Hidden Markov Models (HMM).

Once the system internal parameters are set, it can be employed for speech recognition. A typical simple HMM based recognizer uses the Viterbi algorithm to find the word sequence which best matches the vector stream of speech feature (a fast sub-optimal algorithm). The search space can be represented in terms of a network in which possible sentences are modeled in terms of word sequences (following lexical rules), which in turn are composed of simpler acoustic units, as, for example, the phonemes. Finally the acoustic units are represented by Hidden Markov Models whose statistical parameters (means and variances) were estimated during the training phase [18, 83].

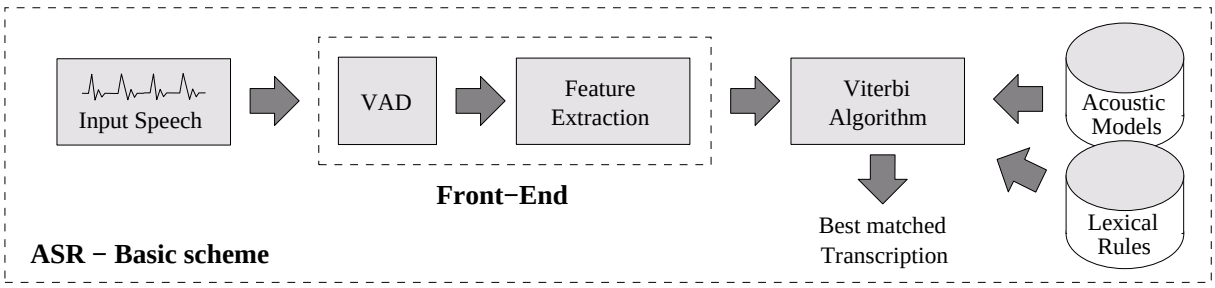

Figure 3.7: *Simplified model of an HMM based Automatic Speech Recognition system.*

The good performance achieved by modern ASR system, usually refers to systems which operate in ideal conditions, that is, on speech signals captured by close-talk microphones to avoid reverberation effects, in the

presence of a single speaker and in absence of ambient noise. Also there is a substantial performance difference between speaker dependent and speaker independent systems. The former one achieves better results because their acoustical models are trained in advance with speech data recorded from the same user that will use the system.

In any case, when the common speech recognition system have to face adverse acoustic conditions, that is, the conditions in which such systems would be most helpful, there is a considerable drop in the performance. An example of this weakness is reported in [2]: just by changing the microphone from a close-talking type to a desk-mounted type, which has the effect of introducing a small amount of reverberation, made that the speech recognition accuracy fall from 85% to 19%.

Several methods are employed to improve ASR systems performance in noisy and reverberant conditions. Common approaches consist in adapting the system acoustic models to the acoustic characteristics of the considered scenario[7] or, the other way around, to preprocess the distorted speech signal in order to remove the noisy and reverberant components [34, 46]. Both approaches aim to guarantee the best matching between the features extracted from the speech signal of the end-user and the recognizer acoustic models.

Pitch information plays an important role in the ASR process and represents one of the distinctive features useful to distinguish the different parts of an utterance. Pitch, in fact, other than providing a measure of the signal periodicity, it implicitly provides information about voicing. The type of phonation, that is, voiced or unvoiced speech, is very useful to dis-

[7]The adaptation is carried out training the recognizer using speech databases previously corrupted by noise and reverberation effects. These new datasets are usually obtained with computer simulations which can recreate the desired acoustic conditions by means of noise and reverberation models, as described in Section 3.4.3.

ambiguate between certain phonemes which are very similar among them as, for example, /z/ (voiced) and /s/ (unvoiced). Additionally, its variation with time conveys prosodic information, useful for deciding between statements, questions or other sentence patterns [99, 105].

Moreover, in case of scenarios characterized by adverse acoustic conditions (noise and reverberation), pitch related high-level features, such as voiced/unvoiced and prosodic informations, turn out to be more robust than low-level features, such as short-term informations related to the speech spectrum (e.g. MFCCs). This could be explained considering that high-level features extracted from succesive signal frames are generally correlated between each other. Pitch values, for example, vary slowly with time and can be approximated with a lognormal distribution[8] [104].

For all the above reasons, if f_0 is accurately estimated in a noisy and reverberating context, it can be used to improve the robustness of ASR systems designed to work on distant-talking scenarios [54, 75].

3.3.7 Blind Source Separation (BSS)

Blind source separation (BSS) refers to the problem of estimating original source signals from their linear mixtures. The general approach does not need a priori information about the sources or mixing process, or about the mixing matrix, sensor or source positions, and the only assumption is the statistical independence of the source signals [48]. Another important distinction is between experimental setups in which the number of sensors is equal to or greater than the numbers of sources, i.e., the determined or overdetermined case, and situations where the source signals outnumber the sensors, i.e., the underdetermined case.

Figure 3.8 shows an example of underdetermined Blind Source Separation system applied to speech signals. Three speech sources (blue, red and

[8]The logarithmic of pitch can be approximated with a Gaussian distribution.

green waveforms) are active at the same time and two microphones provide the BSS system with the captured speech mixture. The signals plotted at the right of the BSS block, represent the extracted signals, obtained from the mixture.

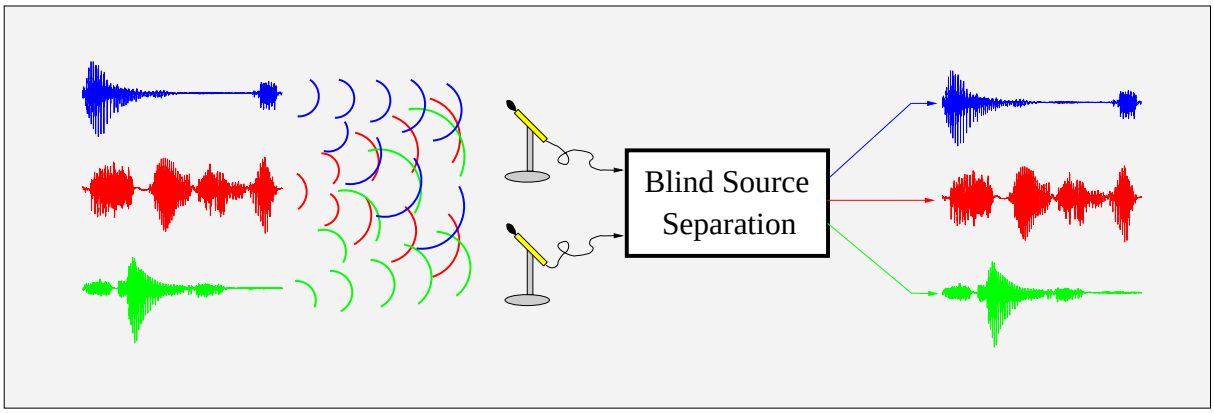

Figure 3.8: *Example of underdetermined Blind Source Separation System. The three speech sources plotted at the left are active at the same time. The BSS system capture the speech mixture by means of two microphones and recover the original individual signals.*

In order to separate speech signals when the underdetermined scenario is considered, a possible approach considers the speech signals sufficiently sparse in the time-frequency domain, i.e., they are believed to rarely overlap in this domain [12].

But when this assumption is not verified, for example when high energy regions (i.e. formants of voiced segments) that belong to different speakers overlap, each separated signal will be affected by distortion. In this case it is still possible to recover or enhance the degraded output, exploiting the fundamental frequency information. This will be shown in Chapter 6, where an underdetermined BSS system and an f_0 based post-processing scheme, which recovers and enhances the separated signals, will be described in depth.

Given their ability to distinguish each speaker contribute among several

speech sources, the BSS systems are very useful in all applications that have to cope with the cocktail party problem, i.e., several speaker talking simultaneously. Noise robust speech recognition, high-quality hands-free telecommunication systems or speech enhancement in hearing aids are, for example, some of the possible applications of these systems. Also, speech encryption applications based on BSS systems exist, where the objective is to separate speech signals that were intentionally mixed beforehand.

3.3.8 Dereverberation

A speech signal that propagates in an indoor scenario is generally smeared by the reverberation effect. One of the possible approaches to enhance the quality of the distorted signal bases on the inverse filtering approach. As will be shown in Section 3.4.3, the acoustic channel through which the sound propagates to reach an acoustic sensor, can be modeled as a linear filter. If the transfer function of this filter is known, the original non-reverberant speech signal can be recovered applying inverse filtering to the microphone reverberant acquisition.

In many practical applications the environment impulse response is not known in advance and has to be estimated from the reverberant speech signal.

Several works addressed this problem, taking into account the different statistical properties of speech signals and reverberation effects: speech signals, in fact, can be characterized by short-term features ($< 100\ ms$) while reverberation effects usually have longer duration ($500 \div 1000\ ms$). Analyzing the speech signal over both short and long segments, permits therefore to separate the contributes of the original clean speech signal from those induced by the reverberation. This approach is adopted in [13, 110], where an example of dereverberation techniques based on spectral subtraction in the linear and logarithmic domain, is given.

Also speech fundamental frequency can be used as a spectral feature to distinguish between reverberant and clean speech components. An example of a pitch based dereverberation approach is given in [50], where the Harmonicity based dEReverBeration (HERB) algorithm is presented. Basically, this algorithm processes the reverberant signal in the frequency domain and extract, by means of adaptive filtering, the harmonic components from voiced segments. These components are then used to compute the dereverberation transfer function which is finally employed to recover the clean speech message.

3.4 Noise and Reverberation

Pitch estimation algorithms performance degrades as the quality of the analyzed speech signal get worse. Most of the work done so far relates to algorithms tested on clean speech signal, that is, free from noise and reverberation, and recorded using close talk microphones, from speakers required to clearly utter some test sentences.

Thanks to the advances in the pitch estimation techniques, the results achieved in clean conditions, employing state of the art pitch extraction algorithms as, for example, those in [17, 66], are very good, getting close to 100% correct pitch estimates. Given that, the focus is now moving toward pitch estimation algorithms capable of dealing with speech signals as those audible in real-world scenarios. In these contexts, the effects of noise and reverberation are often not negligible and cause a severe performance reduction of pitch estimation systems.

Nevertheless, speech databases recorded in real noisy environments, provided with the reference pitch values for evaluating the performance of a given PDA, are completely missing. Collecting pitch labeled speech data

is an expensive and time-consuming activity, not sustainable in all laboratories.

Some research works report of tests done on small databases, locally recorded in a noisy context. But this material is not publicly distributed or not yet officially recognized by the pitch estimation research community.

Often the solution is to test one's system on noisy and/or reverberant signals obtained, as it will be shown in Section 3.4.3, from clean speech material that was opportunely preprocessed. This method has the advantage that from a single and universally acknowledged database of clean speech signals, by means of simple signal processing operations, it is possible to obtain several noisy and/or reverberant datasets.

Simulating the noisy and reverberant scenario represents thus an easy and straightforward method to test the PDA robustness. However, often the measured performance results different from that provided by the same PDA, when tested on speech signals recorded in a real reverberant and noisy environment.

Nowadays, commercial applications relying on pitch extraction algorithms and designed to work in real environments, are becoming more and more popular. For this reason the research community is starting to evaluate with a certain caution new pitch extraction algorithms, reported to improve state of the art results, but tested only on noisy and reverberant data obtained by means of computer simulations.

Even though the reverberation can be considered, to a certain extent, as a convolutional noise[9], in the following it will be described separately. The distinction used here is that convolutional noises refer to phenomena involving the presence of noise sources, while reverberation is strictly associated with signal distortion induced by the target signal itself, that is,

[9]As it will be shown, both the reverberation and convolutional noise effects can be modeled using convolutional operations.

in the context of this thesis, speech.

3.4.1 Environmental noise

In the real-world context, speech processing applications have to deal with many types of noise, which are classified using three main distinction criteria. Noise can be considered *coherent* or *diffuse*, *stationary* or *non-stationary*, *additive* or *convolutional* [77].

A noise signal is considered coherent when its components have a definable direction of propagation from the source, while it is considered diffuse when its flow of energy is uniform in all directions. Ideally, acquiring a coherent sound using microphones placed in distinct positions, will provide two signals being one, just a scaled and delayed version of the other. In reality, each sound or noise signal, when propagates, generate a sound field which is partly coherent and partly diffuse.

Stationarity instead, has to deal with the statistical properties of the noise source. When these do not change over time, the noise is said to be stationary, otherwise is considered non-stationary[10].

The most known stationary noises generated by computers are the white, pink and red noises. The former has a uniform spectral power density at all frequencies. Pink and red noises have a power spectral density proportional to $1/f$ and $1/f^2$, respectively. Pink noise, in particular, is useful in audio testing, since its energy distribution is uniform across octaves, the same scale used by the human auditory system to process sounds.

An example of the effect on pitch estimation of white noise added to a clean voiced speech signal is shown in Figure 3.9. The three panels on the

[10]In case of statistical characteristics which repeat periodically over time, the noise source is said to be cyclostationary.

left side of the figure show the original clean speech segment $x_1(n)$ (top) and signals $x_2(n)$ (middle) and $x_3(n)$ (bottom), obtained from $x_1(n)$ after adding white noise with a signal to noise ratio[11] (SNR) of 0 and -5 dB, respectively. On the right is reported the effect of the noisy signals on the Weighted autocorrelation function, which is computed for each signal shown at the left. White noise is, by definition, uncorrelated and, as reported in Section 2.3.1, the WAUTOC function is capable to detect periodic components while rejecting those uncorrelated, as white noise. This is evident from the figure in the case of SNR $= 0$ dB, where the estimated pitch period is $\tau = 197$, in accordance with the actual pitch period value.

However, for lower levels of SNR, the analyzed signal is dominated by the noise components. For signal $x_3(n)$ in the bottom panel, the pitch period was incorrectly estimated at $\tau = 104$ samples. The latter is a typical octave error, which happens when the estimated pitch period results twice or half the actual value. In this case, the noise components affected significantly the frequency region relative to the signal fundamental frequency, and the first harmonic was wrongly detected as f_0.

In Figure 3.10 another example of pitch estimation on a noisy speech signal is shown. The Cumulative Mean Normalized Difference Function of YIN algorithm (see Section 2.3.1) is applied to the same voiced speech segment used in Figure 3.9.

The robustness to noise of this state of the art pitch extraction algorithm is evident from the figure. As shown, the correct estimate is still provided when white noise is added to the clean signal, with a signal to noise ratio of -5 dB (third couple of panels). Decreasing the SNR to -10 dB, makes the speech signal (bottom left panel) become practically unintelligible and

[11] Signal to noise ratio, expressed in decibels, is given by SNR(dB) $= 10 \log_{10}\{P_s/P_n\}$, where P_s and P_n are the average power of the signal and of the noise, respectively.

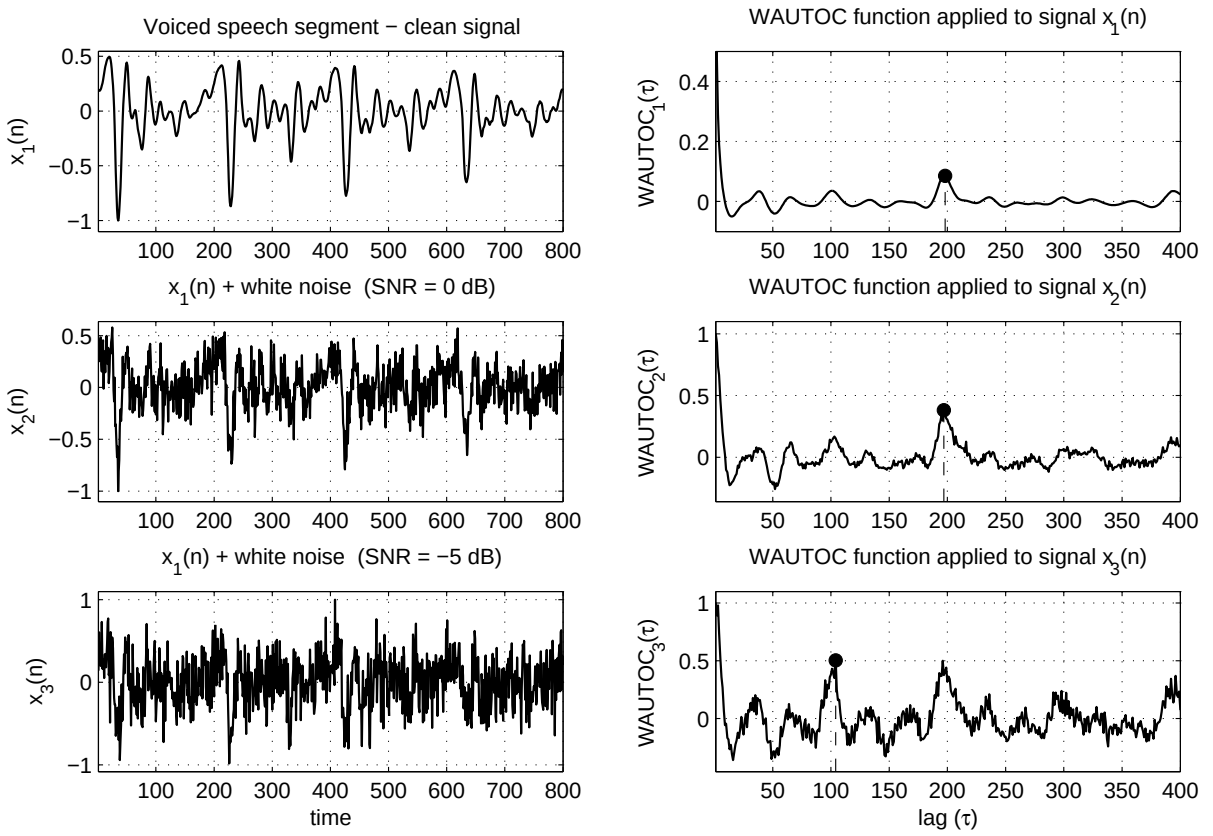

Figure 3.9: *Weighted autocorrelation function computed on noisy signals. Left panels show, from the top, a clean voiced speech signal, the same signal with white noise added with a signal to noise ratio of* 0*, and* -5 *dB, respectively. The WAUTOC function is computed for each speech signal and plotted in the right panels. The estimated pitch period values are* $\tau = 198,\ 197$ *and* 104 *samples, respectively.*

make, consequently, the CMNDF turn very noisy, providing thus the wrong pitch estimate at $\tau = 304$ samples.

Artificial noises, as that used in the above described experiments, is just an approximation of the actual acoustic interferences, characteristic of real world scenarios. In fact, when real world noises are considered instead, speech processing systems have to cope with an infinite variety of noise types. Real stationary noises can be: those produced by the fans of the heating, ventilating or air conditioning systems; the hum produced by the

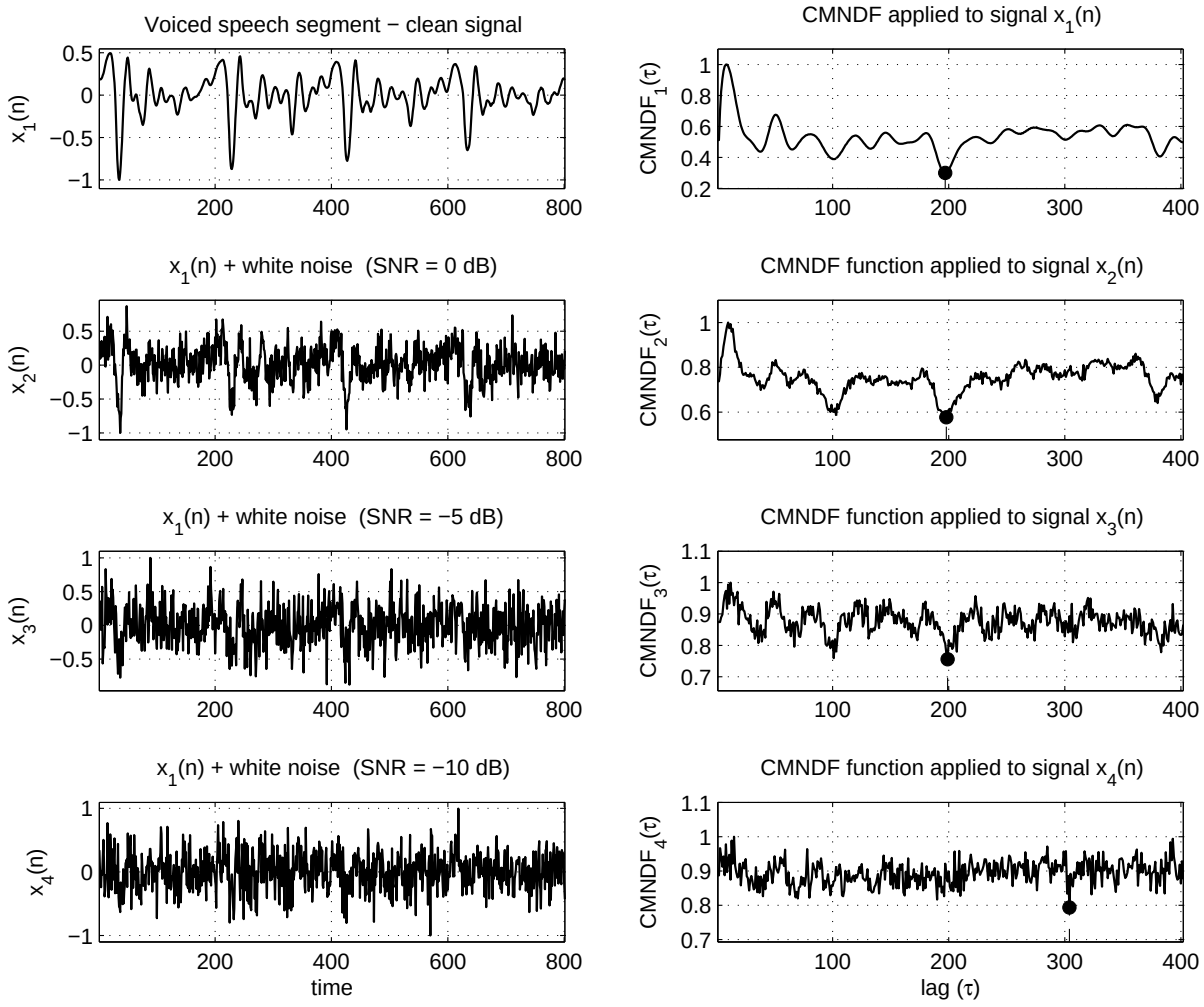

Figure 3.10: *Cumulative Mean Normalized Difference Function computed on noisy signals. Left panels show, from the top, a clean voiced speech signal, the same signal with white noise added with a signal to noise ratio of* 0, -5, *and* -10 *dB, respectively. The CMNDF is computed for each speech signal and plotted in the right panels. The estimated pitch period values are* $\tau = 197,\ 198,\ 199,$ *and* 304 *samples, respectively.*

engine and tires of a car running at a constant speed; noise produced by uniformly vibrating plant machineries or jet engine noise audible in the cockpit[12].

[12]Unfortunately, the military interest in speech processing applications is very high. It is quite common, recently, to come up with articles relating about robust speech recognition in presence of tank, jet cockpit, machine gun or helicopter rotor thickness noises.

Among non-stationary noises, there are all burst-like noises, as coughs, door slams, phone rings, etc., or as speech babble[13] and helicopter rotor thickness noise[14]. These kind of noises have to be treated differently than the stationary counterpart. In fact, while the latter can be assumed to have constant statistical properties which can be estimated during the speech processing task, non-stationary noises are unpredictable and of short duration, thus making their modeling more difficult.

An example of real world noise is presented in Figure 3.11, where noise recorded in a train coach is added to a clean speech signal recorded from a female speaker. In the first two panels at the top, the spectrogram and the waveform of the clean signal are shown. The panels below instead, are the spectrogram and waveform of the clean speech signal with noise added with an SNR of 10 dB. It is evident, in the latter case, how noise components are spread almost uniformly over the whole spectrogram, and only the ridges relative to the fundamental frequency and its harmonics are still discernible.

The last distinction refers to the way the noise and the target signal interact: additive noise is supposed to add linearly to the target signal in the time domain, while convolutional noise is more related to the room acoustical properties. The former represents ideal acoustical conditions, which are never met in reality, but is useful for system analysis purposes. The latter is a closer representation of what actually happens in any real context, even though it does not take into account possible non-linear acoustic effects.

[13]Speech babble is audible during, for example, a cocktail party and it is one of the most difficult noise that pitch extractor algorithms have to deal with. In fact, depending on its intensity, several additional pitches and formants, belonging to the different voices in the babble, can add to the target speech signal.

[14]See Note 12.

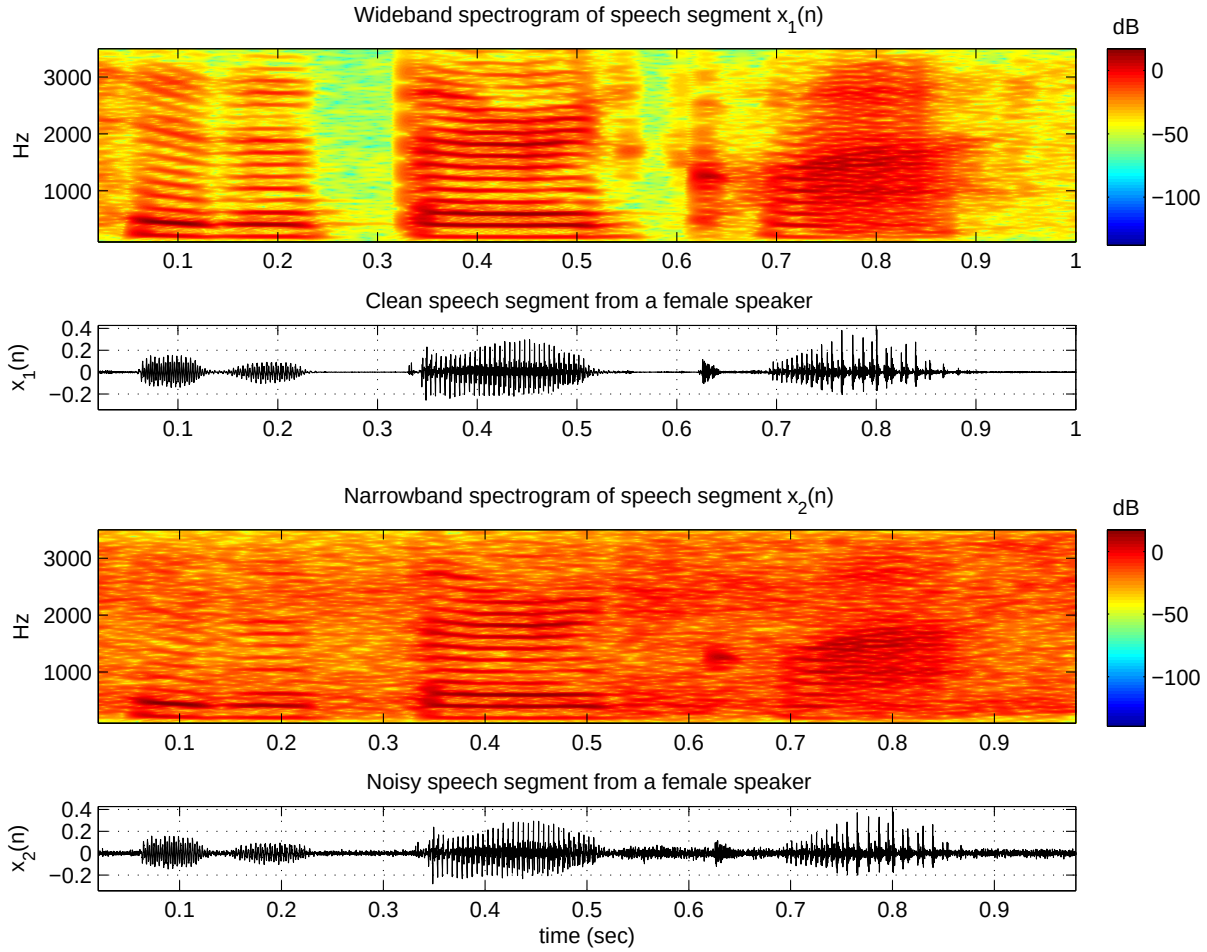

Figure 3.11: *Effect of train coach noise on a segment of speech signal. The two panels at the top show the spectrogram and the waveform of a clean speech signal recorded from a female speaker. In the two panels at the bottom, noise recorded in a train coach was added to the clean speech signal with an SNR of* 10 *dB.*

3.4.2 Reverberation

Whenever a pitch extractor algorithm has to deal with speech signals captured by a microphone positioned far from the talker, its performance decreases. Compared to the close-talk recording case, when the speech sound has to propagate through the environment to reach a distant acoustic sensor, is more subject to noise and reverberation effects.

The reverberation word has its root in Latin, meaning to “beat back

again". In other words, when a sound is produced in an enclosed space, multiple reflections originate from the sound-reflecting surfaces as walls, floor, ceiling and the various objects present, and mix together creating reverberation.

One way to quantify the reverberation effect is to measure the *reverberation time* T_{60} or RT, that is, the time it takes for the sound pressure level to decay 60 decibels, after the sound source has stopped. This variable represents just a global quantitative criterion[15] and different scenarios, with the same reverberation time, can have completely different acoustic. In fact, the latter is determined by the ambient shape and size, as well as by the materials used in its construction and the number and type of objects or persons present [52].

Large chambers as, for example, cathedrals or halls, are places where the reverberation can be clearly heard, although any indoor space is affected by this phenomenon. Humans are used to it and would find it strange to hear speech or music as if they were in an anechoic chamber or in a wide open field. Our auditory system is able to clearly decode a speech stream in a reverberant ambient with $T_{60} = 0.5 \div 1.5$ seconds, which are quite common values for lecture and conference rooms.

A common procedure to model the reverberation effects, is to estimate the ambient impulse response, or transfer function, under the hypothesis that linear effects are predominant during reverberation. Non-linear effects cannot be modeled by the impulse response method which results however a good approximation of the environment acoustic characteristics. Considering a sound source (or speaker) signal $x(n)$ in a reverberant scenario and a distant microphone (or listener) signal $y(n)$, the linear relation which takes into account the environmental acoustic effects can be written as

[15] Other measures are the *Early Decay Time* (EDT), *clarity*, *definition*, *Initial Time Delay Gap* (ITDG), *Intimacy ITDG*, *texture*, *spaciousness*, *diffusion*, ... [9].

$$y(n) = h(n) * x(n), \tag{3.6}$$

where $h(n)$ is the ambient impulse response to be estimated. A simple and effective procedure to estimate $h(n)$ consists in reproducing a chirp-like[16] signal $p(n)$ with a loudspeaker placed in the same position of the sound source. The microphone output becomes thus

$$y(n) = h(n) * p(n). \tag{3.7}$$

As pointed out in [107], the chirp-like signals with a flat overall power spectrum have the important property that their autocorrelation is an almost perfect Dirac delta function. As a consequence, the sequence $y(n)$ of Equation 3.7 can be easily deconvolved by simply cross-correlating it with the original sequence $p(n)$. The result, apart from the contribution of the loudspeaker frequency response, is the impulse response $h(n)$ of the considered acoustic channel.

In case several reverberant speech dataset are needed, various microphones can be placed to record the chirp-like signal at the same time. Once the impulse response relative to the acoustic channel from the loudspeaker to each microphone is estimated, it can be used to convolve the signals from any clean speech database available.

In Figure 3.12, the top panel shows a segment of voiced speech signal recorded by a close-talk microphone. Middle panel shows an example of room impulse response. The first peak from the left, occurring at about 35 ms, determines the delay with which the direct sound propagates from the source position to the particular point in the room, where the impulse response was estimated. The successive peaks, as that occurring at about 42 ms, are due to early reflections. These are directional reflections gener-

[16] A linear swept-frequency cosine signal.

ally well defined and are directly related to the shape and size of the room, as well as to the furniture and to wall surface materials. The tail of the impulse response is formed by diffuse reverberation, or late reflections, which are more random and difficult to relate to the physical characteristics of the room.

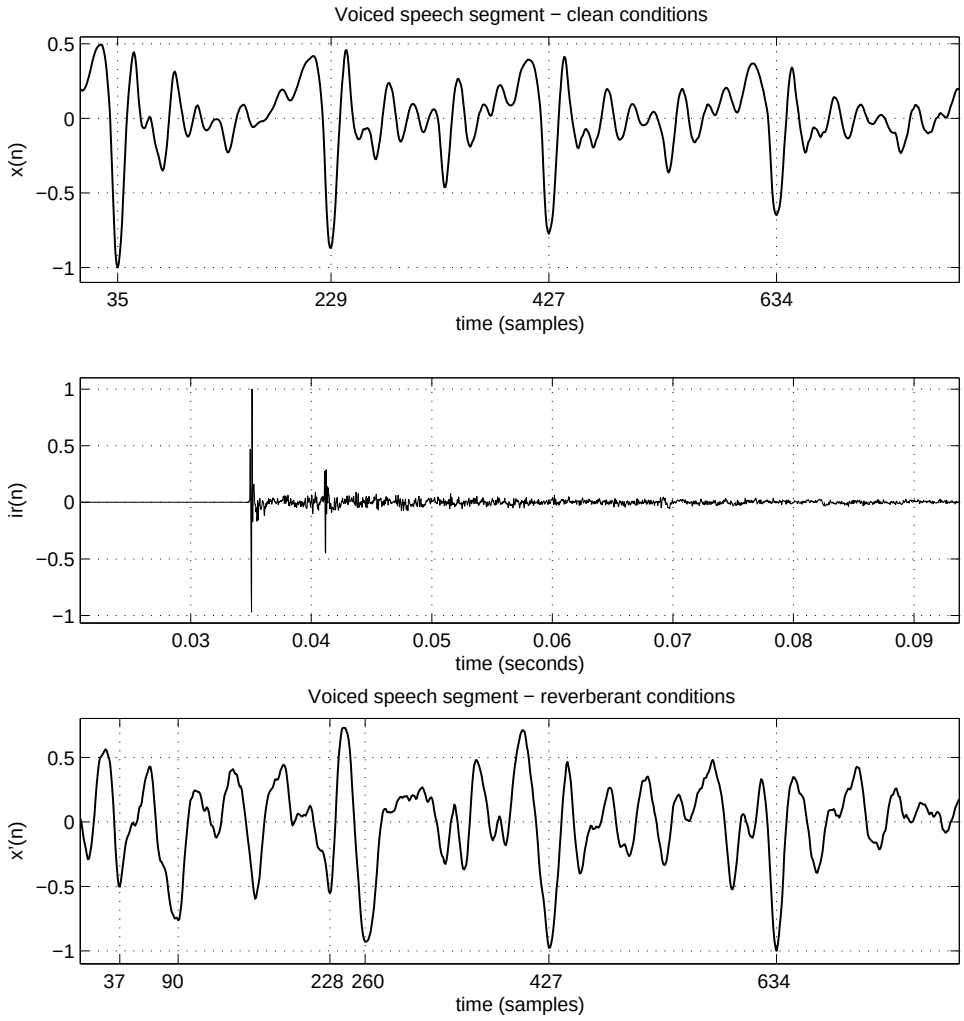

Figure 3.12: *Top panel: voiced speech signal. Middle panel: reverberant room impulse response. Bottom Panel: reverberant voiced speech signal obtained as the convolution of clean signal and room impulse response.*

The bottom panel shows the result of the convolution of the close-talk speech signal (top panel) with the room impulse response (middle panel).

To show the effect of reverberation on each pitch epoch, the result has been time-aligned with the plot of the top panel. In the clean speech signal, the glottal closure instants are clearly visible at lag values $\tau = 35,\ 229,\ 427$ and 634. The reverberation effect changes significantly the waveform and, as evident from the figure, only the local minima at lag values $\tau = 427$ and 634 are still unambiguously detectable. Instead the negative peaks at $\tau = 37$, and 228, correspondent to peaks of the clean segment at $\tau = 35$, and 229, are exceeded by those at $\tau = 90$, and 260.

This is the reason why reverberation is regarded as a convolutive noise, that degrades the speech quality and intelligibility. Examples of the effects of this speech quality degradation are visible in Figure 3.13 and 3.14, where the weighted autocorrelation (WAUTOC) and YIN algorithms have been applied to the reverberant speech segment showed in the bottom panel of Figure 3.12. In Figure 3.13, the WAUTOC function is calculated, first on the clean speech signal, providing a pitch estimate of $\tau = 198$ samples, then on the reverberant signal, providing the wrong pitch estimate $\tau = 340$ samples.

It is interesting to note that not even a peak is present in the bottom right panel around $\tau = 200$. The reverberation effect has completely canceled out the period component relative to the fundamental frequency. Nevertheless, the reverberant speech signal is perfectly decoded by the human auditory system resulting thus perfectly intelligible.

The same considerations are applicable considering Figure 3.14, where the Cumulative Mean Normalized Difference Function (CMNDF) is computed. Still the estimation of the pitch period on the reverberant signal fails, providing $\tau = 378$ samples, while the correct one is around $\tau = 197$ samples, value provided by the CMNDF on the clean speech segment.

Beside all the above considerations, it has to be pointed out that, con-

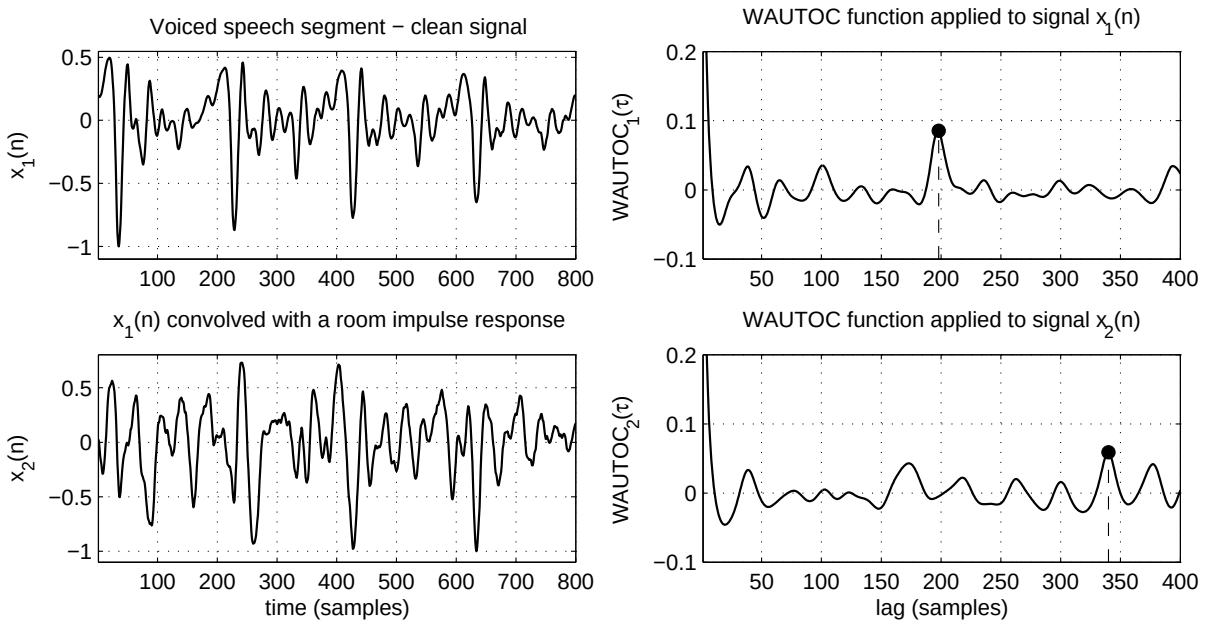

Figure 3.13: *Weighted autocorrelation function computed on a reverberant signal. Left panels show, from the top, a clean voiced speech signal and its reverberant version, respectively. The WAUTOC function is computed for each speech signal and plotted in the right panels. The estimated pitch period values are* $\tau = 198$ *and* 340 *samples, respectively.*

volving the room impulse responses with clean speech signals, produces just an approximation of the reverberation effects. Reverberation, in fact, includes also non linear phenomena which can not be modeled by this method. To test a system in a real reverberant scenario, it will be thus preferable to acquire the speech data recording it directly from the ambient where the talker is speaking, or where a loudspeaker is used to reproduce a given database. This operation, though being more time-consuming and less versatile, compared to the room impulse responses method, provides the closest test conditions to the real environment. The latter approach, that is, the direct acquisition of reverberant speech data, has been used for testing the pitch extractor algorithms that are proposed in this thesis. The databases resulted from the recordings that have been carried out in different real noisy and reverberant scenarios, will be described in

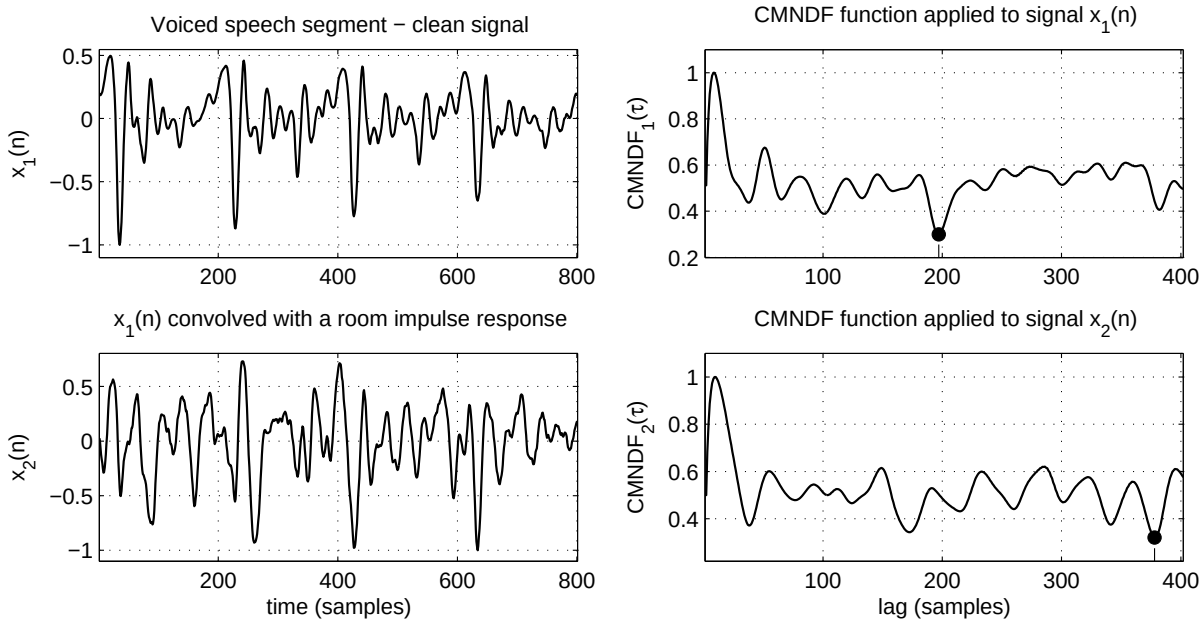

Figure 3.14: *Cumulative Mean Normalized Difference Function computed on a reverberant signal. Left panels show, from the top, a clean voiced speech signal and its reverberant version, respectively. The CMNDF is computed for each speech signal and plotted in the right panels. The estimated pitch period values are* $\tau = 197$ *and* 378 *samples, respectively.*

Chapter 5.

3.4.3 Modeling noise and reverberation

When a PDA has to be tested on real-world speech data, that is, on data affected by environmental noise and reverberation, a speech dataset reflecting such conditions is needed, along with the reference pitch values necessary for performance evaluation. To obtain the speech dataset for the target scenario a solution is to record a talker (or several talkers) in the considered environment. Once the data is collected, the pitch reference values, needed to evaluate a PDA performance, have to be derived. Since this results in a time-consuming procedure, an alternative is to reproduce in the selected scenario an already pitch-labeled speech database by means of a loudspeaker. The latter procedure permits, in fact, to obtain a new

speech database with relative little effort and the original pitch references can be reused to evaluate a given PDA.

When one of the above methods is applied, several microphones are usually employed, placed in different environment positions. This permits to obtain several versions of the original database, each characterized by different noisy and reverberant conditions. Also the whole set of microphone outputs can be needed, in case a PDA with multi-microphone processing capabilities has to be tested. The speaker or loudspeaker position is usually fixed during the recordings, although when spontaneous speech is needed the talker is not constrained to stand in a given position. An example of spontaneous speech used to test PDAs performance in this thesis is represented by the seminar sessions recordings described in Section 5.3.

An alternative to the above approaches, is to derive a mathematical model reflecting the acoustic characteristics of the target environment. Once the model is derived, it can be used to obtain speech data with given reverberant and noisy characteristics, by means of computer simulations.

If only additive noise and reverberation are initially considered by the model, a speech signal acquired by the i-th acoustic sensor can be obtained, to a first approximation, as

$$y_i(n) = x(n) * h_i(n) + r_i(n), \tag{3.8}$$

where $x(n)$, $h_i(n)$ and $r_i(n)$ represent the speech signal, the acoustic impulse response between the speech source and the i-th microphone, and the noise signal affecting the considered sensor, respectively. $x(n)$ represents here the speech signal recorded by a close-talk microphone, which is convolved with the room impulse response $h_i(n)$ to obtain its reverberant version, as previously introduced in Section 3.4.2.

Considering the additive term $r(n)$ in Equation 3.8, this model permits to artificially add noise in the time domain using random signal generators that can simulate noisy patterns with a given distribution. Or, alternatively, it can be added using noise recorded from real environments. The result provided by this model is not realistic since it considers the noise effects as additive and independent from the environment acoustic.

A more precise model, which takes into account both convolutional and additive noise effects, can thus be written as

$$y_i(n) = x(n) * h_i(n) + \sum_{k=1}^{K} r_k(n) * \hat{h}_{ki}(n) + r'_i(n), \tag{3.9}$$

where the new variables K and $\hat{h}_{ki}(n)$, indicate the number of noise sources, each located in a known position, and the impulse response between the k-th noise source $r_k(n)$ and the i -th microphone, respectively. The term $r'_i(n)$ can still be used to model possible additive noise components.

When the pitch extractor algorithm has to be tested on close-talk signals, the above described models can be used setting the term $h_i(n)$ in Equation 3.8 and 3.9, so that it introduces just the delay with which the speech signal reaches the acoustic sensor[17], and a possible attenuation. Whenever the reverberation effects have to be modeled instead, the term $h_i(n)$ is set to the value of the room impulse response, measured as described in Section 3.4.2.

[17] A delayed version of a discrete signal $x(n)$ is obtained convolving it with the delta of Kronecker function, centered at the time sample corresponding to the propagation time.

Chapter 4

Multi-Microphone Approach

Nowadays, many speech processing systems are required to work in contexts where hand-held or headset microphones represent unfeasible solutions. However, the use of a distant omnidirectional microphone would provide poor speech quality while, employing a directional one, would constrain the talker to keep a specific position, as well as a specific direction. This is mainly due to the fact that any far microphone interaction is strongly affected by the ambient noises and reverberation.

To overcome all these limitations, on the one hand new efficient signal processing methods, as noise reduction and dereverberation techniques (see Chapter 3), are being investigated to improve the signal speech quality. On the other hand, the focus has gradually moved toward the microphones type, quality, number and position.

The reason for this lies in the fact that, a signal propagating in a reverberant and noisy ambient, is differently distorted depending on the location where it is captured from. If several microphones placed in different positions are used, different versions of the same speech source will be available, thus providing information redundancy, that can be exploited for speech enhancement.

This line of reasoning led to the introduction of the microphone array,

which can be thought of as a set of microphones, operating simultaneously. The microphone array sensors are usually omnidirectional and are generally arranged along one dimension (linear array), equally spaced between each other. Other microphones spatial configurations exist, such as the harmonic arrays, consisting of a distinct sub-array for each frequency band, or the two or three-dimensional microphone arrays, which have the microphones distributed on a surface or occupying a volume in the space, respectively.

All the different spatial configurations of microphone arrays, are usually driven by a common processing scheme, based on beamforming algorithms. These algorithms are used to form a directivity pattern, used to attenuate the contributes proceeding from unwanted directions while, at the same time, enhancing the desired signal propagating from a particular direction.

The most common beamforming techniques, employed to steer the array toward the talker position, are the Delay and Sum (DS) and the Matched Filtering (MF) algorithms [29, 75].

The former is a simpler approach and is mainly useful to compensate for diffuse additive noise, while the latter demonstrated to be efficient also when dealing with convolutional noise, such as reverberation. When microphone arrays are used, the talker position is continuously estimated and the array steering angle consequently updated. This allows the speaker to change her/his position while talking, even though within a range of a few meters in front of the array.

This technique proved to be very effective for speech enhancement and ASR in noisy and reverberant conditions [58, 76], but very little work has explicitly been done on microphone array based pitch estimation.

However, several pitch extraction algorithms have been tested on reverberant and noisy speech signals and the results, reported in Chapter 5, show that the best pitch estimation performance is achieved with close-

talk speech signals. Reverberated and noisy signals instead, recorded in different environments and conditions, caused performance degradation. As a consequence of this, it can be stated that any speech processing technique aimed to the enhancement of speech quality, as the microphone array approach, will result beneficial for pitch estimation.

Microphone arrays are already employed by several current commercial products based on speech processing. One of these is speaker localization, which exploits the spatial localization capability of the beamforming techniques, to estimate the target speaker Direction Of Arrival (DOA) and to track her/his position.

Nevertheless, even though the use of microphone arrays guarantees more spatial freedom to the end-user of such speech applications, their use still implies a coverage limitation, that is, the speaker has to interact with the device within a certain distance range and has to continuously face toward it.

To overcome these limitations, so that the end-user of a speech processing systems is allowed to move freely in the space, the *Distributed Microphone Network* is introduced in Section 4.1. In this new context, the multi-channel extensions of state of the art pitch extraction algorithms are reported and a new approach, the *Multi-microphone Periodicity Function* (MPF) is described.

4.1 Distributed Microphone Network

One of the scenarios on which the interest of the speech processing community has recently moved on, is the "meeting room" context, where several talkers are involved in the speech recognition process. In this context, a uniform coverage must be guaranteed so that each speaker position can be estimated and tracked. Along with distant ASR, this new environment

introduces the concept of "ambient intelligence", realized through a wide usage of sensors (cameras, microphones, etc.) which are connected through a computer network that fade in the background. The aim of this setup is to create a smart innovative environment, where computer services are delivered to people in an implicit, indirect and unobtrusive way.

The scenario results thus quite complex and involves several research disciplines that have to interact together. Audio-visual person localization, tracking and identification, voice and events acoustic detection, emotion recognition as well as far-field automatic speech recognition and others techniques have to be integrated in a unique framework.

An example of such a scenario, is the CHIL[1] room, where several pitch estimation experiments have been carried out. The CHIL room description as well as the results obtained from such experiments are reported in Chapter 5.

The concept of *Distributed Microphone Network* (DMN) is strictly related with this framework, and refers to a generic set of microphones localized in space without any specific geometry. The microphone outputs are connected to a recording and computing system, that will ensure a sample-level synchronous processing of the corresponding signals. This scheme has been devised so that any acoustic ambient can be fully covered by sensors, without being forced to employ expensive, and often difficult to place microphone arrays. The latter can still be employed and make part of the final DMN though, which will also include single microphones as well as microphone clusters.

In this case, the relaxation of the geometrical constrain comes at the expense of the applicability of the beamforming algorithms. In fact, in mi-

[1]The Computers In the Human Interaction Loop (CHIL) project, is an Integrated Project under the European Commission's Sixth Framework Program.

crophone arrays, the inter-microphone distance represents a key element in order to avoid the so-called spatial aliasing effect. If this distance does not fulfill specific requirements[2], spatial aliasing takes place. In this case, grating lobes are introduced at higher frequencies, that is, the array will pick-up interfering signals or reverberation components from directions other than the desired one. This implies that the DMN can not be considered as a large or extended microphone array, since the microphones inter-distance can even be in the meters range. Such a multi-channel distribution would introduce spatial aliasing at frequencies of interest for speech analysis and recognition, if combined with traditional beamforming techniques. To confirm the latter statement, an experiment described in [6] reports the detrimental use of the resulting delay-and-sum "beamformed" signal proceeding from a DMN.

A pitch estimation method, which exploits the information redundancy across all the Distributed Microphone Network channels, avoiding the space aliasing problems, is described in next sections. Section 4.1.1 and 4.1.2 describe the multi-microphone extensions of the YIN and WAUTOC algorithms presented in Chapter 2, respectively. Section 4.1.3 is dedicated to the Multi-microphone Periodicity Function (MPF), which has been designed to further improve the results obtained from the previous approaches. The results obtained with the three approaches will be presented and discussed in Chapter 5.

[2]The maximum inter-microphone distance d allowed for a linear array, is given by $d \leq \frac{c}{2f_{\max}}$, where $c \equiv 330.7\ m/s$ is the sound speed, and $f_{\max}$ is the maximum frequency present in the signal. For a 16 kHz sampled speech signal, considering $f_{\max} = 8\ kHz$, it results approximatively $d \leq 2\ cm$.

4.1.1 Multi-microphone WAUTOC

The weighted autocorrelation (WAUTOC) algorithm described in Section 2.3.1 consists of an autocorrelation function weighted by the reciprocal of the Average Magnitude Difference Function (AMDF). The advantage of using this function is that the autocorrelation and the AMDF functions have a maximum and a minimum, respectively, at the lag value corresponding to the period of the analyzed signal. Their response to non periodic components though, is not correlated as it was for periodic components, making thus the WAUTOC function more robust to uncorrelated noise.

$$\text{wautoc}_i(\tau, q) = \frac{\sum_{n=0}^{N-1} [x_i(q+n)\ w(n)] \cdot [x_i(q+n+\tau)\ w(n+\tau)]}{\epsilon + \sum_{n=q}^{q+N-1} |x_i(n) - x_i(n+\tau)|} \tag{4.1}$$

The resulting WAUTOC function, rewritten for the i-th channel in Equation 4.1, exploits thus this common behaviour to provide a more robust pitch estimate. For this reason and for its straightforward extendibility to the multi-microphone context [6, 31], it is used here for comparison purposes.

Equation 4.2, represents the multi-microphone version of WAUTOC,

$$\text{wautoc}_M(\tau, q) = \sum_{i=1}^{M} w_i \cdot \text{wautoc}_i(\tau, q) \tag{4.2}$$

where M denotes the number of microphone of the Distributed Microphone Network, and the coefficients w_i have been introduced to represent the reliability of each channel, which may depend on the speaker position and head orientation. Quantifying the channels reliability is not a trivial

task. In Section 4.1.3 describes the Multi-microphone Periodicity Function, which estimates the channels reliability to provide a better pitch estimate, most of all, when few microphones in the area covered by a microphone network are strongly affected by noise. Considering instead the case where the speaker is allowed to move in the scenario, a more complex weighting scheme, based on the output of a speaker localization device, would be necessary. In the following the constant value $\frac{1}{M}$ is assigned to each w_i in Equation 4.2.

An example of the advantages provided by the multi-microphone version of the WAUTOC function is given in Figure 4.1.

A segment of clean voiced speech $x(n)$ from a male speaker and its reverberant version $x_{\text{mic}}(n)$, are shown in the top and middle panels, respectively. The microphone-speaker distance was about 3 meters, and the ambient reverberation effects are clearly noticeable if the two waveforms are compared. In fact, the glottal closure instants occurring at samples 106, 231, 353, 476, 594, and 712 are evident in $x(n)$, while some of them are misplaced in $x_{\text{mic}}(n)$, and others are even completely missing. The consequences of the introduced distortion reflects on the pitch estimation performance of the weighted autocorrelation algorithm, as shown in the bottom panel. The black line is the WAUTOC (Equation. 2.17) computed on the clean signal $x(n)$ and its maximum non zero peak at $\tau = 121$ samples represents a good average estimate of the several pitch periods included in the analysis frame. The fundamental frequency estimate provided, considering the signal sampling frequency of $f_s = 20\ kHz$, is thus $f_0 \approx 165\ Hz$.

The blue line is the WAUTOC (Equation. 4.1) computed on the reverberant speech signal $x_{\text{mic}}(n)$. The maximum non zero peak occurs at $\tau = 179$ and will determine an estimated fundamental frequency of about

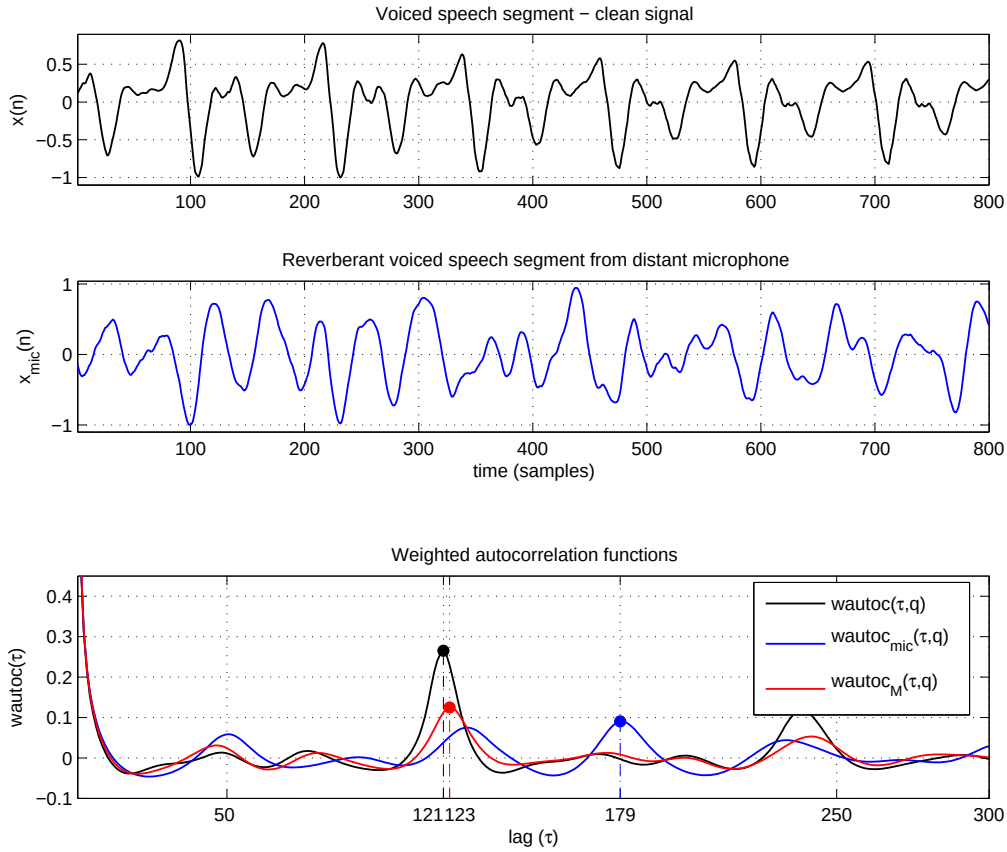

Figure 4.1: *A clean voiced speech signal and its reverberant version are shown in the top and middle panels, respectively. The bottom panel shows the WAUTOC computed on: the clean speech signal (black line), providing the correct pitch estimate* $\tau = 121$*; the reverberant speech signal (blue line), providing the wrong estimate* $\tau = 179$*; ten outputs of a Distributed Microphone Network (red line), providing, within a small error, the correct estimate* $\tau = 123$.

$\frac{2}{3}f_0$. This is due to the regular presence of secondary peaks in the speech signal, as those occurring at samples 27, 155, 280, etc., which mix in the reverberant signal those produced by the glottal closure instants.

To plot the red line, a Distributed Microphone Network consisting of 10 microphones, was used. The speech reverberant outputs $x_i(n)$, $1 \leq i \leq 10$ were used in the computation of the multi-microphone WAUTOC (Equation. 4.2), which peaks at $\tau = 123$. The correct pitch period value is thus obtained despite the distorted speech signals used. The small error in

the estimate is not considered a serious issue here since, as stated in [17], if an initial estimate is correct to within 20% respect the actual one, several techniques are available to refine it.

4.1.2 Multi-microphone YIN

The *YIN* algorithm was introduced in Section 2.3.1 as one of the state of the art among the pitch detection algorithms [17]. For this reason it was chosen here to derive a multichannel version, suitable to work with inputs provided by a DMN.

Given a speech signal $x_i(n)$ captured by the i-th microphone, and recalling the difference function $d(\tau, q)$ of Equation 2.18, a channel dependent difference function can be written as,

$$d_i(\tau, q) = \sum_{n=0}^{N-1} [x_i(q+n) - x_i(q+n+\tau)]^2, \tag{4.3}$$

which is computed for each speech frame of length N samples, starting from sample q. This function is based on the autocorrelation function and assumes a local minimum value for a lag value τ corresponding to the pitch period of the analyzed signal. From this, the YIN authors derived the *Cumulative Mean Normalized Difference Function* shown in Equation 2.20 and quoted below to include the channel dependency.

$$d'_i(\tau, q) = \begin{cases} 1, & \tau = 0, \\ \dfrac{d_i(\tau, q)}{(1/\tau)\sum_{j=1}^{\tau} d_i(j, q)}, & \text{otherwise.} \end{cases} \tag{4.4}$$

The main reason for deriving Equation 4.4 is twofold: on the one hand it does not have a dip in correspondence of the zero lag ($\tau = 0$), as Equation 4.3 does. This implies that no limit in the search range of τ is needed. On the other hand, it provides a normalized function to which a threshold

can be applied to avoid subharmonic error due to other dips, deeper than that relative to the pitch period. Normalization is also exploited by the YIN algorithm to apply post-processing in a later step so that pitch estimation errors are further reduced.

Given a Distributed Microphone Network providing M synchronous versions of a speech event, the multi-microphone *YIN* version is derived here by normalizing the difference function computed for each channel, $d_i(\tau)$, and averaging then over all channels

$$d_M(\tau) = \frac{1}{M} \sum_{i=1}^{M} \frac{d_i(\tau)}{\max_{\tau}\{d_i(\tau)\}}. \tag{4.5}$$

Equation 4.5 is then used to derive the multi-microphone cumulative mean normalized difference function $d''(\tau)$:

$$d''(\tau) = \begin{cases} 1, & \text{if } \tau = 0, \\ d_M(\tau)/[(1/\tau)\sum_{j=1}^{\tau} d_M(j)], & \text{otherwise,} \end{cases} \tag{4.6}$$

which will be used in the original YIN algorithm framework instead of Equation 2.20.

Other alternatives had been explored, as for instance averaging the cumulative mean normalized difference function in Equation 4.4, rather than the difference function of Equation 4.3. In the experiments the approach based on the latter equation gave the best performance and is therefore used in the following experiments.

The here proposed extension of YIN to the multi-microphone case does not represent an ultimate best YIN-based solution to the given problem, since the whole algorithm should be taken into consideration in all its parts. For instance, a specific work should be conducted to check if a

more effective post-processing can be conceived in this case[3]. Nevertheless, Equation 4.6 demonstrated to be a plausible derivation and is used here just for comparison purposes.

A justification for the here proposed multi-microphone version of YIN is shown in Figure 4.2. A segment of clean voiced speech $x(n)$ from a male speaker and its reverberant version $x_{\text{mic}}(n)$, are shown in the top and middle panels, respectively. The same considerations discussed in Section 4.1.1, regarding the reverberation effects, apply here too. In fact, the test signals $x(n)$, $x_{\text{mic}}(n)$ and those provided by the Distributed Microphone Network used to plot the Figure 4.1 and 4.2 are the same.

In the bottom panel, the behaviour of the YIN algorithm to the different test conditions is shown. The black line is the CMNDF (Equation. 2.20) computed on the clean signal $x(n)$ and its negative peak at $\tau = 122$ samples represents a good estimate of the slowly varying pitch period comprised in the analysis frame. The estimated f_0, considering a sampling frequency of $f_s = 20\ kHz$, is thus $f_0 \approx 164\ Hz$.

The blue line is the CMNDF (Equation. 4.4) computed on the reverberant speech signal $x_{\text{mic}}(n)$. The negative peak occurs at $\tau = 181$ and, as explained in Section 4.1.1, this is due to the presence in $x_{\text{mic}}(n)$ of secondary peaks mixing with those relative to the glottal closure instants. This is a common undesirable consequence of the reverberation effect which makes that, as in the current case, the pitch estimate results wrong.

To show the effectiveness of the multi-microphone CMNDF (Equation. 4.6), ten speech reverberant outputs $x_i(n),\ 1 \leq i \leq 10$ were used, provided by a Distributed Microphone Network. This resulted in a minimum peak at $\tau = 125$, which reflects the correct pitch estimate. As previously stated, small deviation from the actual pitch value are acceptable in such reverberant conditions, since several techniques can be used to refine it [17].

[3]See details on the various steps of the algorithm in [17].

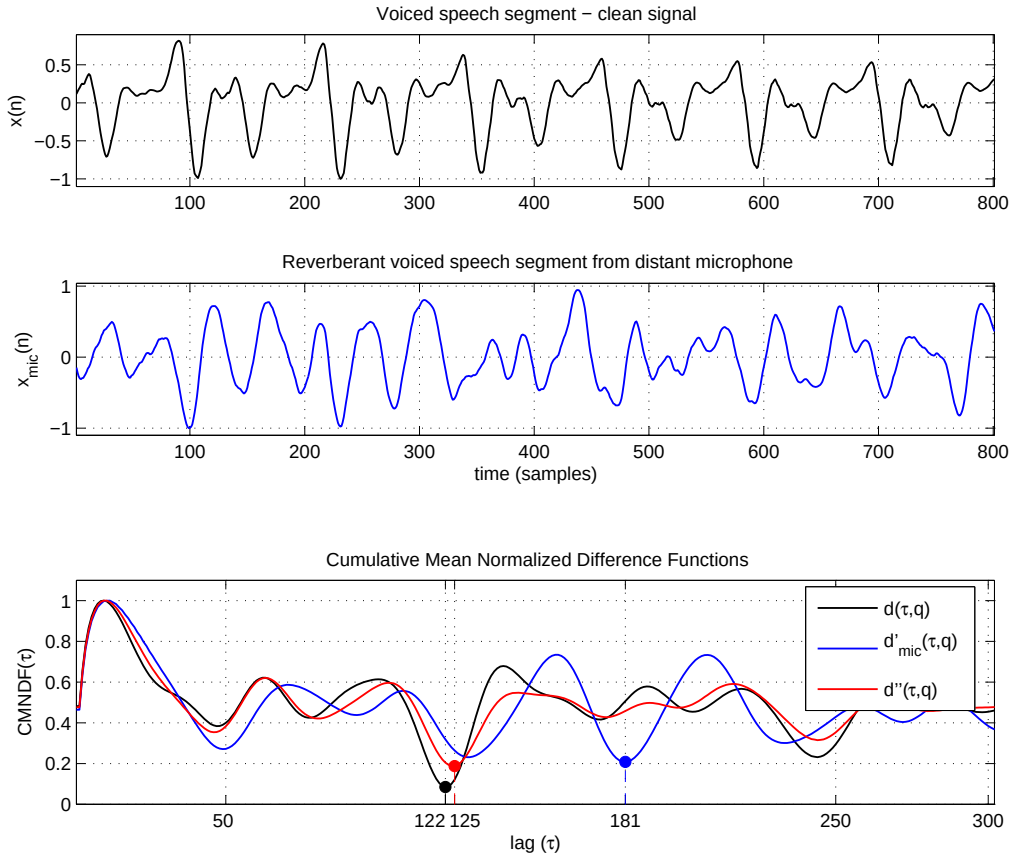

Figure 4.2: *A clean voiced speech signal and its reverberant version are shown in the top and middle panels, respectively. The bottom panel shows the CMNDF computed on: the clean speech signal (black line), providing the correct pitch estimate* $\tau = 122$*; the reverberant speech signal (blue line), providing the wrong estimate* $\tau = 181$*; ten outputs of a Distributed Microphone Network (red line), providing, within a small error, the correct estimate* $\tau = 125$*.*

4.1.3 Multi-microphone Periodicity Function (MPF)

The f_0 extraction algorithm based on the MPF can be classified under the frequency domain category and, in particular, it includes a processing that resembles that described in [90].

Considering a Distributed Microphone Network context, the different paths, from the source to each microphone, are affected differently by the non linear reverberation effects, which can enhance some frequencies while

attenuating others.

The peaks in the magnitude spectrum which refer to f_0 and its harmonics, are thus altered by the linear and non-linear reverberation effects in both their dynamics and frequency location. Nevertheless, as shown in Section 3.4.2, the reverberation effects, that common speech processing applications usually deal with, can be linearly approximated by means of the room impulse responses. This implies that the actual amount of non linear distortion introduced by the ambient can be considered limited and, as a consequence of this, the peak frequency shifts will be limited to a small frequency interval. Hence, the common harmonic structure across the different magnitude spectra, can be exploited for better estimating the fundamental frequency.

An example of this is shown in Figure 4.3, where each output of a DMN consisting of 10 microphones, is plotted (left) along with the corresponding frequency spectrum (right). The corresponding clean speech segment, captured by a close-talk microphone, and its spectrum, are shown at the top of the figure, plotted in blue.

The considered clean speech segment corresponds to that used to test the multi-microphone versions of the WAUTOC and the YIN algorithms in Sections 4.1.1 and 4.1.2, respectively. The pitch period that was then estimated was of $\tau = 121$ and 122 samples, respectively. Considering the sampling frequency $f_s = 20kHz$, this corresponds to a value for the fundamental frequency falling in the approximated range $164 \leq f_0 \leq 165.3\ Hz$.

Comparing the reverberant signal spectra in the right column of the figure, with that of the clean signal (top) it is interesting to note how the reverberation detrimental effects changes the spectra shape. The peaks, relative to f_0 and its harmonics are attenuated differently and their positions is not constant across the different channels. Also spurious peaks appear beside those corresponding to f_0 and its multiples. If the funda-

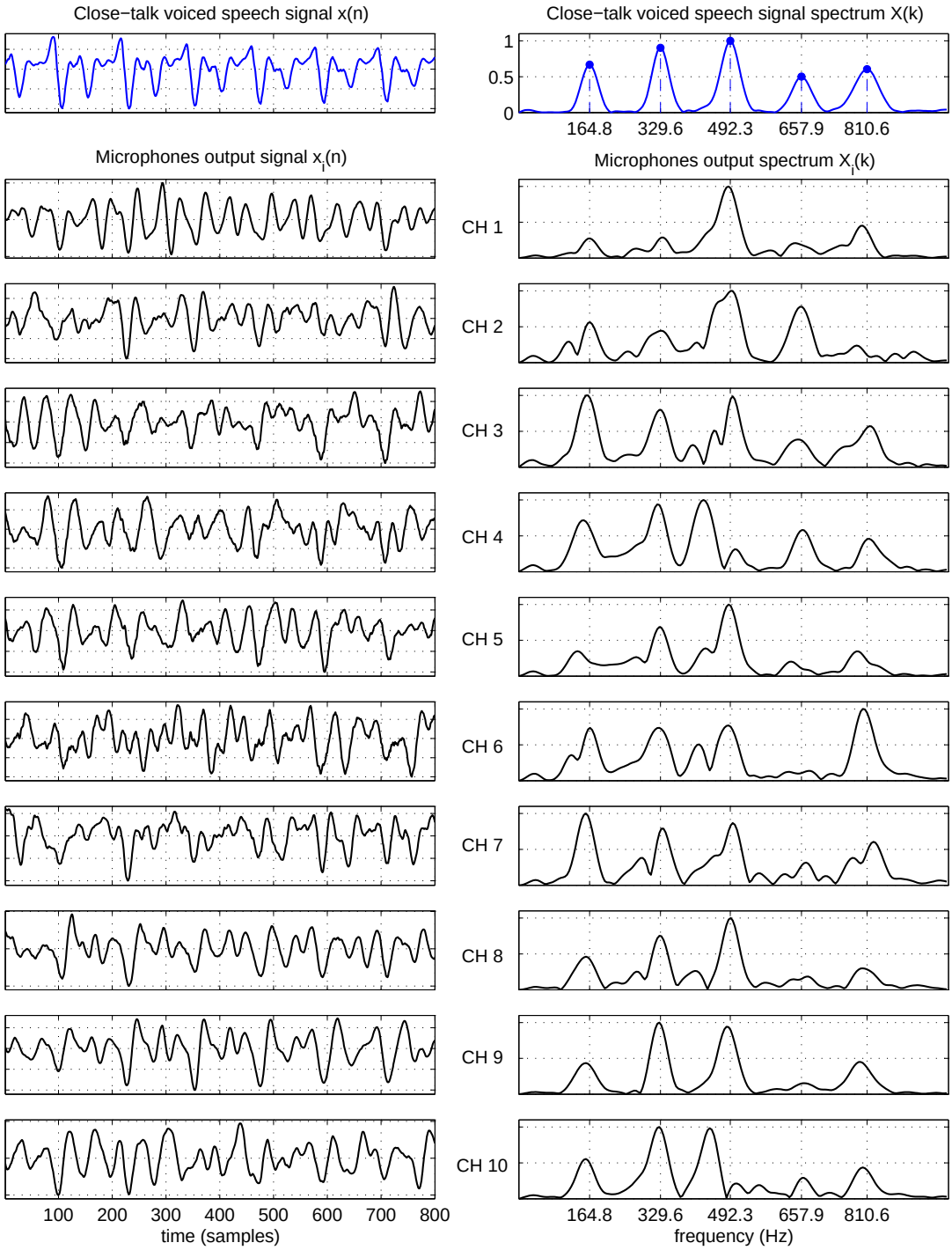

Figure 4.3: *Voiced speech segments output from a Distributed Microphone Network consisting of* 10 *microphones. Left: time domain; Right: frequency domain. The top panels show the speech segment captured by a close-talk microphone and its spectrum.*

mental frequency is considered, its peak in the clean signal spectrum is found at 164.8 Hz, in accordance with the values provided by the WAUTOC and YIN algorithms. But if the f_0 peaks of the channel signal spectra are taken into account, their position jitter from a minimum of 136.6 Hz to a maximum of 165.6 Hz and, in some cases, their amplitude is largely exceed by the harmonic peaks.

To exploit the information redundancy provided by the several channels of a DMN, a new approach based on the MPF is proposed here, which merge the frequency components belonging to the speech harmonic pattern, while rejecting the spurious one. In the following, the main algorithm steps will be described while the various processing blocks are outlined in Figure 4.4.

Let denote with $x_i(n)$, $1 \leq n \leq N$, a frame obtained from the source speech signal recorded at the i-th microphone. At each processing step, a $x_i(n)$ is weighted by the window function $w(n)$ and then zero-padded to produce the vector $\mathbf{x}_i^w$ of length N_f, with $N_f \geq N$ to result a power of 2. Then, the FFT is applied to the result and its absolute value is derived as follows:

$$X_i(k) = |\mathrm{FFT}\{\mathbf{x}_i^w\}(k)|, \quad 1 \leq k \leq N_f, \tag{4.7}$$

being k the frequency bin index. The real valued contributes $X_i(k)$ are then normalized and used to compute a weighted sum over all DMN channels:

$$X_{\mathrm{ave}}(k) = \sum_{i=1}^{M} c_i \cdot \frac{X_i(k)}{\max\limits_k\{X_i(k)\}}, \quad 1 \leq k \leq \frac{N_f}{2} + 1, \tag{4.8}$$

where the weights c_i represent the reliability of each channel and their

expression will be derived in the following. The index k is limited to $\frac{N_f}{2}+1$ since the result of Equation 4.7 is an even function, respect to the index $\frac{N_f}{2}+1$.

The last step, to derive the *Multi-microphone Periodicity Function*, is obtained computing the Inverse FFT (IFFT) of $X_{\text{ave}}(k)$, as follows:

$$\text{mpf}(\tau) = \text{IFFT}\{X_{\text{ave}}([1, \ldots, \tfrac{N_f}{2}+1, \tfrac{N_f}{2}, \ldots, 2])\} \tag{4.9}$$

where the argument of the IFFT is a vector whose N_f elements are the $X_{\text{ave}}(k)$ values, with k first ranging from 1 to $N_f/2+1$, then decreasing from $N_f/2$ to 2, so that the original symmetry of $X_i(k)$ is restored. The function $\text{mpf}(\tau)$ results thus a minimum phase signal with characteristics similar to those of the autocorrelation function described in Section 2.3.1. The main difference is that, while the ACF can be obtained as the inverse Fourier transform of the magnitude spectrum raised to the second power (power spectrum), the mpf is obtained raising the magnitude spectra, provided by Equation 4.7, to the first power. The reason for this choice will be given in Appendix C.

The resulting mpf function has thus the same properties of a generalized autocorrelation function, and the lag value at which a maximum is found, can be considered as the fundamental period T_0 of the analyzed frame. Interpolation can also be applied before searching for its maximum, in order to compensate for the resolution loss, occurred when the input signal was originally down-sampled. Once established the minimum and maximum value that the estimated pitch period can assume, T_0 is computed as

$$T_0 = \arg\max_{\tau}\{\text{mpf}(\tau)\}, \quad T_{\min} \leq \tau \leq T_{\max} \tag{4.10}$$

and all the process is repeated for the next frame of speech. Estimating the reliability of each channel contribute in Equation 4.8, that is, evaluating

how much each $X_i(k)$ gets closer to the spectrum of the close-talk signal, is carried out estimating the weights c_i in a blind fashion. This is accomplished in two steps. First a reference spectrum is derived as the product of the channel magnitude normalized spectra:

$$X_P(k) = \prod_{i=1}^{M} \frac{X_i(k)}{\max_k\{X_i(k)\}}. \tag{4.11}$$

This results in a function $X_P(k)$ that will retain the information common to the different channels while rejecting frequency patterns not common to all channels. The result of Equation 4.11 can be though of as an estimate of the close-talk speech spectrum.

The second step compute each weight c_i basing on the Cauchy-Schwartz inequality applied to functions $X_i(k)$ and $X_P(k)$ considering them as if they were vectors:

$$c_i = \frac{\sum_{k=1}^{K} X_P(k)X_i(k)}{\sqrt{\sum_{k=1}^{K} X_P^2(k)}\sqrt{\sum_{k=1}^{K} X_i^2(k)}}, \quad K = \frac{N_f}{2} + 1. \tag{4.12}$$

The coefficients provide by Equation 4.12, will thus be comprised in the range $0 \leq c_i \leq 1$, and their value may depend on the speaker position, head orientation or on the presence of other sources of noise in the operative ambient.

To better understand the weighting procedure, and considering the same close-talk and channel contributes of Figure 4.3, Figure 4.5 shows in the middle the averaged spectrum $X_{\text{ave}}(k)$ and the reference spectrum $X_P(k)$, respectively, computed in accordance with Equations 4.8 and 4.11. The reference spectrum (red line) reflects the similarities among the several channel spectra, which lie mostly in the frequency regions relative to the first and second harmonics. To see this, vertical dotted lines relative to the

Figure 4.4: *Simplified scheme of the pitch extractor based on the Multi-microphone Periodicity Function. The speech signal down-sampling blocks and the MPF interpolation blocks are not shown.*

frequency positions of f_0 and its harmonics are plotted across all spectra panels of Figure 4.3. The first harmonic at 329.6 Hz, is the one that is best matched in the several channel spectra. Disregarding the first and the second channels, in all the others it reflects quite well the amplitude and position of that in the close-talk speech spectrum. For the second harmonic at 492.3 Hz, a similar consideration can be made. Except for the fourth and tenth channels, were it is completely misplaced, in the other spectra it matches the reference one. For f_0 and the other harmonics, the spectra peaks either do not match in amplitude (as mostly happen for the third harmonic), either in frequency position (fourth harmonic), either in both amplitude and position (f_0).

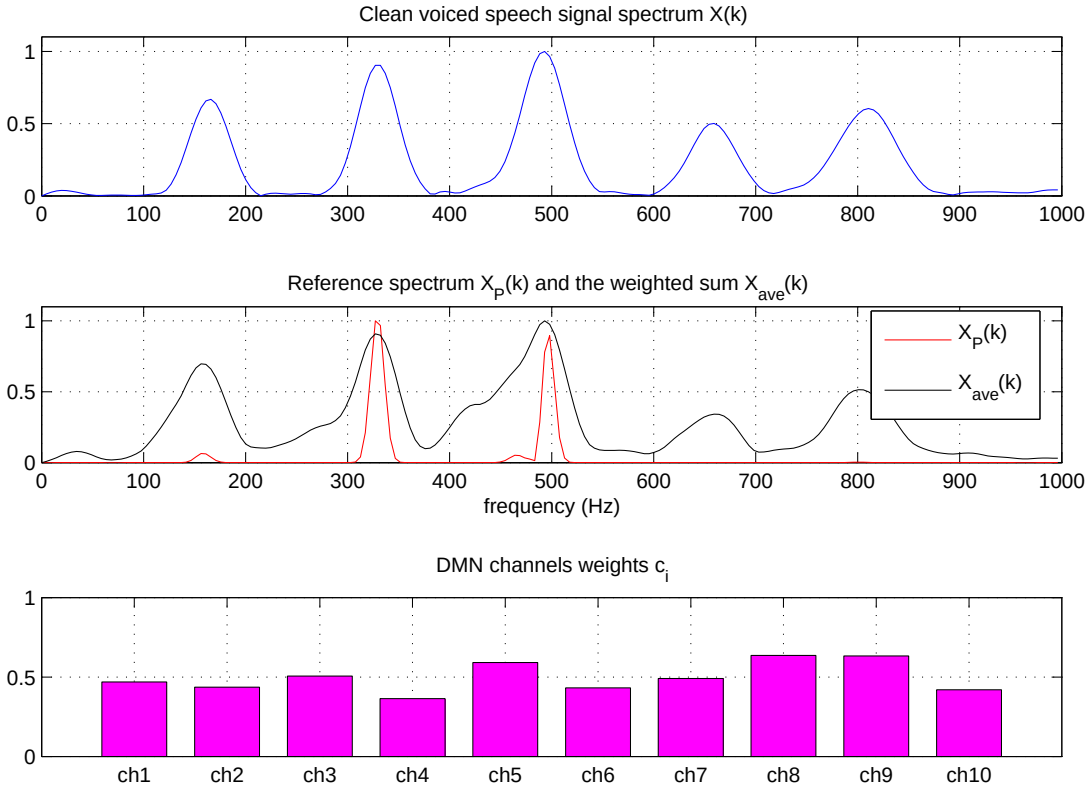

Figure 4.5: *Top panel: spectrum of the close-talk voiced speech segment plotted in the top left panel of Figure 4.3. Middle panel: reference spectrum X_P and averaged spectrum X_{ave} computed on the channel contributes shown in the left column of Figure 4.3.*

The function $X_P(k)$ is used to obtained the channel weights, which are plotted in the bottom panel of the figure. The situation described above is coherently reflected by the c_i values: channels 4 and 10 are considered the least reliable, while channels 8 and 9 demonstrated the best similarity to the spectrum of the close-talk speech signal.

The use of the channel weights c_i demonstrated particularly efficient in the cases where some DMN channels were particularly affected by noise or provided a spectral pattern particularly different from that shared by the majority of the channels. Tests with white noise sequences added at different SNR to specific channels are reported in Chapter 5.

To compare the behaviour of the MPF algorithm with the WAUTOC and YIN ones described previously, the same test relative to Figure 4.1 and 4.2 is reported in Figure 4.6.

The top and middle panels show the voiced speech segment captured by a close-talk and distant microphones, respectively. The bottom panel reports the MPF function computed on the clean speech signal, $\text{mpf}_{\text{cl}}(\tau)$, on the reverberant signal captured by a distant microphone[4], $\text{mpf}_{\text{mic}}(\tau)$, and on all the DMN contributes, $\text{mpf}(\tau)$, respectively. The frequency domain approach results more robust compared to the other time domain based PDAs tested. In fact, the correct pitch estimate is provided also when the speech signal provided by the distant microphone (blue line) is processed. In the latter case though, two secondary strong peaks at about $\tau = 179$ and 50 samples appear, denoting the low reliability of the incoming signal.

In Chapter 5 the results obtained from testing the MPF algorithm in different reverberant conditions are reported and discussed.

[4]In the cases where a single speech input is used, the latter represent the unique term in the summation of Equation 4.8, and the relative c_i coefficient is set to 1.

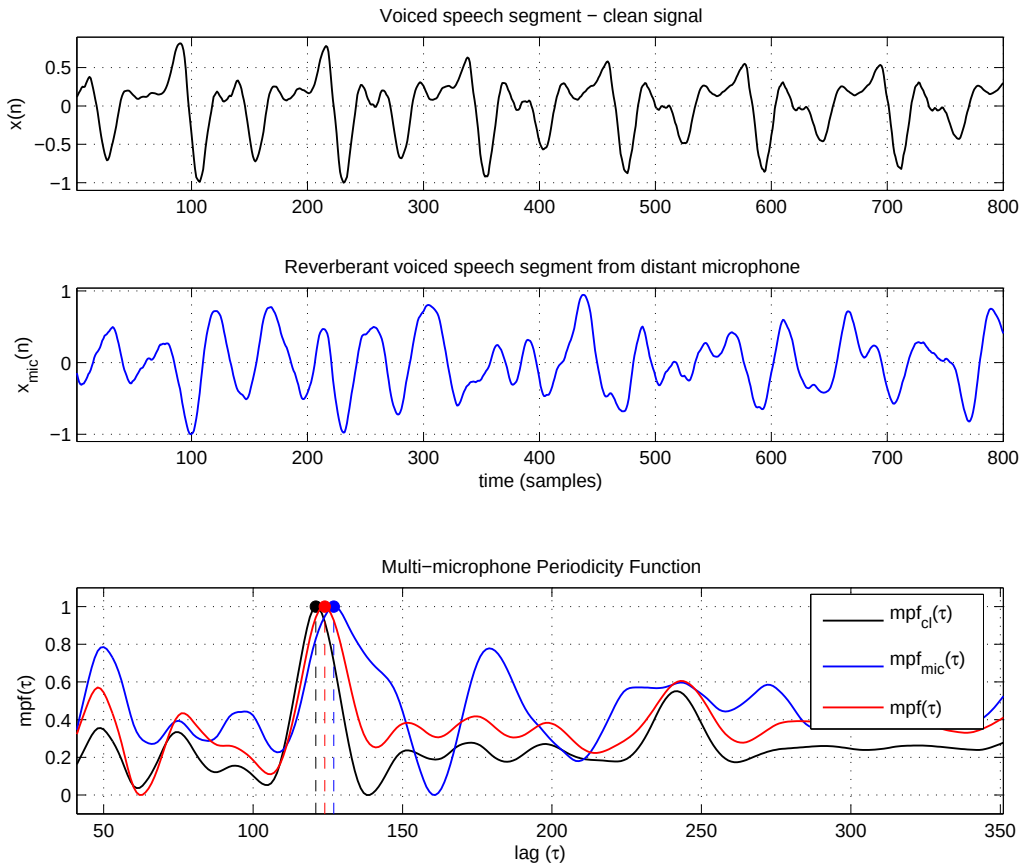

Figure 4.6: *A clean voiced speech signal and its reverberant version are shown in the top and middle panels, respectively. The bottom panel shows the MPF computed on: the clean speech signal (black line); the reverberant speech signal (blue line); ten outputs of a Distributed Microphone Network (red line). All tests provided, within a small error, the correct pitch estimate, that is,* $\tau = 121,\ 127$ *and* 124*, respectively.*

Chapter 5

Experimental Results

In the previous chapters, many pitch extraction algorithms have been described and discussed. However, the focus has been restricted to two state of the art algorithms, the Weighted Autocorrelation and the YIN algorithms, and to the Multi-microphone Periodicity Function, proposed in this thesis to overcome some limitations imposed by the traditional approaches, when tested in reverberant and noisy scenarios. This chapter describes the speech material that was collected and adapted to test these algorithms, the error measure adopted for their evaluation and the obtained results. Particular care and effort were dedicated to this testing phase, since it is very important to have a large amount of speech data and a particularly precise pitch reference, against which to compare the PDAs output. Some further considerations about the characteristics that pitch reference labels shall possess, are given in Appendix B.

5.1 Performance evaluation

Empirically evaluating the performance of a pitch extraction algorithm is a difficult task. Recalling that here and in the general pitch extraction context, the term "pitch" is given a meaning of fundamental frequency, the results provided by a PDA are not easily judged by a human listener, who

has a subjective perception of pitch. However, whenever this evaluation system is chosen, several experts have to listen to a large number of speech files to lend credibility to the obtained measure. Although the effort for doing this is consistent, once data has been labeled it can be used to test other PDAs in an automatic way.

Alternatively, when speech data is acquired, a laryngograph can be used to record the signal which reflects the vocal folds vibration. This signal is generally used as input to a pitch extractor algorithm and the result is then manually checked to correct possible octave errors. Since the vocal tract effects are avoided, reliability of this measure, compared to the one obtained from the speech signal, results much higher. This solution results in a very precise reference pitch estimate, even if there is the need of specific hardware as the laryngograph and a device to convert its output to a digital representation. There are few speech databases publicly available provided with pitch reference labels. The Keele database [80], is one of them and it is used to evaluate the performance of the pitch extraction algorithm proposed in this thesis. The results obtained are reported in Section 5.2, as well as the comparison with those obtained from other state of the art algorithms.

Another method to obtain a speech database with reference pitch values, is based entirely on existing pitch extractors algorithms. These are run on a set of speech files, and the estimates provided by each PDA are compared. In case their difference is smaller than a specific fixed threshold, the result, obtained as their average or other merging technique, is used as a reference pitch value. Otherwise, in case an estimates mismatch occurs, the pitch value on which the majority of the PDAs agree, can be taken as the correct one. However, in the case the PDAs provide different pitch estimates, it

is not rare that the majority of the pitch extractors agree on an erroneous f_0 estimation. To be sure of the final labels reliability, a manual checking is needed anyway. This method was used to obtain the CHIL database, described in Section 5.3.

5.1.1 Error measures

There are various measures that can be used to evaluate the quality of a pitch extraction algorithm. The principal ones are response time, accuracy, resolution and complexity. The *response time* measures the delay with which the device adapts to a sudden change in the pitch or provides its estimate after a unvoiced/voiced transition occurs. *Accuracy* is related with the result reliability, while *resolution* is concerned with the precision with which the provided value matches the reference pitch. *Complexity* is principally involved with the amount of hardware resources needed to run the algorithm, in terms of memory and computational requirements. This measure turns out to be very important in real-time applications, where no delay is allowed between the current analyzed frame and the provided pitch estimate.

In this thesis, the principal method used to evaluate the PDAs performance is the Gross Error Rate (GER). This is calculated considering the number of f_0 estimates which differ by more than a certain percentage θ from the reference values. Considering a total of N_{fr} pitch values estimated, its formulation can be written as

$$\text{GER}(\theta) = \frac{100}{N_{\text{fr}}} \sum_{i=1}^{N_{\text{fr}}} \left\{ \frac{|\hat{f}_{0_i} - f_{0_i}|}{f_{0_i}} > \theta\% \right\}, \tag{5.1}$$

where $\hat{f}_{0_i}$ and f_{0_i} are the fundamental frequency estimated and the reference one relative to the i-th frame, respectively. The term in curly brackets,

returns the value 1 or 0 depending on the result of the inequality. Generally, values of 20 and 5 are used for θ in the experiments. The former indicates the PDA capability to avoid large estimated/reference pitch mismatches, as the octave errors. The latter gives more an indication of the pitch estimate resolution.

A measure that better quantifies the PDA resolution capacity is the Root Mean Square Error (RMSE) or fine pitch error, which is computed considering only the set $\Omega(\theta)$ of pitch estimates that differ by less than θ percent from the ground truth:

$$\mathrm{RMSE}(\theta) = \sqrt{\frac{1}{N_{\mathrm{GER}(\theta)}} \sum_{i \in \Omega(\theta)} \left(\frac{\hat{f}_{0_i} - f_{0_i}}{f_{0_i}} \right)^2}, \tag{5.2}$$

where $N_{\mathrm{GER}(\theta)}$ equals the number of elements in the set $\Omega(\theta)$. Although this measure provides a very precise indication of the PDA resolution capabilities, the GER is mainly used here for the experiments evaluation. In fact, as stated in [17], once the initial estimate provided is within 20% of being correct, there exist many further processing techniques available to refine its value.

Error measures of Equations 5.1 and 5.2 have to be computed considering only the pitch estimates obtained from voicing sections of the analyzed speech signal. This is a self-evident truth, given that no reference pitch can be available for unvoiced speech. To detect voicing sections a Voiced/UnVoiced (V/UV) detector is usually used besides a PDA. It can be also included in the pitch extractor, often exploiting the periodicity degree of the main function that is used to estimate the pitch. However, the problem arises whenever different PDAs have to be compared. In fact, it

may happen that if the V/UV information provided by each PDA is used to evaluate its estimates, these voicing segmentations do not coincide for all devices. Therefore it will not be possible to establish whether a particular device performs better than the other due to its good pitch estimation capabilities, or to its precision in V/UV decisions. To avoid this possible ambiguity, tests carried out in the following, assume a common V/UV sections segmentation for all tested algorithms.

5.2 Keele

The Keele database consists of five male and five female English speakers who pronounced phonetically balanced sentences from the "The North Wind Story", for a total duration of 9 minutes. A close-talk microphone was used in a soundproof room to record the readers while a laryngograph was simultaneously employed to track the signal generated by the speakers vocal folds. The sampling frequency used to digitize the signal was set to $f_s = 20\ kHz$ with 16 bit resolution. The pitch reference files contain V/UV information and a pitch estimate every 10 ms frame length of the speech signal. Pitch values were extracted applying the autocorrelation function to the laryngograph output [80].

Following the suggestion of the Keele database description authors, the laryngograph data was used in this thesis to derive new pitch references. The reason for this is twofold: on the one hand a shorter time interval between pitch values was needed in order to gain a more accurate feed-back from tests. On the other hand, a different and a more reliable method than the autocorrelation was needed to extract the new reference values. This assumption is justified if the example of Figure 5.1 is considered.

The first two panels show a voiced speech segment from a male speaker

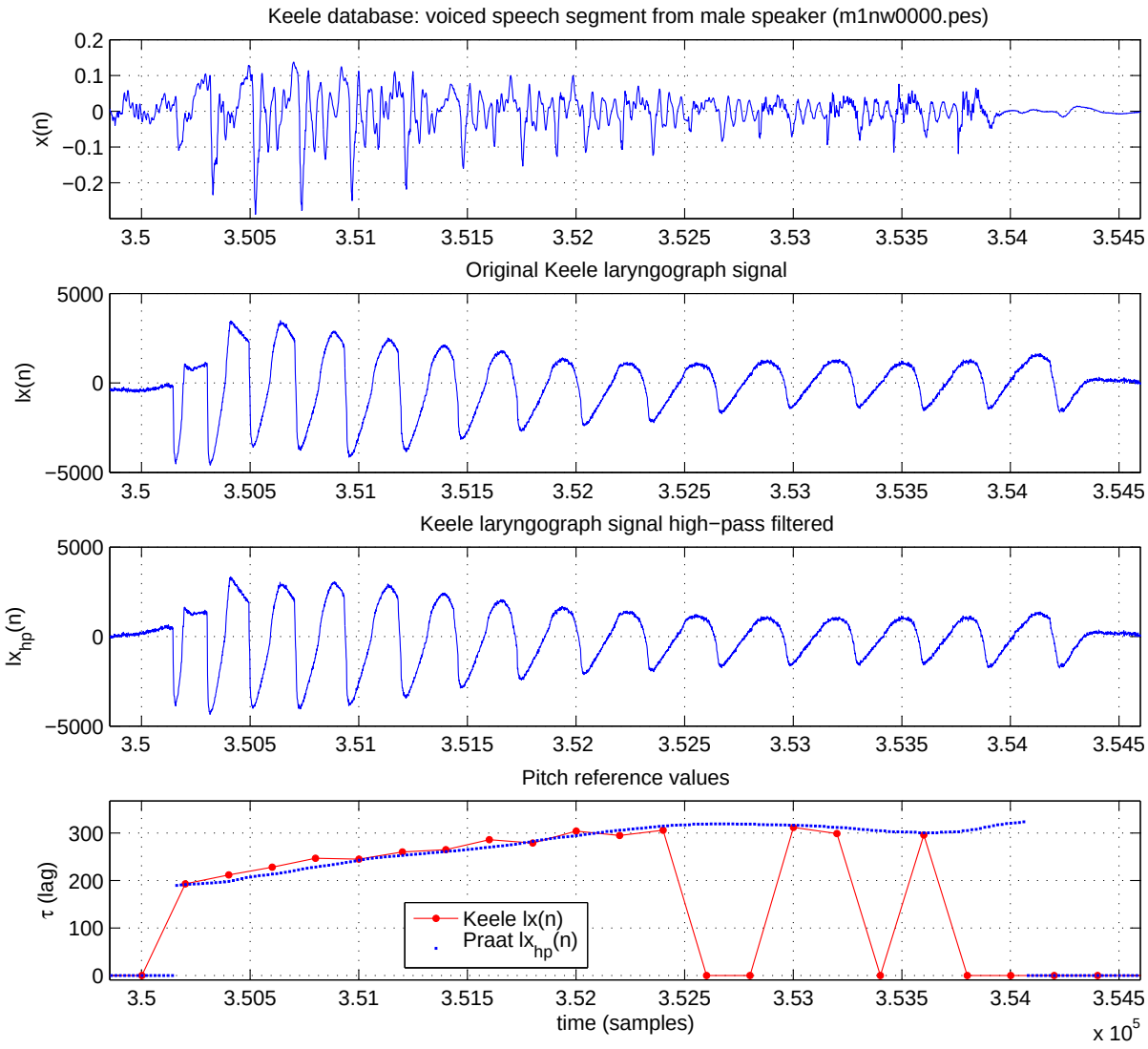

Figure 5.1: *Example of pitch references based on laryngograph signal. A voiced speech segment of a male speaker and the relative laryngograph signal $lx(n)$ are plotted in the first and second panels, respectively. The third panel shows the high-pass filtered version, $lx_{\mathrm{hp}}(n)$, obtained from $lx(n)$. The bottom panel compares the pitch references obtained using the Praat tool [11], applied to $lx_{\mathrm{hp}}(n)$, and the original ones, provided with the Keele database.*

$x(n)$, and the relative laryngograph signal $lx(n)$, respectively. The latter is affected by a slowly varying bias, more evident in the leftmost part of the panel. This is probably due to movements of the speaker during the original recordings. To eliminate this bias a high-pass linear phase filter,

with a 3 dB cut-off frequency of 34 Hz was applied to the signal $lx(n)$ to obtain the unbiased version $lx_{\mathrm{hp}}(n)$ plotted in the third panel.

Signal $lx_{\mathrm{hp}}(n)$, was then used to reestimate pitch references, with an analysis step of 1 ms by means of the Praat tool [11]. The final result was then manually checked in order to correct possible octave errors, and mismatches between some laryngograph segments that could result voiced and their correspondent speech segments. These, actually, were unvoiced because the speaker had her/his mouth closed or for some constrictions occurring along their vocal tract. The fourth panel in the figure shows the resulting pitch estimate (black line) plotted along with the Keele original pitch labels (red line). The pitch reference value of unvoicing sections was set conventionally to 0, so that it was used as a reference to detect the voicing frames selected to evaluate the PDAs performance.

5.2.1 Scenario

To measure the performance of the proposed algorithms, the Keele database, was reproduced to derive a multichannel database. As depicted in Figure 5.2, a Distributed Microphone Network consisting of 10 omnidirectional sensors (plotted in magenta), was used to record the Keele database reproduced with a very high quality, dual-concentric (TANNOY 600A) loudspeaker.

To have different sound propagation contexts, this was done twice.First placing the speaker in position $P1$ (plotted in green), then in position $P2$ (plotted in red), with orientations as shown in the figure. The office in which this was carried out is 3 m $\times$ 7 m wide and 3 m high, and is characterized by a reverberation time $T_{60} = 0.35$ s. As shown in the figure, adjacent microphones were from 0.2 to 2 meters far from each other. During recordings there were no people in the room, and the only source of noise was the computer fan, marked with a blue asterisk [6].

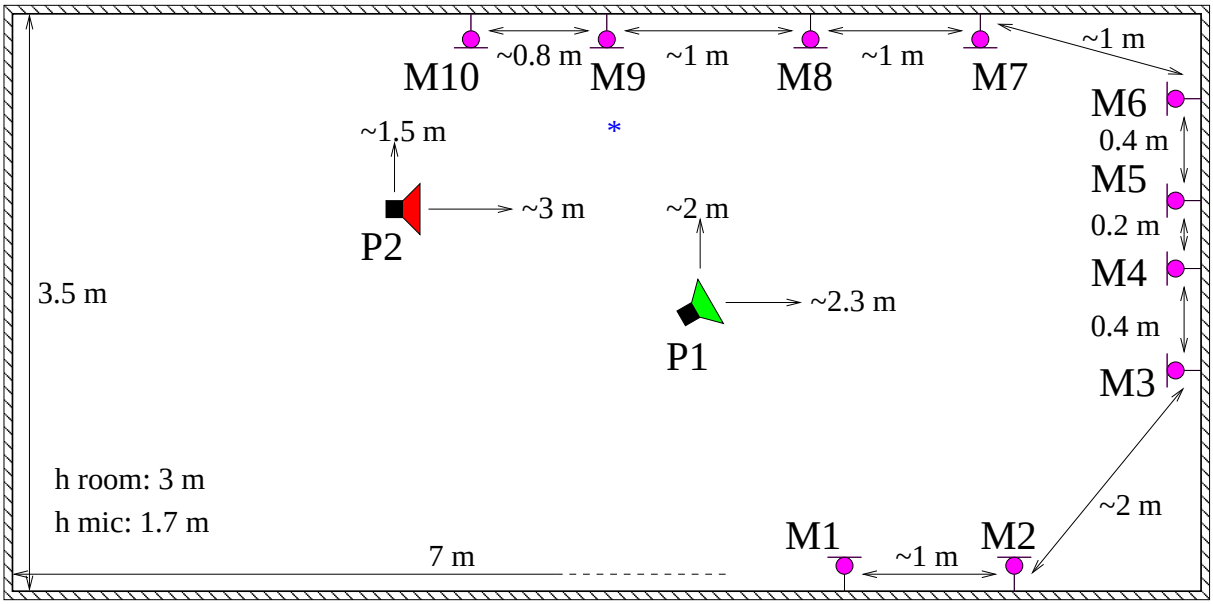

Figure 5.2: *Office with ten microphones and loudspeaker placed in two positions, one marked with $P1$ with 30 degrees top right orientation, the second in $P2$, directed from left to right. The room is quiet, except for a computer fan marked with a "$*$".*

Once the multi-microphone speech dataset was collected, each microphone contribute had to be aligned to compensate for the different delay with which the source sound propagated to the microphones. This was done manually in order to ensure the best reliable alignment.

5.2.2 Results

The multi-microphone database previously obtained was used to test WAUTOC, YIN and MPF algorithms, described in Section 4.1.1, 4.1.2, and 4.1.3, respectively.

For both speaker positions, $P1$ and $P2$, three contexts have been considered for the evaluation. First of all the algorithms were tested using close-talk data, that is, the original Keele database. Then each distant microphone contribute was considered individually, and finally all chan-

nels were used jointly to test the algorithm multi-microphone versions. In all the three conditions, the analysis step was set to 1 ms, that is, the PDAs provided 1000 pitch estimates/sec. The length of the analysis windows instead, was set to 30 ms, 40 ms, and 60 ms, for YIN, WAUTOC (rectangular window) and MPF (Hamming window) based algorithms, respectively. These different window types and durations were determined by preliminary experiments aimed to optimize each algorithm performance.

Graphs of Figure 5.3, 5.4, 5.5 and 5.6 refer to pitch estimation results measured in terms of GER(20) and GER(5), respectively. On x-axis, the number of the considered DMN microphone is indicated, while on y-axis, the measured GER is shown. Each algorithm is marked with a specific symbol, that is, the "▼" for WAUTOC, "●" for YIN and "■" for MPF. To distinguish between the different analyzed contexts, results have been plotted with different colors: red for single distant channel context, black for joint multi-microphone scenario, and blue for results obtained using the close-talk signal. Table 5.1, 5.2, 5.3 and 5.4 numerically summarize the results shown in the graphs.

Keele reproduced in position $P1$

On the top right position of Figure 5.3 is shown the office environment with the DMN and the speaker position and orientation. As shown by the red graphs, which report the results obtained by the three PDAs applied on each distant microphone individually, the 3-rd, 4-th, 5-th and 10-th microphones provided the most corrupted signal. The high GER(20) values obtained are due to the presence of windows located above the first group of microphones. These are characterized by a higher reflection coefficient compared to the surrounding walls. Instead, microphone 10 falls almost outside the sound field produced by the speaker. Consequently, it cannot

capture sound proceeding from the direct path properly and, thus, the reflected components have a greater detrimental influence. This holds, to some extent, also for microphones 1 and 9. The best result is obtained using the speech signal recorded by the 8-th microphone. It is positioned almost in front of the source signal and far enough from the reflecting windows before mentioned.

Among the single-channel versions of tested algorithms, WAUTOC provided the worse results, while the MPF the best one. YIN algorithm gave instead, GER(20) values in between.

GER(20)	WAUTOC	YIN	MPF
close-talk	4.51	**2.04**	2.68
single-mic	14.56	11.56	**9.42**
multi-mic	8.09	6.80	**5.39**

Table 5.1: Gross error rates (20%) obtained applying WAUTOC, YIN and MPF, respectively, to the Keele speech dataset. Values refer to the curves depicted in Figure 5.3 and in the second row the averages, computed for each red curve, are reported. Bold font is used to indicate the best result obtained in each acoustic condition.

Disregarding WAUTOC, which provided the worse results in both contexts, trend is inverted observing the results relative to the close-talk speech signals (blue line). As expected, YIN provided the lowest GER, 2.04%, against 2.68% of the MPF, confirming the good performance pointed out in [17]. A first temporary conclusion that can be drawn from these figures, is that an approach based on the frequency domain, the MPF, can result more advantageous for pitch estimation on reverberant signals. The plots of Figure 4.3, where is shown how the reverberant versions of the voiced speech segment lose their periodic characteristics, further strength this hypothesis.

When multi-microphone versions of the three algorithms are tested

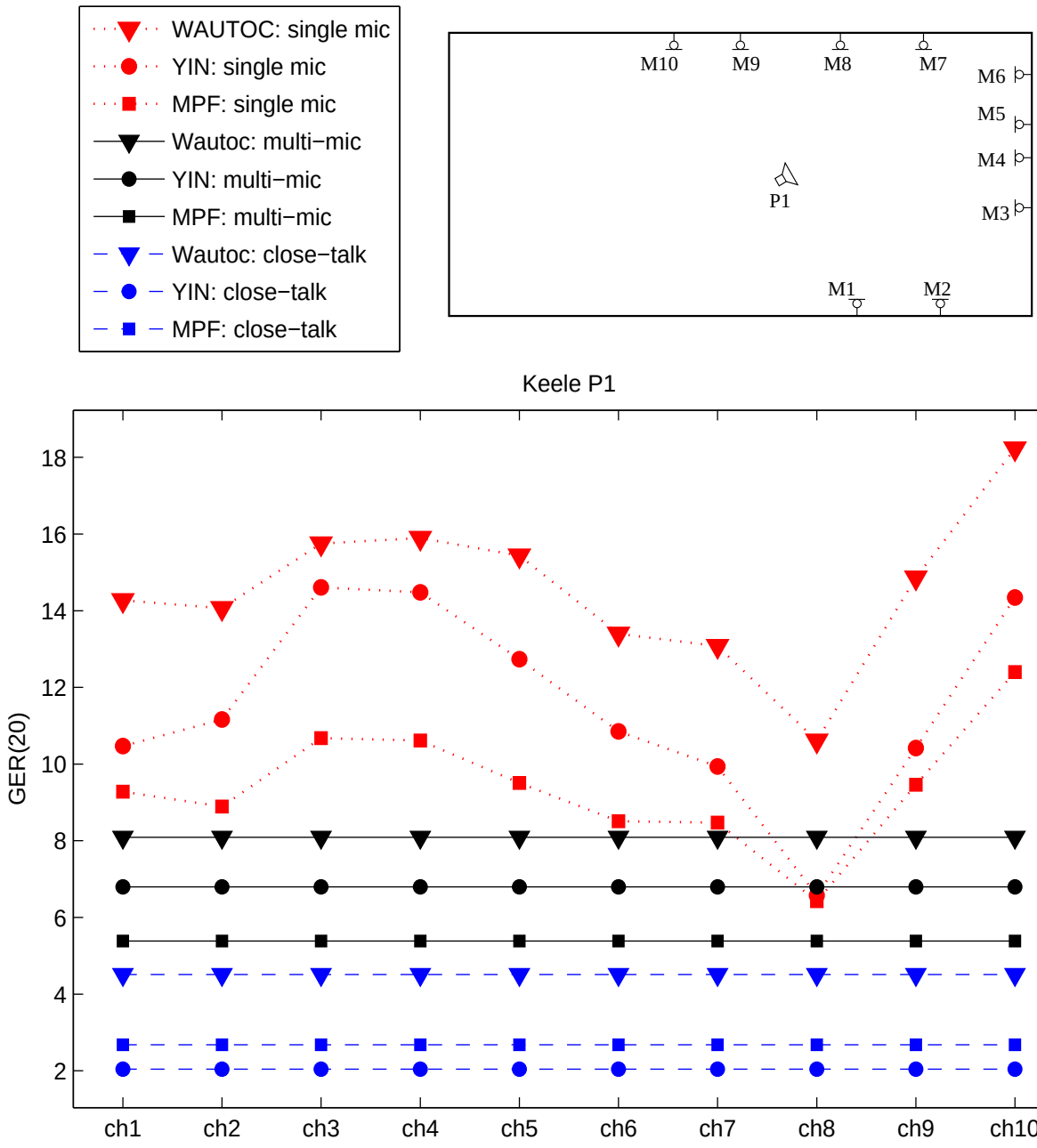

Figure 5.3: *The three red curves show gross error rates (20%) derived by applying WAUTOC (▼), YIN (●) and MPF (■), respectively, to each of the 10 microphone signals. Results refer to the loudspeaker in position P1. The six horizontal lines indicate the performance provided by each of the three algorithms on the close-talking signals (blue) and the corresponding performance obtained by their multichannel version applied to all the far microphone signals (black).*

(black lines), the trend provided by the single distant microphones is confirmed. In this case too, the MPF provided the lowest GER(20), 5.39%, compared to the values of 6.8% and 8.09% given by YIN and WAUTOC, respectively. It is interesting to note that the MPF further reduced the GER in comparison with the best result that had previously achieved from any single microphone. This to confirm, as pointed out in Section 4.1.3, the ability of the proposed algorithm to exploit the information redundancy offered by the DMN, and to reject reverberant contributes which affect differently each channel.

Besides, comparing the above results with the value 16.2%, obtained applying traditional beamforming techniques, as shown in [6], it confirms how the latter techniques are unsuitable for such a microphone disposition (Section 4.1).

The Multi-microphone Periodicity Function based PDA was not integrated with an estimate refinement block, as is YIN, for example. The reason for this is that, as already pointed out previously, the most difficult task in pitch detection is to avoid gross errors, being relatively simple to refine a correct estimate. Figure 5.4 reflects this design choice, showing that when the GER(5) error measure is used, YIN performs generally better, both in the close-talk and multi-microphone contexts. In fact, the refinement step of this algorithm consists, as stated in [17], in "shopping around the vicinity of each analysis point for a better estimate". This means that, once a short-segment of voiced speech is processed, for each obtained pitch estimate, the neighbouring values are checked for a more reliable estimate. If this happens, the first is replaced with the new value and possible fine, as well as gross errors are avoided.

Despite that, in reverberant conditions, the proposed algorithm provides comparable error rate in the close-talk case respect to YIN. And, in confir-

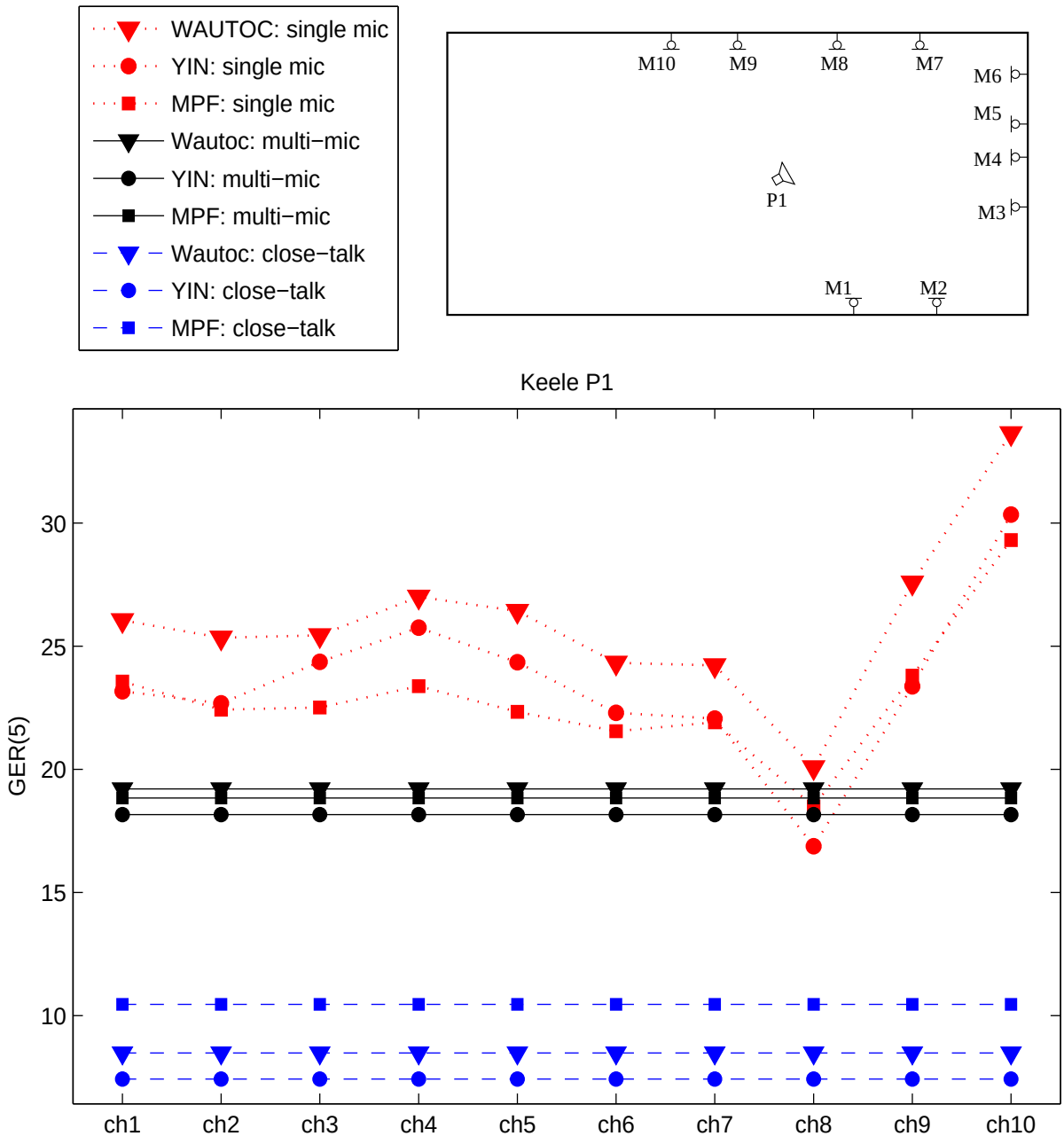

Figure 5.4: *The three red curves show gross error rates (5%) derived by applying WAUTOC (▼), YIN (●) and MPF (■), respectively, to each of the* 10 *microphone signals. Results refer to the loudspeaker in position P*1*. The six horizontal lines indicate the performance provided by each of the three algorithms on the close-talking signals (blue) and the corresponding performance obtained by their multichannel version applied to all the far microphone signals (black).*

GER(5)	WAUTOC	YIN	MPF
close-talk	8.49	**7.42**	10.46
single-mic	26.01	23.52	**22.93**
multi-mic	19.20	**18.16**	18.83

Table 5.2: Gross error rates (5%) obtained applying WAUTOC, YIN and MPF, respectively, to the Keele speech dataset. Values refer to the curves depicted in Figure 5.4 and in the second row the averages, computed for each red curve, are reported. Bold font is used to indicate the best result obtained in each acoustic condition.

mation of the strength against high signal distortion, better values for each single distant microphone are obtained. This behaviour will be confirmed for tests carried out with the loudspeaker placed in position $P2$. In this scenario the overall reverberation effect is stronger and MPF is shown to provide even better results.

Keele reproduced in position $P2$

The first thing to note when considering the experiments run with the loudspeaker located in position $P2$, is the higher reverberation which affects the speech signals. This is visible if Figure 5.5 is considered. In fact, red curves, obtained by testing the three algorithms on each single distant microphone, have a higher average value compared to those of the $P1$ scenario. In particular, it is interesting to note that the presence of windows on the top part of the right wall, still affects negatively the signal acquisition by the microphones which are below it. This is true especially for the 5-th microphone, which provides one of the worst contributes, as shown by the high GER.

The best acquisitions were those of microphones 8, 9 and 10, for their close placement near the sound source and far from the windowed wall. As it did in the $P1$ case, YIN results still lay in between those provided by the MPF algorithm, and the WAUTOC results, which turned out to be

the worst.

GER(20)	WAUTOC	YIN	MPF
close-talk	4.51	**2.04**	2.68
single-mic	17.10	14.50	**10.98**
multi-mic	10.14	8.97	**7.00**

Table 5.3: Gross error rates (20%) obtained applying WAUTOC, YIN and MPF, respectively, to the Keele speech dataset. Values refer to the curves depicted in Figure 5.5 and in the second row the averages, computed for each red curve, are reported. Bold font is used to indicate the best result obtained in each acoustic condition.

The results obtained from the close-talk scenario (blue line), were already commented in the previous section. They represent a GER lower-bound for the three tested algorithms and are reported in this graph just for comparison purposes. What it is interesting to note here, is the general worsening of the GER figures when the reverberant signals are used, both in single or multi-channel fashion.

In the latter case (black line), the MPF achieved the best result, GER(20) = 7%, followed by YIN and WAUTOC algorithms with a GER(20) of 8.97% and 10.14, respectively. Also in this scenario, MPF further reduced GER respect to the best result that had previously achieved from any single microphone. This happened also for the WAUTOC algorithm which obtained the best improvement, comparing with single-distant microphone scenario. However, this time domain based algorithm demonstrated its ineffectiveness to process reverberant signals, compared to the other PDAs that have been considered.

For a further comparison, GER(20) value of 19%, is here reported from [6], where WAUTOC algorithm was applied to the beamformed signal obtained using traditional techniques, as the "delay and sum" approach. Being this value almost twice higher than the one obtained with the multi-

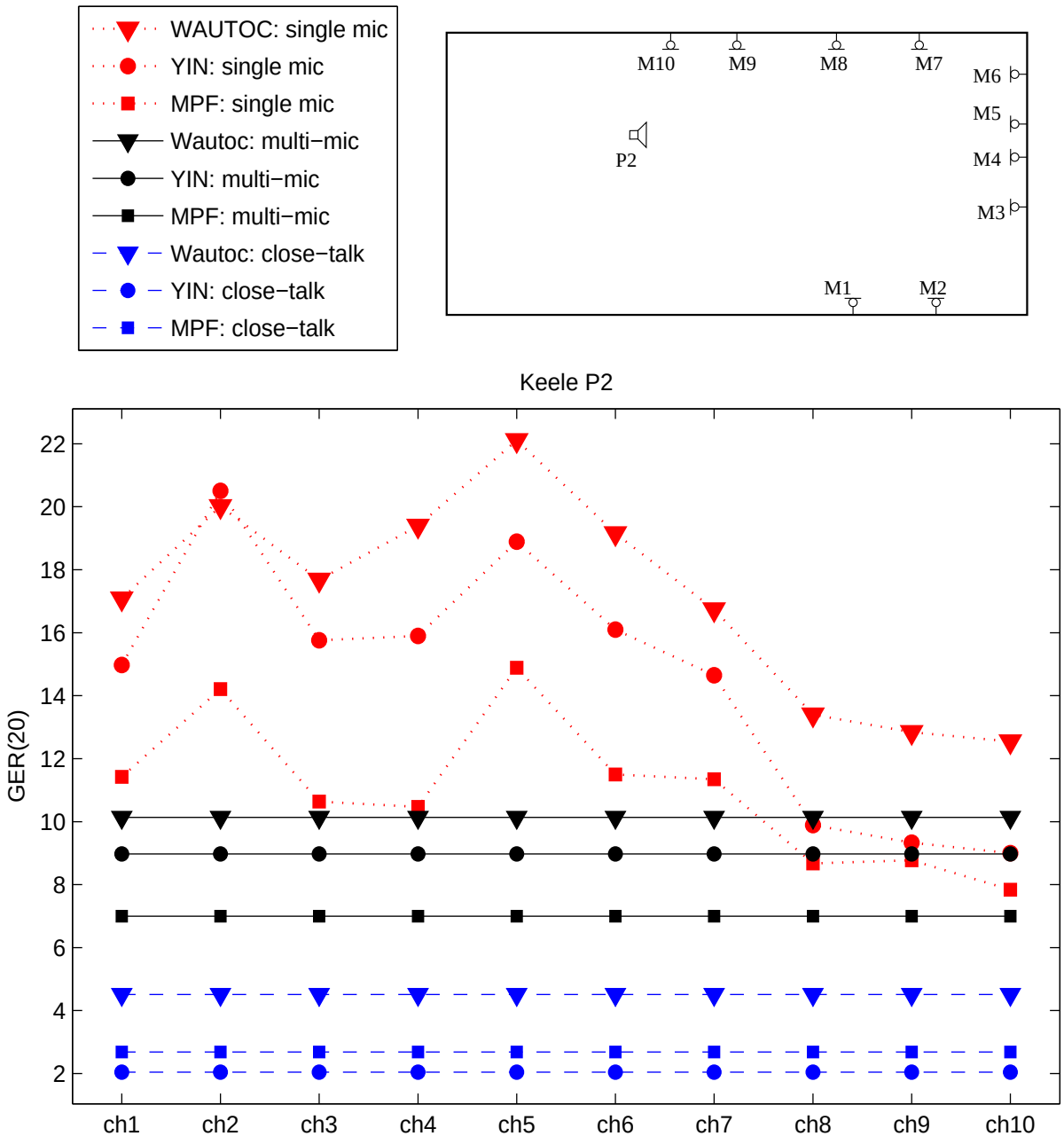

Figure 5.5: *The three red curves show gross error rates (20%) derived by applying WAUTOC (▼), YIN (●) and MPF (■), respectively, to each of the 10 microphone signals. Results refer to the loudspeaker in position P2. The six horizontal lines indicate the performance provided by each of the three algorithms on the close-talking signals (blue) and the corresponding performance obtained by their multichannel version applied to all the far microphone signals (black).*

microphone version of WAUTOC, it underlines once more the impossibility to apply signal processing techniques suited for microphone arrays, in a DMN context.

As pointed out in the previous section, despite the fact that MPF algorithm does not perform post-processing for pitch estimate refinement, in case of strong reverberation conditions, it is able to provide the best performance. As shown in Figure 5.6, in the close-talk case (blue line) YIN and WAUTOC performed better, being designed to cope better with the clear periodicity of close-talk speech signals. Instead, when reverberant signals are considered, that is in the single distant and multi-microphone contexts, MPF still provided pitch estimates with the best resolution.

GER(5)	WAUTOC	YIN	MPF
close-talk	8.49	**7.42**	10.46
single-mic	29.99	27.80	**25.77**
multi-mic	22.89	21.88	**21.59**

Table 5.4: Gross error rates (5%) obtained applying WAUTOC, YIN and MPF, respectively, to the Keele speech dataset. Values refer to the curves depicted in Figure 5.6 and in the second row the averages, computed for each red curve, are reported. Bold font is used to indicate the best result obtained in each acoustic condition.

Channel reliability estimation

To assess the effectiveness of introducing weights c_i in Equation 4.8, some experiments were conducted in which the signals provided by specific microphones of the DMN were contaminated with noise at different levels of SNR, while the remaining DMN channels, or a subset of them, were used in their original version.

For the first experiment only the signals from the 1-st, 2-nd and 3-rd microphones were considered so that to constitute a reference dataset.

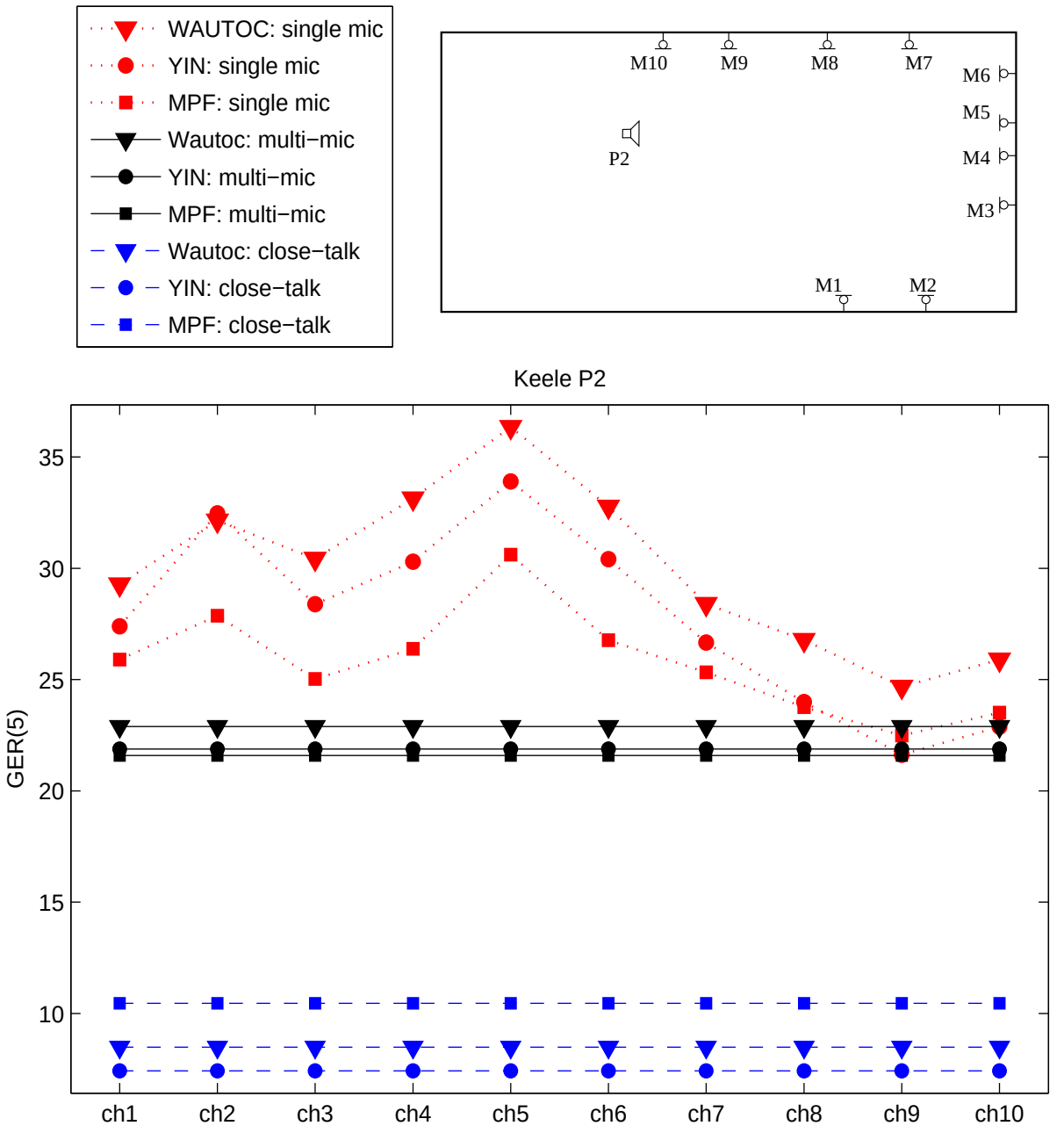

Figure 5.6: *The three red curves show gross error rates (5%) derived by applying WAUTOC (▼), YIN (●) and MPF (■), respectively, to each of the* 10 *microphone signals. Results refer to the loudspeaker in position P*2*. The six horizontal lines indicate the performance provided by each of the three algorithms on the close-talking signals (blue) and the corresponding performance obtained by their multichannel version applied to all the far microphone signals (black).*

From this, two other datasets were derived adding to the 3-rd channel white noise with a SNR of 0 and 5 dB, respectively. The procedure was then repeated to derive two more datasets using babble noise instead of white noise.

GER(20) results provided by the multichannel version of WAUTOC (blue line), YIN (black line) and MPF (red line) applied to the five speech datasets obtained, are shown in Figure 5.7. The upper panel in the figure shows tests conducted on the speech signals contaminated by white noise, while the lower panel describes the performance obtained employing speech data to which babble noise was added. In both scenarios two versions of the MPF were tested: the first with all weights c_i set to 1 (dashed line) so that all microphone contributes where equally considered in Equation 4.8, the second with weights provided by Equation 4.12.

The common x-axis reports which of three speech datasets was considered for each scenario, indicating, from left to right, decreasing SNR levels measured on the third microphone output.

In the upper right part of the figure the loudspeaker position and direction and the three microphones considered, $M1$, $M2$ and $M3$ are shown. Red color was used to plot microphone $M3$ to indicate that its output was contaminated with different SNR values.

As shown in the figure, the GER(20) provided by the three algorithms in both the white and babble noise scenarios, worsened as the SNR of the third channel decreased. Also a common observable trend in all tests is that MPF provided the lowest GER(20) while YIN and WAUTOC performed worse. As indicated from the results relative to the "no noise added" case, that is, when the original speech signals were used, the MPF function provided almost the same GER(20) value when its two versions were tested. Channel reliability estimation resulted thus not particularly advantageous in this particular noise-free scenario. The opposite can be stated instead for the

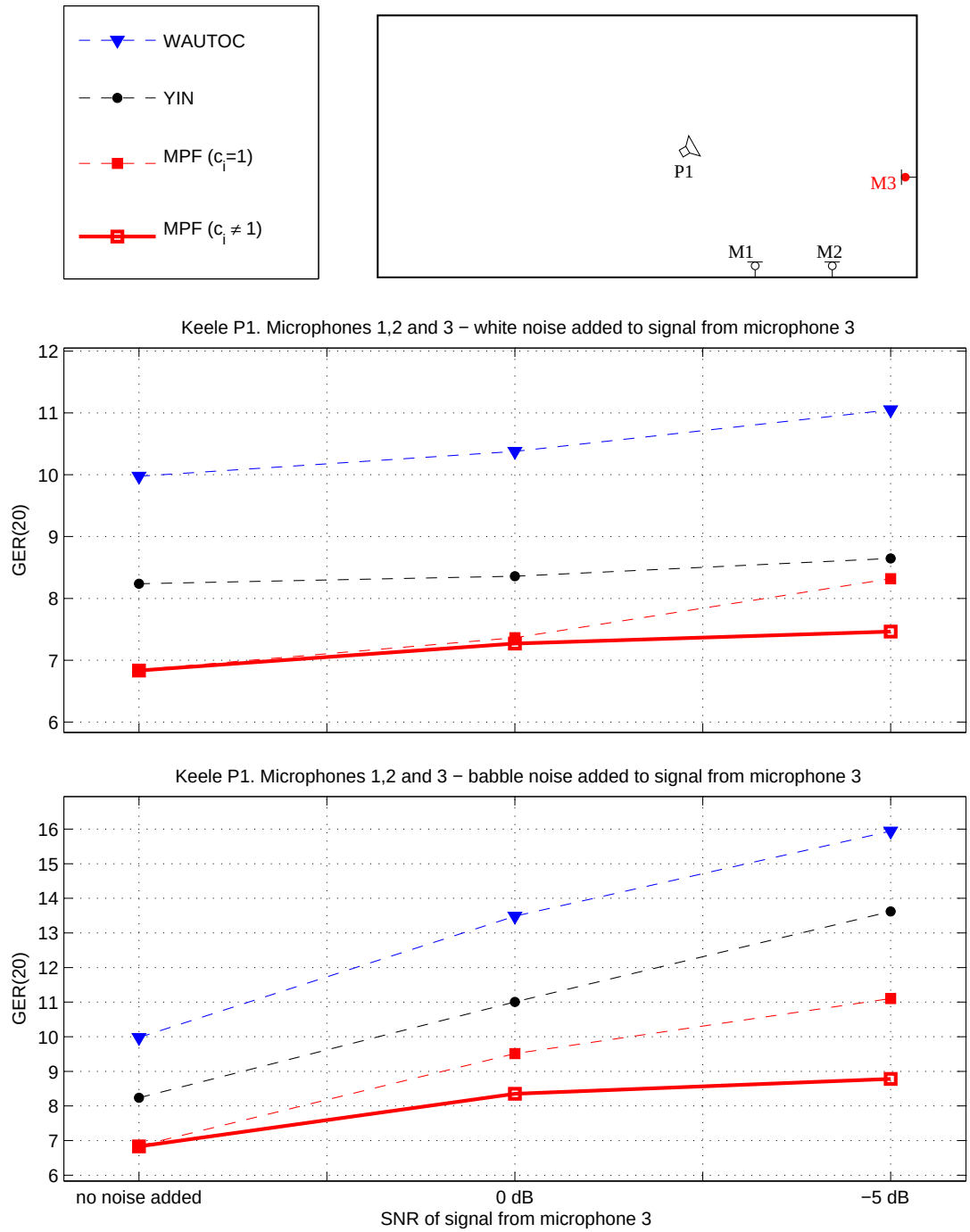

Figure 5.7: *Gross error rates obtained by the multichannel version of each algorithm under different noisy conditions. Only three microphones were used and noise was added to channel 3 at different SNR levels. The upper panel shows results obtained on speech data contaminated with white noise, while lower panel refers to the babble noise scenario.*

results obtained in noisy conditions. For decreasing SNR levels, the MPF version with weights estimation provided by Equation 4.12, demonstrated to be the more robust in both the white and babble noise conditions.

As it also resulted for WAUTOC and YIN in the tests showed in Figure 3.9 and 3.10, the three algorithms resulted more robust to white noise, in fact the curves plotted in the upper panel of Figure 5.7 resulted more flat compared with those of the lower panel.

Babble noise instead represents a more difficult noise that the algorithms have to cope with, since its spectrum rather than being flat, as for the case of white noise, can resemble that of voiced speech, becoming thus a misleading source of information for the f_0 estimator. This can be seen in the lower panel observing that the GER(20) provided by both WAUTOC and YIN increased of almost 6% passing from the clean scenario to the -5 dB SNR one. Also the MPF version with all weights c_i set to 1 performed considerably worse with decreasing babble noise SNRs, passing from a GER(20) of almost 7% in clean conditions to about 11% in the worst conditions.

When MPF channel reliability estimation was instead exploited, the GER(20) increase with decreasing SNR values, resulted the lowest compared to all other cases. The reason for this is that channel reliability estimation permitted to perform the f_0 estimation basing on the most noise-free channels, that is, those relative to microphones $M1$ and $M2$.

A second test, that was carried out to test the usefulness of weights c_i, considered the whole set of DMN channels. In this case microphones $M5$, $M6$ and $M7$ were contaminated with babble noise with decreasing SNR. As the Figure 5.8 reports, GER(20) values estimated in the noise-free scenario are the same showed with black curves in Figure 5.3. All algorithms performed worse with decreasing SNR values although MPF

provided the best results and its weights estimation based version limited performance deterioration due to the more difficult acoustic conditions. The three algorithms behaviour resulted similar to that shown in the lower panel of Figure 5.7 confirming thus the conclusions already drawn for that scenario.

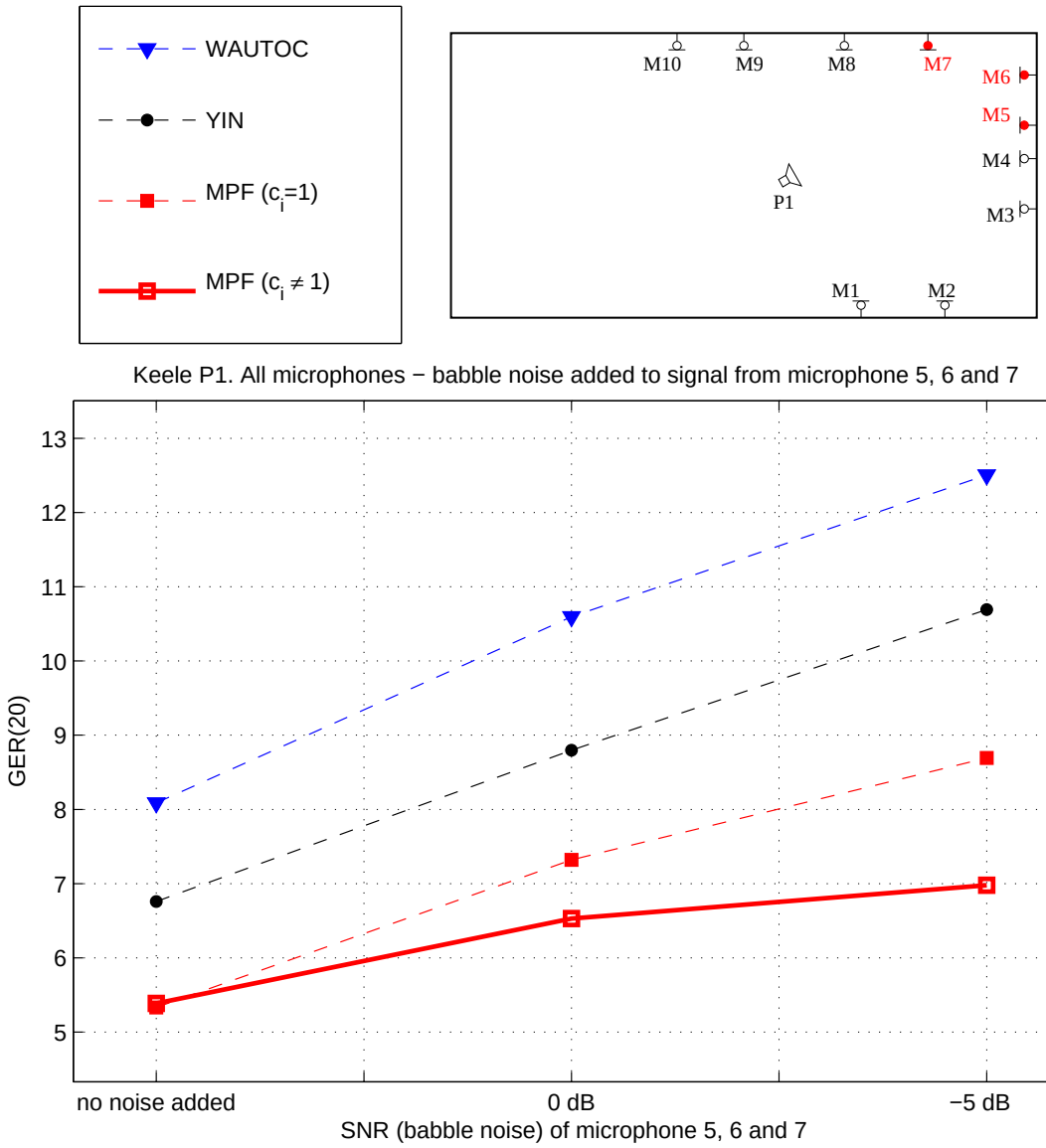

Figure 5.8: *Gross error rates obtained by the multichannel version of each algorithm under different noisy conditions. The whole set of DMN outputs was used and babble noise was added to channels 5, 6 and 7 at different SNR levels.*

5.3 CHIL

One of the speech corpora collected under the CHIL project[1], consists of 13 recordings, each about 5 minutes length, from female and male speakers extracted from real seminar sessions. These were scientific presentations, held at the Karlsruhe University, and the main difference with the Keele corpora, is that in this case spontaneous speech is dealt with. Each speaker, during the talk, wore a "Countryman E6" close-talking microphone, to capture a noise-free, non reverberant speech signal, and moved freely in the area labeled "speaker area", showed in Figure 5.10. Other distant-talk microphones were used for the recordings and will be described in the next section. The sampling frequency of the recorded signal was set to 44.1 kHz.

To obtain the reference pitch labels, three existing pitch extractor algorithms were used:

Praat: a computer program with which phoneticians can analyze, synthesize, and manipulate speech [11];

SFS: a free computing environment for conducting research into the nature of speech [47];

WaveSurfer: a multi-platform open source application, for speech/sound analysis and sound annotation/transcription [100].

To merge the tern of pitch estimates provided by the three PDAs at each processed frame, their variance was computed and, in case it was below a certain threshold δ, the mean was retained as the merged pitch value. Otherwise, 0 was assigned to the final reference value, with the convention that a null pitch estimate means that the underlying speech segment is to

[1]Computers in the Human Interaction Loop (CHIL) is an Integrated Project (IP 506909) under the European Commission's Sixth Framework Program. A description of the used speech corpora can be found at http://chil.server.de, http://www.nist.gov/speech and http://www.clear-evaluation.org.

be considered unvoiced. As a result of this approach, some discontinuities due to the estimates that were forced to 0, resulted in the series of the final pitch values. To overcome this problem, all the voiced segments that resulted shorter than 50 ms after the merging procedure, were regarded as unvoiced. The overall duration of the voicing speech sections resulting from this labeling process depended thus on the value of δ. Setting it to lower values, provided more precise estimates at the expense of the final amount of available voiced speech data.

Figure 5.9, shows in the top panel a portion of pitch estimates from each of the above cited algorithms. Some discontinuities and mismatches are visible where the circles of different colors do not superimpose perfectly.

To obtain 31% of voicing parts[2] out of the whole dataset, and precise estimates, a value of $\delta = 3\ Hz$ was chosen. Using this setting, an example of the result deriving from the merging procedure applied to the values showed in the upper panel, is reported in the bottom panel of the Figure 5.9.

The pitch values obtained with this method, can be considered a very reliable reference against which to test the performance of the algorithm proposed in this thesis. In fact, it is unlikely, even if not impossible, that all the three PDAs described above provide the wrong estimate. But each PDA bases on a different internal algorithm and the probability that all of them provide exactly the same wrong estimate, can be considered a rare case. Nevertheless, it could be that a few references can still result wrong, but considering the amount of data collected for testing, the latter will reduce to an insignificant percentage.

[2]This corresponded to about 20 minutes of voicing parts, for a total amount of about 125000 pitch reference values.

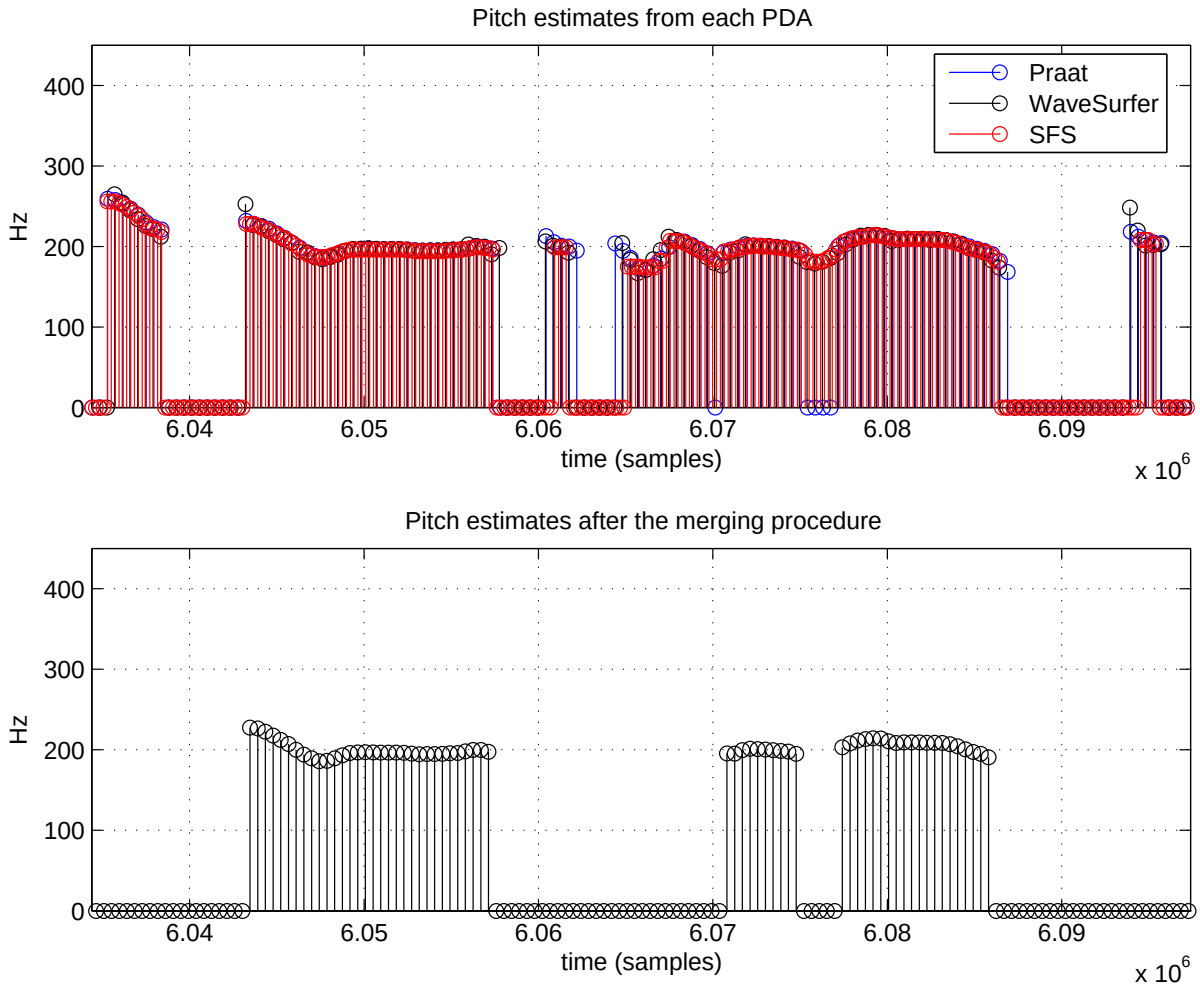

Figure 5.9: *The top panel shows the pitch estimates obtained from the CHIL speech corpora using the Praat, WaveSurfer, and SFS algorithms, respectively. The merging procedure which creates pitch reference labels for this speech dataset, considers only values from each PDA which are very close to each other. The resulting reference is plotted in the bottom panel.*

5.3.1 Scenario

In Figure 5.10, the plan of the CHIL room prepared at the Karlsruhe University for seminars and meetings recording is shown. The room is 7.10 m $\times$ 5.90 m wide and the ceiling height is 3 m. There is one entrance in the north wall, and two more doors in the south wall leading to other offices. The room was filled with different audio/video sensors, since it

was prepared to be used in the CHIL project context, which will be briefly outlined in Chapter 7. Among others devices, some of which not shown in the figure, 4 fixed color cameras positioned in the corners, and 4 inverted "T"-shaped microphone arrays (drawn in magenta) are shown, as well as 4 single-distant microphones placed on the top of a table.

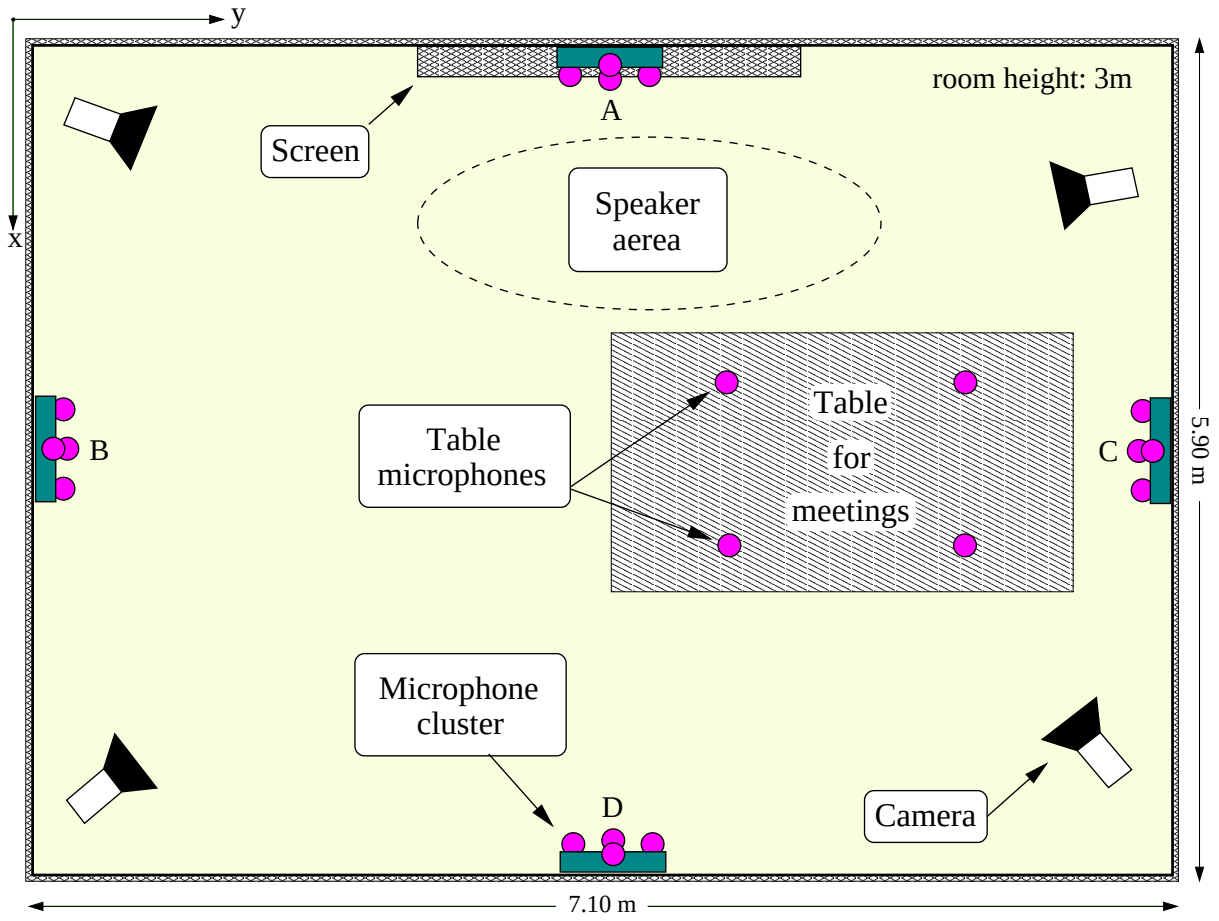

Figure 5.10: *Plan of the CHIL seminar and meeting room at the Karlsruhe University. Four cameras were placed at each corner and four inverted "T"-shaped microphone arrays (magenta color), labeled with letters "A", "B", "C" and "D", are positioned as shown. The four single microphones on the table and other devices not shown in the figure were not used for the test described in this thesis.*

As shown in the figure, the microphone arrays are labeled with the letters "A", "B", "C" and "D", and their layout and coordinates are shown

in Table 5.5, where each microphone is assigned an index. Therefore, to refer for example to the 4 microphones of the "A" array, the labels "A1","A2","A3", and "A4" are used.

Microphone coordinates			
Array	x	y	z
A1	105	3060	2370
B1	2150	105	2290
C1	2700	6210	2190
D1	5795	4280	2400

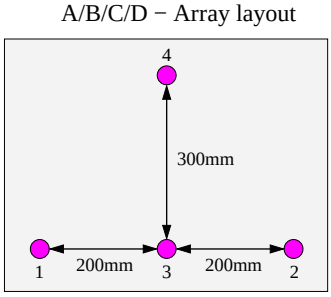

Table 5.5: Microphones coordinates of the inverted "T"-shaped arrays used in the CHIL meeting room at Karlsruhe University. Left table reports the coordinates x, y and x of the bottom-left microphone (labeled 1) of each array. Figure on the right shows the frontal view of each array and its microphones relative positions and distances.

For the experiments carried out in this thesis, the 16 outputs from the microphones arrays were recorded synchronously with the signal proceeding from the close-talk microphone worn by the speaker. After that, they were aligned to compensate for the propagation delay with which the talker speech reached each far microphone. Considering that the talker was moving during the speech, her/his average position was calculated, considering the speaker area shown in the room map.

5.3.2 Results

The CHIL multichannel database obtained as described previously, was used to test WAUTOC, YIN and MPF algorithms, described in Sections 4.1.1, 4.1.2, and 4.1.3, respectively. As done for the Keele database, three contexts have been considered for the evaluation. First, the seminar data recorded by the close-talk microphone was used, then each distant sensor of the microphone arrays was considered individually. Finally all channels were used jointly to test the multi-microphone versions of the above cited

algorithms. In all the three conditions, the analysis step was set to 10 ms, that is each PDAs computed 100 pitch estimates/sec. To optimize each algorithm performance, each algorithm was tested varying the analysis windows length and type (the latter only for WAUTOC and MPF). Setting the analysis windows to 30 ms, 40 ms, and 60 ms, for YIN, WAUTOC (rectangular window) and MPF (Hamming window) based algorithms, respectively, each algorithms achieved the best results.

The graphs reported in Figure 5.11 and 5.12, refer to pitch estimation results measured in terms of GER(20) and GER(5), respectively. The two-letter labels in the x-axis indicate the array and which of its microphone is considered, in accordance with the convention explained in Table 5.5. The top-right part of the figure recalls the relative position between the DMN elements and the talker, so that dependency of the results on the latter can be verified. On the y-axis, is shown the measured GER and different symbols are used to mark the results obtained from each PDA: "▼" for WAUTOC, "●" for YIN and "■" for MPF. To distinguish between the different analyzed contexts, the results have been plotted with different colors: red for the single distant channel context, black for the joint multi-microphone scenario and blue for the results obtained using the close-talk signal. Table 5.6 and 5.7 numerically summarize the results shown in the graphs.

Considering Figure 5.11, it is interesting to note the strong dependency of PDAs performances on the microphone position. This can be found out analyzing the red graphs which report the results obtained by the three PDAs applied on each distant microphone individually. It is evident as the "A" microphone array provided the best quality versions of the seminars. The GER(20) values relative to microphones "A1","A2","A3", and "A4"

are, in fact, the lowest ones, considering each method separately. This is due to the proximity of the capturing device to the speaker and by the fact that the latter often turns her/his head toward the screen, which is situated just beneath the microphone array. The curves show then an increasing GER(20) value, as the talker-microphone distance increases, and the trend is confirmed for each method.

In these conditions, WAUTOC provided the worse results, while MPF the best. The YIN algorithm gave instead, GER(20) values in between, even if closer to the WAUTOC curve.

GER(20)	WAUTOC	YIN	MPF
close-talk	1.60	**0.13**	0.14
single-mic	15.84	13.83	**6.30**
multi-mic	7.05	4.05	**2.15**

Table 5.6: Gross error rates (20%) obtained applying WAUTOC, YIN and MPF, respectively, to the CHIL speech dataset. Values refer to the curves depicted in Figure 5.11 and in the second row the averages, computed for each red curve, are reported. Bold font is used to indicate the best result obtained in each acoustic condition.

The lowest GER(20) achievable by each of the analyzed algorithms, is represented by the blue curves which report the results relative to the close-talk speech signals. YIN and MPF provided the same result, i.e., GER(20) = 0.14%, while WAUTOC performed a little worse, GER(20) = 1.6%. This very low values however, are due to the method used to derive the reference pitch estimates. To evaluate WAUTOC, YIN and MPF based algorithms, only the voicing sections where all the PDAs used to obtain the reference values, were used. This means that the voicing sections of the close-talk signal, where the pitch estimation was more difficult, have not been considered for the evaluation.

When multi-microphone versions of the three algorithms are tested

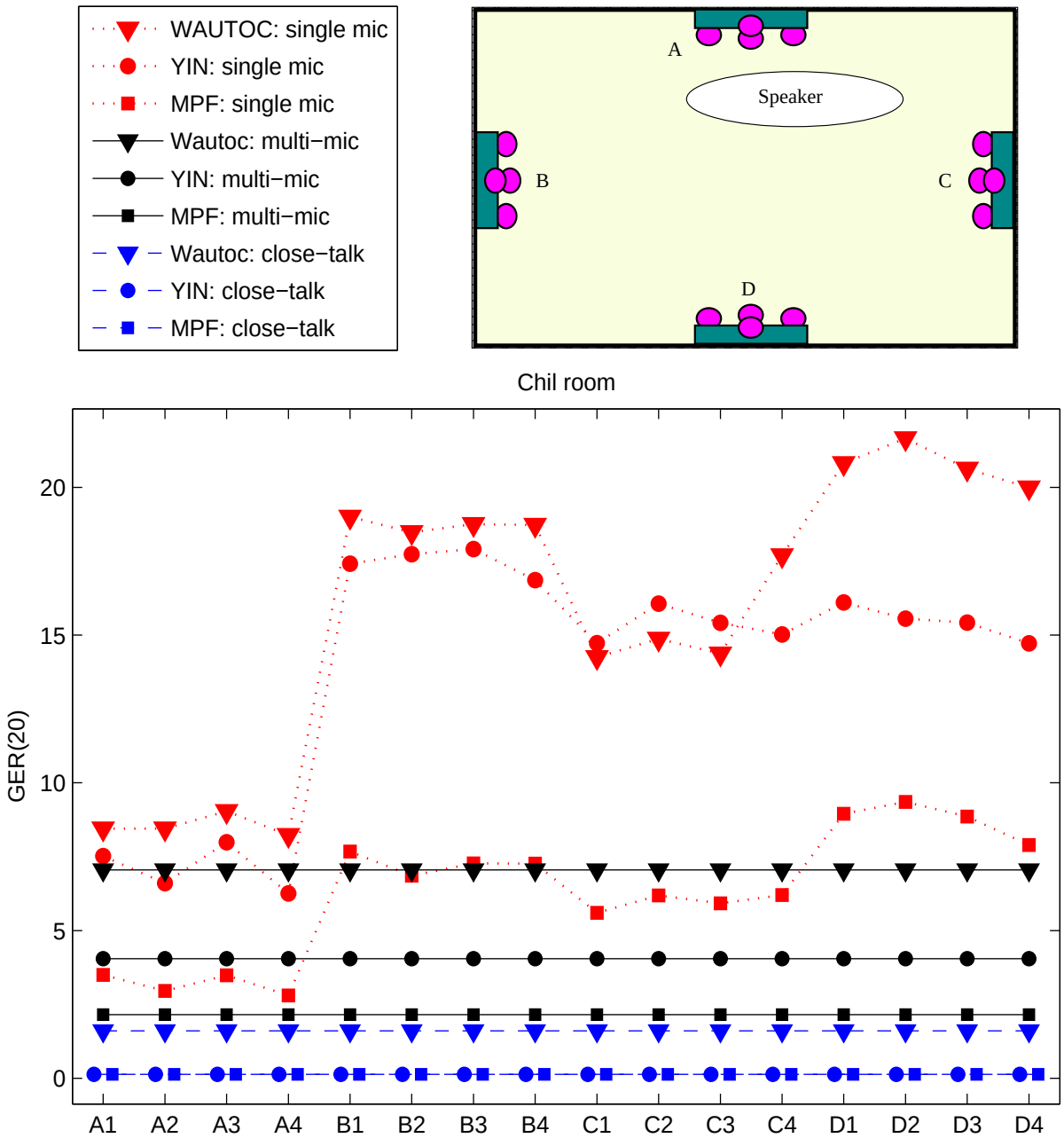

Figure 5.11: *The three red curves show gross error rates (20%) derived by applying WAUTOC (▼), YIN (●) and MPF (■), respectively, to each of the* 16 *microphone signals. The six horizontal lines indicate the performance provided by each of the three algorithms on the close-talking signals (blue) and the corresponding performance obtained by their multichannel version applied to all the far microphone signals (black).*

(black lines), the trend provided by the single distant microphones is confirmed. Even if YIN and WAUTOC algorithms demonstrated the best relative improvement, compared to the single channel case, in this case too, MPF provided the lowest GER(20), 2.16%, compared to the values of 4.05% and 7.05% given by YIN and WAUTOC, respectively.

In case GER(5) is used to compare PDAs performance, the results reported in Figure 5.12 have to be considered. The relative positions of the curves are not very much different to those of the previous figure. Apart from the close-talk case, where YIN performs slightly better than MPF (0.23% against 0.28%), and WAUTOC provides the worse result, the single-distant (red) and multi-microphones (black) curves follow the same trend of the counterpart GER(20) curves.

GER(5)	WAUTOC	YIN	MPF
close-talk	1.99	**0.23**	0.28
single-mic	20.82	18.56	**11.75**
multi-mic	11.52	8.02	**6.78**

Table 5.7: Gross error rates (5%) obtained applying WAUTOC, YIN and MPF, respectively, to the CHIL speech dataset. Values refer to the curves depicted in Figure 5.12 and in the second row the averages, computed for each red curve, are reported. Bold font is used to indicate the best result obtained in each acoustic condition.

Exploiting the whole set of contributes provided by the DMN, guaranteed better results with all algorithms. But the overall distance between the black curves and the blue ones, which represents the lower bound for the GER(5), resulted higher than in the GER(20) case. This demonstrates the difficulty to recover a very precise pitch estimate from the signal degraded by the reverberation effect (see waveforms examples in Figure 4.3).

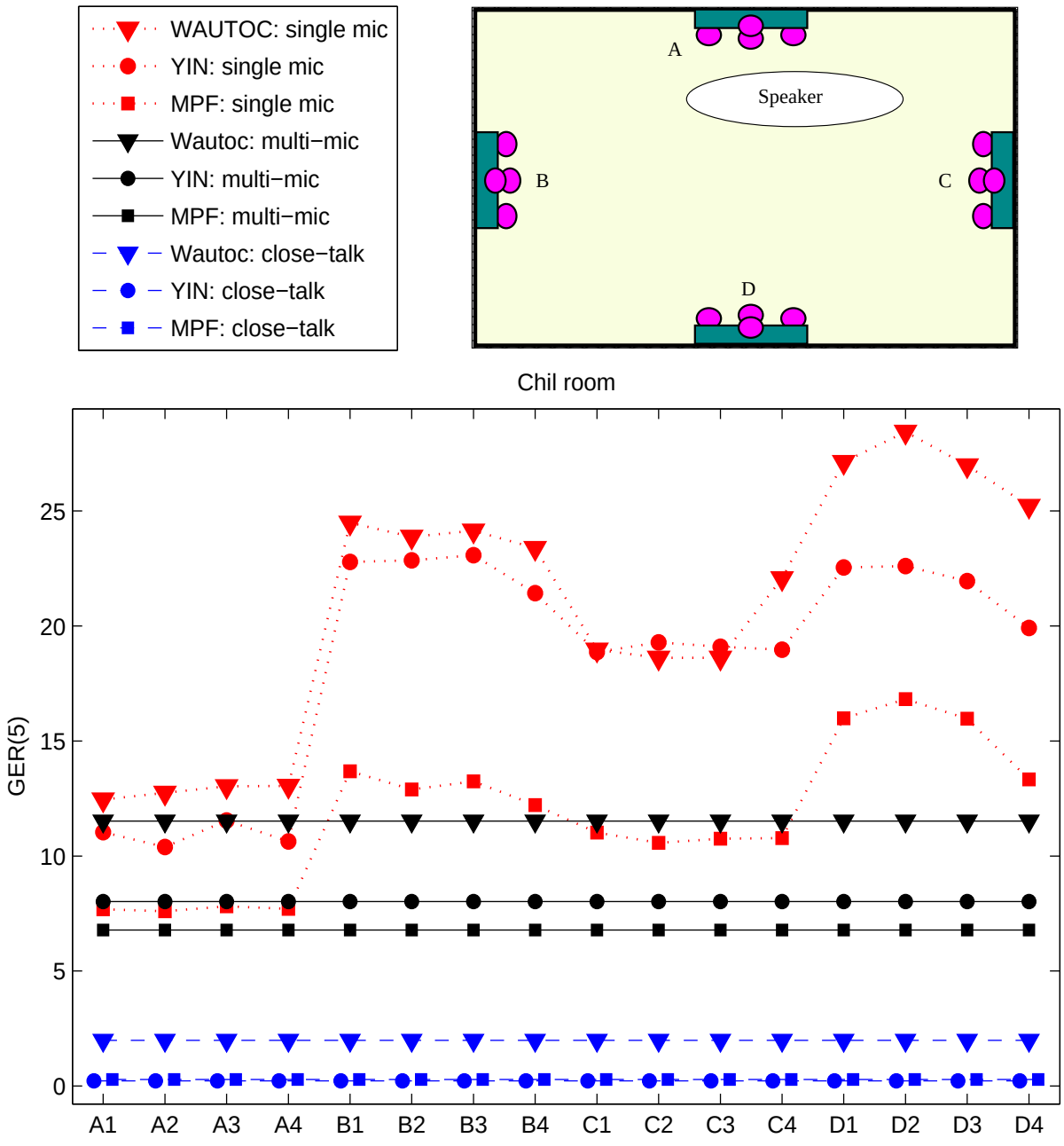

Figure 5.12: *The three red curves show gross error rates (5%) derived by applying WAUTOC (▼), YIN (●) and MPF (■), respectively, to each of the* 16 *microphone signals. The six horizontal lines indicate the performance provided by each of the three algorithms on the close-talking signals (blue) and the corresponding performance obtained by their multichannel version applied to all the far microphone signals (black).*

Chapter 6

f_0 in Blind Source Separation

As reported in Section 3.3.7, pitch information can be also used to recover and enhance the individual outputs of a Blind Source Separation System (BSS) [30]. The experiments reported in this thesis[1] are based on an underdetermined (more speakers than sensors) system based on binary masks, which will be briefly described in Section 6.1. The results obtained, in terms of pitch estimation accuracy, of Signal to Interference Ratio (SIR), and of Signal to Distortion Ratio (SDR), will be presented in Section 6.2.

6.1 Binary mask based BSS

A commonly used setup for a BSS system in a real environment considers M sensors observing N signals, which are modeled as convolutive mixtures

$$x_j(n) = \sum_{i=1}^{N} \sum_{l=1}^{L} h_{ji}(l) s_i(n-l+1), \quad j = 1, \cdots, M, \tag{6.1}$$

where $s_i(n)$ represents the i-th source, $x_j(n)$ the signal observed by the j-th sensor, and $h_{ji}(n)$ the room impulse response of length L, which models

[1] The activity presented in this chapter was conducted while I was at the NTT Communication Science Laboratories, Kyoto, JAPAN.

the delay and reverberation room effects from the i-th source to the j-th sensor.

Here, the under-determined case is addressed, that is, $N > M$, with $N = 3$ and $M = 2$ and separation is carried out in the time-frequency domain. In this domain, speech signals sparseness can be assumed [12], and the convolutive mixtures of Equation 6.1 can be written in terms of instantaneous mixtures

$$\begin{bmatrix} X_1(\omega, m) \\ X_2(\omega, m) \end{bmatrix} = \begin{bmatrix} H_{11}(\omega, m) & H_{12}(\omega, m) & H_{13}(\omega, m) \\ H_{21}(\omega, m) & H_{22}(\omega, m) & H_{23}(\omega, m) \end{bmatrix} \begin{bmatrix} S_1(\omega, m) \\ S_2(\omega, m) \\ S_3(\omega, m) \end{bmatrix}, \quad (6.2)$$

or, in matrix notation,

$$\mathbf{X}(\omega, m) = \mathbf{H}(\omega, m)\mathbf{S}(\omega, m). \quad (6.3)$$

The variables ω and m indicate the frequency and frame indexes of the short-time Fourier transforms of the sources $\mathbf{S}(\omega, m)$, the observed signals $\mathbf{X}(\omega, m)$, and of the mixing matrix $\mathbf{H}(\omega, m)$, respectively. Each j, i-th component of the latter 2×3 matrix represents the transfer function from the i-th source to the j-th sensor.

In the determined or overdetermined case, the inverse of the mixing matrix $\mathbf{H}(\omega, m)$ can be computed and used to easily solve Equation 6.3 for the sources values S_i. Considering the underdetermined case though, the solution is not straightforward since the mixing matrix, as in this example, is not invertible. To solve the under-determined BSS problem, several methods based on source sparseness have been proposed [12, 88].

The method that will be explained in the following and whose building blocks are reported in Figure 6.1, exploits the sparseness assumption and supposes consequently, that most of the signal samples can be considered

null in the given domain. This makes it possible to assume that sources overlap at rare intervals [10]. Given this hypothesis, each target speaker can be extracted by selecting from the mixture just those time-frequency bins at which the speaker is considered to be active or predominant.

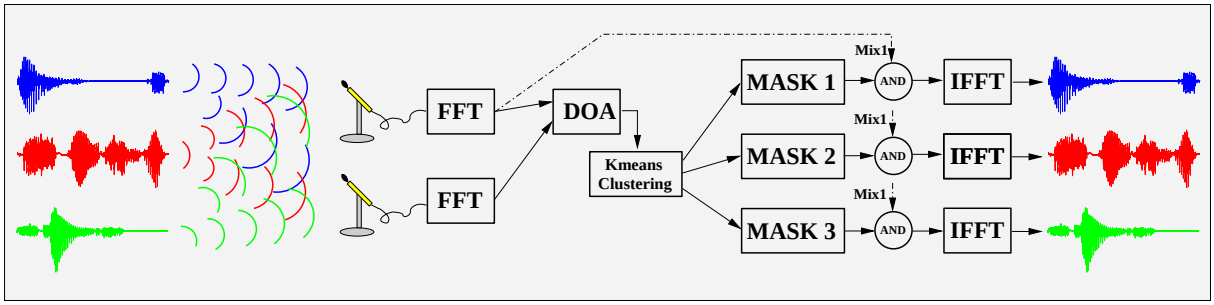

Figure 6.1: *The scheme shows the basic building blocks of an underdetermined (three speakers, two microphones) BSS system. A binary mask, designed exploiting the Direction Of Arrival (DOA) of each speaker signal, is applied to the common time-frequency representation to extract each output.*

One way to localize such time-frequency bins, is to use the microphones observations $X_1(\omega, m)$ and $X_2(\omega, m)$, and compute their phase difference as follows

$$\varphi(\omega, m) = \angle \frac{X_1(\omega, m)}{X_2(\omega, m)}. \tag{6.4}$$

The result of Equation 6.4 permits then to obtain the Direction Of Arrival (DOA) for each time-frequency bin, computed as

$$\theta(\omega, m) = \cos^{-1}\left\{\frac{\varphi(\omega, m) \cdot c}{\omega \cdot d}\right\}, \tag{6.5}$$

where c is the speed of sound and d is the microphone spacing. For each frequency index, computing the histogram of $\theta(\omega, m)$ reveals three peaks centered approximately on the actual DOA of the sources (an example is given in Figure 6.2), which can therefore be estimated by employing a clustering algorithm such as k-means.

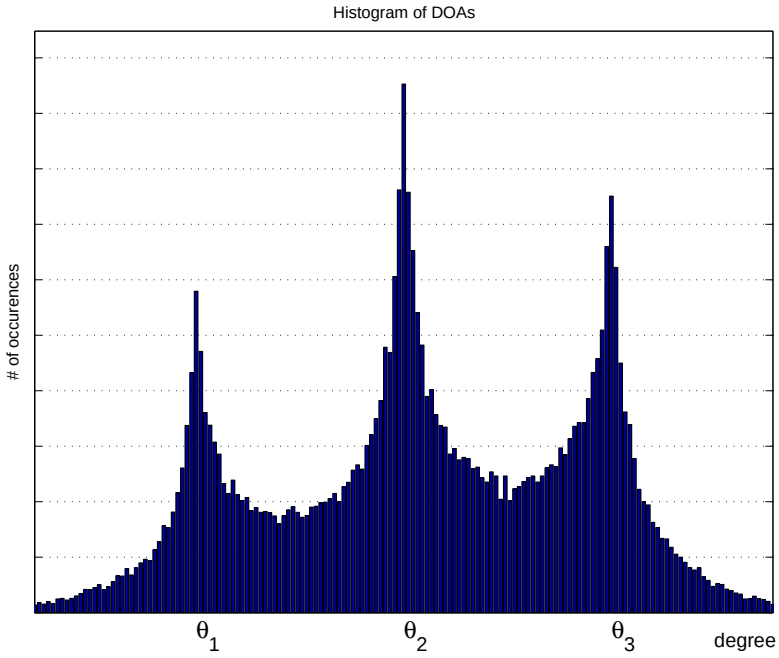

Figure 6.2: *Histogram of the DOAs computed from Equation 6.5. The peaks of the histogram are centered on the actual directions of arrival, θ_1, θ_2 and θ_3, of the three speakers talking at the same time.*

If the centroids provided by the clustering algorithm are indicated with $\tilde{\theta}_1$, $\tilde{\theta}_2$ and $\tilde{\theta}_3$ where $\tilde{\theta}_1 \leq \tilde{\theta}_2 \leq \tilde{\theta}_3$, the binary masks can be obtained as follows

$$M_k(\omega, m) = \begin{cases} 1, & \tilde{\theta}_k - \Delta \leq \theta(\omega, m) \leq \tilde{\theta}_k + \Delta \\ 0, & \text{otherwise} \end{cases} \qquad k = 1, 2, 3 \qquad (6.6)$$

where Δ is an extraction range parameter that determines the trade-off between the separation performance and sound fidelity. To finally extract each target speaker $Y_k(\omega, m)$, the three binary masks obtained with Equation 6.6 are applied to the speech mixture, using

$$Y_k(\omega, m) = M_k(\omega, m) X_j(\omega, m), \quad j = 1 \text{ or } 2, \quad k = 1, \ldots, 3. \qquad (6.7)$$

For each couple of time-frequency indexes (ω, m), $Y_k(\omega, m)$ is assigned the same value of the mixture $X_j(\omega, m)$, in case $M_k(\omega, m) = 1$, otherwise $Y_k(\omega, m)$ is set to 0. The last algorithm step, is to convert the short-time Fourier transforms $Y_k(\omega, m)$ back in the time domain, by means of the IFFT, to finally obtain the individual speech contributes $y_k(n)$, $k = 1, \cdots, 3$.

Figure 6.3 shows a graphical example of the process just described. On the left it displays the spectrograms computed from the speech signal of each talker, recorded individually. When the speakers are active at the same time, what is actually recorded by the microphone is the mixture reported in the middle upper part of the figure. This shows the sum of the time-frequency contributes of each speaker. As it can be seen, there are regions where they overlap, while there are other areas where they do not.

The binary masks, obtained for this example, are shown in the middle lower part of the figure. To show which time-frequency bins are set to 1 for each of the $M_1(\omega, m)$, $M_2(\omega, m)$ and $M_3(\omega, m)$ masks, the blue, red and green colors were used, respectively.

The right column of Figure 6.3, shows the results obtained after masks $M_i(\omega, m)$ were applied to the speech mixture. Comparing these spectrograms with those displayed at the left, the time-frequency regions with high energy, characteristic of voiced speech, are recognizable. Unfortunately, the regions were the speaker contributes were overlapping, result considerably deteriorated.

The binary mask method just described, results in too much discontinuous zero-padding of the extracted signals, producing distortion and musical noise. This side effect is clearly visible, for example, in the right part of the mixture spectrogram shown in the figure, where high energy regions (i.e. formants of voiced segments) that belong to different speakers, overlap.

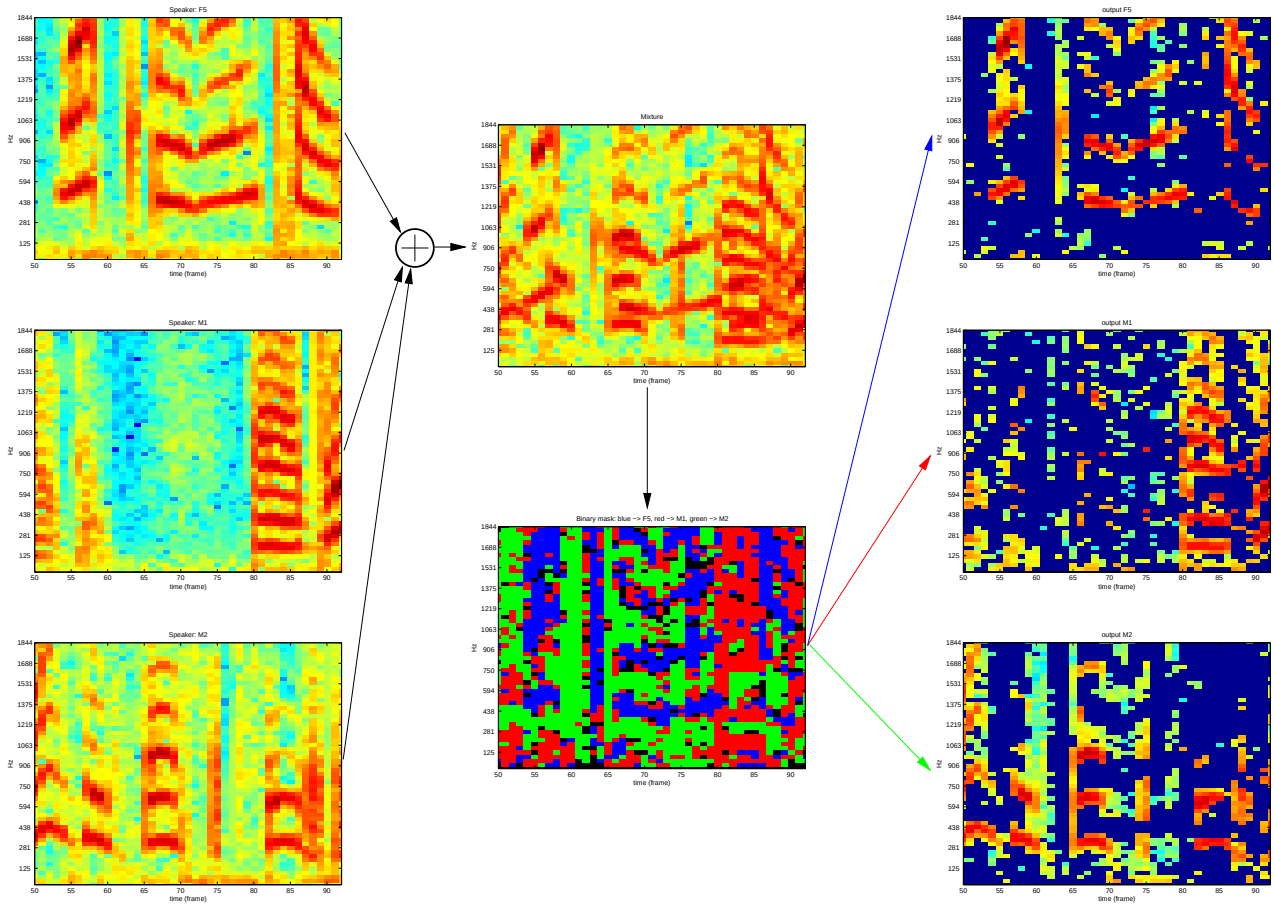

Figure 6.3: *On the left side are represented the spectrograms computed on a speech segment uttered by speakers $F5$, $M1$ and $M2$, respectively. The mixture spectrogram provided by each microphone is shown at the top of the center column. Below it, the estimated binary mask is plotted, using the blue, red and green colors, to indicate the time-frequency locations that will be used for the spectrogram reconstruction of the $F5$, $M1$ and $M2$ speakers, respectively. The latter are shown in the right column.*

However, as it will be shown in the following, exploiting the knowledge of speech related features, such as the fundamental frequency information, makes it possible to partly recover the original signal structure, thus improving the performance of the BSS system. For this, first a different masking method is proposed, in order to precondition the signal for the final f_0 based comb-filtering processing.

6.1.1 Continuous mask based BSS

Although effective, the binary mask approach introduces musical noise [4]. To mitigate this side effect, the use of a continuous mask in place of a binary mask, is proposed here. As seen in the previous section, the sparseness assumption is not always satisfied for all the time-frequency bins in the mixture spectrogram. For these bins, the DOA cannot be properly estimated and, consequently, the correspondent values in the binary masks will be wrongly assigned.

A simple way to improve this situation, is to design continuous masks, using the distance of each $\theta(\omega, m)$ from the estimated centroids $\tilde{\theta}_i$, as a "reliability" indicator for the underlying mixture value $X_j(\omega, m)$.

This is easily obtained assigning each $M_j(\omega, m)$ a value proportional to that distance. An example of this is reported in Figure 6.4: based on the distance of $\theta(\omega, m)$ from each centroid indicated in the x-axis, linear interpolation is used to assign to mask $M_1(\omega, m)$, $M_2(\omega, m)$ and $M_3(\omega, m)$, the values marked with the blue, red and green circles, respectively.

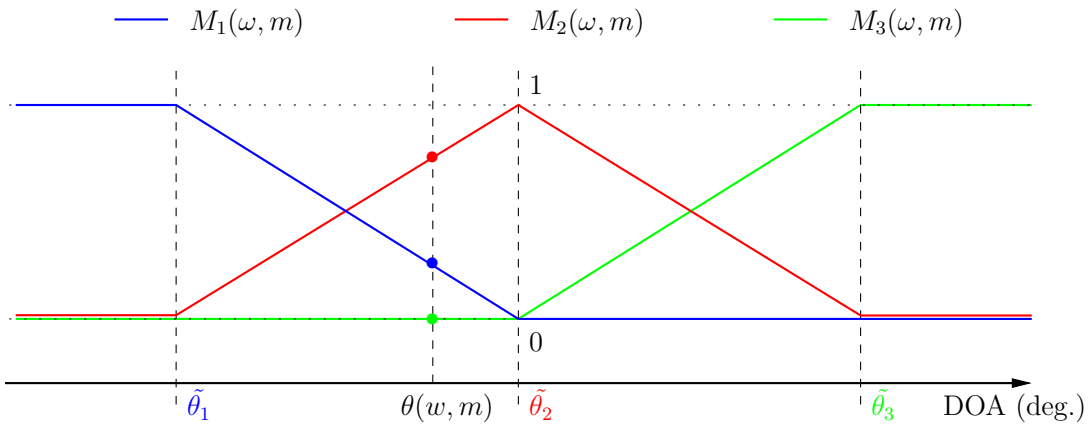

Figure 6.4: *The graph shows the continuous mask obtained by means of linear interpolation of the DOA of each speaker. Given the current estimated DOA, $\theta(\omega, m)$, the time-frequency bin with coordinates (ω, m) of mask $M_1(\omega, m)$, $M_2(\omega, m)$ and $M_3(\omega, m)$, is assigned the value marked with the blue, red and green circle, respectively.*

As shown in the graph, the mask value $M_3(\omega, m)$, is set to 0, since the estimated DOA lies in between the first and the second actual DOAs. Moreover, being the latter closer to $\tilde{\theta}_2$, $M_2(\omega, m)$ will be given the highest coefficient (red circle), since it is more likely that the mixture bin with position (ω, m), belongs to the second speaker. Other alternatives to the linear interpolation are polynomial interpolation or directivity pattern based masks, as described in [5].

6.1.2 f_0 driven comb filtering based BSS

As stated in the previous section, the outputs of a binary masks based BSS system result distorted as a consequence of the fact that the sparseness assumption is not always satisfied. Applying continuous masks, instead of binary ones, demonstrated to be more beneficial for reducing the overall distortion than cross-speaker interference.

In fact, each signal $y_k(n)$ extracted by means of continuous masks accounts for the target speaker $s_i(n)$, $i = k$, and a certain amount of residual interference due to interfering speakers $s_i(n)$, $i \neq k$. To improve separation and sound quality, thus reducing musical noise, an extra processing stage is employed as shown in Figure 6.5. In the scheme proposed here, the f_0−VUV estimation block is responsible for estimating both the fundamental frequency and the voiced/unvoiced (V/UV) information from each of the extracted signals $y_k(n)$. Each signal f_{0_k} will then be used to tune one different adaptive FIR or IIR filter, which will be active only on voiced segments indicated by the VUV_k signal with which it is driven.

The FIR filter is responsible for the harmonic enhancement of the target speaker $y_k(n)$, while IIR filters suppress the interference caused by the other speakers in the mixture. The final output $y'_k(t)$ is then obtained by selecting the FIR filter output for speech segments labeled as voiced, and the IIR filter output for unvoiced segments. To drive this selection, the

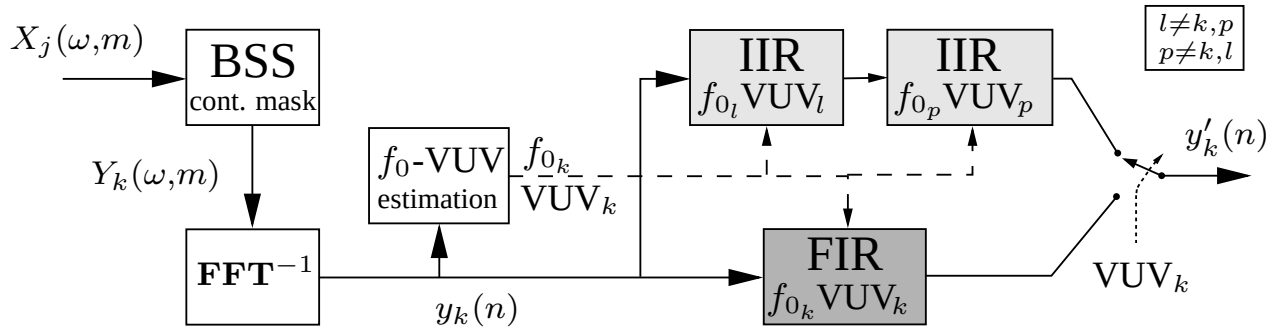

Figure 6.5: *After blind source separation, each output $y_k(n)$ is processed by a PDA to extract the pitch information f_{0_k}, as well as the voiced/unvoiced (V/UV) information* VUV$_k$. *These signals are used to drive FIR and IIR comb filters that enhance the deteriorated harmonic structure of voiced segments (FIR) and remove the interference due to the voicing parts of the competing speakers.*

signal VUV$_k$ is employed.

Harmonic enhancement of target speaker

To enhance the voicing sections of each output $y_k(n)$, an adaptive FIR comb filter is used [32]. Figure 6.6 shows an example of the FIR impulse response $h(n)$ (plotted in red), superimposed to the speech segment currently analyzed (black waveform).

In the figure, successive pitch periods are indicated with labels T_{m-1}, T_m, T_{m+1}, T_{m+2}. Since the fundamental frequency is not constant during phonation, they vary with time, that is, $T_r \neq T_f$, for $r \neq f$. Therefore, to take into account these pitch values fluctuations, the spacing between the values a_i of the filter impulse response, is continuously adjusted to coincide with the spacing of the individual pitch periods T$_r$ of the waveform being processed. At each time instant, the pitch period value is provided by the f_{0_k} signal, which was previously estimated from the voiced parts of $y_k(n)$, and is used to tune the filter.

The effect of this filtering procedure is that of averaging successive pitch

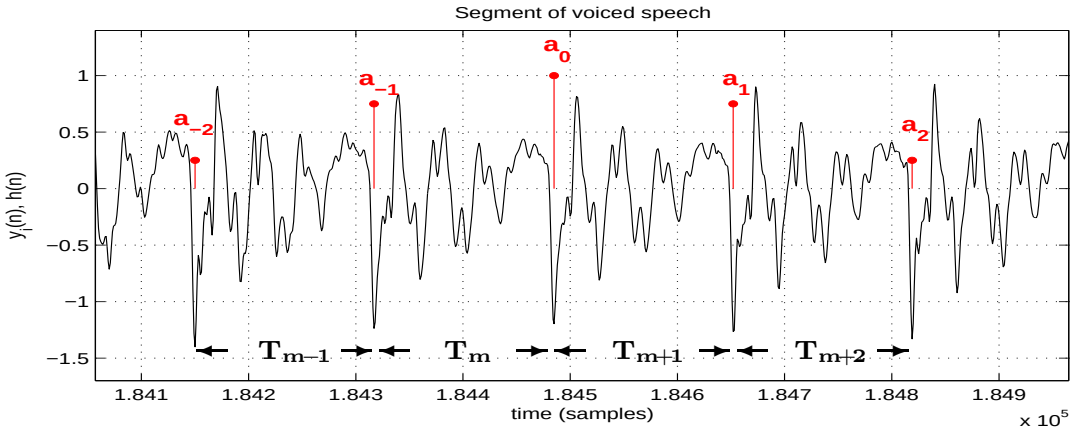

Figure 6.6: *Adaptive FIR filter (red) and speech waveform (black) with varying pitch period. The FIR filter coefficients a_i are plotted with red circles superimposed to the voiced speech segment currently being processed. The spacing between each coefficient is adjusted using the pitch information, which, at each time instant, provides the different pitch periods T_r values.*

periods of the target speaker, so that they will add constructively. Since residual components from interfering speakers do not exhibit such periodic behaviour, they will be further reduced by the averaging procedure. This results in the restoration of harmonic components continuity, being advantageous for reducing musical noise.

The choice of a FIR filter for performing harmonic structure enhancement, is motivated by the fact that this filter adapts faster to f_0 fluctuations and has linear phase characteristics. The filter impulse response coefficients a_i are obtained from a Hanning window of length N_{FIR}. An example of the filter frequency response obtained using $N_{\text{FIR}} = 5$ coefficients, adapted to match the pitch periods of a voiced segment with $f_0 \approx 155\ Hz$, is plotted at the top of Figure 6.7.

Removal of harmonics of interfering speakers

While the FIR filter enhances the voiced sections of the target speaker $y_k(n)$, IIR filters are given the task of removing interferences of competing speakers. This is carried out by filtering the $y_k(n)$ sections which are unvoiced at the same time while the competing speakers are voicing. The filter used is an adaptive IIR comb filter [68], with a transfer function given by

$$H(z) = \frac{\prod_{k=1}^{N_{\text{IIR}}}(1 + \alpha_k z^{-1} + z^{-2})}{\prod_{k=1}^{N_{\text{IIR}}}(1 + \rho\alpha_k z^{-1} + z^{-2})}, \tag{6.8}$$

where $\alpha_k = -2\cos(k\omega_0)$ and $\omega_0 = 2\pi f_0$. To avoid filter instability, parameter ρ must be set to $\rho < 1$. This variable is used to control the $H(z)$ steepness: the closer to 1 its value is set, the more notch-like becomes the transfer function at the frequency points multiple of f_0. This comes at the expense of the transient state length, which becomes larger for increasing values of ρ.

A plot of the transfer function specified in Equation 6.8 is given in the bottom part of Figure 6.7. In this example, the parameter N_{IIR}, which determines the number of harmonics to be canceled out, is set to 5 and $f_0 = 120\ Hz$. The zeros occurring at the frequency values $f = k \cdot f_0,\ k = 1, \ldots, 5$, are responsible for canceling out the harmonic structure of interfering speakers, which could not be separated previously by the BBS system.

As the scheme of Figure 6.5 shows, the IIR filtering is employed twice, first setting ω_0 at the f_{0_l} values of the first interfering speaker, $(l \neq k, p)$, then with the f_{0_p} values of the second interfering speaker, $(p \neq k, l)$. In this way, harmonics relative to the voiced segments of interfering speakers s_i, $i \neq k$, are removed from signal $y_k(n)$.

Despite its nonlinear phase characteristic, this filter is suitable for har-

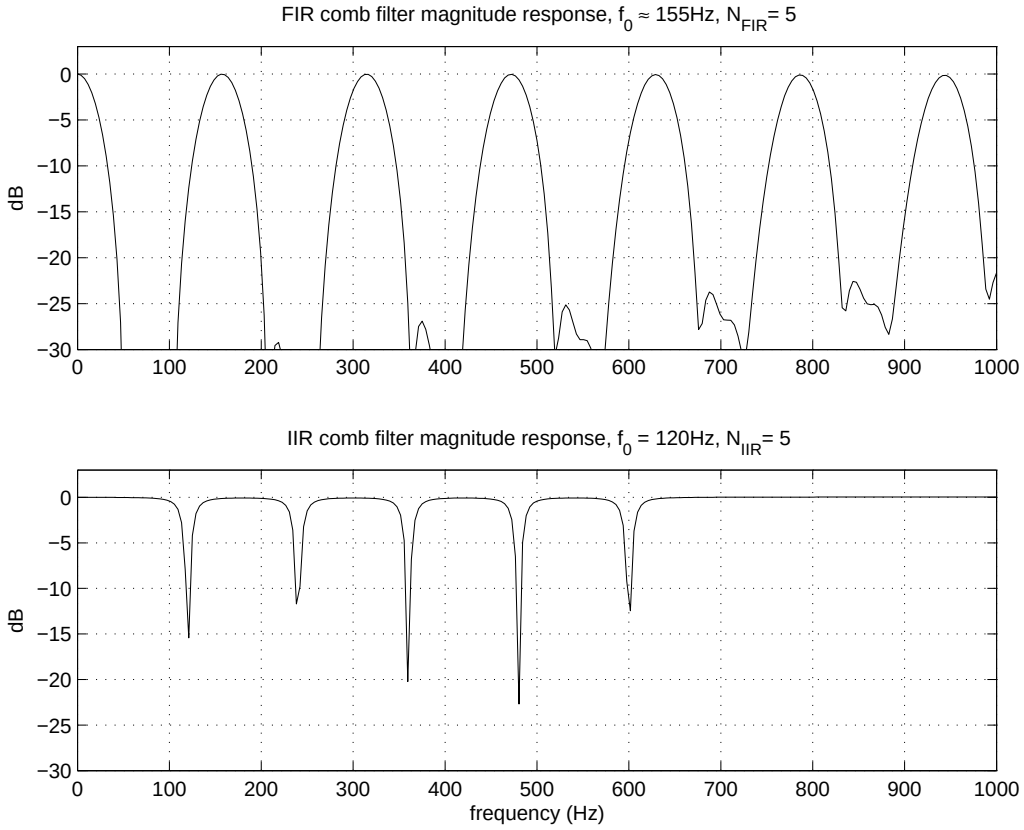

Figure 6.7: *Frequency response of FIR and IIR comb filters.*

monics removal. This because it provides a more abrupt and higher cutoff ratio in the frequency locations of interest than its FIR counterpart. The latter in fact, must have a short impulse response to satisfy the quasi-stationarity assumption valid for voiced segments.

6.2 BSS performance

This section describes the results obtained applying the f_0 based method just described, to enhance the quality of the outputs of a binary mask based Blind Source Separation (BSS) system. Several factors affect the overall performance of such an extended BSS system as, for example, the reverberation level of the considered environment, the characteristics of the speech

inputs, and the PDAs ability to estimate the pitch values correctly. Therefore, to evaluate the proposed BSS system performance, speech input data was carefully prepared to include both the reverberant and non-reverberant scenario, and several (?) pitch extraction techniques were tested. Results are thus given both in terms of Signal to Distortion Ratio (SDR), Signal to Interference Ratio (SIR), and in terms of GER(20) and RMSE(20).

6.2.1 Error measures

To measure the separation performance and sound quality of a BSS system, the Signal to Interference Ratio (SIR) and Signal to Distortion Ratio (SDR) were used, respectively. For each separated output, SIR takes into account the amount of energy of the signal components which belong to the target speaker, and of those belonging to interfering speakers, measuring their proportion. SDR instead, provides an indication about the difference between the signal of the target speaker, acquired when the other speakers are not active, and the same signal extracted from the mixture by the BSS system. The SIR and SDR expressions are

$$\mathrm{SIR}_k = 10 \log \frac{\sum_n y^2_{ks_k}(n)}{\sum_n (\sum_{i \neq k} y_{ks_i}(n))^2} \tag{6.9}$$

$$\mathrm{SDR}_k = 10 \log \frac{\sum_n x^2_{js_k}(n)}{\sum_n (x_{js_k}(n) - \alpha y_{ks_k}(n - D))^2} \tag{6.10}$$

Indicating with s_k the speech signal generated by the k-th speaker, and with y_k the relative output provided by the BSS system, the following meaning is given to variables of Equations 6.9 and 6.10: y_{ks_i} is the k-th separating system output when only s_i is active and s_l, $l \neq i$ is silent; x_{js_k} is the observation provided by microphone j when only s_k is active. Parameters α and D are used to compensate for the amplitude and phase

difference between x_{js_k} and y_{ks_k}. To evaluate the performance of the proposed method, SIR and SDR are computed using measurements from both microphones and the best value is retained.

The proposed BSS system exploits pitch information to improve its separation performance which, in turn, depends on the accuracy and the resolution with which the employed PDA estimates the pitch values. To show the dependency of separation performance on pitch estimation quality, the GER(20) and RMSE (or "fine pitch error") measures will be computed. These error measures were previously described in Section 5.1.1.

6.2.2 BSS scenario

In Figure 6.8 the setup used for the BSS experiments is shown. Speakers positions are indicated with loudspeaker symbols, each of which refers to signals s_1, s_2 and s_3, respectively. The DOAs for the three speakers was set to 45°, 90° and 135°, respectively, and the distance microphones-speaker, for the reverberant case, was set to 1.1 meters. Two omnidirectional microphones, distant 4 *cm* from each other, were used and are marked with circles.

To simulate an anechoic environment, i.e. $T_{60} = 0\,ms$, mixtures $X_j(\omega, m)$ were obtained computing Equation 6.2 with values $H_{ji}(\omega)$ set as follows

$$\begin{bmatrix} X_1(\omega,m) \\ X_2(\omega,m) \end{bmatrix} = \begin{bmatrix} e^{(j\omega\tau_{11})} & e^{(j\omega\tau_{12})} & e^{(j\omega\tau_{13})} \\ e^{(j\omega\tau_{21})} & e^{(j\omega\tau_{22})} & e^{(j\omega\tau_{23})} \end{bmatrix} \begin{bmatrix} S_1(\omega,m) \\ S_2(\omega,m) \\ S_3(\omega,m) \end{bmatrix}, \tag{6.11}$$

where τ_{ji} represents the time delay with which sound propagates from the i-th speaker to the j-th microphone. Its value is computed as $\tau_{ji} = \frac{d_j}{c}\cos\theta_i$, being d_j the j-th microphone position, and θ_i the i-th source direction.

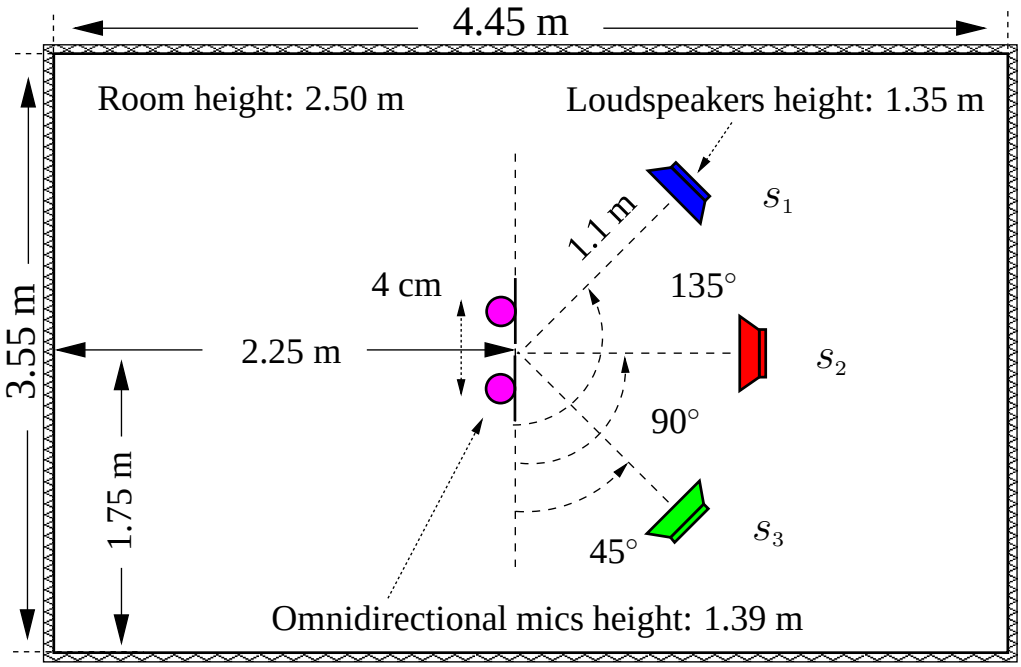

Figure 6.8: *Room for BSS tests. The setup used comprises* 2 *microphones (black circles) and* 3 *loudspeakers (used to reproduce messages* s_i, $i = 1, 2, 3$*) positioned as shown.*

For the reverberant case instead, the speech data was convolved with room impulse responses recorded in a real room, 4.45 m $\times$ 3.55 m wide and 2.50 m high, as shown in the figure. The measured reverberation time was $T_{60} = 130\ ms$ and each impulse response $h_{ji}(n)$ has been used to model the reverberation effects on a sound propagating from source s_i to the j-th microphone.

6.2.3 Results

The Keele database, whose characteristics were described in Section 5.2, was down-sampled to 8 kHz and used to test the proposed system performance [80]. Each of the 10 audio files forming the dataset contain the same sentence uttered by a different speaker. To avoid mixtures contributes to

be all in phase respect to the reference sentence, a 10 seconds segment was extracted, with a different initial offset, from each of them. After that, the 10 obtained segments were taken three at a time and used to derive 20 speech mixtures for the anechoic scenario, applying Equation 6.11.

For the reverberant scenario, the same speech segments were convolved with the room impulse responses h_{ji} introduced in Section 6.2.2, and the resulting signals were then added to produce other 20 mixtures.

To transform signals $s_i(n)$ (anechoic case) or mixtures $x_j(n)$ (reverberant case) into short-time Fourier transforms $X_j(\omega, m)$, an analysis frame and frame shift of 64 ms and 32 ms length respectively, were used.

For estimating the pitch values necessary to drive the comb filters described in Section 6.1.2, three algorithms were tested: WAUTOC, YIN and MPF[2]. WAUTOC and YIN were introduced in Chapter 2 and MPF in Section 4.1.3. Although the latter algorithm is not used here in its multi-microphone derivation, becoming thus similar to the ACF approach, it is tested for comparison purposes and to show the advantages of the frequency domain analysis applied to reverberant signals.

For f_0 estimation, the frame size was set to 30 ms, 40 ms, and 60 ms considering the YIN, WAUTOC (rectangular window) and the MPF (Hamming window) based algorithms, respectively. Pitch values were estimated every 1 ms and the same VUV_k signals were used in all experiments to provide uniform test conditions to the different PDAs employed. These signals were derived from the re-estimated Keele pitch reference values, as explained in Section 5.2.

The parameters of the comb FIR and IIR filters instead, were set to $N_{\text{FIR}} = 5$, $N_{\text{IIR}} = 5$ and $\rho = 0.995$. The values used for the several pa-

[2]The proposed BSS system was also tested using the pitch estimated values provided by the REPS algorithm [67], and the obtained results were reported in [30].

rameters involved, were chosen to obtain the best performance from each pitch estimation algorithm and from the proposed BSS system.

To compare the results provided by the continuous mask based approach and the f_0 based post-processing block, the binary mask based BSS system was used as reference system. The value of Δ used in Equation 6.6 to derive the binary masks was set so that all the values belonging to each estimated cluster were used in the design of the corresponding mask. This also implies that the assignment of each mixture bin is mutually exclusive, that is, every bin from the mixture is used for only one target speaker reconstruction.

Binary mask

The binary mask based BSS approach, described in Section 6.1, is assumed here as the baseline system against which to compare the proposed BSS system. The results obtained with this system, in terms of SIR and SDR are reported in Table 6.1, where the left column refers to the anechoic scenario, and the right column to the reverberant (or echoic) one.

Speech signals acquired in the anechoic scenario better satisfy the sparseness assumption. As a consequence of this, the histograms (Figure 6.2) computed on the estimated DOAs $\theta(\omega, m)$ have well localized and sharp peaks along the θ axes, making the estimation of $\tilde{\theta}_i$ values more reliable.

When the reverberant scenario is considered instead, reverberation causes signals to overlap more in the time-frequency domain. This makes the estimation of $\theta(\omega, m)$ more difficult and less reliable. This in turn explains the performance degradation shown in the table, for both the SIR and SDR values.

Estimating correctly the pitch values from the outputs provided by the binary mask based BSS system, turns out to be a difficult task. In fact, if

Binary mask BSS system (dB)

	anechoic	**echoic**
SIR	13.50	10.65
SDR	11.46	8.92

Table 6.1: SIR and SDR values obtained by the binary mask based BSS system. Left column refers to tests performed in an anechoic scenario, right column reports results measured in a reverberant context.

the results of GER(20) and RMSE(20) obtained from the three considered PDAs on the original Keele signals, are compared with those computed on the outputs of the BSS system, an evident performance degradation occurs. Table 6.2 shows the results obtained using the unprocessed Keele signals, while Table 6.3 shows those obtained after the mixtures were processed by the BSS system. It turns out that the most difficult scenario is the reverberant one, where the best GER(20), provided by the MPF algorithm, was not lower than 16.59%. A better trend is observable for the anechoic case, where the best performance is achieved by the YIN algorithm, with GER(20) = 4.72%.

The better performance demonstrated by the MPF algorithm in reverberant conditions, which reduces to an ACF computed through FFT in the single channel case, further strengthens the hypothesis that the frequency domain based approach is more suitable for processing signals severely deteriorated by reverberation.

Continuous mask

When the estimated DOA for a particular mixture time-frequency bin differs considerably from any estimated centroid $\tilde{\theta}_i$, the probability of speaker superposition is considered to be higher than when the DOA coincides with one of the centroids. In such a case, this time-frequency bin will generate

PDAs performance on Keele database (%)

Keele	WAUTOC	YIN	MPF
GER(20)	5.38	**1.57**	1.93
RMSE(20)	**2.24**	2.35	2.30

Table 6.2: The GER(20) and RMSE(20) obtained processing the Keele database with the WAUTOC, YIN and MPF algorithms, are compared. The original Keele signals were down-sampled to 8 kHz before the estimation was carried out.

PDAs performance on binary mask BSS system (%)

anechoic	WAUTOC	YIN	MPF
GER(20)	7.87	**4.72**	5.59
RMSE(20)	**2.88**	2.97	2.97
echoic	WAUTOC	YIN	MPF
GER(20)	19.25	18.53	**16.59**
RMSE(20)	3.35	**3.32**	3.34

Table 6.3: Performance evaluation of the WAUTOC, YIN and MPF algorithms applied to the output of the binary mask based BSS system. Results, given in terms of GER(20) and RMSE(20), show the deterioration that occurs when reverberant signals are processed (bottom) if compared with the anechoic scenario (top).

distortion in the speaker signal selected for the target, whereas there will be information missing in the spectrograms of the other extracted signals.

To partially overcome this problem, continuous masks are employed, and each mask weight is assigned a value linearly proportional to the distance of the estimated DOA from each centroid, for every bin under consideration.

The resulting SIR and SDR measured on the output of the continuous mask based BSS system, are reported in Table 6.4. Although the SIR measured in echoic conditions slightly decreases, there is an overall improvement in interference, as well as in distortion reduction, for both the echoic and anechoic scenarios. The greater improvement obtained in terms of SDR, demonstrates the advantages of using continuous masks,

particularly in the echoic case where DOA estimation is more difficult.

Continuous mask BSS system (dB)

	anechoic	**echoic**
SIR	13.86	10.55
SDR	12.06	9.83

Table 6.4: SIR and SDR values obtained by the continuous mask based BSS system. Left column refers to tests performed in an anechoic scenario, right column reports results measured in a reverberant context.

Also the pitch estimation results, computed in terms of GER(20) and RMSE(20) on the outputs of the considered BSS system, show an improvement with respect to the binary mask case. These are reported in Table 6.5 which, as expected, reports the same trend but higher rates compared to Table 6.3. The f_0 values obtained by the WAUTOC, YIN and MPF algorithms will be used to tune the adaptive FIR and IIR comb filters described in Section 6.1.2, and the results obtained will be presented in the following.

PDAs performance on continuous mask BSS system (%)

anechoic	WAUTOC	YIN	MPF
GER(20)	7.46	**4.03**	4.77
RMSE(20)	**2.78**	2.86	2.85
echoic	WAUTOC	YIN	MPF
GER(20)	18.25	16.34	**14.62**
RMSE(20)	3.25	**3.22**	3.23

Table 6.5: Performance evaluation of the WAUTOC, YIN and MPF algorithms applied to the output of the continuous mask based BSS system. The better quality output signals provided by this BSS system reflects in higher GER(20) and RMSE(20), compared to the binary mask scenario. The first two rows show the results obtained in the anechoic context, while the bottom part of the table reports the higher error rates relative to the reverberant case.

f_0 driven comb filtering

After applying comb filtering to the BSS outputs obtained with continuous masks, the results shown in Table 6.6 were obtained. Comparing the SIR and SDR values with those from Table 6.4, it could be noted that SIR values generally increased at the expense of SDR values. That is, the f_0 based comb filtering technique proved to be effective for eliminating interference and restoring signal harmonics, though at the expense of introducing little distortion. The highest improvement was that of the SIR value in reverberant conditions, which increased from 10.55 dB to 11.45 dB, after comb filtering was applied. This was obtained employing the f_0 values provided by the MPF algorithm, which in turn produced the lowest GER(20) compared to WAUTOC and YIN, in the same conditions.

Continuous mask + f_0 based post-processing BSS system (dB)

anechoic	WAUTOC	YIN	MPF
SIR	14.46	14.50	**14.51**
SDR	11.78	**11.80**	**11.80**
echoic	WAUTOC	YIN	MPF
SIR	11.32	11.42	**11.45**
SDR	9.49	**9.56**	**9.56**

Table 6.6: Results obtained in terms of SIR and SDR after applying f_0 based comb filtering to the outputs provided by the continuous masks BSS system.

In the anechoic scenario, there are very little variations between the SIR and SDR values that were obtained employing the different PDAs to estimate f_0. Despite YIN provided the best GER(20) score in anechoic conditions (Table 6.5), this is not clearly reflected by the figures of the upper part of Table 6.6. This could be explained considering that, during processing, the impulse response $h(n)$ and the transfer function $H(z)$ of the FIR and IIR comb filters, respectively, are updated at the sample level.

Given that the original time interval between each estimated f_0 is of 1 ms, parabolic interpolation was applied to obtain the pitch values needed in between. Possible octave errors in pitch estimation, will inevitably affect the result of interpolation, but in a way not easy to foresee, since it also depends on the way these errors group together.

Also the fine precision with which f_0 values are estimated influences the comb filtering processing, most of all that based on the FIR filter. The measured GER(1) for the YIN and MPF algorithms, derived in anechoic conditions, were of 31.48% and 29.77%, respectively. This could explain why the use of these algorithms provided almost the same SIR and SDR values in anechoic conditions, while their performance in Table 6.5 were different. The latter consideration does not imply, in this context, the superiority of an algorithm with respect to the other. In fact, none of the considered PDAs was designed to provide very precise f_0 values since, once a pitch estimate is correctly estimated within a neighborhood of the reference, its value is easily refined by many available techniques. Instead, the main point here, is the important role of f_0 information in restoring and enhancing the harmonic structure of speech voiced sections.

This can be verified when observing the relative improvement given by the proposed BSS approach over the baseline system, as reported in Table 6.7.

Even though the comb filtering procedure reduced slightly the SDR values obtained after the continuous mask application, combining the two techniques provided an overall increase of both SIR and SDR values, in both the anechoic and reverberant scenarios.

Relative improvement respect to the reference BSS system (%)

anechoic	WAUTOC	YIN	MPF
SIR	7.11	7.41	**7.48**
SDR	2.79	**2.97**	**2.97**
echoic	WAUTOC	YIN	MPF
SIR	6.29	7.23	**7.51**
SDR	6.39	**7.17**	**7.17**

Table 6.7: Relative improvement obtained by the continuous mask + f_0 based BSS system with respect to the reference BSS system. Results are presented in terms of relative improvement percentages, calculated comparing values from Table 6.6 with those from Table 6.1.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation the problem of f_0 estimation was addressed. In particular the focus was on a design of a f_0 extractor, robust to reverberant and noisy conditions and capable of processing, in a parallel fashion, the speech signal provided by a microphone network. Before describing the proposed algorithm, a review of the state of the art pitch extraction algorithms was given, considering first the algorithms belonging to the early phase of pitch estimation research. The reason for this is that these first approaches to f_0 estimation constitute the basis for many of the modern proposed solutions, which are described thereafter. In particular, the here proposed technique is based on a generalized version of the autocorrelation function.

Then, the human speech production mechanism was recalled, showing the relation between the vocal folds oscillating frequency f_0, which takes place during voiced speech, and the pitch perception. The speech production process can be approximated by means of the source-filter model, which results a very useful tool to analyze the speech signal. The importance of pitch information for speech applications is then highlighted, indicating briefly how f_0 is exploited by some of these applications to improve

their performance. Current speech processing techniques, and the derived commercial products, can provide nowadays very good performance, provided that they are used in contexts characterized by good acoustic conditions. Whenever noisy and reverberant scenarios are addressed instead, performance drops dramatically. The effect of noise and reverberation is thus considered in the course of the dissertation, and a mathematical model for reverberant and noisy speech signals is reported. This will be used to provide examples of the performance degradation of some state of the art pitch extraction algorithms, when tested on such low quality speech signals.

The proposed f_0 estimation algorithm employs a Distributed Microphone Network to guarantee the talker the maximum mobility freedom. The microphone acquisitions are processed in a parallel fashion, performing blind estimation of the reliability of each of them. The latter information is then exploited to derive, from the whole channel set, a common representation which takes more into account those contributes showing a similar harmonic structure. Signals are principally processed in the frequency domain applying FFTs, making the algorithm computational cost low and making the proposed solution suitable for real-time processing.

Regarding the speech datasets used for performance evaluation, real-world speech acquisitions were used. These accounted for professional talkers uttering phonetically balanced sentences, and spontaneous speech recorded during seminars and meetings. Tests based on speech data artificially obtained, that is, derived exploiting the described model for reverberant and noisy speech, were not carried out. This because the latter model represents just an approximation of the actual acoustic conditions of a real-world scenario.

As a first indication provided by the given experimental results, it can

be stated that the frequency domain based analysis shall be preferred to the time domain one, in case distributed-microphone speech signals are processed. In fact, if the average of the results obtained using each single microphone are considered for the office scenario (both P1 and P2 loudspeaker position), a GRE(20)=10.2% is provided by the MPF, compared to the best result obtained employing state of the art algorithms, which was 13.03%. When all channels are processed in parallel, MPF further reduces the error providing GER(20)=6.19%, which result closer to the performance obtained by the considered algorithms, when tested on close-talk signals. This considerations are confirmed by the results relative to the CHIL scenario.

The proposed microphone setup is to be considered as a not yet explored sensor arrangement for pitch estimation. For this reason, the work described in this thesis represents just one of the possible approaches based on multi-microphone speech input. Although traditional beamforming techniques cannot be applied to the outputs of a DMN, other fusion methods, designed to derive an enhanced signal representation from the reverberant channels, can be devised.

To demonstrate the importance of pitch information for speech applications, a chapter of this thesis is dedicated to the description of a f_0 based Blind Source Separation (BSS) system. The BSS system taken as a reference is a binary mask based blind source separation system. This system separates each talker contribute from a mixture of three speakers uttering simultaneously. To improve the baseline BSS system separation performance, the proposed approach bases on continuous masks and on a scheme of f_0 based adaptive comb filters. The comb filters are employed to restore the harmonic structure of each separated signal voiced segments. The usefulness of pitch information for the given application is confirmed

by the improvements obtained respect to the baseline BSS system. Considering, for example, the reverberant scenario, a relative improvement of 7.51% and 7.17% was obtained in terms of SIR and SDR, respectively.

7.2 Future work

Estimating f_0 from speech signals provided by a Distributed Microphone Network represents a challenging operation. The main difficulties, faced during the design of the proposed algorithms, were the poor acoustic quality of the analyzed speech signals, due to the strong reverberation effects, and the consequent waveform diversity, observable between each pair of microphones.

The proposed solution, though representing just one of the possible approaches based on a DMN setup, expands the current f_0 estimation research field to a context in which many f_0 based speech applications can be tested.

Future work can be thus organized along at least two research lines. The first line shall focus on the development of alternative strategies to exploit the information redundancy offered by the DMN. Provided that traditional beamforming techniques cannot be applied to the DMN outputs, otherwise spatial aliasing would take place, different beamforming approaches shall be addressed. Referring to the here proposed one, a more sophisticated method can be devised to evaluate each channel acoustic reliability. For example, statistical methods can be employed to establish whether, and to what extent, a given channel contributes for a correct and accurate f_0 estimation, or not.

The second research line shall consider the use of f_0 information, extracted in the above described environment, to increase the robustness of speech applications, adapted to work in the same context. For example,

f_0 can be used to perform speech prosodic analysis, exploited, in turn, for emotion detection and for spontaneous speech recognition. Another context, where f_0 information would result useful, is in the presence of the cocktail party effect, that is, when multiple speakers are active simultaneously. Tracking pitch variations of each talker would help in interpreting the acoustic scene, so that the separation of each speaker message could result easier.

Another application that can take advantage of pitch information is the Acoustic Event Detection, which is currently under development at the ITC-irst[1] research center in the context of the CHIL project. For it, an extension of the proposed pitch extraction approach, to include environmental periodic acoustic events, can be derived. The new source of information, f_0, will then be used to better detect and classify non-speech events[2].

Non-speech periodic audio patterns can be also detected in order to perform environmental classification. An example for this is provided by the typical periodic noise produced by a car engine or tires. Once the car environment is automatically detected, scenario dependent parameters can be consequently set in order to increase the robustness of speech applications, such as ASR, in the given context.

[1]Centro per la Ricerca Scientifica e Tecnologica, Trento, Italy.

[2]This information can be used, in turn, to avoid that the speech recognizer associates a nonsense transcription to a non-speech event, such as, for example, a cough.

Bibliography

[1] T. Abe, T. Kobayashi, and S. Imai. Harmonics tracking and pitch extraction based on instantaneous frequency. *IEEE Trans. ASSP*, 1:756–759, 1995.

[2] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 1991.

[3] F. Anderson. An experimental pitch indicator for training deaf scholars. *J. Acoust. Soc. Am.*, 32:1065–1074, 1960.

[4] S. Araki, S. Makino, A. Blin, Mukai R., and H. Sawada. Undetermined blind separation for speech in real environments with sparseness and ICA. In *Proc. IEEE ICASSP*, volume 3, pages 881–884, 2004.

[5] S. Araki, S. Makino, H. Sawada, and R. Mukai. Underdetermined blind separation of convolutive mixtures of speech with directivity pattern based mask and ICA. In *Proc. ICA*, pages 898–905, 2004.

[6] L. Armani and M. Omologo. Weighted autocorrelation-based f_0 estimation for distant talking interaction with a distributed microphone network. In *Proc. IEEE ICASSP*, volume 1, pages 113–116, 2004.

[7] B. S. Atal. Automatic speaker recognition based on pitch contours. *J. Acoust. Soc. Am.*, 52(6):1687–1697, 1972.

[8] B. S. Atal and S. Hanauer. Speech analysis and synthesis by prediction of the speech wave. *J. Acoust. Soc. Am.*, 50:637–655, 1971.

[9] L. L. Beranek. *Concert and Opera Halls: How They Sound*. Acoustical Society of America, New York, 1996.

[10] A. Blin, S. Araki, and S. Makino. Blind source separation when speech signals outnumber sensors using a sparseness-mixing matrix combination. In *Proc. IWAENC*, pages 211–214, 2003.

[11] P. Boersma. PRAAT, a system for doing phonetics by computer. *Glot International*, 5:341–345, 2001.

[12] P. Bofill and M. Zibulevsky. Blind separation of more sources than mixtures using sparsity of their short-time Fourier transform. In *Proc. ICA*, pages 87–92, 2000.

[13] S. F. Boll. Speech enhancement in the 1980s: Noise suppression with pattern matching. In S. Furui and M.M. Sondhi, editors, *Advances in Speech Signal Processing*, chapter 10. Marcel Dekker, 1991.

[14] J.P. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85:1437–1462, September 1997.

[15] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[16] S. Coren, L. M. Ward, and J. T. Enns. *Sensation and Perception*. Wiley, New York, 6th edition, 2003.

[17] A. de Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.*, 111:1917–1930, 2002.

[18] R. De Mori. *Spoken Dialogues with Computers.* Academic Press, Inc., Orlando, FL, USA, 1997.

[19] P. B. Denes and E. N. Pinson. *The Speech Chain; the Physics and Biology of Spoken Language.* Freeman, New York, 2nd edition, 1993.

[20] L. O. Dolansky. Instantaneous pitch period indicator. *J. Acoust. Soc. Am.*, 26:953, 1954.

[21] L. O. Dolansky. An instantaneous pitch period indicator. *J. Acoust. Soc. Am.*, 27:67–72, 1955.

[22] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner. Real time digital hardware pitch detector. *IEEE Trans. ASSP*, 24:2–8, 1976.

[23] H. Duifhuis, L. F. Willems, and R. J. Sluyter. Measurement of pitch in speech; an implementation of Goldstein's theory of pitch perception. *J. Acoust. Soc. Am.*, 71:1568–1580, 1982.

[24] G. Fant. *Acoustic Theory of Speech Production.* Mouton, The Hague, 1970.

[25] G. Fant. *Acoustic Theory of Speech Production : with Calculations Based on X-Ray Studies of Russian Articulations.* Mouton, The Hague, 2nd edition, 1970.

[26] M. Filip. Some aspects of high-accuracy analogue fundamental frequency recordings. *ICPhS-6*, pages 319–322, 1967.

[27] M. Filip. Envelope periodicity detection. *J. Acoust. Soc. Am.*, 45:719–732, 1969.

[28] J. Flanagan. *Speech Analysis and Perception.* Springer-Verlag, 2 edition, 1972.

[29] J. L. Flanagan, J. D. Johnston, R. Zahn, and G.W. Elko. Computer-steered microphone arrays for sound transduction in large rooms. *J. Acoust. Soc. Am.*, 78:1508–1518, Nov. 1985.

[30] F. Flego, S. Araki, H. Sawada, T. Nakatani, and S. Makino. Underdetermined blind separation for speech in real environments with F0 adaptive comb filtering. In *Proc. IWAENC*, pages 93–96, 2005.

[31] F. Flego, L. Armani, and M. Omologo. On the use of a weighted autocorrelation based fundamental frequency estimation for a multi-dimensional speech input. In *Proc. ICSLP*, pages 2441–2444, 2004.

[32] R. H. Frazier, S. Samsam, Braida L. D., and A. V. Oppenheim. Enhancement of speech by adaptive filtering. In *Proc. IEEE ICASSP*, pages 251–253, 1976.

[33] D. H. Friedman. Pseudo-maximum likelihood pitch estimation. *IEEE Trans. ASSP*, 25:418–221, 1977.

[34] D. Giuliani, M. Matassoni, M. Omologo, and P. Svaizer. Training of HMM with filtered speech material for hands-free recognition. In *Proc. IEEE ICASSP*, volume 1, pages 449–452, 1999.

[35] B. Gold. Digital speech networks. *Proc. IEEE*, 65:1636–1658, 1977.

[36] B. Gold and N. Morgan. *Speech and audio signal processing; Processing and perception of speech and music.* John Wiley & Sons, 2000.

[37] B. Gold and L. R. Rabiner. Parallel processing technique for estimating pitch period of speech in the time domain. *J. Acoust. Soc. Am.*, 46:442–448, 1969.

[38] J. L. Goldstein. An optimal processor theory for the central formation of the pitch of complex tones. *J. Acoust. Soc. Am.*, 35:1358–1366, 1973.

[39] M. Goto. A predominant-F0 estimation method for polyphonic musical audio signals. In *Proc. ICA*, pages 1085–1088, 2004.

[40] 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*, 43(4):311–329, 2004.

[41] O. O. Gruenz and L. O. Schott. Extraction and portrayal of pitch of speech sounds. *J. Acoust. Soc. Am.*, 21:487–495, 1949.

[42] R. W. Hamming. *Coding and Information Theory.* Prentice Hall, Englewood Cliffs, NJ, 1980.

[43] W. Hess. *Pitch Determination of Speech Signals.* Information Sciences. Springer-Verlag, second edition, 1983.

[44] D. M. Howard. Digital peak-picking fundamental frequency estimation. *Speech Hearing and Language - Work in Progress, London: UCL*, 2:151–164, 1986.

[45] D. M. Howard. Peak-picking fundamental frequency estimation for hearing prostheses. *J. Acoust. Soc. Am.*, 86:902–910, 1989.

[46] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall PTR, April 2001.

[47] M. Huckvale. Sfs: Speech Filing System, 1987-1998. http://www.phon.ucl.ac.uk/resource/sfs.html.

[48] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley & Sons, 2001.

[49] H. Indefrey, W. Hess, and G. Seeser. Design and evaluation of double-transform pitch determination algorithms with nonlinear distortion in

the frequency domain - preliminary results. In *Proc. IEEE ICASSP*, volume 1, pages 415–418, 1985.

[50] K. Kinoshita, T. Nakatani, and M. Miyoshi. Fast Estimation of a Precise Dereverberation Filter based on Speech Harmonicity. In *Proc. IEEE ICASSP*, volume 1, pages 1073–1076, 2005.

[51] A. Klapuri. *Signal Processing Methods for the Automatic Transcription of Music*. PhD thesis, Tampere University of Technology, Tampere, Finland, 2004.

[52] H. Kuttruff. *Room Acoustics*. Elsevier Applied Science, London, UK, 3rd edition, 1991.

[53] C. E. Liedtke. Rechnergesteuerte sprachzeugung. Technical Report Nr. 137, Heinrch-Hertz-Institut, Berlin, 1971.

[54] D. Macho et al. Automatic speech activity detection, source localization, and speech recognition on the CHIL seminar corpus. *Proc. ICME*, 2005.

[55] J. Makhoul. Linear prediction: A tutorial review. *Proc. IEEE*, 63(4):561–580, 1975.

[56] J. D. Markel. The sift algorithm for fundamental frequency estimation. *IEEE Trans. Audio and Electroacoustics*, 20:367–377, 1972.

[57] J. D. Markel and A. H. Gray. *Linear Prediction of Speech, Communications and Cybernetics*, volume 12. Springer, Berlin and New York, 1976.

[58] C. Marro, Y. Mahieux, and K. Uwe Simmer. Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering. *IEEE trans. Speech and Audio Processing*, 6(3):240–259, May, 1998.

[59] P. Martin. Mesure de la fréquence fondamentale par intercorrélation avec une fonction peigne. *XIIéme Journées d'étude sur la parole, Montrèal*, pages 221–232, 1981.

[60] P. Martin. Comparison of pitch detection by cepstrum and spectral comb analysis. *IEEE Trans. ASSP*, pages 180–183, 1982.

[61] N. P. McKinney. Laryngeal frequency analysis for linguistic research. *Report No. 14. Ann Arbor: Communication Sciences Laboratory, Univ. of Michigan*, 1965.

[62] Y. Medan, E. Yair, and D. Chazan. Super resolution pitch determination of speech signals. *IEEE Trans. Signal Processing*, 39:40–48, 1991.

[63] A. R. Meo and G. U. Righini. A new technique for analyzing speech by computer. *Acustica*, (25):261–268, November 1971.

[64] N. J. Miller. Pitch detection by data reduction. *IEEE Symp. Speech Recognition, Paper T9*, pages 122–128, 1974.

[65] N. J. Miller. Pitch detection by data reduction. *IEEE Trans. ASSP*, 23:72–29, 1975.

[66] T. Nakatani and T. Irino. Robust fundamental frequency estimation against background noise and spectral distortion. *Proc. ICSLP*, 3:1733–1736, 2002.

[67] T. Nakatani and T. Irino. Robust and accurate fundamental frequency estimation based on dominant harmonic components. *J. Acoust. Soc. Am.*, 116(6):3690–3700, 2004.

[68] A. Nehorai and B. Porat. Adaptive comb filtering for harmonic signal enhancement. *IEEE Trans. ASSP*, ASSP-34:1124–1138, Oct. 1986.

[69] H. Ney. A time warping approach to fundamental period estimation. *IEEE trans. SMC*, 12:383–388, 1982.

[70] L. P. Nguyen and S. Imai. Vocal pitch detection using generalized distance function associated with a voiced-unvoiced decision logic. *Bull. P.M.E. (T.1.T)*, 39:11–21, 1977.

[71] A. M. Noll. Short time spectrum and cepstrum techniques for vocal pitch detection. *J. Acoust. Soc. Am.*, 36:296–302, 1964.

[72] A. M. Noll. Cepstrum pitch determination. *J. Acoust. Soc. Am.*, 41:293–309, 1967.

[73] A. M. Noll. Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In *Proc. Symp. Comput. Processing Commun.*, pages 779–797, 1969.

[74] D. O Shaughnessy. *Speech communications, human and machine.* IEEE Press, second edition, 2000.

[75] M. Omologo, M. Matassoni, and P. Svaizer. Speech recognition with microphone arrays. In *Microphone Arrays Signal Processing Techniques and Applications*, pages 331–353. Published in Brandstein M., Ward D. (eds.), Springer, 2001. Ref. No.: 0012-18.

[76] M. Omologo, M. Matassoni, P. Svaizer, and D. Giuliani. Microphone array based speech recognition with different talker-array positions. In *Proc. IEEE ICASSP*, pages 227–230, 1997.

[77] M. Omologo, P. Svaizer, and M. Matassoni. Environmental conditions and acoustic transduction in hands-free speech recognition. *Speech Communication*, 25:75–95, 1998.

[78] A.V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1999.

[79] K. K. Paliwal and P. V. S. Rao. A synthesis-based method for pitch extraction. *Speech Communication*, 2(1):37–45, 1983.

[80] F. Plante, G. F. Meyer, and W. A. Ainsworth. A pitch extraction reference database. *Proc. EUROSPEECH*, pages 837–840, 1995.

[81] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, N.J., USA, 1993.

[82] L. R. Rabiner. On the use of autocorrelation analysis for pitch detection. *IEEE Trans. ASSP*, 25:24–33, 1977.

[83] L. R. Rabiner. "a tutorial on hidden markov models and selected applications in speech recognition". In *Proc. IEEE*, volume 77, pages 257–286, Feb. 1989.

[84] L. R. Rabiner, M. H. Cheng, A. E. Rosenberg, and C. A. McGonegal. A comparative study of several pitch detection algorithms. *IEEE Trans. ASSP*, 24:399–413, 1976.

[85] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs,NJ, 1978.

[86] D. R. Reddy. An approach to computer speech recognition by direct analysis of the speech wave. Technical Report CS-49, Stanford University, Berkeley, CA, 1966.

[87] D. R. Reddy. Pitch determination of speech sounds. *Communications of the ACM*, 10:343–348, 1967.

[88] S. Rickard and O. Yilmaz. On the W-disjoint orthogonality of speech. In *Proc. IEEE ICASSP*, volume 1, pages 529–532, 2002.

[89] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. J. Manley. Average magnitude difference function pitch extractor. *IEEE Trans. ASSP*, 22:353–362, 1974.

[90] S. Sagayama and S. Furui. Pitch extraction using the lag window method. *Proc. of IECEJ*, 1978. (in Japanese).

[91] F. J. Sanchez Gonzales. Application of dissimilarity and aperiodicity function to fundamental frequency measure of speech and voiced/unvoiced decision. *9th International Congress on Acoustics*, 17:523, 1977.

[92] F. J. Sanchez Gonzales. Dissimilarity and aperiodicity functions. temporal processing of quasi-periodic signals. *9th International Congress on Acoustics*, 13:859, 1977.

[93] K. Schafer-Vincent. Pitch period detection and chaining: Method and evaluation. *Phonetica*, 40(3):177–202, 1983.

[94] P. Schniter. Time-frequency uncertainty principle, Oct. 2005. http://cnx.rice.edu/content/m10416/2.18/.

[95] M. Schroeder and B. Atal. Code excited linear prediction (CELP): High quality speech at very low bit rates. *Proc. IEEE ICASSP*, pages 937–940, 1985.

[96] M. R. Schroeder. Period histograms and product spectrum: New methods for fundamental frequency measurement. *J. Acoust. Soc. Am.*, 43:829–834, 1968.

[97] M. R. Schroeder. Parameter estimation in speech: A lesson in unorthodoxy. *Proc. IEEE*, 58:707–712, 1970.

[98] T. Shimamura and H. Kobayashi. Weighted autocorrelation for pitch extraction of noisy speech. *IEEE trans. SAP*, 9(7):727–730, Oct, 2001.

[99] H. Singer and S. Sagayama. Pitch dependent phone modelling for HMM based speech recognition. In *Proc. IEEE ICASSP*, volume 36, pages 273–276, 1992.

[100] K. Sjölander and J. Beskow. WaveSurfer - an open source speech tool. In *Proc. ICSLP*, volume 4, pages 464–467, 2000.

[101] C. P. Smith. Device for extracting the excitation function from speech signals. United States Patent No. 2,691,137, Oct 1954.

[102] C. P. Smith. Speech data reduction: Voice communication by means of binary signals at rates under 1000 bits/sec. Technical Report MA, DDC-AD-117290, AFCRC, 1957.

[103] M. M. Sondhi. New methods of pitch extraction. *IEEE trans. Audio Elettroacoust.*, Av-16:262–266, Jun, 1968.

[104] M. K. Sönmez, L. Heck, M. Weintraub, and E. Shriberg. A lognormal tied mixture model of pitch for prosody based speaker recognition. *Proc. EUROSPEECH*, 3:1391–1394, 1997.

[105] T. Stephenson, J. Escofet, M. Magimai-Doss, and H. Bourlard. Dynamic bayesian network based speech recognition with pitch and energy as auxiliary variables. In *Proc. IEEE NNSP*, pages 637–646, 2002.

[106] H. W. Strube. Determination of the instant of glottal closure from the speech wave. *J. Acoust. Soc. Am.*, 56(5), Nov. 1974.

[107] Y. Suzuki, F. Asano, H. Y. Kim, and T. Sone. An optimum computer-generated pulse signal suitable for the measurement of very long impulse responses. *J. Acoust. Soc. Am.*, 97, 1995.

[108] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. *IEEE trans. Speech and Audio Processing*, 8(6):708–716, Nov, 2000.

[109] B. Truax. *Handbook for Acoustic Ecology.* A.R.C. Publications, Vancouver, 1978.

[110] D. Van Compernolle. Spectral estimation using a log-distance error criterion applied to speech recognition. *Proc. IEEE ICASSP*, pages 258–261, 1989.

[111] J. D. Wise, J. R. Caprio, and T. W. Parks. Maximum likelihood pitch estimation. *IEEE Trans. ASSP*, 24:418–423, 1976.

[112] L. A. Yaggi. Full duplex digital vocoder. Technical Report Nr. sp16-A63, Texas Instruments, 1963.

[113] C. Zmarich and S. Bonifacio. I piani formantici acustici e uditivi delle vocali di infanti, bambini, e adulti maschi e femmine. In P. Cosi, E. Caldognetto Magno, and A. Zamboni, editors, *Voce, Canto, Parlato. Studi in onore di Franco Ferrero*, pages 311–320. Unipress, Padova, 2003.

Appendix A

Time-frequency Uncertainty Principle

When studying wave or signal properties in the signal processing field, it holds a principle which is analogue to the Heisenberg uncertainty principle of quantum theory. The time and frequency quantities in signal processing turn out to be analogous to the position and momentum of a particle in space in quantum physics.

The time-frequency uncertainty principle states that it is not possible to determine exactly the frequency of a given signal at a precise time instant. In fact, whenever the frequency has to be determined, there is the need to process a finite length of the given signal, thus making time precision less accurate. The opposite holds too, that is generally, the shorter the time segment is chosen, the less accurate will result the signal frequency estimation.

In the signal processing field, the Fourier transform represents a common tool which permits to characterize linear systems and to identify the frequency components making up a continuous or a sampled waveform.

Let consider the Fourier transformation for a generic continuous signal $x(t)$,

$$X(f) = \int_{-\infty}^{\infty} x(t) \ \left[e^{j2\pi ft}\right]^* \ dt, \tag{A.1}$$

the complex conjugate term $e^{j2\pi ft}$ against which $x(t)$ is integrated, represents the transformation kernel. Indicating with $b_p(t)$ a generic kernel function, being p a parameter, the above Fourier kernel can be written as $b_p(t) = e^{j2\pi pt}$ and provides poor time resolution. In fact this particular kernel is defined over all $t \in (-\infty, \infty)$ and, its Fourier transform $B_p(f)$ is a Dirac delta centered in p, $\Delta(f-p)$, being thus well localized in frequency.

On the other hand, if the kernel in the transformation in (A.1) were set to $b_p(t) = \Delta(t-p)$, optimal time resolution will be provided, but no frequency resolution at all will be available. This is also evident considering the absolute value of the Fourier transform of $b_p(t)$, which is $|B_p(f) = 1|$ for all frequencies regardless of the parameter p.

Defining with Δ_t^2 and Δ_f^2 the variances of $b_p(t)$ and $B_p(f)$, respectively, the time-bandwidth product $\Delta_t\Delta_f$ depends on the particular choice of $b_f(t)$ and holds the *time-frequency uncertainty principle*

$$\Delta_t\Delta_f \geq \frac{1}{2}. \tag{A.2}$$

This limits the time and frequency resolutions achievable with a particular kernel, being them tied by equation (A.2). The lowest achievable value for the time-bandwidth product is $\Delta_t\Delta_f = \frac{1}{2}$ and is provided by the Gaussian pulse,

$$b(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}, \qquad B(f) = e^{-\frac{1}{2}f^2}, \tag{A.3}$$

which will thus provide the best joint time-frequency resolution.

Both equations in (A.3) are neither band-limited nor time-limited, but concentrated around their mean (which is zero) thus providing the lowest

joint variances [94].

Appendix B

Characteristics of the Reference Pitch Values

During my research work on pitch estimation, one of the most interesting and profitable discussion that I had, was with one of the authors of [80], which I esteem and thank very much. The object of the discussion were, on the one hand, the *characteristics* that the pitch reference values, used to test the pitch extraction algorithms performance, should possess. On the other hand, the *method* to obtain these values. As a result of this discussion and considering the experience that I gained in this field through the Ph.D. experience, I'd like to resume here some convictions that I matured about it.

Method

A proposed method to obtain the pitch references is to derive them applying the PDA that has to be tested to the laryngograph signal, when available. Even if the latter signal permits to obtain precise and reliable pitch estimates, in my humble opinion, it would be preferable to use a state of theart PDA to create the references, or a combination of such PDAs, merging then the final results, as showed in Section 5.3. This because it is very likely that a pitch extraction algorithm makes some octave errors, even

when run on the laryngograph signal. In case no post processing is applied by the algorithm or no further checking of the results is done manually, the risk is to have wrong pitch references. Therefore, when the considered PDA is tested on the speech signal, it could occur that it correctly estimates the pitch value whose reference is wrong, thus underestimating its performance. If instead, the PDA provides a wrong estimate for the considered value, it is possible that it matches the wrong reference and the device performance will thus result overestimated.

Characteristics

Another issue is represented by the characteristics of the reference pitch estimates. As explained in Chapter 2, PDAs can be classified into *time domain* and *short term analysis* based algorithms [43]. While in the former case, the pitch values provided reflects precisely the duration of each pitch period, in the latter case each estimate can be considered an "average" of several contiguous pitch periods. Adopting the short term analysis approach provides generally a more robust pitch estimation, given that redundant information can be exploited for guessing its value. However, the drawback introduced is represented by the limited precision with which each pitch period is estimated. In addition, voiced/unvoiced selection based on this approach can only approximately locate the start/end points of the voicing sections in the speech signal, compared with the time domain based algorithms.

It seems obvious that, independently from the target application, a PDA providing pitch estimates both reliable and with a pitch period level precision, would be preferable to other devices lacking of one or the other capability[1]. It seems obvious too, that the reference labels provided with

[1]The statement refers to PDAs that perform "fundamental frequency" estimation. Different would be the case of pitch estimation, when the term "pitch" is given the meaning of subjective perception.

a given speech database, shall have the same characteristics, that is, they shall be very precise at the pitch period level and reliable. Such reference pitch values would thus represent the upper bound quality achievable by a PDA and shall represent, in my humble opinion, the unique term of comparison for all PDAs that have to be tested. Otherwise, adapting the reference pitch values to the reliability or precision characteristics of the tested PDA, will provide a biased feed-back on its performance, not even useful to be compared with the performance obtained by other devices.

Appendix C

Generalized Autocorrelation

The generalized autocorrelation function $\mathrm{ACF}_g(\tau)$ of a signal $x(n)$ can be computed by means of the Discrete Fourier Transform (DFT) and its inverse (IDFT) as follows

$$\mathrm{ACF}_g(\tau) = \mathrm{IDFT}\{|\mathrm{DFT}\{x(n)\}|^g\} \tag{C.1}$$

where the parameter g determines the magnitude compression of the spectral representation. When $g = 2$ Equation C.1 provides the ACF function as it was described in Section 2.3.1, with the difference that, when the time-domain is used for its computation, it is not possible to apply any spectral compression as can instead be specified in the above equation.

Spectral compression can be also obtained applying other non-linear functions to the magnitude spectrum, as for example the logarithmic function which was used to derive the cepstrum, described in Section 2.3.2.

The effects of spectral compression on pitch extraction accuracy were initially studied in [49]. In this work PDAs based on cepstrum and on ACF_2, ACF_1 and $\mathrm{ACF}_{0.5}$ were tested on different acoustic conditions. Results demonstrated the more robustness of ACF_2 to noise but also its worse performance when applied to clean speech signals. This can be explained considering that raising to the second power the speech spectrum empha-

sizes spectral peaks in relation to noise but, at the same time, flattens spectrum dynamics.

On the contrary, cepstrum was reported to perform better than ACF_2 on clean speech signals, but rather poorly on noisy signals. The authors conclusions on the four considered approaches are that PDAs based on $\text{ACF}_{0.5}$ and ACF_1 demonstrated to be *"less sensitive to noise than the cepstrum and less sensitive to strong formants than the autocorrelation PDA and thus represent a good compromise when the environmental signal conditions are unknown"*.

Spectral compression was also tested in [108] where a multi-pitch extractor is described. In this case a value of $g = 0.67$ is experimentally found to provide the best results when the proposed PDA is tested on synthetic harmonic tones with added Gaussian noise at different SNR levels.

In this thesis a value of $g = 1$ was chosen for deriving the MPF function described in Section 4.1.3. This value was suggested by some preliminary tests conducted on a large amount of reverberant speech data and reported in Figure C.1. In these experiments the MPF algorithm was tested on both the Keele and CHIL databases and their relative scenarios (see Section 5.2 and 5.3), varying the g parameter of Equation C.2, which was used instead of Equation 4.7, in the range $0.2 \div 2$.

$$X_i(k) = |\text{FFT}\{\mathbf{x}_i^w\}(k)|^g, \quad 1 \leq k \leq N_f \tag{C.2}$$

The figures shows the GER(20) obtained for each value of g considering first the close-talk signals (left panels), then the reverberant outputs of the Distributed Microphone Network employed (right panels).

The results show that in all conditions the lowest GER(20), indicated

with a red circle, is obtained using $g < 2$. Apart from the close-talk version of the CHIL spontaneous speech database, for which $g = 1.7$ provided the lowest GER(20), in all other cases the best mpf performance were obtained with $0.5 \leq g < 1$.

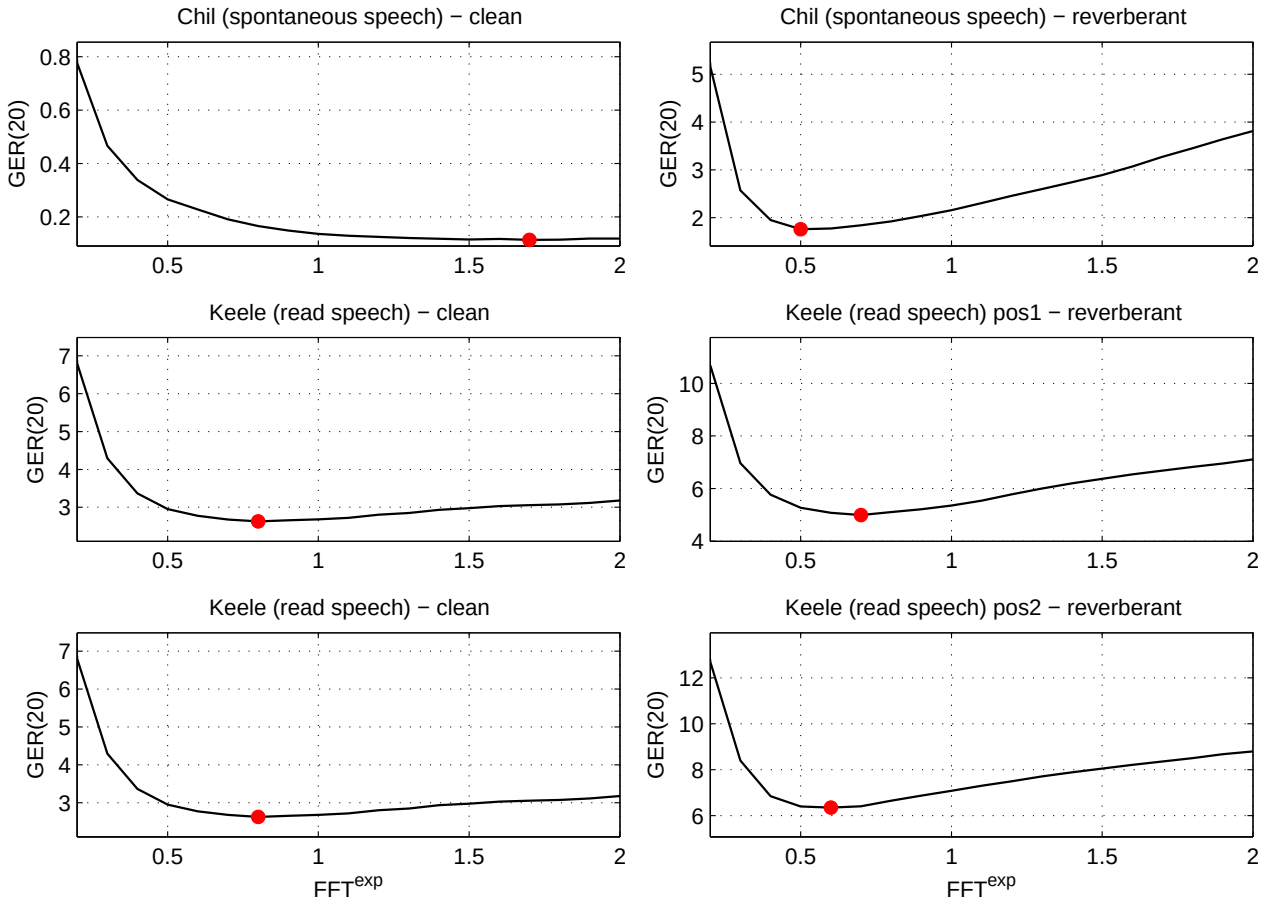

Figure C.1: *Dependency of the mpf GER(20) on the parameter g of Equation C.2.*

Another interesting indication is that the plotted curves resulted more flat in the case of clean speech signals, thus implying a less strong dependence of the mpf accuracy on the parameter g.

In the case of reverberant speech data instead (right panels), a stronger influence of the parameter g on the GER(20) appears. Also, despite the reverberant CHIL and Keele databases represent very different type of speech signals (real and spontaneous versus reproduced and read speech),

there is a strong indication that a value for g close to 0.5 results beneficial for pitch extraction from reverberant speech signals.

Given these results, it could have been possible to set g to achieve the best results for each given scenario. Alternatively, the average of the best g values measured in the different contexts could have been used.

However, to avoid the introduction of a critical parameter to be estimated from the data, turning thus the performance of the proposed mpf function strongly dependent on the given task, $g = 1$ was used for all tests described in Section 5. Setting $g = 0.5$ would have provided even better performance than that reported, but it was considered not a good general choice given that for $g < 0.5$, as shown in the figure, GER(20) starts to increase noticeably.